TECHNICAL REPORT No. 13

THE MK-IIC CONTROLLER

P. Dooley, P. Harden, C. Janes

July 1991

NATIONAL RADIO ASTRONOMY OBSERVATORY
VLBA • ARRAY OPERATIONS CENTER (AOC)
P.O. BOX 0 • SOCORRO, NEW MEXICO 87801

• TABLE OF CONTENTS •

THE MK-IIC CONTROLLER

INTRODUCTION.

The MK-IIC Controller is an electronics chassis that can be used for remote control of MK-IIC data taking sessions. In the case of the VLBA, the Station Computer communicates with the Controller via the MCB serial bus. The Controller may be used to remotely control the VCRs and to select which baseband signal to be input to the MKIIC formatter. Status information from the formatter and the VCRs may be read back through the MCB.

A data quality analysis (DQA) function is included in the Controller which may be used to assure the viability of the cassette recordings.

The unit may be disabled by disconnecting from the MCB and by disconnecting the control cables that lead from the Controller to the VCRs.

THE VIDEO DATA QUALITY ANALYZER (DQA)

Formatted video data is received on BNC J9 (EXT Video), BNCs J11--J16 (VCR's 1--6), or from the MK-IIC formatter. A 74HCT151 8:1 multiplexer (BA13) routes one of the data signals to a T7033 clock recovery circuit (BC13) which strips the 4 Mhz clock from the data and decodes the video signal to provide a digital data stream. The data stream and a phase locked 8 MHz clock pass to a XC3030 Logic Cell Array (BA01). Here the data are converted from serial to 8-bit bytes, and the following status characters are decoded: beginning-of-frame (BOF), end-of-frame (EOF), and the 512 microsecond sync character (DPSYNC). Following BOF, the Helical Frame Count (HFC) is read and the Load Header (LDHDR) term generated. LDHDR activates the FIFO shift register to load the HFC and following video data to the microprocessor for analysis.

The program for the Logic Cell Array (LCA) is normally downloaded from a XC1736 Serial PROM (BE01). For development or diagnostic work, the PROM can be removed and a PC connected to that socket. The source is selectable by a jumper at BG22-23. Power-on resets the Array.

The decoded status and parallel data word are buffered to the CV. SRC 87C51 Video DQA Processor (BG01) by a CY7C421 8-bit FIFO Shift Register (BE09). The Video DQA Processor connects indirectly to the MCB through the MCB Interface Processor. The DQA Processor can send status and data bytes to the MCB or can receive a control byte for selecting which of 8 video data input channels to select. Like the Logic Array, the Processor is reset by power-on.

The DQA Processor further decodes the data from the Logic Array to track a running average of the data, to access test patterns in the head gap, and to detect frame count errors. The data average, a sampling of the 1 and 0 transitions in each frame, is available to the MCB to verify no DC bias exists in the video data. The first 10 bytes of test pattern recorded may be decoded and accessed by the MCB. The Processor checks that the Helical Frame Count (HFC) increments by 1 each frame until the TIC resets the counter from 59 to 0 at the end of each second. The MCB can check for a nonzero HFC or an HFC parity error.

THE MK-II CONTROLLER PROCESSOR

All the control and monitor functions not assigned to the DQA processor are performed by the CTL.SRC 87C51-FA MK-II Controller Processor (AC22). The Processor decodes the audio data lines from the VCRs, receives status information from the VCRs and the MKIIC formatter, controls the VCRs, and selects the upper or lower sidebands (USB or LSB). An 8-bit bi-directional bus is used to pass status to and receive control information from the MCB via the MCB Processor. Power-on resets the Processor.

The same VCR command functions available on a remote control paddle are available to the MCB by tying a serial output on the Controller Processor to the IR code signal line in the VCR. The term IR derives from the design of the remote paddle which works via a modulated IR (infrared) beam. The MKIIC Controller does not use an IR beam, nor is the remote control paddle used for VCR control. The functions commanded are Stop, VCR/TV, Rewind, Fast Forward, Record, Play, Pause, and Power on/off.

A serial data line from the VCR circuitry provides VCR status to the Controller Processor. A 74HCT164 Serial Data Register (AF20) converts the serial line to 8 bits parallel for input to the Processor. A second signal line from the VCR clocks the data into the serial-to-parallel converter; and clocks the parallel bytes into the Processor using a 74HCT193 Load Counter (AF23). Status information includes Pause Mode, Normal Play or Slow Play, Rewind, Fast Forward, Record, Power On/Off, and Tape Loaded/Unloaded. The Processor also senses if no VCR is present.

Time code is decoded from the Audio Line by the Processor. Each VCR audio line is received with a 74C04 input buffer.

Up to 6 VCRs are supported by use of one 74HCT138 1:8 multiplexer and three 74HCT151 8:1 multiplexers. The 1:8 multiplexer (AH49) selects which VCR will receive the IR serial code, and the other three multiplexers select serial data (AH20), serial clock (AH29), and audio (AD43) inputs from the same VCR. The audio multiplexer can also select time code from EXT Audio (a BNC connector) or from the MKIIC formatter. Connection of the IR code, serial data, serial clock, and audio signals between the VCR and the MKIIC controller are via a ribbon cable that passes from a 10 pin connector on the Controller to the "D" connector on the VCR. Three address lines from the Processor select the same VCR for all four multiplexers.

Another 74HCT151 multiplexer (AF43) selects status lines from the MKIIC formatter for the Processor. Status information includes Data Error, Time Error, Lock Error, Test Pattern On/Off, 1PPS sync edge + or -, Calibration On/Off, and Run or Set Mode. Data Error means that the base band data signal into the formatter is absent or weak; Time Error means the 1PPS sync input to the formatter is missing; and Lock Error means the 5 MHz is or was missing. The status input is selected by address lines from the Processor. Power-on resets the processor.

Finally, the Controller Processor selects the sideband via a 75447 Coax Relay Driver (AH38). A relay on the back of the formatter uses the driver output to select USB (Upper Sideband) or LSB (Lower Sideband).

THE MONITOR AND CONTROL BUS (MCB) INTERFACE PROCESSOR

The MCB.SRC 87C51FA MCB Interface Processor (AC01) connects to the MCB via an RS422 serial port, using a 75176 Bi-directional transceiver (AH01). The processor converts the serial information to an 8-bit bi-directional data bus (MCB-0 through MCB-7). Using the bus, the MCB Processor receives data from and sends control to the Video DQA Processor and the MK-II Controller Processor. DIP switches (AA19) select the MCB ID byte. During the MCB set-up from the station computer, the DIP switches are read as the "Interface ID byte" which determines the MCB block address. The MK-IIC Controller is currently assigned ID = 44; starting address = 4400. The Processor is reset by power-on. For all practical purposes, the MK-IIC Controller Interface Processor acts and functions as any VLBA Standard Interface Board.

RECOMMENDED MCB "SET-UP"

The VLBA Station Computer should properly initialize the MK-II Controller during normal operation and when the "MARK2" overlay is invoked. The SETALL function should also initialize the MK-II Controller. In the event the MK-II Controller needs to be manually initialized, this is performed via the SETADDRESS function. Recommended parameters are as follows:

```
K— SET INTERFACE ADDRESS BLOCK
     EXECUTE
   INTERFACE ID #44
    1st ADDRESS #4400
    BLOCK SIZE #050
```

POWER AND FRONT PANEL

Front panel LEDs indicate the status of the MCB communications, power, selected sideband and DQA/tape status as follows:

XACTV LED indicates the MCB transmit line is active (addressed to any Interface Board)

MSG LED illuminates when an MCB command or monitor request is received specific to the MK-IIC controller

DOUT illuminates when the MK-IIC Controller is responding to a monitor request.

PARX LED indicates a parity error occurred during the MCB request.

LSB and USB LED indicates the sideband from the Baseband Converter selected as input to the MK-II Formatter.

DQA AUDIO LED displays the audio data from the selected VCR. It should have a burst of activity once per second.

DQA EOF/BOF LED is derived from the video data from the selected VCR. The BOF and EOF from alternate frames illuminates the LED for a 30-pps "blink." Pauses in this blink-rate indicates frame drop-outs.

EOF/BOF TP The front panel test point goes logic HI at EOF and LO on BOF on each frame, or 60 pulses per second, decoded from the selected video input to the DQA (VCR's or the Formatter). This is used for playback verification and for VCR alignment purposes.

POWER LED indicates +5v power is available to the MK-II Controller. Since power is derived from the formatter +12v, this LED indicates 1) power is being properly derived and 2) the ribbon cable between the formatter and controller is properly connected.

POWER ON SWITCH on the rear panel disconnects the +12v formatter DC from the controller. This allows the controller to be powered off (or cycled to reset the controller) without interrupting the formatter.

OPERATION

OPERATION OF THE MK-IIC FORMATTER

To record data in the MK-IIC format, the following items should be checked:

1. Formatter, Controller and VCR's turned ON.

2. All signals connected (video, audio, sync, etc.)

3. Select proper 1PPS sync (normally POS edge).

4. Formatter time display correctly set.

5. Test Pattern and Calibrator OFF; mode switch in RUN.

6. Select desired sideband: LSB or USB via MARK2 overlay.

7. **If the clock timer is used to begin the recording session, remote control of the VCR via the MCB is blocked!**

Formatter front panel indicators, and/or the displayed errors on the MARK2 overlay, are evaluated as follows:

DATA OK means the selected LSB/USB input signal is present and a sufficient level for the formatter.

TIME OK indicates the 1PPS is present.

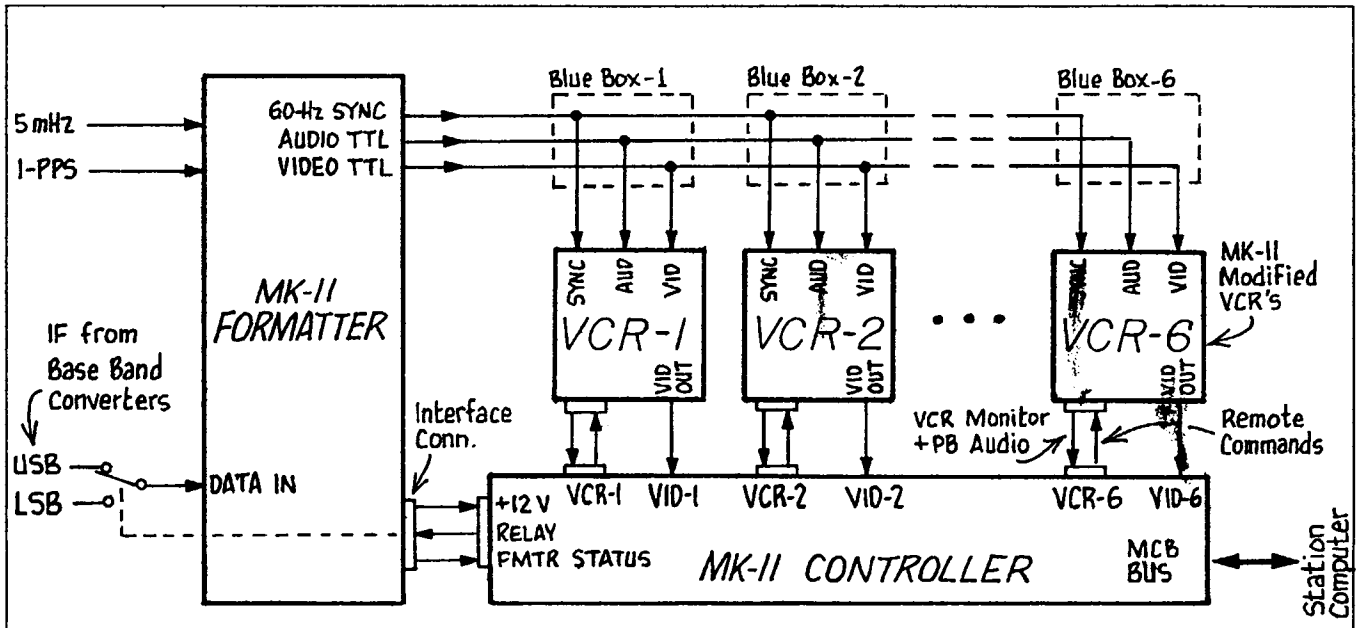
LOCK OK indicates the 5 MHz is present. If the 5 MHz drops a few cycles and returns, the Lock Error must be manually cleared by the formatter RESET switch.

1PPS POLARITY switch should be set to positive. If set to negative, an error message is displayed on the MARK2 overlay.

FORMATTER TIME display must be checked carefully with WWV or other time standard, to the second.

LSB/USB sideband selected is shown on both the MARK2 overlay and LED's on the Controller front panel. The relay defaults to LSB upon initialization.

TEST PATTERN and CALIBRATOR functions are not used and must be turned OFF to properly record MK-II data.

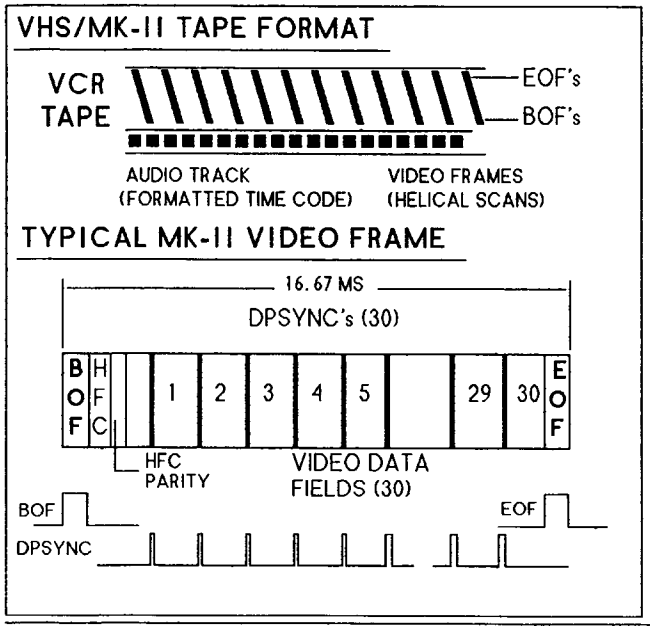


BLOCK DIAGRAM OF VLBA MK-II ARRANGEMENT

USE OF THE DQA

The MK-IIC Controller DQA (Data Quality Analyzer) may be used to verify that the proper VIDEO format has been recorded on a tape. The Operator must enter the VCR number on the MARK2 Overlay to select which DQA input is to be analyzed. (Or, select via MCB #441F)

Video is recorded onto the VCR tape in standard VHS format: sixty helical scans, or frames, per second. Each recorded frame consists of a BOF (Beginning-of-Frame) at the beginning of each 16.67 ms frame of data, HFC (Helical Frame Count), HFC parity, a DPSYNC (a sync character recorded every 2048 data bits or every 512 microseconds after BOF), and an EOF (End-of-Frame) at the end of the data frame. These special characters, BOF, EOF, Helical Frame Count, parity and DPSYNC's are all added by the MK-II Formatter. The MK-IIC Controller DQA reads the video signal real-time on a frame-by-frame basis to decode BOF, DPSYNC and EOF, reads the helical frame count (HFC), and checks the 1/0 transitions of the data fields (the Astronomical data). These frame characteristics are available as MCB monitor data to verify proper operation of the VCR, formatter and tape quality. Deviations from the normal represent errors and thus reflect the quality of the MK-II recording session.



BOF - There are 60 BOF's per second. The video BOF character read each frame is not stored. (BOF at #4410 is derived from the audio track and should not be used).

EOF - Like BOF, there are also 60 EOF's per second. The decoded EOF character is not stored. Each valid BOF-EOF occurance increments a non-resettable counter at address #442E-442F to tally the number of properly decoded frames per second. (EOF at #4417 is from the audio track and should not be used).

DPSYNC - There are 30 DPSYNC characters per frame, or 1800 per second. Each successfully decoded DPSYNC increments a non-resettable counter at address #442C-442D. Missing DPSYNC's are called "drop outs" (DO) and the data recorded in that frame may be corrupted. The MARK2 screen calculates and displays the drop out rate as DO=00 (no drop-outs) from the difference in DPSYNC's per second from the 1800/sec. ideal. More than 6 DO's/sec. signifies questionable tape quality.

BOF ERROR - The MARK2 screen compares the EOF-BOF tallies at #442E-2F with 60/sec. ideal and displays the difference as BOF (frame) errors per second. A zero difference is normal (BOF=00) with an error rate less than 3/sec. acceptable.

FRAME COUNT (HFC) - The formatter records an incremental Helical Frame Count (HFC) after each BOF. The DQA reads this frame count, plus the associated parity value, and checks for an increment of 1 for each frame. A failure to increment by one, and/or a HFC parity error, is reported as a DQA FRAME COUNT ERROR at address #442B and the MARK2 screen. Bit 7 of this word is set to 1 by the DQA signifying the video data is OK by 10 successive BOF's being decoded. Thus, the proper value at #442B should be 0080; the MARK2 screen displays the error DQA BAD VIDEO if bit 7 is not set. The error count is reset each second. Note that two errors per frame could occur if both the parity count and increment were in error.

1/0 AVERAGE - A digital sampler in the MK-II Formatter converts the selected LSB/USB signal from the Baseband Converter to a data stream of 1's and 0's, forming the video signal recorded on each frame. The DQA samples the video data between the 1st and 2nd DPSYNC characters, which at 2 mHz bandwidth is about 400 transitions, and reported at address #4420 (a count of 40=400). The MARK2 screen displays an error rate based on the difference from the actual average to the 400 ideal. A positive error value (AVG=+) means more 1's than 0's. Plus or minus three is an acceptable error; larger values suggests the baseband signal has a DC bias, contains a modulation component, or insufficient level to properly drive the sampler in the Formatter.

1-SECOND TIC-COUNTER - Address #4430 is a non-resettable 1-second counter used to update the BOF, EOF, DPSYNC and Frame Count errors within the DQA and for the MARK2 screen. This 1-second TIC is generated within the MK-II Controller and does not reflect proper operation of the *system 1-PPS.*

DQA SELECT - Address #441F is a command address to select the source of audio and video to be analyzed by the MK-II Controller and DQA. For example, to monitor VCR#3, a command word of 0004 is issued to address #441F; the last received command is echoed as monitor data. The MARK2 screen uses this address to select the DQA channel desired by the operator.

LSB/USB RELAY CONTROL - Address #441B is a command address to select either LSB or USB from the Baseband Converter as the input to the Formatter, and hence the sideband to be recorded on the MK-II recorders. A word of 0000 selects LSB; word 0008 selects USB. The monitor value at this address echoes the last received command. Default on power up is LSB. The MARK2 screen uses this command/echo address to toggle between LSB and USB.

TEST PATTERN - Test pattern is invoked by a switch on the Formatter *and must be OFF* to record data. The first 10 words of test pattern written are available to the MCB at address #4421-442A. No value can be considered normal since this feature is not used by VLBA. It is not supported by the MARK2 screen, other than to display the error TEST PATTERN ON since this condition inhibits normal recording of the Formatter data.

QUICK SYSTEM CHECK-OUT (Making a Test Tape with the DQA)

The following are the basic procedures for making a test tape. This can be performed locally or remotely using the MARK2 overlay screen. A brief session of recording a few minutes of data on each VCR and playing back the tape for DQA analysis prior to a MK-II run will verify the system and VCR's are functioning properly. ***NOTE: Remote operation by the MCB is blocked if VCR is in the timer mode (set to record at a preset time).***

1. Access the MARK2 overlay screen. Select VCR UNIT and DQA UNIT numbers to VCR to be used for test. Toggle the LSB/USB INPUT to check proper operation.

2. Install fully rewound data tape into selected VCR. Ensure MODE indicates TAPE LOADED. RECORD for 5 minutes with data connected. Record the TIME displayed on the MARK2 screen (Formatter time) when recording was started. This is the time being recorded on the MK-II tape.

3. REWind and PLAYback tape and evaluate. Ensure TIME on MARK2 screen is updating. *It should be the time recorded in step 2 above (not current station time).* This verifies the audio/time loop is functioning properly.

4. In PLAYback, check for acceptable data drop-outs (DO=0, +/- 6); zero BOF errors (+/- 3) and no data averaging errors (AVG=0, +/- 3). This verifies the video loop is functioning properly and tape quality is good.

5. Rewind and record the data session, or switch to the next recorder based on a judgement about the quality of the recording.

6. Send the test tape to the AOC, c/o MK-II Correlator, Socorro, NM, if any problems need further identification.

Expected Error Rates

Ideally, the error rate for DO, BOF, HFC (frame count) and AVG should be zero. In practice, occasional errors will likely occur just from tape quality. The MARK2 screen flags excessive counts by highlighting these errors when exceeding an acceptable rate. For DO (drop-outs of DPSYNC), over 12 errors per second are flagged. BOF errors are highlighted over 3 per second. AVG (1/0 data average) typically changes a few counts over many seconds, but is flagged when exceeding plus or minus three counts per second. An HFC frame count error over one per second is flagged by an error message on the MARK2 screen.

Evaluate these errors with caution! In RECORD, the DQA is analyzing the Formatter video *before it is recorded on the VCR tape.* Values tend to be very close to ideal during record and not indicative of true tape quality. IN PLAYBACK, however, the DQA is "reading" exactly what was recorded on the VCR tape and presents a more faithful analysis of the quality of the MK-II tape(s). It is for this reason that the quick system check-out above be performed on each recorded VCR tape: to allow for proper DQA analysis *in the PLAYBACK mode.*

REFERENCES:

The MKII Correlator System, VLBA Technical Report 9, C. Janes, March 1991.

The Mark II Data Quality Analyzer, Electronics Division Internal Report No. 185, R. Lacasse, Feb. 1978.

The Mark II Formatter, Unnumbered Report, A. Gallerani, 1988.

THE MONITOR AND CONTROL BUS "MARK2" OVERLAY SCREEN

The MK-II system is controlled and monitored by an overlay screen on the VLBA station computer called the "MARK2." It is imbedded within the FORMAT menu. Its general appearance is as follows:

```
K ——MARK II CONTROLLER ——————————
        INPUT LSB
 VCR UNIT 1    MODE TAPE LOADED
DQA_INPUT 1       TIME 20:11:06
  DO 3     BOF 0     AVG -2
NO ERRORS
 (IF ERRORS... DISPLAYED HERE)
```

The function of each element is described below. The MARK2 screen shows the status of ONE VCR and ONE DQA input which must be selected by the operator. The keys CTRL-N and CTRL-P (for Next and Previous) cycle through each option for selection. For HELP type "?"

INPUT LSB (USB) indicates the current sideband selected as input to the Formatter. To change the sideband, position the cursor on the present selection; the ENTER key will toggle the selection.

VCR UNIT indicates the VCR selected. Enter the proper VCR number to change, or select via CTRL-N.

MODE indicates the mode of the VCR selected by VCR UNIT. This includes tape loaded, play, stop, record, rewind, etc. To change the mode via the screen, position the cursor on the displayed mode, cycle through the options with CTRL-N, and activate with the ENTER key. Control is blocked, however, if the VCR is in the timer mode (set to record from VCR clock).

DQA INPUT indicates the VCR selected input to the DQA. The following screen displays of TIME, DO, BOF, AVG and displayed error messages are derived from the DQA input selected. *Normally, the VCR UNIT and DQA INPUT should be selected to the same VCR.*

TIME displays either the Formatter time or the VCR playback time as follows: If the DQA input selected is a VCR in the *playback mode,* the TIME is displaying the time previously recorded on the tape. Formatter time is displayed here under all other circumstances. In other words, TIME displayed should be the station time EXCEPT when playing back a previously recorded tape.

DO - is the drop-out error rate per second calculated from missing DPSYNC's. During RECord, there should be no DO's; during PLAYback, a few DO's occur due to tape quality. Above 12 DO's/sec. the error count is highlighted to signify excessive drop-outs.

BOF - is the count of missing beginning-of-frame (BOF) characters. During RECord, BOF error should be zero; during PLAYback occassional BOF errors can occur. Above 3 BOF errors/sec. the count is highlighted to show excessive missing frames.

AVG - is the error in the data 1/0 transitions for each frame. This value will typically vary plus or minus 2-3 counts over several seconds in both RECord and PLAYback modes. Above 3 AVG errors the count is highlighted to alert of possible noise or offset in the 1/0 data transitions.

NO ERRORS - This field is where up to 3 error messages are displayed; otherwise No Errors is displayed. Formatter errors include Calibrator or Test Pattern ON, time or data not OK, or not in the RUN mode. DQA error messages include DQA BAD VIDEO, FRAME COUNT ERROR, etc.

MK-II CONTROLLER
MCB ADDRESS ASSIGNMENTS & FORMATS

MCB ADDRESS & DATA

(BLOCK ADDRESS xx00—xx19)

ADDR	15 BIT 8	7 BIT 0
*4400	VCR #1 COMMAND	
4401	VCR #2 COMMAND	
4402	VCR #3 COMMAND	
4403	VCR #4 COMMAND	
4404	VCR #5 COMMAND	
4405	VCR #6 COMMAND	
4406		
4407		
4408	VCR #1 MONITOR	
4409	VCR #2 MONITOR	
440A	VCR #3 MONITOR	
440B	VCR #4 MONITOR	
440C	VCR #5 MONITOR	
440D	VCR #6 MONITOR	
440E		
440F		
4410	4410—4417 AUDIO DECODE MONITOR DATA FROM SOURCE SELECTED BY ADDR 441F	BOF (0FFH)
4411		FRAME COUNT
4412		SEC x10 / SEC x1
4413		MIN x10 / MIN x1
4414		HRS x10 / HRS x1
4415		DAY x10 / DAY x1
4416		nu / DAY x100
4417		EOF / STATUS
4418		
4419		

* Absolute Address with Block Address=#4400H

ADDRESSES 441A—4430H ON NEXT PAGE . . .

MESSAGE FORMATS

4400—4405 VCR COMMANDS

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

- **VCR STOP** 80
- VCR/TV 0040
- REWIND 0020
- FAST FORWARD 0010
- RECORD 0008
- **PLAY** 0004
- PAUSE MODE 0002
- POWER 0001

4408-440D VCR MONITOR

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

MONITOR WORDS OF BITS 0-7:

NO VCR PRESENT-BD
PAUSE MODE=C8
NORM PLAY=D8
SLOW PLAY=DD
PWR OFF/NO TAPE=E0
PWR OFF/TAPE IN=E2
PWR OFF/TAPE IN=F0
PWR ON/TAPE IN=F2
REWIND=F5
FAST FORWARD=F6
SYNCING UP TO PLAY=F8
TIMER RECORD MODE=FA
RECORD=FE

END-OF-TAPE WARNINGS: **AUTO REWIND** IN PROGRESS=D5; AUTO **TAPE REWIND** DONE=D2

4410 - AUDIO BOF

BEGINNING OF FRAME

BOF=0FFH • ANYOTHER WORD=ERROR

4411 FRAME COUNT (FC)

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

- FRAME COUNT (0-59) (BITS 2-7)
- FRAME LENGTH
  - 0=66.000 ms
  - 1=66.001 ms

BD=NO FC DECODED

4412—4416 FORMATTER TIME

DAYS • HOURS • MINUTES • SECONDS

BD=NO TIME/AUDIO BEING DECODED

4417—AUDIO EOF & STATUS

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

END OF FRAME EOF=00H

BD=NO EOF

- NEW FORMAT
- NO DATA
- CALIBRATOR ON

MK-II CONTROLLER
MCB ADDRESS ASSIGNMENTS & FORMATS

MCB ADDRESS & DATA
(BLOCK ADDRESS xx1A—xx32)

ADDR	15 BIT 8	7 BIT 0
*441A	MON	FMTR STATUS
441B	COMMAND	RELAY CONTROL
441C		
441D		
441E		
441F	COMMAND	DQA SELECT
4420	MON - See Note 1	1/0 AVERAGE
4421	MON - See Note 2	TEST PATTERN 1
4422		TEST PATTERN 2
4423		TEST PATTERN 3
4424		TEST PATTERN 4
4425		TEST PATTERN 5
4426		TEST PATTERN 6
4427		TEST PATTERN 7
4428		TEST PATTERN 8
4429		TEST PATTERN 9
442A		TEST PATTERN 10
442B	MON	FCERR
442C	MON	DPSCT-LO BYTE
442D	MON	DPSCT-HI BYTE
442E	MON	EOFCT-LO BYTE
442F	MON	EOFCT-HI BYTE
4430	MON	TIC-COUNTER
4431		
4432		

* Absolute Address with
Block Address=#4400H

NOTES
1— 1/0 (ones/zeroes) average taken over each frame and should average approx. 40H
2— Test pattern NOT USED on all VCR tapes and no standard for VLBA use established; thus no pattern can be considered "normal."

