REQUIREMENTS AND FUNCTIONAL SPECIFICATION

Recirculation Controller FPGA

RFS Document: **A25090N0000**

Revision: 3.2c

Brent Carlson, October 2, 2012

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

DPA – Dynamic Phase Alignment. This is a circuit in the Altera FPGA that implements the Recirculation Controller which automatically locks to incoming serial Gbps data, and produces output after demultiplexing to recover the original data sent into the Gbps multiplexer.

DPSRAM – Dual Port synchronous Static Random Access Memory. This memory allows simultaneous access to the same memory location on different ports. It is used here to form the recirculation FIFO buffer.

LVTTL – Low voltage TTL signaling standard. Normally, this is 3.3 V, but can be 2.5 V as well.

HSTL – High speed signaling standard, "High Speed Transceiver Logic".

PECL – Positive Emitter-Coupled Logic. A high-speed signaling standard that supports multi-Gbit/sec speeds.

LVDS – Low-voltage Differential Signaling.

1 Revision History

Revision	Date	Changes/Notes	Author
DRAFT	Jan. 14, 2003	Initial DRAFT Revision	B. Carlson
DRAFT 1	Feb. 5, 2003	Change auto-synchronization to manual, add static recirulation delay offset, add internal probing ability.	B. Carlson
DRAFT 2	Feb. 21, 2003	Remove station-station deskew buffer, add more status bits to the MCSR as well as a TSTAMP register and input DATA ID status register. Remove REF_TICK and RAW_TICK since these are no longer required for synchronization.	B. Carlson
DRAFT 3	March 25, 2003	Simplify the ODSCR register (output data switch register) and hardware requirements	B. Carlson
DRAFT 4	March 31, 2003	Change the RSSR0_1 and RSSR2_3 register definitions: allow recirculation streams to also be phase.	B. Carlson
1.0	August 27, 2003	Remove the Static Recirculation Delay Configuration Register. Clean up typos and bugs	B. Carlson
2.0	May 5, 2004	Replace parallel data input via the MDR-80 connector, with Gbps multiplexed data using DPA clock and data recovery, according to the HM Gbps Signaling standard (A25022N0041)	B. Carlson
2.1	May 12, 2004	Add the PMSR register for more active phase model detection capability.	B. Carlson
2.2	Sept. 10, 2004	1 clock to Correlator Chip for daisy-chaining. All I/O except for DPA are 2.5 V LVTTL.	B. Carlson

Revision	Date	Changes/Notes	Author
2.3	Aug. 26, 2005	PHASEMOD for each data stream extracted from own wafer. Add recirculation block counter reset command (110) capability from DUMPTRIG (No change to this document) Add Correlator Chip autocorrelator mode capability. Add R2 (recirculation on all 8 streams) capability, if FPGA and board is so-equipped.	B. Carlson
2.4	Mar. 28, 2006	Add AMUX register for accessing corr chip core voltages. Minor correction in CSSR SEL[0:1] bit descriptions (p 26), DUMPTRIG description on p 28.	B. Carlson
2.5	Nov. 28, 2006	Add more robustness to HM receiver lock acquisition. Update CSSR, LCSR register sections; add the HDLSR register.	B. Carlson
2.6	Mar. 10, 2007	Add clock phase control registers (CLKPH-0, CLKPH-1): separate clocks for each Correlator Chip now generated by this FPGA (CLOCKA, CLOCKA1, …, CLOCKA7). Remove TPSR0, 2 register. These test pins are no longer used, pins are required for new clock lines. Remove TPSR1, 3, 4 pins. Add DUMMY register for S/W read/write test. AMUX register bit assignments (Table 5-8) completely changed to correct bit assignments.	B. Carlson
2.7	August 8, 2007	RBLKSIZE register: add read/write pointer collision detect bit; R2 now uses internal memory for recirculation; add ability to select 8k channels or 16k channels for R2; The ODSCR register now describes the total recirculation channels available, depending on number of streams active. Update diagrams for R2 using internal memory accordingly.	B. Carlson

2.8	May 30, 2008	Add the RECIRC_TYPE, OEDR, and FPCR (Fine Phase Calibration Register). CLKPH-0/1 registers will be obsolete once FPCR takes effect; protocol for change described in the CLKPH-0/1 and FPCR register descriptions. Change the definition of the RBLKSIZE register so that block sizes of n x 128 (max 2048) can be accommodated. Add Appendix with description of external RAM test registers.	B. Carlson
2.81	June 19, 2008	Change the definition of the PLL bit in the CSSR register: writing to this bit resets or enables chip PLLs. This is to allow for on-chip PLL reset functionality.	B. Carlson
2.82	June 25, 2008	Add PLL_L and PLL_U bits to the new FCPCR register to indicate clock phase generation PLL lock status.	B. Carlson
2.9	June 17, 2009	Add PHASERR read regs, at addresses 0x36 and 0x37	B. Carlson
2.91	October 1, 2009	HM Gbps error and loss of lock recovery is now automatic in the FPGA. See the CSSR and LCSR register descriptions.	B. Carlson
3.0	Feb. 5, 2010	Update appendices with device pin, notes, package information.	B. Carlson
3.1	May 21, 2010	Add the DT_RRC_PBID read-only registers at addresses 0x20, 0x21, 0x23, 0x24, 0x2C, 0x2D, 0x2E, 0x2F. Refer to p. 66 Ensure consistent “CTRL” naming nomenclature throughout.	B. Carlson
3.1a	July 7, 2010	The PLL bit of the CSSR register now latches a loss of PLL lock condition; cleared on read. Applicable to FPGA binaries dated July 7, 2010 or later.	B. Carlson
3.1b	Feb. 9, 2012	Add more description to the D_P0 bit of the RSSR0_1 register (p. 52) quantifying the effects on “serial phase” during recirculation[1]. Also, indicate how full sampling phase can be used.	B. Carlson
3.2	March 9, 2012	Add the “rep23” bit to the RSRCR reg.	B. Carlson
3.2a	July 27, 2012	Add the “rep45” bit to the RSRCR reg.	B. Carlson

[1] From a January 28, 2011 email the author sent to Michael Rupen and Ken Sowinski regarding this issue.

3.2b	August 31, 2012	Add note on page 55 regarding correlating a mixture of recirculation internal and external RAM, due to buffer size differences.	B. Carlson
3.2c	October 2, 2012	Add explanatory text in section 5.4.1.16 regarding dynamic recirculation at low output sample rates to the Correlator Chip, and the interaction of these low sample rates with the “holdoff” time and DUMPTRIG signaling.	B. Carlson

2 Introduction

This document describes detailed requirements and design concepts for the Recirculation Controller FPGA. This FPGA resides on the Baseline Board and is responsible for decoding data, timing, and control information from Station Boards and re-formatting it for transmission to a row or column of Correlator Chips. It also performs recirculation operations using external DPSRAM—recirculation is used to obtain more correlator lags than is directly available with Correlator Chip hardware.

The RFS specification for data, timing, and control information coming from Station Boards can be found in [1], and Correlator Chip input requirements can be found in [2]. Background information on the system design can be found in [3], a more recent description of the system can be found in [4], and a description of the "new connectivity scheme" which is the most applicable to understand how the FPGA fits into the overall system can be found in [5].

3 Context

Recirculation Controllers reside on the Baseline Board and are used to feed, in a daisy-chained fashion, a row or column of Correlator Chips with data and a clock. There are 16 Recirculation Controllers on a board—8 for 'X' inputs and 8 for 'Y' inputs. The basic design of each one is the same, although some modes of operation require configuring the chip as an 'X' or 'Y' controller. Inputs to the controller include data, control, timing, and model information coming from Station Boards via the RXPs FPGAs on the HM Gbps cable via an HM connector [1] along with the associated 128 MHz clock. Outputs are Interrupt_TICK, data, phase, timing, control information, and clocks to a row or column of Correlator Chips. A simplified diagram of the environment is shown in Figure 3-1.

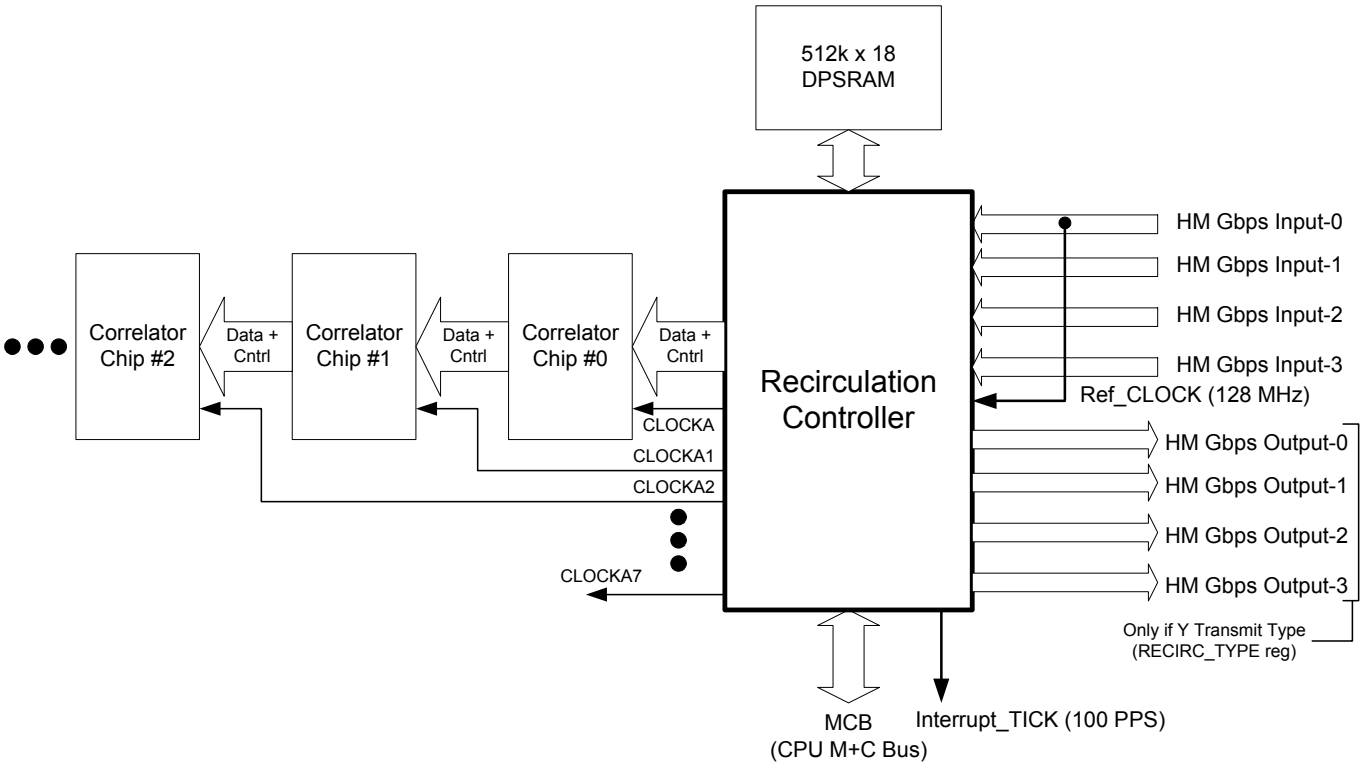


Figure 3-1 Simplified block diagram of the Recirculation Controller FPGA environment. The "Ref_CLOCK" along with timing derived from TIMECODE on the 4 HM Gbps inputs forms the local timebase to which the FPGA synchronizes. There are 4 HM Gbps inputs, one from each source Station Board. The external 512k x 18 DPSRAM (Dual-Port Synchronous Static RAM) is the recirculation buffer for 4 streams active; when 8 streams are active, internal memory is used. The output CLOCKs to the Correlator Chips are synchronous to Ref_CLOCK, and are edge-aligned mid-cell with the 256 Mbit/sec data arriving at each Correlator Chip. Each Correlator Chip gets its own phase-controlled clock (CLOCKA, CLOCKA1, …, CLOCKA7) from this FPGA. Interrupt_TICK is extracted from the input TIMECODE and generated to provide a controlling microprocessor with an interrupt signal, synchronous with the MCB interface clock.

Ref_CLOCK is a 128 MHz clock provided by the RXP FPGA and its phase relative to the same clock of other HM Gbps inputs or at other Recirculation Controllers is irrelevant. The Interrupt_TICK is a 10 millisecond tick (30 nsec pulse synchronous with

MCB_CLK) extracted from a TIMECODE selected from one of the HM Gbps inputs. Interrupt_TICK is generated so it can be used as an interrupt for a controlling microprocessor. Each Recirculation Controller operates in its own absolute time reference frame—final synchronization between X and Y data is performed on the Correlator Chip. However, the data and clocks transmitted from a particular Recirculation Controller to the Correlator Chip are phase synchronous.

4 Overview

A simplified block diagram of the Recirculation Controller FPGA is shown in Figure 4-1.

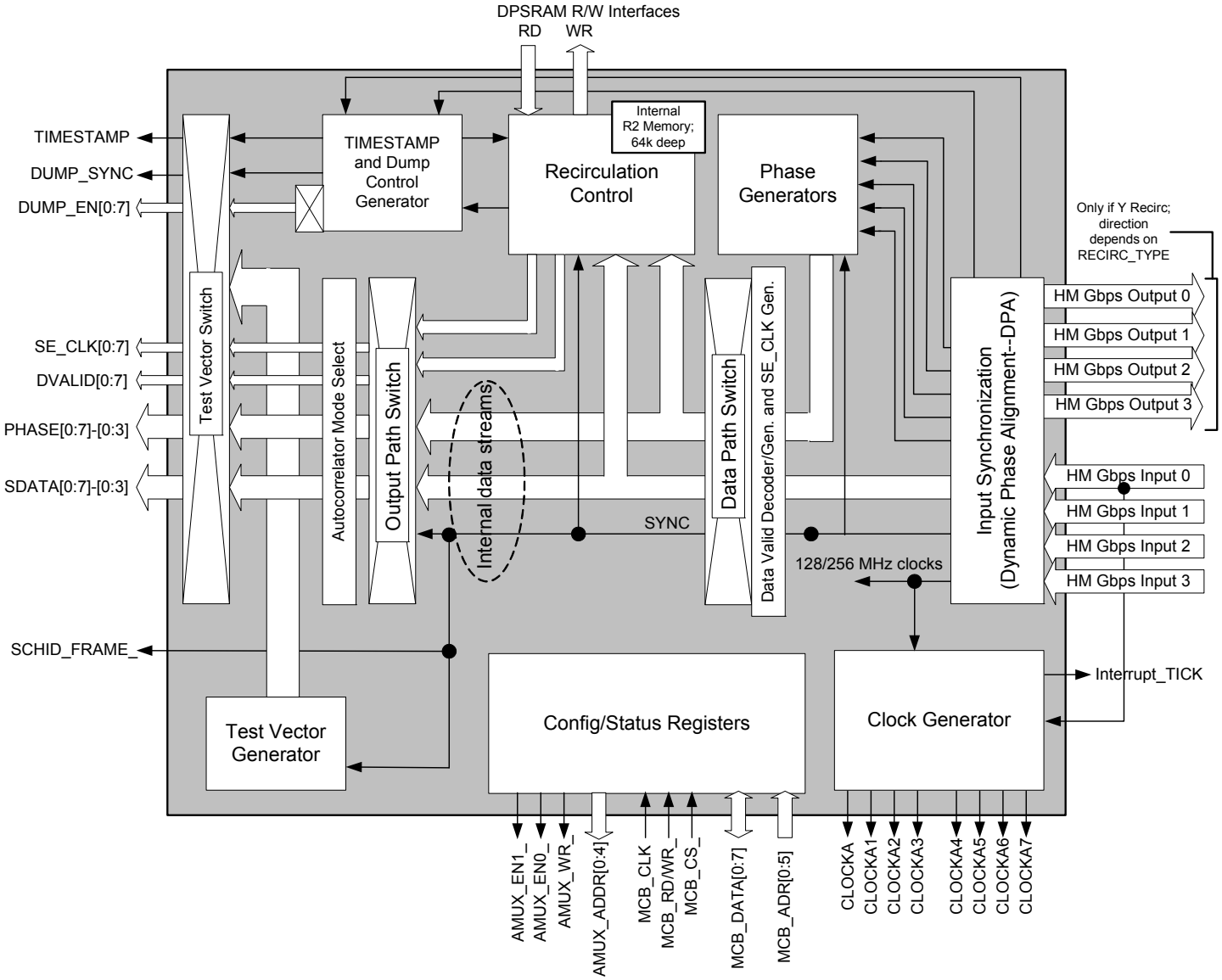


Figure 4-1 Simplified block diagram of the Recirculation Controller FPGA. The input synchronization DPA section de-skews the data and control signals relative to the clock, and lines them all up for use by the rest of the chip. Phase generation, recirculation control, data path switching, and test vector generation are other functions performed by the device.

The chip consists of several separate but interconnected sections. A brief description of each of these sections is as follows:

- **Input Synchronization**. This block de-skews and synchronizes all incoming data and control information present on the 1.024 Gbps HM Gbps signals [1].

Additional logic works automatically by checking for errors using the streams' built-in error checking (CRC-4) capability, and adjusting the phase of the recovered bit streams until there are no errors. Finally, this block de-multiplexes the 7-bit wide multiplexed data into two, ½ speed data streams (i.e. when an input stream is operating in 7-bit mode).

- **Phase Generators**. This block generates the 4-bit phase for each of the 8 sampled data streams. The phase includes the model from PHASEMOD and PHASERR for very fine delay tracking.

- **Recirculation Control**. This block interfaces with external DPSRAM (Dual-Port Synchronous static RAM) and DUMPTRIG synchronization to perform the write and read recirculation functions. It produces output that contains the properly placed embedded identifiers (and an additional data valid line to invalidate the input identifiers) so that going into the output data path switch it looks just like any other sampled data and phase. This block can handle 4 sampled data streams by serializing the phase that goes into the external memory since the external DPSRAM memory is only 18 bits wide (i.e. 16 bits for data, 1 bit for phase, and 1 bit for synchronization). If more than 4 streams are active, then additional internal "R2" 64kx18 RAM is active and either 8k or 16k channels maximum per cross-correlation can be produced.

- **Data Valid Decoder/Gen. And SE_CLK Gen**. This block decodes data valid from the 4-bit or 7-bit data and generates a separate data valid line for each sampled data stream. It ensures that the generated signals flag as invalid any embedded identifiers or CRC check codes. This block also generates the SE_CLK (shift enable clocks) that determine the sample rate the data stream is *effectively* running at.

- **Data Path Switch.** This is a full cross-bar switch that allows any output to connect to any input sampled data stream. This switch is necessary for full correlation flexibility in the correlator. The output of this switch forms the "internal" data streams from which the Recirculation Control and the Output Path Switch take their data.

- **Output Path Switch**. This switch allows the output to be connected to "internal" data streams or to recirculation data streams. This is not a full cross-bar switch since full switching is already provided by the Data Path Switch—it *only* allows the selection of whether outputs are connected to internal data streams or recirculation streams.

- **Autocorrelator Mode Select**. This block contains logic that prepares each data and phase stream for use by the column of Correlator Chips in autocorrelator mode, such that the peak of the autocorrelation function shows up at CCC-15 lag 0 so that 2048 spectral channels can be obtained. In this mode, the phase stream contains delayed data rather than phase, and the Correlator Chip is set to

concatenate all lags for one correlation. Each stream can be independently selected for this mode or for normal operation.

- **TIMESTAMP and Dump Control Generator.** This block decodes DUMPTRIG and TIMECODE to produce the output necessary for the Correlator Chip. It interacts with the Recirculation Control block to provide required synchronization and to obtain the "recirculation block number" that is embedded in the output DUMP_EN line. There is a cross-bar switch on the output of this block that matches the setting of the Data Path Switch so that a given DUMP_EN line properly matches its associated output data stream.

- **Test Vector Generator.** This block generates test vectors used for testing connectivity of the Recirculation Controller-to-Correlator Chip connection.

- **Test Vector Switch.** This switch allows the output to be switched between on-line normal operating mode and test vector generation mode. This block is also capable of generating 0's or all 1's on the individual outputs (i.e. each output can be independently set as 0 or 1). This functionality is useful so that only 'X' or 'Y' test vectors can be turned on to detect short-circuit errors that would otherwise show up as ok (also to double-check that the receiver on the Correlator Chip is actually detecting errors).

- **Config/Status Registers.** This is the MCB (Monitor and Control Bus) interface and configuration and status register block.

- **Clock Generator.** This block generates the clocks needed on-chip and off-chip. Eight off-chip, phased controlled 32 MHz clocks are generated, one for each correlator chip in the row or column the FPGA is driving. Each of these clocks can be phased controlled in 122 psec steps.

4.1 Input Synchronization Block External Interface Signals

All input signals on this interface are LVDS, with 200 mV pk-pk receiver thresholds. Outputs (for Y Recirc FPGAs only) are either[2] ~800 mV pk-pk differential (2.5 LVDS levels), or ~1200 mV pk-pk differential (Hyper-Transport levels). The Recirc FPGA LVDS receivers use on-chip 100 ohm differential termination for maximum signal integrity.

- HM Gbps Inputs 0-3. These are the inputs from Station Boards, via the RXP FPGAs. Each input consists of 2, 1.024 Gbps "BB" streams (each containing a sampled data stream from a Station Board FIR filter), one control line (CTRL) running at 1.024 Gbps containing TIMECODE, DUMPTRIG, PHASEMOD, and

[2] The FPGA design compile-time supports either LVDS or Hyper-Transport levels. Full system testing indicates that Hyper-Transport levels are required for error-free operation, transmitting data to the companion Baseline Board via the Patch Board.

PHASERR for the 2 data streams. The RXP FPGA generates exactly one 128 MHz clock for each Recirc FPGA. Refer to [1] for more detailed protocol information. The DPA Fast PLL is able to tolerate +/-200 psec of jitter and generates a maximum of +/-100 psec of jitter. Signals are as follows:

- CTRL0_1_pad – CTRL signal associated with BB_pad[1:0].
- CTRL2_3_pad – CTRL signal associated with BB_pad[3:2].
- CTRL4_5_pad – CTRL signal associated with BB_pad[5:4].
- CTRL6_7_pad – CTRL signal associated with BB_pad[7:6].
- BB_pad[7:0] – The eight 1.024 Gbps multiplexed sampled data streams.

- HM Gbps Outputs 0-3. These are copies of the inputs, repeated to go to the Y-ERNI output connector for the **Y Recirc FPGAs only**. These signals are not aligned or synchronized inside the FPGA; they are raw copied from the DPA receivers to the DPA transmitters, with the output clock being a PLL-generated copy of the input clock. Signals are as follows:
  - CTRL0_1_buf_pad – Copied/repeated CTR0_1_pad output.
  - CTRL2_3_buf_pad – Copied/repeated CTRL2_3_pad output.
  - CTRL4_5_buf_pad – Copied/repeated CTRL4_5_pad output.
  - CTRL6_7_buf_pad – Copied/repeated CTRL6_7_pad output.
  - BB_buf_pad[7:0] – Copied/repeated BB_pad output.

Note: The Baseline Board is designed to support an alternative configuration where the Y Recirc FPGA receives HM Gbps data from the Y-ERNI connector rather than from the RXP. In this case, the HM Gbps inputs of the FPGA are used with different pin assignments than for RXP reception, but the HM Gbps outputs are disabled and set to high-impedance. This particular configuration requires 0201 resistors to be installed on pads on the Baseline Board so it operates in a "split-T" termination (section 9.3.1) mode, with both receivers and transmitters (set for hi-Z) set for no termination; but with resistor termination at the "T" junction. Normally this mode of operation would never be needed for the EVLA, unless it is expanded beyond the current 32-station correlator. This mode has not been tested.

4.2 Clock Generator Block External Interface Signals

All signals on this interface are 2.5 V LVTTL.

- 128 MHz and 256 MHz clocks into this block from the DPA receiver's PLL generate the clock signals, properly aligned with output data that go to the Correlator Chip.

- Interrupt_TICK This is a pulse with a width of one MCB_CLK cycle, asserted every 10 milliseconds. It is derived from the selected input TIMECODE.

- CLOCKA, CLOCKA1, …, CLOCKA7. These are 8 phase-settable 32 MHz output clocks that are fed to a row or column of Correlator Chips. These clocks have a maximum of 75 ps of cycle-to-cycle jitter, with 122 psec phase setting steps, and are generated by the on-chip dynamically re-configurable Enhanced Phase-Locked Loops (PLLs). These clocks are distributed to a row or column of Correlator Chips as shown in Figure 3-1.

4.3 Recirculation Control DPSRAM Interface Signals

All signals on this interface are 2.5 V LVTTL unless otherwise noted. Connections are short so no termination is used.

For simplicity, details of this interface are not shown in the overview block diagram of Figure 4-1. Generally, this interface consists of a read and write interface, each one having a "bank A" and "bank B" DPSRAM (since the DPSRAMs run at a maximum frequency of 133 MHz). Each bank consists of 18 bits of address, 18 bits of data, and several control bits (chip select, read/write etc.). . Signals on this interface are as follows (also refer to section 6 and Figure 6-1 for more information):

- wr_addr_ram_pad[17:0] – 18-bit write address bus, which goes to both banks of RAM.

- wr_data_ramA_pad[17:0] – 18-bit write data bus for bank A RAM.

- wr_data_ramB_pad[17:0] – 18-bit write data bus for bank B RAM.

- rd_addr_ram_pad[17:0] – 18-bit read address bus, which goes to both banks of RAM.

- rd_data_ramA_pad[17:0] – 18-bit read data bus for bank A RAM. Set for 3.3 V LVTTL input so PCI clamping diodes can be used for reduced signal ringing.

- rd_data_ramB_pad[17:0] – 18-bit read data bus for bank B RAM. Set for 3.3 V LVTTL input so PCI clamping diodes can be used for reduced signal ringing.

- dpsram_clockA_pad – 128 MHz clock for the bank A RAM.

- dpsram_clockB_pad – 128 MHz clock for the bank B RAM.

4.4 Correlator Chip Interface Signals

All signals on this interface are 2.5 V LVTTL. Connections are short so no termination is used. Refer to [2] for more detailed information.

- SCHID_FRAME_ – "Station Channel ID Frame" pulse, active low. This pulse is low for one 256 MHz clock period (3.9 nsec) to indicate the beginning of embedded identifier and format information in the SDATA, PHASE, DVALID, and SE_CLK outputs.

- SDATA[0:7]-[0:3] – Eight, 4-bit sampled data streams. These streams contain embedded identifier information, synchronized to SCHID_FRAME_. These streams operate at 256 Mbits/sec.

- PHASE[0:7]-[0:3] – Eight, 4-bit phase streams. Phase in these streams is for the associated SDATA stream. These streams contain embedded on-line error check information and operate at 256 Mbits/sec.

- DVALID[0:7] – Eight data valid lines that flag each associated SDATA sample as valid (1) or invalid (0). If flagged invalid, then the sample is not correlated and has no effect on the final output. These lines operate at 256 Mbits/sec.

- SE_CLK[0:7] – Eight, shift-enable clocks that if high (1) indicate that the associated SDATA, PHASE, and DVALID can be shifted through the lag shift registers in the correlator.

- DUMP_SYNC – This signal is asserted high for one 128 MHz clock cycle if any one of the DUMP_EN outputs is asserted (i.e. it tells the downstream Correlator Chips that a dump is occurring on one or more of the DUMP_EN lines). Note that dump control signaling to the Correlator Chip is taken from one and only one CTRL input and that the CTRL input that is used for this purpose is dynamically selected in the Recirc FPGA.

- DUMP_EN[0:7] – Each line indicates that Correlator Chip lags that use the associated SDATA stream are to be dumped to the LTA. These lines contain embedded information that tells the Correlator Chip and the LTA what to do with the dumped data. This signal operates at 128 Mbits/sec.

- TIMESTAMP – This line contains a 64-bit timestamp, derived from TIMECODE embedded in the CTRL line selected for DUMP_SYNC signaling, which is the time when the DUMP_SYNC signal occurred. This signal operates at 128 Mbits/sec and is framed with DUMP_SYNC. Refer to section 5.1.4 bullet 5. for more detailed information.

4.5 MCB Interface I/O Signals (Config/Status Registers Block)

All signals on this interface are 2.5 V LVTTL.

- MCB_ADDR[0:5] – Microprocessor address for accessing chip configuration and status registers.

- MCB_DATA[0:7] – Bi-directional 8-bit microprocessor data bus.

- MCB_CS_ – Input chip select that enables the MCB interface drivers and internal circuitry.

- MCB_RD/WR_ – Input that when low enables writing data into the FPGA. Otherwise, data is driven onto the MCB_DATA bus by the chip, when MCB_CS_ is asserted.

- MCB_CLK – Input clock for the synchronous MCB interface. The phase and frequency of this input must be independent of the main CLOCK, as this clock is also used for input CLOCK monitoring This clock has a nominal frequency of 33 MHz.

4.6 Miscl I/O Signals

- AMUX_ADDR[0:4] – These address lines are used for selection of the external analog mux (switch) by Y-7 Recirculation FPGA for Correlator Chip core voltage monitoring. Controlled by writing to the AMUX register. 2.5 V LVTTL.

- AMUX_WR_ – This line, when low, enables writing an address to the external analog mux. Controlled by writing to the AMUX register. 2.5 V LVTTL.

- AMUX_EN0_, AMUX_EN1_ – Each line, when low enables one of the external analog muxes. Only one line should be low at a time. Controlled by writing to the AMUX register. 2.5 V LVTTL.

- RESET_pad_ – Active-low asynchronous chip reset. Asserting this signal resets all the registers in the chip.

- TP6-TP9 – Test points broken out on the PCB to test pads. Not assigned within the FPGA design.

5 Requirements

The following is a list of Recirculation Controller requirements.

5.1 Functional Requirements

5.1.1 Input Synchronization

1. Station-based input signals contained within the HM Gbps signals can have signal-to-signal skew of up to ±8 bit times at 256 MHz (i.e. ±31 nsec). The reference 128 MHz clock (that is part of the HM Gbps input) phase is also arbitrary, but stable, relative to the input data. The FPGA must be capable of compensating for this skew and arbitrary phase by using the codes and error checks embedded in the signal streams.

2. There will be station-to-station signal arrival mismatches in the system (i.e. to different Recirculation Controllers). It is *not* necessary for the Recirculation Controller to compensate for this mismatch since it is done in the Correlator Chip. It is only necessary for the Recirculation Controller to operate in its own time reference given by the clock on the HM Gbps input.

3. Input HM Gbps signal de-multiplexing, synchronization, and bit alignment will be automatic with only the ability to enable, disable, or restart synchronization under the control of the microprocessor.

4. Sufficient error status detection must be provided to determine if an input is synchronized and locked, or whether it is not. It is not necessary to do any explicit error counting or error statistics.

5. Must be able to select/deselect inputs that are active/inactive since in some correlator configurations not all inputs will be active. Given the facilities provided in item 3. above, this may entirely be a software responsibility.

5.1.2 Phase Generation

1. The FPGA must contain 8, 32-bit linear frequency synthesizers for phase generation. The frequency synthesizers will always operate at a clock rate of 64 MHz—1/4 the maximum clock rate in the system.

2. The 0^{th} and 1^{st} order coefficients for the frequency synthesizers for the data streams *within a wafer* are provided by the PHASEMOD signal *within the same wafer*. Models in PHASEMOD get activated on the subsequent 10 millisecond time tick embedded in TIMECODE. If PHASEMOD does not contain a phase model for a synthesizer, then the synthesizer continues operating with its current model.

3. Four PHASERR inputs, one from each HM Gbps input provide phase offset/error models for very fine delay tracking for associated sampled data streams. The final phase outputs to the Correlator Chip are formed from the most significant 4 bits of the addition of the frequency synthesizers' phase and the appropriate PHASERR.

