Lattice Semiconductor Corporation

ORCA® ORT82G5

1.0-3.7 Gbits/s
8b/10b Backplane Interface FPSC

October 2002 **Preliminary Data Sheet**

Introduction

Lattice Semiconductor has developed a next generation FPSC intended for high-speed serial backplane data transmission. Built on the Series 4 reconfigurable embedded System-on-a-Chip (SoC) architecture, the ORT82G5 is made up of backplane transceivers containing eight channels, each operating at up to 3.7 Gbits/s (2.96 Gbits/s data rate), with a full-duplex synchronous interface with built-in Rx Clock and Data Recovery (CDR), and transmitter preemphasis along with more than 400K usable FPGA system gates. The CDR circuitry is a macrocell available from Lattice's smart silicon macro library and has already been implemented in numerous applications, including ASICs, standard products, and FPSCs, to create interfaces for SONET/SDH, Fibre Channel, and Ethernet (GbE, 10 GbE) applications. With the addition of protocol and access logic such as protocol-independent framers, Fibre Channel link layer capabilities, 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 any generic system. For example, designers can build a 20 Gbits/s bridge for 10 G Ethernet; the high-speed SERDES interfaces can comprise two XAUI interfaces with configurable back-end interfaces such as XGMII. The ORT82G5 can also be used to provide a full 10 G backplane data connection with protection between a line card and switch fabric.

The ORT82G5 provides a clockless high-speed interface for interdevice communication on a board or across a backplane. The built-in clock recovery of the ORT82G5 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 device supports embedded 8b/10b encoding/decoding and link state machines for 10 G Ethernet, and Fibre Channel.

The ORT82G5 is pinout compatible with a sister device, the ORSO82G5, which implements 8 channels of SERDES with SONET scrambling and cell processing.

Table 1. ORCA ORT82G5 Family – Available FPGA Logic

Device	PFU Rows	PFU Columns	Total PFUs	FPGA Max User I/O	LUTs	EBR Blocks	EBR Bits (K)	Usable Gates (K)[1, 2]
ORT82G5	36	36	1296	372	10,368	12	111	333 - 643

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

Embedded Function Features

- High-speed SERDES with programmable serial data rates including 1.0 Gbits/s, 1.25 Gbits/s, 2.5 Gbits/s, 3.125 Gbits/s, and 3.7 Gbits/s. Operation has been demonstrated on design tolerance devices at 3.7 Gbits/s across 26 in. of FR-4 backplane and at 3.2 Gbits/s across 40 in. of FR-4 backplane across temperature and voltage specifications.
- Asynchronous operation per receive channel with the receiver frequency tolerance based on one reference clock per quad channels (separate PLL per channel).
- Ability to select full-rate or half-rate operation per transmit or receive channel by setting the appropriate control registers.
- Programmable one-half amplitude transmit mode for reduced power in chip-to-chip application.
- Transmit preemphasis (programmable) for improved receive data eye opening.
- 32-bit (8b/10b) or 40-bit (raw data) parallel internal bus for data processing in FPGA logic.
- Provides a 10 Gbits/s backplane interface to switch fabric. Also supports port cards at 40 Gbits/s or 2.5 Gbits/s.
- 3.125 Gbits/s SERDES compliant with XAUI serial data specification for 10 G Ethernet applications with protection.
- Most XAUI features for 10 G Ethernet are embedded including the required link state machine.
- Compliant to Fibre Channel physical layer specification.
- High-Speed Interface (HSI) function for clock/data recovery serial backplane data transfer without external clocks.
- Eight-channel HSI function provides 2.96 Gbits/s serial user data interface per channel for a total chip bandwidth of 23.68 Gbits/s (full duplex).
- SERDES has low-power CML buffers. Support for 1.5 V/1.8 V I/Os. Allows use with optical transceiver, coaxial copper media, shielded twisted pair wiring or high-speed backplanes such as FR-4.
- Power down option of SERDES HSI receiver or transmitter on a per-channel basis.
- Automatic lock to reference clock in the absence of valid receive data.
- Per channel Pseudo-Random Bit Sequence (PRBS) generator and checker.
- High-speed and low-speed loopback test modes.
- Requires no external component for clock recovery and frequency synthesis.
- SERDES characterization pins available to control/monitor the internal interface to one SERDES quad macro.
- SERDES HSI automatically recovers from loss-of-clock once its reference clock returns to normal operating state.
- Built-in boundary scan (IEEE ® 1149.1 and 1149.2 JTAG) for the programmable I/Os, not including the SERDES interface.
- FIFOs can align incoming data either across all eight channels (all eight channels, across two groups of four channels, or across four groups of two channels). Alignment is done either using comma characters or by using the /A/ character in XAUI mode. Optionally, the alignment FIFOs can be bypassed for asynchronous operation between channels. (Each channel includes its own clock and frame pulse or comma detect.)
- Addition of two 4K X 36 dual-port RAMs with access to the programmable logic.
- Pinout compatible to the ORCA ORSO82G5 SONET backplane driver FPSC.

Programmable Features

- High-performance programmable logic:
  - 0.16 µm 7-level metal technology.
  - Internal performance of >250 MHz.
  - Over 400K usable system gates.
  - Meets multiple I/O interface standards.
  - 1.5 V operation (30% less power than 1.8V 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 3.3V PCI compliance.
  - Individually programmable drive capability: 24 mA sink/12 mA source, 12 mA sink/6 mA source, or 6 mA sink/3 mA source.
  - Two slew rates supported (fast and slew-limited).
  - Fast-capture input latch and input Flip-Flop (FF)/latch for reduced input setup time and zero hold time.
  - Fast open-drain drive capability.
  - Capability to register 3-state enable signal.
  - Off-chip clock drive capability.
  - Two-input function generator in output path.
- New programmable high-speed I/O:
  - Single-ended: GTL, GTL+, PECL, SSTL3/2 (class I and II), HSTL (Class I, III, IV), ZBT, and DDR.
  - Double-ended: LVDS, bused-LVDS, and LVPECL. Programmable (on/off) internal parallel termination (100 Ω) is also supported for these I/Os.
- New capability to (de)multiplex I/O signals:
  - New DDR on both input and output at rates up to 350 MHz (700 MHz effective rate).
  - New 2x and 4x downlink and uplink capability per I/O (i.e., 50 MHz internal to 200 MHz I/O).
- Enhanced twin-quad Programmable Function Unit (PFU):
  - Eight 16-bit Look-Up Tables (LUTs) per PFU.
  - Nine user registers per PFU, one following each LUT, and organized to allow two nibbles to act independently, plus one extra for arithmetic operations.
  - New register control in each PFU has two independent programmable clocks, clock enables, local SET/RESET, and data selects.
  - New LUT structure allows flexible combinations of LUT4, LUT5, new LUT6, 4 → 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 Supplemental Logic and Interconnect Cell (SLIC) decoders as bank drivers.
  - Soft-Wired LUTs (SWL) allow fast cascading of up to three levels of LUT logic in a single PFU through fast internal routing which reduces routing congestion and improves speed.
  - Flexible fast access to PFU inputs from routing.
  - Fast-carry logic and routing to all four adjacent PFUs for nibble-wide, byte-wide, or longer arithmetic functions, with the option to register the PFU carry-out.
- Abundant high-speed buffered and nonbuffered routing resources provide 2x average speed improvements over previous architectures.
- Hierarchical routing optimized for both local and global routing with dedicated routing resources. This results in faster routing times with predictable and efficient performance.
- SLIC provides eight 3-statable buffers, up to a 10-bit decoder, and PAL®-like AND-OR-Invert (AOI) in each programmable logic cell.
- New 200 MHz embedded quad-port RAM blocks,
  2 read ports, 2 write ports, and 2 sets of byte lane enables. Each embedded RAM block can be configured as:
  - 1—512 x 18 (quad-port, two read/two write) with optional built in arbitration.

  - 1—256 x 36 (dual-port, one read/one write).
  - 1—1K x 9 (dual-port, one read/one write).
  - 2—512 x 9 (dual-port, one read/one write for each).
  - 2 RAMS with arbitrary number of words whose sum is 512 or less by 18 (dual-port, one read/one write).
  - Supports joining of RAM blocks.
  - Two 16 x 8-bit content addressable memory (CAM) support.
  - FIFO 512 x 18, 256 x 36, 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 standard cell blocks with 100 MHz bus performance. Included are built-in system registers that act as the control and status center for the device.
- Built-in testability:
  - Full boundary scan (IEEE 1149.1 and Draft 1149.2 JTAG).
  - Programming and readback through boundary scan port compliant to IEEE Draft 1532:D1.7.
  - TS_ALL testability function to 3-state all I/O pins.
  - New temperature-sensing diode.
- Improved built-in clock management with Programmable Phase-Locked Loops (PPLLs) provide optimum clock modification and conditioning for phase, frequency, and duty cycle from 20 MHz up to 420 MHz. Multiplication of the input frequency up to 64x and division of the input frequency down to 1/64x possible.
- New cycle stealing capability allows a typical 15% to 40% internal speed improvement after final place and route. This feature also enables compliance with many setup/hold and clock to out I/O specifications and may provide reduced ground bounce for output buses by allowing flexible delays of switching output buffers.

Programmable Logic System Features

- PCI local bus compliant for FPGA I/Os.
- Improved PowerPC ® 860 and PowerPC II high-speed synchronous microprocessor interface can be used for configuration, readback, device control, and device status, as well as for a general-purpose interface to the FPGA logic, RAMs, and embedded standard cell blocks. Glueless interface to synchronous PowerPC processors with user-configurable address space provided.
- New embedded AMBA ™ specification 2.0 AHB system bus (ARM ® processor) facilitates communication among the microprocessor interface, configuration logic, Embedded Block RAM, FPGA logic, and embedded standard cell blocks.
- Variable size 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 OR4E4).
- 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; as well as POS-PHY3.

Description

What is a FPSC?

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

FPSC Overview

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

FPSC Gate Counting

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

FPGA/Embedded Core Interface

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

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

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

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

FPSC Design Kit

Development is facilitated by an FPSC design kit which, together with ispLEVER System 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 complied Verilog simulation model, HSPICE and/or IBIS models for I/O buffers, and complete online documentation. The kit's software coupled with the design environment, provides a seamless FPSC design environment. More information can be obtained by visiting the Lattice web site or contacting a local sales office, both listed on the last page of this document.

FPGA Logic Overview

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

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

The architecture consists of four basic elements: Programmable Logic Cells (PLCs), Programmable I/O cells (PIOs), Embedded Block RAMs (EBRs), plus supporting system-level features. These elements are interconnected with a rich routing fabric of both global and local wires. An array of PLCs is surrounded by common interface blocks which provide an abundant interface to the adjacent PLCs or system blocks. Routing congestion around these critical blocks is eliminated by the use of the same routing fabric implemented within the programmable logic core.

Each PLC contains a PFU, SLIC, local routing resources, and configuration RAM. Most of the FPGA logic is performed in the PFU, but decoders, PAL-like functions, and 3-state buffering can be performed in the SLIC. The PIOs provide device inputs and outputs and can be used to register signals and to perform input demultiplexing, output multiplexing, uplink and downlink functions, and other functions on two output signals.

Large blocks of 512 x 18 quad-port RAM complement the existing distributed PFU memory. The RAM blocks can be used to implement RAM, ROM, FIFO, multiplier, and CAM. Some of the other system-level functions include the MPI, PLLs, and the Embedded System Bus (ESB).

PLC Logic

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

The PFU is organized in a twin-quad fashion; two sets of four LUTs and FFs that can be controlled independently. Each PFU has two independent programmable clocks, clock enables, local set/reset, and data selects. LUTs may also be combined for use in arithmetic functions using fast-carry chain logic in either 4-bit or 8-bit modes. The carry-out of either mode may be registered in the ninth FF for pipelining.

Each PFU may also be configured as a synchronous 32 x 4 single- or dual-port RAM or ROM. The FFs (or latches) may obtain input from LUT outputs or directly from invertible PFU inputs, or they can be tied high or tied low. The FFs also have programmable clock polarity, clock enables, and local set/reset.

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

Programmable I/O

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

Each PIO contains four programmable I/O pads and is interfaced through a common interface block to the FPGA array. The PIO is split into two pairs of I/O pads with each pair having independent clock enables, local set/reset, and global set/reset. On the input side, each PIO contains a programmable latch/Flip-Flop which enables very fast latching of data from any pad. The combination provides for very low setup requirements and zero hold times for signals coming on-chip. It may also be used to demultiplex an input signal, such as a multiplexed address/data signal, and register the signals without explicitly building a demultiplexer with a PFU.

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

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

The Series 4 I/O logic has been enhanced to include modes for speed uplink and downlink capabilities. These modes are supported through shift register logic, which divides down incoming data rates or multiplies up outgoing data rates. This new logic block also supports high-speed DDR mode requirements where data 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, and 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 allow for easy glueless system interfacing and the capability to adjust to varying conditions in today's high-speed networking systems.

Microprocessor Interface

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

A system-level microprocessor interface to the FPGA user-defined logic following configuration, through the system bus, including access to the Embedded Block RAM and general user-logic, is provided by the MPI. The MPI supports burst data read and write transfers, allowing short, uneven transmission of data through the interface by including data FIFOs. Transfer accesses can be single beat (1 x 4 bytes or less), 4-beat (4 x 4 bytes), 8-beat (8 x 2 bytes), or 16-beat (16 x 1 bytes).