441A FORMATTER STATUS

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

- DATA OK (0=ERR)
- TIME ERROR — 0=OK
- LOCK ERROR — 0=OK
- TEST PATTERN OFF — 0=ON
- +SYNC EDGE — 0=NEG EDGE
- CAL ON — 0=CAL OFF
- MODE SW IN RUN — 0=SET MODE

441B COAX RELAY CONTROL
RELAY CONTROLLED BY BIT 3 ONLY
00=RELAY OFF (LSB) 0008=RELAY ON (USB)

441F AUDIO/DQA VIDEO SELECT

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

SELECT SOURCE OF AUDIO/VIDEO DATA

WORD 00H= DEFAULTS TO FORMATTER

- EXTERNAL — 80
- FORMATTER — 0040
- VCR #6 — 0020
- VCR #5 — 0010
- VCR #4 — 0008
- VCR #3 — 0004
- VCR #2 — 0002
- VCR #1 — 0001

FRAME COUNT ERROR

RAW *DP SYNC* COUNT
•1800/SEC., NOT RESETTABLE

RAW *EOF* COUNT
•60/SEC., NOT RESETTABLE

RAW *1-SECOND TICK* COUNTER
•COUNTS SECONDS, NOT RESETTABLE

442B FRAME COUNT ERROR

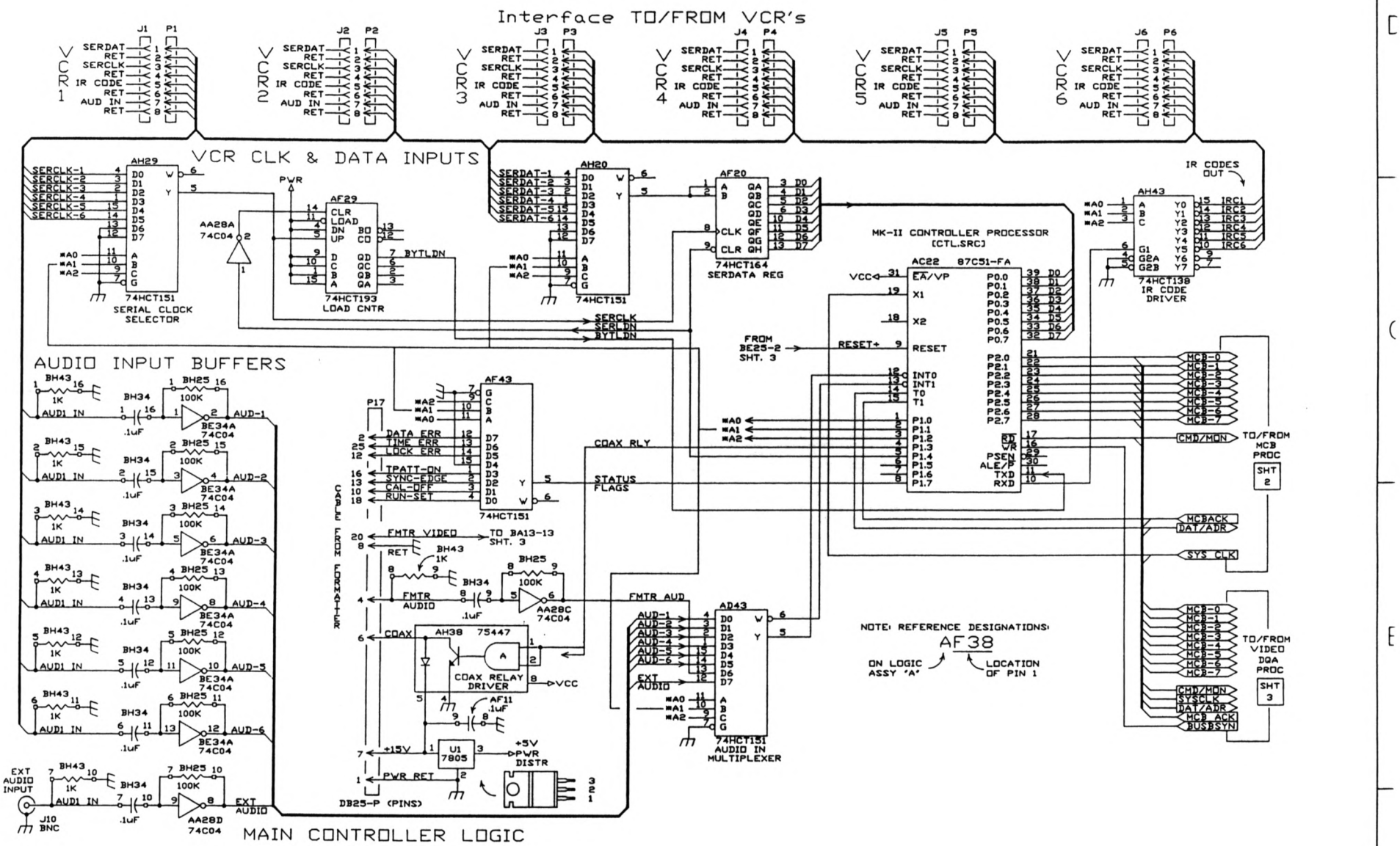
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

- VIDEO DATA OK BIT
- FRAME COUNT ERROR

0080=NORMAL
0083=3 ERRORS (EXAMPLE)
0000=NO VIDEO PRESENT

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

	DAT	WN BY	D BY	DES
A	3-91	HARDEN		REDESIGNED AND REDRAWN ;

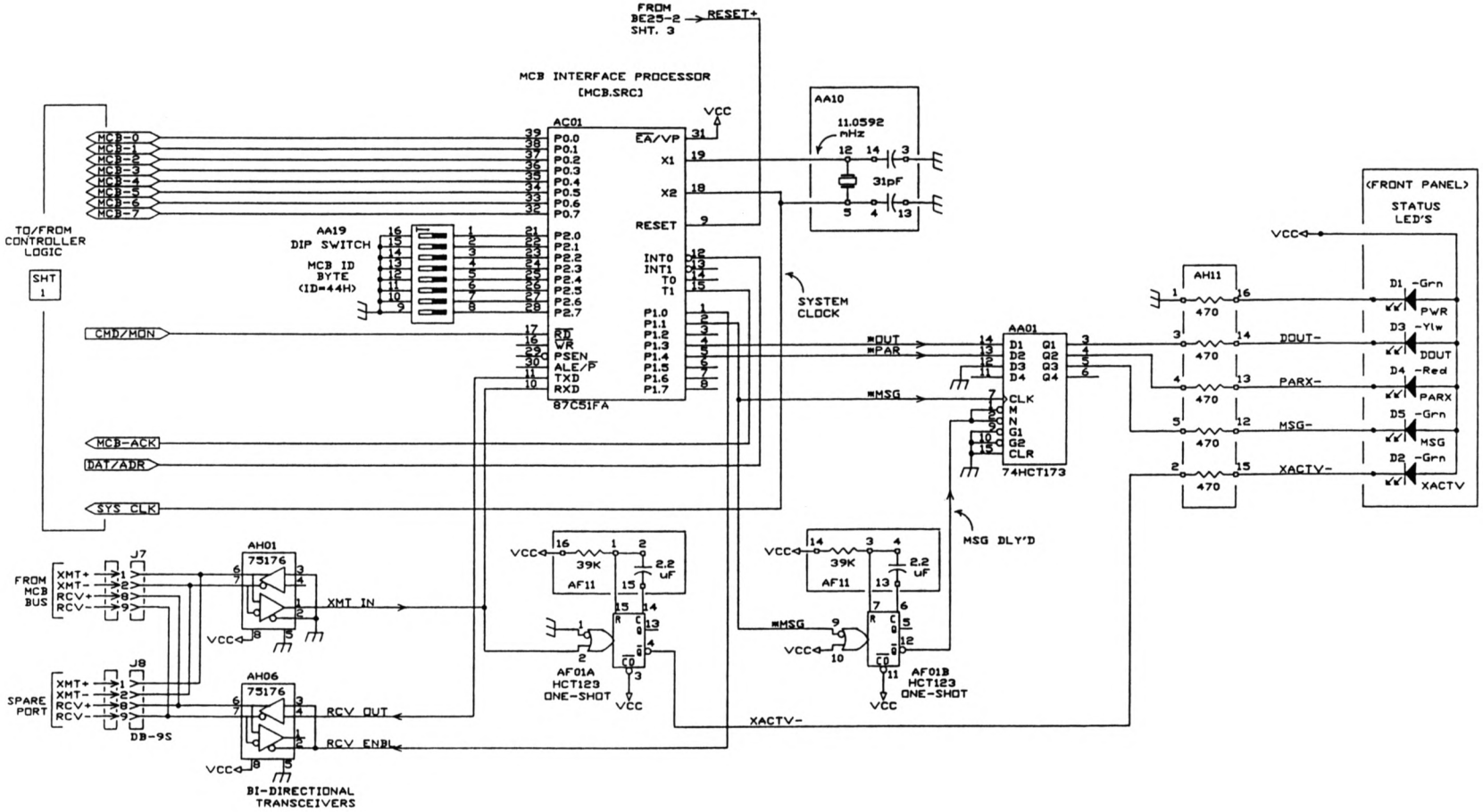


ACAD : MKIISK-1
REF : ORCAD

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES			
TOLERANCES : ANGLES ±	—		
3 PLACE DECIMALS (.000) ±	—		
2 PLACE DECIMALS (.00) ±	—		
1 PLACE DECIMALS (.X) ±	—		
MATERIAL :	—		
FINISH :	—		
NEXT ASSEMBLY	DWG. TYPE		

VLBA PROJECT	MKII CONTROLLER	NATIONAL RADIO ASTRONOMY OBSERVATORY SOCORRO, NEW MEXICO 87801	
TITLE	MKII CONTROLLER SCHEMATIC DIAGRAM	DRAWN BY HARDEN	DATE 3-91
		DESIGNED BY DOOLEY	DATE 8-90
		APPROVED BY	DATE
SHEET NUMBER 1 of 3	DRAWING NUMBER C54002S004	REV. A	SCALE —

REV	DATE	DRAWN BY	APPRV'D BY	DESCRIPTION
A	3-91	HARDEN		REDESIGNED AND REDRAWN)



MCB INTERFACE LOGIC

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES

MATERIAL :

FINISH :

VLBA PROJECT: MKII. CONTROLLER

TITLE: MKII CONTROLLER SCHEMATIC DIAGRAM

NATIONAL RADIO ASTRONOMY OBSERVATORY
SOCORRO, NEW MEXICO 87801

DRAWN BY	DATE
HARDEN	3-91
DESIGNED BY	DATE
DOOLEY	8-90
APPROVED BY	DATE

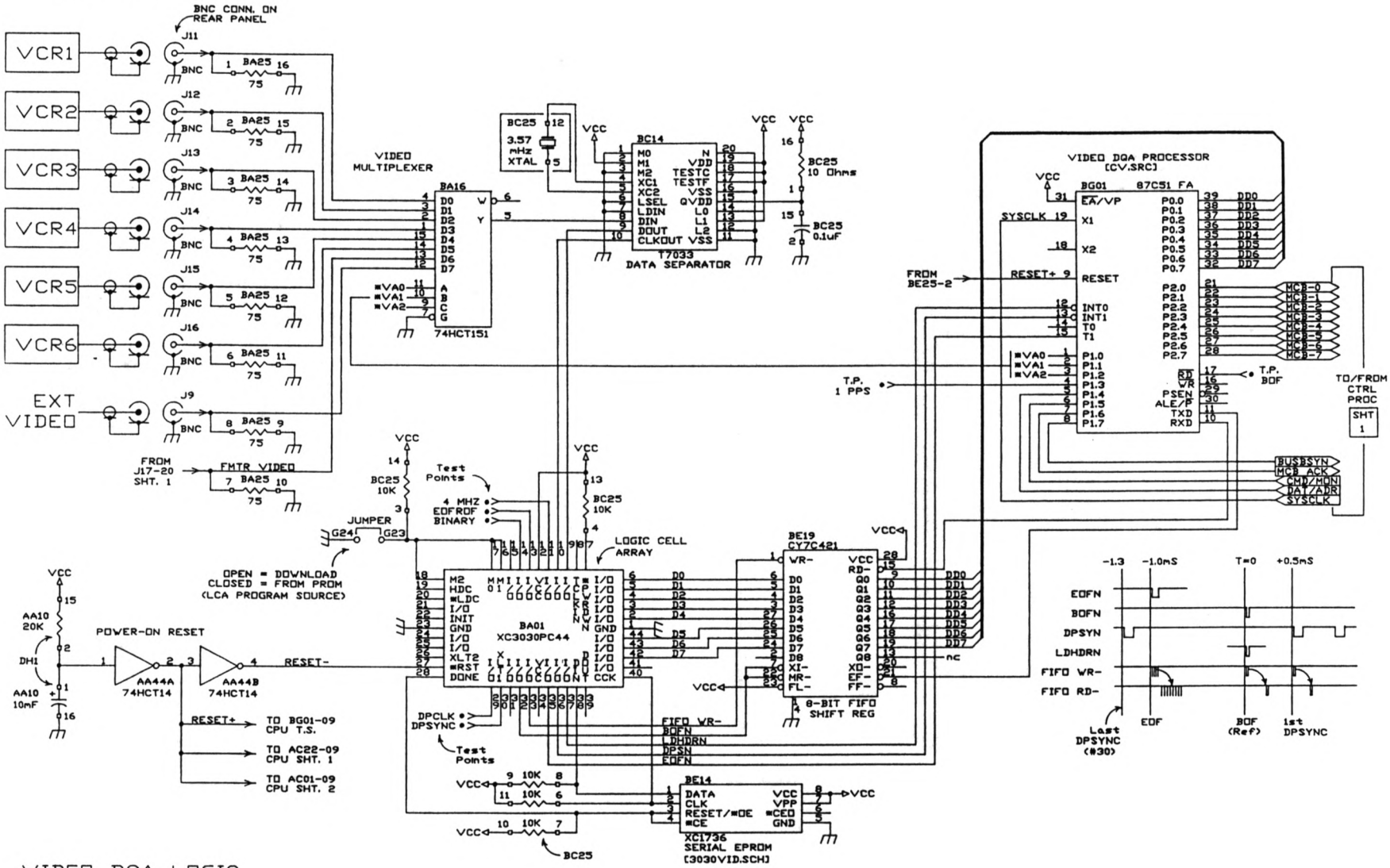
SCALE

ACAD : MKIISK-2
REF : ORCAD

PROPERTY OF NRAO

	DATE	DRAWN BY	CHK'D BY	DESCRIPTION
A	3-91	HARDEN		REDESIGNED AND REDRAWN)

VCR VIDEO INPUTS

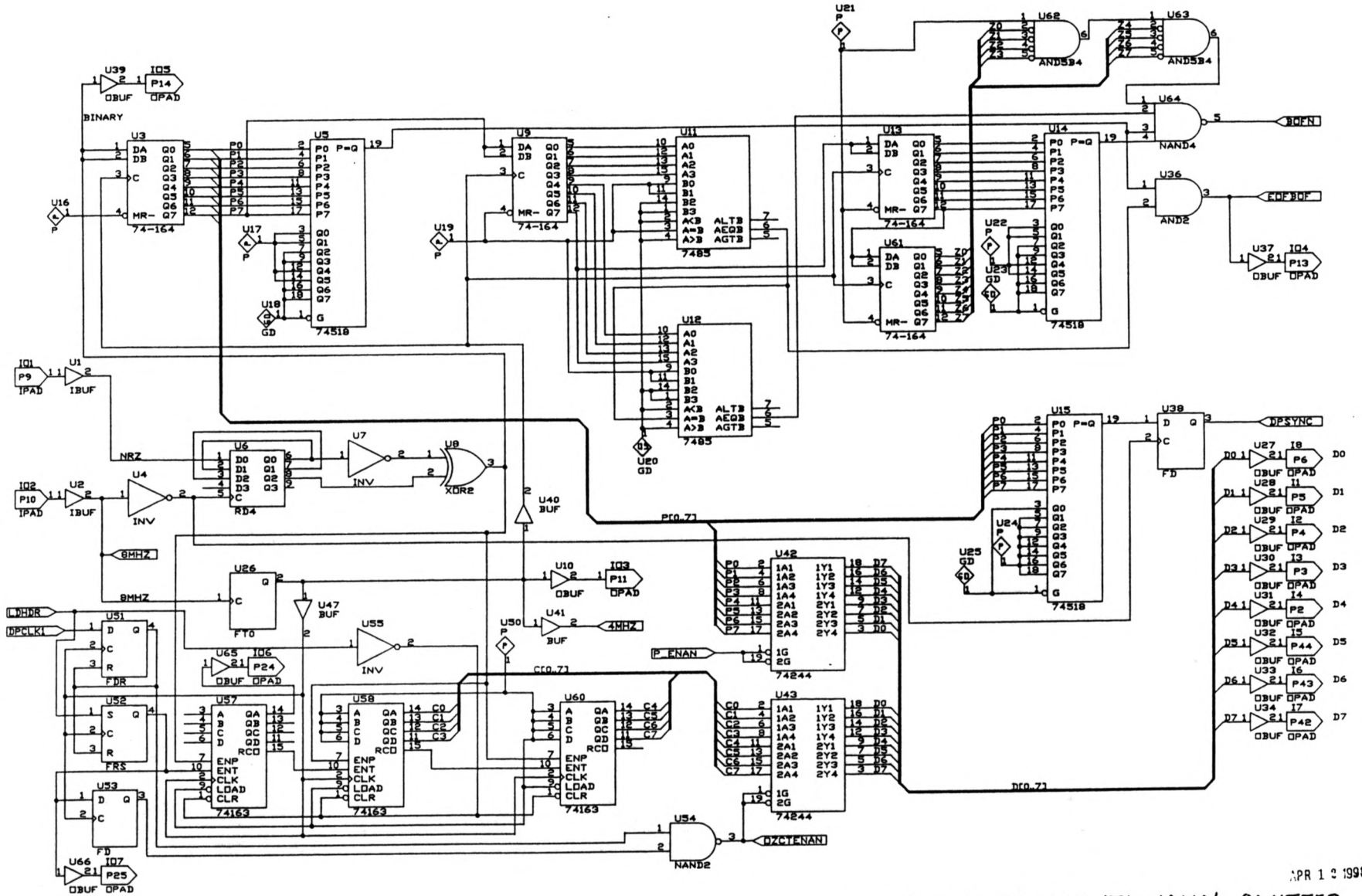


VIDEO DQA LOGIC

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		VLBA PROJECT	MKII CONTROLLER	NATIONAL RADIO ASTRONOMY OBSERVATORY SOCORRO, NEW MEXICO 87801
TOLERANCES : ANGLES ±	—	TITLE	MKII CONTROLLER SCHEMATIC DIAGRAM	DRAWN BY HARDEN — DATE 3-91
3 PLACE DECIMALS (.XXX) ±	—			DESIGNED BY DOOLEY — DATE 8-90
2 PLACE DECIMALS (.XX) ±	—			APPROVED BY — DATE
1 PLACE DECIMALS (.X) ±	—			
MATERIAL :	—			
FINISH :	—			
NEXT ASSEMBLY	DWG. TYPE	SHEET NUMBER 3 of 3	DRAWING NUMBER C54002S004	REV. A — SCALE —

ACAD : MKIISK-3
REF : ORCAD

REV	DATE	DRAWN BY	APPRV'D BY	DESCRIPTION

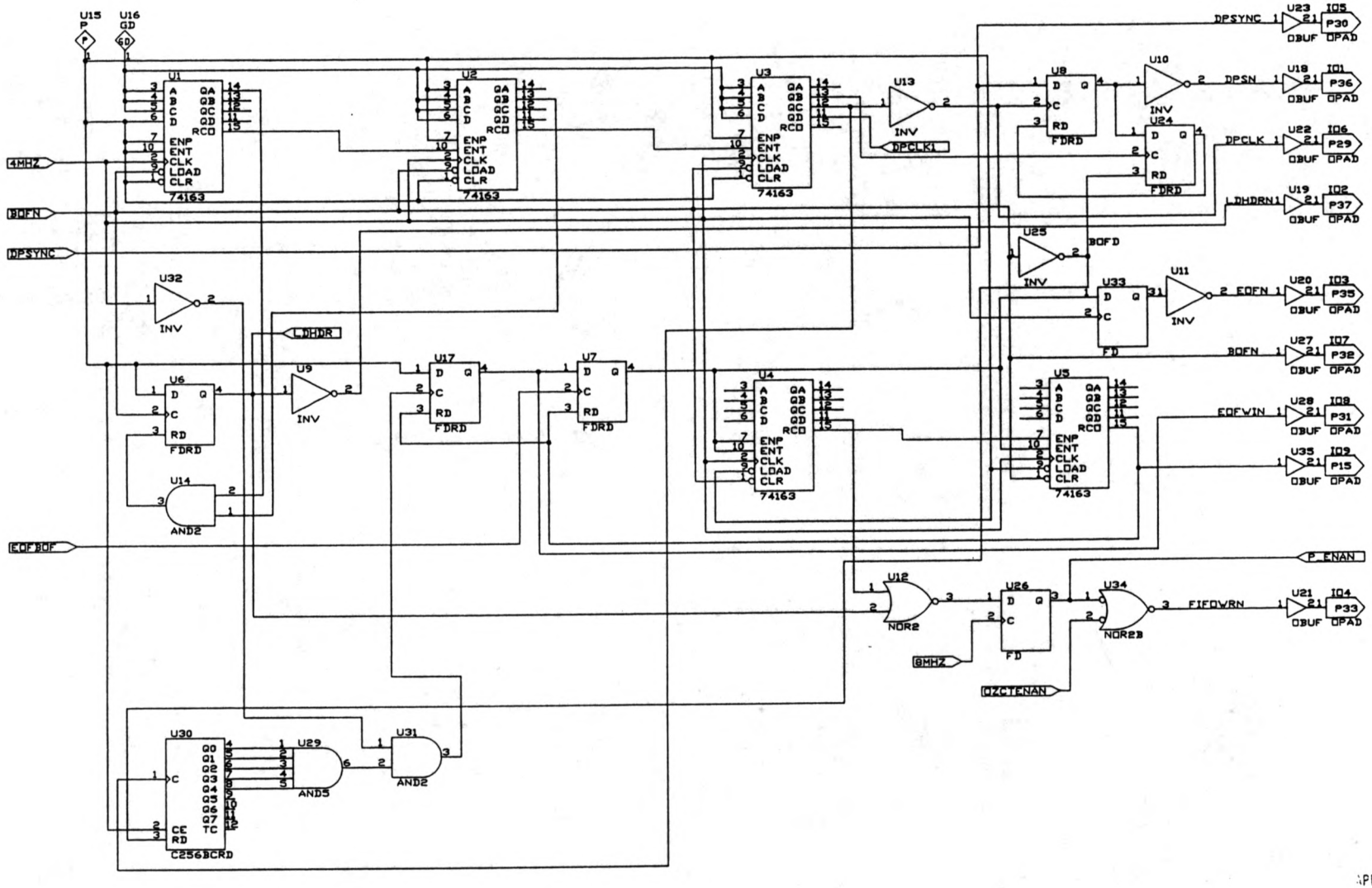


APR 1 2 1991

NOTE: SHT 1 of 3 INTENTIONALLY OMITTED

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES : ANGLES ± 3 PLACE DECIMALS (.000) ± 2 PLACE DECIMALS (.00) ± 1 PLACE DECIMALS (.0) ± MATERIAL : FINISH :	PROJECT: VLBA MKII	NATIONAL RADIO ASTRONOMY OBSERVATORY SOCORRO, NEW MEXICO 87801	
	TITLE: MKII VIDEO DECODER SCHEMATIC DIAGRAM	DRAWN BY DOOLEY	DATE 4-91
		DESIGNED BY DOOLEY	DATE 4-91
		APPROVED BY	DATE
NEXT ASSY [redacted] TYPE [redacted]		SHEET NUMBER 2 of 3	DRAWING C54002S008 REV. — SCALE —

ACAD : 3030V-2
[redacted] REF [redacted] CAD



APR 1 2 1991

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES

TOLERANCES : ANGLES ±

3 PLACE DECIMALS (.000) ±

2 PLACE DECIMALS (.00) ±

1 PLACE DECIMALS (.0) ±

MATERIAL :

FINISH :

