REQUIREMENTS AND FUNCTIONAL SPECIFICATION

Station Board Wideband Correlator FPGA

RFS Document: **A25051N0000**

Revision: 1.8

Dave Fort, 01 April 2011

National Research Council Canada
Herzberg Institute of Astrophysics
Dominion Radio Astrophysical Observatory

P.O. Box 248, 717 White Lake Rd
Penticton, B.C., Canada
V2A 6J9

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

CMIB	Correlator Module Interface Board
DLL	Delay Locked Loop
EVLA	Expanded Very Large Array
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
ISR	Interrupt Service Routine
PCB	Printed Circuit Board
PLL	Phase Locked Loop
SNR	Signal-to-Noise Ratio
VLBA	Very Long Baseline Array
VLBI	Very Long Baseline Interferometry
WBC	Wide Band Correlator
WIDAR	Wideband Interferometric Digital ARchitecture

1 Revision History

Revision	Date	Changes/Notes	Author
0.0	01 Nov 2003	Initial Draft	D. Fort
0.1	01 Dec 2003	Lag range reduced	D. Fort
1.0	01 Jun 2005	Updated, Lag range expanded	D. Fort
1.1	15 Aug 2005	Small Changes, simplified TIC, stick	D. Fort
1.2	25 Jan 2007	Input delays replaced by DDR clocking.	D. Fort
1.3	25 Apr 2007	More test abilities. Removed PROTECT.	D. Fort
1.4	25 Mar 2008	More test abilities. Separate input and output CRCs for testing.	D. Fort
1.5	25 May 2008	More test abilities. Allow A-B alignment.	D. Fort
1.6	04 Jun 2008	Small change to wideband test generator	D. Fort
1.7	04 Feb 2010	Added DCM reset bit. Added DCM phase shifting for edge testing.	D. Fort
1.8	01Apr 2011	Add pinouts, etc	D. Fort

2 Introduction

This document describes detailed requirements and design concepts for the wideband correlator FPGA. This device is usually referred to as the WBC FPGA but it can perform both auto-correlation and cross-correlation on the two wideband inputs.

Background on the WBC FPGA can be found in NRC-EVLA Memo #014 (EVLA Memo 31), "Refined EVLA WIDAR Correlator Architecture".

The WBC FPGA will be an FPGA (Xilinx Virtex-4 SX35). The design will be done in Verilog-2001 HDL.

3 Overview

The Station Board receives two wide band inputs that may be right and left circularly polarized components from the same radio source. The primary function of the WBC FPGA is to measure the spectra of each of the wide band inputs and possibly the corresponding cross-polarized spectra. The resulting spectra will be used for diagnostic purposes and may be used as a tool for Radio Frequency Interference mitigation. The design outlined in this document does not actually produce a spectrum directly but forms the cross lag function that can be turned into the spectrum by software using a Fourier Transform. There is one wide band WBC FPGA on the Station Board. A simplified block diagram of the Station Board is shown in Figure 3-1.

The input of the WBC FPGA comes from the two Delay Modules. The output is a re-timed copy of the input and goes on to the two filter banks. The WBC FPGA does not modify this data but uses it to accumulate lag domain cross-correlation functions. The results of the correlations are available to the CMIB every interrupt. The reason for passing the data through the FPGA is to simplify routing and improve signal integrity by avoiding multi-drop transmission lines at the expense of somewhat increased power dissipation.

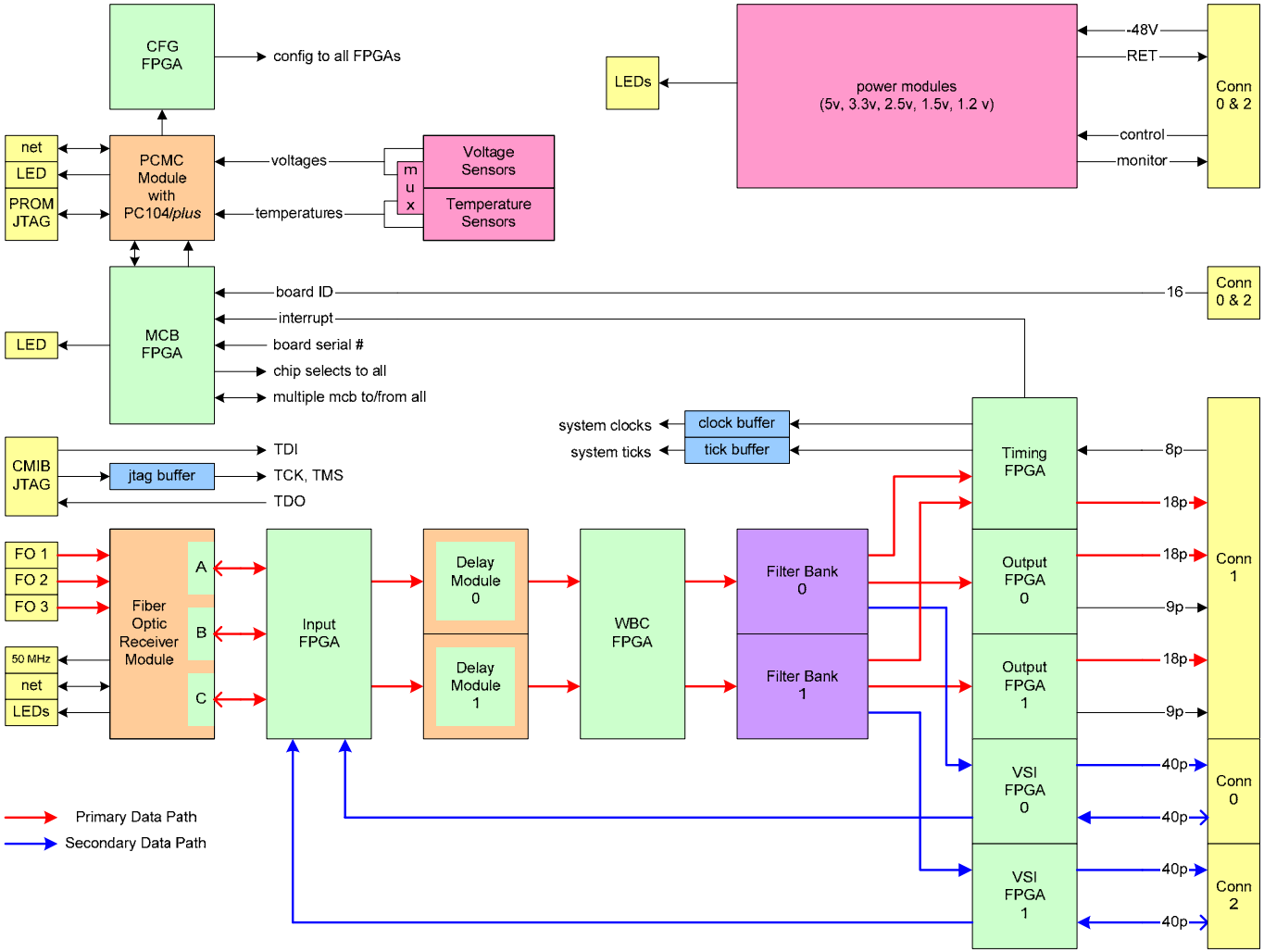

Figure 3-1 Block diagram of the Station Board.

A simplified input/output diagram of the WBC FPGA is shown below. The JTAG interface and the FPGA configuration interface are not shown.

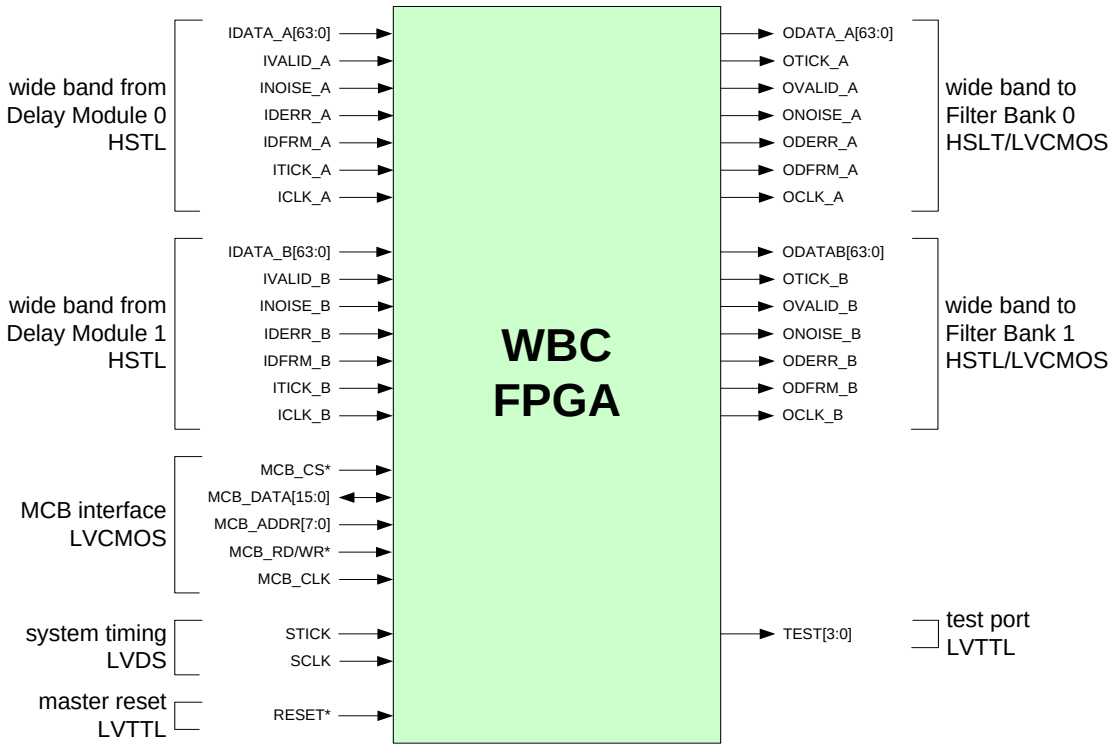

Figure 3-2 Input/Output diagram of the WBC FPGA.

The two wide band data inputs each consists of 64 data lines as well as valid, noise diode on/off, delay error, delay frame, tick and clock lines. The data, valid, noise diode and tick lines are running at 256 Mbits/second. The delay error and delay frame lines are running at 128 Mbits/second and the clock is 128 MHz. In addition, there are system tick and system clock lines coming to each FPGA.

Number of Bands	Width of Band (MHz)	Demux Factor	Bits per Sample	Mode ID
1	2048	16	4	4_16
2	1024	8	4	4_8
4	512	4	4	4_4
8	256	2	4	4_2
16	$128/2^n$ n=0,...12	1	4	4_1
1	1024	8	8	8_8
2	512	4	8	8_4
4	256	2	8	8_2
8	$128/2^n$ n=0,...12	1	8	8_1

Table 3-1 Wide band input data organizations.

Modes 4_16 and 8_8 are the standard EVLA data organizations. Mode 8_4 is for the 8-bit e-MERLIN case. Mode 4-1 could be used by the current VLBA and represents cases where the bandwidth is 128 MHz or less. The others are included for completeness. The Station Board FPGAs should deal with all of the above organizations, if possible. In order that the Delay Module, the WBC FPGA and the Filter FPGA be able to implement the above organizations, it is necessary that all the bits that are sampled at the same time are adjacent in the 64-bit data word. This allows the bits and bands to be treated as a single (wide) sample and be delayed together. Note that the Station Board is capable of dealing with 4-bit or 8-bit data but the EVLA only uses a 3-bit sampler.

A number of architectures are possible for this FPGA including four FFTs that produce the desired spectra directly. The chosen architecture is a lag correlator that accumulates a fixed number of lags starting at a variable lag offset for one of the four products for a period of one interrupt. The CMIB is responsible for accumulating the requested lag range and total integration time in an internal buffer. It is likely that the accumulated buffer would be sent to the EVLA Monitor and Control System where the transformation to the frequency domain would be done. A simplified block diagram of the WBC FPGA is shown below.

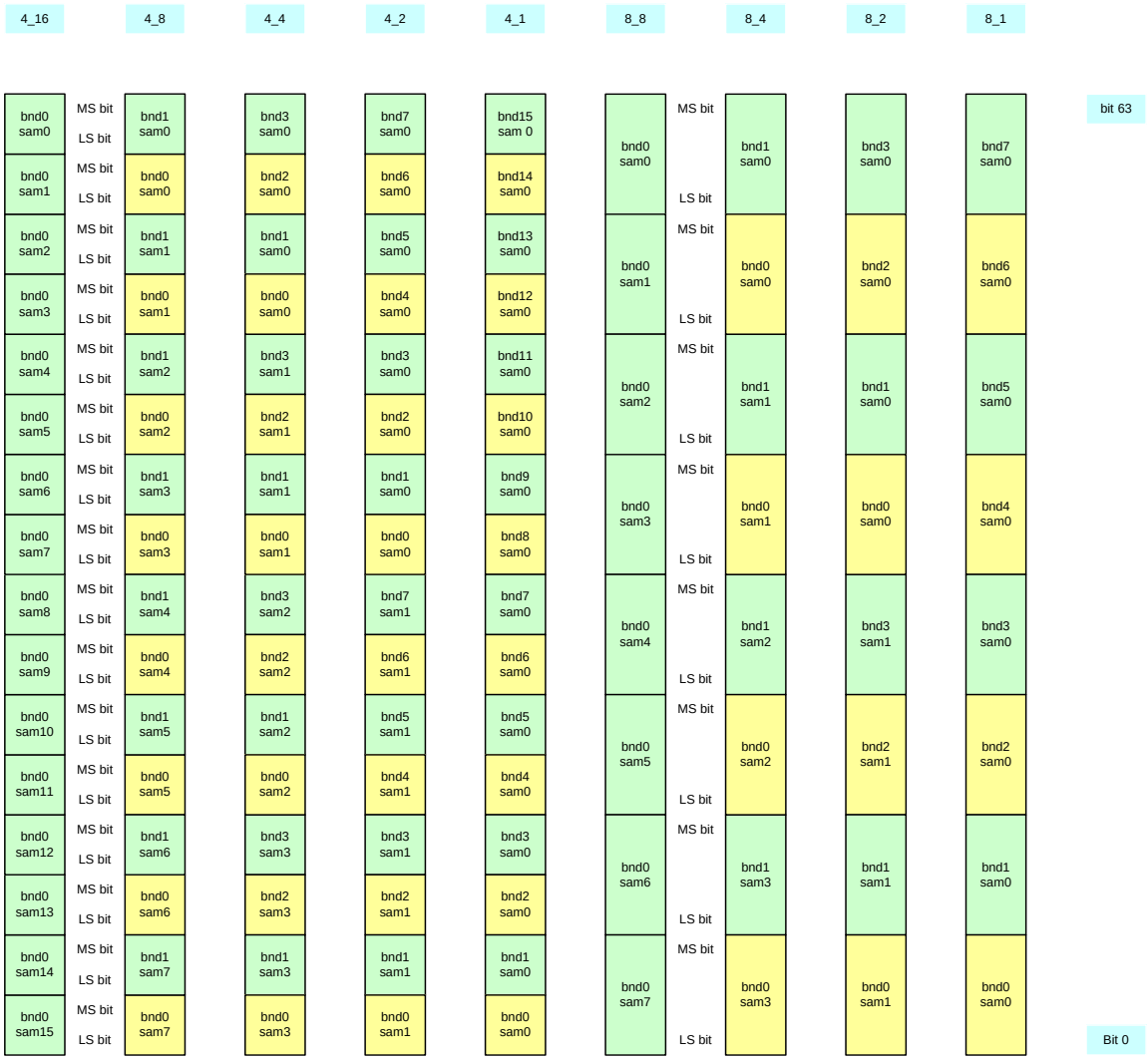

Figure 3-3 Wideband Input Data Organization

A simplified block diagram of the WBC FPGA is shown below.

