# SENTRY

# COMMUNICATION LINK

## User's Manual

FAIRCHILD
SYSTEMS TECHNOLOGY
A DIVISION OF FAIRCHILD CAMERA AND INSTRUMENT CORPORATION

# SENTRY
# COMMUNICATION LINK
## User's Manual

**Publication No: 57000003**
**Initial Release: November, 1977**


1725 Technology Drive, San Jose, California 95110

FAIRCHILD
SYSTEMS TECHNOLOGY
A DIVISION OF FAIRCHILD CAMERA AND INSTRUMENT CORPORATION

## TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

## TABLE OF CONTENTS
## (Continued)

## TABLE OF CONTENTS
## (Continued)



## LIST OF FIGURES

## TABLE OF CONTENTS
## (Continued)

## LIST OF TABLES

# SECTION 1

# FST - 2 COMMUNICATION LINK OVERVIEW

## 1.1 GENERAL

The Sentry Communication Link (referred to in this manual as Com Link) is an RS232 serial interface which is used to connect the FST-2 computer to the Integrator.

## 1.2 FEATURES

The Com Link interface allows the user the following features:

- Downloading of test programs from the Integrator to the Sentry.
- Uploading of test programs from the Sentry to the Integrator.
- Transferring of tester data (i.e., DATALOG or WRITE statements) from the Sentry to the Integrator.
- Transferring messages between the Sentry and the Integrator operators.
- Controlling the Sentry from command input from the Integrator.

## 1.3 COM LINK SPECIFICATIONS

The basic Com Link specifications are:

| | |
|---|---|
| • Standard Interface | RS232C |
| • Maximum transfer rate with direct connection | 9600 baud |
| • Maximum transfer rate with a modem | 1200 baud |
| • Maximum message size (using 8-bit characters) | 128 characters |
| • Block transfer rate (using 24- bit words) | 12,000 words/ minute |

# SECTION 2

# COMMUNICATION LINK OPERATION

This section discusses the usage of DOPSY and MASTR in the operation of the Com Link.

## 2.1 DOPSY COM LINK OPERATION

### 2.1.1 Initiating Communication with the Integrator

The command

NOTE CLO

initiates communication with the Integrator via the Com Link. A status test is issued to the Com Link to determine if line connection has been made. If the line connection has not yet been established, then the line is enabled, a connected flag is set in bit 15 of location 101B and control is returned to DOPSY.

### 2.1.2 Line Connection and Message Transmission

The command

NOTE CLO 'operator message'

initiates communcation with the Integrator as described above, and then transmits the operator message contained in quotes (either single or double quotes) to the Integrator for display on the VKT. If line connection has not been established, the software waits for 15 seconds. If line connection has still not been made, an error message is issued. An operator message may be transmitted at any time the DOPSY prompting character (*) appears on the VKT.

### 2.1.3 Line Disconnection

The command

NOTE HANG

causes the Com Link to be disconnected.

### 2.1.4 Transfer Contents from Sentry Disc to Integrator Disc

The command

FDUMP CLO 'file'

transfer the contents of the named file from the Sentry disc to the Integrator disc. String, data, or object files may be transferred. Upon successful transfer of files control is returned to the operator and an asterisk (*) is displayed on the VKT.

### 2.1.5 Transfer Contents from Integrator Disc to Sentry Disc

The command

CREATE CLI 'file' (STRING/DATA/OBJ)

transfers the contents of the named file from the integrator disc to the Sentry disc. The file may be one that has been previously uploaded to the Integrator, or a file that was created initially at the Integrator (i.e., a STRING file may be created via the HP editor). String, data, or object files may be transferred. If no file is specified the default file type is STRING. Upon successful transfer of files control is returned to the operator and an asterisk (*) is displayed on the VKT.

### 2.1.6 Enable Command Input from the Integrator

The command

SET CLI

enables command input from the Integrator. After this command has been issued, input from both the Integrator (CLI) and the Sentry keyboard (TTK) is enabled. All commands input from the Com Link are echoed on the Sentry VKT preceded by CL. Any error messages are output to both the Integrator (CLO), and the Sentry (TTP).

### 2.1.7 Disable Command Input from the Integrator

The commands

SET TTK or NOTE HANG

may be entered from the Integrator or Sentry keyboards and disables command input from the Integrator. Any Com Link hardware error also causes the command input mode to be reset.

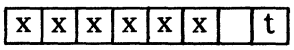
### 2.1.8 Delete a File from the Integrator Disc

The command

DELETE CLO 'file'

allows a file to be deleted from the Integrator's disc. Sentry disc files are not affected by this command.

The file to be deleted must contain the exact ASCII character string as entered in the Integrator's directory. For DOPSY file types the directory format is as follows:




xxxxxx = 6 character file name
t = file type where 0 = STRING
2 = DATA
4 = OBJECT

### 2.1.9 Operator Messages from the Com Link

Operator messages may be sent at any time from the Integrator. The message is held in memory until the operation currently being processed at the Sentry is complete, and is then displayed on the VKT preceded by the letters 'CL' to signify Com Link.

If a second message is received before the DOPSY monitor is able to display the first message, only the last message is retained. Also, since some DOPSY programs use all of the memory, and swap out the Com Link driver, the link is temporarily disabled. This occurs when using the editor, compiler or assembler.

If the command SET CLI has previously been issued, the message is displayed on the VKT and is then operated upon as a DOPSY command.

## 2.2 MASTR COM LINK OPERATION

### 2.2.1 Set System Primary Input and Output

The command

SET (input) (output)
SET (CLI 'filename') (CLO 'filename')

changes the system primary input and output to the devices specified. When the system is initially loaded, PID1 and POD1 are set to VK1 and VP1 and PID2 and POD2 are set to VK2 and VP2.

The systems PID/POD are always considered to be in source format. CLI or CLO data is read or written in TRASCII characters.

The filename must immediately follow the device mnemonic. The file is opened at the time the command is entered and closed when the EOF condition occurs or another SET command is entered. When the EOF or error condition occurs on input, PID1 or PID2 is automatically switched to VK1 or VK2, respectively. When the CLO filename starts with !, the POD is the VKT screen at the Integrator.

### 2.2.2 Set Station Primary Output

The command

USE (output) (STATn)
USE (CLO 'filename') (STATn)

changes the station POD used for FACTOR I/O and datalog output. The device specified is the device used for FACTOR I/O statements which either does not specify a device or which uses the mnemonic POD. The device specified will be the device used by the datalogger for output if no device is specified in the datalog command.

The station POD is initially set to system POD1. Loading a new test program to the station without specifying a SAVE option and/or the CLEAR command reset the station PID and POD to the system PID1 and POD1. The usage of the mnemonic POD In the USE command also resets the station primary output device to the system POD1.

If the system POD is used for datalog output, the output is always in printed format regardless of the device.

CLI is an illegal input device for a station.

When the command

USE CLO 'filename' (STATn)

is entered the system responds by asking

LOT, DEVICE and CATEGORY.

Refer to OPEN command for details.

### 2.2.3 Set Default Station Number

The command

SET STATn

establishes the default station number. Once it is entered, the station number can be omitted in commands which normally require a station number. Those commands are SN, DATALOG, LOAD, Manual Analysis commands, etc. If the station number is omitted the default station is used. Otherwise the station number entered is used.

### 2.2.4 Load a Test Program

The command

/. LOAD 'test program name' ('test program name') ('test program name') (SAVE) (STATn) (KEEP) (CLI/MTR1/MTR2)

loads a test program into memory from the specified device and attaches it to the specified station or to the default station. The next time a start request is viewed for this station the test program is executed.

If the test program is already in memory then it is attached to the station without reloading.

The default input is the device from which the system was loaded.

The SAVE option requests retaining all operator requested conditions, the values of globals, and overlay requests in use for the station. If SAVE is not specified loading a test program to a station performs the following resets:

- SN is set to 1
- GLOB1 - GLOB40 are set to 0
- SWTICH, VALUE, DATALOG are set to 0
- Datalog requests are turned off
- Analysis requests for MANUAL STEP, PAUSE, SYNC, MODIFY, OVERRIDE, LOOP, STOP, ALTER, MEASURE, READ, WRITE and DISPLAY are turned off.
- DC Fail and Distribution requests are turned off.

Attaching a test program to a station causes the test program previously attached to the station to be detached from the station.

The KEEP option prevents paging from the disc. The test programs loaded from the Integrator with KEEP must fit in memory. The programs then remain in memory until released by a command.

### 2.2.5 Load Files

The command

LOAD 'filename' (expansion-number) (KEEP) (CLI)

loads a system overlay, a user overlay (ALLINK), or a string file into memory from the specified device.

In the disc based system an overlay is automatically loaded from disc when the overlay command is entered and the overlay is not currently in memory. A user overlay may also be automatically loaded when an EXEC statement is executed and the overlay is not in memory.

Loading an overlay may force test program paging depending on the availability of memory space.

The expansion-number allows expansion of the buffer area for overlays which collect data (e.g., parameter distribution). It must be specified in words. The requirements of the buffer area are described in the overlay discription of the user's manual.

If the input device is the disc, the file is searched first in the current job and then in the system job.

### 2.2.6 Dump a File

The command

DUMP 'filename' ('filename') ('filename') (CLO)

transfers the file from memory to the output device specified. If the device is CLO, the same file name may not exist at the Integrator.

### 2.2.7 Clear

The command

CLEAR (STATn)

resets all altered conditions established by MANUAL, STEP, PAUSE, MODIFY, SYNC, LOOP, STOP, READ, WRITE, MEASURE, DISPLAY, OVERRIDE, AND ALTER, and zeros the globals GLOB1 to GLOB40, SWITCH, VALUE and DATALOG. Clear cancels DATALOG requests and the TITLE. Clear suspends

data accumulation by Parameter Distribution and DC FAIL ANALYSIS. The value of SN is reset to 1. The station PID/POD are reset to VKT1.

### 2.2.8 Open Files

The command

OPEN CLO 'filename' (ADD) (STATn)

opens a file to be used for FACTOR program I/O or for datalogging output. If a test program is performing I/O on the Integrator or datalogging to CLO, it must be opened first.

If the device is for output, then the following questions are asked:

LOT# = (12 characters maximum)
DEVICE = (8 characters maximum)
CATEGORY = (6 characters maximum)

The above information is used as the part of the file identification at the Integrator and it is written onto the disc or mag tape as the first record in the file.

The ADD option specifies that data is to be appended to an already existing file at the Integrator.

Datalog output to devices opened by the command is packed in binary format in variable length records.

STATn must be entered unless the SET STATn has been issued previously to specify a default station.

### 2.2.9 Close Files

The command

CLOSE CLO (STATn)

closes a file previously opened and used for binary I/O.

Any open binary file is closed on exit to DOPSY via the DOPSY command. Closing an Integrator file allows the work plan at the Integrator to start processing the data.

### 2.2.10 Restart

The command

RESTART CLO (STATn)

causes the file which is currently open at the Integrator for STATn to be purged of all data. After the current file has been purged from the host, the OPEN

routine is entered and the LOT, DEVICE and CATEGORY questions are requested via the PID as for OPEN.

### 2.2.11 Disconnect Linkage to the Integrator

The command

HANG

allows the user to reset the linkage to the Integrator. When the linkage is disconnected, files are automatically closed at the Integrator and the driver is initialized.

### 2.2.12 Note

The command

NOTE 'message' (output)

displays the message specified to the VKT screen at the Integrator. NOTE is used to annotate listings or datalog output. It may also be used to give the operator a message from a DIF file.

### 2.2.13 Datalog

The command

DATALOG (DCT) (MEAS) (LOG) (TRIP) (EOT) (FRQn)
(FCT (COUNT) (IFM) (n)) (output) (STATn)
DATALOG (FRQn) (output) (STATn)
DATALOG OFF (STATn)

where:

DCT — Log all parametric test failures.

MEASURE — Log the results of all MEASURE statements.

LOG — Log the results of all MEASURE statements in the test program specifying LOG.

FCT (n) — Log all functional test failures. If n is specified, the first failure plus n additional failures for each ENABLE TEST statement is logged. If neither COUNT nor IFM Is requested, log the functional failures by address only. (Only one failure per address may occur.) Test program IFAIL statements requesting the COUNT mode are ignored.

COUNT (n) — Log all functional test failures by test step (test counts). This is a SPM option. If n is specified the first failure plus n additional failures for each ENABLE TEST statement is logged by count.

COUNT mode allows logging of multiple failures at an address if they occur and display the test count of the failure. Test program IFAIL statements not requesting COUNT mode are ignored.

IFM (n) — Log all functional test failures by address or by count mode depending on the mode of the test program IFAIL statements. If COUNT is also entered, begin logging in COUNT mode until the first IFAIL statement, otherwise begin in address mode. If n is specified, the first failure plus n additional failures for each ENABLE TEST statement is logged.

TRIP — Log all DPS trip failures.

FRQn — Log data on every nth device. The first device is logged followed by every nth device. For example, if FRQ10 is entered, devices 1, 11, 21, 31, . . . etc. are logged. If the frequency is changed without changing other positions the next device is logged followed by every nth device. FRQ1 causes every device to be logged.

OUTPUT — Specifies the output device. The default output device is the POD if no device is specified in the DATALOG station command. If the output device is a binary device (tape, disc, Integrator) it must be opened by the OPEN or SET command.

OFF — Logging of the test results is turned off for the station specified. The frequency is set to 1.

lists the requested test results for the station to the selected output device or to the station or system output device if no output unit is specified.

Any or all of the datalog options may be entered in any order in the same command request. If any of the above options are specified, any option not selected is turned off. No logging occurs if the enable condition is not met for the device tested.

The frequency or output unit may be specified in the options command or may be changed independently if none of the log type options is requested. The datalog device has not effect on FACTOR I/O.

When FCT (or COUNT or IFM) is specified with a number, the number requests the multiple fail datalog mode. In this mode, n additional failures are logged. After the first functional failure is logged, MASTR restarts local memory setting the ignore failure value to the address or test count (depending on the mode) of the last failure. This causes all failures to that address or count to be ignored so that the next failure can be logged. Local memory is restarted at 0 for non-SPM test programs and at the local memory start address for SPM test programs.

Functional fail datalogging is inhibited in the enable latches mode (set by the test program) and no fails are logged except the accumulated result when the functional test sequence is complete.

CLEAR turns off the datalog option requests.

Datalog Output:

A station header is output once for each test start request if any test program or datalog output occurs. The station header displays the station identifer, the test program name, the data, time, and the serial number of the device test which is incremented for each completed test execution.

The station header line is followed by a title line. The title line is blank unless specified by the operater command TITLE.

Individual heading lines are printed above the columns of datalogged output for each type of output which occurs. When the type of data being logged changes (or after the title) an appropriate header is output. The same header is used for DCT, MEASURE, and LOG requests. DC measurements in printout format are always in engineering units and expected values are always output regardless of the pass/fail condition.

When the output device is the CLO, then the datalog output is in packed binary format.

### 2.2.14 Compile

The command

COMPILE (CLI 'filename') (data output) (list output) (options)

where:

| | |
|---|---|
| data output | (OBJ/LOBJ) (DOF 'filename'/MOF 'filename') |
| list output | (LIST) (TTP/VP1/VP2/LP/MTW1/MTW2/CLO 'filename') |
| options | (NOBJ/NADDR/NSEQ) (XREF/SYM) (OMV/TMV) |

The input group in the above command indicates that the compiler source input is from the keyboard (TTK), from cards via the card reader (CR), from magnetic tape via tape unit 1 (MTR1) or unit 2 (MTR2), from a disc file (DIF 'filename'), from a file on the Integrator system (CLI 'filename') or from a memory file (MIF 'filename').

Not more than one of these sources may be specified. The user may, however, elect not to specify an option, in which case the compiler expects its input from the current principal input device (PID).

The data output group defines the object code output destination to be either a disc file (DOF 'filename') or memory file (MOF 'filename'). The selection is preceded by OBJ or LOBJ. Since OBJ is the default condition, it need not be entered; however, the output file must always be specified if object code is produced.

The list output group defines the device on which the listing is to be output. The selections include the console (TTP), line printer (LP), magnetic tape unit 1 (MTW1) or unit 2 (MTW2) or Integrator system file (CLO 'filename'. A single magnetic tape unit cannot be used for input and output simultaneously. If no entry is made, the output, if any, goes to the current principal output device (POD). If LIST is selected, then only source statements are listed. If LOBJ is selected, then both the source statements and their resulting object codes are generated. The local memory address is listed for all SET F statements, these are the default cases. A cross reference listing can be produced at the end of the listing by entering XREF. A cross reference listing provides a listing of all labels, variables, arrays, subroutines, functions and subroutine formal parameters used in the FACTOR program.

The options group allows the selecting or disabling of various compiler outputs. NOBJ disables object code output. NADDR inhibits listing of local memory addresses. NSEQ disables the check of sequence numbers on input source statements.

The SYM option provides a symbol table listing at the end of compilation.

The one millivolt or two millivolt system option can be selected at compile time by entering OMV (one millivolt) or TMV (two millivolt).

Examples:

COMPILE CLI 'BIGD' MOF 'DARL' XREF

Compilation source is from Integrator file, object code produced is stored in memory and a cross reference listing is produced and sent to POD.

COMPILE DIF 'SCO' DOF 'name2' LIST CLO '!name3' XREF

Compilation source is from a disc file, object code produced is stored in disc and a listing with corss reference is sent to the line printer at the Integrator.

### 2.2.15 Operator Messages from Com Link

Operator messages may be received at any time while in MASTR. The message displayed on the VKT preceded by the letters 'CL' to signify Com Link.

## 2.3 TOPSY COM LINK OPERATION

### 2.3.1 NOTE CLO

NOTE CLO 'operator message'
NOTE CLO HANG

These TOPSY instructions perform the same functions as described in DOPSY, section 2.0.

### 2.3.2 OPEN CLO (ADD) STATn

This command will initiate communication between the Host and the Sentry for a particular station. Once the CLO has been opened, datalogging to the host will be allowed.

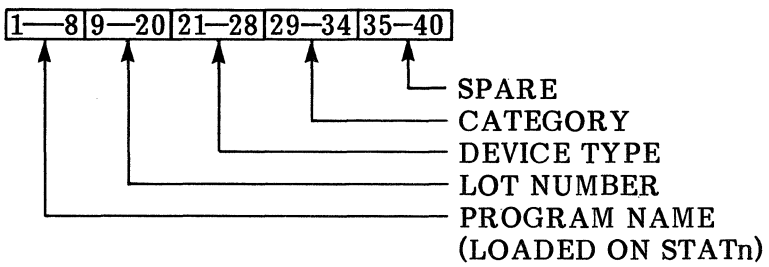
When OPEN CLO has been entered the command processor will respond with the following questions which will always be printed on the VKT.

| | | |
|---|---|---|
| LOT | = | (12 characters maximum) |
| DEVICE | = | (8 characters maximum) |
| CATEGORY | = | (6 characters maximum) |

The answer will be read from the PID. If the PID is not the VKT, the response will be printed on the screen. Therefore, the command may be entered from a DIF file or a card reader and the operator will be able to see the full command entries. Only the first 12, 8 or 6 characters will be used, including blanks. Any TASCII character is allowed. If the OPEN command is entered via Com Link command the questions are asked of the Integrator.

If the add option (ADD) is included the existing file will be re-opened and data added to it. The lot, device, and category must be specified when ADD is entered.

The format of the open message, and the format of the directory entry at the Host is as follows:




A test plan must be loaded for the station requested and no CLO file may be open for this station. A duplicate CLO file must not exist at the Host if ADD is not entered.

