VLBA Technical Report No. 8

VLBA FOCUS-ROTATION CONTROL SYSTEM

VOLUME I

David Weber and Wayne Koski
June 1990

TABLE OF CONTENTS

1.0 INTRODUCTION

This two-volume manual describes the VLBA Focus-Rotation Control System and was written to serve as the maintenance guide for the control system and modules. The purpose of this manual is to provide a comprehensive functional description of the operation of the system, modules and control firmware. The manual is bound in two volumes because a single volume would be too bulky for convenient handling; secondly, having the descriptive text in one volume and the drawings in the other facilitates reference to the drawings when reading the descriptive text. Volume I contains the system description and Volume II contains the system drawings and component data sheets.

Manual Overview

This manual answers the basic maintenance question: How does the VLBA F-R System work? The answer is in the form of a detailed description of the operation of the three system components: 1) the FRM mechanism and its drive-readout equipment; 2) The F-R Control System which drives the mechanism and reads out the states of the FRM; and 3) the control-data algorithms and firmware which control the FRM, reads the system states, and interacts with the Antenna Control Computer.

Because the FRM mechanism performs a fundamental function on the VLBA antenna and is the device to be controlled, it is vital to understand what it does, how it works, and how it is aligned. This Section (1.0) briefly describes this manual, the FRM mechanism and the location of the FRM drive components. Section 1.1 describes the system specifications. Section 2.1 describes the FRM drive motor-position readout packages. Section 2.2 is a detailed description of the Focus Drive Mechanism. Section 2.3 is a detailed description of the Rotation Drive Mechanism. Section 3.14 describes the alignment of the position readout resolvers, limit switches, and the S101 and S102 modules.

The F-R Control System performs the function of positioning the mechanism drives; reading the mechanism's positions, velocities and alarm conditions; and interacting with the antenna control computer. The F-R Control System is the interface agent of the control firmware. Sections 3.1 through 3.6 describe F-R Control System components and Sections 3.10 and 3.11, respectively, describe cable junction boxes and the isolation transformer box.

The control-data algorithms and the firmware which implement these algorithms are the heart of the control system - they control everything. Although there are components external to the control processor which protect the mechanism from faults and overload conditions, no action takes place outside the purview of the control firmware. Sections 3.7 and 3.8 describe the control algorithms and firmware. Control Firmware listings are included in Section 7.0.

Special system features protect the FRM and control system from malfunctions and lightning-induced transient currents. Section 3.12 describes these features.

Practical information, useful at an antenna or on the lab bench, is provided in many drawings, figures and tables. These include isometric views of the FRM and drives, the layout of the Pedestal Room components, block diagrams, signal flow diagrams, timing diagrams, module schematic and assembly drawings, cable drawings and bin wire lists. Section 7 lists the F-R Control system drawings and Volume II contains these drawings. Tabular data includes limit switch settings, control-data formats and lists of analog and interface signals.

Several specialized components are used in the FRM and Focus-Rotation Control System. The characteristics of these devices are very important system and module parameters; Volume II contains a complete set of data sheets for these components.

Electronics modules and bin fabrication and assembly details are not included since this manual is primarily a maintenance manual but this information may be obtained by reference to fabrication drawings which are listed in Section 7.0.

Telescope Operator information is included in Section 6.0.

System scaling parameters and limit values are tabulated in Section 4.0.

The asssembly and alignment of the FRM mechanism is not included in this manual.

1.1 Focus - Rotation System Description

FRM Description

The FRM is a mechanism mounted at the VLBA antenna apex which supports the subreflector by means of the subreflector support tube. Figure 1 (following this section) depicts the FRM and major Apex components.

The subreflector is the secondary reflective surface for the antenna feed system. Astronomical signals are reflected from the Antenna primary surface to the Subreflector, which in turn reflects them to receiver feed horns located on the locus of a 1.7 meter diameter feed circle centered on the primary reflective surface axis. The subreflector is shaped so as to direct the astronomical signal into one of the 10 receiver feed horns; band selection is accomplished by rotating the subreflector to the appropriate angular position. This positioning is defined as Rotation motion. The signal is focused in the selected feed horn by raising or lowering the Subreflector on the dish axis. This positioning is defined as Focus motion. Both motions are accomplished by the FRM under control of the F-R Control System as commanded by the antenna control computer. Figure 1 shows the senses of Focus and Rotation motions.

The FRM is notable for mechanical simplicity; the two drives are implemented with a minimum of moving parts. Referring to the section view, Figure 2, the FRM is seen to be a cylindrical assembly consisting of three large cast aluminum rings, four precision guide rods, two lead screws, three motors, two bevel gear boxes and a flexible shaft. The structural members are the three aluminum rings and the four guide rods. The top and bottom rings are fixed to the guide rods; this assembly forms a structural cage for the two drive mechanisms. The cage is attached to the Apex ring by four mounting feet which are bolted to the top ring.

The Focus drive mechanism consists of the middle ring, leadscrews, bevel gearboxes, flexible shaft and a drive motor. The middle ring moves axially on the four precision guide rods; this provides the Focus motion. The middle ring is positioned by rotating two synchronized screws which are driven by the bevel gear boxes mounted directly above the screws. A flexible shaft between the gearboxes synchronizes the screw motion. The Focus motor, fixed to one of the gearboxes, drives the Focus mechanism. Figure 7 illustrates the essentials of the Focus drive mechanism.

The Rotation drive mechanism is mounted on the middle ring (also called traveling platform) which has a 46.5 inch diameter ball bearing race attached to the inner diameter; the rotating portion of the bearing is fixed to the subreflector support tube which supports the subreflector. Two rotation drive motor packages are mounted on the traveling platform. A 57.25 inch (pitch diameter) ring gear is fixed to the rotating portion of the bearing and the rotation drive motor pinion gears drive this gear to produce the rotation motion. Figure 10 illustrates the essentials of the Rotation drive mechanism.

Both axes of motion are referenced to a set of position transducers mounted on the FRM drive motors. Limit switches on both axes restrict the ranges of motion. The two drive systems are described and pictured in more detail in Sections 2.2 and 2.3.

F-R control signal and motor power cables from the F-R Control System located in the antenna pedestal room terminate on control and motor junction boxes mounted on the Apex ring. Short cables from these boxes connect to the motors and position sensors mounted on the FRM drive motors.

The drive motor packages are described in Section 2.1.

The FRM and subreflector are aligned to the antenna primary reflecting surface with a tolerance of a few thousandths of an inch. First, the FRM Rotation axis is adjusted to coincide with the axis of the primary reflecting surface. Shims are then placed under the FRM mounting feet to compensate for any tilt of the apex ring relative to the axis of the primary reflecting surface; this adjustment interacts with the first adjustment, so these two alignments are an iterative procedure. The subreflector is adjusted on the subreflector mounting tube so that its reflection center of rotation coincides with the FRM rotation axis. The subreflector is shaped to "point" the astronomical signals into a feed horn; during manufacture, the pointing axis is marked by an optical alignment target mounted on the edge of the subreflector. The last stage of FRM alignment is to locate the subreflector reflective axis at the feedhorn angular positions; this is done optically using the alignment target. At these angular positions, the rotation position readout is recorded and is used as the band selection command set point for the Antenna Control Computer. These alignment measurements are made by a theodolite. Focus set points are determined in pointing tests by noting the Focus position which produces a maximum receiver output. The alignment operations described above require a local manual control capability from the both the Antenna Apex and Pedestal Room. Manual control switches on the Apex Control Junction box and the S105 module in the Antenna Pedestal Room provide this manual control capability.

The following dimensions provide some idea of the physical sizes of the Focus-Rotation System mechanical components. The FRM cage described above is about 45 inches high by 6 feet in diameter. The subreflector is about 125 inches in diameter and the rotational center is offset from the apparent center by 12.33 inches. The subreflector support tube is 40 inches in diameter and 144 inches long. The FRM mounting flange on the tube is about 53.75 inches from the subreflector bolting interface and the tube projects about 69 inches above the FRM mounting flange.

A pair of 300 Mhz and 600 Mhz crossed dipole feeds are mounted on a support about two feet in front of the subreflector; the reflective face of the subreflector acts as a ground-plane for these feeds. When these feeds are used, the subreflector is raised to the top extreme position, which positions these dipoles at the prime focus. Receivers in the subreflector mounting tube amplify the 300 and 600 Mhz signals which are sent down to the Vertex room via cables routed through the cable tube assembly shown in Figure 1. The routing of these cables is the reason why the subreflector rotation motion is not continuous; the motion must be limited to avoid damaging the cables.

FRM Drive Electronics Layout

Figure 3 (drawing C52502M012, following this section) depicts the layout of the F-R Control System components located in the antenna pedestal room. The control modules are located in two bins in the F-R Control System rack (the Electrospace ACU and NPL encoder are mounted above the F-R bins). The pedestal room Control Junction Box and F-R motor servo amplifiers are mounted on the wall behind the rack. Figures 13 and 14 (in Section 3.0) depict the front and back views of the F-R Control Modules and bin I/O panel. An isolation transformer and motor power contactor are mounted in a box on the floor in the corner of the room. Cables from the Control Junction Box and Servo Amplifiers run up to the Apex Junction Boxes.

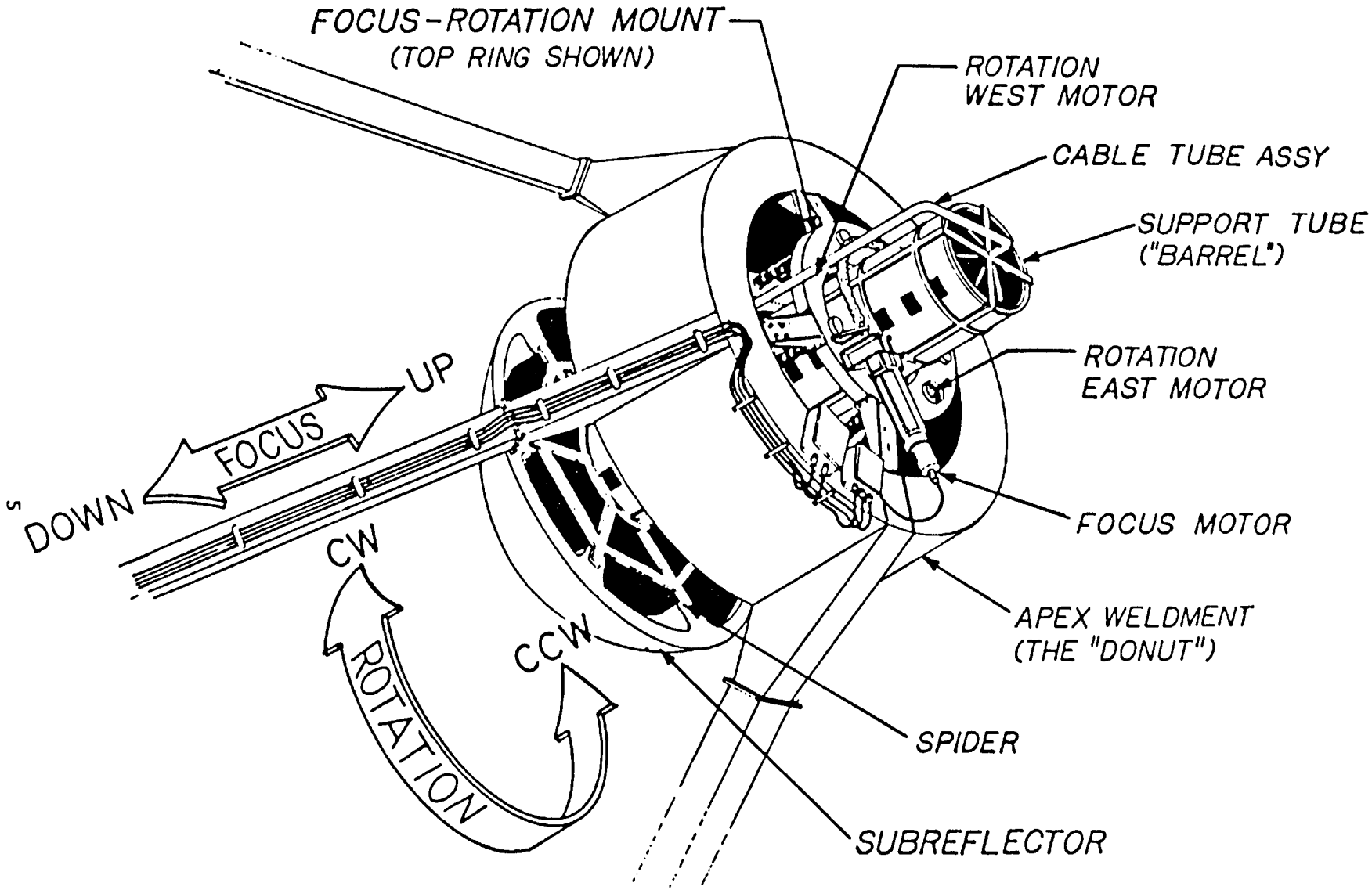


FIG. 1 - ISOMETRIC VIEW OF APEX

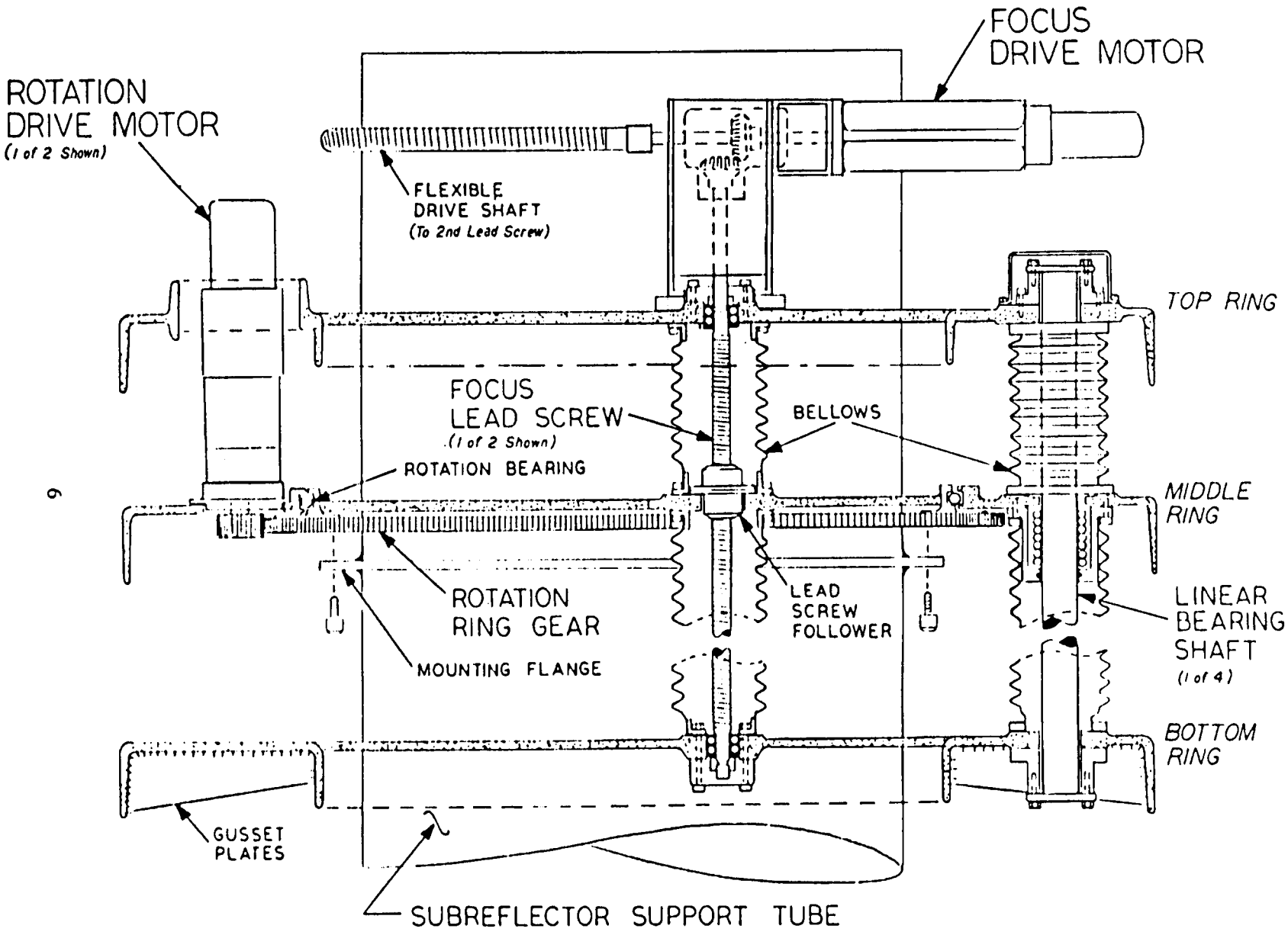
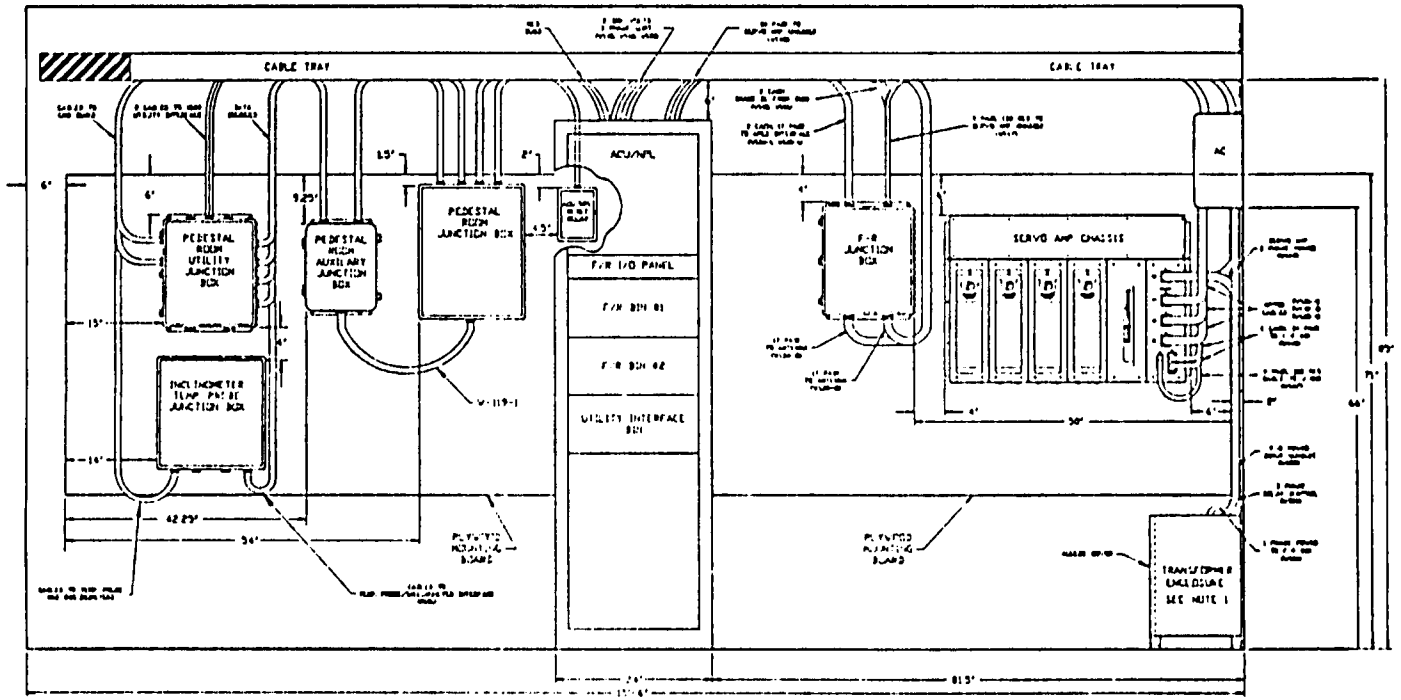


FIG. 2 - SECTION VIEW OF F-R MECHANISM

REV	DATE	DRAWN BY	APPROVD BY	DESCRIPTION
A	5-89	ANSREATTA		ADDED PEDESTAL ROOM AUXILARY JUNCTION BOX



RIGHT HAND WALL LAYOUT

SCALE : 1" = 1"

NOTES :

1. INSTALL ISOLATION TRANSFORMER IN CORNER WITH ACCESS COVER ORIENTATION AS SHOWN.

ACAD : PEDRMLY1

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES	
MATERIAL :	
FINISH :	
NEXT ASSEMBLY	USED ON

VLBA 25 METER ANTENNA	NATIONAL RADIO ASTRONOMY OBSERVATORY SOCORRO, NEW MEXICO 87801
25 METER ANTENNA PEDESTAL ROOM NRAO EQUIPMENT LAYOUT	DRAWN BY ANSREATTA 10-88
	DESIGNED BY WEBER 10-88
	APPROVED BY
	SIZE A

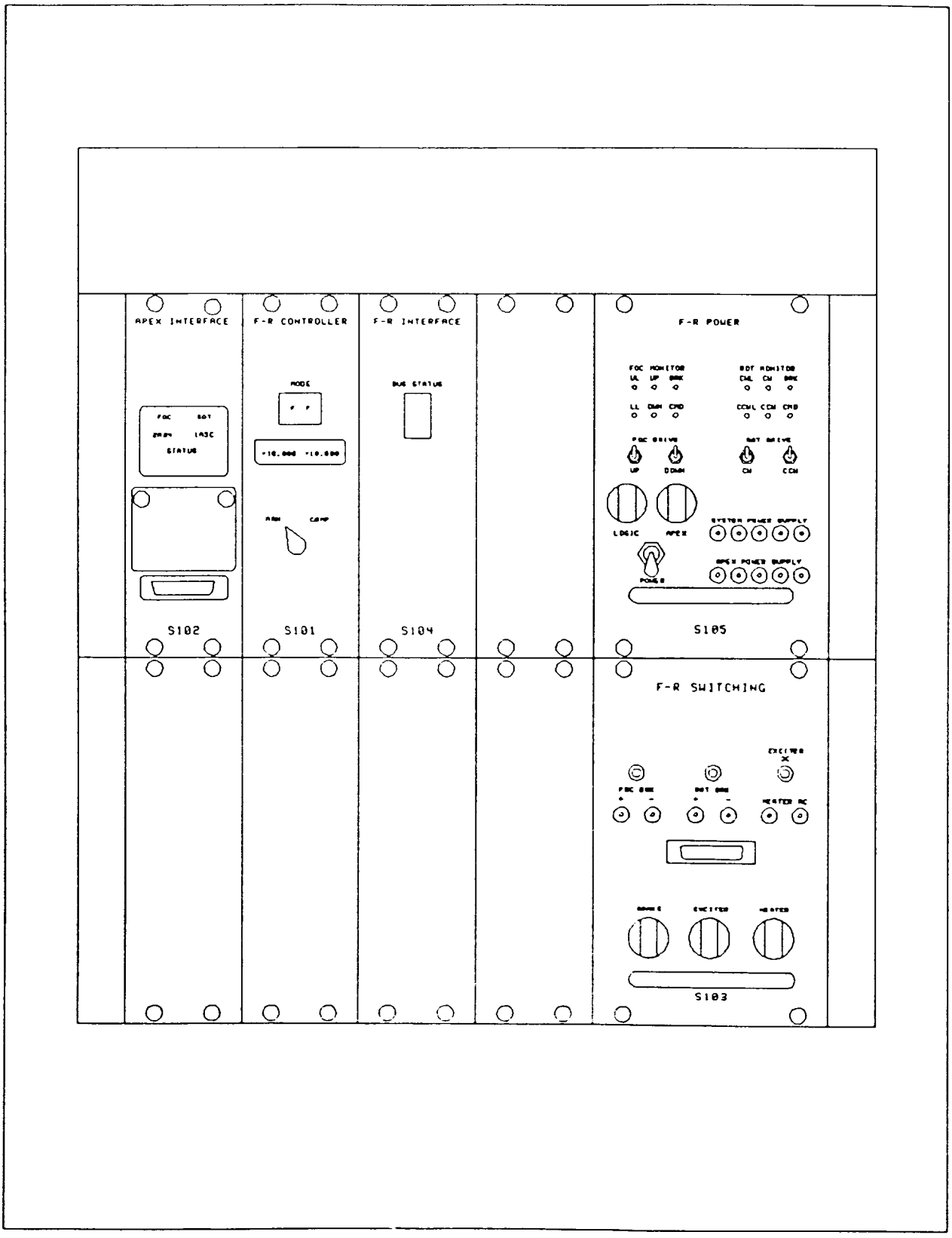


FIGURE 4. F-R CONTROL BIN LAYOUT

1.2 Focus and Rotation System Specifications

The Focus drive range is 28.25 inches; the normal observing focus working range for all the feeds (with the exception of the 300 Mhz and 600 Mhz feeds) is about 3 inches at the lowest region of motion. The Subreflector is raised to the full 28 inch position (i.e. top or prime focus position) when the 300 and 600 Mhz feeds are to be used;

The Rotation drive range is 420 degrees. Although a 360 degree drive range would direct signals into all feed horns, the additional 60 degrees of realizable motion provides a 30 degree margin at each extreme for overlap and limit switch margins.

The operating environment is: minus 20 Degrees to plus 120 degrees Fahrenheit; sea level to 14000 feet altitude; 0 to 50 Meters/sec winds and drenching rain. Heaters on the Focus flexible shaft and bevel gearboxes reduce the mechanism lubricant viscosity at cold temperatures. Accumulating sleet and ice loading effects are also reduced by these heaters.

Focus and Rotation positions are converted to 14 and 16 bit digital values respectively. The position readouts must be approximately linear, smoothly continuous and should repeat within +/- 1 count over the full range of travel. The converters must not have missing codes. In service, a unique pair of Focus and Rotation position values is determined for optimum position settings for each band for a given antenna. The reason for these unique settings is that the physical positions of the feed horns and height of the Apex Ring will vary slightly from one antenna to another. When a motor package is replaced it is necessary to re-establish these band position set-points as part of the alignment.

Focus position is read out as a left-shifted, 14-bit value in a 16-bit word (the two least-significant bits are filled in by zeros). This 14 bit resolution provides a physical resolution of 0.0017 inches per count. The position of the Focus Drive is command-settable to within +/- 1 count of the 14-bit value. The highest VLBA receiving frequency is 86 GHz. The wavelength of an 86 GHz signal is about 3.5mm or 0.138 inches. The subreflector Focus position must be settable and repeatable to a tolerance of 1/16 of this wavelength. This tolerance is 0.0086 inches; the control settability is thus a factor of 5 better than this tolerance.

Focus position is defined as 0.000 inches when the middle ring is positioned at the lowest (i.e. nearest the primary reflector surface) physical hard stop. The associated position readout value is 0000H; this value results from adjustment of the position readout resolver body during alignment. The physical span of Focus motion is 28.25 inches so the upper hard-stop position is 28.25 inches. The stop-to-stop span of the digital position readout resulting from this physical span is about 16216 (14-bit) counts; thus there are about 168 unrealized counts in the digital span. The reason for these values is described in Section 2.2.

In operational service, both extremes of Focus motion are limited by elastomer bumpers to prevent hard metal-to-metal contact of the middle ring with either the top or bottom ring. The bumpers should never contact the Movable Platform because the limit switches are adjusted to inhibit Focus drive before bumper contact. Section 2.2 describes the details of Focus position readout.

Rotation position is read out as a 16 bit value with a resolution of .384 arc-minutes/count and Rotation position is settable to +/- 2 counts. Rotation motion is defined as Clock-Wise when looking at the dish from the sky. Full clockwise (CW) position is 420 degrees and full CCW position is 0 degrees. Section 2.3 describes the details of Rotation position readout.

Focus Position Limit switches restrict the range of Focus motion at the upper and lower extremes of Focus travel. Two sets of limit switches are used; the first (1st) switch is actuated when the Movable Platform approaches a top or bottom ring. The second switch (2nd) is actuated after the 1st and serves as a redundant backup in the event that the 1st switch does not actuate or is not sensed. Settings for the Focus limit switches are tabulated below.

	UL1 (INCHES)	UL1 (COUNTS)	UL2 (inches)	UL2 (counts)
Switches	27.316	F500H	27.428	F600H
Lower Limit	LL1 (inches)	LL1 (counts)	LL2 (inches)	LL2 (counts)
Switches	0.780	0700H	0.669	0600H

The Rotation drive mechanism is inherently capable of continuous motion in either direction but has position limit switches which restrict the range of Rotation motion to protect the 300 Mhz and 600 Mhz feed cables from being twisted apart. 1st and 2nd limit switches are used at each drive extreme position. The settings are tabulated below. Full CCW position is the zero reference, i.e a readout value of 0000H.

	WEST MOTOR GEARBOX		EAST MOTOR GEARBOX	
CW Limit	CW1 (degrees)	CW1 (counts)	CW2 (degrees)	CW2 (counts)
Switches	410.976	F500H	411.797	F600H
CCW Limit	CCW1 (degrees)	CCW1 (counts)	CCW2 (degrees)	CCW2 (counts)
Switches	5.742	0700H	4.922	0600H

Software position limits impose additional constraints on the realizable motion. These are control firmware values which are applied by the controller to new commands to test the validity of command arguments and to prevent drive into the hardware limit switches. The Focus Software limits are: 1.1035 and 26.705 inches, (0A00H and F200H). Rotation Software limits are: 16.4 and 397 degrees, (0A00H and F200H).

The S102 front panel displays the position of the two drives as numeric, fixed-point, 4-digit hexadecimal values driven by the Resolver-to-Digital converters. The Focus position display is a left-shifted 16 bit representation of the 14 bit value in which the two least significant bits are always zero. As a consequence, the least significant digit of the Focus display has only the 0, 4, 8 and C states. The Rotation position resolution is 16 bits; hence the display shows the full 16 bit resolution. (The S102 and S101 modules are described in Sections 3.3 and 3.2 below).

The S102 front panel also displays the state of discrete terms which show the state of drive mechanism limit switches, the activity of data readout to the S101, the A/D Converter activity and analog multiplexer address bits.

The S101 front panel has two numeric 5-digit-plus sign displays which are driven by the control firmware. These displays may be programmed to display fixed-point hexadecimal values and states or signed, 5-digit decimal values with tens and units digits to the left of the decimal point and tenths, hundredths and thousandths digits to the right of the decimal point.

The F-R Control System has features to protect the FRM from mechanical damage in the event that there is mechanical jamming or sticking. The brake voltage and current shall be verified, motor

torques (i.e. motor current) shall be monitored, limit switch and servo amplifier fault discretes shall be tested and realized motion shall be compared with projected motion at 0.5 millisecond intervals during the drive. The Focus 2nd screw position sensor state shall be tested against Focus position to detect excessive lag or a static state. In the event that a fault is detected the motor drive shall be inhibited and the fault condition shall be made available to the Antenna Control Computer via Monitor data readout.

Drive execution time for both axes is about 20 seconds from one position extreme to the other position extreme.

The F-R Control system is capable of concurrent execution of both Focus and Rotation position commands.

The controller executes the basic control loop scan at about a 2 Khz rate.

Monitor Data readout provides a readout of system states and modes, all system analog, Apex and servo amplifier discretes and all fault states.

Position Control commands are discrete set-point commands to drive the designated axis to the commanded position; having reached the commanded position the drive motor power shall be inhibited and brakes engaged to lock the drives in position. Prior to the execution of these commands the control firmware shall test the arguments for validity, test the states of the servo amplifiers, verify that limit switches are not activated, verify that the brake voltage and current are correct, calculate drive ramping parameters, release the brake, verify the brake voltage and current, initiate the drive sequence, and during the drive sequence, verify that the drive is moving as commanded (and for Focus verify that the 2nd screw position is within tolerance). During the drive, the limit switches, servo amplifier fault discretes and motor torques shall be tested at 0.5 millisecond intervals. In the event that a malfunction is detected, the drive shall be inhibited and fault flags set for Monitor Data Readout. When most of the drive motion has been attained, the drive speed shall be ramped down and the drive positioned at the commanded set point within the specified tolerance and the brake shall be engaged. In the event that an over-riding command is received while a drive is active, the S101 F-R Controller shall slow the drive to zero speed and execute the over-riding command. Command arguments shall be echoed back via the monitor data readout to the Antenna Control Computer to verify proper communication with the F-R Control System.

The position of the FRM is manually controllable from both the Apex and Pedestal Room. Manual slew switches on the Apex Control Signal Junction box permit Apex control of FRM position when a special jumper connector plug is installed in J2 in the F-R Bin I/O panel. The FRM position readouts are not available at the Apex. Manual slew control of the FRM position in the Pedestal Room is permitted by slew switches on the S105 front panel. FRM position is displayed on the S102 front panel.

Reset Commands activate the reset logic of the F-R Controller processors.

Nap commands shut down the designated drive until cleared by the Reset Command. Monitor data is available during Nap mode periods.

Manual Override commands permit the Antenna Control Computer to override the F-R Control System in the manual mode.

Monitor Data requests read out monitor data to the Antenna Control Computer. The analog values to be read out are power supply voltages, motor drive currents, FRM and bin temperatures, analog ground and +/- 10 volt reference voltages. Digital values to be read out are drive positions, position errors, velocities and command echoes. Discretes to be read out are limit switch status, servo amplifier status

and faults, brake voltage-current state and the state of the front panel S101 MAN/CMP switch. The control processor shall provide monitor data describing processor command modes, states and values, Apex fault words, Servo Amplifier fault words, 2nd screw (Focus only), motion analysis faults and calculated fault flags.

Position command arguments shall be echo-ed in the monitor data to provide a verification of the proper reception of commands by the F-R Control System.

The Focus drive Second Screw (driven by the flexible shaft) angular position shall not differ from the first screw position by more than 40 degrees. This position tolerance is continuously monitored during Focus motion commands. Excursions greater than this tolerance immediately inhibit the Focus drive motor.

Hardware and software logic inhibit the drives if fault conditions are sensed. Implementation details are described in Sections 2.0, 3.0, 3.1, 3.2, 3.6 and 3.12.

Lightning protection shunt Metal Oxide Varistors are installed on all cable lines in all junction boxes to shunt lightning-induced currents to antenna frame ground. All Apex position and limit switch circuitry is sensed by an Apex Interface module which is optically isolated from the F-R Controller and the rest of the VLBA antenna electronics.

Power to the FRM motors and brakes may be inhibited by an Emergency Stop switch located on the Apex control junction box.

2.0 FOCUS AND ROTATION DRIVE MECHANISM DESCRIPTIONS

Because of the difficulty of access to the FRM on the antenna, the mechanical design incorporates a minimum number of moving parts, very few gears and the ability to maintain and replace many drive components without removal of the mechanism from the antenna. These features significantly improve the reliability and maintainability of the FRM. At the remote VLBA antenna sites a large crane is required for installation and removal of the FRM, Subreflector and Subreflector Support Tube - an expensive and time-consuming operation. Mechanical simplicity and the capability for in-situ repair and maintenance is a major operational consideration. A significant number of major drive components are removed when the Motor Packages are taken off the FRM.

Viscous Friction Loads

An important component of the Focus and Rotation mechanism loads is viscous friction which is the lubricant shearing force exerted between lubricated bearing surfaces. Viscous friction is strongly dependent upon lubricant temperature; over the VLBA temperature environment, the change in viscous friction loads can be greater than ten-fold. The viscous friction of a grease-lubricated mechanism is difficult to model because it is impossible (according to grease manufacturers) to analytically describe grease viscosity as a function of temperature. Grease manufactures assert that the only safe way to determine the viscous friction load is to temperature test a mechanism, measure the viscous load as a function of temperature and then size the drive motor to suit the empirically determined viscous load. A complication of such measurements is that in driving the mechanism, power is dissipated in the grease which raises its temperature and lowers the viscosity; the measurements must take into account this phenomena.

The VLBA FRM viscous friction load calculations were based upon viscous friction models of the Focus and Rotation drives in the VLA F-R mechanism and torque versus temperature measurements of these mechanisms. The models determined the lubricant shearing areas of the gears, bearings, screws etc. Taking into account the mechanism gear ratios, screw pitches and component sizes, the torque versus temperature data was used to estimate the viscous loads of these components, referred to the input drive shafts. The viscous friction loads of the VLBA FRM were modelled using the same rules. The viscous loads were added to the easily-calculated loads due to work and inertial loads to determine the drive motors torque. These loads were doubled to provide a reserve torque margin. The total torque load and margin was used to select the RMS motor torque rating. The IDD 210 and 310 motors have a peak torque capacity about three times the RMS torque rating.

As suggested by the discussion above, temperature is the determining factor in viscous friction. A good, low-temperature grease reduces viscous friction effects (probably by a factor of two) but does not solve the problem. Several years ago a search was made for suitable low temperature grease; this grease is used in the FRM. The phenomena can be alleviated by heating the mechanism in cold weather to lower the lubricant viscosity. This is done on the VLBA FRM. Heaters on the bevel gear boxes and flexible shaft housings add heat to the Focus drive components. Other Focus drive components such as the linear bearings, screws and screw bearings, etc. probably receive some heat due to conduction from the heated areas; it is very difficult to directly heat these components. An industrial temperature controller senses ambient temperature and turns on heater power at about 40 degrees F. This controller, the heaters and wiring are not described in this manual.

The FRM has been temperature tested to temperatures below zero degrees Fahrenheit in a temperature chamber, both with and without the heaters powered. The Rotation drive operated satisfactorily without heat but the Focus drive was found marginally adequate at low temperature, even with heating. Temperature-induced Focus drive sticking occurred; the Focus servo amplifier went into

foldback and about half the time was unable to move the drive. The temperature test was somewhat equivocal because the boots on the drive screws and guide rods became very stiff and eventually cracked; this flexing obviously required a lot of motor torque. Experience at the Los Alamos site during the 89-90 winter, in which temperatures dropped to about zero degrees Fahrenheit for sustained periods, showed temperature-induced Focus dragging and sticking. A Focus motor torque load of about 13 foot-pounds under these conditions was observed but in these events the viscous loading is also somewhat equivocal because low temperature grease was not used in the drive screws.

Grease-lubricated mechanisms with many lubricated bearing surfaces can be very intractable at low temperatures.

2.1 Drive Motor Package Description

High-torque, low-speed brushless DC torque motors are used in the FRM mechanical design; these provide adequate torque at very low shaft speeds to enable direct drive to the moving parts without the use of reduction gearing. The motors are an integrated package consisting of motor, fail-safe brake and position readout transducer. The motor-brake-readout package is an axial assembly with the brake and readout transducer directly coupled to the motor shaft. The brake is adjacent to the motor and the readout transducer is attached to the brake housing. The motor packages are a custom design for NRAO by the manufacturer, Industrial Drives Division of Kollmorgen in Radford, Virginia. The adaptation consisted of a custom-designed end-bell to house the brake and mount the position readout transducer. A double-ended shaft was also required. Figures 5,6,8 and 9 (below) depict the Focus and Rotation motor packages and Torque-Speed curves. These figures are simplified abstractions of the IDD drawings in Volume II.

The motors are DC brushless motors with a permanent magnet rotor and a three-phase stator driven by a chopped three-phase servo amplifier. The stator magnetic field rotates at the shaft speed (i.e. with the rotor magnetic field) in a manner similar to a synchronous motor. The Servo amplifier uses motor shaft position feedback from a resolver internal to the motor and modulates the power drive to the motor in accordance with the motor shaft load. As shaft load increases, the chopped drive power on-time increases and the phase angle of the stator magnetic field increases (to a maximum of 90 degrees) relative to the rotor magnetic field. The motors are capable of operating with a torque overload up to about 3 times the RMS load rating for several seconds; the period is determined by an integrator in the servo amplifier. This feature enables robust acceleration and the ability to overcome static friction in the motor load. The motor is capable of continuous operation at full stall at rated torque without damage; this condition is sustained by the servo amplifier and is a convenient safety feature. The motors have a normally-closed contact safety thermostat enclosed in the motor housing. The contacts are wired to the Pedestal Room Junction Box but at present are not connected to any circuitry. IDD motor data sheets for the BR-3105A (Focus) and BR-2102A (Rotation) motors and the associated Test Limits (an internal IDD test data sheet) are included in the IDD data sheets in Volume II.

The servo amplifiers are also manufactured by IDD and have two different power ratings because the Focus Drive motor needs to deliver much higher torque than the Rotation Drive motor. The servo amplifiers are standard IDD designs with a small servo compensation board which particularizes the servo parameters associated with the inertial load presented to the motors. The servo amplifiers have current sensing circuitry to read out motor current as an analog signal. In addition they contain status and fault sensing logic circuitry which detect fault conditions in the motor and amplifier circuits. These logic circuits inhibit motor drive in the event of a fault. The status and fault conditions may be read out as TTL discretes and the motor current and velocity may be read out as analog signals. The F-R Controller (S101) reads the discretes, fault states and analog signals in the course of controlling the F-R mount; this is discussed in Section 3.1. In addition to the fault protection circuitry mentioned above, the servo amplifiers have an integrator in the current drive circuitry to protect the motors and amplifiers in the event of an overload. The integrator forces the servo amplifier to fold back the motor drive to the motor RMS rating when the energy delivered to the motor is equivalent to the RMS rating for about 15 seconds. This feature protects the motors and servo amplifiers and enables the motors to produce rated torque at full stall for an indefinite period.

The brake is a fail-safe brake; that is, it is engaged when unpowered and requires about 90 volts DC to be released. The actuation and release times are about 100 milliseconds. The brake has zero slack and uses a thin, circular disk attached to the motor shaft; the disk has a permanent magnet attached to the periphery and is attracted to the pole of a fixed electromagnet when the electromagnet is unpowered. When the electromagnet is powered the electromagnet field opposes the permanent magnet field and

releases the brake. One braking surface is attached to the disk and bears against the fixed braking surface on the brake case. Both motor packages use the same brake which is rated for service in the Focus Motor package. This brake rating is higher than necessary for the Rotation drives which have much lower braking torque requirements. The extra braking margin in the Rotation drive insures that the Rotation drive cannot be dragged from position by wind or Subreflector unbalance-induced torques. The brake excitation is 90 Volts DC and the current is 0.39 Amps.

Position readout from the motor shafts has the advantage of being directly coupled to the FRM mechanisms which are directly coupled to the drives; there is absolutely no slack or lost motion in driving the readout transducers.

The Position Readout transducers are commercial units manufactured by the Micron Corporation of New York City. The transducers contain both electrical resolvers and limit switches and are attached to the brake housing. Position readout is accomplished by two resolvers in a dual-speed (i.e. coarse-fine) readout. The fine position resolver is driven from the input shaft by 1:1 anti-backlash gears and the coarse position resolver is driven by anti-backlash reduction gears. The coarse-fine gear ratios are 145:1 and 32:1 for Focus and Rotation respectively. Volume II includes data sheets for the two Position Readout transducers.

The resolvers are high-precision size 11, brushless rotary transformers (Micron part #73-205-144, manufactured by Harowe, part # 11DREX-300-G8/3). The resolver rotors are excited by a 26 Volts RMS, sinusoidal, 400 Hz signal from the S103 module. The excitation signal is sometimes referred to as the excitation carrier signal; the stator signals are also carriers which are amplitude and phase modulated by the resolver shaft angle. The resolver stator uses two orthogonal windings (commonly referred to as the Sine and Cosine) which have a maximum (carrier) output of 11.8 volts RMS (i.e. when a stator is at an angle which produces a maximum output). The transformation ratio from rotor to stator is .445. The Stator impedance is about 500 ohms at 400 Hz. The stator windings produce an output whose carrier signal amplitude varies as the Sine and Cosine of the resolver shaft angle. At the point of transition through a minimum, the phase angle of the stator carrier signals shift by 180 degrees (relative to the excitation carrier phase). By comparison of the amplitudes of these two stator carrier signals and their phase relative to the excitation carrier phase it is possible to convert these analog signals to high precision digital values which are an analog of the input (i.e. motor) shaft position. This process is described in Section 3.2.

400 Hz was chosen as the Position Readout resolver carrier frequency over the alternate frequency of 2600 Hz because the IDD motor resolver operates with a 2600 Hz carrier; using a 400 Hz carrier for the position readout resolvers eliminates the possibility that cross-talk between the two resolver signals (which might occur in the junction boxes or wiring) could produce coherent interference. The resolver shaft positions must be aligned to the mechanism positions and to each other prior to installation of the F-R mechanism on the antenna. This alignment is discussed in Section 4.0.

The maximum angular error of the resolvers is +/- 3 minutes. The relative error is 3min/3600min (1/1200) or less than 0.1 % and the stator carrier output is a smooth, continuous sinusoidal function with a maximum error of 3 minutes over the 360 degrees of shaft angle input. The F-R Controller uses the digital values converted from the resolver stators to position the mechanisms.

The limit switches are driven by the transducer input shaft and use a differential gearing scheme to drive two cams which actuate a switch. The cams are notched and each cam rotates relative to the other (and to the frame) because of the differential drive; when the two cam notches coincide under a limit switch actuator the switch makes a transition to the opposite state. The switch actuation is directional; that is, when the cam notches approach the switching position, the contact closure sequence (i.e. transition from closed NO-C to NC-C and the complementary case) depends upon the direction of shaft

rotation (i.e. CW or CCW). The switch actuator is an alternate action mechanism which drives a micro-switch with NO-C-NC contacts. Each switch actuation is independent of the other switch actuations. The two switch cams may be adjusted by loosening spline set screws and slipping the cams so as to actuate the switch when the input shaft is set at a known angular position. The transducers contain more than one limit switch; the focus transducer contains four switches (for 1st and 2nd limits) and the Rotation transducer contains two switches. One transducer is used for 1st and 2nd CW limits and the second is used for 1st and 2nd CCW limits (there are two Rotation drive motors; this is discussed in Section 2.3). The readout transducers input shaft can revolve up to 500 revolutions before the switch actuation is repeated. When aligning the switches it is important to recognize that there are two regions of limit switch actuation. The first is the permitted region of input shaft position delimited by proper actuation of the 1st and 2nd limit switches. The forbidden region must be avoided because it will produce apparently erroneous actuation of the switches because of the directional properties mentioned above; in this region the sequence of 1st and 2nd limit switches will be reversed and the resolver readouts at these actuation points will be incorrect.

2.2 Focus Drive Mechanism

Figure 5 (below) depicts the Focus Motor Package. The drive motor is an IDD 310 series with an RMS rated torque of 10 foot-lb and a peak torque rating of 36 foot-lb. Figure 6 (next page) depicts the Focus motor torque-speed curves; these are a simplified version of the IDD torque-speed curves found in the data sheet section of Volume II.

The Focus motor torque-speed characteristics are very important. This motor must lift a large dead weight (nearly 1000 pounds) and drive a heavy friction load consisting of both coulomb and viscous friction. Because the motor shaft never rotates more than 143.5 revolutions and the acceleration is very gentle, it never reaches high rotational speeds and the inertial load is very small. Clearly, the Focus drive motor should have high-torque, low-speed characteristics.

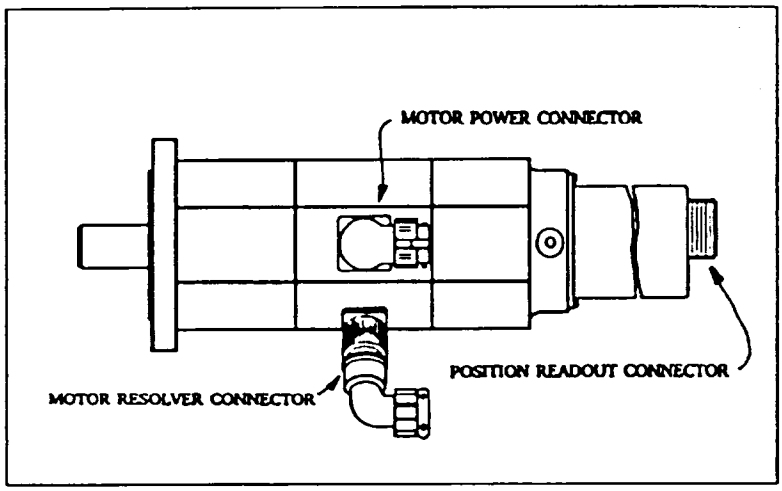


Figure 5, Focus Motor Package

The torque-speed curves shown in Figure 6 show three torque regions delimited by two curves. The left-most region is the continuous duty region in which the motor can operate continuously at any combination of speed and torque within the continuous-duty curve at any temperature up to the maximum rating.

The region to the right of the continuous duty region, bounded by the peak torque-speed curve, is the intermittent duty region. In this region, the motor can operate for short periods at any combination of torque and speed which does not exceed the bound of the peak torque-speed curve. The excursions into this region can be either a momentary intrusion from the continuous duty region or an immediate intrusion from the unpowered state.

The third region is to the right of the peak torque-speed curve. Motor loads which exceed this curve will cause the motor to stall (i.e. motor speed is zero).

In the event of motor overloads, foldback circuitry in the BDS3 servo amplifier protects the motor and amplifier by literally "folding back" the developed torque to the continuous duty curve. Foldback conditions are: over a short time period, if the energy input to the motor reaches the equivalent of sustained operation at the maximum rating in the continuous duty region, developed torque is reduced to the continuous duty region. Foldback operation is bounded by the continuous duty torque-speed curve. In the foldback mode, the motor can continue to operate indefinitely in the continuous duty region. When the motor is started with an immediate motor overload located on the peak torque-speed curve, the integrator will time out in about 15 seconds. If the motor has been operating for a long time with a steady-state load on the continuous duty curve, the foldback circuit will not permit excursions into the intermittent region. If the motor is started with an immediate load which exceeds the peak torque-speed curve, the motor will stall and the intregator will fold back the developed torque to the continuous duty curve at zero speed. The motor and amplifiers are not damaged by sustained operation at stall conditions.

The Focus motor maximum speed speed is 507 RPM (8.456 RPS) and the drive up (antenna at zenith) lifting torque load is about 2.6 lb-ft. It has been observed that at temperatures around zero Fahrenheit, the Focus foldback circuitry has become active to restrict torque to the continuous duty region. In this condition the motor continued to drive the Focus mechanism. Note the Focus operating torque-speed region in the curves of Figure 6. The continuous duty curve is almost vertical from zero to 1400 RPM. In the intermittent duty region at 500 RPM, the motor torque is reduced by only about 10 percent relative to the zero-speed torque.

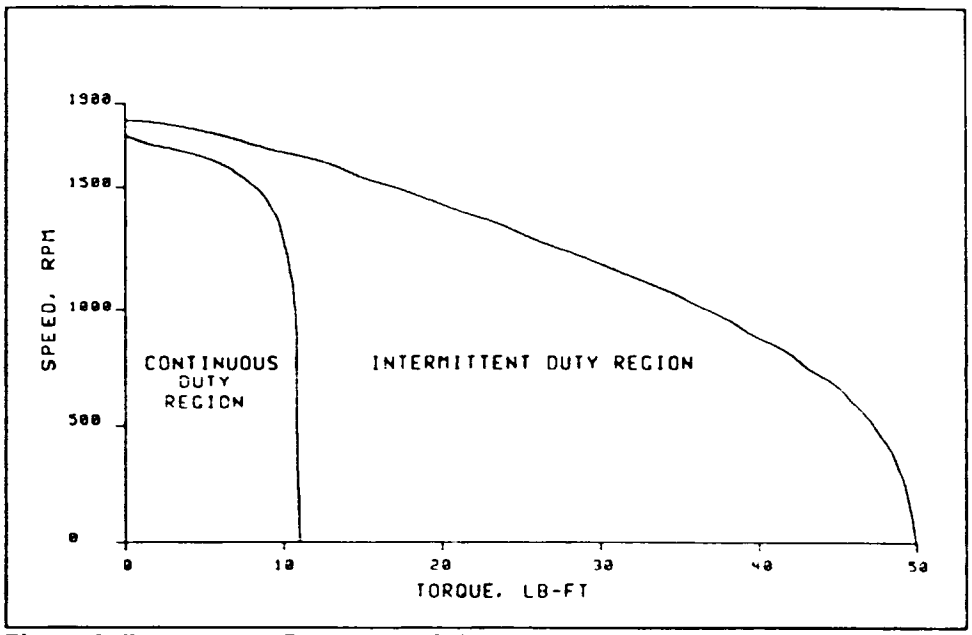


Figure 6, Focus Motor Torque-Speed Curves

The Focus position readout transducer is Micron part #50-306-520-1457 with four limit switches and a 145:1 dual speed position readout resolver.

Figure 7 (next page) is a simplified drawing of the Focus drive mechanism. Two synchronized, diametrically opposed drive screws position the traveling platform; the nuts are on the traveling platform. The screws are synchronized by a 6 foot flexible shaft between two 1:1 bevel gear boxes mounted on the top ring above the screws. The gear boxes have three shafts which may be used for input or output; two shafts are in line and the third is at right angles to the in-line shafts. The motor package is mounted horizontally on the South drive bevel gear box (above the South drive screw) and is coupled to one of the in-line shafts; the other in-line shaft is coupled to the flexible shaft. The flexible shaft runs to the North (second) gear box where it is attached to an in-line shaft. The second gear box right angle shaft is coupled to the second screw. The flexible shaft is supported in a semi-circular run above the top ring so as to provide clearance for the Subreflector Support Tube and counterweights. If we were to consider the gear boxes and flexible shaft to be a set of coupled straight rigid shafts, it is evident that the lead screws will rotate identically when the input shaft is rotated by the drive motor. In practice, flexible shafts are not as stiff (i.e. strong) as rigid shafts of the same size; hence they "spring" slightly under load which causes the flexible shaft-driven second screw angular position to slightly lag the motor-driven screw. The two screws each carry about the same weight load; the combined weight of the Subreflector and Subreflector Support Tube is about 600 pounds and the Movable Platform weighs about 400 pounds so the total screw lifting load (antenna vertical) is about 1000 pounds; hence each screw supports about 500 pounds (worst case) lifting load and this is the lifting load imposed upon the flexible shaft. Viscous (grease and lubricant friction) and coulomb (sliding) friction of the flexible shaft, drive screw and the linear bearings add to the torque load imposed upon the flexible shaft. The flexible shaft has been sized with a sufficient margin so that the shaft's positions should be synchronized within 40 degrees. If the flexible shaft were to break (it's the weakest link in the Focus Drive), the FRM could be overstressed with the possibility of damage to the expensive mechanism. To verify that the flexible shaft-driven screw is

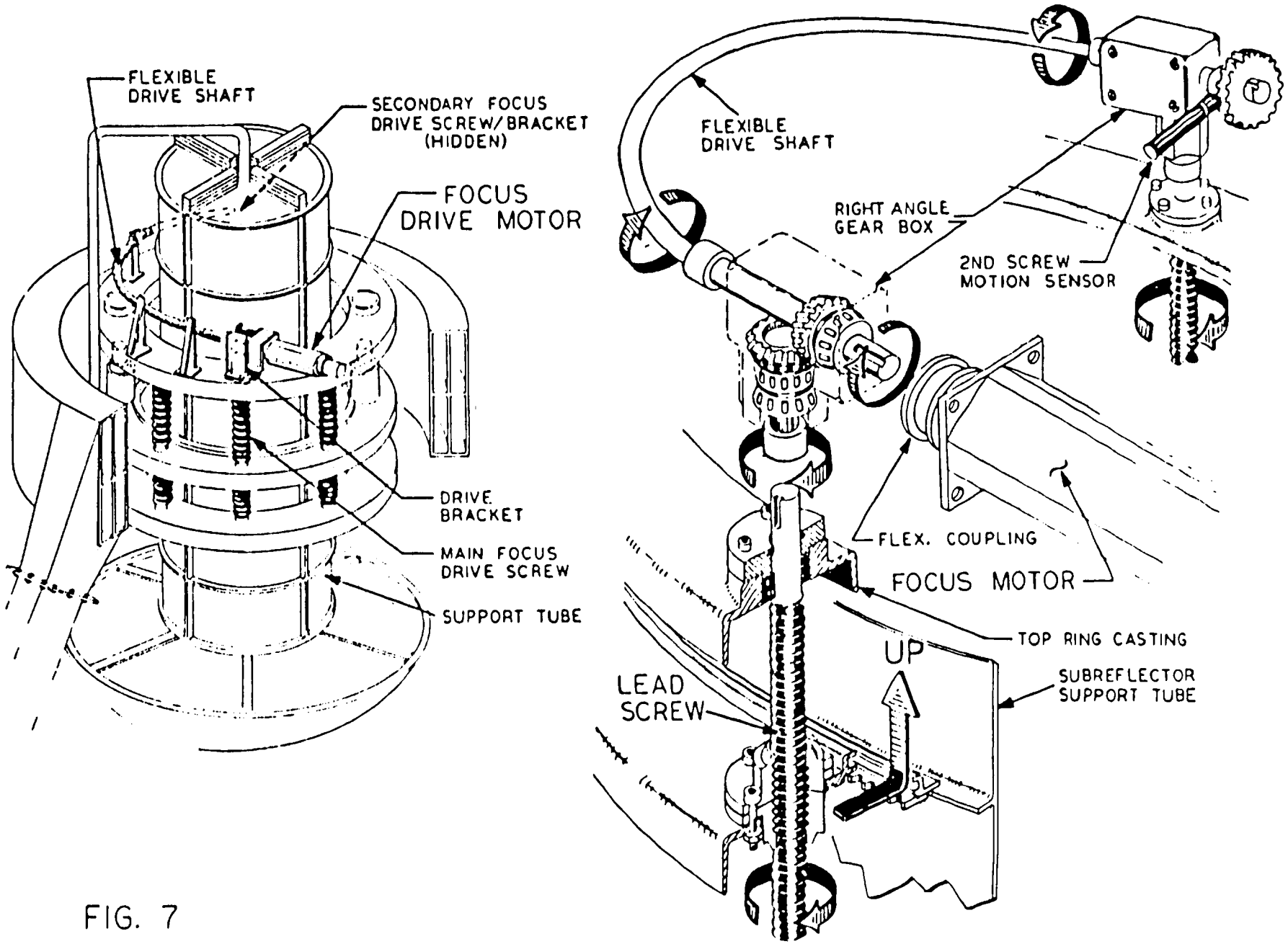


FIG. 7
SIMPLIFIED FOCUS MECHANISM

properly synchronized, a position sensor is mounted on the second in-line shaft of the second gear box. This sensor output is continuously compared with the position of the motor-driven shaft: if there is excessive lag or the sensor output is static, the F-R Controller will immediately inhibit the drive to the Focus drive motor. This tracking is discussed in Sections 3.7 and 3.8.

The Focus Drive lead screws are precision-ground 25 mm diameter, 5mm pitch (thread advance per turn) screws capable of supporting a large load in tension. The nuts are roller nuts mounted on the Movable Platform. The Movable Platform moves along the four 2.5 inch guide rods and is supported by 8 linear ball bushings. The roller drive lead screws and linear bushings moving on four precision-ground guide rods make for a low coulomb friction drive system.

The Focus Drive is locked in position by the brake in the motor package.

The stop-to-stop span of the digital position readout is 16216 counts; this span is determined by the mechanical parameters: lead screw pitch (5mm/rev), the stop-to-stop distance traversed (28.25 inches) and the Focus position readout coarse Resolver reduction gear ratio (145:1). The stop-to-stop motion results from 143.51 revolutions of the Focus leadscrew which drives the coarse resolver 145:1 reduction gears. The 143.51 screw revolutions thus results in a resolver shaft rotation of 143.51/145 (0.989724) or 356.3 degrees. In terms of the digital position readout span, 356.30 degrees is 16216 counts (3F58H). This slightly reduced readout range insures that the readout values never cycle through zero (assuming that the resolver is properly aligned). The Focus fine resolver is not used because the resolution available from the coarse resolver and converter is adequate for control and position readout. The scaling of the Focus coarse resolver is 16384 counts/21600 arc-seconds = 0.785 counts/arc-second. One count on the circumference of the 1 inch resolver is 1.0*pi/21600 = 0.00015 inches/count so the alignment adjustment is very sensitive to resolver body positioning. One revolution of the screw corresponds to 16384 counts/145 = 112.993 counts/screw revolution. One of the 20 states of the 2nd screw sensor corresponds to 112.993 or 5.649 counts/gear tooth change.

The Subreflector Support Tube projects about six feet above the mounting flange; this projection serves as a mount for counter-weights attached to the top of the tube. The tube projection and the counter-weights serve to counter-balance the Subreflector and the associated lower portion of the Subreflector Support Tube around the linear bushings on the guide shafts.

Although the Focus Drive moves about 1000 pounds, the inertial load imposed upon it is greatly reduced by the Focus Drive screws; the linear motion inertia of the Subreflector, Subreflector Support Tube and counterweights are transformed by the factor $(p/2pi)^2$ (p is the screw pitch) to a modest rotation moment of inertia roughly equivalent to the motor rotor.

An important load on the Focus drive is temperature-dependent viscous friction. This phenomena was described above. This lubricant shearing force load is developed in the motor bearings, bevel gear boxes, screw thrust bearings, lead screw nuts, guide shaft linear bearings and the flexible drive shaft. At 70 degrees Fahrenheit, the load is small; as temperatures drop below freezing the viscous friction becomes more dominant and at minus 20 degrees F, the viscous friction load (in conjunction with the other loads described above) can (without heating) exceed the torque capability of the Focus motor. Experience has shown that at low temperatures, the Focus drive, even with heating is marginal.

The lubricated components of the Focus drive, as originally designed, consisted of one lead screw and four sets of linear bearings. Viscous friction load calculations to select the drive motor were based upon this configuration. The second screw, bevel gearboxes and flexible shaft were added to improve the physical stability of the Focus drive mechanism. These added components significantly increased the viscous shearing area (probably by a factor of about two); as a result, the friction load imposed upon the Focus motor was greatly increased. Cold-temperature Focus torque loads experienced at the Los Alamos

VLBA antenna during the 89-90 winter were observed to be about 13 foot-pounds at temperatures around zero degrees Fahrenheit. This load exceeds the RMS rating of the Focus drive motor.

The 2nd screw position sensor is an inductive sensor which senses the proximity of the teeth of a 10-tooth steel spur gear. A Hall effect sensor driven by a 10-pole cylindrical magnet was previously used but availability problems have forced a change to the inductive sensor. The inductive (and Hall effect) sensor output a bi-stable TTL level signal as a function of the tooth position relative to the sensor. Thus the sensor output is ten pulses per shaft rotation. The sensor-tooth gap is adjusted to produce approximately a 40/60 duty cycle (ideally a 50% duty cycle is best), so there are 20 rising and falling edges per shaft revolution; each transition corresponds to roughly 18 degrees of shaft motion.

2.3 Rotation Drive Mechanism

Figure 8 (below) depicts the Rotation Motor package. The drive motor is an IDD 210 series with an RMS rated torque of 4 foot-lb and a peak torque rating of 10 foot-lb. Figure 9 (next page) depicts the Rotation motor torque-speed curves; these are a simplified version of the IDD torque-speed curves found in the data sheet section of Volume II.

The Rotation motor load torque-speed requirements are similar to the Focus conditions described above except that the Rotation drive does not have the heavy lifting load of the Focus drive. The modest lifting torque load due to subreflector unbalance is probably less than one lb-ft. Secondly, two motors equally share the load (with a small torque bias). The Rotation motor loads are the small subreflector unbalance load, acceleration load (very small) and the viscous and coulomb friction loads.

Unlike the Focus drive, the Rotation drive does not have heaters to alleviate low temperature-induced viscous friction. The Rotation drive has not been observed to foldback or stall in temperatures near zero Fahrenheit.

The maximum Rotation motor motion is 32 revolutions and the maximum commanded velocity is 2190 RPM (36.5 RPS) and the warm temperature torque load is less than 1 lb-ft. Note the Rotation operating torque-speed region in the curves of Figure 9. The continuous duty curve is almost vertical from zero to 2000 RPM. In the intermittent duty region at 2000 RPM, the motor torque is reduced by only about 1 percent relative to the zero-speed torque.

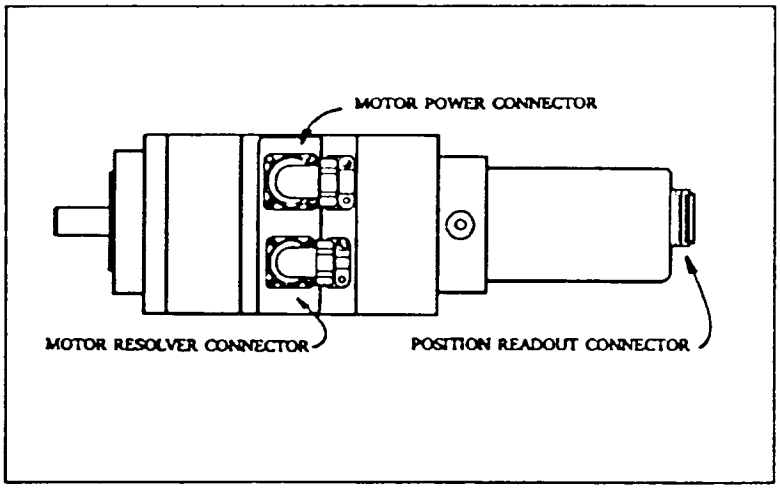


Figure 8, Rotation Motor Package

Like the Focus drive, the Rotation motors and servo amplifiers are protected by foldback circuitry set up for the torque-speed characteristics of the Rotation drive motors and amplifiers. The Rotation foldback characteristics are identical to those described in Section 2.2.

The Rotation readout transducer is Micron part #50-306-532-1456 with 1:32, dual speed position readout resolvers and two limit switches. Figure 11 depicts the 210 motor torque-speed curve.

Figure 10 (two page pages ahead) is a simplified drawing of the Rotation Drive. The traveling platform contains all the Rotation Drive components. The Subreflector, Subreflector Support Tube and counterweights are supported by a 46.5 inch diameter wire-race bearing which has the outer race attached to the inner diameter of the Movable Platform. The inner race is attached to the drum support flange and a large steel 621 tooth, ring gear. The gears are 12 dimetral pitch spur gears, with a 0.75 inch face (width of teeth) which is appropriate for a heavy duty drive train. Two drive motors are diametrically mounted on the Movable Platform. 23 tooth, 12 dimetral pitch, nylon pinion gears mounted on the motor shafts drive the ring gear to produce the rotation motion. The inexpensive, easily-replaced Nylon gears are used to reduce wear on the expensive and difficult to replace ring gear.

One very important consequence of the use of nylon gears for the motor pinions is the gear wear-induced shift in the Rotation position readout. When the Rotation drive is moving, the nylon gear teeth will wear in meshing with the steel ring gear teeth (the presence of dust and grit will aggravate the effect); this will shift the position readout which is driven by the motor shaft. A tooth wear of 0.010 inch on the tooth bearing surface (the thickness of three sheets of paper) will cause a position readout shift of (0.010/1.9166*pi)*2048 counts, or 3.4 counts which is equivalent to 1.3 minutes of rotation position. This is a small but non-trivial error which can accumulate unobtrusively. Since the pitch diameters of the gears are fixed, the scaling of the readout will remain constant as the gears wear - the effect is a position zero-shift. How much is this 0.010 tooth wear? The width of a tooth (at the pitch diameter) is 0.131 inches so this 0.010 wear is only about 10 percent on either side of the gear which may not be visually noticeable. For this reason the VLBA antennas should have periodic pointing error observations to calibrate this effect. It is important that the VLBA Engineers and Technicians be aware of this effect in the event that the antenna pointing changes with time. The potential for wear-induced shifts in motor shaft position is the reason why the coarse-fine position readout resolvers converters are both wired to the West motor package (only) rather than have one converter connected to a coarse resolver in one motor package and the fine in the other motor package. A consequence of this configuration is that wear would shift the resolver zero alignment.

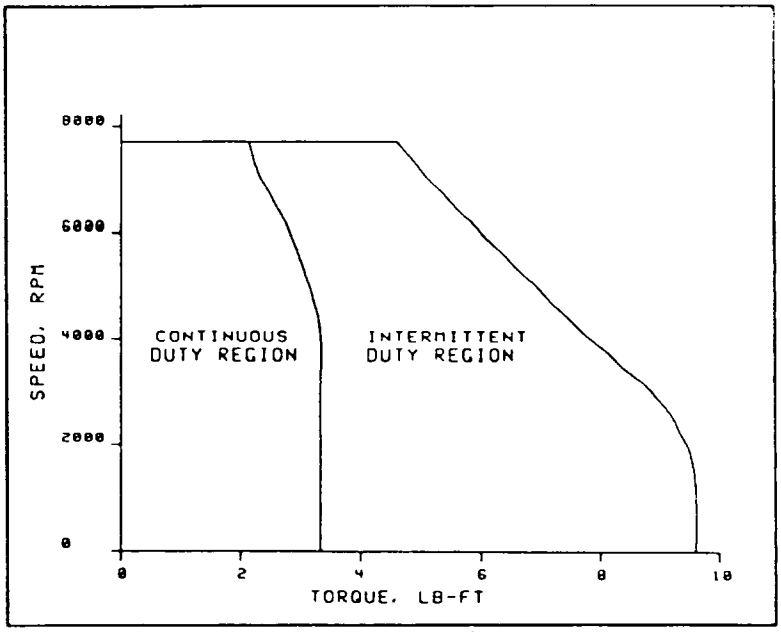


Figure 9, Rotation Motor Torque-Speed Curves

Two Rotation drive motors are used to eliminate the drive motor pinion to ring gear back-lash which would result from the use of a single drive motor. If a single gear was used to drive the ring gear the resultant back-lash (i.e. gear slack) would cause excessive rotational position error which would degrade antenna pointing. The motors act in concert during drive but one motor has a slight torque bias relative to the other; the result is that, at all times, the torque bias keeps both pinion gears continuously engaged against the ring gear teeth. When the brakes are engaged there is absolutely no slack in the ring gear position; the Rotation Drive position is rock-solid stable.

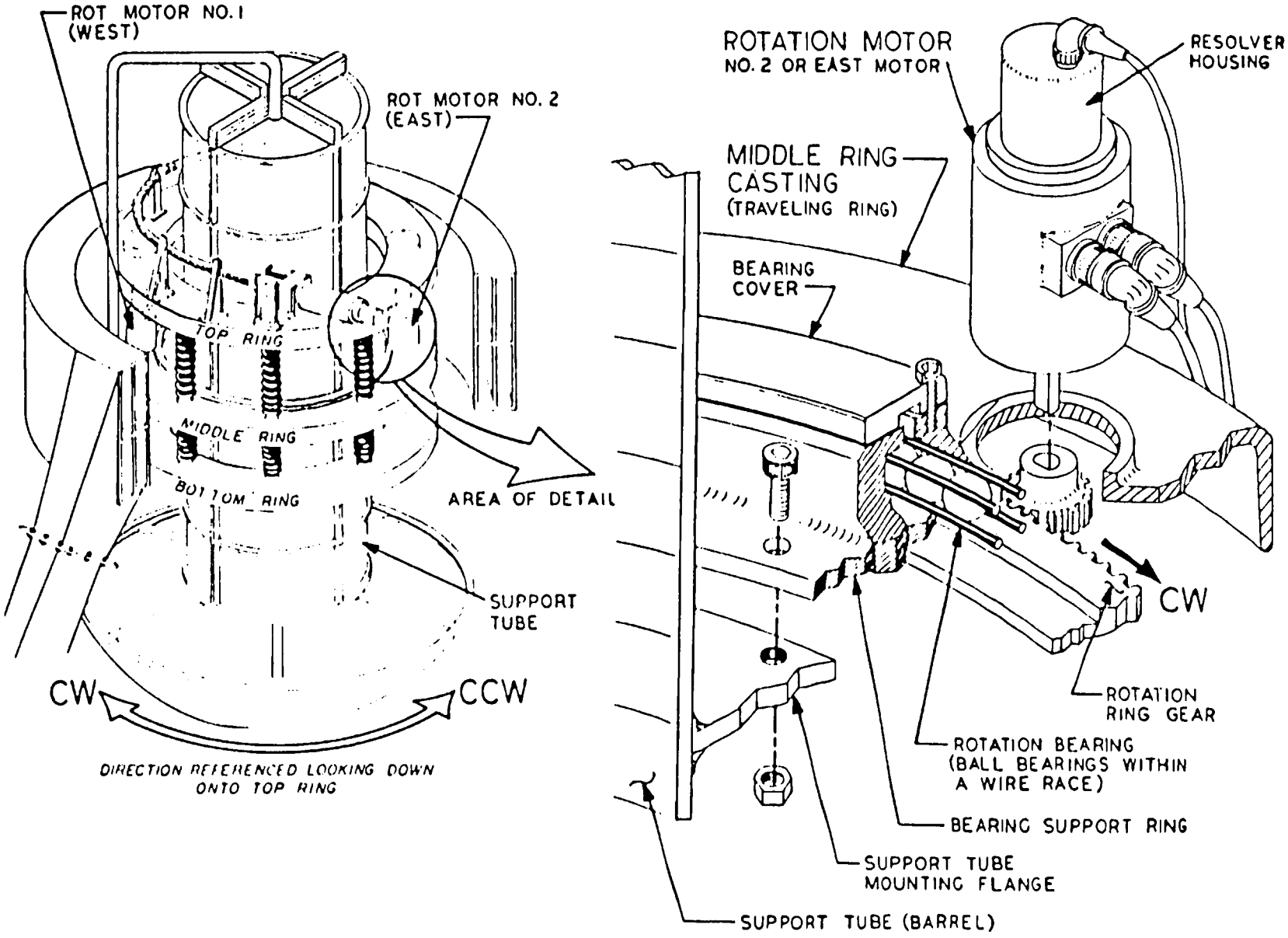


FIG. 10- SIMPLIFIED ROTATION MECHANISM

The Rotation drive (ignoring the limit switch constraints) is mechanically capable of continuous rotation in either direction; the reason for limiting the rotation is that cables from the 300 Mhz and 600 Mhz feeds would be damaged if the motion were not limited.

In contrast to the Focus Drive, the Rotation Drive has a small lifting torque load consisting of the Subreflector unbalance (remember that it is not a figure of revolution and the rotational center is off-set from its apparent center); this unbalanced load is on the order of a small fraction of a foot-pound at the drive motors. The Rotation drive is also subject to wind loading effects because the Subreflector acts like a large sail. Like the Focus Drive the Rotation Drive has viscous and coulomb friction loads resulting from bearing and motor pinion-ring gear torque loading.

Unlike the Focus Drive, the Rotation Drive has a huge inertial torque load resulting from the large Subreflector Support Tube and large Subreflector; this inertial load is transformed by the motor pinion gear-ring gear ratio of $(1/27)^2$ This load is about 250 times the sum of the two drive motor rotor moments of inertia. Consequently, the Rotation Drive servo amplifiers have a large compensation for this large load inertia. If the motor were to directly drive the drum, the load inertia - motor inertia ratio would be on the order of 180,000.

The Motor pinion-ring gear ratio is 27:1; that is 27 revolutions of the motor pinions will rotate the ring gear (and hence the Subreflector) 360 degrees; an additional 5 rotations of the motor pinion gears causes the ring gear to rotate 420 degrees. This motor shaft rotation of 32 revolutions is the basis for the Rotation Readout Transducer coarse-fine ratio of 32:1. With 420 degrees of rotation of the Subreflector the coarse resolver rotates 360 degrees and the fine resolver rotates 32 turns. If a single 1:1 +/- 3 minutes coarse resolver was the only position readout, the realizable accuracy would be unacceptable; the use of a dual speed resolver pair provides a high precision readout in which the coarse resolver provides a coarse reference which is digitally combined with the high resolution fine resolver output. The two resolver signals are converted to digital values by resolver to digital converters and the two signals are combined to produce a composite, 16-bit, high precision measure of Rotation position. The readout resolution is 420*60/65536 = 0.397 arc-minute/count. The implementation of this combination is discussed in Section 3.2.

The scaling of the Rotation fine resolver is 65536 counts/32 = 2048 counts/revolution in terms of the composite 16-bit readout. This corresponds to a readout resolution of 2048 counts/360 degrees = 5.69 counts/degree. Strictly speaking, the coarse and fine R/D converters are 14-bit units but the fine resolver resolution is the composite 16-bit resolution since the fine R/D converter output is scaled to correspond to the lower 14 bits of the composite value. It should be noted, however the fine resolver readout will not necessarily correspond to the lower 14 bits of the composite readout. The scaling of the Rotation Coarse resolver is 16384 counts/360 degrees or 45.5 counts per degree.

The Rotation drive viscous loads are developed in the motor bearings, gear-mesh motion and ring gear bearing. Temperature tests and cold weather experience have shown that to date, viscous friction loads are well within the capacity of the Rotation drive motors.

3.0 FOCUS AND ROTATION CONTROL SYSTEM

The F-R Control System releases the brakes and drives the servo amplifiers to cause the two drives to reach the positions commanded by the antenna control computer. In performing this action, the control system reads the current drive position, calculates the driving parameters for the servo amplifiers, tests the fault states of the amplifiers and limit switches, releases the brakes, initiates the drive sequence to ramp the drives up to speed, ramps the speed down to a convergence speed and, finally, nulls the position error and engages the brakes. During the course of drive, at 0.5 ms intervals the controller samples the current drive positions, calculates a predicted drive position for the next sample time, compares the position with the previously predicted position, checks for servo and limit faults and causes a fault shutdown in the event that the predicted position is not consistent with the predicted position or faults have occurred since the previous samples. In the case of the Focus Drive the 2nd screw sensor is tested for motion consistent with the change in the Focus drive position.

It is important to recognize that although the microprocessor controller actuates the control signals and inputs system states, it is only the agent of the firmware control program which is the control logic of the system.

Figure 12 (drawing D55007K001, two pages ahead) depicts the F-R Control System Block Diagram.

Figure 13, (drawing D55007W001, following Figure 12) depicts the F-R System cable structure. This drawing shows the cable W-numbers, wire count and routing; note the differences with Figure 12.

Figures 14 and 15 (following 12 and 13 above) depict the Focus and Rotation control bin front and rear (i.e. connector) Bin Layouts.

The digital portions of the control system are S101, S102 and S104. S104 is the F-R Interface which interfaces the Antenna Control Computer to the F-R Control System, the IDD servo states and the S101 F-R Controller.

S101 is a dual-axis (each controller is independent) microprocessor-based controller for the system; firmware control programs regulate all activities in the Focus-Rotation System. S102 is the Apex Interface which interfaces all Apex signals to the F-R Controller.

S102 has an additional function. Because the Antenna Apex is an attractive target for lightning and S102 is connected to Apex position and limit switch sensors, it is electrically isolated from the balance of the control system by optical isolators. In the event of a lightning strike on the Apex, lightning-induced currents in the cables are limited to the S102 module (there are surge arrestors which mitigate lightning effects, see Section 3.12 for details). In effect, S102 is the sacrificial module. This protects the MCB which connects to the Antenna Control Computer and all VLBA electronics. Lightning protection is discussed in Section 3.12.

S104 receives commands and data requests from the Antenna Control Computer via the Monitor and Control Bus (MCB) and routes the commands and data to the S101. S104 also buses sets of IDD servo Amplifier fault and status discretes and servo amplifier analog data to the F-R Controller S101 under control of the programs executing in S101.

S105 is the F-R Control System power supply and the local manual control panel.

S103 is the F-R Switching Module which generates the 400 Hz resolver excitation and the brake DC power; it contains solid state relays to switch the brake power supplies and the 208 Volts servo

amplifier power contactor. The S103 is connected to the Apex via the resolver excitation. Solid state relays in the S103 power the brake power supplies; these relays provide isolation between the S101 control logic and the Apex.

The Servo Amplifier Chassis contains five IDD modules which drive the three FRM motors at the Apex. A 300 volt DC power supply provides motor drive power to the three servo amplifiers. One Focus and two Rotation servo amplifiers drive the three motors. A backlash controller equalizes the drives to the two Rotation Servo amplifiers and provides a small settable bias to the drives so that the two Rotation motors take up the mechanical lash of the two motor pinion gears on the ring gear. The IDD Servo Amplifiers are also subject to lightning-induced currents but the high power motor drive transistors are not easily damaged. The IDD resolver circuits are also somewhat less vulnerable than digital logic.

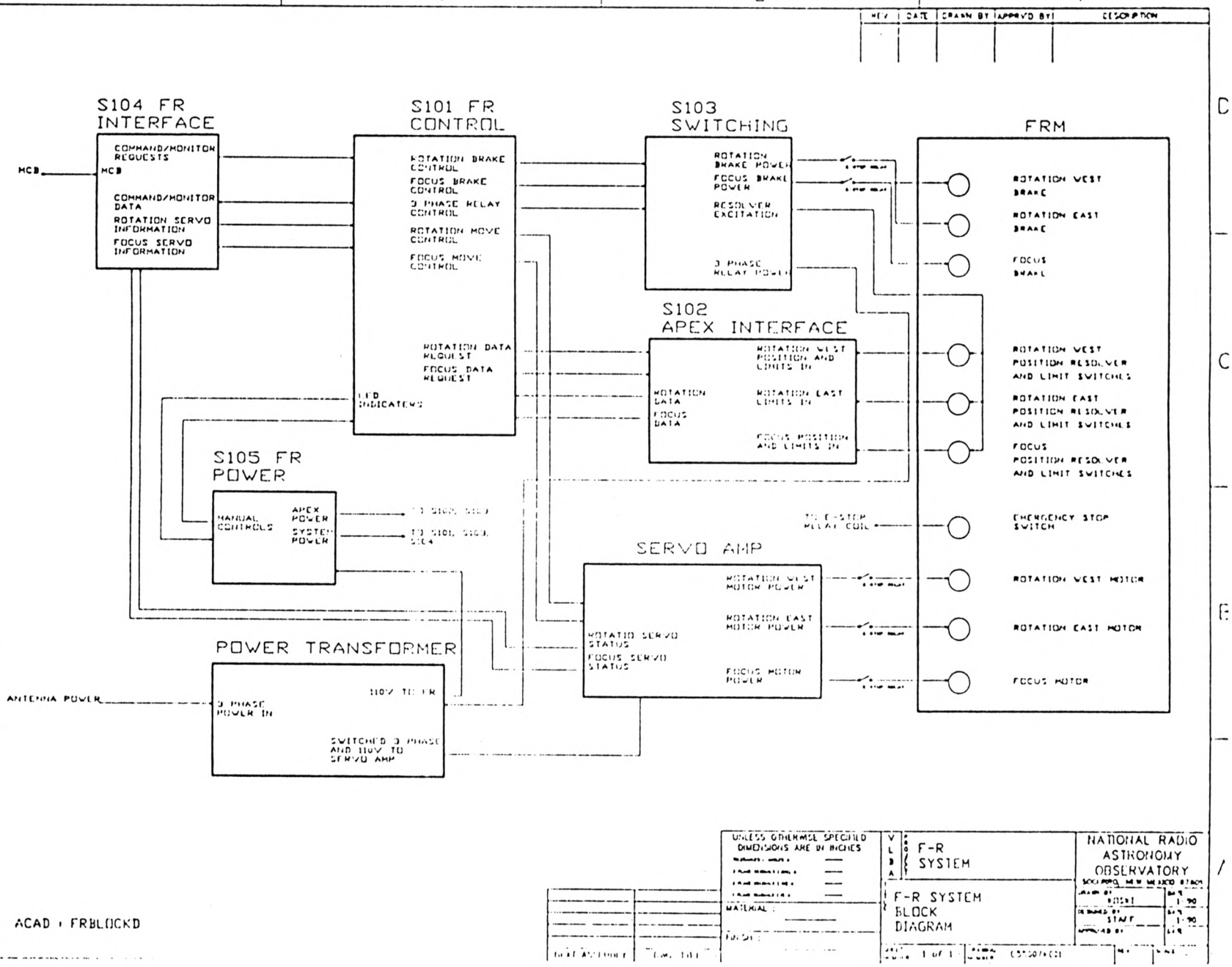
The Isolation Transformer box has a three phase delta primary and a wye secondary isolation transformer for the three phase 208 volt AC servo power. This transformer isolates the servo amplifiers from common-mode power line perturbations and provides some protection to the F-R Control system because of this isolation from AC line surge effects. The box also includes a power contactor for the 3-phase 208 volts AC. The servo amplifiers logic and analog power must be applied before the 300 volts DC to protect the servo amplifiers. This permits the servo amplifiers motor shaft position R/D converter to become stable before the drive to the power switching transistors is enabled. Taps on the secondary permit adjustment of the 3-phase, 208 volts power delivered to the PSR1 servo amplifier power supply.

The Apex Control Junction Box is the distribution point for the cable runs from the Antenna Pedestal Room and the FRM Cables. Manual control slewing switches on the cover of the box permits the FRM to be manually operated from the Apex. An LED display panel in the box provides a visual indication of limit switch status. Metal Oxide Varistor (MOV) surge arrestors in the box clip lightning-induced surges on all the Apex signal lines. The clipping levels are about 10% above the peak signal levels.

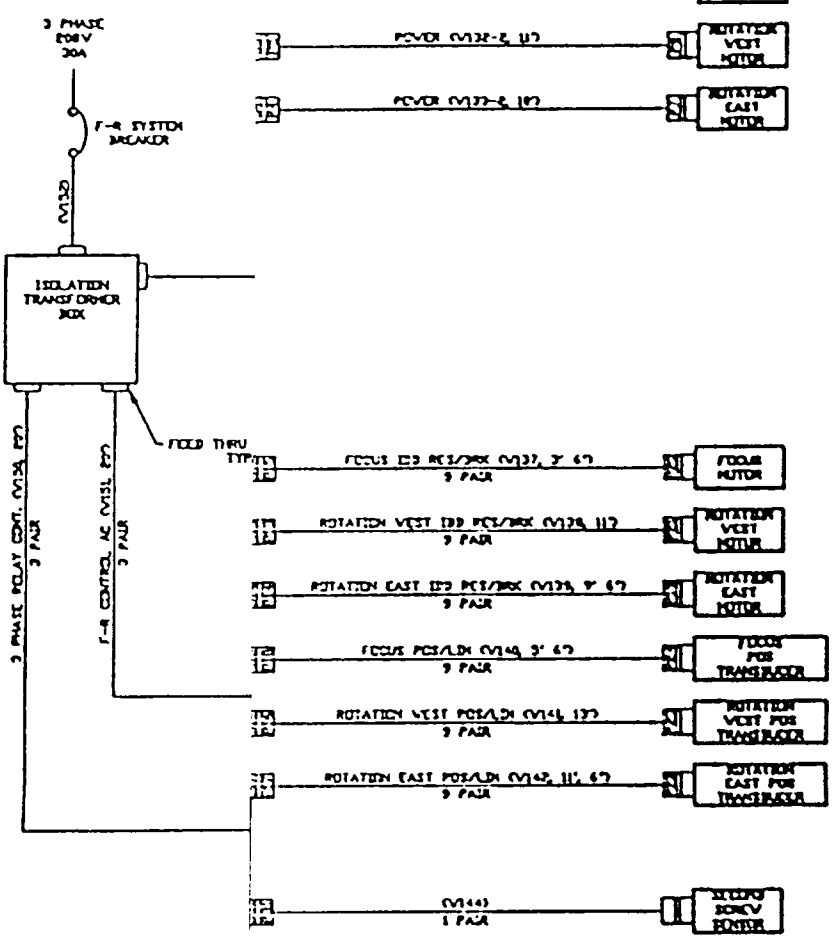
An Emergency Stop switch mounted on the Apex Control Junction box permits the drives to be manually inhibited at the Apex. Control circuitry senses the switch state to inhibit motor drive amplifiers and brakes when actuated.

The Pedestal Room Junction Box is the distribution point for the signal cables to the Apex and the F-R Control System modules in the F-R System rack. This junction box also has MOV surge arrestors on all signal lines.

The Apex Motor Junction Box is the connection point for the motor power cables from the servo amplifiers and the motors on the FRM. It also has MOV surge arrestors to mitigate the effects of lightning-induced surges on the motor cables.


REV | DATE | DRAWN BY | APPROVD BY | DESCRIPTION
S104 FR INTERFACE
COMMAND/MONITOR REQUESTS
MCB
MCB
COMMAND/MONITOR DATA
ROTATION SERVO INFORMATION
FOCUS SERVO INFORMATION
S101 FR CONTROL
ROTATION BRAKE CONTROL
FOCUS BRAKE CONTROL
3 PHASE RELAY CONTROL
ROTATION MOVE CONTROL
FOCUS MOVE CONTROL
ROTATION DATA REQUEST
FOCUS DATA REQUEST
LED INDICATORS
S103 SWITCHING
ROTATION BRAKE POWER
FOCUS BRAKE POWER
RESOLVER EXCITATION
3 PHASE RELAY POWER
FRM
ROTATION WEST BRAKE
ROTATION EAST BRAKE
FOCUS BRAKE
S102 APEX INTERFACE
ROTATION WEST POSITION AND LIMITS IN
ROTATION DATA
FOCUS DATA
ROTATION EAST LIMITS IN
FOCUS POSITION AND LIMITS IN
ROTATION WEST POSITION RESOLVER AND LIMIT SWITCHES
ROTATION EAST POSITION RESOLVER AND LIMIT SWITCHES
FOCUS POSITION RESOLVER AND LIMIT SWITCHES
S105 FR POWER
MANUAL CONTROLS
APEX POWER
SYSTEM POWER
TO EASTOP RELAY COIL
EMERGENCY STOP SWITCH
SERVO AMP
ROTATION WEST MOTOR POWER
ROTATION EAST MOTOR POWER
FOCUS MOTOR POWER
ROTATID SERVO STATUS
FOCUS SERVO STATUS
ROTATION WEST MOTOR
ROTATION EAST MOTOR
FOCUS MOTOR
POWER TRANSFORMER
ANTENNA POWER
3 PHASE POWER IN
110V TO FR
SWITCHED 3 PHASE AND 110V TO SERVO AMP
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
MATERIAL
FINISH
F-R SYSTEM
F-R SYSTEM BLOCK DIAGRAM
NATIONAL RADIO ASTRONOMY OBSERVATORY
SOCORRO, NEW MEXICO 87801
DRAWN BY
DESIGNED BY
APPROVED BY
STAFF
SIZE
SHEET 1 OF 1
ACAD : FRBLOCKD

DATE	DRAWN BY / APPROVED BY	DESCRIPTION
[illegible]	[illegible]	REDRAWN WITH ACAD



F-R
SYSTEM

NATIONAL RADIO
ASTRONOMY
OBSERVATORY
SOCORRO, NEW MEXICO 87801

-R SYSTEM
ABLE STRUCTURE
IRING DIAGRAM

DRAWN BY	DATE
VEBER	4-87
DESIGNED BY	DATE
VEBER	4-87
APPROVED BY	DATE

ACAD : FRCABLST

1 of 1 | DRAWING NUMBER D55007V001 | REV. D | SCALE : —

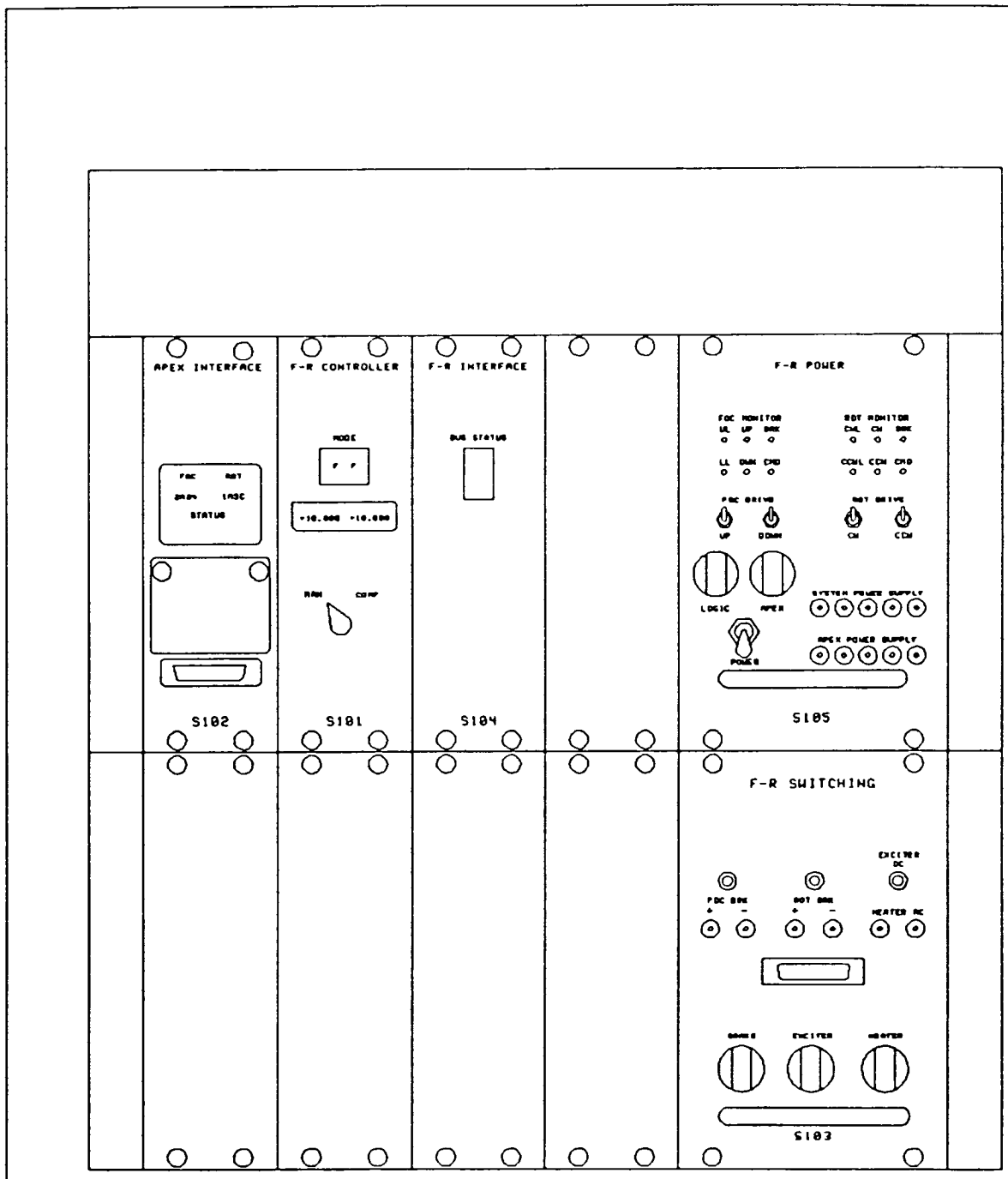
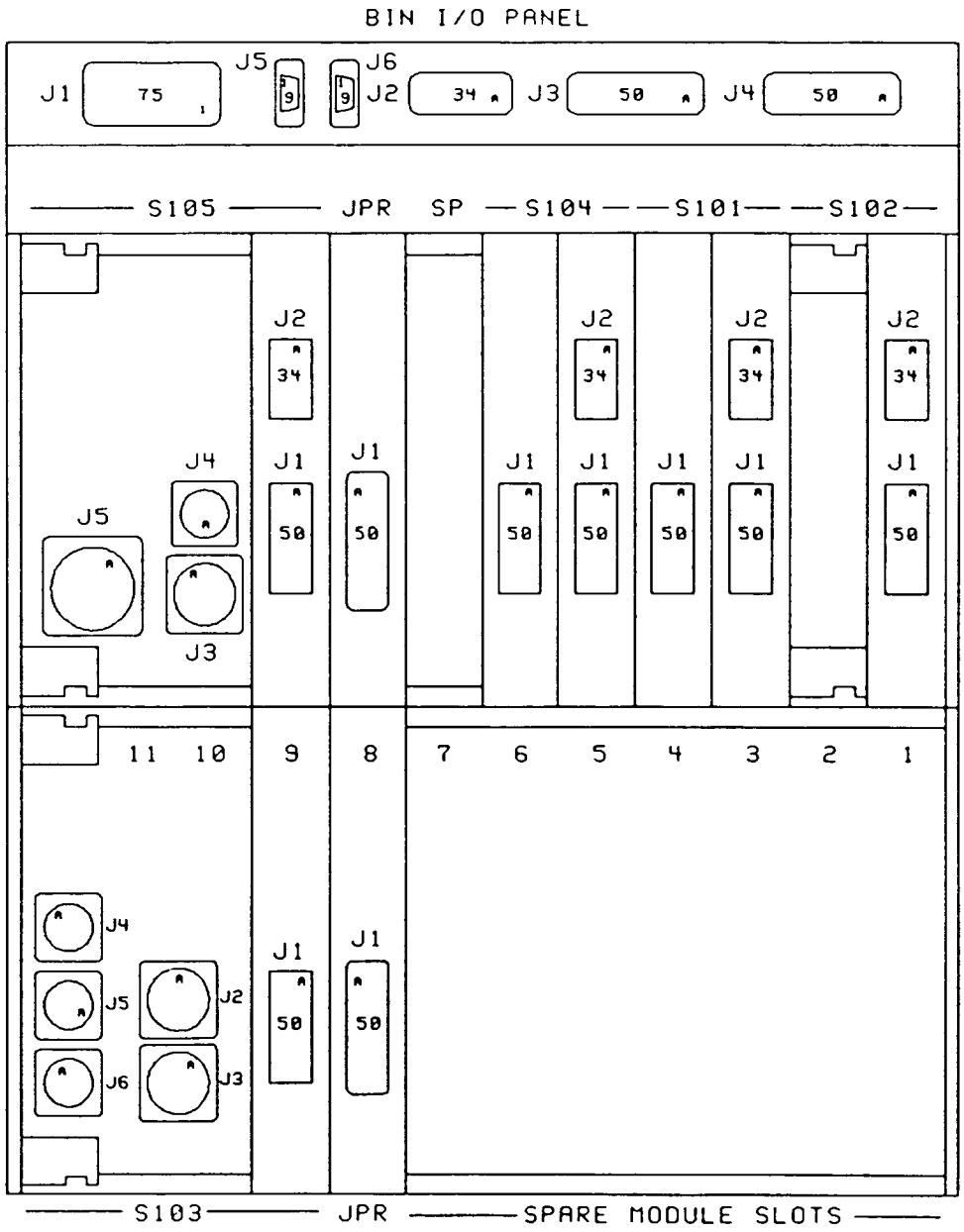


FIGURE 13. F-R CONTROL SYSTEM BIN LAYOUT, FRONT VIEW



NOTES:

CABLES NOT SHOWN
JPR IS THE BIN-TO-BIN JUMPER CONNECTOR

FIGURE 14. F-R CONTROL SYSTEM BIN LAYOUT, REAR VIEW

3.1 S101, The Focus-Rotation Controller

We now address the most important and complex element in the F-R Control System: the S101 F-R Controller module which has two separate, independent (Focus and Rotation) controllers. It is convenient to consider these controllers to consist of four components: (1) the 80188 Control Microprocessor; (2) logic to interface the Control Microprocessor to the FRM Drive-Readout Electronics, the Front Panel Control-Display Logic and the Command-Monitor Interface; (3) control algorithms which define the control actions and (4), a control firmware program which is a set of machine instructions to implement the control algorithms. This section describes the Control Microprocessor and the digital logic which interfaces it to the FRM Drive-Readout Electronics, the Front Panel Control-Display Logic and the Command-Monitor Interface. Section 3.7 describes the control algorithms and Section 3.8 describes the control firmware programs.

In the S101 module, the functions shared by the two controllers are power reset, Front Panel Control-Display logic, the three-phase contactor control and the YOWP-Drive Lockout logic. Although a single microprocessor controller probably could have been used to drive the two mechanism axes (with the consequent small hardware savings), the control firmware would have been much more complicated for two independent, asynchronous control programs to drive wide band (about 70 Hz bandwidth) servo amplifiers. The control function is computationally intensive which requires a fast microprocessor, capable of driving a large number of I/O devices. The control tasks, programmed in assembly language, operate in 4 kilobytes of EPROM memory and 256 bytes of static RAM memory.

The Focus Control Microprocessor is distributed over the first four sheets of the logic schematic and the Rotation processor is similarly distributed over the second four sheets. The chip layouts on the logic connector boards (C,D,E & F) are identical for the two processors. The final two sheets are for shared functions such as Front Panel Control-Display logic, Reset, Limit and Servo Amp discretes input. This shared logic is distributed over logic connector boards A and B. The 80188 chips are a ceramic leadless chip carrier; they are installed in socket adaptors which are plugged into Universal logic connector boards.

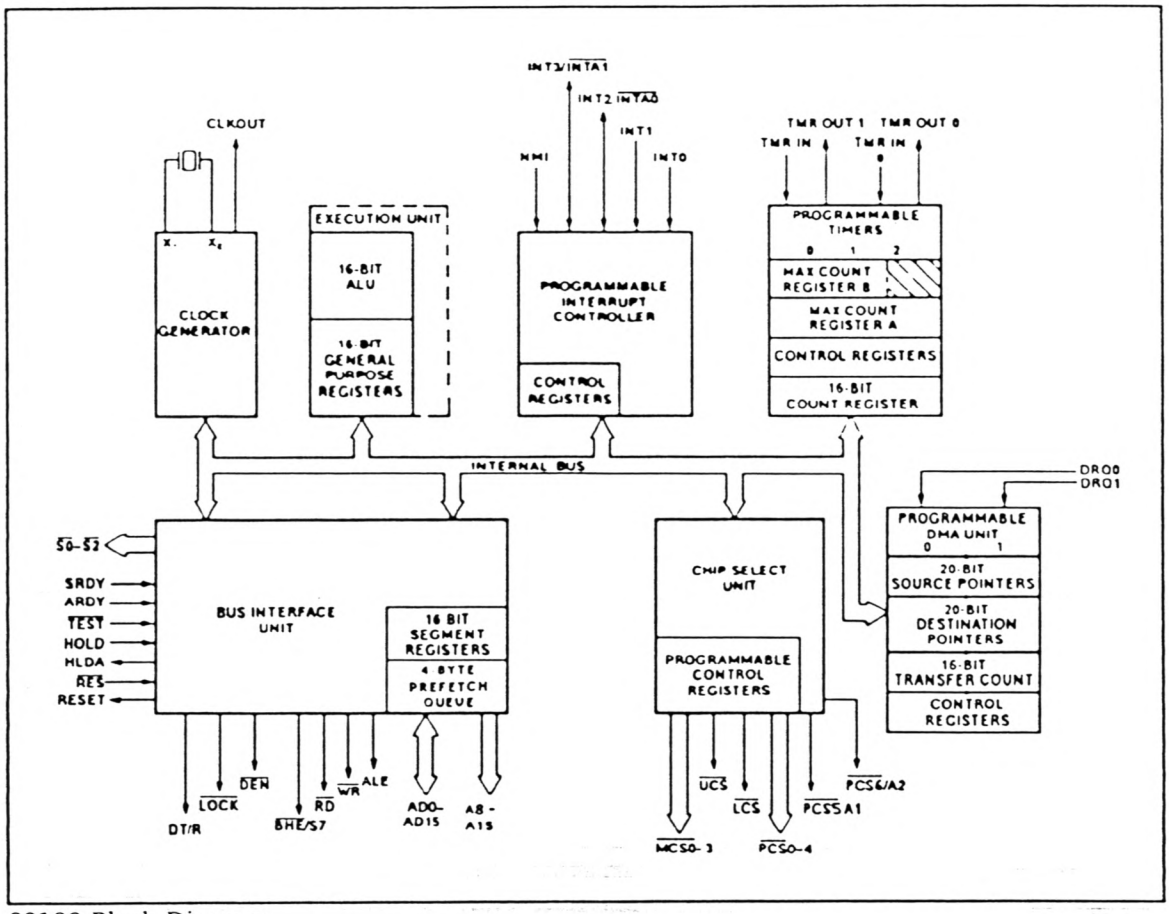
Signal names have an F or R prefix designating Focus and Rotation signals respectively.

Drawing D55007S004 is the F-R Controller logic schematic and D55007A002 is the controller assembly drawing.

Control Microprocessor

The 80188 Control Microprocessor consists of an Intel 80188 microprocessor, two Intel 8755 2 Kilo-byte EPROM read-only memories for the firmware program, an Intel 8156, 256 byte RAM memory for storage of values and states, a 74LS245 bi-directional bus driver to buffer the Address-Data Bus (BADX) and a 74LS373 latch to store the least significant byte of the program address. These six chips (with a little supportive logic) are a small, self-sufficient microprocessor which can execute firmware programs as large as 4 Kilo-bytes and store and recall up to 256 bytes of value or state data. The processor clock rate is 5 Mhz.

The choice of the 80188 over other possible microprocessors was based upon powerful features of the 80188. Examples are: operation at clock rates up to 8 Mhz, 16 bit internal architecture, an internal 16-bit fixed-point hardware multiply-divide capability, a two-channel programmable DMA Access unit, three programmable 16-bit Timer/Counters, a programmable Interrupt Controller, an Internal Peripheral Interface and a more powerful instruction set than the 8088/8086, Z80 or 8085. An example of these enhanced instructions is the BOUND instruction which compares a 16-bit value in a register with two other 16-bit values in two memory locations. This single 80188 instruction would require several instructions with other processors.



80188 Block Diagram

An Intel block diagram of the 80188 is shown above. The major components of the 80188 are a CLOCK GENERATOR, a BUS INTERFACE UNIT, an EXECUTION UNIT, a PROGRAMMABLE INTERRUPT CONTROLLER, PROGRAMMABLE TIMERS, a PROGRAMMABLE DMA UNIT and a CHIP SELECT UNIT. The Interrupt Controller, Timers, DMA Unit and Chip Select Unit is briefly described later in this section because their operating characteristics are particularized to the F-R control function. Intel Block diagrams of these units are included in the descriptions. The Clock Generator, Bus Interface Unit and Execution Unit are not described in this manual since their operation is not particularized to the F-R control function. The reader is referred to Intel manuals for descriptions of these units.

Not shown on the Intel 80188 block diagram is the INTERNAL PERIPHERAL INTERFACE (IPI) which is a set of 256 control registers distributed over the Interrupt Controller, the Timers, DMA Unit and Chip Select Unit. These registers are initialized for the desired hardware and software operating characteristics and are updated during the course of program execution. Descriptions of the IPI in Intel data books give the impression that the IPI is a distinct block in the 80188 logic. This impression is a bit misleading; the reader should recognize the distributed nature of this control logic. The IPI description in this manual narrowly focuses on the setup and updating of these control registers for the F-R control function.

Before we plunge into the S101 logic, it is important to state a few principles which influenced the description. This description is restricted to the 80188 Control Microprocessor outlined above. A complete description of the all the 80188 features and capabilities is beyond the scope of this manual for three reasons: (1) This is an F-R System manual, not an 80188 manual; while important features of the

80188, 8755 and 8156 chips are discussed, the manual does not contain complete descriptions; these are available in Intel literature; (2) The 80188 and 8156 are capable of modes of operation beyond those actually implemented in the F-R control system - it is inappropriate to describe these unused features in this manual; (3) the 8755 EPROM programming operations are not discussed since it is accomplished on a commercial EPROM programmer which is not part of the F-R System.

It is assumed that the reader has access to data books for the 80188, 8755 and 8156 chips and has a working knowledge of their characteristics. Readers unfamiliar with these chips are urged to augment this manual by reference to this literature. For convenience in working on the S101 module, an abbreviated set of data sheets for these chips is included in Section 7.0.