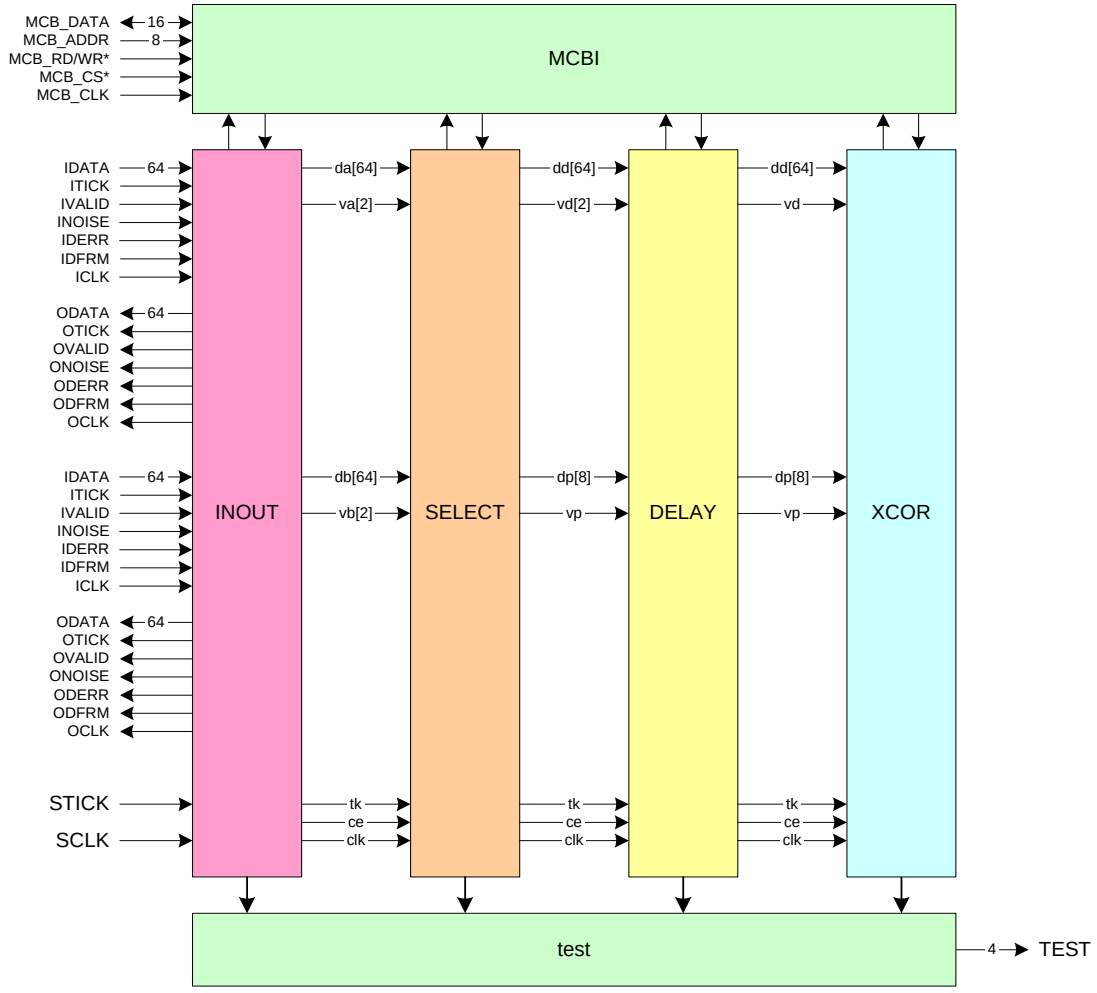

Figure 3-4 Block diagram of the WBC FPGA

The WBC FPGA is divided into six modules that are expanded in the sections below. MCBI is the monitor and control bus interface that contains the register set used to configure and control the other blocks as well as received status and monitor information. INOUT receives wide band data from the two Delay Modules and re-transmits it to the first FPGA in each of the two base band filter banks as well as providing signals for use within the FPGA. SELECT chooses which of the two wide band data inputs become the "lagged" data and the "prompt" data for the lag correlator and rearranges the data to allow all the data organizations in Table 3-1. DELAY implements the gross lag delay that will be stepped through the desired lag range by the CMIB to form the complete correlation function. XCOR consists of 64 identical multiplier/accumulators that are fed by the prompt data on one side and 64 consecutive data samples on the other.

4 Requirements

The following is a list of WBC FPGA requirements.

4.1 Functional Requirements

1. The WBC FPGA must allow the production of spectral displays for the four possible pairings of the two wide band inputs with a resolution of up to 1024 spectral bins across the input band. It is not necessary that the four spectra be derived from data taken simultaneously or that each accumulation uses all the data points. Throwing away input samples lowers the signal-to-noise ratio of the result so this should be kept to a minimum. In this case, the SNR should not be degraded by more that a factor of 4.

2. The WBC FPGA must be able to cope with data organizations modes 4_16 (EVLA), 4_1 (VLBA), 8_8 (EVLA) and 8_4 (e-MERLIN) as listed in Table 3-1. If possible, allow all modes.

4.2 Performance Requirements

1. The WBC FPGA shall operate on the input data and associated signals sampled at 256 MHz. The FPGA will take a 128 MHz input clock and perform double-edged sampling of the data.

2. The synchronous MCB interface shall be capable of operating with a clock that is neither frequency nor phase synchronous with the 128 MHz clock. The FPGA will support an MCB interface clock with a maximum rate of 33 MHz.

3. The power dissipation of the FPGA shall not exceed 7 Watts; however, the goal for the maximum power dissipation in 5 Watts.

4.3 Environmental Requirements

1. The WBC FPGA will be surface mounted on the Station Board PCB. Additional heat sinks may be attached to the FPGA to reduce temperature.

2. The board will use forced-air cooling with a normal operating ambient temperature of 20 °C and with a maximum ambient temperature of 40 °C.

4.4 Interface Requirements

4.4.1 Master Reset

- RESET* is the input low-true master reset signal.

- RESET* is LVTTL.

4.4.2 System Input Signals

- SCLK is the system 128 MHz clock. This clock is derived on the Station Board from a 64 MHz that comes from the TIMECODE Generator Box (RFS Document AN25151N0000) using a crystal phase-locked loop. All FPGAs on the Station Board use SCLK in order to prevent the buildup of clock jitter. Input signals from the previous FPGA are delayed using CMIB controlled tapped delay lines in each IOB and then clocked in using a zero delay buffered 2X version of SCLK, namely SCLK_256. The amount of delay required is dependent on the length of PCB traces and will be constant for a specific FPGA on all boards.

- STICK is the system timing tick (10 millisecond period). It is used, along with a time interval counter, to help determine the best setting for the wide band input signal tapped delay line.

- All system input signals are 2.5V LVDS.

4.4.3 Wideband Input Signals (A and B)

- ICLK is a 128 MHz clock used for clocking in its accompanying data.

- IDATA[63:0] is clocked in by 2x ICLK. IDATA is the input data highway and can be in any of the organizations listed in Table 4-1. IDATA[63] is the most significant bit of the first sample in time.

- IVALID is clocked in by 2x ICLK. If low, the associated data is invalid and does not form part of the output data.

- INOISE is clocked in by 2x ICLK. If high, the calibration noise diode at the antenna was on for the associated samples. If low, the noise source was off.

- ITICK is clocked in by 2x ICLK. ITICK is a single high bit every tick time (10 milliseconds) and marks the first sample in IDATA as being associated with the tick time. The length of ITICK is one half a cycle of SCLK.

- IDFRM and IDERR are clocked in by ICLK and clocked out by SCLK but are not used in the WBC FPGA.

- All wideband input signals are LVCMOS.

4.4.4 Wideband Output Signals (A and B)

- OCLK is a 128 MHz clock derived from SCLK. It is used to clock in its accompanying data by the next FPGA in the chain. The timing relationship of OCLK to its accompanying data is the same as for ICLK.
- ODATA, OVALID, ONOISE and OTICK are copies of IDATA, IVALID, INOISE and ITICK clocked out by 2x SCLK. ODERR and ODFRM are copies of IDERR and IDFRM clocked out by SCLK.
- All wideband output signals are LVCMOS or HSTL.

4.4.5 MCB Interface Signals

- MCB_ADDR[7:0] is the input 8-bit address bus for accessing internal WBC FPGA configuration, monitor and control registers.
- MCB_DATA[15:0] is the bi-directional 16-bit microprocessor data bus.
- MCB_CS* is the input low-true FPGA select that enables the MCB interface drivers.
- MCB_CLK is the input clock for the synchronous MCB interface. The phase and frequency of MCB_CLK is independent of SCLK.
- MCB_RD/WR* is the input read/write enable.
- All MCB signals are LVTTL.

4.4.6 Test Port

These four outputs can be attached to a number of internal signals TBD to provide a simple diagnostic capability.

4.4.7 System Interface Timing Requirements

Shown below is the functional timing for the signals described in section 4.1.

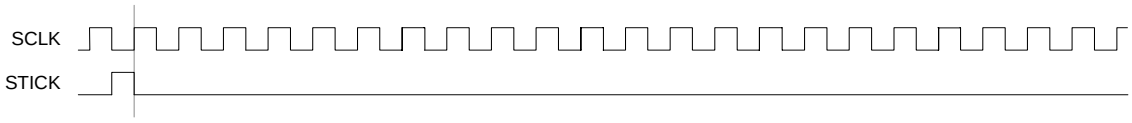

Figure 4-1 System Functional Relative Timing

The relative timing inside and outside the FPGA are the same as shown below. The left vertical line indicates the position of the internally generated rising clock edge that is

used to clock the I/O registers. The actual position of the 256 MHz clock is at 0.75 of the bit cell. Changes in the value of the output data coincides with the falling edge of the tick, regardless of the decimation ratio.

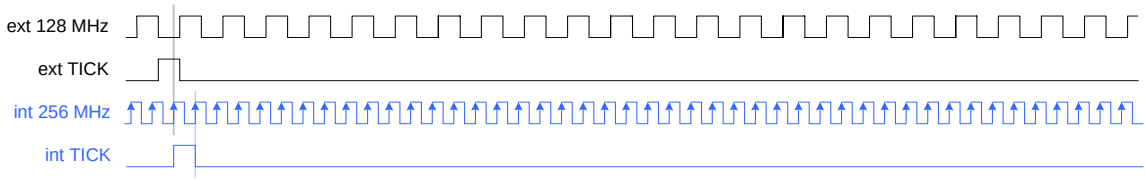

Figure 4-2 External/Internal Functional Relative Timing

4.4.8 Wide Band Data Interface Timing Requirements

Shown below is the functional relative timing for the wide band input and output signals. The signal names are prefixed with I for input and O for output.

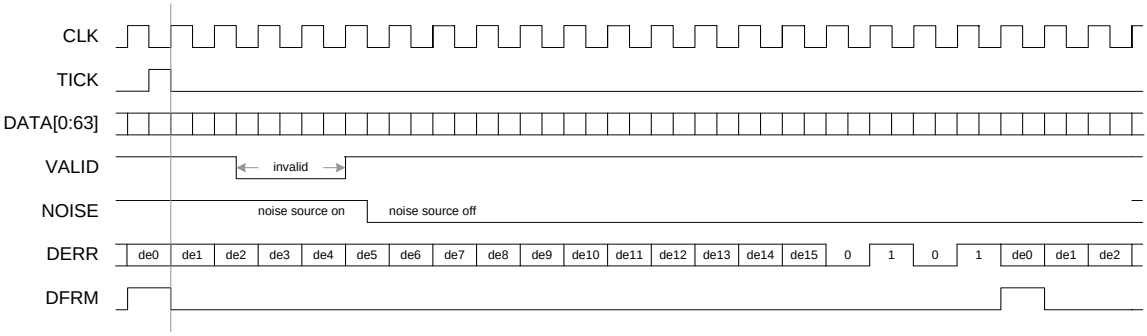

Figure 4-3 Wide Band Functional Input and Output Relative Timing

4.4.9 Relative MCB Interface Requirements

The MCB (Monitor & Control Bus) interface allows a microprocessor, the CMIB, to write to and read from the FPGA. Writing will be used to configure and control the hardware. Reading will be used to verify the configuration, receive status indicators and obtain monitor information.

The microprocessor writes to the FPGA by putting the data on the MCB_DATA bus, the target register address on the MCB_ADDR bus, driving MCB_RD/WR* and MCB_CS* low some time before a rising edge of the MCB_CLK and keeping the signals stable until some time after the rising edge of the MCB_CLK. If MCB_CS* and MCB_RD/WR* are both low, the WBC FPGA then writes the data into the specified register on the rising edge of MCB_CLK. A write requires one clock cycle as shown below.

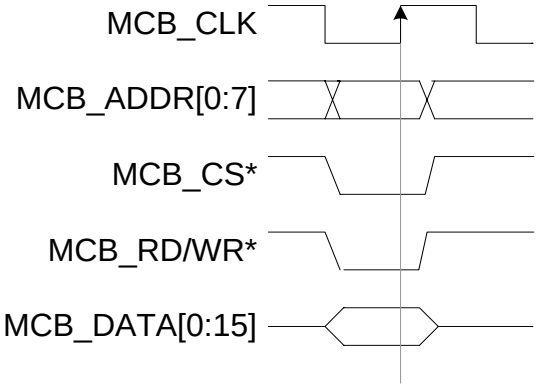

Figure 4-4 MCB Interface WRITE Functional Relative Timing

The microprocessor reads from a FPGA by driving MCB_RD/WR* high, putting the desired register address on the MCB_ADDR bus and driving the corresponding MCB_CS* low some time before a rising edge of MCB_CLK. If MCB_CS* is low and MCB_RD/WR* is high, the FPGA then clocks the address into a register on the rising edge (A) of MCB_CLK and places the contents of the address contained in the register onto the MCB_DATA bus as long as MCB_CS* is low. The microprocessor then takes the data on the next rising edge (B) of MCB_CLK. The address says valid until the microprocessor takes the data. Read cycles require one clock cycle to setup and an additional clock cycle to read as shown below. Subsequent reads at the same address take only one clock cycle depending on the microprocessor. It should be noted that for the read case, the MCB_CLK from the PCMC through the Decode FPGA and into the WBC FPGA suffers a delay and for the data path back to the PCMC the delay is even worse (bus selectors in the Decode FPGA); therefore, the read should be placed onto the MCB_DATA bus as soon as possible (no intervening registers).

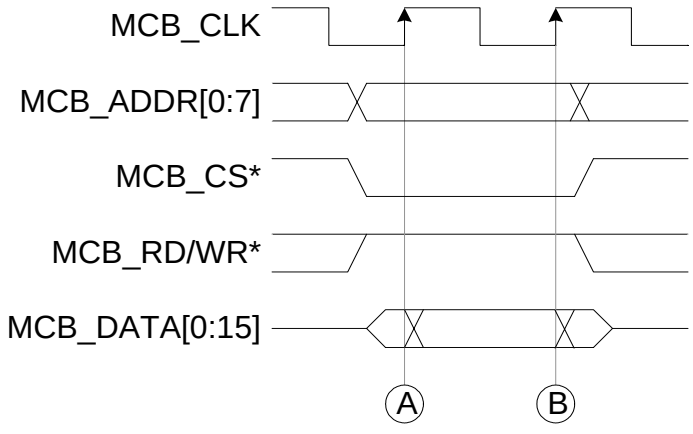

Figure 4-5 MCB Interface READ Functional Timing

The microprocessor will be interrupted on a version of STICK that has been delayed sufficiently to make sure that all FPGAs on the board have received their local version of the TICK. This allows for a pipeline delay through the FPGAs on the board, including the tap registers in the Filter FPGA. Configuration information sent to the WBC FPGA may take effect immediately; therefore, it should only be sent when the output of the FPGA is irrelevant. Delay information sent from the microprocessor to the WBC FPGA from within the Interrupt Service Routine (ISR) will be become active in hardware at the next TICK. Monitor information, such as counts or accumulations, read by the microprocessor from within the ISR will be the result of operations occurring between the previous two TICKS. It is assumed that status information, such as error bits, will be read by the ISR on every interrupt (TICK); therefore, the information is latched by the TICK and the primary register cleared to ensure that no momentary error bits are missed or counted twice.

5 Hardware Functional Design

This section expands on the blocks shown in Figure 3-2.

5.1 MCBI

The MCBI block contains the register set. The configuration information contained therein is fed to the other blocks as "constant" input. Some of these inputs will be latched in the block by the local version of the tick to insure correct timing. The other blocks may also return information that has usually been latched in the block by the local tick to insure correct timing. By "timing" in these cases we mean, for example, that a power measurement has been made by summing the squares of the input data over a precisely known time period or that the delay model takes effect at an exactly known time relative to the data.

5.2 INOUT

A simplified block diagram of the INOUT block is shown below.

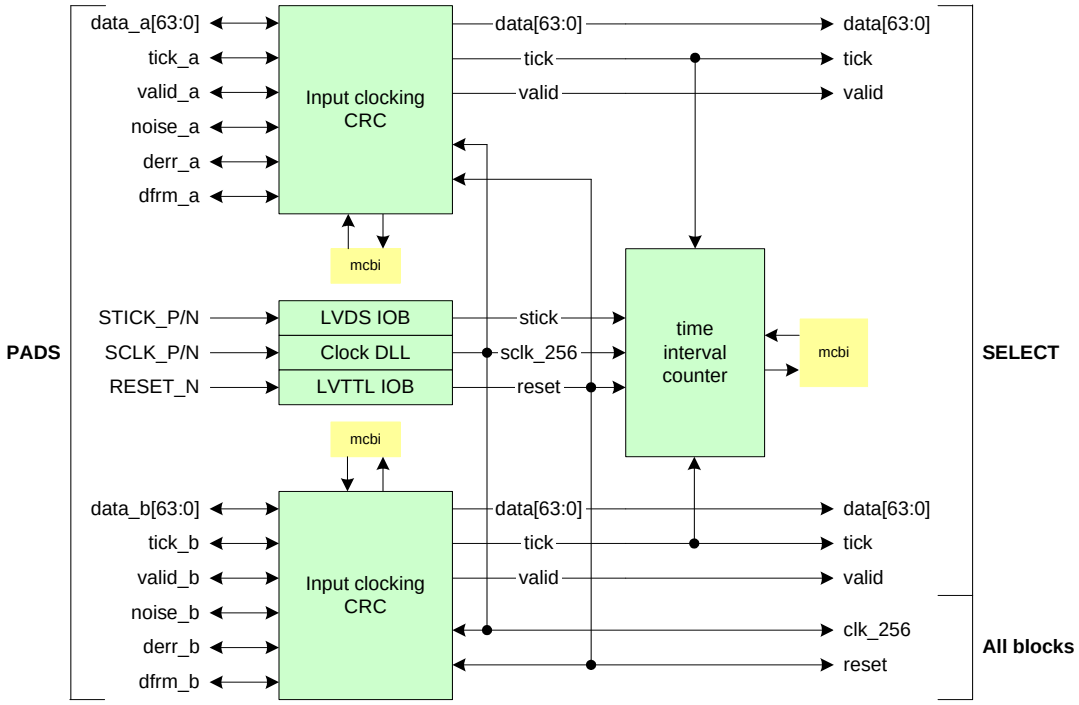

Figure 5-1 Block diagram of INOUT

INOUT receives signals from the two Delay Modules and then clocks them in with both edges of an internal 256 MHz version of the system clock. The edge that is used is set by the CMIB and is independent for the two base bands. The actual settings will be determined empirically during Station Board testing and should be the same for all

Station Boards. Copies of these signals are also clocked out by the system clock to the filter banks. The accompanying tick and system tick are used to ensure that the output timing of the FPGA has a constant timing offset relative to the system timing. Passing the data through the FPGA and out again allows point-to-point connections between FPGAs, simplifying PCB layout at the cost of somewhat increased power consumption. The design includes a time interval counter and CRC generators.

5.3 SELECT

A simplified block diagram of SELECT is shown below.

