27114A EXTERNAL REFERENCE SPECIFICATION

A-27114-90011-1 Rev A

CONTENTS

UPDATE HISTORY

Dec 12, 1985
* The correct definition for backplane interface registers is given (in term of BP0, BP1, CBYT and CEND).
* Frontplane pin A10 now is a safety ground pin and pin B10 is the test hood presence pin.

Feb 12, 1986
* Cable offering (single-ended: high and low true)/differential.
* General clean up of documentation

Apr 9, 1986
* Extra information on group 0 polling
* Information about series 550 is removed
* More information on grounding and RFI performance

June 26, 1986
* General clean up

Nov 7, 1986
* General clean up
* Clarify data handshake

Jan 14, 1986
* General clean up

1. SCOPE

This External Reference Specification provides all the information required to install and operate the AFI (Asynchronous FIFO Interface) card. It details the pin out and function of the frontplane interface as well as the backplane interface in terms of register descriptions.

The intention of this document is to provide sufficient information such that drivers can be written for high level language users. This document does not provide full information on how the card operates internally or how programming is done at the user level. The 27114 IRS provides information on how the card operates internally. For user programming information, please refer to the appropriate driver's ERS.

Eventhough any well written ERS could provide enough information to guide how to use the described product, it is believed that the AFI card belongs to an exception group. This ERS alone probably would be sufficient to provide enough information for most intended use but wise users should also be intimately familiar with the hardware via the IRS and specially, the schematics, the granddaddy of all electrical documents.

It should be noted here that this is not an ERS for the AFI product but only an ERS for the AFI hardware. Attempt to interpret this otherwise could mislead to other confusion. A good example is the parity feature of the AFI card. While there are hardware circuits available on the AFI assembly to support the parity feature, the host channel adapter and device adapter driver/operating systems must also support it to claim that AFI product has the parity feature. The ERS for AFI product should be written after close examination of ERS's for both hardware and software produtcts to see how and what kind of interactions could be expected.

As of this release, most of the performance testing has been done to verify that AFI actually could go as high as 5 Mbytes per sec for cables of length 3 to 12 m long. This figure may also be misleading if not carefully interpretted by the product ERS writer since other system factors could greatly affect it.

Throughout this document, temporary NOTEs will be used to describe unresolved issues as of the latest revision date. Please consult RND' R&D for latest information on those subjects.

2. GENERAL

2.1 Description

The 27114 card is an extended level 1 CIO card intended for fast parallel data transfer between CIO channel and an external peripheral. It interfaces to user's equipment via a 16 data bit parallel bus. Two additional parity bits are also provided for suitable future application.

NOTE:

The 8 bit mode can be achieved by having the byte packing/unpacking done by the CIO channel adapter.

This bus can be configured either as differential or single-ended (separate in/out only). Pulsed asynchronous handshake is provided. The 27114A card can be used in any Spectrum host with CIO channel adapter.

The 27114 is used as a dedicated interface to a single peripheral device. It is not and cannot be made to be compatible to either HPIB, SCSI or IPI. The 27114 is designed to be compatible to most other HP's parallel cards.

2.2 Environment

POWER CONSUMPTION:

The 27114A card consumes about 2 A at +5 volts. The 2 sigma value is about 2.25 A.

FUSE:

The 27114A is shipped with a built-in fuse printed circuit trace. Under abnormal condition, this trace will burn out to protect the main power supply. After the abnormal cause is removed, part number 2110-0712 fuse (4 A) should be used to replace the built-in fuse. This part is available directly from CPC.

OPERATING ENVIRONMENT:

The 27114 is intended to be operated over the Hewlett-Packard CLASS B environmental specifications.

2.3 Installation

The 27114A card is a regular mode card as far as power consumption is concerned. It draws typically about 10 watts compared to the normal 12 watt allowance for regular CIO cards. The installation of this card in any CIO card cage does not require budgeting of power supply resource.

Since the AFI responds to polling via its UAD line and its channel address is the same as the slot address, it can be guaranteed to function correctly in group 0 slots only. Note that other level 2 CIO cards have a programmable (via address teaching) channel address register which would pull on the appropriate data line (one of D0 through D7) when they respond to any poll, regardless what address (0 through 15) they are at (group 1 cards belong to address 8 to 15 but also responds to poll via D0 to D7).

Please refer to the appropriate channel adapter manual for further information. Section 4.2.3 gives some extra information on how group 0 polling is necessary for the AFI card.

The configuration switches (SW1 switch 1 through 8) must be set as follow:

SWITCH SW1:

Switches 1 through 6:

For standard cable length (3/12 meter differential, and 3 meter single-ended), set switches SW1 #1,#3,#5 to the ON position. SW1 #2,#4,#6 are don't cares.

The 25 m cable length is not recommended for single-ended cables due to excessive resistance of the common return current ground lines.

SW1 switches 7 and 8:

These two switches are used to program the parity bit in the ID (read) register (bit #11). The bit is read as 1 when switch 7 is closed and switch 8 is open. If switch 7 is open and switch 8 is closed, a 0 will be read. Note that the generation and checking of parity is not affected by the setting of these two switches.

ALTERNATE CABLE LENGTH RESISTOR VALUES:

When the 24 m length is selected, a different custom length cable can be used when resistors R5, R7 and R9 are installed with the desired values as recommended in section 3.6

LINE CONFIGURATION:

As shipped from the factory, all of the frontplane signals are differential and can be re-configured as single-ended (high true logic) as follow:

* Remove resistor network packs RP26, RP56 and RP406 from their DIP sockets. These (part # 1810-0338) should be saved if future re-conversion is anticipated.

* Install part number 1810-0906's in these sockets such that their pin 1 corresponds to pin 10 and 20 of each of the U26, U56 and U406 mentioned above (two -0906's per DIP socket).

* For low true single ended line configuration, install these 1810-0906 such that their pin 1 corresponds to pin 1 and 11 of the sockets U26, U56 and U406

* To convert back to differential lines, install the 1810-0338 resistor network DIPs in sockets U26, U56 and U406 such that their pin 1 corresponds to pin 2 of the sockets.

6 part number 1810-0906's are supplied with each 27114 card with single-ended cable option.

CABLE INSTALLATION:

Each 27114A AFI card is shipped with either a differential cable (3 or 12 m long) or a single-ended cable (3 m long) depending on the option selected. The differential cable is ambidextrous and can be installed with either end plugged in to the 27114A card. The single-ended cable is labeled on both connectors. The connectors are labeled "High-true Periph. End" and "Low-true Periph. End". With the "High-true Periph. End" connected to the peripheral, the cable conforms to high-true application and vice-versa for the "Low-true Periph. End" connector.

It is recommended that the user effectively ground the cable at the peripheral end via a reliable, low resistance, broad surface connection between the exposed metal part ("Rook") and the nearest chassis. THE RADIO FREQUENCY INTERFERENCE CHARACTERISTIC OF THE TOTAL SYSTEM CAN BE SEVERELY DEGRADED IF THIS CONNECTION IS NOT MADE PROPERLY AND COULD LEAD TO POSSIBLE VIOLATIONS OF MOST LOCAL LAWS AND REGULATIONS. THE IMPORTANCE OF THIS CANNOT BE OVERSTATED HERE.

3. FRONTPLANE FUNCTIONAL DESCRIPTION

The frontplane is used to pass data back and forth between the IO card and the user's equipment. The frontplane is organized around 16 bits of data plus two parity bits (one for each byte). Status and control lines are also used to pass information (at a slower rate). An attention line can be used to interrupt the host processor for priority service.

3.1 Physical Characteristics of Frontplane Signals

3.1.1 Cable - differential version

The cable has 48 twisted pairs of 32 gauge wire wrapped in a combination of foil/braid shield and jacketed in standard HP jade gray PVC. The connectors on both 27114 ends of the cable are DIN 41650 type C headers (Eurocard connector).

Both ends behave in the same manner and can be installed either way. The wiring is made such that if the cable is used to connect two AFI cards, the positive output of a signal on one card is connected to the positive input of a matching signal on the other card (e.g. PCTL+ to PFLG+). This arrangement allows the same cable to be used for AFI to AFI connection.

RF shielding is done via the Rook to slot contact at the point the cable exits the CIO card cage and also via the clipping mechanism. Safety grounding is also provided via the rook to slot contact.

The cable assembly is mated to a male (pin) 96-position Eurocard connector on the 27114A card. A 1/2" wire-wrap connector (DIN header type R) is supplied as the matching connector at the user end. Only two lengths are provided: 3 m and 12 m. Custom made cables for other length can be ordered directly from the OEM supplier, DuPont Connector Systems. Its impedance is about 120 ohms. The resistance per foot is about 60 milliohms. Propagation delay in the cable is about 5 nanoseconds per meter.

3.1.2 Cable (single-ended version):

The single-ended cable is very similar to the differential cable. There are some differences however:

* There are only 50 conductors. Two of which are for ground return.

* If this cable is used to connect two AFI cards together, it will map the output of a signal to the opposite sense input of a matching signal on the other card. Connected as thus would complicate the use of this cable as AFI to AFI connection but provide a side benefit: only one cable is needed for both high and low true applications.

The cable is wired such that if installed in one direction, it will provide high true single-ended interface and low true

interface in the other direction. The cable is labeled on both connectors as stated in section 2.3.

Length longer than 3 m is not recommended unless the data rate is reduced substantially to maintain data integrity. Even then, the excessive length can cause the signal to be more susceptible to noise or become unreliable.

NOTE:

Unless otherwise specified, all performance figures in this document are applicable only in the differential mode with the standard 3 m cable and with an zero delay user's interface (instantaneous PFLG timing with respect to PCTL).

The single-ended cable shares other common characteristics with the differential cables. Same technique of grounding and cable termination/ connectors are provided (mating wire-wrap connector included).

3.1.3 Drivers and Receivers

Drivers: MC3487 differential drivers which complies to RS-422 specs. The HP part number is 1820-2145.

Receivers: 26LS32B differential receivers. This is a higher speed/less skew/ more hysteresis 26LS32. These parts are designed to "receive" a "one" if left open. In differential mode, since there is a resistor across each input pair, the outputs cannot be guaranteed of any certain value if left disconnected. The HP part number is 1820-2830.

To configure the frontplane interface as single-ended lines, refer to section 2.3 (installation).

Even though these line drivers and receivers were designed to withstand the common mode voltage which may appear between the 27114A card and the user's equipment as specified per RS-422 specs, the card was not designed to interface to +12 volt open collector devices.

3.1.4 Bi-directional bus

There is no current plan to supply a bi-directional cable for the 27114 card. A possibility exists whereas the user can shorten corresponding inputs and outputs at the peripheral end to form a makeshift cable. Line reflections and impedance mismatch may deteriorate performance and thus this solution is not recommended unless data transfer rate is reduced substantially and experiments are done by the user to assure data integrity.

Notes on bi-directional bus contention: The users should take every precaution in determining which end is driving the bi-directional bus at any one point in time. If both ends are in the same talking/listening state, unsatisfactory results are is inevitable. The line drivers provided on the 27114 card are

electrically protected against this abuse as stated by the RS-422 standard but abuses should be avoided if possible.

3.2 Signal Definitions

The following description is for the differential cable mode. It would also generally apply to the single-ended cable mode. Wherever differences occur, these will be pointed out. Note that the single-ended cable has only one conductor per signal and all signals share two common ground return lines.

In the differential cable, each pair of signal includes two conductors wound in a twisted manner to minimized induced noise and to keep line impedance within tolerance. Since each conductor in a pair swings in the opposite direction of the other one, they tend to cancel each other as far as the return current is concerned. This effectively makes the line less susceptible to length and induced noise.

Output Data: 16 twisted pairs. Each of these should be terminated with around 120 ohms at the user's end. For single ended lines, it is recommended that these are terminated with a 3.3 K ohm resistor connected to +5 volt source.

Input Data: 16 twisted pairs. Each of these is terminated on the 27114 with 120 ohm terminator (for both separate and common in-out bus configuration). When configured as single-ended lines, these inputs are terminated by a voltage divided connected between +5 and ground consisting of a 1500 ohm resistor (to +5) and a 3300 ohm resistor (to ground). The other line is biased internally on the card to a voltage around 1.5 volts.

PCNTL: (sometime referred to as PCTL) It is asserted in a master relationship to the peripheral to indicate that either data is available or data can now be accepted. Set up time of data assertion prior to assertion of PCNTL is controlled by an RC time constant on board. This time is nominally 75 nsec. Recommended termination is 120 ohms at receiver. The asserted state is that of the pin labeled PCNTL+ at a high voltage with respect to PCNTL-. For high true single-ended lines, the PCNTL+ line is used.

This line can be preset or reset by careful manipulation of data output sequence to the card. Its state is available as a status bit in the 27114 status register. These manipulations are not accessible directly to the user above the driver's level.

PFLAG: It is asserted by the peripheral to acknowledge the acceptance of data or to indicate that valid data exists on the lines. Data must be valid at the same time as PFLAG is asserted on transfer from peripheral to 27114. Received by 26LS32B with 120 ohms across line. De-skewing of data received by the 27114 is done based on an RC time constant on the board. This is nominally about 75nsec. The asserted state is of the same sense as PCNTL.

