ORCA® ORT8850

Field-Programmable System Chip (FPSC)
Eight-Channel x 850 Mbits/s Backplane Transceiver

February 2008 **Data Sheet**

Introduction

Field Programmable System-on-a-Chip (FPSCs) bring a whole new dimension to programmable logic: Field Programmable Gate Array (FPGA) logic and an embedded system solution on a single device. Lattice has developed a solution for designers who need the many advantages of FPGA-based design implementation, coupled with high-speed serial backplane data transfer. Built on the Series 4 reconfigurable embedded System-on-a-Chip (SoC) architecture, the ORT8850 family is made up of backplane transceivers (SERDES) containing eight channels, each operating at up to 850 Mbits/s (6.8 Gbits/s when all eight channels are used). This is combined with a full-duplex synchronous interface, with built-in Clock and Data Recovery (CDR) in standard-cell logic, along with over 600K usable FPGA system gates (ORT8850H). 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. For example, designers can build a 6.8 Gbits/s PCI-to-PCI half bridge using our PCI soft core.

The ORT8850 family offers a clockless High-Speed Interface for inter-device communication on a board or across a backplane. The built-in clock recovery of the ORT8850 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 the backplane transceiver as a network termination device. The backplane transceiver offers SONET scrambling/descrambling of data and streamlined SONET framing, pointer moving, and transport overhead handling, plus the programmable logic to terminate the network into proprietary systems. For non-SONET applications, all SONET functionality is hidden from the user and no prior networking knowledge is required.

Table 1. ORCA ORT8850 Family – Available FPGA Logic (equivalent to OR4E02 and OR4E06 respectively)

Device	PFU Rows	PFU Columns	Total PFUs	FPGA Max User I/Os	LUTs	EBR Blocks	EBR Bits (K)	FPGA System Gates (K)
ORT8850L	26	24	624	278	4,992	8	74	201 - 397
ORT8850H	46	44	2,024	297	16,192	16	148	471 - 899

Note: 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 PFUs are used for logic only (No PFU RAM) with 40% EBR usage and 2 PLL's. Maximum System Gates assumes 80% of the PFUs are for logic, 20% are used for PFU RAM, with 80% EBR usage and 6 PLLs.

© 2008 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal. All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table of Contents

Introduction 1
Table of Contents 2
Features 3
- Embedded Core Features 3
- FPGA Features 4
- Programmable Logic System Features 5

Description 6
- What is an FPSC? 6
- FPSC Overview 6
- ispLEVER Development System 7
- FPSC Design Kit 7
- FPGA Logic Overview 8
- System-Level Features 9
- Configuration 10
- Additional Information 10

ORT8850 Overview 11
- Embedded Core Overview 11
- SONET Logic Blocks - Overview 12
- System Considerations for Reference Clock Distribution 15
- SONET Bypass Mode 16
- STM Macrocells - Overview 17
- HSI Macrocell - Overview 19
- Supervisory and Test Support Features - Overview 19
- Protection Switching - Overview 21
- FPSC Configuration - Overview 22

Backplane Transceiver Core Detailed Description 25
- SONET Logic Blocks, Detailed Description 25
- Receive Path Logic 34

FPGA/Embedded Core Interface Signals 47
- Clock and Data Timing at the FPGA/Embedded Core Interface - SONET Block 49
- Powerdown Mode 56
- Protection Switching 56

Memory Map 57
- Registers Access and General Description 57

Electrical Characteristics 69
- Absolute Maximum Ratings 69

Recommended Operating Conditions 69
- Power Supply Decoupling LC Circuit 70
- HSI Electrical and Timing Characteristics 71
- Embedded Core LVDS I/O 73

Pin Information 77
Package Pinouts 82
Package Thermal Characteristics Summary 100
- qJA 100
- YJC 100
- qJC 100
- qJB 100
- FPSC Maximum Junction Temperature 101
- Package Thermal Characteristics 101
- Heat Sink Information 101
- Package Coplanarity 101

Package Parasitics 102
Package Outline Diagrams 102
- Terms and Definitions 102
- Package Outline Drawings 103

Ordering Information 104
Revision History 105

Features

Embedded Core Features

- Implemented in an *ORCA* Series 4 FPGA.
- Allows a wide range of high-speed backplane applications, including SONET transport and termination.
- No knowledge of SONET/SDH needed in generic applications. Simply supply data, 78 MHz—106 MHz clock, and a frame pulse.
- High-Speed Interface (HSI) function for clock/data recovery serial backplane data transfer without external clocks.
- Eight-channel HSI function provides 850 Mbits/s serial interface per channel for a total chip bandwidth of 6.8 Gbits/s (full duplex).
- HSI function uses Lattice's 850 Mbits/s serial interface core. Rates from 126 Mbits/s to 850 Mbits/s are supported.
- LVDS I/Os compliant with *EIA*®-644 support hot insertion. All embedded LVDS I/Os include both input and output on-board termination to allow long-haul driving of backplanes.
- Low-power 1.5 V HSI core.
- Low-power LVDS buffers.
- Programmable STS-3, and STS-12 framing.
- Independent STS-3, and STS-12 data streams per quad channels.
- 8:1 data multiplexing/demultiplexing for 106.25 MHz byte-wide data processing in FPGA logic.
- On-chip, Phase-Lock Loop (PLL) clock meets (type B) jitter tolerance specification of ITU-T recommendation G.958.
- Powerdown option of HSI receiver on a per-channel basis.
- HSI automatically recovers from loss-of-clock once its reference clock returns to normal operating state.
- Frame alignment across multiple ORT8850 devices for work/protect switching at OC-192/STM-64 and above rates.
- In-band management and configuration through transport overhead extraction/insertion.
- Supports transparent modes where either the only insertion is A1/A2 framing bytes, or no bytes are inserted.
- Streamlined pointer processor (pointer mover) for 8 kHz frame alignment to system clocks.
- Built-in boundry scan (*IEEE*®1149.1 JTAG).
- FIFOs align incoming data across all eight channels (two groups of four channels or four groups of two channels) for both SONET scrambling. Optional ability to bypass alignment FIFOs.
- 1 + 1 protection supports STS-12/STS-48 redundancy by either software or hardware control for protection switching applications. STS-192 and above rates are supported through multiple devices.
- *ORCA* FPGA soft intellectual property core support for a variety of applications.
- Programmable Synchronous Transport Module (STM) pointer mover bypass mode.
- Programmable STM framer bypass mode.
- Programmable Clock and Data Recovery (CDR) bypass mode (clocked LVDS High-Speed Interface).
- Redundant outputs and multiplexed redundant inputs for CDR I/Os allow for implementation of eight channels with redundancy on a single device.

FPGA Features

- High-performance platform design:
  - 0.16 µm 7-level metal technology.
  - Internal performance of >250 MHz.
  - Over 600K FPGA system gates (ORT8850H).
  - 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:
  - LVTTL (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/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 & II), HSTL (Class I, III, IV), ZBT, and DDR.
  - Double-ended: LVDS, bused-LVDS, LVPECL.
  - LVDS include optional on-chip termination resistor per I/O and on-chip reference generation.
- New capability to (de)multiplex I/O signals:
  - New Double-Data Rate (DDR) on both input and output at rates up to 350 MHz (700 Mbits/s 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 → 1 MUX, new 8 → 1 MUX, and ripple mode arithmetic functions in the same PFU.
  - 32 x 4 RAM per PFU, configurable as single- 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-State Buffers, up to 10-bit decoder, and *PAL*®-like AND-OR-INVERT (AOI) in each programmable logic cell.
- Improved built-in clock management with dual-output 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 200 MHz embedded quad-port RAM blocks, two read ports, two write ports, and two sets of byte lane enables. Each embedded RAM block can be configured as:
  - One—512 x 18 (quad-port, two read/two write) with optional built-in arbitration.
  - One—256 x 36 (dual-port, one read/one write).
  - One—1K x 9 (dual-port, one read/one write).
  - Two—512 x 9 (dual-port, one read/one write for each).
  - Two RAM 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, 1K x 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 backplane transceiver 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.
- Cycle stealing capability allows a typical 15% to 40% internal speed improvement after final place and route. This feature also supports 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.

Programmable Logic System Features

- PCI local bus compliant for FPGA I/Os.
- Improved *PowerPC/Power QUICC MPC860* and *PowerPC* II *MPC8260* 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 backplane transceiver 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 backplane transceiver logic.
- New network PLLs meet ITU-T G.811 specifications and provide clock conditioning for DS-1/E-1 and STS-3/STM-1 applications.
- Variable size bused 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 (<200 ps for OR4E04).
- 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.

- Meets Universal Test and Operations PHY Interface for ATM (UTOPIA) Levels 1, 2, and 3. Also meets proposed specifications for UTOPIA Level 4 POS-PHY, Level 3 (2.5 Gbits/s), and POS-PHY 4 (10 Gbits/s) interface standards for Packet-over-SONET as defined by the Saturn Group.
- ispLEVER development system software. Supported by industry-standard CAE tools for design entry, synthesis, simulation, and timing analysis.

Description

What is an FPSC?

FPSCs, or Field Programmable System-on-a-Chip devices, 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. Columns of programmable logic are replaced on one side 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 provide 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.

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 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 supports user-programmable options in the embedded core, in turn allowing 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's 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.

The ispLEVER development system interfaces to front-end design entry tools and provides the tools to produce a configured FPGA. In the design flow, the user defines the functionality of the FPGA at two points in the design flow, the design entry and the bit stream generation stage. Recent improvements in ispLEVER allow the user to provide timing requirement information through logical preferences only; thus, the designer is not required to have physical knowledge of the implementation.

Following design entry, the development system's map, place, and route tools translate the netlist into a routed FPGA. A floor planner is available for layout feedback and control. A static timing analysis tool is provided to determine design speed, and a back-annotated netlist can be created to allow simulation and timing.

Timing and simulation output files from ispLEVER are also compatible with many third-party analysis tools. A bit stream generator is then used to generate the configuration data which is loaded into the FPGAs internal configuration RAM, embedded block RAM, and/or FPSC memory.

When using the bit stream generator, the user selects options that affect the functionality of the FPGA. Combined with the front-end tools, ispLEVER produces configuration data that implements the various logic and routing options discussed in this data sheet.

FPSC Design Kit

Development is facilitated by an FPSC design kit which, together with ispLEVER software 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*®, and/or compiled *Verilog*® simulation model, *HSPICE*® and/or IBIS models for I/O buffers, and complete online documentation. The kit's software couples with ispLEVER software, providing a seamless FPSC design environment. More information can be obtained by visiting the Lattice website at www.latticesemi.com or contacting a local sales office.

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-a-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 MicroProcessor Interface (MPI), Phase-Locked Loops (PLLs), and the Embedded System Bus (ESB).

PLC Logic

Each PFU within a PLC contains eight 4-input (16-bit) LUTs, eight latches/flip-flops, 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 flip-flops 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 flip-flop for pipelining. Each PFU may also be configured as a synchronous 32 x 4 single- or dual-port RAM or ROM. The flip-flops (or latches) may obtain input from LUT outputs or directly from invertible PFU inputs, or they can be tied high or tied low. The flip-flops 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 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 flip-flop, 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 is 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 3.3V/ 2.5V/1.8V/1.5V 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 support 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 MPC*860 and MPC8260 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.

The MPI provides a system-level MicroProcessor Interface to the FPGA user-defined logic, following configuration, through the system bus, including access to the embedded block RAM and general user-logic. 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 1-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. Mas-

ter and slave elements are also available for the user-logic and embedded backplane transceiver portion of the ORT8850.

The system bus control registers can provide control to the FPGA such as signaling for reprogramming, reset functions, and PLL programming. Status registers monitor $\overline{\text{INIT}}$, 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 MicroProcessor 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
Four user PLLs are provided for ORCA Series 4 FPSCs. Programmable PLLs can be used to manipulate the frequency, phase, and duty cycle of a clock signal. Each PPLL is capable of manipulating and conditioning clocks from 20 MHz to 420 MHz. Frequencies can be adjusted from 1/64x to 64x 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% increments of the clock period. An automatic input buffer delay compensation mode is available for phase delay. Each PPLL 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 power-up 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).

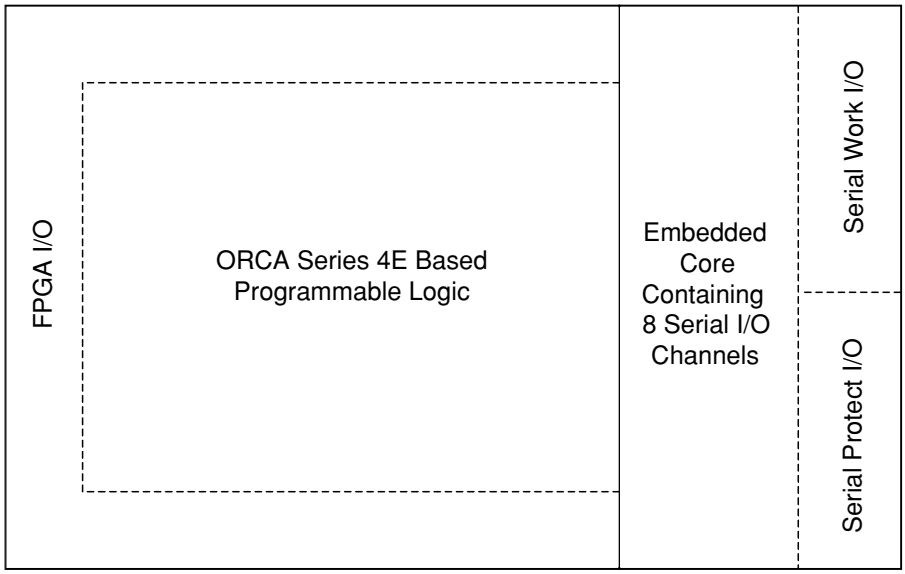
Additional Information

Contact your local Lattice representative for additional information regarding the ORCA Series 4 FPSC and FPGA devices, or visit our website at www.latticesemi.com.

ORT8850 Overview

The ORT8850 FPSCs provide high-speed backplane transceivers combined with FPGA logic. There are two devices in the ORT8850 family. The ORT8850L device is based on 1.5 V OR4E02 ORCA FPGA and has a 26 x 24 array of Programmable Logic Cells (PLCs). The ORT8850H device is based on 1.5V OR4E06 ORCA FPGA and has a 46 x 44 array. 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.

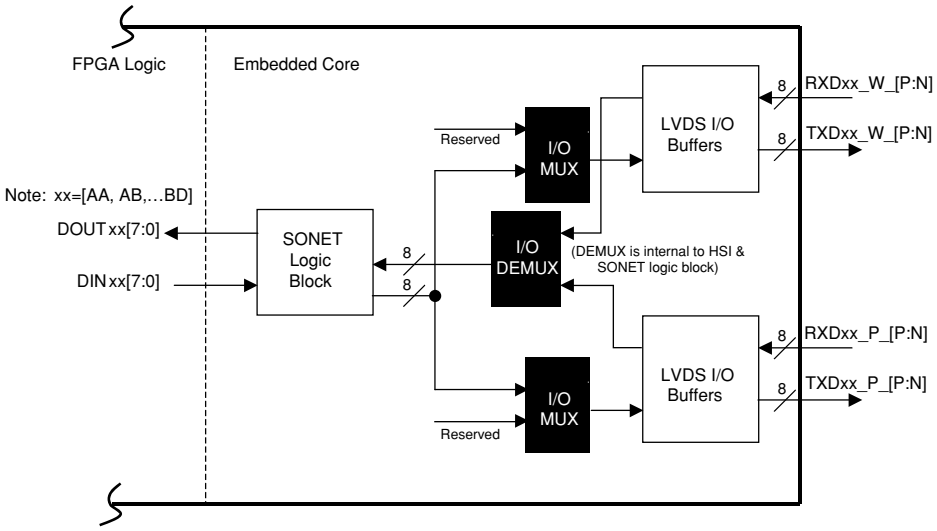
Figure 1. ORT8850 Top Level Diagram



Embedded Core Overview

The ORT8850 embedded core contains a pseudo-SONET block for backplane or intra-board, chip-to-chip communication. The SONET block includes a High-Speed Interface (HSI) macrocell and a Synchronous Transport Module (STM) macrocell. It supports eight full-duplex channels and performs data transfer, scrambling/descrambling and SONET framing at the maximum rate of 850 Mbits/s. Figure 2 shows a top level diagram of the ORT8850 and the basic data flows through the device.

Figure 2. ORT8850 Embedded Core, Top Level Functionality and Data Flow

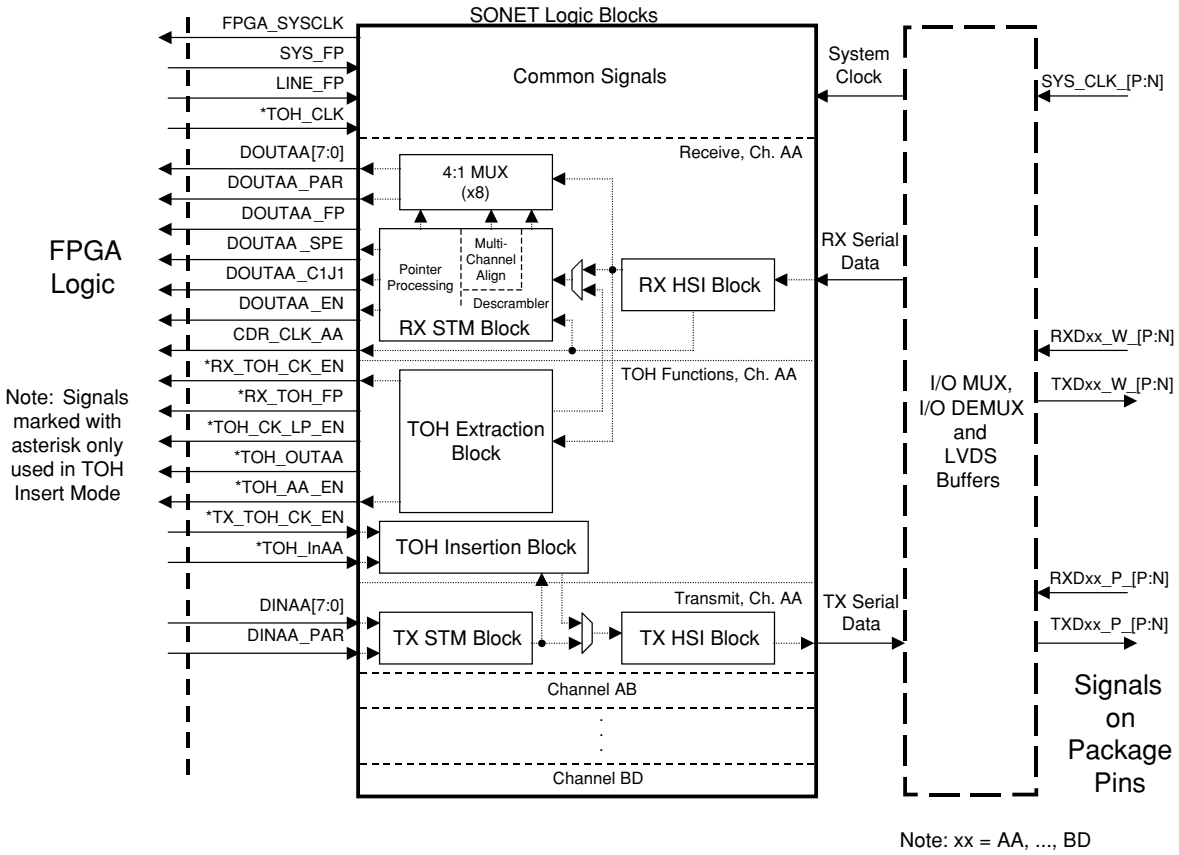


SONET Logic Blocks - Overview

The 850 Mbits/s SONET logic blocks allows the ORT8850 to communicate across a backplane or on a given board at an aggregate speed of 6.8 Gbits/s, allowing high-speed asynchronous serial data transfer between system devices. The external serial interfaces are implemented as eight channels of bidirectional 850 Mbits/s LVDS links and use a pseudo-SONET framing protocol, which can be bypassed.

The SONET logic blocks are organized into two quads. Each quad supports four full duplex serial links (quad A contains channels AA, AB, AC, and AD while quad B contains channels BA, BB, BC, and BD). A top level block diagram of one channel of the SONET logic is shown in Figure 3.

Figure 3. Top Level Block Diagram ORT8850 Embedded Core SONET Logic Block Common Signals and Channel AA Data Flow



Each quad can frame independently in STS-3, STS-12 or STS-48 format. If using STS-48 format all channels in the quad will be used and be treated as a single STS-48 channel using the quad STS-12 format in which each independent channel carries entire STS-12 frames. The byte order for STS-48 must be created by the designer in the FPGA design. Note that the recovered data will always continue to be in the same order as transmitted data.

Each channel contains transmit path and receive path logic, both of which are organized around High Speed Interconnect (HSI) and Synchronous Transport Mode (STM) macrocells. Additional logic allows insertion and extraction of information in the Transport Overhead area of the SONET frame. (Support for loopback and for switching between redundant serial links 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 ORT8850 and a top level overview of the Synchronous Transport Module (STM) and High Speed Interconnect (HSI) macrocells, which provide the SONET functionality.

SONET Framing

Each 850 Mbits/s serial link uses a pseudo-SONET protocol. SONET A1/A2 framing is used on the link to detect the 8 kHz frame location. The link is also scrambled using the standard SONET scrambler definition to ensure proper transitions on the link for improved CDR performance. The ORT8850 can do SONET framing and scrambling in both STS-12 and STS-3 formats.

Elastic buffers (FIFOs) are used to align each incoming STS-12 link to the local 77.76 MHz clock and 8 kHz frame. These FIFOs will absorb delay variations between the eight channels due to timing skews between cards and along backplane traces. For greater variations, a streamlined pointer processor (pointer mover) within the STM macro will align the 8 kHz frames regardless of their incoming frame position.

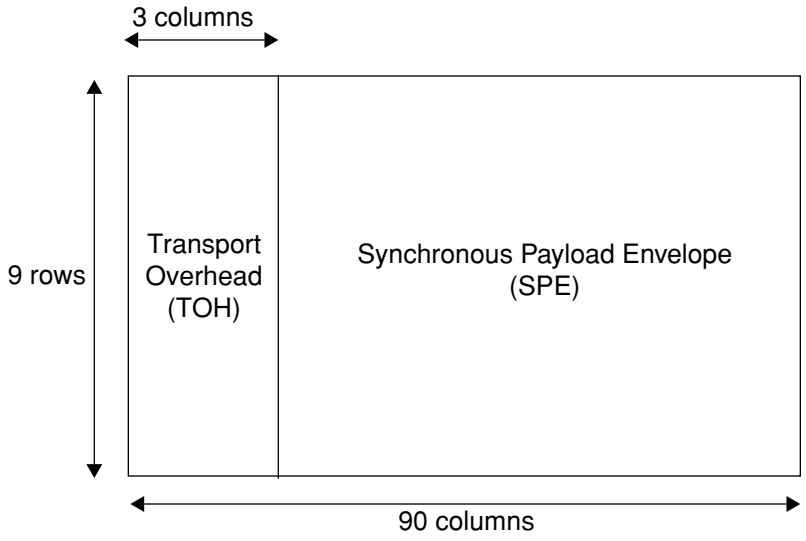
The data rates for SONET are covered in the following table. Values that fall in between those shown in the table for each mode are supported (126.00 Mbits/s - 212.50 Mbits/s, 504.00 Mbits/s - 850.00 Mbits/s). 63.00 MHz is the slowest reference clock while 106.25 MHz is the fastest reference clock frequency supported.

Table 2. Supported SONET Data Rates

Reference Clock	STS-12 Mode	STS-3 Mode
63 MHz	504.00 Mbits/s	126.00 Mbits/s
77.76 MHz	622.08 Mbits/s	155.52 Mbits/s
106.25 MHz	850.00 Mbits/s	212.50 Mbits/s

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 4. 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 ORT8850 offers both a transparent option and a serial insertion option for processing the TOH bytes.

Figure 4. STS-1 Frame Format



LVDS Reference Clock

The reference clock for the ORT8850 SERDES is an LVDS input (SYS_CLK_[P:N]). This reference clock can run in the range from 63.00 MHz to 106.25 MHz and is used to clock the entire Embedded Core. This clock is also available in the FPGA interface as the output signal FPGA_SYSCLK at the Embedded Core/FPGA Logic interface.

The supported range of reference clock frequencies will drive the internal and link serial rates from 504 MHz to 850 MHz. For standard SONET applications a reference clock rate of 77.76 MHz will allow the ORT8850 to communicate with standard SONET devices. If the ORT8850 is communicating with another ORT8850, the reference clock can run anywhere in the defined range. When using a non 77.76 MHz reference clock, the frame pulse will now need to be derived from the non standard rate thus making the frame pulse rate not 8 kHz, but rather a single clock pulse every 9720 clock cycles.

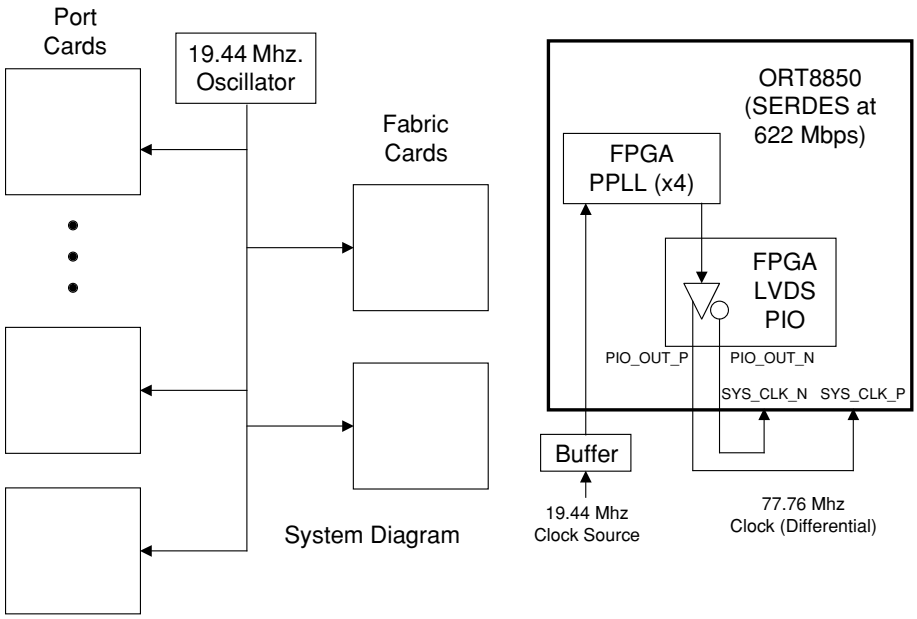
System Considerations for Reference Clock Distribution

There are two main system clocking architectures that can be used with the ORT8850 at the system level to provide the LVDS reference clocks. The recommended approach is to distribute a single reference clock to all boards. However, independent clocks can be used on each board provided that they are matched with sufficient accuracy and the alignment is not used. These two approaches are summarized in the following paragraphs

Distributed Clocking

A distributed clock architecture, shown in Figure 5, uses a single source for the system reference clock. This single source drives all devices on both the line and switch sides of the backplane. Typically this is a lower speed clock such as a 19.44 MHz signal. An external PLL on each board or and internal ORT8850 FPGA PLL is then used to multiply the clock to the desired reference clock rate (i.e. by 4x to 77.76 MHz if the distributed clock is at 19.44 MHz). Using this type of clock architecture the ORT8850 data channels are fully synchronous and no domain transfer is required from the transmitter to the receiver.

Figure 5. Distributed Clock Architecture

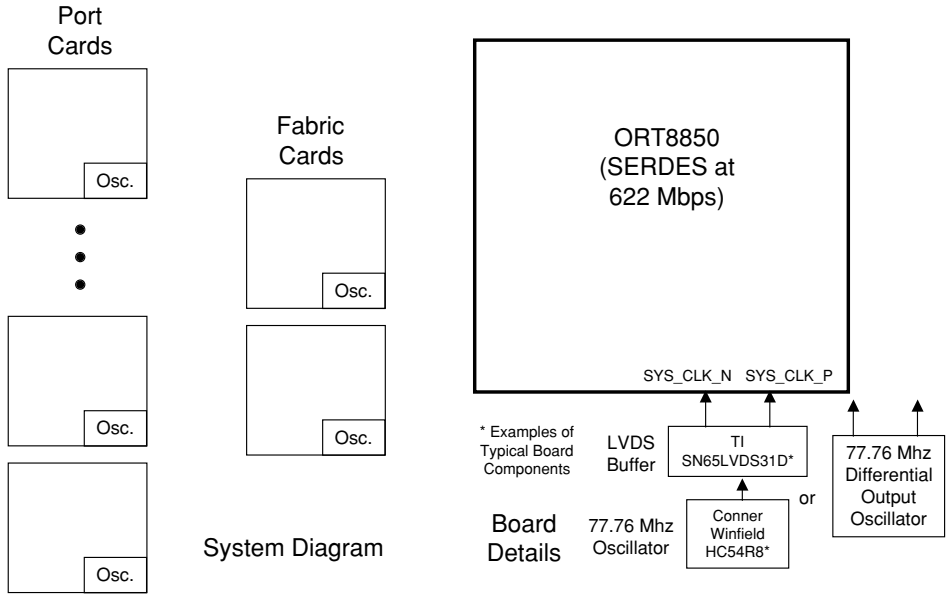


Independent Clocking

An independent clock architecture uses independent clock sources on each ORT8850 board. With this architecture, for the SERDES to sample correctly the independent oscillators must be within reference clock tolerance requirements for the Clock and Data Recovery (CDR) to correctly sample the incoming data and recover data and clock. The local reference clock and the recovered clock will not be synchronous since they are created from a different source. The alignment FIFO uses the recovered clock for write and the local reference clock for read. Due to

this feature the alignment FIFO cannot be used with this clock architecture. The recovered clock is used for all receive timing in the embedded core and supplied to the FPGA logic which must provide the clock domain transfer functionality.

Figure 6. Independent Clock Architecture



SONET Bypass Mode

It is possible to utilize only the serializer and deserializer (SERDES) blocks in the ORT8850 and to bypass all of the SONET framing and scrambling/descrambling. In this mode the parallel data from the FPGA is serialized and sent out the LVDS pins. The serial data in the receive direction will be run through the SERDES and then received as parallel data with a recovered clock into the FPGA.

In the SONET Bypass mode there exists half and quarter rate selection options. Half rate allows the SERDES to operate at 4x the reference clock. When using half rate mode only the bits 7:4 of the parallel FPGA bus are utilized. Quarter rate allows the SERDES to operate at 2x the reference clock. When using quarter rate mode only bits 7:6 of the parallel FPGA bus are utilized. Half rate and quarter rate are selectable per channel and can be mixed per channel so that some channels can run in full rate mode while others operate in half rate mode and still others operate in quarter rate mode.

As shown in Table 3, 63.00 MHz is the slowest reference clock and 106.25 MHz is the fastest reference clock frequency supported. For all three modes, all bandwidths within the reference clock limits are supported. Note that there are gaps between the bandwidths supported in the three modes.

Table 3. SONET Bypass Mode Bandwidth Options

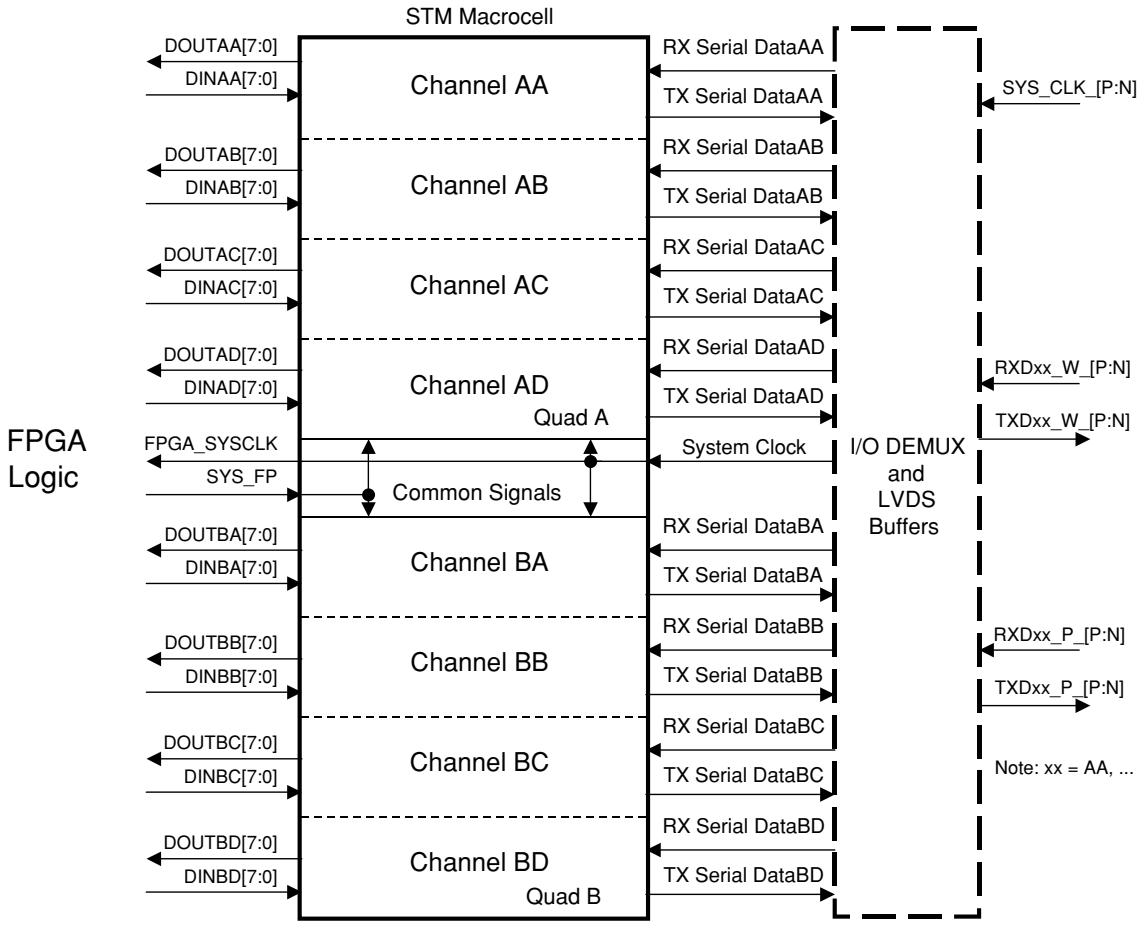
Reference Clock	Full Mode	Half Mode Bits [7:4] Used	Quarter Mode Bits [7:6] Used
63	504.00 Mbits/s	252.00 Mbits/s	126.00 Mbits/s
77.76	622.08Mbits/s	311.04Mbits/s	155.52Mbits/s
106.25	850.00 Mbits/s	425.00 Mbits/s	212.50 Mbits/s

In the SONET Bypass mode a 1's density function similar to SONET scrambling must be implemented in the FPGA logic to assure reliable clock recovery at the receiver.

STM Macrocells - Overview

The Synchronous Transport Module (STM) portion of the embedded core consists of two quads, STM A and B. The STM macrocells provide transmitter and receiver logic blocks on a per SERDES basis channel and are located in the data path between the FPGA interface and the HSI macrocell. The STM macrocells' main functions are framing and aligning data into standard STS-N frames as well as providing a 1's density through scrambling/descrambling.

Figure 7. STM Macrocell Partitioning



Transmit STM Macrocell Logic - Overview

In the transmit direction (FPGA interface to the backplane), each STM macrocell will receive frame aligned streams of STS-12 data (maximum of four streams) from the FPGA logic. The transmitter receive data interface is in a parallel 8-bit format. A common frame pulse for all 8 channels is provided as an input from the FPGA logic to the transmit SONET block.