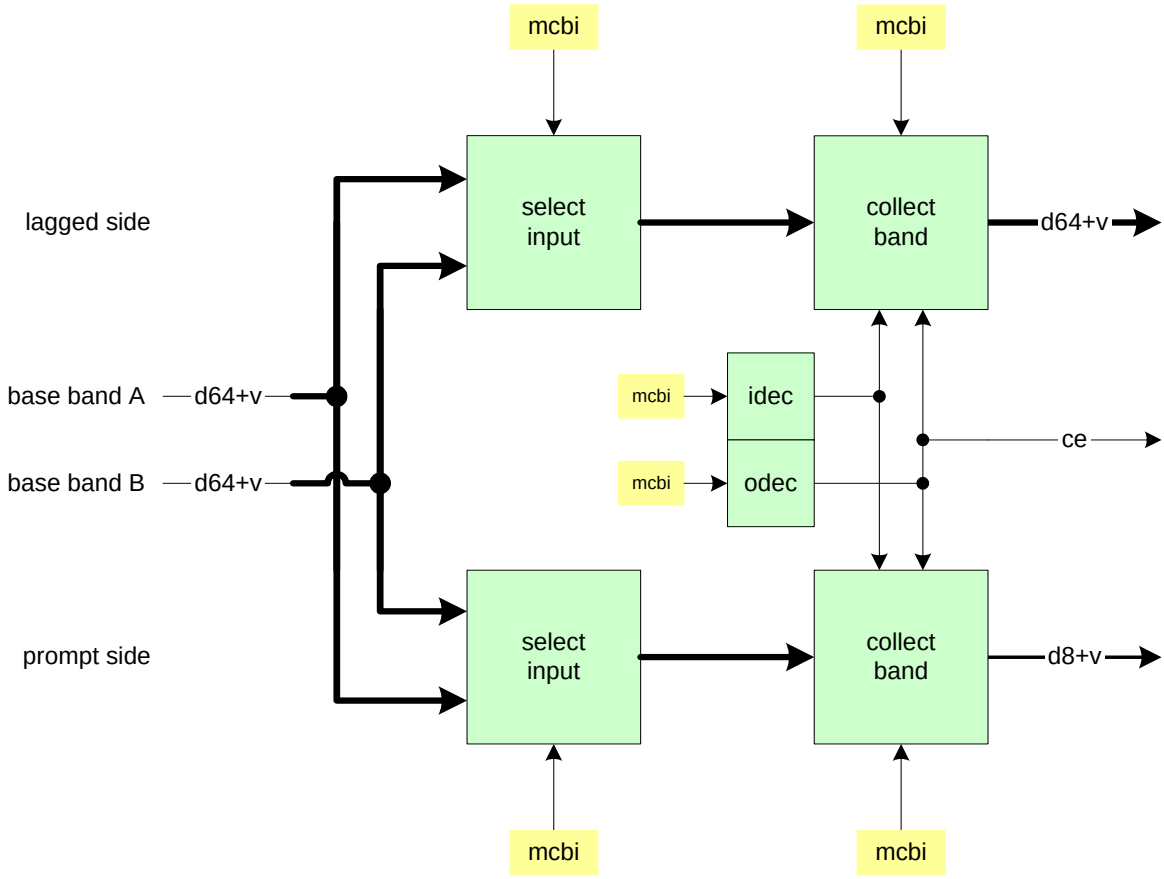

Figure 5-2 Block diagram of SELECT

SELECT chooses two output from the two wide band inputs, one will become the "lagged" data and the other will become the "prompt" data. All streams of the "lagged" data are necessary, so all streams are passed on to the next block. Only one stream of "prompt" data is used, so only one stream of eight bits (the maximum needed) is passed on to the next block. The single sample width for the "prompt" side is due to FPGA resource limitations, and hence the SNR is reduced by a factor of √16 for 4-bit samples and √8 for 8-bit samples. Additionally, a spectral line may show sub-carriers above and

below due to the under sampling of the prompt side. The block in Figure 5-2 labeled "collect band" contains a 16-word shift register to accommodate up to 16 bands per word (see Table 3-1). The lagged side output contains eight 8-bit samples or sixteen 4-bit samples but at a reduced word rate depending on the number of bands.

5.4 DELAY

A simplified block diagram of DELAY is shown below.

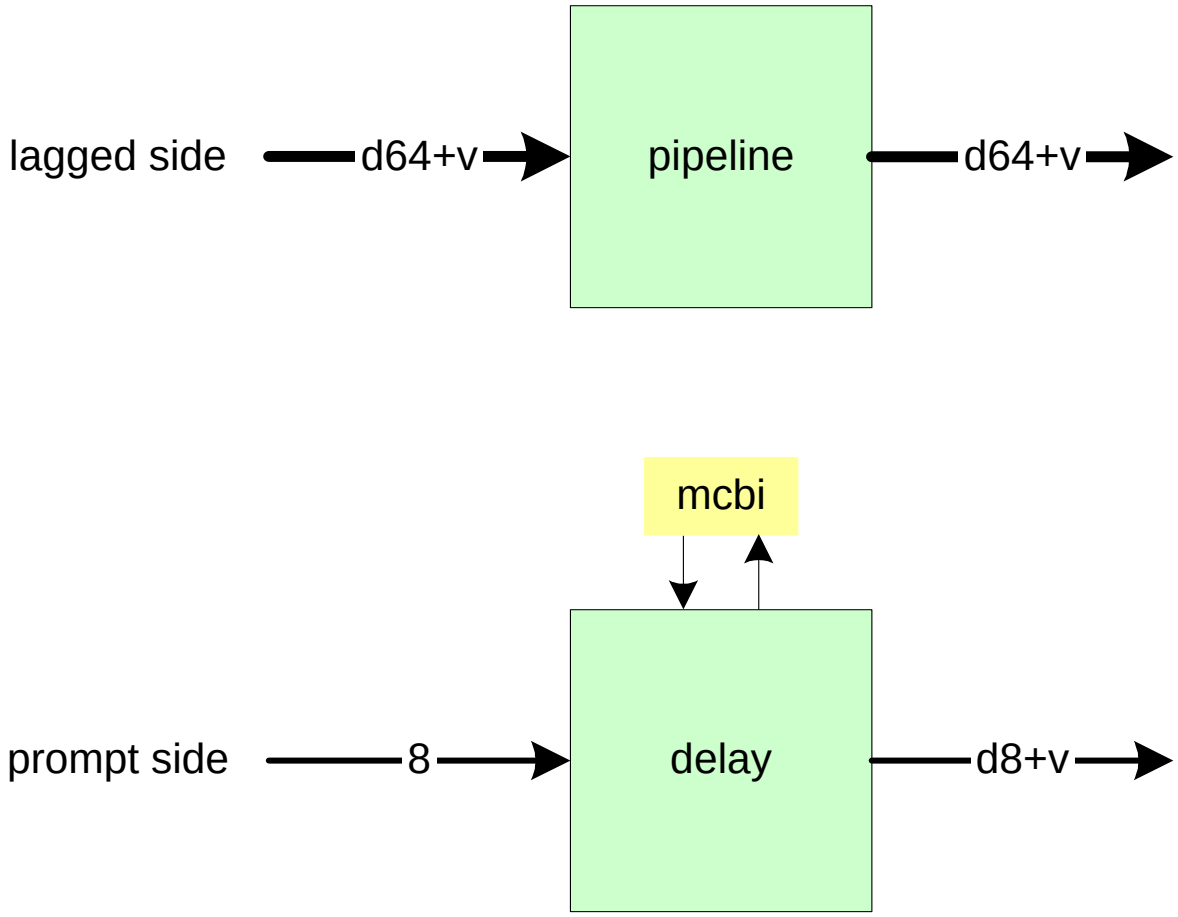

Figure 5-3 Block diagram of DELAY

DELAY delays the "prompt" data relative to the "lagged" data. The delay to be implemented is given to the WBC FPGA by the CMIB and is to be correct on the next tick. The length of the delay line is 16384 samples. The prompt side is delayed in this design to reduce the size of the delay line.

5.5 XCOR

It is assumed that the samples are encoded as "offset binary" by lookup tables in the Input FPGA. For this encoding the values are 0 to 7 for 3-bit samples (the MS bit of the '4-bit' sample is ignored) and 0 to 255 for 8-bit samples and that 0 represents the most negative value and 7(255) represent the most positive value. This form is chosen over 2s complement so that the 8-bit samples only need 8 bits (as opposed to 9). Offset binary is carried through to the multiplier-accumulator where it is finally converted to 2s complement by bit manipulation (see Table 5-1).

A simplified block diagram of XCOR is shown below.

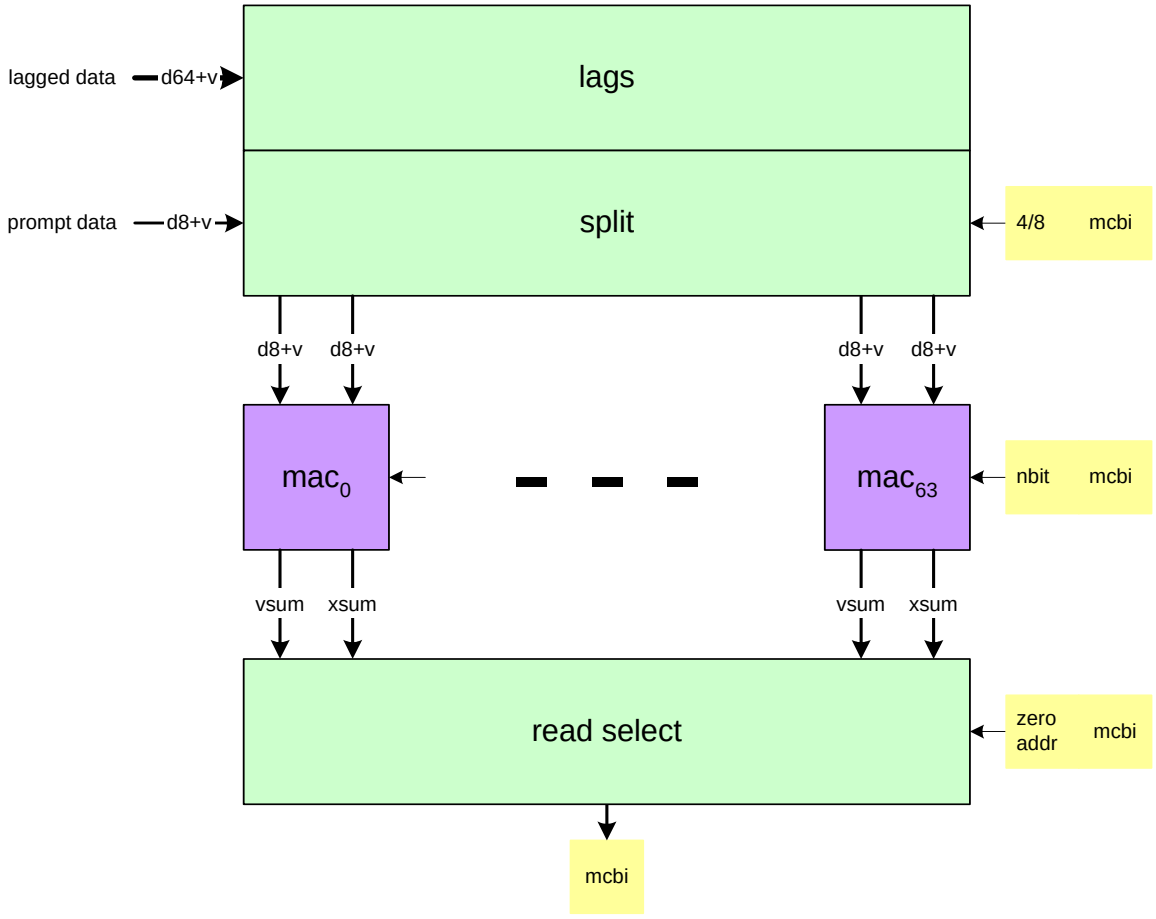

Figure 5-4a Block diagram of XCOR

XCOR consists of 64 multiplier/accumulators (MACs). There are two inputs to each MAC – lagged and prompt, both coming from the DELAY block. The prompt input is the same for all MACs. The lagged input must be one of 64 consecutive samples. These come from a 64-bit wide shift register eight words long. The shift register only needs to be four words long for the 4-bit case but eight are necessary for the 8-bit case.

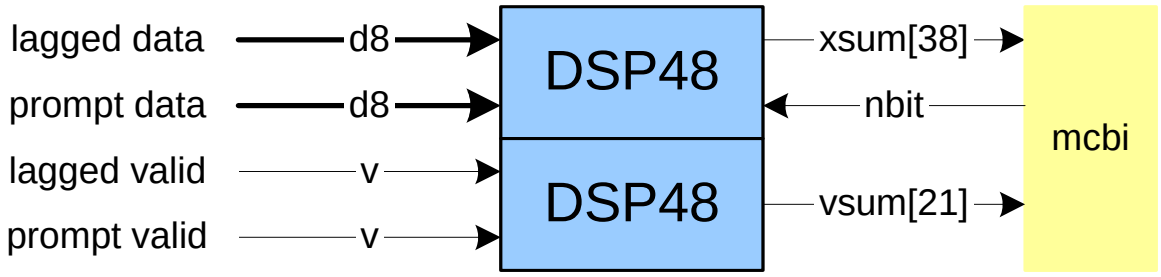

Figure 5-4b Block diagram of MAC

The use of the DSP48 for the valid counter is a bit of a waste but keeps power down. If an attempt to do 128 lags simultaneously were desired then the valid counter could be put in general logic. The cross-correlator DSP48 is given NBIT so that it can convert the offset binary input to symmetric 2's complement for any number of bits from 1 to 8. For example, the following conversion is done for the 3-bit case.

Voltage	Offset Binary	2's Complement
Most positive	7	+7
	6	+5
	5	+3
	4	+1
	3	-1
	2	-3
	1	-5
Most negative	0	-7

Table 5-1 Conversion for 3-bit Samples

6 Software Design Information

6.1 MCB Interface Register Map

The register map of the WBC FPGA is shown below.

Address	Name	Dir	Bits	Description
COMMON				**Common Registers**
0x00	CM_STS	R/W	15:0	Status Register
0x01	CM_CFG	R/W	15:0	Configuration Register
0x02	CM_CTL	R/W	15:0	Control Register
0x03	CM_ERR	R/W	2:0	Error Register
0x04	CM_DEF	R	15:0	Default Register
0x05	CM_DID	R	15:0	Design Identifier
0x06	CM_TST0	R/W	7:0	Address of signal to connect to test port 0
0x07	CM_TST1	R/W	7:0	Address of signal to connect to test port 1
0x08	CM_TST2	R/W	7:0	Address of signal to connect to test port 2
0x09	CM_TST3	R/W	7:0	Address of signal to connect to test port 3

Address	Name	Dir	Bits	Description
INOUT				**Wideband input/output**
0x10	IO_ESEL_A	R/W	6:0	Error select for wide band A CRC.
0x11	IO_ESEL_B	R/W	6:0	Error select for wide band B CRC.
0x12	IO_DSEL_A	R/W	5:0	Wide band A bit for CRC calculation
0x13	IO_ICRC_A	R/W	3:0	4-bit CRC for selected A input bit
0x14	IO_DSEL_B	R/W	5:0	Wide band B bit for CRC calculation
0x15	IO_ICRC_B	R/W	3:0	4-bit CRC for selected B input bit
0x16	IO_TINT1_A	R/W	15:0	MS time interval data tick to system tick
0x17	IO_TINT0_A	R/W	15:0	LS time interval
0x18	IO_TINT1_B	R/W	15:0	MS time interval data tick to system tick
0x19	IO_TINT0_B	R/W	15:0	LS time interval
0x1A	IO_SEED_A	R/W	15:0	Seed for pseudo random bit stream generator A
0x1B	IO_SEED_B	R/W	15:0	Seed for pseudo random bit stream generator B
0x1C	IO_SDLY	R/W	15:0	System tick delay to offset IO_TINT_A/B.
0x1D	IO_OCRC_A	R/W	3:0	4-bit CRC for selected A output bit
0x1E	IO_OCRC_B	R/W	3:0	4-bit CRC for selected B output bit
0x1F	IO_DERR_A	R/W	15:0	Delay error for wideband test pattern generator
0x20	IO_DERR_B	R/W	15:0	Delay error for wideband test pattern generator
SELECT				**Select signals to process**
0x30	SL_IDEC	R/W	3:0	Input data rate divider
0x31	SL_NBND	R/W	3:0	Number of bands in word – 1
0x32	SL_DBND	R/W	3:0	Lagged band to process – 1

Address	Name	Dir	Bits	Description
0x33	SL_PBND	R/W	3:0	Prompt band to process – 1
0x34	SL_ODEC	R/W	3:0	Output data rate divider
DELAY				**Lag delay**
0x40	DL_LDLY	R/W	14:0	Lag delay
0x41	DL_PLDLY	R	14:0	Previous Lag delay
XCOR				**Results**
0x60	XC_NBIT	R/W	2:0	Number of bits in samples– 1
0x61	XC_VCNT1	R	5:0	Valid count [21:16]
0x62	XC_VCNT0	R	15:0	Valid count [15:0]
0x63	XC_PACC2	R	6:0	Product accumulation [38:32]
0x64	XC_PACC1	R	15:0	Product accumulation [32:16]
0x65	XC_PACC0	R	15:0	Product accumulation [15:0]

Table 6-1 Register Map

6.2 Selecting the Signals to Correlate

Either baseband can be selected for each of the lagged and prompt sides using bits in the configuration register. Any band can be selected for each of the lagged and prompt sides using the SL_DBND and SL_PBND registers.

6.3 Changing the Lag range.

DL_LDLY is the extra delay in words used to increase the lag range in sets of 64 lags. Normally, DL_LDLY would be started at zero and incremented every interrupt until the maximum number of lags had been reached. The whole process would then be repeated *ad infinitum* with accumulation taking place in the CMIB or elsewhere. The value of the increment to DL_LDLY for a new set of 64 lags is dependent on XC_NBIT. For XC_NBIT = 0 to 3, the increment is 4. For XC_NBIT = 4 to 7, the increment is 8. The increment is independent of the number of bands.