The state of this line can be read as a bit in the 27114A status register. However, this is not visible to the user above the driver's level.

PDIR: It is asserted by the 27114 indicating the direction of transfer. Transfer is to peripheral in case of PDIR = 1 (output/write). This state is indicated with PDIR+ high with respect to PDIR-.

This line is asserted during output mode. The actual timing is driver's dependent but should not be changing between consecutive outputs or inputs.

CONTROL[2..0]: Three bits under program control. They are simply bits in a register on the card. 27114 asserts these and the peripheral watches them. They are set to all zero's in case of a clear of the card. The "zero" state is that of the pin labeled CTL?+ at a low voltage with respect to CTL?-. Program logic sense is the opposite of that seen on cable.

These lines can be reset or preset independently of each other and of other activities on the card. They should be maintained at the commanded levels (by user) regardless of other data input/output activities.

STATUS[2..0]: Three bits which are readable by a program as register bits. The STATUS[1..0] bits are updated with every /SYNC rising edge from the CIO backplane. The STATUS 2 line receiver's output is enabled only when DIR is false (data input); the corresponding status bit is valid only in this condition. Another way to look at this restriction is to consider that status 2 does not exist.

Received by 26LS32B terminated with 120 ohms. The sense is such that if STS?+ is at a high voltage with respect to STS?- the value read from the 27114 by a program will be a zero.

These lines are received and passed directly to the user and do not affect other activities on the 27114 card.

ATTN: Asynchronous interrupt line. When switched, it causes sense[arq] to be set. If software has enabled the card to pass ATTN on through as an ARQ interrupt, it will cause an interrupt to the channel adapter. If the interrupt in the channel adapter is enabled, it will cause an interrupt to a CPU. Received by 26LS32B and 120 ohms. This is an edge-triggering signal which activates only when the ATTN+ signal is changing from high to low level while ATTN- is changing in the other way around.

The interrupt action is sensed by CIO channel adapter and passed on to the host accordingly.

TEST: This is a special testing pin. When grounded, the data line drivers are enabled at all time, allowing the loopback of data with the test hood during diagnostic tests. The supplied

testhood grounds this input. Unless necessary, user should leave this pin untouched.

This line can also be detected as a bit in the 27114 status register. The presence of a test hood would ground this pin, resulting in a reading of 1. The status of this line is inaccessible to most user above driver's level.

Grounds: Two signal ground wires in the cable (two pairs). A drain wire is also provided. This wire is connected to the frame ground on the card.

3.3 pin out

Four wiring tables are presented here. The first one is for the frontplane connector. The second cable is for the differential wire-wrap connector. The third and fourth tables are for single-ended wire-wrap connector pin outs.

The first one follows immediately as three tables. These three diagrams illustrate the assignment of pins versus signals for rows A, B and C of the frontplane connector.

Perhaps of more importance is the wire-wrap connector pin out diagram which follows the frontplane connector pin out diagram. As the 27114 cable is defined, most signal pairs are swapped in the cable, resulting in a slightly different wiring diagram.

A quick observation will reveal that the wire-wrap connector wiring diagram is an inverted image of that of the frontplane interface connector. This is very useful when one attempts to interconnect two 27114A cards to each other. In this case, only one cable assembly is required.

FRONTPLANE CONNECTOR PIN OUT DIAGRAM - ROW A

27114 CONN. OPIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
A1	RD0-		Data Bit 0, Low True, RECEIVE
A2	RPAR0-		Parity bit 0, Low True, RECEIVE
A3	RD1-		Data Bit 1, Low True, RECEIVE
A4	SPAR0+		Parity bit 0, High True, SEND
A5	RD2-		Data Bit 2, Low True, RECEIVE
A6	RPAR1+		Parity bit 1, High True, RECEIV
A7	RD3		Data Bit 3, Low True, RECEIVE
A8	SPAR1-		Parity bit 1, Low True, SEND
A9	RD4-		Data Bit 4, Low True, RECEIVE
A10	SHIELD		Frame ground
A11	RD5-		Data Bit 5, Low True, RECEIVE
A12	STS2+		STATUS 2 , High True, RECEIVE
A13	RD6-		Data Bit 6, Low True, RECEIVE
A14	PFLG-		PFLAG , Low True, RECEIVE
A15	RD7-		Data Bit 7, Low True, RECEIVE
A16	STS0+		STATUS 0 , High True, RECEIV
A17	RD8+		Data Bit 8, High True, RECEIVE
A18	STS1-		STATUS 1 , Low True, RECEIVE
A19	RD9+		Data Bit 9, High True, RECEIVE
A20	GND		Signal Ground
A21	SD9-		Data Bit 9, Low True, SEND
A22	PCTL-		PCTL , Low True, SEND
A23	RD11+		Data Bit 11, High True, RECEIVE
A24	CTL0+		CTRL 0 , High True, SEND
A25	RD12+		Data Bit 12, High True, RECEIVE
A26	CTL1+		CTRL 1 , High True, SEND
A27	RD13+		Data Bit 13, High True, RECEIVE
A28	CTL2-		CTRL 2 , Low True, SEND
A29	RD14+		Data Bit 14, High True, RECEIVE
A30	ATTN+		ATTN , High True, RECEIV
A31	RD15+		Data Bit 15, High True, RECEIVE
A32	PDIR- *		PDIR , Low True, SEND

WIRECOLOR (LISTED WITH COLOR)

0 - BK (BLACK)
1 - BR (BROWN)
2 - R (RED)
3 - O (ORANGE)
4 - Y (YELLOW)
5 - G (GREEN)
6 - B (BLUE)
7 - GR (GRAY)
8 - V (VIOLET)
9 - W (WHITE)
X - BG (BEIGE)

OW = OFF WHITE

FRONTPLANE CONNECTOR PIN OUT DIAGRAM - ROW B

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
B1	SD0-		Data Bit 0, Low True, SEND
B2	RPAR0+		Parity bit 0, High True, RECEIV
B3	SD1-		Data Bit 1, Low True, SEND
B4	SPAR0-		Parity bit 0, Low True, SEND
B5	SD2+		Data Bit 2, High True, SEND
B6	RPAR1-		Parity bit 1, Low True, RECEIVE
B7	SD3-		Data Bit 3, Low True, SEND
B8	SPAR1+		Parity bit 1, High True, SEND
B9	SD4+		Data Bit 4, High True, SEND
B10	TEST-		Data line driver enable
B11	SD5-		Data Bit 5, Low True, SEND
B12	STS2-		STATUS 2 , Low True, RECEIVE
B13	SD6+		Data Bit 6, High True, SEND
B14	PFLG+		PFLAG , High True, RECEIV
B15	SD7-		Data Bit 7, Low True, SEND
B16	STS0-		STATUS 0 , Low True, RECEIVE
B17	SD8+		Data Bit 8, High True, SEND
B18	STS1+		STATUS 1 , High True, RECEIV
B19	RD10-		Data Bit 10, Low True, RECEIVE
B20	GND		Signal Ground
B21	SD10+		Data Bit 10, High True, SEND
B22	PCTL+		PCTL , High True, SEND
B23	SD11+		Data Bit 11, High True, SEND
B24	CTL0-		CTRL 0 , Low True, SEND
B25	SD12+		Data Bit 12, High True, SEND
B26	CTL1-		CTRL 1 , Low True, SEND
B27	SD13-		Data Bit 13, Low True, SEND
B28	CTL2+		CTRL 2 , High True, SEND
B29	SD14+		Data Bit 14, High True, SEND
B30	ATTN-		ATTN , Low True, RECEIVE
B31	SD15-		Data Bit 15, Low True, SEND
B32	PDIR+		PDIR , High True, SEND

FRONTPLANE CONNECTOR PIN OUT DIAGRAM - ROW C

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
C1	RD0+		Data Bit 0, High True, RECEIVE
C2	SD0+		Data Bit 0, High True, SEND
C3	RD1+		Data Bit 1, High True, RECEIVE
C4	SD1+		Data Bit 1, High True, SEND
C5	RD2+		Data Bit 2, High True, RECEIVE
C6	SD2-		Data Bit 2, Low True, SEND
C7	RD3+		Data Bit 3, High True, RECEIVE
C8	SD3+		Data Bit 3, High True, SEND
C9	RD4+		Data Bit 4, High True, RECEIVE
C10	SD4-		Data Bit 4, Low True, SEND
C11	RD5+		Data Bit 5, High True, RECEIVE
C12	SD5+		Data Bit 5, High True, SEND
C13	RD6+		Data Bit 6, High True, RECEIVE
C14	SD6-		Data Bit 6, Low True, SEND
C15	RD7+		Data Bit 7, High True, RECEIVE
C16	SD7+		Data Bit 7, High True, SEND
C17	RD8-		Data Bit 8, Low True, RECEIVE
C18	SD8-		Data Bit 8, Low True, SEND
C19	RD9-		Data Bit 9, Low True, RECEIVE
C20	RD10+		Data Bit 10, High True, RECEIVE
C21	SD9+		Data Bit 9, High True, SEND
C22	SD10-		Data Bit 10, Low True, SEND
C23	RD11-		Data Bit 11, Low True, RECEIVE
C24	SD11-		Data Bit 11, Low True, SEND
C25	RD12-		Data Bit 12, Low True, RECEIVE
C26	SD12-		Data Bit 12, Low True, SEND
C27	RD13-		Data Bit 13, Low True, RECEIVE
C28	SD13+		Data Bit 13, High True, SEND
C29	RD14-		Data Bit 14, Low True, RECEIVE
C30	SD14-		Data Bit 14, Low True, SEND
C31	RD15-		Data Bit 15, Low True, RECEIVE
C32	SD15+		Data Bit 15, High True, SEND

WIRE-WRAP CONNECTOR PIN OUT DIAGRAM - DIFFERENTIAL, ROW A

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
A1	SD0-		Data Bit 0, Low True, SEND
A2	SPAR0-		Parity bit 0, Low True, SEND
A3	SD1-		Data Bit 1, Low True, SEND
A4	RPAP0+		Parity bit 0,High True,RECEIVE
A5	SD2-		Data Bit 2, Low True, SEND
A6	SPAR1+		Parity bit 1, High True, SEND
A7	SD3-		Data Bit 3, Low True, SEND
A8	RPAR1-		Parity bit 1,Low True,RECEIVE
A9	SD4-		Data Bit 4, Low True, SEND
A10	SHIELD		Frame ground
A11	SD5-		Data Bit 5, Low True, SEND
A12	PDIR+		DIRECTION , High True,SEND
A13	SD6-		Data Bit 6, Low True, SEND
A14	PCTL-		PCTL , Low True, SEND
A15	SD7-		Data Bit 7, Low True, SEND
A16	CTL0+		CONTROL 0 , High True, SEND
A17	SD8+		Data Bit 8, High True, SEND
A18	CTL1-		CONTROL 1 , Low True, SEND
A19	SD9+		Data Bit 9, High True,SEND
A20	GND		Signal Ground
A21	RD9-		Data Bit 9, Low True,RECEIVE
A22	PFLG-		PFLAG , Low True,RECEIVE
A23	SD11+		Data Bit 11, High True,SEND
A24	STS0+		STATUS 0 , High True,RECEIVE
A25	SD12+		Data Bit 12, High True, SEND
A26	STS1+		STATUS 1 , High True,RECEIVE
A27	SD13+		Data Bit 13, High True, SEND
A28	ATTN-		ATTENTION , Low True,RECEIVE
A29	SD14+		Data Bit 14, High True,SEND
A30	CTL2+		CONTROL 2 , High True, SEND
A31	SD15+		Data Bit 15, High True,SEND
A32	STS2-		STATUS 2 , Low True,RECEIVE

COLOR ("TWISTED WITH" COLOR)

0 - BK BLACK
1 - BR BROWN
2 - R RED
3 - O ORANGE
4 - Y YELLOW
5 - G GREEN
6 - B BLUE
7 - V VIOLET
8 - GR GRAY
9 - WH WHITE
X - BG
OW

WIRE-WRAP CONNECTOR PIN OUT DIAGRAM - DIFFERENTIAL, ROW B

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
B1	RD0-		Data Bit 0, Low True, RECEIVE
B2	SPAR0+		Parity bit 0, High True, SEND
B3	RD1-		Data Bit 1, Low True, RECEIVE
B4	RPAR0-		Parity bit 0, Low True, RECEIVE
B5	RD2+		Data Bit 2, High True, RECEIVE
B6	SPAR1-		Parity bit 1, Low True, SEND
B7	RD3-		Data Bit 3, Low True, RECEIVE
B8	RPAR1+		Parity bit 1, High True, RECEIVE
B9	RD4+		Data Bit 4, High True, RECEIVE
B10			
B11	RD5-		Data Bit 5, Low True, RECEIVE
B12	PDIR-		DIRECTION , Low True, SEND
B13	RD6+		Data Bit 6, High True, RECEIVE
B14	PCTL+		PCTL , High True, SEND
B15	RD7-		Data Bit 7, Low True, RECEIVE
B16	CTL0-		CONTROL 0 , Low True, SEND
B17	RD8+		Data Bit 8, High True, RECEIVE
B18	CTL1+		CONTROL 1 , High True, SEND
B19	SD10-		Data Bit 10, Low True, SEND
B20	GND		Signal Ground
B21	RD10+		Data Bit 10, High True, RECEIVE
B22	PFGL+		PFLG , High True, RECEIVE
B23	RD11+		Data Bit 11, High True, RECEIVE
B24	STS0-		STATUS 0 , Low True, RECEIVE
B25	RD12+		Data Bit 12, High True, RECEIVE
B26	STS1-		STATUS 1 , Low True, RECEIVE
B27	RD13-		Data Bit 13, Low True, RECEIVE
B28	ATTN+		ATTENTION , High True, RECEIVE
B29	RD14+		Data Bit 14, High True, RECEIVE
B30	CTL2-		CONTROL 2 , Low True, SEND
B31	RD15-		Data Bit 15, Low True, RECEIVE
B32	STS2+		STATUS 2 , High True, RECEIVE