4. Sampled data streams that undergo recirculation can only have their phase values sampled at $1/8^{th}$ of the *sampled data stream rate*, even though the phase synthesizer is operating at a 64 MHz clock rate. Thus, if all phase states are to be properly sampled (and this is not an absolutely necessary condition), the maximum phase rate is (fs/8)/16 or fs/128. For example, if the sampled data stream that is to be recirculated is operating at a 1 Ms/s rate (500 kHz bandwidth), then the maximum frequency of the phase synthesizer is 1MHz/128=7.8 kHz. The exception to this rule is if a recirculation stream is programmed to carry phase (instead of data) at the full sample rate in the RSSR0_1 and RSSR2_3 registers. In this case, the maximum phase rate for full phase sampling is fs/16.

5. The X Recirc FPGA includes a phase offset that exactly compensates for the phase bias inherent in the Correlator Chip's 3-level phase rotator. Thus, the output of the Correlator Chip will have no inherent phase bias.

5.1.3 Recirculation

This section defines recirculation requirements. For more detailed information on how recirculation operates, refer to [3], section 4.14 and 4.15.

1. Must be able to perform recirculation on up to 4 of the input data streams in 4-bit sample mode, and up to 2 of the input data streams in 7-bit sample mode. In all cases, as indicated in the previous section, phase is updated every 8 samples, or every sample, depending on configuration.

2. Must be able to perform "**R2**" recirculation on all 8 data streams in 4-bit sample mode, and 4 data streams in 7-bit sample mode, using additional internal 64kx18 RAM. This is only possible if the EP2S30 FPGA is installed on the board.

3. All sampled data streams that are being recirculated, including R2 data streams, must be operating at the same sampled data rate (or, stated another way, must have originated from sub-band FIR filters with the same width) and must use the same ***recirculation factor***. The recirculation factor is how many lag blocks/chunks are being synthesized.

4. A 512k x 18 recirculation buffer is required. This buffer allows recirculation to synthesize up to 512k lags (256k samples of delay), with an integration time of 1 millisecond. It is not possible to synthesize more than 512k lags since the maximum number of recirculation blocks is 256 (i.e. the maximum recirculation factor is 256), and since the Correlator Chip only contains 2048 lags. When R2 is

active, an additional internal 64kx18 recirculation buffer is used, and a maximum of 8k or 16k channels can be acquired.

5. Since the maximum integration time of the Correlator Chip is 500 μsec, recirculation integrate and dump must operate at this speed or faster. A shorter integration time than 500 μsec can be used, subject to the maximum dump rate capability in the Correlator Chip, and bearing in mind that the relative correlation blanking percentage increases with decreasing integration time (i.e. due to having to flush old samples out of the Correlator Chip lag shift register pipeline). The integration time is entirely determined by DUMPTRIG signaling.

6. Recirculation must be able to operate (i.e. sampled streams being recirculated) at all sample rates supported by the correlator. Namely, 256 Ms/s, 128 Ms/s, 64 Ms/s, …, 62.5 ks/s.

7. It must be possible to set recirculation control to use 128xN (N is an integer) up to 2048 lags in the Correlator Chip. The number of lags used is known as the ***recirculation block size***.

8. It must be possible to set the range of "recirculation lag blocks" that are acquired. This determines how many lags, and the range of lags that are to be synthesized. It must be possible to configure the chip as an 'X' or 'Y' generator since this setting controls whether the lag block counter is a down or up counter respectively.

   Example: if the lag block size is 2k, and 64k lags are being acquired by two Correlator Chips (i.e. two sub-band correlators), then the first chip does lags 0…32k and the second chip does lags 32k…64k. Thus, the first chip would do lag block 0 to 15 and the second chip would do lag blocks 16 to 31.

9. If only one lag block is being synthesized, then recirculation must operate in a flow-through fashion where the recirculation buffer acts as a pure static integer delay. **This is often referred to as "static recirculation".**

10. Recirculation synchronization occurs with the use of DUMPTRIG. Recirculation Controllers generating recirculation data that is correlated, must generate DUMP_SYNC and DUMP_EN pulses that are properly aligned across the system. The primary mechanism for ensuring this is to ensure that DUMPTRIG generation at each source Station Board is synchronized to its TIMECODE signal identically across the system. A secondary check is with DUMP_SYNC synchronization status in each Correlator Chip, or, Correlator Chip DESSR status registers. A tertiary method is with the use of DUMPTRIG "synchronization test frames" [1]. The chip must have the ability to detect whether these dump pulses are properly synchronized to TIMECODE.

11. Recirculation output streams must contain the same embedded format as other, non-recirculation streams. Any additional or residual embedded format must be properly flagged as invalid before going to the Correlator Chip.

12. Recirculation control must provide a "holdoff" signal that guarantees that data in the Correlator Chip lag pipeline is flushed before correlation occurs. The duration of the holdoff depends on the recirculation block size and the sampled data stream rate going to the Correlator Chip. The holdoff function is part of DUMP_EN signaling.

13. If the desired recirculation factor is less than the factor that is possible, then the recirculation read controller reads out data a corresponding factor slower than is possible, and this data going to the Correlator Chip is handled in a normal way with a shift enable clock.

    Example: If the sub-band sample rate is 16 Ms/s, and only a recirculation factor of 4 is desired, then the effective sample rate going to the Correlator Chip is 64 Ms/s. Data is read out of the recirculation buffer at a 64 Ms/s rate, and the shift-enable clock going to the Correlator Chip for this stream is at a 64 MHz rate (one pulse high, every four 256 MHz clock cycles).

14. Recirculation stream 0 is the master control stream. All dumping and recirculation operations for other streams are slaved to stream 0.

15. Except for the selection of internal data streams to operate on, R2 recirculation operates in an identical fashion to standard recirculation, including sharing of all recirculation parameters.

5.1.4 Miscellaneous

1. The FPGA must be able to decode the encoded data valid flagging in 4-bit and 7-bit mode. This encoding is defined in [1].

2. Must be able to generate shift-enable clocks (SE_CLKs) going to the Correlator Chips that cover the range of sample frequencies that are possible. This range is 256 Ms/s, 128 Ms/s, 64 Ms/s, …, 62.5 ks/s.

3. The controller must be capable of simultaneous recirculation and non-recirculation operation on different data streams.

4. A data path switch must be provided to allow any of the 8 output streams that go to the Correlator Chip to be connected to any of the 16 input streams (8 internal

streams and 8 recirculation streams). This means that it is possible to simultaneously correlate a given stream with *and* without recirculation[3].

5. The generated TIMESTAMP must be the time when the DUMP_SYNC pulse occurred. The format of TIMESTAMP is as defined in [2] and is as follows:

    a. **Word 0**: COUNTPPS extracted from TIMECODE.

    b. **Word 1**: B0-B28 – count of number of 128 MHz clock cycles since the last 1PPS epoch (i.e. the last 1PPS TIMECODE 'T' bit). B29-B31 – epoch (000b = 00:00:00 Jan 1/1970; other epochs are TBD), extracted from TIMECODE.

    Note that due to pipelining in the chip, it is possible for the count of the number of 128 MHz clock cycles (Word 1, B0-B28) since the last 1PPS epoch to exceed 128,000,000 in which case it is the count since the previous 1 PPS. However, at all times the total timestamp will yield the correct result.

6. The test vector generator must match the specifications defined in the Correlator Chip RFS document [2].

7. If the FPGA is configured to generate test vectors to the Correlator Chip, then normal internal operation is not suspended: the output is simply replaced with test vectors. The only inputs *required* in this case, are REF_CLOCK and the input TIMECODE (i.e. only the 128 MHz clock, and CTRL line of the HM Gbps signal need be active). DUMP_SYNC that synchronizes the generation of test vectors on DUMP_EN and TIMESTAMP in this case is generated internally. <u>To facilitate faster Correlator Chip clock phase calibration, when in test vector mode, SCHID_FRAME_ pulses are generated, and the test sequence repeats, every 10 microseconds, synchronized to the input TIMECODE signal.</u>

8. All configuration registers accessed over the MCB interface shall be read/write and all status-only registers shall be read only, clear on read (if appropriate—there may be some cases where this is not appropriate).

9. DUMP_EN and DUMP_SYNC shall have the format defined in the Correlator Chip RFS specification [2].

10. The chip must contain 4 output pins that, using internal selection logic, can be connected to internal signals. The number of internal signals and the actual internal signals that can be connected is TBD. Note that this is not explicitly in the FPGA design, but is provided by the PCB design.

---

[3] Except for stream-0 of recirculation which is the master stream. This is because stream-0 dump control signals must be sequenced for recirculation, and this is not compatible with dump control for non-recirculation modes. However, some other stream that is being recirculated, and thus its dump control is ignored, could be simultaneously correlated with and without recirculation.

11. The FPGA must support autocorrelator mode in the Correlator Chip by selectively delaying data and feeding it to the Correlator Chip on the associated phase stream such that the peak of the autocorrelation function resides at lag 0 or CCC-15.

12. Allow control of dual external 32:1 analog muxes/switches for Correlator Chip core voltage monitoring. Only one Recirculation Controller FPGA will be used for this purpose.

13. When an HM Gbps input receiver loss of lock condition occurs, or when the "W0,W1" pattern in the DATA input is bad, the chip must force the associated outgoing SDATA SID to all zeros to signal to the LTA that the stream is bad and any data from it should be discarded. This is a measure to help ensure that the LTA does not knowingly accumulate or generate known bad data. The Recirc chip does this silently, with no register indication that this is occurring.

5.2 Performance Requirements

1. The chip shall operate on HM Gbps multiplexed signals at a bit rate of 1.024 Gbps. From these inputs, sampled data can flow at 256 Mbps, and control signals can flow at 128 Mbps. The chip will use the 128 MHz reference clock to generate all internal clocks (except the MCB_CLK), and all clocks that go to downstream Correlator Chips.

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

3. The power dissipation of the chip should be as low as possible, however it is possible to heatsink the chip if necessary since there are only 16 of these on a board.

5.3 Environmental Requirements

1. The Recirculation Controller FPGA will be surface-mounted on the Baseline Board motherboard. Additional heat sinks may be attached to the chip to reduce temperature.

2. The board will use forced-air cooling with a normal operating ambient temperature of 20 ^{o}C, and with a maximum ambient temperature of 40 ^{o}C.

5.4 Interface Requirements

1. All HM Gbps input interface requirements are defined in [1]. Input levels are differential LVDS, with 200 mV differential minimum detection levels.

2. All interface requirements for signals to the Correlator Chip are defined in [2].

3. The DPSRAM interface requirements may be obtained by referring to the appropriate data sheet. The intent is to use the IDT70V3319S synchronous dual-port static RAM or equivalent. The interface levels are LVTTL 2.5 V.

4. All outputs to the Correlator Chips shall be single-ended LVTTL 2.5 V.

5.4.1 MCB (Microprocessor) Interface Requirements

The MCB (Monitor & Control Bus) interface allows a microprocessor to write into the Correlator Chip to configure it, and to read from it to verify configuration information and obtain status information. Interface timing requirements are shown in Figure 5-1. It is a simple synchronous read/write interface and the interface voltage levels are 2.5V TTL.

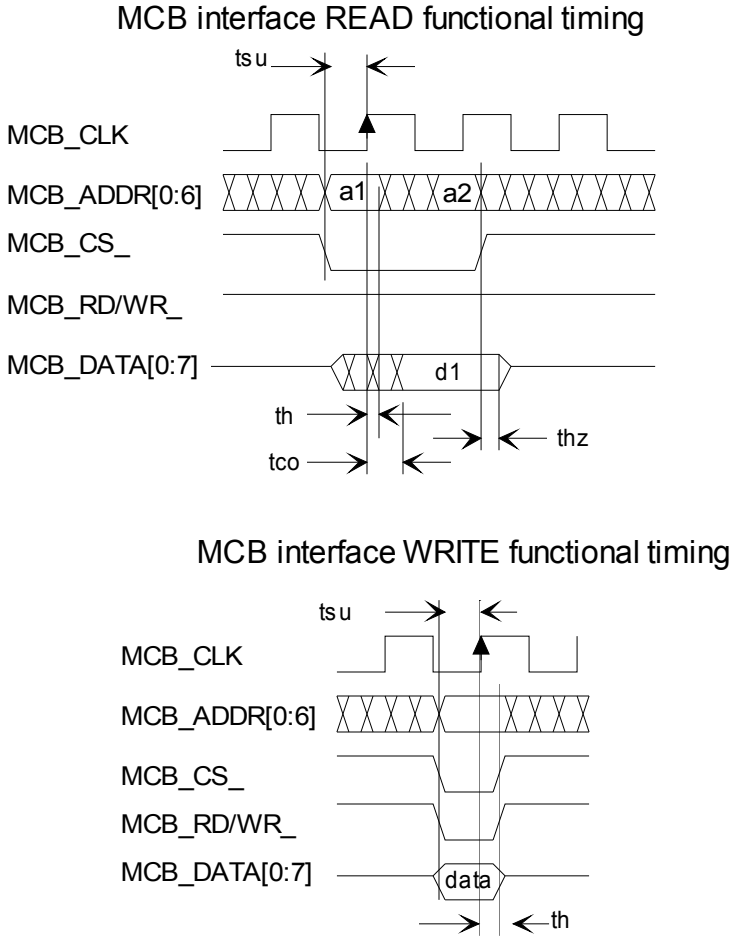


Figure 5-1 MCB interface functional timing requirements. READs require a one clock cycle setup before data is ready—however, successive reads take one clock cycle each. WRITEs require one clock cycle.

The Recirculation Controller contains a number of configuration and status registers that can be accessed via the MCB. These registers are defined in the following sub-sections. **Note that RESET_pad_ is the reset signal for all MCB register bits in the device**. Therefore, 128 PLL lock status does not affect them.

5.4.1.1 Master Control/Status Register (MCSR); R/W Address=0x00

This register sets global chip configuration, and detects global status information such as input synchronization errors and input synchronization lock. A complete description of the register bits is as follows.

MCSR (R/W) Addr=0x00 (Reset 0x00)

7	6	5	4	3	2	1	0
DSYNC	TIDER	IDEN	PMOD	TC	ISERR	REN	TVEN

TVEN (R/W) – Test Vector generation ENable. If set (1), the FPGA output to the Correlator Chips is generating test vectors. If reset (0), then the output is generating data as it normally does during normal on-line operation. This setting does not affect any internal chip operation other than the output Test Vector Switch setting.

REN (R/W) – Recirculation ENable. If set (1), recirculation is active and occurs according to the various recirculation control/configuration registers. If reset (0), recirculation is not active. The recirculation control/configuration registers should only be changed when this bit is reset (0), otherwise indeterminate operation could occur.

ISERR (R) – This bit is a read-only bit that indicates if there is any CRC or offset error in any status register (i.e. LCSR register). If reset (0), then there are no errors. If set (1), then it indicates that at least one status register is reporting an error. Note that this bit is not affected by reading it—it is only cleared when one or more status registers causing the error is cleared.

TC (R) – This is a read-only bit that when set (1) indicates that a TIMECODE has been received and that the TIMECODE is in the TSTAMP register. This bit is cleared on read.

PMOD (R) – This is a read-only bit that when set (1) indicates that a phase model has been received on the PHASEMOD input. This is for information purposes to ensure that a controlling microprocessor is aware that phase models are being received.

IDEN (R/W) – Input Decode ENable. If set (1), then the HM Gbps inputs are actively decoded. If reset (0), then the HM Gbps inputs are not actively decoded.

TIDER (R) – If set (1) this bit indicates that two DUMPTRIG frames with different TIDs (Trigger Ids) have been received without an intervening Dump Trigger. This indicates that there is a possible timing problem and/or that Dump Triggers are not properly being used/set. Cleared on read.

DSYNC (R) – This bit indicates if a synchronization test frame has been detected on the DUMPTRIG input since this register was last read. If set (1), then a test frame has been

detected. If reset (0), then no test frame has been detected. This bit is cleared on read. Whether a synchronization test frame is properly aligned with TIMECODE is determined by the OffEr bit in the LCSR register for DUMPTRIG—this bit merely indicates the presence or absence of the test frames and a corresponding Dump Trigger.

5.4.1.2 Timestamp Register (TSTAMP); (R) Addr=0x01…0x06

These 6 registers are used to allow the controlling microprocessor to capture the timestamp that occurred on the last 10 millisecond time tick in TIMECODE. This register should only be read after a 10 millisecond tick has been detected (e.g. by an interrupt) and if the **TC** bit is set in the MCSR register. Register address assignments to TIMECODE timing information are according to the following table:

Address	TIMECODE Byte
0x01	COUNTPPS [7:0]
0x02	COUNTPPS[15:8]
0x03	COUNTPPS[23:16]
0x04	COUNTPPS[31:24]
0x05	COUNTMS[7:0]
0x06	000,EPOCH[2:0],COUNTMS[9:8]

Table 5-1 TSTAMP register address and bit allocations.

5.4.1.3 Control Input Select/Status Register (CSSR); (R/W) Addr=0x07

This register controls the selection of which HM Gbps control (i.e. "CTRL"—refer to [1]) input is used for decoding and generating TIMECODE, DUMPTRIG and PHASEMOD. It also is used to control/enable the dynamic DPA locking circuitry, provide lock status for each CTRL input, and provide the lock status of the chip's PLL (Phase Locked Loop)

CSSR (R/W) Addr=0x07 (Reset 0x00)

7	6	5	4	3	2	1	0
PLL	LS-3	LS-2	LS-1	LS-0	LE	SEL1	SEL0

SEL[0:1] (R/W) – These select lines determine which of the 4 HM Gbps inputs is the source of the chip's TIMECODE and DUMPTRIG. If SEL1=0; SEL0=0, then HM Gbps input 0 is used. If SEL1=0; SEL0=1, then HM Gbps input 1 is used etc. Note that the LE bit should be reset (0) anytime these configuration values are changed. If the selected signal (CTRL line) has gone dead or is in a loss of lock condition, the FPGA will automatically select another good input to use. To force the FPGA to use a particular desired input, that input must have a good signal; forcing the selection to an input that is bad will automatically cause the FPGA to switch to a good signal. At all times, reading this register will indicate which CTRL input is chosen.

LE (R/W) – Lock enable. When reset, the internal DPA locking circuitry is disabled for the CTRL inputs. When set, it is enabled and the status of locking for each HM Gbps CTRL input can be checked with the LS bits. Once lock is achieved, the FPGA automatically monitors the input and will try to relock if the signal is lost or a high error rate is encountered. To manually prompt the chip to once-again try to re-lock, the LE bit must be pulled low and then high again—an action which also resets the built-in DPA synchronizers on the chip. Normally this manual re-lock attempt should never be required.

LS-0…LS-3 (R) – These status bits indicate the lock status of each of the CTRL inputs. If reset (0), the receiver has not acquired lock on the particular CTRL input. If set (1), the receiver has acquired lock. **Note that independent of the SEL[0:1] settings, it is necessary to acquire lock on all CTRL inputs** since each CTRL input contains PHASERR signals for each of the BBs (sampled data streams) associated with that particular HM Gbps input.

PLL (R/W) – ***Reading*** this bit indicates the latched status of the on-chip PLLs (Phase Locked Loops). If reset (0), the PLLs are not locked or have lost lock since the last time this register was read. A latched reset (0) condition is cleared (set 1) on read, if the PLL has since re-established lock. If set (1), the PLLs are locked. Writing a 0 to this bit resets (and holds reset) the PLLs on the chip, at which point this bit should read 0. ***Writing*** a 1

to this bit activates the PLLs, at which point this bit should read 1. Note that for backwards compatibility, this write bit defaults high, meaning the PLL is active; only an explicit write will cause this bit to change state.

5.4.1.4 Lock Control/Status Register (LCSR); (R/W) Addr=0x08

This register allows the microprocessor to enable or disable dynamic DPA signal lock acquisition for the HM Gbps sampled data stream inputs. Additionally, this register allows the controlling microprocessor to determine if a de-multiplexed stream is reporting errors, and whether the stream is toggling or not.

LCSR (R/W) Addr=0x08 (Reset 0x00)

7	6	5	4	3	2	1	0
OffEr	CRCEr	WIG	LE	SEL3	SEL2	SEL1	SEL0

SEL[0:3] (R/W) These bits select which HM Gbps input and demultiplexed data stream the OffEr, CRCEr, and WIG bits indicate status for according to Table 5-2. Note that error detection and capture for all inputs is provided all of the time—these select lines simply allow the controlling microprocessor to set which input the status is for.

SEL[3:0]	HM Gbps Input # / "Stack #"	Signal
0	HM Gbps-0: BB0	DATA-0
1	HM Gbps-0: BB1	DATA-1
2	HM Gbps-1: BB2	DATA-2
3	HM Gbps-1: BB3	DATA-3
4	HM Gbps-2: BB4	DATA-4
5	HM Gbps-2: BB5	DATA-5
6	HM Gbps-3: BB6	DATA-6
7	HM Gbps-3: BB7	DATA-7
8	Selected HM Gbps CTRL[4]	TIMECODE
9	Selected HM Gbps CTRL	DUMPTRIG
10	OR of all active CTRL inputs	PHASEMOD
11	HM Gbps-0: CTRL0_1	PHASERR0_1
12	HM Gbps-1: CTRL2_3	PHASERR2_3
13	HM Gbps-2: CTRL4_5	PHASERR4_5
14	HM Gbps-3: CTRL6_7	PHASERR6_7
15	N/C	N/C

Table 5-2 LCSR SEL[3:0] status selection table.

LE (R/W**) –** Lock Enable. This bit enables (1) or disables locking on all of the HM Gbps DATA/BB inputs, but only if the associated CSSR LS bit is set (1). e.g. if CSSR LS-0 is set (1), then lock on the BB0 and BB1 inputs is attempted; if CSSR LS-1 is set (1), then lock on the BB2 and BB3 inputs is attempted etc.

Once lock is acquired (determined by the status of the bits in the **HDLSR** register—section 5.4.1.27), if lock is lost or an excessive error rate is encountered, the FPGA will automatically try to re-acquire lock on each BB independently. Thus, once lock is acquired, the occasional CRCEr or OffEr will not cause any glitches in input timing. With 10-millisecond time ticks and embedded information (i.e. the 'C' bit of TIMECODE is always 1), the worst-case locking time for the inputs is 9 seconds, although in most cases it should only take about 2 seconds to lock. Once locking is enabled, lock acquisition is completely automatic. Finally, lock acquisition does not have to happen every time there is a configuration change (i.e. different observation) in the correlator—it only needs to be done after board power-up and boot.

WIG (R) This bit is used to indicate whether the particular input is toggling (wiggling) or not. If set (1), then the input is toggling. If reset (0) then no toggling on the input has been detected. This bit is cleared on read. The LE bit must be set for this to be active.

CRCEr (R) This bit is used to indicate whether the particular input has a CRC error or not. If set (1), then a CRC error has been detected. If reset (0), then no CRC error has

[4] i.e. selected by the SEL[0:1] bits of the CSSR register

been detected. This bit is cleared on read. The LE bit must be set and the associated CSSR LS bit must be set, otherwise this bit is indeterminate.

OffEr (R) This bit is used to indicate whether the particular input is offset from TIMECODE or not. If reset (0) then no offset is detected. This bit is cleared on read. The LE bit must be set and the associated CSSR LS bit must be set, otherwise this bit is indeterminate.

An explicit description of how this bit is used for the various input signals is as follows:

TIMECODE: The OffEr bit is not used—should always be 0.

PHASERR: The OffEr bit is set if the fixed bit pattern (01100011010…), coincident with the TIMECODE T bit, is in error.

DATA: The OffEr bit is set if the W0, W1 pattern coincident with the TIMECODE T bit is in error or is not found.

PHASEMOD: The OffEr bit is set if the "Always 0" bit after the CRC-4 code is in error.

DUMPTRIG: The OffEr bit is set if the "Dump Trigger" bit (second consecutive 1 outside of a DUMPTRIG frame) is not aligned with the TIMECODE T bit after a Dump Trigger for a sync test frame, or when an "Always 1" bit is not present following the CRC-4 code in the frame.

5.4.1.5 Dump Request Status Register (DRSR); (R) Addr=0x09

This register is used to determine which output DUMP_EN lines have been active since the last time it was read. If a bit in this register is set (1), then a dump request on the corresponding DUMP_EN line has been transmitted to the Correlator Chips. If reset (0), then no dump request has been generated. This register is not affected when output test vectors are generated or if a bitstream force on an associated DUMP_EN line is in effect. This register is used to allow the controlling microprocessor to determine if DUMPTRIG is being decoded and if dump requests are going to the Correlator Chips.

5.4.1.6 Sample Stream ID Select Register (SSIDSR); (R/W) Addr=0x0A

This is an address select register that is used to determine which SSIDR register is read according to the following table. The SSIDR registers addressed by writing into this register contain extracted embedded ID information in the input DATA streams.

(Reset 0x00)

Selected Register	SSIDSR Value
SSIDR0-A	0x00
SSIDR0-B	0x01
SSIDR1-A	0x02
SSIDR1-B	0x03
SSIDR2-A	0x04
SSIDR2-B	0x05
SSIDR3-A	0x06
SSIDR3-B	0x07
SSIDR4-A	0x08
SSIDR4-B	0x09
SSIDR5-A	0x0A
SSIDR5-B	0x0B
SSIDR6-A	0x0C
SSIDR6-B	0x0D
SSIDR7-A	0x0E
SSIDR7-B	0x0F

Table 5-3 SSIDSR values that select which SSIDR register is read.

5.4.1.7 Sample Stream ID Register (SSIDR); (R) Addr=0x0B

This register actually consists of 16 registers that are addressed or selected according to Table 5-3. The registers allow the controlling microprocessor to determine the embedded identifiers in the incoming DATA/BB streams. For each DATA stream, there are 16 identifier bits, and these are logically divided into an **A** or a **B** register. The contents of the **A** and **B** registers is as follows:

SSIDR*n*-A (R)

7	6	5	4	3	2	1	0
SID7	SID6	SID5	SID4	SID3	SID2	SID1	SID0

SID[7:0] – The 8-bit embedded SID (Station ID).

SSIDR*n*-B (R)

7	6	5	4	3	2	1	0
BBI2	BBI1	BBI0	SBI4	SBI3	SBI2	SBI1	SBI0

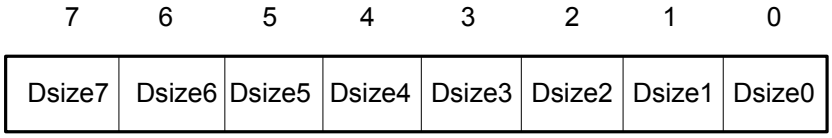
SBI[4:0] – The 5-bit embedded Sub-band ID.

BBI[2:0] – The 3-bit embedded BaseBand ID.

5.4.1.8 Sample Size Configuration Register (SSCR); (R/W) Addr=0x0C

This register is used to define the number of bits per sample for each of the incoming DATA/BB streams.

SSCR (R/W) Addr=0x0C (Reset 0x00)

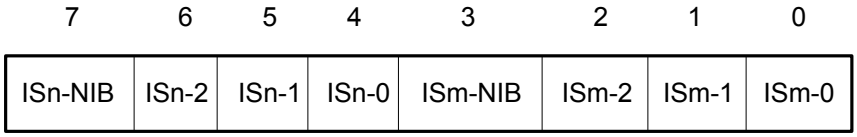


Dsize[0:7] Each one of these bits defines the sample word size for each of the input DATA streams. For example, Dsize4 defines the size for the DATA4 input. If reset (0), then the data stream is 4 bits wide. If set (1), then the data stream is 7 bits wide—in this case the input signal contains multiplexed LSN and MSN words as described in detail in [1].

5.4.1.9 Internal Stream Mapping Control Registers (ISMCRm_n); (R/W) Addr=0x0D, 0x0E, 0x0F, 0x10

These 4 registers are used to define the mapping between the internal (i.e. internal to the FPGA) 8 sampled data streams and the 8 input sampled DATA/BB streams. **This register is needed to properly define the mapping of internal 4-bit wide data streams to input streams defined to be 4 or 7 bits wide**, but it must be configured regardless of whether or not given inputs are defined as 4 or 7 bits wide. This mapping allows the inputs to have an arbitrary assignment of 4 or 7 bits per stream. "m" and "n" are even/odd pairs referring to internal streams "m" and "n" (e.g. m and n can be 0_1, 2_3, etc).

ISMCRm_n (R/W) Addr=0x0D…0x10 (Reset 0x00)



ISm-[0:2] These three bits define what DATA input (i.e. DATA0…DATA7) the "m^{th}" internal data stream is connected to. For example (for m=2) if IS2-[0:2] is set to 011b, then internal data stream 2 is connected to input DATA stream 3.

ISm-NIB This bit is used to define whether the "m^{th}" internal data stream is connected to the MSN (most significant nibble) or LSN (least significant nibble) of the input DATA stream. If reset (0), then the internal stream is connected to the LSN. If set (1), then the internal stream is connected to the MSN. If the input DATA stream is defined as being 4 bits wide (in the SSCR register), then this bit *must* be reset (0) or indeterminate operation will occur.

NOTE: According to the above registers and definitions, if an input DATA stream is defined as 7 bits wide, it requires 2 internal data streams to properly handle the data. Thus, the FPGA can handle a maximum of 4 sampled data streams that are 7 bits wide.

ISn-[0:2] These three bits define what DATA input (i.e. DATA0…DATA7) the "n^{th}" internal data stream is connected to. For example (for n=2) if IS3-[0:2] is set to 011b, then internal data stream 3 is connected to input DATA stream 3.

ISn-NIB This bit is used to define whether the "n^{th}" internal data stream is connected to the MSN (most significant nibble) or LSN (least significant nibble) of the input DATA stream. If reset (0), then the internal stream is connected to the LSN. If set (1), then the internal stream is connected to the MSN. If the input DATA stream is defined as being 4 bits wide (in the SSCR register), then this bit *must* be reset (0) or indeterminate operation will occur.

Example: If internal data streams 4 and 5 are used to carry 7-bit data that is resident on input DATA stream 3, then the ISMCR4_5 register would have a value of 10110011b (stream 4 carrying the LSN and stream 5 carrying the LSN).

Example: If internal data streams 6 and 7 are used to carry 4-bit data that is resident on input DATA streams 6 and 7 (i.e. a direct flow-through mapping), then the ISMCR6_7 register would have a value of 01110110b.

5.4.1.10 Sample Rate Configuration Registers (SRCRm_n); (R/W) Addr=0x11, 0x12, 0x13, 0x14

These registers contain sample rate configuration information for *internal* data streams 0 through 7, where "m" and "n" take on even/odd values and indicate which data streams the register is applicable to (e.g. 0_1, 2_3, etc). Address locations apply to registers in ascending order (i.e. SRCR0_1=Addr 0x0A; SRCR2_3=Addr 0x0B etc.)

SRCRm_n (R/W) Addr=0x11…0x14 (Reset 0x00)

7	6	5	4	3	2	1	0
SRn-3	SRn-2	SRn-1	SRn-0	SRm-3	SRm-2	SRm-1	SRm-0

SRm-[0:3] These four bits define the sample rate for the mth internal stream according to the following table. Any settings other than those defined in the table are not allowed, and a setting of 0000 (Clock divide factor=1) is not allowed (i.e. will result in indeterminate operation) if the sample stream is part of a 7 bit word.

SR0 (3 2 1 0)	Clock divide factor	Sample rate (Ms/s)
0000	1	256
0001	2	128
0010	4	64
0011	8	32
0100	16	16
0101	32	8
0110	64	4
0111	128	2
1000	256	1
1001	512	0.5
1010	1024	0.25
1011	2048	0.125
1100	4096	0.0625

Table 5-4 Sample rate configuration register settings.

