REQUIREMENTS AND FUNCTIONAL SPECIFICATION

Station Board Output FPGA

RFS Document: **A25055N0000**

Revision: 2.4

Zoran Ljusic and 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

01 April 2011

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

CMIB	Correlator Monitor & control Interface Board
CRC	Cyclic Redundancy Check
DCM	Digital Clock Manager.
DDR	Double Data Rate
EVLA	Expanded Very Large Array
FIFO	First In First Out.
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
Gbps	Giga bit per second
HSTL	High Speed Transceiver Logic
ISR	Interrupt Service Routine
LVDS	Low Voltage Differential Signaling
LVTTL	Low Voltage Transistor-Transistor Logic
LSB	Least Significant Bit
MSB	Most Significant Bit
PCB	Printed Circuit Board
PPS	Pulse Per Second (one pulse per second).
PPS/10	Pulse Per Second/10 (one pulse every 10 seconds).
RAM	Random Access Memory
RFI	Radio Frequency Interference
SB	Station Board
SBOF	Station Board Output FPGA
SIND	Sample INDicator
TBD	To Be Determined
VLBA	Very Long Baseline Array
VLBI	Very Long Baseline Interferometry
VSI	VLBI Standard Interface
WIDAR	Wideband Interferometric Digital ARchitecture

1 Revision History

Revision	Date	Changes/Notes	Author
0.0	Dec 2003	Initial Draft	D. Fort
0.1	April 2005	Updated initial draft	Z. Ljusic
1.0	May 2005	Complete version; incorporates changes suggested during the peer review.	Z. Ljusic
1.1	Sep 2006	Implemented changes imposed by the Filter FPGA modifications. All of the sections, except for the pulsar block, have been more or less modified. Effort has been made to minimize changes on the MCB interface.	Z. Ljusic
2.0	April 2007	Implemented changes based on the prototype testing, mainly within the MCB interface.	Z. Ljusic
2.1	03 Jan 2008	Time Interval Counter.	D. Fort
2.2	20 Feb 2009	Radar Control Changes. Output chopper (with seeds) added.	D. Fort
2.2	14 Oct 2009	Added SCLK PLL reset. Normalized FPGA version. Added clock phase shifting for test purposes.	D. Fort
2.3	07 Oct 2009	Added additional System Tick Delay bits.	D. Fort
2.4	01 Apr 2011	Pinouts, etc added.	D. Fort

2 Introduction

This document describes detailed requirements and design concepts for the Station Board Output FPGA, SBOF. Background on the SBOF can be found in NRC-EVLA Memo #014 (EVLA Memo 31 where it is referred to as the “cross-bar switch”).

The development plan for the SBOF is as follows:

1. Develop and test the design in an FPGA. The design will be done in Verilog HDL.

3 Context

A simplified block diagram of the Station Board is shown in Figure 1.

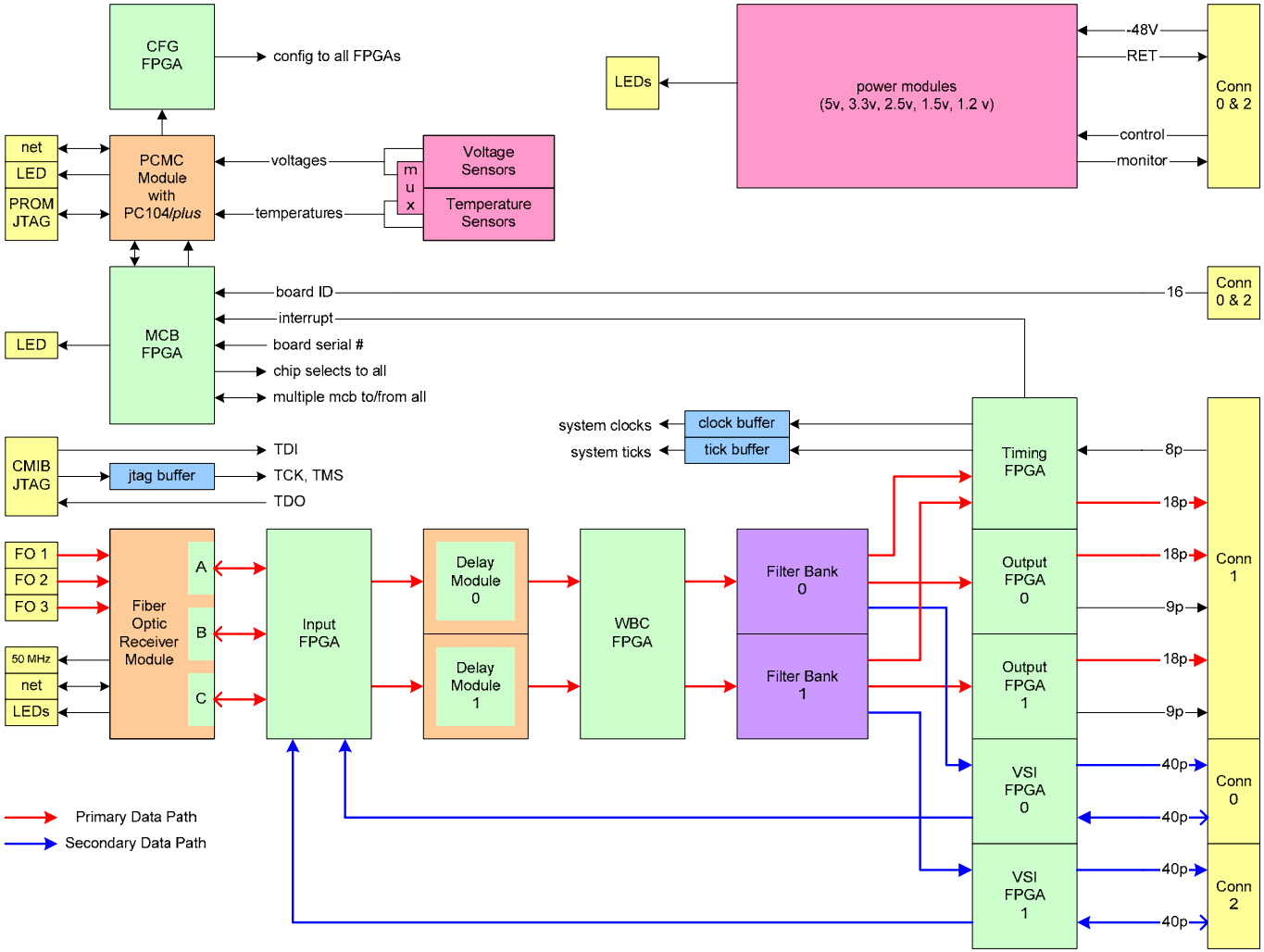


Figure 1 Block diagram of the Station Board

There are two SBOFs on the Station Board, one for each base band. Each SBOF receives the sub-band outputs from the 18 Filter FPGAs. The primary function of the SBOF is to prepare the sub-band data for output to the baseline boards.

4 Overview

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

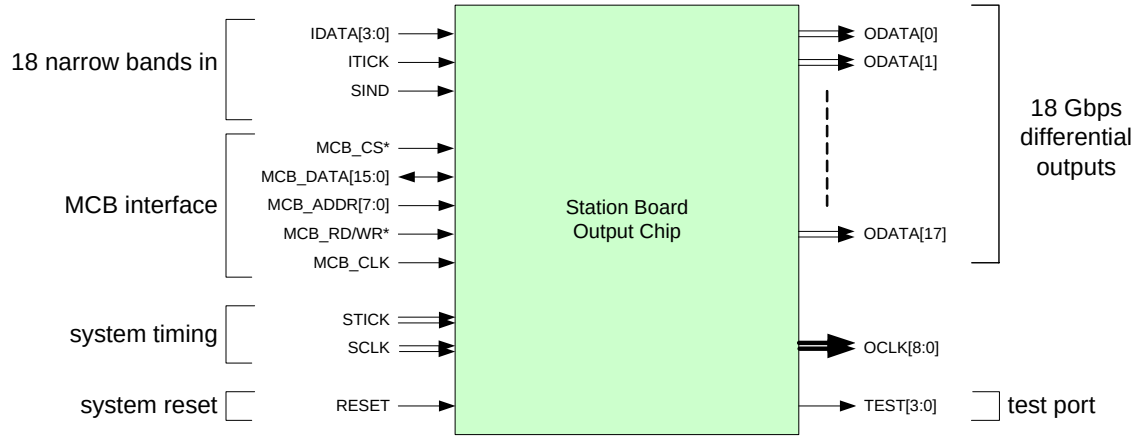


Figure 2 Input/Output diagram of the SBOF

The 18 sub-band inputs consist of four data lines, tick (ITICK) and sample indicator (SIND). The data valid has been encoded into the data lines as the “forbidden” state (different for 4-bit and 8-bit encoding). The data and tick lines are running at 256 MHz rate. In addition, there are system tick and system clock lines coming to each FPGA.

The output of the SBOF consists of 18 differential high-speed, Gbps, data outputs. Within each data stream, there are embedded tick, station ID, sub-band ID, base band ID and CRC. The format of the outputs is given in the ‘HM Gbps Cable Signaling Specification’ (A25022N0041). There are nine differential 128 MHz clock signals leaving the FPGA. The clock signals are directly routed to the connector.

4.1 System Input Signals

- SCLK is the system 128 MHz clock supplied as an LVDS pair. This clock is the buffered output of a crystal PLL that is locked to the 64 MHz clock from the Fan-out Board. All chips on the Station Board use SCLK in order to prevent the buildup of clock jitter.

- STICK is the system timing tick (10 millisecond period) supplied as an LVDS pair.

4.2 Narrow Band Input Signals

There are 18 input data buses. Each bus consists of the following signals:

- IDATA [3:0], single ended HSTL-III, is clocked in by 256 MHz clock.
- ITICK, single ended HSTL-III, is clocked in by 256 MHz. 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 width of ITICK is one half a cycle of 128MHz (SCLK).
- SIND, single ended HSTL-III, indicates sample position and it is clocked in by 256MHz clock.

4.3 Narrow Band Output signals

The following signals go from the SBOF through connectors, cables and a Station Data Fan-out Board to a baseline board.

- ODATA[17:0], 18 differential Gbps LVPECL data outputs, a pair per narrow band.
- OCLK[8:0], 9 differential LVPECL 128 MHz clock outputs.

4.4 MCB Interface Signals

- MCB_ADDR [7:0] is the input 8-bit address bus for accessing internal Filter 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 chip 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 the MCB signals are LVTTL.

4.5 Test Port

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

5 Requirements

The following is a list of SBOF functional requirements.

5.1 Functional Requirements

1. Re-clock input data to the internal clock domain.
2. Check sample alignment of different input data.
3. Generate pulsar gate(s) and modify data validity.
4. Data invalidation gate for bad data after model changes.
5. Permanently invalidate data on narrow band basis (from CMIB).
6. Insert station, base-band and sub-band ID (from CMIB).
7. Calculate CRC for all the data lines and insert the value into the output stream.
8. Switch any input sub-band to any output sub-band.
9. Create a two wire differential Gbps output for each input narrow band.
10. Format and store requested sub-band in given format (radar mode).

5.2 Performance Requirements

1. The SBOF shall operate on the input sampled data and control signals with a clock rate of 256 MHz. The FPGA will take a 128 MHz input clock and develop its own internal 256 MHz clock.
2. The synchronous MCB interface shall be capable of operating with a clock that is neither frequency nor phase synchronous with the 128 MHz or 256 MHz data clock. The FPGA will support an MCB interface clock with a maximum rate of 33 MHz.
3. The power dissipation of the FPGA running at 256MHz, based on Altera's Power Analyzer calculation, is 3.3W.

5.3 Environmental Requirements

1. The SBOF will be surface mounted on the Station Board PCB. Additional heat sinks may be attached to the FPGA to reduce its junction 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. Device chosen for this design is Altera's StratixGX EP1SGX10CF762C5.

5.4 Interface Requirements

As shown in Figure 4-1, the SBOF has several interfaces. The following sections show the functional relative timing between the signals comprising each interface.

5.4.1 System Timing Interface Requirements

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

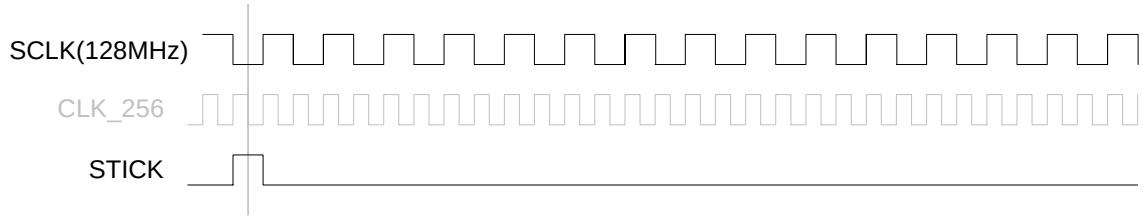


Figure 3 System Functional Timing

5.4.2 Narrow Band Data Input Requirements

Shown below is the functional timing for the signals described in section 4.2. The SIND signal shown on the figure indicates 128MHz data rate in four-bit mode.

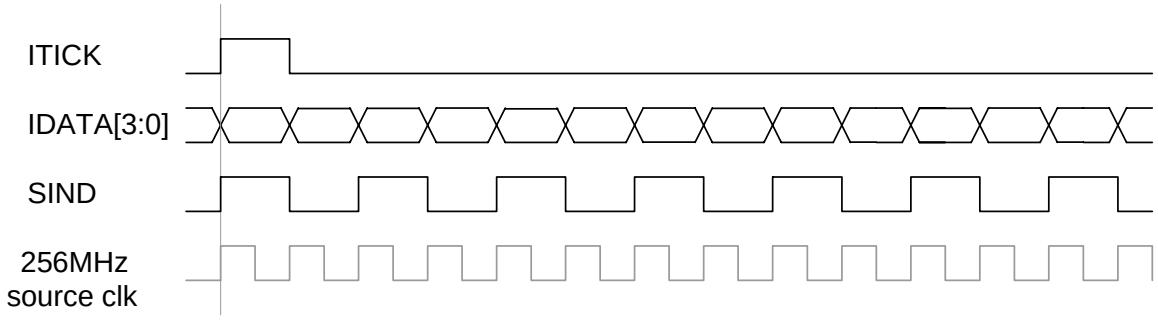


Figure 4 Narrow Band Input Functional Timing

5.4.3 MCB Interface Requirements

The MCB (Monitor & Control Bus) interface allows a microprocessor interface to write into the SBOF to configure and control it and to read from it to verify configuration information and obtain status and monitor information.

When the microprocessor wants to write to a SBOF, it puts the data on the MCB_DATA bus, the target register address on the MCB_ADDR bus and drives MCB_RD/WR* and MCB_CS* low some time before a rising edge of the MCB_CLK and keeps 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 input chip then writes the data into the specified register on the rising edge of MCB_CLK. A write requires one clock cycle as shown below.

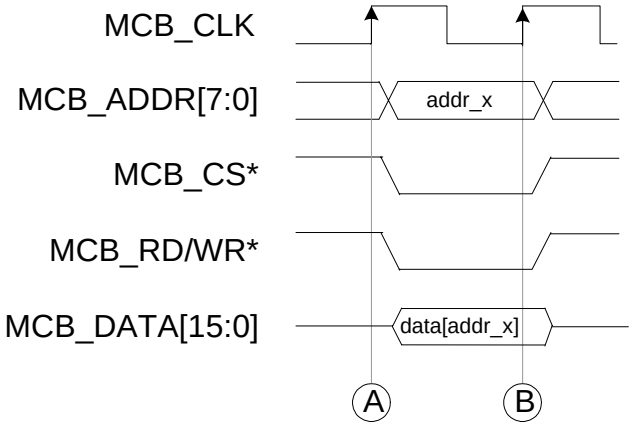


Figure 5 MCB Interface WRITE Functional Timing

When the microprocessor wants to read from SBOF, it drives MCB_RD/WR* high, puts the desired register address on the MCB_ADDR bus and drives the corresponding MCB_CS* low. Read cycles require one clock cycle to setup and an additional clock cycle to read as shown below. Subsequent reads at the same address may take only one clock cycle depending on the microprocessor.

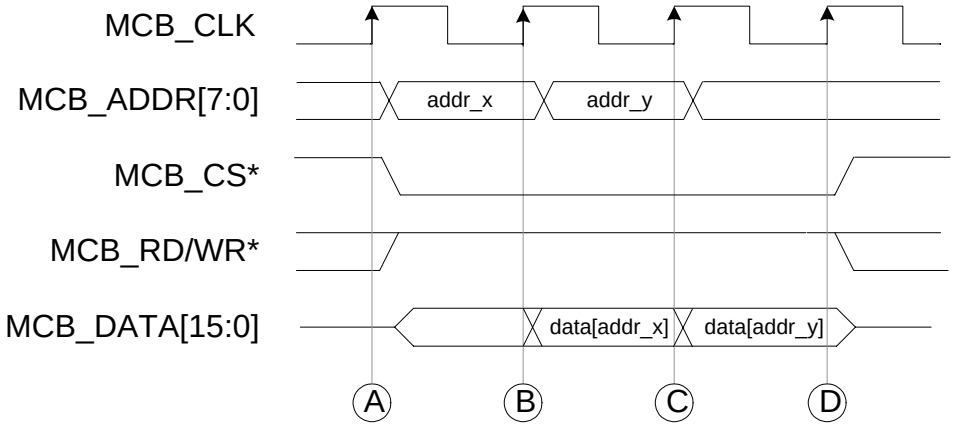


Figure 6 MCB Interface READ Functional Timing

The microprocessor will be interrupted on a version of STICK that has been delayed sufficiently to make sure that all chips on the board have received their local version of the TICK. This allows a pipeline delay through the chips on the board. Configuration information sent from the microprocessor to the SBOF may take effect immediately and should only be sent when the output of the chip is irrelevant.

Status information, such as error bits, will show any occurrence of an error since the last read of that information. Another method would be to latch the status information on the TICK and clear the primary register. This method assumes that the status information is read every TICK by the ISR. Control information may include bits which, if set; cause a specified error to occur to allow testing of the reaction of the software to that error.

6 Functional Specifications

On the Figure 7 a simplified functional block diagram of the SBOF is given.