COLOR (TWISTED WITH " COLOR)

0 – BK BLACK
1 – BR BROWN
2 – R RED
3 – O ORANGE
4 – Y YELLOW
5 – G GREEN
6 – B BLUE
7 – V VIOLET
8 – GR GRAY
9 – W WHITE
X – BG BEIGE
OW OFF-WHITE

WIRE-WRAP CONNECTOR PIN OUT DIAGRAM - DIFFERENTIAL, ROW C

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
C1	SD0+		Data Bit 0, High True, SEND
C2	RD0+		Data Bit 0, High True, RECEIVE
C3	SD1+		Data Bit 1, High True, SEND
C4	RD1+		Data Bit 1, High True, RECEIVE
C5	SD2+		Data Bit 2, High True, SEND
C6	RD2-		Data Bit 2, Low True, RECEIVE
C7	SD3+		Data Bit 3, High True, SEND
C8	RD3+		Data Bit 3, High True, RECEIVE
C9	SD4+		Data Bit 4, High True, SEND
C10	RD4-		Data Bit 4, Low True, RECEIVE
C11	SD5+		Data Bit 5, High True, SEND
C12	RD5+		Data Bit 5, High True, RECEIVE
C13	SD6+		Data Bit 6, High True, SEND
C14	RD6-		Data Bit 6, Low True, RECEIVE
C15	SD7+		Data Bit 7, High True, SEND
C16	RD7+		Data Bit 7, High True, RECEIVE
C17	SD8-		Data Bit 8, Low True, SEND
C18	RD8-		Data Bit 8, Low True, RECEIVE
C19	SD9-		Data Bit 9, Low True, SEND
C20	SD10+		Data Bit 10, High True, SEND
C21	RD9+		Data Bit 9, High True, RECEIVE
C22	RD10-		Data Bit 10, Low True, RECEIVE
C23	SD11-		Data Bit 11, Low True, SEND
C24	RD11-		Data Bit 11, Low True, RECEIVE
C25	SD12-		Data Bit 12, Low True, SEND
C26	RD12-		Data Bit 12, Low True, RECEIVE
C27	SD13-		Data Bit 13, Low True, SEND
C28	RD13+		Data Bit 13, High True, RECEIVE
C29	SD14-		Data Bit 14, Low True, SEND
C30	RD14-		Data Bit 14, Low True, RECEIVE
C31	SD15-		Data Bit 15, Low True, SEND
C32	RD15+		Data Bit 15, High True, RECEIVE

WIRE-WRAP CONNECTOR PIN OUT DIAGRAM - HIGH TRUE, ROW A

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
A1	SD0		Data Bit 0, High True, SEND
A2	SPAR0		Parity bit 0,High True, SEND
A3	SD1		Data Bit 1, High True, SEND
A4			
A5	SD2		Data Bit 2, High True, SEND
A6			
A7	SD3		Data Bit 3, High True, SEND
A8	RPAR1		Parity bit 1,Low True,RECEIVE
A9	SD4		Data Bit 4, High True, SEND
A10	SHIELD		Frame ground
A11	SD5		Data Bit 5, High True, SEND
A12			
A13	SD6		Data Bit 6, High True, SEND
A14	PCTL		PCTL , High True, SEND
A15	SD7		Data Bit 7, High True, SEND
A16			
A17			
A18	CTL1		CONTROL 1 , High True, SEND
A19			
A20	GND		Signal Ground
A21	RD9		Data Bit 9,High True,RECEIVE
A22	PFLG		PFLAG ,High True,RECEIVE
A23			
A24			
A25			
A26			
A27			
A28	ATTN		ATTENTION ,High True,RECEIVE
A29			
A30			
A31			
A32	STS2		STATUS 2 ,High True,RECEIVE

WIRE-WRAP CONNECTOR PIN OUT DIAGRAM - HIGH TRUE, ROW B

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
B1	RD0		Data Bit 0,High True,RECEIVE
B2			
B3	RD1		Data Bit 1, High True, RECEIVE
B4	RPAR0		Parity bit 0,High True, RECEIVE
B5			
B6	SPAR1		Parity bit 1,High True, SEND
B7	RD3		Data Bit 3, High True, RECEIVE
B8			
B9			
B10			
B11	RD5		Data Bit 5, High True, RECEIVE
B12	PDIR		DIRECTION , High True, SEND
B13			
B14			
B15	RD7		Data Bit 7, High True, RECEIVE
B16	CTL0		CONTROL 0 , High True, SEND
B17			
B18			
B19	SD10		Data Bit 10, High True, SEND
B20	GND		Signal Ground
B21			
B22			
B23			
B24	STS0		STATUS 0 , High True, RECEIVE
B25			
B26	STS1		STATUS 1 , High True, RECEIVE
B27	RD13		Data Bit 13, High True, RECEIVE
B28	ATTN+		ATTENTION , High True, RECEIVE
B29	RD14+		Data Bit 14, High True, RECEIVE
B30	CTL2		CONTROL 2 , High True, SEND
B31	RD15		Data Bit 15, High True, RECEIVE
B32	STS2+		STATUS 2 , High True, RECEIVE

WIRE-WRAP CONNECTOR PIN OUT DIAGRAM - HIGH TRUE, ROW C

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
C1			
C2			
C3			
C4			
C5			
C6	RD2		Data Bit 2, High True, RECEIVE
C7			
C8			
C9			
C10	RD4		Data Bit 4, High True, RECEIVE
C11			
C12			
C13			
C14	RD6		Data Bit 6, High True, RECEIVE
C15			
C16			
C17	SD8		Data Bit 8, High True, SEND
C18	RD8		Data Bit 8, High True, RECEIVE
C19	SD9		Data Bit 9, High True, SEND
C20			
C21			
C22	RD10		Data Bit 10, High True, RECEIVE
C23	SD11		Data Bit 11, High True, SEND
C24	RD11		Data Bit 11, High True, RECEIVE
C25	SD12		Data Bit 12, High True, SEND
C26	RD12		Data Bit 12, High True, RECEIVE
C27	SD13		Data Bit 13, High True, SEND
C28			
C29	SD14		Data Bit 14, High True, SEND
C30	RD14		Data Bit 14, High True, RECEIVE
C31	SD15		Data Bit 15, High True, SEND
C32			

WIRE-WRAP CONNECTOR PIN OUT DIAGRAM - LOW TRUE, ROW A

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
A1			
A2			
A3			
A4	RPAP0		Parity bit 0, Low True,RECEIVE
A5			
A6	SPAR1		Parity bit 1, Low True, SEND
A7			
A8			
A9			
A10	SHIELD		Frame ground
A11			
A12	PDIR		DIRECTION , Low True,SEND
A13			
A14			
A15			
A16	CTL0		CONTROL 0 , Low True, SEND
A17	SD8		Data Bit 8, Low True, SEND
A18			
A19	SD9		Data Bit 9, Low True,SEND
A20	GND		Signal Ground
A21			
A22			
A23	SD11		Data Bit 11, Low True,SEND
A24	STS0		STATUS 0 , Low True,RECEIVE
A25	SD12		Data Bit 12, Low True, SEND
A26	STS1		STATUS 1 , Low True,RECEIVE
A27	SD13		Data Bit 13, Low True, SEND
A28			
A29	SD14		Data Bit 14, Low True,SEND
A30	CTL2		CONTROL 2 , Low True, SEND
A31	SD15		Data Bit 15, Low True,SEND
A32			

WIRE-WRAP CONNECTOR PIN OUT DIAGRAM - LOW TRUE, ROW B

27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
B1			
B2	SPAR0		Parity bit 0, Low True, SEND
B3			
B4			
B5	RD2		Data Bit 2, Low True, RECEIVE
B6			
B7			
B8	RPAR1		Parity bit 1, Low True, RECEIVE
B9	RD4		Data Bit 4, Low True, RECEIVE
B10			
B11			
B12			
B13	RD6		Data Bit 6, Low True, RECEIVE
B14	PCTL		PCTL , Low True, SEND
B15			
B16			
B17	RD8		Data Bit 8, Low True, RECEIVE
B18	CTL1		CONTROL 1 , Low True, SEND
B19			
B20	GND		Signal Ground
B21	RD10		Data Bit 10, Low True, RECEIVE
B22	PFGL		PFLG , Low True, RECEIVE
B23	RD11		Data Bit 11, Low True, RECEIVE
B24			
B25	RD12		Data Bit 12, Low True, RECEIVE
B26			
B27			
B28	ATTN		ATTENTION , Low True, RECEIVE
B29	RD14		Data Bit 14, Low True, RECEIVE
B30			
B31			
B32	STS2		STATUS 2 , Low True, RECEIVE

WIRE-WRAP CONNECTOR PIN OUT DIAGRAM - LOW TRUE, ROW C

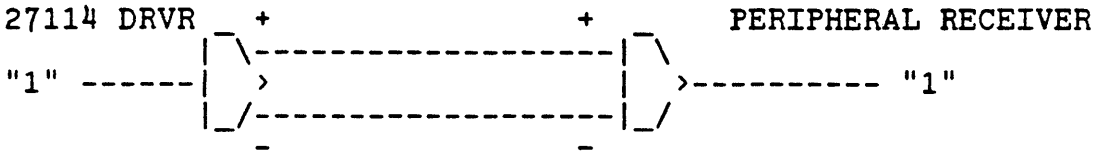
27114 CONN. PIN NO.	SIGNAL NMEM.	WIRE COLOR	SIGNAL DEFINITION
C1	SD0		Data Bit 0, Low True, SEND
C2	RD0		Data Bit 0, Low True, RECEIVE
C3	SD1		Data Bit 1, Low True, SEND
C4	RD1		Data Bit 1, Low True, RECEIVE
C5	SD2		Data Bit 2, Low True, SEND
C6			
C7	SD3		Data Bit 3, Low True, SEND
C8	RD3		Data Bit 3, Low True, RECEIVE
C9	SD4		Data Bit 4, Low True, SEND
C10			
C11	SD5		Data Bit 5, Low True, SEND
C12	RD5		Data Bit 5, Low True, RECEIVE
C13	SD6		Data Bit 6, Low True, SEND
C14			
C15	SD7		Data Bit 7, Low True, SEND
C16	RD7		Data Bit 7, Low True, RECEIVE
C17			
C18			
C19			
C20	SD10		Data Bit 10, Low True, SEND
C21	RD9		Data Bit 9, Low True, RECEIVE
C22			
C23			
C24			
C25			
C26			
C27			
C28	RD13		Data Bit 13, Low True, RECEIVE
C29			
C30			
C31			
C32	RD15		Data Bit 15, Low True, RECEIVE

3.4 Logic sense on frontplane interface

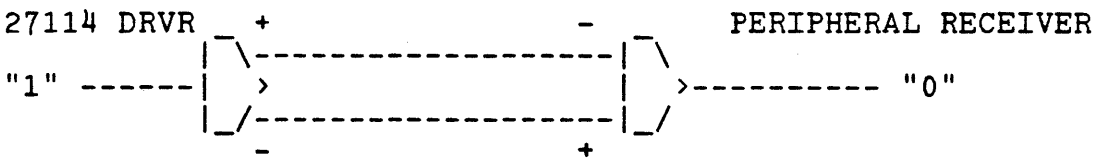
Low/high true:

With differential drivers and receivers, it is all a matter of how the cable is wired up or how the peripheral connector is wired up. As an example, if the peripheral connector and receiver is hooked up as in case 1 below, the output at the peripheral receiver will be of the same sense as that at the 27114 driver input. If the peripheral were hooked up as in case 2, the sense of the signal would be inverted.

Case 1:



Case 2:



NOTE:

Due to the fact that the CIO backplane bus is low true, a 0 bit in the program domain will be translated to a high logic in the corresponding circuit on the board, resulting in the + line being at a more positive voltage than the - line at the frontplane interface. This is true for both types of input and output signals.

The notation used in this document will be kept consistent in the following manner:

* 0 and 1 are for bits in program domain
* high/low are for logic lines in hardware (TTL level)
* Assertion of a signal means it is being brought high

3.5 Data Transfer

The data handshakes in words of 16 bits plus two parity bits. The 27114 is the master in each transfer.

