

**Fermilab**

Particle Physics Division/CDF Upgrade Project

**DRAFT**

**Specification for  
XFT Finder Modules  
SL1/3 Module  
and  
SL2/4 Module**

T. Shaw, J. Freeman, S. Holm, T. Wesson  
Fermi National Accelerator

R. Hughes, P. Koehn, B. Winer  
Ohio State University

## Documentation History

original	TMS	2/19/97	
	TMS	7/8/97	Update the backplane pinout to include two additional wires for neighbor sharing
	TMS	8/4/97	Work on input/alignment section; update block diagrams
	TMS	8/22/97	Adjust Memory Map
	TMS	10/7/97	Adjust Memory Map due to hardware changes
	TMS	11/12/97	Adjust Memory Map
	TMS	12/1/97	Adjust Memory Map
	SH	1/9/98	Corrections
	TMS	2/4/98	Adjust Memory Map
	TMS	2/23/98	Adjust Memory Map of Finder Diagnostic Input RAM
	SH	3/5/98	Adjust Memory Map - Control Register
	TMS	4/1/98	Adjust Memory Map - Finder Depth Registers
	SH	6/15/98	Added Pixel Driver Segment Pipeline depth registers
	TMS	6/15/98	Revise Memory Map/Control Register for Production version of board; primary difference allows for Flash RAM Programming of all FPGA designs
	TMS	6/17/98	General Edit
	TMS	7/16/98	Added bit description of Pixel Driver Diagnostic RAM
	TMS	7/31/98	Mod Control and Error Register definitions
	SH	8/28/98	Added bit descriptions to memory map
	SH	10/8/98	Added bit 17 in control register.
	SH	3/10/99	Added text to control register & reset definition.
	SH	3/29/99	Added text to Memory Map.
	SH	4/7/99	Edited the Finder Chip information.
	SH	4/8/99	Edited the Finder & Pixel Chip information.
	SH	4/9/99	Edited the Finder & Pixel Chip information.
	SH	4/27/99	Edited Block Diagrams.
	SH	4/28/99	General Edit.
	SH	5/4/99	General Edit.
	SH	7/27/99	Added text to Memory Map.
	SH	7/29/99	Added Text to Input Alignment section.
	SH	8/4/99	Added Text to Input Alignment section
	SH	11/5/99	Changed Transition board text
	SH	9/6/00	Added text to pipeline registers description

## XFT Hardware Overview

The purpose of the eXtremely Fast Tracker (XFT) [1] is to quickly find and identify tracks in the Central Open cell Tracker (COT) [2] which can be used by the Level 2 Trigger [3]. The XFT hardware design consists of several pieces. These pieces are as follows:

- COT TDC Mezzanine Module [4]

The COT TDC Mezzanine Module will determine prompt and delay hits and sends the data to the COT TDC Transition Module.

- COT TDC Transition Module [5]

The COT TDC Transition Module will drive the data onto Ansley cables, which will carry the data to the Finder Crates.

- Finder Transition Modules [6]

The Finder Transition Modules will receive the data coming from the COT TDC Transition Modules and drive it into a Finder module and, in the case of “neighbor” information, across one slot on the Finder Custom J3 Backplane.

- Finder Custom J3 Backplane

The Finder Crate requires a custom J3 backplane to enable the sharing of necessary neighbor information across Finder Module boundaries.

- Finder Modules

The Finder Modules will find track segments, which span a 15-degree slice of a superlayer and report these segments to the Linker Modules.

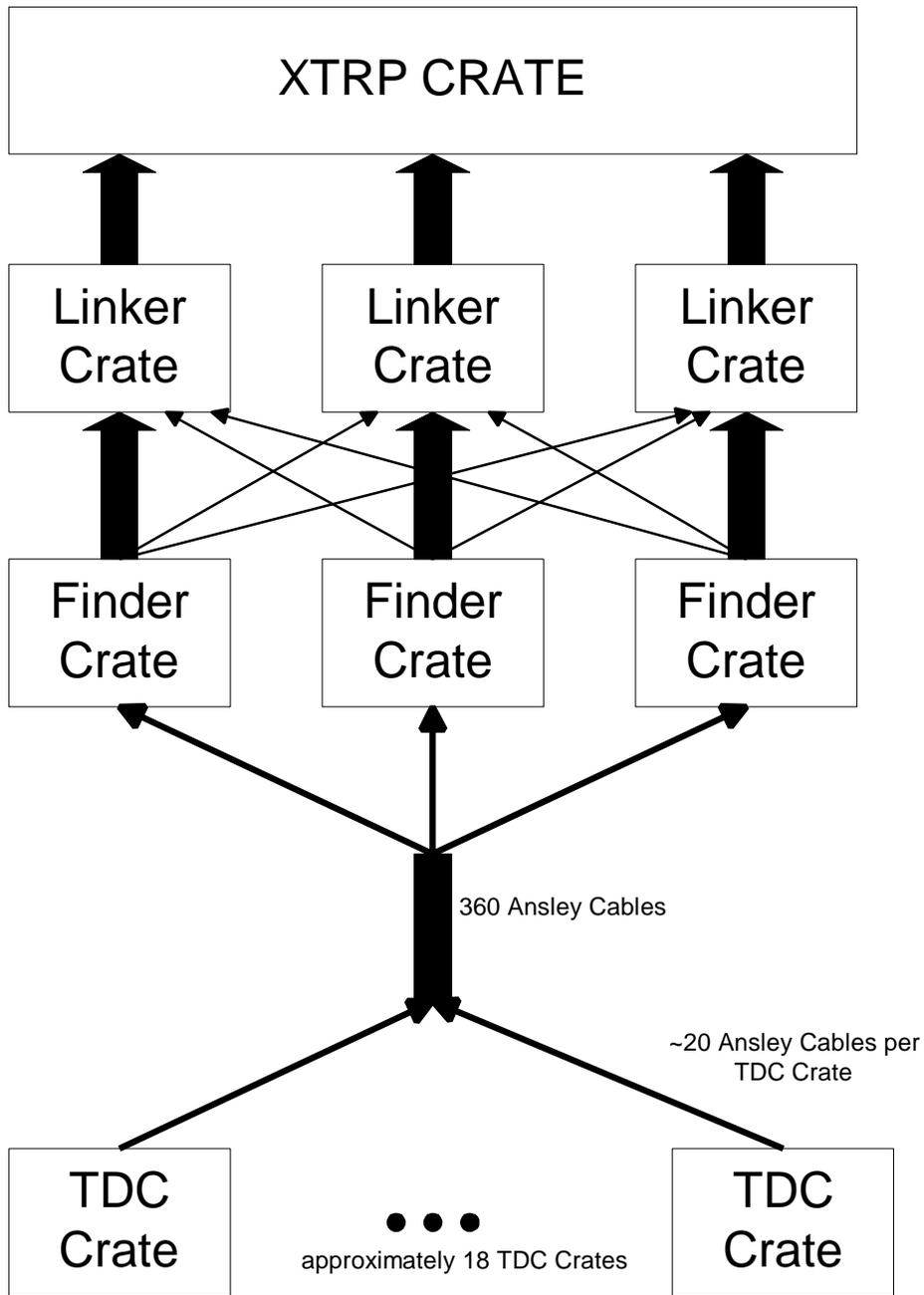
- Linker Modules [7]

The Linker Modules receive segment information from the Finder Modules and look for tracks crossing all four (or three out of 4) superlayers.

- Linker Transition Modules [8]

The Linker Transition Module will drive track information to the XTRP [9].

Figure 1 shows the general flow of the XFT data from the COT TDCs to the XTRP.



Each Ansley Cable will carry the prompt and delay hit information from 48 TDC Channels every 132ns, or at a rate of 45.5Mhz

Figure 1. Data Flow for XFT

## Finder Main Modules

Architecturally, the Finder Modules have been broken down into two types of modules. Each type of Finder Module will span 15 degrees of the COT. The SL1/3 Finder module will contain the logic for dealing with COT Superlayers 1 and 3. The SL2/4 Finder, will contain the logic for dealing with COT Superlayers 2 and 4. Figures 2 and 3 are block diagrams of Finder SL1/3 and Finder SL2/4, respectively.

The main logic of these modules will reside in the Finder circuits. Superlayers 1, 2, 3 and 4 will be instrumented with 48, 72, 96 and 120 Finder circuits respectively (see Figure 4). Each Finder circuit is implemented in a FPGA device. Each Finder FPGA receives its core input from a single Ansley cable and some additional information from “neighbor” cables. The Finder FPGA outputs track segment information which must be passed to the Linker Modules. The Finder FPGAs will be discussed later in more detail.

A SL1/3 Finder Module will contain two SL1 Finder FPGAs and four SL3 FPGAs. There will be a total of 24 of these modules, which will be spread out over 3 crates, with a total of 8 per crate. The custom J3 backplane, which is being designed, requires that SL1/3 Finder modules be located in slots 4-11. These cards are being built with the assumption that a future upgrade will cause the wires of SL1 to be split and cause the number of SL1 Ansley cables into the module to double, which will increase the number of Finder FPGAs by two.

A SL2/4 Finder Module will contain three SL2 Finder FPGAs and five SL4 FPGAs. There will be a total of 24 of these modules, which will be spread out over the three Finder crates for a total of 8 per crate. The custom J3 backplane being designed for the Finder crate requires those SL2/4 Finder modules be located in slots 13-20.

*Throughout this document, the following nomenclature will be used to describe Ansley Cables and Finder FPGAs:*

*+N will refer to the nearest neighbor in increasing Phi*

*-N will refer to the nearest neighbor in decreasing Phi*

*A, B, C, D, E, ... represent cables/Finders in increasing Phi*

*A' - indicates the possibility of a split wire for SL1*

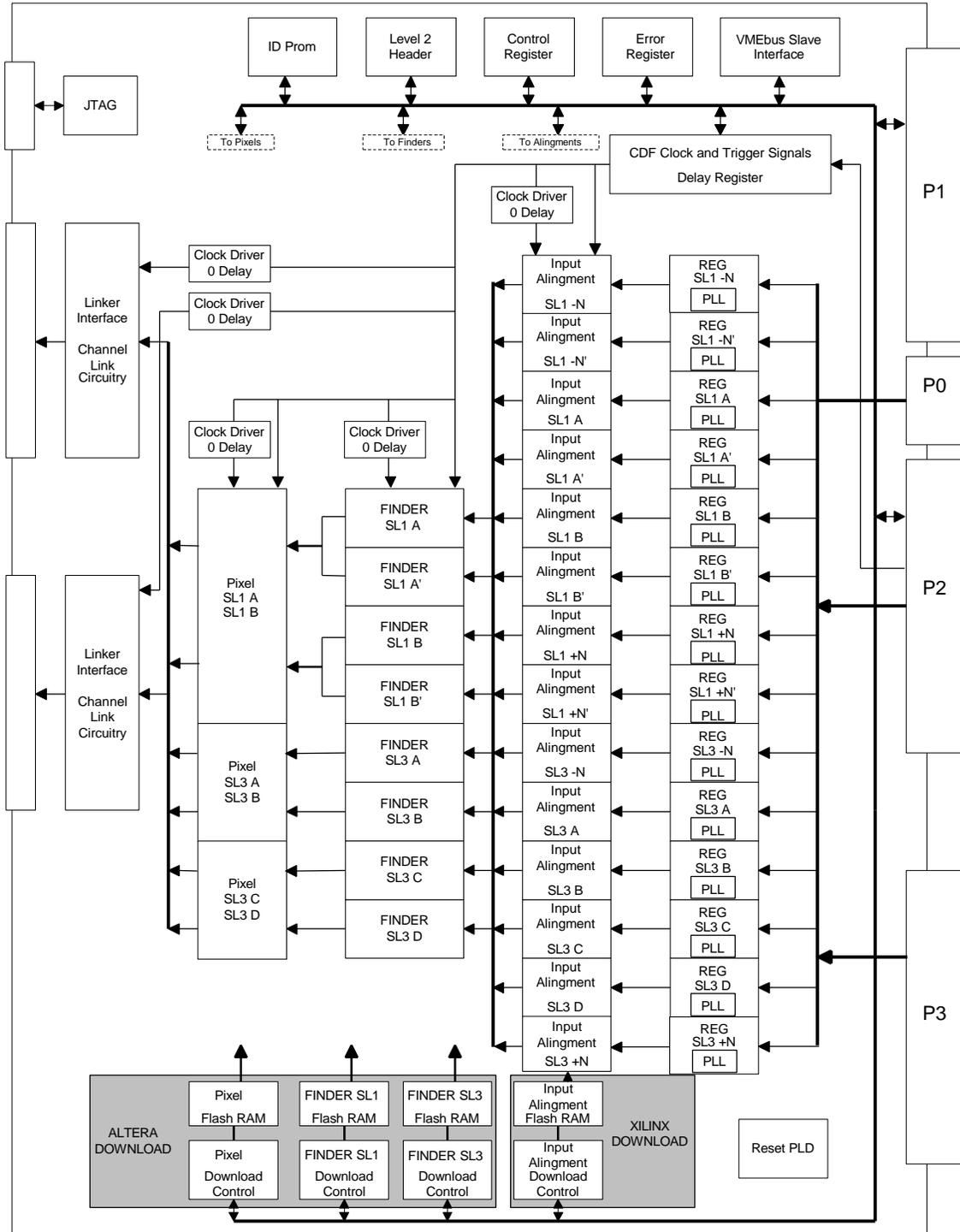


Figure 2. Finder SL1/3 Block Diagram

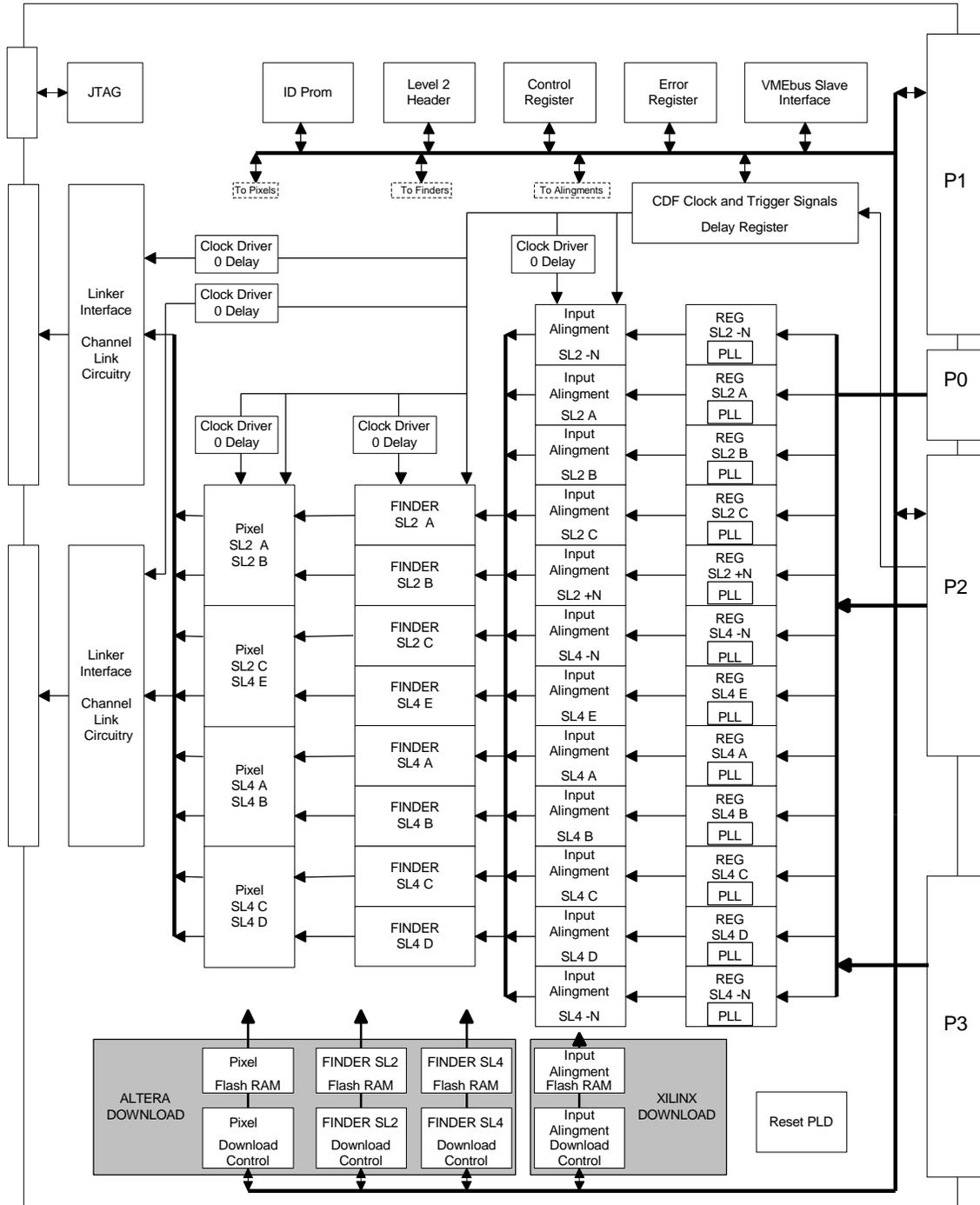
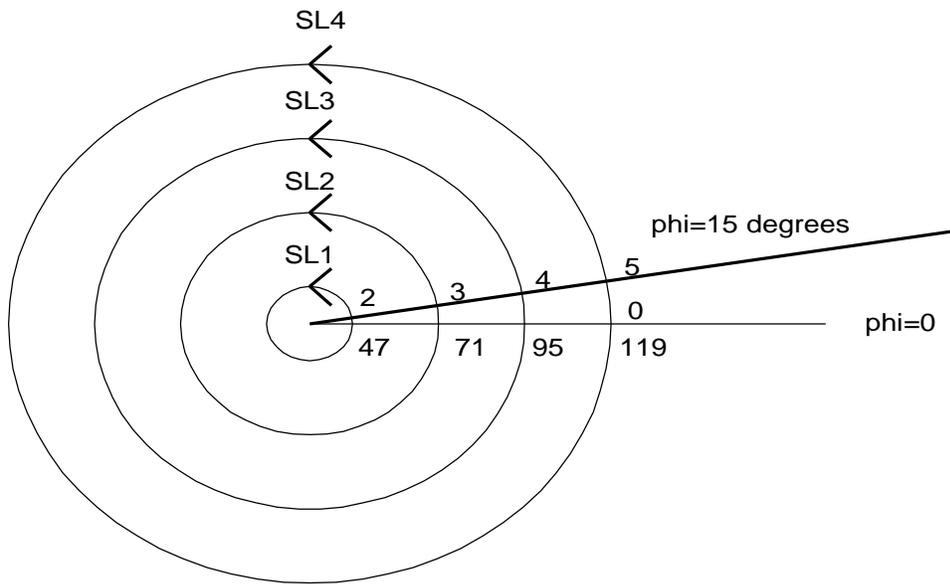


Figure 3. Finder SL2/4 Block Diagram



### Finder Mapping to the COT

Figure 4. Finder Mapping to the COT

Figure 5 illustrates a front view of a fully populated Finder crate. There will be three such Finder Crates in an XFT System. Figure 6 is a front view of a Linker crate and Figure 7 is a diagram illustrating interconnections between XFT crates.

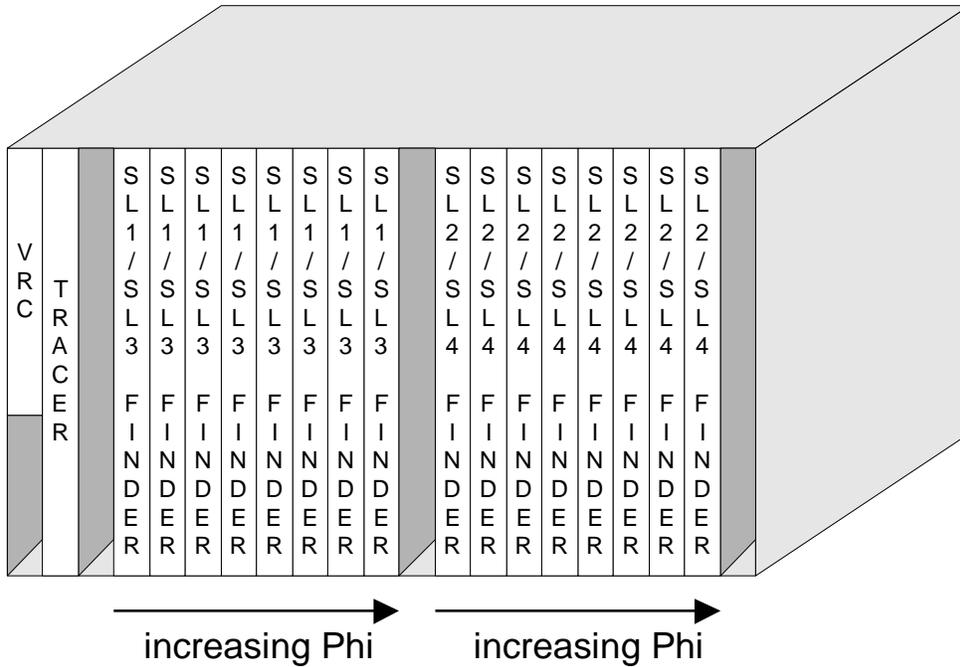


Figure 5. Finder Crate Configuration

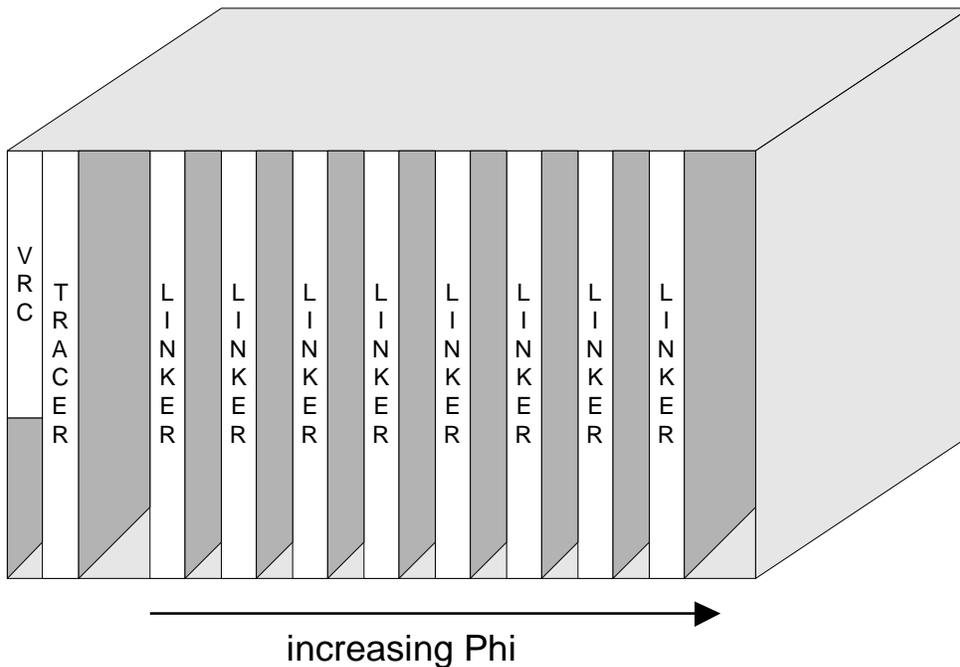


Figure 6. Linker Crate Configuration

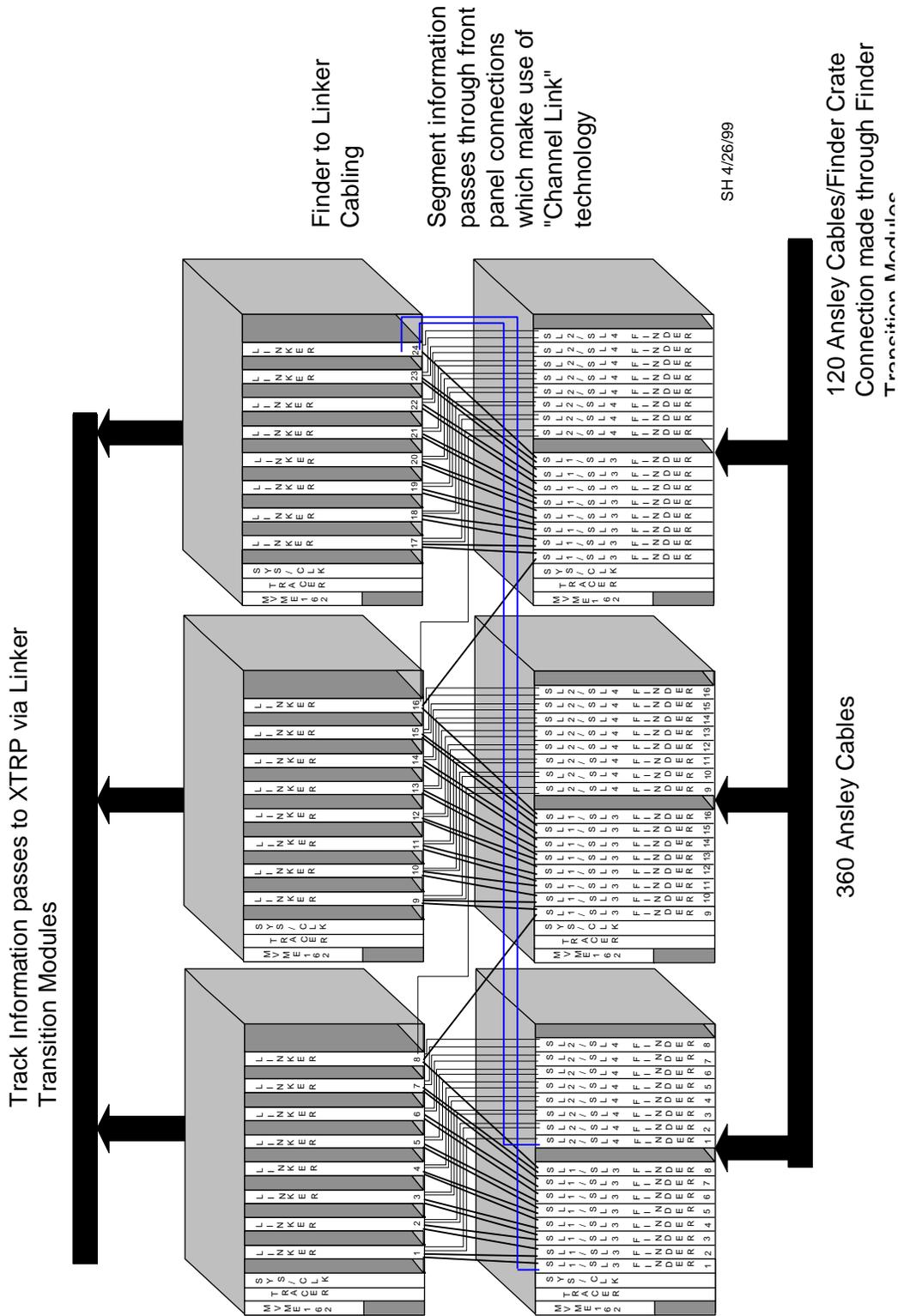


Figure 7. Overall View of XFT System

## Data Transmission - COT TDC Mezzanine Card to Finder Transition Module

The Finder modules receive data from the COT TDC mezzanine card. The COT TDC mezzanine card looks at the COT signal and determines whether a prompt and/or delayed hit has occurred. This information is then sent to the COT TDC Transition module (see Appendix A) which drives the information into the Ansley cables.

