VLBA Technical Report No. 43
THE VLBA CORRELATOR
PLAYBACK INTERFACE SUB-SYSTEM
VOLUME 1 of 2

Chuck Broadwell

Ray Escoffier

Gene Runion

January 1, 1998

pbiman.vol1.doc Microsoft Word not under SCCS control

master file stored in /corrdwgs/manuals/pbiman/SCCS
(where the SCCS file would have been stored)

This page intentionally left blank

TABLE OF CONTENTS

VOLUME 1

TABLE OF FIGURES

VOLUME 1

LIST OF TABLES

VOLUME 1

Chapter 1 The VLBA Correlator Playback Interface System

1.1.0) INTRODUCTION

The VLBA Correlator provides a Playback Interface System (PBI)that handles data recovery from 24 Playback Drives (PBD). See Figure 1 VLBA CORRELATOR RACK LAYOUT for a front view of the VLBA Correlator racks, showing the PBI bins. A single PBI consists of two Track Recovery Cards (TRC), described in Chapter 2 of this manual, and one Deformatter Card (DEF), described in Chapter 5.

As seen in Figure 1, there are four PBI bins, each containing six PBI. Each PBI connects to a PBD over a set of four multi-signal flat cables that exit the rear of the rack and go under the floor to the PBD.

Each PBI contains a total of five embedded microprocessors. There are two 87C51 processors on each TRC, and one 68000 processor on each Deformatter. These processors communicate with the VME system seen in Figure 1, over a byte wide Hardware Control Bus (HCB) that is referred to in the figure. The Deformatter is connected to the HCB. The TRC communicates through the Deformatter, rather than being directly connected to the HCB. Data transfers from the VME to the TRC are transparent to the Deformatter in that the Deformatter passes the bytes directly without interpreting them.

Tape frames from the PBD are clocked at a nominal rate of 9 MHz into the TRC. The nominal time between frames at playback is 2.5 msec. The time between frames at record time is 2.5, 5.0 or 10.0 msec, depending on the record rate (8, 4 or 2 MHz respectively). Data bits are extracted from the tape frames (by the TRC) to produce 8 MHz data streams into the Deformatter. In the Deformatter, the 8 MHz data streams are re-combined (or otherwise manipulated) to produce 32 MHz data streams into the FFT cards.

For test purposes, a simulated tape frame is generated in the Master Control Card (MCC) located in the MAC bin of each rack. This data source may be selected in place of the PBD data for system testing. This data may alternately be routed over to the PBD and back, in a loopback test, in order to test the cable interface between the PBD and PBI.

See Figure 2 PBI BIN CARD LAYOUT for the layout of the PBI cards in a single PBI bin.

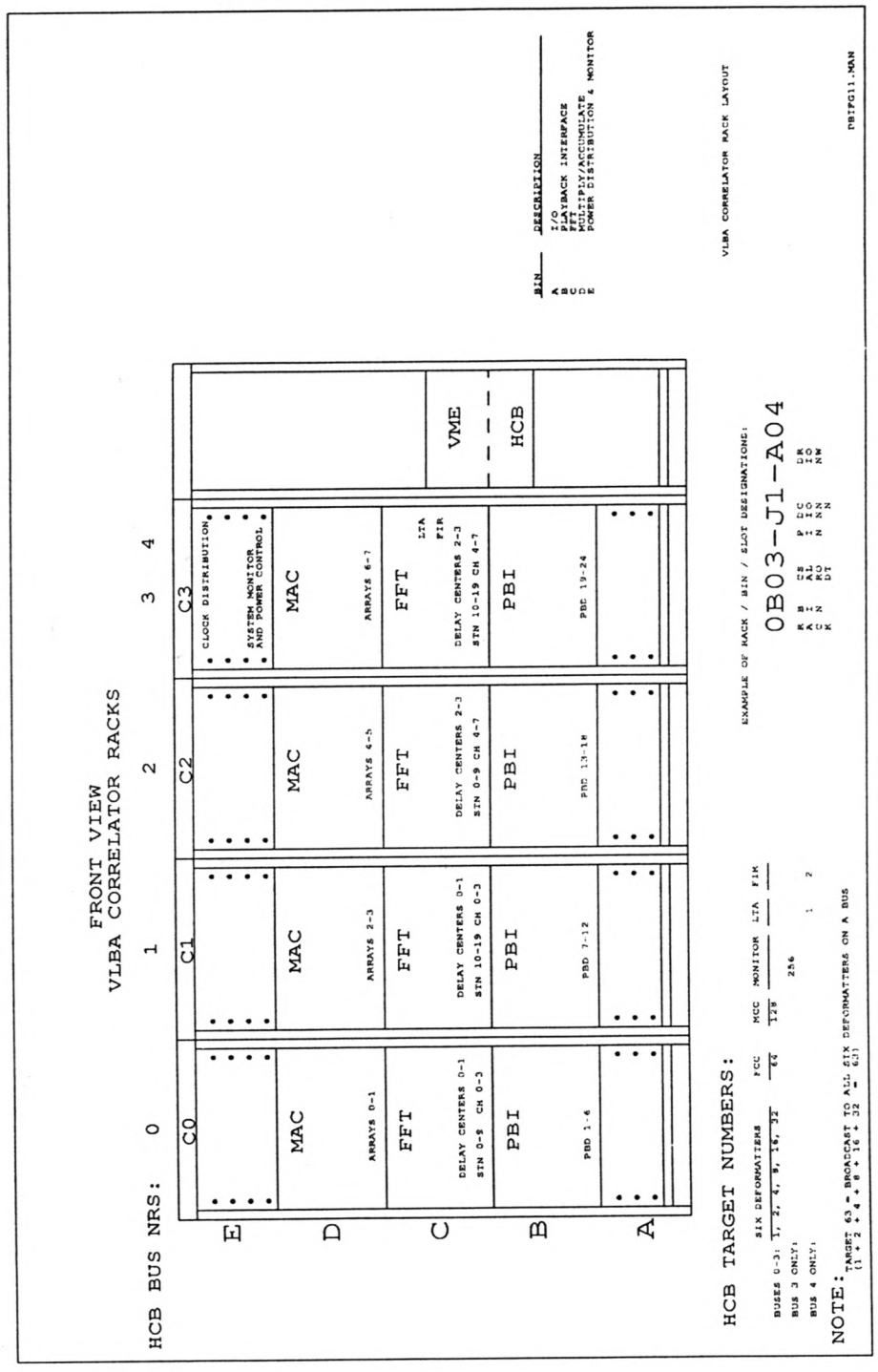
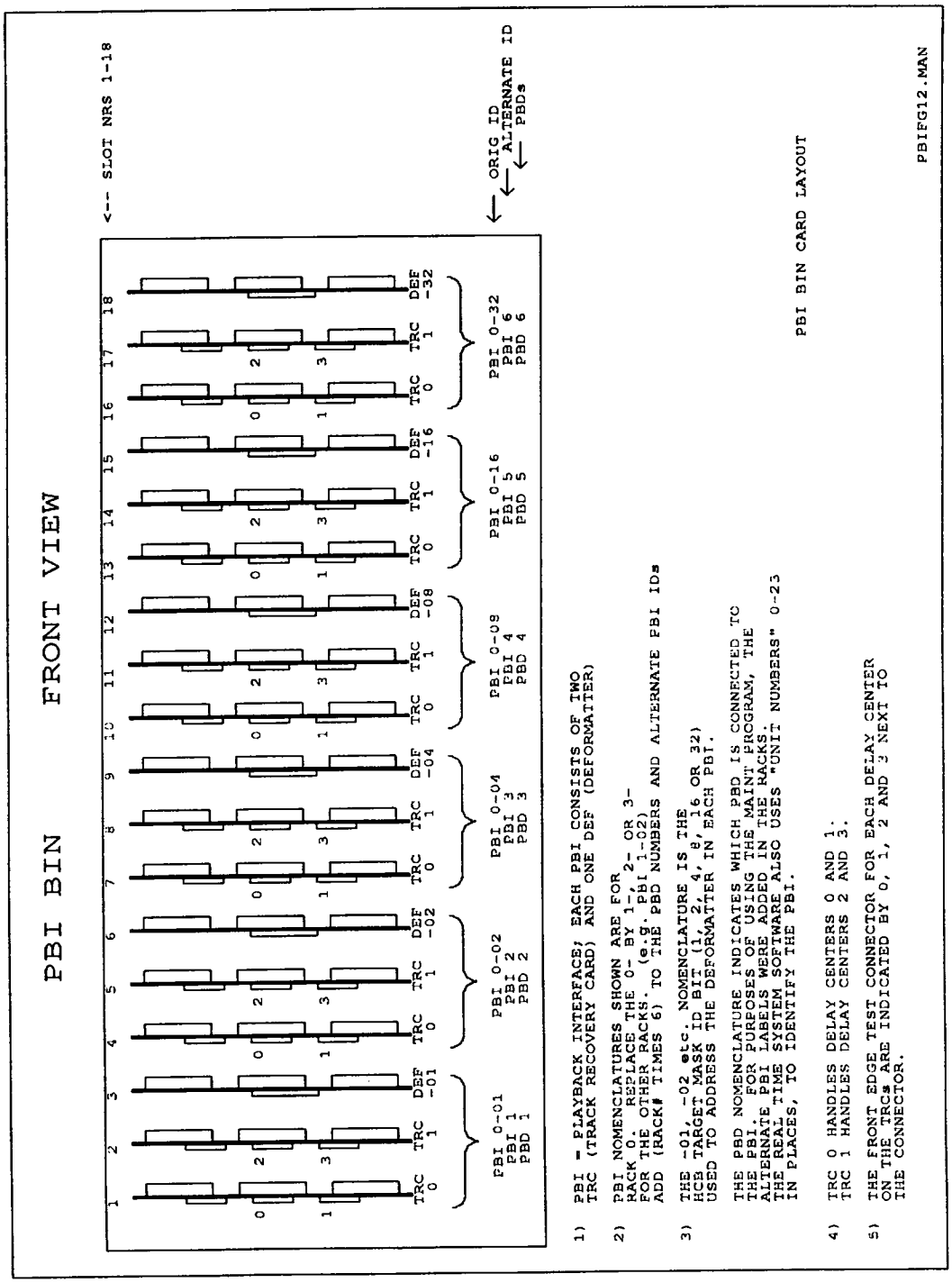


Figure 1 VLBA CORRELATOR RACK LAYOUT



1) PBI = PLAYBACK INTERFACE; EACH PBI CONSISTS OF TWO TRC (TRACK RECOVERY CARD) AND ONE DEF (DEFORMATTER)

2) PBI NOMENCLATURES SHOWN ARE FOR RACK 0. REPLACE THE 0- BY 1-, 2- OR 3- FOR THE OTHER RACKS. (e.g. PBI 1-02) ADD (RACK# TIMES 6) TO THE PBD NUMBERS AND ALTERNATE PBI IDs

3) THE -01, -02 etc. NOMENCLATURE IS THE HCB TARGET MASK ID BIT (1, 2, 4, 8, 16 OR 32) USED TO ADDRESS THE DEFORMATTER IN EACH PBI.

THE PBD NOMENCLATURE INDICATES WHICH PBD IS CONNECTED TO THE PBI. FOR PURPOSES OF USING THE MAINT PROGRAM, THE ALTERNATE PBI LABELS WERE ADDED IN THE RACKS. THE REAL TIME SYSTEM SOFTWARE ALSO USES "UNIT NUMBERS" 0-23 IN PLACES, TO IDENTIFY THE PBI.

4) TRC 0 HANDLES DELAY CENTERS 0 AND 1.
TRC 1 HANDLES DELAY CENTERS 2 AND 3.

5) THE FRONT EDGE TEST CONNECTOR FOR EACH DELAY CENTER ON THE TRCs ARE INDICATED BY 0, 1, 2 AND 3 NEXT TO THE CONNECTOR.

Figure 2 PBI BIN CARD LAYOUT

1.2.0) VLBA Correlator Drawings

All hardware and firmware related files for the VLBA correlator are maintained under SCCS version control, in one of two areas. These two areas are identified as the vlbsoft and corrdwgs areas. The path to vlbsoft is /home/magnolia2/vlbsoft and the path to corrdwgs is /home/azalea/corrdwgs (as of May 1998). All drawing filenames are lower case, but may appear in documentation as either upper or lower case.

The vlbsoft area is where all source code for the correlator is maintained. It is also used to store all source files that are processed in any manner to produce objects that are downloaded into the correlator, or programmed into devices that are installed in the correlator. This includes pal source code, Xilinx design files and sequencer source files.

The corrdwgs area is where Orcad schematic files, netlists, partlists, etc. are maintained.

Appendix I of this Technical Report contains a complete list of all drawings for both the TRC and Deformatter.

Volume 2 of this Technical Report contains many of the schematics referred to in Volume 1. Drawings for test fixtures are not included in this report.

Copies of many Postscript files for drawings are presently stored in the /home/w5uxh/plots directory.

1.3.0) Test Fixtures and The PBD Simulator

To support testing of the PBI cards, there are two PBI test fixtures. PBI Test Fixtures #1 and #2 are described in Chapter 6. A third test capability is provided by the VLBA Correlator Testbed Rack. This is a "two station" version of the correlator, using spare cards in a rack built for test purposes.

The PBD Simulator is the Mini-Transport, described in Chapter 7. The output of the Mini-Transport can be connected to PBI Test Fixture #2 or to the Testbed rack, to provide simulated track data and track clock inputs to the Track Recovery Cards.

Chapter 2 The Track Recovery Card

2.1.0) Track Recovery Card Introduction

The Track Recovery Card (TRC) is used to interface a Play Back Drive (PBD) to the VLBA correlator. The functions of one TRC include:

1) interface with 18 individual PBD playback tracks, along with their 18 respective track clocks.

2) remove mechanically induced variations in signal timing from the data streams (wow and flutter).

3) remove individual time offsets from the data streams (due to head skew).

4) find and use the imbedded sync pattern in each data stream to synchronize frame counters to facilitate de-framing of the track signals.

5) recover frame header information, especially the frame time codes, imbedded in each track signal and make the contents of the headers available to the real time VME computer system.

6) perform signal quality analysis of the track frames (parity error counts, CRC checks, frame sync integrity, etc.) and use the results of the analysis to pass validity judgement on individual frames.

7) provide large data buffers to help synchronize recovered track data to the correlator system clock and to assist in the PBD servo mechanism.

8) set the integer portion of the station delay as commanded by the FFT Control Card by way of the Deformatter card (DEF).

9) provide a port for test data insertion at an early point in the VLBA correlator data stream.

2.2.0) The Playback Drive

A PBD has 36 playback reproduce heads, each designed to recover track information recorded during an observation at a remote station. Since each TRC is capable of handling 18 PBD tracks, the correlator has 2 TRCs assigned to each PBD. A given experiment, however, may require only a subset of the read heads to be active at a time. The exact number of active tracks to be processed from a PBD varies with experiment within the range of 1 to all 36.

Of the 36 possible tracks provided by the PBD, 32 are normal data tracks and 4 are referred to as system tracks. The system tracks may be put to various applications such as being used as spare tracks (if a normal track is known to be defective at record time, the data that would normally be recorded on that track can be patched to a system track allowing the recorder to be used until corrective measures can be taken to fix the bad head). Additional system track applications include cross track parity in which parity encoded across active data tracks can be recorded on a system track. In general, however, the system tracks will probably not be used in any way that concerns the TRC (i.e., the system tracks will always be set inactive by the TRC).

The footprint of a bit on the tape is extremely small (on the order of a wavelength of light!) and hence very small offsets in the exact position of each head will cause skews of up to several hundred bits in the playback data streams relative to one another . In addition, a small individual component of the total wow and flutter may exist in each track. For these reasons each active track signal has an individual clock. Both the data and clock signals for a given track are transmitted to the correlator at differential ECL logic levels. The playback data rate is always approximately 9 MHz which, considering the frame overhead, yields a de-framed data rate of approximately 8 MHz (the exact rate deviates from the 8 MHz figure because of the 512/516 FFT cycle factor and by the station doppler component).

The PBD track signals are blocked into frames of exactly 2.5 msec duration (at the playback rate). Each frame consists of 20,000 data bits, 2500 data parity bits, and 180 header bits (in VLBA mode). Each header contains 20 9 bit parity bytes (8 aux data bytes, 4 sync pattern bytes and 8 time/CRC bytes). All parity is encoded in odd parity except for the 4 sync bytes which are even parity. See Appendix V, for the VLBA longitudinal tape format specification, A56000N003.

2.3.0) The Track Recovery Card

Each TRC provides for the recovery of 18 PBD tracks in 2 sets of 9, with the processing of each set being supervised by an 87C51 microprocessor. Each microprocessor sets the delay for one phase center which results in the correlator specification of 4 delay phase centers per PBI (there is one PBI per VLBA station).

Refer to the TRC schematic sheets, L007D01.SCH through L007D04.SCH in Volume 2, for the following circuit descriptions.

The TRC schematics refer to "track numbers" in a generic fashion as follows:

Delay Center	"TRACKS"
0	0-7
1	8-15
2	16-23
3	24-31

These numbers bear no direct relation to Formatter or Tape track numbers.

2.3.1) Data Input

The input circuitry for track data and clock, seen on the left side of sheets 3 and 4 of the schematic, consists of two 26LS32 differential receivers, one for the track data and one for the clock. As mentioned above, the track data and clock each are received in differential ECL and have clock rates of approximately 9 MHz. The 26LS32 line receiver is powered only from the +5 VDC supply, but is still capable of receiving negative going signals. A 100 ohm line-to-line termination resistor is also provided for each differential line on the TRC card.

After the line receiver, a 74ALS74 flip-flop capture stage is provided. This stage uses the individual track clocks to capture data signals onto the card. The track input logic for two system tracks is seen in the upper left hand corner of sheet 2 of the schematic.

2.3.2) Track Mix Stages

Following the input logic on the TRC schematic can be seen the "track mix" stage. For example, track mix PALs U35 and U34 in the upper left corner of sheet 3 of the TRC schematic will take the 4 track data (and 4 track clock) signals from tracks 0, 1, 2, and 3 and provide a 4-way cross-bar switch among these 4 tracks. Each of the 4 outputs of these 2 PALs, TDATA0, TDATA1, TDATA2, and TDATA3 (and associated TCLOCKs) is controlled by 2 mux select bits (MODEA0 and MODEA1 in the case of TDATA0 and TCLOCK0) so that any of the 4 inputs can be selected for any given output.

The track mix stage also provides the injection point for the test signal. The Master Control Card (MCC) produces a simulated PBD signal which is encoded to look exactly like a PBD track output (and associated clock). This test frame signal, TEST DATA, can be inserted in place of the PBD track signals in the track mix stages. The signal MODE-T\, which is controlled by microprocessor U103, is the switching signal that controls the test signal injection. A functional block diagram of the track mix stage is presented in Figure 3 TRC TRACK MIX BLOCK DIAGRAM. The selection of signals to be put into the 4X4 cross bar switches requires some knowledge of the VLBA formatter architecture. Wideband observations require sampling rates of up to 32-Msamp/s. A VLBA recorder has, however, a maximum track data record rate of 8-Mbit/s. Hence, a wideband sampler output must be spread across up to 4 tracks. If 2 bit sampling is being used, the output of a single sampler may be spread across as many as 8 tracks. In processing, these tracks must be brought together into a single TRC delay center (the four delay centers are shown in Figure 3). Each delay center can track a different delay model.

For 2 bit, 32 MHz sampling, the VLBA formatter specification permits the required 8 PBD tracks to be recorded in only 4 ways, across tracks 3, 5, 7, 9, 11, 13 15, and 17, across tracks 2, 4, 6, 8, 10, 12, 14 and 16, across 19, 21, 23, 25, 27, 29, 31, and 33, or across 18, 20, 22, 24, 26, 28, 30, and 32 (track numbers are given in the transport track numbering system). Figure 3 shows how the track

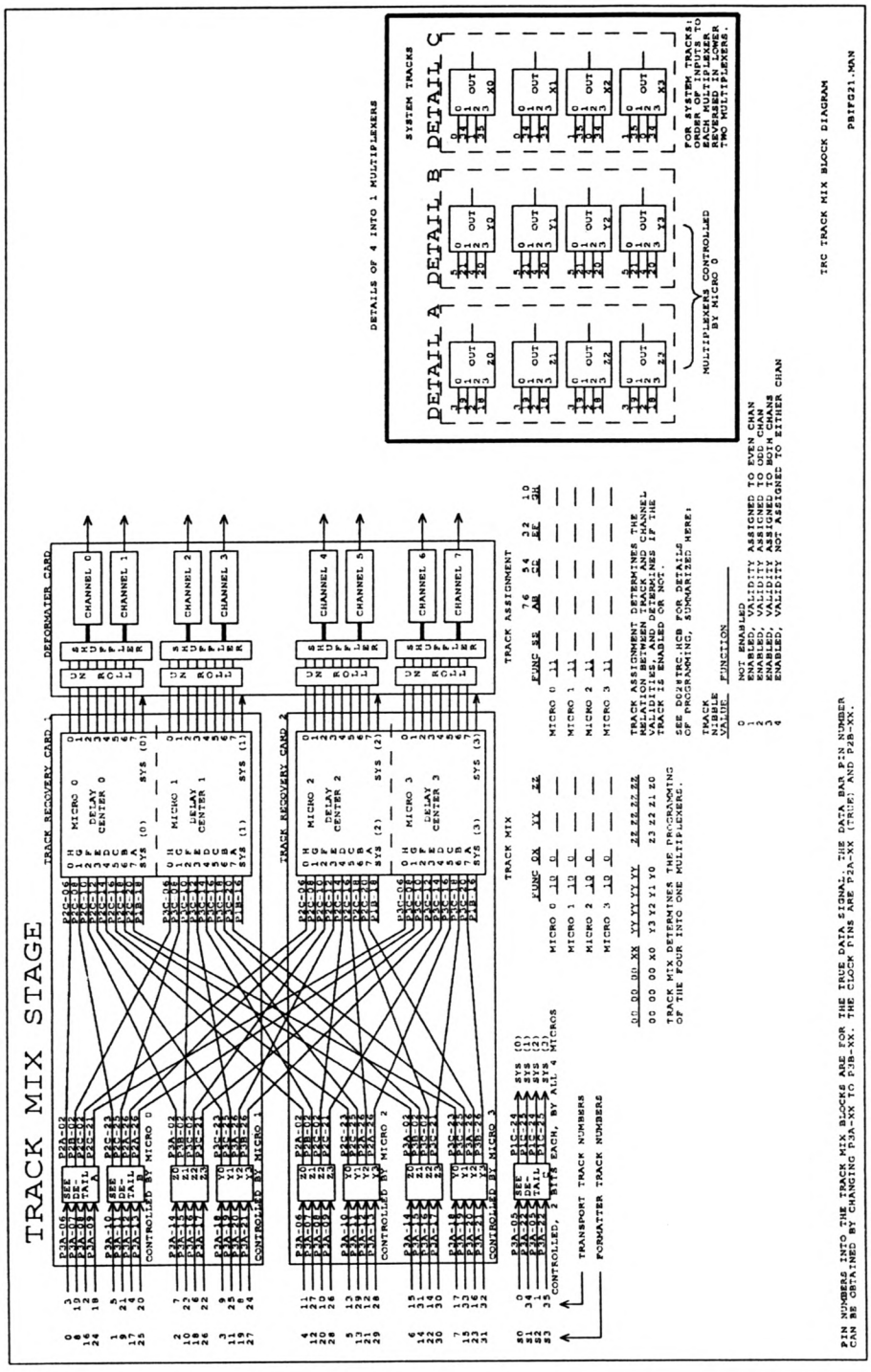


Figure 3 TRC TRACK MIX BLOCK DIAGRAM

signals of a PBD are distributed in a set of 2 TRCs, and also shows the relation between transport track numbers and formatter track numbers. Formatter track numbering is not normally referred to in the PBI system. The square boxes just inside the left hand margin of the track recovery cards (with 4 inputs and 4 outputs) are the 4X4 track mix cross bar switches. As seen, there are 8 such stages across the two TRCs for normal tracks and a 9^{th} (that is distributed across 2 TRCs) for the system tracks (see IC U36 on sheet 2 of the TRC schematic).

The interconnect between the track mix stages and the actual delay center logic involves signals (data and clock) leaving the TRC cards and being wire-wrapped back onto the TRC cards via the bin backplane wiring. Some signals go off a TRC and come right back onto the same card while others are routed to the opposite TRC card. The end product is that whatever the record mode of the recorder, the track mix stage can be programmed to route appropriate signals to appropriate delay center logic blocks.

As an example, use Figure 3 and follow how tracks 3, 5, 7, 9, 11, 13, 15, and 17 can all be patched together into delay center 0 while, say, tracks 2, 4, 6, 8, 10, 12, 14, and 16 are patched into delay center 1.

In cases where fewer than 4 or 8 channels are being processed, the exact same tracks will be routed, via the track mix stage, to more than one TRC delay center. For example, if only one channel is active in a PBD pass and it is recorded on PBD tracks 18, 20, 22, 24, 26, 28, 30, and 32 (the active channel must be the output of a 2 bit 32 MHz sampler since it requires 8 active tracks to record it) then these tracks would be connected through the track-mix stages into all 4 delay centers. (In the 8 FFT engines of the 4 delay centers, the channel signal will be processed so as to enhance the processing sensitivity. For example the transforms can be zero padded and overlapped by a factor of 4 by using the 8 FFT engines. The result would be better sensitivity than if only one FFT engine had been used to process the one channel.)

2.3.3) Track Recovery Logic

Once back on a TRC card, track data and clock signals are fed to an XC-3030 Xilinx chip. Consider, for example, DIN0 and CIN0 on sheet 3 of the TRC schematic driving U95. This Xilinx chip, and the associated 64K X 4 track buffer RAM U96, are all of the logic required to perform the TRCs function on the data of one track. Each of the 36 tracks have a dedicated pair of chips, a "DPC" Xilinx chip (Data Playback Chip) and a RAM buffer. The resulting 18-chips per delay center just about fill the rest of sheet 3 or sheet 4 of the TRC schematic. The only chips remaining on each of these schematic sheets not yet discussed are U86, U85 and U84 (on sheet 3) and U45, U46, and U47 (on sheet 4). Considering the sheet 3 chips, U86 and U85 are Xilinx ICs that provide control logic for the DPC chips of a delay center. U84 is a 10 bit register which provides individual track active signals for the DPC chips. See section 2.5.0 for detailed descriptions of the three Xilinx designs on the TRC card.

2.3.3.0) The Data Playback Chip (DPC)

A block diagram of the DPC chip is seen on sheet 2 of the TRC schematic under the title "TRACK RECOVERY LOGIC BLOCK DIAGRAM". There is one DPC chip for each track data / track clock pair, seen on sheets 3 and 4 of the schematic. The interface to a DPC chip includes:

1) Track data and track clock input lines (nominally at 9 MHz).

2) A 4 bit wide RAM data input/output port to the track buffer RAM (the block diagram shows separate input and output lines but the actual chip has four bi-directional I/O pins).

3) A 16 bit RAM address output bus.

4) Delayed data output line (on a system 8 MHz clock).

5) Header serial data output (to be DMAed into the microprocessor memory).

6) A RAM write signal.

7) A 16 bit RAM address input bus provided by the control Xilinx chips.

8) Various control and monitor lines.

The logic blocks implemented in the DPC Xilinx design include:

1) A 4 bit serial input shift register for the input track data. Track data enter the DPC chip differentially encoded (in Non Return to Zero Mark, NRZM) and the first task of the input logic of the DPC chip is to remove this differential coding. (The logic that removes the NRZM encoding is not shown on the DPC chip block diagram but is logically just before the block labeled DEMOD.)

2) A 4 bit secondary storage register to hold data waiting to be written into the RAM.

3) A real time sync word detector that monitors the incoming track data for occurrences of the frame sync pattern (exact compliance to the sync pattern is required for detection).

4) A frame counter capable of predicting frame boundaries, parity byte boundaries, sync pattern occurrences, etc, once synchronized to the frame by the recognition of a sync pattern by the sync word detector.

5) An 8 bit parity error counter (and secondary storage).

6) A function to demodulate narrow bandwidth data signals (DEMOD). Narrow bandwidth signals must be highly oversampled in the VLBA. The high oversample factor results in track signals being fed to the recorder that do not have

many transitions per unit time. Since the clock is recovered on playback by sensing the flux transitions, a pseudo-random data pattern is EX-ORed with the data at record time to help insure the presence of enough flux transitions for proper clock recovery. This process is called modulation and the demod section removes it at playback time.

7) An asynchronous to synchronous converter that allows write access for the input track data to the RAM.

8) A 16 bit RAM address counter to act as a write address counter.

9) A 16 bit mux to switch the RAM address lines between the write address counter and the external 16 bit read address counter.

10) A 4 bit parallel to serial shift register to provide the delayed data output.

11) A second 4 bit parallel to serial shift register to provide the header output.

12) A vernier delay section to make up delays finer than that provided by the 4 bit RAM addressing.

The Xilinx design for the DPC chip used both of the on-chip clock drivers. The auxiliary clock buffer is driven by the 9 MHz track clock, and is used by items 1 thru 6 above. The global clock buffer is driven by the 8 MHz system clock, and is used by items 8 thru 12 above. Item 7 uses both clocks and resolves timing between the two clocks.

The function of the DPC chip is to write the data bits received from the track output of a PBD into the RAM track buffer. This RAM buffer is central to many of the responsibilities of the TRC card. Some of the buffer capacity is used to absorb the wow and flutter of the track signal. Some of the RAM capacity is used to eliminate the head skew. Some of the RAM is used to set the model delay for the channel as commanded by the DEF card. But mostly the large buffer is used to act as a reservoir, allowing relaxed servo positioning of the PBD by the Deformatter card.

The overall goal of the TRC is to deliver station samples to the correlator precisely timed with respect to the station model, i.e. to output a given sample on a specific clock transition into the correlator, so this sample will be correlated against samples from other stations that reflect the same time epoch.

In order to accomplish this goal, the PBD must be precisely controlled. This control requires an active servo mechanism and having a large sample reservoir in the system makes it easier to provide the servo action. As frames enter the TRC, their headers are recovered, and the time codes of the headers are considered. Having the recovered header times, and knowing the start addresses of headers in the track RAM, the TRC microprocessor uses the model delay provided by the DEF to establish proper read RAM addresses for data coming out of the track RAMs. At any

time, the TRC microprocessor can compare the RAM write and RAM read addresses to measure how full the buffer is. The task of the PBD servo is to control the speed of the PBD so as to keep the track RAM ½ full. When the TRC buffer is less than ½ full, the DEF will command the PBD to speed up and fill the buffer faster. Likewise, when the track buffer is more than ½ full, the DEF will command the PBD to slow down. The larger the buffer provided, the more lax this servo control can be.

Recovery of track data bits from the track RAM is complicated by the use of the RAM for header storage as well as data storage. As frame bits are received from the PBD, data bits (samples) are blocked off in 4 bit nibbles and written into the RAM. Parity bits are stripped off and a parity error count for the data portion of the frame is kept. During reception of the frame header, however, all 180 bits of the header, parity bits and all, are recorded in the RAM.

The header is not written into a special place in the track RAM, the write address counter is a straight 16 bit binary counter and the header bits are written into RAM locations as they come in.

One frame requires 5045 memory storage locations (for VLBA format), 20,000/4 = 5000 locations for the data bits stripped of parity plus 180/4 = 45 locations for the 180 bits of a header. Thus at any one time the 64K X 4 bit track RAM should have stored in it almost 13 complete frames (with up to 13 complete headers scattered throughout the RAM).

Blocking the data bits into 4 bit nibbles, stripping parity bits and counting parity errors, blocking the 180 bits of a header into 4 bit nibbles, establishing header and frame boundaries, and predicting the time of sync word arrival, are all functions of the frame counter and associated logic in the DPC chip. Since the RAM is strictly operated from the system 8 MHz clock, an asynchronous to synchronous conversion stage is needed to provide timely access to the RAM by the input stages of the DPC chip.

The sync detector monitors the incoming track data stream for the 36 bit sync word and when this word is recognized the frame counter is initialized. The sync word is defined in section 3.1 of the Longitudinal Track Format Specification # A56000N003, found in Appendix V. Sync word recognition also causes the RAM write address counter in the DPC chip to be parallel loaded with an address called the "calculated next sync RAM address" supplied by the microprocessor. The last nibble of the sync word is thus written into the RAM in an address that comes from the 16 bit AD[0..15] bus. This bus provides the mechanism whereby the microprocessor establishes time calibration across the track buffer RAM addresses, i.e. by supplying the RAM address into which the sync pattern is written, the microprocessor will subsequently know the RAM address of each data bit of the next frame. Since the frame time code gives the UTC record time for this bit, the microprocessor can use the time code to calibrate the RAM addresses in time.

A software option exists that allows the sync detector either to monitor the input track data for the sync word all the time (wide band sync detection mode) or to limit the search for the sync word to the predicted header time only (narrow band mode). Narrow band sync detect mode makes the DPC chip sensitive to being initialized, by the reception of a sync word, only during the 180 bit wide window of the predicted PBD

header. Use of narrow band sync detect mode prevents spurious detections of the sync word in the data portion of the frame from disrupting the TRC operation.

Narrow band sync detection mode has one pitfall. If some disruption in the PBD signal occurs (such as a tape splice for example), the DPC chip frame counters may get so far off that the sync pulse will not be seen at all in the 180 bit search window. In this situation, the DPC chip for the disrupted track would never recover sync if it remained in narrow band sync detection mode. To correct for this eventuality, a small state machine exists in the DPC logic (see CLB CF and the note just above the title block on the DPC logic diagram K008D06.BLK in Volume 2). In brief, this logic initializes the DPC chip (i.e., it goes to wide band sync detection mode momentarily) if:

1) the sync word is detected in the data portion on a PBD frame, and

2) no sync word is detected in the next header, and

3) a sync word is also detected in the data portion of the very next frame.

If the DPC chip is in sync, the two most likely occurrences of improper operation would be either a failure to detect a sync word because of a bit error in the track playback data, or a spurious sync word detection in the data portion of the frame (also due to playback errors). If the DPC chip is not in sync, however, the operation would result in frequent detection of the sync word in what the DPC chip thinks is the data portion of the frame and no sync detection at all in the header. When this happens, the three state sequence above will quickly occur and the second sync word detection in data will initialize the DPC chip causing it to become synchronized.

The DPC chip also uses the strategy above when in narrow band mode operation to obtain initial frame synchronization when the PBD is first started.

Once the frame counter and RAM write address counter are initialized, they would normally be expected to remain in sync thereafter. The PBD track clock, however, cannot be relied upon to be perfect. Signal drop-outs, caused by tape oxide flaws, can cause the track clock to pick up extra clock transitions or to lose transitions occasionally during a typical tape pass. Without a correcting mechanism for these clock (or sync) slips, loss of sync in the DPC chip would result. The corrective mechanism is the sync word in the PBD track frame. Every time the sync word is detected in the track signal, the frame counter and the RAM write address counters are re-initialized. Thus, if during the course of a frame, the PBD has a data drop-out and either skips or puts out too many clock transitions for the frame, the resulting frame counter error is eliminated when the detection of a subsequent sync word jam sets the frame counter to the proper state. Likewise, if the RAM write address counter gets out of step due to a disruption of the track clock, it will be returned to proper operation by a sync detection.

As mentioned above, the read address counters (there are 2 read address counters, one for the delayed data and one for the header data) are external to the DPC chip. There is a 16 bit RAM address counter for

the delayed data in the Delay Control Xilinx chip and a second 16 bit RAM address counter for header readout in the Header Control Xilinx chip.

The 16 bit RAM address counter in the Header Control Xilinx chip has the secondary function described above to recover from clock slips. It presents a static RAM address (the "calculated next sync RAM address") which is used to initialize the RAM write address counter when a sync pulse is detected. Since header readout occurs only after reception of a header, and since the actual readout takes only a short time, this bus is always available to provide this function by the time the next header comes along.

2.3.3.1) The Delay Control Xilinx Chip

A block diagram of the Delay Control Xilinx chip can be seen on the TRC card schematic, sheet 2. The basic functions of this chip include:

1) Provide the delayed data RAM read address counter.

2) Provide a mechanism that allows the RAM read address counter to "jump over" the headers contained in the track RAM as they are encountered.

3) Provide for the recognition of the start of new frames as the data portion of stored PBD track frames emerge from the RAM into the DEF card.

4) Provide a priority interrupt handler for the microprocessor.

5) Assist in the DMA-ing of headers out of the track RAM and into the microprocessor main memory.

6) Perform header parity error counts and check the time code CRC as the DMA occurs.

The 16 bit read address counter provides a continuous RAM address sequence to all DPC track buffers in a delay center. The counter is, however, parallel loadable. The parallel load function provides the mechanism required to set the delay at the start of each 4.128 msec delay cycle and to "jump over" headers in the track buffer. The counter is output enabled onto the AD[0..15] bus (ADA[0..15] on sheet 3 of the TRC schematic and ADB[0..15] on sheet 4) during the delayed data read cycle (one out of every four 8 MHz clock cycles). At the start of a correlator 4.128 msec new delay cycle, the counter is initialized by parallel loading it with a start address written into a register on the Delay Control chip by the microprocessor. Since the microprocessor knows the time calibration of the track RAM addresses, it can convert the model delay information provided by the DEF card into the proper RAM start address for the new delay cycle. The microprocessor writes the appropriate address into the chip during the 4 msec window allowed by the delay cycle. The NEW DELAY pulse from the DEF does the parallel load at the instant required.

The Delay Control Xilinx has a comparator connected to the output of the read address counter. This comparator monitors the address counter for the start address of the next PBD frame header in the track RAM. When a header start address (written into a register in the Delay Control Xilinx by the microprocessor) is found, the read address counter is parallel loaded with a new address 45 counts ahead (for VLBA format). This action makes the delayed data output from the DPC jump over the header. The DPC will present a continuous data stream to the correlator from the delayed data TRC output. Thus, the non data replacement header is removed from the PBD track signal.

When the TRC card is used in MKIII format, the action is slightly different. In MKIII format the header replaces data on the tape (i.e., samples are lost as header bits write over them in the Formatter). In this case, the jump described above still occurs but only over 5 counts. The 5 address jump is required since the 180 bit header replaces only 160 sampler bits and the extra 20 parity bits fill 20/4 or 5 RAM locations. The other 160 bits of the header are shifted out into the correlator and the resulting pollution of the correlation results are tolerated. An alternate Data Invalid Xilinx personality is available in the Deformatter to gate out the polluted FFT cycles (see section 5.6.2).

When the start of a PBD frame in the track RAM is recognized by the Delay Control comparator, the NEW FRAME logic signal is generated. This signal interrupts the microprocessor and is also used to assert the validity signal which tells the DEF if the new frame just starting is considered valid.

When header data are being DMA-ed into the microprocessor main memory, the Delay Control monitors it in the serial form as it comes out of the DPC chip. The Delay Control logic performs an off-line parity error count on the 20 parity bytes of the headers, checks the sync word position for sync slips, and checks the time CRC for validity. The result of this analysis, done on the headers as the DMA proceeds, is used by the microprocessor to make validity judgments on the frames. When a frame is judged by the microprocessor to be invalid, it sets a flag in memory so that when this frame is output into the DEF, the corresponding validity flag will mark it as invalid. See sections 3.4.0 and 4.2.0 for more details of the validity algorithm.

2.3.3.2) The Header Control Xilinx Chip

The second Xilinx control chip on the TRC card is the Header Control chip. A block diagram of this chip is seen on sheet 2 of the TRC schematic. The basic functions of this chip include:

1) Provide a 16 bit counter to read headers out of the track RAMs. This counter also serves as storage for the "calculated next sync RAM address" discussed above.

2) Provide an off-line fault tolerant sync detector to do after-the-fact clock slip analysis.

3) Provide serial to parallel conversion so the serial header data being extracted from the track RAM can be written into

the microprocessor memory.

4) Provide an 8 bit storage register for frame validity information.

As discussed above there is a 16 bit bus generated in the control Xilinx chips that goes to all 9 DPC chips of a delay center. This bus is labeled ADA[0..15] on the delay center on sheet 3 of the TRC schematic and ADB[0..15] on sheet 4. Each of the two control Xilinx chips have a 3-state source for this bus. The 16 bit source coming out of the Header Control chip has a dual purpose. Most of the time a fixed address sits on this bus which is the proper RAM address into which to write the first nibble following sync detection by the DPC chip (the calculated next sync RAM address). The second function of the Header Control bus is to read the track buffer header contents.

When sync recognition occurs in the DPC chip, the 16 bit write address counter in the DPC chip is jam loaded with this address. The value on the bus acts to maintain the write address counter as a continuous, free running, counter for the duration of the tape pass. This function would not be needed after an initial sync detection if the PBD clock were perfect.

After sync detection has occurred, the address on the 16 bit bus must be updated in anticipation of the next sync word to occur 2.5 msec later. In the mean time, the microprocessor takes advantage of the idle time to initiate the DMA process so it will have the contents of the last set of headers in its memory for analysis.

At this point the 16 bit register in the Header Control chip becomes an active counter instead of a static register. A detailed account of the header DMA is given below.

2.3.3.3) Header Storage in the Microprocessor Memory

As discussed above, when a new frame starts being received by the TRC card it will go thru a sequence that will result in the headers of the PBD frame just received being transferred, via a DMA process, into the microprocessor main memory. Each microprocessor is responsible for the 9 tracks of a delay center although not all of the 9 tracks may be active on any given tape pass.

A TRC microprocessor is notified of the start of a new PBD frame by getting an interrupt on its INT1 interrupt. This interrupt is made the highest priority in software and is actually a summation of 3 hardware interrupt sources. All of the sync word detectors on the active DPC chips of a delay center are logically ORed together, but the DPC chips connected to inactive tracks (as established by the U84 register for the delay center on sheet 3 of the TRC schematic) are excluded from the OR gate. Sync detection by the chips may not all occur at the same time because of the possibility of head skew, but the spread should be small when compared to the response time of the microprocessor so this does not matter.

The presence of several sync detectors in parallel makes the detection of a sync word more robust (i.e. if one track is not error free and the DPC sync detector misses the sync word, another DPC chip can take up the slack). If, however, all active tracks miss the same sync word, the 87C51 microprocessor has a watch dog timer feature to fill in missing sync words. Upon detecting a SYNC interrupt, the microprocessor starts the header input sequence.

The actual hardware control for the DMA process is seen on sheet 2 of the TRC schematic in the block labeled HEADER READOUT SEQUENCER. The first problem to be encountered in the DMA process is that there is only one readout sequencer on a TRC card that will do the DMA work for the 2 microprocessors of the 2 delay centers on the card. To get around this problem requires some microprocessor to microprocessor communication. A single discrete line, with the label HANDSKC, provides the necessary handshake mechanism. A simple micro-to-micro link is reasonable since the two microprocessors should both see sync word recognition at about the same time. A software timeout mechanism exists that has the effect of making handshake failures less disruptive.

The single sequencer has control lines into the DPC and Xilinx control chips and to the DMA address counters (U100 and U61). The sequencer is under the control of microprocessor A (U103). The microprocessor starts the DMA process by putting a logic zero on the lines labeled uRUN. Before this line is activated, both microprocessors must be executing out of their ROM memory since the main memory address source will be taken over by the DMA address generator.

Once the header readout counter is running, the active uRUN signal is removed and the sequencer latches itself on with the S15 signal for one entire header shiftout. The 9 DPC chips of a delay center are tri-stated together and the headers are DMAed into the micro memory one at a time (even inactive tracks go through the motions of header shiftout).

Every time a 180 bit header readout sequence is completed, the sequencer removes the S15 latch on the sequencer run logic, and the sequencer stops. The microprocessor monitors the S15 logic level (on its T0 input) and knows when a header DMA sequence is complete. The microprocessor then reads the header parity count, the CRC result and the off-line sync detector result. Once all tasks for the readout of a DPC header is complete, microprocessor A uses its uRUN line to start up the sequencer to DMA another track header (after a handshake with the other microprocessor).

The sequence for the readout of headers is seen on sheet 2 of the TRC schematic at the bottom-center of the page (labeled HEADER DATA FORMAT SHIFT OUT OF TRACK RECOVERY CHIP). The signals HENA (S0) and HCLK (S1) from the sequencer control the process, and control which of the DPC chips are output enabled for header shiftout. The 9 DPC chips of a delay center form a distributed 9 bit shift register via the pin 34 (input) and pin 35 (output) signal that daisy chains thru the chips (there is a single flip-flop in each DPC chip). A single one is walked down this daisy chain by the HENA and HCLK signals as the 9 headers are sequentially DMAed into the micro memory. The single DPC chip with the logic one in it is the one that is output enabled (on pin 36 of the DPC chip) for header readout at any given time.

The 16 bit counter in the Header Control chip is initialized to the start address in the track RAM, at the beginning of each header readout sequence. All track RAMs are initialized from the same "calculated next sync address" bus and hence a given header starts at the same RAM address in every active DPC chip. During the header readout process, the counter counts thru the 45 RAM locations of a header. The counter is recycled back to the start address for the start of subsequent headers from other DPC chips.

The first 2 bytes DMAed into the microprocessor memory at the start of a header readout sequence are actually not part of a header. The first byte has only 2 valid bits, the "SE" bits. The S bit tells the microprocessor if the DPC chip in question actually detected the sync word for the frame being considered. The E bit is a logic one if the sync word was detected but in the wrong place, i.e. SYNCDET occurred but not simultaneously with the SYNCPRED signal (in which case a sync slip occurred during the frame).

The second non-header byte received during a header DMA is the 8 bit parity error count performed during the frame shift-in. The 8 bit parity error count saturates at a count of 254 (FC in hex) parity errors.

2.3.3.4) Microprocessor to 68000 I/O

The only block of the TRC logic not discussed yet is the communication link with the DEF card. This link is seen on the lower left hand side of the TRC schematic sheet 2. Two NRAO replacement AM2950 chips interface the 68000 microprocessor on the DEF card to the 87C51 microprocessors on the TRC. The AMD AM2950 chip was found to not be available for production purchases, after being designed in to the prototype cards. The NRAO replacement is documented in drawing number L034D01.SCH, in the corrdwgs/hcb/SCCS directory. Each TRC microprocessor has a bi-directional 8 bit bus with DATA READY and DATA TAKEN handshake lines in both directions (a total of 4 handshake lines per microprocessor).

2.4.0) TRC software

Each 87C51 microprocessor on the TRC card has an 8K X 8 main memory RAM. The main program for the TRC microprocessors are downloaded by the RTS (Real Time VME System) into this memory. The 2 microprocessors on a TRC card have identical programs but they have slightly different responsibilities. Micro A (U103) has responsibility for the header readout sequencer, while micro B (U65) will load the Xilinx personalities. Each microprocessor is told it's identity (A or B) by the DEF card when the TRC is reset (the reset comes from the DEF).

Most of the TRC program memory is full as the memory map below illustrates. Only about 500 bytes (broken up into several small segments) are free for subsequent software changes. In addition to the memory allocations below, table storage in pages 3C00, 3D00, 3E00, and 3F00 are required. Some of the modules below, however, are test software

and are not required during observation runs. Thus the memory allocation for TESTOBS, PAINT, and RAMTEST could be made available for more observational software if necessary. (Memory allocations seen below are as of 7 May 1998)

Table 1 TRC MEMORY ALLOCATIONS

Section Name	Starting Address	Ending Address	Mem*
master.obj	0000	00A6	ROM
monitor.obj	00A7	0501	ROM
test.obj	0502	0595	ROM
romtrb.obj	0800	099D	ROM
cover.obj	099E	0B3F	ROM
help.obj	0B40	0F32	ROM
ram.obj	2000	21¹D	RAM
ramtrb.obj	211E	2425	RAM
ramtest.obj	2426	26F0	RAM
obs.obj	2800	2FD2	RAM
testobs.obj	3000	36E8	RAM
time.obj	3800	39A3	RAM
paint.obj	39A4	3C2E	RAM

The source code modules are maintained in the vlbsoft/trcasm area. A Makefile is used to control the building of the trcrom.hex and trcram.hex files. The trcrom.hex file is used to program the 87C51 EPROM, and the trcram.hex file is downloaded over the HCB.

2.4.1) Brief descriptions of the software modules

Below is a brief description of each of the modules listed in the chart above. A more detailed discussion of this software is presented in Chapter 3.

2.4.1.0) MASTER.ASM

MASTER.ASM has the software executed after a hardware reset to the microprocessor. This module will initialize the microprocessor functions like the interrupt logic, the serial port, etc. This module also has the interrupt vectors. After initialization, the microprocessor will execute from an idle loop in MASTER.ASM awaiting instructions from the terminal or from the Deformatter, by way of interrupts.

2.4.1.1) MONITOR.ASM

The monitor software package has various terminal options supported by the TRC microprocessor. All of the monitor functions are in ROM and, hence, are difficult to change. However, there are hooks in place for future additions. The terminal 2 key sets a flag in memory. The terminal 3 key toggles another flag. These flags can be recognized in a RAM based program to provide a new monitor function without a ROM program change.

2.4.1.2) TEST.ASM and RAMTEST.ASM

The TEST.ASM module has a number of hardware test functions that allow an operator to test the TRC card either in PBI Test Fixture # 1, or in the system. Hooks are in place so that software tests can be written in RAM in the RAMTEST module. This capability allows the generation of new tests without changing the TRC ROMs.

2.4.1.3) ROMTRB.ASM and RAMTRB.ASM

ROMTRB.ASM and RAMTRB.ASM handle the microprocessor to microprocessor communications with the DEF 68000. Some of the basic functions, such as memory load, are handled in the ROM based software, but most protocol support is executed out of the RAM based RAMTRB program.

2.4.1.4) COVER.ASM

This module handles the header DMA sequence. Since the DMA process requires control of the microprocessor RAM memory, this software must be run from ROM. The handshaking required so that the header input sequence is coordinated between the two microprocessors occurs in this program.

2.4.1.5) HELP.ASM

This module contains ASCII terminal help screens. MONITOR.ASM has an option to display the help screen. The ROM based HELP.ASM is a large fixed screen (requiring a change to the ROM for modification). A smaller second help screen page is provided in RAM.ASM (see below). Since this page is in ram, it is more easily modified.

2.4.1.6) RAM.ASM

RAM.ASM is mainly used for linking the ROM and RAM based programs of the TRC together. Since there are many TRC cards in the system, it is not desirable to have to change the ROMs on all TRC cards because of a change in a RAM based program. Thus direct linking of the ROM and RAM

based modules is not possible. The strategy used in the TRC software is to always link from the ROM software into the RAM software via jumps to locations that are on page boundaries in RAM modules. These locations will not change as software changes are made in the main bodies of the RAM modules. Thus link addresses in the ROM modules will not change because of changes in a RAM subroutine.

2.4.1.7) OBS.ASM

OBS.ASM is the main observing software for the TRC microprocessor. This module is entered upon reception of an OBSERVE command from the DEF. The first action of the software is to configure the TRC card for the observation using parameters previously received from the DEF.

Once the TRC is configured, the TRC microprocessor starts the observation by enabling interrupts. Most of the actions required to support the observation are executed out of interrupt handlers. The interrupt system is described in more detail below. When not supporting an observation in an interrupt routine, the microprocessor waits in an idle loop in OBS.ASM. There are 2 functions supported in OBS.ASM not in interrupt handlers, the terminal readout function (in PAINT.ASM) and a program called autotest.

Autotest is a program that allows the TRC to perform a stand alone observation. An observation usually requires active support by the DEF card microprocessor with the DEF supplying a delay model to be tracked. Autotest supplies software that allows the TRC to track a nominal linear model itself and hence allows the TRC to observe in stand alone mode. Autotest would normally be used in PBI Test Fixture # 1.

2.4.1.8) TESTOBS.ASM

The TESTOBS.ASM module is mainly used in PBI Test Fixture # 1 for testing either the TRC or DEF cards. This program could be separated from the main body of the TRC program and loaded only when necessary, but as long as there is room for it in the RAM it should be left there. It can be run in the system, and has some value in system trouble shooting since it gives terminal header printouts not supplied by OBS.ASM.

2.4.1.9) TIME.ASM

TIME.ASM logically belongs in OBS.ASM but was made a separate module because OBS.ASM overflowed a page boundary.

2.4.1.10) PAINT.ASM

This module supports 2 types of terminal screen displays during an observation. One screen prints out the system status (observational parameters, recovered time, track parity errors, etc.) and is started via the monitor S command. This display is recurrent but the continually updating display can be stopped by typing another S command. The other

screen can be started by using the TF x2 monitor command and results in 20 consecutive headers from track x being displayed on the terminal.

2.4.2) Microprocessor Interrupt Structure

Almost all of the functions provided by the TRC software are accomplished in interrupt routines. There are several interrupts and a summary is given below:

Table 2 TRC INTERRUPTS

interrupt	function	priority
SERIAL	terminal	low
INT0	DEF communication	medium
INT1	NEW FRAME	high
INT1	SYNC	high
INT1	NEW DELAY	high
TIMER0	watch dog timer	high
TIMER1	not used	-

The serial port interrupt is not normally used during an observation. It supports a terminal and provides the functionality of the MONITOR.ASM software.

The INT0 hardware interrupt is used for TRC-DEF communications with software support provided by the ROMTRB.ASM and RAM.ASM software.

The INT1 hardware interrupt is the summation of three functions. Logic in the Delay Control Xilinx chip supports the summation and the microprocessor polls this chip to find which of the 3 sources is responsible for an interrupt. The three sources of the INT1 interrupts are given below:

1) NEW FRAME: Each time the read address comparator in a Delay Control Xilinx chip sees the start address of a new frame being output to the Data Playback chips (DPCs), it interrupts the associated TRC microprocessor. The microprocessor's response will be to set up the next new frame start address for the comparator. The microprocessor also puts the channel validity bits for this next new frame into secondary storage in the Header Control Xilinx chip, to await detection of the new frame by the comparator.

2) SYNC: Whenever an active DPC chip detects the sync word in PBD track data, an interrupt to the TRC microprocessor is generated. When this happens, the TRC microprocessor must execute a long subroutine that will:

a) DMA header contents from the track RAMs into it's main memory,

b) analyze the contents of the headers to ascertain the status of the tracks (valid or invalid),

c) ascertain and track the header time codes, and

d) do general house keeping functions in support of the PBD data recovery.

After completion of the DMA and analysis of track headers described above, the SYNC interrupt routine writes the "calculated next sync RAM address" in the Header Control chip so it will be ready for the next sync word detection by the DPC chips. The SYNC interrupt routine is by far the most complicated and time consuming function performed by the TRC microprocessor. It takes about one half of the execution cycles available to the microprocessor.

3) NEW DELAY: Every 4.128 msec a new delay model is computed by the hardware model generator on the FFT Control Card (FCC). The integer part of the new delay value is transmitted to the DEF card which, after considerable computation, gives the TRC microprocessor a 16 bit delay code. This 16 bit delay code identifies the bit required by the FCC delay model to be output by the TRC during the first bit of the next delay cycle. The delay code is relative to the last time reference requested by the DEF from the TRC. The DEF converts the time code and calculates the number of bits between this time and the FCC requested bit. The TRC has the RAM buffer address that corresponds to the time code and can thus calculate the RAM address of the desired FCC bit. The TRC microprocessor must perform housekeeping duties in support of setting the delay every time it sees a NEW DELAY interrupt.

The TIMER0 interrupt provides a watch dog timer function to guard against missing sync words. If it were not for this timer, the microprocessor would miss every frame in which no active track saw an error free sync word. The timer is set long every time a sync word is detected so that when it does expire the microprocessor can be certain that the frame boundary was missed. The use of the timer is a software option conveyed to the TRC microprocessor in the observation parameters from the RTS.

2.5.0) Xilinx Designs Used in the TRC

There are three designs made with Xilinx chips on the TRC, the DPC design (see section 2.3.3.0), the Delay Control design (see section 2.3.3.1), and the Header Control design (see section 2.3.3.2). The logic diagrams for the three designs are listed below:

DPC	K008Dxx.BLK (10 sheets)
Delay Control	K009Dxx.BLK (10 sheets)
Header Control	K010Dxx.BLK (9 sheets)

and can be found in Volume 2.

The discussions below will describe these logic diagrams. All diagrams were made using "home made" logic symbols, but the schematics

should still be followable with some effort. Each Xilinx CLB (configurable logic block) is shown in the logic diagrams as a box with two flip-flops just to the right (along with the various programmable multiplexers in the CLB). The box represents a Xilinx PLA with 5 external inputs, two feedback terms and two outputs. The heavy lines inside the boxes indicate which logic terms are included in the equation for a given PLA output. The specific equation can be obtained from the .MAC file used for the design or the Xilinx .LCA file. See Appendix I, Section I.1.5 for a list of the Xilinx design files.

In looking at these schematics, note that the CLB designations, and the floor plans agree with the net list compiled for the design (the .MAC file), but do not agree with the actual logic placements used in the design. The Xilinx autorouter moved most or all CLB at layout time.

2.5.1) The DPC Xilinx Design

See K008D01.BLK through K008D10.BLK in Volume 2 for the logic diagram sheets referred to in this section.

Sheet 1 of the DPC logic diagram shows the floor plan of the chip. The design uses a XC3030 (10-CLB by 10-CLB) chip. As can be seen, there are a reasonable number of spare CLBs for future logic changes.

Sheet 2 shows the chip input circuitry and the DEMOD data generator. Pin P15 is the input for track data to the DPC chip. CLB BA undoes the NRZM coding of the track data (see section 2.3.3.0). The flip flops of P16 and P17 provide the 4 bit serial input register seen in the block diagram on L007D02.SCH.

CLBs GA, GB, GC, GD, GE, GF, GG, and GH are used as a feedback shift register pseudo random data generator to generate the demod pattern (See specification A56000N003 in Appendix V). Signal SPARE0 on pin P18 is a discrete controlled by microprocessor A on the TRC, that enables or disables the DEMOD function. The actual DEMOD process occurs in CLB BA.

Sheet 3 shows the output logic. CLB AA, AB, and one flip-flop of pin P11, are the 4 bit parallel-in, serial-out shift register that produces the header data output. P11, CLB AA (note dual functions), and CLB AD are the 4 bit parallel-in, serial-out shift register that produces the delayed data output. CLB AC is the divide by 4 used for the 4-in-1 parallel strobe. The divide by 4 is synchronized to the system by the HCLK signal. CLBs AE and AF give a programmable 0, 1, 2, or 3 bit vernier delay for the delay model. (Most of the delay model is generated by calculating addresses in the track buffer RAM. Since the RAM is 4 bits wide, however, this strategy yields delay resolutions to the 4 bit level. Finer resolution down to the track bit level requires this vernier delay.)

Four of the I/O pin flip-flops (P8, P9, P10, and P11) provide the data input 4 bit secondary storage register and hence is part of the input logic. After 4 track bits are shifted into the 4 bit register seen on sheet 2 of the logic diagram, they are strobed into these 4 flip-flops and wait for access to the track buffer RAM when their contents are written into the RAM.

Sheet 4 of the DPC logic diagram shows the input PBD frame counter. CLBs JB and JC perform a divide by 4/divide by 8/divide by 9 function (see the state tables at the bottom of the sheet). CLBs IB, IC, ID, IE, IF, and IG perform a divide by 2508/divide by 13/divide by 8 (or a divide by 2488/divide by 13/divide by 8 for MKIII) function. The divide by 8 comes from the number of 9 bit bytes between the end of sync pulse and the start of the data portion of the frame. The divide by 2508 results from the number of 9 bit bytes between the sync pulse and the start of the next header. The divide by 13 results from the number of 9 bit bytes between the start of a header and the end of the sync pulse. CLBs HA, HB, HC, HD, and HE decode the states in the frame counter, and set the proper radix for the appropriate part of the frame. The header delimiter signal, H/D, and the SYNCPRED time are decoded in this logic.

Sheet 5 shows a timing diagram of the frame counter. The counter is reset to 0 when a sync pulse is detected in the track signal (SYNCDET and SDET). Count 8 is decoded to signal the end of the header (signal H/D or S3). Count 2508 (for VLBA) is decoded to signal the start of the next header (and the counter is reset). Finally, count 13 (during the header) is decoded as the sync predict signal.

Sheet 6 of the DPC logic diagram shows the sync detector. CLBs BB, BC, and BD construct a state machine which progresses through states as the track data allows and reaches the final state (RAWSYNC high) only after reception of the sync word. CLBs BE, CF, and CG implement the wide band/narrow band sync detection mode operation. The signal on pin P72 is from the microprocessor on the TRC, and gives the programmed sync mode (logic one for wide band sync detect mode, logic zero for narrow band mode).

If the DPC is operating in narrow band sync detection mode, CLBs CF and CG provide for initial sync-up (or re-sync-up in the case of a tape splice or other disruption). Narrow band mode means that the sync detector is sensitive to detecting the sync word only during the header. This operation reduces the chance of spurious detections of the sync word by limiting the sync search to small intervals of time (the 180 bit headers). On initial sync up or after a tape splice, however, the problem is that the frame boundaries are not reliably known. Hence the sync detector does not know where to look. CLBs CF and CG look for the sequence:

1) sync detected in the data portion of a putative frame
2) sync not found in the next header portion of the putative frame
3) sync detected a second time in the data portion of the next frame

If this sequence is observed to occur in two adjacent frames (and the header interval between), the second sync detection (in what is thought to be data) is allowed to reset the DPC frame counter. Once reset, the frame counters should accurately predict the header boundaries, and normal narrow band operation is resumed (see section 2.3.3.0).

Sheet 7 of the DPC logic diagram shows the parity error counter. CLB DB does the parity error check of the track data (TDATA0/). When a parity error is found, it clocks the 8 bit ripple-thru counter formed by CLBs DC, DD, DE, and DF (this counter stops, or saturates, at a count of 254). CLBs CC, CD, CE, and CF are secondary storage for the parity error count. At the end of the data portion of a frame, the parity error count is strobed into the secondary storage register, and the counter is reset for the next frame. During header readout, the secondary storage is converted into a shift register, and the parity error count is shifted out to the microprocessor (the 8 bit parity error count is inserted in front of the header data during serial header shiftout). CLB CG provides control logic.

Sheets 8 and 9 show the track buffer RAM write address counter and RAM address mux schematics. The counter is externally loadable. When a SYNCDET occurs, the "calculated next sync RAM address" on the AD[0..15] bus is jam loaded into the counter. The mux stage switches the RAM address lines between the counter output and the AD[0..15] bus. The AD[0..15] bus is also used to provide the track buffer RAM with read addresses during delay data readout and during header readout. The counter is a synchronous counter that increments every time the RAM is written into.

Sheet 10 shows the control logic. CLBs EA, EB, EC, EE, and ED are the asynchronous to synchronous (or track clock to system clock) stage. When 4 bits of the input track signal is put into secondary storage (see sheet 3 I/O pins P8, P9, P10, and P11), this logic stores the fact. The logic then waits for the next opportunity to generate RAMWE, DATAOE, CLKADDR, and EXT LOAD signals to get the data written into the track RAM and to increment the RAM write address counter.

CLBs FC and FD are in the output path for the header data. They allow the sync received and sync error (the "SE" bits, see section 2.3.3.3) to be appended onto the serial parity error/header data output.

2.5.2) The Delay Control Xilinx Design

See K009D01.BLK through K009D10.BLK in Volume 2 for the logic diagram sheets referred to in this section.

Sheet 1 of the Delay Control logic diagram shows the floor plan of the chip. The design uses an XC2030 (8-CLB by 8-CLB) chip.

Sheet 2 shows the microprocessor data bus interface with the chip. The flip-flops associated with the 8-input pins of the data bus are used as an 8 bit register, at microprocessor address 6020 (see the description of sheet 6). The storage in the flip-flops of pins P17 and P24 are also available to the TRC microprocessor for storage, at microprocessor address 6026 (see the description of sheet 5).

Sheets 3 and 4 show the 16 bit Delay Control counter. This counter counts as the delayed data are output from the DPC track buffer RAMs. The track buffer RAM address provided by the counter is the same for all DPC chips of a delay center and is conveyed to the DPC chips on the 16 bit external address bus that drives every DPC chip. The counter free runs except for when it is jam loaded. The jam load occurs in two instances.

The first is when the delay changes every NEW DELAY cycle (the DEF supplied model delay is implemented in the TRC by manipulating the track buffer RAM addresses). The second is when a header is encountered in the RAM buffer (the counter is then jam loaded with a microprocessor supplied address just past the header, see the discussion of sheet 6).

Sheet 5 of the Delay Control logic diagram shows the comparator that monitors the delayed data read address counter (sheets 3 and 4) for the start of headers in the track buffers. When a sync pulse is detected in the track data, the write address counters in all of the DPC chips are parallel loaded from the "calculated next sync address" bus. Since this address is supplied by the microprocessor, it has a knowledge of the header boundaries in the track RAMs. The microprocessor also supplies the start address for the read address counter of sheets 3 and 4 at the beginning of every NEW DELAY cycle. Thus the microprocessor knows the general activity of this counter at all times.

The microprocessor will write the address of the last byte of the frame, currently being addressed by the 16 bit counter of sheets 3 and 4, into the 16 bit register formed by the 16 flip-flops of CLBs AG, BG, CG, DG, EG, FG, GG, AND HG (microprocessor address 6022). The comparator of sheet 5 will compare the contents of this register to the 16 bit read address counter. When the read address matches the register contents, the address counter is jam loaded with an address written into the 16-bit register seen on sheet 10. This register gives the start RAM address of the data portion of the next frame in the track buffer RAM. This action causes the read addresses on the TRC track buffer RAMs to jump over the headers. This heals the "wound" in the PBD track data caused by the removal of the header from the delayed data output because of the non-replacement frame format.