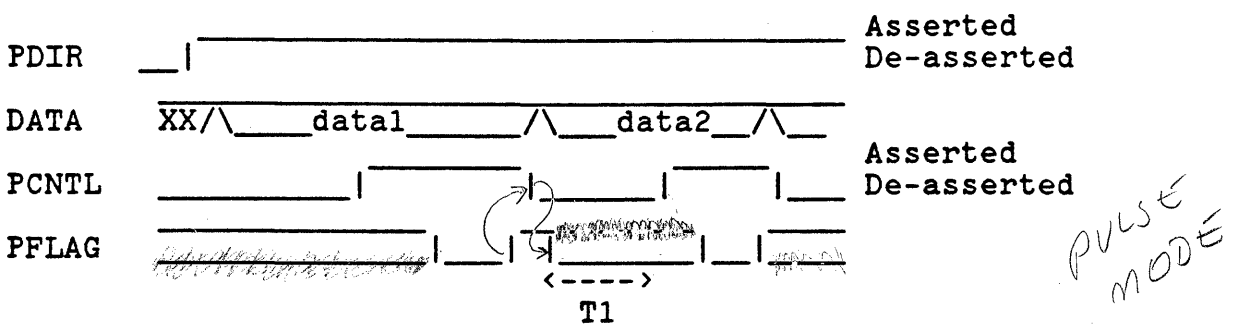
The following discussion presumes PFLG is used as is; not inverted by the use of the EDGE bit (in control output register 7 bit 5).

A 1 programmed in this EDGE bit would allows the falling edge of PFLG+ to trigger the peripheral handshake circuit to indicate that data reception is done (output mode) or data is available

(input mode). In the program domain, PFLG status bit is seen accordingly as a transition from 1 to 0.

The following timing diagram refers to the positive line of each signal.

27114 to Peripheral:



Data output:

The following sequence of events describes how and when data is handshaked across the frontplane with respect to PCTL and PFLG.

* The first output cycle begins when a data word trickles through the FIFO and appears at its output.

* As the cycle starts, data is placed on the data output lines. After T1 (+ 45 ns, PCTL is asserted high.

* A low-to-high PFLG from the peripheral indicates to the 27114A card that the data is received. PCTL is de-asserted (low) immediately here.

* If another data word is already present at the FIFO's output, this same low-to-high PFLG will start next output cycle. Otherwise, the next cycle will not start until another data word appears at the output of the FIFO.

  If or when another data word is available, PCTL will be asserted again. This is referred to as the "pulse" handshake mode. Some other IO cards like the 27112A GPIO also offer a "full" handshake mode whereas PCTL will not be asserted unless PFLG had returned to the deasserted state.

* The data word remains valid on the output for the entire data output cycle until at least after PFLG is asserted.

* The timing was drawn to highlight the fact that the high to low transition of PFLG could happen anytime and is of no concern to the card. Only the low-to- high transition triggers action in the card.

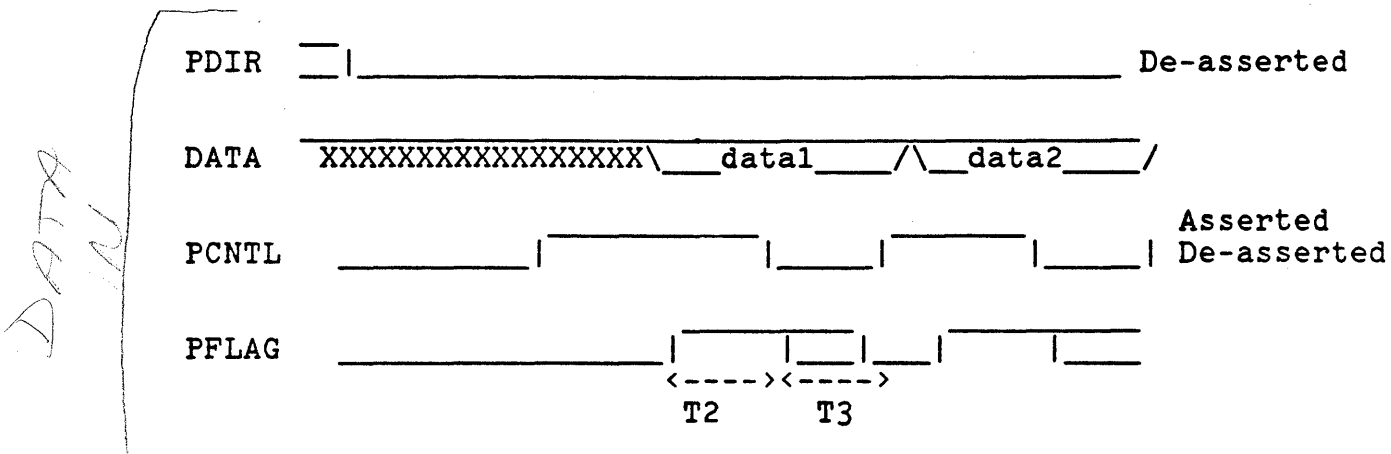
* The PFLG line must not go from low-to-high in any cycle before PCTL goes high. If it does, one of two things may happen: either it is ignored altogether or the cycle continues and

behaves as though PCTL did come high and followed by a rising edge of PFLG (data word is lost, of course).

NOTE:

A similar effect to the rising edge of PFLG can be done by switching the EDGE bit to the opposite state of PFLG.

PERIPHERAL TO 27114:



Data input:

* Right after reading is enabled, PCTL is asserted high.

* Data should be placed on input data bus before PFLG is switched from low to high. This starts the first input cycle.

* After T2 time, the input data is clocked into the FIFO. PCTL is brought low immediately. PFLG must remain high until PCTL is brought low. Erronous handshake could occur otherwise.

* PCTL remains low until the FIFO has room for data or until after T3 + 45 ns, whichever comes later. T3 is used to guarantee that the low pulse on PCTL is seen by the peripheral.

* If a rising edge of PFLG arrives when PCTL is low, one of three things could happen: the correct data word is received, an incorrect data word is received or the edge is ignored altogether.

* As indicated later, data to PFLG rising edge set up time could be negative and still assure proper data transfers.

* As in data output case, only the low-to-high transition of PFLG is used to trigger action in the 27114A card. However, PFLG must be pulsed low for sufficiently long to be recognized by the 27114A card (approx. 60ns).

3.6 Frontplane Timing

This section is limited to restrictions placed on the design of peripherals which are connected to the 27114 and guarantees given on timing to those peripherals. It does not consider overall performance or interaction between data transfer on the backplane of the card with data transfer on the frontplane of the card. A detailed description of overall card performance is to be found in section 5 of this document.

On the 27114A card there are two sets of three switches each. These switches control time values T1, T2 and T3 mentioned in the preceding section (3.5). When switches in the first set are closed, a default value is chosen such that optimal performance is guaranteed for a cable length of 3 m or less. When the other switches are closed, an alternate set of timing values are chosen. These alternate values are defined by R5, R7 and R9. These values are chosen for cables which are not longer than 24 m. Only one set of switches should be closed at any one time. If both sets of switches are closed, the effective result is that longer delays are given according to the new values (which are the equivalent of the two parallel resistors). This would slow down the transactions but otherwise does not cause any concern.

In the near future, a new set of timing will be included which would specify the available set up time for output data before PCTL is asserted, a required set up time for input data before PFLG could be asserted, the minimum available PCTL low time as well as the minimum allowable low time for PFLG; all parameters will be specified at the user's end of the cable. This would make more sense since the user doesn't have to be concerned about the effect of length on these if he/she uses either standard length setting available on the board (3 m and 12 m).

All timing values are referenced at the peripheral connector end.

Data output from device adapter (AFI) to peripheral: PDIR is asserted to indicate output transfer. Initial state: both PCTL and PFLG are deasserted.

AFI ACTIVITIES	PERIPHERAL ACTIVITIES
1. Put data on output data bus	
2. Wait for Tos (data set up) ~130NS USUALLY	
3. Assert PCTL (regardless of state of PFLG)	
	a. Wait for PCTL assertion
	b. latch data into peripheral
	c. Assert PFLG
4. Wait for PFLG assertion	
5. Wait for Tor ns ——— 0NS	
6. Deassert PCTL	
7. Remove data from bus	d. Wait for PCTL deassertion
8. END OF CYCLE (AFI)	e. Deassert PFLG
	f. END OF CYCLE (PERIPHERAL)

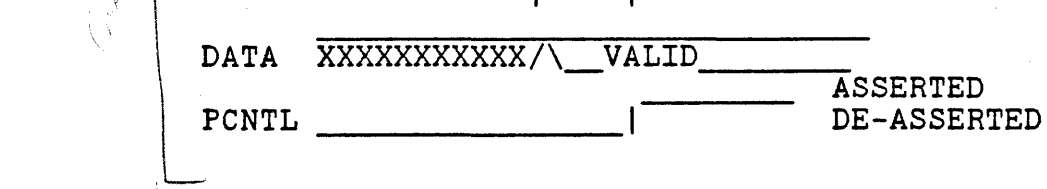
Data input from peripheral to device adapter (AFI):
PDIR is deasserted to indicate input transfer.
Initial state: Both PCTL and PFLG must be deasserted.

AFI ACTIVITIES	PERIPHERAL ACTIVITIES
1. Assert PCTL (regardless of state of PFLG)	
	a. Wait for PCTL assertion
	b. Place data on input data bus
	c. Wait for Tis ns (data set up)
	d. Assert PFLG
2. Wait for PFLG assertion	
3. Wait extra Tir ns Latch data in	
4. Deassert PCTL	
5. Wait extra Ticl ns	e. Wait for PCTL deassertion
6. END OF CYCLE (AFI)	f. Remove data from input data bus
	g. Deassert PFLG
	h. END OF CYCLE (PERIPHERAL)

Data transfer between the 27114 and a peripheral involves the following list of timing numbers:

Tos - Minimum data setup time prior to assertion of PCNTL at the end of a cable. A user must consider the effects of skew in propagation delay on the cable and skew in the peripheral devices receivers.

A value of 54 ns is supplied. Actual measured value centers around 130 ns.



****

tclo - Minimum low time for PCNTL. In the sequence of events during a handshake, some time after PCNTL is asserted, PFLAG is asserted in response. Then, the 27114 is at liberty to de-assert PCNTL. This timing number guarantees that it won't assert PCNTL again until after some minimum amount of time. The reason for specifying a minimum is to ensure that a peripheral will see PCNTL low long enough for the peripheral logic to notice it and see the subsequent assertion of PCNTL.

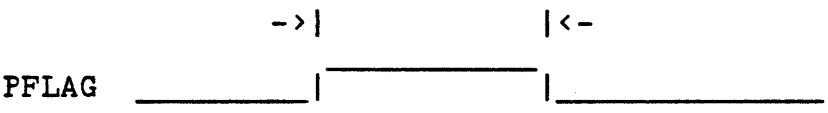
If a user has an application which has a requirement for an extremely long cable, resistance in the cable will reduce this time.

Value = 60nsec minimum for 3 m cable

****

tfh - Minimum time which PFLAG must be high at the 27114 end of the cable to ensure the signal is seen by the 27114. This is a requirement which must be satisfied by the customer's equipment

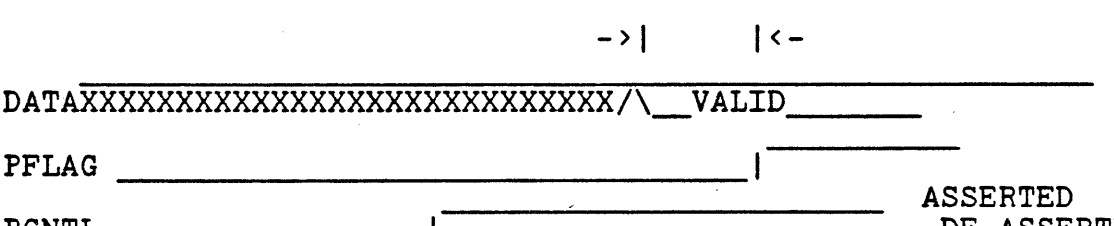
It is specified here that the peripheral cannot deassert PFLG until it sees the deassertion of PCTL.



****

tis - Minimum setup time for data driven by a peripheral prior to PFLAG assertion. Time between data valid and PFLAG asserted measured at 27114 end of cable.

Value = 0 nsec

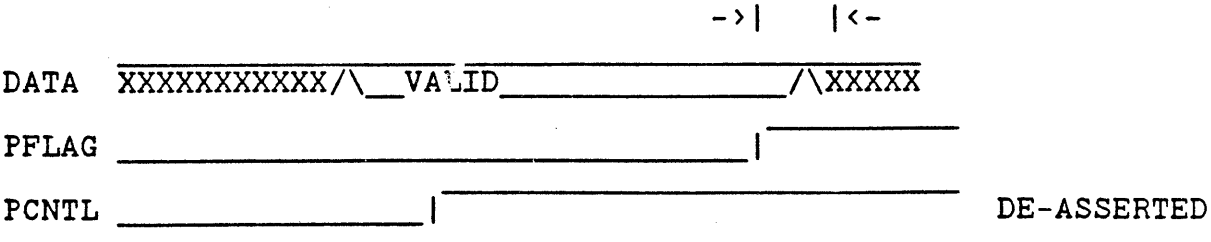


Note: Data driven by peripheral.

