CONTENTS

27114 IRS

1. INTRODUCTION

This document describes the hardware of the 27114. It is intended to be a reference source for those who will have to manufacture, test, and support the 27114 in the future. It may also be of value to anyone who might want to borrow part of the design. An important piece of background reading is the CIO STANDARD document. Without being familiar with that document a reader will have a very difficult time making sense out of this one. Those interested in programming and using an 27114 card should see the ERS.

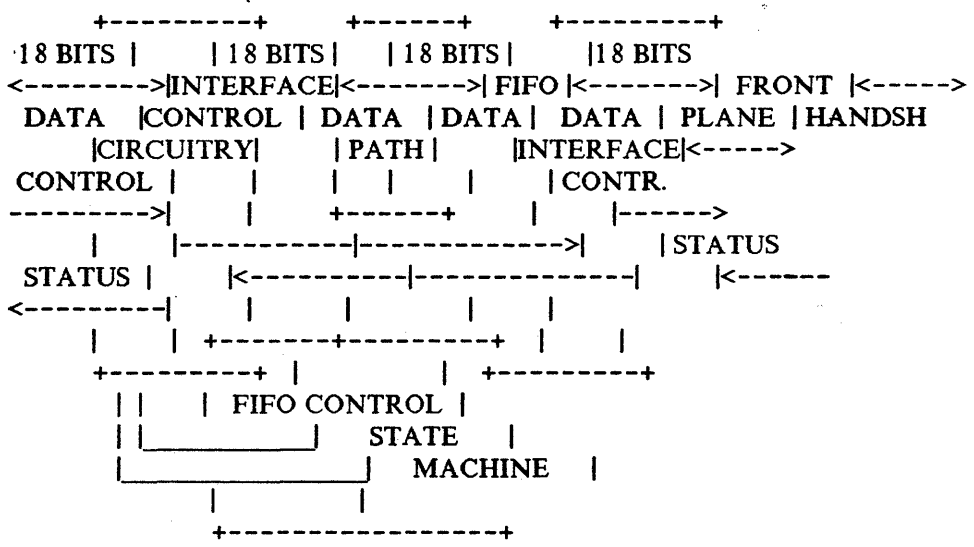
The second section of this document is a general description of how data moves through the 27114 from the CIO backplane to a peripheral or in the other direction. It ignores the practical aspects of programming and the details of the hardware. It is intended to give a reader a feel for how the movement of data takes place without getting bogged down in details.

The hardware implementation is also described in section 2. It is suggested that the reader also reference the 27114 schematics while reading this IRS.

1.1 GENERAL FUNCTIONAL DESCRIPTION

Basic Block Diagram

The four functional blocks of the 27114 are the interface control circuitry, the FIFO data path, the FIFO control state machine, and the peripheral interface. These blocks fit together as shown below.



1.2 BASIC DESCRIPTION:

The 27114A card is a parallel interface card capable of moving data at high speed between any CIO backplane and suitable peripheral devices. The card is a half duplex card, that is it can only either transfer data to or from a peripheral at any point in time. To move the data in the other direction requires the data path to be cleared.

1.3 INTERFACE CONTROL CIRCUITRY

The interface control circuitry is that portion of the 27114 which makes the connection with the CIO backplane. It includes buffers, decoders, and control and status registers. It is what allows a channel adapter, under program control, to control the 27114.

1.4 FIFO DATA PATH.

The FIFO data path is a data pipeline which connects to the interface control circuitry and the peripheral interface. It is controlled by the FIFO control state machine.

The FIFO data path is an asynchronous source and sink for data. Bursts of data timed by the channel adapter are sent to or from the FIFO data path under the control of the FIFO control state machine. The peripheral interface moves data to and from the FIFO at a rate determined by the handshaking going on between the peripheral and some peripheral.

The data path is made of a 18 bit wide by 64 word deep FIFO along with four 18 bit registers in series with the FIFO. Two of these registers are in series with the FIFO input and two are in series with the FIFO output. They are referred to as the INPUT and OUTPUT latches.

These external registers exist to allow the state machine to know it can transfer at least two words of data. This information is necessary in order for the state machine to request a burst transfer.

1.5 FIFO CONTROL STATE MACHINE

The FIFO data path acts as a source or sink for data on the backplane under the control of this state machine. This control is much more complicated than the movement of data between the peripheral and the FIFO data path. This is because there is no nice hold-off mechanism on a CIO backplane. The 27114 must know when it can move a burst of data and ask for it from the backplane, rather than merely asking for it when it can take one word.

1.6 FRONTPLANE INTERFACE.

The peripheral interface consists of drivers, receivers, and handshake circuitry. It performs the task of moving data between the I/O cable and the FIFO. It does this as the master of a handshake between the peripheral and the 27114. As data or space becomes available in the FIFO data path (depending on transfer direction) the peripheral interface asserts a request to the peripheral to move data. When the peripheral has seen this and can do so, it responds.

1.7 DATA FLOW

The basic function of the 27114 is to move data between a CIO backplane and a peripheral device as fast as is possible. It is appropriate at this point to explain in a general way how that data movement occurs.

In order to move data quickly and efficiently between the 27114 and the backplane, the 27114 takes advantage of the data burst capability of the CIO backplane. Unlike some CIO cards which exist, the 27114 is capable of moving up to a full 32 word burst of data each time it receives service if it is ready to do so. This works in both direction of DMA transfer.

The FIFO data path on the 27114 serves as a mindless source or sink of data between the backplane and the peripheral. It is a black box into which data is shoved from one end, when the FIFO says it can take one more word. The FIFO data path is a source of data, when it says it can source one word of data from the other end. The peripheral and backplane asynchronously poke data into and pull data out of the FIFO. The peripheral knows nothing about what the backplane is doing. The backplane knows nothing about what the peripheral is doing. Each end simply does what it must to interface to the FIFO data path.

The peripheral simply asks the peripheral to take data when it becomes available and holds it until the peripheral takes it. When transferring data in the other direction, the peripheral asks the peripheral for data when room in the FIFO exists for more data. Knowledge of the availability of data or room is supplied by the FIFO.

The transfer of data between the backplane and the FIFO data path is controlled by a state machine. The state machine simply waits until there is room to take a burst of data of length equal or greater than one word and begins asking for data by responding to service request (SRQ) polls. If it wins a poll and is given service it asserts the BURST REQUEST line on the backplane only if it can take two or more words. The channel adapter then sources data. The state machine on the 27114 must determine when to shut off the flow of data by de-asserting burst request when it has room for less than two data words. In the case of the transfer from the FIFO to the backplane, it only waits until one piece of data is available before responding to SRQ polls. The state machine would assert the BURST REQUEST line if it has more than one word of data available and de-assert this line otherwise.

Transfers between the 27114 and the backplane will be of some indeterminate burst length depending on the frequency with which the 27114 wins service from the backplane and the rate at which a peripheral device is capable of moving data from or to the 27114. Any time the 27114 must shut off burst request before the channel adapter would, it can not respond to the next two SRQ polls. The rationale is that if the FIFO became completely emptied or completely filled, forcing the shut off of BURST REQUEST, the channel adapter is more than keeping up with the average transfer rate between the peripheral and the 27114. These extra poll cycles are used by the 27114 to sequence some of the data through the FIFO data path. The poll group does not matter in this function.

The use of POLL's to sequence data works differently depending on the direction of transfer.

In the case of transfer to the FIFO from the backplane, the use of POLL- comes into play any time the FIFO fills up. The FIFO fills up and the state machine turns off BURST REQUEST after getting another word of data (if it comes right away). After the channel adapter sees BURST REQUEST it delivers yet another word of data. The state machine in the 27114 stores these extra data in registers outside the FIFO. It then moves the data on into the FIFO on any following rising edge of POLL- which occurs with room available in the FIFO. Until this happens twice to move the extra data into the FIFO, the 27114 will not respond to any more SRQ POLLs. In effect, it saves up its requests until it is capable of at least a two word burst.