On each frame pulse the A1/A2 frame alignment bytes are inserted into the data stream and will overwrite any data in this location of the frame. TOH data can be optionally inserted into the transmitted SONET frame. The SONET frame is then optionally scrambled and sent to the HSI macrocell.

TOH data can be inserted into the transmit data stream in two ways; transparently or by inserting serial TOH data from a TOH serial interface signal in the FPGA logic. In the transparent mode, the SPE and TOH data received on parallel input bus is transferred, unaltered, to the serial LVDS output. However, B1 byte of STS-1 is always replaced with a new calculated value (the 11 bytes following B1 are replaced with all zeros). Likewise, in serial and transport mode A1 and A2 bytes of all STS-1s are always regenerated. In the TOH serial insertion mode the SPE bytes are transferred unaltered from the input parallel bus to the serial LVDS output. TOH bytes, however, are received from the FPGA logic through the serial input port and are inserted in the STS- 12 frame before being sent to the LVDS

output. Although all TOH bytes from the 12 STS-1s are transferred into the device from each serial port, not all of them get inserted in the frame. There are three hard coded exceptions to the TOH byte insertion:

- Framing bytes (A1/A2 of all STS-1s) are not inserted from the serial input bus. Instead, they can always be regenerated.
- Parity byte (B1 of STS#1) is not inserted from the serial input bus. Instead, it is always recalculated (the 11 bytes following B1 are replaced with all zeros).
- Pointer bytes (H1/H2/H3 of all STS-1s) are not inserted from the serial input bus. Instead, they always flow transparently from parallel input to LVDS output.

The data stream is scrambled in the transmit direction and descrambled in the receive direction using a frame synchronous scrambler of sequence length 127, operating at the line rate. The generating polynomial for the scrambler is $1+x^6+x^7$. The polynomial conforms to the standard SONET STS-12 data format. The scrambler is reset to '1111111' on the first byte of the SPE (byte following the Z0 byte in the 12th STS-1). That byte and all subsequent bytes to be scrambled are XOR'd, with the output from the bytewise scrambler. The scrambler runs continuously from that byte, through the remainder of the frame. A1, A2, and J0/Z0 bytes are not scrambled. The B1 byte is calculated (in both transmitter and receiver) on the non-scrambled data. There is a global scrambler/descrambler disable feature, allowing the user to disable the scrambler of the transmitter and the descrambler of the receiver. Following the scrambler block, byte wide data streams are sent to the HSI macrocell.

Receive STM Macrocell Logic - Overview

In the receive direction (backplane to the FPGA interface) each STM macrocell receives four byte wide data streams at the reference clock rate (i.e., 8 X SYS_CLK_[P:N] in normal operation) and four associated clocks from the HSI. The incoming streams are framed and (optionally) descrambled before they are written into a FIFO which absorbs phase and delay variations and allows the shift to system clock and optionally allows frames to be aligned both between quads and between streams on the same quad. Optionally, the pointer interpreter logic will then put the STS SPEs into a small elastic store from which the pointer generator will produce four byte wide STS-12 streams of data that are aligned to the system timing pulse.

The alignment FIFO depth allows for 18 clocks of difference in the arriving A1/A2. If any of the channels in an alignment group are too far out of alignment for the FIFO to absorb the difference an alarm register will indicate the error. Alarm indicators can be programmed to trigger an alarm at different levels of misalignment.

The multichannel alignment option allows separate SERDES data channels to be byte aligned based on the SONET A1/A2 bytes. Data is written into the alignment FIFO using the per channel recovered clocks from the SERDES channel. Data is always read from the alignment FIFO using the local reference clock. (SYS_CLK pin, FPGA_SYS CLK)

SERDES data channels can be placed into an alignment group by 2, by 4, or all 8. In by 2 mode, channels AA and BA, AB and BB, AC and BC, and AD and BD are byte aligned. In by 4 mode channels AA, AB, AC, AD and BA, BB, BC, BD are byte aligned. In the by 8 mode all of the channels are byte aligned.

After the alignment FIFO the receive data can optionally go through the pointer interpreter and pointer mover. The pointer interpreter will identify the SONET payload envelope (SPE) and the C1(J0) bytes and the J1 bytes. For data applications where the user is simply using SONET to carry user defined cells in the payload the SPE signal is very useful as an enable to the cell processor. C1J1 for data applications can be ignored. If the pointer interpreter and pointer mover are bypassed, then the SPE and C1J1 signals to the FPGA logic will be always '0'.In the ORT8850 each frame consists of 12 STS-1 format sub-frames. Thus, in the SPE region, there are 12 J1 pulses, one for each STS-1. There is one C1(J0) (current SONET specifications use J0 instead of C1 as section trace to identify each STS-1 in an STS-N) pulse in the TOH area for one frame. Thus, there are a total of 12 J1 pulses and one C1(J0) pulse per frame. The C1(J0) pulse is coincident with the J0 of STS-1 #1 which is the first byte following the last A2 byte.

With the pointer interpreter option enabled, the SPE flag is active when the data stream is in SPE area. SPE behavior is dependent on pointer movement and concatenation. Note: in the TOH area, H3 can also carry valid

data. When valid SPE data is carried in this H3 slot, SPE is high in this particular TOH time slot also. In the SPE region, if there is no valid data during any SPE column, the SPE signal will be set to low.

After the pointer interpreter comes the pointer mover block. There is a separate pointer mover for each of the two SONET quads, A and B, each of which handles up to one STS-48 (four channels) The K1/K2 bytes and H1-SS bits are also passed through to the pointer generator so that the FPGA can receive them. The pointer mover handles both concatenations inside the STS-12, and to other STS-12s inside the core. The pointer mover block can correctly process any length of concatenation of STS frames (multiple of three) as long as it begins on an STS-3 boundary (i.e., STS-1 number one, four, seven, ten, etc.) and is contained within the smaller of STS-3, 12, or 48.

The pointer generator block then maps the corresponding bytes into their appropriate location in the outgoing byte stream. The generator also creates offset pointers based on the location of the J1 byte as indicated by the pointer interpreter.

HSI Macrocell - Overview

The HSI macrocell consists of three functionally independent blocks: receiver, transmitter, and PLL synthesizer. The HSI logic is used for Clock/Data Recovery (CDR) and to serialize and deserialize between the 106.25 MHz byte-wide internal data buses and the 850 Mbits/s serial LVDS links. For a 622 Mbits/s SONET stream, the HSI will perform Clock and Data Recovery (CDR) and MUX/DEMUX between 77.76 MHz byte-wide internal data buses and 622 Mbits/s serial LVDS links.The transmitter block receives parallel data at its input. The MUX (serializer) module performs a parallel-to-serial conversion using a clock provided by the PLL/synthesizer block. The resulting serial data stream is then transmitted through the LVDS driver.

The receiver block receives a LVDS serial data without clock at its input. Based on data transitions, the receiver selects an appropriate clock phase for each channel to retime the data. The retimed data and clock are then passed to the DEMUX (deserializer) module. The DEMUX module performs serial-to-parallel conversion and provides parallel data and clock to the SONET framer block.

Supervisory and Test Support Features - Overview

The supervisory and test support functions provided by the ORT8850 include data integrity monitoring, error insertion capabilities and loopback support. These functions are described in the following sections.

Integrity Monitoring

FPGA Parallel Bus Integrity: Parity error checking is implemented on each of the four parallel input buses on each STM quad (A & B). "Even" or "Odd" parity can be selected by setting a control register bit. Upon detection of an error, an alarm bit is set. This feature is on a per channel basis. Note that, on parallel output ports, parity is calculated over the 8-bit data bus and not on the SPE and C1J1 lines.

TOH Serial Port Integrity: There is “even” parity generation on each of the four TOH serial output ports. There is “even” parity error checking on each of the four TOH serial input ports. There is one parity bit embedded in the TOH frame. It occupies the Most Significant Bit location of A1 byte of STS#1. Upon detection of an error, an alarm bit is set. This feature is on a per channel basis.LVDS Link Integrity: There is B1 parity generation on each of the four LVDS output channels. There is also performance monitoring on each of the four LVDS input channels, implemented as B1 parity error checking. Upon detection of an error, a counter is incremented (one count per errored bit) and an alarm bit is set. The counter is 7-bits wide plus 1 overflow indicator bit. This feature is on a per channel basis.

Framer Monitor: The framer in the receive direction will report Loss of Frame by setting an alarm bit, as well as a LOF count and errored frame count. The LOF alarm bit is not clearable as long as the channel is in the LOF state. In addition, the errored frame count represents errored frames, and will not increment more than once per frame even if there are multiple errors.

Receiver Internal Path Integrity: There is "even" parity generation in the Receiver section (after descrambler). There is also "even" parity error checking in the Receiver section (before output). Upon detection of an error, an alarm bit is set. This feature is on a per channel basis.

Pointer Mover Performance Monitoring: There is Pointer Mover performance monitoring in the Receiver section. Alarm Indication Signals (AIS-P) and elastic store overflows are reported. AIS-P is implemented as a per STS-1 alarm bit. Elastic store overflow will cause an alarm bit to be set on a per STS-1 basis.

FIFO Aligner Monitoring: There is monitoring of the FIFO aligner operating point, and upon deviating from the nominal operating point of the FIFO by more than user programmable threshold values (min and max threshold values), an alarm bit is set. Threshold values are defined per device; alarm flags are per channel.

Frame Offset Monitoring: There is monitoring of the frame offset between all enabled channels (disabled channels do not interfere with the monitoring). Monitoring is performed continuously. Upon exceeding the maximum allowed frame offset (18 bytes) between all enabled channels, an alarm bit is set.

Error Insertion

A1/A2 Error Insert: There is a Frame Error inject feature in the transmitter section, allowing the user to replace framing bytes A1/A2 (only last A1 byte and first A2 byte) with a selectable A1/A2 byte value for a selectable number of consecutive frames. The number of consecutive frames to alter is specified by a 4-bit field, while A1/A2 value is specified by two 8-bit fields. The error insert feature is on a per channel basis, A1/ A2 values and 4-bit frame count value are on a per device basis.

B1 Error Insert: There is a B1 error insert feature in the transmitter section, allowing the user to insert errors on user selectable bits in the B1 byte. Errors are created by simply inverting bit values. Bits to invert are specified through an 8-bit control. To insert an error, software will first set the bits in the "transmitter B1 error insert mask". Then, on a per channel basis software will write a one to the "B1 error insert command". The insertion circuitry performs a rising edge detect on the bit, and will issue a corruption signal for the next frame, for one frame only. This feature is on a per channel basis.

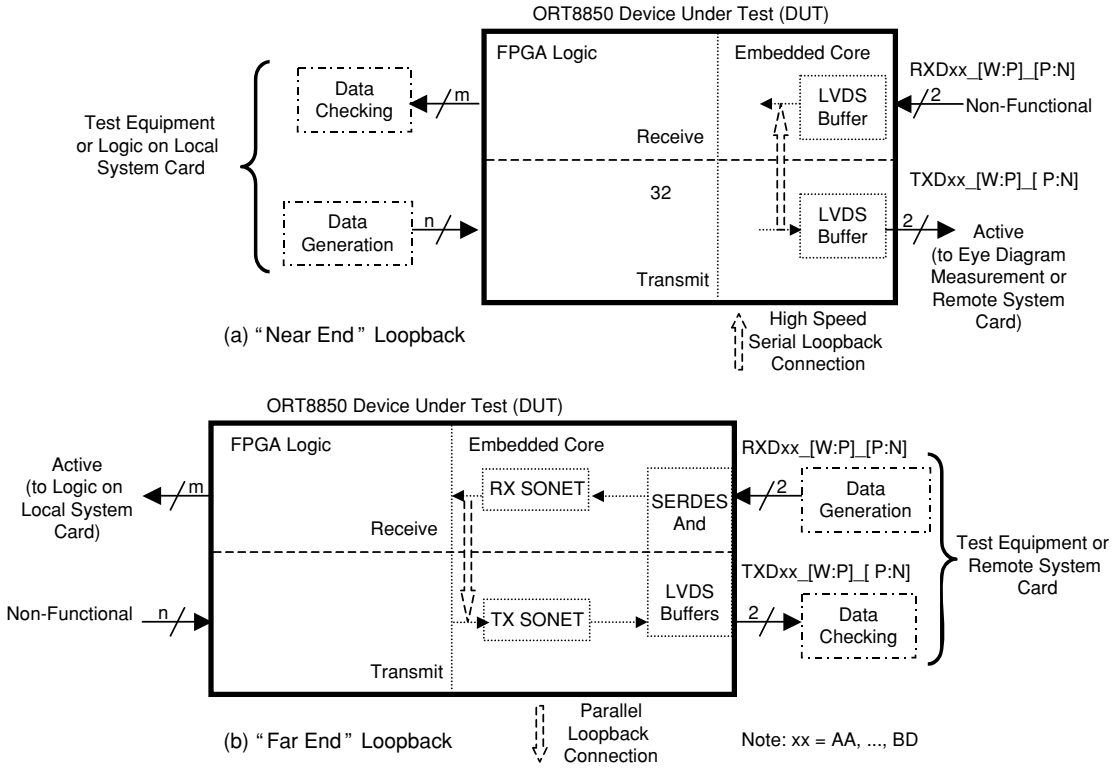
TOH Serial Output Port Parity Error Insert: There is a Parity error inject feature, in the receive section, allowing the user to invert the parity bit of each serial output port. This feature inserts a single error. This feature is on a per channel basis.

Parallel Output Bus Parity Error Insert: There is a Parity error inject feature, in the receive section, allowing the user to invert parity lines (DOUTxx_PAR) associated with each output parallel busses (DOUTxx[7:0]). This feature inserts a single error. This feature is on a per channel basis. This feature supports both 'even' and 'odd' parities.

Loopback

There are two types of loopback that can be utilized inside the embedded ASIC core of the ORT8850, near end loopback and far end (line side) loopback. Both of these loopbacks are controlled by control registers inside the ORT8850 core, which are accessible from the system bus and the MicroProcessor Interface (MPI). In both loopback modes, all channels are placed with a single control. The data paths in the two loopback modes are shown in Figure 8.

Figure 8. Data Paths for Near-End and Far-End Loopbacks (Single Channel)



Near end loopback is a loopback of data from the FPGA transmit into the core and back out of the core to the FPGA. This loopback mode is good for simulation since two ORT8850 devices do not have to be included in the simulation test bench. It is also ideal for system debugging when only working with a single card. There are two depths to the near end loopback, CDR and LVDS. The CDR near end loopback performs the loopback inside the CDR itself. LVDS near end loopback does the loopback just before data is sent out of the LVDS transmit pins. In all near end loopbacks the transmit data is still sent out of the LVDS pins.

Far end loopback is a loopback of the high speed data on the backplane side. Serial data is transmitted into the device from the backplane and then looped back to the backplane side. This loopback is good for backplane connectivity tests and backplane integrity type tests. The actual loopback of data is performed inside the Pointer Mover Block. In this mode the SYS_FP signal from the FPGA logic to the Embedded Core must provide an 8KHz frame pulse. It should also be noted that during the bypass of the Pointer Mover Block, the Far End Loopback cannot be performed inside the Embedded ASIC Block. In that case it can be coded to be performed inside the FPGA.

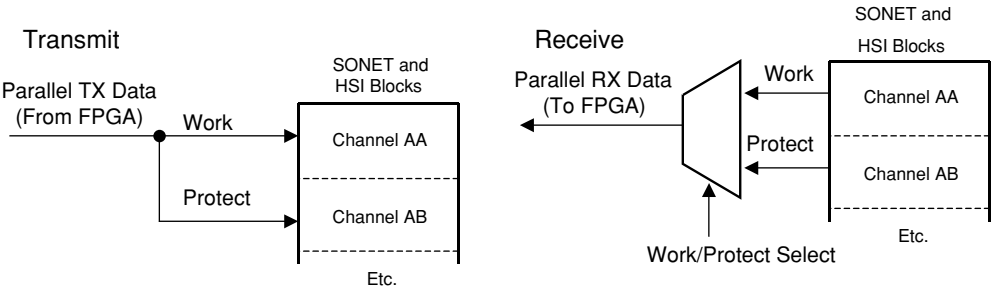
Protection Switching - Overview

The ORT8850 supports 1:1 redundancy within both the transmit and receive data paths. Work/protect selection is controlled by a control register bit which can be set using the system bus or the external MicroProcessor Interface. Protection switching allows a pair of SERDES channels to act as main and protect data links. On the transmit side, a simple broadcast mode is used and the same data is transmitted across both work and protect interfaces.

All data channels have receive work and protect switching capability. There are two types of receive protection switching supported. The switching can be performed at the parallel interface to the FPGA or at the interface to the LVDS buffers. Parallel protection switching (Figure 9) takes place just before the FPGA interface ports and after the alignment FIFO. The alignment FIFO must be used for this type of protection switching. In this mode SERDES channels AA and AB are used as main and protect. When selected for main channel AA is used to provide data on FPGA interface ports AA. When selected for protect channel AB is used to provide data on FPGA interface ports

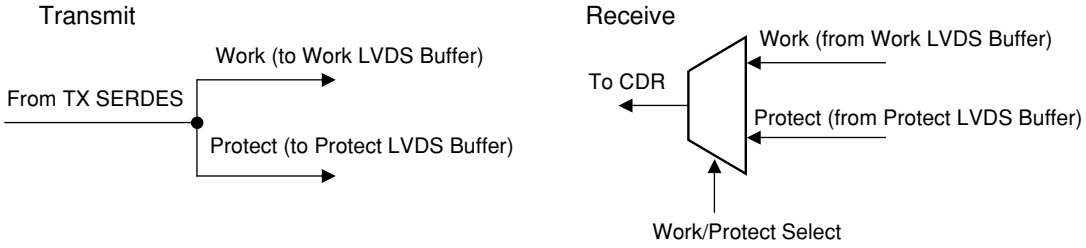
AA. This same scheme is used for channels groupings of AC/AD, BA/BB, and BC/BD. For quad protection when the alignment FIFOs are to be used, the protection switching must be done in FPGA logic.

Figure 9. Parallel Protection Switching



LVDS protection switching (Figure 10) takes place at the LVDS buffer before the serial data is sent into the CDR. The selection is between the main LVDS buffer and the protect LVDS buffer. The main LVDS buffer provide the main receive data on RXDxx_W_[P:N] while the protect LVDS buffers provide protection receive data on RXDxx_P_[P:N]. When operating using the main LVDS buffers (default) no status information is available on the protect LVDS buffers since the serial stream must reach the SONET framer before status information is available on the data stream. The same is also true for the main LVDS buffers when operating with the protect buffers.

Figure 10. LVDS Protection Switching



See Table 17 and Table 18 and the accompanying text for details and register settings for the protection switching options.

FPSC Configuration - Overview

Configuration of the ORT8850 occurs in two stages: FPGA bit stream configuration and embedded core setup.

FPGA Configuration - Overview

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 the 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* technical note (TN1017). This IP block sets up the embedded core via a state machine and allows the ORT8850 to work in an independent system without an external MicroProcessor Interface.

Embedded Core Setup

All options for the operation of the core are configured according to the memory map shown in Table 19.

During the power-up sequence, the ORT8850 device (FPGA programmable circuit and the core) is held in reset. All the LVDS output buffers and other output buffers are held in 3-state. All Flip-Flops in the core area are in reset state, with the exception of the boundry-scan shift registers, which can only be reset by boundary-scan reset. After power-up reset, the FPGA can start configuration. During FPGA configuration, the ORT8850 core will be held in

reset and all the local bus interface signals forced high, but the following active-high signals, PROT_SWITCH_AA, PROT_SWITCH_AC, PROT_SWITCH_BA, PROT_SWITCH_BC, TX_TOH_CK_EN, SYS_FP, LINE_FP, will be forced low. The CORE_READY signal sent from the embedded core to FPGA is held low, indicating that the core is not ready to interact with FPGA logic. At the end of the FPGA configuration sequence, the CORE_READY signal will be held low for six SYS_CLK cycles after DONE, TRI_IO and RST_N (core global reset) are high. Then it will go active-high, indicating the embedded core is ready to function and interact with FPGA programmable circuit. During FPGA reconfiguration when DONE and TRI_IO are low, the CORE_READY signal sent from the core to FPGA will be held low again to indicate the embedded core is not ready to interact with FPGA logic. During FPGA partial configuration, CORE_READY stays active. The same FPGA configuration sequence described previously will repeat again.

The initialization of the embedded core consists of two steps: register configuration and synchronization of the alignment FIFO. The steps to configure the ORT8850 device for normal operation are listed in Table 4 and Table 5.

Generic Backplane Transceiver Application

Independent Channels, Transparent TOH: Table 4 lists the register values to setup the ORT8850 as eight independent SONET channels (no alignment) using transparent TOH. The order is specific. The values are given from the PowerPC point of view. If using the MPI to write data to the ORT8850, the value given in the table is the value that should be used. If using the UMI of the system bus, the data value would need to be byte flipped.

Table 4. Independent Channels, Transparent TOH

Register Address	Value	Description
0x30004	0x05	Lock register. This value must be written to allow writing to any other ORT8850 core register
0x30005	0x80	Lock register. This value must be written to allow writing to any other ORT8850 core register
0x30020	0x07	Turn on Channel AA in functional mode
0x30021	0xFF	Channel AA - Transparent TOH from parallel data
0x30022	0xFF	Channel AA - Transparent TOH from parallel data
0x30038	0x07	Turn on Channel AB in functional mode
0x30030	0xFF	Channel AB - Transparent TOH from parallel data
0x3003A	0xFF	Channel AB - Transparent TOH from parallel data
0x30050	0x07	Turn on Channel AC function mode
0x30051	0xFF	Channel AC - Transparent TOH from parallel data
0x30052	0xFF	Channel AB - Transparent TOH from parallel data
0x30068	0x07	Turn on Channel AD in functional mode
0x30069	0xFF	Channel AD - Transparent TOH from parallel data
0x3006A	0xFF	Channel AD - Transparent TOH from parallel data
0x30080	0x07	Turn on Channel BA functional mode
0x30081	0xFF	Channel BA- Transparent TOH from parallel data
0x30082	0xFF	Channel AD - Transparent TOH from parallel data
0x30098	0x07	Turn on Channel BB in functional mode
0x30099	0xFF	Channel BB- Transparent TOH from parallel data
0x3009A	0xFF	Channel BB- Transparent TOH from parallel data
0x300B0	0x07	Turn on Channel BC in functional mode
0x300B1	0xFF	Channel BC- Transparent TOH from parallel data
0x300B2	0xFF	Channel BC - Transparent TOH from parallel data
0x300C8	0x07	Turn on Channel BD in functional mode
0x300C9	0xFF	Channel BD - Transparent TOH from parallel data
0x300CA	0xFF	Channel BD - Transparent TOH from parallel data

Channel Alignment, Transparent TOH: Table 5 lists the register values to setup the ORT8850 as 4 Channel alignment SONET channels using transparent TOH. The order is specific. The values are given from the PowerPC point of view. If using the MPI to write data to the ORT8850, the value given in the table is the value that should be used. If using the UMI of the system bus, the data value would need to be byte flipped.

Table 5. Channel Alignment, Transparent TOH

Register Address	Value	Description
Initial Register Settings		
0x30004	0x05	Lock register. This value must be written to allow writing to any other ORT8850 core register
0x30005	0x80	Lock register. This value must be written to allow writing to any other ORT8850 core register
0x30020	0x47	Turn on Channel AA in functional mode with AIS-L
0x30021	0xFF	Channel AA - Transparent TOH from parallel data
0x30022	0xFF	Channel AA - Transparent TOH from parallel data
0x30037	0x08	Channel AA- Aligned by 4 (Quad A)
0x30038	0x47	Turn on Channel AB function mode with AIS-L
0x30039	0xFF	Channel AB - Transparent TOH from parallel data
0x3003A	0xFF	Channel AB - Transparent TOH from parallel data
0x3004F	0x08	Channel AB - Aligned by 4 (Quad A)
0x30050	0x47	Turn on Channel AC function mode with AIS-L
0x30051	0xFF	Channel AC - Transparent TOH from parallel data
0x30052	0xFF	Channel AD - Transparent TOH from parallel data
0x30067	0x08	Channel AC- Aligned by 4 (Quad A)
0x30068	0x47	Turn on Channel AD function mode with AIS-L
0x30069	0xFF	Channel AD- Transparent TOH from parallel data
0x3006A	0xFF	Channel AD - Transparent TOH from parallel data
0x3007F	0x08	Channel AC- Aligned by 4 (Quad A)
0x30080	0x47	Turn on Channel BA function mode with AIS-L
0x30081	0xFF	Channel BA- Transparent TOH from parallel data
0x30082	0xFF	Channel BA- Transparent TOH from parallel data
0x30097	0x08	Channel BA- Aligned by 4 (Quad B)
0x30098	0x47	Turn on Channel BB function mode with AIS-L
0x30099	0xFF	Channel BB- Transparent TOH from parallel data
0x3009A	0xFF	Channel BB - Transparent TOH from parallel data
0x300AF	0x08	Channel BB - Aligned by 4 (Quad B)
0x300B0	0x47	Turn on Channel BC function mode with AIS-L
0x300B1	0xFF	Channel BC - Transparent TOH from parallel data
0x300B2	0xFF	Channel BC - Transparent TOH from parallel data
0x300C7	0x08	Channel BC - Aligned by 4 (Quad B)
0x300C8	0x47	Turn on Channel BD function mode with AIS-L
0x300C9	0xFF	Channel BD - Transparent TOH from parallel data
0x300CA	0xFF	Channel BD - Transparent TOH from parallel data
0x300DF	0x08	Channel BD - Aligned by 4 (Quad B)
Wait for 4 SONET Frames to establish an in-frame state (~500us)		
0x30018	0x0C	Alignment command to resync Quad A and Quad B, then modify register settings as follows. Write 0x00 to clear register for normal operation.
0x30020	0x07	Channel AA in functional mode without AIS-L

Table 5. Channel Alignment, Transparent TOH (Continued)

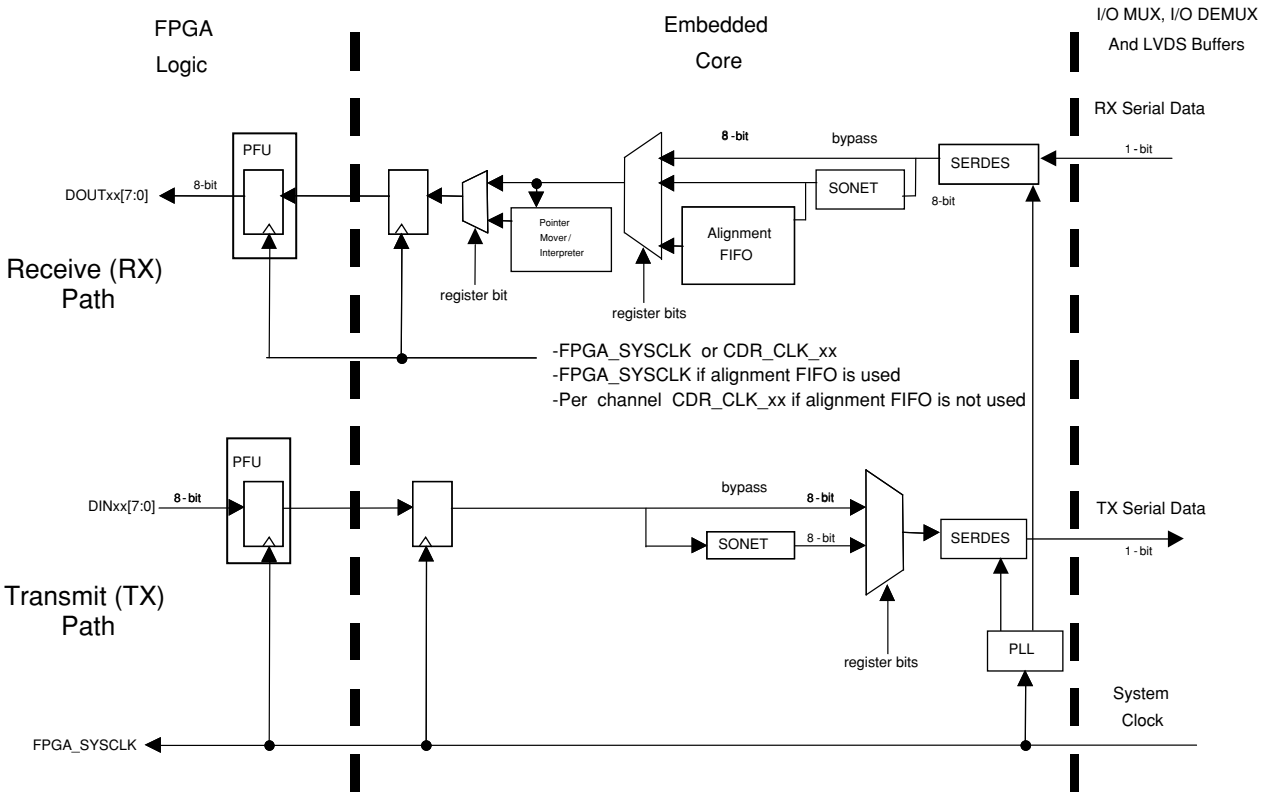
Register Address	Value	Description
Initial Register Settings		
0x30038	0x07	Channel AB in functional mode without AIS-L
0x30050	0x07	Channel AC in functional mode without AIS-L
0x30068	0x07	Channel AD in functional mode without AIS-L
0x30080	0x07	Channel BA in functional mode without AIS-L
0x30098	0x07	Channel BB in functional mode without AIS-L
0x300B0	0x07	Channel BC in functional mode without AIS-L
0x300C8	0x07	Channel BD in functional mode without AIS-L

Backplane Transceiver Core Detailed Description

SONET Logic Blocks, Detailed Description

The following sections describe the data processing performed in the SONET logic blocks. A 622 Mbits/s is assumed in the descriptions however, as noted in the Overview sections, the ORT8850 can operate at variable rates up to 850 Mbits/s. 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). A top level drawing of the two data paths and associated clocks is shown in Figure 11. 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 or the MicroProcessor Interface (MPI). Memory maps and descriptions for the registers are given in Table 19.

Figure 11. ORT8850 Top Level Data Flow



Byte Ordering in SONET Frames

The ORT8850 expects byte ordering in the SONET frames to be in the standard byte interleaved format per the GR-253 SONET standard. Byte ordering is the same in both the transmit and receive direction and treats the data as multiple STS-1 frames. When using the ORT8850 in STS-3 format both the transmitter and receiver device must be framed, based on STS-3 format. Likewise for the STS-12 format, both must operate in STS-12 format.

Table 6. Byte Ordering, STS-3 Format

STS-3 A -->	3	3	3	3	2	2	2	2	1	1	1	1
STS-3 B -->	3	3	3	3	2	2	2	2	1	1	1	1
STS-3 C -->	3	3	3	3	2	2	2	2	1	1	1	1
STS-3 D -->	3	3	3	3	2	2	2	2	1	1	1	1

Table 7. Byte Ordering, STS-12 Format

STS-12 A -->	12	9	6	3	11	8	5	2	10	7	4	1
STS-12 B -->	12	9	6	3	11	8	5	2	10	7	4	1
STS-12 C -->	12	9	6	3	11	8	5	2	10	7	4	1
STS-12 D -->	12	9	6	3	11	8	5	2	10	7	4	1

Table 8. Byte Ordering, Quad STS-12 (OC-48) Format

STS-12 A -->	12	9	6	3	11	8	5	2	10	7	4	1
STS-12 B -->	24	21	18	15	23	20	17	14	22	19	16	13
STS-12 C -->	36	33	30	27	35	32	29	26	34	31	28	25
STS-12 D -->	48	45	42	39	47	44	41	38	46	43	40	37

All internal framing is based on the system frame pulse (SYS_FP) which is a one-cycle pulse at an 8kHz rate. There is one system frame pulse for all 8 channels or both quads. When the framer receives the system frame pulse the individual overhead bytes are identified.

HSI Macrocell

The ORT8850 High-Speed Interface (HSI) provides a physical medium for high-speed asynchronous serial data transfer between ASIC devices. The devices can be mounted on the same PC board or mounted on different boards and connected through the shelf back-plane. The ORT8850 CDR macro is an eight-channel Clock-Phase Select (CPS) and data retime function with serial-to-parallel demultiplexing for the incoming data stream and parallel-to-serial multiplexing for outgoing data. The ORT8850 uses an eight-channel HSI macro cell. The HSI macro consists of three functionally independent blocks: receiver, transmitter, and PLL synthesizer.

The PLL synthesizer block generates the necessary 850 MHz clock for operation from a 106.25 MHz, reference. The PLL synthesizer block is a common asset shared by all eight receive and transmit channels. The PLL reference clock must match the interface frequency.

The HSI_RX block receives differential 850 Mbits/s serial data without clock at its LVDS receiver input. Based on data transitions, the receiver selects an appropriate 850 MHz clock phase for each channel to retime the data. The retimed data and clock are then passed to the deMUX (deserializer) module. DeMUX module performs serial-to-parallel conversion and provides the 106 Mbits/s data and clock.

The HSI_TX block receives 106 Mbits/s parallel data at its input. MUX (serializer) module performs a parallel-to-serial conversion using an 850 MHz clock provided by the PLL/synthesizer block. The resulting 850 Mbits/s serial data stream is then transmitted through the LVDS driver.

The loopback feature built into the HSI macro provides looping of the transmitter data output into the receiver input when desired.

All rate examples described here are the maximum rates possible. The actual HSI internal clock rate is determined by the provided reference clock rate. For example, if a 77.76 MHz reference clock is provided, the HSI macro will operate at 622 Mbits/s.

Transmit Path Logic

In the transmit direction each STM quad will receive frame aligned streams of STS-12 data (maximum of four streams per quad) from the FPGA logic. The transmitter receives data interface in a parallel 8-bit format. A common frame pulse for all 8 channels is provided as an input from the FPGA logic to the transmit SONET block.

The system frame pulse is a single pulse at the reference clock rate every 9720 clock cycles. For a 77.76 MHz reference clock this creates an 8KHz pulse rate. The system frame pulse (SYS_FP) is used to generate the A1/A2 in the transmit direction. It is also used by the Pointer Mover Block to perform the line side loopback, which otherwise uses the LINE_FP frame pulse also provided by the user from the FPGA to the Embeddded ASIC Block. The Function of the LINE_FP is mentioned in the Pointer Mover bypass description.