****

toh - Minimum time data is held on the cable by the 27114 after it sees the assertion of PFLAG acknowledging the acceptance of data by the peripheral.

Value = 0 minimum

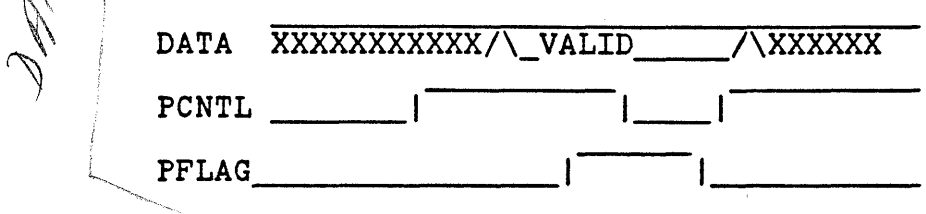


Note: Data driven by 27114

****

tih - Minimum time data must be held on the cable by a peripheral after the peripheral sees the deassertion of PCNTL.

Value = 0 minimum



To extend cable length, the user is recommended to go directly to DuPont Connector Systems, New Cumberland, Pennsylvania, for a custom made cable. The cable will be made with the same specs as that of the supplied cable with the exception of a different cable length.

NOTES ON RADIO FREQUENCY INTERFERENCE PERFORMANCE:

The Hewlett-Packard Company have tested and found that any supported systems configured with this card and the standard length (3 m) cable will comply to all applicable existing radio frequency interference regulations and requirements when the cable is used in a system configuration in which two supported machines are connected to each other.

In cases where users do not use the standard length cable, they are responsible for verifying and bringing their final system configuration into compliance with the applicable regulations/requirements.

3.7 Back-to-Back 27114 Link

Connection of an 27114 to an 27114 requires some special software in order to work. The cable used is the supplied cable which maps appropriate control, status, handshake lines, and interrupts into themselves in a useful fashion. Only the hardware aspects of the connection is described in this section. See section 4.8 for a description of the software aspects.

3.7.1 27114 to 27114 Cable

The mapping of cable pins is symmetric because both ends of the cable plug into the same type of connector. The mapping of connections follows:

PCNTL	---->	PFLAG
PFLG	<----	PCTNL

```
DATAout[0:15]---->  DATAin[0:15]
DATAin[0:15] <----  DATAout[0:15]
PDIR         ---->  STS[2]
STS[2]       <----  PDIR
ATTN         <----  CTL[2]
CTL[2]       ---->  ATTN
CTL[0]       ---->  STS[0]
CTL[1]       ---->  STS[1]
STS[0]       <----  CTL[0]
STS[1]       <----  CTL[1]
GND          ---->  GND
GND          ---->  GND
```

There is no connection for the TEST pins.

The supplied cables, both differential and single-ended ones, have taken in to account this AFI-AFI connection. Both of them are wired exactly as the recommendation above. When used, each end of the cable should plug directly into an 27114 AFI card. For single-ended cables, since one end is for low true and other end is for high true application, care must be given to configure the terminating resistor packs according to the way the cable is used at that end (one card is low true, one card is high true). Note also that the data and control are all inverted in case of the single-ended cable. In this case, normal handshake can be provided by reversing the EDGE bit definition in both cards.

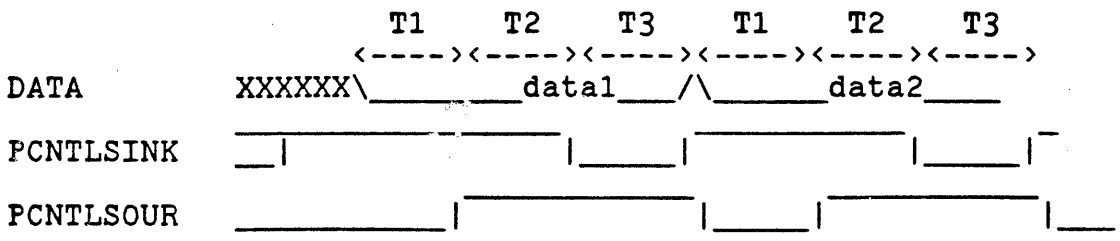
3.7.2 Back-to-Back Data Transfer

The process of connecting two 27114 cards to each other presents a rather odd handshake situation. The card at each end wants to behave as the master. Data transfer takes place as follow: one end or the other will assert its PCNTL first and the other end will see it as a PFLAG prior to having asserted its first PCNTL. One possible scheme is presented here.

This case is illustrated from both the source and sink points of view in the timing diagrams below. This illustrates a two word transfer.

Note: the PCNTLSINK line is left asserted. This is because it has more room in its FIFO for data.

Details of a possible software scheme are left for section 4.8. These include suggestions for the use of CTL[2:0], STS[2:0] and interrupts. The timing diagram below ignores the procedure of setting up the transfer.



a b c d e c d e

Handshake progress (a,b,c,d and e as referred to by the timing diagram):

a. SINK card brings its PCTL line high, waiting for peripheral (SOURCE card) in its data input mode.

b. SOURCE card places data on bus. First data output cycle starts (for SOURCE)

c. SOURCE card brings its PCTL line high, indicating that it has valid output data on bus. The time elapsed between b and c is controlled by timing value T1 in the SOURCE card. The SINK card recognizes PFLG line high and knows that its peripheral just put data on the bus.

d. After an elapsed time T2, SINK card brings its PCTL line low to acknowledge to the peripheral (SOURCE card) that it has seen PFLG coming high.

e. The SINK card brings PCTL high again after T3 time. This could be delayed until its FIFO is not full.

   Recognizing that its peripheral has recognized its PCTL line's going high by asserting PFLG high, the SOURCE card drops its PCTL line and puts a new data word on the bus to start another data transfer cycle. This could be delayed until its FIFO is not empty.

It should be remembered that in both input and output modes, the PFLG rising edge is the key to all timing (falling edge if EDGE bit is a 1).

NOTE: The synchronization between the two cards must be done properly to avoid data words being dropped or added unexpectedly. The sender should check its data path to see if all transmitted words are actually consumed.

4. BACKPLANE CONTROL DESCRIPTION

NOTE:

A lot of what described here only applies to the driver writers or diagnostic writers. The user should not be too concerned about what happens on the CIO backplane or on the 27114A card itself.

The 27114 card is a level 1 CIO card. As such, it behaves somewhat differently than what has been called CIO in the past. This changes the following things:

Subchannel address is device address.

"Commands" are all bit-bucketed.

Status is always reported as AES.


DBYT and DEND are never asserted by the card.

Some specific attributes of the 27114 card:

Write orders and read orders must be suppressed (they are ignored by AFI).

The only reason an 27114 ever responds to SRQ polls is for the movement of data. It does this whenever it is ready for data transfer.

Self test results SENSE[] are always "pass". No selftest is done by the card itself.

The card is always RFC (ready for command). Commands themselves are ignored.

The card must be set up for data transfer by manipulating bits in the AFI CONTROL register.

4.1 Register Definitions

NOTE:

Most registers are inaccessible to the high level (end) users. The degree of accessibility varies with the particular driver on that system.

The decoding of BP1, BP0, CEND and CBYT is as follows (X means don't care):

READ REGISTERS:

BP1	CBYT	CEND	BP0	Register definition
0	X	X	0	DATA
0	0	0	1	SENSE
0	0	1	1	ID
0	1	1	1	27114A STATUS
1	X	X	1	STATUS

WRITE REGISTERS:

BP1	CBYT	CEND	BP0	Register definition
0	X	X	0	DATA
0	0	0	1	CONTROL
0	1	1	1	27114A CONTROL

Writing to illegal address combination will not cause any action on the card.

Reading using illegal address combinations will return unspecified values.

4.1.1 Register descriptions:

*DATA[15..0]: READ/WRITE. U31 & U71 ?

This is a 16 bit wide location on the card. The 27114 must be set up for a data transfer in the appropriate direction and be polling for data (responding to SRQ polling). Otherwise, a read of or write to this register will cause incorrect state changes on the card i.e. the data path will be altered requiring a DCL to clear it up. A read following a write will not necessarily return the same data.

There is one exception to this rule. After a hard reset of the card, a write to the data register with UAD asserted is used to wake up the card.

The most significant bit of a 16 bit word will end up at DOUT[15] on the frontplane connector.

There is no word packing or unpacking on the board. Therefore, accessing the data path with CBYT asserted is seen as an attempt to move a 16 bit word by the 27114 card. CBYT and CEND have no meaning for an 27114 card beyond their usefulness as address bits.

*SENSE: READ ONLY.

The sense register is always immediately accessible once the card has been awakened. The bits returned are as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*R	*R	*R	*R	*R	*R	*R	*R	RFC	PST	PRE	NMI	LV1	ARE	R	ARQ

Some of the bits on the 27114 are hard-wired. What you really get back looks as follows:

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

The ARE and ARQ bits behave per the CIO standard.

✳CONTROL: WRITE ONLY.

This register is used to enable and disable interrupts. It is also used to do an addressed device clear e.g. reset the 27114 without disturbing other cards in the backplane. The bits that are there behave per the CIO standard.

The bit map for the control register follows:

BB stands for Bit Bucket. These bits have no effect.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BB	BB	BB	BB	BB	BB	BB	BB	BB	BB	DCL	DEL	BB	BB	ARE	ARD

✳ID: READ ONLY.

ID is inaccessible until the card has been awakened after a reset. The returned value is a hardware constant used to identify this extended CIO card as an 27114 card. The returned value also contains a revision number for the card.

The bit map is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	PE	R2	R1	R0	0	0	1	0	0	0	0	0

R[2..0] indicate the hardware revision number of the board. They are returned as 0 with this revision of design.

PE is returned as 1 if parity is enabled and 0 otherwise. Parity enable is configured with switches 7 and 8 of SW1. Parity is odd only (parity fault does not result if all CIO backplane data lines including parity lines are low).

NOTE:

Most CIO channel adapters do not generate nor check for parity. The 27114A generates parity only for the CIO read only registers. It does not check parity for CIO write registers. Parity for data read/write is the responsibility of the peripheral if parity is enabled (the card passes parity bits directly between the CIO backplane and the peripheral interface).

✳STATUS: READ ONLY.

The STATUS register always returns the CIO value for AES (asynchronous event sensed). Reading this register clears the flipflop which had latched the interrupt pending condition (ATTN input).

NOTE:

The interrupt circuit is edge sensitive so any reading of the STATUS register could wipe out a pending interrupt. The sense register should be read and handled accordingly to avoid missing interrupt.

The bit pattern returned is as follows:

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

✳27114 STATUS: READ ONLY.

This register is used to monitor several of the basic functions of the card. These bits are the sampled value taken at the last assertion of the backplane line, SYNC-.

The bit map is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								P	P	O	I	T	S	S	S
0	0	0	0	0	0	0	0	C	F	R	R	E	T	T	T
								L	G			S	2	1	0

4.1.1.1 Definition of Status Bits

NOTE:

Except for the 3 status bits STS[2..0], most bits will not be visible to the high level user unless so supported by the respective driver.

PCL -- This bit returns the state of the PCNTL handshake signal as sampled at the time when SYNC- last went low on the backplane. The returned value is a "zero" if PCNTL is asserted to the peripheral. ↖ PROGRAM VALUE; CARD VALUE = 1

PFG -- This bit returns the state of the PFLAG handshake line as sampled at the time when SYNC- last went low on the backplane. The returned value is a "one" if PFLAG is asserted by the peripheral. ↖ PROGRAM VALUE; CARD VALUE = Ø

DATA IN PATH WAITING

OR -- This bit returns the state of the FIFO's output ready flag as sampled at the time when SYNC- last went low on the backplane. The returned value is a "one" if the FIFO was capable of sourcing more data at that point in time. OR is useful in determining if things are hung up in a transfer to a peripheral, and as part of a self test. It must not be used to indicate that there is no data left in the path (input or output mode).

ROOM IN PATH READ

IR -- This bit returns the state of the FIFO's input ready flag as sampled at the time when SYNC- last went low on the backplane. The returned value is a "one" if the FIFO was capable of taking more data at that point in time. This does not necessarily mean that the card is ready for more data. It is useful in determine if things are ~~bye~~ hung up in a transfer to a peripheral, and as part of a self ~~bye~~ test. It must not be used to indicate that there is no room in the data path. bye

STS[2..0] -- These bits return the state of the STS[2..0] lines driven by the peripherals (unconnected lines reported as zeros).

The STS2 bit is valid only when bit DIR is programmed with a 1 to define a data input mode (since the STS2 line receiver is enabled with the data receivers).

NOTE:

If there is any doubt as of the current direction of transfer, STS2 should not be used at all.

TES -- This bit is returned as a 1 if a test hood is present or if pin A10 of the frontplane connector is connected to ground for some reason.

27114 CONTROL: WRITE ONLY.

NOTE:

Like the 27114 STATUS register, most bits in here will not be accessible to the high level user. The exception would be the three CL[2..0] control bits.

