

### Introduction

Lattice has extended its family of high-speed serial backplane devices with the ORSO82G5 device. Built on the Series 4 reconfigurable embedded System-on-a-Chip (SoC) architecture, the ORSO82G5 is a high-speed transceiver with an aggregate bandwidth of over 20 Gbits/s that is targeted towards users needing high-speed backplane interfaces for SONET and other non-SONET applications. The ORSO82G5 has 8 channels of integrated 1.0-2.7G SERDES channels with built-in Clock and Data Recovery (CDR), along with more than 400K usable FPGA system gates. The CDR circuitry is a macrocell available from Lattice's smart silicon macro library and has already been implemented in numerous applications, including ASICs, standard products, and FPSCs, to create interfaces for STS-48/STM-16 and STS-192/STM-64 SONET/SDH applications. With the addition of protocol and access logic such as protocol-independent framers, Asynchronous Transfer Mode (ATM) framers, Packet-over-SONET (PoS) interfaces, and framers for HDLC for Internet Protocol (IP), designers can build a configurable interface retaining proven backplane driver/receiver technology. Designers can also use the device to drive high-speed data transfer across buses within a system that are not SONET/SDH based. The ORSO82G5 can also be used to provide a full 10 Gbits/s backplane data connection with protection between a line card and switch fabric.

The ORSO82G5 supports a clockless high-speed interface for interdevice communication on a board or across a backplane. The built-in clock recovery of the ORSO82G5 allows for higher system performance, easier-to-design clock domains in a multiboard system and fewer signals on the backplane. Network designers will benefit from using the backplane transceiver as a network termination device. A sister device, called the ORT82G5, supports 8B/10B encoding/decoding and link state machines for 10 Gbit Ethernet (XAUI) and Fiber Channel. The ORSO82G5 performs SONET data scrambling/descrambling, streamlined SONET framing, limited Transport Over-Head (TOH) handling, plus the programmable logic to terminate the network into proprietary systems. The cell processing feature in the ORSO82G5 makes it ideal for interfacing devices with any proprietary data format across a high-speed backplane. For non-SONET applications, all SONET functionality is hidden from the user and no prior networking knowledge is required. The ORSO82G5 is completely pin-compatible with the ORT82G5 device.

**Table 1. ORCA ORSO82G5 Family—Available FPGA Logic**

Device	PFU Rows	PFU Columns	Total PFUs	FPGA Max User I/O	LUTs	EBR Blocks	EBR Bits (K)	FPGA System Gates (K) <sup>1,2</sup>
ORSO82G5	36	36	1296	372	10,368	12	111	333 - 643

1. The embedded core, Embedded System Bus, FPGA interface and MPI are not included in the above gate counts. The System Gate ranges are derived from the following: Minimum System Gates assumes 100% of the PFU's are used for logic only (No PFU RAM) with 40% EBR usage and 2 PLL's. Maximum System Gates assumes 80% of the PFU's are for logic, 20% are used for PFU RAM, with 80% EBR usage and 6 PLL's."
2. There are two 4K X 36 (144K bits each) RAM blocks in the embedded core which are also accessible by the FPGA logic.

**Table of Contents**

Introduction .....	1	Recommended Operating Conditions .....	88
Table of Contents .....	2	SERDES Electrical and Timing Characteristics .....	88
Embedded Function Features .....	3	Pin Information .....	91
Programmable Features .....	4	Pin Descriptions .....	91
Programmable Logic System Features .....	6	Power Supplies for ORSO82G5.....	97
Description .....	7	Recommended Power Supply Connections..	98
What Is an FPSC? .....	7	Recommended Power Supply Filtering	
FPSC Overview .....	7	Scheme .....	98
FPSC Gate Counting .....	7	Package Pinouts .....	100
FPGA/Embedded Core Interface .....	7	Package Information .....	117
ispLEVER Development System .....	7	Package Thermal Characteristics Summary .....	117
FPSC Design Kit .....	7	$\theta$ JA .....	117
FPGA Logic Overview .....	8	$\psi$ JC .....	117
PLC Logic .....	8	$\theta$ JC .....	117
Programmable I/O .....	8	$\theta$ JB .....	118
Routing .....	9	FPSC Maximum Junction Temperature .....	118
System-Level Features .....	9	Package Coplanarity .....	118
MicroProcessor Interface .....	9	Heat Sink Vendors for BGA Packages .....	119
System Bus .....	10	Package Parasitics .....	120
Phase-Locked Loops .....	10	Package Outline Diagrams .....	121
Embedded Block RAM .....	10	680-Pin PBGAM .....	122
Configuration .....	10	Ordering Information .....	123
ORSO82G5 Overview .....	11		
Embedded Core Overview .....	12		
ORSO82G5 Main Operating Modes-			
Overview .....	13		
Embedded Core Functional Blocks -			
Overview .....	14		
Loopback - Overview .....	16		
FPSC Configuration - Overview .....	16		
ORSO82G5 Embedded Core Detailed Description ..	17		
Top Level Description - Transmitter (TX) and			
Receiver (RX) Architectures .....	17		
Detailed Description - SERDES Only Mode .	21		
SONET Mode Operation -			
Detailed Description .....	26		
SONET Mode Transmit Path .....	32		
SONET Mode Receive Path .....	35		
Cell Mode Detailed Description .....	46		
Cell Mode Transmit Path .....	49		
Cell Mode Receive Path .....	53		
Cell Extractor.....	54		
Receive FIFO .....	54		
Input Port Controllers .....	55		
IPC Receive Cell Mode Timing Core/FPGA .	56		
Reset Conditions .....	64		
Embedded Core Block RAM .....	65		
Register Maps .....	68		
Types of Registers .....	68		
Absolute Maximum Ratings .....	88		

---

## Embedded Function Features

- High-speed SERDES programmable serial data rates of 1.0 Gbits/s to 2.7 Gbits/s.
- Asynchronous operation per receive channel (separate PLL per channel).
- Transmit pre-emphasis (programmable) for improved receive data eye opening.
- Provides a 10 Gbits/s backplane interface to switch fabric using four work and four protect 2.5 Gbit/s links. Also supports port cards at rates between 1.0 Gbits/s and 2.7 Gbits/s.
- Allows wide range of applications for SONET network termination, as well as generic data moving for high-speed backplane data transfer.
- No knowledge of SONET/SDH needed in generic applications. Simply supply data (125 MHz—168.75 MHz clock) and optionally a frame pulse.
- High-Speed Interface (HSI) function for clock/data recovery serial backplane data transfer without external clocks.
- Eight-channel HSI function provides 2.7 Gbits/s serial user data interface per channel for a total chip bandwidth of >20 Gbits/s (full duplex).
- SERDES has low-power CML buffers and support for 1.5 V/1.8 V I/Os.
- SERDES HSI automatically recovers from loss-of-clock once its reference clock returns to normal operating state.
- Powerdown option of SERDES HSI receiver/transmitter on a per-channel basis.
- Ability to mix half-rate and full-rate between the channels with the same reference clock.
- Ability to configure each quad SERDES block independently with its own reference clock.
- STS-48 framing in SONET mode.
- Programmable enable of SONET scrambler/descrambler.
- Insertion and checking of link assignment values to facilitate interconnection and debugging of backplanes.
- Optional AIS-L insertion during loss-of-frame.
- Optional RDI-L insertion to indicate remote far-end defects for maintenance capabilities.
- SPE signal marks payload bytes when cell processing is not used.
- Frame alignment across multiple ORSO82G5 devices for work/protect switching at STS-768/STM-256 and above rates.
- In-band management and configuration through Transport OverHead (TOH) extraction/insertion.
- Supports transparent mode where internally generated overhead bytes can be overridden by user-generated overhead bytes from FPGA.
- Optional A1A2 corruption, B1 byte corruption, and K2 byte corruption for system debug purposes.
- Built-in boundary scan (*IEEE*® 1149.1 and 1149.2 JTAG), including the SERDES interface.
- FIFOs align incoming data across all eight channels (all eight channels, two groups of four channels, or four groups of two channels). Optional ability to bypass alignment FIFOs for asynchronous operation between channels. (Each channel includes its own clock and frame pulse).
- Optional cell processing blocks included. Cell processing includes cell creation, extraction, idle cell insertion and deletion asynchronous from line rates. Four cell sizes supported:
  - 77 bytes per cell (75 bytes of data payload)
  - 81 bytes per cell (79 bytes of data payload)

- 
- 85 bytes per cell (83 bytes of data payload)
  - 93 bytes per cell (91 bytes of data payload)
  - Automatic cell striping across either pairs of SERDES links or all eight SERDES links.
  - Addition of two 4K X 36 dual-port RAMs with access to the programmable logic.

## Programmable Features

- High-performance programmable logic:
    - 0.16  $\mu\text{m}$  7-level metal technology.
    - Internal performance of >250 MHz.
    - Over 600K usable system gates.
    - Meets multiple I/O interface standards.
    - 1.5 V operation (30% less power than 1.8 V operation) translates to greater performance.
  - Traditional I/O selections:
    - LVTTTL (3.3V) and LVCMOS (2.5 V, and 1.8 V) I/Os.
    - Per pin-selectable I/O clamping diodes provide 3.3 V PCI compliance.
    - Individually programmable drive capability:  
24 mA sink/12 mA source, 12 mA sink/6 mA source, or 6 mA sink/3 mA source.
    - Two slew rates supported (fast and slew-limited).
    - Fast-capture input latch and input Flip-Flop (FF)/latch for reduced input setup time and zero hold time.
    - Fast open-drain drive capability.
    - Capability to register 3-state enable signal.
    - Off-chip clock drive capability.
    - Two-input function generator in output path.
  - New programmable high-speed I/O:
    - Single-ended: GTL, GTL+, PECL, SSTL3/2 (class I and II), HSTL (Class I, III, IV), ZBT, and DDR.
    - Double-ended: LVDS, bused-LVDS, and LVPECL. Programmable (on/off), internal parallel termination (100  $\Omega$ ) is also supported for these I/Os.
  - New capability to (de)multiplex I/O signals:
    - New DDR on both input and output at rates up to 350 MHz (700 MHz effective rate).
    - New 2x and 4x downlink and uplink capability per I/O (i.e., 50 MHz internal to 200 MHz I/O).
  - Enhanced twin-quad Programmable Function Unit (PFU):
    - Eight 16-bit Look-Up Tables (LUTs) per PFU.
    - Nine user registers per PFU, one following each LUT, and organized to allow two nibbles to act independently, plus one extra for arithmetic operations.
    - New register control in each PFU has two independent programmable clocks, clock enables, local set/reset, and data selects.
    - New LUT structure allows flexible combinations of LUT4, LUT5, new LUT6, 4  $\rightarrow$  1 MUX, new 8  $\rightarrow$  1 MUX, and ripple mode arithmetic functions in the same PFU.
    - 32 x 4 RAM per PFU, configurable as single-port or dual-port. Create large, fast RAM/ROM blocks (128 x 8 in only eight PFUs) using the SLIC decoders as bank drivers.
    - Soft-Wired LUTs (SWL) allow fast cascading of up to three levels of LUT logic in a single PFU through fast internal routing which reduces routing congestion and improves speed.
    - Flexible fast access to PFU inputs from routing.
    - Fast-carry logic and routing to all four adjacent PFUs for nibble-wide, byte-wide, or longer arithmetic functions, with the option to register the PFU carry-out.
  - Abundant high-speed buffered and nonbuffered routing resources provide 2x average speed improvements over previous architectures.
  - Hierarchical routing optimized for both local and global routing with dedicated routing resources. This results in
-

- faster routing times with predictable and efficient performance.
- SLIC provides eight 3-statable buffers, up to a 10-bit decoder, and *PAL*<sup>™</sup>-like AND-OR-Invert (AOI) in each programmable logic cell.
  - New 200 MHz embedded quad-port RAM blocks, 2 read ports, 2 write ports, and 2 sets of byte lane enables. Each embedded RAM block can be configured as:
    - 1—512 x 18 (quad-port, two read/two write) with optional built in arbitration.
    - 1—256 x 36 (dual-port, one read/one write).
    - 1—1K x 9 (dual-port, one read/one write).
    - 2—512 x 9 (dual-port, one read/one write for each).
    - 2 RAMS with arbitrary number of words whose sum is 512 or less by 18 (dual-port, one read/one write).
    - Supports joining of RAM blocks.
    - Two 16 x 8-bit Content Addressable Memory (CAM) support.
    - FIFO 512 x 18, 256 x 36, 1Kx 9, or dual 512 x 9.
    - Constant multiply (8 x 16 or 16 x 8).
    - Dual variable multiply (8 x 8).
  - Embedded 32-bit internal system bus plus 4-bit parity interconnects FPGA logic, MicroProcessor Interface (MPI), embedded RAM blocks, and embedded standard cell blocks with 100 MHz bus performance. Included are built-in system registers that act as the control and status center for the device.
  - Built-in testability:
    - Full boundary scan (*IEEE* 1149.1 and Draft 1149.2 JTAG).
    - Programming and readback through boundary scan port compliant to *IEEE* Draft 1532:D1.7.
    - TS\_ALL testability function to 3-state all I/O pins.
    - New temperature-sensing diode.
  - Improved built-in clock management with Programmable Phase-Locked Loops (PPLLs) provide optimum clock modification and conditioning for phase, frequency, and duty cycle from 15 MHz up to 420 MHz. Multiplication of the input frequency up to 64x and division of the input frequency down to 1/64x possible.
  - New cycle stealing capability allows a typical 15% to 40% internal speed improvement after final place and route. This feature also enables compliance with many setup/hold and clock to out I/O specifications and may provide reduced ground bounce for output buses by allowing flexible delays of switching output buffers.
  - PCI local bus compliant for FPGA I/Os.
-

## Programmable Logic System Features

- Improved *PowerPC*® 860 and *PowerPC* II high-speed synchronous MicroProcessor Interface can be used for configuration, readback, device control, and device status, as well as for a general-purpose interface to the FPGA logic, RAMs, and embedded standard cell blocks. Glueless interface to synchronous *PowerPC* processors with user-configurable address space provided.
- New embedded *AMBA*™ specification 2.0 AHB system bus (*ARM*® processor) facilitates communication among the MicroProcessor Interface, configuration logic, Embedded Block RAM, FPGA logic, and embedded standard cell blocks.
- Variable size based readback of configuration data capability with the built-in MicroProcessor Interface and system bus.
- Internal, 3-state, and bidirectional buses with simple control provided by the SLIC.
- New clock routing structures for global and local clocking significantly increases speed and reduces skew.
- New local clock routing structures allow creation of localized clock trees.
- Two new edge clock routing structures allow up to six high-speed clocks on each edge of the device for improved setup/hold and clock to out performance.
- New Double-Data Rate (DDR) and zero-bus turn-around (ZBT) memory interfaces support the latest high-speed memory interfaces.
- New 2x/4x uplink and downlink I/O capabilities interface high-speed external I/Os to reduced speed internal logic.
- ispLEVER development system software. Supported by industry-standard CAE tools for design entry, synthesis, simulation, and timing analysis.
- Meets Universal Test and Operations PHY Interface for ATM (UTOPIA) levels 1, 2, and 3; as well as POS-PHY3.

## Description

### What Is an FPSC?

FPSCs, or Field-Programmable System Chips, are devices that combine field-programmable logic with ASIC or mask-programmed logic on a single device. FPSCs provide the time to market and the flexibility of FPGAs, the design effort savings of using soft Intellectual Property (IP) cores, and the speed, design density, and economy of ASICs.

### FPSC Overview

Lattice's Series 4 FPSCs are created from Series 4 ORCA FPGAs. To create a Series 4 FPSC, several columns of Programmable Logic Cells (see FPGA Logic Overview section for FPGA logic details) are added to an embedded logic core. Other than replacing some FPGA gates with ASIC gates, at greater than 10:1 efficiency, none of the FPGA functionality is changed—all of the Series 4 FPGA capability is retained: Embedded Block RAMs, MPI, PCMs, boundary scan, etc. The columns of programmable logic are replaced at the right of the device, allowing pins from the replaced columns to be used as I/O pins for the embedded core. The remainder of the device pins retain their FPGA functionality.

### FPSC Gate Counting

The total gate count for an FPSC is the sum of its embedded core (standard-cell/ASIC gates) and its FPGA gates. Because FPGA gates are generally expressed as a usable range with a nominal value, the total FPSC gate count is sometimes expressed in the same manner. Standard-cell ASIC gates are, however, 10 to 25 times more silicon-area efficient than FPGA gates. Therefore, an FPSC with an embedded function is gate equivalent to an FPGA with a much larger gate count.

### FPGA/Embedded Core Interface

The interface between the FPGA logic and the embedded core has been enhanced to allow for a greater number of interface signals than on previous FPSC architectures. Compared to bringing embedded core signals off-chip, this on-chip interface is much faster and requires less power. All of the delays for the interface are precharacterized and accounted for in the ispLEVER Development System.

ORCA Series 4 based FPSCs expand this interface by providing a link between the embedded block and the multi-master 32-bit system bus in the FPGA logic. This system bus allows the core easy access to many of the FPGA logic functions including the Embedded Block RAMs and the MicroProcessor Interface.

Clock spines also can pass across the FPGA/embedded core boundary. This allows for fast, low-skew clocking between the FPGA and the embedded core. Many of the special signals from the FPGA, such as DONE and global set/reset, are also available to the embedded core, making it possible to fully integrate the embedded core with the FPGA as a system.

For even greater system flexibility, FPGA configuration RAMs are available for use by the embedded core. This allows for user-programmable options in the embedded core, in turn allowing for greater flexibility. Multiple embedded core configurations may be designed into a single device with user-programmable control over which configurations are implemented, as well as the capability to change core functionality simply by reconfiguring the device.

### ispLEVER Development System

The ispLEVER development system is used to process a design from a netlist to a configured FPGA. This system is used to map a design onto the ORCA architecture, and then place and route it using ispLEVER development system timing-driven tools. The development system also includes interfaces to, and libraries for, other popular CAE tools for design entry, synthesis, simulation, and timing analysis.

### FPSC Design Kit

Development is facilitated by an FPSC design kit which, together with ispLEVER and third-party synthesis and simulation engines, provides all software and documentation required to design and verify an FPSC implementation. Included in the kit are the FPSC configuration manager, *Synopsys Smart Model*<sup>®</sup>, and/or compiled *Verilog* simula-

---

---

tion model, *HSPICE* and/or IBIS models for I/O buffers, and complete online documentation. The kit's software couples with ispLEVER, providing a seamless FPSC design environment. More information can be obtained by visiting the *ORCA* website or contacting a local sales office, both listed on the last page of this document.

## FPGA Logic Overview

The *ORCA* Series 4 architecture is a new generation of SRAM-based programmable devices from Lattice. It includes enhancements and innovations geared toward today's high-speed systems on a single chip. Designed with networking applications in mind, the Series 4 family incorporates system-level features that can further reduce logic requirements and increase system speed. *ORCA* Series 4 devices contain many new patented enhancements and are offered in a variety of packages and speed grades.

The hierarchical architecture of the logic, clocks, routing, RAM, and system-level blocks create a seamless merge of FPGA and ASIC designs. Modular hardware and software technologies enable system-on-chip integration with true plug-and-play design implementation.

The architecture consists of four basic elements: Programmable Logic Cells (PLCs), programmable I/O cells (PIOs), Embedded Block RAMs (EBRs), and system-level features. These elements are interconnected with a rich routing fabric of both global and local wires. An array of PLCs are surrounded by common interface blocks which provide an abundant interface to the adjacent PLCs or system blocks. Routing congestion around these critical blocks is eliminated by the use of the same routing fabric implemented within the programmable logic core. Each PLC contains a PFU, SLIC, local routing resources, and configuration RAM. Most of the FPGA logic is performed in the PFU, but decoders, *PAL*-like functions, and 3-state buffering can be performed in the SLIC. The PIOs provide device inputs and outputs and can be used to register signals and to perform input demultiplexing, output multiplexing, uplink and downlink functions, and other functions on two output signals. Large blocks of 512 x 18 quad-port RAM complement the existing distributed PFU memory. The RAM blocks can be used to implement RAM, ROM, FIFO, multiplier, and CAM. Some of the other system-level functions include the MPI, PLLs, and the Embedded System Bus (ESB).

## PLC Logic

Each PFU within a PLC contains eight 4-input (16-bit) LUTs, eight latches/FFs, and one additional Flip-Flop that may be used independently or with arithmetic functions.

The PFU is organized in a twin-quad fashion; two sets of four LUTs and FFs that can be controlled independently. Each PFU has two independent programmable clocks, clock enables, local set/reset, and data selects. LUTs may also be combined for use in arithmetic functions using fast-carry chain logic in either 4-bit or 8-bit modes. The carry-out of either mode may be registered in the ninth FF for pipelining. Each PFU may also be configured as a synchronous 32 x 4 single- or dual-port RAM or ROM. The FFs (or latches) may obtain input from LUT outputs or directly from invertible PFU inputs, or they can be tied high or tied low. The FFs also have programmable clock polarity, clock enables, and local set/reset.

The SLIC is connected from PLC routing resources and from the outputs of the PFU. It contains eight 3-state, bidirectional buffers, and logic to perform up to a 10-bit AND function for decoding, or an AND-OR with optional INVERT to perform *PAL*-like functions. The 3-state drivers in the SLIC and their direct connections from the PFU outputs make fast, true, 3-state buses possible within the FPGA, reducing required routing and allowing for real-world system performance.

## Programmable I/O

The Series 4 PIO addresses the demand for the flexibility to select I/Os that meet system interface requirements. I/Os can be programmed in the same manner as in previous *ORCA* devices, with the additional new features which allow the user the flexibility to select new I/O types that support High-Speed Interfaces.

Each PIO contains four programmable I/O pads and is interfaced through a common interface block to the FPGA array. The PIO is split into two pairs of I/O pads with each pair having independent clock enables, local set/reset, and global set/reset. On the input side, each PIO contains a programmable latch/Flip-Flop which enables very fast



latching of data from any pad. The combination provides for very low setup requirements and zero hold times for signals coming on-chip. It may also be used to demultiplex an input signal, such as a multiplexed address/data signal, and register the signals without explicitly building a demultiplexer with a PFU.

On the output side of each PIO, an output from the PLC array can be routed to each output Flip-Flop, and logic can be associated with each I/O pad. The output logic associated with each pad allows for multiplexing of output signals and other functions of two output signals.

The output FF, in combination with output signal multiplexing, is particularly useful for registering address signals to be multiplexed with data, allowing a full clock cycle for the data to propagate to the output. The output buffer signal can be inverted, and the 3-state control can be made active-high, active-low, or always enabled. In addition, this 3-state signal can be registered or nonregistered.

The Series 4 I/O logic has been enhanced to include modes for speed uplink and downlink capabilities. These modes are supported through shift register logic, which divides down incoming data rates or multiplies up outgoing data rates. This new logic block also supports high-speed DDR mode requirements where data are clocked into and out of the I/O buffers on both edges of the clock.

The new programmable I/O cell allows designers to select I/Os which meet many new communication standards permitting the device to hook up directly without any external interface translation. They support traditional FPGA standards as well as high-speed, single-ended, and differential-pair signaling. Based on a programmable, bank-oriented I/O ring architecture, designs can be implemented using 3.3 V, 2.5 V, 1.8 V, and 1.5 V referenced output levels.

## Routing

The abundant routing resources of the Series 4 architecture are organized to route signals individually or as buses with related control signals. Both local and global signals utilize high-speed buffered and nonbuffered routes. One PLC segmented (x1), six PLC segmented (x6), and bused half chip (xHL) routes are patterned together to provide high connectivity with fast software routing times and high-speed system performance.

Eight fully distributed primary clocks are routed on a low-skew, high-speed distribution network and may be sourced from dedicated I/O pads, PLLs, or the PLC logic. Secondary and edge-clock routing is available for fast regional clock or control signal routing for both internal regions and on device edges. Secondary clock routing can be sourced from any I/O pin, PLLs, or the PLC logic.

The improved routing resources offer great flexibility in moving signals to and from the logic core. This flexibility translates into an improved capability to route designs at the required speeds when the I/O signals have been locked to specific pins.

## System-Level Features

The Series 4 also provides system-level functionality by means of its MicroProcessor Interface, Embedded System Bus, quad-port Embedded Block RAMs, universal Programmable Phase-Locked Loops, and the addition of highly tuned networking specific Phase-Locked Loops. These functional blocks allow for easy glueless system interfacing and the capability to adjust to varying conditions in today's high-speed networking systems.

## MicroProcessor Interface

The MPI provides a glueless interface between the FPGA and *PowerPC* microprocessors. Programmable in 8-, 16-, and 32-bit interfaces with optional parity to the *Motorola*® *PowerPC* 860 bus, it can be used for configuration and readback, as well as for FPGA control and monitoring of FPGA status. All MPI transactions utilize the Series 4 Embedded System Bus at 66 MHz performance.

A system-level MicroProcessor Interface (MPI) to the FPGA user-defined logic following configuration, through the system bus, including access to the Embedded Block RAM and general user-logic, is provided by the MPI. The MPI supports burst data read and write transfers, allowing short, uneven transmission of data through the interface by

---

including data FIFOs. Transfer accesses can be single beat (1 x 4 bytes or less), 4-beat (4 x 4 bytes), 8-beat (8 x 2 bytes), or 16-beat (16 x 1 bytes).

## System Bus

An on-chip, multimaster, 32-bit system bus with 4-bit parity facilitates communication among the MPI, configuration logic, FPGA control, and status registers, Embedded Block RAMs, as well as user logic. Utilizing the *AMBA* specification Rev 2.0 AHB protocol, the Embedded System Bus offers arbiter, decoder, master, and slave elements. Master and slave elements are also available for the user-logic and a slave interface is used for control and status of the embedded backplane transceiver portion of the ORSO82G5.

The system bus control registers can provide control to the FPGA such as signaling for reprogramming, reset functions, and PLL programming. Status registers monitor INIT, DONE, and system bus errors. An interrupt controller is integrated to provide up to eight possible interrupt resources. Bus clock generation can be sourced from the Micro-Processor Interface clock, configuration clock (for slave configuration modes), internal oscillator, user clock from routing, or from the port clock (for JTAG configuration modes).

## Phase-Locked Loops

Up to eight PLLs are provided on each Series 4 device, with four user PLLs generally provided for FPSCs. Programmable PLLs can be used to manipulate the frequency, phase, and duty cycle of a clock signal. Each PLL is capable of manipulating and conditioning clocks from 15 MHz to 420 MHz. Frequencies can be adjusted from 1/8x to 8x, the input clock frequency. Each programmable PLL provides two outputs that have different multiplication factors but can have the same phase relationships. Duty cycles and phase delays can be adjusted in 12.5% of the clock period increments. An automatic input buffer delay compensation mode is available for phase delay. Each PLL provides two outputs that can have programmable (12.5% steps) phase differences.

## Embedded Block RAM

New 512 x 18 quad-port RAM blocks are embedded in the FPGA core to significantly increase the amount of memory and complement the distributed PFU memories. The EBRs include two write ports, two read ports, and two byte lane enables which provide four-port operation. Optional arbitration between the two write ports is available, as well as direct connection to the high-speed system bus.

Additional logic has been incorporated to allow significant flexibility for FIFO, constant multiply, and two-variable multiply functions. The user can configure FIFO blocks with flexible depths of 512K, 256K, and 1K including asynchronous and synchronous modes and programmable status and error flags. Multiplier capabilities allow a multiple of an 8-bit number with a 16-bit fixed coefficient or vice versa (24-bit output), or a multiply of two 8-bit numbers (16-bit output). On-the-fly coefficient modifications are available through the second read/write port.

Two 16 x 8-bit CAMs per embedded block can be implemented in single match, multiple match, and clear modes. The EBRs can also be preloaded at device configuration time.

## Configuration

The FPGAs functionality is determined by internal configuration RAM. The FPGAs internal initialization/configuration circuitry loads the configuration data at powerup or under system control. The configuration data can reside externally in an EEPROM or any other storage media. Serial EEPROMs provide a simple, low pin-count method for configuring FPGAs.

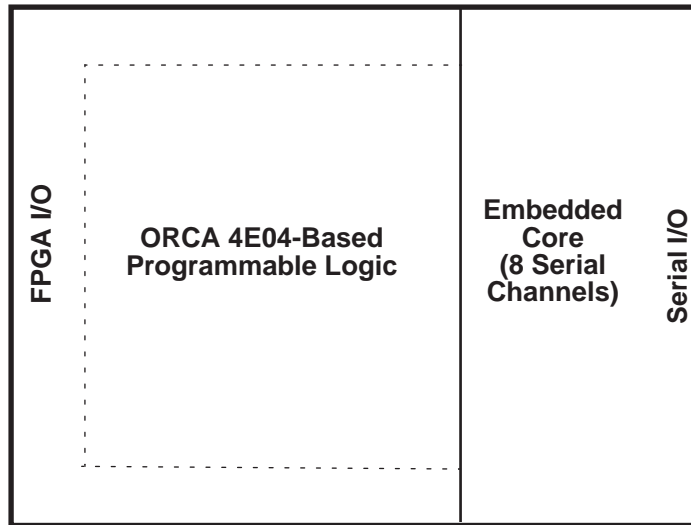
The RAM is loaded by using one of several configuration modes. Supporting the traditional master/slave serial, master/slave parallel, and asynchronous peripheral modes, the Series 4 also utilizes its MicroProcessor Interface and Embedded System Bus to perform both programming and readback. Daisy chaining of multiple devices and partial reconfiguration are also permitted.

Other configuration options include the initialization of the embedded-block RAM memories and FPSC memory as well as system bus options and bit stream error checking. Programming and readback through the JTAG (*IEEE 1149.2*) port is also available meeting In-System Programming (ISP) standards (*IEEE 1532 Draft*).

## ORSO82G5 Overview

The ORSO82G5 FPSC provides high-speed backplane transceivers combined with FPGA logic. The ORSO82G5 device is based on the 1.5 V OR4E04 ORCA FPGA and has a 36 x 36 array of Programmable Logic Cells (PLCs). The embedded core which contains the backplane transceivers is attached to the right side of the device and is integrated directly into the FPGA array. A top level diagram of the basic chip configuration is shown in Figure 1

**Figure 1. ORSO82G5 Basic Chip Configuration**



The ORSO82G5 supports aggregate bandwidths up to 21.6 Gbits/s and is targeted towards users needing high-speed backplane interfaces for SONET and other non-SONET proprietary backplanes. For non-SONET applications, all SONET functionality is hidden from the user and no prior networking knowledge is required.

Built using Series 4 reconfigurable System-on-a-Chip (SoC) architecture, the ORSO82G5 contains the FPGA base array and an embedded core supporting eight serial data channels, with clock and data recovery functions, and provides SONET framing, scrambling/descrambling and cell processing on a single monolithic chip and enables high-speed asynchronous serial data transfer between system devices. Devices can be on the same PC-board, on separate boards connected across a backplane or connected by cables. The ORSO82G5 is completely pin-compatible with the ORT82G5 device.

The ORSO82G5 is considered a pseudo-SONET device because it does not support full overhead processing, pointer processing or meet all SONET jitter/timing requirements. The ORSO82G5 is designed primarily for use as a SONET backplane device and not for network termination. Although it formats and processes data as SONET frames, because the embedded core is not fully SONET compliant on a stand-alone basis, it can not terminate data directly on a SONET ring without additional functionality being implemented in the FPGA logic.

The ORSO82G5 embedded core does support the following:

- Section/Line Overhead: A1/A2 (framing bytes), B1 (BIP-8), K2 (APS)
- Alarms: OOF (Out Of Frame), B1 error, RDI
- Automatic Transport OverHead (TOH) generation and insertion
- AIS-L insertion
- SPE signal generation which support +/- stuff events (but no pointer processing)

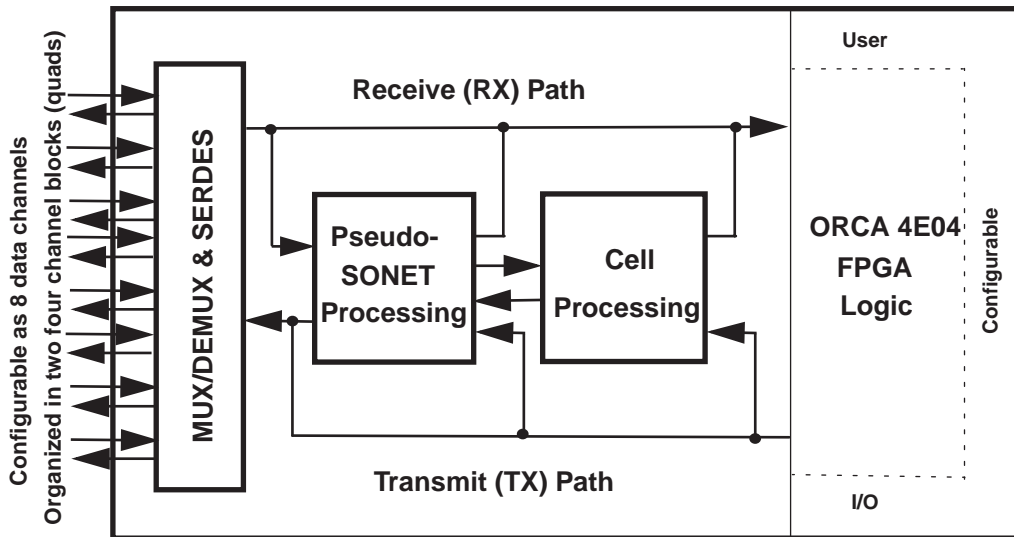
**Embedded Core Overview**

The functions in the embedded core portion of the ORSO82G5 device include:

- Eight channel 2.7 Gbits/s serializer/deserializer functions with Clock and Data Recovery (CDR).
- MicroProcessor Interface (MPI) to the FPGA logic and embedded core logic via the Series 4 system bus
- Support for OC-48 and OC-192 (in quad OC-48) formats.
- SONET framing, scrambling and SONET Mode channel alignment.
- Performance monitoring functions such as Bit Interleaved Parity (BIP-8) generation and checking and Out-Of-Frame (OOF) and Remote Defect Indication (RDI-L) detection.
- Cell Mode cell creation and extraction, idle cell insertion/deletion, destriping and striping functions.
- Additionally, there are two independent memory blocks in the core. Each embedded RAM block has a capacity of 4K word by 36 bits.

The ORSO82G5 embedded core contains an eight-channel clock and data recovery macrocell and logical blocks performing functions such as SONET framing, scrambling/descrambling and cell processing. The eight channels each operate at up to 2.7 Gbits/s with per channel CDR functionality. The CDR interface enables high-speed asynchronous serial data transfer between system devices. Devices can be on the same PC-board, on separate boards connected across a backplane, or connected by cables. Figure 2 shows a top level block diagram of the backplane driver logic in the embedded core (embedded RAM not shown).

**Figure 2. Top Level Block Diagram ORSO82G5 Embedded Core**



---

## ORSO82G5 Main Operating Modes - Overview

The ORSO82G5 supports eight 1.0 to 2.7 Gbits/s serial data channels which can operate independently or can be combined together (aligned) to achieve higher bit rates. The mode of operation of the core is defined by a set of control registers, which can be written through the system bus interface. The status of the core is stored in a set of status registers, which can be read through the system bus interface.

The 8 serial data channels supports OC-48 rates on each channel. OC-192 is also supported but is transmitted and received in quad OC-48 links. The 2.488 Gbits/s scrambled data stream conforms to the GR-255 specified polynomial sequence of  $1+x^6+x^7$ .

There are three main operating modes in the ORSO82G5 as described below:

- SERDES only (bypass) mode
- SONET mode
- Cell mode
  - Two-link sub-mode
  - Eight-link sub-mode

There are sub-modes that can be derived by enabling or disabling certain functions through programmable register bits. Also, in Cell mode, either the two-link alignment mode, for up to four alignment groups, or the eight-link alignment mode, where all eight links are combined into a single group, may be selected.

Data are processed in the transmit direction (FPGA to Backplane) as follows:

- In the **SERDES only mode**, there is the option to bypass all of the SONET and cell functions and pass raw 32-bit data from the FPGA into the 32:8 MUX block. In this mode, the user is responsible for providing an adequate ones transition density in the transmitted stream for clock and data recovery at the receive end of the link.
- In the **SONET mode**, a SONET frame is constructed around the input data and overhead bytes are inserted where appropriate. The 32-bits of data per channel are scrambled before being converted to 8-bits by the 32:8 MUX block and serialized by the SERDES.
- In the **cell mode**, 160 bits of data from the FPGA is sent to the Output Port Controller-8 block (also referred to as OPC8 since it services eight links) or 40 bits of data to Output Port Controller-2 block (referred to as OPC2\_[A1,A2,B1,B2]) which perform cell striping across the different SERDES links. The cells are then transferred to the SONET clock domain of 77.76 MHz through a Transmit FIFO. A SONET frame is constructed around the cell payload and overhead bytes inserted where appropriate before being sent to the MUX block. The data are then converted to 8 bits by the 32:8 MUX block before being serialized by the SERDES.

Data are processed in the receive direction (Backplane to FPGA) as follows:

- In the **SERDES only mode**, there is the option to bypass all of the SONET and cell functions and pass raw 32-bit data from the 8:32 DEMUX block into the FPGA interface.
- In the **SONET mode**, the descrambled data are sent to an alignment FIFO that performs lane-to-lane deskew and aligns data within an alignment group to a single clock domain and frame pulse. The SPE indicator is provided to the FPGA along with 32 bits of aligned data. There is an option to bypass the alignment FIFOs and pass data directly from the descrambler to the FPGA. This mode is programmable and can be controlled per channel.
- In the **cell mode**, the SONET framed data are descrambled and passed into a cell extractor which extracts cells from the payload portion of the SONET frame. The cells are passed through a FIFO which performs lane-to-lane deskew and a clock domain transfer from 77.76 MHz to 156 MHz. The cells are passed into the input port controller block (referred to as IPC8 or IPC2\_[A1, A2, B1, B2] depending on whether 8 or two links are serviced) which performs cell destriping before sending them to the FPGA interface. This cell processing feature makes ORSO82G5 ideal for interfacing devices with proprietary data formats across a backplane.

**Embedded Core Functional Blocks - Overview**

Each channel contains transmit path and receive path logic as shown in Figure 3 and Figure 4. Data are processed on a channel by channel basis in the SERDES only and SONET modes. Channel by channel processing is also performed in the Cell mode except in the Input Port Controller (IPC) and Output Port Controller (OPC) blocks. Support for loopback is also provided but is not shown in Figure 3. The following sections will give an overview of the pseudo SONET protocol supported by the ORSO82G5 and a top level overview of the macrocells, which provide the SERDES Only, SONET and Cell Mode functionality.

**Figure 3. Top Level Overview, TX Path Logic, Single Channel**

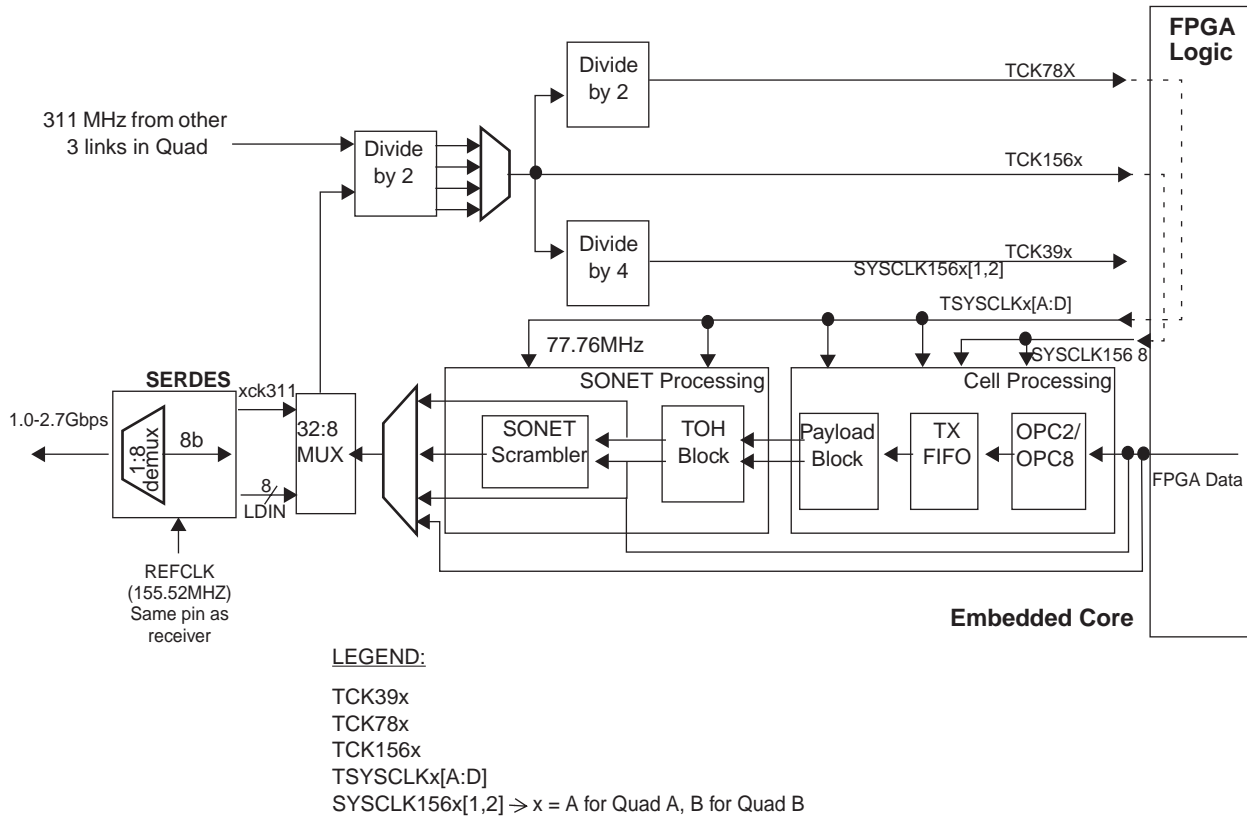
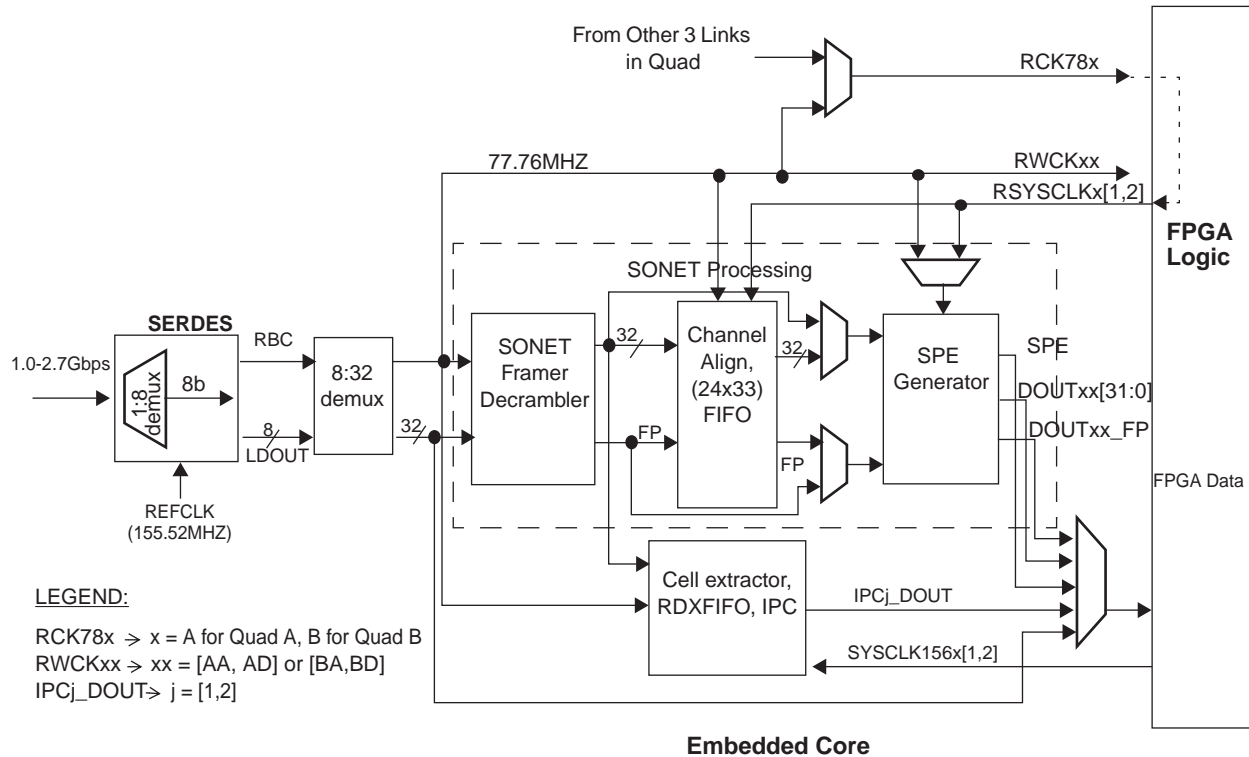


Figure 4. Top Level Overview, RX Path Logic, Single Channel



**SERializer and DESerializer (SERDES)**

Each SERDES block is a quad transceiver containing 4 channels for serial data transmission, with a per-channel selectable data rate of 1.0—2.7 Gbits/s. Each SERDES block features high-speed CML interfaces and is designed to operate in SONET backplane applications. The transceiver is controlled and configured via an 8-bit, 16-bit or 32-bit MicroProcessor Interface through the FPGA. Each channel has dedicated registers that are readable and writable. The device also contains global registers for control of common circuitry and functions. There are two SERDES blocks (or two quads), A and B, in the embedded portion of the device. Each quad supports four full duplex serial links (Slice A contains channels AA, AB, AC, and AD while slice B contains channels BA, BB, BC, and BD).