System Bus

An on-chip, multimaster, 8-bit system bus with 1-bit parity facilitates communication among the MPI, configuration logic, FPGA control, status registers, Embedded Block RAMs, as well as user logic. Utilizing the AMBA specification Rev 2.0 AHB protocol, the Embedded System Bus offers arbiter, decoder, master, and slave elements. Master and slave elements are also available for the user-logic and a slave interface is used for control and status of the embedded backplane transceiver portion of the ORT82G5.

The system bus control registers can provide control to the FPGA such as signaling for reprogramming, reset functions, and PLL programming. Status registers monitor INIT, DONE, and system bus errors. An interrupt controller is integrated to provide up to eight possible interrupt resources. Bus clock generation can be sourced from the 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

Up to eight PLLs are provided on each Series 4 device, with four user PLLs generally provided for FPSCs. Programmable PLLs can be used to manipulate the frequency, phase, and duty cycle of a clock signal. Each PPLL is capable of manipulating and conditioning clocks from 20 MHz to 200 MHz. Frequencies can be adjusted from 1/8x to 8x, the input clock frequency. Each programmable PLL provides two outputs that have different multiplication factors but can have the same phase relationships. Duty cycles and phase delays can be adjusted in 12.5% of the clock period increments. An automatic input buffer delay compensation mode is available for phase delay. Each 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 multiple 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 FPGA devices, or visit the Lattice web site at www.latticesemi.com..

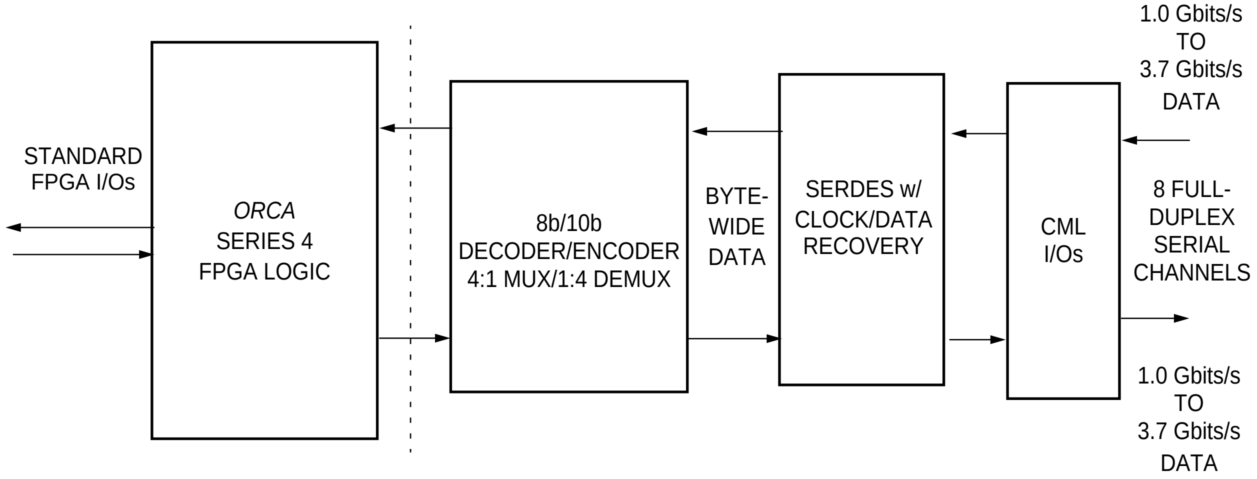
ORT82G5 Overview

The ORT82G5 FPSC provides high-speed backplane transceivers combined with FPGA logic. It is based on 1.5 V OR4E04 ORCA FPGA and has a 36 x 36 array of Programmable Logic Cells (PLCs). The embedded core, which contains the backplane transceivers is attached to the right side of the device and is integrated directly into the FPGA array. A top level diagram of the basic chip configuration is shown in Figure 1.

Embedded Core Overview

The embedded core portion of the ORT82G5 contains eight Clock and Data Recovery (CDR) macrocells and eight Serialize/Deserialize (SERDES) blocks and supports eight channels of 8b/10b (*IEEE* 802.3z) encoded serial links. It is intended for high-speed serial backplane data transmission. Figure 1 shows the ORT82G5 top level block diagram and the basic data flow. Boundary scan for the ORT82G5 only includes programmable I/Os and does not include any of the embedded block I/Os.

Figure 1. ORT82G5 Top Level Block Diagram



The ORT82G5's eight channels can each operate at up to 3.7 Gbits/s (2.96 Gbits/s data rate) with a full-duplex synchronous interface with built-in clock recovery (CDR). The 8b/10b encoding provides guaranteed ones density for the CDR, byte alignment, and error detection. The core is also capable of frame synchronization and physical link monitoring and contains independent 4k x 36 RAM blocks. Overviews of the various blocks in the embedded core are presented in the following paragraphs.

Serializer and Deserializer (SERDES)

The SERDES portion of the core contains two quad transceiver blocks for serial data transmission at a selectable data rate of 1.0-3.7 Gbits/s.Each SERDES channel features high-speed 8b/10b parallel I/O interfaces to other core blocks and high-speed CML interfaces to the serial links.

The SERDES circuitry consists of receiver, transmitter, and auxiliary functional blocks. The receiver accepts high-speed (up to 3.7 Gbits/s) serial data. Based on data transitions, the receiver locks an analog receive PLL for each channel to retime the data, then demultiplexes the data down to parallel bytes and an accompanying clock.

The transmitter operates in the reverse direction. Parallel bytes are multiplexed up to 3.7 Gbits/s serial data for off-chip communication. The transmitter generates the necessary 3.7 GHz clocks for operation from a lower speed reference clock

The transceivers are controlled and configured through the system bus in the FPGA logic and through the external 8-bit microprocessor interface of the FPGA. Each channel has associated dedicated registers that are readable and writable. There are also global registers for control of common circuitry and functions.

The SERDES performs 8b/10b encoding and decoding for each channel. The 8b/10b transmission code can support either Ethernet or Fibre Channel specifications for serial encoding/decoding, special characters, and error detection.

The user can disable the 8b/10b decoder to receive raw 10-bit words which will be rate reduced by the SERDES. If this mode is chosen, the user must bypass the multichannel alignment FIFOs.

The SERDES block contains its own dedicated PLLs for both transmit and receive clock generation. The user provides a reference clock of the appropriate frequency. The receiver PLLs extract the clock from the serial input data and retime the data with the recovered clock.

MUX/DEMUX Block

The MUX/DEMUX block converts the data format for the high speed serial links to a wide, low-speed format for crossing the CORE/FPGA interface. The intermediate interface to the SERDES macrocell runs at 1/10th the bit rate of the data lane. The MUX/DEMUX converts the data rate and bus width so the interface to the FPGA core can run at 1/4th this intermediate frequency, giving a range of 25.0-92.5 MHz for the data rates into and out of the FPGA logic.

Multichannel Alignment FIFOs

The eight incoming data channels (four per SERDES block) can be independent of each other or can be synchronized in several ways. Two channels within a SERDES quad can be aligned together; channels A and B and/or channels C and D. Alternatively, four channels in a SERDES quad can be aligned together to form a communication channel with a bandwidth of 10 Gbits/s. Finally, the alignment can be extended across both SERDES quads to align all eight channels. Individual channels within an alignment group can be disabled (i.e., powered down) without disrupting other channels.

XAUI and Fibre Channel Link State Machines

Two separate link state machines are included in the ORT82G5. A XAUI compliant link state machine is included in the embedded core to implement the IEEE 802.3ae standard. A separate state machine for Fibre Channel is also implemented.

FPGA/Embedded Core Interface

In 8b/10b mode, the FPGA logic will receive/transmit 32-bits of data (up to 92.5 MHz) and 4 K_CTRL bits from/to the embedded core. There are 8 data streams in each direction plus additional timing, status and control signals.

Data sent to the FPGA can be aligned using comma (/K/) characters or /A/ character as specified either by Fibre Channel or by IEEE 802.3ae for XAUI based interfaces. The alignment character is made available to the FPGA along with the data. The special characters K28.1, K28.5 and K28.7 are treated as valid comma characters by the SERDES.

If the receive channel alignment FIFOs are bypassed, then each channel will provide its own receive clock in addition to data and comma character detect signals. If the 8b/10b decoders are bypassed, then 40-bit data streams are passed to the FPGA logic. No channel alignment can be done in 8b/10b bypass mode.

Dual Port RAMs

In addition to the backplane interface blocks, there are two independent memory blocks in the ASB. Each memory block has a capacity of 4k words by 36 bits. It has one read port, one write port, and four byte-write-enable (active-low) signals. The read data from the memory block is registered so that it works as a pipelined synchronous memory block.

FPSC Configuration

Configuration of the ORT82G5 occurs in two stages: FPGA bitstream configuration and embedded core setup.

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

After the FPGA configuration is complete, the options for the embedded core are set based on the contents of registers that are accessed through the FPGA system bus.

The system bus itself can be driven by an external PowerPC 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 register interface and very little FPGA logic is available in the MPI/System Bus Technical Note. This IP block sets up the embedded core via a state machine and allows the ORT82G5 to work in an independent system without an external microprocessor interface.

Backplane Transceiver Core Detailed Description

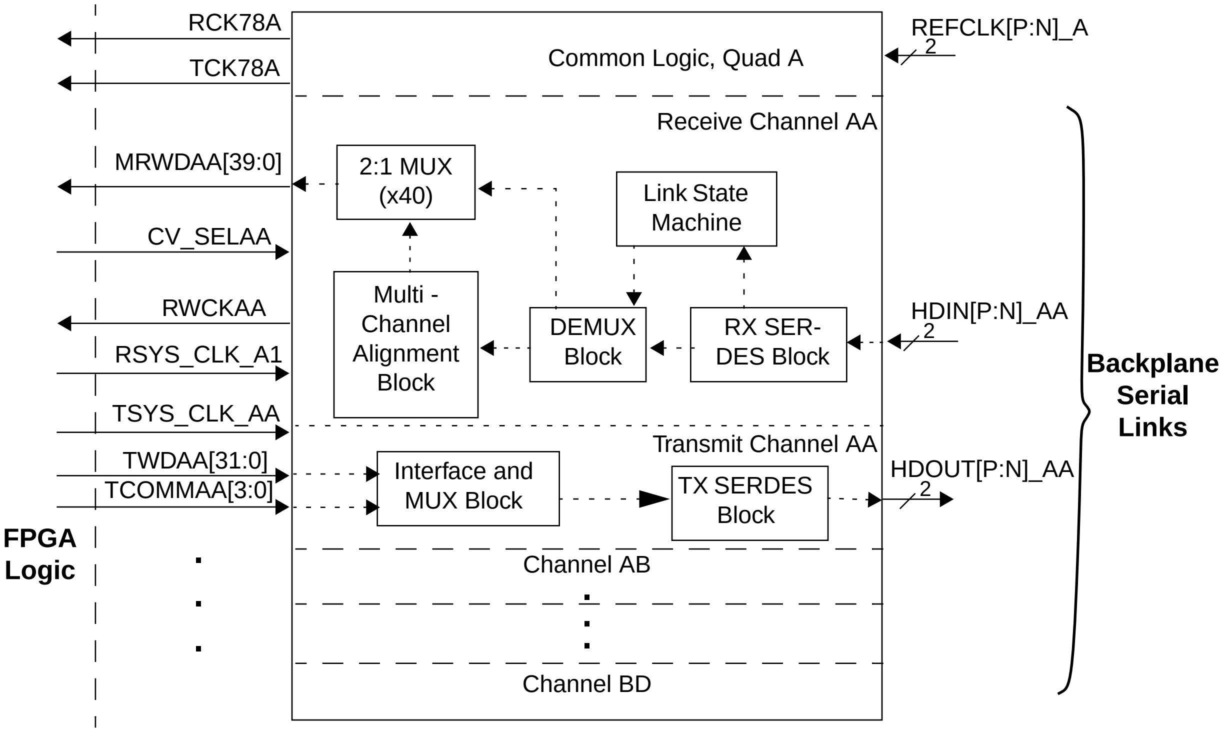
The following sections describe the various logic blocks in the Embedded Core portion of the FPSC. The FPGA section of the FPSC is identical to an ORCA 4E4 FPGA except that the pads on one edge of the FPGA chip are replaced by the Embedded Core. For a detailed description of the programmable logic functions, please see the ORCA Series 4 Data Sheet and related application and technical notes.

The major functional blocks in the Embedded Core include:

- Two quad-channel SERializer-DESerializer (SERDES) blocks
- 8b/10b encoder/decoders
- 4-to-1 multiplexers (MUX) and 1-to-4 demultiplexers (DEMUX)
- Fibre channel synchronization state machine
- XAUI link alignment state machine
- Alignment FIFOs
- Embedded 4K x 36 RAM blocks (independent from transceiver logic).

A top level block diagram of the Embedded Core Logic is shown in Figure 2. The Embedded RAM blocks are not shown. The external pins for the Embedded Core are listed later in this data sheet in and the signals at the Embedded Core/FPGA interface are listed in Table 9 and Table 18.

Figure 2. Top Level Block Diagram, Embedded Core Logic (Channel AA)



The Embedded Core provides transceiver functionality for eight serial data channels and is organized into two quads, each supporting four channels. Each channel is identified by both a quad identifier [A:B] and a channel identifier [A:D]. The data channels can operate independently or they can be combined together (aligned) to achieve higher bit rates. The mode operation of the core is defined by a set of control registers, which can be written through the system bus interface. Also, the status of the core is stored in a set of status registers, which can be read through the system bus interface.

The transmitter section for each channel accepts 40 bits of data or 32 bits of data and eight control/status bits from the FPGA logic and optionally encodes the data using 8b/10b encoding. It also accepts the low-speed reference clock at the REFCLK input and uses this clock to synthesize the internal high-speed serial bit clock. The data is then serialized and the serialized data are available at the differential CML output terminated in 86 Ω to drive either an optical transmitter or coaxial media or circuit board/backplane.