6.4 Setting the Input and Output Rate Dividers

The output decimation, SL_ODEC, depends on SL_IDEC and the number of bands. If the number of bands is 1 (SL_NBND = 0), then SL_ODEC = SL_IDEC. As the number of bands increases the output divider must increase appropriately relative to the input divider. For 4 bands, SL_ODEC = SL_IDEC + 2 (see Table 7-7). For the current VLBA case (mode 4_1 and 32 Ms/s per band), SL_IDEC = 3 and SL_ODEC = 7.

6.5 Reading Results

To read the valid count set the corresponding XCOR control bit to 1 and then back to 0 to reset the count read address to zero, read bits [31:16] first and then bits [15:0]. To read the cross-correlation accumulations set the corresponding XCOR control bit to 1 and then back to 0 to reset the accumulation read address to zero, read bits [47:32], then bits [31:16] and finally bits [15:0]. In both cases the address is incremented after reading bits [15:0]. Both the valid count and the accumulator addresses can be reset simultaneously and the reads can be interleaved but the LS part of each quantity must be read last.

6.6 Setting the Input Clock Edges

The clock edge selection bits are set by the CMIB to make sure that the positive going edge of the global 256 MHz clock is away from transitions in the incoming signal lines. Which edge to use will be determined during prototype testing and should be the same for all Station Boards.

6.7 The Time Interval Counters

The time interval counters can be used to measure the time interval between the data tick for each wide band input and the system tick.

7 Detailed Register Description

The following is a detailed description of the WBC FPGA register set for each block of the design. The number of bits reflects the actual hardware register but on read the upper bits in the register are sign extended or zero; therefore, no masking by the CMIB is necessary when reading the full 16-bit register. Many of the registers are configuration information which is only changed for a new observation.

7.1 COMMON

The following registers are used by all blocks for status, configuration, control, error, testing and design identifier. The status, configuration, control and error registers are collections of single bits gathered together for simplicity.

Address	Name	Access	Bits	Description
0x00	CM_STS	R/W	15:0	Status Register. This register indicates various error conditions in the WBC FPGA as shown in Table 7-2. The status bits are saved and the working register cleared on the tick; therefore, the ISR should read the status register every interrupt to avoid the loss of an error indicator. If the status register is zero, there are no errors. To create errors for test purposes, write bits to the register. The original status bits will be inverted wherever the written bits are 1.
0x01	CM_CFG	R/W	15:0	Configuration Register. This register is composed of individual bits that are normally set during configuration and then left alone during operation. See Table 7-3.
0x02	CM_CTL	R/W	15:0	Control Register. This register is composed of individual bits that can be changed during operation. See Table 7-4.
0x03	CM_ERR	R/W	2:0	Error Register. This register should always be zero. If not, one of the errors in Table 7-5 has occurred since it was last cleared. To clear the register, write zero to it. To create errors, commit an error or write a non-zero value to the register.
0x04	CM_DEF	R/W	15:0	Default Register. This register will contain the last attempt to write to a non-existent or read-only register and will be returned if an attempt is made to read from a non-existent register. It can also used as a test register.
0x05	CM_DID	R	15:0	Design identifier. The bits are divided into design, revision and version. Since the WBC FPGA is programmable, different designs are possible.

0x06	CM_TST0	R/W	7:0	Address of signal to connect to test port 0
0x07	CM_TST1	R/W	7:0	Address of signal to connect to test port 1
0x08	CM_TST2	R/W	7:0	Address of signal to connect to test port 2
0x09	CM_TST3	R/W	7:0	Address of signal to connect to test port 3

Table 7-1 COMMON Registers

The meaning of bits in the status registers is shown below.

Bit	Block	Meaning (1 = error)
0-1		No longer used
2	INOUT	1 => STICK error (width)
3	INOUT	1 => SCLK not locked
4	SELECT	Illegal number of bands
5	SELECT	Illegal lagged band
6	SELECT	Illegal prompt band
7	SELECT	Illegal combination
8	INOUT	1 => DCM phase shift is complete (test only)
9	INOUT	1 => DCM phase shift overflow or underflow (test only)
10	INOUT	1 => STICK with edge + matches STICK with edge - (test only)
11	INOUT	1 => STICK with chosen edge leads STICK not chosen (test only)

Table 7-2 Status Bit Definitions

Bit	Block	Meaning (default = 0)
0	SELECT	0 => A, 1 => B wide band input for lagged data
1	SELECT	0 => A, 1 => B wide band input for prompt data
2	INOUT	0 => normal, 1 => put wide band test signals on output
3	INOUT	Selects which clock edge to use for the system tick input.
4	INOUT	Selects which clock edge to use for the A wide band data input.
5	INOUT	Selects which clock edge to use for the B wide band data input.
6	INOUT	0 => pseudo random, 1 => delta function wide band test signals
7	INOUT	0 => test data always valid, 1 => invalid during tick (one sample)
8	INOUT	add one clock delay to A wide band input to time align with B
9	INOUT	add one clock delay to B wide band input to time align with A

Table 7-3 Configuration Bit Definitions

Bit	Block	Meaning
0	INOUT	0 to 1 => momentary software reset
1	INOUT	1 => disable global clocking to cut heat production, 0 => normal
2	XCOR	0 to 1 => set accumulator starting read address to zero
3	XCOR	0 to 1 => set valid counter starting read address to zero
4	INOUT	1 => increment DCM phase, 0 => decrement DCM phase
5	INOUT	1 => shift DCM phase (must return to zero before another shift)
15	INOUT	1 => SCLK PLL is held in reset

Table 7-4 Control Bit Definitions

Bit	Meaning
0	1=> attempt to write to a read-only register
1	1=> attempt to write to a non-existent register
2	1 => attempt to read from a non-existent register

Table 7-5 Error Bit Definitions

ID Part	Bits	CM_DID
Design	8	[15:8]
Revision	4	[7:4]
Version	4	[3:0]

Table 7-6 Design ID Definition

Divider	Value	MHz
1	0	256
2	1	128
4	2	64
8	3	32
16	4	16
32	5	8
64	6	4
128	7	2
256	8	1
512	9	0.5
1024	10	0.25
2048	11	0.125
4096	12	0.0625
4096	13	0.0625
4096	14	0.0625
4096	15	0.0625

Table 7-7 Clock Divider Values

The range of dividers is limited to 4096 to ensure that there is an integral number of the lowest clock rate (0.0625 MHz) in a tick interval (10 milliseconds). If the tick (interrupt) interval were ever shortened to 5 or 2 milliseconds (the only other feasible values) in order to produce more accurate models for VLBI, then the divider range would be limited to 2048 and 4096 respectively. The upper limit of 10 milliseconds for the tick interval is set by the number of bits in the FPGAs accumulators.

The test port signals may be used during the debugging of the FPGA but are not normally needed. Set all four addresses to zero to prevent the corresponding output pins from changing

7.2 INOUT

Address	Name	Access	Bits	Description
0x10	IO_ESEL_A	R/W	6:0	Error select for wide band A CRC. If bit 6 is set to 1, bits [5:0] will cause a CRC error whenever IO_DSEL_A is set to these bits. This can be used to create CRC errors on a specific wire to test the software.
0x11	IO_ESEL_B	R/W	6:0	Error select for wide band B CRC. If bit 6 is set to 1, bits [5:0] will cause an inverted CRC whenever IO_DSEL_B is set to these bits. This can be used to create CRC errors on a specific wire to test the software.
0x12	IO_DSEL_A	R/W	6:0	Base band A data select for CRC calculation. If bit 6 is set to 1 then other lines replace the LS data bit wires for CRC calculation only: wire[0]=>VALID, wire [1]=>NOISE, wire[2]=>DERR, wire [3]=>DFRM, wire[4]=>CLK
0x13	IO_ICRC_A	R	3:0	4-bit CRC for selected A input bit.
0x14	IO_DSEL_B	R/W	6:0	Base band B data select for CRC calculation. If bit 6 is set to 1 then other lines replace the LS data bit wires for CRC calculation only: wire[0]=>VALID, wire [1]=>NOISE, wire[2]=>DERR, wire [3]=>DFRM, wire[4]=>CLK
0x15	IO_ICRC_B	R	3:0	4-bit CRC for selected B input bit.
0x16	IO_TINT1_A	R/W	15:14 5:0	Bits 15:14 select interval to measure. 00:dtick>stick, 01:dtick, 10:stick, 11:stick>dtick MS time interval in 256 MHz clocks.
0x17	IO_TINT0_A	R	15:0	LS time interval in 256 MHz clocks.
0x18	IO_TINT1_B	R/W	15:14 5:0	Bits 15:14 select interval to measure. 00:dtick>stick, 01:dtick, 10:stick, 11:stick>dtick MS time interval in 256 MHz clocks.
0x19	IO_TINT0_B	R	15:0	LS time interval in 256 MHz clocks

Address	Name	Access	Bits	Description
0x1A	IO_SEED_A	R/W	15:0	Seed for pseudo random bit stream generator. Allows banks A and B to be different for testing. Default value is 0x1357.
0x1B	IO_SEED_B	R/W	15:0	As above
0x1C	IO_SDLY	R/W	15:0	System tick delay for IO_TINT_A/B. This can be used to make the time interval count zero.
0x1D	IO_OCRC_A	R	3:0	4-bit CRC for selected A output bit.
0x1E	IO_OCRC_B	R	3:0	4-bit CRC for selected B output bit.
0x1F	IO_DERR_A	R/W	15:0	Delay error for wideband test pattern generator A
0x20	IO_DERR_B	R/W	15:0	Delay error for wideband test pattern generator B

Table 7-9 INOUT Registers

7.3 SELECT

Address	Name	Access	Bits	Description
0x30	SL_IDEC	R/W	3:0	Input data rate divider–1
0x31	SL_NBND	R/W	3:0	Number of bands in word – 1
0x32	SL_DBND	R/W	3:0	Lagged band to process – 1
0x33	SL_PBND	R/W	3:0	Prompt band to process – 1
0x34	SL_ODEC	R/W	3:0	Output data rate divider–1

Table 7-10 SELECT Registers

7.4 DELAY

Address	Name	Access	Bits	Description
0x40	DL_LDLY	R/W	13:0	Lagged data delay in 64-bit words (max 16384)
0x41	DL_PLDLY	R	13:0	Previous lagged data delay (goes with xcor output)

Table 7-11 DELAY Registers

7.5 XCOR

Address	Name	Access	Bits	Description
0x60	XC_NBIT	R/W	2:0	Number of bits in samples– 1
0x61	XC_VCNT1	R	5:0	Valid count [21:16]
0x62	XC_VCNT0	R	15:0	Valid count [15:0]
0x63	XC_PACC2	R	6:0	Product accumulation [38:32]
0x64	XC_PACC1	R	15:0	Product accumulation [32:16]
0x65	XC_PACC0	R	15:0	Product accumulation [15:0]

Table 7-12 XCOR Registers

8 Pinouts, Pin and Package Notes

8.1 Pinouts by signal name

Note: UNUSED and NC pins have been removed.

```
Release 10.1.03 - par K.39 (nt)
Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.

Thu Mar 25 09:01:15 2010

INPUT FILE:       wbc_top_map.ncd
OUTPUT FILE:      wbc_top_pad.txt
PART TYPE:        xc4vsx35
SPEED GRADE:      -10
PACKAGE:          ff668
```

Pin	Signal Name	Pin Usage	Pin Name	Direction	IO Standard	IO Bank
K6	idata_a<0>	IOB	IO_L3N_10	INPUT	HSTL_III_DCI	10
K7	idata_a<1>	IOB	IO_L3P_10	INPUT	HSTL_III_DCI	10
G1	idata_a<10>	IOB	IO_L30N_6	INPUT	HSTL_III_DCI	6
G2	idata_a<11>	IOB	IO_L30P_6	INPUT	HSTL_III_DCI	6
H5	idata_a<12>	IOB	IO_L29N_6	INPUT	HSTL_III_DCI	6
H6	idata_a<13>	IOB	IO_L29P_6	INPUT	HSTL_III_DCI	6
G4	idata_a<14>	IOB	IO_L28P_6	INPUT	HSTL_III_DCI	6
F3	idata_a<15>	IOB	IO_L27N_6	INPUT	HSTL_III_DCI	6
F4	idata_a<16>	IOB	IO_L27P_6	INPUT	HSTL_III_DCI	6
F1	idata_a<17>	IOB	IO_L26N_6	INPUT	HSTL_III_DCI	6
E1	idata_a<18>	IOB	IO_L26P_6	INPUT	HSTL_III_DCI	6
D1	idata_a<19>	IOB	IO_L25N_CC_LC_6	INPUT	HSTL_III_DCI	6
J4	idata_a<2>	IOB	IO_L2N_10	INPUT	HSTL_III_DCI	10

D2	idata_a<20>	IOB	IO_L25P_CC_LC_6	INPUT	HSTL_III_DCI	6
E2	idata_a<21>	IOB	IO_L24N_CC_LC_6	INPUT	HSTL_III_DCI	6
E3	idata_a<22>	IOB	IO_L24P_CC_LC_6	INPUT	HSTL_III_DCI	6
E4	idata_a<23>	IOB	IO_L22N_6	INPUT	HSTL_III_DCI	6
D3	idata_a<24>	IOB	IO_L22P_6	INPUT	HSTL_III_DCI	6
H7	idata_a<25>	IOB	IO_L21N_6	INPUT	HSTL_III_DCI	6
H8	idata_a<26>	IOB	IO_L21P_6	INPUT	HSTL_III_DCI	6
C2	idata_a<27>	IOB	IO_L20P_6	INPUT	HSTL_III_DCI	6
G7	idata_a<28>	IOB	IO_L19N_6	INPUT	HSTL_III_DCI	6
F7	idata_a<29>	IOB	IO_L19P_6	INPUT	HSTL_III_DCI	6
J5	idata_a<3>	IOB	IO_L2P_10	INPUT	HSTL_III_DCI	10
E5	idata_a<30>	IOB	IO_L18N_6	INPUT	HSTL_III_DCI	6
E6	idata_a<31>	IOB	IO_L18P_6	INPUT	HSTL_III_DCI	6
D6	idata_a<32>	IOB	IO_L17N_6	INPUT	HSTL_III_DCI	6
E7	idata_a<33>	IOB	IO_L17P_6	INPUT	HSTL_III_DCI	6
D4	idata_a<34>	IOB	IO_L16N_6	INPUT	HSTL_III_DCI	6
C4	idata_a<35>	IOB	IO_L16P_6	INPUT	HSTL_III_DCI	6
B4	idata_a<36>	IOB	IO_L15N_6	INPUT	HSTL_III_DCI	6
A4	idata_a<37>	IOB	IO_L15P_6	INPUT	HSTL_III_DCI	6
B3	idata_a<38>	IOB	IO_L14N_6	INPUT	HSTL_III_DCI	6
A3	idata_a<39>	IOB	IO_L14P_6	INPUT	HSTL_III_DCI	6
J6	idata_a<4>	IOB	IO_L1N_10	INPUT	HSTL_III_DCI	10
B9	idata_a<40>	IOB	IO_L13N_6	INPUT	HSTL_III_DCI	6
A9	idata_a<41>	IOB	IO_L13P_6	INPUT	HSTL_III_DCI	6
C5	idata_a<42>	IOB	IO_L12P_6	INPUT	HSTL_III_DCI	6
C7	idata_a<43>	IOB	IO_L11N_6	INPUT	HSTL_III_DCI	6
B7	idata_a<44>	IOB	IO_L11P_6	INPUT	HSTL_III_DCI	6
G8	idata_a<45>	IOB	IO_L10N_6	INPUT	HSTL_III_DCI	6
F8	idata_a<46>	IOB	IO_L10P_6	INPUT	HSTL_III_DCI	6
G9	idata_a<47>	IOB	IO_L9N_CC_LC_6	INPUT	HSTL_III_DCI	6
G10	idata_a<48>	IOB	IO_L9P_CC_LC_6	INPUT	HSTL_III_DCI	6
C6	idata_a<49>	IOB	IO_L8N_CC_LC_6	INPUT	HSTL_III_DCI	6
J7	idata_a<5>	IOB	IO_L1P_10	INPUT	HSTL_III_DCI	10
B6	idata_a<50>	IOB	IO_L8P_CC_LC_6	INPUT	HSTL_III_DCI	6