If interrupts are to be used, a small 80188 system requires at least two program memory areas: an upper-address memory because the processor (power-on) reset vectors the instruction counter to a high memory location (FFFF0) where the processor and system initialization code operates, and a lower-address memory which has the interrupt vector table. (As implemented in S101, the interrupt service code also resides in lower memory because it is convenient to locate this code in the same EPROM read-only memory as the interrupt vector table.) The program memory chips are 2-kilobyte Intel 8755 EPROM's and by Intel convention are Code Segment (CS) memory; program instructions are fetched (only) from the CS memory and are referenced (only) to the contents of the CS register. (See Section 3.8 and the Intel literature for a definition of memory segment conventions and address formulation.) During the course of program execution, the CS register value is switched between F800 and 0000. The F800 value is used to access the upper EPROM memory which has (Physical) addresses ranging from FF800 to FFFFF and the 0000 value is used to access the lower EPROM memory which has Physical addresses ranging from 00000 to 007FF.

A middle memory, an 8156, 256 byte static RAM contains the 80188 STACK, values and states. This memory is defined as the Data Segment (DS) and STACK Segment (SS) memory. The DS and SS registers are set to 0000H. Any memory reference instruction which reads or writes data to this RAM memory uses the contents of either the DS or SS register in formulating the memory address. This property can be overridden by a segment override prefix but is never used in the controller firmware.

A convenient feature of the 8755's and 8156 are address latches; since the 80188 uses a multiplexed address-data bus, the lower 8-address bits of address to a memory or peripheral chip must be stored in an address latch because the address is transient; 8 bits of data are impressed upon the bus after the 8-bit address. These two chips also have I/O ports for sensing and controlling the states of external devices. When used as output ports, the output states are latched. The integration of address latches, memory and I/O ports in these two chips enables a powerful 80188 controller to be built with a very low chip count.

Like the 80188, the 8755 and 8156 chips have internal registers which control the operating modes. The 8755 has a Data Direction Register which designates the mode of each port bit; they may be individually set to be either inputs or outputs. The 8755 DDR is initialized by firmware after a processor reset. All Port bits for EPROM FA22 Ports A and B are set to the output mode. Port bits PB3 through PB7 on EPROM FD22 are set to input mode, the rest (PB0 through PB2) are set to output mode. The 8156 has a command register which controls its modes; the port bits can (as a set) be set to be either inputs or outputs and Port C bits can also be used as hand-shake terms for Ports A and B. This latter mode is not used; all three ports are commanded to the output mode during 80188 initialization. The 8156 also contains a counter/timer which is not used in the S101.

8755 and 8156 Data Sheets in Volume II show the programming details of the control registers for these chips.

The 8156 and 8755 memory and I/O operations are discussed below.

The IO/Mbar input term on the 8156 RAM chip controls the selection of either the memory or I/O mode; when the chip is selected and IO/Mbar is low, the 8156 RAM memory inputs or outputs data to/from the Buffered Address-Data Bus. Conversely, when the chip is selected and IO/Mbar is high, the I/O ports input or output data to/from the Buffered Address-Data Bus. (The I/O port operations are of course dependent upon the modes programmed into the 8156 command register). The 8755 memory-I/O mode control logic is identical to the 8156 memory-I/O mode control logic.

The 80188, like other Intel microprocessors, uses a multiplexed Address-Data Bus in which 8 bits of address are first asserted on the bus followed by 8 bits of data to be input or output. Because the address bits on the BADX bus are transient, they must be sampled and latched for subsequent use in enabling memory or I/O. This Address-Data Bus is the path-way for instructions from the 8755 EPROMs and data to/from the 8156 and interface devices. A 74LS245 bus transceiver buffers the 80188 bus (PADX) because the Address-Data Bus (BADX) must drive a number of logic signal nodes at high speed. The transceiver signal flow direction is determined by the READbar signal from the 80188 which drives the 74LS245 direction pin; it is quiescent high at all times until asserted low during a processor read operation. When READbar is asserted low, the signal flow is from some device on the Buffered Address-Data bus (BADX) to the 80188 Address-Data Bus (PADX). Using READbar to control BADX bus direction (during a major timing cycle, i.e. a READ, WRITE or INTA cycle) makes the lower 8 bits of address (conveyed by the bus) available to the chips which must latch it. These chips are the 8755's, 8156 and LS373. The 245 output enable is tied low so that the transceiver outputs are always active.

The 74LS373 latch loaded by the trailing (i.e. falling) edge of the 80188 ALE (Address Latch Enable) stores the 8-bit address impressed upon the BADX bus. These 8 bits are the low order component of the 20 bit program and 16 bit I/O addresses.

Figure 14 (next page) depicts the Focus Controller block diagram; the Rotation Controller is logically identical. The logic discussions reference the Focus controller; they are equally applicable to the Rotation controller.

The 80188 X1, X2 inputs are driven by a 10 Mhz crystal; the internal clock rate is 5 Mhz and the CCLKOUT term is a 5 Mhz TTL signal used for system clocking.

The 80188 major timing diagrams are depicted on the pages following the Focus Microprocessor block diagram.

We now address the distributed Internal Peripheral Interface (IPI) which provides most of the interface and control logic for an 80188 microprocessor system. The IPI register states determine the operating characteristics of the Interrupt Controller, Timers, DMA Unit and Chip Select Unit. The IPI is a 256 byte internal memory, organized into 16 bit control words which must be initialized by firmware after the 80188 is reset. Control Word locations in the memory are referenced by an offset from the base address in the Relocation Register; the reader will note an offset value for each of these parameters in the Intel literature. These offsets are cited in the description below.

The 80188 Internal Peripheral Interface may be read or written at any time during the execution of the program (DMA, Interrupt and Timer operations in the F-R control firmware require that this be done). We will discuss the relevant features of this interface below and in the firmware description in Section 3.8. The base address stored in the Relocation Register is FF00; this is the default power-on reset value which is retained, unaltered by the processor initialization firmware. Data Sheets in Volume II show the structure of the 80188 Internal Peripheral Interface registers.

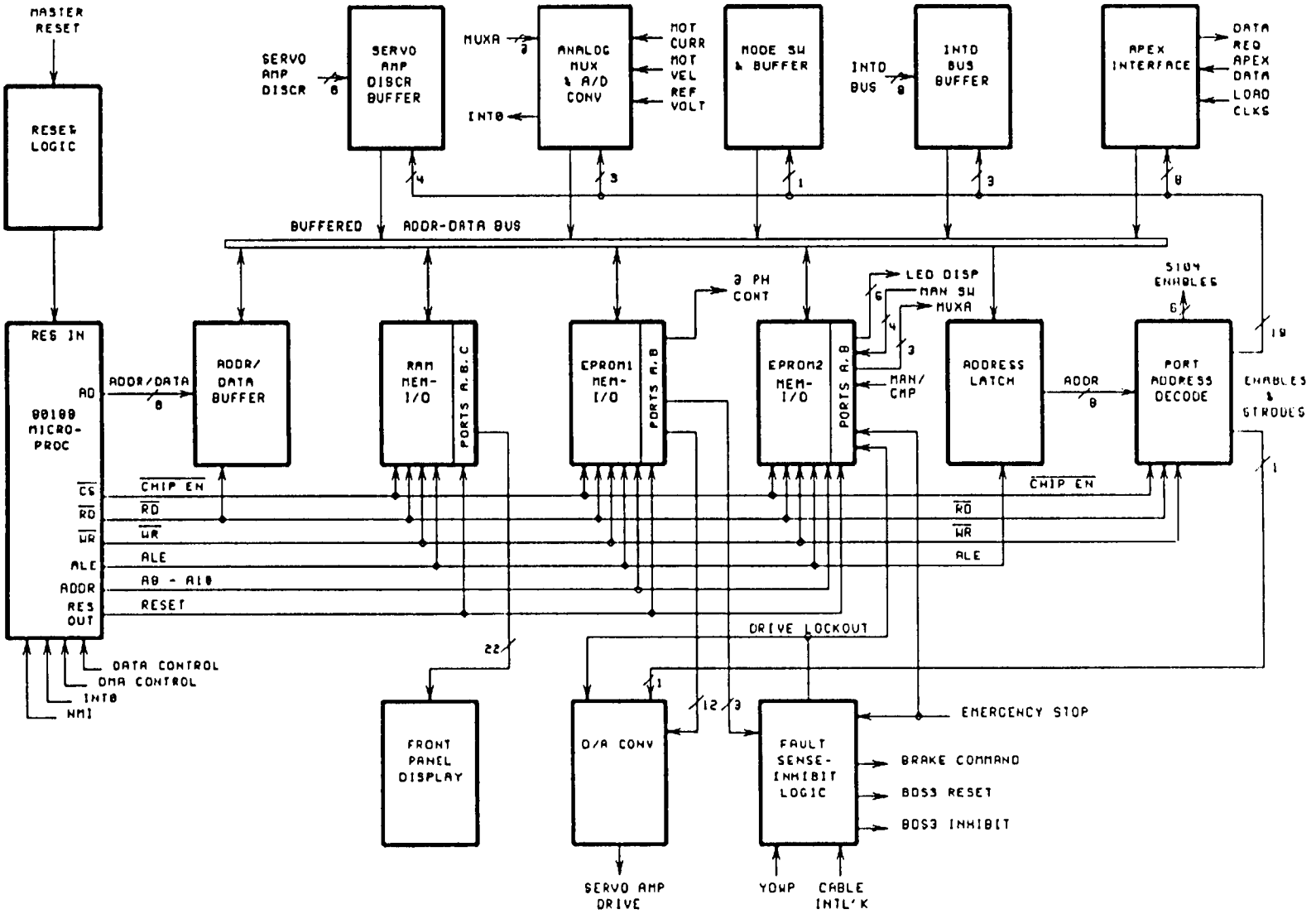
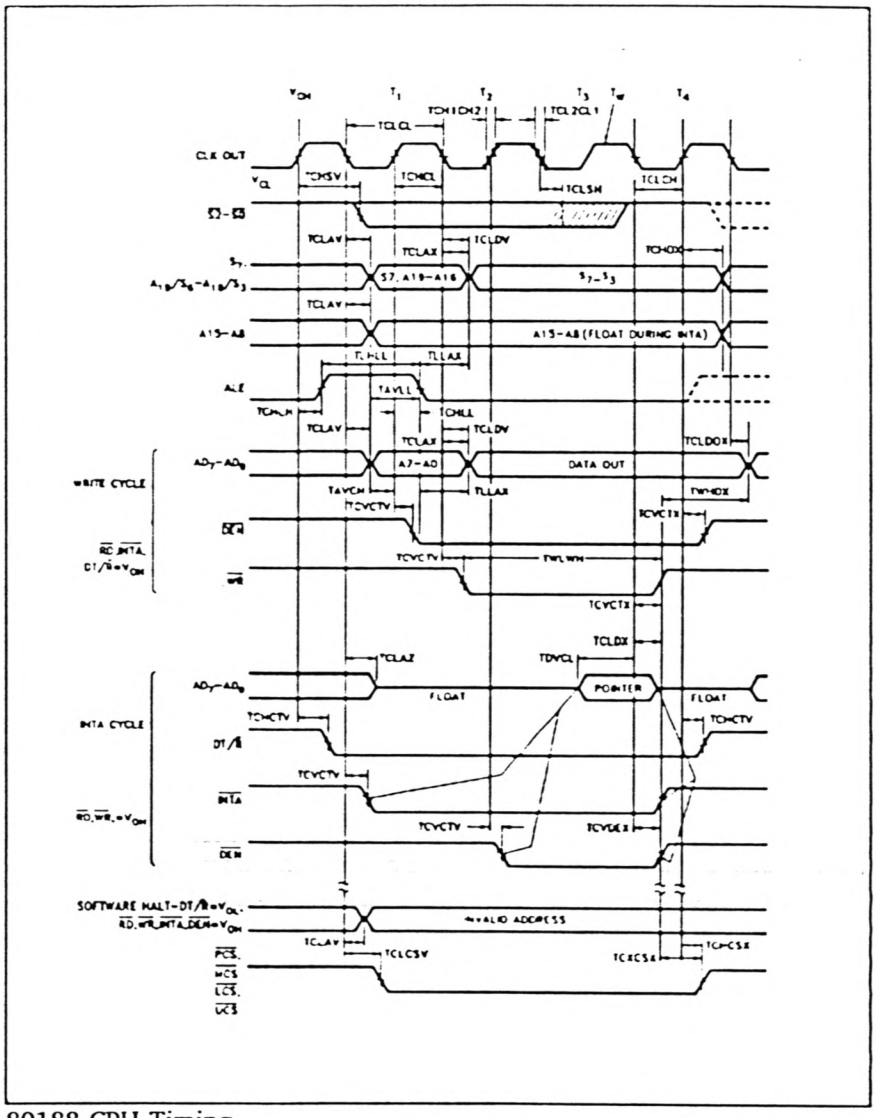


FIGURE 15. FOCUS CONTROLLER BLOCK DIAGRAM

Under control of the associated IPI registers, the Chip Select Unit generates the memory and I/O chip selects; the IPI register states used for S101 chip selects are described below.

The FUCSbar memory chip select (IPI offset A0) is used to select the high (address FFFFF downwards) memory. FUCSbar is initialized by processor reset to enable a 1 kilobyte memory; it must be programmed to a lower address memory boundary by the initialization firmware in order to read the whole 2 kilobyte EPROM memory.

The register in the Internal Peripheral Interface which specifies the FUCSbar base address is the UMCS register. UMCS is initialized by firmware to a value of 0FFBD, which is the value for a 2 kilo-byte memory block with 1 memory wait state; the external ready is ignored. This permits a slower response from the EPROM memory.

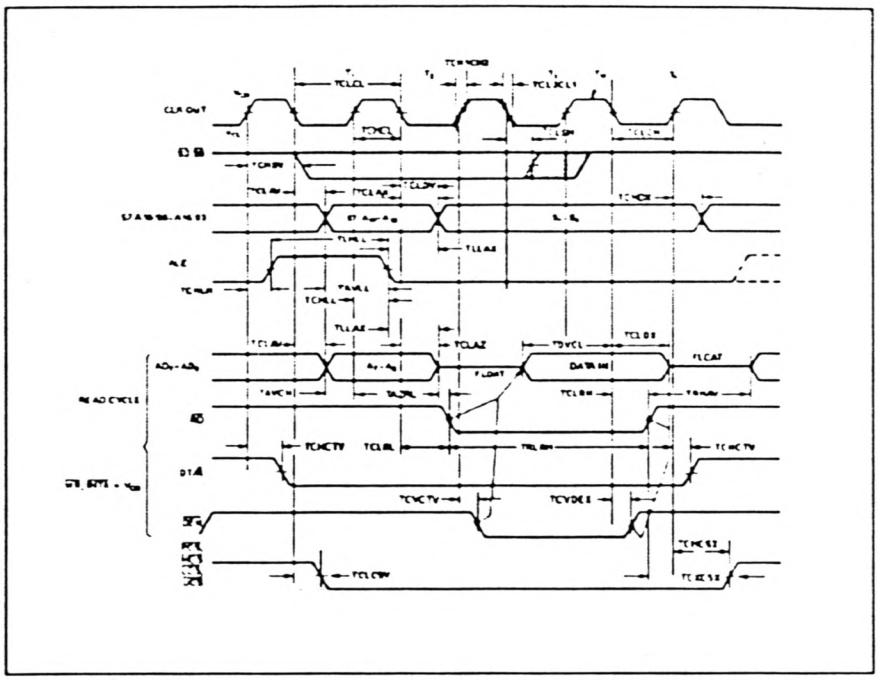


80188 CPU Timing

The FLCSbar memory chip select (offset A2) is used to select the lowest (address 00000 upwards) memory. Unlike FUCSbar, it is not initialized by processor reset and will be disabled until initialized by firmware. The register in the Internal Peripheral Interface which specifies the FLCSbar upper address is the LMCS register. LMCS is initialized to a value of 007D, which is the value for a 2 kilo-byte memory with 1 memory wait state; the external ready is ignored.

Mid-range memory chip selects, MCS0, MCS1, MCS2 and MCS3 are used to select up to four memory chips between the upper and lower memories and are arranged in ascending address order; e.g. MCS0bar selects the lowest address range memory and MCS3bar selects the highest address range memory. The memories must be identically sized and 8 kilobytes is the smallest memory for contiguous address ranges. The address ranges specified may not overlap the upper or lower memory chip select address ranges. After processor reset, these chip selects are disabled and must be initialized by firmware. The registers in the Internal Peripheral Interface which specify the base address and memory sizes are the MMCS (offset A6) and MPCS (offset A8) registers. MMCS (base address) is initialized to 03FD. MPCS is initialized to 81BD which is a 2 kilo-byte segment with 1 memory (both read and write) wait state; the

external ready is ignored. This permits a slower response for RAM memory operations.



80188 CPU Timing

The 8156 RAM is selected by the FMCS0bar memory chip select. This is the first (lowest) of the Mid-range memory chip selects. MCS1, MCS2 and MCS3 are not used in the S101.

The Chip Select Unit provides 7 peripheral Chip Selects (PCS0bar ... PCS6bar) which are used to select I/O devices. These chip selects are active for 7 contiguous blocks of 128 addresses above the base address; FPCS0bar enables the first block of 128 addresses above the base address; FPCS1bar enables the second set, etc. After processor reset, these chip selects are disabled and must be set up by firmware. The IPI register which defines the addresses of these chip selects is the PACS register (offset A4). The Peripheral Base Address is 00000 and the PACS register is initialized to 003D which utilizes 1 wait state; external ready is ignored. The I/O enable addresses thus start at 0000. These peripheral chip selects are programmed for 1 wait state and external ready is ignored. This permits a slight delay in reading from or writing to an I/O port.

FPCS0bar is the first peripheral chip select (for the first set of 128 I/O addresses) and is used to select EPROM 1 (8755, FA22) I/O ports. The ports are addressed as described above for a memory read but the FPCS0bar drives the IO/Mbar term high which selects the I/O ports. When the 80188 WRITEbar term drives the IOWbar term low, the data on the BADX bus is written into the addressed port latches. The ports are selected by the state of address bits AD0 and AD1; when these two bits are low Port A latches are written into, when AD0 and AD1 are high Port B latches are written into. Address 0000 is Port A, Address 0001 is port B, address 0002 is the Port A Data Direction Register and address 003 is the Port B Data Direction Register.

EPROM 2 (FD22) I/O ports are selected by the FPCS1bar term (for the second set of 128 addresses) using logic identical to the EPROM 1 select logic. Address 0080 is Port A, address 0081 is Port B, address 0082 is the Port A Data Direction Register and address 0083 is the Port B Data Direction Register.

FPCS2bar is the third peripheral chip select (for the third set of 128 addresses) and is used to select the RAM I/O ports via or-gate FG43-3. Inverter FE43-6 drives the RAM IO/Mbar pin high which designates an I/O port operation. As described above for memory read/write, the port address on the Buffered Address-Data Bus is latched by ALE; the 80188 then asserts the WRITEbar signal which stores the state of the Buffered Address-Data Bus in the addressed port latches. Address 0100 is the Ram Timer and Control Register, address 0101 is Port A, address 0102 is Port C, address 0103 is Port C, address 0104 is the Ram Timer Low register and address 0105 is the Ram Timer High register.

FPCS3bar is the fourth peripheral chip select (for the fourth set of 128 addresses); it is used to both generate read and write enables for states and values which must be input or output and strobe signals to trigger some I/O action or acknowledgement. Only 32 of the possible 128 addresses are used. This logic is used for processor I/O operations with the FRM Drive - Readout electronics, the Front Panel Control-Display logic and the Command-Monitor Interface. Four 74LS138 decoders are driven by the five lowest bits of the address latch (EA12, a 74LS373). The decoders are enabled by the G2A input which is driven by gates FC43-8 and EA44-3. If the decoder is to generate a strobe or enable, FPCS3bar is low true on EA44-2. Either a FREADbar or FWRITEbar on gate FC43-8 will strobe the decoder array. These enables-strobes are shown on Sheets 3 and 9 and are not listed in detail here.

The Control Microprocessor memory and I/O addresses, chip select and chip assignments are depicted in the memory - I/O map below. Note that I/O is not memory-mapped.

MEMORY - I/O ADDRESS MAP (addresses - Hex)

ADDRESS	MEMORY CHIP	CHIP SELECTS	I/O PORT ADDRESS	CHIP SELECTS	I/O FUNCTION
FFFFF	EPROM 1	FUCSbar			
.	FA22				
.					
FF800					
0FFFF	80188				
.	INT PERIPH		019F	FPCS3bar	ADDR-DATA
.	INTERFACE		.	"	BUS I/O
0FF00			.	"	"
			0180	"	"
			017F	FPCS2bar	RAM I/O
020FF	RAM	FMCS0bar	.	"	FG22
.	FG22		.	"	"
.			0100	"	"
02000			00FF	FPCS1bar	EPROM 2 I/O
			.	"	FD22
			.	"	"
			0080	"	"
007FF	EPROM 2	FLCSbar	007F	FPCS0bar	EPROM 1 I/O
.	FD22		.	"	FA22
.			.	"	"
.			.	"	"
00000			0000	"	"

We now consider the 8755 EPROM memory read cycle using chip FA22 as a typical case. Remember that for 8755 memory read operations, IO/Mbar is low. When the 80188 initiates a memory read operation the memory chip selects become active. The upper memory chip select (FUCSbar) is or-red in gate FC43-3 to enable the upper memory EPROM, FA22. The lower 8-bits of address (AD0...AD7) on the Focus Buffered Address-Data bus (FBADX) and higher order address bits FA8 .. FA10, are sampled and latched by the trailing edge of ALE. This address designates the 8-bit contents of a memory location in the EPROM. FPCS0bar is quiescent (high) when the EPROM memory is being read; inverter FE43-2 holds the IO/Mbar input low which causes the EPROM memory data to be impressed upon the BADX bus when the 80188 drives the READbar signal low. The 80188 samples the FBADX data on the trailing edge of FREADbar; this completes the EPROM memory read cycle.

The 8755 I/O port read and write cycles are similar to the memory operation described above, but in the I/O mode, IO/Mbar is high which causes the I/O ports to become active. Again we use FA22 as a typical case. An 80188 I/O operation activates the peripheral chip select FPCS0bar which selects EPROM FA22 and drives IO/Mbar high via inverter FE43-2. In an I/O operation, the AD0 .. AD11 address code selects among I/O Port A, I/O Port B or Port A or B Data Direction registers. When WRITEbar goes low, the state of the FBADX bus is written into the selected port or DDR latches. The 8755 IOWbar input is not dependent upon the state of IO/Mbar when writing to the port latches. In an I/O read operation, when READbar goes low, port A or B bits (the DDR states cannot be read) in the input mode are impressed upon the BADX bus and sampled by the 80188 on the trailing edge of READbar. IORbar is pulled high because it does not need to be driven in this logic implementation.

The F/R control firmware reads the state of 8755 I/O port output latches states onto the BADX bus; this is an alternate to keeping a state memory in RAM and provides a verification that the output ports are working properly. To read these ports, an I/O read operation is performed: the FPCS0bar signal goes low which selects the FA22 chip and drives IO/Mbar high. When READbar goes low the state of the addressed output register (i.e. Ports A or B but not the DDR latches) are impressed upon the BADX bus and sampled by the trailing edge of READbar.

The lower memory EPROM, FD22 is driven by identical logic activated by the lower memory chip select, FLCSbar and second peripheral chip select: FPCS1bar.

We will now consider the RAM read and write cycles. The 256 byte RAM is used for storage of program values, states and the 80188 stack which is the upper portion of the RAM. The chip is enabled by FMCS0 via or-gate FG4303 (the 8156 chip enable is high-true). For memory operations the FPCS2bar term is high (it is a memory operation, not an I/O operation) so the IO/Mbar pin on the 8156 will be driven low by FE43-06 which designates a memory operation to the 8156. The 8-bit address on the Buffered Address-Data bus is latched into the chip memory latches on the trailing edge of the ALE (just like the EPROMs above). If the operation is a memory read, the 80188 asserts a low-true READbar signal which causes the RAM to assert the data stored in the addressed byte onto the Buffered Address-Data Bus. If the memory operation is a write, the 80188 will assert the low-true WRITEbar signal to cause the RAM to store the data on the bus in the addressed memory.

The 8755 and 8156 I/O ports control many devices; the control-sense functions are described in detail in the S101 FRM Drive-Readout Electronics pages later in this section.

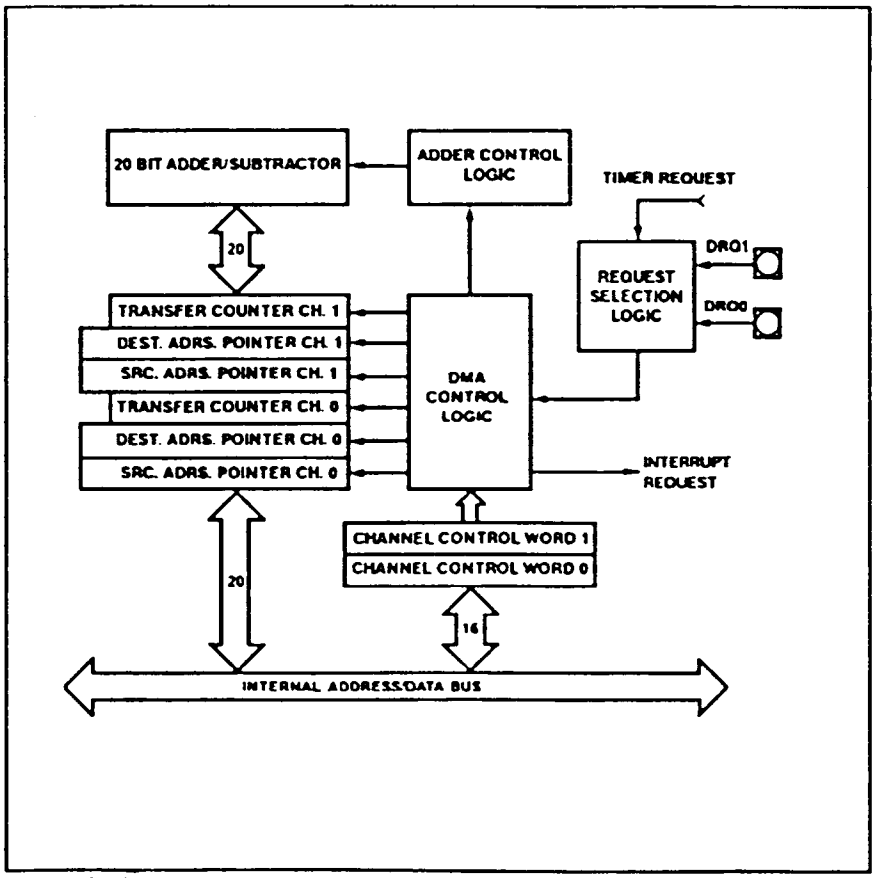
The Control Microprocessor should never be in the halt state because the F-R System is always active if powered. The 80188 has a halt instruction and an associated halt state which may be sensed to induce a processor reset. The firmware does not use halt instructions but noise perturbations on the power line could possibly induce a halt state; this is the reason for the halt reset logic. The logic on the right half of Sheet 1 senses the halt state by testing the low-true 80188 Status lines S0, S1 and S2. These bits are sampled by the trailing edge of ALE (Address Latch Enable). Gate A26-14 decodes the 0,1,1 states for S2, S1 and S0 respectively and drives the D input of flip-flop A21-5. Inverter B3-6 clocks A21-5 reset if the processor is halted. When the processor clock goes low (a few tens of nanoseconds after A21-5 is reset), gate A13-3 triggers one-shot A27-6 (Sheet 14) through gates A22-6 and A22-13. The one-shot A27-6 resets flip-flop A21-5 to re-arm it to sense halt states. The reset one-shot drives the 80188's RESbar inputs. Each processor outputs a RESET output (FRESET and RRESET) which resets the 8156's and D/A registers in both controllers; note that a halt state in either controller will induce a reset. A Reset command from the Antenna Control Computer via the S104 can induce a reset via gate A22-3. Switching the front panel MAN-CMP switch will also induce a reset via A22-3. The period of the one-shot is several tens of milliseconds.

Power-on reset is induced by the resistor-capacitor-diode circuit on dip header A28, sheet 14. The slow charge of the 22 uf capacitor to the switching threshold of 74LS08 A22-10 provides a time delay of about 50 milliseconds after +5 volt power is available (this assumes that the +5 power comes on as a step function which is an idealization of the real case). This delay causes the open-collector 7406 A19-2 to hold the 80188 RESbar low for this 50 milliseconds which is adequate to reset the processor. A 7406 open-collector buffer is used because its output is at +5 when high; this level is a requirement of the 80188 VIH level specification.

After the power reset input goes high, the 80188 begins code execution with the instruction at physical address FFFF0. Processor reset initializes some 80188 internal registers as follows:

Status Word	- F002	Instruction Pointer	- 0000	Code Segment	- FFFF
Data Segment	- 0000	Extra Segment	- 0000	STACK Segment	- 0000
Relocation Register	- 20FF	UMCS	- FFFB		

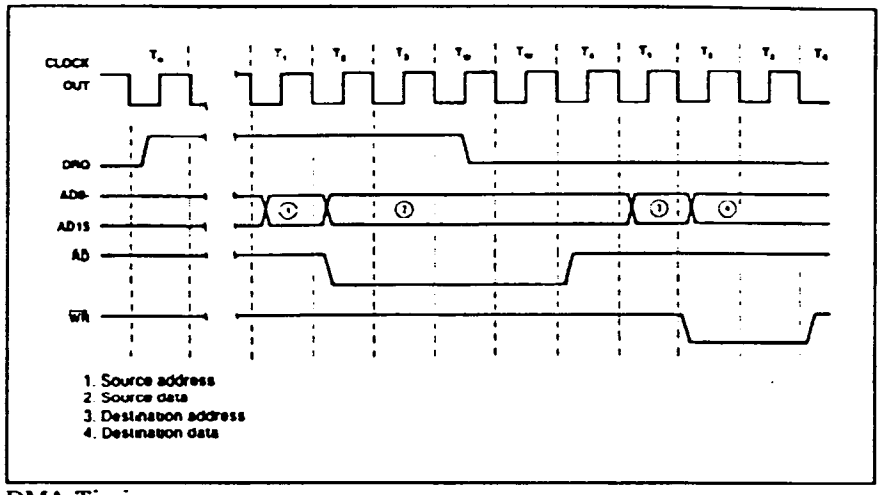
The 80188 has two independent DMA channels. A block diagram of the DMA channels is shown below. DMA 0 channel is used to input four bytes of command information: the Relative Address (via Enable 395) and the low and high bytes of Command argument (via Enables 396 and 397) and a null byte which is ignored. The last (null) byte (Enable 398) activates the Device Acknowledge signal in S104 which terminates the command cycle in the Standard Interface Board in S104. These four bytes of data from fixed source addresses must be input to four fixed memory locations and it is desireable that the input operation be a minimal perturbation to the program execution; this is the reason why a DMA channel was selected for command input. After initialization of the DMA channel registers, when the S104 raises the DRQ 0 request line, the DMA inputs a specified number of bytes from contiguous source addresses into contiguous memory addresses and at completion raises an interrupt to tell the control program that the sequence has been completed. One of the operations of the DMA 0 Interrupt service code is re-initialization of the DMA 0 registers in the Internal Peripheral Interface for another command input sequence. The number of bytes to be input are specified by a transfer count. The reader is urged to read the 80188 data sheets in Volume II for complete details on control of the DMA channels.



DMA Block Diagram

Six registers in the Internal Peripheral Interface control the two DMA channels operations. When a DMA cycle is completed, the channel generates an interrupt to signal to the con-

trol program that the transfer has been completed. The control registers are: CONTROL WORD (offset CA for DMA channel 0), TRANSFER COUNT (offset C8), DESTINATION POINTER (upper 4 bits, offset C6), DESTINATION POINTER, (offset C4), SOURCE POINTER (upper 4 bits, offset C4) and SOURCE POINTER (offset C2).



DMA Timing

The CONTROL WORD (label DOMODE in the firmware) controls the operating modes in DMA 0. The CONTROL WORD is initialized to A766; this sets up memory as the transfer destination, memory address is to be incremented for each transfer, the transfer source is set up as an I/O device and source addresses are to be incremented; the transfers are to be terminated by a zero count in the Transfer Counter Register and an interrupt is to be generated upon transfer count termination. The DMA is synchronized to the source (i.e. S104 causes the transfers) and the DMA 0 channel is set to the highest priority. At each initialization the Start bit is set and Change bit is set for a new initialization.

The TRANSFER COUNT (label D0TC in the firmware) is decremented after transfer and controls the number of transfers. D0TC is initialized to a count of 4.

The DESTINATION POINTER address (label D0DPL & D0DPM in the firmware) is a 20 bit value which is a composite of the 4 and 16 bit values contained in the DESTINATION POINTER registers mentioned above. The upper 4 bits of the pointer are initialized to zero and the lower 16 bits are initialized to 2040 which is location RAC in the firmware temporary storage table in the 8156 RAM. As the DMA sequentially transfers the three bytes, the pointer address is incremented to point to the command argument bytes.

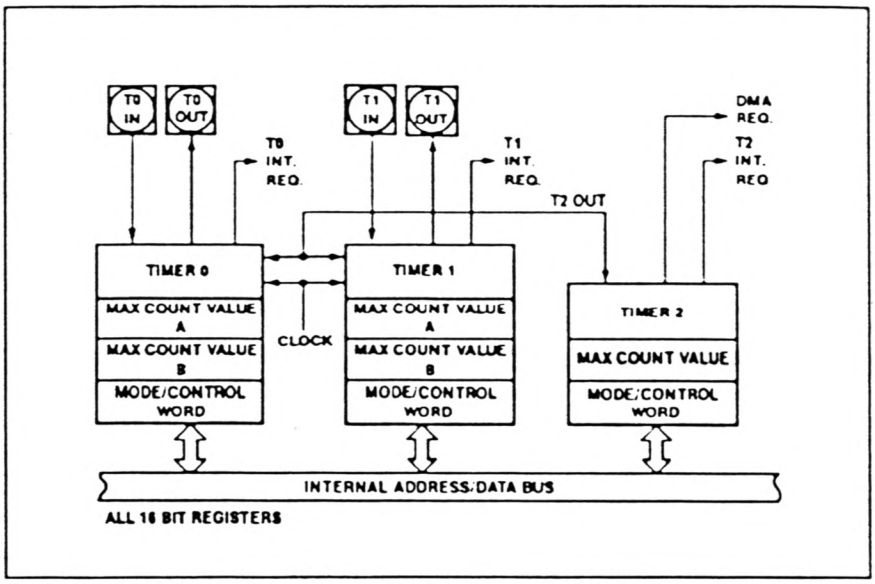
The SOURCE POINTER address (label D0SPL & D0SPM in the firmware) is also a 20 bit composite value and is initialized to zero and 018B (respectively) which is the location RELADD in the firmware external ports equates table. This address is equated to enable F395 and points to the S104 register which contains the Relative Address byte. As the DMA sequentially transfers the three command bytes the pointer address is incremented to F396 and F397 to access the two command argument bytes.

When the 80188 is reset, the Start/Stop bits will be set to Stop and any transfer in process will be aborted.

An example of DMA timing is shown above.

The 80188 has three internal timers for synchronizing program operations with time; two of these (T0 & T1) are connected to 4 external pins and may be used to time external events, etc. They can also be clocked by the processor clock/4 which results in a timer clock rate of 1.25 Mhz. The third timer (T2) is not connected to external pins; it is clocked at the processor clock/4 rate and may be used as a pre-scaler for the other timers. The Control Microprocessor uses T1 and T2; T0 is not used.

The Internal Peripheral Interface uses four registers per timer to control timer operations. The T1 MODE/CONTROL WORD (offset 5E) permits control of the timer modes and a readout of the status of the four registers at any time. The T0 and T1 timers have two MAX COUNT (offset 5C for T1) registers which may be loaded with different values which are the maximum count values for the COUNT REGISTER (offset 58 for T1). The COUNT REGISTER (offset 58 for T1) is a 16 bit register which is incremented by the clock and may be read or written at any time; when the contents reach the MAX COUNT value a timer interrupt will be generated if permitted by the initialization. The reader is urged to consult the 80188 data sheets in Volume II for complete details of these control registers.



Timer Block Diagram

When the 80188 is reset, the timer Enable bit is cleared and the timer output pins are set high.

T1 MODE/CONTROL WORD (label T1MODE in the firmware) is initialized to E009, this sets the ENABLE bit which enables the timer to count, INHbar is set to the non-inhibit state, i.e. the ENABLE bit may be changed by a write to the MODE/CONTROL WORD and INT is set which permits T1 to generate an interrupt at the completion of the time-out. The PERMIT bit enables the output of T2 to be used as a clock input to T1. The CONT bit is set which causes T1 to operate continuously.

T2 MODE/CONTROL WORD (label T2MODE in the firmware) is initialized to C001 which sets the ENABLE bit, permits the control word to be modified by a write and the CONT bit is set which causes T2 to operate continuously.

A block diagram of the timers is shown above.

An important part of a microprocessor is an interrupt system which permits interruption of the operating program to service transcendent conditions which must be dealt with immediately. The 80188 Interrupt Controller accesses vectors which direct program control to code which is particularized for the requirements of the interrupt source.

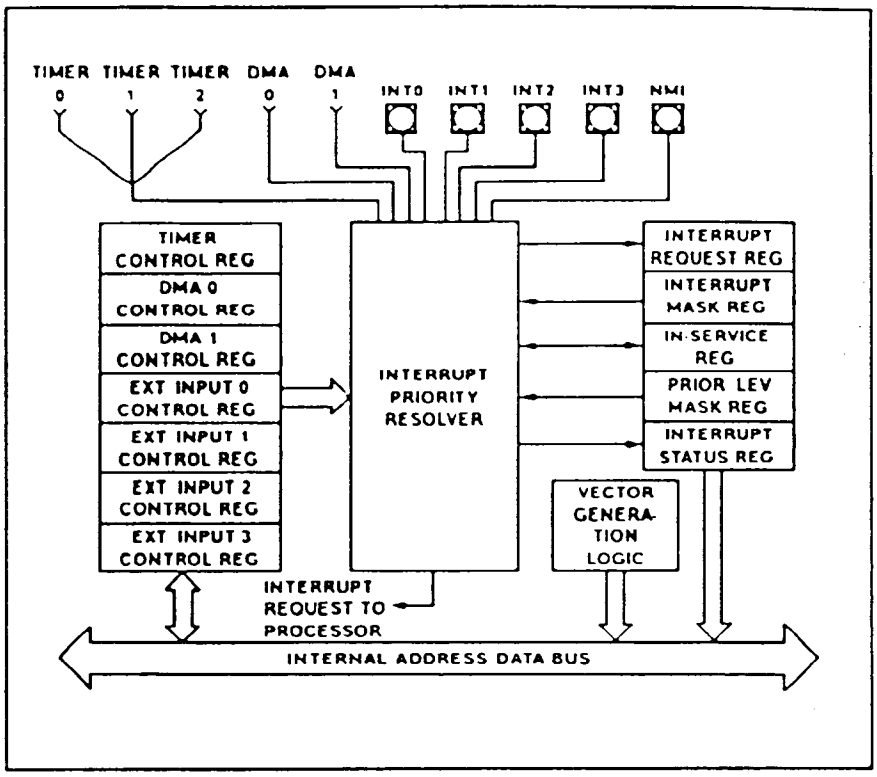
The 80188 possesses a powerful Interrupt Controller capable of operating in many modes. We will only address those features that are used in the Control Microprocessor.

Internally generated and non-cascadable external interrupts generate their own vectors through the Interrupt Controller; maskable hardware interrupts supply the vector to the CPU during the interrupt acknowledge sequence. The non-maskable interrupt uses a predefined internally supplied vector.

The user can program the interrupt sources into any of eight priority levels to resolve contention in the event that different interrupt sources raise their interrupt lines at the same time.

The 80188 can receive interrupts from both internal (e.g. DMA channels, timers, divide error, Bound etc) and up to five external sources. Four of these are maskable (INT0 .. INT3) and one is non-maskable (NMI). The interrupt controller in the 80188 merges these requests on a priority basis for service by the CPU. The NMI (non-maskable interrupt) has a default priority of 1 and is used for transcendent purposes. In the S101, the Control Microprocessor uses the NMI to signal that the Antenna Control Computer is requesting a monitor data readout.

The Interrupt Controller block diagram is shown on this page.



Interrupt Controller Block Diagram

Fifteen registers in the Internal Peripheral Interface control the operation of the Interrupt Controller. The registers and modes used by the Control Microprocessor are described below. The reader is urged to study the register formats in the 80188 data sheets in Volume II.

The Interrupt System is turned on by executing the STI instruction which sets the INTERRUPT ENABLE flag in the FLAG REGISTER; resetting this flag will inhibit the Interrupt System.

INTERRUPT CONTROL registers (offsets 32 through 3E) for INT0 through INT3, the DMA channels and the timer define the priority associated with the interrupt source and have a mask bit which either enables or inhibits the ability of the source to interrupt the CPU.

The EOI (end of interrupt, offset 22) register is a command register which can only be written into. At the end of a set of interrupt service code a bit pattern is written into the EOI register to signal that an interrupt service sequence has been completed. The interrupt service routines used in the S101 set the following values into the EOI register: NMI - 0002; DMA 0 - A; Timer 1 and 2 - 8; INT0 - C and Bound - 5. A 0 is set into the SPECIFIC bit.

The IN-SERVICE register (offset 2C) contains an in-service bit (IS) for each interrupt source. The IS bit is set to indicate that a source's service routine is in process. When an IS bit is set, the Interrupt Controller will not generate interrupts to the CPU when it receives interrupt requests from devices with a lower programmed priority level. The TMR bit is the IS bit for all three timers, the D0 and D1 bits are the IS bits for the DMA channels and 10 through 13 are the IS bits for the four external interrupt pins. The control registers which are used by the Control Microprocessor are the Interrupt 0 (INT0CR in the firmware), the DMA 0 (ID0CR in the firmware) and Timer Control register (ITCR in the firmware).

The Interrupt Controller MASK REGISTER (offset 28) contains a mask bit for each interrupt source; a 1 in a bit position corresponding to a particular source masks the source from generating interrupts. The mask bits are the exact same bits which are used in the individual CONTROL REGISTERS; programming a mask bit using the MASK REGISTER will also change this bit in the individual CONTROL REGISTER and vice versa. The MASK REGISTER (MASK in the firmware) is set to 00E8 which enables T1, T2, DMA 0 and INT0.

All interrupt sources are serviced by an indirect call through an element of a vector table. This vector table is indexed by using the interrupt vector type multiplied by four. (See the Intel data books for additional details on the 80188 interrupt system.)

Internal interrupts are generated by specific instructions or by the result of conditions specified by instructions. These internal interrupts have top priority which cannot be altered; these are: Divide Error, Non-Maskable, Breakpoint, Overflow, BOUND, Unused OP Code and Escape. Single Step has priority level two which also cannot be altered.

External interrupt priorities can be altered; these interrupts are: Timer Interupts, DMA Interrupts and INT0 through INT3.

All externally-generated interrupts are sampled at the end of each instruction. In the event of contention between internal and external interrupts; the internal interrupts will be serviced before the external interrupts. Once the service routine is entered and interrupts are enabled, any external interrupt of sufficient priority can interrupt the service routine in progress.

Some 80188 inputs are not used in the S101; these are tied to ground or to +5 through a pull-up as required to make them inactive. These unused terms are: Synchronous and Asynchronous Ready, Hold, Data Request 1, Timers 0 and 1 inputs, Interrupts 1,2,3 and the Test input.

S101 FRM Drive-Readout Electronics

The FRM drive-readout electronics interface is implemented by I/O Ports on one of the 8755 EPROM memories and tri-state interfaces on the Buffered Address-Data Bus (BADX). The FRM Drive-Readout Logic interfaces on the Buffered Address-Data Bus are the Apex Interface registers, the A/D converter register, and the servo amplifier discretes input buffers.

EPROM1 ports are all control microprocessor command outputs (see Sheet 1 of the logic schematic drawing). The command outputs from EPROM1, Ports A and B are: 12 bits of D/A command for the D/A command register - PA0 ... PB3; BDS3 Inhibit - PB4; BDS3 Reset -PB5; Brake command - PB6; 3-Phase command - PB7. All bits are high-true.

EPROM2 ports are a mixture of both control microprocessor command outputs and manual command and alarm inputs from a human operator or logic. Microprocessor command outputs from EPROM2 are: A/D converter multiplexer address, AC0, AC1 and AC2 - PA0 .. PA2; S105 display LED drive - CMD, BRAKE, DRV DWN, DRV UP, LO LIM, HI LIM - PA3 ... PB0; two spare command bits - PB1 and PB2. All microprocessor command bits are high-true. EPROM2 inputs in ascending bit order are: EMERGENCY STOP, DRIVE LOCKOUT, COMP MODE, DRIVE DWN (or CCW) SWITCH and DRIVE UP (or CW) SWITCH. These are input on PB3 ... PB7 respectively. DRIVE UP and DRIVE DWN are low-true manual command inputs from the S105 slew switches. EMERGENCY STOP is a high-true manual command from a stow switch circuit. COMP is a mode command from the S101 CMP-MAN switch which is high-true in the CMP position. DRIVE LOCKOUT is a low-true alarm input resulting from a fault condition in the Apex or a disconnected cable.

Data from the Apex Interface (S102) is serially loaded via an optical isolator into seven 8-bit registers (Sheet 3) and read by the processor via the 8-bit Buffered Address-Data bus. The clocks are supplied by the S102 and are also input via an optical isolator. The rising clock edge samples the data in the center of the data bits. To request Apex data, the processor outputs an Apex Data Request Strobe (B13-03 Sheet 3). The request strobe clears the seven registers by setting the 8546 register mode control inputs C1 and C2 to the high and low states respectively. After a data acquisition delay, the Apex Interface emits a serial stream of data and load clocks which loads the data into the Apex Interface Registers on sheets 3 and 7. The processor detects the presence of the Apex data by testing the top two bits in the most significant byte register; if the MSB is one and the second bit is a 0 the Apex data has been loaded. DSTOR is the firmware routine which requests the Apex data, tests the register for Apex data presence and unpacks it. In addition to the isolation provided by the serial I/O lines between the S101 and S102, the serial data transmissions reduce the wire count between the two units; six wires serially convey 56 data bits; a great reduction over a parallel interchange and several wires less than a bussed interchange.

The drive to the IDD servo amplifiers is an analog signal from an Analog Devices AD565A D/A converter (EG01, Sheet 5). A 12-bit, two's complement value is loaded into 74LS174's (EE01 and EE10, Sheet 5). The scaling of the D/A converter is 2.5 mv/count or +/- 5 volts full scale. The analog signal working range is 0 to +/- 5.0 volts and is a velocity drive. The minimum signal that the BDS3 will respond to is 10 millivolts. A processor reset (FRESET) or Drive Lockout via 74LS08 FC43-11 will clear the D/A storage register EE01 and EE10.

The control programs sample servo amplifier motor velocity and current (current is a measure of motor load torque). A HI508 analog multiplexer and HI574A A/D Converter (EE24 and EG14 Sheet 6) perform this function. The three-bit multiplexer address (FAC0..FAC2) is provided by the lower memory EPROM (FD22) Port A bits PA0, PA1 and PA2 2 on sheet 2. The A/D converter is triggered by the low true strobe F404. The 12 bit data is read out in two bytes, the lower 8 bits are read by F401 and the upper four bits are read out on the upper four bits of the Buffered Address-Data Bus (FBAD4..FBAD7). The processor masks the four null bits FBAD0..FBAD3, and assembles the values into 12 bit arguments. The A/D chip select is activated for each of these three operations by the LS10 nor gate A26-10 and inverter A24-10. 100 microsecond time constant, low-pass R-C filters filter the two analog signals before conversion. The A/D converter is set up to convert bi-polar signals ranging from +10 volts to -10 volts by the resistor and pots connected to the Ref In, Ref Out and Bipolar Offset pins. Pot EG29 adjusts the zero offset and the gain is set by the adjustment of pots EG29 and CG29. The A/D scaling is 5 mv/count. The low-true A/D end of conversion strobe (STS) signals the 80188 via Interrupt 0 (Int0).

The 12-bit D/A converter (EG01 Sheet 5) is driven by two six-bit 74LS174 storage registers which are loaded by the F406 strobe. The 12 bit register is driven by 8755 EPROM FA22 ports A0..A7 and B0..B3. The reason that the D/A is not driven directly from the EPROM ports is that the ports are loaded at separate times and the D/A output would exhibit wild excursions between loading of Ports A and B. The D/A converter register is cleared by the processor resets and Drive Lockout (discussed below).

Three discretes are used to control the FRM drive electronics: the Focus Brake command which energizes (i.e. releases) the Focus Brake, the Focus BDS3 Reset which initializes the microprocessor on the MC2 board in the BDS3, and the Focus BDS3 Inhibit which inhibits the power switching circuitry in the BDS3. These three discretes are output from EPROM FA22 ports PB6 (brake), PB5 (BDS3 reset) and PB4 (BDS3 Inhibit).

During the quiescent periods between command executions, the BDS3's are inhibited. The TTL low-true drive is derived from gate A6-3 and inverter A8-2. This circuit is enabled by the Drive Lockout described above.

The Focus BDS3 reset actuates a reed relay in the Focus BDS3 servo amplifier; gate A6-6 and inverter A8-4 provide current-sink drive to this relay. This circuit is also enabled by the Drive Lockout described above.

The Focus Brake is energized by a DC supply in the S103 module. The 110 AC to the brake supply is switched by a solid state relay. The current sinking drive to the relay is provided by gate A7-3 and inverter A8-6. This circuit is enabled by the Drive Lockout described above.

When the Focus-Rotation system is powered up, the three-phase 208 VAC to the servo amplifiers must be delayed for several seconds to permit the R/D converters in the BDS3 to stabilize at the correct motor shaft angles; if the 208 power were to be applied immediately, the servo amplifiers would cause large drive excursions on the motors which is undesirable and could stress the FRM structure if the Focus drive is near a stop. The delay is implemented by control firmware as part of power-on initialization. The 208 VAC three-phase power is switched by a contactor in the Isolation Transformer Box which is driven by a high power solid state relay in the S103. Each controller can actuate can actuate the three phase solid state relay via gate EA44-8 and buffer A19-4.

When a limit switch is actuated, the S102 activates the YOWP line which inhibits all motor and brake drive via the Drive Lockout circuitry on Sheet 11. The S103 provides a sustained current drive to optical isolator A1-14 when a limit switch is not actuated. If a switch is actuated or the current path is broken the isolator drives and-gate A7-6 low. This A7-6 signal is the Drive Lockout, which, when high, enables the discrete brake and BDS3 commands to both axes. When Drive Lockout goes low it inhibits the brakes and motor drive BDS3's and clears the D/A register so that the analog drive to the BDS3's is zero volts. The other input to gate A7-6 is the cable interlock signal which senses that a cable on the bin I/O panel has been disconnected. Drive Lockout is sensed by EPROM PB4 which is set to input mode. Both YOWP and the cable interlock have 1 microfarad noise filters on the inverter inputs.

An Analog Devices AD2702 precision +10 and -10 Volts reference chip (CA44) provides these reference voltages to the A/D converters for converter alignment and as a monitor data readout to verify that the A/D converters are operating properly.

S101 Front Panel Control-Display Logic

The Front Panel Control-Display Logic consists of a dual axis numeric display driven by the 8156 RAM ports, the CMP-MAN switch, the dual-axis mode switches which are interfaced to the Buffered Address-Data bus and the toggle switches and discrete LED's on the S105 front panel.

The front panel MAN-CMP mode switch selects either local manual slew control via the switches on the S105 front panel or Antenna Computer control via the MCB. In the manual mode firmware associated with the S105 switches controls the FRM. Some firmware is common to both modes.

Two Hex front panel mode switches permit manual intervention in the program execution. The eight bits from these two switches are read by the BADX buses of both controllers; hence the switches jointly intervene in both Controller's programs. The two programs have an identical response to the switch settings. A Mode switch setting of 00 is the normal state; the switches should always be left at this setting except when manual intervention is required for maintenance purposes. A convenient application is to display the two controller A/D and D/A values on the front panel display. The switch settings and commanded actions are as shown on the next page.

S101 Mode Switch Settings and commanded intervention

00 = Normal operation mode.
01 = Both Brakes continuously released.
02 = Both D/A converters commanded to output - 5 volts to servo amplifiers.
03 = Both D/A converters commanded to output 0 volts to servo amplifiers.
04 = Both D/A converters commanded to output + 5 volts to servo amplifiers.
05 = Both A/D converters digital outputs with - 10 volt inputs.
06 = Both A/D converters digital outputs with 0 volt inputs.
07 = Both A/D converters digital outputs with + 10 volt inputs.
FF = The Focus program bypasses the second screw code. The Rotation program sets normal operation.

Codes 08 through EF are not assigned; these codes set normal operation.

The setting of 01 permits manual drive of the FRM with a wrench or other tools.

The front panel display shows 5 digits from each controller; the display contents may represent values or states depending upon the firmware program. Displayed values may be either decimal or an altered hex code with F displayed as a blank digit (i.e. no segments lit). The left 5 digits are Focus data and the right 5 digits are Rotation data from the two Control Microprocessors (Sheet 17). The displays can show a minus sign for negative values, the tens digit may be blanked and a decimal point may be displayed to indicate values with a resolution of 1/1000. Typical values which may be displayed are: 99.2539, -97.1849 (sign digit used), 7.359, - 8.361, 0.539, - 0.634 (sign and tens digit blanking). The sign and blanking functions are firmware programmed; the decimal point location is hardware logic-driven and is not programmable.

Numeric data for the front panel display is output on 20 of the 22 bits available in the three 8156 I/O ports of the two Control Microprocessors - a total of 40 bits to be converted to 10 display digits. The blanking and sign functions are driven by the 8156 PC5 (FBKI) and PC4 (FSIGN) bits. The multiplexing display logic sequentially scans and converts 4 bits at a time into a 7-segment code which drives a selected display digit. The logic performs two functions in synchronism: 4 bits are selected and encoded into a 7-segment code and the encoded value drives a digit selected by a digit selector. The logic recurrently scans the 40 bits in sequence to drive the 10 numeric digits in the display.

The scanning clock is 74LS14 oscillator A29-6. The logic is driven by the scanning counter U4, a 74LS161 binary counter. Four terms (SC1 .. SC8) from the counter sequence through 16 states. These four terms drive the digit select multiplexer (Sheet 13) and the cathode selector in tandem. The 4-bits are encoded into 7-segment drive states for the display chips. The 74LS49 (U4) encoder chip translates the 4-bits from the digit multiplexer (chips A5 .. A20 Sheet 13) to a 7-segment code which drives the anodes of the HP5082-7415 display chips. These 7 lines are common to the anodes of all three display chips. The digits are sequentially selected for display by the 7445 CMOS 4-to-10 line decoder (A18) which provides a low-true, current sinking drive to the cathodes of the display chips. Ten of these states (DS0 ..DS9) select the cathodes of the numeric digits; the remaining six states drive the decimal point and sign functions.

The digit select multiplexer chips (A5 .. A20, Sheet 13) are 74LS151 8:1 multiplexers which select four bits/digit for 8 of the 10 scan states. 74LS157 A25 selects the final two sets of bits to be multiplexed. The SC8 term from counter A23 enables the 8:1 multiplexers for the first 8 digits and enables the strobe input to the 157 for the last two states. The SC1 term on the 157 controls the selection of the last 8 bits for digits 9 and 10. Or-gates A30 accept the four bits from either the 8:1 or 2:1 multiplexers.

Referring to the Front Panel Display (Sheet 17), we note that each set of 5 value digits is prefixed by a sign digit and there are two blank digits between each numeric value to make the display easier to read. The LED display chips also have a decimal point anode which must be time-state selected in conjunction with the digit drive by the digit selector. The decimal point drive is restricted to the fourth digit; thus there are three digits below the decimal point. Blanking is restricted to the most significant (tens) digit.

The 7-segment outputs of the 74LS49 encoders are open-collector transistors which use the 470 ohm resistors as pull-ups; to illuminate a display segment, a decoder segment output line is high so that current may flow through the pull-up resistor to the segment anode, through the LED segment cathode to ground through the low-true digit selector output. If a segment is not to be illuminated, the decoder segment output is low (about 0.25 volts) which sinks the resistor current to logic ground through the digit selector (A18) outputs, DS0 ... DS9. The 0.25 volt, Vol level of the LS49 encoders is less than the forward voltage (about 1.7 volts) of the LED segment so no current can flow through the LED; it is sunk through the decoder chip output.

We will now consider the sign, blanking and decimal point driving logic on Sheet 16. 8156 Port C bits PC4 (FSIGN) and PC5 (FBKI) and scan terms SC1 .. SC8 drive this logic. As suggested by the name, FSIGN determines the sign of the Focus display and FBK1 blanks the first digit of the Focus display. The Focus and Rotation display decimal point anodes are driven by high true drive from inverters A19-12 and A19-14. The DS3bar and DS8bar terms which drive the inverters are from the digit selector so that anode current flows to the K3 (U1) and K5 (U2) cathodes only during states 3 and 8 of the scanning counter.

The negative sign digit (plus sign is implied) prefixes the five numeric digits in each display. The sign symbol is formed by driving only the "g" LED segment in this prefix digit. This is done during the 14 and 15 states of the scan counter. Gates A4-10 (14) and A4-6 (15) drive low-true and-gates A14-1 and A14-4. If either the FSIGN or RSIGN bits from the two 8156's are high true, the MINUS1bar or MINUS2bar terms from inverters A19-6 and A19-10 will be low true to current-sink drive the U1 K1 and U2 K3 cathodes during time states 15 and 14 respectively. The states 15 and 14 drive to the "g" segment (signal DSPSEG) is orr-ed in gates A13-10 and A13-13 with the "g" segment drive for the ten digit-driving states. On the display PCB the "g" segment drive from the 7-segment encoder does not directly drive the "g" anodes in the displays; it is fed back to the A13-10 and A13-13 or gates to be merged with the sign (states 15 and 14) drive of DSPSEG.

Blanking of the tens digit is controlled by the FBK1 and RBK1 terms from the 8156 ports. Blanking of the digits (i.e. inhibiting drive to all display LED segments) occurs when the BI input of the 7-segment encoder U4 (74LS49) is driven low. Since the display uses multiplexing logic this blanking must be keyed to the time states. Gates A14-15 and A14-12 are high only during the 4 and 9 states if either the FBK1 or RBK1 terms are low true.

The S105 functions as a manual control-display panel. Manual slew switches on S105 permit either axis to be driven when the S101 Manual-Computer switch is in the Man position. The state of the four slew switches (two for Focus and two for Rotation) are input to Ports PB6 and PB7 of EPROM's FD22 and DD22. When the S101 switch is in the Man position, the operating firmware in each Control Microprocessor samples the state of these two switches and causes the drives to move in the direction specified by the switch.

The state of the S101 front panel Manual-Computer switch and Drive Lockout are sensed by Ports PB5 and PB4 on EPROM's FD22 and DD22. The operating firmware samples the states of these two important discretes to determine control action.

S101 Command-Monitor Interface

The Command-Monitor interface is the INTDX bus from the S104 which is interfaced to the Buffered Address-Data Bus. Command bytes to the 80188 control microprocessor from the Antenna Control Computer are input from the S104 via the INTDX bus to the Buffered Address-Data Bus where they are imput to the 80188 via the 80188 DMA channel 0; monitor data to the Antenna Control Computer is output by 80188 direct I/O to the S104 via the Buffered Address-Data Bus and INTDX bus. Bus speed requirements dictated that the Buffered Address-Data Bus should not go out of the S101; the INTDX bus is used to isolate the controller's Buffered Address-Data bus from the relatively long lines between the S101 and S104.

At this juncture, the reader is urged to take a quick look at Sheet 2 of the S104 logic schematic; this look will enable identification of the interface signals. After completing this discussion of the command-monitor data interactions with the S104, the reader should study the S104 logic operations of interacting with the MCB and S101.

The logic of command input is as follows: When a command from the Antenna Control Computer sends a command to the F-R System, the VLBA Standard Interface and associated logic raise (high) the DMA Control line to the address-designated Control Microprocessor via the 80188 DRQ0 input. The 80188 enters a four-word DMA transfer sequence. The 80188 activates Peripheral Chip Select 3 (PCS3bar) which, and-ed with the Focus Enable (from S104) in gate FC43-6 enables the INTDX bus onto the Focus Buffered Address-Data Bus. As the Focus 80188 DMA Controller sequentially inputs the command data bytes, it increments the 80188 I/O address to generate the F395, F396, F397 and F398 enables which drive the 74LS245 bus driver (EC01) directiion input so that the 245 drives the INTDX inputs onto the Buffered Address-Data Bus. The data input by F398 is ignored; F398 is used to cause S104 to send a Device Acknowledge signal to the Standard Interface which terminates the command input sequence.

A request for monitor data from the S104 is initiated by the Focus Data Control term from the S104 which drives the 80188 Non-Maskable Interrupt (NMI). The interrupt response firmware then causes a direct input of the Relative Address byte by activating the FCS3bar signal and the F395 enable. This enables the Relative Address register contents in the S104 to be impressed onto the INTDX bus and hence onto the Buffered Address-Data bus in the same manner as was done for the command transfers described above. After decoding the Relative Address in firmware, the 80188 sequentially outputs the two bytes of monitor data onto the Buffered Addressed Data bus where it is passed through the 245 INTDX bus driver back to the S104. The F400 (Foc Mon Data MSB) and F399 (Foc Mon Data LSB) enables strobe the two bytes into the 74LS374 storage registers in S104. The 80188 issues an F398 enable (Focus Data Acknowledge) which completes the transfer sequence and causes S104 to output the data on the MCB.

3.2 S102, Apex Interface Logic Description

The S102, Apex Interface is the module which senses Apex discretes, converts the Focus and Rotation resolver signals to digital values, multiplexes and converts apex-related analog signals to digital values and outputs these data as a serial data stream upon request from the S101. When any of the 8 limit switches is activated, the limit alarm YOWP goes true which inhibits motor drive in both axes.

Drawings D55007S006 and D55007A003 are the S102 Logic Schematic and Assembly drawings respectively.

Like the S101, the S102 is partitioned into two nearly identical, independent sets of interface circuits, each separately queried by the associated Focus or Rotation controller logic in S101. The chip layouts on the logic connector boards are also nearly identical. The differences for the two axes are in the Resolver to Digital (R/D) converters. The S102 is powered by a triple-output power supply (in S105) to isolate S102 from the other modules.

A Harris HI-5901 analog multiplexer-Sample/Hold and HI-574A A/D converter (Sheet 6) converts analog signals: R/D converter velocity, power supply voltages, precision +/- 10 volt references and bin and mount temperatures. (Mount temperatures are not implemented in the FRM). The A/D converter conversion is synchronized to the data readout sequence and the data is injected into the digital data stream. 100 microsecond time-constant RC filters (Sheet 6) reduce noise on these analog signals.

A momentary action switch, S2 (behind the front panel access cover) enables the input of +10 reference, analog ground and -10 volt reference into the A/D to enable adjustment of the A/D gain and zero offset pots R5, R6 and R7. These adjustments can be made with an Antenna Control Computer to view the converted values for inputs of +10, Ground and -10 volts. A second (at this time not implemented) method is to set the S101 front panel mode switch to a certain setting which will cause the firmware to display the converted value on the S101 front panel display.

The limit switch circuitry senses current through the NC-C limit switch contacts and 470 ohm pull-up resistors to +5 volts (Sheets 4 and 5). When a switch is actuated (or a cable disconnected etc.) the switch current path is interrupted and gate B25-10 (Sheet 6) output goes high true (active) which routes YOWP to S101 through momentary action toggle switch S3. When S3 (behind the S102 front panel access cover) is actuated YOWP goes false which permits manual drive out of the limits. The limit switches have a 0.47 microsecond time constant filter (Sheets 4 and 5) on the switch outputs. The limit switch states are also read out in the digital data stream to the S101 via shift registers E20...E22 (Sheet 4) and F20...F22 (Sheet 5). The limit switch discretes are sampled when the S101 requests an Apex data readout sequence.

The front panel discrete display circuitry shows the state of all Apex discretes and the activity of the A/D converter, Focus and Rotation readout circuitry. When these three indicators are lit the three sets of circuits are active. From left to right these LED indicators are: LL1, LL2, UL1, UL2, FACTV, GND, GND, A/D-ACTV, M1, M2, M4, M8, GND, GND, GND, RACTV, CW1, CW2, CCW1 and CCW2. FACTV and RACTV denote Focus active and Rotation active respectively and A/D-ACTV denotes that the A/D converter is active. M1,..M8 are the analog multiplexer address bits.

The front panel numeric display shows the hex values of Focus and Rotation position. These displays are convenient for adjustments of the FRM position readout resolvers. Since the Focus R/D is a 14 bit converter the Focus LSB display will change in increments of 4 counts, e.g. the Hex values 0,4,8,C. The Rotation axis uses dual readout (i.e. 1:32 ratio coarse and fine) resolvers and associated 14-bit coarse and fine R/D converters whose output is fed into a combiner which generates a 16 bit

composite value. The three resolvers must be mechanically adjusted so that the resolver zeros (i.e. zero counts from the R/D converter) are at the low end of the motion range.

With switch S1 (behind the front panel access cover) in the center position, the display shows the composite Rotation position from the combiner; when S1 is set to either of the other two positions the display shows either the coarse or fine resolver positions. When aligning the Rotation resolvers, the coarse resolver should be adjusted first to produce a zero count when the ring is at zero degrees degrees. The Rotation Fine resolver (which rotates 32 times relative to the coarse) should be adjusted so that the fine resolver zero coincides with the coarse resolver zero and at 11.25 degree intervals on the coarse resolver. These adjustments are easily accomplished using the S102 display.

Sheet 8 shows the numeric display logic. The display digits use a multiplexed drive which is clocked by the time base on Sheet 1. A 10 khz term (80 microseconds) from counter term B19-11 drives binary counter D29 on Sheet 8. The "2","4" and "8" states of D29 serve as a 3-bit address for the 74LS151 eight-to-one multiplexers D8...D23. Each of the four 8:1 multiplexers select a bit to form a hex character on the display. The four hex bits from the Focus Msb are selected during state "0" of the address, the next four hex bits are selected during state "1" etc. This sequence continues through the four Focus and four Rotation digits.

The display chips require a 7-bit ASCII code input so it is necessary to convert the four bit hex value to the ASCII code. Chips D19 (a 74LS85 magnitude comparator), D24 (74LS157 a quad 2:1 multiplexer), D25 and D26 (74LS283 4-bit full adders) form these codes. The 74LS85 is produces a high A>B output when the value of the A_0,A_1 and A_2 inputs from the 8:1 multiplexer is greater than 9. The four 8:1 multiplexer outputs are connected to the A_1,A_2,A_3 and A_4 inputs of the first adder, D25. For hex values of 0 through 9, the multiplexer data passes through the adder without alteration (i.e. addition). For hex values greater than 9, an additional 7 counts (1,2 and 4 from the LS157 plus the adder C_0) is added to the hex value; the result is the lower 4 bits of the ASCII code. The three upper ASCII bits are 011 for ASCII codes of 0 to 9 and 100 for codes A through F. The second adder (D30) generates the 100 code when the first adder C_4 is high and 100 when it is low. The reader should refer to the 74LS283 truth table to augment this description.

The numeric display uses two HPDL-2416 four-digit display chips which use a three bit digit-select code and a write strobe. The three upper bits of D29 are the address bits and the lsb is used as a write strobe. Counter D29 address terms T_0 and T_1 sequentially address the four display digits of both displays; the third term, T_2, selects the Focus or Rotation display. This strobe is the "1" term from counter D29-14 which is enabled by the T_2 term from counter D29. T_2 is high for the upper four states of D29 T_0, T_1 and T_2. The D11-14 inverter and D20 74LS32 generate the WR1 and WR2 terms to sequentially load the ASCII code terms (DS0 ... DS6) into the eight display digits.

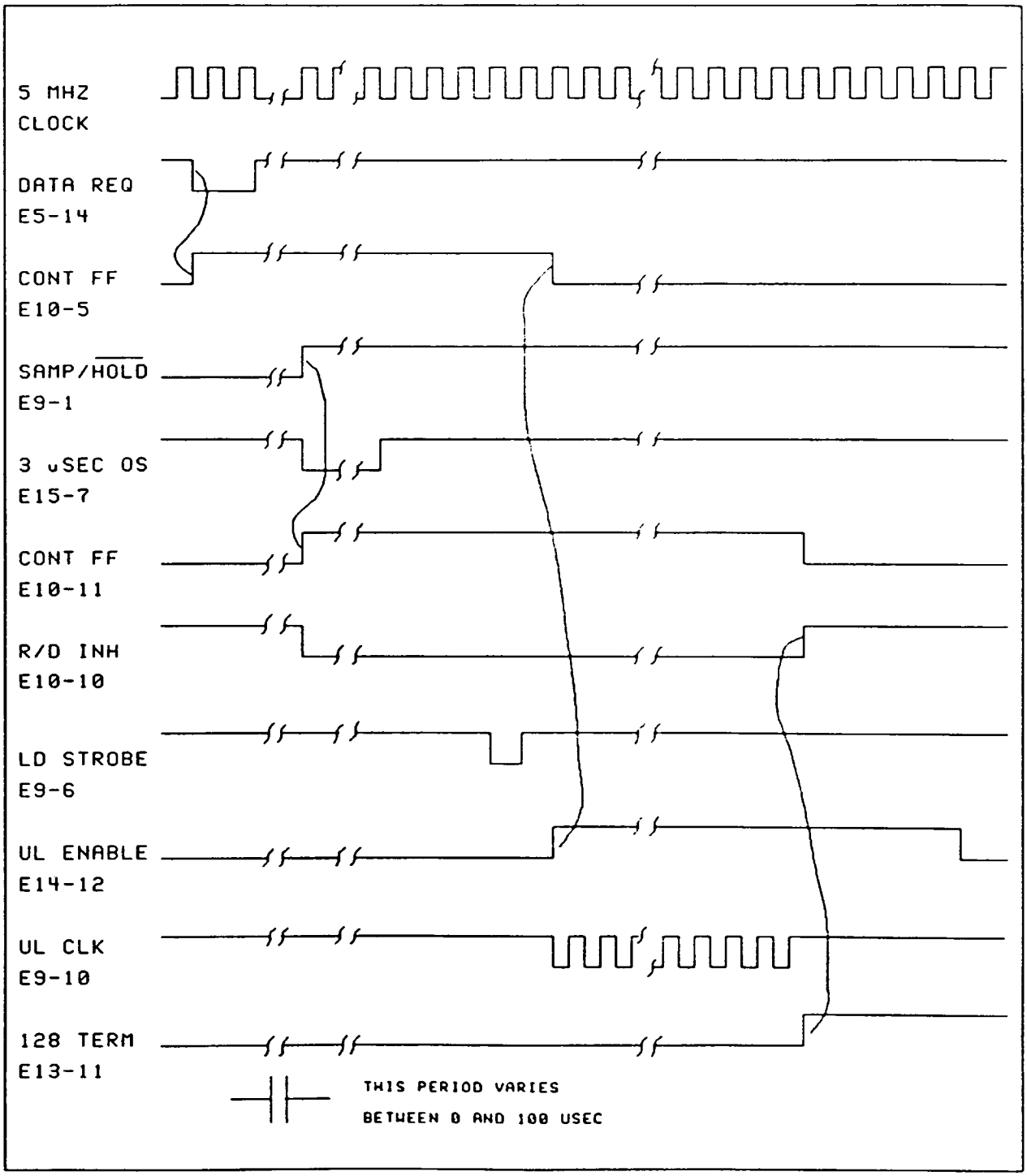
We now consider the selection of the three Rotation R/D converter values for the Rotation position display. The composite 16-bit value from the Coarse-Fine combiner is usually displayed since it is the one that the S101 uses in controlling Rotation position; however, the coarse and fine R/D values must also must be read when aligning the coarse and fine resolvers in the motor package. The 3:1 manual switch-controlled multiplexers D2...D27 perform this function under the control of S1 which is located behind the front panel access cover. S1 is normally left in the center position but may be set to the Coarse Select (upper position) or Fine Select (lower position). The 74LS153 dual 4:1 multiplexers serve as 3:1 multiplexers with 2-bit "A", "B" address states of 10 (C1 select, Coarse), 11 (C3 select, composite) and 01 (C2 select, Fine). The Rotation position output from this bank of multiplexers are fed to the Rotation inputs to the display control and ASCII conversion multiplexers described above.

We next consider the time base and A/D converter sequencing logic (see Sheet 1). A 10 Mhz crystal with a TTL output drives the synchronous counter B9...B19. The low-true 10 state in the 100

Khz to 10 Khz counter B19 is decoded by B18-6 to direct-set flip-flop B23-5 which sets the A/D converter Multiplexer-Sample/Hold chip to the Hold state. 100 microseconds later, the trailing edge of the 10bar term clocks flip-flop B23-11 which starts the A/D conversion sequence. The leading edge of the "80" (60 microseconds after the trailing edge of 10bar) clocks flip-flop B23-5 back to the reset state. The low-true A/D Converter end-of-conversion pulse clocks one-shot B22-6 to generate a 1 microsecond delay. At the end of the delay the trailing edge triggers one-shot B22-10 to generate a 500 ns A/D load strobe to parallel load the A/D data and associated multiplexer addresses into the A/D-mux address storage registers D1..D5 for subsequent readout to the Focus and Rotation data readout registers. The trailing edge of the A/D data load strobe advances the multiplexer address counter B24. M1..M8 terms from this counter drive the HI-5901 Multiplexer-Sample/Hold chip on Sheet 6. The 10 Khz analog data multiplexing and A/D conversion sequence described above proceeds continually and asynchronously with the S101 Apex data readout requests.

We now consider the data readout sequencing and unload logic which is identical for the two axes. Because the logic is identical in the two axes we will discuss only the Focus readout control logic on Sheet 2. Data readout control involves synchronizing S101 data requests with the A/D and R/D converters so that the data is stable when sampled. S101 data requests may occur at any time asynchronous with the S102 time base. Optical isolator E5-14 goes low for X microseconds when S101 requests data. This direct sets flip-flop E10-05. When Sample/Holdbar goes high (for 10 microseconds) it signifies that stable data is available in the A/D storage register D1, D3 & D5. Gate E9-3 triggers one-shot E15-7 which generates a 3 microsecond low-true pulse which starts the readout sequence by setting control flip-flop D10-11 and lifts the clear on shift register E14. The R/D converter is inhibited from further conversions by the low-true inhibit from E10-10, (if the converter is in the process of making a conversion when the inhibit goes low it will complete the conversion but will not begin another conversion until the inhibit goes high). The A&B inputs of E14 are high; this permits the 5 Mhz clock to shift a sequence of 1's through the register. This shifting will start with the next rising 5 Mhz clock edge after D10-11 is set. 400 to 600 ns later, (depending upon the time D10-11 goes high relative to the clock) gate E9-6 generates a 200 ns load strobe which parallel-loads the R/D converter data, the R/D converter velocity, A/D Data, A/D Mux address and discretes such as the limit switch status into the 7-byte Focus data readout register F20..F22. When the first 1 loaded into shift register E14 reaches Q4 (at 800 ns) the Q4 rising edge clocks flip-flop D10-5 reset so that it can respond to the next S101 data request and permits the data shift clocks to be output via gate E9-10. When Q4 rises it stops the pre-set loading of binary counter E8, E13 so that the counter can begin to count out the data unload clocks. The counter was preset to a count of 77; at the count of 128, E13-11 rises and the rising edge clocks flip-flop E10-11 to the reset state which causes shift register E14 to load 0's. When the first 0 reaches Q4 the shift clocks are inhibited. The total number of counts is 128 - 77 in the counter + 5 shifts in E10 & E14 or 56 clock pulses to shift out seven 8-bit shift registers. Buffer E12-3 drives the the shift register and differential driver E17 buffers shift clocks to S101. The four E7 inverters provide about 60 ns delay for the S101 shift clocks so that S101 samples the data well past the rise. The timing for this logic is depicted on the next page.

The R/D converter velocity outputs are a measure of FRM mechanism velocity; the response of the S101 velocity command to the servo amplifiers. The Focus and Rotation (fine) R/D converter velocity outputs are multiplexed and converted by the Multiplexer-Sample/Hold-A/D Converter as described above but the digital velocity values are stored in registers which are sampled and read out when the S101 requests Apex data. The reason for the register storage is to make them immediately available to S101 so that it does not have to do a time-wasting search through a data buffer for the data by decoding the mux addresses associated with the converted Apex analog data (remember that the multiplexing and A/D conversion are not synchronous with the S101). The Focus R/D velocity signal is connected to the multiplexer via the 100 microsecond time-constant RC filter CA10. The Rotation (fine) velocity signal is buffered by differential amplifier B2-10 and fed to the multiplexer via 100 microsecond TC filter CA01. The filters reduce R/D converter clock noise on the velocity signals. The converted velocity signals are



S102 APEX DATA UNLOAD TIMING

stored in registers E21..E26 (Rotation) and F21..F26 (Focus) by the low-true A/D load strobe from B22-10 which drives the G2 inputs of 74LS138 decoder B17. The trailing edge of the strobe from the decoder outputs load the two velocity registers. The strobes are output on Y3, (true for state 3 of M1,M2,M3 & M8bar, Rotation velocity is connected to input 4 of the multiplexer), and Y7, (true for state 7 of these

address bits, Focus velocity is connected to input 7 of the multiplexer).

An important Focus drive component is an inductive proximity sensor, the 2nd Screw Position Sensor which produces a TTL level output as a function of the position of the associated 10-tooth spur gear. The gap is adjusted to produce about a 50% duty cycle signal when the Focus drive screw is rotated. There are thus 20 states of the 2nd screw sensor and 5.649 Focus position counts per state. The 2nd screw sensor status is sampled each time the Focus controller in S101 polls the Focus Apex data readout circuitry in S102. Sheet 5 depicts the 2nd screw sensor interface circuitry. The sensor drives a 74LS14 Schmidt inverter. The sensor output is a clean, glitch-free TTL pulse; the hysteresis provided by the 74LS14 insures that there will not be spurious glitch outputs for any position of the gear teeth. The state of the 74LS14 output is input to the Focus discrete data readout register F20-4.

The Focus R/D converter is manufactured by DDS and the Rotation R/D converters and combiner are manufactured by Natel. Data sheets in Volume II describe these devices theory of operation. The reader is urged to carefully study these data sheets because these converters are very important devices in the F-R System. For further reading, the Synchro and Resolver Conversion Handbook (Sept 1980), published by Analog Devices is an excellent reference on the operation of resolvers, synchros and R/D and S/D converters.

3.3 S104, F-R Interface Description

S104, the F-R Interface performs six functions: it interfaces the F-R Control System to the Antenna Control Computer via the Monitor and Control Bus; it transfers 3-byte address-command arguments from the Standard Interface to the S101 F-R Controllers; it transfers 2-byte monitor data arguments from the S101 controllers to the Standard Interface; it decodes Master Reset commands to reset the 80188 processors in the F-R Controllers; it performs analog signal multiplexing and A/D conversion for monitor data and samples servo amplifier status and fault discretes which are passed to the F-R controllers in S101.

Drawing D55007S003 is the S104 Schematic Diagram and D55001A005 is the Assembly drawing.

S104 communicates with the Antenna Control Computer via the party-line serial Monitor and Control Bus (MCB) which conveys computer commands and monitor data polling requests. The S104 contains a VLBA Standard Interface Board for connection to the MCB. Logic in S104 decodes command and monitor data addresses from the command message stream on the MCB XMIT (Transmit) lines; if the addresses correspond to F-R System addresses, the interface board inputs addresses and arguments to the S101 80188 processors via the DMA channels. Monitor data arguments are output from the 80188 via a Direct I/O channel to the VLBA Standard Interface board which formats and outputs the data on the MCB RCV (Receive) lines.

The 80188 uses two types of I/O operations: Direct I/O in which the program instructions interchange data between CPU registers and external devices, and DMA I/O in which, after initialization, data is input or output direct to memory without involvement of the CPU. At the end of a DMA sequence a programmed transfer count terminates the transfer sequence. F-R Controller commands are input to the 80188 via DMA channel 0 and monitor data commands are signalled by interrupt-driven Direct I/O.

The VLBA Standard Interface Board is a removable PC board which functions as a general-purpose MCB interface which is augmented by custom-designed decode logic to complete an interface system. The VLBA Standard Interface Board is described by Specification A55001N002. The MCB protocol is defined by specification A55001N001. Since these specifications have been previously published, the operation of the interface board will not be discussed but the logic to interface to the Standard Interface board is described.

Interface board signals which are important to the S104 logic are the 8-bit Relative Address, RA0..RA7, the 16-bit Cmd/Mon argument bus (Cmd/Mon0..Cmd/Mon15) and the interactive hand-shaking interface terms: Device Request, Device Acknowledge, Read/Writebar, Analog Enable, Hi/Lo Select and ID Code request. The RA0..RA7 is an output (only) bus but the argument bus is a tri-state bi-directional bus which outputs a command argument or inputs a monitor data argument. The Relative address designates a specific command or data channel. When Device Request is true (high) it signifies that the device controlled by the interface board (e.g. the F-R System) must decode the address and take appropriate action. The type of action to be taken is determined by Read/Writebar and the Relative Address. When Read/Writebar is low it designates that the Interface board has impressed a command argument on the bus which the device must accept; when high it designates that the device must impress a monitor data argument upon the bus.

In Standard Interface terminology, the external logic which interacts with the VLBA Standard Interface board is called "device" logic.

When Device Acknowledge is raised high, the controlled device signifies that it has completed the interface operation and has either accepted the command argument on the Cmd/Mon bus or has

impressed a monitor data argument on the bus.

Analog Enable results from a device decode of the Relative Address logic which designates that analog data is to be multiplexed and converted by the interface board. Control logic on the board will impress the A/D Converter output on the Cmd/Mon bus, the device logic must not impress digital data on this bus when the addresses designate multiplexing and conversion of analog data. Analog multiplexers external to the interface board may be used to input analog data to one of the interface board analog input channels. Hi/Lo Select is generated by a device decode of the Relative Address; when Hi/Lo Select is low it causes the interface board multiplexers to respond to the least significant address bits (RA0, RA1 & RA2); when high, the board multiplexers will respond to RA3, RA4 and RA5. When external (i.e. to the interface board) analog multiplexers are used, the Hi/Lo request line must be set so that the on-board multiplexer selects the output of the external multiplexer.

Since the MCB is a party-line bus, it conveys all commands and monitor data polling requests from the antenna control computer. These messages contain an M&C address which designate device command and data channel addresses serviced by the Standard Interface Boards and other M&C interfaces in the system. The Standard Interface Board in the S104 (as well all the other Standard Interface boards and M&C interfaces in the system) must have some means of identifying the addresses of commands and monitor data polling requests to which it must respond; messages with addresses outside this range should be ignored. M&C addresses for a given interface are a block of addresses within this address space. The lower 8-bits of the M&C address are the Relative Address which is output via the RA0..RA7 bus. These blocks are unique to a device. Standard Interface board address selection is accomplished by a unique ID code. The ID code and firmware in the board's microcontroller determine the M&C addresses to which the board should respond. The M&C address space covers a range of 0 to 32768; the interface address blocks are all contained in this address range. For a given interface application, the address range may be as small as 16 addresses or as large as 256 command and 256 monitor data addresses to external devices. The user must remember that the Standard Interface Board has 16 internal addresses which are both command and monitor functions. These addresses and functions are listed in Section 3.9. The ID code is hard-wired in S104 and is tri-stated onto the Cmd/Mon bus by the ID Request signal from the board. The interface board reads the ID code (7 bits of data + an odd parity bit) during power-on initialization and periodically thereafter. An interface address space is defined by two parameters stored in the lowest 256 bytes of the control firmware, a start address and block size. The ID code is the pointer to these two locations. When the ID code is read by the microcontroller, the code value is doubled; this produces the address whose contents define the size of the M&C address block. The doubled value + 1 is the address whose contents define the start address of the M&C address block. This table of addresses in the lower 256 byte of program memory can be over-written by the antenna control computer so that all M&C addresses may be re-defined by the computer.

The F-R Control System ID code is 72H. The board M&C block size is defined by the contents of address E4 and the start of the address block is defined by the contents of address E5.

The address decode and interfacing logic is depicted on Sheets 2 and 4. The logic decodes addresses and Read/Writebar to determine which processor is being commanded and the type of action to be taken. Device Request sets flip-flop AG33-5 which is the Interface Device Request signal. This signal enables the 74LS138 address decoder BE42. Terms from this decoder and Read/Writebar determine the action to be taken in response to Standard Interface states.

Relative Address X8H (RA3) sets Hi/Lo Select low when true (high). This term controls the analog multiplexer address on the Standard Interface board.

Address 5XH and Write/Readbar are and-ed (in BC3408) to drive the low-true Master Reset line; this line also resets the 80188 processors in S101.

Addresses 1XH and 2XH are or-ed (in BC34-6) to designate a Focus controller interaction.

Addresses 3XH and 4XH are or-ed (in BC34-3) to designate a Rotation controller interaction.

When Write/Readbar is true (high) and either a Focus Enable or Rotation Enable is true (high), a processor DMA sequence is initiated to transfer the Relative Address and Command argument to the addressed 80188 processor. BA33-3 and BA33-6 perform this function.

When Read/Writebar is true (high) and either a Focus Enable or Rotation Enable is true (high), the Focus or Rotation Data Control is set high true which initiates an interrupt-driven two-byte transfer of monitor data from the addressed processor to the S104 data registers. Gates BA33-11 and BA33-8 perform this function.

When either Focus Enable or Rotation Enable is true (low), the Monitor Control is true. Nor gate AG17-10 performs this function.

Note that the decode logic provides 32 possible command addresses and 32 possible monitor data addresses for each axis. The two controllers analyze the RA0..RA7 address to determine the specific command action to take (i.e. position or nap commands or position, position error, command echo etc. arguments to be output as monitor data).

Sheet 4 of the schematic diagram depicts the 8-bit Register-Bus logic by which the S104 interchanges address and argument values between the S104 interface board and the Focus and Rotation Controllers in the S101. The 8-bit, bi-directional tri-state INTDX bus conveys these arguments between S101 and S104. 74LS245 bus transceivers in the S101 enable the INTDX bus to drive either the Focus or Rotation Buffered Address-Data busses in the two controllers. There are five data source/sinks which are enabled onto the INTD0..INTD7 bus to the S101.

The rising edge of Device Request clocks 74LS374's BE01,BE12 and BE23 to store the relative address and command argument which the S101 must read. The address and command argument bytes are enabled onto the INTDX bus by strobes from either controller in the S101. F395 and R395 enable the Relative address on to the bus; F396 and R396 enable the command argument lsb onto the bus and F397 and R397 enable the command argument msb onto the bus.

Either of the F-R Controllers may clock the monitor data lsbyte on the INTDX bus into BG12 74LS374 latch by F399 or R399. The msbyte of monitor data from the INTDX bus is clocked into the BG01 latch by F00 or R400. Latches BG01 and BG12 are subsequently input to the interface board via the Cmd/Mon bus.

The ID code (F2) and Module S/N# code (00, not implemented, the inputs to BG23 are wired to ground) are enabled onto the Cmd/Mon bus by the ID Request line from the interface board (P5-9). A sequence of load strobes and write/read operations is executed by firmware in the S101 F-R Controllers to implement transfers on the INTDX bus. This sequence is treated in Sections 3.7 and 3.8.

Flip-flop AG33 is held reset by low-true Device Request when it is not active (i.e. between Standard Interface interactions with the device logic). When it is necessary to enable the two monitor data registers BG01 and BG12 onto the Cmd/Mon bus, gate BC34-11 sets flip-flop AG33-9 to enable the register data onto the bus. Flip-flop AG33-9 is reset when Device Request returns to the low state at the completion of a monitor data request from the Standard Interface. The disconnect is initiated by the monitor-data-active 80188 processor which issues a Data Acknowledge strobe (F398 or R398) to the (low true) and-or gates AG17-1, 4 and 13 which clock flip-flop AG33-5 reset. When this flip-flop goes reset the

Interface Device Request goes false (via AG09-6) and Device Acknowledge goes high (and stays high until the next command, the interface is not disturbed by the sustained high state).

The S104 contains analog and digital circuitry to monitor the F-R system and IDD servo amplifier power supply voltages and the servo amplifier fault and status discretes. The analog signals that are monitored are the system +/- 15, +5 power supplies and the servo amplifier chassis +/- 12 volt power supplies for the three BDS3 amplifiers and the RDP2 anti-backlash controller.

We will now describe the analog multiplexing and BDS3 fault-status logic. Sheet 5 of the Schematic Diagram depicts the analog multiplexers and analog divider-filter circuitry. The analog monitoring is performed by an 8-channel HI-508A single-ended analog multiplexer which is used in conjunction with the interface board analog multiplexers. Voltage divider-low pass filter circuits on dip headers AA01 - AA37 divide the analog voltages with magnitudes greater than +/-10 volts by a factor of 2. The filter capacitors and 5.1 Kohm input resistors have a time-constant of 0.5 ms to reduce high frequency noise on the analog signals. The HI-508A analog multiplexer single-ended output is connected to the Standard Interface Board differential input Analog 0. The low side of Analog 0 is connected to the S104 analog ground. Analog inputs 2, 3, 4 low side inputs are also connected to analog ground. Analog inputs 5, 6 and 7 are not used so both high and low inputs are grounded. Relative Address bits RA0..RA2 control analog channel selection in the remote (to the Interface board) HI-508A multiplexer. When the remote multiplexer is being used to select and convert an analog signal the on-board multiplexers is controlled by the RA3, RA4 and RA5 bits to select the output of the remote multiplexer. When analog inputs are directly input to the interface board analog channels the RA0, RA1 and RA2 bits should control the on-board channel selection so that addresses associated with these channels may be contiguous. Hi/Lo Select controls the selection of the Relative Address bits for the on-board multiplexer.

The analog data multiplexed and converted by the S104 multiplexers and the Standard Interface are the S105 power supply and servo amplifiers power supply voltages. This data is read out to the Antenna Control Computer as monitor data and is indicative of the F-R Control System power environment. This data is not available to the F-R Controller. Section 3.9 lists the address, value and range of these parameters.

Sheet 3 of the schematic diagram depicts the interface circuits to read the low-true IDD BDS3 servo amplifier fault and status discretes. These digital lines are driven by CMOS 4049CMOS-to-TTL interface chips in the BDS3 MC2 boards. A low-pass filter and pull-up resistor reduce noise on these signals. The BDS3 drives the motors with a high power 4 Khz chopped drive and the MC2 board generates a 2600 Hz resolver excitation; although the cables to the Servo Amplifier Chassis uses shielded wire and the high power motor drive lines are dressed away from the signal cabling in the chassis, some noise pickup is possible; this is the reason for the filters. The discrete filters have a time constant of 1 millisecond.

The high-true BDS3 discretes are inverted by 74HCT04 inverters and sampled by 74LS373 edge-triggered latches. The 373 output enable and load enables are driven by low-true strobes from the Focus and Rotation controllers in S101 (see the description of the S101 strobe logic in Section 3.1 above). When the 373 load enable is high (the normal state of the strobe), the 373 Q outputs follow the D inputs. When the S101 sets the strobe low to read the discretes, the 373 latch states remain static at the state before the strobe dropped low. The low true output enable permits the 373 tri-state output buffers to drive the FBDF0..FBDF7 and RBF0..RBFD7 bus lines to the controller. Since there are two Rotation BDS3 Servo Amplifiers, the two sets of 373 outputs are tri-stated onto the Rotation BDS3 bus; the Rotation discretes are read by two strobes: R407 and R408.

3.4 S103, The F-R Switching Module

The F-R Switching module performs the functions of controlling AC power to the Focus and Rotation motor brake power supplies and the servo amplifier power contactor, measuring the brake voltages and currents and generating the 400 Hz resolver excitation signal. The S103 was designed to package these bulky power switching and power supply components which use voltages which are lethal to digital logic. Front-panel indicator fuses protect the brake power supplies, the 400 Hz excitor power supply and FRM heaters (which are not powered by S103).

Drawing D55007S002 is the S103 Schematic Diagram and D55007A004 is the Assembly Drawing.

The S103 components are mounted on both sides of a "U"-shaped chassis fastened to the module rails - see the assembly drawing for package details.

Low power Sigma 226RE-1 solid state relays switch 110 Volts AC power for the two brake power supplies. A high power Teledyne 611-2 Solid State relay switches 110 volt AC power for the FRM heaters which are not powered by this circuit. A current transformer on the brake Voltage-Current monitor board monitors heater current. Space for an additional Sigma 226RE-1 and Teledyne 611-2 has been provided in the event that additional AC power switching might be required in the F-R System. An unused hole for an MS3102 size 14 connector is available on the rear panel for this future eventuality. The solid-state relays are driven by the S101 F-R Control module.

The MCS-801-1 brake power supplies are manufactured by Warner (the brake supply schematic is shown on the S103 schematic diagram). Note that the power supply is a full-wave bridge and there is no isolation transformer on the 110 volt AC lines; the brake supply DC output lines should not be grounded. The Brake supply contains an internal fuse (disassembly required for replacement) and a protective MOV across the AC input. A 100 volt zenar and 1N4004 diodes absorb the energy stored in the brake inductance when the brakes are de-energized. The brake power supplies are plugged into a phenolic octal socket.

The resolver excitor is a removable PC board mounted on the chassis. Drawing C55007S017 is the 400 Hz excitor PCB schematic diagram. The excitor is powered by a single output DC power supply on the board; a step-down power transformer is mounted on the chassis. The 400 Hz oscillator is a "Bridged Tee" oscillator using a single supply LM3900 amplifier. The phase shift produced by the 180 K-ohm resistors and 1000 pf capacitors; feedback network produces 180 degrees of phase shift at the oscillation frequency. The frequency is given approximately by: $F = 1/2\pi(RC)^{1/2}$. The 20 Kohm pot trims the frequency. The 1N5231 provides non-linearity which RC oscillators require. The 50 K ohm pot adjusts the drive level to the LM378 bridge amplifiers which provide an AC-coupled drive to the power amplifiers. The power amplifiers are an "H" configuration Class "A", push-pull driver using complementary power transistors which drive a custom-built 400 Hz transformer, part # PC4.1495 from the PC Transformer Corp of New York City. The excitor provides a 26 Volt (RMS, 37 volt peak) drive to the three resolvers whose parallel impedance is about 100 ohms, mostly inductive reactance. Since the 400 Hz resolver excitation comes off the transformer secondary it is also ground-isolated from the rest of the F-R system. The green LED provides an indication that the excitor power is present, a consideration when working on the exposed module. The excitor Vcc power is brought out through a 1500 ohm resistor to drive a front panel LED.

The brake voltage-current monitor is a removable PC board mounted on the chassis. Drawing D55007S008 is the PC board schematic diagram. An optical isolator and limiting resistor senses the brake voltage; the isolator output is a low-true TTL level. A second optical isolator connected across a resistor senses the brake current. The optical isolator outputs drive a 74LS32 logic gate to form a brake

Voltage*Current discrete which is sensed by the S101 F-R Controller. Since optical isolators are used to sense these parameters the brake power sensing circuitry is ground-isolated from the balance of the F-R Control System. Drawing D55007S008 is the brake voltage-current monitor PCB schematic.

Front panel LED's display the presence of the two brake and 400 Hz excitor voltages. Front panel test points permit measurement of the brake voltages and the excitor output. The 25 pin "D" connector permits measurement of the TTL logic drive to the SSR's and the excitor board Vcc.

3.5 S105, The F-R Power Supply

The S105 F-R Power supply provides system DC voltages, switches system 110 Volt AC power and serves as a local F-R System manual control and display panel.

Drawing D55007S001 is the Schematic Diagram and D55007A006 is the Assembly Drawing.

A Lambda LRS-53-5 switching power supply (PS1) provides system 5 volt DC power. A Lambda LND-X-152 switching power supply (PS-2) provides system +/- 15 volt power. A Power General 326A modular (i.e. plug-in replaceable) switching power supply (PS-3) provides 5 Volt and +/- 15 volt DC power for the S103 Apex Interface module. The Apex DC power is ground-isolated from the system 5 and +/- 15 volt supplies.

Switch S1, the front panel power switch has four sections which switch the AC hot lines for the system and S103 DC power, 400 Hz excitor (in S103), heater (not used) and the bin fans (not used). An RFI filter (FL1) provides some RFI isolation of the F-R power supplies from the antenna AC power.

Four momentary (down) SPST switches permit manual slew control of the two drive axes when the F-R Controller (S101) mode control switch is in the Man (manual) position.

A front-panel PC board contains LED drivers and discrete LED's to display S101 program execution status; these displays are useful in both the computer and manual control modes in that they indicate important system discretes. The F-R Controller drives the LED display circuitry from an 8755 EPROM I/O port. System states displayed are: CMD which shows that a controller is executing a command; BRK which confirms that a brake interface has sensed both voltage and current; UP or DWN and CW or CCW indicates controller directional steering and UL or LL and CW or CCW which indicate 1st or 2nd limit switch actuations.

3.6 Servo Amplifier Chassis

The Servo Amplifier Chassis contains the servo amplifiers and power supply used to power the FRM drive motors. The amplifiers and power supply are manufactured by Industrial Drives division of Kollmorgen, Radford, Virginia. The F-R System servo amplifiers and power supply must be easily replaceable modules which use I/O connectors rather than direct wiring to terminal strips for electrical connections. To implement this requirement, the IDD amplifiers and power supplies are mounted on plates attached to the chassis by support pins and Southco fasteners. Signal and power wiring to the modules is done by pigtail cable connectors which mate with adjacent chassis power and signal connectors.

The block diagram of the Servo Amplifier Chassis (D55777S010), assembly drawing (D55007A-022) and chassis wire list (A55007W004) are included in Section 8.0 in Volume II of this manual. The block diagram depicts the chassis bussed and signal connections. Each Servo Amplifier Chassis module has a schematic diagram and assembly drawing which are included in the drawings of Volume II. The functional characteristics of these modules are really those of the IDD modules; these module characteristics are the focus of interest in this section.

Amplifier Scaling Factors

The scaling of the servo amplifiers and motors is as follows:

	FOCUS	ROTATION
VELOCITY DRIVE	3.02 REV PER SEC/VOLT	13.04 REV PER SEC/VOLT
VELOCITY READOUT	3750 REV PER SEC/VOLT	13,500 REV PER SEC/VOLT
MOTOR CURRENT READOUT*	2.6 AMPS RMS/VOLT**	4.348 AMPS RMS/VOLT**
TORQUE SCALING	1.46 LB-FT/AMP RMS	0.60 LB-FT/AMP RMS

* per phase ** from IDD TL, differs greatly from values in the BDS3 manual

IDD Documentation

The IDD BDS3 series Installation and Service manual and IDD component data sheets are included in the data sheet section of Volume II. The IDD Test Limits and Modification Data sheets (in Volume II) are very important data sheets because they describe the specific parameters and configuration of the NRAO motors and amplifiers. They also include photographs of servo response waveforms. Data sheets for some of the BDS3 and PSR3 IDD module components are included in the data sheet section of Volume II. The circuit parameters in the following discussion are taken from the IDD BDS3-208/20-3105A2 and BDS3-208/12-2102A22 TL's (test limits) data sheets.

The IDD documentation is incomplete and very brief, consisting only of a few schematics. There is no theory of operation of any of the IDD modules. The theory of operation presented in this section is incomplete because it was derived by analysis of the circuit types and parameters, some measurements and inference. Measurements at module internal nodes are a bit risky because of the large energy storage in the 300 volt power supply, the extensive 300 volt busses through the modules and the difficulty of access to the boards because of the closed-up package. There are no extender cables or boards for any of the modules. The two control programs for the phase-angle microprocessor are not known to NRAO but some of their aspects may be inferred. There is no IDD parts list for the MC2 and RDP2 boards. At the end of this section there is a list of the IC's used on the MC2 board; this list was developed by noting the IC's types and their location. There is no wire list or comprehensive schematic diagram for either the PSR3 or BDS3. Although the BDS3 MC2 board is used in place of the MC1 board, IDD does not supply an assembly drawing for the MC2 board. An IDD spare parts list is included in the IDD data sheets in

Volume II. The spare parts listed are fuses and replacement boards. Small components such as integrated circuits, transistor Paks, etc. are not listed.

The Servo Amplifier chassis has the following IDD modules: PSR3 Power Supply, Focus BDS3 servo amplifier, two BDS3 Rotation servo amplifiers and the RDP2 Rotation Backlash controller. Under the panels, inside the chassis, three sets of thermal overload relays on the motor lines protect the motors and amplifiers in the event of a fault in the drive amplifiers. The overload relay is connected to the apex Stow Switch circuitry.

Module Wiring

We will first briefly describe the IDD module wiring connections because the diversity of the IDD wiring structures significantly affected the chassis design. The IDD modules were designed to be bolted to a wall and the connections between the modules and the motors are to be hard-wired. Unfortunately, IDD does not manufacture any type of chassis or bins to house the modules or cabling harnesses which could have been used on an amplifier chassis. This is a real deficiency.

The IDD drawing which sketchily depicts the IDD PSR3-BDS3 wiring is C-81543 and is included in the set of IDD data sheets in Volume II. Connection of the RDP2 Anti-Backlash module to the two Rotation BDS3 amplifiers is depicted on IDD drawing C-82305-1.

The IDD amplifiers require bussing of 300 Volt DC motor power, 110 Volt amplifier power and a regeneration fault signal. IDD uses copper bus links to daisy-chain the 300 volts from the PSR3 power supply to the BDS3 servo amplifiers. A second bussed cable carries the regeneration fault signal and amplifier 110 Volt AC power in a four-wire daisy-chained cable from the PSR3 to each BDS3. Motor power connectors are wired to a screw terminal strip in the top front of the BDS3. The three-phase 208 AC is also wired to the PSR3 top front terminal strip. The reader should study the BDS3 and PSR3 photographs in the IDD BDS3 Series Installation and Service Manual included in Volume II.

The analog drive to the BDS3, Remote Inhibit, motor current, velocity monitor and the amplifier +/- 12 volt power monitor signals are connected to the IDD P210 connector on the MC2 board. P210 is a 20-screw-terminal connector. Note that in the BDS3 manual which shows the MC1 board, this connector is referred to as P151. The reader must keep in mind this fact in reading the BDS3 manual. The six wires connecting the IDD resolver (in the motor case) to the MC2 board are connected to the 10-screw-terminal IDD P97 connector. The resolver signals are input to the MC1 board on the P59 connector, here again, the reader must keep in mind this connector difference in reading the BDS3 manual.

Discrete TTL level status and fault signals are fed out of the BDS3 MC2 board on an IDD P100 connector. This connector is not shown on the BDS3 documentation. P100 is an IDC (insulation displacement) connector for flat strip ribbon cable. Drive to the power amplifiers on the Base Drive board is via IDD P139 which uses a ribbon cable to the Base Drive board P45 connector. Direct hard-wiring connects the base drive outputs to power switching transistor Paks mounted on the amplifier heat sinks.

Replacing an IDD module directly wired to the F-R Controller would be very error-prone and tedious; these connections could be easily scrambled. This is the reason for the packaging scheme implemented in the Servo Amplifier Chassis. Connectors on the servo modules are keyed to the chassis location.

The discussion above describes the wiring to interconnect these IDD modules; the reader should refer to the Chassis Block Diagram (D55007S010) which depicts the bus and interconnect wiring between the IDD modules.

PSR3 Power Supply

The PSR3 power supply is an unregulated 300 volt DC power supply powered by 3-phase 208 VAC from the Delta-connected secondary of the IDD isolation transformer. The supply uses full-wave rectifiers to charge two paralleled 2200 uf, 450 volt capacitors. A 10 Kohm, 10 watt bleeder resistor discharges the capacitors when the AC power is removed. A neon pilot indicates the presence of the 300 VDC. When the AC power is removed, the neon pilot will continue to glow until the voltage decays to about 55 volts at which time it is extinguished. Warning: there is still energy in the 2200 uf capacitors after the pilot light is extinguished. The discharge time constant of the capacitors and bleeder is 44 seconds; in 5 time constants or about 4 minutes, this supply will discharge to a few volts. Do not touch the 300 volt lines until the supply is completely discharged. Remember that there are about 400 Joules of stored energy and there may also be some dielectric absorption. Always wait several minutes before touching the 300 volt lines and then be sure to short them out with a clip lead.

The 300 volt DC supply powers the totem-pole switching transistor packs in the BDS3 amplifiers. When the drive motors are decelerating with a large inertial load, the kinetic energy stored in the drive can cause the motors to restore charge to the power supply. The PSR3 contains a regeneration control circuit which dumps this energy into a load resistor via a power transistor Pak mounted on the PSR3 heat sink. A failure of this energy dumping circuit will signal a bus fault to the MC2 control board in the BDS3 via the regen bus fault lines which bus to all the amplifiers. This regeneration control circuit contains a latch which must be manually reset by a switch under the removable plastic cover. Since this function is not remotely controllable, it may be necessary to manually reset the bus fault circuit if it latches up. The series path through the regeneration load resistor is fused by a cartridge fuse.

The IDD regeneration board part number is BDS3 - REG1.

The 300 VDC power is isolated from the antenna 3-phase 208 VAC power by the isolation transformer which has a Delta secondary. The 208 VAC from the secondary powers the 300 Volt rectifiers.

The 110 VAC power for the BDS3 amplifiers and RDP2 backlash controller is provided on a terminal strip in the PSR3. This 110 circuit is fused by a 5 amp fuse mounted on the terminal strip panel under the PSR3 removable plastic cover.

The PSR3 rectifiers, regeneration power transistor Pak and regeneration load resistor are mounted on the PSR3 heat sink which is cooled by a fan.

The PSR3 version used with the F-R system is the PSR3-208/50-01-002 which is capable of supplying BDS3 motor loads requiring 50 amps of 208 V AC input power.

BDS3 Servo Amplifiers

The IDD servo amplifiers are velocity servos; the analog drive to the amplifiers causes the motors to run at a speed determined by the amplifier's velocity scaling. Variations in motor torque load do not affect motor speed; the servo amplifiers increase or decrease the motor current and torque angle as required to maintain the speed at the value commanded by the analog drive level.

Two versions of the BDS3 servo amplifier are used: BDS3-208/20-3105A (for Focus) and BDS3 - 208/12-2102A22 (Rotation). The Focus version has a higher current (35 amps) rating than the Rotation version (21 amps) and the heat sink is fan-cooled. ACS3-COMP1 servo compensation boards are set up for the particular inertial and drive requirements of the two applications. These boards are small boards which are plugged into the MC2 board. The IDD BDS3 manual has a photograph of the ACS3-COMP1 board. The configuration of the servo compensation boards is defined by the Focus and Rotation Test

Limits documents TL BDS3-208/20-3105A2 and TL BDS3-208/12-2102A22 respectively.