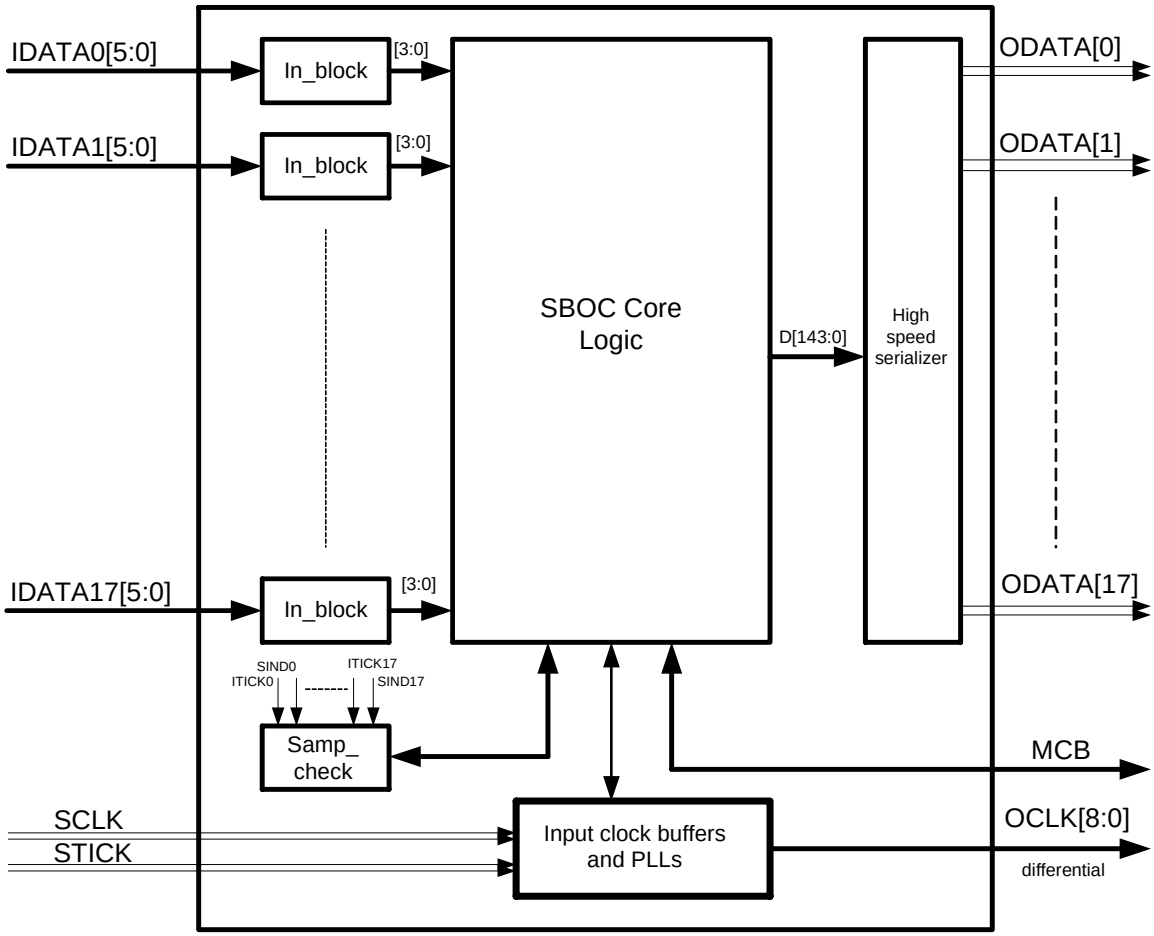


Figure 7 SBOF top-level block diagram

Each functional block on the above figure is described in more detail in the sections that follow.

6.1 Input Block

There are three main functions performed by the Input block:

1.Input signals are being clocked in within the FPGA IOE (Input Output Element). The clocking is performed in a Double Data Rate (DDR) fashion by 256MHz system clock. For the purpose, Altera's MegaWizard is used to generate the DDR module.

2. The DDR module produces two output data buses with the appropriate ITICKs and SINDs. One bus represents data clocked by the rising clock edge and the other one clocked by the falling clock edge. The bus selection is performed based on 'data_sel' signal from MCB interface. The signal could be different for different data ports, depending on relative timing between the system clock and input data. This will be determined during Station Board prototype testing.

3. There is a CRC calculator within each of 18 Input Blocks. It can calculate CRC on any of four input data wires, one at the time. It reports the 4-bits results to the MCB interface. Wire for calculation is selected by the MCB Interface. The calculation is performed between the data ticks.

6.2 Sample Checker

A functional block diagram of the module is given on Figure 8.

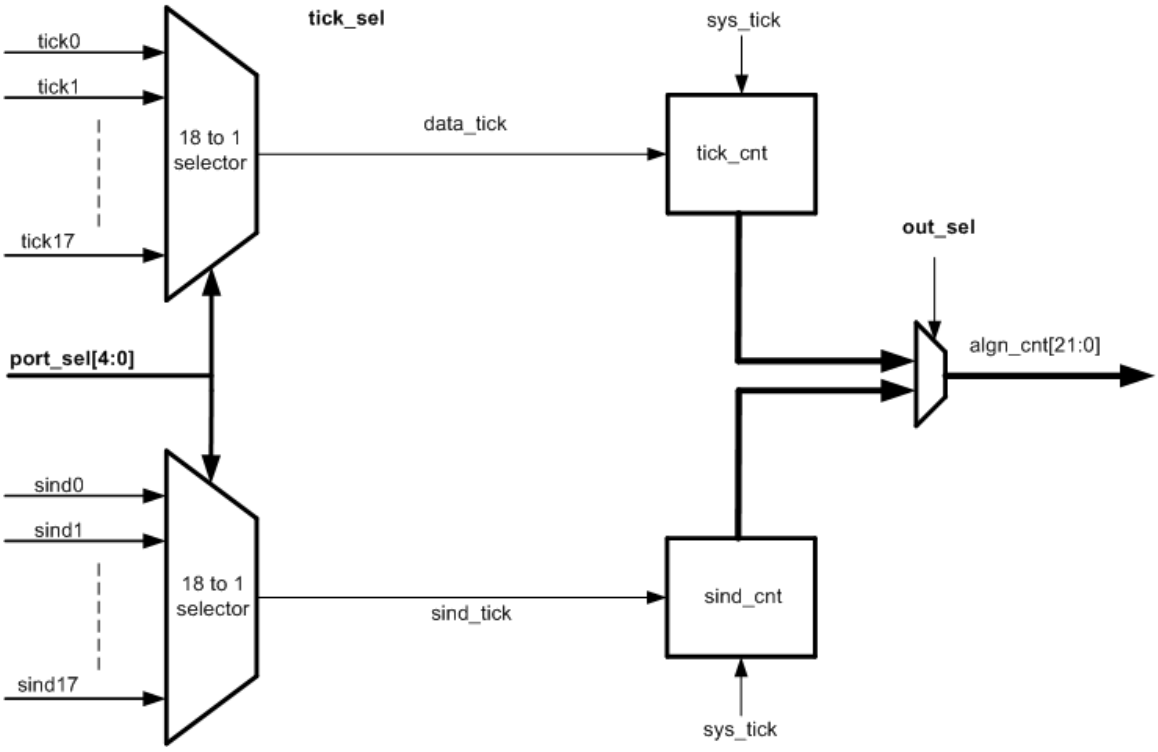


Figure 8 Sample Checker functional block diagram

This block makes two time interval measurements – selected data tick to system tick and selected sample indicator to the system tick. The main devices within the block are two 22-bit counters capable of counting up to 2,560,000 clock cycles at 256MHz, which is duration of 10ms interval. The CMIB selects which port to measure (same for tick and sind) and which result to read.

The data streams coming from different Filter FPGAs are not time aligned. This means that data ticks are time skewed with respect to the system tick. There is no FIFO per port which will absorb the time difference between different ports. The time tuning is performed in Filter FPGA for each port based on the counter's value. The counter's value should be an integer number of sample duration for the port. For 128MHz data rate the counter's value should be: Counter Value = N x 2; for 64MHz it should be: Counter Value = N x 4; for 32MHz it should be: Counter Value = N x 8 etc. This so-called sample alignment insures that downstream data embedding occurs at the beginning of a sample when sample duration is more than one 256MHz clock cycle long. Based on the counter's value the CMIB will adjust the corresponding delay line in the Filter FPGA, which sources the data, so that this condition is satisfied. Once the sample alignment is achieved, this value is passed on to the appropriate Baseline Board in the system so the incoming data can be properly processed. The two figures below illustrate the timing relationship between the data tick, sind and the data itself for both 4-bit and 8-bit samples. Note that at 256 Ms/s (4-bit) and at 128 Ms/s (8-bit) sind is always high so the sind to system tick time interval measurement is meaningless.

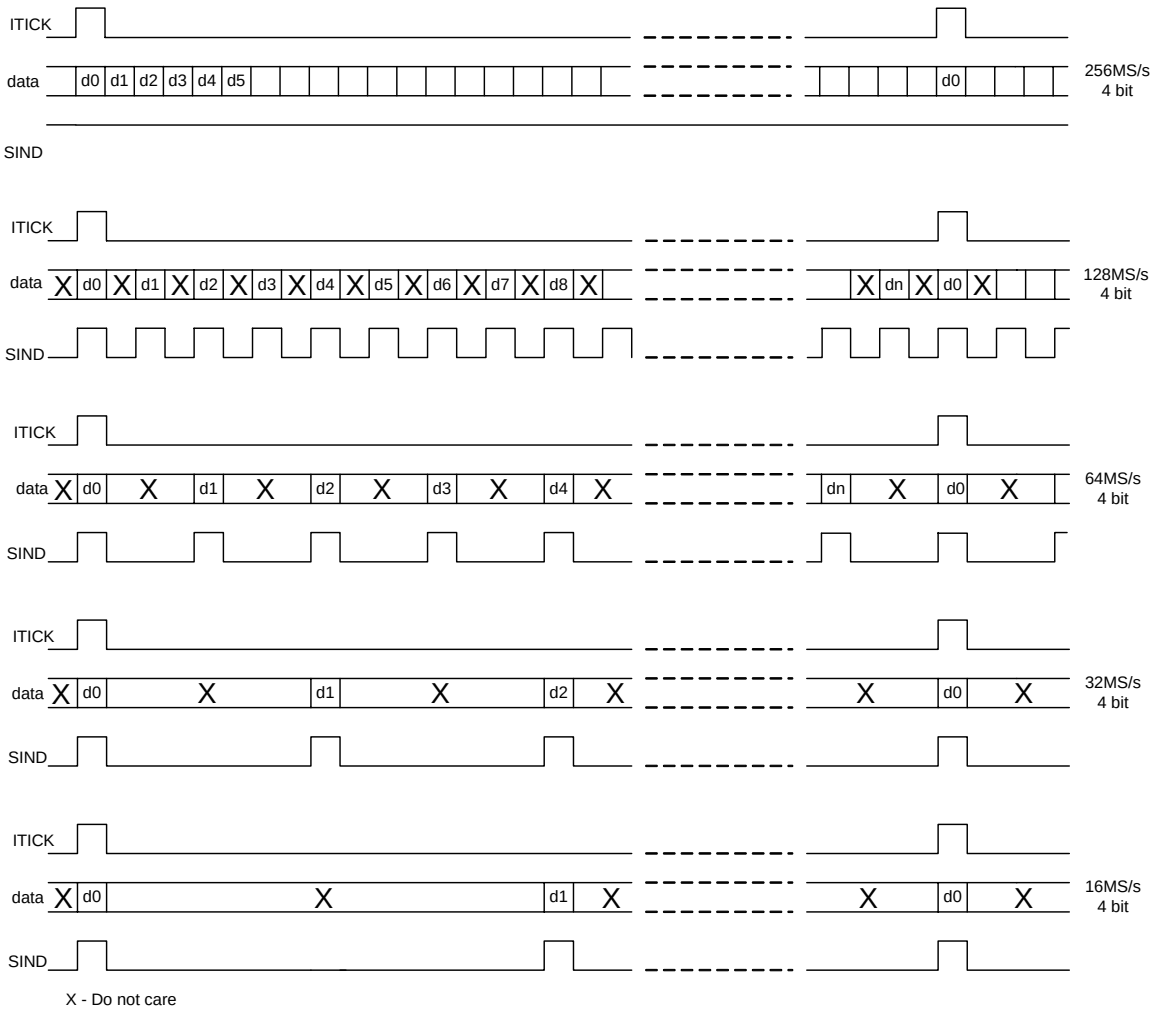


Figure 9 Input data format in 4-bit mode

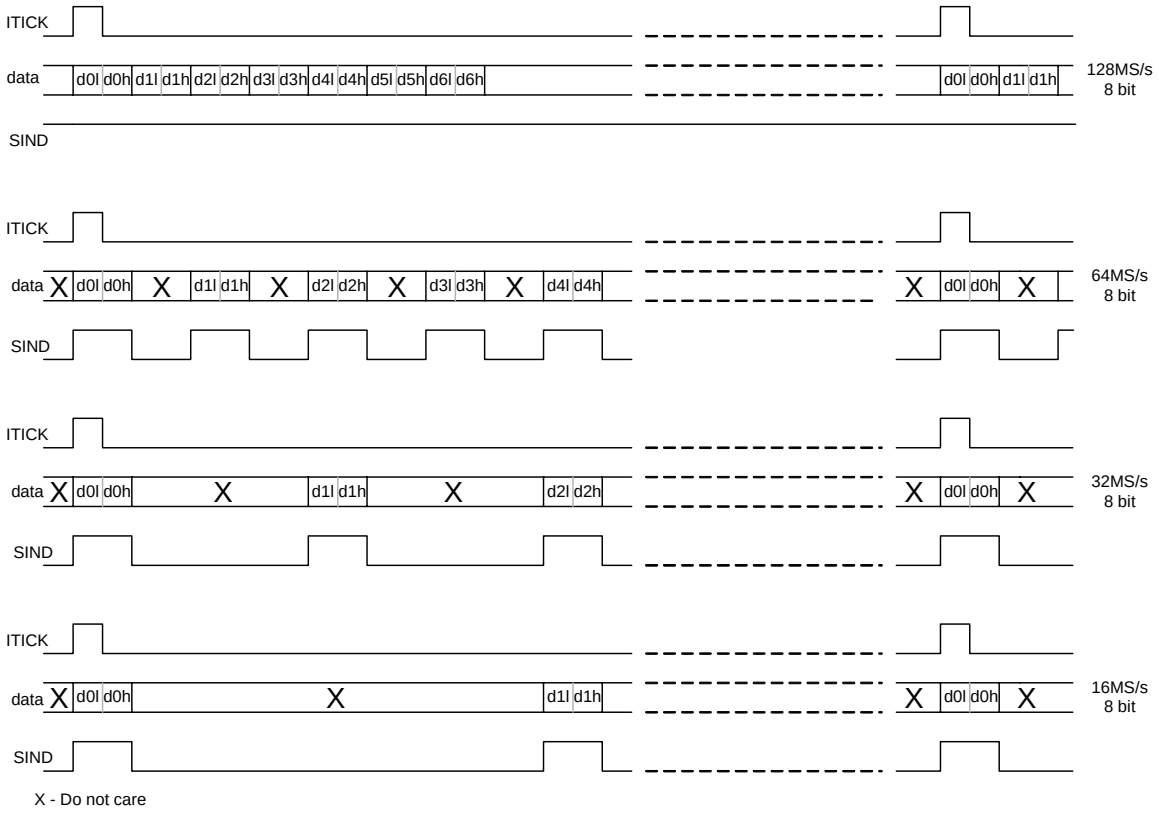


Figure 10 Input data format in 8-bits mode

6.3 Input clock buffers and PLLs

This block contains clock buffers and a PLL. System clock of 128 MHz is received as a LVDS signal and taken to the PLL. Based on 128 MHz input clock the PLL produces 256MHz used within the FPGA.

6.4 High Speed Serial Transmitter

The main function is performed within a Mega Wizard generated block. Each 8-bits wide narrow band data bus at 128MHz is converted to 1024Gb high-speed differential output.

6.5 SBOF Core Logic block

A simplified block diagram of the Core Logic block is given on the Figure 11.

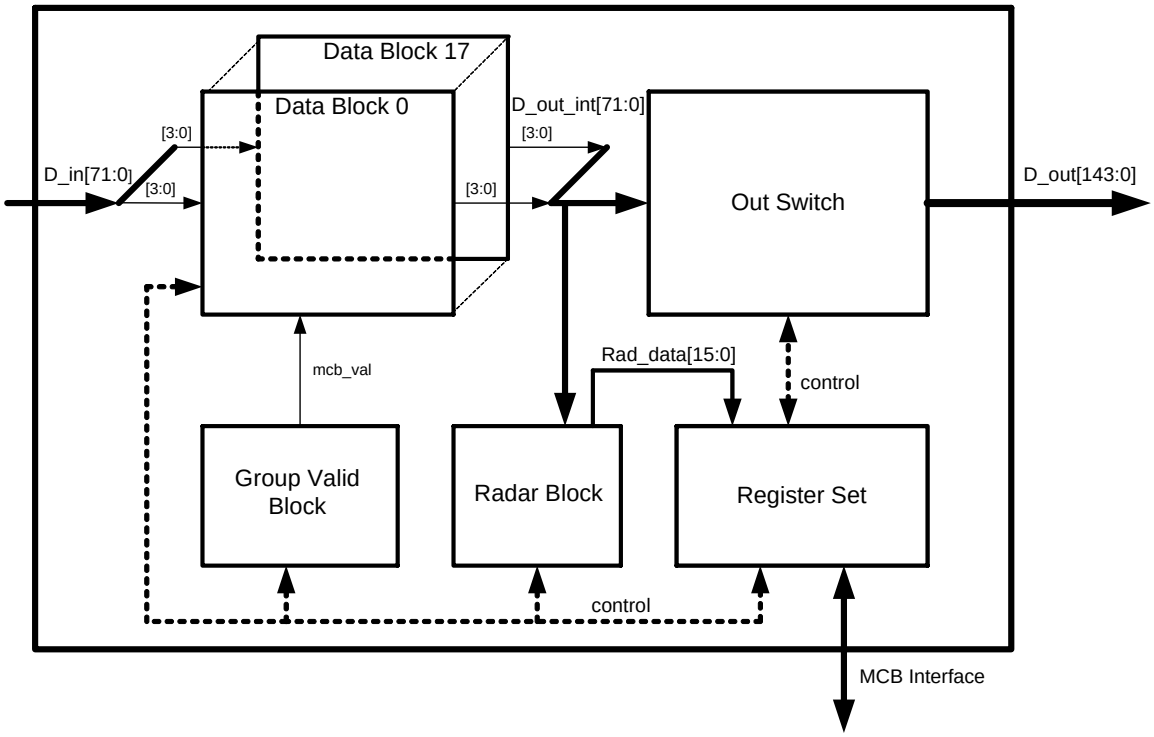


Figure 11 SBOF Core Logic functional block diagram

Within this block main data processing is performed. As it can be seen from the figure above the Core Logic module consists of several functional sub modules. Description of the sub modules is given in sections that follow.

6.5.1 Data Block

A functional block diagram of the module is given on the Figure 12.