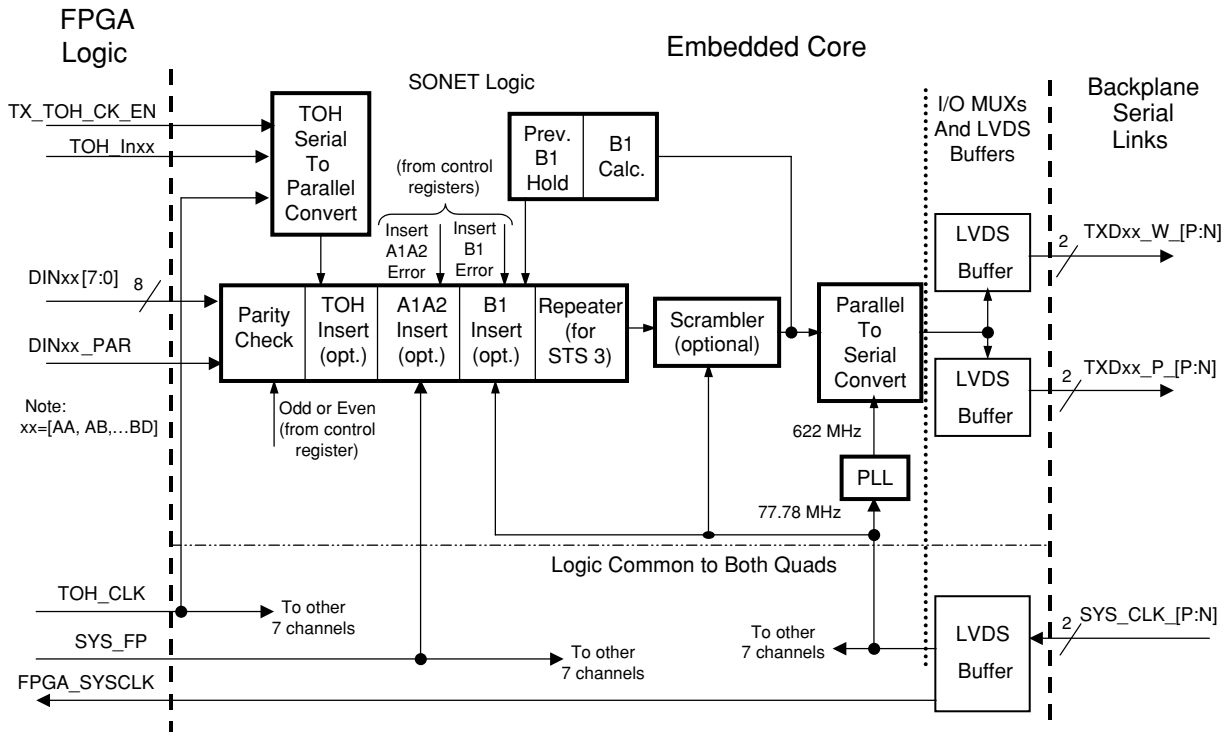
The system frame pulse is common to all channels in the transmit direction. Once it is received from the FPGA logic, the data to be transmitted goes through the following processing steps:

- A parity check is performed on the data
- The Transport Overhead (TOH) data is modified (optional)

- A1 and A2 framing bits are inserted (errored bits may optionally be inserted)
- The bit interleaved parity bit (B1) for the previously transmitted frame is inserted
- The data is scrambled using the standard STS-12 polynomial (optional)
- A parallel to serial conversion is performed on the data
- The serial data is broadcast to the work and protect LVDS buffers

These processing steps are described in more detail in the following sections. A block diagram of the transmit path logic is shown in Figure 12. All processing except the parallel to serial conversion is optional. If all processing except the SERDES is deselected, the device is said to be operating in the "bypass" mode.

Figure 12. Basic Logic Blocks, Transmit Path, Single Channel



Parity Checking

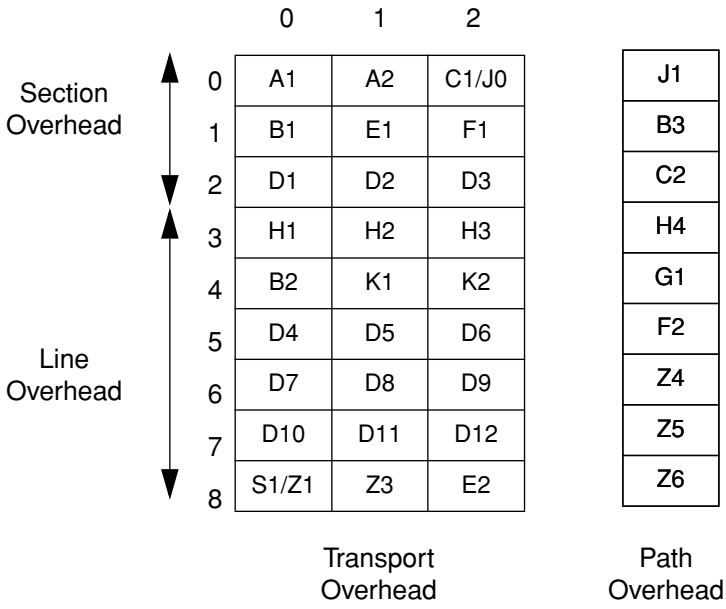
Parity error checking is implemented on each of the four parallel input buses on each STM quad (A & B) on a per channel basis. "Even" or "Odd" parity can be selected by setting a control register bit. Upon detection of an error, an alarm bit in a status register is set.

There is also even parity error checking on each of the four TOH serial input ports on a per channel basis. Upon detection of an error, an alarm bit in a status register is set.

TOH Byte Modification

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 ORT8850 in-band signaling can be efficiently utilized, since the total cost of overhead is only 3.3%. TOH data can be inserted into the transmit data stream in one of two ways, the Transparent Insertion mode and the Serial TOH Insertion mode. The overhead bytes in an STS-1 header are shown in Figure 13. (The path overhead bytes are in the SPE.)

Figure 13. SONET Overhead Bytes



When used in true SONET applications, most TOH bytes would be generated in the FPGA logic or by an external device. The TOH bytes have the following functions. Table 9 and Table 10 show how the Embedded Core modifies these bytes in the transmit direction and Table 12 shows how the bytes are modified in the receive direction.

Section Overhead Bytes:

- A1, A2 - These bytes are used for framing and to mark the beginning of a SONET frame. A1 has the value 0xF6 and A2 has the value 0x28.
- C1/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 if using serial TOH insertion. In transparent TOH mode the other bytes are passed through from DINxx bus.
- E1 - Section orderwire byte - This byte carries local orderwire information, which provides for a 64 Kbits/s 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-1signals 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 Kbits/s 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 (as defined in GR-253) 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.

In the ORT8850 transmit path, the TOH processing is confined to the framing and byte interleaved parity bytes. The remaining bytes are either passed through transparently or inserted from data sent from the serial TOH interface.

In the receive direction, the TOH bytes are stripped and optionally sent to the FPGA logic through the serial TOH interface. Regenerated framing bytes are sent to the FPGA on the parallel data bus. The APS bytes K1 and K2 can be optionally passed through the pointer mover under software control, or can be set to zero. The header bytes, J0 and C1 are also detected and used by the receive path, as will be discussed in a later section.

The serial TOH processing block is clocked by the TOH_CLK. If using the TOH bytes in the serial insertion mode to support a communication channel, this TOH_CLK should be driven from the FPGA interface. The TOH processor operates from 25 MHz to 106 MHz. A domain clock transfer takes place inside the TOH block and the TOH processor does not need to run as fast as the data.

Transparent Insert Mode

In the transmit direction the SPE and TOH data received on parallel input bus is transferred unaltered to the serial LVDS output. However, B1 byte of STS-1 is always replaced with a new calculated value (the 11 bytes following B1 are replaced with all zeros). Also, A1 and A2 bytes of all STS-1s are always regenerated. The source for the TOH bytes in the transparent mode is summarized in Table 9. (The order of transmission is row by row, left to right, and then from top to bottom [most significant bit first]. SPE bytes are not shown.)

Table 9. Transmitter TOH on LVDS Output (Transparent Mode)

A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2												
B1	0	0	0	0	0	0	0	0	0	0	0																								

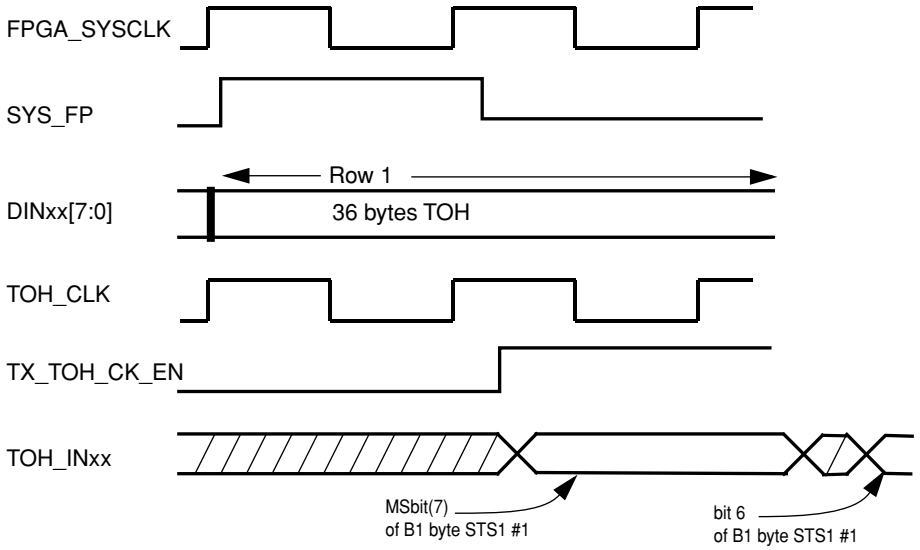
Regenerated bytes.

Transparent bytes from parallel input port.

Serial TOH Insertion Mode

In the transmit direction the SPE bytes are always transferred unaltered from the input parallel bus to the serial LVDS output. On the other hand, TOH bytes are received from the serial input port and are inserted in the STS-12 frame before being sent to the LVDS output in the Serial TOH Insertion mode. The FPGA logic must provide framing information to the Core using the TX_TOH_CK_EN Input signal. TOH data is input on a row by row basis, with a one clock cycle frame pulse delineating the start of a row, as shown in Figure 14. As shown in the figure, while the SPE bytes are being transmitted for one row, the FPGA logic must simultaneously supply the Core with the TOH data for the next row. Detailed timing for the TOH serial input is shown later in Figure 31.

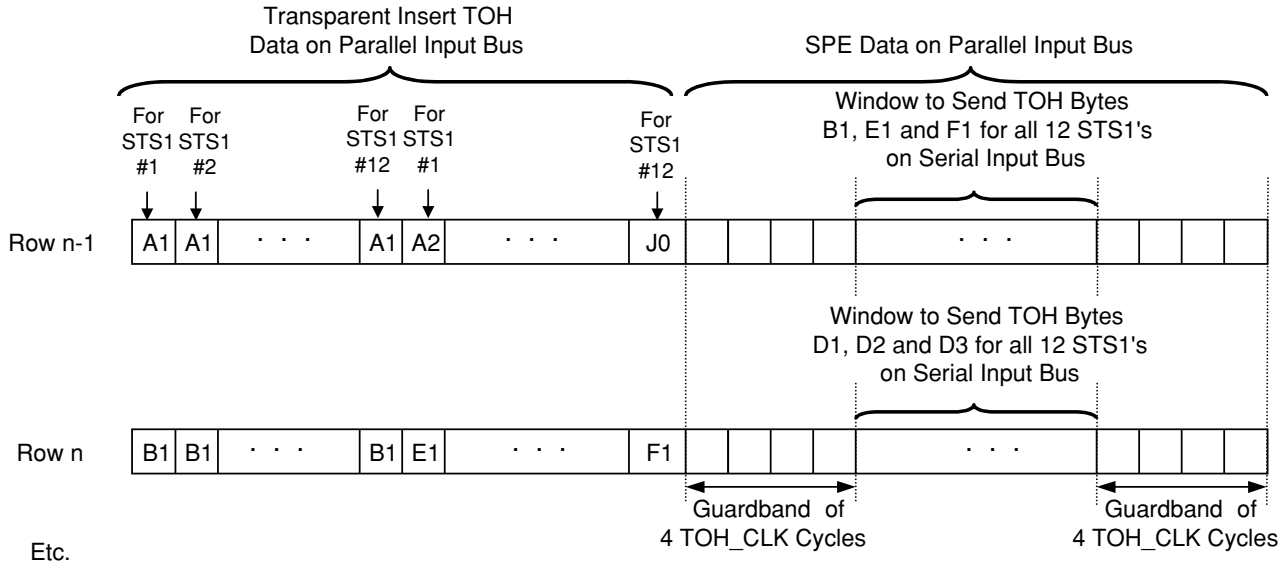
Figure 14. TOH Serial Port Input Framing Signals (FPGA to Core)



Incoming serial TOH data is synchronized initially to the free running clock, TOH_CLK. TOH_CLK can operate from a minimum frequency of 25 Mhz. to a maximum frequency of 106 MHz. TOH bytes are transferred in the order shown in Figure 15. Bytes are transferred over the serial links with the MSB first. Data should be transferred over the serial link on a row-by-row basis. With three TOH bytes/per row for each STS-1 stream and a total of 12 STS-1 streams per STS-12 frame, a total of 288 TOH bits must be transferred for each row. The 288 TOH bits per row can be sent back-to-back. In this case, TX_TO_CLK_EN will be high continuously for 288 TOH_CLK cycles.

It is the responsibility of the user to synchronize transfer of the TOH bytes to a pre-determined window of time relative to the STS-12 frame position on the parallel input bus, i.e., the 36 TOH bytes to be inserted in row number n must be transferred to the Core during the time the SPE bytes of row n-1 are being transferred to the Core over the parallel input bus. Within each SPE row, a guard band of four TOH_CLK cycles must be provided on each side of the TOH transfer window. No data may be transferred in these guard bands.

Figure 15. TOH Serial Port Input Framing Signals (FPGA to Core)



Although all TOH bytes from the 12 STS-1s are transferred into the device from each serial port, not all of them get inserted in the frame. There are three hard coded exceptions to the TOH byte insertion:

- Framing bytes (A1/A2 of all STS-1s) are not inserted from the serial input bus. Instead, they can always be regenerated.
- Parity byte (B1 of STS#1) is not inserted from the serial input bus. Instead, it is always recalculated (the 11 bytes following B1 are replaced with all zeros).
- Pointer bytes (H1/H2/H3 of all STS-1s) are not inserted from the serial input bus. Instead, they always flow transparently from parallel input to LVDS output.

Except for the above hardcode exceptions, the source of some TOH bytes can be controlled by bits in the control registers. The 12 STS-1 bytes forming a single STS-12 TOH header block are controlled as a whole. When configured to be in the transparent mode, the specific bytes must flow transparently from the parallel input. The 15 overhead bytes that can be controlled on a per STS-1 basis are the following:

- K1 and K2 bytes of the 12 STS-1s (24 bytes)
- S1 and M0 bytes of the 12 STS-1s (24 bytes)
- E1, F1, E2 bytes of the STS-1s (36 bytes)
- D1 through D12 bytes of the STS-1s (144 bytes)

The C1(J0) and B2 bytes (unshaded in the following table) are also passed through transparently from the parallel bus to the serial link.

Table 10 shows the order in which data is transferred to the serial LVDS output, starting with the most significant bit of the first A1 byte. The first bit of the first byte is replaced by an even parity check bit over all TOH bytes from the previous TOH frame. The source for the TOH bytes in the Serial TOH insert mode is summarized in the table.

Table 10. Transmitter TOH on LVDS Output (TOH Insert Mode)

A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	J0	J0	J0	J0	J0	J0	J0	J0	J0	J0	J0
B1	0	0	0	0	0	0	0	0	0	0	0	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	E1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1
D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D1	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D2	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3	D3
H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3
B2	B2	B2	B2	B2	B2	B2	B2	B2	B2	B2	B2	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K2	K2	K2	K2	K2	K2	K2	K2	K2	K2	K2
D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D4	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D5	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6	D6
D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D7	D8	D8	D8	D8	D8	D8	D8	D8	D8	D8	D8	D8	D9	D9	D9	D9	D9	D9	D9	D9	D9	D9	D9
D10	D10	D10	D10	D10	D10	D10	D10	D10	D10	D10	D10	D11	D11	D11	D11	D11	D11	D11	D11	D11	D11	D11	D11	D12	D12	D12	D12	D12	D12	D12	D12	D12	D12	D12
S1	S1	S1	S1	S1	S1	S1	S1	S1	S1	S1	S1	M0	M0	M0	M0	M0	M0	M0	M0	M0	M0	M0	M0	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2	E2

Regenerated bytes.

Bytes optionally inserted from TOH serial port data or transparently forwarded from parallel input port

Bytes transparently forwarded from parallel input port

Bytes transparently forwarded from parallel input port

Repeater

This block is essentially the inverse of the sampler block discussed in the receive path description section. It receives byte-wide STS-12 rate data from the TOH insert block. In order to support the STS-3 mode of operation, the HSI (622 Mbits/s) can be connected to a slower speed device (e.g., 155 Mbits/s). The purpose of this block is to rearrange the data being fed to the HSI so that each bit is transmitted four times, thus simulating 155 Mbits/s serial data. In STS-3 mode, the incoming STS-12 stream is composed of four identical STS-3s so only every fourth byte is used. The bit expansion process takes a single byte and stretches it to take up 4 bytes each consisting of 4 copies of the 8 bits from the original byte.

A1/A2 Processing

The A1 and A2 bytes provide a special framing pattern that indicates where a STS-1 begins in a bit stream. All 12 A1 bytes of each STS-12 are set to 0xF6, and all 12 A2 bytes are set to 0x28 automatically by the SONET framer. The latency from the transmit of the first bit of the A1 byte at the device output pins from the system frame pulse on the FPGA interface is between five to seven clock cycles of the reference clock (FPGA_SYSCLK).

The A1 and A2 bytes can also be intentionally corrupted for testing by the A1/A2 error insert control register (0x3000D, 0x3000E). Only the last A1 and first A2 are corrupted. When A1/A2 corruption detection is set for a particular channel, the A1/A2 values in the corrupted A1/A2 value registers are sent for the number of frames defined in the corrupted A1/A2 frame count register (0x3000C). When the corrupted A1/A2 frame count register is set to 0x00, A1/A2 corruption will continue until the A1/A2 error insert register is cleared.

The ORT8850 device only has one control register to set the A1/A2 bytes as well as the number of frames of corruption. To insert the corrupted A1/A2 each channel has an enable A1/A2 insert register. When the per channel error insert register bit is set, the A1/A2 values are corrupted for the number specified in the number of frames to corrupt. To insert errors again, the per channel error insert register bit must be cleared, and set again.

It is also possible to not insert the A1/A2 framing bytes using the per channel register bit “disable A1/A2 insert.”

B1 Processing

In the transmit direction a bit interleaved parity (BIP-8) error check set for even parity over all the bits of an STS-1 frame B1 is defined for the first STS-1 in an STS-12 only, the B1 calculation block computes a BIP-8 code, using even parity over all bits of the previous STS-12 frame after scrambling and is inserted in the B1 byte of the current STS-12 frame before scrambling.

Per-bit B1 corruption is controlled by the force BIP-8 corruption register (0x3000F). For any bit set in this register, the corresponding bit in the calculated BIP-8 is inverted before insertion into the B1 byte position. Each stream has an independent fault insert register that enables the inversion of the B1 bytes. B1 bytes in all other STS- 1s in the stream are filled with zeros.

It is also possible to not insert the B1 byte using the per channel register bit "disable B1 insert."

Scrambling
To ensure a 1's density for the SERDES the data stream is scrambled using a frame-synchronous scrambler with a sequence length of 127. The scrambling function can be disabled by setting a control register bit (0x3000C). The generating polynomial for the scrambler is $1 + x^6 + x^7$. This polynomial conforms to the standard SONET STS-12 data format. The scrambler is reset to 1111111 on the first byte of the SPE (byte following the Z0 byte in the twelfth STS-1). The scrambler runs continuously from that byte on throughout the remainder of the frame. The A1, A2, J0, and Z0 bytes are not scrambled.

After scrambling, the serial data is broadcast to both the work and protect LVDS buffers.

Receive Path Logic

Each of the two SONET logic blocks has four receiving channels which can be treated as one STS-48 stream or as four independent STS-12 or STS-3 channels.The received data streams are processed and passed to the FPGA logic.

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.

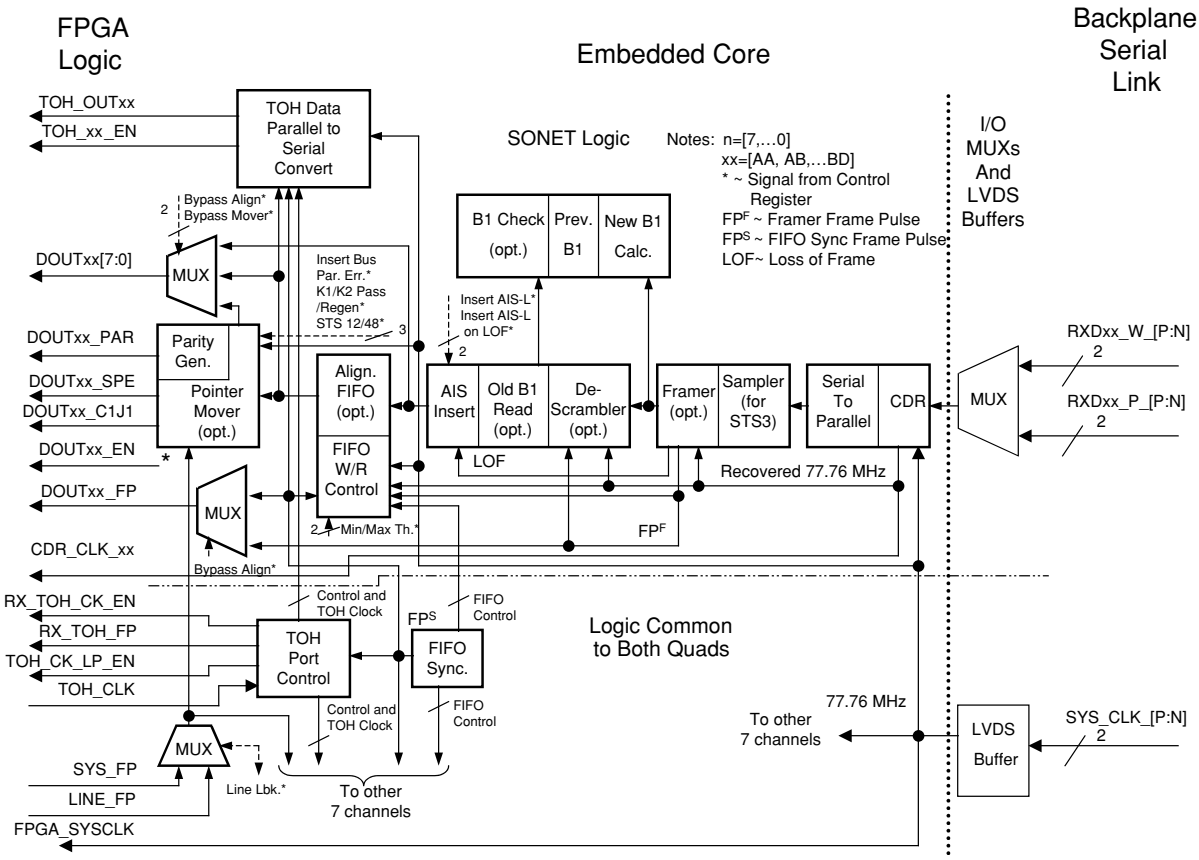
Each STM block receives the serial streams of STS-12 data (maximum of four streams per quad) from the LVDS inputs. There is no received clock input. There are two sets of receive LVDS pins, RXDxx_W_[P:N] and RXDxx_P_[P:N]. If the LVDS protection switching is used, the RXDxx_W_[P:N] LVDS inputs are used to accept the main data while the RXDxx_P_[P:N] LVDS inputs are used to accept the protect data.

Once the serial data is received from the LVDS inputs it goes through the following processing steps:

- The clock for the received data is recovered from the received serial data stream and a serial to parallel conversion is performed on the data
- The frame format of the incoming data is reconstructed (optional). The data is descrambled using the standard STS-12 polynomial (optional)
- A B1 parity error check is performed on the data from the previously transmitted frame
- The channel alignment FIFO perform channel alignments on groups of incoming data streams (optional) and/or simply perform the domain transfer from the recovered clock to the local reference clock.
- The SONET pointer interpreter and pointer mover detect the location of the SPE and C1J1 bytes and additionally inserts 0xFFs instead of received data into the data path, if an line Alarm Indication Signal (AIS-L) is detected. The offset pointers are adjusted to point to the location of the J1 byte.
- The received parallel data, parity, recovered clock, SPE and C1J1 indicators and TOH parallel data is sent to the FPGA logic.

These processing steps are described in more detail in the following sections. A block diagram of the receive path logic is shown in Figure 16. All processing except the CDR functions (clock recovery and serial to parallel conversion) is optional. If all processing except the SERDES is deselected, the device is said to be operating in the "bypass" mode.

Figure 16. Basic Logic Blocks, Receive Path, Single Channel



HSI Functions (Clock Recovery and Deserializer)

The HSI receive path functions include Clock and Data Recovery (CDR) and deserialization of the incoming data from the selected work or protect input stream to the byte-wide internal data bus format. The serial data received from the LVDS buffer does not have an accompanying clock. Based on data transitions, the receiver selects an appropriate internal clock phase for each channel to retime the data. The retimed data and clock are then passed to the DEMUX (deserializer) module. The DEMUX module performs serial-to-parallel conversion and provides parallel data and clock to the SONET framer block. For a 622 Mbits/s SONET stream, the HSI will perform Clock and Data Recovery (CDR) and MUX/DEMUX between 77.76 MHz byte-wide internal data buses and 622 Mbits/s serial LVDS links.

Sampler

This block operates on the byte-wide data directly from the HSI macro. The HSI external interface always runs at 622 Mbits/s (STS-12), or 850 Mbits/s, but it can be connected directly to a 155 Mbits/s STS-3 stream. If connected to a 155 Mbits/s stream, each incoming bit is received four times. This block is used to return the byte stream to the expected STS-12 format. The mode of operation is controlled by a register and can either be STS-12 (pass-through) or STS-3. The output from this block is not bit aligned (i.e., an 8-bit sample does not necessarily contain an entire SONET byte), but it is in standard SONET STS-12 format (i.e., four STS-3s) and is suitable for framing.

SONET Framer Block

The framer block takes byte-wide data from the HSI, and outputs a byte-aligned, byte-wide data stream and 8 kHz sync pulse. The framer algorithm determines the out-of-frame/in-frame status of the incoming data and will set alarm register bits on both an errored frame and an Out-Of-Frame (OOF) state.

The framer block takes byte wide data from the HSI, and outputs a byte aligned byte wide stream and 8 kHz sync pulse asserted coincident the first A1 byte which will be used by following blocks. (Note however that if the pointer

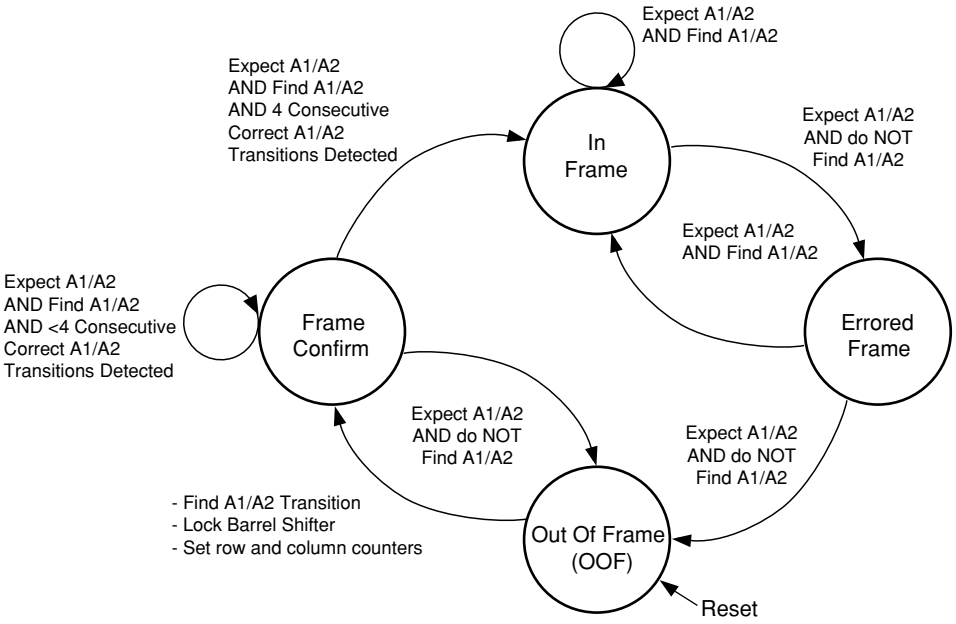
mover is active, there is no fixed timing relationship between the data sent to the FPSC and DOUTxx_FP and the DOUTxx_SPE and DOUTxx_C1. J1 signals should be used instead to determine data alignment within the frame.) The framer algorithm determines the out-of-frame/in-frame status of the incoming data and will cause alarms on both an errored frame and an OOF (out-of-frame) state. Functions performed by this block include:

- A1-A2 framing pattern detection. (Framing similar to SONET specification)
- Generation of timing and an 8kHz frame pulse.
- Detections of Out Of Frame (OOF) (generates an alarm).
- Errored frame detection (increments error counter).

Framer State Machine

Figure 17 shows the state machine for the framer. Because the ORT8850 is primarily intended for use between itself and another ORT8850 or other devices via a backplane, there is only one errored frame state. Thus there is no Severely Errored Frame (SEF) or Loss-Of-Frame (LOF) indication.

Figure 17. Framer State Machine



Notes:
1) Row and column counters are only set/reset by a state transition from the OOF state to to the Frame Confirm state.
2) "Expect A1/A2" means that the row and column counters have counted to the place for the last (12 th.) A1 byte and that the next byte should be an A2 byte.

OOF State

This is the initial state for the state machine after a reset. In this state the A1 pattern is searched for on every clock cycle. A second stage of comparison is implemented to locate the A1/A2 transition. When the A1/A2 transition is found, the following occurs:

- The state machine moves from the OOF state to the Frame Confirm State.
- The A1offset for the byte start location is locked.
- Row and column counters are set

Frame Confirm State

In this state the A1/A2 transition is only compared for at the appropriate location, i.e. beginning at the 12th A1 location. This location is determined from the row and column counters which were set at the transition from OOF to Frame Confirm. If at this time the comparison fails, the state machine reverts to the OOF state. If the comparison

passes, the next state will either still be Frame Confirm or will be In Frame. For the framer to declare an In Frame state the framer must detect 4 consecutive correct A1/A2 framing patterns.

This state is similar to the Frame Confirm state except that if the comparison at the A1/A2 time is incorrect, the next state will be the Errored Frame state. If the comparison is correct, the next state will be In Frame. Data is only valid in the Frame state

Errored Frame State
Once the Errored Frame state has been reached, if the next comparison is incorrect, the next state will be OOF i.e., after two transitions are missed, the state machine goes into the OOF state which will also generate an alarm. Otherwise, if the comparison correct, the next state will be In Frame. Also, when the framer detects an errored frame it increments an A1/A2 frame error counter register accessible from the system bus. The counter can be monitored by a processor to compile performance status on the quality of the backplane.

B1 Parity Error Check
The B1 parity error check block receives byte-wide scrambled byte-wide parallel data and a frame sync from the framer. The B1 error check calculation block computes a BIP-8 (bit interleaved parity 8 bits) code, using even parity over all bits of the current STS-12 frame before descrambling.

The same calculation had previous been done for the previous STS-12 frame. The value obtained then is checked against the B1 byte of the current frame after descrambling. A per-stream B1 error counter is incremented for each bit that is in error. The error counter register is accessible from the system bus.

Descrambler
The received streams from the framer are descrambled using a frame synchronous descrambler with the same polynomial ($1 + x^6 + x^7$) that was used in the transmit path. If the incoming data is not scrambled, the descrambling function can be disabled by setting a control register bit (0x3000C). The A1/A2 framing bytes, the section trace byte (C1/J0) and the growth bytes (Z0) are not descrambled.

AIS-L Insertion
The Alarm Indication Signal (AIS) is a continuous stream of unframed 1s sent to alert downstream equipment that the near-end terminal has failed, lost its signal source, or has been temporarily taken out of service. AIS-L is inserted into the received frame by writing all ones for all bytes of the descrambled stream under two conditions:

1. If a force AIS_L state is enabled by a bit in the AIS-L force register, AIS-L is inserted into the received frame continuously. This will cause all bytes within a STS-12 frame to be FF

2. If an AIS-L Insertion on Out-Of-Frame enabled via a register, AIS-L is inserted into the received frame when the framer indicates that an out-of-frame condition exists.

Since this occurs after the overhead processing block, all Transport Overhead can continue to byte read and B1 can still be used to monitor link integrity.

Alignment FIFO and Multi-Channel Alignment
The alignment FIFO in the ORT8850 performs two functions, clock domain transfer and multi-channel alignment. The depth of the alignment FIFO is 10 bit words which allows it to absorb channel timing differences of up to 18 clock cycles. Multi-channel alignment is based on the incoming A1/A2 bytes.

The alignment FIFO is always written from the SONET framer using the per channel recovered clock. The FIFO is always read using the local reference clock (FPGA_SYSCLK). For this reason when doing multi-channel alignment there must be 0 ppm between the transmit ORT8850 reference clock and the receiving ORT8850 reference clock. This can only be accomplished by using a single clock source for both the transmitting and receiving devices.

The alignment FIFO has several alarm and control indicators that are accessible via control and alarm registers available via the system bus or the MPI. The default alignment threshold values for the alignment FIFO are set in registers at 0x3000A and 0x3000B. Here the min and max threshold values can be programmed. The default min is set to 2 clocks and the max default is set to 15. If the alignment FIFO determines that these thresholds have been

violated a per channel alarm bit will be set indicating that this channel has exceeded the threshold, as well as a FIFO out-of-sync alarm bit to indicate the channel is not longer in sync with the reset of the alignment group.

The incoming data can be considered as 4 STS-12 channels (A, B, C, and D) per quad. Thus we have STS-12 channels AA to AD from quad A of the STM and STS-12 channels BA to BD of quad B. The 8 channels of parallel SONET data can be grouped into an alignment group by 2, by 4 or all 8 channels. As the serial data is run through the backplane and SERDES the parallel data can be slightly varied. The alignment FIFO can absorb this difference in the channels and create a byte aligned grouping.

These streams can be frame aligned in the following patterns. Streams can be aligned on a twin STS-12 basis as shown in Figure 18. In STS-48 mode, all four STS-12s of each STM quad are aligned with each other (i.e. AA, AB, AC, AD) as shown in Figure 19. Optionally in STS-48 mode all eight STS-12s (STMs A and B) can be aligned which allows hitless switching since all streams will be byte aligned (Figure 20). Multiple ORT8850 devices can be aligned with each other using a common system frame pulse to enable STS-192 or higher modes.

Figure 18. Twin Channel Alignment

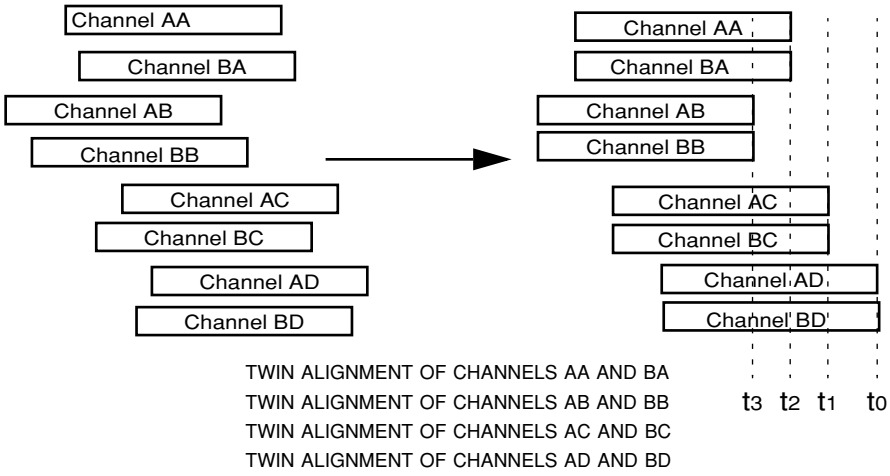


Figure 19. Alignment of SERDES Quads A and B

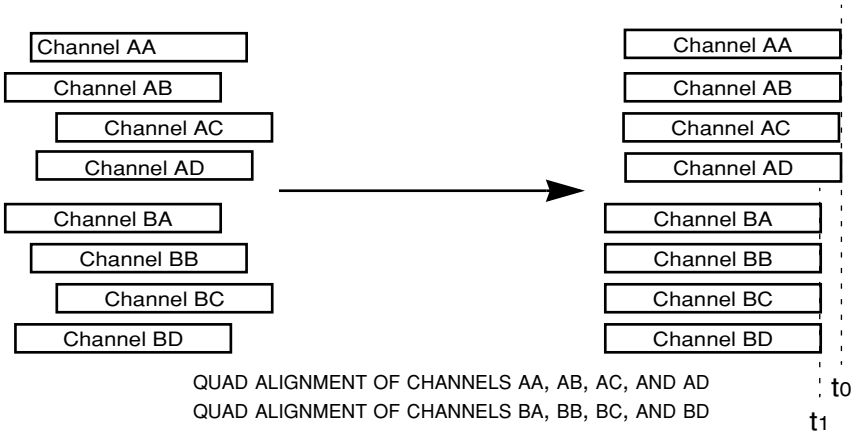
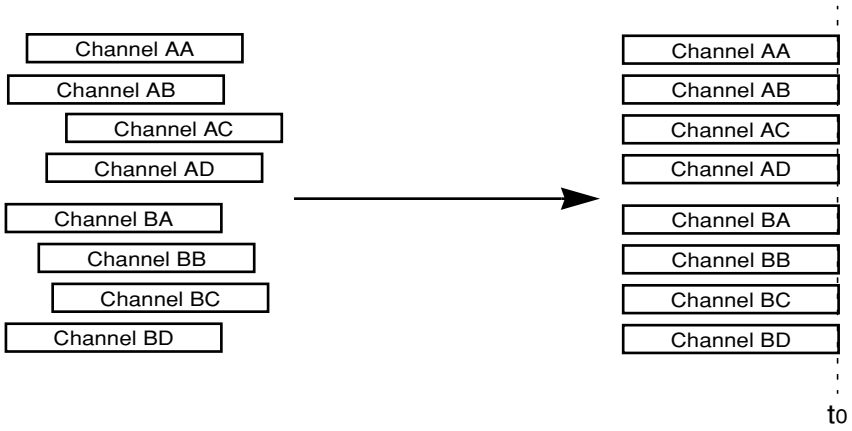


Figure 20. Alignment of all Eight SERDES Channels.