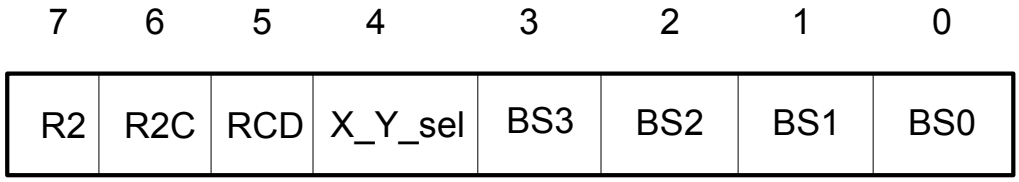
SRn-[0:3] These four bits define the sample rate for the nth internal sample stream according to the above table.

NOTE: streams that will be used for recirculation MUST have the same sample rate definition, otherwise indeterminate operation will occur.

5.4.1.11 Recirculation Block Size Register (RBLKSIZE); (R/W) Addr=0x15

This register sets the recirculation lag block size and sets whether the recirculation controller is for an 'X' or 'Y' station. The recirculation block size is the number of lags that are acquired with each recirculation burst. The block size is a maximum of 2048 and a minimum of 128.

RBLKSIZE (R/W) Addr=0x15 (Reset 0x80)



BS[0:3] (R/W) – Block Size control bits are according to the following equation[5]:

Block Size *in lags* = (BS[3:0] + 1) x 128

Example: *if BS[3:0] = 0101, then the Block Size is 768 lags*

X_Y_sel (R/W) If reset (0), the controller is for an 'X' station, in which case the recirculation block counter is a down counter. If set (1), the controller is for a 'Y' station, in which case the recirculation block counter is an up counter (refer to Figure 5-2).

RCD (R) – Recirculation Collision Detect. If set (1), this indicates that the RAM read pointer has collided with the write pointer during recirculation operations. When this happens, data valid is blanked to prevent corruption of data, and so this flag is primarily for information purposes and does not indicate that data is bad; it does however indicate that data valid will not necessarily be as high as expected. If reset (0), then no pointer collision has occurred. This bit is always 0 when recirculation is not active. Cleared on read.

R2C (R/W) – Recirculation R2 Channel select. This bit sets the maximum number of recirculation frequency channels that can be obtained with R2, when bit streams 4-7 (see ODSCR register, page 54) are active. If reset (0), then the maximum number of *channels* that may be obtained is 8,192 with recirculation, because the R2 recirculation RAM depth is 56 k, requiring a maximum dump period of 224 μsec. In this case, no read/write

[5] Note: the previous Rev. 2.7 definition will remain in effect until such time as a switchover to this definition occurs, coordinated with the S/W group. When the switchover occurs, the old definition will never be reused. NOTE: THE SWITCHOVER HAS OCCURRED AND THE OLD DEFINITION IS DEFUNCT.

pointer collision is ever expected (the RCD bit should never be set), and the correlator chip should not report overrun conditions.

If this bit is set (1), then the maximum number of channels that may be obtained with recirculation is 16,384, since the R2 recirculation RAM depth is set to 48 k, requiring a dump period of ~192 μsec to prevent the read pointer from overwriting the write pointer. In this case, a 192 μsec dump period may cause overruns in the correlator chip, and this may result in overrun frames being discarded (refer to the LTA RFS RE bit in the C3FCSR register(s)). To prevent this condition, the dump period should be set to 200 μsec—and this, depending on the difference between the recirculation RAM write data rate, and the read data rate, may cause the RCD bit to be set, resulting in a small loss of data, but without being corrupted.

Example if the R2C bit is set, the write data rate is 16 Ms/s, and the read data rate is 256 Ms/s: For zero recirculation delay (i.e. when obtaining lag block 0 if Y), in 200 μsec the read pointer will advance by 200 μsec x 256 MHz = 51,200 samples. In the same time, the write pointer will advance by 200 μsec x 16 MHz = 3,200 samples, meaning that the gap will have closed by 48,000 samples, while initially set to 48 k (49,152 samples). Since 48,000 < 49,152 no collision occurs.

Example if the R2C bit is set, the write data rate is 1 Ms/s, and the read data rate is 256 Ms/s: For zero recirculation delay (i.e. when obtaining lag block 0 if Y), in 200 μsec the read pointer will advance by 200 μsec x 256 MHz = 51,200 samples. In the same time, the write pointer will advance by 200 μsec x 1 MHz = 200 samples, meaning that the gap will have closed by 51,000 samples, while initially set to 48 k (49,152 samples). Since 51,000 > 49,152 a collision occurs, and 51,000 – 49,152 = 1,848 samples (~3.7% of the data) is flagged as invalid resulting in a ~1.9% loss of sensitivity. In this example, if the input sample rate is lower than 1 Ms/s, more data loss will occur, whereas if it is higher, less data loss will occur.

R2 (R) – This is the R2 enable bit. If reset (0), then the chip is not enabled for R2 (recirculation on all 8 streams). If set (1), the chip is enabled for R2. Normally, this is always set (1), and R2 operates (on streams 4-7) with internal FPGA RAM.

5.4.1.12 Recirculation Start Block Register (RBLK_START); (R/W) Addr=0x16

(Reset 0x00)

This register is used to set the recirculation start Lag Block, **and is always the same as the "Y Recirc Block" Figure 5-2 below**. This number times the block size is the beginning lag number for the chunk of lags acquired by recirculation *on this Baseline Board*. This number is set the same in both the X and Y Recirculation Controllers (i.e. the hardware performs any calculations/operations that may be different for the X or Y controllers) on the board. This register, and the other recirculation control registers must only be set once—all dynamic internal block counting etc is handled by the FPGA under DUMPTRIG control (section 4.1.5.1 of [3], and Programmer's Guide [6] , section 6.3). A simple example that illustrates this numbering is shown in Figure 5-2 below.

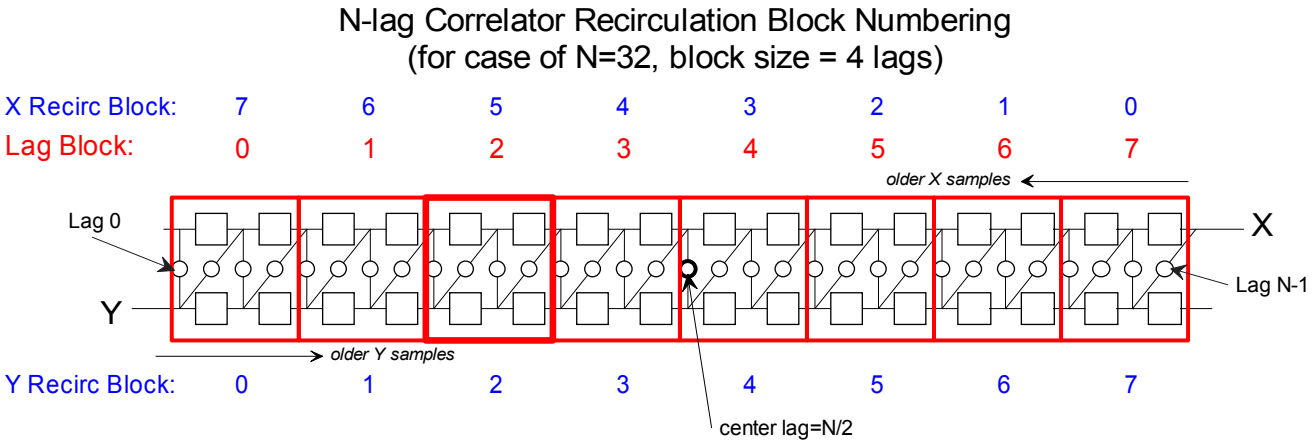


Figure 5-2 Simple example of recirculation lag block numbering. The Start Block is always the lowest numbered block acquired by recirculation, and the End Block is always the highest numbered block acquired by recirculation. In this example a "block size" of 4 lags is shown for clarity; in reality, the minimum block size is 128 lags.

RBLK_START has a maximum value of 255, since there can be a maximum of 256 lag blocks.

NOTE: Recirculation "stream 0" dump control signals are used for control of all recirculation functions[6]. For example, if recirculation stream 0 is connected to chip input data stream 5, then dump control that affects chip input data stream 5 will be used for all recirculation control. Refer to [3], section 4.1.5.1, and the Programmer's Guide [6] , section 6.3 for more information on recirculation synchronization.

[6] A sometimes side-effect of which is that if "stream 0" is connected to an input stream that is not synchronized, no DUMPTRIG decoding is performed, and recirculation/dump control signaling for all recirculation streams dies.

5.4.1.13 Recirculation End Block Register (RBLK_END); (R/W) Addr=0x17

(Reset 0x00)

This register is used to set the recirculation end Lag Block. This number times the block size is the lag number at the beginning of the last lag block acquired by recirculation.

Example: If in Figure 5-2, RBLK_START=1 and RBLK_END=5, then recirculation acquires lags 4-23 inclusive.

IMPORTANT NOTE: Iff RBLK_START=RBLK_END, then recirculation operates in "flow through mode" a.k.a. "static recirculation", wherein only one lag block is continuously acquired and no recirculation-specific signaling from DUMPTRIG is required or should be in place..

IMPORTANT NOTE: If RBLK_END<RBLK_START, indeterminate operation will occur.

5.4.1.14 Total Recirculation Blocks Register (RBLK_TOT); (R/W) Addr=0x18

(Reset 0x00)

This register is used to set the total number of recirculation blocks **- 1** that are acquired for one complete lag set even if some of the lags are acquired on different circuit boards in the system. This is an 8-bit register and so the maximum number of lag blocks is 256 (i.e. a maximum setting of 0xFF results in 256 lag blocks).

Example: If the total number of recirculation lag blocks that are acquired is 32, this register is set to 31.

Example: This register would be set to 7 for the example of Figure 5-2.

5.4.1.15 PHASEMOD Detect Status Register (PMSR); (R) Addr=0x19

This 8-bit read-only register is used to determine if, since the last time it was read, a phase model on PHASEMOD has been detected for each of the corresponding BB inputs. If a bit is set (1), then a phase model on the selected PHASEMOD input has been detected. If a bit is reset (0), then no phase model has been detected. The entire contents of this register are cleared on read.

Example: if bit 3 of this register is set (1), then it indicates that a phase model for input BB3 (i.e. "DATA-3" input present on HM Gbps Input-1—refer to Table 5-2) has been detected and is actively generating phase for the corresponding bitstream in the chip.

Example: if bit 6 of this register is reset (0), then it indicates that a phase model for input BB6 has not been detected. In this case, the on-chip phase generator is freewheeling using the last available linear phase models it was provided with. Depending on the phase acceleration, these models may continue to be valid, or may drift from what they actually should be.

It is important to note that a phase model for a bitstream is only detected on the PHASEMOD input if the corresponding Station ID (SID), BBID, and SBID in the model and the bitstream are exactly matched. This will not be the case if the DPA receiver for a particular bitstream is not locked, and properly decoding these IDs. Thus, if this register indicates that a phase model is not being detected, it could indicate that the DPA receiver for the particular bitstream is not locked, or that the IDs in the bitstream and the PHASEMOD stream are not matched.

Note: There is no requirement for phase models to be updated for every bitstream every 10 milliseconds. For example, if a phase model update is only required every 200 msec, then it would be normal for the corresponding bit in this register to only be set every 20th 10 millisecond interrupt.

Note: If PHASEMOD ceases to be detected for a particular data stream, the phase generator in the chip will not somehow automatically be set to 0. It will keep fly-wheeling with the last detected PHASEMOD indefinitely.

5.4.1.16 Recirculation Sample Rate Configuration Register (RSRCR); (R/W) Addr=0x1B

This register is used to set the sample rate out of the recirculation buffer. Only the lower 4 bits are active in this register.

RSRCR (R/W) Addr=0x1B (Reset 0x00)

7	6	5	4	3	2	1	0
b	b	rep45	rep23	RSR3	RSR2	RSR1	RSR0

RSR[0:3] Recirculation Sample Rate control bits. Settings are identical to those shown in Table 5-4.

The SRCRm_n register and this register along with DUMPTRIG control should be set so that recirculation is active on a continuous basis with a constant integration/dump time. For example, if the input sample rate is 1 Ms/s, and 4 recirculation blocks are being acquired by the controller, then the output sample rate should be set to 4 Ms/s so that recirculation operates in a continuous and predictable way [3]. If the output were to be set to 8 Ms/s, no loss of SNR or change in DUMPTRIG operation results or is required.

Important notes regarding low RSR0-3 sample rate settings:

The inequality:

$$[Block\_size/(2*output\_sample\_rate)] < \sim 0.1^{7} * dump\_time$$

must be satisfied to prevent excessive SNR loss because the Recirc FPGA generates a "holdoff" signal ([2] section 5.4.1) to the Correlator Chip to inhibit (garbage) correlation of previous data still sitting in the Correlator Chip's lag shift registers. This holdoff signal is longer for lower sample rates, and at some point becomes so long that no data is actually correlated. (*Block_size* is set in the RBLKSIZE register; *output_sample_rate* is set in this register's RSR0-3 bits.)

If this equation is not satisfied for the case where:

$$factor = output\_sample\_rate/input\_sample\_rate$$

[7] 0.1 result in ~5% SNR loss; if this is changed to 0.05, the SNR loss is reduced to 2.5% etc.

then the *output_sample_rate* can be set artificially high such that the equation is satisfied. In this case the recirculation write pointer does not move very far in one *dump_time* compared to the movement of the read pointer and therefore effectively for low recirculation factors, redundant correlations are performed resulting in no inherent SNR loss but with skewed SNR calculations if the data_valid counts in Correlator Chip lag frames are used blindly in SNR calculations[8].

Example: The *input_sample_rate* is 125 kHz, the *Block_size* is 128 lags, and the *factor* is 2. Normally this means that the *output_sample_rate* is 250 kHz, but evaluation of the LHS of the above inequality is 128/(2*250e3) = 256 µsec. If the DUMPTRIG period is 200 µsec, then all of the output frames from the Correlator Chip will have data valid counts of zero. Solving the inequality (for 10% loss of data and a 200 µsec *dump_time*) requires that the minimum *output_sample_rate* is 3.2 Ms/s, the next highest being 4 Ms/s. For 5% loss of data, the *output_sample_rate* calculation is 6.4 Ms/s, requiring 8 Ms/s in the real system. If the output_sample_rate is simply set to 256 Ms/s, the holdoff time is 250 nsec resulting in ~0.062% SNR loss.

The general rule then should be that the output_sample_rate should be set to the maximum rate of 256 Ms/s for minimum SNR loss due to holdoff effects.

rep23 – This bit, when set (1), replicates data streams 2 and 3 onto data streams 4, 5, and 6,7 at the output. The purpose of this replication is to allow for full routing and use of downstream Correlator Chips in recirculation modes where only high-capacity external recirculation RAM is used to obtain up to 256k spectral channels. The Correlator Chip allows streams 0 and 1 to be routed to any CCC in the chip; this replication ultimately allows streams 2 and 3 to be routed to any CCCs in the Correlator Chip as well. Replication is part of, and performed within, the "Output Path Switch" of Figure 4-1, but ***after*** the selection of each stream for "internal data stream" or "recirculation stream", set by the ODSCR. If reset (0), no data stream replication is performed and data path switching occurs as normal. **This bit applies only to Recirculation FPGAs with binaries dated on or after February 17, 2012.**

rep45 – Functions the same as "rep23" except it replicates data streams 4 and 5 onto data streams 2,3, and 6,7 at the output, **but only if rep23=0**. Applies to FPGA binaries dated on or after July 18, 2012.

b Read/write bits that are ignored internally.

[8] i.e. The data_valid count taken from the Correlator Chip lag frames for some period of time will be higher than they fundamentally can be.

5.4.1.17 Recirculation Stream 0 and 1 Select Register (RSSR0_1); (R/W) Addr=0x1C

This register is used to select which two *internal* data or phase streams are mapped or selected for recirculation streams 0 and 1. (Note that the internal stream mapping to input DATA streams is defined by setting of the ISMCRm_n registers.) It is important to note that all recirculation streams must be operating at the same sample rate, and all must use the same DUMPTRIG control. Because of these restrictions**, recirculation stream 0 is the "master" stream** and all recirculation happens according to DUMPTRIG signaling associated with stream 0. Refer to [3] section 4.1.5.1 for how DUMPTRIG controls recirculation.

The definition of the register bits is as follows:

RSSR0_1 (R/W) Addr=0x1C (Reset 0x00)

7	6	5	4	3	2	1	0
D_P1	D_P0	RStr1-2	RStr1-1	RStr1-0	RStr0-2	RStr0-1	RStr0-0

RStr0-[0:2] These bits select the internal stream that is mapped to recirculation stream number 0. This mapping is according to the following table.

RStr0- (2 1 0)	Internal Stream Selected
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Table 5-5 Internal stream to recirculation stream mapping.

RStr1-[0:2] These bits select the internal stream that is mapped to recirculation stream number 1. The mapping is according to the above table.

D_P0 This bit determines if recirculation stream 0 is used for data or phase. If reset (0), then the stream is used for an internal data stream. If set (1), then the stream is used for an internal phase stream. Normally this bit is reset (0) and phase for all recirculation streams are serialized and then reconstructed on the output of the buffer. However, in some cases it may be necessary to use phase at the full sample rate (i.e. the same sample

rate as the data), and this mechanism allows for this but at the expense of giving up some total recirculation stream handling capacity.

Within a quad recirculation group (registers RSSR0_1 and RSSR2_3), any recirculation stream can be used for phase (i.e. D_P*n*=1), for any data recirculation stream, as the FPGA contains logic that matches them together after recirculation memory. A recirculation stream used for phase can not explicitly be correlated; doing so causes any resulting Correlator Chip data frames to be rejected by the LTA.

Effects of Recirculation Serial Phase Sampling

Generally, if a data stream uses serial phase, it is sampled with a resolution of one phase sample every 8 data samples, at the original sub-band sample rate. When this occurs, if the phase rate is too high, it can imprint a coherence loss pattern vs lag on the data, that results in spectral artifacts showing up across the band, due to phase jitter imposed by too coarsely sampling the phase vs time function. The pk-pk phase jitter, Φ_{pp} in radians is:

$$\phi_{pp} = 2\pi \cdot \frac{8 \cdot f}{f_s}$$

Where "f" is the phase rate in Hz, and "f_s" is the sample rate in Hz. The coherence loss percentage (CLP) is:

$$CLP = \left(1 - \frac{\sin(\phi_{pp})}{\phi_{pp}}\right) \cdot 100$$

Empirically, testing found that the maximum spectral artifact is approximately:

$$\geq 10 * \log\left(\frac{50}{CLP}\right) dB + 13 dB$$

down from any spectral peak. Roughly speaking, for a spectral artifact to be down by 50 dB from any peak, $f \leq 0.05\% \times f_s$. To ensure random phase jitter, $f_s/(8f)$ should not be an integer, and should have as many unique digits in its calculation as possible; f_X and f_Y should also not have an integer relation.

For X and Y antennas being correlated using different f frequencies f_X and f_Y, Φ_{pp} is calculated as the square-root of the sum of the squares of $\Phi_{X\text{-}pp}$ and $\Phi_{Y\text{-}pp}$ and subsequently used in the above calculation.

If a recirculation stream is used for phase (i.e. D_P*n*=1), then no phase jitter results, and phase is available at full resolution (once every 64 MHz clock cycle, or data sample, whichever is less).

D_P1 Same as D_P0, only it is for recirculation stream 1.

5.4.1.18 Recirculation Stream 2 and 3 Select Register (RSSR2_3); (R/W) Addr=0x1D

This register operates in an identical fashion to the RSSR0_1 register, except that it is for setting the mapping for internal stream to recirculation streams 2 and 3.

5.4.1.19 Recirculation Stream 4 and 5 Select Register (RSSR4_5); (R/W) Addr=0x3C

This register is only useful if the R2 bit of the RBLKSIZE register is set. It operates in an identical fashion to the RSSR0_1 register, except that it is for setting the mapping for internal stream to recirculation streams 4 and 5.

Within a quad recirculation group (registers RSSR4_5 and RSSR6_7), any recirculation stream can be used for phase (i.e. D_P*n*=1), for any DATA recirculation stream, as the FPGA contains logic that matches them together after recirculation memory. A recirculation stream used for phase can not explicitly be correlated; doing so causes any resulting Correlator Chip data frames to be rejected by the LTA.

5.4.1.20 Recirculation Stream 6 and 7 Select Register (RSSR6_7); (R/W) Addr=0x3D

This register is only useful if the R2 bit of the RBLKSIZE register is set. It operates in an identical fashion to the RSSR0_1 register, except that it is for setting the mapping for internal stream to recirculation streams 6 and 7.

5.4.1.21 Output Data Switch Configuration Register (ODSCR); (R/W) Addr=0x1E

This register is used to set the output data selection switches. They define the mapping between the internal data streams, recirculation streams, and output data that goes to the Correlator Chip. Each bit in the register defines the mapping for one output data stream.

ODSCR (R/W) Addr=0x1E (Reset 0x00)

7	6	5	4	3	2	1	0
Ostrm-7	Ostrm-6	OStrm-5	OStrm-4	OStrm-3	OStrm-2	OStrm-1	OStrm-0

Mapping of output data streams to internal data streams and recirculation streams is according to the following table:

ODSCR Reg. Bit	Output Stream	Bit Reset (0) Connection	Bit Set (1)[9] Connection (RBLKSIZE-R2=0)	Bit Set (1) Connection (RBLKSIZE-R2=1)
OStrm-0	0	Internal stream 0	Recirc stream 0	**Recirc stream 0**
OStrm-1	1	Internal stream 1	Recirc stream 1	**Recirc stream 1**
OStrm-2	2	Internal stream 2	Recirc stream 2	**Recirc stream 2**
OStrm-3	3	Internal stream 3	Recirc stream 3	**Recirc stream 3**
OStrm-4	4	Internal stream 4	Recirc stream 2	**Recirc stream 4**
OStrm-5	5	Internal stream 5	Recirc stream 3	**Recirc stream 5**
OStrm-6	6	Internal stream 6	Recirc stream 2	**Recirc stream 6**
OStrm-7	7	Internal stream 7	Recirc stream 3	**Recirc stream 7**

Table 5-6 ODSCR output stream selection

Although the functionality shown in Table 5-6 does not provide full cross-bar capability, it, along with switching in the Correlator Chip, still allows full functionality for recirculation-only, and mixed recirculation modes of operation.

IMPORTANT!
*When **<u>any one</u>** of Output Streams 4-7 are selected for recirculation (when R2 is set (1) for recirculation), then the maximum number of channels for **EVERY** recirculation stream is either 8 k or 16 k, depending on the setting of the R2C bit of the RBLKSIZE register (page 43). Only Output Streams 0-3 are capable of normal recirculation with up to 262,144 channels, and only when Output Streams 4-7 are NOT set to connect to recirculation streams.*

[9] Not normally ever applicable, as RBLKSIZE-R2 bit is always set in the final EVLA Baseline Board implementation (V2.0-V2.2).

IMPORTANT!
When correlating recirculation streams, both the X and Y sources of data must be set for the same use of recirculation RAM. i.e. both X and Y Recirc FPGAs must only set streams 0-3 for use, OR both X and Y Recirc FPGAs must set at least one of streams 4-7 for use. This is because recirculation read pointer offset starting points are set differently for 256k channel operation and for 16k channel operation and ***cannot be mixed.***

5.4.1.22 Autocorrelator Mode Select Register (AMSR); (R/W) Addr=0x26

This register is used to set, for each output data stream, whether it is to be configured to support the Correlator Chip autocorrelator mode. In this mode, the DATA output contains the normal data stream, and the associated PHASE output contains the DATA stream delayed by exactly 1024 samples. The result is that in this mode, the peak of the autocorrelation function is at lag 0 of CCC-15 on the Correlator Chip, and thus a full 2048 spectral channels can be obtained.

AMSR (R/W) Addr=0x26 (Reset 0x00)

7	6	5	4	3	2	1	0
AM-7	AM-6	AM-5	AM-4	AM-3	AM-2	AM-1	AM-0

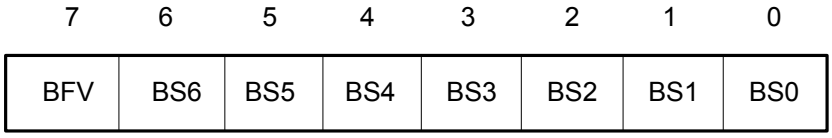
AM-[0:7] Each bit, if set (1), sets the corresponding output DATA and PHASE stream for autocorrelator mode. If reset (0), then the corresponding DATA and PHASE streams are unaffected.

Note that settings in this register affect the data before the bitstream force (BFCR) and before the test vector generator.

5.4.1.23 Bitstream Force Configuration Register (BFCR); (R/W) Addr=0x22

This register is used to allow each individual output bit stream to be forced to be in a reset (0) or set (1) condition for testing connections to the Correlator Chips, and testing whether the Correlator Chips are actually properly detecting errors or not. Bitstream forcing can occur anytime, whether in test vector generation mode or not. Only one bitstream can be forced at a time.

BFCR (R/W) Addr=0x22 (Reset 0x7F)



BS[0:6] These bits select the output bitstream that is to be forced high or low according to the following table.

BS[0:6] (decimal)	Output Stream Forced[10]
0-31	SDATA[0:7]-[0:3]
32-63	PHASE[0:7]-[0:3]
64-71	DVALID[0:7]
72-79	SE_CLK[0:7]
80-87	DUMP_EN[0:7]
88	DUMP_SYNC
89	TIMESTAMP
90	SCHID_FRAME*
91-127	none

Table 5-7 Output bitstream forcing control bits.

BFV Bit Force Value. Reset (0) this bit if the selected output stream is forced low. Set (1) this bit if the selected output stream is forced high.

[10] Numbering is sequential. For example, SDATA number is SDATA0-0, SDATA0-1, SDATA0-2 etc.

5.4.1.24 Analog Mux Selection Register (AMUX); (R/W) Addr=0x27

This register is used to set voltage monitoring selections for the dual 32:1 external analog muxes. The Y-7 Recirculation Controller FPGA is the only FPGA on the board that has this capability, for all other FPGAs, this register has no effect whatsoever.

AMUX (R/W) Addr=0x27 (Reset 0xE0)

7	6	5	4	3	2	1	0
WR	EN1	EN0	AD4	AD3	AD2	AD1	AD0

AD[0:4] – Switch address select. Selection of Correlator Chip (row, column) core voltage is according to Table 5-8.

EN0 – Enable mux 0. If reset (0), then analog mux 0 is selected for output. If set (1), then analog mux 0 output is not enabled.

EN1 – Enable mux 1. If reset (0), then analog mux 1 is selected for output. If set (1), then analog mux 1 output is not enabled.

WR – Write enable. If reset (0), then the AD[0:4] select the address. If set (1), then the AD bits have no effect (i.e. do not change the switch setting from when WR was 0).

EN0 = 0				**EN1 = 0**			
AD[4:0]	**CC (r,c)**	**AD[4:0]**	**CC (r,c)**	**AD[4:0]**	**CC (r,c)**	**AD[4:0]**	**CC (r,c)**
00000	0,7	10000	2,0	00000	4,7	10000	6,0
00001	1,7	10001	3,0	00001	5,7	10001	7,0
00010	3,7	10010	1,0	00010	7,7	10010	5,0
00011	2,7	10011	0,0	00011	6,7	10011	4,0
00100	0,6	10100	2,1	00100	4,6	10100	6,1
00101	1,6	10101	3,1	00101	5,6	10101	7,1
00110	3,6	10110	1,1	00110	7,6	10110	5,1
00111	2,6	10111	0,1	00111	6,6	10111	4,1
01000	0,5	11000	2,2	01000	4,5	11000	6,2
01001	3,5	11001	3,2	01001	7,5	11001	7,2
01010	1,5	11010	1,2	01010	5,5	11010	5,2
01011	2,5	11011	0,2	01011	6,5	11011	4,2
01100	0,4	11100	2,3	01100	4,4	11100	6,3
01101	1,4	11101	3,3	01101	5,4	11101	7,3
01110	3,4	11110	1,3	01110	7,4	11110	5,3
01111	2,4	11111	0,3	01111	6,4	11111	4,3

Table 5-8 External analog mux selection table (AMUX register).

5.4.1.25 CTRL Sync Status Register (CTRL_SSR); (R) Addr=0x28

Bits 0:3 in this register indicate the lock/synchronization status of the corresponding HM CTRL inputs. Bit0==CTRL0_1; Bit1==CTRL2_3 etc.; Bit4-Bit7 are unused. If a particular bit in this register is set (1), the receiver is attempting to lock to the signal. If a particular bit in this register is reset (0), the receiver is not attempting to lock to the signal.

THIS REGISTER IS PROVIDED FOR LOW-LEVEL DEBUG ACCESS ONLY—NO DEVICE DRIVER IS REQUIRED.

5.4.1.26 DATA Sync Status Register (DATA_SSR); (R) Addr=0x29

Bits 0:7 in this register indicate the lock/synchronization status of the corresponding HM DATA inputs (i.e. Bit0==BB0, Bit1==BB1, … Bit7=BB7). If a particular bit in this register is set (1), the receiver is attempting to lock to the signal. If a particular bit in this register is reset (0), the receiver is not attempting to lock to the signal.

THIS REGISTER IS PROVIDED FOR LOW-LEVEL DEBUG ACCESS ONLY—NO DEVICE DRIVER IS REQUIRED.

5.4.1.27 HM Data Lock Status Register (HDLSR); (R) Addr=0x2B

This register is used to determine the receiver lock status of the HM data (Bit0==BB0, Bit1==BB1, …Bit7==BB7) inputs. If the corresponding bit in this register is reset (0), then receiver lock is not acquired. If the corresponding bit in this register is set (1), then receiver lock has been acquired. If all BB inputs are active and receiver lock has been achieved on all of them, this register should read 0xFF.

HDLSR (R) Addr=0x2B (Reset 0x00)

Also refer to the CSSR (section 5.4.1.3) and LCSR (section 5.4.1.4) registers for further information on receiver locking.

5.4.1.28 Correlator Chip Clock Phase Control Registers 0, 1 (CLKPH-0, CLKPH-1); (R/W) Addr=0x31, 0x32

--IMPORTANT INFORMATION—

These registers are NOW OBSOLETE since finer clock phase calibration is required. These registers will always read 0xFF, and the new Fine Clock Phase Calibration Register (FCPCR) MUST be used as these registers are then completely non-functional.

5.4.1.29 Recirculation FPGA Type Register (RECIRC_TYPE); (R/W) Addr=0x33

This register is used to provide software with the ability to determine which recirculation FPGA type this device is, and if it is a "Y transmit type", to allow for HM Gbps outputs to the Y ERNI connector on the board to be disabled, in the case where they are not connected to anything, thereby saving power.

RECIRC_TYPE (R/W) Addr=0x33

7	6	5	4	3	2	1	0
W3-EN	W2-EN	W1-EN	W0-EN	N/C	TxEN	Ty1	Ty0

Ty[1:0] (R/W) – Defines the Recirc FPGA according to the following table:

Ty [1]	Ty [0]	Description
0	0	**X** Recirc FPGA; there is no HM Gbps output.
0	1	**Y** Transmit FPGA; HM Gbps outputs active if enabled.
1	0	**Y** Receive FPGA[11]; there is no HM Gbps output; the board is hard configured for this capability. In this case the Y Recirc FPGA gets its inputs from the Y ERNI connector, rather than the RXP FPGA.
1	1	Reserved/unused.

Table 5-9 RECIRC_TYPE Ty[1:0] bit definitions.

TxEN[12] (R/W) – If Y transmit FPGA (Ty[1:0]=01), then if set (1), the HM Gbps outputs for all channels/wafers from this FPGA are enabled. If reset (0), the HM Gbps outputs for all channels/wafers from this FPGA are disabled. Chip reset value is 1.

W0-EN, …, W3-EN (R/W) – These are individual wafer Y transmit enable/disable bits for output wafers 0-3. If set (1), and TxEN is set (1), then the particular output wafer is enabled (e.g. transmission ON for wafer 0 is the AND of TxEN and W0-EN). Chip reset value for each of these bits is 1. Normally these bits are only used for testing.