Two printed circuit boards perform most of the motor control functions in the BDS3. The Base Drive board contains power supplies, current sensors, a DC-to-DC converter and buffer transistors to drive the totem-pole power transistors mounted on the heat sink. The Base Drive board is described below.

The ACS3 - MC2 (Motor Control) board contains the analog and digital control circuitry to close the velocity loop. The MC2 board is described below.

We will first briefly consider the motor drive circuitry. As shown on IDD drawing C-81543, the three motor lines are driven by totem-pole power transistor Paks mounted on the BDS3 heat sinks. The switching control is such that a motor lead is powered by either the upper or lower transistor depending upon the signal phase. The totem pole transistors are powered by the 300 volt positive and negative buses. The 300 volt common label on the IDD drawing is erroneous; the 300 volt bus lines are totally isolated from the amplifier common. The common line from the junction of the three stator windings in the motor is connected to chassis ground but there is no current path from this ground to either of the 300 volt bus lines. The instantaneous sum of voltages or currents in the center leg of a balanced-load, Y-connected 3-phase circuit is zero. The operation of this power transistor switching circuitry is described in detail in the MC2 board description below.

The totem-pole, dual Darlington circuit switching transistor paks which drive the motor leads are a Powerex KD224575 which is capable of switching up to 75 amperes at voltage levels as high as 600 volts. Data sheets for the KD224575 are included in Volume II. Two versions of this Powerex transistor Pak are used.

The IDD brushless permanent magnet motors are analogous to a synchronous motor. The permanent magnet rotor moves synchronously with the rotating stator magnetic field; there is no "slip" as in conventional AC motors. The stator winding drive is a 208 VAC, (line-to-line, 110 volts, line to ground) chopped three-phase, 4 Khz phase modulated drive. The frequency of the phase modulation is the shaft rotational frequency. The 300 volt supply is about twice the peak voltage (155 volts) of the 110 V AC line. The totem pole switches drive the motor lines to the 300 volt bus levels; this may be seen by (carefully) monitoring the motor drive signal with an oscilloscope.

Base Drive Board (BDS3 - BD1)

The Base Drive board contains the DC-to-DC converter which powers the BDS3 analog and digital circuitry. The 110 V AC line input is full-wave rectified by a bridge and the resultant 150 volts DC powers the converter. Note that the Base Driver board power supply is not transformer isolated from the 110 V AC power.

The frequency of the DC-to-DC converter is about 45 Khz; it is determined by the 40.2 Kohm resistor and the 1000 pf capacitor. The frequency of oscillation is given by: $f = 1.8/RC$. The optical isolator pulse-width modulation drive frequency is 4 Khz. The oscillator is a Unitrode 3842 PWM controller; a 3842 data sheet is included in Volume II. The DC-to-DC converter transformer is driven by a Motorola MTM4N45 power FET; data sheets for the MTM4N45 are included in Volume II.

Windings on the DC-to-DC converter transformer power unregulated +/- 16 and 8 volt power supplies for the analog and digital circuits of the BDS3. The DC power is filtered by 330 uf capacitors. Regulators on the MC2 board produce +/- 12 from the +/- 16 and +5 volts from the 8 volts.

These Base Driver board driver circuits are powered by an AC drive from the DC-to-DC converter which generates the BDS3 DC power. This converter is shown on Sheet 2 of C-81058-1. Four center-

tapped, isolated windings on the converter transformer drive rectifiers and filter capacitors on the six driver circuits. Three center-tapped windings drive the A, B and C "TOP" circuits and the fourth center-tapped winding drives the three "BOTTOM" circuits. For example, the BD A top winding with the 19, 20 and 21 labels drives the "A TOP" terminals labeled 28-19, 28-20 and 28-21. (The 28 is circled on the drawing.) Sixteen volts DC has been measured across pins 8 to 5 of isolator 94 (pin 8 is positive). Eight volts DC has been measured from common to pin 5 (pin 5 is negative with respect to common). This "TOP" driver and power supply are isolated of necessity, the circuit is referenced to the + 300 bus and turns on the "TOP" transistors.

The center-tapped winding labeled BD BOTTOM drives rectifiers and filter capacitors common to the three sets of "A BOTTOM", "B BOTTOM" and "C BOTTOM" driver circuits on the lower half of Sheet 1. 16.9 volts DC has been measured across isolator pins 8 to 5 (pin 8 is positive). The use of separate power supplies for the top circuit and a common power supply for the bottom circuits suggests that the bottom circuits require less power. Assuming that the four transformer windings are identical, it would appear that the top circuits require more power than the bottom circuits but the circuit values do not support this theory. This "BOTTOM" circuit and power supply are also isolated of necessity; the circuit is referenced to the - 300 volt bus.

The power transistor drive circuitry is not very obvious in the IID BDS3 manual. The BDS3 internal signal names and connector pins are listed but there is no interconnect wire list; the user must puzzle out the obscure connector references. Three lines from each of the six driver circuits on the Base Drive board (see Sheet 1 of IDD drawing C-81058-1) are connected to the six power transistors in the three Paks. The six circuits are labeled "A TOP", "A BOTTOM", etc. The three lines from each of these six driver circuits labeled C1, B1 and E1 (TOP) and C2, B2 and E2 (BOTTOM) are connected to the designated power transistor collectors, bases and emitters, respectively.

NRAO drawing B55007S018 shows the functional circuit of one phase of the power switching circuitry. This drawing is in volume II.

The Base Driver board phase modulation drivers are enabled by six HPCL 2200 optical isolators driven by pulse-width modulation logic on the MC2 control board. These isolators and the transformer-coupled AC drive from the DC-to-DC converter maintain the isolation of the 300 volt bus and motor lines from the MC2 control board analog and digital circuitry. This isolation is a very desirable property as it isolates the lightning-susceptible motor lines from the balance of the F-R Control system.

The Base Driver board power transistors are TIP125, MJE250 and MJE16004, all manufactured by Motorola. Data sheets for these transistors are included in Volume II.

Phase A and C current to the motor lines is sensed by Hall-effect sensors in the gap of a gapped magnetic core. The motor drive lines from the totem-pole paks pass through the core apertures. Operational amplifiers buffer the output of the Hall-effect sensors to the MC2 board. The amplifiers are National Semiconductor LF444ACN quad J-FET input operational amplifiers. Data sheets for the LF444 are included in Volume II. The excitation for the Phase A and C Hall-effect sensors is + 5 volts from an LM340 IC regulator.

The +300 volt bus runs through the aperture of a Hall-effect current sensor which is output to the MC2 board. This Hall-effect sensor is excited by + 12 volts. The DC output of the sensor is threshold compared by logic on the MC2 board.

The optical isolator for the regeneration fault bus is an HP6N139 which maintains the isolation of the 300 volt circuits from the digital logic.

MC2 (Motor Control) Board

The heart of the BDS3 servo amplifier is the ACS3 - MC2 control board; note that the control board shown in the IDD BDS3 manual is the MC1 board which is functionally similar but not identical to the MC2 board. The MC2 board is the focus of the following discussion. For this discussion, the reader should have at hand schematics C55007S005, B55007S018 and the IDD TL's for the two BDS3 versions.

The circuit parameters contained on the ACS3-COMP1 board are distributed through the following discussion. Focus and Rotation values are identical in most cases; when they differ it will be noted in the discussion. The microprocessor programs are different but the programs and differences are not described in the IDD manual or drawings.

The servo velocity drive is applied to differential amplifier 199-1,2,3. The input amplifier is a unity-gain differential amplifier for high common-mode suppression. Note the RC filtering on the input lines to reduce high frequency noise. The 3 db frequency for the filters is 10 Khz. Amplifier 199 is a National semiconductor LF412ACN. Jumper 10 is installed and jumper 11 is omitted. The output of this amplifier drives the Sum Stage 289-1,2,3 which sums the analog drive with the velocity feedback signal from the R/D converter (jumper 8 is installed). The gain of the sum stage is 1.33. (Amplifier 289 is a TI TL074B operational amplifier; data sheets for this amplifier are included in Volume II.) In the case of the Rotation BDS3's, the RDP2 Backlash Controller Equalizer signal also drives this summing junction via the SUM 1 input. We will discuss the backlash configuration below.

The Sum Stage is the velocity summing junction in which velocity feedback is compared with the velocity reference. There is also a current feed back summing point further into the amplifier; this will be discussed below.

The BDS3 servo amplifier is a PID amplifier; the feed forward control circuits perform proportional, integrating and differential operations on the error signal.

The proportional state (IDD designation) is amplifier 320-1,2,3 which operates open-loop at DC. Amplifier 320 is a National Semiconductor LF412ACN. The series RC combination of 2.5 Kohms pot resistance and 0.1 uf capacitor makes this stage an augmenting integrator with a time constant of 0.25 milliseconds. The 3 db frequency is 4 Khz.

The derivative stage is amplifier 320-5,6,7. This amplifier is also a National Semiconductor LF412ACN. The Focus derivative stage has a gain of 7.5 and the Rotation derivative stage has a gain of 75. In practice, it has been found necessary to reduce the value of R60 in the Rotation derivative stage to eliminate oscillation.

The integrating stage is the two unity gain (DC) stages of amplifier 289-12,13,14 and 8,9,10. Amplifier 289 is a low-noise, JFET input TI TL074B operational amplifier. (Data sheets for this amplifier are included in Volume II.) The time constant of these two Focus amplifiers is 0.3 milliseconds; the amplifier's 3 db points are 3.3 Khz so the composite response is down 6 db at 3.3 Khz. The time constant of these two Rotation amplifiers is 1.36 milliseconds and the 3 db points are 730 Hz so the composite response is down 6 db at 730 Hz.

Analog switches 262-9,10,11 and 6,7,8 short the feedback paths of the proportional and derivative states when the servo amplifier is inhibited. These switches are Siliconics DG 201 MOS FET switches.

The velocity error signal from the filter stage is fed to two integrating DAC's and to a polarity comparator circuit to develop a direction discrete for input to the microprocessor. For negative polarity

error signals, amplifier 146-1,2,3 (TL074B) operates as a unity-gain amplifier because the output is positive which feeds back to the input via the diode and feedback resistor. With a zero volts input, the amplifier output is about + 0.6 volts. As a result of the diode in the feedback network, when the error signal is positive (even slightly positive), the amplifier operates open-loop because the feedback is disconnected because of the diode polarity. In this mode, the gain is very high and a large signal is fed to LM339 comparator 329-10,11,13. This comparator has an open-collector output pulled up to + 12 volts by the 20 Kohm resistor. The comparator positive input is referenced to - 6 volts; when the negative input is more positive than - 6 volts, the output goes to ground and sinks current through the pull-up resistor. When the input is more negative than - 6 volts (even slightly more negative), the output goes high because of the pull-up resistor but is limited by the resistor-diode network. In this condition, the drop across the 15 Kohm load resistor is + 4.8 volts (ignoring the gate input current). This circuit is thus a high-gain polarity comparator operating around the zero volts region of the error signal; the output signal on the 15 Kohm resistor switches between 0 and + 4.8 volts as the error signal moves through zero volts. The comparator output drives an inverter (145-14,15, a 4049) and buffer (174-7,6, a 4050). The inverter drives Port D7 on the microprocessor to signal the polarity of the error signal.

Amplifier 146 is a TI TL074B, comparator 329 is a National Semiconductor LM 339 and inverter 145 is a CMOS 4049.

The velocity error signal from the filter stage also drives the reference inputs on two multiplying DAC's (AD7528). The output of these DAC's is the product of the reference input (i.e. velocity error) and the input digital value. The microprocessor synthesizes two (phases A and C) sinusoidal signals, 120 degrees apart in phase. The B phase is generated by summing A and C in amplifier 316-12,13,14 (TL074B). After generation of the third (B) phase (120 degrees from A and C), these three signals are fed into filter amplifiers and analog comparators to drive the six optically-coupled isolators that control the driver transistors mentioned above. The amplitude of the sinusoids increases with the magnitude of the error signal. The sense of phase rotation of the A and C sinusoids is determined by the polarity of the logic drive to the Port D7 input. A high on this input will cause one sense of phase rotation and a low will cause the opposite sense. Velocity "hunting" may cause the bit to flicker but either the high or low sense will dominate. When flickering occurs, it causes momentary phase advances or retardations in the sinusoids.

The 3-phase sinusiods drive three filter-summer amplifiers (316, TL074B) which sum three current feedback signals with the sinusiods. The weighting of the current feedback and the sinusoid drives is identical. The DC gain of the amplifiers is 91.66 and the feedback capacitor produces a 3 db frequency of 579 Hz. The series RC combination adds augmenting integrator properties to the amplifier; the 3 db frequency of the augmenting circuit is 6750 Hz for Focus and 640 Hz for Rotation. The current feedback is positive so that as the motor load increases, it causes the power drive to the motors to be increased.

The outputs of the filter-summer amplifiers drive the negative inputs of three LM339 comparators (329) which are modulated by a 4 Khz, 20 volt peak-to-peak ramp signal on the positive input. These three comparators function as pulse-width modulators. The ramp signal is generated by the ramp generator consisting of amplifiers 324-5,6,7 and 324-12,13,14 (TL074B). (The operation of this ramp generator is described below.) The duration of the pulse width modulator output is determined by the relative amplitudes of the ramp signal and the error-current feedback signal. Assume for the moment that the comparator's negative input is grounded; for this configuration, the upper comparator threshold is given by: $V_{ut} = (R_i/R_f)V_{out}$ and the lower threshold is similar with V_{gnd} instead of the comparator positive output. With the input and feedback resistors of this circuit, + V_{sat} is +0.010 volts and - V_{sat} is gnd. Clearly, the comparator outputs will switch from positive high to ground with only a 10 millivolt signal transition. Now consider the real case in which the analog signal from the filter-summer amplifier drives the negative comparator input; if at any time the amplifier signal is 10 millivolts more negative than the ramp level, the output is + V_{sat}, and if the amplifier signal is 10 millivolts more positive

than the ramp level, the output is gnd. Note that we are dealing with the difference relationships of the two signals in the comparator; these determine the width of the comparator output. If, for example, the filter-summer amplifier output is - 5 volts, then the comparator output will be high for the period in which the ramp is more negative than - 5 volts, which is 25% of the waveform period. If the filter-amplifier output is 0 volts, the output will be positive for 50% of the period and if the filter-summer amplifier output is + 5 volts, the output will be high for 75% of the period. This circuit thus transforms a signal level to a pulse duration.

The comparator output drives the resistor-diode circuit discussed above in the direction discrete fed into the microprocessor. The comparator output on the 15 Kohm resistor is 0 and + 4.8 volts which drive the three sets of "TOP" and "BOTTOM" nand gates. The gates are CMOS 4011 two input nands, enabled by the INH MOD term (the inhibit is discussed below). The gates drive the resets of 4520 dual CMOS counters and are clocked on the Enable line by a 270 Khz clock from the ring of 4049 (145) inverters. The Q8 output is wired back to the clock input and the circuit functions as an 8-count delay from the high-to-low transition of the reset input driven by the nand gate. The counter Q8 outputs, inverted by the 4049 inverters, sink current through the optical isolators on the Base Drive board. Since the "TOP" and "BOTTOM" drives are logic complements, the drive alternates to the associated power drivers on the Base Drive board. The drivers on this board alternately switch on the Darlington transistors in the power transistor paks. The 8-count delay logic provides a quiescent period of 30 microseconds between the times that the "TOP" and "BOTTOM" transistors are turned on. This prevents the heavy current spikes on the 300 volt bus which would result from the simultaneous turn-on of the Top transistors and turn off of the Bottom transistors.

Drawing B55007S018 (in Volume II) shows one phase of the driver circuits, optical isolator through the Darlington pairs. The timing diagram (Drawing A55007D001) in Volume II shows how the phase is modulated as a result of the velocity error.

One of the features of the BDS3 is that the torque angle, the physical angle between the motor stator magnetic vector and the poles of the permanent magnet rotor, is increased from a small angle to a maximum of 90 degrees to increase the motor torque as a function of load torque. The rotor position is fed back to the MC2 board and the most significant 11 bits from the Resolver to Digital converter are input to the microprocessor. This shows rotor angle. The program in the processor compares this angle with the sinusoid angle and as a function of load current, increases (or decreases) the sinusiod phase to change the rotor-stator magnetic vector phase angle. The current monitor input to the microprocessor is implemented by two discretes from operational amplifier comparators driven by motor current. These two comparators are 22-1,2,3 and 22-5,6,7 (LM324) and are input to ports D3 and B5. The absolute value of velocity error from the velocity error comparator is input on port D0. It is necessary to study the microprocessor program to quantify the effects of these discrete inputs. Another uncertainty is the connection of the negative input to amplifier 22-5,6,7. The IDD documentation does not show where it is connected.

The ramp generator used by the pulse-width modulators is an integrator which drives a comparator; the comparator output is fed back to drive the integrator. Consider the input resistor to be driven by a bi-polar square wave; the integrator output will ramp negative for a positive input and positive for a negative input. If the integrator output is connected to a comparator, the comparator output will switch high and low as a function of the ramp input. If the comparator output is connected to the integrator input resistor, the integrator will ramp up and down and the amplitude and frequency is a function of the integrator parameters R and C and the comparator parameters R_i and R_f. For a simple circuit, the frequency is given by: $f = R_f/4R_iRC$. In the MC2 ramp generator circuit, the comparator output square wave is clamped to + 2 and - 2 volts by the diodes connected to the +2 and -2 volt sources from the amplifier 324-8,9,10. The 1000 pf capacitor across the comparator input provides filtering to eliminate multiple transitions near the switching thresholds caused by noise. The 30 Kohm resistor fed

back to the comparator input adds Schmidt trigger properties in that the ramp input must exceed the normal threshold level to effect a transition.

The phase A and C current monitor signals from the Base Drive board are summed in amplifier 285-1,2,3 to synthesize the phase B current. The three AC current signals are rectified by 227 (TL074B) amplifiers and diodes; when the AC signals are positive, the amplifier outputs are negative and sink current through the summing amplifier input resistors. When the AC current signals are negative, the amplifier output goes positive to saturation (there is no feedback resistor in this condition because of the diode). Because the currents are rectified by the 227 amplifiers, the current monitor signal is independent of motor rotation direction. IDD labels this signal /I/. The 0.1 uf capacitor filters the summed signal. The 3 db frequency for this amplifier is 50 Hz which is about 1/10 the switching frequency of 4 Khz.

One of the powerful protective features of the MC2 board is the torque foldback circuit. The summed motor current from summing amplifier 227-12,13,14 drives the foldback circuit on sheet 1. Two stages of operational amplifier (285) buffer the integrator amplifier 285-8,9,10. The integrator drive current level is scaled by the R17B resistor in the first amplifier stage. An offset current is fed into the second amplifier stage; this biases the integrator output positive to offset the negative motor current input. If the motor current exceeds the offset current, the integrator begins to charge negatively. When the current value is zero (when the motor is not running), the integrator will charge up to the positive saturation voltage of the operational amplifier. As motion starts, the current begins to increase and the integrator begins to charge negatively. When the drive from the 285-12,13,14 stage exceeds approximately -.6 volts, the drive is increased by the diode-resistor shunt path. This is probably done to decrease the integration time for rapidly increasing motor currents. The integrator time constant is 15 seconds, determined by the 18 uf capacitor and the 820 Kohm resistor. The integrator output is fed to a comparator through a 10 volt zenar diode. The other comparator input is the rectified velocity error signal from amplifier 146-8,9,10. This signal is also fed through a 10 volt zenar diode. The zenars are probably used to provide a 10 volt offset for the comparators. An external current limit signal can limit motor current via the 262-1,2,3 analog switch. This feature is not used by the F-R control system.

Fault Logic

The MC2 board contains fault-status logic which is read out to the F-R Controller. The fault-status discretes are also displayed on the BDS3 front panel LED display. The fault states are summed in or-gate 358 which causes the drives to be inhibited by the inhibit signal on the pulse-width modulator gates described above.

The fault-status data read out of the BDS3 to the F-R Controller is: Bus Fault, Foldback, Over-temperature, Drive Up, Over-current, Power Loss, Drive Inhibited and Over-Speed.

Bus Fault is signaled by the regeneration control circuitry on the Regeneration board in the PSR3.
Foldback is signaled by the discrete from the foldback circuit described above.

Over-temperature of the BDS3 heatsink is detected by a normally-closed thermostat switch; the contact output drives a 4050 buffer which functions as a comparator. The output of the buffer drives a latch input through another small RC filter. The temperature discrete is buffered out by another 4050 for remote readout via the P100 connector. The latch output drives the fault or gate and the overtemperature LED on the front panel LED display.

Drive Up indicates that the BDS3 is not inhibited by faults and has been enabled by the BDS3 enable (on the remote enable input) from the F-R Controller.

The over-current signal from the Hall-effect sensor on the Base Drive board is input on pin 5 of

connector 163. (This connection is totally obscure in the IDD documentation.) A 10 Kohm pull-up to + 12 volts and a small RC filter condition the signal slightly as it drives the input of a 4049 (128). The 4049 acts a comparator and the discrete output drives another RC (7.5 usec time constant) filter and the filter output drives one of the inputs to a 4078B latch. The associated latch output drives the fault or gate 358 (4078) and the overcurrent LED in the front panel LED display.

The Power Loss fault is derived from the latch which is driven by a delay circuit on the MC2 + 5 volt power, a momentary drop or loss of + 5 power will set the latch bit.

Drive Inhibited indicates that a fault or the external inhibit is active.

Over-speed is generated by an LM339 comparator driven by the absolute value of motor speed from the Resolver to Digital converter. The over-speed discrete is sensed by a port (PD1) on the microprocessor. Over-speed sets one of the latches in the fault circuitry.

The latched faults in chip 359 (4078) can only be cleared by a reset from the microprocessor via port B3.

The BDS3 is reset by a low-true BDS3 reset from the F-R Controller which drives the microprocessor reset input.

Rotation Backlash Control

A fundamental mechanical design feature of the Rotation drive is the use of two Rotation drive motors coupled to the Rotation ring gear by individual motor pinion gears. When this mechanical configuration is controlled by a backlash controller, gear backlash can be eliminated so that Rotation position is very stable. In operation, a backlash controller causes the motors to act in concert in moving the ring gear but there is a small torque difference or bias between the two motor torques which causes the motor pinion gears to have an oppositional torque.

The IDD RDP2 functions as a backlash controller; IDD drawing C-82317 shows the signal interconnections between the RDP2 and the two Rotation BDS3 servo amplifiers. Drawing C-82305-1 shows the schematic of the RDP2 controller. Drawing D55007S009 shows the signal wiring of the NRAO Rotation Backlash controller module. The reader should have drawings D55007S005, C-82317, C-82305-1 and D55007S009 at hand during the following discussion.

All the backlash control operations are performed in analog circuitry, no resolver, digital, motor power driver or microprocessor operations are performed. Both feed-forward and feedback operations in controlling the drive to the two Rotation motors.

There are several jumpers and an application-dependent resistor which configure the board. The jumpers are as follows: J17 in, J16 out; J4 in, J3 out; J9 in, J8 out; J23 in, J22 out. R31 is 56.2 Kohms which makes amplifier 32-1,2,3 gain be 1.183.

The Rotation velocity command from the F-R Controller is input to the RDP2 differential high and low inputs; the input is a unity-gain, differential stage (43-5,6,7) with high common-mode suppression. Note the LC filtering on the input lines to reduce high frequency noise; the 3 db frequency of one stage is about 25 Khz. The output of the differential stage drives two tandem unity-gain operational amplifier stages (43-1,2,3 and 13-5,6,7). Jumper 23 causes Amplifier 43-1,2,3 to drive both West and East BDS3 servo amplifier differential inputs. Potentiometer 72 (10 Kohm) is the gain pot for these two stages and potentiometer 73 (10 Kohm) is the zero pot to null any zero offsets between the two tandem unity-gain stages. Note that if the gain pot 72 is set to full gain, the velocity drive to the two

BDS3's is 1.18 times the level commanded by the F-R Controller. This modest gain permits the gain pot (72) to be backed off from full CW to permit operation at nominal unity gain.

The Velocity Error Output Derivative stage of the two Rotation MC2 boards is fed back to the RDP2 via terminal 6 on connector 210. At this pickoff point, the velocity signal in the MC2 has undergone three signal inversions, has been amplified by a factor of 100 by the Sum and Derivative stages and subjected to the filtering of the proportional and derivative stages. Note that this pickoff point is also upstream of the heavy low-pass filtering of the MC2 integrating stages 289-12,13,14 and 289-9,10,8.

These two velocity error signals are fed back to the RDP2 to two differential (for common mode noise suppression), unity gain amplifiers (18-5,6,7 and 32-5,6,7). The output of the West velocity amplifier (32-5,6,7) drives mixing amplifier 32-1,2,3. The output of East amplifier 18-5,6,7 drives a second stage of amplifier 18-1,2,3 whose output is fed into mixing amplifier 32-1,2,3 via jumper 17. The mixing amplifier drives two more unity-gain tandem amplifiers 2-1,2,3 and 2-5,6,7. The output of the first stage, amplifier 2-1,2,3 is fed back to the East BDS3 Sum 1 input on connector 210, terminal 4. The output of the second stage, amplifier 2-5,6,7 is fed to the West BDS3 Sum 1 input on connector 210, terminal 4.

Note that in each of these unity-gain amplifier paths, there are four signal inversions so the feedback to the MC2 Sum 1 inputs has the same signal polarity as the velocity drive signal at the Sum stage in the MC2 analog circuitry. Secondly, note that in mixing amplifier 32-1,2,3 the East and West signal have opposite polarities because the East path has undergone one more signal inversion because of amplifier 18-1,2,3. Now what has all this feedback and feed forward circuitry accomplished? In theory each BDS3 signal path in the two MC2 boards is identical but in actuality each path will subject the signals to slightly different gain and frequency response conditions because of component variations, etc. By mixing opposite polarity signals from the two paths in mixing amplifier 32-1,2,3, the signal differences tend to be cancelled out so that the combined outputs fed out on the output amplifiers are a composite of the response of the first four stages of each BDS3. The Sum 1 input resistor in the MC2 is 15 Kohms so the equalizer feedback signal has the same gain as the velocity drive stage.

The mixing stage, amplifier 32-1,2,3 has back-to-back 8.2 volt zenar diodes across the feedback resistor to limit the output of this stage in the event that the velocity error becomes large.

Potentiometer 75 permits cancellation of zero offsets in the equalizer amplifier paths.

Now how is the backlash bias between the two MC2 boards accomplished? Amplifiers 20-1,2,3 and 20-5,6,7 provide an equal but opposite polarity signal which is input to the MC2 boards on the I Offset terminal (13) of connector 210. This injection point is the input to the two integration stages 289-12,13,14 and 289-8,9,10. The input resistor is 10 Kohms so the offset gain is twice the gain of the derivative stage which drives the integrating stages. Pot 74 sets the level of backlash bias; note that the backlash bias can be either polarity.

Note that the basic operation of the two Rotation servo amplifiers are unchanged; the gains, signal levels, etc. are unchanged. What has been done by the RDP2 is to normalize the two amplifiers response so that they act in concert and deliver equal amounts of torque with only the settable torque bias as a difference. This is the real function of the backlash control circuitry; a backlash torque could have been implemented by simply adding some offsets by a few resistors and perhaps a pot.

Like the BDS3's the RDP2 board has its own +/- 12 volt power supply powered by 110 V AC.

Motor Overload Relay-Emergency Stop Circuitry

The Servo Amplifier Chassis contains thermal overload relays which sense excessive currents in the motor drive lines. The relay circuits are combined with the Apex Emergency Stop switch circuitry. In the event of a thermal relay trip or the actuation of the Apex Emergency Stop switch, power to the motors and brakes is inhibited. Drawing B55007S019 in Volume II depicts the functional schematic of this circuitry.

Three motor lines from the three motors are wired to thermal-trip overload relays mounted under the amplifier panels in the relay chassis. The relays contain a three-pole contactor with normally-open contacts and three heater resistors mounted in the relay body. The contacts and heaters are connected in series in the three motor lines. In the event of excessive motor currents, heat from the heater resistors trip a normally-closed contact in the overload relay. The contactor coil is powered by 110 V AC. As shown by the drawing, 110 V AC powers the contactor coils through the normally-closed thermal-trip contacts; if the contacts are opened as a result of excessive motor current, current to the contactor coil is interrupted and the motor lines are disconnected from the amplifier outputs.

The normally-closed Emergency Stop switch at the Apex is wired in series with the 110 V AC contactor excitation; when the switch is actuated the contactor excitation is interrupted.

The Pedestal Room Junction Box contains a four-pole, 110 V AC relay powered by the servo amplifier chassis overload relay contactor 110 V AC power. Excitation for the three motor brakes runs through the normally-open contacts of this relay. The fourth set of (isolated) contacts are wired to the F-R Controller Emergency Stop input. When the relay contacts open, a logic high is input to EPROM2, Port B3. The controller software senses the contact states but there is no hardware inhibit of the controller FRM interface circuitry from this contact input.

MC2 Board Integrated Circuits List

MC2 BOARD PART NUMBER	IC PART NUMBER	MANUFACTURER	IC TYPE
145, 178, 128, 355, 356	4049		CMOS TO TTL INVERTER INTERFACE
197, 198	HF4011		CMOS 2 INPUT NAND
175, 176, 177	HF4520		DUAL SYNCHRONOUS 4- BIT COUNTER
174,	4050		CMOS BUFFER
52,53	74LS5244		OCTAL CMOS BUFFER
359	HEF4078		QUAD CMOS LATCH
358	HF4078		OCTAL INPUT CMOS NOR
262	DG201A	SILICONICS	ANALOG SWITCH
199, 320	LF412ACN	NATIONAL	OPERATIONAL AMPLIFIER
285, 289, 261, 64, 316, 227, 324, 63, 136, 146,	TL074B	TI	OPERATIONAL AMPLIFIER
329, 361	LM339	NATIONAL	QUAD ANALOG COMPARATOR
22	LM324	NATIONAL	QUAD ANALOG COMPARATOR
101	68705-R5	MOTOROLA	8-BIT MICROPROCESSOR
27	APT1750/0416	ANALOG DEVICES	RESOLVER TO DIGITAL CONVERTER
121	AD7528	ANALOG DEVICES	DUAL D/A CONVERTER
250	7812		+ 12 VOLT REGULATOR
203	7912		- 12 VOLT REGULATOR
180	7805		+ 5 VOLT REGULTOR

3.7 CONTROL ALGORITHMS

This section describes the control algorithms which are the essence of the operations performed by the F-R Control System. Most of these algorithms are expressed in an IF-THEN-ELSE format. Registers and operands in the programs are printed UPPER CASE and control words are printed in UPPER CASE, underlined and bold. For simplicity, the algorithms express the essence of the control actions. Program implementation details such as adjusting D/A converter levels, merging of DAC states and control discretes, etc. are omitted. Section 3.8 describes the operation of the programs in detail.

INITIALIZATION (both programs)

The INITIALIZATION code is entered when the processor emerges from the RESET state. This very important code initializes the processor and system electronics. The conditions established by this code determine the memory address ranges, operating modes of the Interrupt system, DMA, Timers and RAM and EPROM I\O ports. The RAM memory and RAM and EPROM I/O ports are cleared. The servo amplifier logic is reset and the amplifier fault discretes are tested. After a 3 second delay the servo amplifier 300 volt power is energized.

```
1)  SET UMCS (UPPER MEMORY CHIP SELECT) TO OFFB8H, TO MAKE THE UPPER MEMORY = 2 KBYTES.
2)  SET LMCS (LOWER MEMORY BLOCK SIZE) TO 2 KBYTES.
3)  SET MPCS (MID MEMORY SIZE) TO (4 EACH) 2 KBYTE BLOCKS.
4)  SET MMCS (MID MEMORY BASE ADDRESS) TO 8 KBYTES.
5)  SET PACS (PERIPHERAL CHIP SELECT BASE ADDRESS) TO 0000H.
6)  SET EPROM1 PORTS A AND B TO THE OUTPUT MODE.
7)  SET EPROM2 PORT A TO OUTPUT MODE, PORT B BITS 0, 1 AND 2 TO OUTPUT MODE, BITS 4, ... 7 TO INPUT MODE.
8)  SET RAM PORTS A, B AND C TO OUTPUT MODE.
9)  CLEAR EPROM1 AND EPROM2 OUTPUT PORTS.
10) CLEAR RAM PORTS A, B AND C.
11) SET UP DMA-0 IPI CONTROL REGISTERS.
12) INITIALIZE TIMER IPI CONTROL REGISTERS.
13) ZERO RAM MEMORY.
14) INITIALIZE INTERRUPTS IPI REGISTERS.
15) RESET BDS3 SERVO AMPLIFIERS.
16) WAIT 3 SECONDS.
17) TURN ON 3-PHASE 208 VAC POWER.
18) TEST FOR BDS3 BUS FAULTS.
```

DSTOR Subroutine Algorithm (both programs)

DSTOR is the data gathering subroutine which gathers and formats Apex and servo amplifier data for use by the position control and data readout programs. When called, DSTOR emits a data request to the S102 and tests for the arrival of the data in the Apex data registers. Data presence is indicated by a 1,0 pattern in the top two register data bits. After emitting the data request, the subroutine cycles through 12 loops, testing for the 1,0 pattern. If the data does not arrive within this period, a fault is set in the FAUL1 Apex status word which is the Apex fault reference; the FAUL1 state is also read out as monitor data.

```
BEGIN   DSTOR
        REQUEST APEX DATA
DSTOR1  IF  DATA IS PRESENT  THEN   SET APEX OK FLAG IN FAUL1, GO TO DSTOR2
        ELSE
            IF  LOOP COUNTER = 12  THEN  SET APEX FAULT IN FAUL1, GO TO DSTOR6
            ELSE   GO TO DSTOR1
DSTOR2  SET OK IN FAUL1
        FORMAT POSITION DATA IN POSD
        ERROR = POSD - POSCEC
        SET DIR
        FORMAT VELOCITY
        IF  BRAKE RELEASED  THEN  SET BRAKE RELEASED IN FAUL1, GO TO DSTOR3
        ELSE  SET BRAKE ENGAGED IN FAUL1, GO TO DSTOR3
        SAVE SECOND SCREW STATE IN PHASEA
        SAVE LIMITS, BRAKE AND SCREW STATUS IN FAUL1
        FORM RANDOM NUMBER, SAVE IN RANDOM
        IF  APEX ANALOG DATA MUX ADDRESS = ROTATION MUX ADDRESS, GO TO DSTOR6
        ELSE
            IF  FOCUS APEX DATA VALUES WITHIN LIMITS  THEN  GO TO DSTOR6
            ELSE  SET APEX FAULT FLAG IN FAUL1,  GO TO DSTOR6
DSTOR6  SET S105 BRAKE LED TO BRAKE STATE
        IF  LIMIT FAULTS  THEN  SET S105 LIMIT LED,  GO TO DSTOR9
        ELSE  GO TO DSTOR9
DSTOR9  READ AND STOR BDS3 FAULT-STATUS IN FAUL2
        SAVE EMERGENCY STOP AND DRIVE LOCKOUT STATUS IN FAUL1
END     DSTOR
```

BOSS Program (both programs)

The BOSS program is a supervisory and scheduling task program which operates asynchronous with the drive motion programs in TMR2. BOSS senses the presence of a command execution by testing the state of DRVATV. If active, CHKDRV is called and upon return, control is returned to the BOSS entry point. If the drive is not active BOSS tests for the state of the MANOVR flag to determine if the manual mode override mode is set. BOSS next tests the MAN-CMP switch to see if the controller is to operate in the LOCAL mode. If so, control is transferred to the LOCAL program. BOSS next tests to see if the servo amplifier reset request is active; if so, the amplifier is reset, a 1-second delay is executed, the timer is turned off and control returns to the BOSS entry point. BOSS sequentially tests for the soft reset request; the state of DRVATV, and NAPREQ. If any of these requests are active BOSS executes the requested action and returns control to the entry point of BOSS.

```
BEGIN  BOSS
       EXECUTE RANDOM DELAY
       SP = STACK
       IF  APEX FAULTS  THEN  GO TO RSCMD
       ELSE
           IF  DRVATV = 1   THEN CALL CHKDRV,  GO TO BOSS
           ELSE  CALL DSTOR
           IF  MANOVR = 1  THEN  SET MANUAL MODE IN SYSTEM, GO TO NEXT1
           ELSE
               IF  IN LOCAL   THEN  GO TO LOCAL
               ELSE
NEXT1              IF   BDS3 RESET REQUEST   THEN  CALL BDSRS, CX = 1, CALL DELAY, CALL TMROFF, GO TO BOSS
                   ELSE
                       IF  RSCMD = 1  THEN  GO TO CMDRS
                       ELSE
                           IF  NAPATV = 1  THEN   DRVREQ = 0,  GO TO BOSS
                           ELSE
                               IF  DRVREQ = 1  THEN  CALL DRVINT,  GO TO BOSS
                               ELSE
                                   IF  NAPREQ = 1  THEN   NAPREQ = 0, NAPATV = FF,
                                                          SET NAP MODE IN SYSTEM,  GO TO BOSS
                                   ELSE  GO TO BOSS
END    BOSS
```

CHKDRV Algorithm (both programs)

CHKDRV is called from BOSS to test the state of command execution and if a command is not in process, to return to BOSS; if a command is not active, a command request flag is set. DRVATV also tests the argument of a new position command; if the difference between the new command position is less than 12 counts (Focus, 8 for Rotation) from the previous command position, the command request is ignored and control is returned to BOSS.

If the new command passes the test above, control falls into DRVINT which immediately follows.

```
BEGIN  CHKDRV
       EXTTMR = EXTTMR - 1
        IF   EXTTMR = 0   THEN  INT 3    (causes TMR2 interrupt)
        ELSE
             IF  DRVREQ = 0  THEN  RETURN TO BOSS
             ELSE  DRVREQ = 1
             IF   |CMDTMP - POSCEC|ABS < 12 COUNTS  THEN  RETURN TO BOSS

             ELSE  DRVOFF = 1, DRVREQ = 1
             IF    DRVATV = 1    THEN  RETURN TO BOSS
             ELSE  GO TO DRVINT
END    CHKDRV
```

DRVINT Algorithm (both programs)

DRVINT is entered by being called from BOSS or is entered at the termination of CHKDRV.

DRIVINT is the subroutine which calculates breakpoints for the drive profile. During the drive profile, speeds are ramped up to BREAK1 (position breakpoint 1), held at constant speed until BREAK2 (position breakpoint 2), and then ramped down to a low speed for convergence to the command set point.

In addition to the breakpoint calculations, DRVINT tests for limit faults, the Emergency Stop and Drive Lockout inhibits. In the event that any of these conditions exist, control is returned to BOSS. If the faults or inhibits are subsequently cleared, DRVINT will once more attempt to initiate the execution of the new command. In the Focus version, the SECCH (second chance) flag is tested. If the Focus second screw subroutine has twice detected a failure, control is returned to BOSS and the command is aborted.

DRVINT also tests the position ERROR for the new command argument. If ERROR < 12 counts (Focus, 8 for Rotation), control is returned to BOSS; the new command will not be executed. If ERROR is 12 counts or greater, the breakpoint calculations are performed and control discretes are set to start the drive into motion.

RAMPOS is the drive regime index, it is set to zero in DRVINT.

DRVINT is used for both computer and manual commands. LOCTST is the entry point for manual command initiation.

```
BEGIN  DRVINT
       DRVREQ = 0
       IF  SECCH = 2   THEN  RETURN TO BOSS
       ELSE
           IF  EMERGENCY STOP OR DRIVE LOCKOUT  THEN   RETURN TO BOSS
           ELSE   CLEAR FLAGS,  POSCEC = COMTMP,  CALL DSTOR
LOCTST     IF  SECCH = 2  THEN  RETURN TO BOSS
           ELSE
               IF  EMERGENCY STOP OR DRIVE LOCKOUT  THEN   RETURN TO BOSS
               ELSE
                   IF ERROR < 12  THEN  RETURN TO BOSS
                   ELSE
                       IF  LIMIT FAULT   THEN  RETURN TO BOSS
                       ELSE  CALCULATE BREAK1
                           IF  IN MANUAL   THEN  MAX SPEED =  60 COUNTS
                           ELSE  CONTINUE
                           SCRLST = POSD, POSDED = POSD, SCRCNT = 0, FIL = 0, FILOVR = 0
                           CALCULATE BREAK2
                           IF   IN MANUAL  THEN  BREAK2 = BREAK2 - 65
                           ELSE  CONTINUE
                           CALL BRKON, DAC = CONRAM, ENABLE DRIVE, SPEED = 30, RAMPOS = 0, START TMR2,
                           DRVATV = 1, SET S105 LED'S, EXTTMR = 25, RETURN TO BOSS
END    DRVINT
```

LOCAL Program (both programs)

LOCAL is the program which permits manual slew control of the drive position and permits D/A converter and A/D converter alignment programs to be called to align the S101 D/A and A/D converters.

LOCAL is entered from BOSS and returns to BOSS if there are no manually-commanded actions to be performed. As long as a manual command action is to be performed, control remains in LOCAL.

LOCAL tests a number of control states and manual switches; if a manual slew switch is active, the drive will move in the direction indicated by the switch. When the switch is released the drive is stopped.

LOCAL tests the Mode switch setting; the switch value is converted to an index and control is transferred to the subroutines pointed to by indexing an address in a jump table. These subroutines on set D/A states, select voltages for A/D conversion, etc. Upon completion of these subroutine tasks, control is returned to the LOCAL test sequence.

```
BEGIN  LOCAL
       SET LOCAL FLAG IN SYSTEM
       IF  MANOVR = 1  THEN   IF   DRVATV = 1  THEN  DRVOFF = 1  GO TO LOCAL
                              ELSE   GO TO BOSS
       ELSE  SET SP = STACK
       IF   APEX  OK   THEN   IF   DRVATV = 1  THEN  GO TO  LC1B
                              ELSE  CALL DSTOR,  READ MODESWITCH
       ELSE  GO TO RSCMD
       IF  MODESWITCH IN RANGE  THEN  INDEX = MODESWITCH VALUE * 2
       ELSE  INDEX = 0
       CASE MANTBL[INDEX] OF
           0:   CALL NORMIT; NORMIT TO ZERDA; ZERDA TO SETDA; RETURN
           2:   CALL BRKON; BRKON JUMP TO TMROFF; RETURN
           4:   CALL NEGDA; NEGDA JUMP TO SETDA; RETURN
           6:   CALL ZERDA; ZERDA JUMP TO SETDA; RETURN
           8:   CALL POSDA; POSDA JUMP TO SETDA; RETURN
          10:   CALL NEGAD; NEGAD JUMP TO READAD; RETURN
          12:   CALL ZERAD; ZERAD JUMP TO READAD; RETURN
          14:   CALL POSAD; POSAD JUMP TO READAD; RETURN
       END
LC1B   DEC EXTTMR
       IF  EXTTMR = 0  THEN  INTERRUPT 3
       ELSE
           IF  CW SWITCH = 1  THEN   GO TO LUP
           ELSE
               IF  CCW SWITCH = 1  THEN  GO TO LDWN
               ELSE
                   IF  DRVATV = 1  THEN  DRVOFF = 1  GO TO LOCAL
                   ELSE
                       IF  STILL IN LOCAL  THEN  CLEAR FLAGS, DRVATV = 0, GO TO LOCAL
                       ELSE  GO TO RSCMD

LUP    IF  DRVATV(1) = 1  THEN  IF  DRVATV(2) = 2  THEN  GO TO LOCAL
                                ELSE  GO TO LOCAL
       ELSE  POSCEC = HIGH,  CALL DSTOR
       IF  POSD > HIGH    THEN  GO TO LOCAL
       ELSE   CALL LOCTST,  DRVATV(2) = 2,  GO TO LOCAL

LDWN   IF  DRVATV(1) = 1  THEN  IF  DRVATV(4) = 4  THEN  GO TO LOCAL
                                ELSE  GO TO LOCAL
       ELSE  POSCEC = LOW,  CALL DSTOR
       IF  POSD < LOW,    THEN  GO TO LOCAL
       ELSE  CALL LOCTST,  DRVATV(4) = 4,  GO TO LOCAL
END    LOCAL
```

LOCAL initiates a manual slew command when a manual slew switch is actuated. When either of the previously inactive slew switches becomes active, LOCAL initiates a command sequence by setting either the software HIGH or LOW limits (determined by the sense of the switch) in POSCEC. LOCAL then calls LOCTST. LOCTST is the entry point in DRVINT for initiating LOCAL commands. From this point the drive breakpoints are calculated just as in the computer mode. Upon completion of the calculations, control is returned to LUP or LDWN in LOCAL.

Two subroutines, LUP and LDWN initiate the call to DRVINT and maintain the motion as long as the slew switch remains active.

When a drive is active, LOCAL operates in synchronism with the TMR2 subroutine; all the TMR2 operations of ramping (ZIPUP) up the drive to the main drive (MAINNY) regime, ramping down (ZIPDWN), converging (VERGIT), shutting off the drive (OFFIT and OFFIT3) and MOTION analysis are in operation.

TMR2 Subroutine Algorithm (both programs)

TMR2 subroutine is entered by a Timer 2 interrupt or by Interrupt 3 as a result of EXTTMR being decremented to zero. The cycle rate of TMR2 is 1800 Hz. TMR2 is active only when the drive is in motion and is the entry point for the drive regime subroutines, ZIPUP, MAINNY, ZIPDWN and VERGIT. At 20 cycle intervals of SPEED, when SPEED equals 10, there is a jump to MOTION to analyze drive motion.

Original direction (LDIR) is compared with current direction (DIR) to test for command set point overshoot. TMR2 tests for the presence of the Emergency Stop and Drive Lockout inhibits and limit faults.

Transfers to the ZIPUP, MAINNY, ZIPDWN and VERGIT drive regimes are based upon the value of the RAMPOS index which is changed to the next value at the completion of the ZIPUP, MAINNY and ZIPDWN drive regimes.

The exit for all TMR2 subroutines is TMR2EX which resets the interrupts.

```
BEGIN  TMR2
       IF   DRVOFF = 1  THEN  GO TO OFFIT
       ELSE
           IF  EMERGENCY STOP OR DRIVE LOCKOUT   THEN   GO TO OFFIT3
           ELSE
               IF   LIMIT FAULT   THEN  GO TO OFFIT
               ELSE
                   IF  DIR  NE  LDIR  THEN  RAMPOS = 6   GO TO VERGIT
                   ELSE
                       IF  SPEED = 10  THEN  GO TO MOTION
                       ELSE  SPEED NE 10
                       CASE   RAMPTL[INDEX] OF
                           0: JUMP ZIPUP
                           2: JUMP MAINNY
                           4: JUMP ZIPDWN
                           6: JUMP VERGIT
                       END
END    TMR2
```

ZIPUP Algorithm (both programs)

ZIPUP tests both velocity and position for the transition to the MAINNY drive regime. Velocity is compared with the velocity threshold, RAMPLM and position is compared with position threshold, BREAK1 (breakpoint 1).

SPEED is decremented on each pass and if zero, it is reset to 20 to provide an index for the MOTION subroutine in TMR2.

Details of reading the EPROM1 port states, saving control discretes, altering the DAC levels, merging the new DAC level and outputting the reconstituted state to EPROM1 are omitted.

```
BEGIN  ZIPUP
       BFLAG = 0,  SPEED = SPEED - 1
       IF  SPEED = 0  THEN  SPEED = 20,  DAC = DAC + 4
       ELSE  SPEED NE 0
           IF  DAC > RAMPLM  THEN  RAMPOS = 2, GO TO TMR2EX
           ELSE  DAC <= RAMPLM
               IF  LDIR = 1 (CCW)   THEN
                                        IF  BRK1 <= POSD  THEN  GO TO TMR2EX
                                        ELSE  BRK1 > POSD,  RAMPOS = 2, GO TO TMR2EX
               ELSE  LDIR = 0 (CW)
                   IF   BRK1 => POSD  THEN  GO TO TMR2EX
                   ELSE  BRK1 < POSD,  RAMPOS = 2,  GO TO TMR2EX
END ZIPUP
```

ZIPDWN Algorithm (both programs)

ZIPDWN is the drive ramp-down regime code in the TMR2 subroutine. ZIPDWN compares velocity with the CONRAM threshold for the transition to the VERGIT drive regime. When either the velocity or position comparisons indicate that the threshold has been reached, the drive profile jump index RAMPOS is set to a value of 6 which causes TMR2 to jump to the VERGIT drive regime on the next entry to TMR2. The exit from the TMR2 subroutine is TMR2EX (Timer 2 exit).

SPEED is decremented on each pass and if zero, it is reset to 20 to provide an index for the MOTION subroutine in TMR2.

```
BEGIN  ZIPDWN
       SPEED = SPEED - 1
       IF  SPEED = 0  THEN  SPEED = 20, BX = 4
           IF  LDIR = 1 (CCW)  THEN   BX = - 4
           ELSE  BX = 4, DAC = DAC - BX
           IF  DAC = 0    THEN  GO TO  ZIPDN2
           ELSE
               IF  LDIR = 1 (CCW)  THEN  DAC = - DAC
               ELSE
ZIPDWN2            IF  DAC = CONRAM  THEN  SPEED = 1, RAMPOS = 6
                   ELSE  GO TO TMR2EX
       ELSE  GO TO TMR2EX
END  ZIPDWN
```

MAINNY Algorithm (both programs)

MAINNY is the main drive regime in the TMR2 subroutine which accomplishes most of the drive motion. POSD (current position) is compared with BREAK2 (breakpoint 2), the breakpoint threshold for the transition to the ZIPDWN drive regime. If POSD has reached or exceeded the breakpoint RAMPOS, the drive profile jump index is set to 4 so that on the next pass through TMR2, the ZIPDWN drive profile code will be executed. The logic for comparison of POSD and BREAK2 is conditional upon the direction of motion.

SPEED is decremented on each pass and if zero, it is reset to 20 to provide an index for the MOTION subroutine in TMR2.

The exit for MAINNY is TMR2EX; RAMPOS may or may not be set to = 4.

If moving CW (LDIR = 0), numeric values are increasing so if POSD > BREAK2, the transition point has been reached or passed. If so, RAMPOS is set to 4 and control is transferred to TMR2EX. If POSD < BREAK2, then control is transferred to TMR2EX.

If moving CCW (LDIR = 1), numeric values are decreasing so if BREAK2 < POSD, the transition point has been reached or passed. If so RAMPOS is set to 4 and control is transferred to TMR2EX. If BREAK2 > POSD, then control is transferred to TMR2EX.

```
BEGIN  MAINNY
       SPEED = SPEED - 1
       IF  SPEED = 0  THEN SPEED = 20
       ELSE
           IF  LDIR = 1 (CCW)  THEN
               IF  BREAK2 > POSD  THEN  GO TO TMR2EX
               ELSE  RAMPOS = 4,  GO TO TMR2EX
           ELSE  (CW)
               IF  POSD > BREAK2  THEN  RAMPOS = 4, GO TO TMR2EX
               ELSE  GO TO TMR2EX
END    MAINNY
```

VERGIT Algorithm (both programs)

VERGIT is the convergence drive regime code in the TMR2 subroutine. With the drives moving at a slow speed, VERGIT tests position ERROR; if less than 4 counts the DAC output is set to zero and control is transferred to OFFIT3 to turn off the servo amplifiers, engage the brake, etc.

If ERROR is greater than 4 counts, VERGIT tests for a drive overshoot by comparing initial direction (LDIR) with current direction (DIR). If they match, the drive has not overshot the command set point and the DAC drive is reduced by 4 counts. If they differ, the drive has overshot the command set point so the DAC is reduced by 4 counts and tested for zero. If finally zero (after one or more passes through TMR2), the second try flag DRVONE is tested. If the drive over shot the second time, a DRIVE FAULT flag is set in FAUL1. If the second try flag is not set, a new command to the set point is initiated.

```
BEGIN  VERGIT
       IF  ERROR < 4 COUNTS  THEN  DAC = DAC - 4
           IF  DAC = 0   THEN   CLEAR DRIVE FAULT FLAG IN FAUL1,  GO TO OFFIT3
           ELSE   GO TO TMR2EX
       ELSE
           IF  DIR = LDIR  THEN   GO TO TMR2EX
           ELSE   DAC = DAC - 4
               IF   DAC = 0  THEN
                   IF  DRVONE = 1  THEN  SET DRIVE FAULT IN  FAUL1,  GO TO OFFIT3
                   ELSE   DRVONE = 1, DRVATV = 0, DRVREQ = 1, LOAD COMMAND ARGUMENT,  GO TO OFFIT3
               ELSE  GO TO TMR2EX
END    VERGIT
```

OFFIT Algorithm (both programs)

OFFIT is one of the TMR2 subroutines called from a number of the TMR2 subroutines and is used to slow down and shut off a drive. OFFIT has two entry points; the OFFIT entry point causes the drives to be slowed to zero speed over a number of cycles through OFFIT. An example of the use of OFFIT is when a new, overriding command to a different set point has been received while a command is being executed. In this case OFFIT gradually slows the drive to a safe stop before the new command execution is started.

The OFFIT3 entry point causes the drive to be stopped immediately and the brake engaged. The timers are turned off. The OFFIT3 entry is used when the drive is moving very slowly such as in VERGIT, or when there is a suspected fault and the best course is an immediate stop. An example of this situation is the detection of a second screw fault in the Focus Second Screw Algorithm.

An important entry point to OFFIT is the TMR2EX entry which causes the EOI code to be sent to the IPI registers to reset the interrupts for another pass through TMR2 code. The TMR2 entry point is used as an exit by all the TMR2 subroutines.

```
BEGIN   OFFIT
        READ DAC,  SPEED = SPEED - 1
        IF SPEED = 0  THEN  GO TO OFFIT2
        ELSE  DAC = DAC - 4
OFFIT2  IF  DAC = 0
OFFIT3              THEN  TIMER OFF, DAC = 0, EOI = 8, BRAKE OFF, DRVATV = 0,  DRVOFF = 0
                          CLEAR DISPLAY LED'S,  GO TO TMR2EX
TMR2EX  ELSE  EOI = 8, INTERRUPT RETURN
END     OFFIT
```

MOTION Algorithm (both programs)

MOTION is the algorithm which detects dragging and sticking of the drives, a constant concern for lubricated mechanisms which are subject to cold environments. Dragging or sticking could also result from increases in mechanism friction. MOTION is an important program feature designed to protect the delicate and expensive FRM.

MOTION is entered from the TMR2 subroutine each time the SPEED counter reaches a value of 10; this is the mid-point for velocity changes in ZIPUP or ZIPDWN which occur when SPEED is decremented to zero. The exit from MOTION is TMR2EX.

MOTION analyzes drive motion by comparing position values derived from velocity-time products with direct delta positions. The position and velocity samples are taken at 20 Speed cycle intervals which is an 11.1 millisecond period.

SPEED is decremented in MOTION to maintain a constant rate of change in SPEED since the jump to MOTION is an alternate to jumps to ZIPUP, MAINNY and ZIPDWN.

```
BEGIN  MOTION
       READ CURRENT VELOCITY,  SPEED = SPEED -1
       IF  SERVO FAULTS  THEN  DRVOFF = 1   GO TO TMR2EX
       ELSE  |POSD - POSDED|ABS

       DELPOSNEW = CURRENT VELOCITY/20

       READ  DELPOSOLD,  SAVE  DELPOSNEW

       IF  (|POSD - POSDED|ABS + 4) LE |DELPOSNEW + DELPOSOLD|ABS  THEN  SET MOTION FAULT,  GO TO TMR2EX

       ELSE  GO TO TMR2EX
END    MOTION
```

SECOND SCREW POSITION ANALYSIS, TMR21B (Focus only)

This code, near the beginning of Focus TMR2, analyzes motion of the second screw in terms of position change since the last transition of the PHASEA discrete. The position of the transition SCRLST (screw last) is the position reference and the position change from this reference is SCRCNT (screw count). On each pass through this code (at the TMR2 rate of 1800 Hz), the position change since the previous pass is saved in DELNEW. The DEL value from the previous pass is DELOLD. If the DELNEW change is greater than 12 counts, the previous pass change DELOLD is added to SCRCNT rather than the DELNEW value. In this event, a digital filter FIL is set to 1 and a filter pass counter FILOVR is incremented. If the count reaches 10 (during the pass), a FAUL1 flag is set and the drive is quickly shut down by a jump to OFFIT3. The recurrence of erratic DELNEW values in a given transition state suggests that something is erratic and as a precautionary measure, the Focus drive should be immediately shut down. The SECCH (second chance) flag is not set.

On each pass, DELNEW is saved in DELOLD for substitution for DELNEW if the change is greater than 12 counts. If SCRCNT exceeds 50 in any pass since a transition, it indicates that the second screw is lagging the first screw by at least 40 degrees which is the tolerance limit. In this event a FAUL1 flag is set, the SECCH flag is set and the drive is shut down by a jump to OFFIT which gradually slows the Focus drive to zero and clears DRVATV (drive active).

The normal in-tolerance exit is TMR2EX (Timer 2 exit).

This subroutine will be bypassed if the SCWIGN (screw ignore) flag is set.

POSD is the current drive position.

```
BEGIN  TMR21B
       IF  SCWIGN = 1 THEN GO TO TMR22
           IF  FIL = 1 THEN  GO to TMR220
           ELSE GO TO TMR2EX
       ELSE
           IF TRANSITION = 1  THEN  SCRCNT = 0
               IF  FIL = 1    THEN  GO TO TMR220
               ELSE  GO TO TMR2EX
           ELSE  DELNEW = POSD - SCRLST,  AX = DELNEW
               IF  DELNEW > OR = +/- 12   THEN  FIL = 1, AX = DELOLD,  FILOVR = FILOVR + 1
                   IF  FILOVR = 10  THEN  FAUL1 = A00H,  GO TO OFFIT3
                   ELSE  continue
               ELSE  FIL = 0, SCRCNT = SCRCNT + AX,  SCRLST = POSD
           IF  SCRCNT < 50  THEN
               IF  FIL = 1  GO TO  TMR220
               ELSE  GO TO TMR2EX
           ELSE  FAUL1 = 8200,  SECCH = 1  GO TO OFFIT3
END    TMR21B
```

3.8 CONTROL FIRMWARE DESCRIPTION

Introduction

The F-R Controller described in Section 3.1 executes the logical operations determined by the control firmware. Section 3.7 describes the control algorithms that are the essence of the operations performed by the F-R Control System, unencumbered by details of the mechanism, electronics or firmware. In this section, we address the control firmware that is the control logic for the system. We will describe the way the firmware executes these algorithms with the emphasis on the firmware operations, not the control algorithms. In short, this section describes the what, how and why of the firmware.

The description emphasizes the effect of the program operations upon the FRM and control system; is not restricted to just microprocessor code operations.

The control firmware programs described are the December 7, 1989 versions.

The ground-rules followed in the description are: (1) This is an F-R System manual, not a tutorial on 80188 assembly language programming; for information on the 80188 instructions the reader is referred to the Intel 80188 Programmer's Reference Manual (IPRM); (2) An AD 2500 Cross-Assembler running in an IBM-compatible PC was used to assemble and link the assembly language programs; this program uses pseudo-instructions which are interspersed through the listings. Superficially, these pseudo-instructions may appear to be actual 80188 assembly language instructions; however, since these pseudo-instructions influence the firmware code, they are important aspects of the programs. This manual will describe only those pseudo-instructions actually used in assembling the control firmware. For additional information on these pseudo-instructions or other details on the assembler-linker, the reader is referred to the AD 2500 manual. (3) All features of the firmware are described in detail, at a level sufficient to convey the evolving states in the 80188 registers, decision logic, input conditions and output conditions. The description does not simply paraphrase the firmware comments (which are very important aspects of program documentation) but explains the operations performed in the machine code. (4) The machine instructions are not included in this description although in some cases a few instructions are repeated in the text. The reader should frequently refer to the algorithms in Section 3.7 and the program listings in Section 7.0. The reader should understand the code under discussion. (5) It is necessary to study the architecture and instructions of the 80188 and understand major features of the processor. Examples of major features are the register structure and usage, Segment Register usage, the meanings of Address Offset, Logical and Physical addresses, the BIU and EU, Direct, Indirect and Indexed addressing, the DMA channels and controller, the Counter/Timers and controller, the Interrupt system and controller and many other features. This manual alone is not adequate to describe the 80188 firmware; the reader should have the hardware and programmer's reference manual at hand as he reads the following discussion. (6) Controller Check-out programs which operate the microprocessor for system and module test purposes will not be described.

First, a few words about the conventions followed in the program descriptions. The two control programs described are: Focus and Rotation. These programs are very similar but differ in a few particulars. The dissimilarities are so few and distinct that one description will suffice for both. These dissimilarities are described as they are encountered in the narrative and are treated in a parenthetical manner so that the reader is always aware of their application.

The description is a narrative commentary on the operations of the program. At intervals we will comment upon the op codes as they apply to the program execution; the commentary will not be repeated for instruction types which have been previously discussed unless there is some serious or subtle implication. It is not feasible to comment upon the op codes of every instruction but the description

treats just about all types seen in the program to show the inner processes of 80188 program execution.

The program labels, registers and operands are printed in UPPER CASE text. To enable the reader to quickly locate the code under discussion, some line numbers are included in the text as parenthetical notes delimited by square brackets and printed as bold underlined numbers; an example is: [F300] which designates Focus code, line 300. The line numbers and upper case labels, registers and operands enable the reader to easily correlate the descriptions with the program listings in the Appendix (Section 7.0).

Pages in the upper code listing are referenced by: RUM n, and FUM n where R and F denote Rotation and Focus respectively; U denotes upper memory, n is the page number. The lower memory sections are referenced by RLM n and FLM n where L denotes lower memory.

Addresses and values (unless stated otherwise) are always Hex numbers.

An important consideration in the two programs is the numeric representation of operands. The Intel Microsystems Components Handbook, Volume 1 1986 shows the format of 80188 data types. Two 16-bit numeric value formats are used in the control programs: the Unsigned Word and Signed Word formats. The Unsigned Word format represents 16-bit values ranging from 0000H to FFFFH (0 to 65536 Decimal); negative values do not exist in this format. The Signed Word format permits signed representations of values ranging from minus 32768 to plus 32767. The most significant bit is the sign bit and the least significant 15 bits represent the value. This latter format is sometimes referred to as 2's complement format. The reader should study the rules for transformations between these two formats.

String manipulation instructions are used to clear the Flag table but there are no string variables in the firmware.

Command arguments from the Antenna Control Computer are Unsigned Word values. All program arithmetic operations (except for the use of the BOUND instruction) use the Unsigned Word format; examples are drive positions, command echo's, temporary position values, monitor data position readouts, etc. Apex analog data and the controller analog monitor data (addresses 00H through 0FH) are acquired and output to the Antenna Control Computer in the Signed Word format. The BOUND instruction may use the Signed Word format in limit-testing operands which may have negative or positive signs. Positive Unsigned Word format values up to 32767 may also be used with the BOUND instruction without error. Because the BOUND instruction is so widely used with differing operands, there may be transformations between the two formats before and after the usage of the BOUND instruction.

Each program consists of upper and lower EPROM program memory sections; this is a consequence of the 80188 design. Program execution begins in upper memory after processor reset and the interrupt service code operates from lower memory; control operations are performed by code in both memories. Control transfers between these sections are interrupts, interrupt returns and long jumps. The RAM is used for tables of variables and states and for the processor STACK.

The software upper and lower limits (i.e. HIGH and LOW) deserve special mention. These values are the upper and lower bounds of computer and manual drive; program logic does not permit drive past these points. These values are also the command set-points for manual position slew commands. HIGH is F200H (61952 decimal) which is about 1417 counts below the 1st UPPER (or CW) limit switch. LOW is A00H (2560 decimal) which is about 750 counts above the 1st LOWER (or CCW) limit switch.

Now, a word about addresses and Intel conventions in the program listings. The left-most column is a line number, a listing convenience item having no significance in the code execution.

Introduction (continued)

The next two columns are the Code Segment and Offset values separated by a colon; these are the Logical Address. In the 80188, the instruction Physical Address (Intel terminology) is determined by left shifting (by 4 bits) the contents of the (Code, Data, Extra Data or STACK) Segment Base Register and adding it to the Offset. This 20-bit resultant Physical Address is then the actual address used by the code.

A special case second column is the IP:CS column in which the contents of these two registers are printed out. This is an infrequent column, used with tables (described below) and interrupt vector tables in lower memory. An example is 9D00:0000 which is the TRAP 5 interrupt vector found on [F271] in FLM 5. The IP column is 9D00 which designates the 009D offset for the TRAP 5 service code at [F329] on FLM 7. The CS column is 0000; the code is executing in lower memory.

The third column of values is the machine code instruction consisting of one to six bytes. These are instruction code values shown in the Intel Programmer's Reference manual (IPRM).

The fourth column of values is the label, the fifth is the instruction mnemonic, the sixth is the affected register or registers, and the seventh is the operand, and the final column, (prefixed by a semicolon) is the comment column. There is a lot of variation in the register and operand columns as a function of the instruction type.

The listings begin with several sets of tables that neatly organize program parameters into easily recognizable symbolic form. Placing these parameters at the beginning of the program listing simplifies reference and program changes. This is preferable to distributing these parameters through the listings. These tables are: IPI control registers, initial IPI values, I/O Port addresses, monitor data storage memory locations, temporary storage for variables, flag values and program equates. The first table contains equates for the IPI Interrupt Control/Status registers which condition the 80188 Interrupt Controller. The second table has equates for the IPI Timer control registers which condition the 80188 Timer Controller. The third table has equates for the IPI I/O Chip select registers. The fourth table has equates for the IPI memory chip select registers. The fifth table has equates for the IPI DMA control registers which condition the 80188 DMA channels. The sixth table has the equates for the IPI Relocation register which is the reference address for all IPI registers. The seventh table has equates for the IPI values of memory size and starting address. The eighth table are equates for 80188 I/O ports. The ninth table is Monitor Data storage locations in RAM memory. The tenth table contains Apex Analog data storage locations in RAM. The eleventh table contains system states in RAM memory. The twelfth table contains calculated temporary storage values in RAM. The thirteenth table has flag values in RAM. The last table has program equates.

The flag value table contains control discretes which have a 1 written into the lsb to represent a true or set condition. The two exceptions are RAMPOS and DRVATV. DRVATV has three flag bits. The 1 bit indicates that the drive is running, in both computer control and manual modes. The 2 bit indicates that the UP (CW) manual command switch on S105 is actuated. The 4 bit indicates that the S105 DOWN (CCW) manual command switch is actuated. These two DRVATV manual command bits are only active in the manual mode and are cleared in the computer control mode. The RAMPOS flag has values of 0, 2, 4 and 6 which indicate operation in the Ramp-up, Main drive, Ramp-down and Converge drive regions, respectively. RAMPOS is used in both computer control and manual control modes.

The list of PROGRAM equates (FUM 5 [F244]) is a table of symbolic labels which are equated to values. This table is listed separately from the other equates because they are more general references such as numeric values or other parameter which are frequently referenced in the course of program development or maintenance.

Section 3.9 contains tables which describe the command and monitor data message formats. These tables show addresses, bit assignments, analog and digital signal names, fault and status bit assignments, etc. At this point, the reader should turn to these tables and study the character of this data. It is particularly important to note the composition of four status words: SYSTEM, FAUL1, FAUL2 and ANAFL. These parameters are very important; their states determine the course of program execution. SYSTEM indicates system command modes and the brake state. FAUL1 indicates the following faults: S102, limit switch, command and monitor argument range, operator error, brake, drive and 2nd screw motion. FAUL2 indicates the servo amplifier mode and fault states. ANAFL indicates Apex analog data faults. During the course of study of the firmware description, the reader should frequently refer to these status and fault words. During the course of subroutine and control program execution, these bits are set or reset to serve as system state references. These fault and status words are tested to determine if it is safe to initiate a control action. If a fault state occurs during program action, the control programs and subroutines bring the drive to a safe condition. A second, and equally important usage of these status and fault words is to provide diagnostic and system state information to the Antenna Control Computer via the monitor data. Frequent references are made to the states of these four status and fault words during the course of program description.

The F-R control program makes extensive use of subroutines to reduce the amount of in-line code; this greatly simplifies code development and maintenance. After the operation of a subroutine is debugged and verified, the subroutine may be confidently used in many locations throughout the program. A frequent function of these subroutines is interaction with the system hardware; they read FRM data and states and output control values and states. Subroutines may call other subroutines. Some subroutines are used in both upper and lower program memories; these common subroutines will be described only once because the operations are identical.

The two EPROM program segments for a microprocessor controller are assembled independently; linkage facilities for control transfers between the segments are provided by the linker in the assembler. GLOBAL and EXTERNAL pseudo-Instructions on Page 5 provide the linkage references. Each segment's linker pseudo-instructions have label references for the other segment. The sets are complementary; that is, the GLOBAL label references of one segment are the EXTERNAL label references in the other segment. For example on FUM-5 [F255] GLOBAL references INITIAL and RSCMD are entry point parameters to be passed to the lower memory program. In FLM 5 [F261] EXTERNAL references INITIAL:FAR and RSCMD:FAR. The FAR suffix refers to the type of program transfer (i.e. near vs. far calls, etc.). The reader is referred to the AD2500 manual for a description of the properties of the assembler-linker. The GLOBAL and EXTERNAL linkage references are mentioned here to cite their usages.

Intel data books describe CPU register usage; the Intel description should be carefully studied because the registers are the primary address references. The Intel description will not be repeated here, but it is appropriate to comment upon the usage of these registers in the context of the F-R control firmware. The CS register is the address reference for the program EPROM memories; all program instructions reference the CS register. The RAM is referenced by the DS and SS registers. The DS register references variables, tables of data, flags, program states, temporary values, etc. - all of which are written to or read from the lower address of RAM. The SS register references the STACK segment which is the upper portion of RAM. The SP register accesses the top RAM addresses and is decremented as data is pushed onto the STACK. As data is popped off the STACK, the addresses are incremented. The SP is initialized to the highest RAM address. The SI and DI registers are used with string operations in the firmware. The BP and ES registers are not used. The control firmware does not change the settings of these registers from the initial values set by the processor reset. As program control transfers between the upper and lower program memories, the CS register is set to 0000H and FF80H values to reference the lower and upper EPROM memories.

Monitor Data Readout

Although the bulk of the firmware is devoted to controlling the position of the drives, an equally important function is reading out monitor data upon request from the Antenna Control Computer. This diagnostic data is indicative of the state of the FRM components and the execution of the control Monitor programs. Parameters such as the four status-fault words (mentioned above), drive position, command echo, position error, drive velocity and Apex analog data are read out in the monitor data. The NMI subroutine (described below) processes monitor data requests from the Antenna Control Computer. Section 3.9 lists these monitor data, addresses and ranges.

The initialization code and subroutines are described first because the functions are more narrowly focused than the control programs and they tend to be self-contained entities with smaller sets of input-output parameters. These descriptions prepare the reader for the control program functions because he has become familiar with many of the code operations; with this background the reader will find the control program description easier to understand.

Program Overview

The F-R control firmware may be considered to consist of two parts: asynchronous code and synchronous code. The asynchronous code is always running and is primarily concerned with antenna control computer communications and maintaining supportive conditions for the operation of the synchronous code. The asynchronous code is only indirectly linked to drive motion. The synchronous code, the TMR2 subroutine, is normally quiescent and is set active by the asynchronous code to control drive motion. TMR2 is synchronous because it is driven by the Timer 2 interrupt. Linkages between TMR2 and the asynchronous code are primarily via flags.

Because the F-R Control firmware is largely interrupt-driven, high-level flow charts are not very useful. The algorithms of Section 3.7 are an alternate to flow charts. The program components are mostly called or interrupt subroutines of a fairly straightforward character with simple linkages; very few parameters are interchanged between subroutines. There is some high-level control sequencing which is structured to logically order the execution of control tasks. Time and the antenna control computer drive the interrupts. Time is the dominant parameter in controlling FRM motion. The controller must quickly respond to the antenna control computer commands and monitor data requests. The following paragraphs are an overview of the program control and interrupt-driven sequencing.

The first program action following the processor reset is the execution of the INITIALIZATION code which sets up the memory and I/O chip selects, loads the appropriate control states in the processor IPI, segment and STACK registers, sets up the EPROM and RAM I/O ports modes, clears RAM memory, initializes the control discretes states in the EPROM and RAM control ports, enables the interrupts, resets and inhibits the servo amplifier's control logic and turns on the servo amplifier's three-phase AC power. Having accomplished INITIALIZATION, control is transferred to the BOSS program which is the executive control program.

BOSS is a short executive control program which recurrently sequences through the following actions. BOSS executes a RANDOM delay, resets the STACK pointer, tests Apex faults, checks to see if a drive is active; if active, CHKDRV is called. If the drive is not active, it tests mode switches, calls for Apex data via the DSTOR subroutine, tests for the existence of the manual command override mode, tests the manual command to the LOCAL mode and if set, branches to LOCAL. If there is not a request to branch to LOCAL, BOSS tests requests for servo amplifier reset commands, soft reset commands, NAP mode commands and computer-generated position commands. In servicing these command requests, BOSS resets the servo amplifier(s), executes the soft reset command, calls the DRVINT (drive initiate) subroutine and sets the NAP mode flag.

Program Overview (continued)

DRVINT performs calculations of drive ramping parameters, sets the drives into the DRVATV (drive active) mode and returns control to BOSS.

CHKDRV (called when the drive is active) tests the state of a software timer, calls TMR2 if required by the software timer, tests for the existence of new overriding commands or a command to a set-point too close to bother with.

In both the computer and LOCAL modes, when the drive is active the TMR2 (Timer 2) interrupt-driven subroutine manages all drive motion, acquires Apex data, tests for faults and terminates command execution when the command set point is reached or a fault condition is detected. If a fault condition is detected, TMR2 quickly shuts down the drives by calling OFFIT3. If an overriding command requires the drives to move to a different set point than the one current target set point, TMR2 slows down and shuts down the drives by a call to OFFIT. TMR2 operates at an 1800 Hz rate generated by the processor timers. TMR2 operates with both computer-generated commands and with LOCAL to service manually-generated commands. Although it is not complicated, because of the many things which must be controlled and tested to move the delicate and expensive FRM, TMR2 is the most important and critical component of the control firmware. In the intervals between the execution of TMR2, BOSS continues to sequence through its tasks as described above.

In the LOCAL mode, control inputs are taken from manual position slew switches to move the drives or from mode switches to effect some adjustment to the controller circuitry. LOCAL is entered from BOSS by a test of the CMP-MAN switch. Upon entry to LOCAL, the manual command override flag is tested to see if the antenna control computer has overridden manual control. If not, the Apex fault status is tested; if it is OK, DSTOR is called to obtain a fresh set of Apex data. The mode switches are tested next to see if an adjustment command is requested. If so, the switch value is used to develop an index for the jump to the adjustment subroutines. Upon return from the adjustment routine (if requested), the manual slew switches are tested to determine if a new motion command is requested, current motion is to be sustained, stopped or the direction is to be reversed. If a new motion command is to be initiated, DRVINT is called to initialize the ramping parameters. Two LOCAL branches deal with the UP or DOWN slew switches. Upon return from the DRVINT initialization, control remains in LOCAL as long as a slew switch is actuated. When released, control returns to BOSS for another scan. Management of the drive motion is done by TMR2 as in the computer mode.

DSTOR is an important special purpose subroutine which is called by all the major program components mentioned above. DSTOR requests Apex data, tests for its arrival and when present formats and stores position, velocity, discretes and analog data and sets fault flags. DSTOR also acquires brake, DRIVE LOCKOUT and EMERGENCY STOP status. In short DSTOR acquires all the requisite information about the FRM and system status for the use of BOSS, DRVINT, CHKDRV, LOCAL and TMR2. A short version of DSTOR resides in upper memory for quick acquisition of position and limits data.

A number of small, special-purpose subroutines are called by the major program components above. These are: BRKON (brake on) turns on the brake. BRKOFF turns off the brake. DELAY executes an N-second delay, the N argument is conveyed to the subroutine by the contents of CX. TMROFF turns off the timers and reinitializes the timer control registers. BDSRS resets the servo amplifiers. SEC1 executes a 1-second delay using Timers 1 and 2. INT0 reads and formats the A/D converter data. BOUND responds to the BOUND instruction interrupt to set a flag indicating that the array-tested parameter exceeded the limits. Most of these routines are used in lower memory but some are used in both upper and lower memories.

Program Overview (continued)

Data requests from the antenna control computer are serviced by the NMI (non-maskable Program interrupt) subroutine because the Standard Interface imposes response timing constraints for acquisition of this data.

New commands from the antenna control computer are unobtrusively input via the DMA0 channel and their arrival is signaled by the DMA0 interrupt. In response to the interrupt, the DMA0 subroutine sets command request flags for the various types of commands and reinitializes the DMA0 logic.

In general, program components associated with initialization, supervisory scanning and task scheduling, preparation for command execution, control of peripherals, etc. are located in upper memory. Program components associated with interrupts or closely related to interrupt functions are located in lower memory. Some components are located in both memories for expeditious linkage. The following table shows the memory assignments.

LOWER MEMORY:					DSTOR*			BRKOFF	
UPPER MEMORY:	INITIAL	BOSS	CHKDRV	LOCAL	DSTOR	BDRSRS	BRKON	BRKOFF	CMDRS
LOWER MEMORY:		SEC1	TMROFF						
UPPER MEMORY:	DELAY	SEC1	TMROFF	NMI	BOUND	DMA0	INTO	TMR2	INT VECTORS

* Lower memory DSTOR is a brief version of the upper memory DSTOR.

INITIAL (FUM 5, RUM 5)

INITIAL establishes the I/O chip selects and initializes the DMA, Timer and Interrupt control registers in the Internal Peripheral Interface (IPI). The RAM is also cleared, i.e. all memory locations are set to a value of zero. The 80188 emerges from the hardware reset with the RELOCATION REGISTER set to 20FF and the INSTRUCTION POINTER set to execute the instruction at Physical Address FFFF0 (RUM-18). Because we have just started our code description, the initialization code description will be a bit more detailed than the case for subsequent code. Once we see the nature of the operations, it will not always be necessary to repeat the meaning of some of the control states in the code.

First a word about the initial state of important CPU registers. The processor emerges from the processor reset state with these registers initialized as follows: CS = FFFFH, DS = 0000H, IP = 0000H, SS = 0000H, ES = 0000H, REL REG = 20FFH and Status Word = F002H. When the program operates in upper memory, the Code Segment register (CS) must be set to FF80H; the Data Segment (DS) must be 0000H (its initial value); and the STACK Segment (SS) must be 0000H (its initial value). When the program is operating in the lower memory, the CS must be set to 0000H which is accomplished by the INT (interrupt) instruction. During the course of execution, the INT instruction pushes the CS, IP and FLAGS registers onto the stack. The IRET (interrupt return) instruction restores the CS (and IP and FLAGS registers) to the FF80H value by popping this value off the STACK into the CS register.

Consider the addresses of the upper EPROM when the 80188 emerges from the reset state. The first code execution takes place at Physical Address FFFF0H; the first few instructions initialize the processor conditions. After reset, the initial state of the CS and IP registers is: (CS) = FFFFH; (IP) = 00000H. This is a physical address of FFFF0H; remember, Physical Addresses are formed by left shifting (CS) four bits and adding the logical (offset) address. In this case, the (IP) = 0000; thus FFFF0H + 00000H = FFFF0H, the Physical address of the first instruction.

INITIAL (continued)

The 2-kbyte EPROM has an address range of 0 to 7FFFH; for convenience in developing code, it is desirable to arrange the addressing so that EPROM addresses coincide with the program offset values. Reference to the Memory Map of Section 3.1 shows that the upper EPROM physical addresses range from FF800H to FFFFFH. FF800H is the base (i.e. lowest) physical address of this memory (coinciding with an EPROM address of 0000H). The upper memory code is referenced to this base value by the ORG 0FF80:0000 pseudo-instruction [F260] on FUM 5. Remember, these ORG pseudo-instructions are assembler contrivances which locate code at specific addresses. This pseudo-instruction causes the (CS) to be FF80H. The ORG 07F0H pseudo-instruction ([993] on FUM 18) references the first initialization instruction at offset address 07F0H. This 07F0H address is also the initialization physical address FFFF0H, i.e. the address of the first instruction executed after processor reset. How can this 07F0H be FFFF0H? Remembering the upper EPROM Physical addresses are (CS), left shifted four bits and added to the offset, we get: FF800H + 07F0H = FFFF0H, the initialization physical address.

RAM memory accesses use physical addresses formed in the same way as the EPROM instruction memories but instead of using the (CS) values; the physical addresses are formed by left shifting the (DS) and (SS) values which are added to the offset address. Any memory reference instruction which puts data into or out of a memory location uses the DS or SS register. The DS and SS registers are left at the power initialization values of 0000H; they are not changed during the course of program execution.

This initialization code is very important; if incorrect, the processor will not execute the instructions in a manner consistent with the expected operation of the control firmware. Secondly, it is very important that the FRM mechanism drive circuitry be initialized to safe conditions.

The first instruction [F997] (FUM 18) [R981] (RUM 18) loads the address 0FFA0 (equivalent to the symbol UMCS) into the DX register. 0FFA0 is the address of the upper memory chip select storage register in the Internal Peripheral Interface. The address stored in this location defines the lower address boundary of the upper memory chip select (80188 term FUCSbar). The next instruction loads the value 0FFBD into the AX register. 0FFBD is the lowest address of the 2 kilo-byte upper memory EPROM. The OUT instruction loads the value contained in the AX (0FFBD) register into the location pointed to by the contents of the DX register (0FFA0). 0FFBD is a little different from the 0FFB8 address cited in the paragraph above; why the extra bits? The three lower bits are the wait state programming bits for this memory that establish the wait state conditions to be used in reading this memory. In this case, the 101 (lower 3 bits) code specifies 1 wait state and that the Synchronous Ready input to the 80188 is to be ignored. All three memories, i.e. the 8755's and 8156, use the same wait state conditions.

The operation of this first (it must be done first) bit of code is to change the lower bound of the upper memory from 0FFF8 to 0FFB8, which is the value for a 2K memory (it was initialized for a 1K memory by the processor reset).

Having set up the upper memory chip select, the program transfers control to the lower portion of the upper memory to the location INITIAL on RUM-5, [R263]. Note that this is an unconditional FAR jump with the machine code of EA 00 00 80 FF. This is an inter-segment (internal to the existing CS, i.e. upper memory) direct jump to address FF800 (FF80 + 0000). The JMP FAR two 00 00 (offset) bytes are zero because the ORG pseudo-instruction (ORG 0FF80:0000H) set the CS to FF80 and offset to 0000. The program operations described above must not use more than 16 bytes of memory storage.

At INITIAL (RUM 5) [R263], the initialization continues with a series of MOV DX operand, MOV AX operand, OUT DX,AX sequences that are identical to the upper memory chip select initialization described two paragraphs above. The sequence extends to page RUM-6. The stack pointer is first initialized to the top RAM address by the instruction which loads the SP with the value equivalent

to the symbol STCK (2100). The initialization continues with the set-up of the lower, middle memory and I/O chip selects; initialization of the 8755 Data Direction Registers and the 8156 Control Register. The states stored in the 8755 Data Direction Registers and the 8156 Control Register determine the mode of the I/O Ports pins, i,e., they are assigned to be either input or output. [R283] Since the AX register (16-bit word value) is being output, both Port A and Port B Data Direction Registers are loaded with one instruction because the 8755 DDR's have contiguous addresses. Both ports of PROM1 DDR are programmed for output by the 0FFFFH value. PROM2 Port A DDR is programmed to output; Port B DDR bits 0, 1, and 2 are programmed for output; and 3 through 7 are programmed for input by the value 07FFH. The RAM control register (address equivalent to the symbol RAMTC) is loaded with contents of AL which are FFH from the instructions above. This sets Ports A, B and C to the output mode. Reference to the 8156 data sheet command register bit assignments shows that the FFH value causes the 8156 timer to be started (it has no clock and the output is floating) and the Port A and B interrupts are apparently enabled. The latter two are actually null commands since they are used with Port C bits in another mode.

The XOR AX,AX instruction [R280] clears the AX register to 0000. This value is output to PROM1 Ports A and B and then to the PROM2 ports. Clearing PROM1 Port A bits and Port B bits 0, 1, 2 and 3 sets the inputs to the D/A converter latches to zero (but the D/A latches were cleared during processor reset). PROM1 Port A bits 4, 5 and 6 causes the Focus motor brake SSR drive to be inhibited and the Focus Servo Amplifier (BDS3) to be inhibited and reset. Clearing PROM2 Ports A and B turns off the S105 front panel LED's and sets the Focus A/D Converter multiplexer address to be zero.

The address in DX is incremented which points it to RAM Port A. Ports A and B are cleared by the OUT DX,AX instruction. DX is again incremented which then points to RAM Port C which is cleared by another OUT DX,AX instruction. Clearing these three RAM ports causes the S101 front panel to display zeros. The instruction pair MOV DX,SEMCT and OUT DX,AL causes strobe/enable 403 to go true for the duration of FWRITEbar. This function is not used.

We now address the initialization [R301] (RUM-6) of DMA channel 0 in which the DMA control registers in the Internal Peripheral Interface (IPI) must be set to the appropriate states for the desired DMA mode. DMA 0 is the channel which inputs four command bytes in S104 to four addresses in the RAM when the S104 receives a new command from the Antenna Control Computer. This instruction sequence uses string instructions to transfer the contents of a table of control values to the DMA 0 control registers in the IPI. The instruction sequence begins with the instructions MOV SI, OFFSET DSETUP. SI is the Source Index register which points to the first word or byte of a string. OFFSET is an assembler pseudo-instruction which causes the address (rather than the contents) of DSETUP to be loaded into the SI register. DSETUP is the label of a six-word table (the table is found on FLM 16). The .WORD pseudo-instruction causes a word value equivalent to 395 in the equates table to be assigned to two consecutive addresses. RAC is a 1-.BYTE address (in the RAM) in the lower EPROM which contains the values to be loaded into the DMA 0 control registers. The next instruction MOV DX, DOSPL causes the address equivalent to DOSPL (0FFC0H) to be loaded into DX. MOV CX, 6 loads 6 into the CX register to serve as a loop counter. OUTSW is a string word instruction that outputs the value contained in the first word (DSETUP) of the table to the IPI register address DOSPL, which is the DMA 0 Source Pointer LSB register. After the transfer, SI is updated to point to the next string element. After this transfer, the ADD DX,2 instruction, the DX register contents (i.e. address of IPI table) are increased by 2 to point to the next DMA 0 control register. The LOOP instruction decrements CX and transfers control to the location DSET if CX is not zero. In this loop, the six values required for the required mode of the DMA 0 are loaded into the IPI DMA 0 control registers. These registers and values are: Source Pointer LSB (DOSPL) - 395 (RELADD); Source Pointer MSB (D0SPM) - 0; Destination Pointer LSB (D0DPM) register - 2040 (RAC); Destination Pointer MSB (D0DPM) - 0; Transfer Count

Register (D0TC) - 4; and finally, the DMA 0 Mode Control Register (D0MODE) - 0A766 (D0VAL). In this code we set up the DMA 0 data source to be the Command Relative Address register in S104 (which is read by the 395 enable); cause the data to be transferred to the RAC destination; and enabled four transfers to take place so that when a position command is received by the S104, its DRQ 0 will initiate a DMA transfer sequence from the four contiguous S104 source addresses to four contiguous destination addresses in the RAM. This set-up enables the DMA 0 channel to automatically load the four command arguments into the RAM when the Antenna Control Computer outputs a command to the F-R Control System.

The timers are initialized next by a CALL TMROFF instruction (see the TMROFF Subroutine description below) which is an inter-segment call to the TMROFF subroutine on RUM 17 [R918]. The CALL (E8 47 06) [R310] is an intra-segment (in the same Code Segment) direct call to the Offset address 0647 which is the offset of the TMROFF subroutine. CALL TMROFF also pushes the IP (Instruction Pointer) onto the STACK and decrements the STACK Pointer by 2. Note that the timers are set up but they are not running at this stage of the code execution; we are still in the initialization code.

The 256 bytes of RAM memory are cleared by a roughly similar set of code (RUM 6) [R05] which uses string manipulation instructions. The lower byte of the A register is cleared by the XOR AL,AL instruction. The CX register is initialized to 256 (the number of times that the loop must cycle). MOV DI, OFFSET MEMST returns the Address of the start of RAM memory (established by the ORG 2000H pseudo-instruction). The OFFSET pseudo-instruction causes the address of MEMST to be loaded into the string-use DI (Destination Index) register. The REP STOS BYTE PTR MEMST causes the STOS (store string) instruction to be repeated until the contents of CX = 0. The REP instruction decrements CX and increments DI after each transfer. The STOS instruction causes the contents of AL to be stored in the RAM address pointed to by the DI register. The execution of this instruction sequence is very fast.

The Interrupt system is initialized next because we need to be able to use the interrupts in the next phase of initialization. MOV DX, MASK [R313] loads the value equivalent to MASK (0FF2AH) into DX. This is the address of the Interrupt Mask register in the IPI. The IPI Interrupt Equates are on FUM 1; these values are all IPI register addresses. MOV AX, MASKV loads the value of 00E8H (from the program equates table, FUM 5) into AX. The OUT DX, AX causes this 00E8H value to be to the I/O address residing in DX that is the IPI Interrupt Mask register. In an identical manner, the IPI INT0CR (Interrupt 0 Control) register is loaded into DX. MOV AX, 0, clears AX and the OUT DX,AX instruction stores this value in INT0CR. This sets INT0 to the highest priority, makes the T0 interrupt edge-triggered, the INT0 non-maskable and direct and not in the special fully nested mode (used with an external 8259A interrupt controller). STI sets the IF flag to 1 in the 80188 Flag Register. INT0 is now able to respond to an interrupt from S104. In addition to enabling the Interrupt system to respond to S104 interrupts, it enables it to respond to Timer Interrupts which are an immediate requirement of the BDS3RS code described below.

The BDS3 servo amplifier is initialized next by the CALL BDS3RS instruction that calls the BDS3RS subroutine (described below) which resets the digital logic in the servo amplifier. [R323] The CALL instruction pushes the IP onto the STACK and decrements SP by 2.

Having reset the Focus BDS3 servo amplifier, we need to wait awhile to enable the amplifier Resolver-to-Digital converter (which reports the motor shaft position to the amplifier) to stabilize. This pause is accomplished by a CALL DELAY instruction (FUM 7) to the DELAY subroutine (described below) which provides a settable time delay. CX is initialized to a value of 3 for the DELAY subroutine; this value provides a 3-second delay.

INITIAL (continued)

Upon return from the 3-second delay, it is now necessary to turn on the 3-phase, 208 VAC power to the servo amplifier 300 VDC power supply; this 300 volts is required to power the amplifier. Phase: [R332] is the code which turns on this power by setting a discrete drive to the solid state relay that powers the 3-phase contactor. The state of the servo amplifier is determined by testing the bus fault discrete status from the amplifier; if correct, the code continues initializing for entry into the operational FRM control state. If BDS3 faults are sensed, the BDS3 Fault flag is set which signals this condition to the Antenna Control Computer via the monitor data readout. This flag is tested in the course of execution of a position command (this operation is described later).

An important point needs to be emphasized here. If hardware faults are discovered in the course of executing initialization or any other section of the program, the program will not attempt to drive the FRM. The fault conditions will be reported to the Antenna Control Computer via the monitor data. The program will continue to operate (in the BOSS loop) but subsequent attempts to move the FRM will fail if the fault persists.

MOV AX,8000 followed by OUT P1PTA, AX sets Prom 1, Port B7 to the 1 state and all other Port A and Port B bits to 0, the inactive state (at this point we don't want anything else turned on). This command is followed by a 3-second delay to permit the 300 VDC power supply to stabilize and the BDS3 amplifier fault and status logic to become stable. CX is set to a count of three for the LOOP instruction below. PH301: MOV TFLAG1,0 clears the Timer 1 flag; it will be set by the T1 interrupt after T1 times out a 1-second period. MOV DX, BDERL1 sets the BDS3 discretes read address (407, FUM 3) into DX to read BDS3 discretes status.

The Focus system has one BDS3 servo amplifier; the two codes differ slightly in this test. The instruction, PH302: IN DX,AL reads the 8 (low-true) bits of BDS3 fault-status data into AL since there are only 8 bits to read. Now, what are we checking? AND AX,01 performs a logical and of the LSB BADS3 status-data byte and the LSB in AL; the results of the AND are left in AL. (Note that it is actually the FBAD7 bit in S104; the bit order is inverted in going from S104 to S101). This bit selection does not appear to be the case but remember we are and-ing the AX register; the argument of the AND AX,01 instruction could have been written 0001H which is more obvious. Bit 0 is the BADS3 Bus Fault bit which is a 1 before the 300 VDC power is turned on and signals a 0 if there is no problem with the power supply after the 300 V DC is turned on. A bus fault causes the bit to be a 1 after the 300 V DC power is turned on. No other BDS3 fault bits are tested at this juncture; the other fault-status bits will be tested later in the code. The AND AX, 01H instruction sets the ZF (in the FLAG register) if the results of the AND are 0. If there is not a bus fault, the fault bit will be 1 and the ZF will be a 0. In this case the JNZ THREX transfers control to THREX which clears the BDS3 fault in FAUL2.

If there is a bus fault, the fault bit is a 0 and so the ZF will be set so that control falls through the JNZ to test the TFLAG with the TEST TFLAG, 1 instruction. TFLAG had been set to 0 at PH301 [R336] above. Until the TFLAG is set, the TEST instruction will set the ZF and the following JZ PH302 will continue to loop back to PH301 which will again test the servo amplifier status; this looping will continue until the Timer sets the TFLAG to 1. When this happens control falls through the JZ instruction to the LOOP instruction which will decrement CX (it was set to 3 at [R335]), if CX is not 0, the loop instruction will transfer control back to PH301 which will set the TFLAG to 0, read the amplifier status and continue looping on the bus fault status test until CX = 0. After 1-second 3 LOOP cycles, the bus fault bit is set in FAUL2 and the JMP PHASE transfers control back to the start of the three-phase test. The amplifier is unusable so this is as good a place as any for a malfunction loop. BOSS never has a chance to assume control.

INITIAL (continued)

If there is not a bus fault, JNZ THREX (note the displacement of 11H, at this instruction IP is at offset 0085 and THREX is at 0096) causes a jump to THREX [R348] which turns off the timers by CALL TMROFF. The 3-second delay provides time for the 300 V DC power supply and BADS3 bus fault logic to stabilize. (The TMROFF subroutine is discussed below). MOV FAUL2,0 clears the BDS3 fault flags in the Monitor Data Storage table, FUM 4.

The Rotation system has two amplifiers; therefore in reading the servo amplifier status the AX register is used because 16 bits must be read. The PH302 IN AL,DX instruction loads AX with the low-true amplifier's status. The status data in AX is inverted by the NOT AX instruction which makes them high-true. The AND AX, 0101H instruction selects only the bus fault status bits in AX; all other are cleared to zeroes. If there is a bus fault, the ZF will be cleared if the inverted bus fault bit is a 1 so that control will continue to loop for three 1-second cycles while reading and testing the bus fault bit as described immediately above. At the end of this period the bus fault bit is written into FAUL2 as above and control will loop indefinitely in the PHASE loop.

If there is not a bus fault within the 3-second period, the JZ THREX transfers control to THREX which turns off the timers by calling TMROFF. The fault flags in FAUL2 are cleared by the MOV WORD PTR FAUL2,0 instruction.

MOV WORD PTR SERVER,0603H, machine code C7 06 26 20 03 06, is a MOV word immediate operand to memory. C7 is the MOV code, 06 means displacement is absent and not relevant, SERVER is at RAM address 2026 and the data is 0603. Note the byte order of the data and address are interchanged. SERVER is a monitor data storage word which contains the serial number (06) of the S101 and 03 is the firmware revision number.

Having accomplished most of the initialization, it is now appropriate to CALL DSTOR [R746] to acquire the status of the FRM hardware at the Apex. We do this to obtain current FRM position and fault status data before we enter BOSS, the executive control sequence. DSTOR is the FRM and system data and status gathering routine which is called before control actions are taken.

TMROFF (Timer off) Subroutine (RUM 17, FUM 17)

TMROFF is a subroutine which turns off the timers and reinitializes the timer control registers in the IPI for another cycle of timer operations. This subroutine transfers the contents of a table of values into the IPI timer control registers in a manner similar to the DMA 0 initialization sequence described above. The now-familiar MOV SI, OFFSET TSETUP sequence causes the address of the TSETUP table (FLM 16) [F896] to be loaded into the SI (string-related Source Index) register. MOV DX, TOCOUT causes the address equivalent to the Timer 0 Count Register (see Timer Control Registers equates on FUM 2) to be loaded into DX. CX is set to a count of 12. TSET: OUTSW causes the transfer of a word from the source pointed to by the contents of the SI register to the output port pointed to by the contents of the DX register. After the transfer, SI is incremented for the next transfer. The DX register is next advanced by a count of two by the ADD DX, 2 instruction to point to the next timer control register in the IPI. LOOP TSET causes CX to be decremented and transfers control to the OUTSW instruction if CX is not equal to zero. Note that CX was set up for a count of 12 but there are only 11 IPI registers to initialize. Referring to the 80188 Hardware reference manual, note that IPI Timer 2 addresses increase by two but the 64H address is missing because Timer 2 does not have a Max Count B register. This Timer initialization code pretends that there is one because it simplifies the code. The effect is innocuous because the 11th load has no effect. When CX = 0, the RET instruction following LOOP is executed. RET is an intra-segment return which pops the return address off the STACK onto the Instruction Pointer. The STACK Pointer is also incremented by 2.

TMROFF (continued)

Now what has the transfer of these Timer control variables from the table to the IPI Timer Control registers done? The three mode registers are set to a value of 16384 or 4000 hex. This is the INHbar bit which permits the EN bit to be turned on by a write to the EN (enable) bit. T1 has been set to a Max A count of 2000, T1 is clocked at 1.25 Mhz so that it times out at 1600 us or 1.6 ms delay. T2 has been set to a Max A count of 690 which produces a 552 microsecond delay. As a result of the foregoing setup, the timers are set up and ready to go but not counting; they must be turned on by a write to the timer's mode control register with the EN bit set.

BDS3RS (BDS3 reset) Subroutine (RUM 17, FUM 16)

BDS3RS subroutine initializes the BDS3 servo amplifier digital logic by commanding a 1-second reset. IN AX, P1PTA reads the state of the PROM1 Port A and Port B bits into AX. OR AX, 2000H merges the Port A and B states with the 2000H bit which sets the Port B5 bit to a 1 state (the other bits are unaffected by the or operation). This term, when output by the following OUT P1PTA, AX instruction, is a discrete command to send a low-true BDS3 RESETbar command signal to the BDS3 servo amplifier. The duration of the reset command is determined by the duration of the DELAY subroutine described below. MOV CX, 1 sets up a loop count value of 1 for the CALL DELAY instruction which follows. The DELAY subroutine causes a 1 second duration reset of the Focus BDS3 servo amplifier. CALL DELAY pushes the contents of IP onto the STACK and calls DELAY at Offset 0063 [R902]. RET returns control to the next instruction following the CALL BDS3RS instruction.

DELAY Subroutine (FUM 17, RUM 17)

DELAY is a general-purpose delay subroutine which can be called from any place in the code to execute a delay of one or more (integral) seconds; it is called with CX initialized to the required number of seconds (i.e. 1-second loops). The execution of the 1-second time-out is done by a CALL to the subroutine SEC1. CALL SEC1 causes the IP contents to be pushed onto the STACK.

The time out of Timer 1 is signalled by the state of the TFLAG1 flag that is set by the TMR1 code (on FLM 9); see the TMR1 interrupt subroutine described below. DELA1 MOV TFLAG1,0 sets a zero in the TFLAG1 (a byte value) in the RAM flags table (FUM 5). The T1 Interrupt will set this flag to the 1 state. DELA2: TEST TFLAG1,1 is an immediate operand with memory operand instruction (F6 06 67 20 01) that tests the state of TFLAG1 (in the RAM at offset 2067H) against the data value 01. Decoding the 06 in the mod reg-r/m-field, we see that a mod of 00 indicates that displacement low and high would normally be absent; a 000 for reg specifies AX and 110 for r/m specifies that the EA is disp-high:disp-low. TEST sets the ZF (Zero Flag in the Flag Register) if the TFLAG1 value is zero. The JZ DELA2 (74 F9, Jump On Zero) instruction tests the state of the ZF and, if not a 1, control is returned to location DELA2 above. If ZF = 1, control is transferred to the sign-extended displacement which is IP + 1 or the next instruction: LOOP DELA1. LOOP DELA1 decrements CX and tests it for the number of loops to be executed. If CX = 0 after decrementing, the next instruction following LOOP DELA1 is executed. Note the machine code for LOOP DELA1: E2 F2. E2 is the instruction code and F2 is the displacement, in this case -12H (F2 in two's complement). This -12H (minus 12H) means a physical address difference of 12H; the loop instruction is at offset 0682H and the loop return is at offset 0676H; the difference is -12H (or -14 decimal). The value is negative because we are always looping back to a lower address.

SEC1 Subroutine -(FUM 17, RUM 17)

In SEC1, (FUM 17, [F925]) the 80188 T1 and T2 timers are used to determine the 1 second time delay. To use T1 and T2, the IPI control registers must be initialized. MOV DX, T1MODE loads the

SEC1 Subroutine (continued)

address equivalent to the T1MODE into DX. MOV AX, 0E009H loads this value into AX which is output by the next instruction to the IPI T1MODE register in the IPI. What does this 0E009H value do to the the T1 mode register? The EN bit is set which enables T1 to count. ENbar is also set which permits the EN bit to be set, enables the T1 interrupt, and sets T1 to the CONT (continuous operation) mode. MOV DX, T2MODE loads the address equivalent to the IPI T2MODE control register into DX. MOV AX,0C001H loads AX with a control value which is output to this DX address by the next instruction. What does this value do to the T2 mode register? The EN and INHbar bits are set and Timer 2 is set to the CONT (continuous operation) mode. The final RET instruction in SEC1 pops the IP values off the stack which returns control to DELA1 in DELAY.

Note that the Timer 2 interrupt does not occur in this subroutine; Timer 2 functions to produce a time delay but the Timer 2 interrupt is not enabled.

DSTOR Subroutine (Upper Memory Version) (FUM 14, RUM 14)

DSTOR is a very important subroutine that is called from many places in the control program. There are two versions of DSTOR; this upper memory version is used when the F-R Control System is not executing motion commands. The lower memory version (described below) is called by the TMR2 subroutine during the execution of motion commands.

The function of the upper memory DSTOR ([F762] FUM 14) is to cause S102 to sample the FRM data and states and serially output this data to the S101 Apex Interface registers. DSTOR reads, formats and stores the data for access by the control and monitor data readout programs. Examination of the S102 register formats shows the need for shifting, merging, etc. to get the data assembled to a storable format. The response of S102 to a S101 data request is a serial stream of data which is asynchronously loaded into the 7-byte S101 Apex Interface data register. DSTOR reads the 7-byte register one byte at a time by direct input instructions. The subroutine emits the data request, tests for the arrival of the data, formats and stores the position and velocity data, tests and stores the Apex fault discretes, formats the Apex analog data for monitor data readout, and Bounds tests (i.e. tests for faults) the Apex analog data.

At this point it is necessary to have at hand Sheet 3 of the S101 and Sheet 5 of the S102 drawings. The S102 drawing shows how the data is formatted for output and the S101 drawing shows the register structure for the data input to the 80188. Frequent reference to these drawings will clarify the following description. Starting from the output register (F22 on the S102 drawing), the top two bits are a two-bit (a 1,0 pattern) data synchronization bit pattern. Following this bit pattern, the Focus Position data is a 16-bit value distributed over three 8-bit shift registers. Following the position data is a 12-bit velocity value distributed over two 8-bit shift registers. Following the velocity data is a 12-bit Apex analog data value distributed over three 8-bit shift registers. Following the analog data is the 4-bit multiplexer address associated with the analog data. This address is in the same register chip as the lower portion of the analog data value. Three spare bits follow the mux address. The last byte contains six discretes: the lower four are the 1st and 2nd Focus upper and lower limit switch bits; the Focus Brake discrete; and the state of the 2nd screw sensor. The data serially loaded into the seven S101 Apex Interface registers are an exact replica of the these S102 registers.

DSTOR starts by clearing AL using the XOR AL,AL instruction. The address of the Apex request, (APEXREQ, 384, FUM 3) is loaded into DX. OUT DX,AL causes this 384 enable to clear the Apex Data Register and sends a request-data strobe to S102. A series of input instructions will read the 7 bytes of this register. The address (386) of the most significant byte of the register is formed by ADD DX,2 which adds 2 to the contents of DX (i.e. 384). The presence of Apex data in the register is indicated by detection of the 1,0 bit pattern in the most significant bits of the data byte; these will be tested by a 12-

cycle loop below. S102 requires up to 660 microseconds for data readout. The readout timing is synchronized with the S102 A/D converter which has cycle rate of 100 microseconds and is asynchronous with the 80188 code. MOV CX,12 (Decimal) sets up the CX register for a 12 cycle loop to test for the presence of Apex data.

DSTOR1 IN AL,DX [F768] reads the Apex register MSB into AL. AND AL,0C0H (24 C0, AND immediate to accumulator) selects the top two bits in AL and leaves the results in AL. CMP AL, 80H (3C 80) is a compare immediate with accumulator. This instruction subtracts 80H from the contents of AL; if the results are zero (i.e. the arguments are identical), the ZF (Zero Flag) is set (to 1). JZ DSTOR2 tests the ZF; if true, control transfers to DSTOR2. Note the JZ op code: 74 0A; the DSTOR displacement is 0A. When the JZ instruction is executed, IP is at the next instruction (IP = 04E9). DSTOR2 is at FF80:04F3 [F776] and the difference is 04F3 - 04E9 = 0A, which is the displacement shown in the instruction. If the ZF is 0, control is transferred to the next instruction, LOOP DSTOR1. In this LOOP instruction, CX is decremented and tested for a value of 0. If CX = 0, control falls to the next instruction, OR FAUL1 which is interpreted to mean that the S102 is unresponsive or dead. If CX is not zero, control is returned to DSTOR1 IN AL,DX which again tests the top two bits of the data register for the presence of the 1,0 bit pattern. In the event that the 12 attempts to detect Apex data have failed, OR FAUL1,1 will or the 1-bit and the contents of the FAUL1 flag to indicate that the S102 is unresponsive and control is transferred to DSTOR6 by the unconditional jump JMP DSTOR6 instruction.

We have not yet discussed the unconditional JMP instruction; JMP DSTOR6 is a typical example. The op code is E9 C9 00 which is an intra-segment direct jump with a displacement of 00C9. The address of DSTOR6 is FF80:05BC. The IP at the JMP DSTOR6 instruction is 04F3, thus 05BC - 04F3 = C9, the displacement shown in the op code. The jump to DSTOR6 is a fault exit; we will describe the DSTOR6 code later.