When the file is opened the message

'CL FILE OPENED'

is output. Opening a file also enables messages to be received from the Com Link operator. LOAD or CLEAR do not close a Com Link file. After loading a new program file to a station, the user must not DATALOG CLO or else data for two test plans will go into the same Host file. LOAD (without the SAVE option), CLOSE, and DOPSY notify the user if a Com Link file is open.

### 2.3.3 RESTART CLO STATn

This command will cause the file which is currently opened at the Host for STATn to be purged of all data. After the current file has been purged from the host, the message

'CL FILE PURGED'

is displayed, and the LOT, DEVICE and CATEGORY questions are requested via the PID as above.

This facility allows for a lot to be re-tested if it is determined part way into the lot that the data was meaningless. An example would be if the wrong load board was mounted and all devices were failing.

Restrictions:

A test plan must be loaded for the station requested. A file must be OPEN at the Host. This command may be entered from Analysis. A closed CLO file may not be restarted. However, a closed file may be opened with ADD option, and then restarted which would result in the deletion of all information in the file.

### 2.3.4 CLOSE CLO STATn

This command signals the Host that the end of the lot is reached and no more data is to be added at this time. The close command will allow the work plan at the host to start processing the data.

Once a file is closed, the delete or purge function is under the control of the Host software when processing on the file is performed by the work plan.

This command may be entered from Analysis.

Restrictions:

A CLO file must be open for the station requested. Attempting to datalog to the CLo will result in a terminal error 90 after a file has been closed.

### 2.3.5 SET CLI

This instruction performs the same function is DOPSY and TOPSY. Refer to section 2.3.

### 2.3.6 Datalogging Test Output

/. DATALOG (MEAS) (FCT (COUNT) (IFM) (n) ) (LOG) (DCT) (EOT) (TRIP) (FRQn) (TTP/LP/CLO/MTW) STATn
/. DATALOG (TTP/LP/CLO/MTW) (FRQn) STATn
/. DATALOG OFF STATn

Datalogging to the Integrator is initiated by the mnemonic CLO.

If no datalogging options are specified the output unit may be changed without retyping the entire request. This allows datalogging the first test to the line printer or VKT to examine the results and then changing the output unit to CLO.

Specifying EOT will cause a message to be sent to the Host at the completion of each device tested, for both passed and failed tests. This message will contain the 15 bits of the EIR register which provide the pass/fail bits, EOT bits, and the user programmed data.

Specifying FRQn will allow the collection of data on every nth device. Frequency may be changed independently from the options or the output device.

FRQ does not affect the EOT request which is output for all test plan executions if EOT has been selected. N may be any number from 1 to 99.

The datalogger will log the first device followed by every nth device. For example if the command is

/.DATALOG DCT FRQ10 STAT1A

DC failures will be logged for device 1, 11, 21, 31, 41, ...etc.

When any options are requested (MEAS, FCT, COUNT, LOG, DCT, EOT, TRIP) any option not requested is turned off, the frequency becomes 1 (every device is logged) unless specified, and the output device becomes the system POD unless specified.

When OFF is specified all options are reset, the frequency becomes 1 and the output device becomes the system POD unless specified.

Restrictions:

The OPEN command alone does not cause datalogging to the host. The datalog command must include "CLO" to datalog to the host.

If CLO has been specified and a host file has not been opened, terminal error 90 will result at run time. Datalog commands are accepted from Analysis.

### 2.3.7 Operator Message from Com Link

Operator messages may be received at any time while in TOPSY, however, they will only be displayed at EOT on a station. The message will be displayed on the Sentry VKT preceded by the letters 'CL' to signify Com Link.

If a second message is received before the first message is displayed, only the last message is retained.

If the command SET CLI has previously been issued, the message will be operated upon as a standard TOPSY or Manual Analysis command.

## SECTION 3

## DATALOG TO BINARY RECORD FORMAT

### 3.1 DATA RECORD DESCRIPTION

A data record sent to the Integrator is of variable length with a 1-word record identifier at the beginning. The record identifier contains the following information:

B0-11 = Record length (in FST words), 4096 max *

B12-17 = Record Type. Each datalog carries a 2-digit number which specifies the type of record (See Table 3-1). B15-17 specify a record class, i.e., DCMEAS. B12-14 specify the particular record type within the class.

B18-21 = Spare, not used.

B22 = 1 user defined record
= 0 System defined record

B23 = 1 format record
= 0 DATA record

Thus, a data record from the datalogger with 2 items would appear as:

| | | | | | |
|---|---|---|---|---|---|
| WORD 0 | 0 | 0 | Record ID | 3 | |
| WORD 1 | Item 1 | | | | |
| WORD 2 | Item 2 | | | | |

Item format and meaning are described in the following section.

*The record identifier is limited to 40 words for Com Link transfers.

**TABLE 3-1 RECORD ID FOR DATALOG TO INTEGRATOR**

| ID Number | Record is From |
|---|---|
| 00 | DPS Current Trip |
| 01 | DPS Voltage Trip |
| 10 | DC FAIL - Current Mode |
| 11 | DC FAIL - Voltage Mode |
| 12 | DC PASS - Current Mode |
| 13 | DC PASS - Voltage Mode |
| 14 | MEASURE VARIABLE |
| 20 | FUNCTIONAL FAILURE |
| 21 | PPO MEMORY FAIL |
| 22 | PPO MEMORY FAIL, DATA EXTENSION |
| 23 | FUNCTIONAL FAILURE, MESSAGES |
| 30 | EOT RECORD |
| 40 | SHMOO PLOT |
| 50 | DATA IO |

Measurements are sent as FST 24-bit floating point numbers. Records with ASCII information are sent with four 6-bit TRASCII characters per word. Integer and central information varies depending on the record. See each record description for specific format.

Note that although Test type and Module number are not currently used, space has been reserved for their future use.

See section 3.3 for difference of MASTR and TOPSY binary records.

## 3.2 DATALOG RECORD FORMATS

### 3.2.1 DPS Trip Fail Record

A DPS trip fail record is generated when DATALOG TRIP is requested and a power supply trip occurs.

Each power supply trip is sent as a separate record. The record format is:

| 0 | 0 | | Record ID | Record Length |
|---|---|---|---|---|
| INSTRUCTION NUMBER | | | | |
| (TEST TYPE) | | | | |
| (MODULE NUMBER) | | | | |
| TRIP SUPPLY NUMBER | | | | |
| GT/LT | | | | |
| TRIP VALUE | | | | |

where:

RECORD ID = 00 Current TRIP
= 01 Voltage TRIP

RECORD LENGTH = 7 FST Words

INSTRUCTION NUMBER

TEST TYPE, MODULE NUMBER } 16-Bit unassigned integer not currently used

GT/LT = TRASCII 'LT' or 'GT'

TRIP SUPPLY NUMBER = 16-Bit unassigned integer

TRIP VALUE = 24-Bit floating point number

### 3.2.2 DC Fail Record

A DC Fail record is generated when DATALOG DCT/MEAS/LOG is requested and the measurement fails. The record format is:

| 0 | 0 | | Record ID | Record Length |
|---|---|---|---|---|
| INSTRUCTION NUMBER | | | | |
| TEST TYPE | | | | |
| MODULE NUMBER | | | | |
| PIN NUMBER | | | | |
| FLAG WORD | | | | |
| MEASURED VALUE | | | | |
| DCT0 LIMIT | | | | |
| DCT1 LIMIT | | | | |

where:

RECORD ID = 10 → Current mode
= 11 → Voltage mode

RECORD LENGTH = 9 FST Words

INSTRUCTION NUMBER

TEST TYPE } 16-Bit unsigned integer
MODULE NUMBER } Not currently used

PIN NUMBER

FLAG WORD - ASCII formatting information

MEASURED VALUE
DCT0
DCT1
} = 24-Bit floating point number

### 3.2.3 DC Pass Record

A DC Pass record is generated when the DATALOG MEASURE/LOG is requested and the DC Measurement passes. The record format is:

| 0 | 0 | | Record ID | Record Length |
|---|---|---|---|---|
| INSTRUCTION NUMBER | | | | |
| TEST TYPE | | | | |
| MODULE NUMBER | | | | |
| PIN NUMBER | | | | |
| FLAG WORD | | | | |
| MEASURED VALUE | | | | |
| DCT LIM | | | | |
| DCT LIM | | | | |

where:

RECORD ID = 12 → Current mode
= 13 → Voltage mode
= 14 → Measure variable

The remaining record fields are the same as a DC FAIL RECORD.

### 3.2.4 Functional Failure Record

A Functional Failure record is generated when DATALOG FCT is requested and a functional failure occurs. The record format is:

| 0 | 0 | | Record ID | Record Length |
|---|---|---|---|---|
| INSTRUCTION NUMBER | | | | |
| TEST TYPE | | | | |
| MODULE NUMBER | | | | |
| LOCAL MEMORY LOCATION | | | | |
| TEST SEQUENCE COUNT | | | | |
| F-REG RANK 1 | | | | |
| C-REG RANK 1 | | | | |
| F-REG RANK 2 | | | | |
| C-REG RANK 2 | | | | |
| ⋮ | | | | |
| F-REG RANK n | | | | |
| C-REG RANK n | | | | |

where:

RECORD ID = 20

RECORD LENGTH = 14-22 FST Words (depends on number of ranks used)

TEST TYPE
MODULE NUMBER — Currently not used

TEST SEQUENCE COUNT = 24-Bit integer number

All other words, except for the ID word, are 16-bit integers. The C and F Register data is contained in Bits 0 through 14.

### 3.2.5 Functional Failure Record - Fail Message

A Functional Failure record (fail message) is generated for DC Time Out, Loop Count or Clock Burst count failure. The record format is:

| 0 | 0 | | Record ID | Record Length |
|---|---|---|---|---|
| INSTRUCTION NUMBER | | | | |
| TEST TYPE | | | | |
| MODULE NUMBER | | | | |
| LOCAL MEMORY LOCATION | | | | |
| TEST SEQUENCE COUNT | | | | |
| TRASCII MESSAGE | | | | |
| TRASCII MESSAGE | | | | |
| MTRASCII MESSAGE | | | | |

where:

RECORD ID = 23

RECORD LENGTH = 9 FST Words

TEST TYPE
MODULE NUMBER } Currently not used - 16-Bit Integer

INSTRUCTION NUMBER
LOCAL MEMORY LOCATION } 16-Bit integer

TEST SEQUENCE COUNT = 24=Bit integer number

TRASCII messages are =
- T/O
- FC
- LOOP
- T/O, FC
- T/O, LOOP

Each message is 3-FST words, with four 6-bit TRASCII characters per word.

### 3.2.6 PPO Memory Functional Failure

A PPO memory functional failure is generated when a Functional Fail in a DUT occurs. The record format is:

| 0 | 0 | | Record ID | Record Length |
|---|---|---|---|---|
| INSTRUCTION NUMBER | | | | |
| TEST'TYPE | | | | |
| MODULE NUMBER | | | | |
| TEST SEQUENCE COUNT | | | | |
| DATA READOUT #1 REG | | | | |
| DATA READOUT #2 REG | | | | |
| C-REGISTER RANK 1 | | | | |
| C-REGISTER RANK 2 | | | | |
| C-REGISTER RANK 3 | | | | |
| C-REGISTER RANK 4 | | | | |

where:

RECORD ID = 21 (Non-data extension
= 22 (Data extension mode)

RECORD LENGTH = 11-FST Words

Except for ID and Test Sequence count, all words are 16-Bit integer numbers.

TEST TYPE Currently not used.
MODULE NUMBER 16-Bit integer number.

TEST SEQUENCE COUNT = 24-Bit integer number

C by Rank 3 and 4 contains information for Data Extension mode only.

### 3.2.7 EOT Record

An end-of-test record is generated when an end-of-test point is reached and DATALOG EOT is requested. The record format is:

| WORD | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | | Record ID | Record Length |
| 1 | EIR DATA | | | | |

where:

REOCRD LENGTH = 2

RECORD ID = 30

EIR DATA = B0-B9 user defined value

B10 = 1 DC TEST FAIL

B11 = 1 DC TEST PASS

B13 = 1 FUNCTIONAL TEST PASS

B14 = 1 END OF TEST

## 3.3 WRITING ARRAYS FROM A FACTOR PROGRAM

A FACTOR program may cause data to be written by one of the following types of statements:

1. WRITE "ident" ARRAY;

2. WRITE "ident" variable; variable,...;

3. WRITE "ident" "TEXT"

4. WRITE "ident" 'TEXT', V1, V2, V3, 'TEXT', V4, V5, V6;

5. WRITE "ident" /n/'TEXT', variable;

Cases 1 and 2 above result in the following format:

| word 0 | item 1 | item 2 |
|---|---|---|

word 0: length = no items in the array +1 (4096 max.) or no. of variables +1 (32 max.)
ident = TRASCII in double quotes
code = 010000 (binary)
items 1-n: FST floating point numbers

Cases 3 - 5 involve a combination of binary and TRASCII data and therefore may cause more than 1 record to be generated. The records output are as follows:

| word 0 | text 1 |
|---|---|
| word 0 | V1 V2 V3 |
| word 0 | text 2 |
| word 0 | V4 V5 V6 |

word 0: length = no words of test or variables + 1 (33 max.)
ident = TRASCII character in double quotes
codes = 01000 (binary)

## 3.4 MASTR/TOPSY DATALOG TO BINARY RECORD DIFFERENCE

The following items represent the difference between binary records produced by MASTR and those generated by TOPSY.

1. The Measure Variable read in MASTR (ID = 14) does not exist in TOPSY. The latter passes the Measure Variable results as a DCPASS or DCFAIL record.

2. TOPSY DCPASS or DCFAIL records do not have a DCT1 limit. Only the limit that failed is sent as DCT10 limit. The flagword has B5 to signify the failure type; i.e.,

   B5 = 1 = D GT
   = 0 = D LT

3. Functional failure records:

   TOPSY always passes 4-ranks of C and F-Reg data, even if there are no failures in some of the numbers. MASTR passes from one tape to eight ranks of C and F-register data depending on how many ranks are defined in the FACTOR program (default = 4 ranks, 60 pins). Generally, it is expected that the number of ranks are constant throughout the FACTOR program.

4. When using the FACTOR statement (in TOPSY)

   WRITE (CLO) "X" ARRAY;

   The array size, n, must be $2 \geq n \geq 40$. If n=1 , the array is treated as a single variable and ignored. The array identifies "X" as a one-character identification for the array and is optional.

# SECTION 4

# MESSAGE AND LINE PROTOCOL

## 4.1 GENERAL

A message is a sequence of characters that are transmitted in one continuous transmission. The characters transmitted may contain 7 data bits representing an ASCII character, or 8-bits of binary data. Each message is made up of two fields:

- Message header
- Message text

The message header contains information as to the source, destination, type and content of the data in the message text. The message text contains the actual data being transferred.

The message header contains 8 ASCII characters as follows:

| BYTE | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SA | SSA | DA | DSA | MODE | TYPE | SPARE | SPARE |

where:

SA - Source Address (ASCII 0-8)

The source address of a message to be transmitted.

SA = 0 The message to be sent was generated locally.

SA = 1-8 The message to be sent was generated by the system 1-8. The local system acts as a transfer agent.

SSA - Source Sub-Address (ASCII 0-8)

The source sub-channel address.

SSA = 8 A system type message such as part of an upload or download.

SSA = 1-4 A message generated from test station 1-4.

DA - Destination Address (ASCII 0-8)

The destination of the message to be transmitted.

DA = 0 — A message transmitted to the Integrator (upload) or Sentry (download).

DA = 1-8 — A message transmitted to system 1-8 through the Integrator (upload) or through Sentry (download).

DSA - Destination Sub-Address (ASCII 0-8)

This character is used to address 1 of 9 files that may be active concurrently at the Integrator.

DSA = 8 — A system message such as part of an upload or download sequence.

DSA = 1-4 — Tester data from station 1-4 that is to be processed by work plan at the Integrator.

MODE - (Binary 0 or 1-127)

This character determines the mode of transmission for the message text that follows the message header. If MODE = 0, then the message is ASCII and is terminated by the ETX character. If the Mode $\neq$ 0, then it indicates the number of characters in the message including the header.

TYPE - (ASCII 1-6) Message Type

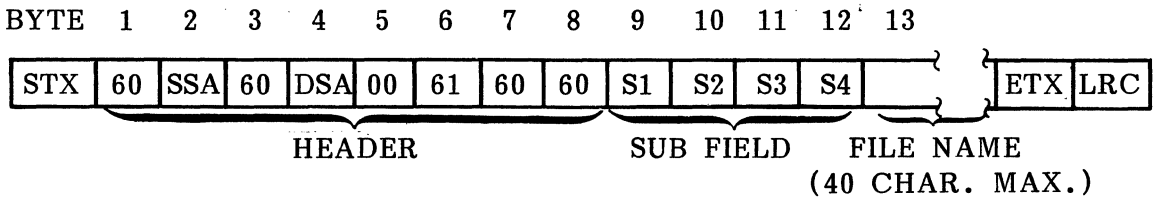
1 = File Request
2 = File Transmit
3 = File Data
4 = File End
5 = Status
6 = Operator Message

SPARE - 7 and 8 are not used at this time.

## 4.2 MESSAGE TYPE FORMATS

### 4.2.1 Message Type 1 - File Request

This message type is transmitted when the download of a file is requested. The format is as follows:




where:

S1, S2 = 60B (0, 0 ASCII)

Indicates a request for a new file or the request to transmit a new file.

S1, S2 = 60B, 61B (0, 1 ASCII)

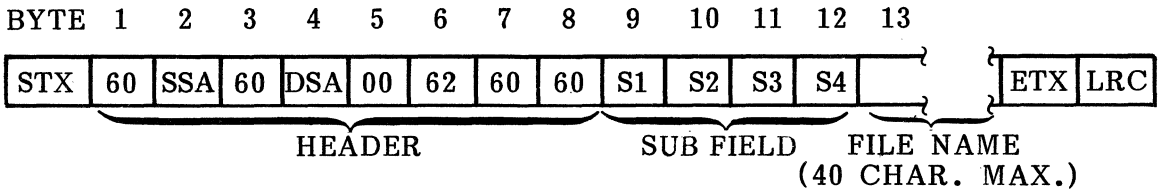
Indicates a request for a file to be appended.

S3, S4

Not used for message type 1 or 2.

### 4.2.2 Message Type 2 - File Transmit

This message type signals the transmission of a file. If the download is being requested, then this message is always sent in response to a message Type 1. The format for the message type is exactly the same as for the message Type 1:



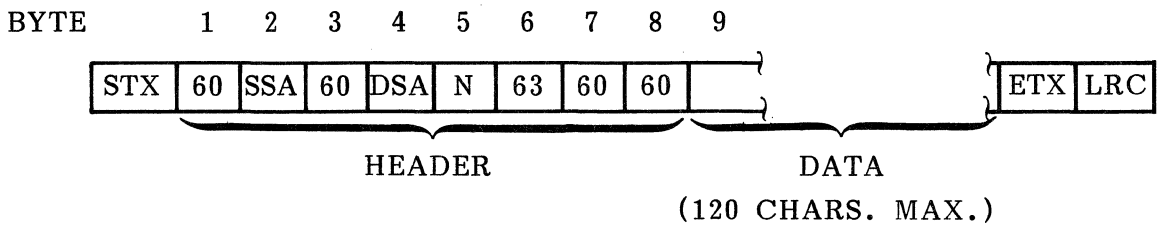

where:

S1, S2,S3, S4

Same definition as message Type 1.