There is a provision to allow certain streams to be disabled (i.e. not producing alarms or affecting synchronization). These streams can be enabled at a later time without disrupting other streams. If the newly enabled stream needs to be a part of a bigger group the entire group must be resynchronized unless the affected stream was active when the initial synchronization was performed. As long as all streams to be aligned were active when the most recent synchronization was performed, individual streams may be enabled or disabled without affecting synchronization.

It is recommended that users select the smallest possible groups for channel alignment. If an application only requires that two channels be aligned then it is best to use by-2 grouping. All of the channels in a group will affect the group's total alignment. If a channel in a group fails or is shut down it will not affect any of the other channels in the group. This channel will simply be removed from the alignment algorithm. When the channel is re-enabled into a working group it will be out of alignment with the rest of the group. It will be necessary to perform a FIFO realignment procedure to realign the group. During a FIFO realignment data will not pass through any of the channels in the alignment group.

Alignment FIFO Algorithm

The algorithm controlling writes to the alignment FIFO and reads from it operates as follows: Prior to detecting the first frame pulse for a link being aligned, each link in the group continually writes to address 0 within its own FIFO (each link has a FIFO). When the first link in the group receives a frame pulse from Framer block the write pointer for the corresponding FIFO increments to next write address on each clock cycle. L inks that have not received a frame pulse continue to write into their respective FIFOs. When any link receives a frame pulse, the write address for that FIFO will be reset to '0'

The operation of the alignment algorithm requires a wait of several clocks from the first arriving frame pulse before reading of FIFO data begins. In this case, when all frame pulses arrive together the alignment algorithm initiates reads after 9 clocks cycles. If, however, the first to last arriving frame pulses are separated by multiple clock cycles, there will be additional clock cycles between the first frame pulse and the first read. If all links in the group have not reported a valid frame pulse signal after 18 clock cycles, an out of sync state is entered and an alarm is generated.

After all links have received frame pulses and are incrementing their write addresses while writing into their FIFOs, data is then read out of each link's FIFO one byte at a time. All aligned links are now Frame/byte/bit synchronous.

FIFO Alignment Procedure

The FIFO alignment block has the ability to be realigned by changing the value of bits in the alignment control registers. This may be done in the FPGA logic or under the control of an external device through the system bus or MPI. Alignment must take place after the stream has settled with valid data to guarantee proper channel alignment and uncorrupted data transmission.

Channel realignment must occur when a channel goes from the Out-Of-Frame (OOF) state to the In-Frame state. This happens when the channels are first powered up and given a valid frame pulse. This is the obvious known

condition. It is also possible during operation for the channel to go into OOF. This may occur due to the removal of either the frame pulse or the cable. If this is the case, AND is is part of a multi-channel alignment group, the realignment procedure must be re-executed once the channel goes back into frame.

When a channel goes from the OOF state to the In-Frame state the OOF alarm bit is set per channel. The OOF alarm bit is a per channel bit contained in the channel alarm register. It takes the receiver at least 4 full SONET frames for the state machine to declare the In-Frame state. When the OOF bit is high the channel is in OOF. When the OOF bit changes to a '0' then the channel is back in frame and the realignment procedure should be executed.

Table 11 lists the register values to set up the ORT8850 for alignment FIFO sync realignment. The order is specific. The values are given from the PowerPC point of view. If using the MPI to write data to the ORT8850, the value given in the table is the value that should be used. If using the UMI of the system bus, the data value would need to be byte flipped. The following setup procedures should be followed after the enabled channels have a valid frame pulse, and are in the Frame state:

Table 11. Alignment FIFO Synch Realignment

Register Address	Value (Binary)	Description
0x30020, bit 6	1	Force AIS-L in all channels of the group to be synchronized.
0x30038, bit 6	1	
0x30050, bit 6	1	
0x30068, bit 6	1	
0x30080, bit 6	1	
0x30098, bit 6	1	
0x300B0, bit 6	1	
0x300C8, bit 6	1	
Wait for 4 SONET Frames (~500µs)		
0x30017, specific bits	1	Issue FIFO realignment commands.
0x30018, specific bits	1	
Wait for Another 4 SONET Frames (~500µs)		
0x30017, specific bits	0	Clear FIFO alignment command register bits written in previous steps.
0x30018, specific bits	0	
0x30020, bit 6	0	Release AIS-L in all channels of the group to allow normal data flow through the reveiver.
0x30038, bit 6	0	
0x30050, bit 6	0	
0x30068, bit 6	0	
0x30080, bit 6	0	
0x30098, bit 6	0	
0x300B0, bit 6	0	
0x300C8, bit 6	0	

RX Serial TOH Processing

Transport overhead is extracted from the receive data stream by the TOH extract block. The incoming data gets loaded into a 36-byte shift register on the system clock domain. This, in turn, is clocked onto the TOH clock domain at the start of the SPE time, where it can be clocked out.

The TOH processor is responsible for serializing all received TOH bytes of each channel through that channel's corresponding serial TOH data port. The TOH serial ports are synchronized to the TOH clock (the same clock that is being used by the serial ports on the transmitter side). This free-running TOH clock is provided to the core by external circuitry and operates at a minimum frequency of 25 MHz and a maximum frequency of 77.76 MHz. Data is transferred over serial links in a bursty fashion as controlled by the RX TOH clock enable signal, and is common

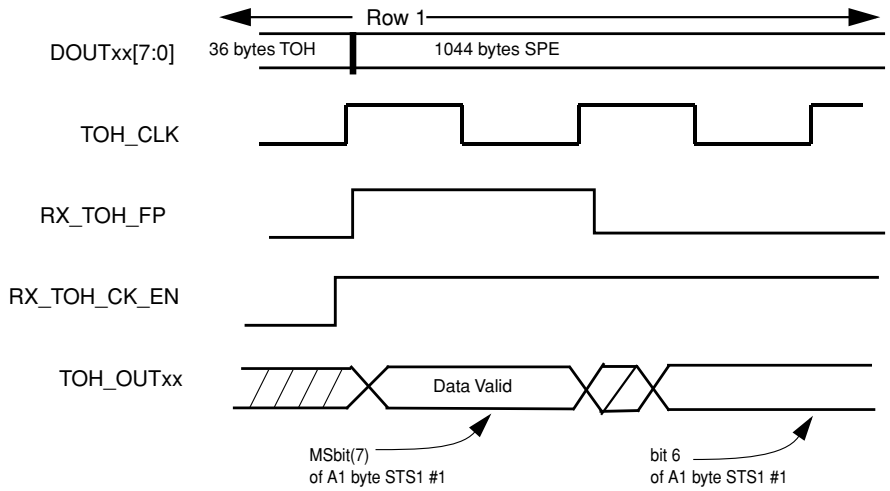
to all eight channels. All TOH bytes from the STS-12 streams are transferred over the appropriate serial link in the same order in which they appear in a standard STS-12 frame.

During the SPE time, the receiver TOH frame pulse is generated (RX_TOH_FP) which indicates the start of the row of 36 TOH bytes. This pulse, along with the receive TOH clock enable (RX_TOH_CK_EN), as well as the TOH data, are all launched on the rising edge of the TOH clock (TOH_CLK).

On the TOH serial port, all TOH bytes are sent as received on the LVDS input (MSB first). The only exception is the most significant bit of byte A1 of STS#1, which is replaced with an even parity bit. This parity bit is calculated over the previous TOH frame. Also, on AIS-L (either resulting from OOF or forced through software), all TOH bits are forced to all ones with proper parity (parity automatically ends up being set to 1 on AIS-L).

The Core logic must provide framing information to the FPGA using the RX_TOH_CK_EN and RX_TOH_FP output signals. TOH data is output on a row by row basis, with the one clock cycle frame pulse (RX_TOH_FP) delineating the start of a row, as shown in Figure 21. Detailed timing for the TOH serial output is shown in later in Figure 29.

Figure 21. TOH Serial Port Output Framing Signals (Core to FPGA)



Receiver TOH reconstruction on the output parallel bus is performed as shown in the following table (if the pointer mover is not bypassed).

Table 12. Receiver TOH Byte Reconstruction (Output Parallel Bus)

A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	A2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3	H3
0	0	0	0	0	0	0	0	0	0	0	0	K2	0	0	0	0	0	0	0	0	0	0	0	K2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Regenerated bytes.

Regenerated bytes (under pointer control. SS bits must be transparent, AIS-P must be supported)

Bytes taken from elastic store buffer on negative stuff opportunity - else forced to all zeroes

Transparent or all zeros (K1/K12 are either taken from K1/K12 buffer or forced to all zero-soft control) In transport mode, AIS-L must be supported.

All zero bytes

Special TOH Byte Functions

The K1 and K2 bytes are used in Automatic Protection Switch (APS) applications. K1 and K2 bytes can be optionally passed through the pointer mover under software control, or can be set to zero with the other TOH bytes.

Pointer Interpreter and Pointer Mover

After the alignment FIFO the receive data can optionally go through the pointer interpreter and pointer mover. The pointer interpreter will identify the SONET payload envelope (SPE), the C1 bytes and the J1 bytes, and provide this information to the FPGA logic. For data applications where the user is simply using SONET to carry user defined cells in the payload the SPE signal is very useful as an enable to the cell processor. C1J1 for generic data applications can be ignored. If the pointer interpreter and pointer mover are bypassed, then the SPE and C1J1 signals will be always '0'.

Since the start of an SPE can be located at any point in a SONET frame, the starting point is identified using pointer bytes H1 and H2. The pointer bytes indicate the offset of the start of the SPE from the pointer byte position. Two payload pointer bytes (H1 and H2) are allocated to a pointer that indicates the offset in bytes between the pointer and the first byte of the STS SPE. The pointer bytes are used in all STS-1s within an STS-N to align the STS-1 Transport Overhead in the STS-N, and to perform frequency justification. These bytes are also used to indicate concatenation, and to detect Alarm Indication Signals (AIS).

The resulting 2 byte pointer is divided into three parts:

1. Four bits of New Data Flag (NDF)
2. Two bits of unassigned bits (These bits are set to 00.)
3. Ten bits for pointer value, which are alternately considered increment (I) bits or decrement (D) bits. The 10 bit pointer is required to represent the maximum SPE offset of 782 (9 rows * 87 columns - 1). Specific combinations of pointer byte values indicate that positive or negative frequency justification will occur and also whether or not the current frame is a concatenated frame.

Normally the NDF bits are set to 0110, which indicates that the current pointer values are unchanged. The inverse bit pattern, 1001, indicates that some data has changed. Any other bit configuration is interpreted using the 3 of 4 rule, i.e., 1110 is interpreted as 0110, etc. Patterns that cannot be resolved are undefined, however all one's in the NDF and in the pointer bits indicates AIS detection.The ORT8850 can correctly process any length of concatenation of STS frames (multiple of three) as long as it begins on an STS-3 boundary (i.e., STS-1 number one, four, seven, ten, etc.) and is contained within the smaller of STS-3, 12, or 48.

Table 13. Valid Starting Positions for and STS MC

STS-1 Number	STS-3cSPE	STS-6cSPE	STS-9cSPE	STS-12cSPE	STS-15cSPE	STS-18c to STS-48c SPEs
1	Yes	Yes	Yes	Yes	Yes	Yes
4	Yes	Yes	Yes	No	Yes	—
7	Yes	Yes	No	No	Yes	—
10	Yes	No	No	No	Yes	—
13	Yes	Yes	Yes	Yes	Yes	—
16	Yes	Yes	Yes	No	Yes	—
19	Yes	Yes	No	No	Yes	—
22	Yes	No	No	No	Yes	—
25	Yes	Yes	Yes	Yes	Yes	—
28	Yes	Yes	Yes	No	Yes	—
31	Yes	Yes	No	No	Yes	—
34	Yes	No	No	No	Yes	No
37	Yes	Yes	Yes	Yes	No	No
40	Yes	Yes	Yes	No	No	No
43	Yes	Yes	No	No	No	No

Table 13. Valid Starting Positions for and STS MC (Continued)

STS-1 Number	STS-3cSPE	STS-6cSPE	STS-9cSPE	STS-12cSPE	STS-15cSPE	STS-18c to STS-48c SPEs
46	Yes	No	No	No	No	No

Note:
Yes = STS-Mc SPE can start in that STS-1.
No = STS-Mc SPE cannot start in that STS-1.
- = Yes or no, depending on the particular value of M.

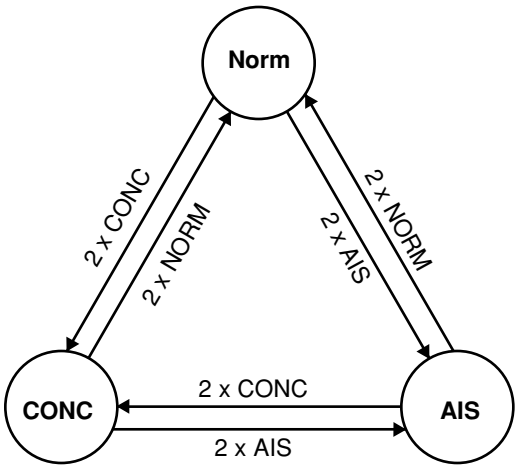
A pointer action byte (H3) is allocated for SPE frequency justification purposes. Frequency justification is discussed in a later section. The H3 byte is used in all STS-1s within an STS-N to carry the extra SPE byte in the event of a negative pointer adjustment. The value contained in this byte when it's not used to carry the SPE byte is undefined.

Pointer Interpreter State Machine.
The pointer interpreter's highest priority is to maintain accurate data flow (i.e., valid SPE only) into the elastic store. This will ensure that any errors in the pointer value will be corrected by a standard, fully SONET compliant, pointer interpreter without any data hits. This means that error checking for increment, decrement, and new data flag (NDF) (i.e., 8 of 10) is maintained in order to ensure accurate data flow. A single valid pointer (i.e., 0-782) that differs from the current pointer will be ignored. Two consecutive incoming valid pointers that differ from the current pointer will cause a reset of the J1 location to the latest pointer value (the generator will then produce an NDF). This block is designed to handle single bit errors without affecting data flow or changing state.

The pointer interpreter has only three states (NORM, AIS, and CONC). NORM state will begin whenever two consecutive NORM pointers are received. If two consecutive NORM pointers that both differ from the current offset are received, then the current offset will be reset to the last received NORM pointer. When the pointer interpreter changes its offset, it causes the pointer generator to receive a J1 value in a new position. When the pointer generator gets an unexpected J1, it resets its offset value to the new location and declares an NDF. The interpreter is only looking for two consecutive pointers that are different from the current value. These two consecutive NORM pointers do not have to have the same value. For example, if the current pointer is ten and a NORM pointer with offset of 15 and a second NORM pointer with offset of 25 are received, then the interpreter will change the current pointer to 25.

If the data is concatenated, the receipt of two consecutive CONC pointers causes CONC state to be entered. Once in this state, offset values from the head of the concatenation chain are used to determine the location of the STS SPE for each STS in the chain. Finally, if two consecutive AIS pointers cause the AIS state to occur. Any two consecutive normal or concatenation pointers will end this AIS state. This state will cause the data leaving the pointer generator to be overwritten with 0xFF.

Figure 22. Pointer Mover State Machine



SPE and C1J1 Identification

In the ORT8850 each frame can be considered as 12 STS-1s. In the SPE region, there are 12 J1 pulses for each STS-1s. There is one C1(J0, new SONET specifications use J0 instead of C1 as section trace to identify each STS-1 in an STS-N) pulse in the TOH area for one frame. Thus, for non concatenated data there are a total of 12 J1 pulses and one C1(J0) pulse per frame. The C1(J0) pulse is coincident with the J0 of STS-1 #1.

The pointer interpreter identifies the payload area of each frame. The SPE flag is active when the data stream is in SPE area. SPE behavior is dependent on pointer movement and concatenation. Note that in the TOH area, H3 can also carry valid data. When valid SPE data is carried in this H3 slot, SPE is high in this particular TOH time slot. In the SPE region, if there is no valid data during any SPE column, the SPE signal will be set to low. SPE allows a pointer processor to extract payload without interpreting the pointers.

Figure 23. SPE and C1J1 Functionality for STS -12

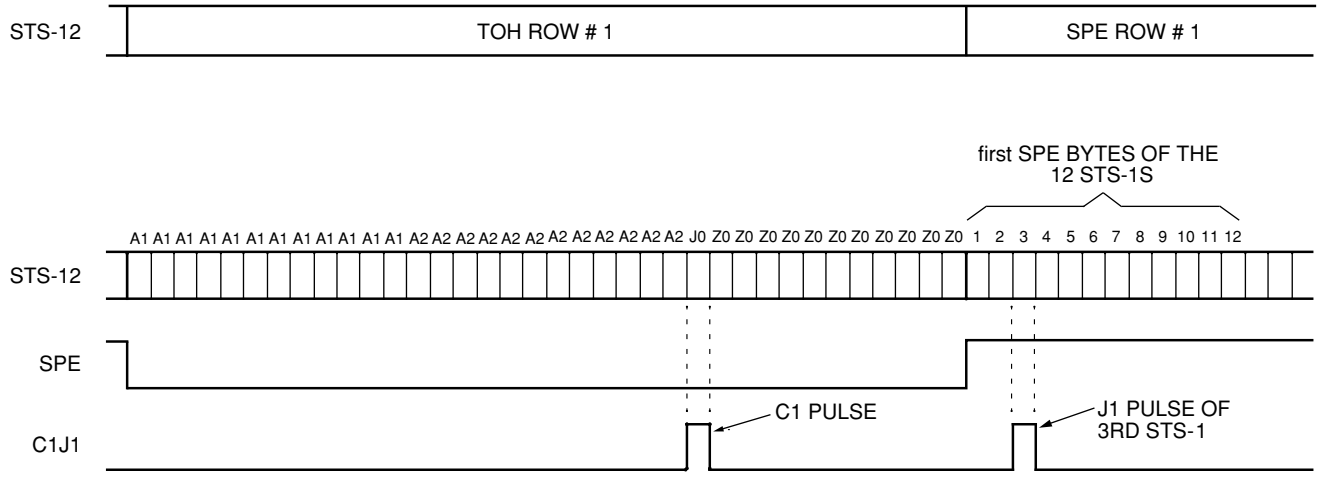
SPE	C1J1	Description
0	0	TOH information excluding C1(J0) of STS-1 #1.
0	1	Position of C1(J0) of STS-1 #1 (one per frame). Typically used to provide a unique link identification (256 possible unique links) to help ensure cards are connected into the backplane correctly or cables are connected correctly.
1	0	SPE information excluding the 12 J1 bytes.
1	1	Position of the 12 J1 bytes.

The following rules are observed for generating SPE and C1J1 signals:

- On occurrence of AIS-P on any of the STS-1, there is no corresponding J1 pulse.
- In case of concatenated payloads (up to STS48c), only the head STS-1 of the group has an associated J1 pulse.
- The C1J1 signal tracks any pointer movements.

This behavior is illustrated in the following figure. Note that the actual bit positions are dependent on the actual payload configuration and offset.

Figure 24. SPE and C1J1 Signals



Pointer Mover

After the pointer interpreter comes the pointer mover block. There is a separate pointer mover for the two SONET framer quads, A and B, each of which handles up to one STS-48 (four channels) The K1/K2 bytes and H1-SS bits are also passed through to the pointer generator so that the FPGA can receive them. The pointer mover handles both concatenations inside the STS-12, and to other STS-12s inside the core. Use of this block is optional, as discussed in a later section.

The pointer mover block can correctly process any length of concatenation of STS frames (multiple of three) as long as it begins on an STS-3 boundary (i.e., STS-1 number one, four, seven, ten, etc.) and is contained within the smaller of STS-3, 12, or 48. The pointer value can be adjusted to change the start of the SPE position and thereby adjust for frequency changes. The variations in frequency are taken care of using justification.

When the incoming data rate exceeds the nominal rate, negative justification is used to compensate for the frequency difference. Data are transmitted in the pointer action byte, H3, and thereby adjust for the additional incoming data. The size of one SPE is still the same, but it is transmitted in less time than usual, and so a higher data rate is achieved for some time. Negative justification causes the start of SPE to move left by one column or by decreasing the pointer by one. The following frames contain the new pointer value.

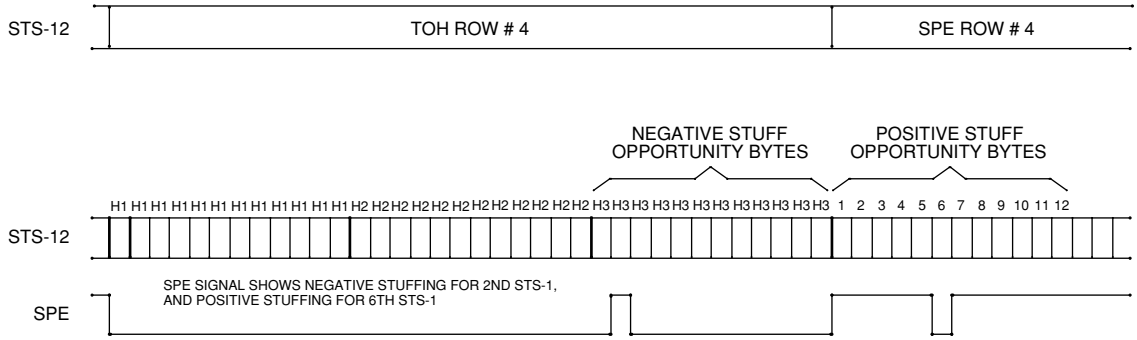
Negative justification is indicated by inverting the D bits (bits 8, 10, 12, 14, 16) of the pointer word. The receiver determines the occurrence of negative justification by examining these bits of the pointer word and applying a 5-bit majority logic on them.

When the incoming data rate lags the nominal rate, positive justification is used to compensate for the frequency difference. Data are not transmitted in the byte following the pointer action byte, H3, and thereby adjust for the lack of incoming data. The size of one SPE is still the same, but it is transmitted in more time than usual, and hence a slower data rate is achieved for some time. Positive justification causes the start of SPE to move right by one column or by increasing the pointer by one. The following frames contain the new pointer value.

Positive justification is indicated by inverting the I bits (bits 7, 9, 11, 13, 15) of the pointer word. The receiver determines the occurrence of positive justification by examining these bits of the pointer word and applying a 5-bit majority logic on them.

The SPE signal to the FPGA logic must be high during negative stuff opportunity byte time slots (H3) for which valid data is carried (negative stuffing). SPE signal must be low during positive stuff opportunity byte time slots for which there is no valid data (positive stuffing). This behavior is shown in the following figure.

Figure 25. SPE Signal During Justification



In either justification, the pointer must remain unchanged for at least three consecutive frames before it can be justified again. The pointer can jump randomly to a new position at any point of time. This can happen in conditions when the transmitting end has just recovered from an error condition. A sudden jump in the pointer value is indicated through NDF, New Data Flag. This information is carried in the four MS bits of the pointer word. A 3-bit majority logic is applied on the NDF bits to determine the status of the pointer jump.

Pointer Generator

The pointer generator maps the corresponding bytes into their appropriate location in the outgoing byte stream. The generator also creates offset pointers based on the location of the J1 byte as indicated by the pointer interpreter. The generator will signal NDFs when the interpreter signals that it is coming out of AIS state. The pointer generator resets the pointer value and generates NDF every time a byte marked J1 is read from the elastic store that doesn't match the previous offset. Increment and decrement signals from the pointer interpreter are latched once per frame on either the F1 or E2 byte times (depending on collisions); this ensures constant values during the H1 through H3 times. The choice of which byte time to do the latching on is made once when the relative frame

phases (i.e., received and system) are determined. This latch point is then stable unless the relative framing changes and the received H byte times collide with the system F1 or E2 times, in which case the latch point would be switched to the collision-free byte time.

There is no restriction on how many or how often increments and decrements are processed. Any received increment or decrement is immediately passed to the generator for implementation regardless of when the last pointer adjustment was made. The responsibility for meeting the SONET criteria for maximum frequency of pointer adjustments is left to an upstream pointer processor.

Receive Bypass Options

Not all of the blocks in the receive direction are required to be used. The following bypass options are valid in the receive (backplane → FPGA) direction:

- STM Pointer Mover bypass:
  - In this mode, data from the alignment FIFOs is transferred to the FPGA logic. All channels are synchronous to the FPGA_SYSCLK signals driven to the FPGA logic, as is also the case when the pointer mover is not bypassed. During bypass SPE, C1J1, and data parity signals are not valid. When the pointer mover is bypassed, eight frame pulses (DOUTxx_FP) from aligned channels are provided by the embedded core to the FPGA.
  - When the pointer mover is used, the FPGA logic provides the frame pulse on the LINE_FP (recall: there is only one LINE_FP just like there is only one SYS_FP) signal essential for the Pointer Mover to move the data. The FPGA gets eight channels of SONET data with the A1 byte position of each channel of the TOH arbitrarily offset from the LINE_FP. The DOUTxx_FP signals are not valid when the pointer mover is used.
- STM Pointer Mover and Alignment FIFO bypass:
  - In this mode, data from the framer block is transferred to the FPGA logic. All channels supply data and frame pulses synchronous with their individual recovered clock (CDR_CLK_xx) per channel. During bypass, SPE, C1J1, and data parity signals are not valid. Additionally, no serial TOH_OUT_xx data and frame pulse signals will be available. The DOUTxx_FP signals are aligned with the A1 byte position of each channel, as shown in Figure 26.

Figure 26. Pointer Mover and Alignment FIFO Bypass Timing

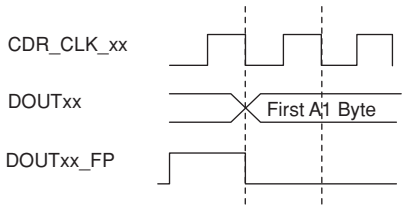


Table 14 shows the register settings to enable the bypass modes.

Table 14. Register Settings for Bypass Mode

Register Address	Value	Description
0x3000C	0x04	Turn off the SONET scrambler/descrambler
0x30020	0x07	Channel AA in functional mode
0x30038	0x07	Channel AB in functional mode
0x30050	0x07	Channel AC in functional mode
0x30068	0x07	Channel AD in functional mode
0x30080	0x07	Channel BA in functional mode
0x30098	0x07	Channel BB in functional mode
0x300B0	0x07	Channel BC in functional mode
0x300C8	0x07	Channel BD in functional mode

Table 14. Register Settings for Bypass Mode (Continued)

Register Address	Value	Description
0x30021	0x01	Channel AA in transparent mode
0x30039	0x01	Channel AB in transparent mode
0x30051	0x01	Channel AC in transparent mode
0x30069	0x01	Channel AD in transparent mode
0x30081	0x01	Channel BA in transparent mode
0x30099	0x01	Channel BB in transparent mode
0x300B1	0x01	Channel BC in transparent mode
0x300C8	0x01	Channel BD in transparent mode
0x30023	0x30	Channel AA - Do not insert A1/A2 or B1
0x3003B	0x30	Channel AB - Do not insert A1/A2 or B1
0x30053	0x30	Channel AC - Do not insert A1/A2 or B1
0x3006B	0x30	Channel AD - Do not insert A1/A2 or B1
0x30083	0x30	Channel BA - Do not insert A1/A2 or B1
0x3009B	0x30	Channel BB - Do not insert A1/A2 or B1
0x300B3	0x30	Channel BC - Do not insert A1/A2 or B1
0x300CB	0x30	Channel BD - Do not insert A1/A2 or B1
0x30037	0x44	Channel AA - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x3004F	0x44	Channel AB - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x30067	0x44	Channel AC - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x3007F	0x44	Channel AD - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x30097	0x44	Channel BA - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x300AF	0x44	Channel BB - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x300C7	0x44	Channel BC - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer
0x300DF	0x44	Channel BD - Bypass Alignment FIFO and Pointer interpreter/mover, disable SONET framer

Note: To select between full, half and quad rate modes, registers 0x300E1 and 0x300E2 are used. See the memory map for details on these registers.

FPGA/Embedded Core Interface Signals

Table 15. FPGA/Embedded Core Interface Signals

ORT8850 FPGA/Embedded Core Interface Signals - SONET Blocks		
FPGA/Embedded Core Interface Signal Name xx=[AA,…,BD]	**Input (I) to or Output (O) from Core**	**Signal Description**
Common Interface Signals		
FPGA_SYSCLK	O	Local reference clock from the core to the FPGA. All of the transmit data is captured on this clock edge inside the ORT8850 core. If using the alignment FIFO all of the parallel data from the ort8850 core will also be clocked from this clock. This signal uses an ORCA Series4 primary clock route.

Table 15. FPGA/Embedded Core Interface Signals (Continued)

ORT8850 FPGA/Embedded Core Interface Signals - SONET Blocks		
FPGA/Embedded Core Interface Signal Name xx=[AA,…,BD]	**Input (I) to or Output (O) from Core**	**Signal Description**
SYS_FP	I	System frame pulse generated inside the FPGA logic. This is a single clock pulse of FPGA_SYSCLK every 9720 clock cycles. For a 77.76 MHz reference clock the system frame pulse is at the SONET standard of 8 KHz. All the eight Transmit channels' first A1 byte should be aligned to the SYS_FP. Internally SYS_FP is used when Far end loopback (line side loopback) needs to be performed. This loopback can only be performed when Pointer Mover is not bypassed.
LINE_FP	I	User-provided frame pulse used by only the Pointer Mover block in the receive direction. The Pointer Mover moves the data to align it with the LINE_FP. If the Pointer Mover is bypassed, LINE_FP is not used.
TOH_CLK	I	Clock driven from the FPGA to clock the TOH processor. This clock can be in the range from 25MHz to 77.76MHz. If not using the TOH communication channel this signal can be connected to GND.
Signals to TX Logic Blocks		
DINxx[7:0]	I	Byte wide data for channel xx. This data is ultimately preset on the serial LVDS pin TXDxx_W_[P:N] (work) and TXDxx_P_[P:N] (protect).
DINxx_PAR	I	Parity input for byte wide data DINxx. Odd or even parity selection is controlled by a bit in the control register at 0x3000C.
Signals from RX Logic Blocks		
DOUTxx[7:0]	O	Byte wide data for channel AA
DOUTxx_PAR	O	Parity output for byte wide data DOUTxx[7:0]. Odd/Even is controlled by control register at 0x3000C.
DOUTxx_FP	O	Frame pulse output from the SONET framer. A single clock pulse to indicate the start of the SONET frame. If bypassing the pointer mover/interpreter this pulse will line up directly with the first A1 on DOUTxx[7:0]. If using the pointer interpreter/mover DOUTxx_FP will fall several clock cycles before the A1 on DOUTxx[7:0] due to the latency from the pointer mover.
DOUTxx_SPE	O	When '1' indicates SPE bytes are on the DOUTxx[7:0] lines. Only available when using the pointer interpreter/mover
DOUTxx_C1J1	O	When '1' indicates the C1J1 bytes are on the DOUTxx[7:0] lines. Only available when using the pointer interpreter/mover.
DOUTxx_EN	O	Indicates the state of register setting for DOUTxx_EN.
CDR_CLK_xx	O	Recovered clock from the Channel xx SERDES. If not using the alignment FIFO all of the parallel data from Channel xx will be clocked from this clock.
Signals to TOH Logic Blocks (Note: These signals are active only in the serial TOH insertion mode)		
TX_TOH_CK_EN	I	Active-hi TOH_CLK enable. If using serial TOH insertion this enable must be active.
TOH_INxx	I	Serial TOH insertion port for channel xx.
Signals from TOH Logic Blocks (Note: These signals are active only in the serial TOH insertion mode)		
RX_TOH_CK_EN	O	When '1' indicates a control register bit has been set to enable the TOH clock and frame pulse.
RX_TOH_FP	O	Single clock frame pulse to indicate the serial link frame pulse.
TOH_CK_FP_EN	O	When '1' indicates the TOH serial link clock is enabled.
TOH_OUTxx	O	TOH serial link output from Channel xx

Table 15. FPGA/Embedded Core Interface Signals (Continued)

ORT8850 FPGA/Embedded Core Interface Signals - SONET Blocks		
FPGA/Embedded Core Interface Signal Name xx=[AA,…,BD]	**Input (I) to or Output (O) from Core**	**Signal Description**
TOH_xx_EN	O	Indicates state of register settings for TOHxx_EN
Protection Switching Signals (Note: See also Table 17 and Table 18)		
PROT_SWITCH_AA	I	Parallel protection switch select, Channels AA and AB
PROT_SWITCH_AC	I	Parallel protection switch select, Channels AC and AD
PROT_SWITCH_BA	I	Parallel protection switch select, Channels BA and BB
PROT_SWITCH_BC	I	Parallel protection switch select, Channels BC and BD
LVDS_PROT_AA	I	LVDS protection switch select, Channel AA
LVSD_PROT_AB	I	LVDS protection switch select, Channel AB
LVDS_PROT_AC	I	LVDS protection switch select, Channel AC
LVDS_PROT_AD	I	LVDS protection switch select, Channel AD
LVDS_PROT_BA	I	LVDS protection switch select, Channel BA
LVDS_PROT_BB	I	LVDS protection switch select, Channel BB
LVDS_PROT_BC	I	LVDS protection switch select, Channel BC
LVDS_PROT_BD	I	LVDS protection switch select, Channel BD

Clock and Data Timing at the FPGA/Embedded Core Interface - SONET Block

(Note: This section assumes a basic understanding of the Lattice Semiconductor ispLEVER design tool set)

This section provides examples of the clock and data timing relationships at the FPGA/Embedded Core interface for both the parallel SONET data and the serial TOH data. The initiation of a change of data is referred to as the "launch" time and the actual time of capture of the data is referred to as the "capture" time. Two relationships are discussed, the relationship between data and clock at the interface itself and the relative timing constraints on the signals in the FPGA logic between the interface and the launch/capture latch in the FPGA portion of the FPSC.

The ispLEVER place and route tool will automatically attempt to meet the timing constraints by placing a frequency constraint on the corresponding clock and will report a non-routed condition if it is unable to do so. Trace reports should also be generated using ispLEVER to evaluate both the setup and the hold margins.