NEXT ASSEMBLY | DWG. TYPE

VLBA | MKII

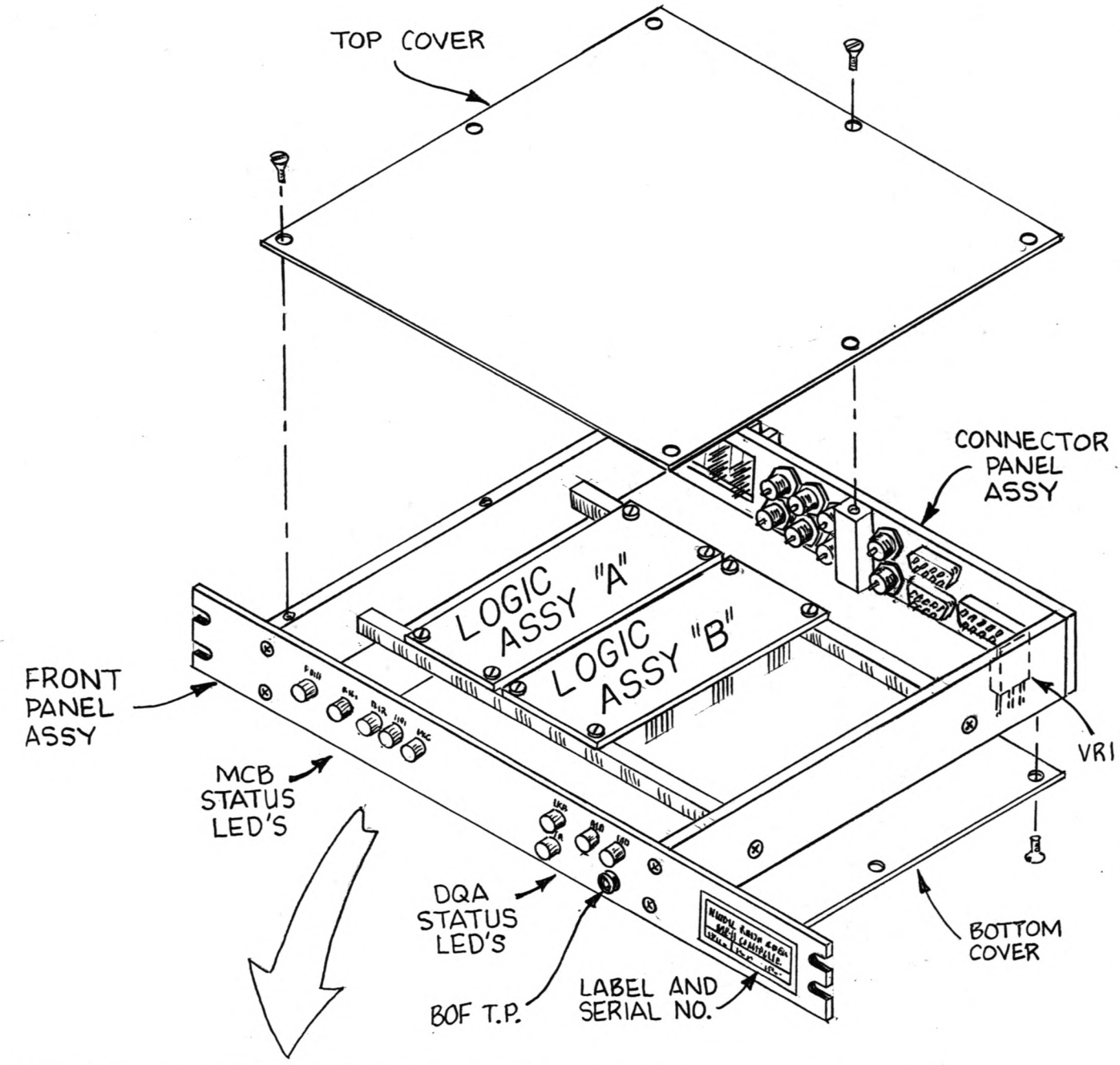
TITLE: MKII VIDEO DECODER SCHEMATIC DIAGRAM

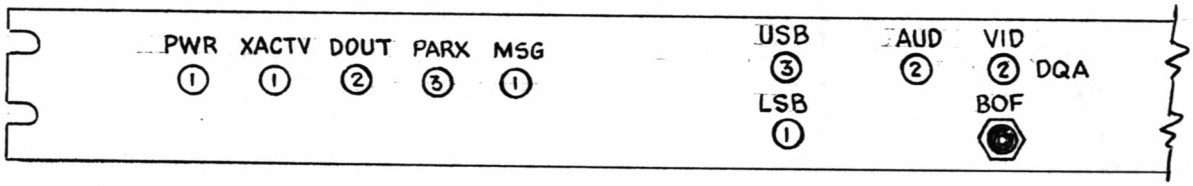
NATIONAL RADIO ASTRONOMY OBSERVATORY
SOCORRO, NEW MEXICO 87801

DRAWN BY DOOLEY	DATE 4-91	
DESIGNED BY DOOLEY	DATE 4-91	
APPROVED BY	DATE	
SHEET NUMBER 3 of 3	DRAWING NUMBER C54002S008	REV. / SCALE —

ACAD : 3030V-3
REF : ORCAD

FILE





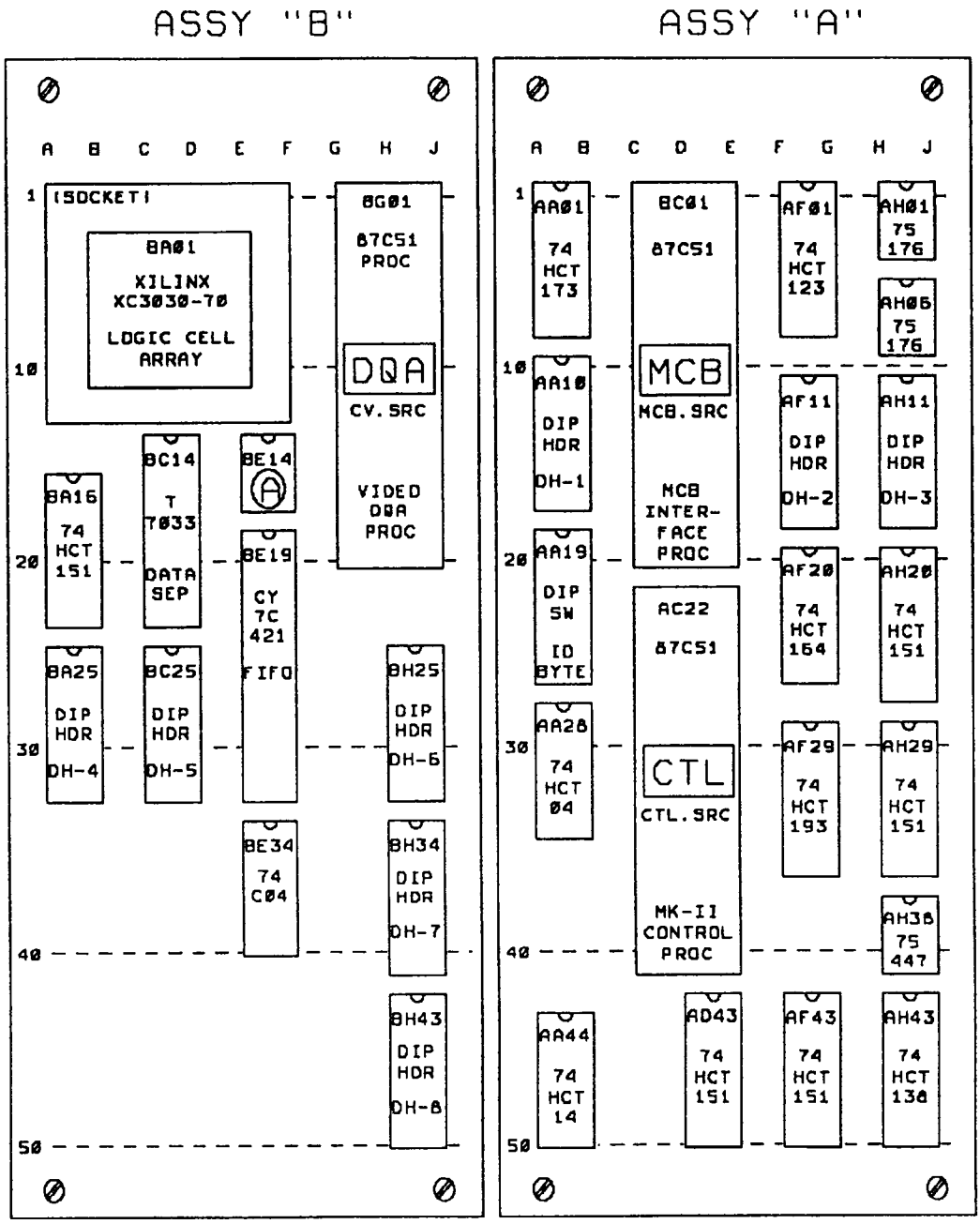
SHEET 1 OF 1

LED COLORS:

① Green
② Yellow/Amber
③ Red

PROJECT MK-II CONTROLLER	NATIONAL RADIO ASTRONOMY OBSERVATORY SOCORRO, NEW MEXICO
TITLE GENERAL ASSEMBLY	DRN BY P. HARDEN
DWG No. NONE	DATE 6-91

FILE: \MK-II\ASSYDWG.ORC
PLOTALL /P /S 0.9



NOTES

1. ITEM "A" AT BE14 IS XILINX PROM XC1798
2. BOM'S: LOGIC ASSY A: A540028003
   LOGIC ASSY B: A540028007
3. SEE BOM'S FOR DIP HEADER DETAILS AND COMPONENT VALUES
4. LOGIC DIAGRAM: 8540025004, 3 SHEETS

MK-II CONTROLLER		
LOGIC ASSEMBLIES A & B		
ASSEMBLY DWG		
SHT 1/1	08 FEB 1991	PMH
DWG NO.	A54002A002	

REVISIONS

REV	DATE	DRAWN BY	APPRV'D BY	DESCRIPTION
A	11-90	P. HARDEN		Added DQA Logic, Logic Assy B, new rear panel
B	5-91	P. HARDEN		Added front panel activity LED's for Audio, BOF, LSB/USB

NOTES:

1. DISK: DRAW1 FILE: \MK-II\MK2-B001
2. LAST FILE UPDATE: 1 JUL 1991 pmh

NEXT ASSY	USED ON
TOP ASSEMBLY DWG	C54002A001
B.O.M.	A54002B001
SCHEMATIC	
WIRE LIST	
LOGIC - ASSY "A"	C54002L001
LOGIC - ASSY "B"	

DRAWN BY Paul Harden	DATE SEPT 1990	
DESIGNED BY Phil Dooley	DATE AUG 1990	
APPROVED BY	DATE	

COVER SHEET

NATIONAL RADIO ASTRONOMY OBSERVATORY
SOCORRO, NEW MEXICO

VLBA

PROJECT MK-II CONTROLLER

TITLE TOP ASSEMBLY BILL OF MATERIALS

DWG NO A54002B001

PROJECT MK-II CONTROLLER

NATIONAL RADIO ASTRONOMY OBSERVATORY
• BILL OF MATERIALS •

BOM# A54002B001 NAME TOP ASSY BOM ASSY DWG C54002A001 SCHEMATIC

ITEM	REF DESIG	MANUFACTURER	MFR PART NO.	DESCRIPTION	QTY	
1	REF		C54002A001	TOP ASSEMBLY DRAWING		
2						
3		NRAO	B54002M001	SIDE SUPPORT RAILS	2	
4		NRAO	B54002M002	REAR PANEL CONN. PLATE	1	
5		NRAO	B54002M003	LOGIC ASSY MTG. RAILS	2	
6		NRAO	B54002M004	COVER PLATES, TOP & BOTTOM	2	
7		BUD	PA-1101 GRAY	FRONT PANEL, 1.75x19" ALUM	1	
8	ASSY "A"	NRAO	C54002A002	LOGIC BD. "A"- CTRLR LOGIC	1	
9	ASSY "B"	NRAO	C54002A004	LOGIC BD. "B"- DQA LOGIC	1	
10	P1-P6	CW INDUSTRIES	CHA-10G	CONN., 10-PIN WW	6	
11	(For J1-J6)	CW INDUSTRIES	CKR-10T	PLUG, 10-PIN RIBBON	6	
12	(For J1-J6)	OKI	R000-10	CABLE, RIBBON 10-COND	40ft	
13	P7, P8	CINCH	DB9S-F179	CONN., 9-PIN D-TYPE WW PINS	2	
14	P9	CINCH	DB25P-F179	CONN., 25-PIN D-TYPE WW PINS	1	
15	J10-J17	KING CONNECTORS	UG-1094/U	CONN., BNC PANEL MOUNT	8	
16	(For J9/P9)	CW INDUSTRIES	C7MFT-2506G	CABLE ASSY, 25-PIN M/F	1	
17	VR1	NATIONAL	7805	VOLT. REG., 5V TO-220	1	
18			6-32x3/8 FH	MACH. SCREW, FLAT HD	20	
19			6-32x3/8	MACH. SCREW, PAN HD	12	
20			2-56x1/2	MACH. SCREW, PAN HD	12	
21			2-56	HEX NUT, 2-56	12	
22			4-40x3/8	MACH. SCREW, PAN HD	14	
23			#4	WASHER, #4 LOCK	4	
24			#6	WASHER, #6 INT. TOOTH STAR	4	
25			#4	WASHER, #4 FLAT	8	

PREPARED BY Paul Harden 10-8-90

PROJECT MK-II CONTROLLER

NATIONAL RADIO ASTRONOMY OBSERVATORY
● BILL OF MATERIALS ●

BOM# A54002B001 NAME TOP ASSEMBLY BOM ASSY DWG SCHEMATIC

ITEM	REF DESIG	MANUFACTURER	MFR PART NO	DESCRIPTION	QTY	
26		EMS	50030R2	WIRE, #30 WW TWISTED PAIR	A/R	
27		NRAO	C13720M17C	LOGIC ASSY INSUL. SPACER	1	
28		DIALIGHT	521-9250	LED, STD. SIZE, GREEN	6	
29		DIALIGHT	521-9248	LED, STD. SIZE, AMBER	2	
30		DIALIGHT	521-9246	LED, STD. SIZE, RED	1	
31		DIALIGHT	515-0004	LED, MTG. CLIP	4	
32		NRAO	A54002M006	BRACKET, MTG. BLOCK	1	
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						

PREPARED BY Paul Harden

REVISIONS

REV	DATE	DRAWN BY	APPRV'D BY	DESCRIPTION
A	11-90	P. Harden		Incorporated DQA Logic; ADDED Logic Assy Bd. B (Obsoletes NRAO A54002B007, Orig. DQA BOM)
B	5-91	P. Harden		Added activity LED's Audio, BOF, LSB/USB, BOF test point

NOTES:

1. FILE: \MK2BOM\ MK2-B002.TSI
2. FILE LAST UPDATED: 1 JUL 1991

NEXT ASSY	USED ON
TOP ASSEMBLY DWG	B54002A001
TOP B.O.M.	A54002B001
ASSY DWG	
B.O.M.	A54002B002
SCHEMATIC	C54002S004
WIRE LIST	

DRAWN BY: Paul Harden	DATE: SEP 1990	
DESIGNED BY: Phil Dooley	DATE: AUG 1990	
APPROVED BY	DATE	

COVER SHEET

NATIONAL RADIO ASTRONOMY OBSERVATORY
SOCORRO, NEW MEXICO

VLBA

PROJECT: MK-II CONTROLLER

TITLE: LOGIC ASSEMBLY "A" and "B" MAIN CONTROLLER & DQA LOGIC BD's

DWG NO: A54002B002

PROJECT MK-II CONTROLLER

NATIONAL RADIO ASTRONOMY OBSERVATORY
• BILL OF MATERIALS •

BOM# A54002B002 NAME LOGIC ASSY "A" ASSY DWG _____ SCHEMATIC C54002S004

ITEM	REF DESIG	MANUFACTURER	MFR PART NO.	DESCRIPTION	QTY	
1	REF	NRAO	A54002A002	ASSY DWG - LOGIC BD. "A"		
2	REF	NRAO	A54002A007	ASSY DWG - LOGIC BD. "B"		
3	REF	NRAO	A54002W001	WIRE LIST, MK-II Logic Assy		
4	BD. A and B	EMC	9406276	WIRE WRAP BOARD (Universal)	2	
5	AC01, AC22, BG01	INTEL	87C51 FA	IC, CMOS MICROPROCESSOR	3	
6	BA01	XLINX	XC3030-70	IC, LOGIC CELL ARRAY, 44 PIN	1	
7	BE14	XLINX	XC1736	IC, SERIAL PROM (for XC3030)	1	
8	BE19	Advanced Micro Devices (AMD)	CY7C421	IC, CMOS FIFO SHIFT REG.	1	
9	BC14	AT+T	T7033 PC	IC, DATA SEP/CLOCK RECOVERY	1	
10	AH01, AH06	TI or equiv.	75176 BP	IC, BI-DIR. LINE RCVR/XMTR	2	
11	AH38		75447	IC, O.C. RELAY DRIVER	1	
12	AA28, BC34	RCA, Harris, or equiv.	74C04	IC, CMOS HEX INVERTER	2	
13	AA44		74HCT14	IC, CMOS SCHMIDT INVERTER	1	
14	AA36	NOTE: Added REV. B	74HCT74	I CMOS DUAL D FLIP-FLOP	1	
15	AF01		74HCT123	IC, CMOS DUAL M.V.	1	
16	AH43		74HCT138	IC, CMOS 3L-8L DATA SEL.	1	
17	AH20, AH29, AD43 AF43, BA13		74HCT151	IC, CMOS 8L-1L MUX SEL	5	
18	AF20		74HCT164	IC, CMOS SI/PO SHIFT REG	1	
19	AA01		74HCT173	IC, CMOS 4-BIT LATCH	1	
20	AF29		74HCT193	IC, CMOS UP/DN COUNTER	1	
21	(FOR ITEM 6)	METHODE	300-715-203G1	SOCKET, 44-PIN, PLCC TO DIP	1	
22	AA19	CTS	206-8	DIP SWITCH, 16 PIN, 8xSPST	1	
23	Dip Hdr Assy's	AUGAT	616-AG1	DIP Header Mtg. Platform, 16-pin	8	
24	BG23-BG24	AUGAT	929950-00	Shorting Jumper, 0.1 in. spacing	1	
25						

PREPARED BY Paul Harden 10-8-90

PROJECT MK-II CONTROLLER

NATIONAL RADIO ASTRONOMY OBSERVATORY
• BILL OF MATERIALS •

BOM# A54002B002 NAME LOGIC ASSY BOARD "A" ASSY DWG _____ SCHEMATIC C54002S004

ITEM	REF DESIG	MANUFACTURER	MFR PART NO.	DESCRIPTION	QTY	
26	■ DH	U.S. CRYSTAL or equiv.	11.0592 mHz	CRYSTAL, HC-18	1	
27	■ DH	U.S. CRYSTAL or equiv.	3.57 mHz	CRYSTAL, HC-18	1	
28	■ DH	A-B or equiv.	RC04-100J	RESISTOR, 10 ohms 1/4 W	1	
29	■ DH		RC04-750J	RESISTOR, 75 ohms	8	
30	■ DH		RC04-471J	RESISTOR, 470 ohms	9	
31	■ DH		RC04-102J	RESISTOR, 1K	8	
32	■ DH		RC04-103J	RESISTOR, 10K	5	
33	■ DH		RC04-203J	RESISTOR, 20K	1	
34	■ DH		RC04-393J	RESISTOR, 39K	2	
35	■ DH		RC04-104J	RESISTOR, 100K	8	
36	■ DH	SPRAGUE or equiv.	30GAQ39	CAP., 39pF Ceramic Disk	2	
37	■ DH		Z5U104M	CAP., 0.1uF MONO	10	
38	■ DH		Z5U225M	CAP., 2.2uF MONO	2	
39	■ DH		513D106M	CAP., 10uF 50V MINI-ELECTRO	2	
40						
41						
42						
43						
44						
45						

■ DH = Discrete components part of Dip Header Assemblies. See next sheet.

PREPARED BY Paul Harden 10-8-90

PROJECT MK-II CONTROLLER

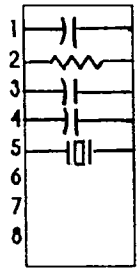
NATIONAL RADIO ASTRONOMY OBSERVATORY
● BILL OF MATERIALS ●

SHEET 4 OF 4

BOM# A54002B002 NAME LOGIC ASSY BD ASSYDWG SCHEMATIC 54002S004

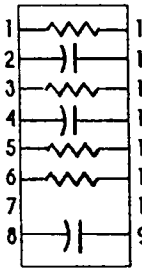
DIP HEADER DETAILS

DH-1
LOC: AA10



10uF
20K
39pF
39pF
11.0592 mHz

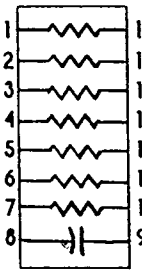
DH-2
LOC: AF11



39K
2.2uF
39K
2.2uF
470
470

0.1uF

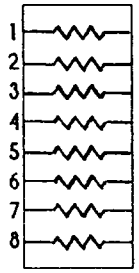
DH-3
LOC: AH11



470 OHMS

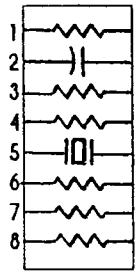
10uF

DH-4
LOC: BA25



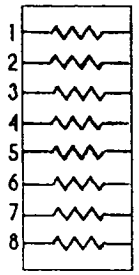
75 OHMS

DH-5
LOC: BC25



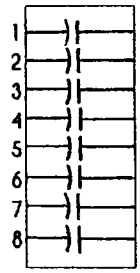
10 Ohms
0.1uF
10K
10K
3.57 mHz
10K
10K
10K

DH-6
LOC: BH25



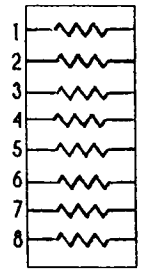
100K

DH-7
LOC: BH34



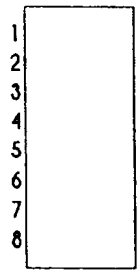
0.1uF

DH-8
LOC: BH43



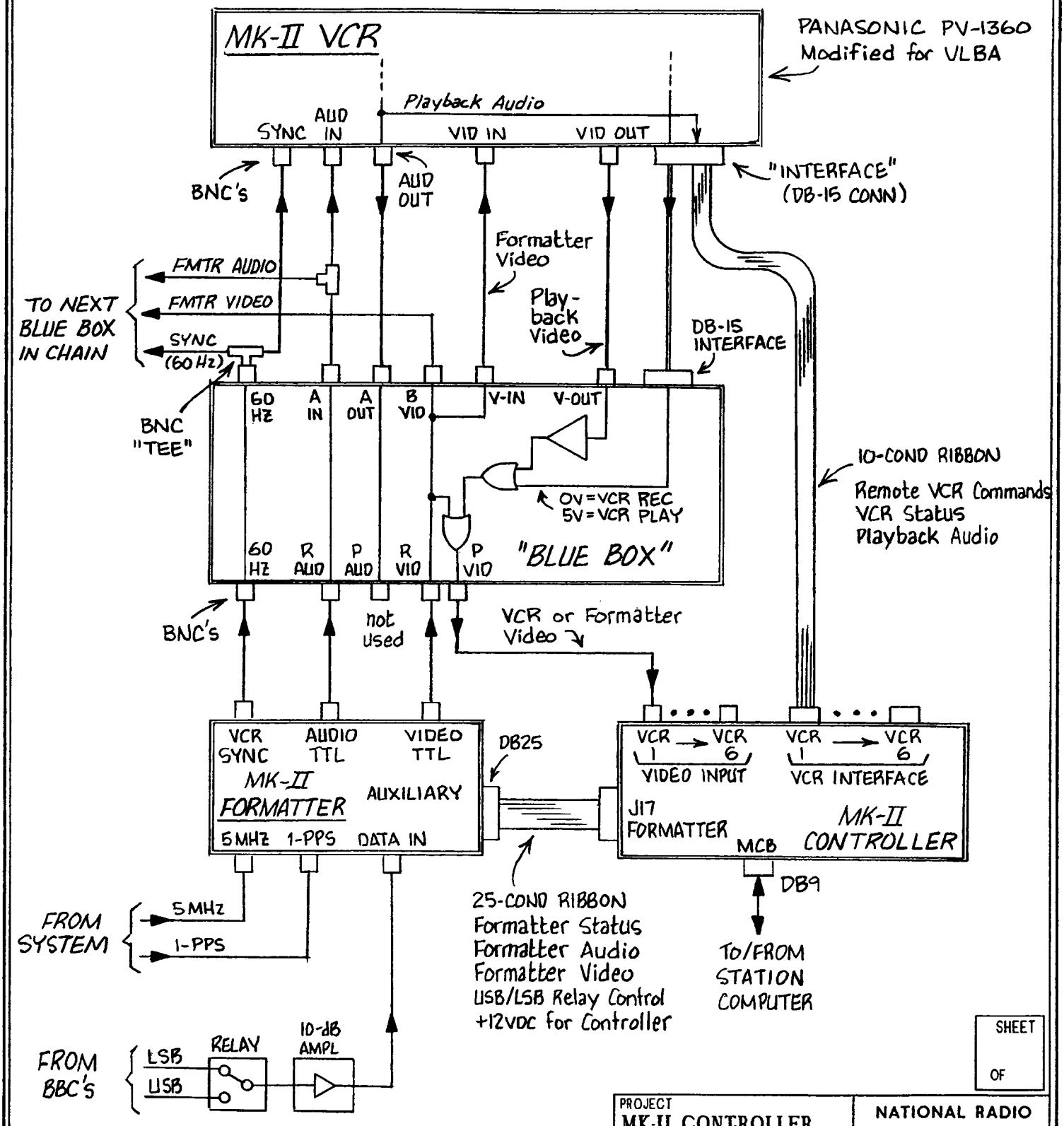
1K

DH-9
Spare/not assigned



PREPARED BY P. Harden

FILE



SHEET
OF

PROJECT: MK-II CONTROLLER	NATIONAL RADIO ASTRONOMY OBSERVATORY SOCORRO, NEW MEXICO
TITLE: INTERCONNECTING CABLING DIAG.	DRN BY: PAUL HARDEN
DWG No.	DATE: MAR 1991

87C51FA
CHMOS SINGLE-CHIP 8-BIT MICROCONTROLLER
8K BYTES USER PROGRAMMABLE EPROM

87C51FA—3.5 MHz to 12 MHz, V_{CC} 5V ± 10%
87C51FA-1—3.5 MHz to 16 MHz, V_{CC} 5V ± 10%

- High Performance CHMOS EPROM
- Power Control Modes
- Three 16-Bit Timer/Counters
- Programmable Counter Array with:
  - — High Speed Output,
  - — Compare/Capture,
  - — Pulse Width Modulator,
  - — Watchdog Timer capabilities
- Up/Down Timer/Counter
- Two Level Program Lock System
- 8K On-Chip EPROM
- 256 Bytes of On-Chip Data RAM
- Quick Pulse Programming™ Algorithm
- Boolean Processor
- 32 Programmable I/O Lines
- 7 Interrupt Sources
- Programmable Serial Channel with:
  - — Framing Error Detection
  - — Automatic Address Recognition
- TTL Compatible Logic Levels
- 64K External Program Memory Space
- 64K External Data Memory Space
- MCS®-51 Compatible Instruction Set
- Power Saving Idle and Power Down Modes
- ONCE™ (On-Circuit Emulation) Mode

MEMORY ORGANIZATION

PROGRAM MEMORY: Up to 8K bytes of the program memory can reside in the on-chip EPROM. In addition the device can address up to 64K of program memory external to the chip.

DATA MEMORY: This microcontroller has a 256 x 8 on-chip RAM. In addition it can address up to 64K bytes of external data memory.

The Intel 87C51FA is a single-chip control oriented microcontroller which is fabricated on Intel's reliable CHMOS II-E technology. Being a member of the MCS®-51 family, the 87C51FA uses the same powerful instruction set, has the same architecture, and is pin for pin compatible with the existing MCS-51 products. The 87C51FA is an enhanced version of the 87C51. It's added features make it an even more powerful microcontroller for applications that require Pulse Width Modulation, High Speed I/O, and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multi-processor communications.