[11] This capability is not required for EVLA Phase-I; it is only required if EVLA (and the correlator) is expanded beyond 32 stations, and requires multiple 0201 resistor installation on all Y-Recirc FPGA sites.

[12] Capability not enabled in the current FPGA release binaries. The design supports this and it could be enabled if desired. Same for the W0-EN…W3-EN bits.

5.4.1.30 Output Enable/Disable Register (OEDR); (R/W) Addr=0x34

This register is used to force high or low output data/signaling lines to the correlator chip row or column that this FPGA is driving. It allows more than one line at a time to be forced high or low, complementing the BFCR register. Normally this register is only used for testing, however it could be used to disable correlation or flag data as invalid if the HM Gbps receivers start indicating that they are getting continuous errors to prevent generation of junk data[13].

Chip reset value of this register is 0xFF (all outputs enabled for normal operation).

Bit assignments are as follows:

B0 – SDATA[0:7]-[0:3] – If set (1), then all SDATA outputs are enabled. If reset (0), then all SDATA outputs are disabled, and either high or low depending on B7.

B1 – PHASE[0:7]-[0:3] – If set (1), then all PHASE outputs are enabled as usual. If reset (0), then all PHASE outputs are disabled, and either high or low depending on B7.

B2 – DVALID[0:7] – If set (1), then all DVALID outputs are enabled as usual. If reset (0), then all DVALID outputs are disabled, and either high or low depending on B7. Note that if DVALID is forced high, this will result in connected correlator chips reporting DVALID input sync errors as DVALID normally must go low when sync words and IDs are present.

B3 – SE_CLK[0:7] – If set (1), then all SE_CLK outputs are enabled as usual. If reset (0), then all SE_CLK outputs are disabled, and either high or low depending on B7.

B4 – DUMP_EN[0:7] – If set (1), then all DUMP_EN outputs are enabled as usual. If reset (0), then all DUMP_EN outputs are disabled, and either high or low depending on B7.

B5 – TIMESTAMP – If set (1), then TIMESTAMP is enabled as usual. If reset (0), then TIMESTAMP is disabled, and either high or low depending on B7.

B6 – ticks – If set (1), then SCHID_FRAME_ and DUMP_SYNC are enabled as usual. If reset (0), then they are disabled, and either high or low depending on B7.

B7 – 0=disable force low; 1=disable force high.

[13] Not required anymore since the chip forces outgoing SDATA SIDs to 0 if loss of receiver sync has been detected. Refer to section 5.1.4 bullet 10.

5.4.1.31 Fine Clock Phase Calibration Register (FCPCR); (R/W) Addr=0x35

This register is an upgrade that provides finer clock phase steps for clock phase calibration to the correlator chip array than can be provided with the CLKPH-0 and CLKPH-1 registers.

IMPORTANT NOTE: This register and the CLKPH-0/1 registers are mutually exclusive and once this register is operational, the CLKPH-0/1 registers are obsolete. This register is operational once the CLKPH-0/1 registers always read 0xFF.

Clock phase calibration using the FCPCR register operates differently than the CLKPH-0/1 registers. Absolute clock phase is never set to some specific value, rather **EACH write to this register either advances or retards clock phase by one step**[14] for the selected clock output. Reading the register provides an indication of whether or not the clock phase adjustment has been completed. The register does not have to be "pre-loaded" with parameters; **every** write causes action for chosen clock and phase direction that are part of the write.

FCPCR (R/W) Addr=0x35 Reset=0x00

7	6	5	4	3	2	1	0
N/C	PLL_U	PLL_L	CPD	A/T	CS2	CS1	CS0

CS[2:0] (R/W) – Clock Select bits. These bits select the particular output clock for which the phase adjustment is being performed. i.e. CS[000]=clock for correlator chip row/column 0; CS[001]=clock for correlator chip row/column 1, etc.

A/T (R/W) – Advance/Retard phase. If reset (0), then the write to this register retards the clock phase (i.e. lags more or positive clock phase); if set (1), then the write to this register advances the clock phase (i.e. leads more or negative clock phase).

CPD (R) – Clock Phase Done. When the previously written clock phase adjustment is complete, this bit goes high. Normally this will go high ~3 usec after the write. Cleared on read.

PLL_L (R) – PLL lock, Lower. If set (1), then the clock generation PLL for the lower 4 rows/columns (i.e. CS[2:0]=0…3) is locked. If reset (0), it is not locked.

[14] A clock phase step is 125 psec.

PLL_U (R) – PLL lock, Upper. If set (1), then the clock generation PLL for the upper 4 rows/columns (i.e. CS[2:0]=4…7) is locked. If reset (0), it is not locked.

Notes on clock phase adjustment procedure.

The S/W algorithm that performs clock phase calibration should follow these basic steps. Calibration is analogous to setting the tuning dial on a radio by "listening" to feed back from a radio, and choosing the best setting, rather than setting the dial to an absolute frequency.

1. Turn on testvectors in the Recirc FPGAs and correlator chips.

2. Starting with X clock 0 (CS=000) successively retard clock phase (A/T=0) with successive writes to the FCPCR register; for each step determine testvector error counts in the row 0 correlator chips.

3. Keep retarding clock phases through phases where there are errors, transitioning to phases where there are no errors, and finally transitioning to phases where errors start to show up again. Keep track of the number of phase steps (FCPCR writes) for each setting.

4. "Back off" the clock phase by advancing (A/T=1) with successive writes to the FCPCR register.

5. Repeat the above procedure for each successive row of correlator chips.

6. Once all rows are complete, repeat the procedure for all columns with the Y Recirc FPGA FCPCR.

Since clock phase steps are very fine (~125 psec), the dwell time on each clock phase should be minimized to minimize the time it takes to perform clock phase calibration. It is recommended that the clock phase be set, wait for it to take effect (CPD bit reads high, ~3 µsec later), clear the correlator chip status registers, wait for ~20 µsec[15], clear the correlator chip status registers, wait for 20 µsec, and then read the correlator chip status registers to determine errors.

[15] When the Recirc FPGA is in test vector generation mode, SCHID_FRAME_ signaling to the Correlator Chip occurs every 10 µsec, rather than every 10 msec.

5.4.1.32 PHASERR Read Select Register (PRSR); (R/W) Addr=0x36

This register is used to select the INPUT stream (0-7) that subsequent PHASERR Read Register (PRR Addr 0x37) reads read. For example if 0x05 is written to this register, subsequent reads of register 0x37 contain the current captured value of PHASERR from stream 5.

5.4.1.33 PHASERR Read Register (PRR); (R) Addr=0x37

Reading this register returns the current captured value of PHASERR for the INPUT stream selected by writing to the PRSR register. The read value of PHASERR could in some cases be incorrect, if it happens to be read whilst PHASERR is changing. Thus, this register should be used for debug and troubleshooting only, and not for any data application (if required for a data application, a double read can be used to determine if a settled value is acquired).

5.4.1.34 DumpTrig RRC PhaseBin ID 0_1 (DT_RRC_PBID_0_1); (R) Addr 0x20=LSByte; Addr 0x21=MSByte

These 2 registers form a 16-bit register, and contain the extracted "PB" (Phase Bin) from CMD=RRC (110) DumpTrig Frames for CTRL0_1 into the FPGA. If both of these registers are zero (0x0000), then it indicates that no PB has been extracted, due to the lack of received RRC frames or due to CRC errors detected in DumpTrig RRC frames (i.e. if an RRC frame is detected, and it contains a CRC error, this register is cleared).

Note that these IDs for all CTRL DumpTrig inputs are extracted independent of which CTRL input is selected as the master/reference for correlator chip dump control.

5.4.1.35 DumpTrig RRC PhaseBin ID 2_3 (DT_RRC_PBID_2_3); (R) Addr 0x23=LSByte; Addr 0x24=MSByte

These 2 registers form a 16-bit register, and contain the extracted "PB" (Phase Bin) from CMD=RRC (110) DumpTrig Frames for CTRL2_3 into the FPGA. If both of these registers are zero (0x0000), then it indicates that no PB has been extracted, due to the lack of received RRC frames or due to CRC errors detected in DumpTrig RRC frames (i.e. if an RRC frame is detected, and it contains a CRC error, this register is cleared).

5.4.1.36 DumpTrig RRC PhaseBin ID 4_5 (DT_RRC_PBID_4_5); (R) Addr 0x2C=LSByte; Addr 0x2D=MSByte

These 2 registers form a 16-bit register, and contain the extracted "PB" (Phase Bin) from CMD=RRC (110) DumpTrig Frames for CTRL4_5 into the FPGA. If both of these registers are zero (0x0000), then it indicates that no PB has been extracted, due to the lack of received RRC frames or due to CRC errors detected in DumpTrig RRC frames (i.e. if an RRC frame is detected, and it contains a CRC error, this register is cleared).

5.4.1.37 DumpTrig RRC PhaseBin ID 6_7 (DT_RRC_PBID_6_7); (R) Addr 0x2E=LSByte; Addr 0x2F=MSByte

These 2 registers form a 16-bit register, and contain the extracted "PB" (Phase Bin) from CMD=RRC (110) DumpTrig Frames for CTRL6_7 into the FPGA. If both of these registers are zero (0x0000), then it indicates that no PB has been extracted, due to the lack of received RRC frames or due to CRC errors detected in DumpTrig RRC frames (i.e. if an RRC frame is detected, and it contains a CRC error, this register is cleared).

5.4.1.38 DUMMY Register (R/W) Addr=0x3F

This is an 8-bit dummy read/write register and has no function in the chip.

5.4.2 Summary of Config/Status Registers and Addresses

Table 5-10 is a summary of all MCB registers and their associated addresses.

Register	Page	R/W?	Address	Description
MCSR	29	R/W	0x00	Master Control/Status Register
TSTAMP (0)	31	R	0x01	Timestamp Reg: COUNTPPS[7:0]
TSTAMP (1)		R	0x02	Timestamp Reg: COUNTPPS[15:8]
TSTAMP (2)		R	0x03	Timestamp Reg: COUNTPPS[23:16]
TSTAMP (3)		R	0x04	Timestamp Reg: COUNTPPS[31:24]
TSTAMP (4)		R	0x05	Timestamp Reg: COUNTMS[7:0]
TSTAMP (5)		R	0x06	Timestamp Reg: EPOCH,COUNTMS
CSSR	32	R/W	0x07	Control Input Select/Status Register
LCSR	33	R/W	0x08	Lock Control/Status Register
DRSR	36	R	0x09	Dump Request Status Register
SSIDSR	37	R/W	0x0A	Sample Stream ID Select Register
SSIDR	38	R	0x0B	Sample Stream ID Register
SSCR	39	R/W	0x0C	Sample Size Configuration Register
ISMCR0_1	40	R/W	0x0D	Internal Stream Mapping Ctrl Register 0_1
ISMCR2_3		R/W	0x0E	Internal Stream Mapping Ctrl Register 2_3
ISMCR4_5		R/W	0x0F	Internal Stream Mapping Ctrl Register 4_5
ISMCR6_7		R/W	0x10	Internal Stream Mapping Ctrl Register 6_7
SRCR0_1	42	R/W	0x11	Sample Rate Configuration Register 0_1
SRCR2_3		R/W	0x12	Sample Rate Configuration Register 2_3
SRCR4_5		R/W	0x13	Sample Rate Configuration Register 4_5
SRCR6_7		R/W	0x14	Sample Rate Configuration Register 6_7
RBLKSIZE	43	R/W	0x15	Recirculation Block Size Register
RBLK_START	45	R/W	0x16	Recirculation Start Block Register
RBLK_END	46	R/W	0x17	Recirculation End Block Register
RBLK_TOT	47	R/W	0x18	Total Recirculation Blocks Register
PMSR	48	R	0x19	PHASEMOD Detect Status Register
RSRCR	49	R/W	0x1B	Recirculation Sample Rate Config. Reg.
RSSR0_1	51	R/W	0x1C	Recirculation Stream 0 and 1 Select Reg.
RSSR2_3	53	R/W	0x1D	Recirculation Stream 2 and 3 Select Reg.
RSSR4_5	53	R/W	0x3C	Recirculation Stream 4 and 5 Select Reg.
RSSR6_7	53	R/W	0x3D	Recirculation Stream 6 and 7 Select Reg.
AMSR	56	R/W	0x26	Autocorrelator Mode Select Register
ODSCR	54	R/W	0x1E	Output Data Switch Config. Reg.
BFCR	57	R/W	0x22	Bitstream Force Configuration Register
AMUX	58	R/W	0x27	Analog mux selection register (AMUX)
HDLSR	59	R	0x2B	HM Data Lock Status Register
CLKPH-0	60	R/W	0x31	Corr Chip Clock Phase Control Register 0
CLKPH-1	60	R/W	0x32	Corr Chip Clock Phase Control Register 1
RECIRC_TYPE	61	R/W	0x33	Recirc FPGA Type and Y Tx control
OEDR	62	R/W	0x34	Output Enable/Disable Register
FCPCR	63	R/W	0x35	Fine Clock Phase Calibration Register
PRSR	65	R/W	0x36	PHASERR Read Select Register
PRR	65	R	0x37	PHASERR Read Register
DT_RRC_PBID_0_1	66	R	0x20, 0x21	DumpTrig RRC PBID CTRL0_1
DT_RRC_PBID_2_3	66	R	0x23, 0x24	DumpTrig RRC PBID CTRL2_3
DT_RRC_PBID_4_5	66	R	0x2C, 0x2D	DumpTrig RRC PBID CTRL4_5
DT_RRC_PBID_6_7	66	R	0x2E, 0x2F	DumpTrig RRC PBID CTRL6_7
DUMMY	67	R/W	0x3F	Dummy Read/Write register

Table 5-10 MCB configuration and status register summary

5.4.3 Additional Interface Requirements

The Recirculation Controller is an FPGA and will require a configuration bitstream to load the logic configuration into it. This will probably be done with 8-bit configuration pins. The Boundary scan/JTAG interface pins (TDI, TDO, TMS, and TCK) will be used for boundary-scan testing only.

6 Functional Specifications

This section presents a plan for implementing the functionality of the Recirculation Controller chip. This plan consists of a more detailed block diagram (Figure 6-1) and description of the functions of the chip.

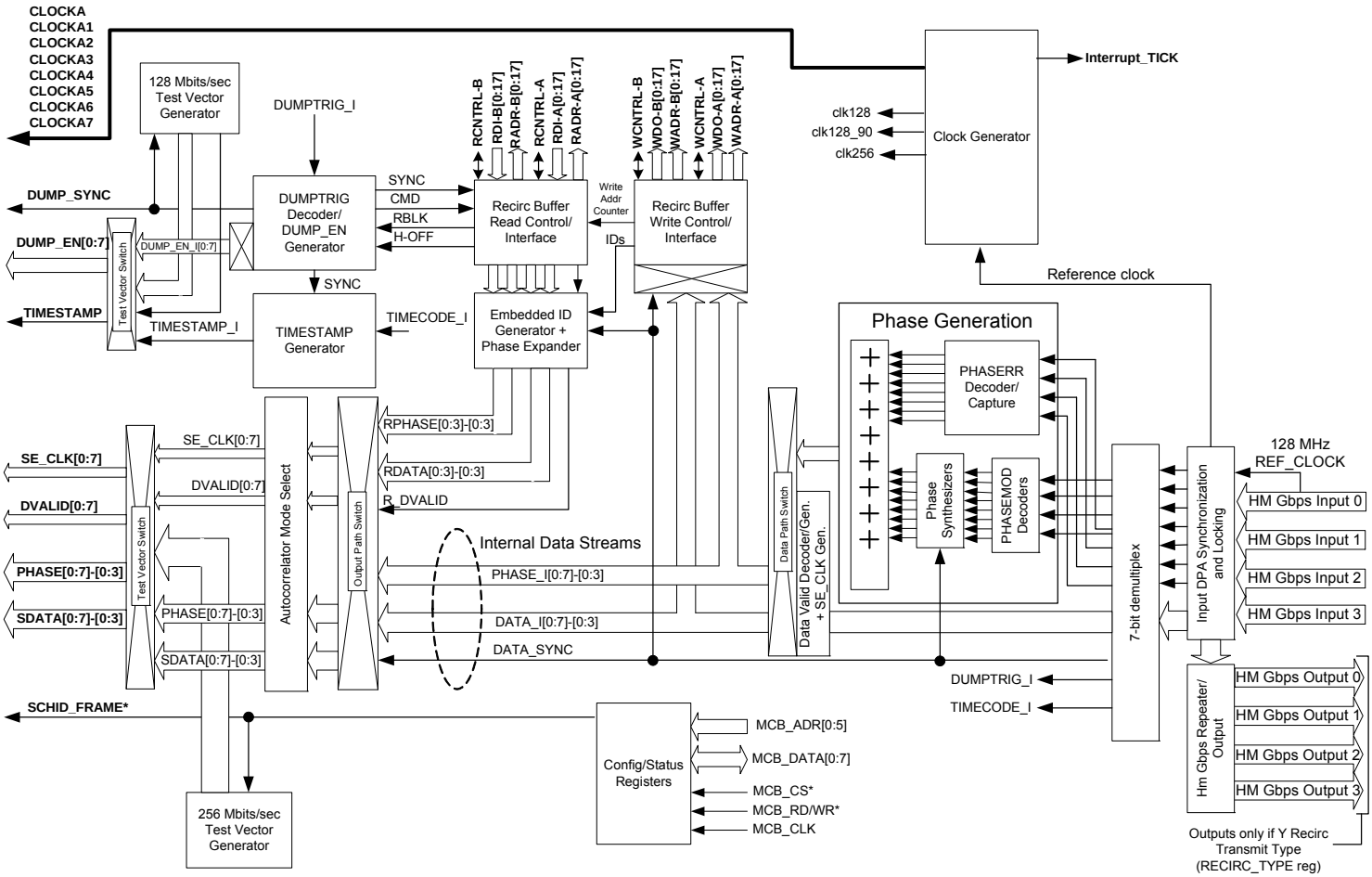


Figure 6-1 Top level block diagram of the Recirculation Controller FPGA. The device performs input signal de-skew, phase generation, recirculation, dump control, and timestamp generation functions. It also contains a test vector generator for checking connections to downstream Correlator Chips. Not shown is internal R2 recirculation RAM.

The general data flow of the chip is as follows.

1. Data and control signals from the HM Gbps Inputs 0-3 enter the chip (from the right in Figure 6-1), and are synchronized to the CPU-selected (but alternately determined internally if the CPU-selected CTRL input has lost sync) TIMECODE in the “Input DPA Synchronization and Locking” block. Note that the synchronization is automatic—it requires only that a microprocessor enable or disable locking, and wait until lock is established by monitoring error status registers. If the FPGA is Y Transmit Type, then these inputs are re-timed, and

transmitted to the Y ERNI connector to go to the next Baseline Board (if connected via a Patch Board or cable). Alternatively, if the FPGA is Y Receive Type, then the HM Gbps inputs come from the Y ERNI connector, rather than from the RXP FPGA.

2. 7-bit streams are extracted and mapped to internal 4 bit streams with a cross-bar switch (shown in Figure 6-1 as the "Data Path Switch"). If a 7-bit stream is defined on the input, it requires two internal 4-bit streams for processing.

3. The Phase Generation block generates the real-time quantized phase models for each of the (up to) 8 data streams. These models contain the phase generated from the point-slope phase models contained in PHASEMODs, added to the real-time PHASERR values that perform the very fine delay tracking.

4. The Recirculation Buffer Write Control/Interface block contains a cross-bar switch and the write control function that allows up to 4 of the internal 4-bit data streams to undergo recirculation (for simplicity, R2 functionality is not shown, but is always active, using internal memory—not shown in the figure—for internal streams 4-7). The write control function simply writes data into the recirculation DPSRAM at the sample stream data rate, and captures the write pointer address when requested by the read controller. All phase is serialized into one stream for writing into the recirculation buffer. Serialization and de-serialization of phase is handled in such a way so that there is a predictable sample-rate dependent phase offset introduced: **phase must be calculated for 3.5 sample times later than it otherwise would be calculated for, for either dynamic or static recirculation.** Refer to [3] for a more detailed description of recirculation write functions.

5. The Recirculation Buffer Read Control/Interface block contains most of the recirculation control logic (for simplicity, R2 functionality is not shown). It is responsible for calculating the start read address based on the block size and the current block from the block counter, for generating the correlation "holdoff" signal, and for starting the recirculation read burst. All of this activity is synchronized to DUMPTRIG. Refer to [3] for a more detailed description of recirculation read functions.

6. The Embedded ID Generator + Phase Expander block inserts embedded ID information into the recirculated streams that are required for the Correlator Chip. Because of the delay through the recirculation buffer, these IDs will not be in their original locations (i.e. where they already are). This block flags as invalid, those parts of the stream that contain the original embedded ID information and inserts them at the correct place relative to the other internal streams that were not recirculated. This block also de-multiplexes the single stream containing all of the phase information into the 4, 4-bit wide phase streams. The output of this block contains data streams that look exactly like any other stream, only they contain recirculated data.

7. The Data Valid Decoder/Generator and Shift Enable Clock Generator decodes data valid signaling that is encoded in the data [1] and generates explicit data valid lines. This block also generates the shift enable clocks that go to the Correlator Chip, and effectively define the sample rate in explicit terms for each data stream.

8. The Data Path Switch following the above block allows each output stream to be connected to either the same internal data stream or a recirculation stream. This is not a full cross-bar function since this is already provided by the Data Path Switch.

9. The Autocorrelator Mode Select block allows each stream to be selectively enabled for Correlator Chip autocorrelator operation or not. In autocorrelator mode, the PHASE stream associated with the DATA stream contains data delayed by 1024 samples such that—and working in concert with autocorrelator mode in the Correlator Chip—the peak of the autocorrelation function is at lag 0 of CCC-15 of the Correlator Chip. Also, data invalid is encoded in the PHASE stream as –8 (1000b), and is decoded inside the Correlator Chip to generate an internal data valid signal. **This means that autocorrelator mode works only on 4-bit data**.

10. The TIMESTAMP Generator decodes the selected TIMECODE signal and contains internal counters that are used to generate the TIMESTAMP signal going to the Correlator Chip when a dump occurs. This block also captures timing information in TIMECODE and provides it to a reading microprocessor via the TSTAMP registers.

11. The DUMPTRIG Decoder decodes DUMPTRIG and generates the DUMP_SYNC and DUMP_EN signals that go to the Correlator Chip. The switch on the output of this block mirrors the setting of the output Data Path Switch, so that DUMP_EN signaling applies to the appropriate output data stream.

12. The Test Vector Switches allow on-board generated test patterns to be connected to the outputs for off-line testing of connectivity to the attached Correlator Chips. These switches also can force each output signal to 0 or 1 to ensure that the Correlator Chips really do pick up pattern errors.

The Clock Generator module generates a number of clocks from the 128 MHz REF_CLOCK input. The phase of REF_CLOCK at each Recirculation Controller w.r.t. other Recirculation Controllers is arbitrary since final alignment to one clock domain is performed in the Correlator Chip. The generated clocks are as follows:

1. On-chip 256 and 128 MHz clocks.

2. Off-chip 32 MHz clocks (CLOCKA, CLOCKA1, …, CLOCKA7), frequency synchronous to REF_CLOCK. These clocks are transmitted to Correlator Chips in a point-to-point fashion, and each one is phase adjustable in 125 psec steps. Refer to Figure 5-1 of the Correlator Chip RFS document [2] for functional timing information.

The Interrupt_TICK is extracted from the TIMECODE ‘T’ bit and is a 30 nsec wide pulse synchronous to MCB_CLK that is asserted every 10 milliseconds. This tick is used to eventually generate an interrupt to a controlling microprocessor.

7 References

[1] Carlson, Brent, PROTOCOL SPECIFICATION, HM Gbps Cable Signaling Specification, Revision 1.2, August 18, 2005.

[2] Carlson, Brent, REQUIREMENTS AND FUNCTIONAL SPECIFICATION: EVLA Correlator Chip, RFS Document number A25082N0000, Rev. 2.4, February 15, 2005.

[3] Carlson, Brent, Refined EVLA WIDAR Correlator Architecture, NRC-EVLA Memo# 014, October 2, 2001.

[4] Carlson, Brent, EVLA 'WIDAR' Correlator Description for the Preliminary Design Review, NRC-EVLA Memo# 024, June 17, 2005.

[5] Carlson, Brent, An Optimized Connectivity Scheme for the EVLA Correlator, NRC-EVLA Memo# 028, April 13, 2007.

[6] Carlson, Brent, Programmer's Guide to EVLA Correlator System Timing, Synchronization, Data Products, and Operation, User Manual Document number A25290N0000, *Revision update in progress*.

8 Appendix I – Recirculation External RAM Test Registers

The Recirculation FPGA connects to two external DPSRAMs, “RAM A”, and “RAM B”. To facilitate better testability, a separate FPGA personality (bb_recirc_ramtest.bin) which allows the CPU to perform a memory test on this RAM has been developed. When this FPGA personality is loaded, normal Recirc FPGA registers and functions are completely non-existent, and the following registers are active:

```
// Addr          Function
// =====         ==========
// 0x00          RAM A wr addr [7:0]
// 0x01          RAM A wr addr [15:8]
// 0x02          RAM A wr addr [17:16] (in bits 1:0; others ignored)
// 0x03          RAM A wr data reg [7:0]
// 0x04          RAM A wr data reg [15:8]
// 0x05          RAM A wr data reg [17:16] Writing this location causes the RAM
write to occur.

// 0x06          RAM A rd addr [7:0]
// 0x07          RAM A rd addr [15:8]
// 0x08          RAM A rd addr [17:16]
// 0x09          RAM A rd data reg [7:0]
// 0x0A          RAM A rd data reg [15:8]
// 0x0B          RAM A rd data reg [17:16]

// 0x10-0x18 -- Same as above except it is for RAM B.
```

The procedure for **writing** to a RAM location is as follows:

1. Write the RAM test write ADDRESS into MCB addresses 0x00, 0x01, 0x02 (0x10, 0x11, 0x12 for RAM B).

2. Write the RAM test write DATA into MCB addresses 0x03, 0x04, 0x05 (0x13, 0x14, 0x15 for RAM B); writing to 0x05 causes the RAM write to occur.

The procedure for **reading** from a RAM location is as follows:

1. Write the RAM test read ADDRESS into MCB addresses 0x06, 0x07, 0x08 (… for RAM B).

2. Read the RAM test read DATA from MCB addresses 0x09, 0x0A, 0x0B (…for RAM B).

Both RAMs are 256k x 18. Note that for this test binary, the REF_CLOCK 128 MHz need not be available or accurate. All RAM read/write functions occur using the MCB_CLK 33 MHz clock.

9 Appendix II – Chip Pinouts, Pin, and Package Notes

This section contains exhaustive chip pinout listings generated by the FPGA compiler, and notes regarding pin connectivity and use on the board. Both X and Y Recirc FPGAs are implemented in Altera EP2S30F672C4 devices.

9.1 Recirc "X" FPGA Pinouts

Refer to section 4 for pin functionality descriptions.

Pin Name/Usage	Location	Dir.	I/O Standard	Voltage	I/O Bank
GND	A2	gnd			
rd_data_ramB_pad[3]	A3	input	3.3-V LVTTL		4
VCCIO4	A4	power		2.5V	4
rd_data_ramA_pad[7]	A5	input	3.3-V LVTTL		4
rd_data_ramB_pad[15]	A6	input	3.3-V LVTTL		4
rd_data_ramB_pad[8]	A7	input	3.3-V LVTTL		4
rd_data_ramB_pad[6]	A8	input	3.3-V LVTTL		4
rd_data_ramB_pad[11]	A9	input	3.3-V LVTTL		4
SDATA4_pad[0]	A10	output	2.5 V		4
VCCIO4	A11	power		2.5V	4
MCB_ADDR_pad[4]	A12	input	2.5 V		9
GND	A13	gnd			
GND	A14	gnd			
PHASE4_pad[0]	A15	output	2.5 V		3
VCCIO3	A16	power		2.5V	3
MCB_DATA_pad[3]	A17	bidir	2.5 V		3
rd_addr_ram_pad[1]	A18	output	2.5 V		3
GND*	A19				3
wr_addr_ram_pad[10]	A20	output	2.5 V		3
rd_addr_ram_pad[14]	A21	output	2.5 V		3
rd_addr_ram_pad[13]	A22	output	2.5 V		3
VCCIO3	A23	power		2.5V	3
rd_addr_ram_pad[8]	A24	output	2.5 V		3
GND	A25	gnd			
DVALID_pad[5]	AA1	output	2.5 V		6
PHASE0_pad[1]	AA2	output	2.5 V		6
UNUSED_LVDS[3]	AA3	input	3.3-V LVTTL		6
GND*	AA4				6
GND*	AA5				6
GND*	AA6				6
GND	AA7	gnd			
GND*	AA8				7
GND*	AA9				7
PHASE3_pad[2]	AA10	output	2.5 V		7
SDATA5_pad[1]	AA11	output	2.5 V		7
DUMP_EN_pad[1]	AA12	output	2.5 V		7
GND	AA13	gnd			
SDATA6_pad[2]	AA14	output	2.5 V		8
DUMP_SYNC_pad	AA15	output	2.5 V		8
PHASE3_pad[3]	AA16	output	2.5 V		8
PHASE6_pad[3]	AA17	output	2.5 V		8
RESERVE_HIZ_pad[4]	AA18	input	3.3-V LVTTL		8
GND*	AA19				8
nCONFIG	AA20				8
GND*	AA21				1
GND*	AA22				1
GND*	AA23				1
UNUSED_LVDS[7]	AA24	input	3.3-V LVTTL		1
BB_pad[3](n)	AA25	input	LVDS		1
BB_pad[3]	AA26	input	LVDS		1
PHASE2_pad[2]	AB1	output	2.5 V		6
DVALID_pad[1]	AB2	output	2.5 V		6
GND*	AB3				6

GND*	AB4					6
nCEO	AB5					7
PLL_ENA	AB6					7
GND*	AB7					7
RESERVE_HIZ_pad[12]	AB8	input	3.3-V LVTTL			7
AMUX_ADDR_pad[0]	AB9	output	2.5 V			7
PHASE6_pad[2]	AB10	output	2.5 V			7
PHASE4_pad[2]	AB11	output	2.5 V			7
SDATA3_pad[0]	AB12	output	2.5 V			7
DVALID_pad[0]	AB13	output	2.5 V			7
SDATA1_pad[0]	AB14	output	2.5 V			8
SE_CLK_pad[3]	AB15	output	2.5 V			8
PHASE7_pad[0]	AB16	output	2.5 V			8
rd_addr_ram_pad[0]	AB17	output	2.5 V			8
SDATA2_pad[2]	AB18	output	2.5 V			8
GND*	AB19					8
GND*	AB20					8
TRST	AB21	input				8
TCK	AB22	input				8
GND*	AB23					1
GND*	AB24					1
CTRL6_7_pad(n)	AB25	input	LVDS			1
CTRL6_7_pad	AB26	input	LVDS			1
VCCIO6	AC1	power			2.5V	6
GND*	AC2					6
GND*	AC3					6
GND*	AC4					7
VREFB7	AC5	power				7
INTERRUPT_TICK_pad	AC6	output	2.5 V			7
PHASE6_pad[1]	AC7	output	2.5 V			7
SDATA2_pad[3]	AC8	output	2.5 V			7
DUMP_EN_pad[5]	AC9	output	2.5 V			7
SDATA6_pad[1]	AC10	output	2.5 V			7
VREFB7	AC11	power				7
SDATA5_pad[3]	AC12	output	2.5 V			10
DUMP_EN_pad[2]	AC13	output	2.5 V			7
SE_CLK_pad[7]	AC14	output	2.5 V			8
UNUSED_LVDS[8]	AC15	input	3.3-V LVTTL			8
VREFB8	AC16	power				8
DUMP_EN_pad[4]	AC17	output	2.5 V			8
DVALID_pad[7]	AC18	output	2.5 V			8
UNUSED_CLOCK_TERM[0]	AC19	output	2.5 V			8
GND*	AC20					8
GND*	AC21					8
VREFB8	AC22	power				8
RESERVE_HIZ_pad[13]	AC23	input	3.3-V LVTTL			8
CTRL2_3_pad(n)	AC24	input	LVDS			1
CTRL2_3_pad	AC25	input	LVDS			1
VCCIO1	AC26	power			2.5V	1
GND*	AD1					6
GND*	AD2					6
AMUX_ADDR_pad[1]	AD3	output	2.5 V			7
AMUX_ADDR_pad[2]	AD4	output	2.5 V			7
PHASE7_pad[2]	AD5	output	2.5 V			7
SDATA4_pad[2]	AD6	output	2.5 V			7
PHASE6_pad[0]	AD7	output	2.5 V			7
SDATA2_pad[0]	AD8	output	2.5 V			7
PHASE2_pad[1]	AD9	output	2.5 V			7
SDATA2_pad[1]	AD10	output	2.5 V			7
SDATA7_pad[0]	AD11	output	2.5 V			7
SDATA0_pad[0]	AD12	output	2.5 V			10
SDATA0_pad[2]	AD13	output	2.5 V			10
SE_CLK_pad[0]	AD14	output	2.5 V			7
GND*	AD15					8
SDATA7_pad[3]	AD16	output	2.5 V			8
DVALID_pad[4]	AD17	output	2.5 V			8
DVALID_pad[3]	AD18	output	2.5 V			8
DUMP_EN_pad[3]	AD19	output	2.5 V			8
rd_addr_ram_pad[9]	AD20	output	2.5 V			8
rd_addr_ram_pad[11]	AD21	output	2.5 V			8
GND*	AD22					8