The Ansley cables (see Appendix B) have 24 available signal pairs, with a 25th pair which carries ground. The Ansley cable is a flat cable and differential information is sent up in GSSGSSGSSG configuration. Each Ansley cable will carry data from 48 COT TDC channels. In order to get the information up the cable, 6 data cycles will occur within the 132ns CDF clock period. This implies data will be coming up the cable every 22ns, or at a rate of 45.5Mhz. Table 1 shows the assignment of cable pairs and ordering of the 6 data cycles.

Pairs 1,2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7	Pairs 8-23	Pairs 24,25
GND	STROBE	GND	Beam_zero	GND	logic high	Channels 0-15 Prompt	GND
GND	STROBE	GND	Beam_zero	GND	logic low	Channels 16-31 Prompt	GND
GND	STROBE	GND	Beam_zero	GND	logic low	Channels 32-47 Prompt	GND
GND	STROBE	GND	Beam_zero	GND	logic low	Channels 0-15 Delay	GND
GND	STROBE	GND	Beam_zero	GND	logic low	Channels 16-31 Delay	GND
GND	STROBE	GND	Beam_zero	GND	logic low	Channels 32-47 Delay	GND

Table 1 Ansley Cable - Data Multiplexing

Figure 8 represents the mapping of the above data to the 48-channel four-cell COT block upon which each Ansley cable reports.

It will be the function of the COT TDC mezzanine card to actually deliver the 16-bits of prompt/delay data along with a Beam\_zero marker (set for all six words on a Beam Zero crossing event) along with an “edge transition” strobe every 22ns. This data will be received by the COT TDC Transition module which will make use of Low Voltage Differential Signaling (LVDS) technology to drive the data directly onto the Ansley Cable.

The Ansley Cable makes a 200 foot run from the COT TDC crates mounted on the CDF detector to the XFT Finder crates on the first floor of the B0 building. Due to the number of Ansley cables that must be dealt with (360+), and the fact that the Ansley cables are fairly bulky and inflexible, an oversize Finder Transition module (see Appendix C) has been designed.

Up to eight Ansley cables will be plugged into each Finder Transition Module, which will also contain the LVDS receivers and buffers. This will allow signals to pass through the backplane in a single-ended fashion. The backplane will also allow the passing of neighbor information that spans Finder module boundaries.



## **Finder Backplane**

Finder FPGAs are designed to look at the information coming up from a single Ansley cable as well as some additional data in both increasing and decreasing phi. The data from the single Ansley cable is referred to as the “core” data. The additional data picked up from adjacent (in phi) cables is referred to as neighbor data.

A custom J3 backplane will be required in order to bring TDC data into the Finder Module. The backplane will act to feed through single-ended TTL signals from all “core” Ansley cables from the Finder Transition Module to the Finder Modules. The backplane will also act to provide for the transmission of single-ended “neighbor” data from slot to slot if needed. Neighbor data must travel at most one slot across the backplane.

Appendix D goes into greater detail on the pin assignments and the routing philosophies of the XFT Custom J3 Backplane.

## FINDER MODULE DESCRIPTION

This section will describe the Finder module in sections that represent the blocks shown in figure 2 and 3. The description will start with the RESET PLD, CLOCKING and JTAG blocks followed by the data flow through the board and then onto the VME interface. The interface includes various FPGAs that make up the Finder board control circuitry. A memory map which details all the registers on the Finder module will conclude the description. The data flow through the board starts with the Input Alignment section followed by the Finder section and finally onto the Finder to Linker Interface section. With the implementation of reprogrammable devices there exist many diagnostic methods for testing the Finder module as a single unit or within the XFT system. A number of different diagnostic chip designs have been made for these tests and they are described in the appropriate sections.

The PLDs on the Finder board can be downloaded via a number of different methods. All PLDs are reconfigured when power is applied to the board or when the board's front panel reset button is engaged. The Altera Finder and Pixel chips can be downloaded via Flash RAM, or the Altera Bitblaster. Xilinx Alignment chips can be downloaded via Flash RAM or through the JTAG chain. All other Xilinx FPGAs including the VME\_SLAVE, ERROR\_REGISTER, CONTROL\_REGISTER, L2HEADER, ALTERA\_DOWNLOAD and XILINX\_DOWNLOAD use a serial PROM download scheme. The RESET\_PLD, ID\_PROM and all serial PROMs are programmed with the use of a DATA I/O tool.

### RESET\_PLD

The RESET\_PLD provides a means to control the configuration of all the FPGAs on the Finder Board. The RESET\_PLD is controlled by a power on reset signal, signals from the CONTROL\_REGISTER and 'done' signals from all the FPGAs or their chains. The RESET\_PLD produces signals that inform all FPGAs to configure(download). A global reset signal is also produced by the RESET\_PLD that informs all Xilinx FPGAs with the exception of the Input Alignment FPGAs to perform an internal reset.

The RESET\_PLD chip was designed using the XXXXXX software package. The design is based in a text design file. The design is implemented in an electrically erasable CMOS PLD device, specifically the MACH211-7JC. See appendix# for the text design that shows the PLD equations. The 44 pin PLCC is programmed using the DATA I/O programmer.

### CLOCKING

It will be possible to control the phasing of all clock and trigger signals on the Finder module. This is necessary to insure proper alignment of the raw data.

The Finder board's clocks are all derived from the CDF 132ns-clock signal found on the VMEBUS backplane. The differential CDF clock signal is received by a MC10350 device, which converts it to a single ended TTL signal. This signal is sent through a programmable delay line and becomes known as the boards MC(master clock). The MC is then sent to a Cypress CY7B991-7JC or Robo Clock. Robo Clock is a programmable skew clock buffer that incorporates a PLL for zero delay - input to output. The clock buffer also provides for clock multiplication and division. Two 132ns clocks along with four 33ns clocks and four 66ns clocks are produced by the Robo Clock device. These ten clock signals are then distributed throughout the board with the 33ns and 66ns being regenerated and buffered by other zero delay clock buffers specifically the CYPRESS CY7B9910-

5SC. This zero delay clock buffer is similar to the Robo Clock device with the exception that it's output frequency is the same as it's input frequency. The CDF trigger signals: /CDF\_B0, /CDF\_RECOVER, /CDF\_HALT, /CDF\_L1A, /CDF\_L1B0 and /CDF\_L1B1 are registered with the single ended TTL clock and then again with the MC. These signals are used by the FPGAs to align data to the CDF signals and also as a means to mark pertinent data for storage.

### JTAG- Boundary Scan

A JTAG-Boundary Scan chain is implemented on the Finder board to provide a method for testing the module for infrastructure and interconnect defects. There is the possibility of using the chain as a method for programming the FPGAs and Flash RAM devices. The JTAG chain connects to all PLDs with the exception of the RESET\_PLD and the L2HEADER FPGA. All boundary scan signals are buffered and fanned out with the use of a CY74FCT162244 16 bit buffer. The TCK and TMS lines are fanned out through six outputs. The Corelis Inc. ScanPlus software will be used for testing, debugging and programming the Finder boards.

### Input Alignment Section

The Finder Modules receive data from 360+ different Ansley cables. While the data should be coming up in lock-step across the system, it is unreasonable to expect all data to arrive within a few nanoseconds. Shifts in data timing may result from the position of the TDC module within a crate, TDC crate to crate differences, and possible shifts on the Ansley cables due to length or transmission qualities.

### Input Section - Capturing the Ansley Cable Data

Since the Ansley cables are presenting data every 22ns, a conservative approach has been taken and an "input" stage will be used to receive the output of each Ansley cable. This input stage will make use of data registers that will be capable of 100Mhz synchronous operation and phase lock loop(PLL) devices to regenerate the 22ns clock signal. The data registers used are the Cypress CY74FCT162823T 18-bit registers and the PLL are the Cypress CY7B991-7JC or Robo Clock devices. The input data from each Ansley cable will be clocked by a signal formed by taking the edge "Strobe" signal coming from the same Ansley cable, running it through an individual Robo Clock, doubling its frequency, and allowing for phase adjustment (see Figure 9). The output of the 18-bit register along with the regenerated 22ns clock is forwarded to the Alignment FPGAs.

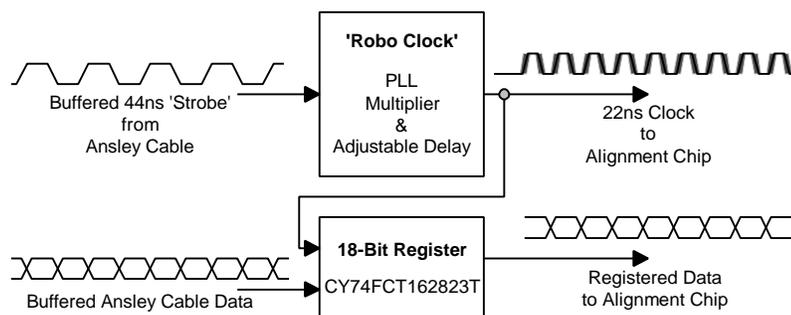


Figure 9. Input Section - Capturing the Ansley Cable Data.

### Alignment Section – Aligning the Data to the 33ns Board clock

The Alignment chips are implemented in Xilinx XC4005E FPGAs. These reprogrammable devices implement the design shown in the block diagram of Figure X. Each FPGA is responsible for aligning the registered data from it’s Ansley cable with the CDF\_Clock signal supplied on the backplane and then sending it on to the Finder stage in the proper format. The 16 and 48-bit register blocks represent banks of registers used to store data for manipulation.

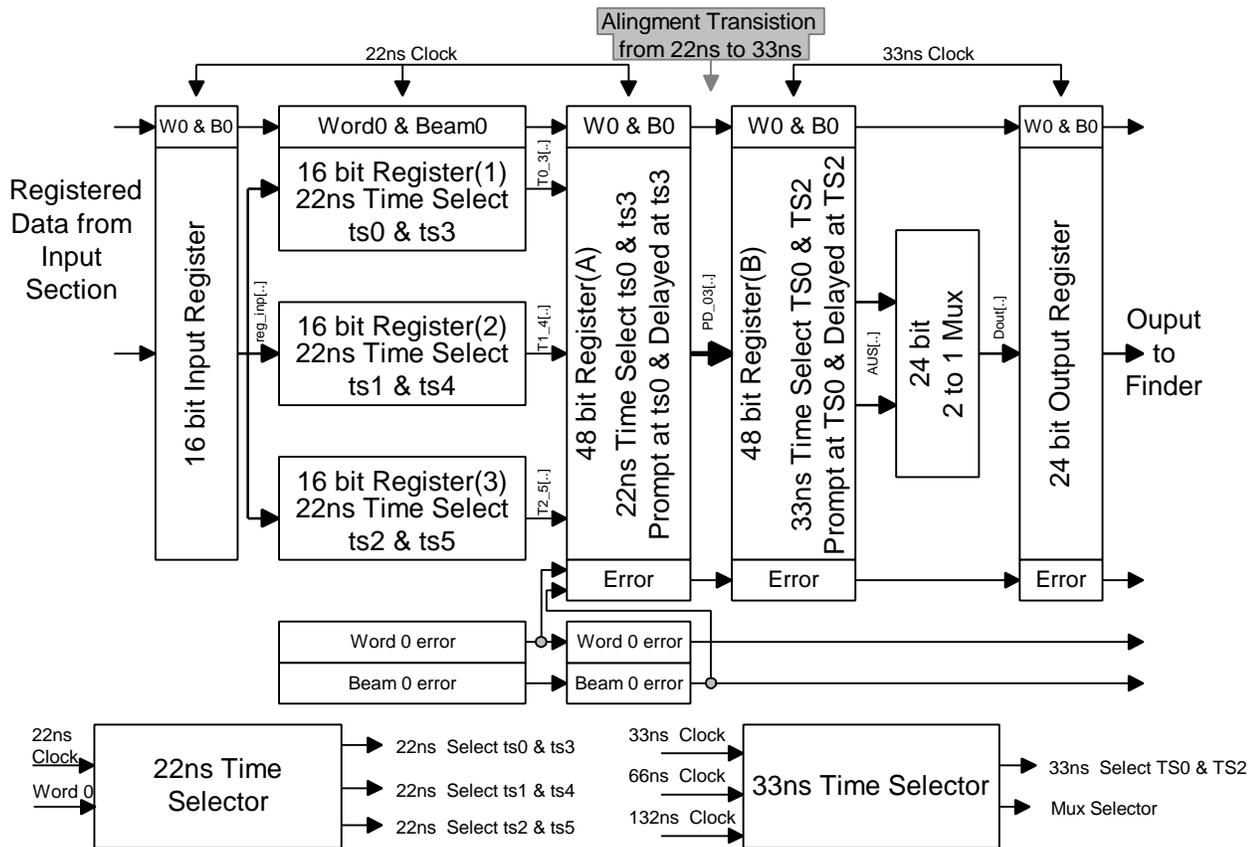
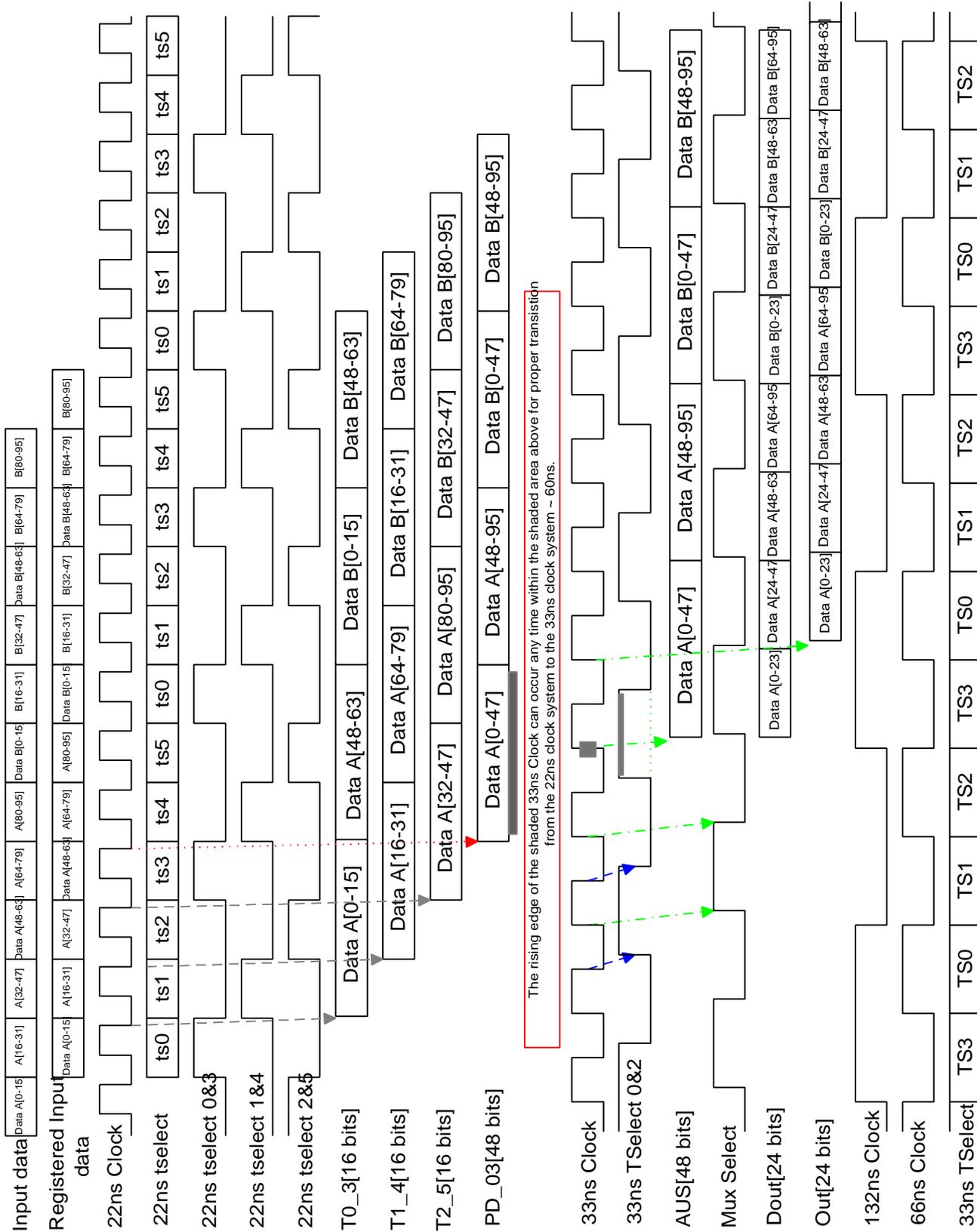


Figure X. Alignment Chip Block Diagram.

The Time selector blocks provide a means to enable the register banks at different times. The 22ns Time selector receives the 22ns clock and the Word 0 bit and creates 3 “22ns time select” bits. Each time select bit refers to the time bin that it will be at a logic high level. There are 6 22ns time bins every 132ns with each of the time select bits being on during 2 of the 6 bins. “Time select 0” is aligned to the Word 0 bit from the Ansley cable data.

The 33ns Time selector block receives the boards 33ns, 66ns, and 132ns clock that are all generated from the 132ns CDF clock on the backplane. There are 4 33ns time bins every 132ns. The 33ns Time Select block generates a “Time Select 0&2” square wave signal that is at logic high level for 33ns during time bins 0 and 2. The block also outputs a “Mux select” bit which is a registered derivative of the “Time Select 0&2” bit. “Time select 0” is aligned to a phase relationship between the 33, 66 and 132ns clocks. A timing diagram is shown on the following page followed by a description of the operation of the FPGA.



The Alignment FPGA design works as follows: The registered Ansley cable data is again registered inside the FPGA. This registered data is then stored in one of three 16 bit registers that is selected by the “22ns Time Select” block. If the registered data contained a ‘Word 0’ signal then that data is time bin 0 and it is stored in the 16-bit register enabled by the “22ns Time Select 0&3” signal. The next two 16-bit slices of data are stored in the next two 16-bit registers at time bins 1 and 2. At this time there is 48-bits of data stored in the three 16-bit registers. At time bin 3 those three 16-bit values are transferred to the preceding 48-bit register(A) and held for 66ns. Also at time bin 3 the fourth 16-bit slice of data is stored in the original or first 16-bit register followed by the fifth and sixth 16-bit slice at time bins 4 and 5 in the 2<sup>nd</sup> and 3<sup>rd</sup> 16-bit registers. At the next time bin 0 the fourth, fifth and sixth 16-bit slices of data are transferred to the 48-bit register(A). The first three time slices of data in the 48-bit register(A) should have been transferred to the preceding 48-bit register(B) that operates off the 33ns clock.

The 33ns clock that registers the 48-bits of data from the 22ns register(A) can be adjusted to occur during the 66ns duration that the data was stored in the first 48-bit register(A). Due to setup and hold times this adjustment is approximately 60ns. Adjusting more than 60ns will cause the data to become unaligned with the Beam 0 and Word 0 signals.

The second 48-bit output or register(B) output is split into two 24-bit signals and fed to a 24-bit 2-to-1 multiplexer that forwards the correct 24 bits to the output register. The “33ns Time Selector” block provides that MUX select signal.

The 24 bits of data along with ‘Beam 0’, ‘Word 0’, ‘Error’ and ‘Operate’ are registered with the 33ns clock signal before they are sent to the output pins and onto the Finder chips. The “Operate” bit will be set and remain set once a “Beam 0” signal occurs. The ‘Word 0’ bit sent to the Finder chip will be at a logical high level during the 2 prompt time slices. The “Error” bit is set if there are not 6 consecutive 22 time bins of the “Beam 0” signal “OR” if there is more than one 22ns “Word 0” signal in 6 consecutive 22ns time bins. The “Error” bit moves along to the Finder in sync with the input data. The design also forwards the individual “Beam 0 Error” and “Word 0 Error” bit to an output pin and onto the Error register of the board.