The receiver section receives high-speed serial data at its differential CML input port. These data are fed to the clock recovery section which generates a recovered clock and retimes the data. The retimed data are also deserialized, optionally 8b/10b decoded and presented as parallel data to the FPGA logic. Two-phase receive byte clocks are available synchronous with the parallel words. The receiver also optionally recognizes the comma characters or code violations and aligns the bit stream to the proper word boundary.

8b/10b Encoding and Decoding

In 8b/10b mode, the FPGA logic will receive/transmit 32 bits of data and 4 K_CTRL bits from/to the embedded core. In the transmit direction, four additional input bits force negative disparity coding. The embedded core logic will encode the data to or decode the data from a 10-bit format according to the FC-PH ANSI X3.230:1994 standard (which is also the encoding used by the IEEE 802.3ae Ethernet standard). This encoding/decoding scheme also allows for the transmission of special characters and supports error detection.

Following the definitions and conventions used in defining the 8b/10b coding rules, each valid coded character has a name corresponding to its 8-bit binary value:

- Dxx.y for data characters
- Kxx.y for special characters
- xx = the 5-bit input value, base 10, for bits ABCDE
- y = the 3-bit input value, base 10, for bits FGH

An 8b/10b encoder is designed to maintain a neutral average disparity. Disparity is the difference between the number of 1s and 0s in the encoded word. Neutral disparity indicates the number of 1s and 0s are equal. Positive disparity indicates more 1s than 0s. Negative disparity indicates more 0s than 1s. The average disparity determines the DC component of the signals on the serial line. Running disparity is a record of the cumulative disparity of every encoded word, and is tracked by the encoder.

In order to maintain neutral disparity, two different codings are defined for each data value. The 8b/10b encoder in the transmit path selects between (+) and (-) encoded word based on calculated disparity of the present data to maintain neutral disparity

In the receive path, the clock and data recovery blocks retime the incoming data and 8b/10b decoders generate 8-bit data based on the received 10-bit data. A sequence of valid 8b/10b coded characters has a maximum run length of 5 bits (i.e., 5 consecutive ones or 5 consecutive zeros before a mandatory bit transition). This assures adequate transitions for robust clock recovery.

The recovered data is aligned on a 10-bit boundaries by detecting and aligning to special characters in the incoming data stream. Data is word aligned using the comma (/K/) character. A comma character is a special character that contains a unique pattern (0011111 or its complement 1100000) in the 10-bit space that makes it useful for delimiting word boundaries. The special characters K28.1, K28.5 and K28.7 contain this comma sequence and are treated as valid comma characters by the SERDES.

The following table shows all of the valid special characters. All of the special characters are made available to the FPGA logic; however only the comma characters are used by the SERDES logic. The different codings that are possible for each data value are shown as encoded word (+) and encoded word (-). The table also illustrates the 8b/10b bit labeling convention. The bit positions of the 8-bit characters are labeled as H,G,F,E,D,C,B and A and the bit positions of the 10-bit encoded characters are labeled as a, b, c, d, e, i, f, g, h, and j. The encoded words are transmitted serially with bit ‘a’ transmitted first and bit ‘j’ transmitted last.

Table 2. Valid Special Characters

K Character	HGF EDCBA 765 43210	K Control	Encoded Word (–)	Encoded Word (+)
			abcdei fghj	abcdei fghj
K28.0	000 11100	1	001111 0100	110000 1011
K28.1 /comma/	001 11100	1	001111 1001	110000 0110
K28.2	010 11100	1	001111 0101	110000 1010
K28.3 /A/	011 11100	1	001111 0011	110000 1100
K28.4	100 11100	1	001111 0010	110000 1101
K28.5 /comma/	101 11100	1	001111 1010	110000 0101
K28.6	110 11100	1	001111 0110	110000 1001
K28.7 /comma/	111 11100	1	001111 1000	110000 0111
K23.7	111 10111	1	111010 1000	000101 0111
K27.7	111 11011	1	110110 1000	001001 0111
K29.7	111 11101	1	101110 1000	010001 0111
K30.7	111 11110	1	011110 1000	100001 0111

Transmit Path (FPGA → Backplane) Logic

The transmitter section accepts either 8-bit unencoded data or 10-bit encoded data at the parallel interface to the FPGA logic. It also uses the reference clock, REFCLK[P:N]_[A:B] to synthesize an internal high-speed serial bit clock. The serialized transmitted data are available at the differential CML output pins to drive either an optical transmitters, coaxial media or a circuit board backplane.

As shown in Figure 3, the basic blocks in the transmit path include:

Embedded Core/FPGA interface and 4:1 multiplexer

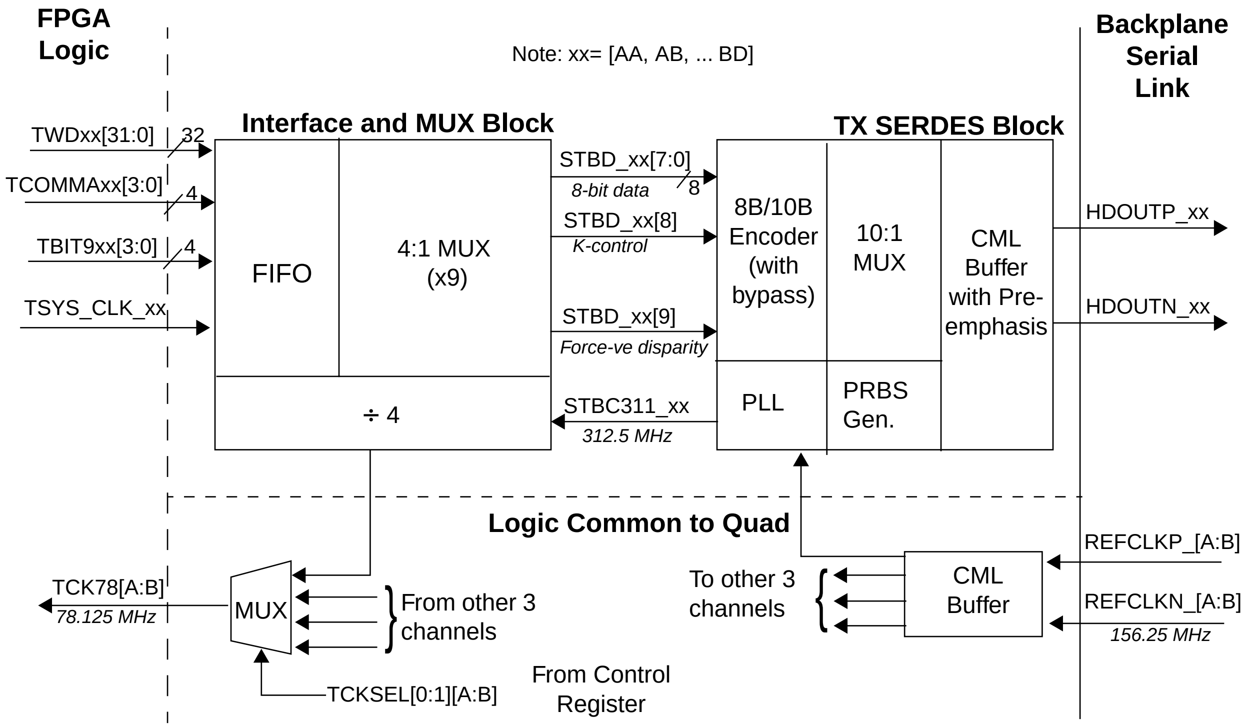
- Low speed parallel core/FPGA interface
- 4:1 multiplexer

Transmit SERDES

- 8b/10b Encoder
- 10:1 Multiplexer
- CML Output Buffer
- Pseudo-Random Bit Sequence (PRBS) Generator and Checker

Detailed descriptions of the logic blocks are given in following sections. Detailed descriptions of transmit clock distribution, including the transmit PLL, and of the Pseudo-Random Bit Sequence (PRBS) generator are given in later sections of this data sheet.

Figure 3. Basic Logic Blocks, Transmit Path, Single Channel (Typical Reference Clock Frequency)



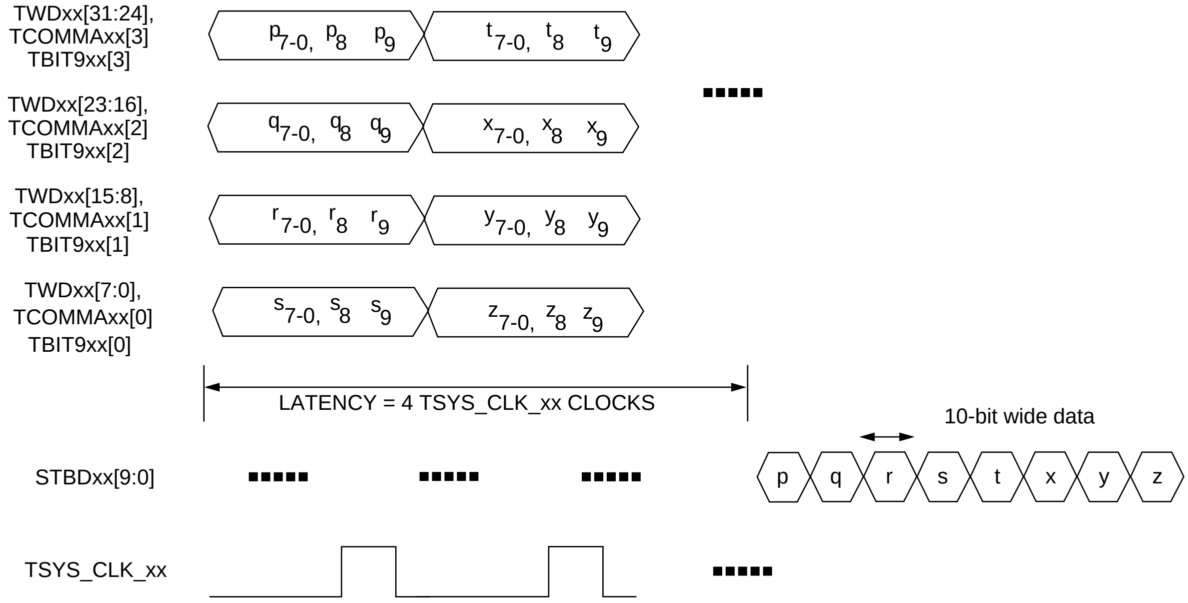
Embedded Core/FPGA Logic Interface and 4:1 Multiplexer

These blocks provide the data formatting and transmit data and clock signal transfers between the Embedded Core and the FPGA Logic. Control and status registers in the FPGA portion of the chip contain to control the transmit logic and record status. These bits are passed to the core using the FPGA System Bus and are described in later sections of this data sheet Table 8.

The low-speed transmit interface consists of a clock and 4 data bytes, each with an accompanying control bit. The data bytes are conveyed to the MUX via the TWDxx[31:0] ports (where xx represents the channel label [AA,...,BD]). The control bits are TCOMMAx[3:0] which define whether the input byte is to be interpreted as data or as a special character and TBIT9xx[3:0] which are used to force negative disparity encoding. The data and control signals are synchronized to the transmit clock, TSYS_CLK_xx. Both the data and control are strobed into the core on the rising edge of TSYS_CLK_xx.

The MUX is responsible for taking 40 bits of data/control at the low-speed transmit interface and up-converting it to 10 bits of data/control at the SERDES transmit interface. The MUX has 2 clock domains - one based on the clock received from the SERDES block and a second that comes from the FPGA at 1/4 the frequency of the SERDES clock. The time sequence of interleaving data/control values is shown in Figure 4 below.

Figure 4. Transmit MUX Block Timing - Single Channel

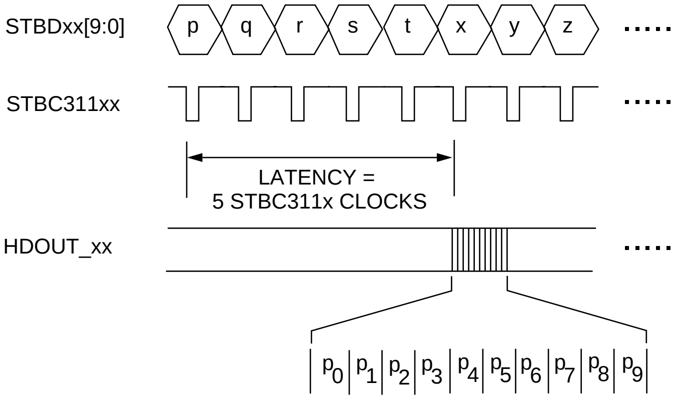


SERDES Block

The SERDES block accepts either 8-bit unencoded data or 10-bit encoded data at the parallel input port from the MUX/DEMUX block. It also accepts the reference clock at the REFCLK_[A:B] input and uses this clock to synthesize the internal high-speed serial bit clock.

The internal STBC311xx clock is derived from the reference clock. The frequency of this clock depends on the setting of the half-rate/full-rate control bit setting the mode of the SERDES and the frequency of the REFCLK_[A:B] and/or that of the high-speed serial data. A falling edge on the STBC311xx clock port will cause a new data character to be transferred into the SERDES block. The latency from the SERDES block input to the high-speed serial output is 5 STBC311xx clock cycles, as shown in Figure 5.