In the case of transfer from the FIFO to the backplane, the use of POLL- to move data takes place at the beginning of a transfer or any time the FIFO goes empty. In order to make bursting work on transfers from the FIFO to the backplane, there are two registers on the FIFO outputs for data. After a DMA is enabled, the 27114 moves the first, and perhaps a second word, if it is immediately available into the two external registers. This occurs on rising edges of POLL-. In the case of a second word of data being immediately avaliable, it is possible to do a burst. In the case of only one word being immediately available, the state machine simply moves the one piece of data on without BURST REQUEST.

2. DETAILED FUNCTIONAL DESCRIPTION

2.1 Protection circuit

As shipped from the factory, the +5 power supply is protected with a built-in PC trace type fuse. This fuse is laid out according to current PC layout standard.

When this fuse is blown, the fuse must be replaced with another fuse of part number 2110-0712.

2.2 Data buffering

The CIO data path is 18 bits wide. This includes 16 data lines and two parity lines. (The parity feature is not supported on series 500 machines). All data passed between registers or DMA data path on the 27114 and the CIO data lines goes through three 74ALS245-1's. These drivers are the 48 milliamp version of 74ALS245's. This version is necessary in order to meet the requirements of a fully loaded 16 slot CIO backplane. These chips are in U31, U71 and U81.

On the 27114 side of the data transceivers each of the 18 data lines is pulled up to +5 through a 2.2K ohm resistor. The effect of this is to cause any unused bits in registers read from the 27114 to be seen as being not asserted, given the fact that data on the CIO backplane is low true. The pull ups are in resistor packs RP21 and RP72.

2.3 Address decoding

The 27114 is an EXTENDED CIO card. This implies that the device address is equal to the subchannel address is equal to slot number. The decoding of the device address is done with four open collector exclusive nor gates. Any time the four address lines, AD[3:0] are exactly equal to the four slot address lines, SW[3:0], the voltage on the outputs of the four exclusive nors will be pulled up close to +5. This output feeds into the PAL16L8 used to do both device address decoding and decoding of specific registers on the 27114.

The PAL16L8 is used to decode specific addresses on the 27114 along with write strobes and to generate MYAD. The definition of the inputs and outputs of that PAL follow along with the equations:

INPUTS:

Pin #1 = CEND- -- CEND- backplane line, no buffering

Pin #2 = CBYT- -- CBYT- backplane line. No buffering

Pin #3 = BP0+ -- Buffered and inverted version of BP0-, CIO Backplane line

Pin #4 = BP1+ -- Buffered and inverted version of BP1-, CIO Backplane line

Pin #5 = No connect

Pin #6 = MYPA+ -- When high, indicates that AD[3:0] = SW[3:0]

Pin #7 = DOUT+ -- Buffered and inverted version of DOUT-, CIO Backplane line

Pin #8 = SYNC+ -- Buffered and inverted version of SYNC-, CIO Backplane line

Pin #9 = IOSB+ -- Buffered and inverted version of IOSB-, CIO Backplane line

Pin #11 = HRST- -- Latched hard reset line. Low = reset

OUTPUTS:

Pin #12 = /FR3IN --/(/DOUT*/BP1*/CBYT*CEND*BP0*/HRST*SYNC*MYPA)

Pin #13 = /FR1OUT -- /(DOUT*/BP1*/CBYT*/CEND*BP0*/HRST*IOSB*SYNC*MYPA)

Pin #14 = /FR7IN -- /(/DOUT*/BP1*CBYT*CEND*BP0*/HRST*SYNC*MYPA)

Pin #15 = DATA -- /(BP1 + BP0 + /SYNC + /MYPA)

Pin #16 = MYAD+ -- /(HRST + /MYPA)

Pin #17 = /FR7OUT -- /(DOUT*/BP1*CBYT*CEND*BP0*/HRST*SYNC*IOSB*MYPA)

Pin #18 = /FR1IN -- /(/DOUT*/BP1*/CBYT*/CEND*BP0*SYNC*MYPA*/HRST)

Pin #19 = FR9IN -- /(DOUT + /BP1 + /BP0 + HRST + /SYNC + /MYPA)

Explanation of PAL Logic Equations

MYAD

MYAD is simply a positive true indication that the address asserted by the channel adapter on AD[3:0] = SW[3:0]. This indicates that the 27114 is being address. The /HRST term is included to prevent the backplane line, MYAD- from being driven and prevent the data transceivers from being enabled if the card has been given a hard reset.

/FR1OUT

/FR1OUT is a clock for data being moved into the WRITE CONTROL register on the 27114. The DOUT*/BP1*/CBYT*/CEND*BP0 portion of the product term is true if the register being addressed is the WRITE CONTROL register. The inclusion of SYNC in the product is because SYNC+ being true is part of the definition of a WRITE CONTROL bus operation. The inclusion of MYPA is to guarantee that this card is the one being addressed. /HRST is included to prevent any activity on the bus from affecting the card state if the card is in a hard reset state. The use of IOSB is to generate a clock for data into the WRITE CONTROL register on the 27114.

/FR1IN

/FR1IN is an enable for data to be asserted onto the buffered data bus on the 27114 from the SENSE register during a READ SENSE operation. It is the same as /FR1OUT with the exception that /DOUT replaces DOUT and IOSB is not used.

FR3IN

/FR3IN is the tri-state output enable for a 74LS125A which implements the ID register. When asserted, /FR3IN enables the 74LS125 to drive four bits on the buffered data bus and all the others are passively pulled up. The product term in the tri-state enable is simply the address and the SYNC*MYPA*/HRST term which conditions it.

/FR7OUT

/FR7OUT is the strobe for data being written to the 27114 CONTROL register. It is just like /FR1OUT except for the address.

/FR7IN

/FR7IN is the enable for the data being read from the 27114 STATUS register. It is just like /FR1IN except for the address.

/FR9IN

FR9IN is the enable for the READ STATUS operation. Since the 27114 always returns a STATUS of AES, this output simply pulls /BD4 low on the data bus while the others are passively pulled up. It also clears the ATTN flip-flop.

DATA

DATA is an enable used to allow IOSB to clock the FIFO data path on the 27114. It is the case that the HRST term is not needed simply because, if HRST is high, clocking the FIFO data path does nothing to it.

2.4 Reset circuit

The 27114, after power up or a hard reset done with the /RST line, is latched in its reset state. As long as the 27114 is held latched in this state, the 27114 will not assert any backplane lines. The latched hard reset feeds into the PAL16L8 used as a decoder inhibiting the 27114 from ever seeing itself as addressed.

The hard reset state of the 27114 is unlatched after a CIO WRITE DATA operation is done with /UAD asserted. Once this is done, the 27114 can see itself as being addressed when addressed.

A hard reset puts the 27114 into the state described in the CIO manual. The basic result on the 27114 is to clear the data path, set the 27114 CONTROL register to known values, disable attention requests, disconnect the card from the bus and set the device clear flipflop.

These states, except for interrupt disable, can also be entered at any time with an addressed device clear. This is done with a WRITE CONTROL bus operation with DCL set and DEN not set. To re-enable the card a WRITE CONTROL with DCL cleared and DEN set is necessary. This is called a device enable and is necessary right after power up. In this case, interrupts and enable of interrupts are not changed or prevented. There is no self test of the card as a result of an addressed device clear. This is simply because there is no intelligence on the 27114 to do it.

The addressed device clear or device enable is held in a 74ALS109 which can be seen on page 1 of the schematic. The not-Q output of that flip flop is called /RESET which is distributed about the board.