At DSTOR2: [F776] we now begin to format and store the Apex position data for subsequent use by the control-data readout parts of the firmware. In DSTOR1, just above, if the 12-loop test for the presence of the 10 bit pattern failed, an Apex fault bit was set in FAUL1 and control was transferred to the no-response DSTOR exit. In DSTOR2: we clear the Apex flag bit (set by a no-response result in a previous cycle through DSTOR) by the AND FAUL1, 0FEH instruction; the Apex 10 bit pattern was detected by the 12-th loop. The op code is 80 26 08 20 FE. This is an AND immediate operand to memory operand instruction in which the immediate data is 0FEH and the memory operand is FAUL1 in RAM. Now what does this op code mean? First, 80 means immediate operand to memory with w = 0, denoting a byte operation. Decoding the mod-reg-r/m field we see: 00 in the mod field normally means that displacement is absent but 110 in the r/m field denotes that the EA = disp-high:disp-low. The 100 in the reg field denotes AH (w = 0). The displacement of 2008 is the offset address of FAUL1 in the RAM. FE is the immediate data to be and-ed with the contents of FAUL1, a word value in the monitor data storage table which contains fault bits which indicate the types of FRM and operator faults. The 0FEH argument of the AND instruction applied to FAUL1 will cause all bits of the lower byte of FAUL1 which are a 1 to be kept intact, except for the least significant bit (in the E) which will be cleared to 0.

The next instruction, DEC DX decrements DX. The op code is 4A which specifies a 16-bit register operand. The three register bits are 010 which designates the DX register. What is in DX? At the start of DSTOR, the code read the contents of the Apex register top byte with (DX), (i.e. the contents of DX) set to address 386. DX has not been altered. Decrementing DX points to 385, the second byte of the Apex register which contains data bits FD2 ... FD9, a component of the Focus position. IN AX,DX reads this data word (both the top and 2nd bytes). MOV BX,AX puts it in BX. Thus the contents of BX are:

(BH) = 1,0,FD15, ... FD10; (BL) = FD9, ... FD2 in which the most significant byte is on the left, just as register bits are represented in the 80188 registers. SHL BX,2 shifts BX left two bits; thus the contents of BX are: (BH) = FD15, ... FD8; (BL) = FD7, ... FD2,nb,nb where nb means null bit. This left shift pushed the leading 1,0 bits off the end of the BX register. ADD DX,2 increases the 387 reading address by two to 389. IN AX,DX reads the next word (i.e. the next two lower position data bytes) into AX. The AX contents are: (AH) = FD1,FD0,FV11 ... FV6: (AL) = FV5 ... SAD11,SAD10. ROL AH,2 makes (AH) = FV11, ... FV6,FD1,FD0. AND AH,03 makes (AH) = 0,...0,FD1,FD0 by masking out all the non-position data bits. OR BL,AH makes (BL) = FD7,... FD2,FD1,FD0. As a result of this shifting and masking, the contents of BX are: (BH) = FD15, ... FDFD8: (BL) = FD7, ... FD0; we have now gotten rid of the 1,0 bit pattern and reassembled the 16-bit Focus position data into a standard word. MOV POSD,BX stores this current Focus position in the Monitor Storage Table in RAM (shown on FUM 3).

We now determine the position command error [F7B9] by reading the most recent position command value from the Antenna Control Computer (POSCEC from the Monitor Storage Table in RAM, FUM 3) and take the difference between this commanded value and the current position. We do not have to be executing a command to form this error although it is a direct measure of position command execution; the value will diminish as the mechanism is driven to the commanded set point. If for some reason the Focus drive were to shift position after a command execution, for example due to brake slippage, the position error will emphasize this condition; it is a sensitive indicator of the physical stability of the drive mechanism. BX holds the current position. MOV AX,POSCEC reads the command value into AX. SUB AX,BX takes the difference. This is a register operand to register operand difference with the difference in AX. The op code is 2B C3. From the programmer's reference manual we see that it is a register-register operation. The meaning of C3 is more evident if we write it as 11 000 011; the mod field is 11 which means that the r/m field is treated as a reg field; in this case, 000 designates AX. The r/m field is 011 which designates the BX register. The SUB AX,DX instruction affects the carry flag; we are soon going to make a decision on this carry. In executing this SUB AX,DX instruction, if AX is greater than DX there is no borrow so the CF is not set (i.e. a 0). If the inverse case, the CF is set to 1.

MOV DIR,0 loads a 0 into the DIR flag in RAM, FUM 5. The DIR flag indicates the direction to move the mechanism to null the error. If (DIR) = 1, the Focus drive should go down (i.e. decrease the position value) to null the error. In the case of Rotation, DIR = 1 indicates that we should drive CCW. Loading DIR with a 0 assumes that the error is positive, i.e. we assume that we must drive up to null the error. The JNC DSTO2A (jump on not carry) causes the next instruction to be executed if the CF = 0. If it is set (CF = 1) control is transferred to DSTO2A. In this case the op code is 73 05, which is a jump to a positive displacement of 05. In executing the JNC DSTO2A instruction, the IP is at 51B, DSTO2A is at address 520, and the difference is 5 which is the displacement shown in the op code. If the CF is 1, control falls through to MOV DIR,1 which puts a 1 in the DIR flag.

DSTOR2A MOV ERROR,AX [F793] puts the position error that was left standing in AX into the error word in the Monitor Storage Table in RAM (FUM 3).

We next address formatting and storing the velocity data [F795], which is a 12-bit value distributed over two 8-bit registers. The techniques of formatting and storing this data are similar to the position formatting above but is a bit simpler since only two Apex registers contain the data. From the code above, DX was incremented to produce the Apex register address: enable 387. IN AX,DX loads the two registers into AX. AX contents are thus: (AH) = FD1,FD2,FV11 ... FV6; (AL) = FV5 ... FV0,SAD11,SAD10. AL is saved in BH for Apex analog data formatting below. SHR AX,2 makes the contents of AX: (AH) = n,n,FD1,FD0,FV11, ... FV8; (AL) = FV7, ... FV0 which puts the 12-bit velocity

value in the lowest 12 bits of AX. AND AX,0FFFH masks off the non-velocity data in AX; the result is that the upper 4 bits are zeros. MOV VEL,AX stores the velocity data in the VEL word of monitor data storage (FUM 4) for subsequent reference and monitor data readout.

The analog data and associated mux address are the next parameters to be formatted. The contents DX were left with enable 387. ADD DX,2 [F800] increases the address to 389 which enables the mux address and lower portion of the analog data to be read. IN AX,DX reads these registers so the contents of AX are: (AH) = SAD9, ... SAD2; (AL) = SAD3, ... SAD0,SM8, ... SM1, where SADx bits are sampled analog data and SMx are mux address bits. SHR AX,2 makes (AX): (AH) = n,n,SAD9, ... SAD4; (AL) = SAD3, ... SAD0,SM8, ... SM1. Remember that BH was left with the top two analog data

bits; ROR BH,2 makes (BH) = SAD11,SAD10,FV5, ... FV0. AND BX,0C000H selects the top two bits and clears the lower six bits which makes the contents of BX: (BH) = SAD11,SAD10,0, ... 0; (BL) = 0 ... 0. OR BX,AX merges AX into BX so that (BX) is: (BH) = SAD11, ... SAD4; (BL) = SAD3, ... SAD0,SM8, ... SM1. BX contents are left in this form for subsequent use in DSTOR3.

We access the discretes data next: ADD DX,2 [F806] increases the data read enable address in DX to 391. IN AL,DX reads the discretes byte (we were reading words above) into AL. The contents of the discretes data register is high-true. AL contents are: SP,SP,2ndSCR,BRK,UL2,UL1,LL2,LL1 - all reading zeros for the true state of the discretes (except for the SP bits which are floating inputs on the S102 discretes register). AND AL,3FH masks off the two SP (spare) bits so that they are replaced by zeros. (The Rotation code use an AND AL,1FH instruction since the 2ndSCR bit is not present.) AND SYSTEM,0FBH zero's the upper byte of SYSTEM, (system status states in RAM, FUM 4) and retains all bits to set to the 1 state except for bit 2 which is cleared by the 0 in FBH. (See the format of the SYSTEM word in Section 3.9.) This instruction sets the state of SYSTEM with the assumption that the brake is released; we will test the brake discrete bit to verify or refute this assumption. (Remember that the discretes are high-true.) TEST AL,16 tests the state of the BRK (Brake bit) in AL with the 16 (decimal) bit which corresponds to the BRK bit in AL. The op code of TEST AL,16 is A8 10 which is an immediate operand with accumultor, w = 0 (i.e. a byte operation), and the data is 16 decimal. The purpose of a TEST instruction is to set a flag so as to be able to make a logical decision. In this case the ZF (zero flag) is set if the logical product of the 16 bit and the BRK bit in AL is true which means that the brake bit is a one; the brake is energized. JZ DSTOR3 bypasses the next instruction, OR SYSTEM,4 which sets the brake release state (i.e. bit 2 is a 1) in SYSTEM.

DSTOR3: MOV PHASEA,AL [F813] loads the state of the six discretes standing in AL into the temporary storage PHASEA in RAM (FUM 4). AND PHASEA,32 (decimal 32 bit) selects the 2nd screw sensor bit (bit 6 in PHASEA); all the other bits in PHASEA wil be zeroed. The 2nd screw bit has a function differing from the limit and brake faults; it will alternate between states as the Focus mechanism is driven; PHASEA will be tested in the context of the second screw analysis code discussed later. (This reading and storage of PHASEA is of course not present in the Rotation code. AND AL,0FH selects the lower four (limit switch) bits in AL. If a limit switch is not actuated (i.e. switch closed), the corresponding bit in AL is a 0. SHL AL,1 shifts AL left one bit; a 0 is loaded on the right. AND FAUL1,0E1H clears the upper byte of FAUL1 and selects the 7,6,5 and 0 bits in the lower byte of FAUL1; only these bits will be a 1 if they were a 1 before the logical AND. OR FAUL1,AL forms the logical-or of the lower byte of FAUL1 and AL; the result is that all bits of the lower byte of FAUL1 are 1's if any limit switches is actuated. The contents of the FAUL1 word will be read out as monitor data.

BX contains the 12 bits of analog data and four mux bits as a result of the operations described two paragraphs above. The upper 12 bits in BX are the analog value and the lower 4 bits are the mux bits. MOV DX,BX [F819] puts these data into DX. OR BX,0FH makes the four lower bits of BL all

ones so that the 12 bits of data are left-adjusted in the 16 bit register field with 1's filling the lower four bits; this is the format that the Standard Interface outputs when an analog signal is converted to a digital value. We will test and store this data after we generate a random number in the RANDOM algorithm.

AND DX,0000FH selects the four lower Apex analog multiplex address bits from the composite data standing in DX. The three leading 0's mask out the analog data bits and if any of the M8 .. M1 bits is a 1, they are retained intact in DX. The program now develops a random number to be put in the RANDOM location of RAM. The RANDOM value is a derivative of the Apex analog mux address and is used in the BOSS loop (discussed below) to generate randomly ordered time delays. The RANDOM algorithm is: RANDOM = (((RANDOM + Mux Value)modulo 64) + 1). This algorithm produces a number between 1 and 64 inclusive (it should never be 0), based upon the 4-bit Apex analog mux values read by DSTOR. The code which generates RANDOM is as follows: ADD DX,RANDOM [F822] adds the existing random value to DX which contains the mux address just isolated above. AND WORD PTR RANDOM, 3FH is an immediate operand to memory instruction using the pseudo instruction WORD PTR. The op code is 81 26 5A 3F 00. 81 is the logical AND with w = 1 (i.e. a word operation). 26 means the next two bytes are the displacement of RANDOM (from DS = 0000) and 003FH is the immediate data value. This instruction performs the logical AND of the 7-bit immediate value with the RANDOM data and leaves this value in RANDOM. The AND result truncates the upper portion of RANDOM to a 7-bit or modulo 64 value. The contents of RANDOM are next incremented by INC RANDOM so that it is never zero.

We now consider the processing and storage of the Apex analog data. XCHG DX,BX [F825] is a register-to-register interchange. Consider the op code 87 D3; 87 is the XCHG code with w = 1. D3 is the mod reg r/m field which is: 11 (mod), 010 (reg) and 011 (r/m). A 11 mod means the r/m field is to be treated as a register; in this case the 011 designates the BX register. 010 designates the DX register. BX still contains (from the operations above) the left-adjusted analog value with 1's in the lower four bits. DX contains the four analog mux bits that identify the data. The next instruction is a CMP BX,03, an immediate operand with register instruction. The op code is 83 FB 03 in which the mod 111 r/m field is 11 (mod), 111 (nul) and 011 (r/m). 11 in mod means that r/m designates a register; in this case r/m designates BX. Finally, the data operand is 03. CMP BX,03H subtracts the source (03) from the destination (BX) without changing the operands; the flags register reflects the result of the comparison. Why subtract a count of 3 from the mux address? A quick glance at the S102 analog multiplexer drawing (D55007S006, SH 6) shows that the first three multiplexer inputs are Rotation analogs: three GND's and Rotation velocity; we have no interest in Rotation parameters since this is a Focus code.) This is the reason for the comparison; if the mux address is for a Rotation parameter, we want to skip the rest of analog processing and go to DSTOR6. JLE DSTOR6, a jump less than or equal instruction, tests the sign flag; if (BX) was greater or equal to 3, the SF = 1. If SF = 1, control is transferred to IP + displacement. The JLE op code is 7E 35, in which 7E designates JLE and 35 is the displacement. At this instruction IP = 0587, the address of DSTOR6 is 05BC. The difference is 35H; this is the displacement value. If the SF = 0, control passes to the SUB BX,4 instruction in which 4 is subtracted from BX, which contains the 4-bit mux address. We are developing a storage index to store this Focus Apex analog data in the monitor data table. Subtracting 4 from the mux address forms an index for the data table: ANADAT, which is in RAM (FUM 4). MOV CX,BX puts the index in CX. ADD CX,1 increases the CX value by one. CX is set to this value for use in a LOOP instruction which is described below. SHL BX,1 multiplies the mux address by 2. MOV ANADT[BX],DX stores the analog data in DX in the ANADT table, indexed by the contents of BX. The op code is 89 97 0E 20. In this case w = 1 so it is a 16-bit move; 97 is a mod of 10 (designating a disp-high:disp-low); reg is 010 which designates DX and r/m is 111 which designates that the operand address is (BX) + disp. Displacement is 200EH, the offset address of ANADT from the DS value of 0000.

In the Rotation version of DSTOR, in testing the mux address of the Apex analog data at [R809] if the mux address is 3 or less, the data is a Rotation parameter and should be range tested. CMP BX, 3 subtracts 3 from the mux address in BX. If the SF is not 0 or the ZF = 1, the JLE DSTO3A jump will transfer control to DSTO3A which develops the pointer as described above for Focus. BX was not disturbed by the comparison. SUB BX, 4 subtracts 4 from this address and CMP BX, 3 compares the modified address to 3 by subtracting 3 from the BX value. If the SF is not 0 or the ZF = 1, the address is for a Focus parameter which is of no interest in the Rotation code so the JLE DSTOR6 transfers control to DSTOR6.

We have stored the analog data; the next operation is to determine whether the Apex analog data values are within acceptable bounds. Since this data is already in the Unsigned Word format, it is not necessary to perform a format transformation. The (DX) value is compared with lower and upper limit values in consecutive memory locations by the BOUND DX,CS:ANATAB[BX] instruction [F835]. This is a Based Indexed addressing scheme to determine operand address. (See the IPRM, page 3-18.) The operand address is determined by adding the contents of BX (the index), to the contents of a base register which is then added to the displacement. ANATAB is at offset address 06A4 (from DS = 0000). ANATAB [F947] (FUM 17) is the table of limit values. SHL BX,1 multiplies the index in BX (the mux address) by two because it is used as an index to access the two limit value words in the table. MOV BFLAG,0 initializes the Bound flag to 0 in Flag Storage (FUM 5). If the parameter under test is outside (i.e. less than the lower or greater than the upper) the limits in the BOUND instruction, a BOUND interrupt is generated which causes a transfer of program control to the BOUND interrupt code (TRAP 5) in lower memory. The BOUND interrupt code will set the BFLAG to the 1 state which is tested after the interrupt return. (The BOUND interrupt service code is be described below.) Reference to the table of BOUND limit values in ANATAB [F947] shows two .WORD values for all parameters: 8000H (first value of a pair) and 7FFFH (second value). (Note the usage of the Signed Word format.) The lower limit value: 8000H is the lowest possible negative value and 7FFFH: the upper limit value is the maximum positive value for the BOUND comparison. With these values, no BOUND test can fail. Specific values for this limit table will be added in a future revision.

The BOUND op code deserves some commentary because it is a powerful and frequently used instruction that involves control transfer via the BOUND interrupt. The BOUND op code is: 2E 62 97 A4 06. The 2E is a segment over-ride prefix (IPRM pages 3-11 and A-1) which specifies that the CS register be used for the base register rather than the SI or DI registers. This prefix is caused by the CS:ANATAB syntax in the instruction mnemonics. 62 is the BOUND code (IPRM page 3-54) and 97 is the mod-reg-r/m code and A4 06 is the displacement. The 97 specifies: mod = disp-high:disp-low; reg = 010 which is the DX register and r/m = 111 specifies (BX) + Disp. Thus the contents of DX are compared with the address determined by the sum of (CS) + (BX) + Disp, the address of the first word of the pair of limit table elements at location ANATAB. The offset of ANATAB [F947] is 06A4 which is the displacement in the instruction.

The IPRM, page 3-53 has a typographical error in the description of the BOUND array comparison: (1EH) is shown as loaded into (CS) and (1CH) is shown as loaded into (IP). These two values are erroneous: (16) is loaded into (CS) and (14) is loaded into (IP). Reference to the interrupt table code on FLM shows that the four-byte table at 0000:0014 to be 9D00 and 0000. These values in CS and IP vector the BOUND (Type 5) interrupt to address 009D which services the BOUND interrupt; thus in preparation for control transfer to this interrupt service code, the CS and IP registers are initialized with the appropriate values to vector control to the BOUND interrupt service code.

Also note that in the register initialization, the IF (interrupt flag) and TF (Trap Flag) are cleared (to 0). Clearing the IF and TF flags disables maskable and single step interrupts so that the TRAP 5

operations cannot be disturbed by another maskable interrupt.

The BOUND flag (BFLAG) will be set to the 1 state in executing the BOUND interrupt code.

Having returned from the BOUND interrupt (if an out-of-limits Apex analog value was detected) or having passed the BOUND check, the BFLAG must be tested to determine the program action. Remember that in ANAFL there are nine possible analog fault flags; we want to set a flag bit for the detected fault. Also, remember that CX was set up to an indexing value just before the BOUND test; this value is the mux address-based index to point to the associated flag bit. MOV AX,8000H puts a single (MSB) flag bit in AX. ROL AX,1 puts this bit in the LSB position of AX. The DSTOR4 loop left-rotates this flag bit to the position determined by the contents of CX which controls loop count. TEST BFLAG,1 tests the BOUND flag (set in the course of executing the BOUND interrupt); if the flag is a 1, the faulty data will be identified by the position of the flag bit in the ANAFL monitor data word. In the TEST BFLAG,1 instruction, the ZF bit will be set (to a 1) if BFLAG is 0, indicating the data value tested by BOUND is within limits. JNZ DSTOR5 tests this ZF and if ZF = 1, control falls through to NOT AX; otherwise control transfers to DSTOR5 which sets the flag in ANAFL. NOT AX is a new instruction with the op code: F7 D0. F7 designates NOT and w = 1, a word rather than byte operation. D0 is the mod010r/m byte in which mod = 11 so r/m designates a register, 010 designates DX and r/m designates the AX register. The AX contents are logically 1's complemented, apparently using DX in the process. AND ANALF,AX logically and's AX and ANAFL which sets ANAFL to zeros because the two are complements. Having cleared ANAFL, control is transferred to DSTOR6.

DSTOR5 is the bad data flag code which causes AX to be loaded into ANAFL to flag the data for subsequent readout as monitor data. If a subsequent pass through DSTOR flags a different analog mux bad, the previously flagged value is overwritten; hopefully the ANAFL monitor data is frequently sampled. A mitigating factor is that if the Apex triple-output power supply starts to become erratic, all voltages will be affected since it is a tracking supply.

DSTOR now addresses the Apex fault bits read and stored by [F818] above.

In DSTOR6, [F847] the PROM2 Port A status is read into AX; these are the Port A and B latches. Five of these (PA3 ... PB0) drive the S105 front panel LED's. AND AX, 0FE6FH forms the logical-and product of the immediate value and the ports status in AX; if a port bit was a 1, it will not be affected but the three zero bits clear the AX bits corresponding to the upper limit, lower limit and brake LED's. We will retain this modified port state in AX because we may want to alter it and output it to the PROM ports to reflect new LED and system command states. TEST SYSTEM,4 tests SYSTEM to determine if the brake is released (it was set to the release state in [F812] in DSTO2A above). The ZF will be set to 1 by the TEST SYSTEM,4 instruction if the 4 bit is 0, i.e., the brake is engaged (i.e. de-energized). JZ DSTOR7 transfers control to DSTOR7 if the ZF is a 1. If the brake is released (i.e. energized), a 1-bit is or-ed into AX corresponding to the PROM2 PA4 bit.

DSTOR7 [F853] now tests the Down limit switch status by loading the Apex fault status bits in FAUL1 into BX. AND BX,6 selects the 1st and 2nd lower limit bits and leaves them intact in BX if they were a 1; all other bits are zeroed. If either 1st or 2nd (or both) limit bits are a 1, the AND clears the ZF to 0; we have detected a down limit fault. OR AX,128 merges a 1 into AX corresponding to the Down limit LED drive on EPROM2 Port A7. The other AX bits are not affected by the OR instruction.

DSTOR8 [F858] next tests the Upper limit switch in a similar manner using the 16 and 8 bits to select the 1st and 2nd upper limit bits in FAUL1. If an upper limit bit is set, OR AX,256 merges a 1 into AX corresponding to the Up limit LED drive bit on EPROM2 Port B0.

DSTOR Subroutine (continued)

In the Rotation version of DSTOR, these two limit switches are tested by identical test patterns but in opposite order because the Focus and Rotation limit switches are connected in the opposite sense in the S102 logic.

In DSTOR9 [F863] the AX contents which reflect the updated state of the limit and brake bits and the other (undisturbed) control states in AX are restored to the PROM2 ports by the OUT P2PTA,AX instruction. This state now replaces the previous state on the Prom 2 command ports.

The fault-status bits of the BDS3 servo amplifier are read by loading DX with the address of the BDS3 fault port, BDERL1. The fault state is read into AL by IN AL,DX. NOT AL inverts the sense of the faults because the BADS3 bits are low-true. They are next saved in FAUL2; this makes them available to be read out as monitor data. AND WORD PTR FAUL1,0E7FFH uses the WORD PTR pseudo-instruction and is an immediate operand to memory operation. The WORD PTR is used because FAUL2 is a .WORD (2-byte) value although only the lower byte is of interest in the Focus program. The op code is 81 26 08 20 FF E7. The op code designates a word AND (w = 1). The mod-100-r/m code designates that EA = disp-high:disp-low and the displacement of FAUL2 is 2008. The offset of FAUL1 is 2008 (from DS = 0000H) which is the displacement of the instruction. The data to be and-ed with the FAUL2 contents is E7FF which retains all the lower byte fault bits; the E7 pattern for the upper byte has no meaning until additional fault bits are assigned to FAUL2. IN AX,P2PTA reads the states of the two PROM 2 ports; bits PB3 ... PB7 are programmed for input to sense discrete states in the system. In this case we are interested in the states of the Emergency Stop switch and Drive Lockout (discussed in the hardware description of Sections 3.1 and 3.12.) The AND AX,1800H selects only these two bits for test. Drive Lockout is low-true and indicates a hardware fault which should inhibit mechanism drive. XOR AH,10H will invert the selected drive lockout bit; remember that dissimilar states result in a 1 or true output in exclusive or logic. The bit should be inverted because fault bits are read out as high-true or 1's. The XOR op code is 80 F4 10, an immediate operand to register: (AH). In this case w = 0; so we are dealing with a byte rather than a word. The mod-110-r/m byte is: mod = 11, r/m is treated as a register; 110 specifies DH and r/m = 100 specifies AH. 10H is the XOR bit data. Apparently DH is involved in forming the XOR product in AH. Finally, the inverted fault bit is merged into FAUL1 by the OR WORD PTR FAUL1,AX instruction which loads this bit into FAUL1 without perturbing the other FAUL1 bits.

The last operation of DSTOR is the RET instruction [F874]. This is an intra-segment return since we are returning to a CALL location within the upper memory.

DSTOR Subroutine (Lower Memory Version) (FLM 14, RLM 13)

This version of DSTOR resides in lower memory (FLM 14 [F773]) and is called (only) by the TMR2 subroutine in executing a position command. The call from TMR2 is a NEAR CALL; see the IPRM for details. Lower Memory DSTOR is a brief version which quickly acquires and formats position data (only), calculates position error and returns to the calling location. The upper memory version should be studied carefully before reviewing this version because the description is very brief since the code is an abstracted version of the upper memory subroutine. The abstracted portions are identical to the upper memory versions; the differences between the two versions are that fewer decoding and formatting operations are performed in this DSTOR.

The beginning loop which requests data from S102 and tests for the presence of the data in the top Apex Data Register is identical to the upper memory version except for the no-response jump to DSTOR3, the RET instruction to the calling location. In the event that the Apex data is not detected, FAUL1 is flagged just as in the longer version so there is no danger that the calling program will

Subroutine (continued)

misinterpret this data if FAUL1 is tested.

The position formatting and position error calculation is identical to the lower memory version. After storing the position and position error in RAM, the subroutine terminates by a RET instruction to the calling location. This quick DSTOR subroutine does not acquire and format velocity and discretes data.

BRKOFF Subroutine (FLM 15, RLM 14)

The purpose of this subroutine is to re-engage the Focus Drive Brake by turning off the power and verifying that it is off. The brake is a fail-safe brake; it is engaged when unpowered. A time delay is involved to permit the brake current in the inductive brake to decay to an engage value. The brake is powered to disengage by a discrete term (PB6) from PROM 1. IN AX,P1PTA reads the state of both ports into AX. The state of the port outputs serve as an alternate to RAM memory for these states. This usage is better than RAM memory because it's closer to the control output, a better measure of the command state. AND AX,0BFFF selects all "1" bits in AX except for the brake bit which is cleared to zero. The OUT AX,P1PTA instruction stores the modified state of AX on the PROM1 A and B ports. Note that when we read the port states we didn't know or care what the states of other bit bits were; we just wanted to turn off the brake bit without perturbing them. The T1 flag is zero-ed next in preparation for the call to the SEC1 subroutine described above. SEC1 is called and starts a 1-second time delay. In BRKOF1: CALL DSTOR the state of the brake power bit (sensed by S102 and input as part of the DSTOR data) will be acquired and stored in bit 4 of SYSTEM (the table of system states in RAM, [F158]). This operation is described in DSTOR above. The TEST SYSTEM,4 instruction will clear (to 0) the ZF if the 4 (brake) bit is a 1 (engaged). If engaged, the JZ TMROFF will turn off the timers and return control to the location at which BRKOFF was called via the RET in TMROFF. This is the non-fault return from BRKOFF.

In the event that the ZF is a 1, control falls through to TEST TFLAG1,1 which tests the TFLAG1 state. This flag is set by the TMR1 interrupt code in lower memory in response to a TIMER 1 interrupt. (The TMR1 interrupt code is described below.) If the ZF flag is not set by the TEST TFLAG1,1 instruction (i.e. the 1-second period has not timed out), control loops back to the BRKOF1 instruction which re-acquires the brake data. When the 1-second period of SEC1 has timed out and set TFLAG1, OR WORD PTR FAUL1,256 causes a 1 (bit 8) to be or-ed into the FAUL1 fault word for subsequent readout in the monitor data. When the brake is suspected of being faulty, drive motion should be terminated immediately. This is accomplished by the unconditional jump, JMP RSCMD. RSCMD is the soft reset command entry point; the soft reset command shuts down the drive.

INT0 Interrupt (A/D converter EOC) Subroutine (FLM 8, RLM 8)

The INT0 subroutine [F427], FLM 8, loads, formats and stores A/D converter data in response to the A/D converter end-of-conversion signal (STS) which drives the 80188 INT0 input. The A/D logic circuitry is shown on Sheet 4 of the S101 logic drawings. The converted analog signals are 12-bit values, read as a word, converted to a 2's complemenet format and stored for subsequent use by the BOSS control program. Three analog parameters are converted: drive motor current, drive motor velocity and three reference voltages: +10 Volts, - 10 Volts and Analog ground. The storage for the formatted data is a single word location; ADVAL, not in an array; hence the parameter will be immediately accessed after conversion. The A/D conversion command signal is not generated as part of INT0 service code.

INT0 is entered as a result of a vector from the TYPE 12 Interrupt table. The table vectors [F278] are 8101 (locations 0030 and 0031) for the IP and 0000 (locations 0032 and 0033) for the CS.

INT0 Interrupt Subroutine (continued)

These two values vector program control to the INT0 code at offset 0181, (CS) = 0000 [F427] in lower memory (FLM 8). The only registers whose contents will be changed in this subroutine are AX and DX; they are immediately pushed onto the STACK. Mov DX, ADL loads DX with the address (enable 401) of the lower byte of analog data. IN AX,DX reads the two data bytes into AX. The upper 8-bits of the value are in AH and the lower 4 bits of the value are in the upper part of AL. This is not the appropriate format for processor usage; SHR AX,4 right-shifts AX by four bits and loads 4 zeros into the left (most significant half) of AH. Thus the 12 bit value is located in the lower 12 bits of AX. XOR AX, 0800H converts the AX value to 2's complement for usage by the subsequent code. MOV ADVAL,AX stores it in the ADVAL: .BLKB 2 location [F204] in the temporary storage table (FLM 4). MOV ADFLAG,1 sets a 1 into the ADFLAG flag location [231] in the Flag table (FLM 5) to signify (to BOSS) that a new analog value is available in ADVAL. It is necessary to signal the end of interrupt code operations to the Interrupt Controller in the IPI so that the 80188 can respond to a new interrupt. Remember that the 80188 clears (sets to 0) the IF and TF in response to an interrupt; these flags must be set to 1 to re-enable the IPI Interrupt Controller. Writing a value to the EOI register does this. The value written is 12H, corresponding to the INT0 Type12 interrupt. The lower 3 bits designate interrupt priority which is assigned to be level 2. MOV AX,12 and MOV DX,EOI set up this load of End of Interrupt transfer and OUT DX,AX loads this value into the EOI register. EOI is the equate for IPI location FF22H. The DX and AX registers are restored to the pre-interrupt condition and IRET returns control to the location at which INT0 was sensed. Note that in the IPRM, the STACK is popped to return IP, CS and FLAGS to the correct state to resume proper code execution.

BOUND Interrupt Subroutine (FLM 6, RLM 6)

This subroutine responds to the interrupt generated by the BOUND instruction when the array comparison determined that the tested parameter is outside the limits.

Entry to the subroutine is similar to that described for the INT0 service subroutine described above. Executing the BOUND instruction (described in detail in the DSTOR subroutine above) causes program control to be directed to the IP and CS values stored in locations 0014, .. 0017 which contain 009D for the (IP) and 0000 for the (CS). The BOUND: subroutine entry point is at offset of 009D [F328] on FLM 6. As the only registers affected by the subroutine are the AX and DX, they are pushed onto the STACK. MOV BFLAG,1 sets a 1 in this flag location to signal that a BOUND interrupt was induced in executing the BOUND instruction. (Which BOUND instruction induced the interrupt is not known at this point.) MOV AX,5, MOV DX,EOI and OUT DX,AX restore the IPI End of Interrupt register to the pre-interrupt condition so that the interrupt can respond to another non-maskable interrupt. The DX and AX registers are restored to the pre-BOUND interrupt condition and the IRET instruction pops the (IP), (CS) and FLAGS register values off the STACK to transfer control to the instruction following the BOUND instruction which induced the interrupt.

Although not a factor in the execution of this subroutine, the operand and limit values must use the Signed Word format. Unsigned Word format operands and limit values less than 32767 may also be used with the BOUND instruction without error.

NMI Interrupt (Data Request) Subroutine (FLM 6, RLM 6)

This subroutine responds to a data request message from the Standard Interface in S104. In contrast to a command message from the interface which always has a simpler interpretation, a data request requires analysis of the relative address to determine which data is requested, access of the appropriate table location and output within the no-response time constraints of the interface.

NMI Interrupt Subroutine (continued)

The Non-Maskable Interrupt is used to signal a data request because it cannot be masked and has a default priority of 1 (highest priority).

The 80188 response to the NMI is to transfer program control (as described above for the INT0 interrupt) to the (IP) and (CS) contents determined by the interrupt vector locations 0008 through 000B.

The Non-Maskable Interrupt service code entry point is NMI: at [F292] on FLM 6. The AX, DX and BX registers will be altered by this subroutine so they are pushed onto the STACK to preserve their pre-interrupt contents; these contents will be restored on completion of the subroutine operations. The BOUND instruction is used in this subroutine so the BFLAG contents are also pushed onto the STACK. Since the Non-Maskable Interrupt cannot be inhibited, the BOUND interrupt subroutine described above could have been interrupted by the NMI so it is necessary to preserve the BFLAG state for return to this subroutine if it was interrupted by the NMI. After saving the BFLAG, it is cleared because the BOUND instruction is going to be used in the NMI interrupt subroutine. RELADD is the equate for the 395 enable to read the S014 register which contains the mux address of the data requested by the Antenna Control Computer. MOV DX,RELADD loads DX with the address of this location. IN AL,DX loads AL with the byte value. AH is set to 0 for the impending BOUND test; it must be zero-ed to avoid confusing the test because the IN AL,DX instruction will not clear AH. If an IN AX,DX instruction were used, the COMML (lower byte of the most recent position command argument) would be loaded into AH which would confuse the address BOUND test. Why perform this address BOUND test on a monitor data request message? A power glitch could have disturbed the command message stream or the S104 logic. A second reason is that it is good programming practice to test all inputs for validity. This test traps invalid addresses and sets the monitor fault flag bit in FAUL1 to indicate the character of the error.

The BOUND AX,CS:MONRG instruction compares the address in AX with the MONRG table at [F875]. The lower and upper address limit values are 16 and 47 (decimal). The op code: 2E 62 06 05. Note the now-familiar CS segment override prefix 2E induced by the DS:MONRG syntax. This was discussed in the BOUND instruction described in DSTOR above and has the same effect here. Verification of the op code effect is an exercise left for the reader. In the event that the monitor request address is out of the specified range, the BOUND subroutine will have set the BFLAG to 1. TEST BFLAG,1 will set (to a 1) the ZF if the BFLAG was not set; this state signifies a valid address. In this case, control is transferred by the JZ NMI1 location. Since the operand and limits are always small values, the BOUND instruction is safely used.

If the requested address is out of range, bit 6 (weight = 64) of FAUL1 is set to a 1 state by the OR FAUL1,64 instruction; none of the other bits in FAUL1 is perturbed. Since the request is erroneous, the JMP NEXIT transfers control to the exit point and no further subroutine action is taken.

At NMI1 [F306] we begin the process of accessing and outputting the data. To do this we must develop an index to the monitor data storage array at [F155]. This array is in the lower portion of RAM at offset address 2000H. We noted that the monitor data address range is 16 through 47; the mux address in AX must be transformed to an array index. SUB AX,CS:MONREG subtracts AX from the contents of the location pointed to by CS:MONRG. MONRG table contains the value 16 (decimal) noted above, the subtraction result is in AX. Note the op code, 2E 2B 06 ED 05, prefixed by the now-familiar CS segment over-ride: 2E. This is a register operand to memory operand word instruction designated by 2B. The mod-reg-r/m value is 06, denoting a mod of 00, reg = AX and r/m = 110; in this case the EA is disp-high:disp-low. The offset of MINRG is 05ED which matches the 05ED shown in the op code. ADD AX,AX doubles the index in AX because we want to index a 2-byte array defined by the .BLKB 2 pseudo-instruction which reserves a block of two bytes per element. This index is put into BX. MOV AX,MEMST[BX] is an indexed address instruction in which the 8B 87 00 20 op code designates NMI

NMI Interrupt Subroutine (continued)

a word operation in which the source operand designates the EA and the DEST operand is a register. 87 is the mod-reg-r/m code in which mod 10 designates disp-high:disp-low; reg 000 designates AX and r/m 111 designates (BX) + disp. The offset address of MEMST is 2000 which matches the instruction displacement of 2000. The I/O port address (for the INTDX bus to S104) equivalent to the S104 monitor data storage lower byte (MONL) is 399; this address enables the S104 register to store the monitor data lower byte. MOV DX, MONL loads this address enable into DX. OUT AX,DX causes the monitor data word in AX to be output to the S104 register pointed to by MONL and MONM. It is necessary to cause the S104 logic to send a Device Acknowledge signal to the Standard Interface board (in S104) so as to cause the monitor data to be output to the Antenna Control computer. The DX register contents are 399, the address of the lower data byte. DEC DX decrements this value to 398 which is the enable which causes S104 to generate the Device Acknowledge. OUT DX,AL causes the contents of AL to be output to the address designated by the contents of DX. The AL contents are irrelevant because S104 ignores the value.

AND WORD PTR FAUL1,0FFBFH is now a familiar instruction (we commented on it in DSTOR), which enables all set (to 1) flag bits to remain a 1 but zero's the monitor fault flag bit (bit 6 in FAUL1), set in a previous cycle by an out-of range mux address. This flag bit clear signifies that the monitor data operation accessed a valid data address.

The NMI interrupt subroutine must terminate by sending an End of Interrupt code to the IPI EOI register. The NMI vector type is 2; this value is loaded into AX and output to the EOI register in the same manner as the interrupt subroutines described above.

The subroutine terminates by popping BFLAG, BX, DX and AX off the STACK into the BFLAG memory location and the registers. IRET pops the IP, CS and FLAGS values off the STACK into theses registers and program execution resumes at this location with the FLAGS register restored.

DMA0 Interrupt (Command Input) Subroutine (FLM 7, RLM 7)

Commands from the Antenna Control computer are input via the DMA 0 channel. After the command and associated relative address have been stored, the DMA controller issues an interrupt to signal to the CPU to indicate that a new command has arrived. The function performed by this subroutine is to check the address for validity; if valid, the type of command is identified, command request flags are set, the modal change commands are executed under the executive control of the BOSS control program.

Focus Addr	Rotation Addr	Command Type & Argument
10H	30H	Position Command; Focus Arg: 2300 counts/inch; Rot Arg: 156 counts/degree
11H	31H	Nap Mode Command Set/Reset; Arg: 1 = Set Nap Mode; 0 = Reset Nap Mode
12H	32H	Manual Mode Over-ride Set/Reset: Arg: 1 = Set; 0 = Reset
13H	33H	Servo Amplifier (BDS3) Reset; Arg: any value
14H	34H	Soft Reset; Arg: any value
5XH	5XH	Master Reset; Arg: any value X: any value

As shown by the table above, five types of commands are detected; each type, identified by a unique address, causes a different control action. The command addresses and types are:

The S104 separates the command stream on the basis of Address; Focus commands are input to only the Focus Control microprocessor and Rotation Commands are input to only the Rotation Control microprocessor.

DMA0 Interrupt Subroutine (continued)

Position Commands are the normal observing commands which cause the drives to be positioned at the command argument value.

Nap Mode Set/Reset Command is a special-purpose command which causes the control firmware to ignore position commands when the Nap Mode is Set. The Nap Mode Reset reverses the action. A Soft Reset command also resets the Nap Mode. The Nap Mode is a special mode which is used to deactivate one drive axis if there is a malfunction but permits the other drive axis to remain active; this permits partial use of the antenna until the problem can be fixed.

Manual Mode Over-ride Set-Reset command permits the S101 front panel Man - CMP switch to be over-ridden if the switch is in the Man position. This command is used to cause the control firmware to ignore the effect of this switch if it should have been inadvertently left in the Man position without a telescope operator at the VLBA site. Like the Nap Mode Set-Reset, this is a special mode used only to over-ride the S101 mode control switch. This command mode poses a risk; the CMP - Man switch could be in the Man position because someone is working on the system. An over-ride could pose a hazard to these people.

The Servo Amplifier Reset command causes a hardware reset of the BDS3 servo amplifiers digital logic in the event that the amplifiers signal a fault condition. The reset will clear the fault which might be the result of a power glitch, a temporary over-load condition in the mechanical drive or a more serious malfunction. The fault may return after the reset but this command may save a service call when the VLBA antenna station is unmanned.

The Soft Reset command causes the control firmware to be re-initialized, just as if it was the result of a power reset. This command is used if there is a suspicion that a power glitch has perturbed the operation of the control system. Like the servo amplifier reset described above, this command may eliminate the need for a service call to the antenna.

The Master Reset command is decoded by the S104 and induces a 50 ms reset pulse which drives both control microprocessors in S101. This command is not decoded by the firmware. This command is the most forceful remote action possible and is used to attempt to clear system hardware or code lock-up faults.

The DMA0 (FLM 7, [F342]) subroutine is entered in response to an interrupt from the DMA controller which signifies that four command arguments have been stored in RAM memory in the RAC, CONL, CONM and ACKF locations. The subroutine must examine the values and take the control action designated by the address and arguments; the addresses, actions and arguments were described above.

Like the interrupt-service subroutines described above, the DMA0 subroutine is entered by an interrupt, the DMA 0 Interrupt from the DMA Controller. Program control is vectored by the low memory interrupt table to this subroutine. The AX, DX, CX, BX and SI registers and the BFLAG are used in this subroutine; they are pushed onto the STACK for restoration of their states to their pre-interrupt condition when leaving this subroutine. The BFLAG is cleared in preparation for a bound test of the command address. BL is loaded with the 8-bit address stored in RAC. BH is cleared for the next instruction, the BOUND BX,CS:CMDRG. This instruction, similar in usage to other BOUND examples, compares the value in BX with the CMDRG table [851] values of 16 and 20 (decimal or 10H and 14H); the hex values correspond to the address range cited above for the Focus commands. Since the command multiplex address is always a small value, the BOUND instruction may be safely used without error. Note the 2E CS register over-ride code in the op code; this forces the CS to be the reference base for the displacement of 05CF which is the offset of the CMDRG table. Why do a BOUND test on the command address?

Only five command addresses are valid; commands having an address outside this small set should be rejected. A telescope operator could have manually entered an erroneous command. Secondly, a power glitch could have perturbed the command message transmission or the S104 logic causing an erroneous command address which would probably be outside the proper range; this instruction tests for this possibility. It is good programming practice to test all inputs; the firmware should not attempt to execute out-of-range commands. If the BOUND instruction caused an interrupt, the BFLAG will be set (to 1). TEST BFLAG,1 tests this flag and if the ZF is set, the (address range ok) JZ DMA01 transfers control to DMA01. If an out-of-bounds address, the 32 (decimal, or bit 5, the command fault flag) is set in FAUL1 by or-ing it into FAUL1 where it can be read out as monitor data. No more command action should be taken since the address is erroneous; control is transferred to DEXIT via the JMP DEXIT to re-initialize the DMA 0 channel.

At DMA01, the FAUL1 flag is cleared by and-ing FAUL1 with the 0FDFFH mask; this clears bit 5 (it could have been set in a previous DMA0 subroutine cycle). Having determined the address to be within the valid range, the subroutine must decode the address to determine the command action to be taken. The address is turned into an index by SUB BX,CS:CMDRG which subtracts the address from the lower bound value of 16 (decimal) in the CMDRG table; the results remain in BX. The value is doubled by ADD BX,BX. MOV AX,CS:CMDTABL[BX] is a based, indexed instruction which moves into AX, the value equivalent to the CMDTABL entry by the index in BX. The base is CS; note the 2E segment override prefix which makes CS the base register rather than SI or DI. This is a memory to register instruction with w = 1 and d = 1; the Source operand (i.e. CS:CMDTBL[BX]) is the EA and the Destination operand is the AX register. The displacement is 05D3, the offset address of the first element of CMDTBL. Note the symbols in CMDTABL: POSCMD (pointing to 00EA); NAPCMD (pointing to 0121); MANCMD (pointing to 0142); BDSCMD (pointing to 01C5) and RECMD which points to 011A. This notation is a more general, symbolic way to load AX with a jump address to a command processing entry point; an absolute address could have been used which superficially looks simpler. The value of this approach is that by doing the load symbolically, rather than by an absolute address, the program may be changed without having to manually update the absolute jump addresses. JMP AX transfers control to the command processing entry point designated by the index in BX; the subroutine has identified the command type using the address as an index.

We will now consider the subroutine action for each type of command beginning with POSCMD which positions the drive at the point commanded by the command argument.

The program next performs a range test of the command arguments; commands outside the software limits will not be executed. MOV AX,CONL loads AX with the contents of CONL and CONM [F200] in RAM where they were stored by the DMA 0 channel. CONL contains the lower byte and CONM contains the upper byte. The natural choice for a comparison of the command position argument is the BOUND instruction, but there is a problem. The command argument value can range from 0000H to FFFFH (65535 decimal). Command argument values greater than 7FFFH (32767 decimal) cannot be used in this format with BOUND because they appear to be negative values since the most significant bit is a 1. To get around this problem three XOR 8000H transformations are performed on the argument and limits. The first transformation is to XOR 8000H the command argument in AX. The instruction: DMA0 XOR,8000H inverts the most significant bit of the argument, thus converting the value from the Unsigned Word format to an apparent Signed Word (2's complement) format. Referring to the POSRG table ([F878] on FLM 16), we see the symbols LOW and HIGH rather than the numeric limit values we have seen in the other applications of the BOUND instruction. The LOW and HIGH limit values are transformed on the PROGRAM EQUATES table (FLM 5) by the pseudo-instructions: LOW: .EQUAL 00A00H .XOR. 8000H and HIGH: .EQUAL 0F200H .XOR. 8000H. The first transformation produces a value for LOW of: 8A00H and the second produces a value of 7200H for HIGH. The reader should verify

these transformations. Now consider a command argument of 9FFFH (decimal 40959) which is within the command limits but greater than useable with BOUND. The transformation of this command argument produces 01777H. With these transformations, the BOUND comparison of this 9FFFH value is (LOW < ARG < HIGH): 8A00H < 01777H < 7200H. It's a little easier to see in binary format (remember bit 15 is the sign bit): 1000 1010 0000 0000 < 0001 1111 1111 111 < 0111 0010 0000 0000. Clearly this command argument is within limits. Let's consider a case in which the command argument is less than the lower limit, for example 0900H. The XOR 8000H transformation produces an argument of 8900H. Clearly 8900H < 8A00 so the BOUND instruction will detect the out-of-range condition. Consider for example a command argument value which exceeds the upper limit: F700H which transforms to 7700H. Clearly 7700H > 7200H so BOUND will detect this out-of-range condition.

As demonstrated in the paragraph above, BOUND AX,CS:POSRG compares the argument in AX against the limit values pointed to by the POSRG table, [F878]. If the command argument is outside the POSRG table bounds, the BFLAG will be set by the BOUND interrupt subroutine described above. The XOR AX,8000H returns the AX msb to its original value. The TEST BFLAG,1 tests the BFLAG for the 1-state, (i.e. the BOUND subroutine was entered; an out-of-bounds command argument was detected). If the BFLAG is a 0, the ZF will be cleared by the TEST instruction and control is transferred to POSCM1 by the JMP POSCM1. If BFLAG is a 1, another case of operator error has been discovered; OR FAUL1,128 or's the 128 (bit 7) bit into FAUL1 for subsequent readout in the monitor data. In this event, no further control processing should be done and the JMP DEXIT transfers control to the DEXIT location which re-initializes the DMA 0 channel IPI registers.

At POSCM1, the AND FAUL1,7FH instruction clears the operator fault flag (bit 7) by and-ing the 07F mask with FAUL1; the upper byte will be cleared to 00 and bit 7 of the lower byte will be cleared. The 1's of 7FH keep intact all other bits of the lower byte of FAUL1. AX still contains the command argument; it is next stored in COMTMP, [F203] a temporary RAM location for further processing. The DRVREQ (drive request) flag is set to the 1 state to signal that a new position command has been requested. This flag will be tested by BOSS. The MOV DRVONE,0 instruction clears the second try flag; the usage of this flag is described in MOTION which is part of the TMR2 subroutine. JMP SHORT DEXIT transfers control to DEXIT. Note that this is a SHORT jump; the target location is within 127 bytes of the current offset. Note the op code: EB 47; EB designates an intra-segment direct jump; 47 designates the displacement. At this jump (IP) is 011A, DEXIT is at 0161, the difference (Hex) is 47 which is the displacement shown in the op code.

At RECMD, [F381] the soft reset command is processed - a simple operation. The RESCMD flag is set to 1 and a short jump to DEXIT ends the processing. BOSS will test this RESCMD flag and execute the soft reset in an orderly manner.

At NAPCMD, [F385] the command argument is tested to see if the command processing should go into the NAP mode so as to ignore subsequent position commands until reset by a Nap Reset command. TEST CONL,1 tests the lower byte of the command argument (loaded by DMA 0 as described above); if it is a 1, the ZF is cleared to zero. If the ZF = 0, the Nap mode is commanded and control is transferred to NAPM1 by the JNZ NAPM1 command. In the event that the command argument lower byte is a 0 (i.e. indicating a Nap reset command), the NAPACTV flag is cleared which signals to BOSS that position commands are to be executed. AND SYSTEM,0FEH clears bit 0 in this system status value. When SYSTEM is read out as monitor data, it will show that the Nap mode is inactive. Having cleared this Nap mode bit in SYSTEM, control is transferred to DEXIT by the JMP SHORT DEXIT.

At NAPCM1: [F391] the NAVACTV flag is tested to see if the Nap mode had been previously commanded; the logic of this test is identical to the TEST CONL,1 instruction above. If it had been

previously set, control is transferred to DEXIT by a SHORT jump. If not; the NAPREQ flag is set to 1. Why not set Nap mode to 1 immediately? At this point the subroutine does not know the state of the BOSS loop or FRM drive; if it is moving, it must first be brought to a stop by a safe shut-down; therefore the NAPREQ (Nap Request) flag is set and the NAPACTV flag will be set to 1 by BOSS when it is appropriate to make the mode transition. Control is transferred to DEXIT via a SHORT JMP.

At MANCMD: [F396] the command argument is tested (just as in the Nap command case above) to see if the command argument is a 1. If this is the case, the ZF will be cleared to a zero which indicates that the set manual override mode is commanded. (Bit 3, the Manual Override flag bit in SYSTEM, will be set at [F396] on FUM 8.) JNZ MANCM1 tests the state of the ZF. If a 1, it indicates that the MANOVR flag should be cleared (to 0) so that when BOSS tests the state of the S101 Man - CMP switch, it will execute position commands only if the switch is in the CMP position. AND SYSTEM,0F7 permits the lower 3 (i.e. Nap flag, Cmp-Man flag and Brake status) to remain a 1 but clears the manual override flag.

Having cleared the MANOVR flag to 0, control is transferred to DEXIT.

At MANCM1:, a 1 is set into the MANOVR flag to indicate that the Man state of the S101 mode switch should be ignored by BOSS; position commands should be executed. After setting this flag, control is transferred to DEXIT.

The transition into the manual command override mode (i.e. setting the manual override in SYSTEM) is made in BOSS when the drive is not active; this mode change should only be made when the Focus drive is quiescent.

At BDSCMD: [F405] a servo amplifier reset command is to be executed. The BDSRST flag is set to 1. This is a request flag; when BOSS tests the state of this flag it will be executed in the context of the control mode and state of command execution. Obviously the command subroutine should not directly reset the servo amplifiers; they could be driving the FRM and an amplifier reset could possibly damage them or the FRM. When moving a drive, the amplifier's power load can be high and there are large inertial loads. BOSS executes the BDS3 reset when the drive is quiescent. Having set the BDSRST flag, control falls through to DEXIT which re-initializes the DMA 0 registers in the IPI. Remember that a table, DSETUP [F866] contains the values to be loaded into the DMA IPI registers. DEXIT: MOV SI,CS:OFFSET DSETUP loads the offset address of DSETUP into the SI register using the assembler OFFSET pseudo-operation. MOV DX,DOSPL loads the value equivalent (in the DMA equates table) into AX. This value is FFC0, the address of the DMA 0 source pointer register in the IPI. CX is next initialized to a loop count of 6. DSET: OUTSW causes the contents of the DSETUP table to be output to the address pointed to by the contents of DX. DX is then incremented by 2 so as to point to the next DMA 0 register. LOOP DSET decrements CX and if it is not 0, transfers control to DSET: OUTSW which outputs the next DSETUP table value to the next DMA 0 register.

When (CX) = 0, the looping terminates, the six DMA 0 control registers have been re-initialized, and 10 (decimal) is loaded into AX. The DMA 0 interrupt must also be re-initialized; the EOI register in the IPI must be loaded with a value which resets the IS bit of the DMA 0 Control Register. The EOI value for DMA 0 is 10 (decimal); this is the Interrupt Type number for DMA 0. MOV DX,EOI loads DX with the address equivalent to EOI; from the Interrupt Equates table, the value is 0FF22H, the address of the EOI register in the IPI. The now-familiar OUT DX,AX stores the 10 value in this EOI register.

The BFLAG, SI, BX, CX, DX and AX values are popped off the STACK to restore these registers to the pre-DMA 0 interrupt condition. IRET pops the IP, CS and FLAGS registers off the STACK to complete

the subroutine return.

A final comment on loading the DSETUP table into the DMA 0 control registers in the IPI. The address of the DSETUP table was loaded into AX by the instruction: MOV SI,CE:OFFSET DSETUP. The end result is functionally identical to that used in INITIALIZATION (see above) by the instruction: MOV SI,OFFSET DSETUP. Why the different instruction usage? The DSETUP table is in the lower EPROM at offset address 05E1; in the DMA 0 subroutine this offset is relative to the CS register. The op code for the DMA 0 case is: BE E1 05. BE is the opcode for an immediate operand to register with W = 1 (i.e. a word operation) and the reg is 110: the SI register. The displacement is 05E1 which is the offset address of DSETUP. For the INITIALIAZATION case, the op code is: BE 00 00. Why the difference? In this case the assembler OFFSET operator supplies the address of the DSETUP table.

BOSS Control Program (FUM 7, RUM 7)

The BOSS program (FUM 7, [373]) is the top level control program which recurrently tests conditions to determine which major action should be taken next. Like a true executive, BOSS actually does not perform any task other than supervisory status testing and task assignments. BOSS operates in conjunction with LOCAL, CHKDRV and TMR2 to control the position of the FRM drive. BOSS arranges for the logical and orderly transitions of control states.

BOSS begins by accessing the RANDOM value generated in DSTOR to execute a randomly ordered delay loop. It is possible for the data multiplexing/conversion operation in S102 and the cycle rate of the BOSS call for DSTOR to approach synchronism; this phenomena tends to intensify the processing rate of some S102 data channels at the expense of other channels. All analog data channels need to be processed at the same rate. By executing a randomly ordered delay at the start of the BOSS loop, it is possible to dither the timing so that quasi-synchronous operation of BOSS and DSTOR is avoided. The MOV ACAX, RANDOM instruction loads CX with the RANDOM value which is the loop count for the LOOP instruction. One cycle in the WAIT: LOOP WAIT loop takes about 4 usec. The RANDOM value can range from 1 to 64; thus the time delay of the wait loop can be as short as 1 usec, as long as 256 usec, and the average delay will be 128 usec. We are now familiar with the LOOP instruction; CX is decremented and control is transferred to the target operand (WAIT:) as long as CX is not zero; when CX becomes zero, control passes to the next instruction.

The STACK Pointer is re-initialized to the value 2100H, the top address of RAM. In this reinitialization the assumption is made that nothing currently in the STACK needs to be recovered; subsequent pushes onto the STACK will destroy the contents of the locations being written into.

The next test is to see if there are S102 faults; the first (least significant) bit in the FAUL1 monitor data word is tested to see if the S102 responded to the most DSTOR request. If the ZF is a 1, the S102 fault bit in FAUL1 is a 0 which means that S102 was responsive to the most recent DSTOR. If the ZF is a 0, it means that S102 is apparently inoperative; nothing further can be done as data from S102 is vital to FRM control. Control is transferred to RSCMD (further down BOSS to [F438]) to clear the current command, if active.

The next operation is to test the DRVATV (drive active) flag [F381] to see if a command is moving the FRM. If the ZF is a 1, the drive is inactive and control is transferred to NEXTO:. If the drive is active CHKDRV is called. CHKDRV is an important subroutine which performs a motion analysis and 2nd screw motion test. CHKDRV is described below. Upon return from the CHKDRV subroutine, control is returned to the start of the BOSS loop by an unconditional jump.

To protect the Focus drive and servo amplifiers, all control and mode change operations of the BOSS loop must be done with the drive quiescent; a sudden change in direction of motion or some other modal change could seriously stress the FRM and servo amplifiers. The BOSS code below performs these mode and command executions under safe conditions.

At this point ([385] in FUM 7), the Focus code differs from the Rotation code by testing the Mode switches to determine whether the Focus second-screw code should be bypassed for test purposes. This bypass feature is to be used for test purposes only; in operational service the second screw performs a vital role in protecting the expensive and delicate FRM from potential damage because of some malfunction in the Focus drive. If the drive is quiescent, the next test is to read the state of the two hex Mode switches on the S101 front panel. These switches provide the facility for operator intervention in program execution. The switch settings and commanded program actions are tabulated in Section 3.1. One byte is read from the two switches and both the Focus and Rotation programs read the switch status. The switches are not particularized to a control program; each program has an identical interpretation of the switch settings.

MOV DX, MODESW, IN AL, DX reads the switch state into AL; we have seen this instruction pair many times by now. Since the switch outputs are low true, the switch data is inverted by the NOT AL instruction. Since this is Focus code, an important aspect of execution is the second screw analysis. The mode switch provides the facility to bypass this 2nd screw test by setting the switch to the FF state. CMP AL, 0FFH compares AL with the immediate operand 0FFH. The comparison is a subtraction operation and sets flags to indicate the results. The ZF will be tested but first the SCWIGN (screw ignore) flag is cleared. If the switch state is FF, the comparison sets the ZF. JNZ BOSS1 transfers control to BOSS1 if the ZF is a 0; the switch setting was not FF. If the states match, the SCWIGN flag is set (to 1).

The Rotation BOSS program does not contain the Mode switch code described above.

BOSS1 [F393] calls DSTOR for new Apex data. Upon return from DSTOR, the MANOVR (manual override command) flag is tested by the TEST MANOVR,1 instruction. (This request flag is set if the Antenna Control Computer sends a command to override the MAN setting of the S101 front panel MAN - CMP switch; the effect of the command is to inhibit the LOCAL mode commanded by the switch when it is in the MAN position and force the CMP or computer control mode.) The override command interpretation was discussed in the DMA0 subroutine description above.) The TEST MANOVR,1 is a logic and test; if both bits are a 1, the ZF is cleared to 0 and the JZ NEXT1A will not be executed. The manual command override flag must be set in SYSTEM to reflect the new mode. OR SYSTEM, 8 or's this 4th bit into SYSTEM which is the mode control status word.

We might ask the question: why set this mode bit in BOSS? Shouldn't this have been done in the DMA0 subroutine where the mode change was decoded? The answer is that the DMA0 subroutine can be called at any time and is independent of operating mode. Any major mode change which could affect the Focus drive must be made when the drive is physically quiescent; this is best done at this place in BOSS where the drive conditions are known.

If we are not in the manual command override mode, JZ transfers control to NEXT1A: [F398]. NEXT1A: IN AL,P2PTA reads the state of EPROM2 into AL to test the state of the CMP - MAN switch. The next instruction, AND AL, 20H tests the state of the COMP MODE discrete on Port B5. (This bit is high-true when the CMP-MAN switch is in the CMP position.) All the other bits in AL are cleared by the AND. If the COMP MODE bit is a 1, the ZF is cleared and control is transferred to NEXT1. If the ZF = 1, control falls through the JNZ NEXT1 instruction to execute the JMP LOCAL instruction which permits

manual control of the FRM and adjustment of the S101 A/D and D/A converters. LOCAL ([F624] on FUM 12) is a major branch of the control program; manually-input control states cause the Focus drive to be moved in a slewing mode when a control switch is actuated. This control path is described below.

Since the FRM Focus drive is quiescent and we are not in LOCAL control mode, it is safe to reset the BDRS3 servo amplifier if a command to do so had been sent by the Antenna Control Computer. This This command would have set the BDSRST flag. At NEXT1, the TEST BDSRST,1 instruction tests the state of this request flag and if set, the ZF is cleared and the program calls the BDS3RS subroutine after clearing the BDSRST flag. After return from the servo amp reset subroutine, CX is loaded with a 1 for a 1-second period and the DELAY subroutine is called to provide a 1 second delay to permit the servo amplifier logic to stabilize and the shaft position R/D converter readout to stabilize at the correct position. CALL TMROFF stops the timers and re-initializes them for the next timing cycle. Having reset the servo amplifiers, control is transferred to the start of BOSS by an unconditional jump.

At NEXT2: [F411] BOSS services the soft reset command. As was the case with the manual override and BDS3 reset commands, this soft reset is only initiated when the drive is quiescent in BOSS. The RSCMD (reset command request) flag was set by the DMA0 subroutine when this modal command was detected. The state of the RSCMD flag is tested and if set, a jump to CMDRS (described below) is executed. If not set JZ NEXT3 is executed.

At NEXT3: [F415], BOSS services the nap mode state set by a previous pass through BOSS. As in the amplifier reset and soft reset cases just above, the NAPATV (nap active) mode flag is tested; if set, the DRVREQ (drive request) flag is cleared (see the code immediately below) and control reverts to the start of BOSS. We will not test to see if a new position command needs to be executed. If the drive request flag is set, control transfers to NEXT4: via the JZ NEXT4 instruction.

The reader should recognize the important logical distinctions between drive request and drive active flags. Examples of the two types are: DRVREQ and DRVATV flags and the NAPREQ and NAPATV flags. A request flag causes the requested mode to be set by BOSS (when conditions are appropriate to do so) and the active flag indicates that the mode has been entered. When a mode is entered, the associated request flag is cleared. Another important requirement is that the code which services the request and mode flags must be disjoint; we cannot tolerate a case where request and active flags are true at any time. For example, it is obvious that we must not be in both the nap and drive active modes at any given time or immediately reverse direction when a new overriding position command is received while executing a position command. We should not assume that every command coming from the Antenna Control Computer is logical and must be immediately executed; a telescope operator could manually intervene in the antenna operating program by manually commanding a nap command while the drive is moving or even send out a drive command for the opposite direction while the drive is actively moving. These and other equally silly operator-commanded situations have occurred and doubtless will happen in the VLBA. The BOSS program is carefully structured to eliminate the possibility of logical conflict between mode requests and mode states.

NEXT4: [F420] tests the DRVREQ (drive request) flag which signals that a new position command has been received from the Antenna Control Computer. The code sequence is structured so that it can only test this DRVREQ request flag if it is not in the nap mode. If the DRVREQ flag is not set, control is transferred to NEXT5: which services the nap request flag.

If the DRVREQ flag is set, the DRVINT (drive initiate, FUM 9, [F482]) subroutine is called which starts the execution of a position command by determining the direction to drive, how far it should go, calculates the drive ramping parameters, etc. Upon return from DRVINT, control is transferred to the start

of BOSS (FUM 7, [F373]) for another logical scan.

NEXT5: tests the NAPREQ (nap request) flag; if it is not set, control is transferred to the start of BOSS. If the nap request flag is set, control is transferred to NEXT6:.

NEXT6: [F429] establishes the nap mode and clears the nap request flag. MOV NAPATV, 0FFH sets the NAPATV flag. OR SYSTEM,1 sets the nap active flag in SYSTEM. SYSTEM reports the state of the Focus program to the Antenna Control Computer via the monitor data readout. After setting the NAPATV flag in SYSTEM, control is transferred to the start of BOSS.

CMDRS (Command Reset) Subroutine (FUM 8, RUM 8)

This subroutine is entered to execute two types of command resets and has two entry points: CMDRS is the entry point from BOSS [F414] and RSCMD (FUM 17, [F916]) is the entry point from BRKOFF [F916] and LOCLB [F635]. The latter entry point results from the discovery (in BRKOFF and LOCAL) that the brake appears to be faulty; clearly the safe course is to shut down the drive via a reset command.

At the CMDRS: entry point, the complicated-looking instruction AND WORD PTR FAUL1, 0FC1FH clears the following fault flag bits in FAUL1: Drive Fault, Brake Fault, Operator Fault, Monitor Fault and Command Fault. All other fault flag bits are left intact; why clear only these FAUL1 flag bits? If the reader looks at the character of the FAUL1 flags, he will note that only faults resulting from computation are cleared. Hardware-induced faults such as limit switch faults, etc. are not cleared; these fault bits should be cleared only when the code that tests them verifies that they are not faulty. Clearly a soft reset command should not clear hardware fault flags.

AND WORD PTR SYSTEM, 0FFFEH clears the nap flag (only) in SYSTEM, which reports the control mode state to the Antenna Control Computer via monitor data. The nap mode active bit in SYSTEM is also cleared in the NAPCMD: (FLM 7, [F385]) code in DMA0.

The two instructions above are peculiar to a soft reset command which only involves clearing flags. The following code deals with shutting down the Focus drive if it is in motion.

AND WORD PTR SYSTEM, 0FFF5H clears the manual mode and manual command override bits in SYSTEM; all other bits are left intact. The STACK pointer is reset to 2100H; the old data in the STACK will be overwritten in subsequent pushes but since this is a reset it doesn't matter.

The DRVOFF flag is set to 1; this flag indicates that the drive is in the process of being shut down. The Focus drive could still be driving; DRVATV is tested and if it is a 0 (drive is inactive), the JZ RSCMD transfers control to RSCMD1 to clear flags and engage the brake. If still active, control is transferred to the start of BOSS.

RSCMD1 [F445] clears AL by an exclusive or in itself; this 0 value will be written into the flags. MOV CX, ENDFLG loads the number of flag locations into CX. Looking at FUM 5 at the flags table we see ENDFLG: .EQUAL $-FLGST; this is an assembler pseudo instruction which subtracts the ENDFLG address from the FLAGST address and leaves the result, the number of addresses (i.e. flags) in ENDFLG. What this pseudo-instruction and the MOV have done is to load CX with a loop count. MOV DI, OFFSET FLAGST loads the DI (Destination Index) register with the starting address of the flag table. Note the use of the OFFSET operator (pseudo-instruction) which returns an address rather than a value. The string instruction: REP STOS BYTE PTR FLAGST loads the AL value (zero) into the bytes pointed to by the

CMDRS Subroutine (continued)

DI register. DI is incremented and CX is decremented by the instruction. When CX = 0, control passes to the CALL BRKPFF subroutine to engage the brake. BRKOFF is described above.

The RSCMD request is cleared. AX is cleared, AL was already cleared in the flag reset operation above but AH could be in an indeterminate state. OUT P2PTA, AX clears all bits of both EPROM2 Ports A and B; some of them are input ports but this instruction does not affect their state. Having cleared the flags, command states of EPROM2 and the RESCMD request flag control reverts to the start of BOSS.

CHKDRV (Check Drive) Subroutine (FUM 9, RUM 8)

The CHKDRV ([F455] on FUM 9) subroutine is CALL-ed near the BOSS loop if the drive is active. CHKDRV serves as an entry point to the TMR2 interrupt subroutine which does the detailed management of Focus drive motion. CHKDRV is the entry point for the initiation of new position commands. In doing so it also checks the new command argument to see if the amount of commanded motion is too small to bother with; if so, the command request is terminated and the flags are cleared.

CHKDRV opens by decrementing a software safety timer, EXTTMR, which is initialized to a count of 25 by the TMR2 interrupt routine as a backup timed clock in the event that somehow the TMR2 interrupt is missed. When the drive is active (DRVATV = 1), the cycle rate of BOSS through CHKDRV is such that the 25-cycle period is a few hundred usec longer than the TMR2 interrupt; thus TMR2 should always be activated. In the event that the EXTTMR count is decremented to zero before the TMR2 interrupt, control falls to the INT 3 instruction which induces the TMR2 interrupt subroutine via the INT 3 instruction. (The TMR2 interrupt subroutine is described below; the brief description above outlines its function.)

If EXTTMR is not zero, CHKDR1: tests the state of the DRVREQ (Drive Request) flag to see if a new position command has been requested; CHKDRV is the entry point for new position commands. If there is not a pending command request, control reverts to the start of BOSS by a RET instruction (CHKDRV was invoked by a CALL instruction) to a JMP BOSS instruction in NEXT.

In the event that there is a pending DRVREQ position command request, CHKDR1: MOV CHKDRV,0 clears the flag and tests the new command argument standing in CONTMP (Command Temporary storage) with the most recent command argument standing in POSCEC (Position Command Echo storage) by loading BX with the echo value and subtracting the new command value; the subtraction results are in BX. If the difference between the arguments produces a negative result, the contents of BX will be a Signed Word format which is a requirement of the BOUND instruction. The BFLAG (BOUND flag) is cleared to 0 and the BOUND BX,CS:CLSETAB instruction compares the command argument difference in BX with the two limit values in the CLSETAB table ([F972], FUM 18). If the difference is large (irrespective of the sign) BOUND will flag the out-of-limits case. The BFLG is tested to see if the amount of commanded motion exceeded the CLSETAB limits. If so, BFLG = 1 and control is transferred to CHKDR2. If BFLG = 0, the temporary command position value (COMTMP) is cleared and a RET instruction returns control to the start of BOSS (via JMP BOSS) in NEXT.

The CLSETAB values are: 0FFF4H and 0000CH. If the difference does not exceed 12 counts (decimal), the program ignores the command because the physical change is too small to bother with. How much is this physical distance? Remember that Focus position is a 14-bit value read out as a left-shifted 16 bit word. 12 counts/4 (remembering the 2-bit left shift) is 3 counts of the 14 bit value. One 14-bit count is equivalent to 0.0017 inches; so position commands requiring a motion less than 0.0054 inches will be ignored. This is a very small amount of motion. This motion testing function also screens out repeated commands to the same set point; sometimes systems programmers repeat commands believing

that doing so enhances the probability of proper execution.

CHKDR2: sets the DRVOFF (Drive Off) flag and sets the DRVREQ flag (it was cleared above); DRVREQ must be set for BOSS to call DRVINT which does the calculations for the commanded motion. (DRVOFF is tested in the TMR2 subroutine and, if set, will shut down the drive.) DRVATV is tested to see if the drive is still active (remember, we got to CHKDRV from BOSS by a DRVATV test); if the drive has stopped (at least from the visibility of the CHKDRV subroutine), control is transferred to CHKDR3 to clear all the flags. If the drive is still active, control is returned to NEXT near the top of the BOSS loop by the RET instruction.

CHKDRV3: clears the flags table with code identical to that described in RSCMD1 above. After clearing the flags, control reverts to the BOSS loop via the RET instruction.

DRVINT (Drive Initiate) Subroutine (FUM 9, RUM 9)

The DRVINT ([F482], FUM 9) subroutine is CALL-ed from BOSS when it sees a DRVREQ flag (at [F423]) in BOSS. This subroutine is invoked only when a new position command is to be executed. The CHKDRV subroutine will have already tested the commanded motion to see if it is too small to bother with. This extensive, subroutine performs motion control calculations to determine direction and distance to drive and ramp break points.

DRVINT is the entry point ([F482] for Focus and [R467] for Rotation) for commands from the antenna control computer. The entry point for manually generated commands is LOCTST ([F502] for Focus and [R485] for Rotation).

The first action is to cancel the drive request; this is common to both the Focus and Rotation programs.

At this point the Focus program (only) tests the state of the SECCH (second chance) flag by a CMP BYTE PTR SECCHC,2 instruction. If there have been two consecutive second screw analysis motion faults, the JE DRVSTP causes control to be transferred to DRVSTP [488] which RET-urns control to NEXT4 [F424] in BOSS. Focus is now inhibited from executing any more position commands until either a soft or hard reset command is executed.

The next step (for both programs) is a test of the state of the Drive Lockout and Emergency Stop bits read and stored in FAUL1 by DSTOR9. Early in BOSS (at BOSS1, [F393]), DSTOR is called to acquire updated system and Apex status, so the Lockout and Emergency Stop data is recent. The TEST WORD PTR FAUL1, 1800 instruction forms the logical and of (only) these two bits; if the product of either test bit and corresponding FAUL1 bit is true, the ZF will be 0. JZ DRVCNT tests the ZF and if not set, the command initialization may continue; if a fault exists the RET instruction returns control to the BOSS program at [409] in NEXT4.

In DRVCNT [F489] the first action is to clear a portion of the flag table by a string write of the zero-ed AL to a selected subset of the table. The operation of clearing the whole flag table was described above in the DMA0 (RSCMD1 section) subroutine description. In this case we need to determine the number and location of the flags that must be cleared. Reference to the flag table shows that the last four flags must be cleared (i.e. DRVATV, RAMPOS, DIR and LDIR) because we are starting to execute a position command. During the course of executing the DRVINT subroutine, these flags will have new values set into them. The OFFSET operator in MOV CX,OFFSET FLAGST+ENDFLG-DRVATV returns not an address (which we have seen in the RSCMD1 code above) but a value to CX which is the number of

cycles required to clear the last four flag locations in the flag table. This instruction is a demonstration of the use of assembler arithmetic using the OFFSET operator. MOV DI, OFFSET DRVATV returns to DI (Destination Index register) the table address of DRVATV (in the DS memory). REP STOS BYTE PTR DRVATV will store the zero-ed AL byte in the location pointed to by DI (set to DRVATV); after storage, the CPU will increment DI and decrement CX. When (CX) = 0, control passes to the next instruction.

After clearing a portion of the flag table, the Apex (FAUL1) and servo amplifier (BDS3) fault words are cleared. The DMA0 code, in processing a new position command, stored the command argument in COMTMP, a temporary memory location. From the visibility of DMA0, it is not clear that the new command will be executed (the delta motion could be too small to pass CHKDRV). At this juncture it is appropriate to make this tentative COMTMP argument be the formal command argument by storing it in POSCEC, the position command echo argument which is the value used throughout the balance of the (computer mode) position control program. Although it is never referenced again, COMTMP is cleared because its the right thing to do . Finally, DSTOR is called to obtain new Apex data.

The DRVINT subroutine was entered by a CALL from Boss and will return to BOSS via a RET instruction after testing states and determining position control parameters for the impending motion. At LOCTST processing continues in starting up a position command but at this point processing is not peculiar to the computer control mode. LOCTST is also the entry point of a CALL from the LOCAL; if LOCTST was entered from LOCAL, the RET instructions will return control to the LOCAL code rather than the computer mode code. The RET distinction is determined by the value of IP which is pushed onto the stack by the CALL instruction. If called by LOCAL, it will pop the IP value for the next instruction following the CALL; the same is true from the CALL from BOSS at [F423]. The reader should keep in mind the dual-usage of this code.

At LOCTST, the Focus program (only) tests the state of the SECCH flag by a CMP BYTE PTR SECCH,2 instruction, just as was done at the computer command entry point. If the SECCH flag is a 2, the JE LOCSTP jump transfers control to LOCSTP which RET-urns control to NEXT4 in BOSS. It is not now possible to manually command the Focus drive in the LOCAL mode without a soft or hard reset.

LOCTST tests the state of the Lockout and Emergency stop bits by code identical to that described above in DRVINT [F482] above. If faults are present, an RET returns control to the CALL-ing location and nothing further happens to start motion.

At LOCCNT: the Rotation code tests the fault states of the two BDS3 servo amplifiers in FAUL2 by a CMP WORD PTR FAUL2, 4848 instruction. The 4848 bit pattern tests the "Drive Up" (i.e. ready to run) and "Remote Inhibited" bits of both amplifiers. This is the correct state at this juncture. The inhibit will be released later in DRVINC [F618] at which time both fault bits should go false. The Focus code uses a single byte 48 bit pattern for this test because there is only one servo amplifier. In the event that the CMP 4848 bit pattern is not matched by the bits in FAUL2, the ZF will be set and control will return to the calling location by a RET instruction; the existence of faults inhibits further execution of the DRVINT code. The Focus code tests the state of the one Focus servo amplifier by a CMP FAUL2, 48H instruction. Other than the fact that only one amplifier is tested, the code is identical for both programs.

DRVIN0 [F511] tests to see if the position error is too small to bother with; a similar test was performed in CHKDRV in which the argument of a new position command, COMTMP (Command Temporary) is less.than a small delta different from the previous command argument, POSCEC (position command echo). The comparison is performed by a BOUND instruction and the logic is identical to that described in CHKDRV above. In the case of Rotation, the tolerance limits are 8 counts; this is equivalent to a tolerance band of about 3.1 arc-minutes. In the case of Focus the corresponding tolerance limits are

0.0054 inches. If the commanded position change is too small, the ZF will be set and control will fall through the JNZ DRVIN1 instruction to return to the CALL-ing location via the RET instruction and execution will be terminated.

In the Rotation code at DRVIN1, the lower byte of the FAUL1 word is loaded into AL and then tests the DIR bit and limit switch fault bits. Only the limit switches associated with the indicated motion direction will be tested. The switch circuits generate a 1 if they are not actuated. DIR is set by DSTOR and indicates which direction the drive should move to null the error. If DIR = 1, the drive should move CCW to null the error. TEST DIR,1 performs a logical and of the DIR flag and the 1 bit; if the DIR flag is not set (indicating a drive CW direction), the ZF will be 1 and the jump to DRVIN2 will test the CW limit switches. If Dir is 0, the CCW limit switch bits are tested.

The instruction AND AL, 24 [501] performs a logical and of the 1st and 2nd CCW limit switch bits in FAUL1. The 24 value (bit weights 8 & 16) correspond to the CCW limit switch bits in an 8-bit format. These bit weights designate the bits to be tested in the lower byte of FAUL1. If the logic product of both switch bits with the 8 and 16 bits is true, the ZF will be 0. If either switch is actuated, the logic product will be a 0 and the ZF will be a 1. If the ZF is a 0, neither of the CCW switches is actuated and control is transferred to DRVIN3. If either switch is actuated, control falls to RET which returns control to the CALL-ing program and no further action is taken to start the drive into motion.

In the Focus code at DRVIN1, the logic of the tests is identical to that described for Rotation, just above but since the order of the Focus UP and DOWN limit switch bits is opposite to the Rotation CW and CCW bits in the two FAUL1 storage words, the 6 and 24 test values are interchanged.

At DRVIN2, [F523] the CW limit switches corresponding to a value of 6 (bit weights 2 and 4) are similarly tested with the AND AL, 6 instruction. Again if either switch is actuated, the RET instruction is executed and no further action is taken to start the drive into motion.

Comparing corresponding limit switch test sections of DRVINT in the Focus and Rotation programs, we see that the correspondence of the limit designations and test bits are reversed between the two code versions. This is a consequence of the way that the switch bits are wired to the data registers in S102. These portions of the two programs are not identical but differ only in the ordering of the test bit weights; otherwise the program logic is identical.

The parameter LDIR (last direction) is initialized next. In the course of driving the mechanism toward the commanded set point, it is possible to over-shoot the set point which will change to sense of DIR. LDIR provides a reference for the convergence code in TMR2. LDIR is initialized to the same sense as DIR.

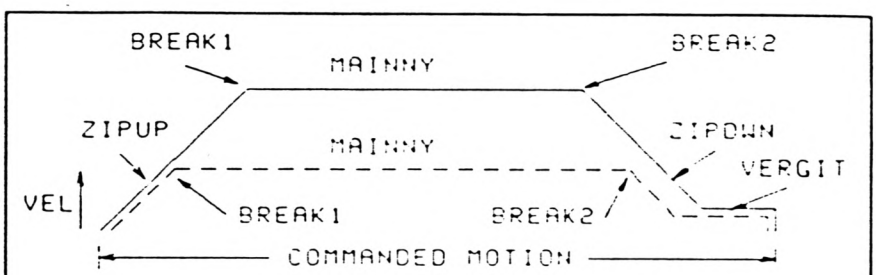
The FRM drives start at zero velocity and are ramped up to a position break point; the drive then operates at a constant velocity. The bulk of motion is realized at the constant velocity; at a second position break point the drive velocity is ramped down to a low value and is driven to the commanded set point at constant velocity. The maximum velocity drive is +/- 2.8 volts; this corresponds to motor speeds of 8.45 rev/sec for Focus and 36.5 rev/sec for Rotation.

The velocity profile is depicted on the next page. Velocity and position parameters calculated in DRVINT are tabulated at the end of the DRVINT description.

The figure below shows four regions labelled: ZIPUP, MAINNY, ZIPDWN and VERGIT; these regions are each driven by a subroutine in TMR2. The RAMPOS flag value is the index used to branch

DRVINT Subroutine (continued)

to these subroutines. The DRVINT subroutine calculates the breakpoints which are the transition points for these subroutines. The TMR2 subroutine manages the motion and tests the current position for transitions to the next region. At the transition the flag value is updated to indicate the new region. The first breakpoint corresponds to a position of 12% of the commanded motion and the second breakpoint corresponds to 80% of the commanded motion. The velocity acceleration and deceleration ramps have a constant slope of 1 volt/sec.



DRIVE REGIME	DRIVE MOTION	RAMPOS
ZIPUP	ACCELERATION	0
MAINNY	CONST VEL	2
ZIPDWN	DECELERATION	4
VERGIT	CONST VEL	6

Drive Profile

The next function of the DRVINT subroutine is to calculate the 12% (Focus, 7% for Rotation) and 80% position break points. This involves multiplication and division; new types of code operations.

The division algorithm is a sequence of subtractions of a fixed point divisor from the dividend resulting in a quotient and remainder. The arithmetic is fixed-point; the quotient is the number of divisor subtractions which may be performed and the remainder is a fixed-point value, less than the divisor, which remains after the last subtraction.

The division operation of DRVINT is a word division; the dividend is in the AX (most significant part) and DX (least significant part). The divisor is in CX. This division operation places the quotient in the AX and remainder in DX. The Flags are not affected by division.

The multiplication operation of DRVINT is a word multiplication. One factor is in AX and the other is in CX; the multiplication operation produces a product in the DX and AX registers. DX contains the most significant part of the product and AX contains the least significant part of the product. The flags are not affected by multiplication.

The reader should review word division and word multiplication in the IPRM.

To calculate the first (12% for Rotation, 7% for Focus) break point, the position error is first divided by 100 and then multiplied by 12 (or 7% respectively for Focus). Although the first break point values are slightly different, the program operations to calculate the breakpoint are identical in the two programs.

DX is set to zero for the impending division; the reason will be evident below.

The position error is loaded into AX and turned into an absolute value. LDIR is tested to determine the direction to drive to null the error. LDIR was set to DIR just above.

Remember that DIR = 1 if the drive must be moved down (in value) to null the error. TEST LDIR,1 sets the ZF if the drive must move up (to a higher value). If the ZF is not set, the error value is set to a positive value by the NEG AX instruction. This makes the contents of AX be the absolute value of position error the negative value was transformed to a positive value. This absolute value is PUSH-ed onto the STACK for later usage. If the ZF is not set by TEST LDIR,1, control is transferred to DRVIN4.

The position error can never exceed a decimal value of 65536 (less if we consider limits) so it is never exceeds the capacity of the AX register. Since this is the most significant portion of the dividend and never exceeds the capacity of a word, the DX part is clearly zero.

All program values in this calculation are decimal as indicated by the absence of the H suffix.

At DRVIN4 [F533] AX is pushed onto the stack. CX, the divisor is loaded with 100. DIV CX divides the AX-DX dividend by the CX divisor. We can picture the division as: {(AX=|error|) + (DX =0)}/(CX=100) = (AX=Quo) + (DX=Rem). The maximum value of the Quotient is 655 (65536/100) and the Remainder will be 99 or less. Numerically the division looks like: 65536/100 = 655 + 36/100. The remainder in DX, the quotient in AX and the DX remainder (again) are pushed onto the STACK for subsequent use.

The quotient in AX is multiplied by 12 in CX. The product is in DX (most significant part) and AX (least significant part). The DX (most significant) component is always zero and is forgotten. We can picture the multiplication as: (AX=Quo)*(DX=12) = (DX=MSPart=0) + (AX=LSPart). The AX (least significant part) component of the product is saved in BX.

The remainder (from the division above) can range from 0 to 99. This Remainder is next popped off the STACK into AX and is also multiplied by CX which is still 12. This second multiplication may pictured as: (AX=Rem)*(CX=12) = (DX=MSPart) + (AX=LSPart). The product can be as large as: 12*99 = 1188.

Finally, to complete the division, the product of the remainder times 12 is divided by 100 and the most significant part added to BX. We can picture this second division as: {(DX=MSPart) + (AX=LSPart)}/(CX=100). Numerically, this is a division of a value ranging from 0 to 1188 by 100. The result could be as large as 11. Clearly the remainder of this second division is less than unity so it may be forgotten.

As a result of the arithmetic above, BX contains a value which is 12 percent of the absolute value of the command position error, (i.e. |commanded position - present position| * 12). Again, remember that the corresponding divisor for Rotation is 7.

At this point in DRVINT ([R528], [F546]), the Focus and Rotation codes differ significantly so we will describe the different implementations separately.

This portion of the Rotation-pecualiar code follows; the end will be noted.

At [R528] a motion parameter, POSDED (old position) is loaded with the current position (we don't care about old position anymore, it is a residue of a previous command). POSDED will be used by the motion analysis code in the TMR2 subroutine.

If the result of the division and multiplication is vanishingly small, the ADD BX,AX operation above will set the ZF. This small result can only be the result of a position command to only a 13 or a

few more counts different than the present position. CHKDRV will reject commands with motions less than 13 counts. If the ZF is not set, control is transferred to JNZ DRVIN5. If BX is zero, 1 is added to it for the BX comparisons below; BX must not have a zero value in these comparisons.

Remember that this code is also used by the LOCAL subroutine to initialize a motion command. The command mode is tested next by DRVIN5: TEST SYSTEM,2 which and's 2 with the manual mode bit (bit 1). The ZF will be set if in computer mode; control is transferred to DRVIN6. If in Manual mode (CMP-MAN switch set to MAN), the 12% break point in BX is compared with 60 by the CMP BX,60 instruction. This instruction subtracts 60 from BX; if BX is > 60, the CF is not set and BX retains its value from above. If BX is 60 or less, a value of 60 is loaded into BX to replace the 12% value calculated above. This latter case is associated with small motions and establishes a first ramp break point of 60 counts, an arbitrary value. In the manual mode we don't want to run as fast as in computer mode; the smaller break point 1 results from this concern.

Now DRVIN6; remember BX is still an absolute value; we must translate this value into an equivalent point in drive position. The direction to drive (LDIR) is tested to see if the drive is to be down; if so the BX value must be subtracted from the present position. If LDIR=1, (i.e. drive down), BX is negated which forms a 2's complement negative value which is added to AX which contains POSD, the current drive position. The result of this subtraction is break point 1 which is 12% (or less) lower that the current position. If LDIR=0, the BX value is added to AX resulting in a break point 1 which is 12% higher than the current position. This break point 1 value is saved in BREAK1 in the RAM.

DRVINT next begins to develop the second break point; the point at which the drive speed ramp-down starts. The ramp down point is conditioned by the magnitude of the position error. At [R542] the position error is loaded into AX which is followed by a direction test of LDIR to develop an absolute value of error by code similar to that used above in preparation for the division-multiplication operations used to calculate BREAK1. After making the AX contents an absolute value (with a positive sign), it is compared with 1000H and 3000H to determine a multiplier to be used in CX. With CX loaded with 40, the first comparison is: CMP AX,1000H in which 1000H is subtracted from AX. If 1000H > AX, the CF is set and the JC DRVIN9 transfers control to DRVIN9, where the second break point calculations begin. If AX > 1000H, control falls through to the next comparison with CX loaded with 60. In this comparison, if AX < 3000H, control is transferred to DRVIN9 with CX loaded with 60. Finally, if AX > 3000H, CX is loaded with a value of 80. Why make the second break point conditional upon the error magnitude? This is a Rotation-peculiar feature which reduces the tendency to over-shoot the command set point during convergence which necessitates a reversal of drive to reach the set point.

At DRVINT9, the STACK is POP-ped onto AX which destroys the present contents of AX, the absolute value of position error but it was saved by being PUSH-ed onto the STACK. What did we pop onto AX? Remember the second PUSH at line [R518] above? That PUSH put the quotient of the first division of the absolute value of position error by 100 onto the stack where it has been waiting to be retrieved for the second set of arithmetic. Refresh your memory by going back to the previous page to review the first division of position error by 100. Using this previous result we multiply the quotient by CX which can be 40, 60 or 80, depending upon the comparisons above. The product is saved in BX.

Another POP loads AX with the remainder of the first division which is again multiplied by CX. CX is next loaded with 100 and AX is divided by CX. The result is added to BX, just as was done in the first set of calculations. Now what do we have in BX? BX contains an absolute value which is 40%, 60% or 80% of the command error.

Another POP loads AX with the absolute value of position error from the STACK; it was PUSHed onto the STACK at line [R514] above at the start of the first set of break-point calculations. The program now modifies the second break point depending on whether the mode is computer or manual. A TEST SYSTEM,2 instruction and JZ perform the test and branch. If in computer mode, control is transferred to DRVINA; if in manual, the calculated break point (in BX) is subtracted from the error (absolute value) in AX. Remember that BX is always less than AX; it is 40%, 60% or 80% of AX so the difference is never negative. The difference remains in AX. The difference is compared with 65 by a CMP AX,65 instruction which subtracts 65 from AX. If AX < 65, the CF is set and control is passed to DRVINA at [R569]. If AX > 65, the error value is put into AX and 65 counts are subtracted from BX which reduces the break point. How is the error put back into AX? Remember we did an AX - BX subtraction, the difference was left in AX. BX was not affected by the subtraction. So adding BX to (AX-BX) leaves us with the original AX value, the position error. Finally, 65 counts is subtracted from BX, the break point 2 value.

Now, what is the reason for these subtractions and comparisons? At [R564] in the manual mode path, we subtracted the break point from the error and then compared the difference to 65 counts. If the difference is greater than 65 counts, control is transferred forward; if less, the error (in AX) is substituted for the calculated break point in BX and 65 counts are subtracted from this new break point. The effect of these calculations and comparisons is to cause the break point to always be at least 65 counts less than the error. The reason for this 65 count reduction in the manual mode is to make an earlier slowdown to VERGIT so as to not overshoot the command set point. Remember that in the manual mode, the command set point is either the HIGH or LOW software limit. Even though the operator probably plans to release the manual slew switch far short of this point, the program has either of these values as the target set point.

What is the Rotation manual velocity? Using the 65 counts and the acceleration ramp of 1 volt/sec^2, which is 400 counts/sec^2, we calculate the (manual mode) Rotation velocity in MAINNY to be 18,360 counts/sec and the DAC drive to be 0.687 volts. In terms of motor shaft rotation this is 8.965 rev/sec.

Where does the 65 value come from? It is an empirically determined value which seems to be about right to avoid overshooting the software limits.

As in the case with the first break point above, it is necessary to translate this absolute value in BX to an equivalent position. At DRVINA [R569] we begin this process which is identical to that performed at DRVIN6; that is test LDIR, if the drive must go down, if so negate BX, if not add BX to the current position at DRVINAB. The resultant position break point is stored in BREAK2 in RAM.

This is the end of this portion of DRVINT peculiar to Rotation.

The portion of Focus-peculiar code follows; the end will be noted.

At [F546] BX is loaded with the first breakpoint; the JNZ DRVIN5 instruction tests to see if it is zero (it could be), if so control falls through the JNZ to set BX to a value of 1. If the result of the division and multiplication is vanishingly small, the ADD BX,AX operation above will set the ZF. This small result can only be the result of a position command to only a 13 or a few more counts different than the present position. CHKDRV will reject commands with motions less than 13 counts. If the ZF is not set, control is transferred to JNZ DRVIN5. If BX is zero, 1 is added to it for the BX comparisons below; BX must not have a zero value in these comparisons.

Remember that this code is also used by the LOCAL subroutine to initialize a motion command. The command mode is tested next by DRVIN5: TEST SYSTEM,2 [F549] which and's 2 with the manual mode bit (bit 1). The ZF will be set if in computer mode; control is transferred to DRVIN6. If in Manual mode (CMP-MAN switch set to MAN), the 7% break point in BX is compared with 60 by the CMP BX,60 instruction. This instruction subtracts 60 from BX; if BX is > 60, the CF is not set and BX retains its value from above. If BX is 60 or less, a value of 60 is loaded into BX to replace the 12% value calculated above. This latter case is associated with small motions and establishes a first ramp break point of 60 counts, an arbitrary value. In the manual mode we don't want to run as fast as in computer mode; the smaller break point 1 results from this concern.

We will need to refer to the old position in the forthcoming code so POSD (present position) is loaded into POSDED (position old) and SCRLST (screw last). SCRLST is the transition position for the second screw sensor and the initialization here, although not a transition gives the Second Screw Code (discussed in TMR2) an initial position reference. In addition, SCRCNT, FILOVR and FIL are cleared. Since these functions are peculiar to the Focus TMR2 code we will not describe them here.

Remember BX is still an absolute value; we must translate this value into an equivalent point in drive position. The direction to drive (LDIR) is tested [F560] to see if the drive is to be down; if so the BX value must be subtracted from the present position. If LDIR = 1, (i.e. drive down), BX is negated which forms a 2's complement negative value which is added to AX which contains POSD, the current drive position. The result of this subtraction is break point 1 which is 7% (or less) lower that the current position. If LDIR=0, the BX value is added to AX resulting in a break point 1 which is 7% higher than the current position. This break point 1 value is saved in BREAK1 in the RAM.

The second, 90% breakpoint is calculated next. The STACK is POP-ped onto AX. What did we pop onto AX? Remember the second PUSH at line [F537] above? That PUSH put the quotient of the first division of the absolute value of position error by 100 onto the stack where it has been waiting to be retrieved for the second set of arithmetic. Refresh your memory by going back to the previous page to review the first division of position error by 100. CX is loaded with the value 90 for the 90% breakpoint. Using this previous result we multiply the quotient by CX and the result is saved in BX.

Another POP loads AX with the remainder of the first division which is again multiplied by CX. CX is next loaded with 100 and AX is divided by CX. The result is added to BX, just as was done in the first set of calculations. Now what do we have in BX? BX contains an absolute value which is 90% of the command error.

Remember that this code is also used by the LOCAL subroutine to initialize a motion command. The absolute value of position error is POP-ped onto AX; it was PUSH-ed onto the STACK in line [F533]. The command mode is tested next by TEST SYSTEM,2 [F575] which and's 2 with the manual mode bit (bit 1). The ZF will be set if in computer mode; control is transferred to DRVIN8. If in Manual mode (CMP-MAN switch set to MAN), the 7% break point in BX is compared with 65 by the CMP BX, 65 instruction. This instruction subtracts 65 from BX; if BX is > 65, the CF is not set and BX retains its value from above. If BX is 65 or less, a value of 65 is loaded into BX to replace the 7% value calculated above. This latter case is associated with small motions and establishes a first ramp break point of 60 counts, an arbitrary value. In the manual mode we don't want to run as fast as in computer mode; the smaller break point 1 results from this concern.

Now, what is the reason for these subtractions and comparisons? At [F577] in the manual mode path we subtracted the break point from the error and then compared the difference to 65 counts. If the difference is greater than 65 counts, control is transferred forward; if less, the error (in AX) is substituted

for the calculated break point in BX and 65 counts are subtracted from this new break point. The effect of these calculations and comparisons is to cause the break point to always be at least 65 counts less than the error. The reason for this 65 count reduction in the manual mode is to make an earlier slowdown to VERGIT so as to not overshoot the command set point. Remember that in the manual mode, the command set point is either the HIGH or LOW software limit. Even though the operator probably plans to release the manual slew switch far short of this point, the program has either of these values as the target set point.

What is the Focus manual velocity? Using the 65 counts and the acceleration ramp of 1 volt/sec^2, which is 400 counts/sec^2, we calculate the (manual mode) Focus velocity in MAINNY to be 938 counts/sec and the DAC drive to be 0.687 volts. In terms of motor shaft rotation this is 2.076 rev/sec.

Where does the 65 value come from? It is an empirically determined value which seems to be about right to avoid overshooting the software limits.

As in the case with the first break point above, it is necessary to translate this absolute value in BX to an equivalent position. At DRVIN8 [F582] we begin this process which is identical to that performed at DRVIN6; that is test LDIR, if the drive must go down, if so negate BX, if not add BX to the current position at DRVIN9. The resultant position break point is stored in BREAK2 in RAM.

This ends this Focus-peculiar portion of DRVINT.

DRVINT now begins the process of turning on the Focus drive. VEL1 is initialized to zero. VEL1 is the location in RAM in which the measured velocity is stored. VEL1 is updated and used by the TMR2 subroutine. The brake is turned on by CALL BRKON. Upon return from BRKON, the state of EPROM1 Ports A and B are read into AX. The top four bits of the Port B state in AX are selected by AND AX,0F000H, all other bits in AX are zero-ed. If any of the selected bits in AX were a 1, they are retained intact. These bits are the 3-Phase command, the brake command, the servo amplifier reset and the servo amplifier inhibit bits. The four lower bits in Port B and all the Port A bits drive the DAC storage register. The program will presently load these bits to start the servo drive. BX is loaded with the value CONRAM which is equal to 4. This is equivalent to a servo amplifier velocity drive of 10 millivolts which is the smallest value to which the servo amplifiers will respond. The direction to drive is tested next, if LDIR = 1, the drive must go down so the servo drive polarity must be reversed; NEG BX does this. The DAC which drives the servo amplifier has a 12-bit drive register. DRVINC: AND BX, 0FFFH truncates the top four bits of BX. OR AX,BX merges into AX the 12 DAC drive bits in BX. The four top bits in AX are the 3-Phase command, brake, servo reset and servo inhibit bits which were retained intact in AX above. The new command state for the DAC and 3-Phase bits is next loaded into EPROM1, Ports A and B. The next two instructions load DX with the D/A latch address and output the 12 bit state standing in EPROM1 into the 12-bit D/A latch.

OR AH,10H merges into AH the servo amplifier enable bit, PB4 (when the amplifier is inhibited by the controller it cannot drive the motor). Remember that AX still contains the state of EPROM1 Ports A and B. The new EPROM1 Ports A and B states (with the enable bit true) is next loaded into the EPROM1 ports. The parameter SPEED is initialized to a value of 30; the TMR2 subroutine tests, sets, resets and decrements SPEED in the course of ramping the drives. The use of the SPEED parameter will be described in the TMR2 subroutine description.

RAMPOS is initialized next with the value 0. RAMPOS is an index parameter which defines which portion of the velocity profile the TMR2 subroutine is operating in. The velocity profile shown in the figure three pages back shows the four velocity regions and the associated TMR2 code section labels.

These are: ZIPUP, MAINNY, ZIPDWN and VERGIT and these regions are assigned RAMPOS values of 0, 2, 4 and 6 respectively. Early in TMR2, RAMPOS is read and used as an index to these velocity control code sections. The drive will activated a few lines below this point, TMR2 will become active, read the RAMPOS value of 0 which will direct control to the ZIPUP (ramp up) code.

Timer 2 is enabled next by loading DX with the address of the IPI T2 mode control register. AX is loaded with the T2 control value of 0E001H which is then output to start T2 into operation. What are these bits and what do they do? The Timer description in Section 3.1 explains the control functions of these bits in the T2 mode control register but we will briefly repeat them here. The EN bit is a 1; this permits the timer to count. The INHbar bit is a 1; this permits selective updating of the EN bit. The INT bit is a 1, this enables Timer 2 to generate an interrupt. The CONT bit is a 1; this causes Timer 2 to run continuously. All other mode bits are 0; the associated functions are not used in this application.

The DRVATV flag bit is set to the 1 state; it will be tested by BOSS and CHKDRV during the recurrent scans through these code segments.

The state of EPROM2 Ports A and B are read into AX next and masked with the 0FF97H pattern by the AND AX,0FF97H instruction. This clears the LED drive to the S105 Drive Up, Drive Down and Command LED's. The DRVINT subroutine will next set up the EPROM2 ports to drive the S105 display LED's to reflect the control states established during this command or local control initialization. The manual mode flag bit in SYSTEM is TEST-ed next, if in manual, control is transferred to DRVIND which avoids setting the CMD bit in EPROM2, Port A3. If not in manual this port bit will be merged into AX. The direction LED states are set up next by testing the state of LDIR; if LDIR is a 1, the drive direction is CW (increasing position values) and the associated CW bit is merged into AX. If LDIR is a 0, the CCW bit is merged into AX. At DRVIND [F619] the new EPROM2 Port A and B states in AX are output to EPROM2.

The last operation in DRVINT is to set a count of 25 into EXTTMR. This software backup to Timer 2 is decremented and tested for a zero count in CHKDRV and reset to a count of 25 early in the TMR2 subroutine. Control is returned to the CALL-ing location [F423] in BOSS by the RET instruction.

Finally, in finishing the DRVINT subroutine description, we repeat that this subroutine is used in both the Computer control and Manual control modes; the applications are very similar and dissimilarities so small that a few conditional jumps easily particularize the subroutine to the peculiar requirements of the two modes.

Table of velocity and breakpoint parameters calculated in DRVINT

Drive acceleration/deceleration: 400 DAC COUNTS/SEC$_2$; 1 VOLT/SEC$_2$
(Both axes, both Computer and Manual modes)

DRIVE REGIME	DAC COUNTS	DAC VOLTS	FOCUS VEL (COUNTS/SEC)	ROTATION VEL (COUNTS/SEC)
Computer MAINNY (max)	1120	2.8	3821	74,776
Manual MAINNY (max)	275	0.687	938.4	18,360
VERGIT	28	0.070	95.54	1869

Focus Breakpoints:	Computer	BREAK1 = 7% of commanded motion	BREAK2 = 90% of commanded motion
	Manual	BREAK1 = 65 counts from start pos	BREAK2 = commanded motion - 65 counts
Rotation Breakpoints:	Computer	BREAK1 = 12% of commanded motion	BREAK2 = 40% if ERROR < 1000 counts
			" = 60% if ERROR < 3000 counts
			" = 80% if ERROR >= 3000 counts
	Manual	BREAK1 = 65 counts from start pos	BREAK2 = HIGH or LOW - 65 counts

LOCAL (Manual slew control) Subroutine (FUM 12, RUM 11)

The LOCAL subroutine (RUM 11 [R610]) is a very important manual control program which permits manual control of the Focus or Rotation drive(s) for test purposes, such as positioning the drive at some specific physical point or aligning circuitry in S101. LOCAL command usage is intended for occasional test purposes; the normal mode of the F-R Control System is Computer Control in which the controller executes commands from the Antenna Control Computer.

There are two forms of manual commands: Position commands and Mode Switch commands. Position commands are input by actuating one of the momentary-action S105 manual slew switches; this causes the drive(s) to move in the direction indicated (e.g. UP or CW vs. DOWN or CCW) when the S101 MAN-CMP switch is in the MAN position. This form of LOCAL command input is primarily intended for a quick check of the system and for mechanical alignment purposes and is easily used by telescope operators or antenna mechanics. The direction of manually commanded motion is determined by sensing the slew switch state; when a switch is released, motion stops. The human in control of the slew switches is able to read drive position on the S102 numeric display. An example of this usage is the mechanical alignment of the subreflector and FRM on the Apex ring during installation of the FRM and F-R system.

The second form of manual control input is to set the mode switches to some state different than 00H. (See the mode switch description in Section 3.1). This second form of manual command input is intended for technicians who understand the F-R Control System and want to test or adjust the S101 A/D or D/A circuitry or to release the drive brakes. This form of program intervention is not for the uninitiated. LOCAL senses the states of the mode switches on the S101 front panel when the S101 front panel MAN-CMP switch is in the MAN position. When adjusting the S101 D/A and A/D circuitry, the LOCAL program will display the associated values on the S101 numeric display. The mode switch commands can only be executed when the drive is not active; since these mode switch commands force test states of the analog signals used by the position control programs, the program logic has been structured to eliminate this conflict.

Although the mode switch commands and slew switch commands are not mutually exclusive in the digital logic, they are effectively made so by the program logic. LOCAL first tests the mode switch settings after DRVATV has been tested and found inactive. After returning from the execution of a mode switch command, the slew switch states are tested to execute a motion commanded by a slew switch. Slew switch commands are not conditional upon a mode switch setting of 00 although this is the normal setting. There is not a conflict here because as soon as the drive is set into motion, DRVATV is set true which causes mode switch commands to be bypassed. When the drive becomes inactive again, the mode switch commands are again tested.

LOCAL is entered from the BOSS program when it detects that the CMP-MAN switch is in the MAN position; after executing the operations in LOCAL, control reverts to BOSS for another sequential scan.

Now a few comments about the structure and flow of LOCAL. LOCAL is entered from BOSS and there is only one return to BOSS, early in the subroutine. LOCAL can be over-ridden by the manual command over-ride command from the Antenna Control Computer; this is described in the DMA0 subroutine description above.

Like computer-directed commands, LOCAL calls DRVINT to initialize conditions for a drive when one of the manual slew switches on the S105 front panel is actuated. The three possible actions in response to the S105 slew switch actuations are: 1) drive UP (or CW) while the UP switch is depressed; 2) drive DOWN (or CCW) while the DOWN switch is depressed, and 3) shut down the drive when either switch is released.

LOCAL Subroutine (continued)

Manually-generated position commands in the LOCAL mode are functionally similar to computer-generated commands in that both use a target set point, both ramp up and down and converge to a set point. In the case of computer commands the command set point is output by the computer; in the case of manual commands the set point is always either one of the soft limits. The limit selection is indicated by the sense of the slew switch actuated. If the UP (or CW) switch is actuated (and held), the command set point is F200H (61952 decimal); if the DOWN (or CCW) switch is actuated (and held), the command set point is A00H (2560 decimal). With this approach, it is not possible to overrun the soft limits; when the drive reaches the second break point it will be ramped down and converged to the command set point. If the manual slew switch is released before the command set point is reached, the drive will slow to a stop. It may take several "jogging" actuations of the slew switch to reach a particular set point but this is inherent in the use of simple slew switches as opposed to a specific numeric manual command input. A numeric manual command capability could have been implemented but would probably have required some type of bulky keyboard input device and associated interface. Slew switches are the simplest and easiest interface for this occasional-use function.

A second similarity of the LOCAL manual mode is that motion is controlled by the TMR2 interrupt subroutine which regulates the motion through the drive regime on a timed-motion basis. Motion analysis protects the expensive and delicate drive mechanism by detecting faults in the event of fault conditions.

LOCAL also senses the states of the S101 mode switches and branches to perform the actions indicated by the switch settings. The branches to align the A/D converters and D/A converters load the converted values into the S101 front panel displays.