Figure 5. ORT82G5 Transmit Path Timing - Single SERDES Channel



Each quad also sends a clock to the FPGA logic. This clock, TCK78[A,B], is sourced from one of the four MUX blocks and has the same frequency as TSYS_CLK_xx, but arbitrary phase. Within each MUX block, the low frequency clock output is obtained by dividing by 4 the SERDES STBC311x clock which is used internally to synchronize the transmit data words. TCKSEL control bits select the channel to source TCK78[A:B].

The internal signals STBDxx[9:0] (where xx is represents AA—BD) from the MUX block carry unencoded character data and control bits. The 10th bit (STBDxx[9]) of each data lane into the SERDES is used to force negative disparity encoding.

8b/10b Encoder and 1:10 Multiplexer

The 8b/10b encoder encodes the incoming 8-bit data into a 10-bit format as described previously. The input signals to the block, STBDxx[7:0] are used for the 8 bit unencoded data. STBDxx[8] is used as the K_control input to indicate whether the 8 data bits need to be encoded as special characters (K_control = 1) or as data characters (K_control = 0). When STBDxx[9:0] = 1, negative disparity encoding is forced. When the encoder is bypassed STBDxx[9:0] serve as the data bits for the 10-bit unencoded data.

Within the definition of the 8b/10b transmission code, the bit positions of the 10-bit encoded transmission characters are labeled as a, b, c, d, e, i, f, g, h, and j in that order. Bit a corresponds to STBDxx[0], bit b to STBDxx[1], bit c to STBDxx[2], bit d to STBDxx[3], bit e to STBDxx[4], bit i to STBDxx[5], bit f to STBDxx[6], bit g to STBDxx[7], bit h to STBDxx[8], and bit j to STBDxx[9].

The 10-bit wide parallel data is converted to serial data by the 10:1 Multiplexer. The serial data are then sent to the CML output buffer and are transmitted serially with STBDxx[0] transmitted first and STBDxx[9] transmitted last.

CML Output Buffer

The transmitter's CML output buffer is terminated on-chip in 86 ohms to optimize the data eye as well as to reduce the number of discrete components required. The differential output swing reaches a maximum of 1.2 VPP in the normal amplitude mode. A half amplitude mode can be selected via configuration register bit HAMP_xx. Half amplitude mode can be used to reduce power dissipation when the transmission medium has minimal attenuation or for testing of the integrity (loss) of the physical medium.

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

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

Applying too much preemphasis when it is not required, for example when driving a short backplane path, will also degrade the data eye opening at the receiver. In the ORT82G5 the degree of transmit preemphasis can be programmed with a two-bit control from the microprocessor interface as shown in Table 3. The high-pass transfer function of the preemphasis circuit is given by the following equation, where the value of a is shown in Table 3.

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

Table 3. Preemphasis Settings

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

Receive Path (Backplane → FPGA) Logic

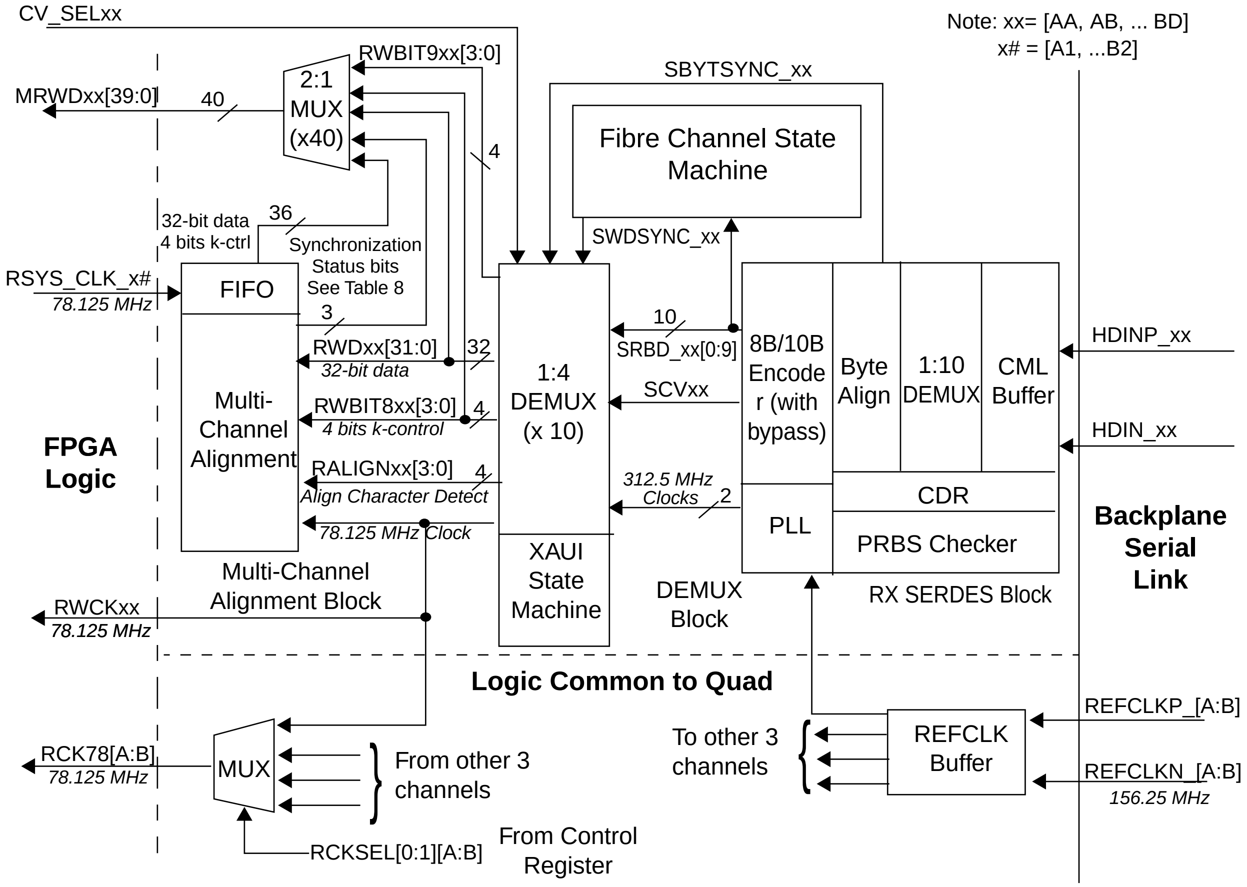
The receiver section receives high-speed serial data at the external differential CML input pins. These data are fed to the clock recovery section which generates a recovered clock and retimes the data. Therefore the receive clocks are asynchronous between channels. The retimed data are deserialized and presented as a 10-bit encoded or a 8-bit unencoded parallel data on the output port. The receiver also optionally recognizes comma characters, detects code violations and aligns the bit stream to the proper word boundary.

As shown in Figure 6, the basic blocks in the receive path include:

Receive SERDES Block

- CML input buffer
- Receive PLL
- 1:10 demultiplexer (DEMUX)
- Clock and Data Recovery (CDR) section
- PRBS checker
- 10B/8B decoder
- **1:4 demultiplexer and Embedded Core/FPGA interface**
- 1:4 DEMUX
- Low speed parallel Embedded Core/FPGA logic interface
- Multi-channel alignment logic

Figure 6. Basic Logic Blocks, Receive Path, Single Channel (Typical Reference Clock Frequency)



Each channel provides its own received clock, received data and K-character detect signals to the FPGA logic. Incoming data from multiple channels can be aligned using comma (/K/) characters or /A/ character (as specified either in Fibre Channel specifications or in IEEE 802.3ae for XAUI based interfaces). If the 8b/10b decoders are bypassed, then 40-bit data streams are passed to the FPGA logic. No channel alignment can be done in this 8b/10b bypass mode.

Each channel also contains a PRBS checker which can be used as a built-in Bit Error Rate Tester (BERT) during loopback testing. When enabled, it produces a one-bit PRBSCHK_xx (where xx stands for one of the letter pairs [AA,...,BD]) output to indicate whether there was an error in the received data. Detailed descriptions of data synchronization, of the SERDES, DEMUX and Multi-Channel Alignment blocks and of the Fibre Channel and XAUI state machines are given in following sections. Receive clock distribution and the PRBS operation are described in later sections of this data sheet.

Synchronization

The ORT82G5 SERDES RX logic performs four levels of synchronization on the incoming serial data stream. Each level builds upon the previous, providing first bit, then byte (character), then channel (32-bit word), and finally multichannel alignment. Each step is described functionally in the following paragraphs. The details of the logical implementations are described in subsequent sections.

Bit alignment is the task of the Clock/Data Recovery (CDR) block. This block utilizes a PLL that locks to the transitions in the incoming high-speed serial data stream, and outputs the extracted clock as well as the data. If the PLL is unable to lock to the serial data stream, it instead locks to REFCLK[A:B] to stabilize the voltage-controlled oscillator (VCO), and periodically switches back to the serial data stream to again attempt synchronization. This process continues until a valid input data stream is detected and lock is achieved. The CDR can maintain lock on data as long as the input data stream contains an adequate data “eye” (i.e., jitter is within specification) and the maximum data stream run length is not exceeded.

Bit alignment times fall into two categories: realignment when the input serial data stream experiences an abrupt phase change (as may occur when protection switching is performed between two paths having different delays), and alignment from a no-signal condition. Realignment is very quick, since the PLL’s VCO is already locked on frequency and only needs to adapt to the new phase. This re-alignment has been observed to require no more than one microsecond when REFCLK[A:B] = 156.25 MHz.

Alignment from a no-signal condition has two components. First, there is the re-acquisition to the data’s frequency and phase. The time required for re-acquisition to the data’s frequency is minimized by logic that periodically switches the PLL to lock to the REFCLK[A:B] when it fails to lock on the serial data stream, thus limiting the VCO’s frequency wander. Second, there is the time spent while the PLL is locking to REFCLK[A:B], which can be from zero to a maximum value, depending on when the serial data stream becomes valid in relation to the PLL’s switching to/from REFCLK[A:B]. This alignment has been observed to require no more than 4 microseconds when REFCLK = 156.25 MHz.

Byte alignment occurs once valid bit alignment is achieved. The byte aligner looks for a particular 7-bit sequence (either 0011111 or its complement, 1100000) that, in data that has been 8b/10b encoded per Fibre Channel or IEEE 802.3ae specifications, only occurs in the comma (/K/) characters K28.1, K28.5 and K28.7. Byte alignment only occurs when the ENBYSYNC_xx signal for that channel is active high, and re-alignment occurs on each 7-bit sequence encountered. However, if ENBYSYNC_xx is asserted active high and no comma character is encountered, and then is brought inactive low, the channel will still perform one byte alignment operation on the next comma character. Byte alignment occurs immediately when an alignment sequence is detected, so the lock time is only one clock period.

Note: Each time the byte aligner performs an alignment, it also corrects the phase of the internal RBC_xx clock. This can result in the “stretching” of the clock by a half-phase in order to cause the output data to align with the rising edge of RBC_xx.

Word (32-bit) alignment can occur after the Fibre Channel (XAUI_MODE_xx = 0) or XAUI (XAUI_MODE_xx = 1) state machine has reached the in-synchronization state. In Fibre Channel mode, synchronization (WDSYNC_xx = 1) will occur after three ordered sets of data have been received in the absence of any code violations. After this, the next ordered set will cause the output data to be aligned such that the comma character is in the most significant byte. Thus, 32-bit word alignment has been achieved when four ordered sets have been detected. The time required is directly dependent on comma-character density.

Note: once word alignment is accomplished, no further alignment occurs unless and until WDSYNC_xx goes to zero and back to one again. Comma characters that are not located in the most significant byte position will not trigger further re-alignment while WDSYNC_xx is active. This behavior is as defined by the Fibre Channel specification. However, it means that, if the channel experiences an abrupt delay change (as could occur if an external MUX performs a protection switch between two links) and if the delay change is close enough to a full character or characters that not enough code violations are generated to cause loss of WDSYNC_xx, the channel could become misaligned and remain that way indefinitely. As mentioned above, this behavior is that defined by the Fibre Channel specification.

In XAUI mode, as the state diagram later in this data sheet indicates, three error-free code-groups containing commas must be detected before synchronization is declared.

Multi (2, 4 or 8) channel alignment (Lane alignment in XAUI mode) can be performed after 32-bit word alignment is complete. Multi-channel alignment is described in later sections of this data sheet.

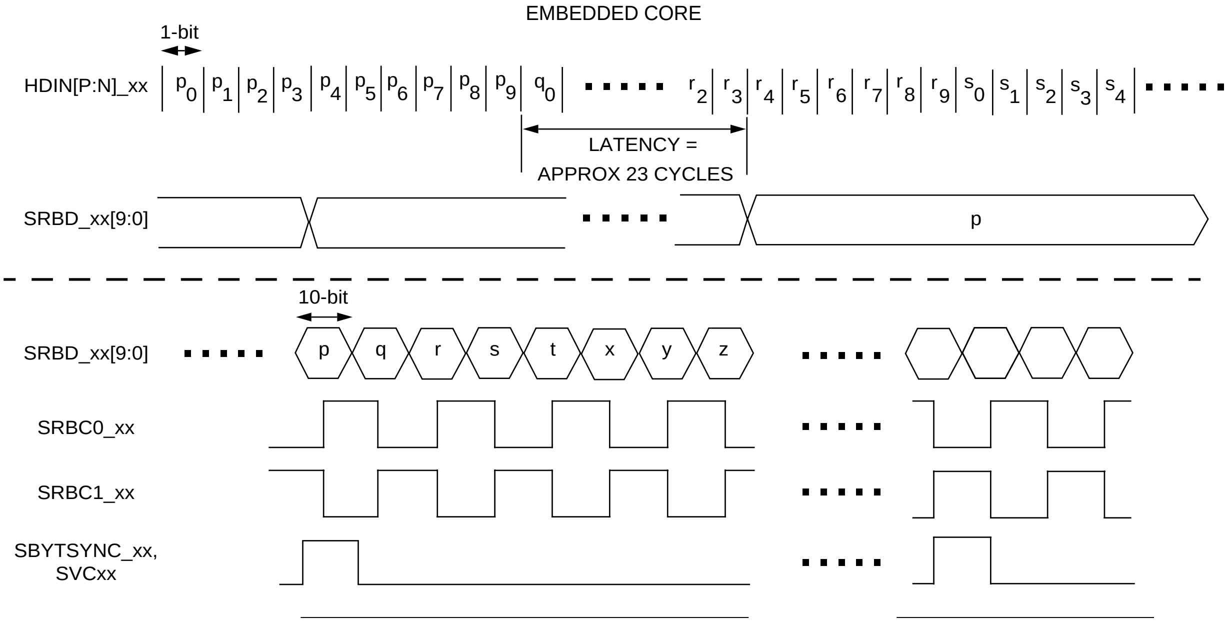
Receive CML Input Buffer and SERDES

The receiver section receives high-speed serial data at its differential CML input port. The receive input is an AC-coupled input. The received data is sent to the clock recovery section which generates a recovered clock and retimes the data. Valid data will be received after the receive PLL has locked to the input data frequency and phase.