2.5 Interrupt circuit

Interrupt to the backplane is done via the ARQ line. This line is driven whenever both the ARE (Attention Request Enable) and ATTN (Attention) flipflops are on. The ATTN flipflop is set by the ATTN line of the peripheral interface and cleared by the FR9OUT line.

The ATTN flipflop is implemented with two cross-coupled NAND gates. The effect is that whichever input was last seen asserted would determine the current state of the flipflop. Also, whenever ATTN is asserted, the output of this flipflop is set, overriding the clearing effect of the FR9OUT input. An addressed device clear has the same effect as the FR9OUT input.

The ARE flipflop is implemented with an ALS109 flipflop. The output can be set or cleared with each write to the CONTROL register, depending on the two control bits 0 and 1. The flipflop is cleared whenever a hard reset happens (at power up).

2.6 Basic CIO interface registers

The 27114 is controlled by the channel adapter as a set of addressable registers which are decoded as described in section 2.3. These locations can be associated with specific pieces of hardware on the 27114. All CIO interface registers can be seen on the first page of the schematics.

The WRITE CONTROL register is made up of two 74ALS109 flip flops. These are controlled by bits 5,4,1, and 0. Bits 5 and 4 control the Device Clear flip flop per the description in the CIO standard. Bits 1 and 0 control the ARE flip flop per the description in the CIO standard.

The READ SENSE register is a 74ALS244. When read, the upper 8 bits of the READ SENSE register will be seen as zeros because they are pulled up to +5 through 1K resistors. The content of these bits is described in the ERS for the 27114.

The 27114 CONTROL register is a 74ALS273. All of the bits in this register are cleared upon a hard reset or an addressed device clear (this is equivalent to a write of all 1's to this register). Note that because the CIO data bus is low true, the outputs of the 27114 CONTROL register are the opposite logical sense from what a driver would write.

The 27114 STATUS register is a 74ALS574. This part is clocked by SYNC+ so that the asynchronous inputs to the part do not change during the actual read of the bits.

The id and status registers don't physically exist. When accessed, either of these is implemented as a simple pull down of one of the buffered data lines. It is done with tri-state output control in the decoder PAL16L8. In the ID register, hardware revision number can be programmed by loading the two jumpers JP1 and JP2 in their appropriate positions. For each of these jumpers, one position will program the bit as a 1 and the other position will program the bit as a 0.

The STATUS register always returns a fixed pattern indicating that AES is on.

The data path, as a register location, will be treated separately.

2.7 Poll response and burst request circuit

Poll response circuit responds positively to polls on CIO backplane only if the card is not in its reset state (addressed device clear) and is ready for data transfer. An output from the FIFO state machine PAL called PUAD is used to indicate whether the card is ready for data transfer or not. This PUAD output is directly disabled by the PREN signal from the 27114 CONTROL register.

The AFI card responses to poll service polling for group 0 only (that is both BP0 and BP1 must be low). Some implementation of CIO channel adapter would restrict certain poll group to the lower or higher group of 8 slots. In case of Indigo, group 0 cards can be used in slot 0 to 7 only. Series 550 machine and its extender IO box do not have this restriction since they have only 8 slots.

Burst request also comes directly from the FIFO state machine PAL indicating that the card is ready for burst transfer of data.

2.8 27114 CONTROL and 27114 STATUS register

27114 CONTROL register:

This register is used to control the interfacing with the peripheral. Three outputs are passed on directly to the peripheral interface (CTL0, CTL1 and CTL2). The PEN output is used to disable data handshaking between the 27114 card and the peripheral if set to logic 0.

The CRF (ClearR FIFO) output is used to reset the FIFO and its state machine if it is set to logic 0. The EDGE output is used to select the operating edge of PFLG. When this line set to logic 1, the rising edge of PFLG is used to activate peripheral handshake circuits.

The DIR output controls the direction of data transfer. A logic 1 on this line defines the output mode where data is moved from the CIO backplane to peripherals. The PREN line is used to disable poll responses done by the CIO backplane if set to logic 0.

This register is a write only register and it is implemented with an ALS273 latch. After an addressed device clear, all outputs are cleared to logic 0.

27114 STATUS register:

This register is a read only register and implemented with an ALS574 latch. Its content is updated with every rising edge of the /SYNC line from the CIO backplane. The information latched in this register includes three status lines coming directly from the peripheral interface. In addition to these, a TEST input is used to indicate the presence or absence of the loopback test hood. The presence of the test hood is latched as a false bit in the register. The input and output ready states of the FIFOs are also reported. Finally, the two peripheral handshake lines PFLG and PCTL are included in this register.

2.9 Data path circuit

PARITY:

The LS280 parity generator is used to generate the low order parity bit whenever any register reading is done (except for DATA register). The high order byte's parity bit is pulled high in this case. With the absence of resistor R1, the parity is of the odd type as required by CIO standard. Use of R1 forces even parity to be generated.

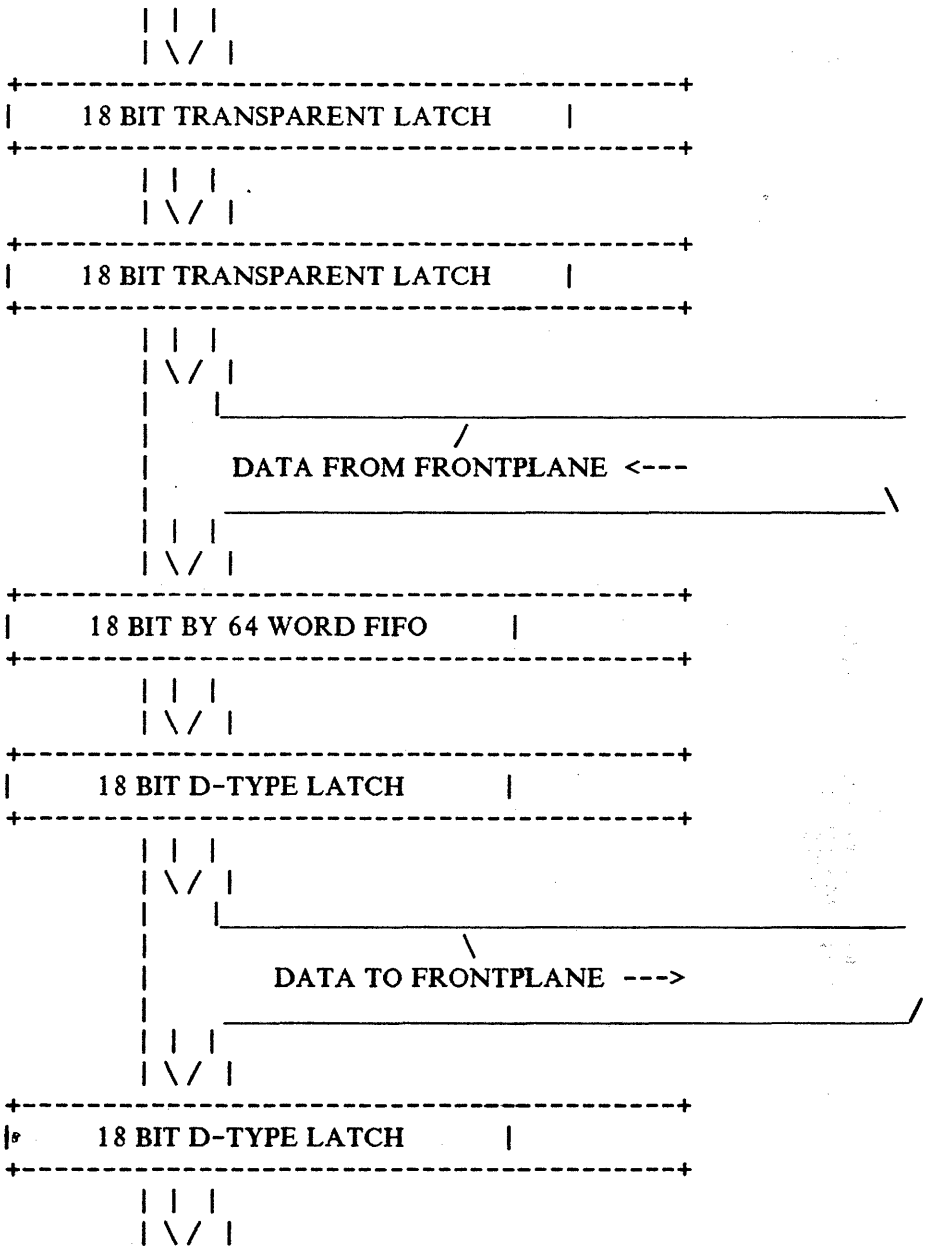
THE FIFO

The FIFO used here is of type 67401, ceramic or plastic package, available only from MMI.