The comparator output comes from CLB AF, output X, seen on sheet 3 of the schematic. This signal, NEW FRAME, also feeds the DEF so that this card will know precise frame boundaries for its track roll and data validity functions. The signal MODE0 (supplied as a data bit by the TRC microprocessor when it resets the NEW FRAME interrupt) can disable the comparator. This disable capability is required because when the microprocessor writes a new address to the comparator it takes two writes for the 8 bit micro to write the new 16 bit address. The comparator is disabled during this process to prevent inadvertent NEW FRAME detections from occurring during the re-load process.

Sheet 6 of the Delay Control logic diagram shows the mux that multiplexes the two jam load addresses for the read address counter. As discussed above, the counter can be jam loaded with either the initial NEW DELAY address or the header skip address. The initial NEW DELAY address is stored by the microprocessor in the 8 bit register seen on sheet 2 of the schematic (DA[0..7]) and in 8 of the flip-flops of CLBs AE, BE, CE, DE, EE, FE, GE, and HE on sheet 6. The skip address, DB[0..15], comes from the 16 bit register seen on sheet 10 (microprocessor address 6021).

Sheet 7 shows the control logic for the chip. CLB HA is a divide by 4 element that generates the one in four track RAM cycle. The CLB HA counter is synchronized to the system by the NEW DELAY signal. CLBs GB and HB decode the divide by 4 and provide control signals in this cycle.

The SELx lines and the outputs of CLBs BB and CB are microprocessor address select lines for memory mapping the various resources in the chip into the micro address space.

Sheet 8 shows the off-line header parity error counter and the interrupt logic. CLBs DA, BA, and CA count parity errors in the header data as it is DMAed into the TRC microprocessor memory. CLB DA is the parity checker. CLBs BA and CA form a 3 bit counter to count the errors detected by CLB DA (the count saturates at 7). The final count is read by the microprocessor (via the data I/O pins seen on sheet 2) after each header DMA is complete.

CLB GA logically ORs the three sources of interrupts for the TRC microprocessor together and provides the interrupt. The individual interrupts are latched into CLBs FA for a NEW DELAY interrupt, FB for a RECEIVED SYNC interrupt, and EA for a NEW FRAME interrupt. After an interrupt, the TRC microprocessor will read the chip (via the I/O pins of sheet 2) to find the source of the interrupt. The interrupts have individual memory mapped resets.

Sheet 9 of the Delay Control logic diagram shows the CRC checker. As headers are shifted out of storage in the track buffer RAMs and DMAed into the microprocessor memory, this circuit, under control of the TRC header readout sequencer (see section 2.3.3.3), checks the CRC encoded in the header for validity. The result of the check, the signal ECC, is read by the microprocessor after the DMA is complete.

Sheet 10 shows two 16 bit storage registers that are memory mapped into the TRC microprocessor memory space. The 16 bit register formed by the flip-flops of the chip I/O pins are used to store the program bits for the TRC track mix stages. The register formed by the eight CLBs on sheet 10 were discussed in the description of the mux on sheet 6.

2.5.3) The Header Control Xilinx Design

See K010D01.BLK through K010D09.BLK in Volume 2 for the logic diagram sheets referred to in this section.

Sheet 1 of the Header Control logic diagram gives the floor plan of the chip. The design uses an XC2030 (8-CLB by 8-CLB) chip.

Sheet 2 shows the microprocessor data bus interface with the Header Control chip (pins P15, P16, P18, P20, P23, P25. P27, and P29). The flip-flops associated with these 8 input pins are used as an 8 bit register (microprocessor address 6031). The microprocessor writes validity bits to this register which are subsequently clocked to the TRC card validity output pins by the NEW FRAME signal (from the DELAY CONTROL chip). Thus when the delayed data output from a DPC chip starts into a new frame, the validity of the data in the new frame, as prescribed by the TRC microprocessor, is strobed into the DEF coincident with the first bit of the new frame.

I/O pins P11, P10, P9 and P8 provide a 4 bit register that the TRC microprocessor can write to. The program bits for the track mix stage of the system track are stored in this register (microprocessor address 6033).

Sheets 3 and 4 show the 16 bit counter used for the readout of headers from the TRC track buffers. The counter also serves as a static 16 bit register for the "calculated next sync address" when header readout is not in progress. When the logic is used as a counter, it is parallel loaded at the beginning of each header DMA transfer with the header start address on bus DB[0..15].

Sheets 5, 6, and 7 of the Header Control logic diagram show the off-line sync detector circuit. As the headers are shifted out of the TRC track buffers, this circuit provides a 36 bit template that searches for the sync pattern in the header data. This sync detector is fault tolerant and will still recognize the sync pattern if it has as many as 2 errors. The logic on sheet 5 shows 28 bits of a 36 bit shift register (the microprocessor data I/O pins of sheet 2 provide the other 8 bits). The CLBs of sheets 6 and 7 sum up all of the deviations of the 36 bit register output from the sync pattern (the sync pattern is implicit in the CLB equations). The output of CLB ED is high if the sum of the deviations of the 36 bit word being scrutinized from the 36- bit sync pattern is less than or equal to 2.

Sheet 8 shows the control logic for the chip. The SELx lines at the outputs of CLBs GD, HD, and FD are microprocessor address select lines for memory mapping the various resources in the chip into the microprocessor address space.

CLBs AD and AE provide the vernier delay (see the DPC chip sheet 3 discussion). CLB AD provides storage for the vernier delay written by the TRC microprocessor. CLB AE provides secondary storage so the delay can be made effective on the NEW DELAY boundary. CLBs GE and HE provide the vernier delay for the NEW FRAME.

CLB FE is a divide by 4 element that generates the one clock in four track RAM cycle (the NEW DELAY signal synchronizes the divider). CLB EE decodes the divide by 4 and provides control signals within this cycle. (Note that all three Xilinx designs on the TRC have a divide by 4 stage, CLB AC in the DPC chip, CLB HA in the DELAY CONTROL chip and FE in this chip. These stages must be synchronized together so that the track buffer memory cycle will be known by all of these chips.)

CLB DD provides storage for the result of the fault tolerant sync detector. Two bits are stored, the offline "SE" bits (S indicates if sync was detected at all during the header shiftout and E indicates if the detection was coincident with the S11 sync predict signal). The SE bits are read by the microprocessor via the CLB DE MONTOR line, which drives the TRC microprocessor T1 input.

Sheet 9 shows a 16 bit register that the TRC microprocessor can write to (see the discussion of sheet 3 and 4).

2.6.0) Utility Displays

There are two terminal based utilities commonly used while the TRC is observing. The first is the STATUS screen. The second is a display of the contents of 20 consecutive (in time) track headers for a specified track.

2.6.1) Status Display

The "s" key toggles the status display on and off. An example of the display is:

Table 3 TRC STATUS DISPLAY

```
TRACK RECOVERY 2-08-0A>s

10 00 00 00
11 00 33 33 33 33
12 1F 10 0C 10 10 00 10 10
13 00 25

00 00 00 00 70 06 50  TIME GOOD

AA (=80 FOR 1/2 FULL)

FF 00 FF 00
00 00 00 00 00

0000 00 00
0000 00 00
0000 00 00
0000 00 00
0000 00 00
0000 00 00
0000 00 00
0000 00 00
0000 00 00
```

The fields in the above display are described in the PAINT.ASM source code module as follows:

```
*     10 0X YY ZZ                  ;TRACK MIX PARAMETERS
*     11 0S AB CD EF GH            ;TRACK ASSIGNMENT PARAMETERS
*     12 P1 P2 P3 P4 P5 P6 P7 P8   ;OBSERVATION PARAMETERS
*     13 T1 T2                     ;FRAME DURATION PARAMETERS
*
*     00 JJ JS SS SS ss ss         ;FILTERED TIME
*
*     FF                           ;TRACK RAM FULLNESS BYTE
*
*     N1 N2 N3 N4                  ;ERRORS     (51,52,53,54)
*
*     TT AA I0 I1 TO               ;ERRORS     (3D70,71,72,73,74)
*
*     PPPP HH RS                   ;TRACK 0 ERRORS
*     PPPP HH RS                   ;TRACK 1 ERRORS
*     PPPP HH RS                   ;TRACK 2 ERRORS
*     PPPP HH RS                   ;TRACK 3 ERRORS
*     PPPP HH RS                   ;TRACK 4 ERRORS
*     PPPP HH RS                   ;TRACK 5 ERRORS
```

```
*      PPPP HH RS                    ;TRACK 6 ERRORS
*      PPPP HH RS                    ;TRACK 7 ERRORS
*      PPPP HH RS                    ;TRACK S ERRORS
*
*WHERE;
*
*      N1   = COUNT OF FRAMES WITH ERROR FREE HEADERS (SAT AT 255)
*      N2   = COUNT OF FRAMES WITH NO ERROR FREE HEADERS (SAT AT 255)
*      N3   = COUNT OF FRAMES WITH GOOD TIME (SATURATES AT 255)
*      N4   = COUNT OF FRAMES WITHOUT GOOD TIME (SATURATES AT 255)
*      TT   = NUMBER OF TIMES FILTERED TIME HAS BEEN RESET
*      AA   = NUMBER OF TIMES THE HEADER READ WAS ABORTED
*      I0   = NUMBER OF TIMES CHANNEL 0 WAS INVALID
*      I1   = NUMBER OF TIMES CHANNEL 0 WAS INVALID
*      TO   = NUMBER OF TIMES SYNC TIMER TIMED OUT
*      PPPP = RUNNING SUM OF TRACK PARITY ERRORS
*      HH   = RUNNING SUM OF TRACK HEADERS WITH ERRORS
*      RS   = RUNNING SUM OF TRACK RE-SYNCS
```

2.6.2) Consecutive Header display

This display provides the header contents from 20 consecutive (in time) headers from a specified track. The display is invoked with the TF x2 command, where x is 0-7 for the delay center input number. The TIME/CRCC data is displayed so that it lines up with the AUX DATA (both sets of data associated with the same tape frame). This is mentioned because the block of data that the TRC treats as one header actually consists of the AUX DATA from one frame and the TIME/CRCC data from the following frame. The SYNC word separates these two blocks of data.

An example of the tf x2 display is:

Table 4 TRC CONSECUTIVE HEADER DISPLAY

```
TRACK RECOVERY 2-08-0A>tf 02

  ROLL  SE PARITY      AUX DATA              TIME/CRCC
   00   10   00   12 34 56 78 9A BC DE AD
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 54 00  EB AE
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 54 25  6B 73
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 54 50  EA 4E
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 54 75  6A 93
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 55 00  6D AD
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 55 25  ED 70
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 55 50  6C 4D
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 55 75  EC 90
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 56 00  67 AD
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 56 25  E7 70
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 56 50  66 4D
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 56 75  E6 90
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 57 00  E1 AE
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 57 25  61 73
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 57 50  E0 4E
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 57 75  60 93
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 58 00  C3 AE
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 58 25  43 73
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 58 50  C2 4E
                                            00 00 01 62 58 75  42 93
```

This example is taken from a test fixture, so the AUX DATA field contents and frame times are not typical of data from real observations. Also, barrel roll was not operating. If roll is on, the ROLL field will show the roll phase associated with the time code. Thus the ROLL field will count 0-7 or 0-15 (modulo 8 or modulo 16) in ascending sequence down the column.

The SE field is the SYNC word status. S = 1 means sync was detected. E = 0 means sync was in the expected position (if detected). The PARITY field is the frame data parity (not the header parity).

The AUX DATA and TIME/CRCC fields are identified as follows:

```
  ROLL  SE PARITY      AUX DATA              TIME/CRCC
  ----  -- ------ ----------------------   ------------------------
   00   10   00   12 34 56 78 9A BC DE AD   00 00 01 62 54 00  EB AE
                  SS RH PP PP TG BS CD 00   JJ JS SS SS.ss ss  CR CC
```

where the first four AUX DATA bytes are BCD digits as follows:

SS is a station identifier 0-99
R is recorder number 0 or 1
H is headstack number 0

PP PP is head position; first P is sign, 0 for positive, 8 for negative
next three P are BCD 0-999 microns

and the last four AUX DATA bytes are "bit reversed" BCD digits as follows:

T is formatter track number 0-7
G is formatter track group 0-4
B is bit number, 0 = LS bit, 1 = MS bit
S is sideband number, 0 = lower, 1 = upper
C is baseband converter number 0-7
D is Data Acquisition Rack number 0-1
00 is the spare byte which must be restricted to BCD values

In each "bit reversed" nibble, the four bits represented by the nibble need to be reversed end to end in order to decode the original value represented by the four bits. The following table shows the bit reversals for all BCD values:

Table 5 TAPE HEADER BIT REVERSAL

BCD value	BCD bits	Reversed Bits	Reversed Nibble
0	0000	0000	0
1	0001	1000	8
2	0010	0100	4
3	0011	1100	C
4	0100	0010	2
5	0101	1010	A
6	0110	0110	6
7	0111	1110	E
8	1000	0001	1
9	1001	1001	9

From the above table, a nibble value read from the tf 02 display would be looked up in the Reversed Nibble column, and the actual value represented would be found in the same row in the BCD value column. For example, in the baseband converter nibble, a value of C on the screen would translate to baseband converter number 3.

The TIME/CRCC bytes are defined as follows:

J JJ is the MJD (three least significant digits of the Modified Julian Date)
S SS SS is integer seconds (0 - 86,399)
ss ss is fractional seconds to a precision of 0.1 msec.

CR CC is a "CRC-16" cyclic redundancy code computed on the the data bits only, of the first six time code bytes; (these two bytes are not BCD)

The details given above are taken from NRAO Specification A56000N003, VLBA Longitudinal Track Format, found in Appendix V.

2.7.0) TRC Drawings

A complete list of files in every directory below vlbsoft and corrdwgs that is used to maintain TRC related files is given in Appendix I. Figure 4 TRC FILE SUMMARY tabulates the main files.

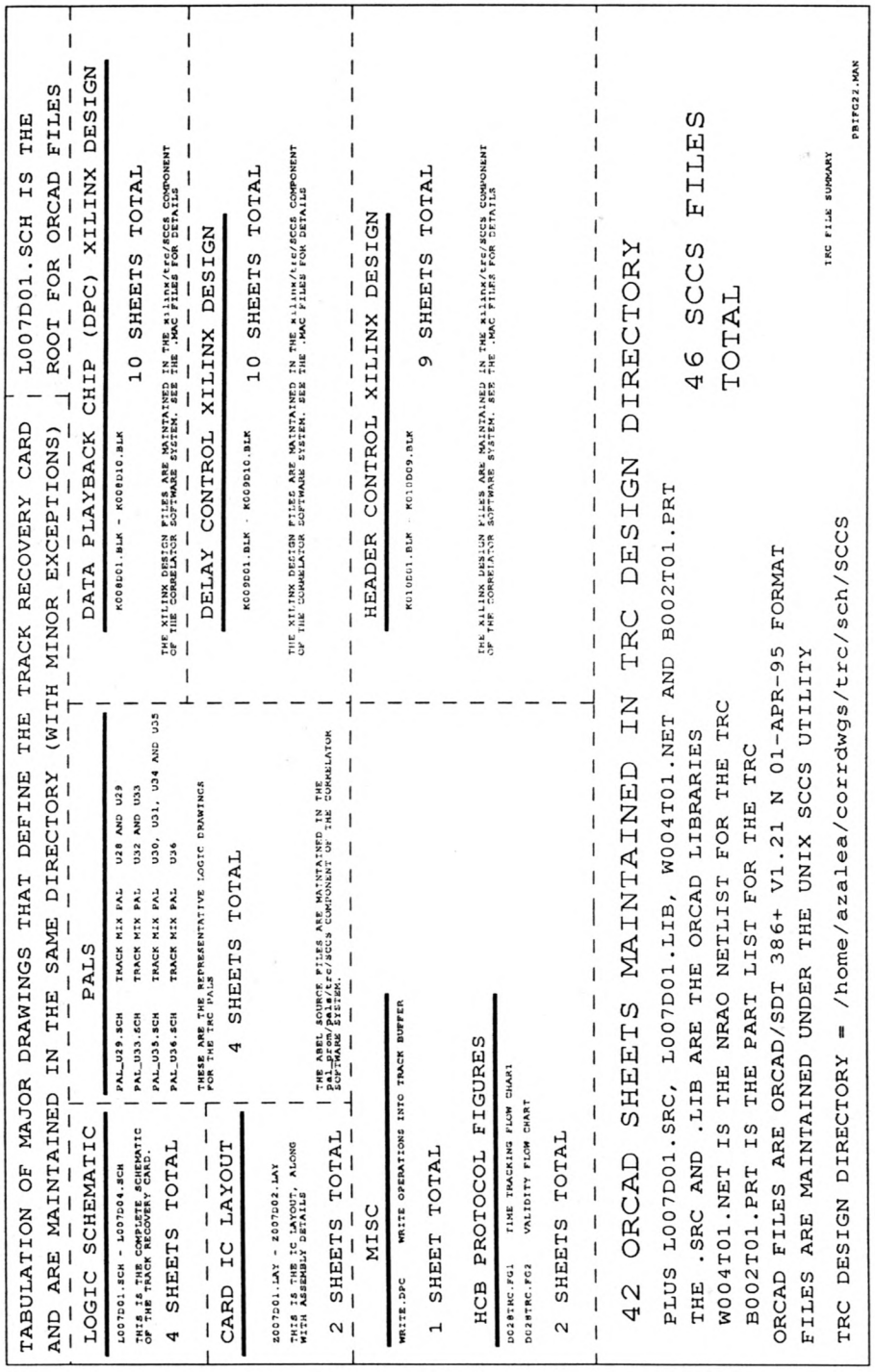


Figure 4 TRC FILE SUMMARY

Chapter 3 The Track Recovery Card Software

3.1.0) Introduction

The Track Recovery Card (TRC) is used to interface a Playback Drive (PBD) to the VLBA correlator. This chapter will describe the software provided for the 87C51 microprocessors on this card.

Two TRCs are required to recover all 36 tracks of a VLBA PBD. Each card has two microprocessors. Each microprocessor controls the recovery of 9 PBD tracks (eight data tracks and one system track). This group of tracks (we will refer to it as a group of 8 tracks since the system track is seldom used) represent a unit called a delay center in the VLBA correlator. Every channel carried by the 8 tracks of a delay center must have the same delay. (In VLBA terminology, a channel represents the output of one of a maximum of 8 samplers that one station of the VLBA may support.) The number of channels that can be supported by the 8 tracks of a delay center depends on the formatter mode and can be either 1 or 2. (The 8 tracks may contain other channels but the delay center in question can process at most 2 of them.) Thus the 4 delay centers of up to 2 channels each yields the 8 channel per station specification of the VLBA correlator.

A switching stage in front of each group of 8 tracks allows a selection of which tracks will be recovered in each microprocessor. If only 8 tracks are active on a given pass, for example, all 4 microprocessors can recover the same 8 tracks but provide 4 independent delay/fringe models and observe 4 different delay centers at the same time.

3.2.0) Microprocessor Program

The TRC uses an 87C51 microprocessor with a 16 MHz clock to control the recovery of PBD tracks. The micro performs all of its assigned tasks in interrupt routines. These interrupt routines include (in order of priority):

1) Track Sync pulse detected
2) New Frame
3) New Delay
4) DEF communication
5) Local terminal RS232 communication.

Interrupts 1, 2, and 3 are ORed together into the 87C51 INT1 interrupt and are prioritized only in the sense that when the INT1 interrupt is received, the three are polled in the order 1, 2, 3. (A timer interrupt in the 87C51 can be used as a sync detection watch dog timer. The

resulting interrupt can be thought of as being in parallel with the Track Sync interrupt.)

The DEF communication uses the INT0 87C51 interrupt and is of lower priority than the INT1 interrupt. A DEF interrupt in progress can be interrupted by a Sync/New Frame/New Delay or watch dog timer interrupt.

Of lowest priority in the TRC is the RS232 interrupt from a local terminal which may be connected for manual testing purposes. Any INT0 or INT1 interrupt will interrupt the RS232 port.

3.2.1) Sync Received Interrupt

The sync interrupt routine is by far the longest interrupt routine in the TRC software. When a sync code is detected by the track recovery circuitry (a single Xilinx XC3030 chip), an interrupt is generated for the controlling 87C51. The source of this interrupt can be any or all of the active tracks being controlled.

Not all of the 8 PBD tracks that a given TRC microprocessor has responsibility for are necessarily used on any given tape pass. In some cases all 8 tracks may belong to the same pass. In other cases, some tracks will belong to one pass and the rest to a recording pass performed some time later. Only the tracks being processed on a given pass are enabled. All of the sync detectors for these active tracks are ORed together to form the INT1 87C51 Sync detect interrupt. The presence of head skew does not hurt since the sync pulses recorded on different heads but on the same recording pass will be within a few hundred 8 MHz bits of each other. Such a separation is too small to affect the microprocessor.

The presence of several head sync detectors in parallel increases the reliability of the sync detection process. If no sync detection logic on any active track detects the sync pulse of a given header, the 87C51 has an internal timer that can be use as a watch dog timer by making the proper mode selection.

When a sync pulse is detected on one or more active PBD tracks, or when the watch dog timer runs out, the response of the 87C51 micro is the following:

1) Calculate a track buffer address for the next sync pulse and write it to the Xilinx chip. The mechanism for calibrating the track buffer RAM in units of frame time is for the detection of a sync pulse to jam set the RAM address counter to a microprocessor supplied address. If a clock slip occurs during a frame, due to extra or skipped track clocks during, say, a long dropout, the track buffer RAM address is corrected by the detection of the next sync word. This detection will parallel load the write address counter with the address computed and supplied by the 87C51.

2) Write the track buffer address of the first bit of the new frame being received into a circular buffer which reflects

the contents of the track buffer at any instant of time. This circular buffer will be used when the time comes to output the new frame into the correlator. A second parallel circular buffer is supported as the header of each active track is processed and contains validity information for each frame in the track buffer.

3) Update the filtered time (see section 3.3.0) for the present frame.

4) Calculate the roll phase of the present frame.

5) DMA the 9 track headers into the microprocessor memory.

6) Analyze the contents of active headers to:

   a) Check the frame time for equality with the filtered time

   b) Tabulate the performance (along track parity errors, missing syncs, clock slips, bad CRC codes, ETC.) for all active tracks

   c) Evaluate the validity of tracks

7) De-roll and channelize the validity flags

3.2.1.1) Execution Times in the Sync Detect Interrupt Routine

The interval between sync detect interrupts is 2.5 msec. at the nominal playback speed. The maximum time spent in the interrupt routine, handling 9 tracks, is 1.35 msec., broken down as follows:

approx time to DMA set of 9 headers into the 87C51 memory	250	usec
maximum time of execution to analyze headers	0.95	msec
number of 87C51 instructions to process an error free header	51	
maximum number of 87C51 instructions to process header with errors	85	
maximum time to process one header	0.11	msec

3.2.2) New Frame Interrupt

The Track Recovery Card has an address comparator in the Xilinx logic that monitors read addresses of the track buffer for the start of new track frames. When the first bit of a new track frame is output to the DEF, a pulse goes with it to tell the DEF to update the track de-roll and to sense the determined validity of the new frame. Only a single pulse is required since all 8 tracks have the same delay and hence all start new frames at the same time.

When the TRC microprocessor receives a New Frame interrupt it must compute the next start of frame in the track buffer and write the new start address to the comparator. The microprocessor also looks through its circular buffer to get the validity of this next frame and writes this status to the DEF interface in advance of the next New Frame.

3.2.3) New Delay Interrupt

The DEF/TRC logic has an opportunity to set a new delay for a delay center every 4 msec. This period is called the New Delay period. When the TRC gets a New Delay interrupt it will write a new start address to the track buffer read circuitry for the start of the next New Delay period. This new address is a track buffer RAM address computed from the delay model and will result in the correct bit being output by the TRC into the DEF at the start of the next delay period.

3.2.4) DEF Communication Interrupt

The TRC 87C51 conforms to the TRC/DEF communication protocol defined in the D028TRC.HCB protocol document (see Appendix III). The TRC HCB communications pass through the Deformatter.

3.3.0) Initial Synchronization Strategy

Upon receiving a command to start recovering data from a PBD, the 4 TRC microprocessors will independently start trying to find frame headers. After configuring the TRC card logic for the mode selected by the real time computer, all 4 microprocessors will enable their INT1 interrupts. They write an initial start address out to the Data Playback Chips which is the track buffer memory location to which the next detected sync pulse will jam set the write address counters.

Once a sync word has been detected on any of the active tracks supervised by a given microprocessor, the microprocessor will initiate the DMA process which results in a set of 9 headers being read from their track buffers and written into the microprocessor memory. As the PBD starts up, spurious operation may occur. The first few attempts at obtaining headers may result in rubbish.

As each microprocessor tries to process the headers, it will look through the set of active tracks being processed. It looks for a header which had an error free sync code, has zero header parity errors, and has a correct CRC (such a header will be described as being "error free"). The first time it receives such a header, the 87C51 uses the contents of the time portion of that header to set its internal time.

Each 87C51 attempts to keep an internal "filtered time". This time is initially taken from an error free header as described above and is subsequently tracked from frame to frame. This is done by adding the duration of a frame (2.5, 5.0 or 10.0 msec.) to the filtered time every time a sync word is detected from the track signals (or when the watch dog timer runs out if this is enabled).

The filtered time is put in a probational state when initially set from the tape and is flagged as "valid" only after P3 successive tape frames, each of which has at least one error free header whose time

agrees with the established filtered time. P3 is a programmable parameter and can vary from 1 to 255.

Any time the filtered time fails to match the time code contained in at least one header, from among the set of headers, in P5 consecutive frames, the time is discarded (again P5 is an 8 bit programmable parameter). Once a filtered time has been discarded, the microprocessor starts again looking for an error free header and uses the first such header to again initialize its filtered time. Once reinitialized, the same sequence of P3 consecutive frames with "correct time" must be recovered before the filtered time is again flagged valid.

P3 and P5 are two of the eight observing parameters sent to the TRC via HCB function code 0x12.

A flow chart of the time tracking routine is presented in the HCB protocol document for the TRC, D028TRC.HCB, Figure 1. (A copy is included in Appendix III.)

Once a stable (flagged valid) filtered time has been established in at least one of the four TRC 87C51s, the Deformatter (DEF) card will use the filtered time from one TRC 87C51 to attempt PBD synchronism. To do this, the DEF will convert the TRC filtered time into a binary number which indicates the number of Nyquist samples taken since midnight of the appropriate Modified Julian Day (MJD). The delay models in the VLBA correlator are kept in terms of a MJD and the number of Nyquist samples taken since midnight of that MJD.

The DEF uses model parameters (a time reference that tracks the number of Nyquist samples since midnight and an integer delay offset from this linear track) to compare with the time from the tape frame headers. The result of this comparison is used to determine if a specific bit of interest is presently in the TRC track buffer. The model parameters identify the exact bit that is required to be output from the DEF as the first bit of every 4 msec. interval. The DEF servos the PBD to keep the target bit near the center of the TRC track buffer every 4 msec. At the start of an observation, the target bit may not yet have reached the TRC buffer, or it may have long ago entered and exited the buffer. See section 5.7.0 for more details on the servo process.

Any time the performance of the drive gets so bad that all continuity is lost, the TRC and DEF return to the initial condition and again attempt to establish correct time and PBD position. The DEF flags every FFT cycle as valid or invalid according to a set of validity criteria. Samples will be flagged as valid and the resulting cross products integrated only if:

1) The MJD in the filtered time is the same as the delay model MJD.

2) The TRC flags all frames from which the samples were recovered as valid.

3) The bit targeted by the model is near the center of the TRC buffer. (See section 5.7.1, Validity Byte, and section 5.7.2.)

4) The real time computer has not declared all data invalid (via DEF function code 0x40.)

3.4.0) Frame Validity Strategy

Once initial synchronism has been established, the TRC microprocessors will track the performance of the active tracks during the remainder of the pass. Each of the four 87C51 microprocessors will supervise the performance of up to 9 active tracks. One of the responsibilities of the 87C51 is to ascertain the probable validity of each bit it sends to the DEF.

Track validity is established on a tape frame basis. Each track frame is analyzed by a TRC 87C51 as the frame is written into its track buffer, and a validity bit is kept that reflects the 87C51's judgement on the quality of the frame. For a frame to be flagged valid the following criteria must be satisfied:

1) Filtered time must be valid.
2) The track must have a sync pulse in the predicted location (meaning that no clock slip was detected by the microprocessor). Detection of this sync pulse is made with an error tolerant sync detector.
3) The number of data parity errors in the frame was less than K (where K, from 1 to 255, is supplied by the realtime computer as an observational parameter)
4) The real time computer has not declared all data invalid (via TRC function code 0x15.)

The TRC software has a validity probation mode which, when enabled by the real time computer system, causes the TRC 87C51 to put a frame on probation if no sync word at all is detected in the frame header (with an error tolerant sync word detector). The 87C51 will wait for the next frame to be evaluated. If the next frame header for that track has a valid sync word at the sync predict time, the frame on probation is flagged as valid. Otherwise the frame is declared invalid. The probational state of a frame lasts for only one frame duration.

All validity decisions made on track frames are made as the tracks are input from the PBD and written into the track buffer. The validity flags are output to the DEF as the frames are output from the TRC into the DEF. The TRC 87C51 keeps a circular buffer of frame validity flags in its memory for this buffering purpose.

See section 4.2.0 for a more detailed description of the validity algorithm.

3.5.0) Delay Set Strategy

The strategy used to set the delay of bits that drive the DEF from the TRC is to calibrate the track buffer RAM address locations in units of time. The units of time used are Nyquist samples.

When a new frame is written into the track buffer, the time code and RAM location of bit 1 of the frame are kept. The DEF converts the

time into a binary number that is the number of Nyquist samples taken since midnight of the appropriate Modified Julian Day (MJD). The DEF takes the delay model from the computer system which is also in Nyquist samples since midnight. A simple conversion yields the address in the track buffer RAM where the appropriate bit is stored. This activity takes place every 4 msec and results in a new delay being set every 4 msec.

The position of the target bit in the track buffer is identified from the computation above. This is used by the DEF to servo the PBD and keep the target bit near the center of the buffer.

This page intentionally left blank.

Chapter 4 Data Validity in the Track Recovery Card

4.1.0) Introduction

The VLBA correlator forms self and cross spectral products from astronomical data recorded on magnetic tape. These products are then integrated for long periods of time. The astronomical data obtained from the tape playback drives will, from time to time, be invalid. The correlator must provide a mechanism for excluding invalid data from the final integration results of the system.

Many reasons for invalid tape data exist. The antennas not being on source, phase lock loops in the receiver being unlocked, and improper operation of the tape recorder or playback drive are a few.

For the purposes of this writeup, invalid data can be broken up into two categories:

1) Temporary but long term conditions that occur at the antenna (such as the antenna not being on source).

2) Improper operation of the tape system.

The first case above produces invalid operation that is, in general, known a priori. Spectra generated by the correlator from data so tainted can be excluded from the correlator output either by the real time computer system (RTS) setting long term invalid flags in the correlator software or by simply discarding the correlator output.

The second category of invalid operation above can be further broken down into more specific cases listed below:

1) Temporary or permanent problem in the recording tape drive head stack (for example, one bad recording head resulting in an unreadable record track).

2) Temporary or permanent problem in the playback tape drive (PBD) head stack (resulting in a bad playback track).

3) PBD not properly synchronized.

4) PBD head stack improperly positioned.

5) PBD encountering a portion of tape with low signal to noise performance, resulting in the momentary production of many parity errors.

6) PBD encountering a portion of tape with low signal to noise performance, resulting in a clock slip in the playback phase lock loop.

Item 6, above, is the most damaging type of invalid data. Other invalid data conditions will degrade the signal to noise ratio of the final correlator output whereas this type of fault can generate spurious results.

The track recovery card (TRC), can detect the occurrence of all the items above except items 3 and 4. Items 3 and 4 must be recognized and provided for by the RTS which has three options:

1) The RTS can use the function code 15 and function code 16 commands in the TRC protocol to control the TRC validity condition.

2) The RTS can use the function code 40 command in the Deformatter card (DEF) HCB protocol to control the DEF validity condition.

3) The RTS could simply discard the resulting LTA (Long Term Accumulator) integrations.

In the discussion below, a description of the TRC software used to deal with data valid considerations will be given, followed by a description of the hardware path of data validity related signals through the correlator.

4.2.0) TRC Software

The TRC can frequently recognize improper operation in the tape system by analyzing PBD track signals. The TRC has frame parity error counts, sync recognition flags, the results of CRC analysis on frame headers, and the analysis of the track time codes to help it in monitoring acceptable operation of the tape playback system. The TRC cannot recognize improper operation due to incorrect head positioning or due to incorrect tape location. In both of these cases the TRC can receive error free track signals and be unable to detect the existence of a problem.

The TRC software analyzes the PBD playback signals on a frame by frame basis, and will tag every frame as valid or invalid according to the results of this analysis. As data from a given frame is output from the TRC, a validity flag is also output to the DEF card giving the TRC microprocessor's judgment of the frame validity.

Before the TRC even begins attempting to ascertain the validity of PBD data, three prerequisite conditions must be met before any data can be flagged as valid by the TRC:

1) The TRC must have permission from the RTS via the TRC protocol function code 0x16 command to allow data to be valid. (This permission is now the default when function code 0x14 starts an observation. Originally it was not the default, and the RTS was required to send function code 0x16.)

2) The TRC must have a valid filtered time established (as defined by the TRC program time algorithm).

3) The TRC must be tracking the roll phase.

While in observing mode, the TRC tries to maintain an internal filtered tape time. By doing this, the TRC desensitizes its operation to occurrences of isolated incorrect time codes recovered from corrupted track signals. The TRC microprocessor will monitor the time recovered from the tape tracks and will set a "good time" flag after the time code recovered from several consecutive tape frames satisfies the good time algorithm. (See D028TRC.HCB, Fig 1, in Appendix III.)

The TRC microprocessor next uses the filtered time to determine when the PBD track roll is at a roll phase of zero. When it is, a "roll set" flag is established and the roll phase is subsequently tracked from frame to frame.

Once all of the prerequisite criteria above have been satisfied, the frame by frame evaluation for validity will begin. If any improper PBD operation occurs that results in time being lost (such as at a tape splice), the TRC program returns to a start condition and must re-establish good filtered time and roll phase before valid data will again be permitted.

As long as the TRC microprocessor has a good filtered time flag and a good roll set flag, it uses the criteria below to decide if a given frame from a given track is valid:

1) Was the sync pulse from the frame detected at a time that coincided with the sync predict time?

2) Was the sync pulse detected at all?

3) Was the frame parity error count less than the permissible threshold (as set by an RTS supplied observation parameter)?

If the sync pulse was detected offset in time from the TRC Xilinx track recovery chip sync predict signal, then the frame is declared invalid (this condition is the worst case fault that can produce spurious results).

If no sync pulse is detected at all, operation can follow one of two paths depending on the RTS supplied observation parameters. In one mode, a failure to detect a sync pulse causes the frame to be declared as invalid. In the other mode, judgement on the frame is withheld for one more frame period. If the next frame has correct sync operation, both are accepted (so far as this criteria is concerned). If the next frame shows the sync detect to be offset in time from the sync predict, both frames are set invalid. If sync is again not detected in the following frame, the earlier frame is set invalid.

If the parity error count from a given frame (on a given track) is above a threshold set by an observation parameter, the frame is set invalid.

The TRC microprocessor maintains a 16-word circular buffer in memory that gives the start memory location of frames in the 64K X 4

track RAMs (there will be 12 and a fraction such frames within the buffer at all times, this circular buffer records the RAM location of the first bit of the last 16 frames received from the PBD). Since all tracks work in parallel, so far as the PBD input is concerned, one such buffer suffices for all of the tracks supported by a given TRC microprocessor. The TRC microprocessor also keeps a second buffer that parallels the first in which it stores the track by track validity flags corresponding to frames addressed by the first buffer.

The TRC microprocessor closely monitors operations as the TRC card outputs recovered track data into the DEF card. The address of the 64K X 4 track buffer RAM location supplying the card output data is monitored for the start address of new frames. Every time a frame boundary is crossed, a NEWFRAME pulse is generated for the DEF card (the DEF card must know frame boundaries for both track de-roll and validity functions). When a frame boundary is crossed, the TRC microprocessor goes to its circular buffer and sets the logic up to watch for the RAM location of the start of the next frame. The microprocessor writes the start location of the next frame into a hardware comparator that monitors the RAM read addresses. The TRC microprocessor also writes the validity flag for the new frame into secondary storage. When the new frame start address is seen by the comparator, it generates a NEWFRAME pulse for the DEF and causes the validity flag appropriate to the new frame to be output to the DEF card.

The validity algorithm used to establish the validity of PBD frames works at the track level. The DEF card, however, wants validity information on a channel basis. Hence, the TRC microprocessor does a track to channel conversion before outputing validity flags to the DEF card.

The first action required by the TRC in the track to channel conversion process is to de-roll the track validities. At the start of an observation, the TRC microprocessors receive information specifying the track to channel assignments for the observation (i.e., what tracks the DEF must combine to reconstruct a given channel). The track roll function of the formatter, however, scrambles this assignment and the TRC microprocessor must de-roll the validities to find which channel will get polluted when a specific track is bad.

Once the track to channel conversion of the validities is made for a given frame, the flags are tucked away in a buffer to await the time when data from that frame is output into the DEF card. Note that the validity of frames are determined as they are received from the PBD. Data from a given frame will stay in the TRC track buffer for up to 32-msec before being output to the DEF and the circular buffers described above will store the validity flags until output of the frame data occurs.

A flow chart of the TRC algorithm described above is presented in the HCB protocol document for the TRC, D028TRC.HCB, Figure 2, in Appendix III. An alternate flow chart of the same algorithm (with more detail) is presented in Figure 5 TRC VALIDITY ALGORITHM at the end of this chapter.

4.3.0) Data Valid Hardware

The hardware path through the correlator for validity signals starts at the TRC output. Each PBD drives two TRC cards, each of which

has two microprocessors for the recovery of the PBD output track signals. Each microprocessor oversees the recovery of up to 8 PBD data tracks and one system track. Each microprocessor has an 8 bit bus over which it can provide the DEF with data validity information.

TRC output pins P1A-20 through P1A-27 (VAL-A[0..7]) and output pins P1B-20 through P1B-27 (VAL-B[0..7]) are available to a TRC for the conveying of validity information to the DEF card. Of the 8 bits in each bus (for a total of 32 bits out of the two TRC cards) only 2 are required for validity purposes since the TRC microprocessors have already made the track to channel conversion for the validity. Thus P1A-20, P1A-21, P1B-20, and P1B-21 from a given TRC card carry data valid information. This results in an 8 bit bus driving the DEF card carrying validity information for the 8 channels the DEF will produce (after the DEF performs the track to channel conversion on the data bits). In spite of the labels on the TRC signals (VAL-A0, for example) indicating a high true data valid signal, these signals are high when the channel is invalid.

One other signal in the VAL-A[0..7] bus (and one in the VAL-B[0..7] bus) is used by the TRC (the other 5 lines in each bus are unassigned spare TRC outputs). VAL-A2 and VAL-B2 (a total of 4 signals from the 2 TRC cards) are used to convey the track roll phase to the DEF card (note that the 4 TRC microprocessors of the two TRCs assigned to a given PBD correspond to the 4 delay centers supported by the VLBA correlator and also to the 4 roll groups supported by the formatter). Every time the track roll phase for a given roll group is zero (i. e., the track roll state is as if the track roll function were not active), the corresponding TRC output (VAL-A2, for example) is high. The DEF card must know the roll phase because one function of the DEF is to de-roll the tracks before the track to channel conversion is made on the track data.

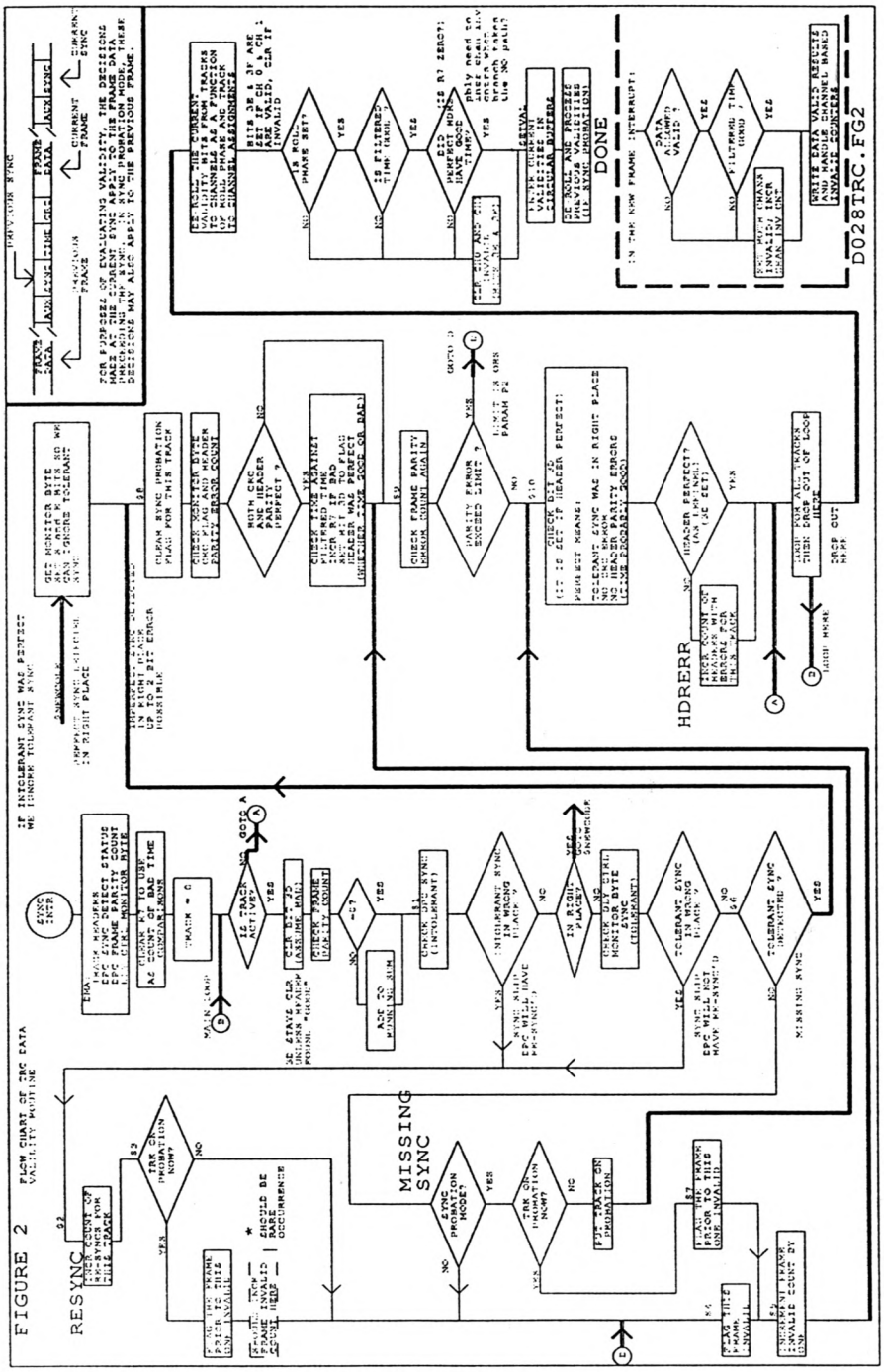
Once on the DEF card, the 8 bit DATA_INV[0..7] bus drives the U40 DATA INV Xilinx chip (see sheet 17, L008D17.SCH of the DEF schematic, in Volume 2). The principal function of this chip is to support the back-up in track data necessitated by the 4 bit gap between FFT cycles. Because of this 4 bit gap, the TRC outputs too many bits of track data into the DEF during a 4-msec fringe cycle. At the start of every NEW DELAY cycle, the TRC must back-up in its track buffer by 256 8 MHz track bits to insure that all PBD bits are used. See section 5.3.4 for more details.

When this 256 bit back-up occurs, action must be taken when a frame boundary is (re)crossed (i.e., track data coming out of the TRC card finishes an old frame and starts into a new frame when a NEW DELAY cycle back-up occurs and lands in data from the old frame). Thus the DATA INV Xilinx chip keeps an old version of the data invalid flags and, by counting the number of clock cycles between a NEW DELAY pulse and the last NEW FRAME signal, can resurrect the last frame's validity states if the NEW DELAY back-up crosses a frame boundary and starts outputing bits from the previous frame.

The dDATA_VAL[0..7] bus out of the DATA INV Xilinx chip is low true (high if the data is invalid) and drives the buffer RAM seen on sheet 14 of the DEF schematic. The DEF buffer RAM has a full bandwidth path for data bits and a narrow band path (ICs U54 and U53) for the validity bits. The PALs U63 and U62 output the channel-wise validity

signals (now high true or high if the data is valid) on an FFT cycle basis. If any bit of an FFT cycle is invalid, the PALs hold the invalid logic level for the remainder of the FFT cycle.

Figure 5 TRC VALIDITY ALGORITHM



This page intentionally left blank.

Chapter 5 The Deformatter Card

5.1.0) Introduction

The Deformatter Card (DEF) is used to re-construct the original 32 MHz sampler bit streams that went into the Formatter. The 32 MHz streams are reconstructed from the 8 MHz track bit streams recovered by the two Track Recovery Cards (TRC) that drive the Deformatter.

5.1.1) Deformatter Overview

The Deformatter Card is a six layer printed circuit board. A total of 24 DEF are installed in the VLBA Correlator, and there are 3 spares. Card layout and fabrication services for the 27 cards were purchased from Tri-Circuits Inc. under NRAO PO number C15415. Photo-plots of the six layers and silk screens are presently stored in a file drawer in drafting. There are no design files available that could be sent to a PCB vendor for production of new cards.

The primary functions of the Deformatter card include:

Barrel un-roll

Track bit stream de-muxing or muxing

Oversampling and overlapping support

Data validity synchronization

Playback drive servoing

A 68000 microprocessor (68K), running at 16 MHz and programmed in assembly language, is used for embedded control. A Hardware Control Bus (HCB) interface is provided for communications with the VME real time control computer. There are three ram based logic sequencers that provide many of the control signals on the card. The sequencer contents are mode dependent.

There are six Xilinx FPGA chips on the card. Four of these use identical personalities that are mode dependent (different personalities are downloaded to these chips as a function of the Formatter mode at the time of the observation). These are the Bit Shuffler Xilinx chips. Of the other two Xilinx chips, one is the Barrel Roll Xilinx, with a fixed personality. The other is the Data Invalid Xilinx, with one standard personality and one alternate personality (for possible use with Mark III / IV tape formats). There are 49 PAL chips on the card, with 30 distinct source files. All PAL source code is written in ABEL.

A PROM based monitor program, running on the 68K, is installed on the card, so that basic terminal functions are available after a power on

reset is applied (via an input on the HCB interface). The monitor program includes a HCB handler that provides bootstrap loading of the RAM based 68K program, along with Deformatter sequencer and Xilinx personality files.

The DEF communicates with a pair of TRC over a 8 bit wide parallel bus. The bus is used for both local DEF/TRC transfers and for HCB/TRC transfers that are passed through the DEF.

5.1.2) Documentation Overview

All hardware and firmware related files for the VLBA correlator are maintained under SCCS version control, in one of two areas. These two areas are identified as the vlbsoft and corrdwgs areas. The path to vlbsoft is /home/magnolia2/vlbsoft and the path to corrdwgs is /home/azalea/corrdwgs (at the present). All drawing filenames are lower case, but may appear in documentation as either upper or lower case.

The vlbsoft area is where all source code for the correlator is maintained. The Deformatter 68000 assembly source code is maintained in the vlbsoft/defasm directory along with the HCB protocol document, D027DEF.HCB (see Appendix IV). There are presently 53 .asm source code files and 16 include files (.h and .mac). The Unix MAKE utility is used to control assembly and linkage, using the 2500AD cross assembler tool set.

The vlbsoft area is also used to store all source files that are processed in any manner to produce objects that are downloaded into the correlator or programmed into devices that are installed in the correlator. This includes pal source code, Xilinx design files and sequencer source files.

The corrdwgs area is where Orcad schematic files, netlists, partlists, etc. are maintained. For the Deformatter, these types of files are maintained in the corrdwgs/def directory.

NRAO drawing number 56000L008 is the schematic drawing of the Deformatter. This Orcad drawing consists of 31 sheets, with the Orcad file names L008D01.SCH - L008D31.SCH. Unless otherwise noted, sheet number references in the Deformatter description refer to these sheets. Copies of these drawings are in Volume 2 of this report.

Sheet 1 contains entry points into other Orcad drawings that are maintained in the same SCCS directory, such as the representative logic drawings for the PAL and Xilinx designs and the component layout drawing.

NRAO drawing number 56000Z009 is the component layout drawing for the Deformatter. This Orcad drawing consists of 2 sheets, with the Orcad file names Z009D01.LAY and Z009D02.LAY. The second sheet of the layout drawing contains details that were of specific interest in the component assembly process. Sheet 1 of the layout drawing is the sheet normally used to identify the location of parts on the card.

A complete list of files in every directory below vlbsoft and corrdwgs that is used to maintain Deformatter related files is given in Appendix I. Figure 6 DEFORMATTER FILE SUMMARY tabulates the main hardware related files maintained in the /corrdwgs/def/SCCS directory.

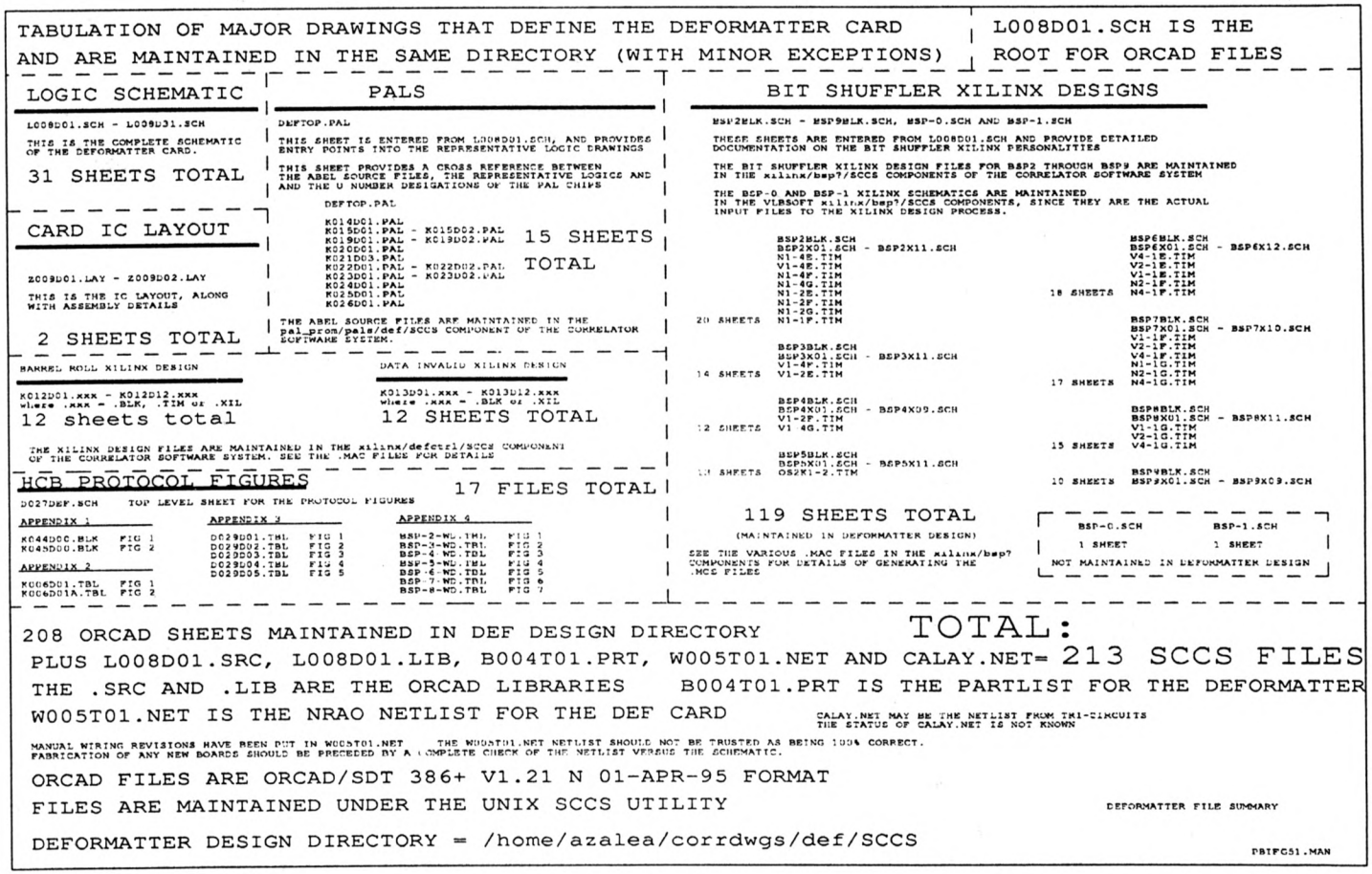


Figure 6 DEFORMATTER FILE SUMMARY

5.2.0) Modes

The Deformatter HCB protocol document, D027DEF.HCB, presents a table of modes in it's Appendix 2, Figure 2. These modes are defined in terms of bit-to-track mode, FFT size and oversample factor.

5.2.1) Bit-to-Track Mode

The Deformatter can handle the following Formatter modes:

Table 6 BIT-TO-TRACK MODES

1 to 4	One bit stream, fan out onto four tape tracks
1 to 2	One bit stream, fan out onto two tape tracks
1 to 1	One bit stream goes to one tape track
2 to 1	Two bit streams, fan in to one tape track
4 to 1	Four bit streams, fan in to one tape track

2 to 1 and 4 to 1 fan in modes are not officially supported, but PBI testing has been done in many of these modes

5.2.2) FFT Size

The Deformatter supports the following FFT sizes:

Table 7 DEFORMATTER FFT SIZES

FFT size	Number of time ordered points into FFT	Number of spectral channels out of FFT
64	64	32
128	128	64
256	256	128
512	512	256
1024	1024	512
2048	2048	1024

5.2.3) Oversample Factor

The Deformatter handles oversample factors of 1 and 2 where every bit is passed to the FFT. For oversample factors of 4, 8 and 16, bits are discarded. For an oversample factor of 2, the first sample of a pair is the "EVEN" sample and the second is the "ODD" sample. For an oversample factor of 4, the first sample is kept as the "EVEN" sample and

the third sample is kept as the "ODD" sample. The second and fourth samples are discarded.

5.2.4) Record Rate

Tapes may be recorded at 8 MB/s, 4 MB/s or 2 MB/s. The following table relates the record rate to oversample factor and bit-to-track mode, for all supported observing bandwidths. Data is always played back into the PBI at 8 MB/s.

Table 8 RECORD RATES

SAMPLE RATE MHz	BAND WIDTH MHz	OVS FACT	RECORD RATE 1-4	1-2	1-1	2-1	4-1 ←-(B-T)
32	16	1	8MB/s				
32	8	2	8MB/s				
32	4	4	8MB/s				
32	2	8	8MB/s				
32	1	16	8MB/s				
16	8	1	4MB/s	8MB/s			
16	4	2	4MB/s	8MB/s			
16	2	4	4MB/s	8MB/s			
16	1	8	4MB/s	8MB/s			
16	0.5	16	4MB/s	8MB/s			
8	4	1	2MB/s	4MB/s	8MB/s		
8	2	2	2MB/s	4MB/s	8MB/s		
8	1	4	2MB/s	4MB/s	8MB/s		
8	0.5	8	2MB/s	4MB/s	8MB/s		
8	0.25	16	2MB/s	4MB/s	8MB/s		
4	2	1		2MB/s	4MB/s	8MB/s	
4	1	2		2MB/s	4MB/s	8MB/s	
4	0.5	4		2MB/s	4MB/s	8MB/s	
4	0.25	8		2MB/s	4MB/s	8MB/s	
4	0.125	16		2MB/s	4MB/s	8MB/s	
2	1	1			2MB/s	4MB/s	8MB/s
2	0.5	2			2MB/s	4MB/s	8MB/s
2	0.25	4			2MB/s	4MB/s	8MB/s
2	0.125	8			2MB/s	4MB/s	8MB/s
2	0.0625	16			2MB/s	4MB/s	8MB/s
1	0.5	1				2MB/s	4MB/s
1	0.25	2				2MB/s	4MB/s
1	0.125	4				2MB/s	4MB/s
1	0.0625	8				2MB/s	4MB/s
0.5	0.25	1					2MB/s
0.5	0.125	2					2MB/s
0.5	0.0625	4					2MB/s

5.2.5) Mode Testing

Checkout of the PBI modes was done using the Mini-Transport as a data source. The Mini-Transport produces a simulated tape frame that contains pseudo-random bit patterns in the data part of the frame. For each mode, the PBI is configured and the recovered data is collected on a logic analyzer. The Mini-Transport is set up to repeat the same data every 4 msec. The logic analyzer captures approximately 8 FFT cycles of data, starting with the first FFT cycle of a 4 msec. interval. The results captured on the logic analyzer were compared to predicteds generated by C programs.

This mode testing verified the basic re-construction of data streams, and helped "fine tune" some of the sequencer control signals.

The mode scripts used to set up the PBI for these tests, and the C programs that generate the predicted results, evolved over the years in which these tests were done. The C programs generated files with approximately the first 4000 states out of the PBI (in a 4 msec. interval). The logic analyzer captured these states and the results were transferred to a file. Versions of both files were edited to a common format and then compared using the UNIX DIFF utility. The scripts, C programs, actual and predicted results are presently still stored on disk, but they are not officially stored as part of the system documentation. The present path to the mode checkout data and programs is /home/w5uxh/cbroadwe/mode_checks. See Appendix II for a sample script, C program and a fragment of predicted and actual results.

5.3.0) Block Diagram

Figure 7 DEFORMATTER BLOCK DIAGRAM is a block diagram of the data flow path through the Deformatter. This diagram reflects the delay center based architecture of the PBI. The recovered data from 36 playback tracks are input to the Deformatter from the TRCs, at a data rate of 8 MHz. The tracks are grouped in four sets of 8 tracks, with a system track associated with each group of eight. Each group of 8 tracks plus one system track are input to one of the four delay centers on the DEF. Each of the four delay centers processes the data for two FFT engines (or pipelines). The TRC DELAY RAM sections at the left of the block diagram represent the inputs from the TRC. The Deformatter schematics refer to "track numbers" in a generic fashion as follows:

Delay Center	"TRACKS"
0	0-7
1	8-15
2	16-23
3	24-31

These numbers bear no direct relation to Formatter or Tape track numbers.

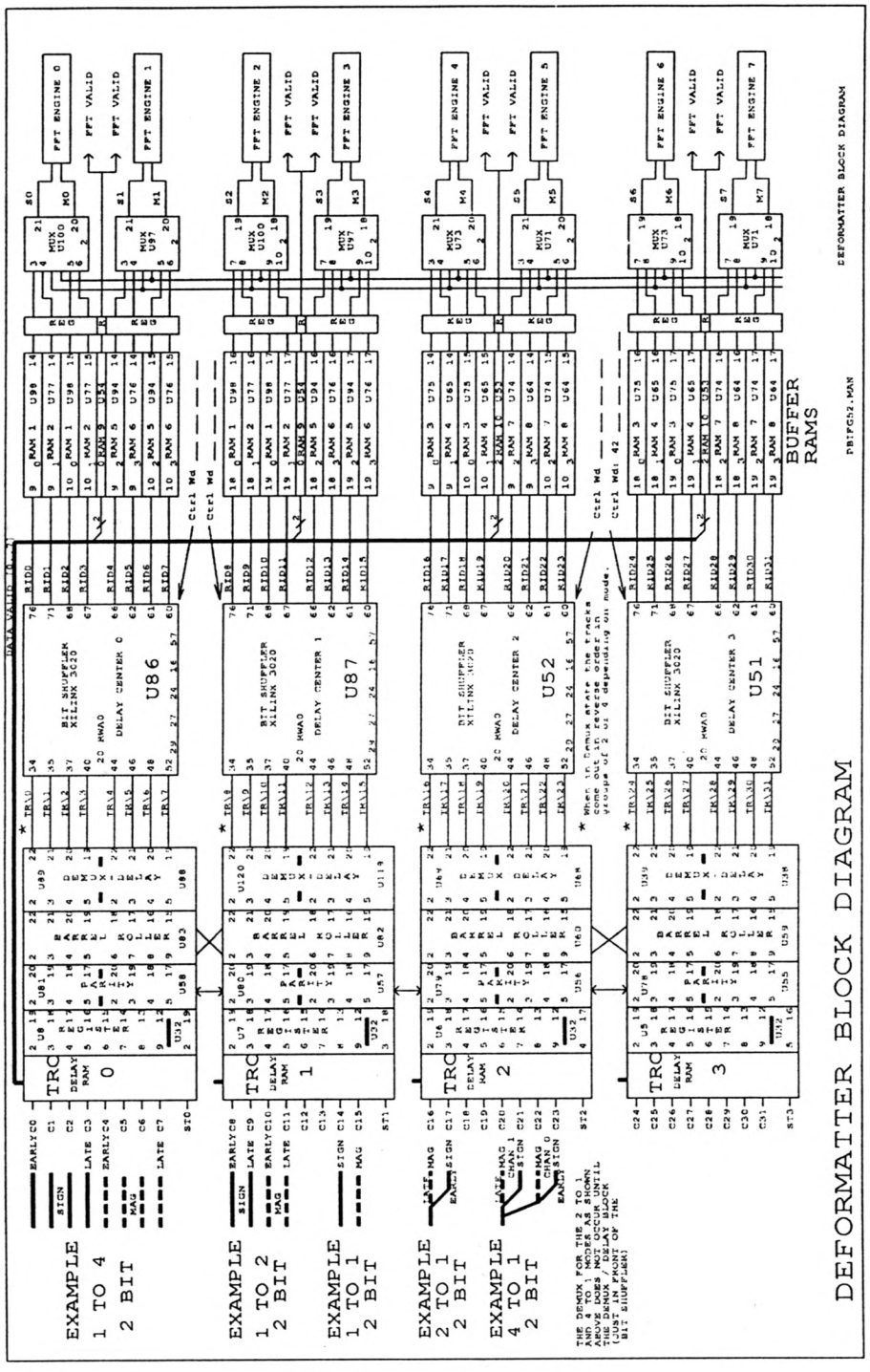


Figure 7 DEFORMATTER BLOCK DIAGRAM

5.3.1) Data Capture and Cross Track Parity

The REGISTER and PARITY sections of Figure 7 represent the logic found on DEF schematic sheets 3, 4, 5 and 6. In the following description, integrated circuit reference designators are given for the logic on sheet 3 (delay center 0). The logic on sheets 4, 5 and 6 is identical.

A group of 8 tracks is captured in register U8 on sheet 3. (The system tracks for all four groups of 8 are captured separately in U32 on sheet 2.) After capture, the data passes through the CROSS TRACK PARITY logic. Cross track parity has not yet been implemented in VLBA operations. The intention of cross track parity is that parity can be generated across 8 or 16 tracks with the result stored on a system track. When a known bad track has been recorded, the Deformatter can be commanded (via HCB function 0x36) to invert the data from the bad track in response to detection of parity errors. (This is on a track bit by track bit basis.)

Cross track parity is used in system test in the correlator. In system test, all 8 tracks plus the system track in each delay center have identical data. This is because the simulated track data from the Master Control Card simulates a single track. This data source is fanned out to all TRC inputs so that all delay center inputs are identical. By commanding the Deformatter to invert one of the tracks when a "parity error" is detected, the selected track will end up producing static data at the output of the cross track parity stage. When this is combined with a command to un-roll the data at the next stage of the logic path, the 8 bit streams coming out of the un-roller will no longer be identical. This allows system test to detect some faults in the un-roll logic that would otherwise not be detected, since un-rolling of 8 identical tracks still results in 8 identical bit streams.

The U142/U116 counters on sheet 3 provide a count of the number of cross track parity errors. The results from these counters are intended to be read over the HCB using HCB function 0x60. It does not appear that the hardware counters are ever read and transferred to memory to be made available to the HCB function. Thus if cross track parity operation is ever implemented, and if access to the counters is needed, some additional Deformatter programming will be required.

An additional function provided in the cross track parity block is track substitution. If a track is known to be bad prior to recording, the system track can be used to substitute for the bad track. The cross track parity logic can then be commanded (by HCB function code 0x36) to substitute the system track back in place of the bad track bit stream.

See Appendix 3, Figure 5 of the Deformatter HCB protocol document (D027DEF.HCB in Appendix IV) for the bit fields used to configure the cross track parity logic.

5.3.2) Barrel Un-Roll and DeMux-Delay

The BARREL ROLLER and DEMUX-DELAY sections of Figure 7 represent the logic found on sheets 7 and 8. Barrel roll is a function used in the VLBA Formatter to guard against complete loss of data as a result of a bad track. If one track is dead and roll is not used, then the channel recorded on that track is declared 100% invalid. When rolling is used, the bit streams from the samplers are switched to use a different tape track each frame. Thus each channel loses a percentage of data, but no channel loses all data.

Figure 8 BARREL ROLL PHASES represents the track rolling and un-rolling actions for all 8 phases of roll in a single group of 8 tracks, with steps of one. At roll phase zero, the roller and un-roller both pass the 8 input bit streams straight through. At roll phase 1, the Formatter switches input 0 to output 7 so the Deformatter must switch input 7 back to input 0. At each remaining phase, it can be seen that similar complementary behavior exists in the Formatter/Deformatter switching.