The 27114 CONTROL register is used to control several of the functions of the 27114. Care should be taken not to disturb the CL[2..0] bits when other input/outputs are activated.

The bit map is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	B	B	B	B	B	B	B	P	D	E	C	P	C	C	C
B	B	B	B	B	B	B	B	R	I	D	L	E	L	L	L
								N	R	G	F	N	2	1	0

PRN -- This bit is used to enable and disable SRQ poll responses from the card. If it is written as a "0" the card will be capable of responding to SRQ polls. If it is written to a "1" the card will not respond to SRQ polls. After a hard reset or device clear, this bit is set to "1".
↑ SW VALUE; HW = 0

CLF -- Clear FIFO: If set to 1 this bit clears the FIFO data path and reset the state machine which controls it. If left at "1" it will continue to do so. To effectively reset the data path, this bit must be set to "1" then reset to "0" while other bit values remain the same. After a device clear or a hard reset this bit will be set to "1".

DIR -- This bit is used to control the set up of the card for data transfer in one direction or the other. If written as a "1" the card will move data toward the CIO channel adapter from the peripheral (input mode). A "0" defines output mode. After a device clear or reset, the direction is set for transfer toward the channel adapter.

EDG -- This is the bit which determines which edge of the line, PFLAG, is used to cause or indicate data movement. If this bit is written as a "0", the rising edge of PFLAG is used. If not, the falling edge is used. After as hard reset or device clear, this bit is set as though it had been written with a "1".

PEN -- This bit is used to enable the frontplane to respond to handshakes. In effect, it turns on and off the frontplane of the card. If set to "0" it enables the frontplane to respond to handshakes. If set to a "1" it prevents the frontplane from responding to handshakes. After a hard reset or device clear, it is as though it was set with a "1", disabled.

CL[2..0] -- These three bits are used to write control information to a peripheral device. They feed out to the frontplane connector. After a hard reset or device clear, they are all set in such a fashion as to be seen as all zero by the peripheral. These are inverted from the sense seen by a program. Writing a "0" to these bits sets the lines seen by a peripheral devices to "1". The state of these three bits does not affect the card.

NOTE:

The best way to ensure that there is no racing condition on the 27114A hardware (which is essentially a state machine) is to change one bit at a time. To prevent unwanted side-effects in data handshaking between the 27114A and the peripheral, the PEN bit should be disabled until data is ready to be exchanged.

4.1.2 Card Identification

NOTE:

This section should be transparent to the high level user. Power up or initialization should be taken care of by the respective drivers.

After card power up or a hard reset via the RST line on the backplane, the 27114 behaves as though it is asleep. It is "awakened" by a Write Data operation with /UAD, a line on the backplane, asserted. The first action taken by code attempting to identify the card, should be a Read Sense operation. The code should "know" that the card is an extended CIO card because the LV1 bit in the SENSE register is set. Being a level one card, the code must read register 3, the ID register to determine the type of card which is in the slot. The returned value, which indicates a 27114 is in the slot, is 32. The least significant byte is the ID. The three least significant bits in the upper byte forms the revision number of the hardware. The driver must deal with the card according to the revision number. Different revision number could imply that there are changes in the hardware. This ERS applies only to rev 0.

4.2 Data path pipeline

The data path consists of a FIFO which contains 64 data words and four extra latches, two in front and two behind the FIFO. A state machine is used to control the flow of data through the pipeline.

4.2.1 FIFO and external latches

The FIFO can store up to 64 words. Since its status outputs (input ready IR and output ready OR) are updated only after the current transfer is done, one extra level of latches is provided should there be an extra transfer already initiated by the channel adapter. Also, since the channel adapter still can initiate (yet!) another transfer even after burst request is turned off (as a consequence of FIFO being full in output or empty in input mode), a second level of latches is required. These two levels of latches are used only to smooth out data transfer between the FIFO and the backplane. Two latches are provided for data output mode (PRE latches) between the backplane and the FIFO input. Likewise, two POST latches are inserted between the FIFO output and the backplane.

THERE ARE ACTUALLY THREE LATCHES BETWEEN THE BACKPLANE AND THE FRONTPLANE FOR DATA GOING TO THE PERIPHERAL FROM THE BACKPLANE.

4.2.2 State machine

The state machine takes one out of four possible states 00, 01, 10 and 11. These states are used to indicate the presence of data words in transit between the backplane and the FIFO (either in PRE set or POST set of latches).

The two bits Y1 and Y0 (less significant bit) are assigned such that a 1 in Y0 means that a valid word is being latched in the near latch. Similarly, a 1 in Y1 means a valid word is in the far latch (further from FIFO). State 11 means both latches have valid words, 01 means only near latch is valid etc.... Note that only one set of PRE or POST latches can be situated between the backplane and the FIFO in a certain input or output mode.

The state machine is reset to state 00 at power up time or any time the CLF (clear FIFO) bit is programmed with a 1.

INPUT MODE:

In input mode (data from peripheral to backplane), the POST latches are used. When data is available from the FIFO, they are trickled into the near POST latch and then into the far POST latch, in that order.

Positive SRQ is possible only when there is a valid data word in the far POST latch. Burst request is asserted only when both POST latches contain valid data words. This is done to guaranteed that the backplane can DMA without being shortchanged with invalid data words.

The possible state transition sequences are ([xN] means repeatable N times):

00, 01, 10, 00 :single data word being read from FIFO
00, 01, 11 [xN], 10, 00 :multiple data word available from FIFO
or
00, 01, 11 [xN], 10, 01 ...

IR = 1
IR = 0

OUTPUT MODE:

In data output mode, the PRE latches are used to buffered data once the FIFO is full. Once the FIFO becomes full, the next data word is latched in the near PRE latch. If the FIFO becomes empty, it will be moved into the FIFO. In case a second word comes when the near PRE latch and the FIFO are both full, the far PRE latch is used. Once this state (11) is reached, the near word must be moved in first before the far word can be. In this case, the far word falls through (transparent to) the near latch. Both latches are of transparent type.

Positive SRQ poll responses are possible only in states 00 and 01. Burst request is asserted to the backplane only in state 00.

The following possibilities for state transition sequences exist:

```
00 [xN]                    :FIFO never full
00, 01, 00                 :FIFO full temporarily
00, 01, 11, 10, 00         :FIFO full with two extra data words
```

NOTE: The above description of the state machine applies to both data bursting and single data word transfer.

4.2.3 SRQ poll response and burst request

SRQ (service request) and burst request are activated accordingly to the state of the state machine depending on the operating mode (input or output).

SRQ is carried out by asserting the UAD backplane line whenever a service poll is conducted (this is done constantly by the channel adapter). SRQ is the only mean which user's program can find out if the card is ready for data transfer or not (Not the status of input ready or output ready from the FIFO).

Since the method for responding to service/attention polling is to assert a data line (D0 through D7) on CIO backplane, the AFI card works only when it is installed in a slot which UAD line is tied to such a data line. These slots belong to poll group 0's. (In machine like DAWN 9920, internal slots 4,5,6,7 and all external IO extender slots 8 through 15 and 16 through 23 are poll group 0's slots. In machine like Indigo, only slot 0 to 6 are poll group 0's slots). Unspecified consequences can happen if the AFI is plugged in an unsupported slot.

Burst request is asserted by driving the proper backplane line active. This indicates to the channel adapter that the card is capable of doing data bursting, at least for two transfers anyway.

4.2.4 Backplane data handshaking

The 27114A does not keep track of how many transfers it should carry out or it have performed. Nor can it interrupt the channel adapter on a similar cause. It is all up to the channel adapter to determine how many more transfer it can afford while the 27114A card keeps demanding for data transfers whenever its

condition permit (room in output mode and data available in input mode).

4.2.5 Frontplane data handshaking

The state machine does not control directly the interaction between the FIFO and the frontplane handshaking. Frontplane handshaking is done when both the FIFO and the peripheral are ready. This is done via the full synchronous handshake of PCTL and PFLG as described in section 3.5 and 3.6 As described earlier, the PEN bit in the 27114 CONTROL register must be written with a 0 to enable frontplane handshake.

Note that any time the EDG bit is programmed to the same state as that of the PFLG bit, a simulated triggering edge is generated to the frontplane interface circuit. This could be very effective in either selftest or when no peripheral handshake is needed or available.

In data input mode, data words are moved directly from the line receivers to the FIFO input. In output mode, the line drivers are actually driven by the near POST latch which provides one extra level of buffering.

4.3 Data Path Loopback

NOTE:

As best as it can be described here, these written paragraphs will guide the users (driver writers) through in a not so efficient way. The circuit diagram (schematic) is the best and utmost description of what the card actually does and should be consulted whenever doubts exist. Many fine tuning processes/tricks cannot be described here without going through hundreds of pages.

This section describes a sequence of events which could be used to loop back data through the data path on the 27114. Such an exercise might be part of a power up self test of the card. The data path loopback can be done with or without the test hood.

The sequence of control given assumes that the card is in the state in which it is left after a device clear and a device enable have been performed.

There are some hardware facts to bear in mind when doing loopback. They relate to the fact that the data path on the 27114 is a pipeline consisting of registers and FIFO. The FIFO on the card is only 64 words deep, but has been extended in depth by two extra registers. This means that the 27114 can take 66 words of data without the frontplane being enabled. However, these extra two words will get lost or messed up in a data loopback situation.

4.3.1 Loopback without test hood

The process of doing a loopback consists of several writes to the 27114 CONTROL register, followed by an enable of DMA to the card. After the DMA to the card, several more writes to the 27114 CONTROL register are required followed by an enable of DMA from the card. Instead of DMAing to/from the card, normal writes and reads can be used to send and receive data.

Any length of transfer from one to 64 16-bit words is acceptable. Any data looped back from the 27114 tests the entire length of the data path. The FIFO is a fall-through type and for that reason, all locations in it are tried, even if only one word is passed through it.

A possible sequence of actions for a loopback follows (all values are to be written to the 27114A CONTROL register unless otherwise noted):

1. write 0001 1000 to define output mode, enable poll response, reset clear FIFO and disable peripheral.
2. write 0000 1000 to set clear FIFO.
3. write N 16-bit words of data to DATA register. N is between 1 and 64.
4. write 0100 1000 to define input mode now.
5. read N 16-bit words of data from DATA register and compare to those sent in step 3. They should be the same.

Read and write access to DATA register in step 3 and 5 should be instantaneous (no need to check for SRQ poll response) for either direct access or DMA bursting mode.

4.3.2 Loopback with test hood

The presence of a test hood can be sensed by looking at bit TES of the AFI status register. A 1 indicates that the test hood is currently plugged in.

The test hood basically connects the data and control lines in the same manner as that of two AFI cards (see the AFI-to-AFI section). With it, most line drivers and receivers are fully tested (this is the only difference between loopback testing with and without test hood).

NOTE:

Even though the drivers are connected to the receivers and checked out, no loading are present at the connection points. Marginally good line drivers/receivers then may check out OK but could fail miserably when used in actual set up with the cable loading being the factor.

The STS2 line receiver and DIR line driver are not completely tested since reading of STS2 requires DIR bit to be defined as 1 (input mode).

With the test hood plugged in, the line drivers are always enabled, even during reception mode.

Either the EDG bit or the PEN bit must be programmed high to fill up the FIFO in output mode. Otherwise data will be automatically latched out to nowhere land as soon as they are available from the FIFO during output operation. Same thing is true in input operation: FIFO is filled up with phantom data words whenever there is room in the FIFO.

Since CTL[0-1] and STS[0-1] are connected to each other, testing these lines is straightforward.

The CTL2-ATTN pair can be tested by attempting to interrupt the card by properly programming the CTL2 bit and observing the response of ATTN flipflop.

The STS2-DIR pair cannot be fully tested due to the fact that STS2 can be read only in the input mode.

PCTL-PFLG pair can be tested by filling up the FIFO with PEN programmed with a 0 while doing the data path loopback test described in 4.3.1 There should be no data words in the FIFO when the direction is reversed. An alternative method is to read the PFLG bit when PCTL is toggled.

Data lines are tested by doing a frontplane loopback test. The test is very similar to the data loopback test mentioned in 4.3.1 In this test 3 words are written out to the FIFO. When the direction is reversed, two data words are latched out to the external latches outside of the FIFO providing room for at least another two data words. Every time the EDG bit is switched from the same state to the opposite state of PFLG, one extra data word will be latched into the FIFO (if there is room). These words must be identical to the one being latched in the outside latch closer to the FIFO (second data word written in output mode). Since PEN is programmed with 1, PCTL and PFLG remain at 1 all the time.

The following sequence illustrates how a data word can be clocked through the test hood:

OUTPUT these:

```
1. control 0001 1XXX   (reset fifo state machine and)
2. control 0000 1XXX   (disable frontplane interface)
3. data    data word 1 (write 3 data words out)
4. data    data word 2
5. data    data word 3
6. control 0010 1XXX   (turn on edge bit)
7. control 0110 1XXX   (now define input mode)
8. control 0110 0XXX   (enable frontplane)
9. control 0110 1XXX   (disable frontplane)
```

then INPUT these

```
10. data    word A     (input four data words)
11. data    word B
```

12. data word C
13. data word D

Data word A, B and C must be the same as data word 1, 2 and 3, respectively. Data word D must be the same as data word 2. This is the data word which gets transferred through the line driver, test hood and line receiver circuits.