GND*	AD23				8	
TMS	AD24	input			8	
CTRL4_5_pad(n)	AD25	input	LVDS		1	
CTRL4_5_pad	AD26	input	LVDS		1	
GND	AE1	gnd				
nIO_PULLUP	AE2				7	
AMUX_ADDR_pad[3]	AE3	output	2.5 V		7	
PHASE0_pad[2]	AE4	output	2.5 V		7	
PHASE1_pad[3]	AE5	output	2.5 V		7	
SDATA7_pad[1]	AE6	output	2.5 V		7	
PHASE7_pad[3]	AE7	output	2.5 V		7	
SE_CLK_pad[4]	AE8	output	2.5 V		7	
SDATA6_pad[3]	AE9	output	2.5 V		7	
SDATA4_pad[3]	AE10	output	2.5 V		7	
PHASE1_pad[1]	AE11	output	2.5 V		7	
SDATA1_pad[1]	AE12	output	2.5 V		10	
DVALID_pad[2]	AE13	output	2.5 V		10	
DUMP_EN_pad[6]	AE14	output	2.5 V		7	
SE_CLK_pad[1]	AE15	output	2.5 V		8	
SDATA3_pad[3]	AE16	output	2.5 V		8	
SE_CLK_pad[6]	AE17	output	2.5 V		8	
TIMESTAMP_pad	AE18	output	2.5 V		8	
PHASE4_pad[1]	AE19	output	2.5 V		8	
rd_addr_ram_pad[6]	AE20	output	2.5 V		8	
rd_addr_ram_pad[7]	AE21	output	2.5 V		8	
GND*	AE22				8	
GND*	AE23				8	
GND*	AE24				8	
TDI	AE25	input			8	
GND	AE26	gnd				
GND	AF2	gnd				
AMUX_ADDR_pad[4]	AF3	output	2.5 V		7	
VCCIO7	AF4	power		2.5V	7	
SDATA3_pad[1]	AF5	output	2.5 V		7	
PHASE4_pad[3]	AF6	output	2.5 V		7	
PHASE2_pad[3]	AF7	output	2.5 V		7	
SDATA3_pad[2]	AF8	output	2.5 V		7	
DUMP_EN_pad[7]	AF9	output	2.5 V		7	
SDATA6_pad[0]	AF10	output	2.5 V		7	
VCCIO7	AF11	power		2.5V	7	
SE_CLK_pad[2]	AF12	output	2.5 V		10	
GND	AF13	gnd				
GND	AF14	gnd				
SE_CLK_pad[5]	AF15	output	2.5 V		8	
VCCIO8	AF16	power		2.5V	8	
SDATA0_pad[3]	AF17	output	2.5 V		8	
SCHID_FRAME_pad_	AF18	output	2.5 V		8	
CLOCKA2_pad	AF19	output	2.5 V		8	
DUMP_EN_pad[0]	AF20	output	2.5 V		8	
rd_addr_ram_pad[4]	AF21	output	2.5 V		8	
GND*	AF22				8	
VCCIO8	AF23	power		2.5V	8	
GND*	AF24				8	
GND	AF25	gnd				
GND	B1	gnd				
MSEL1	B2				4	
AMUX_WR_pad_	B3	output	2.5 V		4	
rd_data_ramB_pad[0]	B4	input	3.3-V LVTTL		4	
rd_data_ramA_pad[14]	B5	input	3.3-V LVTTL		4	
rd_data_ramB_pad[10]	B6	input	3.3-V LVTTL		4	
rd_data_ramA_pad[8]	B7	input	3.3-V LVTTL		4	
rd_data_ramA_pad[6]	B8	input	3.3-V LVTTL		4	
rd_data_ramB_pad[5]	B9	input	3.3-V LVTTL		4	
SDATA5_pad[0]	B10	output	2.5 V		4	
SDATA5_pad[2]	B11	output	2.5 V		4	
MCB_ADDR_pad[0]	B12	input	2.5 V		9	
SDATA4_pad[1]	B13	output	2.5 V		4	
MCB_ADDR_pad[2]	B14	input	2.5 V		4	
MCB_DATA_pad[7]	B15	bidir	2.5 V		3	
dpsram_clockB_pad	B16	output	2.5 V		3	
rd_data_ramA_pad[1]	B17	input	3.3-V LVTTL		3	

GND*	B18					3
wr_addr_ram_pad[5]	B19	output	2.5 V			3
rd_addr_ram_pad[10]	B20	output	2.5 V			3
rd_addr_ram_pad[2]	B21	output	2.5 V			3
CLOCKA6_pad	B22	output	2.5 V			3
GND*	B23					3
rd_addr_ram_pad[16]	B24	output	2.5 V			3
CONF_DONE	B25					3
GND	B26	gnd				
GND*	C1					5
GND*	C2					5
AMUX_EN0_pad_	C3	output	2.5 V			4
rd_data_ramB_pad[16]	C4	input	3.3-V LVTTL			4
rd_data_ramA_pad[3]	C5	input	3.3-V LVTTL			4
rd_data_ramA_pad[16]	C6	input	3.3-V LVTTL			4
rd_data_ramA_pad[10]	C7	input	3.3-V LVTTL			4
PHASE0_pad[0]	C8	output	2.5 V			4
rd_data_ramA_pad[5]	C9	input	3.3-V LVTTL			4
rd_data_ramA_pad[0]	C10	input	3.3-V LVTTL			4
rd_data_ramA_pad[17]	C11	input	3.3-V LVTTL			4
CLOCKA5_pad	C12	output	2.5 V			9
MCB_RD_WR_pad_	C13	input	2.5 V			4
MCB_ADDR_pad[5]	C14	input	2.5 V			4
SDATA0_pad[1]	C15	output	2.5 V			3
MCB_DATA_pad[1]	C16	bidir	2.5 V			3
rd_data_ramB_pad[13]	C17	input	3.3-V LVTTL			3
wr_addr_ram_pad[12]	C18	output	2.5 V			3
GND*	C19					3
rd_addr_ram_pad[5]	C20	output	2.5 V			3
rd_addr_ram_pad[3]	C21	output	2.5 V			3
GND*	C22					3
GND*	C23					3
DCLK	C24					3
GND*	C25					2
GND*	C26					2
VCCIO5	D1	power			2.5V	5
GND*	D2					5
GND*	D3					5
GND*	D4					4
VREFB4	D5	power				4
rd_data_ramA_pad[9]	D6	input	3.3-V LVTTL			4
rd_data_ramA_pad[15]	D7	input	3.3-V LVTTL			4
rd_data_ramB_pad[4]	D8	input	3.3-V LVTTL			4
rd_data_ramA_pad[4]	D9	input	3.3-V LVTTL			4
rd_data_ramA_pad[12]	D10	input	3.3-V LVTTL			4
VREFB4	D11	power				4
rd_data_ramB_pad[12]	D12	input	3.3-V LVTTL			9
MCB_ADDR_pad[3]	D13	input	2.5 V			9
GND*	D14					3
MCB_DATA_pad[4]	D15	bidir	2.5 V			3
VREFB3	D16	power				3
rd_data_ramA_pad[2]	D17	input	3.3-V LVTTL			3
rd_addr_ram_pad[12]	D18	output	2.5 V			3
AMUX_EN1_pad_	D19	output	2.5 V			3
GND*	D20					3
RESERVE_HIZ_pad[1]	D21	input	3.3-V LVTTL			3
VREFB3	D22	power				3
GND*	D23					3
RESERVE_HIZ_pad[19]	D24	input	3.3-V LVTTL			2
GND*	D25					2
VCCIO2	D26	power			2.5V	2
GND*	E1					5
GND*	E2					5
CLOCKA7_pad	E3	output	2.5 V			5
RESERVE_HIZ_pad[21]	E4	input	3.3-V LVTTL			5
TEMPDIODEp	E5					
MSEL2	E6					4
rd_data_ramB_pad[2]	E7	input	3.3-V LVTTL			4
CLOCKA4_pad	E8	output	2.5 V			4
rd_data_ramA_pad[11]	E9	input	3.3-V LVTTL			4
rd_data_ramB_pad[14]	E10	input	3.3-V LVTTL			4

rd_data_ramB_pad[7]	E11	input	3.3-V LVTTL		4	
rd_data_ramA_pad[13]	E12	input	3.3-V LVTTL		4	
MCB_ADDR_pad[1]	E13	input	2.5 V		9	
MCB_DATA_pad[2]	E14	bidir	2.5 V		3	
MCB_DATA_pad[0]	E15	bidir	2.5 V		3	
~DATA0~ / RESERVED_INPUT	E16	input	2.5 V		3	
wr_addr_ram_pad[16]	E17	output	2.5 V		3	
GND*	E18				3	
RESERVE_HIZ_pad[22]	E19	input	3.3-V LVTTL		3	
GND*	E20				3	
nSTATUS	E21				3	
nCE	E22				3	
GND*	E23				2	
GND*	E24				2	
BB_pad[1](n)	E25	input	LVDS		2	
BB_pad[1]	E26	input	LVDS		2	
GND*	F1				5	
GND*	F2				5	
GND*	F3				5	
GND*	F4				5	
TEMPDIODEn	F5					
TDO	F6	output			4	
MSEL3	F7				4	
GND*	F8				4	
GND*	F9				4	
rd_data_ramB_pad[9]	F10	input	3.3-V LVTTL		4	
rd_data_ramB_pad[17]	F11	input	3.3-V LVTTL		4	
UNUSED_LVDS[2]	F12	input	3.3-V LVTTL		4	
GND*	F13				4	
MCB_DATA_pad[5]	F14	bidir	2.5 V		3	
CLOCKA1_pad	F15	output	2.5 V		3	
MCB_DATA_pad[6]	F16	bidir	2.5 V		3	
wr_addr_ram_pad[6]	F17	output	2.5 V		3	
GND*	F18				3	
GND*	F19				3	
dpsram_clockA_pad	F20	output	2.5 V		3	
GND*	F21				2	
RESERVE_HIZ_pad[2]	F22	input	3.3-V LVTTL		2	
GND*	F23				2	
GND*	F24				2	
BB_pad[5](n)	F25	input	LVDS		2	
BB_pad[5]	F26	input	LVDS		2	
wr_data_ramA_pad[8]	G1	output	2.5 V		5	
wr_data_ramB_pad[5]	G2	output	2.5 V		5	
GND*	G3				5	
GND*	G4				5	
VREFB5	G5	power			5	
GND*	G6				5	
RESERVE_HIZ_pad[17]	G7	input	3.3-V LVTTL		5	
MSEL0	G8				4	
GND*	G9				4	
GND*	G10				4	
GND*	G11				4	
VCCA_PLL5	G12	power		1.2V		
GND	G13	gnd				
rd_data_ramB_pad[1]	G14	input	3.3-V LVTTL		3	
MCB_CS_pad_	G15	input	2.5 V		3	
PHASE5_pad[0]	G16	output	2.5 V		3	
rd_addr_ram_pad[17]	G17	output	2.5 V		3	
rd_addr_ram_pad[15]	G18	output	2.5 V		3	
GND*	G19				3	
GND*	G20				2	
GND*	G21				2	
VREFB2	G22	power			2	
GND*	G23				2	
GND*	G24				2	
BB_pad[4](n)	G25	input	LVDS		2	
BB_pad[4]	G26	input	LVDS		2	
wr_data_ramA_pad[4]	H1	output	2.5 V		5	
wr_addr_ram_pad[0]	H2	output	2.5 V		5	
GND*	H3				5	

wr_data_ramB_pad[17]	H4	output	2.5 V		5
GND*	H5				5
RESERVE_HIZ_pad[8]	H6	input	3.3-V LVTTL		5
GND*	H7				5
GND*	H8				5
GND*	H9				4
GND*	H10				4
GND*	H11				4
VCC_PLL5_OUT	H12	power		2.5V	9
GNDA_PLL5	H13	gnd			
VCCD_PLL5	H14	power		1.2V	
RESERVE_HIZ_pad[23]	H15	input	3.3-V LVTTL		3
GND*	H16				3
GND*	H17				3
GND*	H18				3
GND*	H19				2
GND*	H20				2
GND*	H21				2
GND*	H22				2
GND*	H23				2
GND*	H24				2
BB_pad[7](n)	H25	input	LVDS		2
BB_pad[7]	H26	input	LVDS		2
wr_addr_ram_pad[17]	J1	output	2.5 V		5
wr_addr_ram_pad[13]	J2	output	2.5 V		5
GND*	J3				5
wr_data_ramB_pad[6]	J4	output	2.5 V		5
GND*	J5				5
GND*	J6				5
GND*	J7				5
GND*	J8				5
UNUSED_LVDS[1]	J9	input	3.3-V LVTTL		4
UNUSED_LVDS[6]	J10	input	3.3-V LVTTL		4
GND*	J11				4
VCCPD4	J12	power		3.3V	4
GNDA_PLL5	J13	gnd			
GND*	J14				3
GND*	J15				3
VCCPD3	J16	power		3.3V	3
GND*	J17				3
GND*	J18				3
GND*	J19				2
GND*	J20				2
GND*	J21				2
GND*	J22				2
GND*	J23				2
GND*	J24				2
BB_pad[2](n)	J25	input	LVDS		2
BB_pad[2]	J26	input	LVDS		2
wr_addr_ram_pad[4]	K1	output	2.5 V		5
wr_addr_ram_pad[15]	K2	output	2.5 V		5
wr_data_ramA_pad[9]	K3	output	2.5 V		5
wr_addr_ram_pad[7]	K4	output	2.5 V		5
VREFB5	K5	power			5
GND*	K6				5
GND*	K7				5
GND*	K8				5
GND*	K9				5
GND*	K10				4
RESERVE_HIZ_pad[6]	K11	input	3.3-V LVTTL		4
GND	K12	gnd			
VCCIO4	K13	power		2.5V	4
GND	K14	gnd			
VCCIO3	K15	power		2.5V	3
GND*	K16				3
GND*	K17				3
GND*	K18				3
GND*	K19				2
RESERVE_HIZ_pad[16]	K20	input	3.3-V LVTTL		2
GND*	K21				2
GND*	K22				2

UNUSED_LVDS[10]	K23	input	3.3-V LVTTL		2	
GND*	K24				2	
BB_pad[0](n)	K25	input	LVDS		2	
BB_pad[0]	K26	input	LVDS		2	
VCCIO5	L1	power		2.5V	5	
wr_data_ramB_pad[16]	L2	output	2.5 V		5	
wr_data_ramA_pad[0]	L3	output	2.5 V		5	
wr_data_ramB_pad[12]	L4	output	2.5 V		5	
GND*	L5				5	
RESERVE_HIZ_pad[14]	L6	input	3.3-V LVTTL		5	
wr_data_ramA_pad[5]	L7	output	2.5 V		5	
UNUSED_LVDS[0]	L8	input	3.3-V LVTTL		5	
wr_addr_ram_pad[2]	L9	output	2.5 V		5	
VCCINT	L10	power		1.2V		
GND	L11	gnd				
VCCINT	L12	power		1.2V		
GND	L13	gnd				
VCCINT	L14	power		1.2V		
GND	L15	gnd				
VCCINT	L16	power		1.2V		
GND	L17	gnd				
GND*	L18				2	
GND*	L19				2	
GND*	L20				2	
GND*	L21				2	
GND*	L22				2	
GND*	L23				2	
GND*	L24				2	
GND*	L25				2	
VCCIO2	L26	power		2.5V	2	
wr_data_ramB_pad[0]	M1	output	2.5 V		5	
wr_data_ramB_pad[3]	M2	output	2.5 V		5	
GND*	M3				5	
GND*	M4				5	
wr_data_ramA_pad[16]	M5	output	2.5 V		5	
wr_addr_ram_pad[14]	M6	output	2.5 V		5	
wr_data_ramA_pad[17]	M7	output	2.5 V		5	
wr_data_ramA_pad[15]	M8	output	2.5 V		5	
VCCPD5	M9	power		3.3V	5	
GND	M10	gnd				
VCCINT	M11	power		1.2V		
GND	M12	gnd				
VCCINT	M13	power		1.2V		
GND	M14	gnd				
VCCINT	M15	power		1.2V		
GND	M16	gnd				
VCCIO2	M17	power		2.5V	2	
VCCPD2	M18	power		3.3V	2	
GND*	M19				2	
GND*	M20				2	
RESERVE_HIZ_pad[18]	M21	input	3.3-V LVTTL		2	
GND*	M22				2	
GND*	M23				2	
GND*	M24				2	
GND*	M25				2	
RESERVE_HIZ_pad[0]	M26	input	3.3-V LVTTL		2	
GND	N1	gnd				
RESET_pad_	N2	input	2.5 V		5	
GND+	N3				5	
wr_addr_ram_pad[11]	N4	output	2.5 V		5	
wr_data_ramB_pad[1]	N5	output	2.5 V		5	
wr_data_ramB_pad[4]	N6	output	2.5 V		5	
wr_data_ramA_pad[1]	N7	output	2.5 V		5	
GNDA_PLL4	N8	gnd				
GNDA_PLL4	N9	gnd				
VCCIO5	N10	power		2.5V	5	
GND	N11	gnd				
VCCINT	N12	power		1.2V		
GND	N13	gnd				
VCCINT	N14	power		1.2V		
GND	N15	gnd				

VCCINT	N16	power		1.2V		
GND	N17	gnd				
GNDA_PLL1	N18	gnd				
GND*	N19				2	
GND*	N20				2	
GND*	N21				2	
GND*	N22				2	
VREFB2	N23	power			2	
GND+	N24				2	
GND+	N25				2	
GND	N26	gnd				
GND	P1	gnd				
wr_addr_ram_pad[1]	P2	output	2.5 V		5	
wr_data_ramA_pad[14]	P3	output	2.5 V		5	
wr_data_ramA_pad[12]	P4	output	2.5 V		5	
wr_data_ramA_pad[10]	P5	output	2.5 V		5	
GND	P6	gnd				
VCCD_PLL3	P7	power		1.2V		
VCCA_PLL4	P8	power		1.2V		
VCCD_PLL4	P9	power		1.2V		
GND	P10	gnd				
VCCINT	P11	power		1.2V		
GND	P12	gnd				
VCCINT	P13	power		1.2V		
GND	P14	gnd				
VCCINT	P15	power		1.2V		
GND	P16	gnd				
VCCIO1	P17	power		2.5V	1	
GNDA_PLL1	P18	gnd				
VCCD_PLL1	P19	power		1.2V		
VCCD_PLL2	P20	power		1.2V		
VCCA_PLL1	P21	power		1.2V		
GND+	P22				1	
UNUSED_LVDS[4]	P23	input	3.3-V LVTTL		1	
GND*	P24				2	
RESERVE_HIZ_pad[15]	P25	input	3.3-V LVTTL		2	
GND	P26	gnd				
wr_data_ramA_pad[13]	R1	output	2.5 V		6	
wr_data_ramB_pad[9]	R2	output	2.5 V		6	
MCB_CLK_pad	R3	input	2.5 V		6	
GND+	R4				6	
VREFB6	R5	power			6	
VCCA_PLL3	R6	power		1.2V		
GNDA_PLL3	R7	gnd				
GNDA_PLL3	R8	gnd				
VCCPD6	R9	power		3.3V	6	
VCCIO6	R10	power		2.5V	6	
GND	R11	gnd				
VCCINT	R12	power		1.2V		
GND	R13	gnd				
VCCINT	R14	power		1.2V		
GND	R15	gnd				
VCCINT	R16	power		1.2V		
GND	R17	gnd				
VCCPD1	R18	power		3.3V	1	
GNDA_PLL2	R19	gnd				
GNDA_PLL2	R20	gnd				
VCCA_PLL2	R21	power		1.2V		
GND	R22	gnd				
GND*	R23				1	
GND*	R24				1	
REF_CLOCK_pad(n)	R25	input	LVDS		1	
REF_CLOCK_pad	R26	input	LVDS		1	
VCCIO6	T1	power		2.5V	6	
wr_addr_ram_pad[9]	T2	output	2.5 V		6	
wr_data_ramA_pad[6]	T3	output	2.5 V		6	
wr_addr_ram_pad[3]	T4	output	2.5 V		6	
wr_data_ramA_pad[11]	T5	output	2.5 V		6	
SDATA1_pad[3]	T6	output	2.5 V		6	
PHASE0_pad[3]	T7	output	2.5 V		6	
wr_data_ramA_pad[3]	T8	output	2.5 V		6	

wr_data_ramB_pad[7]	T9	output	2.5 V		6	
GND	T10	gnd				
VCCINT	T11	power		1.2V		
GND	T12	gnd				
VCCINT	T13	power		1.2V		
GND	T14	gnd				
VCCINT	T15	power		1.2V		
GND	T16	gnd				
VCCINT	T17	power		1.2V		
GND	T18	gnd				
GND*	T19				1	
GND*	T20				1	
GND*	T21				1	
GND*	T22				1	
VREFB1	T23	power			1	
GND*	T24				1	
RESERVE_HIZ_pad[7]	T25	input	3.3-V LVTTL		1	
VCCIO1	T26	power		2.5V	1	
wr_data_ramB_pad[13]	U1	output	2.5 V		6	
PHASE1_pad[0]	U2	output	2.5 V		6	
wr_data_ramA_pad[7]	U3	output	2.5 V		6	
CLOCKA_pad	U4	output	2.5 V		6	
PHASE3_pad[1]	U5	output	2.5 V		6	
SDATA7_pad[2]	U6	output	2.5 V		6	
wr_data_ramA_pad[2]	U7	output	2.5 V		6	
PHASE1_pad[2]	U8	output	2.5 V		6	
GND	U9	gnd				
VCCINT	U10	power		1.2V		
GND	U11	gnd				
VCCIO7	U12	power		2.5V	7	
GND	U13	gnd				
VCCIO8	U14	power		2.5V	8	
GND	U15	gnd				
VCCINT	U16	power		1.2V		
GND	U17	gnd				
VCCINT	U18	power		1.2V		
GND*	U19				1	
GND*	U20				1	
GND*	U21				1	
GND*	U22				1	
GND*	U23				1	
GND*	U24				1	
GND*	U25				1	
GND*	U26				1	
wr_data_ramB_pad[8]	V1	output	2.5 V		6	
wr_data_ramB_pad[10]	V2	output	2.5 V		6	
SDATA1_pad[2]	V3	output	2.5 V		6	
wr_data_ramB_pad[2]	V4	output	2.5 V		6	
wr_data_ramB_pad[11]	V5	output	2.5 V		6	
PHASE7_pad[1]	V6	output	2.5 V		6	
GND*	V7				6	
GND*	V8				6	
VCCINT	V9	power		1.2V		
UNUSED_LVDS[9]	V10	input	3.3-V LVTTL		7	
VCCPD7	V11	power		3.3V	7	
GND*	V12				7	
VCC_PLL6_OUT	V13	power		2.5V	10	
GND*	V14				8	
VCCPD8	V15	power		3.3V	8	
GND*	V16				8	
GND*	V17				8	
GND*	V18				8	
GND*	V19				1	
GND*	V20				1	
GND*	V21				1	
GND*	V22				1	
GND*	V23				1	
GND*	V24				1	
RESERVE_HIZ_pad[5]	V25	input	3.3-V LVTTL		1	
GND*	V26				1	
wr_data_ramB_pad[15]	W1	output	2.5 V		6	

```
wr_data_ramB_pad[14]          : W2        : output : 2.5 V             :          : 6         :
PHASE3_pad[0]                 : W3        : output : 2.5 V             :          : 6         :
PHASE5_pad[2]                 : W4        : output : 2.5 V             :          : 6         :
GND*                          : W5        :        :                   :          : 6         :
GND*                          : W6        :        :                   :          : 6         :
GND*                          : W7        :        :                   :          : 6         :
GND*                          : W8        :        :                   :          : 6         :
GND*                          : W9        :        :                   :          : 7         :
RESERVE_HIZ_pad[3]            : W10       : input  : 3.3-V LVTTL       :          : 7         :
UNUSED_LVDS[11]               : W11       : input  : 3.3-V LVTTL       :          : 7         :
GND*                          : W12       :        :                   :          : 7         :
GNDA_PLL6                     : W13       : gnd    :                   :          :           :
GNDA_PLL6                     : W14       : gnd    :                   :          :           :
GND*                          : W15       :        :                   :          : 8         :
UNUSED_LVDS[5]                : W16       : input  : 3.3-V LVTTL       :          : 8         :
GND*                          : W17       :        :                   :          : 8         :
GND*                          : W18       :        :                   :          : 8         :
RESERVE_HIZ_pad[11]           : W19       : input  : 3.3-V LVTTL       :          : 1         :
RESERVE_HIZ_pad[10]           : W20       : input  : 3.3-V LVTTL       :          : 1         :
GND*                          : W21       :        :                   :          : 1         :
RESERVE_HIZ_pad[20]           : W22       : input  : 3.3-V LVTTL       :          : 1         :
GND*                          : W23       :        :                   :          : 1         :
GND*                          : W24       :        :                   :          : 1         :
CTRL0_1_pad(n)                : W25       : input  : LVDS              :          : 1         :
CTRL0_1_pad                   : W26       : input  : LVDS              :          : 1         :
wr_addr_ram_pad[8]            : Y1        : output : 2.5 V             :          : 6         :
PHASE2_pad[0]                 : Y2        : output : 2.5 V             :          : 6         :
GND*                          : Y3        :        :                   :          : 6         :
GND*                          : Y4        :        :                   :          : 6         :
VREFB6                        : Y5        : power  :                   :          : 6         :
UNUSED_CLOCK_TERM[1]          : Y6        : output : 2.5 V             :          : 6         :
GND*                          : Y7        :        :                   :          : 6         :
PORSEL                        : Y8        :        :                   :          : 7         :
GND*                          : Y9        :        :                   :          : 7         :
PHASE5_pad[1]                 : Y10       : output : 2.5 V             :          : 7         :
GND*                          : Y11       :        :                   :          : 7         :
GND*                          : Y12       :        :                   :          : 7         :
VCCA_PLL6                     : Y13       : power  :                   : 1.2V     :           :
VCCD_PLL6                     : Y14       : power  :                   : 1.2V     :           :
DVALID_pad[6]                 : Y15       : output : 2.5 V             :          : 8         :
CLOCKA3_pad                   : Y16       : output : 2.5 V             :          : 8         :
PHASE5_pad[3]                 : Y17       : output : 2.5 V             :          : 8         :
GND*                          : Y18       :        :                   :          : 8         :
VCCSEL                        : Y19       :        :                   :          : 8         :
RESERVE_HIZ_pad[9]            : Y20       : input  : 3.3-V LVTTL       :          : 1         :
GND*                          : Y21       :        :                   :          : 1         :
VREFB1                        : Y22       : power  :                   :          : 1         :
GND*                          : Y23       :        :                   :          : 1         :
GND*                          : Y24       :        :                   :          : 1         :
BB_pad[6](n)                  : Y25       : input  : LVDS              :          : 1         :
BB_pad[6]                     : Y26       : input  : LVDS              :          : 1         :
-------------------------------------------------------------------------------------------------
 -- NC           : No Connect. This pin has no internal connection to the device.
 -- DNU          : Do Not Use. This pin MUST NOT be connected.
 -- VCCINT       : Dedicated power pin, which MUST be connected to VCC  (1.2V).
 -- VCCIO        : Dedicated power pin, which MUST be connected to VCC
 --                of its bank.
 --                                          Bank 1:            2.5V
 --                                          Bank 2:            2.5V
 --                                          Bank 3:            2.5V
 --                                          Bank 4:            2.5V
 --                                          Bank 5:            2.5V
 --                                          Bank 6:            2.5V
 --                                          Bank 7:            2.5V
 --                                          Bank 8:            2.5V
 --                                          Bank 9:            2.5V
 --                                          Bank 10:  2.5V
 -- GND          : Dedicated ground pin. Dedicated GND pins MUST be connected to GND.
 -- GND+         : Unused input pin. It can also be used to report unused dual-purpose pins.
 --                                          This pin should be connected to GND.
 -- GND*         : Unused  I/O  pin
 -- RESERVED     : Unused I/O pin, which MUST be left unconnected.
 -- RESERVED_INPUT    : Pin is tri-stated and should be connected to the board.
 -- RESERVED_INPUT_WITH_WEAK_PULLUP    : Pin is tri-stated with internal weak pull-up resistor.
 -- RESERVED_INPUT_WITH_BUS_HOLD       : Pin is tri-stated with bus-hold circuitry.
 -- RESERVED_OUTPUT_DRIVEN_HIGH        : Pin is output driven high.
 -----------------------------------------------------------------------------------------------
```

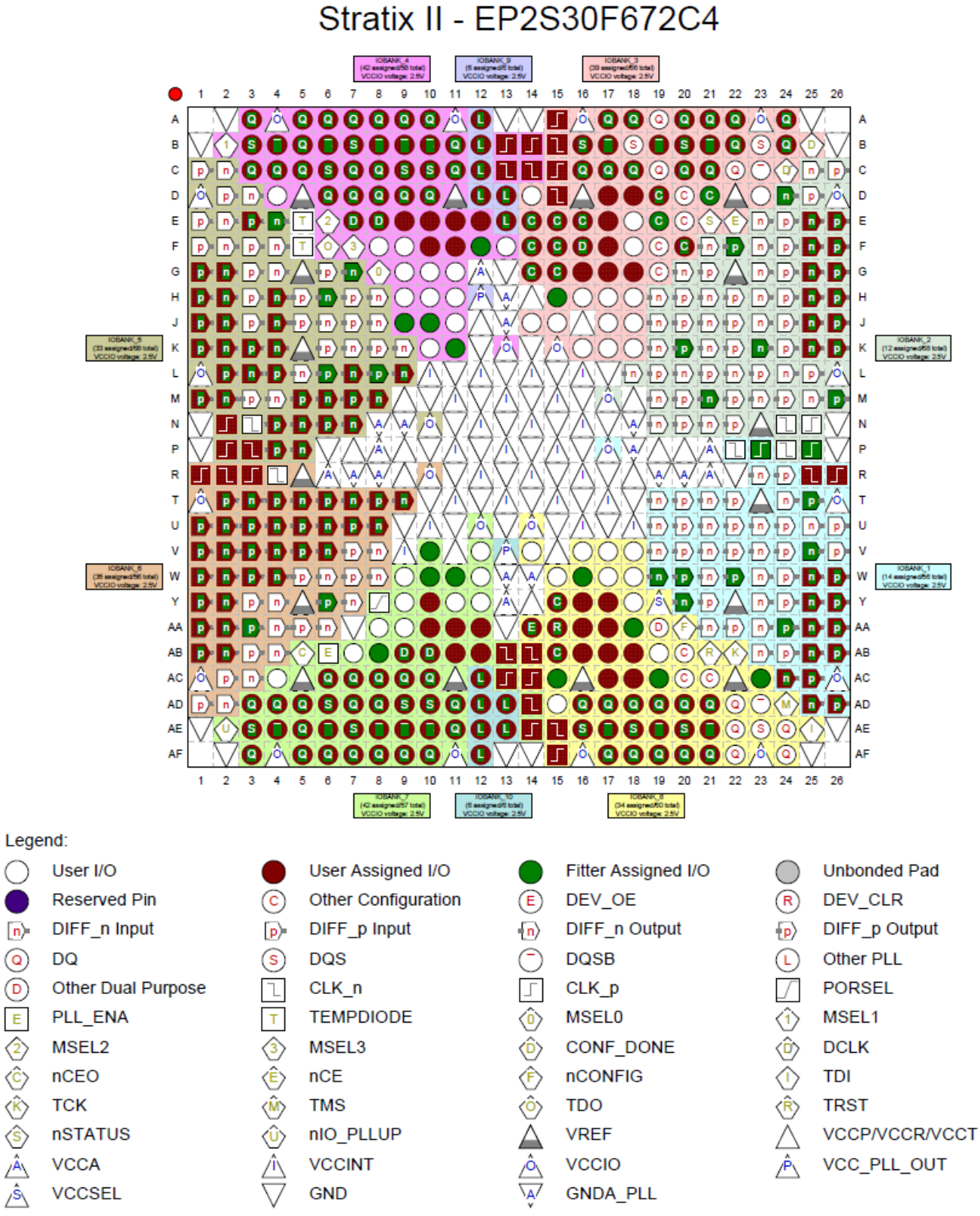


Figure 9-1 Quartus-II V9.1 pin planner output of the "X" Recirc FPGA.