### 4.2.3 Message Type 3 - Data

A series of message Type 3 transmissions are used to transfer a complete file from Integrator to or from the Sentry. The messages are sent one after the other or in response to a Status message Type 5 Data message Type 3 is always transmitted in the binary mode with byte number 5 being a binary number from 1-127 indicating the number of characters in the message (data +8 characters of header). The format is as follows:



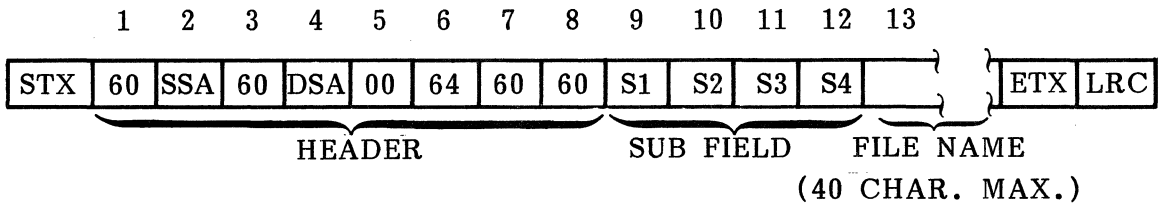

where:

DATA Up to 120 binary characters (8 bits/character).

N Character 5 of the header contains a binary number from 1-177B indicating the number of characters in the message (header + data).

### 4.2.4 Message Type 4 - File End

A file end message signals that the last data message has been transmitted and that the file transfer has been completed. The format is as follows:




where:

S3, S4 Not used for message Type 4 and is ignored.

S1, S2 = 60B, 60B (0,0 ASCII)

The file is closed and made available for parameter processing.

S1, S2 = 60B, 61B (0, 1 ASCII)

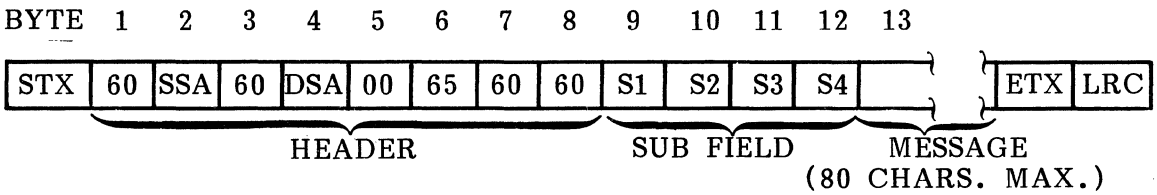
The file is closed and purged. The file name must agree with the file opened with the message Type 2.

S1, S2 = 60B, 62B (0,2 ASCII)

The file is closed and remains a raw data file, i.e., no parameter processing.

### 4.2.5 Message Type 5 - Status Message

A status message is issued in response to other messages to indicate the successful completion on an error condition that has occurred. The format is as follows:




where:

S1, S2 = 60B, 60B (0, 0 ASCII)

Indicates successful completion of an operation with no error. This message is always sent in response to a message Type 2 or Type 4. In response to a (download), S3 and S4 message type 2 (download) are used to signal fast download, strobed download, or random access download.

S1, S2 > 60B, 60B (0, 0 ASCII)

If S1, S2 are non-zero, then an error condition exists, S1, S2 indicate the error number and this along with the message (optional) is displayed on the VKT.

S1, S2 = 60B, 60B (0, 0 ASCII)

S3, S4 = 40B, 40B (blank, blank ASCII)

This subfield when used during a download indicates fast download, i.e., the data messages are to be transmitted as fast as possible without waiting for any status messages.

S1, S2 = 60B, 60B (0, 0 ASCII)

S3, S4 = 40B, 41B (Blank, ! ASCII)

This subfield indicates strobed download. Each time a message Type 5 is received, the next message Type 3 is transmitted.

S1, S2 = 60B 60B (0, 0 ASCII)

S3, S4 > 40B, 41B (Blank, ! ASCII)

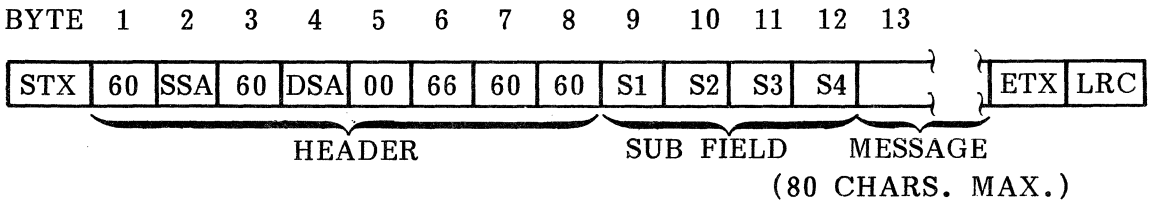
This subfield indicates a random access download. The S3 and S4 subfield are decoded as follows:

u = (S3-40B) *140B + (S4-40B) - 1

This value of u is then used to position the file to record #u which is transmitted. The Integrator then waits for another message Type 5 before proceeding.

### 4.2.6 Message Type 6 - Operator Message

An operator message may be transmitted at any time from either the Integrator or Sentry CPU. The purpose of this message is to allow the operator of the two systems to communicate. The receiving CPU displays the contents of the message on the system output device. The format is as follows:




where:

S1, S2 = 60B, 62B (0, 2 ASCII)

Indicates that the message was generated manually by an operator. It is displayed in bold letters at the Integrator.

S1, S2 = 60B, 63B (0, 3 ASCII)

Indicates that the message was generated by a software process.

### 4.2.7 Message Sequencing

To transfer a file of data via the com link requires that a sequence of messages be adhered to. The following examples indicate this sequence for the transfer conditions indicated.

Example 1 - Upload from Sentry to an Integrator - initiated by Sentry.

| **Sentry** | | **Integrator** |
|---|---|---|
| MSGT 2 (S1, S2 - 0, 0) | → | |
| | ← | MSGT 5 (S1-S3 = 0, 0, 40B, 40B) |
| MSGT 3 | → | |
| . | → | |
| . | → | |
| . | → | |
| MSGT 3 | → | |
| MSGT 4 (S1, S2 = 02) | ← | MSGT 5 (S1, S2 = 0, 0) |

Example 2 - Download from Integrator to Sentry - initiated by Sentry.

| **Sentry** | | **Integrator** |
|---|---|---|
| MSGT1 (S1, S2 = 0, 0) | → | |
| | ← | MSGT 2 (S1, S2 = 0, 0) |
| MSGT 5 (S1-S3 = 0, 0, 40B, 40B) | → | |
| | ← | MSGT 3 |
| | ← | |
| | ← | MSGT 3 |
| | → | MSGT 4 (S1, S2 = 0, 0) |
| MSGT 5 (S1, S2 = 0, 0) | | |

Example 3 - Datalog from Sentry to Integrator - append to existing file

| **Sentry** | | **Integrator** |
|---|---|---|
| MSGT 2 (S1, S2 = 0, 1) | → | |
| | ← | MSGT 5 (S1, S3 = 0, 0, 40B, 40B) |
| MSGT 3 | → | |
| | → | |
| MSGT 3 | → | |
| MSGT 4 (S1, S2 = 0, 0) | → | |
| | ← | MSGT 5 (S1, S2 = 0, 0) |

Example 4 - Download a Command file for processing at Sentry

| **Sentry** | | **Integrator** |
|---|---|---|
| MSGT 1 (S1, S2 = 0, 0) | → | |
| | ← | MSGT 2 (S1, S2 = 0, 0) |
| MSGT 5 (S1-S3 = 0, 0, 40B, 41B) | → | |
| | ← | MSGT 3 |
| MSGT 5 (S1-S3 = 0, 0, 40B, 41B) | → | |
| | ← | MSGT 3 |
| MSGT 5 (S1-S3 = 0, 0, 40B, 41B) | → | |
| | ← | MSGT 4 (S1, S2 = 0, 0) |
| MSGT 5 (S1, S2 = 0, 0) | → | |

## 4.3 LINE PROTOCOL

Line protocol is a set of rules involving 9 ASCII characters that are used for the purpose of achieving orderly and controlled transmission between two asynchronous CPUs.

### 4.3.1 Definition of Line Protocol Characters

| Characters | Definitions |
|---|---|
| STX (02B) | start of message |
| ETX (03B) | end of message (is always followed by an LRC character). |
| ACK (06B) | acknowledgement by the receiver of a message that it was received correctly (i.e., correct number of characters and correct LRC). |
| NAK (25B) | a request to the sender that a message that was incorrectly received be retransmitted. (A message is normally retransmitted 10 times before reporting a transmission error). |
| BID (22B) | a request for line ownership |
| XON (21B) | a granting of line ownership to the computer to which XON is sent. This is the normal response to a BID. |
| XOFF (23B) | the freeing of line ownership by the sender. This character is normally sent when the sender has no more data to transmit. |
| CAN (30B) | indication to the receiver that transmitter has selected an error condition or cannot honor a BID for line ownership. |
| SYN (26B) | sync character, i.e., NOP |

### 4.3.2 Line Protocol Rules

4.3.2.1 SENDING MESSAGES:

1. Send no message unless you own the line.

2. If you send a message, don't send another one until the other computer replies.

4.3.2.2 RECEIVING REPLIES TO A MESSAGE YOU SENT:

1. ACK means that the other computer does not want you to retransmit the message just sent. (The message was received correctly.)

2. NAK means that the other computer wants you to retransmit your last message. (It was not correctly received.)

3. CAN means an error exists, BID for line and try again.

4. Raise an error condition if you receive any other reply.

4.3.2.3 REPLYING TO MESSAGES SENT BY THE OTHER COMPUTER:

1. If you want the other computer to retransmit the message, send NAK. (Message incorrectly received.)

2. If you do not want the other computer to retransmit the message, then:

   a. Send ACK if the message was correctly received,
   b. Raise an error condition, otherwise

3. Whenever you receive STX, receive the entire message, and then reply before you do anything else.

4.3.2.4 DETERMINING LINE OWNERSHIP:

1. If you wish to own the line and are not required to send any other character (for example ACK, NAK, etc.), then send BID.

2. If you receive XON and it does not raise an error condition (for example, an illegal reply to a message you just sent), then you own the line.

3. If you receive XOF (whether or not it raises an error condition), then the other computer does not own the line. If the other computer owned the line, then the line becomes free. If you own the line, or it is free, then line status is unchanged.

4. If you own the line, and are not obligated to follow other protocol rules (such as a reply to a message), you may:

   a. Free the line by sending XOF (preferred if you do not wish to send a message).
   b. Force line ownership to the other computer by sending XON (preferred only if you expect incoming messages).
   c. Send nothing, and thus retain line ownership (not preferred).
   d. Send a message.

5. If you receive BID, then you must do one of the following:

   a. Grant line ownership to the other computer by sending XON.
   b. Reject request for line ownership by sending CAN.

4.3.2.5 ERROR CONDITIONS:

1. If you ever send or receive CAN, then an error condition is raised such that:

   a. The line enters its free state
   b. If you just sent a message, assume that it was not received.
   c. If you were expecting a particular reply (such as ACK, NAK, XON, etc.) you should no longer do so. All pending status' are cancelled. Both computers (perhaps for only an instant) enter their Quiet-state.

4.3.2.6 UNEXPECTED CHARACTERS:

If you receive a character not described in the other rules, you may:

1. Ignore it,

2. Raise an error condition

4.3.2.7 SYN

If you receive SYN, not as part of a message, do not raise an error condition and do not change states.

# SECTION 5

# DATA TERMINAL

## 5.1 GENERAL

This section describes the Data Set Interface (Data Terminal) which is used in providing a serial data link between the FST-2 and the Integrator. The interface of the data link to the FST-2 is described along with interface configuration of the data link to the Integrator (refer to Figure 5-1). The operating discipline of the data link is described with respect to the FST-2 and its interconnection via modem (2025 or equivalent) or null modem (refer to Figure 5-2).

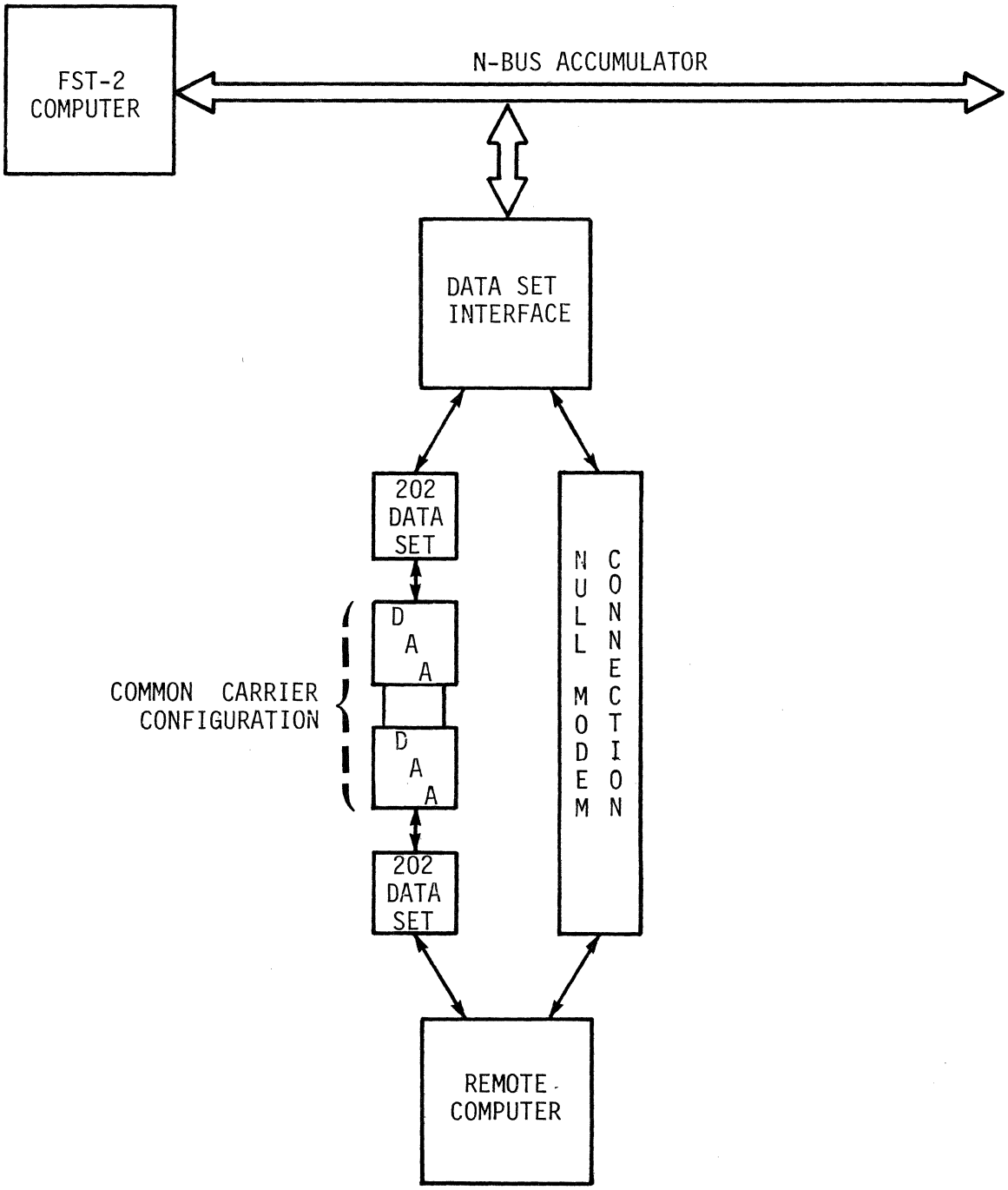



**Figure 5-1 Data Link Configuration**

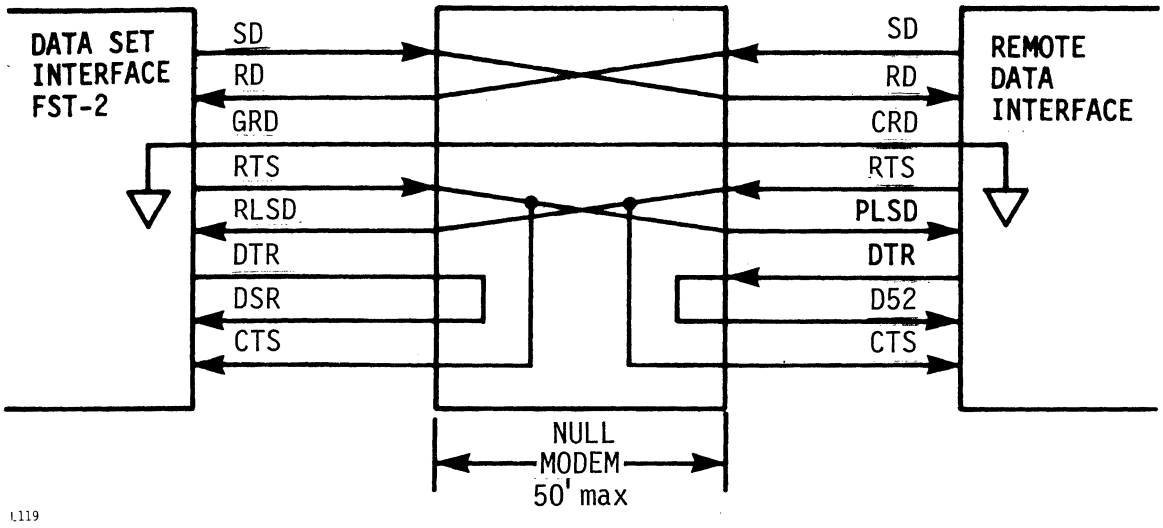



**Figure 5-2 Null Modem Configuration Detail**

## 5.2 DATA TRANSMISSION RATES

### 5.2.1 Rate Selection

Data transmission rates are determined by programming a manual 4 position data switch. The rates available are as follows: 50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, and 19,200 baud. The selection is made from the chart shown in Table 5-1 and manually programmed on dataset interface board #2. Refer to section 8.3 for the exact location of the switch on board #2. The transmission rate governs the speed at which 'send' data is serially placed on the transmission line and rate at which 'receive' data is assembled into the receiver register.

### 5.2.2 FST-2 Host Rate Selection

The data set interface provides asynchronous, half-duplex data link capabilities to the FST-2. The data set interface is selected to send and receive data at up to 9600 baud in a null modem configuration. Operation over common carrier facilities is limited to 1200 baud through switched network service or 1800 baud through dedicated private line service.

### 5.2.3 Dataset Interface Formats

5.2.3.1 FST ACCUMULATOR AND BUS I/O

Data transfers to and from the dataset interface are accomplished during data cycles (@T1) of the SPU instruction. Data transactions of character data utilize the least significant 8-bits of the N-Bus field. Input data received by the data set interface is right justified with the least significant data bit being the right most bit. Transmit data follows the same format.

5.2.3.2 CHARACTER FORMAT TRANSMISSION:

The data character beginning with the least significant bit is serially transmitted immediately following a single start bit. The data character is appended with one stop bit for all data rates except 110 baud where an additional stop bit is appended.

5.2.3.3 CHARACTER FORMAT - FST ACCUMULATOR

A bit in the dataset interface control register is used to select either the ASCII or binary mode of data transmission and reception.

5.2.3.4 ASCII TRANSMIT

When the interface is in the ASCII mode, bits 0-6 of the A-register are output by the SPU write instruction. A-register bits 7-23 are "don't care". The dataset appends the start, stop and even parity bits.