The table below and the picture on the preceding page show the format of the data as it leaves the Alignment chip and enters the Finder chip.

time slice 0:	Error_Flag	Word_zero_Marker	Beam_Zero_Marker	Channel 0-23 Prompt
time slice 1:	Error_Flag	Word_zero_Marker	Beam_Zero_Marker	Channel 24-47 Prompt
time slice 2:	Error_Flag	Word_zero_Marker	Beam_Zero_Marker	Channel 0-23 Delay
time slice 3:	Error_Flag	Word_zero_Marker	Beam_Zero_Marker	Channel 24-47 Delay

Table X. Finder FPGA view of wire numbering

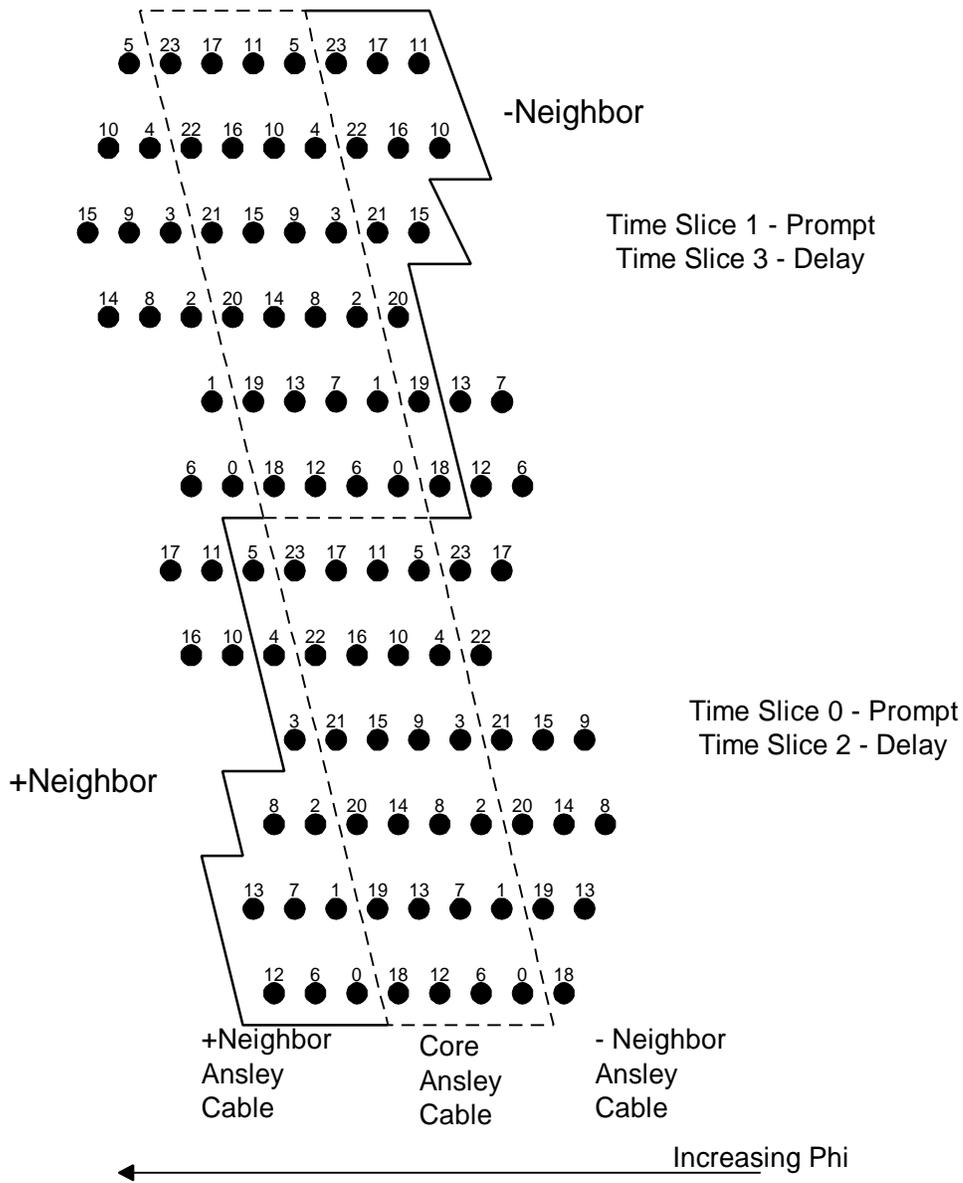


Figure 10. Finder FPGA view of wire numbering

### Alignment Diagnostics

Diagnostic programs in the form of Capture and Driver mode will allow the user to split the system at the Alignment section of a Finder Board. The Capture mode will allow for the capture of data being driven up the Ansley cables at speed, which can be used to verify the connections between the COT TDC Mezzanine Modules, Mezzanine Transition Module, Ansley Cable, Finder Transition Module, Finder Crate Custom Backplane and the Finder Modules Input section. The Driver mode will allow patterns to be sent from the Alignment FPGAs into the Finder chips to test the segment finding algorithm. The Driver mode can also be used to verify the connections to the Finder chips,

Pixel chips, Channel Link devices, Channel Link cables and eventually offboard to the Linker Module.

- Alignment Chip Capture Mode diagnostics

The Capture design is used to catch the wire data sent on the Ansley cable. The graphical design file consists of a 64x18 RAM block for storing the data along with a counter and compare block. The counter and compare block are used to generate the address for the RAM block. The RAM block is loaded with the next 64 22ns slices of wire information once it is enabled. The RAM block is filled when the following happens: the Loop mode bit is set by the control register of the Finder board (low to high transition on the Loop bit along with the requirement that it stays high) and then the combination of Beam\_0, Word\_0 being present for 22ns. The wire information that is in the same 22ns time slice as the Beam\_0, Word\_0 combination will not be written into the RAM block, the information in the next time slice will be located in the first memory location. The counter and compare circuitry disable the RAM from being written after a count of 64 is reached. The RAM block is **Read only** through VME. The loop mode bit should be set low before reading the RAM block. The format of the 18 bits stored in the RAM block can be found in the Memory Map.

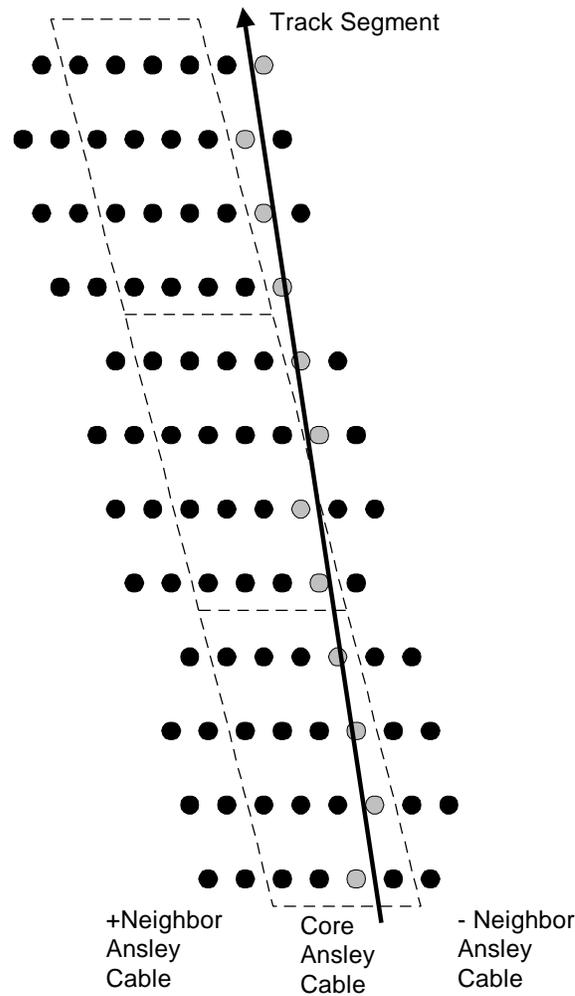
- Alignment Chip Driver Mode diagnostics

The Driver design is used to send preloaded wire information out of the Alignment chip to the Finder chip. The graphical design file consists of a 64x32 RAM block that is used for storing the data that is sent out. The RAM block is R/W through VME. The data is transferred to the Finder chip when the Loop mode bit is set. A circular counter controls the address lines of the RAM block and it will count as long as the Loop mode bit is set. The Loop mode bit should be set low to R/W the RAM block. The format of the 32 bits stored in the RAM block can be found in the Memory Map. The 'Operate', 'Beam\_0' and 'Word\_0' bits need to be set to start the Finder chips.

## Finder FPGAs

The Finder FPGAs have the job of identifying track segments in a given superlayer of the COT. Finders flag “hits” by setting pixels that indicate the position and/or slope of an identified track segment. Each Finder FPGA will report 12 pixels for each COT cell.

A “hit” is identified to have occurred whenever at least 9, 10 or 11 out of 12 wires in a mask have been hit. Figure 11 shows a mask set(gray dots) of 12 wires that may be used to identify the sample track passing through a given cell.



Gray Wires indicate 12 wire mask for the depicted track segment

Figure 11. Possible Mask Set for a Sample Track

The Finders are implemented with Altera Flex 10K Field Programmable Gate Arrays (FPGAs), specifically the Altera EPF10K50RC240-3 for superlayers 1, 2 and 3 and the Altera EPF10K70RC240-3 for superlayer 4. These reprogrammable devices implement the Finder design shown in the block diagram of Figure 12. Figure 13 illustrates the timing path of the segment data through the FPGA. Three Diagnostic designs have been made that can be loaded into the Finder chips. These three designs allow the Finder to act as a receiver, driver or buffer. These designs will be detailed farther in the document.

The Finder FPGA is synched up to the CDF\_B0 signal. The Finder will not start running unless it receives a “Operate” signal from the Input Alignment Section of the board. The CDF\_B0 signal on the Finder board will register the “Operate” signal and that registered signal called “Operate Finder” anded with “Word\_0” core from the Input Alignment Section will start the Finder Algorithm.

The major blocks of the Finder FPGA and shown in figure 12 are described below:

### **DEMULTIPLEXER**

- 4 to 1 Time demultiplexer of wire information from the Data Alignment section of the Finder board operating at 33ns.
- 4 time slices of wire information form a 140 bit register that contains the prompt and delayed wire information of 4 COT cells and the pertinent neighboring cells wire information(11 wires each side).
- 140-bit register is valid for the next 132ns.
- Word\_0 is regenerated for the first time slice.
- Beam\_0 must be present from the Core data and also from the Neighbor data for Beam\_0 to be regenerated.
- Error markers are generated for Word\_0, Beam\_0 and Error flag. Error flag moves along with the data. Word\_0\_Error and Beam\_0\_Error are output to the Error register of the Finder board.
  1. Error\_flag is set if the error flag input from the core data or neighbors is present or if Beam\_0\_Error or Word\_0\_Error is present.
  2. Word\_0\_Error is set if the Word\_0 core and Word\_0 neighbors are not all present during the first time slice or if any of them are present during the other three time slices.
  3. Beam\_0\_Error is set if the Beam\_0 core and Beam\_0 neighbors are not all present together during all four-time slices.
- 140-bit register is the OR'd combination of the input wires and the appropriate dead wire register bit. The dead wire register is a VME R/W register that is used to set all dead wires to a preset state.

The DEMULTIPLEXER is implemented in a text design file that incorporates a state machine with 4 states. Each state represents one of the four 33ns time slices of the 132ns-time slice. The state machine will transition to the second state when a “Operate Finder” signal and a “Word Zero” signal are present during the rising edge of the 33ns clock. The state machine will proceed to the third, fourth and then back to the first state on the next three rising edges of the 33ns clock. The state machine will then again check for the combination of “Operate Finder” and “Word Zero” before it will proceed thru its states again. The state machine will idle in the first state if the above combination is not present.

## MULTIPLEXER

- 1 to 4 multiplexer of wire information from the 140-bit register provided by the DEMULTIPLEXER.
- Requires a different design for each of the Four Super Layers. The number of bits of the 140-bit register varies with each Super Layer. On average it is about 47 bits per cell, with wires overlapping into adjacent cells.
- Multiplexes the 140 bits of wire information into four 47 bits of wire information that relate to pertinent wires required to fire pixels of each of the 4 cells. The 47 bits are sent to the MASK block .
- The 47 wires change every 33ns. In 132ns four cells worth of wire information are pipelined into the MASK block.
- The Multiplexer also sends the 24 bits of prompt and delayed wire information of a cell to the level 1 pipeline every 33ns, so in 132ns the level 1 pipeline receives the prompt and delayed information of all four cells.
- The Multiplexer also passes the Word\_0, Beam\_0 and Error\_flag bits of a cell in time with the wire information to both the MASK block and to the Level 1 Pipeline.

The MULTIPLEXER is implemented in a text design file that incorporates a state machine with 4 states. Each state represents one of the four 33ns time slices of the 132ns-time slice. The state machine will transition to the second state when a “Start” signal is present during the rising edge of the 33ns clock. The state machine will proceed to the third, fourth and then back to the first state on the next three rising edges of the 33ns clock. The state machine will then again check for the “Start” bit before it will proceed thru its states again. The state machine will idle in the first state if “Start” is not present.

## MASK SET

- Looks for the number of misses within a group of 12 wires(MASK).
- If the number of misses is 3 or less, a pixel that relates to a segment in that cell is turned on.
- More than one Mask may turn on the same pixel.
- Requires a different design for each of the Four Super Layers. The number of Masks varies with each Super Layer.
- There is separate MASK design files that look for 1, 2 or 3 misses per Mask.
- The Mask set has 2 pipelined stages running at 33ns so the 12 bits of pixel data for a cell is output 66ns after the wire information for the cell that is selected by the Multiplexer.

The MASK block is implemented in a text design file that incorporates two other functions that are also implemented in text design files. The two functions are called “MC\_#” for missing counter #(3, 2, or 1) and “NM\_#” for number of misses #(3, 2 or 1). The wire information from the Multiplexer block is passed to the MC\_# functions. A MC\_# function looks at 4 wires of a mask and determine how many of the wires are off. The MC\_# registers the number of misses in those four wires. The NM\_# function takes the registered data from three MC\_# functions and adds together the number of misses of those three functions or the twelve wires of a MASK. If the number of misses in the three MC\_# functions is less than the

required(1, 2, or 3) number of misses allowed then a pixel\_hit bit is registered. More than one MASK can set a pixel so the appropriate registered pixel\_hit bits are OR'd together to form a 12 bit pixel output.

The MASK block also passes along the Word\_0, Beam\_0 and Error\_flag bits.

### **LPM\_DFF**

The LPM\_DFF that is located to the right of the MASK block registers the OR'd pixel\_hit data along with the Word\_0, Beam\_0 and Error\_flag bits that is provided by the MASK block every 33ns. These registered bits are sent to output pins of the Finder Chip.

### **LEVEL 1 PIPELINE & LEVEL 2 BUFFERS**

- The 24 bits of wire information, Beam\_0, Word\_0 and Error\_Flag of the four cells are stored in a level 1 pipeline that is programmable up to 256 time slices(33ns) deep.
- Level 1 pipeline depth is R/W thru VME. The depth is set by an 8-bit word that refers to the number of 33ns slices in the pipeline.
- On a level 2 write command the wire information of the 4 cells from one time slice(132ns) is stored in one of four level 2 buffers.
- Consecutive level 2 writes can be performed.
- Level 2 buffers are readable through VME.

The Level 1 Pipeline is implemented in a graphical design file that includes a 256x32 RAM block, 8-bit counter and an 8-bit compare. The wire information is written into the RAM location specified by the 8-bit counter. The 8-bit counter is a loop counter that counts up to the value set by the pipeline depth register value. The pipeline depth register value is fed to the 8-bit compare that compares the value of the pipeline depth register with the value of the 8-bit counter. When the two 8-bit values are equal the counter is reset to zero, thus creating a loop counter or circular buffer for the pipeline. The RAM block operates in such a way that the data\_in(wire info.) and address(counter value) are registered on the rising edge of the 33ns clock and the data\_out(pipelined wire info.) is registered on the falling edge of 33ns clock. The RAM block or EAB of the Altera device create it's own write enable pulse dependent upon the clock signal. This feature allows the data present at the address location specified by the counter to be output and registered before the next slice of wire information is stored at the same address specified by the counter.

The four Level 2 Buffers are implemented in a text based design file. The design incorporates four 108 bit registers, a four-place state machine and a large 4 to 1 multiplexer. Each of the four Level 2 buffers is made from a 108-bit register. The state machine controls the writing of the pipelined wire information into the correct 108-bit register. The state machine will start when a level 1 accepts signal is present. The state machine will proceed thru its four states storing the next four slices of wire information in the specified buffer address(Level 1 address).

The wire information stored in the buffers is read out by specifying the address with the level 2 address lines(4 bits). These address lines control a 27 bit, 16 to 1 multiplexer. The format of the level 2 buffer readout can be found in the Memory Map detailed farther in the document.

### **Finder Chip CONTROL REGISTER**

- Dead Wire register - 70 bits - R/W

- Pipeline Depth register - 8 bits - R/W

The Control register is implemented in a text based design file. These registers are accessible by enabling the chip select pin, selecting the correct address lines, setting the Buf\_RD pin to high for a read and low for a write. A strobe on the wrtstrb pin will register the data for write operations. Setting the correct address performs a read operation. The format of the registers can be found in the Memory Map detailed farther in the document.

### **Miscellaneous logic on the Top Level Graphical Design.**

The Level 1 Accept and Level 1 address lines are registered in the Finder chip with the use of a signal produced from the DEMULTIPLEXER state machine. The signal 'time\_0' occurs every 132ns and aligns the Level 1 signals up with the first(33ns) time slice of data.

The Start Link output pin is used to tell the circuitry following the Finder chip that the data coming on the pixel output pins is valid. The 3 flip-flops proceeding the Start Link pin are used to keep the Start Link pin in time with the pixel data.

### **FINDER Diagnostic Designs**

- FINDER as a Receiver

The Finder as a receiver design is used to catch the wire data sent from the core Input Alignment section. The graphical design file consists of a 256x32 RAM block for storing the data along with a counter and compare block. The counter and compare block are used to generate the address for the RAM block. The RAM block is loaded with the next 256 33ns slices of wire information once it is enabled. The RAM block is filled when the following happens: the Loop mode bit is set by the control register of the Finder board(low to high transition on the Loop bit along with the requirement that it stays high) and then the combination of Beam\_0, Word\_0 and Operate being present for 33ns. The wire information that is in the same 33ns time slice as the Beam\_0, Word\_0 and Operate combination will not be written into the RAM block, the information in the next time slice will be located in the first memory location. The counter and compare circuitry disable the RAM from being written after a count of 256 is reached. The RAM block is R/W thru VME. The loop mode bit should be set low before reading the RAM block. The format of the 32 bits stored in the RAM block can be found in the Memory Map.

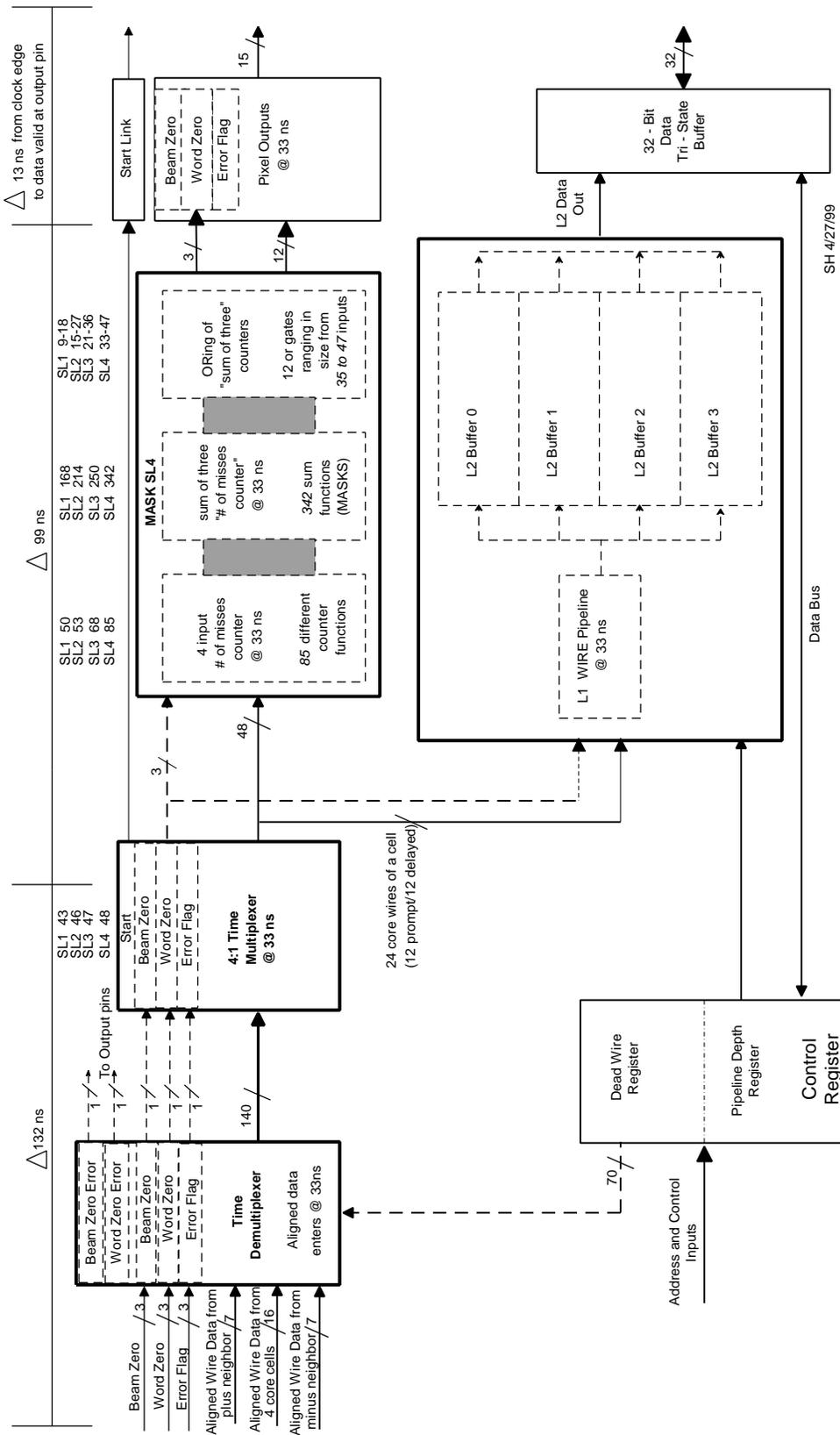
- FINDER as a Driver

The Finder as a driver design is used to send preloaded pixel information out of the Finder chips. The graphical design file consists of a 256x32 RAM block that is used for storing the data that is sent out of the Finder. The RAM block is R/W thru VME. The data is transferred to the Finder chip output pins when the Loop mode bit is sent. A circular counter controls the address lines of the RAM block and it will count as long as the Loop mode bit is set. The Loop mode bit should be set low to R/W the RAM block. The format of the 32 bits stored in the RAM block can be found in the Memory Map.