After step 5, there are three words of data in the fifo. Step 6 and 7 define the input mode and at the same time move the first two words out and latch them in the POST latches. The third data word is at the output of the fifo. The second word is latched in the near POST latch. Since the output of this latch drives the line drivers, and the line drivers drive the line receivers (via the test hood) and the line receivers drive the fifo, further data input to the fifo should have the same pattern as the second data word.

When step 8 enables the frontplane interface, the fifo is filled up instantly since PCTL is routed back to PFLG and the EDGE bit had been cleared. Thus when step 9 disables the frontplane interface, there will be a total of 66 data words with the first three the same as what were written and the 4th through 66th identical to the second word.

4.4 Interrupts

The interrupt discussed here is the ARQ interrupt. This is not to be confused with interrupt schemes used on some other CIO cards.

There is one hardware signal in the cable labeled ATTN. If it is de-asserted, it will set a flip-flop in the card unless that flip-flop is being held reset by a device clear or hard reset state. If ARE is set on in the WRITE CONTROL register on the 27114, and this flip-flop becomes set, the ARQ line on the backplane will be pulled low requesting an ARQ poll from the channel adapter. The channel adapter can then read STATUS from the card, resetting the ATTN flip flop on the card. The STATUS value is always a simple AES status byte.

If ARE is not set, the 27114 will not pull ARQ low upon seeing the ATTN flip flop set, however, SENSE[arq] will still be available to the driver.

4.5 Output to a Peripheral

Output to a peripheral is accomplished with several writes of the 27114 CONTROL register followed by a data transfer being enabled, followed by some clean-up. The following sequence assumes that the card starts in the state it would be left in after a device clear followed by a device enable. This to preclude the possibility that a previous abnormal transfer left data in the data path. During a normal sequence of transfers from in either direction between the peripheral and the 27114, the driver must see to it that such is the case prior to each successive transfer.

A possible sequence follows:

1. Using the CTL[2..0] lines accessible as bits in the 27114 CONTROL register and perhaps forcing a device clear and enable, the driver does whatever is necessary to get the peripheral device ready to receive data.

2. Write 27114 CONTROL with 1000 1XXX to set direction of transfer out toward the peripheral without enabling frontplane or backplane transfers.

Note:The state of PEN doesn't really matter to the 27114 or the backplane here. The state of PEN may matter to the peripheral. If the peripheral won't mis-behave because it sees the wrong direction on PDIR and for some reason generates a PFLAG without a PCNTL, there should be no problem.

3. Write 27114 CONTROL with 0000 1XXX and 0000 0XXX to enable the 27114 to respond to polls and handshake with the peripheral.

Note: At this point the 27114 is responding to SRQ polls and is simply hanging the frontplane by not asserting PCNTL given that it has no data to pass on to the frontplane. Step number 4 could take place prior to step 3 if so desired.

4. Set a timeout for the Device Address.

5. Enable the channel adapter to do data bursting to the 27114.

6. Wait for the DMA to finish.

7. Because of the FIFO pipeline in the data path, check the card to see if the transfer from the 27114 to the peripheral is really done. To do this, read the 27114 STATUS register and check to see if OR indicates the existence of data in the FIFO (/OR bit being "1") and check to see if PCNTL is still asserted to the peripheral (PCL bit being "0"). Until this shows the peripheral to have taken all the data (OR "0" and PCL "1"), the transfer is not really done.

Note: This clean-up procedure is application dependent. In some cases, it might be better if the peripheral device interrupted via the ATTN line after it gets all the data it is going to take. After this, as a check to see that the DMA count sent equals to DMA count taken, it might be appropriate to check on things mentioned in #7 above. In some cases, this additional check may be deemed so unlikely to indicate a problem that it could be dispensed with. It depends on the ability of the application to tolerate data transfer problems. These checks are very useful when debugging a new driver or a new peripheral interface design. The decision to use such checks in a finalized driver is a tradeoff between the ability of the driver to catch and handle error conditions in a "nice" fashion and the speed of execution of the driver.

4.6 Data input from a Peripheral

Input from a peripheral follows much the same scheme as Output to a peripheral. The sequence starts from the point of no data in the FIFO and neither frontplane nor backplane interfaces on the card enabled to transfer data.

A possible sequence follows:

1. Using CTL[2..0] accessible as bits in the 27114CONTROL register and perhaps forcing a device clear and enable, the driver does whatever is necessary to get the peripheral device ready to source data.

2. Write 27114 CONTROL with 1100 1XXX to set direction of transfer in from the peripheral without having enabled frontplane or backplane transfers.

3. Write 27114 CONTROL with 0100 1XXX and 0100 0XXX to enable the 27114 to to polls and handshake with the peripheral.

Note: At this point the 27114 begins to attempt to handshake data into its FIFO from the peripheral. As soon as it gets data from the peripheral, it will begin responding to SRQ polls. Step #4 could be done prior to step #3.

4. Set a timeout for the Device Address.

5. Enable the channel adapter to do the DMA from the 27114.

6. Wait for the DMA to finish.

7. Because of the FIFO pipeline in the data path, check the card to see if the transfer from the peripheral to the 27114 is really done. The first step is to disable the front plane with a write to the 27114 CONTROL register of 0100 1XXX to prevent any more transfers from the peripheral to the 27114. Then check to see if the 27114 is still responding to SRQ polls for data after the DMA finishes. If it is, the peripheral has transferred more data to the 27114 than the size of the DMA which was set up in the channel adapter.

Note: The 27114 doesn't "know" about the DMA count programmed into the CIO channel adapter. If it has space in its FIFO for data it will keep on asking for it. Even if nothing goes wrong, the 27114 will ask for more data than should be transferred simply because it has room for it. This means that the peripheral either has to "know" how many words it is going to send or it must not hurt anything if it sends too much data. In effect, the 27114 will always ask the peripheral to send more data than it should at the tail end of a transfer.

4.7 Dealing with the Data Path Pipeline

This section describes any 27114 behavior which might seem "odd" to a user. These oddities are caused by the pipelined nature of

the data path or by the lack of "intelligence" on the card in the form of a microprocessor. The FIFO problems are those a user would encounter in any pipelined operation. The lack of "intelligence" simply forces the driver to do more of the detailed control of the 27114. This direct control of the hardware requires that the sequence of events must be correct. If it is not, it could result in unintended changes of frontplane control lines or changes in the state of the data path.

Mistakes resulting in unintended changes to the pipeline will show up as "lost" knowledge as in the case of mis-matched DMA counts between the peripheral and the 27114. With a FIFO in between the two ends of the card, it is possible for the source to send more data than the sink takes without either end noticing. If a user has total confidence in the hardware and software at both ends of a link, there is no problem. If not, or if the results of losing the knowledge that a fault has occurred are severe, the clean-up schemes suggested in sections 4.5 and 4.6 may be of value. A user might consider interrupting after the peripheral has moved all its data as part of a scheme to help prevent problems. Another alternative is to have the peripheral set one of the STS[2..0] bits after each DMA. These are decisions the designer of a peripheral and driver must make.

Use of an incorrect control sequence can result in fairly unpleasant results. The note in section 4.3 on loopbacks regarding zero length DMA loopback is a more general concern which has been glossed over in sections 4.5 and 4.6. The assumption that the card is in its reset state after a device clear and a device enable covers this up in those sections.

To avoid this particular problem, a general rule for controlling the 27114 which is worked around in the example of loopback, should be applied. This rule is : Don't change the direction of transfer on the 27114 unless the the FIFO data path is being held cleared either immediately prior to the change or is to be held cleared after the change. This involves the use of the CLF bit in the 27114 CONTROL register.

Examples:

OK:

Don't CHANGE

CURRENT 27114 CONTROL REG: ?0?1 ???? -- CLEAR FIFO set

CURRENT 27114 CONTROL REG: ?1?0 ???? -- CLEAR FIFO
cleared and direction changed

OK:

BOTH CHANGE

CURRENT 27114 CONTROL REG: ?0?0 ???? -- CLEAR FIFO cleared

CURRENT 27114 CONTROL REG: ?1?1 ???? -- CLEAR FIFO set and direction changed

NOT OK:

ONE CHANGES

CURRENT 27114 CONTROL REG: ?0?0 ???? -- CLEAR FIFO cleared

CURRENT 27114 CONTROL REG: ?1?0 ???? -- CLEAR FIFO cleared and direction changed.

```
**************************************************************
*In case of doubt, it should be remembered that the clear    *
*FIFO resets the entire data path and if it is possible,     *
*always program in the following order:                      *
*     * DISABLE FRONTPLANE                                   *
*     * DISABLE BACKPLANE                                    *
*     * DEFINE INPUT OR OUPUT DIRECTION                      *
*     * RESET DATA PATH                                      *
*     * ENABLE DATA PATH                                     *
*     * ENABLE BACKPLANE                                     *
*     * ENABLE FRONTPLANE                                    *
**************************************************************
```

Other control sequence concerns depend on the specific application. Some things to keep in mind:

1. The three control bits that are passed on to the peripheral, CL[2:0] generally have no time relationship to anything going on in a peripheral unless the peripheral design somehow forces it. As though, it would be very irritating to the user if these outputs changes when they are not intended to.

2. Care must be taken with the use of PEN, the enable to the frontplane handshake logic. For example: If PEN is set, enabling the frontplane before direction of transfer (DIR) is changed, the peripheral may see a PCNTL assertion which is not at all intended. This could also happen if both PEN and DIR bits get changed in the same command to the 27114 CONTROL register (due to racing potential on different AFI cards, it may or may not happen at any given instance).

3. It would be extremely nice and appropriate if the sense of DIR does not get changed between back-to-back operations of the same type (input or output). If the frontplane is not disabled between readings (input operations), the continuity of input data is preserved (this may or may not be desirable depending on the data start/stop protocol in a given application).

4.8 27114 to 27114 Connections - Software

The software considerations for a 27114 to 27114 link are slightly different than those for an 27114 to Peripheral device

link if we assume that the peripheral device was designed to behave itself. The 27114 was not designed to behave itself in an 27114 to 27114 link. The basic misbehavior is that both ends of an 27114 to 27114 link want to be the master of the handshakes. Each end tries to "prompt" the other to send data or take data and waits for a response.

The scheme for making them talk to each other anyway is to see to it that the sink end will always start asking for data transfers first. If this is done, it is possible to move data with predictable results despite the "belief" by the 27114 cards at each end that each is in "control". The net result of this trickery is that everything works but data transfers are out of "sync".

The following is a suggested scheme for establishing a link:

1. Through whatever necessary, both ends come to the conclusion that a transfer in a certain direction is to occur. If they need to they can send interrupts followed by an indication of agreement via the CL[2..0] and ST[2..0] lines. I'll leave this up to the creativity and discretion of the writers of the drivers for both ends.

Note: Both ends must do the DMA with the EDG bit in the 27114 control register set to 1 (in case of differential cables).

2. The driver at the receiving end of the link (for this transfer) enables a DMA setting up his card to receive data along with a timeout. Then it lets the other end know that it is ready, with one of the CL[2:0] bits.

3. The driver at the source end of the link (for this transfer) enables a DMA after setting up his card to source data.

Note: For DMA length, perhaps the first word sent can be the length. The receiving end then would simply read the data register for the first word after the 27114 begins responding to SRQ polls. Then set up a DMA of the length required. In this case, the sink card is set up for a DMA with DIR = 1 but does not enable the channel adapter to do the DMA. The driver on the sink end of the transfer just reads the first word from the data path after verifying that the card is responding to SRQ POLLs indicating that data is available. It then goes ahead and programs the channel adapter to do the DMA.

4. The transfer takes place. The channel adapters at both ends see an apparent end to the transfer. Both ends clean up if necessary.

5. PERFORMANCE ISSUES

The following sections are intended to supply a potential user with the information needed in order to predict the performance of the 27114 in terms of transfer rate and the effects of a 27114 on a CIO backplane. It does not provide information on specific implementations of the CIO backplane. Backplane-specific information must be found elsewhere.

5.1 Fundamental Limits to Performance

The fundamental limiters of performance are as follows:

1. Maximum burst data transfer rate of the CIO backplane. This is a function of the channel adapter design. CIO specs a maximum data rate of 5 Mega words per second.

2. Ability of the CIO channel adapter to meet that rate given contention on the bus which the channel adapter resides on. Bursting on CIO backplane is limited to 32 transfers. A poll is always done between bursts.

3. Priority on the backplane and the scheme used to determine the allocation of such bandwidth as exists between device adapters. This depends on channel adapter design, choice of card slot, and the operating system.

4. Overhead on backplane associated with polling. This depends on the design of the poll timing in the channel adapter as well as the service algorithm used by the channel adapter. The service algorithm is typically in channel adapter microcode.