The typical timing numbers used in the discussions are for illustration purposes and can vary due to both process and environmental variations and to differences in the routing through the FPGA logic, especially for the data path. Exact timing numbers should always be obtained from ispLEVER.

In all of the discussions in this section, the maximum reference clock frequency of 106 MHz is assumed. The primary clock path delay was assumed to be 3 ns - this delay is well controlled in the FPGA logic. A secondary clock path delay can vary from 1 to 3.5 ns - a delay of 2.5 ns was used in the payload data discussions and 2 ns for the TOH discussions.

The five cases considered in the discussion are shown in Table 16. The clock routing and timing configurations shown in this section are recommended for the general user since they give the best timing margins. In the discussion, if both the core and FPGA launch and latch data on the same edge, it is referred to as a "full cycle" mode. If they launch and latch on different edges, it is referred to as a "half cycle" mode.

Table 16. Operating Modes and Data Paths - SONET Logic Block

Case	Data (Note: xx =[AA, …BD])	Data Path	Embedded Core Clock Launch/Latch	FPGA Clock Launch/Latch	Clock/Route
1	DOUTxx[7:0]	Core to FPGA	Falling Edge	Falling Edge	CDR_CLK_xx/Secondary
2	DOUTxx[7:0]	Core to FPGA	Falling Edge	Rising Edge	FPGA_SYSCLK/Primary
3	DINxx[7:0]	FPGA to Core	Rising Edge	Rising Edge	FPGA_SYSCLK/Primary
4	TOH_OUTxx	Core to FPGA	Rising Edge	Rising Edge	From FPGA/Secondary
5	TOH_INxx	FPGA to Core	Falling Edge	Rising Edge	From FPGA/Secondary

All timing is referenced to the clock signal at the FPGA/Core interface. Data is also timed for signals at the FPGA/Core interface. There will be additional time delays until the interface signals reach the capturing latch. The primary or secondary path delay is controlled, as noted earlier, and the clock timing at the capture latch can be predicted. The data delay, however, may be unique to each interconnect routing.

The timing diagrams provide a quantitative picture of the relative importance of setup and hold margins for the cases discussed. In the diagrams, the launch and capture times and the time difference between the launching and capturing clock edges are identified. As the time between launch and capture increases (up to a full clock period), the possibility of a setup time problem decreases. Also, the possibility of a setup time problem decreases for smaller maximum propagation delay values.

If capture occurs before the next data is launched, a hold time problem cannot occur. In nearly all cases, the difference between the launch and capture clock edges will be nearly a full clock cycle and the data will be captured before the next data is launched. This is not guaranteed, however, and ispLEVER timing analysis should be done for each application.

The general rules used for the FPGA/Core interface are as follows:

1. If possible, transfers across the FPGA/Core interface should be direct register to register transfers with minimal or preferably no intervening logic.

2. Use positive (rising) edge flip-flops in the FPGA for both input and output unless a timing diagram (case 1) explicitly indicates otherwise, or a special case (long routing path, etc.) is being considered.

3. Attempt to ‘locate’ the FPGA side flip-flops reasonably close to the interface unless other timing constraints prevent this. This ‘locate’ is typically achieved by placing a frequency constraint on the FPGA_CLK signal. In most cases, up the 3 ns of data path delay through the FPGA logic in the ORT8850 is acceptable.

4. Pay attention to the clock routing resource recommended (these are fixed on the ORT8850), and to the delay and skew limits and the clock source points.

5. Run Trace setup and hold checks in ispLEVER on the routed design taking the environmental constraints into account. (See ispLEVER Application Note for details).

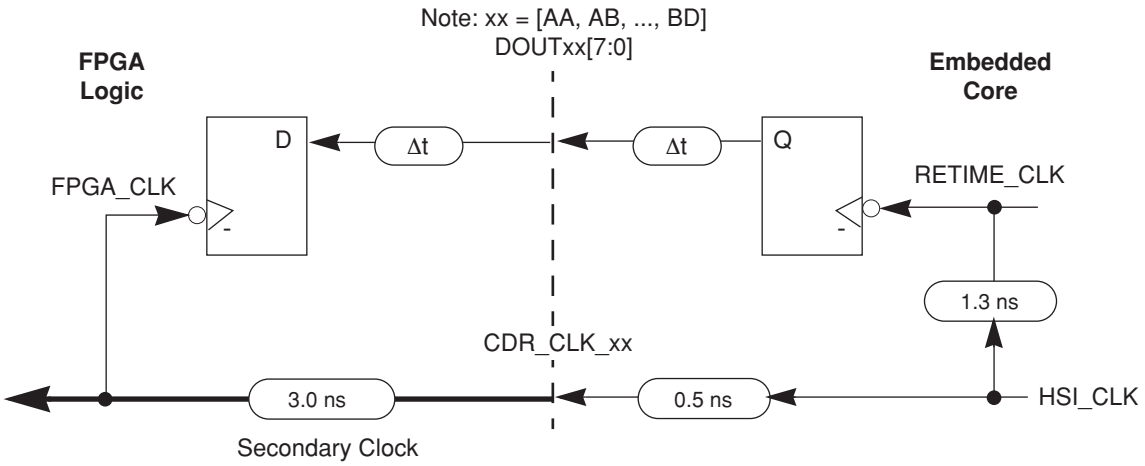
For the cases where parallel data is output from the core, the reference clock is also output from the core and the effects of propagation delay variation are included in the discussion. Propagation delay is defined relative to the interface signals and thus is the time from the enabling (falling) edge of the clock from the core to the time that data is guaranteed to be valid at the interface. As an example, for the first case discussed, the minimum (tprop_min) and maximum (tprop_max) propagation delays are 0.8 ns. and 4.7 ns. respectively. Therefore the data outputs are stable for 6.1 ns. (10 ns. - 3.9 ns.) of each clock cycle. The data must be captured during this stable period, i.e., the data signals must arrive at the capturing latch with adequate setup and hold margins versus the clock signal at the latch.

In the first case, Figure 27, the alignment FIFO is assumed to be bypassed and all timing is with respect to the recovered clock. The FPGA is latched on the falling edge of the clock, an exception to the general recommenda-

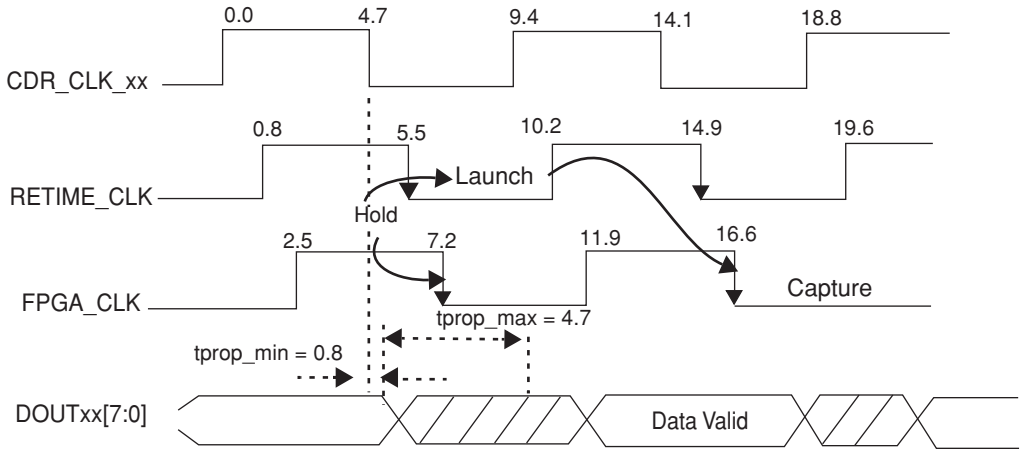
tions. (The clock edge on which data is latched in the core is hard wired to be the falling edge.) Since the falling edge of the clock (FPGA_CLK) at the FPGA latch occurs after the next data byte is launched, the delay from the interface to the FPGA latch must be large enough that an acceptable hold time margin is obtained. However the maximum propagation delay is fairly large, so a half cycle approach might lead to setup time problems.

Figure 27. Full Cycle, Alignment FIFO Bypass Mode Output Configuration and Timing (-1 Speed Grade)

a. Configuration



b. Timing (ns)



In the case shown in Figure 28 the alignment FIFO is used and all timing is with respect to the single reference clock, which is routed through the FPGA as a primary clock. The capturing clock edge occurs after the launch of the next data byte, so hold time margin is of concern and an acceptably margin should be verified. Launched data has nearly a full clock period to become stable at the capture latch, so setup margin should not be a problem. Moving the capture to the rising clock edge might give a setup time margin problem.

Figure 28. Half Cycle, Alignment Mode Output Configuration and Timing (-1 Speed Grade)

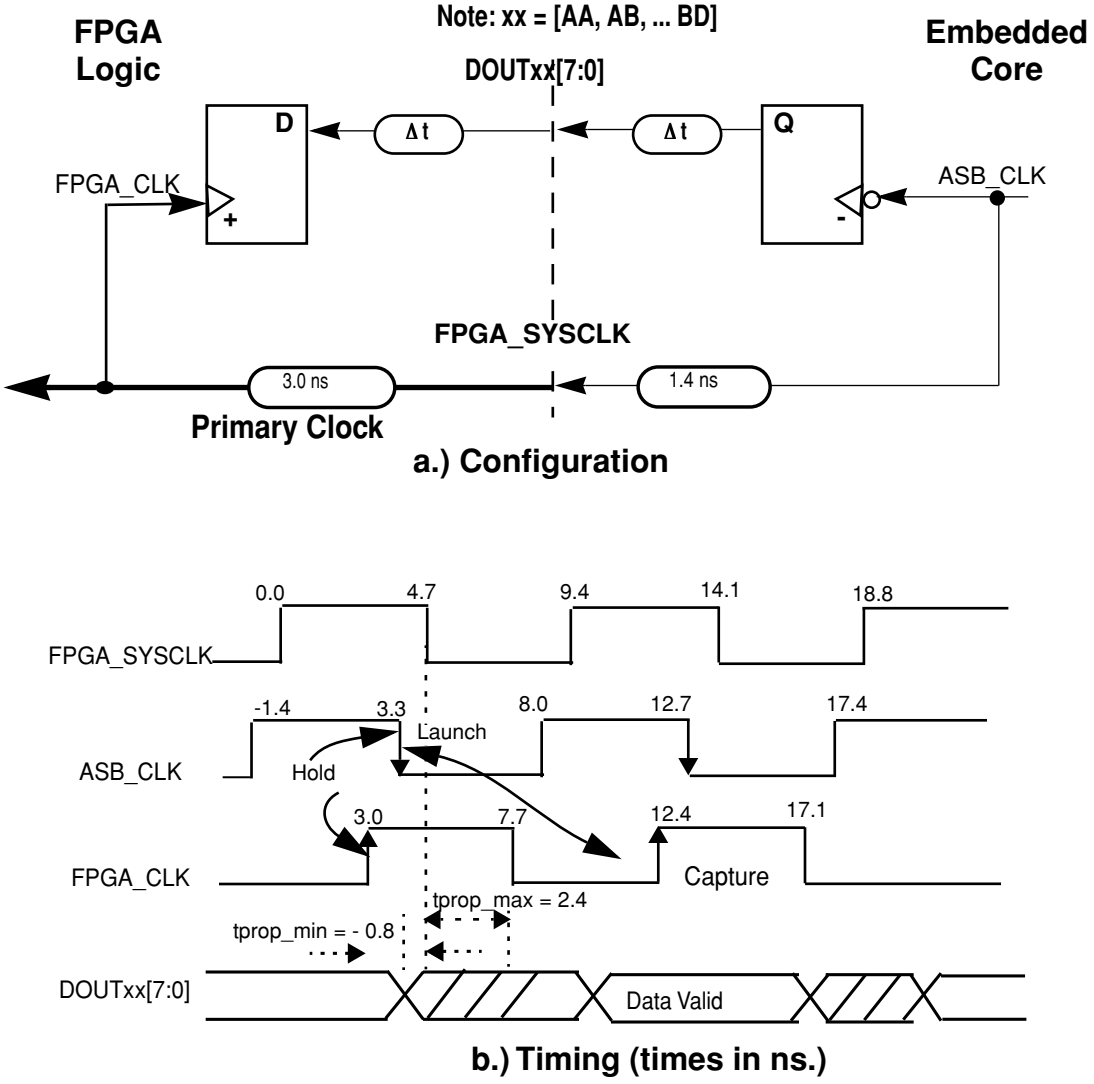
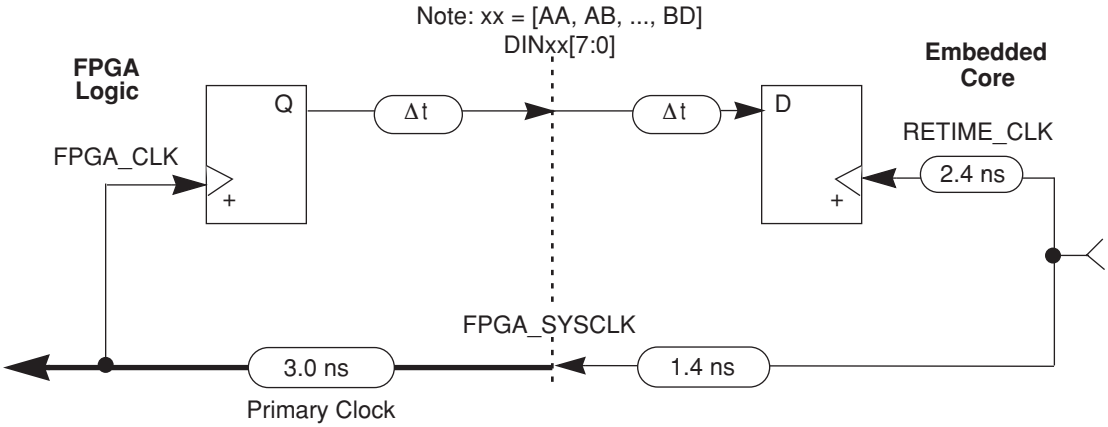


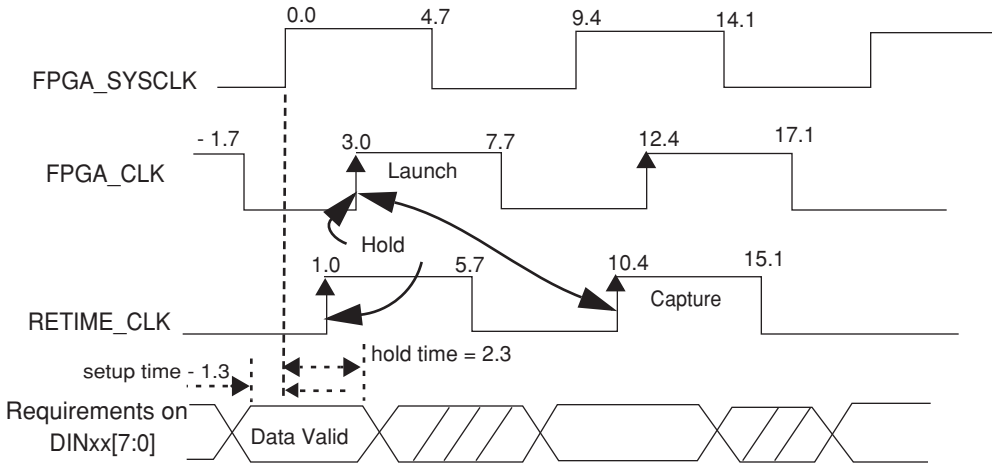
Figure 29 shows the timing for sending data from the FPGA logic to the Core. In the input case, the constraints on the data are specified in terms of setup and hold times on the data at the interface relative to the clock at the interface. For correct operation these constraints must be met. In the case shown, launch and capture occur on the same (rising) clock edge. Data is captured before the next data is launched, so there will be no hold margin problem. Launched data also has nearly a full clock period to become stable at the capture latch, so setup margin should not be a problem.

Figure 29. Full Cycle, Align and Bypass Mode Input Configuration and Timing (-1 Speed Grade)

a.) Configuration



a.) Timing (ns)

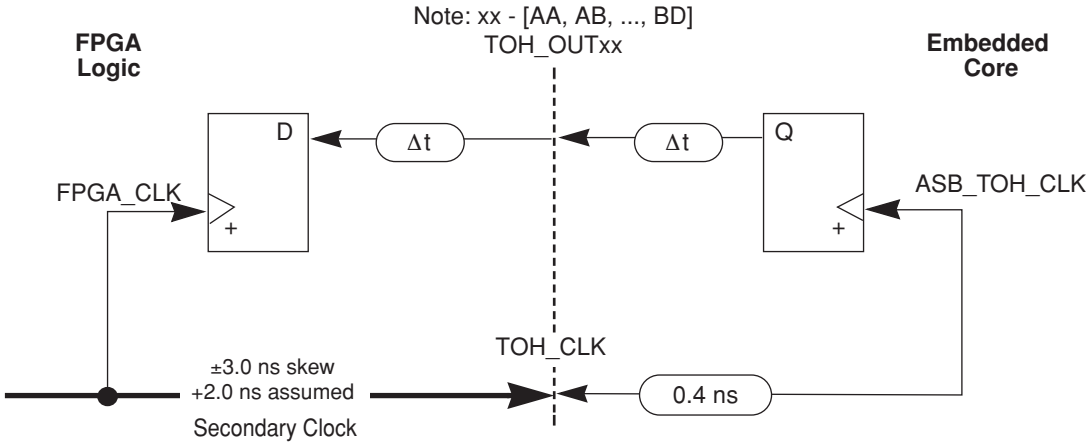


The next two examples show timing for serial TOH data input and output. For these cases, the clock is generated in the FPGA logic and the discussion accounts for the skew between the clock signal at the FPGA latch and at the FPGA/Core interface. The clock is routed over a secondary clock path and the skew can vary by ± 3 ns. A value of + 2 ns was assumed in the discussions.

Figure 30 shows the timing for sending serial TOH data from the Core to the FPGA logic with data being launched and latched on the same (rising) clock edge. As in the previous examples, setup and hold time constraints for the data versus the reference clock at the capturing latch must be met. Data is not captured before the next data is launched, so there might be a hold time margin problem. Launched data has nearly a full clock period to become stable at the capture latch and the maximum propagation delay is only 0.2 ns so setup margin should not be a problem for the timing relationships assumed. Actual timing analysis should be performed for each application because of the wide range of possible skew values.

Figure 30. Full Cycle, TOH Output Configuration and Timing (-1 Speed Grade)

a. Configuration



b. Timing (ns)

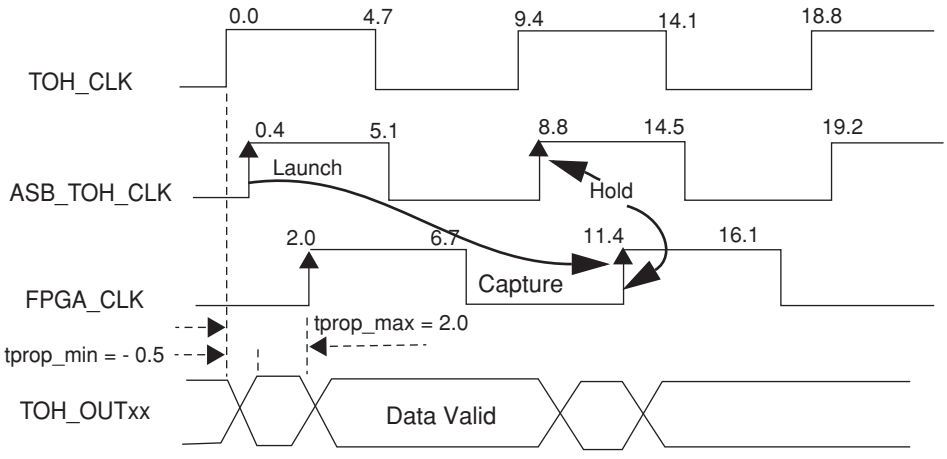
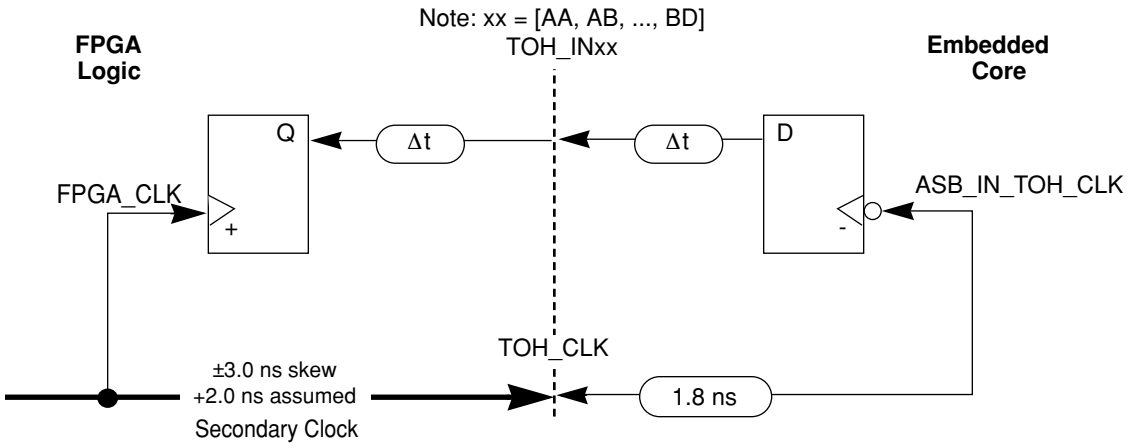


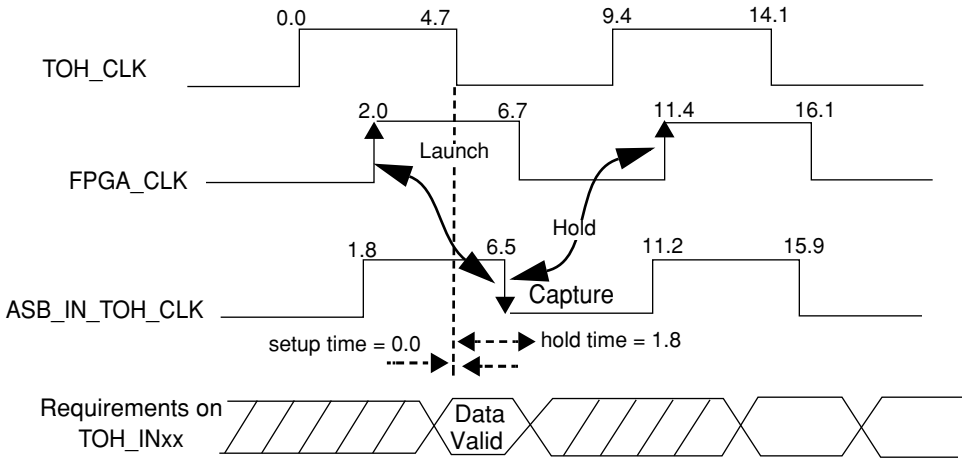
Figure 31 shows the timing for sending TOH data from the FPGA logic to the Core. As in the earlier input example, the constraints on the data are specified in terms of setup and hold times on the data at the interface relative to the clock at the interface. In the case shown, launch and capture occur on different clock edges (rising edge in the FPGA). Data is captured before the next data is launched, so there will be no hold margin problem. Launched data also has nearly a full clock period to become stable at the capture latch, so setup margin should not be a problem for the timing relationships assumed in the example. Actual timing analysis should be performed for each application because of the wide range of possible skew values.

Figure 31. Half Cycle, TOH Input Configuration and Timing (-1 Speed Grade)

a. Configuration



b. Timing (ns)



Powerdown Mode

Powerdown mode will be entered when the corresponding channel is disabled. Channels can be independently enabled or disabled under software control.

Parallel data bus output enable and TOH serial data output enable signals are made available to the FPGA logic. The HSI macrocell's corresponding channel is also powered down. The device will power up with all eight channels in powerdown mode.

Protection Switching

There is built-in protection switching between the SERDES channels, in the receive direction of the ORT8850. Protection switching allows pairs of SERDES channels to act as main and protect data links, and to switch between the main and protect links via a control register or FPGA interface port. There are two types of protection switches: parallel and LVDS.

Parallel protection switching takes place just before the FPGA interface ports, and after the alignment FIFO. The alignment FIFO must be used for this type of protection switching. It is possible to bypass the pointer interpreter/mover and still use the parallel protection switching. In this mode, SERDES channels AA and AB are used as main and protect. When selected for main, channel AA is used to provide data on interface ports AA. When selected for protect, channel AB is used to provide data on FPGA interface ports AA. The same scheme is used for channel groupings AC/AD, BA/BB, and BC/BD

There are two ways to control the parallel protection switching, interface signal and software control. On the FPGA interface, there are 4 input signals to the ORT8850 core that will select between a main and a protect channel. When using the interface signal to control protection switching, only the parallel data is switched; the serial TOH data outputs are not switched.

Software control will switch both the parallel data and the serial TOH data outputs to the FPGA. The software control register is found at 0x30009 in the memory map (Table 19).

Table 17. Register Settings, Parallel Protection Switching

FPGA Interface Signal	When '0'	When '1'
PROT_SWITCH_AA	Channel AB data on DOUTAA	Channel AA data on DOUTAA
PROT_SWITCH_AC	Channel AD data on DOUTAC	Channel AC data on DOUTAC
PROT_SWITCH_BA	Channel BB data on DOUTBA	Channel BA data on DOUTBA
PROT_SWITCH_BC	Channel BD data on DOUTBC	Channel BC data on DOUTBC

LVDS protection switching takes place at the LVDS buffer before the serial data is sent into the Data Recovery (CDR). The selection is between the main LVDS buffer and the protect LVDS buffer. The work LVDS buffers are TXDxx_W_[P:N], while the protect LVDS buffers are TXDxx_P_[P:N]. When operating using the LVDS buffers (default), no status information is available on the protect LVDS buffers since the serial stream must reach the SONET framer before status information is available on the data stream. The same is also true for the work LVDS buffers when operating with the protect buffers.

There are two ways to control the LVDS protection switching, interface and software control. On the FPGA interface, there are eight input signals to the ORT8850 core that will select between the work and protect LVDS buffers.

Table 18. LVDS Protection Switching

FPGA Interface Signal	When '0'	When '1'
LVDS_PROT_AA	Channel AA gets TXD_AA_W_[P:N]	Channel AA gets TXD_AA_P_[P:N]
LVDS_PROT_AB	Channel AB gets TXD_AB_W_[P:N]	Channel AB gets TXD_AB_P_[P:N]
LVDS_PROT_AC	Channel AC gets TXD_AC_W_[P:N]	Channel AC gets TXD_AC_P_[P:N]
LVDS_PROT_AD	Channel AD gets TXD_AD_W_[P:N]	Channel AD gets TXD_AD_P_[P:N]

Table 18. LVDS Protection Switching (Continued)

FPGA Interface Signal	When '0'	When '1'
LVDS_PROT_BA	Channel BA gets TXD_BA_W_[P:N]	Channel BA gets TXD_BA_P_[P:N]
LVDS_PROT_BB	Channel BB gets TXD_BB_W_[P:N]	Channel BB gets TXD_BB_P_[P:N]
LVDS_PROT_BC	Channel BC gets TXD_BC_W_[P:N]	Channel BC gets TXD_BC_P_[P:N]
LVDS_PROT_BD	Channel BD gets TXD_BD_W_[P:N]	Channel BD gets TXD_BD_P_[P:N]

For software control of the LVDS protection switching there is an enable bit to enable software control, and a bit per channel which selects main or protect. The enable register is at 0x30008 in the memory map (Table 19).

Memory Map

The memory map for the ORT8850 core is only part of the full memory map of the ORT8850 device. The ORT8850 is an ORCA Series4 based device and thus uses the system bus as a communication bridge. The ORT8850 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 ORT8850 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 system debugging. The device ID is located at absolute address 0x00000 - 0x00003. The ORT8850H's device ID is 0xDC0123C0 and the ORT8850L's device ID is 0xDC0121C0. More information on the device ID and other Series 4 generic registers can be found in the MPI/System Bus Application Note.

The ORT8850 core registers are clocked by the reference clock SYS_CLK_P/N. If a clock is not provided to the reference clock, the registers will fail to operate.

The ORT8850 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.

Registers Access and General Description

The memory map comprises three address blocks:

- Generic register block: ID, revision, scratch pad, lock and reset register.
- Device register block: control and status bits, common to the eight channels in each of the two quad interfaces.
- Channel register blocks: each of the four channels in both quads have an address block. The four address blocks in both quads have the same structure, with a constant address offset between channel register blocks.

All registers are write-protected by the lock register, except for the scratch pad register. The lock register is a 16-bit read/write register. Write access is given to registers only when the key value 0x0580 is present in the lock register. An error flag will be set upon detecting a write access when write permission is denied. The default value is 0x0000.

After power-up reset or soft reset, unused register bits will be read as zeros. Unused address locations are also read as zeros. Bit in write-only registers will always be read as zeros.

This table is constructed to show the correct values when read and written via the system bus MPI interface. **When using this table while interfacing with the system bus user logic master interface, the data values will need to be byte flipped.** This is due to the opposite orientation of the MPI and master interface bus ordering. More information on this can be found in the MPI/System Bus Application Note (TN1017).

Table 19. Memory Map Descriptions

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30000	[0:7]	R	-	05	Internal device revision
30001	[0:7]	R	-	80	Internal device revision
30002	[0:7]	R	-	80	Internal device revision
30003	[0:7]	R/W	scratch pad	00	The scratch pad has no function and is not used anywhere in the core. However, this register can be written to and read from for debugging purposes.
30004	[0:7]	R/W	lockreg MSB	00	In order to write to registers in memory locations 0x30006 to 0x300FF, lockreg MSB and lockreg LSB must be respectively set to the values of 05 and 80. If the MSB and LSB lockreg values are not set to {05, 80}, then any values written to the registers in memory locations 0x30006 to 0x300FF will be ignored. After reset (both hard and soft), the core is in a write locked mode. The core needs to be unlocked before it can be written to. Also note that the scratch pad register (0 x 30003) can always be written to as it is unaffected by write lock mode.
30005	[0:7]	R/W	lockreg LSB	00	
30006	[0]	R/W	global reset	0	The global reset is a soft (software initiated) reset which will have the exact reset effect as a hard (RST_N pin) reset.This is a pulse register and does not have to be cleared.
	[1-7]	-	Not Used	0	
30007	[0:7]	-	Not Used	00	
Device Register Blocks					
30008	[0]	R/W	LVDS loopback control	0	0 = No Loopback 1 = LVDS loopback, transmit to receive. TX serial data is looped back to the RX serial input. TX data is still available at the TX pins
	[1]	-	Not Used	0	
	[2]	-	Not Used	0	
	[3]	R/W	LVDS Protection Switch enable	0	0 = Protection switching performed via bit settings in registers 0x30037 etc. 1 = Protection switching performed via hardware pins LVDS_PROT_SWITCH_xx
	[4]	R/W	TOH RX serial enable	0	TOH_CK_FP_EN signal
	[5-7]	-	Not Used	0	

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30009	[0]	R/W	TOH serial port output MUX select for AA/AB	1	0 = AB TOH is output on AA 1 = AA TOH is output on AA
	[1]	R/W	TOH serial port output MUX select AC/AD	1	0 = AD TOH is output on AC 1 = AC TOH is output on AC
	[2]	R/W	DOUT parallel port output MUX select for AA/AB	1	0 = AB data is output on AA 1 = AA data is output on AA
	[3]	R/W	DOUT parallel port output MUX select for AC/AD	1	0 = AD data is output on AC 1 = AC data is output on AC
	[4]	R/W	TOH serial port output MUX select for BA/BB	1	0 = BB TOH is output on BA 1 = BA TOH is output on BA
	[5]	R/W	TOH serial port output MUX select for BC/BD	1	0 = BD TOH is output on BC 1 = BC TOH is output on BC
	[6]	R/W	DOUT parallel port output MUX select for BA/BB	1	0 = BB data is output on BA 1 = BA data is output on BA
	[7]	R/W	DOUT parallel port output MUX select BC/BD	1	0 = BD data is output on BC 1 = BC data is output on BC
3000A	[0:4]	R/W	FIFO aligner threshold value (min)	40 decimal 2	Minimum threshold value for the per channel receive direction alignment FIFOs. If and when the minimum threshold value is violated by a particular channel, then the “FIFO aligner threshold error” alarm bit will be generated for that channel and if enabled, latched as a “FIFO aligner threshold error flag” in the respective channel alarm register. The allowable range for minimum threshold values is 1 to 23. Note that the minimum FIFO aligner threshold value applies to all eight channels. MSB bit is 4.
	[5-7]	-	Not Used	N/A	
3000B	[0:4]	-	FIFO aligner threshold value (max)	A8 decimal 15	The allowable range for maximum threshold values is 0 to 22. MSB bit is 4
	[5-7]	-	Not Used	N/A	

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
3000C	[0:3]	R/W	number of consecutive A1 A2 errors to generate [0:3]	00	If a particular channel's "A1 A2 error insert command" control bit is set to the value 1 then the "A1 and A2 error insert values" will be inserted into that channels respective A1 and A2 bytes. The number of consecutive frames to be corrupted is determined by the "number of consecutive A1 A2 errors to generate [0:3]" control bits. MSB is bit 3
	[4]	R/W	backplane side loopback control	0	0 = No loopback. 1 = RX to TX loopback on backplane side. Serial input is run through SERDES and SONET block, then looped back in parallel to SERDES and out serial.
	[5]	R/W	DINxx/DOUTxx parallel bus parity control	1	0 = Odd parity 1 = Even parity
	[6]	R/W	scrambler/descrambler	1	0 = no RX direction, descramble / TX direction scramble 1 = In RX direction, descramble channel after the SONET frame recovery. In TX direction, scramble data just before parallel-to-serial conversion
	[7]	-	Not Used	0	
3000D	[0:7]	R/W	A1 error insert value [0:7]	00	Value of the A1 byte for error insert
3000E	[0:7]	R/W	A2 error insert value [0:7]	00	Value of the A2 byte for error insert
3000F	[0:7]	R/W	transmit B1 error insert mask [0:7]	00	0 = No error insertion. 1 = Invert corresponding bit in B1 byte.
30010	[0]	R	AA alarm	0	Consolidation alarm for channel AA 1 = alarm 0 = no alarm.
	[1]	R	AB alarm	0	Consolidation alarm for channel AB 1 = alarm 0 = no alarm.
	[2]	R	AC alarm	0	Consolidation alarm for channel AC 1 = alarm 0 = no alarm.
	[3]	R	AD alarm	0	Consolidation alarm for channel AD 1 = alarm 0 = no alarm.
	[4-7]	-	Not Used	0	
30011	[0]	R/W	AA/BA alarm enable/mask register	0	AA and BA enable 1 = enabled 0 = not enabled
	[1]	R/W	AB/BB alarm enable/mask register	0	AB and BB enable 1 = enabled 0 = not enabled
	[2]	R/W	AC/BC alarm enable/mask register	0	AC and BC enable 1 = enabled 0 = not enabled
	[3]	R/W	AD/BD alarm enable/mask register	0	AD and BD enable 1 = enabled 0 = not enabled
	[4-7]	-	Not Used	0	

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30012	[0]	R	frame offset error flag	0	If in the receive direction the phase offset between any two channels exceeds 17 bytes, then a frame offset error event will be issued. This condition is continuously monitored. Write a “1” to clear this bit
	[1]	R	write to locked register error flag	0	If the core memory map has not been unlocked (by writing to the lock registers), and any address other than the lockreg registers or scratch pad register is written to, then a “write to locked register” event will be generated. Write a “1” to clear this bit
	[2-7]	-	Not Used	N/A	
30013	[0]	R/W	frame offset error enable	0	Frame offset error flag enable. 0 = not enable 1 = enable
	[1]	R/W	write to locked register for error enable	0	Write to locked register error flag enable 0 = not enable 1 = enable
	[2-7]	-	Not Used	0	
30014	[0]	R	BA alarm	0	Consolidation alarm for channel BA 0 = no alarm 1 = alarm
	[1]	R	BB alarm	0	Consolidation alarm for channel BB 0 = no alarm 1 = alarm
	[2]	R	BC alarm	0	Consolidation alarm for channel BC 0 = no alarm 1 = alarm
	[3]	R	BD alarm	0	Consolidation alarm for channel BD 0 = no alarm 1 = alarm
	[4-7]	R	Not Used	0	
30015	[0:7]	-	Not Used	00	
30016	[0:1]	R/W	STM A mode control	0	00 - Quad STS-12 or STS-48. 10 - Quad STS-3.
	[2:3]	R/W	STM B mode control	0	00 - Quad STS-12 or STS-48. 10 - Quad STS-3.
	[4-7]	-	Not Used	0	

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30017	[0]	R/W	BD resync	0	Channel BD alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO. Write “0” for normal operation
	[1]	R/W	BC resync	0	Channel BC alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO. Write “0” for normal operation
	[2]	R/W	BB resync	0	Channel BB alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO. Write “0” for normal operation
	[3]	R/W	BA resync	0	Channel BA alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO. Write “0” for normal operation
	[4]	R/W	AD resync	0	Channel AD alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO.Write “0” for normal operation
	[5]	R/W	AC resync	0	Channel AC alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO.Write “0” for normal operation
	[6]	R/W	AB resync	0	Channel AB alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO.Write “0” for normal operation
	[7]	R/W	AA resync	0	Channel AA alignment FIFO resync. Write “1” to resync this channel. When no alignment is used, this resets the read pointer to the middle of the FIFO.Write “0” for normal operation
30018	[0]	R/W	AD/BD resync	0	2-link AD/BD alignment FIFO resync. Write “1” to resync this link. Write “0” for normal operation.
	[1]	R/W	AC/BC resync	0	2-link AC/BC alignment FIFO resync. Write “1” to resync this link. Write “0” for normal operation.
	[2]	R/W	AB/BB resync	0	2-link AB/BB alignment FIFO resync. Write “1” to resync this link. Write “0” for normal operation.
	[3]	R/W	AA/BA resync	0	2-link AA/BA alignment FIFO resync. Write “1” to resync this link. Write “0” for normal operation.
	[4]	R/W	STM B resync	0	Quad B alignment resync. Write “0” for normal operation.
	[5]	R/W	STM A resync	0	Quad A alignment FIFO resync Write “0” for normal operation.
	[6]	R/W	All 8 resync	0	All 8 channel alignment FIFO resync. Write “0” for normal operation.
	[7]	-	Not Used	N/A	

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30019	[0:7]	-	Not Used	00	
Channel Register Blocks					
30020* 30038 30050 30068 30080 30098 300B0 300C8	[0]	R/W	AIS-L insert in OOF	0	0 = When RX direction OOF occurs, do not insert AIS-L. 1 = When RX direction OOF occurs, insert AIS-L.
	[1]	R/W	AIS-L control	0	0 = Do not force AIS-L insert 1 = Always force AIS-L insert
	[2]	R/W	TOH output parity error insert	0	0 = Do not insert a parity error 1 = Insert parity error in parity bit of receive TOH serial output for as long as this bit is set
	[3]	R/W	RX K1/K2 source select	0	0 = Set receive direction K1 K2 bytes to 0. 1 = Pass receive direction K1 K2 through pointer mover.
	[4]	R/W	DOUTxx bus parity error insert	0	0 = Do not insert parity error. 1 = Insert parity error in DOUTxx_PAR for as long as this bit is set.
	[5]	R/W	channel enable/disable control	0	0 = Power down CDR channels 1 = Functional mode.
	[6]	R/W	DOUTxx_EN	0	DOUTxx_EN signal
	[7]	R/W	TOH_EN	0	TOHxx_EN signal
30021* 30039 30051 30069 30081 30099 300B1 300C9	[0]	R/W	D9 source select	0	0 = Insert D9 from TOH_INxx 1 = Pass through D9 from DINxx
	[1]	R/W	D10 source select	0	0 = Insert D10 from TOH_INxx 1 = Pass through D10 from DINxx
	[2]	R/W	D11 source select	0	0 = Insert D11 from TOH_INxx 1 = Pass through D11 from DINxx
	[3]	R/W	D12 source select	0	0 = Insert D12 from TOH_INxx 1 = Pass through D12 from DINxx
	[4]	R/W	K1 K2 source select	0	0 = Insert K1, K2 from TOH_INxx 1 = Pass through K1, K2 from DINxx
	[5]	R/W	S1 M0 source select	0	0 = Insert S1, M0, from TOH_INxx 1 = Pass through S1 M0 of DINxx
	[6]	R/W	E1 F1 E2 source select		0 = Insert E1, F1, E2 from TOH_INxx on FPGA interface 1 = Pass through E1, F1, E2 TOH bytes of DINxx
	[7]	R/W	TOH source select	0	0 = Insert TOH from TOH_INxx on FPGA interface for transmit 1 = Pass through all TOH DINxx for transmit
30022* 3003A 30052 3006A 30082 3009A 300B2 300CA	[0:7]	R/W	D1~D8 source select	00	0 = Insert TOH for transmit from TOH_INxx from the FPGA interface. 1 = Pass through D1~D8 TOH bytes from DINxx.