The five FIFO's are arranged to produce a 18 bit wide FIFO with 64 words in depth. In the input path of the FIFO, there are two transparent latches in series. Each 18 bit wide latch is made of a 74ALS573's and a 74ALS841's. In the output path of the FIFO, there are two positive edge triggered latches. Each is made of a 74AS821's and a 74ALS574's. These latches serve to extend the depth of the FIFO by two words on both ends.

The flags on the FIFO, IR and OR, are ANDed and ORed together outside the parts in 74S30's and a 74S260 to provide a set of flags which are meaningful for the entire word width.

The FIFO's used in the DATA path only flow in one direction although the 27114 is capable of moving data either to or from a peripheral. This is accomplished by allowing the buffered CIO data bus to be the source for the input to the transparent latches in the FIFO input path or else allowing the D type latch in the FIFO output path to drive the buffered CIO data bus. The data path between the peripheral and the FIFO is connected at intermediate points in the FIFO data path. See the block diagram on the next page.


18 BIT TRANSPARENT LATCH
18 BIT TRANSPARENT LATCH
DATA FROM FRONTPLANE <---
18 BIT BY 64 WORD FIFO
18 BIT D-TYPE LATCH
DATA TO FRONTPLANE --->
18 BIT D-TYPE LATCH

The input and output of the data path are the same, the 18 bit wide buffered version of the CIO backplane data lines, BD[15:0] and CDP[1:0] The output of the data path can not drive the BD[15:0]/CDP[1:0] lines at the same time as any other device on these lines.

The taps for data to and from the peripheral are located at convenient places in the path for control and access time reasons. The input to the data path from the peripheral receivers is handy for control reasons. When data is being driven by the peripheral receivers, the output of the transparent latch is tri-stated and conversely. There is no potential contention at the point where data is tapped for the peripheral drivers.

2.10 State machine - PAL16R4

The state machine which controls the loading and emptying of the FIFO by the channel adapter is implemented in the PAL16R4 circuit along with some other functions associated with burst request and poll response. Its basic function is to interact with the channel adapter in such a fashion that data gets moved into or our of the FIFO and subsequently to or from a peripheral device. Because of the fashion in which data bursting from or to a CIO channel adapter is turned on and off, the control is somewhat more complicated than might be expected.

At the point in time in which the FIFO becomes full or empty the state machine turns off the flow of data from or to the channel adapter by de-asserting burst request. Unfortunately, due to the definition of the way in which channel adapters shut off, this is too late in time to quit sourcing or sinking data for a channel adapter. This is the motive for the extension of the FIFO with two levels of latches at the input and output. These latches make up a pipeline of data which is explicitly controlled, unlike the inside of the FIFO. This extension of the FIFO makes it possible for the state machine to know when to turn off burst request while it still is able to source or sink more data to fulfill the requirements imposed by a channel adapter.

The state machine, looked at as a simple FIFO controller, is basically two flip flops used as a place holder to indicate which of the external registers, either on the input or output, has data saved in it. Depending on which direction the transfer is going, the indication of data stored external to the FIFO is for either the transparent latches on the input or the edge triggered latches on the output. The meaning of each bit is basically the same in either case.

The flip flops that store this state information are actually included in the PAL16R4 chip with their outputs as Y1 and Y0. From this point on, they will be referred to as simply Y1 and Y0. The meaning of the state of Y1 and Y0 just prior to the time that a new clock to those flip flops occurs is as follows:

Y1	Y0	Meaning
0	0	No data is being held in either of the two latches. This is independent of the direction of transfer.
NEAR 0	1	Data is being held in the latch which is closest to either the input or the output of the FIFO. The question of input or output depends on the direction of transfer.
1	1	Data is being held in both of the latches on either the input or output of the FIFO. The direction of transfer determines the case.
FAR 1	0	Data is being held in the latch furthest from either the input or output of the FIFO. The direction of transfer determines the case.

The state machine is clocked by more than one clock. These clocks do not overlap. The state machine can change to a new state due to either the rising edge of a buffered version of the CIO backplane line, POLL-, or the rising edge of FIOSB-. FIOSB- is a version of the CIO backplane line, IOSB-, conditioned by the DATA output of the PAL address decoder. FIOSB- is a version of IOSB- which only gets through when it is intended to move data to or from the data path on the 27114. These two clocks are ANDed together in the S08 gate outside the PAL16R4. The output of this gate is used to drive both the PAL16R4 and the Sampled Ready (SR) sampling flipflop (via a delay line).

State Machine Inputs

The inputs to the state machine include the following:

Y1 - This is the current state of the Y1 flip flop.

Y1- - This is an inverted version of Y1.

Y0 - This is the current state of the Y0 flip flop.

Y0- - This is an inverted version of Y0.

PT - Poll Trigger. This is high if the next clock edge to be seen is from a POLL-, not from an FIOSB-.

PT- - Inverted version of PT.

SR - Sampled FIFO readiness Flag. This is asynchronized version of either the Input Ready or Output Ready flag from FIFO. If the direction of transfer is from the channel adapter to peripheral, it is a synchronized version of Input Ready. If the direction of transfer is from a peripheral to channel adapter it is a synchronized version of Output Ready.

IR = CAN ACCEPT DATA

OR = CAN SOURCE DATA

SR- - Inverted version of SR.

CLEAR FIFO- - A control line from the 27114 CONTROL register which sets the state to Y1=Y0=0 at the next clock.

DIR - Control line from the 27114 CONTROL register, determines the direction in which the 27114 will attempt to move data. This line is independent of what the DOUT- line on the CIO backplane indicates.

DIR- - Inverted version of DIR.

State Machine Outputs

The outputs of the state machine follow:

Y1 - Next state variable.

Y0 - Next state variable.

SIT - Trigger bit for SI pulse.

SOT- - Trigger bit for SO pulse.

LCK - Clock for registers in serial with FIFO outputs.

LEN - Clock for register in serial with FIFO inputs.

BR- - BACKPLANE line, BURST REQUEST.

The last three outputs are not completely implemented in the PAL16R4.

PAL16R4:

The PAL16R4 is used to implement the majority of the FIFO state machine. The outputs are described in the following equations:

```
IF (VCC) /SIT=/DIR +
         /SR +
         Y0*/PT +
         /Y0*/Y1*PT

IF (VCC) SOT= /DIR*SR*Y0*/PT +
         /DIR*SR*/Y1

IF (VCC) /BR= Y0*/Y1 +
         Y0*DIR +
         /Y0*Y1 +
         Y1*DIR +
         /Y0*/DIR +
         /Y1*/DIR

IF (VCC) /PUAD=PREN*/BP0*/BP1*/Y1*DIR +
         PREN*/BP0*/BP1*Y0*/DIR +
         PREN*/BP0*/BP1*Y1*/DIR +
         /BP0*BP1*AREQ

/Y1:=      /Y1*/Y0 +
         /PT*/Y0 +
         /Y1*PT*DIR +
         /Y0*PT*DIR*SR +
         CFIFO

/Y0:=      /DIR*/PT*/SR +
         DIR*SR*PT +
         DIR*SR*/Y0 +
         /DIR*/SR*/Y1 +
         DIR*PT*/Y0 +
```

* = 'AND' FUNCTION
+ = 'OR' FUNCTION

/Y0*Y1 +
CFIFO

In addition to the DIR, SR and PT inputs as described in the state machine section, the following inputs are also provided:

PREN: Poll response enable, comes from 27114 CONTROL register.