- FINDER as a Buffer

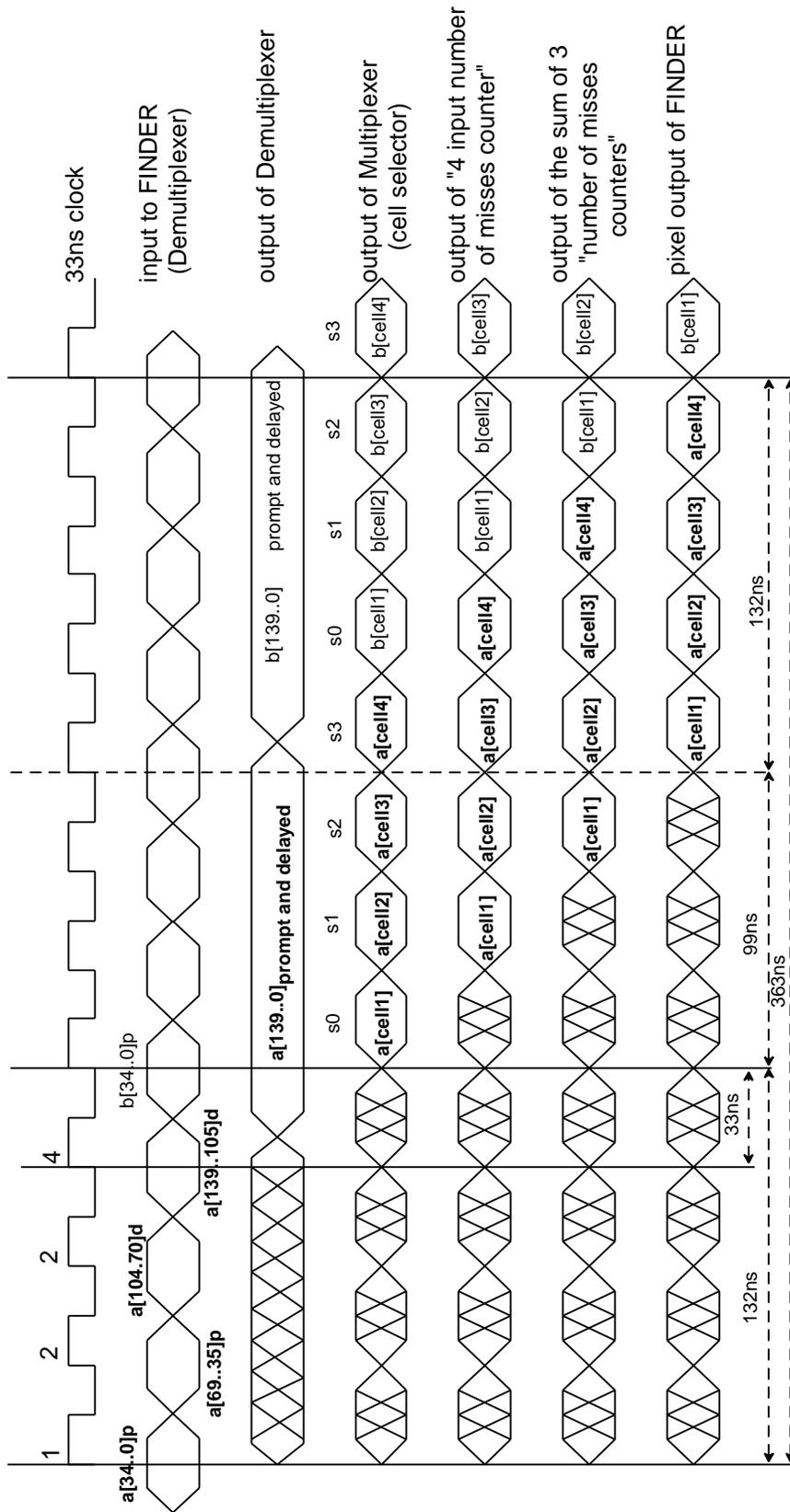
The Finder as a buffer is used to pass wire information from Input Alignment chips thru the Finder to the Pixel chips, the information would usually be in a pattern which is easy to interpret. The pixel information from a Mask set is not easily determined. It can also be used as a receiver to catch the information on all of the inputs from Input Alignment Chips including wire information from the neighboring Input Alignment

chips. This design was done as a method to test all of the connections between Input Alignment chips and their neighbors with the Finders. This design will be obsolete once the JTAG Boundary Scan test is implemented.



XFT FINDER Block Diagram

Figure 12. The FINDER FPGA Block Diagram



XFT FINDER - Timing Diagram SH 4/30/99

Figure 13. Finder Timing Diagram

## Finder to Linker Interface

Each Finder module must present two copies of its pixel (or, pixel and slope) information to the Linker Modules. This task is performed using an FPGA called the Pixel Chip that forwards duplicated information to two different National Semiconductor Channel Link devices that in turn forward the information to two different Linker modules. The Channel Link devices consist of a driver and receiver pair. The driver is located on the Finder board and the receiver is located on the Linker board. The data from three(Finder Board SL1/3) or four(Finder Board SL2/4) Channel Link devices are grouped together in a single cable.

### Pixel Chip FPGAs

The Pixel Chip designs are implemented with Altera Flex 10K Field Programmable Gate Arrays (FPGAs), specifically the Altera EPF10K20RC240-3 for superlayers 2, 3 and 4 and the Altera EPF10K50RC240-3 for superlayer 1. The design for the super layer 1 Pixel chip is larger due to the possibility of the split wire factor. The difference between the designs is detailed below. These reprogrammable devices implement the Pixel design shown in the block diagram of Figure 14. Figure 15 illustrates the timing path of the pixel data through the FPGA. The Pixel chip performs three functions: 1. Combine the Pixel, Beam\_0, Word\_0 and Error\_flag information from two Finders into a single 28-bit word. 2. Duplicate that information and drive it through separate individual outputs. 3. Provide a Level 1 pixel information pipeline and Level two buffers for VME readout. Two Diagnostic designs have been made that can be loaded into the Pixel chips. These two designs allow the Pixel chip to act as a receiver or driver. These designs will be detailed farther in the document.

The top level Pixel Chip is based in a graphical design file. The major blocks of the Pixel chip design that are shown in the diagram are described below:

### Pixel Data Duplication

The OR'ing of the Beam\_0, Word\_0, Start\_link and Error\_flag bits from two Finder chips is done to reduce the number of bits received from the two Finders down to the required 28 bits that can be driven by a Channel Link device. The Pixel information from the two Finders is not OR'd. The 12 bits of pixel information from each of the two Finders along with the OR'd data from the two Finders is registered in two 28 bit registers in the Pixel Chip design. The two separate 28 bits of registered data is then sent to output pins which drive two 28 bit Channel Link devices. The Error\_flag bit that is registered is set if the Beam\_0, Word\_0 and Start\_link bits from the two Finders do not occur at the same time or if either of the two Finders sends an Error\_flag.

### Pixel Pipeline and Level 2 readout

The 12 bits of Pixel information along with the Beam\_0, Word\_0, Error\_flag and Start\_link bits from the two Finders is stored in a 30 bit Pipeline and Level2 buffer design that is the same as that found in the Finder design. Refer to the Finder design information above and to the memory map for bit information.

## Pixel Chip Control Register

- Channel Link Disable - 2 bits - R/W (not yet implemented)
- Pipeline Depth register - 8 bits - R/W

The Control register is implemented in a text based design file. These registers are accessible by enabling the chip select pin, selecting the correct address lines, setting the Buf\_RD pin to high for a read and low for a write. A strobe on the wrtstrb pin will register the data for write operations. Setting the correct address performs a read operation. The format of the registers can be found in the Memory Map detailed farther in the document. The Pixel Pipeline Depth register should receive the same value as the Finder's Wire Pipeline Depth register. There is a block in the Pixel Chip design that takes care of the timing difference between the Finder Chip design and Pixel Chip design. The Channel Link Disable register will be used to set a Channel Links input to a predefined state. This will be accomplished by setting an enable pin on the 28 bit registers that connect to the output pins. By setting the enable low the outputs remain in their previous state.

## Differences between SL1 and SL2, 3, 4 Pixel Chip Design

The SL1 Pixel chip receives pixel data from Four SL1 Finder chips. The design contains two separate Pixel pipelines and level 2 buffers, one for the normal SL1(A, B) Finder data and one for the SL1(A', B') Split wire Finder data. The data that is registered, duplicated and sent to the Channel Link devices is the OR'd combination of SL1A normal with SL1A' split and the OR'd combination of SL1B normal with SL1B' split.

## Pixel Chip Diagnostic Designs

The Pixel Chip Diagnostic designs are similar to the Finder Diagnostic designs.

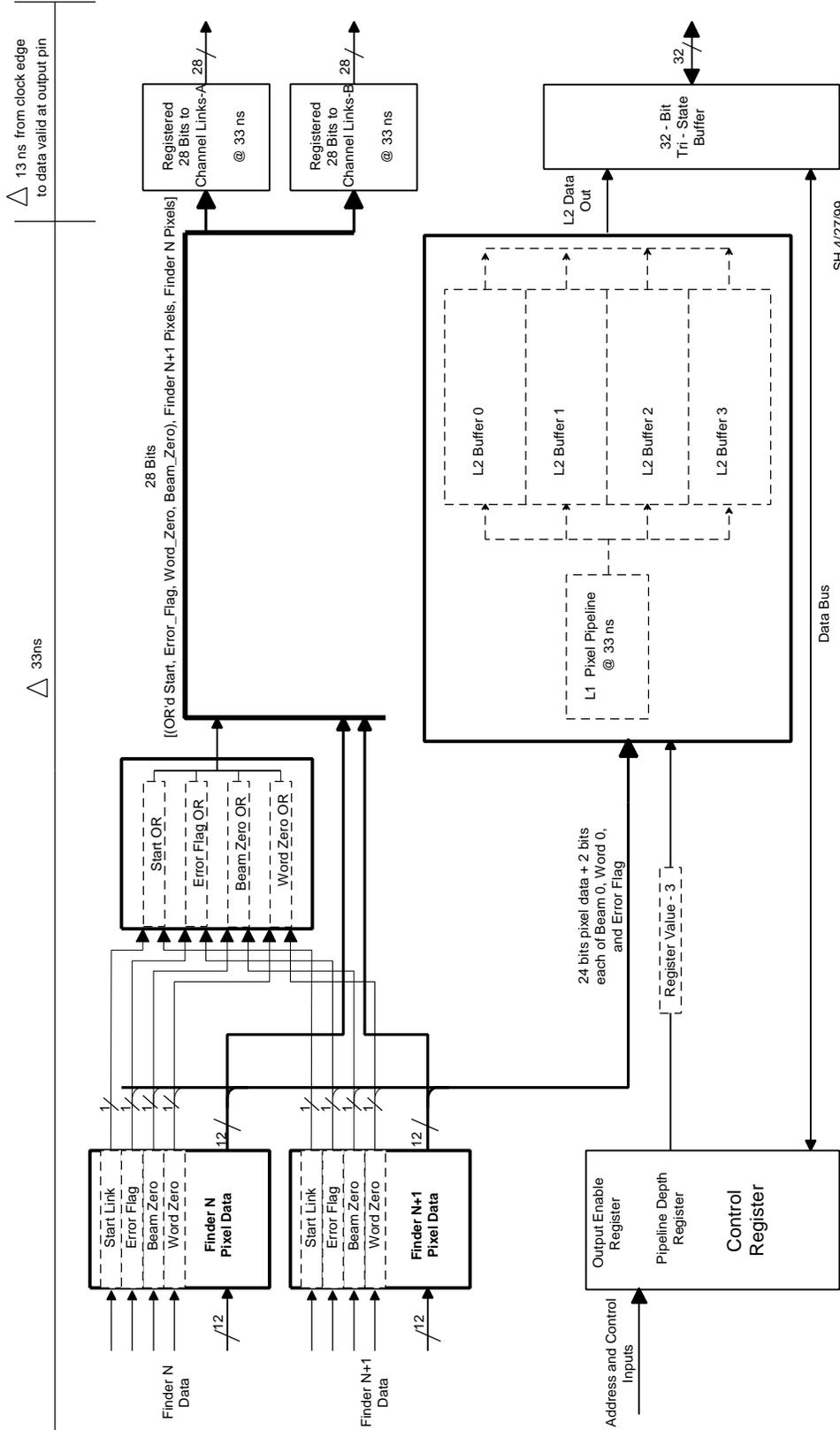
- Pixel Chip as a Receiver

The Pixel Chip as a receiver design is used to catch the pixel data sent from the two Finder chips. A RAM block within the design is loaded with the next 256 33ns slices of pixel information once it is enabled. The RAM block is enabled when the following happens: the Loop mode bit is set by the control register of the Finder board (low to high transition on the Loop bit along with the requirement that it stays high) and then the combination of Beam\_0 and Start\_link from either of the two Finders being present during the rising edge of the 33ns clock. The pixel information that is in the same 33ns time slice as the Beam\_0 and Start\_link combination will not be written into the RAM block, the information in the next time slice will be located in the first memory location. The counter and compare circuitry disable the RAM from being written after a count of 256 is reached. The RAM block is R/W thru VME. The loop mode bit should be set low before reading the RAM block. The format of the 32 bits stored in the RAM block can be found in the Memory Map. The Pixel Chip as a Receiver design also performs the normal Pixel chip algorithm of duplicating and driving the Pixel information to the Channel Link devices. The RAM block replaces the Pipeline and Level 2 Buffers.

- Pixel Chip as a Driver

The Pixel Chip as a driver design is used to send preloaded pixel information out of the Pixel chips. A 256x32 RAM block within the design is used for storing the data that is sent out of the Pixel Chip. The data

stored in the RAM block represents the Pixel information from two Finders along with the OR'd data found that is found in the normal design. The data coming out of the RAM block is registered in two 28-bit registers(same as in the normal design) before it is sent thru the output pins to the Channel Link devices. The data is sent out of the RAM block when the Loop mode bit is set in the Finder Boards control register. A circular counter controls the address lines of the RAM block and it will count as long as the Loop mode bit is set. The RAM block is R/W thru VME. The Loop mode bit should be set low to R/W the RAM block. The format of the 32 bits stored in the RAM block can be found in the Memory Map.



**XFT Pixel Block Diagram**

Figure 14. The Pixel FPGA Block Diagram

## Channel Link

The link between Finder board and Linker board consists of 3 or 4 National Semiconductor's DS90CR281/DS90CR282 28-Bit Channel Link pairs. The DS90CR281 transmitter converts 28 bits of data into four LVDS data streams. A phase-locked transmit clock is transmitted in parallel with the data streams over a fifth LVDS link. Every cycle of the transmit clock 28 bits of input data are sampled and transmitted. The DS90CR282 receiver converts the LVDS data streams back into 28 bits of TTL data in synch with an output clock signal that is derived from the LVDS clock signal thru a phase lock loop. The data on the LVDS pairs will be transmitted at 7 times the clock frequency. The LVDS clock signal is transmitted at the phase lock loop frequency.

- The cable runs from Finder Modules to Linker Modules will be short <<3 meters. Most of the runs are six feet between modules within a rack and up to ten feet between modules that are in adjacent racks.
- By using the **28-bit Channel Link running at 30.3MHz**, it is possible to transfer (30.3 MHz x 7 bits/LVDS link) 212 Mbps per LVDS pair or 848 Mbps

## Data Bandwidth

- Each Finder will have 15 bits of information every 33ns, or a rate of 30.3MHz. This will be made up of 12 bits of pixel information (or 6 bits of Pixel and 6 bits of slope) and a Beam\_0, Word\_0 and Error\_flag marker. The Beam\_0, Word\_0 and Error\_flag marker are OR'd with the adjacent Finder chip.
- A SL1/3 Finder Module will send (28 bits per 2 Finders x 3 channel links every 33ns) for a bandwidth of  $2.54 \times 10^9$  bits/second.
- A SL2/4 Finder Module will send (28 bits per 2 Finders x 4 channel links every 33ns) for a bandwidth of  $3.39 \times 10^9$  bits/second.

## Link Implementation

- The cables are made out of 3M 3600 series cable and 3M Mini D Ribbon 101XX-6000EC series plug and shell.
- Board connectors are the 3M .050" Mini D Ribbon N102XX-52XXVC Series Boardmount Thru Hole Right Angle Receptacle
- A SL1/3 Finder module will require the use of three 28 bit Channel Links, utilizing 18 cable pairs and 36 pin connectors for each copy of information between Finder and Linker Modules. There are two copies of information, thus two sets of three Channel Links and two connectors.
- A SL2/4 Finder module would require the use of four 28 bit Channel Links, utilizing 25 cable pairs and 50 pin connectors for each copy of information sent between Finder and Linker Modules. There are two copies of information, thus two sets of four Channel Links and two connectors.

## VMEbus Slave Interface

The FINDER will implement a modified version of a VMEbus slave interface. Only 32 bit aligned data transfers will be supported; these may be either single word transfers or block transfers. Only extended (32-bit) addressing modes will be supported. All modules will be assigned a unique geographical address through use of backplane pins on VME64extension backplane. FINDER modules will respond to the following address modifier codes: 09, and 0B.

### VME\_SLAVE

The VME\_SLAVE's function is to provide an interface between the VMEBUS and the circuitry on the Finder Board. The main functions is to verify that the module has been selected by the VME controller and decode the 32 bit address lines to determine the appropriate action of the Finder board. The module is selected by decoding the address, address modifier and geographical address lines. If the Finder board is selected the VME\_SLAVE chip will provide the chip select, address lines, write strobe and other various R/W control signals that are needed to perform R/W operations on all the other FPGAs on the Finder board. This decoding is based on the Memory map shown farther in the document.

The VME\_SLAVE chip was designed using the Veribest FPGA DesignView software package. The design is based in a graphical design file using schematic capture. The VME\_SLAVE design is implemented in a Xilinx FPGA device, specifically the XC4013E-3PQ240. See appendix# for the schematic. The chip is configured in Master Serial Mode using the Xilinx XC17256D Serial Configuration Prom. The 8-pin dip EEPROM is programmed using the DATA I/O programmer.

### CONTROL REGISTER

The CONTROL\_REGISTER provides a 32-bit register that is used to control functions in the Input Alignment, Finder and Pixel chips. The main functions are reset, loop and download. The register also provides control of selecting and providing a diagnostic clock signal to the FPGAs. All the programmable chips or chains of chips on the Finder board provide a 'done' signal to the Control Register that verify that the chips have been downloaded with a design. These signals can be read back through the CONTROL\_REGISTER. The design 'ANDs' all the 'done' signals and provides a Ready signal that is used to drive the green LED on the Finder module if all chips are downloaded. The loop bits output from the CONTROL\_REGISTER can be controlled via the CDF: RESET, HALT and B0 signals. The loop bits are registered in the chip with the use of the 33ns, 66ns and 132ns clocks.

The CONTROL\_REGISTER chip was designed using the Veribest FPGA DesignView software package. The design is based in a graphical design file using schematic capture. The CONTROL\_REGISTER design is implemented in a Xilinx FPGA device, specifically the XC4003EPQ100-3. See appendix# for the schematic. The chip is configured in Master Serial Mode using the Xilinx XC17128D Serial Configuration Prom. The 8-pin dip EEPROM is programmed using the DATA I/O programmer.

### ERROR REGISTER

The ERROR\_REGISTER is used to register and count the Word\_0 and Beam\_0 errors generated in the Input Alignment and Finder chips

The CLK\_DELAY register is also implemented in the ERROR\_REGISTER. The CLK\_DELAY register is an 8-bit register that is used to control a delay line that the CDF\_CLOCK signal is fed through before it is distributed on the Finder board.

The ERROR\_REGISTER chip was designed using the Veribest FPGA DesignView software package. The design is based in a graphical design file using schematic capture. The ERROR\_REGISTER design is implemented in a Xilinx FPGA device, specifically the XC4008EPQ208. See appendix# for the schematic. The chip is configured in Master Serial Mode using the Xilinx XC17256D Serial Configuration Prom. The 8-pin dip EEPROM is programmed using the DATA I/O programmer.

#### LEVEL 2 HEADER WORD

The LEVEL 2(L2) HEADER WORD block provides a means to identify the Finder board and also time at which a level 1 accept occurred with respect to the CDF Beam Zero signal. The L2 FPGA accepts a hard coded board type, serial number and geographical address to identify the board. An 8-bit R/W register identifies the Finder boards pipeline depth. When a LEVEL 2 buffer is read from the Finder board the first 32 bit word that is read comes from the L2 FPGA and is used as a 'Header' for that LEVEL 2 data.

The L2 chip was designed using the Veribest FPGA DesignView software package. The design is based in a graphical design file using schematic capture. The L2 design is implemented in a Xilinx FPGA device, specifically the XC4003EPQ100-3. See appendix# for the schematic. The chip is configured in Master Serial Mode using the Xilinx XC17128D Serial Configuration Prom. The 8-pin dip PROM is programmed using the DATA I/O programmer.

#### ALTERA DOWNLOAD Control

The ALTERA Download block consists of three Xilinx FPGAs which act as controllers and three Flash RAMs. The three sets are used to program the Pixel chip and the two different groups of Finder chips that are grouped by super layer. The design in the three Xilinx XC4003PQ100-3 are all the same. The design provides a means to control the Flash RAM that contains the download program. See appendix# for the schematic. The three XC4003 devices are configured in a Master/Slave Serial Mode using one Xilinx XC17128D Serial Configuration PROM for all three. The 8-pin dip PROM is programmed using the DATA I/O programmer.

#### XILINX\_DOWNLOAD Control

The XILINX Download block consists of a Xilinx FPGA which act as controller and Flash RAM. The data in the Flash RAM is used to program the 12 or 14 INPUT ALIGNMENT FPGAs. The design in the Xilinx XC4003PQ100-3 provides a means to control the Flash RAM. See appendix# for the schematic. The XC4003 device is configured in a Master Serial Mode using a Xilinx XC17128D Serial Configuration PROM. The 8-pin dip PROM is programmed using the DATA I/O programmer.

#### FLASH RAM Devices.

The Flash RAMs on the Finder board are used to hold the Finder, Pixel and Input Alignment chips designs. The Flash RAMs are AMD AM29F040B-90PC or AMD AM29F080B-90SC devices. The programming and erase operations for these devices can be found in their data sheets in appendix#.

#### ID\_PROM

The ID\_PROM contains the board serial number, board type and module description. The contents are in the form of 32 ASCII characters. The format should follow the CDF VME standard. The Finder ID\_PROM will contain a 4 bit serial number followed by a board type of '051' for a SL2/4 board or '052' for a SL1/3 board. The module description will be as follows: Finder SL2/4 Ver. 4.0 or Finder S11/3 Ver. 4.0.

## Memory Map

YY00 0000 Diagnostic Register (32 bits) (R/W)

YY00 0004 Control (R/W)

<b>Bit</b>	<b>Function</b>
31	Software Reset (R/W) Resets Loop Mode Bits
30	Software Clock (R/W)
29	Clock Select (R/W)
28	Alignment Section - Diagnostic Loop Mode (R/W) <b>1 - initiates looping pump out of data</b> <b>0 - allows VMEbus Read/Write of Diagnostic RAM</b>
27	Alignment Section - Force Program (R/W)
26	Alignment Section - Reset (R/W)
25	Undefined (R/W)
24	Finder Section - Force Program of SL1/2 Finders (R/W)
23	Finder Section - Force Program of SL3/4 Finders (R/W)
22	Finder Section - Reset (R/W)
21	Finder Section - Diagnostic Loop Mode (R/W)
20	Pixel Section - Force Program (R/W)
19	Pixel Section - Reset (R/W)
18	Pixel Section - Diagnostic Loop Mode (R/W)
17	Enable Loop Starts without /CDF_HALT signal (R/W)
16	Clear Alignment Section Errors (R/W)
15	Clear Finder Section Errors (R/W)
14	Enable VMEbus download of Flash RAM
13	Mask Error
12	Undefined (R/W)
11	Alignment Download Control (Xilinx) - Download Complete (R)
10	VME Slave Interface (Xilinx) - Download Complete (R)
9	Error Register (Xilinx) - Download Complete (R)
8	Finder/Pixel Download Control (Xilinx)-Download Complete (R)
7	Control Register (Xilinx) - Download Complete (R)
6	L2header (Xilinx) - Download Complete (R)
5	Alignment Section (Xilinx) - Download Complete (R)
4	Finder SL1 or SL2 Section (Altera) - Download Complete (R)
3	Finder SL3 or SL4 Section (Altera) - Download Complete (R)
2	Pixel Section (Altera) - Download Complete (R)
1	Ready (R)
0	Error (R)

Note: Bits 19, 22, 26 will reset the individual flip-flops in the FPGA devices. Bit 31 resets the Diagnostic Loop Mode bits 18, 21, 28 in the control register. The front panel reset button will cause all of the FPGA's on the board to be reconfigured with the contents of the appropriate Flash RAM

or serial EPROM. The Control register bits are not self-clearing, the user needs to set the bit and then clear the bit. i.e. in order to force a download or reset a chip the user should set the bit and then clear it. In the case of loop modes the user should set the bit during a loop test and then clear it after the loop test.

YY00 0008 Level 2 Header Word - Pipeline Length in 132ns time slices(**R/W**)  
Description: Depth of pipeline will be 42 minus the number of stages in pre-Finder FPGA circuitry.

<b><u>Bit</u></b>	<b><u>Function</u></b>
24-31	Programmable pipeline depth

YY00 000C Level 2 Header Word - Pipeline Offset in 132ns time slices(**R/W**)  
Description: Offset will equal number of stages in pre-Finder FPGA circuitry. This is the value that is subtracted from 42 to determine the Pipeline length above.