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30023* 3003B 30053 3006B 30083 3009B 300B3 300CB	[0]	R/W	A1 A2 error insert command	0	0 = Do not insert error. 1 = Insert error for number of frames in register 0x3000C. The error insertion is based on a rising edge detector. As such, the control must be set to value 0 before trying to indicate a second A1, A2 corruption
	[1]	R/W	B1 error insert command	0.	0 = Do not insert error 1 = Insert error marked in register 0x3000F. The error insertion is based on a rising edge detector. As such, the control must be set to value 0 before trying to initiate a second B1 corruption.
	[2]	R/W	disable B1 insert	0	0 = B1 is inserted in the transmit direction by the SONET block 1 = B1 is not inserted in the transmit direction
	[3]	R/W	disable A1/A2 insert	0	0 = A1/A2 is inserted in the transmit direction by the SONET block 1 = A1/A2 is not inserted in the transmit direction
	[4-7]	-	Not Used	0	
30024* 3003C 30054 3006C 30084 3009C 300B4 300CC	[0:3]	R	concat indication 3, 6, 9, 12	0	The value 1 in any bit location indicates that STS# is in CONCAT mode. 0 = Not in concatenation mode or is the head of concatenated group 1 = indicates the channel is concatenated
	[4-7]	-	Not Used	0	
30025* 3003D 30055 3006D 30085 3009D 300B5 300CD	[0:7]	R	concat indication 1, 4, 7, 10, 2, 5, 8, 11	0	The value 1 in any bit location indicates that STS# is in CONCAT mode. 0 = Not in concatenation mode or is the head of concatenated group 1 = indicates the channel is concatenated
30026* 3003E 30056 3006E 30086 3009E 300B6 300CE	[0]	R	Channel alarm bit	0	Set when any of the alarms in the channel alarm register (0x30028) are set and the alarm is enabled. This alarm is enabled in 0x30027 bit 0 for channel AA etc.
	[1]	R	AIS-P flag	0	Set when any alarm for AIS-P is set and the corresponding enable is set.
	[2]	R	Pointer mover elastic store overflow flag	0	Set when the elastic store in the pointer mover write and read address is within 1 byte. Alarm enable is 0x30027 bit 2.
	[3-7]	-	Not Used	0	

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30027* 0303F 30057 3006F 30087 3009F 300B7 300CF	[0]	R/W	enable channel alarm	0	Channel alarm bit (30026, ...) enable. Set to 1 to enable alarm bit to propagate to alarm 0x30010
	[1]	R/W	enable AIS-P flag	0	AIS -P flag alarm enable. Set to 1 to enable alarm bit to propagate to alarm 0x30010
	[2]		enable pointer mover elastic store overflow flag	0	Pointer mover elastic store overflow flag enable. Set to 1 to enable alarm bit to propagate to 0x30010
	[3-7]	-	Not Used	0	
30028* 30040 30058 30070 30088 300A0 300B8 300D0	[0]	R	FIFO aligner threshold error flag	00	Alarm is set to 1 if either the min or max FIFO threshold levels are violated, the min and max threshold levels can be set in address 0x3000A and 0x300B. Alarm enable is 0x30029 bit 0. Write 1 to clear this alarm bit This alarm is only valid when FIFO OOS flag is also set.
	[1]		RX internal path parity error flag		Alarm indicator on receive path internal parity error. Alarm is enabled in 0x30029 bit 1. Write 1 to clear
	[2]		OOF flag		Alarm indicator channel is OOF. Alarm enable is 0x30029 bit 2. Write 1 to clear.
	[3]		LVDS link B1 parity error flag		Alarm indicator that channel has found a B1 parity error. Alarm enable is 0x30029 bit 3. Write 1 to clear.
	[4]		DINxx parallel bus parity error flag	0	Alarm indicator channel has found a parity error on the DINxx input from the FPGA.Alarm enable is 0x30029 bit 4. Write 1 to clear.
	[5]		TOH serial input port parity error flag	0	Alarm indicator channel has found a parity error on the TOH_INxx input from the FPGA. Write 1 to clear this alarm. Alarm enable is 0x30028 bit 5.
	[6]		FIFO OOS error flag	0	Alarm indicates channel group is out of sync. Write 1 to clear. Alarm enable is 0x30028.
	[7]	-	Not Used	0	
30029* 30041 30059 30071 30089 300A1 300B9 300D1	[0:6]	R/W	channel alarm enable	00	Enable bits for channel alarm register 0x30028. Set to 1 to enable and to propagate the alarm to register 0x30026 bit 0.
	[7]	-	Not Used	0	
3002A* 30042 3005A 30072 3008A 300A2 300BA 300D2	[0:3]	R	AIS alarm flags 3, 6, 9, 12	0	These are the AIS-P alarm flags. 1 if the LVDS input STS # contains AIS.
	[4-7]	-	Not Used	0	

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
3002B* 30043 3005B 30073 3008B 300A3 300BB 300D3	[0:7]	R	AIS alarm flags 1, 4, 7, 10, 2, 5, 8, 11	00	These are the AIS-P alarm flags. 1 if the LVDS input STS # contains AIS.
3002C* 30044 3005C 30074 3008C 300A4 300BC 300D4	[0:3]	R/W	enable AIS alarm 3, 6, 9, 12	0	Enable bits for AIS alarms. Set to 1 to enable and propagate the alarm to register 0x30026.
	[4-7]	-	Not Used	0	
3002D* 30045 3005D 30075 3008D 300A5 300BD 300D5	[0:7]	R/W	AIS alarm enable 1, 4, 7, 10, 2, 5, 8, 11	00	Enable bits for AIS alarms. Set to 1 to enable and propagate the alarm to register 0x30026.
3002E* 30046 3005E 30076 3008E 300A6 300BE 300D6	[0:3]	R	Pointer mover elastic store overflow flags 12, 9, 6, 3	0	Per STS-1 pointer mover elastic store overflow alarm flags. This alarm will propagate to 0x30026 bit 2 when enabled
	[4-7]	-	Not Used	0	
3002F* 30047 3005F 30077 3008F 300A7 300BF 300D7	[0:7]	R	Pointer mover elastic store overflow flags 4, 7, 10, 2, 5, 8, 11	00	Per STS-1 pointer mover elastic store overflow alarm flags. This alarm will propagate to 0x30026 bit 2 when enabled
30030* 30048 30060 30078 30090 300A8 300C0 300D8	[0:3]	R/W	enable elastic store overflow flag 12, 9, 6, 3	0	Enable Bit for elastic store alarms. Set 1 to enable alarm and propagate alarm to register 0x30026
	[4-7]	-	Not Used	0	

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30031* 30049 30061 30079 30091 300A9 300C1 300D9	[0:7]		enable elastic store overflow flag 1, 4, 7, 10, 2, 5, 8, 11	00	Enable Bit for elastic store alarms. Set 1 to enable alarm and propagate alarm to register 0x30026.
30032* 3004A 30062 3007A 30092 300AA 300C2 300DA	[0:6]	R	B1 parity error counter	00	7 bit counter for the number of B1 parity errors in the receive direction of the channel. Clear on read. Bit 6 is MSB
	[7]	R	B1 parity error counter overflow	0	Overflow bit for B1 parity error counter
30033* 3004B 30063 3007B 30093 300AB 300C3 300DB	[0:6]	R	OOF counter	00	7 bit counter for the number of in-frame to OOF transitions. Clear on read. Bit 6 is MSB
	[7]	R	OOF counter overflow	0	Overflow bit for OOF counter
30034* 3004C 30064 3007C 30094 300AC 300C4 300DC	[0:6]	R	A1 A2 frame error counter	00	This counter increments when an errored frame pattern is detected by the framer. Note that this is different from OOF. In OOF state, you can detect the correct framing pattern and still be out-of-frame.
	[7]	R	A1, A2 error counter overflow	0	Overflow bit for A1/A2 error counter
30035* 3004D 30065 3007D 30095 300AD 300C5 300DD	[0:4]	R	FIFO depth register	30	Current value of the channel's read address of the alignment FIFO. Bit 4 is the MSB
	[5-7]	-	Not Used	0	
30036* 3004E 30066 3007E 30096 300AE 300C6 300DE	[0:7]	R	Sampler phase error counter	00	This is coming from the sampler block. The sampler looks for bit transitions 0->1->0 to determine if the transitions occur after 4 repeated bits. For e.g.: if you have 000011110000 then the 0->1->0 transition occurs in the 5th and 9th positions. It uses this to select one of the 4 repeated bits and form a repeated byte. When the transitions happen at different bit positions, then the phase error merely indicates that this has happened

Table 19. Memory Map Descriptions (Continued)

(0x) Absolute Address	Bit	Type	Name	Reset Value (0x)	Description
30037* 3004F 30067 3007F 30097 300AF 300C7 300DF	[0]	R/W	Bypass pointer mover	0	0 = use pointer mover 1 = Bypass pointer mover.
	[1]	R/W	Bypass pointer mover and alignment FIFO	0	0 = uses alignment FIFO and pointer mover 1 = Bypass alignment FIFO and pointer mover.
	[2]		Enable work/protect channels	0	Bit to control the LVDS receivers to CDR. 0 = Use LVDS receivers from HSI work channels. 1 = Use LVDS receivers from HSI protect channels.
	[3:4]	R	Multichannel alignment control	00	00 = No alignment. 10 = Align with twin (i.e., STM B stream A). 01 = Align with all 4 (i.e., STM A all streams). 11 = Align with all 8 (i.e., STM A and B all streams).
	[5]	R	RX path SONET framer	0	0 = Enable framer. 1 = Disable SONET framing data is passed through
	[6-7]	-	Not Used	0	
300E0	[0]	R/W	Reserved	0	Reserved, must be set to 0.
	[1]	R/W	CDR control register	0	Always set to zero
	[2]	-	Not Used	0	
	[3]	R/W	CDR control register	0	When set to 1, controls bypass of 16 PLL generated phases with 16 low-speed phases.
	[4]	R/W	CDR control register	0	Enables CDR loopback. 0 = No loopback. 1 = Loopback TX to RX.
	[5]	R/W	CDR control register	0	Enables bypassing of the internal 622 MHz clock with TSTCLK. Must be used for simulation 0 = Use PLL. 1 = Bypass PLL (uses TSTCLK as reference clock).
	[6]	R/W	CDR control register	0	Enables CDR test mode. Initiates CDR's built-in self-test: 0 = Regular mode. 1 = Test mode.
	[7]	-	Not Used	0	
300E1	[0:7]	R/W	Half Rate		Per Channel select for half rate mode can only be used in pure bypass mode. Bit 7 is for channel BD, bit 6 is for BC etc. 0 = full rate 1 = half rate
300E2	[0:7]	R/W	Quad Rate		Per Channel select for quad rate mode can only be used in pure bypass mode. Bit 7 is for channel BD, bit 6 is for BC etc. 0 = full rate 1 = quad rate

* For Channels AA, AB, AC, AD, BA, BB, BC, BD respectively

Note: Registers at addresses ≥ 300E3 must remain at their default (reset) settings and must not be changed by the user.

Electrical Characteristics

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 20. Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Storage Temperature	T_{stg}	-65	150	°C
Power Supply Voltage with Respect to Ground	V_{DD33}[2]	-0.3	4.2	V
	V_{DDIO}	-0.3	4.2	V
	V_{DD15}	-0.3	2.0	V
	V_{DDA_STM}[1]	-0.3	2.0	V
Input Signal with Respect to Ground	—	-0.3	V_{DDIO} + 0.3	V
Signal Applied to High-impedance Output	—	-0.3	V_{DDIO} + 0.3	V
Maximum Package Body (Soldering) Temperature	—	—	220	°C

Recommended Operating Conditions

Table 21. Recommended Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Power Supply Voltage with Respect to Ground*	V_{DD33}[2]	2.7	3.6	V
	V_{DD15}	1.425	1.575	V
	V_{DDA_STM}[1]	1.425	1.575	V
Input Voltages	V_{IN}	-0.3	V_{DDIO} + 0.3	V
Junction Temperature	T_J	-40	125	°C

For FPGA Recommended Operating Conditions and Electrical Characteristics, see the Recommended Operating Conditions and Electrical Characteristics tables in the ORCA Series 4 FPGA data sheet (ORT8850L: OR4E02, ORT8850H: OR4E06) and the ORCA Series 4 I/O Buffer Technical Note. FPSC Standby Currents (I_{DD}SB15 and I_{DD}SB33) are tested with the Embedded Core in the powered down state.

Notes:
1. V_{DDA_STM} is an analog power supply input which needs to be isolated from other power supplies on the board.
2. V_{DD33} is an analog power supply for the FPGA PLLs and needs to be isolated from other power supplies on the board.

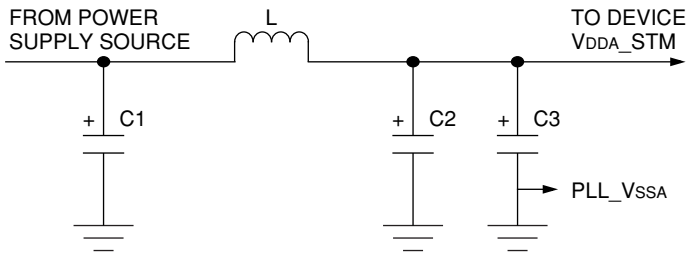
Power Supply Decoupling LC Circuit

The 850 MHz HSI macro contains both analog and digital circuitry. The data recovery function, for example, is implemented as primarily a digital function, but it relies on a conventional analog phase-locked loop to provide its 850 MHz reference frequency. The internal analog phase-locked loop contains a voltage-controlled oscillator. This circuit will be sensitive to digital noise generated from the rapid switching transients associated with internal logic gates and parasitic inductive elements. Generated noise that contains frequency components beyond the bandwidth of the internal phase-locked loop (about 3 MHz) will not be attenuated by the phase-locked loop and will impact bit error rate directly. Thus, separate power supply pins are provided for these critical analog circuit elements.

Additional power supply filtering in the form of a LC filter section will be used between the power supply source and these device pins as shown in Figure 32. The corner frequency of the LC filter is chosen based on the power supply switching frequency, which is between 100 kHz and 300 kHz in most applications.

Capacitors C1 and C2 are large electrolytic capacitors to provide the basic cut-off frequency of the LC filter. For example, the cutoff frequency of the combination of these elements might fall between 5 kHz and 50 kHz. Capacitor C3 is a smaller ceramic capacitor designed to provide a low-impedance path for a wide range of high-frequency signals at the analog power supply pins of the device. The physical location of capacitor C3 must be as close to the device lead as possible. Multiple instances of capacitors C3 can be used if necessary. The recommended filter for the HSI macro is shown below: L = 4.7µH, RL = 1Ω, C1 = 0.01µF, C2 = 0.01µF, C3 = 4.7µF.

Figure 32. Sample Power Supply Filter Network for Analog HSI Power Supply Pins



5-9344(F)

If the programmable PLLs on the FPGA portion of the device are to be used, then the VDD33 supply must isolated in the same way. More information on this and other requirements for the FPGA PLLs can be found in technical note TN1011, *ORCA Series 4 I/O Tuning via PLL* available on the Lattice web site at www.latticesemi.com.

HSI Electrical and Timing Characteristics

Table 22. Maximum Power Dissipation

Parameter	Conditions	Min.	Typ.	Max.	Units
Power Dissipation	SERDES, scrambler/descrambler, framer, FIFO alignment, pointer mover, and I/O (per channel), 622 Mbtis/s	—	—	125	mW

1. With all channels operating, 1.575 V and 3.6 V supplies, 85ºC.

Table 23. Recommended Operating Conditions

Parameter	Conditions	Min.	Typ.	Max.	Units
$V_{DD}15$ Supply Voltage	—	1.425	—	1.575	V
Junction Temperature	T_J	–40	—	125	°C

Table 24. Receiver Specifications

Parameter	Conditions	Min.	Typ.	Max.	Units
Input Data					
Stream of Nontransitions [1]	—	—	—	72	bits
Phase Change, Input Signal	Over a 200 ns time interval [2]	—	—	100	ps
Eye Opening[3]	—	0.4	—	—	UIp-p
Jitter Tolerance @ 622 Mbits/s, Worst Case	300 MV diff eye[4]	—	—	0.6	UIp-p
Jitter Tolerance @ 155 Mbits/s, Worst Case	250 MV diff eye[5]	—	—	0.85	UIp-p

1. This sequence should not occur more than once per minute.
2. Translates to a frequency change of 500 ppm.
3. A unit interval for 622.08 Mbits/s data is 1.6075 ns.
4. With STS-12 data pattern, all channels operating, FPGA logic active, REFCLK jitter of 30 ps., TA=0°C to 85°C, 1.425 V to 1.575 V supply. Jitter measured with a Wavecrest SIA-3000.
5. With STS-3 data pattern, all channels operating, FPGA logic active, REFCLK jitter of 30 ps., TA=0°C to 85°C, 1.425 V to 1.575 V supply. Jitter measured with a Wavecrest SIA-3000.

Table 25. Channel Output Jitter (622 Mbits/s)

Parameter	Conditions	Min.	Typ.[1]	Max.[1]	Units
Deterministic		—	0.09	0.10	Ulp-p
Random		—	0.11	0.14	Ulp-p
Total[2,3]		—	0.20	0.24	Ulp-p

1. With PRBS 2^7 data pattern, all channels operating, FPGA logic active, REFCLK jitter of 30 ps., 0°C to 85°C, 1.425 V to 1.575 V supply.
2. Wavecrest SIA-3000 instrument used to measure one-sigma (rms) random jitter component value. This value is multiplied by 14 to provide the peak-to-peak value that corresponds to a BER of 10^{-12}.
3. Total jitter measurement performed with Wavecrest SIA-3000 at a BER of 10^{-12}. See instrument documentation and other Wavecrest publications for a detailed discussion of jitter types included in this measurement.

Table 26. Channel Output Jitter (155 Mbits/s)

Parameter	Conditions	Min.	Typ.[1]	Max.[1]	Units
Deterministic		—	0.027	0.035	Ulp-p
Random		—	0.053	0.065	Ulp-p
Total[2,3]		—	0.08	0.10	Ulp-p

1. With PRBS 2^7 data pattern, all channels operating, FPGA logic active, REFCLK jitter of 30 ps., 0°C to 85°C, 1.425 V to 1.575 V supply.
2. Wavecrest SIA-3000 instrument used to measure one-sigma (rms) random jitter component value. This value is multiplied by 14 to provide the peak-to-peak value that corresponds to a BER of 10^{-12}.
3. Total jitter measurement performed with Wavecrest SIA-3000 at a BER of 10^{-12}. See instrument documentation and other Wavecrest publications for a detailed discussion of jitter types included in this measurement.

Table 27. Synthesizer Specifications

Parameter	Conditions	Min	Typical	Max	Unit
PLL[1]					
Loop Bandwidth	—	—	—	6	MHz
Jitter Peaking	—	—	—	2	dB
power-up Reset Time	—	10	—	—	µs
Lock Acquisition Time	—	—	—	1	ms
Input Reference Clock					
Frequency	—	62.5	—	106.25	MHz
Frequency Deviation[2]	—	-350	—	350	ppm
Phase Change	Over a 200 ns time interval[3]	—	—	100	ps

1. External 10 kΩ resistor to analog ground required.
2. The frequency deviation allowed between the transmitter reference clock and receiver reference clock on a given link.
3. Translates to a frequency change of 500 ppm.

Embedded Core LVDS I/O

Table 28. Driver DC Data

Parameter	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Output Voltage High, VOA or VOB	VOH	RLOAD = 100 Ω ± 1%	—	—	1.475	V
Output Voltage Low, VOA or VOB	VOL	RLOAD = 100 Ω ± 1%	0.925	—	—	V
Output Differential Voltage	|VOD|	RLOAD = 100 Ω ± 1%	0.25	—	0.45	V
Output Offset Voltage	VOS	RLOAD = 100 Ω ± 1%	1.125	—	1.275	V
Output Impedance, Differential	RO	VCM = 1.0 V and 1.4 V	80	100	120	W
RO Mismatch Between A and B	Δ RO	VCM = 1.0 V and 1.4 V	—	—	10	%
Change in Differential Voltage Between Complementary States	|Δ VOD|	RLOAD = 100 Ω ± 1%	—	—	25	mV
Change in Output Offset Voltage Between Complementary States	Δ VOS	RLOAD = 100 Ω ± 1%	—	—	25	mV
Output Current	ISA, ISB	Driver shorted to GND	—	—	24	mA
Output Current	ISAB	Drivers shorted together	—	—	12	mA
Power-off Output Leakage	|Ixa|, |Ixb|	VDD = 0 V VPAD, VPADN = 0 V—2.5 V	—	—	10	mA

1. VDD33 = 3.1 V—3.5 V, VDD15 = 1.4 V—1.6 V, –40 ˚C.
2. External reference, REF10 = 1.0 V ± 3%, REF14 = 1.4 V ± 3%.

Table 29. Driver AC Data[1]

Parameter	Symbol	Test Conditions	Min.	Typ.	Max.	Units
VOD Fall Time, 80% to 20%	tF	ZL = 100 Ω ± 1% CPAD = 3.0 pF, CPAD = 3.0 pF	100	—	210	ps
VOD Rise Time, 20% to 80%	tR	ZL = 100 Ω ± 1% CPAD = 3.0 pF, CPAD = 3.0 pF	100	—	210	ps
Differential Skew |tPHLA – tPLHB| or |tPHLB – tPLHA|	tSKEW1	Any differential pair on package at 50% point of the transition	—	—	50	ps

1. VDD33 = 3.1V - 3.5 V, VDD15 = 1.4V - 1.6 V, -40˚C.

LVDS Receiver Buffer Requirements

Table 30. Receiver DC Data[1]

Parameter	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Input Voltage Range, VIA or VIB	VI	|VGPD| < 925 mV DC – 1 MHz	0.0	1.2	2.4	V
Input Differential Threshold	VIDTH	|VGPD| < 925 mV 450 MHz	–100	—	100	mV
Input Differential Hysteresis	VHYST	(+VIDTHH) – (–VIDTHL)	25	—	—	mV
Receiver Differential Input Impedance	RIN	With build-in termination, center-tapped	80	100	120	Ω

1. VDD = 3.1V - 3.5V, 0 °C -125 °C.

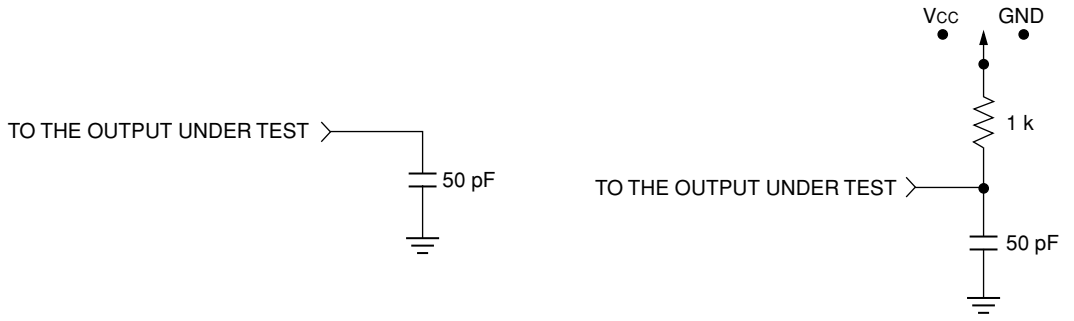
Table 31. LVDS Operating Parameters

Parameter	Test Conditions	Min.	Normal	Max.	Units
Transmit Termination Resistor	—	80	100	120	Ω
Receiver Termination Resistor	—	80	100	120	Ω
Temperature Range	—	–40	—	125	˚C
Power Supply VDD33	—	3.0	—	3.6	V
Power Supply VDD15	—	1.425	—	1.575	V
Power Supply VSS	—	—	0	—	V

Note: Under worst-case operating conditions, the LVDS driver will withstand a disabled or unpowered receiver for an unlimited period of time without being damaged. Similarly, when outputs are short-circuited to each other or to ground, the LVDS will not suffer permanent damage. The LVDS driver supports hot insertion. Under a well-controlled environment, the LVDS I/O can drive backplane as well as cable.

Input/Output Buffer Measurement Conditions (on-LVDS Buffer)

Figure 33. AC Test Loads



A. Load Used to Measure Propagation Delay **B. Load Used to Measure Rising/Falling Edges**

Note: Switch to V_{DD} for T_{PLZ}/T_{PZL}; switch to GND for T_{PHZ}/T_{PZH}.

Figure 34. Output Buffer Delays

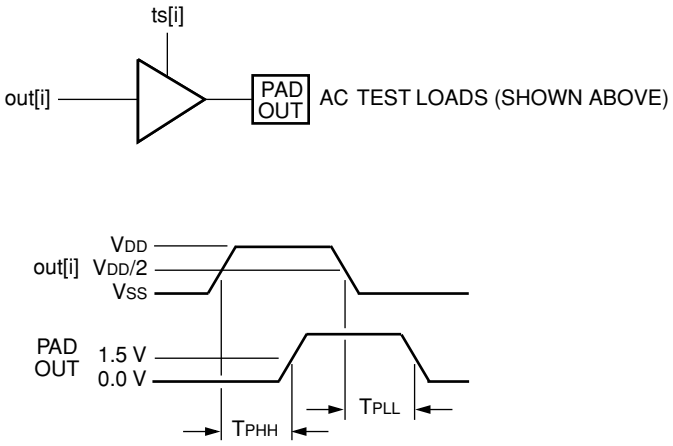
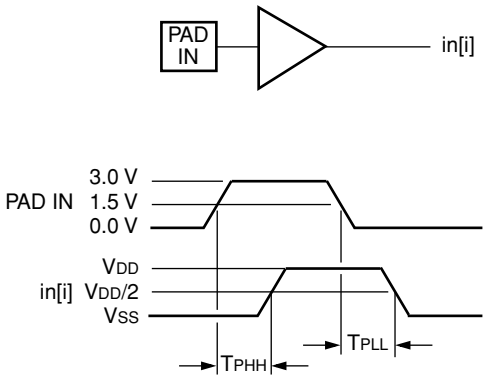


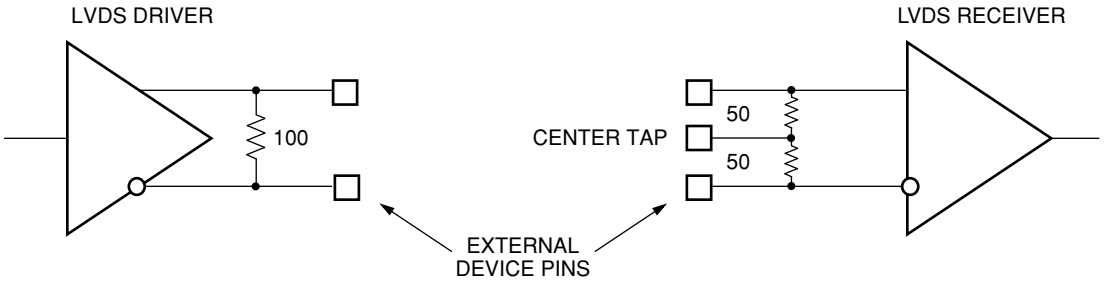
Figure 35. Input Buffer Delays



Termination Resistor

The LVDS drivers and receivers operate on a 100 Ω differential impedance, as shown below. External resistors are not required. The differential driver and receiver buffers include termination resistors inside the device package, as shown in Figure 36 below.

Figure 36. LVDS Driver and Receiver and Associated Internal Components



LVDS Driver Buffer Capabilities

Under worst-case operating condition, the LVDS driver must withstand a disabled or unpowered receiver for an unlimited period of time without being damaged. Similarly, when its outputs are short-circuited to each other or to ground, the LVDS driver will not suffer permanent damage Figure 37 illustrates the terms associated with LVDS driver and receiver pairs.

Figure 37. LVDS Driver and Receiver

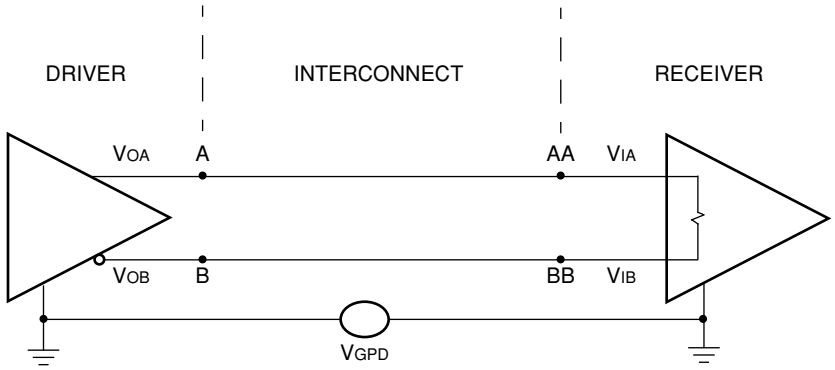
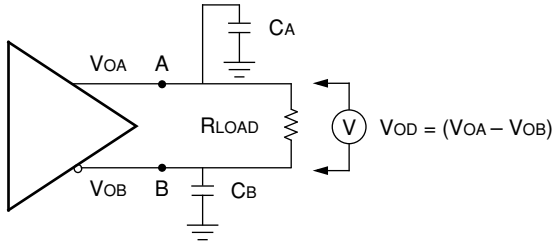


Figure 38. LVDS Driver



Pin Information

This section describes the pins and signals that perform FPGA-related functions. During configuration, the user-programmable I/Os are 3-stated and pulled up with an internal resistor. If any FPGA function pin is not used (or not bonded to package pin), it is also 3-stated and pulled up after configuration.

Table 32. FPGA Common-Function Pin Descriptions

Symbol	I/O	Description
Dedicated Pins		
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.
$\overline{\text{CFG_IRQ}}$/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.

1. 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.

Table 32. FPGA Common-Function Pin Descriptions (Continued)

Symbol	I/O	Description
Special-Purpose Pins		
M[3:0]	I	During power-up 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 in 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.*
$\overline{\text{RD}}$/$\overline{\text{MPI_STRB}}$	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.
$\overline{\text{WR}}$/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.*
PPC_A[14:31]	I	During MPI mode the PPC_A[14:31] are used as the address bus driven by the *PowerPC* bus master utilizing the least-significant bits of the *PowerPC* 32-bit address.
$\overline{\text{MPI_BURST}}$	I	$\overline{\text{MPI_BURST}}$ 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.

1. 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.

Table 32. FPGA Common-Function Pin Descriptions (Continued)

Symbol	I/O	Description
$\overline{\text{MPI_BDIP}}$	I	$\overline{\text{MPI_BDIP}}$ is driven by the *PowerPC* 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 *PowerPC* 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 *AMBA* 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 $\overline{\text{WR}}$ 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 $\overline{\text{RD}}$ 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.*
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.*
LVDS_R	—	Reference resistor connection for controlled impedance termination of Series 4 FPGA LVDS inputs.