9.2 Recirc “Ytx” FPGA Pinouts

This section contains chip pinouts for the “Ytx” Recirc FPGA that is normally used in the EVLA system (i.e. received HM Gbps data from the RXP FPGAs, copy/repeat them to the Y-ERNI connector output).

Refer to section 4 for pin functionality descriptions.

Pin Name/Usage	Location	Dir.	I/O Standard	Voltage	I/O Bank
GND	A2	gnd			
rd_data_ramB_pad[3]	A3	input	3.3-V LVTTL		4
VCCIO4	A4	power		2.5V	4
rd_data_ramA_pad[7]	A5	input	3.3-V LVTTL		4
rd_data_ramB_pad[15]	A6	input	3.3-V LVTTL		4
rd_data_ramB_pad[8]	A7	input	3.3-V LVTTL		4
rd_data_ramB_pad[6]	A8	input	3.3-V LVTTL		4
rd_data_ramB_pad[11]	A9	input	3.3-V LVTTL		4
SDATA4_pad[0]	A10	output	2.5 V		4
VCCIO4	A11	power		2.5V	4
MCB_ADDR_pad[4]	A12	input	2.5 V		9
GND	A13	gnd			
GND	A14	gnd			
PHASE4_pad[0]	A15	output	2.5 V		3
VCCIO3	A16	power		2.5V	3
MCB_DATA_pad[3]	A17	bidir	2.5 V		3
rd_addr_ram_pad[1]	A18	output	2.5 V		3
GND*	A19				3
wr_addr_ram_pad[10]	A20	output	2.5 V		3
rd_addr_ram_pad[14]	A21	output	2.5 V		3
rd_addr_ram_pad[13]	A22	output	2.5 V		3
VCCIO3	A23	power		2.5V	3
rd_addr_ram_pad[8]	A24	output	2.5 V		3
GND	A25	gnd			
DVALID_pad[5]	AA1	output	2.5 V		6
PHASE0_pad[1]	AA2	output	2.5 V		6
GND*	AA3				6
GND*	AA4				6
GND*	AA5				6
GND*	AA6				6
GND	AA7	gnd			
GND*	AA8				7
GND*	AA9				7
PHASE3_pad[2]	AA10	output	2.5 V		7
SDATA5_pad[1]	AA11	output	2.5 V		7
DUMP_EN_pad[1]	AA12	output	2.5 V		7
GND	AA13	gnd			
SDATA6_pad[2]	AA14	output	2.5 V		8
DUMP_SYNC_pad	AA15	output	2.5 V		8
PHASE3_pad[3]	AA16	output	2.5 V		8
PHASE6_pad[3]	AA17	output	2.5 V		8
GND*	AA18				8
GND*	AA19				8
nCONFIG	AA20				8
CTRL6_7_buf_pad(n)	AA21	output	HyperTransport		1
CTRL6_7_buf_pad	AA22	output	HyperTransport		1
GND*	AA23				1
GND*	AA24				1
BB_pad[3](n)	AA25	input	LVDS		1
BB_pad[3]	AA26	input	LVDS		1
PHASE2_pad[2]	AB1	output	2.5 V		6
DVALID_pad[1]	AB2	output	2.5 V		6
GND*	AB3				6
GND*	AB4				6
nCEO	AB5				7

PLL_ENA	AB6					7
GND*	AB7					7
GND*	AB8					7
AMUX_ADDR_pad[0]	AB9	output	2.5 V			7
PHASE6_pad[2]	AB10	output	2.5 V			7
PHASE4_pad[2]	AB11	output	2.5 V			7
SDATA3_pad[0]	AB12	output	2.5 V			7
DVALID_pad[0]	AB13	output	2.5 V			7
SDATA1_pad[0]	AB14	output	2.5 V			8
SE_CLK_pad[3]	AB15	output	2.5 V			8
PHASE7_pad[0]	AB16	output	2.5 V			8
rd_addr_ram_pad[0]	AB17	output	2.5 V			8
SDATA2_pad[2]	AB18	output	2.5 V			8
GND*	AB19					8
GND*	AB20					8
TRST	AB21	input				8
TCK	AB22	input				8
GND*	AB23					1
GND*	AB24					1
CTRL6_7_pad(n)	AB25	input	LVDS			1
CTRL6_7_pad	AB26	input	LVDS			1
VCCIO6	AC1	power			2.5V	6
GND*	AC2					6
GND*	AC3					6
GND*	AC4					7
VREFB7	AC5	power				7
INTERRUPT_TICK_pad	AC6	output	2.5 V			7
PHASE6_pad[1]	AC7	output	2.5 V			7
SDATA2_pad[3]	AC8	output	2.5 V			7
DUMP_EN_pad[5]	AC9	output	2.5 V			7
SDATA6_pad[1]	AC10	output	2.5 V			7
VREFB7	AC11	power				7
SDATA5_pad[3]	AC12	output	2.5 V			10
DUMP_EN_pad[2]	AC13	output	2.5 V			7
SE_CLK_pad[7]	AC14	output	2.5 V			8
GND*	AC15					8
VREFB8	AC16	power				8
DUMP_EN_pad[4]	AC17	output	2.5 V			8
DVALID_pad[7]	AC18	output	2.5 V			8
GND*	AC19					8
GND*	AC20					8
GND*	AC21					8
VREFB8	AC22	power				8
GND*	AC23					8
CTRL2_3_pad(n)	AC24	input	LVDS			1
CTRL2_3_pad	AC25	input	LVDS			1
VCCIO1	AC26	power			2.5V	1
GND*	AD1					6
GND*	AD2					6
AMUX_ADDR_pad[1]	AD3	output	2.5 V			7
AMUX_ADDR_pad[2]	AD4	output	2.5 V			7
PHASE7_pad[2]	AD5	output	2.5 V			7
SDATA4_pad[2]	AD6	output	2.5 V			7
PHASE6_pad[0]	AD7	output	2.5 V			7
SDATA2_pad[0]	AD8	output	2.5 V			7
PHASE2_pad[1]	AD9	output	2.5 V			7
SDATA2_pad[1]	AD10	output	2.5 V			7
SDATA7_pad[0]	AD11	output	2.5 V			7
SDATA0_pad[0]	AD12	output	2.5 V			10
SDATA0_pad[2]	AD13	output	2.5 V			10
SE_CLK_pad[0]	AD14	output	2.5 V			7
GND*	AD15					8
SDATA7_pad[3]	AD16	output	2.5 V			8
DVALID_pad[4]	AD17	output	2.5 V			8
DVALID_pad[3]	AD18	output	2.5 V			8
DUMP_EN_pad[3]	AD19	output	2.5 V			8
rd_addr_ram_pad[9]	AD20	output	2.5 V			8
rd_addr_ram_pad[11]	AD21	output	2.5 V			8
GND*	AD22					8
GND*	AD23					8
TMS	AD24	input				8

CTRL4_5_pad(n)	AD25	input	LVDS		1	
CTRL4_5_pad	AD26	input	LVDS		1	
GND	AE1	gnd				
nIO_PULLUP	AE2				7	
AMUX_ADDR_pad[3]	AE3	output	2.5 V		7	
PHASE0_pad[2]	AE4	output	2.5 V		7	
PHASE1_pad[3]	AE5	output	2.5 V		7	
SDATA7_pad[1]	AE6	output	2.5 V		7	
PHASE7_pad[3]	AE7	output	2.5 V		7	
SE_CLK_pad[4]	AE8	output	2.5 V		7	
SDATA6_pad[3]	AE9	output	2.5 V		7	
SDATA4_pad[3]	AE10	output	2.5 V		7	
PHASE1_pad[1]	AE11	output	2.5 V		7	
SDATA1_pad[1]	AE12	output	2.5 V		10	
DVALID_pad[2]	AE13	output	2.5 V		10	
DUMP_EN_pad[6]	AE14	output	2.5 V		7	
SE_CLK_pad[1]	AE15	output	2.5 V		8	
SDATA3_pad[3]	AE16	output	2.5 V		8	
SE_CLK_pad[6]	AE17	output	2.5 V		8	
TIMESTAMP_pad	AE18	output	2.5 V		8	
PHASE4_pad[1]	AE19	output	2.5 V		8	
rd_addr_ram_pad[6]	AE20	output	2.5 V		8	
rd_addr_ram_pad[7]	AE21	output	2.5 V		8	
GND*	AE22				8	
GND*	AE23				8	
GND*	AE24				8	
TDI	AE25	input			8	
GND	AE26	gnd				
GND	AF2	gnd				
AMUX_ADDR_pad[4]	AF3	output	2.5 V		7	
VCCIO7	AF4	power		2.5V	7	
SDATA3_pad[1]	AF5	output	2.5 V		7	
PHASE4_pad[3]	AF6	output	2.5 V		7	
PHASE2_pad[3]	AF7	output	2.5 V		7	
SDATA3_pad[2]	AF8	output	2.5 V		7	
DUMP_EN_pad[7]	AF9	output	2.5 V		7	
SDATA6_pad[0]	AF10	output	2.5 V		7	
VCCIO7	AF11	power		2.5V	7	
SE_CLK_pad[2]	AF12	output	2.5 V		10	
GND	AF13	gnd				
GND	AF14	gnd				
SE_CLK_pad[5]	AF15	output	2.5 V		8	
VCCIO8	AF16	power		2.5V	8	
SDATA0_pad[3]	AF17	output	2.5 V		8	
SCHID_FRAME_pad_	AF18	output	2.5 V		8	
CLOCKA2_pad	AF19	output	2.5 V		8	
DUMP_EN_pad[0]	AF20	output	2.5 V		8	
rd_addr_ram_pad[4]	AF21	output	2.5 V		8	
GND*	AF22				8	
VCCIO8	AF23	power		2.5V	8	
GND*	AF24				8	
GND	AF25	gnd				
GND	B1	gnd				
MSEL1	B2				4	
AMUX_WR_pad_	B3	output	2.5 V		4	
rd_data_ramB_pad[0]	B4	input	3.3-V LVTTL		4	
rd_data_ramA_pad[14]	B5	input	3.3-V LVTTL		4	
rd_data_ramB_pad[10]	B6	input	3.3-V LVTTL		4	
rd_data_ramA_pad[8]	B7	input	3.3-V LVTTL		4	
rd_data_ramA_pad[6]	B8	input	3.3-V LVTTL		4	
rd_data_ramB_pad[5]	B9	input	3.3-V LVTTL		4	
SDATA5_pad[0]	B10	output	2.5 V		4	
SDATA5_pad[2]	B11	output	2.5 V		4	
MCB_ADDR_pad[0]	B12	input	2.5 V		9	
SDATA4_pad[1]	B13	output	2.5 V		4	
MCB_ADDR_pad[2]	B14	input	2.5 V		4	
MCB_DATA_pad[7]	B15	bidir	2.5 V		3	
dpsram_clockB_pad	B16	output	2.5 V		3	
rd_data_ramA_pad[1]	B17	input	3.3-V LVTTL		3	
GND*	B18				3	
wr_addr_ram_pad[5]	B19	output	2.5 V		3	

rd_addr_ram_pad[10]	B20	output	2.5 V		3	
rd_addr_ram_pad[2]	B21	output	2.5 V		3	
CLOCKA6_pad	B22	output	2.5 V		3	
GND*	B23				3	
rd_addr_ram_pad[16]	B24	output	2.5 V		3	
CONF_DONE	B25				3	
GND	B26	gnd				
pinC1_phase0_0_pad	C1	output	2.5 V		5	
GND*	C2				5	
AMUX_EN0_pad_	C3	output	2.5 V		4	
rd_data_ramB_pad[16]	C4	input	3.3-V LVTTL		4	
rd_data_ramA_pad[3]	C5	input	3.3-V LVTTL		4	
rd_data_ramA_pad[16]	C6	input	3.3-V LVTTL		4	
rd_data_ramA_pad[10]	C7	input	3.3-V LVTTL		4	
PHASE0_pad[0]	C8	output	2.5 V		4	
rd_data_ramA_pad[5]	C9	input	3.3-V LVTTL		4	
rd_data_ramA_pad[0]	C10	input	3.3-V LVTTL		4	
rd_data_ramA_pad[17]	C11	input	3.3-V LVTTL		4	
CLOCKA5_pad	C12	output	2.5 V		9	
MCB_RD_WR_pad_	C13	input	2.5 V		4	
MCB_ADDR_pad[5]	C14	input	2.5 V		4	
SDATA0_pad[1]	C15	output	2.5 V		3	
MCB_DATA_pad[1]	C16	bidir	2.5 V		3	
rd_data_ramB_pad[13]	C17	input	3.3-V LVTTL		3	
wr_addr_ram_pad[12]	C18	output	2.5 V		3	
GND*	C19				3	
rd_addr_ram_pad[5]	C20	output	2.5 V		3	
rd_addr_ram_pad[3]	C21	output	2.5 V		3	
GND*	C22				3	
GND*	C23				3	
DCLK	C24				3	
GND*	C25				2	
GND*	C26				2	
VCCIO5	D1	power		2.5V	5	
GND*	D2				5	
GND*	D3				5	
GND*	D4				4	
VREFB4	D5	power			4	
rd_data_ramA_pad[9]	D6	input	3.3-V LVTTL		4	
rd_data_ramA_pad[15]	D7	input	3.3-V LVTTL		4	
rd_data_ramB_pad[4]	D8	input	3.3-V LVTTL		4	
rd_data_ramA_pad[4]	D9	input	3.3-V LVTTL		4	
rd_data_ramA_pad[12]	D10	input	3.3-V LVTTL		4	
VREFB4	D11	power			4	
rd_data_ramB_pad[12]	D12	input	3.3-V LVTTL		9	
MCB_ADDR_pad[3]	D13	input	2.5 V		9	
GND*	D14				3	
MCB_DATA_pad[4]	D15	bidir	2.5 V		3	
VREFB3	D16	power			3	
rd_data_ramA_pad[2]	D17	input	3.3-V LVTTL		3	
rd_addr_ram_pad[12]	D18	output	2.5 V		3	
AMUX_EN1_pad_	D19	output	2.5 V		3	
GND*	D20				3	
GND*	D21				3	
VREFB3	D22	power			3	
GND*	D23				3	
GND*	D24				2	
GND*	D25				2	
VCCIO2	D26	power		2.5V	2	
GND*	E1				5	
GND*	E2				5	
CLOCKA7_pad	E3	output	2.5 V		5	
GND*	E4				5	
TEMPDIODEp	E5					
MSEL2	E6				4	
rd_data_ramB_pad[2]	E7	input	3.3-V LVTTL		4	
CLOCKA4_pad	E8	output	2.5 V		4	
rd_data_ramA_pad[11]	E9	input	3.3-V LVTTL		4	
rd_data_ramB_pad[14]	E10	input	3.3-V LVTTL		4	
rd_data_ramB_pad[7]	E11	input	3.3-V LVTTL		4	
rd_data_ramA_pad[13]	E12	input	3.3-V LVTTL		4	

MCB_ADDR_pad[1]	E13	input	2.5 V		9	
MCB_DATA_pad[2]	E14	bidir	2.5 V		3	
MCB_DATA_pad[0]	E15	bidir	2.5 V		3	
~DATA0~ / RESERVED_INPUT	E16	input	2.5 V		3	
wr_addr_ram_pad[16]	E17	output	2.5 V		3	
GND*	E18				3	
GND*	E19				3	
GND*	E20				3	
nSTATUS	E21				3	
nCE	E22				3	
GND*	E23				2	
GND*	E24				2	
BB_pad[1](n)	E25	input	LVDS		2	
BB_pad[1]	E26	input	LVDS		2	
GND*	F1				5	
GND*	F2				5	
GND*	F3				5	
GND*	F4				5	
TEMPDIODEn	F5					
TDO	F6	output			4	
MSEL3	F7				4	
GND*	F8				4	
GND*	F9				4	
rd_data_ramB_pad[9]	F10	input	3.3-V LVTTL		4	
rd_data_ramB_pad[17]	F11	input	3.3-V LVTTL		4	
GND*	F12				4	
GND*	F13				4	
MCB_DATA_pad[5]	F14	bidir	2.5 V		3	
CLOCKA1_pad	F15	output	2.5 V		3	
MCB_DATA_pad[6]	F16	bidir	2.5 V		3	
wr_addr_ram_pad[6]	F17	output	2.5 V		3	
GND*	F18				3	
GND*	F19				3	
dpsram_clockA_pad	F20	output	2.5 V		3	
BB_buf_pad[0](n)	F21	output	HyperTransport		2	
BB_buf_pad[0]	F22	output	HyperTransport		2	
GND*	F23				2	
GND*	F24				2	
BB_pad[5](n)	F25	input	LVDS		2	
BB_pad[5]	F26	input	LVDS		2	
wr_data_ramA_pad[8]	G1	output	2.5 V		5	
wr_data_ramB_pad[5]	G2	output	2.5 V		5	
GND*	G3				5	
GND*	G4				5	
VREFB5	G5	power			5	
GND*	G6				5	
GND*	G7				5	
MSEL0	G8				4	
GND*	G9				4	
GND*	G10				4	
GND*	G11				4	
VCCA_PLL5	G12	power		1.2V		
GND	G13	gnd				
rd_data_ramB_pad[1]	G14	input	3.3-V LVTTL		3	
MCB_CS_pad_	G15	input	2.5 V		3	
PHASE5_pad[0]	G16	output	2.5 V		3	
rd_addr_ram_pad[17]	G17	output	2.5 V		3	
rd_addr_ram_pad[15]	G18	output	2.5 V		3	
GND*	G19				3	
GND*	G20				2	
GND*	G21				2	
VREFB2	G22	power			2	
GND*	G23				2	
GND*	G24				2	
BB_pad[4](n)	G25	input	LVDS		2	
BB_pad[4]	G26	input	LVDS		2	
wr_data_ramA_pad[4]	H1	output	2.5 V		5	
wr_addr_ram_pad[0]	H2	output	2.5 V		5	
GND*	H3				5	
wr_data_ramB_pad[17]	H4	output	2.5 V		5	
GND*	H5				5	

GND*	H6				5	
GND*	H7				5	
GND*	H8				5	
GND*	H9				4	
GND*	H10				4	
GND*	H11				4	
VCC_PLL5_OUT	H12	power		2.5V	9	
GNDA_PLL5	H13	gnd				
VCCD_PLL5	H14	power		1.2V		
GND*	H15				3	
GND*	H16				3	
GND*	H17				3	
GND*	H18				3	
GND*	H19				2	
GND*	H20				2	
BB_buf_pad[1](n)	H21	output	HyperTransport		2	
BB_buf_pad[1]	H22	output	HyperTransport		2	
GND*	H23				2	
GND*	H24				2	
BB_pad[7](n)	H25	input	LVDS		2	
BB_pad[7]	H26	input	LVDS		2	
wr_addr_ram_pad[17]	J1	output	2.5 V		5	
wr_addr_ram_pad[13]	J2	output	2.5 V		5	
GND*	J3				5	
wr_data_ramB_pad[6]	J4	output	2.5 V		5	
GND*	J5				5	
GND*	J6				5	
GND*	J7				5	
GND*	J8				5	
GND*	J9				4	
GND*	J10				4	
GND*	J11				4	
VCCPD4	J12	power		3.3V	4	
GNDA_PLL5	J13	gnd				
GND*	J14				3	
GND*	J15				3	
VCCPD3	J16	power		3.3V	3	
GND*	J17				3	
GND*	J18				3	
GND*	J19				2	
GND*	J20				2	
BB_buf_pad[2](n)	J21	output	HyperTransport		2	
BB_buf_pad[2]	J22	output	HyperTransport		2	
GND*	J23				2	
GND*	J24				2	
BB_pad[2](n)	J25	input	LVDS		2	
BB_pad[2]	J26	input	LVDS		2	
wr_addr_ram_pad[4]	K1	output	2.5 V		5	
wr_addr_ram_pad[15]	K2	output	2.5 V		5	
wr_data_ramA_pad[9]	K3	output	2.5 V		5	
wr_addr_ram_pad[7]	K4	output	2.5 V		5	
VREFB5	K5	power			5	
GND*	K6				5	
GND*	K7				5	
GND*	K8				5	
GND*	K9				5	
GND*	K10				4	
GND*	K11				4	
GND	K12	gnd				
VCCIO4	K13	power		2.5V	4	
GND	K14	gnd				
VCCIO3	K15	power		2.5V	3	
GND*	K16				3	
GND*	K17				3	
GND*	K18				3	
GND*	K19				2	
GND*	K20				2	
BB_buf_pad[3](n)	K21	output	HyperTransport		2	
BB_buf_pad[3]	K22	output	HyperTransport		2	
GND*	K23				2	
GND*	K24				2	

BB_pad[0](n)	K25	input	LVDS		2	
BB_pad[0]	K26	input	LVDS		2	
VCCIO5	L1	power		2.5V	5	
wr_data_ramB_pad[16]	L2	output	2.5 V		5	
wr_data_ramA_pad[0]	L3	output	2.5 V		5	
wr_data_ramB_pad[12]	L4	output	2.5 V		5	
GND*	L5				5	
GND*	L6				5	
wr_data_ramA_pad[5]	L7	output	2.5 V		5	
GND*	L8				5	
wr_addr_ram_pad[2]	L9	output	2.5 V		5	
VCCINT	L10	power		1.2V		
GND	L11	gnd				
VCCINT	L12	power		1.2V		
GND	L13	gnd				
VCCINT	L14	power		1.2V		
GND	L15	gnd				
VCCINT	L16	power		1.2V		
GND	L17	gnd				
GND*	L18				2	
GND*	L19				2	
GND*	L20				2	
GND*	L21				2	
BB_buf_pad[4](n)	L22	output	HyperTransport		2	
BB_buf_pad[4]	L23	output	HyperTransport		2	
GND*	L24				2	
GND*	L25				2	
VCCIO2	L26	power		2.5V	2	
wr_data_ramB_pad[0]	M1	output	2.5 V		5	
wr_data_ramB_pad[3]	M2	output	2.5 V		5	
GND*	M3				5	
GND*	M4				5	
wr_data_ramA_pad[16]	M5	output	2.5 V		5	
wr_addr_ram_pad[14]	M6	output	2.5 V		5	
wr_data_ramA_pad[17]	M7	output	2.5 V		5	
wr_data_ramA_pad[15]	M8	output	2.5 V		5	
VCCPD5	M9	power		3.3V	5	
GND	M10	gnd				
VCCINT	M11	power		1.2V		
GND	M12	gnd				
VCCINT	M13	power		1.2V		
GND	M14	gnd				
VCCINT	M15	power		1.2V		
GND	M16	gnd				
VCCIO2	M17	power		2.5V	2	
VCCPD2	M18	power		3.3V	2	
GND*	M19				2	
GND*	M20				2	
GND*	M21				2	
GND*	M22				2	
BB_buf_pad[5](n)	M23	output	HyperTransport		2	
BB_buf_pad[5]	M24	output	HyperTransport		2	
GND*	M25				2	
GND*	M26				2	
GND	N1	gnd				
RESET_pad_	N2	input	2.5 V		5	
GND+	N3				5	
wr_addr_ram_pad[11]	N4	output	2.5 V		5	
wr_data_ramB_pad[1]	N5	output	2.5 V		5	
wr_data_ramB_pad[4]	N6	output	2.5 V		5	
wr_data_ramA_pad[1]	N7	output	2.5 V		5	
GNDA_PLL4	N8	gnd				
GNDA_PLL4	N9	gnd				
VCCIO5	N10	power		2.5V	5	
GND	N11	gnd				
VCCINT	N12	power		1.2V		
GND	N13	gnd				
VCCINT	N14	power		1.2V		
GND	N15	gnd				
VCCINT	N16	power		1.2V		
GND	N17	gnd				

GNDA_PLL1	N18	gnd				
GND*	N19				2	
GND*	N20				2	
GND*	N21				2	
GND*	N22				2	
VREFB2	N23	power			2	
GND+	N24				2	
GND+	N25				2	
GND	N26	gnd				
GND	P1	gnd				
wr_addr_ram_pad[1]	P2	output	2.5 V		5	
wr_data_ramA_pad[14]	P3	output	2.5 V		5	
wr_data_ramA_pad[12]	P4	output	2.5 V		5	
wr_data_ramA_pad[10]	P5	output	2.5 V		5	
GND	P6	gnd				
VCCD_PLL3	P7	power		1.2V		
VCCA_PLL4	P8	power		1.2V		
VCCD_PLL4	P9	power		1.2V		
GND	P10	gnd				
VCCINT	P11	power		1.2V		
GND	P12	gnd				
VCCINT	P13	power		1.2V		
GND	P14	gnd				
VCCINT	P15	power		1.2V		
GND	P16	gnd				
VCCIO1	P17	power		2.5V	1	
GNDA_PLL1	P18	gnd				
VCCD_PLL1	P19	power		1.2V		
VCCD_PLL2	P20	power		1.2V		
VCCA_PLL1	P21	power		1.2V		
GND+	P22				1	
GND+	P23				1	
GND*	P24				2	
GND*	P25				2	
GND	P26	gnd				
wr_data_ramA_pad[13]	R1	output	2.5 V		6	
wr_data_ramB_pad[9]	R2	output	2.5 V		6	
MCB_CLK_pad	R3	input	2.5 V		6	
GND+	R4				6	
VREFB6	R5	power			6	
VCCA_PLL3	R6	power		1.2V		
GNDA_PLL3	R7	gnd				
GNDA_PLL3	R8	gnd				
VCCPD6	R9	power		3.3V	6	
VCCIO6	R10	power		2.5V	6	
GND	R11	gnd				
VCCINT	R12	power		1.2V		
GND	R13	gnd				
VCCINT	R14	power		1.2V		
GND	R15	gnd				
VCCINT	R16	power		1.2V		
GND	R17	gnd				
VCCPD1	R18	power		3.3V	1	
GNDA_PLL2	R19	gnd				
GNDA_PLL2	R20	gnd				
VCCA_PLL2	R21	power		1.2V		
GND	R22	gnd				
Clk_buf_pad(n)	R23	output	LVDS		1	
Clk_buf_pad	R24	output	LVDS		1	
REF_CLOCK_pad(n)	R25	input	LVDS		1	
REF_CLOCK_pad	R26	input	LVDS		1	
VCCIO6	T1	power		2.5V	6	
wr_addr_ram_pad[9]	T2	output	2.5 V		6	
wr_data_ramA_pad[6]	T3	output	2.5 V		6	
wr_addr_ram_pad[3]	T4	output	2.5 V		6	
wr_data_ramA_pad[11]	T5	output	2.5 V		6	
SDATA1_pad[3]	T6	output	2.5 V		6	
PHASE0_pad[3]	T7	output	2.5 V		6	
wr_data_ramA_pad[3]	T8	output	2.5 V		6	
wr_data_ramB_pad[7]	T9	output	2.5 V		6	
GND	T10	gnd				