Like a computer-directed command, LOCAL operates in conjunction with BOSS and the TMR2 subroutine which does the timed management of drive position. Motion analysis (called by the TMR2 subroutine) is also performed in LOCAL so that in the event of mechanical binding in the FRM, motor drive power is shut down to protect the FRM. In the CMP mode, a computer-directed command loads a few command arguments for the program to execute and never changes the command conditions (except in the case of over-riding commands) during the command execution. In contrast, LOCAL must constantly test the manual command input states and dynamically adjust the control actions to meet the requirements of a human operator who may change the control switches in any manner.

Local branching is largely index driven; an index value from the mode switch causes jumps to a number of subroutines.

The first action in LOCAL is to merge the LOCAL flag bit into the SYSTEM status word by the OR SYSTEM, 2 instruction. When set, this bit indicates operation in the manual mode; when reset this bit indicates operation in the CMP mode.

The second action is to test the state of MANOVR, the manual command over-ride flag, by the TEST MANOVR,1 instruction. If not set, control is transferred to LOCLB. If set, (i.e. the manual over-ride mode is set) TEST DRVATV,1 tests to see if the drive is currently running. If the drive is not running, it is safe to shut down the drive; the JNZ to LOCLA permits a 1 to be set in the DRVOFF flag and control is returned to the start of LOCAL. The logical order of these tests is very important; MANOVR should be tested at the start of LOCAL because if in the manual command over-ride mode, no manual control actions should happen; the computer commands have precedence. Secondly, if the manual command over-ride command mode was just set, the DRVOFF flag should not be set if the drive is still moving in response to a manual command; the operator could have just released a slew switch and the drive could be ramping down to a stop. This is the reason that DRVATV is tested for the 1 (or running)

state. If a computer command were permitted to instantly initiate a position command after setting the manual over-ride mode (for example in the opposite direction to the current motion), the FRM and servo amplifiers could be damaged. The drives have large inertial loads; the stress on gears, motors and amplifiers could be severe. The return to the start of LOCAL if the DRVATV indicates that the drive is still moving permits the over-riding command to recurrently test DRVATV. When DRVATV is no longer a 1, control reverts to BOSS which then starts the command initiation via a CALL to DRVINT. Another important point is that the operator who may be using the LOCAL manual mode has no indication that a manual over-ride command has inhibited his manual position control capabilities.

LOCAL next starts the process of reading the manual control inputs to determine the action to be taken. At LOCLB [R619] the STACK is reinitialized to 2100H since this is the start of a main control program; this operation assumes that the previous contents of the STACK are no longer relevant. The Apex Interface (S102) fault bit in FAUL1 is tested next; if not set, the ZF is 0 and control passes to LC1. If S102 is faulty (DSTOR identified it and set the bit in FAUL1), the JMP RSCMD returns control to the RSCMD [R423] point in BOSS which initiates a soft reset command.

At LC1 the TEST DRATV,1 tests to see if the drive is running (DRVATV has two other flags as we shall presently see in LUP and LDWN). If it's running, it's in response to a manual slew switch command set during a previous pass through LOCAL; this being the case we don't want to initiate any mode switch commands in LOCAL until the drive stops. JNZ LC1B transfers control to the EXTTMR test below.

If the drive is not active, DSTOR is called to obtain fresh position and discretes data. The Mode switch address is loaded into DX and the switch state is input into AL and then inverted since it is low-true. In the case of the Focus code the Mode switch state is tested for a setting of FFH. If it is it means that the operator has commanded that the second screw analysis of the TMR2 subroutine should either be bypassed or made active again. The CMP AL, 0FFH will set the ZF if AL = FFH. The SCWIGN flag is first set to zero following the comparison, then if set, control falls through the JNZ LOCLC instruction to set the SCWIGN flag to 1. In this code the bypass mode is either set or cleared. The switch should never be left in the FF position after a test, the Focus drive is protected from damage by the second screw code.

At LB1B, if the drive is active, DRVATV will be 1; the drive is moving so it is appropriate to call TMR2 to control motion. At LC1B, the software timer EXTTMR is decremented; if it is not zero, control is passed (via the JNZ LC1C instruction) down to LC1C which tests the slew switch states. If the counter decrements to zero, it means that a TMR2 interrupt has been missed so the interrupt is invoked by the INT 3 instruction which calls the TMR2 interrupt subroutine.

If the drive is not running, the mode switch is read to determine control action. The mode switch read process is similar to the many input operations discussed above but in this case only AL is loaded; the mode switch is only a byte device. Since the switch inputs a low-true signal, AL is inverted to obtain the proper bit sense. AH is cleared (it may be in an indeterminate state from some previous operation). BFLAG is also cleared since the next instruction is a BOUND instruction. The BOUND AX, CS:MANRG tests the state of AX against the MANRG table (RUM 18 [R960]). The familiar CS: code segment override causes the CS register to be the (i.e. EPROM) memory reference rather than the DS register. Looking at the MANRG table we see that switch values between 0 and 7 are accepted; values outside this range induce a BOUND interrupt. (We described this subroutine above). Note the two data values in the third column of the listing: 0000 and 0700. Why the inverted order? Remember that the order of the two bytes are in fact inverted. Also note that both the Focus and Rotation program respond to the mode switches in an identical manner, They concurrently execute LOCAL in tandem although they are probably

not executing identical instructions at any given time; there are small differences between the two programs. At this point, the reader should refresh his memory by turning to the Mode Switch Settings discussion in Section 3.1.

Having performed the BOUND test of the switch settings, the BFLAG is tested to see if it was set which indicates that an out-of-range setting was detected; only valid switch commands will be executed. The switch value will be converted into an index to transfer control to one of several subroutines which execute code associated with some function described below. If the mode switch value is out of range, control is passed to LC1A by falling through the JZ LC1A, AX is cleared for the operations in LC1A.

If the mode switch settings are in range, the BFLAG is zero, ZF = 1 and control is transferred to LC1A by the JZ LC1A. The AX (i.e. mode switch value) is doubled and set into BX to serve as an index. MOV AX,CS:MANTBL[BX] is an indexed MOV instruction which loads AX with the address designated by the address of MANTBL plus the contents of BX. Glancing at MANTBL (RUM 18 [R966]) we see a series of .WORD pseudo-instructions which are a mnemonic equivalent to the address of subroutine entry points to perform the A/D and D/A operations. Note for example, the address of NORMIT is 0458H. Having

loaded AX with this subroutine address, the next instruction is a CALL AX which transfers control to the address loaded by the table index above. This is a neat, compact way to call these subroutines which avoids the necessity of doing a sequence of: is the value X?, if so got Y1, if not, is the value X+1?, if so go to Y2, if not is the value X+2? etc. tests. Secondly, the symbolic approach avoids the clumsy tactics of having to update a numeric address when the code is changed. Upon return from the subroutine control is transferred to LC1C which starts the slew switch tests.

Now what about the case in which the mode switch value was out of range which set AX = 0? What happens with an index of 0? In this case NORMIT [698] is called; the code defaults to setting the D/A converter register to zero. We will describe the mode switch command subroutines below.

All returns from the mode switch-induced CALL AX subroutines return to the JMP SHORT LC1C instruction [R639] which begins the slew switch tests. IN P2PTA reads the state of EPROM2 into AX (the fact that some of the bits are in the output mode is irrelevant here). The quiescent (i.e. non-actuated) state of these two switches is high-true. The Port B bits will be input to AH. Two bits are of interest: Port B bit 7 (UP or CW) and bit 6 (DOWN or CCW). If either or both these bits are true, it means the operator wants to move in the indicated direction. TEST AH,128 and's bit 7 with the 128 weight (or bit 7) bit of the test value. If the ZF is not set it means that the UP (or CW) bit is true; hence the switch is not actuated. If ZF = 1, the switch was actuated so we jump to LUP to cause the drive to move UP (or CW).

In the event that a mischievous operator actuates both the UP and DOWN switches to see what happens, the UP (or CW) test will direct an UP control response because it is tested first and bypasses the DOWN switch test.

If the ZF = 0 above, we next test bit 6 against the 64 weight bit in a similar test and if the switch is actuated, control is transferred to LDWN which moves the drive DOWN (or CCW). If the DOWN (or CCW) switch is also not true (low), both switches are not actuated so the immediate conclusion is that either the operator just released a switch or that they had been released for some time. If the case is the just released case, we could still be running and ramping down; in this case the "1" bit in DRVATV will still be true. At LC3, DRVATV is tested against the 1 bit and if true (the drive is moving), control falls through JZ LC4 to set a 1 in DRVOFf; this will initiate the process of turning off the drive in a graceful manner by TMR2. Control is returned to the start of LOCAL via JMP LOCAL.

LOCAL Subroutine (continued)

Control got to LC4 because the drive is inactive; things are quiescent so it is time to test the CMP-MAN switch to see if the operator changed control back to the computer mode. AH still has the Port B bits read from EPROM2. The MAN-CMP switch bit, 6 (weight 32) is tested to ascertain the switch position. In the MAN position it is low-true (i.e. a) and in the CMP position it is a 1. In response to the TEST AH,32 instruction, if ZF = 0 we are still in the manual mode but we are not moving so it is appropriate to clear some flags. Control is transferred to LC5 to do so. If ZF = 1, the operator just switched the mode back to CMP so it is appropriate to JMP to RSCMD (in the DMA0 subroutine) to reset the DMA 0 IPI registers so as to be able to respond to computer commands. Control returns to BOSS through the normal DMA0 exit.

At LC5 [R657] the DRVATV, RAMPOS, DIR and LDIR flags are cleared by a code sequence identical to that used in the CHKDRV subroutine at [R472]. Since this code sequence was described CHKDRV it is not necessary to repeat it here. After clearing the flags, control is returned to the start of LOCAL by the unconditional jump. Under the conditions set up immediately above (i.e. DRVATV = 0), control will quickly revert to BOSS as a result of the test of DRVATV.

We next consider the LOCAL LUP: [R663] (LOCAL drive UP or CW) and LDWN: [R680] (LOCAL drive DOWN or CCW) subroutines which are accessed as a result of tests of the states of the S105 manual slew switches. If one of the switches is actuated, the program transfers control to one of these subroutines. In these two subroutines the code performs several logical tests: is the drive moving, if not, should it be shut down or started? Is it moving in the wrong direction relative to the switch command? If so initiate a drive shutdown. If the switch was just actuated but the drive is not moving then we have a new manual slew command which needs a command set point and command initialization. Clearly, these short code sections are very important control logic.

In LUP: [R663] DRVATV is tested to see if the drive is running (using the 1 or running bit). If not, apparently the UP (CW) switch was just actuated and this is the first pass to detect it. In this case, control is passed to LUP1 to get motion started. If the drive is active, the TEST DRVATV,2 instruction tests to see if it is flagged as moving UP (or CW). Why do this? We are in the UP path and just got here from sensing the UP switch. The answer is that the DRVATV UP flag, bit 2 would have been set in a previous pass through LUP and considerable time (possibly several hundred usec) could have elapsed between passes through LUP; the operator could have actuated the DOWN switch in the interval. This test is a consistency test to trap a transition to the opposite drive sense. If the current motion direction is not consistent with the switch command, the DRVOFF flag is set which will flag the TMR2 subroutine to slow down and stop the drive. Having set this flag, control is returned to the start of LOCAL by JMP LOCAL.

At LUP1 [R670] we first want to see if it is safe to turn on the drive in the requested direction. We could be close to one of the upper or lower software limits; if so, we don't want to permit the operator to drive past the software limit. The MOV WORD PTR POSCEC,HIGH instruction loads the LOCAL software command set-point into POSCEC (position command echo) for storage just as in computer mode. This is the value which TMR2 will drive towards while the UP (or CW) is depressed. This software limit-set point value is 61,950 counts, about 5% away from the maximum possible position, but is still 1417 counts below the 1st upper limit switch setting. If the UP slew switch is continuously depressed, the drive will converge to this set-point value; it will not permit drive into the UP limit switches. DSTOR is called next to obtain the present position which puts the present position into POSD. POSD is loaded into AX. POSD (in AX) is compared with this HIGH software limit by the CMP AX,HIGH instruction which subtracts HIGH from POSD. The CF (Carry Flag) is set if HIGH is greater than POSD. If CF is set, it is safe to move a little farther UP so the JC LUP3 permits this motion to start. If POSD is at this limit, CF will not be set; the motion request is disapproved and JMP LOCAL returns control to the start of LOCAL.

LOCAL Subroutine (continued)

At LUP3 the instruction CALL LOCTST calls the DRVINT (drive initiate, described above) LOCAL subroutine at the LOCTST entry point [R485] to set up the break points. Upon return by the RET-urn from DRVTST, bit 2, the DRIVING UP (or CW) bit is merged into DRVATV. Control is then returned to the start of LOCAL by the JMP LOCAL instruction.

LDWN [R680] is the complementary counterpart of the LUP subroutine; it performs the same functions but is used for DOWN (or CCW) motions. The instruction sequence and logic is identical; the only differences are that the LOW software limit has a value of 2560 which is 750 counts above the 1st lower limit switch setting. The second difference is that after the CMP AX,LOW comparison, the JNC jump is used to pass control to the LDWN3 point which calls LOCTST to start the command initiation. A third difference is that bit 4, the DRIVING DOWN bit is first tested and then merged into DRVATV.

In LUP and LDWN we see an interesting instruction: in LUP at [R670] we have MOV WORD PTR POSEC, HIGH. Obviously this is a MOV to load the HIGH value (equal to 0f200H) into POSCEC. How does it work? The op code is C7 06 00 20 00 0A. C7 is the MOV immediate operand to memory code with w = 1, a word operation. 06 is the mod-reg-r/m code which means a mod of 00 which normally means that displacement is missing. In this case the 110 r/m code specifies that if mod = 00 and r/m = 110, then the effective address is disp-high:disp-low. The displacement is 2000H which is the offset address of POSCEC from the DS register value of 0000. Finally, the data is 0A00H, the value of HIGH from the program equates.

Finally, we consider the subroutines that were CALL-ed by the index value in AX which was derived from the mode switch setting. This indexed the call to an address in the MANTBL [R966]. These subroutines are used to adjust the S101 D/A and A/D converters.

The first subroutine is NORMIT (RUM 13, [R698]) which CALL's BRKOFF to turn off (i.e. re-engage the brake, we are not moving); then control is transferred to ZERDA [R706] which loads BX with 0. The jump to SETDA will load AX with the state of EPROM1, Ports A and B.

At SETDA the argument set into BX is output to the D/A command register. Port A state will be in AH and Port B state will be in AL. AND AX, 0F000H clears all bits in AX except for the top four Port A bits which are the 3-phase command bit, the brake command bit and the BDS3 reset and inhibit bits. The states of these four bits are left intact by the AND instruction. The OR AX,BX instruction or's the D/A argument in BX into AX. The new command state in AX is output to EPROM1 Ports A and B. The address of the D/A register strobe (LTCHDA) is loaded into DX and the familiar OUT DX,AX instruction clocks the 12 bits of EPROM1 Ports A and B into the command register. The D/A assumes the commanded analog level within a few tens of nanoseconds time. From SETDA, control is returned to the instruction following the indexed call to these subroutines. This instruction is JMP SHORT LC1C, [R639].

The fourth subroutine is ZERDA which repeats the operation of zero-ing the D/A command register as described immediately above. The D/A command register is loaded with the BX value as a result of the JMP SETDA instruction. The Brake state is unaffected. A DVM is used to measure the D/A output during the zero adjustment.

The third subroutine is NEGDA [R702] which loads BX with 0800H which is a 2's complement minus full-scale argument. The D/A command register is loaded with the BX value as a result of the JMP SETDA. A DVM is used to measure the D/A output during the gain adjustments.

The fifth subroutine is POSDA [R710] which loads BX with 07FFH which is a 2's complement, positive full-scale argument. The D/A command register is loaded with the BX value as a result of the

LOCAL subroutine (continued)

JMP SETDA instruction. A DVM is used to measure the D/A output during the gain adjustment.

The sixth subroutine is NEGAD [R721] which sets the analog multiplexer address to select the -10 volt reference source for conversion, causes an A/D conversion and loads the display with the converted value. In NEGAD the EPROM2 Port A and B bits are loaded into AX. The AND AX, 0FFF7H clears the three lower bits in AX and retains all other bits that are 1's intact. The three lower bits are the analog multiplexer address bits. OR AX,3 merges the mux address for the - 10 volt source into AX. Control is then transferred to READAD [R735].

READAD clears the ADFLAG and outputs the state of AX to EPROM2 Ports A and B. The three multiplexer bits cause the multiplexer to select the - 10 volt source for the impending A/D conversion. The STCNV enable address is loaded into DX and the OUT DX,AL triggers the conversion. The Interrupts are re-enabled by the STI instruction to insure that the INT0 interrupt will be sensed by the processor. The A/D conversion takes about 25 usec and the end of conversion signal activates the INT0 subroutine (described above) which stores the 2's complement value in ADVAL in RAM and sets the ADFLAG to 1 to

indicate that the conversion has been done. After executing the STI instruction, the program loops on the READA1: TEST ADFLAG,1 and JZ READA1 loop which tests the ADFLAG for the 1 state set by the INT0 subroutine. When the ADFLAG becomes a 1, the A/D data stored in ADVAL is loaded into AX and then stored in VEL1. Control is returned to the LOCAL instruction following the indexed CALL AX; which is JMP SHORT LC1C [R639].

The seventh subroutine is ZERAD [R726], which performs a mask-merge of the EPROM1 port states in AX similar to that done in NEGAD. In this subroutine the multiplexer address is 4 which selects analog ground as the analog signal to convert. Control is transferred to READAD which performs the conversion, storage and RET-urn.

The eighth (and last) subroutine is POSAD [R731], which again performs a mask-merge of the EPROM1 port states in AX similar to that done in NEGAD using a multiplexer address of 2 which selects + 10 volts as the analog signal to convert. Control is transferred to READAD which performs the conversion, storage and RET-urn.

TMR2 Interrupt Subroutine (FLM 9, RLM 9)

The TMR2 subroutine may be considered to be the heart of the control program; it manages drive motion. All the program components discussed to this point (with the exception of the NMI subroutine) are supportive elements for TMR2. TMR2 is described last to familiarize the reader with TMR2's interactions with the other program components.

In both computer and manual modes, TMR2 manages drive motion in accordance with the rules of the drive profile introduced in the DRVINT description above. Each portion of the drive profile (i.e. ZIPUP, MAINY, ZIPDWN and VERGIT) is controlled by a section of code with these labels. The index RAMPOS, determines the selection of the code section when TMR2 is entered. Time and drive position are the driving parameters of TMR2; Timer 2 interrupts or the backup software timer EXTTMR invoke the TMR2 subroutine which executes the drive rules and tests drive position for transitions to the next portion of the drive profile. TMR2 interrupts (or the interrupt induced by EXTTMR) occur at fixed time intervals; there is not a time base, which directs that at XXX seconds do YYY, etc.

Normal exits from the four sections of TMR2 are made through TMR2EX which pops the registers and resets the Interrupt controller. TMR2EX will be described after the code for the four drive regimes

has been described. Fault state exits from TMR2 are made via OFFIT and OFFIT3 which will be described later in the TMR2 discussion.

A MOTION analysis subroutine is periodically executed by TMR2 to compare realized drive motion with commanded motion; the object being to quickly detect dragging or sticking conditions in the drive so that physical damage to the expensive and delicate FRM can be averted.

The TMR2 interrupts occur at about an 1800 Hz rate; the closed-loop bandwidth of the Focus servo system is about 70 Hz and the Rotation closed-loop bandwidth is about 20 Hz; thus the sampling rate for Focus is about 26 times the bandwidth and the Rotation sample rate is about 91 times the TMR2 bandwidth. The TMR2 control equations are essentially difference equations which cause motion increments on a time-periodic basis; since the sample rate is so high relative to the servo bandwidth; the incremental motion approach maintains tight control over drive position.

The Focus version of TMR2 has code to analyze the motion of the second screw relative to the first screw which drives the Focus position readout. This second screw analysis code starts at TMR21A, early in TMR2 following the limit switch tests. Other than this second-screw code, the TMR2 codes are very similar, so the following TMR2 description is equally applicable to both programs. The Focus second screw analysis code is described at the end of the TMR2 description.

Upon entry to TMR2 (RLM 9, [R446]) via the Type 19 Timer 2 or the Type 3 interrupt induced by EXTTMR in CHKDRV (see CHKDRV and the interrupt vector tables on RLM 5 & FLM 5), the AX, BX, CX and DX registers and the BFLAG are pushed onto the STACK. The IP, Flag and the CS registers were pushed onto the STACK by the CPU when it executed the interrupt. Although the BOUND interrupt has a higher priority than the Timer 2 interrupt, the safest course is to save the BFLAG in the event of contention between the two interrupts; i.e. the execution of the BOUND instruction concurrent with the occurrence of the Timer 2 interrupt.

After the pushes above, EXTTMR is reset to 25 counts; after the initial set-up to 25 counts at the end of DRVINT, this is the only place where EXTTMR is reset.

The next instruction is an important test of the state of the DRVOFF flag to see if the drive should be shut down; if it has been set, program control is transferred to OFFIT which will be described later. The DRVOFF flag can be set by CHKDRV ([R456], RUM 9), LOCAL [R652], RUM 12) and conditions in TMR2 as we shall see presently. The reader should briefly review these sections of code to see why this flag was set.

Another very important test is to sample the state of the EMERGENCY STOP and DRIVE LOCKOUT sense lines on EPROM2. Why test these signals now? The best possible synchronized sample point of the program is the start of TMR2; if one of these alarm/fault conditions should go true, it is best to take action before additional motion is commanded. Since the TMR2 sample rate is 1800 Hz, the response to these signals is very fast. Let's consider what this response time is in terms of the motor shaft speeds. At the maximum commanded Rotation motor shaft speed of 36.5 rev/sec, which produces a position rate of 74,776 counts/sec, position changes by only 41.5 counts between TMR2 samples. This corresponds to only 438.1 arc-minutes of Rotation motor shaft position. In the case of the Focus motor, the maximum motor shaft speed is 8.456 rev/sec and the position count rate is 3821.9 counts/sec. Position changes by only 2.12 counts between passes through TMR2. This corresponds to 101.5 arc-minutes of Focus shaft position.

Relative to the EMERGENCY STOP and DRIVE LOCKOUT, it should be noted that the drive has probably been stopped before this point but for protection of the drives, TMR2 should perform this test.

The now-familiar read of EPROM2 Port B loads the states of these two lines (and the other lines which we are not interested in) into AL. The state of the two bits are retained intact by the AND 18H instruction; all other bits are cleared. Now, remember that the DRIVE LOCKOUT bit is low-true and the EMERGENCY stop is high-true. XOR AL, 10H [R458] forms the exclusive or of the contents of AL and the value 10H. The 1 bit in 10H corresponds to the low-true DRIVE LOCKOUT bit in AL; the high-true EMERGENCY STOP bit corresponds to a 0. (The other bits are zero but this is irrelevant since the six other 0's will be XOR-ed with six more 0's.) Thus the test pattern bits are the complement of the two fault/alarm bits in the true state. In the XOR instruction, if any of the 8 bits resulting from the XOR of the test value and operand is not zero, the ZF is cleared. Thus if either (or both) bit(s) is/are in the true (i.e. fault or alarm) state and is XOR-ed with the 10H pattern, it will cause either one or two 1's to be formed in AL and clear the ZF. The reader should check this logic by performing an XOR of this pattern against the fault-no fault data bits.

DSTOR is called next to obtain fresh data for the impending limit switch and position tests. The limit switch tests are first; only the switches which could be actuated by sustained drive in the current direction will be tested. Al is loaded with the discretes byte. During DRVINT, LDIR was set to the state which indicated the direction that the drive should move to null the position error; LDIR is not changed subsequently. In contrast, the DIR bit can change sense if the drive overshoots the command set point. If LDIR is a 1, the initial motion was down and the converse if a 0. The LDIR state is used to select either the upper or lower limit switches for test by the TEST LDIR,1 [R464] instruction which will set or clear the ZF depending upon the state of LDIR. If LDIR is a 0, the initial drive direction was up and the ZF is set. Thus the CW limit switches should be tested; this happens at TMR21A. The JZ TMR21A transfers control to test these switches. If the ZF is cleared, the CCW switches will be tested immediately below the ZF jump.

At this point we digress. The UP and CW limit switches are logically analogous in that they are both associated with upper position values; the converse is true for the DOWN and CCW limit switches. From this one would infer that this part of both the Focus and Rotation codes are identical. This is functionally the case but the code sequence is different because the switch selection bit patterns are reversed between the two sets of limit switch bit patterns. The sense of the limit switch bits are interchanged in the two S102 discretes readout registers; this was described in the description of the DRVIN2 and DRVIN3 paragraphs of DRVINT.

In the Rotation LDIR test, if control falls through the JZ TMR21A instruction, the drive is moving CCW so the CCW limit switches should be tested. The AND AL, 0CH instruction selects the CCW limit switch bits for test. (In the comparable test in the Focus code, 0CH selects the UP-per limit switches for test.) If either of the CCW limit switches is set (i.e. actuated), the results of the AND are a 1 bit in AL so the ZF will be cleared. The next instruction is an unconditional jump to OFFIT (RLM 12, [R635]) which turns off the drive. In this event, we want to shut down the drive even if the position values indicate that we are not near a limit switch actuation point. There may be a malfunction in the position readout circuitry so it's best to shut down the drive to protect the delicate and expensive FRM. If the CCW switches are ok, the JZ TMR22 transfers control to TMR22.

At TMR21A, [R469] the CW limit switches are tested using the AND AL, 03H instruction which selects the CW switch bits in AL. In the comparable test in the Focus code, 03 selects the DOWN switches for test. As in the CCW test above, if the switches are ok, control is transferred to TMR22; if the switches are bad, the jump to OFFIT turns off the drive.

TMR22 [R472] tests to see if the present position has overshot the set point. This is determined by comparing DIR (the direction to move to null the position error) with LDIR, the initial direction of motion. If DIR is different than LDIR, it indicates that the drive has passed or overshot the command set point. Since the TMR2 sampling rate is high (see the note above about the maximum possible motion between TMR2 samples), the assumption is made that the drive is very close to the set point and control should be transferred to the convergence code, VERGIT. To do so, the RAMPOS index is set to 6 and control is directly transferred to VERGIT by an unconditional jump.

An important point should be emphasized here, an DIR/LDIR comparison is also an indirect test for R/D converter missing codes and hardware failures; abrupt changes in DIR may be indicative of a mechanical or electronic problem.

If the drive is still moving in the original direction as determined by the DIR-LDIR comparison, above, the SPEED counter is tested. At the start of each pass through ZIPUP, MAINNY, ZIPDWN and OFFIT (VERGIT is the exception), a software counter named SPEED is decremented, if zero, it is reset to a value of 20 (decimal). In ZIPUP and ZIPDWN, D/A values are changed when SPEED is 0 after being decremented. At this TMR22A [R477] point, SPEED is compared with a value of 10 (decimal) which is intermediate between times that drive velocity might have changed (in MAINNY velocity will not change). If SPEED is 10, control is passed to MOTION to see if the drive is dragging or sticking. MOTION will be described later.

We might ask: why test drive motion only at intervals of 20 passes through TMR2? This is a sample rate of 90 Hz. In the above discussion of the maximum possible position changes between TMR2 cycles, we noted that at maximum drive speed (with 2.8 volts drive to the BDS3), Rotation position changed only 41.54 counts and Focus changed only 2.123 counts between the 1800 Hz TMR2 cycles. Testing motion at 20-cycle intervals improves the resolution of the test; comparing positional changes on values with these small changes is numerically awkward and error-prone.

At TMR23, the TMR2 control index is loaded and control is transferred to the TMR2 code section pointed to by the address in RAMPTL (RLM 15, [R834]). Since this control transfer is logically identical to that done in LOCAL to the set D/A, etc. code sections, it is not necessary to describe this control transfer in detail. The control transfers will direct program control to ZIPUP, MAINNY, ZIPDWN and VERGIT.

In ZIPUP (RLM 10, [R552]), as suggested by the label, the drive is ramped up in speed to the first position break point calculated in DRVINT. At the end of ZIPUP, drive position is tested to see if this position break point has been reached; if so the RAMPOS index is changed to the value for the MAINNY code section so that in the next pass through TMR2 the MAINNY code section is executed.

In ZIPUP, the SPEED counter controls the changes in drive velocity. Drive acceleration is constant; the velocity ramp rate (dV/dt) is 1 volt/second.

The first action in ZIPUP is to decrement SPEED to see if it is time to change the D/A value; the D/A output drives the servo amplifiers. If SPEED is not zero after decrementing, control is passed to ZIPUP2. If SPEED is zero after decrementing, it is reset to 20 and the state of the D/A value standing in EPROM1 Ports A and B is loaded into AX. This value is the current drive rate standing in the D/A drive registers (see Sheets 4 and 8 of the S101 logic drawings). The current value in AX is saved in the STACK by a PUSH AX.

TMR2 Interrupt Subroutine (continued)

The D/A drive is 12 bits. The upper 4 bits are used to control the 3-phase relay, brake, BDS3 inhibit and BDS3 reset; we do not want to alter the state of these 4 bits on EPROM1 Port B. These 4 bits are zeroed in AX by the AND AX, 0FFFH. A velocity increment of 4 counts is loaded into BX; since the D/A scaling is 2.5 mv/bit, this is a 10 mv change. Since the D/A output is bi-polar with a 2's complement drive, it is necessary to determine the drive direction to adjust the velocity increment sign before adding it to the current value. The sense of LDIR is tested by the TEST LDIR,1 instruction which will clear the ZF if the drive is CW (or UP). If CW (UP), control is passed to ZIPUP1 which adds the velocity increment in BX to AX. If the drive is CCW (DOWN), the NEG BX instruction forms the 2's complement of the BX value. The next instruction, ZIPUP1, adds this negative increment to AX.

The next instruction again masks off the upper 4 bits in AX while retaining the lower 12 bits. This second AND AX,0FFFH zeroes the upper 4 bits which could be 1's because the AX value could be negative. Even though we are dealing with 12-bit values, the 2's complement format for negative numbers extends the sign through the upper 4 bits. The new D/A value in AX is BOUND tested to insure that it does not exceed a maximum. ROL AX,4 left shifts AX so that the 12 bits are at the top of the register. The BOUND AX,CS:RAMPLM compares the AX value with the two limit values in the RAMPLM table on RLM15, [R830]. Remember that the value in AX is 2's complement so it is immediately compatible with the BOUND instruction. The two values in RAMPLM are 0BA00H (-17920 decimal) and 4600H (17920 decimal); when divided by 16 (remember the 4-bit left shift) and multiplied by the 2.5 mv D/A scaling, we get a maximum D/A drive of +/- 2.8 volts. If the amount of motion is small, this maximum may not be reached but if the distance is large, this velocity bound may be reached. The BFLAG is tested below in ZIPUP2. Having executed the BOUND test, the AX data is right shifted 4 bits to return it to its original value. The 4 most significant bits in AX before the ROL were loaded into the 4 least significant bits in AX. The ROL restores these bits to the top of AX.

The STACK is POP-ped onto BX; this data is the state of the EPROM1 Ports A and B at the time that ZIPUP was entered. We have a new D/A drive value standing in AX but need to merge in the upper discretes control bits states in the upper 4 bits of BX. The AND BX, 0F000H retains these 4 bits and zero's the lower 12 in BX. The OR AX,BX instruction or's these four bits into AX which is now the new command value for the EPROM1 ports. The OUT P1PTA, AX loads these 16 bits into the EPROM1 ports. Since the upper 4 bits were never changed during the calculation of a new D/A value, they are not perturbed by this new load into the EPROM1 ports. We need to get this new D/A value into the D/A register; this is done by the next two instructions. DX is loaded with the address of the D/A strobe enable. The OUT DX,AL instruction strobes the 12-bit EPROM1 Port A and B bits into the D/A storage register. The value in AL is lost; it is irrelevant. Within a few microseconds the D/A output assumes the new value. The D/A operational amplifier has a time constant on the order of 50 nanoseconds.

At ZIPUP3 [R582] the BFLAG is tested; if set it means that the maximum drive speed has been reached. If so, control is transferred to ZIPUP3 which causes the transition to the MAINNY drive regime.

The program now tests to see if the first break point has been reached. This is done by a comparison of current position of POSD (current position) and the break point. The CMP instruction is used and the state of the CF is indicative that it has been reached. There are two cases to consider for this comparison which is a subtraction operation. LDIR is tested to determine the logic of comparison and selects from either of two comparison paths. The comparison is in both cases: CMP AX, BREAK1 in which AX contains POSD. This is a subtraction which will either set or clear the CF as a function of the relative values of BREAK1 and POSD.

If moving CW (UP) and CMP AX, BREAK1 [R579] sets the CF, it means that BREAK1 > POSD and we have not yet reached the break point so control is transferred to TMR2EX which is the exit from

the TMR2 subroutine. If in this moving CW (UP) test the CF is not set, it means that BREAK1 < POSD and we have reached or slightly exceeded the break point so control is transferred to ZIPUP3 which causes the transition to the MAINNY drive regime.

If moving CCW (DOWN) and CMP AX, BREAK1 [R584] sets the CF, it means that the drive has reached or slightly exceeded (moving toward lower values) the break point so control is transferred to ZIPUP3 which causes the transition to the MAINNY drive regime. If the CF is not set, we have not reached the break point so control is transferred to TMR2EX which is the exit from the TMR2 subroutine.

At ZIPUP3 [R582], the transition to the MAINNY drive regime is caused by adding 2 to RAMPOS (it was 0 in ZIPUP) so that the next entry to TMR2 will index to the MAINNY drive regime code. After setting RAMPOS, control is transferred to TMR2EX, the exit from the TMR2 subroutine.

We will describe ZIPDWN next because of its similarity to ZIPUP. Like ZIPUP, ZIPDWN periodically alters the D/A drive as a function of the SPEED count but as suggested by the label, the drive velocity is reduced by decreasing the value in the D/A register. The velocity ramp rate is 1 volt/sec, just as in ZIPUP. ZIPDWN does not have a BOUND test of the D/A drive value since the drive is being ramped to a low speed; there is no need to test to see if a maximum value has been reached. Velocity is however tested to see if it has been sufficiently reduced to cause a transition to the convergence routine, VERGIT.

At the entry to ZIPDWN, [R606] SPEED is decremented and if not zero, control is transferred to ZIPDN3 which is the TMR2EX from ZIPDWN. If SPEED became zero when decremented, it is reset to 20 and the EPROM1 Ports A and B are loaded into AX and the contents of AX are pushed onto the STACK for temporary storage. As in ZIPUP, the upper 4 bits of AX are zeroed by an AND AX, 0FFFH. A velocity decrease value of 4 is loaded into BX. A TEST LDIR,1 determines the direction that the drive is moving; if the ZF is set, the drive is moving CW (or UP). The JZ transfers control to ZIPDN1 which subtracts BX from AX, the current D/A setting.

If LDIR is 1, the drive is moving CCW and control falls through the JZ ZIPDN1. BX is negated to a 2's complement minus 4. The SUB AX, BX subtracts the positive or negative 4 from AX. If AX is a positive value (i.e. the drive is moving CW), the result in AX is a smaller positive value. If AX is a negative value (i.e. the drive is moving CCW), the result in AX is a smaller negative value because BX (being negative) subtracted from AX reduces its negative value. The upper 4 bits in AX are masked off by the AND AX, 0FFFH. If AX is negative, its upper 4 bits are 1's since 2's complement extends the sign through the top bits even if we are dealing with a 12 bit value. The STACK is popped onto BX to retrieve the upper 4 discrete control bits.

In executing the AND AX, 0FFFH above, if the AX value is zero, the ZF flag will be set. AX could be zero because it had been decremented to that value in ZIPDN1. If zero, control is transferred to ZIPDN2. If AX is not zero, the 4 discrete control bits in BX are retained intact by the AND BX, 0F000H. The following instruction or's them into AX and they are then output to EPROM1 Ports A and B just as described above in ZIPUP.

At this point [R629] we want to test to see if the drive speed has been sufficiently reduced to make the transition to the convergence drive regime, VERGIT. This will be done by a CMP AX, CONRAM which subtracts a threshold value from the D/A drive value in AX. The logic of this operation is dependent upon the direction of motion. Again LDIR is tested for direction; if moving CW, the ZF will be set and the JZ AIPDN2 jump is executed.

If moving CCW, the ZF will be cleared. In this case the AX values are negative, so AX is negated to a positive value for the impending comparison instruction. ZIPDN2: NAND AX 0FFFH zero's these 4 upper bits; they could be 1's if we are moving in the CCW direction which involves 12-bit, sign-extended 2's complement values.

CONRAM is the velocity threshold for transition to the convergence code, VERGIT. This velocity threshold is 70 millivolts and corresponds to a Focus velocity of 0.2114 rev/sec and a Rotation velocity of 0.913 rev/sec. The CMP AX, CONRAM instruction subtracts CONRAM (a value of 28 counts in the Program Equates table) from AX. If AX > CONRAM, the CF is not set and the JNC transfers control to TMR2EX; there is nothing more to do in this pass through ZIPDN.

If AX < CONRAM the CF is set, and control falls through the JNC TMR2EX instruction. SPEED is set to 1 which means that on the next pass through the first part of the TMR2 code, when SPEED is tested for a value of 10, it will never be detected and the JMP MOTION instruction will never be executed. Motion analysis will not be executed during VERGIT (the convergence code). The RAMPOS value is made 6 so that on the next entry to TMR2, control is indexed to the VERGIT drive regime. Control is then transferred to TMR2EX which finishes the ZIPDWN code.

MAINNY (RLM, [R588]) is the main drive regime which accomplishes most of the motion control at the highest drive speed. Since the drive rate is constant, the D/A drive remains at the last value set by ZIPUP. In addition to decrementing SPEED, the principal function of MAINNY is to test position for the second break point which is the threshold for the transition to the ZIPDWN code.

The first operation in MAINNY is to decrement SPEED; if the result is not zero, the JNZ MAINN0 transfers control to MAINN0. If SPEED became zero, the ZF is set and control falls through the JNZ to reset SPEED to 20.

The direction of motion must be identified since the tests are comparisons which use the state of the CF to indicate the break point transition. As in the ZIPUP and ZIPDWN codes above, one case uses the set state of the Cf and the other uses the cleared state to indicate that the drive position has reached or slightly passed the break point threshold.

At MAINN0 the current position (POSD) is loaded into AX for the break point tests. In the TEST LDIR,1 instruction, the ZF is set if the drive is moving CW (UP); the position values are increasing. Control falls through the JNZ MAINN2 instruction. If the ZF is not set, the drive is moving in the CCW (DOWN) direction; position values are decreasing. Control is transferred to MAINN2.

In the CW case, control falls through the JNZ MAINN2 to the CMP AX, BREAK2 [R595] instruction which subtracts BREAK2 from POSD in AX. If the CF is not set, it means that POSD > BREAK2 and the drive has just passed through the break point so the JNC MAINN1 transfers control to MAINN1 which sets RAMPOS to the ZIPDWN value. If the CF is set, it means that POSD < BREAK2 and that the drive has not reached the break point; the JNC MAINN3 transfers control to MAINN3 to exit MAINNY.

In the CCW case, the MAINN2: CMP AX, BREAK2 [R600] instruction performs the second comparison. Remember that the drive is moving down and values are decreasing (i.e. going more negative) in the CCW direction. If the CF is set, it means that the POSD value is more negative than the breakpoint; hence the threshold has been reached or passed and it is time to make the transition to the ZIPDWN code. In this case the JC MAINN1 transfers control to MAINN1 which causes the RAMPOS change. If the CF is not set it means that POSD is more positive than the break point so control falls

through the JC to MAINN3.

At MAINN1 [R598], the RAMPOS value is increased to 4 so that in future passes through TMR2, program control will be indexed to the ZIPDWN code.

All paths through MAINNY exit through MAINN3 [R602] which causes the 12 bits of the D/A set-point value standing in Ports A and B of EPROM1 to be re-latched into the D/A drive register. This happens on every pass through MAINNY and is a precautionary measure to protect the D/A drive register from noise perturbations. After latching the D/A, control is passed to TMR2EX which is the exit path for the TMR2 subroutine.

VERGIT (RLM 9, [R484]) is the last drive regime which moves the drives to the set point at a low speed. The transition to VERGIT from ZIPDWN occurred when the velocity drive became less than 28 counts; this produced a Focus velocity of 0.2114 rev/sec and a Rotation velocity of 0.913 rev/sec. Motion analysis is not performed in VERGIT because the drive motion is so small; having reached VERGIT without motion faults, it is safe to assume that motion analysis is no longer required. VERGIT will shut down the drive when the position error is less than 4 counts.

In the event that the drive over shoots the commanded position, the velocity is ramped to zero speed and a new command sequence is initiated. DRVATV (drive active flag) is cleared, the CMDREQ (command request flag) is set, the temporary command argument is put in POSCEC (command echo) and the drive is shut down. When BOSS tests DRVREQ, it will initiate a new command sequence. If in executing the new command the error is still greater than 4 counts, the program assumes that there is something wrong with the drive; it may be sticking or the position readout hardware is malfunctioning. In this case the drive fault flag is set in FAUL1 and the drive is shut down.

VERGIT operates on conjunction with OFFIT which turns off the drives; the linkage condition to OFFIT are described as they occur. OFFIT is described below.

Remember that ZIPDWN made the transition to VERGIT when the velocity was less than 28 TMR2 counts to produce a Rotation drive velocity of 0.913 rev/sec and a data rate of 1869 counts/sec. The TMR2 sample rate is 1800 Hz so that TRMR2 samples Rotation position states 0.962 times for each state. The Focus drive velocity is 0.2114 rev/sec which produces a data rate of 95.54 counts/sec. TMR2 samples Focus position about 18.84 times for each position count state. Since the Focus data is 14-bit data left-shifted to make a 16-bit value, the actual change in Focus position states is four times slower. Thus TMR2 samples Focus position 75.36 times in each 14-bit state. The high TMR2 sampling rate of the position states is an important property. The reason that the drive is ramped down to a low speed is to enable high resolution drive to the set point.

Upon entry to VERGIT, [R484] the first test performed is a BOUND test of the position error against the CLSETAB (RLM 15, [799]) limits of +/- 4 counts. If (as indicated by the state of the BFLAG) the error is greater than 4, counts control is passed to VERGI1 which tests for the possibility that the command set point has been overshot. (VERGI1 is described below.) If the error is less than 4 counts, because of the high sampling rate of TMR2 relative to the position change, the drive has probably not overshot the set point and continued drive with a reduced D/A value will probably bring the drive to a near-perfect stop.

If the position error is less than 4 counts, the state of the D/A drive and the 4 associated control discretes is read [R490] into AX from EPROM1 Ports A and B and is pushed onto the STACK to save the state of the control discretes. The D/A drive is reduced by 4 counts. Since we have discussed the

incremental change of D/A values in the ZIPUP and ZIPDWN descriptions above, it should not be necessary to repeat the details. After this D/A drive reduction, the value is compared with zero by the CMP AX,0 [R501] instruction. If equal to zero (the ideal situation), control will fall through JNE VERGIA where the AND WORD PTR FAUL1,0FDFFH instruction will clear the drive fault flag (bit 9) in FAUL1. This is done just to affirm that the command was successfully completed within the error tolerance and there were no drive faults.

If in the CMP AX,0 [R501] test directly above the D/A drive is not zero, the state of the 4 control discretes in BX is merged [R505] with the reduced D/A drive value in AX, output to the EPROM1 ports and latched into the D/A drive register with the usual sequence of instructions. Having set this reduced drive, control is transferred to TMR2EX (RLM 12, [R668]) which is the TMR2 subroutine exit.

VERGI1 [R511] is entered if the BOUND BX,CS:CLSETAB [R487] test at the start of VERGIT determined that the error was greater than 4 counts; this path is the continuation of the VERGIT drive to the command set point. The code logic in this path tests for the possibility that the drive has overshot the set point; if not, the D/A drive value in EPROM1 Ports A and B is re-latched into the D/A register to insure the continuation of this D/A drive value and control is transferred to TMR2EX, the TMR2 subroutine exit.

The overshoot test [R513] compares the initial direction LDIR with DIR. Remember that DIR shows the direction to move the drive to null the error. The state of DIR was determined near the start of the TMR2 subroutine by a call to DSTOR, at most only a few hundred microseconds earlier than this test. In VERGIT the drive velocities are low; remember that each state of the Focus position data is sampled (by TMR2) 8.9 times per data state and the Rotation position data is sampled 1.9 times per data state. Given this high sampling rate relative to the changes in drive position, Focus position overshoots are improbable but not impossible. In the case of the Rotation drive, gear cogging, localized frictional sticking, improper operation of the backlash controller, etc. could cause the drive to move in a pausing-lurching manner which could cause overshoots, because of the lower state data sample rate. Missing codes in the R/D converters could also induce overshoots; if the command set point is a missing code, overshoots are very likely but not inevitable since the convergence criteria is an error of less than 4 counts.

A sudden reversal of the position error direction (i.e. DIR), particularly when the position error is large, is a serious concern because it suggests a malfunction in the drives or position readout circuitry; the code from VERGIB through VERGI3 reduces the drive to zero and on a one-time basis, it initiates a new command to the set point. In the event that the overshoot reoccurs after a second command attempt, the drive is shut down and a drive fault bit is set.

If the error is greater than 4 counts and the drive has not overshot the command set point, the D/A value in EPROM1 Ports A and B are again latched into the D/A register (just to be sure of the state) and control is transferred by a JMP TMR2EX [R517] to the TMR2 subroutine exit.

If the error is greater than 4 counts and the drive has overshot the command set point as indicated by the change in the sign of DIR relative to LDIR, the VERGIB path is taken. If an overshoot occurs, the VERGIB strategy is to initiate a new command to the set point as if it were a new computer-generated command but with the DRVONE flag set. If on executing the new command it turns out that again the drive overshot the command set point, no further command execution is attempted and the drive fault flag in FAUL1 is set.

At VERGIB [R518] (in the overshot path case) the now-familiar sequence to decrease the D/A drive by four counts is begun. The LDIR sign is tested to determine whether to subtract 4 counts from a

positive D/A drive value or to add 4 counts (by subtracting a minus 4) to make a negative drive 4 counts more positive. After calculating the new D/A drive value, the 4 discrete control bits are merged with the new D/A drive value, set into the EPROM1 and latched into the D/A. After calculating the new D/A value, it is pushed into the STACK for temporary storage.

After the new D/A drive value has been latched, the STACK is POP-ped onto AX to see if the drive has been reduced to zero. The CMP AX, 0 [R535] instruction tests for this case; if not zero, control falls through the JE VERGI2 instruction to the JMP TMR2EX to the TMR2 subroutine exit. Several more passes through this pass may be required to reduce the D/A drive to zero. (Since the VERGIT drive rate is 28 counts and the velocity is reduced in 4-count increments, a value of zero is inevitable).

If the D/A drive is found to be zero in the test [R535] above, control is passed to VERGI2 by the JE VERGI2 instruction.

Control passed to VERGI2 because of an overshoot past the command set point. VERGI2 logic will initiate a new command to the set point if this is the first occurrence of an overshoot. The second occurrence of an overshoot is determined by testing the state of the DRVONE flag, if set, the program assumes that something is faulty and sets a Drive Fault flag bit (bit 9) in FAUL1 by the OR WORD PTR FAUL1,200H instruction. Having set this flag, control is transferred to the OFFIT code by a jump to OFFIT3 (RLM 12, [R656]).

The action of VERGI2 is to permit only one attempt to re-command the drive to the set point in the event of an overshoot; this logic prevents the possibility an endless sequence of such attempts which are probably the result of some malfunction.

If the entry to VERGI2 was the consequence of an overshoot and this is the first occurrence, the DRVONE flag is set, the DRVATV flag is cleared and the DRVREQ flag is set. The current command set point in POSEC is loaded into COMTMP. With the exception of setting DRVONE, these are the conditions set by DRVINT when a new command is about to be initiated. Having set up these conditions, JMP TMR2EX transfers control to OFFIT3.

Does setting DRVONE inhibit the execution of future computer position commands? No, remember that the DRVONE flag is cleared in POSCM1 [R370] in the DMA0 subroutine.

OFFIT (RLM 12, [R635]) is drive problem code which prefixes the TMR2EX code and is called when the drive should be shut down because some malfunction or exceptional condition has interrupted the execution (or impending execution) of an active command. Examples of the malfunction case are limit faults sensed in DRVINT and LOCAL which set the DRVOFF flag; TMR2 tests the DRVOFF flag and if set, a JMP OFFIT transfers control to OFFIT to shut things down. Examples of the non-fault invocation of OFFIT are the manual override test in LOCAL in which it is discovered that the manual override command has just pre-emptively overridden the LOCAL mode; this condition sets the DRVOFF flag. A new command from the antenna control computer will override a command being executed and cause the DRVOFF flag to be set.

OFFIT has two entry points; the top entry, OFFIT [R635] is the entry point for situations in which the drive must be slowed down before being stopped and OFFIT3 [R656] is the entry point which is used when the drive has stopped. OFFIT3 is entered by the drive overshoot code discussed above and by code near the start of TMR2 which senses that the Emergency Stop or Drive Lockout signals have become active (true).

TMR2 Interrupt Subroutine (continued)

At the OFFIT entry, the EPROM1 port states are loaded into AX and SPEED is decremented just as in ZIPUP, MAINNY and ZIPDWN. However in this code, motion analysis is not invoked; SPEED is used to control the ramp-down rate. If decrementing SPEED did not set the ZF (at the count of zero), control is transferred by JNZ OFFIT2 down to OFFIT2 [R653] to test the drive rate.

If decrementing SPEED set the ZF, it is time to reduce the D/A drive by 4 counts. SPEED is first reintialized to 20 counts for subsequent passes through the TMR2-OFFIT loop. As in the many cases of D/A drive change described above, the state of the 4 control discretes are saved by being PUSH-ed onto the STACK. These bits are stripped off the top 4 bits by an AND AX, 0FFFH, the value is reduced by 4 counts by a subtract which is LDIR-dependent. The new drive value is merged with the 4 control discretes POPP-ed off the STACK, output to the EPROM1 ports and latched into the D/A drive registers. Having completed this D/A drive reduction control passes to OFFIT2 which tests the D/A drive level to see if it is zero.

At OFFIT2, [R653] the D/A drive level is tested for zero. AX contains the EPROM1 ports A and B states (either from the OFFIT entry or the code immediately above); the top 4 control discretes are masked off and tested by the CMP AX,0 instruction. If not yet zero, the JNE TMR2EX instruction transfers control to TMR2EX which is the exit point for the TMR2 subroutine. Subsequent passes through TMR2-OFFIT will reduce the D/A drive to zero.

OFFIT3 [R656] was entered if the drive reached zero; it is now safe to start shutting down drive-associated functions. The first action is a CALL TMROFF which shuts down the timers. The EPROM1 ports are loaded with 0C000H which keeps the 3-phase relay and brake energized and inhibits the BDS3 servo amplifier. The EOI (end of interrupt) code of 8 is loaded into AX and EOI register address is loaded into DX and output; this resets the IS (in service) bit associated with Timer 2. See the Interrupt discussion in Section 3.1 or the 80188 data sheets in Section 8, VOL II. The BRKOFF subroutine is called which turns off the brake. The DRVATV and DRVOFF flags are reset. Finally the state of EPROM2 ports A and B are read into AX. AND AX, 0FF97H clears the drive bits in AX associated with the S105 CMD,BRAKE, DRIVE UP and DRIVE DOWN LED's but does not affect the drive associated the limit LED's or analog multiplexer address states. After performing the operations above, control falls to TMR2EX which is the exit point for all the TMR2 paths described above.

TMR2EX 9 (RLM 12, [R668]) is the TMR2 subroutine exit code for all the paths through TMR2 described above and performs the operations of clearing the interrupt IS (in service) bit, re-establishing the TIMER 2 interrupt priority and popping the pre-interrupt status of some registers off the STACK.

The first operation is to load AX with bit 8, the timer IS (in service) bit, and output to the EOI (end of interrupt) register. After this a 3 is loaded into AX and output to the EOI register; this establishes the priority of the Timer 2 interrupt. The BFLAG, DX, CX, BX and AX registers are loaded with their pre-Timer 2 interrupt status. The final instruction is the IRET which pops the IP, CS and Flags status off the STACK onto these registers.

MOTION Subroutine (FLM 14, RLM 13)

MOTION (RLM 13, [R680]) is the code which analyzes drive motion to detect drive dragging or sticking. It operates upon differences of position and velocity. MOTION also samples the BDS3 servo amplifier fault discretes; in the event that a BDS3 fault is detected the drive is shut down.

MOTION is entered from the TMR2 subroutine just before the indexed jump to the ZIPUP, MAINNY and ZIPDWN drive regime codes; therefore it has no particularization to these regime codes. Motion is

entered when SPEED has reached the value of 10 after being decremented from 20; thus it uses positions and velocities which are intermediate from the 20 count points at which they are changed. Since MOTION is invoked every 20 TMR2 cycles, the MOTION sample rate is 90 Hz.

On entry to MOTION [R680], the first action is to enable the interrupts by the STI instruction. Why do this? TMR2 was entered by Type 3 Interrupt, in responding to the interrupt, the CPU disables single-step and the lower priority maskable interrupts. STI enables the processor to recognize maskable interrupts. TMR2 will be reading data from the A/D converter which uses INT0 to signal that the A/D data is available. The EPROM2 port states are loaded into AX and masked by AND AX 0FFF8H which clears the lower three bits while retaining the other bits intact. These lower three bits are the A/D converter multiplexer address bits. An address of 1 is merged into AX and output to the EPROM2 ports. This address selects the BDS3 servo amplifier velocity signal for conversion by the A/D converter. The address of the A/C converter start convert (STCNV) is loaded into DX and the OUT DX,AL instruction initiates the conversion which is triggered by the address enable. The value in AL is irrelevant; it goes nowhere because the important action is triggered by the enable. The conversion time is 35 usec or less; the MOTIO3 code (below) tests the EOC signal (ADFLAG) for availability of the A/D data.

SPEED is decremented; on passes through ZIPUP, MAINNY or ZIPDWN SPEED would be decremented. It is done here to maintain the decrementing rate through TMR2 passes.

BDS3 servo amplifier faults are read next; MOV DX, BDERL1 [R690] loads the address of the read enable and IN AX,DX inputs the data into AX. In the case of Rotation, the fault state of two servo amplifiers are read into AX. Focus has only one amplifier; its fault states are read into AL. After reading the fault states, they are stored in FAUL2 to enable readout of the BDS3 fault status as monitor data. The AND AX, 0FDFDH (Rotation) pattern forms the logic product of all fault bits with the FDFDH bit pattern. In the case of Focus, the AND AX 0FDH forms the logic product of the FDH pattern and AL. If any fault bit is set, the ZF is cleared. The D's in the pattern except the Torque Foldback bits from the test. Foldback will occur if the motor is loaded above its rated torque (but below the peak torque limit) until the energy input to the motor is equivalent to the motor's RMS rating for about 15 seconds. At this time, the amplifier limits the torque developed by the motor to its rated value. The motor and amplifier can operate in this limiting condition indefinitely, even at dead stall. Torque foldback is an abnormal condition resulting from either a mechanical binding in the drive or excessive lubricant viscous friction due to cold weather. Experience has shown that in cold weather, viscous friction can exceed the Focus torque rating. For additional details on viscous friction effects see Section 2.0. For details on the meaning of the BDS3 fault bits see Section 3.6.

If the ZF is not set in the AND AX, 0FDFDH [R694] test above, the DRVOFF flag is set and control is transferred to TMR2EX, the exit from the TMR2 subroutine. On the next pass through TMR2, the state of DRVOFF is tested; if a 1, JMP OFFIT (RLM 9, [R635]) starts the drive slowdown and stop in the OFFIT code.

At MOTIO1, [R698] the analysis of motion is begun using difference equations involving current (i.e. from this pass through TMR2) velocity and position and previous velocity and position data saved during the previous pass through TMR2. The previous pass would have indexed to one of the drive regime codes: ZIPUP, MAINNY or ZIPDWN. DSTOR is called early in TMR2 at TMR210 [R461] to acquire and store the current position in POSD. Thus the interval for position samples is 1/1800 Hz or 0.555 milliseconds. Velocity data is acquired at the entry to MOTION and was acquired by the same call to DSTOR which acquired the position data in POSD. The current position is stored in POSDED (old position) for use in the next pass through MOTION, 20 TMR2 cycles later. SUB BX,AX subtracts POSDED (old position) from POSD (new position) and leaves the difference in BX; the difference can be either

negative or positive depending upon the direction of motion. If moving CW (UP) the difference is positive; if moving CCW (DOWN) the difference is negative. If positive, control is passed to MOTIO2 [R705]; if negative, BX is negated so that in either case, BX contains a positive or absolute value of velocity.

At line [R687] the A/D conversion was started, at MOTIO3: TEST ADFLAG, 1 tests the state of the DFLAG (set to 0 above); if the ZF is zero, the conversion has not been completed so the JZ MOTIO3 returns control to the TEST instruction until the ADFLAG is a 1; at this time the A/D (velocity) data is available. The difference equations deal with absolute values and velocity can be either positive or negative, depending upon drive direction. TEST AH, 8 tests bit 11, the sign bit; if set, the ZF is cleared and control falls through the JZ so that the velocity value in AX is negated to a positive 2's complement value. If positive, the JZ passes control to MOTIO4 where the upper 4 bits of the ADVAL are zeroed. ADVAL data from the A/D converter is right-shifted four bits by the INT0 subroutine so the 12 bits of velocity data are in the lower 12 bits of AX. The masking instruction is an additional precaution to insure that this is the case.

In the division performed in MOTIO4, CL is loaded with a divisor of 25 for Focus and 20 for Rotation. This divisor scales time, the drive velocity and the A/D converter output to produce the average motion delta.

In the next section of code, the old (i.e. previous pass through MOTION) and current-pass velocity data is scaled and multiplied by time to develop an averaged, velocity-derived motion delta which is compared with measured position differences to verify that the drives are moving properly. We want to do arithmetic to transform measured velocity to distance. The scaling is not very obvious; the derivation is as follows.

ADVAL contains the velocity readout from the IDD servo amplifier R/D converter and is a direct measure of motor velocity. ADVAL is scaled at 5 millivolts/count. The two velocity readouts are scaled as follows: for Focus it is 1800 RPM/8 readout volt or 3.75 rev per sec/readout volt and for Rotation it is 1.04 rev per sec/readout volt.

For Focus, in terms of counts/readout volt, the realized motion is: (452 counts/rev)*(3.75 rev per sec/readout volt) = 1695 counts per sec/readout volt. The samples are separated in time by 20 TMR2 cycles and the TMR2 cycle rate is 1800 Hz, so multiplying by time we get: (1695 counts per sec/readout volt)*(20 TMR2 cycles/1800 TMR2 cycles per sec) = 18.83 counts/readout volt. The comparable Rotation value is 23.7 counts/readout volt.

This factor, when multiplied by the 0.005 millivolts/count A/D converter scaling and the velocity readout, is the distance traversed (in counts) during one TMR2 cycle. Two velocity samples are gathered, one sample during the previous pass and one during the current pass. The two velocity-derived motions are summed and divided by two for averaging. To simplify calculations, each motion is divided by two before summing. Thus we can say that the equation for the velocity-derived Focus motion is:
MOTION = ADVAL*0.005*18.83*0.5. Numerically, the product of the 0.005*18.83*0.5 factors is 0.047. This is an awkward factor for a fixed point multiplication processor; a scaling subroutine would have to be used which would require a lot of processor time and code. An alternative to multiplication is to take the reciprocal value and use it as a divisor; thus 1/0.047 = 21.24 for Focus. The comparable Rotation value is 16.875. Clearly division is the simplest scaling technique. In the calculations below, Focus uses a value of 25 and Rotation a value of 20.

The motion value in AX is divided by CL which was loaded with the value of 20 (Rotation) or 25 (Focus) as discussed above. This DIV CL instruction is a byte divide which has the numerator in AX and

MOTION Subroutine (continued)

the divisor in CL; the quotient will be placed in AL and remainder in AH. Note the op code: F6 F1. Looking at the IPRM we see that w = 0 which specifies that the numerator is AX, the divisor is EA (i.e. CL), the quotient will be in AL and remainder in AH. F1 designates a mod of 11 (r/m designates a register); 110 is null and r/m designates CL. The remainder in AH is cleared to zero; it's too small to be significant. Thus we have divided the recent motion by 20. This motion data (divided by 20) is stored in VEL1, the previous value for use in the next pass through MOTION. At [R705], DX was loaded with the previous pass motion sample (divided by 20). ADD AX, DX adds the current and previous motion (divided by 20) samples and leaves the sum in AX. ADD BX, 4 adds 4 counts to the difference between positions obtained between the MOTION pass and the previous MOTION passes. Both AX and BX contain absolute values (i.e. positive values). The next instruction: CMP AX, BX [R719] compares the two values by subtracting BX from AX.

We may picture the CMP AX, BX instruction as:

$$\underset{\text{CMP}}{\phantom{\text{CMP}}}\ \underset{(AX)}{}\qquad\qquad\underset{(BX)}{}$$
$$\text{CMP}\ (\text{VEL MOTION}_{prev} + \text{VEL MOTION}_{curr})\ (\text{POS}_{prev} - \text{POS}_{curr} + 4).$$

The CMP instruction subtracts (BX) from (AX) and sets the flags depending upon the relative values. If (BX) > (AX) or (BX) = (AX), the JLE MOTIO5 instruction transfers control to MOTIO5 [771] which is an unconditional jump to TMR2EX, the normal no-problem exit from the TMR2 subroutine. If (BX) < (AX), it means that the realized motion represented by the position samples was less than the velocity-derived motion and something is wrong with the drives. The 4 counts bias to the position difference gives the position difference values the benefit of the doubt.

If the JLE jump is not executed, the DRVOFF flag is set and the DRIVE FAULT flag is merged into the FAUL1 flag word. The DRVOFF flag will be detected in the first part of the next pass through TMR2 which will initiate a drive shutdown as described above.

Focus Second Screw Code (FLM 9)

The Focus version of TMR2 includes code to analyze the motion of the second screw relative to the Focus position. The second screw sensor outputs a square-wave signal (two-states) when the second screw is rotating. This signal is generated by a 10-cycle/revolution sensor and is alternately high (i.e. a 1) for about 22.6 Focus position counts and low (i.e. a 0) for about 22.6 position counts. The initial state may be high or low. The code tests this signal to verify that the second screw is moving in unison with the directly driven shaft; in the event that the second screw position is outside the allowable angular tracking tolerance, the Focus drive is to be immediately shut down, regardless of the drive velocity. To understand the function performed by this code, the reader should refer to Figure 7 and Section 2.2.

The weakest link in the Focus drive is the flexible shaft between gearboxes; a slippage of couplings or a break in the flexible shaft drive to the second screw could seriously overstress the FRM. The manifestations of a malfunction are a cessation of signal changes or an increase in the number of position counts between transitions. One revolution of the motor shaft produces 451.97 counts of position change. The flexible shaft has a fairly low spring rate so that when a new command causes motion in a direction opposite to the previous command, there will probably be a spring-windup delay in the occurrence of the first signal transition. The consequence is that in this start-up state, there may be more position counts than the typical 22 or 23 experienced after motion is underway. This initial state could be any value from zero to the maximum permissible count of 50. The actual value depends upon many factors: the initial conditions of the second screw sensor and the Focus position, the direction of motion (it's harder to drive Focus up so there is more flexible shaft spring wind-up), whether there has been a direction change, the

polarity of the change (i.e. down to up versus up to down) and the viscous friction load imposed upon the second screw. The mechanical conditions are rather complicated and most of them are not accessible to the program; many of them are not characterized.

From mechanical considerations, the tolerance on the second screw tracking is 40 degrees (that is the second screw angular position must not differ from the first screw angular position by more than 40 degrees); if it exceeds this value the Focus drive is to be shut down. In terms of position counts this is: (40/360)*451.97 = 50.22 counts. The code uses a tolerance (SCREW) of 50 counts. If the signal duty cycle is 50% and the Focus drive is moving smoothly, there are 22.59 counts/state (i.e. 451.97/20) which is 18 degrees/state; thus the fault threshold value is about twice the expected 50% duty cycle value.

A second complication is that the second screw sensor may not have a 50% duty cycle; it may be as poor as 40/60 and be either 60% high or 60% low. The code must tolerate the effects of a state transition-to-transition "jitter" but the states cannot exceed this 50 count tolerance limit. If the tolerance is exceeded, the Focus drive is immediately shut down with fault flags in FAUL1 and the second chance flag (SECCH) is set. SECCH is tested in DRVINT. If in executing subsequent Focus position commands another second screw failure occurs, the drive will be shut down and Focus position commands will not be executed. Second screw failures are indicative of a potentially serious problem in the Focus drive; the drive should be inspected by someone thoroughly familiar with the FRM. The only way to clear SECCH is to issue a soft or hardware reset command which will clear the RAM memory. Clearing SECCH should only be done with the permission of a responsible NRAO supervisor.

Before proceeding further, it's probably best to describe some parameters which are used in the second screw analysis: PHASEA is the value of the current state under test and can be either a 1 or 0. The actual value is not important but the change in PHASEA is important and signals the start of a new state to be tested. POSD is the position of the Focus drive, obtained by the call to DSTOR mentioned above. SCRCNT is the accumulating count of the state under test; this count must never exceed the value of 50 for the reasons described above. SCRLST is the position sample saved in the previous TMR2 pass through the state under test (i.e. there has not been an intervening signal transition). DEL is a more local count of position change and there are two versions of DEL: DEL_{new} is the current difference between POSD and SCRLST in the current pass and DEL_{old} is this difference in the preceding pass through the same state. FIL is a flag which indicates that the POSD - SCRLST changes are larger than should be expected (by a factor about 6 at the highest Focus drive speed). FILOVR is a count of the occurrence of these excessive counts. These events may be attributable to the TMR2 interrupt being skipped.

The second screw code begins at line [F480] (FLM 9) with a test of the screw-ignore (SCWIGN) flag; if the flag is set, the ZF is not set and control is transferred to TMR22 [S16] which tests the state of the FIL flag.

If the SCWIGN flag is not set, the JZ TMR22B directs control to TMR22B. At TMR22B the second screw signal is tested for a 1/0 or a 0/1 transition. In the code immediately above (i.e. TMR210, DX had been set to the address of the Apex data discretes byte which contains the second screw discrete bit. The discretes data is read into AL. The AND AL, 32 [F484] instruction selects this bit and zeroes the balance of AL. On the previous pass through this TMR2 code, PHASEA was set to the state of this discrete. The XOR PHASEA, AL instruction tests the state of this new sample (remember that DSTOR was called above in TMR210) against the previous sample; if they differ, the ZF will not be set and control is passed to TMR21C [R490]. Note that the actual state of the discrete is unimportant since the test is an exclusive or. After the XOR instruction, the state of the second screw bit in AL is stored in PHASEA for the next TMR2 pass.

If there was a transition in the state of PHASEA, the motion counter (SCRCNT) is initialized to zero for analysis in subsequent passes through this state under test. Since this is just the beginning of a new state, there is no action to be taken; control is passed by an unconditional jump to TMR22 [F516] which tests the state of the FIL flag.

If there was not a transition, the current state is to be analyzed in terms of the position change since the last transition. When the Focus drive has reached a steady-state and it is moving properly, the transitions should occur at intervals of 22 or 23 counts of position change. At the maximum Focus drive rate, Focus position changes 2.145 counts for each pass through TMR2; thus there are about 10 TMR2 passes through the state under test at this maximum speed and many more at lower speeds.

If there was not a transition, the position sampled (SCRLST) during the previous pass through the current state is compared with the current position, POSD. SCRLST is loaded into AX and the current position POSD is subtracted from SCRLST; the difference remains in AX. This local value is labelled DEL_{new} (see the note above). The sign of the difference is not important since the BOUND instruction tests the motion change. The BFLAG is first cleared and the BOUND [F493] instruction tests this difference against the DELTA table limits [F862] of +/- 12 counts. The purpose of this comparison is described below. This value in AX, the local difference, (i.e. DEL_{new}) is a measure of the smoothness of the position changes between passes through the state under test. If DEL is less than +/- 12, the test is proceeding smoothly and control is passed to TMR2CC which clears the FIL flag. (The function of the FIL flag is described below.) If the POSD - SCRLST difference is greater than +/- 12, the drive is not moving as smoothly as it should be but this indication is only preliminary. In this event control falls through the JZ TMR2CC instruction [F495] to set the FIL flag and other things described later.

Remember that the Focus readout is a 14-bit value, left shifted to make a 16-bit value convenient for processor operations; thus the 12-count limit above is really a 3-count limit in terms of the 14-bit position.

At TMR2CC, the FIL flag is cleared to zero and DEL_{new} is added to SCRCNT so that SCRCNT increases for each pass through this state under test. DEL_{new} is also stored in DEL and will be used when DEL_{old} is called as described below. The current position POSD is stored in SCRLST to be tested in the next pass through the state under test. SCRCNT is loaded into AX for the impending BOUND test. The prime function of the second screw code is the BOUND AX,CS:SCREW instruction which compares SCRCNT (in AX) with the SCREW table ([F892], FLM 16) limits of +/- 50. If the SCRCNT has reached 50 counts, the BFLAG will be set so that the TEST BFLAG, 1 instruction will clear the ZF. Control will fall through the JZ TMR22 to or into FAUL1 the pattern 8200H. These two bits are the second screw and drive fault flags. After setting these flags, the SECCH (second chance) flag is set and an unconditional jump transfers control to OFFIT3 which immediately shuts down the Focus drive. In the test of the difference between SCRLST and POSD in [F493] above, if the difference exceeded +/- 12 counts, the FIL flag is set indicating a large pass-to-pass difference. In this case the DEL_{old} value is loaded into AX and the FILOVR counter is incremented. The CMP FILOVR, 10 tests this counter; if it equals 10, the 0A00H fault pattern is or-ed into FAUL1. These bits indicate a Drive Fault and Emergency stop. Having set these fault bits, an unconditional jump transfers control to OFFIT3 to immediately shut down the Focus drive.

In the comparison of FILOVR with 10 if FILOVR is less than this value, the JNZ TMR2CD instruction transfers control to TMR2CD: ADD SCRCNT, AX [F504] in the path described above. In this instruction, AX will be added to SCRCNT but in this case AX will contain not the DEL_{new} value but the DEL_{old} value from the previous pass. What has been done is to substitute a more plausible, recent value for DEL than the large, questionable value obtained in this pass. Other than this substitution, the balance

of the pass is just as described above.

This concludes the control firmware description.

3.9 SYSTEM ANALOG MONITOR DATA, FOCUS AND ROTATION DRIVE CONTROL COMMAND AND MONITOR DATA FORMATS

ID CODE = 72H, STARTING ADDRESS = 7200H

SYSTEM ANALOG MONITOR DATA

SYSTEM ANALOG MONITOR DATA FORMAT: SIGNED WORD FORMAT

REL ADDR #(HEX)	ANALOG/ DIGITAL	ACCESS MON/CON	DESCRIPTION BIT POS	FUNCTION	MULT
00	ANALOG	MON		SYSTEM DIGITAL GROUND	1
01	ANALOG	MON		SYSTEM +15 POWER SUPPLY	2
02	ANALOG	MON		SYSTEM -15 POWER SUPPLY	2
03	ANALOG	MON		SYSTEM +5 POWER SUPPLY	1
04	ANALOG	MON		WEST ROTATION SERVO (BDS3) +12 POWER	2
05	ANALOG	MON		WEST ROTATION SERVO (BDS3) -12 POWER	2
06	ANALOG	MON		EAST ROTATION SERVO (BDS3) +12 POWER	2
07	ANALOG	MON		EAST ROTATION SERVO (BDS3) -12 POWER	2
08	ANALOG	MON		SYSTEM DIGITAL GROUND	1
09	ANALOG	MON		BACKLASH CONTROLLER (RDP2) +12 POWER	1
0A	ANALOG	MON		BACKLASH CONTROLLER (RDP2) -12 POWER	1
0B	ANALOG	MON		FOCUS SERVO (BDS3) +12 POWER	2
0C	ANALOG	MON		FOCUS SERVO (BDS3) -12 POWER	2
0D	ANALOG	MON		SYSTEM DIGITAL GROUND (SPARE)	1
0E	ANALOG	MON		SYSTEM DIGITAL GROUND (SPARE)	1
0F	ANALOG	MON		SYSTEM DIGITAL GROUND (SPARE)	1

STANDARD INTERFACE BOARD INTERNAL COMMANDS AND DATA

The Standard Interface board has 16 internal command and monitor data addresses which are peculiar to the function of the interface board. The monitor data values are mostly counters whose states are indicative of the performance of the interface board and the device which the board services. The counters may be set to a new value by a command which has the same address as the monitor data address; the value is the command argument. The most probable usage of these commands is to clear the counters by a command having an argument value of zero. These 16 addresses are the last 16 in the address block assigned to the device. These addresses and functions are as follows:

REL ADDR #(HEX)	ANALOG/ DIGITAL	ACCESS MON/CON	DESCRIPTION BIT POS	FUNCTION
60	DIGITAL			NOT ASSIGNED
61	DIGITAL			NOT ASSIGNED
62	DIGITAL			NOT ASSIGNED
63	DIGITAL	MON		NO COMMAND DEVICE ACKNOWLEDGE, COUNTS
	DIGITAL	CON		RESETS THE COMMAND DEVICE ACKNOWLEDGE COUNTER TO THE COMMAND ARGUMENT
64	DIGITAL	MON		NO MONITOR DEVICE ACKNOWLEDGE, COUNTS
	DIGITAL	CON		RESETS THE MONITOR DEVICE ACKNOWLEDGE COUNTER TO THE COMMAND ARGUMENT
65	DIGITAL	MON		INTERFACE TYPE/SOFTWARE REVISION (OF THE STANDARD INTERFACE BOARD)
66	DIGITAL	MON		LAST COMMAND ADDRESS
	DIGITAL	CON		RESETS THE LAST COMMAND ADDRESS VALUE TO THE COMMAND ARGUMENT
67	DIGITAL	MON		LAST COMMAND ADDRESS ARGUMENT
	DIGITAL	CON		RESETS THE LAST COMMAND ARGUMENT VALUE TO THE COMMAND ARGUMENT
68	DIGITAL	MON		ADDRESS PARITY ERROR COUNTER, ALL ADDRESSES
	DIGITAL	CON		RESETS THE ADDRESS PARITY ERROR COUNTER TO THE COMMAND ARGUMENT
69	DIGITAL	MON		ARGUMENT PARITY ERROR COUNTER, ALL ADDRESSES
	DIGITAL	CON		RESETS THE ARGUMENT PARITY ERROR VALUE TO THE COMMAND ARGUMENT
6A	DIGITAL	MON		INVALID SYNC DETECTED COUNTER
	DIGITAL	CON		RESETS THE INVALID SYNC DETECTED COUNTER TO THE COMMAND ARGUMENT
6B	DIGITAL	MON		DATA PARITY ERROR COUNTER, THIS INTERFACE BOARD ONLY

	DIGITAL	CCN	RESETS THE DATA PARITY ERROR COUNTER TO THE COMMAND ARGUMENT
6C	DIGITAL	MON	ID BYTE VALUE, READ FROM DEVICE
6D	DIGITAL	MON	COMMAND COMPLETED COUNTER
	DIGITAL	CCN	RESETS COMMAND COMPLETED COUNTER TO THE COMMAND ARGUMENT
6E	DIGITAL	MON	MONITOR REQUESTS COMPLETED COUNTER
	DIGITAL	CCN	RESETS MONITOR REQUESTS COMPLETED COUNTER TO THE COMMAND ARGUMENT
6F	DIGITAL	MON	BLOCK START ADDRESS

FOCUS DRIVE COMMANDS AND FOCUS MONITOR DATA

DIGITAL DATA FORMATS: POSITION COMMAND AND POSITION READOUT: UNSIGNED WORD FORMAT
COMMAND ECHO & APEX ANALOG DATA: SIGNED WORD FORMAT
FOCUS STATUS WORDS: SYSTEM, FAUL1, FAUL2 & ANAFL - UNARY FAULT BITS

REL ADDR #(HEX)	ANALOG/ DIGITAL	ACCESS MON/CON	DESCRIPTION BIT POS	FUNCTION	MULT
10	DIGITAL	CON		FOCUS COMMAND: 2300 COUNTS/INCH	
				SOFTWARE LIMITED RANGE: 0A00H - F200H	
11	DIGITAL	MON		FOCUS POSITION	
	DIGITAL	CON		NAP COMMAND SET/RESET: 1=SET, 0=RESET	
12	DIGITAL	MON		FOCUS POSITION ERROR = COMMAND - ACTUAL	
	DIGITAL	CON		MANUAL COMMAND OVER-RIDE SET/RESET: 1=SET, 0=RESET	
13	DIGITAL	MON		FOCUS SYSTEM PARAMETERS, "SYSTEM"	
			0	NAP FLAG: 1 = IN NAP MODE	
			1	MODE: 0 = COMPUTER, 1 = MANUAL	
			2	BRAKE STATUS: 0 = ENGAGED, 1 = RELEASED	
			3	MANUAL OVER-RIDE MODE: 1 IN MANOVR, 0 = NOT IN MANOVR	
			4	BITS 4 - 15 NOT ASSIGNED	
	DIGITAL	CON		FOCUS SERVO AMPLIFIER (BDS3) RESET, ANY VALUE RESETS	
14	DIGITAL	MON		FOCUS FAULT DATA, "FAUL1" (1 = FAULT)	
			0	APEX INTERFACE NOT RESPONDING	
			1	FIRST LOWER LIMIT	
			2	SECOND LOWER LIMIT	
			3	FIRST UPPER LIMIT	
			4	SECOND UPPER LIMIT	
			5	COMMAND FAULT, M&C REQUEST OUT OF RANGE	
			6	MONITOR FAULT, M&C REQUEST OUT OF RANGE	
			7	OPERATOR FAULT, POSITION ARG OUT OF RANGE	
			8	BRAKE FAULT, BRAKES DID NOT RELEASE/ENGAGE PROPERLY	
			9	DRIVE FAULT, COMMANDED POSITION NOT ATTAINED, SOME MALFUNCTION, MECHANICAL OR ELECTRICAL	
			10	MOTION ANALYSIS FAULT, DRAGGING OR STICKING	
			11	EMERGENCY STOP SWITCH ON	
			12	DRIVE LOCKOUT FAULT	
			13	N/A	
			14	N/A	
			15	SECOND SCREW FAULT, SECOND SCREW IS NOT ROTATING	
	DIGITAL	CON		FOCUS SOFTWARE RESET; ANY VALUE RESETS	
15	DIGITAL	MON		FOCUS SERVO AMPLIFIER FAULT DATA,"FAUL2" (1 = FAULT)	
			0	BUS FAULT; 300 VDC HAS DISAPPEARED	
			1	MOTOR TORQUE FOLDBACK; BDS3 IS LIMITING MOTOR TORQUE TO RMS-RATING BECAUSE LOAD TORQUE HAS EXCEEDED RMS RATING FOR OVER 15 SECONDS	
			2	OVER-TEMPERATURE, BDS3 AMPLIFIER TOO HOT	
			3	DRIVE DOWN, SERVO AMPLIFIER NOT READY TO RUN	
			4	OVER-CURRENT, MOTOR TAKING TOO MUCH CURRENT	
			5	POWER LOSS, SERVO AMPLIFIER POWER SUPPLY IS OUT	
			6	REMOTE INHIBITED, F-R CONTROLLER HAS NOT RELEASED INHIBIT	
			7	OVER-SPEED, MOTOR RUNNING FASTER THAN IT SHOULD	
16	DIGITAL	MON		FOCUS APEX ANALOG FAULT FLAGS, "ANAFL" (1 = FAULT)	
			0	GROUND FAULT	1
			1	GROUND FAULT	1
			2	GROUND FAULT	1
			3	FOCUS VELOCITY	1
			4	+15	2
			5	-15	2
			6	+5	1
			7	+10	1
			8	MOUNT TEMP #1	10
			9	-10	1
			10	MOUNT TEMP #2	10

			11	BIN TEMP	10
			12	12 - 15, SPARE BITS	
17	ANALOG	MON		APEX GROUND: 0V	1
18	ANALOG	MON		APEX GROUND: 0V	1
19	ANALOG	MON		APEX GROUND: 0V	1
1A	ANALOG	MON		FOCUS VELOCITY	1
1B	ANALOG	MON		APEX +15 POWER SUPPLY	2
1C	ANALOG	MON		APEX -15 POWER SUPPLY	2
1D	ANALOG	MON		APEX +5 POWER SUPPLY	1
1E	ANALOG	MON		APEX +10 REF POWER SUPPLY	1
1F	ANALOG	MON		MOUNT TEMP 1 (100mV/DEG C)	10
20	ANALOG	MON		APEX -10 REF POWER SUPPLY	1
21	ANALOG	MON		MOUNT TEMP 2 (100mV/DEG C)	10
22	ANALOG	MON		BIN TEMP (100mV/DEG C)	10
23	DIGITAL	MON	0-7	FOCUS SOFTWARE REVISION NUMBER (BIT 7 IS MSB)	
			8-15	MODULE SERIAL NUMBER (BIT 15 IS MSB)	
24	ANALOG	MON		FOCUS VELOCITY (VEL1)	
25	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
26	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
27	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
28	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
29	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
2A	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
2B	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
2C	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
2D	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
2E	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
2F	ANALOG	MON		SPARE, NOT USED, AVAILABLE	

ROTATION DRIVE COMMANDS AND ROTATION MONITOR DATA

DIGITAL DATA FORMATS: POSITION COMMAND AND POSITION READOUT: UNSIGNED WORD FORMAT
COMMAND ECHO & APEX ANALOG DATA: SIGNED WORD FORMAT
ROTATION STATUS WORDS - SYSTEM, FAUL1, FAUL2 & ANAFL - UNARY FAULT BITS

REL ADDR #(HEX)	ANALOG/ DIGITAL	ACCESS MON/CON	DESCRIPTION BIT POS	FUNCTION	MULT
30	DIGITAL	CON		ROTATION COMMAND: 156 COUNTS/DEGREE SOFTWARE LIMITED RANGE: 0A00H - F200H	
31	DIGITAL	MON		ROTATION POSITION	
	DIGITAL	CON		NAP COMMAND SET/RESET: 1=SET, 0=RESET	
32	DIGITAL	MON		ROTATION POSITION ERROR = COMMAND - ACTUAL	
	DIGITAL	CON		MANUAL COMMAND OVER-RIDE SET/RESET: 1=SET, 0=RESET	
33	DIGITAL	MON		ROTATION SYSTEM PARAMETERS, "SYSTEM"	
			0	NAP FLAG: 1 = IN NAP MODE	
			1	MODE: 0 = COMPUTER, 1 = MANUAL	
			2	BRAKE STATUS: 0 = ENGAGED, 1 = RELEASED	
			3	MANUAL OVER-RIDE MODE: 1 IN MANOVR, 0 = NOT IN MANOVR	
			4	BITS 4 - 15 NOT ASSIGNED	
	DIGITAL	CON		ROTATION SERVO AMPLIFIER (BDS3) RESET, ANY VALUE RESETS	
34	DIGITAL	MON		ROTATION FAULT DATA, "FAUL1" (1 = FAULT)	
			0	APEX INTERFACE NOT RESPONDING	
			1	FIRST CW LIMIT	
			2	SECOND CW LIMIT	
			3	FIRST CCW LIMIT	
			4	SECOND CCW LIMIT	
			5	COMMAND FAULT, M&C REQUEST OUT OF RANGE	
			6	MONITOR FAULT, M&C REQUEST OUT OF RANGE	
			7	OPERATOR FAULT, POSITION ARG OUT OF RANGE	
			8	BRAKE FAULT, BRAKES DID NOT RELEASE/ENGAGE PROPERLY	
			9	DRIVE FAULT, COMMANDED POSITION NOT ATTAINED, SOME MECHANICAL OR ELECTRICAL MALFUNCTION	
			10	MOTION ANALYSIS FAULT, DRAGGING OR STICKING	
			11	EMERGENCY STOP SWITCH ON	
			12	DRIVE LOCKOUT FAULT	
			13	N/A	
			14	N/A	
			15	N/A	
	DIGITAL	CON		ROTATION SOFTWARE RESET; ANY VALUE RESETS	
35	DIGITAL	MON		ROTATION SERVO AMPLIFIER FAULT DATA,"FAUL2" (1 = FAULT) (BITS 0-7 DENOTE WEST BDS3; 8-15 DENOTE EAST BDS3)	
			0/8	BUS FAULT; 300 VDC HAS DISAPPEARED	
			1/9	MOTOR TORQUE FOLDBACK; BDS3 IS LIMITING MOTOR TORQUE TO RMS-RATING BECAUSE LOAD TORQUE HAS EXCEEDED RMS RATING FOR OVER 15 SECONDS	
			2/10	OVER-TEMPERATURE, BDS3 AMPLIFIER TOO HOT	
			3/11	DRIVE DOWN, SERVO AMPLIFIER NOT READY TO RUN	
			4/12	OVER-CURRENT, MOTOR TAKING TOO MUCH CURRENT	
			5/13	POWER LOSS, SERVO AMPLIFIER POWER SUPPLY IS OUT	
			6/14	REMOTE INHIBITED, F-R CONTROLLER HAS NOT RELEASED INHIBIT	
			7/15	OVER-SPEED, MOTOR RUNNING FASTER THAN IT SHOULD	
36	DIGITAL	MON		ROTATION APEX ANALOG FAULT FLAGS, "ANAFL" (1 = FAULT)	
			0	GROUND FAULT	
			1	GROUND FAULT	
			2	GROUND FAULT	
			3	ROTATION VELOCITY	1
			4	+15	2
			5	-15	2
			6	+5	1
			7	+10	1
			8	MOUNT TEMP #1	1

			9	-10	1
			10	MOUNT TEMP #2	1
			11	BIN TEMP	1
			12	12 - 15, SPARE BITS	
37	ANALOG	MON		APEX GROUND: 0V	1
38	ANALOG	MON		APEX GROUND: 0V	1
39	ANALOG	MON		APEX GROUND: 0V	1
3A	ANALOG	MON		FOCUS VELOCITY	1
3B	ANALOG	MON		APEX +15 POWER SUPPLY	2
3C	ANALOG	MON		APEX -15 POWER SUPPLY	2
3D	ANALOG	MON		APEX +5 POWER SUPPLY	1
3E	ANALOG	MON		APEX +10 REF POWER SUPPLY	1
3F	ANALOG	MON		MOUNT TEMP 1 (100mV/DEG C)	10
40	ANALOG	MON		APEX -10 REF POWER SUPPLY	1
41	ANALOG	MON		MOUNT TEMP 2 (100mV/DEG C)	10
42	ANALOG	MON		BIN TEMP (100mV/DEG C)	10
43	DIGITAL	MON	0-7	ROTATION SOFTWARE REVISION NUMBER (BIT 7 IS MSB)	
			8-15	MODULE SERIAL NUMBER (BIT 15 IS MSB)	
44	ANALOG	MON		ROTATION VELOCITY (VEL1)	
45	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
46	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
47	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
48	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
49	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
4A	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
4B	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
4C	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
4D	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
4E	ANALOG	MON		SPARE, NOT USED, AVAILABLE	
4F	ANALOG	MON		SPARE, NOT USED, AVAILABLE	

3.10 JUNCTION BOX DESCRIPTIONS

The locations of the three junction boxes used in the F-R Control system are shown in the F-R System Cable Structure (D55007W001) and the Pedestal Room NRAO Equipment Layout drawing (C55007M012).

The junction boxes serve as distribution points for the long cable runs between the Apex and pedestal room and permit short cable runs to the motor and second screw sensor connectors on the FRM. In addition, the junction boxes contain MOV surge arrestors to protect the Apex and Pedestal Room components from damage by lightning-induced transients. MOV surge arrestors are capable of dissipating high peak energies; the signals and thresholds are listed below.

Using junction boxes for this distribution expedites the installation of the FRM and control system on the antenna because it reduces the on-site work and enables all system components to be operationally tested on the ground before installation on the antenna.

The junction boxes use modular two-tier terminal blocks. Signals are wired to the lower tier. The upper tier is a ground bus and surge arrestors are bridged between the two tiers.

As shown on the cable structure drawing, the Apex Control Junction Box distributes signals on the two 17-pair control cables to seven cables which run to the three motors and Focus second-screw sensor. Manual control switches on the junction box enable manual control of the FRM position from the Apex. An Emergency Stop switch on the junction box inhibits all power to the motors and brakes. The states of the eight limit switches are displayed on a modular LED display block inside the junction box.

The Pedestal Room Control Junction box distributes signals on the 17-pair cables to the Servo Amplifier Chassis and F-R Control Bin.

The Apex Motor Junction Box distributes motor drive power to the three motors.

The peak signal levels and MOV surge arrestor clipping levels are as follows:

SIGNAL	PEAK SIGNAL LEVEL, VOLTS	CLIPPING THRESHOLD, VOLTS
RESOLVER EXCITATION	37	45
RESOLVER STATORS	16.7	45
APEX LIMIT SWITCHES & APEX +5 VOLT CIRCUITS	5	8
BRAKE EXCITATION	90	150
MOTOR DRIVE	155	230

3.11 POWER ISOLATION TRANSFORMER BOX DESCRIPTION

The IDD servo amplifier power supply requires three-phase 208 Volt power. A Delta primary-Wye secondary isolation transformer provides isolation of the servos and protection from AC line surges. The 208 Volt AC servo power should not be applied to the servo amplifier power supply at the same time as the logic power; the motor resolver R/D converters and control logic must be allowed to stabilize before motor drive power is available. The consequence of applying both sets of power at the same time is a large uncontrolled change in shaft position which could cause damage to the FRM. For this reason the the F-R Controller firmware is programmed to turn on this 208 power a few seconds after power-on program initialization. S101 actuates (via a solid state relay in S103) a heavy duty 4-pole contactor in the transformer box. Drawing C55007S007 depicts the Isolation Transformer Box schematic diagram. Taps on the transformer secondary may be set up to step the 208 Volts up or down by 10% if the AC lines are not at the 208 Volt level. The secondary taps are accessible when the front cover is removed. The drawing shows the secondary taps set for the normal level. F-R System AC power is wired from AC terminal strips in the box. The Transformer Box Assembly Drawing is D55007A001 and the Schematic Diagram is C55007S007

3.12 SYSTEM PROTECTION FEATURES

The F-R Control System electronics and control firmware have features to protect the FRM, F-R system electronics and personnel on the antenna. This section highlights these features since they are discussed in other sections of the manual.

Features which protect personnel:

Emergency stop switch on Apex; when actuated it immediately shuts down both drives.
Focus and Rotation drives are commandable from Apex which provides the most direct control for manual adjustments and maintenance.
When the drives are commanded from the Apex, drive speeds are limited to a very slow speed which provides safer operating conditions for personnel on the Apex.

Features which protect the FRM:

FRM limit switches are redundant and actuated independently.
FRM limit switch circuits are active; if the switch is actuated or current path broken, it causes a limit switch fault. The use of an active circuit insures that switch contacts are operating properly.
Fault logic external to software causes drive lockout.
Cable Interlocks cause drive lockout if a cable is disconnected.
YOWP and Drive Lockout clear the D/A converter registers; drive to servo amplifiers is forced to zero volts.
YOWP and Drive Lockout inhibit the brake power.
Second screw motion sensor dynamically reads position of the second Focus screw relative to first screw.
Second screw tracking tolerance is 40 degrees.
Brake drive circuitry verifies both Brake voltage and current above test thresholds.
Brake torque equal to or exceeds motor torque, no drive slipping.

Features which protect the Electronics:

Lightning Protection features - surge arrestors on all lines in Apex & Pedestal Room Junction boxes.
S101-S102 signals are opto-coupled; if there is lightning damage, it is limited to S102.
Motor thermal overload relay logic senses excessive motor current, external to BDS3 fault logic
Servo amplifier protective logic senses bus faults, shorts on motor lines, excessive speed or current; causes drive shutdown.
S102 is powered by an isolated power supply which maintains ground isolation between S102 and balance of the system.
Servo amplifiers provide motor torque foldback when the torque load exceeds sustained power capablilty of the amplifiers and motors.
The R/D converters have high common-mode noise rejection.
R/D converters are Type II tracking converters with superior performance.
The switching drive to the servo amplifier power transistors is optically-coupled; there is no direct coupling to the 300 volts DC motor power.
All AC power is fused.
The 208 Volt AC servo amplifier power is transformer isolated from the antenna 208 volt AC power.
The M&C Bus is isolated from the Apex circuitry for lightning protection of other components on the M&C bus.
The control processors are never permitted to be halted.

Control program features which protect the FRM and electronics:

The servo amplifiers are inhibited when the drive is inactive between position commands.
Software limits inside hardware limits prohibit computer or manual mode drive into the hardware limits.
Reset commands clear RAM and reinitialize processors.
The NAP mode permits a limited duty use of FRM; it protects a faulty axis from damage and enables an operational axis to continue operation.
Frequent DIR/LDIR checks test for the possibility of position readout circuitry malfunctions.
Overriding commands cause the drive to be slowed to a stop before motion is started to the new command set point.
A Motion analysis subroutine senses dragging or sticking condition and causes a fault shutdown.
Frequent checks of Apex and servo amplifier faults catch faults before there is any significant motion.
Antenna Commands multiplex address and command arguments are range-checked. If out of range, the command is not executed and a fault flag is set.
Monitor data request command multiplex address is range-checked and if out of range a fault flag is set.
All analog parameters are monitored and range-checked. If out of range, a flag is set.
All discretes states are fault-checked.
No-response conditions of S102 or servo amplifiers cause a shut-down.
Focus second screw code performs a close comparison of the first and second screw position; an out of tolerance condition causes a rapid drive shutdown. If the second screw failure is repeated, the Focus drive is shut down until a reset command is issued.
All mode & fault states are reported to the Antenna Control Computer via Monitor data.
Control states are reported to the antenna control computer during command execution. Command error is read out in the monitor data. Command error and drive velocities are available in the Monitor data.
Monitor data is always available to the antenna control computer.
The high sample rate in TMR2 motion control program permits a quick detection of dragging or sticking drive fault conditions.
Near the command set point, the drives are ramped down and moved to the command set point at low speed.
There are no abrupt changes in drive positions or velocity (except for Second screw faults which causes a rapid focus shutdown).
The VLBA Standard Interface Board maintains records of parity, etc. faults which can be read out as monitor data.

3.13 TABLE OF INTERFACE SIGNALS, LEVELS AND FUNCTIONS

The following table lists the F-R System interface signals.

S101 I/O

S104 INTD bus - from S104, INTD0..INTD7, 8 TTL, high-true & return
S102 Apex Foc & Rot Data Request - to S102, 1 each (per axis) TTL pair, low-true
S102 Apex Foc & Rot Serial Apex Data - from S102, 1 each (per axis) TTL pair, low-true to opto rcvr
S102 Apex Foc & Rot Serial Data Clock - from S102, 1 each TTL pair (per axis), low-true to opto rcvr
Logic drive to S105 LED displays - to S105, 6 each Foc & Rot, high-true TTL lines
Logic drive from S105 slew switches - from S105, 2 each, (per axis) low-true TTL lines
Bus enables from S101 - to S104 - Foc: F395, F396, F397, F398, F399, F400, F407,
- Rot: R395, R396, R397, R398, R399, R400, R407, R408
All TTL low-true
Foc BDS3 data bus from S104 - 8 TTL low-true fault/status lines via FBDFX bus to FBAD bus & return
Rot BDS3 data bus from S104 - 16 TTL low-true fault/status lines via RBDFX bus to RBAD bus & return
Foc & Rot Data request & DMA request from S104, 2 each (per axis) high-true TTL & 1 return
Focus & Rotation Enable - from S104, 1 each (per axis) high-true TTL line
Focus & Rotation Analog drive to BDS3's - to Servo Amp Chassis - 1 each (per axis) +/- 2.6 volt drive & return
Focus & Rotation BDS3 Inhibit & Reset - to Servo Amp Chassis, 2 each (per axis) low-true TTL & 1 return
Focus & Rotation Brake Command - to S103, 2 each low-true TTL lines
Emergency Stop Switch Input - from Servo Amp Chassis, 1 each TTL line & return
System Master Reset - from S014, low-true TTL, 1 line
3-Phase Relay control - to S103, low-true TTL, 1 line & return
Focus & Rotation Motor Velocity - from Servo Amp Chassis, 1 each, +/- 8 volts & return
Focus & Rotation Motor Current - from Servo Amp Chassis, Foc: 4.6 volts = 20 RMS amps/phase; Rot: 4.6 volts = 12 RMS amps/phase
System +5, Common, +/- 15 volts & Analog Common - from S105

S102 I/O

Resolver excitation - from S103, 1 pair, 26 volts, RMS, 400 Hz
Focus & Rotation resolver stator signals - from Apex, 2 pairs (per axis), (S0, S1, S2, S3 & S4), 11.8 volts RMS, 400 Hz
Focus & Rotation Limit Switch circuits - from Apex, 8 lines & return, low-true TTL, active closed circuit, open when switch actuated
S101 Foc & Rot Apex Data request - from S101, 1 each (per axis) low-true TTL & return, to opto rcvr
S102 Foc & Rot Apex Serial data - to S101, 1 each (per axis) low-true TTL & return, to opto rcvr
S102 Foc & Rot Apex Data Clock - to S101, 1 each (per axis) low-true TTL & return, to opto rcvr
YOWP! - to S101, 1 each low-true TTL & return, to opto rcvr
+5, Common, +/- 15 volt & Analog Common Isolated power - from S105

S103 I/O

Resolver excitation - to S102 & Apex, 1 pair, 26 volts, RMS, 400 Hz
110 volts AC drive to 3-phase contactor, 1 pair
Foc & Rot DC brake drive - to Apex, 90 volts DC, 1 pair (per axis)
FRM Heater 110 VAC - to Apex, not used
FRM Heater SSR drive - from FRM heater controller, not used
Focus Brake SSR drive - from S101
Focus Brake DC drive - to Focus Brake on FRM
Rotation Brake SSR drive - from S101
Rotation Brakes DC drive - to Rotation Brakes on FRM
3-Phase SSR drive - from S101
3-Phase Relay 110 V AC drive - to 3-Phase relay in Isolation Transformer Box
System +5 & Common - from S105

S104 I/O

Monitor & Control Bus - from Antenna Control Computer, 2 pairs, RS 485
INTD bus - to S101, INTD0..INTD7, 8 TTL, high-true & return
S104 bus enables - from S101, Foc: F395, F396, F397, F398, F399, F400, F407
Rot: R395, R396, R397, R398, R399, R400, R407, R408
All TTL low-true
Foc & Rot Data & DMA requests - to S101, 2 each (per axis) high-true & 1 return

Foc & Rot Enable - to S101, 1 each (per axis), high-true, TTL
System Master Reset - to S101, low-true TTL
Focus BDS3 Signal Outputs - to S101, 8 low-true TTL, fault-status signals to FBADX bus via FBDFX bus
Focus BDS3 Signal Inputs - from Servo Amp Chassis, 8 high-true TTL, fault-status signals
Rotation BDS3 Signal Outputs - to S101, 16 low-true TTL, fault-status signals to RBADX bus via RBFDX bus
Rotation BDS3 Signal Inputs - from Servo Amp Chassis, 16 high-true TTL, fault-status signals
Focus BDS3 +/- 12 volt power monitor - from Servo Amp Chassis, 2 signals & return, +/- 12 volts dc
Rotation BDS3 +/- 12 volt power monitor - from Servo Amp Chassis, 4 signals & 2 returns, +/- 12 volts dc
RDP2 +/- 12 volt power monitor - from Servo Amp Chassis, 2 signals & 1 return, +/- 12 volts dc
System +5, Common, +/- 15 & Analog Common - from S105

S105 I/O

+5 & Common System logic power
+/- 15 volt & Analog Common system analog power
+5 & Common, +/- 15 & Analog Common - to S102 & Apex (Apex: +5 & common only)
Foc & Rot Slew switch inputs - from S101, 2 each (per axis), low-true TTL lines
S105 display led drive from S101
Foc & Rot Display LED drive - from S101, 6 each (per axis), high-true TTL lines
S103 110 VAC - to S103
FRM 110 VAC Heater power - to Apex (not used)
400 Hz Excitor 110 VAC power - to S103
Bin Fans 110 VAC - to bin fans (not used)
Foc & Rot Brake 110 VAC - to S103
F-R Control System Power - from Isolation Transformer Chassis, AC hot, neutral & ground

SERVO AMPLIFIER CHASSIS I/O

IDD Resolver Excitation - to Apex junction box, 26 volts RMS, 2600 Hz ref high & low, 1 pair/motor, 3 motors
IDD Resolver Stator Signals - from Apex, 11.8 volts RMS, Sin high & low, Cos high & low, 2600 Hz, 2 pairs/motor, 3 motors
Foc & Rot Analog Drive - from S101, +/- 2.6 volts, 1 signal & return (per axis)
Focus & Rotation Motor Velocity - to S101, +/- 8 volts, 1 signal & return (per axis)
Focus & Rotation Motor Current - to S101; Foc: 4.6 volts = 20 RMS amps/phase; Rot: 4.6 volts = 12 RMS amps/phase
Focus BDS3 fault/status - to S104, 8 high-true TTL & return
Rotation BDS3 fault/status - to S104, 16 high-true TTL & return
Foc & Rot BDS3 Inhibits & Resets - from S101, 2 each (per axis) TTL low-true & return
Foc BDS3 +/- 12 volt Power Monitor - to S104, 2 each & return, +/- 12 volts dc
Rot BDS3 +/- 12 volt Power Monitor - to S104, 2 each & returns, +/- 12 volts dc
Focus Motor Drive - to FRM, 3-phase, 208 VAC, 4 KHZ chopped drive to motor, 3 lines & return
Rotation Motor Drive - to FRM, 2 each sets, 208 VAC, 4 KHz chopped drive to motor, 3 lines & return
3-Phase, 208 VAC - from Isolation transformer Box transformer
110 VAC Control Power - from Isolation Transformer Box to power PSR3, BDS3's & RDP2

ISOLATION TRANSFORMER BOX I/O

3-Phase 208 Volt AC F-R input power - to contactor 3 phases, neutral & ground
3-Phase 208 Volt AC motor power - to Servo Amp Chassis, 3 wires & neutral
110 Volt AC Control Power - to Servo Amp Chassis PSR3, BDS3's and RDP2, AC hot, neutral & ground
110 Volt AC Contactor Excitation - from S103, AC hot, neutral & ground
110 Volt AC F-R Control System Power - to S105, AC hot, neutral & ground

3.14 ALIGNMENT OF RESOLVERS, LIMIT SWITCHES, S101 AND S102 MODULES

Resolver and Limit Switch Alignment

This section describes the alignment of the FRM position readout transducers, the S101 A/D and D/A converters and the S102 A/D converters.

The adjustment of the position readout transducers will be described first. For access to the limit switch and resolver adjustments, the cable connectors must be disconnected and the two end screws removed. The cover can then be slid off length-wise. When removing the cover note the "O" rings around the base plate, connector shells and under the heads of the end screws. Do not lose the screws or "O" rings; they protect the transducer from water entry, a deadly enemy of electronics.

The Focus and Rotation position readout transducers are mounted on the end-bells of the three drive motors with servo clamps. The transducer shaft is attached to the motor shaft by a flexible coupling. When these clamps are loosened, the transducer case may be rotated on the motor end bell; this is equivalent to rotating the motor shaft and permits a final adjustment of the readout values.

The position readout resolvers and limit switches must be adjusted on the ground before installation on the FRM. The motor brakes must be released to permit the motor shaft to be rotated. Up to 32 turns of the Rotation motor shaft may be required in aligning the rotation resolver and limit switches. The Focus motor shaft may have to be rotated up to 145 turns in aligning the Focus resolver and limit switches. The position readout transducers are manufactured by the Micron instrument corporation; data sheets for the two types of transducers are included in Volume II of this manual. Drawings C55007A029 (Focus) and C55007A030 (Rotation) depict a section view of the transducers in the resolvers and switch planes. Note that the coarse resolvers are mounted on cylindrical bosses (actually reduction gear boxes) and the fine resolvers are mounted on the base plates. The drawings also illustrate the notch relationships of the switch-actuating cams and identify the switches and switch-adjusting set screws. Tables on the lower right show the switch actuation point in terms of position readout.

When the transducer input shaft is rotated, the limit switch cams and resolver shafts are rotated by independent gear trains; the switch and resolver adjustments do not interact. The resolvers are driven by anti-backlash gear trains and the resolver body is secured to the mounting base plate by servo clamps which must be loosened to permit the resolver case to be rotated in the adjustments.

The limit switches are actuated by two notched cams driven at a slightly different angular rate when the input shaft is rotated. The cams are mounted on two rotating drums driven by differential gearing from the input shaft. When the notches coincide under a switch roller, the switch is actuated. The switching transitions are alternate action; that is, when the input shaft continues to rotate in the same direction through the position where the coincident notches actuate the switch, the switch contacts remain in the switched state until the shaft rotation is reversed and run back through the coincident notches. In the reversed direction, when the coincident notches actuate the switch, it will return to the original state.

The Rotation transducer uses two switches and the Focus transducer uses four switches. To adjust the switches, the cams are loosened on the drums by releasing spline-wrench set screws in the cam hubs. (The screw locations on the hubs are shown on the lower left view of the two drawings.) With the set screws loose, the cams may be rotated so that the cam notches coincide under a switch roller.

The S102 discretes LED display shows the switch states. An energized switch LED indicates that a limit switch has been actuated as the drive approaches an extreme of the drive range. See Section 3.2

for the locations of the limit switch LED's on this display.

We will first describe the Focus resolver and switch alignment. This is done in two stages: the first stage is done on the lab bench and the second stage is done on the antenna after the Focus motor has been installed.

On the bench, the Focus brake should be disengaged (i.e. energized) and the motor shaft rotated until a small S102 Focus readout value (typically about 50 to 100 counts) is obtained. Only the Focus coarse resolver is used for position readout; the fine resolver is not used. Having set this value, the four limit switch actuation points are adjusted next. It may be necessary to rotate the motor shaft to position the cam hubs so that the set screws are accessible for the spline wrench. Loosen all hub set screws. Rotate the Focus motor shaft so that the Focus position readout values increase and then stop at the value 0600H; this is the 2nd lower limit switching position. (See drawing C55007A030 to identify the switches, cams and switching point settings.) Slip the 2nd lower limit cams until both cam notches coincide under the switch roller and lightly tighten the set screws. Verify that the switch actuates at this readout value point by rotating the motor shaft back and forth through the switching point while observing the S102 Focus position display and the LL 2 discrete LED. Gently tighten the set screws and again rotate the shaft back and forth through the switching point to verify that the setting is still correct. Finally, tighten both sets of hub set screws. Continue this procedure for the LL 1 switch, the UL 1 switch and finally the UL 2 switch. These adjustments will require about 143 revolutions of the Focus motor shaft. As a final check, rotate the motor shaft through the 145 turn range and verify the switch actuation. Record the readings; they could be useful at a later date. To complete the bench alignment, rotate the Focus motor shaft to a readout position of 8000H; this is the half-range position.