Each SERDES block contains its own dedicated PLLs for transmit and receive clock generation. The user provides a reference clock of the appropriate frequency (one per SERDES quad). The receiver PLLs extract the clock from the serial input data and retime the data with the recovered clock. Clock divider circuitry also provides reference clocks for the FPGA logic.

**8:32 MUX and 32:8 DEMUX**

The purpose of the MUX/DEMUX block is to provide a wide, low-speed interface at the FPGA portion of the ORSO82G5 for each channel or data lane. The interface to the SERDES macro runs at 1/8th the bit rate of the data lane. The MUX/DEMUX converts the data rate and bit-width so the FPGA core can run at 1/4th this frequency (i.e., 1/32nd the SERDES rate). This gives a range of 31.25 MHz—84.38 MHz for the data crossing the FPGA/embedded core boundary on SERDES only and SONET modes.

**SONET Transmit OverHead (TOH) Processing, Framer and Scrambler/Descrambler (SONET and Cell Modes)**

In the transmit direction, the TOH block is responsible for processing the 144 (48 x 3) TOH bytes at the beginning of each row of the transport frame. The TOH bytes may be transmitted transparently from the FPGA logic or may be inserted by the TOH block (AUTO\_TOH mode).

---

The TOH block can perform A1/A2 corruption by inverting the A1/A2 bytes under software control. The block can also force B1 errors by inverting the B1 byte or inject a fault indication by forcing the K2 byte to "00000110". In the receive direction, the errors will be detected and alarms set. In SONET mode, all TOH bytes can optionally be sent transparently from the FPGA.

For each of the receive channels, the framer logic outputs 4 bytes (32-bits) of received data that are frame-aligned and a frame pulse that is one clock-wide. The framer is responsible also for detecting the in-frame and Out-Of-Frame status of the incoming data and sends out alarms (interrupts) on detecting an Out-Of-Frame (OOF) state.

In the transmit direction, the scrambler logic scrambles the outgoing data using the standard SONET polynomial  $1 + x^6 + x^7$ . In the receive direction, the descrambler logic descrambles the incoming data using the same polynomial.

#### **Multichannel Alignment FIFO and SPE Generation (for SONET mode)**

In SONET mode, the incoming data on the 8 channels can be independent of one another or can be synchronized in several ways. Two channels within a SERDES quad can be aligned together: channel A and B and/or channel C and D. Alternately, all four channels in a SERDES quad can be aligned together to form a communication channel with a bandwidth of 10 Gbits/s. Finally, the alignment can be extended across SERDES to align all 8 channels.

Individual channels within an alignment group can be disabled (i.e., power down) without disrupting other channels. A disabled channel can also be aligned to its group without a disruption to the remaining channels. This holds true only if the disabled channel had been previously enabled during alignments.

#### **Cell Extraction, Striping and Destriping (for Cell Mode)**

In the cell mode, cells are distributed across two-links or across eight-links. In the TX direction, cell data from the FPGA are "striped" across links within a link group. Link groups can be in up to four two-link groups or one group of eight-links. In the receive direction cell processing blocks extract cells from the link groups and send them to the FPGA. This function is referred to as "destriping".

#### **Dual Port RAMs**

There are two independent memory blocks in the core. Each memory block has a capacity of 4K words by 36 bits. It has one read port, one write port, and four byte-write-enable (active-low) signals. The read data from the memory block is registered so that it works as a pipelined synchronous memory block. These memory blocks are completely independent of the backplane driver blocks. They are only accessible from the FPGA logic and are not connected to the system bus.

#### **Loopback - Overview**

There are two types of loopback that can be utilized inside the embedded core of the ORSO82G5, near end loopback and far end (line side) loopback. Both of these loopbacks are controlled by control registers inside the ORSO82G5 core, which are accessible from the system bus and the MicroProcessor Interface (MPI).

#### **FPSC Configuration - Overview**

Configuration of the ORSO82G5 occurs in two stages: FPGA bit stream configuration and embedded core setup.

Prior to becoming operational, the FPGA goes through a sequence of states, including power-up, initialization, configuration, start-up, and operation. The FPGA logic is configured by standard FPGA bit stream configuration means as discussed in the Series 4 FPGA data sheet.

The options for the embedded core are set via registers that are accessed through the FPGA system bus. The system bus can be driven by an external PPC compliant microprocessor via the MPI block or via a user master interface in FPGA logic. A simple IP block, that drives the system by using the user interface and uses very little FPGA logic, is available in the *MPI/System Bus* application note (TN1017). This IP block sets up the embedded core via a state machine and allows the ORSO82G5 to work in an independent system without an external MicroProcessor Interface.



---

## ORSO82G5 Embedded Core Detailed Description

The ORSO82G5 consists of 8 channels each with a high-speed SERDES macro that performs clock data recovery, serializing and deserializing functions. There is also additional logic for SONET mode and Cell mode data synchronization formatting and scrambling/descrambling. For all modes, the data paths can be characterized as the transmit path (FPGA to backplane) and receive path (backplane to FPGA); however the interface signal assignments between the FPGA logic and the core differ depending on the operating mode selected.

The three main operating modes in the ORSO82G5 are:

- SERDES only mode
- SONET mode
- Cell mode
  - Two-link sub-mode
  - Eight-link sub-mode

The SONET and Cell modes each support sub-modes that can be selected by enabling or disabling certain functions through programmable register bits. Following the basic TX and RX architecture descriptions, the data formatting and logical implementations supporting each of the operational modes are described.

### Top Level Description - Transmitter (TX) and Receiver (RX) Architectures

The next sections give a top level description of the transmitter and receive architectures. The high-speed transmit and receive serial data can operate at 1.0—2.7 Gbits/s depending on the state of the control bits from the Micro-Processor Interface and provided reference clock. For all of the architecture and clock distribution descriptions, however, the standard SONET STS-48 rate of 2,488.32 Mbits/s (i.e., REFCLK\_[P:N] = 155.52 MHz for the full rate modes) is assumed.

#### Transmitter Architecture

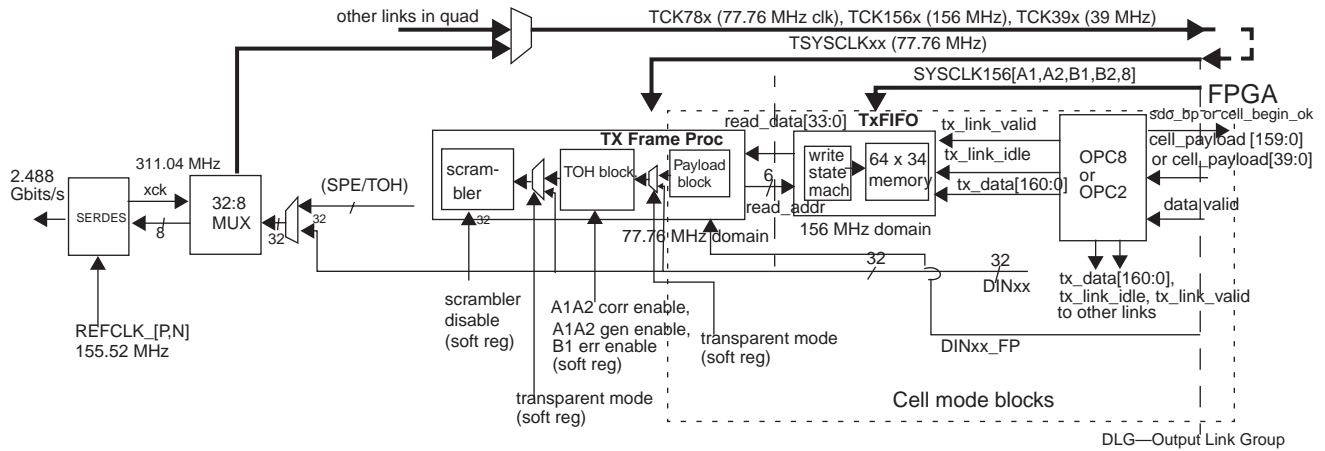
The transmitter section accepts parallel data for transmission from the FPGA logic, formats it for transmission and serializes the data. It also accepts the low-speed reference clock at the REFCLK input and uses this clock to synthesize the internal high-speed serial bit clock. The serialized transmitted data are available at the differential CML output pins to drive either an optical transmitters, coaxial media or a circuit board backplane.

The top level transmit architecture is shown in Figure 5. The main logical blocks in the transmit path are:

- Output Port Controllers (OPCs) which contain the cell processing logic.
- SONET processing logic.
- Transmit SERDES and 32:8 MUX.

Depending on the mode of operation, the FPGA to backplane data path may include or bypass the various logical blocks.

Figure 5. Transmitter Architecture



**Receiver Architecture**

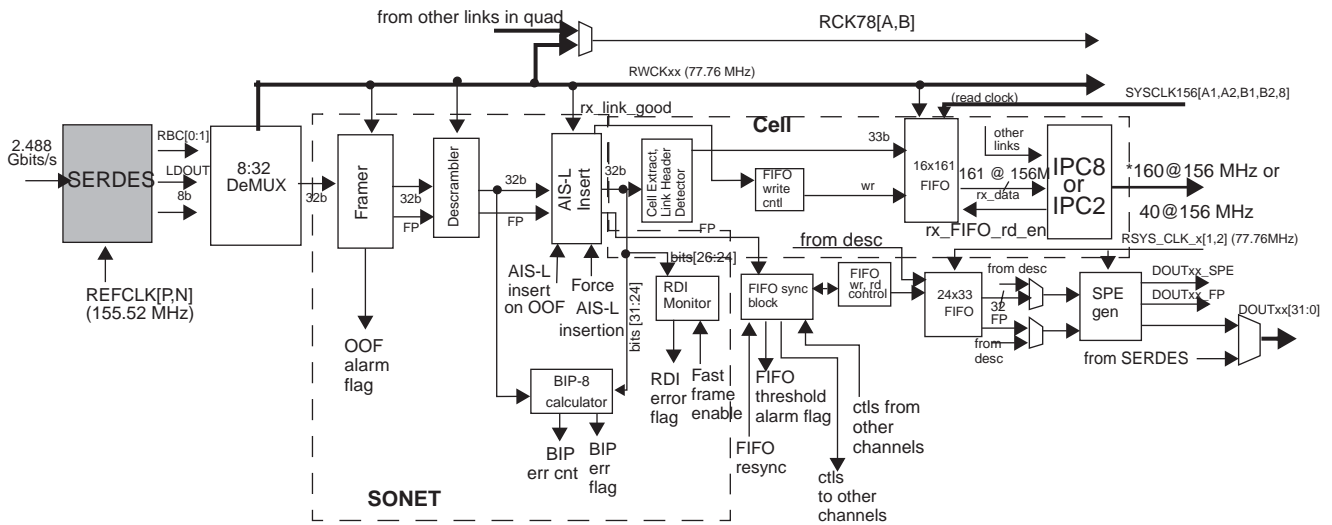
The receiver section receives high-speed serial data at the external differential CML input pins. These data are fed to the clock recovery section which generates a recovered clock and retimes the data. Therefore the receive clocks are asynchronous between channels. The data are then optionally framed, reformatted, aligned and passed to the FPGA logic in various parallel data formats.

The top level receiver architecture is shown in Figure 5. The main logical blocks in the receive path are:

- Receive SERDES and 8:32 DEMUX.
- SONET processing logic.
- Input Port Controllers (IPCs) which contain the cell processing logic.

Depending on the mode of operation, the FPGA to backplane data path may include or bypass the various logical blocks.

Figure 6. Receive Architecture Block Diagram



\* - 160 bits from eight-links combined

In either the SONET or Cell mode, data from the DEMUX is then passed through a framer which aligns and frames the data. Data are then processed based on Cell mode or SONET mode selection.

In the SONET mode, the descrambled data are sent to an alignment FIFO that performs lane-to-lane deskew and optionally aligns data within a two, four or eight channel alignment group. In addition, supervisory features such as BIP error check, OOF check, RDI monitoring and AIS-L insertion during OOF are also implemented. All the supervisory features are controlled through programmable register bits.

In the cell mode, the framed data are then descrambled and passed into a cell extractor which extracts cells from the payload portion of the SONET frame. The cells are passed through a FIFO that performs lane-to-lane deskew and a clock domain transfer from 77.76 MHz to 156 MHz. A key feature in cell mode is the ability to use idle cell insertion and deletion for automatic rate matching between the clock domains. The cells are then passed to the IPC8 or IPC2 block which perform cell destripping before sending the cells to the FPGA logic across the Core/FPGA interface.

**SERDES Transmit and Receive PLLs**

The high-speed transmit and receive serial data can operate at 1.0—2.7 Gbits/s depending on the state of the control bits from the MicroProcessor Interface. Figure 2 shows the relationship between the data rates, the reference clock, and the internal transmit TCK78x clocks.

**Table 2. Transmit PLL Clock and Data Rates**

Data Rate	Reference Clock	TCK78x	Rate
1.0 Gbits/s	125.00 MHz	31.25 MHz	Half
1.244 Gbits/s	155.52 MHz	38.88 MHz	Half
1.35 Gbits/s	168.75 MHz	42.19 MHz	Half
2.0 Gbits/s	125.00 MHz	61.50 MHz	Full
2.488 Gbits/s	155.52 MHz	77.76 MHz	Full
2.7 Gbits/s	168.75 MHz	84.38 MHz	Full

Note:The selection of full-rate or half-rate for a given reference clock speed is set by the TXHR bit in the transmit control register and can be set per channel. (For cell mode all channels of a group must have the same TXHR selection.)

Note:In selections of this data sheet, the differential clocks are simply referred to as the reference clock REFCLK\_[A:B]. REFCLK\_[A:B] is equivalent to REFINP\_[A:B] and REFINN\_[A:B].

The receiver section receives high-speed serial data at its differential CML input port. These data are fed to the clock recovery section which generates a recovered clock and retimes the data. This means that the receive clocks are asynchronous between channels. The retimed data are deserialized and presented as a 8-bit parallel data on the output port. RWCKx receive byte clocks are divide-by-4 clocks of the RBC0 or RBC1 (recovered clock) clocks provided by the SERDES. This is the clock used in the internal receive functions of the embedded core.

The reference clock is also used by the receive PLL for operation when the input data are not toggling appropriately. Table 3 shows the relationship between the data rates, the reference clock, and the RWCKx clocks.

**Table 3. Receive PLL Clock and Data Rates**

Data Rate	Reference Clock	RWCKx Clocks	Rate
1.0 Gbits/s	125.00 MHz	31.25 MHz	Half
1.244 Gbits/s	155.52 MHz	38.88 MHz	Half
1.35 Gbits/s	168.75 MHz	42.19 MHz	Half
2.0 Gbits/s	125.00 MHz	61.50 MHz	Full
2.488 Gbits/s	155.52 MHz	77.76 MHz	Full
2.7 Gbits/s	168.75 MHz	84.38 MHz	Full

Note:The selection of full-rate or half-rate for a given reference clock speed is set by the RXHR bit in the receive control register and can be set per channel. (For cell mode all channels of a group must have the same RXHR selection.)

The differential reference clock is distributed to all four channels per quad SERDES block. Each channel has a differential buffer to isolate the clock from the other channels. The input clock is preferably a differential signal; however, the device can operate with a single-ended input. The input reference clock directly impacts the transmit data eye, so the clock should have low jitter. In particular, jitter components in the dc—5 MHz range should be minimized.

**Detailed Description - SERDES Only Mode**

The SERDES only (or bypass) mode is the simplest of the three operating modes for the ORSO82G5. In this mode, all of the SONET and cell logic block functions are bypassed and data are transferred directly between the MUX and DEMUX blocks to and from the FPGA interface. This mode is utilized when the user wants to perform all data processing and uses only the SERDES portion of the Embedded Core. For example, this mode could be utilized to replace an existing design using stand-alone SERDES and FPGAs.

The basic data paths in the transmit and receive directions are shown in Figure 7. In general, the descriptions in this section are written to describe the SERDES only mode, although the "SERDES blocks" are also used in SONET and cell mode operation. At the backplane interface, data are transmitted and received serially over eight pairs of differential 1.0 to 2.7 Gbit/s links. At the FPGA/Embedded Core interface, the data are transferred across eight 32-bit buses. The SERDES blocks themselves are organized as two blocks (or quads), each supporting four series links. Each of the eight data paths is identified with a quad and channel identifier (i.e., AA,...,BD).

Each channel has a 32-bit TX bus, 32-bit RX bus, a recovered clock, and a transmit clock input.

**Figure 7. Basic Data Flows - SERDES Only Mode**

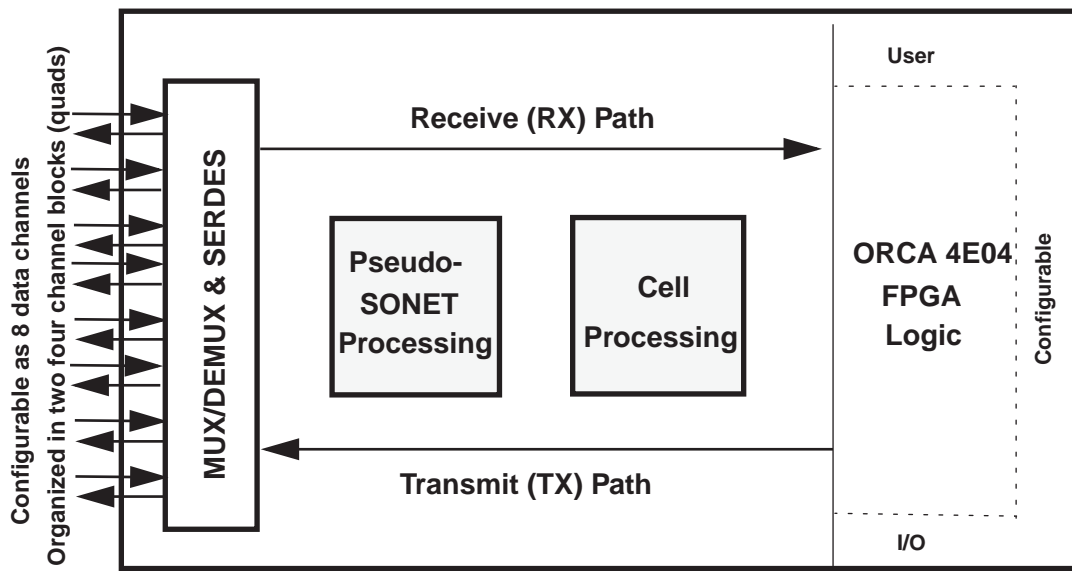
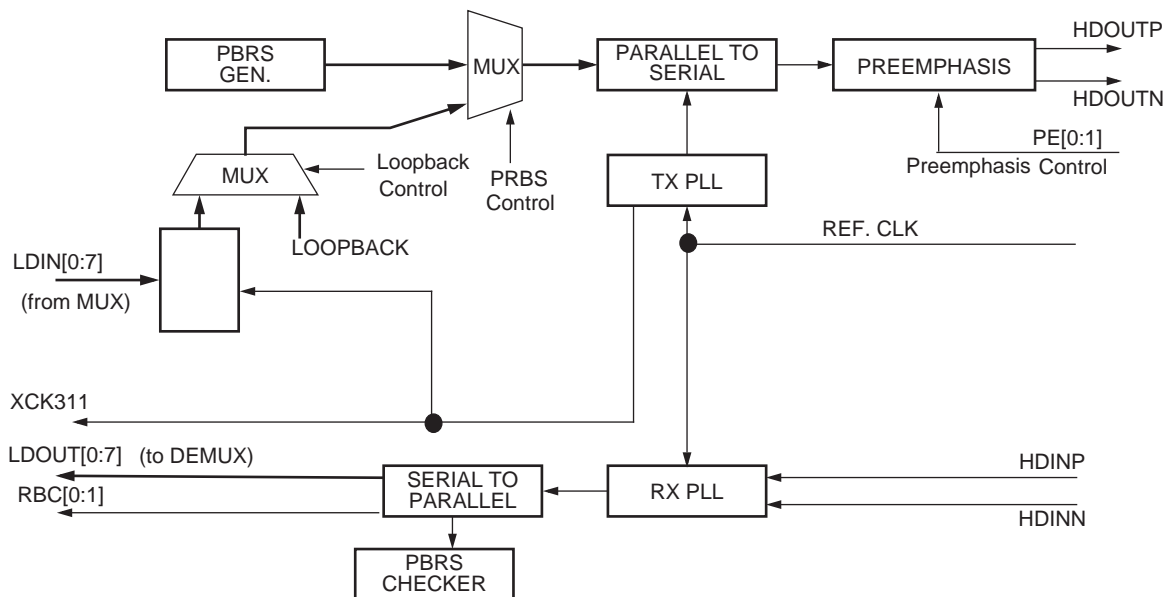


Figure 8 shows a block diagram of a single channel of the SERDES block. The transmitter section accepts either scrambled or un-scrambled data at the parallel input port. It also accepts the low-speed reference clock at the REF-CLK input and uses this clock to synthesize the internal high-speed serial bit clock. The serialized data are available at the differential CML output terminated in 86 Ω to drive either an optical transmitter, coaxial media or a circuit board/backplane.

Figure 8. SERDES Functional Block Diagram for One Channel



1725

The receiver section receives high-speed serial data at its differential CML input port. These data are fed to the clock recovery section which generates a recovered clock and retimes the data. This means that the receive clocks are asynchronous between channels. The retimed data are deserialized and presented as an 8-bit parallel data on the output port. Two-phase receive byte clocks are available synchronous with the parallel words.

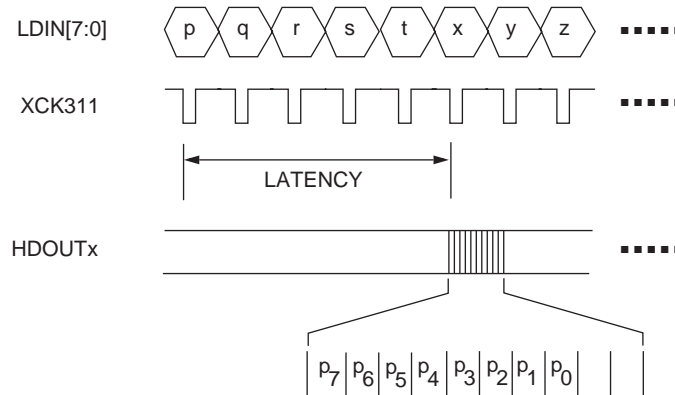
A fractional band-gap voltage generator is included on the design. An external resistor ( $3.32\text{ k}\Omega \pm 1\%$ ), connected between the pins REXT and VSSREXT generates the bias currents within the chip. This resistor should be able to handle at least  $300\text{ }\mu\text{A}$ .

**SERDES Transmit Path**

The transmitter section accepts 8-bit data at the parallel input port from the MUX block in the internal transmit core. Data is transmitted with bit 7 sent first and bit 0 sent last. It also accepts the low-speed reference clock at the REFCLK input and uses this clock to synthesize the internal high-speed serial bit clock.

The embedded core uses the XCK311 SERDES output as the reference clock. The frequency of this clock will depend on the half-rate/full-rate control bit in the SERDES; and the frequency of the REFCLK ports. Transmit timing is shown in Figure 9.

**Figure 9. SERDES Transmit Path Timing**



The serialized data are available at the differential CML outputs to drive either an optical transmitter, coaxial media, or circuit board/backplane. The transmitter’s CML output buffer is terminated on-chip by 86Ω to optimize the data eye as well as to reduce the number of discrete components required. The differential output swing reaches a maximum of 1.2 V<sub>PP</sub> in the normal amplitude mode. A half amplitude mode can be selected via configuration register bit HAMP\_xx. Half amplitude mode can be used to reduce power dissipation when the transmission medium has minimal attenuation or for testing of the integrity (loss) of the physical medium

A programmable preemphasis circuit is provided to boost the high frequencies in the transmit data signal to maximize the data eye opening at the far-end receiver. Preemphasis is particularly useful when the data are transmitted over backplanes or low-quality coax cables which have a frequency-dependent amplitude loss. For example, for FR4 material at 2.5 GHz, the attenuation compared to the 1.0 GHz value is about 3 dB. The attenuation is a result of skin effect loss of the PCB conductor and the dielectric loss of the PCB substrate. This attenuation causes intersymbol interference which results in the closing of the data eye at the receiver.

Since this effect is predictable for a given type of PCB material, it is possible to compensate for this effect in two ways - transmitter preemphasis and receiver equalization. Each of these techniques boosts the high frequency components of the signal but transmit preemphasis is preferred due to the ease of implementation and the better power utilization. It also gives a better signal-to-noise ratio because receiver equalization amplifies both the signal and the noise at the receiver

Applying too much preemphasis when it is not required, for example when driving a short backplane path, will also degrade the data eye opening at the receiver. In the ORSO82G5, the degree of transmit preemphasis can be programmed per channel with two control register bits as shown in Table 4. The high-pass transfer function of the preemphasis circuit is given by the following equation, where the value of a is shown in Table 4.

$$H(z) = (1 - az^{-1})$$

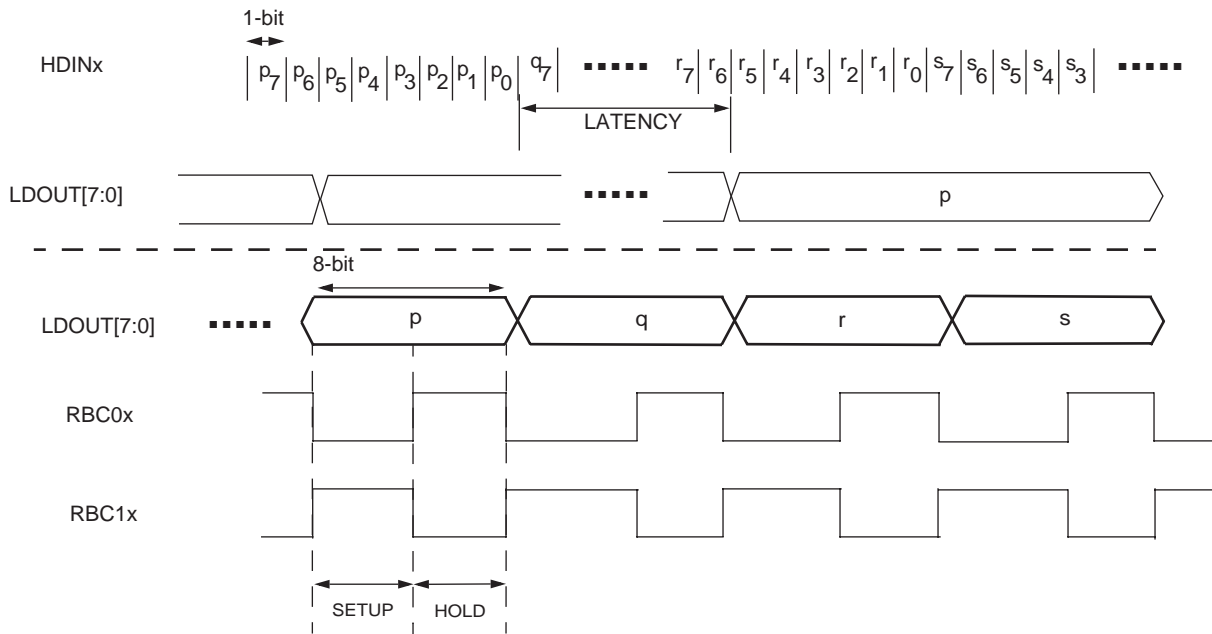
**Table 4. Preemphasis Settings**

PE1	PE0	Amount of Preemphasis (a)
0	0	0% (No Preemphasis)
0	1	12.5%
1	0	12.5%
1	1	25%

**SERDES Receive Path**

The receiver section receives high-speed serial data at its differential CML input port. These data are fed to the clock recovery section which generates a recovered clock and retimes the data. Each SERDES receive channel has its own PLL and this means that the receive clocks are asynchronous between channels. This also enables each receive channel to either operate in half-rate or full-rate mode. The retimed data are deserialized and presented as a 8-bit parallel data on the output port. Two-phase receive byte clocks (RBC0 and RBC1) are available synchronous with the parallel words. RBC0 has its rising edge aligned to the center of the receive byte. RBC1 has its falling edge aligned to the center of the receive byte. The 8-bit data (LDOUT) as shown in Figure 10, changes on a single clock edge (rising edge of RBC0 or falling edge of RBC1).

**Figure 10. SERDES Receive Path Timing**



The receive PLL has two modes of operation as follows: lock to reference and lock to data with retiming. When setup to lock to data and no data or invalid data are present on the HDINP and HDINN pins, the receive VCO will not lock to data and its frequency can drift outside of the nominal  $\pm 100$  ppm range. Under this condition, the receive PLL will lock to REFCLK for a fixed time interval and then will attempt to lock to receive data. The process of



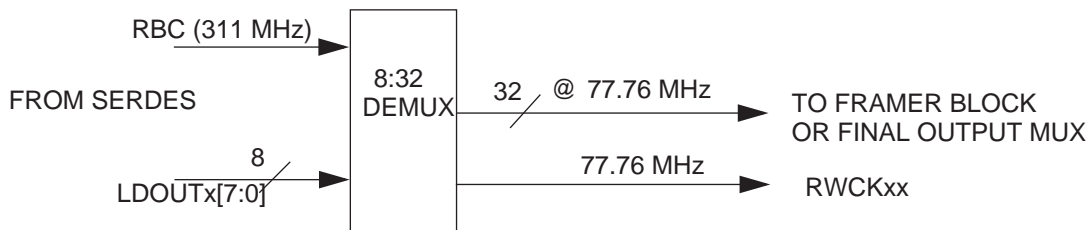
attempting to lock to data, then locking to clock will repeat until valid input data exists. There is also a control register bit per channel (LCKREFN) to force the receive PLL to always lock to the reference clock. The default mode is lock to reference.

The SERDES receives MSB first and LSB last. Hence LDOUTx[7] is the bit that is received first and LDOUTx[0] is the bit that is received last.

**8:32 DEMUX**

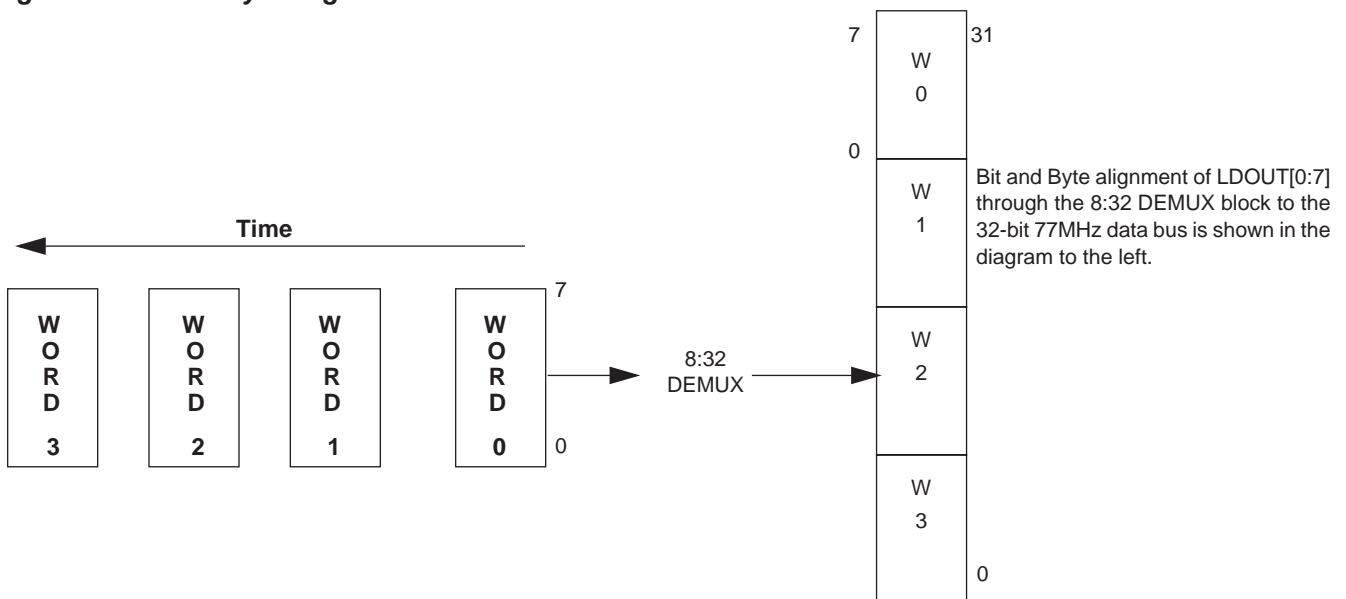
The SERDES provides an eight bit data bus, and a clock, RBC0, which has a rising edge which occurs in the center of the valid data region. The DEMUX block will create one 32-bit data word from the single 8-bit bus. The DEMUX block will also provide a 77.76 MHz clock (divide-by-4 clock of RBC0) called RWCKxx to the rest of the logic blocks such as the framer, descrambler, cell extractor and FIFOs.

**Figure 11. 8:32 DEMUX Block**



In the DEMUX block, LDOUTx[7:0] is demultiplexed to a 32-bit data bus synchronous to the derived 77.76 MHz clock generated by dividing the 311.04 MHz clock by four. The 77.76 MHz clock is used by the remaining embedded blocks, as well as being fed to the FPGA. Receive data from the DEMUX block is unframed. Parallel data can be sent directly to the FPGA logic (SERDES only mode) or to the framer block for processing. Bit and byte alignment for the DEMUX block is shown in Figure 12.

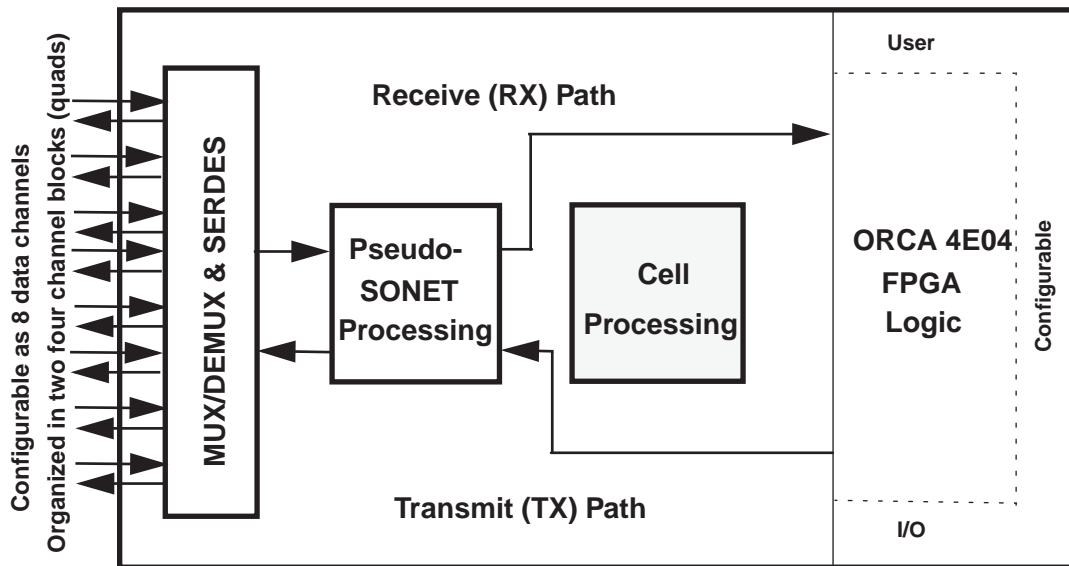
**Figure 12. Bit and Byte Alignment for DEMUX Block**



## SONET Mode Operation - Detailed Description

The following sections describe the data processing performed in the SONET logic blocks. The basic data flows in the SONET Mode are shown in Figure 13. At a top level, the descriptions are separated into processing in the transmit path (FPGA to serial link) and processing in the receive path (serial link to FPGA). In general, the descriptions in the next sections are written to describe SONET mode operation, although some of the "SONET logic blocks" are also used in cell mode operation. The various processing options are selected by setting bits in control registers and status information is written to status registers. Both types of registers can be written and/or read from the System Bus and the MicroProcessor Interface (MPI). Memory maps and descriptions for the registers are given in Table 14 through Table 22.

Figure 13. Basic Data Flows - SONET Mode



In the SONET mode, the transmit block receives 32-bit wide data from the FPGA (DINxx) on each of its 8 channels along with a frame pulse (DINxx\_FP) per channel and a transmit clock (TSYCLKxx). See Table 9 for details of the core/FPGA signal assignments. Typically this will represent a STS-48 stream on each link. The data are first passed through a TOH block which will generate all the timing pulses that are required to isolate individual overhead bytes (e.g., A1, A2, B1, D1-D3 etc.). The timing pulse generation will be based on the system frame pulse (DINxx\_FP) received from the FPGA logic, hence it is recommended that, for multi-channel alignment at the receiver, the frame pulses for the channels in an alignment group be synchronized. The data are then scrambled before being sent to the SERDES.

In the receive direction, each of the two SONET quads process four receiving channels which can be treated as one STS-192 stream or as four independent STS-48 streams. The clocks for the received data are recovered from the received serial data streams and a serial to parallel conversion is performed on the data. The frame format of the incoming data are reconstructed (optional). The data then is descrambled using the standard STS-12 polynomial and a B1 parity error check is performed on the data from the previously transmitted frame.

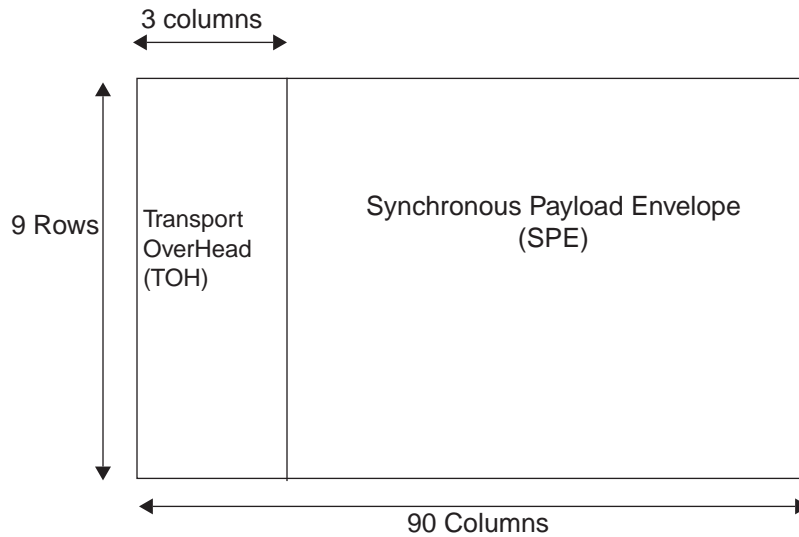
In the SONET mode, the incoming serial data streams can be aligned by groups of two, four or eight. When doing multichannel alignment of two or more data streams, the receiver can handle the data streams with frame offsets of up to 18 bytes due to timing skews between cards and along backplane traces or other transmission medium. For multichannel alignment capability to operate properly, it should be noted that while the skew between channels can be very large, they must operate at the exact same frequency (0 ppm frequency deviation), thus requiring that the transmitters sourcing the data being received be driven by the same clock source. It is highly recommended that the frame pulses, DINxx\_FP, for all links in an alignment group be synchronized. (See Table 9 for details of the core/FPGA transmit direction signal assignments.)

The received data streams are processed and passed to the FPGA logic as 32-bit words. In SONET mode, DOUTxx[31:0] carries the 32 bit data from the alignment FIFO of the respective channel and an accompanying frame pulse, DOUTxx\_FP. Other core/FPGA carries miscellaneous information such as OOF, BIPERR, and SPE indicator. (See Table 10 for details of the core/FPGA receive direction signal assignments.)

**Basic SONET Frame Format**

An STS-N frame can be broadly divided into the Transport OverHead (TOH) and the Synchronous Payload Envelope (SPE) areas. The TOH comprises of bytes that are used for framing, error detection and various other functions. The start of the SPE can begin at any point in a SONET frame. The start of the SPE is determined using the pointer bytes located in the TOH. The basic STS-1 frame is shown in Figure 14. Higher rate STS\_N signals are created by byte interleaving N STS-1 signals. Some TOH bytes have slightly different functions in STS-N frames than in the basic STS-1 frame. The ORSO82G5 offers two options for processing the TOH bytes, as discussed in a later section.

**Figure 14. STS-1 Frame Format**



**Supported Data Formats**

The ORSO82G5 in the SONET mode supports the following formats:

- Single OC-48 on each of the 8 channels at a bit rate of 2.488 Gbits/s.
- OC-192 received in quad OC-48 format at a combined rate of 9.952 Gbits/s.

Figure 15 reveals the byte-ordering of the individual STS-48 streams. STS-192 is supported but it must be received in the quad STS-48 format as shown in Figure 16. Each OC-48 stream is composed of byte-interleaved OC-1 data as described in GR-253 standard. Note that the SPE data is not touched by the core.