VCCINT	T11	power		1.2V		
GND	T12	gnd				
VCCINT	T13	power		1.2V		
GND	T14	gnd				
VCCINT	T15	power		1.2V		
GND	T16	gnd				
VCCINT	T17	power		1.2V		
GND	T18	gnd				
GND*	T19				1	
GND*	T20				1	
BB_buf_pad[6](n)	T21	output	HyperTransport		1	
BB_buf_pad[6]	T22	output	HyperTransport		1	
VREFB1	T23	power			1	
GND*	T24				1	
GND*	T25				1	
VCCIO1	T26	power		2.5V	1	
wr_data_ramB_pad[13]	U1	output	2.5 V		6	
PHASE1_pad[0]	U2	output	2.5 V		6	
wr_data_ramA_pad[7]	U3	output	2.5 V		6	
CLOCKA_pad	U4	output	2.5 V		6	
PHASE3_pad[1]	U5	output	2.5 V		6	
SDATA7_pad[2]	U6	output	2.5 V		6	
wr_data_ramA_pad[2]	U7	output	2.5 V		6	
PHASE1_pad[2]	U8	output	2.5 V		6	
GND	U9	gnd				
VCCINT	U10	power		1.2V		
GND	U11	gnd				
VCCIO7	U12	power		2.5V	7	
GND	U13	gnd				
VCCIO8	U14	power		2.5V	8	
GND	U15	gnd				
VCCINT	U16	power		1.2V		
GND	U17	gnd				
VCCINT	U18	power		1.2V		
GND*	U19				1	
GND*	U20				1	
GND*	U21				1	
GND*	U22				1	
BB_buf_pad[7](n)	U23	output	HyperTransport		1	
BB_buf_pad[7]	U24	output	HyperTransport		1	
GND*	U25				1	
GND*	U26				1	
wr_data_ramB_pad[8]	V1	output	2.5 V		6	
wr_data_ramB_pad[10]	V2	output	2.5 V		6	
SDATA1_pad[2]	V3	output	2.5 V		6	
wr_data_ramB_pad[2]	V4	output	2.5 V		6	
wr_data_ramB_pad[11]	V5	output	2.5 V		6	
PHASE7_pad[1]	V6	output	2.5 V		6	
GND*	V7				6	
GND*	V8				6	
VCCINT	V9	power		1.2V		
GND*	V10				7	
VCCPD7	V11	power		3.3V	7	
GND*	V12				7	
VCC_PLL6_OUT	V13	power		2.5V	10	
GND*	V14				8	
VCCPD8	V15	power		3.3V	8	
GND*	V16				8	
GND*	V17				8	
GND*	V18				8	
GND*	V19				1	
GND*	V20				1	
GND*	V21				1	
GND*	V22				1	
CTRL0_1_buf_pad(n)	V23	output	HyperTransport		1	
CTRL0_1_buf_pad	V24	output	HyperTransport		1	
GND*	V25				1	
GND*	V26				1	
wr_data_ramB_pad[15]	W1	output	2.5 V		6	
wr_data_ramB_pad[14]	W2	output	2.5 V		6	
PHASE3_pad[0]	W3	output	2.5 V		6	

```
PHASE5_pad[2]                : W4        : output : 2.5 V            :         : 6         :
GND*                         : W5        :        :                  :         : 6         :
GND*                         : W6        :        :                  :         : 6         :
GND*                         : W7        :        :                  :         : 6         :
GND*                         : W8        :        :                  :         : 6         :
GND*                         : W9        :        :                  :         : 7         :
GND*                         : W10       :        :                  :         : 7         :
GND*                         : W11       :        :                  :         : 7         :
GND*                         : W12       :        :                  :         : 7         :
GNDA_PLL6                    : W13       : gnd    :                  :         :           :
GNDA_PLL6                    : W14       : gnd    :                  :         :           :
GND*                         : W15       :        :                  :         : 8         :
GND*                         : W16       :        :                  :         : 8         :
GND*                         : W17       :        :                  :         : 8         :
GND*                         : W18       :        :                  :         : 8         :
GND*                         : W19       :        :                  :         : 1         :
GND*                         : W20       :        :                  :         : 1         :
CTRL2_3_buf_pad(n)           : W21       : output : HyperTransport   :         : 1         :
CTRL2_3_buf_pad              : W22       : output : HyperTransport   :         : 1         :
GND*                         : W23       :        :                  :         : 1         :
GND*                         : W24       :        :                  :         : 1         :
CTRL0_1_pad(n)               : W25       : input  : LVDS             :         : 1         :
CTRL0_1_pad                  : W26       : input  : LVDS             :         : 1         :
wr_addr_ram_pad[8]           : Y1        : output : 2.5 V            :         : 6         :
PHASE2_pad[0]                : Y2        : output : 2.5 V            :         : 6         :
GND*                         : Y3        :        :                  :         : 6         :
GND*                         : Y4        :        :                  :         : 6         :
VREFB6                       : Y5        : power  :                  :         : 6         :
GND*                         : Y6        :        :                  :         : 6         :
GND*                         : Y7        :        :                  :         : 6         :
PORSEL                       : Y8        :        :                  :         : 7         :
GND*                         : Y9        :        :                  :         : 7         :
PHASE5_pad[1]                : Y10       : output : 2.5 V            :         : 7         :
GND*                         : Y11       :        :                  :         : 7         :
GND*                         : Y12       :        :                  :         : 7         :
VCCA_PLL6                    : Y13       : power  :                  : 1.2V    :           :
VCCD_PLL6                    : Y14       : power  :                  : 1.2V    :           :
DVALID_pad[6]                : Y15       : output : 2.5 V            :         : 8         :
CLOCKA3_pad                  : Y16       : output : 2.5 V            :         : 8         :
PHASE5_pad[3]                : Y17       : output : 2.5 V            :         : 8         :
GND*                         : Y18       :        :                  :         : 8         :
VCCSEL                       : Y19       :        :                  :         : 8         :
CTRL4_5_buf_pad(n)           : Y20       : output : HyperTransport   :         : 1         :
CTRL4_5_buf_pad              : Y21       : output : HyperTransport   :         : 1         :
VREFB1                       : Y22       : power  :                  :         : 1         :
GND*                         : Y23       :        :                  :         : 1         :
GND*                         : Y24       :        :                  :         : 1         :
BB_pad[6](n)                 : Y25       : input  : LVDS             :         : 1         :
BB_pad[6]                    : Y26       : input  : LVDS             :         : 1         :
-------------------------------------------------------------------------------------------------------------------
-- NC            : No Connect. This pin has no internal connection to the device.
 -- DNU          : Do Not Use. This pin MUST NOT be connected.
 -- VCCINT       : Dedicated power pin, which MUST be connected to VCC  (1.2V).
 -- VCCIO        : Dedicated power pin, which MUST be connected to VCC
 --                of its bank.
 --                                        Bank 1:            2.5V
 --                                        Bank 2:            2.5V
 --                                        Bank 3:            2.5V
 --                                        Bank 4:            2.5V
 --                                        Bank 5:            2.5V
 --                                        Bank 6:            2.5V
 --                                        Bank 7:            2.5V
 --                                        Bank 8:            2.5V
 --                                        Bank 9:            2.5V
 --                                        Bank 10:  2.5V
 -- GND          : Dedicated ground pin. Dedicated GND pins MUST be connected to GND.
 -- GND+         : Unused input pin. It can also be used to report unused dual-purpose pins.
 --                                        This pin should be connected to GND.
 -- GND*         : Unused  I/O  pin
 -- RESERVED     : Unused I/O pin, which MUST be left unconnected.
 -- RESERVED_INPUT    : Pin is tri-stated and should be connected to the board.
 -- RESERVED_INPUT_WITH_WEAK_PULLUP    : Pin is tri-stated with internal weak pull-up resistor.
 -- RESERVED_INPUT_WITH_BUS_HOLD       : Pin is tri-stated with bus-hold circuitry.
 -- RESERVED_OUTPUT_DRIVEN_HIGH        : Pin is output driven high.
 ---------------------------------------------------------------------------------------------------
```

9.2.1 "Ytx" FPGA Pin Notes

- All 2.5 V LVTTL outputs are set for 4 mA drive strength. This is done to minimize output switching noise to minimize coupling of noise into the on-chip PLL, to maximize BB_buf_pad and CTRL*_buf_pad signal integrity.

- All BB_buf*_pad and CTRL*_buf_pad outputs are set for HyperTransport levels (~1200 mV pk-pk differential), for superior operation going across the Patch Board to the next Baseline Board RXP inputs.

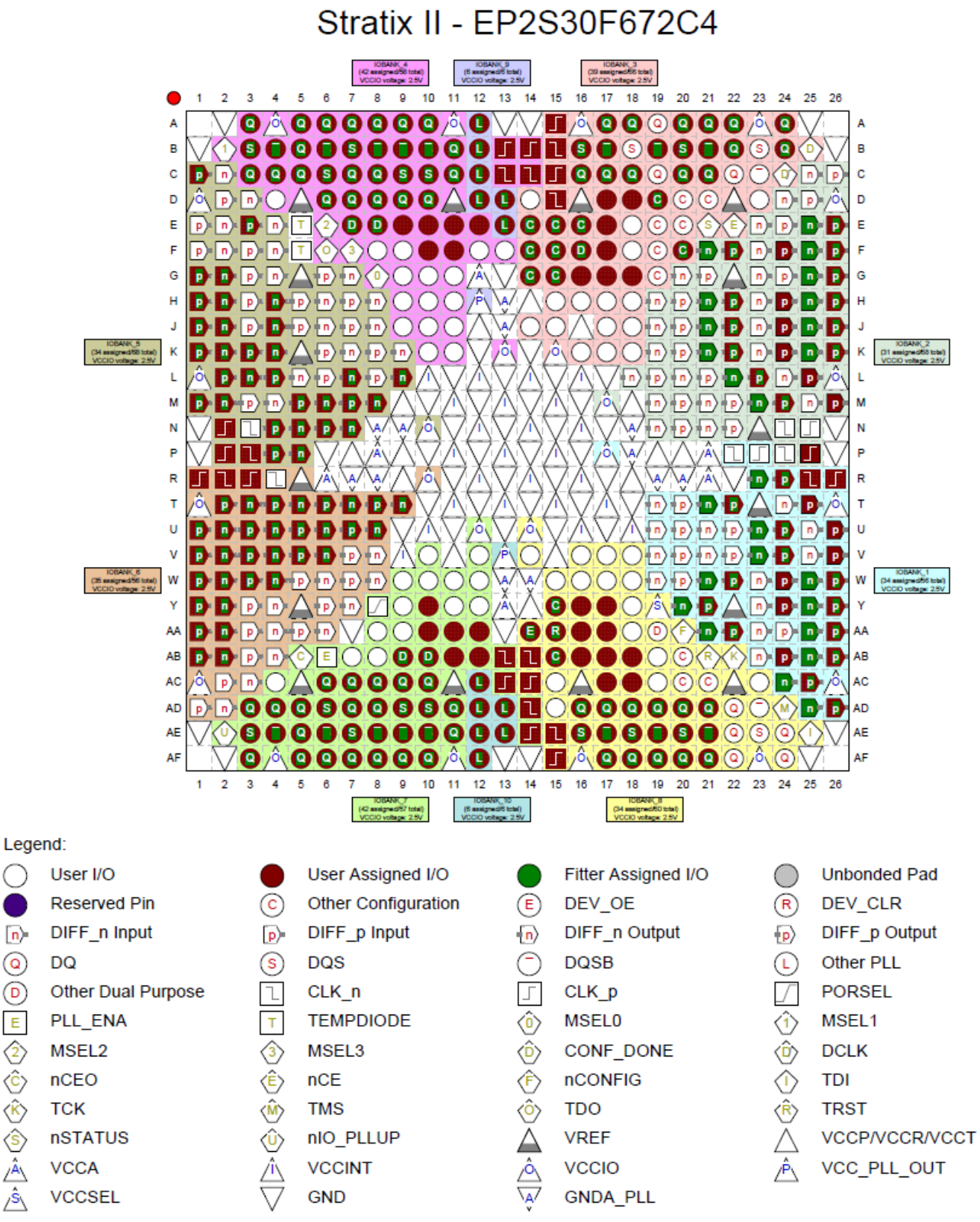


Figure 9-2 Quartus-II V9.1 pin planner output of the "Ytx" Recirc FPGA.

9.3 Recirc "Yrx" FPGA Pinouts

The Recirc "Yrx" FPGA is an alternative design, not normally required in the EVLA system, wherein the Y Recirc FPGAs get their HM Gbps input data (BB_pad[7:0], CTRL*_pad) from the Y-ERNI connector, rather than the RXP FPGA (REF_CLOCK still is obtained from the RXP FPGAs). This design would only be used if the EVLA correlator were expanded beyond 32 antennas to allow a board to correlate any 32x32 portion of an arbitrarily large correlation matrix. The design requires population of a number of 0201 resistors at each Y Recirc FPGA site on the board to enable the capability. Internally, the FPGA design is identical to the X Recirc FGPA; the only difference is different assignments of CTRL* and BB_* pins, with the receivers set for no on-chip differential termination to enable "split-T" termination (described later in this section).

Refer to section 4 for pin functionality descriptions.

Pin Name/Usage	Location	Dir.	I/O Standard	Voltage	I/O Bank
GND	A2	gnd			
rd_data_ramB_pad[3]	A3	input	3.3-V LVTTL		4
VCCIO4	A4	power		2.5V	4
rd_data_ramA_pad[7]	A5	input	3.3-V LVTTL		4
rd_data_ramB_pad[15]	A6	input	3.3-V LVTTL		4
rd_data_ramB_pad[8]	A7	input	3.3-V LVTTL		4
rd_data_ramB_pad[6]	A8	input	3.3-V LVTTL		4
rd_data_ramB_pad[11]	A9	input	3.3-V LVTTL		4
SDATA4_pad[0]	A10	output	2.5 V		4
VCCIO4	A11	power		2.5V	4
MCB_ADDR_pad[4]	A12	input	2.5 V		9
GND	A13	gnd			
GND	A14	gnd			
PHASE4_pad[0]	A15	output	2.5 V		3
VCCIO3	A16	power		2.5V	3
MCB_DATA_pad[3]	A17	bidir	2.5 V		3
rd_addr_ram_pad[1]	A18	output	2.5 V		3
GND*	A19				3
wr_addr_ram_pad[10]	A20	output	2.5 V		3
rd_addr_ram_pad[14]	A21	output	2.5 V		3
rd_addr_ram_pad[13]	A22	output	2.5 V		3
VCCIO3	A23	power		2.5V	3
rd_addr_ram_pad[8]	A24	output	2.5 V		3
GND	A25	gnd			
DVALID_pad[5]	AA1	output	2.5 V		6
PHASE0_pad[1]	AA2	output	2.5 V		6
GND*	AA3				6
GND*	AA4				6
GND*	AA5				6
GND*	AA6				6
GND	AA7	gnd			
GND*	AA8				7
GND*	AA9				7
PHASE3_pad[2]	AA10	output	2.5 V		7
SDATA5_pad[1]	AA11	output	2.5 V		7
DUMP_EN_pad[1]	AA12	output	2.5 V		7
GND	AA13	gnd			
SDATA6_pad[2]	AA14	output	2.5 V		8
DUMP_SYNC_pad	AA15	output	2.5 V		8
PHASE3_pad[3]	AA16	output	2.5 V		8
PHASE6_pad[3]	AA17	output	2.5 V		8
GND*	AA18				8
GND*	AA19				8
nCONFIG	AA20				8
RESERVE_HIZ_pad[0]	AA21	input	2.5 V		1

RESERVE_HIZ_pad[1]	AA22	input	2.5 V		1
GND*	AA23				1
GND*	AA24				1
UNUSED_LVDS[0](n)	AA25	input	LVDS		1
UNUSED_LVDS[0]	AA26	input	LVDS		1
PHASE2_pad[2]	AB1	output	2.5 V		6
DVALID_pad[1]	AB2	output	2.5 V		6
GND*	AB3				6
GND*	AB4				6
nCEO	AB5				7
PLL_ENA	AB6				7
GND*	AB7				7
GND*	AB8				7
AMUX_ADDR_pad[0]	AB9	output	2.5 V		7
PHASE6_pad[2]	AB10	output	2.5 V		7
PHASE4_pad[2]	AB11	output	2.5 V		7
SDATA3_pad[0]	AB12	output	2.5 V		7
DVALID_pad[0]	AB13	output	2.5 V		7
SDATA1_pad[0]	AB14	output	2.5 V		8
SE_CLK_pad[3]	AB15	output	2.5 V		8
PHASE7_pad[0]	AB16	output	2.5 V		8
rd_addr_ram_pad[0]	AB17	output	2.5 V		8
SDATA2_pad[2]	AB18	output	2.5 V		8
GND*	AB19				8
GND*	AB20				8
TRST	AB21	input			8
TCK	AB22	input			8
CTRL6_7_pad(n)	AB23	input	LVDS		1
CTRL6_7_pad	AB24	input	LVDS		1
UNUSED_LVDS[8](n)	AB25	input	LVDS		1
UNUSED_LVDS[8]	AB26	input	LVDS		1
VCCIO6	AC1	power		2.5V	6
GND*	AC2				6
GND*	AC3				6
GND*	AC4				7
VREFB7	AC5	power			7
INTERRUPT_TICK_pad	AC6	output	2.5 V		7
PHASE6_pad[1]	AC7	output	2.5 V		7
SDATA2_pad[3]	AC8	output	2.5 V		7
DUMP_EN_pad[5]	AC9	output	2.5 V		7
SDATA6_pad[1]	AC10	output	2.5 V		7
VREFB7	AC11	power			7
SDATA5_pad[3]	AC12	output	2.5 V		10
DUMP_EN_pad[2]	AC13	output	2.5 V		7
SE_CLK_pad[7]	AC14	output	2.5 V		8
GND*	AC15				8
VREFB8	AC16	power			8
DUMP_EN_pad[4]	AC17	output	2.5 V		8
DVALID_pad[7]	AC18	output	2.5 V		8
GND*	AC19				8
GND*	AC20				8
GND*	AC21				8
VREFB8	AC22	power			8
GND*	AC23				8
UNUSED_LVDS[9](n)	AC24	input	LVDS		1
UNUSED_LVDS[9]	AC25	input	LVDS		1
VCCIO1	AC26	power		2.5V	1
GND*	AD1				6
GND*	AD2				6
AMUX_ADDR_pad[1]	AD3	output	2.5 V		7
AMUX_ADDR_pad[2]	AD4	output	2.5 V		7
PHASE7_pad[2]	AD5	output	2.5 V		7
SDATA4_pad[2]	AD6	output	2.5 V		7
PHASE6_pad[0]	AD7	output	2.5 V		7
SDATA2_pad[0]	AD8	output	2.5 V		7
PHASE2_pad[1]	AD9	output	2.5 V		7
SDATA2_pad[1]	AD10	output	2.5 V		7
SDATA7_pad[0]	AD11	output	2.5 V		7
SDATA0_pad[0]	AD12	output	2.5 V		10
SDATA0_pad[2]	AD13	output	2.5 V		10
SE_CLK_pad[0]	AD14	output	2.5 V		7

GND*	AD15				8	
SDATA7_pad[3]	AD16	output	2.5 V		8	
DVALID_pad[4]	AD17	output	2.5 V		8	
DVALID_pad[3]	AD18	output	2.5 V		8	
DUMP_EN_pad[3]	AD19	output	2.5 V		8	
rd_addr_ram_pad[9]	AD20	output	2.5 V		8	
rd_addr_ram_pad[11]	AD21	output	2.5 V		8	
GND*	AD22				8	
GND*	AD23				8	
TMS	AD24	input			8	
UNUSED_LVDS[10](n)	AD25	input	LVDS		1	
UNUSED_LVDS[10]	AD26	input	LVDS		1	
GND	AE1	gnd				
nIO_PULLUP	AE2				7	
AMUX_ADDR_pad[3]	AE3	output	2.5 V		7	
PHASE0_pad[2]	AE4	output	2.5 V		7	
PHASE1_pad[3]	AE5	output	2.5 V		7	
SDATA7_pad[1]	AE6	output	2.5 V		7	
PHASE7_pad[3]	AE7	output	2.5 V		7	
SE_CLK_pad[4]	AE8	output	2.5 V		7	
SDATA6_pad[3]	AE9	output	2.5 V		7	
SDATA4_pad[3]	AE10	output	2.5 V		7	
PHASE1_pad[1]	AE11	output	2.5 V		7	
SDATA1_pad[1]	AE12	output	2.5 V		10	
DVALID_pad[2]	AE13	output	2.5 V		10	
DUMP_EN_pad[6]	AE14	output	2.5 V		7	
SE_CLK_pad[1]	AE15	output	2.5 V		8	
SDATA3_pad[3]	AE16	output	2.5 V		8	
SE_CLK_pad[6]	AE17	output	2.5 V		8	
TIMESTAMP_pad	AE18	output	2.5 V		8	
PHASE4_pad[1]	AE19	output	2.5 V		8	
rd_addr_ram_pad[6]	AE20	output	2.5 V		8	
rd_addr_ram_pad[7]	AE21	output	2.5 V		8	
GND*	AE22				8	
GND*	AE23				8	
GND*	AE24				8	
TDI	AE25	input			8	
GND	AE26	gnd				
GND	AF2	gnd				
AMUX_ADDR_pad[4]	AF3	output	2.5 V		7	
VCCIO7	AF4	power		2.5V	7	
SDATA3_pad[1]	AF5	output	2.5 V		7	
PHASE4_pad[3]	AF6	output	2.5 V		7	
PHASE2_pad[3]	AF7	output	2.5 V		7	
SDATA3_pad[2]	AF8	output	2.5 V		7	
DUMP_EN_pad[7]	AF9	output	2.5 V		7	
SDATA6_pad[0]	AF10	output	2.5 V		7	
VCCIO7	AF11	power		2.5V	7	
SE_CLK_pad[2]	AF12	output	2.5 V		10	
GND	AF13	gnd				
GND	AF14	gnd				
SE_CLK_pad[5]	AF15	output	2.5 V		8	
VCCIO8	AF16	power		2.5V	8	
SDATA0_pad[3]	AF17	output	2.5 V		8	
SCHID_FRAME_pad_	AF18	output	2.5 V		8	
CLOCKA2_pad	AF19	output	2.5 V		8	
DUMP_EN_pad[0]	AF20	output	2.5 V		8	
rd_addr_ram_pad[4]	AF21	output	2.5 V		8	
GND*	AF22				8	
VCCIO8	AF23	power		2.5V	8	
GND*	AF24				8	
GND	AF25	gnd				
GND	B1	gnd				
MSEL1	B2				4	
AMUX_WR_pad_	B3	output	2.5 V		4	
rd_data_ramB_pad[0]	B4	input	3.3-V LVTTL		4	
rd_data_ramA_pad[14]	B5	input	3.3-V LVTTL		4	
rd_data_ramB_pad[10]	B6	input	3.3-V LVTTL		4	
rd_data_ramA_pad[8]	B7	input	3.3-V LVTTL		4	
rd_data_ramA_pad[6]	B8	input	3.3-V LVTTL		4	
rd_data_ramB_pad[5]	B9	input	3.3-V LVTTL		4	

SDATA5_pad[0]	B10	output	2.5 V		4	
SDATA5_pad[2]	B11	output	2.5 V		4	
MCB_ADDR_pad[0]	B12	input	2.5 V		9	
SDATA4_pad[1]	B13	output	2.5 V		4	
MCB_ADDR_pad[2]	B14	input	2.5 V		4	
MCB_DATA_pad[7]	B15	bidir	2.5 V		3	
dpsram_clockB_pad	B16	output	2.5 V		3	
rd_data_ramA_pad[1]	B17	input	3.3-V LVTTL		3	
GND*	B18				3	
wr_addr_ram_pad[5]	B19	output	2.5 V		3	
rd_addr_ram_pad[10]	B20	output	2.5 V		3	
rd_addr_ram_pad[2]	B21	output	2.5 V		3	
CLOCKA6_pad	B22	output	2.5 V		3	
GND*	B23				3	
rd_addr_ram_pad[16]	B24	output	2.5 V		3	
CONF_DONE	B25				3	
GND	B26	gnd				
GND*	C1				5	
GND*	C2				5	
AMUX_EN0_pad_	C3	output	2.5 V		4	
rd_data_ramB_pad[16]	C4	input	3.3-V LVTTL		4	
rd_data_ramA_pad[3]	C5	input	3.3-V LVTTL		4	
rd_data_ramA_pad[16]	C6	input	3.3-V LVTTL		4	
rd_data_ramA_pad[10]	C7	input	3.3-V LVTTL		4	
PHASE0_pad[0]	C8	output	2.5 V		4	
rd_data_ramA_pad[5]	C9	input	3.3-V LVTTL		4	
rd_data_ramA_pad[0]	C10	input	3.3-V LVTTL		4	
rd_data_ramA_pad[17]	C11	input	3.3-V LVTTL		4	
CLOCKA5_pad	C12	output	2.5 V		9	
MCB_RD_WR_pad_	C13	input	2.5 V		4	
MCB_ADDR_pad[5]	C14	input	2.5 V		4	
SDATA0_pad[1]	C15	output	2.5 V		3	
MCB_DATA_pad[1]	C16	bidir	2.5 V		3	
rd_data_ramB_pad[13]	C17	input	3.3-V LVTTL		3	
wr_addr_ram_pad[12]	C18	output	2.5 V		3	
GND*	C19				3	
rd_addr_ram_pad[5]	C20	output	2.5 V		3	
rd_addr_ram_pad[3]	C21	output	2.5 V		3	
GND*	C22				3	
GND*	C23				3	
DCLK	C24				3	
GND*	C25				2	
GND*	C26				2	
VCCIO5	D1	power		2.5V	5	
GND*	D2				5	
GND*	D3				5	
GND*	D4				4	
VREFB4	D5	power			4	
rd_data_ramA_pad[9]	D6	input	3.3-V LVTTL		4	
rd_data_ramA_pad[15]	D7	input	3.3-V LVTTL		4	
rd_data_ramB_pad[4]	D8	input	3.3-V LVTTL		4	
rd_data_ramA_pad[4]	D9	input	3.3-V LVTTL		4	
rd_data_ramA_pad[12]	D10	input	3.3-V LVTTL		4	
VREFB4	D11	power			4	
rd_data_ramB_pad[12]	D12	input	3.3-V LVTTL		9	
MCB_ADDR_pad[3]	D13	input	2.5 V		9	
GND*	D14				3	
MCB_DATA_pad[4]	D15	bidir	2.5 V		3	
VREFB3	D16	power			3	
rd_data_ramA_pad[2]	D17	input	3.3-V LVTTL		3	
rd_addr_ram_pad[12]	D18	output	2.5 V		3	
AMUX_EN1_pad_	D19	output	2.5 V		3	
GND*	D20				3	
GND*	D21				3	
VREFB3	D22	power			3	
GND*	D23				3	
GND*	D24				2	
GND*	D25				2	
VCCIO2	D26	power		2.5V	2	
GND*	E1				5	
GND*	E2				5	

CLOCKA7_pad	E3	output	2.5 V		5	
GND*	E4				5	
TEMPDIODEp	E5					
MSEL2	E6				4	
rd_data_ramB_pad[2]	E7	input	3.3-V LVTTL		4	
CLOCKA4_pad	E8	output	2.5 V		4	
rd_data_ramA_pad[11]	E9	input	3.3-V LVTTL		4	
rd_data_ramB_pad[14]	E10	input	3.3-V LVTTL		4	
rd_data_ramB_pad[7]	E11	input	3.3-V LVTTL		4	
rd_data_ramA_pad[13]	E12	input	3.3-V LVTTL		4	
MCB_ADDR_pad[1]	E13	input	2.5 V		9	
MCB_DATA_pad[2]	E14	bidir	2.5 V		3	
MCB_DATA_pad[0]	E15	bidir	2.5 V		3	
~DATA0~ / RESERVED_INPUT	E16	input	2.5 V		3	
wr_addr_ram_pad[16]	E17	output	2.5 V		3	
GND*	E18				3	
GND*	E19				3	
GND*	E20				3	
nSTATUS	E21				3	
nCE	E22				3	
GND*	E23				2	
GND*	E24				2	
UNUSED_LVDS[1](n)	E25	input	LVDS		2	
UNUSED_LVDS[1]	E26	input	LVDS		2	
GND*	F1				5	
GND*	F2				5	
GND*	F3				5	
GND*	F4				5	
TEMPDIODEn	F5					
TDO	F6	output			4	
MSEL3	F7				4	
GND*	F8				4	
GND*	F9				4	
rd_data_ramB_pad[9]	F10	input	3.3-V LVTTL		4	
rd_data_ramB_pad[17]	F11	input	3.3-V LVTTL		4	
GND*	F12				4	
GND*	F13				4	
MCB_DATA_pad[5]	F14	bidir	2.5 V		3	
CLOCKA1_pad	F15	output	2.5 V		3	
MCB_DATA_pad[6]	F16	bidir	2.5 V		3	
wr_addr_ram_pad[6]	F17	output	2.5 V		3	
GND*	F18				3	
GND*	F19				3	
dpsram_clockA_pad	F20	output	2.5 V		3	
RESERVE_HIZ_pad[2]	F21	input	2.5 V		2	
RESERVE_HIZ_pad[3]	F22	input	2.5 V		2	
BB_pad[0](n)	F23	input	LVDS		2	
BB_pad[0]	F24	input	LVDS		2	
UNUSED_LVDS[2](n)	F25	input	LVDS		2	
UNUSED_LVDS[2]	F26	input	LVDS		2	
wr_data_ramA_pad[8]	G1	output	2.5 V		5	
wr_data_ramB_pad[5]	G2	output	2.5 V		5	
GND*	G3				5	
GND*	G4				5	
VREFB5	G5	power			5	
GND*	G6				5	
GND*	G7				5	
MSEL0	G8				4	
GND*	G9				4	
GND*	G10				4	
GND*	G11				4	
VCCA_PLL5	G12	power		1.2V		
GND	G13	gnd				
rd_data_ramB_pad[1]	G14	input	3.3-V LVTTL		3	
MCB_CS_pad_	G15	input	2.5 V		3	
PHASE5_pad[0]	G16	output	2.5 V		3	
rd_addr_ram_pad[17]	G17	output	2.5 V		3	
rd_addr_ram_pad[15]	G18	output	2.5 V		3	
GND*	G19				3	
GND*	G20				2	
GND*	G21				2	

VREFB2	G22	power			2	
GND*	G23				2	
GND*	G24				2	
UNUSED_LVDS[3](n)	G25	input	LVDS		2	
UNUSED_LVDS[3]	G26	input	LVDS		2	
wr_data_ramA_pad[4]	H1	output	2.5 V		5	
wr_addr_ram_pad[0]	H2	output	2.5 V		5	
GND*	H3				5	
wr_data_ramB_pad[17]	H4	output	2.5 V		5	
GND*	H5				5	
GND*	H6				5	
GND*	H7				5	
GND*	H8				5	
GND*	H9				4	
GND*	H10				4	
GND*	H11				4	
VCC_PLL5_OUT	H12	power		2.5V	9	
GNDA_PLL5	H13	gnd				
VCCD_PLL5	H14	power		1.2V		
GND*	H15				3	
GND*	H16				3	
GND*	H17				3	
GND*	H18				3	
GND*	H19				2	
GND*	H20				2	
RESERVE_HIZ_pad[4]	H21	input	2.5 V		2	
RESERVE_HIZ_pad[5]	H22	input	2.5 V		2	
BB_pad[1](n)	H23	input	LVDS		2	
BB_pad[1]	H24	input	LVDS		2	
UNUSED_LVDS[4](n)	H25	input	LVDS		2	
UNUSED_LVDS[4]	H26	input	LVDS		2	
wr_addr_ram_pad[17]	J1	output	2.5 V		5	
wr_addr_ram_pad[13]	J2	output	2.5 V		5	
GND*	J3				5	
wr_data_ramB_pad[6]	J4	output	2.5 V		5	
GND*	J5				5	
GND*	J6				5	
GND*	J7				5	
GND*	J8				5	
GND*	J9				4	
GND*	J10				4	
GND*	J11				4	
VCCPD4	J12	power		3.3V	4	
GNDA_PLL5	J13	gnd				
GND*	J14				3	
GND*	J15				3	
VCCPD3	J16	power		3.3V	3	
GND*	J17				3	
GND*	J18				3	
GND*	J19				2	
GND*	J20				2	
RESERVE_HIZ_pad[6]	J21	input	2.5 V		2	
RESERVE_HIZ_pad[7]	J22	input	2.5 V		2	
BB_pad[2](n)	J23	input	LVDS		2	
BB_pad[2]	J24	input	LVDS		2	
UNUSED_LVDS[5](n)	J25	input	LVDS		2	
UNUSED_LVDS[5]	J26	input	LVDS		2	
wr_addr_ram_pad[4]	K1	output	2.5 V		5	
wr_addr_ram_pad[15]	K2	output	2.5 V		5	
wr_data_ramA_pad[9]	K3	output	2.5 V		5	
wr_addr_ram_pad[7]	K4	output	2.5 V		5	
VREFB5	K5	power			5	
GND*	K6				5	
GND*	K7				5	
GND*	K8				5	
GND*	K9				5	
GND*	K10				4	
GND*	K11				4	
GND	K12	gnd				
VCCIO4	K13	power		2.5V	4	
GND	K14	gnd				