On the antenna, after the FRM and F-R system are installed and the FRM may be driven, the second state of Focus alignment may be done. The motor connector should be pointing up (see the 3105 motor data sheet in Volume II). The traveling platform (i.e. middle ring) should be carefully positioned mid-range and verified by careful measurements with a steel tape. Useful reference points are the metal edges of the bumpers, adjacent to the rubber pads. Measure the distance between these two reference points to the top and bottom rings; the distances should be equal. If equal, the middle ring should be equally movable in either direction. If not in mid-position, the platform should be manually positioned to make it so. At this point the motor is installed which might require some shaft rotation to align the motor and coupling keyways; try to rotate the motor shaft through the smallest angle when installing the motor. Remember that the motor shaft was positioned to produce a position readout of 8000H. If after motor installation, the readout value is not 8000H, loosen the transducer mounting servo mount clamps and rotate the transducer case to obtain this value on the S102 Focus display. It should not be necessary to rotate the transducer case more than one-half turn to reach this value.

The Rotation transducer alignment is more complicated since position readout is derived by combining positions from 32:1 ratio, coarse and fine resolvers. The fine resolver cycles through its conversion range 32 times for one rotation of the coarse resolver range. One may in effect consider that the fine resolver range is "wrapped" around the coarse resolver range 32 times. Rotation drive position is read only from the West Rotation drive motor transducer but the two limit switches in each transducer are used for first and second limits. The West transducer is used for first limit switches and the East transducer is used for the second limits.

Inside the S102 front panel access cover is a 3-position, momentary-on, on, momentary-on toggle switch which selects one of three Rotation R/D position data sources for display. The Rotation Fine R/D converter data is selected in the momentary up position; the composite Rotation position from the combiner is selected in the center position and the Rotation Coarse R/D converter data is selected in the momentary down position. The switch is set to each of these two momentary positions while adjusting the West coarse and fine resolvers. The reader should refer to the S102 description and the display logic.

The circular Rotation motor mounting bases are held by eccentric clamps which, when loosened, permit adjustment of the motor pinion gear-ring gear mesh. In performing this adjustment, the motor case is rotated as required for the best gear mesh; thus there is no fixed mounting reference for the Rotation motors as there is for Focus.

Like the Focus motor, the Rotation brakes must be energized to align the resolvers and switches.

Like Focus, the Rotation transducers are aligned in two stages: first alignment is done on the bench and a final alignment is done on the antenna.

In the bench alignment, the coarse resolver zero is set to the approximate mid-point of one the fine resolver ranges; this is the optimum set point for minimum sensitivity to coarse resolver non-linearities. In this case the coarse-fine ratio is relatively small and the resolvers are linear within +/- 3 minutes. The reader should refer to the Natel R/D Converter and Combiner data sheets in Volume II. Although only the West motor transducer is used for position readout, the east transducer resolvers are also aligned so that in the event of a failure of the west resolvers, the motor packages may be interchanged to put the antenna back into service. In this case the limit switches must be readjusted on the antenna.

While observing the position display, carefully tighten the servo clamps to secure the resolver's cases to the base plate and cylindrical boss. The readout value may shift as the clamps are tightened; several adjustment-tightening cycles may be required to obtain the correct zero settings. The resolver alignment should be done very carefully because the resolver wires are very small and easily broken.

Using the techniques described above in the Focus alignment description, after the resolvers are aligned, the two limit switches in each transducer should be adjusted for the switching points specified on the drawing. After setting up the limit switches, rotate the motor shafts to produce a position readout value of 0000H counts.

On the antenna, one subreflector mounting bolt hole is visually positioned adjacent to the East motor; this is the anointed reference bolt hole which is marked by a felt tip marker. This angular position is defined as full CCW. The ring gear should not be rotated after marking this hole until completion of the Rotation alignment. Corresponding holes in the subreflector mounting tube, spider and subreflector assume this reference angle. After the motors have been installed and the gear mesh set up, the servo mount clamps are loosed and the transducer cases are rotated to produce a position readout of 0000H. The West readout cable should be temporarily connected to the East transducer to adjust the East transducer to a 000H readout since the east transducer resolvers are not connected to the S102 converters. Don't forget to return the two cables to the proper transducers. When the motors are installed, be sure that they are not interchanged; this would confuse the limit switch logic in S102 and the control firmware in S101.

S101 Alignment

The S101 A/D converter is aligned in the LOCAL mode (i.e. with the CMP-MAN switch set to MAN) to enable the Mode Switch commands to cause the analog multiplexers and A/D converters to sample and convert the reference voltages used to adjust the A/D converter zero and gain potentiometers. The converted values are displayed on the S101 Focus and Rotation displays. The 12-bit A/D converters are scaled at 5 mv/count and the converter code is 2's complement; thus plus full scale code output is produced with a 10.235 volt input and minus full scale code output is produced with a minus 10.240 volt input.

With an accurate DMM, check the AD2702 +10/-10 volt reference chip CA44 +10 volt and -10 volt outputs; they should be within +/- 2.5 millivolts of these values.

The A/D converter is a Harris HI-574 which, after adjustments, should be accurate to within about +/- 5 millivolts over the conversion range.

With the Mode Switch set to 06 which selects analog ground for multiplexer inputs, potentiometers EG29-5/6/11 (Focus) and CG29-5/6/11 (Rotation) are adjusted to produce a count of 00.000 (decimal).

With the Mode Switch set to 07 which selects + 10 volts for conversion, adjust potentiometers EG29-7/8/9 (Focus) and CG29-7/8/9 (Rotation) to produce + 10.000 on the displays.

With the Mode Switch set to 05 which selects - 10 volts for conversion, check that the displays display - 10.000. If they are slightly off this value, repeat the previous step to equalize the conversion errors for the two 10 volt inputs.

Repeat the zero volts test to insure that the zero volts (Mode Switch = 06) conversion is still zero; re-adjust the zero potentiometer if necessary. In any case, the zero adjustment is the most important adjustment of the zero/gain settings.

The AD565A D/A converters are scaled at 2.5 mv/count and should be accurate within +/- 2.5 millivolts over the conversion range.

Set the Mode Switch to 03 which causes the control microprocessors to output a D/A converter zero code to the D/A converters. With the accurate DMM connected to EE19-6 (Focus) and CE19-6 (Rotation), adjust potentiometers EG29-1/2/15 (Focus) and CG19-1/2/15 (Rotation) to produce a zero volts D/A output.

Set the Mode Switch to 04 which causes the control microprocessors to output a + 5.000 code value; this should produce a + 5.000 D/A output. If not this value, adjust potentiometers EG29-3/4/13 to produce this D/A output.

Set the Mode Switch to 02 which causes the control microprocessors to output a - 5.000 code value; this should produce a - 5.000 D/A output. If not this value, repeat the zero volts and + 5.000 tests to equally distribute the error between the +5 and -5 codes. In any case the zero volts adjustment is the most important of the zero/gain adjustments.

S102 Alignment

To align the S102 it is necessary to view the Apex Interface data on a computer terminal. The S102 uses the AD2702 reference and HI574A A/D converter chips as the S101. First check the AD2702 + 10 and - 10 outputs to determine the AD2702 accuracy.

Open the front panel access cover and adjust potentiometer R6 for a zero volts value on analog ground address 17H. Verify the same value on addresses 18H, 19H (Focus) and 37H,38H and 39H (Rotation). Set the momentary toggle switch S2 to the +10 (up) an -10 (down) positions and adjust potentiometer R6 to produce + 10 and - 10 volt outputs on address 17H. Balance the error between the +10 and -10 settings if necessary. Again, the most important adjustment is the zero adjustment. With switch S2 in the center position verify that addresses 1EH and 3E show a + 10 volts value and addresses 20H and 40H show a - 10 volts value.

Logic Analyzer Connections

A logic analyzer is a convenient tool for program development and can be useful for debugging subtle logic problems. The analyzer connections to the S101 digital logic should be made in ascending order, just as addresses and data are ordered in the listings. To follow code execution, sixteen address bits: FADD0, ... FADD7; and A8, ... A15 provide sufficient range for address tracking. The eight data bits of the Buffered Address-Data bus, BAD0,... BAD7 show data flow. The analyzer should be clocked by the rising edge of logic term FRDbar + FWRbar which is available on gate FC43-08, Sheet 2 of the logic schematics.

To monitor the data input from the Apex Interface, the logic analyzer may be connected to the Apex Interface data registers, B24, B19, ... B25 and clocked by the rising edge of gate B18-03.

4.0 SYSTEM SCALING PARAMETERS AND LIMIT VALUES

This section summarizes important scaling parameters which are distributed through the hardware and firmware parts of the manual.

Calculation resolution: 16 bit unsigned word format, range 0 to 65,535; 16 bit signed word format, range - 32,768 to + 32,767

Focus position resolution: 1 part in 16384 counts (14-bit readout format)
Focus physical resolution: 0.0017 inches/count (14-bit format)
Focus screw travel: 143.51 revolutions, 28.25 inches, hard-stop to hard-stop
Focus readout gear ratio: 145:1
Focus readout counts/screw revolution: 451.97 (16-bit format); 112.99 counts (14-bit format)
Focus readout range: in 14-bit readout format 0 to 16,216; in processor 16 bit format - 64,864 counts
Focus maximum second screw tracking error: 40 degrees, 50 counts (16-bit format), position to screw

Rotation position resolution: 1 part in 65536
Rotation physical resolution: 0.38452 arc-min/count
Rotation drive range: 420 degrees, 25200 arc-minutes
Rotation readout gear ratio: 32:1 (coarse:fine)
Rotation fine resolver readout counts/motor revolution: 2048
Rotation position readout range: 65,536 counts
Rotation motor pinion: ring gear ratio: 32:1

Limit switch settings:	first UP/CW	F500H	second UP/CW	F600H
	first DOWN/CCW	0700H	second DOWN/CCW	0600H

Soft limits: inside hard limits, 0A00H = LOW, F200H = HIGH

Readouts at physical extremes (hard-stop to hard-stop) of Focus drive: 0000H to FD60H for 0 to 28.25 inches
Readouts at physical extremes of Rotation drive: 0000H to FFFFH for 0 to 420 degrees

Drive DAC acceleration (both drives): 1 Volt/sec^2; 400 counts/sec^2

TMR2 Interrupt rate = 1800 Hz

Focus servo velocity scaling = 3.020 rev per sec/volt drive
Focus position (16-bit) counts/rev = 451.97

Focus drive speeds, DAC output, volts:	Max =	2.8	in MAINNY;	0.070	in VERGIT;	0.687	in Manual
Focus drive speeds, rev/second:	Max =	8.456	in MAINNY;	0.2114	in VERGIT;	2.076	in Manual
Focus drive speeds, counts/sec:	Max =	3821.9	in MAINNY;	95.54	in VERGIT;	938.4	in Manual
Focus (16-bit) position states/TMR2 cycle:	Max =	2.123	in MAINNY;	0.054	in VERGIT;	0.521	in Manual
Focus TMR2 samples/position state:	Min =	0.471	in MAINNY;	18.84	in VERGIT;	1.918	in Manual

Rotation servo velocity scaling = 13.04 rev per sec/volt drive
Rotation position counts/rev = 2048

Rotation drive speeds, DAC output, volts:	Max =	2.8	in MAINNY;	0.070	in VERGIT;	0.687	in Manual
Rotation drive speeds, rev/second:	Max =	36.5	in MAINNY;	0.9128	in VERGIT;	8.965	in Manual
Rotation drive speeds, counts/second:	Max =	74,776	in MAINNY;	1869	in VERGIT;	18,360	in Manual
Rotation position states/TMR2 cycle:	Max =	41.54	in MAINNY;	1.038	in VERGIT;	10.20	in Manual
Rotation TMR2 samples/position state:	Min =	0.024	in MAINNY;	0.962	in VERGIT;	0.098	in Manual

Focus Breakpoints:	Computer	BREAK1 = 7% of commanded motion	BREAK2 = 90% of commanded motion
	Manual	BREAK1 = 65 counts from start pos	BREAK2 = commanded motion - 65 counts
Rotation Breakpoints:	Computer	BREAK1 = 12% of commanded motion	BREAK2 = 40% if ERROR < 1000 counts
			" = 60% if ERROR < 3000 counts
			" = 80% if ERROR >= 3000 counts
	Manual	BREAK1 = 65 counts from start pos	BREAK2 = HIGH or LOW - 65 counts

Focus lifting load torque load at a temperature of about 70 deg fahrenheit: about 2.6 pound-feet
Rotation max velocity torque load at a temperature of about 70 degrees Fahrenheit: about 1 pound-foot.

5.0 SUBREFLECTOR POSITION ERROR EFFECTS

This section describes the effects of subreflector Focus and Rotation position errors on the antenna performance. These may be separated into two groups: subreflector positioning errors and physical alignment errors.

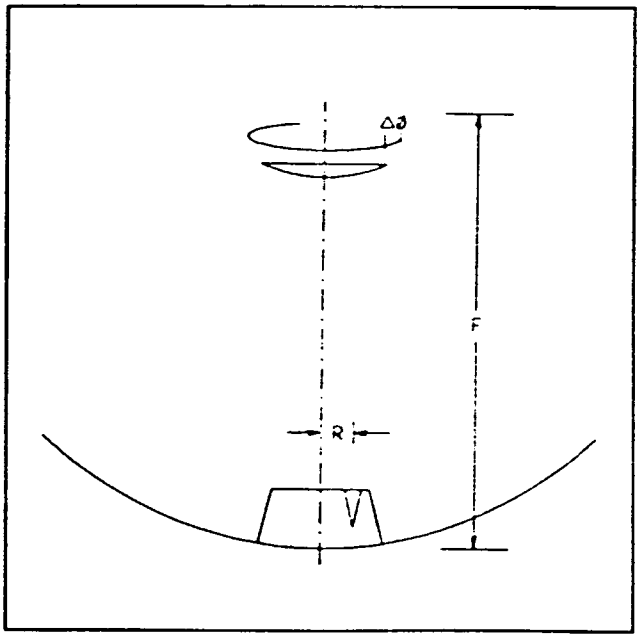
We will first describe subreflector positioning errors resulting from imperfections in the control system performance. Repeatable positioning errors such as non-linearity, etc. do not influence antenna pointing as long as the subreflector positioning is repeatable, smooth and continuous over the working range of motion. The effects of repeatable errors are canceled since a single pair of set point values are used for each band position. Non-repeatable errors are the primary concern in positioning the subreflector; the effects of these errors are discussed below.

Since the feed circle is offset from the axis of the primary reflecting surface, non-repeatable errors in Rotation position cause errors in antenna pointing. Non-repeatable errors in Focus position degrade the signal strength in the feed horns and cause variations in the signal path length between the reflective surfaces.

The antenna pointing error resulting from a Rotation position error is: $\Delta\Theta = \Delta\emptyset R/MF$ where $\Delta\Theta$ is the pointing error on the sky and $\Delta\emptyset$ is the subreflector Rotation positioning error. M is the Magnification factor, R is the feed circle radius and F is the focal length of the primary reflector. For the VLBA antenna, M is 7, R is 0.853 meters and F is 9 meters. For the VLBA antenna, a Rotation position error of 1 arc-minute causes a pointing error of 0.813 arc-seconds. The figure below illustrates the Rotation pointing error relationships. The repeatability of the Rotation positioning is better than +/- 4 counts or +/- 1.54 arc-minutes.

The effect of non-repeatable Focus position errors is inversely proportional to wavelength. Non-repeatable Focus position errors should be less than 1/16 of a wavelength at the shortest operating wavelength. At 86 GHz (wavelength = 3.5 mm), non-repeatable Focus position errors should be less than 0.009 inches. The repeatability of the Focus positioning system is better than +/- 4 counts which is +/- 0.007 inches.

FRM and subreflector physical alignment is a complicated sequence of positioning and tilt adjustments. The reader should refer to the FRM description in Section 1 and Figures 1 and 2 during the following description of physical alignment.



When the FRM, subreflector mounting tube, spider and subreflector are installed on the VLBA antenna, the FRM and associated components are carefully aligned by a series of adjustments. The FRM is installed first and a precision-machined alignment bar is bolted to the subreflector tube mounting flange. The alignment bar has an optical mirror target mounted in the center of the alignment bar and the mirror plane is exactly parallel with the subreflector mounting flange plane. The target mirror has cross hairs which are positioned exactly on the center of

the alignment bar so that as the Rotation drive is moved, the cross hairs remain centered on the center of rotation of the Rotation drive.

A theodolite mounted on the elevation axis tube below the Vertex Room is oriented to look up at the mirror target on the alignment bar. The theodolite position is adjusted to look exactly along the axis of the primary reflecting surface.

The FRM mounting feet and mounting plates on the Apex ring are slotted so that the FRM can be moved around on the mounting bolts; the theodolite looking at the cross hairs on the alignment bar target is used to determine the correct FRM position. Using the theodolite as an autocollimator, the Rotation drive is moved to measure the tilt of the FRM with respect to the axis of the primary reflecting surface (most of the tilt is probably in the Apex ring). Shims are placed under the FRM mounting feet to eliminate the tilt.

The alignment bar is removed and the subreflector mounting tube, spider and subreflector are installed. The subreflector also has a cross hair mirror target installed in the center of the reflecting surface. The plane of the mirror is normal to the axis of the reflecting surface. The subreflector is adjusted so that its reflective center and the center of rotation of the FRM coincide. As the Rotation drive is moved, the theodolite, operating as an autocollimator, is used to measure the tilt of the subreflector. Shims, placed between the spider and subreflector are used to eliminate the tilt. As a final check, the Focus drive is moved to verify that the Focus motion is parallel to the axis of the primary reflecting surface.

These adjustments enable the subreflector position to be aligned to the primary reflecting surface with an accuracy of a few thousandths of an inch.

6.0 TELESCOPE OPERATOR INFORMATION

This section describes the F-R Overlay Screens which present F-R system status and data to the VLBA telescope operator. Although the information presented is available in other sections of this manual, it is summarized here to provide a convenient reference for telescope operators.

The F-R System Operator's Overlays consist of two screens: the Control-Status and Power Supply Status screens. The Control-Status screen is described first since it is the primary operator interface. The Power Supply screen presents the status of the system power supplies.

The Control-Status screen shows operating modes, commanded position, actual position, position error and fault flags. The telescope operator may issue position, mode and reset commands via this screen. This screen is more frequently used than the Power Supply screen because of the comprehensive displays and operator command facilities.

The Control-Status screen format is depicted on the next page; the state of both drive axes are displayed. The system mode is shown at the top and Focus and Rotation position data is shown immediately below on the left and right, respectively. The position data format is four-digit unsigned hexadecimal and the normal values range from 0A00H to F200H. Below the position data are the nap mode and reset command labels. Below these labels are the Focus and Rotation fault flags, present on the screen when a fault is sensed. Logic in the F-R Controller protects the expensive and delicate FRM in the event of a system malfunction.

Commanded position indicates the position set point commanded by the observing program or telescope operator via the antenna control computer. Position indicates the actual position of the two drive mechanisms and is reported by the F-R Controller. Position error is the absolute value of the difference between the commanded and actual position and is reported by the F-R Controller. During the execution of a position command, the operator may observe the error value diminish to zero or a very small value.

Bold-printed labels on the Control-Status screen illustration are F-R system modes and fault flags. Underlined, bold-printed labels are operator command entry labels. To initiate a command the operator should position the cursor under a command entry label and press the keyboard ENTER key to activate the selected command. Operator command intervention is possible only when permitted by the Antenna Computer operating system.

The COMPUTER mode label indicates that the Antenna Control Computer is controlling the F-R System, either through the observing system or through the operator's intervention. Manual mode indicates that the F-R Controller has been put into the manual mode by setting the F-R Controller front panel MANUAL-COMPUTER switch to the MAN position. Computer control (the normal mode) is enabled by setting the switch to the CMP position. The manual mode is used for local system and maintenance checks.

NAP mode commands permit the Telescope Operator to temporarily shut down the designated drive so that it does not execute position commands. This command literally commands the designated controller to "take a nap". This mode permits limited use of the F-R System in the event of a malfunction. An example of the use of this command is to shut down the Focus drive in the event that cold weather sticking is impeding the Focus drive motion. In this case there is some degradation in the intensity of the received signal since it is not properly focused in the feed horns but the NAP mode may permit observations to continue until the fault condition is corrected. The NAP mode is rescinded by the SOFTWARE RESET or MASTER CLEAR command.

The MASTER CLEAR command causes a power reset and reinitialization of the F-R Controller microprocessors; it also clears the NAP mode. All fault flags are cleared by MASTER CLEAR.

The SOFTWARE RESET command causes a processor reinitialization and clears all faults.

The EMERGENCY STOP mode is used to disable all Focus and Rotation drive motion and is set by an EMERGENCY STOP SWITCH at the antenna apex.

When the F-R controller receives a new position command, it clears the fault flags stored in its RAM memory and starts the execution of the new command. In the event that a fault state persists, the controller will terminate the command execution and will report the fault state to the Antenna Control computer via the monitor data. The telescope operator should not attempt to clear persistent fault conditions by repeatedly attempting position commands.

```
                    FOCUS/ROTATION

MODE

  COMPUTER                EMERGENCY STOP
  MANUAL

              __ FOCUS                __ ROTATION
COMMAND         XXXX                    XXXX
POSITION        XXXX                    XXXX
ERROR           XXXX                    XXXX

SOFTWARE RESET              FOC/ROT

MASTER/CLEAR          NAP   FOC/ROT

                LL  1  2                CWL  1  2
                UL  1  2                CCWL 1  2
                APEX FAULT              APEX FAULT
                BRAKE FAULT             BRAKE FAULT
                DRIVE FAULT             DRIVE FAULT
                2ND SCREW FAULT
                MOTION ANALYSIS FAULT   MOTION ANALYSIS FAULT
                DRIVE FAULT             DRIVE FAULT
```

COMMAND-STATUS SCREEN

The LL 1, UL 2, etc. fault flags indicate the actuation of FRM limit switches. Mechanically-actuated LIMIT SWITCHES sense drive position at the extremes of the drive range. The first limit switches (LL 1, UL 1, CW 1 and CCWL 1) are actuated first; additional travel toward the end of the range actuates the second limit switches (LL 2, UL 2, CW 2 and CCWL 2) which serve as redundant backups to the first limit switches. When any of these switches is actuated, the servo amplifier drives are inhibited; no further motion in any direction is possible. Under normal conditions, it is not possible to drive mechanisms into the limit switches in either computer or manual modes; the F-R Controller will not permit drive outside software limit bounds (0A00H and F200H) inside these hardware limits. Actuation of any of these limit switches indicates a potential malfunction; the telescope operator should not attempt any control action to drive out of the limit conditions. An F-R Controller internal switch permits drive into and out of the limits; this switch should only be operated by individuals knowledgeable about the F-R Control System.

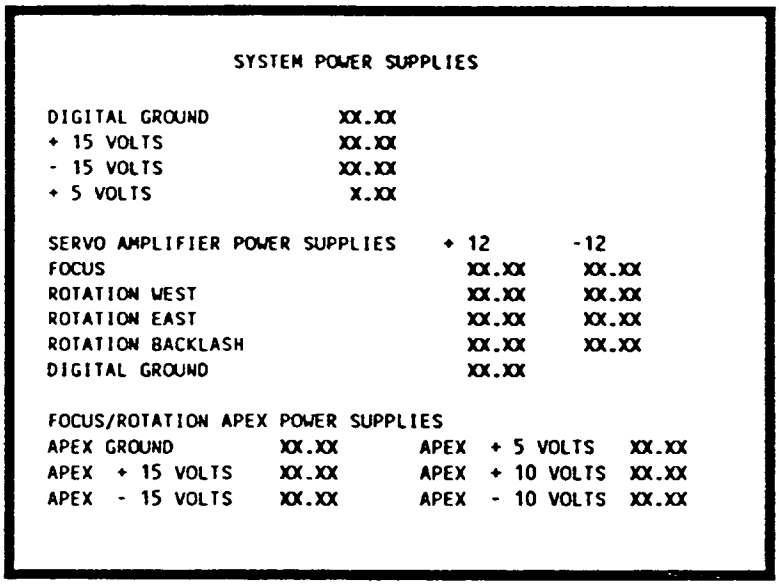
The APEX FAULT flag indicates that the Apex Interface module is not supplying position and fault status data when requested by the F-R Controller module. This is a serious malfunction; if it persists, the telescope operator should not attempt any further F-R control action.

The BRAKE FAULT flag indicates that the brake that locks the drive is either not disengaging or engaging properly. This is a serious malfunction; if it persists, the telescope operator should not attempt any further F-R control actions.

The DRIVE FAULT flag indicates a malfunction in the drive; if it persists, the telescope operator should not attempt any further F-R control actions.

The SECOND SCREW FAULT flag is a Focus (only) fault condition in which the two Focus Drive positioning leadscrews are not moving in unison. This is a serious malfunction; if it persists, the telescope operator should not attempt any further F-R control actions. The SECOND SCREW FAULT is cleared with a MASTER CLEAR or SOFTWARE RESET.

The MOTION ANALYSIS FAULT flag indicates that the drive is not moving at the commanded rate; some mechanical binding or cold temperature lubricant viscous friction may be impeding the motion. This condition is most probable at cold temperatures with the Focus drive; it is less probable in the Rotation drive. The servo amplifiers have torque foldback features which protect the FRM from damage in the event of mechanical sticking or excessive viscous friction. In the event that the problem is cold temperature related, pointing the dish at the sun may heat the Apex sufficiently to permit the drive to be operated.



POWER SUPPLY STATUS SCREEN

The DRIVE LOCKOUT flag indicates that all brake and drive motion is inhibited by digital logic external to the firmware in the F-R Controller. When this fault is active (i.e. true), the F-R Controller firmware will sense and flag this condition to the F-R Command-Status overlay but will not execute motion commands. This flag is set in conjunction with limit faults or when a disconnected cable is sensed. This flag indicates a serious malfunction; the telescope operator should not attempt any further F-R control actions.

The Power Supply screen displays the F-R system power supply voltages. This screen is used to monitor the state of these supplies in the event of a fault; the fault may be the result of a power supply failure. Data values are decimal. The monitor data for power supply voltages above 10 volts are sampled from divide-by-two voltage dividers; the overlay drive software multiplies these half-scale values by two for presentation to the screen. This feature is mentioned to explain the apparent difference between the overlay screen and Block Screen format values. The Block Screen is not described here.

All the F-R System power supply voltages are shown; data values are bold-printed. The average values should not differ from the nominal values (indicated by the labels) by more than 5% and the sample-to-sample variations (from one screen scan to the next) should not exceed this 5% limit.

SECTION 7.0 SYSTEM DRAWINGS LIST

For reference purposes, the following is a complete list of all Focus-Rotation system electronics drawings. Fabrication and assembly drawings are included. FRM fabrication and assembly drawings are not included.

System Configuration

D55007K001 F-R Control System Block Diagram
D55007W001 F-R System Cable Structure
D55007W003 Apex to Pedestal Room Signal Cables
D55007W002 Apex to Pedestal Room Motor Power Cables
C52502M012 Pedestal Room NRAO Equipment Layout

S101, F-R Controller Module

D55007A002 F-R Controller Assembly
D55007S004 F-R Controller Logic Schematic
A55007B002 F-R Controller Module Assembly BOM
A55007B010 F-R Controller Front Panel Assembly BOM
B55007W006 F-R Controller Module Master Wire List
B55007W008 F-R Controller Module Hand Wire List
B55007W007 F-R Controller Module Machine Wire List
A55007A013 F-R Controller IC Location Diagram
A55007A014 F-R Controller Dip Header Assembly
C55007M036 F-R Controller Front Panel Filter
C55007Q003 F-R Controller Front Panel PCB Artwork
D55007A015 F-R Controller Front Panel PCB Assembly
D55007I001 F-R Controller Front Panel Silkscreen Artwork
D55007P001 F-R Controller Front Panel PCB Drill Drawing
D55007M010 F-R Controller Front Panel
B13050M03 Module Rails, Plain
C13720P68 Side Panel, Insulated
B13050M06 Side panel, plain
C13720M15 Module Rails, Scrub-Brush Mounting
C13740M21 Rear Panel
C13720M13 Insulating Spacer
C13720M53 Logic Connector Board, 30 Pos, 16 Pin
C13720M54 Logic Connector Board, Universal
B13050M04 Guide Block
C13050M22-1 Perforated Screen, 2-wide
C13050M70 Module Puller Knob

S102, Apex Interface Module

D55007A003 Apex Interface Module Assembly
D55007S006 Apex Interface Logic Schematic
A55007B003 Apex Interface Module Assembly Bill of Materials
A55007A016 Apex Interface Assembly IC Location Diagram
B55007W009 Apex Interface Module Master Wire List
B55007W011 Apex Interface Module Hand Wire List
B55007W010 Apex Interface Module Machine Wire List
D55007A011 Apex Interface Front Panel Display Assembly
D55007P003 Apex Interface Front Panel PCB Drill Drawing
D55007Q002 Apex Interface Front Panel PCB Artwork
D55007A010 Apex Interface Resolver/Digital Converter PCB Assembly
D55007P002 Apex Interface Res/Dig Converter PCB Drill Drawing
D55007Q001 Apex Interface Resolver/Digital Converter PCB Artwork
D55007M011 Apex Interface Module Front Panel
C55007M030 Apex Interface Module Front Panel Access Cover

C55007M031 Apex Interface Module Front Panel Filter
D550071002 Apex Interface Module Front Panel Silkscreen Artwork
A55007A017 Apex Interface Dip Header Assembly
C55007M038 Rear Panel
B13050M06 Side Panel, Plain
B13050M03 Module rails, Plain
C13720M15 Module rails, Scrub-Brush Mounting
C13740M45 Insulating Strip
B13050M04 Guide Block
C13720M53 Logic Connector Board, 30 Pos, 16 Pin
C13720M54 Logic Connector Board, Universal
C13050M22-1 Perforated Screen, 2-wide
B55007M030 Front Panel Access Cover
C13720M70 Module Puller Knob

S103, F-R Switching Module

D55007A004 F-R Switching Module Assembly
D55007S002 F-R Switching Module Schematic
A55007B004 F-R Switching Module Bill of Materials
A55007W012 F-R Switching Module Wire List
C55007S017 F-R Switching Module Resolver Excitor Schematic
NOT DRAWN F-R Switching Module Resolver Excitor Assembly
D55007P006 F-R Switching Module Resolver Excitor PCB Drill Drawing
D55007Q006 F-R Switching Module Resolver Excitor PCB Artwork
C55007A021 F-R Switching Module Brake V*I Monitor PCB Assembly
A55007B012 F-R Switching Module Brake V*I Mon PCB Ass'y BOM
D55007S008 F-R Switching Module Brake V*I PCB Schematic
C55007Q007 F-R Switching Module Brake V*I PCB Artwork
C55007P005 F-R Switching Module Brake V*I PCB Drill Drawing
D55007M006 F-R Switching Module Front Panel
D55007M007 F-R Switching Module Rear Panel
D55007M009 F-R Switching Module Excitor Mounting Bracket
D55007M008 F-R Switching Module Mounting Rails
D550071003 F-R Switching Module Front Panel Silkscreen
B13050M04 Guide Block
C13050M22-3 Perforated Screen, 4-wide

S104, F-R Interface Module

D55007A005 F-R Interface Assembly
D55007S003 F-R Interface Schematic
A55007B005 F-R Interface Module Assembly Bill of Materials
C55002A002 Standard Interface PCB Assembly, Model D
C55002S004 Standard Interface Schematic, Model D
A55007W012 F-R Interface Master Wire List
A55007W013 F-R Interface Hand Wire List
A55007W014 F-R Interface Machine Wire List
A55007A018 F-R Interface IC Location Diagram
A55007A019 F-R Interface Dip Header Assemblies
C55007M021 F-R Interface Front Panel
D550071004 F-R Interface Front Panel Silkscreen Artwork
C55007M037 Rear Panel
B13050M06 Side panel, Plain
B13050M03 Module Rails, Plain
C13720M15 Module Rails, Scrub-Brush Mounting
C13740M45 Insulating Spacer
C13720M54 Logic Connector Board, Universal
B13050M04 Guide Block
C13720M70 Module Puller Knob
C13050M22-1 Perforated Screen, 2-wide

S105, F-R Power Supply Module

D55007A006 F-R Power Supply Module Assembly
D55007S001 F-R Power Supply Module Schematic
A55007B006 F-R Power Supply Module Assembly Bill of Materials
A55007W013 F-R Power Supply Module Wire List
C55007A012 F-R Power Supply Module Front Panel PCB Assembly
C55007P004 F-R Power Supply Module Front Panel PCB Drill Drawing
C55007Q005 F-R Power Supply Module Front Panel PCB Artwork
D55007M012 F-R Power Supply Module Front Panel
D55007M001 F-R Power Supply Module Rear Panel
D55007M002 F-R Power Supply Module Rails
B13740M38 F-R Power Supply Apex Power Supply Mounting Bracket
C55007M003 F-R Power Supply Module LRS Power Supply Mounting Bracket
C55007M005 F-R Power Supply Module LND Terminal Strip Mounting Bracket
C55007M004 F-R Power Supply Module LND Power Supply Mounting Bracket
D550071005 F-R Power Supply Module Front Panel Silkscreen Artwork
B13050M04 Guide Block

Servo Amplifier Assembly

D55007A022 Servo Amplifier Chassis Assembly
D55007S010 Servo Amplifier Chassis Schematic
B55007S019 Motor Overload Relay-Emergency Stop Functional Circuit
A55007W004 Servo Amplifier Chassis Wire List
A55007B016 Servo Amplifier Chassis Assembly Bill of Materials
D55007M013 Servo Amplifier Chassis
C55007M023 Servo Amplifier Chassis Guide Pin
D55007M019 Servo Amplifier Chassis I/O Panel
D550071010 Servo Amplifier Chassis I/O Panel Silkscreen Artwork
D55007M014 Servo Amplifier Chassis Connector Panel
D550071011 Servo Amplifier Chassis Connector Panel Silkscreen Artwork
D55007M015 Servo Amplifier Chassis Top Cover Plate
D55007M017 Servo Amplifier Chassis Side Cover Plate
D55007M016 Servo Amplifier Chassis Bottom Cover Plate

D55007A023 Servo Amplifier Power Supply Module Assembly
D55007S011 Servo Amplifier Power Supply Module Schematic
A55007B015 Servo Amplifier Power Supply Module Assembly BOM
D550071006 Servo Amplifier Power Supply Mtg Pnl Silkscreen Artwork
D55007M020 Servo Amplifier Chassis BDS/PSR/RDP Mounting Plate

D55007A024 Focus Servo Amplifier Module Assembly
D55007S012 Focus Servo Amplifier Module Schematic
A55007B016 Focus Servo Amplifier Module Bill of Materials
D550071007 Focus Servo Amplifier Mounting Panel Silkscreen Artwork
C55007S005 IDD MC2 Board Schematic
B55007S018 Switching Circuit Schematic Diagram, One Phase
A55007D001 IDD BDS3 Servo Amplifier Phase Modulator Timing Diagram
D55007A025 Rotation Servo Amplifier Module Assembly
D55007S013 Rotation Servo Amplifier Module Schematic
A55007B011 Rotation Servo Amplifier Module Assembly Bill of Materials
D550071008 Rotation Servo Amplifier Mounting Panel Silkscreen Artwork

D55007A020 Rotation Backlash Controller Module Assembly
D55007S009 Rotation Backlash Controller Module Schematic
A55007B009 Rotation Backlash Controller Module Bill of Materials
D550071009 Rotation Backlash Controller Mounting Panel Silkscreen Artwork

Alignment Drawings

C55007A029 Rotation Limit Switch-Resolver Alignment
C55007A030 Focus Limit Switch-Resolver Alignment

Isolation Transformer Box

D55007A001 Isolation Transformer Box Assembly
C55007S007 Isolation Transformer Box Schematic
A55007B008 Isolation Transformer Box Bill of Materials
D55007M032 Isolation Transformer Box Enclosure

F-R Control Bins

D55007A028 F-R Control Bin Wiring Assembly
A55007W005 F-R Control Bins Wire List
C55001M007 42/50/34 Pin Connector Bin Panel
C55007M022 F-R Control Bin I/O Panel
D13740M14 F-R Control Bin Top Cover
C13740M16 F-R Control Bin Top Cover Plate Mounting Bracket, Left
C13740M15 F-R Control Bin Top Cover Plate Mounting Bracket, Right

Apex Motor Junction Box

D55007A007 Apex Motor Junction Box Assembly
C55007S015 Apex Motor Junction Box Schematic
A55007B001 Apex Motor Junction Box Assembly Bill of Materials
D55007M033 Apex Motor Junction Box Details (Fab Drawing)

Apex Control Junction Box

D55007A008 Apex Control Junction Box Assembly
D55007S014 Apex Control Junction Box Schematic
A55007B007 Apex Control Junction Box Assembly Bill of Materials
D55007M034 Apex Control Junction Box Details (Fab Drawing)

Pedestal Room Junction Box

D55007A009 Pedestal Room Control Junction Box Assembly
A55007B013 Pedestal Room Control Junction Box Ass'y Bill of Materials
D55007M018 Pedestal Room Control Junction Box Details

Cable Fabrication Drawings

C55007W019 W131-1/W132-1/W133-1 Cables, Servo Amp Chassis to Apex Motor J-Box
C55007W014 W148 F-R Controller to Servo Amplifier Chassis
C55007W017 W134-2 Pedestal Room to Apex Control J-Box J11
C55007W018 W135-2 Pedestal Room to Apex Control J-Box J12
C55007W015 W134-1 F-R Bin J3 to Pedestal Room J-Box
C55007W016 W135-1 F-R Bin J4 to Pedestal Room J-Box
C55007W025 W145/W146 Brake Excitor to Pedestal Room J-Box
C55007W028 W151 Transformer AC Power to S105
C55007W020 W131-2/W132-2/W133-2 Apex J-Box to Motors
C55007W023 W147 IDD Signals, Pedestal Room J-Box to Servo Amplifier Chassis
C55007W022 W141/W142 Rotation Position-Limit Switch to Apex Control J-Box
C55007W021 W140 Focus Position-Limit Switch to Apex Control J-Box
C55007W026 W149 Servo Amplifier 3-Phase Power Cable
C55007W027 W150 3-Phase Power to Isolation Transformer
C55007W024 W137/W138/W139 Apex J-Box to Motors

Program Listings

The Focus and Rotation Program listings follow this header page.

2500 A.D. 80186 Cross Assembler - Version 4.00g

Input Filename : REINT.asm
Output Filename : REINT.obj

```
1
2                                       LIST ON
3
4                               ;       INTERRUPT SECTION FOR ROTATION AXIS FOR VLBA ANTENNAS
5                               ;       WRITTEN BY:     WAYNE M. KOSKI
6                               ;       LAST REVISION:  NOVEMBER 29, 1989
7                               ;
8                               ;
9                               ;       THINGS TO DO AND GENERAL NOTES:
10                              ;
11                              ;         1.  THIS SECTION OF CODE WILL BE THE INTERRUPT
12                              ;             ROUTINES AND SHALL EXCLUDE THE MAIN RUNNING ROUTINES
13                              ;             WHICH SHALL BE IN THE OTHER EPROM.
14                              ;         2.  ROTATION AXIS SHOULD LOOK SIMULAR TO FOCUS AT
15                              ;             PRESENT.
16                              ;         3.  THIS VERSION UPDATES TO THE NEW F/R CONTROLLER MODULE.
17
18                                      ASSUME  CS:CODE, DS:DSEG
19
20                                      .OUTPUT 2500AD
21                                      .OPTIONS H
22
23
24                                      LIST OFF
25
26                              ;       80188 INTERNAL PORTS FOR CONTROL AND GUIDANCE
27
28
29
30                              ;       80188   INTERRUPT CONTROL / STATUS REGISTARS
31
32          0000:FF22           EOI:    .EQUAL  0FF22H          ;END OF INTERRUPT REGISTAR
33          0000:FF24           POLL:   .EQUAL  0FF24H          ;INTERRUPT POLL REGISTAR
34          0000:FF26           POLLS:  .EQUAL  0FF26H          ;INTERRUPT POLL STATUS REGISTAR
35          0000:FF28           MASK:   .EQUAL  0FF28H          ;INTERRUPT MASK REGISTAR
36          0000:FF2A           PMASK:  .EQUAL  0FF2AH          ;INTERRUPT PRIORITY MASK REGISTAR
37          0000:FF2C           ISR:    .EQUAL  0FF2CH          ;INTERRUPT IN SERVICE REGISTAR
38          0000:FF2E           IRR:    .EQUAL  0FF2EH          ;INTERRUPT REQUEST REGISTAR
39          0000:FF30           ICSR:   .EQUAL  0FF30H          ;INTERRUPT CONTROL STATUS REGISTAR
40          0000:FF32           ITCR:   .EQUAL  0FF32H          ;INTERRUPT TIMER CONTROL REGISTAR
41          0000:FF34           IDOCR:  .EQUAL  0FF34H          ;INTERRUPT DMA 0 CONTROL REGISTAR
42          0000:FF36           ID1CR:  .EQUAL  0FF36H          ;INTERRUPT DMA 1 CONTROL REGISTAR
43          0000:FF38           INTOCR: .EQUAL  0FF38H          ;INT 0 CONTROL REGISTAR
```

```
                0000:FF3A       INT1CR: .EQUAL  0FF3AH          ;INT 1 CONTROL REGISTAR
                0000:FF3C       INT2CR: .EQUAL  0FF3CH          ;INT 2 CONTROL REGISTAR
                0000:FF3E       INT3CR: .EQUAL  0FF3EH          ;INT 3 CONTROL REGISTAR


                                ;       80188 TIMER CONTROL REGISTARS

                0000:FF50       T0COUT: .EQUAL  0FF50H          ;TIMER 0 COUNT REGISTAR
                0000:FF52       T0MAXA: .EQUAL  0FF52H          ;TIMER 0 MAXIMUM COUNT A REGISTAR
                0000:FF54       T0MAXB: .EQUAL  0FF54H          ;TIMER 0 MAXIMUM COUNT B REGISTAR
                0000:FF56       T0MODE: .EQUAL  0FF56H          ;TIMER 0 MODE REGISTAR
                0000:FF58       T1COUT: .EQUAL  0FF58H          ;TIMER 1 COUNT REGISTAR
                0000:FF5A       T1MAXA: .EQUAL  0FF5AH          ;TIMER 1 MAXIMUM COUNT A REGISTAR
                0000:FF5C       T1MAXB: .EQUAL  0FF5CH          ;TIMER 1 MAXIMUM COUNT B REGISTAR
                0000:FF5E       T1MODE: .EQUAL  0FF5EH          ;TIMER 1 MODE REGISTAR
                0000:FF60       T2COUT: .EQUAL  0FF60H          ;TIMER 2 COUNT REGISTAR
                0000:FF62       T2MAXA: .EQUAL  0FF62H          ;TIMER 2 MAXIMUM COUNT A REGISTAR
                0000:FF66       T2MODE: .EQUAL  0FF66H          ;TIMER 2 MODE REGISTAR


                                ;       80188 CHIP SELECT CONTROL REGISTARS

                0000:FFA0       UMCS:   .EQUAL  0FFA0H          ;80188 INTERNAL UPPER MEMORY CHIP
                                                                ;SELECT CONTROL BLOCK REGISTAR
                0000:FFA2       LMCS:   .EQUAL  0FFA2H          ;80188 INTERNAL LOWER MEMORY CHIP
                                                                ;SELECT CONTROL BLOCK REGISTAR
                0000:FFA4       PACS:   .EQUAL  0FFA4H          ;80188 INTERNAL PERIPHERAL CHIP
                                                                ;SELECT CONTROL BLOCK REGISTAR
                0000:FFA6       MMCS:   .EQUAL  0FFA6H          ;80188 INTERNAL MIDDLE MEMORY
                                                                ;START ADDRESS REGISTAR
                0000:FFA8       MPCS:   .EQUAL  0FFA8H          ;80188 INTERNAL MIDDLE MEMORY CHIP
                                                                ;SELECT CONTROL BLOCK REGISTAR


                                ;       80188 DMA CHANNEL CONTROL REGISTARS

                0000:FFC0       D0SPL:  .EQUAL  0FFC0H          ;DMA 0 SOURCE POINTER LSB REGISTAR
                0000:FFC2       D0SPM:  .EQUAL  0FFC2H          ;DMA 0 SOURCE POINTER MSB REGISTAR
                0000:FFC4       D0DPL:  .EQUAL  0FFC4H          ;DMA 0 DESTINATION POINTER LSB REGISTAR
                0000:FFC6       D0DPM:  .EQUAL  0FFC6H          ;DMA 0 DESTINATION POINTER MSB REGISTAR
                0000:FFC8       D0TC:   .EQUAL  0FFC8H          ;DMA 0 TRANSFER COUNT REGISTAR
                0000:FFCA       D0MODE: .EQUAL  0FFCAH          ;DMA 0 MODE REGISTAR
                0000:FFD0       D1SPL:  .EQUAL  0FFD0H          ;DMA 1 SOURCE POINTER LSB REGISTAR
                0000:FFD2       D1SPM:  .EQUAL  0FFD2H          ;DMA 1 SOURCE POINTER MSB REGISTAR
                0000:FFD4       D1DPL:  .EQUAL  0FFD4H          ;DMA 1 DESTINATION POINTER LSB REGISTAR
                0000:FFD6       D1DPM:  .EQUAL  0FFD6H          ;DMA 1 DESTINATION POINTER MSB REGISTAR
                0000:FFD8       D1TC:   .EQUAL  0FFD8H          ;DMA 1 TRANSFER COUNT REGISTAR
                0000:FFDA       D1MODE: .EQUAL  0FFDAH          ;DMA 1 MODE REGISTAR


                                ;       80188 INTERNAL I/O RELOCATION REGISTAR

                0000:FFFE       RELOC:  .EQUAL  0FFFEH          ;I/O RELOCATION REGISTAR


                                ;       80188 INITIAL VALUES FOR INTERNAL REGISTARS

```

```
                0000:007D       LMBS:   .EQUAL  007DH           ;LOWER MEMORY BLOCK SIZE = 2K
                0000:81BD       MMBS:   .EQUAL  81BDH           ;MIDDLE MEMORY BLOCK SIZE = 8K
                0000:03FD       MMST:   .EQUAL  03FDH           ;MIDDLE MEMORY START POSITION = 8K
                0000:003D       PST:    .EQUAL  003DH           ;PERIPHERAL START ADDRESS = 0
                0000:FFBD       UMBS:   .EQUAL  0FFBDH          ;UPPER MEMORY BLOCK SIZE = 2K


                                ;       EXTERNAL PORTS FOR CONTROLLING THE ROTATION AXIS

                0000:0000       P1PTA:  .EQUAL  0               ;PROM 1 PORT A
                0000:0001       P1PTB:  .EQUAL  1               ;PROM 1 PORT B
                0000:0002       P1PTAD: .EQUAL  2               ;PROM 1 PORT A DIRECTION
                0000:0003       P1PTBD: .EQUAL  3               ;PROM 1 PORT B DIRECTION
                0000:0080       P2PTA:  .EQUAL  128             ;PROM 2 PORT A
                0000:0081       P2PTB:  .EQUAL  129             ;PROM 2 PORT B
                0000:0082       P2PTAD: .EQUAL  130             ;PROM 2 PORT A DIRECTION
                0000:0083       P2PTBD: .EQUAL  131             ;PROM 2 PORT B DIRECTION
                0000:0100       RAMTC:  .EQUAL  256             ;RAM TIMER AND CONTROL
                0000:0101       RAMPTA: .EQUAL  257             ;RAM PORT A
                0000:0102       RAMPTB: .EQUAL  258             ;RAM PORT B
                0000:0103       RAMPTC: .EQUAL  259             ;RAM PORT C
                0000:0104       RAMTLO: .EQUAL  260             ;RAM TIMER LOW
                0000:0105       RAMTHI: .EQUAL  261             ;RAM TIMER HIGH
                0000:0180       APEXRQ: .EQUAL  384             ;APEX REQ
                0000:0182       APEXRS: .EQUAL  386             ;APEX RESPONSE
                0000:0182       POSM:   .EQUAL  386             ;POSITION MSB
                0000:0181       POSL:   .EQUAL  385             ;POSITION LSB
                0000:0184       VELM:   .EQUAL  388             ;VELOCITY MSB
                0000:0183       VELL:   .EQUAL  387             ;VELOCITY LSB
                0000:0186       ANAH:   .EQUAL  390             ;ANALOGS MSB
                0000:0185       ANAL:   .EQUAL  389             ;ANALOGS LSB
                0000:0187       DISCR:  .EQUAL  391             ;FOC DISCRETES
                0000:018B       RELADD: .EQUAL  395             ;GET RELATIVE ADDRESS
                0000:018C       COMML:  .EQUAL  396             ;CONTROL VALUE LSB
                0000:018D       COMMM:  .EQUAL  397             ;CONTROL VALUE MSB
                0000:018E       DEVACK: .EQUAL  398             ;DEVICE ACKNOWLEDGE
                0000:018F       MONL:   .EQUAL  399             ;MONITOR DATA MSB
                0000:0190       MONM:   .EQUAL  400             ;MONITOR DATA LSB
                0000:0191       ADL:    .EQUAL  401             ;READ A/D LSB
                0000:0192       ADH:    .EQUAL  402             ;READ A/D MSB
                0000:0193       SEMCT:  .EQUAL  403             ;SELECT MOTOR CURRENT/TORQUE
                0000:0194       STCNV:  .EQUAL  404             ;START A/D CONVERT
                0000:0195       MODESW: .EQUAL  405             ;READ MODE SWITCH
                0000:0196       LTCHDA: .EQUAL  406             ;LATCH DRIVE D/A
                0000:0197       BDERL1: .EQUAL  407             ;BDS3 ERROR LSB #1
                0000:0198       BDERM1: .EQUAL  408             ;BDS3 ERROR MSB #1


0000:0000                       DSEG:   SEGMENT

                                ;       DATA STORAGE

0000:2000                               ORG     2000H

                0000:2000       MEMST:  .EQUAL  $
```

```
158                             ;       MONITOR STORAGE
159
160   0000:2000                 POSCEC: .BLKB   2               ;POSITION COMMAND ECHO
161   0000:2002                 POSD:   .BLKB   2               ;CURRENT POSITION DATA
162   0000:2004                 ERROR:  .BLKB   2               ;POSCEC-POSD
163   0000:2006                 SYSTEM: .BLKB   2               ;SYSTEM PARAMETERS
164   0000:2008                 FAUL1:  .BLKB   2               ;FAULT BITS SET 1
165   0000:200A                 FAUL2:  .BLKB   2               ;FAULT BITS SET 2
166   0000:200C                 ANAFL:  .BLKB   2               ;ANALOG FAULT FLAGS
167
168             0000:200E       ANADT:  .EQUAL  $               ;ANALOG STORAGE IS HERE
169
170   0000:200E                 GND1:   .BLKB   2               ;GND
171   0000:2010                 GND2:   .BLKB   2               ;GND
172   0000:2012                 GND3:   .BLKB   2               ;GND
173   0000:2014                 VEL:    .BLKB   2               ;ROTATION VELOCITY
174   0000:2016                 V15P:   .BLKB   2               ;+15V/2
175   0000:2018                 V15N:   .BLKB   2               ;-15V/2
176   0000:201A                 V5:     .BLKB   2               ;+5V
177   0000:201C                 V10P:   .BLKB   2               ;+10V
178   0000:201E                 MTEMP1: .BLKB   2               ;MOUNT TEMP 1
179   0000:2020                 V10N:   .BLKB   2               ;-10V
180   0000:2022                 MTEMP2: .BLKB   2               ;MOUNT TEMP 2
181   0000:2024                 BTEMP:  .BLKB   2               ;BIN TEMP
182
183   0000:2026                 SERVER: .BLKB   2               ;SERIAL/VERSION
184   0000:2028                 VEL1:   .BLKB   2
185   0000:202A                 X22:    .BLKB   2
186   0000:202C                 X23:    .BLKB   2
187   0000:202E                 X24:    .BLKB   2
188   0000:2030                 X25:    .BLKB   2
189   0000:2032                 X26:    .BLKB   2
190   0000:2034                 X27:    .BLKB   2
191   0000:2036                 X28:    .BLKB   2
192   0000:2038                 X29:    .BLKB   2
193   0000:203A                 X30:    .BLKB   2
194   0000:203C                 X31:    .BLKB   2
195   0000:203E                 X32:    .BLKB   2
196
197                             ;       TEMPORARY STORAGE
198
199   0000:2040                 RAC:    .BLKB   1               ;RELATIVE CONTROL ADDRESS
200   0000:2041                 CONL:   .BLKB   1               ;CONTROL VALUE LSB
201   0000:2042                 CONM:   .BLKB   1               ;CONTROL VALUE MSB
202   0000:2043                 ACKF:   .BLKB   1               ;ACKNOWLEDGE FLAG
203   0000:2044                 COMTMP: .BLKB   2               ;TEMPORARY COMMAND STORAGE
204   0000:2046                 ADVAL:  .BLKB   2               ;A/D TEMPORARY STORAGE
205   0000:2048                 SPEED:  .BLKB   1               ;RAMP LEVEL
206   0000:2049                 BREAK1: .BLKB   2               ;RAMP UP BREAK POINT
207   0000:204B                 BREAK2: .BLKB   2               ;RAMP DOWN BREAK POINT
208   0000:204D                 POSDOD: .BLKB   2               ;OLD POSITION
209   0000:204F                 EXTTMR: .BLKB   1               ;EXTERNAL SAFETY TIMER
210   0000:2050                 RANDOM: .BLKB   2               ;RANDOM NUMBER STORAGE
211
212
213                             ;       FLAGS
214
```

```
                0000:2052           FLAGST: .EQUAL  $

0000:2052                           RESCMD: .BLKB   1
0000:2053                           NAPATV: .BLKB   1
0000:2054                           NAPREQ: .BLKB   1
0000:2055                           DRVREQ: .BLKB   1
0000:2056                           MANOVR: .BLKB   1
0000:2057                           BDSRST  .BLKB   1
0000:2058                           DRVONE: .BLKB   1
0000:2059                           ADFLAG: .BLKB   1
0000:205A                           EFLAG:  .BLKB   1
0000:205B                           TFLAG0: .BLKB   1
0000:205C                           TFLAG1: .BLKB   1
0000:205D                           TFLAG2: .BLKB   1
0000:205E                           DRVOFF: .BLKB   1
0000:205F                           DRVATV: .BLKB   1
0000:2060                           RAMPOS: .BLKB   2
0000:2062                           DIR:    .BLKB   1
0000:2063                           LDIR:   .BLKB   1

                0000:0012           ENDFLG: .EQUAL  $-FLAGST

                                            ENDS


                                    ;       PROGRAM EQUATES

                0000:2100           STCK:   .EQUAL  2100H               ;STACK LOCATION
                0000:A766           DOVAL:  .EQUAL  0A766H              ;DMA 0 CONTROL VALUE
                0000:00E8           MASKV:  .EQUAL  00E8H               ;DMA 0, INT0, TIMER ENABLE
                0000:001C           CONRAM: .EQUAL  28                  ;?? REV/SEC CONVERGE LEVEL
                0000:8A00           LOW:    .EQUAL  00A00H .XOR. 8000H  ;LOW VALUE
                0000:7200           HIGH:   .EQUAL  0F200H .XOR. 8000H  ;HIGH VALUE

                                            LIST ON

                                    ;       EXTERNAL REFERENCES

                                            GLOBAL  DSETUP, TSETUP, TSET1, CLSETAB
                                            EXTERNAL        INITAL:FAR, RSCMD:FAR


0000:0000                                   ORG     0000H

0000:0000   0000 0000               TYPE0:  .LONG   INITAL              ;DIVIDE ERROR EXCEPTION
0000:0004   0000 0000               TYPE1:  .LONG   INITAL              ;SINGLE STEP EXCEPTION
0000:0008   5000 0000               TYPE2:  .LONG   NMI                 ;NMI
0000:000C   B001 0000               TYPE3:  .LONG   TMR2                ;BREAKPOINT INTERRUPT
0000:0010   0000 0000               TYPE4:  .LONG   INITAL              ;INTO EXCEPTION
0000:0014   9D00 0000               TYPE5:  .LONG   BOUND               ;ARRAY BOUND EXCEPTION
0000:0018   0000 0000               TYPE6:  .LONG   INITAL              ;UNUSED OPCODE EXCEPTION
0000:001C   0000 0000               TYPE7:  .LONG   INITAL              ;ESC OPCODE EXCEPTION
0000:0020   0000 0000               TYPE8:  .LONG   INITAL              ;TIMER 0 INTERRUPT
0000:0024   0000 0000               TYPE9:  .LONG   INITAL              ;RESERVED
0000:0028   AE00 0000               TYPE10: .LONG   DMA0                ;DMA 0
0000:002C   0000 0000               TYPE11: .LONG   INITAL              ;DMA 1
0000:0030   8101 0000               TYPE12: .LONG   INT0                ;INT 0 INTERRUPT
```

```
0000:0034  0000 0000          TYPE13: .LONG   INITAL              ;INT 1 INTERRUPT
0000:0038  0000 0000          TYPE14: .LONG   INITAL              ;INT 2 INTERRUPT
0000:003C  0000 0000          TYPE15: .LONG   INITAL              ;INT 3 INTERRUPT
0000:0040  0000 0000          TYPE16: .LONG   INITAL              ;RESERVED
0000:0044  0000 0000          TYPE17: .LONG   INITAL              ;RESERVED
0000:0048  9F01 0000          TYPE18: .LONG   TMR1                ;TIMER 1 INTERRUPT
0000:004C  B901 0000          TYPE19: .LONG   TMR2                ;TIMER 2 INTERRUPT

                              ;       NMI INTERRUPT IS LOCATED HERE AFTER TABLE

                              ;       THIS INTERRUPT IS USED TO SEND MONITOR DATA TO
                              ;       THE STANDARD INTERFACE.
0000:0050
0000:0050                     NMI:
0000:0050  50                         PUSH    AX                  ;SAVE REGISTARS, ETC
0000:0051  52                         PUSH    DX
0000:0052  53                         PUSH    BX
0000:0053  FF 36 5A 20                PUSH    BFLAG
0000:0057  C6 06 5A 20 00             MOV     BFLAG,0             ;RELATIVE ADDRESS TEST
0000:005C  BA 8B 01                   MOV     DX,RELADD           ;GET RELATIVE ADDRESS
0000:005F  EC                         IN      AL,DX
0000:0060  B4 00                      MOV     AH,0
0000:0062  2E 62 06 62 05             BOUND   AX,CS:MONRG         ;RELATIVE ADDRESS IN RANGE?
0000:0067  F6 06 5A 20 01             TEST    BFLAG,1
0000:006C  74 07                      JZ      NMI1                ;IF OK
0000:006E  80 0E 08 20 40             OR      FAUL1,64            ;SET MONITOR FAULT FLAG
0000:0073  EB 19                      JMP     SHORT NEXIT         ;EXIT ROUTINE
0000:0075                     NMI1:
0000:0075  2E 2B 06 62 05             SUB     AX,CS:MONRG         ;SUBTRACT OFFSET
0000:007A  03 C0                      ADD     AX,AX               ;TIMES 2
0000:007C  8B D8                      MOV     BX,AX               ;BX IS NOW THE INDEX VALUE
0000:007E  8B 87 00 20                MOV     AX,MEMST[BX]        ;GET MONITOR DATA
0000:0082  BA 8F 01                   MOV     DX,MONL             ;SEND IT
0000:0085  EF                         OUT     DX,AX
0000:0086  4A                         DEC     DX
0000:0087  EE                         OUT     DX,AL               ;SEND DEVICE ACKNOWLEDGE
0000:0088  81 26 08 20 BF FF          AND     WORD PTR FAUL1,0FFBFH ;RESET MONITOR FAULT FLAG
0000:008E  B8 02 00           NEXIT:  MOV     AX,2                ;SPECIFIC EOI
0000:0091  BA 22 FF                   MOV     DX,EOI
0000:0094  EF                         OUT     DX,AX
0000:0095  8F 06 5A 20                POP     BFLAG               ;RESTORE REGISTARS, ETC
0000:0099  5B                         POP     BX
0000:009A  5A                         POP     DX
0000:009B  58                         POP     AX
0000:009C  CF                         IRET

                              ;       THIS INTERRUPT SETS THE OUT OF BOUNDS FLAG

0000:009D                     BOUND:
0000:009D  50                         PUSH    AX                  ;SAVE REGISTARS
0000:009E  52                         PUSH    DX
0000:009F  C6 06 5A 20 01             MOV     BFLAG,1             ;SET OUT OF BOUNDS FLAG
0000:00A4  B8 05 00                   MOV     AX,5                ;SPECIFIC EOI
0000:00A7  BA 22 FF                   MOV     DX,EOI
0000:00AA  EF                         OUT     DX,AX
0000:00AB  5A                         POP     DX                  ;RESTORE REGISTARS
0000:00AC  58                         POP     AX
```

```
0000:00AD  CF                          IRET
                                ;       THIS INTERRUPT COMPLETES THE PROCESSING OF A COMMAND
                                ;       INPUTTED FROM THE STANDARD INTERFACE
0000:00AE                       DMA0:
0000:00AE  50                          PUSH    AX                      ;SAVE REGISTARS, ETC
0000:00AF  52                          PUSH    DX
0000:00B0  51                          PUSH    CX
0000:00B1  53                          PUSH    BX
0000:00B2  56                          PUSH    SI
0000:00B3  FF 36 5A 20                 PUSH    BFLAG
0000:00B7  C6 06 5A 20 00              MOV     BFLAG,0                 ;ASSUME IN BOUNDS
0000:00BC  8A 1E 40 20                 MOV     BL,RAC                  ;GET RELATIVE ADDRESS
0000:00C0  B7 00                       MOV     BH,0                    ;STRIP UPPER NYBBLE
0000:00C2  2E 62 1E 48 05              BOUND   BX,CS:CMDRG             ;VALID COMMAND?
0000:00C7  F6 06 5A 20 01              TEST    BFLAG,1                 ;IN BOUND?
0000:00CC  74 08                       JZ      DMA01                   ;IF OK
0000:00CE  80 0E 08 20 20              OR      FAUL1,32                ;SET COMMAND INVALID
0000:00D3  E9 8B 00                    JMP     DEXIT
0000:00D6                       DMA01:
0000:00D6  81 26 08 20 DF FF           AND     WORD PTR FAUL1,0FFDFH   ;CLEAR COMMAND INVALID
0000:00DC  2E 2B 1E 48 05              SUB     BX,CS:CMDRG             ;SUBTRACT OFFSET
0000:00E1  03 DB                       ADD     BX,BX                   ;TIMES 2
0000:00E3  2E 8B 87 4C 05              MOV     AX,CS:CMDTBL[BX]        ;GET NEW PROGRAM POINT
0000:00E8  FF E0                       JMP     AX                      ;GO THERE
                                ;       THE NEXT FIVE ROUTINES ARE THE COMMAND PROCESSING ENTRY POINTS
0000:00EA                       POSCMD:
0000:00EA  A1 41 20                    MOV     AX,CONL                 ;GET COMMANDED POSITION
0000:00ED  35 00 80                    XOR     AX,8000H                ;FOR BOUND TEST
0000:00F0  2E 62 06 66 05              BOUND   AX,CS:POSRG             ;POSITION WITHIN RANGE?
0000:00F5  35 00 80                    XOR     AX,8000H                ;UNDO AFTER TEST
0000:00F8  F6 06 5A 20 01              TEST    BFLAG,1
0000:00FD  74 07                       JZ      POSCM1                  ;IF IN RANGE
0000:00FF  80 0E 08 20 80              OR      FAUL1,128               ;SET OPERATOR FAULT
0000:0104  EB 5B                       JMP     SHORT DEXIT
0000:0106  80 26 08 20 7F       POSCM1: AND    FAUL1,7FH               ;RESET OPERATOR FAULT
0000:010B  A3 44 20                    MOV     COMTMP,AX               ;SAVE COMMANDED POSITION HERE
0000:010E  C6 06 55 20 01              MOV     DRVREQ,1                ;SET DRIVE REQUEST
0000:0113  C6 06 58 20 00              MOV     DRVONE,0                ;CLEAR SECOND TRY FLAG
0000:0118  EB 47                       JMP     SHORT DEXIT
0000:011A                       RECMD:
0000:011A  C6 06 52 20 01              MOV     RESCMD,1                ;SET SOFT RESET REQUEST
0000:011F  EB 40                       JMP     SHORT DEXIT
0000:0121                       NAPCMD:
0000:0121  F6 06 41 20 01              TEST    CONL,1                  ;TEST FOR SET / RESET
0000:0126  75 0C                       JNZ     NAPCM1
0000:0128  C6 06 53 20 00              MOV     NAPATV,0                ;IF CLEAR NAP
0000:012D  80 26 06 20 FE              AND     SYSTEM,0FEH
0000:0132  EB 2D                       JMP     SHORT DEXIT
0000:0134  F6 06 53 20 01       NAPCM1: TEST   NAPATV,1                ;ALREADY NAPPED?
0000:0139  75 26                       JNZ     DEXIT
0000:013B  C6 06 54 20 01              MOV     NAPREQ,1                ;SET NAP REQUEST
```

```
0000:0140  EB 1F                    JMP     SHORT DEXIT

0000:0142                   MANCMD:
0000:0142  F6 06 41 20 01           TEST    CONL,1              ;TEST FOR SET / RESET
0000:0147  75 0C                    JNZ     MANCM1
0000:0149  C6 06 56 20 00           MOV     MANOVR,0            ;IF CLEAR OVER RIDE
0000:014E  80 26 06 20 F7           AND     SYSTEM,0F7H
0000:0153  EB 0C                    JMP     SHORT DEXIT
0000:0155  C6 06 56 20 01   MANCM1: MOV     MANOVR,1            ;SET MANUEL OVER-RIDE
0000:015A  EB 05                    JMP     SHORT DEXIT

0000:015C  C6 06 57 20 01   BDSCMD: MOV     BDSRST,1            ;SET BDS3 SERVO RESET

0000:0161  BE 56 05         DEXIT:  MOV     SI,CS:OFFSET DSETUP ;RE-INITIALIZE DMA 0
0000:0164  BA C0 FF                 MOV     DX,DOSPL
0000:0167  B9 06 00                 MOV     CX,6
0000:016A  6F               DSET:   OUTSW
0000:016B  83 C2 02                 ADD     DX,2
0000:016E  E2 FA                    LOOP    DSET
0000:0170  B8 0A 00                 MOV     AX,10               ;SPECIFIC EOI
0000:0173  BA 22 FF                 MOV     DX,EOI
0000:0176  EF                       OUT     DX,AX
0000:0177  8F 06 5A 20              POP     BFLAG               ;RESTORE REGISTARS, ETC
0000:017B  5E                       POP     SI
0000:017C  5B                       POP     BX
0000:017D  59                       POP     CX
0000:017E  5A                       POP     DX
0000:017F  58                       POP     AX
0000:0180  CF                       IRET

                            ;       THIS INTERRUPT ROUTINE LOADS THE ANALOG TO DIGITAL VALUES
                            ;       AFTER CONVERSION

0000:0181                   INT0:
0000:0181  50                       PUSH    AX                  ;SAVE REGISTARS
0000:0182  52                       PUSH    DX
0000:0183  BA 91 01                 MOV     DX,ADL              ;GET A/D VALUE
0000:0186  ED                       IN      AX,DX
0000:0187  C1 E8 04                 SHR     AX,4                ;PUT INTO POSITION
0000:018A  35 00 08                 XOR     AX,0800H            ;TWOS COMPLEMENT
0000:018D  A3 46 20                 MOV     ADVAL,AX            ;STORE HERE
0000:0190  C6 06 59 20 01           MOV     ADFLAG,1            ;SET ADFLAG
0000:0195  B8 0C 00                 MOV     AX,12               ;SPECIFIC EOI
0000:0198  BA 22 FF                 MOV     DX,EOI
0000:019B  EF                       OUT     DX,AX
0000:019C  5A                       POP     DX                  ;RESTORE REGISTARS
0000:019D  58                       POP     AX
0000:019E  CF                       IRET

0000:019F                   TMR1:
0000:019F  50                       PUSH    AX                  ;SAVE REGISTARS
0000:01A0  52                       PUSH    DX
0000:01A1  C6 06 5C 20 01           MOV     TFLAG1,1            ;SET TIMER 1 FLAG
0000:01A6  B8 08 00                 MOV     AX,8                ;SPECIFIC EOI
0000:01A9  BA 22 FF                 MOV     DX,EOI
0000:01AC  EF                       OUT     DX,AX
0000:01AD  5A                       POP     DX                  ;RESTORE REGISTARS
```

```
0000:01AE  58                    POP     AX
0000:01AF  CF                    IRET
0000:01B0                TMR2:
0000:01B0  50                    PUSH    AX                ;SAVE REGISTARS
0000:01B1  53                    PUSH    BX
0000:01B2  51                    PUSH    CX
0000:01B3  52                    PUSH    DX
0000:01B4  FF 36 5A 20           PUSH    BFLAG
0000:01B8  C6 06 4F 20 19        MOV     EXTTMR,25         ;RESET TIMER
0000:01BD  F6 06 5F 20 01        TEST    DRVATV,1
0000:01C2  75 06                 JNZ     GO
0000:01C4  E8 6D 03              CALL    TMROFF
0000:01C7  E9 4D 02              JMP     TMR2EX
0000:01CA  F6 06 5E 20 01    GO: TEST    DRVOFF,1          ;GO INTO SHUTDOWN?
0000:01CF  74 03                 JZ      TMR21             ;IF NOT SHUT DOWN
0000:01D1  E9 E9 01              JMP     OFFIT             ;IF SHUT DOWN
0000:01D4  E4 81             TMR21: IN      AL,P2PTB          ;GET E-STOP, DRIVE LOCKOUT
0000:01D6  24 18                 AND     AL,18H            ;STRIP JUNK
0000:01D8  34 10                 XOR     AL,10H            ;INVERT DRIVE LOCKOUT
0000:01DA  74 03                 JZ      TMR210            ;IF OK
0000:01DC  E9 15 02              JMP     OFFIT3            ;ELSE QUIT
0000:01DF  E8 C0 02          TMR210: CALL    DSTOR             ;GET LATEST DATA
0000:01E2  BA 87 01              MOV     DX,DISCR          ;POINT TO DISCRETES
0000:01E5  EC                    IN      AL,DX             ;GET INFO
0000:01E6  F6 06 63 20 01        TEST    LDIR,1            ;TEST DIRECTION
0000:01EB  74 07                 JZ      TMR21A            ;IF POSITIVE
0000:01ED  24 0C                 AND     AL,0CH            ;CCW LIMIT?
0000:01EF  74 0A                 JZ      TMR22
0000:01F1  E9 C9 01              JMP     OFFIT             ;IF IN CCW LIMIT
0000:01F4  24 03             TMR21A: AND     AL,3              ;CW LIMIT?
0000:01F6  74 03                 JZ      TMR22
0000:01F8  E9 C2 01              JMP     OFFIT             ;IF IN CW LIMIT
0000:01FB  A0 62 20          TMR22: MOV     AL,DIR            ;GET DIRECTION
0000:01FE  3A 06 63 20           CMP     AL,LDIR           ;EQUAL?
0000:0202  74 08                 JZ      TMR22A            ;IF EQUAL
0000:0204  C6 06 60 20 06        MOV     RAMPOS,6          ;KILL ALL BUT CONVERGE
0000:0209  E9 15 00              JMP     VERGIT            ;GOTO CONVERGE
0000:020C  80 3E 48 20 0A    TMR22A: CMP     BYTE PTR SPEED,10 ;MOTION ANALYSIS?
0000:0211  75 03                 JNZ     TMR23             ;IF NOT
0000:0213  E9 15 02              JMP     MOTION
0000:0216  8B 1E 60 20       TMR23: MOV     BX,RAMPOS         ;GET CURRENT RAMP
0000:021A  2E 8B 87 6E 05        MOV     AX,CS:RAMPTL[BX]  ;GET INDEX
0000:021F  FF E0                 JMP     AX                ;GO THERE

0000:0221                VERGIT:
0000:0221  8B 1E 04 20           MOV     BX,ERROR          ;ARE WE CLOSE?
0000:0225  C6 06 5A 20 00        MOV     BFLAG,0           ;SET IN BOUNDS
0000:022A  2E 62 1E 44 05        BOUND   BX,CS:CLSETAB     ;CLOSE?
0000:022F  F6 06 5A 20 01        TEST    BFLAG,1
0000:0234  75 35                 JNZ     VERGI1            ;IF NOT
0000:0236  E5 00                 IN      AX,P1PTA          ;FINAL RAMP TO STOP
0000:0238  50                    PUSH    AX                ;SAVE IT
0000:0239  25 FF 0F              AND     AX,0FFFH          ;JUST RAMP VALUE
0000:023C  BB 04 00              MOV     BX,4              ;PREPARE TO DECREASE
0000:023F  F6 06 63 20 01        TEST    LDIR,1            ;CHECK DIRECTION
0000:0244  74 02                 JZ      VERGI0            ;IF POSITIVE
```

```
0000:0246  F7 DB                    NEG   BX                    ;IF NEGATIVE
0000:0248  2B C3            VERGI0: SUB   AX,BX
0000:024A  25 FF 0F                 AND   AX,0FFFH              ;JUST RAMP VALUE
0000:024D  5B                       POP   BX                    ;RESTORE
0000:024E  81 E3 00 F0              AND   BX,0F000H             ;JUST CONTROL
0000:0252  3D 00 00                 CMP   AX,0                  ;DONE?
0000:0255  75 09                    JNE   VERGIA                ;NOT DONE
0000:0257  81 26 08 20 FF FD        AND   WORD PTR FAUL1,0FDFFH ;RESET DRIVE FAULT FLAG
0000:025D  E9 94 01                 JMP   OFFIT3                ;COMPLETE
0000:0260  0B C3            VERGIA: OR    AX,BX                 ;MERGE
0000:0262  E7 00                    OUT   P1PTA,AX              ;SEND NEW RAMP
0000:0264  BA 96 01                 MOV   DX,LTCHDA             ;NOW LATCH D/A
0000:0267  EE                       OUT   DX,AL                 ;NEW RAMP COMPLETE
0000:0268  E9 AC 01                 JMP   TMR2EX

0000:026B                   VERGI1:
0000:026B  A0 62 20                 MOV   AL,DIR                ;CHECK FOR DIRECTION SWITCH
0000:026E  3A 06 63 20              CMP   AL,LDIR
0000:0272  75 07                    JNE   VERGIB                ;WE OVER SHOT
0000:0274  BA 96 01                 MOV   DX,LTCHDA             ;NOW LATCH D/A
0000:0277  EE                       OUT   DX,AL                 ;OLD RAMP COMPLETE
0000:0278  E9 9C 01                 JMP   TMR2EX                ;IF NOT SWITCHED
0000:027B  E5 00            VERGIB: IN    AX,P1PTA              ;FINAL RAMP TO STOP
0000:027D  50                       PUSH  AX                    ;SAVE IT
0000:027E  25 FF 0F                 AND   AX,0FFFH              ;JUST RAMP VALUE
0000:0281  BB 04 00                 MOV   BX,4                  ;PREPARE TO DECREASE
0000:0284  F6 06 63 20 01           TEST  LDIR,1                ;CHECK DIRECTION
0000:0289  74 02                    JZ    VERGIC                ;IF POSITIVE
0000:028B  F7 DB                    NEG   BX                    ;IF NEGATIVE
0000:028D  2B C3            VERGIC: SUB   AX,BX
0000:028F  25 FF 0F                 AND   AX,0FFFH              ;JUST RAMP VALUE
0000:0292  5B                       POP   BX                    ;RESTORE
0000:0293  81 E3 00 F0              AND   BX,0F000H             ;JUST CONTROL
0000:0297  50                       PUSH  AX                    ;SAVE NEW RAMP
0000:0298  0B C3                    OR    AX,BX                 ;MERGE
0000:029A  E7 00                    OUT   P1PTA,AX              ;SEND NEW RAMP
0000:029C  BA 96 01                 MOV   DX,LTCHDA             ;NOW LATCH D/A
0000:029F  EE                       OUT   DX,AL                 ;NEW RAMP COMPLETE
0000:02A0  58                       POP   AX                    ;RESTORE NEW RAMP
0000:02A1  3D 00 00                 CMP   AX,0                  ;DONE?
0000:02A4  74 03                    JE    VERGI2                ;DONE
0000:02A6  E9 6E 01                 JMP   TMR2EX

0000:02A9                   VERGI2:
0000:02A9  F6 06 58 20 01           TEST  DRVONE,1              ;ON SECOND TRY?
0000:02AE  75 1A                    JNZ   VERGI3                ;IF SECOND TRY
0000:02B0  C6 06 58 20 01           MOV   DRVONE,1              ;SET SECOND TRY FLAG
0000:02B5  C6 06 5F 20 00           MOV   DRVATV,0              ;SET DRIVE TO NON ACTIVE
0000:02BA  C6 06 55 20 01           MOV   DRVREQ,1              ;RESTART COMMAND
0000:02BF  8B 1E 00 20              MOV   BX,POSCEC             ;GET COMTMP = POSCEC
0000:02C3  89 1E 44 20              MOV   COMTMP,BX
0000:02C7  E9 2A 01                 JMP   OFFIT3                ;LEAVE
0000:02CA                   VERGI3:
0000:02CA  81 0E 08 20 00 02        OR    WORD PTR FAUL1,200H   ;SET DRIVE FAULT
0000:02D0  E9 21 01                 JMP   OFFIT3                ;STOP

0000:02D3  C6 06 5A 20 00   ZIPUP:  MOV   BFLAG,0               ;RESET BOUND FLAG
```

```
0000:02D8  FE 0E 48 20          DEC   SPEED
0000:02DC  75 34                JNZ   ZIPUP2          ;IF NO CHANGE IN RAMP
0000:02DE  C6 06 48 20 14       MOV   SPEED,20        ;RESET SPEED
0000:02E3  E5 00                IN    AX,P1PTA        ;GET CURRENT RAMP POINT
0000:02E5  50                   PUSH  AX              ;SAVE IT
0000:02E6  25 FF 0F             AND   AX,0FFFH        ;STRIP CONTROL NYBBLE
0000:02E9  BB 04 00             MOV   BX,4            ;DO NEXT RAMP STEP
0000:02EC  F6 06 63 20 01       TEST  LDIR,1          ;NEGATIVE?
0000:02F1  74 02                JZ    ZIPUP1          ;IF POSITIVE
0000:02F3  F7 DB                NEG   BX              ;REVERSE SIGN
0000:02F5  03 C3          ZIPUP1: ADD   AX,BX           ;NEW RAMP
0000:02F7  25 FF 0F             AND   AX,0FFFH        ;STRIP UPPER NYBBLE
0000:02FA  C1 C0 04             ROL   AX,4            ;ROTATE FOR BOUND TEST
0000:02FD  2E 62 06 6A 05       BOUND AX,CS:RAMPLM
0000:0302  C1 C8 04             ROR   AX,4            ;PUT BACK
0000:0305  5B                   POP   BX              ;GET OLD CONTROL BITS
0000:0306  81 E3 00 F0          AND   BX,0F000H       ;STRIP OLD RAMP
0000:030A  0B C3                OR    AX,BX           ;MERGE
0000:030C  E7 00                OUT   P1PTA,AX        ;SEND NEW RAMP
0000:030E  BA 96 01             MOV   DX,LTCHDA       ;NOW LATCH D/A
0000:0311  EE                   OUT   DX,AL           ;NEW RAMP COMPLETE
0000:0312  F6 06 5A 20 01 ZIPUP2: TEST  BFLAG,1       ;OUT OF BOUNDS?
0000:0317  75 13                JNZ   ZIPUP3          ;IF OUT
0000:0319  A1 02 20             MOV   AX,POSD         ;CURRENT POSITION
0000:031C  F6 06 63 20 01       TEST  LDIR,1          ;CHECK DIRECTION
0000:0321  75 11                JNZ   ZIPUP4          ;IF NEGATIVE
0000:0323  3B 06 49 20          CMP   AX,BREAK1       ;RAMP INTO MAIN?
0000:0327  73 03                JNC   ZIPUP3          ;GO INTO MAIN
0000:0329  E9 EB 00             JMP   TMR2EX          ;NOPE
0000:032C  80 06 60 20 02 ZIPUP3: ADD   RAMPOS,2      ;GO INTO MAIN
0000:0331  E9 E3 00             JMP   TMR2EX          ;NOW LEAVE
0000:0334  3B 06 49 20    ZIPUP4: CMP   AX,BREAK1     ;RAMP INTO MAIN?
0000:0338  72 F2                JC    ZIPUP3          ;GO INTO MAIN
0000:033A  E9 DA 00             JMP   TMR2EX          ;NOPE
                                
0000:033D                 MAINNY:
0000:033D  FE 0E 48 20          DEC   SPEED
0000:0341  75 05                JNZ   MAINN0          ;FOR MOTION ANALYSIS
0000:0343  C6 06 48 20 14       MOV   SPEED,20        ;RESET SPEED
0000:0348  A1 02 20       MAINN0: MOV   AX,POSD         ;CURRENT POSITION
0000:034B  F6 06 63 20 01       TEST  LDIR,1          ;CHECK DIRECTION
0000:0350  75 0F                JNZ   MAINN2          ;IF NEGATIVE
0000:0352  3B 06 4B 20          CMP   AX,BREAK2       ;MAIN INTO RAMP DOWN?
0000:0356  73 02                JNC   MAINN1          ;GO INTO RAMP DOWN
0000:0358  EB 0D                JMP   SHORT MAINN3    ;NOPE
0000:035A  80 06 60 20 02 MAINN1: ADD   RAMPOS,2      ;SET RAMP DOWN
0000:035F  EB 06                JMP   SHORT MAINN3    ;NOW LEAVE
0000:0361  3B 06 4B 20    MAINN2: CMP   AX,BREAK2     ;MAIN INTO RAMP DOWN?
0000:0365  72 F3                JC    MAINN1          ;GO INTO RAMP DOWN
0000:0367  BA 96 01       MAINN3: MOV   DX,LTCHDA       ;NOW LATCH D/A
0000:036A  EE                   OUT   DX,AL           ;OLD RAMP COMPLETE
0000:036B  E9 A9 00             JMP   TMR2EX          ;DONE
                                
0000:036E  FE 0E 48 20    ZIPDWN: DEC   SPEED
0000:0372  75 46                JNZ   ZIPDN3          ;IF NO CHANGE IN RAMP
0000:0374  C6 06 48 20 14       MOV   SPEED,20        ;RESET SPEED
0000:0379  E5 00                IN    AX,P1PTA        ;GET CURRENT RAMP POINT
```

```
614  0000:037B  50                    PUSH   AX                ;SAVE IT
615  0000:037C  25 FF 0F              AND    AX,0FFFH          ;STRIP CONTROL NYBBLE
616  0000:037F  BB 04 00              MOV    BX,4              ;DO NEXT RAMP STEP
617  0000:0382  F6 06 63 20 01        TEST   LDIR,1            ;NEGATIVE?
618  0000:0387  74 02                 JZ     ZIPDN1            ;IF POSITIVE
619  0000:0389  F7 DB                 NEG    BX                ;REVESRE SIGN
620  0000:038B  2B C3          ZIPDN1: SUB    AX,BX             ;NEW RAMP
621  0000:038D  25 FF 0F              AND    AX,0FFFH          ;STRIP UPPER NYBBLE
622  0000:0390  5B                    POP    BX                ;GET OLD CONTROL BITS
623  0000:0391  74 15                 JZ     ZIPDN2            ;IF ZERO
624  0000:0393  81 E3 00 F0           AND    BX,0F000H         ;STRIP OLD RAMP
625  0000:0397  0B C3                 OR     AX,BX             ;MERGE
626  0000:0399  E7 00                 OUT    P1PTA,AX          ;SEND NEW RAMP
627  0000:039B  BA 96 01              MOV    DX,LTCHDA         ;NOW LATCH D/A
628  0000:039E  EE                    OUT    DX,AL             ;NEW RAMP COMPLETE
629  0000:039F  F6 06 63 20 01        TEST   LDIR,1            ;CHECK DIRECTION
630  0000:03A4  74 02                 JZ     ZIPDN2            ;IF POSITIVE
631  0000:03A6  F7 D8                 NEG    AX                ;INVERT FOR CHECK
632  0000:03A8  25 FF 0F       ZIPDN2: AND    AX,0FFFH          ;CURRENT RAMP
633  0000:03AB  3D 1C 00              CMP    AX,CONRAM         ;RAMP INTO CONVERGE?
634  0000:03AE  73 67                 JNC    TMR2EX            ;NOPE
635  0000:03B0  C6 06 48 20 01        MOV    SPEED,1           ;RESET SPEED
636  0000:03B5  80 06 60 20 02        ADD    RAMPOS,2          ;KILL RAMP DOWN
637  0000:03BA  E9 5A 00       ZIPDN3: JMP    TMR2EX            ;NOW LEAVE
638
639  0000:03BD  E5 00          OFFIT:  IN     AX,P1PTA          ;GET CURRENT RAMP POINT
640  0000:03BF  FE 0E 48 20           DEC    SPEED
641  0000:03C3  75 27                 JNZ    OFFIT2            ;IF NO CHANGE IN RAMP
642  0000:03C5  C6 06 48 20 14        MOV    SPEED,20          ;RESET SPEED
643  0000:03CA  50                    PUSH   AX                ;SAVE IT
644  0000:03CB  25 FF 0F              AND    AX,0FFFH          ;STRIP CONTROL NYBBLE
645  0000:03CE  BB 04 00              MOV    BX,4              ;DO NEXT RAMP STEP
646  0000:03D1  F6 06 63 20 01        TEST   LDIR,1            ;NEGATIVE?
647  0000:03D6  74 02                 JZ     OFFIT1            ;IF POSITIVE
648  0000:03D8  F7 DB                 NEG    BX                ;REVESRE SIGN
649  0000:03DA  2B C3          OFFIT1: SUB    AX,BX             ;NEW RAMP
650  0000:03DC  25 FF 0F              AND    AX,0FFFH          ;STRIP UPPER NYEBLE
651  0000:03DF  5B                    POP    BX                ;GET OLD CONTROL BITS
652  0000:03E0  81 E3 00 F0           AND    BX,0F000H         ;STRIP OLD RAMP
653  0000:03E4  0B C3                 OR     AX,BX             ;MERGE
654  0000:03E6  E7 00                 OUT    P1PTA,AX          ;SEND NEW RAMP
655  0000:03E8  BA 96 01              MOV    DX,LTCHDA         ;NOW LATCH D/A
656  0000:03EB  EE                    OUT    DX,AL             ;NEW RAMP COMPLETE
657  0000:03EC  25 FF 0F       OFFIT2: AND    AX,0FFFH          ;CURRENT RAMP
658  0000:03EF  3D 00 00              CMP    AX,0              ;GOT TO 0?
659  0000:03F2  75 23                 JNE    TMR2EX            ;NOPE
660  0000:03F4  E8 3D 01       OFFIT3: CALL   TMROFF            ;KILL TIMER
661  0000:03F7  B8 00 C0              MOV    AX,0C000H         ;KILL DRIVE
662  0000:03FA  E7 00                 OUT    P1PTA,AX
663  0000:03FC  B8 08 00              MOV    AX,8              ;SPECIFIC EOI
664  0000:03FF  BA 22 FF              MOV    DX,EOI
665  0000:0402  EF                    OUT    DX,AX
666  0000:0403  E8 EA 00              CALL   BRKOFF            ;KILL BRAKE
667  0000:0406  C6 06 5F 20 00        MOV    DRVATV,0          ;KILL DRIVE ACTIVE FLAG
668  0000:040B  C6 06 5E 20 00        MOV    DRVOFF,0          ;KILL DRIVE OFF FLAG
669  0000:0410  E5 80                 IN     AX,P2PTA
670  0000:0412  25 97 FF              AND    AX,0FF97H         ;KILL SPECIFIC LIGHTS
```

```
671  0000:0415  E7 80                    OUT   P2PTA,AX
672  0000:0417  B8 08 00          TMR2EX: MOV   AX,8                ;SPECIFIC EOI
673  0000:041A  BA 22 FF                  MOV   DX,EOI
674  0000:041D  EF                        OUT   DX,AX
675  0000:041E  B8 03 00                  MOV   AX,3                ;AGAIN
676  0000:0421  EF                        OUT   DX,AX
677  0000:0422  8F 06 5A 20               POP   BFLAG
678  0000:0426  5A                        POP   DX                  ;RESTORE REGISTARS
679  0000:0427  59                        POP   CX
680  0000:0428  5B                        POP   BX
681  0000:0429  58                        POP   AX
682  0000:042A  CF                        IRET
683
684  0000:042B                    MOTION:
685  0000:042B  FB                        STI                       ;ENABLE INTERRUPTS
686  0000:042C  E5 80                     IN    AX,P2PTA            ;GET PORT 2
687  0000:042E  25 F8 FF                  AND   AX,0FFF8H           ;STRIP OLD A/D REQ
688  0000:0431  0D 01 00                  OR    AX,1                ;REQUEST VELOCITY
689  0000:0434  E7 80                     OUT   P2PTA,AX
690  0000:0436  BA 94 01                  MOV   DX,STCNV            ;START CONVERSION
691  0000:0439  EE                        OUT   DX,AL
692  0000:043A  C6 06 59 20 00            MOV   ADFLAG,0            ;RESET FLAG
693  0000:043F  FE 0E 48 20               DEC   SPEED               ;UPDATE SPEED
694  0000:0443  BA 97 01                  MOV   DX,BDERL1           ;GET SERVO AMP FAULTS
695  0000:0446  ED                        IN    AX,DX
696  0000:0447  F7 D0                     NOT   AX                  ;COMPLEMENT THEM
697  0000:0449  A3 0A 20                  MOV   FAUL2,AX            ;SAVE IT
698  0000:044C  25 FD FD                  AND   AX,0FDFDH           ;IGNORE FOLDBACK
699  0000:044F  74 08                     JZ    MOTI01              ;ANY PROBLEMS?
700  0000:0451  C6 06 5E 20 01            MOV   DRVOFF,1            ;YES, SO KILL DRIVE
701  0000:0456  E9 BE FF                  JMP   TMR2EX
702  0000:0459                    MOTI01:
703  0000:0459  8B 1E 02 20               MOV   BX,POSD             ;GET CURRENT POSITION
704  0000:045D  A1 4D 20                  MOV   AX,POSDOD           ;GET OLD POSITION
705  0000:0460  89 1E 4D 20               MOV   POSDOD,BX           ;CURRENT = OLD
706  0000:0464  2B D8                     SUB   BX,AX               ;GET DIFFERENCE
707  0000:0466  79 02                     JNS   MOTI02              ;IF POSITIVE
708  0000:0468  F7 DB                     NEG   BX                  ;MAKE POSITIVE
709  0000:046A  8B 16 28 20       MOTI02: MOV   DX,VEL1             ;GET PREVIOUS VELOCITY
710  0000:046E  B1 14                     MOV   CL,20               ;PRELOAD DIVIDE VALUE
711  0000:0470  F6 06 59 20 01    MOTI03: TEST  ADFLAG,1            ;ANALOG READY?
712  0000:0475  74 F9                     JZ    MOTI03              ;IF NOT
713  0000:0477  A1 46 20                  MOV   AX,ADVAL            ;GET CURRENT VELOCITY
714  0000:047A  F6 C4 08                  TEST  AH,8                ;MINUS?
715  0000:047D  74 02                     JZ    MOTI04              ;IF MINUS
716  0000:047F  F7 D8                     NEG   AX                  ;MAKE POSITIVE
717  0000:0481  25 FF 0F          MOTI04: AND   AX,0FFFH            ;STRIP UNWANTED BITS
718  0000:0484  F6 F1                     DIV   CL                  ;DIVIDE BY 20
719  0000:0486  B4 00                     MOV   AH,0                ;KILL REMAINDER
720  0000:0488  A3 28 20                  MOV   VEL1,AX             ;CURRENT = PREVIOUS
721  0000:048B  03 C2                     ADD   AX,DX               ;ADD IN PREVIOUS VELOCITY
722  0000:048D  83 C3 04                  ADD   BX,4                ;4 COUNT SLOP
723  0000:0490  3B C3                     CMP   AX,BX               ;OK MOTION?
724  0000:0492  7E 0B                     JLE   MOTI05              ;IF OK
725  0000:0494  C6 06 5E 20 01            MOV   DRVOFF,1
726  0000:0499  81 0E 08 20 00 04         OR    WORD PTR FAUL1,1024 ;SET MOTION FAULT
727  0000:049F  E9 75 FF          MOTI05: JMP   TMR2EX
```

```
728
729  0000:04A2                      DSTOR:
730  0000:04A2  32 C0                       XOR     AL,AL
731  0000:04A4  BA 80 01                    MOV     DX,APEXRQ               ;GET APEX DATA
732  0000:04A7  EE                          OUT     DX,AL
733  0000:04A8  83 C2 02                    ADD     DX,2                    ;POINT TO APEXRS
734  0000:04AB  B9 0C 00                    MOV     CX,12                   ;12 TRIALS
735  0000:04AE                      DSTOR1:
736  0000:04AE  EC                          IN      AL,DX                   ;SEE IF DATA READY
737  0000:04AF  24 C0                       AND     AL,0C0H                 ;STRIP DATA
738  0000:04B1  3C 80                       CMP     AL,80H                  ;LOOK FOR PATTERN
739  0000:04B3  74 0A                       JZ      DSTOR2                  ;IF READY
740  0000:04B5  E2 F7                       LOOP    DSTOR1                  ;IF NOT
741  0000:04B7  80 0E 08 20 01              OR      FAUL1,1                 ;SET APEX DEAD
742  0000:04BC  E9 30 00                    JMP     DSTOR3                  ;SKIP APEX GATHERING
743  0000:04BF                      DSTOR2:
744  0000:04BF  80 26 08 20 FE              AND     FAUL1,0FEH              ;SET APEX OK
745  0000:04C4  4A                          DEC     DX
746  0000:04C5  ED                          IN      AX,DX                   ;GET POSITION
747  0000:04C6  8B D8                       MOV     BX,AX                   ;PUT INTO BX
748  0000:04C8  C1 E3 02                    SHL     BX,2                    ;GET RID OF RESPONSE BITS
749  0000:04CB  83 C2 02                    ADD     DX,2                    ;POINT TO NEXT SECTION
750  0000:04CE  ED                          IN      AX,DX                   ;GET NEXT INFO
751  0000:04CF  C0 C4 02                    ROL     AH,2                    ;GET FINAL POSITION
752  0000:04D2  80 E4 03                    AND     AH,03                   ;PREPARE TO MERGE
753  0000:04D5  0A DC                       OR      BL,AH                   ;MERGE BITS
754  0000:04D7  89 1E 02 20                 MOV     POSD,BX                 ;SAVE POSITION
755  0000:04DB  A1 00 20                    MOV     AX,POSCEC
756  0000:04DE  2B C3                       SUB     AX,BX                   ;GET ERROR
757  0000:04E0  C6 06 62 20 00              MOV     DIR,0                   ;ASSUME POSITIVE
758  0000:04E5  73 05                       JNC     DSTO2A                  ;IF POSITIVE
759  0000:04E7  C6 06 62 20 01              MOV     DIR,1                   ;SET NEGATIVE
760  0000:04EC  A3 04 20            DSTO2A: MOV     ERROR,AX                ;STORE IT HERE
761  0000:04EF  C3                  DSTOR3: RET
762
763
764  0000:04F0                      BRKOFF:
765  0000:04F0  E5 00                       IN      AX,P1PTA                ;GET CONTROL PORT
766  0000:04F2  25 FF BF                    AND     AX,0BFFFH               ;ENGAGE THE BRAKE
767  0000:04F5  E7 00                       OUT     P1PTA,AX                ;DO IT
768  0000:04F7  C6 06 5C 20 00              MOV     TFLAG1,0                ;SET TIMER FLAG = OK
769  0000:04FC  E8 26 00                    CALL    SEC1                    ;1 SEC
770  0000:04FF  E8 A0 FF            BRKOF1: CALL    DSTOR                   ;GET UPDATE
771  0000:0502  BA 87 01                    MOV     DX,DISCR                ;POINT TO BRAKE DISCRETES
772  0000:0505  EC                          IN      AL,DX                   ;GET INFO
773  0000:0506  A8 10                       TEST    AL,16                   ;BRAKE FREE?
774  0000:0508  74 12                       JZ      BRKOF2                  ;IF FREE CONTINUE
775  0000:050A  F6 06 5C 20 01              TEST    TFLAG1,1                ;TIMED OUT?
776  0000:050F  74 EE                       JZ      BRKOF1                  ;IF NOT
777  0000:0511  81 0E 08 20 00 01           OR      WORD PTR FAUL1,256      ;SET BRAKE FAULT
778  0000:0517  EA 00 00 00 00              JMP     FAR RSCMD               ;RESET COMMAND
779  0000:051C                      BRKOF2:
780  0000:051C  81 26 08 20 FF FE           AND     WORD PTR FAUL1,0FEFFH   ;CLEAR BRAKE FAULT
781  0000:0522  E9 0F 00                    JMP     TMROFF                  ;STOP TIMER
782
783  0000:0525                      SEC1:
784  0000:0525  BA 5E FF                    MOV     DX,TIMODE               ;ENABLE TIMER1 FIRST
```

```
0000:0528  B8 09 E0          MOV   AX,0E009H
0000:052B  EF                OUT   DX,AX
0000:052C  BA 66 FF          MOV   DX,T2MODE          ;NOW TIMER2
0000:052F  B8 01 C0          MOV   AX,0C001H
0000:0532  EF                OUT   DX,AX
0000:0533  C3                RET

0000:0534                TMROFF:
0000:0534  BE 76 05          MOV   SI,OFFSET CS:TSETUP  ;POINT TO TABLE
0000:0537  BA 50 FF          MOV   DX,TOCOUT          ;POINT TO SOURCE POINTER
0000:053A  B9 0C 00          MOV   CX,12
0000:053D  6F                TSET: OUTSW
0000:053E  83 C2 02          ADD   DX,2
0000:0541  E2 FA             LOOP  TSET
0000:0543  C3                RET
0000:0544

                         ;       TABLE SECTION FOR BOUND

0000:0544                CLSETAB:
0000:0544  FCFF                  .WORD  0FFFCH        ;4 COUNTS BEFORE WE MOVE
0000:0546  0400                  .WORD  00004H

0000:0548                CMDRG:
0000:0548  3000                  .WORD  48
0000:054A  3400                  .WORD  52

0000:054C                CMDTBL:
0000:054C  EA00                  .WORD  POSCMD
0000:054E  2101                  .WORD  NAPCMD
0000:0550  4201                  .WORD  MANCMD
0000:0552  5C01                  .WORD  BDSCMD
0000:0554  1A01                  .WORD  RECMD

0000:0556                DSETUP:
0000:0556  8B01                  .WORD  RELADD
0000:0558  0000                  .WORD  0
0000:055A  4020                  .WORD  RAC
0000:055C  0000                  .WORD  0
0000:055E  0400                  .WORD  4
0000:0560  66A7                  .WORD  DOVAL

0000:0562                MONRG:
0000:0562  3000                  .WORD  48
0000:0564  4F00                  .WORD  79

0000:0566                POSRG:
0000:0566  008A                  .WORD  LOW           ;SOFTWARE LIMITS
0000:0568  0072                  .WORD  HIGH

0000:056A                RAMPLM:
0000:056A  00BA                  .WORD  0BA00H        ;RAMP LIMITS
0000:056C  0046                  .WORD  04600H

0000:056E  D302          RAMPTL: .WORD  ZIPUP         ;RAMP ENTRY
0000:0570  3D03                  .WORD  MAINNY
0000:0572  6E03                  .WORD  ZIPDWN
0000:0574  2102                  .WORD  VERGIT
```

```
842
843   0000:0576                  TSSTOP:
844   0000:0576   0000                  .WORD   0
845   0000:0578   0000                  .WORD   0
846   0000:057A   0000                  .WORD   0
847   0000:057C   0040                  .WORD   16384           ;TIMER 0 IS NOT USED
848   0000:057E   0000            TSET1:.WORD   0
849   0000:0580   D007                  .WORD   2000            ;1 SEC DELAY
850   0000:0582   0000                  .WORD   0
851   0000:0584   0040                  .WORD   16384           ;TIMER 1 IS READY TO GO
852   0000:0586   0000                  .WORD   0
853   0000:0588   B202                  .WORD   690             ;552uS TIMER
854   0000:058A   0000                  .WORD   0
855   0000:058C   0040                  .WORD   16384           ;TIMER 2 IS READY ALSO
856
857   0000:058E                         END
```