**TABLE 5-1 BOARD NUMBER 2 SWITCH SELECTION CHART**

| Manual Programmed Baud Rate | | | | | | |
|---|---|---|---|---|---|---|
| **Baud Rate** | **S1-1** | **S1-2** | **S1-3** | **S1-4** | **S1-5*** | **S1-6*** |
| 50 | ON | OFF | ON | ON | OFF | OFF |
| 75 | OFF | OFF | ON | ON | OFF | OFF |
| 110 | OFF | OFF | OFF | OFF | OFF | OFF |
| 134.5 | ON | ON | OFF | ON | OFF | OFF |
| 150 | ON | OFF | OFF | OFF | OFF | OFF |
| 200 | OFF | ON | OFF | ON | OFF | OFF |
| 300 | OFF | ON | OFF | OFF | OFF | OFF |
| 600 | ON | OFF | OFF | ON | OFF | OFF |
| 1200 | OFF | OFF | ON | OFF | OFF | OFF |
| 1800 | ON | OFF | ON | OFF | OFF | OFF |
| 2400 | ON | ON | OFF | OFF | OFF | OFF |
| 4800 | OFF | ON | ON | OFF | OFF | OFF |
| 9600 | ON | ON | ON | OFF | OFF | OFF |
| 19,200 | ON | ON | ON | ON | OFF | OFF |

* S1-5 and S1-6 are only used for self-test diagnostics. During normal operation however, they must be in the OFF position to insure proper functioning.

# SECTION 6

# ERROR MESSAGES

## 6.1 COM LINK DRIVER ERROR MESSAGES

| Error Number | Description |
|---|---|
| 0 | No error. |

The following error messages are issued when a violation of hardware line protocol occurs (refer to hardware logic description for more detailed information regarding hardware functioning):

| | |
|---|---|
| 1 | Data Set Change (DSC) interrupt occurred with no change of hardware state. |
| 2 | Clear To Send True (CTST) interrupt occurred out of sequence. |
| 3 | Clear To Send False (CTSF) interrupt occurred out of sequence. |
| 4 | Data Set Ready True (DSRT) interrupt occurred out of sequence. |
| 5 | Data Set Ready False (DSRF) interrupt occurred out of sequence. |
| 6 | Receive Line Signal Detect True (RLSDT) interrupt occurred out of sequence. |
| 7 | Receive Line Signal Detect False (RLSDF) interrupt occurred out of sequence. |
| 8 | Not applicable. |
| 9 | Not applicable. |
| 10 | Receiver Done (RDONE) interrupt occurred out of sequence. |
| 11 | Transmitter Done (TDONE) interrupt occurred out of sequence. |

The following error messages are issued when a violation of line protocol characters occurs (refer to section 4.3.1 for a complete definition of the line protocol characters):

| Error Number | Description |
|---|---|
| 12 | XON not received after a bid for line ownership. |
| 13 | ACK or NAK not received after a message. |
| 14 | Message transmitted in error 10 times and NAK received 10 times. |
| 15 | Start of message (9STX) received without line ownership being established. |
| 16 | Message received in error and NAK transmitted 10 times. |
| 17 | No end of message (ETX) character in ASCII message. |
| 18 | Unrecognizable protocol character. |

## 6.2 INTEGRATOR STATUS MESSAGES

The following are error messages produced by the Tester Interface Package of the Integrator. The messages are divided into Upload and Download errors. Each message has an additional message code which indicates the nature of the error message. The additional message codes are as follows:

- * Message printed on the Operator Interface Package VKT.
- # The negative File Manager error code appearing between the brackets is meaningful. Refer to the File Manager messages listed in section 6.3.
- ! The message subtype was sent to the external CPU.
- ? A Tester Interface Package fatal error occurred, followed by possible system ABEND.

### 6.2.1 Upload Error Messages (UPLDR)

| Error Code | Description | Message Code |
|---|---|---|
| 00 | Operation performed without error. | ! |
| 01 | Message recieved with illegal internal CPU number. | * |
| 02 | Bad parameter in a message | * |
| 03 | Destination CPU was not linked or ready. | *, ! |
| 04 | Attempt to open 2 upload files on one particular subchannel. | *, ! |
| 05 | Unable to open user file because of directory error. | *, #, ! |
| 06 | Attempt to open (not append to) a file which already exists. | *, ! |
| 07 | Error while opening (appending) a user file. | *, #, ! |
| 08 | Attempt to add data to a file which is not open. | *, ! |
| 09 | Error while appending file data message to a legally opened file. | *, #, ! |

| Error Code | Description | Message Code |
|---|---|---|
| 10 | Attempt to close an upload file which is not open. | *, ! |
| 11 | Error while attempting to close a user file. | *, #, ! |
| 12 | Error while attempting to close-purge a user file. | *, #, ! |
| 13 | Unable to find the directory entry on a close-purge operation. | *, ! |
| 14 | Directory write error during a close-purge operation. | *, #, ! |
| 15 | Page file open or a CREATE error. | *, #, ? |
| 16 | Illegal parameter passed to "PAGER" routine. | *, ? |
| 17 | Page out error returned by "PAGER" routine. | *, #, ? |
| 18 | Page in error returned by "PAGER" routine. | *, #, ? |
| 19 | User ID mismatch during close-purge operation. | *, ! |

### 6.2.2 Download Error Messages (DNLDR)

| Error Code | Description | Message Code |
|---|---|---|
| 00 | Operation performed without error. | ! |
| 50 | Page file open error. | *, #, ? |
| 51 | Page file CREATE error. | *,#, ? |
| 52 | Error while reading data from disk. | *, #, ! |
| 53 | Request for second download (on a particular subchannel before previous download has completed. | *, ! |
| 54 | Any error while searching a raw data directory. | *, ! |
| 55 | FMGR error while repositioning a download file. | *, #, ! |
| 56 | Illegal message type encountered. | * |
| 57 | Illegal CPU number found in a message. | * |
| 58 | Illegal subaddress found in a message. | * |
| 59 | WRITF error while paging out an old FDB. | *, #, ? |
| 60 | READF error while paging in a new FDB. | *, #, ? |
| 61 | WRITF error while posting to disk. | *, #, ? |
| 62 | POST error while posting to the disk. | *, #, ? |
| 63 | Error attempting to open raw data directory. | *, #, ! |
| 64 | Read error while searching raw data directory. | *, #, ! |
| 65 | Error while closing directory. | *, #, ! |
| 66 | Error opening user file for download. | *, #, ! |
| 67 | Underflow in output queue. | *, ? |

## 6.3 File Manager Error Codes

Table 6-1 lists the file manager (FMGR) error codes referenced in the bracks of the Integrator status messages.

**TABLE 6-1 FMCR ERROR CODES**

| Error Code | Message | Meaning & Corrective Action |
|---|---|---|
| 000 | (No error) | none |
| -001 | DISC ERROR | The disc is down; try again and then report it to the system manger of facility. |
| -002 | DUPLICATE FILE NAME | A file already exists with specified name; repeat with new name or purge existing file. |
| -003 | BACKSPACE ILLEGAL | Attempt was made to backspace a device (or type 0 file) that cannot be backspaced; check device type. |
| -004 | MORE THAN 32767 RECORDS IN A TYPE 2 FILE | Attempt was made to create a type 2 file with too many records or record size too large; check size paramter. |
| -005 | RECORD LENGTH ILLEGAL | Attempt to read or position a record not written, or on update to write an illegal record length; check position or size parameters. |
| -006 | CR OR FILE NOT FOUND OR NO ROOM | Attempt to access a cartridge or file that cannot be found or which has no more room; check the file name or cartridge number, if no more room on cartridge try another, or decrease file size (Cp pack cartridge). |
| -007 | BAD FILE SECURITY CODE | Attempt to access a file with no security code or the wrong code; find out the correct code and use it or do not access file. |
| -008 | FILE OPEN OR LOCK REJECTED | Attempt to open file already open exclusively or open to eight programs or cartridge containing file is locked; use CL or DL to locate lock; if file being packed, check if spool shut down. |

**TABLE 6-1 FMCR ERROR CODES**
**(Continued)**

| Error Code | Message | Meaning & Corrective Action |
|---|---|---|
| -009 | ATTEMPT TO USE APOSN OR FORCE TO 1 A TYPE 0 FILE | Type 0 files cannot be positioned with APOSN or be forced to type 1: check file type. |
| -010 | NOT ENOUGH PARAMTERS | Required parameters ommitted from call: enter the parameters. |
| -011 | DCB NOT OPEN | Attempt to access an unopened DCB; use CREATE, or OPEN to open DCB; check for errors. |
| -012 | EOF OR SOF ERROR | Attempt to read or write or position beyond the file boundaries: check record position parameters, result depends on file type & call. |
| -013 | DISC LOCKED | Cartridge is locked; initialize cartridge if not intialized, otherwise keep trying. |
| -014 | DIRECTORY FULL | No more room in file directory; pruse files and pack directory if possible, or try another cartridge. |
| -015 | ILLEGAL NAME | File name does not conform to syntax rules; correct name. |
| -016 | ILLEGAL TYPE OR SIZE = 0 | Wrong type code supplied; attempt to create or purge type 0 file or create 0-length file; check size and type parameters. |
| -017 | ILLEGAL READ/WRITE ON TYPE 0 FILE | Attempt to read/write or position type 0 file that does not support the operation; check file paramters, from FMGR check namr. |
| -101 | ILLEGAL PARAMETER IN D. RTR CALL | Possible operator error; recheck previous entries for illegal or misplaced parameters. |
| -102 | ILLEGAL D. RTR CALL SEQUENCE | Lock not requested first or file not opened exclusively; possible operator error such as removal of cartridge without DC command. |

Except for 10 and 11, any of these error codes can be returned from FMGR.

The matrix shown in Table 6-2 shows which interface routines can be the cause of each error. For example, error 08 can occur as a result of the PURGE, OPEN, or NAMF routines. Since these routines can be called by FMGR, the matrix also indicates the errors that result in a transfer to the log device (X) and those that do not (0).

## TABLE 6-2 FILE MANAGER ERROR AND ROUTINE MATRIX

| Code | Message | CREAT | PURGE | OPEN | CLOSE | READF | WRITF | LOCF | APOSN* | RWNDF | POSNT | FCONT* | NAMF | POST | IDCBS* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -001 | DISC ERROR | X | X | X | X | X | X | | 0 | X | X | | X | X | |
| -002 | DUPLICATE FILE NAME | X | | | | | | | | | | | X | | |
| -003 | BACKSPACE ILLEGAL | | | | | | | | | | X | | | | |
| -004 | MORE THAN 32786 RECORDS IN A TYPE 2 FILE | X | | | | | | | | | | | | | |
| -005 | RECORD LENGTH ILLEGAL | | | | | X | X | | 0 | | X | | | | |
| -006 | CR OR FILE NOT FOUND OR NO ROOM | X | 0 | X | X | | X | | | | | | X | | |
| -007 | BAD FILE SECURITY CODE | | X | X | | | X | | | | | | X | | |
| -008 | FILE OPEN OR LOCK REJECTED | | X | X | | | | | | | | | X | | |
| -009 | ATTEMPT TO USE APOSN OR FORCE TO 1 A TYPE 0 FILE | | | X | | | | | 0 | | | | | | |
| *-010 | NOT ENOUGH PARAM-ETERS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | | 0 | | |
| *-011 | DCB NOT OPEN | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |
| -012 | EOF OR SOF ERROR | | | | | X | X | | 0 | | X | 0 | | | |
| -013 | DISC LOCKED | X | X | X | | | | | | | | | X | | |
| -014 | DIRECTORY FULL | X | | | | | X | | | | | | | | |
| -015 | ILLEGAL NAME | X | | | | | | | | | | | X | | |
| -016 | ILLEGAL TYPE OR SIZE=0 | X | X | | | | | | | | | | | | |
| -017 | ILLEGAL READ/WRITE ON TYPE 0 FILE | | | | | X | X | | | | X | | | | |

X - transfer to log device (FMGR)

0 - no transfer to log device

* - error never returned to FMGR or routine never called by FMGR

# SECTION 7

# RS232 DATA SET INTERFACE SELF TEST DIAGNOSTIC

## 7.1 TEST OPTIONS

The data set interface self-test diagnostic is composed of seven test options. Tests 1 through 4 are performed on auto request, 5 through 7 require operator intervention and are manual test options.

The seven tests perform the following functions:

1 - DSC, DTR, DSR
2 - DSC, RTS, CTS, RLSD
3 - Transmit/receive ASCII character
4 - Transmit/receive binary character
5 - BAUD rate verification
6 - Transmit loop scope check
7 - Receiver loop scope check

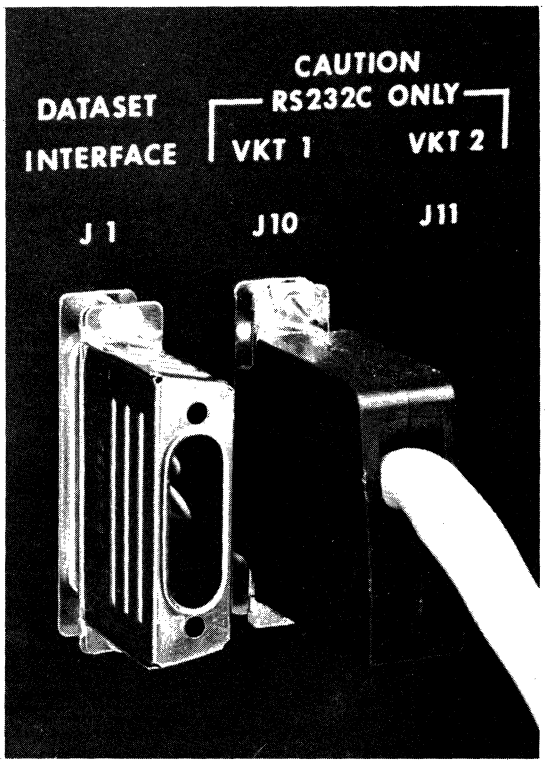
The data set interface is verified by attaching a termination jack to the output plug of the data set. The jack connects the signals as shown in Figure 7-1. This simulates a condition where the data set is both a transmitter and receiver. All control, status, interrupt, transmit, and receiver functions are verified except for the stack function. A failure in any test should be correct before proceeding to the next test. The data set interface has been designed such that a failure in the string of events causes the sequencer to hang up. By determining which sequence has caused the hang up, the problem can be isolated. The sequencer status register should indicate at any time what is happening in the data set interface.

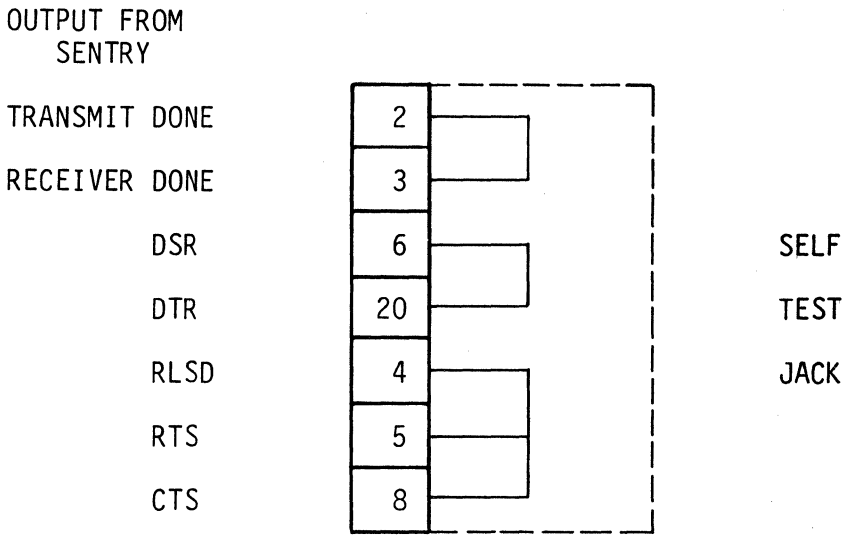
## 7.2 EXECUTION

### 7.2.1 Clink

CLINK is a coreimage program executed under control of DOPSY by entering the following command:

```
// EXEC 'CLINK'
```




L120

**Figure 7-1 Self Test Jack and Signal Connection**

The program responds with the following output:

```
COMM LINK SELF-TEST DIAGNOSTIC
***ATTACH SELF-TEST PLUG***
AUTO TESTS ARE 9600 BAUD
AUTO-MANUAL-EXIT
```

7.2.1.1 AUTO

If A is entered, the four auto tests are performed with pass/fail status information displayed on the VKT.

7.2.1.2 MANUAL

If M is entered, the following table is displayed on the VKT:

```
SELECT TEST TO BE PERFORMED

1.  SEND DTR - RECEIVE DSR
2.  SEND RTS - RECEIVE CTS/RLSD
3.  TRANSMIT OCTAL 125B
4.  TRANSMIT BINARY 252B
5.  BAUD RATE TEST
6.  TRANSMIT LOOP
7.  RECEIVE LOOP
```

To select one of the seven tests for execution the number of the test desired must be entered. Only one test may be entered at a time.

If any one of the first five tests are chosen for execution, upon completion the list of tests is redisplayed and the user may enter another test for execution. The transmit (test 6) and receive (test 7) loop tests are special cases and require additional set up. These two tests remain in the loop until terminated by depressing the CR key on the VKT. Refer to each test description for an explanation of the test structure.

7.2.1.3 EXIT

An E may be entered to exit the CLINK diagnostic program and return control back to the DOPSY monitor. E can only be entered whenever the AUTO-MANUAL-EXIT message is displayed.

## 7.3 DESCRIPTIONS

### 7.3.1 Test 1 and Test 2

Tests 1 and 2 verify the data set control and status signals. The functions tested are as follows:

DSC - Data set change interrupt)
DTR - Data terminal ready (control)
DSR - Data set ready (status)
RTS - Request to send (control)
CTS - Clear to send (status)
RLSD - Line signal detected (status)

To ensure that the data set can be reset and that the sequencer is in an idle state, a data set reset is issued. The sequencer status is read and checked for SEQ Ø. If the sequencer cannot be reset, an error message is printed showing the sequence number read from the status register. Control is returned to DOPSY for further user instructions.

With the sequencer at idle (D1 on), the transmit mode is set and the data set interrupt enabled. Light D1 should not be on, indicating the sequencer has left SEQ0. D5 should now be one. The DTR command is loaded to the transfer function register. A 60 s wait loop is entered, from which a DSC interrupt should occur. The data set status is read and verified that a DSR signal was detected. In test 2 the data set is again reset and verified that the sequencer is idle. The RTS entered, a DSC interrupt should occur, the status register should indicate that CTS and RLSD occurred. Refer to the error message section if an error is indicated.

### 7.3.2 Test 3 and Test 4

Tests 3 and 4 verify the ability of the data set to transmit and receive ASCII and binary characters. The steps are similar to tests 1 and 2, where the DTR and RTS commands are sent, a DSC Interrupt is detected, and the DSR, CTS, and RLSD status bits are checked. At the completion of the sequence, the character is transmitted. A wait loop is entered to allow the transmitter to complete its operation. The data set status is then read and checked for the transmit done status. Another wait loop is entered,then the reciever done status is checked. Should the transmit done or receive done interrupt fail to occur, an error message will indicate the sequencer status at the time of the error. Test 6 and 7 may need to be executed to detect the problem.

When receiver done interrupt has occurred, the UART status is read and checked for data available and data error. If no error is detected, the data set interface has performed all functions of transmitting and receiving one character of ASCII and binary data.