October 1989
Order Number: 270258-004

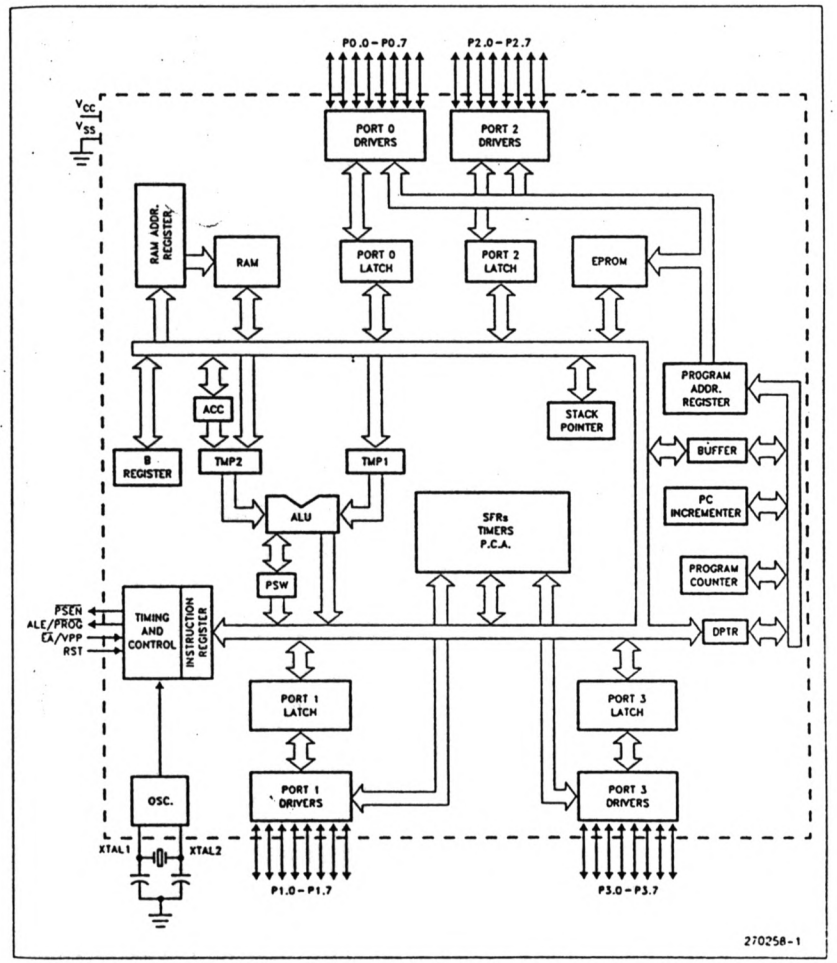


Figure 1. 87C51FA Block Diagram

PACKAGES

Part	Prefix	Package Type
87C51FA	P	40-Pin Plastic DIP
	D	40-Pin CERDIP
	N	44-PIN PLCC

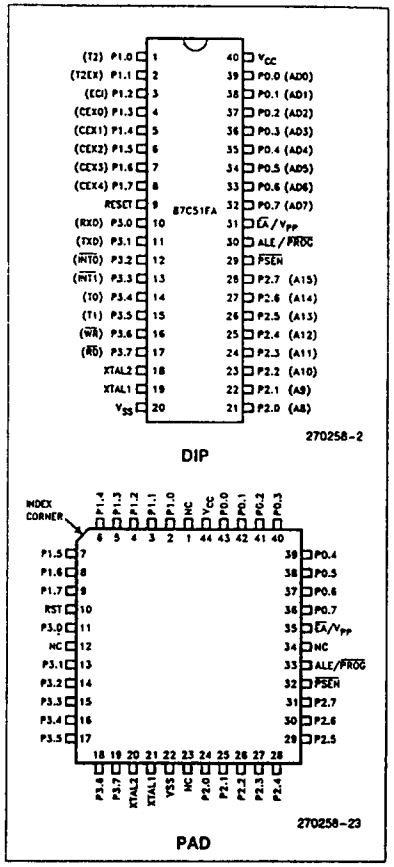


Figure 2. Pin Connections

PIN DESCRIPTIONS

V_{CC}: Supply voltage.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit, open drain, bidirectional I/O port. As an output port each pin can sink several LS TTL inputs. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting1's, and can source and sink several LS TTL inputs.

Port 0 also receives the code bytes during EPROM programming, and outputs the code bytes during program verification. External pullup resistors are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can drive LS TTL inputs. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

In addition, Port 1 serves the functions of the following special features of the 87C51FA:

Port Pin	Alternate Function
P1.0	T2 (External Count Input to Timer/ Counter 2)
P1.1	T2EX (Timer/Counter 2 Capture/ Reload Trigger and Direction Control)
P1.2	ECI (External Count Input to the PCA)
P1.3	CEX0 (External I/O for Compare/ Capture Module 0)
P1.4	CEX1 (External I/O for Compare/ Capture Module 1)
P1.5	CEX2 (External I/O for Compare/ Capture Module 2)
P1.6	CEX3 (External I/O for Compare/ Capture Module 3)
P1.7	CEX4 (External I/O for Compare/ Capture Module 4)

Port 1 receives the low-order address bytes during EPROM programming and verifying.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can drive LS TTL inputs. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

MCS®-51 INSTRUCTION SET

Table 10. 8051 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings(1)

Instruction	Flag C	OV	AC	Instruction	Flag C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C,bit	X		
DA	X			ORL C,bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

(1)Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Note on instruction set and addressing modes:

Rn — Register R7–R0 of the currently selected Register Bank.

direct — 8-bit internal data location's address. This could be an Internal Data RAM location (0–127) or a SFR [i.e., I/O port, control register, status register, etc. (128–255)].

@Ri — 8-bit internal data RAM location (0–255) addressed indirectly through register R1 or R0.

#data — 8-bit constant included in instruction.

#data 16 — 16-bit constant included in instruction.

addr 16 — 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.

addr 11 — 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.

rel — Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to +127 bytes relative to first byte of the following instruction.

bit — Direct Addressed bit in Internal Data RAM or Special Function Register.

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with Carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with Carry	1	12
ADDC	A,#data	Add immediate data to Acc with Carry	2	12
SUBB	A,Rn	Subtract Register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from ACC with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment direct RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS (Continued)				
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A & B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
LOGICAL OPERATIONS (Continued)				
RL	A	Rotate Accumulator Left	1	12
RLC	A	Rotate Accumulator Left through the Carry	1	12
RR	A	Rotate Accumulator Right	1	12
RRC	A	Rotate Accumulator Right through the Carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	Rn,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Oscillator Period
DATA TRANSFER (Continued)				
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX	A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX	@Ri,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order Digit indirect RAM with Acc	1	12
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear Carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set Carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement Carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to CARRY	2	24
ANL	C,/bit	AND complement of direct bit to Carry	2	24
ORL	C,bit	OR direct bit to Carry	2	24
ORL	C,/bit	OR complement of direct bit to Carry	2	24
MOV	C,bit	Move direct bit to Carry	2	12
MOV	bit,C	Move Carry to direct bit	2	24
JC	rel	Jump if Carry is set	2	24
JNC	rel	Jump if Carry not set	2	24
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute Subroutine Call	2	24
LCALL	addr16	Long Subroutine Call	3	24
RET		Return from Subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute Jump	2	24
LJMP	addr16	Long Jump	3	24
SJMP	rel	Short Jump (relative addr)	2	24

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)				
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is Zero	2	24
JNZ	rel	Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24
CJNE	Rn,#data,rel	Compare immediate to register and Jump if Not Equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ	Rn,rel	Decrement register and Jump if Not Zero	2	24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP		No Operation	1	12

XC3000
Logic Cell™ Array Family

Product Specification

FEATURES

- High Performance—50, 70 and 100 MHz Toggle Rates
- Second Generation User-Programmable Gate Array
  - I/O functions
  - Digital logic functions
  - Interconnections
- Flexible array architecture
  - Compatible arrays, 2000 to 9000 gate logic complexity
  - Extensive register and I/O capabilities
  - High fan-out signal distribution
  - Internal three-state bus capabilities
  - TTL or CMOS input thresholds
  - On-chip oscillator amplifier
- Standard product availability
  - Low power, CMOS, static memory technology
  - Performance equivalent to TTL SSI/MSI
  - 100% factory pre-tested
  - Selectable configuration modes
- Complete XACT™ development system
  - Schematic Capture
  - Automatic Place/Route
  - Logic and Timing Simulation
  - Design Editor
  - Library and User Macros
  - Timing Calculator
  - XACTOR In-Circuit Verifier
  - Standard PROM File Interface

DESCRIPTION

The CMOS XC3000 Logic Cell™ Array (LCA) family provides a group of high-performance, high-density, digital, integrated circuits. Their regular, extendable, flexible, user-programmable array architecture is composed of a configuration program store plus three types of configurable elements: a perimeter of I/O Blocks, a core array of Logic Blocks and resources for interconnection. The general structure of a Logic Cell Array is shown in Figure 1 on the next page. The XACT development system provides schematic capture and auto place-and-route for design entry. Logic and timing simulation, and in-circuit emulation are available as design verification alternatives. The design editor is used for interactive design optimizaton, and to compile the data pattern which represents the configuration program.

The Logic Cell Array's user logic functions and interconnections are determined by the configuration program data stored in internal static memory cells. The program can be loaded in any of several modes to accommodate various system requirements. The program data resides externally in an EEPROM, EPROM or ROM on the application circuit board, or on a floppy disk or hard disk. On-chip initialization logic provides for optional automatic loading of program data at power-up. Xilinx's companion XC1736 Serial Configuration PROM provides a very simple serial configuration program storage in a one-time-programmable eight-pin DIP.

Basic Array	Logic Capacity (usable gates)	Configurable Logic Blocks	User I/Os	Program Data (bits)
XC3020	2000	64	64	14779
XC3030	3000	100	80	22176
XC3042	4200	144	96	30784
XC3064	6400	224	120	46064
XC3090	9000	320	144	64160

The XC3000 Logic Cell Arrays are an enhanced family of Programmable Gate Arrays, which provide a variety of logic capacities, package styles, temperature ranges and speed grades.

ARCHITECTURE

The perimeter of configurable I/O Blocks (IOBs) provides a programmable interface between the internal logic array and the device package pins. The array of Configurable Logic Blocks (CLBs) performs user-specified logic functions. The interconnect resources are programmed to form networks, carrying logic signals among blocks, analogous to printed circuit board traces connecting MSI/SSI packages.

The blocks' logic functions are implemented by programmed look-up tables. Functional options are implemented by program-controlled multiplexers. Interconnecting networks between blocks are implemented with metal segments joined by program-controlled pass transistors. These functions of the Logic Cell Array are established by a configuration program which is loaded into an internal, distributed array of configuration memory cells. The configuration program is loaded into the Logic Cell Array at power-up and may be reloaded on command. The Logic Cell Array includes logic and control signals to implement automatic or passive configuration. Program data may be either bit serial or byte parallel. The XACT development system generates the configuration program bit-stream used to configure the Logic Cell Array. The memory loading process is independent of the user logic functions.

CONFIGURATION MEMORY

The static memory cell used for the configuration memory in the Logic Cell Array has been designed specifically for high reliability and noise immunity. Integrity of the LCA configuration memory based on this design is assured even under adverse conditions. Compared with other programming alternatives, static memory provides the best combination of high density, high performance, high reliability and comprehensive testability. As shown in Figure 2, the basic memory cell consists of two CMOS inverters plus a pass transistor used for writing and reading cell data. The cell is only written during configuration and only read during readback. During normal operation the cell provides continuous control and the pass transistor is "off" and does not affect cell stability. This is quite different from the operation of conventional memory devices, in which the cells are frequently read and re-written.

The memory cell outputs Q and $\overline{Q}$ use full Ground and Vcc levels and provide continuous, direct control. The additional capacitive load together with the absence of address decoding and sense amplifiers provide high stability to the

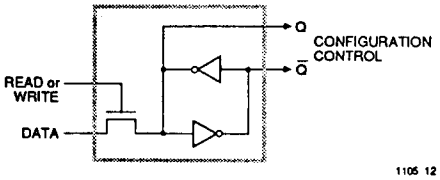


Figure 2. A static configuration memory cell is loaded with one bit of configuration program and controls one program selection in the Logic Cell Array.

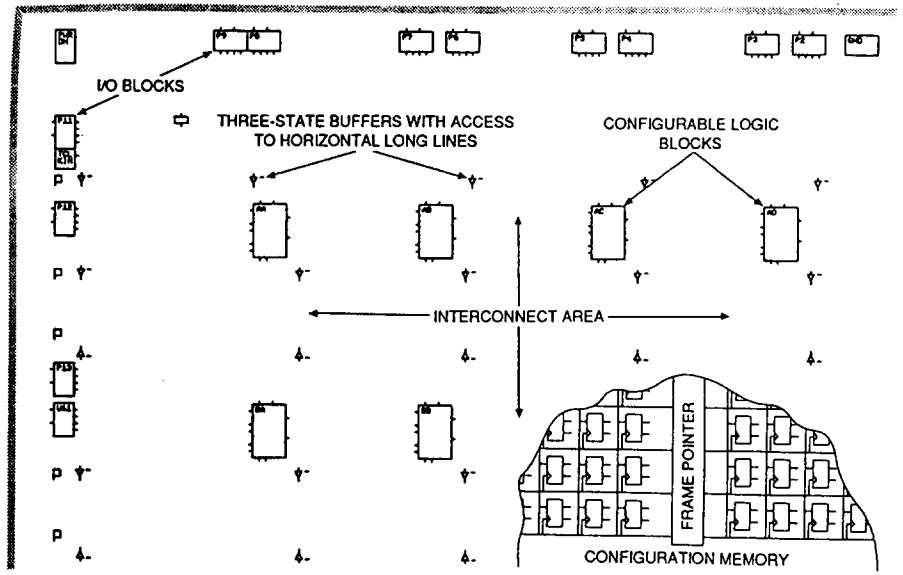


Figure 1. The structure of the Logic Cell Array consists of a perimeter of programmable I/O blocks, a core of configurable logic blocks and their interconnect resources. These are all controlled by the distributed array of configuration program memory cells.

PIN DESCRIPTIONS

1. Permanently Dedicated Pins.

Vcc
Two to eight (depending on package type) connections to the nominal +5 V supply voltage. All must be connected.

GND
Two to eight (depending on package type) connections to ground. All must be connected.

PWRDWN
A LOW on this CMOS compatible input stops all internal activity to minimize Vcc power, and puts all output buffers in a high impedance state, but configuration is retained. When the PWRDWN pin returns HIGH, the device returns to operation with the same sequence of buffer enable and DONE/PROGRAM as at the completion of configuration. All internal storage elements are reset. If not used, PWRDWN must be tied to Vcc.

RESET
This is an active low input which has three functons.

Prior to the start of configuration, a LOW input will delay the start of the configuration process. An internal circuit senses the application of power and begins a minimal time-out cycle. When the time-out and RESET are complete, the levels of the "M" lines are sampled and configuration begins.

If RESET is asserted during a configuration, the LCA is re-initialized and will restart the configuration at the termination of RESET.

If RESET is asserted after configuration is complete it will provide an asynchronous reset of all IOB and CLB storage elements of the LCA.

CCLK
During configuration, Configuration Clock is an output of an LCA in Master mode or Peripheral mode. LCAs in Slave mode use it as a clock input. During a Readback operation it is a clock input for the configuration data being shifted out.

DONE
The DONE output is configurable as open drain with or without an internal pull-up resistor. At the completion of configuration, the circuitry of the LCA becomes active in a synchronous order, and DONE may be programmed to occur one cycle before or after that.

PROG
Once configuration is done, a HIGH to LOW transition of this pin will cause an initialization of the LCA and start a reconfiguration.

M0
As Mode 0, this input and M1, M2 are sampled before the start of configuration to establish the configuration mode to be used.

RTRIG
As a Read Trigger, a LOW-to-HIGH input transition, after configuration is complete, will initiate a Readback of configuration and storage element data by CCLK. This operation may be limited to a single request, or be inhibited altogether, by selecting the appropriate readback option when generating the bit stream.

M1
As Mode 1, this input and M0, M2 are sampled before the start of configuration to establish the configuration mode to be used. If Readback is to be used, a 5 KΩ resistor should be used to define mode level inputs.

RDATA
As an active low Read Data, after configuration is complete, this pin is the output of the readback data.

2. User I/O Pins that can have special functions.

M2
As Mode 2 this input has a passive pullup during configuration. Together with M0 and M1 it is sampled before the start of configuration to establish the configuration mode to be used. After configuration this pin becomes a user programmable I/O pin.

HDC
High During Configuration is held at a HIGH level by the LCA until after configuration. It is available as a control output indicating that configuration is not yet completed. After configuration this pin is a user I/O pin.

LDC
Low During Configuration is held at a LOW level by the LCA until after configuration. It is available as a control output indicating that configuration is not yet completed. It is particularly useful in Master mode as a LOW enable for an EPROM. After configuration this pin is a user I/O pin. If used as a LOW EPROM enable, it must be programmed as a HIGH after configuration.

INIT
This is an active low open drain output which is held LOW during the power stabilization and internal clearing of the configuration memory. It can be used to indicate status to a configuring microprocessor or, as a wired AND of several slave mode devices, a hold-off signal for a master mode device. After configuration this pin becomes a user programmable I/O pin.

BCLKIN
This is a direct CMOS level input to the alternate clock buffer (Auxiliary Buffer) in the lower right corner.

XTL1
This user I/O pin can be used to operate as the output of an amplifier driving an external crystal and bias circuitry.

XTL2
This user I/O pin can be used as the input of an amplifier connected to an external crystal and bias circuitry. The I/O Block is left unconfigured. The oscillator configuration is activated by routing a net from the oscillator buffer symbol output and by the MAKEBITS program.

CS0, CS1, CS2, WS
These four inputs represent a set of signals, three active low and one active high, which are used in the Peripheral mode to control configuration data entry. The assertion of all 4 generates a write to the internal data buffer. The removal of any assertion, clocks in the D0–D7 data present.

RCLK
During Master parallel mode configuration RCLK represents a "read" of an external dynamic memory device (normally not used).

RDY/BUSY
During Peripheral parallel mode configuration this pin indicates when the chip is ready for another byte of data to be written to it. After configuration is complete, this pin becomes a user programmed I/O pin.

D0–D7
This set of 8 pins represent the parallel configuration byte for the parallel Master and Peripheral modes. After configuration is complete they are user programmed I/O pin.

A0–A15
This set of 16 pins present an address output for a configuration EPROM during Master parallel mode. After configuration is complete they are user programmed I/O pin.

DIN
This user I/O pin is used as serial Data Input during Slave or Master Serial configuration. This pin is Data 0 input in Master or Peripheral configuration mode.

DOUT
This user I/O pin is used during configuration to output serial configuration data for daisy-chained slaves' Data In.

TCLKIN
This is a direct CMOS level input to the global clock buffer.

3. Unrestricted User I/O Pins.

I/O
A pin which may be programmed by the user to be Input and/or Output pin following configuration. Some of these pins present a high impedance pull-up (see next page) or perform other functions before configuration is complete (see above).

XC1736 Serial Configuration PROM

Product Specification

FEATURES

- One-Time Programmable (OTP) 36,288 x 1 bit serial memory designed to store configuration programs for Programmable Gate Arrays
- Simple interface to a XILINX Logic Cell™ Array (LCA) requires only two I/O pins
- Daisy chain configuration support for multiple LCAs
- Cascadable to provide more memory for additional configurations or future higher-density arrays
- Storage for multiple configurations for a single Logic Cell Array
- Low power CMOS EPROM process
- Space-efficient, 8-pin plastic/ceramic DIP package
- PC-based Xilinx programmer for development. Production programming support from leading programmer manufacturers

DESCRIPTION

The XC1736 Serial Configuration PROM (SCP) provides an easy-to-use, cost-effective configuration memory for the Xilinx family of programmable gate arrays. Packaged in an economical 8-pin plastic DIP package, the XC1736 uses a simple serial access procedure to configure one or more Logic Cell Arrays (LCAs). The 36,288 x 1 organization of the configuration PROM supplies enough memory to configure three XC2064's, two XC2018's, two XC3020s, one XC3030 or one XC3042. Multiple Serial Configuration PROMs can be cascaded to provide a larger memory for higher density arrays. Multiple configurations for a single LCA can also be loaded from the XC1736.

The XC1736 can be programmed with the PC-based Xilinx XC-DS112 Configuration PROM Programmer or with programmers from other manufacturers. The Logic Cell Array design file is first compiled into a standard HEX format with the XC-DS21 Development System. It can then be transferred to the XC-DS112 programmer connected to the serial port.

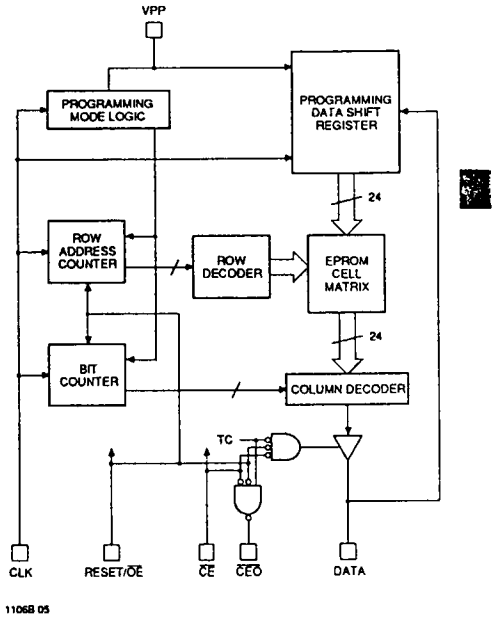
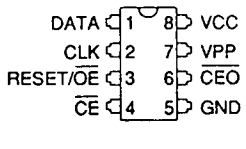


Figure 1. XC1736 Block Diagram



XC1736 Pin Assignments

Table 1. XC1736 Pin Assignments

Pin	Name	I/O	Description
1	DATA	O	Three-state DATA output for reading. Input/Output pin for programming.
2	CLK	I	Clock input. Used to increment the internal address and bit counters for reading and programming.
3	RESET/ $\overline{OE}$	I	Output Enable input. A LOW level on both the $\overline{CE}$ and $\overline{OE}$ inputs enables the data output driver. A HIGH level on RESET/$\overline{OE}$ resets both the address and bit counters.
4	$\overline{CE}$	I	Chip Enable input. A LOW level on both $\overline{CE}$ and $\overline{OE}$ enables the data output driver. A HIGH level on $\overline{CE}$ disables both the address and bit counters and forces the device into a low power mode. Used for device selection.
5	GND		Ground pin.
6	$\overline{CEO}$	O	Chip Enable Out output. This signal is asserted LOW on the clock cycle following the last bit read from the memory. It will stay LOW as long as $\overline{CE}$ and $\overline{OE}$ are both LOW. It will follow $\overline{CE}$, but if $\overline{OE}$ goes HIGH, $\overline{CEO}$ will stay HIGH until the entire PROM is read again.
7	V_{PP}		Programming Voltage Supply. Used to enter programming mode (+6V) and to program the memory (+21V) Must be connected directly to Vcc for normal read operation. No overshoot above +22V permitted.
8	Vcc		+5 volt power supply input.

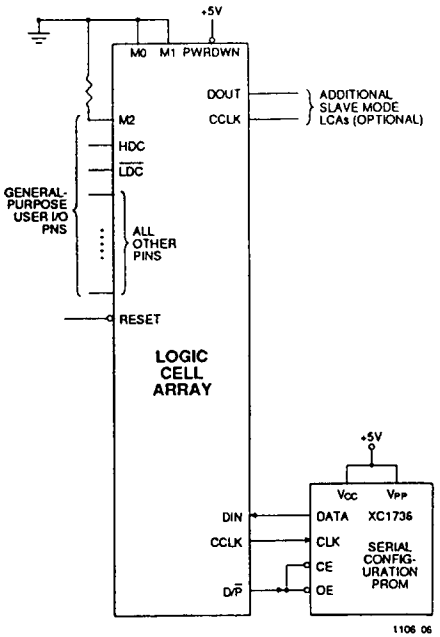


Figure 2. Master Serial Mode Configuration

T7033 Clock Recovery Circuit

Features

- Pin-programmable for 1-MHz to 50-MHz operation
- Fiber and wire applications
- Single 5 V supply
- Only one external component required: 3.58-MHz crystal

Description

The T7033 Clock Recovery Circuit integrated circuit operates over a 1-MHz to 50-MHz frequency range and provides clock recovery and data retiming. This device accepts TTL-NRZ data from a receiver (optical or electrical), recovers the clock, and retimes the data to the recovered clock. The inputs and outputs are TTL-compatible and the circuit requires a single 5 V supply. The T7033 Clock Recovery Circuit is manufactured using CMOS technology and it is available in a 300-mil, 20-pin plastic DIP. The device is intended for applications where an internal descrambler is not required.

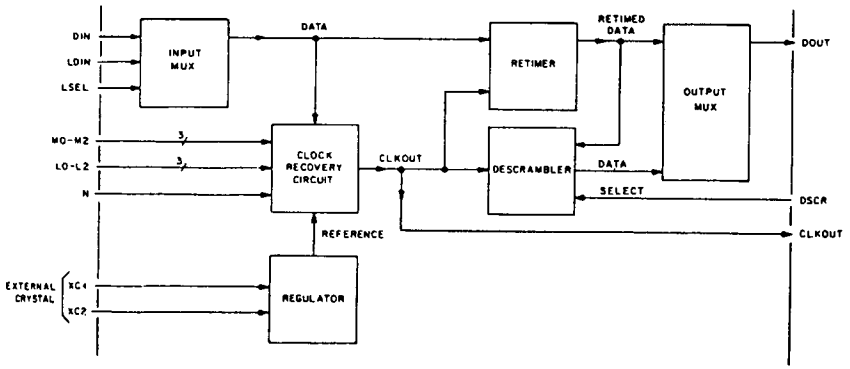


Figure 1. Block Diagram

Table 1. Pin Descriptions (Continued)

Pin	Symbol	Type	Name/Function
15	QVDD	—	**Quiet VDD 5 V Supply.** Extra care may be required when filtering this voltage. The required filtering can be supplied with a 10-Ω resistor connected to VDD (pin 19) and a 0.1-μF capacitor connected to ground.
16	VSS	—	**Ground.** 0 V.
17	TESTF	I	**Test F Pin.** Used for manufacturing test purposes; should be tied to 5 V for normal operation.
18	TESTC	I	**Test C Pin.** Used for manufacturing test purposes; should be tied to 5 V for normal operation.
19	VDD	—	**5 V Supply.**
20	N	I	**Frequency Select.** This pin, with pins 1—3 and 12—14, sets the operating frequency range of the circuit. See Table 2 for frequency band selection.

Note: A circuit board ground plane is required for optimum performance.

Overview

The on-chip clock recovery circuit consists of a digital frequency-locked loop and a phase-locked loop, which extract the clock from the positive going edges of the input data. This recovered clock is used with the input data in the retimer section to synchronize the output data (DOUT) with the positive edge of the clock output (CLKOUT).

To ensure accurate frequency selection, the T7033 Clock Recovery Circuit uses an external 3.58-MHz crystal in its oscillator reference section. The operating frequency of the device is then determined by the circuit's seven frequency select pins, which are made high (5 V) or low (0 V). (See Table 2.) Special care is required for filtering the 5 V supply (VDD) on pin 15 (QVDD) since voltage variations on this pin may cause excessive jitter on the clock and data outputs.

DIN and LDIN are equivalent inputs. Typically, LDIN is used for the data loopback mode of system operation. For normal operation, TESTC (pin 18) and TESTF (pin 17) must be tied to 5 V. These pins are used only for testing during manufacture.

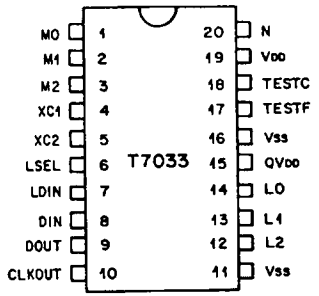
There are six octave selections that can be chosen from the frequency band selections in Table 2. Nine frequency bands can be selected from each octave. For optimal performance, a frequency of operation that is within the bands set by the seven frequency select pins must be selected.