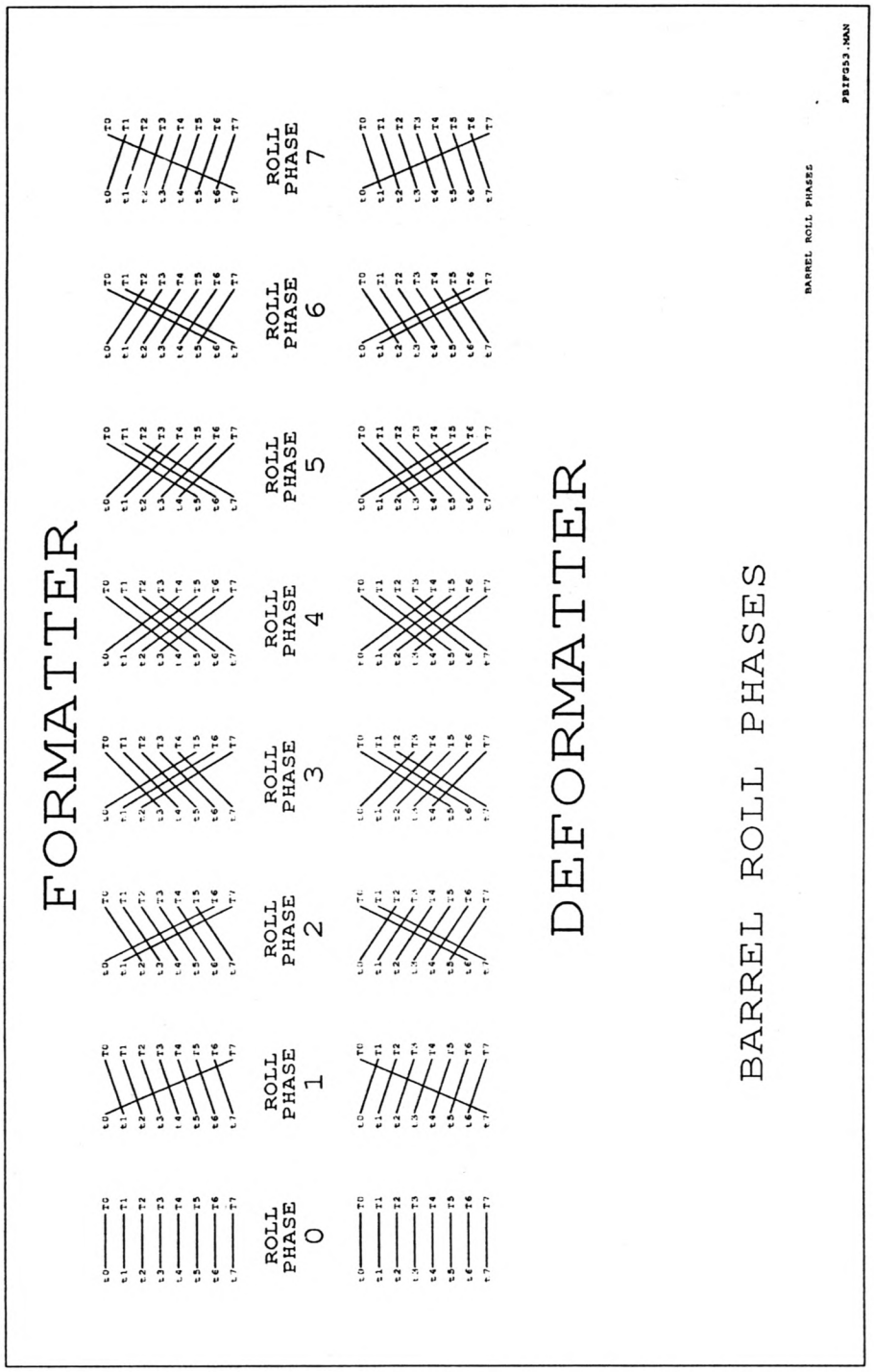


Figure 8 BARREL ROLL PHASES

The Deformatter supports roll steps of 1, 2 or 4 tracks per frame, and roll in groups of 8 or 16. Only steps of 1 have been tested. Un-roll within groups of 8 is done at pals U83, U82 on sheet 7 and pals U60, U59 on sheet 8. The pal outputs go to 2 into 1 multiplexers that swap the groups of 8 for un-roll in groups of 16 (U105, U117, U106 and U118 on sheet 7 for Delay Centers 0 and 1). For roll phases 0-7, the multiplexers are straight through. For roll phases 8-15, the two sets of 8 tracks in pairs of delay centers are swapped by the multiplexers (delay centers 0 and 1 form one such pair, delay centers 2 and 3 form the other pair). Control of the un-roll operations is determined by the Barrel Roll Xilinx, U128, on sheet 16.

Roll phase synchronization is handled by the TRC, which uses the time field to identify when roll phase 0 occurs. The signal from the TRC to the DEF that indicates roll phase 0, enters the DEF on DIN pin P3B10 (sheet 1). This is the on card signal named I/O_SPARE1 that goes to the Data Invalid Xilinx at U40 pin 37 on sheet 17. The signal is routed through the Xilinx to U40 pin 5 and then to U124 pin 3 on sheet 23 where it is read by the 68K microprocessor. At each NEW_FRAME interrupt, the DEF checks the roll phase 0 signal to see if the current frame is defined as having a roll phase of 0. This allows the un-roll to be re-synchronized in case synchronization has been lost.

Un-roll operations are complicated by the existence of the four cycle gap. Every FFT cycle of 512 points has an additional four clock cycles at 32 MHz, for 516 clocks per FFT cycle. These extra four clock cycles are referred to as the four cycle gap. The data from the TRC does not have a corresponding gap. The un-roll actions seen in Figure 8 need to change at every frame boundary. U128 controls this operation. The complication due to the four cycle gap relates to the operation of the read and write pointers in the output buffer ram, and is described in section 5.3.4.

The DeMux-Delay stage Delay function provides a vernier delay for the 1 bit stream to 4 track and 1 bit stream to 2 track modes. Since the TRC can only resolve delay to the track bit level, the Deformatter must provide the finer delay steps in these two modes. The setting of the delay is an internal function of the Deformatter and not controlled directly by the real time computer. As a result, HCB function code 0x33, described in Appendix 3, Figure 2 of the Deformatter HCB protocol document (D027DEF.HCB) defaults to requiring all four parameters equal zero for 1 to 4, 1 to 2 and 1 to 1 bit to track modes. (See Appendix IV for a copy of this protocol.)

The DeMux-Delay stage Demux function provides de-multiplexing for the 2 bit stream to 1 track and 4 bit stream to 1 track fan in modes. Present plans call for no further support of these two modes.

5.3.3) Bit Shuffler

The eight 8 MHz bit streams output from the Demux-Delay stage are ready to be re-combined into the fina. bit streams to be output from the DEF. This operation is a function of the original Formatter mode and is

handled in four re-configurable Xilinx chips seen on sheets 10 through 13 and in the center of Figure 7 as U51, U52, U86 and U87. There are ten different Xilinx bit shuffler personalities. For a given Formatter mode, the correct bit shuffler personality must be downloaded to the Deformatter, and the correct bit shuffler control words must be shifted into the Xilinx chips.

The bit shuffler chips re-combine the 8 MHz streams into 16 MHz streams that are written into the buffer rams seen on sheet 14. The outputs from pairs of buffer rams are re-combined into the final 32 MHz streams.

The Deformatter HCB protocol document, D027DEF.HCB, presents a table of modes in it's Appendix 2, Figure 2. This table defines which bit shuffler Xilinx personality file to download for each mode. The necessary input parameters for making this choice are bit-to-track mode, FFT size and oversample factor. The personalities are identified as BSP-0 through BSP-9. BSP-2 is the most widely used. It is used in 11 out of 33 total distinct modes. Figure 9 MODES FOR EACH BSP tabulates the modes for each of the BSP personalities.

Each of the four bit shuffler Xilinx chips requires a 32 bit control word. The generation of the control words is a function of bit-to-track mode, FFT size, oversample factor, number of bits sampling and which delay center inputs (0-7) are to be routed to the two FFT pipelines driven by the delay center. The control words primarily drive multiplexer select lines inside the Xilinx. The software function that generates the control words is getBitShuf(). The function uses simple equations to define the multiplexer select bit fields in the control word, as a function of the specified parameters.

Figure 10 BIT SHUFFLER EXAMPLE presents an example of how the 32 bit control word for one delay center BSP would be generated manually. The mode used in this example is:

```
1 bit stream to 4 tracks
FFT size = 512
oversample factor = 1
number of sample bits = 2
delay center input # 0 is lowest input assigned to FFT chan 0
delay center input # 0 is lowest input assigned to FFT chan 1
```

CH0TRK and CH1TRK in the figure are defined as the lowest delay center input numbers assigned to FFT channels 0 and 1 respectively. All eight delay center inputs are required to handle the sign and magnitude bit streams for a single FFT channel. In this example, it only makes sense to assign identical data to both FFT channels.

The VLBA two bit sampling mode does not encode the two bits as sign/magnitude, but the two bits are sometimes referred to in this manner. The two bit sample is defined as follows:

Table 9 TWO BIT SAMPLE STATES

```
                    B1 B0   where B1 = bit 1 = MS bit = Sign bit
                    -- --         B0 = bit 0 = LS bit = Magnitude bit

                +n=  1  1
+Vth    ......................  positive sampling threshold

                +1=  1  0

0V      ______________________

                -1=  0  1

-Vth    ......................  negative sampling threshold

                -n=  0  0
```

BSP-2

NR	B-T MODE	FFT SIZE	OVS MODE
1	1-4	<= 512	NOT OVS
2	1-2	<= 512	NOT OVS
3	1-1	<= 512	NOT OVS
4	2-1	<= 512	NOT OVS
5	4-1	<= 512	NOT OVS
6	1-4	<= 512	OVS2
7	1-4	1K	NOT OVS
8	1-2	1K	NOT OVS
9	1-1	1K	NOT OVS
10	1-4	2K	NOT OVS
11	1-2	2K	NOT OVS

BSP-3

NR	B-T MODE	FFT SIZE	OVS MODE
1	1-2	<= 512	OVS2
2	1-4	1K	OVS2

BSP-4

NR	B-T MODE	FFT SIZE	OVS MODE
1	1-2	1K	OVS2
2	1-4	2K	OVS2

BSP-5

NR	B-T MODE	FFT SIZE	OVS MODE
1	1-2	2K	OVS2

BSP-6

NR	B-T MODE	FFT SIZE	OVS MODE
1	1-1	<= 512	OVS2,4,8,16
2	2-1	<= 512	OVS2,4,8,(16)
3	4-1	<= 512	OVS2,4,(8),(16)
4	2-1	1K	NOT OVS
5	4-1	1K	NOT OVS

() IN OVS COLUMN INDICATES NOT ALL SMALLER FFT SIZES HAVE VALID MODES

BSP-7

NR	B-T MODE	FFT SIZE	OVS MODE
1	1-1	1K	OVS2,4,8,16
2	2-1	1K	OVS2,4,8
3	4-1	1K	OVS2,4
4	1-1	2K	NOT OVS
5	2-1	2K	NOT OVS
6	4-1	2K	NOT OVS

BSP-8

NR	B-T MODE	FFT SIZE	OVS MODE
	1-1	2K	OVS
	2-1	2K	OVS
	4-1	2K	OVS

MODES FOR EACH BSP

SEE K006D01A.TBL FOR DETAILS OF BSP ASSIGNMENTS

K006D01A.TBL IS APPENDIX 2, FIGURE 2, IN THE DEFORMATTER HCB PROTOCOL DOCUMENT, D027DEF.HCB

BSP-0

BSP-0 WAS DERIVED FROM BSP-5 FOR 1-4, 2K, OVS4

BSP-1

BSP-1 WAS DERIVED FROM BSP-3 FOR 1-4, 512, OVS4

BSP-9

BSP-9 WAS DERIVED FROM BSP-4 FOR 1-4, 1K, OVS4

MODES FOR EACH BSP

PBIFG54.MAN

Figure 9 MODES FOR EACH BSP

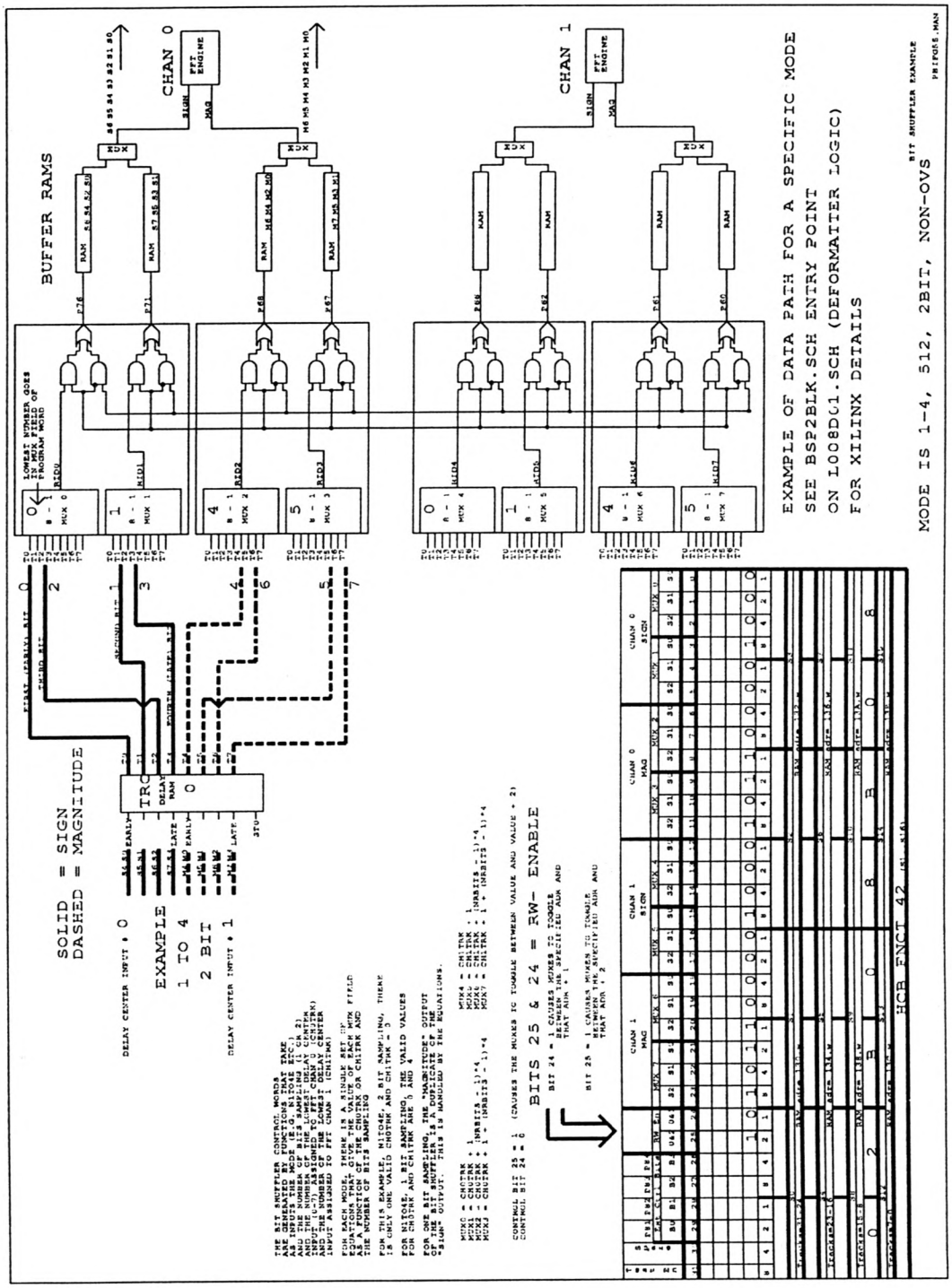


Figure 10 BIT SHUFFLER EXAMPLE

In Figure 10, the sign and magnitude bit streams are identified as S0, S1, S2, ... M0, M1, M2 etc. Delay center input #0 is seen to have samples S0 and S4 and the other samples are distributed over the other delay center inputs. Looking only at the SIGN bits, S0, S1, S2 and S3 are presented simultaneously to the top two multiplexers MUX 0 and MUX 1. S0 and S1 are then selected at the two multiplexers and written into the pair of buffer rams. At the next ram write, the multiplexer addresses both toggle to the next higher address plus 1, so that samples S2 and S3 are written into the rams. Thus MUX 0 needs to toggle between inputs 0 and 2, while MUX 1 toggles between 1 and 3. The control of the address toggles is by way of control word bits 24 and 25. In this case bit 25 is set to accomplish the correct toggle. The MUX fields in the control word need to be programmed to the lowest of the two addresses of interest at the multiplexer. Thus MUX 0 is programmed to 0 and MUX 1 is programmed to 1. The magnitude bit stream is handled in a similar manner, so MUX 2 is programmed to 4 and MUX 3 is programmed to 5. The final 32 bit control word then becomes 0x02B08B08 as seen in the table in Figure 10

5.3.4) Buffer Rams

The buffer rams seen in Figure 7 and on sheet 14 of the schematic are circular, and 8192 locations deep. They are implemented with two interleaved 4096 deep rams. U98 and U77 are interleaved and handle the bit streams for Channels 0 and 2 (FFT Engines 0 & 2). Three other pairs of physical rams handle the remaining six channels.

A buffer holds 16 FFT cycles of data (512 bits times 16 = 8192). The Deformatter operates on two different 4 msec. cycles when writing to and reading from the buffer. One of the 4 msec. cycles is called NEW_DELAY. The other is called NEW_FRINGE. At the start of the NEW_DELAY cycle, the buffer ram write address is reset to zero and data from the TRC starts filling the buffer. At the start of the NEW_FRINGE cycle, the buffer ram read address is reset to zero and data is read out to the FFT card.

The NEW_FRINGE read cycle is delayed by a minimum of two FFT cycles from the NEW_DELAY cycle, in order to provide time to fill the buffer with enough data to be read out at 32 MHz. In some modes, the NEW_DELAY to NEW_FRINGE delay is as much as 128 FFT cycles, which is one-half of the 4 msec interval. (There are 256 FFT cycles in a 4 msec. interval.)

The write pointer gains on the read pointer due to the four cycle gap. Every FFT cycle of 512 points has an additional four clock cycles at 32 MHz, for 516 clocks per FFT cycle. In any mode where there is no overlapping (discussed below), the Deformatter puts out 512 samples during one FFT cycle. During the next four clock cycles, the Deformatter continues with the present stream of samples. It then backs up four samples and starts the next FFT cycle. The data from the TRC does not have a corresponding gap. At the beginning of every 4 msec. NEW_DELAY cycle, the TRC is backed up 256 8 MHz bits (256 FFT cycles in 4 msec. times 4 bits each is 1024 32 MHz bits, or 256 8 MHz bits). This compensates for write pointer gain.

Looking at a simple example, for the case of 1 bit stream to 4 track mode, 512 point FFT, non-oversampled, we see that data is written into the Deformatter output buffer at a 32 MHz rate. It is read at the same rate, but with the addition of the four cycle gap. In the 4.128 msec interval, there are 256 FFT cycles. Each FFT cycle contains 516 32 MHz clocks (256 x 516 x 31.25 nsec = 4.128 msec.).

There are 256 x 516 bits written into the circular buffer which is 16.125 times the 8192 physical bits contained in the ram. The buffer is written 16 complete times plus an additional 1024 bits. When NEW_DELAY occurs, the write pointer is reset back to zero, which means it backs up 1024 bits. Since the mode is 1-4, the 1024 bits come from four tracks, so each track needs to be backed up 256 bits. When we assume that there is no change in the integer delay (discussed later), the resetting of the write pointer to zero coincides exactly with the backup of 256 bits in the track data streams, so that the last 1024 bits that were just written are re-written again in the exact same locations. If the integer delay has changed, the data will be re-written with a possible offset of plus or minus one location. This is because the integer delay will change by at most plus or minus one bit in a 4 msec interval. (This was the case before Orbiting VLBI (VSOP) came along. It is possible that higher delay rates will occur for the spacecraft, so delays may change by more than one bit in those experiments.)

Looking at a second example, in 1 bit stream to 2 track mode, 512 point FFT, non-oversampled, the data is written into the Deformatter output buffer at a 16 MHz rate. In the 4.128 msec interval, there are half as many bits written as in the previous example, or 8.0625 times the 8192 physical bits contained in the ram. Thus the buffer is written 8 complete times plus an additional 512 bits. When NEW_DELAY occurs, the write pointer is reset back to zero, backing up 512 bits. Since the mode is 1-2, the 512 bits come from two tracks, so each track needs to be backed up 256 bits again. This backup of 256 bits on each track is a constant for all modes. The following table provides tabulation of several examples, including the two given above.

Table 10 DEFORMATTER BUFFER RAM ACTIVITY

B-T MODE	OV LP	FFT SIZE	WRITE RATE	NR BITS WRITTEN	NR FFTs	NR PASSES ARND BUF	EXTR BITS	BITS PER TRACK	OVS FACT	BITS THROWN AWAY
1-4	1	512	32MHz	132,096	258	16.1250	1024	256	1	0%
1-2	2	512	16MHz	66,048	129	8.0625	512	256	1	0%
1-1	4	512	8MHz	33,024	64.5	4.03125	256	256	1	0%
2-1	8	512	4MHz	16,512	32.25	2.015625	128	256	1	0%
4-1	16	512	2MHz	8,256	16.125	1.0078125	64	256	1	0%
1-1	8	512	4MHz	16,512	32.25	2.015625	128	256	4	50%
1-1	8	1K	4MHz	16,512	32.25	2.015625	128	256	1	0%

(note 1K bit stream is written to a pair of buffers)

B-T MODE is the bit-to-track mode. OVLP is the overlap factor that results from the mode (discussed below). NR_FFTS times 512 bits per FFT equals the NR_BITS_WRITTEN. NR_FFTs divided by 16 FFTs per one pass around the buffer equals the NR_PASSES_ARND_BUF. The fractional part of NR_PASSES_ARND_BUF times 8192 equals the EXTR_BITS, which is how far past address zero the write pointer has reached when NEW_DELAY causes it to reset to zero.

There is a complication relating to the back up of 256 bits in the track bit streams and barrel roll. When un-rolling, the Deformatter is required to update the un-roll action every tape frame. If the backup of 256 bits crosses a tape frame boundary, then the un-roll action needs to revert to the previous un-roll state and then switch back to the new un-roll state when the boundary is crossed again. In addition, data validity is declared on frame boundaries, so the backup across a frame boundary requires extra consideration in the data validity logic. The barrel roll Xilinx, U128 on sheet 16, and the data invalid Xilinx, U40 on sheet 17, handle the frame boundary crossing situations. U128 also protects against a double interrupt from the NEW_FRAME event in the case where the back up of 256 bits crosses a tape frame boundary. In such a case, the NEW_FRAME interrupt is only asserted once.

When an oversample factor of 2 is used, the buffer ram is read out twice, skipping every other bit, covering a span of 1024 bits instead of 512. During the first pass, the EVEN samples (0, 2, 4 etc.) are read. During the second pass, the ODD samples are read. For oversample factors higher than 2, bits are discarded. For example, with an oversample factor of 8, samples S0 and S4 would be retained (as the EVEN and ODD samples respectively) and samples S1, S2, S3, S5, S6 and S7 would be discarded.

The output bit streams from the Deformatter to the FFT card are clocked at 32 MHz in all cases. In the cases where the aggregate data rate into the buffer ram is less than 32 MHz, data is read out multiple times in an overlapped manner. Overlap factors range from 1 to 64 across the mode table. Overlap factors are larger for 1K and 2K FFT sizes since the bit streams are processed in parallel FFT pipelines.

For FFT sizes of 512, 1K or 2K the overlap sequences are straight forward. For an overlap factor of 2, the sequence backs up half way through the previous buffer. For an overlap factor of 4, the sequence backs up to the one quarter point in the previous buffer.

For FFT sizes of less than 512, the overlap sequences are more complicated. For example, when doing 256 point FFTs, the FFT buffer of 512 points has two time ordered sets of 256 points. The FFT pipeline operates on both sets of points in a single FFT hardware cycle, to produce two sets of 128 spectral points at the FFT output. For an overlap factor of 2, the Deformatter thus needs to back up to the halfway point of the first set of 256 points, which is the one quarter position in the previous 512 bits.

The control of the overlap and oversample sequences is handled in part by Sequencer 3 (see section 5.4.3).

5.3.5) Buffer Output and Special Polar Mode #2

The 16 MHz output streams from the buffer rams are re-combined in the PALs seen on the right side of Figure 7 (MUX U100 etc.) and on sheet 14 of the schematic.

The SWAP\ signal seen on sheet 14, going to pin 23 of each of U100, U73, U97 and U71 provides support for Special Polar Mode #2. This mode is when 1 bit stream to 4 track mode is used with 2 bit sampling and polarization. Since 1-4, 2 bit mode requires 8 tracks, then one polarization channel fully uses one PBI delay center. Normally, one delay center drives two adjacent FFT pipelines.

Polar mode requires that the two different polarizations be processed in adjacent pairs of FFT pipelines (0 and 1, 2 and 3 etc.).

Looking at Figure 7, Delay Center 0, U86, it can be seen that the 8 tracks being processed would normally be the only source of data to FFT Engines 0 and 1. Looking at schematic sheet 14, it can be seen that the four PALs driven by the SWAP\ signal each handle two FFT pipelines (or engines).

For example, U100 handles FFT pipelines 0 and 2, while U97 handles pipelines 1 and 3. The SWAP\ signal is used only at U97 and U71, in order to swap the pairs of outputs in these chips. (The signal was a spare signal that was routed to all four PALs, but the swap function is only done in two of the PALs.)

Figure 13 SPECIAL POLAR MODE #2 shows an example of the SWAP function. Delay Center 0 handles the RIGHT (R) polarized signal and Delay Center 1 handles the LEFT (L) signal. R goes to FFT Engine 0, and with the SWAP\ signal asserted, L ends up going to FFT Engine 1, since the PAL at U97 on schematic sheet 14 will swap 1 with 3. Likewise, the PAL at U71 will swap 5 with 7.

5.4.0) Sequencers

The three ram based sequencers are identified as sequencers 1, 2 and 3. These sequencers provide many of the control signals on the Deformatter Card and are described in more detail below.

The sequencer files that are downloaded over the HCB to the Deformatter are Motorola S record hex files. The sequencer filenames are derived from the following sets of characters:

s1, s2 or s3	Sequencer number
n,v,w,x or y	Oversample factors (n=1, v=2, w=4, x=8, y=16)
1-4, 1-2, 1-1, 2-1, or 4-1	Bit-to-Track mode
b,c,d,e,f or g	FFT size (b=64, c=128, d=256, e=512, f=1K, g=2K)

For example, the three files for 1-4, 2K non-oversampled mode are:

```
s1n1-4g.hex
s2n1-4g.hex
s3n1-4g.hex
```

Many modes use identical sequencer files. Thus every filename derived in the above manner will not exist as an actual file. The real time system software function defWrSequencer() will load the correct set of files in all cases.

The sequencer files are derived from source files (plain text), using the same type names as above, where the extension is .dat instead of .hex. The vlbsoft/defseq directory contains the .dat source files and a Makefile that controls the processing of the source files. This directory also contains a file named tables.txt that provides some details of the bit definitions in the sequencers, and some tabulations of the various bit patterns used in the sequencers.

The .dat files are processed by an executable named genbits. The source code for this program is genbits.c. The program treats all sequencers as if they are 16 bits wide. The .dat files define how deep the sequencers are. All sequencer 1 and 2 files are defined over the address range 0-2047. All sequencer 3 files are defined over the address range 0-255. The pseudo language used in the .dat files is defined in a comment section of genbits.c.

5.4.1) Sequencer 1

Sequencer 1, which is 16 bits wide, is seen on sheet 19. The ram storage for this seqencer is provided by U153 and U186. The address to the rams cycles from 0 through 1031, covering two FFT cycles plus two four cycle gaps. The FFT_INIT pulse from the Master Control Card synchronizes the address counter for Sequencer 1 once every 1032 32 MHz clocks (once every other FFT cycle). No one presently remembers a precise reason for the sequencer to run through two FFT cycles instead of just one. The best guess is that it may have something to do with oversampling, where the EVEN/ODD pairs of FFT cycles are done.

U154 and U187 provide the data path for loading the sequencer contents. U155, U190 and U188 capture the sequencer ram outputs. The 68K_SEL\23 control line to U152, U170 and U185 asserts low for loading the ram. This signal also jam sets U191A at pin 4 in order to prevent the FFT_INIT pulses from clearing the counters while they are being used for loading the sequencer rams. The functions of sequencer ram bits 0-15 are defined as follows:

Table 11 DEFORMATTER SEQUENCER 1 BIT DEFINITIONS

Ram Bit	IC-PIN	DESCRIPTION
0	U153-9	8 MHz clocks for DEF; buffered by U155 followed by U90 and U37
1	U153-10	8 MHz clocks to TRC; buffered by U155
2	U153-11	One bit of a two bit pseudo random test pattern sequence; TEST_DATA_S (sign) bit (MS bit of two bit sample)
3	U153-13	16 MHz clock to TRC; buffered by U155; 16 MHz clocks to DEF; buffered by U155/U158; pipelined by U191 for phase shifted versions
4	U153-14	Other bit of the two bit pseudo random test pattern sequence TEST)DATA_M (magnitude) bit (LS bit of two bit sample)
5	U153-15	NEW_FRINGE pulse to the buffer ram address generator; gated with NEW_FRINGE_EN\ at U156
6	U153-16	NEW_FRINGE pulse to the data valid logic; gated with NEW_FRINGE_EN\ at U156
7	U153-17	131 msec. tic; gated with c131msTIC_EN at U156; result sensed by 68K processor to control the modulo 32 NEW_DELAY counter operations; delayed result used at U71 (sheet 14) to select the test pattern for self test (not currently supported)
8	U186-9	NOT USED; signal name is FFT_TEST going into PAL U189; D009T16.ABL does not use this input;
9	U186-10	DSEL0
10	U186-11	DSEL1 These two bits are decoded in U189 to generate NEW_DELAY pulses: 0 = none; 1 = TRC; 2 = Barrel Roll 3 = Demux-Delay
11	U186-13	NEW_DELAY to buffer ram address generator
12	U186-14	NEW_DELAY to READ/WRITE sequencer
13	U186-15	NEW_DELAY to Data Invalid PALs on sheet 14 (U62, U63)
14	U186-16	ND_NF_INTERVAL_INVALID\ to Data Invalid PALs on sheet 14 (used in certain modes to declare data invalid for the entire interval between NEW_DELAY and NEW_FRINGE)
15	U186-17	NEW_FFT\ another FFT_INIT pulse

On sheet 19, the BIT0 - BIT14 labels seen to the right of U155 and U188 are not the same as the sequencer bits 0-15. When looking at the

tables.txt file referred to above, sequencer bit numbers are used, not the BIT0 - BIT14 labels. U153 pin 9 is sequencer bit 0. U186 pin 17 is sequencer bit 15.

Although the two pseudo random test pattern bits (2 and 4) are addressed with 32 MHz resolution, they are captured by a 16 MHz clock on sheet 14 (at U99, U72, U96 and U70), so they effectively have only 16 MHz resolution at the output of the Deformatter.

5.4.2) Sequencer 2

Sequencer 2, which is 8 bits wide, is seen on sheet 18. The ram storage for this sequencer is provided by U166. The address to the ram cycles from 0-2047, overflowing multiple times during a 4 msec. interval. It is reset once per 4 msec. by the RW_NEW_DELAY\ signal from Sequencer 1.

U165 provides the data path for loading the sequencer contents. U164 captures the ram outputs. The functions of sequencer ram bits 0-7 are defined as follows:

Table 12 DEFORMATTER SEQUENCER 2 BIT DEFINITIONS

Ram Bit	IC-PIN	DESCRIPTION
0	U166-9	Spare bit, unused
1	U166-10	CAP_CLK: captures the data at the input to the Bit Shuffler Xilinx
2	U166-11	CLK_EN: clock enable to the serial shift register in the Bit Shuffler Xilinx
3	U166-13	WRITE_ADR_CNT_EN\: clock enable to the write address counter for the buffer ram
4	U166-14	R/W0 R/W[0..3] are gated in U95 (sheet 18) to generate the buffer ram write pulses
5	U166-15	R/W1
6	U166-16	R/W2
7	U166-17	R/W3

5.4.3) Sequencer 3

Sequencer 3, which is 12 bits wide, is seen on sheet 13. The ram storage for this sequencer is provided by U168, U137 and U138. The address to the ram cycles from 0-255, counting the 256 FFT cycles in a 4 msec. fringe cycle interval.

The lower 8 bits of this sequencer, stored in U137 and U138, is the "CAPTURE ADDRESS" field. This is the address that we need to store so that we can back up when doing overlapped FFTs. This field is loaded into a down counter (U136). When the counter reaches zero, the actual

address that we need to store is asserted in the read address counter (U135, U134), and a snapshot is stored in registers in the counter so that we can rewind to the appropriate address.

Bit 8, U168-10, goes to U189 on sheet 19. It is used in developing the various NEW_DELAY signals output by U189.

Bit 9, U168-12, is a spare unused bit.

Bit 10, U168-14, is a spare unused bit.

Bit 11, U168-16, is asserted low for overlap control.

Provisions can be seen on sheet 13 for applying an offset address to the odd banks of the buffer ram. This feature was intended to be used to support overlapped FFTs in adjacent FFT pipelines, but has never been used. The ram read offset is presently always programmed to zero in HCB protocol function 0x34.

5.5.0) 68K Microprocessor

There are two RESET inputs for re-setting the 68K microprocessor, but only one is used in the system. Sheet 20, U209 pins 2 and 5 receive the reset from the HCB and this is the only reset path wired up in the system. U209 pins 3 and 6 come from the card edge pin 12, which is wired to ground (not asserted) in the system. This input is available for power on or push button resets in test fixtures.

Originally the 68000 uP ran from a crystal oscillator, U222. The clock frequency was either 12.0 or 12.5 MHz. The clock frequency was later changed to 16.0 MHz in an attempt to reduce the number of times that the NEW_DELAY interrupt routine did not complete on time. Problems then developed with loading of sequencer 3, and the crystal clock source was removed from the U222 socket, and a wire added to U192-18 (sheet 19), to provide a synchronous 16 MHz clock to the 68K. This means that in the original PBI test fixture, where the system clock can be varied from 20 to 40 MHz, there is the possibility that the 68K (a 16.67 MHz part) may not operate reliably much above the 32 MHz setting in the test fixture.

Sheet 20 is the "root" sheet for the 68K portion of the Deformatter Card. Sheet 21 has address and data bus buffers. Pin 2 of U150 on sheet 21 experiences bus contention with pin 34 of the Barrel Roll Xilinx (U128 on sheet 16) after power on and prior to download of U128. This problem is overcome by brute force, using a 74F device at U150. The data buffering on sheet 21 includes the 8 bit parallel I/O bus used to communicate with the four 87C51 processors on the two TRC cards.

Sheet 22 has the address decoder and a table that outlines the memory map for the processor. This table is not quite in address order. Sheet 23 has the interrupt generation and discrete input sense logic. Possible interrupt sources are defined in the 68K IPL Source Table on sheet 23, where MASTER_IRQ is the interrupt from the HCB.

Sheet 24 has the program and data RAM storage, organized as 128K 16 bit words. Sheet 25 has the program PROM storage, organized as 64K 16

bit words. The 68K source code is maintained in the vlbsoft/defasm directory. A Makefile controls the processing of the source files. A single Motorola s-record file is generated by the linker. This file is then processed twice by the sRecExtract program in order to separate out the ROM and RAM based blocks of code into the defrom.hex and defram.hex files. The RAM code is all code between the addresses 0x1500 and 0x85ffff inclusive. The ROM code is all code above and including 0x860000.

Sheet 26 has the serial registers that receive the 16 bit integer delay model offsets every 4 msec. There is one delay word for each of the four delay centers. Sheet 26 has multiplexer U34 to select between two possible sources of delay words. Delay words are shifted from a FFT Control Card (FCC). The Deformatter must receive the delay model offset from the FCC that controls the FFT card connected to the Deformatter output via the crossbar. In most cases, the Deformatter will always be configured to select the "NORMAL" source (using HCB function code 0x3B). Only the Deformatters connected to PBD 1, 12, 13 and 24 will ever need to select the "ALTERNATE" source. This can be seen by looking at Figure 11 PBD TO FFT CROSSBAR. Each PBI connects to either three or four FFT cards. When the Deformatter connected to PBD 1 is routed to FFT 0, the integer delay comes from FCC 0 (normal). When the Deformatter connected to PBD 1 is routed to FFT 19, the integer delay comes from FCC 1 (alternate) since FCC 1 handles FFT 19. In general, anytime the line from DEF to FFT in Figure 11 crosses the heavy center line, then the DEF will need to select the "ALTERNATE" source for the integer delay input.

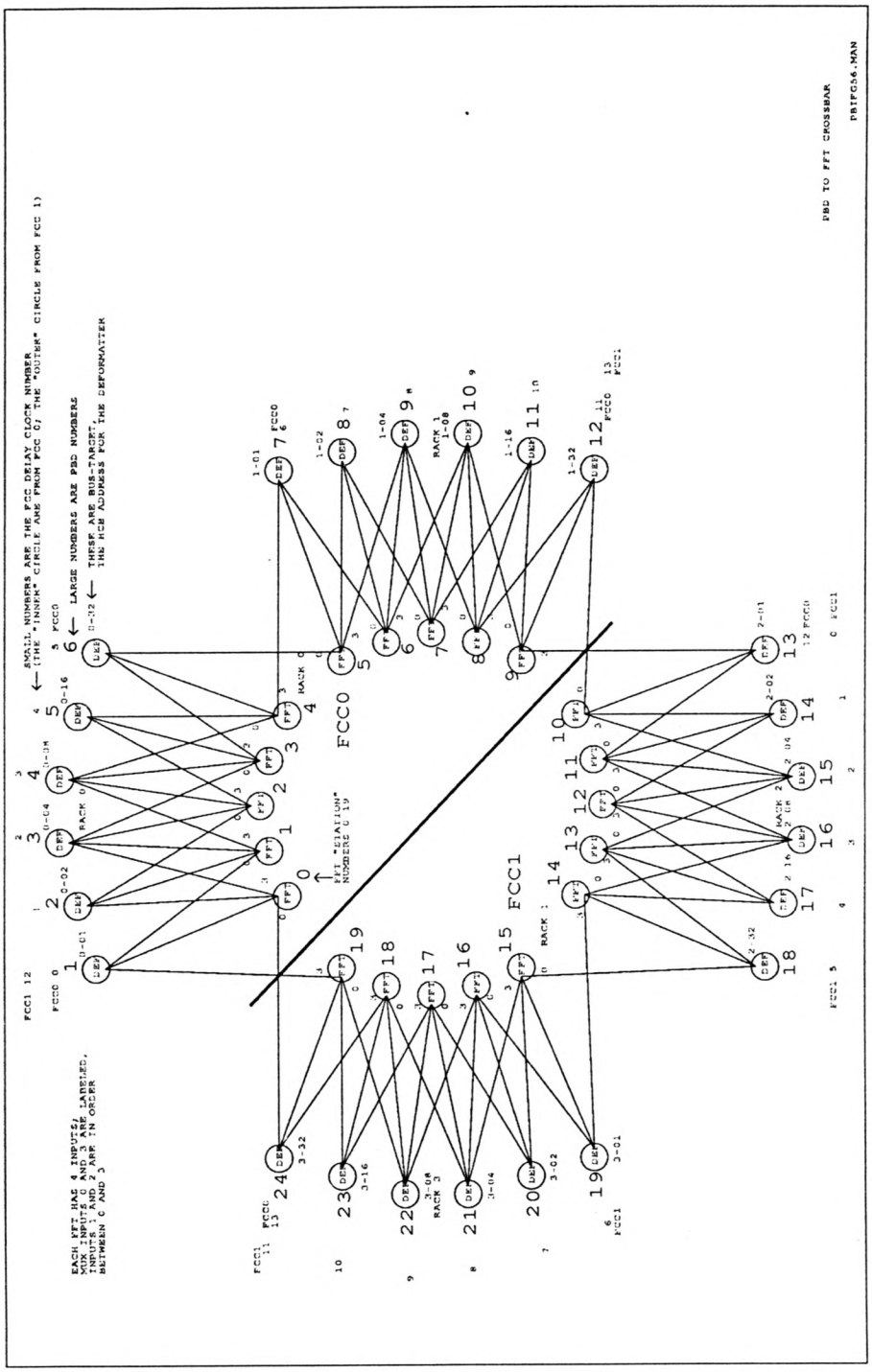


Figure 11 PBD TO FFT CROSSBAR

While looking at the crossbar, this is a good time to be reminded that FFT are counted 0-19 while PBD are counted 1-24. Since PBD are connected directly to PBI, we now tend to count PBI as 1-24 also, when not referring to them by the long standing bus-target numbers (0-01, 0-02, 0-04, 0-08, 0-16 and 0-32 for example in rack 0). The bus-target numbers refer to the HCB bus number and HCB target mask used to address targets over the HCB.

Sheet 28 has the serial output logic for the 68K. This is used to shift serial control words to various parts of the Deformatter.

Sheet 29 has the HCB interface over which the 68K communicates with the VME system.

Sheet 30 has the two asynchronous serial communications ports. One is used for running the debug and monitor utilities on a 9600 baud serial terminal. The other is used to send serial servo commands to the PBD to speed up or slow down the tape.

5.6.0) Data Validity

For every FFT cycle of data processed, the Deformatter is required to declare if the data is valid or invalid. There are several reasons why data may be declared invalid:

1) Quality of playback as determined by the TRC

2) The state of the tape servo
   (required data is not in TRC buffer)

3) Some number of FFT cycles declared invalid every
   4 msec (number depends on mode); see section 5.6.1

4) Data forced invalid by HCB function code 0x40

5) Some FFT cycles declared invalid every tape frame
   in Mark III / IV modes due to header pollution;
   see section 5.6.2

5.6.1) FFT Cycles Declared Invalid Every 4 msec.

The Deformatter declares some number of FFT cycles invalid every 4 msec., where the number is between 0 and 128 inclusive. There are two mechanisms that determine how many FFT cycles are declared invalid. Both of these mechanisms are handled by the PALs U62, U63 on sheet 14, and are a function of the observing mode.

The first mechanism involves overlapped FFT operations, in modes where the overlap factor is less than 32. In the interval between NEW_DELAY and NEW_FRINGE, if an overlap read operation crosses the NEW_DELAY boundary (address 0 in the buffer), reading out data that has been written since the NEW_DELAY event, then the data is declared

invalid. If the integer delay has not changed, the data would not actually be invalid. The data is still declared invalid due to the possibility that the delay may have changed.

The second mechanism involves modes where less than one full circle of the 8192 bit buffer is written during the 4 msec. interval between two NEW_DELAY events. In these cases, overlap operations read beyond where data was written. The only mechanism available for handling this situation is to declare all FFTs between NEW_DELAY and NEW_FRINGE invalid. This only happens in modes where the overlap factor is 32 or 64.

5.6.2) FFT Cycles Declared Invalid Every Tape Frame (Mark III / Mark IV Modes)

The VLBA tape format is a non-data replacement format, so the header does not replace data. Mark III and Mark IV tape formats replace data with the header. The PBI design allows header bits to be present at the PBI output in these non-VLBA formats. This pollution of data with header bits was not thought to be a problem, since the headers of station pairs should not correlate. It was later determined that the station spectra show artifacts from the header bits.

The data validity logic normally operates on frame boundaries. In order to remove the effect of the header bits, an alternate Data Invalid Xilinx design was generated that provides the capability of gating data on FFT cycle boundaries. At each NEW_FRAME, data is forced invalid. After three NEW_FFT pulses occur, the forcing function is removed. In 1 bit stream to four track mode, the 160 header bits on each of the four tracks at a 8 MHz rate produce 640 32 MHz bits out of the Deformatter. Since this is more than the 512 bits in a single FFT cycle, it is possible the header bits could totally cover one FFT cycle, and partially cover two other FFT cycles. This is the basis for waiting until 3 NEW_FFT pulses have passed before un-gating the data validity output.

The Xilinx hex file for VLBA modes is named defctrl.mcs. The alternate file for Mark III/ IV is named mk3ctrl.mcs.

5.6.3) Data Validity and Low Bandwidth Mode

After it was realized that some modes would have very little (or no) valid data, the PBI was modified to support what is now called Low Bandwidth Mode. In this mode, if the integer delay has not changed, the NEW_DELAY pulse can be skipped for up to a maximum of six 4 msec. intervals. As long as the integer delay has not changed, it is not necessary to declare FFT cycles invalid due to either of the mechanisms in section 5.6.1. For each NEW_DELAY that is skipped, the TRC must eventually be backed up 256 track bits. The limit of 6 skips results from the Barrel roll Xilinx design, where a counter must keep track of how many multiples of 256 bits are skipped, in order to look for possible NEW_FRAME events in the region of backup. (See section 5.8.1 for more detail.)

Since the time when Low Bandwidth Mode was developed, the worst case modes that inspired this mode were found to not be possible for other reasons. At the present, Low Bandwidth Mode has never been used.

In order to see if this mode might still be of use, we need to consider both of the mechanisms for declaring data invalid described above. In the first situation, there are a total of 67 modes where the PBI will declare 0 or more FFTs invalid every 4 msec. due to overlapped FFTs crossing the possible discontinuity at address 0 in the Deformatter buffer. The following list summarizes the number of modes for each different number of invalid FFTs:

Table 13 INVALID FFT CYCLES FOR MECHANISM 1

Nr of modes	Nr of invalids
8	0
5	1
5	2
6	3
6	6
6	7
10	14
6	15
15	30
---	
67 modes	

In the above cases, every bit gets into at least one FFT. The FFTs declared invalid are overlapped FFTs.

There are 32 modes where the PBI will declare either 32, 64 or 128 FFTs invalid every 4 msec (if low bandwidth mode is not turned on) due to overlapped FFTs reading beyond the point where data has been written in the Deformatter buffer. (These are cases where fewer than 8192 bits are written in 4 msec.) In these cases, the hardware will invalidate some FFTs that were actually valid. These cases are summarized below:

Table 14 INVALID FFT CYCLES FOR MECHANISM 2

Nr of modes	Nr of invalids declared	Nr of invalids that were actually valid	Nr of valid ffts per each 256 (4 msec)	Nr of ffts retained
2	32	3	227	224
3	64	18	192	174
3	64	10	192	182
3	64	6	192	186
8	64	4	192	188
1	64	3	192	189
2	128	10	128	118
2	128	6	128	122
8	128	4	128	124
32 modes				

An attempt was made to determine what percentage of the valid data written into the Deformatter buffer actually get into at least one FFT. The following list tabulates these results and is presently thought to be reasonably accurate.

Table 15 PERCENTAGE OF BITS USED

pct bits used	nr of modes	
62	2	128, 2-1, ovs16; 128, 4-1, ovs8;
74	2	256, 2-1, ovs16; 256, 4-1, ovs8;
78	3	64, 1-1, ovs16; 64, 2-1, ovs8; 64, 4-1, ovs4;
81	3	128, 1-1, ovs16; 128, 2-1, ovs8; 128, 4-1, ovs4;
87	5	256, 1-1, ovs16; 256, 2-1, ovs8; 256, 4-1, ovs4; 1K, 4-1, ovs1; 2K, 2-1, ovs1;
99	17	(too many to list here)
	32 modes	

The worst case is seen to be two modes where 38% of the available data is actually discarded. Based on the above, there has been no rush to actually utilize Low Bandwidth Mode.

The following two tables list the number of invalid FFTs due to the above two mechanisms, for all modes, when Low Bandwidth Mode is not being used.

Table 16 INVALID FFTs, FFTSIZE >= 512

NUMBER OF INVALID FFTS FOR EACH MODE WHERE FFTSIZE >= 512
(This is the number of FFT cycles declared invalid when playback is perfect.)

FFT SIZE	B-T MODE	OVER SAMP	FFTs ND-NF	OVER LAP	WR BIT RATE	BITS WRITTEN IF < 8K	NR FFT INVALID EACH 4 MSEC.		
512	1-4	1	2	1	32		0		OVERSAMP is the
512	1-4	2	2	1	32		0		oversample factor
512	1-2	1	2	2	16		1		
512	1-2	2	4	2	16		2		
512	1-1	1	4	4	8		3		OVERLAP is the
512	1-1	2	8	4	8		6		overlap factor
512	1-1	4	16	8	4		14		
512	1-1	8	32	16	2		30		
512	1-1	16	64	32	1	0-4127	60	(*)	
512	2-1	1	8	8	4		7		2-1 and 4-1 bit-
512	2-1	2	16	8	4		14		to-track modes no
512	2-1	4	32	16	2		30		longer supported
512	2-1	8	64	32	1	0-4127	60	(*)	
512	2-1	16	128	64	0.5	0-2063	124	(*)	
512	4-1	1	16	16	2		15		
512	4-1	2	32	16	2		30		
512	4-1	4	64	32	1	0-4127	60	(*)	
512	4-1	8	128	64	0.5	0-2063	124	(*)	
1K	1-4	1	2	2	16		1		
1K	1-4	2	4	2	16		2		
1K	1-2	1	4	4	8		3		
1K	1-2	2	8	4	8		6		
1K	1-1	1	8	8	4		7		
1K	1-1	2	16	8	4		14		
1K	1-1	4	32	16	2		30		
1K	1-1	8	64	32	1	0-4127	60	(*)	
1K	1-1	16	128	64	0.5	0-2063	124	(*)	
1K	2-1	1	16	16	2		15		
1K	2-1	2	32	16	2		30		
1K	2-1	4	64	32	1	0-4127	60	(*)	
1K	2-1	8	128	64	0.5	0-2063	124	(*)	
1K	4-1	1	32	32	1	0-4127	29	(*)	
1K	4-1	2	64	32	1	0-4127	60	(*)	
1K	4-1	4	128	64	0.5	0-2063	124	(*)	
2K	1-4	1	4	4	8		3		
2K	1-4	2	8	4	8		6		
2K	1-2	1	8	8	4		7		
2K	1-2	2	16	8	4		14		
2K	1-1	1	16	16	2		15		
2K	1-1	2	32	16	2		30		
2K	1-1	4	64	32	1	0-4127	60	(*)	
2K	1-1	8	128	64	0.5	0-2063	124	(*)	
2K	2-1	1	32	32	1	0-4127	29	(*)	

2K	2-1	2	64	32	1	0-4127	60 (*)
2K	2-1	4	128	64	0.5	0-2063	124 (*)
2K	4-1	1	64	64	0.5	0-2063	61 (*)
2K	4-1	2	128	64	0.5	0-2063	124 (*)

(*) Indicates modes where fewer than 8192 bits are written in the 4 msec interval; in these cases the complete interval between ND and NF is declared invalid so the number of invalid FFTs shown needs to be increased to the next binary value (32, 64 or 128).
The BITS WRITTEN column shows the portion of the 8K output buffer that is written each 4 msec, if less than 0 - 8191.

The FFTs ND-NF column shows the number of FFT cycles between the start of NEW_DELAY and the start of NEW_FRINGE.

WR BIT RATE is the rate (in MHz) at which bits are written into the output buffer.

Table 17 INVALID FFTs, FFTSIZE < 512

NUMBER OF INVALID FFTS FOR EACH MODE WHERE FFTSIZE < 512
(This is the number of FFT cycles declared invalid when playback is perfect.)

FFT SIZE	B-T MODE	OVER SAMP	FFTS ND-NF	OVER LAP	WR BIT RATE	BITS WRITTEN IF < 8K	NR FFT INVALID EACH 4 MSEC.	
64	1-4	1	2	1	32		0	OVERSAMP is
64	1-4	2	2	1	32		0	the oversample
64	1-2	1	2	2	16		1	factor
64	1-2	2	4	2	16		2	
64	1-1	1	4	4	8		3	OVERLAP is the
64	1-1	2	8	4	8		6	overlap factor
64	1-1	4	16	8	4		14	
64	1-1	8	32	16	2		30	
64	1-1	16	64	32	1	0-4127	46	(*)
64	2-1	1	8	8	4		7	2-1 and 4-1 bit
64	2-1	2	16	8	4		14	to track modes
64	2-1	4	32	16	2		30	not supported
64	2-1	8	64	32	1	0-4127	46	(*)
64	4-1	1	16	16	2		15	
64	4-1	2	32	16	2		30	
64	4-1	4	64	32	1	0-4127	46	(*)
128	1-4	1	2	1	32		0	
128	1-4	2	2	1	32		0	
128	1-2	1	2	2	16		1	
128	1-2	2	4	2	16		2	
128	1-1	1	4	4	8		3	
128	1-1	2	8	4	8		6	
128	1-1	4	16	8	4		14	
128	1-1	8	32	16	2		30	
128	1-1	16	64	32	1	0-4127	54	(*)
128	2-1	1	8	8	4		7	
128	2-1	2	16	8	4		14	
128	2-1	4	32	16	2		30	
128	2-1	8	64	32	1	0-4127	54	(*)
128	2-1	16	128	64	0.5	0-2063	118	(*)
128	4-1	1	16	16	2		15	
128	4-1	2	32	16	2		30	
128	4-1	4	64	32	1	0-4127	54	(*)
128	4-1	8	128	64	0.5	0-2063	118	(*)
256	1-4	1	2	1	32		0	
256	1-4	2	2	1	32		0	
256	1-2	1	2	2	16		1	
256	1-2	2	4	2	16		2	
256	1-1	1	4	4	8		3	
256	1-1	2	8	4	8		6	
256	1-1	4	16	8	4		14	
256	1-1	8	32	16	2		30	
256	1-1	16	64	32	1	0-4127	58	(*)
256	2-1	1	8	8	4		7	

FFT SIZE	B-T MODE	OVER SAMP	FFTS ND-NF	OVER LAP	WR BIT RATE	BITS WRITTEN IF < 8K	NR FFT INVALID EACH 4 MSEC.	
256	2-1	2	16	8	1		14	
256	2-1	4	32	16	2		30	
256	2-1	8	64	32	1	0-4127	58	(*)
256	2-1	16	128	64	0.5	0-2063	122	(*)
256	4-1	1	16	16	2		15	
256	4-1	2	32	16	2		30	
256	4-1	4	64	32	1	0-4127	58	(*)
256	4-1	8	128	64	0.5	0-2063	122	(*)

(*) Indicates modes where fewer than 8192 bits are written in the 4 msec interval. In these cases the complete interval between ND and NF is declared invalid so the number of invalid FFTs shown needs to be increased to the next binary value (64 or 128). The BITS WRITTEN column shows the portion of the 8K output buffer that is written each 4 msec, if less than 0 - 8191.

The FFTs ND-NF column shows the number of FFT cycles between the start of NEW_DELAY and the start of NEW_FRINGE. WR BIT RATE is the rate (in MHz) at which bits are written into the output buffer.

5.7.0) Tape Servoing

It is the responsibility of the VME system to position the tape within plus or minus 1,000,000 bits of the desired bit. Once the tape is positioned in this range, the responsibility for tape positioning is normally turned over to the Deformatter which then attempts to maintain the tape position within a block of 200,000 bits.

The Deformatter handles this responsibility by processing the lowest level of the delay model. Refer to Appendix I of the HCB protocol for the Deformatter (D027DEF.HCB) for a detailed description of how delay is set in the system. (In a very small nutshell, the Deformatter tracks the number of Nyquist samples since midnight as a linear ramp and applies the integer delay that was received from the FCC as a signed offset to this value to identify the specific bit required by the model. This operation occurs once every 4 msec.)

As described in the above reference, the Deformatter calculates two track bit numbers every 4 msec. Both are track bit numbers referenced from midnight of the current day. One is the bit number that the delay model requires be output at the start of the next 4 msec. delay cycle. The other is the first bit in the tape frame whose time stamp was last reported by the TRC, which will be close to being the most recent bit written into the TRC buffer. This buffer contains approximately 200,000 track bits, and the time stamp is reported every 4 msec. The difference between the model bit number and tape bit number is called the calculated offset, in units of track bits.

The DEF servos the tape position such that the calculated offset is on the order of -100,000 track bits. This maintains the position of the target bit near the center of the TRC buffer.

The DEF uses a servo table that is downloaded by the VME system. The calculated offset is used as a pointer into this table. Based on the pointer into the table, the DEF will either send the servo command found in the table or not, based on a flag in the table. (When the calculated offset is very close to the center of the table, there is no need to send a servo command.) The DEF will also declare data invalid if the calculated offset is beyond limits specified by another servo table flag.

The servo table contains 256 entries of 10 bytes each. The start address of the table in 68K memory space can be identified by looking at the servo.lst assembly listing in the vlbsoft/defasm component.

5.7.1) Servo Table Format

The servo table format is defined in servo.asm as follows:

Table 18 SERVO TABLE FORMAT

offset from center	Timer 2 Bytes	Action Byte	Validity Byte	Speed 4 ASCII bytes	Duration 2 ASCII bytes
-128					
0					
+127					

Column description:

Timer: How often a servo command can be sent to the PBD.
Timer binary value = Duration time in units of msec/5.
The Deformatter will load this value into the ServoTimer after Sending the last character of a servo command to the PBD.
The DEF will not end another servo command until good time is received from the TRC and the ServoTimer has expired. The ServoTimer counts down on a 5ms tic, and the code looks for a neg value.

Action: If this byte is not 00 then DEF will not send servo commands to the PBD.

Validity: Used to control the Validity output on the DEF card. If this byte is 0 then the validity is determined by the TRC otherwise the validity is forced invalid by the Deformatter.

Speed: Four ASCII characters that will be sent to the PBD as parameters to the PBD BB command (speed).
The units are 0.01 ips and includes the normal reference speed.
Normal VLBA speed = 3e80h (160 dec * 100).
MK III speed = 6978h (270 dec * 100).
Max speed = 8ca0h (360 dec * 100)

Duration: Two ASCII characters that will be sent to the PBD as parameters to the PBD BC command (duration of the speed command). The units are 0.01s. Two hex characters give a max of 2.55 sec.

5.7.2) Servo Table Example

From the VME vxWorks prompt, the function dsplyServo(0) command can be run with the output re-directed to a file. This will produce a readable output file showing the contents of the servo table for the VLBA tape format.

The following listing is a fragment from the output of the above command (with comments added). The order is the same as in DEF memory.

Table 19 SERVO TABLE EXAMPLE

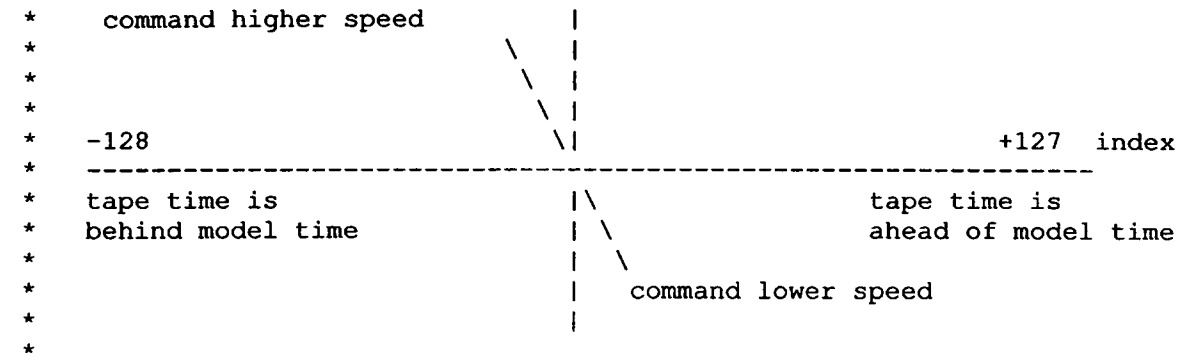
```
-- VLBA format table -- nominal speed = 158.75 ips

index timer  action valid  speed  duration pct_spd
-128   213     00    FF   193.52     53     21.8   <-- calculated offset
-127   213     00    FF   193.24     53     21.7       is near +1 million
-126   212     00    FF   193.62     52     21.9        (max speed up)
  |
  |
 -14    78     00    FF   166.69     25      5.0  (data inv above here)
 -13    74     00    FF   166.69     23      5.0  (edge of TRC buffer)
 -12    70     00    00   166.69     21      5.0  <-- calculated offset
 -11    66     00    00   166.69     19      5.0      is near 0;
 -10    62     00    00   166.69     17      5.0      need to speed up
  -9    58     00    00   166.69     15      5.0
  -8    56     00    00   166.69     14      5.0
  -7    52     00    00   166.69     12      5.0
  -6    48     00    00   166.08     11      4.6
  -5    44     00    00   165.37     10      4.1
  -4    40     00    00   164.59      9      3.6
  -3     5     FF    00   158.75      0      0.0
  -2     5     FF    00   158.75      0      0.0
  -1     5     FF    00   158.75      0      0.0
   0     5     FF    00   158.75      0      0.0  <-- calc offset is
   1     5     FF    00   158.75      0      0.0      near -100,000; no
   2     5     FF    00   158.75      0      0.0      servo action
   3    40     00    00   152.91      9      3.6
   4    44     00    00   152.13     10      4.1
   5    48     00    00   151.43     11      4.6
   6    52     00    00   150.81     12      5
   7    56     00    00   150.81     14      5.0
   8    58     00    00   150.81     15      5.0
   9    62     00    00   150.81     17      5.0
  10    66     00    00   150.81     19      5.0
  11    70     00    00   150.81     21      5.0   (data inv below here)
  12    74     00    FF   150.81     23      5.0 <-- calc offset is near
  13    78     00    FF   150.81     25      5.0     -200,000; need to
  14    80     00    FF   150.81     26      5.0     slowdown
  15    84     00    FF   150.81     28      5.0     (edge of TRC buffer)
  16    88     00    FF   150.81     30      5.0
  |
  |
```

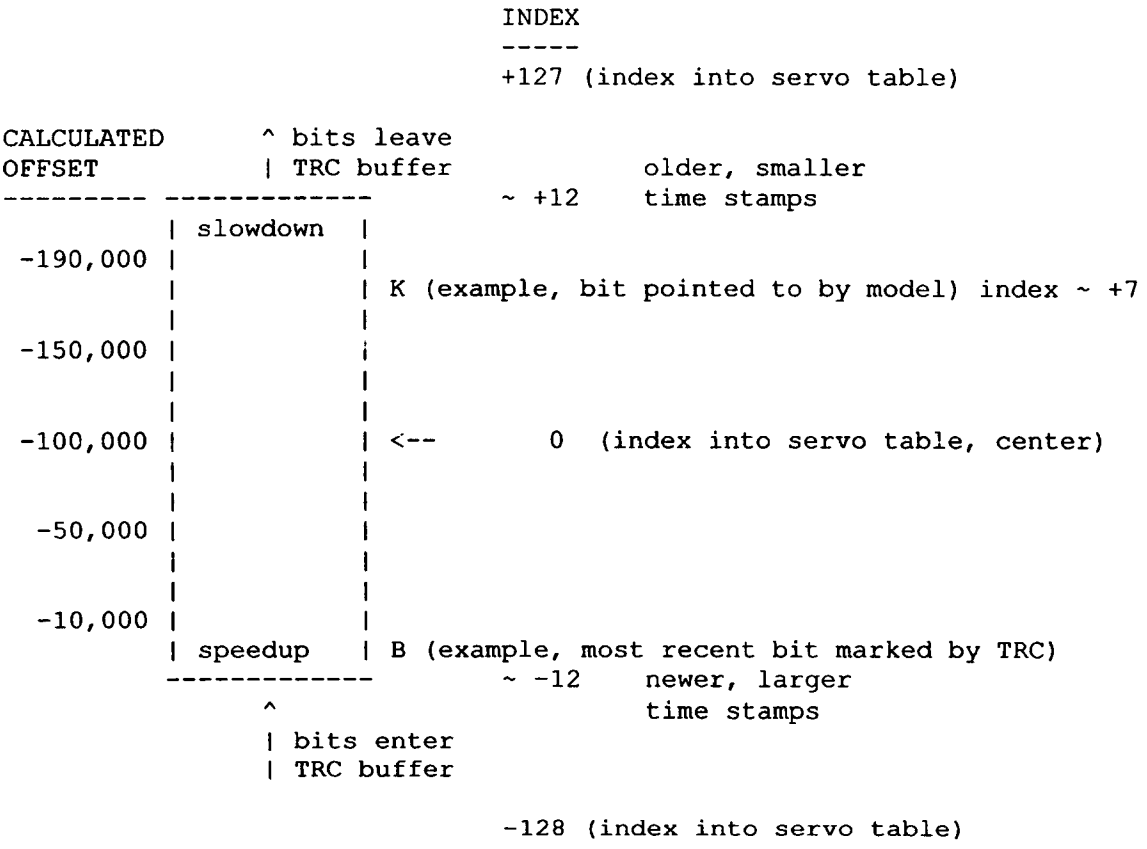
116	204	00	FF	125.09	50	21.2	
117	205	00	FF	125.46	51	20.9	
118	206	00	FF	125.18	51	21.1	
119	207	00	FF	124.90	51	21.3	
120	208	00	FF	124.62	51	21.5	
121	208	00	FF	124.34	51	21.6	
122	209	00	FF	124.71	52	21.4	
123	210	00	FF	124.45	52	21.6	
124	211	00	FF	124.17	52	21.7	
125	212	00	FF	123.89	52	21.9	
126	213	00	FF	124.28	53	21.7	
127	213	00	FF	124.00	53	21.8	<--calculated offset is near -1 million (max slowdown)

5.7.3) Servo Algorithm and Example

The following sketch is a simplistic view of the algorithm that relates tape time, model time, index and servo commands.