F9	idata_a<51>	IOB	IO_L7N_6	INPUT	HSTL_III_DCI	6
E9	idata_a<52>	IOB	IO_L7P_6	INPUT	HSTL_III_DCI	6
A5	idata_a<53>	IOB	IO_L6N_6	INPUT	HSTL_III_DCI	6
A6	idata_a<54>	IOB	IO_L6P_6	INPUT	HSTL_III_DCI	6
E10	idata_a<55>	IOB	IO_L5N_6	INPUT	HSTL_III_DCI	6
F10	idata_a<56>	IOB	IO_L5P_6	INPUT	HSTL_III_DCI	6
D8	idata_a<57>	IOB	IO_L4P_6	INPUT	HSTL_III_DCI	6
A7	idata_a<58>	IOB	IO_L3N_6	INPUT	HSTL_III_DCI	6
A8	idata_a<59>	IOB	IO_L3P_6	INPUT	HSTL_III_DCI	6
H1	idata_a<6>	IOB	IO_L32N_6	INPUT	HSTL_III_DCI	6
C8	idata_a<60>	IOB	IO_L2N_6	INPUT	HSTL_III_DCI	6
D9	idata_a<61>	IOB	IO_L2P_6	INPUT	HSTL_III_DCI	6
C10	idata_a<62>	IOB	IO_L1N_6	INPUT	HSTL_III_DCI	6
D10	idata_a<63>	IOB	IO_L1P_6	INPUT	HSTL_III_DCI	6
H2	idata_a<7>	IOB	IO_L32P_6	INPUT	HSTL_III_DCI	6
H3	idata_a<8>	IOB	IO_L31N_6	INPUT	HSTL_III_DCI	6
H4	idata_a<9>	IOB	IO_L31P_6	INPUT	HSTL_III_DCI	6
AB9	idata_b<0>	IOB	IO_L29N_8	INPUT	HSTL_III_DCI	8
AC9	idata_b<1>	IOB	IO_L29P_8	INPUT	HSTL_III_DCI	8
AC6	idata_b<10>	IOB	IO_L24P_CC_LC_8	INPUT	HSTL_III_DCI	8
AD4	idata_b<11>	IOB	IO_L22N_8	INPUT	HSTL_III_DCI	8
AD5	idata_b<12>	IOB	IO_L22P_8	INPUT	HSTL_III_DCI	8
Y9	idata_b<13>	IOB	IO_L21N_8	INPUT	HSTL_III_DCI	8
AA9	idata_b<14>	IOB	IO_L21P_8	INPUT	HSTL_III_DCI	8
AA7	idata_b<15>	IOB	IO_L20P_8	INPUT	HSTL_III_DCI	8
AF5	idata_b<16>	IOB	IO_L19N_8	INPUT	HSTL_III_DCI	8
AF6	idata_b<17>	IOB	IO_L19P_8	INPUT	HSTL_III_DCI	8
AC3	idata_b<18>	IOB	IO_L18N_8	INPUT	HSTL_III_DCI	8
AD3	idata_b<19>	IOB	IO_L18P_8	INPUT	HSTL_III_DCI	8
AC7	idata_b<2>	IOB	IO_L28P_8	INPUT	HSTL_III_DCI	8
AE4	idata_b<20>	IOB	IO_L17N_8	INPUT	HSTL_III_DCI	8
AF4	idata_b<21>	IOB	IO_L17P_8	INPUT	HSTL_III_DCI	8
AD1	idata_b<22>	IOB	IO_L16N_8	INPUT	HSTL_III_DCI	8
AD2	idata_b<23>	IOB	IO_L16P_8	INPUT	HSTL_III_DCI	8

AE3	idata_b<24>	IOB	IO_L15N_8	INPUT	HSTL_III_DCI	8
AF3	idata_b<25>	IOB	IO_L15P_8	INPUT	HSTL_III_DCI	8
AC1	idata_b<26>	IOB	IO_L14N_8	INPUT	HSTL_III_DCI	8
AC2	idata_b<27>	IOB	IO_L14P_8	INPUT	HSTL_III_DCI	8
AB5	idata_b<28>	IOB	IO_L13N_8	INPUT	HSTL_III_DCI	8
AC5	idata_b<29>	IOB	IO_L13P_8	INPUT	HSTL_III_DCI	8
AA10	idata_b<3>	IOB	IO_L27N_8	INPUT	HSTL_III_DCI	8
AB3	idata_b<30>	IOB	IO_L12P_8	INPUT	HSTL_III_DCI	8
AB4	idata_b<31>	IOB	IO_L11N_8	INPUT	HSTL_III_DCI	8
AC4	idata_b<32>	IOB	IO_L11P_8	INPUT	HSTL_III_DCI	8
AA1	idata_b<33>	IOB	IO_L10N_8	INPUT	HSTL_III_DCI	8
AB1	idata_b<34>	IOB	IO_L10P_8	INPUT	HSTL_III_DCI	8
Y5	idata_b<35>	IOB	IO_L9N_CC_LC_8	INPUT	HSTL_III_DCI	8
Y6	idata_b<36>	IOB	IO_L9P_CC_LC_8	INPUT	HSTL_III_DCI	8
Y3	idata_b<37>	IOB	IO_L8N_CC_LC_8	INPUT	HSTL_III_DCI	8
Y4	idata_b<38>	IOB	IO_L8P_CC_LC_8	INPUT	HSTL_III_DCI	8
AA3	idata_b<39>	IOB	IO_L7N_8	INPUT	HSTL_III_DCI	8
Y10	idata_b<4>	IOB	IO_L27P_8	INPUT	HSTL_III_DCI	8
AA4	idata_b<40>	IOB	IO_L7P_8	INPUT	HSTL_III_DCI	8
Y1	idata_b<41>	IOB	IO_L6N_8	INPUT	HSTL_III_DCI	8
Y2	idata_b<42>	IOB	IO_L6P_8	INPUT	HSTL_III_DCI	8
W5	idata_b<43>	IOB	IO_L5N_8	INPUT	HSTL_III_DCI	8
W6	idata_b<44>	IOB	IO_L5P_8	INPUT	HSTL_III_DCI	8
W4	idata_b<45>	IOB	IO_L4P_8	INPUT	HSTL_III_DCI	8
V7	idata_b<46>	IOB	IO_L3N_8	INPUT	HSTL_III_DCI	8
W7	idata_b<47>	IOB	IO_L3P_8	INPUT	HSTL_III_DCI	8
V5	idata_b<48>	IOB	IO_L2N_8	INPUT	HSTL_III_DCI	8
V6	idata_b<49>	IOB	IO_L2P_8	INPUT	HSTL_III_DCI	8
Y8	idata_b<5>	IOB	IO_L26N_8	INPUT	HSTL_III_DCI	8
W1	idata_b<50>	IOB	IO_L1N_8	INPUT	HSTL_III_DCI	8
W2	idata_b<51>	IOB	IO_L1P_8	INPUT	HSTL_III_DCI	8
U5	idata_b<52>	IOB	IO_L32N_10	INPUT	HSTL_III_DCI	10
U6	idata_b<53>	IOB	IO_L32P_10	INPUT	HSTL_III_DCI	10
U7	idata_b<54>	IOB	IO_L31N_10	INPUT	HSTL_III_DCI	10

T8	idata_b<55>	IOB	IO_L31P_10	INPUT	HSTL_III_DCI	10
V1	idata_b<56>	IOB	IO_L30N_10	INPUT	HSTL_III_DCI	10
V2	idata_b<57>	IOB	IO_L30P_10	INPUT	HSTL_III_DCI	10
U4	idata_b<58>	IOB	IO_L29N_10	INPUT	HSTL_III_DCI	10
V4	idata_b<59>	IOB	IO_L29P_10	INPUT	HSTL_III_DCI	10
AA8	idata_b<6>	IOB	IO_L26P_8	INPUT	HSTL_III_DCI	8
U3	idata_b<60>	IOB	IO_L28P_10	INPUT	HSTL_III_DCI	10
T6	idata_b<61>	IOB	IO_L27N_10	INPUT	HSTL_III_DCI	10
T7	idata_b<62>	IOB	IO_L27P_10	INPUT	HSTL_III_DCI	10
T3	idata_b<63>	IOB	IO_L26N_10	INPUT	HSTL_III_DCI	10
AF7	idata_b<7>	IOB	IO_L25N_CC_LC_8	INPUT	HSTL_III_DCI	8
AF8	idata_b<8>	IOB	IO_L25P_CC_LC_8	INPUT	HSTL_III_DCI	8
AB6	idata_b<9>	IOB	IO_L24N_CC_LC_8	INPUT	HSTL_III_DCI	8
K3	idclk_a	IOB	IO_L7P_10	INPUT	HSTL_III_DCI	10
AC8	idclk_b	IOB	IO_L32N_8	INPUT	HSTL_III_DCI	8
K5	iderr_a	IOB	IO_L6P_10	INPUT	HSTL_III_DCI	10
AE9	iderr_b	IOB	IO_L31N_8	INPUT	HSTL_III_DCI	8
K4	idfrm_a	IOB	IO_L6N_10	INPUT	HSTL_III_DCI	10
AD8	idfrm_b	IOB	IO_L32P_8	INPUT	HSTL_III_DCI	8
L6	inoise_a	IOB	IO_L5N_10	INPUT	HSTL_III_DCI	10
AF9	inoise_b	IOB	IO_L31P_8	INPUT	HSTL_III_DCI	8
J2	itick_a	IOB	IO_L4P_10	INPUT	HSTL_III_DCI	10
AE6	itick_b	IOB	IO_L30P_8	INPUT	HSTL_III_DCI	8
L7	ivalid_a	IOB	IO_L5P_10	INPUT	HSTL_III_DCI	10
AD6	ivalid_b	IOB	IO_L30N_8	INPUT	HSTL_III_DCI	8
AA13	MCB_ADDR<0>	IOB	IO_L4N_D8_VREF_LC_2	INPUT	LVCMOS25	2
AB13	MCB_ADDR<1>	IOB	IO_L4P_D9_LC_2	INPUT	LVCMOS25	2
AA15	MCB_ADDR<2>	IOB	IO_L3N_D10_LC_2	INPUT	LVCMOS25	2
AA16	MCB_ADDR<3>	IOB	IO_L3P_D11_LC_2	INPUT	LVCMOS25	2
AC11	MCB_ADDR<4>	IOB	IO_L2N_D12_LC_2	INPUT	LVCMOS25	2
AC12	MCB_ADDR<5>	IOB	IO_L2P_D13_LC_2	INPUT	LVCMOS25	2
AB14	MCB_ADDR<6>	IOB	IO_L1N_D14_CC_LC_2	INPUT	LVCMOS25	2
AA14	MCB_ADDR<7>	IOB	IO_L1P_D15_CC_LC_2	INPUT	LVCMOS25	2
AE14	MCB_CLK	IOB	IO_L5P_GC_LC_4	INPUT	LVCMOS25	4

AE12	MCB_CS_N	IOB	IO_L1N_GC_LC_4	INPUT	LVCMOS25	4
D12	MCB_DATA<0>	IOB	IO_L8N_D16_CC_LC_1	BIDIR	LVCMOS25	1
E13	MCB_DATA<1>	IOB	IO_L8P_D17_CC_LC_1	BIDIR	LVCMOS25	1
F15	MCB_DATA<10>	IOB	IO_L3N_D26_LC_1	BIDIR	LVCMOS25	1
F16	MCB_DATA<11>	IOB	IO_L3P_D27_LC_1	BIDIR	LVCMOS25	1
F11	MCB_DATA<12>	IOB	IO_L2N_D28_LC_1	BIDIR	LVCMOS25	1
F12	MCB_DATA<13>	IOB	IO_L2P_D29_LC_1	BIDIR	LVCMOS25	1
F13	MCB_DATA<14>	IOB	IO_L1N_D30_LC_1	BIDIR	LVCMOS25	1
F14	MCB_DATA<15>	IOB	IO_L1P_D31_LC_1	BIDIR	LVCMOS25	1
C16	MCB_DATA<2>	IOB	IO_L7N_D18_LC_1	BIDIR	LVCMOS25	1
D16	MCB_DATA<3>	IOB	IO_L7P_D19_LC_1	BIDIR	LVCMOS25	1
D11	MCB_DATA<4>	IOB	IO_L6N_D20_LC_1	BIDIR	LVCMOS25	1
C11	MCB_DATA<5>	IOB	IO_L6P_D21_LC_1	BIDIR	LVCMOS25	1
E14	MCB_DATA<6>	IOB	IO_L5N_D22_LC_1	BIDIR	LVCMOS25	1
D15	MCB_DATA<7>	IOB	IO_L5P_D23_LC_1	BIDIR	LVCMOS25	1
D13	MCB_DATA<8>	IOB	IO_L4N_D24_VREF_LC_1	BIDIR	LVCMOS25	1
D14	MCB_DATA<9>	IOB	IO_L4P_D25_LC_1	BIDIR	LVCMOS25	1
AF12	MCB_RD	IOB	IO_L1P_GC_LC_4	INPUT	LVCMOS25	4
K21	odata_a<0>	IOB	IO_L3N_9	OUTPUT	HSTL_I_DCI	9
K22	odata_a<1>	IOB	IO_L3P_9	OUTPUT	HSTL_I_DCI	9
H23	odata_a<10>	IOB	IO_L30N_5	OUTPUT	HSTL_I_DCI	5
H24	odata_a<11>	IOB	IO_L30P_5	OUTPUT	HSTL_I_DCI	5
E26	odata_a<12>	IOB	IO_L29N_5	OUTPUT	HSTL_I_DCI	5
F26	odata_a<13>	IOB	IO_L29P_5	OUTPUT	HSTL_I_DCI	5
G24	odata_a<14>	IOB	IO_L28P_5	OUTPUT	HSTL_I_DCI	5
E24	odata_a<15>	IOB	IO_L27N_5	OUTPUT	HSTL_I_DCI	5
E25	odata_a<16>	IOB	IO_L27P_5	OUTPUT	HSTL_I_DCI	5
H21	odata_a<17>	IOB	IO_L26N_5	OUTPUT	HSTL_I_DCI	5
H22	odata_a<18>	IOB	IO_L26P_5	OUTPUT	HSTL_I_DCI	5
D25	odata_a<19>	IOB	IO_L25N_CC_LC_5	OUTPUT	HSTL_I_DCI	5
J22	odata_a<2>	IOB	IO_L2N_9	OUTPUT	HSTL_I_DCI	9
D26	odata_a<20>	IOB	IO_L25P_CC_LC_5	OUTPUT	HSTL_I_DCI	5
F23	odata_a<21>	IOB	IO_L24N_CC_LC_5	OUTPUT	HSTL_I_DCI	5
F24	odata_a<22>	IOB	IO_L24P_CC_LC_5	OUTPUT	HSTL_I_DCI	5

G20	odata_a<23>	IOB	IO_L22N_5	OUTPUT	HSTL_I_DCI	5
H20	odata_a<24>	IOB	IO_L22P_5	OUTPUT	HSTL_I_DCI	5
C23	odata_a<25>	IOB	IO_L21N_5	OUTPUT	HSTL_I_DCI	5
D23	odata_a<26>	IOB	IO_L21P_5	OUTPUT	HSTL_I_DCI	5
C26	odata_a<27>	IOB	IO_L20P_5	OUTPUT	HSTL_I_DCI	5
E20	odata_a<28>	IOB	IO_L19N_5	OUTPUT	HSTL_I_DCI	5
F20	odata_a<29>	IOB	IO_L19P_5	OUTPUT	HSTL_I_DCI	5
J23	odata_a<3>	IOB	IO_L2P_9	OUTPUT	HSTL_I_DCI	9
E22	odata_a<30>	IOB	IO_L18N_5	OUTPUT	HSTL_I_DCI	5
E23	odata_a<31>	IOB	IO_L18P_5	OUTPUT	HSTL_I_DCI	5
F19	odata_a<32>	IOB	IO_L17N_5	OUTPUT	HSTL_I_DCI	5
G19	odata_a<33>	IOB	IO_L17P_5	OUTPUT	HSTL_I_DCI	5
C24	odata_a<34>	IOB	IO_L16N_5	OUTPUT	HSTL_I_DCI	5
D24	odata_a<35>	IOB	IO_L16P_5	OUTPUT	HSTL_I_DCI	5
A21	odata_a<36>	IOB	IO_L15N_5	OUTPUT	HSTL_I_DCI	5
A22	odata_a<37>	IOB	IO_L15P_5	OUTPUT	HSTL_I_DCI	5
C22	odata_a<38>	IOB	IO_L14N_5	OUTPUT	HSTL_I_DCI	5
D22	odata_a<39>	IOB	IO_L14P_5	OUTPUT	HSTL_I_DCI	5
J20	odata_a<4>	IOB	IO_L1N_9	OUTPUT	HSTL_I_DCI	9
A19	odata_a<40>	IOB	IO_L13N_5	OUTPUT	HSTL_I_DCI	5
A20	odata_a<41>	IOB	IO_L13P_5	OUTPUT	HSTL_I_DCI	5
E21	odata_a<42>	IOB	IO_L12P_5	OUTPUT	HSTL_I_DCI	5
E18	odata_a<43>	IOB	IO_L11N_5	OUTPUT	HSTL_I_DCI	5
F18	odata_a<44>	IOB	IO_L11P_5	OUTPUT	HSTL_I_DCI	5
B23	odata_a<45>	IOB	IO_L10N_5	OUTPUT	HSTL_I_DCI	5
B24	odata_a<46>	IOB	IO_L10P_5	OUTPUT	HSTL_I_DCI	5
G17	odata_a<47>	IOB	IO_L9N_CC_LC_5	OUTPUT	HSTL_I_DCI	5
G18	odata_a<48>	IOB	IO_L9P_CC_LC_5	OUTPUT	HSTL_I_DCI	5
A23	odata_a<49>	IOB	IO_L8N_CC_LC_5	OUTPUT	HSTL_I_DCI	5
J21	odata_a<5>	IOB	IO_L1P_9	OUTPUT	HSTL_I_DCI	9
A24	odata_a<50>	IOB	IO_L8P_CC_LC_5	OUTPUT	HSTL_I_DCI	5
D18	odata_a<51>	IOB	IO_L7N_5	OUTPUT	HSTL_I_DCI	5
C19	odata_a<52>	IOB	IO_L7P_5	OUTPUT	HSTL_I_DCI	5
B21	odata_a<53>	IOB	IO_L6N_5	OUTPUT	HSTL_I_DCI	5