Noise Properties

Noisy data under worst-case conditions produces a small eye opening and a large amount of phase jitter on the data input. Under these conditions, the T7033 Clock Recovery Circuit recovers the average clock and retimes the data, reducing jitter. Figures 3 and 4 illustrate the improvement of data quality through the circuit. At the limit of sensitivity, the clock recovery circuit imposes a typical noise penalty (noise factor) of < 1 dB.

User Information

Pin Descriptions



Symbol	Pin	Symbol	Pin
CLKOUT	10	M2	3
DIN	8	N	20
DOUT	9	QVDD	15
L0	14	TESTC	18
L1	13	TESTF	17
L2	12	VDD	19
LDIN	7	VSS	11
LSEL	6	VSS	16
M0	1	XC1	4
M1	2	XC2	5

Figure 2. Pin Function Diagram and Alphabetical Listing of Symbols

Table 1. Pin Descriptions

Pin	Symbol	Type	Name/Function
1 2 3	M0 M1 M2	I	**Frequency Selects.** These pins, with pins 12—14 and 20, set the operating frequency range of the circuit. See Table 2 for frequency band selection.
4,5	XC1,XC2	I	**XTAL1 and XTAL 2.** Connect a 3.579545-MHz crystal between these pins.
6	LSEL	I	**Alternate Data Select.** Tie to 0 V to select data on DIN (pin 8); tie to 5 V for data on LDIN (pin 7).
7	LDIN	I	**Alternate Data In.**
8	DIN	I	**Data In.**
9	DOUT	O	**Data Out.** Serial data output.
10	CLKOUT	O	**Clock Out.**
11	VSS	—	**Ground.** 0 V.
12 13 14	L2 L1 L0	I	**Frequency Selects.** These pins, with pins 1—3 and 20, set the operating frequency range of the circuit. See Table 2 for frequency band selection.

Note: A circuit board ground plane is required for optimum performance.

CY7C420, CY7C421, CY7C424
CY7C425, CY7C428, CY7C429

CYPRESS SEMICONDUCTOR

Cascadeable 512 x 9 FIFO
Cascadeable 1024 x 9 FIFO
Cascadeable 2048 x 9 FIFO

Features

- 512 x 9, 1024 x 9, 2048 x 9 FIFO buffer memory
- Dual port RAM cell
- Asynchronous read/write
- High speed 25 MHz read/write independent of depth/width
- Low operating power
  I_{CC} (max.) = 125 mA commercial
  I_{CC} (max.) = 140 mA military
- Half full flag in standalone
- Empty and full flags
- Retransmit in standalone
- Expandable in width and depth
- Parallel Cascade minimizes bubblethrough
- 5V ±10% supply
- 300 mil DIP packaging
- 300 mil SOJ (512 x 9) packaging
- TTL compatible
- Three-state outputs
- CY7C421 pin compatible and functional equivalent to IDT7201

Functional Description

The (CY7C420, CY7C421,) (CY7C424, CY7C425,) and (CY7C428, CY7C429) are, respectively, 512, 1024 and 2048 words by 9-bit wide first-in first-out (FIFO) memories offered in 600 mil wide and 300 mil wide packages, respectively. Each FIFO memory is organized such that the data is read in the same sequential order that it was written. Full and Empty flags are provided to prevent over-run and under-run. Three additional pins are also provided to facilitate unlimited expansion in width, depth, or both. The depth expansion technique steers the control signals from one device to another in parallel, thus eliminating the serial addition of propagation delays so that throughput is not reduced. Data is steered in a similar manner.

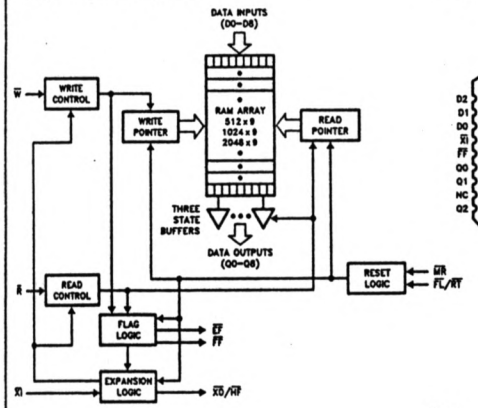
The read and write operations may be asynchronous; each can occur at a rate of 25 MHz. The write operation occurs when the Write ($\overline{W}$) signal is LOW. Read occurs when Read ($\overline{R}$) goes LOW. The 9 data outputs go to the high impedance state when $\overline{R}$ is HIGH.

A Half-Full ($\overline{HF}$) output flag is provided that is valid in the standalone and width expansion configurations. In the depth expansion configuration this pin provides the expansion out ($\overline{XO}$) information which is used to tell the next FIFO that it will be activated.

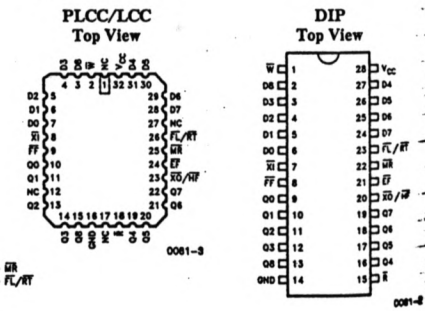
In the standalone and width expansion configurations a LOW on the Retransmit ($\overline{RT}$) input causes the FIFO's to retransmit the data. Read Enable ($\overline{R}$) and Write Enable ($\overline{W}$) must both be HIGH during a retransmit cycle, and then $\overline{R}$ is used to access the data.

The CY7C420, CY7C421, CY7C424, CY7C425, CY7C428 and CY7C429 are fabricated using an advanced 0.8 micron N-well CMOS technology. Input ESD protection is greater than 2000V and latchup is prevented by careful layout, guard rings and a substrate bias generator.

Logic Block Diagram



Pin Configurations



Selection Guide

		7C420-30, 7C421-30 7C424-30, 7C425-30 7C428-30, 7C429-30	7C420-40, 7C421-40 7C424-40, 7C425-40 7C428-40, 7C429-40	7C420-65, 7C421-65 7C424-65, 7C425-65 7C428-65, 7C429-65
Frequency (MHz)		25	20	12.5
Access Time (ns)		30	40	65
Maximum Operating Current (mA)	Commercial	125	115	100
	Military	140	130	115

Shaded area contains preliminary information.

Maximum Ratings

(Above which the useful life may be impaired. For user guidelines, not tested.)

Storage Temperature ..... −65°C to +150°C

Ambient Temperature with Power Applied ..... −55°C to +125°C

Supply Voltage to Ground Potential .... −0.5V to +7.0V

DC Voltage Applied to Outputs in High Z State..... −0.5V to +7.0V

DC Input Voltage ..... −3.0V to +7.0V

Power Dissipation 1.0W

Output Current, into Outputs (Low) ..... 20 mA

Static Discharge Voltage ..... >2001V (per MIL-STD-883 Method 3015)

Latch-up Current ..... >200 mA

Operating Range

Range	Ambient Temperature	V_{CC}
Commercial	0°C to +70°C	5V ±10%
Military[3]	−55°C to +125°C	5V ±10%

Electrical Characteristics Over Operating Range[4]

Parameters	Description	Test Conditions		CY7C420-30 CY7C421-30 CY7C424-30 CY7C425-30 CY7C428-30 CY7C429-30		CY7C420-40 CY7C421-40 CY7C424-40 CY7C425-40 CY7C428-40 CY7C429-40		CY7C420-65 CY7C421-65 CY7C424-65 CY7C425-65 CY7C428-65 CY7C429-65		Units
				Min.	Max.	Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = −2 mA		2.4		2.4		2.4		V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 8.0 mA			0.4		0.4		0.4	V
V_{IH}	Input HIGH Voltage		Commercial	2.0	V_{CC}	2.0	V_{CC}	2.0	V_{CC}	V
			Military	2.2	V_{CC}	2.2	V_{CC}	2.2	V_{CC}	V
V_{IL}	Input LOW Voltage			−3.0	0.8	−3.0	0.8	−3.0	0.8	V
I_{IX}	Input Leakage Current	GND ≤ V_I ≤ V_{CC}		−10	+10	−10	+10	−10	+10	μA
I_{OZ}	Output Leakage Current	$\overline{R}$ ≥ V_{IH}, GND ≤ V_O ≤ V_{CC}		−10	+10	−10	+10	−10	+10	μA
I_{CC}	Operating Current	V_{CC} = Max., I_{OUT} = 0 mA	Commercial[5]		125		115		100	mA
			Military[6]		140		130		115	mA
I_{SB_1}	Standby Current	All Inputs = V_{IH} Min.	Commercial		25		25		25	mA
			Military		30		30		30	mA
I_{SB_2}	Power Down Current	All Inputs V_{CC} −0.2V	Commercial		20		20		20	mA
			Military		25		25		25	mA
I_{OS}	Output Short Circuit Current[1]	V_{CC} = Max., V_{OUT} = GND			−90		−90		−90	mA

Shaded area contains preliminary information.

Capacitance[2]

Parameters	Description	Test Conditions	Max.	Units
C_{IN}	Input Capacitance	T_A = 25°C, f = 1 MHz	5	pF
C_{OUT}	Output Capacitance	V_{CC} = 4.5V	7	

Notes:
1. For test purposes, not more than one output at a time should be shorted. Short circuit test duration should not exceed 30 seconds.
2. Tested initially and after any design or process changes that may affect these parameters.
3. T_A is the "instant on" case temperature.
4. See the last page of this specification for Group A subgroup testing information.
5. I_{CC} (commercial) = 100 mA + [(f − 12.5) • 2 mA/MHz] for f ≥ 12.5 MHz
   where f = the larger of the write or read operating frequency.
6. I_{CC} (military) = 115 mA + [(f − 12.5) • 2 mA/MHz] for f ≥ 12.5 MHz
   where f = the larger of the write or read operating frequency.

5

SOURCE CODE FOR CONTROLLER PROCESSOR

```
;****** CTL.SRC ***** 9-25-90 ***** P.DOOLEY
**********************
;    CHIP 87C51FA

$ INCLUDE (87C51FA.INC)

A0   BIT  P1.0      ;LSB VCR ADDRESS SELECTION
A1   BIT  P1.1      ;     "    "
A2   BIT  P1.2      ;MSB  "    "
COAX BIT  P1.3      ;CONTROL RF RELAY TO SWITCH SIDEBANDS
SERLDN    BIT  P1.4     ;CONTROL VCR STATUS READIN
;LOCK     BIT  P1.5     ;STATUS FROM MKII FORMATTER
;TIME     BIT  P1.6     ;  "       "    "      "
DAT  BIT  P1.7      ;  "       "    "      "

IRCODE    BIT  P3.0     ;SERIAL IRCODE TO CONTROL VCR'S
BYTLD     BIT  P3.1     ;
SERIN     BIT  P3.2     ;EXTERNAL 'AUDIO' DATA INPUT AND CLK
EDGES
SERCLK    BIT  P3.3     ;SERIAL CLK FROM VCR'S MICRO
DAT_ADR   BIT  P3.4     ;DATA/ADDRESS LINE TO SLAVE
MCB_ACK   BIT  P3.5     ;ACK FROM MCB MICRO
BUSBUSYN BIT   P3.6     ;CTL uP TO MCB uP BUSY LINE
CMD_MON   BIT  P3.7     ;CMD/MON LINE FROM MASTER  MICRO

BITRDY    BIT  20H.0        ;DATA STROBED
STBCTR    BIT  20H.1        ;STROBES DATA AT 3.84 KHZ RATE
FRAME     BIT  20H.2        ;VALID FRAME
DONE BIT  20H.3         ;
BOW  BIT  20H.4         ;BEGIN OF WORD"FF" FROM VCR SERIAL
STATUS
EDGE BIT  20H.5         ;EDGES OF IRCODE BIT STREAM
AUDIO_MODE BIT 20H.6        ;SWITCH BETWEEN AUDIO MODE AND
IRCODE MODE

NRZBYT    EQU  30H      ;NRZI SERIAL BYTE DATA
DATBYTL   EQU  31H      ;NRZI CONVERTED TO BINARY LSBYTE
DATBYTH   EQU  32H      ;NRZI CONVERTED TO BINARY MSBYTE
BITCTR    EQU  33H      ;BIT COUNTER
NRZBYT1   EQU  34H      ;SECOND NRZBYT
BYTCTR    EQU  35H      ;BYTE COUNTER
DELAY     EQU  36H          ;GENERAL PURPOSE VARIABLE
BUS_ADR   EQU  37H      ;ADDR FROM INTERPROCESSOR BUS
DELAY0    EQU  38H      ;GEN. PURPOSE VARIABLE
LOOPCTR   EQU  39H      ;
ROACH     EQU  3AH

BUF_ADR   EQU  50H      ;BASE OF AUDIO FRAME BUFFER
;*************************************************************
*********
;    RESET VECTOR
```



```
     ORG  0000H
     SJMP INIT
;*************************************************************
*********
;    EXTERNAL INTERRUPT 0 - RESPONDS TO LEADING DATA EDGES

     ORG  0003H          ;INTERRUPT INT0N VECTOR (EXTERNAL
INTERRUPT 0)
     MOV  TL0,#0DEH ;TIMER 0 IS SYNCED TO EXTERNAL DATA CLK
     RETI           ;ALLOWS ABOUT 37 MICROSECONDS BEFORE TIMER 0
                    ;INTERRUPT   I.E. STROBE IN SERIAL DATA
;*************************************************************
*********
;    EXTERNAL INTERRUPT 1 - RESPONDS TO TRAILING DATA EDGES
;
     ORG  0013H          ;INTERRUPT INT1N VECTOR (EXTERNAL
INTERRUPT 1)
     MOV  TL0,#0DEH ;TIMER 0 IS SYNCED TO EXTERNAL DATA CLK
     RETI           ;ALLOWS ABOUT 37 MICROSECONDS BEFORE TIMER 0
                    ;INTERRUPT   I.E. STROBE IN SERIAL DATA
;*************************************************************
********
;    TIMER 0 INTERRUPT   ;RATE = 7.68 KHZ, 2X DATA FOR AUDIO DATA
                    ;RATE = 4.8 KHZ, KLOKS OUT IR_CODE TO
                    ;VCR'S AT THIS RATE
     ORG  000BH          ;INTERRUPT TF0 VECTOR (TIMER 0 OVERFLOW
)
     ACALL     SWITCH
     RETI
;*************************************************************
********
;    INITIALIZE PROGRAM PARAMETERS

     ORG  0030H
INIT:     MOV  NRZBYT,#0 ;CLEAR CONTENTS OF NRZ BYTE REG
     MOV  NRZBYT1,#0     ;     "         2ND    "
     MOV  TMOD,#22H ;TIMER 0 & 1 SET TO AUTO RELOAD
     MOV  P2,#0FFH  ;
     MOV  P3,#0
     MOV  ROACH,#0
     SETB SERIN          ;SET PIN TO SERIAL DATA INPUT   INT1N
     SETB SERCLK         ;SET PIN TO CLK INPUT ,EXTERNAL
INTERRUPT INT0N
     SETB DAT_ADR        ;SET PIN FOR DATA
     SETB BUSBUSYN  ;
     SETB CMD_MON        ;
     SETB MCB_ACK        ;SET PIN TO INPUT
     SETB SERLDN         ;SET HI
     SETB BYTLD
     CLR  COAX
     CLR  BOW       ;NO BEGIN OF WORD
     CLR  BITRDY         ;
```

```
     CLR  AUDIO_MODE      ;
     SETB IT0         ;MAKE INT0N EDGE ACTIVATED
     SETB IT1         ;MAKE INT1N EDGE ACTIVATED

;*************************************************************
*******

MAIN:    JB      BUSBUSYN,$      ;WAIT FOR VIDEO uP TO SEND ITS DATA
     JNB  BUSBUSYN,$      ;  "          "          "       "
     ACALL       CTL_VCR

     ACALL       COAX_RELAY

     ACALL       MON_VCR

     ACALL       DECODE_AUDIO

     ACALL       FMTR_STATUS

     ACALL       DEBUG
     AJMP MAIN
;*************************************************************
****
DEBUG:
     MOV  BUS_ADR,#06H
     MOV  A,ROACH
     ACALL       SEND_MON
     RET
;************************ CONTROL VCR TASK
************************

CTL_VCR:SETB    P1.5
     CLR  P1.5
     MOV  R6,#0           ;SET TO VCR #1
CV0: MOV  BUS_ADR,R6      ;POINT TO COMMAND ADDRESS FOR VCR #1
     ACALL       GET_CMD         ;GET THE COMMAND FROM THE MCB MICRO
     JZ   CV2         ;IF COMMAND IS ZERO, BYPASS SENDING CMD TO
VCR
CV1: ACALL       GET_CODES_PTR   ;DPTR POINTS TO CODE STREAM FOR CMD
     ACALL       SEND_IR_CODE    ;SEND IR CMD CODE STREAM TO VCR
     MOV  A,#0        ;CLEAR REG A
     ACALL       SEND_MON   ;CLEAR THE VCR CONTROL BYTE IN THE MCB
MICRO
CV2: INC  R6          ;PT TO NEXT VCR
     CJNE R6,#6,CV0 ;NO MORE VCR'S - EXIT
     RET
;.....
........

SEND_IR_CODE:
     MOV  A,R6        ;GET THE VCR ADDRESS
     ANL  P1,#11111000B   ;AND OUT THE VCR ADDRESS ON PORT 1
```



```
     ORL  P1,A        ;OR IN THE NEW ADDRESS
        MOV     TH0,#40H  ;TIMER 0 RELOAD SET FOR 4.8 KHZ SAMPLE
     MOV  TL0,#40H  ; "
     SETB     PT0          ;SET TIMER 0 INTERRUPT TO HIGHEST
PRIORITY
     SETB ET0         ;ENABLE TIMER 0 INTERRUPT
     SETB       TR0         ;START TIMER 0
     SETB EA
LOOP:       CLR  A           ;
     CLR  C
     MOVC A,@A+DPTR ;MOVE LENGTH OF "1" AND "0" TO ACC
     ACALL       TX_IR_BITS      ;SEND IT TO VCR
     CJNE A,#50,MORE      ;TEST TO SEE IF END OF CODE
     SJMP MOREO

MORE:       MOV  A,DPL            ;
     ADD  A,#1             ;
     MOV  DPL,A            ;
     JNC  LOOP             ;
     INC  DPH              ;
     SJMP LOOP             ;

MOREO:  CLR     EA
     CLR  TR0         ;STOP TIMER 0
     CLR  ET0         ;DISABLE TIMER 0 INTERRUPT
     CLR  PT1         ;RESET INTERRUPT PRIORITY
     CLR  IRCODE           ;INSURE OUTPUT IS 0 WHEN DONE SENDING
CODE
     RET
;.....
.....

TX_IR_BITS:
     PUSH ACC              ;SAVE REG A
     INC  A
     SETB EDGE
TIC0:       JNB  BITRDY,$  ;WAIT FOR 4.8 KHZ INTERRUPT
     CLR  BITRDY           ;
     JB   EDGE,TIC1 ;EDGE DETECTED - COMPLEMENT IRCODE
     SJMP TIC2        ;NO EDGE - BYPASS COMPLEMENT

TIC1:       CPL  IRCODE           ;COMPLEMENT IRCODE
     CLR  EDGE             ;
TIC2:       DJNZ ACC,TIC0  ;SEND IRCODE BITS FOR .1MS X ACC
     POP  ACC         ;RESTORE ACC VALUE
     RET
;.....
.....

GET_CODES_PTR:
     JB   ACC.0,CMD_0
     JB   ACC.1,CMD_1
```

```
     JB    ACC.2,CMD_2
     JB    ACC.3,CMD_3
     JB    ACC.4,CMD_4
     JB    ACC.5,CMD_5
     JB    ACC.6,CMD_6
     JB    ACC.7,CMD_7
CMD_0:    MOV      DPTR,#POWER
     SJMP CMD_EX
CMD_1:    MOV      DPTR,#PAUSE
     SJMP CMD_EX
CMD_2:    MOV      DPTR,#PLAY
     SJMP CMD_EX
CMD_3:    MOV      DPTR,#RECORD
     SJMP CMD_EX
CMD_4:    MOV      DPTR,#F_FWD
     SJMP CMD_EX
CMD_5:    MOV      DPTR,#REWIND
     SJMP CMD_EX
CMD_6:    MOV      DPTR,#VCR
     SJMP CMD_EX
CMD_7:    MOV      DPTR,#STOP
CMD_EX:   RET

;********************** MONITOR VCR TASK
*****************************

MON_VCR:
     MOV  R6,#0          ;R6 CONTAINS VCR NUMBER 0-5
MV0: ACALL     GET_SER_DAT
        INC     R6
     CJNE R6,#6,MV0
     RET
;.....................................................................
......

GET_SER_DAT:
     CLR  BOW
     MOV  A,R6       ;GET THE VCR ADDRESS
     ANL  P1,#11111000B  ;AND OUT THE VCR ADDRESS ON PORT 1
     ORL  P1,A       ;OR IN THE NEW ADDRESS
     MOV  LOOPCTR,#25    ;LOOP 25 TIMES LOOKING FOR BOW - NOT
FND-EXIT
GSDA:     DJNZ LOOPCTR,GSDB   ;
     SJMP GSD00

GSDB:   CLR    SERLDN
     SETB SERLDN
     MOV  DELAY0,#20     ;SET UP DOUBLE LOOP FOR LONG TIME DELAY
GSDC:     MOV  DELAY,#255     ;INNER LOOP CT
GSD0:     JB   BYTLD,GSD1     ;WAIT FOR SERIAL CLOCK FOR VCR
STATUS
     DJNZ DELAY,GSD0     ;INNER LOOP
```



```
     DJNZ DELAY0,GSDC    ;OUTER LOOP
GSD00:    MOV  A,#0BDH        ;NO CLOCK IN ~12 MS - FLAG "BD" =
BAD DATA
     SJMP GSD3

GSD1:     MOV  A,P0      ;GET BYTE FROM PORT 0
     CLR  SERLDN
     SETB SERLDN         ;RESET SHIFT REG
     JB   BOW,GSD2  ;BEGIN OF WORD FOUND - NOW SEARCH FOR BYTE 3
     CJNE A,#0FFH,GSDA   ;NO BOW - SEACRCH 25 BYTES FOR "0FFH"
     SETB BOW       ;BEGINNING OF STATUS WORD FROM SERIAL BUS
     MOV  BYTCTR,#5 ;COUNTDOWN 5 BYTES TO FIND STATUS BIT
     SJMP GSDB      ;GET ANOTHER BYTE

GSD2:     DJNZ BYTCTR,GSDB    ;NOT BYTE 2 - GET ANOTHER
     CLR  BOW
GSD3:     PUSH ACC
     MOV  A,R6
     ADD  A,#08H         ;BASE ADDRESS OF VCR MONITOR DATA
     MOV  BUS_ADR,A ;SET UP MONITOR ADDRESS
     POP  ACC
     ACALL     SEND_MON
     RET

;****************** DECODE AUDIO TASK
*******************************
;    DECODE AUDIO DATA FROM THE MKII RECORDING SYSTEM
;    GET A GOOD FRAME AND SEND RESULTS TO MCB MICRO

DECODE_AUDIO:
     MOV  BUS_ADR,#1FH         ;ADDRESS OF DQA SELECT ON MCB
     ACALL     GET_CMD         ;
     JZ       DA1         ;IF NO INPUT'S ARE SPECIFIED - EXIT
     ACALL     GET_ADR         ;CONVERTS BITS TO OCTAL ADDRESS
     ANL  P1,#11111000B  ;CLEAR OUT THE EXTERN ADDRESS BITS FOR
PORT 1
     ORL  P1,A      ;AUDIO INPUT ADDRESS
     SETB AUDIO_MODE     ;ROUTE TIMER 0'S INTERRUPT'S TO "INPUT:"
     MOV  TH0,#88H  ;TIMER 0 RELOAD SET FOR 7.68 KHZ
     MOV  TL0,#88H  ; "
     SETB PT0       ;SET TIMER 0 HIGHEST INTERRUPT PRIORITY
     SETB ET0       ;ENABLE TIMER 0 INTERRUPT
     SETB EX0       ;ENABLE EXTERNAL INTERRUPT 0
     SETB EX1       ;ENABLE EXTERNAL INTERRUPT 1
     SETB TR0       ;START TIMER 0 FOR AUDIO CLK
     SETB EA        ;ENABLE ALL INTERRUPTS
     MOV  DELAY0,#250
DA0: MOV  DELAY,#255
DA00:     JB   DONE,DAEND     ;WAIT TILL ONE GOOD FRAME IS LOADED
     DJNZ DELAY,DA00     ;***** MUST TIMEOUT IF NOTHING DECODED
     DJNZ DELAY0,DA0
     CLR  EA
```

```
DA1: ACALL     NO_AUD_FRM      ;SET'S AUDIO BUFFER SPACE TO 0BDH'S
DAEND:    CLR  TR0
     CLR  ET0
     CLR  EX0
     CLR  EX1
     CLR  PT0        ;REMOVE TIMER 0 INTERRUPT LEVEL
     CLR  DONE
     CLR  AUDIO_MODE     ;
     ACALL     AUDIO_2_MCB     ;MOVES DATA FROM CTL AUDIO BUFFER
TO MCB MICRO
     RET                       ;MUST NOT HANG !!!!!!!!
;.....
.....
GET_ADR:
     JB   ACC.0,VCR_1
     JB   ACC.1,VCR_2
     JB   ACC.2,VCR_3
     JB   ACC.3,VCR_4
     JB   ACC.4,VCR_5
     JB   ACC.5,VCR_6
     JB   ACC.6,FMTR_IN
     JB   ACC.7,EXT_IN
VCR_1:    MOV      A,#0
     SJMP GET_ADR_EXIT
VCR_2:    MOV  A,#1
     SJMP     GET_ADR_EXIT
VCR_3:    MOV      A,#2
     SJMP     GET_ADR_EXIT
VCR_4:    MOV      A,#3
     SJMP     GET_ADR_EXIT
VCR_5:    MOV      A,#4
     SJMP     GET_ADR_EXIT
VCR_6:    MOV      A,#5
     SJMP     GET_ADR_EXIT
FMTR_IN:MOV      A,#6
     SJMP     GET_ADR_EXIT
EXT_IN:   MOV      A,#7
GET_ADR_EXIT:
     RET

;****************** FORMATTER STATUS TASK
*****************************
;
FMTR_STATUS:
     MOV  A,#0       ;
     ANL  P1,#11111000B  ;
     JNB  P1.7,STAT_1    ;
     SETB ACC.0          ;
STAT_1:   ANL  P1,#11111000B  ;
     ORL  P1,#00000001B  ;
     JNB  P1.7,STAT_2
     SETB ACC.1
```



```
STAT_2:   ANL  P1,#11111000B  ;
     ORL  P1,#000000010B  ;
     JNB  P1.7,STAT_3     ;
     SETB ACC.2           ;
STAT_3:   ANL  P1,#11111000B  ;
     ORL  P1,#000000011B  ;
     JNB  P1.7,STAT_4     ;
     SETB ACC.3           ;
STAT_4:   ANL  P1,#11111000B  ;
     ORL  P1,#000000100B  ;
     JNB  P1.7,STAT_5     ;
     SETB ACC.4           ;
STAT_5:   ANL  P1,#11111000B  ;
     ORL  P1,#000000101B  ;
     JNB  P1.7,STAT_6     ;
     SETB ACC.5           ;
STAT_6:   ANL  P1,#11111000B  ;
     ORL  P1,#000000110B  ;
     JNB  P1.7,STAT_7     ;
     SETB ACC.6           ;
STAT_7:   ANL  P1,#11111000B  ;
     ORL  P1,#000000111B  ;
     JNB  P1.7,STAT_8     ;
     SETB ACC.7           ;
STAT_8:
;    MOV  A,P1       ;GET THE STATUS
;    ANL  A,#11100000B    ;MASK THE IMPORTANT BITS
     MOV  BUS_ADR,#1AH    ;FORMATTER STATUS MCB ADDRESS
     ACALL     SEND_MON   ;SEND TO MCB MICRO'S MCB ADDR SPACE
     RET

;***************** CONTROL COAX RELAY TASK
*************************
;
COAX_RELAY:
     MOV  BUS_ADR,#1BH    ;COAX RELAY COMMAND ADDRESS IN MCB MICRO
     ACALL     GET_CMD         ;GET COMMAND IN MCB MICRO
     ANL  A, #00001000B   ;MASK THE COAX CONTROL BIT
     ANL  P1,#11110111B
     ORL  P1,A       ;SEND IT TO PORT 1
     RET

;****************** VARIOS SUBROUTINES
*******************************
     ;GET COMMAND FROM MCB MICRO'S MCB RELATIVE ADDRESS SPACE
     ; ADDRESS IS IN "BUS_ADR" AND DATA IS RETURNED IN REG "A"
GET_CMD:
     MOV  DELAY,#255      ;100 uSEC TIMEOUT
GC0: JB   BUSBUSYN,GC00   ;TEST IF uP BUS IS BUSY
     DJNZ DELAY,GC0 ;
     MOV  A,#0       ;NO COMMAND
     SJMP GC40       ;
```

```
GC00:     CLR  BUSBUSYN  ;GET CONTROL OF BUS BY THIS uP
     MOV  P2,BUS_ADR     ;OUTPUT ADDRESS TO PORT 2
     CLR  DAT_ADR        ;SIGNAL TO MCB MICRO ADDRESS IS ONLINE
     MOV  DELAY,#255     ;SETUP FOR 100 MICROSEC TIME IF NO ACK
GC1:    JNB    MCB_ACK,GC2    ;IF MCB ACK - GOTO GC2
     DJNZ DELAY,GC1 ;NO MCB ACK AT TIMEOUT - ESCAPE
;    MOV  A,#0BDH        ;NO COMMAND
     MOV  A,#0       ;NO COMMAND
     MOV  P2,#0FFH  ;SET PORT 2 TO INPUT
     SETB DAT_ADR        ;RESTORE DAT_ADR
     SJMP GC40

GC2:    SETB   DAT_ADR        ;TELL MCB MICRO TO PUT DATA ON THE
BUS
     MOV  P2,#0FFH  ;SET PORT 2 TO INPUT
     MOV  DELAY,#255
GC3: JB      MCB_ACK,GC4 ;WAIT TILL ACK LINE GOES HI OR TIMEOUT
     DJNZ DELAY,GC3 ;
;    MOV  A,#0BDH        ;NO COMMAND
     MOV  A,#0       ;NO COMMAND
     SJMP GC40

GC4: MOV  A,P2       ;GET COMMAND FROM PORT 2
GC40:     MOV  P2,#0FFH  ;SET PORT 2 TO INPUT
     MOV  DELAY,#25 ;
     DJNZ DELAY,$        ;HOLD BUSBUSYN FOR ~50 uS
     SETB BUSBUSYN  ;RELINQUISH uP BUS
     MOV  ROACH,#0
     ORL  ROACH,A
     RET

;*************************************************************
*******
     ;SEND MONITOR BYTE TO MCB REL. ADDRESS IN MCB MICRO
     ; ADDRESS IS IN "BUS_ADR" AND DATA IS IN REG "A"
SEND_MON:
     MOV  DELAY,#255     ;100 uSEC TIMEOUT
SEND0:    JB   BUSBUSYN,SEND00     ;TEST IF uP BUS IS BUSY
     DJNZ DELAY,SEND0    ;
     SJMP SEND5
SEND00:   CLR  BUSBUSYN  ;ASSERT BUS BUSY BY THIS uP
     CLR  CMD_MON        ;SET TO MONITOR MODE
     MOV  P2,BUS_ADR     ;OUTPUT ADDRESS TO PORT 2
     CLR  DAT_ADR        ;SIGNAL MCB MICRO - ADDRESS ONLINE
     MOV  DELAY,#255     ;SETUP FOR 100 MICROSEC TIMEOUT IF NO
ACK
SEND1:    JNB  MCB_ACK,SEND2  ;IF MCB ACK - GOTO SEND2
     DJNZ DELAY,SEND1    ;NO MCB ACK AT TIMEOUT - ESCAPE
     SJMP SEND4

SEND2:    MOV  P2,A      ;OUTPUT DATA TO MICRO ON PORT 2
     SETB DAT_ADR        ;TELL MCB MICRO DATA IS AVAILABLE
```



```
     MOV  DELAY,#255     ;100 MICROSEC TIMEOUT IF ACK DOESN'T GO
HI
SEND3:    JB   MCB_ACK,SEND4  ;MCB ACK RETURNS HI - MCB MICRO - I
GOT IT
     DJNZ DELAY,SEND3    ;NO MCB ACK - ESCAPE
SEND4:  MOV    P2,#0FFH  ;SET PORT 2 TO INPUT
SEND5:    MOV  DELAY,#25 ;
     DJNZ DELAY,$        ;HOLD BUSBUSYN FOR ~50 uS
     SETB CMD_MON        ;
     SETB BUSBUSYN  ;RELINQUISH uP BUS
     RET

;*************************************************************
*******
     ;MOVE AUDIO DATA FROM CTL MICRO BUFFER TO MCB MICRO BUFFER

AUDIO_2_MCB:
     MOV  BYTCTR,#8
     MOV  R0,#BUF_ADR
     MOV  BUS_ADR,#BUF_ADR AND 1FH ;POINTS TO MCB ADR OF 1XH
A2M0:     MOV  A,@R0
     ACALL     SEND_MON
     INC  R0
     INC  BUS_ADR
     DJNZ BYTCTR,A2M0
     RET

;*************************************************************
*******
     ;SET AUDIO BUFFER SPACE TO BD'S WHEN NO FRAME FOUND IN ~60
MS

NO_AUD_FRM:
     MOV  BYTCTR,#8
     MOV  R0,#BUF_ADR
NAF0:     MOV  @R0,#0BDH
     INC  R0
     DJNZ BYTCTR,NAF0
     RET
;*************************************************************
*******

     ;SWITCH TIMER INTERRUPTS BETWEEN AUDIO & IRCODE ROUTINES
SWITCH:
     JB   AUDIO_MODE,SWITCH0
     SETB BITRDY
     SJMP SWITCH1

SWITCH0:
     ACALL     INPUT          ;GET'S AN NRZ BIT FROM THE AUDIO
SERIAL INPUT
```

```
SWITCH1:                              ;AND CONVERTS TO BINARY - THEN
DEFORMATS DATA
     RET

;*****************************************************************
*******
;    THE REST OF TIMER 0 INTERRUPT

INPUT:
     PUSH ACC
     PUSH PSW
     PUSH 00H
     MOV  A,#0              ;CLEAR A
     MOV  C,SERCLK  ;GET NEW BIT VALUE FROM SER NRZI INPUT
     MOV  ACC.0,C           ;MOVE IT TO ACCUMULATOR
     CJNE A,NRZBYT,CONT0 ;COMPARE WITH PREVIOUS NRZBYT
     CLR  C          ;DO XOR ON DATA
     CLR  STBCTR            ;RESET STROBE COUNTER
     SJMP CONT00

CONT0:    SETB C
CONT00:   MOV  NRZBYT,A
     JB   STBCTR,EXIT     ;BYPASS STROBEING DATA WHEN SET
     MOV  A,DATBYTL ;GET OLD DATA LSBYTE
     RRC  A         ;SHIFT LEFT WITH CARRY
     MOV  DATBYTL,A         ;STORE NEW DATA BYTE
     MOV  A,DATBYTH ;GET OLD DATA MSBYTE
     RRC  A         ;MOVE THE CARRY INTO THE LSBIT OF A
     MOV  DATBYTH,A ;STORE NEW HIGH DATA BYTE

;    NOW TEST FOR BEGIN OF FRAME

     MOV  A,DATBYTL
     CJNE A,#0FFH,CONT2  ;JUMP IF PART BEGIN OF FRAME NOT FOUND
     MOV  A,DATBYTH         ;GET HIGH DATA BYTE
     ANL  A,#0F0H           ;CHECK IF LOWER NIBBLE IS 0
     CJNE      A,#0,CONT2      ;JUMP IF REST OF BOFRM NOT FOUND

;    BEGIN OF FRAME FOUND - BYTE IS READY

     SETB FRAME             ;START OF FRAME
     MOV  BITCTR,#1 ;SET BIT COUNTER
     MOV  R1,#BUF_ADR       ;START OF AREA FOR AUDIO FROM FORMATTER
     MOV  @R1,DATBYTL       ;MOVE DATA TO MCB BUFFER
     SJMP EXIT

CONT2:    MOV  A,BITCTR  ;GET BIT COUNTER
     CJNE A,#8,CONT3        ;TEST IF 8
     MOV  BITCTR,#0 ; IT IS, CLEAR BIT COUNTER
     JNB  FRAME,CONT3

;    BYTE IS READY WITHIN FRAME
```



```
     INC  R1          ;
     MOV  @R1,DATBYTL       ;MOVE DATA TO BUFFER
     MOV  A,R1         ;GET BYTE COUNTER
     CJNE A,#BUF_ADR+7,CONT3  ;TEST IF > 8 BYTES SINCE BOFRM
     CLR  EA
     SETB DONE
     CLR  FRAME             ;
     SJMP EXIT

CONT3:    INC  BITCTR            ;
EXIT:     CPL  STBCTR            ;
     POP  00H
     POP  PSW
     POP  ACC
     RET

;*****************************************************************
******
;                VCR COMMANDS
POWER:    DB
17,15,4,2,5,10,4,2,5,2,4,2,5,2,4,2,4,2,5,2,4,2,4,11,4,11,4,2,4,11
,4,11,4,10,5,10,5,10,4,11,4,11,4,10,5,2,4,50

PAUSE:    DB
18,14,5,2,4,11,4,2,5,2,4,2,4,3,4,10,5,10,5,2,4,2,5,2,4,11,4,2,4,1
1,4,11,4,11,4,11,4,2,5,2,4,11,4,10,5,10,5,50

PLAY:     DB
17,15,5,1,5,10,5,2,4,2,5,2,4,2,5,10,5,2,4,10,5,2,4,2,5,10,5,2,4,1
1,4,11,4,10,5,10,5,2,4,11,4,2,5,10,5,10,4,50

RECORD:   DB
17,14,5,1,5,10,5,2,4,2,5,2,4,2,5,2,4,2,4,11,4,3,4,2,4,11,4,2,5,10
,5,10,5,10,4,11,4,11,4,11,4,2,5,9,5,10,4,50

F_FWD:    DB
17,15,4,2,5,10,5,2,3,2,5,2,4,11,4,11,4,2,4,3,4,2,4,2,5,10,5,2,4,1
1,4,11,4,11,4,2,4,2,5,10,5,10,5,10,5,10,4,50

REWIND:   DB
17,15,4,2,5,10,5,2,4,2,4,3,4,2,3,11,4,2,5,2,4,2,5,2,4,11,4,2,4,11
,4,11,4,11,4,11,4,2,5,10,4,11,4,11,4,11,4,50

VCR: DB
18,14,5,2,4,11,4,2,4,2,4,2,4,3,4,10,5,2,4,11,4,2,5,10,5,10,4,3,4,
10,5,10,5,10,5,10,5,2,4,11,4,2,4,11,4,2,5,50

STOP:     DB
17,15,4,2,5,10,5,2,4,2,5,2,4,2,4,3,4,2,4,2,5,2,4,2,5,10,5,2,4,11,
4,10,5,10,5,10,5,9,5,10,5,10,4,11,4,11,4,50
```