### 7.3.3 Test 5

Test 5 checks the ability of the data set interface to operate at various BAUD rates. The normal execution of tests 3 and 4 is at 9600 BAUD. Four switches on the data set interface board #2 are used to select the desired BAUD rate. Upon entering test 5, the following table is displayed on the VKT.

SET BAUD RATE SWITCHES BD #2

110 - ALL OFF
300 - S2 ON
600 - S1-S4 ON
1200 - S3 ON
ENTER BAUD TEST NUMBER (1, 2, 3, 4)

The four BAUD rate switches should be set according to the BAUD rate test desired, then answer the test number request. The transmit/receive ASCII character test is performed at the requested BAUD rate. After completing the test, the AUTO-MANUAL-EXIT query is again displayed for the next user request. The BAUD rate switches should be returned to the normal operating configuration when done with test 5.

### 7.3.4 Test 6

Test 6 is a debug loop to be used when a problem is suspected in the transmitter section of the data set interface. When in this mode, the transmitter limit counter is inhibited so that the transmitter cycles between sequence 3 and 5 until terminated. Upon entering test 6, the following messages are displayed on the VKT:

TRANSMIT LOOP TEST
RECYCLE SEQ3 to SEQ5
DEPRESS CR TO EXIT LOOP

Providing the buffer loading and limit reset occurred in SEQ1 and SEQ2 none of the sequence lights should be on. If D1 is on, a transmit mode has not been entered and the sequence is still at idle. The TRBLD signal should be checked. If the sequencer has left SEQ0, the signals at SEQ3, SEQ4, and SEQ5 should be checked to ensure that each sequence can be completed. Refer to Table 7-2 for events at each sequence, the event which causes the sequencer to proceed to the next event and possible cause of a sequence hang up.

When it is desired to terminate the transmitter loop, depress CR on the VKT. The AUTO-MANUAL-EXIT request again is displayed for user request.

### 7.3.5 Test 7

Test 7 is a debug loop to be used when a problem is suspected in the receiver section of the data set interface. Two switches on data set interface board #2 must be used to cycle the receiver section. Upon entering test 7, the following messages are displayed on the VKT:

RECEIVER LOOP TEST
SET SW6 ON (BD #2)
TO EXIT RECEIVER LOOP

1. SET SW6 OFF
2. SET SW5 ON, THEN OFF
3. DEPRESS CR ON VKT

When the above instructions have been displayed on the VKT, set switch 6 to on position. Sequence 7 is entered with no limit programmed. The receiver cycles between SEQ7 and SEQ10 until terminated. The RLSD signal and the signals at each sequence should be checked to ensure that each sequence can be completed. Any of the signals as shown in Table 7-2 for each sequence that does not occur causes the receiver to hang up.

To exit the receiver loop, the procedure shown above should be followed.

## 7.4 ERROR MESSAGES

On a detected error, a message as shown in Table 7-1 is displayed on the VKT. The error associated with three types of cautions are:

Data set change
Transmitter error
Receiver error

Errors 1, 2, and 3 of Table 7-1 reflect dataset change errors. The dataset must be activated prior to transmitting or receiving a character. A failure in this group indicates that the dataset could not be made ready.

Error 4 indicates that a ASCII or binary character could not be transmitted. The sequencer status is printed to indicate which sequence the hang up occurred. Test option 3 may be executed to enter a transmit loop for scope debugging of the circuit. By referring to the sequence number, the events causing the problem can be determined.

TABLE 7-1

DATASET, TRANSMITTER AND RECEIVER ERROR MESSAGES

| Error Message | Cause of Error Message | Possible Fault |
|---|---|---|
| 1. DATA SET CHANGE NOT RECEIVED | Failed to detect data set change interrupt when CTS, DSR, or RLSD signal line changed. (data link status DB23) | |
| 2. DATA SET READY NOT RECEIVED | Failed to detect DSR (DB1) on RS232 status when DTR command is issued. | B3 on DB #2 |
| 3. CLEAR TO SEND NOT RECEIVED RSLD NOT REQUIRED | Failed to detect CTS (DB0) or DSLD (DB2) on RS232 status when RTS command is issued. | C2 or B3 on DB #2 |
| 4. XMIT DONE NOT RECEIVED | Failed to detect transmitter done interrupt when transmitting signle character (data link status DB20). Refer to sequencer status. | |
| 5. RECEIVER DONE NOT RECEIVED | Failed to detect receiver done interrupt when transmitting single character (data link status DB21). Refer to sequencer status. | |
| 6. SEQUENCER STATUS SEQ 0-10 | Refer to Table 7-3. | |
| 7. DATA AVAILABLE NOT RECEIVED | Failed to detect via available status (UART DB0) after receiver done. | |
| 8. DATA COMPARE ERROR | The character read from receiver does not compare with transmitted character. | |
| 9. INTERRUPT WITH NO BITS SET | Interrupt occurred at data link address with no bits set in data link status register. | |
| 10. UART STATUS ERROR | Refer to Table 7-5. | |

## TABLE 7-2

## TRANSMITTER/RECEIVER SEQUENCE ERRORS

| Sequence | Sequence Event | Possible Cause of Error |
|---|---|---|
| SEQ1 | Load transmitter buffer.<br>Reset limit counter. | |
| SEQ 2 | Proceed at end of reset. | |
| | Increment limit counter. | UART or transmitter buffer not ready. |
| | Monitor buffer ready & UART ready status. | |
| SEQ 3 | Proceed at end of buffer delay. | |
| | Load UART with character and monitor UART status for E0C. | Unable to transmit character. Did not detect EOC signal high. |
| | Proceed at end of transmission | |
| SEQ 4 | Check for limit character. | Character limit not reached. |
| SEQ 5 | Proceed when limit reached. | |
| SEQ 6 | Create transmitter done interrupt. Wait for PCOMP ACK. | Transmitter done interrupt not issued or PCOMP did not reset TRBLD command. |
| | Monitor UART for data available Monitor receiver buffer ready. | UART data available status missing. |
| SEQ 7 | Proceed at data available. | |
| | Latch UART status data. | Unable to latch status. |
| SEQ 8 | Proceed at end of latch. | |
| | Latch UART receiver data clock into memory. | Unable to latch receiver data. |
| SEQ 9 | Proceed at end of latch. | |
| | Check for limit character and continue when limit reached. | Character limit not reached. Receiver buffer flag not ready. |
| SEQ 10 | Proceed when buffer flag ready. | |
| SEQ 11 | Create receiver done interrupt. Wait for PCOMP ACK. | Receiver done interuupt not issued. PCOMP did not reset RLSD. |

**TABLE 7-3 UART STATUS ERRORS**

| Error Message | Cause of Error Message |
|---|---|
| TRANS. BUFFER EMPTY | |
| END OF CHAR. | Transmitter register empty. |
| PARITY ERROR | An even number of bits not received. |
| FRAME ERROR | Incorrect number of bits found in serial bit stream dictated by rate. |
| OVER RUN ERROR | Next character was jammed into the receiver hold register before current character was read into CPU. |

## 7.5 TRANSMIT SEQUENCE

The following flowchart shows the Transmit sequence.

SEQ 0

IDLE

STATE

SEQ 1

| | |
|---|---|
| TRANSMIT BUFFER LOAD | 1. Load buffer with number of characters indicated by limit counter<br>2. Go to SEQ 2 when limit is reached |

SEQ 2

| | |
|---|---|
| RESET LIMIT COUNTER | 1. Reset limit counter, go to SEQ 3 when limit is zero. |

SEQ 3

| | |
|---|---|
| MONITOR BUFFER READY & UART STATUS | 1. Increment limit<br>2. Monitor buffer ready and UART status, proceed when buffer delay complete. |

SEQ 4

| | |
|---|---|
| TRANSMIT CHARACTER | 1. Load UART with character<br>2. Monitor UART status for EOC, proceed at end of transmission. |

SEQ 5

| | |
|---|---|
| MONITOR LIMIT | 1. Compare count against limit counter<br>2. If limit reached, go to SEQ 6, otherwise go to SEQ 3. |

SEQ 6

| | |
|---|---|
| CREATE TRANSMIT DONE INTERRUPT | 1. Create transmit done interrupt<br>2. Wait for PCOMP<br>3. Reset limit counter and transmitter buffer<br>4. Return to SEQ 1 |

## 7.6 RECEIVER SEQUENCE

The following flowchart shows the Receiver sequence:

SEQ 0

IDLE
STATE

SEQ 7

| | |
|---|---|
| MONITOR<br>UART STATUS &<br>BUFFER READY | 1. Increment limit counter<br>2. Proceed at data available |

SEQ 8

| | |
|---|---|
| LATCH UART<br>STATUS DATA | 1. Generate latch status data pulse<br>2. Proceed at end of latch status |

SEQ 9

| | |
|---|---|
| LATCH UART<br>RECEIVE<br>DATA | 1. Generate latch receive data pulse<br>2. Clock data into memory<br>3. Proceed at end of latch data |

SEQ 10

| | |
|---|---|
| MONITOR<br>LIMIT | 1. Compare program limit against increment count<br>2. If no limit, go back to SEQ 7, else continue |

SEQ 11

| | |
|---|---|
| CREATE<br>RECEIVE DONE<br>INTERRUPT | 1. Create receive done interrupt<br>2. Wait for PCOMP<br>3. Reset limit counter and receiver buffer<br>4. Return to SEQ 0 |

# SECTION 8

# RS232 LOGIC DESCRIPTION

## 8.1 INTRODUCTION

The Data Set Interface consists of two printed circuit boards; Data Set Board 1 (Figure 8-2) and Data Set Board 2 (Figure 8-3). They interface with the FST-2 CPU plugging into the CPU backplane in slots E5 (Board 1) and E7 (Board 2) (Figure 8-1). Board 2 requires a ±15VDC power cable coming from the M2 chassis and a Data Set I/O cable going to the Sentry I/O bulkhead.

The FST-2/Data Set Interface is accomplished using the accumulator (N-Bus) with a Select Peripheral Unit (SPU) address of 130 - 137. The priority interrupt address of 30 - 37 follows the least two significant digits of of the SPU address. In this manner up to eight Data Set Interfaces maybe driven by the processor. Switches on Board 1 determine the least significant digit for SPU and priority address. A list of the command mnemonics, descriptions and opcodes are included in the Data Set Interface Board 1 block diagram (Figure 8-4). The two sheets of Board 1's block diagram give an overview of the relationships between Board 1, Board 2 and the FST-2.

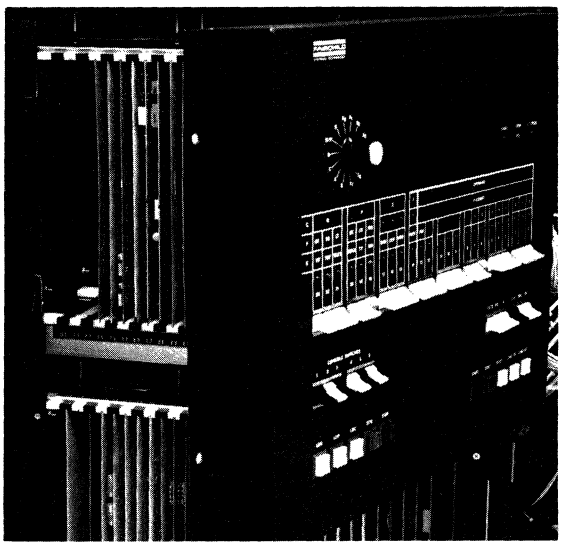


Figure 8-1 Data Set Interface Board Locations

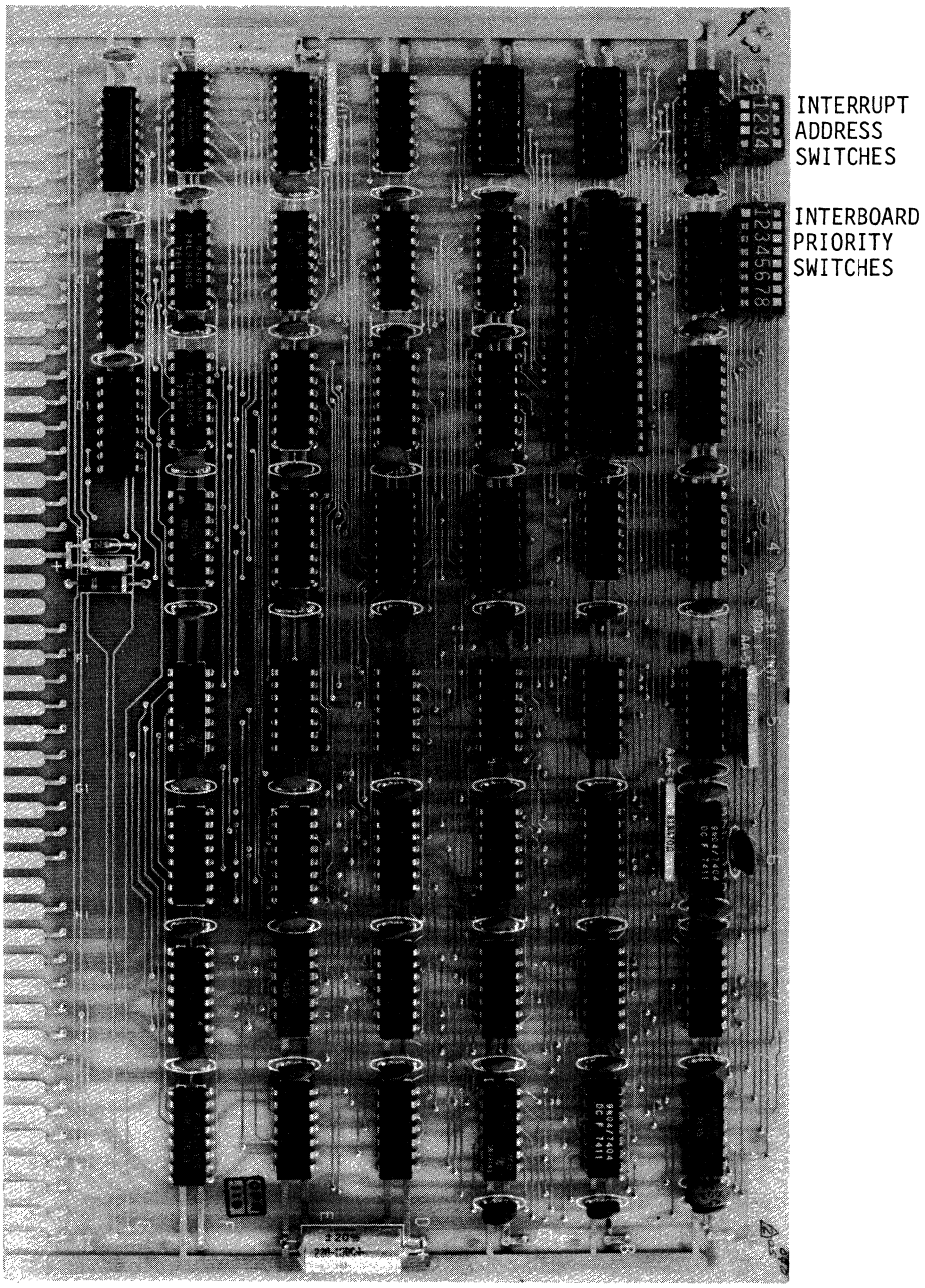



**Figure 8-2 Data Set Interface Board #1**

## 8.2 DATA SET INTERFACE BOARD #1 DESCRIPTION

### 8.2.1 Circuits

Board #1 consists of the following circuits:

- FST-2 Accumulator Bus Interface
- FST-2 Timing Bus Interface
- Command Data Decode Logic
- Priority Interrupt Logic
- Interrupt Address Logic
- UART and Fifo Memory with Control Logic
- RS232C Status Register

The principle function of Board #1 is to decode SPU commands and issue appropriate responses in managing data flow and starting sequences on Board #2. Board #1 has two data buses; FST Out Data Bus, and FST In Data Bus. All data coming from the FST-2 N-Bus enters Board #1 via the FST Out Data Bus, all data exiting the Interface and entering the FST-2 does so on the In Data Bus.

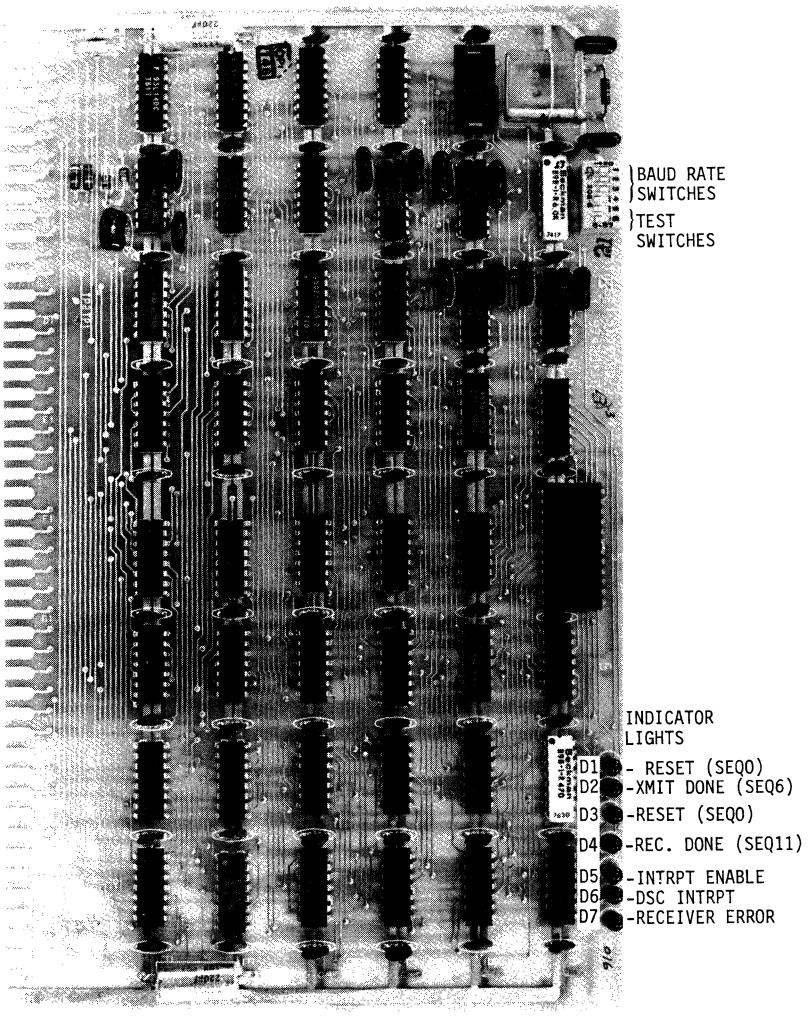



**Figure 8-3 Data Set Interface Board #2**

### 8.2.2 SPU Command Decoding

An SPU command from the FST-2 is decoded as follows:

1. The Interface sees its address come true at PS time, and latches the command data.

2. If the command is an I/O instruction a data transfer is required. All Data transfers occur at time T1. Therefore, at PS time the command is latched, and at time T1 is executed. The following time T2 resets the I/O cycle latches (clears the latched command after execution).

3. If the SPU is an interface command only (no data transfer in or out), then the Interface Cycle Only Latch (which is on for the duration of the PS Time) enables a decoder for the PS time cycle, after which the Interface Cycle Only Latch is reset.

4. The appropriate I/O cycle and Interface Only Cycle commands are shown on Board 1's Block diagram (Figure 8-4, sheet 1).

### 8.2.3 Interrupt Sequence

An Interrupt Sequence is initiated as follows:

1. PON/(an Interface Only Cycle) command instructs the interrupt priority circuit to turn on.

2. The interrupt priority circuit, determines two things before intiating interrupt action.

   a. Does Board 2 have interrupt requests pending?

   b. Does a higher priority device control the line?