The received serial data is converted to10-bit wide parallel data by the 1:10 demultiplexor. Clock recovery is performed by the SERDES block for each of the eight receive channels. This recovered data is then aligned to a 10-bit word boundary by detecting and aligning to a comma special character. Word alignment is done for either polarity of the comma character. The 10-bit code word is passed to the 8b/10b decoder, which provides an 8-bit byte of data, a special character indicator bit and a SBYTSYNC_xx signal (where again xx is a placeholder for AA,...,BD).

Data from a SERDES channel is sent to the DEMUX block in 10-bit raw form or 8-bit decoded form across the SRBD_xx [9:0] port with a latency of approximately 14-23 cycles (bit periods of the incoming data). Accompanying this data are the comma-character indicator (SBYTSYNC_xx), link-state indicator (SWDSYNC_xx), clocks (SRBC0_xx, and SRBC1_xx), and code-violation indicator (SCVxx). The two internal clocks operated at twice the reference clock frequency. Figure 7 shows the receive path timing for a single SERDES channel.

Figure 7. ORT82G5 Receive Path Timing for a Single SERDES Channel



With the 8b10bR_xx control bit of the SERDES channel set to 1, the data presented at SRBD_xx[9:0] will be decoded characters. Bit 8 will indicate whether SRBDxx[7:0] represents an ordinary data character (bit 8 = 0), or whether SRBD_xx[7:0] represents a special character, like a comma. Bit 9 may be either a code violation indicator or one of seven out of synchronization state indicators, as described later.

When 8b10bR is set to 0, the data at SRBD_xx[9:0] will not be decoded. The XAUI link-state machine should not be used in this mode of operation. When in XAUI mode, the MUX/DEMUX looks for /A/ (as defined in *IEEE* 802.3ae v.2.1) characters for channel alignment and requires the characters to be in decoded form for this to work

1:4 Demultiplexer (DEMUX)

The1:4 DEMUX has to accumulate four sets of characters presented to it at the SERDES receive interface and put these out at one time at the low-speed receive interface.

Another task of the 1:4 DEMUX is to recognize the synchronizing event and adjust the 4-byte boundary so that the synchronizing character leads off a new 4-byte word. In Fibre Channel mode, this synchronizing character is a comma. This feature will be referred to as DEMUX word alignment in other areas of this document. DEMUX word alignment will only occur when the communication channel is synchronized. When there is no synchronization of the link, the 1:4 DEMUX will continue to output 4-byte words at some arbitrary, but constant, boundary.

There are 2 control register bits available for each channel for word alignment. They are DOWDALGN_xx and NOWDALGN_xx. The DOWDALGN_xx bit is positive edge triggered. Writing a 0 followed by a 1 to this register bit will cause the corresponding DEMUX to look for a new comma character and align the 32-bit word such that the comma is in the most significant byte position. It is important that the comma is in the most significant byte position since the multi-channel aligner looks for comma in the most significant byte only.

Typically, it is not necessary to set the DOWDALGN_xx bit. When the link state machine loses synchronization (DEMUXWAS_xx register bit is 0), the DEMUX block automatically looks for a new comma character irrespective of whether the DOWDALGN_xx bit is set or not. However, as discussed earlier, the comma character may become misaligned without the Fibre Channel link state machine indicating a loss of synchronization. In such cases, the DOWDALGN_xx bit must be toggled to force resynchronization.

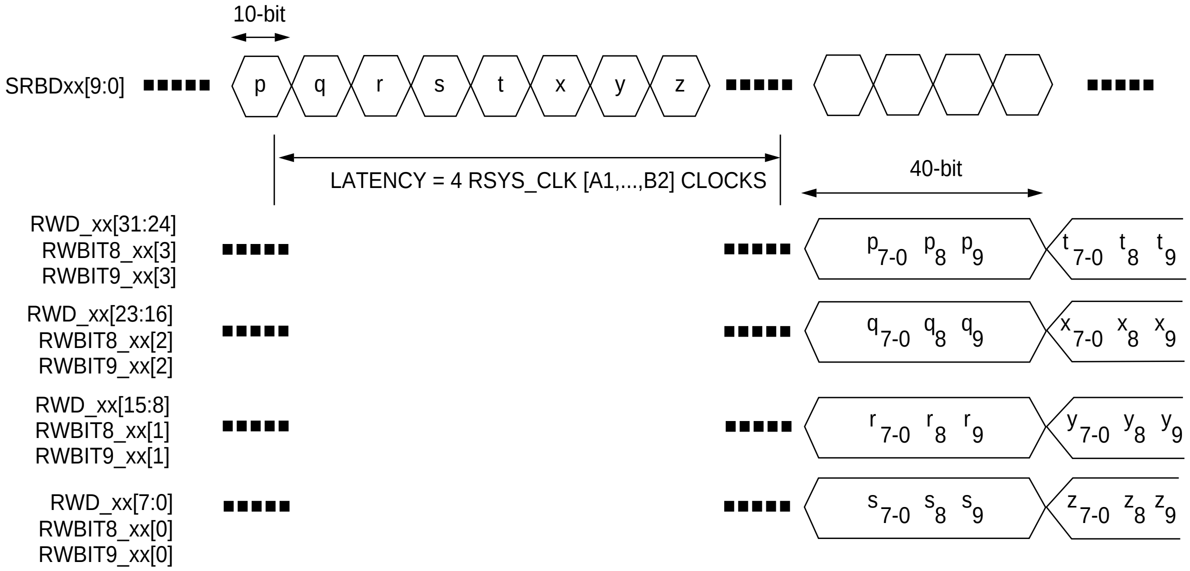
The NOWDALGN_xx bit is a level-sensitive bit. If it is a 1, then the DEMUX does not dynamically alter the word boundary based on comma and SWDSYNC_xx output of the SERDES. This might be useful if a channel were configured to bypass the multi-channel alignment FIFO and raw 40-bits of data are directed from SERDES to FPGA.

In Fibre Channel mode, the default setting (NOWDALGN_xx = 0) causes the word boundary to be set as soon as the SERDES SWDSYNC_xx output is a 1 and a comma character has been detected. The character that is the comma becomes the most-significant portion of the demultiplexed word. When the SERDES loses link synchronization it will drop SWDSYNC_xx low. The DEMUX will begin search for word alignment as soon as SWDSYNC_xx goes to 1 again.

The DEMUX passes on to the channel alignment FIFO block a set of control signals that indicate the location of the synchronizing event. RALIGN_xx[3:0] are these indicators. If there is no link synchronization, all of the RALIGN_xx[[3:0] bits will be zeros independent of synchronizing events that come in. When the link is synchronized, then the bit that corresponds to the time of the synchronization event will be set to a 1.

The relationship between a time sequence of values input at SRBDxx[7:0] to the values output at RWD_xx[39:0] is shown in Figure 8 below. A parallel relationship exists between SRBDx[8] and RWBIT8_xx[3:0] as well as between SRBD_xx[9] and RWBIT9_xx[3:0].

Figure 8. Receive DEMUX Block for a Single SERDES Channel



One clock per bank of 4 channels called RCK78[A,B] is sent to the FPGA. The control bits RCKSEL[A,B] are used to select the channel that is the source for these clocks.

Link State Machines

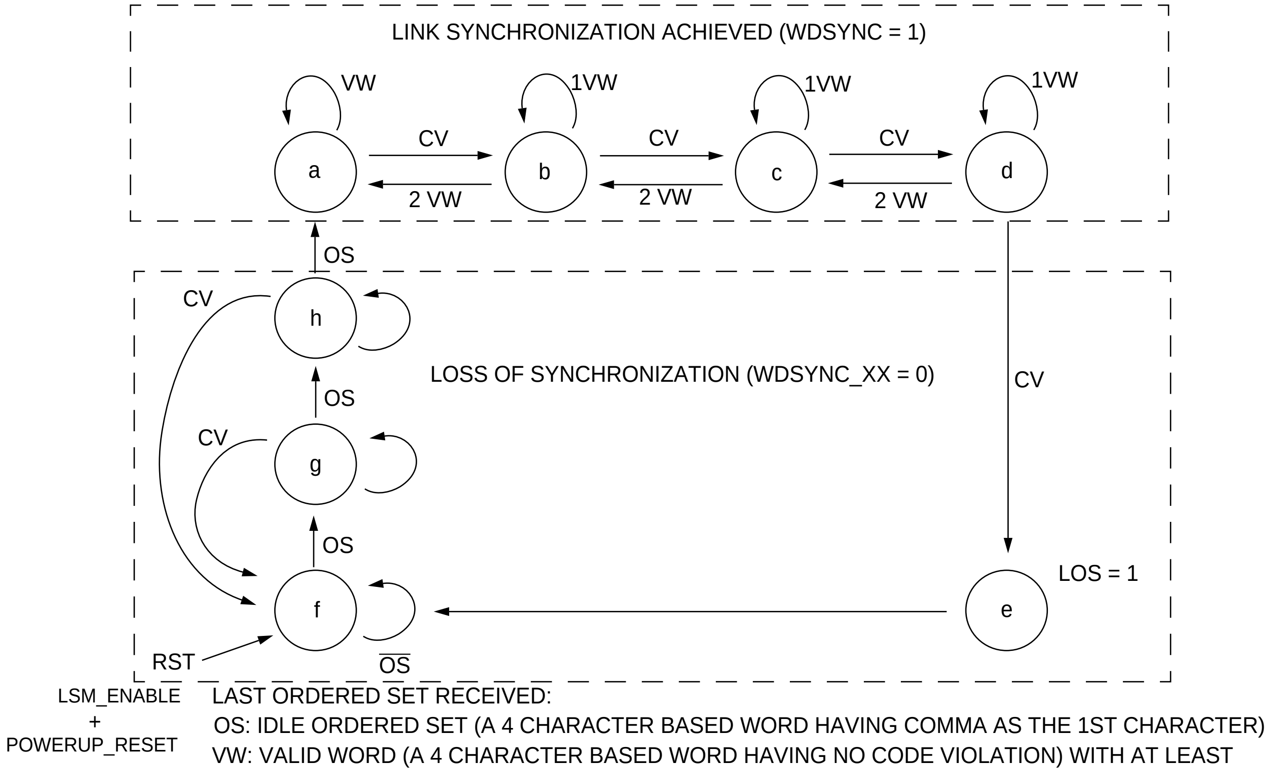
Two link state machines are included in the ORT82G5, one for XAUI applications and a second for Fibre Channel applications.

The Fibre Channel link state machine is responsible for establishing a valid link between the transmitter and the receiver and for maintaining link synchronization. The machine is initially in the Loss Of Synchronization (LOS) state upon power-on reset. This is indicated by WDSYNC_xx = 0. While in this state, the machine looks for a particular number of consecutive idle ordered sets without any invalid data transmission in between before declaring synchronization achieved. Achievement of synchronization is indicated by asserting WDSYNC_xx = 1. Specifically, the machine looks for three continuous idle ordered sets without any misaligned comma character or any running disparity based code violation in between. In the event of any such code violation, the machine would reset itself to the ground state and start its search for the idle ordered sets again. A typical valid sequence for achieving link synchronization would be K28.5 D21.4 D21.5 D21.5 repeated 3 times.

In the synchronization achieved state, the machine constantly monitors the received data and looks for any kind of code violation that might result due to running disparity errors. If it were to receive four such consecutive invalid words, the link machine loses its synchronization and once again enters the loss of synchronization state (LOS). A pair of valid words received by the machine overcomes the effect of a previously encountered code violation. LOS is indicated by the status of WDSYNC_xx output which now transitions from 1 to 0. At this point the machine attempts to establish the link yet again. Figure 9 shows the state diagram for the Fibre Channel link state machine.

In the ORT82G5 LOS is also indicated by DEMUXWAS_xx status register bit. This bit is set to 0 during loss of synchronization.