<b><u>Bit</u></b>	<b><u>Function</u></b>
24-31	Programmable pipeline depth offset

## YY00 0010 Alignment Section Word Zero Error (R)

<b>SL 1/3 Board</b>		<b>SL 2/4 Board</b>	
<b><u>Bit</u></b>	<b><u>Function</u></b>	<b><u>Bit</u></b>	<b><u>Function</u></b>
0	SL1 A	0	SL2 A
1	SL1 A split	1	SL2 B
2	SL1 B	2	SL2 C
3	SL1 B split	3	SL2 N+
4	SL1 N+	4	SL2 N-
5	SL1 N+ split	5	SL4 A
6	SL1 N-	6	SL4 B
7	SL1 N- split	7	SL4 C
8	SL3 A	8	SL4 D
9	SL3 B	9	SL4 E
10	SL3 C	10	SL4 N+
11	SL3 D	11	SL4 N-
12	SL3 N+	23:12	Undefined
13	SL3 N-	31:24	Total Error Count
23:14	Undefined		
31:24	Total Error Count		

## YY00 0014 Alignment Section Beam Zero Error (R)

<b>SL 1/3 Board</b>		<b>SL 2/4 Board</b>	
<b><u>Bit</u></b>	<b><u>Function</u></b>	<b><u>Bit</u></b>	<b><u>Function</u></b>
0	SL1 A	0	SL2 A
1	SL1 A split	1	SL2 B
2	SL1 B	2	SL2 C
3	SL1 B split	3	SL2 N+
4	SL1 N+	4	SL2 N-
5	SL1 N+ split	5	SL4 A
6	SL1 N-	6	SL4 B
7	SL1 N- split	7	SL4 C
8	SL3 A	8	SL4 D
9	SL3 B	9	SL4 E
10	SL3 C	10	SL4 N+
11	SL3 D	11	SL4 N-
12	SL3 N+	23:12	Undefined
13	SL3 N-	31:24	Total Error Count
23:14	Undefined		
31:24	Total Error Count		

YY00 0018 Finder Section Word Zero Error (R)

<b>SL 1/3 Board</b>		<b>SL 2/4 Board</b>	
<b><u>Bit</u></b>	<b><u>Function</u></b>	<b><u>Bit</u></b>	<b><u>Function</u></b>
0	SL1 A	0	SL2 A
1	SL1 A split	1	SL2 B
2	SL1 B	2	SL2 C
3	SL1 B split	3	SL4 A
4	SL3 A	4	SL4 B
5	SL3 B	5	SL4 C
6	SL3 C	6	SL4 D
7	SL3 D	7	SL4 E
23:8	Undefined	23:8	Undefined
31:24	Total Error Count	31:24	Total Error Count

YY00 001C Finder Section Beam Zero Error (R)

<b>SL 1/3 Board</b>		<b>SL 2/4 Board</b>	
<b><u>Bit</u></b>	<b><u>Function</u></b>	<b><u>Bit</u></b>	<b><u>Function</u></b>
0	SL1 A	0	SL2 A
1	SL1 A split	1	SL2 B
2	SL1 B	2	SL2 C
3	SL1 B split	3	SL4 A
4	SL3 A	4	SL4 B
5	SL3 B	5	SL4 C
6	SL3 C	6	SL4 D
7	SL3 D	7	SL4 E
23:8	Undefined	23:8	Undefined
31:24	Total Error Count	31:24	Total Error Count

YY00 0020	Clock Delay Register (R/W)
	<b><u>Bit</u></b> <b><u>Function</u></b>
	31-24 Clock delay value (1 count = 1ns)
	23-0 Undefined
YY00 0024	Trigger Delay Register (R/W) <b>!not used</b>
	<b><u>Bit</u></b> <b><u>Function</u></b>
	31-24 Clock delay value (1 count = 1ns)
	23-0 Undefined
YY00 0028	Flash RAM Bank Select Register (R/W)
	<b><u>Bit</u></b> <b><u>Function</u></b>
	31-24 Select value
	23-0 Undefined
	0 - Alignment Flash RAM
	1 - Finder 1/2 Flash RAM
	2 - Finder 3/4 Flash RAM
	3 - Pixel Flash RAM
YY10 0000 - YY10 007F	ID PROM (upper 8 bits)

Finder #	SL1/SL3 Module	SL2/SL4 Module
0	SL3 A (lowest phi)	SL4 A (lowest phi)
1	SL3 B	SL4 B
2	SL3 C	SL4 C
3	SL3 D (highest phi)	SL4 D
4	SL1 A (lowest phi)	SL4 E (highest phi)
5	SL1 B (highest phi)	SL2 A (lowest phi)
6	SL1 A' (split wire upgrade)	SL2 B
7	SL1 B' (split wire upgrade)	SL2 C (highest phi)

**DEAD WIRE REGISTERS (R/W)**

YY30 0000 Finder 0 Dead Wire Register (23:0)  
Maps to wires 23:0

YY30 0004 Finder 0 Dead Wire Register (23:0)  
Maps to wires 48:24

YY30 0008 Finder 0 Dead Wire Register (10:0)  
Maps to Neighbor Wires +N(10:0)

YY30 000C Finder 0 Dead Wire Register (10:0)  
Maps to Neighbor Wires -N(10:0)

YY30 0010 Finder 1 Dead Wire Register (23:0)  
Maps to wires 23:0

YY30 0014 Finder 1 Dead Wire Register (23:0)  
Maps to wires 48:24

YY30 0018 Finder 1 Dead Wire Register (10:0)  
Maps to Neighbor Wires +N(10:0)

YY30 001C Finder 1 Dead Wire Register (10:0)  
Maps to Neighbor Wires -N(10:0)

YY30 0020 Finder 2 Dead Wire Register (23:0)  
Maps to wires 23:0

YY30 0024 Finder 2 Dead Wire Register (23:0)  
Maps to wires 48:24

YY30 0028 Finder 2 Dead Wire Register (10:0)  
Maps to Neighbor Wires +N(10:0)

- YY30 002C Finder 2 Dead Wire Register (10:0)  
Maps to Neighbor Wires -N(10:0)
- YY30 0030 Finder 3 Dead Wire Register (23:0)  
Maps to wires 23:0
- YY30 0034 Finder 3 Dead Wire Register (23:0)  
Maps to wires 48:24
- YY30 0038 Finder 3 Dead Wire Register (10:0)  
Maps to Neighbor Wires +N(10:0)
- YY30 003C Finder 3 Dead Wire Register (10:0)  
Maps to Neighbor Wires -N(10:0)
- YY30 0040 Finder 4 Dead Wire Register (23:0)  
Maps to wires 23:0
- YY30 0044 Finder 4 Dead Wire Register (23:0)  
Maps to wires 48:24
- YY30 0048 Finder 4 Dead Wire Register (10:0)  
Maps to Neighbor Wires +N(10:0)
- YY30 004C Finder 4 Dead Wire Register (10:0)  
Maps to Neighbor Wires -N(10:0)
- YY30 0050 Finder 5 Dead Wire Register (23:0)  
Maps to wires 23:0
- YY30 0054 Finder 5 Dead Wire Register (23:0)  
Maps to wires 48:24
- YY30 0058 Finder 5 Dead Wire Register (10:0)  
Maps to Neighbor Wires +N(10:0)
- YY30 005C Finder 5 Dead Wire Register (10:0)  
Maps to Neighbor Wires -N(10:0)
- YY30 0060 Finder 6 Dead Wire Register (23:0)  
Maps to wires 23:0
- YY30 0064 Finder 6 Dead Wire Register (23:0)  
Maps to wires 48:24

YY30 0068 Finder 6 Dead Wire Register (10:0)  
Maps to Neighbor Wires +N(10:0)

YY30 006C Finder 6 Dead Wire Register (10:0)  
Maps to Neighbor Wires -N(10:0)

YY30 0070 Finder 7 Dead Wire Register (23:0)  
Maps to wires 23:0

YY30 0074 Finder 7 Dead Wire Register (23:0)  
Maps to wires 48:24

YY30 0078 Finder 7 Dead Wire Register (10:0)  
Maps to Neighbor Wires +N(10:0)

YY30 007C Finder 7 Dead Wire Register (10:0)  
Maps to Neighbor Wires -N(10:0)

**Finder Pipeline Depth REGISTERS (R/W)**

Description: Depth of pipeline in 33ns time slices. The value written into these registers should be 4 times the value written into the Level 2 Header Word - Pipeline Length register (YY00 0008) minus 1.

YY31 0000 Finder 0 Wire Pipeline Depth Register (7:0)

YY31 0004 Finder 1 Wire Pipeline Depth Register (7:0)

YY31 0008 Finder 2 Wire Pipeline Depth Register (7:0)

YY31 000C Finder 3 Wire Pipeline Depth Register (7:0)

YY31 0010 Finder 4 Wire Pipeline Depth Register (7:0)

YY31 0014 Finder 5 Wire Pipeline Depth Register (7:0)

YY31 0018 Finder 6 Wire Pipeline Depth Register (7:0)

YY31 001C Finder 7 Wire Pipeline Depth Register (7:0)

**Pixel Pipeline Depth REGISTERS (R/W)**

Description: Depth of pipeline in 33ns time slices. The value written into these registers should be the same as the Finder Pipeline Depth Registers. A function inside the Pixel chip allows for the delay between Finder and Pixel chips.

YY32 0000 Pixel 0 Segment Pipeline Depth Register (7:0)

YY32 0004 Pixel 1 Segment Pipeline Depth Register (7:0)

YY32 0008 Pixel 2 Segment Pipeline Depth Register (7:0)

YY32 000C Pixel 3 Segment Pipeline Depth Register (7:0)

\*\*\*\*\* These address spaces are only valid when the Xilinx Alignment FPGAs have been downloaded with diagnostic programs.

The Following Input Register Addressing is Valid for a SL13 Module.

**INPUT ALINGMENT DIAGNOSTIC Register (Ansley Cable data)  
(Read only)**

YY40 0000 SL1 -N Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0004 SL1 -N' Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0008 SL1 A Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 000C SL1 A' Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0010 SL1 B Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0014 SL1 B' Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0018 SL1 +N Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 001C SL1 +N' Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0020 SL3 -N Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0024 SL3 A Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0028 SL3 B Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 002C SL3 C Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0030 SL3 D Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))  
YY40 0034 SL3 +N Input Register (17:0) (B0\_marker, W0\_Marker, Data(15:0))

\*\*\*\*\* These address spaces are only valid when the Xilinx Alignment FPGAs have been downloaded with diagnostic programs.

The Following Input Register Addressing is Valid for a SL24 Module.

**INPUT ALINGMENT DIAGNOSTIC Register (Ansley Cable data)  
(Read Only)**

YY40 0000	SL2 -N Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 0004	SL2 A Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 0008	SL2 B Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 000C	SL2 C Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 0010	SL2 +N Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 0014	SL4 -N Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 0018	SL4 A Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 001C	SL4 B Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 0020	SL4 C Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 0024	SL4 D Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 0028	SL4 E Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))
YY40 002C	SL4 +N Input Register (17:0) (B0_marker, W0_Marker, Data(15:0))

**\*\*\*\*\* These address spaces are only valid when the Xilinx Alignment FPGAs have been downloaded with diagnostic programs. The Alignment FPGA's can: 1). Be used to capture data from the Ansley cables in the RAM block at a 22ns rate. 2). Be used as a RAM block that feeds wire information to the Finder at a 33ns rate.**

**All RAM blocks are 64 deep by 32 bits.**

**ALIGNMENT INPUT DIAGNOSTIC RAMs(Ansley Cable data) (32 bit Read only)**

**Bits 15..0 match up with the 16 wire group of Figure 8.**

**Bit 16 is W0\_Marker**

**Bit 17 is B0\_Marker**

**Bits 31..18 are not used**

**ALIGNMENT OUTPUT DIAGNOSTIC RAMs (Aligned wire data) (32 bit R/W)**

**Bits 23..0 match up with wires 23..0 of Figure 10.**

**Bit 24 is B0\_Marker**

**Bit 25 is Error\_Marker**

**Bit 26 is W0\_Marker**

**Bit 27 is Operate ! Operate is only used to feed the Finder**

**Bits 31..28 are not used**

**The Following Alignment DIAGNOSTIC RAM Addressing is Valid for a SL13 Module.**

YY50 0000 - YY50 00FC	SL1 -N DIAGNOSTIC RAM (R/W)
YY51 0000 - YY51 00FC	SL1 -N' DIAGNOSTIC RAM (R/W)
YY52 0000 - YY52 00FC	SL1 A DIAGNOSTIC RAM (R/W)
YY53 0000 - YY53 00FC	SL1 A' 1 DIAGNOSTIC RAM (R/W)
YY54 0000 - YY54 00FC	SL1 B DIAGNOSTIC RAM (R/W)
YY55 0000 - YY55 00FC	SL1 B' DIAGNOSTIC RAM (R/W)
YY56 0000 - YY56 00FC	SL1 +N DIAGNOSTIC RAM (R/W)
YY57 0000 - YY57 00FC	SL1 +N' DIAGNOSTIC RAM (R/W)
YY58 0000 - YY58 00FC	SL3 -N DIAGNOSTIC RAM (R/W)
YY59 0000 - YY59 00FC	SL3 A DIAGNOSTIC RAM (R/W)
YY5A 0000 - YY5A 00FC	SL3 B DIAGNOSTIC RAM (R/W)
YY5B 0000 - YY5B 00FC	SL3 C DIAGNOSTIC RAM (R/W)

YY5C 0000 - YY5C 00FC SL3 D DIAGNOSTIC RAM (R/W)

YY5D 0000 - YY5D 00FC SL3 +N DIAGNOSTIC RAM (R/W)

\*\*\*\*\* These address spaces are only valid when the Xilinx Alignment FPGAs have been downloaded with Diagnostic programs.

The Following Alignment DIAGNOSTIC RAM Addressing is Valid for a SL24 Module.

YY50 0000 - YY50 00FC	SL2 -N DIAGNOSTIC RAM (R/W)
YY51 0000 - YY51 00FC	SL2 A DIAGNOSTIC RAM (R/W)
YY52 0000 - YY52 00FC	SL2 B 1 DIAGNOSTIC RAM (R/W)
YY53 0000 - YY53 00FC	SL2 C DIAGNOSTIC RAM (R/W)
YY54 0000 - YY54 00FC	SL2 +N DIAGNOSTIC RAM (R/W)
YY55 0000 - YY55 00FC	SL4 -N DIAGNOSTIC RAM (R/W)
YY56 0000 - YY56 00FC	SL4 A DIAGNOSTIC RAM (R/W)
YY57 0000 - YY57 00FC	SL4 B DIAGNOSTIC RAM (R/W)
YY58 0000 - YY58 00FC	SL4 C DIAGNOSTIC RAM (R/W)
YY59 0000 - YY59 00FC	SL4 D DIAGNOSTIC RAM (R/W)
YY5A 0000 - YY5A 00FC	SL4 E DIAGNOSTIC RAM (R/W)
YY5B 0000 - YY5B 00FC	SL4 +N DIAGNOSTIC RAM (R/W)

**\*\*\*\*\* These address spaces are only valid when the Altera Finder CPLDs have been downloaded with diagnostic programs. In Diagnostic mode the Finder can: 1). Be used to capture the wire information sent from the Alignment FPGA. 2). Be used as a RAM block to send pixel information to the Pixel CPLDs.**

**FINDER INPUT DIAGNOSTIC RAMs(Aligned wire data) (32 bit R/W)**

Captures wire information sent from the Alignment FPGA.

**Bits 23..0 match up with wires 23..0 of Figure 10.**

**Bit 24 is Error\_Marker**

**Bit 25 is W0\_Marker**

**Bit 26 is B0\_Marker**

**Bits 31..27 are not used**

**FINDER INPUT DIAGNOSTIC RAMs(Aligned wire data) (32 bit R/W)**

Captures wire information sent from the Core and Neighbor Alignment FPGAs.

**Bits 1..0 match up with negative neighbor wires 11..10 of Figure 10.**

**Bits 10..2 match up with negative neighbor wires 23..15 of Figure 10.**

**Bits 19..11 match up with positive neighbor wires 8..0 of Figure 10.**

**Bits 21..20 match up with positive neighbor wires 13..12 of Figure 10.**

**Bit 22 is W0\_Marker negative neighbor**

**Bit 23 is W0\_Marker positive neighbor**

**Bit 24 is B0\_Marker negative neighbor**

**Bit 25 is B0\_Marker positive neighbor**

**Bit 26 is Error\_Marker negative neighbor**

**Bit 26 is Error\_Marker positive neighbor**

**Bits 31..28 are not used**

**FINDER OUTPUT DIAGNOSTIC RAMs (Pixel data) (32 bit R/W)**

Pixel data that is sent to the Pixel chips. Format is the same as the normal Finder algorithm.

**Bits 11..0 Pixel data**

**Bit 12 is Error\_Marker**

**Bit 13 is W0\_Marker**

**Bit 14 is B0\_Marker**

**Bits 31..15 are not used**

**The Following Finder DIAGNOSTIC RAM Addressing is Valid for a SL13 Module. All RAM blocks are 256 deep by 32 bits.**

YY60 0000 - YY60 03FC SL1 A Diagnostic RAM (15:0) (R/W)

YY61 0000 - YY61 03FC SL1 B Diagnostic RAM (15:0) (R/W)

YY62 0000 - YY62 03FC SL1 A' Diagnostic RAM (15:0) (R/W)

YY63 0000 - YY63 03FC SL1 B' Diagnostic RAM (15:0) (R/W)

YY64 0000 - YY64 03FC SL3 A Diagnostic RAM (15:0) (R/W)

YY65 0000 - YY65 03FC SL3 B Diagnostic RAM (15:0) (R/W)

YY66 0000 - YY66 03FC SL3 C Diagnostic RAM (15:0) (R/W)

YY67 0000 - YY67 03FC SL3 D Diagnostic RAM (15:0) (R/W)

**The Following Finder DIAGNOSTIC RAM Addressing is Valid for a SL24 Module.**

YY60 0000 - YY60 03FC SL2 A Diagnostic RAM (15:0) (R/W)

YY61 0000 - YY61 03FC SL2 B Diagnostic RAM (15:0) (R/W)

YY62 0000 - YY62 03FC SL2 C Diagnostic RAM (15:0) (R/W)

YY63 0000 - YY63 03FC SL4 A Diagnostic RAM (15:0) (R/W)

YY64 0000 - YY64 03FC SL4 B Diagnostic RAM (15:0) (R/W)

YY65 0000 - YY65 03FC SL4 C Diagnostic RAM (15:0) (R/W)

YY66 0000 - YY66 03FC SL4 D Diagnostic RAM (15:0) (R/W)

YY67 0000 - YY67 03FC SL4 E Diagnostic RAM (15:0) (R/W)

\*\*\*\*\* These address spaces are only valid when the Altera Pixel CPLDs have been downloaded with diagnostic programs.

In Diagnostic mode the Pixel chips can: 1). Be used to capture the pixel information sent from the Finder. 2). Be used as a RAM block to send pixel information to the Channel Link devices.

**Pixel chip INPUT DIAGNOSTIC RAMs (Pixel data) (32 bit R/W)**

If the Pixel chip is used to capture data from the Finder the bit format is as follows:

**Bits 15..0 match up with Finder N [15..0] output**

Bit 11..0 is pixel information

Bit 12 is Beam\_0 marker

Bit 13 is Word\_0 marker

Bit 14 is Error Marker

Bit 15 is Start

**Bits 31..16 match up with Finder N +1 [15..0] output**

Bit 27..16 is pixel information

Bit 28 is Beam\_0 marker

Bit 29 is Word\_0 marker

Bit 30 is Error Marker

Bit 31 is Start

**Pixel chip OUTPUT DIAGNOSTIC RAMs(Pixel data) (32 bit R/W)**

If the Pixel chip is used to drive data to the Channel Link devices the bit format is the same as the normal algorithm of the Pixel chip:

**Bits 11-0 match up with pixels 11-0 of Finder N.**

**Bit 23-12 match up with pixels 11-0 of Finder N + 1.**

**Bit 24 is 'OR'd combination of Finder N -Beam\_0 marker and Finder N+1 -Beam\_0 marker.**

**Bit 25 is 'OR'd combination of Finder N -Word\_0 marker and Finder N+1 -Word\_0 marker.**

**Bit 26 is 'OR'd combination of Finder N -Error marker and Finder N+1 -Error marker.**

**Bit 27 is 'OR'd combination of Finder N -Start marker and Finder N+1 -Start marker.**

**Bits 31..28 are not used.**

**The Following DIAGNOSTIC RAM Addressing is Valid for a SL13 Module.**

YY70 0000 - YY70 03FC SL1 Pixel Data (27:0) (R/W)

YY71 0000 - YY71 03FC SL3 B,A Pixel Data (27:0) (R/W)

YY72 0000 - YY72 03FC SL3 D,C Pixel Data (27:0) (R/W)

**The Following DIAGNOSTIC RAM Addressing is Valid for a SL24 Module.**

YY70 0000 - YY70 0CFC SL2 B,A Pixel Data (27:0) (R/W)

YY71 0000 - YY71 0CFC SL4 E, SL2 C Pixel Data (27:0) (R/W)

YY72 0000 - YY72 0CFC SL4 B,A Pixel Data (27:0) **(R/W)**

YY73 0000 - YY73 0CFC SL4 D,C Pixel Data (27:0) **(R/W)**

**L2 Buffer Space** (NOTE: All L2 Buffers are Read Only)

YY80 0000	L2 Buffer 0 Header Word																						
	<table> <thead> <tr> <th><u>Bit</u></th> <th><u>Function</u></th> </tr> </thead> <tbody> <tr> <td>0-7</td> <td>Bunch ID: 8 bit counter from Bunch Zero</td> </tr> <tr> <td>8-12</td> <td>Geographical Address</td> </tr> <tr> <td>13-22</td> <td>Module Serial Number</td> </tr> <tr> <td>23-31</td> <td>Module Type</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Function</u>	0-7	Bunch ID: 8 bit counter from Bunch Zero	8-12	Geographical Address	13-22	Module Serial Number	23-31	Module Type												
<u>Bit</u>	<u>Function</u>																						
0-7	Bunch ID: 8 bit counter from Bunch Zero																						
8-12	Geographical Address																						
13-22	Module Serial Number																						
23-31	Module Type																						
YY80 0004	Finder 0 - Cell 0,1 Segment Information (segment info passed on bits 30:0)																						
	<table> <thead> <tr> <th><u>Bit</u></th> <th><u>Function</u></th> </tr> </thead> <tbody> <tr> <td>0-11</td> <td>Cell 0 - Pixel (11:0)</td> </tr> <tr> <td>12</td> <td>Cell 0 - B0 Marker</td> </tr> <tr> <td>13</td> <td>Cell 0 - W0 Marker</td> </tr> <tr> <td>14</td> <td>Cell 0 - Error Marker</td> </tr> <tr> <td>15</td> <td>Undefined</td> </tr> <tr> <td>16-27</td> <td>Cell 1 - Pixel (11:0)</td> </tr> <tr> <td>28</td> <td>Cell 1 - B0 Marker</td> </tr> <tr> <td>29</td> <td>Cell 1 - W0 Marker</td> </tr> <tr> <td>30</td> <td>Cell 1 - Error Marker</td> </tr> <tr> <td>31</td> <td>Undefined</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Function</u>	0-11	Cell 0 - Pixel (11:0)	12	Cell 0 - B0 Marker	13	Cell 0 - W0 Marker	14	Cell 0 - Error Marker	15	Undefined	16-27	Cell 1 - Pixel (11:0)	28	Cell 1 - B0 Marker	29	Cell 1 - W0 Marker	30	Cell 1 - Error Marker	31	Undefined
<u>Bit</u>	<u>Function</u>																						
0-11	Cell 0 - Pixel (11:0)																						
12	Cell 0 - B0 Marker																						
13	Cell 0 - W0 Marker																						
14	Cell 0 - Error Marker																						
15	Undefined																						
16-27	Cell 1 - Pixel (11:0)																						
28	Cell 1 - B0 Marker																						
29	Cell 1 - W0 Marker																						
30	Cell 1 - Error Marker																						
31	Undefined																						
YY80 0008	Finder 0 - Cell 2,3 Segment Information																						
YY80 000C	Finder 1 - Cell 0,1 Segment Information																						
YY80 0010	Finder 1 - Cell 2,3 Segment Information																						
YY80 0014	Finder 2 - Cell 0,1 Segment Information																						
YY80 0018	Finder 2 - Cell 2,3 Segment Information																						
YY80 001C	Finder 3 - Cell 0,1 Segment Information																						
YY80 0020	Finder 3 - Cell 2,3 Segment Information																						
YY80 0024	Finder 4 - Cell 0,1 Segment Information																						
YY80 0028	Finder 4 - Cell 2,3 Segment Information																						
YY80 002C	Finder 5 - Cell 0,1 Segment Information																						
YY80 0030	Finder 5 - Cell 2,3 Segment Information																						
YY80 0034	Finder 6 - Cell 0,1 Segment Information																						
YY80 0038	Finder 6 - Cell 2,3 Segment Information																						
YY80 003C	Finder 7 - Cell 0,1 Segment Information																						
YY80 0040	Finder 7 - Cell 2,3 Segment Information																						