3. If Board 2 generates an interrupt request, and no higher priority is on line, then Board 1 gates priority data bits at the next interrupt priority (IP) cycle and sets the interrupt in Process F/F. This locks out other device requests of a lower priority.

4. If the above priority sequence has taken place, the interrupt address is gated at the next T5 clock time and the circuitry guarantees the address is generated only once.

5. Once the interrupt in process F/F has been latched each IP cycle causes the interrupt circuitry to gate and data bus interface's priority data bits. When the FST-2 has recognized the interrupt, it acknowledges with a PCOMP/ (Priority Complete). This resets the interrupt circuitry completing the interrupt processing cycle; if PON/ (Priority On) is still valid the interface may issue a new interrupt immediately pending an interrupt request from Board 2.

6. The POFF/ command (Priority Off) resets a PON/ condition and therefore rests and disables interrupt circuitry on Board #1. Any pending interrupts must await the PON/ command before an interrupt sequence can begin.

### 8.2.4 N-Bus Timing Signals

FST-2 N-B Timing Signals are used as follows:

1. T1 time: if a data transfer I/O cycle command has been latched, data transfer takes place for the duration of the T1 cycle.

2. T2 time: resets I/O cycle latches, and synchronizes the Integrated sequence processor on Board #2 to the FST-2.

3. IP time: resolves interrupt priority and determines the device address if interrupt request, priority, and PON/ conditions are met.

4. PS time: enables command latch clocking if the Data Set Interface is addressed.

5. T5 time: if an interrupt priority (IP) signal was issued, causes a one time reading of the device interrupt address.

### 8.2.5 RS232C Status Register

The RS232 status register receives its signals from the RS232C circuitry on Data Set Board #2. When the I/O cycle command RRSS/ is decoded and latched at PS time, at the next T1 the RS232C status is placed on the FST in Data Bus.

### 8.2.6 UART and FIFO Memory Buffer

8.2.6.1 UART:

UART means Universal Asynchronous Receiver and Transmitter. The 4 registers listed below are contained within the device:

1. Transmit Data Input Register

2. Receiver Data Output Register

3. UART Status of Operation Register

4. UART Mode Control Register

The UART is used in an asynchronous manner indicating that the transmit function is independent of the receive function. A single clock called Baud Rate (Data Bit Transfer Rate) is used to serialize and de-serialize transmit and receive data (both operating at the same clock rate). The basic function of a UART is to convert parallel input transmitter data to serial, add start, stop and parity bits to the input data, then tramsit this in serial format to a receiver of like kind. The receiver accepts the start bit, then retrieves the serial received data and checks

it for parity, framing or overrun errors. The last thing the receiver does, is convert the serial received data to a parallel format and places it in the receiver data register.

8.2.6.2 FIFO MEMORY:

The FIFO memory (first in first out memory) allows data entered in a specific sequence to be retrieved from the output of the FIFO in the identical sequence. Asynchronous to the input.

FIFO Memories have been inserted between the parallel I/O data ports of the UART and the I/O data lines of the FST-2 N-Bus to allow the Data Set Interface to receive and transmit data in a buffered block mode. Referring to Figure 8-4, sheet 2, a data transmission cycle occurs as follows:

1. A baud clock is supplied to the UART form Board #2. This clock is used to clock the serial transmit and receive circuitry of the UART at identical rates.

2. The data to be transmitted enters the transmitter data latch from FST Out Data Bus during the time T1 following decoding a LUXR/command at PS time of an I/O cycle.

3. As LUXR/ returns to a high state (+5), the FIFO control logic notifies the FIFO that valid data has been latched and the FIFO enters an input cycle if the decoded sequence from Board #2 indicates sequence 1 state is valid. Upon shifting the data through its cells to the end (last row for first out) the FIFO responds by raising its output ready flag.

4. With Data at the output of the FIFO memory, the control register of the UART must be intialized. XWRD being a '1' indicates that 8-bit binary data transmission is to take place, consequently the UART control register must be conditioned such that no parity bit is added to transmitted data bytes. If XWRD is a '0' ASCII data transmission is called for and an even parity bit is appended to each byte. In this manner 7-bit ASCII or 8-bit binary data may be transmitted through the data set interface with no change in transmission word length.

5. When the integrated sequencer on Board #2 reaches sequence 4, a negative going pulse is developed by the UART pulsing network going to the UART DS input as transmit strobe. This strobe causes the UART to input the Parallel Data from the FIFO memory, add start, stop, and parity bits, then transmits it in serial format to Board #2's RS232C buffer. The end of character transmission is recognized by the transmission complete logic which notifies the sequencer on Board #2 by the Transcom line going high.

6. When the Integrated Sequencer on Board #2 reads UART status the signal SWE/ (Status Word Enable) is generated. SWE gates TBMT (Transmitter Buffer Empty) onto status condition lines to the sequencer. If the Transmitter Buffer is empty, recognized by TBMT going high, the next transmit character can be clocked into the UART transmitter.

7. The Clock Out (1) signal going to the FIFO CTL Logic causes the next data byte to be clocked to the output of the FIFO memory. (Next in line for transmission.)

8. LUSD/ (Latch UART Status Data) is a command used during an interrupt service routine to interrogate UART Status. This command causes the present status of the UART to be latched. Thus the Interface command cycle LUSD/ followed by a RUSR/ (Read UART Status Request) command results in UART status being gated onto the FST-2 in data bus during T1 time.

Referring to Figure 8-4, sheet 2, a receive cycle takes place as follows:

1. Board #1 receives serial input data via the RS232C I/O buffer on Board #2. Parity, framing, and overrun error conditions are checked and are caught at this time by the UART's internal check circuitry. The serial data, with start, stop and parity bits removed is gated in parallel format to the UART receive register, and the status of the received data is available in the UART status register.

2. The sequencer Board #2 generates SWE/ and reads a DA (Data Available) true condition on one of its lines and sends sequence 8/ which latches the UART status data. The sequencer then pauses until acknowledgement of the status being latched by LDS/ (Latch Status Data) going high.

3. The sequencer on Board #2 next generates sequence 9/ which latches the UART received data into the received data latch and resets the DA flag. The completion of this sequence is realized by the signal LRD/ (Latch Received Data) going high.

4. Next the sequencer on Board #2 looks at the Received Data FIFO Memory control lines to see if the input 2 ready is high indicating the memory is ready to accept data. If so, the sequencer generates clock-in 2 which clocks the latched received data into the FIFO memory. When the data has reached the last row (first in - first out) the FIFO memory raises its output (2) ready flag to inform the sequencer.

5. When UART status is latched the receiver error logic circuitry raises the RERR (Receiver Error) signal to be generated to Board #2 if either a parity, framing or overrun error condition has occurred.

6. The FST-2 generates the SPU RURR/ to read received data. (Read UART Receive Register). This SPU requires a data I/O cycle therefore, at T1 time the received data is multiplexed onto the FST-2 input data bus.

7. The FST-2 generates the SPU RUSR/ (Read UART Status Request) to read UART Status Data. This signal requies a data I/O cycle, therefore, at T1 time the UART status data is multiplexed onto the FST-2 input data bus.

8. An interrupt request is generated by the sequencer on Board #2 when a receive or transmit cycle is complete. Board #1 receives the interrupt

request from Board #2, and then generates the priority and interrupt address. The FST-2 then reads the data set status register on Board #2 to determine which of the following conditions have interrupted the processor:

a. Transmitter Done
b. Receiver Done
c. Receiver Error
d. Data Set Change

The above status signals are discussed at greater length in the Board #2 operation description (Section 8.3).

## 8.3 DATA SET INTERFACE BOARD #2 DESCRIPTION

### 8.3.1 Circuits

Board #2 contains the following circuitry:

- Programmable Baud Rate Generator
- Transfer Function Register
- RS232 I/O Circuit with Data Set Change Detection
- Master Reset Pulse Network
- Limit Control Counter
- Sequencer State L.E.D. Display
- Data Link Status Register
- Interrupt Request Logic
- Sequence Controller with Status Register

The principle function of Data Set Interface Board #2 is to provide operational control of the FIFO memory and the UART. This control is performed by a sequence counter and decoding array which essentially runs the UART and FIFO memory though predetermined sequences. When the last sequence of a transmit or a receive function is reached, the interface notifies the FST-2 via an interrupt request which is issued by Board #1. The FST-2 in turn (during an interrupt service routine) fetches the Data Link Status Register to determine what event on the interface has cuased the interrupt. Upon determining the function to be serviced, the FST-2 proceeds to service the interrupt function, and upon completion notifies the interface via the PCOMP/command. The PCOMP/command resets the sequence counter to its idle state.

### 8.3.2 Programmable Baud Rate Generator

The programmable baud rate generator consists of the following:

1. A one chip programmable bit rate generator using a crystal reference is used to establish the baud rate.

2. The bit rate generator is programmed with four rate select switches (S1-1 through S1-4). The ON Position of the switches is towards the front of the P.C. Board and the OFF position is towards the edge finder end of the P.C. board. The switches are in a minidip package located on the front edge corner of the P.C. board.

3. The UART on Board #1 requires a bit rate frequency of 16 times the actual baud rate selected for data transmission. The bit rate generator compensates these requirements by generating clock rates 16 times the selected baud rate.

4. Refer to Figure 8-4 sheet 1 of 2 where a chart of programmable baud rates is included with the necessary positions of switches S1-1 through S1-4.

### 8.3.3 Transfer Fuontion Register

The Transfer Function Register description is as follows:

1. Refer to Figure 8-5, sheet 1 of Board #2 block diagram upper left-hand corner and locate the transfer function register with RS232C logic and the character limit counter.

2. The transfer function register is latched with Board #1's decoded SPU I/O cycle command LXFR/ (Load Transfer Function Register).

3. The purpose of the Transfer Function Register is to latch the RS232C signals DTR (DAta Terminal Ready) and RTS (Request to Send) which are driven at the interface. The break bit is included if a "break" or open send line condition is required.

4. In addition to latching the RS232 signals listed above, it also conditions the UART (with XWRD) appropriately depending on whether the data transfer mode is ASCII or binary data. XWRD being true (+5) indicates binary mode and false (0) indicates ASCII data mode. The remaining 3-bit limitation control word of the register sets a limit counter to what the programmed character limit is going to be. The number of data words to be transmitted or received per buffered Block is 1, 4, 8, 16, 32 or 64.

5. The remaining bit in this register puts the interface into a test mode. When used in conjunction with switches S1-5 and S1-6, a diagnostic program can cause the transmitter or receive sequences to enter an autocycle mode. This mode of operation is discussed in detail in the system diagnostics' package for the data set interface.

### 8.3.4 RS232 I/O Interface

RS232C I/O Interface Circuitry description is as follows:

1. Refer to Figure 8-5, sheet 1 of Board #2 block diagram and note that +12 and -12 volts are supplied to the RS232C drivers. The driver ships are level converters which converts an input (0) logic level to a -12v output level, an input of (+5) lgoic level converts to +12v at the output. Therefore, all signals on the RS232C interface are at either +12v or -12v levels.

2. The RS232C receiver circuits shown just below the drivers on the block diagram convert the $\pm$ 12v levels of RS232C received signals to TTL logic levels.

3. RD (Received Data) goes through inhibit logic directly to the UART and the RS232C status register on Board #1. The inhibit logic prevents the UART of Board #1 from receiving noise whenever RLSD (Receive Line Signal Detector) goes low (indicating that no valid receiver signal is present).

4. DSR/ (Data Set Ready), CTS (Clear to Send), and RLSD (Receive Line Signal Detector) are all going to edge detection circuitry. The edge detector generates a pulse if any of these 3 signals going through a high to low or low to high transition and causes a latch to hold DSC (Data Set Change). DSC causes an interrupt request to be issued as shown in Figure 8-5, sheet 2 of Board #2 block diagram. This notifies theFST-2 that the RS232C interface has a signal line change of state. The same signals are also input to a register on Board #1 where the FST-2 by reading the RS232C status Register can determine which signal line changed.

5. The RS232C signal RI (Ring Indicator) goes high for 1 second and low for 2 seconds during a data link connect sequence. The RI line is also brought to the status register on Board #1 so that possible problems between the DAA (Data Access Arrangement) and Data Set can be diagnosed. In a data set the RI signal causes DSR to be set which notifies the FST-2 with an interrupt that the host is on line. The FST-2 must set DTR (Data Terminal Ready) signal true, otherwise DSR remains low.

### 8.3.5 MASTR Reset Pulse Network

The MASTR Reset Pulse Network operates as follows:

1. Refer to Figure 8-5, Sheet 1 of Board #2 block diagram. In the lower left-hand corner a one shot circuit with two input signals IRST/ (Interface Reset) and RESET/ is shown. IRST/ is a decoded interface only cycle command which comes from Board #1 via an FST-2 SPU and directs all circuitry to a reinitialized state. RESET/ is a signal derived from RESET switch on the CPU front panel that when depressed also causes the interface to be reinitialized. All current operations are suspended during reset and all registers are returned to a zero state pending new instructions to begin an operation.

### 8.3.6 Limit Control Counter

The Limit Control Counter consists of the following:

1. The Limit Control Counter briefly described in the transfer register discussion, is shown at top left corner of Figure 8-2 sheet 1 of Board #2 block diagram. The binary counter chain while clocked with INC LIM (Increment Limit) places its output data inot a 1 of 8 multiplexer (used as a 1 of 7 mux). The transfer function register holds the multiplexer's three control select lines which make the 1 of 7 choise of which binary counter output is to be the limit control. From looking at the binary counter it can be seen that 1, 2, 4, 8, 16, 32 or 64 are the count values which can be selected as the character buffer size.

2. When the selected limit (counter value) comes true, the preselected counter output going through the multiplexer halts the counter. The output logic places LIMIT CTL on an output line to the sequencer shown in Figure 8-5, sheet 2 of the block diagram. The sequencer interprets this signal as follows:

a. If in sequence 1, the Transmit FIFO is loaded with proper number of characters.
b. If in sequence 5, the UART has transmitted the correct number of characters.
c. If in sequence 10, the Receiver FIFO is loaded with proper number of characters.

### 8.3.7 Sequencer State LED Display

Seven L.E.D.s with driver logic are located at the top right-hand corner of P.C. Board #2 and are assigned D1 through D7 as names. D1 through D7 relate to the following states of the sequencer:

1. D1 indicates that the sequencer is in the idle or SEQ 0 state. (No operation in process; waiting for command.)

2. D2 indicates that a transmission sequence has taken place and the sequencer is waiting at sequence #6 for a PCOMP (FST-2 acknowledgement). The acknowledgement puts the sequencer back to SEQ 0 state lighting led #D1.

3. D3 indicates that data set signal RLSD has initiated automatic receive mode and the sequencer has entered sequence #7 waiting for the DA (Data Available) signal from the UART to come true before beginning a receive sequence process.

4. D4 indicates that received data and status have been processed and the sequencer is in sequence #11 waiting for a PCOMP (FST-2 acknowledgement).

5. D5 indicates that the PON/, (Interface only command), has enabled the interrupt circuitry on Board #1.

6. D6 indicates that a DSC (Data Set Change) has been detected by the edge detection circuitry which monitors CTS, RLSD and DSR signals at the RS232C interface.

7. D7 indicates that an error has been detected on Board #1 in the receive data. This lamp can only be turned on when the receive sequence is complete (sequence 11) since more than 1 character may have been received. The first receiver error detected latches and saves the error flag. Sequence 11 then displays this error on the D7 lamp.

### 8.3.8 Data Link Status Register

The Data Link Status Register consists of the following:

1. Four discrete signals which can be monitored as N-Bus data bits BN20-BN23 make up the Data Link Status Register.

2. The purpose of this register is to inform the FST-2 what condition has caused an interrupt. The four signals with their respective data bits are shown in Figure 8-5, sheet 2 of Board #2 block diagram and are explained as follows:

   a. N-Bus bit-20:

      TD (Transmitter Done), this informs the FST-2 that a transmit sequence has been accomplished and the interface is waiting for acknowledgement.

   b. N-Bus bit-21:

      RD (Receive Done), this notifies the FST-2 that a receive sequence has been accomplished and the interface is waiting for acknowledgement.

   c. N-Bus bit-22:

      RERR (Receive Error), this notifies the FST-2 that some character or characters received in the receive data block contains an error.

   d. N-Bus bit-23:

      DSC (Data Set Change), this notifies the FST-2 that the RS232 edge detector circuit has detected a state change of either DSR (Data Set Ready), CTS (Clear to Send) or RLSD (Rec. Line Signal Detector).

3. Each time the Data Set Interface is addressed by the FST-2, the status bits are made available on N-Bus. An SPU command status (06000130) which is an SPU NO OP also places these status bits onto the N-Bus.

### 8.3.9 Interrupt Request Logic

The Interrupt Request Logic consists of the following:

1. If the Data Set Interface finishes the process of transmitting or receiving data, the interface notifies the FST-2 of the completion of a sequence of events by generating an interrupt request. The interrupt request decision logic is located on Board #2. Board #1 has the actual interrupt priority logic which accepts the request from Board #2 and generates the interrupt to the CPU.

2. The interrupt request logic is triggered by any of the following events:

   a. A transmission cycle was completed;

   b. A receive cycle was completed;

   c. A data set change was detected.

3. A Lockout circuit consisting of flip-flops and interconnecting logic exists on Board #2 within the binary counter and decoding logic blocks shown on sheet 2 of Board #2 block diagram. The function of this cirucit is to determine whether the sequencer or data set change circuitry was first in generating an interrupt request. The first request holds the others in an inhibit mode until serviced. Upon acknowledgement of the request the inhibit is removed such that the second in line may then generate an interrupt request.

4. The interrupt request lockout feature along with the ability of the CPU to read the data link status register guarantees that the interrupted CPU be able to determine the event responsible for the interrupt, thus effective interrupt servicing can take place.

### 8.3.10 Sequence Controller and Status Register

8.3.10.1 SEQUENCE CONTROLLER:

The Sequence Controller and Status Register consists of the following:

1. The basic purpose of the Integrated Sequence Controller is to accept specific commands and step through a sequence of events necessary in executing the prescribed command. When the command has been executed, the sequencer generates an interrupt request.

2. The internal sequences of the controller are divided into 2 main categories.

   a. Load the transmitter buffer with data from the CPU to be transmitted then transmit the data.

   b. Monitor the RS232 interface for a receive condition then enter an auto receive mode and store the receive data and status.

3. The transmit sequence begins at sequence 0 (the idle state). Upon receiving the SPU command TRBLD/ (Transmitter Buffer Load) enters sequence 1.

4. The receive sequence also begins at sequence 0 (the idle state). The sequencer advances to sequence 7 upon determining that data set signal RLSD (Receive Line Signal Detector) has gone true.