The next sketch shows the 200,000 bit TRC buffer in relation to a range of calculated offset values (-10,000 to -190,000) and in relation to the index into the servo table. A total range of plus and minus 1 million bits is divided into 256 blocks of 8192 bits to generate the pointer into the servo table. (2^21 divided by 256 = 8192) This range is centered on the middle of the TRC buffer. The limits of the TRC buffer are thus approximately plus and minus 12 on the index scale.



In Appendix I of D027DEF.HCB referenced above, K is the bit that the delay model requires be put out at the start of the next delay interval. B is the most recent bit identified by the TRC from the tape frame time code. The calculated offset is then generated by calculating (K - B). In the example above, let's assume the bit pointed to by the model is at a calculated offset near -160,000 (K). K-B will thus be on the order of -160,000, meaning that the bit of interest has been in the buffer for awhile and will soon exit the buffer. Thus the tape is running too fast and needs to be slowed down. The pointer into the table is generated from:

(-100,000) - (calculated offset) = (-100,000) - (-160,000) = 60,000

When 60,000 is divided by 8,192, we get an index of approximately +7 into the servo table. The speed and duration commands would be read from index 7 in the servo table and sent to the PBD.

5.8.0) Xilinx Designs

The Xilinx designs are:

U128	Barrel Roll Xilinx
U40	Data Invalid Xilinx
U86, U87, U52 and U51	Bit Shuffler Xilinx

The Barrel Roll and Data Invalid Xilinx chips are daisy chained together and downloaded from a single HEX file. Originally there was only one version of this file, defctrl.mcs. An alternate version, mk3ctrl.mcs now exists that can be used for blanking out the header bits from Mark III / IV formats. All design files for these two Xilinx chips are maintained in the vlbsoft/xilinx/defctrl/SCCS directory.

The four Bit Shuffler Xilinx chips are daisy chained and downloaded with identical personalities, from a total of ten mode dependent HEX file choices:

bsp0.mcs through bsp9.mcs.

Refer to Figure 9 for a list of the modes supported by each of the bit shuffler personalities.

BSP-0 and BSP-1 are recent additions, developed in the newer Xilinx tools, using Orcad schematic capture. The other designs, BSP-2 through BSP-9, and the Barrel Roll and Data Invalid designs use the older XACT tools based on macros. (BSP-9 is also a recent addition, but it did not use the newer tools.)

The design files for the ten BSP personalities are maintained in the ten vlbsoft/xilinx/bsp?/SCCS directories (bsp0 through bsp9).

The older DOS based Xilinx tools used for the designs are:

	Barrel Roll Data Invalid	BSP-7, BSP-8
XACT	Version 4.3	Version 4.12
MAKEBITS	Version 4.3	
MAKEPROM	Version 4.3	
APR	Version 3.3	Version 3.13

BSP-2, BSP-3, BSP-4, BSP-5, BSP-6 and BSP-9 did not use APR (Automatic Place and Route). The designs can probably be re-produced using XACT version 4.3. BSP-7 and BSP-8 can probably only be reproduced using the even older versions indicated above. Changes to BSP-7 or BSP-8 should be done in newer tools.

These tools require a hardware key. The Xact V4.3 and APR V3.3 tools are presently stored in the /home/azalea/pcstuff/xact directory.

The steps for generating the HEX files for the designs are usually listed in the top level macro for each design. The top level macros for all designs are:

Design	Top Level Macro
Barrel Roll	lbwrol.mac
Data Invalid	inv.mac OR mk3inv.mac
BSP0 and BSP1	bsp0prom.mac and bsp1prom.mac
BSP2 through BSP9	bsp-2.mac through bsp-9.mac

Additional files of possible interest are README.defctrl, lbwnotes.txt and invnotes.txt in the vlbsoft/xilinx/defctrl directory.

5.8.1) Barrel Roll Xilinx

The drawings for the internals of this Xilinx personality may be entered from sheet 1 of the Deformatter logic, via the K012D01.BLK entry point (see Volume 2).

There are 12 sheets numbered K012D01 through K012D12, with suffixes of either .BLK or .XIL. The .BLK suffix indicates the "representative logic" schematic. These are similiar to TTL type logic. The .XIL suffix indicates the "representative Xilinx logic block" schematic. These show actual CLB and IOB type symbols. Neither type of drawing is a precise definition of the design

Only the actual design macro files that are input to the Xilinx design tools provide the precise netlist and logic block configuration information. The lbwrol-n.mac macro file is the precise "netlist". The lbwrol-c.mac macro file is the precise CLB configuration. Some understanding of the archaic methods of Xilinx design is a prerequisite for working with the .XIL drawings and the macro files.

In this section, sheet numbers refer to the 12 sheets of the K012 drawings unless otherwise noted.

The serial control word registers are seen on sheet 2. The 11 bit GAIN[0..10] register contents are loaded into the counters seen on sheets 4 and 5. This parameter is sent over the HCB using function code 0x37. The counters are used to protect against double NEW_FRAME interrupts, if a NEW_DELAY event causes the data to back up past a NEW_FRAME event. If a NEW_FRAME event occurs within the last 256 bits prior to a NEW_DELAY event, then the back up in data would normally cause the NEW_FRAME event to be repeated. The logic on sheets 4 and 5 protect against the double NEW_FRAME, allowing only a single one to interrupt the 68K processor.

The barrel roll Xilinx design was modified at some point to support LOW BANDWIDTH MODE. This is when the macro file names were re-named to the lbwrol-n.mac type names for Low BandWidth. The modifications added bits 9 and 10 to the gain counter, and supported backups in data of greater than 256 bits. The original 9 bit gain counter only supported the normal back up of 256 bits. In Low Bandwidth Mode, if the integer delay does not change from one 4 msec interval to the next, then the NEW_DELAY interrupt can be skipped for up to six times and the write pointer to the buffer ram is not reset to zero every 4 msec. It is only reset when a NEW_DELAY interrupt does occur, or when the limit of 6 is reached. The limit of skipping at most six NEW_DELAY interrupts is a function of the 11 bit GAIN counter in the barrel roll Xilinx. The actual GAIN parameter used in non Low Bandwidth Mode is 0x0101 or 257 decimal. For every NEW_DELAY that is skipped, an additional 257 counts must be added to the GAIN parameter:

NR ND SKIPPED	GAIN	
0	1 x 257 = 257	
1	2 x 257	
2	3 x 257	
3	4 x 257	
4	5 x 257	max value that fits
5	6 x 257	/in the 11 bit counter
6	7 x 257 = 1799	<--/
7	8 x 257 = 2056	<-- greater than the 11 bit counter

As noted previously, Low Bandwidth Mode has never been used, and probably will never be required.

The NEWROLL/OLDROLL register on sheet 2 is updated every NEW_FRAME interrupt by the 68K. The contents of this input register go to the multiplexers seen on sheets 4 and 5. The outputs of the multiplexers go to the ROLL PALS to control the actual un-rolling function. The logic on sheets 4 and 5 detect the condition where a back up in the TRC data crosses the NEW_FRAME boundary, and switches to the OLDROLL until the NEW_FRAME boundary is re-crossed in the forward direction again, at which time the multiplexer switches back to the NEWROLL.

5.8.2) Data Invalid Xilinx

The drawings for the internals of this Xilinx personality (both the normal "inv" and alternate "mk3inv" versions) may be entered from sheet 1 of the Deformatter logic, via the K013D01.BLK sheet entry point (see Volume 2).

There are 12 sheets numbered K013D01 through K013D12, with suffixes of either .BLK or .XIL. The .BLK suffix indicates the "representative logic" schematic. These are similiar to TTL type logic. The .XIL suffix indicates the "representative Xilinx logic block" schematic. These show actual CLB and IOB type symbols. Neither type of drawing is a precise definition of the design.

Only the actual design macro files that are input to the Xilinx design tools provide the precise netlist and logic block configuration information. The inv-r.mac (or mk3inv-r.mac) macro file is the precise "netlist". The inv-c.mac (or mk3inv-c.mac) macro file is the precise CLB configuration. Some understanding of the archaic methods of Xilinx design is a prerequisite for working with the .XIL drawings and the macro files.

In this section, sheet numbers refer to the 12 sheets of the K013 drawings unless otherwise noted. Notes have been added to the drawings to show the difference between the normal and alternate personalities.

The serial program word control registers are seen on sheet 2. The 8 bit INV[0..7] register allows us to force data invalid independent of the validity lines from the TRC. HCB function code 0x40 provides access to this register from the real time system.

The 8 bit DELAY[0..6] plus TESTFRAME[7] register has two functions. Bit 7 enables or disables (1 or 0 respectively) the test frame loopback function in the PBD. Bits [0..6] control the alignment of the DATA INVALID BITS with respect to the DATA BITS. The pseudo random counters seen on sheet 4 are loaded with this value every NEW_FRAME.

The optimum DELAY value for use in the correlator is presently defined to be 5. There was a time when it was thought to be mode dependent, and then at some point a value of 4 was determined to be correct for one or more modes. Tests done in 1995 found the value of 5 to be slightly better than 4, but the system software still uses the value of 4 for all modes. Notes from those tests indicate that the skews between invalid data and the associated invalid line were at most + or - 40 nsec with a parameter of 5, while some skews of both 40 and 150 nsec were seen with the parameter of 4. It appears that the skew with a parameter of 4 was always in the direction that declares a few extra bits invalid rather than allowing invalid bits to pass, so the parameter value of 4 is reasonable to use.

Validities have been channelized by the time they reach the Deformatter, so the only un-rolling operations required are when rolling in groups of 16. The data invalid Xilinx is required to swap validities from adjacent pairs of Delay Centers (swap channels 0,1 with 2,3 and channels 4,5 with 6,7) as a function of the ROLL3 bit inputs at U40 pins 84, 68, 66 and 61. This operation can be seen on sheets 4 and 5.

As discussed in section 5.3.4, a backup in the TRC buffer that crosses a new frame boundary requires a back up to the old validity. This

operation can be seen on sheet 4, where F1 stores the validity from the previous frame. If backup occurs across a frame boundary, then F2 is latched with the validity from F1, and this remains asserted until the frame boundary is re-crossed again.

The fundamental difference between the normal (inv) and alternate (mk2) personalities is shown on sheet 2. In order to allow gating out the several FFT cycles with Mark III or Mark IV header pollution at each new frame boundary, the control bits from the INVALID register are re-routed to be gated with new logic. The new logic, seen in the lower left corner of sheet 2, forces data invalid at each NEW_FRAME event. The occurrence of three NEW_FFT events is then required before the forced invalid is removed, as discussed previously in section 5.6.2.

The 8 channelized validities eventually are associated with the 8 cross multiplier arrays. In polar modes, special handling is required for validities, since data from pairs of FFT pipelines goes to pairs of multiplier arrays (e.g. array 0 does RR and RL, array 1 does LR and LL). Thus invalid data in one FFT pipeline affects two arrays. For this reason, in polar modes the validites for adjacent pipelines are OR'd together. In normal polar modes, this is done with the Track Assignment command (TRC function code 0x11) "BOTH" option. Invalid frames for either the R or L polarization data will be used to flag both of the associated channel validities.

In Special Polar Mode #2 (see section 5.3.5), an additional step is necessary. Since the two polarizations are handled in two different delay centers in the PBI, the Data Invalid Xilinx provides part of the "BOTH" option by OR'ing together the validities for pairs of channels as follows: ch0/ch2, ch4/ch6, ch1/ch3 and ch5/ch7. This action is invoked in U40 with the same SWAP\ signal that swaps the data channels.

As an example of the above, in Figure 13 SPECIAL POLAR MODE #2, assume that there is invalid data in the LEFT channel. When the TRC assigns the invalid flag to "BOTH" FFT engines 2 and 3, and the Data Invalid Xilinx OR's pairs of channels together as described, then the multiplier arrays driven by FFT engines 0 and 1 will have the data flagged invalid.

5.8.3) Bit Shuffler Xilinx Personalities

The drawings for the BSP-0 through BSP-9 designs can all be entered from sheet 1 of the Deformatter schematic. For BSP-2 through BSP-9, the top level sheets are BSP2BLK.SCH through BSP9BLK.SCH respectively. Each of these top level drawings presents the functional logic involved in processing the 8 delay center track inputs in order to sequence them into the interleaved Buffer Ram banks and drive the two FFT pipelines.

Each bit shuffler personality has a 32 bit serial program word that controls the multiplexers. The bit formats of these control words is given in the BSP-2-WD.TBL through BSP-8-WD.TBL drawings found in the HCB protocol document (and entered from the Deformatter schematic sheet 1 via the D0??DEF.SCH sheet).

Since BSP-0 was derived from BSP-5, BSP-1 from BSP-3 and BSP-9 from BSP-4, the program word formats for the derivatives are the same as for the originals.

BSP-0 differs from BSP-5 only by which delay center inputs go to the multiplexers. On BSP5BLK.SCH, Mux 4 and Mux 6 have the even delay center track inputs (TR0, TR2, TR4 and TR6) while Mux 5 and Mux 7 have the odd inputs. On BSP-0.SCH, the equivalent multiplexers are U93, U95, U97 and U99 and only the even inputs are connected. The odd inputs contain oversample bits that are discarded in the modes where BSP-0 is used.

BSP-1 differs from BSP-3 only by which delay center inputs go to which inputs on the multiplexers. On BSP3BLK.SCH, each multiplexer has delay center inputs 0-7 connected to multiplexer inputs 0-7 respectively. On BSP-1.SCH, the even delay center inputs 0,2,4,6 go to the multiplexer inputs 0,1,2,3 and the odd delay center inputs go to the upper multiplexer inputs (1,3,5,7 go to 4,5,6,7).

BSP-9 differs from BSP-4 in the same way that BSP-0 differs from BSP-5. Only the even delay center inputs are used. The odd ones are discarded.

The bit shuffler functions are simple, but a detailed understanding of them can only come from a study of the bit streams presented at the input, and the desired bit streams that must be produced at the output, in each mode. An example of this was given in section 5.3.3, where Figure 5-5 shows the bit streams in and out for 1-4, 512, 2 bit, non-oversampled mode.

5.9.0) PALS

There are 49 PAL devices on the Deformatter card. There are 30 distinct PAL personalities. Figure 12 DEFORMATTER PALS tabulates which ABEL source file goes with which PAL, and indicates the drawing number for the representative logic drawing. Figure 12 also identifies which sheet(s) of the Deformatter schematic contain which PAL, and indicates the general function of the PAL.

REPRESENTATIVE LOGIC	ABEL SOURCE	U # AND JEDEC FILE NAME	SCHEMATIC SHEET #	FUNCTION	
K014D01.PAL	D009T01.ABL	U31	L008D02.SCH	PARITY CTRL	
K015D01.PAL	D009T02.ABL	U78,U79,U80,U81	L008D03 - D06.SCH	CROSS TRK PAR	
K015D02.PAL	D009T03.ABL	U55,U56,U57,U58	L008D03 - D06.SCH	CROSS TRK PAR	
K020D01.PAL	D009T04.ABL	U60,U83	L008D07 - D08.SCH	BARREL ROLL	
K020D01.PAL	D009T05.ABL	U59,U82	L008D07 - D08.SCH	BARREL ROLL	
K019D01.PAL	D009T06.ABL	U38,U39,U68,U69 U88,U89,U119,U120	L008D07 - D08.SCH L008D07 - D08.SCH	DEMUX DELAY	
K023D01.PAL	D009T07.ABL	U135	L008D13.SCH	RAM READ ADR	
K023D02.PAL	D009T08.ABL	U134	L008D13.SCH	RAM READ ADR	
	D009T09.ABL	U70,U72,U96,U99	L008D14.SCH	TEST PAT SEL	
	D009T10.ABL	U100	L008D14.SCH	OUTPUT	
	D009T11.ABL	U73	L008D14.SCH	OUTPUT	
	D009T12.ABL	U97	L008D14.SCH	OUTPUT	
	D009T13.ABL	U71	L008D14.SCH	OUTPUT	
K021D03.PAL	D009T14.ABL	U62,U63	L008D14.SCH	VALIDITY	
K022D01.PAL	D009T15.ABL	U156 WHEN MORE THAN ONE U #, THE	L008D19.SCH	FFT SEQ	
K022D02.PAL	D009T16.ABL	U189 JEDEC FILE NAME	L008D19.SCH	FFT SEQ	
K024D01.PAL	D009T17.ABL	U214 USES THE ONE THAT IS UNDERLINED	L008D22.SCH	DTACK	
	D009T18.ABL	U215	L008D22.SCH	SEL 0-6	
	D009T19.ABL	U216	L008D22.SCH	SEL 7-16	
	D009T20.ABL	U217	L008D22.SCH	SEL 17-26	
	D009T21.ABL	U218	L008D22.SCH	SEL 27-36	
	D009T22.ABL	U219	L008D22.SCH	R/W	
	D009T23.ABL	U179	L008D23.SCH	IACK	
K025D01.PAL	D009T24.ABL	U180	L008D23.SCH	INTR LATCH	
	D009T25.ABL	U181	L008D23.SCH	IPL	
K026D01.PAL	D009T26.ABL	U160	L008D28.SCH	68K SERIAL	
	D009T27.ABL	U95	L008D18.SCH	R/W	
	D009T28.ABL	U192	L008D19.SCH	68K CLK	DEFORMATTER PALS
	D009T29.ABL	U198	L008D23.SCH	IRQ7	
	D009T30.ABL	U178	L008D29.SCH	HCB	

JEDEC FILE NAME EXAMPLE: U178.JED

PBIFG57.MAN

Figure 12 DEFORMATTER PALS

5.10.0) Software

The assembly language source for the 68000 microprocessor interleaves ROM and RAM sections, using SECTION pseudo-ops. Some care must be taken if modifying the code, to make sure ROM changes are not made unless absolutely necessary. (This avoids the need to replace the pair of PROMs on all the Deformatter cards when making changes to the code.)

The assembly language source files are briefly described below:

Table 20 DEFORMATTER SOFTWARE MODULES

abortex.asm	abort / exception handler
bcd2bin.asm	convert BCD tape time to binary
boot.asm	initialization routines at boot time
bpcmd.asm	debugger break point routines
caldel.asm	model calculation of calculated offset; delays to TRC
command.asm	command handler for commands from Terminal
duart.asm	ROM based handler for Dual UART (serial I/O)
duartir.asm	RAM based handler for Dual UART
dv.asm	data variable storage allocation
dwcmd.asm	test fixture routine to test integer delay registers
flcmd.asm	flag display routine (status display, FL 1 command)
gocmd.asm	debugger GO to address command
hcb.asm	HCB handler, download Xilinx etc.
hdrdspl.asm	Header displays: HD3, HD20, HD21, HD22, HD23
inch.asm	Input Character routine (from DUART)
jell.asm	"jell" meaning to solidify; initializations after prompt cmd
job1.asm	check time reference discontinuity
job2.asm	check integer delay discontinuity
job3.asm	check timer time out for response from PBD
job4.asm	used for debugging lost new delay sync problem
job5.asm	null job, not used yet
job6.asm	null job, not used yet
job7.asm	null job, not used yet
loadpgm.asm	hex record loader, from DUART
loopcmd.asm	loops on routine until key pressed
memcmds.asm	copy mem, fill mem, mem display, mem modify type commands
miscmds.asm	tm command, several misc commands
ndirq.asm	NEW DELAY interrupt handler; this is one of the MAIN routines
nfirq.asm	NEW FRAME interrupt handler; barrel roll handled here
nmirq.asm	Non-Maskable interrupt handler; no real function
observe.asm	start obs, stop obs etc.; handles HCB function code 0x50
romend.asm	must be linked last, no operational use
rtcmd.asm	ram test command
servo.asm	routines for servoing the PBD
swdata.asm	test fixture test using substitute TRC card
t-bw.asm	lookup table used in calculating delay

t-d2hex.asm	lookup table used to convert BCD byte to hex value
t-db1.asm	lookup table to convert decimal time to bin count of bits
t-db2.asm	lookup table to convert decimal time to bin count of bits
t-db3.asm	lookup table to convert decimal time to bin count of bits
t-db4l.asm	lookup table to convert decimal time to bin count of bits
t-db4u.asm	lookup table to convert decimal time to bin count of bits
t-db5l.asm	lookup table to convert decimal time to bin count of bits
t-db5u.asm	lookup table to convert decimal time to bin count of bits
tcirq.asm	rom based, handles intr from TRC (not supposed to happen)
tdcmd.asm	TD command, enter demux or delay values
trace.asm	debug function
traps.asm	68K trap handler
trirq.asm	functions dealing with the TRC
ucmd.asm	"U" command; from Terminal, send commands to TRC
util.asm	utility commands, ASCII2Hex and others
valuate.asm	misc functions
wxcmds.asm	write byte and write word to address
1autovec.h	68000 Auto vectors 1 through 7
1cntvals.h	low bandwidth mode parameter; this file not used?
1delays.h	various timer values, not all used
1duart.h	DUART related equates
1goxy.h	cursor movements equates for header displays
1irq-sen.h	interrupt related equates
1pbd.h	PBD serial link related equates
1shtreg.h	address equates for serial data streams on the card
1sysadr.h	equates for registers, selects etc
1time.h	equates used in converting time to binary bits
1trc.h	equates for TRC related I/O
1trkhdr.h	equates to define buffer of headers from TRC
1m-2ram.mac	macros for branch and branch subroutine, rom to ram
1m-2rom.mac	macros for branch and branch subroutine, ram to rom
1m-goto.mac	macros for positioning cursor, and dsplying data
1m-trc.mac	macros for buffer and byte transfers DEF <--> TRC

5.11.0) Utility Displays

The Deformatter provides a HELP screen when either HE or ? is typed at the terminal prompt. Several of the options available are described in this section.

When the Deformatter is observing, there are several status display screens that are often of interest, along with a few address locations of interest. The HD displays are invoked by typing:

HD x

at the Deformatter prompt, where x is usually 3, 4, 20, 21, 22 or 23. These displays are periodically updated until stopped with the HD 0 command.

5.11.1) HD 3 Display

The HD 3 display provides an overall view of operation.

Table 21 DEFORMATTER HD 3 DISPLAY

HD 3 Display:

```
Micro 0 Time: 00 00 00 04 77 17 25 01  Micro 1 Time: 00 00 00 04 77 17 25 01
Micro 2 Time: 00 00 00 04 77 17 25 01  Micro 3 Time: 00 00 00 04 77 17 25 01
Time Ref    : 00 03 8E 1C 00 00        Result      : 00 03 8E 1C 00 00
Track Bits3 : 00 00 E3 86 FF F1        Binary Bits3: 00 00 E3 88 98 A0
Cal Offset3 : FF FF FF FE 67 51        DemDel bytes: 00 00 00 J0
PBDcmdBuffer: W BB RTSv^MW BC 00^M     TRC3 dly fnc: 00 83 66 2B
```

The right hand most byte of the Micro x Time fields is the flag byte that indicates if filtered time is good or bad. A value of 01 indicates time is good, a value of 00 indicates time is bad.

The Cal Offset field shows the calculated offset for micro #3. The nominal value expected here when the servo is operating is -100,000 decimal. The 48 bit two's complement integer hex value in the example above is -104,623 and is typical. There are usually 7 nibbles of leading F, followed by an E as seen in the example.

The PBDcmdBuffer field shows the servo status when the DEF has the servo function. In the example above, the Real Time System (RTS) has the servo function. If the Deformatter has the servo function, then the RTSv field would either show SYNC while the calculated offset was in the narrow window where no servo commands are needed, or it would show the last sent hex command for the speed up or slow down of the tape. The BB is the PBD address for the speed command and the BC is the PBD address for the duration command. The four character HEX speed command and two character HEX duration commands come from the servo table. (See section 5.7.2 for a servo table example.)

5.11.2) HD 2X Displays

The HD 20 Display is shown below. The X in HD 2X specifies which of the four TRC micros is displayed. This display shows the track header fields and errors and status:

Table 22 DEFORMATTER HD 20 DISPLAY

uP	0	Aux1	Aux2	Aux3	Aux4	MJDS	SSSS.ssss	CRC	E1E2	Parity-Er	HdrE	Rsyn	Inva	Frms
Tk	0	1234	5678	9ABC	DEAD	0000	0534 9225	AB47	00A0	0000	0000	0000	0000	43AC
Tk	1	1234	5678	9ABC	DEAD	0000	0534 9225	AB47	00A0	0000	0000	0000	0000	43AC
Tk	2	1234	5678	9ABC	DEAD	0000	0534 9225	AB47	00A0	0000	0000	0000	0000	43AC
Tk	3	1234	5678	9ABC	DEAD	0000	0534 9225	AB47	00A0	0000	0000	0000	0000	43AC
Tk	4	1234	5678	9ABC	DEAD	0000	0534 9225	AB47	00A0	0000	0000	0000	0000	43AC
Tk	5	1234	5678	9ABC	DEAD	0000	0534 9225	AB47	00A0	0000	0000	0000	0000	43AC
Tk	6	1234	5678	9ABC	DEAD	0000	0534 9225	AB47	00A0	0000	0000	0000	0000	43AC
Tk	7	1234	5678	9ABC	DEAD	0000	0534 9225	AB47	00A0	0000	0000	0000	0000	43AC
Tk	S	01A1	425B	A224	B104	1451	ADEA 4589	D691	040F	0000	0000	0000	0000	43AC

This display shows the contents of the various fields in the track headers along with the error counters and TRC status (the E1E2 fields). The column headings are pretty self explanatory except for the E1E2 fields. These represent two bytes of status, defined as follows:

```
        E1:     frame parity error count

        E2:     MS nibble = sese
                            ----
                            0000 no sync detected at all
                            1000 imperfect sync detected, no bit slip
                            1100 imperfect sync detected, with bit slip
this is "normal" -------->  1010 perfect sync, no slip
                            1111 perfect sync, with bit slip

                LS nibble = cppp        c   = CRC header error
                                        ppp = header parity error count
```

The E1E2 = 00A0 shown in the example is the normal "good" state. The CRC and E1E2 fields are "snapshot" copies of a recent frame evaluation. The Parity-Er, HdrE, Rsyn and Inva fields are running sums, that are reset each time the RTS reads headers, or can be reset by the Deformatter ZA terminal command.

The Frms (Frames) field is the count of tape frames since the last RTS readout or reset. (Readout of error counts causes them to be reset.)

5.11.3) HD 4 Display

The HD 4 Display shows the channelized parameters from the TRC:

Table 23 DEFORMATTER HD 4 DISPLAY

	Mode	Spare	TimeRst	Aborts	0inv	1inv	W-Dog	Spare	Frms	0MJD 0SSSSS.ssss
uP0	01	00	0000	0000	0000	0000	0000	0000	7008	0000 000563 3350
uP1	01	00	0000	0000	0000	0000	0000	0000	7008	0000 000563 3350
uP2	01	00	0000	0000	0000	0000	0000	0000	7009	0000 000563 3350
uP3	01	00	0000	0000	0000	0000	0000	0000	700A	0000 000563 3350

Each TRC micro (0-3) handles two FFT engines (0-1). The channelized invalids for the two engines are the 0inv and 1inv columns.

The TimeRst, Aborts and W-Dog columns are some of the error conditions that might be detected by a TRC microprocessor. Mode gives the filtered time status. See D028TRC.HCB in Appendix III, function code 0x1E, for a description of these error conditions.

For any of the HD displays, the HD 0 command (HD zero) stops the display from updating. A KW (kill window) command will then clear the screen.

5.11.4) FL 1 Display

The FL 1 status screen is not a periodically updated display. In response to typing FL 1 at the prompt, the following data is displayed:

Table 24 DEFORMATTER FL 1 DISPLAY

Label	Active Value	HCB Rcv Buffer Value
MJDate	0000	0000
TimeRef	0004 7B46 0000	0000 0000 0000
Format	VLBA	VLBA
NRSamples	0002 0000	0002 0000
PipeLineDly	FFC7	FFC7
RecordRate	8 Mhz	8 Mhz
Mode	1 to 4	1 to 4
SampleFactor	x1	x1
Model Source	00	01
TRC High Limit	0000 0000	0000 0400
TRC Low Limit	0000 0000	0101 0000
BW Factor	0000	(active value only, initialized from BWTable)

A subsequent <return> produces the second part of this status display:

Label		Val	Ph Cntr 0	Ph Cntr 1	Ph Cntr 2	Ph Cntr 3
Demux-Delay	=		00	00	00	00
xTrackParity	=		00	00	00	00
BitShufCtrlWd	=		02B0 8B08	02B0 8B08	02B0 8B08	02B0 8B08
ForceChInvalid	=	00				

```
CtrlServo      = No
OverlapCnt     = 0000  and count by 1
DataValidDelay = 04
BarrelRoll dly = 0101  and Roll is off
```

The Active Value column normally will match the HCB column when observing, since the values sent over the HCB will be applied when observing begins, except for the TimeRef that increments every 4 msec.

5.11.5) Memory Display of location B38 (integer delays)

The Memory Display (MD) command may be used to display memory contents at any readable address. The following example shows the use of this command to display the four 16 bit integer delay values:

```
DEF2-08> md b38 4
          +0      +2      +4      +6      +8      +A      +C      +E
00000B38= 00 00   00 00   00 00   00 00
MD >q
DEF2-08>
```

where b38 is the starting address and 4 is the number of 16 bit words to display.

Other addresses of possible interest can be identified by looking at the DV.LST file that results from the assembly and linking process. DV.ASM is the data variable allocation file, and DV.LST is the listing file that includes the relocated addresses after linking.

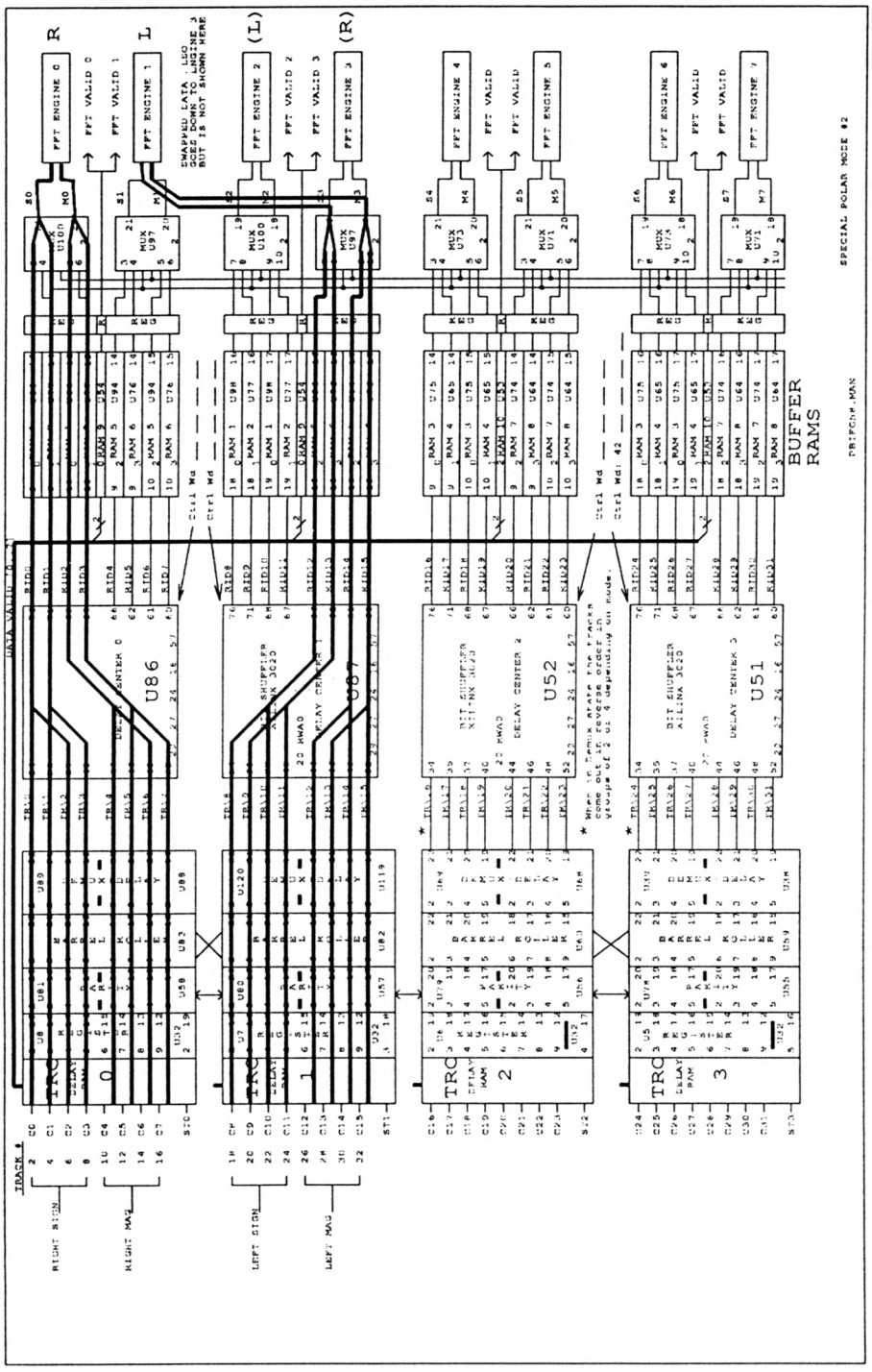


Figure 13 SPECIAL POLAR MODE #2

Chapter 6 The PBI Test Fixtures

6.1.0) INTRODUCTION

The PBI test fixtures are used to test both Track Recovery Cards (TRCs) and Deformatter Cards (DEFs). There are two PBI test fixtures. This chapter mainly discusses PBI test fixture #1, which supports only one TRC, but which has some built in test functions for GO/NO-GO testing. PBI test fixture #2 has no built in functions. It only provides the capability of running two TRC and one DEF as if they were in the system. A brief discussion of this fixture is found in section 6.9.0.

PBI test fixture #1 is capable of being operated in a variety of ways as seen below:

1) Test TRC cards in "stand alone mode" using the Deformatter replacement card (with or without a CRT terminal).

2) Test TRC cards with a Deformatter host.

3) Test DEF cards with no TRC present.

4) Test DEF cards with a TRC card as the DEF data source.

5) Test DEF cards with a special data generator card as the DEF data source.

In some of the modes above, an HCB connection between the DEF card under test and a host VME computer is required (as a source, for example, of Xilinx personalities and of software). The test fixture supports the HCB requirement.

The PBI test fixture has two logic cards for control of the TRC/DEF cards under test. Documentation for these cards and other drawings that describe the PBI test fixture are listed below:

drawing	function
L010D01.SCH	logic diagram of PBI test fix card #1
L011D01.SCH	logic diagram of PBI test fix card #2
L022D01.SCH	Deformatter simulator card
L023D01.SCH	interconnect diagram of the PBI test fixture
L037D01.SCH	Data generator card
L046D01.SCH	PBI test fixture clock generator

6.2.0) PBI Test Fixture #1 General Description

The PBI test fixture is meant to generate a realistic environment for the testing of TRC and DEF cards. A test data generator supplies 8

track signals that satisfy the VLBA PBD frame specification (for both VLBA and MKIII modes) to a TRC under test. The DEF input can come from the TRC data output or from a special data generator card.

A VCO internal to the test fixture supplies the card(s) under test with a system clock that is adjustable between about 20 and 40 MHz. A phase lock loop that uses the system clock as the PLL reference, controls a second oscillator that supplies the prcper frequency for generating the track signals (a different frequency is required for the PBD frame signal because of the bit overhead of the frame parity bits and the frame header). On the test fixture logic card schematics, the system clock signal is labeled C32.0 and the track clock signal is labeled C9.072 (or CLK36 where it exists before it is divided by 4) in spite of their variability. This is because 32 MHz and 9.072 MHz are the nominal frequencies for these clocks in the system (for VLBA mode). In much of the discussion that follows, clocks and clock rates will be spoken of as if they were at the nominal rate. Thus track signals will be described as being at 9.072 MHz and the system clock as being at 32 MHz irregardless of their variability. As noted in section 5.5.0, the Deformatter 16.0 MHz clock to the 68000 microprocessor is no longer derived from a crystal. It now comes from the 32 MHz "system clock". As a result, it is possible the Deformatter may not operate properly much above 32 MHz in the test fixture.

The central idea behind the test fixture is to have it generate PBD like track signals to drive the TRCs. The data portion of the PBD signals are derived from a 15 bit pseudo-random data generator in the test fixture. The TRC under test will recover both the PBD frame data and headers as it would in the VLBA correlator. To verify the proper operation of the TRC, the contents of the headers can be displayed on a CRT terminal while the data output can be checked against a 15 bit predict data generator in the fixture. Error indication LEDs are latched on whenever an error is detected in the TRC recovered data output.

The TRC track data outputs are connected to a DEF card (each output of the TRC is connected to two DEF input pins to simulate the 2 TRCs normally connected to a DEF in the system). The DEF deformats the TRC output data and blocks the reconstructed "sampler" data streams into short bursts of 512 bits (in the correlator, each 512 bit burst would provide the FFT cards with data for one FFT cycle). The short bursts of 512 bits out of the DEF are then checked in the test fixture against another 15 bit predict data generator. Again, errors on DEF output lines are detected and latched for a test fixture error display.

The test fixture can simulate most of the modes that the TRC-DEF card sets must contend with in the VLBA correlator. The data generator can generate data at all standard VLBA sample rates from the PBD playback standpoint (the lowest VLBA sample rates need not be supported because the PBD speed-up factor will always give these low sample rates a factor of four increase at the TRC input). The DEF can be tested supporting any standard FFT size, any standard overlap factor, or any formatter mode (VLBA 1:4, VLBA 1:2, VLBA 1:1, VLBA 2:1, VLBA 4:1, or MKIII).

6.2.1) Front Panel Features

Some of the actions of the PBI test fixture can be controlled and monitored from the front panel switches and lamps. Other functions and displays require a terminal or an HCB connection to a VME computer. The functions of the test fixture front panel are given below:

1) Power switch and power indicator LED: controls the AC power to the test fixture.

2) TRC track go/no-go LEDs. The 18 TRC track error lamps reflect the error status of the 18 TRC tracks. When the test fixture is properly setup, a 15 bit pseudo random data pattern is formatted into PBD tracks for recovery by the TRC. If the output of a given recovered track data signal out of the TRC deviates from the original formatter data pattern, an error is latched on the appropriate LED. The LEDs can be made to be momentary error indications instead of being latched, via a microprocessor provided option.

3) DEF channel go/no-go LEDs. The 16 DEF error lamps reflect the error status of the 16 DEF data output signals. When the test fixture is properly setup, a 15 bit pseudo random data pattern is formatted into PBD tracks for recovery by the TRC. The TRC will de-frame the track data and drive the DEF with the recovered signals. The DEF will deformat the TRC output and develop drive signals for an FFT card. If a given DEF channel output deviates from the data pattern predicted for it, an error is latched on the appropriate LED. Again, the LEDs can be made to be momentary error indicators via an option provided by the microprocessor. Both the TRC and DEF error latches can be reset by pushing the LED RESET push button switch.

4) Manual data generator switches. There are two rows of 8 toggle switches on the front panel of the PBI test fixture. The bottom row of switches controls the source of data of the 8 PBD tracks generated by the PBI Test Fixture Card # 1. The two options for each of the track signals are either the 15 bit pseudo random data generator (switch handle down) or an 8 bit switch generated data pattern (switch handle up). The 8 bit switch generated data pattern is controlled by the states of the top row of front panel switches.

5) Microprocessor reset switch, clock frequency control, and clock frequency indicator. The microprocessor reset switch resets the 87C51 on test card # 1. The clock frequency control knob controls the clock frequency to the test fixture within a range of about 20 to 40 MHz. The exact clock frequency is displayed on the front panel frequency readouts (because of the PLL, the frame clock will always be 9.072/32 times the readout frequency).

6.3.0) PBI Test Fixture # 1, Card # 1 Description

Logic diagram L010D01.SCH gives the schematic of the PBI Test Fixture Control Card # 1. This card serves mainly to generate PBD-like

track signals for the TRC track inputs. This card also has a microprocessor that sets the test fixture mode.

6.3.0.0) Microprocessor

The microprocessor is seen in the card # 1 logic diagram in the upper left hand corner. The micro can communicate with a terminal via its serial port. The functions of the microprocessor include;

1) Put the test fixture in a default configuration mode on power up.

2) Communicate with a terminal for the purpose of setting non-default test fixture configurations.

3) Set the track frame phase lock loop frequency (two setups are possible, one for VLBA mode and one for MKIII mode).

4) Provide active control of the PBD header contents (principally, make the header time run).

The microprocessor writes control bits into the 2A, 2B, 1A, 2D, and 2E registers. These bits control the function of both test fixture control cards and the PLL.

6.3.0.1) PBD Frame Sequencer

The PBD frame sequencer is a 22,680 step state machine (in VLBA mode). The counter ICs 10A, 11A, 10B, and 11B count through 22,680 addresses of the sequencer ROMs 11C and 10C. Sequencer signal S15 controls the radix of the counter (for MKIII the radix is 22,500). The other sequencer output signals provide logic control of the rest of the card and result in the generation of PBD like track signals for output to the TRC.

6.3.0.2) Data Generator

The data generator is made up of ICs 9A, 7A, 8A, and 6A. IC 9A is controlled by the RATE[0..2] signals provided by the microprocessor and produces gating signals required for the 7A and 8A data generator and the 6A mode generator to run at the "sample" rate selected. The test fixture is capable of duplicating all of the legal VLBA sample rates (after possible PBD speed up factors).

The 7A, 8A data generator develops a 15 bit pseudo random data sequence at the selected data rate. This data generator does not run during data parity bits or during the frame header (signal S3 from the sequencer prevents operation during these periods). The result is that the data content of the frame satisfies the non-replacement VLBA frame format (see the VLBA PBD frame spec, A56000N003 in Appendix V). IC 6A is

programmed by the microprocessor supplied MODE[0..2] signals and controls the sample dispersal (formatting) among the track data signals (VLBA 1:4, VLBA 1:2, VLBA 1:1, etc).

The output of the 6A IC is four 9.072 MHz (nominal) data signals that will be used as the data portion of the track frames. The signals appear to be the data of PBD frames at a selected sample rate and with a selected formatter track mux/demux mode.

To simulate VLBA 1:4 mode, for example, the 7A, 8A data generator runs at the full 36 MHz clock rate (actually, the effective clock rate is 32 MHz since it is clocked with the 36 MHz clock but does nor run during data parity bits or during the frame header). IC 6A is clocked at 9.072 MHz (8 MHz effective rate) so that every 4^{th} pseudo random data pattern bit goes down a given RAWTRACK output imitating the 1 line into 4 track operation of a VLBA formatter in VLBA 1:4 mode.

For VLBA 1:2 mode, the 7A, 8A data generator runs at ½ the 36 MHz clock rate. IC 6A is still clocked at 9.072 MHz so now every 2^{nd} pseudo random data pattern bit goes down a given RAWTRACK output, which simulates the 1 line into 2 track operation of a VLBA formatter. As seen on the card # 1 logic diagram, track data signals RAWTRACK0 and RAWTRACK2 are the same and signals RAWTRACK1 and RAWTRACK3 are the same.

To simulate VLBA 4:1 mode, the 7A, 8A data generator runs at 1/16 the 36 MHz clock rate (or lower). IC 6A is still clocked at 9.072 MHz with the result that the pseudo random data pattern appears out of all four of the 6A outputs but has a bit rate ¼ of 9.072 MHz. Thus the original pseudo random data pattern can be reconstructed by selecting every 4^{th} bit from a track output. This duplicates a VLBA formatter 4 line into 1 track mode where the 4 lines have identical data.

6.3.0.3) PBI Test Fixture #1 Formatter

ICs 3D, 3B, 3C, and 4F take the RAWTRACK data signals and, under control of the frame sequencer, create the PBD frame outputs. Each PAL generates two track signals from a single RAWTRACK signal. The two track signals are identical except for the possible switch option discussed below.

The 256X4 RAM 5E has a PBD frame header written into it by the card microprocessor during the data portion of a frame. When it comes time for the frame header to be transmitted, counters 6E and 6F, under control of the frame sequencer, count through RAM addresses where the header is stored. Thus the microprocessor has full control of the AUX-data field, the sync field, and the time field of the header. By incrementing the header time code every frame, the micro can make the header time appear to run in a normal manner.

The 5E RAM has four outputs, 5E-10 provides the serial header pattern discussed above while the other three are used by the microprocessor to make controllable errors in the frames. When 5E-12 (ERR0) is high, for example, the parity of a PBD frame parity byte will be encoded incorrectly. When 5E-14 (ERR1) is high, a crc error will be encoded in the frame header and when 5E-16 (ERR2) is high, a bit in the header (such as a bit of the sync word) is inverted from the pattern in

the RAM. Enable signals uE0, uE1, and uE2 allow the microprocessor to turn the errors on and off without having to change the contents of the RAM. By making controllable errors in the PBD frame signals, the ability of the TRC frame error analysis software can be tested.

IC 5D, again under control on the frame sequencer, encodes the CRC across the time field (or across the header field for MKIII mode) during the frame header.

Parity counting and encoding is done in the 3D, 3B, 3C and 4F ICs. The sequencer control signals S4, S5, S6 and S7 control the action of these formatter PALs. Sequencer control signals S0 and S1 control the header readout while S9 and S10 control the CRC generation. All of the PBD frame generation logic runs on the 9.072 MHz clock.

As discussed earlier, the front panel of the PBI test fixture has two rows of 8 toggle switches. These switches control the action of the formatter PALs. The bottom row of switches controls the source of the data for the 8 track signals generated in the 3D, 3B, 3C, and 4F PALs. Each of these 8 outputs can be individually switched between the pseudo random data generator and a manually controlled serial 8 bit data pattern. ICs 7D and 8D are driven by the 8 source switches and generate the TRACKMODE signals. Through these ICs, the switches can select the sources of data for the 8 TRACK signals. The MODE[0..2] signals also drive 8D and 7D so that for VLBA 2:1 or VLBA 4:1 modes the switch data will get into only one of the 2, or one of 4 lines multiplexed onto the track signal.

IC 9D is a parallel to serial converter that takes the top row of front panel switches and generates a recurrent 8 bit pattern to replace the pseudo random data in the formatter PALs. This 8 bit pattern is synchronized to the PBD frame parity bytes and hence every data parity byte will be the same. (One exception to this statement is an option whereby the switch pattern will be sent only in the very first and in the very last parity bytes of each frame. This option is helpful in seeing that the "wound" in the data caused by the header of the VLBA non-replacement frame format has been properly healed. IC 8C, under programming by the microprocessor, controls this option with the uS10 control bit.)

6.3.0.4) Track Mix, Track Roll, Cross Track Parity, and NRZM Generator

The logic in the upper right hand corner of the logic diagram (L010D01.SCH), prepares the formatter track signals to drive the TRC. IC 8B provides an opportunity to scramble the track order should the need ever arise (this PAL is presently programmed for a straight through function). ICs 7B and 8C provide a formatter like track roll function. The microprocessor signals uS8 and uS9 control the track roll (no roll, roll by 1, roll by 2 or, roll by 4). IC 8C generates the ROLL[0..2] signals which drive PAL 7B where the roll is implemented.

IC 7C either selects a track signal (track 0) to drive the system track output, or it generates parity across the other tracks as data for the system track (i.e., it outputs the parity count of the 8 active

tracks and this output becomes the system track data source). The micro control signals uS5 and uS6 select the operation.

ICs 10D and 10E do the differential encoding of the track signals. The VLBA formatter drives the recorders at a VLBA station with a differential NRZM signal and this stage of the test fixture simulates this action. A logic one input to the NRZM stage will cause the stage output to toggle, while a logic zero input will cause the output to hold.

6.3.0.5) Differential ECL Drivers

In the system, the PBD drives the TRCs with differential ECL track signals. ICs 1D, 1E, 1B, 1C, and 4E convert the 8 track and 1 system track signals developed as described above into differential ECL. Each of the resulting differential signals is wired to 2 TRC track inputs (each TRC handles the recovery of 18-tracks in the system).

6.3.1) PBI Test Fixture #1, Card # 1 Software Description

The 87C51 microprocessor on the PBI test card # 1 has no external RAM main memory. The main program for the this microprocessor is fully contained in the on-board UVROM.

6.3.1.0) Brief Descriptions of the Software Modules

Below is a brief description of each of the modules listed in the chart below. (Memory allocations seen below are as of 7/1/92). As can be seen the microprocessor is almost out of ROM space. In addition, the OBS.ASM program uses almost all of the 87C51 RAM space for an image of the header file.

Section Name	Starting Address	Ending Address	Size
MASTER.ASM	0000	00F5	00F6
MONITOR.ASM	00F6	0767	0672
TEST.ASM	0768	07B5	004E
HEADER.ASM	0800	0A1E	021F
OBS.ASM	0A1F	0FB1	0593

6.3.1.1) MASTER.ASM

MASTER.ASM has the software executed after a hardware reset to the microprocessor. This software will initialize the card, the test fixture clock PLL, and initialize microprocessor functions like the interrupt logic, the serial port, etc. This module also has the interrupt vectors. After a reset, the micro will execute from an idle loop in MASTER.ASM awaiting instructions from the terminal.

6.3.1.2) MONITOR.ASM

The monitor software package has various terminal options supported by the microprocessor.

6.3.1.3) TEST.ASM

The TEST.ASM module has a number of hardware test functions that allow an operator to test PBI test fixture test card # 1.

6.3.1.4) HEADER.ASM

The HEADER.ASM module contains 4 pre-canned PBD frame headers. These headers can be selected via the monitor routine (type H0, H1, H2, or H3). The contents of the AUX-data field is different in the 4 headers as is the time code. In addition to different header contents, the ERR0, ERR1, and ERR3 bit fields of the pre-canned headers can make controllable errors (see section 3.0.3). The pre-coded errors are listed below;

H0	has 1 header parity error
H1	has 1 frame data parity error and a CRC error
H2*	has 3 header parity errors and 7 frame data parity errors
H3	has 6 header parity errors, 3 frame data errors and a sync error

*H2 can also produce spurious sync words in the data. The way to do this is to put one track in switch data mode with a data pattern of all one's and have parity errors enabled.

6.3.1.5) OBS.ASM

After a reset, the PBI test card # 1 microprocessor comes up in a "non-observing" mode. The time code in the PBD frames does not run and the micro is idle. This mode of operation is adequate for low level testing of the TRC card (such as TESTOBS, see section 6.7.0) but the use of more advanced TRC software (such as AUTOTEST) requires the PBD time code to run properly. By typing O (no carriage return required) on the PBI test fixture card # 1 terminal, the OBS.ASM software is executed. This software makes the header time code run and puts incrementing counts in the AUX-data field on a frame. While the OBS program is running, MONITOR.ASM has support for writing directly into the AUX and time fields of the frame header. Both VLBA and MKIII modes are supported in OBS.ASM.

6.4.0) PBI Test Fixture #1, Card # 2 Description

Logic diagram L011D01.SCH gives the schematic of the PBI test fixture control card # 2. This card checks the output of the TRC and DEF

cards under test and will detect errors and latch their occurrence on front panel lamps. The output of the both the TRC and DEF cards are known to be portions of 15 bit feedback shift register pseudo random data patterns, thus they are completely predictable. The main problem in testing the output of either card for errors is the discontinuous nature of their outputs. The TRC must back up in data every 4.128 msec NEW DELAY cycle and the DEF outputs channel dat.. in 512 bit bursts with a 4 bit gap between bursts. Both these disruptions in the flow of data from the cards under test must be dealt with.

6.4.0.0) Sequencer

The FFT cycle/fringe cycle sequencer, seen in the upper right hand corner of the logic diagram, generates control signals for the rest of the logic on the card and for the cards under test. The sequencer simulates the VLBA correlator system in generating FFT cycle and the fringe cycle signals.

Counter ICs 11D, 10E, and 9E count through 516 states of an FFT cycle (with the radix set by S7). IC 7D counts through the 256 FFT cycles of a fringe cycle.

6.4.0.1) TRC Error Indications

The 18 tracks from the TRC under test are checked for errors in ICs 1A, 2A, 1B, 2B, and 1C. The predicted signals for these PAL comparators are generated by ICs 3B, 3C, and 3A. The rational behind the error checking is for ICs 3B, 3C, and 3A to exactly duplicate the operation of ICs 7A, 8A, and 6A on PBI test fixture card # 1 and for the comparator PALs to check the output data pattern from these chips against the TRC outputs.

First, however, the predict generator must be synchronized to the TRC output. This synchronism is made more difficult by the fact that the TRC outputs have time discontinuities in them every 4.128 msec fringe cycle. The time discontinuities result from the 512/516 inefficiency of the VLBA correlator. In the system, the FFT cards process PBD data for only 512 clocks out of every 516. Thus, in the system, the TRC gets ahead of itself and must "back up" in data every fringe cycle.

In order to synchronize the 3B-3C-3A predict data generators, the first 15 bits of every fringe cycle (at whatever bit rate is selected on card # 1) is serially loaded into the 3B-3C chips. For this purpose, the 3B-3C generator is made temporarily into a shift register for 15 clocks at the start of every fringe cycle and then returned to its function as a feedback data generator for the rest of the fringe cycle. The 15 bit pattern loaded into 3B-3C is generated by IC 2C. If the 15 bits loaded into the data generator is error free, this data generator should, after it is returned to its feedback shift register function, duplicate the operation of the TRC source data generator for the rest of the fringe cycle. Its output can thus be used as a predict data generator to predict the TRC track data output.

The predict generator is initialized from TRC output signals TRACK0, TRACK1, TRACK2, and TRACK3. The exact operation of these ICs depends on the operating mode. In VLBA 1:4 mode, for example, the four track signals above are multiplexed together to make the signal MUX DATA\. For the first 15 bits of each fringe cycle (as delimited by the NEW DELAY input) the signal MUX DATA\ is clocked, at 32 MHz, into the 3B-3C data generator.

For VLBA 1:2 mode two of the TRACK signals are multiplexed together and the first 15 bits of each NEW DELAY cycle are clocked, at 16 MHz, into the 3B-3C data generator. For VLBA 4:1 mode, every 4^{th} bit of the TRACK0 signal is selected and the first 15 bits of the resulting data pattern is used to initialize the 3B-3C generator at the appropriate (low) clock rate. The 2C-3B- 3C-3A circuit always clocks at the selected data rate.

The signal INIT, generated by ICs 8E and 8A, is the 15-clock wide signal (at the data clock rate) that loads the 3B-3C generator. The NEW DELAY signals comes from the DEF (or DEF simulator) board. The INIT signal also de- sensitizes the error comparators (1A, 2A, 1B, 2B, and 1C) so errors will not be detected during the predict load process.

6.4.0.2) DEF Error Indications

DEF error detection works in much the same way as the TRC error detectors. The main differences for DEF error testing are:

1) the mux/demux action of IC 2C is not required since the Deformatter performs this function (in order to reconstruct the original sampled outputs) and,

2) time discontinuities in the DEF data occur every 516 clocks since the DEF supports the FFT cycles. An additional complication of error detection in the DEF outputs are the need to support 1024-point and 2048-point transforms.

For DEF error detection, four predict data generators are required instead of one data generator with four outputs like the TRC. Each is configured into a shift register during the first 15 bits of every FFT cycle and initialized to the DEF data output. In the case of the DEF predict data generators, the clock rate is always 32 MHz. For transform sizes of 512-point or smaller IC 8C generates a 15 bit wide gate (LOADx) signal which serial loads the predict generators.

When supporting 1024-point transforms, however, the DEF breaks up the deformatted data stream into odd and even bits and sends each to adjacent FFT engines. In order to predict the DEF outputs for this mode, the predict generator consisting of ICs 7A, 7B, 6A, and 6B and the predict generator consisting of ICs 5A and 4A are loaded from the outputs of two DEF outputs (in order to put the odd and even bits back together).

Since the data generator cannot be clocked at 64 MHz, the data generator is made to shift 2 bits per 32 MHz clock transition. The PAL data generators hence get a little complicated but in essence they are still 15 bit feedback shift register pseudo random data generators (see Figure 14 PSEUDO RANDOM DATA GENERATORS).

In 1024-point transform mode, the last two predict data generators (5B-4B and 6C-5C) are not used. The gate needed to load the active 15 bit (equivalent) data generators need be only 9 bits long since the data generator is being loaded 2 bits at a time (and hence one bit per FFT cycle does not get properly checked).

When supporting 2048-point transforms, the DEF card breaks up the deformatted data into 4 bit streams, sending every 4^{th} sample to each of 4 adjacent FFT engines. In order to predict the DEF outputs for this mode, the predict generator consisting of ICs 7A, 7B, 6A, and 6B is loaded from the outputs of four DEF outputs. Now the PAL data generators are shifted 4 bits per 32 MHz clock transition and the gate needed to load the active 15 bit (equivalent) data generators need be only 4 bits long (see Figure 14).

IC 4C decodes the transform size selected by the microprocessor via the uS26 and uS27 control lines. The outputs of this PAL, PRED[0..3], is taken from all 4 predict generators for short transforms, from only 2 predict generators for 1024-point transforms, and from only one predict generator for 2048-point transforms.

Comparator PALs 2D and 1D test 8 DEF output lines against the output of the predict data generators. PALs 2E and 1E test the last 8 DEF outputs. These PALs use the first 8 DEF outputs as predicted data sources. When observing the error indications on the PBI test fixture LEDs, this fact must be remembered. If the LEDs for this second set of 8 DEF outputs are off, it means that these DEF outputs are error free only if the first 8 LEDs are also off.

6.4.0.3) Data to Sequence Conversion

The logic seen in the lower right corner of the card # 2 logic diagram was meant to be used with a logic analyzer. The purpose of this logic was to capture the first 15 bits of each FFT cycle and, by using the look table in ROMs 11A and 11B, convert it into a sequence of numbers for the 15 bit random data generator pattern. By recording consecutive start patterns from consecutive FFT cycles, the analyzer will yield the overlap/offset action of the DEF and hence it can be thus checked (manually) for proper operation.

The principle behind this logic is that of numbering all of the samples going into the formatter and then looking at the number of the first bit entering the FFT card for many consecutive FFT cycle. If the DEF is programmed to support FFT sizes of 64 bits, with an overlap factor of 4, for example, the expected sequence would by N, N+16, N+32, N+48, N+512, N+512+16, N+512+32, N+512+48, N+1024, etc., for whatever N the logic analyzer storage started at. This logic has never been used.

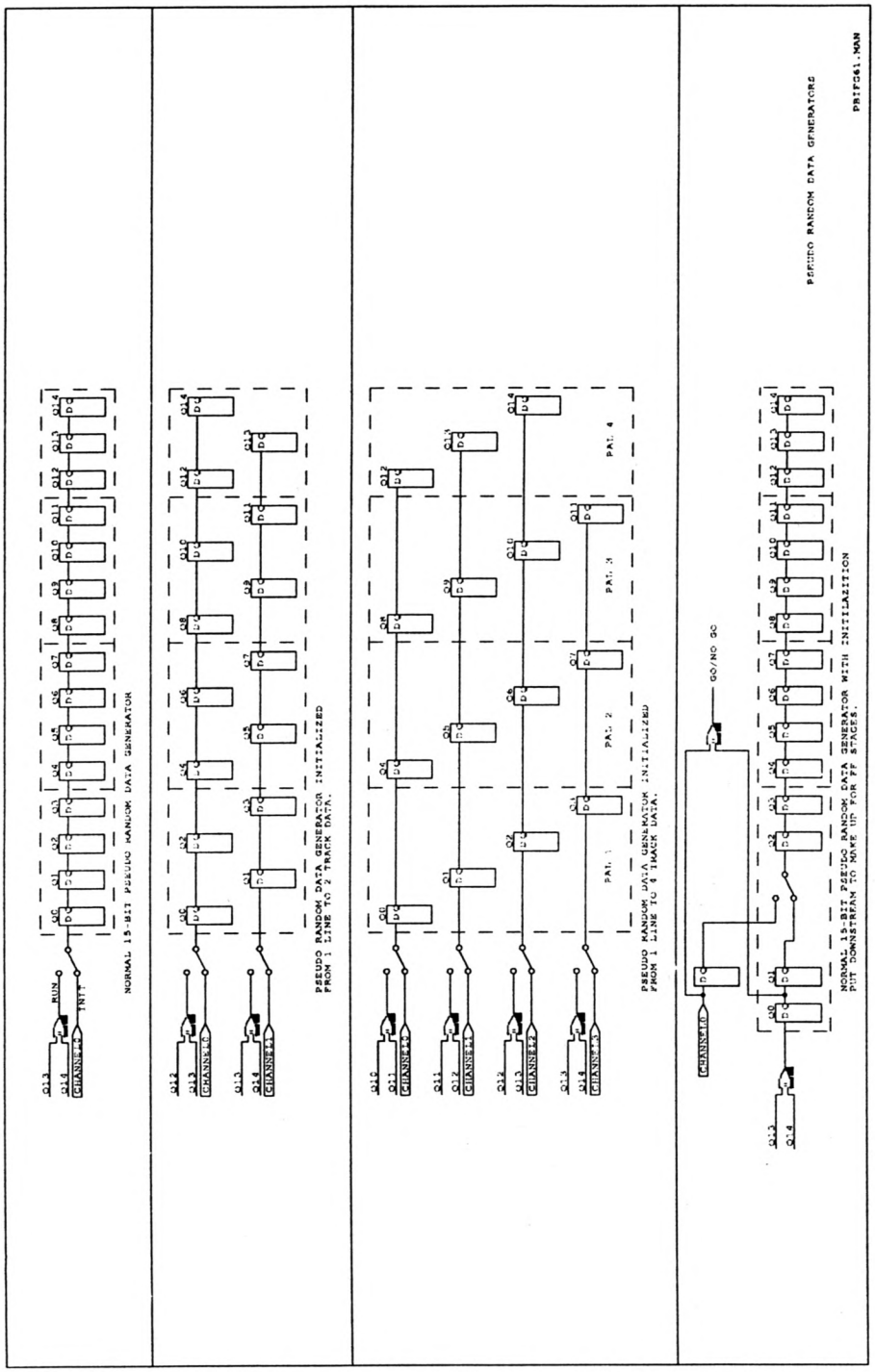


Figure 14 PSEUDO RANDOM DATA GENERATORS

6.5.0) PBI Test Fixture #1 Deformatter Simulator Card Description

The Deformatter simulator can be used to provide a DEF-like communication interface to a TRC in the PBI test fixture without requiring a DEF card. The two advantages to using the simulator are, first, in not blocking physical access to the TRC (an aid to troubleshooting) and second, the test fixture need not have an HCB connection with a computer. The simulator supplies the following functions:

1) An 87C51 microprocessor to control communication with the TRC.

2) A ROM with the TRC Xilinx personality.

3) A ROM with the TRC main program.

4) A source for the 8 MHz clock and the NEW DELAY signals that normally come from the DEF.

The simulator plugs into the DEF card slot but uses only two of the DEF DIN connectors (P2 and P3, the left most connectors when viewed from the front). There are 3 switches on this board. The push button switch provides a reset signal for the on-board microprocessor. When reset, the microprocessor will read the state of the two toggle switches and carry out the one of four functions selected by them. These selections are:

1) Toggle switches both down (= 00), reset the TRC microprocessors and tell the TRC microprocessor their identities (function code 04 or 05). The two LEDs on the simulator will indicate the success of this operation. If both LEDs light green, the process was successful. If one or more of the LEDs light red, the operation to the corresponding TRC micro failed. If one or both LEDs do not light at all, the operation was never completed (it failed).

2) Toggle switches = 01, the simulator downloads the Xilinx personality to TRC micro B. As discussed above, the simulator has a ROM with the TRC Xilinx personality stored in it. The simulator uses the TRC-DEF communication link to download the ROM contents to the TRC (using the normal TRC-DEF protocol). As the transfer takes place, the simulator LEDs will blink. When the transfer is complete, the final state of these LEDs indicated the success of the download (green for successful and red or blank for failure).

3) Toggle switches = 10, the simulator downloads the TRC main program to TRC microprocessors. As indicated above, the simulator also has a ROM with the TRC software stored in it. The simulator uses the TRC-DEF communication link to download the ROM contents to the TRC. As the transfer takes place, the simulator LEDs will blink. After the download, the simulator will read back the TRC RAMs to verify the transfer. The final state of the LEDs indicated the success of the download (again, green for successful and red or blank for failure).

4) Toggle switches = 11, the simulator instructs the TRC

microprocessors to execute TESTOBS. TESTOBS is a stand alone program in the TRC software that will allow testing of the TRC.

6.6.0) PBI Test Fixture #1 Deformatter Data Generator Card Description

A card was built to be used as a "substitute TRC" card, to provide input data for the Deformatter when installed in place of the TRC. The schematic drawing for this card is L037D01.SCH. No one has used this card in many years.

See Appendix I, List of Files, for the location of the files associated with this card.

6.7.0) TRC Test Procedure

There are several levels of testing possible for a TRC in the PBI test fixture. In all of the test descriptions below, it is assumed that one TRC and the Deformatter simulator card are plugged into the PBI test fixture. It is assumed that the only time a DEF card would be plugged into the fixture is when the DEF and not the TRC is being tested.

6.7.1) Stand-alone TRC Testing (TESTOBS)

Stand-alone testing is the fastest and simplest test that can be run on the PBI test fixture to test a TRC card. After plugging in the TRC and Deformatter simulator, turn on the power. Then follow the procedure below:

Step 1: Both simulator toggle switches down (= 00), push the simulator push button switch. This action will reset the TRC and the result should be both simulator LEDs are green.

Next perform steps 2 and 3, below, (in either order). One step (the one where the LEDs blink the longest, will download the TRC Xilinx personalities. The other, the one in which the LEDs blink the shortest, will download the TRC 87C51 software).