**Figure 15. Byte Ordering of Input/Output Interface in STS-48 Mode**

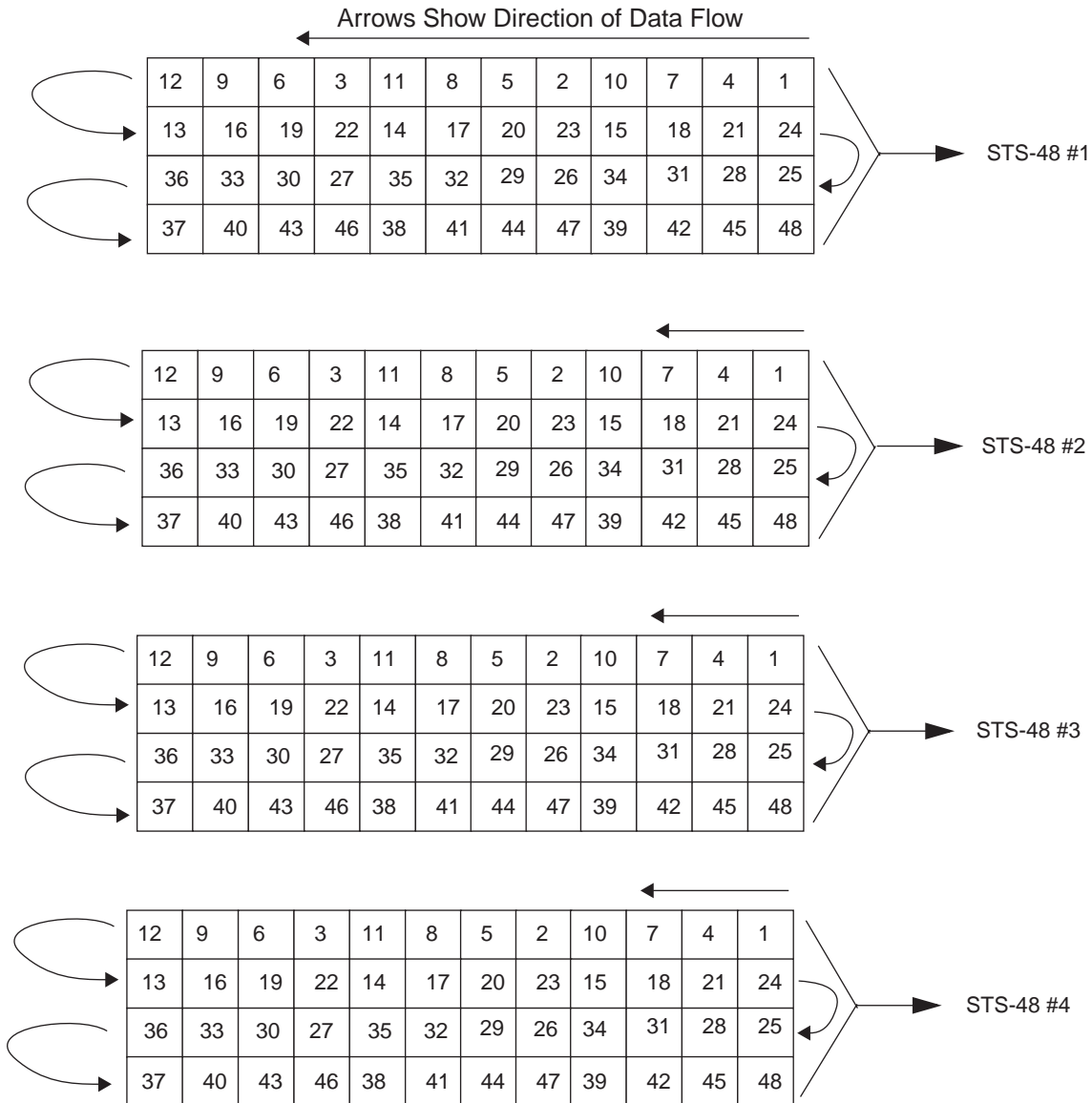
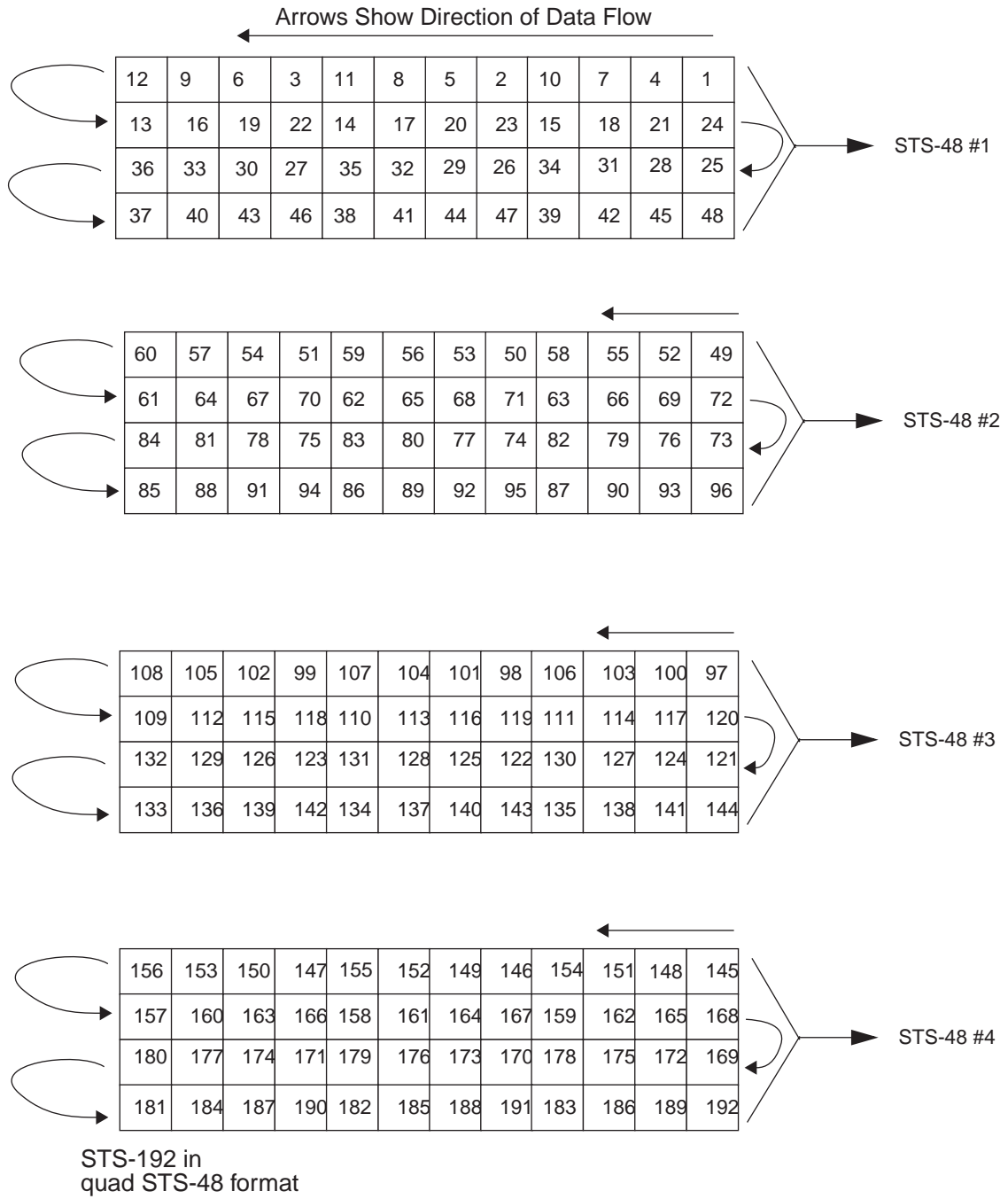


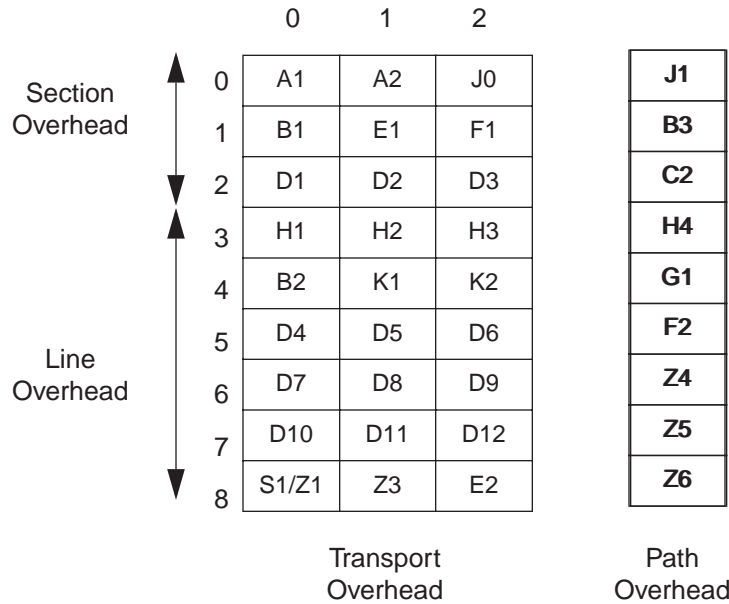
Figure 16. . Byte Ordering of Input/Output Interface in STS-192 (Quad STS-48) Mode



**SONET TOH Byte Definitions**

The Transport OverHead bytes of the SONET frame can be used for in-band configuration, service, and management since it is carried along the same channel as data. In the ORSO8850 in-band signaling can be efficiently utilized, since the total cost of overhead is only 3.3%. The overhead bytes in an STS-1 header are shown in Table 17 (The path overhead bytes are in the SPE.)

**Figure 17. SONET Overhead Bytes**



When used in true SONET applications, most TOH bytes will be generated in the FPGA logic or by an external device. When used to transfer non-SONET data, all of the overhead bytes except those set by the embedded core can be ignored using AUTO\_TOH mode. The TOH bytes have the following functions in true SONET applications.

**Section Overhead Bytes:**

- A1, A2 - These bytes are used for framing and to mark the beginning of a SONET frame. A1 has the value 0x F6 and A2 has the value 0x28.
- J0 - Section Trace Message - This byte carries the section trace message. The message is interpreted to verify connectivity to a particular node in the network.
- B1 - Section Bit Interleaved Parity (BIP-8) byte - This byte carries the parity information which is used to check for transmission errors in a section. The computed parity value is transmitted in the next frame in the B1 position. It is defined only for the first STS-1 of a STS-N signal. The other bytes have a default value of 0x00.
- E1 - Section orderwire byte - This byte carries local orderwire information, which provides for a 64 Kbps voice channel between two Section Termination Equipment (STE) devices.
- F1 - Section user channel byte - This byte provides a 64 Kbits/s user channel which can be used in a proprietary fashion.
- D1, D2, D3 - Section Data Communications Channel (SDCC) bytes - These bytes provide a 192 Kbits/s channel for transmission of information across STEs. This information could be for control and configuration, status monitoring, alarms, network administration data etc.

**Line Overhead Bytes:**

- H1, H2 - STS Payload Pointers (H1 and H2) - These bytes are used to locate the start of the SPE in a SONET frame. These two bytes contain the offset value, in bytes, between the pointer bytes and the start of the SPE. These bytes are used for all the STS-1 signals contained in an STS-N signal to indicate the individual starting positions of the SPEs. They bytes also contain justification indications, concatenation indications and path alarm indication (AIS-P).
- H3 - Pointer Action Byte (H3) - This byte is used during frequency justifications. When a negative justification is performed, one extra payload byte is inserted into the SONET frame. The H3 byte is used to hold this extra byte and is hence called the pointer action byte. When justification is not being performed, this byte contains a default value of 0x00.
- B2 - Line Bit-Interleaved Parity code (BIP-8) byte - This byte carries the parity information which is used to check for transmission errors in a line. This is a even parity computed over all the bytes of the frame, except section overhead bytes, before scrambling. The computed parity value is transmitted in the next frame in the B2 position. This byte is defined for all the STS-1 signals in an STS-N signal.
- K1, K2 - Automatic Protection Switching (APS channel) bytes - These bytes carry the APS information. They are used for implementing automatic protection switching and for transmitting the line Alarm Indication Signal (AIS-L) and the Remote Defect Indication (RDI-L) signal.
- D4 to D12 - Line Data Communications Channel (DCC) bytes - These bytes provide a 576 Kbps channel for transmission of information.
- S1 - Synchronization Status - This byte carries the synchronization status of the network element. It is located in the first STS-1 of an STS-N. Bits 5 through 8 of this byte carry the synchronization status.
- Z1 - Growth - This byte is located in the second through Nth STS-1s of an STS-N and are allocated for future growth. An STS-1 signal does not contain a Z1 byte.
- M0 - STS-1 REI-L - This byte is defined only for STS-1 signals and is used to convey the Line Remote Error Indication (REI-L). The REI-L is the count of the number of B2 parity errors detected by an LTE and is transmitted to its peer LTE as feedback information. Bits 5 through 8 of this byte are used for this function.
- E2 - Orderwire byte - This byte carries for line orderwire information.

**SONET Mode Transmit Path**

The transmit block performs the following functions in SONET mode:

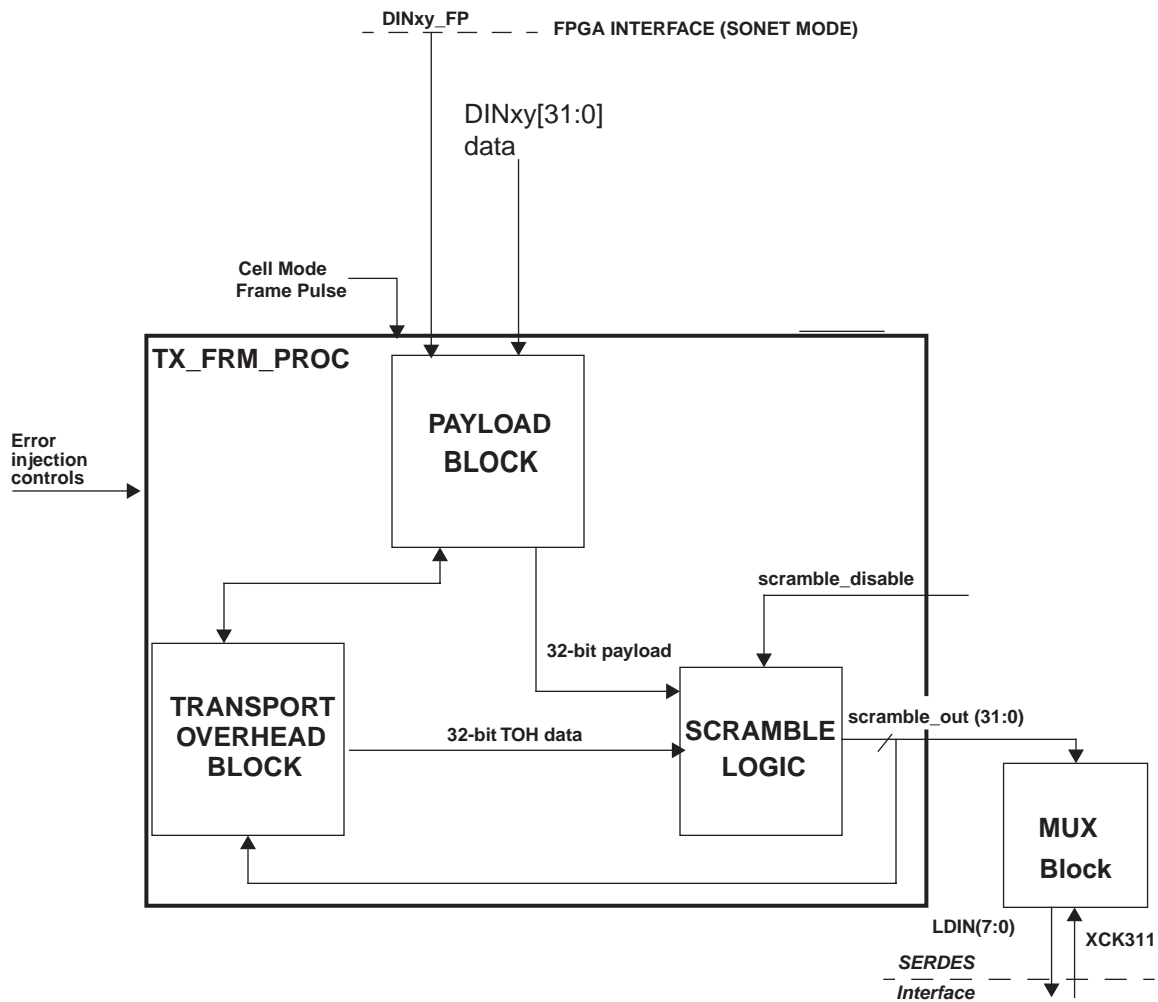
- A1 and A2 insertion and optional corruption
- BIP-8 parity calculation, B1 byte insertion and optional corruption. (B1 byte is inverted.)
- Performs RDI insertion (K2 byte is set to “0000 0110”).
- Scrambling of outgoing data with optional scrambler disabling.

In either STS-192 or STS-48 mode, each link operates at an STS-48 rate.

**TX Frame Processor**

The Tx\_Frame\_Processor (TFP) block is the primary data processing block in the both SONET mode and cell mode. It organizes the cell data into a SONET frame before sending it to the SERDES. The TFP is on the TSY-CLKxx clock domain (77.76 MHz). In SONET mode, the 32-bit data comes from the FPGA interface. (In cell mode the data comes from the cell processing block as described in the cell mode section) The TFP block contains three major sub-blocks: payload block, TOH block and scrambler block. The interfaces for the TFP block are shown in Figure 18.

**Figure 18. TX Frame Processor (TFP) Block Diagram**





**Payload Sub-block**

The Payload sub block is activated by the cell mode frame pulse (cell mode) or DINxx\_FP from FPGA (SONET mode). A pulse on this signal indicates the start of a frame.

In SONET mode, only two types of data bytes are in each frame:

- TOH bytes
- SPE data bytes

There are N x 3 (N = 48) bytes of TOH per row and there are a total of 9 rows in a SONET frame. The rest of the bytes in each row are SPE data bytes in SONET mode.

**TOH Sub-block**

This block is responsible for the 144 (48 x 3) bytes of TOH at the beginning of each row of the transport frame. The TOH bytes may be transmitted transparently from the FPGA logic or may be inserted by the TOH block (AUTO\_TOH mode). The TOH data is transferred across the FPGA/core interface as 32-bit words, hence 36 clock cycles (12 x 3) are needed to transfer a TOH row. TOH insertion is controlled by software register bits as shown in the Register Map tables. The TOH values inserted in Auto TOH mode are shown in Table 5. The values are for all STS-1s in the STS-48 frame unless noted otherwise.

**Table 5. Inserted TOH Values (all 0x)**

A1 = F6	A2 = 28	J0 = STS-1 ID, every 4th. STS-1 J0 = 00, other STS-1s
B1 = calculated, 1st. STS-1 B1 = 00, other STS-1s	E1 = 00	F1 = link number, 1st. STS-1 F1 = 00, other STS-1s
D1 = 00	D = 002	D3 = 00
H1 = 62, 1st. STS-1 H1 = 93 other STS-1s	H2 = 0A, 1st. STS-1 H2 = FF other STS-1s	H3 = 00
B2 = 00	K1 = 00	K2 = 06 for RDI, K2 = 00 otherwise, 1st. STS-1 K2 = 00, other STS-1s
D4 = 00	D5 = 00	D6 = 00
D7 = 00	D8 = 00	D9 = 00
D10 = 00	D11 = 00	D12 = 00
S1 = 00	M1 = 00	E2 = 00

The TOH block can perform A1/A2 corruption by inverting the A1/A2 bytes and also can forces B1 errors by inverting the B1 byte. A RDI can be injected by forcing the K2 byte to “00000110”. In SONET mode, all TOH bytes can be transparently sent from the FPGA as an option. Error and RDI insertion are controlled by software register bits as shown in the Register Map tables.

**Scramble Sub-block**

The scrambler scrambles the incoming 32-bit data using the standard SONET polynomial  $1 + x^6 + x^7$ . The scrambler can be disabled by a software register bit.

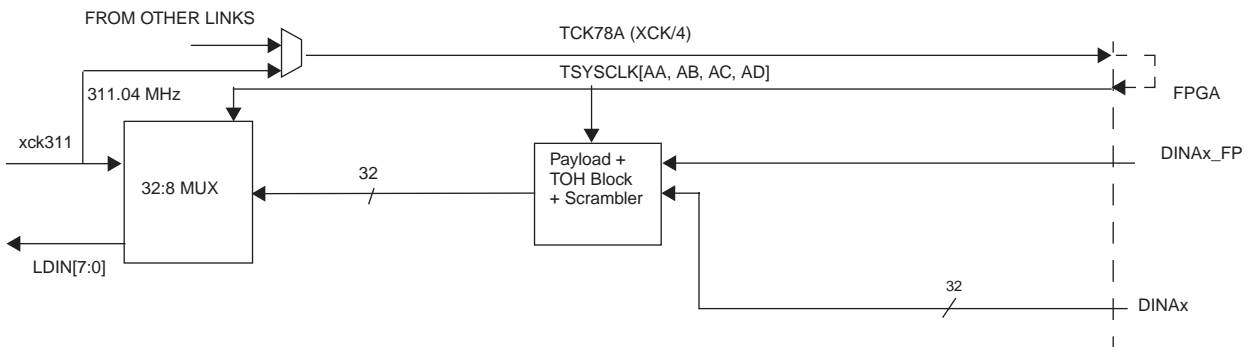
**32:8 MUX**

The MUX block is responsible for converting 32 bits of data at 77.76 MHz to 8 bits of data at 311.04 MHz. It will contain a small elastic store for clock domain transfer between the write clock from the FPGA to the divide-by-4 clock from the SERDES output clock (XCK311). It is recommended to use the clocking scheme shown to guarantee that this elastic store will not be overrun. The 32:8 MUX is also responsible for producing the divide-by-4 clock from the SERDES output clock (XCK311) which is 311.04 MHz at a line rate of 2.488 Gbits/s.

**SONET Mode Transmit Timing**

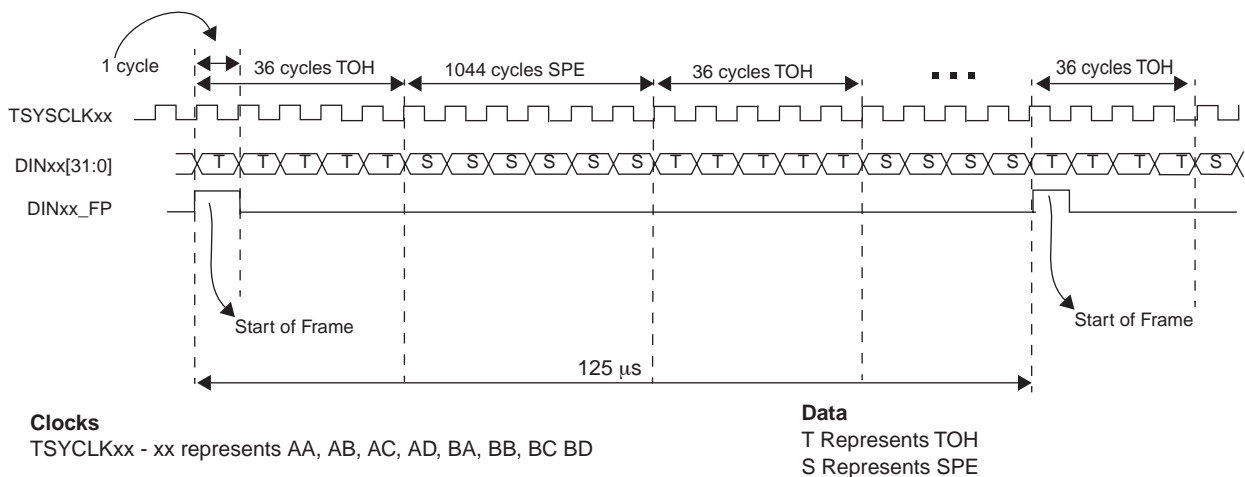
Figure 19 shows the transmit clocks and the recommended clocking scheme. TCK78[A,B] should be used to source TSYCLK[A,B] respectively.

**Figure 19. Transmit Clocking Diagram for Quad A in SONET Mode**



- When operating in SONET mode, the entire SONET frame is sent by the user. Optionally the TOH bytes can be overwritten by the transmit block (AUTO\_TOH) before sending to scrambler and SERDES block.
- Each SONET frame is 125 μs given a 155.52MHz reference clock.
- The frame starts with 36 clock cycles (77.76 MHz) of TOH followed by 1044 clock cycles of SPE, followed by 36 clock cycles of TOH, 1044 cycles of SPE etc. for all 9 rows.

**Figure 20. Transmit SONET Mode**



**Clocks**  
TSYCLKxx - xx represents AA, AB, AC, AD, BA, BB, BC, BD

**Data**  
T Represents TOH  
S Represents SPE

---

## SONET Mode Receive Path

The receiver block receives a byte from the SERDES blocks for each of the eight channels. The byte arrives at the receiver block at 311.04 MHz. This data are not frame-aligned or word aligned. The data are first passed through a divide-by-4 DEMUX which produces a 32-bit word at 77.76 MHz. Data from the DEMUX is then passed through a framer which aligns and frames the data. Data are processed based on cell mode or SONET mode.

In the SONET mode, the descrambled data are sent to an alignment FIFO that performs lane-to-lane deskew and aligns data within an alignment group to the RSYCLK[A1, A2, B1, B2] clock domain. Both the write and read clocks to the alignment FIFO should be at exactly the same frequency (0 ppm difference), i.e., from a common clock source. Later in this datasheet, Figures 25, 26, and 27 show the recommended clocking scheme to adhere to this requirement. In addition, supervisory features such as BIP error check, OOF check, RDI monitoring and AIS-L insertion during OOF are also implemented. All the supervisory features are controlled through programmable register bits.

### Framer

The frame and byte phase of the bits within the 32-bit word from the DEMUX is random. For each of the STS-48 channels, the framer outputs 4 bytes (32-bits) that are frame-aligned and a frame pulse that is one clock-wide. The transition from A1 bytes to A2 bytes should happen on a 32-bit boundary. If any two consecutive bytes of the 4-byte-aligned word match the A1-A2 pattern (0xF628 in standard SONET framing), the byte-aligned word will be byte rotated to achieve frame alignment.

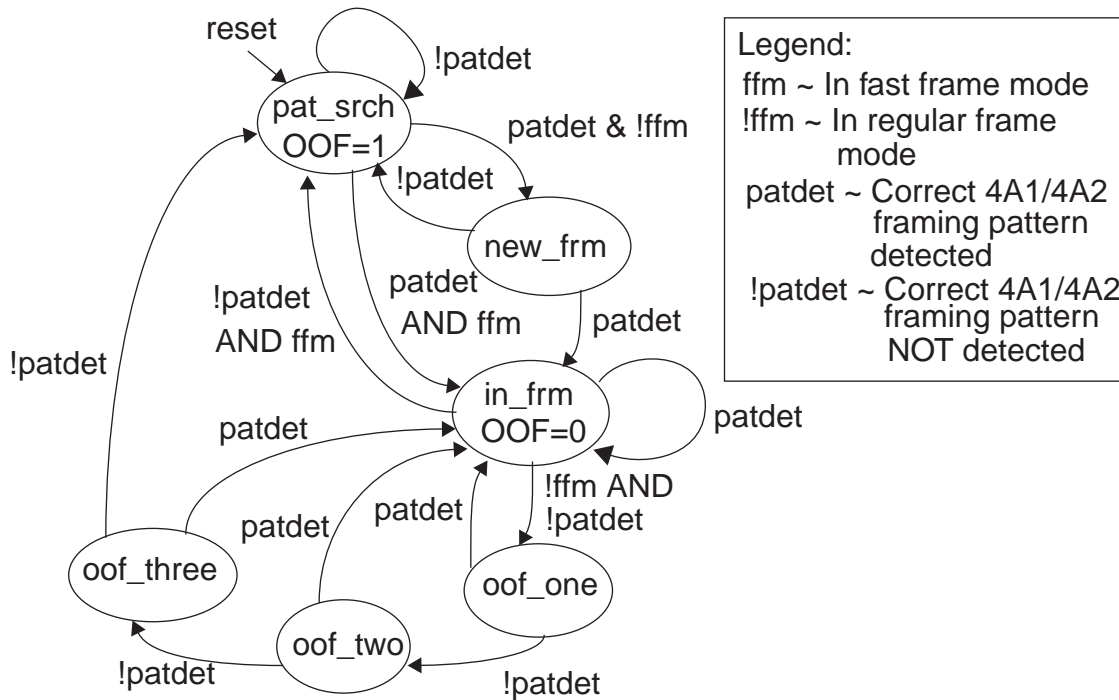
### Framer State Machine (FSM)

The framer FSM is responsible for detecting the in-frame and Out-Of-Frame status of the incoming data and sends out alarms (interrupts) on Out-Of-Frame (OOF). The framer is a pseudo-SONET framer in the sense that it does not support LOF or SEF detect-alarm signal as specified in the GR-253 standard.

The framer has a fast frame mode where a single good framing pattern can cause the framer state machine to go the "in\_frm" state and a single bad frame can cause the state machine to declare "OOF" (See Fig. 21). The fast frame mode can be set by the software register bit, FFRM\_EN\_xx.

The FSM is a four byte framer and searches for the framing pattern based on the 32-bit words. Accordingly, the framing pattern is four A1 bytes (F6,F6,F6,F6) followed by four A2 bytes (28,28,28,28).

Figure 21. Framer State Machine



The framer FSM comes out of reset in the “OOF” state with the OOF alarm set. The framer goes in frame if it finds 2 consecutive frames with the desired framing bytes and goes out of frame if it finds 4 consecutive frames with at least one framing bit error in each frame. Frame timing is also synchronized based on the STS-48 row and column counters. This corresponds to SONET specification that it will take two consecutive valid framing patterns to frame to an incoming signal. Outside the “OOF” state, the OOF alarm output is low.

**Section (B1) BIP-8 Calculator**

The section BIP-8 B1 byte in a given STS-N frame contains the scrambled BIP value for all scrambled bytes of the previous frame. Except for the A1,A2 and J0 section overhead bytes, all bytes in a frame are scrambled. The Section (B1) BIP-8 is calculated as the even parity of all bits in the current STS-48 frame. This value is compared to the Section Overhead B1 byte of the next frame. A B1 parity error flag is also generated as a software alarm bit.

**Descrambler**

The data from the framer is descrambled using the SONET/SDH standard generator polynomial  $1 + x^6 + x^7$ . The descrambling is performed in parallel on each 32-bit word per channel, synchronized to the frame pulse and can be disabled through the software register bit.

**RDI (Remote Defect Indicator) Monitor**

The line RDI (RDI-L) is monitored through bits 2-0 of the K2 byte. Within the 32-bit descrambled data, a pattern of “110” on bits 26-24 will indicate a RDI-L status. RDI-L must be detected in two consecutive frames before an rdi alarm register bit is set. If fast\_frame\_mode is enabled, then the RDI alarm register bit will be set if RDI-L is detected in one frame.

**Receive FIFO**

Clock domain transfers and multi-link de-skew are one of the most critical parts of this device. The main clock domain transfer for the datapath is handled by the receive FIFO. For each link, there are two FIFOs. A 24 x 33 FIFO is used in SONET mode.

The use of the FIFO is controlled by configuration bits.

- Data can be sent from the descrambler directly to the FPGA bypassing the alignment FIFO. Data from each

---

channel will have an associated clock (RWCKxx at 77.76 MHz). Each channel will also provide a FP and SPE indicator along with the data. Descrambling can be inhibited through the DSCR\_INH\_xx control bit.

- Data can be sent directly from the 8:32 DEMUX block to the FPGA bypassing the alignment FIFO and SONET framer and descrambler. Data from each channel will have an associated clock. No SPE or FP indicator is provided with the data.

#### Receive FIFO in SONET mode

The receive FIFO used in SONET mode will allow for an inter-link skew of about 300 ns ( $24 \times 32 = 768$  bits, 400 ps per bit gives 307 ns). The FIFO is written at 77.76 MHz and read at 77.76 MHz. Once frame synchronization has occurred, the write control logic will cause data to be written to the memory. The write control block is required to insure that the word containing the first A1 byte is written to the same location (address 0) in the FIFO. The synchronization algorithm issues a sync pulse and sync error signals to the read control block based on the alignment option chosen (align all 8, align by 4 or align by pairs). This sync pulse will coordinate the reading of the FIFOs.

The read control logic synchronizes the reading of the FIFO for the streams that are to be aligned. The block begins reading with the FIFO sync sub block signals that all of the applicable A1s with the appropriate margin have been written to the FIFO. All of the read blocks to be synchronized begin reading at the same time and same location in the memory (address 0). The block also takes the difference between the write and read address to indicate the relative skews between the links. If this difference exceeds a certain limit (programmable), then an alarm (alignment overflow) is provided to the register interface.

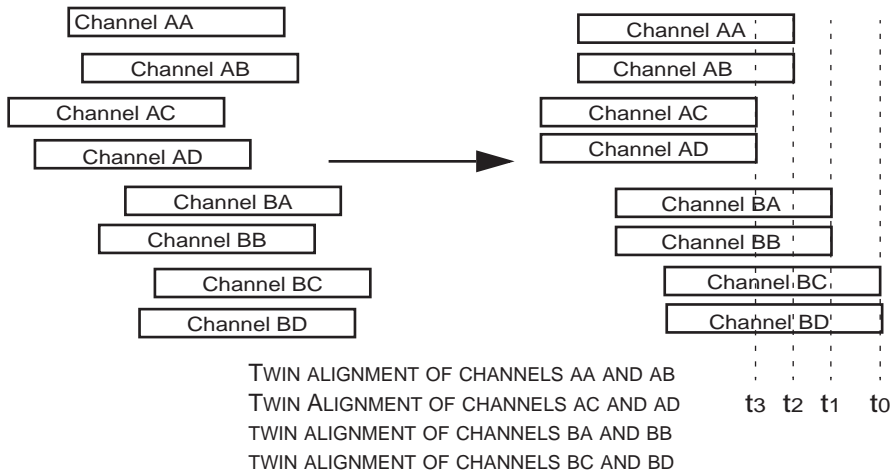
#### Multi-channel Alignment in SONET Mode

The alignment FIFO allows the transfer of all data to a common clock. The FIFO sync block allows the system to be configured to allow the frame alignment of multiple slightly varying data streams. This optional alignment ensures that matching SERDES streams will arrive at the FPGA end in perfect data sync. It is important to note that for all aligned channels in a group, the SERDES transmitters on the other side of the high-speed link must all be transmitting data at exactly the same frequency (0 ppm difference), i.e., using a common clock source.

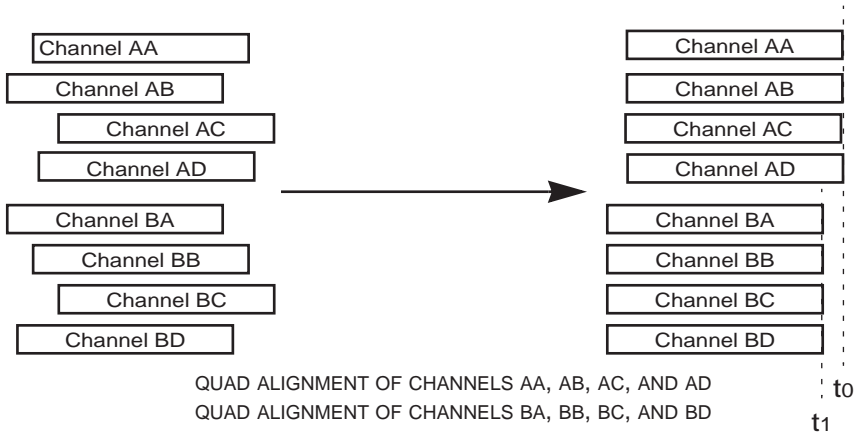
The ORSO82G5 has a total of 8 channels (4 per quad SERDES). The incoming data of these channels can be synchronized in several ways, or they can be independent of one other. Two channels within a SERDES can be aligned together; channel A and B and/or channel C and D can form a pair as shown in Figure 22. Alternately, all four channels in a SERDES quad can be aligned together to form a communication channel with a bandwidth of 10 Gbits/s as shown in Figure 23.

Finally, the alignment can be extended across SERDES to align all 8 channels in ORSO82G5 as shown in Figure 23. Individual channels within an alignment group can be disabled (i.e., powered-down) without disrupting other channels. Note that the SERDES channel that is powered down can not be the source of the RSYCLKxx that is clocking the read side of the alignment FIFO. When a disabled channel becomes active as part of an alignment group, the group may need to be re-aligned. Then the whole group needs to be resynched. This would only need to occur if the transmitting frame pulse for the new link is different from the rest of the group.

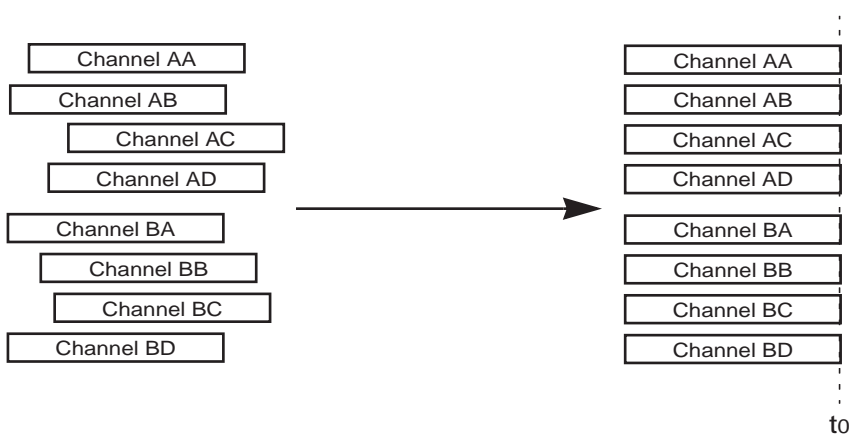
**Figure 22. Twin Channel Alignment**



**Figure 23. Alignment of SERDES Quads A and B**



**Figure 24. Alignment of all Eight SERDES Channels.**



Each channel is provided with a 24 word x 33-bit FIFO. The FIFO can perform two tasks: (1) to change the clock domain from receive clock to a clock from the FPGA side, and (2) to align the receive data over 2, 4, or 8 channels. This FIFO allows a timing budget of 307 ns that can be allocated to skew between the data lanes and for transfer to

the common clock. The input to the FIFO consists of 32-bit data and a frame pulse that indicates the start of a frame (or A1A2 framing bytes). This frame pulse is used to synchronize multiple channels within an alignment group.

If a channel is not in any alignment group, the FIFO control logic will set the FIFO-write-address to the beginning of the FIFO, and will set the FIFO-read-address to the middle of the FIFO at the first assertion of frame pulse after reset or after the resync command.

The RX\_FIFO\_MIN register bits can be used to control the threshold for minimum unused buffer space in the alignment FIFOs between read and write pointers before OVFL status is flagged. The synchronization algorithm consists of a down counter which starts to count down by 1 from its initial value of 18 (decimal) when a frame pulse from any channel within an alignment group has been received. The OOS alarm indicates the FIFO is out-of sync and the channel skew exceeds that which can be handled by the FIFO. Once the frame pulse for all channels within the alignment group have been received, the count is decremented by 2 until 0 is reached. Data are then read from the FIFOs and output to the SPE generator before being sent to the FPGA.

For every alignment group, there is an OVFL and OOS status register bit. The OOS bit is flagged when the down counter in the synchronization algorithm has reached a value of 0 and frame pulse from all channels within an alignment group have not been received. The OVFL bit is flagged when the read address at the time of receiving a frame pulse, is less than the minimum threshold set by RX\_FIFO\_MIN. In the memory map section OOS is referred to as SYNC[2,4]\_[A1,A2,B1,B2]\_OOS, SYNC8\_OOS. OVFL is referred to as SYNC[2,4]\_[A1,A2,B1,B2]\_OVFL, SYNC8\_OVFL.

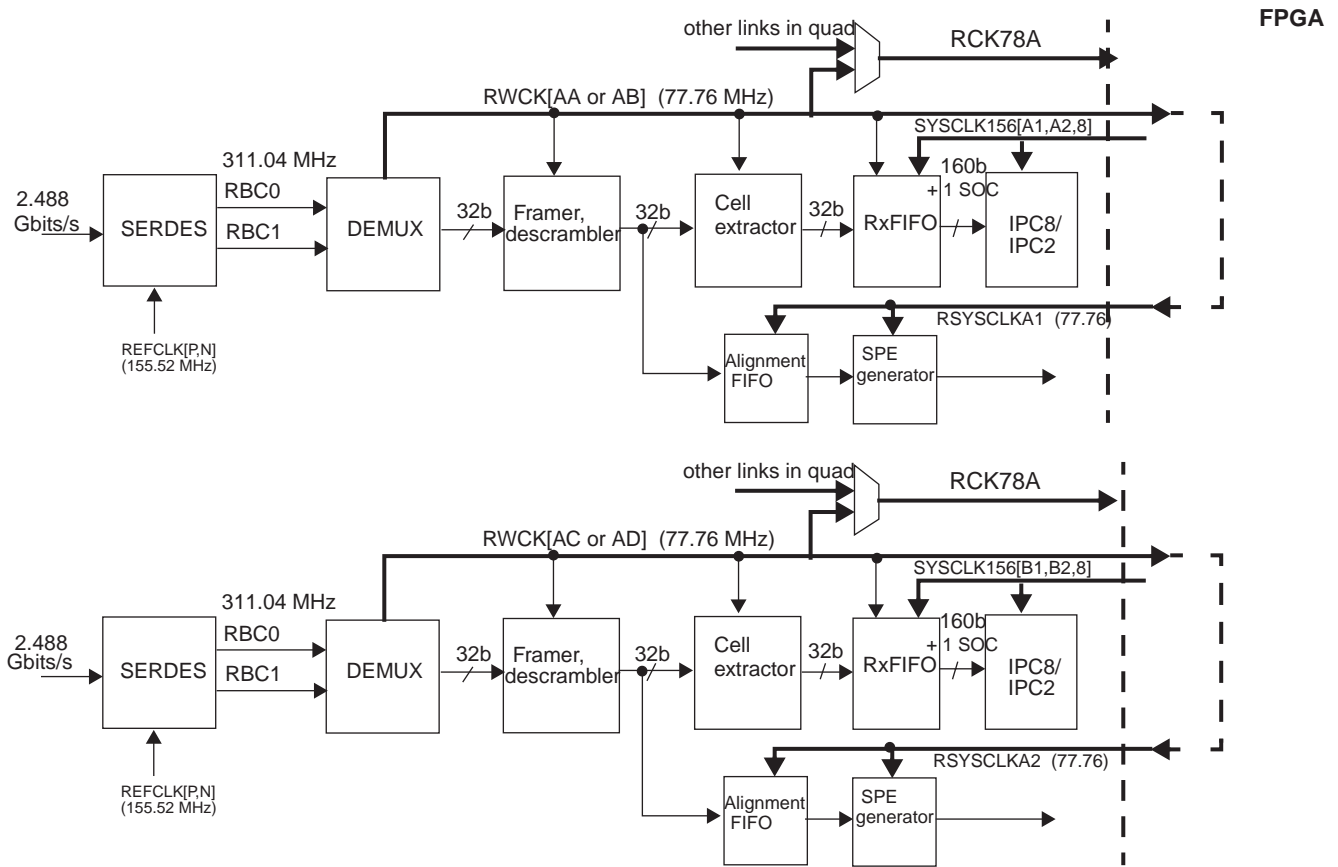
#### **Receive Clocking for Multi-channel Alignment**

There are a total of nine clocks for the receive path, from FPGA to the core. The four used in SONET mode are RSYSCCLKA1 and RSYSCCLKA2 (both for quad A), and RSYSCCLKB1 and RSYSCCLKB2 (both for quad B). The following diagrams show the recommended clock distribution approaches for SONET mode multi-channel alignment modes. Cell mode alignment is discussed in the section describing the Input Port Controller (IPC).)

**SONET Mode Twin Alignment**

Figure 25 describes the clocking scheme for twin alignment. In twin alignment, the valid channel pairs are AA,AB and AC,AD in quad A and BA,BB and BC,BD in quad B. The figure provides the clocking scheme for quad A as an example. RSYSCLKA1 should be sourced from RWCKAA or RWCKAB. RSYSCLKA2 should be sourced from RWCKAC or RWCKAD. This clocking approach provides the required 0 ppm clock frequency matching for each pair and provides flexibility in applications where the two pairs are received from asynchronous sources.

**Figure 25. Receive Clocking Diagram for Twin Alignment in Quad A**

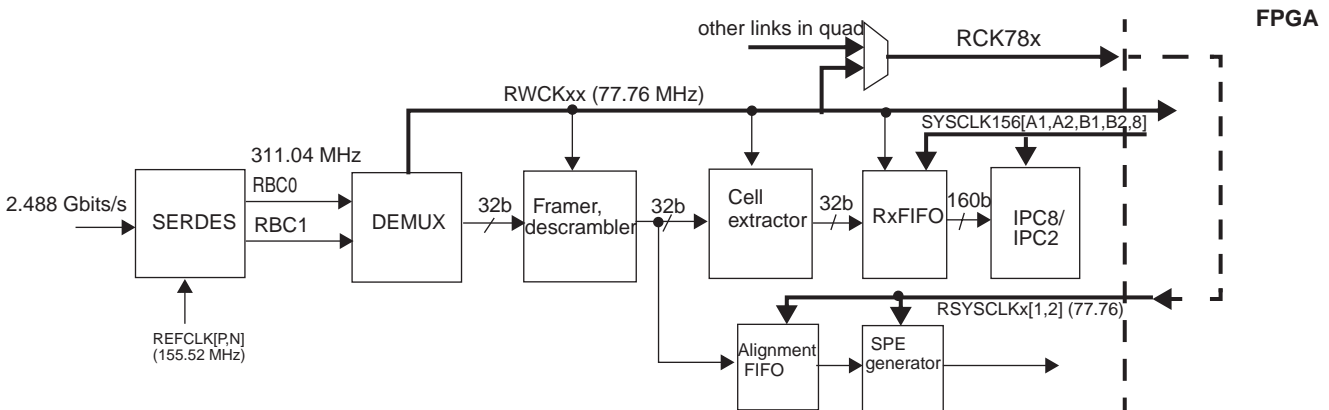




**SONET Mode Quad Alignment**

Figure 26 describes the clocks and recommended clocking for quad alignment in the SONET mode. For quad alignment, the low speed portion for each quad should be sourced by a single clock. As the figure shows, for quad A, RSYCLKA1 and RSYCLKA2 should be sourced by RCK78A. For quad B, RSYCLKB1 and RSYCLKB2 should be sourced by RCLK78B. RCLK78A can be sourced by any channel in quad A and RCLK78B can be sourced by any channel in quad B.