5. An explanation of the transmit sequences are as follows:

   a. Sequence 0:

      Idle state of sequence.

   b. Sequence 1:

      TRBLD/ command to enter transmit sequence and monitor transmitter buffer load. The signal limit CTL (block diagram of Board 2 sheet 2) causes the sequencer to advance to sequence 2. As each character is loaded into the transmitter buffer, the limit counter is incremented. When the programmed limit is reached, the sequencer advances to sequence 2.

c. Sequence 2:

Sequence 2 generates a pulse back on Board #1 which resets the limit CTL counter. The sequencer monitors SQLMRST/ (Sequence Limit Reset) and advances to Sequence 3 indicating that the reset has taken place.

d. Sequence 3:

Sequence 3 increments the limit counter +1 (entering a 1 character transmit cycle) and monitors TBMT (Transmitter buffer ready) flag of the UART. It also checks the OUT (1) RDY (transmitter FIFO memory) flag and proceeds to sequence 4 when these 2 conditions are true indicating that the transmitter is ready.

d. Sequence 4:

Sequence 4 generates the transmit strobe to the UART on Board #1 and monitors the UART TRANSCOM (Transmission complete) signal. When transcom comes true the sequencer proceeds to sequence 5.

f. Sequence 5:

Sequencer 5 compares the incremented limit to the programmed character limit and proceeds to sequence 6 if the limit is reached. If the programmed limit is not reached (LIMCTL not high) the sequencer recycles Sequence 3 to Sequence 5 until the programmed character limit does come true then proceeds to Sequence 6.

g. Sequence 6:

Sequence 6 rests the limit control counter and issues transmitter done interrupt request. It should be noted that sequence 6 signal is also availabe at the LED display lgoic and the data link status buffer. A visual indication as well as CPU access is provided for identification of the event happening. The sequencer pauses at Sequence 6 until PCOMP/ from Board #1 indicates the interrupt acknowledgement then the sequencer goes back to sequence 0.

6. An explanation of the receive sequences is as follows:

a. Sequence 7:

Initially the sequence controller is at sequence 0 when RLSD (Receive Line Signal Detector) from the data set goes high the change is detected and the sequencer advances to sequence 7 to begin automatic receive mode operation.

In sequence 7 the sequencer monitors the UART on Board #1 for a DA (Data Available) signal and the receiver FIFO memory for input (2) RDY (memory ready flag). When both of these signals are true, the sequencer proceeds to sequence 8. The limit counter is also incremented indicating the first received data byte is being captured.

b. Sequence 8:

Sequence 8 generates a pulse to Board #1's UART pulse circuitry which latches the Receive Status Data. The decoding logic of the sequence monitors LSD/ from Board #1 and advance to sequence 9 when LSD/ makes a low to high transition indicating the latching has taken place.

c. Sequence 9:

Sequence 9 generates a pulse to Board #1's UART pulse circuitry which latches the Receive Data and resets the DA (Data Available) flag. The decoding logic of the sequence monitors LRD/ from Board #1 and advances to sequence 10 when LRD makes a low to high transition indicating that the command has been executed.

d. Sequence 10:

Sequence 10 monitors the limit control signal. If the limit is not true the sequence recyles from Sequence 7 to Sequence 10 until the limit counter comes true at which time the sequencer then checks for OUT 2 RDY being true (FIFO Memory Loaded) and proceeds to Sequence 11.

e. Squence 11:

Sequence 11 resets the limit counter and issues the interrupt request indicating Receive Done. The data link status register also holds the Receive Done bit along with the L.E.D. Display Logic. The sequencer remains at sequence 11 until Board #1 decodes PCOMP/ (the interrupt acknowledgement) which places it back to sequence 0.

8.3.10.2 SEQUENCE STATUS REGISTER:

The Sequence Status Register consists of the following:

1. The Sequence State Status may be interrogated by the CPU by Board #1 decoding RSEQSR/ (Read Sequence Status Register). Figure 8-5, sheet 2 of Board #2 block diagram shows that RSEQSR/ selects a tri-state buffer to gate sequence status data onto the FST-2 Input Data Bus.

2. The status register data indicates the status as follows:

   a. Bit 0 = Sequence 0 (idle state)

   b. Bit 1 = Sequence 3 (transmit sequence)

c. Bit 2 = Sequence 4 (transmit sequence)

d. Bit 3 = Sequence 5 (transmit sequence)

e. Bit 4 = Sequence 7 (receive sequence)

f. Bit 5 = Sequnece 8 (receive sequence)

g. Bit 6 = Sequence 9 (receive sequence)

h. Bit 7 = Sequence 10 (receive sequence)

3. The Data Link Status Register when placed on the line by INTF ARD/ displays the following information relating to sequencer:

   a. Bit 20 = Sequence 6 (transmit done)

   b. Bit 21 = Sequence 11 (receive done)

4. If the sequence status register and the data link status register shows not bits high the sequencer could be in Sequence 1 or 2, otherwise the sequence decoder is bad (no output).

## 8.4 DETAILED REGISTER DEFINITION

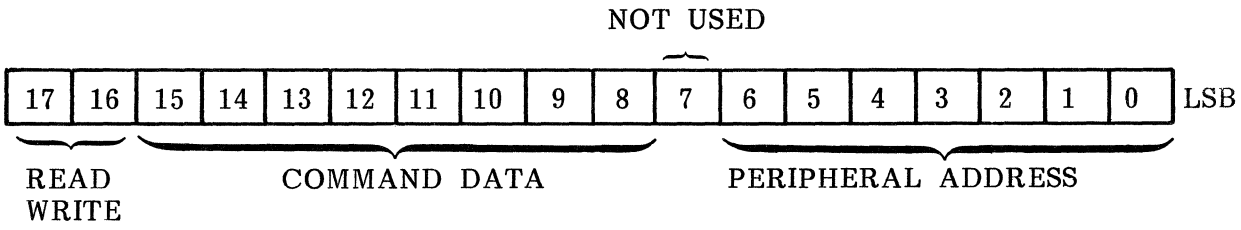
A listing of the registers used on the Data Set Interface are as follows:

1. FST-2 SPU Command Decode (write only)
2. UART Status Register (read only)
3. UART Transmitter Register (write only)
4. RS232C Status Register (read only)
5. Data Link Status Register (read only)
6. Transfer Function Register (write only)
7. UART Receive Character Register (read only)
8. Sequence Status Register (read only)

### 8.4.1 FST-2 SPU Command Decode Register

SPUs are divided into I/O data commands and interface action only commands. All SPU commands are latched at PS time, but only I/O data commands (data transfer required) are saved and activated the following T1 time.

The FST-2 Accumulator data is decoded per the following at PS (Peripheral Select) time:




As shown above bits 0-6 are compared for a Peripheral Select Address. If an address compare comes true, the command data N Bus bits 8-15 are latched along with bits 16 and 17 which indicate SPU type. Data instructions are held until the following T1 and non-Data SPUs are executed during the PS decode cycle.

A list of the SPU commands applicable to the interface is shown in Table 8-1 with corresponding N-Bus Data bits to decode.

## TABLE 8-1 SPU COMMAND TO BIT EQUIVALENT LIST

| | FST-2 N - BUS DATA BITS | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COMMAND | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PCOMP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| PON | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| POFF | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| LUXR | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| RURR | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| RUSR | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| RRSS | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| LXFR | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| IRST | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| TRBLD | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| LUSD | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |
| RSEQSR | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | X | X | X |

LSB OF ADDRESS
COULD BE 0 THRU 7

Those commands with either bit 16 or 17 true data instructions and all others are command only cycles.

Refer to Board #1's block diagram (Figure 8-4) for FST-2 OPCODE summary of the above SPU commands.

The UART Status Register has the following bit definitions:

| DB | Signal Name | Descriptions |
|---|---|---|
| 0 | D/A | Data Available |
| 1 | TBMT | Transmitter Buffer Empty |
| 2 | EOC | End of Character |
| 3 | PE | Parity Error |
| 4 | FE | Framing Error |
| 5 | ORE | Over Run Error |

The RS232C Status Register has the following bit definitions:

| DB | Signal Name | |
|---|---|---|
| 0 | CTS | Clear to Send |
| 1 | DSR | Data Set Read |
| 2 | RLSD | Receive Line Signal Detector |
| 3 | RI | Ring Indicator |
| 4 | SPARE | Spare |
| 5 | RD | Received Data |

The Data Link Status Register has the following bit defintion:

| DB | Signal Name | |
|---|---|---|
| 20 | TD | Transmitter Done |
| 21 | RD | Receiver Done |
| 22 | RERR | Receiver Error |
| 23 | DSC | Data Set Change |

The Transfer Function Register has the following bit definitions:

| DB | Signal | | |
|---|---|---|---|
| 0 | XWRD | Transfer Word; 1 = Binary & 0 = ASCII | |
| 1 | RTS | Request to Send | |
| 2 | DTR | Data Terminal Read | RS232C Functions |
| 3 | TBRK | Transmission Break | |
| 4 | TEST | Test | |
| 5 | CHAR. LIM0 | Character Limit Sel 0 | Provides |
| 6 | CHAR. LIM1 | Character Limit Sel 1 | Mux Chip |
| 7 | CHAR. LIM2 | Character Limit Sel 2 | w/1 of 7 Selection |

The UART Transmitter Register has the following definitions:

| DB | Signal |
|---|---|
| 0 | |
| 1 | |
| 2 | All Data |
| 3 | Right |
| 4 | Justified |
| 5 | Db 0 = LSB |
| 6 | |
| 7 | |

The UART Receiver Register has the following definitions:

| DB | Signal |
|---|---|
| 0 | |
| 1 | |
| 2 | All Data |
| 3 | Right |
| 4 | Justified |
| 5 | Db0 = LSB |
| 6 | |
| 7 | |

The Sequence Status Register has the following definitions:

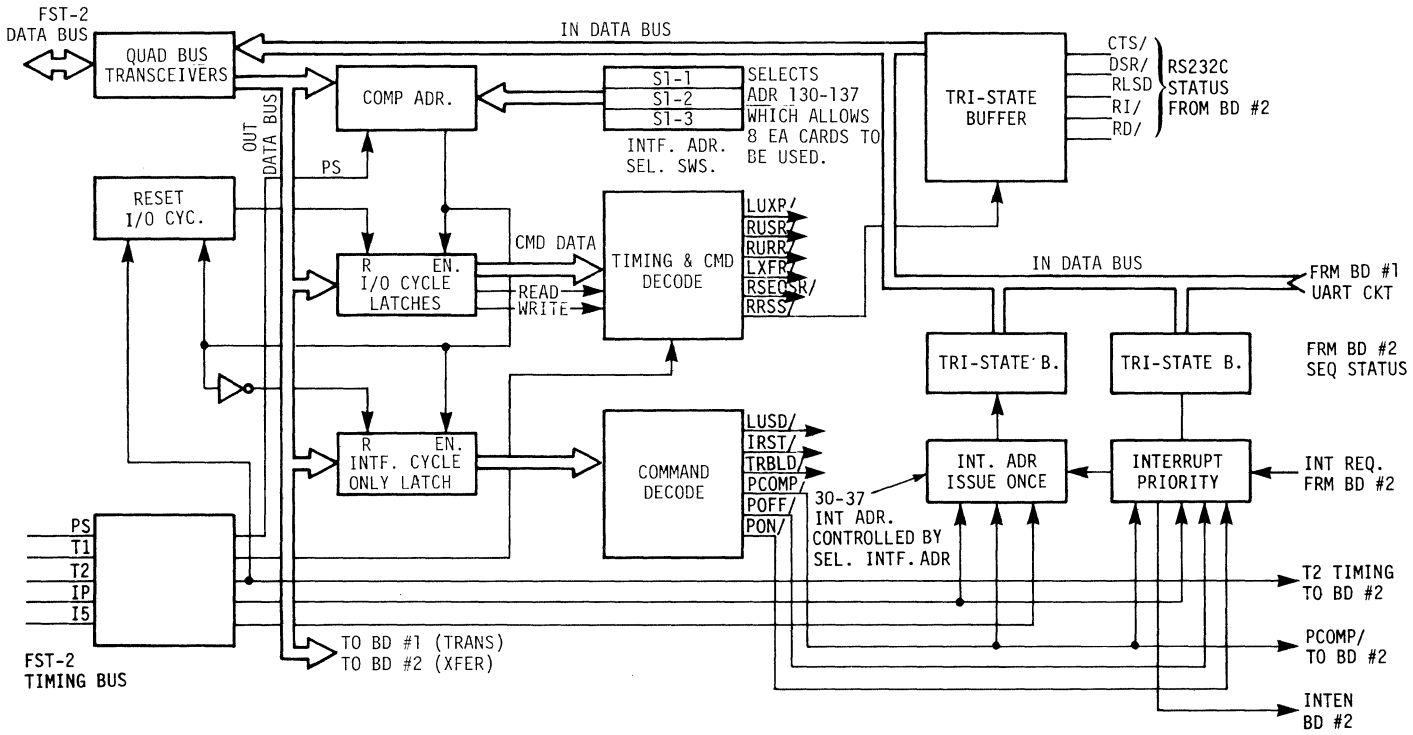
| DB | Signal | | |
|---|---|---|---|
| 0 | Seq 0 | Sequence State 0 | (idle state) |
| 1 | Seq 3 | Sequence State 3 | |
| 2 | Seq 4 | Sequence State 4 | Transmission |
| 3 | Seq 5 | Sequence State 5 | |
| 4 | Seq 7 | Sequence State 7 | |
| 5 | Seq 8 | Sequence State 8 | |
| 6 | Seq 9 | Sequence State 9 | Receive |
| 7 | Seq 10 | Sequence State 10 | |

**NOTE**

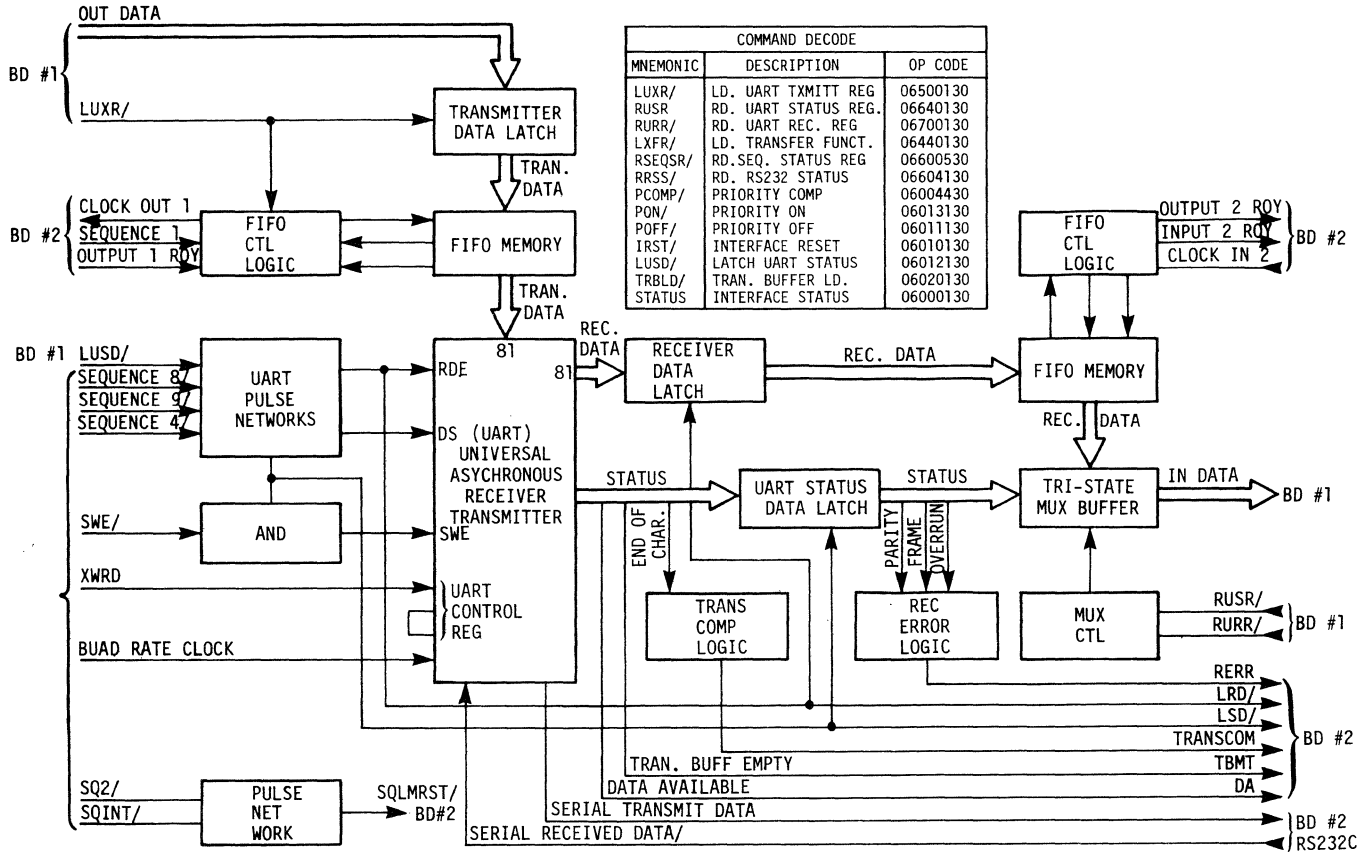
As previously mentioned in the Board #2 description, if Data Link Status is read, 2 bits of that register also contain the following sequencer status information.

| DB | Signal | |
|---|---|---|
| 20 | TD | Transmitter Done = Sequence State 6 |
| 21 | RD | Receiver Done = Sequence State11 |

## Board #1, Sheet 1




## Board #1, Sheet 2

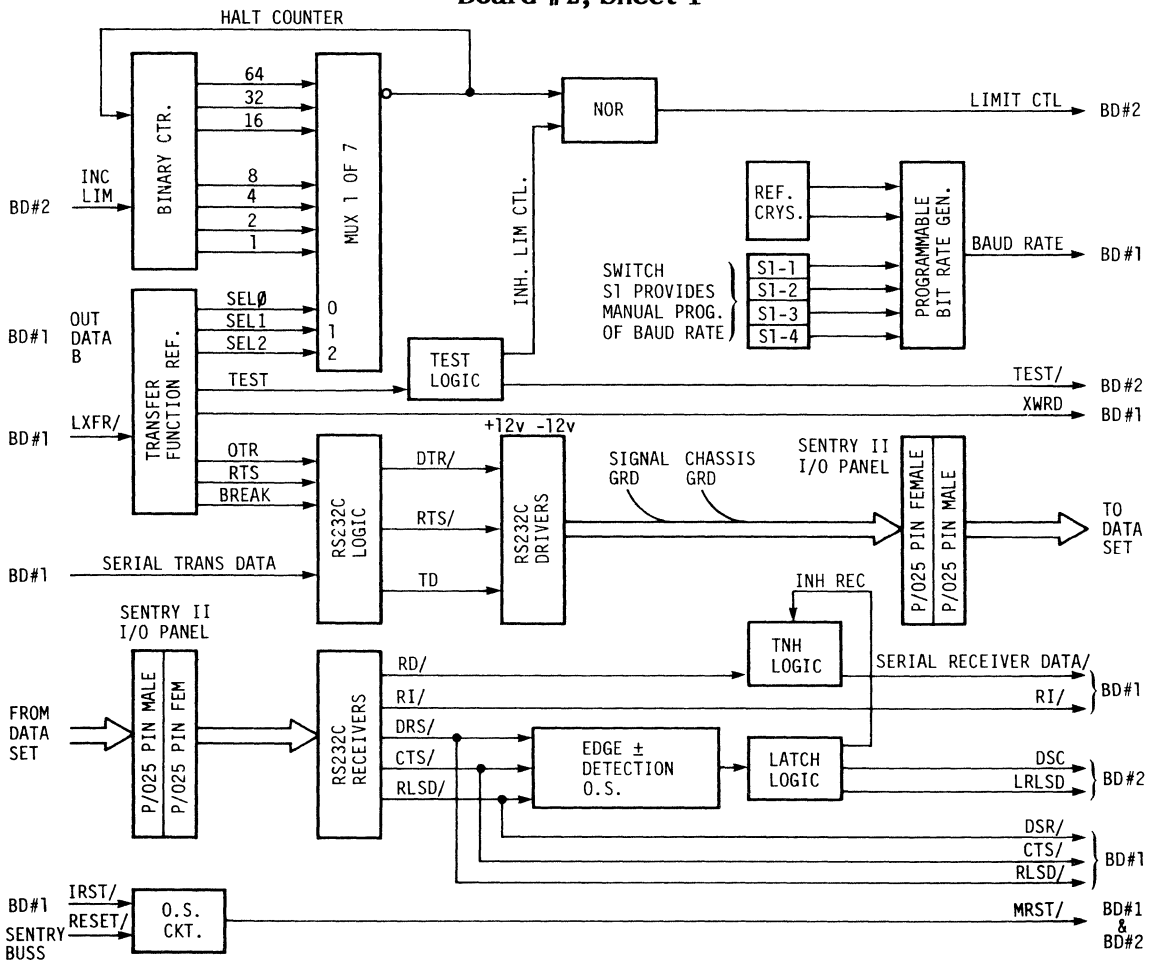



| COMMAND DECODE | | |
|---|---|---|
| MNEMONIC | DESCRIPTION | OP CODE |
| LUXR/ | LD. UART TXMITT REG | 06500130 |
| RUSR | RD. UART STATUS REG. | 06640130 |
| RURR/ | RD. UART REC. REG | 06700130 |
| LXFR/ | LD. TRANSFER FUNCT. | 06440130 |
| RSEQSR/ | RD.SEQ. STATUS REG | 06600530 |
| RRSS/ | RD. RS232 STATUS | 06604130 |
| PCOMP/ | PRIORITY COMP | 06004430 |
| PON/ | PRIORITY ON | 06013130 |
| POFF/ | PRIORITY OFF | 06011130 |
| IRST/ | INTERFACE RESET | 06010130 |
| LUSD/ | LATCH UART STATUS | 06012130 |
| TRBLD/ | TRAN. BUFFER LD. | 06020130 |
| STATUS | INTERFACE STATUS | 06000130 |