C21	odata_a<54>	IOB	IO_L6P_5	OUTPUT	HSTL_I_DCI	5
F17	odata_a<55>	IOB	IO_L5N_5	OUTPUT	HSTL_I_DCI	5
E17	odata_a<56>	IOB	IO_L5P_5	OUTPUT	HSTL_I_DCI	5
D20	odata_a<57>	IOB	IO_L4P_5	OUTPUT	HSTL_I_DCI	5
A18	odata_a<58>	IOB	IO_L3N_5	OUTPUT	HSTL_I_DCI	5
B18	odata_a<59>	IOB	IO_L3P_5	OUTPUT	HSTL_I_DCI	5
H25	odata_a<6>	IOB	IO_L32N_5	OUTPUT	HSTL_I_DCI	5
B20	odata_a<60>	IOB	IO_L2N_5	OUTPUT	HSTL_I_DCI	5
C20	odata_a<61>	IOB	IO_L2P_5	OUTPUT	HSTL_I_DCI	5
D17	odata_a<62>	IOB	IO_L1N_5	OUTPUT	HSTL_I_DCI	5
C17	odata_a<63>	IOB	IO_L1P_5	OUTPUT	HSTL_I_DCI	5
H26	odata_a<7>	IOB	IO_L32P_5	OUTPUT	HSTL_I_DCI	5
G25	odata_a<8>	IOB	IO_L31N_5	OUTPUT	HSTL_I_DCI	5
G26	odata_a<9>	IOB	IO_L31P_5	OUTPUT	HSTL_I_DCI	5
AB18	odata_b<0>	IOB	IO_L29N_SM4_7	OUTPUT	LVCMOS15	7
AC18	odata_b<1>	IOB	IO_L29P_SM4_7	OUTPUT	LVCMOS15	7
AC21	odata_b<10>	IOB	IO_L24P_CC_LC_7	OUTPUT	LVCMOS15	7
AE24	odata_b<11>	IOB	IO_L22N_7	OUTPUT	LVCMOS15	7
AF24	odata_b<12>	IOB	IO_L22P_7	OUTPUT	LVCMOS15	7
Y18	odata_b<13>	IOB	IO_L21N_7	OUTPUT	LVCMOS15	7
AA18	odata_b<14>	IOB	IO_L21P_7	OUTPUT	LVCMOS15	7
Y20	odata_b<15>	IOB	IO_L20P_7	OUTPUT	LVCMOS15	7
AE23	odata_b<16>	IOB	IO_L19N_7	OUTPUT	LVCMOS15	7
AF23	odata_b<17>	IOB	IO_L19P_7	OUTPUT	LVCMOS15	7
W19	odata_b<18>	IOB	IO_L18N_7	OUTPUT	LVCMOS15	7
Y19	odata_b<19>	IOB	IO_L18P_7	OUTPUT	LVCMOS15	7
AB20	odata_b<2>	IOB	IO_L28P_7	OUTPUT	LVCMOS15	7
AF20	odata_b<20>	IOB	IO_L17N_7	OUTPUT	LVCMOS15	7
AF19	odata_b<21>	IOB	IO_L17P_7	OUTPUT	LVCMOS15	7
AC24	odata_b<22>	IOB	IO_L16N_7	OUTPUT	LVCMOS15	7
AC23	odata_b<23>	IOB	IO_L16P_7	OUTPUT	LVCMOS15	7
AD23	odata_b<24>	IOB	IO_L15N_7	OUTPUT	LVCMOS15	7
AD22	odata_b<25>	IOB	IO_L15P_7	OUTPUT	LVCMOS15	7
AA23	odata_b<26>	IOB	IO_L14N_7	OUTPUT	LVCMOS15	7

AB23	odata_b<27>	IOB	IO_L14P_7	OUTPUT	LVCMOS15	7
AB22	odata_b<28>	IOB	IO_L13N_7	OUTPUT	LVCMOS15	7
AC22	odata_b<29>	IOB	IO_L13P_7	OUTPUT	LVCMOS15	7
AA17	odata_b<3>	IOB	IO_L27N_SM5_7	OUTPUT	LVCMOS15	7
Y22	odata_b<30>	IOB	IO_L12P_7	OUTPUT	LVCMOS15	7
AD26	odata_b<31>	IOB	IO_L11N_7	OUTPUT	LVCMOS15	7
AD25	odata_b<32>	IOB	IO_L11P_7	OUTPUT	LVCMOS15	7
AA26	odata_b<33>	IOB	IO_L10N_7	OUTPUT	LVCMOS15	7
AB26	odata_b<34>	IOB	IO_L10P_7	OUTPUT	LVCMOS15	7
AC26	odata_b<35>	IOB	IO_L9N_CC_LC_7	OUTPUT	LVCMOS15	7
AC25	odata_b<36>	IOB	IO_L9P_CC_LC_7	OUTPUT	LVCMOS15	7
Y24	odata_b<37>	IOB	IO_L8N_CC_LC_7	OUTPUT	LVCMOS15	7
AA24	odata_b<38>	IOB	IO_L8P_CC_LC_7	OUTPUT	LVCMOS15	7
AB25	odata_b<39>	IOB	IO_L7N_7	OUTPUT	LVCMOS15	7
Y17	odata_b<4>	IOB	IO_L27P_SM5_7	OUTPUT	LVCMOS15	7
AB24	odata_b<40>	IOB	IO_L7P_7	OUTPUT	LVCMOS15	7
Y26	odata_b<41>	IOB	IO_L6N_7	OUTPUT	LVCMOS15	7
Y25	odata_b<42>	IOB	IO_L6P_7	OUTPUT	LVCMOS15	7
V20	odata_b<43>	IOB	IO_L5N_7	OUTPUT	LVCMOS15	7
W20	odata_b<44>	IOB	IO_L5P_7	OUTPUT	LVCMOS15	7
W23	odata_b<45>	IOB	IO_L4P_7	OUTPUT	LVCMOS15	7
W22	odata_b<46>	IOB	IO_L3N_7	OUTPUT	LVCMOS15	7
W21	odata_b<47>	IOB	IO_L3P_7	OUTPUT	LVCMOS15	7
W26	odata_b<48>	IOB	IO_L2N_7	OUTPUT	LVCMOS15	7
W25	odata_b<49>	IOB	IO_L2P_7	OUTPUT	LVCMOS15	7
AA20	odata_b<5>	IOB	IO_L26N_SM6_7	OUTPUT	LVCMOS15	7
V22	odata_b<50>	IOB	IO_L1N_7	OUTPUT	LVCMOS15	7
V21	odata_b<51>	IOB	IO_L1P_7	OUTPUT	LVCMOS15	7
V25	odata_b<52>	IOB	IO_L32N_9	OUTPUT	LVCMOS15	9
V26	odata_b<53>	IOB	IO_L32P_9	OUTPUT	LVCMOS15	9
T19	odata_b<54>	IOB	IO_L31N_9	OUTPUT	LVCMOS15	9
U20	odata_b<55>	IOB	IO_L31P_9	OUTPUT	LVCMOS15	9
T20	odata_b<56>	IOB	IO_L30N_9	OUTPUT	LVCMOS15	9
T21	odata_b<57>	IOB	IO_L30P_9	OUTPUT	LVCMOS15	9

U21	odata_b<58>	IOB	IO_L29N_9	OUTPUT	LVCMOS15	9
U22	odata_b<59>	IOB	IO_L29P_9	OUTPUT	LVCMOS15	9
AA19	odata_b<6>	IOB	IO_L26P_SM6_7	OUTPUT	LVCMOS15	7
U25	odata_b<60>	IOB	IO_L28P_9	OUTPUT	LVCMOS15	9
V23	odata_b<61>	IOB	IO_L27N_9	OUTPUT	LVCMOS15	9
U23	odata_b<62>	IOB	IO_L27P_9	OUTPUT	LVCMOS15	9
U26	odata_b<63>	IOB	IO_L26N_9	OUTPUT	LVCMOS15	9
AC19	odata_b<7>	IOB	IO_L25N_CC_SM7_LC_7	OUTPUT	LVCMOS15	7
AD19	odata_b<8>	IOB	IO_L25P_CC_SM7_LC_7	OUTPUT	LVCMOS15	7
AB21	odata_b<9>	IOB	IO_L24N_CC_LC_7	OUTPUT	LVCMOS15	7
K24	odclk_a	IOB	IO_L7P_9	OUTPUT	HSTL_I_DCI	9
AD21	odclk_b	IOB	IO_L32N_SM1_7	OUTPUT	LVCMOS15	7
L21	oderr_a	IOB	IO_L6P_9	OUTPUT	HSTL_I_DCI	9
AE18	oderr_b	IOB	IO_L31N_SM2_7	OUTPUT	LVCMOS15	7
L20	odfrm_a	IOB	IO_L6N_9	OUTPUT	HSTL_I_DCI	9
AE21	odfrm_b	IOB	IO_L32P_SM1_7	OUTPUT	LVCMOS15	7
K20	onoise_a	IOB	IO_L5N_9	OUTPUT	HSTL_I_DCI	9
AF18	onoise_b	IOB	IO_L31P_SM2_7	OUTPUT	LVCMOS15	7
J26	otick_a	IOB	IO_L4P_9	OUTPUT	HSTL_I_DCI	9
AF21	otick_b	IOB	IO_L30P_SM3_7	OUTPUT	LVCMOS15	7
L19	ovalid_a	IOB	IO_L5P_9	OUTPUT	HSTL_I_DCI	9
AF22	ovalid_b	IOB	IO_L30N_SM3_7	OUTPUT	LVCMOS15	7
A10	reset_N	IOB	IO_L6P_GC_LC_3	INPUT	LVCMOS25	3
C14	sclk_N	LOWCAPIOB	IO_L3N_GC_LC_3	INPUT	LVDS_25	3
C15	sclk_P	LOWCAPIOB	IO_L3P_GC_LC_3	INPUT	LVDS_25	3
B14	stick_N	IOB	IO_L1N_GC_CC_LC_3	INPUT	LVDS_25	3
B15	stick_P	IOB	IO_L1P_GC_CC_LC_3	INPUT	LVDS_25	3
AF11	test<0>	IOB	IO_L4P_GC_LC_4	OUTPUT	LVCMOS25	4
AC17	test<1>	IOB	IO_L3N_GC_LC_4	OUTPUT	LVCMOS25	4
AB17	test<2>	IOB	IO_L3P_GC_LC_4	OUTPUT	LVCMOS25	4
AB10	test<3>	IOB	IO_L2N_GC_LC_4	OUTPUT	LVCMOS25	4

Table 8-1 Pinout by Signal Name

8.2 Pinouts by pin number

Note: UNUSED and NC pins have been removed.

```
Release 10.1.03 - par K.39 (nt)
Copyright (c) 1995-2008 Xilinx, Inc.  All rights reserved.

Thu Mar 25 09:01:15 2010

INPUT FILE:       wbc_top_map.ncd
OUTPUT FILE:      wbc_top_pad.txt
PART TYPE:        xc4vsx35
SPEED GRADE:      -10
PACKAGE:          ff668
```

Pin	Signal Name	Pin Usage	Pin Name	Direction	IO Standard	IO Bank
A10	reset_N	IOB	IO_L6P_GC_LC_3	INPUT	LVCMOS25*	3
A13			GND			
A14			GND			
A18	odata_a<58>	IOB	IO_L3N_5	OUTPUT	HSTL_I_DCI	5
A19	odata_a<40>	IOB	IO_L13N_5	OUTPUT	HSTL_I_DCI	5
A2			GND			
A20	odata_a<41>	IOB	IO_L13P_5	OUTPUT	HSTL_I_DCI	5
A21	odata_a<36>	IOB	IO_L15N_5	OUTPUT	HSTL_I_DCI	5
A22	odata_a<37>	IOB	IO_L15P_5	OUTPUT	HSTL_I_DCI	5
A23	odata_a<49>	IOB	IO_L8N_CC_LC_5	OUTPUT	HSTL_I_DCI	5
A24	odata_a<50>	IOB	IO_L8P_CC_LC_5	OUTPUT	HSTL_I_DCI	5
A25			GND			
A3	idata_a<39>	IOB	IO_L14P_6	INPUT	HSTL_III_DCI	6
A4	idata_a<37>	IOB	IO_L15P_6	INPUT	HSTL_III_DCI	6
A5	idata_a<53>	IOB	IO_L6N_6	INPUT	HSTL_III_DCI	6

A6	idata_a<54>	IOB	IO_L6P_6	INPUT	HSTL_III_DCI	6
A7	idata_a<58>	IOB	IO_L3N_6	INPUT	HSTL_III_DCI	6
A8	idata_a<59>	IOB	IO_L3P_6	INPUT	HSTL_III_DCI	6
A9	idata_a<41>	IOB	IO_L13P_6	INPUT	HSTL_III_DCI	6
AA1	idata_b<33>	IOB	IO_L10N_8	INPUT	HSTL_III_DCI	8
AA10	idata_b<3>	IOB	IO_L27N_8	INPUT	HSTL_III_DCI	8
AA13	MCB_ADDR<0>	IOB	IO_L4N_D8_VREF_LC_2	INPUT	LVCMOS25*	2
AA14	MCB_ADDR<7>	IOB	IO_L1P_D15_CC_LC_2	INPUT	LVCMOS25*	2
AA15	MCB_ADDR<2>	IOB	IO_L3N_D10_LC_2	INPUT	LVCMOS25*	2
AA16	MCB_ADDR<3>	IOB	IO_L3P_D11_LC_2	INPUT	LVCMOS25*	2
AA17	odata_b<3>	IOB	IO_L27N_SM5_7	OUTPUT	LVCMOS15	7
AA18	odata_b<14>	IOB	IO_L21P_7	OUTPUT	LVCMOS15	7
AA19	odata_b<6>	IOB	IO_L26P_SM6_7	OUTPUT	LVCMOS15	7
AA2			VCCO_8			8
AA20	odata_b<5>	IOB	IO_L26N_SM6_7	OUTPUT	LVCMOS15	7
AA21			GND			
AA22			VCCO_7			7
AA23	odata_b<26>	IOB	IO_L14N_7	OUTPUT	LVCMOS15	7
AA24	odata_b<38>	IOB	IO_L8P_CC_LC_7	OUTPUT	LVCMOS15	7
AA25			VCCO_7			7
AA26	odata_b<33>	IOB	IO_L10N_7	OUTPUT	LVCMOS15	7
AA3	idata_b<39>	IOB	IO_L7N_8	INPUT	HSTL_III_DCI	8
AA4	idata_b<40>	IOB	IO_L7P_8	INPUT	HSTL_III_DCI	8
AA5			VCCO_8			8
AA6			GND			
AA7	idata_b<15>	IOB	IO_L20P_8	INPUT	HSTL_III_DCI	8
AA8	idata_b<6>	IOB	IO_L26P_8	INPUT	HSTL_III_DCI	8
AA9	idata_b<14>	IOB	IO_L21P_8	INPUT	HSTL_III_DCI	8
AB1	idata_b<34>	IOB	IO_L10P_8	INPUT	HSTL_III_DCI	8
AB10	test<3>	IOB	IO_L2N_GC_LC_4	OUTPUT	LVCMOS25*	4
AB11			VCCO_2			2
AB12			GND			
AB13	MCB_ADDR<1>	IOB	IO_L4P_D9_LC_2	INPUT	LVCMOS25*	2
AB14	MCB_ADDR<6>	IOB	IO_L1N_D14_CC_LC_2	INPUT	LVCMOS25*	2

AB15			GND				
AB16			VCCO_2				2
AB17	test<2>	IOB	IO_L3P_GC_LC_4	OUTPUT	LVCMOS25*	4
AB18	odata_b<0>	IOB	IO_L29N_SM4_7	OUTPUT	LVCMOS15	7
AB19			VCCO_7				7
AB20	odata_b<2>	IOB	IO_L28P_7	OUTPUT	LVCMOS15	7
AB21	odata_b<9>	IOB	IO_L24N_CC_LC_7	OUTPUT	LVCMOS15	7
AB22	odata_b<28>	IOB	IO_L13N_7	OUTPUT	LVCMOS15	7
AB23	odata_b<27>	IOB	IO_L14P_7	OUTPUT	LVCMOS15	7
AB24	odata_b<40>	IOB	IO_L7P_7	OUTPUT	LVCMOS15	7
AB25	odata_b<39>	IOB	IO_L7N_7	OUTPUT	LVCMOS15	7
AB26	odata_b<34>	IOB	IO_L10P_7	OUTPUT	LVCMOS15	7
AB3	idata_b<30>	IOB	IO_L12P_8	INPUT	HSTL_III_DCI	8
AB4	idata_b<31>	IOB	IO_L11N_8	INPUT	HSTL_III_DCI	8
AB5	idata_b<28>	IOB	IO_L13N_8	INPUT	HSTL_III_DCI	8
AB6	idata_b<9>	IOB	IO_L24N_CC_LC_8	INPUT	HSTL_III_DCI	8
AB8			VCCO_8				8
AB9	idata_b<0>	IOB	IO_L29N_8	INPUT	HSTL_III_DCI	8
AC1	idata_b<26>	IOB	IO_L14N_8	INPUT	HSTL_III_DCI	8
AC11	MCB_ADDR<4>	IOB	IO_L2N_D12_LC_2	INPUT	LVCMOS25*	2
AC12	MCB_ADDR<5>	IOB	IO_L2P_D13_LC_2	INPUT	LVCMOS25*	2
AC17	test<1>	IOB	IO_L3N_GC_LC_4	OUTPUT	LVCMOS25*	4
AC18	odata_b<1>	IOB	IO_L29P_SM4_7	OUTPUT	LVCMOS15	7
AC19	odata_b<7>	IOB	IO_L25N_CC_SM7_LC_7	OUTPUT	LVCMOS15	7
AC2	idata_b<27>	IOB	IO_L14P_8	INPUT	HSTL_III_DCI	8
AC21	odata_b<10>	IOB	IO_L24P_CC_LC_7	OUTPUT	LVCMOS15	7
AC22	odata_b<29>	IOB	IO_L13P_7	OUTPUT	LVCMOS15	7
AC23	odata_b<23>	IOB	IO_L16P_7	OUTPUT	LVCMOS15	7
AC24	odata_b<22>	IOB	IO_L16N_7	OUTPUT	LVCMOS15	7
AC25	odata_b<36>	IOB	IO_L9P_CC_LC_7	OUTPUT	LVCMOS15	7
AC26	odata_b<35>	IOB	IO_L9N_CC_LC_7	OUTPUT	LVCMOS15	7
AC3	idata_b<18>	IOB	IO_L18N_8	INPUT	HSTL_III_DCI	8
AC4	idata_b<32>	IOB	IO_L11P_8	INPUT	HSTL_III_DCI	8
AC5	idata_b<29>	IOB	IO_L13P_8	INPUT	HSTL_III_DCI	8