**Figure 26. Receive Clocking Diagram for a Quad Channel (Quad Alignment)**



**Total clocks from core to FPGA**

RWCKxx - 4 (one for each channel)  
 where xx = [AA,AB..BD]  
 RCK78x - 1 for each quad where x = [A,B]

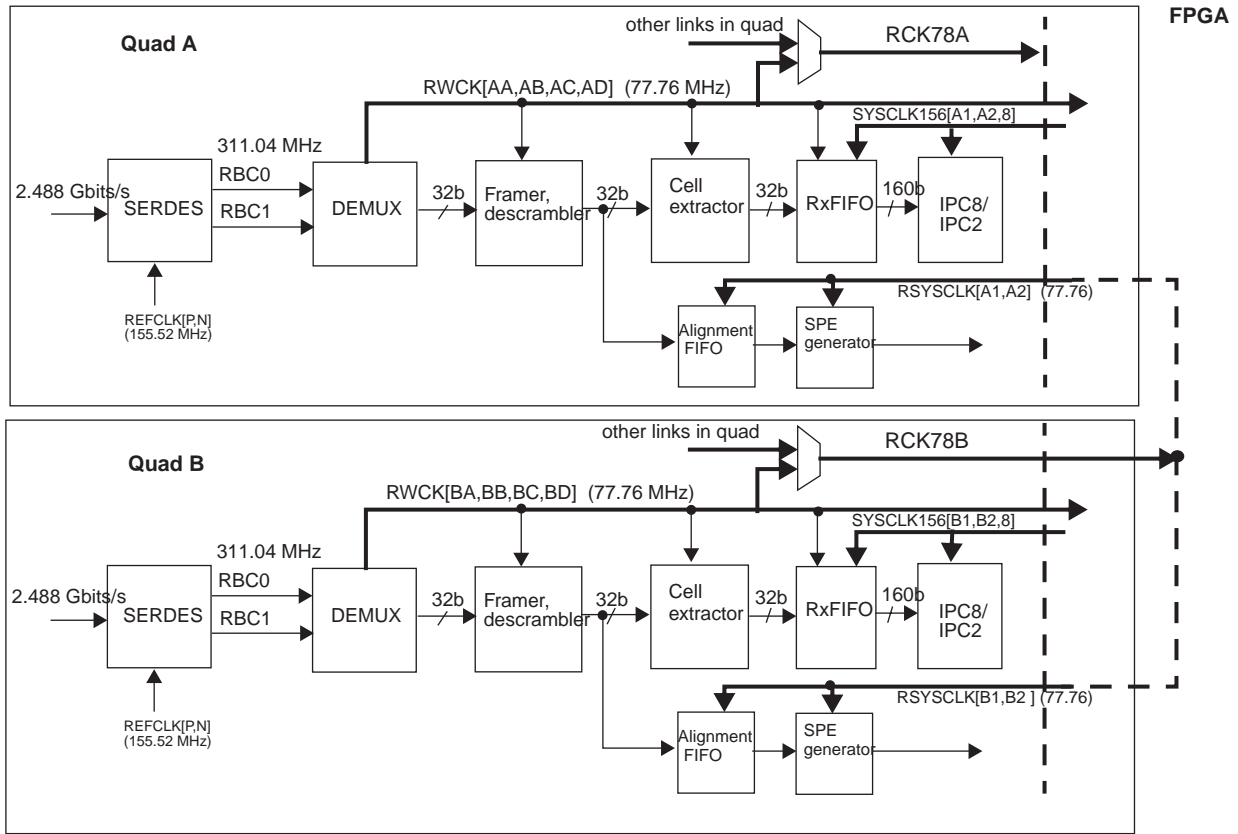
**Total clocks from FPGA to core**

RSYCLKx[1,2] - 77.76MHz clock (2 for each quad) (x is A or B)

**SONET Mode Octal Alignment**

Figure 27 shows the clocking scheme for eight-channel alignment. In this application, all four clocks RSY\_SCLKA1, RSY\_SCLKA2, RSY\_SCLKB1 and RSY\_SCLKB2 should be sourced from a common clock. Either RCK78A or RCK78B can be used as a common clock source. The figure shows RCK78B being used as the clock source.

**Figure 27. Receive Clocking for Octal Alignment**



**SPE Generator**

The SPE generator is used to indicate the payload and overhead portions of a SONET frame. It is present in the SONET data path only. The SPE generator generates row, column and sts counters based on the frame pulse received from the (24 x 33) alignment FIFO or from the descrambler if alignment FIFOs are bypassed. It also retimes the 32-bit data in order to align it with the SPE indicator. The SPE generator will also detect negative or positive pointer justification (if justification is enabled) by looking at the ID bits in the H1 and H2 bytes and adjust the SPE indicator for the STS-1 frame being justified as follows:

- During positive pointer justification, the SPE will be low during H3 byte and the SPE byte following it.
- During negative pointer justification, the SPE will be high during H3 byte.
- During no justification, the SPE will be low during H3 byte.

This block only detects the incoming pointer bytes for SPE generation. This capability can be enabled by software control. By default, the SPE generator will ignore any pointer justification. This block has no capability of any pointer processing, pointer checking or pointer mover operation and ignores "New Data" indications from the SONET specification.

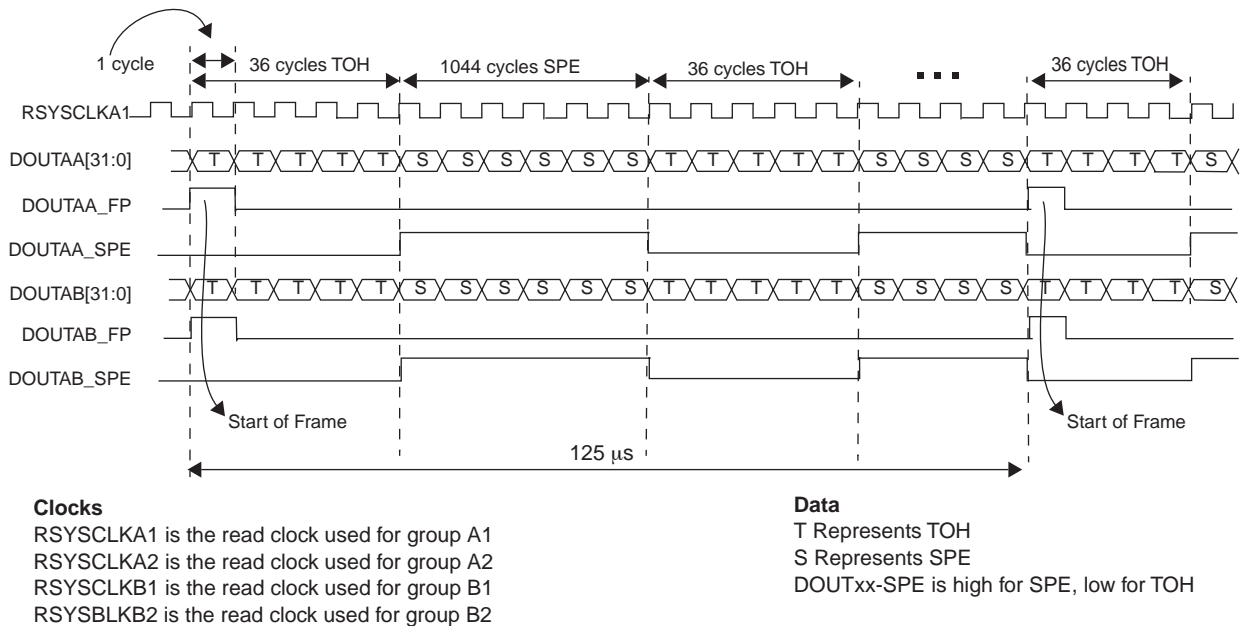
**SONET Mode Receive Timing**

This section contains timing diagrams for major interfaces of this block to the FPGA logic when SONET frames are to be transferred.

- When operating in SONET mode, the entire SONET frame is sent to the FPGA. In multi-channel alignment mode(s), data from all channels within an alignment group are aligned to the A1A2 framing bytes.
- Each SONET frame is 125µs. The frame starts with 36 clock cycles (77.76Mhz) of TOH followed by 1044 clock cycles of SPE, followed by 36 clock cycles of TOH, 1044 cycles of SPE.
- The DOUTxx\_SPE signal indicates TOH or SPE in the data (low for TOH, high for SPE)
- Twin pairs are AA, AB (group A1), AC, AD (group A2), BA, BB (group B1) and BC, BD (group B2)

Figure 28 shows the SONET twin alignment mode timing. The frame pulse and SPE indicators are show for each of the two channels (AA, AB) in twin alignment.

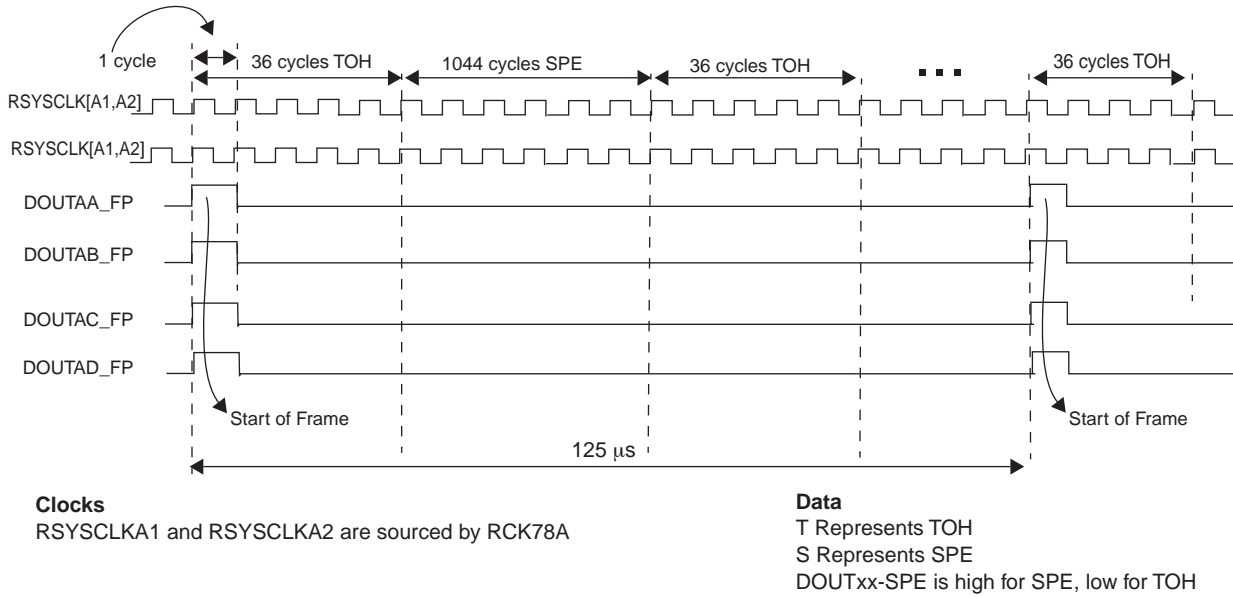
**Figure 28. Receive SONET Mode—Twin Align Mode**



- Only frame pulse (DOUTxx\_FP) and clocks are shown for understanding of quad alignment.
- Timing of data and SPE indicators are the same as shown for twin alignment.
- Twin groups are A1 - AA and AB, A2 - AC and AD, B1 - BA, and BB and B2 - BC and BD

Figure 29 shows the SONET quad alignment mode.

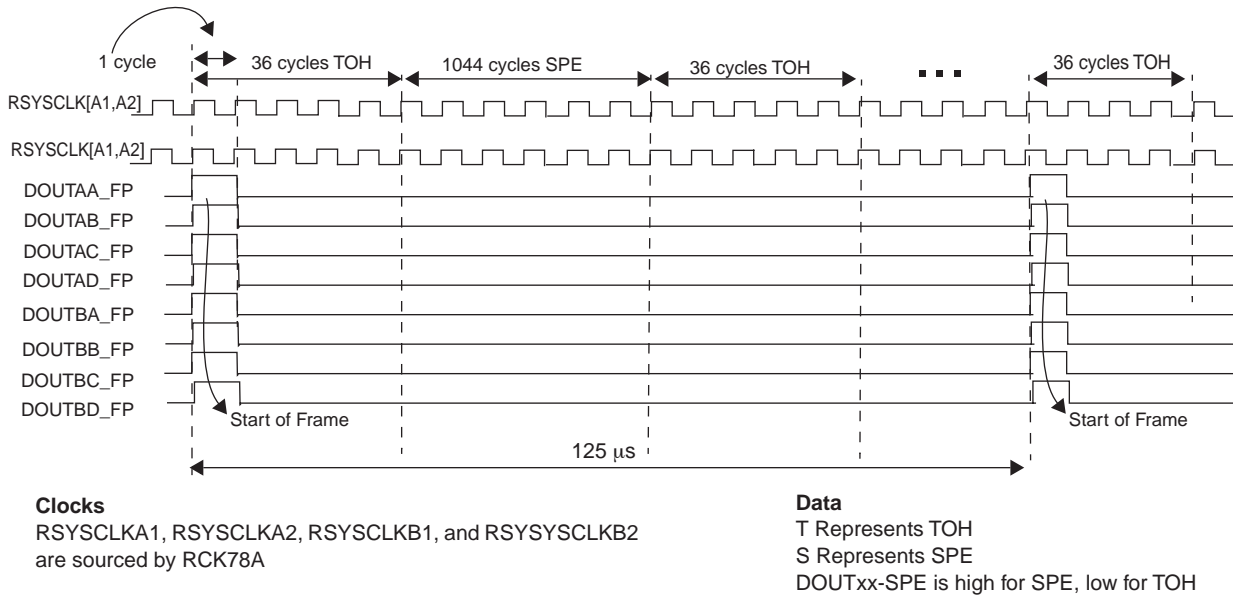
**Figure 29. Receive SONET Mode—Quad Align Mode**



- Only frame pulse (DOUTxx\_FP) and clocks are shown for understanding of quad alignment.
- Timing of data and SPE indicators are the same as shown for twin alignment.
- Quad groups are Group A - AA,AB,AC,AD and Group B - BA,BB,BC,BD

Figure 30 shows the octal alignment mode.

**Figure 30. Receive SONET Mode—Octal Alignment Mode**



**Alignment Mode Setup Procedures**

The control register bits for alignment FIFO in ORSO82G5 are described below.

**Table 6. Multichannel Alignment Modes**

Register Bits FMPU_SYNMODE_xx	Mode
00	No multichannel alignment. (default)
01	Twin channel alignment.
10	Quad channel alignment.
11	Eight channel alignment.

Note: Where xx is one of A[A:D] and B[A:D].

**To align all eight channels:**

- FMPU\_SYNMODE\_A[A:D] = 11
- FMPU\_SYNMODE\_B[A:D] = 11

**To align twin channels in SERDES A:**

- FMPU\_SYNMODE\_A[A:D] = 01

**To align four channels in SERDES A:**

- FMPU\_SYNMODE\_AB = 10
- FMPU\_SYNMODE\_AD = 10

Similar alignment can be defined for SERDES B. To enable/disable synchronization signal of individual channel within a multi-channel alignment group:

- FMPU\_STR\_EN\_xx = 1 enabled
- FMPU\_STR\_EN\_xx = 0 disabled  
where xx is one of A[A:D] and B[A:D].

**To re-synchronize a multi-channel alignment group**

Set the following bit to zero, and then set it to 1 since it is a rising edge sensitive bit.

- FMPU\_RESYNC8 for eight channel A[A:D] and B[A:D]
- FMPU\_RESYNC4A for quad channel A[A:D]
- FMPU\_RESYNC2A1 for twin channel A[A:B]
- FMPU\_RESYNC2A2 for twin channel A[C:D]
- FMPU\_RESYNC4B for quad channel B[A:D]

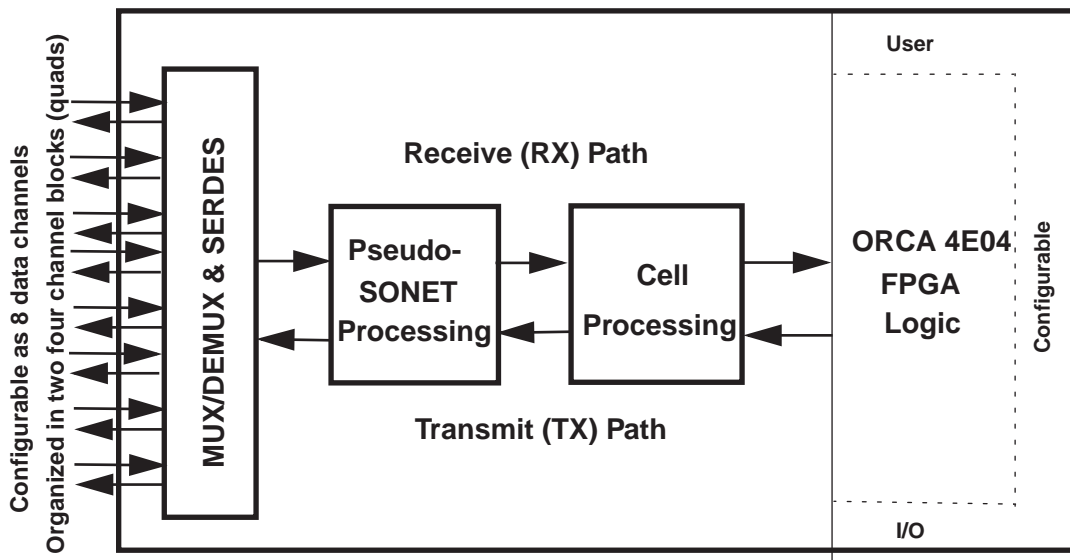
Re-synchronization will cause the multi-channel alignment group to perform a new synchronization procedure. This would need to occur if a link is removed from an alignment group and then paced back into service as part of the alignment group. This is not required at power up if the alignment mode is set before the channels receive the first frame pulse.

**Cell Mode Detailed Description**

A common application for the ORSO82G5 is to provide a bridge between a port card and a cell-based switch fabric. In cell mode, the data in the Synchronous Payload Envelope (SPE) of the SONET frames is further formatted into fixed-length cells by the ORSO82G5. The cell contents will typically be unique to specific port card and switch devices. The ORSO82G5 supports this application with a "cell mode" of operation

The basic data flows in cell mode are shown in Table 31. Data to be transmitted is received from the FPGA logic (see Table 9 for details of the core/FPGA signal assignments which differ significantly from the SERDES only and SONET modes), inserted into the SPE of the SONET frame, scrambled and transmitted from the SERDES block. In cell mode, multiple SERDES links are used to achieve desired bandwidth. Both two-link and eight-link cell modes are supported. For such interfaces, data are cell-stripped in a round-robin fashion across multiple links by the transmitter.

**Figure 31. Basic Data Flows - Cell Mode**



In the receive direction, the framed data are received from the SERDES block, descrambled and are passed into a cell extractor which extracts individual cells from the payload portion of the SONET frame. The cells are then passed through a FIFO that performs lane-to-lane deskew and a clock domain transfer. The clock domain transfer is handled automatically using idle cell insertion and deletion.

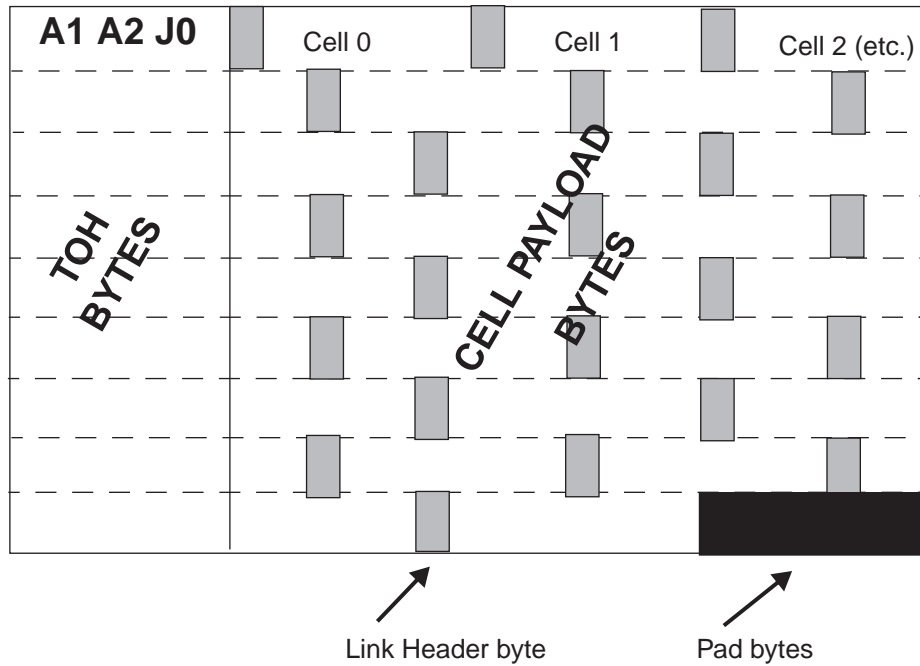
The cells are passed into either the eight-link Input Port Controller IPC8 block or to one of the four two-link IPC2\_[A:B][1:2] block(s) which reassemble the cells back into a single cell stream (destriping) which is sent to the FPGA logic. (See Table 10 for details of the core/FPGA signal assignments. As with the transmit path, the cell mode assignments differ significantly from those for the SERDES only and SONET modes).

SERDES and SONET processing has been described in previous sections and only features unique to the cell mode will be discussed in the following sections. The cell format will be discussed first, followed by a description of the transmit path, which will include either a two-link or an eight-link Output Port Controller (OPC) block, and a description of the receive path, including the two-link or eight-link Input Port Controller (IPC) blocks.

**Cell Formats**

Cells are arranged within a SONET (STS-48c) frame as shown in Figure 32. A SONET STS-48c frame has 4176 (87 x 48) columns of SPE and 9 rows that gives a total of 37,584 bytes. In this implementation, data in a SPE is limited to fixed size cells. Though four cell sizes are supported, only one cell size can be used at a time.

Figure 32. Cell Mapping in SONET Frame



The cells are placed in a SONET frame such that the first cell starts at the first SPE column of the first row (145th column. 144 columns are taken up by TOH). Subsequent cells are placed contiguously, skipping the Transport OverHead (TOH) columns when appropriate.

The ORSO82G5 supports four cell sizes with varying payloads. The total cell size sent across the backplane is the combined size of the user cell payload (cell header and data), user BIP and the Link Header byte. Table 7 indicates the cell sizes supported by the ORSO82G5 within a STS-48c frame. Only one cell size can be used at a time.

Table 7. ORSO82G5 Supported Cell Sizes

Total Cell Size (across B/P)	User Cell Payload Size (header/data)	Link Header Byte	User BIP Field	Cells/Frame	Number of Pad Bytes Per SPE
77	75	1	1	488	8
81	79	1	1	464	0
85	83	1	1	442	14
93	91	1	1	404	12

NOTE: To calculate the number of Cells per SPE:  

$$[(87\text{Rows}/\text{STS}-1 * 9\text{Columns}/\text{STS}-1)\text{octets}/(\text{STS}-1 \text{ SPE})] * 48 \text{ STS}-1 = 37584 \text{ octets}$$

$$337584 / \text{TOTAL CELL SIZE} = \# \text{ of cells per SPE.}$$

A cell cannot span multiple SONET frames. This implies that there may be some cell sizes for which some bytes will be unused at the end of a SPE. These are called pad bytes.

Each cell is preceded by a Link Header byte as shown in Figure 33. Table 8. defines the format of the Link Header. The Link Header byte is useful for cell delineation when cell data are striped across multiple links. The link header is inserted automatically in the transmit direction by the IPC block. The Link Header is checked in the receive direction and removed by the OPC before the cell is sent across the core/FPGA interface.

Figure 33. Link Header Byte

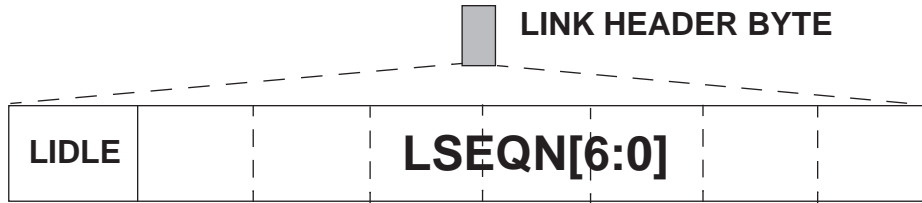


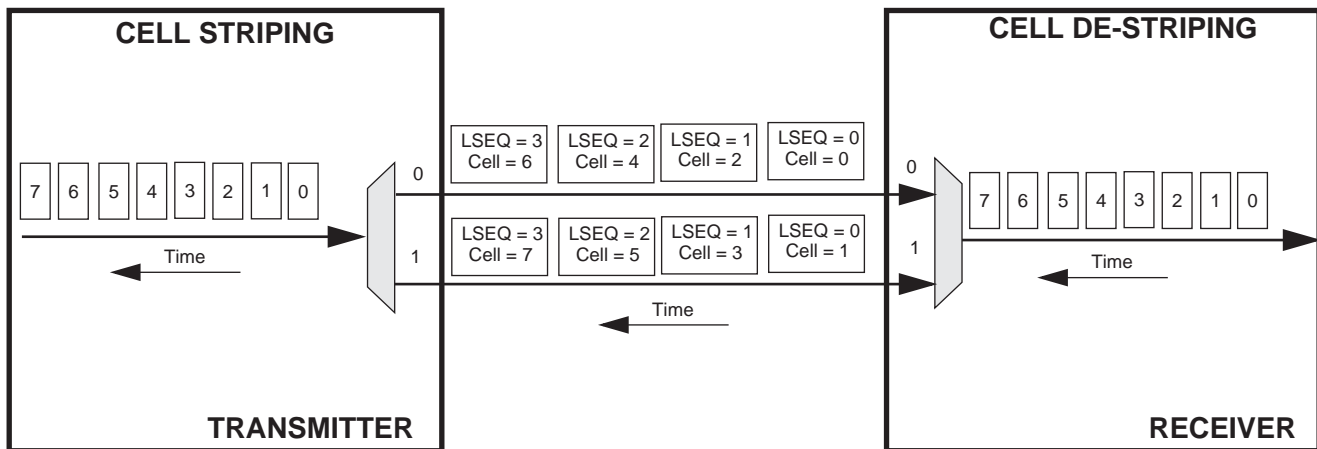
Table 8. Link Header Format

Location	Field/Description
7	Idle - Idle Cell Indicator 0: User Cell (contains valid data) 1: Idle Cell (no data in the cell payload)
6:0	LSEQ - Link Sequence Number. This value is used when aligning cells from multiple links when doing link group multiplexing.

There is no way to tell where a cell starts unless one counts the cells from the beginning of the SPE. That means, there is no way to regain lost cell delineation other than wait for the next SONET frame.

In Cell Mode, multiple SERDES links are used to achieve desired bandwidth. Data are cell-stripped in a round-robin fashion across two or eight links by the transmitter and then re-assembled back into a single cell stream (destriping) by the receiver. This is shown in Figure 34.

Figure 34. Multi-Link Interface - Two-Link Example



To assist with cell delineation, each link transmitter assigns sequence numbers to cells (LSEQN[6:0] bits in the Link Header byte) before sending them out on the link. Each link increments its sequence numbers independently as shown in Figure 34. All links reset their sequence number generator at the beginning of a SONET frame (All links are synchronized to the start of a frame).

On the receiving side, each receiver uses the sequence numbers to verify the correct cell delineation. Since the links were synchronized to the start of the SONET frame, all links will have cells with the same sequence number available at the same time (although deskew needs to happen to properly align the cells). This allows the receiver to correctly reconstruct the original cell stream.

If an unexpected sequence number is received, the receiver does not use the received value as the basis for the next expected sequence number. Rather the old expected value is incremented by one, forming the new expected value. An error flag is sent to the software register interface and the cell will be marked with an error. For example, assume that the receiver expected to receive a cell with sequence number 27, but received one with sequence



---

number 37. The cell will be marked with an error. The receiver then expects to receive a cell with sequence number 28.

### Cell Mode Transmit Path

In the transmit path in cell mode, the transmit logic creates a SONET-like transport frame for the data, adds the required Transport OverHead bytes (cell mode automatically uses AUTO\_TOH mode) and retimes the cell data from the FPGA interface rate of 156 MHz to the framer rate of 77.76 MHz. The data are then sent to the SONET logic blocks and SERDES. The Payload sub-block of the SONET logic operates somewhat differently than in SONET mode, however.

### Output Port Controller

In cell mode the Output Port Controller (OPC) is the block responsible for directing traffic for the egress cell traffic flow. These are the blocks that provides the interface between FPGA and the Transmit FIFO. There is one eight-link controller (OPC8) and four two-link controllers (OPC2s). The user provides the 160-bit (OPC8) or 40-bit (OPC2) cell data, and a cell\_vaild strobe to the OPC. No Link Header byte is sent with the cell data. The OPC provides the following functions:

- Accepts cell payload from the FPGA logic and assembles legal output cells from these.
- Schedules, manages and performs writes of cell data into TX FIFOs in the transmit framer blocks of all the eight-links or up to 4 pairs of two-links.
- Provides backpressure information to the FPGA to stop writes to the TXFIFO if the FIFO is not ready to accept data.

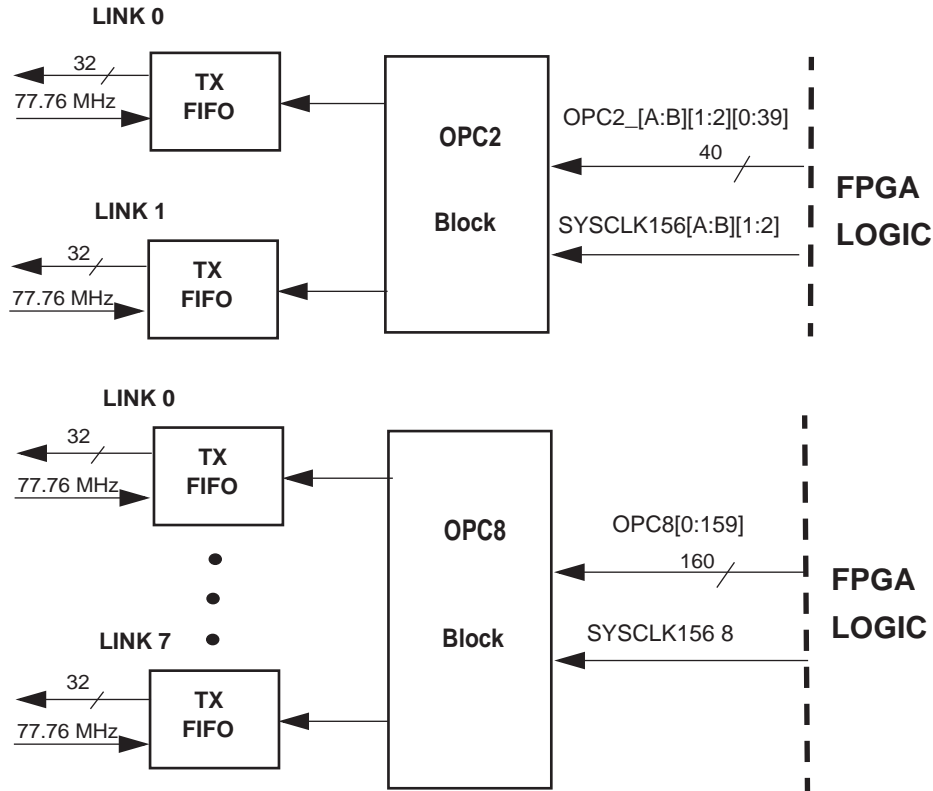
The 5 OPC blocks operate as follows:

- OPC8 to stripe cells across eight-links
- OPC2\_A1 to service links AA,AB
- OPC2\_A2 to service links AC,AD
- OPC2\_B1 to service links BA,BB
- OPC2\_B2 to service links BC,BD

When operating with some links in the two-link cell mode, links not in an alignment group can optionally be operated in a SONET mode.

The OPC2 blocks run at 39 MHz. The input from FPGA is a 40-bit bus at 156 MHz (4 times the required frequency). A 1:4 deMUX block provides a 160-bit bus at 39 MHz

Figure 35. OPC2 and OPC8 Block Diagrams



**TX FIFO Block**

In cell mode, the TxFIFO block contains the write state machine and FIFO memory. The FIFO is used to retime the cell data from the FPGA interface at a rate of 156 MHz to the framer rate of 77.76 MHz.

The FIFO memory is implemented as a 64 x 34 FIFO. The FIFO receives data as 33-bit words with the start-of-cell as the MSB of each word. Thus each cell occupies a maximum size of 23 words. For each link, the memory must be capable of holding at least 2 cells or 46 words (one for write and one for read). To ensure extra space, this capacity has been increased to 64 words. In addition, an extra bit has been reserved to store the link idle cell indicator bit which is used for indicating the internally generated idle cells. Thus each word in a cell is 34 bits. Data are written to the memory on the 156 MHz clock domain and read on the 77.76 MHz clock domain by the Tx\_Frame\_Processor block. The OPC requires no response from the TxFIFO for writing data. The TxFIFO is guaranteed by design to not overflow or underrun.

**TX Frame Processor**

The Tx\_Frame\_Processor (TFP) block is the primary data processing block in the both SONET mode and cell mode. It organizes the cell data into a SONET frame before sending it to the SERDES. In cell mode, the 32-bit data comes from the TX FIFO block. The three major TFP sub-blocks were described in the SONET mode section.

In cell mode, the Payload sub block is activated by the link\_frm\_sync in cell mode. A pulse on this signal from the OPC indicates the start of a frame. Each frame contains 4 different types of bytes in cell mode:

- TOH bytes (Auto TOH mode only)
- Link Header (LH) bytes
- Cell payload bytes
- Pad bytes

There are  $N \times 3$  ( $N = 48$ ) bytes of TOH per row and there are a total of 9 rows in a SONET frame. In cell mode, the rest of the bytes in each row after the TOH bytes are filled by cells. The first byte in a cell is a Link Header (LH) byte. At the end of each frame, there are pad bytes if required.

An important function of the payload block is the grouping of bytes together to be presented to the scramble logic. Due to the insertion of the LH byte in each cell, the total cell data are not divisible by 4 (4 bytes are sent per 77.76 MHz clock cycle). At the end of each row within a SONET frame, the payload block stops sending cell data and indicates to the TOH block to insert the next row's TOH bytes. At the end of the TOH byte transmission, the cell data transmission is resumed.

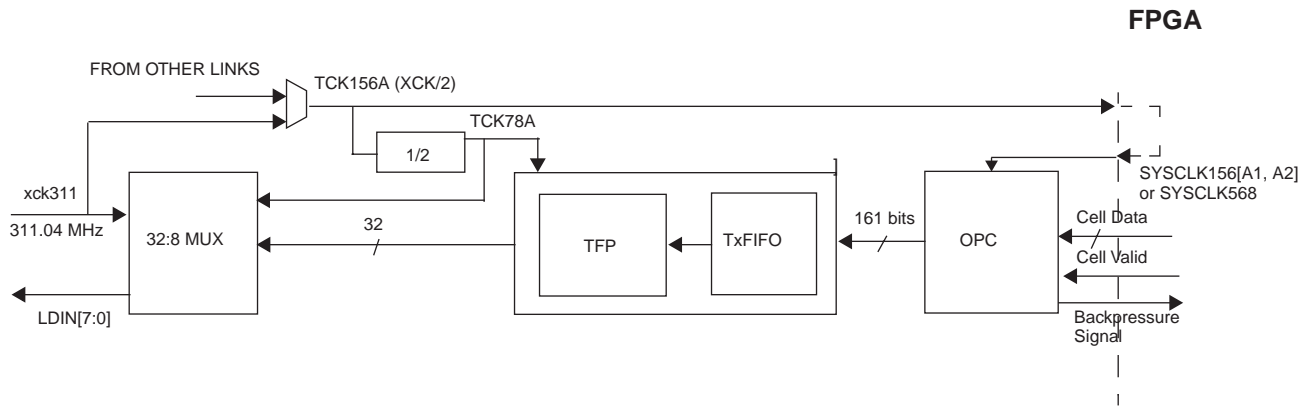
At the end of a cell, the cell's BIP-8 byte is inserted. The next cell's Link Header byte (LH) immediately follows the previous cell's BIP-8 byte. The MSB of the LH byte is the link idle cell indicator bit. The payload block gets this bit from the MSB of the first word of a cell in the memory and inserts it into the LH byte for the appropriate cell. The rest of the LH byte is the link sequence number. This number is incremented for each subsequent cell.

After all cells have been transmitted, the appropriate number of pad bytes are sent. At a link speed of 2.5 GHz, there are 38,880 bytes (SPE + TOH) per frame. There are 1296 bytes of TOH and 37,584 bytes of SPE. For cell data of 85 bytes this translates to 442 cells per frame and 14 pad bytes (# of cells per frame = # of bytes of SPE/# of bytes in a cell).

**Cell Mode Transmit Timing**

Figure 36 shows the transmit clocks and recommended clocking scheme in cell mode. TCK156A, TCK151B can be used as a 156 MHz clock source for SYSCLK156[A1, A2] and SYSCLK156[B1, B2] respectively. SYSCLK156[A1, A2] and SYSCLK[B1, B2] are shared with the receive logic in cell mode.

**Figure 36. Cell Mode Transmit Timing**



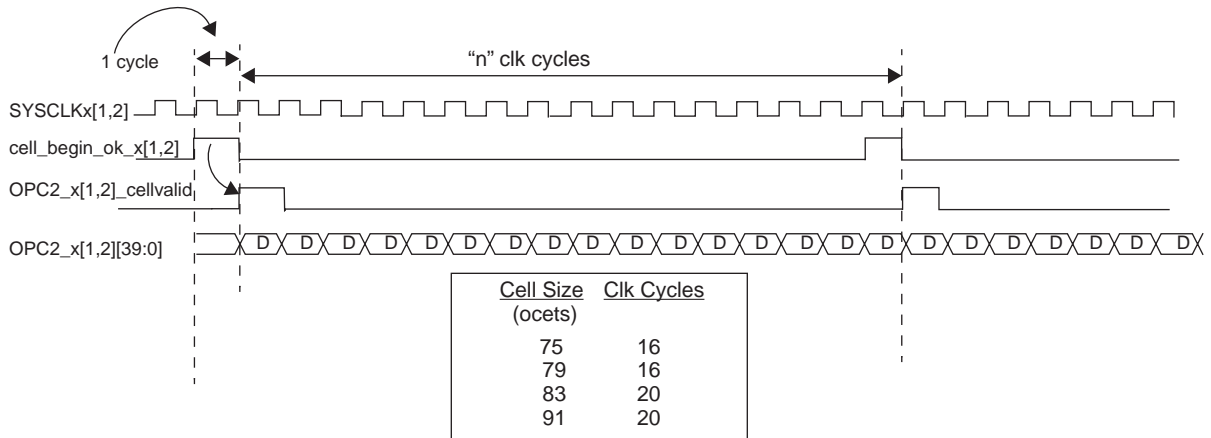
When operating in the two-link CELL MODE, each OPC2 Block passes User cells from FPGA to embedded core. Depending upon the configured CELL SIZE, cell transfers will take a variable number of SYSCLK156 cycles to be transmitted across the interface. Data are always transferred across a 40-bit bus (5 octets per clock cycle). Figure 37 shows 16 clock cycles for a cell transfer this corresponds to a User Cell size of 79 octets. The two control signals in the figure are defined as:

**cell\_begin\_ok:** cell request signal from core to FPGA. It will be asserted every 20 or 16 clock cycles (depending on cell size) when the core is ready to accept cells from FPGA.

- If core FIFO cannot accept cells, cell\_begin\_ok will be low.
- If core FIFO is empty then cell\_begin\_ok will be asserted every 4 clock cycles until cellvalid is asserted by user to indicate valid cell data.

**cellvalid:** Clock-wide pulse asserted by user to indicate valid data. Asserted on the clock cycle following cell\_begin\_ok.

Figure 37. ORSO82G5 Transmit FPGA Interface OPC2 Cell Mode

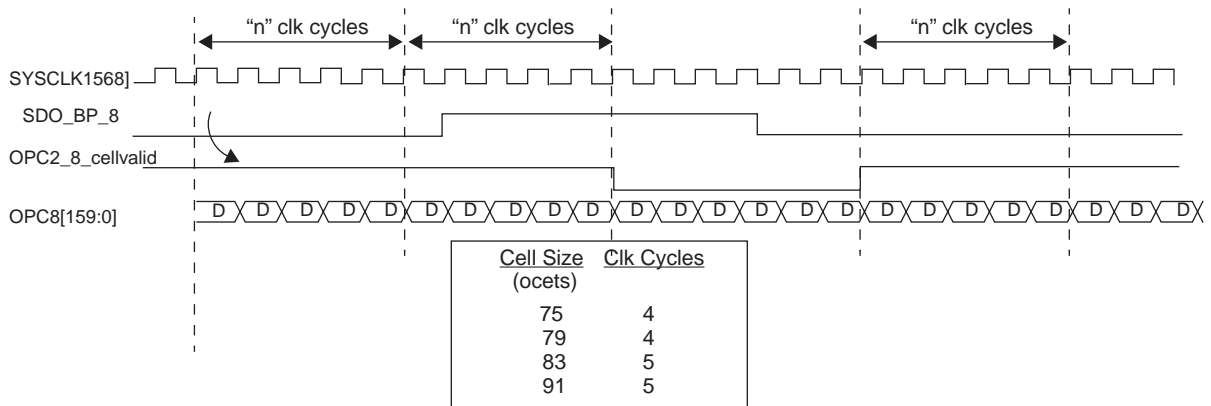


When operating in the eight-link cell mode, the OPC8 block passes user cells from FPGA to embedded core. Depending upon the configured CELL SIZE, cell transfers will take a variable number of SYSCLK156 cycles to be transmitted across the interface. Data are always transferred across a 160-bit bus (20 octets per clock cycle). Figure 38 shows five clock cycles for a cell transfer this corresponds to a User Cell size of 91 octets. The two control signals in the figure are defined as:

**sdo\_bp\_8:** Backpressure signal from core instructing user to stop sending cell data. User should complete transmitting the current cell and can send one more cell before deasserting cellvalid.

**cellvalid:** Is high throughout a cell transfer to indicate valid cell data

Figure 38. ORSO82G5 Transmit FPGA Interface OPC8 Cell Mode

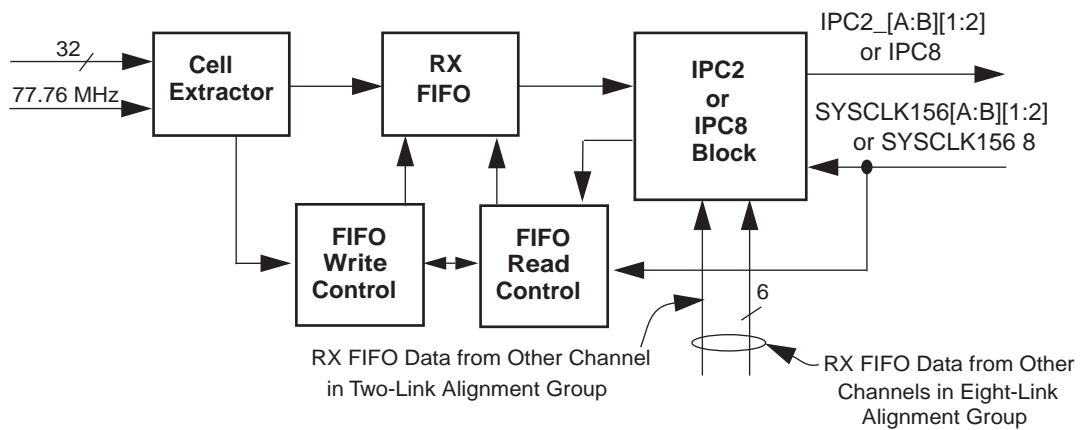


**Cell Mode Receive Path**

The receive logic blocks unique to the cell mode are shown in Figure 39 and are described in the next sections. Prior to reaching this logic the received data has been demultiplexed, frame aligned and descrambled by the SERDES and SONET logic and is formatted on a per channel basis as 32-bit words with an accompanying clock. The clock is a 77.76 MHz clock provided by the DEMUX block performing a divide-by-4 operation on RWCKxx.

The Data Extractor and receive FIFO (RXFIFO) process the data on a per channel basis. The receive FIFO also performs a clock domain transfer to the 156 MHz domain of the Input Port Controller (IPC2/8) blocks. The IPC2/8 blocks perform the two-link or eight-link alignment functions. In two-link alignment mode, the received data are passed to the FPGA logic as 40-bit words at the 156 MHz rate. In eight-link alignment mode, the received data are passed to the FPGA logic as a single 160-bit word, again at the 156 MHz rate. Additional mode-dependent status information is also provided across the Core/FPGA interface.

**Figure 39. Receive Path Logic Unique to Cell Mode**



---

## Cell Extractor

This block is used only in cell mode and does the following:

- Extracts User cells from the SPE
- Performs BIP calculation/checks to verify cell integrity
- Link Header Sequence Interrogation

Processing options include:

- Cell handling for invalid sequence (drop or pass to FPGA)
- S/W configurable 'link removal' due to excessive sequence errors

Data from the cell extractor block(s) is sent to the receive FIFO which aligns the data to the system clock domain and provides for deskew between the links.

### Cell Extraction and BIP Calculation/Checking

The data from the descrambler are passed into the data extractor which strips the cell data from the payload of a SONET frame. The block extracts the BIP value from the data stream and also perform an internal cell BIP calculation. The block also determines when the next Link Header is coming in the frame and what the cell sequence number contained within it should be. If the value of the cell sequence counter is not equal to the expected value, then an error flag will be sent to the software register interface.

### Link Header Detector

The Link Header detector determines when the next Link Header is coming in the frame and what the sequence number contained within it should be. If the value of the cell sequence counter is not equal to the expected value, then an alarm signal will be set to the register map, and increment the sequence counter to the next sequence number. The sequence count value is NOT updated with the incorrect value, but is incremented each time a Link Header is received.

If excessive sequence errors are detected (three or more), and the AUTO\_REMOVE register bit is set then the corresponding link will be treated as not valid. If while the link is in the invalid state, 3 consecutive valid sequence numbers are detected, and (the AUTO\_REJOIN register bit is set) then the link is again valid. If the AUTO\_REMOVE register bit is not set then the link is still valid when excessive sequence errors are detected. While trying to regain sequence number synchronization the sequence number used for comparison will be reloaded from the Link Header until synchronization is achieved.