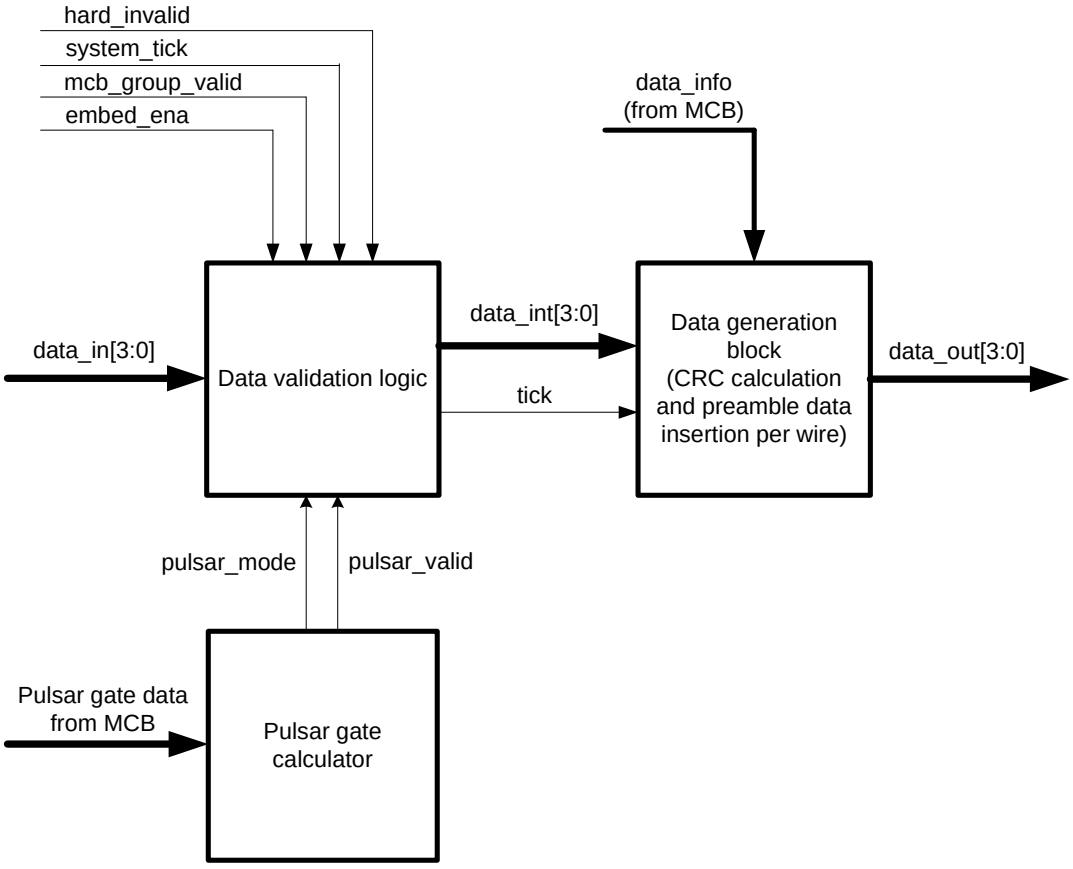


Figure 12 Data Block functional diagram

Main data processing is performed within the Data block. The Data Block can be further divided into three functional sub blocks: Data generation, Pulsar gate and Data validation sub block.

Within the Data generation sub block each data line, each bit, is modified according to the requirements described in A25022N0041 Protocol Specification. CRC are calculated between the system ticks and are embedded with the rest of the data into the data stream. An illustration of the operation is given on the two following figures.

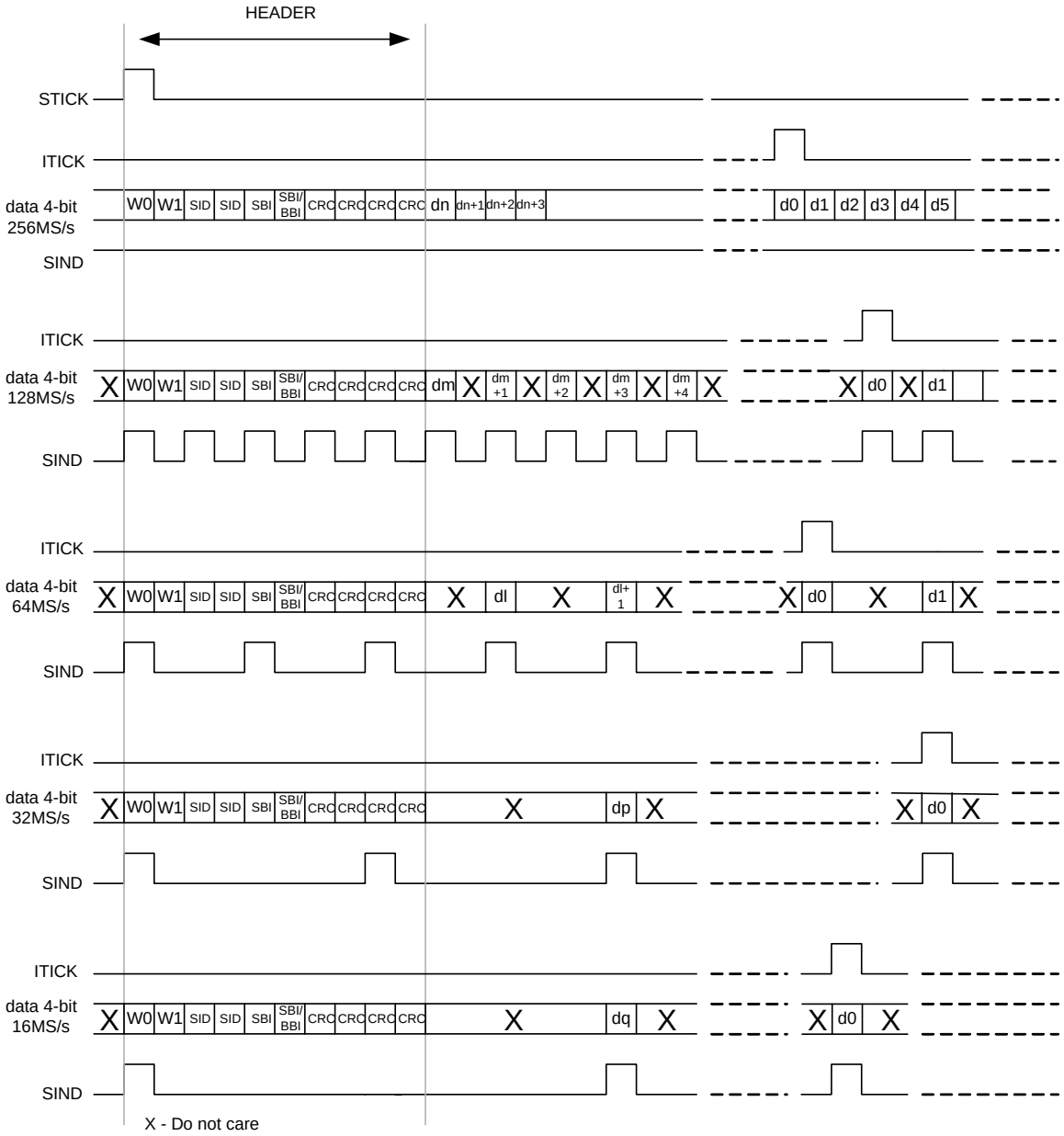


Figure 13 Header embedding into 4-bit data

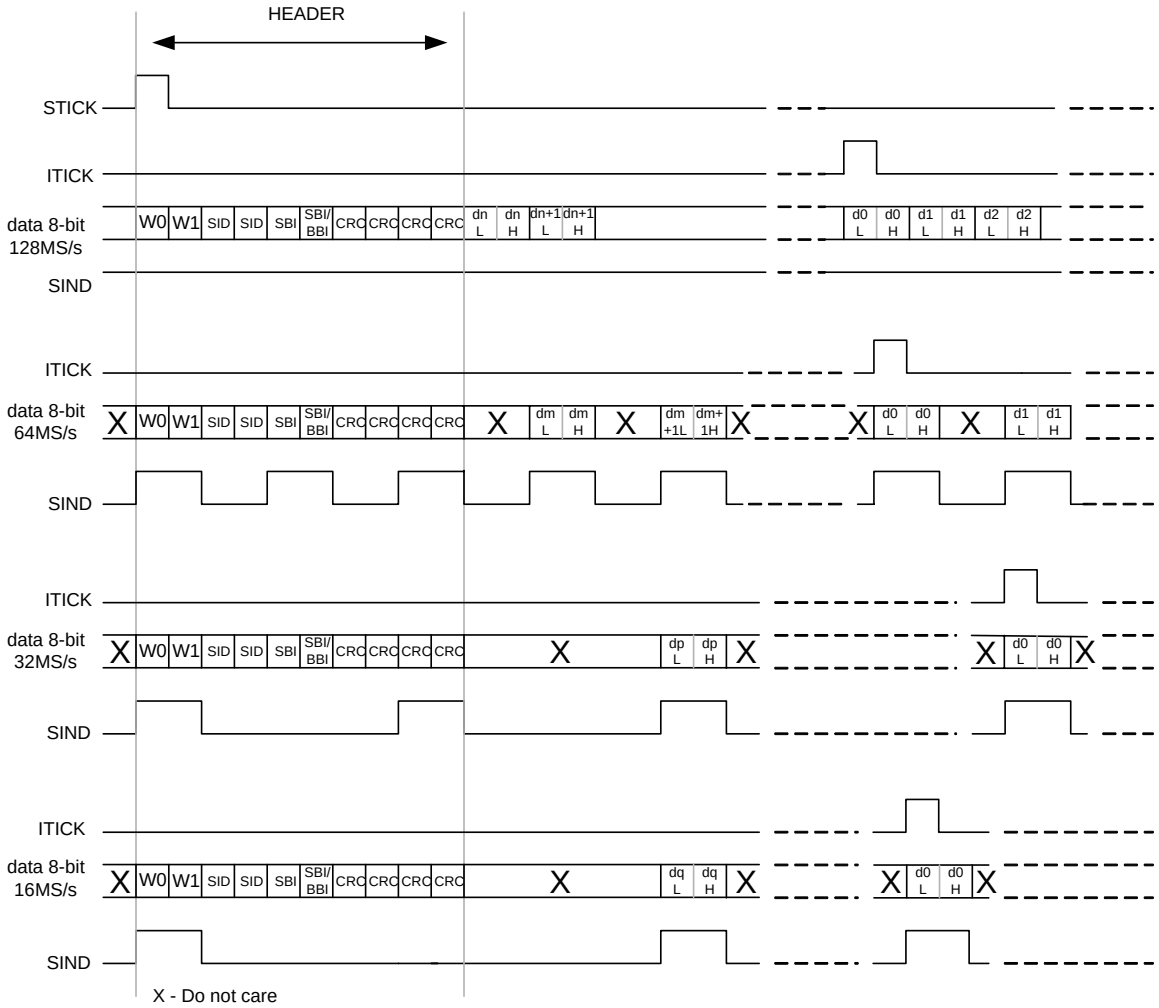
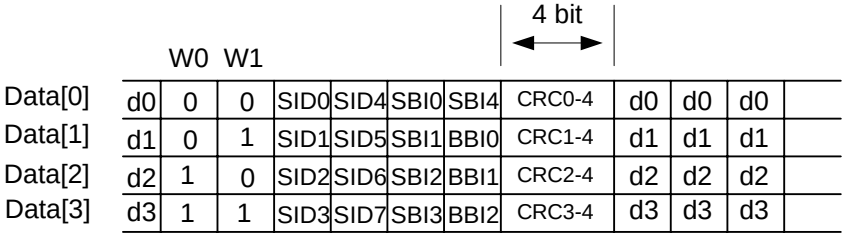


Figure 14 Header embedding into 8-bit data



SID[7:0] - Station ID
SBI[4:0] - Sub-band ID
BBI[2:0] - Base band ID

Figure 15 Header definition

Station ID, Sub-band ID and Base-band ID are provided to the block thru the MCB interface. CRC 4-bit values are available to the MCB interface. The CMIB has to select which CRC value will be available, CRC0-4, CRC1-4, CRC2-4 or CRC3-4. These values can be compared with CRC calculation in the Baseline Board to verify connectivity. Data embedding can only happen if the signal 'embed_ena' is high. Even in case when data embedding is not enabled data is delayed the same amount of time. To invalidate all outgoing data the CMIB simply asserts 'hard invalid' signal. This signal is unique for each of 18 data channels. If the invalidation of all 18 data channels is required at the same time then the signal 'mcb_group_valid' should be set low by the CMIB, see Figure 17. Care should be taken with data and system tick pipelining so that first header nibble, W0, replaces sample, which was aligned with system tick within the Sample Checker module.

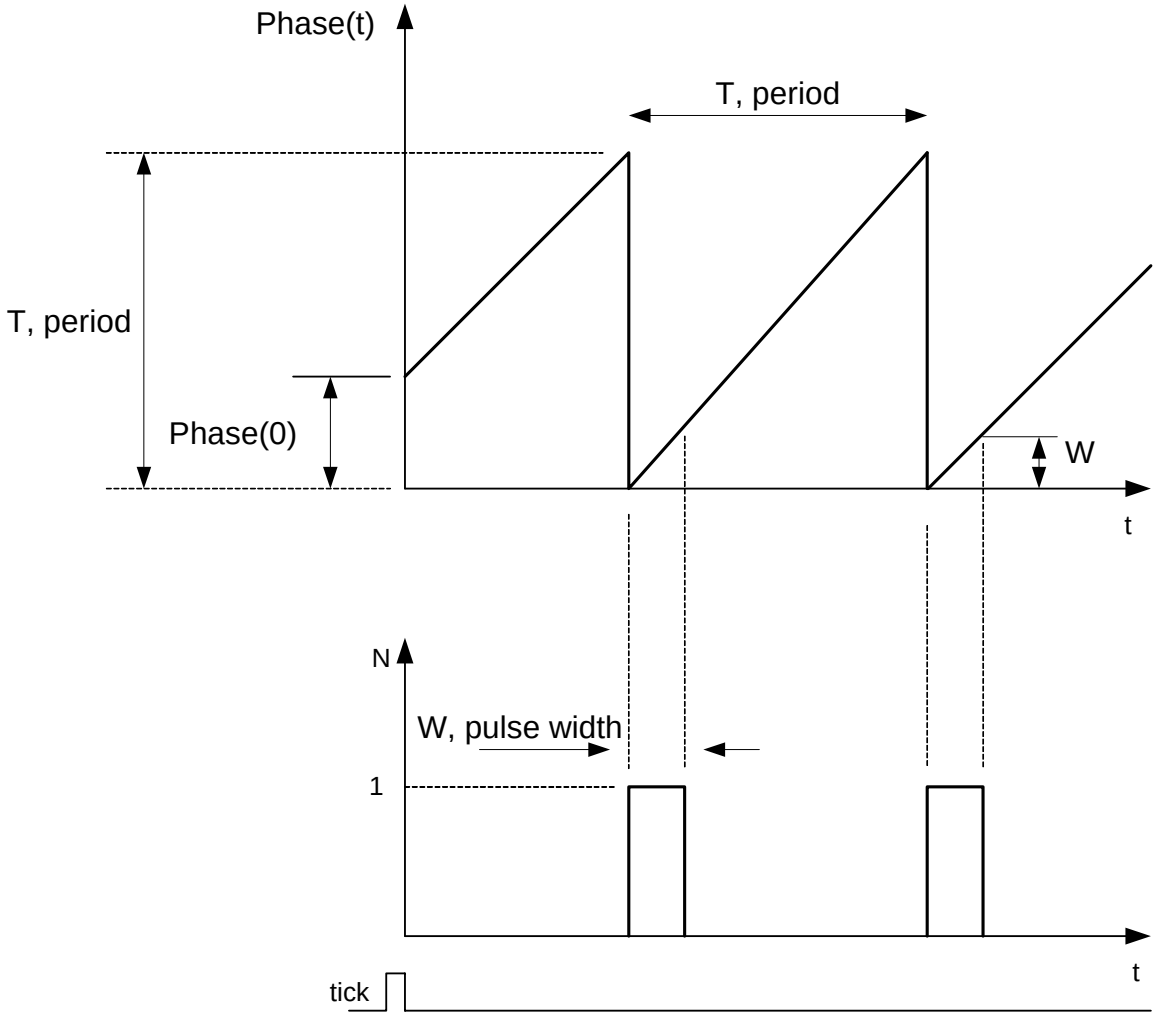


Figure 16 Pulsar gate calculation

Pulsar mode of operation is enabled by the CMIB. All the parameters shown on the figure above are defined in number of microseconds.

T is pulsar period and is represented by a 22-bit number. Its range is from 0 to 4,194,303.00 μs, (0x00_0000 to 0x3F_FFFF). It can be updated on each tick.

Phase(0) is phase at zero time , i.e. at the 10ms pulse. This parameter is represented by a 22-bit number as well. Its range is from 0 to 4,194,303.00 μs, (0x00_0000 to 0x3F_FFFF). New value can be written on each tick.

Pulse width is represented by a 22-bit number. Its range is from 0 to 4,194,303.00 μs, (0x00_0000 to 0x3F_FFFF). The Pulse width can be updated on each tick.

Phase(t) is phase at any time and is represented by a 22-bit number. Its range is from 0 to 4,194,303.00 μs, (0x00_0000 to 0x3F_FFFF). This parameter is implemented as a counter. The counter is loaded with Phase(0) at every tick and it is incremented by one every 1μs after that. When the Phase(t) counter reaches value assigned to Period T it resets itself. At that moment, Phase(T) = 0, the logic which defines the pulse goes high. During the pulse duration data is qualified as valid to the downstream hardware. When the Phase(t) counter reaches value assigned to parameter Pulse width the logic which defines the pulse goes low. This makes data invalid to the downstream hardware.

The CMIB can keep on updating these parameters on every 10ms or set the logic in a 'self lock mode'. This means that after the values are taken by the logic on a first tick on which they were available and correct, the CMIB can assert 'self_lock' bit. The assertion forces logic to take the existing, internally calculated, values at the tick rather than values provided by the CMIB.

6.5.2 Group Valid Block

This block is used to invalidate all the data during system reconfiguration. The word 'Group' here refers to the group of all 18 narrow band data streams. It is enabled by the CMIB. It is implemented as a simple 16-bit counter. On each tick, the counter is reset to zero and incremented by one after that. The CMIB defines the moment within the 10ms interval after which the valid signal goes high. This is defined by a 16-bit number. The logic takes in a new number value on every tick. The given value multiplied by 156.25 ns, determines the transition time after the tick (40 256 MHz clock cycles). The maximum value that the counter can reach during the 10ms interval is 64,000.00 or FA00h. If the entire 10ms interval needs to be invalidated, the CMIB writes FA01h, at least, to the proper address**. Writing 0x0000 to the location makes data valid all the time, if the mode is enabled. Not enabling the mode, i.e. if 'Mode_ena' stays low, causes the valid signal to go low, effectively invalidating the data, regardless of the numeric value for the transition**. An illustration of the function is given in the figure below. After valid goes high in the middle of the 10ms interval the CMIB should change

the transition number to zero or on the next tick valid goes low for the first half of the interval.