VCCIO3	K15	power		2.5V	3	
GND*	K16				3	
GND*	K17				3	
GND*	K18				3	
GND*	K19				2	
GND*	K20				2	
RESERVE_HIZ_pad[8]	K21	input	2.5 V		2	
RESERVE_HIZ_pad[9]	K22	input	2.5 V		2	
BB_pad[3](n)	K23	input	LVDS		2	
BB_pad[3]	K24	input	LVDS		2	
UNUSED_LVDS[6](n)	K25	input	LVDS		2	
UNUSED_LVDS[6]	K26	input	LVDS		2	
VCCIO5	L1	power		2.5V	5	
wr_data_ramB_pad[16]	L2	output	2.5 V		5	
wr_data_ramA_pad[0]	L3	output	2.5 V		5	
wr_data_ramB_pad[12]	L4	output	2.5 V		5	
GND*	L5				5	
GND*	L6				5	
wr_data_ramA_pad[5]	L7	output	2.5 V		5	
GND*	L8				5	
wr_addr_ram_pad[2]	L9	output	2.5 V		5	
VCCINT	L10	power		1.2V		
GND	L11	gnd				
VCCINT	L12	power		1.2V		
GND	L13	gnd				
VCCINT	L14	power		1.2V		
GND	L15	gnd				
VCCINT	L16	power		1.2V		
GND	L17	gnd				
GND*	L18				2	
GND*	L19				2	
GND*	L20				2	
GND*	L21				2	
RESERVE_HIZ_pad[10]	L22	input	2.5 V		2	
RESERVE_HIZ_pad[11]	L23	input	2.5 V		2	
BB_pad[4](n)	L24	input	LVDS		2	
BB_pad[4]	L25	input	LVDS		2	
VCCIO2	L26	power		2.5V	2	
wr_data_ramB_pad[0]	M1	output	2.5 V		5	
wr_data_ramB_pad[3]	M2	output	2.5 V		5	
GND*	M3				5	
GND*	M4				5	
wr_data_ramA_pad[16]	M5	output	2.5 V		5	
wr_addr_ram_pad[14]	M6	output	2.5 V		5	
wr_data_ramA_pad[17]	M7	output	2.5 V		5	
wr_data_ramA_pad[15]	M8	output	2.5 V		5	
VCCPD5	M9	power		3.3V	5	
GND	M10	gnd				
VCCINT	M11	power		1.2V		
GND	M12	gnd				
VCCINT	M13	power		1.2V		
GND	M14	gnd				
VCCINT	M15	power		1.2V		
GND	M16	gnd				
VCCIO2	M17	power		2.5V	2	
VCCPD2	M18	power		3.3V	2	
GND*	M19				2	
GND*	M20				2	
GND*	M21				2	
GND*	M22				2	
RESERVE_HIZ_pad[12]	M23	input	2.5 V		2	
RESERVE_HIZ_pad[13]	M24	input	2.5 V		2	
BB_pad[5](n)	M25	input	LVDS		2	
BB_pad[5]	M26	input	LVDS		2	
GND	N1	gnd				
RESET_pad_	N2	input	2.5 V		5	
GND+	N3				5	
wr_addr_ram_pad[11]	N4	output	2.5 V		5	
wr_data_ramB_pad[1]	N5	output	2.5 V		5	
wr_data_ramB_pad[4]	N6	output	2.5 V		5	
wr_data_ramA_pad[1]	N7	output	2.5 V		5	

GNDA_PLL4	N8	gnd				
GNDA_PLL4	N9	gnd				
VCCIO5	N10	power		2.5V	5	
GND	N11	gnd				
VCCINT	N12	power		1.2V		
GND	N13	gnd				
VCCINT	N14	power		1.2V		
GND	N15	gnd				
VCCINT	N16	power		1.2V		
GND	N17	gnd				
GNDA_PLL1	N18	gnd				
GND*	N19				2	
GND*	N20				2	
GND*	N21				2	
GND*	N22				2	
VREFB2	N23	power			2	
GND+	N24				2	
GND+	N25				2	
GND	N26	gnd				
GND	P1	gnd				
wr_addr_ram_pad[1]	P2	output	2.5 V		5	
wr_data_ramA_pad[14]	P3	output	2.5 V		5	
wr_data_ramA_pad[12]	P4	output	2.5 V		5	
wr_data_ramA_pad[10]	P5	output	2.5 V		5	
GND	P6	gnd				
VCCD_PLL3	P7	power		1.2V		
VCCA_PLL4	P8	power		1.2V		
VCCD_PLL4	P9	power		1.2V		
GND	P10	gnd				
VCCINT	P11	power		1.2V		
GND	P12	gnd				
VCCINT	P13	power		1.2V		
GND	P14	gnd				
VCCINT	P15	power		1.2V		
GND	P16	gnd				
VCCIO1	P17	power		2.5V	1	
GNDA_PLL1	P18	gnd				
VCCD_PLL1	P19	power		1.2V		
VCCD_PLL2	P20	power		1.2V		
VCCA_PLL1	P21	power		1.2V		
GND+	P22				1	
GND+	P23				1	
GND*	P24				2	
GND*	P25				2	
GND	P26	gnd				
wr_data_ramA_pad[13]	R1	output	2.5 V		6	
wr_data_ramB_pad[9]	R2	output	2.5 V		6	
MCB_CLK_pad	R3	input	2.5 V		6	
GND+	R4				6	
VREFB6	R5	power			6	
VCCA_PLL3	R6	power		1.2V		
GNDA_PLL3	R7	gnd				
GNDA_PLL3	R8	gnd				
VCCPD6	R9	power		3.3V	6	
VCCIO6	R10	power		2.5V	6	
GND	R11	gnd				
VCCINT	R12	power		1.2V		
GND	R13	gnd				
VCCINT	R14	power		1.2V		
GND	R15	gnd				
VCCINT	R16	power		1.2V		
GND	R17	gnd				
VCCPD1	R18	power		3.3V	1	
GNDA_PLL2	R19	gnd				
GNDA_PLL2	R20	gnd				
VCCA_PLL2	R21	power		1.2V		
GND	R22	gnd				
UNUSED_CLOCK_TERM[1]	R23	output	2.5 V		1	
UNUSED_CLOCK_TERM[0]	R24	output	2.5 V		1	
REF_CLOCK_pad(n)	R25	input	LVDS		1	
REF_CLOCK_pad	R26	input	LVDS		1	

VCCIO6	T1	power		2.5V	6	
wr_addr_ram_pad[9]	T2	output	2.5 V		6	
wr_data_ramA_pad[6]	T3	output	2.5 V		6	
wr_addr_ram_pad[3]	T4	output	2.5 V		6	
wr_data_ramA_pad[11]	T5	output	2.5 V		6	
SDATA1_pad[3]	T6	output	2.5 V		6	
PHASE0_pad[3]	T7	output	2.5 V		6	
wr_data_ramA_pad[3]	T8	output	2.5 V		6	
wr_data_ramB_pad[7]	T9	output	2.5 V		6	
GND	T10	gnd				
VCCINT	T11	power		1.2V		
GND	T12	gnd				
VCCINT	T13	power		1.2V		
GND	T14	gnd				
VCCINT	T15	power		1.2V		
GND	T16	gnd				
VCCINT	T17	power		1.2V		
GND	T18	gnd				
GND*	T19				1	
GND*	T20				1	
RESERVE_HIZ_pad[14]	T21	input	2.5 V		1	
RESERVE_HIZ_pad[15]	T22	input	2.5 V		1	
VREFB1	T23	power			1	
BB_pad[6](n)	T24	input	LVDS		1	
BB_pad[6]	T25	input	LVDS		1	
VCCIO1	T26	power		2.5V	1	
wr_data_ramB_pad[13]	U1	output	2.5 V		6	
PHASE1_pad[0]	U2	output	2.5 V		6	
wr_data_ramA_pad[7]	U3	output	2.5 V		6	
CLOCKA_pad	U4	output	2.5 V		6	
PHASE3_pad[1]	U5	output	2.5 V		6	
SDATA7_pad[2]	U6	output	2.5 V		6	
wr_data_ramA_pad[2]	U7	output	2.5 V		6	
PHASE1_pad[2]	U8	output	2.5 V		6	
GND	U9	gnd				
VCCINT	U10	power		1.2V		
GND	U11	gnd				
VCCIO7	U12	power		2.5V	7	
GND	U13	gnd				
VCCIO8	U14	power		2.5V	8	
GND	U15	gnd				
VCCINT	U16	power		1.2V		
GND	U17	gnd				
VCCINT	U18	power		1.2V		
GND*	U19				1	
GND*	U20				1	
GND*	U21				1	
GND*	U22				1	
RESERVE_HIZ_pad[16]	U23	input	2.5 V		1	
RESERVE_HIZ_pad[17]	U24	input	2.5 V		1	
BB_pad[7](n)	U25	input	LVDS		1	
BB_pad[7]	U26	input	LVDS		1	
wr_data_ramB_pad[8]	V1	output	2.5 V		6	
wr_data_ramB_pad[10]	V2	output	2.5 V		6	
SDATA1_pad[2]	V3	output	2.5 V		6	
wr_data_ramB_pad[2]	V4	output	2.5 V		6	
wr_data_ramB_pad[11]	V5	output	2.5 V		6	
PHASE7_pad[1]	V6	output	2.5 V		6	
GND*	V7				6	
GND*	V8				6	
VCCINT	V9	power		1.2V		
GND*	V10				7	
VCCPD7	V11	power		3.3V	7	
GND*	V12				7	
VCC_PLL6_OUT	V13	power		2.5V	10	
GND*	V14				8	
VCCPD8	V15	power		3.3V	8	
GND*	V16				8	
GND*	V17				8	
GND*	V18				8	
GND*	V19				1	

```
GND*                        : V20       :        :                    :         : 1         :
GND*                        : V21       :        :                    :         : 1         :
GND*                        : V22       :        :                    :         : 1         :
RESERVE_HIZ_pad[18]         : V23       : input  : 2.5 V              :         : 1         :
RESERVE_HIZ_pad[19]         : V24       : input  : 2.5 V              :         : 1         :
CTRL0_1_pad(n)              : V25       : input  : LVDS               :         : 1         :
CTRL0_1_pad                 : V26       : input  : LVDS               :         : 1         :
wr_data_ramB_pad[15]        : W1        : output : 2.5 V              :         : 6         :
wr_data_ramB_pad[14]        : W2        : output : 2.5 V              :         : 6         :
PHASE3_pad[0]               : W3        : output : 2.5 V              :         : 6         :
PHASE5_pad[2]               : W4        : output : 2.5 V              :         : 6         :
GND*                        : W5        :        :                    :         : 6         :
GND*                        : W6        :        :                    :         : 6         :
GND*                        : W7        :        :                    :         : 6         :
GND*                        : W8        :        :                    :         : 6         :
GND*                        : W9        :        :                    :         : 7         :
GND*                        : W10       :        :                    :         : 7         :
GND*                        : W11       :        :                    :         : 7         :
GND*                        : W12       :        :                    :         : 7         :
GNDA_PLL6                   : W13       : gnd    :                    :         :           :
GNDA_PLL6                   : W14       : gnd    :                    :         :           :
GND*                        : W15       :        :                    :         : 8         :
GND*                        : W16       :        :                    :         : 8         :
GND*                        : W17       :        :                    :         : 8         :
GND*                        : W18       :        :                    :         : 8         :
GND*                        : W19       :        :                    :         : 1         :
GND*                        : W20       :        :                    :         : 1         :
RESERVE_HIZ_pad[20]         : W21       : input  : 2.5 V              :         : 1         :
RESERVE_HIZ_pad[21]         : W22       : input  : 2.5 V              :         : 1         :
CTRL2_3_pad(n)              : W23       : input  : LVDS               :         : 1         :
CTRL2_3_pad                 : W24       : input  : LVDS               :         : 1         :
UNUSED_LVDS[11](n)          : W25       : input  : LVDS               :         : 1         :
UNUSED_LVDS[11]             : W26       : input  : LVDS               :         : 1         :
wr_addr_ram_pad[8]          : Y1        : output : 2.5 V              :         : 6         :
PHASE2_pad[0]               : Y2        : output : 2.5 V              :         : 6         :
GND*                        : Y3        :        :                    :         : 6         :
GND*                        : Y4        :        :                    :         : 6         :
VREFB6                      : Y5        : power  :                    :         : 6         :
GND*                        : Y6        :        :                    :         : 6         :
GND*                        : Y7        :        :                    :         : 6         :
PORSEL                      : Y8        :        :                    :         : 7         :
GND*                        : Y9        :        :                    :         : 7         :
PHASE5_pad[1]               : Y10       : output : 2.5 V              :         : 7         :
GND*                        : Y11       :        :                    :         : 7         :
GND*                        : Y12       :        :                    :         : 7         :
VCCA_PLL6                   : Y13       : power  :                    : 1.2V    :           :
VCCD_PLL6                   : Y14       : power  :                    : 1.2V    :           :
DVALID_pad[6]               : Y15       : output : 2.5 V              :         : 8         :
CLOCKA3_pad                 : Y16       : output : 2.5 V              :         : 8         :
PHASE5_pad[3]               : Y17       : output : 2.5 V              :         : 8         :
GND*                        : Y18       :        :                    :         : 8         :
VCCSEL                      : Y19       :        :                    :         : 8         :
RESERVE_HIZ_pad[22]         : Y20       : input  : 2.5 V              :         : 1         :
RESERVE_HIZ_pad[23]         : Y21       : input  : 2.5 V              :         : 1         :
VREFB1                      : Y22       : power  :                    :         : 1         :
CTRL4_5_pad(n)              : Y23       : input  : LVDS               :         : 1         :
CTRL4_5_pad                 : Y24       : input  : LVDS               :         : 1         :
UNUSED_LVDS[7](n)           : Y25       : input  : LVDS               :         : 1         :
UNUSED_LVDS[7]              : Y26       : input  : LVDS               :         : 1         :
-------------------------------------------------------------------------------------------------------------------
-- NC           : No Connect. This pin has no internal connection to the device.
 -- DNU          : Do Not Use. This pin MUST NOT be connected.
 -- VCCINT       : Dedicated power pin, which MUST be connected to VCC  (1.2V).
 -- VCCIO        : Dedicated power pin, which MUST be connected to VCC
 --                of its bank.
 --                                        Bank 1:            2.5V
 --                                        Bank 2:            2.5V
 --                                        Bank 3:            2.5V
 --                                        Bank 4:            2.5V
 --                                        Bank 5:            2.5V
 --                                        Bank 6:            2.5V
 --                                        Bank 7:            2.5V
 --                                        Bank 8:            2.5V
 --                                        Bank 9:            2.5V
 --                                        Bank 10:  2.5V
```

```
-- GND           : Dedicated ground pin. Dedicated GND pins MUST be connected to GND.
-- GND+          : Unused input pin. It can also be used to report unused dual-purpose pins.
--                                    This pin should be connected to GND.
-- GND*          : Unused  I/O  pin
-- RESERVED      : Unused I/O pin, which MUST be left unconnected.
-- RESERVED_INPUT     : Pin is tri-stated and should be connected to the board.
-- RESERVED_INPUT_WITH_WEAK_PULLUP    : Pin is tri-stated with internal weak pull-up resistor.
-- RESERVED_INPUT_WITH_BUS_HOLD       : Pin is tri-stated with bus-hold circuitry.
-- RESERVED_OUTPUT_DRIVEN_HIGH        : Pin is output driven high.
----------------------------------------------------------------------------------------------------
```

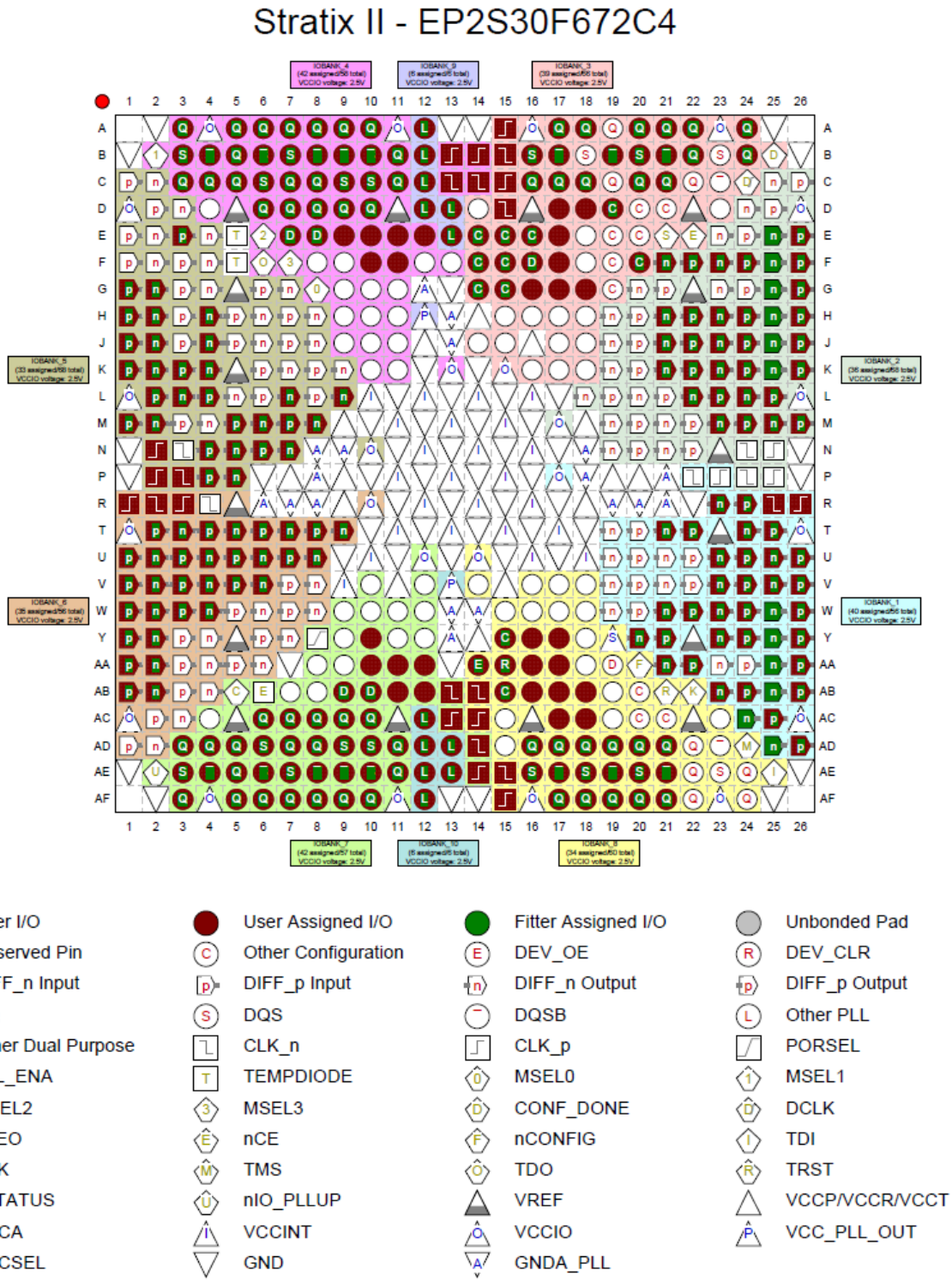


Figure 9-3 Quartus-II V9.1 pin planner output of the "Yrx" Recirc FPGA.

9.3.1 "Yrx" Recirc FPGA "split-T" Termination

To facilitate one board design, whilst allowing for the possibility of using the Y-ERNI connector for inputs rather than outputs, the board is designed for split-T termination, enabled with the population of a number of 0201 pkg resistors[16] at each Y Recirc FPGA site (bottom-side of the board, opposite each FPGA).

split-T termination is shown in the following figure.

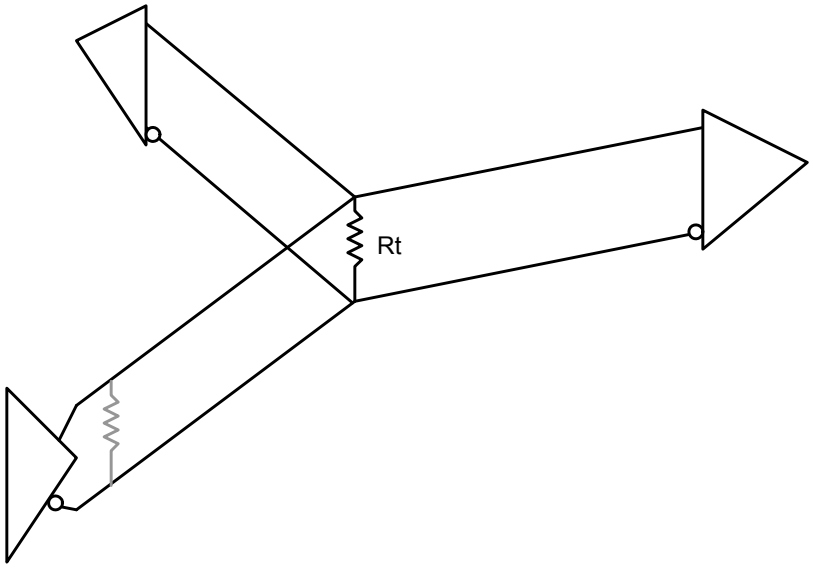
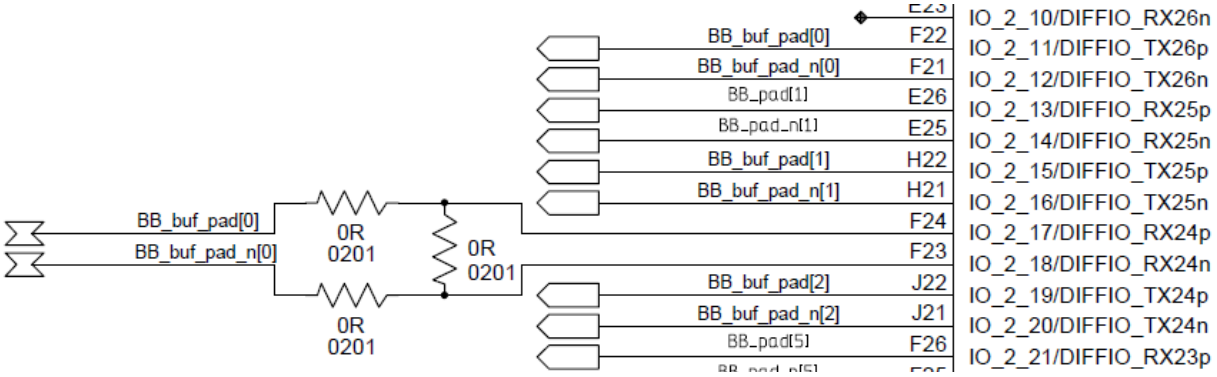


Figure 9-4 Split-T termination. Provided transmission lines after Rt to un-terminated receivers are the same length, each receiver sees a proper signal with correct amplitude. Ideally, a source parallel termination transmitter should be used to minimize propagating reflections caused when the incident wave hits Rt in parallel with 2 transmission lines.

The Baseline Board schematic contains a number of instances similar to the following:



[16] The design was specifically done to NOT require the installation of 0201 resistors in the normal EVLA (<=32 antenna) configuration, as 0201 resistor installation is somewhat problematic due to their very small size.

In this example, “BB_buf_pad[0]” pin pairs F24/F23 (Rx), are connected to pin pairs F22/F21 (Tx for “Ytx” design, but unconnected hi-impedance for the “Yrx” design) when the series 0201 0R resistors are installed. The parallel 0201 0R resistor—just a placeholder value—is also installed, but with a value of 100 ohms.

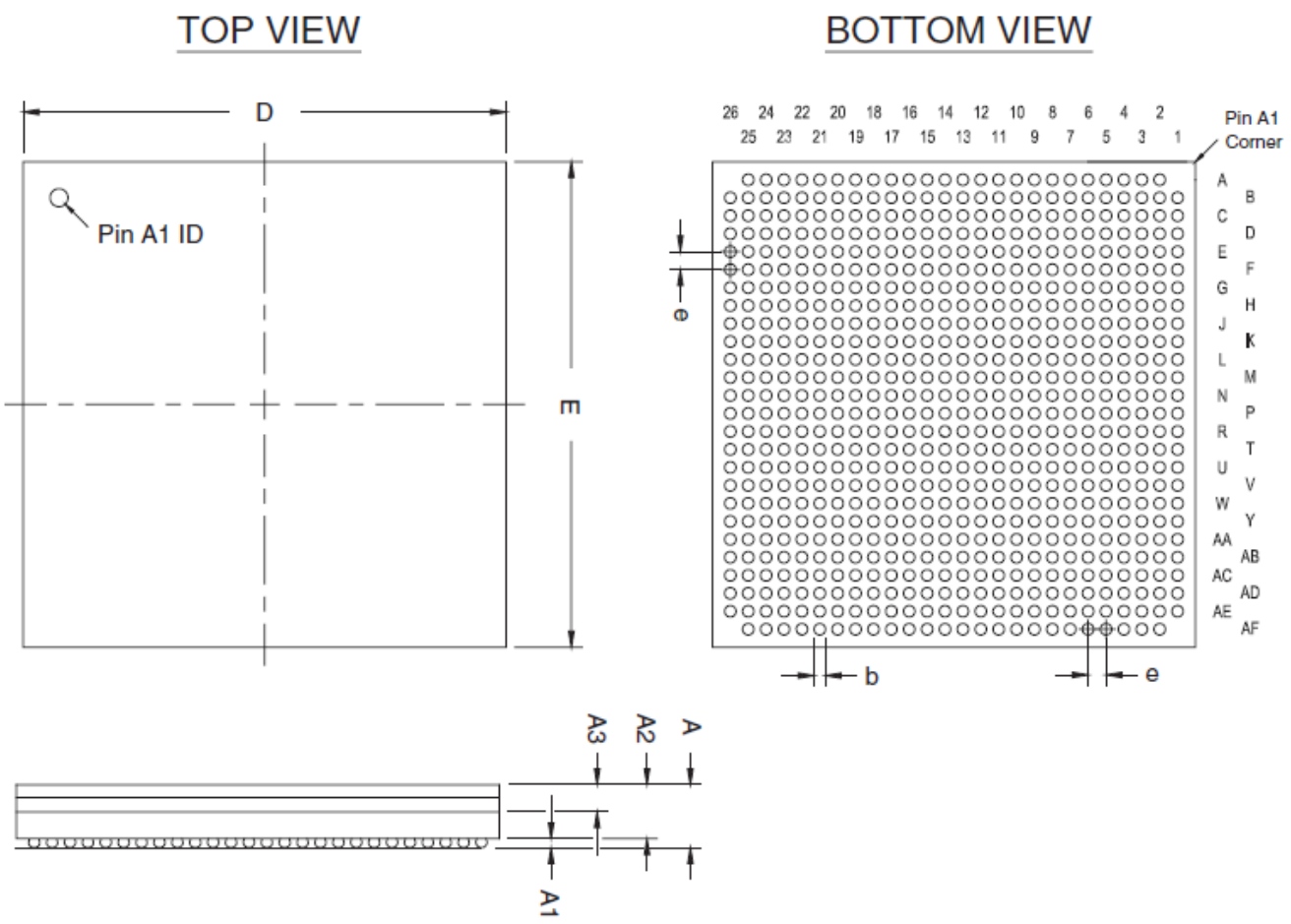
Note that the split-T concept is sound, although the specific implementation on the Baseline Board has not been directly tested due to the problem of mounting tiny 0201 resistors. However, 0201 pads on the board have been carefully designed so that the transmission lines to high impedance Rx and high impedance disable Tx are matched as well as possible, assuming that Tx and Rx transmission lines in the FPGA package are also reasonably matched.

The board is designed and populated to AC-couple and re-establish proper 2.5 V LVDS bias when the Yrx Recirc FPGA design is used—all that is required is a different FPGA binary, and correct population of 0201 resistors as noted above.

An alternative to the split-T termination, if it doesn’t work, is to NOT install the parallel resistor in the above figure, and set the Rx input for on-chip differential termination, resulting in “only” a reflection off the un-terminated Tx.

For design continuity, the X Recirc FPGA also has these 0201 resistor pads, but they are never used.

9.4 Altera EP2S30F672C4 FBGA Package Drawing



Symbol	Millimeters		
	Min.	Nom.	Max.
A	–	–	3.50
A1	0.30	–	–
A2	0.25	–	3.00
A3	–	–	2.50
D	27.00 BSC		
E	27.00 BSC		
b	0.50	0.60	0.70
e	1.00 BSC		

Figure 9-5 Recirc FPGA F672 package outline and dimensions. Drawings and dimensions taken from the Altera Stratix-II package information data sheet.

9.5 Altera FPGA Programming Notes

The Baseline Board on which this Recirc FPGAs reside is set up for 1-bit Passive Serial programming via the PCMC FPGA. The Recirc FPGAs (all 16 of them, 8 X and 8 Y) are in series to allow each FPGA to have a different personality, and so there is one file that contains the program for all FPGAs. The following figure is a screen shot of the "Convert Programming Files" dialog in the Altera Quartus-II software, showing the setup required to produce the .rbf (raw binary file), needed by the software to program the FPGA.

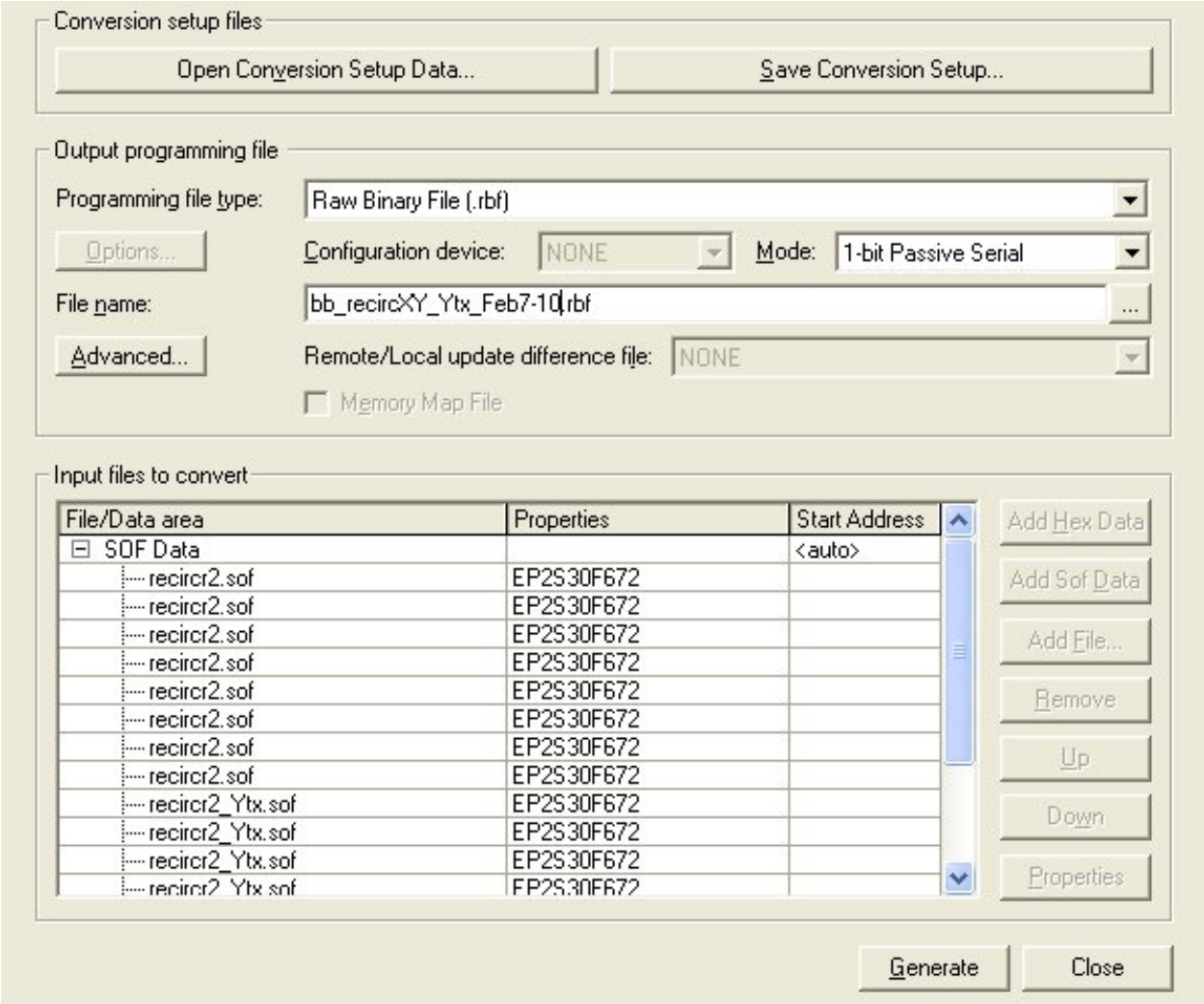


Figure 9-6 Altera Quartus-II "Convert Programming Files" screenshot showing settings necessary for generating the .rbf file for the RXP FPGAs.

Note that "recircr2.sof" is the .sof file produced for the compilation and place and route of the "X" Recirc FPGA, and "recircr2_Ytx.sof" is for the "Ytx" Recirc FPGA. A similar setup would be used for the "Yrx" ("recircr2_Yrx.sof") FPGA.

10 Index

H

I

J

L

M

O

P

R

S

T

W

X

Y