BP0 and BP1: buffered versions of the BP0 and BP1 address line from CIO backplane.

ARE*ARQ: the ANDed result of ARE and ARQ.

All of the above inputs are used to generate the PUAD output which is used to drive the poll response circuit.

The BR (burst request) output is also generated directly by this PAL.

This PAL (as well as the PAL16L8) is the fast 15 ns type and comes preprogrammed by TI.

OUTPUT DATA TRANSFER:

If the 27114 has been appropriately controlled with writes and reads of registers on board, it will be set up for to receive a DMA of data from the backplane. The scheme for doing this is outlined in the ERS.

The description which follows is made assuming that the 27114 starts out from a known state. That state is an empty FIFO and set up for a transfer out to a peripheral.

In this state, Y1 = Y0 = 0. This simply reflects the fact that there is no data in the two registers in series with the FIFO input. Assume for now that the 27114 has not yet been successful in getting service, therefor, the input, PT, is true. The input, SR, is true because the input to the FIFO is ready and this has been clocked into the SR flip flop by a previous rising edge of POLL-.The outputs are as follows;

Y1 = Y0 = 0

SIT = 0

SOT- = 1

LCK = 0 'OUT' LATCH CLOCK

LEN = 1 'IN' LATCH CLOCK

BR- = 0

As a result of the state and the enabling of SRQ POLL responses, the UAD- line is driven low in order to request service during the appropriate SRQ POLL bus operation. The result of this is eventually service to the 27114. This service is in the form of addressing the 27114 with SYNC- asserted and addressing the data path on the 27114. When the 27114 sees itself addressed, it asserts BURST REQUEST.

After this, the channel adapter begins pulsing IOSB- to move data. The pulsing of IOSB- shows up at the state machine as a pulsing of FIOSB- which sets PT = 0 and causes state clocks to occur.
POLL TRIGGER

As long as the FIFO is not allowed to fill up, either because the transfer is short, or the peripheral can keep up, the state machine has a simple task. The state variables remain Y1 = Y0 = 0. When PT is set to 0 by the falling edge of any FIOSB- SIT becomes = 1. As a result, every time a rising edge of FIOSB- occurs a state machine clock creates a shift-in pulse to the FIFO. This pulse is formed with an F74 flip flop, a delay line, and feedback to clear the flip flop. When the Q- output of that flip flop goes low as a result of the clock, SI goes high to move data into the FIFO. After the delay line time the pulse on SI goes away due to the feedback to the clear input of the flip flop. The data at the input to the FIFO is held after the trailing edge of each FIOSB- in the latch furthest from the FIFO inputs. FIOSB+ simply holds it there.

In the case of the FIFO filling up, the state machine gets notice of this event through the SR or SR- inputs. An explanation of what SR does is appropriate at this point. SR stands for sampled readiness. In this case, with data moving from the backplane to the 27114, the readiness is readiness to take more data into the FIFO. The 74S51 selects this on the basis of the DIR control bit.

SR is sampled nominally 150 nanoseconds after a rising edge of the state machine clock. This sampling is necessary because the output from the FIFO, IR(INPUT READY) will change due to the emptying of the FIFO, a set of events which is not synchronized to the filling of the FIFO. This means that the IR line could be changing at precisely the same time as a rising edge of the state machine clock. If this were the case, the next state of the machine is indeterminate. The results might be multiple clocks to shift data into the FIFO and a state machine left in a strange and unintended state. The sampling of IR in the 74S74 which sources SR is intended to provide unambiguous knowledge to the state machine through the input, SR, at the time when a state change decision is made. The input to the SR flip flop is, of course, also asynchronous with respect to the delayed version of the state machine clock. However, the delay from the clocking of the SR flip flop to the next state machine clock is long enough that, with extremely high probability, the output is stable at one level or the other. It doesn't matter which level it is at, as long as all the logic downstream sees the same stable level.

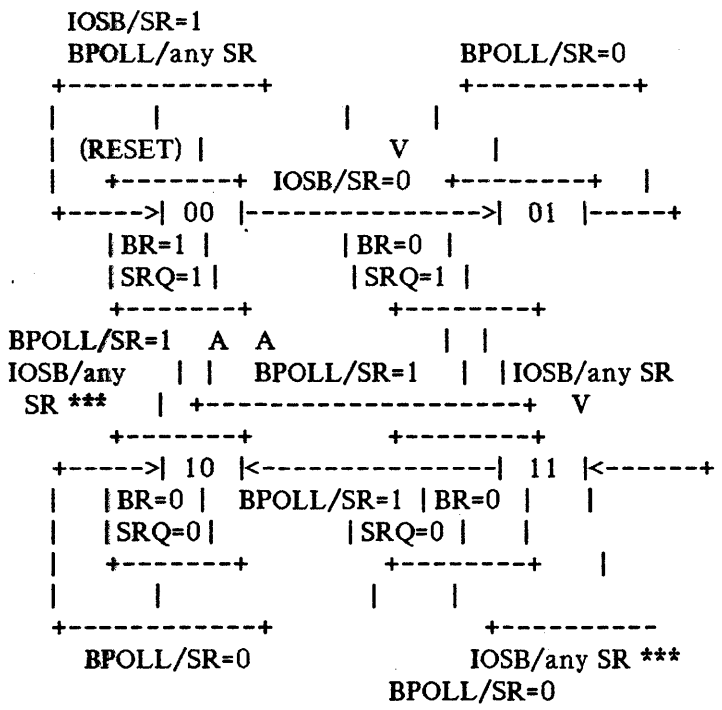
As soon as the state machine is clocked by an FIOSB- sourced state machine clock (PT = 0 is the indication) with SR = 0, the state machine goes to Y1 = 0 and Y0 = 1. The fact that SR = 0 prevents another shift-in of data to the FIFO because it sets SIT = 0 prior to the state machine clock.

When the state machine moves to this state, BURST REQUEST is de-asserted. The data sent on that FIOSB- is saved in the Y0 register because LEN goes to 0.

The channel adapter may or may not source one more word of data. Normally it would, unless the last word transferred happened to coincide with the CIO limit of 32 words transferred in a burst or with the end of a DMA. In either case, the 27114 will never respond to the next SRQ POLL. It will not respond to another SRQ POLL until after all data stored external to the FIFO in the Y1 and Y0 registers is moved on into the FIFO.

Data which gets caught in the external registers in series with the FIFO inputs is moved on into the FIFO by rising edges of POLL-. Data is in those registers because the indication from the FIFO that no more room is available comes too late to shut off the flow of data from the backplane. As soon as a state machine clock is seen with SR = 1, indicating that room for data is now available, and PT = 1, indicating that a POLL- is the cause of the edge, data is moved on into the FIFO. If both Y1 and Y0

registers have data in them, the Y0 data is by necessity moved on the first instance of this. Y1 data is moved on the second instance of this. The sequence would be: ;image Y1 = Y0 = 1 Y1 = 1, Y0 = 0 Y1 = 0, Y0 = 0 Of course, if the FIFO input is not ready, it may take a few polls to move through this sequence of states. Once things get back to Y1 = Y0 = 0, the 27114 again begins responding to SRQ POLLs and when serviced asks for bursts of data.