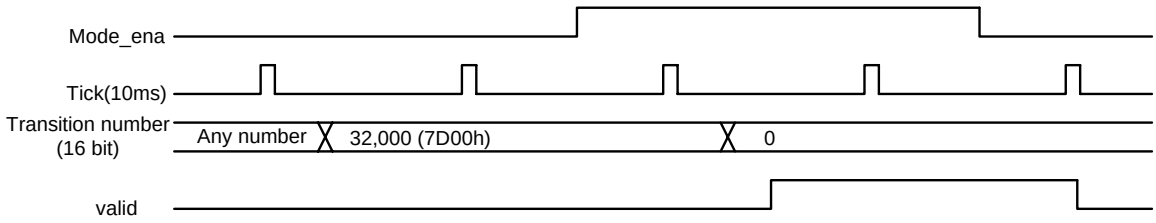


Figure 17 Group Valid signal

6.5.3 Radar Block

A functional block diagram of the module is given on the Figure 18.

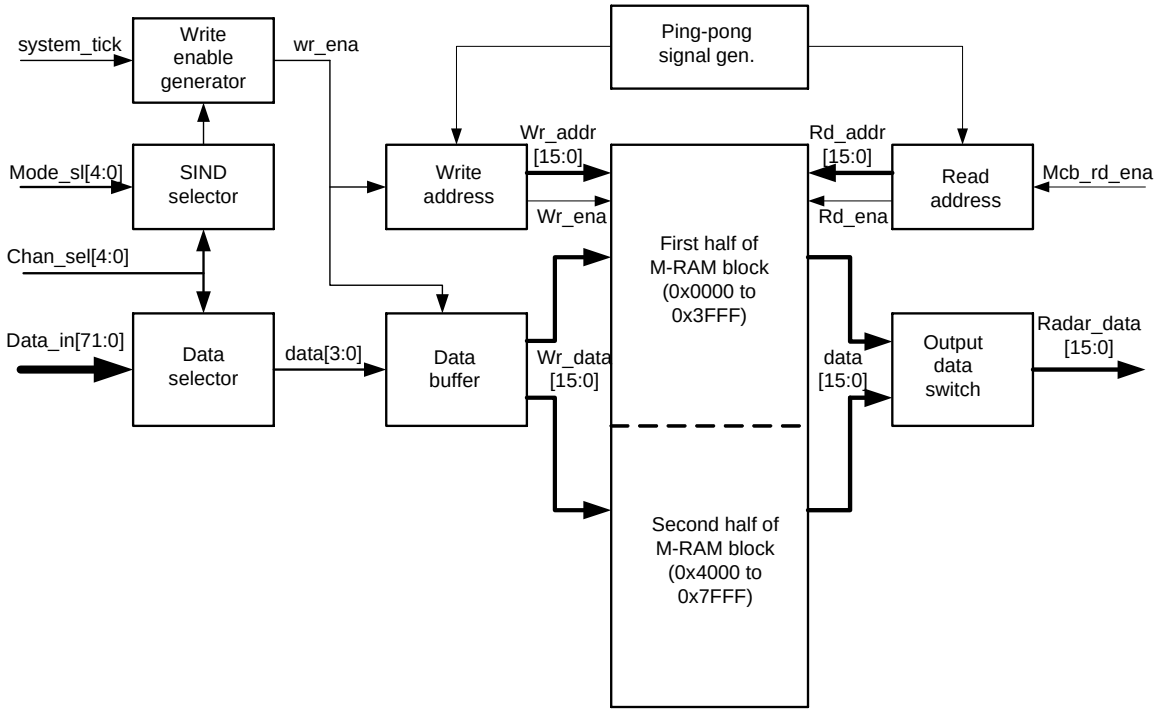


Figure 18 Radar block functional diagram

The heart of the Radar module is the Altera's 32767X16 M-RAM block, 524272 bits, 131068 4-bits nibbles, or 65534 bytes. Based on this calculation one half of the block

memory can store (131068/2) –10 = 65524 data samples at 256MHz rate in four bit mode. The total number of samples within 10ms at this rate is 2,560,000; this gives relative capacity of the memory of 65524/2560000 = 2.6%. For the 128MHz data rate, this number increases by approximately 2 times. The RAM block is divided into the two sub banks: first half from 0x0000 to 0x3FFF and second half from 0x4000 to 0x7FFF. The M-RAM block division is transparent to the user. This prevents data loss that can happen within one 10ms time interval. The CMIB selects one out of 18 narrow band channels via 5-bits channel selection lines. The incoming 18 narrow band data stream have already passed thru the data block and have all the information, (SID, SBI, BBI and CRC), already embedded. Selected data is first written to a 16 bits wide buffer and then to the memory. Write addresses are generated based on the data rate, which is defined by the selected channel SIND signal, and the ping-pong signal. During one 10ms interval 32KB of data can be stored in one of the banks. Data overwriting is not allowed. If the write address reaches its maximum value before the tick comes, the incoming data would be written to the last location repeatedly. The last word written is set to 0xffff.

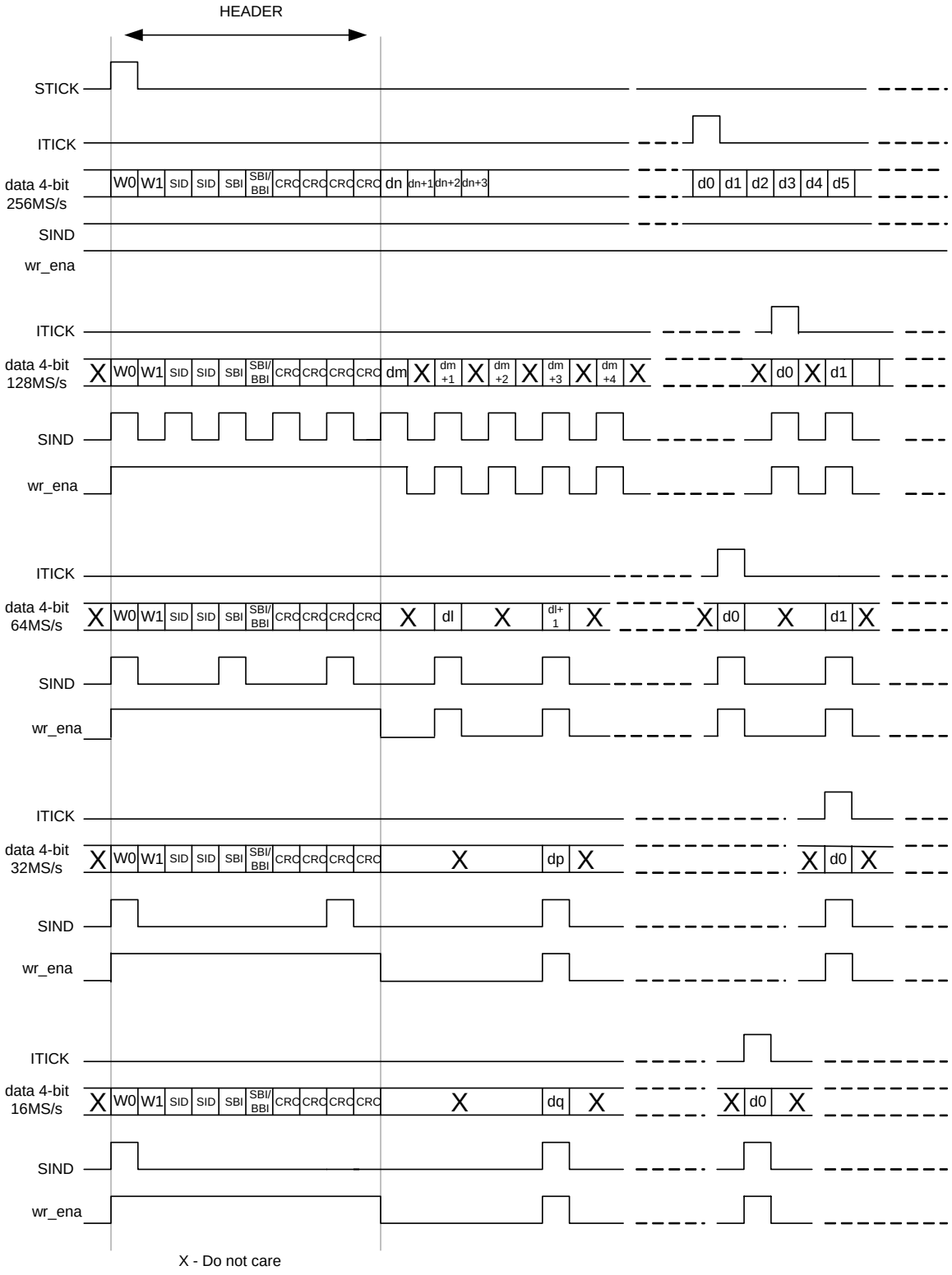


Figure 19 Radar data write in 4-bits data format

The Figure 19 above illustrates how the data is written the memory in 4-bit data format. In 8-bits data format writing is similar – the header portion of the ‘wr_ena’ stays the same and the rest is replaced with appropriate SIND signal.

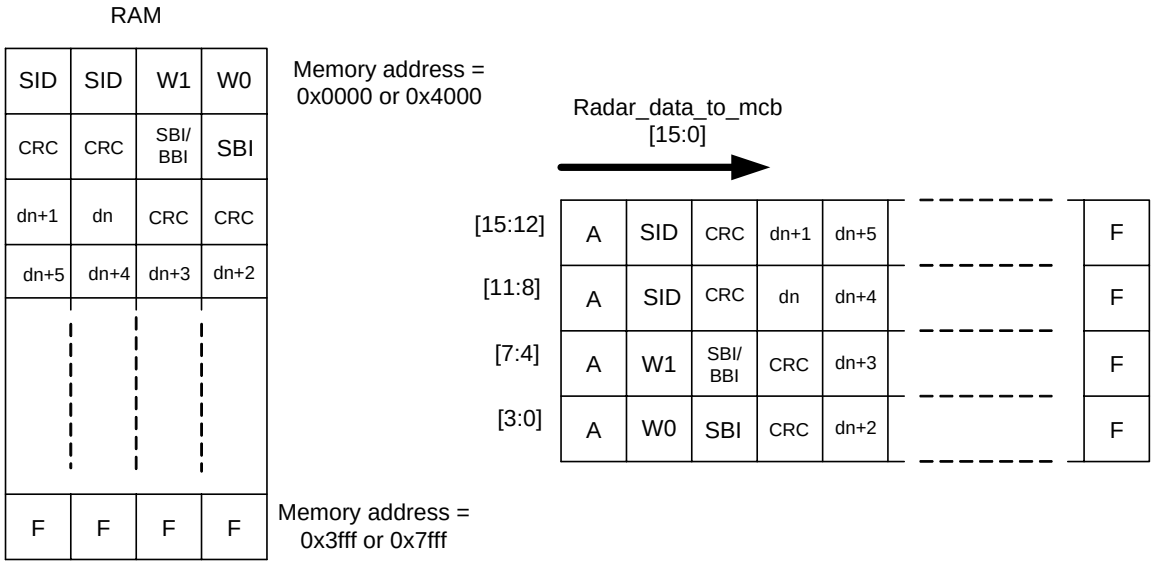


Figure 20 Radar mode read data

The Figure 20 illustrates how data is being written into the RAM and read out by the CMIB. The example is for 256 MS/s in 4-bits mode. Writing to the RAM is performed on a first-come-first-in basis. When the CMIB asserts read enable signal via its address bus the data starts coming out down the CMIB data bus. First word is set to 0xaaaa followed by the payload data. Last word is always 0xffff. For other rates RAM data is similar, header data stay at the same place followed by payload data indicated by their ‘wr_ena’ signals.

6.5.4 Out Switch

A building block of the output switch module is given on the Figure 21.

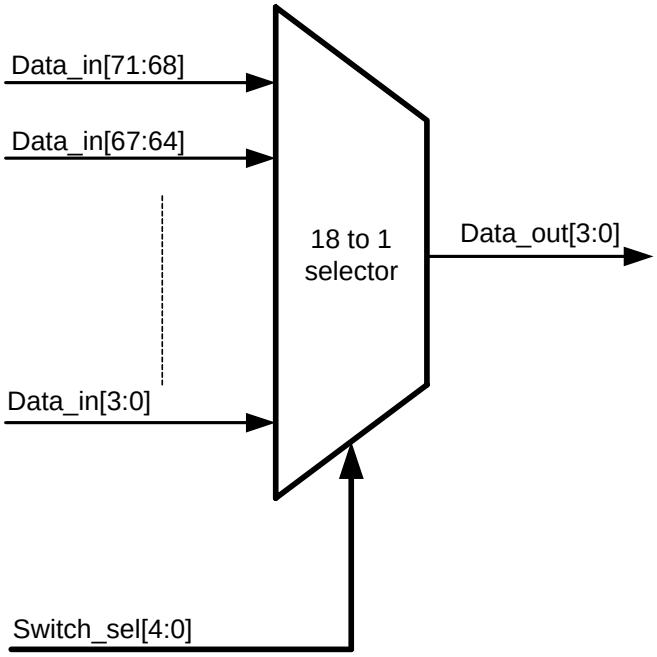


Figure 21 18 to 1 selections multiplexer

The multiplexer selects one 4-bits data stream out of the 18 available. The selection is performed based on the switch selection lines provided by the CMIB. This basic block is instantiated 18 times to enable 18 to 18 selections. There are only 18 valid selections for any individual multiplexer. Switch select value of 00000 binary selects data_in[3:0], 10001 selects data_in[71:68]. In case that the selection does not make sense the multiplexer selects data_in[3:0]. There is a bit within the register set that reports this faulty condition.

Second function performed within the block is data format conversion. Data rate is changes from 256MHz to 128MHz. To maintain the same data flow the data bus has to be double width. The operation is imposed by the high-speed serializer. This is shown on the Figure 22

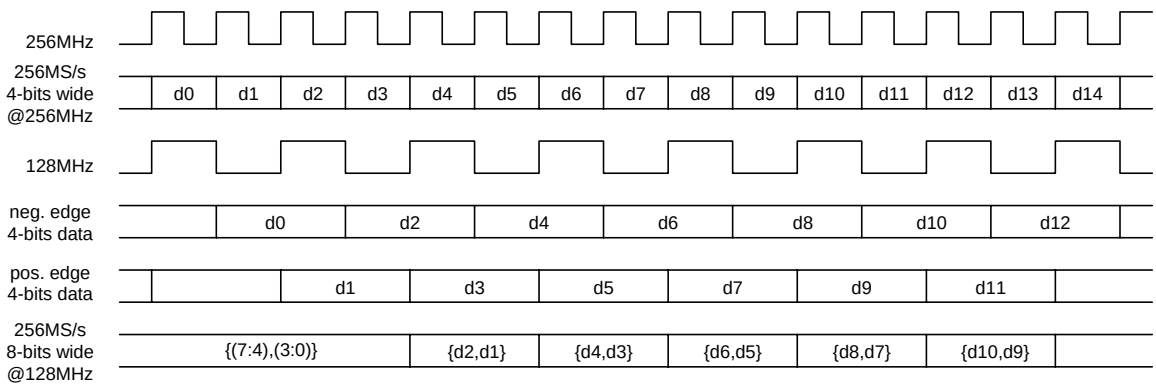


Figure 22 Output Switch data width conversion

7 MCB Interface Register Map

The register map of the SBOF is shown below.

Address	Mnemonic	Register Name	Access	Reset Value
00h	FVR	DM FPGA Version/Revision	R	0001h
01h	RBT	Read Back Test register	R/W	0000h
02h	MC	Main Control register	R/W	0000h
03h	STAT	Status register	R	0000h
04h	PWL	Pulse Width Low register	R/W	0000h
05h	PWH	Pulse Width High register	R/W	0000h
06h	PMPL	Pulse Mode Period Low register	R/W	0000h
07h	PMPH	Pulse Mode Period High register	R/W	0000h
08h	MCBG	MCB Gate register	R/W	0000h
09h	RM	Radar Mode register	R/W	0000h
0Ah	RAD	Radar Data register	R	0000h
0Bh	RDRP	Radar Data Read Pointer	R	0000h
0Ch	IPL	Input Ports Low register	R/W	0000h
0Dh	IPH	Input Ports High register	R/W	0000h
0Eh	ITCL	Input Tick Counter Low register	R	0000h
0Fh	ITCH	Input Tick Counter High register	R	0000h
		CRC Selection Registers		
10h	CRCIS1	CRC Input Selection 1 register	R/W	0000h
11h	CRCIS2	CRC Input Selection 2 register	R/W	0000h

12h	CRCIS3	CRC Input Selection 3 register	R/W	0000h
13h	CRCOS1	CRC Output Selection 1 register	R/W	0000h
14h	CRCOS2	CRC Output Selection 2 register	R/W	0000h
15h	CRCOS3	CRC Output Selection 3 register	R/W	0000h
16h	ERFO1	Error Force 1 register	R/W	0000h
17h	ERFO2	Error Force 2 register	R/W	0000h
		Output Switch Registers		
18h	OSWS1	Output Switch Select 1 register	R/W	0000h
19h	OSWS2	Output Switch Select 2 register	R/W	0000h
1Ah	OSWS3	Output Switch Select 3 register	R/W	0000h
1Bh	OSWS4	Output Switch Select 4 register	R/W	0000h
1Ch	OSWS5	Output Switch Select 5 register	R/W	0000h
1Dh	OSWS6	Output Switch Select 6 register	R/W	0000h
		CRC Input Value Registers		
1Eh	CRCIV1	CRC Input Value 1. Register	R	0000h
1Fh	CRCIV2	CRC Input Value 2. Register	R	0000h
20h	CRCIV3	CRC Input Value 3. Register	R	0000h
21h	CRCIV4	CRC Input Value 4. Register	R	0000h
22h	CRCIV5	CRC Input Value 5. Register	R	0000h
		CRC Output Value Registers		
23h	CRCOV1	CRC Output Value 1. Register	R	0000h
24h	CRCOV2	CRC Output Value 2. Register	R	0000h
25h	CRCOV3	CRC Output Value 3. Register	R	0000h

26h	CRCOV4	CRC Output Value 4. Register	R	0000h
27h	CRCOV5	CRC Output Value 5. Register	R	0000h
		Data Info Registers		
28h	DI0	Data Info 0. Register	R/W	0000h
29h	DI1	Data Info 1. Register	R/W	0000h
2ah	DI2	Data Info 2. Register	R/W	0000h
2bh	DI3	Data Info 3. Register	R/W	0000h
2ch	DI4	Data Info 4. Register	R/W	0000h
2dh	DI5	Data Info 5. Register	R/W	0000h
2eh	DI6	Data Info 6. Register	R/W	0000h
2fh	DI7	Data Info 7. Register	R/W	0000h
30h	DI8	Data Info 8. Register	R/W	0000h
31h	DI9	Data Info 9. Register	R/W	0000h
32h	DI10	Data Info 10. Register	R/W	0000h
33h	DI11	Data Info 11. Register	R/W	0000h
34h	DI12	Data Info 12. Register	R/W	0000h
35h	DI13	Data Info 13. Register	R/W	0000h
36h	DI14	Data Info 14. Register	R/W	0000h
37h	DI15	Data Info 15. Register	R/W	0000h
38h	DI16	Data Info 16. Register	R/W	0000h
39h	DI17	Data Info 17. Register	R/W	0000h
		Phase Registers		
3Ah	PL0	Phase Low 0. Register	R/W	0000h