5. Fixed overhead in the frontplane handshake hardware. This time is determined by the design of the 27114.

6. Cable length. The longer the cable, the slower the handshake cycle.

7. Time taken by a peripheral to move data after seeing PCNTL asserted. This depends on the peripheral design.

The limits listed in numbers 1, 2, and 3 are outside the scope of this document with the exception of a description of the general effects. For the details, please consult the CIO standard manuals.

5.2 Frontplane Limits to Performance

The frontplane limits to performance are made up of the cycle time required to access the FIFO and logic delays on the board. The following discussion assumes that the backplane is keeping up with the average transfer rate on the frontplane in a timely fashion e.g. the Frontplane handshake logic is never held off by a shortage of data at the FIFO output or a shortage of space in the FIFO. This is not normally true.

The cycle time for data to move from the 27114 to a peripheral is made up of the events in the sequence below:

1. PCNTL asserted onto cable to indicate to the peripheral that data is available.

2. Propagation delay down the cable to the peripheral + rise time effects on when the transition is seen at the peripheral end.

3. Receiver delay at the peripheral end. (Depends on peripheral design.)

4. Time for the peripheral to act to accept data. (If any hold-off is needed.)

5. Assertion of PFLAG at driver input (On peripheral).

6. Driver delay to PFLAG on cable (On peripheral end).

7. Propagation delay to the 27114 + rise time effects on when the transition is seen at the 27114 end.

8. Receiver delay in PFLAG path at 27114.

9. Time needed in 27114 to move next word onto data lines and cycle PCNTL low then back high with a set up of data prior to PCNTL being asserted again.

10. Driver delay in PCNTL path (on 27114).

The known numbers for the 27114 are:

8 = 21nsec max 16nsec nominal 9 = 210nsec max 183nsec nominal
10 = 15nsec max 10nsec nominal

246nsec max 209nsec nominal

A good cycle time would be in the case of a 3 m cable with a simple device on the end of it. In this case the peripheral simply stuffs the data away as it gets them and feeds PCNTL back as PFLAG. Don't forget that this assumes that the backplane keeps up. The cycle time is represented as follows:

246 + 16(prop delay) + 16 (prop delay) + 21(receiver) + 15(driver)

= 314 nsec or a transfer rate of 6.3 M bytes per second

Nominally, the cycle time would be more like 209 + 16 + 16 + 16 + 10

= 267 nsec or a transfer rate of 7.4 M bytes per second

The same set of numbers for the transfer of data to the 27114 from a peripheral is

8 = 21nsec max 16nsec nominal 9 = 268nsec max 240nsec nominal
10 = 15nsec max 10nsec nominal

304nsec max 266nsec nominal

The same number as calculated above.

304 + 16 + 16 + 21 + 15

= 372 nsec or a transfer rate of 5.3 M bytes per second.

266 + 16 +16 + 16 + 10

= 324 nsec or a transfer rate of 6.1 M bytes per second.

These transfer rates are somewhat high unless the peripheral can truly take the data without spending time processing them.

5.2.1 Cable Length

The cycle time for data transfer on the frontplane is increased with increased cable length. The propagation delay of the cable and the effects of resistive losses in the lines come into play.

The propagation delay in the cable is about 1.8 nsec per foot of cable. This delay in the signal path shows up in the cycle time as length(in feet) * 1.8 * 2. The factor of 2 is because the delay is incurred in PCNTL to the peripheral and PFLAG returning.

The effects of resistive loss in the line show up in two ways. The first is that in long cables (becomes noticeable at > 150 feet), pulses on lines may not even show up at the far end. The second is in skew between signals due to the rise time of the signals.

If the cable is more than about 150 feet long, the rise time and fall times of the signals become significant. The PCNTL line pulses low for some guaranteed minimum time at the 27114 end of the cable and it is up to the system designer or peripheral designer to ensure that this is sufficient to be seen by the peripheral. However, as the cable is lengthened, the rise and

fall times increase due to resistive losses in the lines resulting in narrower and narrower pulses seen at the receiving end of the cable. The only option open to a designer is to set a new time constant on the 27114. The time constants involved are those controlled by R4 through R9 on the board. It is suggested that the designer follow the formula:

***FORMULA

The other problem with long cables is the skew seen due to the rise times. The thresholds on receivers will vary and the rise time of signals in long cables is slow, signals driven at the same time may cross through receiver thresholds at different times. The danger exists that the set up time for data prior to the reception of PCNTL could be affected enough to cause problems. The time constant for the set up of data driven by the 27114 is set with R5/R6. It should be increased in value to increase the set up time seen at the peripheral. The time constant for the de-skewing of received data is controlled by R7/R4. It should be increased if necessary. Note: Both the set-up for driven data and the low pulse width guarantee on PCNTL when driving data out are controlled by the same RC time constant.

5.3 Backplane Performance

CIO backplanes are capable of potentially moving data at a high transfer rate. However, the scheme it uses for multiplexing this resource between specific I/O device adapters and the capability of the specific device adapters makes a large difference in what actual performance is seen. It is also the case that CIO channel adapters are not all capable of the same raw performance.

5.3.1 Data Bursting

The 27114 card is capable of moving bursts of data between the FIFO on it and the backplane. The 27114 will move a burst of data which is as long as it can, somewhere between 1 word and the maximum of 32 (imposed by CIO backplane), any time it responds to an SRQ poll and is granted service by the channel adapter. The size of these bursts is dependent on the rate at which the frontplane is capable of moving data into or out of the 27114 and the rate at which the 27114 is capable of winning data transfer cycles from the channel adapter.

The size of the transfer depends on the number of words (or spaces, depending on transfer direction) become available at the output of the FIFO during a burst. If data is not available at the FIFO output (or space at input) the burst is terminated early by the 27114. It deasserts the BR- control line to the channel adapter. If it ever has to do this, even if data almost immediately becomes available again at the FIFO output, it does not respond to at least the next two SRQ POLL's by the channel adapter. This tends to give the frontplane a chance to catch up while increasing the size of the next burst. It also helps to prevent a particular 27114 from tying up all lower priority cards

on the backplane with short and inefficient transfers. If the frontplane cannot keep up with back-to-back full length bursts, it will let a lower priority card have the opportunity to win an SRQ POLL.

5.3.1.1 Burst Hardware Variation

The goodwill of being "nice" to other cards in the same backplane mentioned in 5.3.1 is due to the fashion in which the 27114 controls Burst Request. This 27114 hardware scheme makes some assumptions that are not strictly called out in the CIO standard.

This "nice" behavior was implemented because of the special requirements on Burst Request. The FIFO used on the 27114 gives no warning prior to being unable to accept more data. Burst Request requires a warning. The hardware has two extra registers on the FIFO input in which the two extra words are stored after the 27114 ends burst request.

The data stored in these two extra registers moves on into the FIFO and subsequently out to the peripheral, only after seeing that there is room in the FIFO and seeing a rising edge of POLL-.

There is a more serious implication for the last two words of a transfer to a peripheral. Part of the time, by accident, it may be the case that the last two words of a transfer hit on a full FIFO condition and are stored in the external registers on the FIFO input. In the case of the "intelligent" channel adapter, the last two words transferred out could be delayed for a while. This is true because a channel adapter is not required to be POLLing all the time. It may not be easy to predict or guarantee that two more POLLs will happen after a transfer without explicitly forcing them in software or depending on the good will of fortune. When and if such channel a channel adapter is built, drivers will have to explicitly force a two POLLs after a DMA to guarantee that the data moves on into the FIFO and out to the peripheral. The channel adapters for the series 500, INDIGO, FIREFOX, and CHEETAH happen to POLL all the time.

The same problem does not apply in movement of data from a peripheral to the 27114. The two polls will always happen in this direction.

5.3.2 Interaction Between Backplane and Frontplane

There are two cases here that are of interest. The first is the case of a frontplane connection which is capable of outrunning the average transfer rate seen on the backplane side. The second is the case of a backplane connection which is capable of outrunning the average transfer rate of the frontplane.

The fast frontplane and slow backplane situation shows up on the backplane as an 27114 card which always responds to SRQ polls and always moves the maximum allowed burst of data when it is

provided service. If it is a high priority card, it will tend to lock out lower priority cards and hog the entire backplane unless some relatively intelligent SRQ service algorithm happens to be implemented in the channel adapter. (No such "intelligent" channel adapters are presently being built or developed.) Keep in mind that unless data is continuously flowing into the machine from somewhere else, this hogging of the backplane bandwidth cannot go on indefinitely.

In the case of the existing IOP on series 500 machines, it will simply starve out all lower priority cards. If it is a low priority card, any time it can win an SRQ poll, it will lock out other cards (higher or lower priority) for the length of time required to do the maximum burst size. Depending on the capability of the higher priority cards with which it is contending it may well starve them, reducing the transfer rate on higher priority cards.

This is definitely the case when competing with a BIC chip based card. The HPIB card can be starved down to about 268 K bytes per second by a lower priority 27114 running at its maximum possible rate. This is only for the time during which DMA's for both cards are enabled and both cards are attempting to move data at the same time. The issue of whether or not a card is ever really starved to the degree that it has noticeable undesirable affects depends on the statistical nature of traffic on that channel adapter. This question becomes important in real-time control.

In this fast frontplane and slow backplane situation, the frontplane will find itself moving bursts of data as it gets them. Looking at the frontplane transfers, one would see bursts of data mirroring the bursts being moved between the channel adapter and the FIFO.

The slow frontplane and fast backplane situation shows up on the backplane as a 27114 taking something less than the maximum burst size in a transfer. It also would not bother to respond to SRQ polls after taking a short transfer until at least two SRQ polls had gone by.

In this case, the frontplane transfers will appear smooth and not reflect the burst nature of transfers on the backplane side of the card (unless the peripheral is designed such that it tends to bunch up transfers).

5.4 27114 to 27114 Performance

The performance of a 27114 to 27114 link will be somewhat less than that which would be achievable with a simple peripheral as described in section 5.2. This is because the FIFO cycle time in a 27114 adds to the cycle time and because the 27114 de-skews data with respect to PFLAG on receiving data. The performance numbers below give numbers for "worst case" and nominal transfer rates pretending that there is no performance limitation imposed by the backplane. Those limitations would include contention on

the backplane with other cards and starvation of the channel adapter on the memory bus. This was calculated for a 2.5 meter cable.

1. PCNTL asserted onto cable to indicate to the peripheral that data is available.

2. Propagation delay down the cable to the peripheral + rise time effects on when the transition is seen at the peripheral end.

3. Receiver delay at the peripheral end.

4. Time for the peripheral to act to accept data (if any hold-off is needed).

5. Assertion of PFLAG at driver input.

6. Driver delay to PFLAG on cable.

7. Propagation delay to the 27114 + rise time effects on when transition is seen at the 27114 end.

8. Receiver delay in PFLAG path at 27114.

9. Time needed in 27114 to move next word onto data lines and cycle PCNTL low then back high with a set up of data prior to PCNTL being asserted again.

10. Driver delay in PCNTL path Source asserts it.

	Max	Nominal	
2 =	16nsec	16nsec	Propagation delay
3 =	21nsec	16nsec	Sink receiver delay
4 =	292nsec	223nsec	Sink eats data
6 =	15nsec	10nsec	Sink de-asserts its PCNTL (Source's PFLAG)
7 =	16nsec	16nsec	Propagation delay
8 =	21nsec	16nsec	Source receiver delay
9 =	210nsec	183nsec	Source coughs up more data
10 =	15nsec	10nsec	Source drives PCNTL

---

max cycle time is 617nsec ==> xfer rate of 3.2 B bytes per second.

nominal cycle time is 490nsec ==> xfer rate of 4.0 M bytes per second.

2 sigma cycle time is 516nsec ==> xfer rate of 3.85 M bytes per second.

6. PRELIMINARY PROGRAMMING INFORMATION

2.1 Description

This section only touches the most basic programming techniques available so far with the series 550 driver running under Unix. It is not intended to be a replacement for the actual driver's ERS.

2.2 Installation

The installation of the AFI driver under Unix is similar to other drivers.

* Retrieve the driver from the supplied tape with the appropriate Unix utility (tar - tape archive e.g.)

* Do an oscp to put the driver in the system boot area. If /dev/rhd contains the system boot area, something like this might do:

  oscp -av afidriv.unix /dev/rhd

  In which afidriv.unix is the name of the driver in the working directory.

* Verify this perhaps with an osck command.

* Reboot the system to let the driver be recognized

To make a device file for an 27114A AFI card on series 550 machine, major device number 44 must be used. For example:

```
/etc/mknod /dev/afi1 c 44 0x010000
```

This would create the device file afi1 for an AFI card at select code 1 on the CIO backplane.

2.3 Driver features

The driver supports common IO like write, read. It defaults to 8 bits unless otherwise set. Special ioctl's are provided:

```
ioctl(f#,0x80044102,&control_value)   :write control lines 0,1,2

ioctl(f#,0x40044101,&status_value)    :return status lines 0,1,2

ioctl(f#,0x80004103,&edge_bit)        :set/reset EDGE bit

io_width_ctl(f#,16)                   :set data width to 16 bits

io_reset(f#)                          :reset data card
```

These and other features of the driver can be explained in more detail by Paul Christofanelli of FSD (hpfclo!paulc).