AC6	idata_b<10>	IOB	IO_L24P_CC_LC_8	INPUT	HSTL_III_DCI	8
AC7	idata_b<2>	IOB	IO_L28P_8	INPUT	HSTL_III_DCI	8
AC8	idclk_b	IOB	IO_L32N_8	INPUT	HSTL_III_DCI	8
AC9	idata_b<1>	IOB	IO_L29P_8	INPUT	HSTL_III_DCI	8
AD1	idata_b<22>	IOB	IO_L16N_8	INPUT	HSTL_III_DCI	8
AD15			VCCO_4			4
AD18			GND			
AD19	odata_b<8>	IOB	IO_L25P_CC_SM7_LC_7	OUTPUT	LVCMOS15	7
AD2	idata_b<23>	IOB	IO_L16P_8	INPUT	HSTL_III_DCI	8
AD21	odclk_b	IOB	IO_L32N_SM1_7	OUTPUT	LVCMOS15	7
AD22	odata_b<25>	IOB	IO_L15P_7	OUTPUT	LVCMOS15	7
AD23	odata_b<24>	IOB	IO_L15N_7	OUTPUT	LVCMOS15	7
AD24			GND			
AD25	odata_b<32>	IOB	IO_L11P_7	OUTPUT	LVCMOS15	7
AD26	odata_b<31>	IOB	IO_L11N_7	OUTPUT	LVCMOS15	7
AD3	idata_b<19>	IOB	IO_L18P_8	INPUT	HSTL_III_DCI	8
AD4	idata_b<11>	IOB	IO_L22N_8	INPUT	HSTL_III_DCI	8
AD5	idata_b<12>	IOB	IO_L22P_8	INPUT	HSTL_III_DCI	8
AD6	ivalid_b	IOB	IO_L30N_8	INPUT	HSTL_III_DCI	8
AD8	idfrm_b	IOB	IO_L32P_8	INPUT	HSTL_III_DCI	8
AD9			GND			
AE1			GND			
AE11			VCCO_4			4
AE12	MCB_CS_N	IOB	IO_L1N_GC_LC_4	INPUT	LVCMOS25*	4
AE14	MCB_CLK	IOB	IO_L5P_GC_LC_4	INPUT	LVCMOS25*	4
AE15			VREFN_SM			
AE16			VREFP_SM			
AE17			AVSS_SM			
AE18	oderr_b	IOB	IO_L31N_SM2_7	OUTPUT	LVCMOS15	7
AE19			VCCO_7			7
AE2			GND			
AE21	odfrm_b	IOB	IO_L32P_SM1_7	OUTPUT	LVCMOS15	7
AE22			VCCO_7			7
AE23	odata_b<16>	IOB	IO_L19N_7	OUTPUT	LVCMOS15	7

AE24	odata_b<11>	IOB	IO_L22N_7	OUTPUT	LVCMOS15	7
AE25			GND			
AE26			GND			
AE3	idata_b<24>	IOB	IO_L15N_8	INPUT	HSTL_III_DCI	8
AE4	idata_b<20>	IOB	IO_L17N_8	INPUT	HSTL_III_DCI	8
AE5			VCCO_8			8
AE6	itick_b	IOB	IO_L30P_8	INPUT	HSTL_III_DCI	8
AE8			VCCO_8			8
AE9	iderr_b	IOB	IO_L31N_8	INPUT	HSTL_III_DCI	8
AF11	test<0>	IOB	IO_L4P_GC_LC_4	OUTPUT	LVCMOS25*	4
AF12	MCB_RD	IOB	IO_L1P_GC_LC_4	INPUT	LVCMOS25*	4
AF13			GND			
AF14			GND			
AF17			AVDD_SM			
AF18	onoise_b	IOB	IO_L31P_SM2_7	OUTPUT	LVCMOS15	7
AF19	odata_b<21>	IOB	IO_L17P_7	OUTPUT	LVCMOS15	7
AF2			GND			
AF20	odata_b<20>	IOB	IO_L17N_7	OUTPUT	LVCMOS15	7
AF21	otick_b	IOB	IO_L30P_SM3_7	OUTPUT	LVCMOS15	7
AF22	ovalid_b	IOB	IO_L30N_SM3_7	OUTPUT	LVCMOS15	7
AF23	odata_b<17>	IOB	IO_L19P_7	OUTPUT	LVCMOS15	7
AF24	odata_b<12>	IOB	IO_L22P_7	OUTPUT	LVCMOS15	7
AF25			GND			
AF3	idata_b<25>	IOB	IO_L15P_8	INPUT	HSTL_III_DCI	8
AF4	idata_b<21>	IOB	IO_L17P_8	INPUT	HSTL_III_DCI	8
AF5	idata_b<16>	IOB	IO_L19N_8	INPUT	HSTL_III_DCI	8
AF6	idata_b<17>	IOB	IO_L19P_8	INPUT	HSTL_III_DCI	8
AF7	idata_b<7>	IOB	IO_L25N_CC_LC_8	INPUT	HSTL_III_DCI	8
AF8	idata_b<8>	IOB	IO_L25P_CC_LC_8	INPUT	HSTL_III_DCI	8
AF9	inoise_b	IOB	IO_L31P_8	INPUT	HSTL_III_DCI	8
B1			GND			
B11			VCCO_3			3
B14	stick_N	IOB	IO_L1N_GC_CC_LC_3	INPUT	LVDS_25	3
B15	stick_P	IOB	IO_L1P_GC_CC_LC_3	INPUT	LVDS_25	3

B16			VCCO_3			3
B18	odata_a<59>	IOB	IO_L3P_5	OUTPUT	HSTL_I_DCI	5
B19			VCCO_5			5
B2			GND			
B20	odata_a<60>	IOB	IO_L2N_5	OUTPUT	HSTL_I_DCI	5
B21	odata_a<53>	IOB	IO_L6N_5	OUTPUT	HSTL_I_DCI	5
B22			VCCO_5			5
B23	odata_a<45>	IOB	IO_L10N_5	OUTPUT	HSTL_I_DCI	5
B24	odata_a<46>	IOB	IO_L10P_5	OUTPUT	HSTL_I_DCI	5
B25			GND			
B26			GND			
B3	idata_a<38>	IOB	IO_L14N_6	INPUT	HSTL_III_DCI	6
B4	idata_a<36>	IOB	IO_L15N_6	INPUT	HSTL_III_DCI	6
B5			VCCO_6			6
B6	idata_a<50>	IOB	IO_L8P_CC_LC_6	INPUT	HSTL_III_DCI	6
B7	idata_a<44>	IOB	IO_L11P_6	INPUT	HSTL_III_DCI	6
B8			VCCO_6			6
B9	idata_a<40>	IOB	IO_L13N_6	INPUT	HSTL_III_DCI	6
C10	idata_a<62>	IOB	IO_L1N_6	INPUT	HSTL_III_DCI	6
C11	MCB_DATA<5>	IOB	IO_L6P_D21_LC_1	BIDIR	LVCMOS25*	1
C14	sclk_N	LOWCAPIOB	IO_L3N_GC_LC_3	INPUT	LVDS_25	3
C15	sclk_P	LOWCAPIOB	IO_L3P_GC_LC_3	INPUT	LVDS_25	3
C16	MCB_DATA<2>	IOB	IO_L7N_D18_LC_1	BIDIR	LVCMOS25*	1
C17	odata_a<63>	IOB	IO_L1P_5	OUTPUT	HSTL_I_DCI	5
C18			GND			
C19	odata_a<52>	IOB	IO_L7P_5	OUTPUT	HSTL_I_DCI	5
C2	idata_a<27>	IOB	IO_L20P_6	INPUT	HSTL_III_DCI	6
C20	odata_a<61>	IOB	IO_L2P_5	OUTPUT	HSTL_I_DCI	5
C21	odata_a<54>	IOB	IO_L6P_5	OUTPUT	HSTL_I_DCI	5
C22	odata_a<38>	IOB	IO_L14N_5	OUTPUT	HSTL_I_DCI	5
C23	odata_a<25>	IOB	IO_L21N_5	OUTPUT	HSTL_I_DCI	5
C24	odata_a<34>	IOB	IO_L16N_5	OUTPUT	HSTL_I_DCI	5
C26	odata_a<27>	IOB	IO_L20P_5	OUTPUT	HSTL_I_DCI	5
C3			GND			

C4	idata_a<35>	IOB	IO_L16P_6	INPUT	HSTL_III_DCI	6
C5	idata_a<42>	IOB	IO_L12P_6	INPUT	HSTL_III_DCI	6
C6	idata_a<49>	IOB	IO_L8N_CC_LC_6	INPUT	HSTL_III_DCI	6
C7	idata_a<43>	IOB	IO_L11N_6	INPUT	HSTL_III_DCI	6
C8	idata_a<60>	IOB	IO_L2N_6	INPUT	HSTL_III_DCI	6
C9			GND			
D1	idata_a<19>	IOB	IO_L25N_CC_LC_6	INPUT	HSTL_III_DCI	6
D10	idata_a<63>	IOB	IO_L1P_6	INPUT	HSTL_III_DCI	6
D11	MCB_DATA<4>	IOB	IO_L6N_D20_LC_1	BIDIR	LVCMOS25*	1
D12	MCB_DATA<0>	IOB	IO_L8N_D16_CC_LC_1	BIDIR	LVCMOS25*	1
D13	MCB_DATA<8>	IOB	IO_L4N_D24_VREF_LC_1	BIDIR	LVCMOS25*	1
D14	MCB_DATA<9>	IOB	IO_L4P_D25_LC_1	BIDIR	LVCMOS25*	1
D15	MCB_DATA<7>	IOB	IO_L5P_D23_LC_1	BIDIR	LVCMOS25*	1
D16	MCB_DATA<3>	IOB	IO_L7P_D19_LC_1	BIDIR	LVCMOS25*	1
D17	odata_a<62>	IOB	IO_L1N_5	OUTPUT	HSTL_I_DCI	5
D18	odata_a<51>	IOB	IO_L7N_5	OUTPUT	HSTL_I_DCI	5
D2	idata_a<20>	IOB	IO_L25P_CC_LC_6	INPUT	HSTL_III_DCI	6
D20	odata_a<57>	IOB	IO_L4P_5	OUTPUT	HSTL_I_DCI	5
D22	odata_a<39>	IOB	IO_L14P_5	OUTPUT	HSTL_I_DCI	5
D23	odata_a<26>	IOB	IO_L21P_5	OUTPUT	HSTL_I_DCI	5
D24	odata_a<35>	IOB	IO_L16P_5	OUTPUT	HSTL_I_DCI	5
D25	odata_a<19>	IOB	IO_L25N_CC_LC_5	OUTPUT	HSTL_I_DCI	5
D26	odata_a<20>	IOB	IO_L25P_CC_LC_5	OUTPUT	HSTL_I_DCI	5
D3	idata_a<24>	IOB	IO_L22P_6	INPUT	HSTL_III_DCI	6
D4	idata_a<34>	IOB	IO_L16N_6	INPUT	HSTL_III_DCI	6
D6	idata_a<32>	IOB	IO_L17N_6	INPUT	HSTL_III_DCI	6
D8	idata_a<57>	IOB	IO_L4P_6	INPUT	HSTL_III_DCI	6
D9	idata_a<61>	IOB	IO_L2P_6	INPUT	HSTL_III_DCI	6
E1	idata_a<18>	IOB	IO_L26P_6	INPUT	HSTL_III_DCI	6
E10	idata_a<55>	IOB	IO_L5N_6	INPUT	HSTL_III_DCI	6
E11			VCCO_1			1
E12			GND			
E13	MCB_DATA<1>	IOB	IO_L8P_D17_CC_LC_1	BIDIR	LVCMOS25*	1
E14	MCB_DATA<6>	IOB	IO_L5N_D22_LC_1	BIDIR	LVCMOS25*	1

E15			GND				
E16			VCCO_1			1	
E17	odata_a<56>	IOB	IO_L5P_5	OUTPUT	HSTL_I_DCI	5	
E18	odata_a<43>	IOB	IO_L11N_5	OUTPUT	HSTL_I_DCI	5	
E19			VCCO_5			5	
E2	idata_a<21>	IOB	IO_L24N_CC_LC_6	INPUT	HSTL_III_DCI	6	
E20	odata_a<28>	IOB	IO_L19N_5	OUTPUT	HSTL_I_DCI	5	
E21	odata_a<42>	IOB	IO_L12P_5	OUTPUT	HSTL_I_DCI	5	
E22	odata_a<30>	IOB	IO_L18N_5	OUTPUT	HSTL_I_DCI	5	
E23	odata_a<31>	IOB	IO_L18P_5	OUTPUT	HSTL_I_DCI	5	
E24	odata_a<15>	IOB	IO_L27N_5	OUTPUT	HSTL_I_DCI	5	
E25	odata_a<16>	IOB	IO_L27P_5	OUTPUT	HSTL_I_DCI	5	
E26	odata_a<12>	IOB	IO_L29N_5	OUTPUT	HSTL_I_DCI	5	
E3	idata_a<22>	IOB	IO_L24P_CC_LC_6	INPUT	HSTL_III_DCI	6	
E4	idata_a<23>	IOB	IO_L22N_6	INPUT	HSTL_III_DCI	6	
E5	idata_a<30>	IOB	IO_L18N_6	INPUT	HSTL_III_DCI	6	
E6	idata_a<31>	IOB	IO_L18P_6	INPUT	HSTL_III_DCI	6	
E7	idata_a<33>	IOB	IO_L17P_6	INPUT	HSTL_III_DCI	6	
E8			VCCO_6			6	
E9	idata_a<52>	IOB	IO_L7P_6	INPUT	HSTL_III_DCI	6	
F1	idata_a<17>	IOB	IO_L26N_6	INPUT	HSTL_III_DCI	6	
F10	idata_a<56>	IOB	IO_L5P_6	INPUT	HSTL_III_DCI	6	
F11	MCB_DATA<12>	IOB	IO_L2N_D28_LC_1	BIDIR	LVCMOS25*	1	
F12	MCB_DATA<13>	IOB	IO_L2P_D29_LC_1	BIDIR	LVCMOS25*	1	
F13	MCB_DATA<14>	IOB	IO_L1N_D30_LC_1	BIDIR	LVCMOS25*	1	
F14	MCB_DATA<15>	IOB	IO_L1P_D31_LC_1	BIDIR	LVCMOS25*	1	
F15	MCB_DATA<10>	IOB	IO_L3N_D26_LC_1	BIDIR	LVCMOS25*	1	
F16	MCB_DATA<11>	IOB	IO_L3P_D27_LC_1	BIDIR	LVCMOS25*	1	
F17	odata_a<55>	IOB	IO_L5N_5	OUTPUT	HSTL_I_DCI	5	
F18	odata_a<44>	IOB	IO_L11P_5	OUTPUT	HSTL_I_DCI	5	
F19	odata_a<32>	IOB	IO_L17N_5	OUTPUT	HSTL_I_DCI	5	
F2			VCCO_6			6	
F20	odata_a<29>	IOB	IO_L19P_5	OUTPUT	HSTL_I_DCI	5	
F21			GND				

F22			VCCO_5			5	
F23	odata_a<21>	IOB	IO_L24N_CC_LC_5	OUTPUT	HSTL_I_DCI	5	
F24	odata_a<22>	IOB	IO_L24P_CC_LC_5	OUTPUT	HSTL_I_DCI	5	
F25			VCCO_5			5	
F26	odata_a<13>	IOB	IO_L29P_5	OUTPUT	HSTL_I_DCI	5	
F3	idata_a<15>	IOB	IO_L27N_6	INPUT	HSTL_III_DCI	6	
F4	idata_a<16>	IOB	IO_L27P_6	INPUT	HSTL_III_DCI	6	
F5			VCCO_6			6	
F6			GND				
F7	idata_a<29>	IOB	IO_L19P_6	INPUT	HSTL_III_DCI	6	
F8	idata_a<46>	IOB	IO_L10P_6	INPUT	HSTL_III_DCI	6	
F9	idata_a<51>	IOB	IO_L7N_6	INPUT	HSTL_III_DCI	6	
G1	idata_a<10>	IOB	IO_L30N_6	INPUT	HSTL_III_DCI	6	
G10	idata_a<48>	IOB	IO_L9P_CC_LC_6	INPUT	HSTL_III_DCI	6	
G11			CS_B_0				
G12			D_IN_0				
G13			TDN_0				
G14			CCLK_0				
G15			INIT_0				
G16			HSWAPEN_0				
G17	odata_a<47>	IOB	IO_L9N_CC_LC_5	OUTPUT	HSTL_I_DCI	5	
G18	odata_a<48>	IOB	IO_L9P_CC_LC_5	OUTPUT	HSTL_I_DCI	5	
G19	odata_a<33>	IOB	IO_L17P_5	OUTPUT	HSTL_I_DCI	5	
G2	idata_a<11>	IOB	IO_L30P_6	INPUT	HSTL_III_DCI	6	
G20	odata_a<23>	IOB	IO_L22N_5	OUTPUT	HSTL_I_DCI	5	
G24	odata_a<14>	IOB	IO_L28P_5	OUTPUT	HSTL_I_DCI	5	
G25	odata_a<8>	IOB	IO_L31N_5	OUTPUT	HSTL_I_DCI	5	
G26	odata_a<9>	IOB	IO_L31P_5	OUTPUT	HSTL_I_DCI	5	
G4	idata_a<14>	IOB	IO_L28P_6	INPUT	HSTL_III_DCI	6	
G7	idata_a<28>	IOB	IO_L19N_6	INPUT	HSTL_III_DCI	6	
G8	idata_a<45>	IOB	IO_L10N_6	INPUT	HSTL_III_DCI	6	
G9	idata_a<47>	IOB	IO_L9N_CC_LC_6	INPUT	HSTL_III_DCI	6	
H1	idata_a<6>	IOB	IO_L32N_6	INPUT	HSTL_III_DCI	6	
H10			VCCO_6			6	