3Bh	PH0	Phase High 0. Register	R/W	0000h
3Ch	PL1	Phase Low 1. Register	R/W	0000h
3Dh	PH1	Phase High 1. Register	R/W	0000h
3Eh	PL2	Phase Low 2. Register	R/W	0000h
3Fh	PH2	Phase High 2. Register	R/W	0000h
40h	PL3	Phase Low 3. Register	R/W	0000h
41h	PH3	Phase High 3. Register	R/W	0000h
42h	PL4	Phase Low 4. Register	R/W	0000h
43h	PH4	Phase High 4. Register	R/W	0000h
44h	PL5	Phase Low 5. Register	R/W	0000h
45h	PH5	Phase High 5. Register	R/W	0000h
46h	PL6	Phase Low 6. Register	R/W	0000h
47h	PH6	Phase High 6. Register	R/W	0000h
48h	PL7	Phase Low 7. Register	R/W	0000h
49h	PH7	Phase High 7. Register	R/W	0000h
4Ah	PL8	Phase Low 8. Register	R/W	0000h
4Bh	PH8	Phase High 8. Register	R/W	0000h
4Ch	PL9	Phase Low 9. Register	R/W	0000h
4Dh	PH9	Phase High 9. Register	R/W	0000h
4Eh	PL10	Phase Low 10. Register	R/W	0000h
4Fh	PH110	Phase High 10. Register	R/W	0000h
50h	PL11	Phase Low 11. Register	R/W	0000h
51h	PH11	Phase High 11. Register	R/W	0000h

52h	PL12	Phase Low 12. Register	R/W	0000h
53h	PH12	Phase High 12. Register	R/W	0000h
54h	PL13	Phase Low 13. Register	R/W	0000h
55h	PH13	Phase High 13. Register	R/W	0000h
56h	PL14	Phase Low 14. Register	R/W	0000h
57h	PH14	Phase High 14. Register	R/W	0000h
58h	PL15	Phase Low 15. Register	R/W	0000h
59h	PH15	Phase High 15. Register	R/W	0000h
5Ah	PL16	Phase Low 16. Register	R/W	0000h
5Bh	PH16	Phase High 16. Register	R/W	0000h
5Ch	PL17	Phase Low 17. Register	R/W	0000h
5Dh	PH17	Phase High 17. Register	R/W	0000h
5Eh	STDR	System Tick Delay Register	R/W	0000h
5Fh	CSRR	Chopper Seed Root Register	R/W	0000h
60h	STDR_MS	System Tick Delay Register – MS bits	R/W	0000h

Table 1 SBOF Memory Map

7.1 FPGA Version/Revision Number register – FVR

Address	**15:8**	**7:4**	**3:0**	**Access**	**Reset**
0x00	Design	Version	Revision	R	0x0001

Table 2 FPGA Version/Revision – FVR register

This register specifies the FPGA Version/Revision.

The fields within the register are defined as follows:

Design [15:8]: The design could be different for different purposes.

Version [7:4]: This version number is to be incremented whenever the new FPGA load is incompatible with the previous version from the software standpoint and would require change to the driver. The initial version will be numbered 1(b0001). Version 0 is reserved for prototyping and proof of concept.

Revision [3:0]: This revision number is to be incremented whenever the new FPGA load incorporates new functionality but does not require software upgrades.

7.2 Read Back register – RBT

Address	**15:0**	**Access**	**Reset**
0x01	Read Back	R/W	0x0000

Table 3 Read Back - RBT register

This is a test register. Writes to this register will have no effect. It simply provides means for communication verification between the FPGA and the MCB. The register keeps its last value until next write to its location.

The fields within the register are defined as follows:

Read Back [15:0]: any 16 bits long combination of zeros and ones.

7.3 Main Control register – MC

Address	6	5	4	3	2	1	0	Access	Reset
0x02	ST	HI	PG	GM	SL	R	SR	R/W	0x0000

Address	15	14:7	Access	Reset
0x02	PR	R	R/W	0x0000

Table 4 Main Control – MC register

This is a main control register. The fields within the register are defined as follows:

SR[0]: Software Reset, this bit resets the FPGA bringing it to its default state. After removing the address, 0x02, it resets itself as well.

1: resets the FPGA.

0: does not reset the FPGA.

SL[2]: Self Lock, this bit enables self locking mode when the FPGA processes pulsar data. After setting all the necessary parameters for each channel the MCB can assert this bit and the logic will be taking internally calculated values on each tick rather than ones delivered thru the register set. The bit is not settable on a channel basis; its assertion is applicable to all 18 narrow band channels.

1: self lock is enabled.

0: self lock is not enabled.

GM[3]: Gate Mode enables one of two modes of operation. When asserted it enables non-pulsar mode of operation.

1: Gate Mode of operation is enabled.

0: Gate Mode of operation is not enabled.

PM[4]: Pulsar Mode sets pulsar mode of operation.

1: pulsar mode of operation is enabled.

0: pulsar mode of operation is not enabled.

Combination of the two last bits, [PGE, MGE], produces the following:

[0,0] = No valid data are coming out of the FPGA.

[0,1] = Enables non-pulsar mode of operation.

[1,0] = Enables pulsar mode of operation.

[1,1] = Enables pulsar mode of operation.

HI[5]: Header Insertion (see 6.5.1).

1: enable data info embedding (default) at the next 10 millisecond tick.

0: disable data info embedding at the next 10 millisecond tick.

ST[6]: System Tick select**,** selects DDR system tick stream.

1: rising clock edge data stream is selected.

0: falling clock edge data stream is selected, default selection.

PR[15]: SCLK PLL reset.

1: PLL is held in reset.

0: PLL operates normally.

7.4 Status register – STAT

Address	**15:5**	**4**	**3**	**2**	**1**	**0**	**Access**	**Reset**
0x03	Unused	DWD	TC	RS	OS	PL	R	0x0000

Table 5 Status – STAT register

This register reports the status of the FPGA. The fields within it are defined as follows:

PL[0]: PLL Locked, reports the status of the system PLL.

1: The PLL is locked to the incoming system clock.

0: The PLL is not locked to the incoming system clock.

OS[1]: Output Switch, reports status of the output switch configuration. The Output Switch consists of eighteen 18X1 4-bits selection multiplexers. This concept provides maximum flexibility in the outputs selection; any input data channel can appear at any output. The number of selection bits for each selection multiplexer is five. The improper value for the selection bits would be everything greater than 17 i.e. (10001 binary) because there are only 18 inputs to select from. In this case, the input channel zero is selected as an output. The condition is reported as a sum for all of the 18 multiplexers. The bit stays asserted as long as the faulty condition exists, if any.

1: there is at least one output multiplexer improperly configured.

0: all the output multiplexers are properly configured.

RS[2]: Radar Switch, reports status of the radar switch configuration. It does the same job as the previous bit, OS[2], for the radar selection multiplexer. There is only one selection multiplexer that selects which one out of 18 narrow band data streams will be stored into the local memory. The contents of the memory are available to the MCB interface.

1: signals improper multiplexer configuration, input channel 0 is selected.

0: signals proper multiplexer configuration.

TC[3]: Timing Control. This bit serves to inform the MCB about the status of the control signal that is coming from the Timing FPGA and enables info data embedding.

1: data embedding is permitted by the Timing FPGA.

0: data embedding is not permitted by the Timing FPGA.

DWD[4]: Double Width Detector bit for system tick, 100PPS. The bit reports an error regarding the pulse duration, which is 4ns. If the pulse lasts 8ns or more this bit goes high. This bit should be read by the ISR.

1: system tick, 100PPS, duration is more than 4ns (error).

0: system tick, 100PPS, duration is 4ns.

7.5 Pulse Width registers – PWL and PWH

Address	15:0	Access	Reset

0x04	Pulse Width lower part	R/W	0x0000

Table 6 Pulse Width lower part – PWL register

Address	15:6	5:0	Access	Reset
0x05	Unused	Pulse Width higher part	R/W	0x0000

Table 7 Pulse Width higher part – PWH register

The above two registers define the pulse width in the pulsar mode of operation. The pulse width is given as a number of microseconds. The 22-bits number ranges from zero to 0x3FFFFF (0 to 4,194,303.00). This defines the maximum pulse width of 4.194,303 seconds. Writing 0x0003E8 (1000 decimal) to the register defines pulse width of 1000μs or 1ms. During the pulse, the data is declared as valid to the downstream hardware.

7.6 Pulse Mode Period registers – PMPL and PMPH

Address	15:0	Access	Reset
0x06	Pulse Period lower part	R/W	0x0000

Table 8 Pulse Period lower part - PMPL register

Address	15:6	5:0	Access	Reset
0x07	Unused	Pulse Period higher part	R/W	0x0000

Table 9 Pulse Period higher part - PMPH register

The above two registers represent the pulse mode period. The period is given as a number of microseconds. The 22-bits number ranges from zero to 0x3FFFFF (0 to 4,194,303.00). This defines the maximum pulsar period of 4.194,303 seconds. Writing 0x0003E8 (1000 decimal) to the register defines pulsar period of 1000μs or 1ms.

7.7 MCB Gate register – MCBG

Address	15:0	Access	Reset
0x08	MCB Gate	R/W	0x0000

Table 10 MCB Gate – MCBG register

The register defines so called MCB Gate duration. During initial mode changes on the system level, the incoming data have to be invalidated. For long period, the CMIB can simply disable both modes of operation by writing zeros on appropriate locations for MGE and PGE as explained above. In case that valid signal should transition from low to high value within 10ms interval, between the ticks, this register can be used to define the moment. The 10ms interval is divided into 64,000.00 156.25ns sub intervals. This number represents 40 256MHz clock periods. In case that the valid is low and before the tick comes the CMIB asserts 0x0007F that will cause valid to go high 156.25ns X 127 = 19,843.75ns after the tick. This assumes that MGE bit is already high. On the next tick, the CMIB should change the value to zero or this will be repeated. To keep the value bit low during the entire 10ms interval, when MGE is high, CMIB should write 64,001 or greater number to the register.

7.8 Radar Mode register – RM

Address	11	10	9:5	4:0	Access	Reset
0x09	R	RWE	R	RCS	R/W	0x0000

Table 11 Radar Mode – RM register

Address	15:8	Access	Reset
0x09	R	R/W	0x0000

Table 12 Radar Mode – RM register

This register determines radar mode of operation. This mode does not conflict with the previous two modes. Any of 18 narrow band data streams could be stored into the internal memory and read by the CMIB after next tick. The fields within the register are defined as follows:

RCS[4:0]: radar channel selection, defines which one out of 18 narrow band channels is selected for writing to the local memory.

00000 selects input channel 0.

00001 selects input channel 1.

.

.

10001 selects input channel 17.

10010 and greater select channel 0 and status bit that reports this faulty condition goes high.

RWE[10]: radar mode write enable, enables writing to the radar memory block.

1: writing to the memory is enabled.

0: writing to the memory is not enabled.

R: Reserved.

7.9 Radar Data register – RAD

Address	15:0	Access	Reset
0x0A	Radar Data	R	0x0000

Table 13 Radar Data – RAD register

This register is used by the CMIB to read data stored in the internal memory when operating in the radar mode. Asserting 0x0A address and setting the FPGA in read mode of operation will enable internal read addresses generation and the memory data appears on the data bus word by word at 33MHz rate.

7.10 Radar Data Read Pointer register – RDRP

Address	15:0	Access	Reset
0x0B	Read Pointer	R	0x0000

Table 14 Radar Data Read pointer – RDRP register

This register provides info regarding the internal radar mode read pointer, read address within the radar mode block.

7.11 Input Ports Low and High registers – IPL and IPH

Address	15:0	Access	Reset
0x0C	HLS[15:0]	R/W	0x0000

Table 15 Input Ports Low – IPL register

This register contains first 16 bits for DDR stream selection for the first 16 ports. Incoming data are clocked in by the internal 256MHz clock signal in DDR mode. This method produces two data streams, one obtained by the rising clock edge and the other by the falling clock edge. The data stream selection will be determined during Station Board prototype testing for each port. The bits within the register are defined as follows:

HLS[0]: High Low Select**,** selects DDR data stream within port 0.

1: rising clock edge data stream is selected.

0: falling clock edge data stream is selected, default selection.

The rest of the bits are defined in the same manner for the remaining 15 ports.

Address	15:10	9:5	4:2	1:0	Access	Reset
0x0D	Unused	IPS	ICC	HLS[17:16]	R/W	0x0000

Table 16 Input Ports High – IPH register

The bits within the register are defined as follows:

HLS[1:0]: High Low Select; DDR stream selection for the remaining top two ports.

ICC[2:0]: Input Counters Control; 3 bits that control operation of the two counters within Sample Checker block. The bits are defined as follows:

ICC[0]: Counter Select; select which one out of the two available counters within the Sample Checker module will be visible to the CMIB.

1: selects Sind Counter.

0: selects Tick Counter.

ICC[2:1]: Selects start and stop tick combinations.

00: tick to stick

01: tick to tick

10: stick to stick

11**:** stick to tick

IPS[4:0]: Input Port Selection; selects input port.

7.12 Input Tick Counter Low and High – ITCL and ITCH

Address	15:0	Access	Reset
0x0E	ITC[15:0]	R	0x0000

Table 17 Input Tick Counter Low – ITCL register

This register defines first 16 out of 22 bits of the Sample Checker counter.

Address	15:6	5:0	Access	Reset
0x0F	Unused	ITC[21:16]	R	0x0000

Table 18 Input Tick Counter High – ITCH register

The bits within the register are defined as follows:

ITC[5:0]: remaining six bits of the Sample Checker counter described above.

7.13 CRC Input Selection registers – CRCIS1 to CRCIS3

Address	15:0	Access	Reset
0x10	CRCIS1	R/W	0x0000

Table 19 CRC Input Selection 1 - CRCIS1 register

The register defines on which data line, one out of four, to perform the CRC calculation within the first group of eight input ports. The bits within the register are defined as follows:

CRCIS1[1:0]: selects the bit within input data port zero.

CRCIS1[3:2]: selects the bit within input data port one.

.

.

CRCIS1[15:14]: selects the bit within input data port seven.

Address	15:0	Access	Reset
0x11	CRCIS2	R/W	0x0000

Table 20 CRC Input Selection 2 – CRCIS2 register

The register defines on which data line, one out of four, to perform the CRC calculation within the second group of eight input ports. The bits within the register are defined as follows:

CRCIS2[1:0]: selects the bit within input data port eight.

CRCIS2[3:2]: selects the bit within input data port nine.

.

.

CRCIS2[15:14]: selects the bit within input data port fifteen.

Address	15:5	4	3:0	Access	Reset
0x12	Unused	ODD	CRCIS3	R/W	0x0000

Table 21 CRC Input Selection 3 – CRCIS3 register

This register defines CRC calculation for the last two input ports. The bits within the register are defined as follows:

CRCIS3[1:0]: selects the bit within input data port sixteen.

CRCIS3[3:2]: selects the bit within input data port seventeen.

ODD[4]: selects SIND signal for CRC checking in 18 input ports.

1: Replaces data[0] with SIND signal.

0: Does not replace data[0] with SIND signal.

7.14 CRC Output Selection registers – CRCOS1 to CRCOS3 registers

Address	15:0	Access	Reset
0x13	CRCOS1	R/W	0x0000

Table 22 CRC Output Selection 1 - CRCOS1 register

The CRC is calculated on each output data wire and is included within the header. The CMIB can select to read one out of four values within each port. The bits within the register are defined as follows:

CRCOS1[1:0]: selects bit within data port zero.

CRCOS1[3:2]: selects bit within data port one.

.

.

CRCOS1[15:14]: selects bit within data port seven.

Address	15:0	Access	Reset
0x14	CRCOS2	R/W	0x0000

Table 23 CRC Output Selection 2 – CRCOS2 register

The CRC is calculated on each output data wire. The CMIB can select to read one out of four values within each port. The bits within the register are defined as follows:

CRCOS2[1:0]: selects bit within data port eight.

CRCOS2[3:2]: selects bit within data port nine.

.

.

CRCOS2[15:14]: selects bit within data port fifteen.

Address	15:4	3:0	Access	Reset
0x15	Unused	CRCOS3	R/W	0x0000

Table 24 CRC Output Selection 3 – CRCOS3 register

This register defines CRC calculation for the last two output ports. The bits within the register are defined as follows:

CRCOS3[1:0]: selects the bit within output data port sixteen.

CRCOS3[3:2]: selects the bit within output data port seventeen.

7.15 Error Force registers – ERFO1 and ERFO2

Address	15:0	Access	Reset
0x16	ERFO1	R/W	0x0000

Table 25 Error Force 1 – ERFO1 register

This register is used for testing purposes only. Each bit within it forces erroneous output CRC calculation within a port. When the bit is asserted the transmitted CRC value is actually inverted real one. The bits are defined as follows:

ERFO1[0]: forces output CRC error in output port zero, all four wires.

ERFO1 [1]: forces output CRC error in output port one, all four wires.

The rest of the bits are defined in the same manner.

Address	15:2	1:0	Access	Reset
0x17	Unused	ERFO2	R/W	0x0000

Table 26 Error Force 2 – ERFO2 register

This register has the same function as the previous one. Only the last two bits are used to force errors on channels 16 bit ERFO2[0] and 17 ERFO2[1]. The rest is not used.

7.16 Output Switch Selection registers – OSWS1 to OSWS6

Address	15	14:10	9:5	4:0	Access	Reset
0x18	Unused	OSW2	OSW1	OSW0	R/W	0x0000

Table 27 Output Switch Selections 1 - OSWS1 register

The bits within the register are defined as follows:

OSW0[4:0]: Output Switch 0; selects output for the output channel zero.

OSW1[4:0]: Output Switch 1; selects output for the output channel one.

OSW2[4:0]: Output Switch 2; selects output for the output channel two.

Address	15	14:10	9:5	4:0	Access	Reset
0x19	Unused	OSW5	OSW4	OSW3	R/W	0x0000

Table 28 Output Switch Selections 2 – OSWS2 register

The bits within the register are defined as follows:

OSW3[4:0]: Output Switch 3; selects output for the output channel three.

OSW4[4:0]: Output Switch 4; selects output for the output channel four.

OSW5[4:0]: Output Switch 5; selects output for the output channel five.