## Receive FIFO

The main clock domain transfer for the data path is handled by the receive FIFO. A 16 x 161 FIFO is used in cell mode. The FIFO is implemented as a dual-port memory which will support simultaneous reads and writes. The receive FIFO block is written to at 77.76 MHz and read at 156 MHz.

The receive FIFO can allow for inter-link skew of about 800 ns ( $16 \times 160 = 2560$  bits, 400 ps per bit gives 1024 ns). The 160 LSBs in the memory are received data and the 161st bit indicates the start of a new cell. The FIFO write control logic indicates to the IPC, the start of a new frame of data. This signal will only be active for the A1 word of a frame.

Once frame synchronization has occurred and the IPC has responded with a FIFO enable signal, data will be written into the memory. Only the payload (cells) is written to the FIFO. The TOH bytes are not written into the FIFO. The cell octets immediately following the A1A2 bytes will be always written to the top of the FIFO.

Once a full cell has been written to the memory, the write control logic will send a control signal to the IPC8 or IPC2 block which will start the process of reading data from the FIFO. The IPC will read one whole cell at a time from each of the 8 FIFO blocks, if configured for the eight-link cell mode or from each of 2 FIFO blocks if configured for the two-link cell mode.

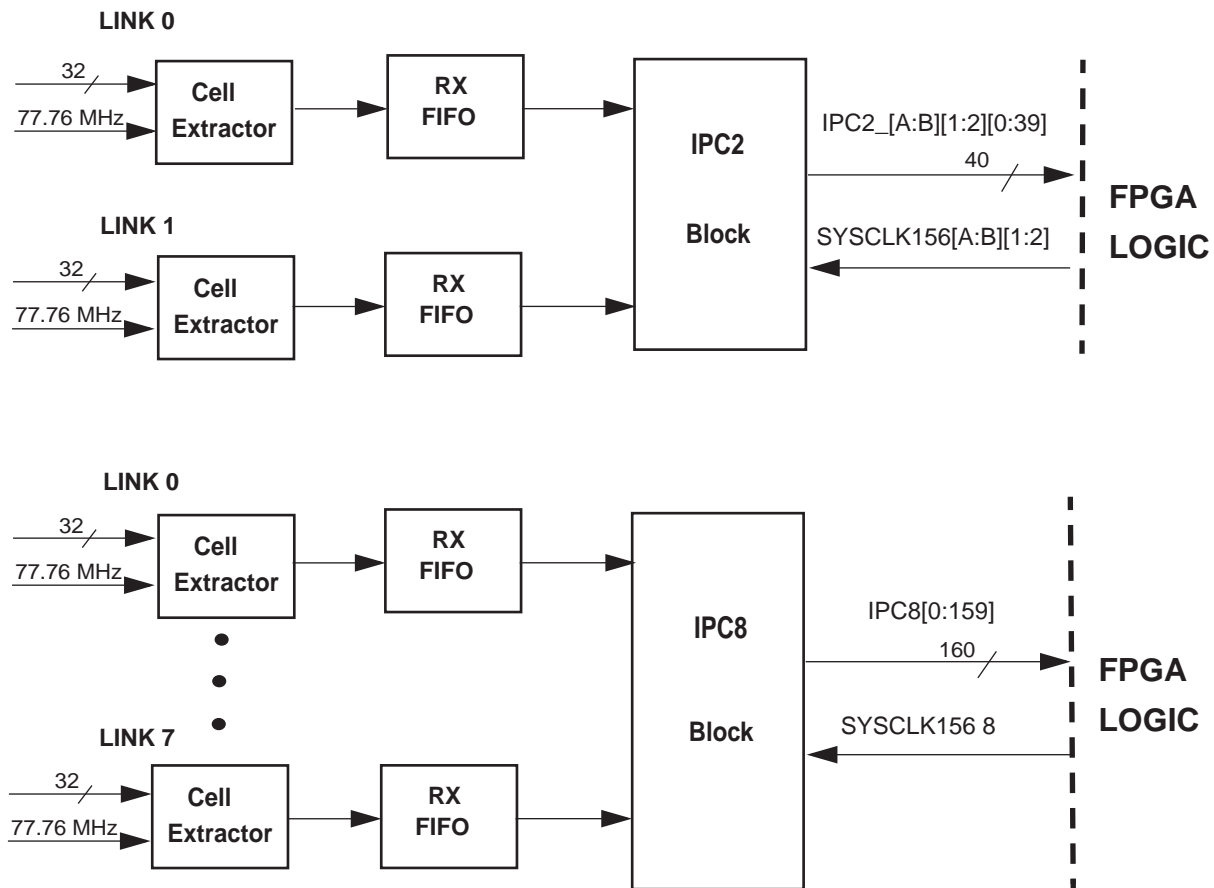
A FIFO occupancy counter generates a RX\_FIFO\_OVRUN indication to the register interface if it detects a FIFO overflow condition. The cell mode allows for alignment of all eight-links or alignment of two-links. Thus there will be 2 IPC blocks for 2 pairs of channels per quad. Each of the twin-bundle IPC blocks will operate at 1/4 of 156MHz frequency.

### Input Port Controllers

The input port controllers (IPCs) are the block responsible for “directing traffic” for the ingress receive cell traffic flow. The block diagrams for the 2-link and 8-link IPCs are shown in Figure 40. They provide the following essential functions.

- Determining when cell data can be read from the FIFOs of the individual links.
- Insuring group bundles are properly aligned.
- Scheduling reads from the RX FIFOs. Cells are read one at a time from the configured links.
- Parsing the cell data into payload data (along with selected header information). Cells which have errors that make them unusable (such as BIP or sequence number errors) are thrown away. This dropping of errored cells can be disabled through register bits CELL\_BIP\_INH\_xx and CELL\_SEQ\_INX\_xx

Figure 40. IPC2 and IPC8 Block Diagrams



There are 5 IPC blocks in the embedded core. There is an IPC2 block for every channel pair:

- IPC2\_A1 combines links from channels AA,AB
- IPC2\_A2 combines links from channels AC,AD

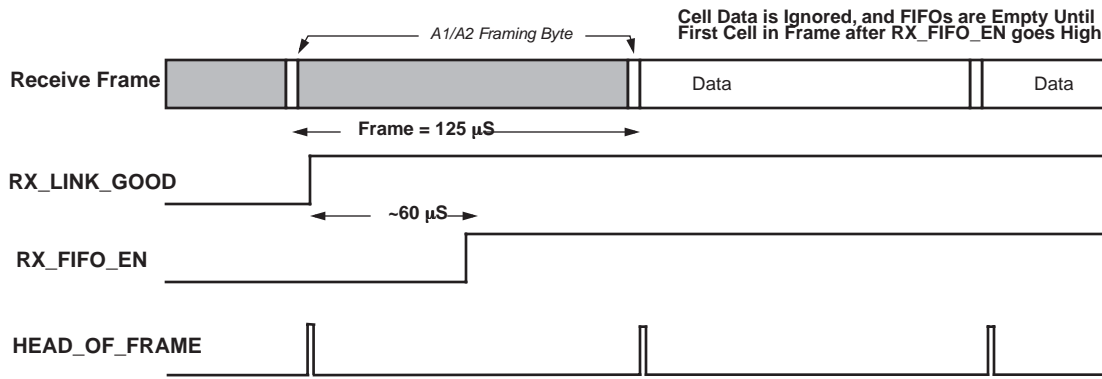
- IPC2\_B1 combines links from channels BA, BB
- IPC2\_B2 combines links from channels BC, BD

The IPC8 block combines cells from all 8 aligned links and transmits them to the FPGA logic.

Before an IPC can begin reading data from the RXFIFOs and assembling cells, it must first align all FIFOs in a port bundle. This is accomplished by handshaking signals between the framer and IPC. The framer indicates to the IPC that framing has been acquired. The behavior of the IPC is dependent on the AUTO\_BUNDLE register bit. If AUTO\_BUNDLE is set for all links in a bundle, the link is brought up when the link is considered good by the framers. If AUTO\_BUNDLE is not set, all links in the bundle must be valid before the link is brought up. The framer does not start filling the FIFOs, however, until the next A1/A2 SONET signal.

As shown in Figure 41, the FIFOs are not ready to be read by the IPC until 60 μs after all links in a code group is up. This ensures that all cells are aligned coming out of the FIFOs for the code group.

**Figure 41. Frame Start-up Timing**



The next thing the IPC must do is determine when the reads may begin. Before reading any data from a FIFO, the FIFO must have a full cell available to be read. This is indicated by each FIFO and is registered by the IPC. The IPC then makes sure that the cells in a given port are received in the order that they are transmitted

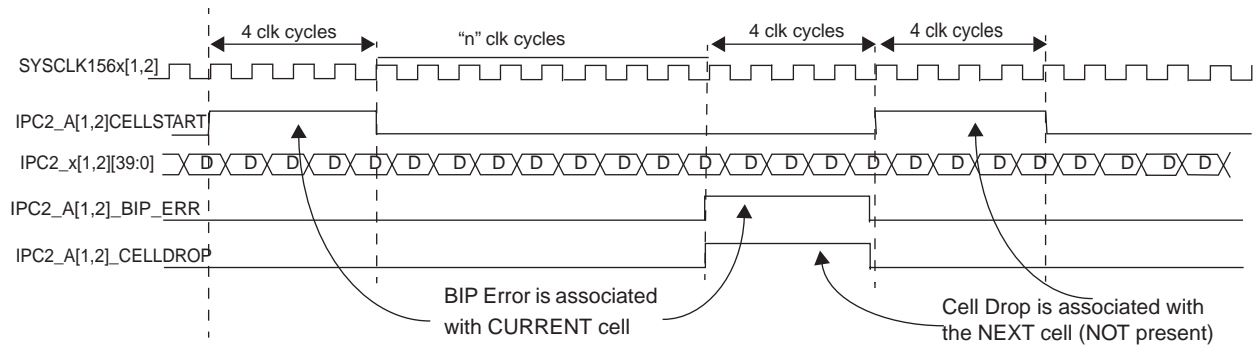
The twin-link IPC2 blocks can be run at 1/4 the of 156 MHz rate (39 MHz) since each block parses cells from only two-links. A 4:1 MUX is used for each IPC2 block to provide a 40-bit bus at 156 MHz across the boundary.

**IPC Receive Cell Mode Timing Core/FPGA**

This section contains timing diagrams for major interfaces of this block to the FPGA logic when cells are to be transferred. Figure 42 shows the cell twin-link mode timing. The number of clock cycles to transfer the cell data depends on the payload size enabled. Error indications for CELL BIP errors and CELL DROP are also shown.



Figure 42. ORSO82G5 Receive FPGA/Embedded Core Interface IPC2 Mode



**CELL BIP ERROR**

If a Cell BIP Error occurs, the CELL\_BIP\_ERR signal reflects the occurrence, as shown in the Figure. For 2-Link CELL MODE, the CELL\_BIP\_ERR signal is asserted during the last 4 clock cycles of the receive cell.

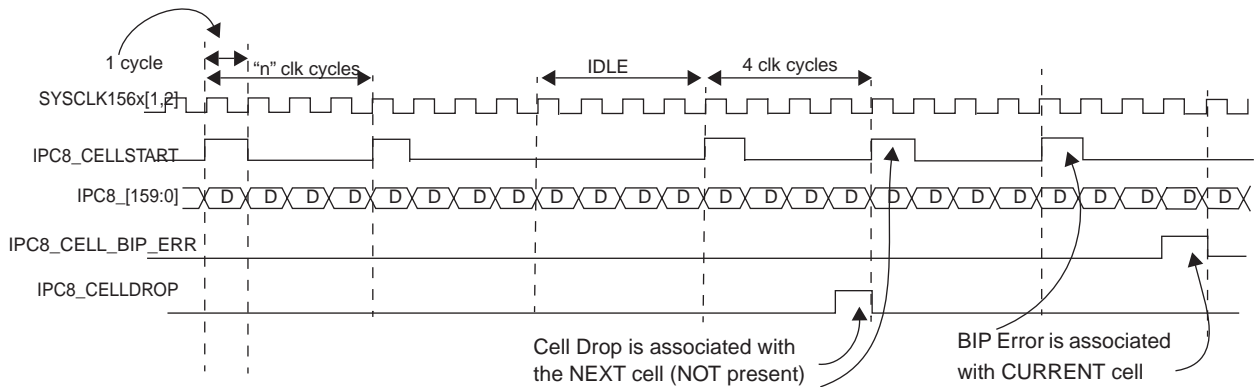
**CELL BIP ERROR**

If a cell error occurs within the ASB and;  
 1. CELL\_BIP\_INH=0 ...Do not drop BIP errored cells (s/w selectable)  
 2. A BIP error occurs  
 The drop indicator will PRECEED the user cell that contains the BIP error. All data will be passed w/o modification.

When operating in CELL MODE, the IPC2 Block passes user cells as well as control and status signals to the user. Depending upon the configured CELL SIZE, cell transfers will take a variable number of SYSCLK156 cycles to be received across the interface. Data are always transferred across a 40-bit bus (5 octets per clock cycle). Figure 42 shows 16 clock cycles for a cell transfer. This corresponds to a User Cell size of 79 octets.

Figure 43 shows cell octal alignment mode timing. When operating in CELL MODE, the IPC8 Block aligns all 8 channels of receive data on a FRAME basis. The IPC8 also passes user cells as well as control and status signals to the user. Depending upon the configured CELL SIZE, cell transfers will take a variable number of SYSCLK156 cycles to be received across the interface. Data are always transferred across an 160-bit bus (20 octets per clock cycle). Figure 43 shows 4clock cycles for a cell transfer. This corresponds to a User Cell size of 79 octets.

Figure 43. ORSO82G5 Receive FPGA/Embedded Core Interface IPC8 Mode



**CELL BIP ERROR**

If a Cell BIP Error occurs, the CELL\_BIP\_ERR signal reflects the occurrence, as shown in the Figure. For 2-Link CELL MODE, the CELL\_BIP\_ERR signal is asserted during the last 4 clock cycles of the receive cell.

**CELL BIP ERROR**

If a cell error occurs within the ASB and;  
 1. CELL\_BIP\_INH=0 (Do not drop BIP errored cells)  
 2. A BIP error occurs  
 The drop indicator will PRECEED the user cell that contains the BIP error. All data will be passed w/o modification.

In the SERDES-only mode the data are simply transferred as 32-bit wide words to and from the FPGA logic. The next sections describe the signal definitions for the TX and RX paths in the SONET, OPC2 and OPC8 modes. The signal names unique to an operating mode are preferred for design and are generally the ones used in the *ispLEVER* design environment. The labels in the left most column are the hardware FPGA interface names. The *ispLEVER* software creates an HDL module with specific names based on the mode selected for each channel. The pin mapping that is performed by the *ispLEVER* is shown in Table 9 and Table 10.

The interface signals for the embedded RAM are completely independent of these signals. The memory signals are described in a later section.

**Signal Description for TX Path (FPGA to SERDES Core)**

- Signals are divided across 8 channels with 40 signals per channel. TXDxx[39:0] is the set of 40 signals for a channel xx.
- The data signals multiplexing scheme is similar to the one used for the RXD signals. However, the status signals multiplexing is different. Please refer to Table 9. for the detailed description of the TXD multiplexing scheme.
- For all channels the TXDxx[39:33] signals are not used.

Table 9 summarizes the signals at the FPGA/Core interface in the transmit direction.

**Table 9. TX FPGA/Core Interface Signaling**

<b>TXDAA</b>	<b>SONET Mode</b>	<b>OPC2 A1 Mode</b>	<b>OPC8 Mode</b>
[39:33]	—	—	—
32	DINAA_FP	—	—
[31:21]	DINAA[31:21]	—	—
20	DINAA[20]	OPC2_A1_CELLVALID	—
[19:0]	DINAA[19:0]	OPC2_A1[39:20]	OPC8[159:140]
<b>TXDAB</b>	<b>SONET Mode</b>	<b>OPC2 A1 Mode</b>	<b>OPC8 Mode</b>
[39:33]	—	—	—
32	DINAB_FP	—	—
[31:20]	DINAB[31:20]	—	—
[19:0]	DINAB[19:0]]	OPC2_A1[19:0]]	OPC8[139:120]]
<b>TXDAC</b>	<b>SONET Mode</b>	<b>OPC2 A2 Mode</b>	<b>OPC8 Mode</b>
[39:33]	—	—	—
32	DINAC_FP	—	—
[31:21]	DINAC[31:21]	—	—
20	DINAC[20]	OPC2_A2_CELLVALID	—
[19:0]	DINAC[19:0]	OPC2_A2[39:20]	OPC8[119:100]
<b>TXDAD</b>	<b>SONET Mode</b>	<b>OPC2 A1 Mode</b>	<b>OPC8 Mode</b>
[39:33]	—	—	—
32	DINAD_FP	—	—
[31:21]	DINAD[31:21]	—	—
20	DINAD[20]	—	OPC8_CELLVALID
[19:0]	DINAD[19:0]	OPC2_A2[19:0]	OPC8[99:80]

**Table 9.. TX FPGA/Core Interface Signals (continued)**

<b>TXDBA</b>	<b>SONET Mode</b>	<b>OPC2 B1 Mode</b>	<b>OPC8 Mode</b>
[39:33]	—	—	—
32	DINBA_FP	—	—
[31:21]	DINBA[31:21]	—	—
20	DINBA[20]	OPC2_B1_CELLVALID	—
[19:0]	DINBA[19:0]]	OPC2_B1[39:20]	OPC8[79:60]
<b>TXDBB</b>	<b>SONET Mode</b>	<b>OPC2 B1 Mode</b>	<b>OPC8 Mode</b>
[39:33]	—	—	—
32	DINBB_FP	—	—
[31:20]	DINBB[31:20]	—	—
[19:0]	DINBB[19:0]	OPC2_B1[19:0]	OPC8[59:40]
<b>TXDBC</b>	<b>SONET Mode</b>	<b>OPC2 B2 Mode</b>	<b>OPC8 Mode</b>
[39:33]	—	—	—
32	DINBC_FP	—	—
[31:21]	DINBC[31:21]	—	—
20	DINBC[20]	OPC2_B2_CELLVALID	—
[19:0]	DINBC[19:0]	OPC2_B2[39:20]	OPC8[39:20]
<b>TXDBD</b>	<b>SONET Mode</b>	<b>OPC2 B2 Mode</b>	<b>OPC8 Mode</b>
[39:33]	—	—	—
32	DINBD_FP	—	—
[31:20]	DINBD[31:20]	—	—
[19:0]	DINBD[19:0]	OPC2_B2[19:0]	OPC8[19:0]

**Signal Description for RX Path (SERDES Core to FPGA)**

- Signals are divided across 8 channels with 40 signals per channel. RXDxx[39:0] is the set of 40 signals for a channel xx. Figure 44 illustrates how the signals are multiplexed.
- All RX direction signals are outputs from the core.
- See Table 9 and Figure 44 for clock transfers across the FPGA/Core interface.
- In SONET mode, RXDxx[31:0] carries 32 bit data from the alignment FIFO of the respective channel. RXDxx[35:32] carries miscellaneous information such as OOF, BIPERR, Frame Pulse (FP), and SPE.
- In cell mode, data from each of the four 2-link IPC bundles are spread across all eight channels and are assigned to the 20 LSB's (RXDxx[19:0]) of each channel output. Data from IPC2\_A1 is distributed across RXDAA[19:0] and RXDAB[19:0]. Data from IPC2\_A2 is distributed across RXDAC[19:0] and RXDAD[19:0]. This symmetry is maintained for IPC2 data signals from quad B.
- Data from the 8-link IPC block IPC8 is spread across all eight channels and assigned to the 20 LSB's (RXDxx[19:0] of each channel output.
- The IPC status signals for Cell Mode operation are contained in RXDxx[39:36] and RXDxx[33].
- The signals for SONET Mode operation are assigned to RXDxx[35:34] and RXDxx[33].
- Note that RXDxx[39:32] signal assignments are the same no matter what mode the RX blocks are in.

Figure 44. MUX Structure RX Core/FPGA Interface

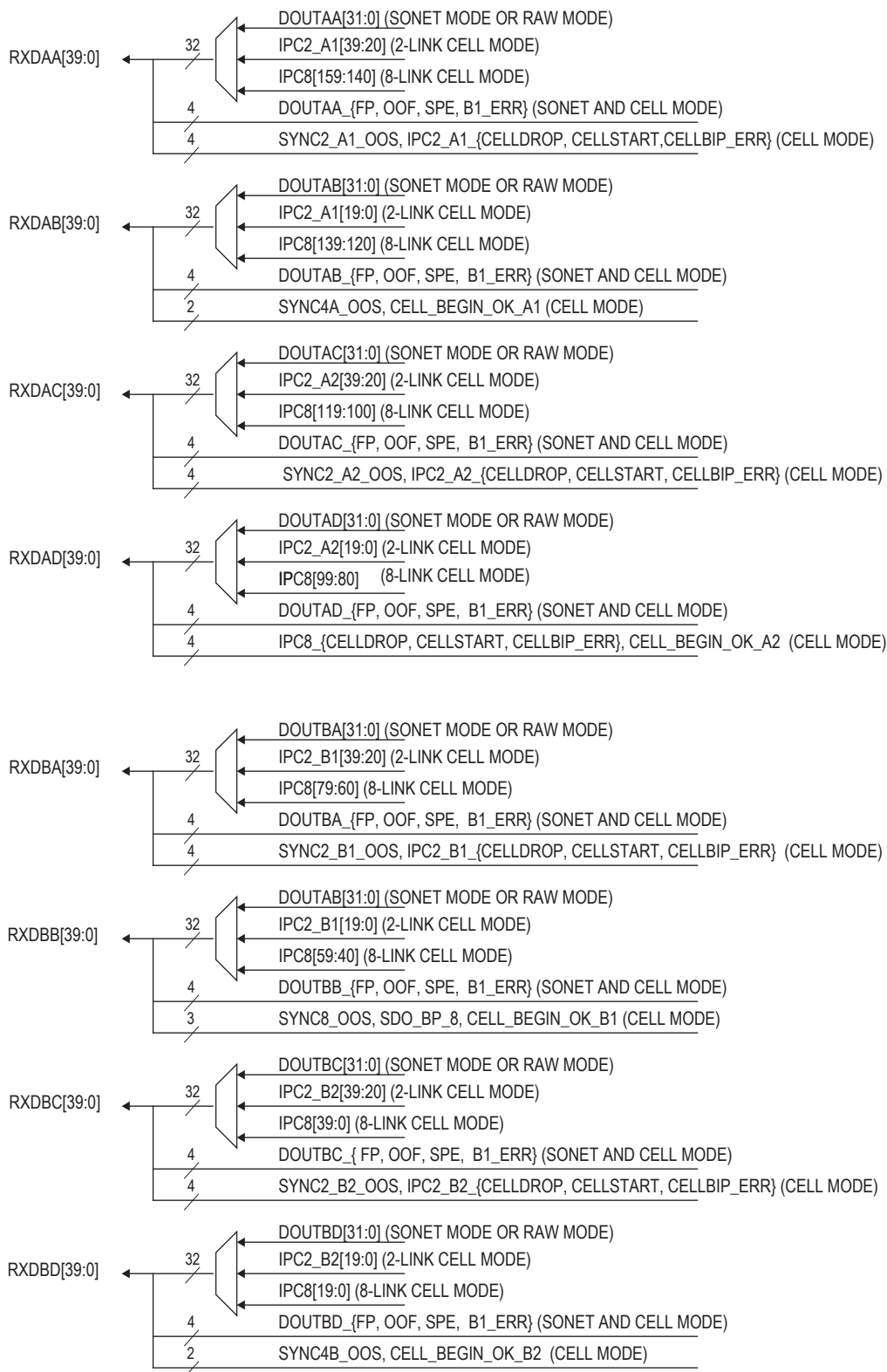


Table 10 summarizes the signals across the Core/FPGA interface in the receive direction.

**Table 10. RX Core/FPGA Interface Signals**

<b>RXDAA[39:0]</b>	<b>SONET mode</b>	<b>IPC2 A1 Mode</b>	<b>IPC8 Mode</b>
39	SYNC2_A1_OOS	—	—
38	—	IPC2_A1_CELLDROP	—
37	—	IPC2_A1_CELLSTART	—
36	DOUTAA_FP	—	—
35	—	DOUTAA_OOF	—
34	DOUTAA_SPE	—	—
33	—	IPC2_A1_CELL_BIP_ERR	—
32	—	DOUTAA_B1_ERR	—
[31:20]	DOUTAA[31:20]	—	—
[19:0]	DOUTAA[19:0]	IPC2_A1[39:20]	—
<b>RXDAB[39:0]</b>	<b>SONET mode</b>	<b>IPC2 A1 Mode</b>	<b>IPC8 Mode</b>
39	—	—	—
38	SYNC4_A_OOS	—	—
37	—	CELL_BEGIN_OK_A1	—
36	DOUTAB_FP	—	—
35	—	DOUTAB_OOF	—
34	DOUTAB_SPE	—	—
33	—	—	—
32	—	DOUTAB_B1_ERR	—
[31:20]	DOUTAB[31:20]]	—	—
[19:0]	DOUTAB[19:0]	IPC2_A1[19:0]	IPC8[139:120]
<b>RXDAC[39:0]</b>	<b>SONET mode</b>	<b>IPC2 A2 Mode</b>	<b>IPC8 Mode</b>
39	SYNC2_A2_OOS	—	—
38	—	IPC2_A2_CELLDROP	—
37	—	IPC2_A2_CELLSTART	—
36	DOUTAC_FP	—	—
35	—	DOUTAC_OOF	—
34	DOUTAC_SPE	—	—
33	—	IPC2_A2_CELL_BIP_ERR	—
32	—	DOUTAC_B1_ERR	—
[31:20]	DOUTAC[31:20]]	—	—
[19:0]	DOUTAC[19:0]	IPC2_A2[39:20]	IPC8[119:100]
<b>RXDAD[39:0]</b>	<b>SONET Mode</b>	<b>IPC2 A2 Mode</b>	<b>IPC8 Mode</b>
39	—	—	IPC8_CELLDROP
38	—	—	IPC8_CELLSTART
37	—	CELL_BEGIN_OK_A2	—
36	DOUTAD_FP	—	—
35	—	DOUTAD_OOF	—
34	DOUTAD_SPE	—	—

**Table 10. RX Core/FPGA Interface Signals (continued)**

<b>RXDAD[39:0]</b>	<b>SONET Mode</b>	<b>IPC2 A2 Mode</b>	<b>IPC8 Mode</b>
33	—	—	IPC8_CELL_BIP_ERR
32	DOUTAD_B1_ERR		
[31:20]	DOUTAD[31:20]	—	
[19:0]	DOUTAD[19:0]	IPC2_A2[19:0]	IPC8[99:80]
<hr/>			
<b>RXDBA[39:0]</b>	<b>SONET Mode</b>	<b>IPC2 B1 Mode</b>	<b>IPC8 Mode</b>
39	SYNC2_B1_OOS	—	
38	—	IPC2_B1_CELLDROP	—
37	—	IPC2_B1_CELLSTART	—
36	DOUTBA_FP	—	
35	DOUTBA_OOF		
34	DOUTBA_SPE	—	
33	—	IPC2_B1_CELL_BIP_ERR	—
32	DOUTBA_B1_ERR		
[31:20]	DOUTBA[31:20]	—	
[19:0]	DOUTBA[19:0]	IPC2_B1[39:20]	IPC8[79:60]
<hr/>			
<b>RXDBB[39:0]</b>	<b>SONET Mode</b>	<b>IPC2 B1 Mode</b>	<b>IPC8 Mode</b>
39	SYNC8_OOS	—	
38	—	—	SDO_BP_8
37	—	CELL_BEGIN_OK_B1	—
36	DOUTBB_FP	—	
35	DOUTBB_OOF		
34	DOUTBB_SPE	—	
33	—		
32	DOUTBB_B1_ERR		
[31:20]	DOUTBB[31:20]	—	
[19:0]	DOUTBB[19:0]	IPC2_B1[19:0]	IPC8[59:40]
<hr/>			
<b>RXDBC[39:0]</b>	<b>SONET Mode</b>	<b>IPC2 B2 Mode</b>	<b>IPC8 Mode</b>
39	SYNC2_B2_OOS	—	
38	—	IPC2_B2_CELLDROP	—
37	—	IPC2_B2_CELLSTART	—
36	DOUTBC_FP	—	
35	DOUTBC_OOF		
34	DOUTBC_SPE	—	
33	—	IPC2_B2_CELL_BIP_ERR	—
32	DOUTBC_B1_ERR		
[31:20]	DOUTBC[31:20]	—	
[19:0]	DOUTBC[19:0]	IPC2_B2[39:20]	IPC8[39:20]

Table 10. RX Core/FPGA Interface Signals (continued)

RXDBD[39:0]	SONET Mode	IPC2 B2 Mode	IPC8 Mode
39		—	
38	SYNC4_B_OOS		—
37	—	CELL_BEGIN_OK_B2	—
36	DOUTBD_FP		—
35		DOUTBD_OOF	
34	DOUTBD_SPE		—
33		—	
32		DOUTBD_B1_ERR	
[31:20]	DOUTBD[31:20]		—
[19:0]	DOUTBD[19:0]	IPC2_B2[19:0]	IPC8[19:0]

### Reset Conditions

The SERDES block can be reset in one of three different ways: on power up, using the hardware reset (PASB\_RESETN) or by setting bits in the control registers. The power up reset process begins when the power supply voltage ramps up to approximately 80% of the nominal value of 1.5 V. Following this event, the device will be ready for normal operation after 3 ms.

A hardware reset is initiated by making the PASB\_RESETN low for at least two microprocessor clock cycles. The device will be ready for operation 3 ms after the low to high transition of the PASB\_RESETN. This reset function affects all SERDES blocks and resets all core control, status and data path registers and counters.

Using the software reset option, each channel can be individually reset by setting SWRST bit to a logic 1 in the SERDES channel configuration register. The device will be ready 3 ms after the SWRST bit is deasserted. Similarly, all four channels per quad SERDES can be reset by setting the global reset bit GSWRST. The device will be ready for normal operation 3 ms after the GSWRST bit is deasserted. Note that the software reset options resets only the SERDES internal registers and counters on a per channel or per quad basis. The core non-SERDES registers and logic blocks are not affected. It should also be noted that the embedded core registers and logic blocks cannot be accessed until after FPGA configuration is complete.

Table 11. summarizes the conditions under which the embedded core registers, SERDES, and embedded core logic are reset under user control. The embedded core status registers are also reset on read.

**Table 11. ORSO82G5 Reset Conditions**

Reset Signal	Embedded Core Non-SERDES Registers	SERDES Registers and Blocks	Other Embedded Core Blocks	SERDES I/O	Notes
Powerup	reset	reset	reset	Disabled	Power on reset
PASB_RESETN pin = 0 (Hard Reset)	reset	reset	reset	Disabled	External input pin
FPGA Configuration	reset	reset	reset	Disabled	DONE pin = 0
Partial FPGA Reconfiguration (with option disable TRI_IO)	—	—	—	Active	DONE pin = 0
Internal Signal FPGA_RESET = 1	—	—	reset	Active	FPGA_RESET is FPGA sourced
FPGA GSRN signal = 0	—	—	reset	Active	GSRN is FPGA sourced. Set GSRN_DISABLE = 1 to disable this reset
SOFT_RESET = 0, 1, 0 (System Bus register based)	—	—	reset	Active	write SOFT_RESET = 1 (ON) then write SOFT_RESET = 0 (OFF)
TS_ALL Pin = 1	—	—	—	Active	External input pin
SWRST_xx= 0,1,0 xx = [AA,...,BD]	—	Selected channel reset	—	Active	Per channel software reset (Not self-clearing, must be manually set and cleared.)
GSWRST_[A:B] = 0,1,0	—	Selected quad reset	—	Active	Per quad software reset (Not self-clearing, must be manually set and cleared.)



**Embedded Core Block RAM**

There are two independent memory slices (labeled A and B) in the embedded core. Each memory slice has a capacity of 4K words by 36 bits. These are in addition to the block RAMs found in the FPGA portion of the ORSO82G5. Although the memory slices are in the embedded core part of the chip, they do not interact with the rest of the embedded core circuits, but are standalone memories designed specifically to increase RAM capacity in the ORSO82G5 chip. They can be used by the soft IP cores implemented in the FPGA portion of the FPSC.

A block diagram of a memory slice is shown in Figure 45. Each memory slice is organized into two sections (labeled SRAM A and SRAM B) and has one read port, one write port and four byte-write-enable (active-low) signals. Each byte has eight data bits and a control/parity bit. The control/parity bit responds to the same byte enable (BYTEWN\_x[x]) as it's corresponding data. No special logic such as parity checking is performed on this bit by the core. The read data from the memory is registered so that it works as a pipelined synchronous memory block. The minimum timing specifications are shown in Figure 46 and Figure 47. Signal names and functions are summarized later in Table 12 and follow the general Series 4 naming conventions.

**Figure 45. Block Diagram, Embedded Core Memory Slice**

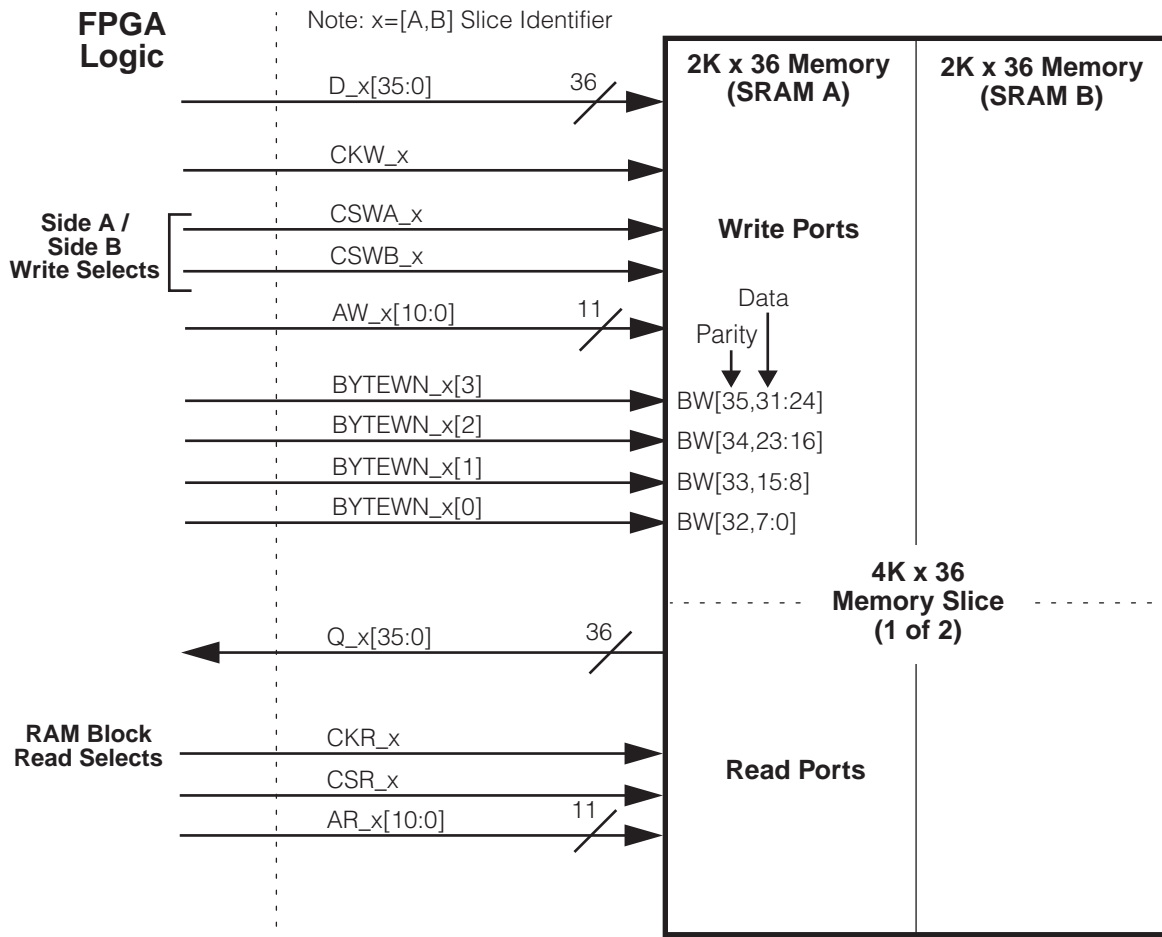
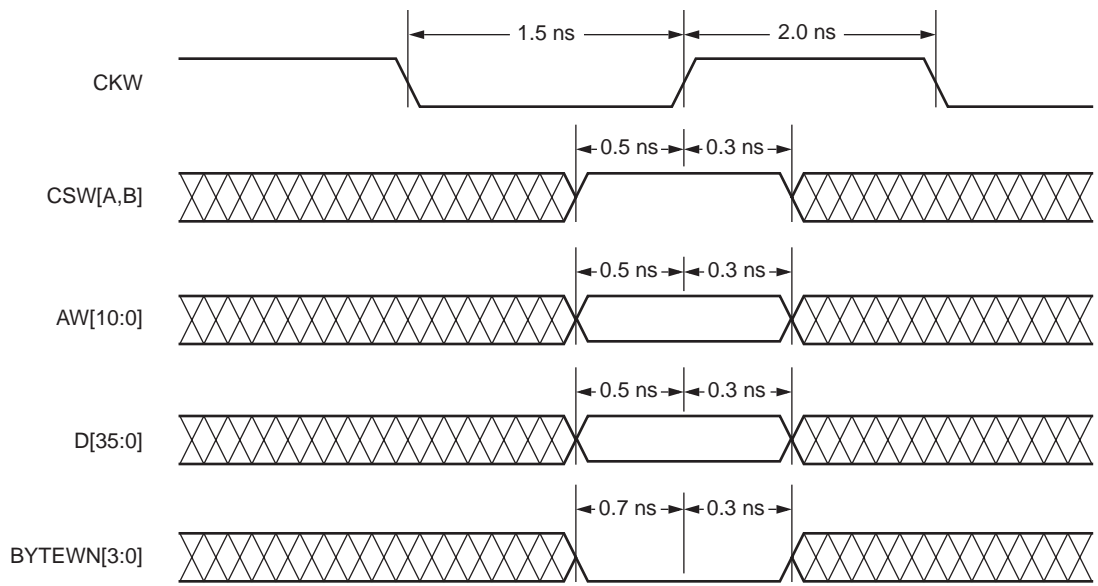


Figure 46. Minimum Timing Specs for Memory Blocks-Write Cycle (-1 Speed Grade)



2271(F)

Figure 47. Minimum Timing Specs for Memory Blocks-Read Cycle (-1 Speed Grade)

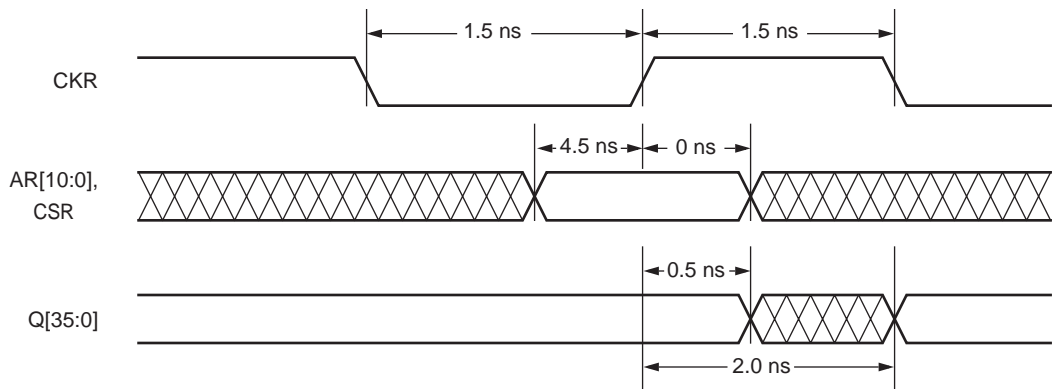


Table 12 summarizes the Embedded Memory Signals at the Core/FPGA interface. In the table, an input refers to a signal flowing into the embedded core and an output refers to a signal flowing out of the embedded core.

**Table 12. Embedded Memory Core/FPGA Interface Signal Description**

FPGA/Embedded Core Interface Signal Name]	Input (I) to or Output (O) from Core	Signal Description
<b>Memory Slice Interface Signals</b>		
D_[A:B][35:0]	I	Data in—memory slice [A:B]
CKW_[A:B]	I	Write clock—memory slice [A:B].
CSWA_[A:B]	I	Write chip select for SRAM A—memory slice [A:B].
CSWB_[A:B]	I	Write chip select for SRAM B—memory slice [A:B].
AW_[A:B][10:0]	I	Write address—memory slice [A:B].
BYTEWN_[A:B][3:0]	I	Write control pins for byte-at-a-time write-memory slice [A:B].
Q_[A:B][35:0]	O	Data out—memory slice [A:B].
CKR_[A:B]	I	Read clock—memory slice [A:B].
CSR_[A:B]	I	Read chip select—memory slice [A:B]. CSR_[A:B]= 0 selects SRAM A. CSR_[A:B]= 1 selects SRAM B.
AR_[A:B][10:0]	I	Read address—memory slice [A:B].

---

## Register Maps

The memory map for the ORSO82G5 core is only part of the full memory map of the ORSO82G5 device. The ORSO82G5 is an ORCA Series4 based device and thus uses the system bus as a communication bridge. The ORSO82G5 core register map contained in this data sheet only covers the embedded ASIC core of the device, not the entire device. The system bus itself, and the generic FPGA memory map, are fully documented in the MPI/System Bus Application Note. As part of the system bus, the embedded ASIC core of an FPSC is located at address offset 0x30000. The ORSO82G5 embedded core is an eight-bit slave interface on the Series 4 system bus.

Each ORCA device contains a device ID. This device ID is unique to each ORCA device and can be used for device identification and assist in the system debugging. The device ID is located at absolute address 0x0-0x3. The ORSO82G5's device ID is 0xDC012282. More information on the device ID and other Series 4 generic registers can be found in the MPI/System Bus Application Note.

The ORSO82G5 core registers are clocked by the system bus main clock. If a clock is not provided to the reference clock, the registers will fail to operate.

The ORSO82G5 core registers do not check for parity on a write operation. On a read operation, no parity is generated, and a "0" is passed back to the initiating bus master interface on the parity signal line.

## Types of Registers

The registers in ORSO82G5 are 8-bit memory locations which, in general, can be classified into several types:

### General Core Status Registers

Read-only registers to convey the status information of various operations within the FPSC core. An example is the state of the LKI-xx receive PLL lock indicator outside the SERDES.

### Alarm Status and Mask Registers

The alarm status registers are enabled or masked by the corresponding alarm enable registers. An example of such an alarm is the Out-Of-Frame (OOF) bit OOF\_xx which is enabled by the corresponding alarm enable bit OOF\_ENxx (xx indicates one of the 8 channels AA-BD). (The OOF and BIP error alarms are also available as signals across the core-FPGA boundary for each channel.) All the alarms for a given channel will be read into a single status bit (ALARM\_STATUS\_[AA-BD]). In addition, an event on any of these alarm status bits will generate an interrupt to the FPSC slave of the interrupt cause register on the system bus interface (see technical note TN1017). All alarm and status registers are clear on read.

### Control Registers

Read-write registers to setup the control inputs that define the operation of the FPSC core. The SERDES block within the ORSO82G5 core has a set of status and control registers for its operation. There is another group of status and control registers which are implemented outside the SERDES, which are related to the SERDES and other functional blocks in the FPSC core. They will be described in detail here.

Each SERDES has four independent channels, which are named A, B, C or D. Using this nomenclature, the SERDES A channels are named as AA, AB, AC and AD, while SERDES B channels will be BA, BB, BC and BD. The register address allocation for the ORSO82G5 is shown in Table 13.. Detailed descriptions of all of the register bits are provided in Table 14. through Table 22..

**Table 13. Memory Space Allocation in the FPSC Core**

Address (Hex)	Description
3000x	Channel A in SERDES A quad, internal registers
3001x	Channel B in SERDES A quad, internal registers
3002x	Channel C in SERDES A quad, internal registers
3003x	Channel D in SERDES A quad, internal registers
3010x	Channel A in SERDES B quad, internal registers
3011x	Channel B in SERDES B quad, internal registers
3012x	Channel C in SERDES B quad, internal registers
3013x	Channel D in SERDES B quad, internal registers
3080x	Channel AA registers outside the SERDES
3081x	Channel AB registers outside the SERDES
3082x	Channel AC registers outside the SERDES
3083x	Channel AD registers outside the SERDES
3090x	Channel BA registers outside the SERDES
3091x	Channel BB registers outside the SERDES
3092x	Channel BC registers outside the SERDES
3093x	Channel BD registers outside the SERDES
30Axx	All Channels registers

**Register Reset Conditions**

The reset conditions for the registers vary depending on register type. See Table 11. and the accompanying text for a description of the reset conditions.