H11			VCCAUX			
H12			RDWR_B_0			
H13			TDP_0			
H14			DONE_0			
H15			PROGRAM_B_0			
H16			VCCAUX			
H17			VCCAUX			
H18			VCCO_5			5
H19			VCCO_5			5
H2	idata_a<7>	IOB	IO_L32P_6	INPUT	HSTL_III_DCI	6
H20	odata_a<24>	IOB	IO_L22P_5	OUTPUT	HSTL_I_DCI	5
H21	odata_a<17>	IOB	IO_L26N_5	OUTPUT	HSTL_I_DCI	5
H22	odata_a<18>	IOB	IO_L26P_5	OUTPUT	HSTL_I_DCI	5
H23	odata_a<10>	IOB	IO_L30N_5	OUTPUT	HSTL_I_DCI	5
H24	odata_a<11>	IOB	IO_L30P_5	OUTPUT	HSTL_I_DCI	5
H25	odata_a<6>	IOB	IO_L32N_5	OUTPUT	HSTL_I_DCI	5
H26	odata_a<7>	IOB	IO_L32P_5	OUTPUT	HSTL_I_DCI	5
H3	idata_a<8>	IOB	IO_L31N_6	INPUT	HSTL_III_DCI	6
H4	idata_a<9>	IOB	IO_L31P_6	INPUT	HSTL_III_DCI	6
H5	idata_a<12>	IOB	IO_L29N_6	INPUT	HSTL_III_DCI	6
H6	idata_a<13>	IOB	IO_L29P_6	INPUT	HSTL_III_DCI	6
H7	idata_a<25>	IOB	IO_L21N_6	INPUT	HSTL_III_DCI	6
H8	idata_a<26>	IOB	IO_L21P_6	INPUT	HSTL_III_DCI	6
H9			VCCO_6			6
J10			VCCINT			
J11			VCCINT			
J12			VCCAUX			
J13			GND			
J14			GND			
J15			VCCO_0			0
J16			VCCINT			
J17			VCCINT			
J19			VCCO_5			5
J2	itick_a	IOB	IO_L4P_10	INPUT	HSTL_III_DCI	10

J20	odata_a<4>	IOB	IO_L1N_9	OUTPUT	HSTL_I_DCI	9
J21	odata_a<5>	IOB	IO_L1P_9	OUTPUT	HSTL_I_DCI	9
J22	odata_a<2>	IOB	IO_L2N_9	OUTPUT	HSTL_I_DCI	9
J23	odata_a<3>	IOB	IO_L2P_9	OUTPUT	HSTL_I_DCI	9
J24			GND			
J26	otick_a	IOB	IO_L4P_9	OUTPUT	HSTL_I_DCI	9
J3			GND			
J4	idata_a<2>	IOB	IO_L2N_10	INPUT	HSTL_III_DCI	10
J5	idata_a<3>	IOB	IO_L2P_10	INPUT	HSTL_III_DCI	10
J6	idata_a<4>	IOB	IO_L1N_10	INPUT	HSTL_III_DCI	10
J7	idata_a<5>	IOB	IO_L1P_10	INPUT	HSTL_III_DCI	10
J8			VCCO_6			6
K10			VCCINT			
K11			GND			
K12			GND			
K13			GND			
K14			GND			
K15			GND			
K16			GND			
K17			VCCINT			
K18			VCCINT			
K19			VCCO_9			9
K20	onoise_a	IOB	IO_L5N_9	OUTPUT	HSTL_I_DCI	9
K21	odata_a<0>	IOB	IO_L3N_9	OUTPUT	HSTL_I_DCI	9
K22	odata_a<1>	IOB	IO_L3P_9	OUTPUT	HSTL_I_DCI	9
K24	odclk_a	IOB	IO_L7P_9	OUTPUT	HSTL_I_DCI	9
K3	idclk_a	IOB	IO_L7P_10	INPUT	HSTL_III_DCI	10
K4	idfrm_a	IOB	IO_L6N_10	INPUT	HSTL_III_DCI	10
K5	iderr_a	IOB	IO_L6P_10	INPUT	HSTL_III_DCI	10
K6	idata_a<0>	IOB	IO_L3N_10	INPUT	HSTL_III_DCI	10
K7	idata_a<1>	IOB	IO_L3P_10	INPUT	HSTL_III_DCI	10
K8			VCCO_10			10
K9			VCCINT			
L10			VCCINT			

L11				VCCINT			
L12				GND			
L13				GND			
L14				GND			
L15				GND			
L16				VCCINT			
L17				VCCINT			
L18				VCCINT			
L19	ovalid_a	IOB		IO_L5P_9	OUTPUT	HSTL_I_DCI	9
L2				VCCO_10			10
L20	odfrm_a	IOB		IO_L6N_9	OUTPUT	HSTL_I_DCI	9
L21	oderr_a	IOB		IO_L6P_9	OUTPUT	HSTL_I_DCI	9
L22				VCCO_9			9
L25				VCCO_9			9
L5				VCCO_10			10
L6	inoise_a	IOB		IO_L5N_10	INPUT	HSTL_III_DCI	10
L7	ivalid_a	IOB		IO_L5P_10	INPUT	HSTL_III_DCI	10
L9				VCCINT			
M10				GND			
M11				GND			
M12				VCCINT			
M13				GND			
M14				GND			
M15				VCCINT			
M16				GND			
M17				GND			
M18				VCCO_9			9
M9				VCCAUX			
N1				VCCO_10			10
N10				GND			
N11				GND			
N12				GND			
N13				GND			
N14				GND			

N15			GND			
N16			GND			
N17			GND			
N18			VCCAUX			
N26			GND			
N6			GND			
N9			VCCAUX			
P1			GND			
P10			GND			
P11			GND			
P12			GND			
P13			GND			
P14			GND			
P15			GND			
P16			GND			
P17			GND			
P18			VCCAUX			
P21			GND			
P26			VCCO_9			9
P9			VCCAUX			
R10			GND			
R11			GND			
R12			VCCINT			
R13			GND			
R14			GND			
R15			VCCINT			
R16			GND			
R17			GND			
R18			VCCAUX			
R9			VCCO_10			10
T10			VCCINT			
T11			VCCINT			
T12			GND			
T13			GND			

T14			GND			
T15			GND			
T16			VCCINT			
T17			VCCINT			
T18			VCCINT			
T19	odata_b<54>	IOB	IO_L31N_9	OUTPUT	LVCMOS15	9
T2			VCCO_10			10
T20	odata_b<56>	IOB	IO_L30N_9	OUTPUT	LVCMOS15	9
T21	odata_b<57>	IOB	IO_L30P_9	OUTPUT	LVCMOS15	9
T22			VCCO_9			9
T25			VCCO_9			9
T3	idata_b<63>	IOB	IO_L26N_10	INPUT	HSTL_III_DCI	10
T5			VCCO_10			10
T6	idata_b<61>	IOB	IO_L27N_10	INPUT	HSTL_III_DCI	10
T7	idata_b<62>	IOB	IO_L27P_10	INPUT	HSTL_III_DCI	10
T8	idata_b<55>	IOB	IO_L31P_10	INPUT	HSTL_III_DCI	10
T9			VCCINT			
U10			VCCINT			
U11			GND			
U12			GND			
U13			GND			
U14			GND			
U15			GND			
U16			GND			
U17			VCCINT			
U18			VCCINT			
U19			VCCO_9			9
U20	odata_b<55>	IOB	IO_L31P_9	OUTPUT	LVCMOS15	9
U21	odata_b<58>	IOB	IO_L29N_9	OUTPUT	LVCMOS15	9
U22	odata_b<59>	IOB	IO_L29P_9	OUTPUT	LVCMOS15	9
U23	odata_b<62>	IOB	IO_L27P_9	OUTPUT	LVCMOS15	9
U25	odata_b<60>	IOB	IO_L28P_9	OUTPUT	LVCMOS15	9
U26	odata_b<63>	IOB	IO_L26N_9	OUTPUT	LVCMOS15	9
U3	idata_b<60>	IOB	IO_L28P_10	INPUT	HSTL_III_DCI	10

U4	idata_b<58>	IOB	IO_L29N_10	INPUT	HSTL_III_DCI	10
U5	idata_b<52>	IOB	IO_L32N_10	INPUT	HSTL_III_DCI	10
U6	idata_b<53>	IOB	IO_L32P_10	INPUT	HSTL_III_DCI	10
U7	idata_b<54>	IOB	IO_L31N_10	INPUT	HSTL_III_DCI	10
U8			VCCO_10			10
U9			VCCINT			
V1	idata_b<56>	IOB	IO_L30N_10	INPUT	HSTL_III_DCI	10
V10			VCCINT			
V11			VCCINT			
V12			VCCO_0			0
V13			GND			
V14			GND			
V15			VCCAUX			
V16			VCCINT			
V17			VCCINT			
V19			VCCO_7			7
V2	idata_b<57>	IOB	IO_L30P_10	INPUT	HSTL_III_DCI	10
V20	odata_b<43>	IOB	IO_L5N_7	OUTPUT	LVCMOS15	7
V21	odata_b<51>	IOB	IO_L1P_7	OUTPUT	LVCMOS15	7
V22	odata_b<50>	IOB	IO_L1N_7	OUTPUT	LVCMOS15	7
V23	odata_b<61>	IOB	IO_L27N_9	OUTPUT	LVCMOS15	9
V24			GND			
V25	odata_b<52>	IOB	IO_L32N_9	OUTPUT	LVCMOS15	9
V26	odata_b<53>	IOB	IO_L32P_9	OUTPUT	LVCMOS15	9
V3			GND			
V4	idata_b<59>	IOB	IO_L29P_10	INPUT	HSTL_III_DCI	10
V5	idata_b<48>	IOB	IO_L2N_8	INPUT	HSTL_III_DCI	8
V6	idata_b<49>	IOB	IO_L2P_8	INPUT	HSTL_III_DCI	8
V7	idata_b<46>	IOB	IO_L3N_8	INPUT	HSTL_III_DCI	8
V8			VCCO_8			8
W1	idata_b<50>	IOB	IO_L1N_8	INPUT	HSTL_III_DCI	8
W10			VCCAUX			
W11			VCCAUX			
W12			TCK_0			

W13			PWRDWN_B_0			
W14			M2_0			
W15			M0_0			
W16			VCCAUX			
W17			VCCO_7			7
W18			VCCO_7			7
W19	odata_b<18>	IOB	IO_L18N_7	OUTPUT	LVCMOS15	7
W2	idata_b<51>	IOB	IO_L1P_8	INPUT	HSTL_III_DCI	8
W20	odata_b<44>	IOB	IO_L5P_7	OUTPUT	LVCMOS15	7
W21	odata_b<47>	IOB	IO_L3P_7	OUTPUT	LVCMOS15	7
W22	odata_b<46>	IOB	IO_L3N_7	OUTPUT	LVCMOS15	7
W23	odata_b<45>	IOB	IO_L4P_7	OUTPUT	LVCMOS15	7
W25	odata_b<49>	IOB	IO_L2P_7	OUTPUT	LVCMOS15	7
W26	odata_b<48>	IOB	IO_L2N_7	OUTPUT	LVCMOS15	7
W4	idata_b<45>	IOB	IO_L4P_8	INPUT	HSTL_III_DCI	8
W5	idata_b<43>	IOB	IO_L5N_8	INPUT	HSTL_III_DCI	8
W6	idata_b<44>	IOB	IO_L5P_8	INPUT	HSTL_III_DCI	8
W7	idata_b<47>	IOB	IO_L3P_8	INPUT	HSTL_III_DCI	8
W8			VCCO_8			8
W9			VCCO_8			8
Y1	idata_b<41>	IOB	IO_L6N_8	INPUT	HSTL_III_DCI	8
Y10	idata_b<4>	IOB	IO_L27P_8	INPUT	HSTL_III_DCI	8
Y11			TMS_0			
Y12			TDI_0			
Y13			TDO_0			
Y14			DOUT_BUSY_0			
Y15			M1_0			
Y16			VBATT_0			
Y17	odata_b<4>	IOB	IO_L27P_SM5_7	OUTPUT	LVCMOS15	7
Y18	odata_b<13>	IOB	IO_L21N_7	OUTPUT	LVCMOS15	7
Y19	odata_b<19>	IOB	IO_L18P_7	OUTPUT	LVCMOS15	7
Y2	idata_b<42>	IOB	IO_L6P_8	INPUT	HSTL_III_DCI	8
Y20	odata_b<15>	IOB	IO_L20P_7	OUTPUT	LVCMOS15	7
Y22	odata_b<30>	IOB	IO_L12P_7	OUTPUT	LVCMOS15	7

Y24	odata_b<37>	IOB	IO_L8N_CC_LC_7	OUTPUT	LVCMOS15	7
Y25	odata_b<42>	IOB	IO_L6P_7	OUTPUT	LVCMOS15	7
Y26	odata_b<41>	IOB	IO_L6N_7	OUTPUT	LVCMOS15	7
Y3	idata_b<37>	IOB	IO_L8N_CC_LC_8	INPUT	HSTL_III_DCI	8
Y4	idata_b<38>	IOB	IO_L8P_CC_LC_8	INPUT	HSTL_III_DCI	8
Y5	idata_b<35>	IOB	IO_L9N_CC_LC_8	INPUT	HSTL_III_DCI	8
Y6	idata_b<36>	IOB	IO_L9P_CC_LC_8	INPUT	HSTL_III_DCI	8
Y8	idata_b<5>	IOB	IO_L26N_8	INPUT	HSTL_III_DCI	8
Y9	idata_b<13>	IOB	IO_L21N_8	INPUT	HSTL_III_DCI	8

Table 8-2 Pinout by Pin Number

8.3 Xilinx XC4VSX35-10FF668-CS2 Package Drawing

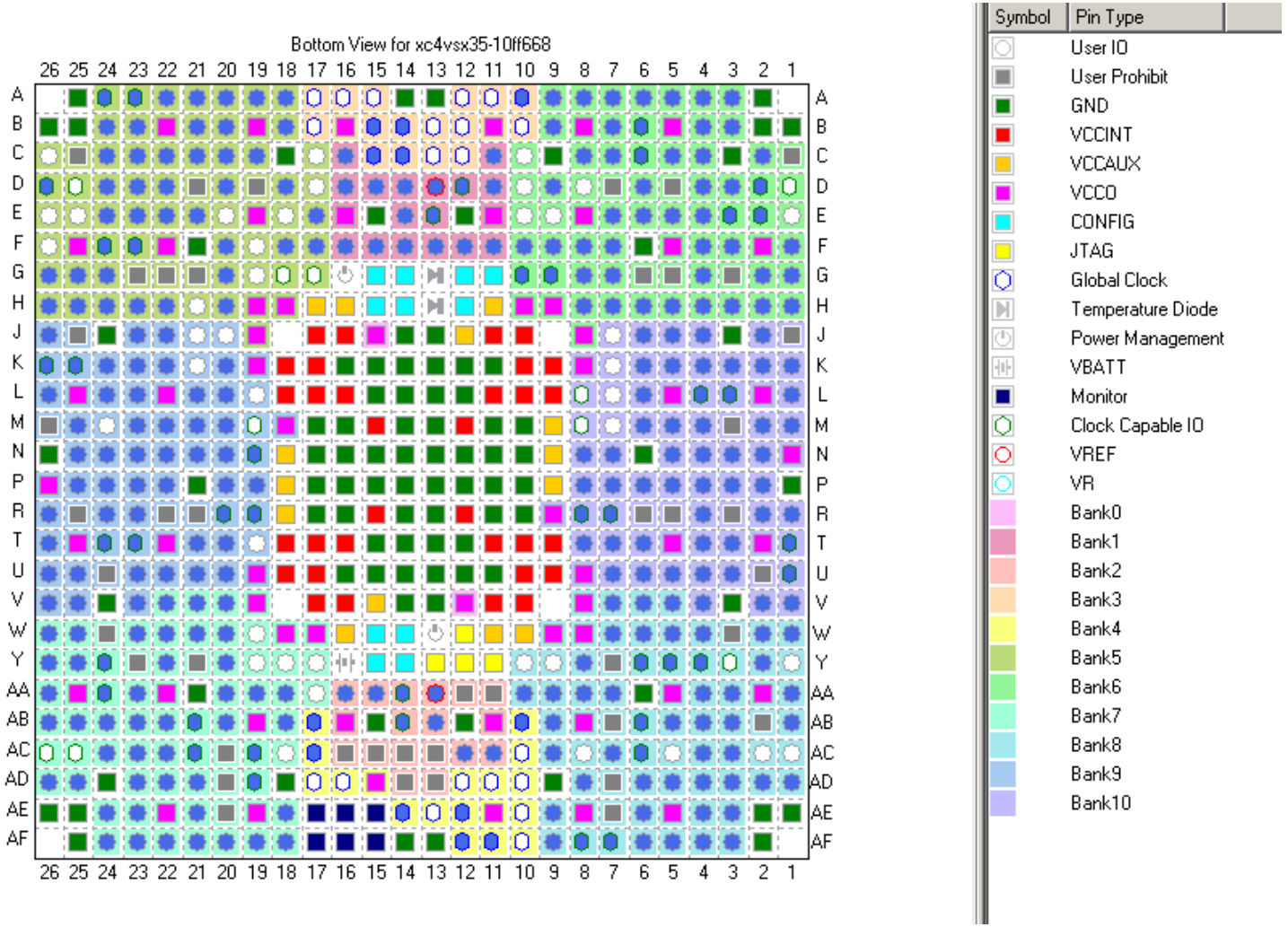

Figure 8-1 Pin Locations

8.4 Programming Notes

All FPGAs on the Station Board are programmed through their 8-bit wide configuration port. The Station Board CMIB software requires the Binary (.bin) output file to program the Xilinx FPGAs. This is set by selecting "Properties…" from the "Process" pull-down menu in the Xilinx ISE software. Select the "General Options" and check "Create Binary Configuration File".

9 References

Brent Carlson, “Refined EVLA WIDAR Correlator Architecture”, NRC-EVLA Memo #014, October 2, 2001.

10 Index