Address	15	14:10	9:5	4:0	Access	Reset
0x1A	Unused	OSW8	OSW7	OSW6	R/W	0x0000

Table 29 Output Switch Selections 3 – OSWS3 register

The bits within the register are defined as follows:

OSW6[4:0]: Output Switch 6; selects output for the output channel six.

OSW7[4:0]: Output Switch 7; selects output for the output channel seven.

OSW8[4:0]: Output Switch 8; selects output for the output channel eight.

Address	15	14:10	9:5	4:0	Access	Reset
0x1B	Unused	OSW11	OSW10	OSW9	R/W	0x0000

Table 30 Output Switch Selections 4 – OSWS4 register

The bits within the register are defined as follows:

OSW9[4:0]: Output Switch 9; selects output data for the output channel nine.

OSW10[4:0]: Output Switch 10; selects output data for the output channel ten.

OSW11[4:0]: Output Switch 11; selects output data for the output channel eleven.

Address	15	14:10	9:5	4:0	Access	Reset
0x1C	Unused	OSW14	OSW13	OSW12	R/W	0x0000

Table 31 Output Switch Selections 5 – OSWS5 register

The bits within the register are defined as follows:

OSW12[4:0]: Output Switch 12; selects output data for the output channel twelve.

OSW13[4:0]: Output Switch 13; selects output data for the output channel thirteen.

OSW14[4:0]: Output Switch 14; selects output data for the output channel fourteen.

Address	15	14:10	9:5	4:0	Access	Reset
0x1D	Unused	OSW17	OSW16	OSW15	R/W	0x0000

Table 32 Output Switch Selections 6 – OSWS6 register

The bits within the register are defined as follows:

OSW15[4:0]: Output Switch 12; selects output data for the output channel fifteen.

OSW16[4:0]: Output Switch 13; selects output data for the output channel sixteen.

OSW17[4:0]: Output Switch 14; selects output data for the output channel seventeen.

7.17 CRC Input Value registers – CRCIV1 to CRCIV5

Address	15:12	11:8	7:4	3:0	Access	Reset
0x1E	CRCIP3	CRCIP2	CRCIP1	CRCIP0	R	0x0000

Table 33 CRC Input Value 1 - CRCIV1 register

The bits within the register are defined as follows:

CRCIP0 [3:0]: CRC calculated value for the selected wire within input port 0.

CRCIP1 [3:0]: CRC calculated value for the selected wire within input port 1.

CRCIP2 [3:0]: CRC calculated value for the selected wire within input port 2.

CRCIP3 [3:0]: CRC calculated value for the selected wire within input port 3.

The next register, **CRCIV2**, at the address **0x1F** provides the CRC info for the next four input ports (CRCIP4, CRCIP5, CRCIP6 and CRCIP7) in the same manner as the previous one.

The next register, **CRCIV3**, at the address **0x20** provides the CRC info for the next four input ports (CRCIP8, CRCIP9, CRCIP10 and CRCIP11) in the same manner as the previous one.

The next register, **CRCIV4**, at the address **0x21** provides the CRC info for the next four input ports (CRCIP12, CRCIP13, CRCIP14 and CRCIP15) in the same manner as the previous one.

Address	15:8	7:4	3:0	Access	Reset
0x22	Unused	CRCIP17	CRCIP16	R	0x0000

Table 34 CRC Input Value 5 – CRCIV5 register

The bits within the register are defined as follows:

CRCIP16 **[3:0]:** CRC calculated value for the selected wire within input port 16.

CRCIP17 [3:0]: CRC calculated value for the selected wire within input port 17.

7.18 CRC Output Value registers – CRCOV1 to CRCOV5

Address	15:12	11:8	7:4	3:0	Access	Reset
0x23	CRCOP3	CRCOP2	CRCOP1	CRCOP0	R	0x0000

Table 35 CRC Output Value 1 - CRCOV1 register

The bits within the register are defined as follows:

CRCOP0 [3:0]: CRC calculated value for the selected wire within output port 0.

CRCOP1 [3:0]: CRC calculated value for the selected wire within output port 1.

CRCOP2 [3:0]: CRC calculated value for the selected wire within output port 2.

CRCOP3 [3:0]: CRC calculated value for the selected wire within output port 3.

The next register, **CRCOV2**, at the address **0x24** provides the CRC info for the next four output ports (CRCOP4, CRCOP5, CRCOP6 and CRCOP7) in the same manner as the previous one.

The next register, **CRCOV3**, at the address **0x25** provides the CRC info for the next four output ports (CRCOP8, CRCOP9, CRCOP10 and CRCOP11) in the same manner as the previous one.

The next register, **CRCOV4**, at the address **0x26** provides the CRC info for the next four output ports (CRCOP12, CRCOP13, CRCOP14 and CRCOP15) in the same manner as the previous one.

Address	15:8	7:4	3:0	Access	Reset
0x27	Unused	CRCOP17	CRCOP16	R	0x0000

Table 36 CRC Output Value 5 – CRCOV5 register

The bits within the register are defined as follows:

CRCOP16 [3:0]: CRC calculated value for the selected wire within output port 16.

CRCOP17 [3:0]: CRC calculated value for the selected wire within output port 17.

7.19 Data Info registers – DI0 to DI17

Address	15:13	12:8	7:0	Access	Reset
0x28	BBID	SBID	SID	R/W	0x0000

Table 37 Data Info 0 – DI0 register

The bits within the register are defined as follows:

SID[7:0]: Station Identification.

SBID[4:0]: Sub-Band Identification.

BBID[2:0]: Base Band Identification.

This info data applies to the port 0 data stream. In the same way the rest of the ports are defined. Data info for port 1 is defined by the register at address 0x29, data info for port 2 is defined by the register at address 0x2a, … and data info for port 17 is defined by the register at address 0x39.

7.20 Phase registers – PL0 to PL17 and PH0 to PH17

Address	15:0	Access	Reset
0x3A	PHASE[15:0]	R/W	0x0000

Table 38 Phase Low 0 – PL0 register

This register defines first 16 bits, out of 22, for the port's zero phase value when in the pulsar mode of operation.

Address	15:8	7	6	5:0	Access	Reset
0x3B	Unused	HI	U.	PHASE[21:16]	R/W	0x0000

Table 39 Phase High 0 – PH0 register

The bits within the register are defined as follows:

PHASE[5:0]: Remaining 6 bits for the port zero phase value.

HI[7]: Hard invalid; invalidates data that leave port zero.

1: data from port zero are being invalidated.

0: data from port zero are not being invalidated.

Bits PH0[15:8] and PH0[6] are unused.

The phase value and control bits are defined in the same manner for the rest of the ports. Addresses 0x3C and 0x3D define port one, addresses 0x3E and 0x3F define port two…and addresses 0x5C and 05D define port seventeen.

7.21 System Tick Delay register – STDR

Address	15:0	Access	Reset
0x5E	STDR[15:0]	R/W	0x0000

Table 40 System Tick delay – STDR register

Defines for how many 256MHz clock cycles system tick is to be delayed after entering the FPGA.

7.22 Chopper Seed Root register – CSRR

This register sets the chopper seed root.

Address	15:8	7	6:4	3:1	0	Access	Reset
0x5F	R	CID	SID	R	FID	R/W	0x0000

Table 41 Chopper Seed Root – CSRR register

The bit assignments here are suggested. Any 8-bit value may be sent. The Output FPGA will add the wafer number (0 to 17) to whatever is sent and will protect against the result being 0xFF. The aim here is to provide as many different chopping waveforms to the RXP FPGAs on the baseline board. The same values for the seed root must be set in the Timing FPGA for the scheme to work.

CID[7] Crate ID (0 or 1)

SID[6:4] Slot ID (0 to 7)

FID[0] FPGA ID (0 or 1)

7.23 System Tick Delay register MS Bits – STDR_MS

Address	15:0	1:0	Access	Reset
0x60	Reserved	STDR[17:16]	R/W	0x0000

Table 42 System Tick delay MS bits – STDR_MS register

8 Pinouts, Pin Locations and Programming Notes

8.1 Pinout by signal name

Note: UNUSED and NC pins have been removed.

```
Quartus II Version 9.1 Build 304 01/25/2010 Service Pack 1 SJ Full Version
CHIP  "output_top"  ASSIGNED TO AN: EP1SGX10CF672C5

-----------------------------------------------------------------------------------------
Pin Name/Usage               : Location  : Dir.   : I/O Standard       : Voltage : I/O Bank  :
-----------------------------------------------------------------------------------------
clk_128                      : AE14      : input  : LVDS               :         : 8         :
clk_128(n)                   : AF14      : input  : LVDS               :         : 8         :
clock_out_p[0]               : B13       : output : LVDS               :         : 10        :
clock_out_p[0](n)            : A13       : output : LVDS               :         : 10        :
clock_out_p[1]               : F14       : output : LVDS               :         : 10        :
clock_out_p[1](n)            : E14       : output : LVDS               :         : 10        :
clock_out_p[2]               : C12       : output : LVDS               :         : 9         :
clock_out_p[2](n)            : B12       : output : LVDS               :         : 9         :
clock_out_p[3]               : D13       : output : LVDS               :         : 9         :
clock_out_p[3](n)            : C13       : output : LVDS               :         : 9         :
clock_out_p[4]               : H19       : output : LVDS               :         : 2         :
clock_out_p[4](n)            : H20       : output : LVDS               :         : 2         :
clock_out_p[5]               : AE13      : output : LVDS               :         : 12        :
clock_out_p[5](n)            : AF13      : output : LVDS               :         : 12        :
clock_out_p[6]               : AA14      : output : LVDS               :         : 12        :
clock_out_p[6](n)            : AB14      : output : LVDS               :         : 12        :
clock_out_p[7]               : AA13      : output : LVDS               :         : 11        :
clock_out_p[7](n)            : AB13      : output : LVDS               :         : 11        :
clock_out_p[8]               : AC13      : output : LVDS               :         : 11        :
```

clock_out_p[8](n)	AD13	output	LVDS		11
data_01[0]	AC4	input	1.5-V HSTL Class I		7
data_01[1]	AD4	input	1.5-V HSTL Class I		7
data_01[2]	AE4	input	1.5-V HSTL Class I		7
data_01[3]	AF4	input	1.5-V HSTL Class I		7
data_02[0]	AD5	input	1.5-V HSTL Class I		7
data_02[1]	AC5	input	1.5-V HSTL Class I		7
data_02[2]	AC6	input	1.5-V HSTL Class I		7
data_02[3]	AD6	input	1.5-V HSTL Class I		7
data_03[0]	AF7	input	1.5-V HSTL Class I		7
data_03[1]	AE7	input	1.5-V HSTL Class I		7
data_03[2]	AD7	input	1.5-V HSTL Class I		7
data_03[3]	AC7	input	1.5-V HSTL Class I		7
data_04[0]	AD8	input	1.5-V HSTL Class I		7
data_04[1]	AC8	input	1.5-V HSTL Class I		7
data_04[2]	AC9	input	1.5-V HSTL Class I		7
data_04[3]	AB9	input	1.5-V HSTL Class I		7
data_05[0]	AE9	input	1.5-V HSTL Class I		7
data_05[1]	AE10	input	1.5-V HSTL Class I		7
data_05[2]	AE11	input	1.5-V HSTL Class I		7
data_05[3]	AD9	input	1.5-V HSTL Class I		7
data_06[0]	AB7	input	1.5-V HSTL Class I		7
data_06[1]	AB8	input	1.5-V HSTL Class I		7
data_06[2]	AA8	input	1.5-V HSTL Class I		7
data_06[3]	Y8	input	1.5-V HSTL Class I		7
data_07[0]	W7	input	1.5-V HSTL Class I		7
data_07[1]	W8	input	1.5-V HSTL Class I		7
data_07[2]	U8	input	1.5-V HSTL Class I		7
data_07[3]	U9	input	1.5-V HSTL Class I		7
data_08[0]	AE17	input	1.5-V HSTL Class I		8
data_08[1]	AE18	input	1.5-V HSTL Class I		8
data_08[2]	AD17	input	1.5-V HSTL Class I		8
data_08[3]	AD18	input	1.5-V HSTL Class I		8
data_09[0]	AD15	input	1.5-V HSTL Class I		8

data_09[1]	AD16	input	1.5-V HSTL Class I		8
data_09[2]	AC15	input	1.5-V HSTL Class I		8
data_09[3]	AC16	input	1.5-V HSTL Class I		8
data_10[0]	AE19	input	1.5-V HSTL Class I		8
data_10[1]	AE20	input	1.5-V HSTL Class I		8
data_10[2]	AD19	input	1.5-V HSTL Class I		8
data_10[3]	AD20	input	1.5-V HSTL Class I		8
data_11[0]	AE21	input	1.5-V HSTL Class I		8
data_11[1]	AE22	input	1.5-V HSTL Class I		8
data_11[2]	AD21	input	1.5-V HSTL Class I		8
data_11[3]	AD22	input	1.5-V HSTL Class I		8
data_12[0]	AB17	input	1.5-V HSTL Class I		8
data_12[1]	AB18	input	1.5-V HSTL Class I		8
data_12[2]	AA17	input	1.5-V HSTL Class I		8
data_12[3]	AA18	input	1.5-V HSTL Class I		8
data_13[0]	C15	input	1.5-V HSTL Class I		3
data_13[1]	B16	input	1.5-V HSTL Class I		3
data_13[2]	B17	input	1.5-V HSTL Class I		3
data_13[3]	A17	input	1.5-V HSTL Class I		3
data_14[0]	B18	input	1.5-V HSTL Class I		3
data_14[1]	B19	input	1.5-V HSTL Class I		3
data_14[2]	C18	input	1.5-V HSTL Class I		3
data_14[3]	C19	input	1.5-V HSTL Class I		3
data_15[0]	F15	input	1.5-V HSTL Class I		3
data_15[1]	E16	input	1.5-V HSTL Class I		3
data_15[2]	D16	input	1.5-V HSTL Class I		3
data_15[3]	C16	input	1.5-V HSTL Class I		3
data_16[0]	E17	input	1.5-V HSTL Class I		3
data_16[1]	E18	input	1.5-V HSTL Class I		3
data_16[2]	D18	input	1.5-V HSTL Class I		3
data_16[3]	D19	input	1.5-V HSTL Class I		3
data_17[0]	B20	input	1.5-V HSTL Class I		3
data_17[1]	B21	input	1.5-V HSTL Class I		3
data_17[2]	C20	input	1.5-V HSTL Class I		3

data_17[3]	C21	input	1.5-V HSTL Class I		3
data_18[0]	B22	input	1.5-V HSTL Class I		3
data_18[1]	B23	input	1.5-V HSTL Class I		3
data_18[2]	C22	input	1.5-V HSTL Class I		3
data_18[3]	B24	input	1.5-V HSTL Class I		3
data_out_p[0]	J19	output	LVDS		2
data_out_p[0](n)	J20	output	LVDS		2
data_out_p[1]	J21	output	LVDS		2
data_out_p[1](n)	J22	output	LVDS		2
data_out_p[10]	T19	output	LVDS		1
data_out_p[10](n)	T20	output	LVDS		1
data_out_p[11]	T21	output	LVDS		1
data_out_p[11](n)	T22	output	LVDS		1
data_out_p[12]	U21	output	LVDS		1
data_out_p[12](n)	U22	output	LVDS		1
data_out_p[13]	U19	output	LVDS		1
data_out_p[13](n)	U20	output	LVDS		1
data_out_p[14]	U17	output	LVDS		1
data_out_p[14](n)	U18	output	LVDS		1
data_out_p[15]	V21	output	LVDS		1
data_out_p[15](n)	V22	output	LVDS		1
data_out_p[16]	V19	output	LVDS		1
data_out_p[16](n)	V20	output	LVDS		1
data_out_p[17]	W19	output	LVDS		1
data_out_p[17](n)	W20	output	LVDS		1
data_out_p[2]	K19	output	LVDS		2
data_out_p[2](n)	K20	output	LVDS		2
data_out_p[3]	K21	output	LVDS		2
data_out_p[3](n)	K22	output	LVDS		2
data_out_p[4]	K17	output	LVDS		2
data_out_p[4](n)	K18	output	LVDS		2
data_out_p[5]	L21	output	LVDS		2
data_out_p[5](n)	L22	output	LVDS		2
data_out_p[6]	L19	output	LVDS		2

data_out_p[6](n)	L20	output	LVDS		2
data_out_p[7]	M21	output	LVDS		2
data_out_p[7](n)	M22	output	LVDS		2
data_out_p[8]	M19	output	LVDS		2
data_out_p[8](n)	M20	output	LVDS		2
data_out_p[9]	M17	output	LVDS		2
data_out_p[9](n)	M18	output	LVDS		2
mcb_addr_p[0]	C7	input	3.3-V LVTTL		4
mcb_addr_p[1]	C8	input	3.3-V LVTTL		4
mcb_addr_p[2]	C9	input	3.3-V LVTTL		4
mcb_addr_p[3]	D4	input	3.3-V LVTTL		4
mcb_addr_p[4]	D5	input	3.3-V LVTTL		4
mcb_addr_p[5]	D6	input	3.3-V LVTTL		4
mcb_addr_p[6]	D7	input	3.3-V LVTTL		4
mcb_addr_p[7]	D8	input	3.3-V LVTTL		4
mcb_clk_p	E12	input	3.3-V LVTTL		4
mcb_cs_p	B2	input	3.3-V LVTTL		4
mcb_data_p[0]	A4	bidir	3.3-V LVTTL		4
mcb_data_p[1]	A5	bidir	3.3-V LVTTL		4
mcb_data_p[10]	B7	bidir	3.3-V LVTTL		4
mcb_data_p[11]	B8	bidir	3.3-V LVTTL		4
mcb_data_p[12]	B9	bidir	3.3-V LVTTL		4
mcb_data_p[13]	C4	bidir	3.3-V LVTTL		4
mcb_data_p[14]	C5	bidir	3.3-V LVTTL		4
mcb_data_p[15]	C6	bidir	3.3-V LVTTL		4
mcb_data_p[2]	A6	bidir	3.3-V LVTTL		4
mcb_data_p[3]	A7	bidir	3.3-V LVTTL		4
mcb_data_p[4]	A8	bidir	3.3-V LVTTL		4
mcb_data_p[5]	A9	bidir	3.3-V LVTTL		4
mcb_data_p[6]	A10	bidir	3.3-V LVTTL		4
mcb_data_p[7]	B4	bidir	3.3-V LVTTL		4
mcb_data_p[8]	B5	bidir	3.3-V LVTTL		4
mcb_data_p[9]	B6	bidir	3.3-V LVTTL		4
mcb_rw_p	B3	input	3.3-V LVTTL		4

reset_p	C1	input	3.3-V LVTTL	4	
sind_01	AE2	input	1.5-V HSTL Class I	7	
sind_02	AF5	input	1.5-V HSTL Class I	7	
sind_03	AF6	input	1.5-V HSTL Class I	7	
sind_04	AF8	input	1.5-V HSTL Class I	7	
sind_05	AF9	input	1.5-V HSTL Class I	7	
sind_06	Y7	input	1.5-V HSTL Class I	7	
sind_07	V7	input	1.5-V HSTL Class I	7	
sind_08	AF17	input	1.5-V HSTL Class I	8	
sind_09	AE15	input	1.5-V HSTL Class I	8	
sind_10	AF19	input	1.5-V HSTL Class I	8	
sind_11	AF21	input	1.5-V HSTL Class I	8	
sind_12	AC17	input	1.5-V HSTL Class I	8	
sind_13	A14	input	1.5-V HSTL Class I	3	
sind_14	A18	input	1.5-V HSTL Class I	3	
sind_15	D15	input	1.5-V HSTL Class I	3	
sind_16	C17	input	1.5-V HSTL Class I	3	
sind_17	A20	input	1.5-V HSTL Class I	3	
sind_18	A22	input	1.5-V HSTL Class I	3	
stick	AB12	input	LVDS	7	
stick(n)	AC12	input	LVDS	7	
tc_cbit_p	D9	input	3.3-V LVTTL	4	
tc_out_p	E8	output	3.3-V LVTTL	4	
test_port[0]	E4	output	3.3-V LVTTL	4	
test_port[1]	E5	output	3.3-V LVTTL	4	
test_port[2]	E6	output	3.3-V LVTTL	4	
test_port[3]	E7	output	3.3-V LVTTL	4	
tick_01	AE3	input	1.5-V HSTL Class I	7	
tick_02	AE5	input	1.5-V HSTL Class I	7	
tick_03	AE6	input	1.5-V HSTL Class I	7	
tick_04	AE8	input	1.5-V HSTL Class I	7	
tick_05	AF10	input	1.5-V HSTL Class I	7	
tick_06	AA7	input	1.5-V HSTL Class I	7	
tick_07	V8	input	1.5-V HSTL Class I	7	

tick_08	AF18	input	1.5-V HSTL Class I		8
tick_09	AE16	input	1.5-V HSTL Class I		8
tick_10	AF20	input	1.5-V HSTL Class I		8
tick_11	AF22	input	1.5-V HSTL Class I		8
tick_12	AC18	input	1.5-V HSTL Class I		8
tick_13	B15	input	1.5-V HSTL Class I		3
tick_14	A19	input	1.5-V HSTL Class I		3
tick_15	E15	input	1.5-V HSTL Class I		3
tick_16	D17	input	1.5-V HSTL Class I		3
tick_17	A21	input	1.5-V HSTL Class I		3
tick_18	A23	input	1.5-V HSTL Class I		3