Figure 9. Fibre Channel Link State Machine State Diagram



LAST ORDERED SET RECEIVED:

OS: IDLE ORDERED SET (A 4 CHARACTER BASED WORD HAVING COMMA AS THE 1ST CHARACTER)

VW: VALID WORD (A 4 CHARACTER BASED WORD HAVING NO CODE VIOLATION) WITH AT LEAST TWO PRECEEDING VALID WORDS

CV: CODE VIOLATION (RUNNING DISPARITY BASED ON ILLEGAL COMMA POSITION)

1VW: FIRST VALID WORD AFTER A CODE VIOLATION

2VW: SECOND VALID WORD AFTER A CODE VIOLATION

XAUI Link Synchronization Function

For each lane, the receive section of the XAUI link state machine incorporates a synchronization state machine that monitors the status of the 10-bit alignment. A 10-bit alignment is done in the SERDES based on a comma character such as K28.5. A comma (0011111 or its complement 1100000) is a unique pattern in the 10-bit space that cannot appear across the boundary between any two valid 10-bit code-groups. This property makes the comma useful for delimiting code-groups in a serial stream.This mechanism incorporates a hysteresis to prevent false synchronization and loss of synchronization due to infrequent bit errors. For each lane, the sync_complete signal is disabled until the lane achieves synchronization. The synchronization state diagram is shown in Figure 10. This state machine is implemented as defined in draft *IEEE* 802.3ae, version 2.1 but will also operate properly with version 4.1 implementations. Table 4 and Table 5 describe the state variables used in Figure 10.

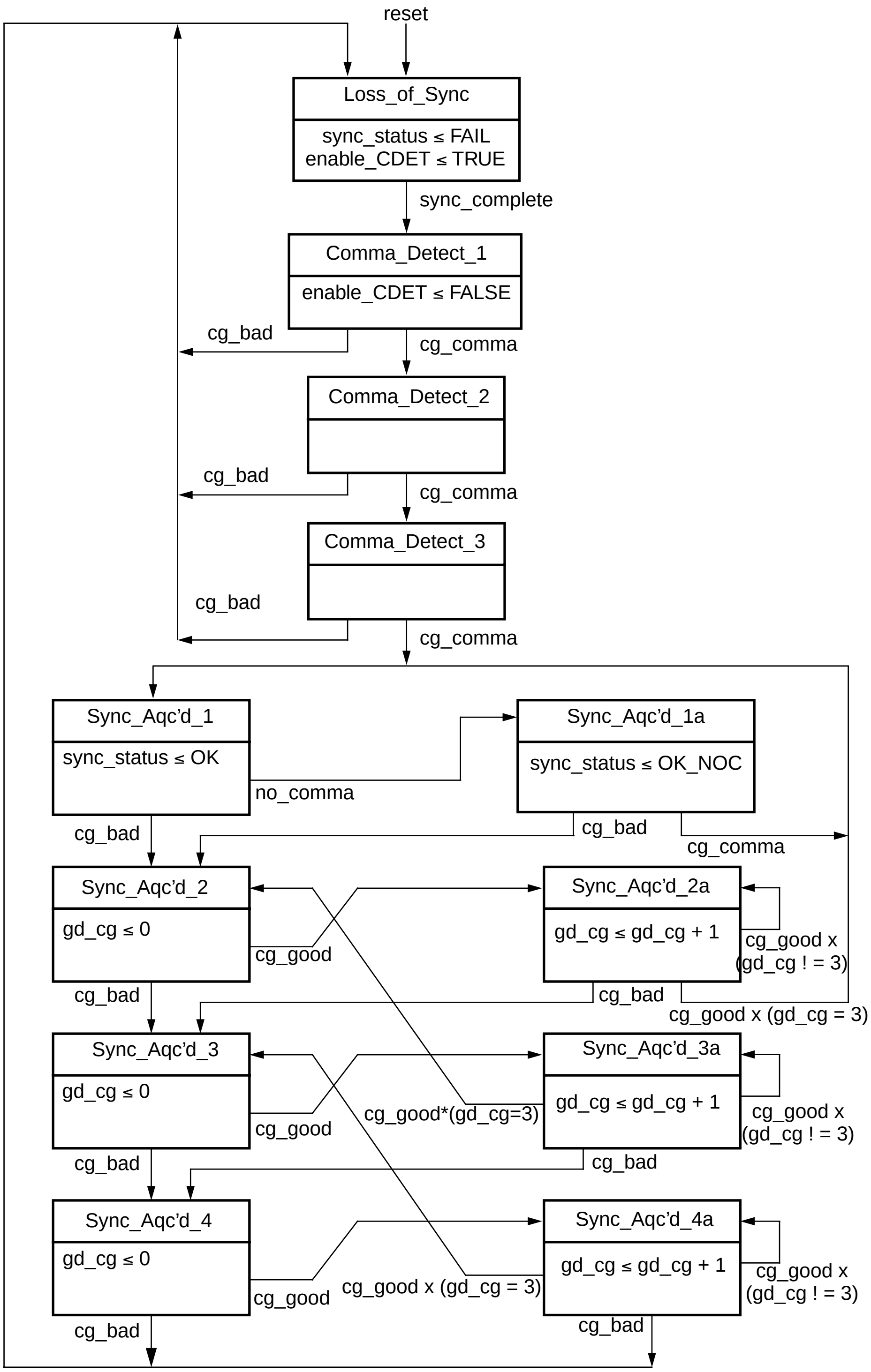
Table 4. XAUI Link Synchronization State Diagram – Functions

Function	Description
sync_complete	Indication that alignment code-group alignment has been established at the boundary indicated by the most recently received comma.
cg_comma	Indication that a valid code-group, with correct running disparity, containing a comma has been received.
cg_good	Indication that a valid code-group with the correct running disparity has been received.
cg_bad	Indication that an invalid code-group has been received.
no_comma	Indication that comma timer has expired. The timer is initialized upon receipt of a comma.

Table 5. XAUI Link Synchronization State Diagram Notation – Variables

Variable	Description
sync_status	FAIL: Lane is not synchronized (correct 10-bit alignment has not been established). OK: Lane is synchronized. OK_NOC: Lane is synchronized but a comma character has not been detected in the past TBD seconds.
enable_CDET	TRUE: Align subsequent 10-bit words to the boundary indicated by the next received comma. FALSE: Maintain current 10-bit alignment.
gd_cg	Current number of consecutive cg_good indications.

Figure 10. XAUI Link Synchronization State Diagram



Multichannel Alignment

The alignment FIFO allows the transfer of all data to the system clock. The Multi-Channel Alignment block (Figure 6) allows the system to be configured to allow the frame alignment of multiple slightly varying data streams. This optional alignment ensures that matching SERDES streams will arrive at the FPGA end in perfect data synchronization.

Each channel is provided with a 24 word x 36-bit FIFO. The FIFO can perform two tasks: (1) to change the clock domain from receive clock to a clock from the FPGA side, and (2) to align the receive data over 2, 4, or 8 channels. This FIFO allows a timing budget of ±230.4 ns that can be allocated to skew between the data lanes and for transfer to the system clock. The input to the FIFO consists of 36 bits of demultiplexed data, RALIGN_xx[3:0], RWD_xx[31:0], and RWBIT8_xx[3:0].

The four RALIGN_xx bits are control signals, and can be the alignment character detect signals indicating the presence of a comma character in Fibre Channel mode and the /A/ character in XAUI mode. The other 32 RWD_xx bits are the 8-bit data bytes from the 8b/10b decoder. The alignment character, if present, is the MSB of the data. The RWBIT8_xx indicates the presence of a Km.n control character in the receive data byte. Only RWBIT8_xx and RWD_xx inputs are stored in the FIFO. During alignment process, RALIGN[3]_xx is used to synchronize multiple channels.

If a channel is not in any alignment group, it will set the FIFO-write-address to the beginning of the FIFO, and will set the FIFO-read-address to the middle of the FIFO, at the first assertion of RALIGN[3]_xx after reset or after the resync command.

The RX_FIFO_MIN_xx register bits can be used to control the threshold for minimum unused buffer space in the alignment FIFOs between read and write pointers before overflow (OVFL) status is flagged. The synchronization algorithm consists of a down counter which starts to count down by 1 from its initial value of 18 (decimal) when an alignment character from any channel within an alignment group has been received. Once all the alignment characters within the alignment group have been received, the count is decremented by 2 until 0 is reached. Data is then read from the FIFOs and output to the FPGA. This algorithm is not repeated after multi-channel alignment has been achieved; resynchronization must be forced by toggling the appropriate FMPU_RESYNC bit.

The ORT82G5 has a total of 8 channels (4 per SERDES). The incoming data of these channels can be synchronized in several ways or they can be independent of one other. Two channels within a SERDES quad can be aligned together. Channel A and B and/or channel C and D can form a pair as shown in Figure 11. Alternately, all four channels of a SERDES quad can be aligned together to form a communication channel with a bandwidth of 10 Gbits/s as shown in Figure 12. Finally, the alignment can be extended across both SERDES quad to align all 8 channels in ORT82G5 as shown in Figure 13. Individual channels within an alignment group can be disabled (i.e., powered down) without disrupting other channels.

Figure 11. Twin Channel Alignment

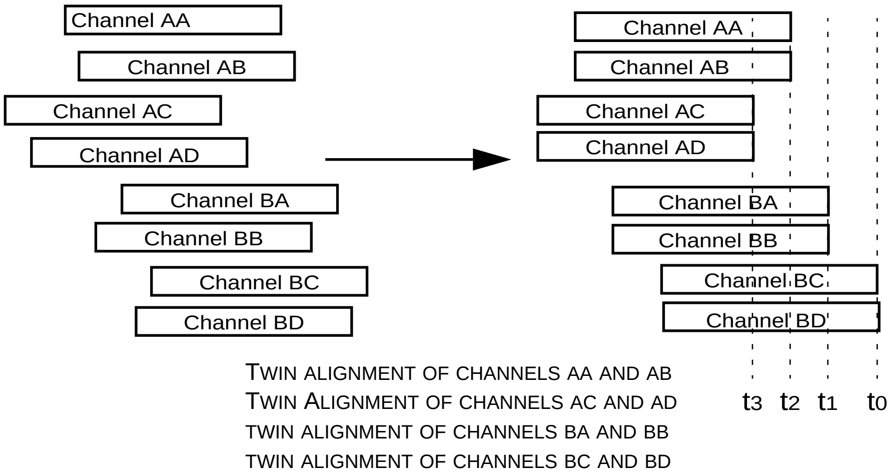


Figure 12. Alignment of SERDES Quads A and B

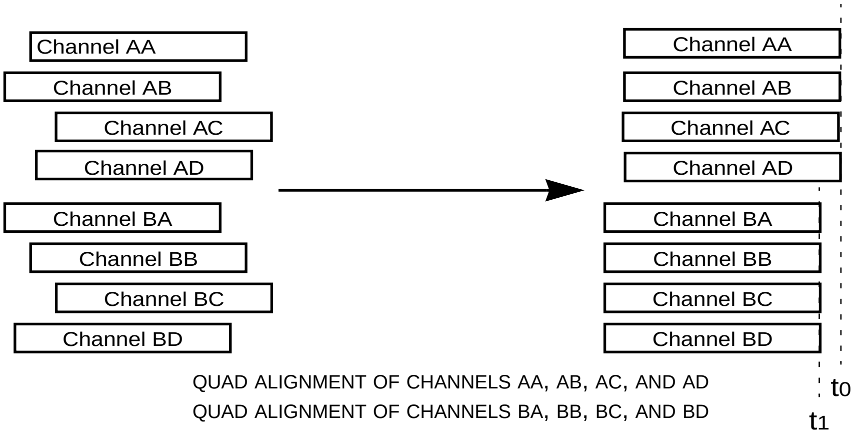
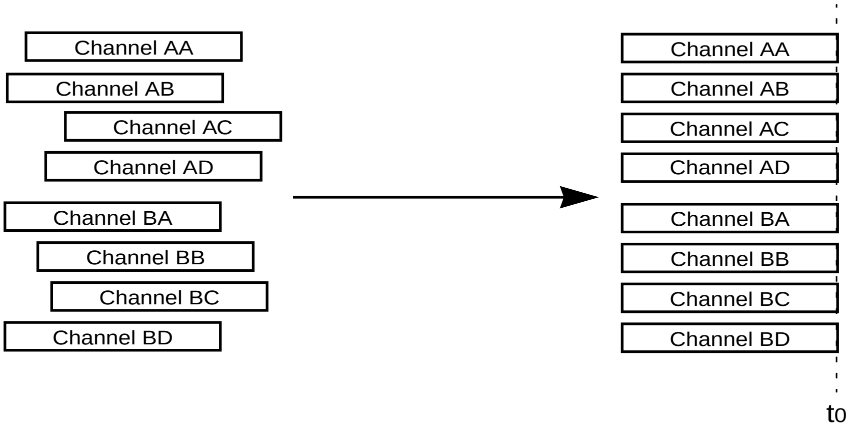


Figure 13. Alignment of all Eight SERDES Channels.



Note that any channel within an alignment group can be removed from that alignment group by setting FMPU_STR_EN_xx to 0. The disabling of any channel(s) within an alignment group will not affect the operation of the remaining active channels. If the active channels are synchronized, that synchronization will be maintained and no data loss will occur.

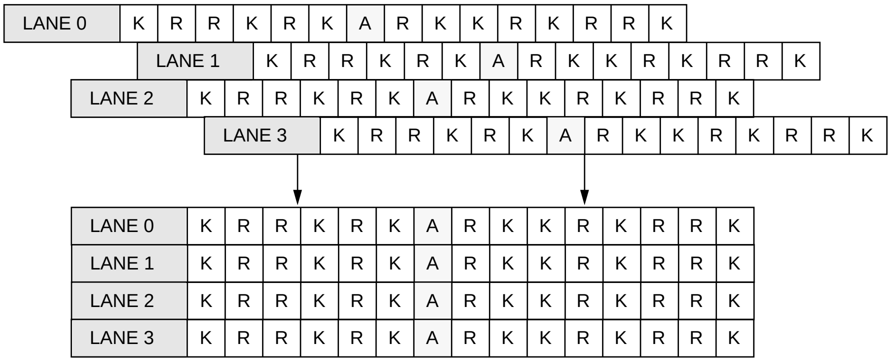
For every alignment group, there are both an OVFL and an OOS status register bit. The OVFL bit is set when alignment FIFO overflow occurs. The OOS bit is flagged when the down counter in the synchronization algorithm has reached a value of 0 and alignment characters from all channels within an alignment group have not been received. In the memory map section the bits indicating OOS and OVFL are referred to as SYNC2_[A1,A2,B1,B2]_OOS, SYNC4_[A:B]_OOS and SYNC8_OOS and the bits indicating OVFL are SYNC2_[A1,A2,B1,B2]_OVFL, SYNC4_[A:B]_OVFL and SYNC8_OVFL.