OUTPUT TRANSFER STATE DIAGRAM

NOTE:
- *** :output data will be lost, other IO's is OK
- Dual clocking sources BPOLL or IOSB

INPUT DATA TRANSFER:

The movement of data from the 27114 to the backplane is somewhat similar to the movement in the opposite direction. The major source of hardware is once again the fact that, in order to ask for a burst, the 27114 must be able to guarantee at least two words of data. This adds the two registers to the output of the FIFO and creates the need for the state machine. The only difference is that the data must be moved out of the FIFO into registers prior to beginning the transfer rather than cleaning up after receiving a transfer in the other direction. This means that if the peripheral keeps up with the backplane the normal state will be Y1 = Y0 = 1 once things get started.

This description makes the assumption that the card starts with Y1 = Y0 = 0 with a DMA enabled, SRQ POLL responses enabled, and no data yet available at the output of the FIFO. The direction control bit, DIR, is set to 0, as seen on the board.

When in this state, the 27114 is not capable of sourcing data to the backplane so it does not respond to SRQ POLLs. It doesn't do anything until some data shows up at the FIFO output. This happens because some time earlier the peripheral managed to convince a peripheral to give it some data. After the fall through time in the FIFO, it showed up at the outputs.

When data shows up at the FIFO outputs, the output ready flag on each of the five FIFO's goes high and this produces a 1 at the D input to the SR flip flop. As soon as a rising edge of a POLL- causes that flip flop to be clocked, SR goes high. The next time a rising edge on POLL- is seen, it causes a shift-out pulse and moves the state to Y1 = 0 and Y0 = 1. This also forces a clock on LCK to put the data at the FIFO output into the register associated with Y0.

If, after this shift-out, another word of data is immediately available, the delayed version of the state machine clock edge which performed the shift-out will set SR to 1 again. If this is not the case, SR will be set to zero on this edge. If more data is immediately available, a burst will eventually happen. If not, a single word will be moved. This insures that a single word transfer will work.

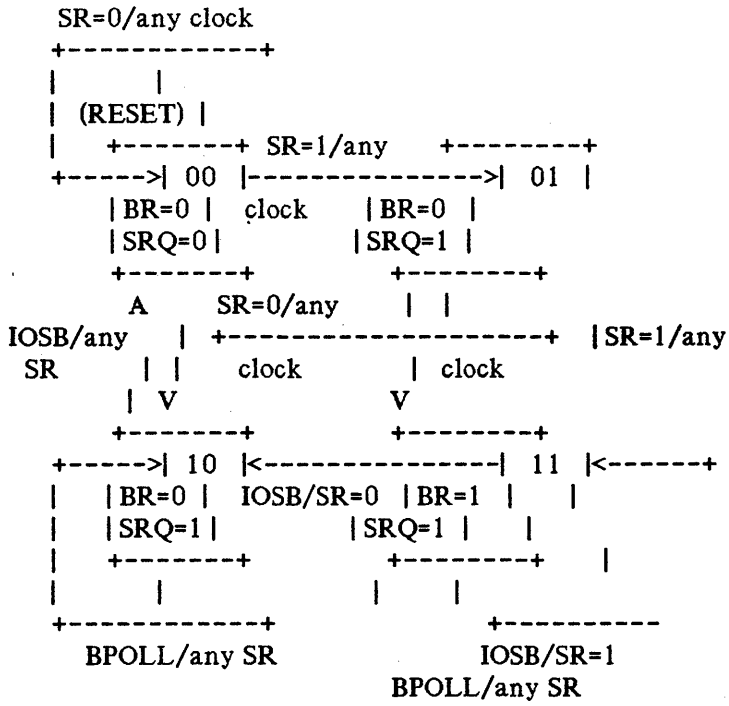
Note that SRQ response is positive in this state (01) eventhough the far latch does not contain valid data. The reason is that when this output is known to the channel adapter in a poll transaction, the poll itself would push the state machine into either state 11 or 10, depending on whether the FIFO has more data or not (respectively).

In the case of more data being available immediately, the next rising edge of POLL- will cause a state machine clock which causes a shift-out and sets the state to Y1 = Y0 = 1. LCK occurs at the same time and shifts the data at the FIFO output into the registers. Once Y1 = 1, the 27114 will respond to the next SRQ POLL, requesting service.

When the 27114 receives service with Y1 = Y0 = 1, it will assert BURST REQUEST and move data to the backplane in bursts as long as it can. As long as every time an FIOSB- sourced (PT = 0) state machine clock occurs with SR = 1, the state stays Y1 = Y0 = 1, a shift out occurs, and LCK moves that data one step through the two registers on the FIFO output.

As soon as it is the case that SR = 0 on a state machine clock, the next state will be Y1 = 1, Y0 = 0. The data is sequenced exactly as in the case of SR = 1 as described above with the exception that no shift-out takes place. This state is exactly that of the case mentioned above in which a second word of data was not immediately available. The ability to respond to SRQ POLLs is still there but BURST REQUEST is off in this state. The backplane either takes the last word in register Y1 at the tail end of a current burst, or it is delayed and taken as a single word during the next data operation.

As the single word is taken, the state machine goes to state Y1 = Y0 = 0 and the whole sequence starts over again. The external registers must once again be filled by state machine clocks generated by rising edges of POLL- as data becomes available at the FIFO outputs.



INPUT TRANSFER STATE DIAGRAM

NOTE:
- Dual clocking sources BPOLL or IOSB
- Input data valid only if SRQ=1 (positive poll response)

This section is a description of the logic which makes up the FIFO control state machine. It is a mess of logic equations along with an explanation of each. It is not obvious, looking at each equation by itself, how they work together to produce the results described above.

Next Y1 = /(/Y1*/Y0 + /PT*/Y0 + /Y1*PT*DIR + DIR*PT*SR*/Y0)

/Y1*/Y0 : Data never gets into the register furthest from the FIFO input or output first. It is necessary to put data into closest register first, Y0.

/PT*/Y0 : Data never moves to register furthest from FIFO inputs or outputs if none is in Y0 and clock is from an FIOSB- edge.

/Y1*PT*DIR : When moving data to FIFO, data can't be moved into Y1 as a result of a clock caused by a POLL- rising edge.

DIR*PT*SR*/Y0 : When moving data to FIFO, data can't be held in Y1 as a result of a clock caused by a POLL- rising edge if the FIFO input is ready and Y0 is empty.

Next Y0 = /(/DIR*/PT*/SR + (PT+/Y0)*DIR*SR + /Y1*/SR*/DIR + (DIR*PT*/Y0 + /Y0*Y1))

/DIR*/PT*/SR : When moving data from FIFO, after an FIOSB- with no more data at FIFO output, there will be no data in register closest to FIFO outputs.

(PT+/Y0)*DIR*SR : When moving data to FIFO, if FIFO input is ready, there will never be data left in register nearest FIFO inputs after a FIOSB- if none was already there and after a POLL- whether any was already there or not.

/Y1*/SR*/DIR : When moving data from FIFO, if no data is available at the FIFO outputs, and no data exists in register furthest from FIFO outputs, data will not be in register nearest FIFO outputs after a state machine clock. It either wasn't there to start with or it is moved on to the register furthest from the FIFO outputs. /DIR*PT*/Y0 : When moving data to FIFO, if there is nothing in register nearest FIFO inputs, a POLL- caused state machine clock will never result in data being put there.

Y1*/Y0 : If there is data in register furthest from FIFO and none in closest, it is never the case that data is put into register closest to FIFO on a state machine clock. Direction of transfer does not matter.

SIT = /(/DIR + /SR + Y0*/PT + /Y0*/Y1*PT)

/DIR : When moving data from FIFO to backplane, it is not acceptable to put data into the FIFO.

/SR : When the FIFO isn't ready, it is not acceptable to try to put data into the FIFO.