Table 43 Pinout by signal name

8.2 Pinout by pin number

Note: UNUSED and NC pins have been removed.

```
Quartus II Version 9.1 Build 304 01/25/2010 Service Pack 1 SJ Full Version
CHIP  "output_top"  ASSIGNED TO AN: EP1SGX10CF672C5

-------------------------------------------------------------------------------------------
Pin Name/Usage               : Location  : Dir.   : I/O Standard       : Voltage : I/O Bank  :
-------------------------------------------------------------------------------------------
GND                          : A2        : gnd    :                    :         :           :
VCCIO4                       : A3        : power  :                    : 3.3V    : 4         :
mcb_data_p[0]                : A4        : bidir  : 3.3-V LVTTL        :         : 4         :
mcb_data_p[1]                : A5        : bidir  : 3.3-V LVTTL        :         : 4         :
mcb_data_p[2]                : A6        : bidir  : 3.3-V LVTTL        :         : 4         :
mcb_data_p[3]                : A7        : bidir  : 3.3-V LVTTL        :         : 4         :
mcb_data_p[4]                : A8        : bidir  : 3.3-V LVTTL        :         : 4         :
mcb_data_p[5]                : A9        : bidir  : 3.3-V LVTTL        :         : 4         :
mcb_data_p[6]                : A10       : bidir  : 3.3-V LVTTL        :         : 4         :
GND                          : A11       : gnd    :                    :         :           :
VCCIO4                       : A12       : power  :                    : 3.3V    : 4         :
clock_out_p[0](n)            : A13       : output : LVDS               :         : 10        :
sind_13                      : A14       : input  : 1.5-V HSTL Class I :         : 3         :
VCCIO3                       : A15       : power  :                    : 1.5V    : 3         :
GND                          : A16       : gnd    :                    :         :           :
data_13[3]                   : A17       : input  : 1.5-V HSTL Class I :         : 3         :
sind_14                      : A18       : input  : 1.5-V HSTL Class I :         : 3         :
tick_14                      : A19       : input  : 1.5-V HSTL Class I :         : 3         :
sind_17                      : A20       : input  : 1.5-V HSTL Class I :         : 3         :
tick_17                      : A21       : input  : 1.5-V HSTL Class I :         : 3         :
sind_18                      : A22       : input  : 1.5-V HSTL Class I :         : 3         :
tick_18                      : A23       : input  : 1.5-V HSTL Class I :         : 3         :
```

VCCIO3	A24	power		1.5V	3
GND	A25	gnd			
GXB_VCC*	AA1			1.5V	15
GXB_GND*	AA2				15
GXB_GND	AA3				
GXB_VCC*	AA4			1.5V	15
GXB_GND*	AA5				15
GND	AA6	gnd			
tick_06	AA7	input	1.5-V HSTL Class I		7
data_06[2]	AA8	input	1.5-V HSTL Class I		7
NC	AA9				
NC	AA10				
RESERVED_INPUT	AA11				7
RESERVED_INPUT	AA12				7
clock_out_p[7]	AA13	output	LVDS		11
clock_out_p[6]	AA14	output	LVDS		12
NC	AA15				
RESERVED_INPUT	AA16				8
data_12[2]	AA17	input	1.5-V HSTL Class I		8
data_12[3]	AA18	input	1.5-V HSTL Class I		8
NC	AA19				
NC	AA20				
NC	AA21				
NC	AA22				
NC	AA23				
NC	AA24				
NC	AA25				
NC	AA26				
GXB_GND	AB1				
GXB_GND	AB2				
GXB_GND	AB3				
GXB_GND	AB4				
GXB_GND	AB5				
VCCINT	AB6	power		1.5V	

data_06[0]	AB7	input	1.5-V HSTL Class I		7	
data_06[1]	AB8	input	1.5-V HSTL Class I		7	
data_04[3]	AB9	input	1.5-V HSTL Class I		7	
NC	AB10					
NC	AB11					
stick	AB12	input	LVDS		7	
clock_out_p[7](n)	AB13	output	LVDS		11	
clock_out_p[6](n)	AB14	output	LVDS		12	
NC	AB15					
RESERVED_INPUT	AB16				8	
data_12[0]	AB17	input	1.5-V HSTL Class I		8	
data_12[1]	AB18	input	1.5-V HSTL Class I		8	
NC	AB19					
NC	AB20					
NC	AB21					
NC	AB22					
NC	AB23					
NC	AB24					
NC	AB25					
NC	AB26					
GXB_VCC*	AC1			1.5V	15	
GXB_GND*	AC2				15	
GND	AC3	gnd				
data_01[0]	AC4	input	1.5-V HSTL Class I		7	
data_02[1]	AC5	input	1.5-V HSTL Class I		7	
data_02[2]	AC6	input	1.5-V HSTL Class I		7	
data_03[3]	AC7	input	1.5-V HSTL Class I		7	
data_04[1]	AC8	input	1.5-V HSTL Class I		7	
data_04[2]	AC9	input	1.5-V HSTL Class I		7	
NC	AC10					
NC	AC11					
stick(n)	AC12	input	LVDS		7	
clock_out_p[8]	AC13	output	LVDS		11	
GND+	AC14				8	

data_09[2]	AC15	input	1.5-V HSTL Class I		8
data_09[3]	AC16	input	1.5-V HSTL Class I		8
sind_12	AC17	input	1.5-V HSTL Class I		8
tick_12	AC18	input	1.5-V HSTL Class I		8
NC	AC19				
NC	AC20				
NC	AC21				
NC	AC22				
NC	AC23				
NC	AC24				
NC	AC25				
NC	AC26				
GXB_GND	AD1				
GXB_GND	AD2				
GND	AD3	gnd			
data_01[1]	AD4	input	1.5-V HSTL Class I		7
data_02[0]	AD5	input	1.5-V HSTL Class I		7
data_02[3]	AD6	input	1.5-V HSTL Class I		7
data_03[2]	AD7	input	1.5-V HSTL Class I		7
data_04[0]	AD8	input	1.5-V HSTL Class I		7
data_05[3]	AD9	input	1.5-V HSTL Class I		7
NC	AD10				
NC	AD11				
RESERVED_INPUT	AD12				11
clock_out_p[8](n)	AD13	output	LVDS		11
RESERVED_INPUT	AD14				8
data_09[0]	AD15	input	1.5-V HSTL Class I		8
data_09[1]	AD16	input	1.5-V HSTL Class I		8
data_08[2]	AD17	input	1.5-V HSTL Class I		8
data_08[3]	AD18	input	1.5-V HSTL Class I		8
data_10[2]	AD19	input	1.5-V HSTL Class I		8
data_10[3]	AD20	input	1.5-V HSTL Class I		8
data_11[2]	AD21	input	1.5-V HSTL Class I		8
data_11[3]	AD22	input	1.5-V HSTL Class I		8

NC	AD23					
NC	AD24					
NC	AD25					
VCCIO1	AD26	power		3.3V	1	
GND	AE1	gnd				
sind_01	AE2	input	1.5-V HSTL Class I		7	
tick_01	AE3	input	1.5-V HSTL Class I		7	
data_01[2]	AE4	input	1.5-V HSTL Class I		7	
tick_02	AE5	input	1.5-V HSTL Class I		7	
tick_03	AE6	input	1.5-V HSTL Class I		7	
data_03[1]	AE7	input	1.5-V HSTL Class I		7	
tick_04	AE8	input	1.5-V HSTL Class I		7	
data_05[0]	AE9	input	1.5-V HSTL Class I		7	
data_05[1]	AE10	input	1.5-V HSTL Class I		7	
data_05[2]	AE11	input	1.5-V HSTL Class I		7	
RESERVED_INPUT	AE12				11	
clock_out_p[5]	AE13	output	LVDS		12	
clk_128	AE14	input	LVDS		8	
sind_09	AE15	input	1.5-V HSTL Class I		8	
tick_09	AE16	input	1.5-V HSTL Class I		8	
data_08[0]	AE17	input	1.5-V HSTL Class I		8	
data_08[1]	AE18	input	1.5-V HSTL Class I		8	
data_10[0]	AE19	input	1.5-V HSTL Class I		8	
data_10[1]	AE20	input	1.5-V HSTL Class I		8	
data_11[0]	AE21	input	1.5-V HSTL Class I		8	
data_11[1]	AE22	input	1.5-V HSTL Class I		8	
NC	AE23					
NC	AE24					
NC	AE25					
GND	AE26	gnd				
GND	AF2	gnd				
VCCIO7	AF3	power		1.5V	7	
data_01[3]	AF4	input	1.5-V HSTL Class I		7	
sind_02	AF5	input	1.5-V HSTL Class I		7	

sind_03	AF6	input	1.5-V HSTL Class I		7	
data_03[0]	AF7	input	1.5-V HSTL Class I		7	
sind_04	AF8	input	1.5-V HSTL Class I		7	
sind_05	AF9	input	1.5-V HSTL Class I		7	
tick_05	AF10	input	1.5-V HSTL Class I		7	
GND	AF11	gnd				
VCCIO7	AF12	power		1.5V	7	
clock_out_p[5](n)	AF13	output	LVDS		12	
clk_128(n)	AF14	input	LVDS		8	
VCCIO8	AF15	power		1.5V	8	
GND	AF16	gnd				
sind_08	AF17	input	1.5-V HSTL Class I		8	
tick_08	AF18	input	1.5-V HSTL Class I		8	
sind_10	AF19	input	1.5-V HSTL Class I		8	
tick_10	AF20	input	1.5-V HSTL Class I		8	
sind_11	AF21	input	1.5-V HSTL Class I		8	
tick_11	AF22	input	1.5-V HSTL Class I		8	
RESERVED_INPUT	AF23				8	
VCCIO8	AF24	power		1.5V	8	
GND	AF25	gnd				
GND	B1	gnd				
mcb_cs_p	B2	input	3.3-V LVTTL		4	
mcb_rw_p	B3	input	3.3-V LVTTL		4	
mcb_data_p[7]	B4	bidir	3.3-V LVTTL		4	
mcb_data_p[8]	B5	bidir	3.3-V LVTTL		4	
mcb_data_p[9]	B6	bidir	3.3-V LVTTL		4	
mcb_data_p[10]	B7	bidir	3.3-V LVTTL		4	
mcb_data_p[11]	B8	bidir	3.3-V LVTTL		4	
mcb_data_p[12]	B9	bidir	3.3-V LVTTL		4	
NC	B10					
NC	B11					
clock_out_p[2](n)	B12	output	LVDS		9	
clock_out_p[0]	B13	output	LVDS		10	
GND+	B14				3	

tick_13	B15	input	1.5-V HSTL Class I		3
data_13[1]	B16	input	1.5-V HSTL Class I		3
data_13[2]	B17	input	1.5-V HSTL Class I		3
data_14[0]	B18	input	1.5-V HSTL Class I		3
data_14[1]	B19	input	1.5-V HSTL Class I		3
data_17[0]	B20	input	1.5-V HSTL Class I		3
data_17[1]	B21	input	1.5-V HSTL Class I		3
data_18[0]	B22	input	1.5-V HSTL Class I		3
data_18[1]	B23	input	1.5-V HSTL Class I		3
data_18[3]	B24	input	1.5-V HSTL Class I		3
NC	B25				
GND	B26	gnd			
reset_p	C1	input	3.3-V LVTTL		4
RESERVED_INPUT	C2				4
RESERVED_INPUT	C3				4
mcb_data_p[13]	C4	bidir	3.3-V LVTTL		4
mcb_data_p[14]	C5	bidir	3.3-V LVTTL		4
mcb_data_p[15]	C6	bidir	3.3-V LVTTL		4
mcb_addr_p[0]	C7	input	3.3-V LVTTL		4
mcb_addr_p[1]	C8	input	3.3-V LVTTL		4
mcb_addr_p[2]	C9	input	3.3-V LVTTL		4
NC	C10				
NC	C11				
clock_out_p[2]	C12	output	LVDS		9
clock_out_p[3](n)	C13	output	LVDS		9
RESERVED_INPUT	C14				3
data_13[0]	C15	input	1.5-V HSTL Class I		3
data_15[3]	C16	input	1.5-V HSTL Class I		3
sind_16	C17	input	1.5-V HSTL Class I		3
data_14[2]	C18	input	1.5-V HSTL Class I		3
data_14[3]	C19	input	1.5-V HSTL Class I		3
data_17[2]	C20	input	1.5-V HSTL Class I		3
data_17[3]	C21	input	1.5-V HSTL Class I		3
data_18[2]	C22	input	1.5-V HSTL Class I		3

NC	C23					
NC	C24					
NC	C25					
VCCIO2	C26	power		3.3V	2	
GXB_GND	D1					
GXB_GND	D2					
GND	D3	gnd				
mcb_addr_p[3]	D4	input	3.3-V LVTTL		4	
mcb_addr_p[4]	D5	input	3.3-V LVTTL		4	
mcb_addr_p[5]	D6	input	3.3-V LVTTL		4	
mcb_addr_p[6]	D7	input	3.3-V LVTTL		4	
mcb_addr_p[7]	D8	input	3.3-V LVTTL		4	
tc_cbit_p	D9	input	3.3-V LVTTL		4	
NC	D10					
NC	D11					
RESERVED_INPUT	D12				4	
clock_out_p[3]	D13	output	LVDS		9	
GND+	D14				3	
sind_15	D15	input	1.5-V HSTL Class I		3	
data_15[2]	D16	input	1.5-V HSTL Class I		3	
tick_16	D17	input	1.5-V HSTL Class I		3	
data_16[2]	D18	input	1.5-V HSTL Class I		3	
data_16[3]	D19	input	1.5-V HSTL Class I		3	
RESERVED_INPUT	D20				3	
NC	D21					
NC	D22					
NC	D23					
NC	D24					
NC	D25					
NC	D26					
GND*	E1					
GND*	E2					
GXB_GND	E3					
test_port[0]	E4	output	3.3-V LVTTL		4	

test_port[1]	E5	output	3.3-V LVTTL		4	
test_port[2]	E6	output	3.3-V LVTTL		4	
test_port[3]	E7	output	3.3-V LVTTL		4	
tc_out_p	E8	output	3.3-V LVTTL		4	
NC	E9					
RESERVED_INPUT	E10				4	
RESERVED_INPUT	E11				4	
mcb_clk_p	E12	input	3.3-V LVTTL		4	
RESERVED_INPUT	E13				9	
clock_out_p[1](n)	E14	output	LVDS		10	
tick_15	E15	input	1.5-V HSTL Class I		3	
data_15[1]	E16	input	1.5-V HSTL Class I		3	
data_16[0]	E17	input	1.5-V HSTL Class I		3	
data_16[1]	E18	input	1.5-V HSTL Class I		3	
NC	E19					
NC	E20					
NC	E21					
NC	E22					
NC	E23					
NC	E24					
NC	E25					
NC	E26					
GXB_GND	F1					
GXB_GND	F2					
GXB_GND	F3					
GXB_GND	F4					
GXB_GND	F5					
VCCINT	F6	power		1.5V		
RESERVED_INPUT	F7				4	
~DATA1~ / RESERVED_INPUT	F8	input	3.3-V LVTTL		4	
TMS	F9	input			4	
TDI	F10	input			4	
TDO	F11	output			4	
RESERVED_INPUT	F12				4	

RESERVED_INPUT	F13			9	
clock_out_p[1]	F14	output	LVDS	10	
data_15[0]	F15	input	1.5-V HSTL Class I	3	
NC	F16				
RESERVED_INPUT	F17			3	
RESERVED_INPUT	F18			3	
NC	F19				
NC	F20				
NC	F21				
NC	F22				
NC	F23				
NC	F24				
GND+	F25			2	
GND+	F26			2	
GND*	G1				
GND*	G2				
GXB_GND	G3				
GND*	G4				
GND*	G5				
GND	G6	gnd			
RESERVED_INPUT	G7			4	
RESERVED_INPUT	G8			4	
RESERVED_INPUT	G9			4	
RESERVED_INPUT	G10			4	
RESERVED_INPUT	G11			4	
GND+	G12			4	
GNDA_PLL5	G13	gnd			
NC	G14				
NC	G15				
RESERVED_INPUT	G16			3	
~DATA7~ / RESERVED_INPUT	G17	input	1.5 V	3	
RESERVED_INPUT	G18			3	
NC	G19				
NC	G20				