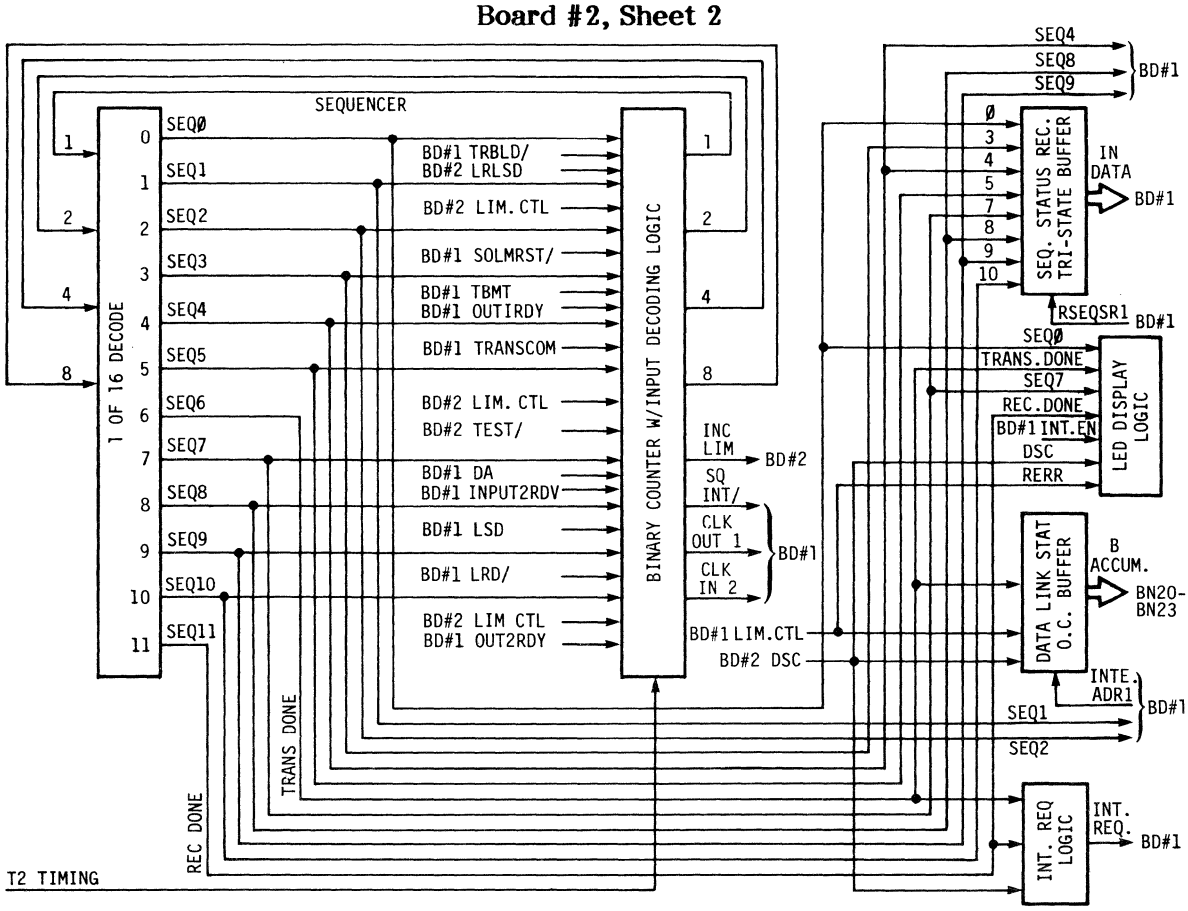
L121

**Figure 8-4 Block Diagram Data Set Interface Board #1**

## Board #2, Sheet 1




## Board #2, Sheet 2



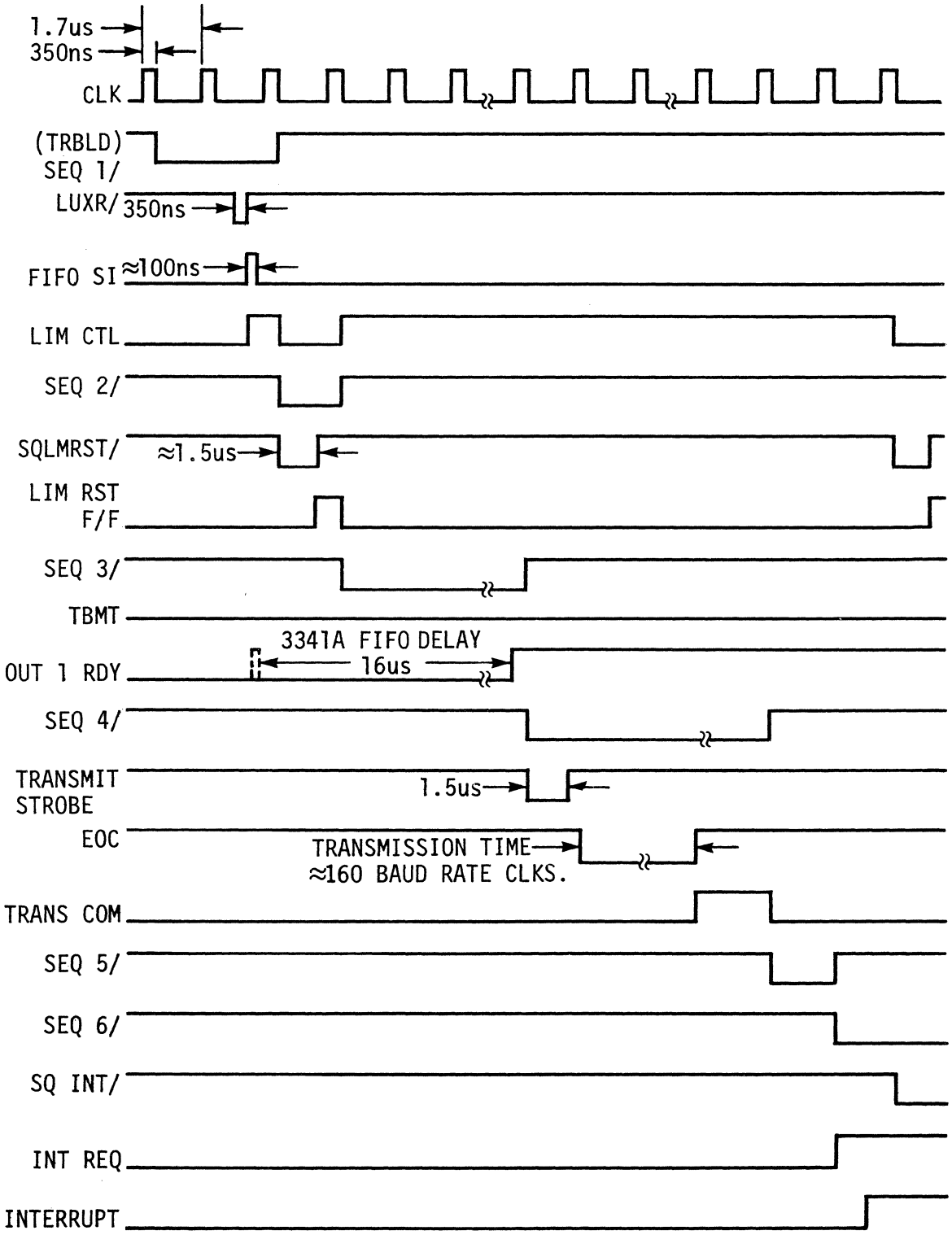

L122

8-24,25

**Figure 8-5 Block Diagram Data Set Interface Board #2**

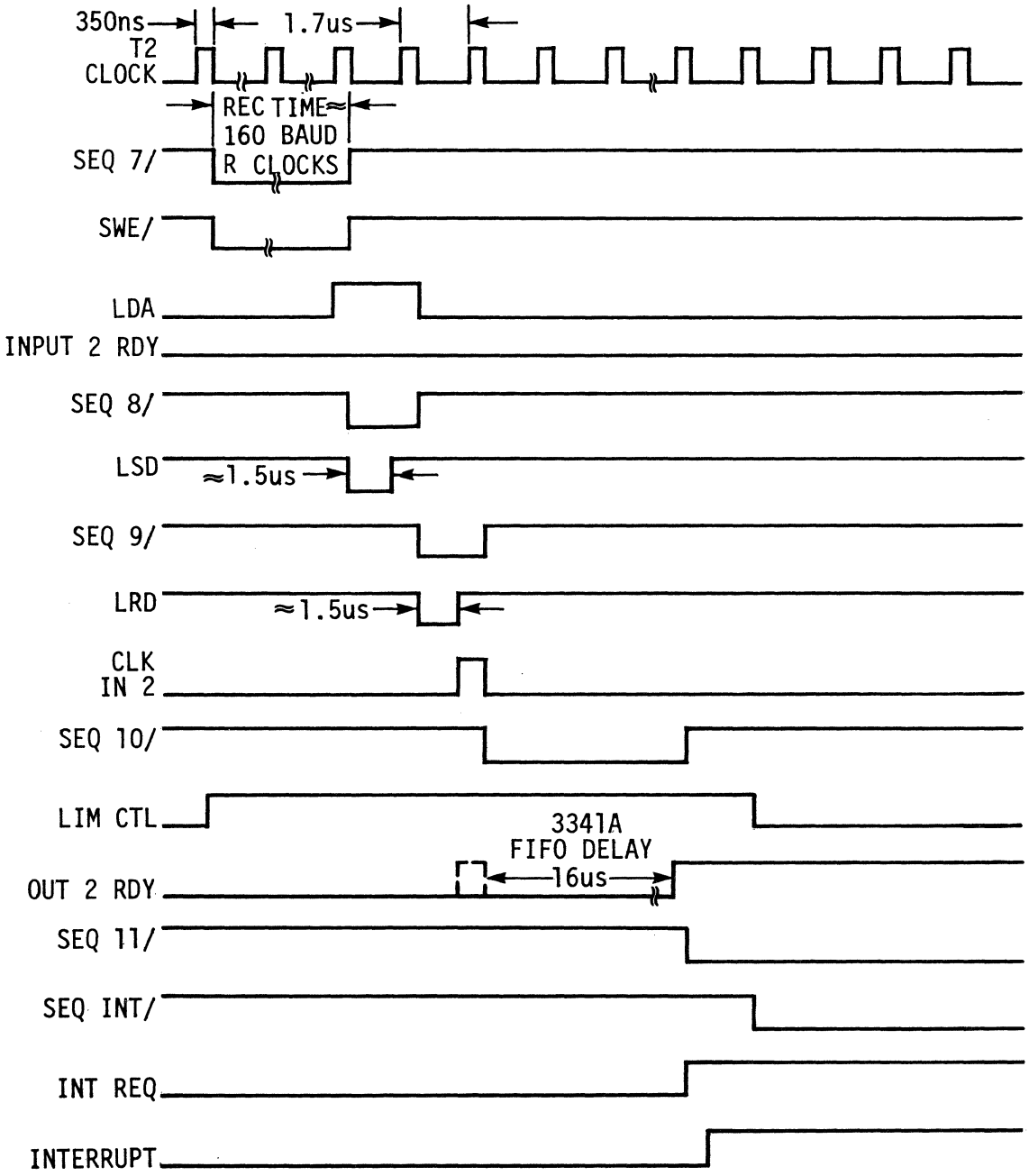
## APPENDIX A

## TYPICAL TRANSMISSION CYCLE DATA SET INTERFACE

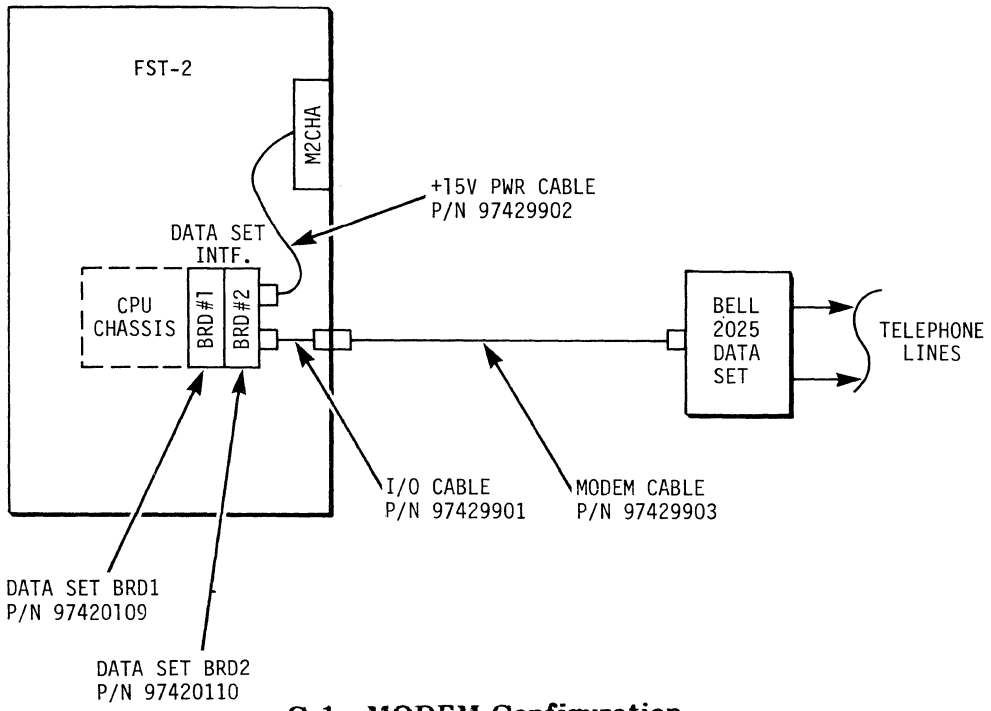


## APPENDIX B

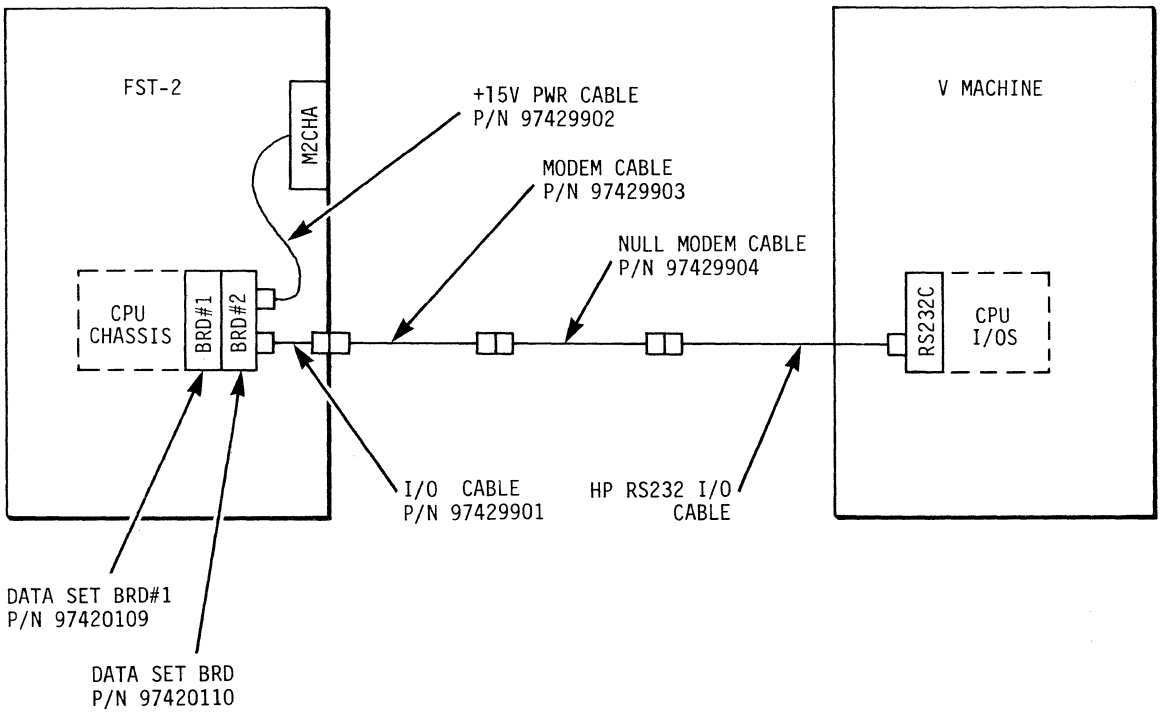
## TYPICAL RECEIVE CYCLE DATA SET INTERFACE



# APPENDIX C
# DATA SET INTERFACE CONFIGURATION




**C-1 MODEM Configuration**




**C-2 Null MODEM Configuration**

# GLOSSARY

**Baud Rate:** Baud Rate is normally accepted to be the bit rate of serial digital data transfer using an RS232C interface. In the case of the Data Set Interface it is a frequency sixteen times the actual bit rate since this is what is required by the UART device.

**Data Set:** A modem or null modem used to establish a data link between the FST-2's Data Terminal and a remote processor.

**Data Terminal:** The interface or hardware used to interface the FST-2 to a data set.

**Modem:** A modulator/demodulator used in converting logic levels onto carrier frequencies such that serial digital data may be transmitted via common carrier facilities.

**Null Modem:** A cable connection scheme used to accomplish a 'hard-wired' connection between the FST-2 data set and the remote computer's data set.

**RS232C:** A means of sending and receiving serial digital data over an interface approved by EIA as a standard for data communcations.

**UART:** A Universal Asynchronous Receiver Transmitter which converts parallel input data to serial output data for the transmitter section. It also converts serial input data to parallel output for the receiver section. Data transmitted has start, stop and parity bits added. Received data is checked for parity, framing or overrun errors.

# INDEX