Y0*/PT : When data is being held in the external register nearest the input of the FIFO, if the state machine clock is from an FIOSB- the data is not to go into the FIFO.

/Y0*/Y1*PT : If there is no data in the two external registers on the FIFO inputs a rising edge on POLL- never causes a shift-in pulse to the FIFO.

/SOT = DIR + /SR + /Y0*Y1 + PT*Y1

DIR : If the direction of transfer is from the backplane to the FIFO it is not acceptable to do a shift-out of the FIFO.

/SR : If the FIFO is not ready with data at the outputs, it is not acceptable to do a shift-out of the FIFO.

/Y0*Y1 : If there is data in the register furthest from the FIFO outputs but none in the register closest, no shift-out is to occur.

PT*Y1 : If there is data in the register furthest from the FIFO and the clock is from a POLL- rising edge, no shift-out is allowed.

LCK = /Y0*Y1*/DIR + SO

/Y0*Y1*/DIR : If data is being moved from the FIFO to the backplane, and the current state is Y0*/Y1, the next state will be /Y0*Y1 or it will be Y0*Y1. This term forces it to be true when moving from Y0*/Y1 to /Y0*Y1. LCK will go from 0 to 1 on that state transition moving data up one place in the registers on the FIFO output from Y0 to Y1.

SO : Any time a shift-out pulse is applied to the FIFO outputs, it is because that data is to move out into the register nearest the FIFO outputs. Whatever happens to already be there will, of course be moved on into the register furthest from the FIFO outputs, whether it is garbage or useful data.

LEN = /(Y1*/BPOLL) */Y0

= (/Y1 + BPOLL) */Y0

= /Y1*/Y0 + BPOLL*/Y0

/Y1*/Y0 : If nothing is to be held in either Y1 or Y0, the Y0 latch is to be transparent.

BPOLL*/Y0 : If nothing is to be held in the latch closer to the FIFO input, there may be something in the latch further from from the FIFO input. In this case, the latch closer to the FIFO sees its control input go high then low during a POLL and as a result copies in the data in the latch further from the FIFO inputs. inlft 8;inrht 8;format;center 1

NOTE

It might seem like there could be a hold time problem at the FIFO inputs. There is not. If data is in both registers prior to moving it on into the FIFO, then the fact that /Y1 = 0 keeps the Y0 data latched after a shift-in on the FIFO. If there is only data in the latch nearer the FIFO, then there is a copy of it in the latch further from the FIFO, the LEN can wiggle and simply pass another copy of the same thing through. Unless the 27114 asks for more data or is abused by a driver, the data in the register further from the card will stay there forever. ;inlft 0;inrht 0;image BR- = IF(SMYAD) THEN /(Y1*Y0*/DIR + /Y0*/Y1*DIR)

IF(SMYAD) : A tri-state control which allows the 27114 to assert burst request only when it is addressed and SYNC- is asserted.

Y1*Y0*/DIR : When transferring data from FIFO to backplane, only assert BURST REQUEST if two or more words of data will be immediately available. Note; this means BURST REQUEST will be turned off after the next to the last piece of data is taken due to the state change to Y1 = 1 and Y0 = 0.

/Y0*/Y1*DIR : Same situation when transferring in opposite direction except the critical commodity here is space for data rather than data. (Holes and Electrons?)

/UAD = /(/RESET*BPOLL)*((/BP1*/BP0*PREN*(DIR*/Y1 + Y0*/DIR + Y1*/DIR)

+ ARE*arq*BP1*/BP0))

/RESET*BPOLL : Channel adapter must be polling and 27114 not reset

/BP1*/BP0*PREN : The terms ANDed with this are terms used to generate SRQ POLL responses based on the FIFO control state machine state. These bits see to it that the POLL is a group 0 SRQ POLL and that the card has been explicitly enabled to respond to SRQ POLLS.

DIR*/Y1 : As long as room for one more word exists on board, the 27114 asks for it.

Y0*/DIR : As long as at least one word is going to be available after the current assertion of POLL-, the 27114 asks to send it.

Y1*/DIR : As long as at least one word is currently available, the 27114 asks to send it.

ARE*arq*BP1*/BP0 : IF attention requests are enabled and the an asynchronous event has been detected, 27114 can respond to a group 0 ARQ poll.

2.11 Peripheral line interface Peripheral Interface

The peripheral interface connects the inputs and outputs of the FIFO data path to drivers and receivers on the cable. It generates control signals to the FIFO to move data in sync with handshakes it generates between itself and a peripheral device. It drives control lines on the cable and receives status lines on the cable. It receives interrupts from the peripheral.

The drivers used on the 27114 are MC3487's. The receivers are AM26LS32B's.

In differential line mode, input lines are terminated in 120 ohms on the 27114 card. Output lines should be terminated on the receiving end. In single ended line mode, input lines are biased to 2.9 volts with a 470 ohm/330 ohm bridge, yielding an effective 193 ohms. The referenced inputs are biased to 1.5 volts with a 1.5K/3.3K bridge each.

The receivers used on the 27114, AM26LS32B's have a built-in feature, such that if the input is not driven, they produce a 1 at the output.

Three of the pins on the peripheral connector are logic ground pins. These would normally be connected to logic ground in a peripheral in an attempt to keep the common mode voltage difference between the ends of the cable from exceeding the specifications of the drivers and receivers. However, a particular application in a noisy environment might benefit from some play with the idea of not connecting these.

CONTROL BITS

There are three control bits driven to a peripheral. These are simply bits in the 27114 CONTROL register which can be written under program control. These as always enabled to drive the peripheral connector pins. Whatever the driver chooses to write to them will simply be asserted on pins on the peripheral connector.

STATUS BITS

There are three status bits received by the 27114. They are simply passed on to the 27114 STATUS register where they are synchronized by SYNC+. These can be read by a driver, if it is of value.

ATTN. The ATTN signal is received by the 27114 and used to clock a flip flop. The receiver is intentionally wired up backwards so that if ATTN+ and ATTN- are not driven, it will produce an output as though ATTN were not asserted.

If the ATTN flip flop is not being held reset at the time of the assertion of the ATTN signal, it will be set. The arq- output of this flip flop is the input to the READ SENSE register. The arq output of this flip flop, if attention requests are enabled, forces ARQ- low on the backplane and allows the 27114 to respond to subsequent ARQ polls. A reset, DCL, or READ STATUS operation will clear the ATTN flip flop.

PDIR. This is simply the DIR bit from the 27114 CONTROL register driven onto a pair of pins on the peripheral connector. The logic sense is such that if PDIR+ is high and PDIR- is low it means that data transfer is to be from the 27114 to a PERIPHERAL.

DATA LINES

Data is moved to and from the peripheral in 18 bit words transmitted and received on 18 pairs of pins. This is a bi-directional data bus. Each end is always either sending or receiving data. It better not be the case that both ends are attempting to send data at the same time.

PFLAG AND PCNTL. These two lines form the hand shake for the transfer of data between the 27114 and a peripheral device. PCNTL is driven by the 27114 to ask for a transfer of data. PFLAG is received by the 27114 from the peripheral when the peripheral acknowledges data. The receiver for the PFLAG signal is hooked up such that it will be seen as not asserted if no peripheral is driving the line.

2.12 Input handshake circuit

In the beginning, PFLAG = 0, PCNTL = 0, PEN = 0, EDGE = 0, DIR = 0, /CLRFIFO = 1, /RESET = 1, /IR = 0, /IRL = 0. The DMA is set up from the FIFO to the backplane and the channel adapter programmed to take it. No data exists in the FIFO because the peripheral hasn't put any there yet. As a consequence, the interface to the backplane must ignore the SRQ POLLs going by.

The trigger event is the writing of PEN to force it to a 1 as seen on the board, by the driver. This event allows the peripheral to begin taking data from the peripheral. At this point in time, the SI (shift in) PULSING flipflop is high at the /Q output.