NC	G21					
NC	G22					
NC	G23					
NC	G24					
GND+	G25				2	
GND+	G26				2	
GXB_GND	H1					
GXB_GND	H2					
GXB_GND	H3					
GXB_GND	H4					
GXB_GND	H5					
VCCINT	H6	power		1.5V		
RESERVED_INPUT	H7				4	
RESERVED_INPUT	H8				4	
TRST	H9	input			4	
~DATA0~ / RESERVED_INPUT	H10	input	3.3-V LVTTL		4	
RESERVED_INPUT	H11				4	
VCCG_PLL5	H12	power		1.5V		
VCCA_PLL5	H13	power		1.5V		
NC	H14					
NC	H15					
RESERVED_INPUT	H16				3	
GND	H17					
RESERVED_INPUT	H18				3	
clock_out_p[4]	H19	output	LVDS		2	
clock_out_p[4](n)	H20	output	LVDS		2	
NC	H21					
NC	H22					
GND+	H23				2	
GND+	H24				2	
GND+	H25				2	
GND+	H26				2	
GND*	J1					
GND*	J2					

GXB_GND	J3					
GND*	J4					
GND*	J5					
GND	J6	gnd				
RESERVED_INPUT	J7					4
TCK	J8	input				4
~DATA3~ / RESERVED_INPUT	J9	input	3.3-V LVTTL			4
TEMPDIODEn	J10					
TEMPDIODEp	J11					
GNDG_PLL5	J12	gnd				
VCC_PLL5_OUTA	J13	power		3.3V		9
NC	J14					
NC	J15					
VREF0B3	J16			0.75V		3
VREF1B3	J17			0.75V		3
NC	J18					
data_out_p[0]	J19	output	LVDS			2
data_out_p[0](n)	J20	output	LVDS			2
data_out_p[1]	J21	output	LVDS			2
data_out_p[1](n)	J22	output	LVDS			2
GND+	J23					2
GND+	J24					2
GND+	J25					2
GND+	J26					2
GXB_GND	K1					
GXB_GND	K2					
GXB_GND	K3					
GXB_GND	K4					
GXB_GND	K5					
VCCINT	K6	power		1.5V		
RREFB15A	K7					
~DATA2~ / RESERVED_INPUT	K8	input	3.3-V LVTTL			4
GND	K9	gnd				
GND	K10	gnd				

NC	K11					
VCC_PLL5_OUTB	K12	power		3.3V	10	
DCLK	K13				3	
nSTATUS	K14				3	
CONF_DONE	K15				3	
~DATA6~ / RESERVED_INPUT	K16	input	1.5 V		3	
data_out_p[4]	K17	output	LVDS		2	
data_out_p[4](n)	K18	output	LVDS		2	
data_out_p[2]	K19	output	LVDS		2	
data_out_p[2](n)	K20	output	LVDS		2	
data_out_p[3]	K21	output	LVDS		2	
data_out_p[3](n)	K22	output	LVDS		2	
GND+	K23				2	
GND+	K24				2	
GND+	K25				2	
GND+	K26				2	
GND*	L1					
GND*	L2					
GXB_GND	L3					
GND*	L4					
GND*	L5					
VCCT_B15	L6	power		1.5V	15	
VCCR_B15	L7	power		1.5V	15	
VCCP_B15	L8	power		1.5V	15	
VCCINT	L9	power		1.5V		
VCCINT	L10	power		1.5V		
VCCINT	L11	power		1.5V		
VCCIO4	L12	power		3.3V	4	
VCCIO3	L13	power		1.5V	3	
nCONFIG	L14				3	
~DATA4~ / RESERVED_INPUT	L15	input	1.5 V		3	
RESERVED_INPUT	L16				3	
GND	L17	gnd				
GND	L18	gnd				

data_out_p[6]	L19	output	LVDS		2	
data_out_p[6](n)	L20	output	LVDS		2	
data_out_p[5]	L21	output	LVDS		2	
data_out_p[5](n)	L22	output	LVDS		2	
GND+	L23				2	
GND+	L24				2	
GND+	L25				2	
GND	L26	gnd				
GXB_GND	M1					
GXB_GND	M2					
GXB_GND	M3					
GXB_GND	M4					
GXB_GND	M5					
VCCT_B15	M6	power		1.5V	15	
VCCR_B15	M7	power		1.5V	15	
VCCP_B15	M8	power		1.5V	15	
VCCINT	M9	power		1.5V		
VCCINT	M10	power		1.5V		
GND	M11	gnd				
GND	M12	gnd				
GND	M13	gnd				
VCCINT	M14	power		1.5V		
~DATA5~ / RESERVED_INPUT	M15	input	1.5 V		3	
RESERVED_INPUT	M16				3	
data_out_p[9]	M17	output	LVDS		2	
data_out_p[9](n)	M18	output	LVDS		2	
data_out_p[8]	M19	output	LVDS		2	
data_out_p[8](n)	M20	output	LVDS		2	
data_out_p[7]	M21	output	LVDS		2	
data_out_p[7](n)	M22	output	LVDS		2	
GND+	M23				2	
GND+	M24				2	
GND+	M25				2	
VCCIO2	M26	power		3.3V	2	

GND*	N1				
GND*	N2				
GXB_GND	N3				
GND*	N4				
GND*	N5				
GND	N6	gnd			
VCCA_B15	N7	power	3.3V	15	
VCCG_B15	N8	power	1.5V	15	
VCCINT	N9	power	1.5V		
GND	N10	gnd			
VCCINT	N11	power	1.5V		
GND	N12	gnd			
VCCINT	N13	power	1.5V		
GND	N14	gnd			
VCCINT	N15	power	1.5V		
VCCINT	N16	power	1.5V		
GND	N17	gnd			
VCCIO2	N18	power	3.3V	2	
GNDG_PLL1	N19	gnd			
VCCA_PLL1	N20	power	1.5V		
VCCG_PLL1	N21	power	1.5V		
GNDA_PLL1	N22	gnd			
GND+	N23			2	
RESERVED_INPUT	N24			2	
GND+	N25			2	
GND+	N26			2	
GXB_GND	P1				
GXB_GND	P2				
GXB_GND	P3				
GXB_GND	P4				
GXB_GND	P5				
GND	P6	gnd			
VCCG_B15	P7	power	1.5V	15	
VCCA_B15	P8	power	3.3V	15	

nCE	P9				7	
VCCINT	P10	power		1.5V		
GND	P11	gnd				
VCCINT	P12	power		1.5V		
GND	P13	gnd				
VCCINT	P14	power		1.5V		
GND	P15	gnd				
VCCINT	P16	power		1.5V		
GND	P17	gnd				
VCCIO1	P18	power		3.3V	1	
GNDG_PLL2	P19	gnd				
VCCG_PLL2	P20	power		1.5V		
VCCA_PLL2	P21	power		1.5V		
GNDA_PLL2	P22	gnd				
GND+	P23				1	
RESERVED_INPUT	P24				1	
GND+	P25				1	
GND+	P26				1	
GXB_VCC*	R1			1.5V	15	
GXB_GND*	R2				15	
GXB_GND	R3					
GXB_VCC*	R4			1.5V	15	
GXB_GND*	R5				15	
VCCT_B15	R6	power		1.5V	15	
VCCR_B15	R7	power		1.5V	15	
VCCP_B15	R8	power		1.5V	15	
VCCSEL	R9				7	
RESERVED_INPUT	R10				7	
VCC_PLL6_OUTB	R11	power		3.3V	12	
GND	R12	gnd				
RESERVED_INPUT	R13				8	
GND	R14	gnd				
RESERVED_INPUT	R15				8	
GND	R16	gnd				

RESERVED_INPUT	R17				1	
RESERVED_INPUT	R18				1	
RESERVED_INPUT	R19				1	
RESERVED_INPUT	R20				1	
RESERVED_INPUT	R21				1	
RESERVED_INPUT	R22				1	
GND+	R23				1	
GND+	R24				1	
GND+	R25				1	
VCCIO1	R26	power		3.3V	1	
GXB_GND	T1					
GXB_GND	T2					
GXB_GND	T3					
GXB_GND	T4					
GXB_GND	T5					
VCCT_B15	T6	power		1.5V	15	
VCCR_B15	T7	power		1.5V	15	
VCCP_B15	T8	power		1.5V	15	
nCEO	T9				7	
nIO_PULLUP	T10				7	
VCCA_PLL6	T11	power		1.5V		
VCCIO7	T12	power		1.5V	7	
MSEL1	T13				8	
VCCIO8	T14	power		1.5V	8	
RESERVED_INPUT	T15				8	
RESERVED_INPUT	T16				8	
NC	T17					
GND	T18	gnd				
data_out_p[10]	T19	output	LVDS		1	
data_out_p[10](n)	T20	output	LVDS		1	
data_out_p[11]	T21	output	LVDS		1	
data_out_p[11](n)	T22	output	LVDS		1	
GND+	T23				1	
GND+	T24				1	

GND+	T25				1	
GND	T26	gnd				
GXB_VCC*	U1			1.5V	15	
GXB_GND*	U2				15	
GXB_GND	U3					
GXB_VCC*	U4			1.5V	15	
GXB_GND*	U5				15	
GND	U6	gnd				
RREFB15	U7					
data_07[2]	U8	input	1.5-V HSTL Class I		7	
data_07[3]	U9	input	1.5-V HSTL Class I		7	
PORSEL	U10				7	
RESERVED_INPUT	U11				7	
GNDG_PLL6	U12	gnd				
MSEL0	U13				8	
RESERVED_INPUT	U14				8	
RESERVED_INPUT	U15				8	
RESERVED_INPUT	U16				8	
data_out_p[14]	U17	output	LVDS		1	
data_out_p[14](n)	U18	output	LVDS		1	
data_out_p[13]	U19	output	LVDS		1	
data_out_p[13](n)	U20	output	LVDS		1	
data_out_p[12]	U21	output	LVDS		1	
data_out_p[12](n)	U22	output	LVDS		1	
GND+	U23				1	
GND+	U24				1	
GND+	U25				1	
GND+	U26				1	
GXB_GND	V1					
GXB_GND	V2					
GXB_GND	V3					
GXB_GND	V4					
GXB_GND	V5					
VCCINT	V6	power		1.5V		

sind_07	V7	input	1.5-V HSTL Class I		7	
tick_07	V8	input	1.5-V HSTL Class I		7	
NC	V9					
VREF1B7	V10			0.75V	7	
VREF0B7	V11			0.75V	7	
VCCG_PLL6	V12	power		1.5V		
VCC_PLL6_OUTA	V13	power		3.3V	11	
MSEL2	V14				8	
NC	V15					
RESERVED_INPUT	V16				8	
VREF1B8	V17			0.75V	8	
VREF0B8	V18			0.75V	8	
data_out_p[16]	V19	output	LVDS		1	
data_out_p[16](n)	V20	output	LVDS		1	
data_out_p[15]	V21	output	LVDS		1	
data_out_p[15](n)	V22	output	LVDS		1	
GND+	V23				1	
GND+	V24				1	
GND+	V25				1	
GND+	V26				1	
GXB_VCC*	W1			1.5V	15	
GXB_GND*	W2				15	
GXB_GND	W3					
GXB_VCC*	W4			1.5V	15	
GXB_GND*	W5				15	
GND	W6	gnd				
data_07[0]	W7	input	1.5-V HSTL Class I		7	
data_07[1]	W8	input	1.5-V HSTL Class I		7	
RESERVED_INPUT	W9				7	
RESERVED_INPUT	W10				7	
RESERVED_INPUT	W11				7	
GNDA_PLL6	W12	gnd				
NC	W13					
NC	W14					

PLL_ENA	W15				8
RESERVED_INPUT	W16				8
RESERVED_INPUT	W17				8
RESERVED_INPUT	W18				8
data_out_p[17]	W19	output	LVDS		1
data_out_p[17](n)	W20	output	LVDS		1
NC	W21				
NC	W22				
GND+	W23				1
GND+	W24				1
GND+	W25				1
GND+	W26				1
GXB_GND	Y1				
GXB_GND	Y2				
GXB_GND	Y3				
GXB_GND	Y4				
GXB_GND	Y5				
VCCINT	Y6	power		1.5V	
sind_06	Y7	input	1.5-V HSTL Class I		7
data_06[3]	Y8	input	1.5-V HSTL Class I		7
RESERVED_INPUT	Y9				7
RESERVED_INPUT	Y10				7
RESERVED_INPUT	Y11				7
GND+	Y12				7
NC	Y13				
NC	Y14				
RESERVED_INPUT	Y15				8
RESERVED_INPUT	Y16				8
RESERVED_INPUT	Y17				8
RESERVED_INPUT	Y18				8
NC	Y19				
NC	Y20				
NC	Y21				
NC	Y22				

GND+	: Y23	:	:	:	: 1	:
GND+	: Y24	:	:	:	: 1	:
GND+	: Y25	:	:	:	: 1	:
GND+	: Y26	:	:	:	: 1	:

Table 44 Pinout by pin number

8.3 Altera Stratix GX – EP1SGX10CF672C5 Pin Locations

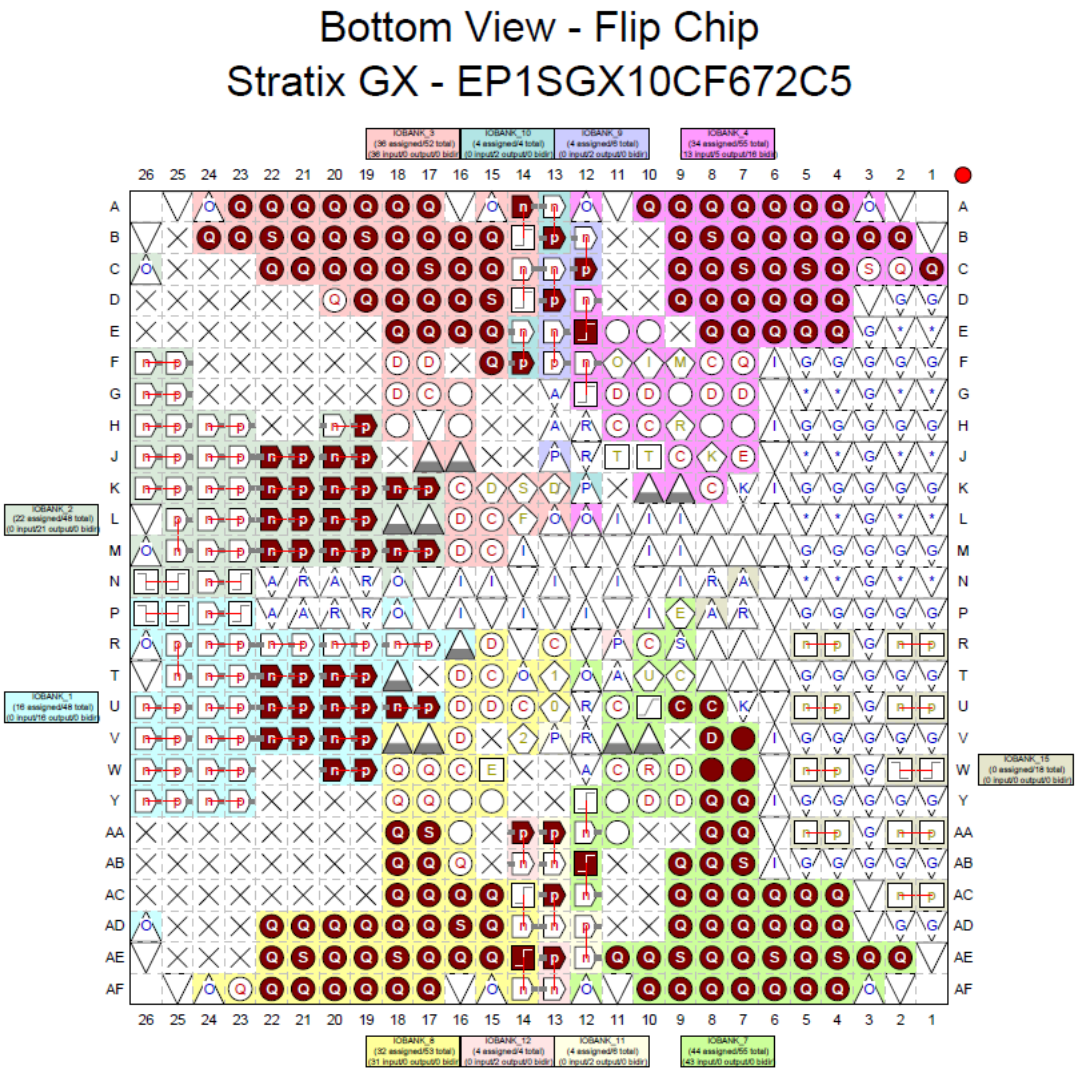


Figure 23 Pin Locations

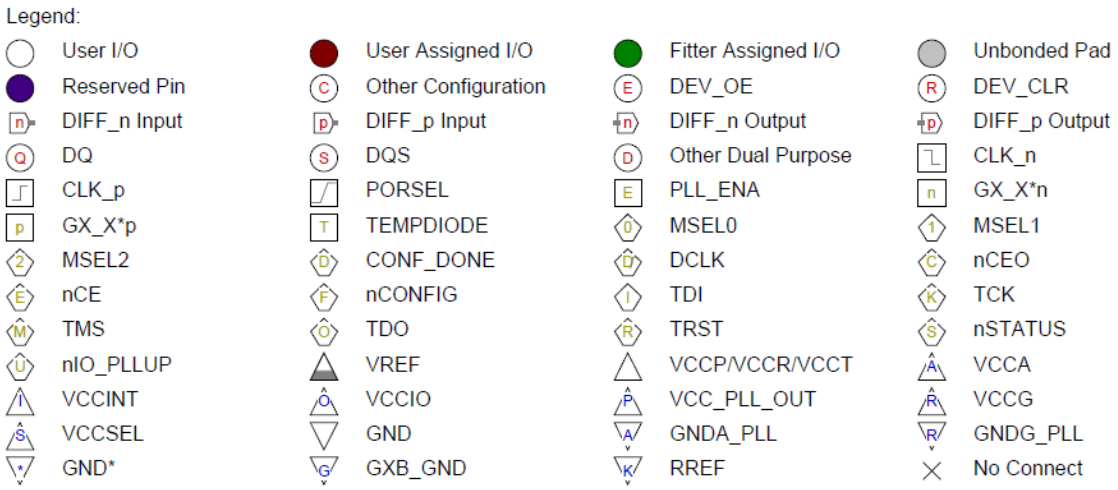


Figure 24 Pin Locations Legend

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 Raw Binary File (.rbf) output file to program the Altera FPGAs. This is set by choosing “Device” from the “Assignments” menu and pushing the “Device and Pin Options” button in the Altera Quartus II software. Select the “Programming Files” tab and check “Raw Binary File (.rbf)”.

9 References

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

Brent Carlson, “MDR-80 Cable and Interface RFS”, EVLA Document A25022N0000.

10 Index