Alignment can also be done between the receive channels on two ORT82G5 devices. Each of the two devices needs to provide its aligned K_CTRL or other alignment character to the other device, which will delay reading from a second alignment FIFO until all channels requesting alignment on the current device and all channels requesting alignment on the other device are aligned (as indicated on the K_CTRL character). This second alignment FIFOs will be implemented in FPGA logic on the ORT82G5. This scheme also requires that the reference clock for both devices be driven by the same signal.

XAUI Lane Alignment Function (Lane Deskew)

In XAUI mode, the receive section in each lane uses the /A/ code group to compensate for lane-to-lane skew. The mechanism restores the timing relationship between the 4 lanes by lining up the /A/ characters into a column. Figure 14 shows the alignment of four lanes based on /A/ character. A minimum spacing of 16 code-groups implies that at least ± 80 bits of skew compensation capability should be provided, which the ORT82G5 significantly exceeds.

Figure 14. Deskew Lanes by Aligning /A/ Columns



LANE 0 K R R K R K A R K K R K R R K
LANE 1 K R R K R K A R K K R K R R K
LANE 2 K R R K R K A R K K R K R R K
LANE 3 K R R K R K A R K K R K R R K

LANE 0 K R R K R K A R K K R K R R K
LANE 1 K R R K R K A R K K R K R R K
LANE 2 K R R K R K A R K K R K R R K
LANE 3 K R R K R K A R K K R K R R K

Mixing Half-rate, Full-rate Modes

When channel alignment is enabled, all receive channels within an alignment group should be configured at the same rate. For example, channels AA, AB, can be configured for twin alignment and full-rate mode, while channels AC, AD that form an alignment group can be configured for half-rate mode. In quad alignment mode, each receive quad can be configured in either half or full-rate mode.

When channel alignment is disabled within a quad, any receive channel within the quad can be used in half-rate or full-rate mode. The clocking strategy for half-rate mode in both scenarios (channel alignment enabled or disabled) is described in the section *ORT82G5 Reference Clocks and Internal Clock Distribution* later in this data sheet.

Multi-channel alignment configuration

Register settings for multi-channel alignment are shown in Table 6.

Table 6. Multichannel Alignment Modes

Register Bits FMPU_SYNMODE_xx[0:1]	Mode
00	No multichannel alignment.
10	Twin channel alignment.
01	Quad channel alignment.
11	Eight channel alignment.

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

To align all eight channels:

- FMPU_SYNMODE_A[A:D] = 11
- FMPU_SYNMODE_B[A:D] = 11
- To align all four channels in SERDES A:
- FMPU_SYNMODE_A[A:D] = 01
- To align two channels in SERDES A:

- FMPU_SYNMODE_A[A:B] = 10 for channel AA and AB
- FMPU_SYNMODE_A[C:D] = 10 for channel AC and AD

A similar alignment can be defined for SERDES B.

To enable/disable synchronization signal of individual channel within a multi-channel alignment group:

- FMPU_STR_EN_xx = 1 enabled
- FMPU_STR_EN_xx = 0 disabled
  where xx is one of A[A:D] and B[A:D].
- To resynchronize a multichannel alignment group set the following bit to zero, and then set it to 1.
- FMPU_RESYNC8 for eight channel A[A:D] and B[A:D]
- FMPU_RESYNC4A for quad channel A[A:D]
- FMPU_RESYNC2A1 for twin channel A[A:B]
- FMPU_RESYNC2A2 for twin channel A[C:D]
- FMPU_RESYNC4B for quad channel B[A:D]
- FMPU_RESYNC2B1 for twin channel B[A:B]
- FMPU_RESYNC2B2 for twin channel B[C:D]

To resynchronize an independent channel (resetting the write and the read pointer of the FIFO) set the following bit to zero, and then set it to 1.

- FMPU_RESYNC1_xx

A two-to-one multiplexer is used to select between aligned or nonaligned data to be sent to the FPGA on MRWDxx[39:0]. The 40-bit MRWDxx[39:0] output format is shown in Table 4. and .

Alignment Sequence

1. Follow steps 1 and 2 in the start up sequence described in a later section.
2. Initiate a SERDES software reset by setting the SWRST bit to 1 and then to 0. Note that, any changes to the SERDES configuration bits should be followed by a software reset.
3. Wait for 3 ms. REFCLK should be toggling by this time. During this time, configure the following registers.

Set the following bits in registers 30820, 30920

- XAUI_MODE_xx-set to 1 for XAUI mode or keep the default value of 0.
- Enable channel alignment by setting FMPU_SYNMODE bits in registers 30811, 30911.
- FMPU_SYNMODE_xx. Set to appropriate values for 2, 4, or 8 alignment based on Table 6.
- Set RCLKSEL[A:B] and TCKSEL[A:B] bits in registers 30A00.
- RCKSEL[A:B]-choose clock source for 78 MHz RCK78x (Table 12).
- TCKSEL[A:B]-Choose clock source for 78 MHz TCK78x (Table 11).Send data on serial links. Monitor the following status/alarm bits:
- Monitor the following alarm bits in registers 30000, 30010, 30020, 30030, 30100, 30110, 30120, 30130.
- LKI-PLL_xx lock indicator. A 1 indicates that PLL has achieved lock.
- Monitor the following status bits in registers 30804, 30904

- XAUISTAT_xx - In XAUI mode, they should be 10.

Monitor the following status bits in registers 30805, 30905

- DEMUXWAS_xx-They should be 1 indicating word alignment is achieved.
- CH248_SYNCxx-They should be 1 indicating channel alignment. This is cleared by resync.

4. Write a 1 to the appropriate resync registers 30820, 30920. Note that this assumes that the previous value of the resync bits are 0. The resync operation requires a rising edge. Two writes are required to the resync bits: write a 0 and then write a 1.

Check out-of-sync and FIFO overflow status in registers 30814 (Bank A).

- SYNC4_A_OOS, SYNC4_A_OVFL-by 4 alignment.
- SYNC2_A2_OOS, SYNC_A2_OVFL or SYNC2_A1_OOS, SYNC2_A1_OVFL-by 2 alignment.
- Check out-of-sync status in registers 30914 (Bank B).
- SYNC4_B_OOS, SYNC4_B_OVFL-by 4 alignment.
- SYNC_B2_OOS, SYNC2_B2_OVFL or SYNC2_B1_OOS, SYNC_B1_OVFL-by 2 alignment.
- Check out-of-sync status in register 30A03
- SYNC8_OOS, SYNC8_OVFL-by 8 alignment.
- If out-of-sync bit is 1, then rewrite a 1 to the appropriate resync registers and monitor the OOS bit again. If Out of Synchronization (OOS) bit is 0 but OVFL bit is 1, then check if the RX_FIFO_MIN value has been programmed to a value > 0. (Default value is 0.) Change the value to 0 and check the OVFL bit again. If OOS and OVFL are 1, then rewrite a 1 to the appropriate resync registers. The resync operation requires a rising edge. Two writes are required to the resync bits: write a 0 and then write a 1.

Embedded Core/FPGA Interface

This block provides the data formatting and receive data and clock signal transfers between the Embedded Core and the FPGA Logic. There are also control and status registers in the FPGA portion of the chip which contain bits to control the receive logic and to record status. These are described in later sections of this data sheet, Table 8, and communicate with the core using the System Bus.

The demultiplexed, receive word outputs to the FPGA are shown in Figure 6. . These are each 40 bits wide. There are eight of these interfaces, one for each SERDES channel. Each consist of four groups of 10-bit data or four groups of decoded information depending on setting of 8b10bR_xx control register bits.

Each 10-bit group of decoded information includes 8 bits of data and a 1 bit K_CTRL indicator derived from the received data and a tenth bit of status information. The function of the tenth bit varies from group to group and includes code violation, Out of Synchronization (OOS) indicators and the CH248_SYNC_xx status bit. CH248_SYNC_xx indicates the status of multi-channel alignment of channel xx and is high when the count for the multi-channel alignment block reaches zero regardless of whether or not multi-channel alignment is successful. The mapping of the 10-bit groups to the MRWD_xx[39:0] bits output to the FPGA logic is summarized in Table 5 and Table . The various functions of the bits that vary from channel to channel, i.e., bits 29 and 19, are also described in Table .]

Table 7. Definition of Bits of MRWDxx[39:0]

Bit Index	8b10bR=0	8b10bR=1	
	NOCHALGN[A:B]=1 CV_SELxx=0	**NOCHALGN[A:B]=1 CV_SELxx=1**	**NOCHALGN[A:B]=0 CV_SELxx=1**
39	bit 9 of 10-bit data 3	CV_AA3, code violation, byte 3	CH248_SYNCxx
38	bit 8 of 10-bit data 3	K_CTRL for byte 3	K_CTRL for byte 3
37	bit 7 of 10-bit data 3	bit 7 of byte3	bit 7 of byte3
36	bit 6 of 10-bit data 3	bit 6 of byte 3	bit 6 of byte 3
35	bit 5 of 10-bit data 3	bit 5 of byte 3	bit 5 of byte 3
34	bit 4 of 10-bit data 3	bit 4 of byte 3	bit 4 of byte 3
33	bit 3 of 10-bit data 3	bit 3 of byte 3	bit 3 of byte 3
32	bit 2 of 10-bit data 3	bit 2 of byte 3	bit 2 of byte 3
31	bit 1of 10-bit data 3	bit 1 of byte 3	bit 1 of byte 3
30	bit 0of 10-bit data 3	bit 0 of byte 3	bit 0 of byte 3
29	bit 9 of 10-bit data 2	CV_AA2, code violation, byte 2	See
28	bit 8 of 10-bit data 2	K_CTRL for byte 2	K_CTRL for byte 2
27	bit 7 of 10-bit data 2	bit 7 of byte 2	bit 7 of byte 2
26	bit 6 of 10-bit data 2	bit 6 of byte 2	bit 6 of byte 2
25	bit 5 of 10-bit data 2	bit 5 of byte 2	bit 5 of byte 2
24	bit 4 of 10-bit data 2	bit 4 of byte 2	bit 4 of byte 2
23	bit 3 of 10-bit data 2	bit 3 of byte 2	bit 3 of byte 2
22	bit 2 of 10-bit data 2	bit 2 of byte 2	bit 2 of byte 2
21	bit 1of 10-bit data 2	bit 1 of byte 2	bit 1 of byte 2
20	bit 0 of 10-bit data 2	bit 0 of byte 2	bit 0 of byte 2
19	bit 9 of 10-bit data 1	CV_AA1, code violation, byte 1	See
18	bit 8 of 10-bit data 1	K_CTRL for byte 1	K_CTRL for byte 1
17	bit 7 of 10-bit data 1	bit 7 of byte 1	bit 7 of byte 1
16	bit 6 of 10-bit data 1	bit 6 of byte 1	bit 6 of byte 1
15	bit 5 of 10-bit data 1	bit 5 of byte 1	bit 5 of byte 1
14	bit 4 of 10-bit data 1	bit 4 of byte 1	bit 4 of byte 1
13	bit 3 of 10-bit data 1	bit 3 of byte 1	bit 3 of byte 1
12	bit 2 of 10-bit data 1	bit 2 of byte 1	bit 2 of byte 1
11	bit 1 of 10-bit data 1	bit 1 of byte 1	bit 1 of byte 1
10	bit 0 of 10-bit data 1	bit 0 of byte 1	bit 0 of byte 1
09	bit 9 of 10-bit data 0	CV_AA0, code violation, byte 1	VL (connected to ground)
08	bit 8 of 10-bit data 0	K_CTRL for byte 0	K_CTRL for byte 0
07	bit 7 of 10-bit data 0	bit 7 of byte 0	bit 7 of byte 0
06	bit 6 of 10-bit data 0	bit 6 of byte 0	bit 6 of byte 0
05	bit 5 of 10-bit data 0	bit 5 of byte 0	bit 5 of byte 0
04	bit 4 of 10-bit data 0	bit 4 of byte 0	bit 4 of byte 0
03	bit 3 of 10-bit data 0	bit 3 of byte 0	bit 3 of byte 0
02	bit 2 of 10-bit data 0	bit 2 of byte 0	bit 2 of byte 0
01	bit 1 of 10-bit data 0	bit 1 of byte 0	bit 1 of byte 0
00	bit 0o f 10-bit data 0	bit 0 of byte 0	bit 0 of byte 0

Table 8. Definition of the Status Bits of MRWDxx that Vary for Different Channels

Channel Index	Bit Index	Name	Description
all	39	CH248_SYNCxx	Multi-channel alignment attempt complete if 1
AA	29	CV_AA_OR	Code violation in one or more of the received 10-bit groups for channel AB
AA	19	SYNC2_A1_OOS	Dual channel synchronization of channels AA and AB not successful if 1
AB	29	CV_AB_OR	Code violation in one or more of the received 10-bit groups for channel AB
AB	19	SYNC4_A_OOS	Quad channel synchronization of SERDES quad A not successful if 1
AC	29	CV_AC_OR	Code violation in one or more of the received 10-bit groups for channel AC
AC	19	SYNC2_A2_OOS	Dual channel synchronization of channels AC and AD not successful if 1
AD	29	CV_AD_OR	Code violation in one or more of the received 10-bit groups for channel AD
AD	19	SYNC8_OOS	Eight channel synchronization not successful if 1
BA	29	CV_BA_OR	Code violation in one or more of the received 10-bit groups for channel BA
BA	19	SYNC2_B1_OOS	Dual channel synchronization of channels BA and BB not successful if 1
BB	29	CV_BB_OR	Code violation in one or more of the received 10-bit groups for channel BB
BB	19	SYNC4_B_OOS	Quad channel synchronization of SERDES quad A not successful if 1
BC	29	CV_BC_OR	Code violation in one or more of the received 10-bit groups for channel BD
BC	19	SYNC2_B2_OOS	Dual channel synchronization of channels BC and BD not successful if 1
BD	29	CV_BD_OR	Code violation in one or more of the received 10-bit groups for channel BD
BD	19	SYNC8_OOS	Eight channel synchronization not successful if 1

The code violation signals will only be valid if the corresponding CV_SELxx = 1. (If 8b10bR=0, CV_SEL should also be zero. The CV_xx_OR signals are obtained by ORing four code violation signals from the 1:4 DEMUX block. These are primarily indicators of received signal quality since a single code violation will not force a loss of sync (LOS) state in the word alignment state machines. Since these signals come from the DEMUX block, if multi-channel alignment is enabled, the code violation signals correspond to data that must still be multi-channel aligned. Hence these signals provide advance notification of detected violations in data that will appear at the core/FPGA interface several clock cycles later. The exact number of clock cycles that the data is delayed depends on the skew between the incoming data for the different channels.