Lines Assembled : 857 Assembly Errors : 0

2500 A.D. 80186 Cross Assembler - Version 4.00g

Input Filename : RS.asm
Output Filename : RS.obj

```
1
2                                   ;       ROTATION AXIS FOR VLBA ANTENNAS
3                                   ;       WRITTEN BY:     WAYNE M. KOSKI
4                                   ;       LAST REVISION:  DECEMBER 01, 1989
5                                   ;
6                                   ;       THINGS TO DO AND GENERAL NOTES:
7                                   ;
8                                   ;         1.  THIS SECTION OF CODE WILL BE THE MAIN RUNNING
9                                   ;             ROUTINES AND SHALL EXCLUDE THE INTERRUPT ROUTINES
10                                  ;             WHICH SHALL BE IN THE OTHER EPROM.
11                                  ;         2.  ROTATION AXIS SHOULD LOOK SIMULAR TO FOCUS AT
12                                  ;             PRESENT.
13                                  ;         3.  THIS VERSION UPDATES TO THE NEW F/R CONTROLLER MODULE.
14  0000:0000
15                                          ASSUME  CS:CODE, DS:DSEG
16
17                                      .OUTPUT 2500AD
18                                      .OPTIONS H
19
20
21                                  ;       80188 INTERNAL PORTS FOR CONTROL AND GUIDANCE
22
23
24
25                                  ;       80188 INTERRUPT CONTROL / STATUS REGISTARS
26
27              0000:FF22           EOI:    .EQUAL  0FF22H          ;END OF INTERRUPT REGISTAR
28              0000:FF24           POLL:   .EQUAL  0FF24H          ;INTERRUPT POLL REGISTAR
29              0000:FF26           POLLS:  .EQUAL  0FF26H          ;INTERRUPT POLL STATUS REGISTAR
30              0000:FF28           MASK:   .EQUAL  0FF28H          ;INTERRUPT MASK REGISTAR
31              0000:FF2A           PMASK:  .EQUAL  0FF2AH          ;INTERRUPT PRIORITY MASK REGISTAR
32              0000:FF2C           ISR:    .EQUAL  0FF2CH          ;INTERRUPT IN SERVICE REGISTAR
33              0000:FF2E           IRR:    .EQUAL  0FF2EH          ;INTERRUPT REQUEST REGISTAR
34              0000:FF30           ICSR:   .EQUAL  0FF30H          ;INTERRUPT CONTROL STATUS REGISTAR
35              0000:FF32           ITCR:   .EQUAL  0FF32H          ;INTERRUPT TIMER CONTROL REGISTAR
36              0000:FF34           IDOCR:  .EQUAL  0FF34H          ;INTERRUPT DMA 0 CONTROL REGISTAR
37              0000:FF36           ID1CR:  .EQUAL  0FF36H          ;INTERRUPT DMA 1 CONTROL REGISTAR
38              0000:FF38           INT0CR: .EQUAL  0FF38H          ;INT 0 CONTROL REGISTAR
39              0000:FF3A           INT1CR: .EQUAL  0FF3AH          ;INT 1 CONTROL REGISTAR
40              0000:FF3C           INT2CR: .EQUAL  0FF3CH          ;INT 2 CONTROL REGISTAR
41              0000:FF3E           INT3CR: .EQUAL  0FF3EH          ;INT 3 CONTROL REGISTAR
42
43
```

```
                        ;       80188 TIMER CONTROL REGISTARS

0000:FF50       T0COUT: .EQUAL  0FF50H          ;TIMER 0 COUNT REGISTAR
0000:FF52       T0MAXA: .EQUAL  0FF52H          ;TIMER 0 MAXIMUM COUNT A REGISTAR
0000:FF54       T0MAXB: .EQUAL  0FF54H          ;TIMER 0 MAXIMUM COUNT B REGISTAR
0000:FF56       T0MODE: .EQUAL  0FF56H          ;TIMER 0 MODE REGISTAR
0000:FF58       T1COUT: .EQUAL  0FF58H          ;TIMER 1 COUNT REGISTAR
0000:FF5A       T1MAXA: .EQUAL  0FF5AH          ;TIMER 1 MAXIMUM COUNT A REGISTAR
0000:FF5C       T1MAXB: .EQUAL  0FF5CH          ;TIMER 1 MAXIMUM COUNT B REGISTAR
0000:FF5E       T1MODE: .EQUAL  0FF5EH          ;TIMER 1 MODE REGISTAR
0000:FF60       T2COUT: .EQUAL  0FF60H          ;TIMER 2 COUNT REGISTAR
0000:FF62       T2MAXA: .EQUAL  0FF62H          ;TIMER 2 MAXIMUM COUNT A REGISTAR
0000:FF66       T2MODE: .EQUAL  0FF66H          ;TIMER 2 MODE REGISTAR


                        ;       80188 CHIP SELECT CONTROL REGISTARS

0000:FFA0       UMCS:   .EQUAL  0FFA0H          ;80188 INTERNAL UPPER MEMORY CHIP
                                                ;SELECT CONTROL BLOCK REGISTAR
0000:FFA2       LMCS:   .EQUAL  0FFA2H          ;80188 INTERNAL LOWER MEMORY CHIP
                                                ;SELECT CONTROL BLOCK REGISTAR
0000:FFA4       PACS:   .EQUAL  0FFA4H          ;80188 INTERNAL PERIPHERAL CHIP
                                                ;SELECT CONTROL BLOCK REGISTAR
0000:FFA6       MMCS:   .EQUAL  0FFA6H          ;80188 INTERNAL MIDDLE MEMORY
                                                ;START ADDRESS REGISTAR
0000:FFA8       MPCS:   .EQUAL  0FFA8H          ;80188 INTERNAL MIDDLE MEMORY CHIP
                                                ;SELECT CONTROL BLOCK REGISTAR


                        ;       80188 DMA CHANNEL CONTROL REGISTARS

0000:FFC0       D0SPL:  .EQUAL  0FFC0H          ;DMA 0 SOURCE POINTER LSB REGISTAR
0000:FFC2       D0SPM:  .EQUAL  0FFC2H          ;DMA 0 SOURCE POINTER MSB REGISTAR
0000:FFC4       D0DPL:  .EQUAL  0FFC4H          ;DMA 0 DESTINATION POINTER LSB REGISTAR
0000:FFC6       D0DPM:  .EQUAL  0FFC6H          ;DMA 0 DESTINATION POINTER MSB REGISTAR
0000:FFC8       D0TC:   .EQUAL  0FFC8H          ;DMA 0 TRANSFER COUNT REGISTAR
0000:FFCA       D0MODE: .EQUAL  0FFCAH          ;DMA 0 MODE REGISTAR
0000:FFD0       D1SPL:  .EQUAL  0FFD0H          ;DMA 1 SOURCE POINTER LSB REGISTAR
0000:FFD2       D1SPM:  .EQUAL  0FFD2H          ;DMA 1 SOURCE POINTER MSB REGISTAR
0000:FFD4       D1DPL:  .EQUAL  0FFD4H          ;DMA 1 DESTINATION POINTER LSB REGISTAR
0000:FFD6       D1DPM:  .EQUAL  0FFD6H          ;DMA 1 DESTINATION POINTER MSB REGISTAR
0000:FFD8       D1TC:   .EQUAL  0FFD8H          ;DMA 1 TRANSFER COUNT REGISTAR
0000:FFDA       D1MODE: .EQUAL  0FFDAH          ;DMA 1 MODE REGISTAR


                        ;       80188 INTERNAL I/O RELOCATION REGISTAR

0000:FFFE       RELOC:  .EQUAL  0FFFEH          ;I/O RELOCATION REGISTAR


                        ;       80188 INITIAL VALUES FOR INTERNAL REGISTARS

0000:007D       LMBS:   .EQUAL  007DH           ;LOWER MEMORY BLOCK SIZE = 2K
0000:81BD       MMBS:   .EQUAL  81BDH           ;MIDDLE MEMORY BLOCK SIZE = 8K
0000:03FD       MMST:   .EQUAL  03FDH           ;MIDDLE MEMORY START POSITION = 8K
0000:003D       PST:    .EQUAL  003DH           ;PERIPHERAL START ADDRESS = 0
0000:FFBD       UMBS:   .EQUAL  0FFBDH          ;UPPER MEMORY BLOCK SIZE = 2K
```

```
101
102
103
104                                     ;       EXTERNAL PORTS FOR CONTROLLING THE ROTATION AXIS
105
106                     0000:0000       P1PTA:  .EQUAL  0               ;PROM 1 PORT A
107                     0000:0001       P1PTB:  .EQUAL  1               ;PROM 1 PORT B
108                     0000:0002       P1PTAD: .EQUAL  2               ;PROM 1 PORT A DIRECTION
109                     0000:0003       P1PTBD: .EQUAL  3               ;PROM 1 PORT B DIRECTION
110                     0000:0080       P2PTA:  .EQUAL  128             ;PROM 2 PORT A
111                     0000:0081       P2PTB:  .EQUAL  129             ;PROM 2 PORT B
112                     0000:0082       P2PTAD: .EQUAL  130             ;PROM 2 PORT A DIRECTION
113                     0000:0083       P2PTBD: .EQUAL  131             ;PROM 2 PORT B DIRECTION
114                     0000:0100       RAMTC:  .EQUAL  256             ;RAM TIMER AND CONTROL
115                     0000:0101       RAMPTA: .EQUAL  257             ;RAM PORT A
116                     0000:0102       RAMPTB: .EQUAL  258             ;RAM PORT B
117                     0000:0103       RAMPTC: .EQUAL  259             ;RAM PORT C
118                     0000:0104       RAMTLO: .EQUAL  260             ;RAM TIMER LOW
119                     0000:0105       RAMTHI: .EQUAL  261             ;RAM TIMER HIGH
120                     0000:0180       APEXRQ: .EQUAL  384             ;APEX REQ
121                     0000:0182       APEXRS: .EQUAL  386             ;APEX RESPONSE
122                     0000:0182       POSM:   .EQUAL  386             ;POSITION MSB
123                     0000:0181       POSL:   .EQUAL  385             ;POSITION LSB
124                     0000:0184       VELM:   .EQUAL  388             ;VELOCITY MSB
125                     0000:0183       VSLL:   .EQUAL  387             ;VELOCITY LSB
126                     0000:0186       ANAH:   .EQUAL  390             ;ANALOGS MSB
127                     0000:0185       ANAL:   .EQUAL  389             ;ANALOGS LSB
128                     0000:0187       DISCR:  .EQUAL  391             ;FOC DISCRETES
129                     0000:018B       RELADD: .EQUAL  395             ;GET RELATIVE ADDRESS
130                     0000:018C       COMML:  .EQUAL  396             ;CONTROL VALUE LSB
131                     0000:018D       COMMM:  .EQUAL  397             ;CONTROL VALUE MSB
132                     0000:018E       DEVACK: .EQUAL  398             ;DEVICE ACKNOWLEDGE
133                     0000:018F       MONL:   .EQUAL  399             ;MONITOR DATA MSB
134                     0000:0190       MONM:   .EQUAL  400             ;MONITOR DATA LSB
135                     0000:0191       ADL:    .EQUAL  401             ;READ A/D LSB
136                     0000:0192       ADH:    .EQUAL  402             ;READ A/D MSB
137                     0000:0193       SEMCT:  .EQUAL  403             ;SELECT MOTOR CURRENT/TORQUE
138                     0000:0194       STCNV:  .EQUAL  404             ;START A/D CONVERT
139                     0000:0195       MODESW: .EQUAL  405             ;READ MODE SWITCH
140                     0000:0196       LTCHDA: .EQUAL  406             ;LATCH DRIVE D/A
141                     0000:0197       BDERL1: .EQUAL  407             ;BDS3 ERROR LSB #1
142                     0000:0198       BDERM1: .EQUAL  408             ;BDS3 ERROR MSB #1
143
144
145     0000:0000                       DSEG:   SEGMENT
146
147                                     ;       DATA STORAGE
148
149     0000:2000                               ORG     2000H
150
151                     0000:2000       MEMST:  .EQUAL  $
152
153                                     ;       MONITOR STORAGE
154
155     0000:2000                       POSCEC: .BLKB   2               ;POSITION COMMAND ECHO
156     0000:2002                       POSD:   .BLKB   2               ;CURRENT POSITION DATA
157     0000:2004                       ERROR:  .BLKB   2               ;POSCEC-POSD
```

```
158  0000:2006                SYSTEM: .BLKB  2          ;SYSTEM PARAMETERS
159  0000:2008                FAUL1:  .BLKB  2          ;FAULT BITS SET 1
160  0000:200A                FAUL2:  .BLKB  2          ;FAULT BITS SET 2
161  0000:200C                ANAFL:  .BLKB  2          ;ANALOG FAULT FLAGS
162
163           0000:200E       ANADT:  .EQUAL $          ;ANALOG STORAGE IS HERE
164
165  0000:200E                GND1:   .BLKB  2          ;GND
166  0000:2010                GND2:   .BLKB  2          ;GND
167  0000:2012                GND3:   .BLKB  2          ;GND
168  0000:2014                VEL:    .BLKB  2          ;ROTATION VELOCITY
169  0000:2016                V15P:   .BLKB  2          ;+15V/2
170  0000:2018                V15N:   .BLKB  2          ;-15V/2
171  0000:201A                V5:     .BLKB  2          ;+5V
172  0000:201C                V10P:   .BLKB  2          ;+10V
173  0000:201E                MTEMP1: .BLKB  2          ;MOUNT TEMP 1
174  0000:2020                V10N:   .BLKB  2          ;-10V
175  0000:2022                MTEMP2: .BLKB  2          ;MOUNT TEMP 2
176  0000:2024                BTEMP:  .BLKB  2          ;BIN TEMP
177
178  0000:2026                SERVER: .BLKB  2          ;SERIAL/VERSION
179  0000:2028                VEL1:   .BLKB  2
180  0000:202A                X22:    .BLKB  2
181  0000:202C                X23:    .BLKB  2
182  0000:202E                X24:    .BLKB  2
183  0000:2030                X25:    .BLKB  2
184  0000:2032                X26:    .BLKB  2
185  0000:2034                X27:    .BLKB  2
186  0000:2036                X28:    .BLKB  2
187  0000:2038                X29:    .BLKB  2
188  0000:203A                X30:    .BLKB  2
189  0000:203C                X31:    .BLKB  2
190  0000:203E                X32:    .BLKB  2
191
192
193                           ;       TEMPORARY STORAGE
194
195  0000:2040                RAC:    .BLKB  1          ;RELATIVE CONTROL ADDRESS
196  0000:2041                CONL:   .BLKB  1          ;CONTROL VALUE LSB
197  0000:2042                CONM:   .BLKB  1          ;CONTROL VALUE MSB
198  0000:2043                ACKF:   .BLKB  1          ;ACKNOWLEDGE FLAG
199  0000:2044                CONTMP: .BLKB  2          ;TEMPORARY COMMAND STORAGE
200  0000:2046                ADVAL:  .BLKB  2          ;A/D TEMPORARY STORAGE
201  0000:2048                SPEED:  .BLKB  1          ;RAMP LEVEL
202  0000:2049                BREAK1: .BLKB  2          ;RAMP UP BREAK POINT
203  0000:204B                BREAK2: .BLKB  2          ;RAMP DOWN BREAK POINT
204  0000:204D                POSOLD: .BLKB  2          ;OLD POSITION
205  0000:204F                EXTTMR: .BLKB  1          ;EXTERNAL TROUBLE TIMER
206  0000:2050                RANDOM: .BLKB  2          ;RANDOM NUMBER STORAGE
207
208                           ;       FLAGS
209
210           0000:2052       FLAGST: .EQUAL $
211
212  0000:2052                RESCMD: .BLKB  1
213  0000:2053                NAPATV: .BLKB  1
214  0000:2054                NAPREQ: .BLKB  1
```

```
0000:2055                       DRVREQ: .BLKB   1
0000:2056                       MANOVR: .BLKB   1
0000:2057                       SDSRST: .BLKB   1
0000:2058                       DRVONE: .BLKB   1
0000:2059                       ADFLAG: .BLKB   1
0000:205A                       BFLAG:  .BLKB   1
0000:205B                       TFLAG0: .BLKB   1
0000:205C                       TFLAG1: .BLKB   1
0000:205D                       TFLAG2: .BLKB   1
0000:205E                       DRVOFF: .BLKB   1
0000:205F                       DRVATV: .BLKB   1
0000:2060                       RAMPOS: .BLKB   2
0000:2062                       DIR:    .BLKB   1
0000:2063                       LDIR:   .BLKB   1

            0000:0012           ENDFLG: .EQUAL  $-FLAGST

                                        ENDS


                                ;       PROGRAM EQUATES

            0000:2100           STCK:   .EQUAL  2100H               ;STACK LOCATION
            0000:A766           DOVAL:  .EQUAL  0A766H              ;DMA 0 CONTROL VALUE
            0000:00E8           MASKV:  .EQUAL  00E8H               ;DMA 0, INT0, TIMER ENABLE
            0000:0A00           LOW:    .EQUAL  00A00H              ;LOW VALUE
            0000:F200           HIGH:   .EQUAL  0F200H              ;HIGH VALUE
            0000:0004           CONRAM: .EQUAL  4                   ;STARTING RAMP VALUE

                                ;       GLOBAL LIST

                                        GLOBAL  INITAL, RSCMD
                                        EXTERNAL DSETUP:WORD, TSETUP:WORD, TSET1:WORD

                                ;       PROGRAM ENTERS HERE AFTER RESET ROUTINE (AS IF BY MAGIC)

FF80:0000                               ORG     0FF80:0000H


FF80:0000                       INITAL:
FF80:0000   BC 00 21                    MOV     SP,STCK             ;SET STACK POINTER

                                ;       LETS INITIALIZE MORE 80188 CHIP SELECT LOGIC

FF80:0003   BA A2 FF                    MOV     DX,LMCS             ;SET UP LOWER MEMORY BLOCK
FF80:0006   B8 7D 00                    MOV     AX,LMBS             ;TO 2K
FF80:0009   EF                          OUT     DX,AX
FF80:000A   BA A8 FF                    MOV     DX,MPCS             ;NOW SET MIDDLE MEMORY BLOCK TO
FF80:000D   B8 BD 81                    MOV     AX,MMBS             ;2K AND I/O MAPPED PCS LINES
FF80:0010   EF                          OUT     DX,AX
FF80:0011   BA A6 FF                    MOV     DX,MMCS             ;SET MIDDLE MEMORY TO START AT
FF80:0014   B8 FD 03                    MOV     AX,MMST             ;8K
FF80:0017   EF                          OUT     DX,AX
FF80:0018   BA A4 FF                    MOV     DX,PACS             ;SET UP THE PERIPHERAL ADDRESS
FF80:001B   B8 3D 00                    MOV     AX,PST              ;TO START AT 0
FF80:001E   EF                          OUT     DX,AX
```

```
272                             ;       LETS INITIALIZE PORTS AND PORT VALUES
273
274  FF80:001F  B8 FF FF                MOV     AX,0FFFFH               ;EPROM 1 PORT A, B = OUTPUT
275  FF80:0022  E7 02                   OUT     P1PTAD,AX
276  FF80:0024  B8 FF 07                MOV     AX,07FFH                ;UPPER NYBBLE = INPUT
277  FF80:0027  E7 82                   OUT     P2PTAD,AX
278  FF80:0029  BA 00 01                MOV     DX,RAMTC                ;NOW DO RAM'S I/O
279  FF80:002C  EE                      OUT     DX,AL                   ;SET TO OUTPUT
280  FF80:002D  33 C0                   XOR     AX,AX                   ;CLEAR ACC
281  FF80:002F  E7 00                   OUT     P1PTA,AX                ;CLEAR EPROM 1 PORTS A AND B
282  FF80:0031  E7 80                   OUT     P2PTA,AX                ;CLEAR EPROM 2 PORTS A AND B
283  FF80:0033  42                      INC     DX                      ;POINT TO RAM PORT A
284  FF80:0034  EF                      OUT     DX,AX                   ;CLEAR RAM PORTS A AND B
285  FF80:0035  42                      INC     DX
286  FF80:0036  EF                      OUT     DX,AX                   ;CLEAR PORT C TOO
287  FF80:0037  BA 93 01                MOV     DX,SEMCT                ;CLEAR LIGHTS
288  FF80:003A  EE                      OUT     DX,AL
289
290                             ;       NOW LETS SET UP THE DMA 0 CHANNEL
291
292  FF80:003B  BE 00 00                MOV     SI,OFFSET DSETUP        ;POINT TO TABLE
293  FF80:003E  BA C0 FF                MOV     DX,DOSPL                ;SET UP SOURCE POINTER
294  FF80:0041  B9 06 00                MOV     CX,6
295  FF80:0044  6F              DSET:   OUTSW
296  FF80:0045  83 C2 02                ADD     DX,2
297  FF80:0048  E2 FA                   LOOP    DSET
298
299                             ;       NOW LETS INITIALIZE THE TIMERS
300
301  FF80:004A  E8 1F 06                CALL    TMROFF
302
303                             ;       NOW LETS ZERO MEMORY
304
305  FF80:004D  32 C0                   XOR     AL,AL
306  FF80:004F  B9 00 01                MOV     CX,256                  ;256 LOCATIONS
307  FF80:0052  BF 00 20                MOV     DI,OFFSET MEMST         ;START OF INDEX
308  FF80:0055  F3 AA                   REP STOS BYTE PTR MEMST
309
310
311                             ;       NOW LETS TURN ON THE INTERRUPTS
312
313  FF80:0057  BA 28 FF                MOV     DX,MASK                 ;LETS TURN ON DMA, INT0, TIMER
314  FF80:005A  B8 E8 00                MOV     AX,MASKV                ;INTERRUPTS
315  FF80:005D  EF                      OUT     DX,AX
316  FF80:005E  BA 38 FF                MOV     DX,INT0CR               ;SET INT0 PRIORITY
317  FF80:0061  B8 00 00                MOV     AX,0                    ;HIGHEST PRIORITY
318  FF80:0064  EF                      OUT     DX,AX
319  FF80:0065  FB                      STI
320
321                             ;       LETS RESET THE BDS3 SERVO AMPLIFIERS
322
323  FF80:0066  E8 72 05                CALL    BDS3RS                  ;INITIATE RESET
324  FF80:0069
325                             ;       LETS JUST WAIT FOR THREE SECONDS FOR THINGS TO SETTLE
326
327  FF80:0069  B9 03 00                MOV     CX,3                    ;THREE SECONDS TOTAL
328  FF80:006C  E8 DC 05                CALL    DELAY
```

```
329
330                             ;       NOW LETS WAIT FOR THE THREE PHASE
331
332  FF80:006F                  PHASE:
333  FF80:006F  B8 00 80                MOV     AX,8000H            ;TURN ON 3 PHASE
334  FF80:0072  E7 00                   OUT     P1PTA,AX
335  FF80:0074  B9 03 00                MOV     CX,3                ;THREE SECONDS TOTAL
336  FF80:0077  C6 06 5C 20 00  PH301:  MOV     TFLAG1,0            ;SET TIMER FLAG = OK
337  FF80:007C  BA 97 01                MOV     DX,BDERL1           ;GET BDS3 INFO
338  FF80:007F  ED              PH302:  IN      AX,DX
339  FF80:0080  F7 D0                   NOT     AX                  ;INVERT
340  FF80:0082  25 01 01                AND     AX,0101H            ;TEST FOR BUS FAULTS
341  FF80:0085  74 0F                   JZ      THREX               ;EXIT IF OK
342  FF80:0087  F6 06 5C 20 01          TEST    TFLAG1,1            ;TIME OUT?
343  FF80:008C  74 F1                   JZ      PH302               ;LOOP UNTIL TIME OUT
344  FF80:008E  E2 E7                   LOOP    PH301               ;UNTIL THREE SEC HAVE PASSED
345  FF80:0090  A3 0A 20                MOV     FAUL2,AX            ;SET BDS3 DEAD FLAG
346  FF80:0093  E9 D9 FF                JMP     PHASE               ;IF BUS FAULT
347
348  FF80:0096                  THREX:
349  FF80:0096  E8 D3 05                CALL    TMROFF              ;STOP TIMERS
350  FF80:0099  C7 06 0A 20 00 00       MOV     WORD PTR FAUL2,0    ;CLEAR BDS3 DEAD FLAG
351
352
353                             ;       SERIAL NUMBER / REVISION
354
355  FF80:009F  C7 06 26 20 03 06       MOV     WORD PTR SERVER,0603H
356
357                             ;       LETS DO ONE DSTOR PRIOR TO ENTRY
358
359  FF80:00A5  E8 09 04                CALL    DSTOR
360
361
362                             ;       HERE WE ARE INTO THE MAIN PROGRAM
363
364
365  FF80:00A8                  BOSS:
366  FF80:00A8  8B 0E 50 20             MOV     CX,RANDOM           ;GET RANDOM DELAY
367  FF80:00AC  E2 FE           WAIT:   LOOP    WAIT                ;DELAY
368  FF80:00AE  BC 00 21                MOV     SP,STCK             ;LETS SET UP STACK POINTER
369  FF80:00B1  F6 06 08 20 01          TEST    FAUL1,1             ;APEX OK?
370  FF80:00B6  74 03                   JZ      NEXT                ;IF OK
371  FF80:00B8  E9 90 00                JMP     RSCMD               ;IF APEX BROKE
372  FF80:00BB                  NEXT:
373  FF80:00BB  F6 06 5F 20 01          TEST    DRVATV,1            ;DRIVE ACTIVE?
374  FF80:00C0  74 06                   JZ      NEXT0               ;IF NOT
375  FF80:00C2  E8 B8 00                CALL    CHKDRV              ;IF ACTIVE
376  FF80:00C5  E9 E0 FF                JMP     BOSS
377  FF80:00C8                  NEXT0:
378  FF80:00C8  E8 E6 03                CALL    DSTOR               ;LETS GET APEX DATA
379  FF80:00CB  F6 06 56 20 01          TEST    MANOVR,1            ;MANUAL OVERRIDE?
380  FF80:00D0  74 07                   JZ      NEXT1A              ;IF NOT OVER RIDE
381  FF80:00D2  80 0E 06 20 08          OR      SYSTEM,8            ;MANUAL OVER RIDE FLAG
382  FF80:00D7  EB 09                   JMP     SHORT NEXT1
383  FF80:00D9  E4 81           NEXT1A: IN      AL,P2PTB            ;IN LOCAL?
384  FF80:00DB  24 20                   AND     AL,20H
385  FF80:00DD  75 03                   JNZ     NEXT1               ;IF NOT
```

```
386  FF80:00DF  E9 71 02              JMP   LOCAL                ;IF LOCAL
387  FF80:00E2                  NEXT1:
388  FF80:00E2  F6 06 57 20 01        TEST  BDSRST,1             ;RESET BDS3 SERVO AMP?
389  FF80:00E7  74 14                 JZ    NEXT2                ;IF NOT
390  FF80:00E9  C6 06 57 20 00,       MOV   BDSRST,0             ;RESET REQUEST
391  FF80:00EE  E8 EA 04              CALL  BDS3RS               ;DO IT
392  FF80:00F1  B9 01 00              MOV   CX,1                 ;WAIT 1 SEC
393  FF80:00F4  E8 54 05              CALL  DELAY
394  FF80:00F7  E8 72 05              CALL  TMROFF               ;STOP TIMERS
395  FF80:00FA  E9 AB FF              JMP   BOSS
396  FF80:00FD                  NEXT2:
397  FF80:00FD  F6 06 52 20 01        TEST  RESCMD,1             ;SOFT RESET?
398  FF80:0102  74 03                 JZ    NEXT3                ;IF NOT
399  FF80:0104  E9 38 00              JMP   CMDRS                ;IF RESET
400  FF80:0107                  NEXT3:
401  FF80:0107  F6 06 53 20 01        TEST  NAPATV,1             ;IN NAP?
402  FF80:010C  74 08                 JZ    NEXT4                ;IF NOT
403  FF80:010E  C6 06 55 20 00        MOV   DRVREQ,0             ;IF NAP, KILL REQ
404  FF80:0113  E9 92 FF              JMP   BOSS                 ;IF NAP, GOTO BOSS
405  FF80:0116                  NEXT4:
406  FF80:0116  F6 06 55 20 01        TEST  DRVREQ,1             ;DRIVE REQUEST?
407  FF80:011B  74 06                 JZ    NEXT5                ;IF NO DRIVE REQUEST
408  FF80:011D  E8 B0 00              CALL  DRVINT               ;IF DRIVE REQUEST
409  FF80:0120  E9 85 FF              JMP   BOSS
410  FF80:0123                  NEXT5:
411  FF80:0123  F6 06 54 20 01        TEST  NAPREQ,1             ;NAP REQUEST?
412  FF80:0128  75 03                 JNZ   NEXT6
413  FF80:012A  E9 7B FF              JMP   BOSS                 ;IF NO NAP REQUEST
414  FF80:012D                  NEXT6:
415  FF80:012D  C6 06 54 20 00        MOV   NAPREQ,0             ;REMOVE NAP REQUEST
416  FF80:0132  C6 06 53 20 FF        MOV   NAPATV,0FFH          ;SET NAP ACTIVE
417  FF80:0137  80 0E 06 20 01        OR    SYSTEM,1             ;SET NAP FLAG
418  FF80:013C  E9 69 FF              JMP   BOSS                 ;END OF MAIN ROUTINE
419
420  FF80:013F                  CMDRS:
421  FF80:013F  81 26 08 20 1F FC     AND   WORD PTR FAUL1,0FC1FH ;CLEAR CERTAIN FAULTS
422  FF80:0145  81 26 06 20 FE FF     AND   WORD PTR SYSTEM,0FFFEH ;CLEAR  NAP
423  FF80:014B                  RSCMD:
424  FF80:014B  81 26 06 20 F5 FF     AND   WORD PTR SYSTEM,0FFF5H ;CLEAR LOCAL, OVER RIDE
425  FF80:0151  BC 00 21              MOV   SP,STCK              ;RESET STACK
426  FF80:0154  C6 06 5E 20 01        MOV   DRVOFF,1             ;TURN OFF DRIVE
427  FF80:0159  F6 06 5F 20 01        TEST  DRVATV,1             ;DRIVING?
428  FF80:015E  74 03                 JZ    RSCMD1               ;IF NOT DRIVING
429  FF80:0160  E9 45 FF              JMP   BOSS                 ;IF STILL DRIVING
430  FF80:0163  32 C0           RSCMD1: XOR   AL,AL              ;NOW ZERO FLAGS
431  FF80:0165  B9 12 00              MOV   CX,ENDFLG            ;NUMBER OF FLAG LOCATIONS
432  FF80:0168  BF 52 20              MOV   DI,OFFSET FLAGST     ;START OF FLAGS
433  FF80:016B  F3 AA                 REP STOS BYTE PTR FLAGST
434  FF80:016D  E8 B2 04              CALL  BRKOFF               ;ENGAGE BRAKE
435  FF80:0170  C6 06 52 20 00        MOV   RESCMD,0             ;KILL RESET REQUEST
436  FF80:0175  B8 00 00              MOV   AX,0                 ;KILL LIGHTS
437  FF80:0178  E7 80                 OUT   P2PTA,AX
438  FF80:017A  E9 2B FF              JMP   BOSS
439
440  FF80:017D                  CHKDRV:
441  FF80:017D  FE 0E 4F 20           DEC   BYTE PTR EXITMR      ;DEC SAFETY TIMER
442  FF80:0181  75 01                 JNZ   CHKDR0               ;IF OK
```

```
FF80:0183  CC                      INT     3                       ;DO TIMER THIS WAY
FF80:0184  F6 06 55 20 01  CHKDR0: TEST    DRVREQ,1                ;NEW REQUEST
FF80:0189  75 01                   JNZ     CHKDR1                  ;IF REQUEST
FF80:018B  C3                      RET                             ;IF NOT
FF80:018C  C6 06 55 20 00  CHKDR1: MOV     DRVREQ,0                ;KILL REQUEST
FF80:0191  8B 1E 00 20             MOV     BX,POSCEC               ;BUT TEST FOR CLOSE
FF80:0195  2B 1E 44 20             SUB     BX,COMTMP
FF80:0199  C6 06 5A 20 00          MOV     BFLAG,0                 ;ASSUME IN BOUND
FF80:019E  2E 62 1E AC 06          BOUND   BX,CS:CLSETAB           ;CLOSE?
FF80:01A3  F6 06 5A 20 01          TEST    BFLAG,1
FF80:01A8  75 07                   JNZ     CHKDR2                  ;IF NOT CLOSE
FF80:01AA  C7 06 44 20 00 00       MOV     WORD PTR COMTMP,0       ;FOR DRIVE OK
FF80:01B0  C3                      RET                             ;IF CLOSE
FF80:01B1  C6 06 5E 20 01  CHKDR2: MOV     DRVOFF,1                ;TURN OFF DRIVE
FF80:01B6  C6 06 55 20 01          MOV     DRVREQ,1                ;RETAIN REQUEST
FF80:01BB  F6 06 5F 20 01          TEST    DRVATV,1                ;DRIVING?
FF80:01C0  74 01                   JZ      CHKDR3                  ;IF STOPPED
FF80:01C2  C3                      RET                             ;IF STILL DRIVING
FF80:01C3  32 C0           CHKDR3: XOR     AL,AL                   ;NOW ZERO FLAGS
FF80:01C5  B9 12 00                MOV     CX,ENDFLG               ;NUMBER OF FLAG LOCATIONS
FF80:01C8  BF 52 20                MOV     DI,OFFSET FLAGST        ;START OF FLAGS
FF80:01CB  F3 AA                   REP STOS BYTE PTR FLAGST
FF80:01CD  E8 52 04                CALL    BREOFF                  ;ENGAGE BRAKE

FF80:01D0                  DRVINT:
FF80:01D0  C6 06 55 20 00          MOV     DRVREQ,0                ;KILL REQUEST
FF80:01D5  F7 06 08 20 00 18       TEST    WORD PTR FAUL1,1800H    ;E-STOP, DRIVE LOCKOUT?
FF80:01DB  74 01                   JZ      DRVCNT                  ;IF OK
FF80:01DD  C3                      RET
FF80:01DE  32 C0           DRVCNT: XOR     AL,AL                   ;CLEAR ALL DRIVE FLAGS
FF80:01E0  B9 05 00                MOV     CX,OFFSET FLAGST+ENDFLG-DRVATV
FF80:01E3  BF 5F 20                MOV     DI,OFFSET DRVATV
FF80:01E6  F3 AA                   REP STOS BYTE PTR DRVATV
FF80:01E8  C7 06 08 20 00 00       MOV     WORD PTR FAUL1,0        ;CLEAR FAULTS
FF80:01EE  C7 06 0A 20 00 00       MOV     WORD PTR FAUL2,0
FF80:01F4  8B 1E 44 20             MOV     BX,COMTMP               ;MAKE TEMP COMMAND ACTUAL COMMAND
FF80:01F8  89 1E 00 20             MOV     POSCEC,BX
FF80:01FC  C7 06 44 20 00 00       MOV     WORD PTR COMTMP,0
FF80:0202  E8 AC 02                CALL    DSTOR

                                   ;LOCAL SETUP ENTERS HERE

FF80:0205                  LOCTST:
FF80:0205  F7 06 08 20 00 18       TEST    WORD PTR FAUL1,1800H    ;E-STOP, DRIVE LOCKOUT?
FF80:020B  74 01                   JZ      LOCCNT                  ;IF OK
FF80:020D  C3                      RET
FF80:020E  81 3E 0A 20 48 48 LOCCNT: CMP   WORD PTR FAUL2,4848H    ;NORMAL BDS3?
FF80:0214  74 01                   JZ      DRVIN0                  ;IF OK
FF80:0216  C3                      RET                             ;ELSE DO NOTHING
FF80:0217  8B 1E 04 20     DRVIN0: MOV     BX,ERROR                ;GET ERROR
FF80:021B  C6 06 5A 20 00          MOV     BFLAG,0
FF80:0220  2E 62 1E AC 06          BOUND   BX,CS:CLSETAB           ;CHECK FOR NEARNESS
FF80:0225  F6 06 5A 20 01          TEST    BFLAG,1
FF80:022A  75 01                   JNZ     DRVIN1                  ;IF NOT NEAR
FF80:022C  C3                      RET                             ;IF NEAR
FF80:022D  A0 08 20        DRVIN1: MOV     AL,FAUL1                ;GET LIMIT INFO
FF80:0230  F6 06 62 20 01          TEST    DIR,1                   ;TEST DIRECTION
```

```
500  FF80:0235  74 05              JZ      DRVIN2          ;IF POSITIVE
501  FF80:0237  24 18              AND     AL,24           ;CCW LIMIT?
502  FF80:0239  74 06              JZ      DRVIN3
503  FF80:023B  C3                 RET                     ;IF IN CCW LIMIT
504  FF80:023C  24 06        DRVIN2: AND     AL,6            ;CW LIMIT?
505  FF80:023E  74 01              JZ      DRVIN3
506  FF80:0240  C3                 RET                     ;IF IN CW LIMIT
507  FF80:0241  A0 62 20     DRVIN3: MOV     AL,DIR
508  FF80:0244  A2 63 20             MOV     LDIR,AL
509  FF80:0247  BA 00 00             MOV     DX,0            ;SET UP FOR DIVIDE
510  FF80:024A  A1 04 20             MOV     AX,ERROR        ;GET ERROR
511  FF80:024D  F6 06 63 20 01       TEST    LDIR,1          ;TEST FOR NEGATIVE
512  FF80:0252  74 02                JZ      DRVIN4          ;IF POSITIVE
513  FF80:0254  F7 D8                NEG     AX              ;MAKE ABSOLUTE VALUE
514  FF80:0256  50           DRVIN4: PUSH    AX              ;SAVE ERROR
515  FF80:0257  B9 64 00             MOV     CX,100          ;DIVIDE BY 100
516  FF80:025A  F7 F1                DIV     CX
517  FF80:025C  52                   PUSH    DX
518  FF80:025D  50                   PUSH    AX
519  FF80:025E  52                   PUSH    DX
520  FF80:025F  B9 0C 00             MOV     CX,12           ;TIMES 12
521  FF80:0262  F7 E1                MUL     CX
522  FF80:0264  8B D8                MOV     BX,AX           ;STORE HERE
523  FF80:0266  58                   POP     AX              ;NOW REMAINDER
524  FF80:0267  F7 E1                MUL     CX
525  FF80:0269  B9 64 00             MOV     CX,100          ;ADJUST
526  FF80:026C  F7 F1                DIV     CX
527  FF80:026E  03 D8                ADD     BX,AX
528  FF80:0270  A1 02 20             MOV     AX,POSD
529  FF80:0273  A3 4D 20             MOV     POSDOD,AX       ;OLD POS = CURRENT
530  FF80:0276  75 03                JNZ     DRVIN5          ;ZERO?
531  FF80:0278  BB 01 00             MOV     BX,1            ;IF ZERO
532  FF80:027B  F6 06 06 20 02 DRVIN5: TEST  SYSTEM,2        ;IN MANUAL?
533  FF80:0280  74 08                JZ      DRVIN6          ;IF NOT
534  FF80:0282  83 FB 3C             CMP     BX,60           ;>60?
535  FF80:0285  72 03                JC      DRVIN6          ;IF LESS
536  FF80:0287  BB 3C 00             MOV     BX,60           ;SET MAX RAMP
537  FF80:028A  F6 06 63 20 01 DRVIN6: TEST  LDIR,1          ;CHECK DIRECTION
538  FF80:028F  74 02                JZ      DRVIN7          ;IF POSITIVE
539  FF80:0291  F7 DB                NEG     BX
540  FF80:0293  03 C3          DRVIN7: ADD   AX,BX           ;FIRST BREAK POINT
541  FF80:0295  A3 49 20             MOV     BREAK1,AX
542  FF80:0298  A1 04 20             MOV     AX,ERROR        ;GET ERROR
543  FF80:029B  F6 06 63 20 01       TEST    LDIR,1          ;TEST FOR NEGATIVE
544  FF80:02A0  74 02                JZ      DRVIN8          ;IF POSITIVE
545  FF80:02A2  F7 D8                NEG     AX              ;MAKE ABSOLUTE VALUE
546  FF80:02A4  B9 28 00       DRVIN8: MOV   CX,40           ;SET SMALL DELTA
547  FF80:02A7  3D 00 10             CMP     AX,1000H        ;LARGER DELTA?
548  FF80:02AA  72 0B                JC      DRVIN9          ;IF SMALL
549  FF80:02AC  B9 3C 00             MOV     CX,60           ;SET MEDIUM DELTA
550  FF80:02AF  3D 00 30             CMP     AX,3000H        ;SMALL DELTA?
551  FF80:02B2  72 03                JC      DRVIN9          ;IF MEDIUM
552  FF80:02B4  B9 50 00             MOV     CX,80           ;ELSE SET LARGE
553  FF80:02B7  58             DRVIN9: POP   AX              ;NOW SECOND BREAK POINT
554  FF80:02B8  F7 E1                MUL     CX
555  FF80:02BA  8B D8                MOV     BX,AX           ;STORE HERE
556  FF80:02BC  58                   POP     AX              ;NOW REMAINDER
```

```
557  FF80:02BD  F7 E1                    MUL   CX
558  FF80:02BF  B9 64 00                 MOV   CX,100              ;ADJUST
559  FF80:02C2  F7 F1                    DIV   CX
560  FF80:02C4  03 D8                    ADD   BX,AX
561  FF80:02C6  58                       POP   AX                  ;RESTORE ERROR
562  FF80:02C7  F6 06 06 20 02           TEST  SYSTEM,2            ;IN MANUAL?
563  FF80:02CC  74 0C                    JZ    DRVINA              ;IF NOT
564  FF80:02CE  2B C3                    SUB   AX,BX               ;FIND DIFFERENCE
565  FF80:02D0  3D 41 00                 CMP   AX,65               ;>65?
566  FF80:02D3  72 05                    JC    DRVINA              ;IF LESS
567  FF80:02D5  03 D8                    ADD   BX,AX               ;GET ERROR IN BX
568  FF80:02D7  83 EB 41                 SUB   BX,65               ;MAKE BREAK 65 COUNTS LESS
569  FF80:02DA  A1 02 20          DRVINA: MOV   AX,POSD
570  FF80:02DD  F6 06 63 20 01           TEST  LDIR,1              ;CHECK DIRECTION
571  FF80:02E2  74 02                    JZ    DRVINB              ;IF POSITIVE
572  FF80:02E4  F7 DB                    NEG   BX
573  FF80:02E6  03 C3             DRVINB: ADD   AX,BX               ;SECOND BREAK POINT
574  FF80:02E8  A3 4B 20                 MOV   BREAK2,AX
575  FF80:02EB  C7 06 28 20 00 00        MOV   WORD PTR VEL1,0     ;INITIAL VEL = 0
576  FF80:02F1  E8 FC 02                 CALL  BRON                ;TURN ON BRAKE
577  FF80:02F4  E5 00                    IN    AX,P1PTA            ;GET CURRENT DATA
578  FF80:02F6  25 00 F0                 AND   AX,0F000H           ;JUST CURRENT STATUS
579  FF80:02F9  BB 04 00                 MOV   BX,CONRAM           ;START DRIVE AT 10mV
580  FF80:02FC  F6 06 63 20 01           TEST  LDIR,1              ;NEGATIVE?
581  FF80:0301  74 02                    JZ    DRVINC              ;IF POSITIVE
582  FF80:0303  F7 DB                    NEG   BX                  ;SET NEGATIVE
583  FF80:0305  81 E3 FF 0F       DRVINC: AND   BX,0FFFH            ;STRIP UNWANTED BITS
584  FF80:0309  0B C3                    OR    AX,BX               ;MERGE
585  FF80:030B  E7 00                    OUT   P1PTA,AX            ;SEND IT
586  FF80:030D  BA 96 01                 MOV   DX,LTCHDA           ;LATCH IT
587  FF80:0310  EE                       OUT   DX,AL
588  FF80:0311  80 CC 10                 OR    AH,10H              ;ENABLE DRIVE
589  FF80:0314  E7 00                    OUT   P1PTA,AX
590  FF80:0316  C6 06 48 20 1E           MOV   SPEED,30            ;SET RAMP VALUE
591  FF80:031B  C7 06 60 20 00 00        MOV   WORD PTR RAMPOS,0   ;SET RAMP UP
592  FF80:0321  BA 66 FF                 MOV   DX,T2MODE           ;NOW START TIMER 2
593  FF80:0324  B8 01 E0                 MOV   AX,0E001H
594  FF80:0327  EF                       OUT   DX,AX
595  FF80:0328  C6 06 5F 20 01           MOV   DRVATV,1            ;SET DRIVE ACTIVE
596  FF80:032D  E5 80                    IN    AX,P2PTA            ;GET CURRENT STATUS
597  FF80:032F  25 97 FF                 AND   AX,0FF97H           ;STRIP IT
598  FF80:0332  F6 06 06 20 02           TEST  SYSTEM,2            ;IN MANUAL?
599  FF80:0337  75 02                    JNZ   DRVIND              ;IF MANUAL
600  FF80:0339  0C 08                    OR    AL,8                ;SET CMD LIGHT
601  FF80:033B  F6 06 63 20 01    DRVIND: TEST  LDIR,1              ;NEGATIVE?
602  FF80:0340  74 06                    JZ    DRVINE              ;IF POSITIVE
603  FF80:0342  0D 20 00                 OR    AX,32               ;SET CCW LIGHT
604  FF80:0345  E9 03 00                 JMP   DRVINF
605  FF80:0348  0D 40 00          DRVINE: OR    AX,64               ;SET CW LIGHT
606  FF80:034B  E7 80             DRVINF: OUT   P2PTA,AX            ;SEND IT
607  FF80:034D  C6 06 4F 20 19           MOV   EXTTMR,25           ;INITIALIZE TIMER
608  FF80:0352  C3                       RET
609
610  FF80:0353                    LOCAL:
611  FF80:0353  80 0E 06 20 02           OR    SYSTEM,2            ;SET LOCAL FLAG
612  FF80:0358  F6 06 56 20 01           TEST  MANOVR,1            ;MANUAL OVERRIDE?
613  FF80:035D  74 12                    JZ    LOCLB               ;IF NOT
```

```
614  FF80:035F  F6 06 5F 20 01          TEST   DRVATV,1              ;DRIVE ACTIVE?
615  FF80:0364  75 03                   JNZ    LOCLA                 ;IF NOT OFF
616  FF80:0366  E9 3F FD                JMP    BOSS
617  FF80:0369  C6 06 5E 20 01    LOCLA: MOV    DRVOFF,1              ;SHUT DOWN
618  FF80:036E  E9 E2 FF                JMP    LOCAL
619  FF80:0371  BC 00 21           LOCLB: MOV    SP,STCK               ;SET STACK
620  FF80:0374  F6 06 08 20 01          TEST   FAUL1,1               ;APEX OK?
621  FF80:0379  74 03                   JZ     LC1                   ;IF APEX OK
622  FF80:037B  E9 CD FD                JMP    RSCMD                 ;IF APEX BROKE
623  FF80:037E  F6 06 5F 20 01    LC1:   TEST   DRVATV,1              ;DRIVE ACTIVE?
624  FF80:0383  75 2C                   JNZ    LC1B                  ;IF ACTIVE
625  FF80:0385  E8 29 01                CALL   DSTOR                 ;GET INFO
626  FF80:0388  BA 95 01                MOV    DX,MODESW             ;GET MODE SWITCH
627  FF80:038B  EC                      IN     AL,DX                 ;GET INFO
628  FF80:038C  F6 D0                   NOT    AL                    ;INVERT AL
629  FF80:038E  B4 00                   MOV    AH,0                  ;SET AH=0
630  FF80:0390  C6 06 5A 20 00          MOV    BFLAG,0               ;SET BLAG=0
631  FF80:0395  2E 62 06 B0 06          BOUND  AX,CS:MANRG           ;MODE SWITCH IN RANGE?
632  FF80:039A  F6 06 5A 20 01          TEST   BFLAG,1               ;OUT OF BOUNDS?
633  FF80:039F  74 03                   JZ     LC1A                  ;IF IN RANGE
634  FF80:03A1  B8 00 00                MOV    AX,0                  ;PROCESS=NORMAL
635  FF80:03A4  03 C0              LC1A:  ADD    AX,AX                 ;AX*2
636  FF80:03A6  8B D8                   MOV    BX,AX                 ;BX = INDEX
637  FF80:03A8  2E 8B 87 B4 06          MOV    AX,CS:MANTBL[BX]      ;GET ROUTINE
638  FF80:03AD  FF D0                   CALL   AX                    ;DO ROUTINE
639  FF80:03AF  EB 07                   JMP    SHORT LC1C
640  FF80:03B1  FE 0E 4F 20        LC1B:  DEC    BYTE PTR EXITMR       ;DEC SAFETY TIMER
641  FF80:03B5  75 01                   JNZ    LC1C                  ;IF OK
642  FF80:03B7  CC                      INT    3                     ;DO TIMER THIS WAY
643  FF80:03B8  E5 80              LC1C:  IN     AX,P2PTA              ;GET SWITCH STATUS
644  FF80:03BA  F6 C4 80                TEST   AH,128                ;DRIVE CW?
645  FF80:03BD  75 03                   JNZ    LC2                   ;NOT DRIVE CW
646  FF80:03BF  E9 2C 00                JMP    LUP                   ;DRIVE CW
647  FF80:03C2  F6 C4 40           LC2:   TEST   AH,64                 ;DRIVE CCW
648  FF80:03C5  75 03                   JNZ    LC3                   ;NOT DRIVE CCW
649  FF80:03C7  E9 59 00                JMP    LDWN                  ;DRIVE CCW
650  FF80:03CA  F6 06 5F 20 01    LC3:   TEST   DRVATV,1              ;DRIVE ACTIVE?
651  FF80:03CF  74 08                   JZ     LC4                   ;IF NOT ACTIVE
652  FF80:03D1  C6 06 5E 20 01          MOV    DRVOFF,1              ;SET DRIVE OFF
653  FF80:03D6  E9 7A FF                JMP    LOCAL
654  FF80:03D9  F6 C4 20           LC4:   TEST   AH,32                 ;STILL IN LOCAL
655  FF80:03DC  74 03                   JZ     LC5                   ;IF IN LOCAL
656  FF80:03DE  E9 6A FD                JMP    RSCMD                 ;LEAVE IF NOT IN LOCAL
657  FF80:03E1  32 C0              LC5:   XOR    AL,AL                 ;CLEAR ALL DRIVE FLAGS
658  FF80:03E3  B9 05 00                MOV    CX,OFFSET FLAGST+ENDFLG-DRVATV
659  FF80:03E6  BF 5F 20                MOV    DI,OFFSET DRVATV
660  FF80:03E9  F3 AA                   REP STOS BYTE PTR DRVATV
661  FF80:03EB  E9 65 FF                JMP    LOCAL
662
663  FF80:03EE                     LUP:
664  FF80:03EE  F6 06 5F 20 01          TEST   DRVATV,1              ;ALREADY RUNNING?
665  FF80:03F3  74 0F                   JZ     LUP1                  ;IF NOT MOVING
666  FF80:03F5  F6 06 5F 20 02          TEST   DRVATV,2              ;MOVING CW?
667  FF80:03FA  75 05                   JNZ    LUP2                  ;ALREADY GOING CW
668  FF80:03FC  C6 06 5E 20 01          MOV    DRVOFF,1              ;KILL DRIVE
669  FF80:0401  E9 4F FF           LUP2:  JMP    LOCAL
670  FF80:0404  C7 06 00 20 00 F2  LUP1:  MOV    WORD PTR POSCEC,HIGH  ;SET UPPER BOUNDARY
```

```
671  FF80:040A  E8 A4 00          CALL   DSTOR                ;GET DATA
672  FF80:040D  A1 02 20          MOV    AX,POSD              ;GET CURRENT POSITION
673  FF80:0410  3D 00 F2          CMP    AX,HIGH              ;< DRIVE LIMITS
674  FF80:0413  72 03             JC     LUP3                 ;IF LESS
675  FF80:0415  E9 3B FF          JMP    LOCAL
676  FF80:0418  E8 EA FD    LUP3: CALL   LOCTST               ;SET UP DRIVE PARAMETERS
677  FF80:041B  80 0E 5F 20 02    OR     DRVATV,2
678  FF80:0420  E9 30 FF          JMP    LOCAL
679
680  FF80:0423              LDWN:
681  FF80:0423  F6 06 5F 20 01    TEST   DRVATV,1             ;ALREADY RUNNING?
682  FF80:0428  74 0F              JZ     LDWN1                ;IF NOT MOVING
683  FF80:042A  F6 06 5F 20 04     TEST   DRVATV,4             ;MOVING CCW?
684  FF80:042F  75 05              JNZ    LDWN2                ;ALREADY GOING CCW
685  FF80:0431  C6 06 5E 20 01     MOV    DRVOFF,1             ;KILL DRIVE
686  FF80:0436  E9 1A FF    LDWN2: JMP    LOCAL
687  FF80:0439  C7 06 00 20 00 0A LDWN1: MOV    WORD PTR POSCEC,LOW  ;SET LOWER BOUNDARY
688  FF80:043F  E8 6F 00           CALL   DSTOR                ;GET DATA
689  FF80:0442  A1 02 20           MOV    AX,POSD              ;GET CURRENT POSITION
690  FF80:0445  3D 00 0A           CMP    AX,LOW               ;> DRIVE LIMITS
691  FF80:0448  73 03              JNC    LDWN3                ;IF LESS
692  FF80:044A  E9 06 FF           JMP    LOCAL
693  FF80:044D  E8 B5 FD    LDWN3: CALL   LOCTST               ;SET UP DRIVE PARAMETERS
694  FF80:0450  80 0E 5F 20 04     OR     DRVATV,4             ;SET CCW FLAG
695  FF80:0455  E9 FB FE           JMP    LOCAL
696
697
698  FF80:0458              NORMIT:
699  FF80:0458  E8 C7 01           CALL   BRKOFF               ;TURN OFF BRAKE
700  FF80:045B  E9 06 00           JMP    ZERDA                ;ZERO D/A
701
702  FF80:045E              NEGDA:
703  FF80:045E  BB 00 08           MOV    BX,800H              ;SET -5V
704  FF80:0461  E9 09 00           JMP    SETDA
705
706  FF80:0464              ZERDA:
707  FF80:0464  BB 00 00           MOV    BX,0                 ;SET 0V
708  FF80:0467  E9 03 00           JMP    SETDA
709
710  FF80:046A              POSDA:
711  FF80:046A  BB FF 07           MOV    BX,7FFH              ;SET +5V
712  FF80:046D              SETDA:
713  FF80:046D  E5 00              IN     AX,P1PTA             ;GET CURRENT STATUS
714  FF80:046F  25 00 F0           AND    AX,0F000H            ;JUST STATUS
715  FF80:0472  0B C3              OR     AX,BX                ;MERGE
716  FF80:0474  E7 00              OUT    P1PTA,AX             ;SEND IT
717  FF80:0476  BA 96 01           MOV    DX,LTCHDA            ;NOW LATCH IT IN
718  FF80:0479  EE                 OUT    DX,AL
719  FF80:047A  C3                 RET
720
721  FF80:047B  E5 80       NEGAD: IN     AX,P2PTA             ;GET PORT 2
722  FF80:047D  25 F8 FF           AND    AX,0FFF8H            ;STRIP OLD A/D REQ
723  FF80:0480  0D 03 00           OR     AX,3                 ;REQUEST -10V
724  FF80:0483  EB 12              JMP    SHORT READAD
725
726  FF80:0485  E5 80       ZERAD: IN     AX,P2PTA             ;GET PORT 2
727  FF80:0487  25 F8 FF           AND    AX,0FFF8H            ;STRIP OLD A/D REQ
```

```
728  FF80:048A  0D 04 00          OR      AX,4              ;REQUEST OV
729  FF80:048D  EB 08             JMP     SHORT READAD
730
731  FF80:048F  E5 80     POSAD:  IN      AX,P2PTA          ;GET PORT 2
732  FF80:0491  25 F8 FF          AND     AX,0FFF8H         ;STRIP OLD A/D REQ
733  FF80:0494  0D 02 00          OR      AX,2              ;REQUEST +10V
734
735  FF80:0497  C6 06 59 20 00 READAD: MOV ADFLAG,0         ;RESET FLAG
736  FF80:049C  E7 80             OUT     P2PTA,AX
737  FF80:049E  BA 94 01          MOV     DX,STCNV          ;START CONVERSION
738  FF80:04A1  EE                OUT     DX,AL
739  FF80:04A2  FB                STI                       ;ENABLE INTERRUPTS
740  FF80:04A3  F6 06 59 20 01 READA1: TEST ADFLAG,1        ;ANALOG READY?
741  FF80:04A8  74 F9             JZ      READA1            ;IF NOT
742  FF80:04AA  A1 46 20          MOV     AX,ADVAL          ;GET CURRENT VELOCITY
743  FF80:04AD  A3 28 20          MOV     VEL1,AX           ;SAVE VELOCITY
744  FF80:04B0  C3                RET
745
746  FF80:04B1            DSTOR:
747  FF80:04B1  32 C0             XOR     AL,AL
748  FF80:04B3  BA 80 01          MOV     DX,APEXRQ         ;GET APEX DATA
749  FF80:04B6  EE                OUT     DX,AL
750  FF80:04B7  83 C2 02          ADD     DX,2              ;POINT TO APEXRS
751  FF80:04BA  B9 0C 00          MOV     CX,12             ;12 TRIALS
752  FF80:04BD            DSTOR1:
753  FF80:04BD  EC                IN      AL,DX             ;SEE IF DATA READY
754  FF80:04BE  24 C0             AND     AL,0C0H           ;STRIP DATA
755  FF80:04C0  3C 80             CMP     AL,80H            ;LOOK FOR PATTERN
756  FF80:04C2  74 0A             JZ      DSTOR2            ;IF READY
757  FF80:04C4  E2 F7             LOOP    DSTOR1            ;IF NOT
758  FF80:04C6  80 0E 08 20 01    OR      FAUL1,1           ;SET APEX DEAD
759  FF80:04CB  E9 C6 00          JMP     DSTOR6            ;SKIP APEX GATHERING
760  FF80:04CE            DSTOR2:
761  FF80:04CE  80 26 08 20 FE    AND     FAUL1,0FEH        ;SET APEX OK
762  FF80:04D3  4A                DEC     DX
763  FF80:04D4  ED                IN      AX,DX             ;GET POSITION
764  FF80:04D5  8B D8             MOV     BX,AX             ;PUT INTO BX
765  FF80:04D7  C1 E3 02          SHL     BX,2              ;GET RID OF RESPONSE BITS
766  FF80:04DA  83 C2 02          ADD     DX,2              ;POINT TO NEXT SECTION
767  FF80:04DD  ED                IN      AX,DX             ;GET NEXT INFO
768  FF80:04DE  C0 C4 02          ROL     AH,2              ;GET FINAL POSITION
769  FF80:04E1  80 E4 03          AND     AH,03             ;PREPARE TO MERGE
770  FF80:04E4  0A DC             OR      BL,AH             ;MERGE BITS
771  FF80:04E6  89 1E 02 20       MOV     POSD,BX           ;SAVE POSITION
772  FF80:04EA  A1 00 20          MOV     AX,POSCEC
773  FF80:04ED  2B C3             SUB     AX,BX             ;GET ERROR
774  FF80:04EF  C6 06 62 20 00    MOV     DIR,0             ;ASSUME POSITIVE
775  FF80:04F4  73 05             JNC     DSTO2A            ;IF POSITIVE
776  FF80:04F6  C6 06 62 20 01    MOV     DIR,1             ;SET NEGATIVE
777  FF80:04FB            DSTO2A:
778  FF80:04FB  A3 04 20          MOV     ERROR,AX          ;STORE IT HERE
779  FF80:04FE  ED                IN      AX,DX             ;GET VELOCITY
780  FF80:04FF  8A F8             MOV     BH,AL             ;SAVE PARTIAL DATA
781  FF80:0501  C1 E8 02          SHR     AX,2
782  FF80:0504  25 FF 0F          AND     AX,0FFFH          ;STRIP UPPER NYBBLE
783  FF80:0507  A3 14 20          MOV     VEL,AX            ;SAVE VELOCITY
784  FF80:050A  83 C2 02          ADD     DX,2              ;POINT TO ANALOGS
```

```
785  FF80:050D  ED                      IN      AX,DX           ;GET DATA
786  FF80:050E  C1 E8 02                SHR     AX,2            ;SHIFT INTO POSITION
787  FF80:0511  C0 CF 02                ROR     BH,2            ;SHIFT PARTIAL
788  FF80:0514  81 E3 00 C0             AND     BX,0C000H       ;PREPAR TO MERGE
789  FF80:0518  0B D8                   OR      BX,AX           ;SAVE FOR FURTHER PROCESSING
790  FF80:051A  83 C2 02                ADD     DX,2            ;POINT TO DISCRETE INFO
791  FF80:051D  EC                      IN      AL,DX           ;GET DATA
792  FF80:051E  24 1F                   AND     AL,1FH          ;ONLY DISCRETES
793  FF80:0520  80 26 06 20 FB          AND     SYSTEM,0FBH     ;ASSUME BRAKE NOT RELEASED
794  FF80:0525  A8 10                   TEST    AL,16           ;SEE IF BRAKE IS RELEASED
795  FF80:0527  74 05                   JZ      DSTOR3
796  FF80:0529  80 0E 06 20 04          OR      SYSTEM,4        ;SET BRAKE RELEASE
797  FF80:052E  24 0F           DSTOR3: AND     AL,0FH          ;SET LIMIT INFO
798  FF80:0530  D0 E0                   SHL     AL,1
799  FF80:0532  80 26 08 20 E1          AND     FAUL1,0E1H      ;CLEAR LIMIT FAULTS
800  FF80:0537  08 06 08 20             OR      FAUL1,AL
801  FF80:053B  8B D3                   MOV     DX,BX           ;SAVE TABLE
802  FF80:053D  81 CB 0F 00             OR      BX,0FH          ;MAKE IT LIKE STANDARD ANALOG
803  FF80:0541  81 E2 0F 00             AND     DX,0000FH       ;DEVELOP TABLE ADDRESS
804  FF80:0545  01 16 50 20             ADD     RANDOM,DX       ;DEVELOP NEW NUMBER
805  FF80:0549  81 26 50 20 3F 00       AND     WORD PTR RANDOM,3FH
806  FF80:054F  FE 06 50 20             INC     RANDOM
807  FF80:0553  87 D3                   XCHG    DX,BX
808  FF80:0555  83 FB 03                CMP     BX,3
809  FF80:0558  7E 08                   JLE     DST03A
810  FF80:055A  83 EB 04                SUB     BX,4
811  FF80:055D  83 FB 03                CMP     BX,3
812  FF80:0560  7E 32                   JLE     DSTOR6          ;OUT OF RANGE
813  FF80:0562  8B CB           DST03A: MOV     CX,BX           ;SAVE ANALOG POINTER IN C
814  FF80:0564  83 C1 01                ADD     CX,1            ;PLUS 1 FOR BIT PLACEMENT
815  FF80:0567  D1 E3                   SHL     BX,1
816  FF80:0569  89 97 0E 20             MOV     ANADT[BX],DX    ;SAVE ANALOG DATA
817  FF80:056D  D1 E3                   SHL     BX,1            ;NOW FOR BOUND CHECK
818  FF80:056F  C6 06 5A 20 00          MOV     BFLAG,0         ;BOUND FLAG = OK
819  FF80:0574  2E 62 97 7C 06          BOUND   DX,CS:ANATAB[BX] ;CHECK ANALOG BOUNDS
820  FF80:0579  B8 00 80                MOV     AX,8000H        ;ASSUME FAIL
821  FF80:057C                  DSTOR4:
822  FF80:057C  D1 C0                   ROL     AX,1            ;PLACE FLAG BIT
823  FF80:057E  E2 FC                   LOOP    DSTOR4
824  FF80:0580  F6 06 5A 20 01          TEST    BFLAG,1         ;BOUND OK?
825  FF80:0585  75 09                   JNZ     DSTOR5          ;IF BAD
826  FF80:0587  F7 D0                   NOT     AX              ;SET OK
827  FF80:0589  21 06 0C 20             AND     ANAFL,AX        ;STRIP BAD ANALOG FLAG
828  FF80:058D  E9 04 00                JMP     DSTOR6
829  FF80:0590                  DSTOR5:
830  FF80:0590  09 06 0C 20             OR      ANAFL,AX        ;SET BAD ANALOG FLAG
831  FF80:0594                  DSTOR6:
832  FF80:0594  E5 80                   IN      AX,P2PTA        ;GET LED STATUS
833  FF80:0596  25 6F FE                AND     AX,0FE6FH       ;KILL ALL LIGHTS
834  FF80:0599  F6 06 06 20 04          TEST    SYSTEM,4        ;IS BRAKE RELEASED?
835  FF80:059E  74 03                   JZ      DSTOR7          ;IF NOT
836  FF80:05A0  0D 10 00                OR      AX,16           ;SET BRAKE LED ON
837  FF80:05A3                  DSTOR7:
838  FF80:05A3  8B 1E 08 20             MOV     BX,FAUL1        ;NOW DO CCW LIMIT
839  FF80:05A7  81 E3 18 00             AND     BX,24
840  FF80:05AB  74 03                   JZ      DSTOR8          ;IF SET LIGHT LED
841  FF80:05AD  0D 80 00                OR      AX,128          ;SET CCW LIMIT LED ON
```

```
842  FF80:05B0                      DSTOR8:
843  FF80:05B0  8B 1E 08 20                MOV     BX,FAUL1              ;NOW DO CW LIMIT
844  FF80:05B4  81 E3 06 00                AND     BX,6
845  FF80:05B8  74 03                      JZ      DSTOR9                ;IF SET LIGHT LED
846  FF80:05BA  0D 00 01                   OR      AX,256                ;SET CW LIMIT LED ON
847  FF80:05BD                      DSTOR9:
848  FF80:05BD  E7 80                      OUT     P2PTA,AX
849  FF80:05BF  BA 97 01                   MOV     DX,BDERL1             ;GET IDD STUFF
850  FF80:05C2  ED                         IN      AX,DX
851  FF80:05C3  F7 D0                      NOT     AX                    ;COMPLEMENT
852  FF80:05C5  A3 0A 20                   MOV     FAUL2,AX              ;SAVE IN FAUL2
853  FF80:05C8  81 26 08 20 FF E7          AND     WORD PTR FAUL1,0E7FFH ;STRIP OUT FAULTS
854  FF80:05CE  E5 80                      IN      AX,P2PTA              ;GET E-STOP, DRIVE LOCIOUT
855  FF80:05D0  25 00 18                   AND     AX,1800H              ;STRIP JUNK
856  FF80:05D3  80 F4 10                   XOR     AH,10H                ;INVERT DRIVE LOCIOUT
857  FF80:05D6  09 06 08 20                OR      WORD PTR FAUL1,AX     ;MERGE
858  FF80:05DA  C3                         RET
859
860  FF80:05DB                      BDS3RS:
861  FF80:05DB  E5 00                      IN      AX,P1PTA              ;GET CONTROL PORT
862  FF80:05DD  0D 00 20                   OR      AX,2000H              ;DO RESET
863  FF80:05E0  E7 00                      OUT     P1PTA,AX
864  FF80:05E2  B9 01 00                   MOV     CX,1                  ;DELAY 1 SEC
865  FF80:05E5  E8 63 00                   CALL    DELAY
866  FF80:05E8  E5 00                      IN      AX,P1PTA              ;GET CONTROL AGAIN
867  FF80:05EA  25 FF DF                   AND     AX,0DFFFH             ;KILL RESET
868  FF80:05ED  E7 00                      OUT     P1PTA,AX
869  FF80:05EF  C3                         RET
870
871  FF80:05F0                      BRION:
872  FF80:05F0  E5 00                      IN      AX,P1PTA              ;GET CONTROL PORT
873  FF80:05F2  0D 00 40                   OR      AX,4000H              ;RELEASE THE BRAKE
874  FF80:05F5  E7 00                      OUT     P1PTA,AX              ;DO IT
875  FF80:05F7  C6 06 5C 20 00             MOV     TFLAG1,0              ;SET TIMER FLAG = OK
876  FF80:05FC  E8 5E 00                   CALL    SEC1                  ;1 SEC
877  FF80:05FF  E8 AF FE            BRION1: CALL    DSTOR                 ;GET UPDATE
878  FF80:0602  F6 06 06 20 04             TEST    SYSTEM,4              ;BRAKE FREE?
879  FF80:0607  75 10                      JNZ     BRION2                ;IF FREE CONTINUE
880  FF80:0609  F6 06 5C 20 01             TEST    TFLAG1,1              ;TIMED OUT?
881  FF80:060E  74 EF                      JZ      BRION1                ;IF NOT
882  FF80:0610  81 0E 08 20 00 01          OR      WORD PTR FAUL1,256    ;SET BRAKE FAULT
883  FF80:0616  E9 32 FB                   JMP     RSCMD                 ;RESET COMMAND
884  FF80:0619                      BRION2:
885  FF80:0619  81 26 08 20 FF FE          AND     WORD PTR FAUL1,0FEFFH ;CLEAR BRAKE FAULT
886  FF80:061F  E9 4A 00                   JMP     TMROFF                ;STOP TIMER
887
888  FF80:0622                      BRIOFF:
889  FF80:0622  E5 00                      IN      AX,P1PTA              ;GET CONTROL PORT
890  FF80:0624  25 FF BF                   AND     AX,0BFFFH             ;ENGAGE THE BRAKE
891  FF80:0627  E7 00                      OUT     P1PTA,AX              ;DO IT
892  FF80:0629  C6 06 5C 20 00             MOV     TFLAG1,0              ;SET TIMER FLAG = OK
893  FF80:062E  E8 2C 00                   CALL    SEC1                  ;1 SEC
894  FF80:0631  E8 7D FE            BRIOF1: CALL    DSTOR                 ;GET UPDATE
895  FF80:0634  F6 06 06 20 04             TEST    SYSTEM,4              ;BRAKE FREE?
896  FF80:0639  74 31                      JZ      TMROFF                ;IF FREE CONTINUE
897  FF80:063B  F6 06 5C 20 01             TEST    TFLAG1,1              ;TIMED OUT?
898  FF80:0640  74 EF                      JZ      BRIOF1                ;IF NOT
```

```
899  FF80:0642  81 0E 08 20 00 01          OR      WORD PTR FAUL1,256      ;SET BRAKE FAULT
900  FF80:0648  E9 00 FB                  JMP     RSCMD                   ;RESET COMMAND
901
902  FF80:064B  E8 0F 00           DELAY:  CALL    SEC1                    ;START TIMER FOR 1 SECOND
903  FF80:064E  C6 06 5C 20 00     DELA1:  MOV     TFLAG1,0                ;RESET TFLAG
904  FF80:0653  F6 06 5C 20 01     DELA2:  TEST    TFLAG1,1                ;TIME OUT?
905  FF80:0658  74 F9                      JZ      DELA2                   ;WAIT
906  FF80:065A  E2 F2                      LOOP    DELA1                   ;UNTIL TIME HAS PASSED
907  FF80:065C  C3                         RET
908
909  FF80:065D                     SEC1:
910  FF80:065D  BA 5E FF                   MOV     DX,T1MODE               ;ENABLE TIMER1 FIRST
911  FF80:0660  B8 09 E0                   MOV     AX,0E009H
912  FF80:0663  EF                         OUT     DX,AX
913  FF80:0664  BA 66 FF                   MOV     DX,T2MODE               ;NOW TIMER2
914  FF80:0667  B8 01 C0                   MOV     AX,0C001H
915  FF80:066A  EF                         OUT     DX,AX
916  FF80:066B  C3                         RET
917
918  FF80:066C                     TMROFF:
919  FF80:066C  BE 00 00                   MOV     SI,OFFSET TSETUP        ;POINT TO TABLE
920  FF80:066F  BA 50 FF                   MOV     DX,TOCOUT               ;POINT TO SOURCE POINTER
921  FF80:0672  B9 0C 00                   MOV     CX,12
922  FF80:0675  6F                 TSET:   OUTSW
923  FF80:0676  83 C2 02                   ADD     DX,2
924  FF80:0679  E2 FA                      LOOP    TSET
925  FF80:067B  C3                         RET
926  FF80:067C
927
928                                ;       TABLES FOR BOUND CHECKS
929
930
931  FF80:067C  0080               ANATAB: .WORD   8000H                   ;GND
932  FF80:067E  FF7F                       .WORD   7FFFH
933  FF80:0680  0080                       .WORD   8000H                   ;GND
934  FF80:0682  FF7F                       .WORD   7FFFH
935  FF80:0684  0080                       .WORD   8000H                   ;GND
936  FF80:0686  FF7F                       .WORD   7FFFH
937  FF80:0688  0080                       .WORD   8000H                   ;ROTATION VELOCITY
938  FF80:068A  FF7F                       .WORD   7FFFH
939  FF80:068C  0080                       .WORD   8000H                   ;+15V/2
940  FF80:068E  FF7F                       .WORD   7FFFH
941  FF80:0690  0080                       .WORD   8000H                   ;-15V/2
942  FF80:0692  FF7F                       .WORD   7FFFH
943  FF80:0694  0080                       .WORD   8000H                   ;+5V
944  FF80:0696  FF7F                       .WORD   7FFFH
945  FF80:0698  0080                       .WORD   8000H                   ;+10V
946  FF80:069A  FF7F                       .WORD   7FFFH
947  FF80:069C  0080                       .WORD   8000H                   ;GND
948  FF80:069E  FF7F                       .WORD   7FFFH
949  FF80:06A0  0080                       .WORD   8000H                   ;-10V
950  FF80:06A2  FF7F                       .WORD   7FFFH
951  FF80:06A4  0080                       .WORD   8000H                   ;MOUNT TEMP 2
952  FF80:06A6  FF7F                       .WORD   7FFFH
953  FF80:06A8  0080                       .WORD   8000H                   ;BIN TEMP
954  FF80:06AA  FF7F                       .WORD   7FFFH
955
```

```
956  FF80:06AC                    CLSETAB:
957  FF80:06AC  F8FF                      .WORD   0FFF8H              ;8 COUNTS BEFORE WE MOVE
958  FF80:06AE  0800                      .WORD   00008H
959
960  FF80:06B0                    MANRG:
961  FF80:06B0  0000                      .WORD   0                   ;MODES 0 - 4 ACTIVE
962  FF80:06B2  0700                      .WORD   7
963
964                               ;       MODE SWITCH ENTRY CONTROL
965
966  FF80:06B4                    MANTBL:
967  FF80:06B4  5804                      .WORD   NORMIT
968  FF80:06B6  F005                      .WORD   BRKON
969  FF80:06B8  5E04                      .WORD   NEGDA
970  FF80:06BA  6404                      .WORD   ZERDA
971  FF80:06BC  6A04                      .WORD   POSDA
972  FF80:06BE  7B04                      .WORD   NEGAD
973  FF80:06C0  8504                      .WORD   ZERAD
974  FF80:06C2  8F04                      .WORD   POSAD
975
976
977  FF80:07F0                            ORG     07F0H
978
979                               ;THIS AREA IS RESERVED FOR RESET PARAMETERS.  CANNOT BE LARGER THEN 16 BYTES
980
981  FF80:07F0  BA A0 FF                  MOV     DX,UMCS             ;MAKE UPPER MEMORY 2K BLOCK
982  FF80:07F3  B8 BD FF                  MOV     AX,UMBS
983  FF80:07F6  EF                        OUT     DX,AX
984  FF80:07F7  EA 00 00 80 FF            JMP     FAR INITAL          ;ENTER INTO LOWER EPROM SECTION
985
986  FF80:07FC                            END


        Lines Assembled :  986          Assembly Errors :  0
```

2500 A.D. 80186 Cross Assembler - Version 4.00g

Input Filename : FEINT.asm
Output Filename : FEINT.obj

```
1
2                                       LIST ON
3
4                               ;       INTERRUPT SECTION FOR FOCUS AXIS FOR VLBA ANTENNAS
5                               ;       WRITTEN BY:     WAYNE M. KOSKI
6                               ;       LAST REVISION:  NOVEMBER 29, 1989
7                               ;
8                               ;
9                               ;       THINGS TO DO AND GENERAL NOTES:
10                              ;
11                              ;         1.  THIS SECTION OF CODE WILL BE THE INTERRUPT
12                              ;             ROUTINES AND SHALL EXCLUDE THE MAIN RUNNING ROUTINES
13                              ;             WHICH SHALL BE IN THE OTHER EPROM.
14                              ;         2.  ROTATION AXIS SHOULD LOOK SIMULAR TO FOCUS AT
15                              ;             PRESENT.
16                              ;         3.  THIS VERSION UPDATES TO THE NEW F/R CONTROLLER MODULE.
17
18                                      ASSUME  CS:CODE, DS:DSEG
19
20                                      .OUTPUT 2500AD
21                                      .OPTIONS H
22
23
24                                      LIST OFF
25
26                              ;       80188 INTERNAL PORTS FOR CONTROL AND GUIDANCE
27
28
29
30                              ;       80188 INTERRUPT CONTROL / STATUS REGISTARS
31
32          0000:FF22           EOI:    .EQUAL  0FF22H              ;END OF INTERRUPT REGISTAR
33          0000:FF24           POLL:   .EQUAL  0FF24H              ;INTERRUPT POLL REGISTAR
34          0000:FF26           POLLS:  .EQUAL  0FF26H              ;INTERRUPT POLL STATUS REGISTAR
35          0000:FF28           MASK:   .EQUAL  0FF28H              ;INTERRUPT MASK REGISTAR
36          0000:FF2A           PMASK:  .EQUAL  0FF2AH              ;INTERRUPT PRIORITY MASK REGISTAR
37          0000:FF2C           ISR:    .EQUAL  0FF2CH              ;INTERRUPT IN SERVICE REGISTAR
38          0000:FF2E           IRR:    .EQUAL  0FF2EH              ;INTERRUPT REQUEST REGISTAR
39          0000:FF30           ICSR:   .EQUAL  0FF30H              ;INTERRUPT CONTROL STATUS REGISTAR
40          0000:FF32           ITCR:   .EQUAL  0FF32H              ;INTERRUPT TIMER CONTROL REGISTAR
41          0000:FF34           IDOCR:  .EQUAL  0FF34H              ;INTERRUPT DMA 0 CONTROL REGISTAR
42          0000:FF36           ID1CR:  .EQUAL  0FF36H              ;INTERRUPT DMA 1 CONTROL REGISTAR
43          0000:FF38           INTOCR: .EQUAL  0FF38H              ;INT 0 CONTROL REGISTAR
```

```
                0000:FF3A       INT1CR: .EQUAL  0FF3AH          ;INT 1 CONTROL REGISTAR
                0000:FF3C       INT2CR: .EQUAL  0FF3CH          ;INT 2 CONTROL REGISTAR
                0000:FF3E       INT3CR: .EQUAL  0FF3EH          ;INT 3 CONTROL REGISTAR


                                ;       80188 TIMER CONTROL REGISTARS

                0000:FF50       T0COUT: .EQUAL  0FF50H          ;TIMER 0 COUNT REGISTAR
                0000:FF52       T0MAXA: .EQUAL  0FF52H          ;TIMER 0 MAXIMUM COUNT A REGISTAR
                0000:FF54       T0MAXB: .EQUAL  0FF54H          ;TIMER 0 MAXIMUM COUNT B REGISTAR
                0000:FF56       T0MODE: .EQUAL  0FF56H          ;TIMER 0 MODE REGISTAR
                0000:FF58       T1COUT: .EQUAL  0FF58H          ;TIMER 1 COUNT REGISTAR
                0000:FF5A       T1MAXA: .EQUAL  0FF5AH          ;TIMER 1 MAXIMUM COUNT A REGISTAR
                0000:FF5C       T1MAXB: .EQUAL  0FF5CH          ;TIMER 1 MAXIMUM COUNT B REGISTAR
                0000:FF5E       T1MODE: .EQUAL  0FF5EH          ;TIMER 1 MODE REGISTAR
                0000:FF60       T2COUT: .EQUAL  0FF60H          ;TIMER 2 COUNT REGISTAR
                0000:FF62       T2MAXA: .EQUAL  0FF62H          ;TIMER 2 MAXIMUM COUNT A REGISTAR
                0000:FF66       T2MODE: .EQUAL  0FF66H          ;TIMER 2 MODE REGISTAR


                                ;       80188 CHIP SELECT CONTROL REGISTARS

                0000:FFA0       UMCS:   .EQUAL  0FFA0H          ;80188 INTERNAL UPPER MEMORY CHIP
                                                                ;SELECT CONTROL BLOCK REGISTAR
                0000:FFA2       LMCS:   .EQUAL  0FFA2H          ;80188 INTERNAL LOWER MEMORY CHIP
                                                                ;SELECT CONTROL BLOCK REGISTAR
                0000:FFA4       PACS:   .EQUAL  0FFA4H          ;80188 INTERNAL PERIPHERAL CHIP
                                                                ;SELECT CONTROL BLOCK REGISTAR
                0000:FFA6       MMCS:   .EQUAL  0FFA6H          ;80188 INTERNAL MIDDLE MEMORY
                                                                ;START ADDRESS REGISTAR
                0000:FFA8       MPCS:   .EQUAL  0FFA8H          ;80188 INTERNAL MIDDLE MEMORY CHIP
                                                                ;SELECT CONTROL BLOCK REGISTAR


                                ;       80188 DMA CHANNEL CONTROL REGISTARS

                0000:FFC0       D0SPL:  .EQUAL  0FFC0H          ;DMA 0 SOURCE POINTER LSB REGISTAR
                0000:FFC2       D0SPM:  .EQUAL  0FFC2H          ;DMA 0 SOURCE POINTER MSB REGISTAR
                0000:FFC4       D0DPL:  .EQUAL  0FFC4H          ;DMA 0 DESTINATION POINTER LSB REGISTAR
                0000:FFC6       D0DPM:  .EQUAL  0FFC6H          ;DMA 0 DESTINATION POINTER MSB REGISTAR
                0000:FFC8       D0TC:   .EQUAL  0FFC8H          ;DMA 0 TRANSFER COUNT REGISTAR
                0000:FFCA       D0MODE: .EQUAL  0FFCAH          ;DMA 0 MODE REGISTAR
                0000:FFD0       D1SPL:  .EQUAL  0FFD0H          ;DMA 1 SOURCE POINTER LSB REGISTAR
                0000:FFD2       D1SPM:  .EQUAL  0FFD2H          ;DMA 1 SOURCE POINTER MSB REGISTAR
                0000:FFD4       D1DPL:  .EQUAL  0FFD4H          ;DMA 1 DESTINATION POINTER LSB REGISTAR
                0000:FFD6       D1DPM:  .EQUAL  0FFD6H          ;DMA 1 DESTINATION POINTER MSB REGISTAR
                0000:FFD8       D1TC:   .EQUAL  0FFD8H          ;DMA 1 TRANSFER COUNT REGISTAR
                0000:FFDA       D1MODE: .EQUAL  0FFDAH          ;DMA 1 MODE REGISTAR


                                ;       80188 INTERNAL I/O RELOCATION REGISTAR

                0000:FFFE       RELOC:  .EQUAL  0FFFEH          ;I/O RELOCATION REGISTAR


                                ;       80188 INITIAL VALUES FOR INTERNAL REGISTARS

```

```
101             0000:007D       LMBS:   .EQUAL  007DH           ;LOWER MEMORY BLOCK SIZE = 2K
102             0000:81BD       MMBS:   .EQUAL  81BDH           ;MIDDLE MEMORY BLOCK SIZE = 8K
103             0000:03FD       MMST:   .EQUAL  03FDH           ;MIDDLE MEMORY START POSITION = 8K
104             0000:003D       PST:    .EQUAL  003DH           ;PERIPHERAL START ADDRESS = 0
105             0000:FFBD       UMBS:   .EQUAL  0FFBDH          ;UPPER MEMORY BLOCK SIZE = 2K
106
107
108
109                             ;       EXTERNAL PORTS FOR CONTROLLING THE FOCUS AXIS
110
111             0000:0000       P1PTA:  .EQUAL  0               ;PROM 1 PORT A
112             0000:0001       P1PTB:  .EQUAL  1               ;PROM 1 PORT B
113             0000:0002       P1PTAD: .EQUAL  2               ;PROM 1 PORT A DIRECTION
114             0000:0003       P1PTBD: .EQUAL  3               ;PROM 1 PORT B DIRECTION
115             0000:0080       P2PTA:  .EQUAL  128             ;PROM 2 PORT A
116             0000:0081       P2PTB:  .EQUAL  129             ;PROM 2 PORT B
117             0000:0082       P2PTAD: .EQUAL  130             ;PROM 2 PORT A DIRECTION
118             0000:0083       P2PTBD: .EQUAL  131             ;PROM 2 PORT B DIRECTION
119             0000:0100       RAMTC:  .EQUAL  256             ;RAM TIMER AND CONTROL
120             0000:0101       RAMPTA: .EQUAL  257             ;RAM PORT A
121             0000:0102       RAMPTB: .EQUAL  258             ;RAM PORT B
122             0000:0103       RAMPTC: .EQUAL  259             ;RAM PORT C
123             0000:0104       RAMTLO: .EQUAL  260             ;RAM TIMER LOW
124             0000:0105       RAMTHI: .EQUAL  261             ;RAM TIMER HIGH
125             0000:0180       APEXRQ: .EQUAL  384             ;APEX REQ
126             0000:0182       APEXRS: .EQUAL  386             ;APEX RESPONSE
127             0000:0182       POSM:   .EQUAL  386             ;POSITION MSB
128             0000:0181       POSL:   .EQUAL  385             ;POSITION LSB
129             0000:0184       VELM:   .EQUAL  388             ;VELOCITY MSB
130             0000:0183       VELL:   .EQUAL  387             ;VELOCITY LSB
131             0000:0186       ANAH:   .EQUAL  390             ;ANALOGS MSB
132             0000:0185       ANAL:   .EQUAL  389             ;ANALOGS LSB
133             0000:0187       DISCR:  .EQUAL  391             ;FOC DISCRETES
134             0000:018B       RELADD: .EQUAL  395             ;GET RELATIVE ADDRESS
135             0000:018C       COMML:  .EQUAL  396             ;CONTROL VALUE LSB
136             0000:018D       COMMM:  .EQUAL  397             ;CONTROL VALUE MSB
137             0000:018E       DEVACK: .EQUAL  398             ;DEVICE ACKNOWLEDGE
138             0000:018F       MONL:   .EQUAL  399             ;MONITOR DATA MSB
139             0000:0190       MONM:   .EQUAL  400             ;MONITOR DATA LSB
140             0000:0191       ADL:    .EQUAL  401             ;READ A/D LSB
141             0000:0192       ADH:    .EQUAL  402             ;READ A/D MSB
142             0000:0193       SEMCT:  .EQUAL  403             ;SELECT MOTOR CURRENT/TORQUE
143             0000:0194       STCNV:  .EQUAL  404             ;START A/D CONVERT
144             0000:0195       MODESW: .EQUAL  405             ;READ MODE SWITCH
145             0000:0196       LTCHDA: .EQUAL  406             ;LATCH DRIVE D/A
146             0000:0197       BDERL1: .EQUAL  407             ;BDS3 ERROR LSB #1
147
148
149
150 0000:0000                    DSEG:   SEGMENT
151
152                             ;       DATA STORAGE
153
154 0000:2000                           ORG     2000H
155
156             0000:2000       MEMST:  .EQUAL  $
157
```

```
158                                 ;       MONITOR STORAGE
159
160   0000:2000                     POSCEC: .BLKB   2               ;POSITION COMMAND ECHO
161   0000:2002                     POSD:   .BLKB   2               ;CURRENT POSITION DATA
162   0000:2004                     ERROR:  .BLKB   2               ;POSCEC-POSD
163   0000:2006                     SYSTEM: .BLKB   2               ;SYSTEM PARAMETERS
164   0000:2008                     FAUL1:  .BLKB   2               ;FAULT BITS SET 1
165   0000:200A                     FAUL2:  .BLKB   2               ;FAULT BITS SET 2
166   0000:200C                     ANAFL:  .BLKB   2               ;ANALOG FAULT FLAGS
167
168             0000:200E           ANADT:  .EQUAL  $               ;ANALOG STORAGE IS HERE
169
170   0000:200E                     GND1:   .BLKB   2               ;GND
171   0000:2010                     GND2:   .BLKB   2               ;GND
172   0000:2012                     GND3:   .BLKB   2               ;GND
173   0000:2014                     VEL:    .BLKB   2               ;FOCUS VELOCITY
174   0000:2016                     V15P:   .BLKB   2               ;+15V/2
175   0000:2018                     V15N:   .BLKB   2               ;-15V/2
176   0000:201A                     V5:     .BLKB   2               ;+5V
177   0000:201C                     V10P:   .BLKB   2               ;+10V
178   0000:201E                     MTEMP1: .BLKB   2               ;MOUNT TEMP 1
179   0000:2020                     V10N:   .BLKB   2               ;-10V
180   0000:2022                     MTEMP2: .BLKB   2               ;MOUNT TEMP 2
181   0000:2024                     BTEMP:  .BLKB   2               ;BIN TEMP
182
183   0000:2026                     SERVER: .BLKB   2
184   0000:2028                     VEL1:   .BLKB   2
185   0000:202A                     X22:    .BLKB   2               ;OCCURANCE COUNT
186   0000:202C                     X23:    .BLKB   2               ;COUNT WITHIN OCCURANCE
187   0000:202E                     X24:    .BLKB   2
188   0000:2030                     X25:    .BLKB   2
189   0000:2032                     X26:    .BLKB   2
190   0000:2034                     X27:    .BLKB   2
191   0000:2036                     X28:    .BLKB   2
192   0000:2038                     X29:    .BLKB   2
193   0000:203A                     X30:    .BLKB   2
194   0000:203C                     X31:    .BLKB   2
195   0000:203E                     X32:    .BLKB   2
196
197                                 ;       TEMPORARY STORAGE
198
199   0000:2040                     RAC:    .BLKB   1               ;RELATIVE CONTROL ADDRESS
200   0000:2041                     CONL:   .BLKB   1               ;CONTROL VALUE LSB
201   0000:2042                     CONM:   .BLKB   1               ;CONTROL VALUE MSB
202   0000:2043                     ACKF:   .BLKB   1               ;ACKNOWLEDGE FLAG
203   0000:2044                     COMTMP: .BLKB   2               ;TEMPORARY COMMAND STORAGE
204   0000:2046                     ADVAL:  .BLKB   2               ;A/D TEMPORARY STORAGE
205   0000:2048                     SPEED:  .BLKB   1               ;RAMP LEVEL
206   0000:2049                     BREAK1: .BLKB   2               ;RAMP UP BREAK POINT
207   0000:204B                     BREAK2: .BLKB   2               ;RAMP DOWN BREAK POINT
208   0000:204D                     POSOD:  .BLKB   2               ;OLD POSITION
209   0000:204F                     SCRLST: .BLKB   2               ;LAST 500US POSITION
210   0000:2051                     SCRCNT: .BLKB   2               ;SCREW COUNT
211   0000:2053                     EXTTMR: .BLKB   1               ;EXTERNAL SAFETY TIMER
212   0000:2054                     DEL:    .BLKB   2               ;FILTER DELTA
213   0000:2056                     FIL:    .BLKB   1               ;FILTER FAIL FLAG
214   0000:2057                     FILOVR: .BLKB   1               ;FILTER FAILURE COUNTER
```

```
0000:2058                       PHASEA: .BLKB   1                       ;CURRENT SCREW SENSOR LEVEL
0000:2059                       SCWIGN: .BLKB   1
0000:205A                       RANDOM: .BLKB   2                       ;RANDOM NUMBER
0000:205C                       SECCHC: .BLKB   1                       ;SECOND CHANCE FOR 2ND SCREW

                                ;       FLAGS

            0000:205D           FLAGST: .EQUAL  $

0000:205D                       RESCMD: .BLKB   1
0000:205E                       NAPATV: .BLKB   1
0000:205F                       NAPREQ: .BLKB   1
0000:2060                       DRVREQ: .BLKB   1
0000:2061                       MANOVR: .BLKB   1
0000:2062                       BDSRST  .BLKB   1
0000:2063                       DRVONE: .BLKB   1
0000:2064                       ADFLAG: .BLKB   1
0000:2065                       BFLAG:  .BLKB   1
0000:2066                       TFLAG0: .BLKB   1
0000:2067                       TFLAG1: .BLKB   1
0000:2068                       TFLAG2: .BLKB   1
0000:2069                       DRVOFF: .BLKB   1
0000:206A                       DRVATV: .BLKB   1
0000:206B                       RAMPOS: .BLKB   2
0000:206D                       DIR:    .BLKB   1
0000:206E                       LDIR:   .BLKB   1

            0000:0012           ENDFLG: .EQUAL  $-FLAGST

                                        ENDS


                                ;       PROGRAM EQUATES

            0000:2100           STCK:   .EQUAL  2100H                   ;STACK LOCATION
            0000:A766           DOVAL:  .EQUAL  0A766H                  ;DMA 0 CONTROL VALUE
            0000:00E8           MASKV:  .EQUAL  00E8H                   ;DMA 0, INT0, TIMER ENABLE
            0000:001C           CONRAM: .EQUAL  28                      ;?? REV/SEC CONVERGE LEVEL
            0000:8A00           LOW:    .EQUAL  00A00H .XOR. 8000H      ;LOW VALUE
            0000:7200           HIGH:   .EQUAL  0F200H .XOR. 8000H      ;HIGH VALUE

                                        LIST ON

                                ;       EXTERNAL REFERENCES

                                        GLOBAL  DSETUP, TSETUP, TSET1, CLSETAB
                                        EXTERNAL        INITAL:FAR, RSCMD:FAR


0000:0000                               ORG     0000H

0000:0000   0000 0000           TYPE0:  .LONG   INITAL                  ;DIVIDE ERROR EXCEPTION
0000:0004   0000 0000           TYPE1:  .LONG   INITAL                  ;SINGLE STEP EXCEPTION
0000:0008   5000 0000           TYPE2:  .LONG   NMI                     ;NMI
0000:000C   B001 0000           TYPE3:  .LONG   TMR2                    ;BREAKPOINT INTERRUPT
0000:0010   0000 0000           TYPE4:  .LONG   INITAL                  ;INTO EXCEPTION
0000:0014   9D00 0000           TYPE5:  .LONG   BOUND                   ;ARRAY BOUND EXCEPTION
```

```
0000:0018  0000 0000          TYPE6:  .LONG   INITAL              ;UNUSED OPCODE EXCEPTION
0000:001C  0000 0000          TYPE7:  .LONG   INITAL              ;ESC OPCODE EXCEPTION
0000:0020  0000 0000          TYPE8:  .LONG   INITAL              ;TIMER 0 INTERRUPT
0000:0024  0000 0000          TYPE9:  .LONG   INITAL              ;RESERVED
0000:0028  AE00 0000          TYPE10: .LONG   DMA0                ;DMA 0
0000:002C  0000 0000          TYPE11: .LONG   INITAL              ;DMA 1
0000:0030  8101 0000          TYPE12: .LONG   INT0                ;INT 0 INTERRUPT
0000:0034  0000 0000          TYPE13: .LONG   INITAL              ;INT 1 INTERRUPT
0000:0038  0000 0000          TYPE14: .LONG   INITAL              ;INT 2 INTERRUPT
0000:003C  0000 0000          TYPE15: .LONG   INITAL              ;INT 3 INTERRUPT
0000:0040  0000 0000          TYPE16: .LONG   INITAL              ;RESERVED
0000:0044  0000 0000          TYPE17: .LONG   INITAL              ;RESERVED
0000:0048  9F01 0000          TYPE18: .LONG   TMR1                ;TIMER 1 INTERRUPT
0000:004C  B001 0000          TYPE19: .LONG   TMR2                ;TIMER 2 INTERRUPT

                              ;       NMI INTERRUPT IS LOCATED HERE AFTER TABLE

                              ;       THIS INTERRUPT IS USED TO SEND MONITOR DATA TO
                              ;       THE STANDARD INTERFACE.

0000:0050
0000:0050                     NMI:
0000:0050  50                         PUSH    AX                  ;SAVE REGISTARS, ETC
0000:0051  52                         PUSH    DX
0000:0052  53                         PUSH    BX
0000:0053  FF 36 65 20                PUSH    BFLAG
0000:0057  C6 06 65 20 00             MOV     BFLAG,0             ;RELATIVE ADDRESS TEST
0000:005C  BA 8B 01                   MOV     DX,RELADD           ;GET RELATIVE ADDRESS
0000:005F  EC                         IN      AL,DX
0000:0060  B4 00                      MOV     AH,0
0000:0062  2E 62 06 FA 05             BOUND   AX,CS:MONRG         ;RELATIVE ADDRESS IN RANGE?
0000:0067  F6 06 65 20 01             TEST    BFLAG,1
0000:006C  74 07                      JZ      NMI1                ;IF OK
0000:006E  80 0E 08 20 40             OR      FAUL1,64            ;SET MONITOR FAULT FLAG
0000:0073  EB 19                      JMP     SHORT NEXIT         ;EXIT ROUTINE
0000:0075                     NMI1:
0000:0075  2E 2B 06 FA 05             SUB     AX,CS:MONRG         ;SUBTRACT OFFSET
0000:007A  03 C0                      ADD     AX,AX               ;TIMES 2
0000:007C  8B D8                      MOV     BX,AX               ;BX IS NOW THE INDEX VALUE
0000:007E  8B 87 00 20                MOV     AX,MEMST[BX]        ;GET MONITOR DATA
0000:0082  BA 8F 01                   MOV     DX,MONL             ;SEND IT
0000:0085  EF                         OUT     DX,AX
0000:0086  4A                         DEC     DX
0000:0087  EE                         OUT     DX,AL               ;SEND DEVICE AKNOWLEDGE
0000:0088  81 26 08 20 BF FF          AND     WORD PTR FAUL1,0FFBFH ;RESET MONITOR FAULT FLAG
0000:008E  B8 02 00           NEXIT:  MOV     AX,2                ;SPECIFIC EOI
0000:0091  BA 22 FF                   MOV     DX,EOI
0000:0094  EF                         OUT     DX,AX
0000:0095  8F 06 65 20                POP     BFLAG               ;RESTORE REGISTARS, ETC
0000:0099  5B                         POP     BX
0000:009A  5A                         POP     DX
0000:009B  58                         POP     AX
0000:009C  CF                         IRET

                              ;       THIS INTERRUPT SETS THE OUT OF BOUNDS FLAG

0000:009D                     BOUND:
```

```
0000:009D  50                      PUSH    AX                      ;SAVE REGISTARS
0000:009E  52                      PUSH    DX
0000:009F  C6 06 65 20 01          MOV     BFLAG,1                 ;SET OUT OF BOUNDS FLAG
0000:00A4  B8 05 00                MOV     AX,5                    ;SPECIFIC EOI
0000:00A7  BA 22 FF                MOV     DX,EOI
0000:00AA  EF                      OUT     DX,AX
0000:00AB  5A                      POP     DX                      ;RESTORE REGISTARS
0000:00AC  58                      POP     AX
0000:00AD  CF                      IRET

                           ;       THIS INTERRUPT COMPLETES THE PROCESSING OF A COMMAND
                           ;       INPUTTED FROM THE STANDARD INTERFACE

0000:00AE                  DMA0:
0000:00AE  50                      PUSH    AX                      ;SAVE REGISTARS, ETC
0000:00AF  52                      PUSH    DX
0000:00B0  51                      PUSH    CX
0000:00B1  53                      PUSH    BX
0000:00B2  56                      PUSH    SI
0000:00B3  FF 36 65 20             PUSH    BFLAG
0000:00B7  C6 06 65 20 00          MOV     BFLAG,0                 ;ASSUME IN BOUNDS
0000:00BC  8A 1E 40 20             MOV     BL,RAC                  ;GET RELATIVE ADDRESS
0000:00C0  B7 00                   MOV     BH,0                    ;STRIP UPPER NYBBLE
0000:00C2  2E 62 1E DC 05          BOUND   BX,CS:CMDRG             ;VALID COMMAND?
0000:00C7  F6 06 65 20 01          TEST    BFLAG,1                 ;IN BOUND?
0000:00CC  74 08                   JZ      DMA01                   ;IF OK
0000:00CE  80 0E 08 20 20          OR      FAUL1,32                ;SET COMMAND INVALID
0000:00D3  E9 8B 00                JMP     DEXIT
0000:00D6                  DMA01:
0000:00D6  81 26 08 20 DF FF       AND     WORD PTR FAUL1,0FFDFH   ;CLEAR COMMAND INVALID
0000:00DC  2E 2B 1E DC 05          SUB     BX,CS:CMDRG             ;SUBTRACT OFFSET
0000:00E1  03 DB                   ADD     BX,BX                   ;TIMES 2
0000:00E3  2E 8B 87 E0 05          MOV     AX,CS:CMDTBL[BX]        ;GET NEW PROGRAM POINT
0000:00E8  FF E0                   JMP     AX                      ;GO THERE

                           ;       THE NEXT FIVE ROUTINES ARE THE COMMAND PROCESSING ENTRY POINTS

0000:00EA                  POSCMD:
0000:00EA  A1 41 20                MOV     AX,CONL                 ;GET COMMANDED POSITION
0000:00ED  35 00 80                XOR     AX,8000H                ;FOR BOUND TEST
0000:00F0  2E 62 06 FE 05          BOUND   AX,CS:POSRG             ;POSITION WITHIN RANGE?
0000:00F5  35 00 80                XOR     AX,8000H                ;UNDO AFTER TEST
0000:00F8  F6 06 65 20 01          TEST    BFLAG,1
0000:00FD  74 07                   JZ      POSCM1                  ;IF IN RANGE
0000:00FF  80 0E 08 20 80          OR      FAUL1,128               ;SET OPERATOR FAULT
0000:0104  EB 5B                   JMP     SHORT DEXIT
0000:0106  80 26 08 20 7F  POSCM1: AND     FAUL1,7FH               ;RESET OPERATOR FAULT
0000:010B  A3 44 20                MOV     COMTMP,AX               ;SAVE COMMANDED POSITION HERE
0000:010E  C6 06 60 20 01          MOV     DRVREQ,1                ;SET DRIVE REQUEST
0000:0113  C6 06 63 20 00          MOV     DRVONE,0                ;CLEAR SECOND TRY FLAG
0000:0118  EB 47                   JMP     SHORT DEXIT

0000:011A                  RECMD:
0000:011A  C6 06 5D 20 01          MOV     RESCMD,1                ;SET SOFT RESET REQUEST
0000:011F  EB 40                   JMP     SHORT DEXIT

0000:0121                  NAPCMD:
```

```
0000:0121  F6 06 41 20 01          TEST   CONL,1              ;TEST FOR SET / RESET
0000:0126  75 0C                   JNZ    NAPCM1
0000:0128  C6 06 5E 20 00          MOV    NAPATV,0            ;IF CLEAR NAP
0000:012D  80 26 06 20 FE          AND    SYSTEM,0FEH
0000:0132  EB 2D                   JMP    SHORT DEXIT
0000:0134  F6 06 5E 20 01  NAPCM1: TEST   NAPATV,1            ;ALREADY MAPPED?
0000:0139  75 26                   JNZ    DEXIT
0000:013B  C6 06 5F 20 01          MOV    NAPREQ,1            ;SET NAP REQUEST
0000:0140  EB 1F                   JMP    SHORT DEXIT

0000:0142                  MANCMD:
0000:0142  F6 06 41 20 01          TEST   CONL,1              ;TEST FOR SET / RESET
0000:0147  75 0C                   JNZ    MANCM1
0000:0149  C6 06 61 20 00          MOV    MANOVR,0            ;IF CLEAR OVER RIDE
0000:014E  80 26 06 20 F7          AND    SYSTEM,0F7H
0000:0153  EB 0C                   JMP    SHORT DEXIT
0000:0155  C6 06 61 20 01  MANCM1: MOV    MANOVR,1            ;SET MANUEL OVER-RIDE
0000:015A  EB 05                   JMP    SHORT DEXIT

0000:015C  C6 06 62 20 01  BDSCMD: MOV    BDSRST,1            ;SET BDS3 SERVO RESET

0000:0161  BE EE 05        DEXIT:  MOV    SI,CS:OFFSET DSETUP ;RE-INITIALIZE DMA 0
0000:0164  BA C0 FF                MOV    DX,DOSPL
0000:0167  B9 06 00                MOV    CX,6
0000:016A  6F              DSET:   OUTSW
0000:016B  83 C2 02                ADD    DX,2
0000:016E  E2 FA                   LOOP   DSET
0000:0170  B8 0A 00                MOV    AX,10               ;SPECIFIC EOI
0000:0173  BA 22 FF                MOV    DX,EOI
0000:0176  EF                      OUT    DX,AX
0000:0177  8F 06 65 20             POP    BFLAG               ;RESTORE REGISTARS, ETC
0000:017B  5E                      POP    SI
0000:017C  5B                      POP    BX
0000:017D  59                      POP    CX
0000:017E  5A                      POP    DX
0000:017F  58                      POP    AX
0000:0180  CF                      IRET

                           ;       THIS INTERRUPT ROUTINE LOADS THE ANALOG TO DIGITAL VALUES
                           ;       AFTER CONVERSION

0000:0181                  INT0:
0000:0181  50                      PUSH   AX                  ;SAVE REGISTARS
0000:0182  52                      PUSH   DX
0000:0183  BA 91 01                MOV    DX,ADL              ;GET A/D VALUE
0000:0186  ED                      IN     AX,DX
0000:0187  C1 E8 04                SHR    AX,4                ;PUT INTO POSITION
0000:018A  35 00 08                XOR    AX,0800H            ;TWOS COMPLEMENT
0000:018D  A3 46 20                MOV    ADVAL,AX            ;STORE HERE
0000:0190  C6 06 64 20 01          MOV    ADFLAG,1            ;SET ADFLAG
0000:0195  B8 0C 00                MOV    AX,12               ;SPECIFIC EOI
0000:0198  BA 22 FF                MOV    DX,EOI
0000:019B  EF                      OUT    DX,AX
0000:019C  5A                      POP    DX                  ;RESTORE REGISTARS
0000:019D  58                      POP    AX
0000:019E  CF                      IRET
```

```
0000:019F                       TMR1:
0000:019F  50                           PUSH    AX                      ;SAVE REGISTARS
0000:01A0  52                           PUSH    DX
0000:01A1  C6 06 67 20 01               MOV     TFLAG1,1                ;SET TIMER 1 FLAG
0000:01A6  B8 08 00                     MOV     AX,8                    ;SPECIFIC EOI
0000:01A9  BA 22 FF                     MOV     DX,EOI
0000:01AC  EF                           OUT     DX,AX
0000:01AD  5A                           POP     DX                      ;RESTORE REGISTARS
0000:01AE  58                           POP     AX
0000:01AF  CF                           IRET
0000:01B0                       TMR2:
0000:01B0  50                           PUSH    AX                      ;SAVE REGISTARS
0000:01B1  53                           PUSH    BX
0000:01B2  51                           PUSH    CX
0000:01B3  52                           PUSH    DX
0000:01B4  FF 36 65 20                  PUSH    BFLAG
0000:01B8  C6 06 53 20 19               MOV     EXITMR,25
0000:01BD  F6 06 6A 20 01               TEST    DRVATV,1
0000:01C2  75 06                        JNZ     GO
0000:01C4  E8 01 04                     CALL    TMROFF
0000:01C7  E9 E2 02                     JMP     TMR2EX
0000:01CA  F6 06 69 20 01       GO:     TEST    DRVOFF,1                ;GO INTO SHUTDOWN?
0000:01CF  74 03                        JZ      TMR21                   ;IF NOT SHUT DOWN
0000:01D1  E9 7E 02                     JMP     OFFIT                   ;IF SHUT DOWN
0000:01D4  E4 81                TMR21:  IN      AL,P2PTB                ;GET E-STOP, DRIVE LOCKOUT
0000:01D6  24 18                        AND     AL,18H                  ;STRIP JUNK
0000:01D8  34 10                        XOR     AL,10H                  ;INVERT DRIVE LOCKOUT
0000:01DA  74 03                        JZ      TMR210                  ;IF EVERYTHING OK
0000:01DC  E9 AA 02                     JMP     OFFIT3                  ;ELSE KILL DRIVE
0000:01DF  E8 54 03             TMR210: CALL    DSTOR                   ;GET LATEST DATA
0000:01E2  BA 87 01                     MOV     DX,DISCR                ;POINT TO DISCRETES
0000:01E5  EC                           IN      AL,DX                   ;GET INFO
0000:01E6  F6 06 6E 20 01               TEST    LDIR,1                  ;TEST DIRECTION
0000:01EB  74 07                        JZ      TMR21A                  ;IF POSITIVE
0000:01ED  24 03                        AND     AL,3                    ;DOWN LIMIT?
0000:01EF  74 0A                        JZ      TMR21B                  ;IF NOT
0000:01F1  E9 5E 02                     JMP     OFFIT                   ;IF IN DOWN LIMIT
0000:01F4  24 0C                TMR21A: AND     AL,0CH                  ;UP LIMIT?
0000:01F6  74 03                        JZ      TMR21B                  ;IF NOT
0000:01F8  E9 57 02                     JMP     OFFIT                   ;IF IN UP LIMIT
0000:01FB  F6 06 59 20 01       TMR21B: TEST    SCWIGN,1                ;IGNORE?
0000:0200  74 03                        JZ      TMR2BB                  ;NO
0000:0202  E9 7C 00                     JMP     TMR22                   ;YES
0000:0205  EC                   TMR2BB: IN      AL,DX                   ;GET SCREW LEVEL
0000:0206  24 20                        AND     AL,32                   ;JUST SCREW INFO
0000:0208  30 06 58 20                  XOR     PHASEA,AL               ;TRANSITION?
0000:020C  A2 58 20                     MOV     PHASEA,AL               ;UPDATE
0000:020F  74 09                        JZ      TMR21C                  ;IF NO TRANSITION
0000:0211  C7 06 51 20 00 00            MOV     WORD PTR SCRCNT,0       ;SCREW COUNT 0
0000:0217  E9 67 00                     JMP     TMR22                   ;CONTINUE
0000:021A  A1 4F 20             TMR21C: MOV     AX,SCRLST               ;GET LAST OK POSITION
0000:021D  2B 06 02 20                  SUB     AX,POSD                 ;DEVELOP DELTA
0000:0221  C6 06 65 20 00               MOV     BFLAG,0                 ;FOR BOUND CHECK
0000:0226  2E 62 06 EA 05               BOUND   AX,CS:DELTA             ;IN BOUNDS?
0000:022B  F6 06 65 20 01               TEST    BFLAG,1
0000:0230  74 1C                        JZ      TMR2CC                  ;IF IN BOUNDS
```

```
0000:0232  C6 06 56 20 01          MOV   FIL,1                ;SET FIL BLOWN
0000:0237  A1 54 20                MOV   AX,DEL               ;GET LAST OK DELTA
0000:023A  FE 06 57 20             INC   FILOVR               ;INC FILTER NUMBER FAIL COUNT
0000:023E  80 3E 57 20 0A          CMP   FILOVR,10            ;TOO MANY?
0000:0243  75 0E                   JNZ   TMR2CD               ;IF NOT
0000:0245  81 0E 08 20 00 0A       OR    WORD PTR FAUL1,00A00H ;SET FAULT
0000:024B  E9 3B 02                JMP   OFFIT3               ;MAJOR PROBLEM
0000:024E  C6 06 56 20 00    TMR2CC: MOV   FIL,0               ;RESET FIL
0000:0253  01 06 51 20       TMR2CD: ADD   SCRCNT,AX           ;ADD IN DELTA
0000:0257  A3 54 20                MOV   DEL,AX               ;UPDATE DELTA
0000:025A  A1 02 20                MOV   AX,POSD              ;UPDATE POSITION
0000:025D  A3 4F 20                MOV   SCRLST,AX
0000:0260  C6 06 65 20 00          MOV   BFLAG,0              ;SET IN BOUNDS
0000:0265  A1 51 20                MOV   AX,SCRCNT
0000:0268  2E 62 06 0E 06          BOUND AX,CS:SCREW          ;WITHIN +/- 38 COUNTS
0000:026D  F6 06 65 20 01          TEST  BFLAG,1              ;IN BOUNDS?
0000:0272  74 0D                   JZ    TMR22                ;IF OK
0000:0274  81 0E 08 20 00 82 TMR21D: OR   WORD PTR FAUL1,8200H ;SET FAULT BITS
0000:027A  FE 06 5C 20             INC   SECCHC               ;INC SECOND CHANCE
0000:027E  E9 08 02                JMP   OFFIT3               ;KILL DRIVE
0000:0281  F6 06 56 20 01    TMR22:  TEST  FIL,1               ;BAD AREA
0000:0286  74 03                   JZ    TMR220
0000:0288  E9 21 02                JMP   TMR2EX
0000:028B  A0 6D 20          TMR220: MOV   AL,DIR               ;GET DIRECTION
0000:028E  3A 06 6E 20             CMP   AL,LDIR              ;EQUAL?
0000:0292  74 08                   JZ    TMR22A               ;IF EQUAL
0000:0294  C6 06 6B 20 06          MOV   RAMPOS,6
0000:0299  E9 15 00                JMP   VERGIT               ;GOTO CONVERGE
0000:029C  80 3E 48 20 0A    TMR22A: CMP   BYTE PTR SPEED,10   ;MOTION ANALYSIS?
0000:02A1  75 03                   JNZ   TMR23                ;IF NOT
0000:02A3  E9 1A 02                JMP   MOTION
0000:02A6  8B 1E 6B 20       TMR23:  MOV   BX,RAMPOS           ;GET CURRENT RAMP
0000:02AA  2E 8B 87 06 06          MOV   AX,CS:RAMPTL[BX]     ;GET INDEX
0000:02AF  FF E0                   JMP   AX                   ;GO THERE
0000:02B1
0000:02B1                    VERGIT:
0000:02B1  8B 1E 04 20             MOV   BX,ERROR             ;ARE WE CLOSE?
0000:02B5  C6 06 65 20 00          MOV   BFLAG,0              ;SET IN BOUNDS
0000:02BA  2E 62 1E D8 05          BOUND BX,CS:CLSETAB        ;CLOSE?
0000:02BF  F6 06 65 20 01          TEST  BFLAG,1
0000:02C4  75 3A                   JNZ   VERGI1               ;IF NOT
0000:02C6  E5 00                   IN    AX,P1PTA             ;FINAL RAMP TO STOP
0000:02C8  50                      PUSH  AX                   ;SAVE IT
0000:02C9  25 FF 0F                AND   AX,0FFFH             ;JUST RAMP VALUE
0000:02CC  BB 04 00                MOV   BX,4                 ;PREPARE TO DECREASE
0000:02CF  F6 06 6E 20 01          TEST  LDIR,1               ;CHECK DIRECTION
0000:02D4  75 02                   JNZ   VERGI0               ;IF POSITIVE
0000:02D6  F7 DB                   NEG   BX                   ;IF NEGATIVE
0000:02D8  2B C3             VERGI0: SUB   AX,BX
0000:02DA  25 FF 0F                AND   AX,0FFFH             ;JUST RAMP VALUE
0000:02DD  5B                      POP   BX                   ;RESTORE
0000:02DE  81 E3 00 F0             AND   BX,0F000H            ;JUST CONTROL
0000:02E2  3D 00 00                CMP   AX,0                 ;DONE?
0000:02E5  75 0E                   JNE   VERGIA               ;NOT DONE
0000:02E7  81 26 08 20 FF FD       AND   WORD PTR FAUL1,0FDFFH ;RESET DRIVE FAULT FLAG
0000:02ED  C6 06 5C 20 00          MOV   BYTE PTR SECCHC,0    ;CLEAR 2ND SCREW SECOND CHANCE
0000:02F2  E9 94 01                JMP   OFFIT3               ;COMPLETE
```

```
0000:02F5  0B C3              VERGIA: OR      AX,BX                   ;MERGE
0000:02F7  E7 00                      OUT     P1PTA,AX                ;SEND NEW RAMP
0000:02F9  BA 96 01                   MOV     DX,LTCHDA               ;NOW LATCH D/A
0000:02FC  EE                         OUT     DX,AL                   ;NEW RAMP COMPLETE
0000:02FD  E9 AC 01                   JMP     TMR2EX

0000:0300                     VERGI1:
0000:0300  A0 6D 20                   MOV     AL,DIR                  ;CHECK FOR DIRECTION SWITCH
0000:0303  3A 06 6E 20                CMP     AL,LDIR
0000:0307  75 07                      JNE     VERGIB                  ;WE OVER SHOT
0000:0309  BA 96 01                   MOV     DX,LTCHDA               ;NOW LATCH D/A
0000:030C  EE                         OUT     DX,AL                   ;OLD RAMP COMPLETE
0000:030D  E9 9C 01                   JMP     TMR2EX                  ;IF NOT SWITCHED
0000:0310  E5 00              VERGIB: IN      AX,P1PTA                ;FINAL RAMP TO STOP
0000:0312  50                         PUSH    AX                      ;SAVE IT
0000:0313  25 FF 0F                   AND     AX,0FFFH                ;JUST RAMP VALUE
0000:0316  BB 04 00                   MOV     BX,4                    ;PREPARE TO DECREASE
0000:0319  F6 06 6E 20 01             TEST    LDIR,1                  ;CHECK DIRECTION
0000:031E  75 02                      JNZ     VERGIC                  ;IF POSITIVE
0000:0320  F7 DB                      NEG     BX                      ;IF NEGATIVE
0000:0322  2B C3              VERGIC: SUB     AX,BX
0000:0324  25 FF 0F                   AND     AX,0FFFH                ;JUST RAMP VALUE
0000:0327  5B                         POP     BX                      ;RESTORE
0000:0328  81 E3 00 F0                AND     BX,0F000H               ;JUST CONTROL
0000:032C  50                         PUSH    AX                      ;SAVE NEW RAMP
0000:032D  0B C3                      OR      AX,BX                   ;MERGE
0000:032F  E7 00                      OUT     P1PTA,AX                ;SEND NEW RAMP
0000:0331  BA 96 01                   MOV     DX,LTCHDA               ;NOW LATCH D/A
0000:0334  EE                         OUT     DX,AL                   ;NEW RAMP COMPLETE
0000:0335  58                         POP     AX                      ;RESTORE NEW RAMP
0000:0336  3D 00 00                   CMP     AX,0                    ;DONE?
0000:0339  74 03                      JE      VERGI2                  ;DONE
0000:033B  E9 6E 01                   JMP     TMR2EX

0000:033E                     VERGI2:
0000:033E  F6 06 63 20 01             TEST    DRVONE,1                ;ON SECOND TRY?
0000:0343  75 1A                      JNZ     VERGI3                  ;IF SECOND TRY·
0000:0345  C6 06 63 20 01             MOV     DRVONE,1                ;SET SECOND TRY FLAG
0000:034A  C6 06 6A 20 00             MOV     DRVATV,0                ;SET DRIVE TO NON ACTIVE
0000:034F  C6 06 60 20 01             MOV     DRVREQ,1                ;RESTART COMMAND
0000:0354  8B 1E 00 20                MOV     BX,POSCEC               ;GET COMTMP = POSCEC
0000:0358  89 1E 44 20                MOV     COMTMP,BX
0000:035C  E9 2A 01                   JMP     OFFIT3                  ;LEAVE
0000:035F                     VERGI3:
0000:035F  81 0E 08 20 00 02          OR      WORD PTR FAUL1,200H     ;SET DRIVE FAULT
0000:0365  E9 21 01                   JMP     OFFIT3                  ;STOP

0000:0368  C6 06 65 20 00     ZIPUP:  MOV     BFLAG,0                 ;RESET BOUND FLAG
0000:036D  FE 0E 48 20                DEC     SPEED
0000:0371  75 34                      JNZ     ZIPUP2                  ;IF NO CHANGE IN RAMP
0000:0373  C6 06 48 20 14             MOV     SPEED,20                ;RESET SPEED
0000:0378  E5 00                      IN      AX,P1PTA                ;GET CURRENT RAMP POINT
0000:037A  50                         PUSH    AX                      ;SAVE IT
0000:037B  25 FF 0F                   AND     AX,0FFFH                ;STRIP CONTROL NYBBLE
0000:037E  BB 04 00                   MOV     BX,4                    ;DO NEXT RAMP STEP
0000:0381  F6 06 6E 20 01             TEST    LDIR,1                  ;NEGATIVE?
0000:0386  75 02                      JNZ     ZIPUP1                  ;IF POSITIVE
```

```
0000:0388  F7 DB              NEG   BX              ;REVESRE SIGN
0000:038A  03 C3    ZIPUP1:   ADD   AX,BX           ;NEW RAMP
0000:038C  25 FF 0F           AND   AX,0FFFH        ;STRIP UPPER NYEBLE
0000:038F  C1 C0 04           ROL   AX,4            ;ROTATE FOR BOUND TEST
0000:0392  2E 62 06 02 06     BOUND AX,CS:RAMPLM
0000:0397  C1 C3 04           ROR   AX,4            ;PUT BACI
0000:039A  5B                 POP   BX              ;GET OLD CONTROL BITS
0000:039B  81 E3 00 F0        AND   BX,0F000H       ;STRIP OLD RAMP
0000:039F  0B C3              OR    AX,BX           ;MERGE
0000:03A1  E7 00              OUT   P1PTA,AX        ;SEND NEW RAMP
0000:03A3  BA 96 01           MOV   DX,LTCHDA       ;NOW LATCH D/A
0000:03A6  EE                 OUT   DX,AL           ;NEW RAMP COMPLETE
0000:03A7  F6 06 65 20 01 ZIPUP2: TEST  BFLAG,1     ;OUT OF BOUNDS?
0000:03AC  75 13              JNZ   ZIPUP3          ;IF OUT
0000:03AE  A1 02 20           MOV   AX,POSD         ;CURRENT POSITION
0000:03B1  F6 06 6E 20 01     TEST  LDIR,1          ;CHECK DIRECTION
0000:03B6  75 11              JNZ   ZIPUP4          ;IF NEGATIVE
0000:03B8  3B 06 49 20        CMP   AX,BREAK1       ;RAMP INTO MAIN?
0000:03BC  73 03              JNC   ZIPUP3          ;GO INTO MAIN
0000:03BE  E9 EB 00           JMP   TMR2EX          ;NOPE
0000:03C1  80 06 6B 20 02 ZIPUP3: ADD  RAMPOS,2     ;SET MAIN
0000:03C6  E9 E3 00           JMP   TMR2EX          ;NOW LEAVE
0000:03C9  3B 06 49 20    ZIPUP4: CMP   AX,BREAK1   ;RAMP INTO MAIN?
0000:03CD  72 F2              JC    ZIPUP3          ;GO INTO MAIN
0000:03CF  E9 DA 00           JMP   TMR2EX          ;NOPE

0000:03D2                 MAINNY:
0000:03D2  FE 0E 48 20        DEC   SPEED
0000:03D6  75 05              JNZ   MAINN0          ;FOR MOTION ANALYSIS
0000:03D8  C6 06 48 20 14     MOV   SPEED,20        ;RESET SPEED
0000:03DD  A1 02 20       MAINN0: MOV   AX,POSD     ;CURRENT POSITION
0000:03E0  F6 06 6E 20 01     TEST  LDIR,1          ;CHECK DIRECTION
0000:03E5  75 0F              JNZ   MAINN2          ;IF NEGATIVE
0000:03E7  3B 06 4B 20        CMP   AX,BREAK2       ;MAIN INTO RAMP DOWN?
0000:03EB  73 02              JNC   MAINN1          ;GO INTO RAMP DOWN
0000:03ED  EB 0D              JMP   SHORT MAINN3    ;NOPE
0000:03EF  80 06 6B 20 02 MAINN1: ADD  RAMPOS,2     ;SET RAMP DOWN
0000:03F4  EB 06              JMP   SHORT MAINN3    ;NOW LEAVE
0000:03F6  3B 06 4B 20    MAINN2: CMP   AX,BREAK2   ;MAIN INTO RAMP DOWN?
0000:03FA  72 F3              JC    MAINN1          ;GO INTO RAMP DOWN
0000:03FC  BA 96 01       MAINN3: MOV   DX,LTCHDA   ;NOW LATCH D/A
0000:03FF  EE                 OUT   DX,AL           ;OLD RAMP COMPLETE
0000:0400  E9 A9 00           JMP   TMR2EX          ;DONE

0000:0403  FE 0E 48 20    ZIPDWN: DEC   SPEED
0000:0407  75 46              JNZ   ZIPDN3          ;IF NO CHANGE IN RAMP
0000:0409  C6 06 48 20 14     MOV   SPEED,20        ;RESET SPEED
0000:040E  E5 00              IN    AX,P1PTA        ;GET CURRENT RAMP POINT
0000:0410  50                 PUSH  AX              ;SAVE IT
0000:0411  25 FF 0F           AND   AX,0FFFH        ;STRIP CONTROL NYEBLE
0000:0414  BB 04 00           MOV   BX,4            ;DO NEXT RAMP STEP
0000:0417  F6 06 6E 20 01     TEST  LDIR,1          ;NEGATIVE?
0000:041C  75 02              JNZ   ZIPDN1          ;IF POSITIVE
0000:041E  F7 DB              NEG   BX              ;REVESRE SIGN
0000:0420  2B C3    ZIPDN1:   SUB   AX,BX           ;NEW RAMP
0000:0422  25 FF 0F           AND   AX,0FFFH        ;STRIP UPPER NYBBLE
0000:0425  5B                 POP   BX              ;GET OLD CONTROL BITS
```

```
671  0000:0426  74 15              JZ      ZIPDN2          ;IF ZERO
672  0000:0428  81 E3 00 F0        AND     BX,0F000H       ;STRIP OLD RAMP
673  0000:042C  0B C3              OR      AX,BX           ;MERGE
674  0000:042E  E7 C0              OUT     P1PTA,AX        ;SEND NEW RAMP
675  0000:0430  BA 96 01           MOV     DX,LTCHDA       ;NOW LATCH D/A
676  0000:0433  EE                 OUT     DX,AL           ;NEW RAMP COMPLETE
677  0000:0434  F6 06 6E 20 01     TEST    LDIR,1          ;CHECK DIRECTION
678  0000:0439  75 02              JNZ     ZIPDN2          ;IF POSITIVE
679  0000:043B  F7 D8              NEG     AX              ;INVERT FOR CHECK
680  0000:043D  25 FF 0F   ZIPDN2: AND     AX,0FFFH        ;CURRENT RAMP
681  0000:0440  3D 1C 00           CMP     AX,CONRAM       ;RAMP INTO CONVERGE?
682  0000:0443  73 67              JNC     TMR2EX          ;NOPE
683  0000:0445  C6 06 48 20 01     MOV     SPEED,1         ;RESET SPEED
684  0000:044A  80 06 6B 20 02     ADD     RAMPOS,2        ;SET CONVERG
685  0000:044F  E9 5A 00   ZIPDN3: JMP     TMR2EX          ;NOW LEAVE
686
687  0000:0452  E5 00      OFFIT:  IN      AX,P1PTA        ;GET CURRENT RAMP POINT
688  0000:0454  FE 0E 48 20        DEC     SPEED
689  0000:0458  75 27              JNZ     OFFIT2          ;IF NO CHANGE IN RAMP
690  0000:045A  C6 06 48 20 14     MOV     SPEED,20        ;RESET SPEED
691  0000:045F  50                 PUSH    AX              ;SAVE IT
692  0000:0460  25 FF 0F           AND     AX,0FFFH        ;STRIP CONTROL NYBBLE
693  0000:0463  BB 04 00           MOV     BX,4            ;DO NEXT RAMP STEP
694  0000:0466  F6 06 6E 20 01     TEST    LDIR,1          ;NEGATIVE?
695  0000:046B  75 02              JNZ     OFFIT1          ;IF POSITIVE
696  0000:046D  F7 DB              NEG     BX              ;REVESRE SIGN
697  0000:046F  2B C3      OFFIT1: SUB     AX,BX           ;NEW RAMP
698  0000:0471  25 FF 0F           AND     AX,0FFFH        ;STRIP UPPER NYBBLE
699  0000:0474  5B                 POP     BX              ;GET OLD CONTROL BITS
700  0000:0475  81 E3 00 F0        AND     BX,0F000H       ;STRIP OLD RAMP
701  0000:0479  0B C3              OR      AX,BX           ;MERGE
702  0000:047B  E7 00              OUT     P1PTA,AX        ;SEND NEW RAMP
703  0000:047D  BA 96 01           MOV     DX,LTCHDA       ;NOW LATCH D/A
704  0000:0480  EE                 OUT     DX,AL           ;NEW RAMP COMPLETE
705  0000:0481  25 FF 0F   OFFIT2: AND     AX,0FFFH        ;CURRENT RAMP
706  0000:0484  3D 00 00           CMP     AX,0            ;GOT TO 0?
707  0000:0487  75 23              JNE     TMR2EX          ;NOPE
708  0000:0489  E8 3C 01   OFFIT3: CALL    TMROFF          ;KILL TIMER
709  0000:048C  B8 00 C0           MOV     AX,0C000H       ;KILL DRIVE
710  0000:048F  E7 00              OUT     P1PTA,AX
711  0000:0491  B8 08 00           MOV     AX,8            ;SPECIFIC EOI
712  0000:0494  BA 22 FF           MOV     DX,EOI
713  0000:0497  EF                 OUT     DX,AX
714  0000:0498  E8 E9 00           CALL    BREOFF          ;KILL BRAKE
715  0000:049B  C6 06 6A 20 00     MOV     DRVATV,0        ;KILL DRIVE ACTIVE FLAG
716  0000:04A0  C6 06 69 20 00     MOV     DRVOFF,0        ;KILL DRIVE OFF FLAG
717  0000:04A5  E5 80              IN      AX,P2PTA
718  0000:04A7  25 97 FF           AND     AX,0FF97H       ;KILL SPECIFIC LIGHTS
719  0000:04AA  E7 80              OUT     P2PTA,AX
720  0000:04AC  B8 08 00   TMR2EX: MOV     AX,8            ;SPECIFIC EOI
721  0000:04AF  BA 22 FF           MOV     DX,EOI
722  0000:04B2  EF                 OUT     DX,AX
723  0000:04B3  B8 03 00           MOV     AX,3            ;AGAIN
724  0000:04B6  EF                 OUT     DX,AX
725  0000:04B7  8F 06 65 20        POP     BFLAG
726  0000:04BB  5A                 POP     DX              ;RESTORE REGISTARS
727  0000:04BC  59                 POP     CX
```

```
0000:04BD 5B                    POP     BX
0000:04BE 58                    POP     AX
0000:04BF CF                    IRET
0000:04C0               MOTION:
0000:04C0 FB                    STI                     ;ENABLE INTERRUPTS
0000:04C1 E5 80                 IN      AX,P2PTA        ;GET PORT 2
0000:04C3 25 F8 FF              AND     AX,0FFF8H       ;STRIP OLD A/D REQ
0000:04C6 0D 01 00              OR      AX,1            ;REQUEST VELOCITY
0000:04C9 E7 80                 OUT     P2PTA,AX
0000:04CB BA 94 01              MOV     DX,STCNV        ;START CONVERSION
0000:04CE EE                    OUT     DX,AL
0000:04CF C6 06 64 20 00        MOV     ADFLAG,0        ;RESET FLAG
0000:04D4 FE 0E 48 20           DEC     SPEED           ;UPDATE SPEED
0000:04D8 BA 97 01              MOV     DX,BDERL1       ;GET SERVO AMP FAULTS
0000:04DB EC                    IN      AL,DX
0000:04DC F6 D0                 NOT     AL              ;COMPLEMENT THEM
0000:04DE A2 0A 20              MOV     FAUL2,AL        ;STORE THEM
0000:04E1 24 FD                 AND     AL,0FDH         ;IGNORE FOLDBACK
0000:04E3 74 08                 JZ      MOTI01          ;ANY PROBLEMS?
0000:04E5 C6 06 69 20 01        MOV     DRVOFF,1        ;YES, SO KILL DRIVE
0000:04EA E9 BF FF              JMP     TMR2EX
0000:04ED               MOTI01:
0000:04ED 8B 1E 02 20           MOV     BX,POSD         ;GET CURRENT POSITION
0000:04F1 A1 4D 20              MOV     AX,POSDOD       ;GET OLD POSITION
0000:04F4 89 1E 4D 20           MOV     POSDOD,BX       ;CURRENT = OLD
0000:04F8 2B D8                 SUB     BX,AX           ;GET DIFFERENCE
0000:04FA 79 02                 JNS     MOTI02          ;IF POSITIVE
0000:04FC F7 DB                 NEG     BX              ;MAKE POSITIVE
0000:04FE 8B 16 28 20   MOTI02: MOV     DX,VEL1         ;GET PREVIOUS VELOCITY
0000:0502 B1 19                 MOV     CL,25           ;PRELOAD DIVIDE VALUE
0000:0504 F6 06 64 20 01MOTI03: TEST    ADFLAG,1        ;ANALOG READY?
0000:0509 74 F9                 JZ      MOTI03          ;IF NOT
0000:050B A1 46 20              MOV     AX,ADVAL        ;GET CURRENT VELOCITY
0000:050E F6 C4 08              TEST    AH,8            ;MINUS?
0000:0511 74 02                 JZ      MOTI04          ;IF MINUS
0000:0513 F7 D8                 NEG     AX              ;MAKE POSITIVE
0000:0515 25 FF 0F      MOTI04: AND     AX,0FFFH        ;STRIP UNWANTED BITS
0000:0518 F6 F1                 DIV     CL              ;DIVIDE BY 20
0000:051A B4 00                 MOV     AH,0            ;KILL REMAINDER
0000:051C A3 28 20              MOV     VEL1,AX         ;CURRENT = PREVIOUS
0000:051F 03 C2                 ADD     AX,DX           ;ADD IN PREVIOUS VELOCITY
0000:0521 83 C3 04              ADD     BX,4            ;4 COUNT SLOP
0000:0524 3B C3                 CMP     AX,BX           ;OK MOTION?
0000:0526 7E 0B                 JLE     MOTI05          ;IF OK
0000:0528 C6 06 69 20 01        MOV     DRVOFF,1
0000:052D 81 0E 08 20 00 04     OR      WORD PTR FAUL1,1024     ;SET MOTION FAULT
0000:0533 E9 76 FF      MOTI05: JMP     TMR2EX
0000:0536               DSTOR:
0000:0536 32 C0                 XOR     AL,AL
0000:0538 BA 80 01              MOV     DX,APEXRQ       ;GET APEX DATA
0000:053B EE                    OUT     DX,AL
0000:053C 83 C2 02              ADD     DX,2            ;POINT TO APEXRS
0000:053F B9 0C 00              MOV     CX,12           ;12 TRIALS
0000:0542               DSTOR1:
0000:0542 EC                    IN      AL,DX           ;SEE IF DATA READY
```

```
0000:0543  24 C0                    AND     AL,0C0H             ;STRIP DATA
0000:0545  3C 80                    CMP     AL,80H              ;LOOK FOR PATTERN
0000:0547  74 0A                    JZ      DSTOR2              ;IF READY
0000:0549  E2 F7                    LOOP    DSTOR1              ;IF NOT
0000:054B  80 0E 08 20 01           OR      FAUL1,1             ;SET APEX DEAD
0000:0550  E9 30 00                 JMP     DSTOR3              ;SKIP APEX GATHERING
0000:0553                   DSTOR2:
0000:0553  80 26 08 20 FE           AND     FAUL1,0FEH          ;SET APEX OK
0000:0558  4A                       DEC     DX
0000:0559  ED                       IN      AX,DX               ;GET POSITION
0000:055A  8B D8                    MOV     BX,AX               ;PUT INTO BX
0000:055C  C1 E3 02                 SHL     BX,2                ;GET RID OF RESPONSE BITS
0000:055F  83 C2 02                 ADD     DX,2                ;POINT TO NEXT SECTION
0000:0562  ED                       IN      AX,DX               ;GET NEXT INFO
0000:0563  C0 C4 02                 ROL     AH,2                ;GET FINAL POSITION
0000:0566  80 E4 03                 AND     AH,03               ;PREPARE TO MERGE
0000:0569  0A DC                    OR      BL,AH               ;MERGE BITS
0000:056B  89 1E 02 20              MOV     POSD,BX             ;SAVE POSITION
0000:056F  A1 00 20                 MOV     AX,POSCEC
0000:0572  2B C3                    SUB     AX,BX               ;GET ERROR
0000:0574  C6 06 6D 20 00           MOV     DIR,0               ;ASSUME POSITIVE
0000:0579  73 05                    JNC     DST02A              ;IF POSITIVE
0000:057B  C6 06 6D 20 01           MOV     DIR,1               ;SET NEGATIVE
0000:0580  A3 04 20         DST02A: MOV     ERROR,AX            ;STORE IT HERE
0000:0583  C3               DSTOR3: RET
0000:0584                   BRKOFF:
0000:0584  E5 00                    IN      AX,P1PTA            ;GET CONTROL PORT
0000:0586  25 FF BF                 AND     AX,0BFFFH           ;ENGAGE THE BRAKE
0000:0589  E7 00                    OUT     P1PTA,AX            ;DO IT
0000:058B  C6 06 67 20 00           MOV     TFLAG1,0            ;SET TIMER FLAG = OK
0000:0590  E8 26 00                 CALL    SEC1                ;1 SEC
0000:0593  E8 A0 FF         BRKOF1: CALL    DSTOR               ;GET UPDATE
0000:0596  BA 87 01                 MOV     DX,DISCR            ;POINT TO BRAKE DISCRETES
0000:0599  EC                       IN      AL,DX               ;GET INFO
0000:059A  A8 10                    TEST    AL,16               ;BRAKE FREE?
0000:059C  74 12                    JZ      BRKOF2              ;IF FREE CONTINUE
0000:059E  F6 06 67 20 01           TEST    TFLAG1,1            ;TIMED OUT?
0000:05A3  74 EE                    JZ      BRKOF1              ;IF NOT
0000:05A5  81 0E 08 20 00 01        OR      WORD PTR FAUL1,256  ;SET BRAKE FAULT
0000:05AB  EA 00 00 00 00           JMP     FAR RSCMD           ;RESET COMMAND
0000:05B0                   BRKOF2:
0000:05B0  81 26 08 20 FF FE        AND     WORD PTR FAUL1,0FEFFH ;CLEAR BRAKE FAULT
0000:05B6  E9 0F 00                 JMP     TMROFF              ;STOP TIMER
0000:05B9                   SEC1:
0000:05B9  BA 5E FF                 MOV     DX,T1MODE           ;ENABLE TIMER1 FIRST
0000:05BC  B8 09 E0                 MOV     AX,0E009H
0000:05BF  EF                       OUT     DX,AX
0000:05C0  BA 66 FF                 MOV     DX,T2MODE           ;NOW TIMER2
0000:05C3  B8 01 C0                 MOV     AX,0C001H
0000:05C6  EF                       OUT     DX,AX
0000:05C7  C3                       RET
0000:05C8                   TMROFF:
0000:05C8  BE 12 06                 MOV     SI,OFFSET CS:TSETUP ;POINT TO TABLE
```

```
842  0000:05CB  BA 50 FF          MOV   DX,TOCOUT          ;POINT TO SOURCE POINTER
843  0000:05CE  B9 0C 00          MOV   CX,12
844  0000:05D1  6F          TSET: OUTSW
845  0000:05D2  83 C2 02          ADD   DX,2
846  0000:05D5  E2 FA             LOOP  TSET
847  0000:05D7  C3                RET
848  0000:05D8
849                           ;     TABLE SECTION FOR BOUND
850
851  0000:05D8                CLSETAB:
852  0000:05D8  FCFF               .WORD 0FFFCH             ;4 COUNTS BEFORE WE MOVE
853  0000:05DA  0400               .WORD 00004H
854
855  0000:05DC                CMDRG:
856  0000:05DC  1000               .WORD 16
857  0000:05DE  1400               .WORD 20
858
859  0000:05E0                CMDTBL:
860  0000:05E0  EA00               .WORD POSCMD
861  0000:05E2  2101               .WORD NAPCMD
862  0000:05E4  4201               .WORD MANCMD
863  0000:05E6  5C01               .WORD BDSCMD
864  0000:05E8  1A01               .WORD RECMD
865
866  0000:05EA                DELTA:
867  0000:05EA  F4FF               .WORD -12
868  0000:05EC  0C00               .WORD 12
869
870  0000:05EE                DSETUP:
871  0000:05EE  8B01               .WORD RELADD
872  0000:05F0  0000               .WORD 0
873  0000:05F2  4020               .WORD RAC
874  0000:05F4  0000               .WORD 0
875  0000:05F6  0400               .WORD 4
876  0000:05F8  66A7               .WORD DOVAL
877
878  0000:05FA                MONRG:
879  0000:05FA  1000               .WORD 16
880  0000:05FC  2F00               .WORD 47
881
882  0000:05FE                POSRG:
883  0000:05FE  008A               .WORD LOW                ;SOFTWARE LIMITS
884  0000:0600  0072               .WORD HIGH
885
886  0000:0602                RAMPLM:
887  0000:0602  00BA               .WORD 0BA00H             ;RAMP LIMITS
888  0000:0604  0046               .WORD 04600H
889
890  0000:0606                RAMPTL:
891  0000:0606  6803               .WORD ZIPUP              ;RAMP ENTRY
892  0000:0608  D203               .WORD MAINNY
893  0000:060A  0304               .WORD ZIPDWN
894  0000:060C  B102               .WORD VERGIT
895
896  0000:060E                SCREW:
897  0000:060E  CEFF               .WORD -50                ;SECOND SCREW LIMITS
898  0000:0610  3200               .WORD 50
```

```
899
900  0000:0612                  TSETUP:
901  0000:0612  0000                    .WORD   0
902  0000:0614  0000                    .WORD   0
903  0000:0616  0000                    .WORD   0
904  0000:0618  0040                    .WORD   16384           ;TIMER 0 IS NOT USED
905  0000:061A  0000            TSET1:   .WORD   0
906  0000:061C  D007                    .WORD   2000            ;1 SEC DELAY
907  0000:061E  0000                    .WORD   0
908  0000:0620  0040                    .WORD   16384           ;TIMER 1 IS READY TO GO
909  0000:0622  0000                    .WORD   0
910  0000:0624  B202                    .WORD   690             ;552uS TIMER
911  0000:0626  0000                    .WORD   0
912  0000:0628  0040                    .WORD   16384           ;TIMER 2 IS READY ALSO
913
914  0000:062A                          END
```