**Table 14. SERDES Alarm and Alarm Mask Register Descriptions**

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
SERDES Alarm Registers (Read Only) xx = [AA, ...,BD]					
30000 - AA	[0]	RSVD	00	Reserved - May be non-zero	Both
30010 - AB	[1]	LKI_xx		Receive PLL Lock Indication, Channel xx. LKI_xx = 1 indicates the receive PLL is locked	
30020 - AC				Reserved - May be non-zero	
30030 - AD	[2:7]	RSVD			
30100 - BA					
30110 - BB					
30120 - BC					
30130 - BD					
SERDES Alarm Mask Registers (Read/Write) xx = [AA, ...,BD]					
30001 - AA	[0]	RSVD	FF	Reserved	Both
30011 - AB	[1]	MLKI_xx		Mask Receive PLL Lock Indication, Channel MLKI_xx = 1 indicates LKI_xx is enabled	
30021 - AC				Reserved	
30031 - AD	[2:7]	RSVD			
30101 - BA					
30111 - BB					
30121 - BC					
30131 - BD					

**Table 15. SERDES Per-Channel Transmit Configuration Register Descriptions**

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
SERDES Transmit Per-Channel Configuration Registers (Read/Write) xx = [AA, ...,BD]					
30002 - AA 30012 - AB 30022 - AC 30032 - AD	[0]	TXHR <sub>xx</sub>	00	Transmit Half Rate Selection Bit, Channel xx. When TXHR <sub>xx</sub> = 1, HDOUT <sub>xx</sub> 's baud rate = (REFCLK[A:B]*8) and TCK78[A:B]=(REFCLK[A:B]/4); when TXHR <sub>xx</sub> =0, HDOUT <sub>xx</sub> 's baud rate = (REFCLK[A:B]*16) and TCK78[A:B]=(REFCLK[A:B]/2). TXHR <sub>xx</sub> = 0 on device reset.	Both
30102 - BA 30112 - BB 30122 - BC 30132 - BD	[1]	PWRDNT <sub>xx</sub>		Transmit Powerdown Control Bit, Channel xx. When PWRDNT <sub>xx</sub> = 1, sections of the transmit hardware are powered down. PWRDNT <sub>xx</sub> = 0 on device reset.	Both
	[2]	PE0 <sub>xx</sub>		Transmit Preemphasis Selection Bit 0, Channel xx. PE0 <sub>xx</sub> and PE1 <sub>xx</sub> select one of three preemphasis settings for the transmit section. PE0 <sub>xx</sub> =PE1 <sub>xx</sub> = 0, Preemphasis is 0% PE0 <sub>xx</sub> =1, PE1 <sub>xx</sub> = 0 or PE0 <sub>xx</sub> =0, PE1 <sub>xx</sub> = 1, Preemphasis is 12.5% PE0 <sub>xx</sub> =PE1 <sub>xx</sub> = 1, Preemphasis is 25%. PE0 <sub>xx</sub> =PE1 <sub>xx</sub> = 0 on device reset.	Both
	[3]	PE1 <sub>xx</sub>			Both
	[4]	HAMP <sub>xx</sub>			Transmit Half Amplitude Selection Bit, Channel xx. When HAMP <sub>xx</sub> = 1, the transmit output buffer voltage swing is limited to half its normal amplitude. Otherwise, the transmit output buffer maintains its full voltage swing. HAMP <sub>xx</sub> = 0 on device reset.
	[5:7]	RSVD		Reserved, Always set to "000"	

**Table 16. SERDES Per-Channel Receive Configuration Register Descriptions**

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
SERDES Receive Per-Channel Configuration Registers (Read/Write) xx = [AA, ...,BD]					
30003 - AA 30013 - AB 30023 - AC 30033 - AD	[0]	RXHR <sub>xx</sub>	20	Receive Half Rate Selection Bit, Channel xx. When RXHR <sub>xx</sub> =1, HDIN <sub>xx</sub> 's baud rate = (REFCLK[A:B]*8) and RCK78[A:B]=(REFCLK[A:B]/4); When RXHR <sub>xx</sub> =0, HDIN <sub>xx</sub> 's baud rate = (REFCLK[A:B]*16) and RCK78[A:B]=(REFCLK/2). RXHR <sub>xx</sub> = 0 on device reset.	Both
30103 - BA 30113 - BB 30123 - BC 30133 - BD	[1]	PWRDNR <sub>xx</sub>		Receiver Power Down Control Bit, Channel xx. When PWRDNR <sub>xx</sub> = 1, sections of the receive hardware are powered down. PWRDNR <sub>xx</sub> = 0 on device reset.	Both
	[2:7]	RSVD		Reserved (Bit 2 = 1 on device reset)	

Table 17. SERDES Common Configuration Register Descriptions

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
<b>SERDES Common Transmit and Receive Channel Configuration Registers (Read/Write) xx = [AA, ...,BD]</b>					
30004 - AA 30014 - AB 30024 - AC 30034 - AD  30104 - BA 30114 - BB 30124 - BC 30134 - BD	[0]	RSVD	40	Reserved	
	[1]	MASK_xx		Transmit and Receive Alarm Mask Bit, Channel xx. When MASK_xx = 1, the transmit and receive alarms of a channel are prevented from generating an alarm (i.e., they are masked or disabled). The MASK_xx bit overrides the individual alarm mask bits in the Alarm Mask Registers. MASK_xx = 1 on device reset.	Both
	[2]	SWRST_xx		Transmit and Receive Software Reset Bit, Channel xx. When SWRST_xx = 1, this bit provides the same function as the hardware reset, except that all configuration register settings are unaltered. This is not a self-clearing bit. Once set, this bit must be manually set and cleared. SWRST = 0 on device reset.	Both
	[3:6]	RSVD		Reserved	
	[7]	TESTEN_xx		Transmit and Receive Test Enable Bit, Channel xx. When TESTEN_xx = 1, the transmit and receive sections are placed in test mode. The TestMode_[A:B][4:0] bits (0x3006, 0x30106) specify the particular test, and must also be set. NOTE: When the global test enable bit GTESTEN_[A:B] = 0, the individual channel test enable bits are used to selectively place a channel in test or normal mode. When GTESTEN_[A:B] = 1, all channels are set to test mode regardless of their TESTEN setting. TESTEN_xx = 0 on device reset.	Both

Table 18. SERDES Per-Quad Control Register Descriptions

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
<b>SERDES Per-Quad Control Register (Read/Write) xx = [AA, ...,BD]</b>					
30005 - A 30105 - B	[0]	RSVD	44	Reserved	
	[1]	GMASK_[A:B]		Global Mask. When GMASK_[A:B] = 1, the transmit and receive alarms of all channel in the SERDES quad are prevented from generating an alarm (i.e., they are masked or disabled). The GMASK_[A:B] bit overrides the individual MASK_xx bits. GMASK_[A:B] = 1 on device reset.	Both
	[2]	GSWRST_[A:B]		Software reset bit. The GSWRST_[A:B] bit provides the same function as the hardware reset for the transmit and receive sections of all four channels, except that the device configuration settings are not affected when GSWRST_[A:B] is asserted. This is not a self-clearing bit. Once set, this bit must be manually set and cleared. The GSWRST_[A:B] bit overrides the individual SWRST_xx bits. GSWRST_[A:B] = 0 on device reset.	Both
	[3]	GPWRDNT_[A:B]		Powerdown Transmit Function. When GPWRDNT_[A:B] = 1, sections of the transmit hardware for all four channels are powered down. The GPWRDNT_[A:B] bit overrides the individual PWRDNT_xx bits. GPWRDNT_[A:B] = 0 on device reset.	Both
	[4]	GPWRDNR_[A:B]		Powerdown Receive Function. When GPWRDNR_[A:B] = 1, sections of the receive hardware for all four channels are powered down. The GPWRDNR_[A:B] bit overrides the individual PWRDNR_xx bits. GPWRDNR_[A:B] = 0 on device reset.	Both
	[5]	GTRISTN_[A:B]		Active-low TRISTN function. When GTRISTN = 1, the CMOS output buffers for SERDES A are tristated.	Both
	[6:7]	RSVD		Reserved	
30006 - A 30106 - B		TESTMODE_[A:B]	00	Test mode for SERDES characterization	Both
		RSVD		Reserved	



Table 19. Per-Channel Control Register Descriptions

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30800 - AA 30810 - AB 30820 - AC 30830 - AD  30900 - BA 30910 - BB 30920 - BC 30930 - BD	[0:4]	RSVD	00	Reserved	
	[5]	CELL_ALIGN_ERR_EN_xx		Cell Alignment Error, CELL_ALIGN_ERR = 1 indicates that the internal transmit frame processor did not detect a start of cell indicator when it was expecting a new cell. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
	[6]	TX_URUN_ERR_EN_xx		Transmit Underrun Error, TX_URUN_ERR = 1 indicates an underrun error in the transmit Asynchronous FIFO. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
	[7]	TX_ORUN_ERR_EN_xx		Transmit Overrun Error, TX_ORUN_ERR = 1 indicates an overrun error in the transmit Asynchronous FIFO. The TX FIFO is designed to not overflow since it sends backpressure signal to the FPGA when it cannot accept more cells. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
30801 - AA 30811 - AB 30821 - AC 30831 - AD  30901 - BA 30911 - BB 30921 - BC 30931 - BD	[0]	RSVD	00	Reserved	
	[1]	OOF_EN_xx		Out of Frame, OOF_xx = 1 indicates OOF has been detected in this link. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Both
	[2]	EX_SEQ_ERR_EN_xx		Excessive Sequence Errors, EX_SEQ_ERR = 1 indicates that three consecutive cells containing sequence errors have been detected in this link. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
	[3]	SEQ_ERR_EN_xx		Sequence Error, SEQ_ERR = 1 indicates that a sequence error has been detected for a cell on this link. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
	[4]	CELL_BIP_ERR_EN_xx		'1' = Alarm enabled for CELL_BIP_ERR_xx	Cell
	[5]	B1_ERR_EN_xx		'1' = Alarm enabled for B1_ERR_xx	Both
	[6]	RX_FIFO_OVRUN_EN_xx		'1' = Alarm enabled for RX_FIFO_OVRUN_xx	Cell
	[7]	RDI_EN_xx		'1' = Alarm enabled for RDI_xx	Both

**Table 19. Per-Channel Control Register Descriptions (continued)**

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30802 - AA 30812 - AB 30822 - AC 30832 - AD	[0]	ENABLE_JUST_xx	00	ENABLE_JUST_xx =1 causes the core to interpret pointer bytes for positive or negative justification	SONET
30902 - BA 30912 - BB 30922 - BC 30932 - BD	[1]	FMPU_STR_EN_xx		FMPU_STR_EN_xx = 1 enables a channel for alignment within a multi-channel alignment group	SONET
	[2:3]	FMPU_SYNMODE_xx		“00” - No channel alignment “01” - Twin channel alignment “10” - 4 channel alignment “11 - By-8 alignment	SONET
	[4]	DSCR_INH_xx		Descramble Inhibit, DSCR_INH=1 inhibits descrambling (in the RX direction) and scrambling (in the TX direction).	Both
	[5]	FFRM_EN_xx		Fast Frame Enable, FFRM_EN=1 enables the fast frame mode.	Both
	[6]	AIS_ON_xx		Alarm Indication Signal (control), AIS_ON =1 forces AIS-L insertion.	Both
	[7]	AIS_ON_OOF_xx		Alarm Indication Signal on Out of Frame, AIS_ON_OOF =1 forces AIS-L insertion during OOF =1.	Both

Table 19. Per-Channel Control Register Descriptions (continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30803 - AA 30813 - AB 30823 - AC 30833 - AD  30903 - BA 30913 - BB 30923 - BC 30933 - BD	[0]	BYPASS_ALGN_FIFO_xx	00	Bypass alignment FIFO =1 in RX path in SONET mode. DOUT_xx data is checked off RWCKxx.	SONET
	[1]	RAW_SERDES_MODE_xx		Raw SERDES Mode, RAW_SERDES_MODE =1 bypasses all the SONET and cell functions. Data is sent directly from the SERDES to the FPGA logic in the RX path and is passed directly from the FPGA logic to the SERDES in the TX path.	SERDES Only
	[2]	FORCE_B1_ERR_xx		Force B1 Error, FORCE_B1_ERR =1 forces a Section B1 error (Bit Interleaved Parity Error) in the TX path. Valid only when AUTO_TOH_xx bit is set to 1.	Both
	[3]	FORCE_BIP8_ERR_xx		Force BIP8 Error, FORCE_BIP8_ERR =1 forces cell BIP8 error in the TX path.	Cell
	[4]	FORCE_A1A2_ERR_xx		Force A1A2 Error, FORCE_A1A2_ERR =1 forces an error in A1A2 bytes (framing error) in the TX path. Valid only when the corresponding AUTO_TOH bit is set to 1.	Both
	[5]	FORCE_EX_SEQ_ERR_xx		Force BIIP8 Error, FORCE_BIP8_ERR =1 forces cell BIP8 error in the TX path.	Cell
	[6]	FORCE_SEQ_ERR_xx		Force Sequence Error, FORCE_SEQ_ERR =1 forces one sequence error in the TX path. After this bit is set, the next Link Header byte's sequence number is inverted. The Link Header after the errored Link Header byte will have the correct sequence number	Cell
	[7]	FORCE_RDI_xx		Force Remote Defect Indication, FORCE_RDI =1 forces RDI errors in the TX path. The K2 byte in TOH is set to "00000110. Valid only when AUTO_TOH_xx bit is set to 1.	Cell

Table 19. Per-Channel Control Register Descriptions (continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30804 - AA 30814 - AB 30824 - AC 30834 - AD  30904 - BA 30914 - BB 30924 - BC 30934 - BD	[0]	LCKREFN_xx	00	0 = Lock receiver to reference clock (REFCLK) 1 = Lock receiver to HDINxx data	Both
	[1]	LOOPENB_xx		LOOPENB_xx =1 Enable high-speed internal loopback from TX to RX. Disable the HDOUT buffers.	Both
	[2]	DISABLE_TX_xx		Disable Transmitter, For DISABLE_TX = 1 the TX Link is disabled. The disabled link is ignored by the Output Port Controller (OPC) and internally generated idle cells are transmitted on the link. If the link is disabled during the transmission of a cell on the link, the entire cell is transmitted before the link is declared invalid.	Cell
	[3]	DISABLE_RX_xx		Disable Receiver, DISABLE_RX = 1 disables the RX link for cell processing by the Input Port Controller (IPC). The IPC will not read cells from a link if this bit is set for that link	Cell
	[4]	CELL_BIP_INH_xx		Cell BIP (Check) Inhibit, CELL_BIP_INH = 1 prevents cells from being dropped due to a Cell BIP error, in the RX path. If this bit is not set, then cells will be dropped automatically if a cell bip error is detected by the core. The CELLDROP signal across the core-FPGA interface will be active only if this bit is NOT set.	Cell
	[5]	CELL_SEQ_INH_xx		Cell Sequence (Checking) Inhibit, CELL_SEQ_INH = 1 prevents cells in the RX path from being dropped due to a sequence error. If this bit is not set, then cells will be dropped automatically if a sequence error is detected internally. The CELLDROP signal across the core-FPGA interface will be active only if this bit is NOT set.	Cell
	[6]	AUTO_TOH_xx		Automatic TOH Generation, AUTO_TOH_xx =1 enables the TX core to automatically generate TOH bytes. All the FORCE_* register bits are valid only if this bit is set. This bit should be set to 1 in Cell Mode. It can be set to 1 or 0 in SONET Mode. If this bit is not set, then user has to provide all the TOH bytes.	SONET
	[7]	FMPU_RESYNC1_xx		Single channel alignment FIFO reset. Rising edge sensitive. Write a 0 and then a 1 to enable this bit. When enabled, the read pointer in the alignment FIFO is reset to the middle of the FIFO. This bit is valid only when FMPU_SYNMODE_xx = 00 (no multi channel alignment)	SONET

Table 19. Per-Channel Control Register Descriptions (continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30805 - AA 30815 - AB 30825 - AC 30835 - AD  30905 - BA 30915 - BB 30925 - BC 30935 - BD	[0:7]	LINK_NUM_TX_xx	00	Transmit Link Number, This value is transmitted in the "F1" byte of the TOH. This value is used to verify that the links are connected properly.	Both

**Table 20. Per-Channel Status Register Descriptions**

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
<b>Channel Status Registers (Read Only) xx = [AA, ...,BD]</b>					
	[0:4]	RSVD		Reserved	
30808 - AA 30818 - AB 30828 - AC 30838 - AD	[5]	CELL_ALIGN_ERR_xx	00	Cell Alignment Error, CELL_ALIGN_ERR = 1 indicates that the internal transmit frame processor did not detect a start of cell indicator when it was expecting a new cell. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
30908 - BA 30918 - BB 30928 - BC 30938 - BD	[6]	TX_URUN_ERR_xx		Transmit Underrun Error, TX_URUN_ERR = 1 indicates an underrun error in the transmit Asynchronous FIFO. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
	[7]	TX_ORUN_ERR_xx		Transmit Overrun Error, TX_ORUN_ERR = 1 indicates an overrun error in the transmit Asynchronous FIFO. The TX FIFO is designed to not overflow since it sends backpressure signal to the FPGA when it cannot accept more cells. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell

**Table 20. Per-Channel Status Register Descriptions (continued)**

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30809 - AA 30819 - AB 30829 - AC 30839 - AD  30909 - BA 30919 - BB 30929 - BC 30939 - BD	[0]	RSVD	00	Reserved	
	[1]	OOF_xx		OOF_xx = 1 indicates OOF has been detected in this link. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm	Both
	[2]	EX_SEQ_ERR_xx		Excessive Sequence Errors, EX_SEQ_ERR = 1 indicates that three consecutive cells containing sequence errors have been detected in this link. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
	[3]	SEQ_ERR_xx		Sequence Error, SEQ_ERR = 1 indicates that a sequence error has been detected for a cell on this link. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
	[4]	CELL_BIP_ERR_xx		Cell Mode BIP Error, CELL_BIP_ERR = 1 indicates that a BIP error has been detected in a cell on the link. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
	[5]	B1_ERR_xx		Bit Interleaved Parity Error, B1_ERR = 1 indicates that a Section B1 error has been detected on the link. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Both
	[6]	RX_FIFO_OVRUN_xx		Receive FIFO Overrun, RX_FIFO_OVRUN_xx = 1 indicates that the asynchronous RX FIFO has detected an overrun condition. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm.	Cell
	[7]	RDI_xx		Remote Defect Indication, RDI = 1 indicates that a RDI has been detected on the link. If the corresponding alarm enable bit has been set, a 1 on this bit will cause an alarm	Both

**Table 20. Per-Channel Status Register Descriptions (continued))**

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
3080A - AA 3081A - AB 3082A - AC 3083A - AD  3090A - BA 3091A - BB 3092A - BC 3093A - BD	[0:4]	RSVD	00	Reserved	
	[5]	STAT_CELL_ALIGN_ERR_xx		STAT_CELL_ALIGN_ERR_xx = 1 same as CELL_ALIGN_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[6]	STAT_TX_URUN_ERR_xx		STAT_TX_URUN_ERR_xx = 1 same as TX_URUN_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[7]	STAT_TX_ORUN_ERR_xx		STAT_TX_ORUN_ERR_xx = 1 same as TX_ORUN_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Cell
3080B - AA 3081B - AB 3082B - AC 3083B - AD  3090B - BA 3091B - BB 3092B - BC 3093B - BD	[0]	RSVD	00	Reserved	
	[1]	STAT_OOF_xx		STAT_OOF_xx = 1 same as OOF_xx except that a 1 on this bit can NOT cause an interrupt	Both
	[2]	STAT_EX_SEQ_ERR_xx		STAT_EX_SEQ_ERR_xx = 1 same as EX_SEQ_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[3]	STAT_SEQ_ERR_xx		STAT_SEQ_ERR_xx = 1 same as SEQ_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[4]	STAT_CELL_BIP_ERR_xx		STAT_CELL_BIP_ERR_xx = 1 same as CELL_BIP_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[5]	STAT_B1_ERR_xx		STAT_B1_ERR_xx = 1 same as B1_ERR_xx except that a 1 on this bit can NOT cause an interrupt	Both
	[6]	STAT_RX_FIFO_OVRUN_xx		STAT_RX_FIFO_OVRUN_xx = 1 same as RX_FIFO_OVRUN_xx except that a 1 on this bit can NOT cause an interrupt	Cell
	[7]	STAT_RDI_xx		STAT_RDI_xx = 1 same as RDI_xx except that a 1 on this bit can NOT cause an interrupt	Both
3080C - AA 3081C - AB 3082C - AC 3083C - AD  3090C - BA 3091C - BB 3092C - BC 3093C - BD	[0:7]	LINK_NUM_RX_xx	00	Link number received that is stored in the F1 byte position of the SONET TOH	Both
3080E - AA 3081E - AB 3082E - AC 3083E AD  3090E - BA 3091E - BB 3092E - BC 3093E - BD	[0:5]	RSVD	00	Reserved	
	[6]	CH248_SYNC_xx		CH248_SYNC_xx = 1 indicates that A1A2 bytes have been detected by link xx for multi-channel alignment	SONET
	[7]	RX_LINK_GOOD_xx		RX_LINK_GOOD_xx = 1 indicates that frame has been acquired and the link has been stable. Goes high at an A1A2 boundary but can go low at any point in a frame due to excessive errors	Cell



**Table 21. Common Control Register Descriptions**

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
<b>Common Control Registers (Read/Write)</b>					
30A00	[0:1]	RCKSELB	00	“00” - Channel BA source for clock RCK78B “01” - Channel BB source for clock RCK78B “10” - Channel BC source for clock RCK78B “11” - Channel BD source for clock RCK78B	Both
	[2:3]	TCKSELB		“00” - Channel BA source for clock TCK78B “01” - Channel BB source for clock TCK78B “10” - Channel BC source for clock TCK78B “11” - Channel BD source for clock TCK78B	Both
	[4:5]	RCKSELA		“00” - Channel AA source for clock RCK78A “01” - Channel AB source for clock RCK78A “10” - Channel AC source for clock RCK78A “11” - Channel AD source for clock RCK78A	Both
	[6:7]	TCKSELA		“00” - Channel AA source for clock TCK78A “01” - Channel AB source for clock TCK78A “10” - Channel AC source for clock TCK78A “11” - Channel AD source for clock TCK78A	Both
30A01	[0:2]	CELL_SIZE	00	Cell Size, Three bits to set cell size. “000” - Cell size is 76 bytes, “001” - Cell size is 80 bytes, “010” - Cell size is 84 bytes, “011” - Cell size is 92 bytes These are the only supported cell sizes.	Cell
	[3:7]	RX_FIFO_MIN		Set Minimum threshold value for alignment FIFO in SONET mode. When the read address for the FIFO is below this value at the time when write address is zero, it indicates that the FIFO is near overflow. This event will go high only once during a frame when a framing byte has been detected by the aligner. The default threshold value is “00000”.	SONET
30A02	0	TX_DISABLE_ON_RDI	01	Transmitter Disable on RDI (Detection), If TX_DISABLE_ON_RDI = 1 - No cell data is transmitted on a link in which a RDI has been detected by the corresponding link’s receiver. If this bit is set to 0, cell data will be transmitted on a link irrespective of detection of a RDI.	Cell
	[1:3]	DEVICE_MODE		These bits should always be set to “001”	Cell
	4	SCHAR_ENA		SCHAR_ENA = 1 enables SERDES characterization of SERDES B. Refer to section	Both
	5	SCHAR_TXSEL		SCHAR_TXSEL =1 is a Select Tx option which will cause chip ports to directly control the SERDES low-speed transmit ports of one of the channels selected by SCHAR_CHAN	Both
	6:7	SCHAR_CHAN		“00” - Select channel BA to test “01” - Select channel BB to test “10” - Select channel BC to test “11” - Select channel BD to test	Both

Table 21. Common Control Register Descriptions (continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30A03	[0]	NO_TX_RDI_EXSEQ	00	Not Transmission of RDI, If NO_TX_RDI_EXSEQ = 1, a transmit link will not send data if its corresponding receive link is not good due to excessive sequence errors. If this bit is set to 0, a transmit link will still send data even if its corresponding receive link has excessive sequence errors. This bit should always be set during simulation.	Cell
	[1]	AUTO_BUNDLE		Automatic (Link) Bundle, AUTO_BUNDLE = 1 allows a link within a link group to remain active even when another link within that group is defective. Cell data from all links within that group will continue to be sent to the FPGA. If this bit is set to 0, then all links within a link group must be good before cell data are read from the links by the IPC and passed to the FPGA.	Cell
	[2]	RSVD		Reserved	
	[3]	AUTO_REJOIN_A		Automatic (Link) Rejoin, AUTO_REJOIN = 1 forces any link in a SERDES quad to reassert a "RX link good" signal automatically when three consecutive sequence numbers are correct on that link.	Cell
	[4]	AUTO_REMOVE_A		Automatic (Link) Remove, AUTO_REMOVE = 1 indicates that any link in a SERDES quad which sees three excessive sequence errors should deassert the "RX link good" signal which will cause the link to be inactive.	Cell
	[5]	RSVD		Reserved	
	[6]	AUTO_REJOIN_B		Automatic (Link) Rejoin, AUTO_REJOIN = 1 forces any link in a SERDES quad to reassert a "RX link good" signal automatically when three consecutive sequence numbers are correct on that link.	Cell
	[7]	AUTO_REMOVE_B		AUTO_REMOVE_B = 1 indicates that any link in SERDES quad B which sees three excessive sequence errors should deassert the "RX link good" signal which will cause the link to be inactive.	Cell

Table 21. Common Control Register Descriptions (continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30A04	[0]	RSVD	00	Reserved	
	[1]	FMPU_RESYNC4_B		Control to resync 4 channels which have been configured for multi channel alignment in SONET mode in quad B. Requires a rising edge on this bit. Write a 0 followed by a 1.	SONET
	[2]	FMPU_RESYNC2_B2		Control to resync channels BC and BD which have been configured for multi channel alignment in SONET mode in quad B. Requires a rising edge on this bit. Write a 0 followed by a 1.	SONET
	[3]	FMPU_RESYNC2_B1		Control to resync channels BA and BB which have been configured for multi channel alignment in SONET mode in quad B. Requires a rising edge on this bit. Write a 0 followed by a 1.	SONET
	[4]	FMPU_RESYNC4_A		Control to resync 4 channels which have been configured for multi channel alignment in SONET mode in quad A. Requires a rising edge on this bit. Write a 0 followed by a 1.	SONET
	[5]	FMPU_RESYNC2_A2		Control to resync channels AC and AD which have been configured for multi channel alignment in SONET mode in quad A. Requires a rising edge on this bit. Write a 0 followed by a 1.	SONET
	[6]	FMPU_RESYNC2_A1		Control to resync channels AA and AB which have been configured for multi channel alignment in SONET mode in quad A. Requires a rising edge on this bit. Write a 0 followed by a 1.	SONET
	[7]	FMPU_RESYNC8		Control to resync all 8 channels which have been configured for multi channel alignment.	SONET

Table 21. Common Control Register Descriptions (continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30A05	[0:2]	ERRCNT_CHSEL	00	Error Count Channel Select, Control bits to select which channel's Section B1 error and Cell BIP error counts are recorded by the BIP_ERR_CNT and CELL_BIP_ERR_CNT registers. "000" - Channel AA, "001" - Channel AB, "010" - Channel AC, "011" - Channel AD, "100" - Channel BA, "101" - Channel BB, "110" - Channel BC, "111" - Channel BD	Both
	[3]	CELL_MODE_B2		Cell Mode Enable, CELL_MODE_B2 = 1 enables Cell Mode for the channel group BC and BD.	Cell
	[4]	CELL_MODE_B1		Cell Mode Enable, CELL_MODE_B1 = 1 enables Cell Mode for the channel group BB and BA.	Cell
	[5]	CELL_MODE_A2		Cell Mode Enable, CELL_MODE_A2 = 1 enables Cell Mode for the channel group AC and AD.	Cell
	[6]	CELL_MODE_A1		Cell Mode Enable, CELL_MODE = 1 enables Cell Mode for the channel group AB and AA.	Cell
	[7]	CELL_MODE_8		Cell Mode Enable, CELL_MODE_8 =1 enables cell mode for grouping all 8 channels within a group. If this bit is set then the CELL_MODE_[A1,A2,B1,B2] bits are not valid	Cell
30A06	[0:4]	RSVD	00	Reserved	
	[5:6]	RESET_PHASE		Reset Phase, Two bits to select delay phase for delaying the soft reset bit SOFT_RESET with respect to the synchronizing clock. Four delay phases can be selected through the values "00", "01", "10" and "11".	Both
	[7]	SOFT_RESET		Soft Reset, SOFT_RESET=1 resets the embedded core flip flops except for the software registers. This bit does not affect the state of the registers inside the SERDES quads.	Both
30A07	[0:6]	RSVD	00	Reserved	
	[7]	TX_CFG_DONE		Transmitter Configuration Done, Edge sensitive bit to indicate that all TX configuration bits are set. After all register bits have been set for Transmit direction, write a 0 and then a 1 to this bit.	Cell

Table 22. Common Status Register Descriptions

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
<b>Common Status Registers (Read Only)</b>					
30A08	[0]	ALARM_STATUS_BD	00	Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel BD. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[1]	ALARM_STATUS_BC		Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel BC. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[2]	ALARM_STATUS_BB		Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel BB. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[3]	ALARM_STATUS_BA		Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel BA. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[4]	ALARM_STATUS_AD		Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel AD. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[5]	ALARM_STATUS_AC		Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel AC. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both
	[6]	ALARM_STATUS_AB		OR of all alarm status bits for channel AB. A 1 on this bit will set the alarm pin on the system bus interrupt cause register (on the FPGA side)	Both
	[7]	ALARM_STATUS_AA		Alarm Status Indicator, This bit is an OR of all alarm status bits for the channel AA. A 1 on this bit will also set the alarm pin on the system bus interrupt cause register (on the FPGA side).	Both

**Table 22. Common Control Register Descriptions (continued)**

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30A09	[0:7]	B1_ERR_CNT	00	Error counter that increments when a section B1 error is detected on a link. The link is selected using ERRCNT_CHSEL. This counter is cleared on read.	Both
30A0A	[0:7]	CELL_BIP_ERR_CNT	00	Cell BIP Error Counter, Error counter that increments when a Cell BIP error is detected on a link. The link being monitored is selected using ERRCNT_CHSEL. This counter is cleared on read.	Cell
30A0B	[0:2]	RSVD	00	Reserved	
	[3]	CELL_DRP_B2		Cell Drop, CELL_DRP_B2 = 1 indicates that a cell has been dropped from the link group BC and BD	Cell
	[4]	CELL_DRP_B1		Cell Drop, CELL_DRP_B1 = 1 indicates that a cell has been dropped from the link group BB and BA	Cell
	[5]	CELL_DRP_A2		Cell Drop, CELL_DRP_A2 = 1 indicates that a cell has been dropped from the link group AC and AD	Cell
	[6]	CELL_DRP_A1		Cell Drop, CELL_DRP_A1 = 1 indicates that a cell has been dropped from the link group AB and AA	Cell
	[7]	CELL_DRP_ALL8		CELL_DRP_ALL8 = 1 indicates that cells have been dropped on link group comprising of all 8 channels	Cell
30A0C	[0]	RSVD		00	Reserved
	[1]	SYNC4_B_OOS	SYNC4_B_OOS = 1 indicates that channels are not aligned within the 4 links in quad B in SONET mode		SONET
	[2]	SYNC2_B2_OOS	SYNC2_B2_OOS = 1 indicates that channels are not aligned within the links BC and BD in SONET mode		SONET
	[3]	SYNC2_B1_OOS	SYNC2_B1_OOS = 1 indicates that channels are not aligned within the links BB and BA in SONET mode		SONET
	[4]	SYNC4_A_OOS	SYNC4_A_OOS = 1 indicates that channels are not aligned within the 4 links in quad A in SONET mode		SONET
	[5]	SYNC2_A2_OOS	SYNC2_A2_OOS = 1 indicates that channels are not aligned within the AC and AD links in SONET mode		SONET
	[6]	SYNC2_A1_OOS	SYNC2_A1_OOS = 1 indicates that channels are not aligned within the AB and AD links in SONET mode		SONET
	[7]	SYNC8_OOS	SYNC8_OOS = 1 indicates that channels are not aligned within the 8 channels in SONET mode		SONET

Table 22. Common Status Register Descriptions (continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description	Mode
30A0D	[0]	RSVD	00	Reserved	
	[1]	SYNC4_B_OVFL		SYNC8_OOS = 1 indicates that the alignment FIFO(s) in the links in quad B are near overflow (i.e., at the time of writing into address 0, the read address was less than RX_FIFO_MIN)	SONET
	[2]	SYNC2_B2_OVFL		SYNC2_B2_OVFL = 1 indicates that the alignment FIFO(s) in the links BC and BD are near overflow (i.e., at the time of writing into address 0, the read address was less than RX_FIFO_MIN)	SONET
	[3]	SYNC2_B1_OVFL		SYNC2_B1_OVFL = 1 indicates that the alignment FIFO(s) in the links BA and BB are near overflow (i.e., at the time of writing into address 0, the read address was less than RX_FIFO_MIN)	SONET
	[4]	SYNC4_A_OVFL		SYNC4_A_OVFL = 1 indicates that the alignment FIFO(s) in the links in quad A are near overflow (i.e., at the time of writing into address 0, the read address was less than RX_FIFO_MIN)	SONET
	[5]	SYNC2_A2_OVFL		SYNC2_A2_OVFL = 1 indicates that the alignment FIFO(s) in the links AC and AD are near overflow (i.e., at the time of writing into address 0, the read address was less than RX_FIFO_MIN)	SONET
	[6]	SYNC2_A1_OVFL		SYNC2_A1_OVFL = 1 indicates that the alignment FIFO(s) in the links AA and AB are near overflow (i.e., at the time of writing into address 0, the read address was less than RX_FIFO_MIN)	SONET
	[7]	SYNC8_OVFL		SYNC8_OVFL = 1 Indicates that the alignment FIFO(s) in eight-links are near overflow (At the time of writing into address 0, the read address was less than RX_FIFO_MIN)	SONET
30A0E	[0:2]	RSVD	00	Reserved	
	[3]	BDL_ALIGN_ERR_B2		Alignment Error, BDL_ALIGN_ERR = 1 indicates that an alignment error has occurred in the link group pairs BC and BD	Cell
	[4]	BDL_ALIGN_ERR_B1		Alignment Error, BDL_ALIGN_ERR = 1 indicates that an alignment error has occurred in the link group pairs BA and BB	Cell
	[5]	BDL_ALIGN_ERR_A2		Alignment Error, BDL_ALIGN_ERR = 1 indicates that an alignment error has occurred in the link group pairs AC and AD	Cell
	[6]	BDL_ALIGN_ERR_A1		Alignment Error, BDL_ALIGN_ERR = 1 indicates that an alignment error has occurred in the link group pairs AA and AB	Cell
	[7]	BDL_ALIGN_ERR_ALL8		BDL_ALIGN_ERR_ALL8 = 1 -indicates that an alignment error has occurred in cell group of all eight-links	Cell

## Absolute Maximum Ratings

Stresses in excess of the absolute maximum ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operations sections of this data sheet. Exposure to absolute maximum ratings for extended periods can adversely affect device reliability.

The ORCA Series 4 FPSCs include circuitry designed to protect the chips from damaging substrate injection currents and to prevent accumulations of static charge. Nevertheless, conventional precautions should be observed during storage, handling, and use to avoid exposure to excessive electrical stress.

**Table 23. Absolute Maximum Ratings**

Parameter	Symbol	Min	Max	Unit
Storage Temperature	T <sub>stg</sub>	- 65	150	°C
Power Supply Voltage with Respect to Ground	VDD33	- 0.3	4.2	V
	VDDIO	- 0.3	4.2	V
	VDD15	- 0.3	2.0	V
Input Signal with Respect to Ground	V <sub>IN</sub>	VSS - 0.3	VDDIO + 0.3	V
Signal Applied to High-impedance Output	—	VSS - 0.3	VDDIO + 0.3	V
Maximum Package Body (Soldering) Temperature	—	—	220	°C

## Recommended Operating Conditions

**Table 24. Recommended Operating Conditions**

Parameter	Symbol	Min	Max	Unit
Power Supply Voltage with Respect to Ground*	VDD33	3.0	3.6	V
	VDD15	1.425	1.575	V
Input Voltages	V <sub>IN</sub>	VSS - 0.3	VDDIO + 0.3	V
Junction Temperature†	T <sub>J</sub>	- 40	125	°C
SERDES VDD15 Supply Voltage (VDDRx, VDDTx, VDDAUX, VDDGB)		1.425	1.575	V
SERDES CML I/O Supply Voltage (VDDIB, VDDOB)		1.425	1.890	V

\* For FPGA Recommended Operating Conditions and Electrical Characteristics, see Recommended Operating Conditions and Electrical Characteristics tables in the *Series 4 FPGA Data Sheet* and the *Series 4 I/O Buffer Application Note*. FPSC Standby Currents (IDDSB15 and IDDSB33) are tested with the Embedded Core in the powered down state

† Designed for greater than 10 year electromigration life at 3.125 Gbits/s at 100 °C junction temperature.

Note: VDDIB is the center tap of the CML input buffer. In some cases this signal may be left floating, or tied to another voltage level when not interfacing to CML output buffers. See the *SERDES CML Buffer Interface Technical Note* for details.

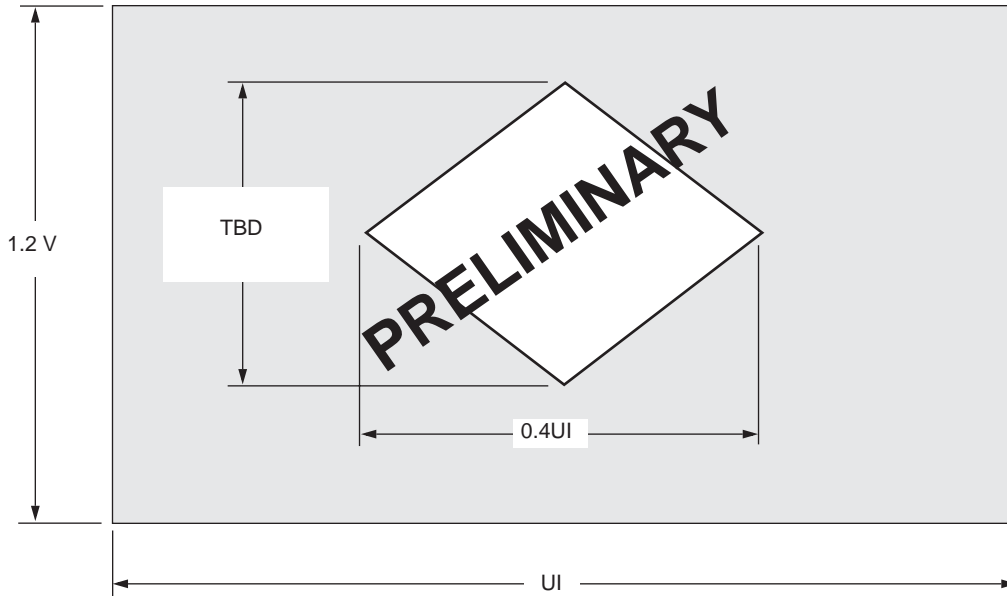
## SERDES Electrical and Timing Characteristics

**Table 25. Maximum Power Dissipation**

Parameter	Conditions	Min	Typ	Max	Unit
Power Dissipation	SERDES, MUX/DEMUX, Align FIFO, and I/O (per channel) 1.25 Gbits/s	—	—	195	mW
	SERDES, MUX/DEMUX, Align FIFO, and I/O (per channel) 2.5 Gbits/s			225	mW
	Scrambler and Cell Processing Logic (per channel)	—	—	50	mW



Figure 48. Receive Data Eye-Diagram Template (Differential)



2391(F)

Figure 48 provides a graphical characterization of the SERDES receiver input requirements. It provides guidance on a number of input parameters, including signal amplitude and rise time limits, noise and jitter limits, and P and N input skew tolerance. It is believed that incoming data patterns falling within the shaded region of the template will be received without error (BER < 10E-12), over all specified operating conditions.

Data pattern eye-opening at the receive end of a link is considered the ultimate measure of received signal quality. Almost all detrimental characteristics of transmit signal and the interconnection link design result in eye-closure. This combined with the eye-opening limitations of the line receiver can provide a good indication of a link's ability to transfer data error-free.

Signal jitter is of special interest to system designers. It is often the primary limiting characteristic of long digital links and of systems with high noise level environments. An interesting characteristic of the Clock and Data Recovery (CDR) portion of the ORSO82G5 SERDES receiver is its ability to filter incoming signal jitter that is below the clock recovery PLL bandwidth (estimated to be about 3 MHz). For signals with high levels of low frequency jitter the receiver can detect incoming data, error-free, with eye-openings significantly less than that of Figure 48. This phenomena has been observed in the laboratory.

Eye-diagram measurement and simulation are excellent tools of design. They are both highly recommended when designing serial link interconnections and evaluating signal integrity.

Table 26. Receiver Specifications

Parameter	Conditions	Min	Typ	Max	Unit
<b>Input Data</b>					
Stream of Nontransitions	—	—	—	72	Bits
Eye Opening Interval	—	0.4	—	—	UI-P
Eye Opening Voltage	—	200	—	—	mVP-P

**Table 27. Reference Clock Specifications (REFINP and REFINN)**

Parameter	Min	Typ	Max	Unit
Frequency Range	125	—	168.75	MHz
Frequency Tolerance	– 100	—	100	ppm
Duty Cycle (Measured at 50% Amplitude Point)	40	50	60	%
Rise Time	—	500	1000	ps
Fall Time	—	500	1000	ps
P–N Input Skew	—	—	75	ps
Differential Amplitude	500	800	2 x VDD	mV <sub>p-p</sub>
Common Mode Level	V <sub>single-ended/2</sub>	0.75	VDD15 – (V <sub>single-ended/2</sub> )	V
Single-Ended Amplitude	250	400	VDD15	mV <sub>p-p</sub>
Input Capacitance (at REFINP)	—	—	5	pF
Input Capacitance (at REFINPIT)	—	—	3	pF
Inband (< 10 MHz) Jitter (2.5 Gbits/s)	—	—	30	pSp-p
Inband (< 10 MHz) Jitter (1.25 Gbits/s)	—	—	60	pSp-p

Note: Additional (<10 MHz) REFCLK jitter will increase the total transmit output jitter.

**Table 28. Channel Output Jitter (1.25 Gbits/s)**

Parameter	Min	Typ	Max	Unit
Deterministic	—	—	0.08	UI <sub>p-p</sub>
Random	—	—	0.12	UI <sub>p-p</sub>
Total	—	—	0.20	UI <sub>p-p</sub>

**Table 29. Channel Output Jitter (2.5 Gbits/s)**

Parameter	Min	Typ	Max	Unit
Deterministic	—	—	0.10	UI <sub>p-p</sub>
Random	—	—	0.14	UI <sub>p-p</sub>
Total	—	—	0.24	UI <sub>p-p</sub>

**Table 30. Serial Output Timing and Levels (CML I/O)**

Parameter	Min	Typ	Max	Unit
Rise Time (20%—80%)	50	80	110	ps
Fall Time (80%—20%)	50	80	110	ps
Common Mode	VDDOB –0.30	VDDOB –0.25	VDDOB –0.15	V
Differential Swing (Full Amplitude)	800	900	1100	mV <sub>p-p</sub>
Differential Swing (Half Amplitude)	400	500	600	mV <sub>p-p</sub>
Output Load	—	50	—	Ω

Note: Differential swings are based on direct CML to CML connections.

**Table 31. Serial Input Timing and Levels (CML I/O)**

Parameter	Min	Typ	Max	Unit
Rise Time (See Eye Diagram in Table 48)	—	—	—	ps
Fall Time (See Eye Diagram in Table 48)	—	—	—	ps
Differential Swing	200	—	—	mV <sub>p-p</sub>
Common-mode Level	0.5	—	VDD15	V
Internal Buffer Resistance (Each input to VDDIB)	40	50	60	Ω

## Pin Information

### Pin Descriptions

This section describes the pins found on the Series 4 FPGAs. Any pin not described in this table is a user-programmable I/O. During configuration, the user-programmable I/Os are 3-stated with an internal pull-up resistor. If any pin is not used (or not bonded to a package pin), it is also 3-stated with an internal pull-up resistor after configuration. The pin descriptions in Table 32. and throughout this data sheet show active-low signals with an overscore. The package pinout tables that follow, show this as a signal ending with \_N. For example  $\overline{\text{LDC}}$  and LDC\_N are equivalent.