Step 2: Simulator toggle switches = 01, push the DEF simulator push button switch. The simulator LEDs blink for a little while but stop both green.

Step 3: Simulator toggle switches = 10, push the simulator push button switch. The simulator LEDs blink for a little while but stop both green.

If, in any of the tests above, one or both of the DEF simulator LEDs stops either red or off, the step encountered a failure. Occasionally, the loading of the TRC 87C51 software will fail (it seems to depend on the DEF simulator ROM that holds the 87C51 software, some fail a lot and most never fail). If a failure does

occur, try to reload several times. If every load fails, there is a problem with the TRC. If less than one in 10 loads fail, the TRC is probably OK. The real test of the reliability of the loading function can be made with a DEF card in the test fixture.

Step 4: Both simulator toggle switches up (= 11), push the simulator push button switch. This action will send both TRC microprocessors the TESTOBS command.

The test should now be running. Press the LED RESET push button switch on the PBI test fixture and all TRACK GO/NO GO LEDs on the test fixture should go off and stay off. If one or more LEDs come back on, check that the bottom row of toggle switches on the PBI test fixture are in the handle down position. If they are not, put them in the down position and try the LED RESET switch again.

If the TRC error LEDs are all off there is still one more test to perform. The feedback shift register data predict generator has a hang-up state of all zeros. If the TRC data outputs were completely dead, the predict data generator would load with all zeros and proceed to predict all zeros. The way to test things is to momentarily switch one of the bottom row toggle switches to the up position. This action should make some of the TRC error lamps latch on. If no LEDs come on because of the switch up action, then the TRC is not being tested and this must be investigated.

Which LEDs will come on, as a function of which switch was switched to the up position, is given in the chart below:

SWITCH	LEDs
0 (left most)	all
1	all
2	all
3	all
4	tracks 4 and 12
5	tracks 5 and 13
6	tracks 6 and 14
7 (right most)	tracks 7 and 15

The actual response depends on the formatter mode (VLBA 1:4 ... VLBA 4:1). The chart above assumes VLBA 1:4 mode which is the default mode. The reason that switches 0, 1, 2, and 3 cause all track LEDs to come on is that, in the default VLBA 1:4 mode, the first 4 tracks are muxed together to initialize the predict generator. Hence if any of the first 4 tracks are bad, the initialization fails and all tracks appear be bad. Other modes are selectable via the test fixture card MONITOR.ASM program and a terminal.

This stand-alone test uses the TESTOBS.ASM program in the TRC 87C51 software. It tests the TRC to about a 90% level and is quite adequate for a quick look test of a TRC card. More complete testing of the TRC can be done with a terminal connected to the PBI test fixture.

6.7.2) TESTOBS with a Terminal

By adding a terminal to the PBI test fixture, the header contents of the PBD frames can be checked. In addition, performance can be monitored by inspecting parity error counts, CRC failures, etc. When using a terminal, two options exist, the test can either be started as above or the test can be started with the terminal. If the test is started with the terminal, two other options exist. The test can be run with both A and B TRC microprocessors or with only the A microprocessor (it cannot be run with the B micro alone).

To start the test with the terminal, with only the TRC A microprocessor, connect the terminal to the A micro and type TO (for TESTOBS) C/R. The test will immediately start. The terminal will display a formatted screen that gives the header contents of all 9 tracks microprocessor A monitors (typing S on the terminal will toggle this screen on and off). When the TRC error LEDs are reset, only the LEDs for tracks 0 thru 7 and S0 should stay off.

To start the test from a terminal with both microprocessors running is more complicated. The terminal must be switchable between the 2 micros. The procedure can be done starting with either micro first but the steps below assume starting with micro A:

Step 1: Connect the terminal to micro A and type B (no carriage return required). This action tells the A micro that both microprocessors will participate in the TESTOBS run.

Step 2: Switch the terminal to micro B and type B again. Now the B micro knows that both microprocessors will participate in the TESTOBS run.

Step 3: With the terminal still connected to micro B, type TO (carriage return). The microprocessor will print a terminal screen full and then clear the screen.

Step 4: Connect the terminal to micro A and type TO (carriage return). The test starts and the microprocessor will print the normal TESTOBS recurrent screen.

The test is now running and the error lamps can be turned off (once in a great while the synchronizing process that occurs between micro A and micro B to start TESTOBS will fail and LEDs 7 through 15 and S1 will remain lit, if this happens, try it all again). The terminal display gives the performance of TRC microprocessor A. To look at the micro B screen, type S to stop the A display, switch the terminal to micro B and type S to start the micro B display (the S command is an alternate action command).

Before the start-up process with the TRC microprocessors is initiated, the terminal can be connected to the PBI test fixture card # 1 and other test selections can be made. If H0, H1, H2, or H3 is typed, one of the 4 canned headers can be selected (H0 is the default). If O is typed, the PBI test fixture card #1 micro will start executing the test fixture OBS.ASM software. With this software running, the TESTOBS header display will show the time running and show changing AUX data. Also the test fixture card MONITOR.ASM program will allow an operator to write

directly to the AUX data field (Axxxxxxxx) or the time field (Txxxxxx) when OBS.ASM is running.

By selecting different pre-canned headers, the operator can test the TRC response to controlled parity errors, sync errors, or CRC errors (see section 6.3.1.4). To enable the frame errors to be active, type 1 (for parity errors), 2 (for CRC errors), or 3 (for sync errors) on the card # 1 terminal.

There is a modified RS232 distributor box with a toggle switch added that can facilitate TRC testing. The modification was made so that, when the toggle switch is in the down position and the distributor box selector switch is in the fully counter-clockwise position, the terminal display comes from TRC micro A but the terminal keyboard talks to the test fixture card # 1 micro. With this special mode, the test fixture can be controlled and the result on micro A can be observed simultaneously.

6.7.3) TRC Testing With AUTOTEST

AUTOTEST is, in some ways, a better test of the TRC because it uses the regular OBS software in the TRC. This software, however, does not support the terminal screen that TESTOBS does. To start this test follow the steps below:

Step 1: Connect the terminal to the PBI test fixture card # 1 micro and type O (no carriage required). The OBS.ASM program in the TRC requires the PBD frame time to be running and hence the test fixture microprocessor must itself be in "observe" mode.

Step 2: Connect the terminal to one TRC processor and type O.

Step 3: Connect the terminal to the other TRC micro and type O.

Now the two microprocessors are observing and the test fixture TRC error LEDs can be extinguished. If the LEDs do go out, test for a dead TRC by momentarily switching one of the toggle switches in the bottor row of switches on the test fixture front panel up (all these switches must be down for the LEDs to go out).

In AUTOTEST, the only terminal support is the MONITOR.ASM support and the two terminal screens supported in regular observing mode, S and TF 2.

In order to perform any test above in MKIII mode, all microprocessors participating must be put in MKIII mode (type X on a terminal connected to a given microprocessor). The test fixture LEDs will not stay off in a MKIII AUTOTEST since the header will come out of the TRC data output. TESTOBS does, however, work completely in MKIII mode.

6.8.0) DEF Test Procedure

For GO/NO-GO testing, the following script file, named ldpbitf.cmd, has been used in the past. The comments in the file provide some useful information for testing of a Deformatter in PBI Test Fixture #1. As noted in the script, Deformatter functions are limited, since the test fixture only contains a single TRC, and normal DEF operations expect two Track Recovery Cards.

```
# October 23, 1994, G. Runion    script file name= ldpbitf in hcbnew

# This script file is for loading Def and TRC in the PBI test fixture;
# the one with 1 TRC and the error lights.  After this script file loads, all
# the error lights should go out when the error light RESET switch is pushed.
# If not, try entering rs then t0 in the def terminal from the top connector.
# The TRACK lights check the output of the TRC.  These lights must be off
# before a valid check of the output of the Def can be determined by the
# CHANNEL lights.

# Normal operating files are downloaded to the TRC.  The deformatter is
# downloaded with special sequencer files.
# As of this day seq 1 and seq 2 files, (st1n1-4e.hex and st2n1-4e.hex),
# are identical to the ones that are loaded into the system, (s1n1-4e.hex and
# s2n1-4e.hex).
# Seq 3 (st3n1-4e.hex) is different from the one for the system (s3n1-4e.hex).

# This test checks the data output of both the TRC and the Deformatter.
# The TRC is given a "test obs" command.  This causes the TRC to track a
# linear model without any interaction from the Def except for receiving the
# New Delay signal.

# The Deformatter is told to barrel roll by 4.  Since the Def is in a 1-4 mode
# a roll by 4 will not effect the test but will check the roll function on
# the Def.  The 68k micro is not doing
# anything for this test except handling the barrel roll interrupt.

# The Deformatter provides CK8 and CK16 and New Delay to the TRC.

# NOTE: There is not much the 68k micro can do in this test fixture because
# all the normal observing code expects 2 TRCs.

# Useful Deformatter commands:
#   dw  - Delay Word, test the 4 Delay word Model shift registers.  This test
#         does not effect the error lights, results are displayed on CRT.
#         These are the Delay Words that normally come from the MCC in the
#         system.  See L008D26.SCH for shift register U numbers.
#   pt  - Port Test, Test the serial port that connects to the play back
#         drive.  Loop back connector must be in place.  This test does not
#         effect the error lights, the test result is displayed on the CRT.
#         SEE NOTE BELOW ON THE SWITCH BOX!
#   rs  - resest the TRC, all track error lights on.
#   to  - tell the TRC to do a test obs, all error lights off.
#   r0  - stop barrel roll, channel error lights off.
#   r4  - roll by 4 in group of 8, channel error lights off.(default)
#   r9  - roll by 4 in group of 16, channel error lights off.
#   r1  - roll by 1, channel error lights on.
#   r2  - roll by 2, channel error lights on.
#   r7  - roll by 1 in group of 16, channel error lights on.
#   r8  - roll by 2 in group of 16, channel error lights on.
#   4a  - select t0 - t3 for inputs to all the channels (4 tracks to All
#         8 channels.  The second channel (4 tracks) duplicates the first).
```

```
#    4e  - select a different group of 4 tracks to each channel
#       note: for the test fixture the sign and mag data streams for each
#             chan must be identical because of the way the output is checked.
#    1a  - select one track to all channels, channel error lights on.
#    1e  - select a different track to each channel, channel error lights on.
#       Note, 4a or 4e should turn the error lights back off after 1a or 1e.
#    et  - Emulate Terminal.  Connect the Def monitor port to the PBD port.
#          These are the 2 RS232 ports out of the DEF.
#          The Def will transparently pass all characters except for Ctrl Z
#          which terminates this mode.  This command provides an easy way for
#          the CRT connected to the DEF to be connected to all the TRC micros
#          and the micro in the test fixture.  To use remove the loop
#          back connector from the PBD connector and plug the PBD connector into
#          the common connector on the switch box.  Use the switch box to
#          choose which micro to talk to.
#          This command is also very useful in the system to verify
#          communications with a tape drive.
# SWITCH BOX NOTES:
#    Selection A - TRC micro A
#    Selection B - TRC micro B
#    Selection C - Test fixture micro
#    Selection D - Def port to PBD which is also in parallel with another
#                  connector.  This second connector can either have a
#                  loop back connector on it (required for the "pt" test)
#                  or be plugged into the common connector on the switch box.
#                  See the description for the "et" command above.
#       The normal connection for the Def's CRT in this test fixture is
#       thru the modular connector located along the top edge of the Def card.

# TEST FIXTURE TOGGLE SWITCHES:
#    The 8 TRACK switches should all be off.  If any or all the 0-3 track
#      switches are on then all the TRACK error lights will be on along
#      with the first 2 columns of CHANNEL lights.  Turning on any of the
#      4-7 track switches will result in just 2 of the TRACK error lights
#      coming on and the first 2 columns of CHANNEL lights.
#    The 8 PATTERN toggle switches determine the static pattern for the
#      track(s) selected by the TRACK switches.  The position of these
#      switches are don't care unless a track switch is on.  The selected
#      pattern entering the DEF can then be viewed on a scope.
hcbRstTarget pbibus,def
taskDelay(240)
ldseq "st1n1-4e",pbibus,def
ldseq "st2n1-4e",pbibus,def
ldseq "st3n1-4e",pbibus,def
ldctrlxil "defctrl",pbibus,def
ldbspxil "bsp-2",pbibus,def
hcbSrecLoad "defram.hex",0,0,pbibus,def
defWrServoTable (pbi_index, 160.0); /* pbi_index for bus 0, target 1 is 0 */
hcbSrecLoad("trcram.hex",1,0,pbibus,def)
hcbSrecLoad("trcram.hex",1,1,pbibus,def)
ldtrcxil 1,pbibus,def
hcbShufCtl(0x02b2c208,0x02b2c208,0x02b2c208,0x02b2c208,pbibus,def)
# Let's roll by 4.
hcbCode(pbibus,def,4,0x37,1,1,3)
# Reset the TRC
hcbCode(pbibus,def,1,0x82)
# Give the TRC time to get reset
taskDelay(240)
# Tell the TRC to do test obsv, same as 'to' from the deformatter prompt.
hcbTrcCode(pbibus,def,6,1,0x06)
# Push the error light reset switch and hope that the error lights stay off!
# If not try entering rs then t0 in the def terminal from the top connector.
```

6.9.0) PBI Test Fixture # 2

This fixture connects a MCC, two TRC and one DEF together the same as in the system. TRC inputs can come from the Mini-Transport (see Chapter 7), or from the MCC test frame. The following script is typically used to initialize this fixture:

```
# ldpbitf2.cmd
# for loading the PBI test fixture #2 containing deformatter, 2 TRCs
and MCC
# pbi_index for bus 0, target 1 is 0

hcbRstTarget 0,129
taskDelay(240)
hcbSrecLoad "mccram.hex",0,0,0,128
hcbCode(pbibus,mcc,1,0x23)
defRsSequencer (pbi_index);
defRsBitShuffFile (pbi_index);
defRsServoTable (pbi_index);

printf "%s\n",defWrSequencer(pbi_index,512,1,4)
printf "%s\n",defWrCtrlXilinx(pbi_index,"defctrl.mcs")
printf "%s\n",defWrProgram(pbi_index,"defram.hex")
printf "%s\n",trcWrProgram(pbi_index,"trcram.hex")
printf "%s\n",trcWrXilinx(pbi_index,"trc.mcs")
printf "%s\n",trcWrPrompt(pbi_index)
printf "%s\n",defWrServoTable(pbi_index,160.0)
printf "%s\n",defWrBitShuffFile(pbi_index,512,1,2)
```

In normal use, after the above default initialization, a second script is used to set up a specific mode. A scope or logic analyzer is used to look for outputs of interest. For example, the mode check out script shown in Appendix II, section II.2, would be a typical type of script to use.

Now that the testbed rack is available, this test fixture only offers the advantage of easier probing of the Deformatter card.

Chapter 7 The Mini-Transport

7.1.0) Introduction

The mini-transport is used for testing the VLBA correlator. This device simulates a VLBA play back drive (PBD) with a variety of controllable attributes. It is used in providing test signals for testing various parts of the system.

7.2.0) General Description

The mini-transport has slots for 2 Shalloway cards. The front slot seats the mini-transport logic card, which has frame formatting circuitry on it. This card develops PBD-like track signals that can drive a track recovery card (TRC) in the VLBA correlator.

The second card slot in the mini-transport holds a card that supplies the data for the PBD frames generated by the first card. At present, one of two cards (known as the Son of Barf Card and the Son of Barf Replacement Card) may be plugged into the mini-transport rear card slot depending on the test function desired from the unit.

The schematics for the Shalloway cards designed to go in the mini-transport are seen below:

card	schematic	prompt	height
mini-transport logic card	L030D01.SCH	MINI-TRANSPORT>	tall card
son of barf card	L036D01.SCH		tall card
son of barf replacement card	L036D02.SCH	SON OF BARF REPLACEMENT>	short card

The mini-transport card has the data formatters, frame sequencer, an alternate frame data source, and two microprocessors on it. This card develops eight data track signals and one system track signal, plus all required track clocks. These track signals are fanned out in the mini-transport to drive 16 TRC data track inputs and 2 system track inputs (the TRC track mix stage can be programmed to make the mini-transport provide input for all 36 track recovery chips in 2 TRCs that interface with a PBD in the VLBA correlator).

The son of barf card provides the primary frame data source for the mini-transport card. The data source can be programmed via the mini-transport front panel switches to simulate either a continuum like signal or a spectral line signal where the frequency and strength of the line is controllable. The son of barf card has a microprocessor on it that reads the front panel switches and fills a large RAM on board with a signal that complies with the switch settings.

The son of barf card replacement is a special frame data source used for mode testing in the VLBA correlator. This card plugs into the

card slot normally occupied by the son of barf card. Its output (the frame data source) comes from an on board RAM. The RAM contents are periodically output into the mini-transport logic card in the form of short bursts. The period of the RAM bursts and the initial delay on the first burst relative to a specific frame time code start point (usually midnight) is controllable.

The mini-transport actually has two sources of frame data for the simulated track signals. One data source resides on the second mini-transport card as mentioned above and the other comes from a RAM on the mini-transport logic card. In general, the dedicated data generating card is the primary data source and the mini-transport logic card data source is considered the alternate data source. The primary/alternate roles of the two data sources may, however, be interchanged by a terminal command.

In some modes of operation in the mini-transport, the data source is fixed, while in other modes the two data sources are dynamically switched during operation. In the discussions below, reference will be made to the primary and alternate data sources. Keep in mind that the primary data source is normally the high performance dedicated data generating card. The alternate data source is the primitive RAM data source on the mini-transport logic card. A terminal command can interchange the two.

7.2.1) The Mini-Transport Card

The mini-transport card generates the PBD frames. The exact format of a frame is controlled by the local microprocessor. A second microprocessor controls the clock rate of the resulting track signals.

7.2.1.0) Frame Sequencer

The frame sequencer for the mini-transport card is seen in the upper left hand corner of the schematic. The counter ICs 6A, 6B, 5B, and 5A run on a 9 MHz (nominal) clock. The two RAMs, 6C and 5C, control the frame formatter process and supply the alternate source of frame data. Both RAMs are filled by the 9A microprocessor in response to the programmed mode. The mode can be set either from the mini-transport front panel switches or via a terminal communicating with the microprocessor over a serial link.

Once the RAMs are filled, the RAM address counter free runs with a radix set by a code in the 6C RAM. Two lengths are standard, one for VLBA mode and one for MKIII mode. The 6C RAM supplies the logic on the card with timing and control signals that allow the logic to develop PBD frames. The 5C RAM holds an alternate data source for the track frame data (normally the frame data come from the other card in the mini-transport). The eight outputs of the 5C RAM supply the data for the 8 tracks generated on the card. This alternate data source is selected two ways, first, via a command on the terminal connected to the 9A microprocessor and, second, during special circumstances. One example of such a special circumstance is when the system is programmed to make frame parity errors. If an isolated frame is programmed for ,say, 50 data

parity errors, the data for that isolated frame will come from the alternate data source. See section 7.3.0.0.4 for more detail.

7.2.1.1) Frame Generation

ICs 4A, 4B, 4C, and 4D are the formatter PALs in the mini-transport. Each PAL provides 2 track signals that meet the specifications of VLBA or MKIII PBD frames. The formatter PALs receive data signals, header signals, CRC signals and control signals from various parts of the card and puts these signals together to form PBD frames. The PALs themselves encode the parity and do the NRZM encoding (NRZM encoding is the differential scheme used by the VLBA and MKIII transports at record time).

Data for a frame comes from either the on board 5C RAM or from the second card in the mini-transport. Data generated in the 5C RAM goes through PAL 2E which performs a barrel roll function if so programmed.

Header data comes from RAM 7C after being supplied by the 9A microprocessor. ICs 7B and 7D provide support in the header generation process. IC 5E generates the CRC portion of the header.

IC 6D decodes the sequencer timing signals and the mode information to provide control signals required by the frame generation process. The START signal on 6D-14 is used to start the son of barf card replacement card data generator on a specific frame time tag (usually midnight). The ODD FRAME signal on 6D-16 is used by IC 5D, in the lower right hand corner of the mini-transport logic card schematic, to switch between the main and alternate data sources for the track frames. As explained above, this signal can be dynamic in causing an isolated frame to have a different data source from the other frames. (Also see section 7.2.2.3.)

The CONT[0..2] bus output by IC 6D is the barrel roll programming counter required by IC 2E. The other signals output by IC 6D are timing signals used in the frame generation process.

IC 5D, mentioned above, decodes the sequencer timing signals and the mode information to produce the gated clock GCLK from 5D-17. This clock drives the data generator card in the mini-transport. The clock is gated so that the data source card will not clock during the header time of the frame or during parity bits. Thus the data source meets the non-data replacement specification of the VLBA.

For low data rate modes (VLBA 2:1 or VLBA 4:1) GCLK also gates the data card so it runs at an appropriate rate. The M0, M1, and M2 control signals output by IC 5D control the output stage of the son of barf card so that the mode set on the mini-transport front panel switches is obtained. For low data rate modes, the frame data source switches between the external data card and the internal alternate (RAM 5C) data source. For VLBA 4:1 mode, for example, only one of the 4 "signals" multiplexed onto the one track signal has the primary data source, the other 3 have the alternate data source.

7.2.1.2) The Frame Control Microprocessor

IC 9A is an 87C51 microprocessor that controls most of the operations on the mini-transport card. This microprocessor reads the front panel switches and configures the logic on the board to support the mode selected. Additional mode control can be implemented with a terminal connected to the micro on its serial port. The microprocessor fills the 6C and 5C RAMs. The 6C RAM is set for the selected VLBA or MKIII mode, and 5C is loaded with a pseudo random pattern. The 5C RAM can, however, be modified in a variety of ways with the microprocessor monitor subroutine via the terminal (see section 7.3.0.0.4).

After the set-up is complete, the microprocessor provides the frame header information for the 7C header RAM. During the data portion of each frame the microprocessor writes data for the next header in the 7C RAM. In this way the frame header time code runs as it would in a real formatter. By selecting the time increment between frames, the microprocessor can simulate the three standard record rates used in the VLBA.

7.2.1.3) The Clock Control Microprocessor

IC 11A controls the frequency of the 74LS624 VCO. The micro looks at the VLBA/MKIII mode line from the 9A microprocessor and sets the VCO for the appropriate frequency. The microprocessor has a serial port over which it can communicate with a terminal or with a Deformatter card. A terminal can be used to select clock frequencies other than the standard frequencies.

The microprocessor also supports the speed control protocol of a PBD when communicating with a Deformatter card from the VLBA correlator. With this feature, the mini-transport can be "slewed" in speed like a PBD. As the Deformatter card issues speed commands in its PBD servo control role, the 11A microprocessor changes the track clock rate in compliance with the commands. Thus the Deformatter can servo the mini-transport in a simulated observation.

The mini-transport has 5 LEDs on its front panel to display the servo action of the Deformatter commands. The 11A microprocessor drives these LEDs as commands from the Deformatter are received (see section 7.3.0.1.2).

7.2.1.4) Track Fanout

The logic in the upper right hand corner of the mini-transport card schematic provides fanout of the 8 generated track signals (and clocks) to the TRC. Conversion of the track and clock signals to differential ECL occurs here.

7.2.2) The Son of Barf Card

The son of barf card is the primary data source for the frames generated in the mini-transport. Data from this card can be made to look like either continuum or spectral line data. For spectral line data, the frequency and strength of the line can be controlled.

The source of continuum data (and the continuum part of spectral line data) comes from very long feedback shift register pseudo random generators. The source of the spectral line component of the data card output comes from the NCO seen in the lower right hand corner of the schematic. The RAMs are addressed by both the NCO output and the random data generators and their output produces the desired output data streams.

7.2.2.0) The Pseudo Random Data Generators

Feedback shift register pseudo random data generators supply the continuum part of the signals simulated by the card. There are four such data generators (seen at the top of the schematic). The data generators are made to shift the equivalent of 8 bits every clock (see the PAL ABL files) and hence each data generator has 8 independent (non-correlated) outputs. The 4 times 8 pseudo random outputs drive the four RAMs 2A, 2B, 2C, and 2D with each RAM getting a 6 bit "noise signal". (Think of the 6 bits of random data generator that each RAM gets as a random noise voltage digitized in a 6 bit A/D.)

7.2.2.1) The NCO

The NCO (number controlled oscillator) is seen in the lower right hand corner of the card schematic. The A[8..15] bus is supplied by the microprocessor and sets the "frequency" of the NCO. The number equivalent of this bus is added to the oscillator phase (PHASE[0..9]) every clock cycle. The PHASE[0..9] signal can be thought of as a 10 bit phase code which rotates through 360 degrees with a period set by the A[0..15] phase increment. The 10 bit oscillator phase drives the RAMs with the phase of the spectral line.

7.2.2.2) The Microprocessor

IC 6A controls the activity of the card. Every time it is reset it performs the following:

1) It reads the position of the front panel switches.

2) It fills the RAMs with contents consistent with the spectral line strength selected.

3) It sets the A[8..15] NCO frequency consistent with the spectral line frequency selected.

4) It pulses the OE1\ line (when filling the RAMs) which starts the pseudo random data generators (insuring that none are in the all zeros hang up state).

After this action, the microprocessor has no other tasks and hangs up in a endless loop. Note that the card only gets initialized upon a reset to the microprocessor. Hence, front panel switch change will not get be implemented until the son of barf card is reset by pushing the bottom push button switch (upper left hand corner) on the mini-transport front panel (labeled INIT).

7.2.2.3) The Data RAMs

RAMs 2A, 2B, 2C, and 2D supply the mini-transport logic card with frame data. The address to these RAMs consists of a 10 bit spectral line phase and a 6 bit noise signal. The card microprocessor, in compliance with the mini-transport front panel switches, programs the RAMs with a look-up sampler output table for every combination of line amplitude (as derived from the line phase expressed by the NCO output and the line strength implied by the CORR COEF switch) and noise amplitude (as expressed by the random data generators).

The RAM outputs are, hence, simulated sampler outputs. The two output pins of each RAM represent the 2 bit outputs from a 2 bit sampler. In VLBA 1:4 mode, the four RAMs are understood to have 90 deg phase offsets in their NCO address lines (i.e., RAM 2A is programmed by the card microprocessor as if the 10 bit PHASE[0..9] phase were exact, RAM 2B is programmed as if the 10 bit PHASE[0..9] phase had an understood 90 deg shift, RAM 2C is programmed as if the 10 bit PHASE[0..9] phase had a 180 deg shift, and RAM 2D is programmed as if the 10 bit PHASE[0..9] phase had a 270 deg shift). This action yields an effective phase resolution of 12 bits instead of the 10 bits output by the NCO. With the samples so simulated in the RAMs, the Deformatter card in VLBA 1:4 mode will multiplex four tracks together and the four RAM outputs will become 4 consecutive samples as if taken at 32 MHz (with the appropriate phase separation).

In VLBA 1:2 operation, a similar operation is performed except that RAMs 2A and 2C are considered to have zero phase shift (from the PHASE[0..9] phase) and RAMs 2B and 2D are 180 deg shifted from them. Now the Deformatter in VLBA 1:2 mode will take coincident samples from two adjacent tracks and multiplex them together to represent two consecutive samples taken at 16 MHz.

Output PAL 1C implements some of the mode selection. The mode table seen just below this IC illustrates the action of this PAL. Signal M2 into the PAL is the output enable for the chip. This line, and its application on IC 2E on the mini-transport card, provides the mechanism of switching between the primary and alternate data sources in the unit. For low bandwidth modes, like VLBA 2:1 or VLBA 4:1, this line is active and results in only one of the two (for VLBA 2:1 mode) or one of the 4 (for VLBA 4:1 mode) time multiplexed channels of a track getting the primary data source. The other channel(s) get the alternate data source.

7.2.3) The Son of Barf Replacement Card

The son of barf replacement card provides data for the mini-transport of a different sort. This card has a small data RAM and the contents of this RAM can be used to drive the mini-transport logic card in periodic bursts. The phasing of the initial burst (with respect to a specific frame time code) and the period of the bursts can be precisely controlled.

This card is useful for testing the VLBA correlator where signals that repeat themselves every 4.128 msec fringe cycle, or every 131 msec integration cycle are needed. The card was used in the checkout of the correlator modes, models, FSTC performance and for testing the pulsar gate.

7.2.3.0) The Initial Burst and Burst Period Control

ICs 3B, 3A, 4B, and 4A control the timing of the data bursts output by the card. These ICs form a 32 bit programmable radix counter with a programmable initial cycle down counter.

The microprocessor loads the two registers on these chips with two 32 bit codes. The code that goes in the counter part of the ICs is the initial delay count. When the START signal arrives from the mini-transport card (being timed on that card to coincide with a pre-programmed frame time code, usually midnight) the counter counts until terminal count (3B-9 low) at which time the first data burst is emitted. After the first delay, the 32 bit counter reloads from the internal 32 bit secondary storage register and the data bursts are emitted at the set period. IC 4C provides control logic for this process.

7.2.3.1) The RAM Data Source

ICs 2A, 2B, and 2C count through the RAM addresses every time an enable signal appears on 2A-10. When the counter overflows, the STOP signal halts the counter and it waits for the next cycle. RAM 1B is a 8K X 8 memory but only 4K X 8 is used. It is initially filled by the microprocessor and thereafter waits for IC 4C to start the readout sequence.

7.2.3.2) The Microprocessor

IC 3D is an 87C51 microprocessor that programs the RAM data generator and the initial-delay/burst-period counter. It has a serial port for terminal communication. After the initial programming, the micro has no functions and hangs up in an endless loop waiting for additional terminal commands.

7.3.0) Software

There are a total of 4 87C51 microprocessors used in the various cards of the mini-transport. The text below describes the software provided for these micros.

7.3.0.0) Mini-transport Card Microprocessor 9A Software

The main microprocessor in the mini-transport logic card is the 9A micro. A summary of the software modules written for this microprocessor is seen below (as of 9/10/92).

Section Name	Starting Address	Ending Address	Size
A:MASTER.ASM	0000	0325	0303
A:TEST.ASM	0326	0441	011C
A:TIME.ASM	0442	05BC	017B
A:ERROR.ASM	05BD	0679	00BD
A:MONITOR.ASM	0800	0FFB	07FC

7.3.0.0.0) MASTER.ASM

MASTER.ASM is entered upon a reset to the 9A microprocessor. First, the card is initialized. The three main initialization subroutines are FRAME, RDATA, and HEADER. FRAME will fill the 6C frame sequencer RAM. At reset, VLBA mode is assumed and all errors are cleared (some errors, such as data parity errors must be stored in the 6C RAM).

RDATA will fill the 5C RAM with a 17 bit pseudo random data pattern. The contents of the 5C RAM can be subsequently modified by using the monitor program.

The HEADER subroutine performs the initial 7C header RAM load. The time code inserted into the header is seen in TABLE1. After reset, the time code can be changed to anything.

Once the card initialization is complete, the microprocessor goes into a loop waiting for interrupts.

7.3.0.0.1) TEST.ASM

TEST.ASM contains 4 software tests that can be used to test the mini-transport logic card. TEST0 will loop testing the 6C frame sequencer RAM. TEST1 will loop testing the 5C data RAM. TEST2 will loop testing the 7C header RAM. TEST3 will loop testing all three of these RAMs. At the completion of each loop a cumulative errors count is printed on the terminal. To escape a test routine, press the space bar.

7.3.0.0.2) TIME.ASM

The TIME.ASM software module is executed every time an INT0 interrupt occurs, i.e., every time a new PBD frame starts. When a new frame starts, this software reviews the error conditions set (see section 7.3.0.0.3) and performs support needed for the error conditions selected. The software then updates the header time code so that the next header will have the correct time. The new time is then loaded into the 7C RAM to wait for the end of the frame.

7.3.0.0.3) ERROR.ASM

ERROR.ASM is logically part of the monitor routine. The two entries to the ERROR.ASM routine are via a terminal E or A command. The form of the E command is Exy where x is the number of consecutive frames in which an error is to be made and y is the type of error to be made. An upper case E is shown for clarity but the monitor is case insensitive.

There are eight categories of errors that the mini-transport can simulate as seen in the table below;

y	error condition
0	long frame
1	short frame
2	2 bit sync error
3	3 bit sync error
4	one second positive time step
5	one second negative time step
6	data parity errors
7	tape splice

The long and short frame errors will cause the frame sequencer to run one parity byte (9 bits) long or short. This type of error will test the TRC cards ability to recover from a sync slip. The 2 and 3 bit sync errors will test the TRCs ability to recover from bad sync. A 2 bit sync error is still detectable by the Xilinx off-line sync detector but the 3 bit error is not.

The time code time step errors test the TRC software's ability and the Deformatter servo's ability to recover from a big discontinuity in the PBD data.

Controllable frame data parity errors can be used in many ways to test the TRC hardware and software. The number of sync errors made per frame due to the Ex6 command is set with the monitor B command (see section 7.3.0.0.4).

The tape splice error will cause a big discontinuity in the frame data and test the systems ability to recover.

The x parameter of the E monitor command will cause the error condition to be repeated for x consecutive frames. The monitor A command (no carriage return needed) will cause the last Exy sequence to be repeated.

7.3.0.0.4) MONITOR.ASM

The MONITOR.ASM software module is instrumental in setting all of the PBD attributes the mini-transport is capable of supporting. Below is a summary of the more important monitor commands;

The D P, and M monitor commands Display, Page, or Modify the 6C, 5C, or 7C card RAMs. The target RAM is determined by typing the 1 key (for 6C), the 2 key (for 5C), or the 3 key (for 7C). No carriage return if required after the 1, 2, or 3 key entry. The Display or Modify operations are disruptive to the mini-transport frame generation. Also, the W command (no carriage return required) switches the D and P commands between the external RAMs as described above and the microprocessor memory.

On reset, the microprocessor writes a pseudo random pattern into the 5C frame data RAM. This RAM can be modified with two monitor commands besides the M command. The Fxx monitor command will fill the RAM with the xx byte (i.e. every 8 bit RAM location will have xx written into it). The Nx yyyy monitor command will write the 16 bit word given by yyyy lengthwise down bit x of the 8 bit wide RAM (x = 0 thru 7). If, for example, the command N3 45 is typed, bit 3 of RAM 5C will repeat the data pattern 45 (hex) every 16 RAM addresses. The other 7 bits of the RAM will remain unchanged after an N command.

An Ixxyy monitor command will set the time increment from frame to frame as xxyy (xx is stored in RAM location 72 and yy is stored in location 73). The only legal commands are I0025 (for VLBA 8MHz, the default setting), I0050 (for VLBA 4 MHz or MKIII), and I0100 (for VLBA 2 MHz).

The Exy, Bnn, and Cnn monitor commands set the various controllable error modes of the mini-transport. The Exy command was described in section 7.3.0.0.3. The Bnn command sets the number of frame parity errors to be made by each E16 monitor command. The Bnn command will set up nn parity errors by writing nn bad parity encodes in nn RAM 6C memory locations. This process is disruptive to the mini-transport frame generating operation but otherwise the Bnn command has no other immediate effect. The Cnn monitor command, in addition to messing up cable TV reception, establishes a loop of nn frames in software. Every nn frames the last (or future) Exy command is reinstated.

As an example of the Exy, Bnn, and Cnn commands, consider the monitor command sequence E46 B23 C10. After typing this sequence on the terminal, the following mini-transport operation will start:

> Every 16 frames (set by the C10 command), the mini-transport will output 4 consecutive frames (the 4 being set by the E46 command) with 23 (hex) data parity errors (the 23 being set by the B23 command and the parity errors by the E46 command). This operation will continue until changed in some way. If, for example, E16 were typed, the 4 consecutive bad frames would go down to one frame with 23 parity errors every 16 frames. If C00 is typed, the error sequence stops.

The Exy command always starts on an even second when the barrel roll is enabled. When an error frame is transmitted, the data source for the frame is switched. That is, whatever data source (RAM 5C or the other mini-transport card) is selected during normal frames, the other data source is selected for the bad frame(s). One reason for this switching of data sources during frames containing errors is to "mark" the error frames. For example, if the parity errors in each frame with errors are enough for the TRC to tag them as invalid, the effect of the exclusion by the FFT card of invalid data can be readily seen by the lack of contamination of the FFT card output spectra by the spectra that would result from the other data source.

The monitor command A* (the * indicates that no carriage return is required) was discussed in section 7.3.0.0.3. The monitor command S* is used in conjunction with the son of barf replacement card and will be discussed in section 7.4.0.

The monitor command V* turns the barrel roll on and off. The roll is disabled upon reset and the V key toggles the roll state on\off. Only a roll of 8 is supported. The monitor command X* interchanges the primary and alternate frame data source (see section 7.2.0). The monitor command Y* switches the mini-transport between the VLBA and MKIII modes.

7.3.0.1) Mini-transport Card Microprocessor 11A Software

The clock microprocessor in the mini-transport logic card is the 11A micro. A summary of the software modules written for this microprocessor is seen below (as of 9/10/92).

Section Name	Starting Address	Ending Address	Size
A:CLKMAS.ASM	0000	00AF	008D
A:CLKMON.ASM	00B0	0405	0356
A:CLKSPEED.ASM	0406	04C6	00C1
A:CLKTST.ASM	0800	0811	0012

7.3.0.1.0) CLKMAS.ASM

CLKMAS.ASM is entered upon a reset to the 11A microprocessor. The UART is initialized and then the microprocessor tests the VLBA/MKIII bit (INT1). The micro then sets the VCO clock rate appropriate to the mode. After this setup, the microprocessor will loop testing the VLBA/MKIII flag and change the VCO frequency if so instructed by the 9A micro. Otherwise the microprocessor will loop doing nothing until an interrupt from the UART is received. The UART port supports either communication with a terminal or with a Deformatter card.

When connected to a terminal, the clock microprocessor provides a limited monitor command set. When connected to a Deformatter card it supports the speed control protocol of a PBD.

7.3.0.1.1) CLKMON.ASM

The CLKMON.ASM software module offers a limited set of commands to a terminal operator. About the only important monitor function are the ^ and v commands (no carriage return required). These commands will increment or decrement the VCO frequency.

7.3.0.1.2) CLKSPEED.ASM

CLKSPEED.ASM provides a simulation of the PBD speed protocol. The Deformatter card serial port that is used to control the speed of a PBD in the system, can be connected to the microprocessor 11A serial port. As the Deformatter observes, using the mini-transport as a source for the TRC input, the Deformatter can servo the speed (clock frequency) of the mini-transport. This operation simulates the normal observing process of the VLBA correlator PBI cards.

As speed commands from the Deformatter card are received by the 11A microprocessor, it will drive the mini-transport front panel LEDs so an operator can monitor the operation of the PBI. Five LEDs are in place with the two outer red LEDs meaning reception of large speed control commands (speed up in one direction and slow down in the other direction). The 2 yellow LEDs indicate the reception of smaller speed commands and the green LED indicates the reception of a very small speed change command. Each LEDs stay lit for the duration of the Deformatter speed change command. (In general speed control commands with small speed changes are effective for short times and speed control commands that force large speed changes are effective for longer times.)

7.3.0.1.3) CLKTST.ASM

CLKTST.ASM provides hooks for installing software tests of the card if any are ever necessary. Otherwise, this software module is empty.

7.3.0.2) Son of Barf Card Software

A summary of the software modules written for the son of barf card microprocessor is seen below (as of 9/10/92).

Section Name	Starting Address	Ending Address	Size
A:MASTER.ASM	0000	0017	0018
A:SIGNAL.ASM	0018	022F	0218
A:CC.ASM	0700	07FF	0100
A:SINE.ASM	0800	0FFF	0800

7.3.0.2.0) MASTER.ASM

MASTER.ASM is entered upon a reset to the son of barf card microprocessor. The only function of the routine is to fill the data rams. After this action the micro hangs up in an endless loop. There are no interrupts in this microprocessor application.

7.3.0.2.1) SIGNAL.ASM

SIGNAL.ASM is the main software module in this microprocessor. Its function is to read the mini-transport front panel switches and fill the card data generation RAMs with a signal that complies with the switch settings.

The switches that this microprocessor reads are the LINE FREQ and the CORR COEF rotary switches. The CORR COEF switch is used set the line intensity. The program calculates the RAM contents for each RAM location as follows;

For a given RAM location, the portion of the RAM address set by the PHASE[0..9] address lines is taken as a spectral line phase. A sinusoidal amplitude for this phase is looked up in the SINE.ASM table. The 6 RAM address pins connected to the pseudo random generator outputs are taken as the 6 bit digitization of a noise voltage. The microprocessor then calculates the value

$$V = (CC)x(S) + (1-CC)x(N)$$

where CC is the correlation coefficient from the CORR COEF switch setting, S is the voltage of the signal sine wave (obtained from the SINE.ASM look table from PHASE[0..9]), and N is the 6 bit noise voltage. The V result is then sampled with a 2 bit resolution and the resulting 2 bits are stored in the RAM location. The microprocessor does this for all 64K RAM locations.

RAMs 2B, 2C, and 2D are treated as if the PHASE[0..9] phase were phase shifted as described in section 7.2.2.3.

The LINE FREQ and CORR COEF rotary switches each provide 6 preset selections for the line frequency and line intensity. FREQ 1 is the lowest line frequency and FREQ 6 is the highest. Likewise, LEVEL 1 is the lowest line intensity (in fact LEVEL 1 is continuum) and LEVEL 6 is the highest line intensity. The DELTA UP and DELTA DOWN switch positions allows an operator to get intermediate values of line frequency and intensity from the 6 set levels. Each time the INIT reset switch is pushed with a rotary switch in the DELTA UP or DELTA DOWN positions the line frequency or intensity is incremented up or down a small amount. The LINE FREQ switch is used to set the NCO rate.

The front panel switches are read only after the reset switch is pressed. Hence, in order to change the line frequency or the line intensity, the lower push button switch (INIT) must be pressed after the desired switch changes.

7.3.0.2.2) CC.ASM

CC.ASM is a look up table of square root values.

7.3.0.2.3) SINE.ASM

SINE.ASM is a look up table of SINE values (from 0 to 90 deg. This table was generated by a C program SINE.C (executable program was SINE.EXE).

7.3.0.3) Son of Barf Replacement Card Software

A summary of the software modules written for the son of barf replacement card microprocessor is seen below (as of 9/10/92).

Section Name	Starting Address	Ending Address	Size
A:MASTER.ASM	0000	0095	0073
A:HELP.ASM	0096	0713	067E
A:MONITOR.ASM	0800	0D2D	052E
A:TEST.ASM	0D2E	0E42	0115

7.3.0.3.0) MASTER.ASM

MASTER.ASM is entered upon a reset to the son of barf replacement card microprocessor. This subroutine initializes the UART and a few card functions. After setup, it goes into a loop waiting for a UART interrupt.

7.3.0.3.1) HELP.ASM

HELP.ASM provides a help screen for the terminal operator.

7.3.0.3.2) MONITOR.ASM

The MONITOR.ASM software module supports the terminal monitor routine. The D and P monitor options can be used to display either the microprocessor memory or the 1B data RAM. The S* (* means no carriage return required) switches between the two. The M command can be used to modify the 1B RAM.

The L monitor command is the most useful monitor option. This command displays or loads the 32 bit rep-rate code (in micro RAM locations 50, 51, 52, and 53) and the 32 bit delay code (in micro RAM locations 54, 55, 56, and 57). A bare L C/R will display these 8 RAM locations and a Lxxxxxxxxyyyyyyyy (C/R) will load the rep-rate code with xxxxxxxx and the delay code with yyyyyyyy. (see section 7.4.0 for more details.)

The TM monitor option will loop testing the 1B RAM. Fxxxx will fill the 1B data ram with a pseudo random data pattern where the xxxx code is the data generator seed (xxxx = 0000 will fill the RAM with all zeros).

7.3.0.3.3) TEST.ASM

TEST.ASM provides a series of tests to both test the card and use it operationally. TEST0 (type T0 C/R) will take the RAM 50..57 rep-rate and delay codes and load then into the 6A, 6B, 5B, 5A counter.

TEST1 loads a rep-rate of one frame and a delay of one frame into the 6A, 6B, 5B, 5A counter. It also writes all zeros into the 1B data RAM except for RAM location 010 where it writes an FF. The result of TEST1 will be to have track data from the mini-transport of all zeros except for the very first bit of every frame which will be a logic one (actually, to get this condition a sequence of commands is necessary; 1) type S* on the mini-transport logic card terminal, 2) type T1 on the other terminal, 3) type S* again on the first terminal).

TEST2 is the same as TEST1 above except that the rep-rate and delay are one 4.128 msec delay cycle. The result of this test is that bit one of every new delay cycle will be a logic one, all other bits will be zero (again a sequence of commands is required).

7.4.0) How to Use the Mini-Transport with the Son of Barf Replacement Card

In order to use the mini-transport with the son of barf replacement card, the procedure below must be followed. The set up involves communication with both the mini-transport logic card 9A microprocessor and the son of barf replacement card microprocessor. Both must be connected to terminals (or one terminal must be moved between them).

1) Type an S* (* means no carriage return required) on the mini transport logic card terminal (the micro responds with TOP spelling STOP on the terminal, if it responds TART spelling START on the terminal, hit the S* key again). This action stops the mini-transport header time increment and puts the mini-transport in halt mode.

2) Type in the rep-rate and delay desired on the other terminal into the 50..57 RAM location of the son of barf replacement card (or do TEST1 or TEST2 for canned times of one frame or one delay cycle). The rep-rate and delay values can be entered with the Lxxxxxxxxyyyyyyyy monitor command, where the xxxxxxx is the complement of the 32 bit rep-rate in track (8 MHz) bits and yyyyyyyy is the complement of the 32 bit delay in track bits. If, for example, LFFFFB1E1FFFFB1F0 is entered, the FFFFB1E1 (= (2**32 - 20000) + 1) will set the rep-rate at 20000 bits (one frame). The FFB1F0 (= (2**32 - 20000) + 16) will result in one frames initial delay (plus 16 bits so that RAM 1B memory address 010 will be the first bit out).

3) Fill the 1B RAM with the desired data pattern. If TEST1 or TEST2 were used in step 2, the RAM has all zeros written into it

except for loc 010 which has an FF in it.

4) Type T0 on the son of barf replacement card so that the 50..57 RAM codes loaded in step 1 will get written into the 6A, 6B, 5B, 5A counter. If TEST1 or TEST2 were used in step 2, this step is unnecessary.

5) Type S* on the mini-transport logic card terminal (it responds with TART spelling START on the terminal). This action will start the counter on an even frame. Then yyyyyyyy + 16 frame data bits later RAM location 010 drives the mini-transport logic card and the contents of the RAM appears as data in the output frames. The RAM contents are repeated every xxxxxxxx bits.

APPENDIX I LIST OF FILES

NOTE: Full path names given here are as of early 1998

I.1.0) Track Recovery Card

I.1.1) Files maintained in corrdwgs/trc/sch/SCCS

FILE NAMES	DESCRIPTION	NR FILES
L007D01.SCH - L007D04.SCH	TRC SCHEMATIC	4
Z007D01.LAY - Z007D02.LAY	TRC IC LAYOUT	2
L007D01.SRC	ORCAD LIB SOURCE	1
L007D01.LIB	ORCAD LIB	1
W004T01.NET	TRC WIRELIST	1
B002T01.PRT	TRC PARTS LIST	1
K008D01.BLK - K008D10.BLK	DPC XILINX DESIGN	10
K009D01.BLK - K009D10.BLK	DELAY CONTROL XILINX DESIGN	10
K010D01.BLK - K010D09.BLK	HEADER CONTROL XILINX DESIGN	9
PAL_U29.SCH	TRK MIX PAL, U28, U29	1
PAL_U33.SCH	TRK MIX PAL, U32, U33	1
PAL_U35.SCH	TRK MIX PAL, U30, U31, U34, U35	1
PAL_U36.SCH	TRK MIX PAL, U36	1
D028TRC.FG1	HCB PROTO, FIG 1 TIME TRACKING	1
D028TRC.FG2	HCB PROTO, FIG 2 VALIDITY	1
WRITE.DPC	WRITE OPS INTO TRK BUFFER	1
		46 TOTAL

I.1.2) 87C51 source files maintained in vlbsoft/trcasm/SCCS

FILE NAMES	DESCRIPTION	NR FILES
Makefile	creates trcrom.hex, trcram.hex	1
*.asm	see SOURCES list in Makefile	13
d028trc.hcb	HCB/TRB protocol	1
trcmem.doc	normal memory assignments	1
trctestmem.doc	test mode memory assignments	1
		17 total

I.1.3) PAL files maintained in vlbsoft/pal_prom/pals/trc/SCCS

FILE NAMES	DESCRIPTION	NR FILES
pal_u29.abl	ABEL src, trk mix, U28,29	1
pal_u33.abl	ABEL src, trk mix, U32,33	1
pal_u35.abl	ABEL src, trk mix, U30,31,34,35	1
pal_u36.abl	ABEL src, trk mix, U36	1
u29.jed	JEDEC file, U28 and U29	1
u33.jed	JEDEC file, U32 and U33	1
u35.jed	JEDEC file, U30,31,34 and U35	1
u36.jed	JEDEC file, U36	1
		8 total

I.1.4) Sequencer source files maintained in vlbsoft/pal_prom/proms/trcseq/SCCS

FILE NAMES	DESCRIPTION	NR FILES
Makefile	creates u72u73.hex	1
vlbaseq.asm	source for VLBA header format	1
mkiiiseq.asm	source for MKIII header format	1
u72u73.hex	used by PROM Programmer	1
		4 total

I.1.5) Xilinx files maintained in vlbsoft/xilinx/trc/SCCS

FILE NAMES	DESCRIPTION
mktrcmcs.DOC	Notes detailing the design steps: MUST READ
cont_ur.mac	original macro for Header Control DESIGN
cont_ur.lca	un-routed result of macro above
cont_ar.lca	routed result of apr on lca above (the above was from some previous XACT version)
contmod.mac	final macro, run on lca above (at first XACT prompt, saves control.lca)
control.lca	final lca design from steps above
control.bit	from: makebits -t control.lca (ttl inputs)
dpc_ur.mac	original macro for DPC DESIGN
dpc_ur.lca	un-routed result of macro above
dpc_ar.lca	routed result of apr on lca above (the above was from some previous XACT version)
dpcmod.mac	final macro, run on lca above (at first XACT prompt, saves dpc.lca)
dpc.lca	final lca design from steps above
dpc.bit	from: makebits -tc dpc.lca (cmos inputs)
delay_ur.mac	top level XACT macro for Delay Control design
delay_ur.lca	un-routed result of macro above
delay.lca	result of: apr -a3 -r4 -s5433 -c lockio.cst delay_ur.lca delay.lca
delayapr.bat	contains the apr command line above
lockio.cst	constraint file to lock I/O pins
delay.rpt	apr report from above
delay.bit	from: makebits -t delay.lca (ttl inputs)
mktrcmcs.mac	macro to generate trc.mcs from control.bit plus dpc.bit plus delay.bit
trc.mcs	result of mktrcmcs.mac and dos2unix (this is the final product)

I.2.0) PBI Test Fixture #1

I.2.1) Orcad files maintained in corrdwgs/testfix/re/sch/SCCS

FILE NAMES	DESCRIPTION
L010D01.SCH	PBI Test Fixture #1, Card #1
L011D01.SCH	PBI Test Fixture #1, Card #2
L022D01.SCH	PBI Test Fixture #1, Substitute Deformatter Card
L023D01.SCH	PBI Test Fixture #1, Interconnect diagram
L046D01.SCH	PBI Test Fixture #1, Clock generator

I.2.2) Source files maintained in corrdwgs/testfix/re/pbi_x875

This directory contains assembly source files for the 87C51 processor in PBI Test Fixture #1, Card #1. If also contains source files for the ROM 7C firmware on the card. None of these files have been placed under SCCS.

Refer to the read.me file in the directory for a list of the files and other comments.

I.2.3) Source files maintained in corrdwgs/testfix/re/pbi1seq

This directory contains assemble source files used to generate the firmware for the ROM 10C and 11C frame sequencer roms on PBI Test Fixture #1, Card #1.

Refer to the read.me file in the directory for a list of the files and other comments.

I.2.4) ABEL files maintained in corrdwgs/testfix/re/pbi1pal

This directory contains ABEL source files for the PALs on PBI Test Fixture #1, card #1.

Refer to the read.me file in the directory for a list of the files and other comments.

I.2.5) Files maintained in corrdwgs/testfix/re/pbi2pal

This directory contains files for PBI Test Fixture #1, Card #2. There are ABEL files for the PALs and source files for firmware in the ROM 8D sequencer.

Refer to the read.me file in the directory for a list of the files and other comments.

I.2.6) Files maintained in corrdwgs/testfix/re/def_simulator/SCCS

FILE NAMES	DESCRIPTION
Makefile	produces simrom.hex from simaster.asm
simaster.asm	Source code for the 87C51 on the Deformatter simulator
simrom.hex	Hex file for programming the 87C51

I.3.0) Mini-transport

I.3.1) Files maintained in corrdwgs/minix/sch/SCCS

FILE NAMES	DESCRIPTION
L030D01.SCH	Mini-transport "front" card
L036D01.SCH	Mini-transport "back" card, TALL (son of barf card)
L036D02.SCH	Mini-transport "back" card, SHORT, IC layout included (replacement son of barf card)
Z036D01.LAY	IC layouts for L030D01.SCH and L036D01.SCH
MINIX.SRC	Orcad source for L030D01, L036D01, L036D02.SCH
MINIX.LIB	Orcad library for the above

I.3.2) Files maintained in corrdwgs/testfix/re/mini_pal

This directory contains ABEL and JED files for the PALs on L030D01.SCH.

Refer to the read.me file in the directory.

I.3.3) Files maintained in corrdwgs/minix/pgms/SCCS

There is a Makefile in this directory that is used to generate the hex files for programming the two 87C51 processors on L030D01.SCH at locations 9A and 11A.

See the Makefile for a list of the .ASM source files that are also maintained here.

I.3.4) Files maintained in corrdwgs/testfix/re/son_barf

This directory contains ABEL source files, assembly source files and C source files for the L036D01.SCH "son of barf" card.

There is a SCCS directory that some of the files have been checked into.

Refer to the read.me file in the directory for details.

I.3.5) Files maintained in corrdwgs/testfix/re/barf_rep

This directory contains ABEL and assembly source files for the L036D02.SCH "son of barf replacement" card.

Refer to the read.me file.

I.4.0) Deformatter Card

I.4.1) Files maintained in corrdwgs/def/SCCS

FILE NAMES	DESCRIPTION	NR FILES
L008D01.SCH - L008D31.SCH	DEFORMATTER SCHEMATIC	31
Z009D01.LAY - Z009D02.LAY	DEFORMATTER IC LAYOUT	2
DEFTOP.PAL	TOP LEVEL FOR PALS	1
K014D01.PAL	PAL REPRESENTATIVE LOGIC	1
K015D01.PAL - K015D02.PAL	PAL REPRESENTATIVE LOGIC	2
K019D01.PAL - K019D02.PAL	PAL REPRESENTATIVE LOGIC	2
K020D01.PAL	PAL REPRESENTATIVE LOGIC	1
K021D03.PAL	PAL REPRESENTATIVE LOGIC	1
K022D01.PAL - K022D02.PAL	PAL REPRESENTATIVE LOGIC	2
K023D01.PAL - K023D02.PAL	PAL REPRESENTATIVE LOGIC	2
K024D01.PAL	PAL REPRESENTATIVE LOGIC	1
K025D01.PAL	PAL REPRESENTATIVE LOGIC	1
K026D01.PAL	PAL REPRESENTATIVE LOGIC	1
D027DEF.SCH	TOP LEVEL FOR PROTOCOL FIGURES	1
K044D00.BLK	HCB PROTO, APPEN 1, FIG 1	1
K045D00.BLK	HCB PROTO, APPEN 1, FIG 2	1
K006D01.TBL	HCB PROTO, APPEN 2, FIG 1	1
K006D01A.TBL	HCB PROTO, APPEN 2, FIG 2	1
D029D01.TBL - D029D05.TBL	HCB PROTO, APPEN 3, FIG 1-5	5
BSP-2-WD.TBL - BSP-8-WD.TBL	HCB PROTO, APPEN 4, FIG 1-7	7
K012D01.XXX - K012D12.XXX	BARREL ROLL XILINX DESIGN (XXX = BLK, TIM OR XIL)	12
K013D01.XXX - K013D12.XXX	DATA INVALID XILINX DESIGN (XXX = BLK OR XIL)	12
BSP2BLK.SCH	BSP-2 XILINX DESIGN, TOP LEVEL	1
BSP2X01.SCH - BSP2X11.SCH	REPRESENTATIVE XILINX SCHEMATIC	11
N1-4E.TIM	TIMING FOR N1-4E MODE	1
V1-4E.TIM	TIMING FOR V1-4E MODE	1
N1-4F.TIM	TIMING FOR N1-4F MODE	1
N1-4G.TIM	TIMING FOR N1-4G MODE	1
N1-2E.TIM	TIMING FOR N1-2E MODE	1
N1-2F.TIM	TIMING FOR N1-2F MODE	1
N1-2G.TIM	TIMING FOR N1-2G MODE	1
N1-1F.TIM	TIMING FOR N1-1F MODE	1
BSP3BLK.SCH	BSP-3 XILINX DESIGN, TOP LEVEL	1
BSP3X01.SCH - BSP3X11.SCH	REPRESENTATIVE XILINX SCHEMATIC	11
V1-4F.TIM	TIMING FOR V1-4F MODE	1
V1-2E.TIM	TIMING FOR V1-2E MODE	1
BSP4BLK.SCH	BSP-4 XILINX DESIGN, TOP LEVEL	1
BSP4X01.SCH - BSP4X09.SCH	REPRESENTATIVE XILINX SCHEMATIC	9

V1-2F.TIM	TIMING FOR V1-2F MODE	1
V1-4G.TIM	TIMING FOR V1-4G MODE	1
BSP5BLK.SCH	BSP-5 XILINX DESIGN, TOP LEVEL	1
BSP5X01.SCH - BSP5X11.SCH	REPRESENTATIVE XILINX SCHEMATIC	11
OS2K1-2.TIM	TIMING FOR OVERSAMP 2K 1-2 MODE	1
BSP6BLK.SCH	BSP-6 XILINX DESIGN, TOP LEVEL	1
BSP6X01.SCH - BSP6X12.SCH	REPRESENTATIVE XILINX SCHEMATIC	12
V4-1E.TIM	TIMING FOR V4-1E MODE	1
V2-1E.TIM	TIMING FOR V2-1E MODE	1
V1-1E.TIM	TIMING FOR V1-1E MODE	1
N2-1F.TIM	TIMING FOR N2-1F MODE	1
N4-1F.TIM	TIMING FOR N4-1F MODE	1
BSP7BLK.SCH	BSP-7 XILINX DESIGN, TOP LEVEL	1
BSP7X01.SCH - BSP7X10.SCH	REPRESENTATIVE XILINX SCHEMATIC	10
V1-1F.TIM	TIMING FOR V1-1F MODE	1
V2-1F.TIM	TIMING FOR V2-1F MODE	1
V4-1F.TIM	TIMING FOR V4-1F MODE	1
N1-1G.TIM	TIMING FOR N1-1G MODE	1
N2-1G.TIM	TIMING FOR N2-1G MODE	1
N4-1G.TIM	TIMING FOR N4-1G MODE	1
BSP8BLK.SCH	BSP-8 XILINX DESIGN, TOP LEVEL	1
BSP8X01.SCH - BSP8X11.SCH	REPRESENTATIVE XILINX SCHEMATIC	11
V1-1G.TIM	TIMING FOR V1-1G MODE	1
V2-1G.TIM	TIMING FOR V2-1G MODE	1
V4-1G.TIM	TIMING FOR V4-1G MODE	1
BSP9BLK.SCH	BSP-9 XILINX DESIGN, TOP LEVEL	1
BSP9X01.SCH - BSP9X09.SCH	REPRESENTATIVE XILINX SCHEMATIC	9
L008D01.SRC	ORCAD SRC FOR L008D01 HIERARCY	1
L008D01.LIB	ORCAD LIB	1
B004T01.PRT	DEFORMATTER BILL OF MATERIALS	1
W005T01.NET	DEFORMATTER NETLIST	1
CALAY.NET	NETLIST FROM TRI-CIRCUITS (?)	1
		213 total

I.4.2) 68K source files maintained in vlbsoft/defasm/SCCS

FILE NAMES	DESCRIPTION	NR FILES
Makefile	creates defrom.hex, defram.hex	1
*.asm	see SOURCES list in Makefile	53
*.h and *.mac include files	see INCLUDES list in Makefile	16
d027def.hcb	HCB protocol	1
sRecExtract.c	brks hex file into ROM and RAM	1
		72 total

I.4.3) PAL files maintained in vlbsoft/pal_prom/pals/def/SCCS

FILE NAMES	DESCRIPTION	NR FILES
D009T01.ABL - D009T30.ABL	ABEL source for Def PALs	30
*.jed (e.g. u100.jed)	JEDEC files (see Figure 5-7 for list)	30

I.4.4) Other PAL files in vlbsoft/pal_prom/pals/def/SCCS

These files are mostly for the L037D01.SCH Deformatter Data Generator logic card. This is a "substitute TRC" that was used as a source of input data for testing the Deformatter in PBI Test Fixture #1. No one has used this card in many years. The Orcad schematics for this card (16 bit versions) are in /home/azalea/corrdwgs/def/misc/dwgs.

There are several other PAL files in this directory, as described below.

FILE NAMES	DESCRIPTION	NR
D016T01.ABL - D016T06.ABL	Pals for L037D01.SCH	6
FORM.ABL	This is a "template" for DEF PALs	1
PBIRESET.ABL	Switch debouncing, most likely in PBI Test Fixture #1	1

I.4.5) Xilinx files maintained in vlbsoft/xilinx/defctrl/SCCS

There are two different Xilinx designs (Barrel Roll and Data Invalid) that are combined into one downloadable file (defctrl.mcs)

There is an alternate Data Invalid design for possible Mark III / IV use that is used to create the alternate downloadable file (mk3ctrl.mcs).

FILE NAMES	DESCRIPTION
lbwrol.mac	Barrel Roll Xilinx top level XACT macro (comments in this macro describe all necessary steps)

```
lbwrol-n.mac     Barrel Roll Xilinx netlist XACT macro
lbwrol-c.mac     Barrel Roll Xilinx configuration XACT macro
lbwrolnr.lca     Barrel Roll Xilinx, configured but not routed
                 (created from the three macros above)
lbwrolr.lca      Barrel Roll Xilinx, after place and route by APR
  (generated from apr -a3 -r4 -s20940 -c lbwrol.cst lbwrolnr.lca
lbwrolr.lca)
lbwrol.cst       Barrel Roll Xilinx constraint file used by APR
lbwrolr.rpt      Barrel Roll Xilinx report file from APR
lbwapr.bat       DOS batch file used to invoke APR
lbwrolr.bit      Barrel Roll Xilinx bit file
                 (generated from makebits -nt lbwrolr.lca)

inv.mac          Data Invalid Xilinx top level XACT macro
                 (comments in this macro describe all necessary steps)
inv-r.mac        Data Invalid Xilinx netlist XACT macro
inv-c.mac        Data Invalid Xilinx configuration XACT macro
inv-a.lca        Data Invalid Xilinx, configured but not routed
                 (created from the three macros above)
inv.lca          Data Invalid Xilinx, after place and route by APR
        (generated from apr -a3 -r4 -s2254 -c inv.cst inv-a.lca inv.lca)
inv.cst          Data Invalid Xilinx constraint file used by APR
inv.rpt          Data Invalid Xilinx report file from APR
invapr.bat       DOS batch file used to invoke APR
inv.bit          Data Invalid Xilinx bit file
                 (generated from makebits -nt inv.lca)

defctrl.mcs      The downloadable HEX file for the combined Barrel Roll
                 and Data Invalid Xilinx designs (this is the normal one)
                 (generated from makeprom -u lbwrolr.bit inv.bit, which
                  produces lbwrolr.mcs;  a dos2unix lbwrolr.mcs
defctrl.mcs
                  is the final step)

mk3inv.mac       Data Invalid Xilinx top level XACT macro
                 (comments in this macro describe all necessary steps)
mk3inv-r.mac     Data Invalid Xilinx netlist XACT macro
mk3inv-c.mac     Data Invalid Xilinx configuration XACT macro
mk3inv-a.lca     Data Invalid Xilinx, configured but not routed
                 (created from the three macros above)
mk3inv.lca       Data Invalid Xilinx, after place and route by APR
                 (generated from apr -c mk3inv.cst -s17987 mk3inv-a.lca
mk3inv)
mk3inv.cst       Data Invalid Xilinx constraint file used by APR
mk3inv.rpt       Data Invalid Xilinx report file from APR
mk3inv.bit       Data Invalid Xilinx bit file
                 (generated from makebits -nt inv.lca)

mk3ctrl.mcs      The alternate downloadable HEX file for the combined
                 Barrel Roll and alternate Data Invalid designs
                 generated from makeprom -u 0 lbwrolr.bit mk3inv.bat,
                 which produces lbwrolr.mcs;
                 a dos2unix lbwrolr.mcs mk3ctrl.mcs is the final step
```

I.4.6) Xilinx files maintained in vlbsoft/xilinx/bsp0/SCCS

FILE NAMES	DESCRIPTION
bsp-0.sch	Orcad / Xilinx schematic; processed by XDM (Xilinx Design Manager) to produce bsp-0.lca
bsp-0.lca	Routed and configured design, from XDM
bsp0prom.mac	XACT macro, operates on bsp-0.lca to produce bsp-0.bit and bsp-0.mcs
bsp-0.bit	Bit file from previous macro
bsp-0.mcs	HEX file from previous macro AND a dos2unix conversion

I.4.7) Xilinx files maintained in vlbsoft/xilinx/bsp1/SCCS

FILE NAMES	DESCRIPTION
bsp-1.sch	Orcad / Xilinx schematic; processed by XDM (Xilinx Design Manager) to produce bsp-1.lca
bsp-1.lca	Routed and configured design, from XDM
bsp1prom.mac	XACT macro, operates on bsp-1.lca to produce bsp-1.bit and bsp-1.mcs
bsp-1.bit	Bit file from previous macro
bsp-1.mcs	HEX file from previous macro AND a dos2unix conversion
xc3000.lib	Xilinx Orcad library used by both bsp-0 and bsp-1

I.4.8) Xilinx files in vlbsoft/xilinx/bsp2-bsp6 and bsp9/SCCS

Bit Shuffler Xilinx designs for bsp-2 through bsp-6 and bsp-9 are all similar. They do not use APR. The top level XACT macro for each design invokes all steps required to generate the .mcs HEX file. A final dos2unix conversion is required on the .mcs file in each case.