Lines Assembled : 914 Assembly Errors : 0

2500 A.D. 80186 Cross Assembler - Version 4.00g

Input Filename : FE.asm
Output Filename : FE.obj

```
1
2                               ;       FOCUS AXIS FOR VLBA ANTENNAS
3                               ;       WRITTEN BY:     WAYNE M. KOSKI
4                               ;       LAST REVISION:  DECEMBER 01, 1989
5                               ;
6                               ;       THINGS TO DO AND GENERAL NOTES:
7                               ;
8                               ;         1.  THIS SECTION OF CODE WILL BE THE MAIN RUNNING
9                               ;             ROUTINES AND SHALL EXCLUDE THE INTERRUPT ROUTINES
10                              ;             WHICH SHALL BE IN THE OTHER EPROM.
11                              ;         2.  ROTATION AXIS SHOULD LOOK SIMULAR TO FOCUS AT
12                              ;             PRESENT.
13                              ;         3.  THIS VERSION UPDATES TO THE NEW F/R CONTROLLER MODULE.
14
15                                      ASSUME  CS:CODE, DS:DSEG
16
17                                  .OUTPUT 2500AD
18                                  .OPTIONS H
19
20
21                              ;       80188 INTERNAL PORTS FOR CONTROL AND GUIDANCE
22
23
24
25                              ;       80188 INTERRUPT CONTROL / STATUS REGISTERS
26
27          0000:FF22           EOI:    .EQUAL  0FF22H          ;END OF INTERRUPT REGISTER
28          0000:FF24           POLL:   .EQUAL  0FF24H          ;INTERRUPT POLL REGISTER
29          0000:FF26           POLLS:  .EQUAL  0FF26H          ;INTERRUPT POLL STATUS REGISTER
30          0000:FF28           MASK:   .EQUAL  0FF28H          ;INTERRUPT MASK REGISTER
31          0000:FF2A           PMASK:  .EQUAL  0FF2AH          ;INTERRUPT PRIORITY MASK REGISTER
32          0000:FF2C           ISR:    .EQUAL  0FF2CH          ;INTERRUPT IN SERVICE REGISTER
33          0000:FF2E           IRR:    .EQUAL  0FF2EH          ;INTERRUPT REQUEST REGISTER
34          0000:FF30           ICSR:   .EQUAL  0FF30H          ;INTERRUPT CONTROL STATUS REGISTER
35          0000:FF32           ITCR:   .EQUAL  0FF32H          ;INTERRUPT TIMER CONTROL REGISTER
36          0000:FF34           ID0CR:  .EQUAL  0FF34H          ;INTERRUPT DMA 0 CONTROL REGISTER
37          0000:FF36           ID1CR:  .EQUAL  0FF36H          ;INTERRUPT DMA 1 CONTROL REGISTER
38          0000:FF38           INT0CR: .EQUAL  0FF38H          ;INT 0 CONTROL REGISTER
39          0000:FF3A           INT1CR: .EQUAL  0FF3AH          ;INT 1 CONTROL REGISTER
40          0000:FF3C           INT2CR: .EQUAL  0FF3CH          ;INT 2 CONTROL REGISTER
41          0000:FF3E           INT3CR: .EQUAL  0FF3EH          ;INT 3 CONTROL REGISTER
42
43
```

```
                        ;       80188 TIMER CONTROL REGISTERS

0000:FF50       T0COUT: .EQUAL  0FF50H          ;TIMER 0 COUNT REGISTER
0000:FF52       T0MAXA: .EQUAL  0FF52H          ;TIMER 0 MAXIMUM COUNT A REGISTER
0000:FF54       T0MAXB: .EQUAL  0FF54H          ;TIMER 0 MAXIMUM COUNT B REGISTER
0000:FF56       T0MODE: .EQUAL  0FF56H          ;TIMER 0 MODE REGISTER
0000:FF58       T1COUT: .EQUAL  0FF58H          ;TIMER 1 COUNT REGISTER
0000:FF5A       T1MAXA: .EQUAL  0FF5AH          ;TIMER 1 MAXIMUM COUNT A REGISTER
0000:FF5C       T1MAXB: .EQUAL  0FF5CH          ;TIMER 1 MAXIMUM COUNT B REGISTER
0000:FF5E       T1MODE: .EQUAL  0FF5EH          ;TIMER 1 MODE REGISTER
0000:FF60       T2COUT: .EQUAL  0FF60H          ;TIMER 2 COUNT REGISTER
0000:FF62       T2MAXA: .EQUAL  0FF62H          ;TIMER 2 MAXIMUM COUNT A REGISTER
0000:FF66       T2MODE: .EQUAL  0FF66H          ;TIMER 2 MODE REGISTER


                        ;       80188 CHIP SELECT CONTROL REGISTERS

0000:FFA0       UMCS:   .EQUAL  0FFA0H          ;80188 INTERNAL UPPER MEMORY CHIP
                                                ;SELECT CONTROL BLOCK REGISTER
0000:FFA2       LMCS:   .EQUAL  0FFA2H          ;80188 INTERNAL LOWER MEMORY CHIP
                                                ;SELECT CONTROL BLOCK REGISTER
0000:FFA4       PACS:   .EQUAL  0FFA4H          ;80188 INTERNAL PERIPHERAL CHIP
                                                ;SELECT CONTROL BLOCK REGISTER
0000:FFA6       MMCS:   .EQUAL  0FFA6H          ;80188 INTERNAL MIDDLE MEMORY
                                                ;START ADDRESS REGISTER
0000:FFA8       MPCS:   .EQUAL  0FFA8H          ;80188 INTERNAL MIDDLE MEMORY CHIP
                                                ;SELECT CONTROL BLOCK REGISTER


                        ;       80188 DMA CHANNEL CONTROL REGISTERS

0000:FFC0       D0SPL:  .EQUAL  0FFC0H          ;DMA 0 SOURCE POINTER LSB REGISTER
0000:FFC2       D0SPM:  .EQUAL  0FFC2H          ;DMA 0 SOURCE POINTER MSB REGISTER
0000:FFC4       D0DPL:  .EQUAL  0FFC4H          ;DMA 0 DESTINATION POINTER LSB REGISTER
0000:FFC6       D0DPM:  .EQUAL  0FFC6H          ;DMA 0 DESTINATION POINTER MSB REGISTER
0000:FFC8       D0TC:   .EQUAL  0FFC8H          ;DMA 0 TRANSFER COUNT REGISTER
0000:FFCA       D0MODE: .EQUAL  0FFCAH          ;DMA 0 MODE REGISTER
0000:FFD0       D1SPL:  .EQUAL  0FFD0H          ;DMA 1 SOURCE POINTER LSB REGISTER
0000:FFD2       D1SPM:  .EQUAL  0FFD2H          ;DMA 1 SOURCE POINTER MSB REGISTER
0000:FFD4       D1DPL:  .EQUAL  0FFD4H          ;DMA 1 DESTINATION POINTER LSB REGISTER
0000:FFD6       D1DPM:  .EQUAL  0FFD6H          ;DMA 1 DESTINATION POINTER MSB REGISTER
0000:FFD8       D1TC:   .EQUAL  0FFD8H          ;DMA 1 TRANSFER COUNT REGISTER
0000:FFDA       D1MODE: .EQUAL  0FFDAH          ;DMA 1 MODE REGISTER


                        ;       80188 INTERNAL I/O RELOCATION REGISTER

0000:FFFE       RELOC:  .EQUAL  0FFFEH          ;I/O RELOCATION REGISTER


                        ;       80188 INITIAL VALUES FOR INTERNAL REGISTERS

0000:007D       LMBS:   .EQUAL  007DH           ;LOWER MEMORY BLOCK SIZE = 2K
0000:81BD       MMBS:   .EQUAL  81BDH           ;MIDDLE MEMORY BLOCK SIZE = 8K
0000:03FD       MMST:   .EQUAL  03FDH           ;MIDDLE MEMORY START POSITION = 8K
0000:003D       PST:    .EQUAL  003DH           ;PERIPHERAL START ADDRESS = 0
0000:FFBD       UMBS:   .EQUAL  0FFBDH          ;UPPER MEMORY BLOCK SIZE = 2K
```

```
101
102
103
104                                     ;       EXTERNAL PORTS FOR CONTROLLING THE FOCUS AXIS
105
106             0000:0000               P1PTA:  .EQUAL  0               ;PROM 1 PORT A
107             0000:0001               P1PTB:  .EQUAL  1               ;PROM 1 PORT B
108             0000:0002               P1PTAD: .EQUAL  2               ;PROM 1 PORT A DIRECTION
109             0000:0003               P1PTBD: .EQUAL  3               ;PROM 1 PORT B DIRECTION
110             0000:0080               P2PTA:  .EQUAL  128             ;PROM 2 PORT A
111             0000:0081               P2PTB:  .EQUAL  129             ;PROM 2 PORT B
112             0000:0082               P2PTAD: .EQUAL  130             ;PROM 2 PORT A DIRECTION
113             0000:0083               P2PTBD: .EQUAL  131             ;PROM 2 PORT B DIRECTION
114             0000:0100               RAMTC:  .EQUAL  256             ;RAM TIMER AND CONTROL
115             0000:0101               RAMPTA: .EQUAL  257             ;RAM PORT A
116             0000:0102               RAMPTB: .EQUAL  258             ;RAM PORT B
117             0000:0103               RAMPTC: .EQUAL  259             ;RAM PORT C
118             0000:0104               RAMTLO: .EQUAL  260             ;RAM TIMER LOW
119             0000:0105               RAMTHI: .EQUAL  261             ;RAM TIMER HIGH
120             0000:0180               APEIRQ: .EQUAL  384             ;APEX REQ
121             0000:0182               APEIRS: .EQUAL  386             ;APEX RESPONSE
122             0000:0182               POSM:   .EQUAL  386             ;POSITION MSB
123             0000:0181               POSL:   .EQUAL  385             ;POSITION LSB
124             0000:0184               VELM:   .EQUAL  388             ;VELOCITY MSB
125             0000:0183               VELL:   .EQUAL  387             ;VELOCITY LSB
126             0000:0186               ANAH:   .EQUAL  390             ;ANALOGS MSB
127             0000:0185               ANAL:   .EQUAL  389             ;ANALOGS LSB
128             0000:0187               DISCR:  .EQUAL  391             ;FOC DISCRETES
129             0000:018B               RELADD: .EQUAL  395             ;GET RELATIVE ADDRESS
130             0000:018C               COMML:  .EQUAL  396             ;CONTROL VALUE LSB
131             0000:018D               COMMM:  .EQUAL  397             ;CONTROL VALUE MSB
132             0000:018E               DEVACK: .EQUAL  398             ;DEVICE ACKNOWLEDGE
133             0000:018F               MONL:   .EQUAL  399             ;MONITOR DATA MSB
134             0000:0190               MONM:   .EQUAL  400             ;MONITOR DATA LSB
135             0000:0191               ADL:    .EQUAL  401             ;READ A/D LSB
136             0000:0192               ADH:    .EQUAL  402             ;READ A/D MSB
137             0000:0193               SELCT:  .EQUAL  403             ;SELECT MOTOR CURRENT/TORQUE
138             0000:0194               STCNV:  .EQUAL  404             ;START A/D CONVERT
139             0000:0195               MODESW: .EQUAL  405             ;READ MODE SWITCH
140             0000:0196               LTCHDA: .EQUAL  406             ;LATCH DRIVE D/A
141             0000:0197               BDERL1: .EQUAL  407             ;BDS3 ERROR LSB #1
142
143
144
145   0000:0000                         DSEG:   SEGMENT
146
147                                     ;       DATA STORAGE
148
149   0000:2000                                 ORG     2000H
150
151             0000:2000               MEMST:  .EQUAL  $
152
153                                     ;       MONITOR STORAGE
154
155   0000:2000                         POSCEC: .BLKB   2               ;POSITION COMMAND ECHO
156   0000:2002                         POSD:   .BLKB   2               ;CURRENT POSITION DATA
157   0000:2004                         ERROR:  .BLKB   2               ;POSCEC-POSD
```

```
158  0000:2006                SYSTEM: .BLKB  2          ;SYSTEM PARAMETERS
159  0000:2008                FAUL1:  .BLKB  2          ;FAULT BITS SET 1
160  0000:200A                FAUL2:  .BLKB  2          ;FAULT BITS SET 2
161  0000:200C                ANAFL:  .BLKB  2          ;ANALOG FAULT FLAGS
162
163            0000:200E      ANADT:  .EQUAL $          ;ANALOG STORAGE IS HERE
164
165  0000:200E                GND1:   .BLKB  2          ;GND
166  0000:2010                GND2:   .BLKB  2          ;GND
167  0000:2012                GND3:   .BLKB  2          ;GND
168  0000:2014                VEL:    .BLKB  2          ;FOCUS VELOCITY
169  0000:2016                V15P:   .BLKB  2          ;+15V/2
170  0000:2018                V15N:   .BLKB  2          ;-15V/2
171  0000:201A                V5:     .BLKB  2          ;+5V
172  0000:201C                V10P:   .BLKB  2          ;+10V
173  0000:201E                MTEMP1: .BLKB  2          ;MOUNT TEMP 1
174  0000:2020                V10N:   .BLKB  2          ;-10V
175  0000:2022                MTEMP2: .BLKB  2          ;MOUNT TEMP 2
176  0000:2024                BTEMP:  .BLKB  2          ;BIN TEMP
177
178  0000:2026                SERVER: .BLKB  2
179  0000:2028                VEL1:   .BLKB  2
180  0000:202A                X22:    .BLKB  2
181  0000:202C                X23:    .BLKB  2
182  0000:202E                X24:    .BLKB  2
183  0000:2030                X25:    .BLKB  2
184  0000:2032                X26:    .BLKB  2
185  0000:2034                X27:    .BLKB  2
186  0000:2036                X28:    .BLKB  2
187  0000:2038                X29:    .BLKB  2
188  0000:203A                X30:    .BLKB  2
189  0000:203C                X31:    .BLKB  2
190  0000:203E                X32:    .BLKB  2
191
192
193                           ;       TEMPORARY STORAGE
194
195  0000:2040                RAC:    .BLKB  1          ;RELATIVE CONTROL ADDRESS
196  0000:2041                CONL:   .BLKB  1          ;CONTROL VALUE LSB
197  0000:2042                CONM:   .BLKB  1          ;CONTROL VALUE MSB
198  0000:2043                ACKF:   .BLKB  1          ;ACKNOWLEDGE FLAG
199  0000:2044                COMTMP: .BLKB  2          ;TEMPORARY COMMAND STORAGE
200  0000:2046                ADVAL:  .BLKB  2          ;A/D TEMPORARY STORAGE
201  0000:2048                SPEED:  .BLKB  1          ;RAMP LEVEL
202  0000:2049                BREAK1: .BLKB  2          ;RAMP UP BREAK POINT
203  0000:204B                BREAK2: .BLKB  2          ;RAMP DOWN BREAK POINT
204  0000:204D                POSOLD: .BLKB  2          ;OLD POSITION
205  0000:204F                SCRLST: .BLKB  2          ;LAST 500US POSITION
206  0000:2051                SCRCNT: .BLKB  2          ;SCREW COUNT
207  0000:2053                EXTTMR: .BLKB  1          ;EXTERNAL TROUBLE TIMER
208  0000:2054                DEL:    .BLKB  2          ;FILTER DELTA
209  0000:2056                FIL:    .BLKB  1          ;FILTER FAIL FLAG
210  0000:2057                FILOVR: .BLKB  1          ;MAX FILTER FAILURE COUNT
211  0000:2058                PHASEA: .BLKB  1          ;CURRENT SENSOR LEVEL
212  0000:2059                SCWIGN: .BLKB  1
213  0000:205A                RANDOM: .BLKB  2          ;RANDOM NUMBER GENERATOR
214  0000:205C                SECCHC: .BLKB  1          ;SECOND CHANCE FOR 2ND SCREW
```

```
215
216                                     ;       FLAGS
217
218                     0000:205D       FLAGST: .EQUAL  $
219
220     0000:205D                       RSSCMD: .BLKB   1
221     0000:205E                       NAPATV: .BLKB   1
222     0000:205F                       NAPREQ: .BLKB   1
223     0000:2060                       DRVREQ: .BLKB   1
224     0000:2061                       MANOVR: .BLKB   1
225     0000:2062                       BDSRST: .BLKB   1
226     0000:2063                       DRVONE: .BLKB   1
227     0000:2064                       ADFLAG: .BLKB   1
228     0000:2065                       BFLAG:  .BLKB   1
229     0000:2066                       TFLAG0: .BLKB   1
230     0000:2067                       TFLAG1: .BLKB   1
231     0000:2068                       TFLAG2: .BLKB   1
232     0000:2069                       DRVOFF: .BLKB   1
233     0000:206A                       DRVATV: .BLKB   1
234     0000:206B                       RAMPOS: .BLKB   2
235     0000:206D                       DIR:    .BLKB   1
236     0000:206E                       LDIR:   .BLKB   1
237
238                     0000:0012       ENDFLG: .EQUAL  $-FLAGST
239
240
241                                             ENDS
242
243
244                                     ;       PROGRAM EQUATES
245
246                     0000:2100       STCK:   .EQUAL  2100H                   ;STACK LOCATION
247                     0000:A766       DOVAL:  .EQUAL  0A766H                  ;DMA 0 CONTROL VALUE
248                     0000:00E8       MASKV:  .EQUAL  00E8H                   ;DMA 0, INT0, TIMER ENABLE
249                     0000:0A00       LOW:    .EQUAL  00A00H                  ;LOW VALUE
250                     0000:F200       HIGH:   .EQUAL  0F200H                  ;HIGH VALUE
251                     0000:001C       CONRAM: .EQUAL  28                      ;STARTING RAMP VALUE
252
253                                     ;       GLOBAL LIST
254
255                                             GLOBAL  INITAL, RSCMD
256                                             EXTERNAL DSETUP:WORD, TSETUP:WORD, TSET1:WORD
257
258                                     ;       PROGRAM ENTERS HERE AFTER RESET ROUTINE (AS IF BY MAGIC)
259
260     FF80:0000                               ORG     0FF80:0000H
261
262
263     FF80:0000                       INITAL:
264     FF80:0000   BC 00 21                    MOV     SP,STCK                 ;SET STACK POINTER
265
266                                     ;       LETS INITIALIZE MORE 80188 CHIP SELECT LOGIC
267
268     FF80:0003   BA A2 FF                    MOV     DX,LMCS                 ;SET UP LOWER MEMORY BLOCK
269     FF80:0006   B8 7D 00                    MOV     AX,LMBS                 ;TO 2K
270     FF80:0009   EF                          OUT     DX,AX
271     FF80:000A   BA A8 FF                    MOV     DX,MPCS                 ;NOW SET MIDDLE MEMORY BLOCK TO
```

```
272  FF80:000D  B8 BD 81          MOV   AX,MMBS          ;2K AND I/O MAPPED PCS LINES
273  FF80:0010  EF                OUT   DX,AX
274  FF80:0011  BA A6 FF          MOV   DX,MMCS          ;SET MIDDLE MEMORY TO START AT
275  FF80:0014  B8 FD 03          MOV   AX,MMST          ;8K
276  FF80:0017  EF                OUT   DX,AX
277  FF80:0018  BA A4 FF          MOV   DX,PACS          ;SET UP THE PERIPHERAL ADDRESS
278  FF80:001B  B8 3D 00          MOV   AX,PST           ;TO START AT 0
279  FF80:001E  EF                OUT   DX,AX
280
281                          ;     LETS INITIALIZE PORTS AND PORT VALUES
282
283  FF80:001F  B8 FF FF          MOV   AX,0FFFFH        ;EPROM 1 PORT A, B = OUTPUT
284  FF80:0022  E7 02             OUT   P1PTAD,AX
285  FF80:0024  B8 FF 07          MOV   AX,07FFH         ;UPPER NYBBLE = INPUT
286  FF80:0027  E7 82             OUT   P2PTAD,AX
287  FF80:0029  BA 00 01          MOV   DX,RAMIC         ;NOW DO RAM'S I/O
288  FF80:002C  EE                OUT   DX,AL            ;SET TO OUTPUT
289  FF80:002D  33 C0             XOR   AX,AX            ;CLEAR ACC
290  FF80:002F  E7 00             OUT   P1PTA,AX         ;CLEAR EPROM 1 PORTS A AND B
291  FF80:0031  E7 80             OUT   P2PTA,AX         ;CLEAR EPROM 2 PORTS A AND B
292  FF80:0033  42                INC   DX               ;POINT TO RAM PORT A
293  FF80:0034  EF                OUT   DX,AX            ;CLEAR RAM PORTS A AND B
294  FF80:0035  42                INC   DX
295  FF80:0036  EF                OUT   DX,AX            ;CLEAR PORT C TOO
296  FF80:0037  BA 93 01          MOV   DX,SEMCT         ;CLEAR LIGHTS
297  FF80:003A  EE                OUT   DX,AL
298
299                          ;     NOW LETS SET UP THE DMA 0 CHANNEL
300
301  FF80:003B  BE 00 00          MOV   SI,OFFSET DSETUP ;POINT TO TABLE
302  FF80:003E  BA C0 FF          MOV   DX,DOSPL         ;SET UP SOURCE POINTER
303  FF80:0041  B9 06 00          MOV   CX,6
304  FF80:0044  6F            DSET:OUTSW
305  FF80:0045  83 C2 02          ADD   DX,2
306  FF80:0048  E2 FA             LOOP  DSET
307
308                          ;     NOW LETS INITIALIZE THE TIMERS
309
310  FF80:004A  E8 47 06          CALL  TMROFF
311
312                          ;     NOW LETS ZERO MEMORY
313
314  FF80:004D  32 C0             XOR   AL,AL
315  FF80:004F  B9 00 01          MOV   CX,256           ;256 LOCATIONS
316  FF80:0052  BF 00 20          MOV   DI,OFFSET MEMST  ;START OF INDEX
317  FF80:0055  F3 AA             REP STOS BYTE PTR MEMST
318
319
320                          ;     NOW LETS TURN ON THE INTERRUPTS
321
322  FF80:0057  BA 28 FF          MOV   DX,MASK          ;LETS TURN ON DMA, INTO, TIMER
323  FF80:005A  B8 E8 00          MOV   AX,MASKV         ;INTERRUPTS
324  FF80:005D  EF                OUT   DX,AX
325  FF80:005E  BA 38 FF          MOV   DX,INTOCR        ;SET INTO PRIORITY
326  FF80:0061  B8 00 00          MOV   AX,0             ;HIGHEST PRIORITY
327  FF80:0064  EF                OUT   DX,AX
328  FF80:0065  FB                STI
```

```
329
330                                 ;       LETS RESET THE BDS3 SERVO AMPLIFIERS
331
332  FF80:0066  E8 9A 05                    CALL    BDS3RS                  ;INITIATE RESET
333
334                                 ;       LETS JUST WAIT FOR THREE SECONDS FOR THINGS TO SETTLE
335
336  FF80:0069  B9 03 00                    MOV     CX,3                    ;THREE SECONDS TOTAL
337  FF80:006C  E8 04 06                    CALL    DELAY
338
339                                 ;       NOW LETS WAIT FOR THE THREE PHASE
340
341  FF80:006F                      PHASE:
342  FF80:006F  B8 00 80                    MOV     AX,8000H                ;TURN ON 3 PHASE
343  FF80:0072  E7 00                        OUT     P1PTA,AX
344  FF80:0074  B9 03 00                    MOV     CX,3                    ;THREE SECONDS TOTAL
345  FF80:0077  C6 06 67 20 00      PH301:   MOV     TFLAG1,0                ;SET TIMER FLAG = OK
346  FF80:007C  BA 97 01                    MOV     DX,BDERL1               ;GET BDS3 INFO
347  FF80:007F  EC                  PH302:   IN      AL,DX
348  FF80:0080  25 01 00                    AND     AX,01H                  ;TEST FOR BUS FAULT
349  FF80:0083  75 11                        JNZ     THREX                   ;EXIT IF OK
350  FF80:0085  F6 06 67 20 01               TEST    TFLAG1,1                ;TIME OUT?
351  FF80:008A  74 F3                        JZ      PH302                   ;LOOP UNTIL TIME OUT
352  FF80:008C  E2 E9                        LOOP    PH301                   ;UNTIL THREE SEC HAVE PASSED
353  FF80:008E  80 0E 0A 20 01               OR      FAUL2,01H               ;SET BDS3 DEAD FLAG
354  FF80:0093  E9 D9 FF                     JMP     PHASE                   ;IF BUS FAULT
355
356  FF80:0096                      THREX:
357  FF80:0096  E8 FB 05                    CALL    TMROFF                  ;STOP TIMERS
358  FF80:0099  C6 06 0A 20 00               MOV     FAUL2,0                 ;RESET BDS3 DEAD FLAG
359  FF80:009E
360
361                                 ;       SERIAL NUMBER / SOFTWARE REVISION
362
363  FF80:009E  C7 06 26 20 03 06            MOV     WORD PTR SERVER,0603H
364
365                                 ;       LETS DO ONE DSTOR PRIOR TO ENTRY
366
367  FF80:00A4  E8 2F 04                     CALL    DSTOR
368
369
370                                 ;       HERE WE ARE INTO THE MAIN PROGRAM
371
372
373  FF80:00A7                      BOSS:
374  FF80:00A7  8B 0E 5A 20                  MOV     CX,RANDOM               ;LETS DELAY A BIT
375  FF80:00AB  E2 FE               WAIT:    LOOP    WAIT                    ;TO KILL APEX SYNCING
376  FF80:00AD  BC 00 21                     MOV     SP,STCK                 ;LETS SET UP STACK POINTER
377  FF80:00B0  F6 06 08 20 01               TEST    FAUL1,1                 ;APEX OK?
378  FF80:00B5  74 03                        JZ      NEXT                    ;IF OK
379  FF80:00B7  E9 A4 00                     JMP     RSCMD                   ;IF APEX BROKE
380  FF80:00BA                      NEXT:
381  FF80:00BA  F6 06 6A 20 01               TEST    DRVATV,1                ;DRIVE ACTIVE?
382  FF80:00BF  74 06                        JZ      NEXTO                   ;IF NOT
383  FF80:00C1  E8 CC 00                     CALL    CHKDRV                  ;IF ACTIVE
384  FF80:00C4  E9 E0 FF                     JMP     BOSS
385  FF80:00C7                      NEXTO:
```

```
386  FF80:00C7  BA 95 01              MOV   DX,MODESW           ;GET MODE SWITCH
387  FF80:00CA  EC                    IN    AL,DX               ;GET INFO
388  FF80:00CB  F6 D0                 NOT   AL                  ;INVERT AL
389  FF80:00CD  3C FF                 CMP   AL,0FFH             ;A = 0FFH?
390  FF80:00CF  C6 06 59 20 00        MOV   SCWIGN,0            ;ASSUME ON
391  FF80:00D4  75 05                 JNZ   BOSS1               ;IF ON
392  FF80:00D6  C6 06 59 20 01        MOV   SCWIGN,1            ;ELSE TURN OFF
393  FF80:00DB  E8 F8 03        BOSS1: CALL  DSTOR               ;LETS GET APEX DATA
394  FF80:00DE  F6 06 61 20 01        TEST  MANOVR,1            ;MANUAL OVERRIDE?
395  FF80:00E3  74 07                 JZ    NEXT1A              ;IF OVERRIDE
396  FF80:00E5  80 0E 06 20 08        OR    SYSTEM,8            ;MANUAL OVER RIDE FLAG
397  FF80:00EA  EB 09                 JMP   SHORT NEXT1
398  FF80:00EC  E4 81           NEXT1A: IN    AL,P2PTB            ;IN LOCAL?
399  FF80:00EE  24 20                 AND   AL,20H
400  FF80:00F0  75 03                 JNZ   NEXT1               ;IF NOT
401  FF80:00F2  E9 75 02              JMP   LOCAL               ;IF LOCAL
402  FF80:00F5                  NEXT1:
403  FF80:00F5  F6 06 62 20 01        TEST  BDSRST,1            ;RESET BDS3 SERVO AMP
404  FF80:00FA  74 14                 JZ    NEXT2               ;IF NOT
405  FF80:00FC  C6 06 62 20 00        MOV   BDSRST,0            ;RESET REQUEST
406  FF80:0101  E8 FF 04              CALL  BDS3RS              ;DO IT
407  FF80:0104  B9 01 00              MOV   CX,1                ;WAIT 1 SEC
408  FF80:0107  E8 69 05              CALL  DELAY
409  FF80:010A  E8 87 05              CALL  TMROFF              ;STOP TIMERS
410  FF80:010D  E9 97 FF              JMP   BOSS
411  FF80:0110                  NEXT2:
412  FF80:0110  F6 06 5D 20 01        TEST  RESCMD,1            ;SOFT RESET?
413  FF80:0115  74 03                 JZ    NEXT3               ;IF NOT
414  FF80:0117  E9 38 00              JMP   CMDRS               ;IF RESET
415  FF80:011A                  NEXT3:
416  FF80:011A  F6 06 5E 20 01        TEST  NAPATV,1            ;IN NAP?
417  FF80:011F  74 08                 JZ    NEXT4               ;IF NOT
418  FF80:0121  C6 06 60 20 00        MOV   DRVREQ,0            ;IF NAP, KILL ALL REQ
419  FF80:0126  E9 7E FF              JMP   BOSS                ;IF NAP, GOTO BOSS
420  FF80:0129                  NEXT4:
421  FF80:0129  F6 06 60 20 01        TEST  DRVREQ,1            ;DRIVE REQUEST?
422  FF80:012E  74 06                 JZ    NEXT5               ;IF NO DRIVE REQUEST
423  FF80:0130  E8 B0 00              CALL  DRVINT              ;IF DRIVE REQUEST
424  FF80:0133  E9 71 FF              JMP   BOSS
425  FF80:0136                  NEXT5:
426  FF80:0136  F6 06 5F 20 01        TEST  NAPREQ,1            ;NAP REQUEST?
427  FF80:013B  75 03                 JNZ   NEXT6
428  FF80:013D  E9 67 FF              JMP   BOSS                ;IF NO NAP REQUEST
429  FF80:0140                  NEXT6:
430  FF80:0140  C6 06 5F 20 00        MOV   NAPREQ,0            ;REMOVE NAP REQUEST
431  FF80:0145  C6 06 5E 20 FF        MOV   NAPATV,0FFH         ;SET NAP ACTIVE
432  FF80:014A  80 0E 06 20 01        OR    SYSTEM,1            ;SET NAP FLAG
433  FF80:014F  E9 55 FF              JMP   BOSS                ;END OF MAIN ROUTINE
434
435  FF80:0152                  CMDRS:
436  FF80:0152  81 26 08 20 1F FC     AND   WORD PTR FAUL1,0FC1FH  ;CLEAR CERTAIN FAULTS
437  FF80:0158  81 26 06 20 FE FF     AND   WORD PTR SYSTEM,0FFFEH ;CLEAR  NAP
438  FF80:015E                  RSCMD:
439  FF80:015E  81 26 06 20 F5 FF     AND   WORD PTR SYSTEM,0FFF5H ;CLEAR LOCAL, OVER RIDE
440  FF80:0164  BC 00 21              MOV   SP,STCK             ;RESET STACK
441  FF80:0167  C6 06 69 20 01        MOV   DRVOFF,1            ;TURN OFF DRIVE
442  FF80:016C  F6 06 6A 20 01        TEST  DRVATV,1            ;DRIVING?
```

```
443  FF80:0171  74 03              JZ      RSCMD1              ;IF NOT DRIVING
444  FF80:0173  E9 31 FF           JMP     BOSS                ;IF STILL DRIVING
445  FF80:0176  32 C0      RSCMD1: XOR     AL,AL               ;NOW ZERO FLAGS
446  FF80:0178  B9 12 00           MOV     CX,ENDFLG           ;NUMBER OF FLAG LOCATIONS
447  FF80:017B  BF 5D 20           MOV     DI,OFFSET FLAGST    ;START OF FLAGS
448  FF80:017E  F3 AA              REP STOS BYTE PTR FLAGST
449  FF80:0180  E8 C7 04           CALL    BRKOFF              ;ENGAGE BRAKE
450  FF80:0183  C6 06 5D 20 00     MOV     RESCMD,0            ;KILL RESET REQUEST
451  FF80:0188  B8 00 00           MOV     AX,0                ;KILL LIGHTS
452  FF80:018B  E7 80              OUT     P2PTA,AX
453  FF80:018D  E9 17 FF           JMP     BOSS
454
455  FF80:0190              CHKDRV:
456  FF80:0190  FE 0E 53 20        DEC     BYTE PTR EXTTMR     ;DEC SAFETY TIMER
457  FF80:0194  75 01              JNZ     CHKDR0              ;IF OK
458  FF80:0196  CC                 INT     3                   ;DO TIMER THIS WAY
459  FF80:0197  F6 06 60 20 01 CHKDR0: TEST DRVREQ,1           ;NEW REQUEST
460  FF80:019C  75 01              JNZ     CHKDR1              ;IF REQUEST
461  FF80:019E  C3                 RET                         ;IF NOT
462  FF80:019F  C6 06 60 20 00 CHKDR1: MOV DRVREQ,0            ;KILL REQUEST
463  FF80:01A4  8B 1E 00 20        MOV     BX,POSCEC           ;BUT TEST FOR CLOSE
464  FF80:01A8  2B 1E 44 20        SUB     BX,COMTMP
465  FF80:01AC  C6 06 65 20 00     MOV     BFLAG,0             ;ASSUME IN BOUND
466  FF80:01B1  2E 62 1E D4 06     BOUND   BX,CS:CLSETAB       ;CLOSE?
467  FF80:01B6  F6 06 65 20 01     TEST    BFLAG,1
468  FF80:01BB  75 07              JNZ     CHKDR2              ;IF NOT CLOSE
469  FF80:01BD  C7 06 44 20 00 00  MOV     WORD PTR COMTMP,0   ;FOR DRIVE OK
470  FF80:01C3  C3                 RET                         ;IF CLOSE
471  FF80:01C4  C6 06 69 20 01 CHKDR2: MOV DRVOFF,1            ;TURN OFF DRIVE
472  FF80:01C9  C6 06 60 20 01     MOV     DRVREQ,1            ;RETAIN REQUEST
473  FF80:01CE  F6 06 6A 20 01     TEST    DRVATV,1            ;DRIVING?
474  FF80:01D3  74 01              JZ      CHKDR3              ;IF STOPPED
475  FF80:01D5  C3                 RET                         ;IF STILL DRIVING
476  FF80:01D6  32 C0      CHKDR3: XOR     AL,AL               ;NOW ZERO FLAGS
477  FF80:01D8  B9 12 00           MOV     CX,ENDFLG           ;NUMBER OF FLAG LOCATIONS
478  FF80:01DB  BF 5D 20           MOV     DI,OFFSET FLAGST    ;START OF FLAGS
479  FF80:01DE  F3 AA              REP STOS BYTE PTR FLAGST
480  FF80:01E0  E8 67 04           CALL    BRKOFF              ;ENGAGE BRAKE
481
482  FF80:01E3              DRVINT:
483  FF80:01E3  C6 06 60 20 00     MOV     DRVREQ,0            ;KILL REQUEST
484  FF80:01E8  80 3E 5C 20 02     CMP     BYTE PTR SECCHC,2   ;SECOND SCREW FAILED TWICE?
485  FF80:01ED  74 08              JE      DRVSTP              ;IF EQUAL
486  FF80:01EF  F7 06 08 20 00 18  TEST    WORD PTR FAUL1,1800H ;E-STOP, DRIVE LOCKOUT?
487  FF80:01F5  74 01              JZ      DRVCNT              ;IF OK
488  FF80:01F7  C3         DRVSTP: RET                         ;ELSE DON'T DRIVE
489  FF80:01F8  32 C0      DRVCNT: XOR     AL,AL               ;CLEAR ALL DRIVE FLAGS
490  FF80:01FA  B9 05 00           MOV     CX,OFFSET FLAGST+ENDFLG-DRVATV
491  FF80:01FD  BF 6A 20           MOV     DI,OFFSET DRVATV
492  FF80:0200  F3 AA              REP STOS BYTE PTR DRVATV
493  FF80:0202  C7 06 08 20 00 00  MOV     WORD PTR FAUL1,0    ;CLEAR FAULTS
494  FF80:0208  C7 06 0A 20 00 00  MOV     WORD PTR FAUL2,0
495  FF80:020E  8B 1E 44 20        MOV     BX,COMTMP           ;MAKE TEMP COMMAND ACTUAL COMMAND
496  FF80:0212  89 1E 00 20        MOV     POSCEC,BX
497  FF80:0216  C7 06 44 20 00 00  MOV     WORD PTR COMTMP,0
498  FF80:021C  E8 B7 02           CALL    DSTOR
499
```

```
500                                     ;LOCAL SETUP ENTERS HERE
501
502  FF80:021F                  LOCTST:
503  FF80:021F  80 3E 5C 20 02          CMP     BYTE PTR SECCHC,2       ;SECOND SCREW FAILED TWICE?
504  FF80:0224  74 08                   JE      LOCSTP                  ;IF EQUAL
505  FF80:0226  F7 06 08 20 00 18       TEST    WORD PTR FAUL1,1800H    ;E-STOP, DRIVE LOCKOUT?
506  FF80:022C  74 01                   JZ      LOCCNT                  ;IF OK
507  FF80:022E  C3              LOCSTP: RET                             ;ELSE DON'T DRIVE
508  FF80:022F  80 3E 0A 20 48  LOCCNT: CMP     FAUL2,48H               ;NORMAL BDS3?
509  FF80:0234  74 01                   JZ      DRVIN0                  ;IF OK
510  FF80:0236  C3                      RET                             ;ELSE DO NOTHING
511  FF80:0237  8B 1E 04 20     DRVIN0: MOV     BX,ERROR                ;GET ERROR
512  FF80:023B  C6 06 65 20 00          MOV     BFLAG,0
513  FF80:0240  2E 62 1E D4 06          BOUND   BX,CS:CLSETAB           ;CHECK FOR NEARNESS
514  FF80:0245  F6 06 65 20 01          TEST    BFLAG,1
515  FF80:024A  75 01                   JNZ     DRVIN1                  ;IF NOT NEAR
516  FF80:024C  C3                      RET                             ;IF NEAR
517  FF80:024D  A0 08 20        DRVIN1: MOV     AL,FAUL1                ;GET LIMIT INFO
518  FF80:0250  F6 06 6D 20 01          TEST    DIR,1                   ;TEST DIRECTION
519  FF80:0255  74 05                   JZ      DRVIN2                  ;IF POSITIVE
520  FF80:0257  24 06                   AND     AL,6                    ;DOWN LIMIT?
521  FF80:0259  74 06                   JZ      DRVIN3
522  FF80:025B  C3                      RET                             ;IF IN DOWN LIMIT
523  FF80:025C  24 18           DRVIN2: AND     AL,24                   ;UP LIMIT?
524  FF80:025E  74 01                   JZ      DRVIN3
525  FF80:0260  C3                      RET                             ;IF IN UP LIMIT
526  FF80:0261  A0 6D 20        DRVIN3: MOV     AL,DIR
527  FF80:0264  A2 6E 20                MOV     LDIR,AL
528  FF80:0267  BA 00 00                MOV     DX,0                    ;SET UP FOR DIVIDE
529  FF80:026A  A1 04 20                MOV     AX,ERROR                ;GET ERROR
530  FF80:026D  F6 06 6E 20 01          TEST    LDIR,1                  ;TEST FOR NEGATIVE
531  FF80:0272  74 02                   JZ      DRVIN4                  ;IF POSITIVE
532  FF80:0274  F7 D8                   NEG     AX                      ;MAKE ABSOLUTE VALUE
533  FF80:0276  50              DRVIN4: PUSH    AX                      ;SAVE ERROR
534  FF80:0277  B9 64 00                MOV     CX,100                  ;DIVIDE BY 100
535  FF80:027A  F7 F1                   DIV     CX
536  FF80:027C  52                      PUSH    DX
537  FF80:027D  50                      PUSH    AX
538  FF80:027E  52                      PUSH    DX
539  FF80:027F  B9 07 00                MOV     CX,7                    ;FIRST BREAK VALUE
540  FF80:0282  F7 E1                   MUL     CX
541  FF80:0284  8B D8                   MOV     BX,AX                   ;STORE HERE
542  FF80:0286  58                      POP     AX                      ;NOW REMAINDER
543  FF80:0287  F7 E1                   MUL     CX
544  FF80:0289  B9 64 00                MOV     CX,100                  ;ADJUST
545  FF80:028C  F7 F1                   DIV     CX
546  FF80:028E  03 D8                   ADD     BX,AX
547  FF80:0290  75 03                   JNZ     DRVIN5                  ;ZERO?
548  FF80:0292  BB 01 00                MOV     BX,1                    ;MAKE ONE IF ZERO
549  FF80:0295  F6 06 06 20 02  DRVIN5: TEST    SYSTEM,2                ;IN MANUAL?
550  FF80:029A  74 08                   JZ      DRVIN6                  ;IF NOT
551  FF80:029C  83 FB 3C                CMP     BX,60                   ;>60?
552  FF80:029F  72 03                   JC      DRVIN6                  ;IF LESS
553  FF80:02A1  BB 3C 00                MOV     BX,60                   ;SET MAX RAMP
554  FF80:02A4  A1 02 20        DRVIN6: MOV     AX,POSD
555  FF80:02A7  A3 4D 20                MOV     POSDOD,AX               ;OLD POS = CURRENT
556  FF80:02AA  A3 4F 20                MOV     SCRLST,AX               ;500US POSITION
```

```
FF80:02AD C7 06 51 20 00 00          MOV   WORD PTR SCRCNT,0     ;SCREW COUNT = 0
FF80:02B3 C6 06 57 20 00             MOV   FILOVR,0
FF80:02B8 C6 06 56 20 00             MOV   FIL,0
FF80:02BD F6 06 6E 20 01             TEST  LDIR,1                ;CHECK DIRECTION
FF80:02C2 74 02                      JZ    DRVIN7                ;IF POSITIVE
FF80:02C4 F7 DB                      NEG   BX
FF80:02C6 03 C3              DRVIN7: ADD   AX,BX                 ;STORE FIRST BREAK POINT
FF80:02C8 A3 49 20                   MOV   BREAK1,AX
FF80:02CB 58                         POP   AX
FF80:02CC B9 5A 00                   MOV   CX,90                 ;NOW SECOND BREAK VALUE
FF80:02CF F7 E1                      MUL   CX
FF80:02D1 8B D8                      MOV   BX,AX                 ;STORE HERE
FF80:02D3 58                         POP   AX                    ;NOW REMAINDER
FF80:02D4 F7 E1                      MUL   CX
FF80:02D6 B9 64 00                   MOV   CX,100                ;ADJUST
FF80:02D9 F7 F1                      DIV   CX
FF80:02DB 03 D8                      ADD   BX,AX
FF80:02DD 58                         POP   AX                    ;RESTORE ERROR
FF80:02DE F6 06 06 20 02             TEST  SYSTEM,2              ;IN MANUAL?
FF80:02E3 74 0C                      JZ    DRVIN8                ;IF NOT
FF80:02E5 2B C3                      SUB   AX,BX                 ;FIND DIFFERENCE
FF80:02E7 3D 41 00                   CMP   AX,65                 ;>65?
FF80:02EA 72 05                      JC    DRVIN8                ;IF LESS
FF80:02EC 03 D8                      ADD   BX,AX                 ;GET ERROR BACK IN BX
FF80:02EE 83 EB 41                   SUB   BX,65                 ;MAKE BREAK 65 COUNTS LESS
FF80:02F1 A1 02 20           DRVIN8: MOV   AX,POSD
FF80:02F4 F6 06 6E 20 01             TEST  LDIR,1                ;CHECK DIRECTION
FF80:02F9 74 02                      JZ    DRVIN9                ;IF POSITIVE
FF80:02FB F7 DB                      NEG   BX
FF80:02FD 03 C3              DRVIN9: ADD   AX,BX                 ;SECOND BREAK POINT
FF80:02FF A3 4B 20                   MOV   BREAK2,AX
FF80:0302 C7 06 28 20 00 00          MOV   WORD PTR VEL1,0       ;INITIAL VEL = 0
FF80:0308 E8 0D 03                   CALL  BRON                  ;TURN ON BRAKE
FF80:030B E5 00                      IN    AX,P1PTA              ;GET CURRENT DATA
FF80:030D 25 00 F0                   AND   AX,0F000H             ;JUST CURRENT STATUS
FF80:0310 BB 1C 00                   MOV   BX,CONRAM             ;START DRIVE AT 130mV
FF80:0313 F6 06 6E 20 01             TEST  LDIR,1                ;NEGATIVE?
FF80:0318 75 02                      JNZ   DRVINA                ;IF POSITIVE
FF80:031A F7 DB                      NEG   BX                    ;SET NEGATIVE
FF80:031C 81 E3 FF 0F        DRVINA: AND   BX,0FFFH              ;STRIP UNWANTED BITS
FF80:0320 0B C3                      OR    AX,BX                 ;MERGE
FF80:0322 E7 00                      OUT   P1PTA,AX              ;SEND IT
FF80:0324 BA 96 01                   MOV   DX,LTCHDA             ;LATCH IT
FF80:0327 EE                         OUT   DX,AL
FF80:0328 80 CC 10                   OR    AH,10H                ;ENABLE DRIVE
FF80:032B E7 00                      OUT   P1PTA,AX
FF80:032D C6 06 48 20 1E             MOV   SPEED,30              ;SET SPEED VALUE
FF80:0332 C7 06 6B 20 00 00          MOV   WORD PTR RAMPOS,0     ;SET RAMP UP
FF80:0338 BA 66 FF                   MOV   DX,T2MODE             ;NOW START TIMER 2
FF80:033B B8 01 E0                   MOV   AX,0E001H
FF80:033E EF                         OUT   DX,AX
FF80:033F C6 06 6A 20 01             MOV   DRVATV,1              ;SET DRIVE ACTIVE
FF80:0344 E5 80                      IN    AX,P2PTA              ;GET CURRENT LIGHTS
FF80:0346 25 97 FF                   AND   AX,0FF97H             ;STRIP IT
FF80:0349 F6 06 06 20 02             TEST  SYSTEM,2              ;IN MANUAL?
FF80:034E 75 02                      JNZ   DRVINB                ;IF MANUAL
FF80:0350 0C 08                      OR    AL,8                  ;SET CMD LIGHT
```

```
614  FF80:0352  F6 06 6E 20 01      DRVINB: TEST    LDIR,1          ;NEGATIVE?
615  FF80:0357  74 06                       JZ      DRVINC          ;IF POSITIVE
616  FF80:0359  0D 20 00                     OR      AX,32           ;SET DOWN LIGHT
617  FF80:035C  E9 03 00                     JMP     DRVIND
618  FF80:035F  0D 40 00             DRVINC: OR      AX,64           ;SET UP LIGHT
619  FF80:0362  E7 80                DRVIND: OUT     P2PTA,AX        ;SEND IT
620  FF80:0364  C6 06 53 20 19               MOV     EXITMR,25       ;INITIALIZE TIMER
621  FF80:0369  C3                           RET
622
623  FF80:036A                       LOCAL:
624  FF80:036A  80 0E 06 20 02               OR      SYSTEM,2        ;SET MANUAL FLAG
625  FF80:036F  F6 06 61 20 01               TEST    MANOVR,1        ;MANUAL OVERRIDE?
626  FF80:0374  74 12                        JZ      LOCLB           ;IF NOT
627  FF80:0376  F6 06 6A 20 01               TEST    DRVATV,1        ;DRIVE ACTIVE?
628  FF80:037B  75 03                        JNZ     LOCLA           ;IF NOT OFF
629  FF80:037D  E9 27 FD                     JMP     BOSS
630  FF80:0380  C6 06 69 20 01       LOCLA:  MOV     DRVOFF,1        ;SHUT DOWN
631  FF80:0385  E9 E2 FF                     JMP     LOCAL
632  FF80:0388  BC 00 21             LOCLB:  MOV     SP,STCK         ;SET STACK
633  FF80:038B  F6 06 08 20 01               TEST    FAUL1,1         ;APEX OK?
634  FF80:0390  74 03                        JZ      LC1             ;IF APEX OK
635  FF80:0392  E9 C9 FD                     JMP     RSCMD           ;IF APEX BROKE
636  FF80:0395  F6 06 6A 20 01       LC1:    TEST    DRVATV,1        ;DRIVE ACTIVE?
637  FF80:039A  75 3A                        JNZ     LC1B            ;IF ACTIVE
638  FF80:039C  E8 37 01                     CALL    DSTOR           ;GET DATA
639  FF80:039F  BA 95 01                     MOV     DX,MODESW       ;GET MODE SWITCH
640  FF80:03A2  EC                           IN      AL,DX           ;GET INFO
641  FF80:03A3  F6 D0                        NOT     AL              ;INVERT AL
642  FF80:03A5  3C FF                        CMP     AL,0FFH         ;A = 0FFH?
643  FF80:03A7  C6 06 59 20 00               MOV     SCWIGN,0        ;ASSUME ON
644  FF80:03AC  75 05                        JNZ     LOCLC           ;IF ON
645  FF80:03AE  C6 06 59 20 01               MOV     SCWIGN,1        ;ELSE TURN OFF
646  FF80:03B3  B4 00                LOCLC:  MOV     AH,0            ;SET AH=0
647  FF80:03B5  C6 06 65 20 00               MOV     BFLAG,0         ;SET BFLAG=0
648  FF80:03BA  2E 62 06 D8 06               BOUND   AX,CS:MANRG     ;MODE SWITCH IN RANGE?
649  FF80:03BF  F6 06 65 20 01               TEST    BFLAG,1         ;OUT OF BOUNDS?
650  FF80:03C4  74 03                        JZ      LC1A            ;IF IN RANGE
651  FF80:03C6  B8 00 00                     MOV     AX,0            ;PROCESS=NORMAL
652  FF80:03C9  03 C0                LC1A:   ADD     AX,AX           ;AX*2
653  FF80:03CB  8B D8                        MOV     BX,AX           ;BX = INDEX
654  FF80:03CD  2E 8B 87 DC 06               MOV     AX,CS:MANTBL[BX] ;GET ROUTINE
655  FF80:03D2  FF D0                        CALL    AX              ;DO ROUTINE
656  FF80:03D4  EB 07                        JMP     SHORT LC1C
657  FF80:03D6  FE 0E 53 20          LC1B:   DEC     BYTE PTR EXITMR ;DEC SAFETY TIMER
658  FF80:03DA  75 01                        JNZ     LC1C            ;IF OK
659  FF80:03DC  CC                           INT     3               ;DO TIMER THIS WAY
660  FF80:03DD  E5 80                LC1C:   IN      AX,P2PTA        ;GET SWITCH STATUS
661  FF80:03DF  F6 C4 80                     TEST    AH,128          ;DRIVE UP?
662  FF80:03E2  75 03                        JNZ     LC2             ;NOT DRIVE UP
663  FF80:03E4  E9 2C 00                     JMP     LUP             ;DRIVE UP
664  FF80:03E7  F6 C4 40             LC2:    TEST    AH,64           ;DRIVE DOWN
665  FF80:03EA  75 03                        JNZ     LC3             ;NOT DRIVE DOWN
666  FF80:03EC  E9 59 00                     JMP     LDWN            ;DRIVE DOWN
667  FF80:03EF  F6 06 6A 20 01       LC3:    TEST    DRVATV,1        ;DRIVE ACTIVE?
668  FF80:03F4  74 08                        JZ      LC4             ;IF NOT ACTIVE
669  FF80:03F6  C6 06 69 20 01               MOV     DRVOFF,1        ;SET DRIVE OFF
670  FF80:03FB  E9 6C FF                     JMP     LOCAL
```

```
671  FF80:03FE  F6 C4 20          LC4:   TEST   AH,32                ;STILL IN LOCAL
672  FF80:0401  74 03                    JZ     LC5                 ;IF IN LOCAL
673  FF80:0403  E9 58 FD                  JMP    RSCMD               ;LEAVE IF NOT IN LOCAL
674  FF80:0406  32 C0             LC5:    XOR    AL,AL               ;CLEAR ALL DRIVE FLAGS
675  FF80:0408  B9 05 00                  MOV    CX,OFFSET FLAGST+ENDFLG-DRVATV
676  FF80:040B  BF 6A 20                  MOV    DI,OFFSET DRVATV
677  FF80:040E  F3 AA                     REP STOS BYTE PTR DRVATV
678  FF80:0410  E9 57 FF                  JMP    LOCAL
679
680  FF80:0413                    LUP:
681  FF80:0413  F6 06 6A 20 01            TEST   DRVATV,1            ;ALREADY RUNNING?
682  FF80:0418  74 0F                     JZ     LUP1                ;IF NOT MOVING
683  FF80:041A  F6 06 6A 20 02            TEST   DRVATV,2            ;MOVING UP?
684  FF80:041F  75 05                     JNZ    LUP2                ;ALREADY GOING UP
685  FF80:0421  C6 06 69 20 01            MOV    DRVOFF,1            ;KILL DRIVE
686  FF80:0426  E9 41 FF          LUP2:   JMP    LOCAL
687  FF80:0429  C7 06 00 20 00 F2 LUP1:   MOV    WORD PTR POSCEC,HIGH ;SET UPPER BOUNDARY
688  FF80:042F  E8 A4 00                   CALL   DSTOR               ;GET DATA
689  FF80:0432  A1 02 20                   MOV    AX,POSD             ;GET CURRENT POSITION
690  FF80:0435  3D 00 F2                   CMP    AX,HIGH             ;< DRIVE LIMITS
691  FF80:0438  72 03                      JC     LUP3                ;IF LESS
692  FF80:043A  E9 2D FF                   JMP    LOCAL
693  FF80:043D  E8 DF FD          LUP3:    CALL   LOCTST              ;SET UP DRIVE PARAMETERS
694  FF80:0440  80 0E 6A 20 02             OR     DRVATV,2
695  FF80:0445  E9 22 FF                   JMP    LOCAL
696
697  FF80:0448                    LDWN:
698  FF80:0448  F6 06 6A 20 01             TEST   DRVATV,1            ;ALREADY RUNNING?
699  FF80:044D  74 0F                      JZ     LDWN1               ;IF NOT MOVING
700  FF80:044F  F6 06 6A 20 04             TEST   DRVATV,4            ;MOVING DOWN?
701  FF80:0454  75 05                      JNZ    LDWN2               ;ALREADY GOING DOWN
702  FF80:0456  C6 06 69 20 01             MOV    DRVOFF,1            ;KILL DRIVE
703  FF80:045B  E9 0C FF          LDWN2:   JMP    LOCAL
704  FF80:045E  C7 06 00 20 00 0A LDWN1:   MOV    WORD PTR POSCEC,LOW ;SET LOWER BOUNDARY
705  FF80:0464  E8 6F 00                   CALL   DSTOR               ;GET DATA
706  FF80:0467  A1 02 20                   MOV    AX,POSD             ;GET CURRENT POSITION
707  FF80:046A  3D 00 0A                   CMP    AX,LOW              ;> DRIVE LIMITS
708  FF80:046D  73 03                      JNC    LDWN3               ;IF LESS
709  FF80:046F  E9 F8 FE                   JMP    LOCAL
710  FF80:0472  E8 AA FD          LDWN3:   CALL   LOCTST              ;SET UP DRIVE PARAMETERS
711  FF80:0475  80 0E 6A 20 04             OR     DRVATV,4            ;SET DOWN FLAG
712  FF80:047A  E9 ED FE                   JMP    LOCAL
713
714  FF80:047D                    NORMIT:
715  FF80:047D  E8 CA 01                   CALL   BRKOFF              ;TURN OFF BRAKE
716  FF80:0480  E9 06 00                   JMP    ZERDA               ;ZERO D/A
717
718  FF80:0483                    NEGDA:
719  FF80:0483  BB 00 08                   MOV    BX,800H             ;SET -10V
720  FF80:0486  E9 09 00                   JMP    SETDA
721
722  FF80:0489                    ZERDA:
723  FF80:0489  BB 00 00                   MOV    BX,0                ;SET 0V
724  FF80:048C  E9 03 00                   JMP    SETDA
725
726  FF80:048F                    POSDA:
727  FF80:048F  BB FF 07                   MOV    BX,7FFH             ;SET +10V
```

```
728  FF80:0492                    SETDA:
729  FF80:0492  E5 00                     IN      AX,P1PTA          ;GET CURRENT STATUS
730  FF80:0494  25 00 F0                  AND     AX,0F000H         ;JUST STATUS
731  FF80:0497  0B C3                     OR      AX,BX             ;MERGE
732  FF80:0499  E7 00                     OUT     P1PTA,AX          ;SEND IT
733  FF80:049B  BA 96 01                  MOV     DX,LTCHDA         ;NOW LATCH IT IN
734  FF80:049E  EE                        OUT     DX,AL
735  FF80:049F  C3                        RET
736
737  FF80:04A0  E5 80             NEGAD:  IN      AX,P2PTA          ;GET PORT 2
738  FF80:04A2  25 F8 FF                  AND     AX,0FFF8H         ;STRIP OLD A/D REQ
739  FF80:04A5  0D 03 00                  OR      AX,3              ;REQUEST -10V
740  FF80:04A8  EB 12                     JMP     SHORT READAD
741
742  FF80:04AA  E5 80             ZERAD:  IN      AX,P2PTA          ;GET PORT 2
743  FF80:04AC  25 F8 FF                  AND     AX,0FFF8H         ;STRIP OLD A/D REQ
744  FF80:04AF  0D 04 00                  OR      AX,4              ;REQUEST 0V
745  FF80:04B2  EB 08                     JMP     SHORT READAD
746
747  FF80:04B4  E5 80             POSAD:  IN      AX,P2PTA          ;GET PORT 2
748  FF80:04B6  25 F8 FF                  AND     AX,0FFF8H         ;STRIP OLD A/D REQ
749  FF80:04B9  0D 02 00                  OR      AX,2              ;REQUEST +10V
750
751  FF80:04BC  C6 06 64 20 00    READAD: MOV     ADFLAG,0          ;RESET FLAG
752  FF80:04C1  E7 80                     OUT     P2PTA,AX
753  FF80:04C3  BA 94 01                  MOV     DX,STCNV          ;START CONVERSION
754  FF80:04C6  EE                        OUT     DX,AL
755  FF80:04C7  FB                        STI                       ;ENABLE INTERRUPTS
756  FF80:04C8  F6 06 64 20 01    READA1: TEST    ADFLAG,1          ;ANALOG READY?
757  FF80:04CD  74 F9                     JZ      READA1            ;IF NOT
758  FF80:04CF  A1 46 20                  MOV     AX,ADVAL          ;GET CURRENT VELOCITY
759  FF80:04D2  A3 28 20                  MOV     VEL1,AX           ;SAVE VELOCITY
760  FF80:04D5  C3                        RET
761
762  FF80:04D6                    DSTOR:
763  FF80:04D6  32 C0                     XOR     AL,AL
764  FF80:04D8  BA 80 01                  MOV     DX,APEXRQ         ;GET APEX DATA
765  FF80:04DB  EE                        OUT     DX,AL
766  FF80:04DC  83 C2 02                  ADD     DX,2              ;POINT TO APEXRS
767  FF80:04DF  B9 0C 00                  MOV     CX,12             ;12 TRIALS
768  FF80:04E2                    DSTOR1:
769  FF80:04E2  EC                        IN      AL,DX             ;SEE IF DATA READY
770  FF80:04E3  24 C0                     AND     AL,0C0H           ;STRIP DATA
771  FF80:04E5  3C 80                     CMP     AL,80H            ;LOOK FOR PATTERN
772  FF80:04E7  74 0A                     JZ      DSTOR2            ;IF READY
773  FF80:04E9  E2 F7                     LOOP    DSTOR1            ;IF NOT
774  FF80:04EB  80 0E 08 20 01            OR      FAUL1,1           ;SET APEX DEAD
775  FF80:04F0  E9 C9 00                  JMP     DSTOR6            ;SKIP APEX GATHERING
776  FF80:04F3                    DSTOR2:
777  FF80:04F3  80 26 08 20 FE            AND     FAUL1,0FEH        ;SET APEX OK
778  FF80:04F8  4A                        DEC     DX
779  FF80:04F9  ED                        IN      AX,DX             ;GET POSITION
780  FF80:04FA  8B D8                     MOV     BX,AX             ;PUT INTO BX
781  FF80:04FC  C1 E3 02                  SHL     BX,2              ;GET RID OF RESPONSE BITS
782  FF80:04FF  83 C2 02                  ADD     DX,2              ;POINT TO NEXT SECTION
783  FF80:0502  ED                        IN      AX,DX             ;GET NEXT INFO
784  FF80:0503  C0 C4 02                  ROL     AH,2              ;GET FINAL POSITION
```

```
785  FF80:0506  80 E4 03              AND     AH,03                   ;PREPARE TO MERGE
786  FF80:0509  0A DC                 OR      BL,AH                   ;MERGE BITS
787  FF80:050B  89 1E 02 20           MOV     POSD,BX                 ;SAVE POSITION
788  FF80:050F  A1 00 20              MOV     AX,POSCEC
789  FF80:0512  2B C3                 SUB     AX,BX                   ;GET ERROR
790  FF80:0514  C6 06 6D 20 00        MOV     DIR,0                   ;ASSUME POSITIVE
791  FF80:0519  73 05                 JNC     DST02A                  ;IF POSITIVE
792  FF80:051B  C6 06 6D 20 01        MOV     DIR,1                   ;SET NEGATIVE
793  FF80:0520                  DST02A:
794  FF80:0520  A3 04 20              MOV     ERROR,AX                ;STORE IT HERE
795  FF80:0523  ED                    IN      AX,DX                   ;GET VELOCITY
796  FF80:0524  8A F8                 MOV     BH,AL                   ;SAVE PARTIAL DATA
797  FF80:0526  C1 E8 02              SHR     AX,2
798  FF80:0529  25 FF 0F              AND     AX,0FFFH                ;STRIP UPPER NYBBLE
799  FF80:052C  A3 14 20              MOV     VEL,AX                  ;SAVE VELOCITY
800  FF80:052F  83 C2 02              ADD     DX,2                    ;POINT TO ANALOGS
801  FF80:0532  ED                    IN      AX,DX                   ;GET DATA
802  FF80:0533  C1 E8 02              SHR     AX,2                    ;SHIFT INTO POSITION
803  FF80:0536  C0 CF 02              ROR     BH,2                    ;SHIFT PARTIAL
804  FF80:0539  81 E3 00 C0           AND     BX,0C000H               ;PREPAR TO MERGE
805  FF80:053D  0B D8                 OR      BX,AX                   ;SAVE FOR FURTHER PROCESSING
806  FF80:053F  83 C2 02              ADD     DX,2                    ;POINT TO DISCRETE INFO
807  FF80:0542  EC                    IN      AL,DX                   ;GET DATA
808  FF80:0543  24 3F                 AND     AL,3FH                  ;ONLY DISCRETES
809  FF80:0545  80 26 06 20 FB        AND     SYSTEM,0FBH             ;ASSUME BRAKE NOT RELEASED
810  FF80:054A  A8 10                 TEST    AL,16                   ;SEE IF BRAKE IS RELEASED
811  FF80:054C  74 05                 JZ      DSTOR3
812  FF80:054E  80 0E 06 20 04        OR      SYSTEM,4                ;SET BRAKE RELEASE
813  FF80:0553  A2 58 20        DSTOR3: MOV     PHASEA,AL               ;LOAD CURRENT SCREW LEVEL
814  FF80:0556  80 26 58 20 20        AND     PHASEA,32               ;MASK IT
815  FF80:055B  24 0F                 AND     AL,0FH                  ;SET LIMIT INFO
816  FF80:055D  D0 E0                 SHL     AL,1
817  FF80:055F  80 26 08 20 E1        AND     FAUL1,0E1H              ;CLEAR LIMIT FAULTS
818  FF80:0564  08 06 08 20           OR      FAUL1,AL
819  FF80:0568  8B D3                 MOV     DX,BX                   ;SAVE TABLE
820  FF80:056A  81 CB 0F 00           OR      BX,0FH                  ;MAKE IT LIKE STANDARD ANALOG
821  FF80:056E  81 E2 0F 00           AND     DX,0000FH               ;DEVELOP TABLE ADDRESS
822  FF80:0572  01 16 5A 20           ADD     RANDOM,DX               ;DEVELOP RANDOM NUMBER
823  FF80:0576  81 26 5A 20 3F 00     AND     WORD PTR RANDOM,3FH
824  FF80:057C  FE 06 5A 20           INC     RANDOM
825  FF80:0580  67 D3                 XCHG    DX,BX
826  FF80:0582  83 FB 03              CMP     BX,3
827  FF80:0585  7E 35                 JLE     DSTOR6
828  FF80:0587  83 EB 04              SUB     BX,4
829  FF80:058A  8B CB                 MOV     CX,BX                   ;SAVE ANALOG POINTER IN C
830  FF80:058C  83 C1 01              ADD     CX,1                    ;PLUS 1 FOR BIT PLACEMENT
831  FF80:058F  D1 E3                 SHL     BX,1
832  FF80:0591  89 97 0E 20           MOV     ANADT[BX],DX            ;SAVE ANALOG DATA
833  FF80:0595  D1 E3                 SHL     BX,1                    ;NOW FOR BOUND CHECK
834  FF80:0597  C6 06 65 20 00        MOV     BFLAG,0                 ;BOUND FLAG = OK
835  FF80:059C  2E 62 97 A4 06        BOUND   DX,CS:ANATAB[BX]        ;CHECK ANALOG BOUNDS
836  FF80:05A1  B8 00 80              MOV     AX,8000H                ;ASSUME FAIL
837  FF80:05A4                  DSTOR4:
838  FF80:05A4  D1 C0                 ROL     AX,1                    ;PLACE FLAG BIT
839  FF80:05A6  E2 FC                 LOOP    DSTOR4
840  FF80:05A8  F6 06 65 20 01        TEST    BFLAG,1                 ;BOUND OK?
841  FF80:05AD  75 09                 JNZ     DSTOR5                  ;IF BAD
```

```
842  FF80:05AF  F7 D0                    NOT   AX                    ;SET OK
843  FF80:05B1  21 06 0C 20              AND   ANAFL,AX               ;STRIP BAD ANALOG FLAG
844  FF80:05B5  E9 04 00                 JMP   DSTOR6
845  FF80:05B8                    DSTOR5:
846  FF80:05B8  09 06 0C 20              OR    ANAFL,AX               ;SET BAD ANALOG FLAG
847  FF80:05BC                    DSTOR6:
848  FF80:05BC  E5 80                    IN    AX,P2PTA               ;GET LED STATUS
849  FF80:05BE  25 6F FE                 AND   AX,0FE6FH              ;KILL ALL LIGHTS
850  FF80:05C1  F6 06 06 20 04           TEST  SYSTEM,4               ;IS BRAKE RELEASED?
851  FF80:05C6  74 03                    JZ    DSTOR7                 ;IF NOT
852  FF80:05C8  0D 10 00                 OR    AX,16                  ;SET BRAKE LED ON
853  FF80:05CB                    DSTOR7:
854  FF80:05CB  8B 1E 08 20              MOV   BX,FAUL1               ;NOW DO DOWN LIMIT
855  FF80:05CF  81 E3 06 00              AND   BX,6
856  FF80:05D3  74 03                    JZ    DSTOR8                 ;IF SET LIGHT LED
857  FF80:05D5  0D 80 00                 OR    AX,128                 ;SET DOWN LIMIT LED ON
858  FF80:05D8                    DSTOR8:
859  FF80:05D8  8B 1E 08 20              MOV   BX,FAUL1               ;NOW DO UP LIMIT
860  FF80:05DC  81 E3 18 00              AND   BX,24
861  FF80:05E0  74 03                    JZ    DSTOR9                 ;IF SET LIGHT LED
862  FF80:05E2  0D 00 01                 OR    AX,256                 ;SET UP LIMIT LED ON
863  FF80:05E5                    DSTOR9:
864  FF80:05E5  E7 80                    OUT   P2PTA,AX
865  FF80:05E7  BA 97 01                 MOV   DX,BDERL1              ;GET IDD STUFF
866  FF80:05EA  EC                       IN    AL,DX
867  FF80:05EB  F6 D0                    NOT   AL
868  FF80:05ED  A2 0A 20                 MOV   FAUL2,AL               ;SAVE IN FAUL2
869  FF80:05F0  81 26 08 20 FF E7        AND   WORD PTR FAUL1,0E7FFH  ;STRIP OUT FAULTS
870  FF80:05F6  E5 80                    IN    AX,P2PTA               ;GET E-STOP, DRIVE LOCKOUT
871  FF80:05F8  25 00 18                 AND   AX,1800H               ;STRIP JUNK
872  FF80:05FB  80 F4 10                 XOR   AH,10H                 ;INVERT DRIVE LOCKOUT
873  FF80:05FE  09 06 08 20              OR    WORD PTR FAUL1,AX      ;MERGE
874  FF80:0602  C3                       RET
875
876  FF80:0603                    BDS3RS:
877  FF80:0603  E5 00                    IN    AX,P1PTA               ;GET CONTROL PORT
878  FF80:0605  0D 00 20                 OR    AX,2000H               ;DO RESET
879  FF80:0608  E7 00                    OUT   P1PTA,AX
880  FF80:060A  B9 01 00                 MOV   CX,1                   ;DELAY 1 SEC
881  FF80:060D  E8 63 00                 CALL  DELAY
882  FF80:0610  E5 00                    IN    AX,P1PTA               ;GET CONTOL PORT AGAIN
883  FF80:0612  25 FF DF                 AND   AX,0DFFFH              ;KILL RESET
884  FF80:0615  E7 00                    OUT   P1PTA,AX
885  FF80:0617  C3                       RET
886
887  FF80:0618                    BRKON:
888  FF80:0618  E5 00                    IN    AX,P1PTA               ;GET CONTROL PORT
889  FF80:061A  0D 00 40                 OR    AX,4000H               ;RELEASE THE BRAKE
890  FF80:061D  E7 00                    OUT   P1PTA,AX               ;DO IT
891  FF80:061F  C6 06 67 20 00           MOV   TFLAG1,0               ;SET TIMER FLAG = OK
892  FF80:0624  E8 5E 00                 CALL  SEC1                   ;1 SEC
893  FF80:0627  E8 AC FE          BRKON1: CALL  DSTOR                  ;GET UPDATE
894  FF80:062A  F6 06 06 20 04           TEST  SYSTEM,4               ;BRAKE FREE?
895  FF80:062F  75 10                    JNZ   BRKON2                 ;IF FREE CONTINUE
896  FF80:0631  F6 06 67 20 01           TEST  TFLAG1,1               ;TIMED OUT?
897  FF80:0636  74 EF                    JZ    BRKON1                 ;IF NOT
898  FF80:0638  81 0E 08 20 00 01        OR    WORD PTR FAUL1,256     ;SET BRAKE FAULT
```

```
899  FF80:063E  E9 1D FB              JMP   RSCMD                ;RESET COMMAND
900  FF80:0641                  BRKON2:
901  FF80:0641  81 26 08 20 FF FE    AND   WORD PTR FAUL1,0FEFFH  ;CLEAR BRAKE FAULT
902  FF80:0647  E9 4A 00              JMP   TMROFF               ;STOP TIMER
903
904  FF80:064A                  BRKOFF:
905  FF80:064A  E5 00                 IN    AX,P1PTA             ;GET CONTROL PORT
906  FF80:064C  25 FF BF              AND   AX,0BFFFH            ;ENGAGE THE BRAKE
907  FF80:064F  E7 00                 OUT   P1PTA,AX             ;DO IT
908  FF80:0651  C6 06 67 20 00        MOV   TFLAG1,0             ;SET TIMER FLAG = OK
909  FF80:0656  E8 2C 00              CALL  SEC1                 ;1 SEC
910  FF80:0659  E8 7A FE        BRKOF1: CALL  DSTOR                ;GET UPDATE
911  FF80:065C  F6 06 06 20 04        TEST  SYSTEM,4             ;BRAKE FREE?
912  FF80:0661  74 31                 JZ    TMROFF               ;IF FREE CONTINUE
913  FF80:0663  F6 06 67 20 01        TEST  TFLAG1,1             ;TIMED OUT?
914  FF80:0668  74 5F                 JZ    BRKOF1               ;IF NOT
915  FF80:066A  81 0E 08 20 00 01     OR    WORD PTR FAUL1,256   ;SET BRAKE FAULT
916  FF80:0670  E9 EB FA              JMP   RSCMD                ;RESET COMMAND
917
918  FF80:0673  E8 0F 00        DELAY:  CALL  SEC1                 ;START TIMER FOR 1 SECOND
919  FF80:0676  C6 06 67 20 00  DELA1:  MOV   TFLAG1,0             ;RESET TFLAG
920  FF80:067B  F6 06 67 20 01  DELA2:  TEST  TFLAG1,1             ;TIME OUT?
921  FF80:0680  74 F9                   JZ    DELA2               ;WAIT
922  FF80:0682  E2 F2                   LOOP  DELA1               ;UNTIL TIME HAS PASSED
923  FF80:0684  C3                      RET
924
925  FF80:0685                  SEC1:
926  FF80:0685  BA 5E FF              MOV   DX,T1MODE            ;ENABLE TIMER1 FIRST
927  FF80:0688  B8 09 E0              MOV   AX,0E009H
928  FF80:068B  EF                    OUT   DX,AX
929  FF80:068C  BA 66 FF              MOV   DX,T2MODE            ;NOW TIMER2
930  FF80:068F  B8 01 C0              MOV   AX,0C001H
931  FF80:0692  EF                    OUT   DX,AX
932  FF80:0693  C3                    RET
933
934  FF80:0694                  TMROFF:
935  FF80:0694  BE 00 00              MOV   SI,OFFSET TSETUP     ;POINT TO TABLE
936  FF80:0697  BA 50 FF              MOV   DX,TOCOUT            ;POINT TO SOURCE POINTER
937  FF80:069A  B9 0C 00              MOV   CX,12
938  FF80:069D  6F              TSET: OUTSW
939  FF80:069E  83 C2 02              ADD   DX,2
940  FF80:06A1  E2 FA                 LOOP  TSET
941  FF80:06A3  C3                    RET
942  FF80:06A4
943
944                             ;      TABLES FOR BOUND CHECKS
945
946
947  FF80:06A4  0080            ANATAB: .WORD  8000H               ;GND1
948  FF80:06A6  FF7F                    .WORD  7FFFH
949  FF80:06A8  0080                    .WORD  8000H               ;GND2
950  FF80:06AA  FF7F                    .WORD  7FFFH
951  FF80:06AC  0080                    .WORD  8000H               ;GND3
952  FF80:06AE  FF7F                    .WORD  7FFFH
953  FF80:06B0  0080                    .WORD  8000H               ;FOCUS VELOCITY
954  FF80:06B2  FF7F                    .WORD  7FFFH
955  FF80:06B4  0080                    .WORD  8000H               ;+15V/2
```

```
956  FF80:06B6  FF7F                    .WORD   7FFFH
957  FF80:06B8  0080                    .WORD   8000H                ;-15V/2
958  FF80:06BA  FF7F                    .WORD   7FFFH
959  FF80:06BC  0080                    .WORD   8000H                ;+5V
960  FF80:06BE  FF7F                    .WORD   7FFFH
961  FF80:06C0  0080                    .WORD   8000H                ;+10V
962  FF80:06C2  FF7F                    .WORD   7FFFH
963  FF80:06C4  0080                    .WORD   8000H                ;GND
964  FF80:06C6  FF7F                    .WORD   7FFFH
965  FF80:06C8  0080                    .WORD   8000H                ;-10V
966  FF80:06CA  FF7F                    .WORD   7FFFH
967  FF80:06CC  0080                    .WORD   8000H                ;MOUNT TEMP 2
968  FF80:06CE  FF7F                    .WORD   7FFFH
969  FF80:06D0  0080                    .WORD   8000H                ;BIN TEMP
970  FF80:06D2  FF7F                    .WORD   7FFFH
971
972  FF80:06D4                  CLSETAB:
973  FF80:06D4  F4FF                    .WORD   0FFF4H               ;12 COUNTS BEFORE WE MOVE
974  FF80:06D6  0C00                    .WORD   0000CH
975
976  FF80:06D8                  MANRG:
977  FF80:06D8  0000                    .WORD   0                    ;MODES 0 - 4 ACTIVE
978  FF80:06DA  0700                    .WORD   7
979
980                             ;       MODE SWITCH ENTRY CONTROL
981
982  FF80:06DC                  MANTBL:
983  FF80:06DC  7D04                    .WORD   NORMIT
984  FF80:06DE  1806                    .WORD   BRKON
985  FF80:06E0  8304                    .WORD   NEGDA
986  FF80:06E2  8904                    .WORD   ZERDA
987  FF80:06E4  8F04                    .WORD   POSDA
988  FF80:06E6  A004                    .WORD   NEGAD
989  FF80:06E8  AA04                    .WORD   ZERAD
990  FF80:06EA  B404                    .WORD   POSAD
991
992
993  FF80:07F0                          ORG     07F0H
994
995                             ;THIS AREA IS RESERVED FOR RESET PARAMETERS.  CANNOT BE LARGER THEN 16 BYTES
996
997  FF80:07F0  BA A0 FF                MOV     DX,UMCS              ;MAKE UPPER MEMORY 2K BLOCK
998  FF80:07F3  B8 BD FF                MOV     AX,UMBS
999  FF80:07F6  EF                      OUT     DX,AX
1000 FF80:07F7  EA 00 00 80 FF          JMP     FAR INITAL           ;ENTER INTO LOWER EPROM SECTION
1001
1002 FF80:07FC                          END


        Lines Assembled :  1002          Assembly Errors :  0
```