YY80 0044 Finder 0 - Wire Cell 0 Information (wire info passed on bits 26:0)

**Bit**    **Function**

0-11    Wire Data Prompt (11:0)

12-23    Wire Data Delay (11:0)

24        B0 Marker

25        W0 Marker

26        Error Marker

YY80 0048 Finder 0 - Wire Cell 1 Information  
 YY80 004C Finder 0 - Wire Cell 2 Information  
 YY80 0050 Finder 0 - Wire Cell 3 Information

YY80 0054 Finder 1 - Wire Cell 0 Information  
 YY80 0058 Finder 1 - Wire Cell 1 Information  
 YY80 005C Finder 1 - Wire Cell 2 Information  
 YY80 0060 Finder 1 - Wire Cell 3 Information

YY80 0064 Finder 2 - Wire Cell 0 Information  
 YY80 0068 Finder 2 - Wire Cell 1 Information  
 YY80 006C Finder 2 - Wire Cell 2 Information  
 YY80 0070 Finder 2 - Wire Cell 3 Information

YY80 0074 Finder 3 - Wire Cell 0 Information  
 YY80 0078 Finder 3 - Wire Cell 1 Information  
 YY80 007C Finder 3 - Wire Cell 2 Information  
 YY80 0080 Finder 3 - Wire Cell 3 Information

YY80 0084 Finder 4 - Wire Cell 0 Information  
 YY80 0088 Finder 4 - Wire Cell 1 Information  
 YY80 008C Finder 4 - Wire Cell 2 Information  
 YY80 0090 Finder 4 - Wire Cell 3 Information

YY80 0094 Finder 5 - Wire Cell 0 Information  
 YY80 0098 Finder 5 - Wire Cell 1 Information  
 YY80 009C Finder 5 - Wire Cell 2 Information  
 YY80 00A0 Finder 5 - Wire Cell 3 Information

YY80 00A4 Finder 6 - Wire Cell 0 Information  
 YY80 00A8 Finder 6 - Wire Cell 1 Information  
 YY80 00AC Finder 6 - Wire Cell 2 Information  
 YY80 00B0 Finder 6 - Wire Cell 3 Information

YY80 00B4 Finder 7 - Wire Cell 0 Information  
 YY80 00B8 Finder 7 - Wire Cell 1 Information  
 YY80 00BC Finder 7 - Wire Cell 2 Information  
 YY80 00B8 Finder 7 - Wire Cell 3 Information

**L2 Buffer Space** (NOTE: All L2 Buffers are Read Only)

YY90 0000	L2 Buffer 1 Header Word																						
	<table> <thead> <tr> <th><u>Bit</u></th> <th><u>Function</u></th> </tr> </thead> <tbody> <tr> <td>0-7</td> <td>Bunch ID: 8 bit counter from Bunch Zero</td> </tr> <tr> <td>8-12</td> <td>Geographical Address</td> </tr> <tr> <td>13-22</td> <td>Module Serial Number</td> </tr> <tr> <td>23-31</td> <td>Module Type</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Function</u>	0-7	Bunch ID: 8 bit counter from Bunch Zero	8-12	Geographical Address	13-22	Module Serial Number	23-31	Module Type												
<u>Bit</u>	<u>Function</u>																						
0-7	Bunch ID: 8 bit counter from Bunch Zero																						
8-12	Geographical Address																						
13-22	Module Serial Number																						
23-31	Module Type																						
YY90 0004	Finder 0 - Cell 0,1 Segment Information (segment info passed on bits 30:0)																						
	<table> <thead> <tr> <th><u>Bit</u></th> <th><u>Function</u></th> </tr> </thead> <tbody> <tr> <td>0-11</td> <td>Cell 0 - Pixel (11:0)</td> </tr> <tr> <td>12</td> <td>Cell 0 - B0 Marker</td> </tr> <tr> <td>13</td> <td>Cell 0 - W0 Marker</td> </tr> <tr> <td>14</td> <td>Cell 0 - Error Marker</td> </tr> <tr> <td>15</td> <td>Undefined</td> </tr> <tr> <td>16-27</td> <td>Cell 1 - Pixel (11:0)</td> </tr> <tr> <td>28</td> <td>Cell 1 - B0 Marker</td> </tr> <tr> <td>29</td> <td>Cell 1 - W0 Marker</td> </tr> <tr> <td>30</td> <td>Cell 1 - Error Marker</td> </tr> <tr> <td>31</td> <td>Undefined</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Function</u>	0-11	Cell 0 - Pixel (11:0)	12	Cell 0 - B0 Marker	13	Cell 0 - W0 Marker	14	Cell 0 - Error Marker	15	Undefined	16-27	Cell 1 - Pixel (11:0)	28	Cell 1 - B0 Marker	29	Cell 1 - W0 Marker	30	Cell 1 - Error Marker	31	Undefined
<u>Bit</u>	<u>Function</u>																						
0-11	Cell 0 - Pixel (11:0)																						
12	Cell 0 - B0 Marker																						
13	Cell 0 - W0 Marker																						
14	Cell 0 - Error Marker																						
15	Undefined																						
16-27	Cell 1 - Pixel (11:0)																						
28	Cell 1 - B0 Marker																						
29	Cell 1 - W0 Marker																						
30	Cell 1 - Error Marker																						
31	Undefined																						
YY90 0008	Finder 0 - Cell 2,3 Segment Information																						
YY90 000C	Finder 1 - Cell 0,1 Segment Information																						
YY90 0010	Finder 1 - Cell 2,3 Segment Information																						
YY90 0014	Finder 2 - Cell 0,1 Segment Information																						
YY90 0018	Finder 2 - Cell 2,3 Segment Information																						
YY90 001C	Finder 3 - Cell 0,1 Segment Information																						
YY90 0020	Finder 3 - Cell 2,3 Segment Information																						
YY90 0024	Finder 4 - Cell 0,1 Segment Information																						
YY90 0028	Finder 4 - Cell 2,3 Segment Information																						
YY90 002C	Finder 5 - Cell 0,1 Segment Information																						
YY90 0030	Finder 5 - Cell 2,3 Segment Information																						
YY90 0034	Finder 6 - Cell 0,1 Segment Information																						
YY90 0038	Finder 6 - Cell 2,3 Segment Information																						
YY90 003C	Finder 7 - Cell 0,1 Segment Information																						
YY90 0040	Finder 7 - Cell 2,3 Segment Information																						

YY90 0044 Finder 0 - Wire Cell 0 Information (wire info passed on bits 26:0)

**Bit      Function**

0-11 Wire Data Prompt (11:0)

12-23 Wire Data Delay (11:0)

24 B0 Marker

25 W0 Marker

26 Error Marker

YY90 0048 Finder 0 - Wire Cell 1 Information

YY90 004C Finder 0 - Wire Cell 2 Information

YY90 0050 Finder 0 - Wire Cell 3 Information

YY90 0054 Finder 1 - Wire Cell 0 Information

YY90 0058 Finder 1 - Wire Cell 1 Information

YY90 005C Finder 1 - Wire Cell 2 Information

YY90 0060 Finder 1 - Wire Cell 3 Information

YY90 0064 Finder 2 - Wire Cell 0 Information

YY90 0068 Finder 2 - Wire Cell 1 Information

YY90 006C Finder 2 - Wire Cell 2 Information

YY90 0070 Finder 2 - Wire Cell 3 Information

YY90 0074 Finder 3 - Wire Cell 0 Information

YY90 0078 Finder 3 - Wire Cell 1 Information

YY90 007C Finder 3 - Wire Cell 2 Information

YY90 0080 Finder 3 - Wire Cell 3 Information

YY90 0084 Finder 4 - Wire Cell 0 Information

YY90 0088 Finder 4 - Wire Cell 1 Information

YY90 008C Finder 4 - Wire Cell 2 Information

YY90 0090 Finder 4 - Wire Cell 3 Information

YY90 0094 Finder 5 - Wire Cell 0 Information

YY90 0098 Finder 5 - Wire Cell 1 Information

YY90 009C Finder 5 - Wire Cell 2 Information

YY90 00A0 Finder 5 - Wire Cell 3 Information

YY90 00A4 Finder 6 - Wire Cell 0 Information

YY90 00A8 Finder 6 - Wire Cell 1 Information

YY90 00AC Finder 6 - Wire Cell 2 Information

YY90 00B0 Finder 6 - Wire Cell 3 Information

YY90 00B4 Finder 7 - Wire Cell 0 Information

YY90 00B8 Finder 7 - Wire Cell 1 Information

YY90 00BC Finder 7 - Wire Cell 2 Information

YY90 00B8 Finder 7 - Wire Cell 3 Information

**L2 Buffer Space** (NOTE: All L2 Buffers are Read Only)

YYA0 0000	L2 Buffer 2 Header Word																						
	<table> <thead> <tr> <th><u>Bit</u></th> <th><u>Function</u></th> </tr> </thead> <tbody> <tr> <td>0-7</td> <td>Bunch ID: 8 bit counter from Bunch Zero</td> </tr> <tr> <td>8-12</td> <td>Geographical Address</td> </tr> <tr> <td>13-22</td> <td>Module Serial Number</td> </tr> <tr> <td>23-31</td> <td>Module Type</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Function</u>	0-7	Bunch ID: 8 bit counter from Bunch Zero	8-12	Geographical Address	13-22	Module Serial Number	23-31	Module Type												
<u>Bit</u>	<u>Function</u>																						
0-7	Bunch ID: 8 bit counter from Bunch Zero																						
8-12	Geographical Address																						
13-22	Module Serial Number																						
23-31	Module Type																						
YYA0 0004	Finder 0 - Cell 0,1 Segment Information (segment info passed on bits 30:0)																						
	<table> <thead> <tr> <th><u>Bit</u></th> <th><u>Function</u></th> </tr> </thead> <tbody> <tr> <td>0-11</td> <td>Cell 0 - Pixel (11:0)</td> </tr> <tr> <td>12</td> <td>Cell 0 - B0 Marker</td> </tr> <tr> <td>13</td> <td>Cell 0 - W0 Marker</td> </tr> <tr> <td>14</td> <td>Cell 0 - Error Marker</td> </tr> <tr> <td>15</td> <td>Undefined</td> </tr> <tr> <td>16-27</td> <td>Cell 1 - Pixel (11:0)</td> </tr> <tr> <td>28</td> <td>Cell 1 - B0 Marker</td> </tr> <tr> <td>29</td> <td>Cell 1 - W0 Marker</td> </tr> <tr> <td>30</td> <td>Cell 1 - Error Marker</td> </tr> <tr> <td>31</td> <td>Undefined</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Function</u>	0-11	Cell 0 - Pixel (11:0)	12	Cell 0 - B0 Marker	13	Cell 0 - W0 Marker	14	Cell 0 - Error Marker	15	Undefined	16-27	Cell 1 - Pixel (11:0)	28	Cell 1 - B0 Marker	29	Cell 1 - W0 Marker	30	Cell 1 - Error Marker	31	Undefined
<u>Bit</u>	<u>Function</u>																						
0-11	Cell 0 - Pixel (11:0)																						
12	Cell 0 - B0 Marker																						
13	Cell 0 - W0 Marker																						
14	Cell 0 - Error Marker																						
15	Undefined																						
16-27	Cell 1 - Pixel (11:0)																						
28	Cell 1 - B0 Marker																						
29	Cell 1 - W0 Marker																						
30	Cell 1 - Error Marker																						
31	Undefined																						
YYA0 0008	Finder 0 - Cell 2,3 Segment Information																						
YYA0 000C	Finder 1 - Cell 0,1 Segment Information																						
YYA0 0010	Finder 1 - Cell 2,3 Segment Information																						
YYA0 0014	Finder 2 - Cell 0,1 Segment Information																						
YYA0 0018	Finder 2 - Cell 2,3 Segment Information																						
YYA0 001C	Finder 3 - Cell 0,1 Segment Information																						
YYA0 0020	Finder 3 - Cell 2,3 Segment Information																						
YYA0 0024	Finder 4 - Cell 0,1 Segment Information																						
YYA0 0028	Finder 4 - Cell 2,3 Segment Information																						
YYA0 002C	Finder 5 - Cell 0,1 Segment Information																						
YYA0 0030	Finder 5 - Cell 2,3 Segment Information																						
YYA0 0034	Finder 6 - Cell 0,1 Segment Information																						
YYA0 0038	Finder 6 - Cell 2,3 Segment Information																						
YYA0 003C	Finder 7 - Cell 0,1 Segment Information																						
YYA0 0040	Finder 7 - Cell 2,3 Segment Information																						

YYA0 0044 Finder 0 - Wire Cell 0 Information (wire info passed on bits 26:0)

**Bit      Function**

0-11    Wire Data Prompt (11:0)

12-23   Wire Data Delay (11:0)

24      B0 Marker

25      W0 Marker

26      Error Marker

YYA0 0048 Finder 0 - Wire Cell 1 Information

YYA0 004C Finder 0 - Wire Cell 2 Information

YYA0 0050 Finder 0 - Wire Cell 3 Information

YYA0 0054 Finder 1 - Wire Cell 0 Information

YYA0 0058 Finder 1 - Wire Cell 1 Information

YYA0 005C Finder 1 - Wire Cell 2 Information

YYA0 0060 Finder 1 - Wire Cell 3 Information

YYA0 0064 Finder 2 - Wire Cell 0 Information

YYA0 0068 Finder 2 - Wire Cell 1 Information

YYA0 006C Finder 2 - Wire Cell 2 Information

YYA0 0070 Finder 2 - Wire Cell 3 Information

YYA0 0074 Finder 3 - Wire Cell 0 Information

YYA0 0078 Finder 3 - Wire Cell 1 Information

YYA0 007C Finder 3 - Wire Cell 2 Information

YYA0 0080 Finder 3 - Wire Cell 3 Information

YYA0 0084 Finder 4 - Wire Cell 0 Information

YYA0 0088 Finder 4 - Wire Cell 1 Information

YYA0 008C Finder 4 - Wire Cell 2 Information

YYA0 0090 Finder 4 - Wire Cell 3 Information

YYA0 0094 Finder 5 - Wire Cell 0 Information

YYA0 0098 Finder 5 - Wire Cell 1 Information

YYA0 009C Finder 5 - Wire Cell 2 Information

YYA0 00A0 Finder 5 - Wire Cell 3 Information

YYA0 00A4 Finder 6 - Wire Cell 0 Information

YYA0 00A8 Finder 6 - Wire Cell 1 Information

YYA0 00AC Finder 6 - Wire Cell 2 Information

YYA0 00B0 Finder 6 - Wire Cell 3 Information

YYA0 00B4 Finder 7 - Wire Cell 0 Information

YYA0 00B8 Finder 7 - Wire Cell 1 Information

YYA0 00BC Finder 7 - Wire Cell 2 Information

YYA0 00B8 Finder 7 - Wire Cell 3 Information

**L2 Buffer Space** (NOTE: All L2 Buffers are Read Only)

YYB0 0000 L2 Buffer 3 Header Word

<u>Bit</u>	<u>Function</u>
0-7	Bunch ID: 8 bit counter from Bunch Zero
8-12	Geographical Address
13-22	Module Serial Number
23-31	Module Type

YYB0 0004 Finder 0 - Cell 0,1 Segment Information (segment info passed on bits 30:0)

<u>Bit</u>	<u>Function</u>
0-11	Cell 0 - Pixel (11:0)
12	Cell 0 - B0 Marker
13	Cell 0 - W0 Marker
14	Cell 0 - Error Marker
15	Undefined
16-27	Cell 1 - Pixel (11:0)
28	Cell 1 - B0 Marker
29	Cell 1 - W0 Marker
30	Cell 1 - Error Marker
31	Undefined

YYB0 0008 Finder 0 - Cell 2,3 Segment Information

YYB0 000C Finder 1 - Cell 0,1 Segment Information

YYB0 0010 Finder 1 - Cell 2,3 Segment Information

YYB0 0014 Finder 2 - Cell 0,1 Segment Information

YYB0 0018 Finder 2 - Cell 2,3 Segment Information

YYB0 001C Finder 3 - Cell 0,1 Segment Information

YYB0 0020 Finder 3 - Cell 2,3 Segment Information

YYB0 0024 Finder 4 - Cell 0,1 Segment Information

YYB0 0028 Finder 4 - Cell 2,3 Segment Information

YYB0 002C Finder 5 - Cell 0,1 Segment Information

YYB0 0030 Finder 5 - Cell 2,3 Segment Information

YYB0 0034 Finder 6 - Cell 0,1 Segment Information

YYB0 0038 Finder 6 - Cell 2,3 Segment Information

YYB0 003C Finder 7 - Cell 0,1 Segment Information

YYB0 0040 Finder 7 - Cell 2,3 Segment Information

YYB0 0044 Finder 0 - Wire Cell 0 Information (wire info passed on bits 26:0)

**Bit      Function**

0-11 Wire Data Prompt (11:0)

12-23 Wire Data Delay (11:0)

24 B0 Marker

25 W0 Marker

26 Error Marker

YYB0 0048 Finder 0 - Wire Cell 1 Information

YYB0 004C Finder 0 - Wire Cell 2 Information

YYB0 0050 Finder 0 - Wire Cell 3 Information

YYB0 0054 Finder 1 - Wire Cell 0 Information

YYB0 0058 Finder 1 - Wire Cell 1 Information

YYB0 005C Finder 1 - Wire Cell 2 Information

YYB0 0060 Finder 1 - Wire Cell 3 Information

YYB0 0064 Finder 2 - Wire Cell 0 Information

YYB0 0068 Finder 2 - Wire Cell 1 Information

YYB0 006C Finder 2 - Wire Cell 2 Information

YYB0 0070 Finder 2 - Wire Cell 3 Information

YYB0 0074 Finder 3 - Wire Cell 0 Information

YYB0 0078 Finder 3 - Wire Cell 1 Information

YYB0 007C Finder 3 - Wire Cell 2 Information

YYB0 0080 Finder 3 - Wire Cell 3 Information

YYB0 0084 Finder 4 - Wire Cell 0 Information

YYB0 0088 Finder 4 - Wire Cell 1 Information

YYB0 008C Finder 4 - Wire Cell 2 Information

YYB0 0090 Finder 4 - Wire Cell 3 Information

YYB0 0094 Finder 5 - Wire Cell 0 Information

YYB0 0098 Finder 5 - Wire Cell 1 Information

YYB0 009C Finder 5 - Wire Cell 2 Information

YYB0 00A0 Finder 5 - Wire Cell 3 Information

YYB0 00A4 Finder 6 - Wire Cell 0 Information

YYB0 00A8 Finder 6 - Wire Cell 1 Information

YYB0 00AC Finder 6 - Wire Cell 2 Information

YYB0 00B0 Finder 6 - Wire Cell 3 Information

YYB0 00B4 Finder 7 - Wire Cell 0 Information

YYB0 00B8 Finder 7 - Wire Cell 1 Information

YYB0 00BC Finder 7 - Wire Cell 2 Information

YYB0 00B8 Finder 7 - Wire Cell 3 Information

YYC0 0000 - YYFF FFFC Flash RAM Data Register (R/W)  
**(8-Megabit Flash Ram for download configuration for  
the Finder Module's PLDs{Alignment, Finder & Pixel Chips})**

YY is the VME geographical address of the Finder board.

References

- [1] xft design report
- [2] COT design report
- [3] L2 design report
- [4] COT TDC Mezzanine Module
- [5] COT TDC Transition Module
- [6] Finder Transition Modules
- [7] Linker Modules
- [8] Linker Transition Modules
- [9] XTRP

## **Appendix A**

### **TDC Transition Module Specification**

# Note on TDC Mezzanine pin out for the XFT System

August 27, 1997

Theresa Shaw

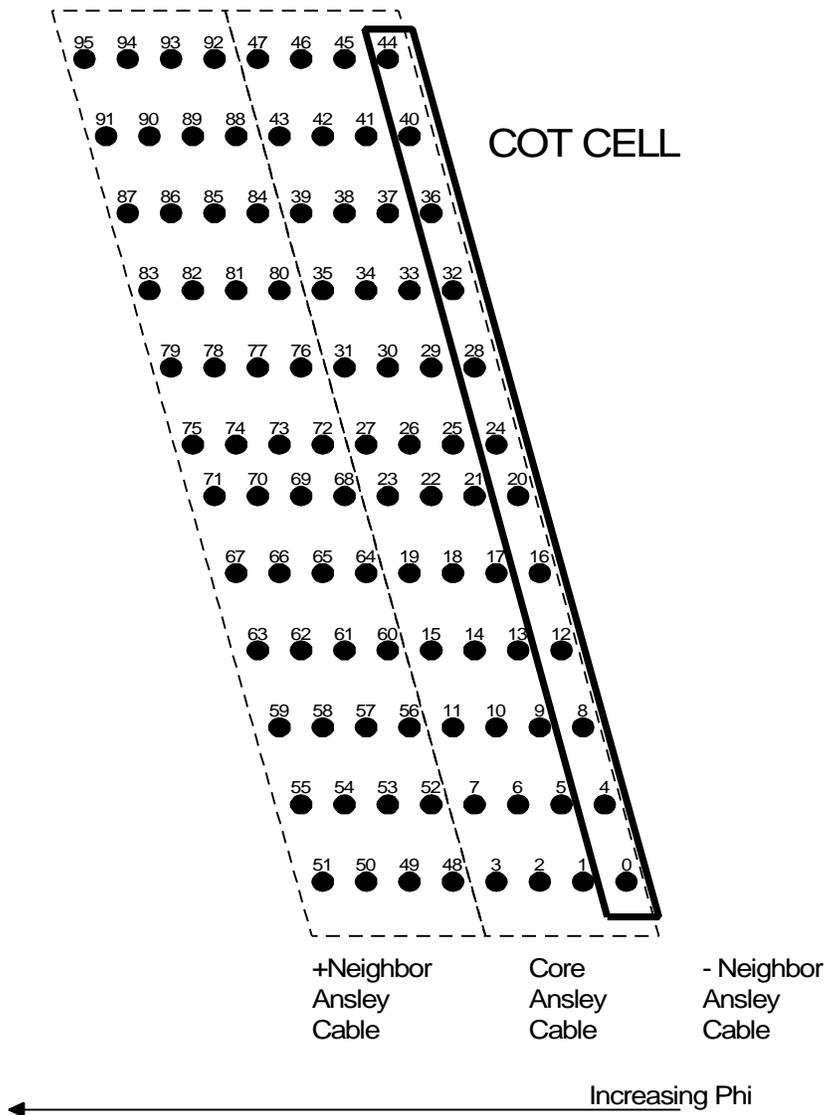
## IMPORTANT - Please Note

\*\*\*\*\*

The Wire number referred to in the below Tables represent a specific COT Wire geometry. **This is not necessarily the same as the WIRE\_DATA\_# referred to in the TDC document.** To understand that relationship, one must know how the TDC is being cabled at its input. This information was not known when we last inquired.