1. 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 Embedded Core Function Pin Description (xx = AA, ..., BD)

Symbol	I/O	Description
HSI LVDS Receive Pins		
RXDxx_W_P	I	Positive LVDS work link—Channel xx
RXDxx_W_N	I	Negative LVDS work link—Channel xx
RXDxx_P_P	I	Positive LVDS protect link—Channel xx
RXDxx_P_N	I	Negative LVDS protect link—Channel xx
DAUTREC	I	Disable auto recovery for the PLL. Internal pull-down.
VDDA_STM	I	Analog VDD 1.5 V power supply for the HSI block.
VSSA_STM*	I	Analog VSS for the HSI block.
HSI LVDS Transmit Pins		
TXDxx_W_P	I	Positive LVDS work link—channel xx
TXDxx_W_N	I	Negative LVDS work link—channel xx
TXDxx_P_P	I	Positive LVDS protect link—channel xx
TXDxx_P_N	I	Negative LVDS protect link—channel xx
HSI Test Signals		
TSTCLK	I	Test clock for emulation of 622 MHz clock during PLL bypass. Internal pull-down.
MRESET	I	Test mode reset. Internal pull-down.
TESTRST	I	Resets receiver clock division counter. Internal pull-up.
RESETTX	I	Resets transmitter clock division counter. Internal pull-up.
TSTMUX[9:0]S	O	Test mode output port.
SCAN_TSTMD	I	Test mode enable. Must be tie-low for normal operation.
SCAN_EN	I	Scan test enable. Internal pull-up.
TSTSUFTLD	I	Internal pull-down.
E_TOGGLE	I	Internal pull-down.
ELSEL	I	Internal pull-down.
EXDNUP	I	Internal pull-down.
LVDS Interface Special Pins		
LVCTAP_W[4:0]	—	LVDS work input center tap (use 0.01 µF to GND).
LVCTAP_P[4:0]	—	LVDS protect input center tap (use 0.01 µF to GND).
REF10	—	LVDS reference voltage: 1.0 V ± 3%.
REF14	—	LVDS reference voltage: 1.4 V ± 3%.
RESHI	—	LVDS resistor high pin (100 Ω in series with reslo).
RESLO	—	LVDS resistor low pin (100 Ω in series with reshi).
MISC System Signals		
RST_N	I	Reset the core only. The FPGA logic is not reset by rst_n. Internal pull down allows chip to stay in reset state when external driver loses power.
SYS_CLK_P	I	Positive LVDS system clock, 50% duty cycle, also the reference clock of PLL.
SYS_CLK_N	I	Negative LVDS system clock, 50% duty cycle, also the reference clock of PLL.
LVCTAP_SK	O	LVDS center-tap for SYS_CLK (use 0.01 µf to GND).

Package Information

Table 34 summarizes the programmable I/O clock and power pins available to the ORT8850 devices.

Table 34. ORT8850 IO and Power Pin Summary

I/O or Power Type	ORT8850L	ORT8850H
User I/O Single Ended	278	297
User I/O Differential Pairs (LVDS, LVPECL)	129	129
Configuration	7	7
Dedicated Function	3	3
VDD15	48	48
VDD33	28	28
VDDIO	38	38
Vss	89	89
Single-Ended/Differential I/O per Bank		
Bank 0	64/32	68/32
Bank 1	47/20	47/20
Bank 2	ASIC I/O	ASIC I/O
Bank 3	ASIC I/O	ASIC I/O
Bank 4	ASIC I/O	ASIC I/O
Bank 5	44/18	44/18
Bank 6	76/32	76/32
Bank 7	55/27	62/27

There are some incompatibilities between the ORT8850H and ORT8850L due to the fact that the ORT8850L is a much smaller array and hence does not provide as many programmable IOs (PIOs). In order to allow pin-for-pin compatible board layouts that can accommodate either device, key compatibility issues include the following:

- **Unused Pins** Table 35 shows a list of bonded ORT8850H PIOs that are unused in the ORT8850L. As shown in the table, there are 19 balls that are not available in the ORT8850L, but are available in the ORT8850H. These user I/Os should not be used if an ORT8850L will be used.
- **Shared Control Signals on I/O Registers.** The ORCA Series 4 architecture shares clock and control signals between two adjacent I/O pads. If I/O registers are used, incompatibilities may arise between ORT8850L and ORT8850H when different clock or control signals are needed on adjacent package pins. This is because one device may allow independent clock or control signals on these adjacent pins, while the other may force them to be the same. There are two ways to avoid this issue.
  - Always keep an open bonded pin (non-bonded pins for the ORT8850L do not count) between pins that require different clock or control signals. Note that this open pin can be used to connect signals that do not require the use of I/O registers to meet timing.
  - Place and route the design in both the ORT8850H and ORT8850L to verify both produce valid designs. Note that this method guarantees the current design, but does not necessarily guard against issues that can occur when design changes are made that affect I/O registers.
  - **2X/4X I/O Shift Registers.** If 2X I/O shift registers or 4X I/O shift registers are used in the design, this may cause incompatibilities between the ORT880L and ORT8850H because only the A and C I/Os in a PIC support 2X I/O shift registers and only A I/Os supports 4X I/O shift register mode. A and C I/Os are shown in the following pinout tables under the I/O pad columns as those ending in A or C.
- **Edge Clock Input Pins.** The input buffers for fast edge clocks are only available at the C I/O pad. The C I/Os are shown in the following pinout tables under the I/O pad columns as those ending in C.

Table 35. ORT8850H Pins That Are Unused in ORT8850L

BGA Ball Bonds	ORT8850H PIOs
K4	PL11A
M5	PL13A
R5	PL20A
T5	PL21A
W4	PL27A
AA2	PL28A
Y4	PL29A
AC4	PL35A
AD5	PL37A
AG1	PL38A
AP4	PB3A
AK10	PB9A
AK11	PB10A
AM9	PB11A
AN9	PB12A
AM14	PB19A
AN14	PB20A
D11	PT12A
E13	PT11A

Users should avoid using these pins if they plan to migrate their ORT8850H design to an ORT8850L.

Package Pinouts

Table 36 provides the package pin and pin function for the ORT8850 FPSC and packages. The bond pad name is identified in the PIO nomenclature used in the *ispLEVER* 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.

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
A1	—	—	VSS	VSS	VSS	—	—
E4	—	—	VDD33	VDD33	VDD33	—	—
F5	—	—	O	PRD_DATA	PRD_DATA	RD_DATA/TDO	—
D2	—	—	I	PRESET_N	PRESET_N	RESET_N	—
E3	—	—	I	PRD_CFG_N	PRD_CFG_N	RD_CFG_N	—
G5	—	—	I	PPRGRM_N	PPRGRM_N	PRGRM_N	—
C4	0 (TL)	—	VDDIO0	VDDIO0	VDDIO0	—	—
F4	0 (TL)	7	IO	PL2D	PL2D	PLL_CK0C/HPPLL	L21C_D2
D1	0 (TL)	7	IO	PL2C	PL2C	PLL_CK0T/HPPLL	L21T_D2
A2	—	—	VSS	VSS	VSS	—	—
E2	0 (TL)	7	IO	PL2B	PL3D	—	L22C_D0
F3	0 (TL)	7	IO	PL2A	PL3C	VREF_0_07	L22T_D0
G4	0 (TL)	7	IO	PL3D	PL4D	D5	L23C_D0
H5	0 (TL)	7	IO	PL3C	PL4C	D6	L23T_D0
D3	0 (TL)	—	VDDIO0	VDDIO0	VDDIO0	—	—
E1	0 (TL)	8	IO	PL3B	PL5D	—	L24C_D0
F2	0 (TL)	8	IO	PL3A	PL5C	VREF_0_08	L24T_D0
J5	0 (TL)	8	IO	PL4D	PL6D	HDC	L25C_D1
G3	0 (TL)	8	IO	PL4C	PL6C	LDC_N	L25T_D1
A18	—	—	VSS	VSS	VSS	—	—
H4	0 (TL)	8	IO	PL4B	PL7D	—	L26C_D2
F1	0 (TL)	8	IO	PL4A	PL7C	—	L26T_D2
G2	0 (TL)	9	IO	PL5D	PL8D	TESTCFG	L27C_D0
H3	0 (TL)	9	IO	PL5C	PL8C	D7	L27T_D0
E5	0 (TL)	—	VDDIO0	VDDIO0	VDDIO0	—	—
K5	0 (TL)	9	IO	PL5B	PL9D	VREF_0_09	L28C_D0
J4	0 (TL)	9	IO	PL5A	PL9C	A17/PPC_A31	L28T_D0
G1	0 (TL)	9	IO	PL6D	PL10D	CS0_N	L29C_D3
L5	0 (TL)	9	IO	PL6C	PL10C	CS1	L29T_D3
A33	—	—	VSS	VSS	VSS	—	—
H2	0 (TL)	10	IO	PL6B	PL11D	—	L30C_D0
J3	0 (TL)	10	IO	PL6A	PL11C	—	L30T_D0
H1	0 (TL)	10	IO	PL7D	PL12D	INIT_N	L31C_D0
J2	0 (TL)	10	IO	PL7C	PL12C	DOUT	L31T_D0
K3	0 (TL)	10	IO	PL7B	PL13D	VREF_0_10	L32C_D0
L4	0 (TL)	10	IO	PL7A	PL13C	A16/PPC_A30	L32T_D0
J1	7 (CL)	1	IO	PL8D	PL14D	A15/PPC_A29	L1C_D0
K2	7 (CL)	1	IO	PL8C	PL14C	A14/PPC_A28	L1T_D0
L1	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
M4	7 (CL)	1	IO	PL8B	PL15D	—	L2C_D0
L3	7 (CL)	1	IO	PL8A	PL15C	—	L2T_D0
K1	7 (CL)	1	IO	PL9D	PL16D	VREF_7_01	L3C_D3

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
N5	7 (CL)	1	IO	PL9C	PL16C	D4	L3T_D3
AM22	—	—	VSS	VSS	VSS	—	—
L2	7 (CL)	2	IO	PL9B	PL17D	—	L4C_D1
N4	7 (CL)	2	IO	PL9A	PL17C	—	L4T_D1
P5	7 (CL)	2	IO	PL10D	PL18D	RDY/BUSY_N/RCLK	L5C_D2
M2	7 (CL)	2	IO	PL10C	PL18C	VREF_7_02	L5T_D2
M3	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
M1	7 (CL)	2	IO	PL10B	PL19D	A13/PPC_A27	L6C_D2
P4	7 (CL)	2	IO	PL10A	PL19C	A12/PPC_A26	L6T_D2
N2	7 (CL)	3	IO	PL11D	PL20D	—	L7C_D0
P3	7 (CL)	3	IO	PL11C	PL20C	—	L7T_D0
AM32	—	—	VSS	VSS	VSS	—	—
R4	7 (CL)	3	IO	PL11B	PL21D	A11/PPC_A25	L8C_D2
N1	7 (CL)	3	IO	PL11A	PL21C	VREF_7_03	L8T_D2
P2	7 (CL)	3	IO	PL12D	PL22D	—	L9C_A0
P1	7 (CL)	3	IO	PL12C	PL22C	—	L9T_A0
T4	7 (CL)	3	IO	PL12B	PL22B	—	L10C_D1
R2	7 (CL)	3	IO	PL12A	PL22A	—	L10T_D1
U5	7 (CL)	4	IO	PL13D	PL23D	RD_N/MPI_STRB_N	L11C_D3
R1	7 (CL)	4	IO	PL13C	PL23C	VREF_7_04	L11T_D3
AN1	—	—	VSS	VSS	VSS	—	—
V5	7 (CL)	4	IO	PL13B	PL23B	—	L12C_D1
T3	7 (CL)	4	IO	PL13A	PL23A	—	L12T_D1
T2	7 (CL)	4	IO	PL14D	PL24D	PLCK0C	L13C_A0
T1	7 (CL)	4	IO	PL14C	PL24C	PLCK0T	L13T_A0
R3	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
U4	7 (CL)	4	IO	PL14B	PL24B	—	L14C_A0
U3	7 (CL)	4	IO	PL14A	PL24A	—	L14T_A0
AN2	—	—	VSS	VSS	VSS	—	—
U2	7 (CL)	5	IO	PL15D	PL25D	A10/PPC_A24	L15C_A0
V2	7 (CL)	5	IO	PL15C	PL25C	A9/PPC_A23	L15T_A0
AN33	—	—	VSS	VSS	VSS	—	—
V3	7 (CL)	5	IO	PL15B	PL25B	—	L16C_A0
V4	7 (CL)	5	IO	PL15A	PL25A	—	L16T_A0
W5	7 (CL)	5	IO	PL16D	PL26D	A8/PPC_A22	L17C_A2
W2	7 (CL)	5	IO	PL16C	PL26C	VREF_7_05	L17T_A2
W3	7 (CL)	5	IO	PL16B	PL27D	—	L18C_D1
Y1	7 (CL)	5	IO	PL16A	PL27C	—	L18T_D1
Y2	7 (CL)	6	IO	PL17D	PL28D	PLCK1C	L19C_D0
AA1	7 (CL)	6	IO	PL17C	PL28C	PLCK1T	L19T_D0
AN34	—	—	VSS	VSS	VSS	—	—
Y5	7 (CL)	6	IO	PL17B	PL29D	VREF_7_06	L20C_D3

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
AB1	7 (CL)	6	IO	PL17A	PL29C	A7/PPC_A21	L20T_D3
AA5	7 (CL)	6	IO	PL18D	PL30D	A6/PPC_A20	L21C_A1
AA3	7 (CL)	6	IO	PL18C	PL30C	A5/PPC_A19	L21T_A1
U1	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
AB2	7 (CL)	7	IO	PL18B	PL31D	—	—
AA4	7 (CL)	7	IO	PL19D	PL32D	WR_N/MPI_RW	L22C_D2
AC1	7 (CL)	7	IO	PL19C	PL32C	VREF_7_07	L22T_D2
AB5	7 (CL)	7	IO	PL19B	PL33D	—	L23C_D2
AC2	7 (CL)	7	IO	PL19A	PL33C	—	L23T_D2
AB4	7 (CL)	8	IO	PL20D	PL34D	A4/PPC_A18	L23C_D0
AC5	7 (CL)	8	IO	PL20C	PL34C	VREF_7_08	L23T_D0
W1	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
AD2	7 (CL)	8	IO	PL20B	PL35D	A3/PPC_A17	L23C_D0
AE1	7 (CL)	8	IO	PL20A	PL35C	A2/PPC_A16	L23T_D0
AD3	7 (CL)	8	IO	PL21D	PL36D	A1/PPC_A15	L24C_D0
AE2	7 (CL)	8	IO	PL21C	PL36C	A0/PPC_A14	L24T_D0
AF1	7 (CL)	8	IO	PL21B	PL37D	DP0	L25C_D2
AD4	7 (CL)	8	IO	PL21A	PL37C	DP1	L25T_D2
AE3	6 (BL)	1	IO	PL22D	PL38D	D8	L1C_D0
AF2	6 (BL)	1	IO	PL22C	PL38C	VREF_6_01	L1T_D0
AB13	—	—	VSS	VSS	VSS	—	—
AE4	6 (BL)	1	IO	PL22B	PL39D	D9	L2C_D0
AF3	6 (BL)	1	IO	PL22A	PL39C	D10	L2T_D0
AE5	6 (BL)	2	IO	PL23D	PL40D	—	L3C_D1
AG2	6 (BL)	2	IO	PL23C	PL40C	VREF_6_02	L3T_D1
AK5	6 (BL)	—	VDDIO6	VDDIO6	VDDIO6	—	—
AH1	6 (BL)	2	IO	PL23B	PL41D	—	L4C_D3
AF5	6 (BL)	2	IO	PL23A	PL41C	—	L4T_D3
AF4	6 (BL)	3	IO	PL24D	PL42D	D11	L5C_D0
AG3	6 (BL)	3	IO	PL24C	PL42C	D12	L5T_D0
AB14	—	—	VSS	VSS	VSS	—	—
AH2	6 (BL)	3	IO	PL24B	PL43D	—	L6C_D0
AJ1	6 (BL)	3	IO	PL24A	PL43C	—	L6T_D0
AG4	6 (BL)	3	IO	PL25D	PL44D	VREF_6_03	L7C_A0
AG5	6 (BL)	3	IO	PL25C	PL44C	D13	L7T_A0
AL3	6 (BL)	—	VDDIO6	VDDIO6	VDDIO6	—	—
AH3	6 (BL)	4	IO	PL25B	PL44B	—	—
AK1	6 (BL)	4	IO	PL25A	PL45A	—	—
AJ2	6 (BL)	4	IO	PL26D	PL45D	—	L8C_D2
AH5	6 (BL)	4	IO	PL26C	PL45C	VREF_6_04	L8T_D2
AB15	—	—	VSS	VSS	VSS	—	—
AH4	6 (BL)	4	IO	PL26B	PL46D	—	—

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
AJ3	6 (BL)	4	IO	PL26A	PL46A	—	—
AK2	6 (BL)	4	IO	PL27D	PL47D	PLL_CK7C/HPPLL	L9C_D0
AL1	6 (BL)	4	IO	PL27C	PL47C	PLL_CK7T/HPPLL	L9T_D0
AB20	—	—	VSS	VSS	VSS	—	—
AJ5	6 (BL)	4	IO	PL27B	PL47B	—	L10C_A0
AJ4	6 (BL)	4	IO	PL27A	PL47A	—	L10T_A0
AB21	—	—	VSS	VSS	VSS	—	—
AK3	—	—	I	PTEMP	PTEMP	PTEMP	—
AM1	6 (BL)	—	VDDIO6	VDDIO6	VDDIO6	—	—
AL2	—	—	IO	LVDS_R	LVDS_R	LVDS_R	—
AK4	—	—	VDD33	VDD33	VDD33	—	—
AB22	—	—	VSS	VSS	VSS	—	—
AK6	—	—	VDD33	VDD33	VDD33	—	—
AL5	6 (BL)	5	IO	PB2A	PB2A	DP2	L11T_D1
AN4	6 (BL)	5	IO	PB2B	PB2B	—	L11C_D1
AM2	6 (BL)	—	VDDIO6	VDDIO6	VDDIO6	—	—
AM5	6 (BL)	5	IO	PB2C	PB2C	PLL_CK6T/PPLL	L12T_D1
AK7	6 (BL)	5	IO	PB2D	PB2D	PLL_CK6C/PPLL	L12C_D1
AL6	6 (BL)	5	IO	PB3A	PB3C	—	L13T_D1
AN5	6 (BL)	5	IO	PB3B	PB3D	—	L13C_D1
AM4	6 (BL)	—	VDDIO6	VDDIO6	VDDIO6	—	—
AM6	6 (BL)	5	IO	PB3C	PB4C	VREF_6_05	L14T_D0
AL7	6 (BL)	5	IO	PB3D	PB4D	DP3	L14C_D0
AK8	6 (BL)	6	IO	PB4A	PB5C	—	L15T_D3
AP5	6 (BL)	6	IO	PB4B	PB5D	—	L15C_D3
AB32	—	—	VSS	VSS	VSS	—	—
AK9	6 (BL)	6	IO	PB4C	PB6C	VREF_6_06	L16T_D2
AN6	6 (BL)	6	IO	PB4D	PB6D	D14	L16C_D2
AM7	6 (BL)	6	IO	PB5A	PB7C	—	L17T_D1
AP6	6 (BL)	6	IO	PB5B	PB7D	—	L17C_D1
AN3	6 (BL)	—	VDDIO6	VDDIO6	VDDIO6	—	—
AL8	6 (BL)	7	IO	PB5C	PB8C	D15	L18T_D1
AN7	6 (BL)	7	IO	PB5D	PB8D	D16	L18C_D1
AM8	6 (BL)	7	IO	PB6A	PB9C	D17	L19T_D0
AL9	6 (BL)	7	IO	PB6B	PB9D	D18	L19C_D0
AL4	—	—	VSS	VSS	VSS	—	—
AP7	6 (BL)	7	IO	PB6C	PB10C	VREF_6_07	L20T_D0
AN8	6 (BL)	7	IO	PB6D	PB10D	D19	L20C_D0
AL10	6 (BL)	8	IO	PB7A	PB11C	D20	L21T_D2
AP8	6 (BL)	8	IO	PB7B	PB11D	D21	L21C_D2
AL11	6 (BL)	8	IO	PB7C	PB12C	VREF_6_08	L22T_D0
AM10	6 (BL)	8	IO	PB7D	PB12D	D22	L22C_D0

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
AK12	6 (BL)	9	IO	PB8A	PB13A	—	L23T_D3
AP9	6 (BL)	9	IO	PB8B	PB13B	—	L23C_D3
AL31	—	—	VSS	VSS	VSS	—	—
AN10	6 (BL)	9	IO	PB8C	PB13C	D23	L24T_D1
AL12	6 (BL)	9	IO	PB8D	PB13D	D24	L24C_D1
AM11	6 (BL)	9	IO	PB9A	PB14A	—	L25T_D1
AP10	6 (BL)	9	IO	PB9B	PB14B	—	L25C_D1
AP3	6 (BL)	—	VDDIO6	VDDIO6	VDDIO6	—	—
AK13	6 (BL)	9	IO	PB9C	PB14C	VREF_6_09	L26T_D2
AN11	6 (BL)	9	IO	PB9D	PB14D	D25	L26C_D2
AL13	6 (BL)	9	IO	PB10A	PB15C	—	L27T_D0
AK14	6 (BL)	9	IO	PB10B	PB15D	—	L27C_D0
AM3	—	—	VSS	VSS	VSS	—	—
AN12	6 (BL)	10	IO	PB10C	PB16C	D26	L28T_D1
AL14	6 (BL)	10	IO	PB10D	PB16D	D27	L28C_D1
AP12	6 (BL)	10	IO	PB11A	PB17C	—	L29T_D0
AN13	6 (BL)	10	IO	PB11B	PB17D	—	L29C_D0
AP13	6 (BL)	10	IO	PB11C	PB18C	VREF_6_10	L30T_D3
AK15	6 (BL)	10	IO	PB11D	PB18D	D28	L30C_D3
AL15	6 (BL)	11	IO	PB12A	PB19C	D29	L31T_D0
AK16	6 (BL)	11	IO	PB12B	PB19D	D30	L31C_D0
AM13	—	—	VSS	VSS	VSS	—	—
AP14	6 (BL)	11	IO	PB12C	PB20C	VREF_6_11	L32T_D2
AL16	6 (BL)	11	IO	PB12D	PB20D	D31	L32C_D2
AN15	5 (BC)	1	IO	PB13C	PB21A	—	—
AP15	5 (BC)	1	IO	PB14A	PB21C	—	L1T_D3
AK17	5 (BC)	1	IO	PB14B	PB21D	—	L1C_D3
Y15	—	—	VSS	VSS	VSS	—	—
AM16	5 (BC)	1	IO	PB14C	PB22A	—	—
AN16	5 (BC)	1	IO	PB15A	PB22C	VREF_5_01	L2T_D1
AL17	5 (BC)	1	IO	PB15B	PB22D	—	L2C_D1
AM12	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
AP16	5 (BC)	2	IO	PB15C	PB23A	—	L3T_D1
AM17	5 (BC)	2	IO	PB15D	PB23B	—	L3C_D1
AN17	5 (BC)	2	IO	PB16A	PB23C	PBCK0T	L4T_D1
AL18	5 (BC)	2	IO	PB16B	PB23D	PBCK0C	L4C_D1
AN18	5 (BC)	2	IO	PB16C	PB24A	—	L5T_A0
AM18	5 (BC)	2	IO	PB16D	PB24B	—	L5C_A0
AN19	5 (BC)	2	IO	PB17A	PB24C	VREF_5_02	L6T_D2
AK18	5 (BC)	2	IO	PB17B	PB24D	—	L6C_D2
Y20	—	—	VSS	VSS	VSS	—	—
AM19	5 (BC)	2	IO	PB17C	PB25C	—	L7T_A0

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
AL19	5 (BC)	2	IO	PB17D	PB25D	—	L7C_A0
AP20	5 (BC)	3	IO	PB18A	PB26C	—	L8T_D3
AK19	5 (BC)	3	IO	PB18B	PB26D	VREF_5_03	L8C_D3
AM15	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
AN20	5 (BC)	3	IO	PB18C	PB27A	—	—
Y21	—	—	VSS	VSS	VSS	—	—
AP21	5 (BC)	3	IO	PB19A	PB27C	—	L9T_D2
AL20	5 (BC)	3	IO	PB19B	PB27D	—	L9C_D2
Y22	—	—	VSS	VSS	VSS	—	—
AK20	5 (BC)	3	IO	PB19C	PB28A	—	—
AN21	5 (BC)	3	IO	PB20A	PB28C	PBCK1T	L10T_A0
AM21	5 (BC)	3	IO	PB20B	PB28D	PBCK1C	L10C_A0
AM20	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
AK21	5 (BC)	3	IO	PB20C	PB29A	—	—
AP22	5 (BC)	4	IO	PB21A	PB29C	—	L11T_D2
AL21	5 (BC)	4	IO	PB21B	PB29D	—	L11C_D2
AA15	—	—	VSS	VSS	VSS	—	—
AN22	5 (BC)	4	IO	PB21C	PB30A	—	—
AP23	5 (BC)	4	IO	PB22A	PB30C	—	L12T_A0
AN23	5 (BC)	4	IO	PB22B	PB30D	VREF_5_04	L12C_A0
AA13	—	—	VSS	VSS	VSS	—	—
AK22	5 (BC)	4	IO	PB22C	PB31C	—	L13T_A0
AL22	5 (BC)	4	IO	PB22D	PB31D	—	L13C_A0
AN24	5 (BC)	5	IO	PB23C	PB32C	—	L14T_D2
AK23	5 (BC)	5	IO	PB23D	PB32D	VREF_5_05	L14C_D2
AA14	—	—	VSS	VSS	VSS	—	—
AL23	5 (BC)	5	IO	PB24C	PB33C	—	L15T_D0
AM24	5 (BC)	5	IO	PB24D	PB33D	—	L15C_D0
AP25	5 (BC)	5	IO	PB25A	PB34C	—	L16T_A0
AN25	5 (BC)	5	IO	PB25B	PB34D	—	L16T_A0
AP26	5 (BC)	6	IO	PB25C	PB35A	—	—
AK25	5 (BC)	6	IO	PB26A	PB35C	—	L17T_A0
AN26	5 (BC)	6	IO	PB26B	PB35D	VREF_5_06	L17C_A0
AP27	5 (BC)	6	IO	PB26C	PB36A	—	—
AM25	5 (BC)	6	IO	PB27A	PB36C	—	L18T_D3
AK26	5 (BC)	6	IO	PB27B	PB36D	—	L18C_D3
N32	—	—	VSS	VSS	VSS	—	—
AL24	—	—	O	TXDAA_P_N	TXDAA_P_N	—	L1N_A0
AK24	—	—	O	TXDAA_P_P	TXDAA_P_P	—	L1P_A0
A32	—	—	VDD33	VDD33	VDD33	—	—
AN27	—	—	O	TXDAB_P_N	TXDAB_P_N	—	L2N_D0
AP28	—	—	O	TXDAB_P_P	TXDAB_P_P	—	L2P_D0

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
P13	—	—	VSS	VSS	VSS	—	—
AL25	—	—	O	TXDAC_P_N	TXDAC_P_N	—	L3N_A0
AL26	—	—	O	TXDAC_P_P	TXDAC_P_P	—	L3P_A0
B32	—	—	VDD33	VDD33	VDD33	—	—
AM26	—	—	O	TXDAD_P_N	TXDAD_P_N	—	L4N_A0
AM27	—	—	O	TXDAD_P_P	TXDAD_P_P	—	L4P_A0
P14	—	—	VSS	VSS	VSS	—	—
AN28	—	—	O	Reserved	Reserved	—	L5N_D0
AP29	—	—	O	Reserved	Reserved	—	L5P_D0
C31	—	—	VDD33	VDD33	VDD33	—	—
AL27	—	—	O	TXCLK_P_N	TXCLK_P_N	—	L6N_A0
AK27	—	—	O	TXCLK_P_P	TXCLK_P_P	—	L6P_A0
P15	—	—	VSS	VSS	VSS	—	—
AL28	—	—	O	TXDBA_P_N	TXDBA_P_N	—	L7N_A0
AK28	—	—	O	TXDBA_P_P	TXDBA_P_P	—	L7P_A0
C33	—	—	VDD33	VDD33	VDD33	—	—
AM28	—	—	O	TXDBB_P_N	TXDBB_P_N	—	L8N_D0
AN29	—	—	O	TXDBB_P_P	TXDBB_P_P	—	L8P_D0
P20	—	—	VSS	VSS	VSS	—	—
AL29	—	—	O	TXDBC_P_N	TXDBC_P_N	—	L9N_A0
AK29	—	—	O	TXDBC_P_P	TXDBC_P_P	—	L9P_A0
C34	—	—	VDD33	VDD33	VDD33	—	—
AP30	—	—	O	TXDBD_P_N	TXDBD_P7_N	—	L10N_D0
AN30	—	—	O	TXDBD_P_P	TXDBD_P_P	—	L10P_D0
P21	—	—	VSS	VSS	VSS	—	—
AM29	—	—	I	DAUTREC	DAUTREC	—	—
AP31	—	—	I	TSTCLK	TSTCLK	—	—
D32	—	—	VDD33	VDD33	VDD33	—	—
AM30	—	—	I	TESTRST	TESTRST	—	—
AN31	—	—	I	TSTSHFTLD	TSTSHFTLD	—	—
P22	—	—	VSS	VSS	VSS	—	—
R13	—	—	VSS	VSS	VSS	—	—
R14	—	—	VSS	VSS	VSS	—	—
E30	—	—	VDD33	VDD33	VDD33	—	—
AL30	—	—	I	RESETTX	RESETTX	—	—
E31	—	—	VDD33	VDD33	VDD33	—	—
AH30	—	—	I	ETOGGLE	ETOGGLE	—	—
AJ30	—	—	I	ELSEL	ELSEL	—	—
R15	—	—	VSS	VSS	VSS	—	—
AL33	—	—	I	EXDNUP	EXDNUP	—	—
AH31	—	—	I	MRESET	MRESET	—	—
L34	—	—	VDD33	VDD33	VDD33	—	—

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
AK32	—	—	I	RXDAA_P_N	RXDAA_P_N	—	L11N_D0
AJ31	—	—	I	RXDAA_P_P	RXDAA_P_P	—	L11P_D0
R20	—	—	VSS	VSS	VSS	—	—
AL34	—	—	I	RXDAB_P_N	RXDAB_P_N	—	L12N_D0
AK33	—	—	I	RXDAB_P_P	RXDAB_P_P	—	L12P_D0
AJ32	—	—	I	LVCTAP_P_0	LVCTAP_P_0	—	—
M32	—	—	VDD33	VDD33	VDD33	—	—
AF30	—	—	I	RXDAC_P_N	RXDAC_P_N	—	L13N_A0
AG30	—	—	I	RXDAC_P_P	RXDAC_P_P	—	L13P_A0
R21	—	—	VSS	VSS	VSS	—	—
AG31	—	—	I	RXDAD_P_P	RXDAD_P_N	—	L14N_A0
AF31	—	—	I	RXDAD_P_P	RXDAD_P_P	—	L14P_A0
AK34	—	—	I	LVCTAP_P_1	LVCTAP_P_1	—	—
R32	—	—	VDD33	VDD33	VDD33	—	—
AJ33	—	—	I	Reserved	Reserved	—	L15N_A0
AH32	—	—	I	Reserved	Reserved	—	L15P_A0
R22	—	—	VSS	VSS	VSS	—	—
AJ34	—	—	I	Reserved	Reserved	—	L16N_D0
AH33	—	—	I	Reserved	Reserved	—	L16P_D0
AD30	—	—	I	LVCTAP_P_2	LVCTAP_P_2	—	—
U34	—	—	VDD33	VDD33	VDD33	—	—
AG32	—	—	I	RXDBA_P_N	RXDBA_P_N	—	L17N_A0
AG33	—	—	I	RXDBA_P_P	RXDBA_P_P	—	L17P_A0
T16	—	—	VSS	VSS	VSS	—	—
AH34	—	—	I	LVCTAP_P_3	LVCTAP_P_3	—	—
AE30	—	—	I	RXDBB_P_N	RXDBB_P_N	—	L18N_A0
AE31	—	—	I	RXDBB_P_P	RXDBB_P_P	—	L18P_A0
W34	—	—	VDD33	VDD33	VDD33	—	—
AF32	—	—	I	RXDBC_P_N	RXDBC_P_N	—	L19N_A0
AF33	—	—	I	RXDBC_P_P	RXDBC_P_P	—	L19P_A0
T17	—	—	VSS	VSS	VSS	—	—
AC30	—	—	I	LVCTAP_P_4	LVCTAP_P_4	—	—
AG34	—	—	I	RXDBD_P_N	RXDBD_P_N	—	L20N_A0
AF34	—	—	I	RXDBD_P_P	RXDBD_P_P	—	L20P_A0
Y32	—	—	VDD33	VDD33	VDD33	—	—
AB30	—	—	VDDA_STM	VDDA_STM	VDDA_STM	—	—
AD31	—	—	VSSA_STM	VSSA_STM	VSSA_STM	—	—
T18	—	—	VSS	VSS	VSS	—	—
AE32	—	—	I	SYS_CLK_N	SYS_CLK_N	—	L21N_D0
AE33	—	—	I	SYS_CLK_P	SYS_CLK_P	—	L21P_D0
AC32	—	—	VDD33	VDD33	VDD33	—	—
AE34	—	—	O	LVCTAP_SK	LVCTAP_SK	—	—

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
T19	—	—	VSS	VSS	VSS	—	—
AC31	—	—	I	RXDAA_W_N	RXDAA_W_N	—	L22N_A0
AB31	—	—	I	RXDAA_W_P	RXDAA_W_P	—	L22P_A0
T34	—	—	VSS	VSS	VSS	—	—
AD32	—	—	I	RXDAB_W_N	RXDAB_W_N	—	L23N_A0
AD33	—	—	I	RXDAB_W_P	RXDAB_W_P	—	L23P_A0
AA30	—	—	I	LVCTAP_W_0	LVCTAP_W_0	—	—
AD34	—	—	VDD33	VDD33	VDD33	—	—
AC33	—	—	I	RXDAC_W_N	RXDAC_W_N	—	L24N_A0
AC34	—	—	I	RXDAC_W_P	RXDAC_W_P	—	L24P_A0
U16	—	—	VSS	VSS	VSS	—	—
AB33	—	—	I	RXDAD_W_N	RXDAD_W_N	—	L25N_A0
AB34	—	—	I	RXDAD_W_P	RXDAD_W_P	—	L25P_A0
Y30	—	—	I	LVCTAP_W_1	LVCTAP_W_1	—	—
AK30	—	—	VDD33	VDD33	VDD33	—	—
AA31	—	—	I	Reserved	Reserved	—	L26N_A0
AA32	—	—	I	Reserved	Reserved	—	L26P_A0
U17	—	—	VSS	VSS	VSS	—	—
W30	—	—	I	Reserved	Reserved	—	L27N_D0
Y31	—	—	I	Reserved	Reserved	—	L27P_D0
AA33	—	—	I	LVCTAP_W_2	LVCTAP_W_2	—	—
AK31	—	—	VDD33	VDD33	VDD33	—	—
AA34	—	—	I	RXDBA_W_N	RXDBA_W_N	—	L28N_A0
Y34	—	—	I	RXDBA_W_P	RXDBA_W_P	—	L28P_A0
U18	—	—	VSS	VSS	VSS	—	—
Y33	—	—	I	LVCTAP_W_3	LVCTAP_W_3	—	—
W31	—	—	I	RXDBB_W_N	RXDBB_W_N	—	L29N_A0
W32	—	—	I	RXDBB_W_P	RXDBB_W_P	—	L29P_A0
AL32	—	—	VDD33	VDD33	VDD33	—	—
V30	—	—	I	RXDBC_W_N	RXDBC_W_N	—	L30N_A0
V31	—	—	I	RXDBC_W_P	RXDBC_W_P	—	L30P_A0
U19	—	—	VSS	VSS	VSS	—	—
W33	—	—	I	LVCTAP_W_4	LVCTAP_W_4	—	—
V32	—	—	I	RXDBD_W_N	RXDBD_W_N	—	L31N_A0
V33	—	—	I	RXDBD_W_P	RXDBD_W_P	—	L31P_A0
V1	—	—	VSS	VSS	VSS	—	—
U33	—	—	I	RESLO	RESLO	—	—
U32	—	—	I	RESHI	RESHI	—	—
U31	—	—	I	REF14	REF14	—	—
T33	—	—	I	REF10	REF10	—	—
AM31	—	—	VDD33	VDD33	VDD33	—	—
V16	—	—	VSS	VSS	VSS	—	—