The FILE NAMES listed below are for the bsp2 directory. The other directories, are identical except for the number (3 through 6, and 9).

The top level macro contains descriptive comments.

FILE NAMES	DESCRIPTION
bsp-2.mac	XACT top level macro
bsp-2-r.mac	XACT routing macro; invoked from top level
bsp-2-c.mac	XACT configuration macro; invoked from top level
bsp-2.lca	Routed and configured design, produced by the macros
bsp-2.bit	Bit file produced by the top level macro
bsp-2.mcs	HEX file produced by the top level macro (a final dos2unix conversion is required)
bsp-2drc.rpt	Design Rule Checker report file, from top level macro

I.4.9) Xilinx files in vlbsoft/xilinx/bsp7 and bsp8/SCCS

Bit Shuffler Xilinx designs for bsp-7 and bsp-8 are both similiar. They do use APR. The FILE NAMES listed below are for the bsp7 directory. Replace the number 7 with 8 for the bsp8 directory.

The top level macro contains descriptive comments.

FILE NAMES	DESCRIPTION
bsp-7.mac	XACT top level macro
bsp-7-r.mac	XACT netlist macro, invoked by the top level
bsp-7-c.mac	XACT configuration macro, invoked by the top level
bsp-7-a.lca	Configured, un-routed design produced by the macros
bsp-7.lca	Placed and routed design, produced by APR (see top level macro for the APR command line)
bsp-7.cst	Constraint file used by APR
bsp7prom.mac	XACT macro, produces the .bit and .mcs files
bsp-7.bit	Bit file
bsp-7.mcs	HEX file; dos2unix conversion required as final step
bsp-7drc.rpt	Design Rule Checker report file
bsp-7.rpt	APR report file

I.4.10) Files in vlbsoft/defseq/SCCS

There are too many files here to list. This directory contains the source files (.dat) for the Deformatter sequencers. A Makefile controls the generation of the .hex files that are downloaded into the Deformatter. See Section 5.4.0 for a description of file names.

The file genbits.c is the processor that reads the .dat files and writes the .hex files.

The file tables.txt provides some details on the bit definitions in the sequencers.

This page intentionally left blank.

APPENDIX II PBI Mode Checkout Examples

II.1) Example C source program for Mode Checkout Predicteds

```
#include <stdio.h>
#include <math.h>


/*********************************************************************/

struct Rnd
{
  unsigned int           :16;
  unsigned int bit15     : 1;
  unsigned int bit14     : 1;
  unsigned int           :12;
  unsigned int bit1      : 1;
  unsigned int bit0      : 1;
};

union rnd
{
  struct Rnd             bits;
  unsigned int           ul;
};

union rnd val;

unsigned char minix_buf[65536];  /* bits 0-7 = eight data streams 0-7 */
/* minix has 4096-16 actual states; in 4-1 mode, we run beyond the last
*/
/* of these states, so made this buf extra large in order to not worry
about */
/* exceeding it; am depending on it being initialized by compiler to zero
*/

char bit_trk_strings[5][10]= {"1-4", "1-2", "1-1", "2-1", "4-1"};

/*********************************************************************/
/*
  We need to predict the FFT card data input stream as a function of
  PBI mode, when the Mini-transport is used as the data source.
  Barrel "un-rolling" will not be turned on in the PBI.  The mode
  switch on the Mini-X has no effect on the 8 data streams generated
  by the Mini-X for this test, at least as far as selecting 1-4, 1-2
  etc.

  The Mini-transport puts out 8 data streams, D0-D7.  These eight data
  streams are cabled to the PBI TRC input tracks as shown by the first
  two columns below, and the track-mix is used to copy the data
```

streams to the other three sets of eight tracks as indicated by the last three columns below. All "track" numbers are what Haystack called "logical" tracks and the PBI group calls "formatter" tracks.

Mini-X Data Streams	PBI Tracks	PBI Tracks	PBI Tracks	PBI Tracks
D0	0	8	16	24
D1	1	9	17	25
D2	2	10	18	26
D3	3	11	19	27
D4	4	12	20	28
D5	5	13	21	29
D6	6	14	22	30
D7	7	15	23	31

The eight data streams are generated from a 15 bit pseudo-random data generator as follows:

15 bit shift register with exclusive OR gate to form random number generator

"hardware" bits 14 and 15 provide the feedback

(zero is a hang up state, so seed must not be zero)

"Hardware" bits 1-8 become D0-D7 respectively

unsigned short int number format:

```
            15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
```

"hardware" shift register form:

```
Default seed: (In Mini-X     is 0xABCD, where MSB is off end)
              (In pbisrand() is 0x579A, where LSB is off end)
MINI-X:               |           |           |           |
          1  0  1  0  1  0  1  1  1  1  0  0  1  1  0  1  0
pbisrand():             |           |           |           |
            15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
             |  |                                          ^
    --------  |  |                                          |
    | EX   |<-   |                                          |
 ---| OR   |     |                                          |
 |  | gate |<----                                          |
 |  --------                                               |
 |_________________________________________________________|

         <------------- shift direction
```

The steps for running the MINI-X are:

Hit both RESET and INIT buttons several times after power on.

Talking to the CRT ports on the small "SON OF BARF" card and on the large "MINI-X" card:

```
       barf> t2          (sets rep rate for delay cycle)
       barf> f           (fills 4K ram with pseudo, seed= ABCD)
       mini> s           (to stop the sequence)
       barf> t1          (to prime things)
       mini> s           (to start the sequence)
                         (must hit start obs on Real Time System at same
time)


*/
/*************************************************************************/
unsigned char pbirand()
{
  unsigned int bit1_new_val;

 /* After shifting one place left, bit 1 must take on the value of the */
 /* exclusive or of bits 14 and 15 from before the shift, thus we do */
 /* the exclusive or of the current bits 14 and 15, then copy the */
 /* result to bit 0 so that the shift will cause that value to shift */
 /* into the bit 1 position */

 /* the byte returned represents the 8 data streams out of the Mini-X */
 /* where bit zero is data stream 0; bit 7 is data stream 7           */

  bit1_new_val = val.bits.bit14 ^ val.bits.bit15;
  val.bits.bit0 = bit1_new_val;

  val.ul <<= 1;

  return (val.ul >> 1);
}

/*************************************************************************/
pbisrand(seed)
unsigned int seed;
{
  if (  (seed&0xfffe)    ==0)
    {
      fprintf(stderr,"\b invalid seed= 0\n");
      fprintf(stderr,"Using 0x579a as default\n");
      val.ul = 0x579a;
      return;
    }
  val.ul = seed;
}

/*************************************************************************/


main(argc,argv)
int argc;
char *argv[];
{
  int seed;
  int state;
  int test;
```

```
  char response[20];
  int bt;

  fprintf(stderr,"Select mode (1-4, 1-2 etc.) for which to generate
predicteds\n");
  fprintf(stderr,"(oversample factor = 4 for these tests)\n");
  fprintf(stderr,"For 1-4 mode enter 1-4, for 1-2 mode enter 1-2
etc.\n");
  fprintf(stderr,"Mode:  ");
  gets(response);
  fprintf(stderr,"\n");
  fprintf(stderr,"Mode %s selected.\n",response);

  for(bt=0; bt<5; bt++)
    {
      if ( (strcmp(response,bit_trk_strings[bt])==0) ) break;
    }
  if (bt==5)
    {
      fprintf(stderr,"Invalid bit to track mode specified=
%s\n",response);
      exit();
    }


  if (argc > 1)
    {
      seed= atoi(*++argv);
    }
  else
    {
      seed= 0x579a;
    }

  fprintf(stderr,"seed= %04.4X\n",seed);

  pbisrand(seed);

  minix_buf[0]= val.ul >> 1;
  for(state=1; state<4080; state++)  /* 4096 minus 16 = 4080 states total
*/
    {                                /* since minix starts at adr hex 10
*/
      minix_buf[state]= pbirand();
    }


  switch(bt)
    {
    case 0: do_1to4_128_ovs8(seed); break;
    case 1: do_1to2_128_ovs8(seed); break;
    case 2: do_1to1_128_ovs8(seed); break;
    }

}

/*****************************************. ******************/
```

```
do_1to4_128_ovs8(seed)
int seed;
{
  /*
    See set-up sheets in notebook for bit sequences


  */

  unsigned char sample0, sample1, sample2, sample3;
  unsigned char sign0, sign1, sign2, sign3;
  unsigned char mag0, mag1, mag2, mag3;
  int minix_state;
  int fft_state;
  FILE *outfile;
  int fftnr;
  int fftstart, fftend;
  int arium_index;
  int index;

/*  static int start1to4[8]= {0,0,32,32,64,64,96,96}; */
  static int start1to4[8]= {64,64,96,96,512,512,544,544};

  outfile= fopen("1to4-128.ovs8.mode.prd","w");
  if (outfile==NULL)
    {
      fprintf(stderr,"Can't open 1to4-128.ovs8.mode.prd for writing.\n");
      return;
    }

  fprintf(outfile,"*PBI 1-4,128,ovs8  mode, predicteds, seed= %04.4X
\n",seed);
  fprintf(outfile,"Arium  FFT  Index   Ch0 Ch1 Ch2 Ch3\n");
  fprintf(outfile,"----- ----- -----   --- --- --- ---\n");

  arium_index= -1;
  index=0;

  for(fftnr=0; fftnr<8; fftnr++)
    {
      fftstart= start1to4[fftnr];
      minix_state= fftstart*2;

      fftend= fftstart+516;
      for(fft_state=fftstart; fft_state<fftend; )
        {
          if ((fftnr%2)==0)
            {
              sign0= (minix_buf[minix_state+0] & 0x01) >> 0;
              mag0 = (minix_buf[minix_state+0] & 0x10) >> 4;

              sign1= sign0;
              mag1 = mag0;

              sign2= sign0;
              mag2 = mag0;
```

```
            sign3= sign1;
            mag3 = mag1;
          }
        else
          {
            sign0= (minix_buf[minix_state+1] & 0x01) >> 0;
            mag0 = (minix_buf[minix_state+1] & 0x10) >> 4;

            sign1= sign0;
            mag1 = mag0;

            sign2= sign0;
            mag2 = mag0;

            sign3= sign1;
            mag3 = mag1;

          }

        sample0 =  (sign0 & 1) << 1;
        sample0 |= (mag0  & 1);

        sample1 =  (sign1 & 1) << 1;
        sample1 |= (mag1  & 1);

        sample2 =  (sign2 & 1) << 1;
        sample2 |= (mag2  & 1);

        sample3 =  (sign3 & 1) << 1;
        sample3 |= (mag3  & 1);

        fprintf(outfile,"%5d %4d %6d     %01.1X   %01.1X   %01.1X
%01.1X",
                arium_index,fft_state,index,sample0, sample1, sample2,
sample3);

        if (fft_state == (fftend-1))
          {
            fprintf(outfile,"    End of FFT # %d",fftnr);
          }

        fprintf(outfile,"\n");

        fft_state++;
        arium_index++;
        index++;
        minix_state+=2;
      }

  }
 fclose(outfile);
 fprintf(stderr,"Wrote predicteds for 128, 1to4 ovs8 mode to file: 1to4-
128.ovs8.mode.prd\n");

}
```

```
/*****************************************************************/
do_1to2_128_ovs8(seed)
int seed;
{
  /*
    See set-up sheets in notebook for bit sequences


  */

  unsigned char sample0, sample1, sample2, sample3;
  unsigned char sign0, sign1, sign2, sign3;
  unsigned char mag0, mag1, mag2, mag3;
  int minix_state;
  int fft_state;
  FILE *outfile;
  int fftnr;
  int fftstart, fftend;
  int arium_index;
  int index;

/*  static int start1to2[8]= {0,0,16,16,32,32,48,48}; */
  static int start1to2[8]= {96,96,112,112,512,512,528,528};


  outfile= fopen("1to2-128.ovs8.mode.prd","w");
  if (outfile==NULL)
    {
      fprintf(stderr,"Can't open 1to2-128.ovs8.mode.prd for writing.\n");
      return;
    }

  fprintf(outfile,"*PBI 1-2,128,ovs8  mode, predicteds, seed= %04.4X
\n",seed);
  fprintf(outfile,"Arium  FFT  Index   Ch0 Ch1 Ch2 Ch3\n");
  fprintf(outfile,"----- ----- -----   --- --- --- ---\n");

  arium_index= -1;
  index=0;

  for(fftnr=0; fftnr<8; fftnr++)
    {
      fftstart= start1to2[fftnr];
      minix_state= fftstart*4;

      fftend= fftstart+516;
      for(fft_state=fftstart; fft_state<fftend; )
        {
          if ((fftnr%2)==0)
            {
              sign0= (minix_buf[minix_state+0] & 0x01) >> 0;
              mag0 = (minix_buf[minix_state+0] & 0x04) >> 2;

              sign1= (minix_buf[minix_state+0] & 0x10) >> 4;
              mag1 = (minix_buf[minix_state+0] & 0x40) >> 6;

              sign2= sign0;
```

```
            mag2 = mag0;

            sign3= sign1;
            mag3 = mag1;
          }
        else
          {
            sign0= (minix_buf[minix_state+2] & 0x01) >> 0;
            mag0 = (minix_buf[minix_state+2] & 0x04) >> 2;

            sign1= (minix_buf[minix_state+2] & 0x10) >> 4;
            mag1 = (minix_buf[minix_state+2] & 0x40) >> 6;

            sign2= sign0;
            mag2 = mag0;

            sign3= sign1;
            mag3 = mag1;

          }

        sample0 =  (sign0 & 1) << 1;
        sample0 |= (mag0  & 1);

        sample1 =  (sign1 & 1) << 1;
        sample1 |= (mag1  & 1);

        sample2 =  (sign2 & 1) << 1;
        sample2 |= (mag2  & 1);

        sample3 =  (sign3 & 1) << 1;
        sample3 |= (mag3  & 1);

        fprintf(outfile,"%5d %4d %6d     %01.1X   %01.1X   %01.1X
%01.1X",
                arium_index,fft_state,index,sample0, sample1, sample2,
sample3);

        if (fft_state == (fftend-1))
          {
            fprintf(outfile,"    End of FFT # %d",fftnr);
          }

        fprintf(outfile,"\n");

        fft_state++;
        arium_index++;
        index++;
        minix_state+=4;
      }

  }
 fclose(outfile);
 fprintf(stderr,"Wrote predicteds for 128, 1to2 ovs8 mode to file: 1to2-
128.ovs8.mode.prd\n");

}
```

```
/*****************************************************************/
do_1to1_128_ovs8(seed)
int seed;
{
  /*
    See set-up sheets in notebook for bit sequences


  */

  unsigned char sample0, sample1, sample2, sample3;
  unsigned char sign0, sign1, sign2, sign3;
  unsigned char mag0, mag1, mag2, mag3;
  int minix_state;
  int fft_state;
  FILE *outfile;
  int fftnr;
  int fftstart, fftend;
  int arium_index;
  int index;

/*  static int start1to1[8]= {0,0,8,8,16,16,24,24}; */
  static int start1to1[8]= {112,112,120,120,512,512,520,520};


  outfile= fopen("1to1-128.ovs8.mode.prd","w");
  if (outfile==NULL)
    {
      fprintf(stderr,"Can't open 1to1-128.ovs8.mode.prd for writing.\n");
      return;
    }

  fprintf(outfile,"*PBI 1-1,128,ovs8  mode, predicteds, seed= %04.4X
\n",seed);
  fprintf(outfile,"Arium  FFT  Index   Ch0 Ch1 Ch2 Ch3\n");
  fprintf(outfile,"----- ----- -----   --- --- --- ---\n");

  arium_index= -1;
  index=0;

  for(fftnr=0; fftnr<8; fftnr++)
    {
      fftstart= start1to1[fftnr];
      minix_state= fftstart*8;

      fftend= fftstart+516;
      for(fft_state=fftstart; fft_state<fftend; )
        {
          if ((fftnr%2)==0)
            {
              sign0= (minix_buf[minix_state+0] & 0x01) >> 0;
              mag0 = (minix_buf[minix_state+0] & 0x02) >> 1;

              sign1= (minix_buf[minix_state+0] & 0x04) >> 2;
              mag1 = (minix_buf[minix_state+0] & 0x08) >> 3;
```

```
            sign2= (minix_buf[minix_state+0] & 0x10) >> 4;
            mag2 = (minix_buf[minix_state+0] & 0x20) >> 5;

            sign3= (minix_buf[minix_state+0] & 0x40) >> 6;
            mag3 = (minix_buf[minix_state+0] & 0x80) >> 7;
          }
        else
          {
            sign0= (minix_buf[minix_state+4] & 0x01) >> 0;
            mag0 = (minix_buf[minix_state+4] & 0x02) >> 1;

            sign1= (minix_buf[minix_state+4] & 0x04) >> 2;
            mag1 = (minix_buf[minix_state+4] & 0x08) >> 3;

            sign2= (minix_buf[minix_state+4] & 0x10) >> 4;
            mag2 = (minix_buf[minix_state+4] & 0x20) >> 5;

            sign3= (minix_buf[minix_state+4] & 0x40) >> 6;
            mag3 = (minix_buf[minix_state+4] & 0x80) >> 7;

          }

        sample0 =  (sign0 & 1) << 1;
        sample0 |= (mag0  & 1);

        sample1 =  (sign1 & 1) << 1;
        sample1 |= (mag1  & 1);

        sample2 =  (sign2 & 1) << 1;
        sample2 |= (mag2  & 1);

        sample3 =  (sign3 & 1) << 1;
        sample3 |= (mag3  & 1);

        fprintf(outfile,"%5d %4d %6d     %01.1X   %01.1X   %01.1X
%01.1X",
                arium_index,fft_state,index,sample0, sample1, sample2,
sample3);

        if (fft_state == (fftend-1))
          {
            fprintf(outfile,"    End of FFT # %d",fftnr);
          }

        fprintf(outfile,"\n");

        fft_state++;
        arium_index++;
        index++;
        minix_state+=8;
      }
    }
  fclose(outfile);
  fprintf(stderr,"Wrote predicteds for 128, 1to1 ovs8 mode to file: 1to1-
128.ovs8.mode.prd\n");
}
```

II.2) Example Mode Checkout Script

```
#
/home/azalea1/vlb/cbroadwe/mdchk/128.ovs8/md.scripts/do1to4.128.ovs8.cmd

#  Logic analyzer setup:
#    CH  0           CH   1           CH  2           CH   3
#  ------------    --------------    ------------    --------------
#  SIGN   MAG      SIGN    MAG       SIGN   MAG      SIGN    MAG
#  -----  -----    ------  ------    -----  -----    ------  ------
#  U48-5  U48-7    U48-10  U48-11    U26-5  U26-7    U26-10  U26-11
#  BIT 1  BIT 0    BIT  3  BIT  2    BIT 5  BIT 4    BIT  7  BIT  6

#  FFTINIT = U191 pin 2    C32 clock = U191 pin 3
#  NEW_FRINGE (4 msec) = Test Point 19 (at top of Deformatter)

#  Logic Anal Bit   PIN         Logic Anal Bit   PIN
#  --------------  -----        --------------  ------
#       0          U48-7            10          U191-2
#       1          U48-5            11          TP 19
#       2          U48-11
#       3          U48-10
#       4          U26-7
#       5          U26-5
#       6          U26-11
#       7          U26-10

#  Trigger on 1 of NEW_FRINGE low, then 6 of don't care to get
#  FFT state 0 at index -1

# barf:  t2    f          t0
# minix:          stop        start

pbibus = 0;
def    = 1;

pbi= 0;    /* test fixture is bus 0, target is 1 */
fftsiz=128;
bits = 2;
ovs= 8;
spd = 1;
pol = 0;       /* not polar */
demod = 0;

# A comma separated string of ints is passed to provide the mapping from
# recorder tracks to FFT channels;  For purposes of mode checkouts, we
# need the low odds routed to each delay center;
# The order is FFT chan 0 thru 7;

hcbRstTarget pbibus,def;
taskDelay(180);
```

```
ldPbi(pbi_index, fftsiz);

p = pbi;    /* keep next line under 80 characters */
mode_setpbi(p,fftsiz,"1-
4","rolloff",bits,"3,3,3,3,3,3,3,3",ovs,spd,pol,demod);

/* setpbi() does not send time reference */
hcbDefTimeVlba (0, 0, 0, 0, 0, 0, 0, 0, pbibus, def);

# pipeline delay should be:  0xfff6

/* setpbi() does not send servo table */
defWrServoTable (pbi_index, 160.0);

# hcbCode pbibus,def,3,0x50,1,1    /* simultaneous start with minix start
*/
```

II.3) Example Mode Checkout Results

PREDICTEDS							ACTUAL RESULTS							
*PBI 1-4,128,ovs8 mode, predicteds							Copyright 1991, American Automation							
Arium	FFT	Index	Ch0	Ch1	Ch2	Ch3								
							STATE	CH0	CH1	CH2	CH3	INI	4MS	NCO
-1	0	0	2	2	2	2	-00001	2	2	2	2	1	1	0
0	1	1	3	3	3	3	TRIG	3	3	3	3	1	1	0
1	2	2	3	3	3	3	+00001	3	3	3	3	1	1	0
2	3	3	1	1	1	1	+00002	1	1	1	1	1	1	0
3	4	4	1	1	1	1	+00003	1	1	1	1	1	1	0
4	5	5	0	0	0	0	+00004	0	0	0	0	1	1	0
5	6	6	0	0	0	0	+00005	0	0	0	0	1	1	0
6	7	7	2	2	2	2	+00006	2	2	2	2	1	1	0
7	8	8	0	0	0	0	+00007	0	0	0	0	1	1	0
8	9	9	1	1	1	1	+00008	1	1	1	1	1	1	0
9	10	10	2	2	2	2	+00009	2	2	2	2	1	1	0
10	11	11	0	0	0	0	+00010	0	0	0	0	1	1	0
11	12	12	3	3	3	3	+00011	3	3	3	3	1	1	0
12	13	13	2	2	2	2	+00012	2	2	2	2	1	1	0
13	14	14	1	1	1	1	+00013	1	1	1	1	1	1	0
14	15	15	1	1	1	1	+00014	1	1	1	1	1	1	0
15	16	16	0	0	0	0	+00015	0	0	0	0	1	1	0
16	17	17	2	2	2	2	+00016	2	2	2	2	1	1	0
17	18	18	0	0	0	0	+00017	0	0	0	0	1	1	0
18	19	19	1	1	1	1	+00018	1	1	1	1	1	1	0
19	20	20	0	0	0	0	+00019	0	0	0	0	1	1	0
20	21	21	2	2	2	2	+00020	2	2	2	2	1	1	0
21	22	22	2	2	2	2	+00021	2	2	2	2	1	1	0
22	23	23	1	1	1	1	+00022	1	1	1	1	1	1	0
23	24	24	3	3	3	3	+00023	3	3	3	3	1	1	0
24	25	25	2	2	2	2	+00024	2	2	2	2	1	1	0
25	26	26	3	3	3	3	+00025	3	3	3	3	1	1	0
26	27	27	1	1	1	1	+00026	1	1	1	1	1	1	0
27	28	28	3	3	3	3	+00027	3	3	3	3	1	1	0
28	29	29	0	0	0	0	+00028	0	0	0	0	1	1	0
29	30	30	1	1	1	1	+00029	1	1	1	1	1	1	0
30	31	31	2	2	2	2	+00030	2	2	2	2	1	1	0
31	32	32	2	2	2	2	+00031	2	2	2	2	1	1	0
32	33	33	1	1	1	1	+00032	1	1	1	1	1	1	0
33	34	34	1	1	1	1	+00033	1	1	1	1	1	1	0

This page intentionally left blank.

APPENDIX III Track Recovery Card HCB Protocol

This page intentionally left blank.

TRC PROTOCOL (DEFORMATTER CARD TO TRC COMMUNICATION) NRAO DRAWING #A56000D028
TEXT FILE D028TRC.HCB

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA 22903

Revision History	Date	By
Preliminary	07/09/92	RPE
	06/22/93	RPE

XFER TYPE	PROTOCOL	FREQ	SOURCE
NOP	00	-	-
WRITE MEMORY	*01 00 00 AA AA CC CC DD----DD	INIT	RT (1)
READ MEMORY	02 00 00 AA AA CC CC	TEST	-
CALCULATE CHECKSUM	03 00 00 AA AA CC CC	INIT	RT
YOU ARE MICRO A	*04	INIT	DEF
YOU ARE MICRO B	*05	INIT	DEF
TEST OBSERVATION	*06	TEST	-
XILINX 1ST XFER	*07 DD----DD (1910 BYTES)	INIT	RT (2)
XILINX 2ND XFER	*DD----DD (1910 BYTES)	INIT	RT (2)
...	...	...	...
XILINX 15TH XFER	*DD----DD (1910 BYTES)	INIT	RT (2)
XILINX GO/NO GO FLAG	08 [1 BYTE RETURNED]	INIT	RT
VLBA TAPE FORMAT	*09	TEST	-
MKIII TAPE FORMAT	*0A	TEST	-
TEST ECHO	0B XX [1 BYTE RETURNED]	TEST	-
CONNECT TO TEST FRAME	*0C	TEST	-
CONNECT TO TRANSPORT	*0D	TEST	-
TRACK MIX	*10 0X YY ZZ	OBS	RT
TRACK ASSIGNMENT	*11 0S AB CD EF GH	OBS	RT
OBSERVATION PARAMETERS	*12 P1 P2 P3 ... P8	OBS	RT
FRAME DURATION	*13 T1 T2	OBS	RT
GO TO OBSERVING MODE	*14	OBS	RT
DATA INVALID MODE	*15 (3)	(4)	RT
AUTO DATA VALID MODE	*16 (3)	(4)	RT
END OBSERVATION	*17 (3)	END OBS	RT
HEADER SNAPSHOT	*18	INTEG CYC	DEF
HEADER READOUT	19 0T [GET 24 BYTES IN RETURN]	INTEG CYC	DEF
TIME REQUEST	1A [GET 8 BYTES IN RETURN]	DELAY CYC	DEF
AUTO TEST	*1B PT	TEST	-
DEMOD ON	*1C	TEST	-
DEMOD OFF	*1D	TEST	-
SYSTEM STATUS	1E [GET 10 BYTES IN RETURN]	INTEG CYC	DEF
CRT PROMPT	1F AA----AA 00----00 (16 BYTES)	INIT	RT
DELAY	8V DD DD	DELAY CYC	DEF

FUNCTION DURATIONS
The table below shows the name for the function code in the include file hcbFuncCodes.h. Also shown are the times in milliseconds from a survey of how long the functions take. Approximate data rates in KBytes/second are shown. There are potential problems for functions that have an equivalent transfer rate of less than 50 KBytes/sec, and take more than 10 ms. For a detailed explanation of the measurements taken see /home/azalea1/staff/jgreenbe/notes/survey.txt. Functions labled INIT below are only used during initialization, where timing is not critical. The timing of functions labeled TEST is not of concern when observing. Data rates are not give where minimal data is transferred.

FUNC NAME	FUNC CODE	FREQ	TIME (ms)	FUNC NAME	FUNC CODE	FREQ	TIME (ms)
YOUAREA	0x04	INIT		TRCOBSPARAMS	0x12	OBS	0.338
YOUAREB	0x05	INIT		TRCFRAMEDUR	0x13	OBS	0.265
LDTRCXIL	0x07	INIT			0x14	OBS	0.715
TRCXILFLG	0x08	INIT		TRCDINVAL	0x15	(3)(4)	0.219
TRCVLBAFMT	0x09	TEST		TRCAUTODINV	0x16	(3)(4)	0.221
TRCMKIIIFMT	0x0a	TEST			0x17	(3)	0.221
PBITESTSRC	0x0C	TEST		AUTOTEST	0x1B	TEST	
PBITAPESRC	0x0D	TEST		DEMODON	0x1c	TEST	
TRCTRKMIX	0x10	OBS	0.285	DEMODOFF	0x1d	TEST	
TRCTRKASGN	0x11	OBS	0.305	TRCID	0x1f	INIT	

* can be multi-processor or individual transfer

(1) transfer verified by checksum calculated on stored bytes
(2) transfer verified by go/no go flag
(3) normally the RS system should not communicate with a TRC when
it is in OBSERVING mode. The only exception to this rule is these
three commands.
(4) as needed during observation

---

1) MAIN MEMORY

NOP:	00
WRITE MEMORY:	01 00 00 AA AA CC CC DD----DD
READ MEMORY:	02 00 00 AA AA CC CC
CALCULATE CHECKSUM:	03 00 00 AA AA CC CC

where,

AA AA is a 16-bit start address and CC CC is a transfer count with MS byte first.

2) MICRO IDENTIFY (Deformatter responsibility)

YOU ARE MICRO A:	04
YOU ARE MICRO B:	05

There are two microprocessors on each Track Recovery Card. Each micro has identical software but slightly different responsibilities (for example, micro A runs the header sequencer and micro B loads the Xilinx personality). Hence the Deformatter Card passes each microprocessor a byte just after reset that tells each microprocessor it's identity.

3) PERFORM TEST "OBSERVATION" (used in conjunction with the PBI test fixture)

TEST OBSERVATION:	06

4) XILINX PERSONALITY (micro B only) (28,650 byte to transfer)
(The TRC romtrb.asm file has ENDCNT EQU 6FEAH which is 28,650 bytes)
(The Xilinx downloader C function does 15 transfers of 1910 bytes each.)

XILINX PROGRAM:	07 DD----DD (1ST XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (2ND XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (3RD XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (4TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (5TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (6TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (7TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (8TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (9TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (10TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (11TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (12TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (13TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (14TH XFER = 1910 BYTES)
XILINX PROGRAM:	DD----DD (15TH XFER = 1910 BYTES)
GO/NO GO FLAG:	08 [1 BYTE RETURNED]

the format of the go/no go flag byte is seen below;

BYTE	FUNCTION
00	both loads unsuccessful
01	only Xilinx A load successful
02	only Xilinx B load successful
03	both loads successful

5) TAPE FORMAT (function can also done by observational parameters command)

VLBA TAPE FORMAT:	09
MKIII TAPE FORMAT:	0A

6) TEST ECHO

TEST ECHO:	0B XX [XX BYTE IS RETURNED]

7) INPUT SOURCE (only Micro A responds and controls the source selection for both micros on the card, so command only need be sent to Delay Centers 0 and 2;)

CONNECT TO TEST FRAME:	0C
CONNECT TO TRANSPORT:	0D

8) TRACK MIX PROGRAM

TRACK MIX: 10 0X YY ZZ

where,

The two LS bits of X program which system track is assigned to the micro being addressed. YY and ZZ together define 8 2-bit fields each of which selects an input PBD track for a Track Recovery Card DPC chip. See Appendix I for more detail.

9) TRACK ASSIGNMENT (assigns track validities to channel validities)
(enables/disables delay center inputs)

```
TRACK ASSIGNMENT:       11 SS AB CD EF GH
                                76 54 32 10 <-- Delay Center
                                                Input Number
```

where,

each delay center input has one nibble (4-bits) to define it's assignment (only three of these bits are used). In bit format, AB = 0aaa 0bbb, CD = 0ccc 0ddd, etc.
The Delay Center Input Number shows which nibble is used to assign which Delay Center Input.
The SS byte controls the use of the system track

For the A, B, C, D, E, F, G and H nibbles, the defined values are listed below:

NIBBLE	FUNCTION
0000	TRACK INACTIVE
0001	TRACK ACTIVE, VALIDITY ASSIGNED TO EVEN CHANNEL
0010	TRACK ACTIVE, VALIDITY ASSIGNED TO ODD CHANNEL
0011	TRACK ACTIVE, VALIDITY ASSIGNED TO BOTH CHANNELS
0100	TRACK ACTIVE, VALIDITY ASSIGNED TO NEITHER CHANNEL

For the SS byte, the defined values are listed below:

SS BYTE	FUNCTION
00	No system track substitution, system track disabled
01	System track substitutes for delay center input 0
02	System track substitutes for delay center input 1
04	System track substitutes for delay center input 2
08	System track substitutes for delay center input 3
10	System track substitutes for delay center input 4
20	System track substitutes for delay center input 5
40	System track substitutes for delay center input 6
80	System track substitutes for delay center input 7
FF	No substitition, but system track is enabled (this is for test purposes only)

When a system track is substituted for a broken track, the broken track must receive normal track assignment bits (A, B, C, etc.) as if it were not broken.

When a system track is used to substitute for a delay center input, the actual PBD track being replaced depends on the Track Mix programming. The following table shows the four possible PBD tracks that may appear at each delay center input. (As shown, these are the same in all four delay centers.) The table also shows the four possible system tracks that can appear at the system track input to each delay center. The order of these is not the same in each delay center.

INPUT	TRC 0, MICRO A	TRC 0, MICRO B	TRC 1, MICRO A	TRC 1, MICRO B
s	00,34,01, or 35	00,34,01, or 35	01,35,00, or 34	01,35,00, or 34
7	17,33,16, or 32	17,33,16, or 32	17,33,16, or 32	17,33,16, or 32
6	15,31,14, or 30	15,31,14, or 30	15,31,14, or 30	15,31,14, or 30
5	13,29,12, or 28	13,29,12, or 28	13,29,12, or 28	13,29,12, or 28
4	11,27,10, or 26	11,27,10, or 26	11,27,10, or 26	11,27,10, or 26
3	09,25,08, or 24	09,25,08, or 24	09,25,08, or 24	09,25,08, or 24

```
:2     : 07,23,06, or 22 : 07,23,06, or 22 : 07,23,06, or 22 : 07,23,06, or 22 :
:1     : 05,21,04, or 20 : 05,21,04, or 20 : 05,21,04, or 20 : 05,21,04, or 20 :
:0     : 03,19,02, or 18 : 03,19,02, or 18 : 03,19,02, or 18 : 03,19,02, or 18 :
-------------------------------------------------------------------------------
```

10) OBSERVATION PARAMETERS (see Appendix II for more detail)

OBSERVATION PARAMETERS: 12 P1 P2 P3 ... P8

where (for P1 = 000p wdsm),

p
- p = 0, not in sync probation mode, if no sync is detected in a frame, data for the preceding frame is flagged invalid.
- p = 1, in sync probation mode, if no sync is detected in a frame, judgement of the data validity for the preceding frame is reserved until one more frame has been received at which time the data validity for both frames is set according to whether a sync slip is detected in the second frame.

w
- w = 0, sync detect watch dog timer not used
- w = 1, sync detect watch dog timer is used

d
- d = 0, demod off
- d = 1, demod active

s
- s = 0, look for frame sync always (wide band sync detect mode)
- s = 1, look for frame sync during header only (narrow band sync det mode)

m
- m = 0, for MKIII
- m = 1, for VLBA

P2 is the number of parity errors at which and above we tag the frame invalid (max is FB, >FB = infinity and 0 = infinity)

P3 is the number of consecutive frames in which error free headers must be seen before the filtered time can be tagged reliable (see function code 1A).

P4 not defined at this time

P5 is the number of consecutive frames in which no good time comparisons occur before filtered time must be reset.

P6 barrel roll parameter,
- 00 = no roll
- x1 = roll by 1 per frame
- x2 = roll by 2 per frame
- x3 = roll by 4 per frame
- 0y = roll within group of 8
- 8y = roll within group of 16

P7 not defined at this time

P8 not defined at this time

11) FRAME DURATION

FRAME DURATION: 13 T1 T2

where,

T1 T2 is the time duration of one frame in BCD format. the TRC micro will add this number to the BCD time it obtains from the track to predict the time code for the next frame. Only the fractional seconds field of the frame time stamp is given.

RECORD MODE	T1	T2
VLBA 8MHz	00	25
VLBA 4MHz	00	50
VLBA 2MHz	01	00
MKIII	00	50

12) START OBSERVATION

GO TO OBSERVING MODE: 14

13) OBSERVATION CONTROL

DATA INVALID MODE: 15
AUTO DATA VALID MODE: 16
END OBSERVATION: 17

An observation is started by a command given to the Track Recovery Card to go to OBSERVING. This action will automatically put the TRC in AUTO DATA VALID mode. In this mode the Track Recovery Card will use it's data valid algorithm to determine the validity status of each track. If things get flaky later on, the Track Recovery Cards can be told to go to DATA INVALID mode while the RT computer system or Deformatter tries to sort things out. After things look better the command to go back into AUTO mode may be given.

In DATA INVALID mode the TRC will go through the motions of observing (reading headers, tracking time, maintaining track parity error counts, etc) but all track data will be flagged invalid regardless of conditions.

14) HEADER READOUT

HEADER SNAPSHOT: 18
HEADER READOUT: 19 0T [GET 24 BYTES IN RETURN]

where,

T is the track number = 0 thru 7 or 8 for the system track.

The header readout strategy is as follows, every 131 msec the Deformatter will broadcast to all 4 TRC micros the snapshot command. It will then transfer the snapshot results, one track at a time into its own memory. The content of each

transfer is seen below.

TRANSFER	BYTE	TIME FORMAT : VLBA	TIME FORMAT : HKIII
1	1ST AUX BYTE		
2	2ND AUX BYTE		
3	3RD AUX BYTE		
4	4TH AUX BYTE		
5	5TH AUX BYTE		
6	6TH AUX BYTE		
7	7TH AUX BYTE		
8	8TH AUX BYTE		
9	1ST TIME BYTE	MJ	YD
10	2ND TIME BYTE	DS	DD
11	3RD TIME BYTE	SS	HH
12	4TH TIME BYTE	SS.	MM
13	5TH TIME BYTE	ss	SS.
14	6TH TIME BYTE	ss	ss
15	7TH TIME/CRC BYTE	CC	sC
16	2ND CRC BYTE	CC	CC
17	1ST ERRORS BYTE		
18	2ND ERRORS BYTE		
19	FRAME PARITY COUNT MS BYTE		
20	FRAME PARITY COUNT LS BYTE		
21	NUMBER OF HEADERS WITH ERRORS		
22	NUMBER OF RESYNCS		
23	NUMBER OF INVALID FRAMES		
24	NUMBER OF FRAMES SINCE LAST REQUEST		

The 1st error byte is an 8-bit frame parity error count (saturating at 253). The 2nd error byte is in two nibbles. The LS nibble is of the form cppp, where c = 1 for a header CRC error and ppp is the header parity error count (saturating at 7). The MS nibble contains 4 bits, sese. The 2nd s bit indicates whether the Track Recovery Card DPC chip assigned to the track of interest detected a sync word, the 2nd e bit indicated if that sync word was received at a time other than at the sync predict time (i.e., if there was a bit slip during the frame). The 1st s and the 1st e bits are the sync-det and sync-in-the-wrong-place bits for the fault tolerant off-line sync detector. The possibly legal values for the sese nibble are;

sese	CONDITION
0000	no sync word detected nohow
1000	imperfect sync detected, no bit slip during frame
1100	imperfect sync detected but there was a bit slip in frame
1010	perfect sync detected, no frame slip (error free condition)
1111	perfect sync detected but there was a bit slip in frame

The last six bytes of the header request transfer consists of quality monitor information. The two frame parity error count bytes is a 16-bit summation of all of the frame parity errors encountered (on the track requested) since the last header readout. Byte 21 of the readout gives the number of headers since the last readout (again for the track requested) that had errors (parity

errors, CRC errors, or incorrect sync), byte 22 gives the number of re-syncs since the last header readout, byte 23 the number of invalid frames for the track in question since the last header readout, and byte 24 is the total number of frames since the last readout. Byte 24 should be either 52 or 53 when header readouts are performed every 131 msec. The DFE will keep higher level integrations of these last 6 bytes for the RT system.

15) TIME REQUEST

TIME REQUEST: 1A [GETS 8 BYTES IN RETURN]

Every delay cycle the Deformatter will request the time from the TRCs. The result of this request will be a 7 byte time code and a one byte validity flag from each TRC micro. The Deformatter will use the time code to set the delay for each TRC during the next delay cycle. The format of the time returned is the same as the formatter time code;

00 JJ JS SS SS ss ss 0V FOR VLBA

or

YD DD HH MM SS ss s0 0V FOR MKIII

where the lower case s indicates the fractional part of the seconds count

and for 0V = 0000 000v, if v = 0, the TRC filtered time is not reliable
if v = 1, the TRC filtered time is reliable.

16) AUTO TEST

AUTO TEST: 1B PT

(default command is 1B 7F on a PBD or 1B 3F on the MINI TRANSPORT)

where for PT = wdsm tttt,

wdsm are observational parameters (see function code 12)
wdsm = 1111, watch dog timer on, demod on, narrow band sync mode, VLBA
wdsm = 1011, watch dog timer on, demod off, narrow band sync mode, VLBA
wdsm = 0001, watch dog timer off, demod off, wide band sync mode, VLBA
tttt = 1111, ALL TRACKS ACTIVE
tttt = 1110, NO CHANGE TO PRESENTLY SELECTED TRACKS
tttt = 0000, TRACK 0 ONLY ACTIVE
tttt = 0001, TRACK 1 ONLY ACTIVE
tttt = 0010, TRACK 2 ONLY ACTIVE
tttt = 0011, TRACK 3 ONLY ACTIVE
tttt = 0100, TRACK 4 ONLY ACTIVE
tttt = 0101, TRACK 5 ONLY ACTIVE
tttt = 0110, TRACK 6 ONLY ACTIVE
tttt = 0111, TRACK 7 ONLY ACTIVE
tttt = 1000, SYS TRACK ONLY ACTIVE

17) DEMOD BIT CONTROL (normally given by the function code 12 command)

DEMOD ON: 1C
DEMOD OFF: 1D

18) STATUS REPORTING

SYSTEM STATUS: 1E [10 BYTES RETURNED], SS XX TT AA I0 I1 TO YY YY FF

where,

SS gives the system status, where for SS = 0000 00dt;
- t = 0 the TRC filtered time is not reliable
- t = 1 the TRC filtered time is reliable
- d = 0 delay was updated
- d = 1 delay was not updated

XX not defined at this time.

TT is the number times the filtered time kept by the TRC micro has been reset since the last error report

AA is the number times the header DMA was aborted since the last error report

I0 is the number frames that channel 0 (for TRC micro being queried) has been invalid since the last error report.

I1 is the number frames that channel 1 (for TRC micro being queried) has been invalid since the last error report.

TO is the number times the sync watch dog timer has timed out since the last error report.

YY not defined at this time (2 bytes).

FF the total number of frames since the last error report.

19) CRT PROMPT MESSAGE

CRT PROMPT: 1F AA---AA 00---00 (16 BYTES TRANSFERRED)

Up to 15 ASCII bytes AA---AA are transferred. The remaining byte(s) 00---00 are needed to get the total data portion of the command to 16 (the total transfer is 17-bytes, the one byte function code plus 16 data bytes). The ASCII portion of the transfer is added to the CRT prompt after "TRACK RECOVERY".

20) DELAY VALUE (Deformatter responsibility)

DELAY: 8V DD DD

where,

V BITS 0 & 1 ARE THE 0, 1, 2, OR 3 BIT VERNIER DELAY
BIT 2 IS THE SELF TEST INTERVAL FLAG, WHERE 0 IS NORMAL AND 1 INDICATES THE SELF TEST INTERVAL

and

DD DD IS A 16-BIT DELAY IN NIBBLES REFERENCED TO THE LAST DEFORMATTER TIME REQUEST

APPENDIX I) TRACK MIX STAGE PROGRAMMING

TRACK MIX: 10 0X YY ZZ

where,

the two LSBs of X program which system track is assigned to the micro being addressed. YY and ZZ together define 8 2-bit fields each of which selects an input PBD track for a Track Recovery Card DPC chip. Track numbers are given in the recorder track numbering system.

in bit format where X YY ZZ = xx yyyy yyyy zzzz zzzz;

FOR TRACK RECOVERY CARD 1, MICRO A (DELAY CENTER 0)

xx	yyyy yyyy	zzzz zzzz	TRACK INTO DELAY CENTER	
		00	03	0
		01	19	0
		10	02	0
		11	18	0
	00		03	1
	01		19	1
	10		02	1
	11		18	1
00			03	2
01			19	2
10			02	2
11			18	2
	00		03	3
	01		19	3
	10		02	3
	11		18	3
		00	05	0
		01	21	0
		10	04	0
		11	20	0
	00		05	1
	01		21	1
	10		04	1
	11		20	1
00			05	2
01			21	2
10			04	2
11			20	2
	00		05	3
	01		21	3
	10		04	3
	11		20	3
00			00 (sys tk)	0
01			34 (sys tk)	0
10			01 (sys tk)	0
11			35 (sys tk)	0

FOR TRACK RECOVERY CARD 1, MICRO B (DELAY CENTER 1)

xx	yyyy yyyy	zzzz zzzz	TRACK INTO DELAY CENTER		
		00	07		0
		01	23		0
		10	06		0
		11	22		0
		00	07		1
		01	23		1
		10	06		1
		11	22		1
	00		07		2
	01		23		2
	10		06		2
	11		22		2
	00		07		3
	01		23		3
	10		06		3
	11		22		3
00			09		0
01			25		0
10			08		0
11			24		0
00			09		1
01			25		1
10			08		1
11			24		1
00			09		2
01			25		2
10			08		2
11			24		2
00			09		3
01			25		3
10			08		3
11			24		3
00			00	(sys tk)	1
01			34	(sys tk)	1
10			01	(sys tk)	1
11			35	(sys tk)	1

FOR TRACK RECOVERY CARD 2, MICRO A (DELAY CENTER 2)

xx	yyyy yyyy	zzzz zzzz	TRACK INTO DELAY CENTER		
		00	11		0
		01	27		0
		10	10		0
		11	26		0
		00	11		1
		01	27		1
		10	10		1
		11	26		1
	00		11		2
	01		27		2
	10		10		2
	11		26		2
00			11		3
01			27		3
10			10		3
11			26		3
	00		13		0
	01		29		0
	10		12		0
	11		28		0
00			13		1
01			29		1
10			12		1
11			28		1
	00		13		2
	01		29		2
	10		12		2
	11		28		2
00			13		3
01			29		3
10			12		3
11			28		3
00			01	(sys tk)	2
01			35	(sys tk)	2
10			00	(sys tk)	2
11			34	(sys tk)	2

FOR TRACK RECOVERY CARD 2, MICRO B (DELAY CENTER 3)

xx	yyyy yyyy	zzzz zzzz	TRACK INTO DELAY CENTER		
		00	15		0
		01	31		0
		10	14		0
		11	30		0
		00	15		1
		01	31		1
		10	14		1
		11	30		1
	00		15		2
	01		31		2
	10		14		2
	11		30		2
00			15		3
01			31		3
10			14		3
11			30		3
	00		17		0
	01		33		0
	10		16		0
	11		32		0
	00		17		1
	01		33		1
	10		16		1
	11		32		1
00			17		2
01			33		2
10			16		2
11			32		2
00			17		3
01			33		3
10			16		3
11			32		3
00			01	(sys tk)	3
01			35	(sys tk)	3
10			00	(sys tk)	3
11			34	(sys tk)	3

APPENDIX II) PBI OPERATION

When a playback drive (PBD) is started with the intention of processing an experiment, it is the responsibility of the RT system to establish initial PBD synchronization. Initial synchronization is defined as positioning the tape such that frame times recovered from track headers fall within a narrow time window relative to time as specified by the station delay model. The window requirement is satisfied when the Track Recovery Card (TRC) track buffer RAMs are between 20% and 80% full.

Once the initial transport positioning has been done, the TRC and Deformatter Card (DEF) work together to control the speed of the transport so as to keep the TRC track buffer RAMs approximately one half full as the system follows the delay model.

Below is a step by step procedure of the sync-up and operation of the PBI logic from the TRC protocol perspective. Some of the operational sync maintenance strategy described below is seen in flow chart form in figure 1.

1) The PBD is commanded to start playing. All preliminary commands such as head positioning, etc., have already been issued. All of the activity above is the responsibility of the RT computer system.

2) When the transport reaches at least 99% of the normal playback speed, the TRC and DEF are given commands to go into OBSERVE mode by the RT system. All TRC and DEF observational parameters necessary for the PBI cards to conform to the experiment being processed would have been issued by the RT computer prior to the OBSERVE command.

3) Once sync patterns begin to be recognized in the PBD track data, the TRC microprocessors will start analyzing the contents of the accompanying frame headers. The first task for the TRC is to find the experiment time. The TRC will use the time code of the first error free track header it sees to initialize it's internal filtered version of the experiment time. An error free header is one with no detected parity errors, good sync, and good CRC.

Each TRC microprocessor tries to maintain this filtered version of time. As frames continue to be received by the TRC, the TRC micro uses the parameters from the frame duration command (function code 13) to predict future header time codes from present time (adding the duration to the present time predicts the time code of the next header). The filtered time will allow the TRC to "ride through" occasional losses of track data.

4) The DEF micro will periodically read the TRC filtered time and compute the PBD position error using it and the RT supplied time reference parameter.

The DEF reads the TRC filtered time every 4.128 msec delay cycle. This time is presented to the Deformatter in the original

00 JJ JS SS SS ss ss

header time stamp format in BCD (see function code 1A). For MKIII the time stamp format will be of the form

YD DD HH MM SS ss s0.

A large PBD position error will probably occur at initial transport start-up but could also occur at any time during a tape pass (for example after a tape splice). If a large PBD position error is observed during the course of operations, the RS system has the responsibility to re-position the drive so as to eliminate this error.

The DEF also uses the filtered time from the TRC to set the delay to conform with the station delay model.

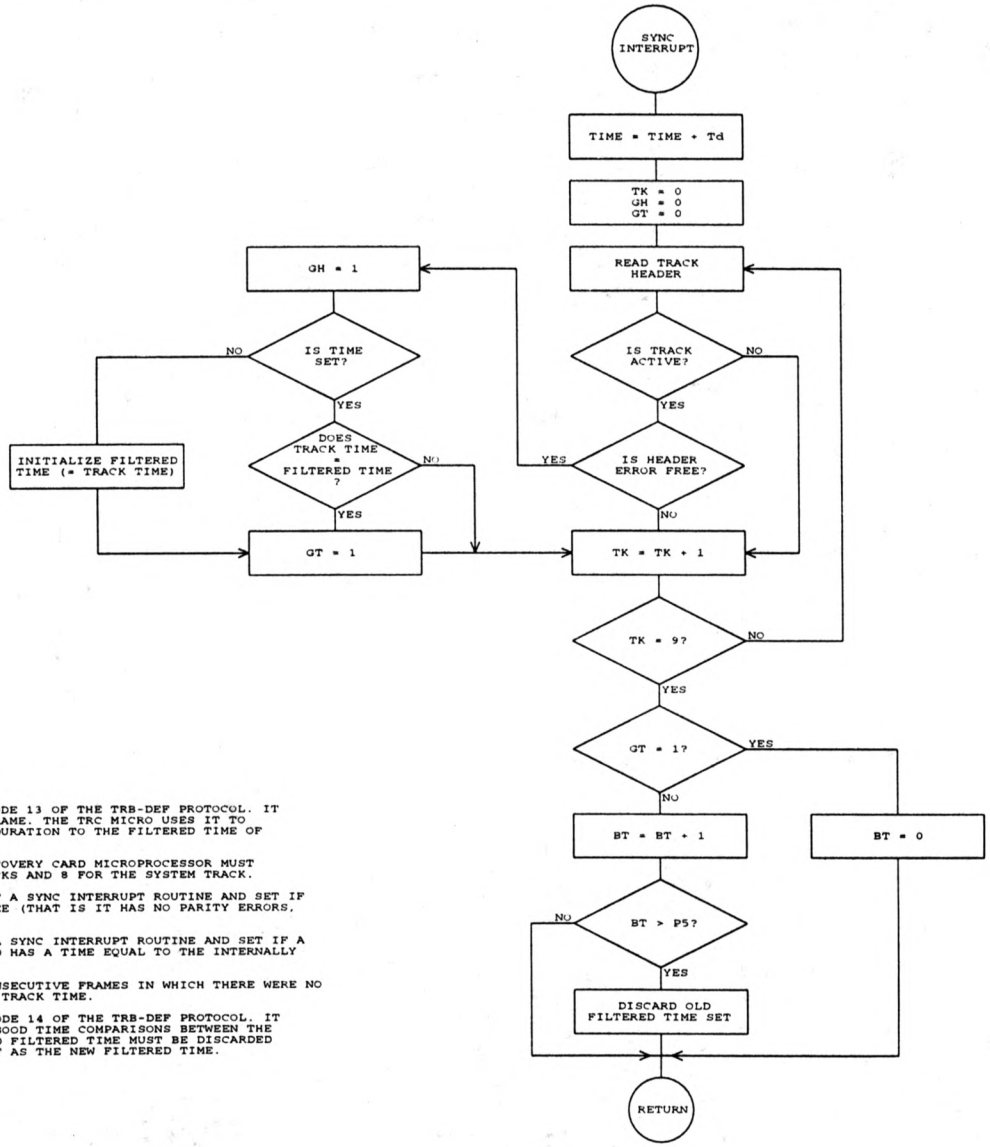
5) Initially, the TRC flags all track data as invalid. Once the RT computer confirms that the system is running smoothly by monitoring the status of the TRC and DEF, it can issue the AUTO DATA VALID command to the TRC cards. The AUTO DATA VALID command gives the TRC micros the authority to declare data to be valid. To this end they will use their own internal validity algorithm to determine on a track frame by track frame basis the validity of data driving

the DEF. A flow chart of the TRC data valid algorithm us seen in figure 2.

If at any time during a tape pass smooth operation ceases, such as would occur if a large PBD position error develops, the DEF or RT computer should issue DATA INVALID commands to the TRC micros. Once proper PBD operation is restored the return to AUTO DATA VALID mode can be commanded.

6) When a PBD is about to be stopped the TRC and DEF cards must be given END OBSERVATION commands. This action, taken while the PBD is still at full speed, will return the PBI cards to an idle mode insuring that the sudden loss or grabbling of frame sync as the PBD speed ramps down will not cause erratic operation. If the transport is then re-started in the other direction to continue with the experiment, system restart is accomplished by returning to step 1.

FIGURE 1
FLOW CHART OF TRC
TIME TRACKING ROUTINE



DEFINITIONS;

Td, AN OBSERVATION PARAMETER TRANSMITTED VIA FUNCTION CODE 13 OF THE TRB-DEF PROTOCOL. IT DEFINES THE TIME DURATION (IN BCD) OF A PBD TRACK FRAME. THE TRC MICRO USES IT TO PREDICT THE CURRENT FRAME TIME BY ADDING THE FRAME DURATION TO THE FILTERED TIME OF THE LAST FRAME.

TK, TRACK NUMBER. ENUMERATES THE TRACKS THAT A TRACK RECOVERY CARD MICROPROCESSOR MUST SUPERVISE. TK RUNS FROM 0 TO 8, 0 TO FOR 7 DATA TRACKS AND 8 FOR THE SYSTEM TRACK.

GH, GOOD HEADER FLAG. THIS FLAG IS RESET AT THE START OF A SYNC INTERRUPT ROUTINE AND SET IF ANY OF THE TRACK HEADERS BEING ANALYZED IS ERROR FREE (THAT IS IT HAS NO PARITY ERRORS, HAS GOOD SYNC AND HAS GOOD CRC).

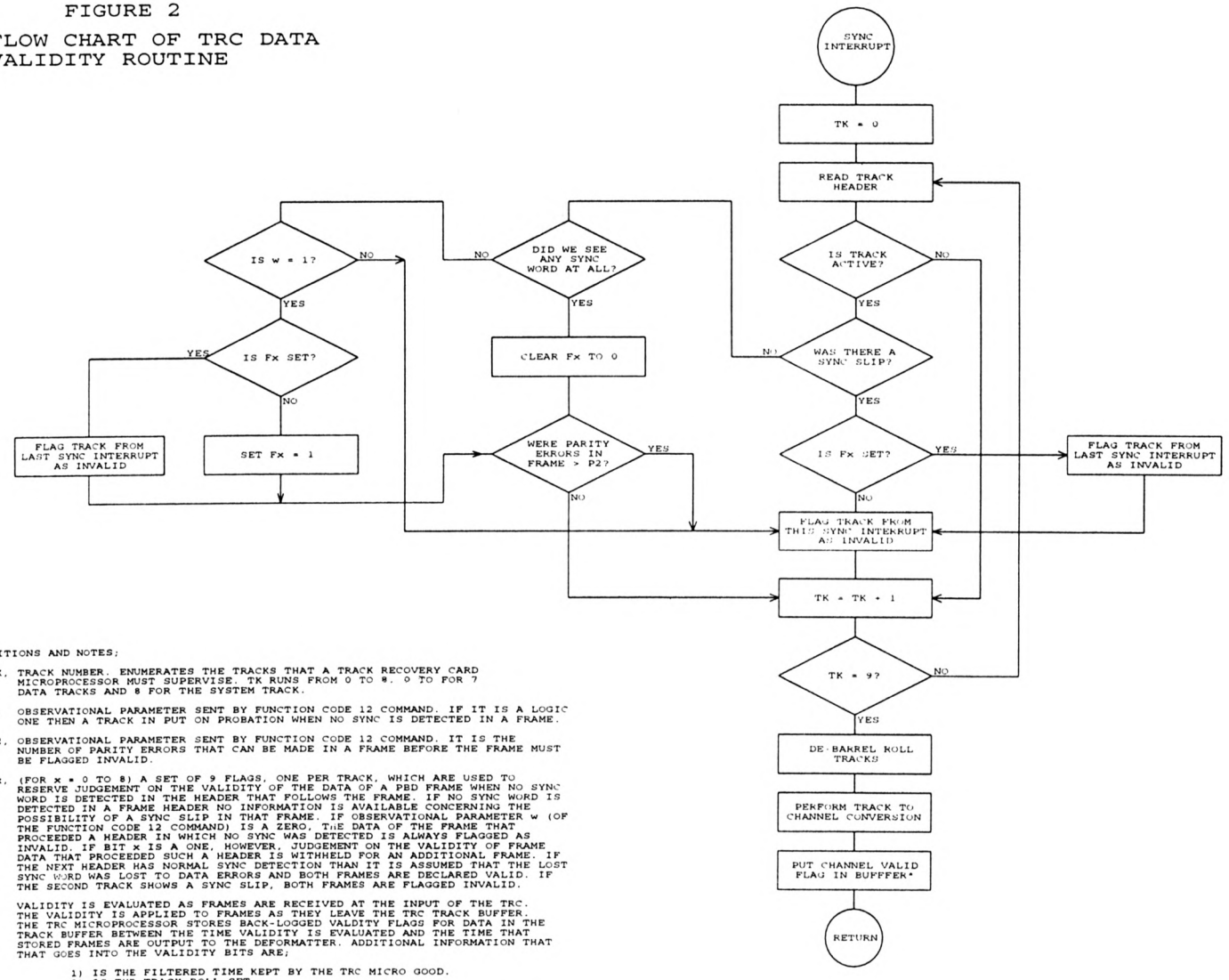
GT, GOOD TIME FLAG. THIS FLAG IS RESET AT THE START OF A SYNC INTERRUPT ROUTINE AND SET IF A TIME CODE IN ANY OF THE TRACK HEADERS BEING ANALYZED HAS A TIME EQUAL TO THE INTERNALLY KEPT FILTERED TIME.

BT, BAD TIME COUNT. THIS NUMBER COUNTS THE NUMBER OF CONSECUTIVE FRAMES IN WHICH THERE WERE NO AGREEMENTS BETWEEN THE FILTERED TIME AND ANY ACTIVE TRACK TIME.

P5, AN OBSERVATION PARAMETER TRANSMITTED VIA FUNCTION CODE 14 OF THE TRB-DEF PROTOCOL. IT GIVES THE NUMBER OF CONSECUTIVE FRAMES IN WHICH NO GOOD TIME COMPARISONS BETWEEN THE FILTERED TIME ANY A TRACK HEADER TIME BEFORE THE OLD FILTERED TIME MUST BE DISCARDED AND A NEW TIME, FROM AN ERROR FREE TRACK HEADER, SET AS THE NEW FILTERED TIME.

FIGURE 2

FLOW CHART OF TRC DATA VALIDITY ROUTINE



DEFINITIONS AND NOTES;

TK, TRACK NUMBER. ENUMERATES THE TRACKS THAT A TRACK RECOVERY CARD MICROPROCESSOR MUST SUPERVISE. TK RUNS FROM 0 TO 8. 0 TO FOR 7 DATA TRACKS AND 8 FOR THE SYSTEM TRACK.

P, OBSERVATIONAL PARAMETER SENT BY FUNCTION CODE 12 COMMAND. IF IT IS A LOGIC ONE THEN A TRACK IN PUT ON PROBATION WHEN NO SYNC IS DETECTED IN A FRAME.

P2, OBSERVATIONAL PARAMETER SENT BY FUNCTION CODE 12 COMMAND. IT IS THE NUMBER OF PARITY ERRORS THAT CAN BE MADE IN A FRAME BEFORE THE FRAME MUST BE FLAGGED INVALID.

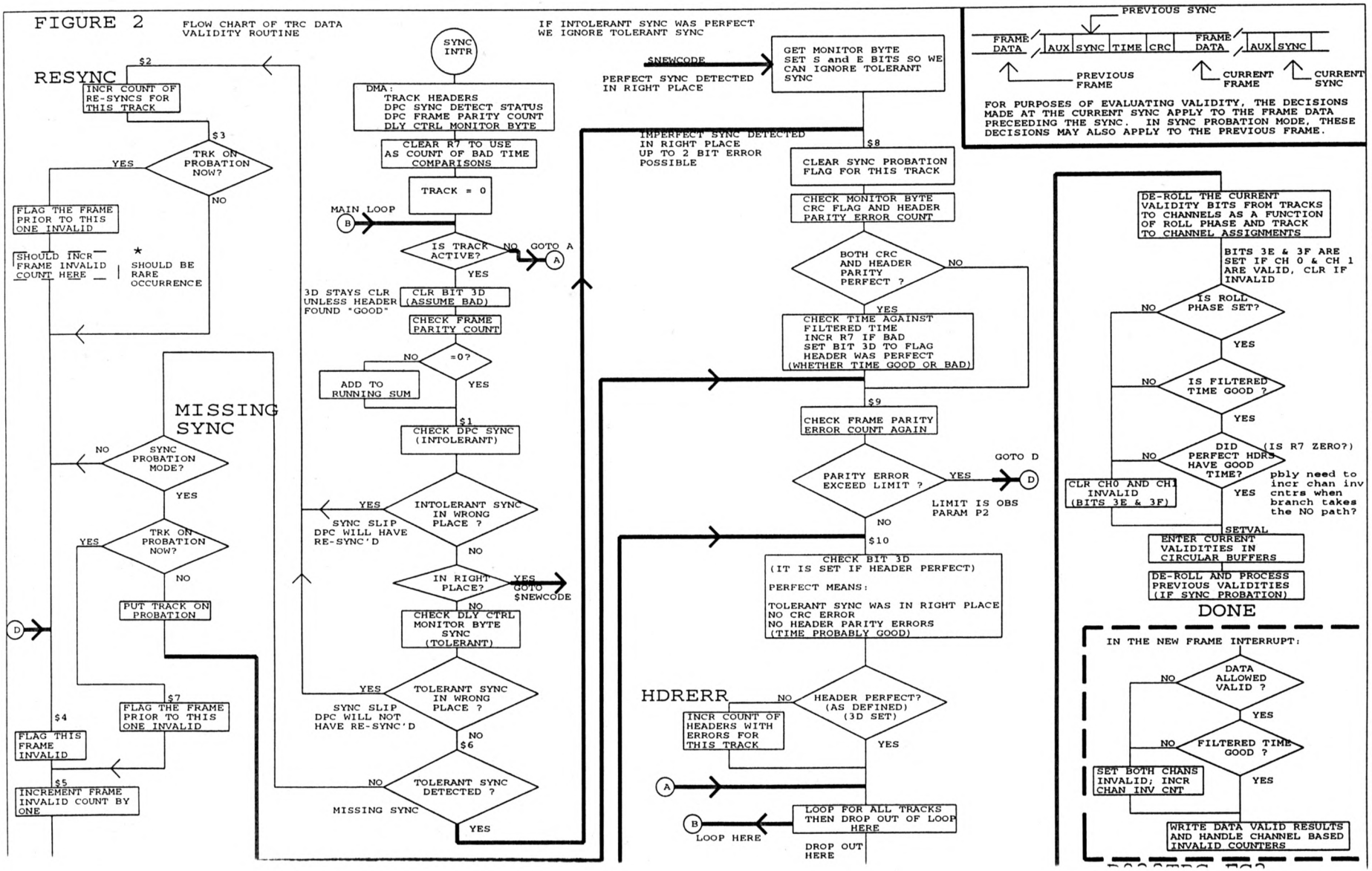
Fx, (FOR x = 0 TO 8) A SET OF 9 FLAGS, ONE PER TRACK, WHICH ARE USED TO RESERVE JUDGEMENT ON THE VALIDITY OF THE DATA OF A PBD FRAME WHEN NO SYNC WORD IS DETECTED IN THE HEADER THAT FOLLOWS THE FRAME. IF NO SYNC WORD IS DETECTED IN A FRAME HEADER NO INFORMATION IS AVAILABLE CONCERNING THE POSSIBILITY OF A SYNC SLIP IN THAT FRAME. IF OBSERVATIONAL PARAMETER w (OF THE FUNCTION CODE 12 COMMAND) IS A ZERO, THE DATA OF THE FRAME THAT PROCEEDED A HEADER IN WHICH NO SYNC WAS DETECTED IS ALWAYS FLAGGED AS INVALID. IF BIT x IS A ONE, HOWEVER, JUDGEMENT ON THE VALIDITY OF FRAME DATA THAT PROCEEDED SUCH A HEADER IS WITHHELD FOR AN ADDITIONAL FRAME. IF THE NEXT HEADER HAS NORMAL SYNC DETECTION THAN IT IS ASSUMED THAT THE LOST SYNC WORD WAS LOST TO DATA ERRORS AND BOTH FRAMES ARE DECLARED VALID. IF THE SECOND TRACK SHOWS A SYNC SLIP, BOTH FRAMES ARE FLAGGED INVALID.