\*\*\*\*\*

The figure below represents our definition of 8 COT cells, Wires 0-95, with respect to the COT Wire geometry:



The TDC Transition card will drive two Ansley Cables. Information will be sent in 6 time slices within the 132 ns CDF\_CLOCK. An edge strobe will be sent with the data, as well as a Beam\_zero Marker (set high for all Beam\_zero events) and a Word\_zero Marker (set high in time slice 0).

Time Slice	Pairs 1,2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7	Pairs 8-23	Pairs 24,25
0	GND	STROBE	GND	Beam_zero	GND	logic high	Wires 0-15 Prompt	GND
1	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 16-31 Prompt	GND
2	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 32-47 Prompt	GND
3	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 0-15 Delay	GND
4	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 16-31 Delay	GND
5	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 32-47 Delay	GND

TDC Transition Module Cable 1 - Low Phi Channels (Wires) 0-47

Time Slice	Pairs 1,2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7	Pairs 8-23	Pairs 24,25
0	GND	STROBE	GND	Beam_zero	GND	logic high	Wires 48-63 Prompt	GND
1	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 64-79 Prompt	GND
2	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 80-95 Prompt	GND
3	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 48-63 Delay	GND
4	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 64-79 Delay	GND
5	GND	STROBE	GND	Beam_zero	GND	logic low	Wires 80-95 Delay	GND

TDC Transition Module Cable 2 - High Phi Channels (Wires) 48-95

To feed these cables, we require the following signals from the Mezzanine card:

Data (15:0) - represents multiplexed Wire data from Wire set (47:0)

Data (31:16) - represents multiplexed Wire data from Wire set (95:48)

Beam\_Zero

Word\_Zero

Strobe

All data and control signals must be in phase with minimal skewing.

TABLE TDC (J4/J5) Connector Pin Assignments

PIN	ROW A	ROW B	ROW C	ROW D	ROW E
1					Data (0)
2					Data (1)
3					Data (2)
4					Data (3)
5					Data (4)
6					Data (5)
7					Data (6)
8					Data (7)
9					Data (8)
10					Data (9)
11					Data (10)
12					Data (11)
13					Data (12)
14					Data (13)
15					Data (14)
16					Data (15)
17					
18					Word_Zero
19					Beam_Zero
20		NC	NC	NC	NC
21	NC	NC	NC	NC	NC
22	NC	NC	NC	NC	NC
23					Strobe
24					
25					Data (16)
26					Data (17)
27					Data (18)
28					Data (19)
29					Data (20)
30					Data (21)
31					Data (22)
32					Data (23)
33					Data (24)
34	Alignment Pins				
35					
36					
37					Data (25)
38					Data (26)
39					Data (27)
40					Data (28)
41					Data (29)
42					Data (30)
43					Data (31)
44					
45		NC	NC	NC	NC
46	NC	NC	NC	NC	NC
47	NC	NC	NC	NC	NC

TABLE Mezzanine Connector 2

Pin	Signal	Pin	Signal	Pin	Signal
1	<b>GND</b>	48	Data (7)	95	
2	<b>GND</b>	49		96	Data (13)
3	<b>GND</b>	50		97	
4	<b>GND</b>	51		98	
5		52		99	
6		53	Data (8)	100	
7		54		101	Data (14)
8		55		102	
9	Data (0)	56		103	
10		57	<b>GND</b>	104	
11		58	<b>GND</b>	105	
12		59	<b>SPARE</b>	106	Data (15)
13		60	<b>SPARE</b>	107	<b>GND</b>
14	Data (1)	61	<b>VCC</b>	108	<b>GND</b>
15		62	<b>VCC</b>	109	
16		63	<b>VCC</b>	110	
17		64	<b>VCC</b>	111	
18		65	<b>VCC</b>	112	
19	Data (2)	66	<b>VCC</b>	113	
20		67	<b>VCC</b>	114	
21	<b>GND</b>	68	<b>VCC</b>	115	
22	<b>GND</b>	69	<b>SPARE</b>	116	
23		70	<b>SPARE</b>	117	
24		71	<b>GND</b>	118	Word_Zero
25		72	<b>GND</b>	119	
26	Data (3)	73		120	
27		74	Data (9)	121	
28		75		122	
29		76		123	Beam_Zero
30		77		124	
31	Data (4)	78		125	<b>GND</b>
32		79	Data (10)	126	<b>GND</b>
33		80		127	<b>AS*</b>
34		81		128	<b>AK*</b>
35		82		129	<b>DS*</b>
36	Data (5)	83		130	<b>DK*</b>
37		84	Data (11)	131	<b>R/W*</b>
38		85		132	<b>INIT*</b>
39	<b>GND</b>	86		133	<b>PROG*</b>
40	<b>GND</b>	87		134	<b>SCLK*</b>
41		88		135	<b>SDATA</b>
42		89	<b>GND</b>	136	<b>DONE</b>
43	Data (6)	90	<b>GND</b>	137	<b>GND</b>
44		91	Data (12)	138	<b>GND</b>
45		92		139	<b>GND</b>
46		93		140	<b>GND</b>
47		94			

TABLE Mezzanine Connector 3

Pin	Signal	Pin	Signal	Pin	Signal
1	<b>GND</b>	48	Data (21)	95	
2	<b>GND</b>	49		96	Data (27)
3	<b>GND</b>	50		97	
4	<b>GND</b>	51		98	
5		52		99	
6		53	Data (22)	100	
7		54		101	Data (28)
8		55		102	
9	Strobe	56		103	
10		57	<b>GND</b>	104	
11		58	<b>GND</b>	105	
12		59	<b>SPARE</b>	106	Data (29)
13		60	<b>SPARE</b>	107	<b>GND</b>
14		61	<b>VCC</b>	108	<b>GND</b>
15		62	<b>VCC</b>	109	
16		63	<b>VCC</b>	110	
17		64	<b>VCC</b>	111	
18		65	<b>VCC</b>	112	
19	Data (16)	66	<b>VCC</b>	113	Data (30)
20		67	<b>VCC</b>	114	
21	<b>GND</b>	68	<b>VCC</b>	115	
22	<b>GND</b>	69	<b>SPARE</b>	116	
23		70	<b>SPARE</b>	117	
24		71	<b>GND</b>	118	Data (31)
25		72	<b>GND</b>	119	
26	Data (17)	73		120	
27		74	Data (23)	121	
28		75		122	
29		76		123	
30		77		124	
31	Data (18)	78		125	<b>GND</b>
32		79	Data (24)	126	<b>GND</b>
33		80		127	<b>AD0</b>
34		81		128	<b>AD1</b>
35		82		129	<b>AD2</b>
36	Data (19)	83		130	<b>AD3</b>
37		84	Data (25)	131	<b>AD4</b>
38		85		132	<b>AD5</b>
39	<b>GND</b>	86		133	<b>AD6</b>
40	<b>GND</b>	87		134	<b>AD7</b>
41		88		135	<b>NC</b>
42		89	<b>GND</b>	136	<b>NC</b>
43	Data (20)	90	<b>GND</b>	137	<b>GND</b>
44		91	Data (26)	138	<b>GND</b>
45		92		139	<b>GND</b>
46		93		140	<b>GND</b>
47		94			

**Appendix B**  
**Ansley Cable Specification**

### Ansley Cable Specification

Each piece is a flat cable 200.0 +/- 0.1 feet in length with 24 differential signal channels. There are three wires per channel consisting of a balanced pair of adjacent signal wires and a ground wire which provides isolation from the next channel. The basic cable consists of 75 or 76 wires (28 gauge solid copper) on a 0.33" +/- 0.005" pitch (wire spacing) between two sheets of 20 mil thick polyethylene dielectric making an approximately 50 mil thick sandwich. There are a minimum of two ground on each edge of the cable and the margin (center of edge wire to edge of the dielectric is 0.050" +/- 0.010").

The cable has a characteristic impedance of about 125 ohms. The rise-time is less than 7 nsec (10% to 50% pulse height) and the cross-talk is less than 3%. The cable delay channel to channel varies less than +/- 1 nsec.

Each end of the cables is terminated with a 50-pin backplane connector (two rows of sockets on 100 mil centers) as shown in the accompanying drawing. The backplane connector should be able to mate with a PCB right angle male header with retainer such as the Blue Max 609-5002MR. The first 48 sockets are for signals and the last two sockets are for ground. The construction is such that corresponding socket numbers are connected between the two ends. (If the cable is laid flat on a horizontal surface, then the top row of sockets on one end is connected to the bottom row of sockets on the other end!) The connection to the backplane connector is via a small circuit board and strain relief is provided.

## **Appendix C**

### **Finder Transition Module Specification**

Each of the SL1/SL3 Finder and SL2/SL4 Finder Module has its own transition module. The transition modules provide the means by which the Ansley Cables are connected to the XFT system.

These transition modules will be oversized. The standard area of the module will hold LVDS receivers and buffers. An extended area will hold the connectors that will mate with the Ansley cables coming from the COT TDC crates.

A SL1/SL3 Finder Transition Module must accept two (four with the COT upgrade which will split the wires) Superlayer 1 cables and four Superlayer 3 cables. Figure C-1 represents a functional drawing of the transition module.

A SL2/SL4 Finder Transition Module must accept three Superlayer 2 cables and five Superlayer 4 cables. Figure C-2 represents a functional drawing of the transition module.

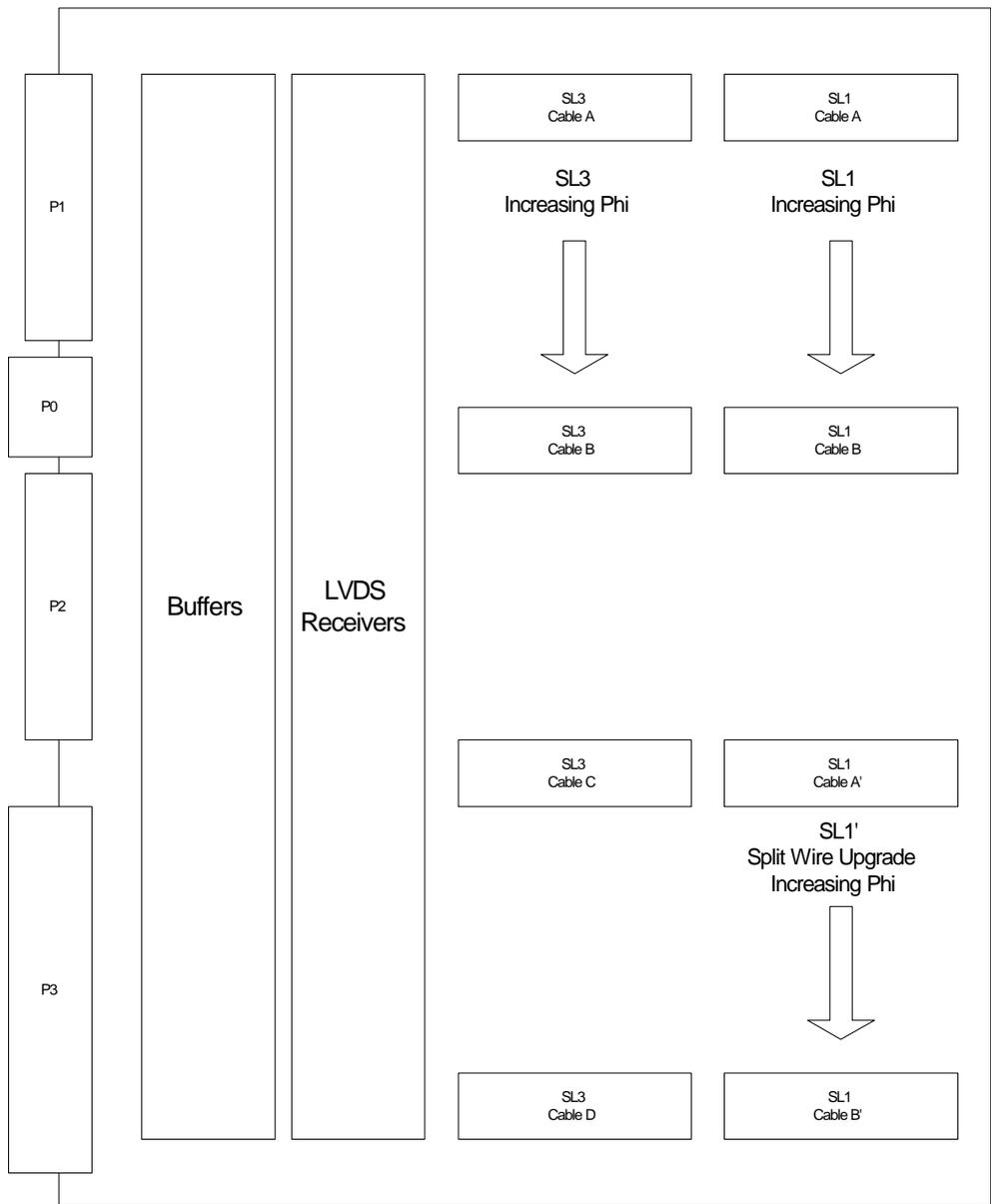


Figure C-1. Functional Layout of a Finder SL1/SL3 Transition Module

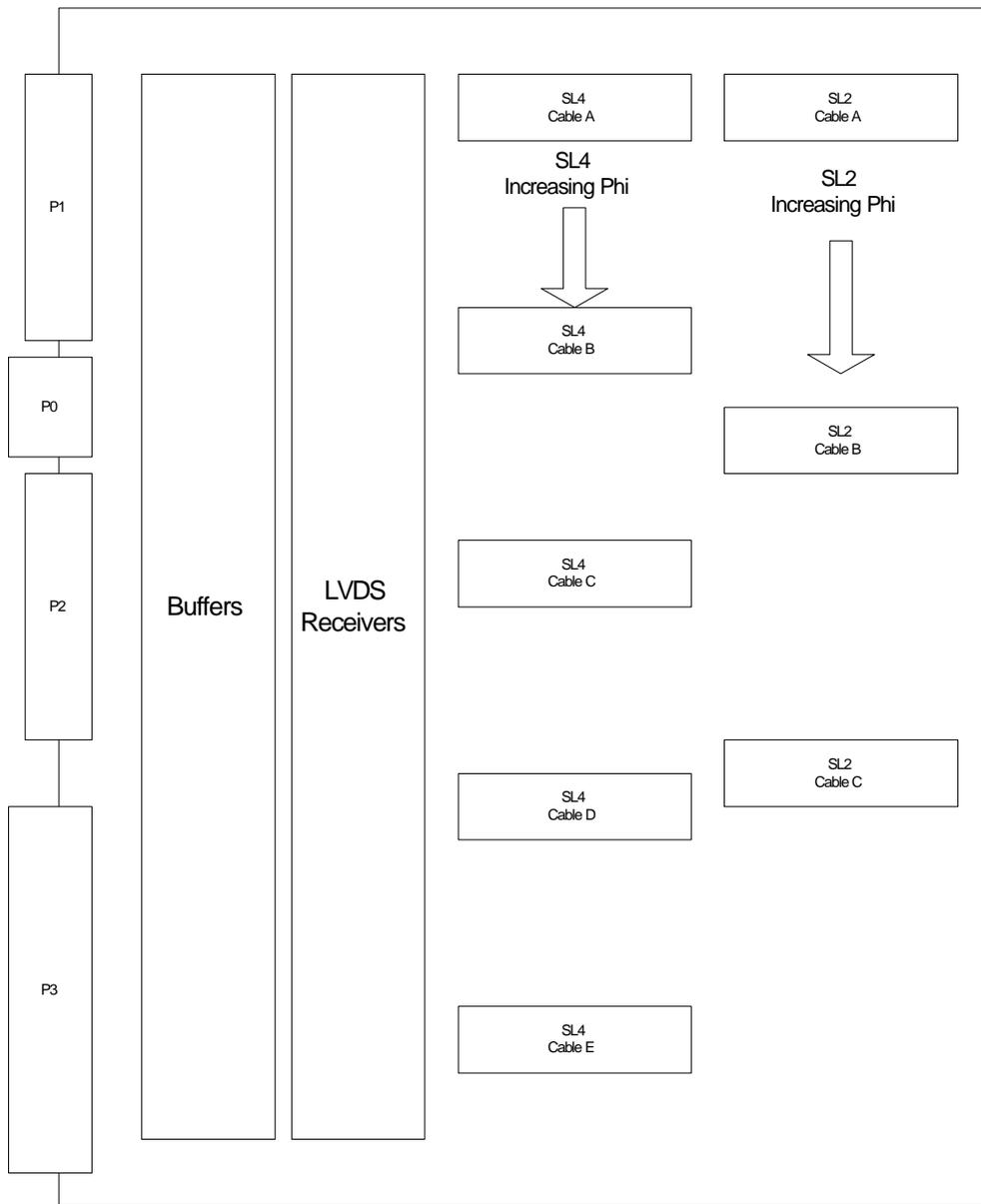


Figure C-2. Functional Layout of a Finder SL2/SL4 Transition Module

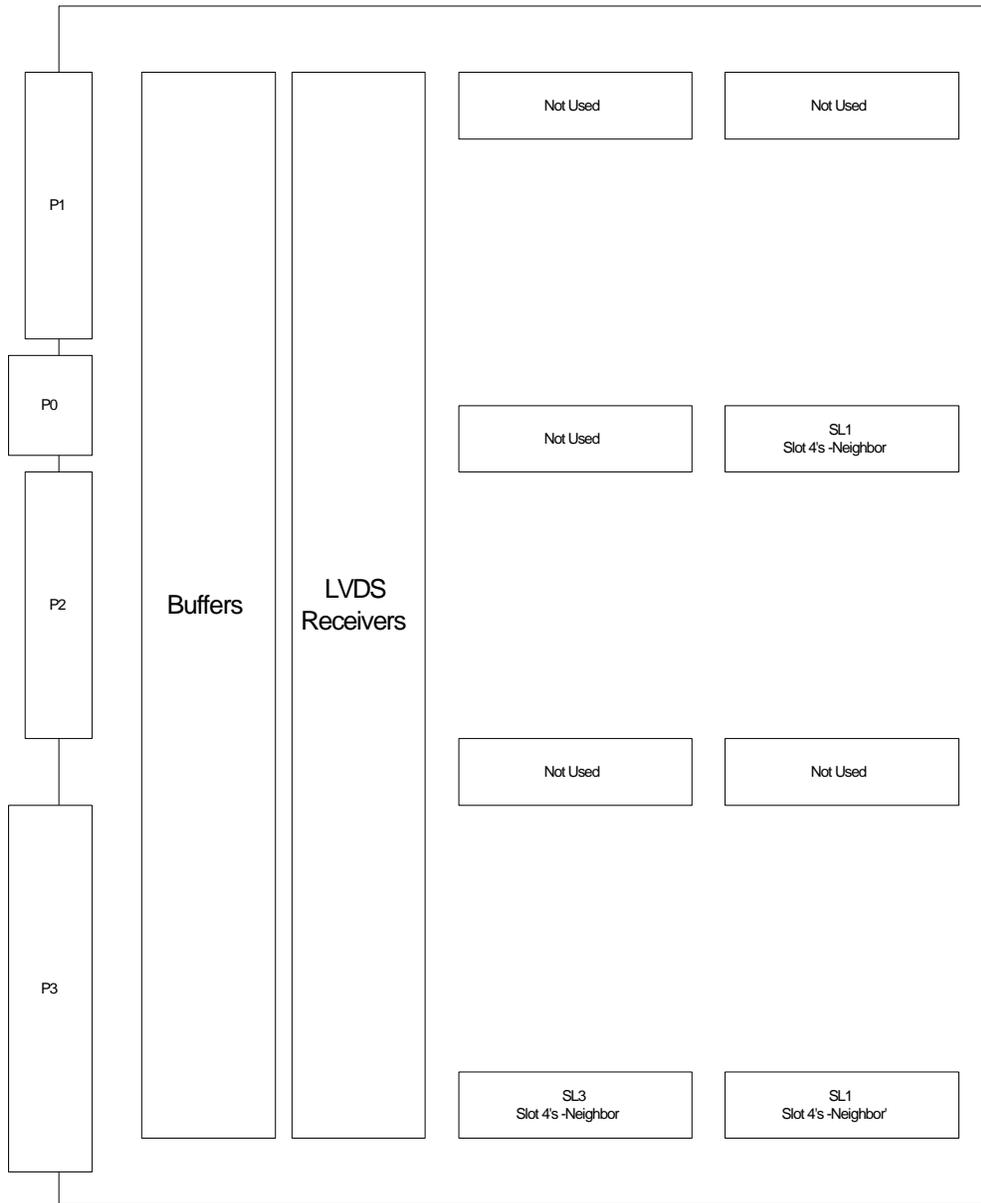


Figure C-3. SLOT 3 Cable Usage of a Finder SL1/SL3 Transition

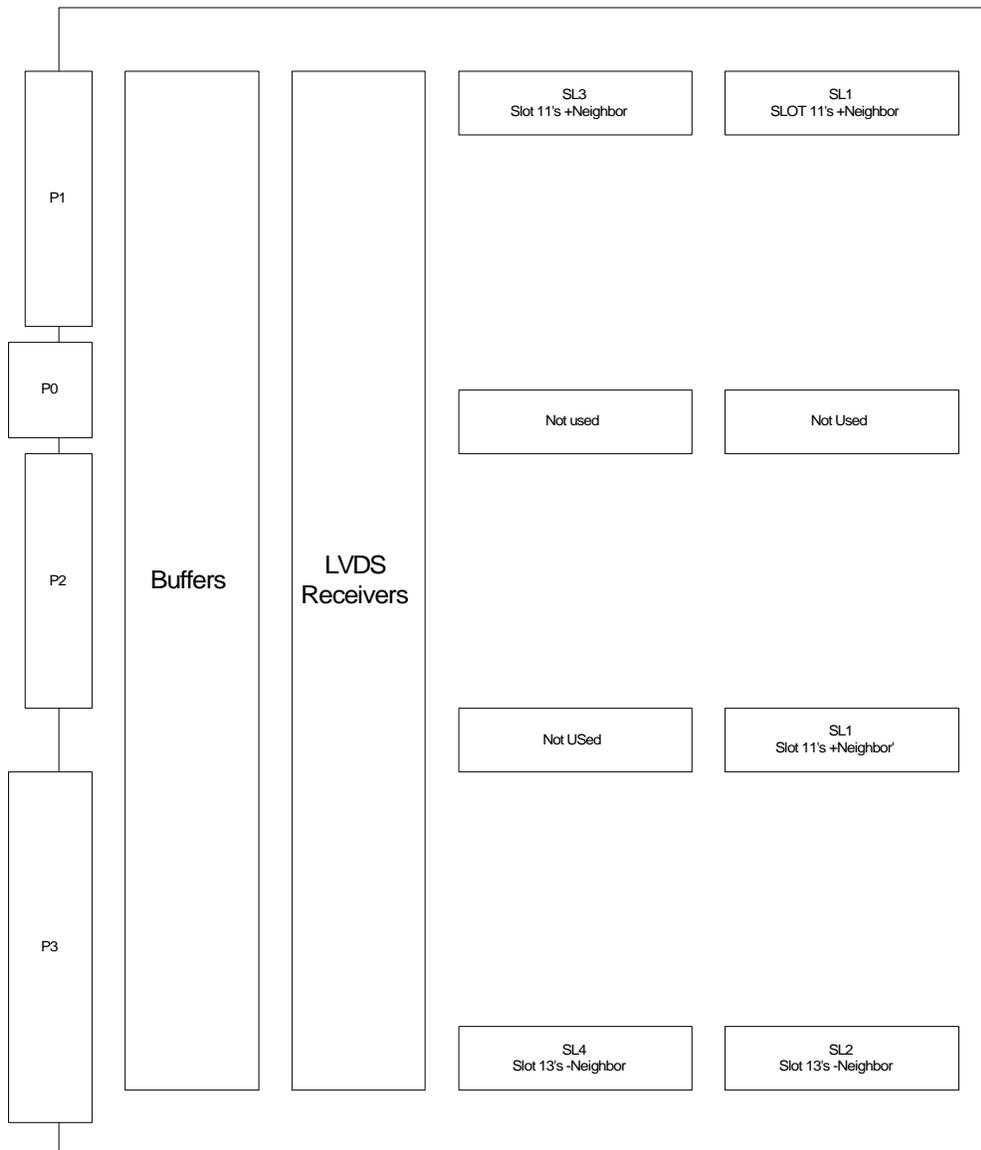


Figure C-4. SLOT 12 Cable Usage of a Finder SL1/SL3 Transition Module to handle Finder SL1/SL3 plus neighbors and Finder SL2/SL4 minus neighbors.