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
T32	—	—	O	TXDAA_W_N	TXDAA_W_N	—	L32N_D1
R34	—	—	O	TXDAA_W_P	TXDAA_W_P	—	L32P_D1
AM33	—	—	VDD33	VDD33	VDD33	—	—
U30	—	—	O	TXDAB_W_N	TXDAB_W_N	—	L33N_D0
T31	—	—	O	TXDAB_W_P	TXDAB_W_P	—	L33P_D0
V17	—	—	VSS	VSS	VSS	—	—
R33	—	—	O	TXDAC_W_N	TXDAC_W_N	—	L34N_D0
P34	—	—	O	TXDAC_W_P	TXDAC_W_P	—	L34P_D0
AM34	—	—	VDD33	VDD33	VDD33	—	—
P33	—	—	O	TXDAD_W_N	TXDAD_W_N	—	L35N_D0
N34	—	—	O	TXDAD_W_P	TXDAD_W_P	—	L35P_D0
V18	—	—	VSS	VSS	VSS	—	—
T30	—	—	O	Reserved	Reserved	—	L36N_D0
R31	—	—	O	Reserved	Reserved	—	L36P_D0
AN32	—	—	VDD33	VDD33	VDD33	—	—
P32	—	—	O	Reserved	Reserved	—	L37N_D1
R30	—	—	O	Reserved	Reserved	—	L37P_D1
V19	—	—	VSS	VSS	VSS	—	—
N33	—	—	O	TXDBA_W_N	TXDBA_W_N	—	L38N_D0
M34	—	—	O	TXDBA_W_P	TXDBA_W_P	—	L38P_D0
AP32	—	—	VDD33	VDD33	VDD33	—	—
P31	—	—	O	TXDBB_W_N	TXDBB_W_N	—	L39N_D1
M33	—	—	O	TXDBB_W_P	TXDBB_W_P	—	L39P_D1
V34	—	—	VSS	VSS	VSS	—	—
N31	—	—	O	TXDBC_W_N	TXDBC_W_N	—	L40N_D0
P30	—	—	O	TXDBC_W_P	TXDBC_W_P	—	L40P_D0
L33	—	—	O	TXDBD_W_N	TXDBD_W_N	—	L41N_D0
K34	—	—	O	TXDBD_W_P	TXDBD_W_P	—	L41P_D0
W16	—	—	VSS	VSS	VSS	—	—
M31	—	—	I	Reserved	Reserved	—	L42N_D0
L32	—	—	I	Reserved	Reserved	—	L42P_D0
K33	—	—	I	Reserved	Reserved	—	—
W17	—	—	VSS	VSS	VSS	—	—
N30	—	—	VDDA_PDI	Reserved	Reserved	—	—
L30	—	—	VSSA_PDI	Reserved	Reserved	—	—
W18	—	—	VSS	VSS	VSS	—	—
M30	—	—	I	Reserved	Reserved	—	L43N_D0
L31	—	—	I	Reserved	Reserved	—	L43P_D0
W19	—	—	VSS	VSS	VSS	—	—
J34	—	—	I	Reserved	Reserved	—	L44N_D1
K32	—	—	I	Reserved	Reserved	—	L44P_D1
J33	—	—	I	Reserved	Reserved	—	—

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
H34	—	—	I	Reserved	Reserved	—	L45N_D1
J32	—	—	I	Reserved	Reserved	—	L45P_D1
Y13	—	—	VSS	VSS	VSS	—	—
K31	—	—	I	Reserved	Reserved	—	L46N_A0
K30	—	—	I	Reserved	Reserved	—	L46P_A0
H33	—	—	I	Reserved	Reserved	—	—
J31	—	—	I	Reserved	Reserved	—	L47N_A0
J30	—	—	I	Reserved	Reserved	—	L47P_A0
Y14	—	—	VSS	VSS	VSS	—	—
G34	—	—	I	Reserved	Reserved	—	L48N_D1
H32	—	—	I	Reserved	Reserved	—	L48P_D1
H31	—	—	I	Reserved	Reserved	—	—
G33	—	—	I	Reserved	Reserved	—	L49N_D0
F34	—	—	I	Reserved	Reserved	—	L49P_D0
H30	—	—	I	Reserved	Reserved	—	—
G32	—	—	I	Reserved	Reserved	—	L50N_D0
F33	—	—	I	Reserved	Reserved	—	L50P_D0
G30	—	—	I	Reserved	Reserved	—	L51N_A0
G31	—	—	I	Reserved	Reserved	—	L51P_A0
E34	—	—	I	Reserved	Reserved	—	—
F32	—	—	I	Reserved	Reserved	—	L52N_A0
E33	—	—	I	Reserved	Reserved	—	L52P_A0
F31	—	—	O	TSTMUX0S	TSTMUX0S	—	—
E32	—	—	O	TSTMUX1S	TSTMUX1S	—	—
D34	—	—	O	TSTMUX2S	TSTMUX2S	—	—
D33	—	—	O	TSTMUX3S	TSTMUX3S	—	—
F30	—	—	O	TSTMUX4S	TSTMUX4S	—	—
D30	—	—	O	TSTMUX5S	TSTMUX5S	—	—
E29	—	—	O	TSTMUX6S	TSTMUX6S	—	—
C30	—	—	O	TSTMUX7S	TSTMUX7S	—	—
B31	—	—	O	TSTMUX8S	TSTMUX8S	—	—
D29	—	—	O	TSTMUX9S	TSTMUX9S	—	—
B30	—	—	I	SCANEN	SCANEN	—	—
A31	—	—	I	SCAN_TSTMD	SCAN_TSTMD	—	—
B29	—	—	I	RST_N	RST_N	—	—
E28	—	—	O	Reserved	Reserved	—	L53N_D1
C29	—	—	O	Reserved	Reserved	—	L53P_D1
D28	—	—	O	Reserved	Reserved	—	L54N_D0
E27	—	—	O	Reserved	Reserved	—	L54P_D0
A30	—	—	O	Reserved	Reserved	—	L55N_D1
C28	—	—	O	Reserved	Reserved	—	L55P_D1
B28	—	—	O	Reserved	Reserved	—	L56N_D0

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
A29	—	—	O	Reserved	Reserved	—	L56P_D0
D27	—	—	O	Reserved	Reserved	—	L57N_D0
E26	—	—	O	Reserved	Reserved	—	L57P_D0
C27	—	—	O	Reserved	Reserved	—	L58N_D0
D26	—	—	O	Reserved	Reserved	—	L58P_D0
A28	—	—	O	Reserved	Reserved	—	L59N_D0
B27	—	—	O	Reserved	Reserved	—	L59P_D0
C26	—	—	O	Reserved	Reserved	—	L60N_D0
D25	—	—	O	Reserved	Reserved	—	L60P_D0
A27	—	—	O	Reserved	Reserved	—	L61N_D0
B26	—	—	O	Reserved	Reserved	—	L61P_D0
D24	—	—	O	Reserved	Reserved	—	L62N_D0
C25	—	—	O	Reserved	Reserved	—	L62P_D0
C22	—	—	VSS	VSS	VSS	—	—
A26	1 (TC)	1	IO	PT26D	PT35D	—	L1C_D3
E25	1 (TC)	1	IO	PT26C	PT35C	—	L1T_D3
A25	1 (TC)	1	IO	PT26B	PT35B	—	L2C_A0
B25	1 (TC)	1	IO	PT26A	PT35A	—	L2T_A0
C24	1 (TC)	1	IO	PT25D	PT34D	VREF_1_01	L3C_D0
D23	1 (TC)	1	IO	PT25C	PT34C	—	L3T_D0
C32	—	—	VSS	VSS	VSS	—	—
B24	1 (TC)	1	IO	PT25B	PT33D	—	L4C_A2
E24	1 (TC)	1	IO	PT25A	PT33C	—	L4T_A2
D22	1 (TC)	2	IO	PT24D	PT32D	—	L5C_D1
B23	1 (TC)	2	IO	PT24C	PT32C	VREF_1_02	L5T_D1
E23	1 (TC)	2	IO	PT24B	PT31D	—	L6C_A3
A23	1 (TC)	2	IO	PT24A	PT31C	—	L6T_A3
D21	1 (TC)	2	IO	PT23D	PT30D	—	L7C_D1
B22	1 (TC)	2	IO	PT23C	PT30C	—	L7T_D1
D4	—	—	VSS	VSS	VSS	—	—
A22	1 (TC)	3	IO	PT22D	PT29D	—	L8C_D1
C21	1 (TC)	3	IO	PT22C	PT29C	VREF_1_03	L8T_D1
E22	1 (TC)	3	IO	PT22A	PT29A	—	—
D20	1 (TC)	3	IO	PT21D	PT28D	—	L9C_D1
B21	1 (TC)	3	IO	PT21C	PT28C	—	L9T_D1
D31	—	—	VSS	VSS	VSS	—	—
E21	1 (TC)	3	IO	PT21A	PT28A	—	—
A21	1 (TC)	3	IO	PT20D	PT27D	—	L10C_D0
B20	1 (TC)	3	IO	PT20C	PT27C	—	L10T_D0
A11	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
A20	1 (TC)	3	IO	PT20A	PT27A	—	—
E20	1 (TC)	4	IO	PT19D	PT26D	—	L11C_D0

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
D19	1 (TC)	4	IO	PT19C	PT26C	—	L11T_D0
C19	1 (TC)	4	IO	PT19B	PT25D	—	L12C_A0
B19	1 (TC)	4	IO	PT19A	PT25C	—	L12T_A0
N3	—	—	VSS	VSS	VSS	—	—
E19	1 (TC)	4	IO	PT18D	PT24D	—	L13C_D0
D18	1 (TC)	4	IO	PT18C	PT24C	VREF_1_04	L13T_D0
A17	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
B18	1 (TC)	4	IO	PT18B	PT24B	—	L14C_A0
C18	1 (TC)	4	IO	PT18A	PT24A	—	L14T_A0
B17	1 (TC)	5	IO	PT17D	PT23D	PTCK1C	L15C_A0
C17	1 (TC)	5	IO	PT17C	PT23C	PTCK1T	L15T_A0
N13	—	—	VSS	VSS	VSS	—	—
A16	1 (TC)	5	IO	PT17B	PT23B	—	L16C_D2
D17	1 (TC)	5	IO	PT17A	PT23A	—	L16T_D2
B16	1 (TC)	5	IO	PT16D	PT22D	PTCK0C	L17C_A0
C16	1 (TC)	5	IO	PT16C	PT22C	PTCK0T	L17T_A0
D16	1 (TC)	5	IO	PT16A	PT22A	—	—
E18	1 (TC)	5	IO	PT15D	PT21D	VREF_1_05	L18C_D3
A15	1 (TC)	5	IO	PT15C	PT21C	—	L18T_D3
A19	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
B15	1 (TC)	5	IO	PT15A	PT21A	—	—
D15	1 (TC)	6	IO	PT14D	PT20D	—	L19C_D2
A14	1 (TC)	6	IO	PT14C	PT20C	—	L19T_D2
N14	—	—	VSS	VSS	VSS	—	—
B14	1 (TC)	6	IO	PT14A	PT20A	—	—
E17	1 (TC)	6	IO	PT13D	PT19D	—	L20C_D2
C14	1 (TC)	6	IO	PT13C	PT19C	VREF_1_06	L20T_D2
D14	1 (TC)	6	IO	PT13A	PT19A	—	—
N15	—	—	VSS	VSS	VSS	—	—
E16	0 (TL)	1	IO	PT11D	PT18D	MPI_RTRY_N	L1C_D3
A13	0 (TL)	1	IO	PT11C	PT18C	MPI_ACK_N	L1T_D3
B13	0 (TL)	1	IO	PT11B	PT17D	—	L2C_D0
A12	0 (TL)	1	IO	PT11A	PT17C	VREF_0_01	L2T_D0
B12	0 (TL)	1	IO	PT10D	PT16D	M0	L3C_D1
D13	0 (TL)	1	IO	PT10C	PT16C	M1	L3T_D1
A34	—	—	VSS	VSS	VSS	—	—
E15	0 (TL)	2	IO	PT10B	PT15D	MPI_CLK	L4C_D3
B11	0 (TL)	2	IO	PT10A	PT15C	A21/MPI_BURST_N	L4T_D3
A10	0 (TL)	2	IO	PT9D	PT14D	M2	L5C_D3
E14	0 (TL)	2	IO	PT9C	PT14C	M3	L5T_D3
A3	0 (TL)	—	VDDIO0	VDDIO0	VDDIO0	—	—
D12	0 (TL)	2	IO	PT9B	PT13D	VREF_0_02	L6C_D0

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
C11	0 (TL)	2	IO	PT9A	PT13C	MPI_TEA_N	L6T_D0
B10	0 (TL)	3	IO	PT8D	PT12D	—	L7C_D0
A9	0 (TL)	3	IO	PT8C	PT12C	—	L7T_D0
C10	0 (TL)	3	IO	PT8B	PT11D	VREF_0_03	L8C_D0
B9	0 (TL)	3	IO	PT8A	PT11C	—	L8T_D0
A8	0 (TL)	3	IO	PT7D	PT10D	D0	L9C_D2
D10	0 (TL)	3	IO	PT7C	PT10C	TMS	L9T_D2
B1	—	—	VSS	VSS	VSS	—	—
C9	0 (TL)	4	IO	PT7B	PT9D	A20/MPI_BDIP_N	L10C_D0
B8	0 (TL)	4	IO	PT7A	PT9C	A19/MPI_TSZ1	L10T_D0
A7	0 (TL)	4	IO	PT6D	PT8D	A18/MPI_TSZ0	L11C_D4
E12	0 (TL)	4	IO	PT6C	PT8C	D3	L11T_D4
B3	0 (TL)	—	VDDIO0	VDDIO0	VDDIO0	—	—
D9	0 (TL)	4	IO	PT6B	PT7D	VREF_0_04	L12C_D0
C8	0 (TL)	4	IO	PT6A	PT7C	—	L12T_D0
E11	0 (TL)	5	IO	PT5D	PT6D	D1	L13C_D3
B7	0 (TL)	5	IO	PT5C	PT6C	D2	L13T_D3
B2	—	—	VSS	VSS	VSS	—	—
A6	0 (TL)	5	IO	PT5B	PT5D	—	L14C_D2
D8	0 (TL)	5	IO	PT5A	PT5C	VREF_0_05	L14T_D2
C7	0 (TL)	5	IO	PT4D	PT4D	TDI	L15C_D1
A5	0 (TL)	5	IO	PT4C	PT4C	TCK	L15T_D1
C1	0 (TL)	—	VDDIO0	VDDIO0	VDDIO0	—	—
E10	0 (TL)	5	IO	PT4B	PT4B	—	L16C_D2
D7	0 (TL)	5	IO	PT4A	PT4A	—	L16T_D2
A4	0 (TL)	6	IO	PT3D	PT3D	—	L17C_D4
E9	0 (TL)	6	IO	PT3C	PT3C	VREF_0_06	L17T_D4
B33	—	—	VSS	VSS	VSS	—	—
B6	0 (TL)	6	IO	PT3B	PT3B	—	L18C_A0
C6	0 (TL)	6	IO	PT3A	PT3A	—	L18T_A0
B5	0 (TL)	6	IO	PT2D	PT2D	PLL_CK1C/PPLL	L19C_D1
D6	0 (TL)	6	IO	PT2C	PT2C	PLL_CK1T/PPLL	L19T_D1
C2	0 (TL)	—	VDDIO0	VDDIO0	VDDIO0	—	—
C5	0 (TL)	6	IO	PT2B	PT2B	—	L20C_D0
B4	0 (TL)	6	IO	PT2A	PT2A	—	L20T_D0
E8	—	—	O	PCFG_MPI_IRQ	PCFG_MPI_IRQ	CFG_IRQ_N/MPI_IRQ_N	—

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
E7	—	—	IO	PCCLK	PCCLK	CCLK	—
D5	—	—	IO	PDONE	PDONE	DONE	—
E6	—	—	VDD33	VDD33	VDD33	—	—
B34	—	—	VSS	VSS	VSS	—	—
A24	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
AM23	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
AP1	—	—	VSS	VSS	VSS	—	—
K4	0 (TL)	10	IO	UNUSED	PL11A	—	—
M5	0 (TL)	10	IO	UNUSED	PL13A	—	—
R5	7 (CL)	3	IO	UNUSED	PL20A	—	—
T5	7 (CL)	3	IO	UNUSED	PL21A	—	—
W4	7 (CL)	5	IO	UNUSED	PL27A	—	—
AA2	7 (CL)	6	IO	UNUSED	PL28A	—	—
Y4	7 (CL)	6	IO	UNUSED	PL29A	—	—
AC4	7 (CL)	8	IO	UNUSED	PL35A	—	—
AD5	7 (CL)	8	IO	UNUSED	PL37A	—	—
AG1	6 (BL)	1	IO	UNUSED	PL38A	—	—
AK10	6 (BL)	7	IO	UNUSED	PB9A	—	—
AK11	6 (BL)	7	IO	UNUSED	PB10A	—	—
AM9	6 (BL)	8	IO	UNUSED	PB11A	—	—
AN9	6 (BL)	8	IO	UNUSED	PB12A	—	—
AM14	6 (BL)	11	IO	UNUSED	PB19A	—	—
AN14	6 (BL)	11	IO	UNUSED	PB20A	—	—
D11	0 (TL)	3	IO	UNUSED	PT12A	—	—
E13	0 (TL)	3	IO	UNUSED	PT11A	—	—
AP4	6 (BL)	5	IO	UNUSED	PB3A	—	—
Y3	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
AC3	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
AD1	7 (CL)	—	VDDIO7	VDDIO7	VDDIO7	—	—
AP11	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
AP17	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
AP19	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
AP24	5 (BC)	—	VDDIO5	VDDIO5	VDDIO5	—	—
C12	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
C15	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
C20	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
C23	1 (TC)	—	VDDIO1	VDDIO1	VDDIO1	—	—
W22	—	—	VDD15	VDD15	VDD15	—	—
Y16	—	—	VDD15	VDD15	VDD15	—	—
V22	—	—	VDD15	VDD15	VDD15	—	—
U22	—	—	VDD15	VDD15	VDD15	—	—
T22	—	—	VDD15	VDD15	VDD15	—	—

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
P17	—	—	VDD15	VDD15	VDD15	—	—
P18	—	—	VDD15	VDD15	VDD15	—	—
N16	—	—	VDD15	VDD15	VDD15	—	—
N17	—	—	VDD15	VDD15	VDD15	—	—
N18	—	—	VDD15	VDD15	VDD15	—	—
N19	—	—	VDD15	VDD15	VDD15	—	—
P16	—	—	VDD15	VDD15	VDD15	—	—
P19	—	—	VDD15	VDD15	VDD15	—	—
R16	—	—	VDD15	VDD15	VDD15	—	—
R17	—	—	VDD15	VDD15	VDD15	—	—
R18	—	—	VDD15	VDD15	VDD15	—	—
R19	—	—	VDD15	VDD15	VDD15	—	—
T13	—	—	VDD15	VDD15	VDD15	—	—
T14	—	—	VDD15	VDD15	VDD15	—	—
T15	—	—	VDD15	VDD15	VDD15	—	—
T20	—	—	VDD15	VDD15	VDD15	—	—
T21	—	—	VDD15	VDD15	VDD15	—	—
U13	—	—	VDD15	VDD15	VDD15	—	—
U14	—	—	VDD15	VDD15	VDD15	—	—
U15	—	—	VDD15	VDD15	VDD15	—	—
U20	—	—	VDD15	VDD15	VDD15	—	—
U21	—	—	VDD15	VDD15	VDD15	—	—
V13	—	—	VDD15	VDD15	VDD15	—	—
V14	—	—	VDD15	VDD15	VDD15	—	—
V15	—	—	VDD15	VDD15	VDD15	—	—
V20	—	—	VDD15	VDD15	VDD15	—	—
V21	—	—	VDD15	VDD15	VDD15	—	—
W13	—	—	VDD15	VDD15	VDD15	—	—
W14	—	—	VDD15	VDD15	VDD15	—	—
W15	—	—	VDD15	VDD15	VDD15	—	—
W20	—	—	VDD15	VDD15	VDD15	—	—
W21	—	—	VDD15	VDD15	VDD15	—	—
Y17	—	—	VDD15	VDD15	VDD15	—	—
Y18	—	—	VDD15	VDD15	VDD15	—	—
Y19	—	—	VDD15	VDD15	VDD15	—	—
AA16	—	—	VDD15	VDD15	VDD15	—	—
AA17	—	—	VDD15	VDD15	VDD15	—	—
AA18	—	—	VDD15	VDD15	VDD15	—	—
AA19	—	—	VDD15	VDD15	VDD15	—	—
AB16	—	—	VDD15	VDD15	VDD15	—	—
AB17	—	—	VDD15	VDD15	VDD15	—	—
AB18	—	—	VDD15	VDD15	VDD15	—	—

Table 36. ORT8850L and ORT8850H 680-Pin PBGAM Pinout (note: pins labeled "Reserved" should be left unconnected) (Continued)

BM680	VDDIO Bank	VREF Group	I/O	ORT8850L	ORT8850H	Additional Function	Pair
C3	—	—	VSS	VSS	VSS	—	—
C13	—	—	VSS	VSS	VSS	—	—
AP2	—	—	VSS	VSS	VSS	—	—
AP18	—	—	VSS	VSS	VSS	—	—
AP33	—	—	VSS	VSS	VSS	—	—
AP34	—	—	VSS	VSS	VSS	—	—
AA20	—	—	VSS	VSS	VSS	—	—
AA21	—	—	VSS	VSS	VSS	—	—
AA22	—	—	VSS	VSS	VSS	—	—
N21	—	—	VSS	VSS	VSS	—	—
N22	—	—	VSS	VSS	VSS	—	—
AB3	—	—	VSS	VSS	VSS	—	—
AB19	—	—	VDD15	VDD15	VDD15	—	—
N20	—	—	VSS	VSS	VSS	—	—

Note: Pins labeled "reserved" should be left unconnected.

Package Thermal Characteristics Summary

There are three thermal parameters that are in common use: θ_{JA}, ψ_{JC}, and θ_{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.

θJA

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

$$\theta_{JA} = \frac{T_J - T_A}{Q} \quad (1)$$

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

Experimentally, θJA 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 θJA is expressed in units of °C/watt.

ΨJC

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:

$$\Psi_{JC} = \frac{T_J - T_C}{Q} \quad (2)$$

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 θJA 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. ψJC is also expressed in units of °C/W.

θJC

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:

$$\theta_{JC} = \frac{T_J - T_C}{Q} \quad (3)$$

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 θJC from ψJC. θJC is a true thermal resistance and is expressed in units of °C/W.

θJB

This is the thermal resistance from junction to board (ΘJL). It is defined by:

$$\theta_{JB} = \frac{T_J - T_B}{Q} \quad (4)$$

where T_B 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 θJB 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, 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_{Amax}, and the power dissipated by the device, Q (expressed in °C), the maximum junction temperature is approximated by:

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

Table 37 lists the thermal characteristics for the package used with the *ORCA* ORT8850 Series of FPSCs.

Package Thermal Characteristics

Table 37. ORCA ORT8850 Plastic Package Thermal Guidelines

Package	ΘJA (°C/W)			Maximum Power (W)
	0 fpm	200 fpm	500 fpm	T = 70 °C Max, TJ = 125 °C Max, 0 fpm
680-Pin PBGAM*	13.4	11.5	10.5	4.10

* The 680-Pin PBGAM package includes 2 oz. copper plates.

Heat Sink Information

The estimated worst-case power requirements for the ORT8850 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 38. Heat Sink Vendors

Vendor	Location	Phone
Aavid Thermalloy	Concord, NH	(603) 224-9988
Chip Coolers (Tyco Electronics)	Harrisburg, PA	(800) 468-2023
IERC (CTS Corp.)	Burbank, CA	(818) 842-7277
R-Theta	Buffalo, NY	(800) 388-5428
Sanyo Denki	Torrance, CA	(310) 783-5400
Wakefield Thermal Solutions	Pelham, NH	(603) 635-2800

Package Coplanarity

The coplanarity limits of the Lattice packages are as follows:

- PBGAM: 8.0 mils

Package Parasitics

The electrical performance of an IC package, such as signal quality and noise sensitivity, is directly affected by the package parasitics. Table 39 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.

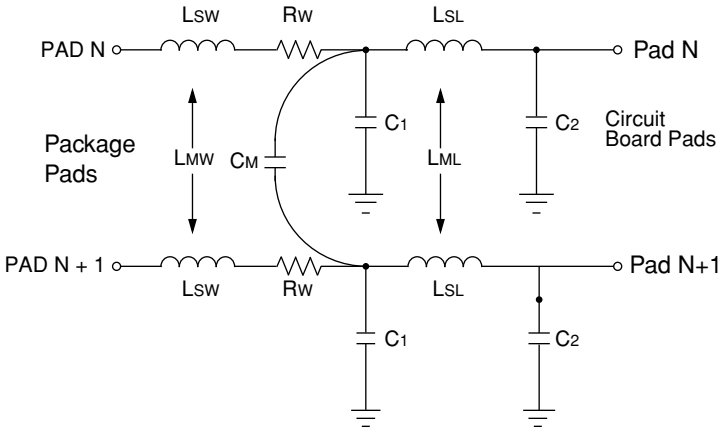
Four inductances in nH are listed: L_{SW} and L_{SL}, the self-inductance of the lead; and L_{MW} and L_{ML}, the mutual inductance to the nearest neighbor lead. These parameters are important in determining ground bounce noise and inductive crosstalk noise. Three capacitances in pF are listed: C_M, the mutual capacitance of the lead to the nearest neighbor lead; and C_1 and C_2, 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 39 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_1 and C_2 capacitors.

Table 39. ORCA ORT8850 Package Parasitics

Package Type	L_{SW}	L_{MW}	R_W	C_1	C_2	C_M	L_{SL}	L_{ML}
680-Pin PBGAM	3.8	1.3	250	1.0	1.0	0.3	2.8 - 5.0	0.5 - 1.0

Figure 39. 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.

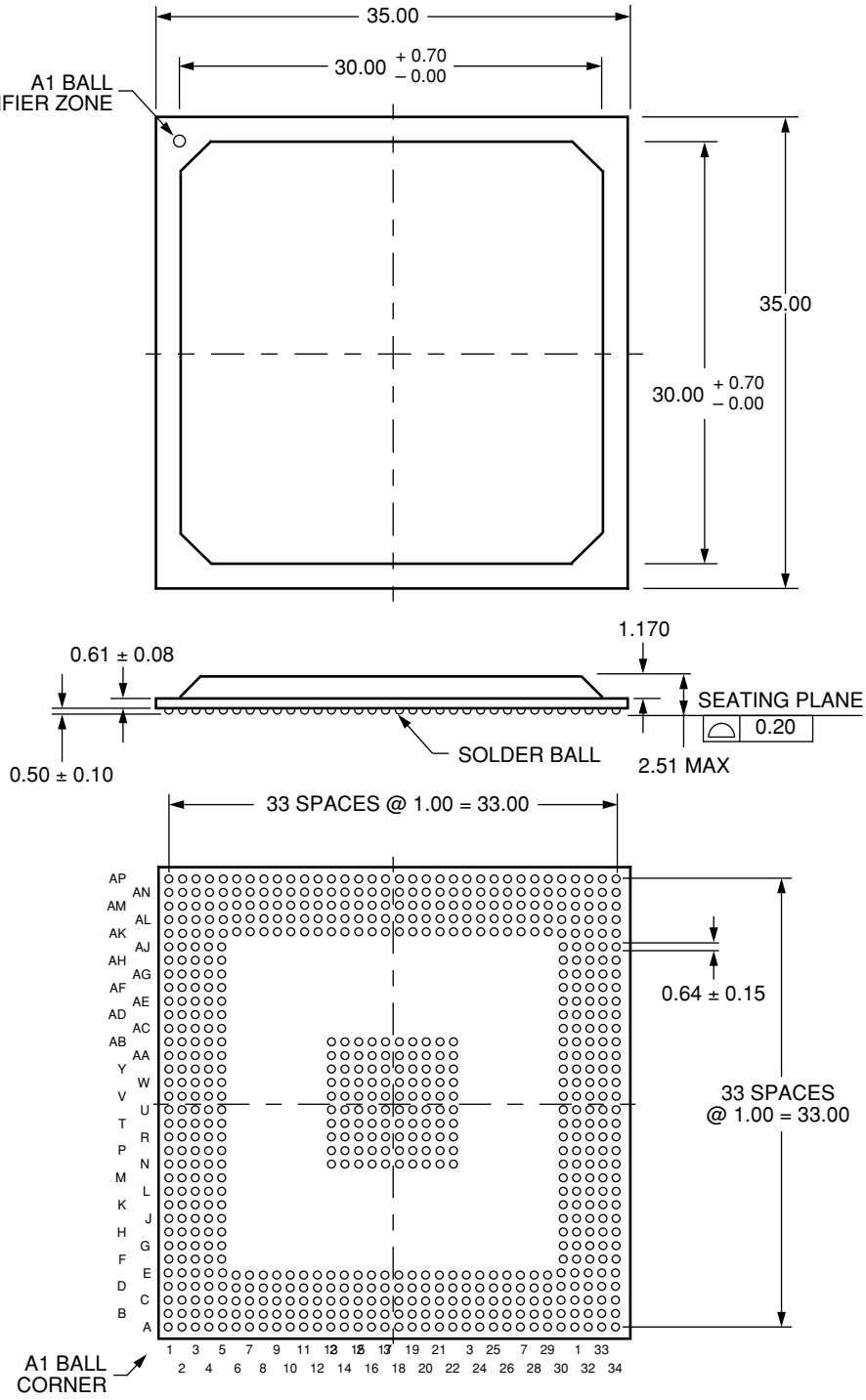
Reference (REF): The reference dimension is an untoleranced dimension used for informational purposes only. It is a repeated dimension or one that can be derived from other values in the drawing.

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

Package Outline Drawings

Figure 40. 680-Pin PBGAM Outline Drawings

Dimensions are in millimeters.



Ordering Information

Figure 41. Part Number Description

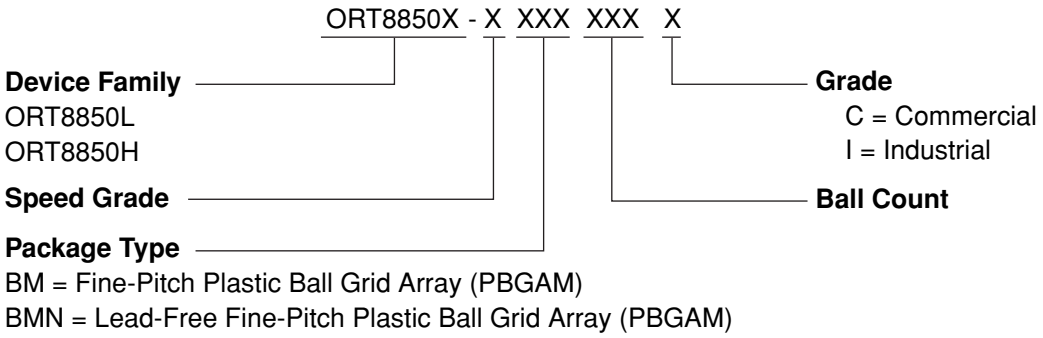


Table 40. Device Type Options

Device	Voltage
ORT8850L	1.5 V internal 3.3 V/2.5 V/1.8 V/1.5 V I/O
ORT8850H	1.5 V internal 3.3 V/2.5 V/1.8 V/1.5 V I/O

Table 41. Temperature Range

Symbol	Description	Ambient Temperature	Junction Temperature
C	Commercial	0 ˚C to +70 ˚C	0 ˚C to +85 ˚C
I	Industrial	–40 ˚C to +85 ˚C	–40 ˚C to +100 ˚C

Table 42. Conventional Packaging – Commercial Ordering Information[1]

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade
ORT8850L	ORT8850L-3BM680C	3	PBGAM (fpBGA)	680	C
	ORT8850L-2BM680C	2	PBGAM (fpBGA)	680	C
	ORT8850L-1BM680C	1	PBGAM (fpBGA)	680	C
ORT8850H	ORT8850H-2BM680C	2	PBGAM (fpBGA)	680	C
	ORT8850H-1BM680C	1	PBGAM (fpBGA)	680	C

Table 43. Conventional Packaging – Industrial Ordering Information[1]

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade
ORT8850L	ORT8850L-2BM680I	2	PBGAM (fpBGA)	680	I
	ORT8850L-1BM680I	1	PBGAM (fpBGA)	680	I
ORT8850H	ORT8850H-1BM680I	1	PBGAM (fpBGA)	680	I

1.For all but the slowest commercial speed grade, the speed grades on these devices are dual marked. For example, the commercial speed grade -2XXXXXC is also marked with the industrial grade -1XXXXXI. The commercial grade is always one speed grade faster than the associated dual mark industrial grade. The slowest commercial speed grade is marked as commercial grade only.

Table 44. Lead-Free Packaging – Commercial Ordering Information[1]

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade
ORT8850L	ORT8850L-3BMN680C	3	Lead-Free PBGAM (fpBGA)	680	C
	ORT8850L-2BMN680C	2	Lead-Free PBGAM (fpBGA)	680	C
	ORT8850L-1BMN680C	1	Lead-Free PBGAM (fpBGA)	680	C
ORT8850H	ORT8850H-2BMN680C	2	Lead-Free PBGAM (fpBGA)	680	C
	ORT8850H-1BMN680C	1	Lead-Free PBGAM (fpBGA)	680	C

Table 45. Lead-Free Packaging – Industrial Ordering Information[1]

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade
ORT8850L	ORT8850L-2BMN680I	2	Lead-Free PBGAM (fpBGA)	680	I
	ORT8850L-1BMN680I	1	Lead-Free PBGAM (fpBGA)	680	I
ORT8850H	ORT8850H-1BMN680I	1	Lead-Free PBGAM (fpBGA)	680	I

1.For all but the slowest commercial speed grade, the speed grades on these devices are dual marked. For example, the commercial speed grade -2XXXXXC is also marked with the industrial grade -1XXXXXI. The commercial grade is always one speed grade faster than the associated dual mark industrial grade. The slowest commercial speed grade is marked as commercial grade only.

Revision History

Date	Version	Change Summary
–	8	Previous Lattice releases.
January 2004	8.1	Added lead-free package designator.
August 2004	9	Added lead-free package ordering part numbers (OPNs).
October 2005	10	Added clarification to the STM Pointer Mover bypass. Added clarification to the signal description for LINE_FP.
April 2006	11	Added clarification to the B1, section bit interleavered parity (BIP-8) byte. Added clarification to B1 processing.
February 2008	11.1	Corrected name of Register 30009, Bits 5 & 6 in Memory Map Description table.