Because the five IR inputs to the S30 have been high from the reset event, the output of this gate is low and as a consequence PCNTL is asserted, asking for data. The 27114 is ready and willing to take data and asking for it.

After PCNTL is asserted, eventually, the peripheral will chose to assert data onto the cable and assert PFLAG to indicate it has done so. The PFLAG shows up at the input to an XOR gate along with /EDGE. The result is a change from high to low at the output of the XOR. This edge is delayed to allow for the data to become set up at the FIFO inputs and allow for de-skewing between the data and the received PFLAG edge.

The result of this delayed edge is the clocking of PCNTL into the SI PULSING flip flop. PCNTL at this time is a 1. The pulse needed at /PSI is created by feedback from /IRL. This does not happen until all five FIFO chips'input ready come low. The result of a low pulse on /PSI is a high pulse on /IR and /IRL. As soon as /IRL goes high, it clears out the SI PULSING flip flop. Also, /IR goes high it forces the output of the 74S30 to go high killing PCNTL. The delayed trailing edge of PSI keeps it deasserted for at least the minimum guaranteed time. This gives the FIFOs enough time to report its input readiness status. At the end of this time, PCTL is asserted only if all FIFO chips'IR are asserted. The cycle then repeats.

It is the case, when transferring in this direction, that PCNTL will be asserted again after the peripheral sends what it thinks to be or should be its last word. The 27114 simply keeps trying to take data because there is room in the FIFO.

FALLING EDGE TRANSFER MODE

If the driver uses falling edge data transfer mode, it will be the case that in the descriptions of transfers given above that the signal, EDGE, will be = 1 throughout. This causes the falling, or de-asserted edge, of PFLAG to be the edge used to move data or aknowledge its movement.

MEANING OF PFLG: to clear out the confusion, PFLAG as described here means the PFLG line that is connected to the 27114 STATUS register. A 0 on this line corresponding to no connection to the PFLG+/PFLG- line or that PFLG+ is at a positive voltage compared to PFLG- line.

In both input and output mode, timing will essentially remain the same. Since the EDGE line is now at 1, the PFLG's phase must be off 180 degrees to yield the same result.

In both cases, PCNTL does not go away until after the trailing edge of PFLAG.

2.13 Output handshake circuit

The data transfers described assume that some clever driver writer has carefully read the ERS and set up the transfers via whatever wiggling of the conrol, status, and interrupt lines necessary. The description is from the point of both ends set to do the same transfer and then being started. The transfers described are for the case of the bit in the 27114 CONTROL register which controls which edge of a PFLAG moves data having been set to the rising edge. The falling edge operation is very similar with only PFLAG to be 180 degrees off in phase.

In the beginning, PFLAG as seen at the output of the S240, is low. PEN, peripheral enable is high. DIR is high. /RESET and /CLRFIFO are high. PCNTL, as seen at the input to the MC3487 driver. /OR = 1. EDGE = 0. The 74S74 used to detect the rising edge of EDGE XOR PFLAG has a 1 at the Q output. This flipflop is called ACKNOWLEDGEMENT FF. The SO (shift out) PULSING FF (also 74S74) used to produce the /PSO (peripheral shift-out) pulse has a 1 at the /Q output. The 74S74 with an output driving the S00 which produces PCNTL has a 1 at the /Q output (this is the CTL (control) HOLDING FF).

After the above set of circumstances has been made to be true, the FIFO starts being filled from the other end by the channel adapter under the control of the state machine. When the first one of those words finally falls through the FIFO to the output, the peripheral interface receive notice becaues /OR goes low. This immediately causes a /PSO pulse low because it clocks the SO PULSING flipflop. The pulse duration is controlled by the delay line in the feed back path to the clear input of that flip flop.

The pulsed outputs of the SO PULSING FF has three other effects. It clears out the ACKNOWLEDGEMENT flipflop which in turn allows the CTL HOLDING flipflop to be clocked with a 1. After a delay from the trailing edge of the pulse, the pulsed output of SO PULSING clocks the CTL HOLDING flipflop. It also does a shift-out of data from the FIFO resulting in that data being clocked into the register nearest the output of the FIFO.

The delay before the PCNTL flip flop is set allows for the data to move from the output of the register on the FIFO output, through the MC3487 data drivers and onto the cable prior to the first assertion of PCNTL. It provides set up time on the data to the peripheral. The card timing is such that the set up time can be guaranteed to be more than 50 nsec at the 27114 end of the cable. It also guarantees a minimum de-asserted pulse time on PCNTL of 65 nsec.

The dual use of the register on the output of the FIFO's is a scheme to improve the transfer rate on the peripheral. It allows for most of the access time of of the FIFO to be overlapped with the time needed for handshakes with the peripheral. It costs nothing in terms of additional hardware.

After PCNTL has been asserted by the 27114, some time later the peripheral sees it and some time later still, it chooses to do something about it. Eventually the peripheral asserts PFLAG in order to tell the 27114 that it has taken the data and it can be change.

When PFLAG is asserted, it shows up as and edge at the clock input of the ACKNOWLEDGEMENT FF and clocks a 0 to the /Q output of this flipflop, which quickly clears the PCTL HOLDING FF which in turn de-asserts the PCNTL line.

As soon as both the ACKNOWLEDGEMENT flipflop is set and the FIFO output is ready as indicated by /OR = 0, a rising edge will be seen at the clock input of the /PSO flip flop. The result is a shift-out of the second word of data into the register on the output, and a basic repeat of the cycle. It just goes on and on until the transfer is finished. Note that when the transfer if finished, if both ends of the transferred agreed on transfer count, the edge detecting flip flop will be stuck with Q = 1 forcing PCNTL low. No subsequent /OR = 0 will occur to change this. Things are set up for a new transfer.

2.14 Loopback test hood

The loopback test hood is supplied with every 27114 card. It is used to allow the testing of most line drivers and receivers on this card. Control lines are wired to related status lines, data out lines are connected to data in lines. The test hood also grounds the TEST pin of the peripheral connector. This can be seen by the program by reading bit 3 of the 27114 STATUS register. Once this TEST pin is asserted, the data out and parity out line drivers are always enabled. This allows the loopback of data via the test hood since normally the data line drivers are disabled whenever the card is in the input mode. By simulating a PFLG edge using the EDGE bit, the content of the near OUTPUT latch can be clocked into the FIFO via the line driver-test hood-line receiver path.

Wiring of loopback test hood:

PCTL ----------> PFLG
PDIR ----------> STS2
CTL2 ----------> ATTN
DATAOUT----------> DATAIN (16 pairs)
PARITYOUT----------> PARITYIN (2 pairs)
CTL0----------> STS0
CTL1----------> STS1
TEST----------> GND

2.15 Differential cable and mating connector

The peripheral connector is a male (pin) Eurocard connector with all 96 positions loaded. Of these pins, 92 are used for control and data lines. Each signal line requires two complementary line for differential interface. These are: 16 data in, 16 data out, 2 parity in, 2 parity out, 3 control, 3 status, ATTN, PDIR, PCTL and PFLG. Three pins are for logic ground; one pin is for the loopback test hood.

The differential cable is made out of stranded 30 gage (AWG) conductors. All conductors are paired as twisted lines with one conductor being black and the other one white. There are both braid and foil shielding. The braid shield is terminate at both ends to the mold-in bimp. The cable is terminated at both ends with a female (socket) Eurocard 96 position connector. The connector wiring is not 1 to 1 between the two end connectors. Instead, the wiring is done such that the cable can be used to connect two 27114 cable together as described by the ERS.

A wire-wrap matching connector is supplied to the customer to facilitate the hook-up to his/her equipment.

The loopback test hood can also be used to test the differential cable.