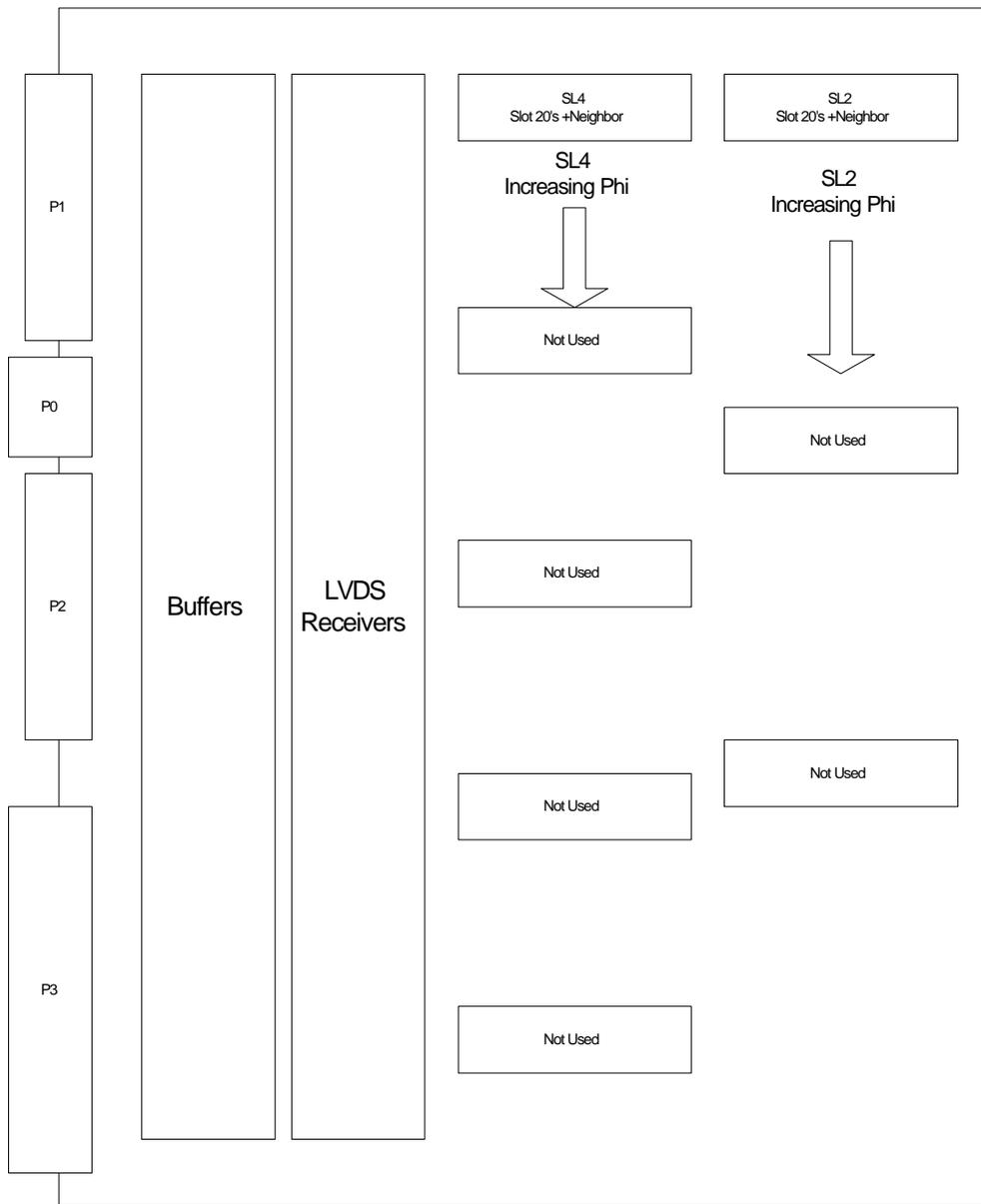


Figure C-5. SLOT 21 Cable Usage of a Finder SL2/SL4 Transition Module

Title: Plot of E:\Ppd\_jobs2\XFT\_FINDER1\_3\_TRAN\_V2\vbpcb\design.vbd  
Creator: MicroStation 95 5.5.3.54  
CreationDate: 12/02/97 13:59:53

Figure C-6. SL1/3 Transition Module PCB

Title: Plot of E:\Ppd\_jobs2\XFT\_FINDER2\_4\_TRAN\_V2\vbpcb\design.vbd  
Creator: MicroStation 95 5.5.3.54  
CreationDate: 12/02/97 14:06:48

Figure C-7. SL2/4 Transition Module PCB

**Appendix D**

**XFT Backplane and  
Custom J3 Specification**

## Finder Backplane

A custom J3 backplane will be required in order to bring TDC data into the Finder Module.

A SL1/SL3 Finder module requires connections to up to 8 (assumes 2 cables per SL1 Finder due to possible wire split) Ansley Cables as well as neighbor information from 6 additional cables. The neighbor information will be bussed along the custom J3 from adjacent slots. (-N represents the Neighbor in decreasing phi, +N represents the Neighbor in increasing phi)

The 8 cables plus Neighbors will be labeled as follows:

Neighbors from Cable SL1(-N) and SL1(-N')

Cable SL1(A)

Cable SL1(A')      A' is second cable from a possible split wire on SL1

Cable SL1(B)

Cable SL1(B')      B' is second cable from a possible split wire on SL1

Neighbors from Cable SL1(+N) and SL1(+N')

Neighbors from Cable SL3(-N)

Cable SL3(A)

Cable SL3(B)

Cable SL3(C)

Cable SL3(D)

Neighbors from Cable SL3(+N)

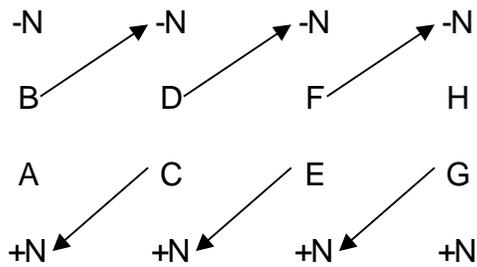


Illustration of Neighbor sharing between adjacent slots of the backplane.

A,B,C,... represent cable information in increasing phi

-N represents the neighbor at the lowest phi

+N represents the neighbor at the next highest phi

Illustration of Neighbor Sharing Scheme Across Adjacent Finder Slots

PIN #	Row Z	Row a	Row b	Row c	Row d	Row e	Row f
1	GND	+5V	+5V	+5V	+5V	+5V	GND
2	GND	RET_WX	Reserved	+5V	TBUS 1+	TBUS 1-	GND
3	GND	RET_WX	Reserved	Reserved	TBUS 2+	TBUS 2-	GND
4	GND	Vw	Reserved	SL3C-W0	SL3C-STRB	SL3C-B0	GND
5	GND	Vw	Reserved	USR I/O	SL3C-D(0)	SL3C-D(1)	GND
6	GND	RET_WX	Reserved	SL3B-STRB	SL3C-D(2)	SL3C-D(3)	GND
7	GND	AREF_WX	Reserved	SL3B-B0	SL3C-D(4)	SL3C-D(5)	GND
8	GND	RET_WX	Reserved	SL3B-W0	SL3C-D(6)	SL3C-D(7)	GND
9	GND	Vx	Reserved	SL3B-D(0)	SL3C-D(8)	SL3C-D(9)	GND
10	GND	Vx	Reserved	SL3B-D(1)	SL3C-D(10)	SL3C-D(11)	GND
11	GND	Vy	Reserved	SL3B-D(2)	SL3C-D(12)	SL3C-D(13)	GND
12	GND	Vy	Reserved	SL3B-D(3)	SL3C-D(14)	SL3C-D(15)	GND
13	GND	RET_YZ	Reserved	SL3B-D(4)	SL3B-D(8)	SL3B-D(9)	GND
14	GND	AREF_YZ	Reserved	SL3B-D(5)	SL3B-D(10)	SL3B-D(11)	GND
15	GND	RET_YZ	Reserved	SL3B-D(6)	SL3B-D(12)	SL3B-D(13)	GND
16	GND	Vz	Reserved	SL3B-D(7)	SL3B-D(14)	SL3B-D(15)	GND
17	GND	Vz	Reserved	Reserved	TBUS 3+	TBUS 3-	GND
18	GND	RET_YZ	Reserved	Reserved	TBUS 4+	TBUS 4-	GND
19	GND	RET_YZ	Reserved	Reserved	TBUS OC1	TBUS OC2	GND

Table 1. Finder SL1/SL3 J0 Connector

PIN #	Row Z	Row a	Row b	Row c	Row d	Row e	Row f
1	GND	SL3A-STB	SL3A-B0	GND	SL3+N-STB	SL3+N-B0	GND
2	GND	SL3A-W0	GND	GND	SL3+N-W0	SL3+N-D(0)	GND
3	GND	SL3A-D(0)	SL3A-D(1)	SL3+N-D(1)	SL3+N-D(2)	SL3+N-D(4)	GND
4	GND	SL3A-D(2)	SL3A-D(3)	SL3+N-D(5)	SL3+N-D(6)	SL3+N-D(8)	GND
5	GND	SL3A-D(4)	SL3A-D(5)	GND	SL3+N-D(9)	SL3+N-D(12)	GND
6	GND	SL3A-D(6)	SL3A-D(7)	GND	SL3D-STRB	SL3D-B0	GND
7	GND	SL3A-D(8)	SL3A-D(9)	SL3D-W0	SL3D-D(0)	SL3D-D(1)	GND
8	GND	SL3A-D(10)	SL3A-D(11)	GND	SL3D-D(2)	SL3D-D(3)	GND
9	GND	SL3A-D(12)	SL3A-D(13)	GND	SL3D-D(4)	SL3D-D(5)	GND
10	GND	SL3A-D(14)	SL3A-D(15)	GND	SL3D-D(6)	SL3D-D(7)	GND
11	GND	SL3-N-STB	SL3-N-B0	GND	SL3D-D(8)	SL3D-D(9)	GND
12	GND	SL3-N-W0	SL3-N-D(3)	GND	SL3D-D(10)	SL3D-D(11)	GND
13	GND	SL3-N-D(6)	SL3-N-D(7)	SL3-N-D(9)	SL3D-D(12)	SL3D-D(13)	GND
14	GND	SL3-N-D(10)	SL3-N-D(11)	SL3-N-D(13)	SL3D-D(14)	SL3D-D(15)	GND
15	GND	SL3-N-D(14)	SL3-N-D(15)	GND	SL1B-STB	SL1B-B0	GND
16	GND	SL1-N-STB	SL1-N-B0	GND	GND	SL1B-W0	GND
17	GND	SL1-N-W0	SL1-N-D(3)	GND	SL1B-D(0)	SL1B-D(1)	GND
18	GND	SL1-N-D(6)	SL1-N-D(7)	SL1-N-D(9)	SL1B-D(2)	SL1B-D(3)	GND
19	GND	SL1-N-D(10)	SL1-N-D(11)	SL1-N-D(13)	SL1B-D(4)	SL1B-D(5)	GND
20	GND	SL1-N-D(14)	SL1-N-D(15)	GND	SL1B-D(6)	SL1B-D(7)	GND
21	GND	SL1A-STB	SL1A-B0	GND	SL1B-D(8)	SL1B-D(9)	GND
22	GND	SL1A-W0	GND	GND	SL1B-D(10)	SL1B-D(11)	GND

PIN #	Row Z	Row a	Row b	Row c	Row d	Row e	Row f
23	GND	SL1A-D(0)	SL1A-D(1)	GND	SL1B-D(12)	SL1B-D(13)	GND
24	GND	SL1A-D(2)	SL1A-D(3)	GND	SL1B-D(14)	SL1B-D(15)	GND
25	GND	SL1A-D(4)	SL1A-D(5)	GND	SL1+N-STB	SL1+N-B0	GND
26	GND	SL1A-D(6)	SL1A-D(7)	GND	SL1+N-W0	SL1+N-D(0)	GND
27	GND	SL1A-D(8)	SL1A-D(9)	SL1+N-D(1)	SL1+N-D(2)	SL1+N-D(4)	GND
28	GND	SL1A-D(10)	SL1A-D(11)	SL1+N-D(5)	SL1+N-D(6)	SL1+N-D(8)	GND
29	GND	SL1A-D(12)	SL1A-D(13)	GND	SL1+N-D(9)	SL1+N-D(12)	GND
30	GND	SL1A-D(14)	SL1A-D(15)	GND	SL1+N'-STB	SL1+N'-B0	GND
31	GND	SL1A'-STB	SL1A'-B0	SL1A'-W0	SL1+N'-W0	SL1+N'-D(0)	GND
32	GND	SL1A'-D(0)	SL1A'-D(1)	SL1+N'-D(1)	SL1+N'-D(2)	SL1+N'-D(4)	GND
33	GND	SL1A'-D(2)	SL1A'-D(3)	SL1+N'-D(5)	SL1+N'-D(6)	SL1+N'-D(8)	GND
34							
35							
36							
37	GND	SL1A'-D(4)	SL1A'-D(5)	GND	SL1+N'-D(9)	SL1+N'-D(12)	GND
38	GND	SL1A'-D(6)	SL1A'-D(7)	GND	SL1B'-STB	SL1B'-B0	GND
39	GND	SL1A'-D(8)	SL1A'-D(9)	GND	GND	SL1B'-W0	GND
40	GND	SL1A'-D(10)	SL1A'-D(11)	GND	SL1B'-D(0)	SL1B'-D(1)	GND
41	GND	SL1A'-D(12)	SL1A'-D(13)	GND	SL1B'-D(2)	SL1B'-D(3)	GND
42	GND	SL1A'-D(14)	SL1A'-D(15)	GND	SL1B'-D(4)	SL1B'-D(5)	GND
43	GND	SL1-N'-STB	SL1-N'-B0	GND	SL1B'-D(6)	SL1B'-D(7)	GND
44	GND	SL1-N'-W0	SL1-N'-D(3)	GND	SL1B'-D(8)	SL1B'-D(9)	GND
45	GND	SL1-N'-D(6)	SL1-N'-D(7)	SL1-N'-D(9)	SL1B'-D(10)	SL1B'-D(11)	GND
46	GND	SL1-N'-D(10)	SL1-N'-D(11)	SL1-N'-D(13)	SL1B'-D(12)	SL1B'-D(13)	GND
47	GND	SL1-N'-D(14)	SL1-N'-D(15)	GND	SL1B'-D(14)	SL1B'-D(15)	GND

Table 2. CUSTOM J3 - Slot definition for SL1/SL3 Finder

A SL2/SL4 Finder module requires connections to 8 Ansley Cables as well as neighbor information from 4 additional cables. The neighbor information will be bussed along the custom J3 from adjacent slots.

The 8 cables plus Neighbors will be labeled as follows:

Neighbors from Cable SL2(-N)  
Cable SL2(A)  
Cable SL2(B)  
Cable SL2(C)  
Neighbors from Cable SL2(+N)

Neighbors from Cable SL4(-N)  
Cable SL4(A)  
Cable SL4(B)  
Cable SL4(C)  
Cable SL4(D)  
Cable SL4(E)  
Neighbors from Cable SL4(+N)

PIN #	Row Z	Row a	Row b	Row c	Row d	Row e	Row f
1	GND	+5V	+5V	+5V	+5V	+5V	GND
2	GND	RET_WX	Reserved	+5V	TBUS 1+	TBUS 1-	GND
3	GND	RET_WX	Reserved	Reserved	TBUS 2+	TBUS 2-	GND
4	GND	Vw	Reserved	SL4B-W0	SL4B-STRB	SL4B-B0	GND
5	GND	Vw	Reserved	USR I/O	SL4B-D(0)	SL4B-D(1)	GND
6	GND	RET_WX	Reserved	SL4C-STRB	SL4B-D(2)	SL4B-D(3)	GND
7	GND	AREF_WX	Reserved	SL4C-B0	SL4B-D(4)	SL4B-D(5)	GND
8	GND	RET_WX	Reserved	SL4C-W0	SL4B-D(6)	SL4B-D(7)	GND
9	GND	Vx	Reserved	SL4C-D(0)	SL4B-D(8)	SL4B-D(9)	GND
10	GND	Vx	Reserved	SL4C-D(1)	SL4B-D(10)	SL4B-D(11)	GND
11	GND	Vy	Reserved	SL4C-D(2)	SL4B-D(12)	SL4B-D(13)	GND
12	GND	Vy	Reserved	SL4C-D(3)	SL4B-D(14)	SL4B-D(15)	GND
13	GND	RET_YZ	Reserved	SL4C-D(4)	SL4C-D(8)	SL4C-D(9)	GND
14	GND	AREF_YZ	Reserved	SL4C-D(5)	SL4C-D(10)	SL4C-D(11)	GND
15	GND	RET_YZ	Reserved	SL4C-D(6)	SL4C-D(12)	SL4C-D(13)	GND
16	GND	Vz	Reserved	SL4C-D(7)	SL4C-D(14)	SL4C-D(15)	GND
17	GND	Vz	Reserved	Reserved	TBUS 3+	TBUS 3-	GND
18	GND	RET_YZ	Reserved	Reserved	TBUS 4+	TBUS 4-	GND
19	GND	RET_YZ	Reserved	Reserved	TBUS OC1	TBUS OC2	GND

Table 3. Finder SL2/SL4 J0 Connector

PIN #	Row Z	Row a	Row b	Row c	Row d	Row e	Row f
1	GND	SL4A-STB	SL4A-B0	GND	SL4+N-STB	SL4+N-B0	GND
2	GND	SL4A-W0	GND	GND	SL4+N-W0	GND	GND
3	GND	SL4A-D(0)	SL4A-D(1)	GND	SL4+N-D(0)	SL4+N-D(1)	GND
4	GND	SL4A-D(2)	SL4A-D(3)	GND	SL4+N-D(2)	SL4+N-D(4)	GND
5	GND	SL4A-D(4)	SL4A-D(5)	GND	SL4+N-D(5)	GND	GND
6	GND	SL4A-D(6)	SL4A-D(7)	GND	SL4+N-D(6)	SL4+N-D(8)	GND
7	GND	SL4A-D(8)	SL4A-D(9)	GND	SL4+N-D(9)	SL4+N-D(12)	GND
8	GND	SL4A-D(10)	SL4A-D(11)	GND	SL4E-STB	SL4E-B0	GND
9	GND	SL4A-D(12)	SL4A-D(13)	GND	GND	SL4E-W0	GND
10	GND	SL4A-D(14)	SL4A-D(15)	GND	SL4E-D(0)	SL4E-D(1)	GND
11	GND	SL4-N-STB	SL4-N-B0	GND	SL4E-D(2)	SL4E-D(3)	GND
12	GND	SL4-N-W0	SL4-N-D(3)	GND	SL4E-D(4)	SL4E-D(5)	GND
13	GND	SL4-N-D(6)	SL4-N-D(7)	GND	SL4E-D(6)	SL4E-D(7)	GND
14	GND	GND	SL4-N-D(9)	GND	SL4E-D(8)	SL4E-D(9)	GND
15	GND	SL4-N-D(10)	SL4-N-D(11)	GND	SL4E-D(10)	SL4E-D(11)	GND
16	GND	GND	SL4-N-D(13)	GND	SL4E-D(12)	SL4E-D(13)	GND
17	GND	SL4-N-D(14)	SL4-N-D(15)	GND	SL4E-D(14)	SL4E-D(15)	GND
18	GND	SL4D-STRB	SL4D-B0	GND	SL2B-STB	SL2B-B0	GND
19	GND	SL4A-W0	GND	GND	GND	SL2B-W0	GND
20	GND	SL4D-D(0)	SL4D-D(1)	GND	SL2B-D(0)	SL2B-D(1)	GND
21	GND	SL4D-D(2)	SL4D-D(3)	GND	SL2B-D(2)	SL2B-D(3)	GND
22	GND	SL4D-D(4)	SL4D-D(5)	GND	SL2B-D(4)	SL2B-D(5)	GND

PIN #	Row Z	Row a	Row b	Row c	Row d	Row e	Row f
23	GND	SL4D-D(6)	SL4D-D(7)	GND	SL2B-D(6)	SL2B-D(7)	GND
24	GND	SL4D-D(8)	SL4D-D(9)	GND	SL2B-D(8)	SL2B-D(9)	GND
25	GND	SL4D-D(10)	SL4D-D(11)	GND	SL2B-D(10)	SL2B-D(11)	GND
26	GND	SL4D-D(12)	SL4D-D(13)	GND	SL2B-D(12)	SL2B-D(13)	GND
27	GND	SL4D-D(14)	SL4D-D(15)	GND	SL2B-D(14)	SL2B-D(15)	GND
28	GND	SL2A-STB	SL2A-B0	GND	SL2+N-STB	SL2+N-B0	GND
29	GND	SL2A-W0	GND	GND	SL2+N-W0	GND	GND
30	GND	SL2A-D(0)	SL2A-D(1)	GND	SL2+N-D(0)	SL2+N-D(1)	GND
31	GND	SL2A-D(2)	SL2A-D(3)	GND	SL2+N-D(2)	SL2+N-D(4)	GND
32	GND	SL2A-D(4)	SL2A-D(5)	GND	SL2+N-D(5)	GND	GND
33	GND	SL2A-D(6)	SL2A-D(7)	GND	SL2+N-D(6)	SL2+N-D(8)	GND
34							
35							
36							
37	GND	SL2A-D(8)	SL2A-D(9)	GND	SL2+N-D(9)	SL2+N-D(12)	GND
38	GND	SL2A-D(10)	SL2A-D(11)	GND	SL2C-STB	SL2C-B0	GND
39	GND	SL2A-D(12)	SL2A-D(13)	GND	GND	SL2C-W0	GND
40	GND	SL2A-D(14)	SL2A-D(15)	GND	SL2C-D(0)	SL2C-D(1)	GND
41	GND	SL2-N-STB	SL2-N-B0	GND	SL2C-D(2)	SL2C-D(3)	GND
42	GND	SL2-N-W0	SL2-N-D(3)	GND	SL2C-D(4)	SL2C-D(5)	GND
43	GND	SL2-N-D(6)	SL2-N-D(7)	GND	SL2C-D(6)	SL2C-D(7)	GND
44	GND	GND	SL2-N-D(9)	GND	SL2C-D(8)	SL2C-D(9)	GND
45	GND	SL2-N-D(10)	SL2-N-D(11)	GND	SL2C-D(10)	SL2C-D(11)	GND
46	GND	GND	SL2-N-D(13)	GND	SL2C-D(12)	SL2C-D(13)	GND
47	GND	SL2-N-D(14)	SL2-N-D(15)	GND	SL2C-D(14)	SL2C-D(15)	GND

Table 4. CUSTOM J3 - Slot definition for SL2/SL4 Finder

Title: Plot of E:\Ppd\_jobs2\XFT\_BACKPLANE\_V3\vbpcb\design.vbd  
Creator: MicroStation 95 5.5.3.54  
CreationDate: 12/02/97 14:48:32

**Appendix E**  
**Finder SL1/3 PCB Layout**

Title: Plot of D:\PPD\_JOBS\XFT\_FINDER\_SL13\VBPCB\design.vbo  
Creator: MicroStation 95 5.5.3.54  
CreationDate: 12/02/97 15:00:25

## **Appendix F**

### **Finder SL2/4 PCB Layout**

Title: Plot of J:\XFT\_FINDER\_SL24\_V1\vbpcb\design.vbd  
Creator: MicroStation 95 5.5.3.54  
CreationDate: 06/18/98 16:21:44