SOURCE CODE FOR VIDEO DQA PROCESSOR

```
;******* CV.SRC ******** OCT. 31,1990 ******* P.DOOLEY
***********
;
;     REVISED VERSION OF CVID.SRC
;          1. 1PPS IS SYNCRONIZED AFTER 10 GOOD BOF'S TO THE BOF
;          2. TIMER 2 USED IN AUTO RELOAD MODE AND SYNCRONIZED TO
BOF'S
;          3 XTAL CHANGED TO 11.0592 MHZ-TIMER 2 RELOAD VALUE IS
C400H
;          4. MSBIT OF FCERRS SENT TO MCB IS DATA LOK BIT
;
;     GETS BUFFERED FIFO INFO FROM VIDEO DECODER, SUCH AS
;     1. HEADER - WHICH IS THE FRAME COUNT PLUS PARITY
;     2. 16 BYTES OF TEST PATTERN-ONLY 10 USED
;     OTHER DATA FROM VIDEO DECODER
;     3. DP SYNC'S
;     4. EOF'S
;
;     THIS DATA IS SENT TO MCB CONTROLLER EVERY SECOND
;     1. 10 BYTES OF TEST PATTERN FROM THE LAST FRAME
;     2. 1 BYTE FRAME CT COMPARE ERRORS
;     3. 2 BYTES - NO. OF DP SYNCS ARE COUNTED
;     4. 2 BYTES - NO. OF "END OF FRAME'S" COUNTED
;     5. 1 BYTE - TICK COUNTER - 1/SEC WRAPS AROUND
;
;*************************************************************
******
;     CHIP 8751FA
$ INCLUDE (87C51FA.INC)

DAT_ADR   BIT  P1.4      ;DATA/ADDRESS LINE TO SLAVE
CMD_MON   BIT  P1.5      ;CMD/MON LINE TO MCB uP
MCB_ACK   BIT  P1.6      ;ACK FROM MCB uP
BUSBUSYN BIT   P1.7      ;uP BUS BUSY

FFRDN     BIT  P3.0      ;FIFO READ PULSE
FFEMPN    BIT  P3.1      ;FIFO EMPTY
BOFIN     BIT  P3.2      ;
DPSIN     BIT  P3.3      ;
EOFIN     BIT  P3.5      ;

ONEPPS    BIT  20H.0          ;1 PULSE PER SECOND
BOFRM     BIT  20H.1          ;BEGIN OF FRAME
CNT12     BIT  20H.2          ;SIGNALS 12th XFR FROM FIFO TO uP
PARITY    BIT  20H.3          ;USED TO TEST PARITY OF FRAME NUM
IN HEADER
BOFLOCK   BIT  20H.4          ;TEN GOOD BOF'S HAVE COME IN

VARS EQU  30H       ;BEGINNING OF VARIABLES AREA
THIRD_TIMER_BYTE  EQU  VARS+0 ;USED FOR GENERATING 1PPS
DPSCTRL   EQU  VARS+1         ;DP SYNC CTR - 30 COUNTS/FRAME LO
BYTE
```

2 DQA PROCESSOR

```
DPSCTRH   EQU  VARS+2         ;DP SYNC CTR - 30 COUNTS/FRAME HI
BYTE
FIFOPTR   EQU  VARS+3         ;FIFO DATA STORAGE POINTER
EOFCTRL   EQU  VARS+4         ;TALLYS # OF EOF'S/SEC LO BYTE
EOFCTRH   EQU  VARS+5         ;TALLYS # OF EOF'S/SEC HI BYTE
FCERRS    EQU  VARS+6         ;FRAME NUMBER COMPARE ERRORS FROM
HEADER
DELAY     EQU  VARS+7         ;SOFTWARE DELAY
DATPTR    EQU  VARS+8         ;uP TO PC DATA POINTER
FRMNUMP   EQU  VARS+9         ;PREVIOUS FRAME NUMBER
BUS_ADR   EQU  VARS+12        ;ADDR FROM INTERPROCESSOR BUS
DELAY_I   EQU  VARS+13        ;USED FOR DELAYS IN INTERRUPT
ROUTINES
LOCKCTR   EQU  VARS+14        ;
TL2_P     EQU  VARS+15
TH2_P     EQU  VARS+16        ;
BOFCTR    EQU  VARS+17        ;
BOFCTR2   EQU  VARS+18        ;COUNTS BOF'S PER SECOND

DATBUF    EQU  48H
FRMNUM    EQU  DATBUF+0  ;HEADER-WHICH IS THE FRAME NUMBER/PARITY
OZAVG     EQU  DATBUF+1  ;
TSTPAT1   EQU  DATBUF+2  ;FIRST BYTE OF THE TEST PATTERN
TSTPAT2   EQU  DATBUF+3  ;
TSTPAT3   EQU  DATBUF+4        ;
TSTPAT4   EQU  DATBUF+5        ;
TSTPAT5   EQU  DATBUF+6        ;
TSTPAT6   EQU  DATBUF+7        ;
TSTPAT7   EQU  DATBUF+8        ;
TSTPAT8   EQU  DATBUF+9        ;
TSTPAT9   EQU  DATBUF+10       ;
TSTPAT10 EQU   DATBUF+11       ;
FCERRSP   EQU  DATBUF+12 ;FRAME COMPARE ERRORS PER SECOND
                  ; MAX = 60 --- PREV. FRAME
DPSCTPL   EQU  DATBUF+13 ;DP SYNC CT PREVIOUS FRAME LO BYTE
DPSCTPH   EQU  DATBUF+14 ;DP SYNC CT PREVIOUS FRAME HI BYTE
                  ;30 DP'S * 60 FRAMES/SEC = 1800 MAX.
EOFCTPL   EQU  DATBUF+15 ;EOF CT'S PREV. FRAME LO BYTE
EOFCTPH   EQU  DATBUF+16 ;EOF CT'S PREV. FRAME HI BYTE
                  ;NO. OF EOF'S/SEC. = 60 MAX.
TIK_CTR   EQU  DATBUF+17 ;INCREMENTS ONCE PER SECOND
;*************************************************************
******
;     RESET VECTOR

     ORG  0000H
     AJMP INIT
;*************************************************************
******
;     EXTERNAL INTERRUPT 0     DECTECTED BEGINNING OF FRAME

     ORG  0003H          ;
```

```
3 DQA PROCESSOR

     ACALL     EXTERNAL_INT_0
     RETI
;*************************************************************
******
;    EXTERNAL INTERRUPT 1     DECTECTED DP SYNC'S

     ORG  0013H           ;
     ACALL     EXTERNAL_INT_1 ;
     RETI
;*************************************************************
********
;    TIMER 0 INTERRUPT   RATE =  60/SEC  ULTIMATELY GENERATES
1PPS
;
;    ORG  000BH           ;INTERRUPT TF0 VECTOR (TIMER 0 OVERFLOW)
;    ACALL     INTERNAL_INT_2
;    RETI
;*************************************************************
********
;    TIMER 1 INTERRUPT   ;END OF FRAME ON PIN T1(P3.5)
                    ;TRIGERS OVERFLOW OF TIMER 1
     ORG  001BH           ;TF1 VECTOR FOR NEXT COUNT INPUT
     CLR  TR1        ;STOP TIMER 1
     ACALL     EXTERNAL_INT_3 ;
     RETI
;*************************************************************
*********
;    TIMER 2 INTERRUPT   RATE = 60/SEC ULTIMATELY GENERATES 1PPS

     ORG  002BH           ;TF2 VECTOR (TIMER 2 OVERFLOW)
     ACALL     INTERNAL_INT_2
     RETI
;*************************************************************
*********
;    INITIALIZE PARAMETERS

     ORG  0040H
INIT:
     SETB DAT_ADR         ;SET PIN FOR DATA
     SETB CMD_MON         ;
     SETB MCB_ACK         ;SET PIN TO INPUT
     SETB BUSBUSYN   ;SET TO NOT BUSY
     SETB FFRDN           ;SET FIFO READ HIGH
     SETB BOFIN           ; SET PINS TO INPUT
     SETB DPSIN           ;
     SETB EOFIN           ;
     SETB FFEMPN          ;
     SETB PX0        ;MAKE INT0 EXTERNAL INTERRUPT HIGHEST
PRIORITY
     SETB IT0        ;INT0 EDGE ACTIVATED
     SETB IT1        ;INT1 EDGE ACTIVATED
```

```
4 DQA PROCESSOR

     MOV  TMOD,#51H ;TIMER 0: MODE 1,OSC
INPUT(TIMER+SOFTWARE=1PPS)
                    ;TIMER 1: MODE 1,PIN T1 INPUT(END OF FRAMES)
     MOV  RCAP2L,#00H     ;INIT TIMER 2 LO BYTE AUTORELOAD VALUE
     MOV  RCAP2H,#0C4H    ;"    "      HI   "       "
     MOV  THIRD_TIMER_BYTE,#30 ;THIRD_TIMER_BYTE OVERFLOWS ONCE
PER SEC.
     MOV  FIFOPTR,#DATBUF+0 ;INIT FIFO DATA STOREAGE PTR
     MOV  DATPTR,#DATBUF+1  ;INIT uP TO PC DATA PTR
     MOV     DPSCTRL,#0  ;INIT VARIABLES
     MOV  DPSCTRH,#0      ;     "
     MOV  DPSCTPL,#0      ;        "
     MOV  DPSCTPH,#0      ;        "
     MOV  EOFCTRL,#0      ;        "
     MOV  EOFCTRH,#0      ;        "
     MOV  EOFCTPL,#0      ;        "
     MOV  EOFCTPH,#0      ;        "
     MOV  FCERRS,#0 ;        "
     MOV  FCERRSP,#0      ;        "
     MOV  TL1,#0FFH ;SET TIMER 1 TO OVERFLOW WITH 1 COUNT
     MOV  TH1,#0FFH ;INPUT @ PIN T1(P3.5)
     SETB TR1        ;ENABLE TIMER 1 TO ACCEPT PULSES FROM PIN T1
     SETB TR2        ;START TIMER 2
     SETB EX0        ;ENABLE INTON,EXTERNAL INTERRUPT 0
     SETB EX1        ;ENABLE   INT1N,EXTERNAL INTERRUPT 1
     SETB ET1        ;ENABLE TIMER 1 OVERFLOW INTERRUPT
     SETB ET2        ;ENABLE TIMER 2 OVERFLOW INTERRUPT
     SETB EA         ;ENABLE ALL INTERRUPTS INDIVIDUALLY ENABLED
;********************* MAIN
*********************************************
MAIN:     JNB  BOFRM,MAIN1    ;TEST FOR BEGINNING OF FRAME
     CLR  BOFRM           ;CLEAR THE FLAG
     INC  BOFCTR2         ;TEST TO SEE IF 60 BOF'S/SEC
     SETB P3.7       ;EXTERNAL SOFTWARE BOF TEST POINT
     CLR  P3.7       ;  "
     CLR  CNT12           ;USED TO CONTROL 12 BYTES XFER FROM FIFO
     MOV  DPSCTPL,DPSCTRL      ;XFR DP CTR TO DP CT PREV
     MOV  DPSCTPH,DPSCTRH      ;        "
     MOV  EOFCTPL,EOFCTRL      ;XFR EOF CTR TO EOF CT PREV
     MOV  EOFCTPH,EOFCTRH      ;     "
     MOV  FCERRSP,FCERRS ;MOVE PRESENT FRAME CT ERRS TO PREVIOS
     ; FIFO IS RESET BY HARDWARE @ BOF TIME
     MOV  FIFOPTR,#DATBUF+0   ;INIT FIFO READ DATA STORAGE PTR

MAIN1:    JNB  FFEMPN,MAIN2   ;DOES FIFO HAS SOMETHING IN IT?
     JB   CNT12,MAIN2    ;GET ONLY 12 BYTES FROM FIFO
     ACALL     GETFIFO         ;
     MOV  A,FIFOPTR ;
     CJNE    A,#DATBUF+1,MAIN2 ;IS THIS THE FIRST BYTE FROM
FIFO+1?
     ACALL     FRM_CT_COMP    ;YES - CONTAINS FRAME NUMBER
```

```
MAIN2:     JNB  ONEPPS,MAIN      ;
     SETB P1.3       ;EXTERNAL 1PPS TEST POINT
     CLR  P1.3
     CLR  ONEPPS          ;
     ACALL      BOFTEST
;         JNB    BUSBUSYN,$
     CLR  BUSBUSYN
     ACALL      GET_VID_ADR     ;
     ACALL   DUMP_TO_MCB ;
     SETB BUSBUSYN
     SJMP MAIN             ;
;*****************************************************************
**********
BOFTEST:
     MOV  A,BOFCTR2
     CJNE A,#60,BOFTEST_0      ;
     SJMP BOFTEST_E
BOFTEST_0:
     CLR  BOFLOCK
BOFTEST_E:
     MOV  BOFCTR2,#0      ;
     RET
;*****************************************************************
**********
GET_VID_ADR:
     MOV  BUS_ADR,#1FH    ;VIDEO ADDRESS ON THE MCB
     ACALL      GET_CMD         ;GET THE VIDEO ADDR FROM THE MCB
     JZ   GET_VID_ADR_E  ;ADDRESS SET TO '0',"INVALID" EXIT
     ACALL      GET_ADR         ;
     ANL  P1,#11111000B  ;MASK OUT THE INPUT VIDEO ADDR BITS
     ANL  A,#00000111B   ;
     ORL  P1,A       ;SET P1 TO VIDEO ADDR.
GET_VID_ADR_E:
     RET
;.....................................................................
.........
GET_ADR:
     JB   ACC.0,VCR_1     ;
     JB   ACC.1,VCR_2     ;
     JB   ACC.2,VCR_3     ;
     JB   ACC.3,VCR_4     ;
     JB   ACC.4,VCR_5     ;
     JB   ACC.5,VCR_6     ;
     JB   ACC.6,FMTR_IN   ;
     JB   ACC.7,EXT_IN    ;
VCR_1:     MOV  A,#0       ;
     SJMP GET_ADR_EXIT    ;
VCR_2:     MOV  A,#1       ;
     SJMP GET_ADR_EXIT    ;
VCR_3:     MOV  A,#2       ;
     SJMP GET_ADR_EXIT    ;
VCR_4:     MOV  A,#3       ;
```



```
     SJMP GET_ADR_EXIT    ;
VCR_5:     MOV  A,#4       ;
     SJMP GET_ADR_EXIT    ;
VCR_6:     MOV  A,#5       ;
     SJMP GET_ADR_EXIT    ;
FMTR_IN:MOV     A,#6       ;
     SJMP GET_ADR_EXIT    ;
EXT_IN:    MOV  A,#7       ;
GET_ADR_EXIT:
     RET
;*****************************************************************
**********
FRM_CT_COMP:    ;COMPARE FRAME CTR FROM FRAME TO FRAME
          ;MUST ROLL OVER FROM 59 TO 0
          ;MUST SHIFT RAW DATA RIGHT
          ;MASK OUT PARITY AND CHECK IT
     MOV  A,FRMNUM  ;GET RAW FRAME NUMBER FROM HEADER
     RR   A         ;SHIFT RIGHT ONCE
     ANL  A,#7FH         ;MASK OUT TOP BIT
          ;NOW CHECK FOR EVEN PARITY
PARCHK:    MOV  DELAY,#8  ;SET UP TO TEST 8 BITS PARITY
     CLR  PARITY         ;0 = EVEN, 1 = ODD
     PUSH ACC       ;SAVE REG. A
PARCHK1:RRC    A                ;ROTATE A RIGHT THRU CARRY
     JNC  PARCHK2        ;TEST CARRY
     CPL  PARITY         ;CARRY SET - COMPLEMENT PARITY
PARCHK2:DJNZ   DELAY,PARCHK1  ;
     JNB  PARITY,PARCHK3 ;PARITY CHECKED EVEN?
     INC  FCERRS         ;ODD PARITY - INCREMENT FRAME CNT ERRORS
PARCHK3:MOV    A,FRMNUMP ;GET PREVIOUS FRAME NUMBER
     ADD  A,#1       ;ADD 1 TO IT
     CJNE A,#60,FCC_0    ;EQ 60?
     MOV     A,#0        ;ROLLOVER TO 0
FCC_0:  MOV    FRMNUMP,A ;
     POP  ACC        ;
     ANL  A,#3FH         ;NOW MASK OUT PARITY BIT
     CJNE A,FRMNUMP,FCC_1     ;COMPARE PRESENT FRAME NUM + 1 TO
PREV.
     SJMP FCC_E          ;NO ERROR - GOTO EXIT
FCC_1:  INC    FCERRS          ;INCREMENT FRAME CTR COMPARE ERRORS
FCC_E:  MOV    FRMNUMP,A ;STORE PRESENT FRAME NUM TO PREV
     RET
;*****************************************************************
**********
GETFIFO:
     MOV  R0,FIFOPTR     ;
     CLR  FFRDN          ;READ SIGNAL TO THE FIFO
     MOV  @R0,P0         ;FIFO AT PORT P0 TO DATA BUFFER
     SETB FFRDN          ;RESTORE THE READ HI
     INC  FIFOPTR        ;POINT TO NEXT BYTE
     MOV  A,FIFOPTR ;
     CJNE A,#DATBUF+12,GETOUT ;
```

```
     SETB CNT12
GETOUT:   RET
;*****************************************************************
***********
DUMP_TO_MCB:
     MOV  A,FCERRSP ;GET FRAME CNT ERRS PREVIOUS
     ANL  A,#01111111B   ;MASK OUT LOCK BIT
     MOV  C,BOFLOCK ;MOVE LOCK BIT TO 'C'
     MOV  ACC.7,C        ;PUT IT IN 'A'
     MOV  FCERRSP,A ;STORE IT BACK WITH LOCK BIT
     MOV  DATPTR,#DATBUF+1;SET POINTER TO BEGIN OF DATA BUFFER
     MOV  BUS_ADR,#20H   ;START OF MCB VIDEO DATA
DUMP1:    MOV  R0,DATPTR ;
     MOV  A,@R0          ;XFR BYTE TO BUFFER REG CONNECTED TO
PORT 2
     ACALL     SEND_MON  ;SEND DATA TO MCB uP
     INC  BUS_ADR        ;POINT TO NEXT MCB ADDRESS
     INC  DATPTR         ;POINT TO NEXT BUFFER BYTE
     MOV  A,DATPTR  ;
     CJNE A,#DATBUF+18,DUMP1 ;TEST END OF BUFF,IF NOT-GET ANOTHER
BYTE
     MOV  FCERRS,#0 ;CLEAR FRAME COUNT ERRORS
     RET
;********************** INTERUPT ROUTINES
***************************
EXTERNAL_INT_0:          ;*** BOF ENTRY PT ***
     PUSH ACC
     PUSH PSW
     PUSH 00H
     CLR  C
     MOV  A,TH2          ;GET TIMER 2 HI BYTE
     CJNE A,TH2_P,EXT_0_0     ;COMPARE WITH PREVIOUS
VALUE-OK?-CONTINUE
     INC  BOFCTR         ;
     MOV  A,BOFCTR  ;
     CJNE A,#9,EXT_0_1
     MOV  TH2,#0FFH
     MOV  TL2,#80H
EXT_0_1:CJNE   A,#10,EXT_0_EXIT;TEST TO SEE IF 10 GOOD BOF'S HAVE
OCCURRED
     MOV     TL2,#080H   ;SYNCHRONIZE TIMER 2 & 1PPS TO BOF'S
     SETB BOFRM          ;BOF'S WILL NOW OCCUR JUST BEFORE 1PPS
PULSES
     SETB BOFLOCK        ;
     MOV  LOCKCTR,#5     ;
     MOV  BOFCTR,#9 ;
     SJMP EXT_0_EXIT

EXT_0_0:DJNZ   LOCKCTR,EXT_0_2     ;TAKES 5 UNTIMED BOF'S TO GO
OUT OF LOCK
     MOV  BOFCTR,#0 ;
     CLR  BOFLOCK
```



```
EXT_0_EXIT:
     MOV  TH2,#0FFH
     MOV  TL2_P,TL2 ;SAVE THE PREVIOUS TIMER 2 VALUE
     MOV  TH2_P,TH2
EXT_0_2:MOV    DELAY_I,#5     ;WAIT HERE FOR 10 uS IN CASE A
GLITCH HAS
     DJNZ DELAY_I,$ ;COME IN
     CLR  IE0            ;CLEAR THE INTERRUPT GLITCH
     POP  00H
     POP  PSW
     POP  ACC
     RET
;*****************************************************************
***********
EXTERNAL_INT_1:     ;*** DP SYNC ENTRY PT ***
     PUSH ACC
     PUSH PSW
     PUSH 00H
     CLR  C
     MOV  A,DPSCTRL ;
     ADD  A,#1           ;
     MOV  DPSCTRL,A ;
     JNC  EXT_1_E        ;
     INC  DPSCTRH        ;
     CLR  C              ;
EXT_1_E:POP    00H
     POP  PSW
     POP  ACC
     RET
;*****************************************************************
***********
INTERNAL_INT_2:     ;*** GENERATE 1PPS ***
     PUSH ACC
     PUSH PSW
     PUSH 00H
     ANL  T2CON,#0FH          ;SOFTWARE CLEAR TIMER 2 INTERRUPTS
     DJNZ THIRD_TIMER_BYTE,OUT     ;PSEUDO 3RD TIMER BYTE FOR TF0
     MOV  THIRD_TIMER_BYTE,#60     ;60 TIMER 0 OVERFLOWS = 1 SEC.
     SETB ONEPPS              ;
     INC  TIK_CTR             ;
OUT:    POP    00H
     POP  PSW
     POP  ACC
     RET
;*****************************************************************
**********
EXTERNAL_INT_3:     ;*** EOF ENTRY PT ***
     PUSH ACC
     PUSH PSW
     PUSH 00H
     CLR  C                   ;
     MOV  A,EOFCTRL ;
```

```
     ADD  A,#1       ;
     MOV  EOFCTRL,A ;
     JNC  EXT_3_E         ;
     INC     EOFCTRH           ;
     CLR  C          ;
EXT_3_E:
     MOV  TL1,#0FFH ;SET TO MAX VALUE
     MOV  TH1,#0FFH ;SET TO MAX VALUE THIS SETS UP TIMER 1
     MOV  DELAY_I,#5      ;
     DJNZ DELAY_I,$       ;DELAY = 5 x 2 INSTRUCTIONS = 10
MICROSECONDS
     SETB TR1        ;SET TIMER 1 TO RUN
     POP  00H
     POP  PSW
     POP  ACC
     RET
;******************* VARIOS SUBROUTINES
*******************************
     ;GET COMMAND FROM MCB MICRO'S MCB RELATIVE ADDRESS SPACE
     ; ADDRESS IS IN "BUS_ADR" AND DATA IS RETURNED IN REG "A"
GET_CMD:
     ;MOV      DELAY,#255 ;100 MICROSEC TIMEOUT
GC0: ;JB  BUSBUSYN,GC00  ;TEST IF uP BUS IS BUSY
     ;DJNZ      DELAY,GC0 ;STILL BUSY!
     ;SJMP     GC40        ;uP BUS IS STILL BUSY,EXIT
GC00:      ;CLR BUSBUSYN  ;BUS IS BUSY BY THIS PROCESSOR
     MOV  P2,BUS_ADR      ;OUTPUT ADDRESS TO PORT 2
     CLR  DAT_ADR         ;SIGNAL TO MCB MICRO ADDRESS IS ONLINE
     MOV  DELAY,#255      ;SETUP FOR 100 MICROSEC TIME IF NO ACK
GC1:    JNB     MCB_ACK,GC2    ;IF MCB ACK - GOTO GC2
     DJNZ DELAY,GC1 ;NO MCB ACK AT TIMEOUT - ESCAPE
     MOV  A,#0BDH         ;NO COMMAND
     MOV  P2,#0FFH  ;SET PORT 2 TO INPUT
     SETB DAT_ADR         ;RESTORE DAT_ADR
     SJMP GC40

GC2:    SETB    DAT_ADR        ;TELL MCB MICRO TO PUT DATA ON THE
BUS
     MOV  P2,#0FFH  ;SET PORT 2 TO INPUT
     MOV  DELAY,#255
GC3: JB       MCB_ACK,GC4 ;WAIT TILL ACK LINE GOES HI OR TIMEOUT
     DJNZ DELAY,GC3 ;
     MOV  A,#0BDH         ;NO COMMAND
     SJMP GC40

GC4: MOV  A,P2       ;GET COMMAND FROM PORT 2
GC40:      MOV  P2,#0FFH  ;SET PORT 2 TO INPUT
     MOV  DELAY,#25
     DJNZ DELAY,$         ;WAIT FOR ~50 uS
     ;SETB      BUSBUSYN  ;RELINQUISH uP BUS
     RET
```



```
;*****************************************************************
*******
     ;SEND MONITOR BYTE TO MCB REL. ADDRESS IN MCB MICRO
     ; ADDRESS IS IN "BUS_ADR" AND DATA IS IN REG "A"
SEND_MON:
     ;MOV      DELAY,#255 ;100 MICROSEC TIMEOUT
SEND0:     ;JB  BUSBUSYN,SEND00      ;TEST IF uP BUS IS BUSY
     ;DJNZ      DELAY,SEND0
     ;SJMP     SEND5       ;uP BUS IS STILL BUSY,EXIT
SEND00:    ;CLR BUSBUSYN  ;ASSERT BUS BUSY BY THIS uP
     CLR  CMD_MON         ;SET TO MONITOR MODE
     MOV  P2,BUS_ADR      ;OUTPUT ADDRESS TO PORT 2
     CLR  DAT_ADR         ;SIGNAL MCB MICRO - ADDRESS ONLINE
     MOV  DELAY,#255      ;SETUP FOR 100 MICROSEC TIMEOUT IF NO
ACK
SEND1:     JNB  MCB_ACK,SEND2  ;IF MCB ACK - GOTO SEND2
     DJNZ DELAY,SEND1     ;NO MCB ACK AT TIMEOUT - ESCAPE
     SJMP SEND4

SEND2:     MOV  P2,A       ;OUTPUT DATA TO MICRO ON PORT 2
     SETB DAT_ADR         ;TELL MCB MICRO DATA IS AVAILABLE
     MOV  DELAY,#255      ;100 MICROSEC TIMEOUT IF ACK DOESN'T GO
HI
SEND3:     JB   MCB_ACK,SEND4  ;MCB ACK RETURNS HI - MCB MICRO - I
GOT IT
     DJNZ DELAY,SEND3     ;NO MCB ACK - ESCAPE
SEND4:  MOV     P2,#0FFH  ;SET PORT 2 TO INPUT
SEND5:  SETB    CMD_MON         ;
     MOV  DELAY,#25
     DJNZ DELAY,$         ;WAIT FOR ~50 uS
     ;SETB      BUSBUSYN  ;RELINQUISH uP BUS
     RET
;*****************************************************************
******
     END
```