* VALIDITY IS EVALUATED AS FRAMES ARE RECEIVED AT THE INPUT OF THE TRC. THE VALIDITY IS APPLIED TO FRAMES AS THEY LEAVE THE TRC TRACK BUFFER. THE TRC MICROPROCESSOR STORES BACK-LOGGED VALDITY FLAGS FOR DATA IN THE TRACK BUFFER BETWEEN THE TIME VALIDITY IS EVALUATED AND THE TIME THAT STORED FRAMES ARE OUTPUT TO THE DEFORMATTER. ADDITIONAL INFORMATION THAT THAT GOES INTO THE VALIDITY BITS ARE;

1) IS THE FILTERED TIME KEPT BY THE TRC MICRO GOOD.
2) IS THR TRACK ROLL SET.
3) ARE DATA VALIDS ALLOWED BY THE RTS.

FIGURE 2

FLOW CHART OF TRC DATA VALIDITY ROUTINE



APPENDIX IV Deformatter HCB Protocol

This page intentionally left blank.

DEFORMATTER HCB PROTOCOL NRAO DRAWING #A56000D027

Under sccs control in /magnolia2/vlbsoft/defasm/d027def.hcb
The tables and diagrams are under sccs control in
/magnolia2/vlbsoft/defasm/protocol as 386 Orcad files.

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA 22903

Revision History	Date	By
initial release	4DEC91	GR
Fnct 62, typoes	14JAN92	GR
Corrections, Additions, Refinements	12FEB92	GR
Final CV release	7JUL92	GR
Added fnct 3F and 75, misc notes	13Aug92	GR
Modified fnct 50 ZZ 20 added place holders for figures	11Feb93	CB
Added Func 66. Re-numbered bytes 1-7 to 0-6 for Func 61	01Mar93	CB
Clarify section 6	20Apr93	JG
Include GR supplied markups, including new functions 67 & 68	06Jun93	JG
Add new function 69. Add misc changes lost in xfer to sccs.	11Aug93	JG
Tell where diagrams are.	03Feb94	JG

****

INTRODUCTION

This document describes the protocol between the Real Time System (RTS) and the Deformatter cards in the system. The RTS must initiate all transfers. The Deformatter does not initiate any type of transfer to or from the RTS.

There are 24 Deformatters in the system. The break down is as follows:
In each of the 4 correlator racks:
- A HCB bus is assigned to a rack.
- 6 Deformatters in a rack.
  - There are 2 TRCs associated with each Deformatter. All HCB communications with the TRCs take place through the Deformatter. The TRC's HCB protocol is described in D028TRC.HCB.
  - There is one PBD associated with each Deformatter.
  - Each Deformatter has a HCB target number assigned to it. These target numbers in each rack are 1, 2, 4, 8, 16, 32.

The following functions are broken into sections. The sections are arranged as to when and how the the functions are normally used. Of course there are a few functions that may overlap into another section.

Note: All function code numbers are in hex.
All values given for function code parameters are in hex.
All other values, like byte count, are in decimal.

The following page serves as a table of contents for the functions which is then followed by a listing of appendices.

```
    <----- description ------>    <fnct #><----- parameters ------>
SECTION 1, General Functions
    NOP:                                00 ZZ
    WRITE MEMORY:                       01 AA AA AA AA CC CC DD----DD
    READ MEMORY:                        02 AA AA AA AA CC CC
    CALCULATE CHECKSUM:                 03 AA AA AA AA CC CC
    SEND CHECKSUM:                      04 [2 bytes returned]
SECTION 2, Power on System Initiation Functions
    Control Xilinx Personality (4626 bytes to transfer)
      1st    XFER:                      20 DD----DD (1542 byte xfer)
      NEXT   XFERS:                     21 DD----DD (1542 byte xfer, 2 xfers reqd)
      GO/NO GO FLAG:                    22 [1 byte returned]
    DEF PROMPT:                         23 P0 P1 P2 P3
SECTION 3, Observation Parameters/Control Functions
    SPECIAL POLAR MODE                  28 P0
    SPECIAL ROLL  MODE                  29 P0
    BIT SHUFFLER XILINX PERSONALITY (11,094 bytes to transfer)
      1st    XFER:                      30 DD----DD (1849 byte xfer)
      NEXT   XFERS:                     31 DD----DD (1849 byte xfer, 5 xfers reqd)
      GO/NO GO FLAG:                    32 [1 byte returned]
    DEMUX-DELAY PARAMETERS:             33 P0 P1 P2 P3
    OVERLAP & CNT BY 1or2 CTRL:         34 P0 P1
    DATA VALID DELAY WORD:              35 P0
    CROSS TRACK PARITY MODE:            36 P0 P1 P2 P3
    BARREL ROLL PARAMETERS:             37 P0 P1 P2 P3
    ACTIVE TRACKS:                      38 T0 T1 T2 .. T31 T32 .. T35
    SAMPLES PER DELAY:                  39 0N NN NN
    CONVERSION FACTORS:                 3A P0 P1 P2
    MODEL OFFSET SOURCE:                3B SS
    DEFORMATTER PIPELINE FACTOR:        3C P0 P1
    PBD SERVO TABLE (LOAD 1):           3D TT TT ... TT [1280 bytes total]
    PBD SERVO TABLE (LOAD 2):           3E TT TT ... TT [1280 bytes total]
    NEW DELAY INHIBIT COUNT:            3F XX
SECTION 4, Tape Pass Functions
    FORCE CHANNEL INVALID:              40 VV
    TIME REFERENCE:                     41 00 0M JD TT TT TT TT TT TT (VLBA)
                                or      41 01 YD DD TT TT TT TT TT TT (MKIII)
    BIT SHUFFLERS CONTROL BYTES:        42 S0 S1 ... S15
SECTION 5, Observation Start/Stop & Servo Control Functions
    START/STOP/SERVO CTRL:              50 ZZ PP
SECTION 6, Status Control Functions
    RETURN CROSS TRACK PARITY COUNT:    60 [  16 bytes returned]
    STATUS REQUEST:                     61 [   7 bytes returned]
    HEADER REQUEST:                     62 [1156 bytes returned]
    CALCULATED OFFSET:                  63 [  24 bytes returned]
    TIME REQUEST:                       64 [  28 bytes returned]
    RETURN TIME & CALCULATED OFFSETS:   65 [  56 bytes returned]
    RETURN MODEL OUT OF LIMITS STATUS:  66 [   6 bytes returned]
    RETURN ONE HEADER                   67 [  30 bytes returned]
    RETURN TRC GENERAL STATUS           68 [  72 bytes returned]
    SPECIAL STATUS REQUEST:             69 [   7 bytes returned]
SECTION 7, Test Functions
    SELF TEST:                          70 TT
    PBD TEST DATA:                      71 TT
    TEST PATTERN DOWNLOAD:              72 [1032 bytes]
    TEST CYCLE:                         73
    SYNC PULSE ENABLE:                  74
    ERROR DISPLAY DISABLE:              75 TT
```

SECTION 8, Track Recovery Card Communication Functions

WRITE TO TRC:	80 0X CC CC [TRC message]
READ TRC:	81 0Y CC CC RR RR [TRC message]
RESET TRC:	82 ZZ

/***/

FUNCTION DURATIONS

The below table shows the name for the function code in the include file hcbFuncCodes.h. Also shown are the times in milliseconds from a survey of how long the functions take. Approximate data rates in KBytes/second are shown. There are potential problems for functions that have an equivalent transfer rate of less than 50 KBytes/sec, and take more than 10 ms. For a detailed explanation of the measurements taken see /home/azaleal/staff/jgreenbe/notes/survey.txt. Functions labled INIT below are only used during initialization, where timing is not critical. The timing of functions labeled TEST is not of concern when observing. Data rates are not give where minimal data is transferred.

		OBSERVING		NOT OBSERVING	
INCLUDE FILE NAME	FC	TIME(ms)	RATE	TIME	RATE(KBytes/sec)
WRITETARGETMEM	0x01	Used for writing sequencer files. Can't observer while loading.			
Sequencer 1				194	21.1
Sequencer 2				194	21.1
Sequencer 3				53	19.5
RDSTOREDCHKSUM	0x04	Fast since returns previously stored chksum			
CTLXIL	0x20	INIT			
CTLXILCONT	0x21	INIT			
CTLXILFLG	0x22	INIT			
DEFID	0x23	INIT			
SPCL_POLAR	0x28	0.447		0.185	
SPCL_ROLL	0x29	0.469		0.187	
SHUFXIL	0x30	30, 31, and 32 Combined below:			
SHUFXILCONT	0x31	30, 31, and 32 Combined below:			
SHUFXILFLG	0x32			700	15.8 Downloading Xilinx kills observing mode.
DEMUXDLY	0x33	0.540		0.207	
OVERLPCNT	0x34	0.508		0.200	
DATAVALIDDLY	0x35	0.502		0.192	
CROSSTRKPAR	0x36	0.554		0.211	
BARRELROLL	0x37	0.581		0.231	
ACTIVETRACKS	0x38	Not used			
SAMPLESPERDELAY	0x39	0.540		0.220	
CONVFACTS	0x3A	0.540		0.218	
MODOFSSRC	0x3B	0.522		0.205	
PIPEFACT	0x3C	0.555		0.211	
LDPBISERVO0	0x3D	3D and 3E combined below:			
LDPBISERVO1	0x3E	18	142	31.3	81.8
LOWBWPARM	0x3F	0.554		0.218	
FORCECHINVALID	0x40	0.546		0.220	
TIMEREF	0x41	0.688	13.1	0.278	32.37
SHUFCTL	0x42	0.794	20.1	0.330	48.5
DEFOBSCTRL	0x50	61.6		65.0	
RTNCROSSTRKPAR	0x60	0.883	18.1	0.364	44.0
HDRSTATUS	0x61	0.540		0.310	
HDRRQST	0x62	8.4	137	13.3	86
CALOFFSET	0x63	1.065	22.5	0.430	55.8
TIMERQST	0x64	1.195	26.8	0.496	64.5
TIMECAL	0x65	1.621	34.5	0.675	83.0
MODELLIMIT	0x66	0.755	7.95	0.305	19.5
ONEHEADER	0x67	0.914	32.8	0.555	54.0
TRCSTAT	0x68	1.95	36.9	0.842	85.5
SPECSTAT	0x69	0.795	8.81	0.318	22.0
DEFSELFTEST	0x70	0.620		0.254	

PBDTSTDAT	0x71	0.634		0.254	
DEFTSTPAT0	0x72	91.7	11.3	57.3	18.0
DEFTSTPAT1	0x73	0.634		0.251	
DEFSYNCENBL	0x74	TEST			
DEFERRDSPENBL	0x75	0.667		0.258	
WRITETRC	0x80	See the TRC Protocol			
READTRC	0x81	See the TRC Protocol			
RESETTRC	0x82	10.0		10.0	

APPENDICES LIST

SECTION 1

General Functions

* NOP: 00 ZZ
Where ZZ is a dummy byte that the Deformatter must read.

68k Main Memory Write and Read, (including memory mapped sequencers)
* WRITE MEMORY: WRITETARGETMEM 01 AA AA AA AA CC CC DD----DD
* READ MEMORY: 02 AA AA AA AA CC CC

Where:
AA AA AA AA is a 32-bit start address and CC CC is a
transfer count with MSbyte first.

Note:
Three mode dependent sequencers on the Deformatter card are
memory mapped and loaded with the WRITE MEMORY function code.
(Their contents cannot be read. A checksum is done by the Deformatter
as the bytes are received from the HCB). These sequencers will be
loaded with pre-canned files for a particular mode. Appendix 2 describes
the nomenclature for these sequencer file names. The following
table lists the memory mapped address and transfer count:

SEQUENCER	HEX STARTING ADDRESS	TRANSFER COUNT
S1 FRINGE & DELAY CYCLE & CLKS (1)	00 46 00 00	4096
S2 READ/WRITE TIMING GENERATOR	00 48 00 00	4096
S3 BUFFER OVERLAP CONTROL	00 4A 00 00	1032

(1) The Deformatter keeps a copy of this sequencer in its RAM. This
allows function code 72 & 73 to OR a test pattern into the
contents of the sequencer.

* CALCULATE CHECKSUM: CalcCkSum 03 AA AA AA AA CC CC [2 bytes returned]
This is used for the normal 68K ram.
AA AA AA AA is a 32-bit start address and CC CC is a
transfer count with MSbyte first.
The returned checksum is in the form:
AAAAAAAA AAAAAAAA Accumulated sum
+ 00000000 BBBBBBBB current byte being summed into checksum

AAAAAAAA AAAAAAAA new Accumulated sum

* SEND CHECKSUM: SendCkSum 04 [2 bytes returned]
2 bytes returned as above.
This returns the last calculated checksum. It could be from a
sequencer being loaded, or from a call to CALCULATE CHECKSUM above.

SECTION 2

Power on System Initiation Functions - the following functions should be done whenever the system is powered on.

* Load the Sequencers - see above for 68k memory write functions above for details.

* Control Xilinx Personality (4626 bytes to transfer)

1st XFER:	CTLXIL =	20	DD----DD (1542 byte xfer)
NEXT XFERS:	CTLXILCONT =	21	DD----DD (1542 byte xfer, 2 xfers reqd)
GO/NO GO FLAG:	CTLXILFLG =	22	[1 byte returned]

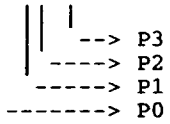
Notes:
1) This function programs both the Barrel Roll and the Data Invalid Xilinx.
2) These Xilinx personalities are programed by the file DEFCTL.MCS.
3) These Xilinx personalities are mode independent.
4) This must be followed at some point by the DEF PROMPT (function 0x23) command so that interrupts will be re-enabled.
5) This command normally must also be followed by the SPECIAL POLAR MODE command to turn off the swap of CH1/CH3. The Barrel Roll Xilinx initial configuration after download turns on the channel swap.

* Load the 68k micro code using the 68k memory write function described above.

* DEF PROMPT: DEFID = 23 P0 P1 P2 P3

Where:
P0 = HCB bus ID number.
P1 = Separator, usually 2Dh, which is a "-".
P2 = Target number.
P3 = Prompt ending character.

Example: DEF06-16>



Note: The Deformatter's hex code, DEFRAM.HEX, and the TRC's hex code must be loaded before this function is sent by the RT system.

The Deformatter Xilinx downloads leave certain interrupts dis-abled. The prompt command is used by the Deformatter to re-enable the interrupts and do other housekeeping, as well as providing the prompt to be used by the monitor routine in the Deformatter. The prompt command is thus the indicator that the most basic downloads have been completed.

It is necessary to repeat the prompt command anytime that a new bit shuffler personality is downloaded. This is the one time other than initial card download where this is necessary.

SECTION 3

Observation Parameters/Control Functions
- these are parameters that remain constant for the duration of an observation. These parameters are sent to the Deformatter while the Deformatter is in the idle state.

* SPECIAL POLAR MODE SPCL_POLAR = 28 P0
P0= 0 for normal (no swap of CH1/CH3 or CH5/CH7)
P0= 1 for swap (CH1 swapped with CH3; CH5 swapped with CH7)

When swapped, this allows the correlator to process a 1 bit stream to 4 track, 2 bit sampling, polar mode experiment. Polar mode requires "adjacent" FFT pipelines (e.g 0 and 1) to process the two polarizations. But 1-4, 2 bit record mode uses all 8 tracks in a delay center for one channel. The swap of channels at the Deformatter output thus makes this mode possible.

This command normally must be used following the download of the Control Xilinx Personality since the Barrel Roll Xilinx default configuration causes the CH1/CH3 swap to occur. This command must be used to turn off the swap unless the swap happens to be required.

* SPECIAL ROLL MODE SPCL_ROLL = 29 P0
P0= 0x00 for normal (no swap of barrel roll phase)
P0= 0x88 for swap (invert MS bit of 4 bit roll phase)

Normally always set P0 = 0x00. Only use P0= 0x88 when rolling in groups of 16 and you need to compensate for the group of 8 swap in the Formatter. When P0= 0x88, the Deformatter swaps the two groups of 8 between adjacent delay centers during roll phases 0-7 and does not swap them during roll phases 8-15. This is opposite of the normal operation.

* BIT SHUFFLER XILINX PERSONALITY (11,094 bytes to transfer)
1st XFER: SHUFXIL = 30 DD----DD (1849 byte xfer)
NEXT XFERS: SHUFXILCONT = 31 DD----DD (1849 byte xfer, 5 xfers reqd)
GO/NO GO FLAG: SHUFXILFLG = 32 [1 byte returned]

Notes:
1) These Xilinx personalities are mode dependent.
2) Figure 1 in Appendix 2 lists the required file for the various modes.
3) This must be followed at some point by the DEF PROMPT (function 0x23) command so that interrupts will be re-enabled.

* DEMUX-DELAY PARAMETERS: DEMUXDLY = 33 P0 P1 P2 P3

Notes:
1) This control word controls the operation of the Demux-Delay PALs found on logic dwgs L008D07.SCH & L008D08.SCH.
2) See Figure 2 in Appendix 3 for details of the program word.

* OVERLAP & CNT BY 1or2 CTRL:OVERLPCNT = 34 P0 P1

Notes:
1) This function programs the RAM READ OFFSET REGISTER found on dwg L008D13.SCH.
2) See Figure 3 in Appendix 3 for details.

* DATA VALID DELAY WORD: DATAVALIDDLY = 35 P0

Notes:
1) This parameter is mode dependent.
2) This byte programs a delay counter that times DATA VALID signals from the TRC with the data as they are written into the Buffer RAM. This byte goes into a control register located within the DATA INVALID Xilinx chip, dwg L008D17.SCH.
3) See Figure 4 in Appendix 3 for the format of the control word.
4) See Figure 1 in Appendix 2 for mode dependent values.
5) The number is unsigned and in the range of 1-FF with zero being an illegal value.

6) This function will disable disable test data from the PBI. If test data is to be enabled then issue HCB cmd 71.

* CROSS TRACK PARITY MODE: CROSSTRKPAR = 36 P0 P1 P2 P3

Notes:
1) These parameters control how the PALs found on drawings L008D03.SCH - L008D06.SCH handle cross track parity.
2) See Figure 5 in Appendix 3 for details.

SECTION 3

* BARREL ROLL PARAMETERS: BARRELROLL = 37 P0 P1 P2

Where:
P0 P1 = Delay setting for New Frame signal from the TRC. We think this is mode independent. The official value is 0x101 as of 6/22/93. This is an unsigned number in the range of 001 to 1FF, with zero being an illegal value.
P2 = 0 no barrel roll. The P0 P1 value should always have the proper value set even if barrel roll is not active. A signal that P0 P1 develops is also used by the data invalid logic.
P2 = 1 roll in 8 track groups - step 1 track per frame.
P2 = 2 roll in 8 track groups - step 2 tracks per frame.
P2 = 3 roll in 8 track groups - step 4 tracks per frame.
P2 = 4 roll in 16 track groups - step 1 track per frame.
P2 = 5 roll in 16 track groups - step 2 tracks per frame.
P2 = 6 roll in 16 track groups - step 4 tracks per frame.

See Figure 1 in Appendix 3 for further details.

Note: If barrel roll is on, then phase center 3 must have the largest delay value!!

* ACTIVE TRACKS: ACTIVETRACKS = 38 T0 T1 T2 .. T31 T32 .. T35
(THIS FUNCTION NOT CURRENTLY USED BY ANY SOFTWARE)
Where:
T0 = data track 0
..
T31 = data track 31
T32 = system track 0
..
T35 = system track 3

Where the value of T0 .. T35:
Tx = FF if PBD track x is active.
Tx = 00 if PBD track x is inactive.

Notes:
track numbers are given in the formatter's track numbering system.

* SAMPLES PER DELAY: SAMPLESPERDELAY = 39 0N NN NN

Where:
0N NN NN is the number of Nyquist samples (at the playback rate) that will be processed during a 4.096 msec delay update cycle. This parameter will stay constant for the entire run.

See Appendix 1 for further information on how this value is used.

SECTION 3

* CONVERSION FACTORS: CONVFACTS = 3A P0 P1 P2

Nyquist rate samples to track bits conversion factors.
Where:
P0 = 80 for a 8 MHz record rate
= 04 for a 4 MHz record rate
= 00 for a 2 MHz record rate

P1 = 00 for 1 bit stream recorded on 4 tracks.
= 20 for 1 bit stream recorded on 2 tracks.
= 40 for 1 bit stream recorded on 1 track.
= 60 for 2 bit streams recorded on 1 track.
= 80 for 4 bit streams recorded on 1 track.

P2 = 00 for a sample factor of 1 (Nyquist sampling)
= 02 for a sample factor of 2 (2 times oversampling)
= 04 for a sample factor of 4
= 06 for a sample factor of 8
= 08 for a sample factor of 16

* MODEL OFFSET SOURCE: MODOFSSRC = 3B SS

Where:
SS = 01 except as below.
= 02 for PBD 1 assigned to FFT engine 19.
= 02 for PBD 12 assigned to FFT engine 10.
= 02 for PBD 13 assigned to FFT engine 9.
= 02 for PBD 24 assigned to FFT engine 0.

This byte is used to select the input source for the DELAY MODEL CORRECTION. That is, it selects which FFT CONTROL CARD model generator supplies DELAY MODEL CORRECTION to the Deformatter. The multiplexer that this function controls is found on dwg L008D26.SCH.

This mux selection is made effective when an observation starts. This control bit is also merged into the Bit Shuffler Control words whenever the Deformatter receives a fnct 42 (Bit Shuffler Control Word).

SECTION 3

* Deformatter PIPELINE FACTOR: PIPEFACT = 3C P0 P1

A 16 bit signed constant that is used in the delay calculation which takes place each New Delay cycle. The units of this number are basically in track bits and is used to adjust for the number of pipeline stages from the TRC to the output of the DEF. For modes 1-2 and 1-4 this number is divided, the remainder is used added to the 0-3 bit delay on the DEF; while the integer part is used in the delay calculation as with the other modes. Refer to appendix 1 for a description of the delay calcuation.

The Pipeline Factor is mode dependent. The value for each mode can be found in figure 1 in Appendix 2.

The values were determined by using the Mini-Transport, MT. The MT was programmed to put out a single one bit at the start of each 4ms delay cycle. The Pipeline Factor was then adjusted so that this one appeared at the input of the FFT card at the start of a fringe cycle.

* PBD SERVO TABLE (LOAD 1): LDPBISERVO0 = 3D TT TT ... TT [1280 bytes total]
PBD SERVO TABLE (LOAD 2): LDPBISERVO1 = 3E TT TT ... TT [1280 bytes total]

The servo table is a 256 X 10 byte table. The offset error calculated by the Deformatter is used to index this table to obtain speed and duration parameters that are sent to the PBD. See Appendix 5 for further details.

SECTION 3

* NEW DELAY INHIBIT COUNT: 3F XX

This byte equals the max number of times that New Delay and New Fringe may be gated off to the RAM read and write address counters. This gating is used when the data into the RAM is at such a slow rate that not enough bits are written to do a FFT before a the next New Delay comes along and we start over.

When the data rate in is slow the delay also changes at a very slow rate. If delay does not change then the reset to the RAM address counters may be inhibited, thus eliminating the time to fill the RAM buffer except when the delay changes or when number of times it has been inhibited equal this byte. The Deformatter checks to see if the delay has changed, if not then reset to the address counters will be inhibited for a max number of consecutive times that is specified by this parameter.

A max value is required for two reasons. Each consecutive time we inhibit a delay change the TRC's read address pointer moves 64 bits away from its ideal center of the buffer. The second reason has to do with the read write pointers to the Deformatter's RAM. The write pointer gains on the read pointer due to the 4 cycle gap.

A value of 0 or FF will prevent the Deformatter from inhibiting New Delays.

* FORCE CHANNEL INVALID: FORCECHINVALID = 40 VV

The byte VV is a validity mask that can force data as being valid/invalid for any of the 8 channels leaving a Deformatter card. This function can be issued after an observation has started if data goes bad for some reason. See Figure 4 in Appendix 3 for more details of the definition of VV. It shows bits 0-7 of VV mapping to channels 0-7. A bit being 1 forces invalid. A bit being 0 enables validity control from the TRC.

When the deformatter does not flag all channels invalid then it will apply the above validity mask. This mask can force any combination of the 8 channels as invalid. If this mask is not forcing a channel invalid then the TRC's validity is used to determine a channel's validity. The validity bit from the servo table is identical to checking to see if the bit is in the TRC buffer.

The following is how the data invalid code is used:

- Force invalid at start observation.
- Force invalid during observation when servo flag is false.
- Force invalid during observation when servo flag is true but validity bit in servo table false.

- Apply the real time system's validity mask and allow the validities from the TRC to determine the channel's validity when in observating, servo flag is true and validity bit from servo table is true.

SECTION 3

- Force invalid at stop observation

Note: this function is also listed below in Section 4

SECTION 4

Tape Pass Functions - These parameters need to be set before an observation starts and on all subsequent tape passes. Some functions, like the time reference, are sent during a tape pass.

* FORCE CHANNEL INVALID: FORCECHINVALID = 40 VV

SEE THE WRITE UP IN SECTION 3 (on the previous page)

* TIME REFERENCE: TIMEREF = 41 XX 0M JD TT TT TT TT TT TT (VLBA)
or 41 XX YD DD TT TT TT TT TT TT (MKIII)

Where:

XX	Is a flag byte for the Deformatter where: XX= 0 when the time reference is for VLBA tape format. XX= 1 when the time reference is for MKIII tape format.
0M JD or	(Modified Julian Day for VLBA format)
YD DD	(Year Day Day Day for MKIII format), are the day/year time tag for the TT ... TT time referenced below. Note: The value for MJD or YDDD is BCD value that should be identical in format to what is read from the tape.
TT TT TT TT TT TT	is a 6 byte fixed point number that is the time reference for the model update cycle. This time reference parameter is the number of Nyquist rate samples (at the playback rate) between the start of the current Julian day and the first bit that is to be processed by the correlator in the new model update cycle. If more than one phase center is being tracked any phase center can be used as the time reference.

Notes:

1) See Appendix 1 for further information.
2) For further discussion on VLBA tape format refer to specification A56000N003.
3) Action taken by the Deformatter when this code is received:
   a) If the Deformatter is not observing
      then when told to start observing the Deformatter will start on the first 131ms tic it sees using this time reference as its initial value. The flag byte is used at this time to set the VLBA / MKIII mode, so a time reference with the correct flag byte ***must*** be received prior to the start obs command.
   b) If the Deformatter is in observe mode
      then the Deformatter updates its time reference on the first 131ms tic it sees after receiving a new time reference from the RT system. The flag byte is ignored while observing, it only takes effect when observation is started as noted in a).

SECTION 4

* BIT SHUFFLERS CONTROL BYTES: SHUFCTL = 42 S0 S1 ... S15

Where:

S0	S1	S2	S3	are the control bytes for channel 7 and 6
S4	S5	S6	S7	are the control bytes for channel 5 and 4
S8	S9	S10	S11	are the control bytes for channel 3 and 2
S12	S13	S14	S15	are the control bytes for channel 1 and 0

Notes:

1) These control bytes select not only which tape tracks are put together as a channel but also what order the data bits are arranged in.
2) These control bytes are loaded into a control register inside each of the 4 Bit Shuffler Xilinx's. These control registers address the multiplexers within the Xilinx's. Drawings L008D09.SCH - L008D12.SCH show the 4 Xilinx's.
3) The format of the bits in these control bytes vary with the various Bit Shuffler personalities which are mode dependent. The Figures in Appendix 4 show these variations.
4) Other dwgs that may be useful in programming these bytes are:
   K004D01.BLK - block diagram of data flow from sampler to the FFT engines.
   K006D0x.BLK - block diagram of the Xilinx personality.
   Where x = the personality number.
   xxxxxxx.TIM - timing information which includes a block diagram showing how the tracks and bits are assembled into a channel for a FFT engine.
5) The Model Offset Control bit, function code 3B, is merged into this function.

SECTION 5

Observation Start/Stop & Servo Control Functions

* START/STOP/SERVO CTRL: DEFOBSCTRL = 50 ZZ PP

Where:

ZZ = 00 Stop observation, both Deformatter and TRC will go into an idle state. PP is don't care.
= 01 Start observation and control the PBD servo.
= 02 Start observation but do not control the PBD servo.
= 03 Deformatter will initialize the Time Reference by reading Time from TRC micro 0, start observation and control the PBD servo.
= 04 Deformatter will initialize the Time Reference by reading Time from TRC micro 0, start observation but not control the PBD servo.
= 10 Continue observation and control the PBD servo.
= 20 Continue observation but do not control the PBD servo.
= 31 Same as 01 except that DEF will not read headers from the TRC.
= 32 Same as 02 except that DEF will not read headers from the TRC.
= 33 Same as 03 except that DEF will not read headers from the TRC.
= 34 Same as 04 except that DEF will not read headers from the TRC.
= 42 Same as 02 except that DEF applies the RTS validity mask instead of forcing data invalid. DEF will normally force data invalid if it doesn't have the servo. In the DEF's HD3 display, "RTSv" will be displayed in the PBD cmd buffer field. The "v" means that the deformatter is allowing the validities from the TRC and the RTS's mask to determine the validity state. If "RTSi" is displayed, then the deformatter has forced data invalid because it isn't controlling the servo.

PP = 00 Deformatter sends nothing to the TRCs. The exception being when ZZ = 0 in which case the Deformatter sends function 17 the TRCs.
= 01 Deformatter sends functions 0D & 14 (observe, connect to the PBD) to the TRCs.
= 02 Deformatter sends function 15 (data invalid mode) to the TRCs.
= 03 Deformatter sends function 16 (data valid mode) to the TRCs.
= 10 Deformatter sends functions 0C & 1B (autotest, connect to test frame) to the TRCs.
= 11 Deformatter sends functions 0C & 14 (connect to test frame and start observing).
= 20 Deformatter enables the test frame output from the PBD, then sends functions 0D & 1B (observe, connect to PBD) to the TRCs. The test frame originates from the Master MCC. The test frame will stay enabled until the Deformatter receives a function code of 71 00.

Notes:
1) It is imperative that when the Deformatter receives the flag to control the PBD servo that the PBD is at the normal drive speed and no commands are waiting to be executed. The RT system should not send any commands to the PBD if the Deformatter is to provide proper servo control.
2) Refer to the description under function 40 of how the deformatter handles validities.

SECTION 6

Status Control Functions - status information returned by the Deformatter
at the request of the RTS.

* RETURN CROSS TRACK PARITY COUNT: RTNCROSSTRKPAR = 60 [16 bytes returned]
Where:
byte 0 = MSB of cross track parity count for tracks 0-7.
..
byte 3 = LSB of cross track parity count for tracks 0-7.

byte 4 = MSB of cross track parity count for tracks 8-15.
..
byte 7 = LSB of cross track parity count for tracks 8-15.

byte 8 = MSB of cross track parity count for tracks 16-23.
..
byte 11 = LSB of cross track parity count for tracks 16-23.

byte 12 = MSB of cross track parity count for tracks 24-31.
..
byte 15 = LSB of cross track parity count for tracks 24-31.

* STATUS REQUEST: HDRSTATUS = 61 [7 bytes returned]

See hcbDefHdrStat in hcbLoadTestFix.c

byte 0 in bit format:

```
  7 6 5 4  3 2 1 0
--------------------
  1 0 0 0  0 0 0 0 = DEF updating headers and TRC status.
  0 0 0 0  0 0 0 0 = DEF has a complete set of headers rdy for the RTS.
  0 0 0 0  1 1 1 1 = DEF is idle.  RTS should check other status info.

= 00  Deformatter has a complete header buffer ready to send to the RTS.
= 0F  Deformatter is idle, not updating the header buffer.
      The deformatter is not observing.  The terminal should receive
      the message "TRC micro(s) not responding, IRQs disabled".
```

This function inhibits header reading. With IRQ's(Interrupt Requests) off, the DEF checks this byte before returning it to the RTS. If the IRQs are not turned off, then the status could change from the time it is read and to when a decision is made. This would not happen very often but it definitely would happen. Turning the IRQs off insures that the interlocking mechanism will work correctly every time. If byte 0 is found equal to 0, then the DEF will stop reading headers from the TRC's until the RTS has read the set of headers from the DEF, using HCB function 62.

Note: The RTS is expected to read headers if this byte is 0.
See function 69, which is identical, except header reading is not inhibited.

SECTION 6

byte 1 = 0 All TRC micro's are responding, normal operation.
>= 1 One or more TRC micros not responding.
The bits are decoded as follows:

```
7 6 5 4  3 2 1 0
  : : :  : : : :.... 1 = micro 0 dead
  : : :  : : :..... 1 = micro 1 dead
  : : :  : :........ 1 = micro 2 dead
  : : :  :.......... 1 = micro 3 dead
  : : :.... 1 = timed out while writing to both A's, 0 & 2
  : :..... 1 = timed out while writing to both B's, 1 & 3
  :........ 1 = timed out while writing all the micros.
```

If a micro does not respond, the below terminal error message results.
"TRC micro(s) not responding, IRQs disabled."
This is a fatal error. The DEF has stopped observing and has set a flag for the RTS, which gives the non-zero byte 1 value.

When a byte is sent to a micro, and the watch dog timer times out, the following message is displayed.
"Timed out, display TRCmicroStatus (FL cmd) for more info."
The flag for the RTS is set, but the DEF does not stopped observing.

byte 2 = 00 Deformatter is not enabled to servo PBD.
= FF Deformatter is enabled to servo PBD.

byte 3 = The number of times a received new time reference did not match the linear model tracked by the Deformatter. The maximum count of FF is held until the observation is stopped and restarted. Each time the Deformatter receives a time reference, the following comparision takes place:
For each of the 4 phase centers the current time reference is added to the current delay for that phase center. The same is done with the new time reference and the new delays for each phase center. Then the above 4 values from the current time reference are compared to the 4 values of the new time reference. If any of the 4 comparisions results in a delta greater than +- 1 then this counter gets bumped.

byte 4 = The number of times that new models, received from the FCC, had a delta greater than +- 1 from the prevous models. The count is reset when read by this function. The maximum count of FF is held until the observation is stopped and restarted.

See the comments for function 0x66 regarding the accuracy of this count.

byte 5 = DEF to PBD Communications Status
(Stored in location 2D2, the PBDerror label)

bit 0 = The character the PBD echoed was different than what was sent.
bit 1 = Unexpected characters received from the PBD.
bit 2 = Can't get a prompt from the PBD.
bit 3 = We have timed out while trying to communicate with the PBD.
bit 4 = Spare
bit 5 = One or more re-tries has occurred due to one or more bit 0-2 problems.
bit 6 = One or more re-tries has occurred due to bit 3 problem (timeout)
bit 7 = DEF is currently trying to re-establish communications.
(Bit 7 is cleared if communication is re-established.)

This byte is cleared at the start of an observation. Reading will not clear this byte for Func 61. For Func 69, reading clears bits 0-3.

byte 6 = Spare

SECTION 6

```
* HEADER REQUEST:          HDRRQST =        62 [1156 bytes returned]
                            HEADERS        (30 bytes/hdr * 36 hdrs =    1080)
                            MICRO STATUS   (18 bytes/micro * 4 micros =   72)
                                           (Spare                 =       2)
                                           (New Delay IRQ Time Out Cnt =   2)
                                                                         ------
                                           (Total Bytes Returned =      1156)
```

See hcbDefHdrRead in hcbLoadTestFix.c.

Note it is necessary to do Function Code 61, before doing this function, to see if the data is ready for reading from the DEF. Hence function 62 should only be requested after function 61 returns a 00 for byte 0, indicating headers ready, or 0F for byte 0 indicating the DEF is idle. After the DEF returns a 00 for byte 0 for function 61, it is expected that function 62 will be requested. The DEF will stop reading headers from the TRC until then.

This function will return header and general status information that the TRC micros maintain and the Deformatter integrates. Also returned are two spare bytes and a New Delay IRQ Time Out Count that the Deformatter maintains.

The returned sequence is as follows:

```
  Headers  0 -  7  data tracks from TRC micro 0
  Headers  8 - 15  data tracks from TRC micro 1
  Headers 16 - 23  data tracks from TRC micro 2
  Headers 24 - 31  data tracks from TRC micro 3
  Header  32       system track from micro 0
  Header  33       system track from micro 1
  Header  34       system track from micro 2
  Header  35       system track from micro 3
  General Status from TRC micro 0
  General Status from TRC micro 1
  General Status from TRC micro 2
  General Status from TRC micro 3
  Spare bytes
  New Delay IRQ Time Out Count
```

HEADER DESCRIPTION:
There are 30 bytes for each header returned to the RT system.

Header information obtained from the TRCs is transferred to the Deformatter as follows:

Starting on each 131 ms tic, the Deformatter requests Snap Shot. Then on the following New Delay reads two headers from a TRC. Two headers are read on each New Delay interrupt until all headers are read. The first 18 for each header from the TRC micro is stored as is while the last 6 bytes read for each header from the TRC is integrated by the Deformatter. Thus, the number of bytes in a header that the Deformatter buffers for the RT computer is 30 as compared to the 24 bytes read from the TRC. After transferring all the headers to the RT computer, the Deformatter zeroes its buffer and then starts reading headers from the TRCs on the next 131 ms tic.

SECTION 6

A description of the header bytes follows in 2 pages.

GENERAL STATUS DESCRIPTION:
A description of the General Status follows the header description.

The format of each header is as follows:

BYTE	HD20-23 NOTATION	DESCRIPTION
0		1st aux byte
1		2nd aux byte
2		3rd aux byte
3		4th aux byte
4		5th aux byte
5		6th aux byte
6		7th aux byte
7		8th aux byte
8		1st time byte -----> MJ | YD
9		2nd time byte -----> DS | DD
10		3rd time byte -----> SS | HH
11		4th time byte -----> SS. | MM
12		5th time byte -----> ss | ss.
13		6th time byte -----> ss | ss
14		7th time byte/CRC -> | s(CRC)
	or	1st VLBA CRC byte--> (CRC)|
15		2nd CRC byte
16	E1E2	1st error byte
17		2nd error byte Should be 00A0 all the time
18	Parity-Er	frame parity count, MS byte
19		frame parity count
20		frame parity count
21		frame parity count, LS byte
22	HdrE	number of headers with errors, MS byte
23		number of headers with errors, LS byte
24	Rsyn	number of resyncs, MS byte
25		number of resyncs, LS byte
26	Inva	number of invalid frames, MS byte
27		number of invalid frames, LS byte
28	Frms	number of frames since last header request, MS byte
29		number of frames since last header request, LS byte

(MJD=Modified Julian Day)
Time Format in BCD

VLBA	MKIII
MJ	YD
DS	DD
SS	HH
SS.	MM
ss	ss.
ss	ss
	s(CRC)
(CRC)	

For further details of the header bytes refer to D028TRC.HCB, the protocol specification for the Deformatter CARD to TRACK RECOVERY CARD.

Two byte counters could overflow and wrap around in 2**16 * 2.5 ms/frame = 163.84 seconds. Hence it is necessary to read them at least every 163 seconds.

SECTION 6

TRC General Status Description:

This function will return general status information that the DEF reads from the 4 TRC's. The DEF reads this status information from the TRCs every 131ms. The first 2 status bytes from each of the 4 micros is updated by overwriting the previous value. The DEF integrates all other status information from the TRC until the RTS makes a request. At which time the DEF resets all status information and then proceeds to integrate the information until the RTS makes another request.

Where:
Status from TRC micro 0.
element bytes

0 0 = The mode that the micro is in. Byte decoded as follows:

7 6 5 4	3 2 1 0	<-- bits
0 0 0 0	0 0 x 0	TRC's filter time is not reliable.
0 0 0 0	0 0 x 1	TRC's filter time is reliable.
0 0 0 0	0 0 0 x	TRC's delay was updated.
0 0 0 0	0 0 1 x	TRC's delay was not updated.

1 = not defined at this time.

TimeRst	1	2-3	= The number of times that the filtered time has been reset.
Aborts	2	4-5	= The number of times the header DMA was aborted.
0INV	3	6-7	= The number of frames that this micro's channel 0 has been invalid.
1INV	4	8-9	= The number of frames that this micro's channel 1 has been invalid.
W-Dog	5	10-11	= The number of times the sync watch dog timer timed out.
	6	12-13	= not defined at this time.
	7	14-15	= not defined at this time.
Frms	8	16-17	= Total number of tape frames since last TRC request. That is, the total number of frames the above errors were found in. This is reset every time read.

DEF HD4 NOTATION

Status from TRC micro 1
bytes 18-35= same as above.

Status from TRC micro 2
bytes 36-53= same as above.

Status from TRC micro 3
bytes 54-71= same as above.

NEW DELAY IRQ TIME OUT COUNT (2 Bytes):
The last thing the DEF does in its New Delay IRQ routine is to check to see if there is another New Delay IRQ pending. If there is another IRQ pending then the DEF bumps the New Delay IRQ Time Out Counter. This time out can occur if the TRC micros are slow in responding. This count is zeroed when the RT computer request this function. This occurs when the DEF does not complete its allotted tasks in the 4ms. allowed. Doing this occasionally is considered ok.

SECTION 6

* CALCULATED OFFSET: CALOFFSET = 63 [24 bytes returned]

See hcbPbiCalOfs (buf, busnr, targetnr) in hcbLoadTestFix.c
The four calculated offsets that the Deformattter computes are returned.
These values are 48 bit 2's complement numbers, ms byte first.

Where:
First 6 bytes - CalOffSet0, calculated offset from TRC micro 0's time.
Second 6 bytes - CalOffSet1, calculated offset from TRC micro 1's time.
Third 6 bytes - CalOffSet2, calculated offset from TRC micro 2's time.
Fourth 6 bytes - CalOffSet3, calculated offset from TRC micro 3's time.

(See Appendix 1 for details on how the DEFORMATTER calculates this value).

* TIME REQUEST: TIMERQST = 64 [32 bytes returned]

Where:
bytes 0 - 7 = time from micro 0, formatted as 0M JD 0S SS SS ss ss 0V.
bytes 8 -15 = time from micro 1, formatted as 0M JD 0S SS SS ss ss 0V.
bytes 16 -23 = time from micro 2, formatted as 0M JD 0S SS SS ss ss 0V.
bytes 24 -31 = time from micro 3, formatted as 0M JD 0S SS SS ss ss 0V.

Where:
MJD = Modified Julian Date
0S = MS byte of Seconds.
SS = Seconds
ss = fractional seconds
0V = 00 if the TRCs filtered time is not reliable.
= 01 if the TRCs filtered time is reliable.

* RETURN TIME & CALCULATED OFFSETS: TIMECAL = 65 [56 bytes returned]

This one function performs both funtions 63 and 64.
The first 32 bytes are returned per TIME REQUEST function, 64.
The last 24 bytes are returned per the CALCULATED OFFSET function, 63.

SECTION 6

*RETURN MODEL OUT OF LIMITS STATUS: MODELLIMIT = 66 [6 bytes returned]
See hcbDefMdlStat (busnr, targetnr) in hcbLoadTestFix.c
Bytes 0 and 1 returned by this function are defined the same as bytes 3 and 4 respectively that are returned by Function 61. Separate pairs of counters are maintained so that the reset of the counters that occurs when the two different functions are accessed do not interact. The maximum count of FF is held until reset.

byte 0 = The number of times a received new time reference did not match the linear model tracked by the Deformatter. The count is reset when read by this function.

Each time the Deformatter receives a time reference the following comparision takes place:
For each of the 4 phase centers the current time reference is added to the current delay for that phase center. The same is done with the new time reference and the new delays for each phase center. Then the above 4 values from the current time reference are compared to the 4 values of the new time reference. If any of the 4 comparisions results in a delta greater than +- 1 then this counter gets bumped.

The idle loop job that bumps the counter is job1.asm. This job also prints messages to the console.

byte 1 = The number of times that new models, received from the FCC, had a delta greater than +- 1 from the previous models. The count is reset when read by this function.

The idle loop job that bumps the counter cannot respond fast enough to handle errors every 4 msec. When errors occur every 4 msec, the counter will only be incremented two or three times per second. Thus the counter is only accurate for cases where the error does not occur more than a few times a second.

The idle loop job is job2.asm, and it also prints messages to the console.

bytes 2-5 = A count of the number of times the DEF has lost synchronization with the 131 ms tic. This is a serious error and data should be considered corrupt. This count is reset each time this function is requested. Byte 5 is the lsbyte. The terminal displays the message "Lost New Delay sync!"

* RETURN ONE HEADER: ONEHEADER = 67 TT [30 bytes returned]
See defhdrsngl (hdrnr, busnr, targetnr, flag) in hcbLoadTestFix.c

TT = 0-35 the track to read the header from. Where:
0-7 Tracks 0-7 from TRC micro 0.
8-15 Tracks 0-7 from TRC micro 1.
16-23 Tracks 0-7 from TRC micro 2.
24-31 Tracks 0-7 from TRC micro 3.
32 System track from micro 0.
33 System track from micro 1.
34 System track from micro 2.
35 System track from micro 3.

This function returns the current contents of the DEF header buffer. This function doesn't reset the contents of the buffer.

For details of bytes returned, refer to the format of a header description that is described in function 62. This function can be requested anytime. It is not interlocked with function 61.

SECTION 6

* RETURN TRC GENERAL STATUS: TRCSTAT = 68 [72 bytes returned]

This function returns the current contents of the DEF's buffer of the TRC's general status. This function does not reset the contents. For details of the bytes returned refer to the description under function 62. This function can be requested anytime. It is not interlocked with function 61.

* SPECIAL STATUS REQUEST: SPECSTAT = 69 [7 bytes returned]

See function 61. This function is identical to function 61, except that header reading is not inhibited.

SECTION 7

Test Functions - functions used in testing various parts of the correlator.

* SELF TEST: DEFSELFTEST = 70 TT
See defst (state, bus, targetnr) in hcbLoadTestFix.c.
Where:
TT = 00 Disable 1, 3, 4. Disable all tests.
= 01 Self test on, Deformatter substitutes a test pattern for data. Substitution takes place at output of card. The test pattern is held in SEQ 1 and is the same at all 8 outputs.
= 03 Enable the test patern at the Bit Shuffler Xilinx's.
= 04 Enable the test patern at the Demux-Delay PAL's.

* PBD TEST DATA: PBDTSTDAT = 71 TT

Where:
TT = 00 PBD in normal operation mode. Turns off test mode.
= 01 PBD is enabled to output the test frame generated by the correlator's Master MCC.

Note: The test frame can also be turned on with function 50, but the only way to turn it off is with this function.

* TEST PATTERN DOWNLOAD: DEFTSTPAT0 = 72 [1032 bytes]

Replace the 2 bit test pattern held in Sequencer 1. The sequencer is 2 FFT cyles long. See note with 68k Main Memory Write and Read function for additional information.

The 2 LSbits of each byte of this transfer determines the 2 bit test pattern.
The LSbit = Test Data S. The S test bit provides the test pattern to:
1) Demux-Delay PAL's.
2) Bit Shuffler Xilinx's.
3) The S bit at the ouptut PAL's.
The NLSbit = Test Data M. The M test bit provides the test pattern to:
1) The M bit at the output PAL's.

SECTION 7

* TEST CYCLE: DEFTSTPAT1 = 73

Perform a test cycle. When the Deformatter receives this command it will output a pseudo random test pattern to the FFT engines. The test pattern will begin at a 131ms tic and will last one fringe cycle. The RT system must get this command to the Deformatter at least 8ms before the test cycle is to start.

* SYNC PULSE ENABLE: DEFSYNCENBL = 74

This function is for test/debugging. The 68k code can test this flag to determine if it should output a sync pulse. This sync pulse is TP22 at the top edge of the DEF card. This pulse can be used to check timing, to see if an event ever happens and etc.

* ERROR DISPLAY DISABLE: DEFERRDSPENBL = 75 TT

Display enable/disable control for Time Reference and Model error display as follow:

TT = 00 Enable all error display jobs.
TT = 01 Disable Time Reference error display.
TT = 02 Disable Model error display.
TT = 03 Disable both 01 and 02 above.

For this disable function to take effect, it must be sent after the Deformater receives the start observe function. The Deformater always enable the error displays when it starts observing. The disable function also inhibits the counting of errors for hcb functions 61 and 66.

SECTION 8

Track Recovery Card Communcication Functions
- Communications to the TRC should not take place while in observing. See document D028TRC.HCB for more details and possible exceptions.

* WRITE TO TRC:	WRITETRC =	80 0X CC CC [TRC message]
* READ TRC:	READTRC =	81 0Y CC CC RR RR [TRC message]

Where:
- X, Y = 0, Indicates a TRC target card of TRC 1 micro A
- X, Y = 1, Indicates a TRC target card of TRC 1 micro B
- X, Y = 2, Indicates a TRC target card of TRC 2 micro A
- X, Y = 3, Indicates a TRC target card of TRC 2 micro B
- X = 4, Indicates TRC broadcast mode (to both A micros)
- X = 5, Indicates TRC broadcast mode (to both B micros)
- X = 6, Indicates TRC broadcast mode (to all four micros)

CC CC = is the transfer count to the TRC
RR RR = is the return byte count expected from the TRC

[TRC message] = indicates a communication conforming to the TRB protocol. This protocol is described in the document D028TRC.HCB.

* RESET TRC:	RESETTRC =	82 ZZ

Where ZZ is a dummy byte that the Deformatter must read.

The Deformatter will reset the TRCs and then identify the TRC micro's.

APPENDIX 1

DELAY SETTING IN THE PBI SYSTEM

Controlling the delay in the VLBA correlator for a station is a joint task requiring the coordination of the RTS, the hardware model generator, and the station PBI cards. Before the start of an observation the RTS downloads all static and dynamic parameters to the correlator. When the RTS issues a start observe command these parameters are then made effective and the correlator starts observing.

During the observation the RTS will compute and then transmit to the appropriate HCB targets model update parameters. The model update cycle is 120 seconds. The targets and the parameters they receive each model update cycle during an observaton are as follows:

1) FCC, 4 of these in the system
   a) A set of model parameters. These parameters are used by the hardware model generator that is located on this card. These parameters will be used by this hardware model generator to track the delay model during the next model update cycle. The model generator then supplies its associated Deformatter cards with four delay offsets every New Delay cycle.

2) DEF, 24 of these in the system.
   a) A time reference, a 6-byte fixed point number (TT TT TT TT TT TT), along with the Julian day and flag byte indicating VLBA or MKIII format. See HCB function code 41 for the exact format.
   The time reference parameter (TT TT TT TT TT TT)
   is the number of Nyquist rate samples between the start
   of the current Julian day and the first bit that is to be
   processed by the correlator in the new model update cycle.
   If more than one delay center is being tracked, then the time
   reference parameter will be the number of bits since the
   start of the Julian day for any one of the delay centers.
   The delay models will then provide for the time difference
   between the phase centers. If barrel roll is being used then
   phase center 3 must have the latest delay value in order for the
   barrel roll to track properly.

The observatonal parameters that the Deformatter receives from the RTS are made effective on the 131ms station tic following a "start observe" command. Thus all observational parameters are referenced to the tic that the Deformatter starts on. During an observation updated time references are made effective on the first 131ms tic after the Deformatter receives it.

APPENDIX 1

Both the Deformatter and TRC are interrupted at the start of each delay cycle. The TRC responds to it's interrupt by storing its filtered time code for the last frame header received before the interrupt. The TRC also stores the track buffer RAM address of the first data bit associated with this header.

The TRC will track time in a processing run by looking at the track headers as they come in from the transport. The time code format is either:

00 MJ DS SS SS ss ss for VLBA (MJD is modified Julian Day)

or

YD DD HH MM SS ss s0 for MKIII

where all bytes are packed BCD numbers and where the lower cases indicate fractional seconds. The TRC will use the time codes extracted from frame headers to predict the time codes of subsequent headers. After a few successful predictions, the TRC will maintain it's own time (which can be thought of as a filtered version of the transport playback time stamp). This filtered time, in the format described above, is what the TRC returns to the Deformatter when time is requested. The Deformatter then converts this time format into a binary count of bits since the start of the current Julian day.

The Deformatter responds to it's delay cycle interrupt by requesting the time, as described above, from each of the 4 microprocessors on the two TRCs under its control. The Deformatter converts the fractional-day portion of each time code into a 6-byte number, BB BB BB BB BB BB. This number is the number of binary track bits since the start of the current day. An overflow past a day boundary will be handled by the Deformatter detecting the overflow and adding a day term into its calculations for the remainder of the model update cycle. The next model update cycle will update the time reference parameter thus eliminating the overflow.

Next the Deformatter must calculate four delays, one for each of the four TRC's micros which can track a different phase center. The description that follows is for one calculation. Each calculation provides a delay for one of TRCs micros, and, depending on the mode, a vernier delay used by the Deformatter. The offset that is calculated for TRC micro 3 is used for servoing the PBD. Each of the offsets that are calculated are saved for the RT system to read.

The description that follows is also depicted by figure 1 in appendix 1. (Orcad drawing k044d00.blk)

First the Deformatter adds, to the time reference, a 16-bit delay offset it receives from the model generator every delay cycle. The model delay is the number of bits the actual delay deviates from the linear model that is being tracked by the Deformatter. The delay model correction is then applied as follows:

```
   TT TT TT TT TT TT   where TT is the linear model tracked by the Deformatter.
 + SS SS SS SS MM MM   where MM is the model delay offset and SS is it's sign extension
 ------------------
   RR RR RR RR RR RR   in Nyquist samples
```

APPENDIX 1

The result, RR RR -- RR, is then converted by the Deformatter from Nyquist rate samples to track bits resulting in a 6-byte number, KK KK KK KK KK KK. This conversion requires the RTS to supply the Deformatter with some conversion factors and the pipeline factor. The following table shows how the RR RR RR RR RR RR is converted to KK KK KK KK KK KK.

The TRC can provide delay in track bits only, thus if the sampled bits are spread across tracks then the TRC will provide delay within a track and the DEF will provide the delay across tracks.

Sample Factor	Mode 1-4	Mode 1-2	Mode 1-1	Mode 2-1	Mode 4-1	
1	/4 2	/2 3	nop 1	x2 1	x4 1	<---- operation <---- note
2	/2 3	nop 4	x2 1	x4 1	x8 1	
4	nop 5	x2 4	x4 1	x8 1	x16 1	
8	x2 5	x4 4	x8 1	x16 1	x32 1	
16	x4 5	x8 4	x16 1	x32 1	x64 1	

The operation from the above table results in kk kk kk kk kk kk to which the pipeline factor must then be applied in one of the following ways:

```
Note 1:  kk kk kk kk kk kk
       + SS SS SS SS PP PP  signed extended value of pipeline factor
         ------------------
         KK KK KK KK KK KK
```

APPENDIX 1

Note 2: SS SS SS SS PP PP signed extended value of pipeline factor
r 2 bit remainder from the division

```
SS SS SS SS YY YY
```

SS SS SS SS YY YY divide by 4 = SS SS SS SS QQ QQ + r
the 2 bit remainder, r, is used to set the
delay on the DEF.

```
  kk kk kk kk kk kk
+ SS SS SS SS QQ QQ
  ----------------
  KK KK KK KK KK KK
```

Note 3: SS SS SS SS PP PP signed extended value of pipeline factor
r 1 bit remainder from the division

```
SS SS SS SS YY YY
```

SS SS SS SS YY YY divide by 2 = SS SS SS SS QQ QQ + r
the 1 bit remainder, r, is used to set the
delay on the DEF.

```
  kk kk kk kk kk kk
+ SS SS SS SS QQ QQ
  ----------------
  KK KK KK KK KK KK
```

Note 4: SS SS SS SS PP PP divide by 2 = SS SS SS SS QQ QQ + r
the 1 bit remainder, r, is used to set the
delay on the DEF.

```
  kk kk kk kk kk kk
+ SS SS SS SS QQ QQ
  ----------------
  KK KK KK KK KK KK
```

Note 5: SS SS SS SS PP PP divide by 4 = SS SS SS SS QQ QQ + r
the 2 bit remainder, r, is used to set the
delay on the DEF.

```
  kk kk kk kk kk kk
+ SS SS SS SS QQ QQ
  ----------------
  KK KK KK KK KK KK
```

APPENDIX 1

The KK -- KK number and the BB -- BB number are now in the same units. Thus one substraction will now yield the offset.

```
   KK KK KK KK KK KK  which is the bit we want
 - BB BB BB BB BB BB
 --------------------
      CalOffset
```

Again, KK -- KK is the number of the bit that the delay model requires to be output at the start of the next delay cycle (with the start of the current Julian day as the zero reference to the number) and BB -- BB is the number (with the same zero reference) of the first bit in the tape frame whose header was captured by the last time request.

If the CalOffset is between 0 and -200,000 bits, the Deformatter will instruct the track recovery card to set the delay (the - sign indicates that the desired bit for the delay set has already been written into in the TRC track buffer; a positive number would indicate that the bit has yet to reach the transport read head). If CalOffset is not within the interval above, the bit of interest is not within (or not comfortably within) the TRC buffer and some action must be taken to slew the transport.

The RTS uses function code 63 to read the CalOffset, and will take over the servoing if the CalOffset gets too large. That is around +- 1,000,000 bits. It should command the tape drive so that the CalOffset is within the 0 to -200000 bits which is within the data buffer range of the TRC.

The CalOffset that the DEF saves to be returned to the RTS is the one calculated during the first 4 ms of the current 131 ms tic interval. The values in the DEF are only updated in that first 4 ms interval. That means that if for some reason the RTS requests the CalOffset two or more times within the 131 ms interval, the same values will be returned.

When CalOffset is within the interval indicated above and hence the first bit to be output by the TRC at the start of the next delay cycle is in the TRC buffer, the Deformatter will convert CalOffset into a command for the TRC of the form:

8V DD DD

where 8V is the function code/vernier in the Deformatter-TRC protocol, DD DD is the CalOffset divided by 4, and V is the TRC vernier delay (the divide by 4 operation is needed since the 64K x 4 track buffer on the TRC is addressed 4 bits at a time and hence delayed 4 bits at a time). The V term is the remainder of the divide by 4.

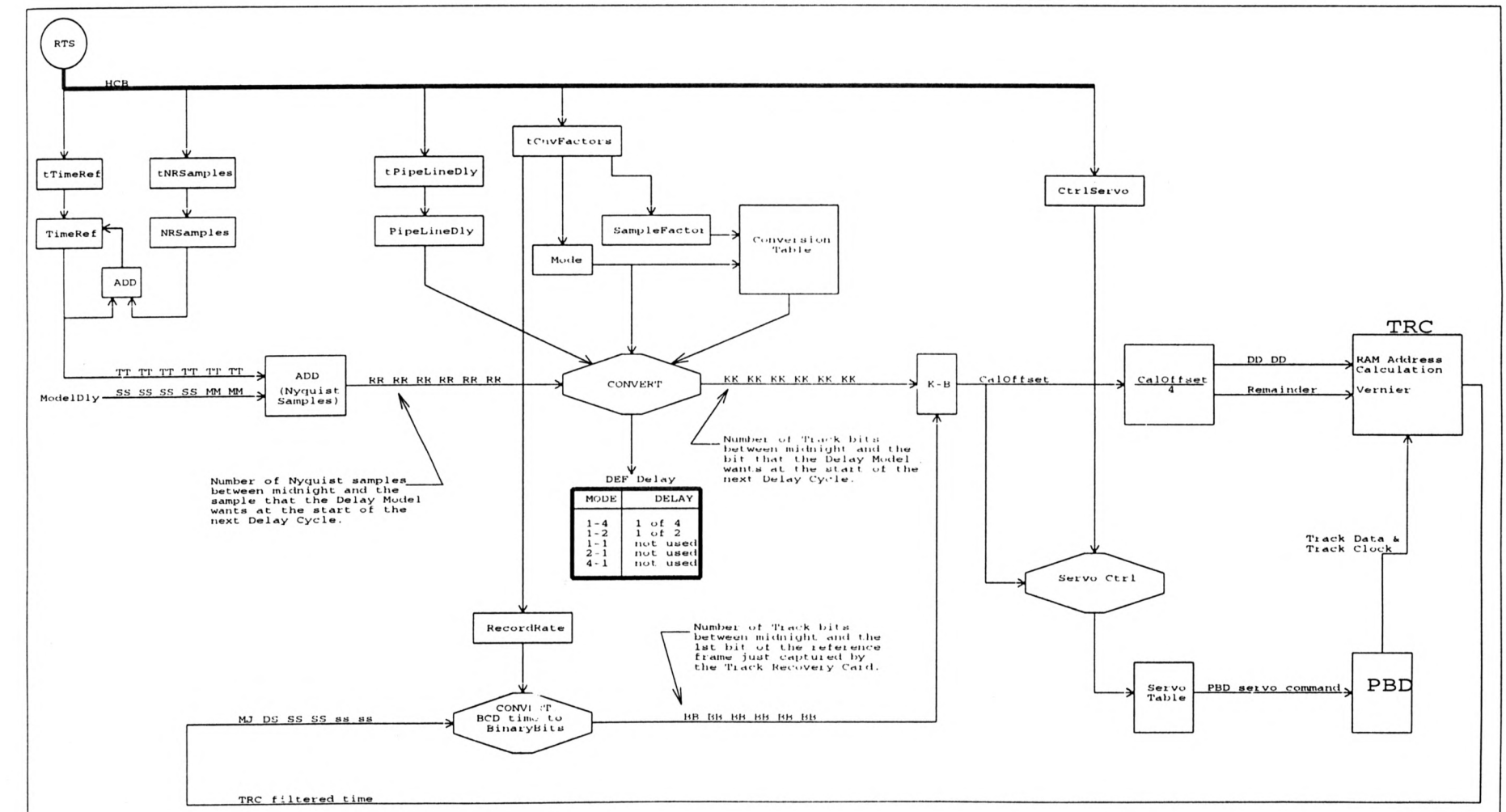
The TRC adds the DD DD value to its stored track buffer RAM address to find the ram address that will start the next delay cycle. (The TRC will compensate for track headers that are also written into its track buffer RAM).

APPENDIX 1

After doing the above calculations for each of the four phase centers (the four TRC micros) the Deformatter updates the time reference by calculating a linear model. This is done by:

```
   TT TT TT TT TT TT
 + 00 00 00 0N NN NN
-----------------------
   TT TT TT TT TT TT
```

where N NN NN is an observational parameter supplied by the RT system to the Deformatter and is equal to the number of Nyquist rate samples in one delay cycle.