**Table 32. Pin Descriptions**

Symbol	I/O	Description
<b>Dedicated Pins</b>		
VDD33	—	3.3 V positive power supply. This power supply is used for 3.3 V configuration RAMs and internal PLLs. When using PLLs, this power supply should be well isolated from all other power supplies on the board for proper operation.
VDD15	—	1.5 V positive power supply for internal logic.
VDDIO	—	Positive power supply used by I/O banks.
VSS	—	Ground.
PTEMP	I	Temperature sensing diode pin. Dedicated input.
$\overline{\text{RESET}}$	I	During configuration, $\overline{\text{RESET}}$ forces the restart of configuration and a pull-up is enabled. After configuration, $\overline{\text{RESET}}$ can be used as a general FPGA input or as a direct input, which causes all PLC latches/FFs to be asynchronously set/reset.
CCLK	O	In the master and asynchronous peripheral modes, CCLK is an output which strobes configuration data in.
	I	In the slave or readback after configuration, CCLK is input synchronous with the data on DIN or D[7:0]. CCLK is an output for daisy-chain operation when the lead device is in master, peripheral, or system bus modes.
DONE	I	As an input, a low level on DONE delays FPGA start-up after configuration.*
	O	As an active-high, open-drain output, a high level on this signal indicates that configuration is complete. DONE has an optional pull-up resistor.
$\overline{\text{PRGM}}$	I	$\overline{\text{PRGM}}$ is an active-low input that forces the restart of configuration and resets the boundary-scan circuitry. This pin always has an active pull-up.
$\overline{\text{RD\_CFG}}$	I	This pin must be held high during device initialization until the $\overline{\text{INIT}}$ pin goes high. This pin always has an active pull-up. During configuration, $\overline{\text{RD\_CFG}}$ is an active-low input that activates the TS_ALL function and 3-states all of the I/O. After configuration, $\overline{\text{RD\_CFG}}$ can be selected (via a bit stream option) to activate the TS_ALL function as described above, or, if readback is enabled via a bit stream option, a high-to-low transition on $\overline{\text{RD\_CFG}}$ will initiate readback of the configuration data, including PFU output states, starting with frame address 0.
RD_DATA/TDO	O	RD_DATA/TDO is a dual-function pin. If used for readback, RD_DATA provides configuration data out. If used in boundary-scan, TDO is test data out.
$\overline{\text{CFG\_IRQ}}/\overline{\text{MPI\_IRQ}}$	O	During JTAG, slave, master, and asynchronous peripheral configuration assertion on this $\overline{\text{CFG\_IRQ}}$ (active-low) indicates an error or errors for block RAM or FPSC initialization. MPI active-low interrupt request output, when the MPI is used.

\* The FPGA States of Operation section contains more information on how to control these signals during start-up. The timing of DONE release is controlled by one set of bit stream options, and the timing of the simultaneous release of all other configuration pins (and the activation of all user I/Os) is controlled by a second set of options.

Symbol	I/O	Description
<b>Special-Purpose Pins</b>		
M[3:0]	I	During powerup and initialization, M0—M3 are used to select the configuration mode with their values latched on the rising edge of $\overline{\text{INIT}}$ . During configuration, a pull-up is enabled.
	I/O	After configuration, these pins are user-programmable I/O.*
PLL_CK[0:7][TC]	I	Semi-dedicated PLL clock pins. During configuration they are 3-stated with a pull up.
	I/O	These pins are user-programmable I/O pins if not used by PLLs after configuration.
P[TBLR]CLK[1:0][TC]	I	Pins dedicated for the primary clock. Input pins on the middle of each side with differential pairing.
	I/O	After configuration these pins are user programmable I/O, if not used for clock inputs.
TDI, TCK, TMS	I	If boundary-scan is used, these pins are test data in, test clock, and test mode select inputs. If boundary-scan is not selected, all boundary-scan functions are inhibited once configuration is complete. Even if boundary-scan is not used, either TCK or TMS must be held at logic 1 during configuration. Each pin has a pull-up enabled during configuration.
	I/O	After configuration, these pins are user-programmable I/O if boundary scan is not used.*
RDY/ $\overline{\text{BUSY}}$ /RCLK	O	During configuration in asynchronous peripheral mode, RDY/RCLK indicates another byte can be written to the FPGA. If a read operation is done when the device is selected, the same status is also available on D7 in asynchronous peripheral mode. During the master parallel configuration mode, RCLK is a read output signal to an external memory. This output is not normally used.
	I/O	After configuration this pin is a user-programmable I/O pin.*
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output, indicating that configuration is not complete.
	I/O	After configuration, this pin is a user-programmable I/O pin.*
$\overline{\text{LDC}}$	O	$\overline{\text{Low During Configuration}}$ is output low until configuration is complete. It is used as a control output, indicating that configuration is not complete.
	I/O	After configuration, this pin is a user-programmable I/O pin.*
$\overline{\text{INIT}}$	I/O	$\overline{\text{INIT}}$ is a bidirectional signal before and during configuration. During configuration, a pull-up is enabled, but an external pull-up resistor is recommended. As an active-low open-drain output, $\overline{\text{INIT}}$ is held low during power stabilization and internal clearing of memory. As an active-low input, $\overline{\text{INIT}}$ holds the FPGA in the wait-state before the start of configuration. After configuration, this pin is a user-programmable I/O pin.*
$\overline{\text{CS0}}$ , CS1	I	$\overline{\text{CS0}}$ and CS1 are used in the asynchronous peripheral, slave parallel, and microprocessor configuration modes. The FPGA is selected when $\overline{\text{CS0}}$ is low and CS1 is high. During configuration, a pull-up is enabled.
	I/O	After configuration, if MPI is not used, these pins are user-programmable I/O pins.*
RD/MPI_STRB	I	$\overline{\text{RD}}$ is used in the asynchronous peripheral configuration mode. A low on $\overline{\text{RD}}$ changes D[7:3] into a status output. $\overline{\text{WR}}$ and $\overline{\text{RD}}$ should not be used simultaneously. If they are, the write strobe overrides. This pin is also used as the MPI data transfer strobe. As a status indication, a high indicates ready, and a low indicates busy.
	I/O	After configuration, if the MPI is not used, this pin is a user-programmable I/O pin.*
$\overline{\text{WR}}$ /MPI_RW	I	$\overline{\text{WR}}$ is used in asynchronous peripheral mode. A low on $\overline{\text{WR}}$ transfers data on D[7:0] to the FPGA. In MPI mode, a high on MPI_RW allows a read from the data bus, while a low causes a write transfer to the FPGA.
	I/O	After configuration, if the MPI is not used, $\overline{\text{WR}}$ /MPI_RW is a user-programmable I/O pin.*

\* The FPGA States of Operation section contains more information on how to control these signals during start-up. The timing of DONE release is controlled by one set of bit stream options, and the timing of the simultaneous release of all other configuration pins (and the activation of all user I/Os) is controlled by a second set of options.

Symbol	I/O	Description
PPC_A[14:31]	I	During MPI mode the PPC_A[14:31] are used as the address bus driven by the <i>PowerPC</i> bus master utilizing the least-significant bits of the <i>PowerPC</i> 32-bit address.
MPI_BURST	I	MPI_BURST is driven low to indicate a burst transfer is in progress in MPI mode. Driven high indicates that the current transfer is not a burst.
MPI_BDIP	I	MPI_BDIP is driven by the <i>PowerPC</i> processor in MPI mode. Assertion of this pin indicates that the second beat in front of the current one is requested by the master. Negated before the burst transfer ends to abort the burst data phase.
MPI_TSZ[0:1]	I	MPI_TSZ[0:1] signals are driven by the bus master in MPI mode to indicate the data transfer size for the transaction. Set 01 for byte, 10 for half-word, and 00 for word.
A[21:0]	O	During master parallel mode A[21:0] address the configuration EPROMs up to 4M bytes.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
MPI_ACK	O	In MPI mode this is driven low indicating the MPI received the data on the write cycle or returned data on a read cycle.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
MPI_CLK	I	This is the <i>PowerPC</i> synchronous, positive-edge bus clock used for the MPI interface. It can be a source of the clock for the Embedded System Bus. If MPI is used this will be the <i>AMBA</i> bus clock.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
MPI_TEA	O	A low on the MPI transfer error acknowledge indicates that the MPI detects a bus error on the internal system bus for the current transaction.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
MPI_RTRY	O	This pin requests the MPC860 to relinquish the bus and retry the cycle.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.*
D[0:31]	I/O	Selectable data bus width from 8, 16, 32-bit in MPI mode. Driven by the bus master in a write transaction and driven by MPI in a read transaction.
	I	D[7:0] receive configuration data during master parallel, peripheral, and slave parallel configuration modes when $\overline{WR}$ is low and each pin has a pull-up enabled. During serial configuration modes, D0 is the DIN input.
	O	D[7:3] output internal status for asynchronous peripheral mode when $\overline{RD}$ is low.
	I/O	After configuration, if MPI is not used, the pins are user-programmable I/O pins.*
DP[0:3]	I/O	Selectable parity bus width in MPI mode from 1, 2, 4-bit, DP[0] for D[0:7], DP[1] for D[8:15], DP[2] for D[16:23], and DP[3] for D[24:31]. After configuration, if MPI is not used, the pins are user-programmable I/O pin.*
	I/O	After configuration, if MPI is not used, the pins are user-programmable I/O pin.*
DIN	I	During slave serial or master serial configuration modes, DIN accepts serial configuration data synchronous with CCLK. During parallel configuration modes, DIN is the D0 input. During configuration, a pull-up is enabled.
	I/O	After configuration, this pin is a user-programmable I/O pin.*
DOUT	O	During configuration, DOUT is the serial data output that can drive the DIN of daisy-chained slave devices. Data out on DOUT changes on the rising edge of CCLK.
	I/O	After configuration, DOUT is a user-programmable I/O pin.*
TESTCFG	I	During configuration this pin should be held high, to allow configuration to occur. A pull up is enabled during configuration.
	I/O	After configuration, TESTCFG is a user programmable I/O pin.*

\* The FPGA States of Operation section contains more information on how to control these signals during start-up. The timing of DONE release is controlled by one set of bit stream options, and the timing of the simultaneous release of all other configuration pins (and the activation of all user I/Os) is controlled by a second set of options.

This section describes device I/O signals to/from the embedded core.

Table 33. FPSC Function Pin Description

Symbol	I/O	Description
<b>Common Signals for Both SERDES Quad A and B</b>		
PASB_RESETN	I	Active low reset for the embedded core. All non-SERDES specific registers (addresses 308***, 309***, 30A***) in the embedded core are not reset.
PASB_TRISTN	I	Active low 3-state for embedded core output buffers. Internally tied high.
PASB_PDN	I	Active low power down of all SERDES blocks and associated I/Os. Internally tied high.
PASB_TESTCLK	I	Clock input for BIST and loopback test. Internally tied high.
PBIST_TEST_ENN	I	Selection of PASB_TESTCLK input for BIST test. Internally tied high.
PLOOP_TEST_ENN	I	Selection of PASB_TESTCLK input for loopback test. Internally tied high.
PMP_TESTCLK	I	Clock input for microprocessor in test mode. Internally tied high.
PMP_TESTCLK_ENN	I	Selection of PMP_TESTCLK in test mode. Internally tied high.
PSYS_DOBISTN	I	Input to start BIST test. Internally tied high.
PSYS_RSSIG_ALL	O	Output result of BIST test.
<b>SERDES Quad A and B Pins</b>		
REFCLKN_A	I	CML reference clock input—SERDES quad A.
REFCLKP_A	I	CML reference clock input—SERDES quad A.
REFCLKN_B	I	CML reference clock input—SERDES quad B.
REFCLKP_B	I	CML reference clock input—SERDES quad B.
REXT_A	—	Reference resistor—SERDES quad A.
REXT_B	—	Reference resistor—SERDES quad B.
REXTN_A	—	Reference resistor—SERDES quad A. A 3.32 K $\Omega \pm 1\%$ resistor must be connected across REXT_A and REXTN_A. This resistor should handle a current of 300 mA.
REXTN_B	—	Reference resistor—SERDES quad B. A 3.32 K $\Omega \pm 1\%$ resistor must be connected across REXT_B and REXTN_B. This resistor should handle a current of 300 mA.
HDINN_AA	I	High-speed CML receive data input—SERDES quad A, channel A.
HDINP_AA	I	High-speed CML receive data input—SERDES quad A, channel A.
HDINN_AB	I	High-speed CML receive data input—SERDES quad A, channel B.
HDINP_AB	I	High-speed CML receive data input—SERDES quad A, channel B.
HDINN_AC	I	High-speed CML receive data input—SERDES quad A, channel C.
HDINP_AC	I	High-speed CML receive data input—SERDES quad A, channel C.
HDINN_AD	I	High-speed CML receive data input—SERDES quad A, channel D.
HDINP_AD	I	High-speed CML receive data input—SERDES quad A, channel D.
HDINN_BA	I	High-speed CML receive data input—SERDES quad B, channel A.
HDINP_BA	I	High-speed CML receive data input—SERDES quad B, channel A.
HDINN_BB	I	High-speed CML receive data input—SERDES quad B, channel B.
HDINP_BB	I	High-speed CML receive data input—SERDES quad B, channel B.
HDINN_BC	I	High-speed CML receive data input—SERDES quad B, channel C.
HDINP_BC	I	High-speed CML receive data input—SERDES quad B, channel C.
HDINN_BD	I	High-speed CML receive data input—SERDES quad B, channel D.
HDINP_BD	I	High-speed CML receive data input—SERDES quad B, channel D.
<b>SERDES quad A and B Pins</b>		
HDOUTN_AA	O	High-speed CML transmit data output—SERDES quad A, channel A.
HDOUTP_AA	O	High-speed CML transmit data output—SERDES quad A, channel A.
HDOUTN_AB	O	High-speed CML transmit data output—SERDES quad A, channel B.

Symbol	I/O	Description
HDOUTP_AB	O	High-speed CML transmit data output—SERDES quad A, channel B.
HDOUTN_AC	O	High-speed CML transmit data output—SERDES quad A, channel C.
HDOUTP_AC	O	High-speed CML transmit data output—SERDES quad A, channel C.
HDOUTN_AD	O	High-speed CML transmit data output—SERDES quad A, channel D.
HDOUTP_AD	O	High-speed CML transmit data output—SERDES quad A, channel D.
HDOUTN_BA	O	High-speed CML transmit data output—SERDES quad B, channel A.
HDOUTP_BA	O	High-speed CML transmit data output—SERDES quad B, channel A.
HDOUTN_BB	O	High-speed CML transmit data output—SERDES quad B, channel B.
HDOUTP_BB	O	High-speed CML transmit data output—SERDES quad B, channel B.
HDOUTN_BC	O	High-speed CML transmit data output—SERDES quad B, channel C.
HDOUTP_BC	O	High-speed CML transmit data output—SERDES quad B, channel C.
HDOUTN_BD	O	High-speed CML transmit data output—SERDES quad B, channel D.
HDOUTP_BD	O	High-speed CML transmit data output—SERDES quad B, channel D.
<b>Power and Ground</b>		
VDDIB_AA	—	1.8 V/1.5 V power supply for high-speed serial input buffers.
VDDIB_AB	—	1.8 V/1.5 V power supply for high-speed serial input buffers.
VDDIB_AC	—	1.8 V/1.5 V power supply for high-speed serial input buffers.
VDDIB_AD	—	1.8 V/1.5 V power supply for high-speed serial input buffers.
VDDIB_BA	—	1.8 V/1.5 V power supply for high-speed serial input buffers.
VDDIB_BB	—	1.8 V/1.5 V power supply for high-speed serial input buffers.
VDDIB_BC	—	1.8 V/1.5 V power supply for high-speed serial input buffers.
VDDIB_BD	—	1.8 V/1.5 V power supply for high-speed serial input buffers.
VDDOB_AA	—	1.8 V/1.5 V power supply for high-speed serial output buffers.
VDDOB_AB	—	1.8 V/1.5 V power supply for high-speed serial output buffers.
VDDOB_AC	—	1.8 V/1.5 V power supply for high-speed serial output buffers.
VDDOB_AD	—	1.8 V/1.5 V power supply for high-speed serial output buffers.
VDDOB_BA	—	1.8 V/1.5 V power supply for high-speed serial output buffers.
VDDOB_BB	—	1.8 V/1.5 V power supply for high-speed serial output buffers.
VDDOB_BC	—	1.8 V/1.5 V power supply for high-speed serial output buffers.
VDDOB_BD	—	1.8 V/1.5 V power supply for high-speed serial output buffers.
VSSRX_AA	—	SERDES analog receive circuitry ground.
VSSRX_AB	—	SERDES analog receive circuitry ground.
VSSRX_AC	—	SERDES analog receive circuitry ground.
VSSRX_AD	—	SERDES analog receive circuitry ground.
VSSRX_BA	—	SERDES analog receive circuitry ground.
VSSRX_BB	—	SERDES analog receive circuitry ground.
VSSRX_BC	—	SERDES analog receive circuitry ground.
VSSRX_BD	—	SERDES analog receive circuitry ground.
VSSGB_A	—	Guard band ground.
VSSGB_B	—	Guard band ground.
VDDGB_A	—	1.5 V guard band power supply.
VDDGB_B	—	1.5 V guard band power supply.
VSSAUX_A	—	SERDES auxiliary circuit ground (no external pin).
VSSAUX_B	—	SERDES auxiliary circuit ground.
VSSIB_AA	—	High-speed input receive buffer ground (no external pin).
VSSIB_AB	—	High-speed input receive buffer ground.

Symbol	I/O	Description
VSSIB_AC	—	High-speed input receive buffer ground.
VSSIB_AD	—	High-speed input receive buffer ground.
VSSIB_BA	—	High-speed input receive buffer ground.
VSSIB_BB	—	High-speed input receive buffer ground.
VSSIB_BC	—	High-speed input receive buffer ground.
VSSIB_BD	—	High-speed input receive buffer ground.
VSSOB_AA	—	High-speed output transmit buffer ground (no external pin).
VSSOB_AB	—	High-speed output transmit buffer ground.
VSSOB_AC	—	High-speed output transmit buffer ground.
VSSOB_AD	—	High-speed output transmit buffer ground.
VSSOB_BA	—	High-speed output transmit buffer ground.
VSSOB_BB	—	High-speed output transmit buffer ground.
VSSOB_BC	—	High-speed output transmit buffer ground.
VSSOB_BD	—	High-speed output transmit buffer ground.
VsSTX_AA	—	SERDES analog transmit circuitry ground (no external pin).
VsSTX_AB	—	SERDES analog transmit circuitry ground.
VsSTX_AC	—	SERDES analog transmit circuitry ground.
VsSTX_AD	—	SERDES analog transmit circuitry ground.
VsSTX_BA	—	SERDES analog transmit circuitry ground.
VsSTX_BB	—	SERDES analog transmit circuitry ground.
VsSTX_BC	—	SERDES analog transmit circuitry ground.
VsSTX_BD	—	SERDES analog transmit circuitry ground.
VDDTX_AA	—	1.5 V Power supply for SERDES analog transmit circuitry.
VDDTX_AB	—	1.5 V Power supply for SERDES analog transmit circuitry (no external pin).
VDDTX_AC	—	1.5 V Power supply for SERDES analog transmit circuitry (no external pin).
VDDTX_AD	—	1.5 V Power supply for SERDES analog transmit circuitry (no external pin).
VDDTX_BA	—	1.5 V Power supply for SERDES analog transmit circuitry.
VDDTX_BB	—	1.5 V Power supply for SERDES analog transmit circuitry (no external pin).
VDDTX_BC	—	1.5 V Power supply for SERDES analog transmit circuitry (no external pin)..
VDDTX_BD	—	1.5 V Power supply for SERDES analog transmit circuitry (no external pin).
VDDR_X_AA	—	1.5 V Power supply for SERDES analog receive circuitry.
VDDR_X_AB	—	1.5 V Power supply for SERDES analog receive circuitry (no external pin).
VDDR_X_AC	—	1.5 V Power supply for SERDES analog receive circuitry (no external pin).
VDDR_X_AD	—	1.5 V Power supply for SERDES analog receive circuitry (no external pin).
VDDR_X_BA	—	1.5 V Power supply for SERDES analog receive circuitry.
VDDR_X_BB	—	1.5 V Power supply for SERDES analog receive circuitry (no external pin).
VDDR_X_BC	—	1.5 V Power supply for SERDES analog receive circuitry (no external pin).
VDDR_X_BD	—	1.5 V Power supply for SERDES analog receive circuitry (no external pin).
VDDAUX_A	—	1.5 V power supply for SERDES auxiliary circuit.
VDDAUX_B	—	1.5 V power supply for SERDES auxiliary circuit.



## Power Supplies for ORSO82G5

### Power Supply Descriptions

In the ORSO82G5 embedded core the TX-RX digital power supplies are used for transmit and receive digital logic including the microprocessor logic. The TX-RX analog power supplies are used for high-speed analog circuitry between the I/O buffers and the digital logic. The RX input buffer power supplies are used to power the input (receive) buffers. The TX output buffer supplies are used to power the output (transmit) buffers. The RX and TX buffer power supplies can be independently set to 1.5 V or 1.8 V, depending on the end application. The auxiliary and guard band supplies are independent connection brought out to pins.

In the ORSO82G5, many of the VDD pads are connected together at the package substrate level and not individually connected to external pins. The same also applies for the VSS pads. Table 34. shows the package connections to the common VSS and VDD groupings.

**Table 34. Power Supply Pin Groupings**

Connections to VSS Group (VSST)		Connections to VDD Group, 1.5 V/1.8 V (VDDR)	
Pad Name	BM680	Pad Name	BM680
VSSTX_BD	H34	VDDTX_BA	K32
VSSTX_BC	H31	VDDTX_AA	L32
VSSTX_BB	AG33	VDDR_X_BA	AD32
VSSTX_BA	AG30	VDDR_X_BA	AE32
VSSTX_AD	J32	VDDR_X_AA	P32
VSSTX_AC	U32	VDDR_X_AA	R32
VSSTX_AB	V32	VDDAUX_B	AA32
VSSOB_BD	H33	VDDAUX_A	Y32
VSSOB_BC	H30		
VSSOB_BB	AG32		
VSSOB_BA	AF32		
VSSOB_AD	M32		
VSSOB_AC	U33		
VSSOB_AB	V33		
VSSIB_BD	H32		
VSSIB_BC	AG34		
VSSIB_BB	AG31		
VSSIB_BA	AC32		
VSSIB_AD	N32		
VSSIB_AC	U34		
VSSIB_AB	V34		
VSSAUX_B	AB32		

---

## Recommended Power Supply Connections

Ideally, a board should have four separate power supplies as described below:

- TX-RX digital auxiliary supplies.

The TX-RX digital and auxiliary power supply nodes should be supplied by a 1.5 V source. A single 1.5 V source can supply power to TX-RX digital and auxiliary nodes.

- TX-RX analog, guardband supplies.

A dedicated 1.5 V power supply should be provided to the analog power pins. This will allow the end user to minimize noise. The guard band pins can also be sourced from the analog power supplies.

- TX output buffers.

The power supplies to the TX output buffers should be isolated from the rest of the board power supplies. Special care must be taken to minimize noise when providing board level power to these output buffers. The power supply can be 1.5 V or 1.8 V depending on the end application.

- RX input buffers.

The power supplies to the RX input buffers should be isolated from the rest of the board power supplies. Special care must be taken to minimize noise when providing board level power to these input buffers. The power supply can be 1.5 V or 1.8 V depending on the end application.

## Recommended Power Supply Filtering Scheme

The board connections of the various SERDES VDD and VSS pins are critical to system performance. An example demonstration board schematic is available at:

<http://www.latticesemi.com>

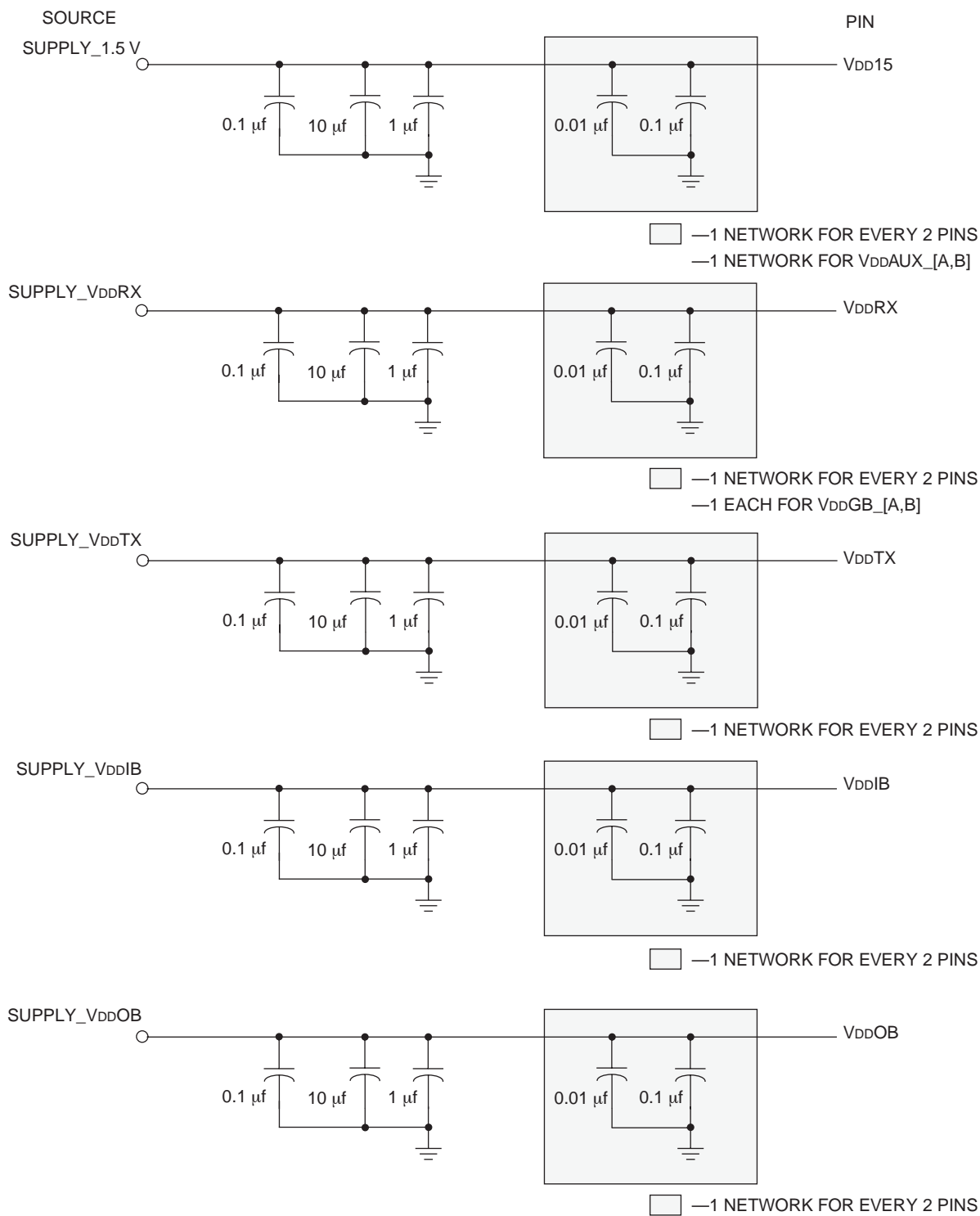
Power supply filtering is in the form of:

- A parallel bypass capacitor network consisting of 10  $\mu\text{f}$ , 0.1  $\mu\text{f}$ , and 1.0  $\mu\text{f}$  caps close to the power source.
- A parallel bypass capacitor network consisting of 0.01  $\mu\text{f}$  and 0.1  $\mu\text{f}$  close to the pin on the ORSO82G5.

Example connections are shown in Figure 49. The naming convention for the power supply sources shown in the figure are as follows:

- Supply\_1.5 V—TX-RX digital, auxiliary power pins.
- Supply\_VDDRX—RX analog power pins, guard band power pins.
- Supply VDDTX—TX analog power pins.
- Supply VDDIB—Input RX buffer power pins.
- Supply\_VDDOB—Output TX buffer power pins.

Figure 49. Power Supply Filtering



## Package Pinouts

Table 35 provides the package pin and pin function for the ORSO82G5 FPSC and packages. The bond pad name is identified in the PIO nomenclature used in the ispLEVER development system design editor. The Bank column provides information as to which output voltage level bank the given pin is in. The Group column provides information as to the group of pins the given pin is in. This is used to show which VREF pin is used to provide the reference voltage for single-ended limited-swing I/Os. If none of these buffer types (such as SSTL, GTL, HSTL) are used in a given group, then the VREF pin is available as an I/O pin.

When the number of FPGA bond pads exceeds the number of package pins, bond pads are unused. When the number of package pins exceeds the number of bond pads, package pins are left unconnected (no connects). When a package pin is to be left as a no connect for a specific die, it is indicated as a note in the device column for the FPGA. The tables provide no information on unused pads.

As shown in the pair columns in Table 35, differential pairs and physical locations are numbered within each bank (e.g., L19C-A0 is the nineteenth pair in an associated bank). A “C” indicates complementary differential, whereas a “T” indicates true differential. An \_A0 indicates the physical location of adjacent balls in either the horizontal or vertical direction. Other physical indicators are as follows:

- \_A1 indicates one ball between pairs.
- \_A2 indicates two balls between pairs.
- \_D0 indicates balls are diagonally adjacent.
- \_D1 indicates balls are diagonally adjacent, separated by one physical ball.

VREF pins, shown in the Pin Description column in Table 35, are associated to the bank and group (e.g., VREF\_TL\_01 is the VREF for group one of the Top Left (TL) bank).

**Table 35. ORSO82G5 680-Pin PBGM Pinout**

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AB20	—	—	Vss	Vss	—	—
C3	—	—	VDD33	VDD33	—	—
E4	—	—	O	PRD_DATA	RD_DATA/TDO	—
F5	—	—	I	PRESET_N	RESET_N	—
G5	—	—	I	PRD_CFG_N	RD_CFG_N	—
D3	—	—	I	PPRGRM_N	PRGRM_N	—
A2	0 (TL)	—	VDDIO0	VDDIO0	—	—
F4	0 (TL)	7	IO	PL2D	PLL_CK0C/HPPLL	L21C_A0
G4	0 (TL)	7	IO	PL2C	PLL_CK0T/HPPLL	L21T_A0
B3	0 (TL)	—	VDDIO0	VDDIO0	—	—
C2	0 (TL)	7	IO	PL3D	—	L22C_D0
B1	0 (TL)	7	IO	PL3C	VREF_0_07	L22T_D0
A1	—	—	Vss	VSS	—	—
J5	0 (TL)	7	IO	PL4D	D5	L23C_A0
H5	0 (TL)	7	IO	PL4C	D6	L23T_A0
B7	0 (TL)	—	VDDIO0	VDDIO0	—	—
E3	0 (TL)	8	IO	PL4B	—	L24C_A0
F3	0 (TL)	8	IO	PL4A	VREF_0_08	L24T_A0
C1	0 (TL)	8	IO	PL5D	HDC	L25C_D0
D2	0 (TL)	8	IO	PL5C	LDC_N	L25T_D0

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
A34	—	—	VSS	VSS	—	—
G3	0 (TL)	8	IO	PL5B	—	L26C_D0
H4	0 (TL)	8	IO	PL5A	—	L26T_D0
E2	0 (TL)	9	IO	PL6D	TESTCFG	L27C_D0
D1	0 (TL)	9	IO	PL6C	D7	L27T_D0
C5	0 (TL)	—	VDDIO0	VDDIO0	—	—
F2	0 (TL)	9	IO	PL7D	VREF_0_09	L28C_D0
E1	0 (TL)	9	IO	PL7C	A17/PPC_A31	L28T_D0
AA13	—	—	VSS	VSS	—	—
J4	0 (TL)	9	IO	PL7B	—	L29C_D0
K5	0 (TL)	9	IO	PL7A	—	L29T_D0
H3	0 (TL)	9	IO	PL8D	CS0_N	L30C_D0
G2	0 (TL)	9	IO	PL8C	CS1	L30T_D0
C9	0 (TL)	—	VDDIO0	VDDIO0	—	—
L5	0 (TL)	9	IO	PL8B	—	L31C_D0
K4	0 (TL)	9	IO	PL8A	—	L31T_D0
H2	0 (TL)	10	IO	PL9D	—	L32C_D0
J3	0 (TL)	10	IO	PL9C	—	L32T_D0
AA14	—	—	VSS	VSS	—	—
M5	0 (TL)	10	IO	PL9B	—	—
F1	0 (TL)	10	IO	PL10D	INIT_N	L33C_A0
G1	0 (TL)	10	IO	PL10C	DOOUT	L33T_A0
K3	0 (TL)	10	IO	PL11D	VREF_0_10	L34C_D0
J2	0 (TL)	10	IO	PL11C	A16/PPC_A30	L34T_D0
AA15	—	—	VSS	VSS	—	—
L4	0 (TL)	10	IO	PL11B	—	—
N5	7 (CL)	1	IO	PL12D	A15/PPC_A29	L1C_D0
M4	7 (CL)	1	IO	PL12C	A14/PPC_A28	L1T_D0
AA3	7 (CL)	—	VDDIO7	VDDIO7	—	—
L3	7 (CL)	1	IO	PL12B	—	L2C_D0
K2	7 (CL)	1	IO	PL12A	—	L2T_D0
H1	7 (CL)	1	IO	PL13D	VREF_7_01	L3C_A0
J1	7 (CL)	1	IO	PL13C	D4	L3T_A0
V18	—	—	VSS	VSS	—	—
N4	7 (CL)	2	IO	PL13B	—	L4C_D0
P5	7 (CL)	2	IO	PL13A	—	L4T_D0
M3	7 (CL)	2	IO	PL14D	RDY/BUSY_N/RCLK	L5C_D0
L2	7 (CL)	2	IO	PL14C	VREF_7_02	L5T_D0
AC2	7 (CL)	—	VDDIO7	VDDIO7	—	—
K1	7 (CL)	2	IO	PL14B	—	L6C_A0
L1	7 (CL)	2	IO	PL14A	—	L6T_A0
P4	7 (CL)	2	IO	PL15D	A13/PPC_A27	L7C_A0
P3	7 (CL)	2	IO	PL15C	A12/PPC_A26	L7T_A0
V19	—	—	VSS	VSS	—	—

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
M2	7 (CL)	2	IO	PL15B	—	L8C_A0
M1	7 (CL)	2	IO	PL15A	—	L8T_A0
N2	7 (CL)	3	IO	PL16D	—	L9C_A0
N1	7 (CL)	3	IO	PL16C	—	L9T_A0
N3	7 (CL)	—	VDDIO7	VDDIO7	—	—
R4	7 (CL)	3	IO	PL16B	—	—
P2	7 (CL)	3	IO	PL17D	A11/PPC_A25	L10C_D0
R3	7 (CL)	3	IO	PL17C	VREF_7_03	L10T_D0
W16	—	—	VSS	VSS	—	—
R5	7 (CL)	3	IO	PL17B	—	—
P1	7 (CL)	3	IO	PL18D	—	L11C_A0
R1	7 (CL)	3	IO	PL18C	—	L11T_A0
T5	7 (CL)	3	IO	PL18B	—	L12C_A0
T4	7 (CL)	3	IO	PL18A	—	L12T_A0
T3	7 (CL)	4	IO	PL19D	RD_N/MPI_STRB_N	L13C_A0
T2	7 (CL)	4	IO	PL19C	VREF_7_04	L13T_A0
W17	—	—	VSS	VSS	—	—
U1	7 (CL)	4	IO	PL19B	—	L14C_A0
T1	7 (CL)	4	IO	PL19A	—	L14T_A0
U4	7 (CL)	4	IO	PL20D	PLCK0C	L15C_A0
U5	7 (CL)	4	IO	PL20C	PLCK0T	L15T_A0
R2	7 (CL)	—	VDDIO7	VDDIO7	—	—
U2	7 (CL)	4	IO	PL20B	—	L16C_D0
V1	7 (CL)	4	IO	PL20A	—	L16T_D0
W18	—	—	VSS	VSS	—	—
V2	7 (CL)	5	IO	PL21D	A10/PPC_A24	L17C_A0
V3	7 (CL)	5	IO	PL21C	A9/PPC_A23	L17T_A0
W19	—	—	VSS	VSS	—	—
V4	7 (CL)	5	IO	PL21B	—	L18C_A0
V5	7 (CL)	5	IO	PL21A	—	L18T_A0
W4	7 (CL)	5	IO	PL22D	A8/PPC_A22	L19C_A0
W3	7 (CL)	5	IO	PL22C	VREF_7_05	L19T_A0
W1	7 (CL)	5	IO	PL22B	—	L20C_A0
Y1	7 (CL)	5	IO	PL22A	—	L20T_A0
Y2	7 (CL)	5	IO	PL23D	—	L21C_D0
AA1	7 (CL)	5	IO	PL23C	—	L21T_D0
Y13	—	—	VSS	VSS	—	—
Y4	7 (CL)	5	IO	PL23B	—	L22C_A0
Y3	7 (CL)	5	IO	PL23A	—	L22T_A0
Y5	7 (CL)	6	IO	PL24D	PLCK1C	L23C_A0
W5	7 (CL)	6	IO	PL24C	PLCK1T	L23T_A0
U3	7 (CL)	—	VDDIO7	VDDIO7	—	—
AB1	7 (CL)	6	IO	PL24B	—	L24C_D0
AA2	7 (CL)	6	IO	PL24A	—	L24T_D0

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AB2	7 (CL)	6	IO	PL25D	VREF_7_06	L25C_D0
AC1	7 (CL)	6	IO	PL25C	A7/PPC_A21	L25T_D0
Y14	—	—	VSS	VSS	—	—
AA4	7 (CL)	6	IO	PL25B	—	—
AB4	7 (CL)	6	IO	PL26D	A6/PPC_A20	L26C_A0
AB3	7 (CL)	6	IO	PL26C	A5/PPC_A19	L26T_A0
W2	7 (CL)	—	VDDIO7	VDDIO7	—	—
AD1	7 (CL)	7	IO	PL26B	—	—
AE1	7 (CL)	7	IO	PL27D	WR_N/MPI_RW	L27C_D0
AD2	7 (CL)	7	IO	PL27C	VREF_7_07	L27T_D0
AC3	7 (CL)	7	IO	PL27B	—	L28C_A0
AC4	7 (CL)	7	IO	PL27A	—	L28T_A0
AF1	7 (CL)	8	IO	PL28D	A4/PPC_A18	L29C_D0
AE2	7 (CL)	8	IO	PL28C	VREF_7_08	L29T_D0
AB5	7 (CL)	8	IO	PL29D	A3/PPC_A17	L30C_A0
AA5	7 (CL)	8	IO	PL29C	A2/PPC_A16	L30T_A0
Y15	—	—	VSS	VSS	—	—
AD3	7 (CL)	8	IO	PL29B	—	—
AG1	7 (CL)	8	IO	PL30D	A1/PPC_A15	L31C_D0
AF2	7 (CL)	8	IO	PL30C	A0/PPC_A14	L31T_D0
AD4	7 (CL)	8	IO	PL30B	—	L32C_D0
AE3	7 (CL)	8	IO	PL30A	—	L32T_D0
AD5	7 (CL)	8	IO	PL31D	DP0	L33C_A0
AC5	7 (CL)	8	IO	PL31C	DP1	L33T_A0
Y20	—	—	VSS	VSS	—	—
AG2	7 (CL)	8	IO	PL31B	—	L34C_D0
AH1	7 (CL)	8	IO	PL31A	—	L34T_D0
AF3	6 (BL)	1	IO	PL32D	D8	L1C_A0
AG3	6 (BL)	1	IO	PL32C	VREF_6_01	L1T_A0
AL7	6 (BL)	—	VDDIO6	VDDIO6	—	—
AE4	6 (BL)	1	IO	PL32B	—	L2C_A0
AF4	6 (BL)	1	IO	PL32A	—	L2T_A0
AE5	6 (BL)	1	IO	PL33D	D9	L3C_A0
AF5	6 (BL)	1	IO	PL33C	D10	L3T_A0
R21	—	—	VSS	VSS	—	—
AJ1	6 (BL)	2	IO	PL34D	—	L4C_D0
AH2	6 (BL)	2	IO	PL34C	VREF_6_02	L4T_D0
AM5	6 (BL)	—	VDDIO6	VDDIO6	—	—
AK1	6 (BL)	2	IO	PL34B	—	L5C_D0
AJ2	6 (BL)	2	IO	PL34A	—	L5T_D0
R22	—	—	VSS	VSS	—	—
AG4	6 (BL)	3	IO	PL35B	D11	L6C_D0
AH3	6 (BL)	3	IO	PL35A	D12	L6T_D0
AL1	6 (BL)	3	IO	PL36D	—	L7C_D0

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AK2	6 (BL)	3	IO	PL36C	—	L7T_D0
AM9	6 (BL)	—	VDDIO6	VDDIO6	—	—
AM1	6 (BL)	3	IO	PL36B	VREF_6_03	L8C_D0
AL2	6 (BL)	3	IO	PL36A	D13	L8T_D0
AJ3	6 (BL)	4	IO	PL37D	—	—
T16	—	—	VSS	VSS	—	—
AJ4	6 (BL)	4	IO	PL37B	—	L9C_A0
AH4	6 (BL)	4	IO	PL37A	VREF_6_04	L9T_A0
AK3	6 (BL)	4	IO	PL38C	—	—
AN2	6 (BL)	—	VDDIO6	VDDIO6	—	—
AG5	6 (BL)	4	IO	PL38B	—	L10C_A0
AH5	6 (BL)	4	IO	PL38A	—	L10T_A0
AN1	6 (BL)	4	IO	PL39D	PLL_CK7C/HPPLL	L11C_D0
AM2	6 (BL)	4	IO	PL39C	PLL_CK7T/HPPLL	L11T_D0
T17	—	—	VSS	VSS	—	—
AL3	6 (BL)	4	IO	PL39B	—	L12C_D0
AK4	6 (BL)	4	IO	PL39A	—	L12T_D0
T18	—	—	VSS	VSS	—	—
AM3	—	—	I	PTEMP	PTEMP	—
AN3	6 (BL)	—	VDDIO6	VDDIO6	—	—
AJ5	—	—	IO	LVDS_R	LVDS_R	—
AL4	—	—	VDD33	VDD33	—	—
T19	—	—	VSS	VSS	—	—
AK5	—	—	VDD33	VDD33	—	—
AM4	6 (BL)	5	IO	PB2A	DP2	L13T_D0
AL5	6 (BL)	5	IO	PB2B	—	L13C_D0
AN7	6 (BL)	—	VDDIO6	VDDIO6	—	—
AP3	6 (BL)	5	IO	PB2C	PLL_CK6T/PPLL	L14T_A0
AP4	6 (BL)	5	IO	PB2D	PLL_CK6C/PPLL	L14C_A0
AN4	6 (BL)	5	IO	PB3B	—	—
U16	—	—	VSS	VSS	—	—
AK6	6 (BL)	5	IO	PB3C	—	L15T_A0
AK7	6 (BL)	5	IO	PB3D	—	L15C_A0
AL6	6 (BL)	5	IO	PB4A	VREF_6_05	L16T_A0
AM6	6 (BL)	5	IO	PB4B	DP3	L16C_A0
AP1	6 (BL)	—	VDDIO6	VDDIO6	—	—
AN5	6 (BL)	6	IO	PB4C	—	L17T_A0
AP5	6 (BL)	6	IO	PB4D	—	L17C_A0
AK8	6 (BL)	6	IO	PB5B	—	—
U17	—	—	VSS	VSS	—	—
AP6	6 (BL)	6	IO	PB5C	VREF_6_06	L18T_D0
AP7	6 (BL)	6	IO	PB5D	D14	L18C_D0
AM7	6 (BL)	6	IO	PB6A	—	L19T_D0
AN6	6 (BL)	6	IO	PB6B	—	L19C_D0



BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AP2	6 (BL)	—	VDDIO6	VDDIO6	—	—
AL8	6 (BL)	7	IO	PB6C	D15	L20T_A0
AL9	6 (BL)	7	IO	PB6D	D16	L20C_A0
AK9	6 (BL)	7	IO	PB7B	—	—
U18	—	—	VSS	VSS	—	—
AN8	6 (BL)	7	IO	PB7C	D17	L21T_A0
AM8	6 (BL)	7	IO	PB7D	D18	L21C_A0
AN9	6 (BL)	7	IO	PB8A	—	L22T_D0
AP8	6 (BL)	7	IO	PB8B	—	L22C_D0
AK10	6 (BL)	7	IO	PB8C	VREF_6_07	L23T_A0
AL10	6 (BL)	7	IO	PB8D	D19	L23C_A0
AP9	6 (BL)	8	IO	PB9B	—	—
U19	—	—	VSS	VSS	—	—
AM10	6 (BL)	8	IO	PB9C	D20	L24T_A0
AM11	6 (BL)	8	IO	PB9D	D21	L24C_A0
AK11	6 (BL)	8	IO	PB10B	—	—
AN10	6 (BL)	8	IO	PB10C	VREF_6_08	L25T_A0
AP10	6 (BL)	8	IO	PB10D	D22	L25C_A0
AN11	6 (BL)	9	IO	PB11A	—	L26T_A0
AP11	6 (BL)	9	IO	PB11B	—	L26C_A0
V16	—	—	VSS	VSS	—	—
AL12	6 (BL)	9	IO	PB11C	D23	L27T_A0
AK12	6 (BL)	9	IO	PB11D	D24	L27C_A0
AN12	6 (BL)	9	IO	PB12A	—	L28T_A0
AM12	6 (BL)	9	IO	PB12B	—	L28C_A0
AP12	6 (BL)	9	IO	PB12C	VREF_6_09	L29T_A0
AP13	6 (BL)	9	IO	PB12D	D25	L29C_A0
AM13	6 (BL)	9	IO	PB13A	—	L30T_D0
AN14	6 (BL)	9	IO	PB13B	—	L30C_D0
V17	—	—	VSS	VSS	—	—
AP14	6 (BL)	10	IO	PB13C	D26	L31T_A0
AP15	6 (BL)	10	IO	PB13D	D27	L31C_A0
AK13	6 (BL)	10	IO	PB14A	—	L32T_A0
AK14	6 (BL)	10	IO	PB14B	—	L32C_A0
AM14	6 (BL)	10	IO	PB14C	VREF_6_10	L33T_A0
AL14	6 (BL)	10	IO	PB14D	D28	L33C_A0
AP17	6 (BL)	11	IO	PB15A	—	L34T_A0
AP16	6 (BL)	11	IO	PB15B	—	L34C_A0
AM15	6 (BL)	11	IO	PB15C	D29	L35T_D0
AN16	6 (BL)	11	IO	PB15D	D30	L35C_D0
AM17	6 (BL)	11	IO	PB16A	—	L36T_A0
AM16	6 (BL)	11	IO	PB16B	—	L36C_A0
AP18	6 (BL)	11	IO	PB16C	VREF_6_11	L37T_A0
AP19	6 (BL)	11	IO	PB16D	D31	L37C_A0

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AL16	5 (BC)	1	IO	PB17A	—	L1T_D0
AK15	5 (BC)	1	IO	PB17B	—	L1C_D0
N22	—	—	VSS	VSS	—	—
AN18	5 (BC)	1	IO	PB17C	—	L2T_A0
AN19	5 (BC)	1	IO	PB17D	—	L2C_A0
AP20	5 (BC)	1	IO	PB18A	—	L3T_A0
AP21	5 (BC)	1	IO	PB18B	—	L3C_A0
AL17	5 (BC)	1	IO	PB18C	VREF_5_01	L4T_D0
AK16	5 (BC)	1	IO	PB18D	—	L4C_D0
P13	—	—	VSS	VSS	—	—
AM19	5 (BC)	2	IO	PB19A	—	L5T_A0
AM18	5 (BC)	2	IO	PB19B	—	L5C_A0
P14	—	—	VSS	VSS	—	—
AN20	5 (BC)	2	IO	PB19C	PBCK0T	L6T_A0
AM20	5 (BC)	2	IO	PB19D	PBCK0C	L6C_A0
AK17	5 (BC)	2	IO	PB20A	—	L7T_D0
AL18	5 (BC)	2	IO	PB20B	—	L7C_D0
AL11	5 (BC)	—	VDDIO5	VDDIO5	—	—
AP22	5 (BC)	2	IO	PB20C	VREF_5_02	L8T_D0
AN21	5 (BC)	2	IO	PB20D	—	L8C_D0
AM22	5 (BC)	2	IO	PB21A	—	L9T_A0
AM21	5 (BC)	2	IO	PB21B	—	L9C_A0
AP23	5 (BC)	3	IO	PB21C	—	L10T_D0
AN22	5 (BC)	3	IO	PB21D	VREF_5_03	L10C_D0
AL19	5 (BC)	3	IO	PB22A	—	L11T_D0
AK18	5 (BC)	3	IO	PB22B	—	L11C_D0
P15	—	—	VSS	VSS	—	—
AP24	5 (BC)	3	IO	PB22C	—	L12T_D0
AN23	5 (BC)	3	IO	PB22D	—	L12C_D0
AP25	5 (BC)	3	IO	PB23A	—	L13T_A0
AP26	5 (BC)	3	IO	PB23B	—	L13C_A0
AL13	5 (BC)	—	VDDIO5	VDDIO5	—	—
AL20	5 (BC)	3	IO	PB23C	PBCK1T	L14T_D0
AK19	5 (BC)	3	IO	PB23D	PBCK1C	L14C_D0
AK20	5 (BC)	3	IO	PB24A	—	L15T_D0
AL21	5 (BC)	3	IO	PB24B	—	L15C_D0
P20	—	—	VSS	VSS	—	—
AN24	5 (BC)	4	IO	PB24C	—	L16T_D0
AM23	5 (BC)	4	IO	PB24D	—	L16C_D0
AN26	5 (BC)	4	IO	PB25A	—	L17T_A0
AN25	5 (BC)	4	IO	PB25B	—	L17C_A0
AL15	5 (BC)	—	VDDIO5	VDDIO5	—	—
AK21	5 (BC)	4	IO	PB25C	—	L18T_D0
AL22	5 (BC)	4	IO	PB25D	VREF_5_04	L18C_D0

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AM24	5 (BC)	4	IO	PB26A	—	L19T_D0
AL23	5 (BC)	4	IO	PB26B	—	L19C_D0
P21	—	—	VSS	VSS	—	—
AP27	5 (BC)	5	IO	PB26C	—	L20T_A0
AN27	5 (BC)	5	IO	PB26D	VREF_5_05	L20C_A0
AL24	5 (BC)	5	IO	PB27A	—	L21T_D0
AM25	5 (BC)	5	IO	PB27B	—	L21C_D0
AN13	5 (BC)	—	VDDIO5	VDDIO5	—	—
AP28	5 (BC)	5	IO	PB27C	—	L22T_A0
AP29	5 (BC)	5	IO	PB27D	—	L22C_A0
AN29	5 (BC)	6	IO	PB28B	—	—
P22	—	—	VSS	VSS	—	—
AM27	5 (BC)	6	IO	PB28C	—	L23T_D0
AN28	5 (BC)	6	IO	PB28D	VREF_5_06	L23C_D0
AM26	5 (BC)	6	IO	PB29B	—	—
AK22	5 (BC)	6	IO	PB29C	—	L24T_A0
AK23	5 (BC)	6	IO	PB29D	—	L24C_A0
AL25	5 (BC)	7	IO	PB30B	—	—
R13	—	—	VSS	VSS	—	—
AP30	5 (BC)	7	IO	PB30C	—	L25T_A0
AP31	5 (BC)	7	IO	PB30D	—	L25C_A0
AK24	5 (BC)	7	IO	PB31B	—	—
AN15	5 (BC)	—	VDDIO5	VDDIO5	—	—
AM29	5 (BC)	7	IO	PB31C	VREF_5_07	L26T_A0
AM28	5 (BC)	7	IO	PB31D	—	L26C_A0
AN30	5 (BC)	7	IO	PB32B	—	—
R14	—	—	VSS	VSS	—	—
AK25	5 (BC)	7	IO	PB32C	—	L27T_D0
AL26	5 (BC)	7	IO	PB32D	—	L27C_D0
AN17	5 (BC)	—	VDDIO5	VDDIO5	—	—
AL27	5 (BC)	8	IO	PB33C	—	L28T_A0
AL28	5 (BC)	8	IO	PB33D	VREF_5_08	L28C_A0
AN31	5 (BC)	8	IO	PB34B	—	—
R15	—	—	VSS	VSS	—	—
AK26	5 (BC)	8	IO	PB34D	—	—
AM30	5 (BC)	9	IO	PB35B	—	—
AL29	5 (BC)	9	IO	PB35D	VREF_5_09	—
AK27	5 (BC)	9	IO	PB36B	—	—
R20	—	—	VSS	VSS	—	—
AL30	5 (BC)	9	IO	PB36C	—	L29T_D0
AK29	5 (BC)	9	IO	PB36D	—	L29C_D0
AK28	—	—	VDD33	VDD33	—	—
AA16	—	—	VDD15	VDD15	—	—
AP32	—	—	IO	PSCHAR_LDIO9	—	—

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AP33	—	—	IO	PSCHAR_LDIO8	—	—
AN32	—	—	IO	PSCHAR_LDIO7	—	—
AM31	—	—	IO	PSCHAR_LDIO6	—	—
AA17	—	—	VDD15	VDD15	—	—
AM32	—	—	VDD33	VDD33	—	—
AL31	—	—	IO	PSCHAR_LDIO5	—	—
AM33	—	—	IO	PSCHAR_LDIO4	—	—
AA18	—	—	VDD15	VDD15	—	—
AK30	—	—	IO	PSCHAR_LDIO3	—	—
AL32	—	—	IO	PSCHAR_LDIO2	—	—
AA19	—	—	VDD15	VDD15	—	—
AB16	—	—	VDD15	VDD15	—	—
AK31	—	—	VDD33	VDD33	—	—
AJ30	—	—	IO	PSCHAR_LDIO1	—	—
AK33	—	—	IO	PSCHAR_LDIO0	—	—
AK34	—	—	IO	PSCHAR_CKIO1	—	—
AJ31	—	—	IO	PSCHAR_CKIO0	—	—
AJ33	—	—	IO	PSCHAR_XCK	—	—
AJ34	—	—	IO	PSCHAR_WDSYNC	—	—
AH30	—	—	IO	PSCHAR_CV	—	—
AH31	—	—	IO	PSCHAR_BYTSYNC	—	—
AH32	—	—	I	ATMOUT_B	—	—
AH33	—	—	VSSGB_B	VSSGB_B	—	—
AH34	—	—	VDDGB_B	VDDGB_B	—	—
AA32	—	—	VDDR	VDDAUX_B	—	—
AF30	—	—	O	REXT_B	—	—
AF31	—	—	O	REXTN_B	—	—
AE30	—	—	I	REFCLKN_B	—	—
AE31	—	—	I	REFCLKP_B	—	—
AB32	—	—	VsST	VSSAUX_B	—	—
AD30	—	—	VDDIB	VDDIB_BA	—	—
AD32	—	—	VDDR	VDDR_BA	—	—
AF33	—	—	I	HDINN_BA	—	—
AC32	—	—	VsST	VSSIB_BA	—	—
AF34	—	—	I	HDINP_BA	—	—
AE32	—	—	VDDR	VDDR_BA	—	—
AD31	—	—	VSSRX	VSSRX_BA	—	—
K32	—	—	VDDR	VDDTX_BA	—	—
AC30	—	—	VDDOB	VDDOB_BA	—	—
AE33	—	—	O	HDOUTN_BA	—	—
AF32	—	—	VsST	VSSOB_BA	—	—
AE34	—	—	O	HDOUTP_BA	—	—
AG30	—	—	VsST	VsSTX_BA	—	—
AB30	—	—	VDDIB	VDDIB_BB	—	—

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AD33	—	—	I	HDINN_BB	—	—
AG31	—	—	VsST	VSSIB_BB	—	—
AD34	—	—	I	HDINP_BB	—	—
AC31	—	—	VSSRX	VSSRX_BB	—	—
AB31	—	—	VDDOB	VDDOB_BB	—	—
AC33	—	—	O	HDOUTN_BB	—	—
AG32	—	—	VsST	VSSOB_BB	—	—
AC34	—	—	O	HDOUTP_BB	—	—
AG33	—	—	VsST	VsSTX_BB	—	—
AA30	—	—	VDDIB	VDDIB_BC	—	—
AB33	—	—	I	HDINN_BC	—	—
AG34	—	—	VsST	VSSIB_BC	—	—
AB34	—	—	I	HDINP_BC	—	—
AA31	—	—	VSSRX	VSSRX_BC	—	—
Y30	—	—	VDDOB	VDDOB_BC	—	—
AA33	—	—	O	HDOUTN_BC	—	—
H30	—	—	VsST	VSSOB_BC	—	—
AA34	—	—	O	HDOUTP_BC	—	—
Y31	—	—	VDDOB	VDDOB_BC	—	—
H31	—	—	VsST	VsSTX_BC	—	—
W30	—	—	VDDIB	VDDIB_BD	—	—
Y33	—	—	I	HDINN_BD	—	—
H32	—	—	VsST	VSSIB_BD	—	—
Y34	—	—	I	HDINP_BD	—	—
W31	—	—	VSSRX	VSSRX_BD	—	—
V30	—	—	VDDOB	VDDOB_BD	—	—
W33	—	—	O	HDOUTN_BD	—	—
H33	—	—	VsST	VSSOB_BD	—	—
W34	—	—	O	HDOUTP_BD	—	—
V31	—	—	VDDOB	VDDOB_BD	—	—
H34	—	—	VsST	VsSTX_BD	—	—
J32	—	—	VsST	VsSTX_AD	—	—
U31	—	—	VDDOB	VDDOB_AD	—	—
T34	—	—	O	HDOUTP_AD	—	—
M32	—	—	VsST	VSSOB_AD	—	—
T33	—	—	O	HDOUTN_AD	—	—
U30	—	—	VDDOB	VDDOB_AD	—	—
T31	—	—	VSSRX	VSSRX_AD	—	—
R34	—	—	I	HDINP_AD	—	—
N32	—	—	VsST	VSSIB_AD	—	—
R33	—	—	I	HDINN_AD	—	—
T30	—	—	VDDIB	VDDIB_AD	—	—
U32	—	—	VsST	VsSTX_AC	—	—
R31	—	—	VDDOB	VDDOB_AC	—	—

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
P34	—	—	O	HDOUTP_AC	—	—
U33	—	—	VsST	VSSOB_AC	—	—
P33	—	—	O	HDOUTN_AC	—	—
R30	—	—	VDDOB	VDDOB_AC	—	—
P31	—	—	VSSRX	VSSRX_AC	—	—
N34	—	—	I	HDINP_AC	—	—
U34	—	—	VsST	VSSIB_AC	—	—
N33	—	—	I	HDINN_AC	—	—
P30	—	—	VDDIB	VDDIB_AC	—	—
V32	—	—	VsST	VsSTX_AB	—	—
N31	—	—	VDDOB	VDDOB_AB	—	—
M34	—	—	O	HDOUTP_AB	—	—
V33	—	—	VsST	VSSOB_AB	—	—
M33	—	—	O	HDOUTN_AB	—	—
M31	—	—	VSSRX	VSSRX_AB	—	—
L34	—	—	I	HDINP_AB	—	—
V34	—	—	VsST	VSSIB_AB	—	—
L33	—	—	I	HDINN_AB	—	—
N30	—	—	VDDIB	VDDIB_AB	—	—
M30	—	—	VDDOB	VDDOB_AA	—	—
K34	—	—	O	HDOUTP_AA	—	—
K33	—	—	O	HDOUTN_AA	—	—
L32	—	—	VDDR	VDDTX_AA	—	—
L31	—	—	VSSRX	VSSRX_AA	—	—
P32	—	—	VDDR	VDDR_X_AA	—	—
J34	—	—	I	HDINP_AA	—	—
J33	—	—	I	HDINN_AA	—	—
R32	—	—	VDDR	VDDR_X_AA	—	—
L30	—	—	VDDIB	VDDIB_AA	—	—
K31	—	—	I	REFCLKP_A	—	—
K30	—	—	I	REFCLKN_A	—	—
J31	—	—	O	REXTN_A	—	—
J30	—	—	O	REXT_A	—	—
Y32	—	—	VDDR	VDDAUX_A	—	—
G34	—	—	VDDGB_A	VDDGB_A	—	—
G33	—	—	VSSGB_A	VSSGB_A	—	—
G32	—	—	I	ATMOUT_A	—	—
G31	—	—	I	PRESERVE01	—	—
F33	—	—	I	PRESERVE02	—	—
G30	—	—	I	PRESERVE03	—	—
F31	—	—	O	PSYS_RSSIG_ALL	—	—
F30	—	—	I	PSYS_DOBISTN	—	—
E31	—	—	VDD33	VDD33	—	—
AB17	—	—	VDD15	VDD15	—	—

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AB18	—	—	VDD15	VDD15	—	—
D32	—	—	I	PBIST_TEST_ENN	—	—
E30	—	—	I	PLOOP_TEST_ENN	—	—
AB19	—	—	VDD15	VDD15	—	—
D31	—	—	I	PASB_PDN	—	—
C32	—	—	I	PMP_TESTCLK	—	—
C31	—	—	VDD33	VDD33	—	—
AJ32	—	—	VDD15	VDD15	—	—
B32	—	—	I	PASB_RESETN	—	—
A33	—	—	I	PASB_TRISTN	—	—
B31	—	—	I	PMP_TESTCLK_ENN	—	—
A32	—	—	I	PASB_TESTCLK	—	—
AK32	—	—	VDD15	VDD15	—	—
AB21	—	—	VSS	VSS	—	—
A31	—	—	VDD33	VDD33	—	—
B30	1 (TC)	7	IO	PT36D	—	—
AB22	—	—	VSS	VSS	—	—
C30	1 (TC)	7	IO	PT36B	—	—
D30	1 (TC)	7	IO	PT35D	—	—
B13	1 (TC)	—	VDDIO1	VDDIO1	—	—
E29	1 (TC)	7	IO	PT35B	—	—
E28	1 (TC)	7	IO	PT34D	VREF_1_07	—
AN33	—	—	VSS	Vss	—	—
D29	1 (TC)	8	IO	PT34B	—	—
B29	1 (TC)	8	IO	PT33D	—	L1C_A0
C29	1 (TC)	8	IO	PT33C	VREF_1_08	L1T_A0
B15	1 (TC)	—	VDDIO1	VDDIO1	—	—
E27	1 (TC)	8	IO	PT32D	—	L2C_A0
E26	1 (TC)	8	IO	PT32C	—	L2T_A0
AP34	—	—	Vss	Vss	—	—
A30	1 (TC)	8	IO	PT32B	—	—
A29	1 (TC)	9	IO	PT31D	—	L3C_D3
E25	1 (TC)	9	IO	PT31C	VREF_1_09	L3T_D3
B17	1 (TC)	—	VDDIO1	VDDIO1	—	—
E24	1 (TC)	9	IO	PT31A	—	—
B28	1 (TC)	9	IO	PT30D	—	L4C_A0
C28	1 (TC)	9	IO	PT30C	—	L4T_A0
B2	—	—	Vss	Vss	—	—
D28	1 (TC)	9	IO	PT30A	—	—
C27	1 (TC)	9	IO	PT29D	—	L5C_A0
D27	1 (TC)	9	IO	PT29C	—	L5T_A0
E23	1 (TC)	9	IO	PT29B	—	L6C_A0
E22	1 (TC)	9	IO	PT29A	—	L6T_A0
D26	1 (TC)	1	IO	PT28D	—	L7C_A0

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
D25	1 (TC)	1	IO	PT28C	—	L7T_A0
B33	—	—	Vss	Vss	—	—
D24	1 (TC)	1	IO	PT28B	—	L8C_A0
D23	1 (TC)	1	IO	PT28A	—	L8T_A0
C26	1 (TC)	1	IO	PT27D	VREF_1_01	L9C_A0
C25	1 (TC)	1	IO	PT27C	—	L9T_A0
D11	1 (TC)	—	VDDIO1	VDDIO1	—	—
E21	1 (TC)	1	IO	PT27B	—	L10C_A0
E20	1 (TC)	1	IO	PT27A	—	L10T_A0
D22	1 (TC)	2	IO	PT26D	—	L11C_A0
D21	1 (TC)	2	IO	PT26C	VREF_1_02	L11T_A0
E34	—	—	Vss	Vss	—	—
A28	1 (TC)	2	IO	PT26B	—	—
B26	1 (TC)	2	IO	PT25D	—	L12C_A0
B25	1 (TC)	2	IO	PT25C	—	L12T_A0
D13	1 (TC)	—	VDDIO1	VDDIO1	—	—
B27	1 (TC)	2	IO	PT25B	—	—
A27	1 (TC)	3	IO	PT24D	—	L13C_A0
A26	1 (TC)	3	IO	PT24C	VREF_1_03	L13T_A0
N13	—	—	Vss	Vss	—	—
C24	1 (TC)	3	IO	PT24B	—	—
C22	1 (TC)	3	IO	PT23D	—	L14C_A0
C23	1 (TC)	3	IO	PT23C	—	L14T_A0
D15	1 (TC)	—	VDDIO1	VDDIO1	—	—
B24	1 (TC)	3	IO	PT23B	—	—
D20	1 (TC)	3	IO	PT22D	—	L15C_A0
D19	1 (TC)	3	IO	PT22C	—	L15T_A0
N14	—	—	Vss	Vss	—	—
E19	1 (TC)	3	IO	PT22B	—	L16C_A0
E18	1 (TC)	3	IO	PT22A	—	L16T_A0
C21	1 (TC)	4	IO	PT21D	—	L17C_A0
C20	1 (TC)	4	IO	PT21C	—	L17T_A0
A25	1 (TC)	4	IO	PT21B	—	L18C_A0
A24	1 (TC)	4	IO	PT21A	—	L18T_A0
B23	1 (TC)	4	IO	PT20D	—	L19C_A0
A23	1 (TC)	4	IO	PT20C	—	L19T_A0
N15	—	—	Vss	Vss	—	—
E17	1 (TC)	4	IO	PT20B	—	L20C_A0
E16	1 (TC)	4	IO	PT20A	—	L20T_A0
B22	1 (TC)	4	IO	PT19D	—	L21C_A0
B21	1 (TC)	4	IO	PT19C	VREF_1_04	L21T_A0
C18	1 (TC)	4	IO	PT19B	—	L22C_A0
C19	1 (TC)	4	IO	PT19A	—	L22T_A0
N20	—	—	Vss	Vss	—	—



BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
A22	1 (TC)	5	IO	PT18D	PTCK1C	L23C_A0
A21	1 (TC)	5	IO	PT18C	PTCK1T	L23T_A0
N21	—	—	Vss	Vss	—	—
D17	1 (TC)	5	IO	PT18B	—	L24C_A0
D18	1 (TC)	5	IO	PT18A	—	L24T_A0
B20	1 (TC)	5	IO	PT17D	PTCK0C	L25C_A0
B19	1 (TC)	5	IO	PT17C	PTCK0T	L25T_A0
A20	1 (TC)	5	IO	PT17B	—	L26C_A0
A19	1 (TC)	5	IO	PT17A	—	L26T_A0
A18	1 (TC)	5	IO	PT16D	VREF_1_05	L27C_A0
B18	1 (TC)	5	IO	PT16C	—	L27T_A0
Y21	—	—	Vss	Vss	—	—
C17	1 (TC)	5	IO	PT16B	—	L28C_D0
D16	1 (TC)	5	IO	PT16A	—	L28T_D0
A17	1 (TC)	6	IO	PT15D	—	L29C_D0
B16	1 (TC)	6	IO	PT15C	—	L29T_D0
E15	1 (TC)	6	IO	PT15B	—	L30C_A0
E14	1 (TC)	6	IO	PT15A	—	L30T_A0
A16	1 (TC)	6	IO	PT14D	—	L31C_A0
A15	1 (TC)	6	IO	PT14C	VREF_1_06	L31T_A0
Y22	—	—	Vss	Vss	—	—
D14	1 (TC)	6	IO	PT14B	—	—
C16	0 (TL)	1	IO	PT13D	MPI_RTRY_N	L1C_A0
C15	0 (TL)	1	IO	PT13C	MPI_ACK_N	L1T_A0
D7	0 (TL)	—	VDDIO0	VDDIO0	—	—
C14	0 (TL)	1	IO	PT13B	—	L2C_A0
B14	0 (TL)	1	IO	PT13A	VREF_0_01	L2T_A0
A14	0 (TL)	1	IO	PT12D	M0	L3C_A0
A13	0 (TL)	1	IO	PT12C	M1	L3T_A0
AA20	—	—	Vss	Vss	—	—
E12	0 (TL)	2	IO	PT12B	MPI_CLK	L4C_A0
E13	0 (TL)	2	IO	PT12A	A21/MPI_BURST_N	L4T_A0
C13	0 (TL)	2	IO	PT11D	M2	L5C_A0
C12	0 (TL)	2	IO	PT11C	M3	L5T_A0
B12	0 (TL)	2	IO	PT11B	VREF_0_02	L6C_A0
A12	0 (TL)	2	IO	PT11A	MPI_TEA_N	L6T_A0
D12	0 (TL)	3	IO	PT10D	—	L7C_D0
C11	0 (TL)	3	IO	PT10C	—	L7T_D0
B11	0 (TL)	3	IO	PT10B	—	—
A11	0 (TL)	3	IO	PT9D	VREF_0_03	L8C_A0
A10	0 (TL)	3	IO	PT9C	—	L8T_A0
AA21	—	—	Vss	Vss	—	—
B10	0 (TL)	3	IO	PT9B	—	—
E11	0 (TL)	3	IO	PT8D	D0	L9C_D0

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
D10	0 (TL)	3	IO	PT8C	TMS	L9T_D0
C10	0 (TL)	3	IO	PT8B	—	—
A9	0 (TL)	4	IO	PT7D	A20/MPI_BDIP_N	L10C_A0
B9	0 (TL)	4	IO	PT7C	A19/MPI_TSZ1	L10T_A0
AA22	—	—	Vss	Vss	—	—
E10	0 (TL)	4	IO	PT7B	—	—
A8	0 (TL)	4	IO	PT6D	A18/MPI_TSZ0	L11C_A0
B8	0 (TL)	4	IO	PT6C	D3	L11T_A0
D9	0 (TL)	4	IO	PT6B	VREF_0_04	L12C_D0
C8	0 (TL)	4	IO	PT6A	—	L12T_D0
E9	0 (TL)	5	IO	PT5D	D1	L13C_D0
D8	0 (TL)	5	IO	PT5C	D2	L13T_D0
AB13	—	—	Vss	Vss	—	—
A7	0 (TL)	5	IO	PT5B	—	L14C_A0
A6	0 (TL)	5	IO	PT5A	VREF_0_05	L14T_A0
C7	0 (TL)	5	IO	PT4D	TDI	L15C_D0
B6	0 (TL)	5	IO	PT4C	TCK	L15T_D0
E8	0 (TL)	5	IO	PT4B	—	L16C_A0
E7	0 (TL)	5	IO	PT4A	—	L16T_A0
A5	0 (TL)	6	IO	PT3D	—	L17C_A0
B5	0 (TL)	6	IO	PT3C	VREF_0_06	L17T_A0
AB14	—	—	Vss	Vss	—	—
C6	0 (TL)	6	IO	PT3B	—	L18C_A0
D6	0 (TL)	6	IO	PT3A	—	L18T_A0
C4	0 (TL)	6	IO	PT2D	PLL_CK1C/PPLL	L19C_A0
B4	0 (TL)	6	IO	PT2C	PLL_CK1T/PPLL	L19T_A0
A4	0 (TL)	6	IO	PT2B	—	L20C_A0
A3	0 (TL)	6	IO	PT2A	—	L20T_A0
D5	—	—	O	PCFG_MPI_IRQ	CFG_IRQ_N/MPI_IRQ_N	—
E6	—	—	IO	PCCLK	CCLK	—
D4	—	—	IO	PDONE	DONE	—
E5	—	—	VDD33	VDD33	—	—
AB15	—	—	Vss	Vss	—	—
AL33	—	—	VDD15	VDD15	—	—
AL34	—	—	VDD15	VDD15	—	—
AM34	—	—	VDD15	VDD15	—	—
AN34	—	—	VDD15	VDD15	—	—
B34	—	—	VDD15	VDD15	—	—
C33	—	—	VDD15	VDD15	—	—
C34	—	—	VDD15	VDD15	—	—
D33	—	—	VDD15	VDD15	—	—
D34	—	—	VDD15	VDD15	—	—
E32	—	—	VDD15	VDD15	—	—

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
E33	—	—	VDD15	VDD15	—	—
F32	—	—	VDD15	VDD15	—	—
F34	—	—	VDD15	VDD15	—	—
N16	—	—	VDD15	VDD15	—	—
N17	—	—	VDD15	VDD15	—	—
N18	—	—	VDD15	VDD15	—	—
N19	—	—	VDD15	VDD15	—	—
P16	—	—	VDD15	VDD15	—	—
P17	—	—	VDD15	VDD15	—	—
P18	—	—	VDD15	VDD15	—	—
P19	—	—	VDD15	VDD15	—	—
R16	—	—	VDD15	VDD15	—	—
R17	—	—	VDD15	VDD15	—	—
R18	—	—	VDD15	VDD15	—	—
R19	—	—	VDD15	VDD15	—	—
T13	—	—	VDD15	VDD15	—	—
T14	—	—	VDD15	VDD15	—	—
T15	—	—	VDD15	VDD15	—	—
T20	—	—	VDD15	VDD15	—	—
T21	—	—	VDD15	VDD15	—	—
T22	—	—	VDD15	VDD15	—	—
U13	—	—	VDD15	VDD15	—	—
U14	—	—	VDD15	VDD15	—	—
U15	—	—	VDD15	VDD15	—	—
U20	—	—	VDD15	VDD15	—	—
U21	—	—	VDD15	VDD15	—	—
U22	—	—	VDD15	VDD15	—	—
V13	—	—	VDD15	VDD15	—	—
V14	—	—	VDD15	VDD15	—	—
V15	—	—	VDD15	VDD15	—	—
V20	—	—	VDD15	VDD15	—	—
V21	—	—	VDD15	VDD15	—	—
V22	—	—	VDD15	VDD15	—	—
W13	—	—	VDD15	VDD15	—	—
W14	—	—	VDD15	VDD15	—	—
W15	—	—	VDD15	VDD15	—	—
W20	—	—	VDD15	VDD15	—	—
W21	—	—	VDD15	VDD15	—	—
W22	—	—	VDD15	VDD15	—	—
Y16	—	—	VDD15	VDD15	—	—
Y17	—	—	VDD15	VDD15	—	—
Y18	—	—	VDD15	VDD15	—	—
Y19	—	—	VDD15	VDD15	—	—

---

<b>BM680</b>	<b>VDDIO Bank</b>	<b>VREF Group</b>	<b>I/O</b>	<b>Pin Description</b>	<b>Additional Function</b>	<b>BM680 Pair</b>
T32	—	—	NC	NC	—	—
W32	—	—	NC	NC	—	—

## Package Information

### Package Thermal Characteristics Summary

There are three thermal parameters that are in common use:  $\theta_{JA}$ ,  $\psi_{JC}$ , and  $\theta_{JC}$ . It should be noted that all the parameters are affected, to varying degrees, by package design (including paddle size) and choice of materials, the amount of copper in the test board or system board, and system airflow.

#### $\theta_{JA}$

This is the thermal resistance from junction to ambient (theta-JA, R-theta, etc.):

$$\theta_{JA} = \frac{T_J - T_A}{Q}$$

where  $T_J$  is the junction temperature,  $T_A$  is the ambient air temperature, and  $Q$  is the chip power.

Experimentally,  $\theta_{JA}$  is determined when a special thermal test die is assembled into the package of interest, and the part is mounted on the thermal test board. The diodes on the test chip are separately calibrated in an oven. The package/board is placed either in a JEDEC natural convection box or in the wind tunnel, the latter for forced convection measurements. A controlled amount of power ( $Q$ ) is dissipated in the test chip's heater resistor, the chip's temperature ( $T_J$ ) is determined by the forward drop on the diodes, and the ambient temperature ( $T_A$ ) is noted. Note that  $\theta_{JA}$  is expressed in units of  $^{\circ}\text{C}/\text{W}$ .

#### $\psi_{JC}$

This JEDEC designated parameter correlates the junction temperature to the case temperature. It is generally used to infer the junction temperature while the device is operating in the system. It is not considered a true thermal resistance and it is defined by:

$$\psi_{JC} = \frac{T_J - T_C}{Q}$$

where  $T_C$  is the case temperature at top dead center,  $T_J$  is the junction temperature, and  $Q$  is the chip power. During the  $\theta_{JA}$  measurements described above, besides the other parameters measured, an additional temperature reading,  $T_C$ , is made with a thermocouple attached at top-dead-center of the case.  $\psi_{JC}$  is also expressed in units of  $^{\circ}\text{C}/\text{W}$ .

#### $\theta_{JC}$

This is the thermal resistance from junction to case. It is most often used when attaching a heat sink to the top of the package. It is defined by:

$$\theta_{JC} = \frac{T_J - T_C}{Q}$$

The parameters in this equation have been defined above. However, the measurements are performed with the case of the part pressed against a water-cooled heat sink to draw most of the heat generated by the chip out the top of the package. It is this difference in the measurement process that differentiates  $\theta_{JC}$  from  $\psi_{JC}$ .  $\theta_{JC}$  is a true thermal resistance and is expressed in units of  $^{\circ}\text{C}/\text{W}$ .

**θ<sub>JB</sub>**

This is the thermal resistance from junction to board (θ<sub>JL</sub>). It is defined by:

$$\theta_{JB} = \frac{T_J - T_B}{Q}$$

where T<sub>B</sub> is the temperature of the board adjacent to a lead measured with a thermocouple. The other parameters on the right-hand side have been defined above. This is considered a true thermal resistance, and the measurement is made with a water-cooled heat sink pressed against the board to draw most of the heat out of the leads. Note that θ<sub>JB</sub> is expressed in units of °C/W and that this parameter and the way it is measured are still being discussed by the JEDEC committee.

**FPSC Maximum Junction Temperature**

Once the power dissipated by the FPSC has been determined (see the Estimating Power Dissipation section), the maximum junction temperature of the FPSC can be found. This is needed to determine if speed derating of the device from the 85 °C junction temperature used in all of the delay tables is needed. Using the maximum ambient temperature, T<sub>Amax</sub>, and the power dissipated by the device, Q (expressed in °C), the maximum junction temperature is approximated by:

$$T_{Jmax} = T_{Amax} + (Q \cdot \theta_{JA})$$

Table 36 lists the thermal characteristics for all packages used with the ORCA ORSO82G5 Series of FPSCs.

**Package Thermal Characteristics**

**Table 36. ORCA ORSO82G5 Plastic Package Thermal Guidelines**

Package	θ <sub>JA</sub> (°C/W)			Maximum Power (W) T = 70 °C Max, T <sub>J</sub> = 125 °C Max, 0 fpm
	0 fpm	200 fpm	500 fpm	
680-Pin PBGAM	9.8	7.8	6.8	4.1

Note: The 680-pin PBGAM package for the ORSO82G5 includes a heat spreader.

**Package Coplanarity**

The coplanarity limits of the Lattice packages are as follows:

- PBGAM: 8.0 mils

---

**Heat Sink Vendors for BGA Packages**

The estimated worst-case power requirements for the ORSO82G5 are in the 4 W to 5 W range. Consequently, for most applications an external heat sink will be required. The following table lists, in alphabetical order, heat sink vendors who advertise heat sinks aimed at the BGA market.

**Table 37. Heat Sink Vendors**

<b>Vendor</b>	<b>Location</b>	<b>Phone</b>
Aavid Thermal Technology	Laconia, NH	(603) 527-2152
Chip Coolers	Warwick, RI	(800) 227-0254
IERC	Burbank, CA	(818) 842-7277
R-Theta	Buffalo, NY	(800) 388-5428
Sanyo Denki	Torrance, CA	(310) 783-5400
Thermalloy	Dallas, TX	(214) 243-4321
Wakefield Engineering	Wakefield, MA	(617) 246-0874

**Package Parasitics**

The electrical performance of an IC package, such as signal quality and noise sensitivity, is directly affected by the package parasitics. Table 38 lists eight parasitics associated with the ORCA packages. These parasitics represent the contributions of all components of a package, which include the bond wires, all internal package routing, and the external leads.

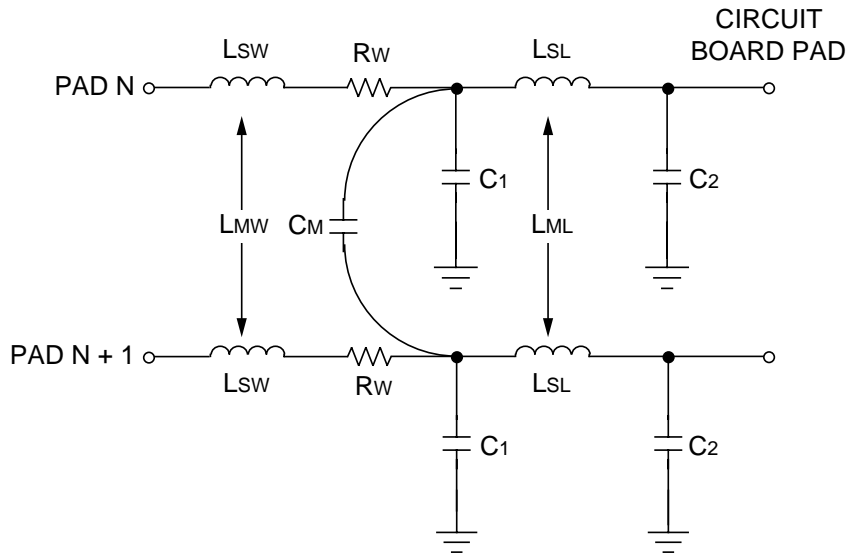
Four inductances in nH are listed: L<sub>SW</sub> and L<sub>SL</sub>, the self-inductance of the lead; and L<sub>MW</sub> and L<sub>M</sub>L, the mutual inductance to the nearest neighbor lead. These parameters are important in determining ground bounce noise and inductive crosstalk noise. Three capacitances in pF are listed: C<sub>M</sub>, the mutual capacitance of the lead to the nearest neighbor lead; and C<sub>1</sub> and C<sub>2</sub>, the total capacitance of the lead to all other leads (all other leads are assumed to be grounded). These parameters are important in determining capacitive crosstalk and the capacitive loading effect of the lead. Resistance values are in mΩ.

The parasitic values in Table 38 are for the circuit model of bond wire and package lead parasitics. If the mutual capacitance value is not used in the designer’s model, then the value listed as mutual capacitance should be added to each of the C<sub>1</sub> and C<sub>2</sub> capacitors.

**Table 38. ORCA ORSO82G5 Package Parasitics**

Package Type	L <sub>SW</sub>	L <sub>MW</sub>	R <sub>W</sub>	C <sub>1</sub>	C <sub>2</sub>	C <sub>M</sub>	L <sub>SL</sub>	L <sub>M</sub> L
680-Pin PBGAM	3.8	1.3	250	1.0	1.0	0.3	2.8—5	0.5—1

**Figure 50. Package Parasitics**





## Package Outline Diagrams

### Terms and Definitions

**Basic Size (BSC):** The basic size of a dimension is the size from which the limits for that dimension are derived by the application of the allowance and the tolerance.

**Design Size:** The design size of a dimension is the actual size of the design, including an allowance for fit and tolerance.

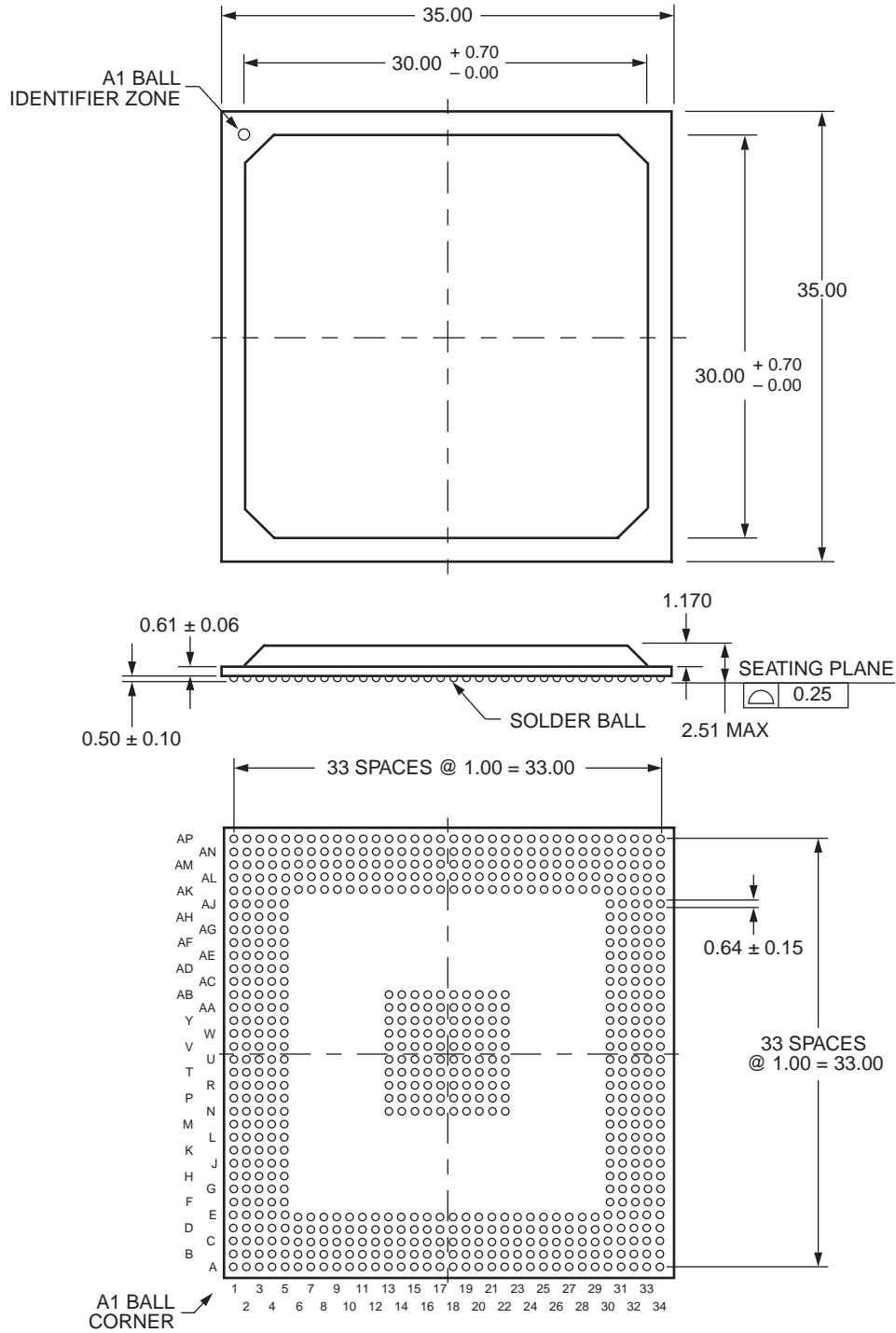
**Typical (TYP):** When specified after a dimension, this indicates the repeated design size if a tolerance is specified or repeated basic size if a tolerance is not specified.

**Reference (REF):** The reference dimension is an untoleranced dimension used for informational purposes only. It is a repeated dimension or one that can be derived from other values in the drawing.

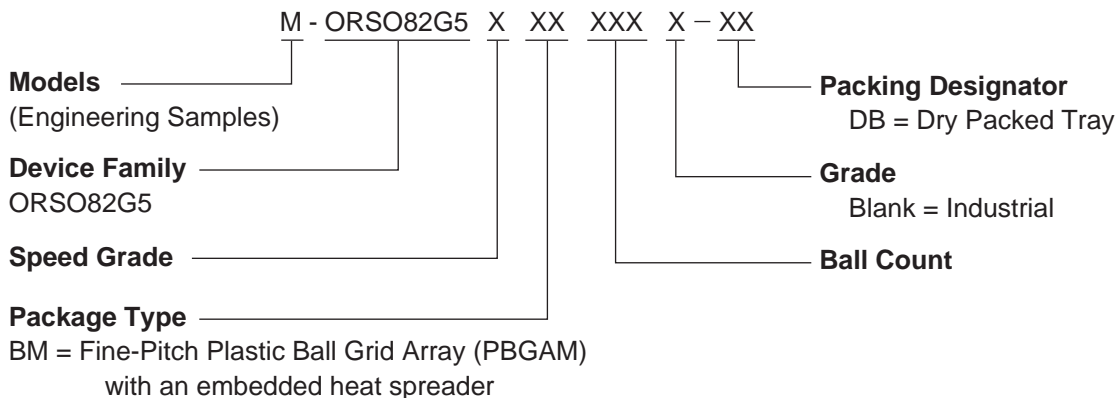
**Minimum (MIN) or Maximum (MAX):** Indicates the minimum or maximum allowable size of a dimension.

680-Pin PBGAM

Dimensions are in millimeters.



### Ordering Information



**Table 39. Device Type Options**

Device	Voltage
ORSO82G5	1.5 V internal 3.3 V/2.5 V/1.8 V/1.5 V I/O

**Table 40. Temperature Range**

Symbol	Description	Ambient Temperature
(Blank)	Industrial	-40 °C to +85 °C

Note: Device junction temperature of -40 °C to +125 °C are recommended

**Table 41. Ordering Information**

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade	Packing Designator
ORT82G5	M-ORSO82G52BM680-DB	2	PBGAM	680	I	DB
	M-ORSO82G51BM680-DB	1	PBGAM	680	I	DB

Note: -3 speed grade to be released with production version of device.