SOURCE CODE FOR MCB PROCESSOR

```
     NAME MCB

     ;MCB.SRC
     ;ORIGINALLY PROCESSOR DATA SET
     ;WRITTEN BY WAYNE M. KOSKI
     ;LAST REVISION: SEPTEMBER 28, 1987
     ;MODIFIED  BY   P.  DOOLEY  FOR  INTEL   ASSEMBLER  AND  MK2
CONTROLLER
     ;10-31-89,1-29-90,2-9-90,9-25-90

     $ INCLUDE (87C51FA.INC)

     ;THE "SETPCON" MACRO IS USED TO DOUBLE THE BAUD RATE OF THE
     ;SERIAL PORT
     ;ENTRY     NONE
     ;AFFECTS   REGISTERS: PCON


SETPCON   MACRO
     DB   75H,87H,80H     ;SET THE DOUBLE BAUD RATE BIT
     ENDM

     ;THE "SET2CON" MACRO IS USED TO TURN ON TIMER 2
     ;ENTRY     NONE
     ;AFFECTS   REGISTERS: T2CON, RCAP2H, RCAP2L

SET2CON   MACRO
     DB   75H,0CAH,0      ;CLEAR COUNTERS  (RCAP2L)
     DB   75H,0CBH,0      ;                (RCAP2H
     DB   75H,0C8H,4      ;TURN ON TIMER 2 (T2CON)
     ENDM

     ;THE  XORB  MACRO IS  USED TO  EXCLUSIVLY  OR ARG1  AND ARG2
TOGETHER
     ;AND IT PLACES THE RESULT IN ARG3, WHERE ARG1,ARG2, AND ARG3
ARE
     ;SINGLE BITS
     ;ENTRY     BITS: ARG1,ARG2,ARG3
     ;AFFECTS   BITS: CARRY FLAG, ARG3

XORB MACRO     ARG1,ARG2,ARG3
MBIT1     BIT  ARG1
MBIT2     BIT  ARG2
     MOV  C,ARG1
     ANL  C,/MBIT2
     MOV  ARG3,C
     MOV  C,ARG2
     ANL  C,/MBIT1
     ORL  C,ARG3
     MOV  ARG3,C
     ENDM
```



```
     ;DATA DEFINITION AREA
     ;THE FIRST TWO LOCATIONS ARE INPUTTED BY THE USER
     ;DECLARE ALL THESE PUBLIC TO LINK WITH OTHER PROGRAMS

PUBLIC    RES1,RES2,RES3,RES4,RES5,RES6,DATRP1,DATRP2
PUBLIC    MONRP1,MONRP2,MOD1,MOD2,CMDAD1,CMDAD2,CMDDT1,CMDDT2
PUBLIC    ADDPE1,ADDPE2,DATPE1,DATPE2,BADSY1,BADSY2
PUBLIC    MDTPE1,MDTPE2,NUMB1,NUMB2,DATCO1,DATCO2,MONCO1,MONCO2
PUBLIC    BLOCK1,BLOCK2,COUNT1,COUNT2,EBASE1,EBASE2,IBASE1,IBASE2
PUBLIC    TIME1,TIME2,TIME3,BUFFC,BUFFN

MODEL     EQU  'E'       ;SMALL MODEL OF MCB CTLR
REV  EQU  'A'       ;THIS REV LEVEL
BASE EQU  7FF0H          ;DEFAULT ADDRESS SPACE
POINTS    EQU  16        ;DEFAULT NUMBER OF MON/CMD POINTS
RES1 DATA 30H       ;RESERVED            ;BE-15
RES2 DATA 31H             ; "    "
RES3 DATA 32H       ; "    "                    ;BE-14
RES4 DATA 33H             ; "    "
RES5 DATA 34H       ; "    "                    ;BE-13
RES6 DATA 35H             ; "    "
DATRP1    DATA 36H       ;COMMAND DEVICE NO RESPONSE   ;BE-12
DATRP2    DATA 37H       ; "    "
MONRP1    DATA 38H       ;MONITOR NO RESPONSE          ;BE-11
MONRP2    DATA 39H       ; "    "
MOD1 DATA 3AH       ;MODEL/REVISION               ;BE-10
MOD2 DATA 3BH       ; "    "
CMDAD1    DATA 3CH       ;COMMAND ADDRESS OF LAST MESSAGE;BE-9
CMDAD2    DATA 3DH       ; "    "
CMDDT1    DATA 3EH       ;COMMAND DATA OF LAST MESSAGE ;BE-8
CMDDT2    DATA 3FH       ; "    "
ADDPE1    DATA 40H       ;ALL ADDRESS PARITY ERRORS    ;BE-7
ADDPE2    DATA 41H       ; "    "
DATPE1    DATA 42H       ;ALL DATA PARITY ERRORS       ;BE-6
DATPE2    DATA 43H       ; "    "
BADSY1    DATA 44H       ;BAD SYNC            ;BE-5
BADSY2    DATA 45H       ; "    "
MDTPE1    DATA 46H       ;MY DATA PARITY ERRORS        ;BE-4
MDTPE2    DATA 47H       ; "    "
NUMB1     DATA 48H       ;"N"                 ;BE-3
NUMB2     DATA 49H            ; DEVICE ID
DATCO1    DATA 4AH       ;COMMAND COUNT            ;BE-2
DATCO2    DATA 4BH       ; "    "
MONCO1    DATA 4CH       ;MONITOR COUNTS               ;BE-1
MONCO2    DATA 4DH       ; "    "
BLOCK1    DATA 4EH       ;START OF BLOCK               ;BE-0
BLOCK2    DATA 4FH       ; "    "
COUNT1    DATA 50H       ;ADDRESS SPACE COUNT
COUNT2    DATA 51H       ;ADDRESS SPACE COUNT
EBASE1    DATA 52H       ;END OF ADDRESS SPACE
EBASE2    DATA 53H       ;END OF ADDRESS SPACE
IBASE1    DATA 54H       ;INTERNAL ADDR
```

```
IBASE2    DATA 55H        ;END OF ADDRESS SPACE
TIME1     DATA 56H        ;ONE MINUTE SYNC TIMER LSB
TIME2     DATA 57H        ;ONE MINUTE SYNC TIMER MSB
TIME3     DATA 58H        ;5 SECOND READ ID BYTE TIMER
BUFFC     DATA 59H        ;CURRENT BUFFER INDEX VALUE
BUFFN     DATA 5AH        ;NEXT BUFFER INDEX VALUE
DELAY     DATA 5BH        ;STOREAGE FOR SHORT TIMING LOOPS
TEMP DATA 5CH        ;TEMP VARIABLE
BUFBASE   DATA 80H        ;CMD/MON BUFFER BASE ADDR
RCVEN     BIT  P1.0       ;XMIT TO COMPUTER ENABLE
MSG  BIT  P1.1       ;VALID MSG RCV STATUS
BUSY BIT  P1.2       ;INTERFACE ACTIVE STATUS
DOUT BIT  P1.3       ;INTERFACE TRANSMITTING STATUS
PARX BIT  P1.4       ;PARITY ERROR DETECTED STATUS
DAT_ADR   BIT  P3.2       ;ADDR & DATA TO SLAVE MICRO & ACK FROM
MCB_ACK   BIT  P3.5       ;ACK TO CTL MICRO
CMD_MON   BIT  P3.7       ;COMMAND/_MONITOR FLAG TO MICRO
BIT0 BIT  20H.0           ;RESULT OF THE XORB MACRO FUNCTION
BIT1 BIT  20H.1           ;TEMP STORAGE LOCATIONS FOR THE
BIT2 BIT  20H.2           ;XORB MACRO FUNCTION
BIT3 BIT  20H.3           ;STORED   RESULT   OF   THE   XORB   MACRO
FUNCTION
BIT4 BIT  20H.4           ;RESULT   OF THE   LOWER ADDR   BOUND CHECK
0=PASS
BIT5 BIT  20H.5           ;RESULT   OF THE   UPPER ADDR   BOUND CHECK
1=PASS
BIT6 BIT  20H.6           ;ID BYTE REQUEST FLAG
BIT7 BIT  20H.7           ;XMIT FLAG  1=CAN XMIT
BIT8 BIT  21H.0           ;SYNC CARRY FLAG
BIT9 BIT  21H.1           ;SIZE ERROR FLAG
;*************************************************************
**********
     ORG  0
START:    AJMP START1          ;JMP OVER INT VECTORS
;*************************************************************
**********
     ;SERIAL PORT INTERRUPT VECTOR
     ORG  23H
     AJMP SERP      ;SERIAL PORT INT ROUTINE
;*************************************************************
**********
     ;EXTERNAL INTERRUPT 0 VECTOR
     ORG  03H
     ACALL     CTL_REQ
     RETI
;*************************************************************
**********
     ;TIMER 2 INTERRUPT VECTOR
     ORG  2BH
     AJMP TIMER           ;TIMER 2 INT ROUTINE
;*************************************************************
**********
```



```
     ;START OF MAIN PROGRAM
     ORG  30H        ;START IT HERE
START1:   ACALL      INIT       ;INITIALIZE HARDWARE

LOOP:     MOV  SP,#7            ;RESET STACK POINTER
     MOV  PSW,#0          ;RESET BANK = 0
     ACALL     SYNC       ;WAIT TILL WE SYNC UP
     SETB BUSY       ;CLEAR LIGHTS
     SETB DOUT       ;  "     "
     SETB PARX            ;  "     "
     ACALL     SPIN       ;GET UPPER ADDRESS
     MOV  BIT3,C          ;SAVE PARITY ERROR
     MOV  B,A        ;SAVE IN REG B
     CLR  ACC.7           ;CLR CMD/MON BIT
     JNZ  START2          ;CONTINUE IF >255
     AJMP STATUS          ;IF <=255

START2:   MOV  R3,A       ;SAVE ADH IN R3
     ACALL     SPIN       ;GET LOWER ADDRESS
     MOV  R2,A       ;SAVE IN REG R2
     ACALL     ADDCK           ;CHECK IF ADDR IS IN RANGE
     JB   BIT0,PARCI      ;IF ADDR PE
     JB   BIT3,PARCI      ;IF ADDR PE
     JB   BIT4,NOTUS      ;OUT OF RANGE
     JNB  BIT5,NOTUS      ;OUT OF RANGE
     SETB RCVEN           ;ENABLE RCV BUS
     CLR  BUSY       ;SET INTERFACE BUSY STATUS LIGHT
     JB   B.7,CMD         ;GOTO COMMAND ROUTINE IF SET
     AJMP MON        ;GOTO MONITOR ROUTINE

PARCI:    CLR  PARX       ;SET PARITY ERROR STATUS LIGHT
     MOV  R0,#41H         ;ALL ADDR PE COUNTER
     ACALL     INCRE           ;ADD ONE
NOTUS:    MOV  B,#40
     DJNZ B,$        ;DELAY ABOUT 87US
     CLR  RCVEN           ;TURN OFF RCV BUS DRIVER
     ACALL     SPIN       ;GET DATA HIGH
     MOV  BIT3,C          ;SAVE PARITY
     ACALL     SPIN       ;GET DATA LOW
     JB   BIT3,PARCI1     ;IF DATA PE
     JB   BIT0,PARCI1     ;IF DATA PE
     AJMP LOOP       ;GET NEXT MESSAGE

PARCI1:   CLR  PARX       ;SET PARITY ERROR STATUS LIGHT
     MOV  R0,#43H         ;ALL DATA PE COUNTER
     ACALL     INCRE           ;ADD ONE
     AJMP LOOP

CMD: MOV  R2,00H          ;MOVE REL ADDRESS   BANK 0 R0 -> R2
     MOV  R3,01H          ;             BANK 0 R1 -> R3
     MOV  DPH,IBASE1      ;IS IT AN INTERNAL COMMAND?
     MOV  DPL,IBASE2
```

```
     ACALL     SUBT
     JC   CMD0            ;EXTERNAL COMMAND!
     AJMP CMD4            ;INTERNAL COMMAND!

CMD0:     ACALL     SPIN      ;GET UPPER DATA BYTE
     MOV  BIT3,C          ;SAVE PE
     MOV  R5,A       ;SAVE IT
     ACALL     ACKN      ;XMIT ACKNOWLEDGE
     ACALL     SPIN      ;GET LOWER DATA BYTE
     MOV  R4,A       ;SAVE IT
     JB   BIT3,CMD1 ;IF PARITY ERROR
     JNB  BIT0,CMD2 ;NO PARITY ERROR
CMD1:   ; PARITY ERROR EXIT POINT
     CLR  PARX       ;SET PARITY ERROR STATUS LIGHT
     MOV  R0,#47H         ;POINT TO DATA PE COUNTER
     ACALL     INCRE          ;INCREMENT IT
     ACALL     NACK      ;SEND NEGATIVE ACK
     AJMP LOOP

CMD2:   ; NO PARITY'S - CONTINUE
     MOV  A,R2       ;LOAD ADDR LOW
     MOV  B,R3       ;LOAD ADDR HIGH
     SETB B.7        ;SET DATA REQ
     CLR  B.6        ;SET TO WRITE
     SETB B.4        ;ENABLE BUFFERS
     PUSH B
     ADD  A,#BUFBASE     ;POINT TO CMD/MON BUFFER (RA + B0H)
     PUSH 00H        ;PUSH R0
     MOV  R0,A       ;R0 CONTAINS BUFBASE
     MOV  @R0,04H         ;STORE LO BYTE IN CMD/MON BUFFER PT'D BY
R4
     POP  00H        ;POP R0
     POP  ACC
     SETB ACC.5
     ACALL     DC1       ;SEND ACKNOWLEDGE
     AJMP CMD3       ;
     ; OLD EXIT POINT FOR NO RESPONSE FROM EXTERNAL DEVICES
     MOV  3CH,R3          ;SAVE ADDRESS BYTES
     MOV  3DH,R2          ;  "    "
     MOV  3EH,R5          ;SAVE DATA BYTES
     MOV  3FH,R4          ;  "    "
     MOV  R0,#37H         ;INC COMMAND NO RESPONSE
     ACALL     INCRE
     AJMP LOOP

CMD3:     ; SUCESSFUL EXIT, CMDS STORED AWAY,NO PARITY'S
     MOV  R0,#4BH         ;POINT TO CMD COUNT
     ACALL     INCRE          ;INC CMD RCV COUNTER
     MOV  3CH,R3          ;SAVE ADDRESS BYTES
     MOV  3DH,R2          ;  "    "
     MOV  3EH,R5          ;SAVE DATA BYTES
     MOV  3FH,R4          ;  "    "
```



```
     AJMP LOOP

CMD4:   ; INTERNAL CMD'S ENTRY PT.
     ACALL     SPIN      ;GET UPPER CMD BYTE
     MOV  BIT3,C          ;SAVE PARITY BIT
     MOV  R5,A       ;STORE BYTE
     ACALL     ACKN      ;XMIT ACKNOWLEDGE
     ACALL     SPIN      ;GET LOWER CMD BYTE
     JB   BIT3,CMD1 ;IF PARITY ERROR
     JC   CMD1       ;IF PARITY ERROR
     MOV  R4,A       ;STORE BYTE
     ACALL     DC1       ;SEND ACKNOWLEDGE
     MOV  DPTR,#CMD5      ;INDEX INTO INTERNAL COMMAND POINT
     MOV  A,R0
     MOV  B,#4
     MUL  AB
     JMP  @A+DPTR

CMD5:     MOV  R0,#30H         ;BE-15
     AJMP LOAD0

     MOV  R0,#32H         ;BE-14
     AJMP LOAD0

     MOV  R0,#34H         ;BE-13
     AJMP LOAD0

     MOV  R0,#36H         ;BE-12
     AJMP LOAD0

     MOV  R0,#38H         ;BE-11
     AJMP LOAD0

     AJMP CMD3       ;BE-10

     NOP
     NOP

     MOV  R0,#3CH         ;BE-09
     AJMP LOAD0

     MOV  R0,#3EH         ;BE-08
     AJMP LOAD0

     MOV  R0,#40H         ;BE-07
     AJMP LOAD0

     MOV  R0,#42H         ;BE-06
     AJMP LOAD0

     MOV  R0,#44H         ;BE-05
     AJMP LOAD0
```

```
     MOV  R0,#46H          ;BE-04
     AJMP LOAD0

     AJMP CMD3       ;BE-03

     NOP
     NOP

     MOV  R0,#4AH          ;BE-02
     AJMP LOAD0

     MOV  R0,#4CH          ;BE-01
     AJMP LOAD0

     AJMP CMD3       ;BE-00

     NOP
     NOP

LOAD0:    MOV  @R0,05H
     INC  R0
     MOV  @R0,04H
     AJMP CMD3

MON: CLR  ES         ;DISABLE SERIAL PORT
     MOV  R2,00H          ;SAVE REL ADDRESS   BANK 0 R0 -> R2
     MOV  R3,01H          ;                   BANK 0 R1 -> R3
     MOV  DPH,IBASE1      ;IS IT AN INTERNAL MONITOR
     MOV  DPL,IBASE2
     ACALL     SUBT
     JNC  MON2       ;GOTO INTERNALS
     MOV  A,R2       ;GET ADDR LOW
     MOV  B,R3       ;GET ADDR HIGH
     CLR  B.7        ;NO DATA REQ
     SETB B.6        ;SET FOR READ
     SETB B.4        ;ENABLE BUFFERS
     SETB ACC.7           ;SET DATA REQ
     SETB ES         ;ENABLE SERIAL PORTS
     ACALL     SPIN       ;GET DATA HIGH
     MOV  BIT3,C          ;SAVE PARITY
     ACALL     ACKN       ;XMIT ACKNOWLEDGE
     ACALL     SPIN       ;GET DATA LOW
     JB   BIT3,MON0 ;IF PARITY ERROR
     JNB  BIT0,MONIN0     ;IF NO PARITY ERROR
MON0:     CLR  PARX       ;LIGHT PARITY LIGHT
     MOV  R0,#47H          ;POINT TO DATA PE COUNTER
     ACALL     INCRE          ;INCREMENT IT
MONIN0:   AJMP    MONIN3
     ; OLD ENTRY POINT FOR EXTERNAL DEVICE NOT RESPONDING
     ACALL     NACK       ;SEND NEGATIVE ACKNOWLEDGE
     MOV  R0,#39H          ;INC MONITOR NO RESPONSE
     ACALL     INCRE
```



```
     AJMP LOOP

MONIN3: PUSH   ACC
     MOV  A,R2       ;GET LO BYTE OF REL. ADDR.
     ADD  A,#BUFBASE      ;PT. TO ADDR. IN CMD/MON BUFFER
     PUSH 00H        ;PUSH R0
     MOV  R0,A
     MOV  02H,@R0         ;MONITOR DATA FROM CMD/MON BUFFER -> R2
     MOV  R3,#0           ;ZERO UPPER BYTE OF MONITOR DATA
     POP  00H        ;POP R0
     POP  ACC
MON1:     CLR  DOUT       ;SET MONITOR MSG LIGHT
     MOV  A,R3       ;GET UPPER MONITOR DATA BYTE
     ACALL     SPOOUT         ;SEND UPPER DATA BYTE
     MOV  A,R2       ;GET LOWER MONITOR DATA BYTE
     ACALL     SPOOUT         ;SEND LOWER DATA BYTE
     MOV  R0,#4DH          ;POINT TO DATA COUNTER
     ACALL     INCRE
     AJMP LOOP

MON2:     MOV  DPTR,#MON3      ;INDEX INTO INTERNAL MONITOR POINT
     MOV  A,R0
     MOV  B,#6
     MUL  AB
     JMP  @A+DPTR

MON3:     MOV  R3,30H          ;BE-15
     MOV  R2,31H
     AJMP MON4

     MOV  R3,32H          ;BE-14
     MOV  R2,33H
     AJMP MON4

     MOV  R3,34H          ;BE-13
     MOV  R2,35H
     AJMP MON4

     MOV  R3,36H          ;BE-12
     MOV  R2,37H
     AJMP MON4

     MOV  R3,38H          ;BE-11
     MOV  R2,39H
     AJMP MON4

     MOV  R3,3AH          ;BE-10
     MOV  R2,3BH
     AJMP MON4

     MOV  R3,3CH          ;BE-09
     MOV  R2,3DH
```

```
      AJMP MON4

      MOV  R3,3EH          ;BE-08
      MOV  R2,3FH
      AJMP MON4

      MOV  R3,40H          ;BE-07
      MOV  R2,41H
      AJMP MON4

      MOV  R3,42H          ;BE-06
      MOV  R2,43H
      AJMP MON4

      MOV  R3,44H          ;BE-05
      MOV  R2,45H
      AJMP MON4

      MOV  R3,46H          ;BE-04
      MOV  R2,47H
      AJMP MON4

      ACALL     NUMBR           ;BE-03
      NOP
      NOP
      AJMP MON4

      MOV  R3,4AH          ;BE-02
      MOV  R2,4BH
      AJMP MON4

      MOV  R3,4CH          ;BE-01
      MOV  R2,4DH
      AJMP MON4

      MOV  R3,4EH          ;BE-00
      MOV  R2,4FH
MON4: SETB ES              ;ENABLE SERIAL PORT INTERRUPT
      ACALL     SPIN       ;GET DATA HIGH
      MOV  BIT3,C          ;SAVE PARITY
      ACALL     ACKN       ;XMIT ACKNOWLEDGE
      ACALL     SPIN       ;GET DATA LOW