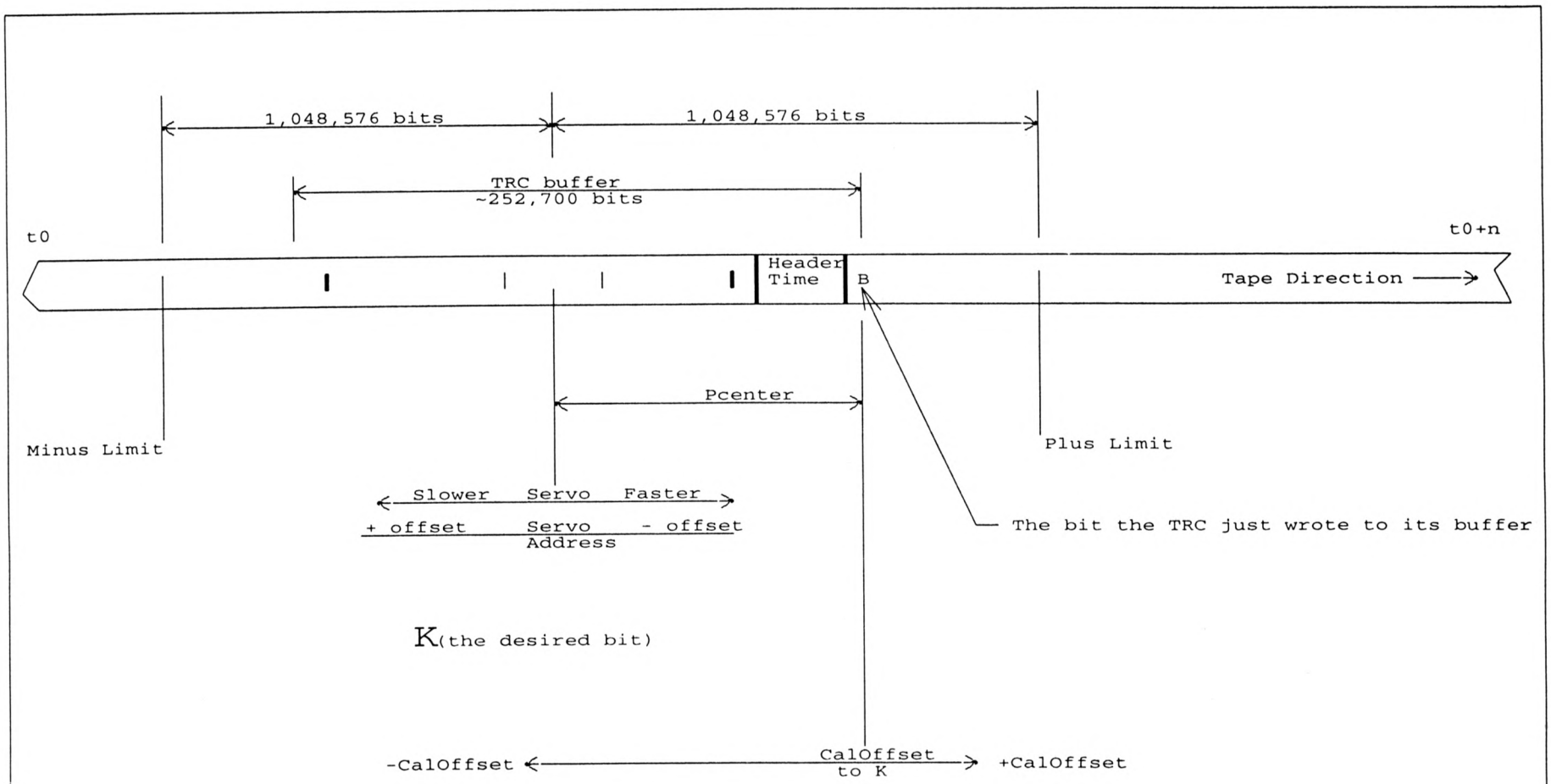
VLBA CORRELATOR PROJECT

NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title

PBI DELAY CALCULATION

Docu[redacted]umber



VLBA CORRELATOR PROJECT NATIONAL RADIO ASTRONOMY OBSERVATORY CHARLOTTESVILLE, VA	
Title TRC BUFFER - TAPE - CALOFFSET	
Size: A	Document Number: 9TapeSer56000K045 (K045D00.BL) REV
Date: June 23, 1992	Sheet 1 of

APPENDIX 1

FIGURE 2

APPENDIX 2

The Deformatter sequencer file names follow the naming convention described below:

s#nb-tf.ext

Where:

s - Fixed, indicates a Sequencer file.

\# - The sequencer number, 1, 2, or 3. Which of the 3 sequencers on the Deformatter that this file is for. The general function of the 3 sequencers are as follows:

Sequencer 1 is 2 FFT cyles in length and clk is at 32mHz.
It's functions are:
1) Generate the 8 & 16mHz clks.
2) Generate FFT signals
3) Generate New Fringe signals
4) Generate New Delay signals
5) Generate test pattern
6) Generate a window for flagging data invalid.

Sequencer 2 is 2k states in length and clk is at 32mHz. This sequencer is independent of FFT size. It's functions are:
1) Generate the Read/Write timing for the Buffer RAM.
2) Generate 2 timing signals required by the Bit Shufflers.
3) Generate a clk enable for the RAM write address counter.

Sequencer 3 is 256 states in length and clk is once per FFT cycle.
It's functions are as follows:
1) Generates an 8 bit count that is loaded in the Capture counter that is used for overlapping.
2) Generates a New Delay Enable that determines the delta between the New Delay and New Fringe. This delta is what determines the seperation of the Write & Read address pointers for the buffer RAM.

n - indicates normal or oversampled data as follows:

n= n for Normal
An "N" will be used if the sequencer is not affected by oversampling.
n= v for oversampled by a factor of 2.
n= w for oversampled by a factor of 4.
n= x for oversampled by a factor of 8.
n= y for oversampled by a factor of 16.

APPENDIX 2

b-t - These 3 characters represent the number of Bitstreams to Tracks.
Some examples would be:
1-4 1 bitstream on 4 tracks
1-1 1 bitstream on 1 track
2-1 2 bitstreams on 1 track

f - Size of the FFT represented as follows:
f= b= 64 point FFT
f= c= 128 point FFT
f= d= 256 point FFT
f= e= 512 point FFT
f= f= 1024 point FFT
f= g= 2048 point FFT
f= i= this sequencer is independent of the FFT size.
For small ffts, sequencers 1 and 2 use the same sequencer files as the corresponding 512 point fft. That is for s1nb-tf or s2nb-tf, if f equals b, c, or d, use s1nb-te or s2nb-te respectively. Links are provided by the defseq make file from the small fft names to the 512 (e) point names. Links are also provided from some generic hex files that are used for multiple modes. See defseq/tables.txt.

ext - The file extension which can be as follows:
hex - the file in Motorola hex format that the RTS downloads to the Deformatter.
dat - The source file that the hex file is generated from. This file is a text file which gets processed by the program genbits.c in the defseq component. genbits generates a .hex sequencer file directly. genbits in invoked by a make in the defseq component. See the file /defseq/tables.txt for a detailed description of the .dat files.

Test Fixture script file naming convention:

ldpbitfn.alB
- FFT size B=64,C=128,D=256,E=512,F=1K,G=2K
- Row from chart below
- Column from Chart below
- n for normal, v for overlap

	FFT SIZE	A: 1 BITSTREAM TO 4 TRACKS	B: 1 BITSTREAM TO 2 TRACKS	C: 1 BITSTREAM TO 1 TRACK	D: 2 BITSTREAMS TO 1 TRACK	E: 4 BITSTREAMS TO 1 TRACK
1	512 or Smaller	n1-4e DEFCTL.MCS BSP-2.MCS Data Valid Delay = 04 Barrel Roll Delay = Pipeline Delay = ff c7 **0**	n1-2e DEFCTL.MCS BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff e3 **1**	n1-1e DEFCTL.MCS BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f1 **2**	n2-1e DEFCTL.MCS BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f2 **3**	n4-1e DEFCTL.MCS BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f0 **4**
2	OVERSAMPLED 512 or Smaller	v1-4e | w1-4e | x1-4e y1-4e DEFCTL.MCS | BSP-1 | BSP-6 BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff c7 **5**	v1-2e | w1-2e x1-2e y1-2e DEFCTL.MCS | BSP-6 BSP-3.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff e5 **6**	v1-1e DEFCTL.MCS BSP-6.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f6 **7**	v2-1e DEFCTL.MCS BSP-6.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff fb **8**	v4-1e DEFCTL.MCS BSP-6.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f6 **9**
3	1K	n1-4f DEFCTL.MCS BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff c7 **10**	n1-2f DEFCTL.MCS BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff e3 **11**	n1-2f DEFCTL.MCS BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f1 **12**	n2-1f DEFCTL.MCS BSP-6.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f1 **13**	n4-1f DEFCTL.MCS BSP-6.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f1 **14**
4	OVERSAMPLED 1K	v1-4f | w1-4f | x1-4f y1-4f DEFCTL.MCS | BSP-9 | BSP-7 BSP-3.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff ef **15**	v1-2f | w1-2f x1-2f y1-2f DEFCTL.MCS | BSP-7 BSP-4.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff fb **16**	v1-1f DEFCTL.MCS BSP-7.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f1 **17**	v2-1f DEFCTL.MCS BSP-7.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f9 **18**	v4-1f DEFCTL.MCS BSP-7.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff fd **19**
5	2k	n1-4g DEFCTL.MCS BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff cb **20**	n1-2g DEFCTL.MCS BSP-2.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff fb **21**	n1-1g DEFCTL.MCS BSP-7.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f3 **22**	n2-1g DEFCTL.MCS BSP-7.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f1 **23**	n4-1g DEFCTL.MCS BSP-7.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f7 **24**
6	OVERSAMPLED 2K	v1-4g | w1-4g | x1-4g y1-4g DEFCTL.MCS | BSP-0 | BSP-8 BSP-4.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f7 **25**	v1-2g | w1-2g x1-2g y1-2g DEFCTL.MCS | BSP-8 BSP-5.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f3 **26**	v1-1g DEFCTL.MCS BSP-8.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f1 **27**	v2-1g DEFCTL.MCS BSP-8.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f0 **28**	v4-1g DEFCTL.MCS BSP-8.MCS Data Valid Delay = Barrel Roll Delay = Pipeline Delay = ff f0 **29**

The numbers 0-29 in the lower right corner of each block are the indices into the table used by the system software.

Explanation of block contents:

DEFCTL.MCS	Xilinx personalities for the validity and barrel roll Xilinxs. Same for all modes
BSP-2.MCS	Personality to be fed to the Bit Shuffler Xilinx, where # is the version number.
Data Valid Delay = 04	Data Valid Delay to the counter in the data valid Xilinx
Barrel Roll Delay = 0x0101	Barrel Roll Delay. A value to the counter in the barrel roll Xilinx. Probably mode independent.
n1-4e	See Appendix 2 text for sequencer file name descriptions
Pipeline Delay = ff c7	A delay number to allow the bit to get to the FFT card at the correct time, based on the number of pipeline stages involved. See hcbSysTestFix.c (in the system software) for the complete table of pipedelay_list1[][][][] values for all OVS factors

Extra dimensions to this chart are factors by which the oversampling is > 2 (4,8 and 16)

Some extra sequencer files may be required for fft sizes less than 512.

The OVERSAMPLED categories in this table are for OVS Factor = 2
(v=2 w=4 x=8 y=16)

1-4 and 1-2, when ovs > 2, are special cases.
The small blocks at indices 5, 6, 15, 16, 25 and 26 specify the BSP to use in each of the special cases.

Figure 2, Appendix 2, K006D01A.TBL is an expanded version of this table with more details about the special cases.

LOCATED IN THE DEFASM/PROTOCOL COMPONENT

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title: MODE TABLE

Size	Document Number	REV
B	5600DK006 (K006D01.TBL)	

Date: Sep 18, Sheet of

FFT SIZES

BIT TO TRACK MODES	64 (B)					128 (C)					256 (D)					512 (E)					1K (F)					2K (G)				
	OVERSAMPLE FACTOR					OVERSAMPLE FACTOR					OVERSAMPLE FACTOR					OVERSAMPLE FACTOR					OVERSAMPLE FACTOR					OVERSAMPLE FACTOR				
	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16
	N	V	W	X	Y	N	V	W	X	Y	N	V	W	X	Y	N	V	W	X	Y	N	V	W	X	Y	N	V	W	X	Y
1 TO 4	N1-4B (OK) BSP-2 FFC7 1	V1-4B (OK) BSP-2 FFC7 1	W1-4B (V1-2B) OK BSP-1 (NEW) FFE5 2	X1-4B (V1-1B) OK BSP-6 FFF6 4	Y1-4B (W1-1B) OK BSP-6 FFFA 8	N1-4C (OK) BSP-2 FFC7 1	V1-4C (OK) BSP-2 FFC7 1	W1-4C (V1-2C) OK BSP-1 (NEW) FFE5 2	X1-4C (V1-1C) OK BSP-6 FFF6 4	Y1-4C (W1-1C) OK BSP-6 FFFA 8	N1-4D (OK) BSP-2 FFC7 1	V1-4D (OK) BSP-2 FFC7 1	W1-4D (V1-2D) OK BSP-1 (NEW) FFE5 2	X1-4D (V1-1D) OK BSP-6 FFF6 4	Y1-4D (W1-1D) OK BSP-6 FFFA 8	N1-4E OK BSP-2 FFC7 0 1	V1-4E OK BSP-2 FFC7 5 1	W1-4E (V1-2E) OK BSP-1 (NEW) FFE5 2	X1-4E (V1-1E) OK BSP-6 FFF6 4	Y1-4E (W1-1E) OK BSP-6 FFFA 8	N1-4F OK BSP-2 FFC7 10 2	V1-4F OK BSP-3 FFEF 15 2	W1-4F (V1-2F) OK BSP-9 (NEW) FFFB 4	X1-4F (V1-1F) OK BSP-7 FFF1 8	Y1-4F (W1-1F) OK BSP-7 FFF8 16	N1-4G OK BSP-2 FFCB 20 4	V1-4G OK BSP-4 FFF7 25 4	W1-4G (V1-2G) OK BSP-0 (NEW) FFF3 8	X1-4G (V1-1G) OK BSP-8 FFF1 16	Y1-4G (W1-1G) OK BSP-8 FFD1 32
1 TO 2	N1-2B OK BSP-2 FFE3 2	V1-2B OK BSP-3 FFE5 2	W1-2B (V1-1B) OK BSP-6 FFF6 4	X1-2B (W1-1B) OK BSP-6 FFFA 8	Y1-2B (X1-1B) OK BSP-6 FFFA 16	N1-2C OK BSP-2 FFE3 2	V1-2C OK BSP-3 FFE5 2	W1-2C (V1-1C) OK BSP-6 FFF6 4	X1-2C (W1-1C) OK BSP-6 FFFA 8	Y1-2C (X1-1C) OK BSP-6 FFFA 16	N1-2D OK BSP-2 FFE3 2	V1-2D OK BSP-3 FFE5 2	W1-2D (V1-1D) OK BSP-6 FFF6 4	X1-2D (W1-1D) OK BSP-6 FFFA 8	Y1-2D (X1-1D) OK BSP-6 FFFA 16	N1-2E OK BSP-2 FFE3 1 2	V1-2E OK BSP-3 FFE5 6 2	W1-2E (V1-1E) OK BSP-6 FFF6 4	X1-2E (W1-1E) OK BSP-6 FFFA 8	Y1-2E (X1-1E) OK BSP-6 FFFA 16	N1-2F OK BSP-2 FFE3 11 4	V1-2F OK BSP-4 FFFB 16 4	W1-2F (V1-1F) OK BSP-7 FFF1 8	X1-2F (W1-1F) OK BSP-7 FFF8 16	Y1-2F (X1-1F) OK BSP-7 FFF0 32	N1-2G OK BSP-2 FFFB 21 8	V1-2G OK BSP-5 FFF3 26 8	W1-2G (V1-1G) OK BSP-8 FFF1 16	X1-2G (W1-1G) OK BSP-8 FFD1 32	Y1-2G (X1-1G) OK BSP-8 FFD1 64
1 TO 1	N1-1B OK BSP-2 FFF1 4	V1-1B OK BSP-6 FFF6 4	W1-1B OK BSP-6 FFFA 8	X1-1B OK BSP-6 FFFA 16	Y1-1B OK BSP-6 FFF0 32	N1-1C OK BSP-2 FFF1 4	V1-1C OK BSP-6 FFF6 4	W1-1C OK BSP-6 FFFA 8	X1-1C OK BSP-6 FFFA 16	Y1-1C OK BSP-6 FFF0 32	N1-1D OK BSP-2 FFF1 4	V1-1D OK BSP-6 FFF6 4	W1-1D OK BSP-6 FFFA 8	X1-1D OK BSP-6 FFFA 16	Y1-1D OK BSP-6 FFF0 32	N1-1E OK BSP-2 FFF1 2 4	V1-1E OK BSP-6 FFF6 7 4	W1-1E OK BSP-6 FFFA 8	X1-1E OK BSP-6 FFFA 16	Y1-1E OK BSP-6 FFF0 32	N1-1F OK BSP-2 FFF1 12 8	V1-1F OK BSP-7 FFF1 17 8	W1-1F OK BSP-7 FFF8 16	X1-1F OK BSP-7 FFF0 32	Y1-1F OK BSP-7 FFF0 64	N1-1G OK BSP-7 FFF3 22 16	V1-1G OK BSP-8 FFF1 27 16	W1-1G OK BSP-8 FFD1 32	X1-1G OK BSP-8 FFD1 64	Y1-1G NOT VALID MODE
2 TO 1	N2-1B BSP-2 FFF2 8	V2-1B OK BSP-6 FFFB 8	W2-1B BSP-6 FFF8 16	X2-1B BSP-6 FFF6 32	Y2-1B NOT VALID MODE	N2-1C BSP-2 FFF2 8	V2-1C BSP-6 FFFB 8	W2-1C BSP-6 FFF8 16	X2-1C BSP-6 FFF6 32	Y2-1C BSP-6 FFF0 64	N2-1D OK BSP-2 FFF2 8	V2-1D BSP-6 FFFB 8	W2-1D BSP-6 FFF8 16	X2-1D BSP-6 FFF6 32	Y2-1D BSP-6 FFF0 64	N2-1E OK BSP-2 FFF2 3 8	V2-1E OK BSP-6 FFFB 8 8	W2-1E OK BSP-6 FFF8 16	X2-1E OK BSP-6 FFF6 32	Y2-1E OK BSP-6 FFF0 64	N2-1F OK BSP-6 FFF1 13 16	V2-1F OK BSP-7 FFF9 18 16	W2-1F OK BSP-7 FFF1 32	X2-1F OK BSP-7 FFF0 64	Y2-1F NOT VALID MODE	N2-1G OK BSP-7 FFF1 23 32	V2-1G OK BSP-8 FFF0 28 32	W2-1G OK BSP-8 FFEC 64	X2-1G NOT VALID MODE	Y2-1G NOT VALID MODE
4 TO 1	N4-1B BSP-2 FFF0 16	V4-1B BSP-6 FFF6 16	W4-1B BSP-6 FFFA 32	X4-1B NOT VALID MODE	Y4-1B NOT VALID MODE	N4-1C BSP-2 FFF0 16	V4-1C BSP-6 FFF6 16	W4-1C BSP-6 FFFA 32	X4-1C BSP-6 FFF6 64	Y4-1C NOT VALID MODE	N4-1D OK BSP-2 FFF0 16	V4-1D BSP-6 FFF6 16	W4-1D BSP-6 FFFA 32	X4-1D BSP-6 FFF6 64	Y4-1D NOT VALID MODE	N4-1E OK BSP-2 FFF0 4 16	V4-1E OK BSP-6 FFF6 9 16	W4-1E OK BSP-6 FFFA 32	X4-1E OK BSP-6 FFF6 64	Y4-1E NOT VALID MODE	N4-1F OK BSP-6 FFF1 14 32	V4-1F OK BSP-7 FFFD 19 32	W4-1F OK BSP-7 FFF0 64	X4-1F NOT VALID MODE	Y4-1F NOT VALID MODE	N4-1G OK BSP-7 FFF7 24 64	V4-1G OK BSP-8 FFF0 29 64	W4-1G NOT VALID MODE	X4-1G NOT VALID MODE	Y4-1G NOT VALID MODE

(*)

OVERLAP FACTOR

BIT SHUF

PIPELINE DELAY

MODE NAME

THE NUMBERS 0-29 IN THIS POSITION ARE THE INDICES INTO THE MODE TABLE USED BY THE SOFTWARE.

THERE ARE SPECIAL CASES FOR 1-4 AND 1-2 WHEN OVS > 2. THESE ARE AT INDICES 5, 6, 15, 16, 25 AND 26.

THE SPECIAL CASES ARE ONES WHERE THE NORMAL BSP FOR OVS2 DOES NOT APPLY WHEN OVS > 2.

SEE APPENDIX 2, FIGURE 1 WHERE THE SPECIAL CASES STAND OUT.

NOTE THAT THE INDICES EXIST ONLY FOR THE 512, 1K, 2K CASES WHERE OVS=1 OR OVS=2. EXCEPT FOR THE SPECIAL CASES NOTED, FFT SIZE < 512 IS SAME AS 512 AND OVS > 2 IS SAME AS OVS = 2

OK — INDICATES MODE CHECKOUT HAS BEEN DONE

(OK) — INDICATES MODE IS IDENTICAL TO ANOTHER MODE IN THE PBI THAT CHECKED OK.

1-TO-4 MODES, WITH OVS 1 AND OVS 2 FOR FFT SIZES < 512 ARE IDENTICAL TO THE SAME MODE WITH FFTSIZE = 512

THESE LINES ARE USED TO TRACE FROM A "ACTUAL MODE" TO A "SUBSTITUTE MODE"

(*) = SUBSTITUTE MODE

SUBSTITUTE MODES ARE USED FOR ALL CASES WHERE OVERSAMPLED BITS ARE DISCARDED BY "IGNORING" TRACKS. THIS OCCURS IN ALL 1-TO-4 AND 1-TO-2 MODES WHERE OVS > 2.

THE SUBSTITUTE MODE IS DETERMINED BY STARTING FROM THE ACTUAL MODE AND STEPPING DIAGONALLY DOWN AND TO THE LEFT UNTIL EITHER THE OVERSAMPLE FACTOR IS 2, OR THE BIT-TO-TRACK MODE IS 1-TO-1

FOR EACH CASE WHERE THE SUBSTITUTE MODE IS 1-TO-1 USE THE SEQUENCERS, BSP, CONVERSION FACTORS AND PIPELINE DELAY ASSOCIATED WITH THE SUBSTITUTE MODE.

FOR EACH CASE WHERE THE SUBSTITUTE MODE IS 1-TO-2 USE THE SEQUENCERS, CONVERSION FACTORS AND PIPELINE DELAY ASSOCIATED WITH THE SUBSTITUTE MODE. NEW BSP WERE REQUIRED FOR THESE MODES, AS INDICATED IN THE TABLE.

THE NEW BSP ARE: BSP-0 BSP-1 BSP-9:

BSP-0 WAS DERIVED FROM BSP-5, WITH THE EVEN DELAY CENTER INPUTS ROUTED TO ALL MULTIPLEXORS.

BSP-1 WAS DERIVED FROM BSP-3, WITH THE EVEN DELAY CENTER INPUTS ROUTED TO THE LOWER (0-3) INPUTS OF ALL MULTIPLEXORS, AND THE ODD DELAY CENTER INPUTS ROUTED TO THE UPPER (4-7) INPUTS OF ALL MULTIPLEXORS.

BSP-9 WAS DERIVED FROM BSP-4, WITH THE EVEN DELAY CENTER INPUTS ROUTED TO ALL MULTIPLEXORS.

NEW BIT SHUFFLER PROGRAM WORD FUNCTIONS ARE REQUIRED FOR THE EXISTING BIT SHUFFLER PERSONALITIES. SEE THE NOTES FOR "BSP PROGRAM WORDS"

NOT VALID MODE:

THESE MODES ARE NOT POSSIBLE.

THE REASON HAS SOMETHING TO DO WITH SEQUENCER 3, AND THE CAPTURE COUNT AND THE NEW_DELAY_ENABLE.

SEE TABLES.TXT IN THE DEFSEQ COMPONENT IF YOU NEED TO TRY TO UNDERSTAND THE LIMITATION. (GOOD LUCK.)

BSP PROGRAM WORDS:

BEFORE THE INCLUSION OF SUPPORT FOR THE MODES WHERE SUBSTITUTE MODES ARE INVOLVED, OVS > 2 HAD NO EFFECT ON BSP PROGRAM WORDS. FOR EXAMPLE, THE BSP PROGRAM WORD FUNCTION FOR V1-1E APPLIED TO W1-1E, X1-1E AND Y1-1E.

THIS IS NOT THE CASE FOR THE 1-4, AND 1-2 MODES WHERE OVS > 2. FOR THESE MODES, THE OFFSETS IN THE EQUATIONS FOR THE SUBSTITUTE MODES NEED FACTORS OF 2 OR 4.

IF THE SUBSTITUTE MODE IS ONE LINE LOWER THAN THE ACTUAL MODE IN THE ABOVE TABLE, THE FACTOR IS 2. iF THE SUBSTITUTE MODE IS TWO LINES LOWER, THEN THE FACTOR IS 4.

FOR EXAMPLE, LOOK AT X1-4E, AND W1-2E:

BOTH OF THESE MODES USE THE SUBSTITUTE MODE V1-1E.

IN V1-1E, MUX3 = INPUT0 + SAMPBITS

FOR W1-2E, THE EQUATION NEEDS TO BE:

MUX3 = INPUT0 + SAMPBITS*2

FOR X1-4E, THE EQUATION NEEDS TO BE:

MUX3 = INPUT0 + SAMPBITS*4

THE NEW FACTORS ACCOUNT FOR THE FACT THAT THE OFFSET FROM THE SIGN TRACKS(S) TO THE MAG TRACK(S) IS DIFFERENT FROM THAT OF THE SUBSTITUTE MODES DUE TO THE INTERMEDIATE TRACKS THAT ARE DISCARDED.

FOR THE THREE NEW BSP, THE BIT FIELD STRUCTURES FOR THE NEW BSP ARE IDENTICAL TO THOSE FROM WHICH EACH WAS DERIVED:

STRUCT BSP0 = STRUCT BSP5
STRUCT BSP1 = STRUCT BSP3
STRUCT BSP9 = STRUCT BSP4

THE EQUATIONS FOR EACH ARE:

BSP-0 FOR W1-4G:

B.MUX4 = (INPUT0) / 2

B.MUX5 = B.MUX4 + 1

B.MUX6 = B.MUX4 + SAMPBITS*2

B.MUX7 = B.MUX6 + 1

B.CB24 = MICRO % 2

WHERE THE ONLY VALID VALUES FOR INPUT0 ARE DELAY CENTER INPUTS 0 AND/OR 4

BSP-1 FOR W1-4E, W1-4D, W1-4C AND W1-4B:

B.MUX0 = B.MUX1 = (INPUT0) / 2

B.MUX2 = B.MUX3 = (INPUT0 + SAMPBITS*4)/2

B.MUX4 = B.MUX5 = (INPUT1) / 2

B.MUX6 = B.MUX7 = (INPUT1 + SAMPBITS*4)/2

B.CB24 = 1

B.CB25 = 0

WHERE THE ONLY VALID VALUES FOR INPUT0 AND INPUT1 ARE DELAY CENTER INPUTS 0 AND/OR 4

BSP-9 FOR W1-4F:

B.MUX4 = (INPUT0) / 2

B.MUX5 = B.MUX4 + 1

B.MUX6 = B.MUX4 + SAMPBITS*2

B.MUX7 = B.MUX6 + 1

WHERE THE ONLY VALID VALUES FOR INPUT0 ARE DELAY CENTER INPUTS 0 AND/OR 4

PIPELINE DELAYS:

VALUES ARE FILLED IN AFTER THE MODE CHECKOUT HAS BEEN DONE, OR FOR THE 1-TO-4 AND 1-TO-2 MODES WHERE OVS > 2 THE VALUE DETERMINED FOR THE SUBSTITUTE MODE IS FILLED IN WHEN AVAILABLE (EVEN THOUGH THE MODE CHECKOUT MAY NOT HAVE BEEN DONE YET)

SEQUENCER FILE NAMES:

MANY OF THE HEX FILE NAMES FOR THE THREE SEQUENCERS REQUIRED FOR EACH MODE ARE DERIVED FROM ITEMS IN THE ABOVE TABLE AS FOLLOWS:

SEQUENCER NUMBER:	s1, s2 or s3
OVERSAMPLE MODE:	n,v,w,x or y
BIT-TRACK MODE	1-4, 1-2, 1-1, 2-1 or 4-1
FFT SIZE:	b,c,d,e,f or g

EXAMPLE:

s1n1-4g.hex
s2n1-4g.hex
s3n1-4g.hex

THESE ARE THE THREE SEQUENCER FILES FOR OVS1, 1-TO-4, 2k

MANY MODES USE IDENTICAL SEQUENCER FILES. THUS EVERY FILENAME DERIVED ABOVE WILL NOT BE ACCOMPANIED BY AN ACTUAL FILE. THE FOLLOWING FUNCTION WILL LOAD THE CORRECT SET OF THREE SEQUENCER FILES IN ALL CASES:

```
defWrSequencer (pbi, fft_size, over_sample_mode, fan_in_out_mode)

  where pbi is 0-23,
  fft_size is 64, 128, 256, 512, 1024 or 2048
  over_sample_mode is 1, 2, 4, 8 or 16
  fan_in_out_mode is same as fmt_mode_enum in tables.h
  (i.e. 0 for 4-1, 1 for 2-1, 2 for 1-1, 3 for 1-2 and 4 for 1-4)
```

APPENDIX 2 | FIGURE 2

Title: EXPANDED MODE TABLE		
Size: C	Document Number: K006D01A.TBL	REV
Date: June 21, 1996	Sheet: of	

Old Roll				New Roll			
R3	R2	R1	R0	R3	R2	R1	R0
7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1

adr= 142

NEW FRAME DELAY

NC	NC	NC	NC	G11	G10	G9	G8	G7	G6	G5	G4	G3	G2	G1	G0
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1

P0	P1
HCB FNCT 37	

adr= 141

adr= 140

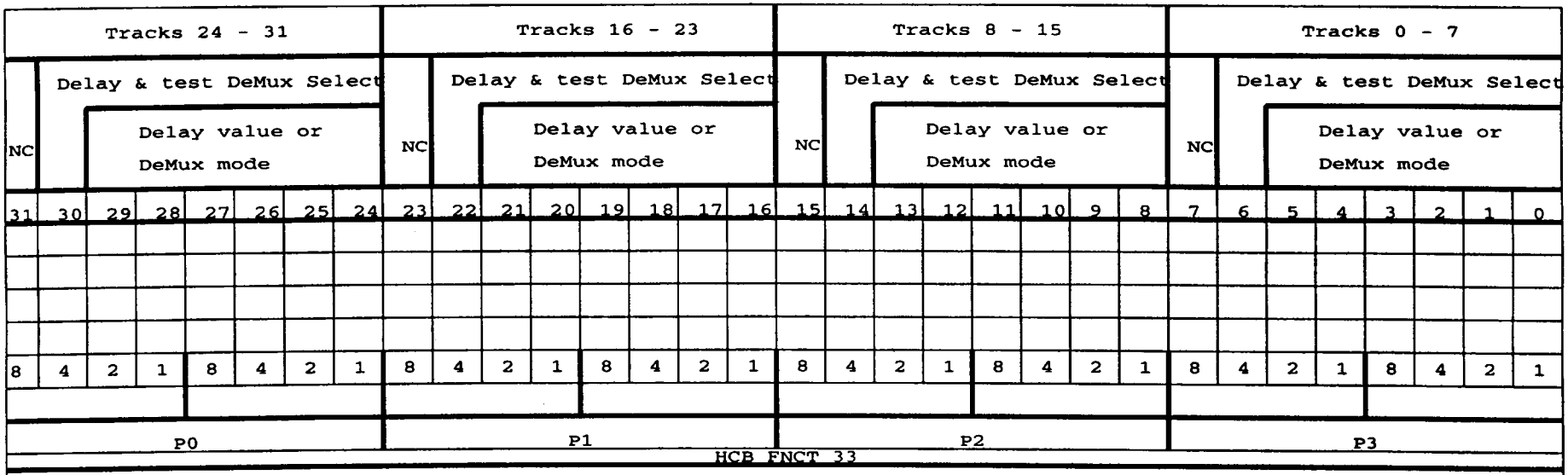
Label= B_Roll
adr 140

NOTES:
1. NC = No Connection, this bit not used.

VLBA CORRELATOR PROJECT NATIONAL RADIO ASTRONOMY OBSERVATORY CHARLOTTESVILLE, VA		
Title CONTROL REGISTER IN BARREL ROLL XILINX		
Size A	Document Number 56000D029 (D029D01.TBL)	REV
Date: April 30, 1993	Sheet 1 of 5	

APPENDIX 3

FIGURE 1



Label= DemDel
address = 11C

NOTES:
1. NC = No Connection, this bit not used.
2. When in Delay mode the RTS should initialize P1-P4 to 0. The Deformatter will set the delay values per calculated offset.
3. Bit 6 (test) of each of the Px parameters of HCB fnct 33 can also be controlled by HCB fnct code 70.

Delay Values & Test test mode:

40 = output test data on the 4 outputs

00 = pass thru
41 = 1 bit delay. 1 bitstream on 4 tracks.
42 = 2 bit delay. 1 bitstream on 4 tracks.
43 = 3 bit delay. 1 bitstream on 4 tracks.

51 = 1 bit delay. 1 bitstream on 2 tracks.

DeMux modes:

00 = pass thru

01 = Tracks 0 & 2 in. 2 bitstreams on 1 track.
36 = Tracks 1 & 3 in. 2 bitstreams on 1 track.

03 = Track 0 in. 4 bitstreams on 1 track.
37 = Track 1 in. 4 bitstreams on 1 track.
35 = Track 2 in. 4 bitstreams on 1 track.
3F = Track 3 in. 4 bitstreams on 1 track.

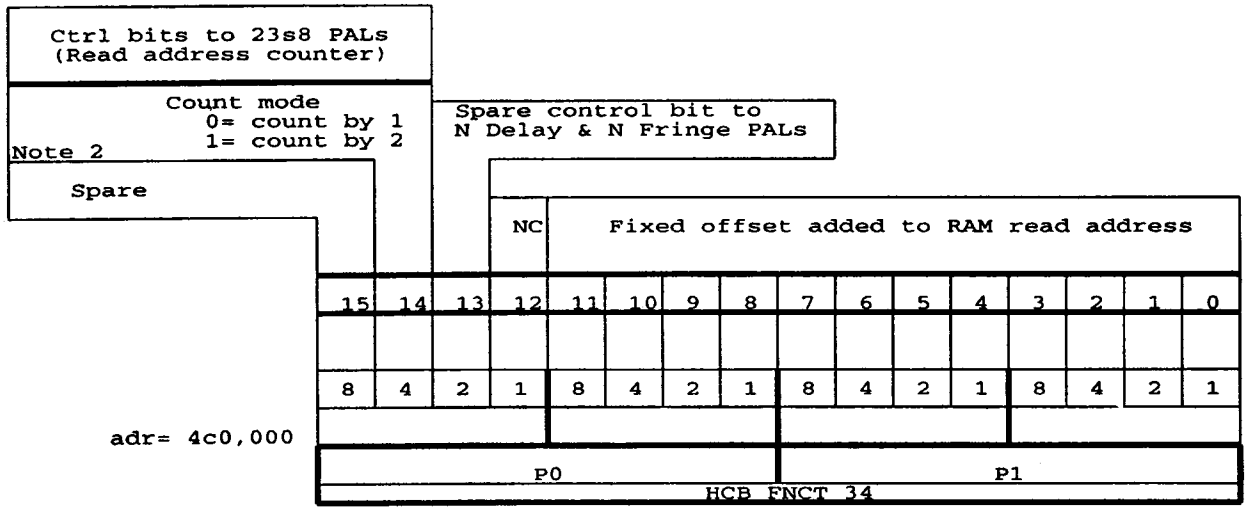
VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title: DEMUX-DELAY CONTROL WORD

Size	Document Number	REV
A	56000D023 (D029D02.TBL)	

Date: June 10, 1995 Sheet 2 of 5

APPENDIX 3 FIGURE 2



FIXED OFFSET FOR RAM READ OFFSET REGISTER HAS NEVER BEEN USED (IT IS ALWAYS 0).

IT WAS INTENDED TO BE USED FOR OVERLAPPED FFTs IN ADJACENT PIPELINES.

NOTES:
1. NC = No Connection, this bit not used.
2. If oersample then count by 2 else count by 1.

VLBA CORRELATOR PROJECTVLBA CORRELATOR PROJE NATIONAL RADIO ASTRONOMY OBSERVATORYNATIONAL CHARLOTTESVILLE, VACHARLOTTESVILLE, VA		
Title OVERLAP & CNT BY 1or2 CTRL REGISTER		
Size A	Document Number 56000D029 (D029D03.TBL)	REV
Date: October 29, 1996	Sheet 3 of 5	

APPENDIX 3

FIGURE 3

Where:
I0 = Channel 0
I1 = Channel 1
I2 = Channel 2
I3 = Channel 3
I4 = Channel 4
I5 = Channel 5
I6 = Channel 6
I7 = Channel 7

1 = Force invalid
0 = Enable validity control from TRC

The RTS can also control this bit by:
1- HCB fnct code 71
2- HCB fnct code 50

TEST FRAME FROM PBD
0= disable
1= enable

FORCE INVALID								DELAY (7 bits)							
I7	I6	I5	I4	I3	I2	I1	I0	T	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
VV HCB FNCT 40								P0 HCB FNCT 35							

adr= 144
Label= DataValid

VLBA CORRELATOR PROJECTVLBA CORRELATOR PROJE NATIONAL RADIO ASTRONOMY OBSERVATORYNATIONAL CHARLOTTESVILLE, VACHARLOTTESVILLE, VA		
Title CONTROL REGISTER IN DATA INVALID XILINX		
Size A	Document Number 56000D029 (D029D04.TBL)	REV
Date: December 5, 1997	Sheet 4 of 5	

APPENDIX 3

FIGURE 4

Delay Center 3								Delay Center 2								Delay Center 1								Delay Center 0							
Tracks 24-31								Tracks 16-23								Tracks 8-15								Tracks 0-7							
Mode TRKs 28-31		Mode TRKs 24-27		TRK Sel 0 to 3		XTrk Parity		Mode TRKs 20-23		Mode TRKs 16-19		TRK Sel 0 to 3		XTrk Parity		Mode TRKs 12-15		Mode TRKs 8-11		TRK Sel 0 to 3		XTrk Parity		Mode TRKs 4-7		Mode TRKs 0-3		TRK Sel 0 to 3		XTrk Parity	
0=Pass all TRKs 1=Subs sys TRK 2=Invert on parity ck						16/8 Sel	Even / Odd	0=Pass all TRKs 1=Subs sys TRK 2=Invert on parity ck						16/8 Sel	Even / Odd	0=Pass all TRKs 1=Subs sys TRK 2=Invert on parity ck						16/8 Sel	Even / Odd	0=Pass all TRKs 1=Subs sys TRK 2=Invert on parity ck						16/8 Sel	Even / Odd
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
P0								P1								P2								P3							

HCB FNCT 36

RAM adr= 118.b	RAM adr= 119.b	RAM adr= 120.b	RAM adr= 121.b

Label= XPARITY
CROSSTRKPAR = 36 P0 P1 P2 P3 hcbCode bus,target,5,0x36,P0,P1,P2,P3

XTrk Parity - 00 gives odd, 8 track cross track parity.
The system track acts as a parity track for the other tracks.

TRK Sel
If the Mode = 1 for substitute system track:
Substitute the system track for the track selected by the TRK Sel field. For example, for Delay Center 0, TRK Sel specifies 0 to 3 if Mode 1 is in the first field or 4 to 7 if Mode 1 is in the second field.
If the Mode = 2 for Invert on parity check:
Use the system track for cross track parity. If there is a parity error, invert the track selected by the TRK Sel field. This is useful if there is a known bad track.

m:\def\d027def.dwg\d029d05.tbl

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title
CROSS TRACK PARITY CONTROL WORD

Size	Document Number	REV
A	56000D029	
Date: June 22, 1993	Sheet 5 of 5	

APPENDIX 3 FIGURE 5

The TEST EN can also be controlled with HCB fnct 70.

CONTROL WORD FOR 1 BIT SHUFFLER XILINX - THERE ARE 4

Test En	Spare	P81	P82	P83	P84			MAG						SIGN						MAG						SIGN					
		Ext Ctrl Bits				RW En		MUX 7			MUX 6			MUX 5			MUX 4			MUX 3			MUX 2			MUX 1			MUX 0		
		B0	B1	B2	B3	0&2	0&1	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1

S0	S1	S2	S3
Tracks=31-24	RAM adr= 130.w	RAM adr= 132.w	
S4	S5	S6	S7
Tracks=23-16	RAM adr= 134.w	RAM adr= 136.w	
S8	S9	S10	S11
Tracks=15-8	RAM adr= 138.w	RAM adr= 13A.w	
S12	S13	S14	S15
Tracks=7-0	RAM adr= 13C.w	RAM adr= 13E.w	

HCB FNCT 42 (S1..S16)

Modes Supported

MODE	RW En 0&2	RW En 0&1
N1-4E	1	0
V1-4E	0	1
N1-4F	0	0
N1-4G	0	0

Modes Supported

MODE	RW En 0&2	RW En 0&1
N1-2E	0	0
N1-2F	0	0
N1-2G	0	0

Modes Supported

MODE	RW En 0&2	RW En 0&1
N1-1E	0	0
N1-1F	0	0

Modes Supported

MODE	RW En 0&2	RW En 0&1
N2-1E	0	0

Modes Supported

MODE	RW En 0&2	RW En 0&1
N4-1E	0	0

Notes:
1) FFT sizes smaller than 512 do not affect personality requirements.
2) Ext ctrl bit B0 in Xilinx for tracks 24-31 is DELAY_SR_SEL which ctrls the selection of the DELAY MODEL source (dwg L008D26.SCH). The RTS should set this bit=0 for fnct 42 and use fnct 3B for control of just this bit.

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title: Bit Shuffler Personality 2 Ctrl Word

Size: A | Document Number: BSP-2-WD.TBL | REV:

Date: October 27, 1992 | Sheet ___ of ___

APPENDIX 4 FIGURE 1

The TEST EN can also be controlled with HCB fnct 70.

CONTROL WORD FOR 1 BIT SHUFFLER XILINX - THERE ARE 4

Test En	Spare	P81	P82	P83	P84			Mag B			Sign B			Mag A			Sign A			Mag B			Sign B			Mag A			Sign A		
		Ext Ctrl Bits				RW En		MUX 7			MUX 5			MUX 3			MUX 1			MUX 6			MUX 4			MUX 2			MUX 0		
		B0	B1	B2	B3	0&2	0&1	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
S0								S1						S2						S3											
Tracks=31-24								RAM adr= 130.w												RAM adr= 132.w											
S4								S5						S6						S7											
Tracks=23-16								RAM adr= 134.w												RAM adr= 136.w											
S8								S9						S10						S11											
Tracks=15-8								RAM adr= 138.w												RAM adr= 13A.w											
S12								S13						S14						S15											
Tracks=7-0								RAM adr= 13C.w												RAM adr= 13E.w											

HCB FNCT 42 (S1..S16)

Modes Supported

MODE	RW En 0&2	RW En 0&1
V1-4F	0	1

Modes Supported

MODE	RW En 0&2	RW En 0&1
V1-2E	0	1

Notes:
1) FFT sizes smaller than 512 do not affect personality requirements.
2) Ext ctrl bit B0 in Xilinx for tracks 24-31 is DELAY_SR_SEL which ctrls the selection of the DELAY MODEL source (dwg L008D26.SCH). The RTS should set this bit=0 for fnct 42 and use fnct 3B for control of just this bit.

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title: Bit Shuffler Personality 3 Ctrl Word

Size	Document Number	REV
A	BSP-3-WD.TBL	

Date: May 28, 1992 Sheet of

APPENDIX 4 FIGURE 2

The TEST EN can also be controlled with HCB fnct 70.

CONTROL WORD FOR 1 BIT SHUFFLER XILINX - THERE ARE 4

Test En	Spare	P81	P82	P83	P84	SPARE		MAG						SIGN						SPARE						SPARE					
		Ext Ctrl Bits						MUX 7			MUX 6			MUX 5			MUX 4													
		B0	B1	B2	B3			S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1

S0	S1	S2	S3
Tracks=31-24	RAM adr= 130.w	RAM adr= 132.w	
S4	S5	S6	S7
Tracks=23-16	RAM adr= 134.w	RAM adr= 136.w	
S8	S9	S10	S11
Tracks=15-8	RAM adr= 138.w	RAM adr= 13A.w	
S12	S13	S14	S15
Tracks=7-0	RAM adr= 13C.w	RAM adr= 13E.w	

HCB FNCT 42 (S1..S16)

Modes Supported:
V1-4G V1-2F

Notes:
1) FFT sizes smaller than 512 do not affect personality requirements.
2) Ext ctrl bit B0 in Xilinx for tracks 24-31 is DELAY_SR_SEL which ctrls the selection of the DELAY MODEL source (dwg L008D26.SCH). The RTS should set this bit=0 for fnct 42 and use fnct 3B for control of just this bit.
3) Mux select bits S2 are not used in this personality.

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title
Bit Shuffler Personality 4 Ctrl Word

Size	Document Number	REV
A	BSP-4-WD.TBL	

[redacted]: [redacted] Ma[redacted], 1[redacted] She[redacted]

A[redacted]EN[redacted]IX [redacted]4

[redacted]FIGURE [redacted]3

The TEST EN can also be controlled with HCB fnct 70.

CONTROL WORD FOR 1 BIT SHUFFLER XILINX - THERE ARE 4

Test En	Spare	P81	P82	P83	P84	Ctrl		MAG						SIGN						SPARE						SPARE					
		Ext Ctrl Bits				NC	Sel	MUX 7			MUX 6			MUX 5			MUX 4														
		B0	B1	B2	B3	CB	CB	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
S0								S1						S2						S3											
Tracks=31-24								RAM adr= 130.w						RAM adr= 132.w																	
S4								S5						S6						S7											
Tracks=23-16								RAM adr= 134.w						RAM adr= 136.w																	
S8								S9						S10						S11											
Tracks=15-8								RAM adr= 138.w						RAM adr= 13A.w																	
S12								S13						S14						S15											
Tracks=7-0								RAM adr= 13C.w						RAM adr= 13E.w																	

HCB FNCT 42 (S1..S16)

Modes Supported:
V1-2G

Notes:
1) 2k FFTs requires directing bits in a data stream to 4 FFT engines. CB24 is used for this direction as follows:
   CB24 = 0 for FFT engines 0&1 and 4&5.
   CB24 = 1 for FFT engines 2&3 and 6&7.
2) FFT sizes smaller than 512 do not affect personality requirements.
3) Ext ctrl bit B0 in Xilinx for tracks 24-31 is DELAY_SR_SEL which ctrls the selection of the DELAY MODEL source (dwg L008D26.SCH). The RTS should set this bit=0 for fnct 42 and use fnct 3B for control of just this bit.
4) NC = No Connection.

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title: Bit Shuffler Personality 5 Ctrl Word

Size	Document Number	REV
A	BSP-5-WD.TBL	

Date: December 1, 1992 Sheet of

APPENDIX 4

FIGURE 4

The TEST EN can also be controlled with HCB fnct 70.

CONTROL WORD FOR 1 BIT SHUFFLER XILINX - THERE ARE 4

Test En	Spare	P81	P82	P83	P84			MAG						SIGN						MAG						SIGN					
		Ext Ctrl Bits				MODE		MUX 7			SPARE			MUX 5			SPARE			MUX 3			SPARE			MUX 1			SPARE		
		B0	B1	B2	B3	CB	CB	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1

S0	S1	S2	S3
Tracks=31-24	RAM adr= 130.w		RAM adr= 132.w
S4	S5	S6	S7
Tracks=23-16	RAM adr= 134.w		RAM adr= 136.w
S8	S9	S10	S11
Tracks=15-8	RAM adr= 138.w		RAM adr= 13A.w
S12	S13	S14	S15
Tracks=7-0	RAM adr= 13C.w		RAM adr= 13E.w

HCB FNCT 42 (S1..S16)

MODE SELECT CTRL

Modes Supported:
V1-1E V2-1E V4-1E
N2-1F N4-1F

Mode	CB25	CB24
V1-1E	0	1
V2-1E	0	1
N2-1F	1	0
V4-1E	0	1
N4-1F	1	0

Notes:
1) FFT sizes smaller than 512 do not affect personality requirements.
2) Ext ctrl bit B0 in Xilinx for tracks 24-31 is DELAY_SR_SEL which ctrls the selection of the DELAY MODEL source (dwg L008D26.SCH). The RTS should set this bit=0 for fnct 42 and use fnct 3B for control of just this bit.

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title: Bit Shuffler Personality 6 Ctrl Word

Size: A | Document Number: BSP-6-WD.TBL | REV

Date: May 28, 1992 | Sheet ___ of ___

APPENDIX 4 FIGURE 5

The TEST EN can also be controlled with HCB fnct 70.

CONTROL WORD FOR 1 BIT SHUFFLER XILINX - THERE ARE 4

Test En	Spare	P81	P82	P83	P84	Ctrl		MAG			SIGN			Not used						Not used											
		Ext Ctrl Bits				NC	Sel	MUX 7			MUX 6			SPARE			SPARE			SPARE			SPARE			SPARE			SPARE		
		B0	B1	B2	B3	CB	CB	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1

S0	S1	S2	S3
Tracks=31-24	RAM adr= 130.w	RAM adr= 132.w	
S4	S5	S6	S7
Tracks=23-16	RAM adr= 134.w	RAM adr= 136.w	
S8	S9	S10	S11
Tracks=15-8	RAM adr= 138.w	RAM adr= 13A.w	
S12	S13	S14	S15
Tracks=7-0	RAM adr= 13C.w	RAM adr= 13E.w	

HCB FNCT 42 (S1..S16)

Mode	CB24
V1-1F	1
V2-1F	1
V4-1F	1
N1-1G	Note 1
N2-1G	Note 1
N4-1G	Note 1

Notes:
1) 2k FFTs requires directing bits in a data stream to 4 FFT engines. CB24 is used for this direction as follows:
   CB24 = 0 for FFT engines 0&1 and 4&5.
   CB24 = 1 for FFT engines 2&3 and 6&7.
2) FFT sizes smaller than 512 do not affect personality requirements.
3) Ext ctrl bit B0 in Xilinx for tracks 24-31 is DELAY_SR_SEL which ctrls the selection of the DELAY MODEL source (dwg L008D26.SCH). The RTS should set this bit=0 for fnct 42 and use fnct 3B for control of just this bit.

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title: Bit Shuffler Personality 7 Ctrl Word

Size	Document Number	REV
A	BSP-7-WD.TBL	
Date: May 28, 1992	Sheet of	

APPENDIX 4

FIGURE 6

The TEST EN can also be controlled with HCB fnct 70.

CONTROL WORD FOR 1 BIT SHUFFLER XILINX - THERE ARE 4

Test En	Spare	P81	P82	P83	P84	Ctrl		MAG			SIGN																				
		Ext Ctrl Bits				NC	Sel	MUX 7			MUX 6			SPARE			SPARE			SPARE			SPARE			SPARE			SPARE		
		B0	B1	B2	B3	CB	CB	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0	S2	S1	S0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1

S0	S1	S2	S3
Tracks=31-24	RAM adr= 130.w	RAM adr= 132.w	
S4	S5	S6	S7
Tracks=23-16	RAM adr= 134.w	RAM adr= 136.w	
S8	S9	S10	S11
Tracks=15-8	RAM adr= 138.w	RAM adr= 13A.w	
S12	S13	S14	S15
Tracks=7-0	RAM adr= 13C.w	RAM adr= 13E.w	

HCB FNCT 42 (S1..S16)

Modes Supported:
V1-1G V2-1G V4-1G

CB24 = 0 for Chs 0-7 & 16-23
CB24 = 1 for Chs 8-15 & 24-31

Notes:
1) FFT sizes smaller than 512 do not affect personality requirements.
2) Ext ctrl bit B0 in Xilinx for tracks 24-31 is DELAY_SR_SEL which ctrls the selection of the DELAY MODEL source (dwg L008D26.SCH). The RTS should set this bit=0 for fnct 42 and use fnct 3B for control of just this bit.

VLBA CORRELATOR PROJECT
NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VA

Title
Bit Shuffler Personality 8 Ctrl Word

Size	Document Number	REV
A	BSP-8-WD.TBL	

Date: Ma[redacted], 1[redacted] Sheet [redacted]

APPENDIX 4 FIGURE 7

APPENDIX 5

The servo tables were created by Jon Romney.

The following is a brief description of the format of the table that the Deformatter expects.

ServoTbl format:

Rel adr to center ref	Timer 1 wd	Action Byte	Validity Byte	Speed 4 ASCII bytes	Duration 2 ASCII bytes	Notes:
-128						Max Speed up
0						No servo action required
+127						Max slow down

Column desription:

- Timer - How often a servo command can be sent to the PBD.
  Timer binary value = Duration time in ms/5.
  The Deformatter will load this value into the ServoTimer after sending the last character of a servo command to the PBD. The DEF will not send another servo command until good time is received from the TRC and the ServoTimer has expired. The ServoTimer counts down on a 5ms tic and the code looks for a neg value.
- Action - If this byte is not 00 then DEF algorithm will not send cmds to the PBD.
  A value of FF is used around the center of the table.
- Validity - This byte used to ctrl the Validity output on the DEF card. If this byte is 0 then the validity is determined by the TRC otherwise the validity is forced invalid by the Deformatter.
- Speed - Four ASCII characters that will be sent to the PBD as parametrs to the BB command (speed).
  The units are 0.01 ips and includes the normal reference speed.
  Normal VLBA speed = 3e80h (160 dec * 100).
  MK III speed = 6978h (270 dec * 100).
  Max speed = 8ca0h (360 dec * 100)
- Duration - Two ASCII characters that will be sent to the PBD as parametrs to the BC command (duration of the speed command).
  The units are 0.01s. Two hex characters give a max of 2.55 seconds.

APPENDIX 6

Loopback Test Description

Loopback test can be started by hcbCode(bus,target,3,0x50,4,0x20)
which is: * START/STOP/SERVO CTRL: DEFOBSCTRL = 50 ZZ PP

ZZ = 04 Deformatter will initialize the Time Reference by reading Time from TRC micro 0, start observation but not control the PBD servo.

PP = 20 Deformatter enables the test frame output from the PBD, then sends functions 0D & 1B (observe, connect to PBD) to the TRCs. The test frame originates from the Master MCC. The test frame will stay enabled until the Deformatter receives a function code of 71 00.

Where function 71 is:
* PBD TEST DATA: PBDTSTDAT = 71 TT

Where:
TT = 00 PBD in normal operation mode. Turns off test mode.
= 01 PBD is enabled to output the test frame generated by the correlator's Master MCC.

Note: The test frame can also be turned on with function 50, but the only way to turn it off is with this function.

On a Deformatter monitor screen, ob 41 should be the same as function 50 above. ob 41 is an observe xy command. For x=4, we start normal observing with the TRC's connected to the transport. The Deformatter sends TRACK MIX, TRACK ASSM, Observe Parameters, and frame duration to the TRC's. The Deformatter will do a delay calculation and send servo commands. For y=1, we enable Test Frame data from the drive. y=0 would disable Test Frame and get data from the tape.

Monitor commands TF 0 or 1 should behave the same as hcb 71 0 or 1 above.

Local Variables:
fill-column: 82
End:

Appendix V Longitudinal Track Format Specification

This page intentionally left blank.

NATIONAL RADIO ASTRONOMY OBSERVATORY
Charlottesville, VA
VLBA PROJECT

SPECIFICATION: A56000N003 DATE: 1990 November 16

TITLE: VLBA Longitudinal Track Format

PREPARED BY: _____ APPROVED BY: _____

1. INTRODUCTION

1.1. Purpose

This specification defines the VLBA Longitudinal Track Format. The Format constitutes part of the interface from the Data Recording and Playback Subsystem to the Correlator Subsystem. In the former, it is applied by the VLBA Formatter to encode sampled VLBI data, plus critical time and auxiliary information, for recording and subsequent reproduction. The correlator's Playback Interface, using the same definition, inverts this process to retrieve the original information.

The Format also implies a secondary interface to the Monitor and Control Subsystem, which must supply some of the auxiliary information to be recorded.

1.2. Scope

The VLBA Longitudinal Track Format specifies the organization of data in a stream of bits recorded sequentially along one magnetic stripe, known as a "track". This specification does not address the "transverse track format" governing relationships among the multiple parallel tracks, recorded at the same or different times across the width of the tape. Nor does it involve the "channelization" and "digitization" attributes relating the bits recorded on a given track to a particular RF signal(s). (This latter relationship is in any event dynamic; it is permuted by a "barrel roll" feature which advances by one step for each Frame.)

1.3. Revision

The Format is implemented in the VLBA Formatter almost entirely as software; the same is true, to a lesser extent, in the correlator's Playback Interface. Thus this Specification can be revised straightforwardly if necessary. The present, initial version is intended for use in testing both the Format itself and early procurements of the VLBA tape inventory.

2. NOMENCLATURE & CONVENTIONS

2.1. Overall Structure of the Format

Bits recorded on a track are organized as a succession of contiguous FRAMEs: this specification essentially defines the structure of a Frame. Each Frame consists of four FIELDs, in order:

Sync Word	(Section 3.1)
Time Code	(Section 3.2)
Data Field	(Section 3.3)
Auxiliary Field	(Section 3.4)

Figure 1 shows the Frame and its subdivisions in graphical form.

2.2. Field Sub-Structuring

2.2.1. Each Field is encoded in BYTEs consisting of eight consecutive DATA BITs followed by a ninth PARITY BIT computed from them. With one exception, noted specifically below, parity is always odd.

2.2.2. Some Fields are further encoded into BCD DIGITs of four bits each, packed two to a byte. Interpreted as unsigned integers and restricted to values 0 - 9, they are used in decimal representations of time and auxiliary information. In the notation specifying these fields, Dn indicates the n'th BCD digit of Field D (starting at D1), in the order recorded; its value is given as PARAMETER // 10^k, meaning the digit in the 10^k place in the decimal representation of PARAMETER.

2.2.3. Some BCD Digits are recorded in order of increasing bit significance, instead of the conventional sequence. These BIT-REVERSED digits are noted explicitly in the descriptions below.

3. SPECIFIC FIELDS

3.1. Sync Word

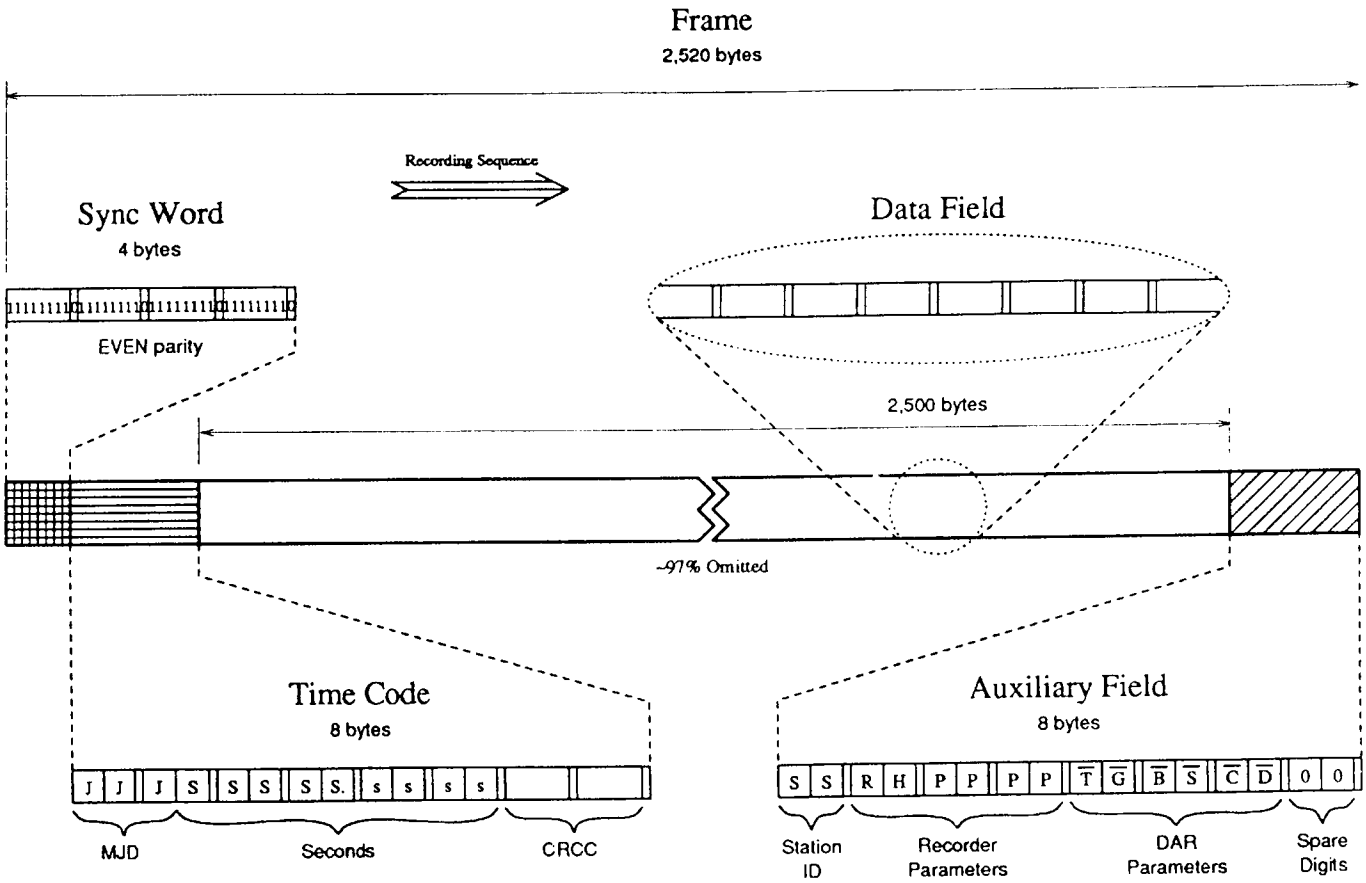
Each Frame begins with a Sync Word, an invariant pattern of four identical bytes, each consisting of eight "one" bits followed by a "zero" parity bit. This is the *only* case in which even parity occurs in the Format. Explicitly, the Sync Word is:

11111111011111111101111111101111111110

3.2. Time Code

The Time Code specifies the exact epoch, according to the station time standard, of the beginning of the Frame. In many VLBA modes this coincides with the sampling epoch of the first bit of the Data Field, but in general that bit may be delayed from the beginning-of-frame by a fixed offset, which depends upon the *transverse* track format and which is not directly relevant to the Longitudinal Track Format.

VLBA Longitudinal Track Format



Legend:

Byte: 8 data bits with parity bit.

A A BCD digits packed 2 in a byte.

$\overline{A}$ $\overline{A}$ Bit-reversed BCD digits.

Note: Parity always ODD except where specified EVEN.

90/11/16 jdr

Fig. 1

This field spans eight bytes, organized as twelve BCD digits and a two-byte error-detection pattern. There are three sub-fields --

3.2.1. MJD. The three least-significant digits of the Modified Julian Date, as defined in Section B of the "Astronomical Almanac".

T1	MJD // 100
T2	MJD // 10
T3	MJD // 1

3.2.2. SECONDS. The time interval elapsed since the beginning of the given MJD, expressed in seconds to a precision of 0.1 milliseconds. An implicit decimal point follows the unit seconds digit at T8.

T4	SECONDS // 10000
T5	SECONDS // 1000
T6	SECONDS // 100
T7	SECONDS // 10
T8	SECONDS // 1
T9	SECONDS // 0.1
T10	SECONDS // 0.01
T11	SECONDS // 0.001
T12	SECONDS // 0.0001

3.2.3. CRCC. A "CRC-16" cyclic redundancy code is computed on the data bits (*only*) of the first six Time Code bytes, using the generating polynomial X^16 + X^15 + X^2 + 1. The 16-bit output is split across the data bits of the final two bytes.

3.3. Data Field

The essence of each Frame is the Data Field, which carries 20,000 sampled VLBI data bits in 2,500 bytes.

3.3.1 A precise sampling epoch can be determined for each data bit, by counting from the first bit of the Field -- whose epoch in turn is determined by the Time Code and the offset described in Section 3.2. The time interval between data bits depends upon factors not directly relevant to the Longitudinal Track Format: details of the channelization, digitization, and recording speed. The last data bit of the Data Field is followed, after the same interval, by the first bit of the succeeding Frame.

3.3.2 Every data bit in the Data Field may, optionally, be modulated by a fixed pseudo-random sequence. The algorithm used to generate this pattern is described in Appendix A. The generator is re-initialized at the beginning of each Frame, and advances by one step on every bit -- including parity bits -- in the Data Field. A "one" output bit from the generator causes inversion of a *data bit* in the Field, but is ignored for parity bits.

3.4. Auxiliary Field

A trailer terminating the Frame, the Auxiliary Field specifies a variety of less-critical values which generally change only infrequently. It comprises eight bytes, organized as 16 BCD digits. The ranges shown are appropriate to current plans for the VLBA recording system. The auxiliary parameters can be grouped in four logical sub-fields --

3.4.1. Station Identification. A unique identification code, assigned to each station which records in this Format.

A1	STATION // 10	
A2	STATION // 1	

3.4.2. Recorder Parameters. The RECORDER and HEADSTACK which wrote this track (the current VLBA implementation has only one headstack per recorder), and SIGN and magnitude of the absolute headstack POSITION commanded for that headstack.

A3	RECORDER		0 - 1
A4	HEADSTACK		0
A5	POSITION SIGN		0: +; 8: -
A6	POSITION // 100		\
A7	POSITION // 10		> 000 - 999 microns
A8	POSITION // 1		/

3.4.3. DAR Parameters. The DAR (VLBA "D"-Rack), baseband CONVERTER, SIDEBAND, and sample BIT from which this data stream originated, and the "formatter track" GROUP and TRACK on which the formatted output is transmitted to the recorder. **All these items are recorded as bit-reversed BCD digits.** Those marked "BR" change every Frame as a result of the barrel roll.

A9	TRACK		0 - 7
A10	GROUP		0 - 4
A11	BIT	BR	0: LS bit; 1: MS bit
A12	SIDEBAND	BR	0: lower; 1: upper
A13	CONVERTER	BR	0 - 7
A14	DAR		0 - 1

3.4.4. Spare Digits. Defined as zeroes pending future assignment, which *must* be restricted to BCD values.

A15	0
A16	0

A. APPENDIX: THE PSEUDO-RANDOM ALGORITHM

As described in Section 3.3.2, a pseudo-random sequence is used to modulate the Data Field. This pattern is generated by a 16-bit feedback shift-register algorithm, implemented by the following C function:

```
unsigned  mod_func()
{
    static unsigned  ff[16] = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};
    unsigned  output, n;

    output  =  ff[10] ^ ff[12] ^ ff[13] ^ ff[15];
    for  ( n = 15;  n > 0;  n-- )  ff[n] = ff[n-1];
    ff[0] = output;

    return  output;

    }
```