      JB   BIT3,MON5 ;IF PARITY ERROR
      JNB  BIT0,MON6 ;IF NO PARITY ERROR
MON5: CLR  PARX            ;LIGHT PARITY LIGHT
      MOV  R0,#47H         ;POINT TO DATA PE COUNTER
      ACALL     INCRE           ;INCREMENT IT
MON6: AJMP MON1            ;NOW SEND IT

STATUS:   MOV  R3,A        ;STORE ADH IN R3
      JNB  BIT3,STAT1      ;PE?
      AJMP START2          ;IF PARITY ERROR
```



```
STAT1:    MOV  A,NUMB2          ;GET "N"
      JB   P,STAT2         ;"N" PE?
      AJMP START2          ;IF "N" PE

STAT2:    CLR  ACC.7            ;CLEAR PE BIT
      RL   A               ;*2
      PUSH B               ;SAVE B
      MOV  B,A             ;PUT "N"*2 INTO B
      ACALL     SPIN        ;GET ADL
      MOV  R2,A            ;SAVE ADL IN R2
      JNB  BIT0,STAT3      ;PE?
      POP  B               ;RESTORE STACK
      AJMP PARCI           ;IF PE

STAT3:    CJNE A,B,STAT4 ;US?
      AJMP STAT5           ;IF US

STAT4:    INC  B               ;B="N"*2+1
      CJNE A,B,STAT4A      ;US? SECOND CHECK
      AJMP STAT5           ;IF US

STAT4A:   POP  B               ;RESTORE STACK
      AJMP NOTUS           ;NOT US

STAT5:    SETB RCVEN           ;TURN ON RCV BUS
      POP  B           ;GET ADH
      JB   B.7,STCMD ;IF SET DO COMMAND
STMON:    JB   ACC.0,STMON1    ;IF UPPER "N"
      MOV  R3,COUNT1
      MOV  R2,COUNT2
      AJMP MON4

STMON1:   MOV  R3,4EH
      MOV  R2,4FH
      AJMP MON4

STCMD:    JB   ACC.0,STCMD4    ;IF UPPER
      MOV  R0,#50H
      ACALL     STCMD5
STCMD2:   MOV  @R0,05H         ;LOAD DATA
      INC  R0
      MOV  @R0,04H
      ACALL     NUM1        ;NOW RESIZE SYSTEM
STCMD3:   AJMP CMD3

STCMD4:   MOV  R0,#4EH
      ACALL     STCMD5           ;GET DATA
      JNB  BIT9,STCMD3     ;START ADDRESS TOO SMALL
      MOV  A,R5        ;GET HIGH ORDER
      CLR  C
      SUBB A,#80H          ;TOO HIGH?
      JNC  STCMD3          ;START IS TOO HIGH
```

```
     AJMP STCMD2

STCMD5:   ACALL     SPIN
     MOV  R5,A      ;SAVE CDH IN R5
     MOV  BIT3,C         ;SAVE PARITY ERROR
     CLR  BIT9
     JZ   STCMD6
     SETB BIT9
STCMD6:   ACALL     ACKN      ;XMIT ACKNOWLEDGE
     ACALL     SPIN      ;GET CDL
     JB   BIT3,STCMD7    ;IF PARITY ERROR
     JNC  STCMD8         ;IF OK
STCMD7:   POP  B          ;RESTORE STACK
     AJMP CMD1      ;IF PARITY ERROR

STCMD8:   MOV  R4,A      ;SAVE CDL IN R4
     AJMP DC1       ;SEND SECOND ACKNOWLEDGE

     ;  "INIT"  THIS ROUTINE INITIALIZES THE SERIAL AND PARALLEL
PORTS
     ;AND INTERNAL MEMORY LOCATIONS.
     ;ENTRY    NONE
     ;AFFECTS  MEMORY LOCATIONS: 30H-50H
     ;         REGISTERS: R0,DPTR,A,SCON,TH1,TMOD,TCON,IE
     ;         BITS: TR1,BIT7,BUSY,MSG,DOUT,PARX

INIT:     MOV  R0,#BUFBASE    ;INITIALIZE CMD/MON BUFFER AREA
     MOV  R1,#20H
INIT0:    MOV  @R0,#00H  ;WITH "ZERO'S"
     INC  R0
     DJNZ R1,INIT0
     MOV  R0,#30H        ;LOAD 30H-50H WITH INITIAL INFO
     MOV  R1,#10
COUL:     MOV  @R0,#0
     INC  R0
     DJNZ R1,COUL
     MOV  @R0,#MODEL
     INC  R0
     MOV  @R0,#REV
     INC  R0
     MOV  R1,#18
COUL1:    MOV  @R0,#0
     INC  R0
     DJNZ R1,COUL1
     MOV  @R0,#BASE/256
     INC  R0
     MOV  @R0,#BASE AND 0FFH
     MOV  COUNT1,#0
     MOV  COUNT2,#POINTS
COUL2:    MOV  IE,#0          ;DISABLE ALL INTERRUPTS
     CLR  RCVEN          ;TURN OFF XMIT TO MAIN COMPUTER
     ACALL     NUMBR          ;GET "DEVICE ID"
```



```
     SETPCON             ;INITIALIZE TIMER1 AND SERIAL PORT
     MOV  SCON,#0D0H     ;SET SERIAL PORT TO MODE 2, ACTIVATE RX
     MOV  TMOD,#20H      ;SET TIMER 1 TO MODE 2
     MOV  TH1,#255  ;57600 BAUD
     SETB TR1       ;START TIMER1
     SETB BIT7      ;SET XMIT FLAG
     SETB BUSY      ;CLEAR LIGHTS
     SETB MSG
     SETB DOUT
     SETB PARX
     SETB CMD_MON        ;SET PIN TO INPUT
     SETB DAT_ADR        ;SET PIN TO INPUT
     SETB MCB_ACK
     MOV  P0,#0FFH  ;SET PORT TO INPUT
     MOV  BUFFC,#60H     ;CURRENT CHAR STORAGE
     MOV  BUFFN,#60H     ;NEXT CHAR STORAGE
     SETB IT0       ;MAKE EXTERNAL INTERRUPT 0 EDGE ACTIVATED
     SETB EX0       ;ENABLE "          "      "
     SET2CON
     SETB PS        ;SERIAL PORT = HIGHEST INT
     SETB ES        ;ENABLE SERIAL PORT INTERRUPT
     SETB EA        ;ENABLE ALL INTERRUPTS PREVIOUSLY ENABLED
     RETI

DSYNC:    MOV  R0,#45H
     ACALL     INCRE

     ;    "SYNC"  THIS ROUTINE WAITS FOR THE SYNC CHAR (16H EVEN
PARITY)
     ;IF IT SEES SYNC AND INCORRECT PARITY
     ;IT INCREMENTS THE DISCARD SYNC COUNTER.
     ;ENTRY    NONE
     ;AFFECTS  REGISTER: A
     ;         BITS: CARRY FLAG,RCVEN,RI
     ;         MEMORY LOCATIONS: 3AH,3BH

SYNC:
     MOV  TIME1,#4CH     ;RESET ONE MINUTE TIMER
     MOV  TIME2,#4
     MOV  A,BUFFC        ;GOT A CHAR?
     CJNE A,BUFFN,SYNC1  ;IF CHAR
     SETB ET2       ;ALLOW TIMER 2 TO INTERRUPT
     JNB  BIT6,SYNC1     ;ID REQ?
     ACALL     NUMBR          ;TIME TO DO NUMBER
SYNC1:    ACALL     READ      ;GET CHAR
     MOV  BIT8,C         ;SAVE PARITY FLAG
     CJNE A,#16H,SYNC    ;WAIT UNTIL IT IS 16H
     JNB  BIT8,DSYNC     ;IF ODD PARITY,NOT SYNC BYTE
     SETB MSG       ;CLOCK IN CURRENT LIGHTS
     CLR  MSG       ;SET ACTIVE STATUS LIGHTS
     RET
```

```
     ;  "SPIN"    THIS ROUTINE GET A CHAR AND TESTS FOR ODD
PARITY.
     ;ENTRY     NONE
     ;AFFECTS   REGISTER: A
     ;          BITS: CARRY FLAG,BIT0,BIT1,BIT2,RCVEN,RI

SPIN:      MOV   A,BUFFC         ;GOT A CHAR?
     CJNE A,BUFFN,SPIN1   ;IF CHAR
     SETB ET2         ;ENABLE TIMER 2 TO INTERRUPT
SPIN1:     ACALL      READ       ;GET CHAR
     CPL   C
     MOV   BIT1,C
     MOV   C,P
     MOV   BIT2,C
     XORB BIT1,BIT2,BIT0
     RET

     ;  "INCRE"   THIS ROUTINE INCREMENTS A 16 BIT COUNTER BY 1
POINTED
     ;TO BY THE INDEX REGISTER R0.
     ;ENTRY     REGISTERS: R0=ADDRESS OF COUNTER LSB
     ;AFFECTS   REGISTERS: A,R0
     ;          MEMORY LOCATIONS: @R0,@R0-1

INCRE:     MOV   A,@R0
     ADD   A,#1
     MOV   @R0,A
     DEC   R0
     MOV   A,@R0
     ADDC A,#0
     MOV   @R0,A
     RET

     ;"NACK"  THIS ROUTINE SENDS THE NEGATIVE ACKNOWLEDGE TO THE
     ;MAIN COMPUTER.
     ;ENTRY     NONE
     ;AFFECTS   REGISTERS: A

NACK:      MOV   A,#15H
     AJMP SPEOUT

     ;" DC2" THIS ROUTINE SENDS THE SECOND NEGATIVE ACKNOWLEDGE
TO THE
     ;MAIN COMPUTER.
     ;ENTRY     NONE
     ;AFFECTS   REGISTERS: A

DC2: MOV   A,#12H
     AJMP SPEOUT

     ;  "ACKN"    THIS ROUTINE SENDS THE  ACKNOWLEDGE TO THE
     ;MAIN COMPUTER.
```



```
     ;ENTRY     NONE
     ;AFFECTS   REGISTERS: A

ACKN:      MOV   A,#6H
     AJMP SPEOUT

     ;   "DC1"    THIS ROUTINE SENDS THE SECOND ACKNOWLEDGE TO THE
     ;MAIN COMPUTER.
     ;ENTRY     NONE
     ;AFFECTS   REGISTERS: A

DC1: MOV   A,#11H

     ;  "SPEOUT"    THIS ROUTINE OUTPUTS CHARACTER WITH EVEN
PARITY
     ;CONTAINED IN (A) TO THE MAIN COMPUTER.
     ;ENTRY     REGISTERS: A=CHAR
     ;AFFECTS   REGISTERS: SBUF
     ;          BITS: BIT7,CARRY FLAG,TB8

SPEOUT:    MOV   C,P
     MOV   TB8,C
     PUSH ACC         ;SAVE CHARACTER
     MOV   A,#50           ;50 LOOP COUNTS
SPE01:     JB    BIT7,SPE02      ;IF READY
     DEC   A          ;A=A-1
     JZ    SPEXIT           ;IF COUNT = 0
     AJMP SPE01            ;LOOP TIME = APPROX 380US

SPE02:     POP   ACC        ;RESTORE CHARACTER
     CLR   BIT7
     MOV   SBUF,A
     RET

     ;  "SPOOUT"   THIS ROUTINE OUTPUTS CHARACTER WITH ODD PARITY
     ;CONTAINED IN (A) TO THE MAIN COMPUTER.
     ;ENTRY     REGISTERS: A=CHAR
     ;AFFECTS   REGISTERS: SBUF
     ;          BITS: BIT7,CARRY FLAG,TB8

SPOOUT:    MOV   C,P
     CPL   C
     MOV   TB8,C
     PUSH ACC         ;SAVE CHARACTER
     MOV   A,#50           ;50 LOOP COUNTS
SP001:     JB    BIT7,SP002      ;IF READY
     DEC   A          ;A=A-1
     JZ    SPEXIT           ;IF COUNT = 0
     AJMP SP001            ;LOOP TIME = APPROX 380US

SP002:     POP   ACC        ;RESTORE CHARACTER
     CLR   BIT7
```

```
     MOV  SBUF,A
     RET

SPEXIT:   POP  ACC        ;RESTORE STACK
     ACALL     COUL2          ;MINI INITIALIZE
     AJMP LOOP       ;RESTART

     ;  "ADDCK"  THIS ROUTINE CHECKS THE ADDRESS BOUNDRY AND PUT
THE
     ;RESULTING FLAGS IN BIT4, AND BIT5
     ;ENTRY     REGISTERS: R2,R3=ADDRESS TO CHECK
     ;AFFECTS   REGISTERS: DPTR
     ;          BITS: BIT4,BIT5

ADDCK:    MOV  DPH,BLOCK1      ;CHECK ADDRESS RANGE
     MOV  DPL,BLOCK2
     ACALL     SUBT
     MOV  BIT4,C          ;STORE CARRY
     MOV  DPH,EBASE1
     MOV  DPL,EBASE2
     ACALL     SUBT
     MOV  BIT5,C
     MOV  DPH,BLOCK1      ;CREATE REL ADD
     MOV  DPL,BLOCK2

     ;  "SUBT"  THIS ROUTINE SUBTRACTS WHAT CONTAINED IN R2,R3 BY
DPTR
     ;AND PLACES THE RESULT IN R0,R1
     ;ENTRY     REGISTERS: R2,R3=MINUEND DPTR=SUBTRAHEND
     ;AFFECTS   REGISTERS: A,R0,R1
     ;          BITS: CARRY FLAG

SUBT:     MOV  A,R2       ;SUBTRACT R3,R2-DPH,DPL
     CLR  C          ;CLEAR BORROW
     SUBB A,DPL
     MOV  R0,A       ;STORE ANS IN R1,R0
     MOV  A,R3
     SUBB A,DPH
     MOV  R1,A
     RET

     ;  "SERP"  THIS IS THE SERIAL PORT INTERRUPT ROUTINE
     ;ENTRY     VIA SERIAL INTERRUPT
     ;AFFECTS   MEMORY LOCATIONS: 59H,5AH,60H-6FH
     ;          BITS: TI,RI,BIT7

SERP:     JNB  TI,SERP1  ;IF NOT TI
     SETB BIT7       ;SAVE TX INT FLAG
     CLR  TI         ;CLR INT FLAG
SERP1:    JNB  RI,SERP4  ;ESCAPE IF NO RCV FLAG
     PUSH PSW        ;SAVE THESE REGS
     PUSH ACC
```



```
     PUSH 00H        ;BANK 0 R0 -> STACK
     MOV  R0,BUFFN   ;INDEX NEXT CHAR
     MOV  @R0,SBUF   ;GET CHAR AND SAVE
     XCH  A,R0       ;EXCHANGE R0,A
     ADD  A,#8       ;ADD 8
     XCH  A,R0       ;PUT BACK
     MOV  A,#0       ;PARITY=0
     JNB  RB8,SERP2 ;IF PARITY BIT = 0
     CPL  A          ;PARITY BIT = 1
SERP2:    MOV  @R0,A          ;STORE PARITY BIT
     INC  BUFFN           ;POINT TO NEXT ENTRY
     MOV  A,#68H          ;END OF BUFFER?
     CJNE A,BUFFN,SERP3
     MOV  BUFFN,#60H      ;IF END RESTART
SERP3:    CLR  RI         ;CLEAR RCV INT FLAG
     POP  00H        ;RESTORE REGS  STACK -> BANK 0 R0
     POP  ACC
     POP  PSW
SERP4:    RETI            ;LEAVE AND RESTORE INT LEVELS

     ;  "TIMER"  TIMER ROUTINE DETERMINES WHEN WE TIME OUT
     ;THE RCV ENABLE LINE AFTER 1 MINUTE OF DEAD
     ;TIME AND REQUESTS NUMBER AFTER 5 SECONDS OF DEAD TIME.
     ;ENTRY     VIA TIMER 2 INTERRUPT ROUTINE
     ;AFFECTS   MEMORY LOCATIONS: 56H-58H

TIMER:    PUSH PSW        ;SAVE REGS
     PUSH ACC
     DEC  TIME1
     MOV  A,TIME1
     JNZ  TIMER3
     DEC  TIME2
     MOV  A,TIME2
     JNZ  TIMER2
TIMER1:   ACALL     COUL2
     AJMP LOOP

TIMER2:   CLR  C
     SUBB A,#5
     JNC  TIMER1
TIMER3:   DEC  TIME3
     MOV  A,TIME3         ;TIME YET
     JNZ  TIMER4          ;IF NOT
     SETB BIT6
TIMER4:   SET2CON
     POP  ACC
     POP  PSW        ;RESTORE REGS
     RETI

     ;  "READ"  THIS ROUTINE READS A CHAR STORED IN THE BUFFER
AND
     ;ITS CORRESPONDING PARITY BIT. IF NO CHAR IN BUFFER
```

```
     ;IT WAITS FOR A CHAR AND INPUTS IT DIRECT.
     ;ENTRY              NONE
     ;AFFECTS       REGISTERS: A
     ;              MEMORY LOCATIONS: 56H-5FH
     ;              BITS: CARRY FLAG,BIT6

READ:     CLR  ES         ;DISABLE SERIAL PORT TEMP
     MOV  A,BUFFC         ;GET CURRENT CHAR LOCATION
     CJNE A,BUFFN,READ2   ;IF CHAR IN BUFFER
READ1:    SETB ET2        ;ENABLE TIMER2
     SETB EX0
     DB   53H,0C8H,84H    ;ANL T2CON,84H
     DB   43H,0C8H,4      ;ORL T2CON,4
     JNB  RI,READ1   ;SERIAL READ INTERRUPT?
     CLR  EX0
     CLR  ET2        ;DISABLE TIMER 2 INTERRUPT
     MOV  A,SBUF          ;GET CHAR
     CLR  RI         ;CLEAR INT FLAG
     MOV  C,RB8           ;GET PARITY BIT
     AJMP READ5           ;ESCAPE

READ2:    CLR  ET2        ;DISABLE TIMER 2 INT
     PUSH B          ;SAVE THESE REGS
     PUSH 00H        ;BANK 0 R0 -> STACK
     MOV  R0,A       ;GET R0 = BUFFC
     MOV  B,@R0           ;B = CHAR
     ADD  A,#8       ;POINT TO PARITY
     MOV  R0,A       ;INDEX IT
     INC  BUFFC           ;POINT TO NEXT CHAR TO UNLOAD
     MOV  A,#68H          ;END OF BUFFER
     CJNE A,BUFFC,READ3
     MOV  BUFFC,#60H      ;RESET TO START OF BUFFER
READ3:    MOV  A,@R0           ;GET PARITY
     CLR  C          ;C = 0
     JZ   READ4           ;IF A = 0
     SETB C
READ4:    MOV  A,B        ;A = CHAR
     POP  00H        ;STACK -> BANK 0 R0
     POP  B          ;RESTORE REG
READ5:    SETB ES         ;ENABLE SERIAL PORT
     RET

NUMBR:  MOV    NUMB1,#0  ;ZERO HIGH BYTE OF ID
     MOV  NUMB2,P2   ;GET ID BYTE FROM PORT 2
     MOV  R3,NUMB1   ;SET UP FOR MONITOR
     MOV  R2,NUMB2
NUM1:     MOV  A,BLOCK2   ;DETERMINE SIZE
     ADD  A,COUNT2
     MOV  DPL,A
     MOV  A,BLOCK1
     ADDC A,COUNT1
     MOV  DPH,A
```



```
     JNB  ACC.7,NUM2
     CJNE A,#80H,NUM3
     MOV  A,DPL
     JNZ  NUM3
NUM2:     MOV  EBASE2,DPL
     MOV  EBASE1,DPH
     CLR  C
     MOV  A,COUNT2
     SUBB A,#16
     MOV  IBASE2,A
     MOV  A,COUNT1
     SUBB A,#0
     MOV  IBASE1,A
     CLR  BIT6
     MOV  TIME3,#4CH
NUM3:     RET

DUMP:     PUSH ACC
     PUSH 00H        ;PUSH R0
     MOV  A,#0
DUMP1:  MOV    R0,A
     MOV  P2,@R0
     SETB P1.6
     CLR  P1.6
     INC  ACC
     JNZ  DUMP1
     POP  00H        ;POP R0
     POP  ACC
     RET
;*****************************************************************
******
;
CTL_REQ:
;    CLR  EX0        ;DISABLE EXTERNAL INTERRUPT 0
     PUSH 00H        ;"R0"
     PUSH ACC
     PUSH PSW
     MOV  A,P0       ;GET ADDRESS FROM PORT 0
     ADD  A,#BUFBASE      ;ADD IT TO THE BUFFER BASE ADDRESS
     MOV     R0,A         ;
     CLR  MCB_ACK         ;SEND ACK TO CTL MICRO
     MOV  DELAY,#255      ;TIMEOUT DELAY
     JNB  CMD_MON,CTL_M   ;JUMP IF MONITOR DATA TO BE RECEIVED
     ;*** SEND COMMAND ***
CTL_C:    JB   DAT_ADR,CTL_C0 ;WAIT FOR DAT_ADR TO GO HI
     DJNZ DELAY,CTL_C     ;TIMEOUT & DON'T HANG IF DAT_ADR NOT HI
     SJMP CTL_M1

CTL_C0:   MOV  P0,@R0          ;SEND COMMAND DATA
     SETB MCB_ACK         ;RESTORE ACK
     MOV  DELAY,#10 ;HOLD DATA FOR ~20 MICROSEC
     DJNZ DELAY,$         ;WAIT HERE
```

```
        MOV    P0,#0FFH  ;RESTORE P0 TO INPUT
     SJMP CTL_M2

CTL_M:    JB   DAT_ADR,CTL_M0 ;WAIT FOR DAT_ADR TO GO HI
     DJNZ DELAY,CTL_M
     SJMP CTL_M1

CTL_M0:   MOV  @R0,P0          ;GET MONITOR BYTE
CTL_M1:   SETB MCB_ACK
CTL_M2: POP    PSW
     POP  ACC
     POP  00H
;    SETB EX0       ;ENABLE EXTERN INT. 0
     RET

;*****************************************************************
******

     END
```