The SYNC2_[A1,A2,B1,B2]_OOS, SYNC4_[A:B]_OOS,and SYNC8_OOS signals can be used with CH248_SYNC_xx to determine if the desired multi-channel alignment was successful. If, when CH248_SYNC_xx goes high the corresponding OOS signal remains low, the data being transferred across the core/FPGA interface is correctly aligned between channels. Note that only the signals corresponding to the selected alignment mode will be meaningful.

Transceiver FPGA/Embedded Core Signals

Table 9 summarizes the interface signals between the FPGA logic and the core. In the table, an input refers to a signal flowing into the embedded core and an output refers to a signal flowing out of the embedded core.

Table 9. Transceiver Embedded Core/FPGA Interface Signal Description

FPGA/Embedded Core Interface Signal Name (xx = [AA, ... ,BD])	Input (I) to or Output (O) from Core	Signal Description
Transmit Path Signals		
TWDxx[31:0]	I	Transmit data—channel xx.
TCOMMAxx[3:0]	I	Transmit comma character—channel xx.
TBIT9xx[3:0]	I	Transmit force negative disparity—channel xx
TSYS_CLK_xx	I	Transmit low-speed clock to the FPGA—channel xx
TCK78[A:B]	O	Transmit low-speed clock to the FPGA—SERDES Quad [A:B].
Receive Path Signals		
MRWDxx[39:0]	O	Receive data—Channel xx (see).
RWCKxx	O	Low-speed receive clock—Channel xx.
RCK78[A:B]	O	Receive low-speed clock to FPGA—SERDES Quad [A:B].
RSYS_CLK_A1	I	Low-speed receive FIFO clock for channels AA, AB
RSYS_CLK_A2	I	Low-speed receive FIFO clock for channels AC, AD
RSYS_CLK_B1	I	Low-speed receive FIFO clock for channels BA, BB
RSYS_CLK_B2	I	Low-speed receive FIFO clock for channels BC, BD
CV_SELxx		Enable detection of code violations in the incoming data
SYS_RST_N	I	Synchronous reset of the channel alignment blocks.

ORT82G5 Reference Clocks and Internal Clock Distribution

Reference Clock Requirements

There are two pairs of reference clock inputs on the ORT82G5. The differential reference clock is distributed to all four channels in a quad. Each channel has a differential buffer to isolate the clock from the other channels. The input clock is preferably a differential signal; however, the device can operate with a single-ended input. The input reference clock directly impacts the transmit data eye, so the clock should have low jitter. In particular, jitter components in the DC to 5 MHz range should be minimized. The required electrical characteristics for the reference clock are given in Table 24.

Note: In sections of this data sheet, the differential clocks are simply referred to as the reference clock as REFCLK_[A:B]. REFCLK_[A:B] is equivalent to REFINP_[A:B] and REFINN_[A:B].

Synthesized and Recovered Clocks

The SERDES Embedded Core block contains its own dedicated PLLs for transmit and receive clock generation. The user provides a reference clock of the appropriate frequency, as described in the previous section. The transmitter PLL uses the REFCLK_[A,B] inputs to synthesize the internal high-speed serial bit clocks. The receiver PLLs extract the clock from the serial input data and retime the data with the recovered clock.

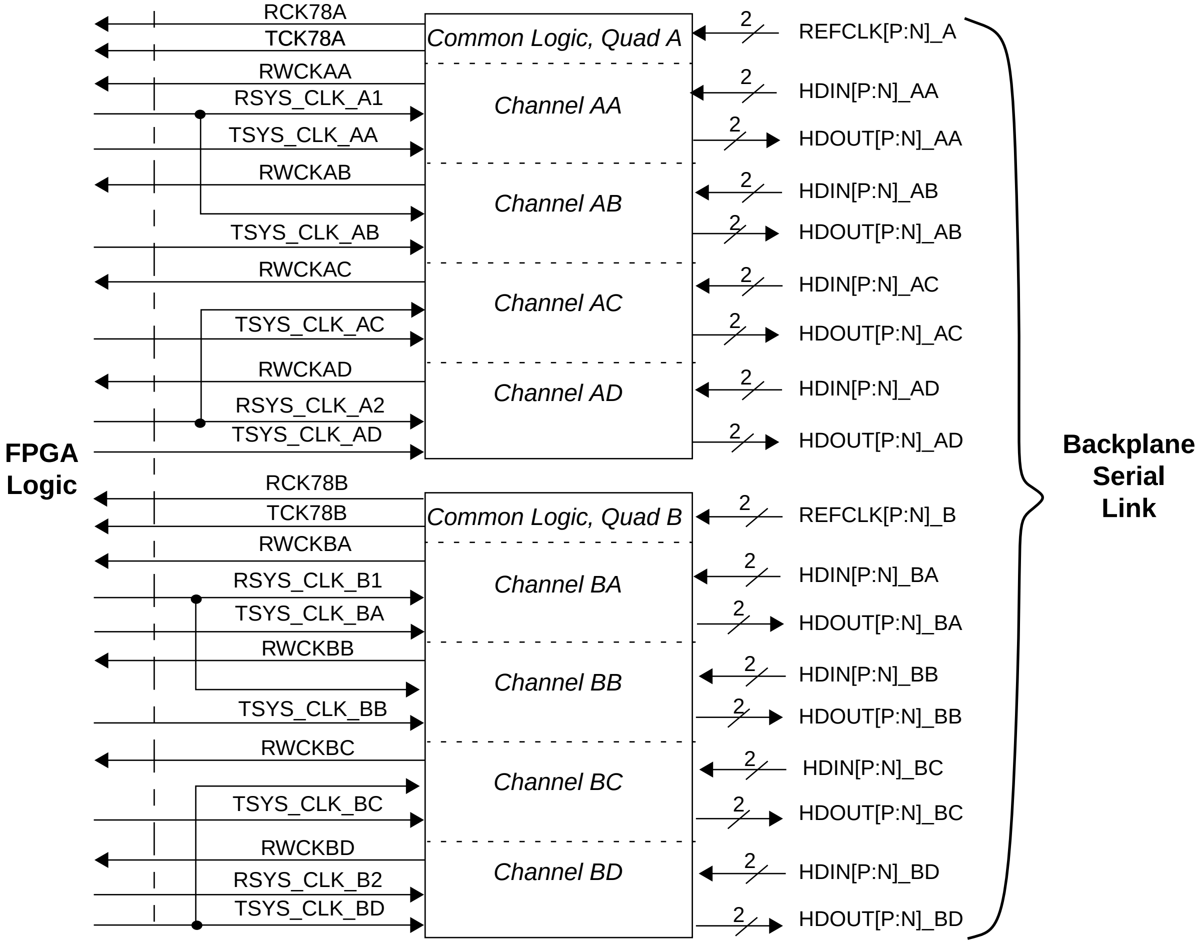
The receive PLL for each channel has two modes of operation - lock to reference and lock to data with retiming. When no data or invalid data is present on the HDINP_xx and HDINN_xx pins, the receive VCO will not lock to data and its frequency can drift outside of the nominal ±100 ppm range. Under this condition, the receive PLL will lock to REFCLK_[A,B] for a fixed time interval and then will attempt to lock to receive data. The process of attempting to lock to data, then locking to clock will repeat until valid input data exists. There is also a control register bit per channel to force the receive PLL to always lock to the reference clock.

The high-speed transmit and receive serial data links can run at 1.0 to 3.7 Gbits/s, depending on the frequency of the reference clock and the state of the control bits from the internal transmit control register. The interface to the serializer/deserializer block runs at 1/10th the bit rate of the data lane. Additionally, the MUX/DEMUX logic converts the data rate and bit-width so the FPGA core can run at 1/4th this frequency which gives a range of 25 to 92.5 MHz for the data in and out of the FPGA.

Internal Clock Signals at the FPGA/Core Interface

There are several clock signals defined at the FPGA/Embedded Core interface in addition to the external reference clock for each SERDES quad. All of the ORT82G5 clock signals are shown in Figure 15 and are described following the figure.

Figure 15. ORT82G5 Clock Signals (High Speed Serial I/O Also Shown)



REFCLKP_[A:B], REFCLKN_[A:B]:
These are the differential reference clocks provided to the ORT82G5 device as described earlier. They are used as the reference clock for both TX and RX paths. For operation of the serial links at 3.125 Gbits/s, the reference clocks will be at a frequency of 156.25 MHz.

RWCK[AA:BD]:
These are the low-speed receive clocks from the embedded core to the FPGA across the core-FPGA interface. These are derived from the recovered low-speed complementary clocks from the SERDES blocks. RWCK_AA belongs to Channel AA, RWCK_AB belongs to channel AB and so on. With a reference clock input of 156.25 MHz, these clocks operate at 78.125 MHz.

RCK78[A:B]:
These are muxed outputs of RWCKA[A:D] and RWCKB[B:D] respectively. With a reference clock input of 156.25 MHz, these clocks operate at 78.125 MHz.

RSYS_CLK_[A:B][1:2]

These clocks are inputs to the SERDES quad block A and B respectively from the FPGA. These are used by each channel as the read clock to read received data from the alignment FIFO within the embedded core. Clocks RSYS_CLK_A[1:2] are used by channels in the SERDES quad block A and RSYS_CLK_B[1:2] by channels in the SERDES quad block B. To guarantee that there is no overflow in the alignment FIFO, it is an absolute requirement that the write and read clocks be frequency locked within 0 ppm. Examples of how to achieve this are shown in the later section on recommended board-level clocking.

TCK78[A:B]:

This is a muxed output from the core to the FPGA across the core-FPGA interface of one of the 4 transmit SERDES clocks per quad operating at up to 92.5 MHz in the embedded core. There is one clock output per SERDES quad block.

TSYS_CLK[AA,…BD]:

These clocks are inputs to the SERDES quad block A and B respectively from the FPGA. These are used by each channel to control the timing of the Transmit Data Path. To guarantee correct transmit operation theses clocks must be frequency locked within 0 ppm to TCK78[A:B].

Transmit and Receive Clock Rates

Table 10. shows typical relationship between the data rates, the reference clock, the transmit TCK78[A:B] clock and the receive RCK78[A:B] clock. The selection of full-rate or half-rate for a given reference clock speed is set by bits in the transmit and receive control registers and can be set per channel.

Table 10. Transmit Data and Clock Rates

Data Rate	Reference Clock	TCK78[A: B] and RCK78[A:B] Clocks	Rate of Channel Selected as Clock Source
1.0 Gbits/s	100 MHz	25 MHz	Half
1.25 Gbits/s	125 MHz	31.25 MHz	Half
2.0 Gbits/s	100 MHz	50 MHz	Full
2.5 Gbits/s	125 MHz	62.5 MHz	Full
3.125 Gbits/s	156 MHz	78 MHz	Full
3.7 Gbits/s	185 MHz	92.5 MHz	Full

Besides taking in a TSYS_CLK_xx from the FPGA logic for each channel, the transmit path logic sends back a clock of the same frequency, but arbitrary phase. This clock, TCK78[A:B], is derived from the MUX block of one of the 4 channels in its SERDES quad. The MUX blocks provide the potential source for TCK78[A:B] by a divide-by-4 of the SERDES STBC311xs clock used in synchronizing the transmit data words in the STBC311xx clock domain. The STBC311xx clocks are internal to the core and are not brought across the core/FPGA interface

The receiver section receives high-speed serial data at its differential CML input port and sends in to the Clock and Data Recovery (CDR) block. The CDR block then generates a recovered clock (RWCKxx) and retimes the data. Thus, the recovered receive clocks are asynchronous between channels.

Transmit Clock Source Selection

The TCKSEL[0:1][A:B] bits select the source channel of TCK78[A:B]. The selection of the source for TCK78[A:B] is controlled by these bits as shown in Table 11.

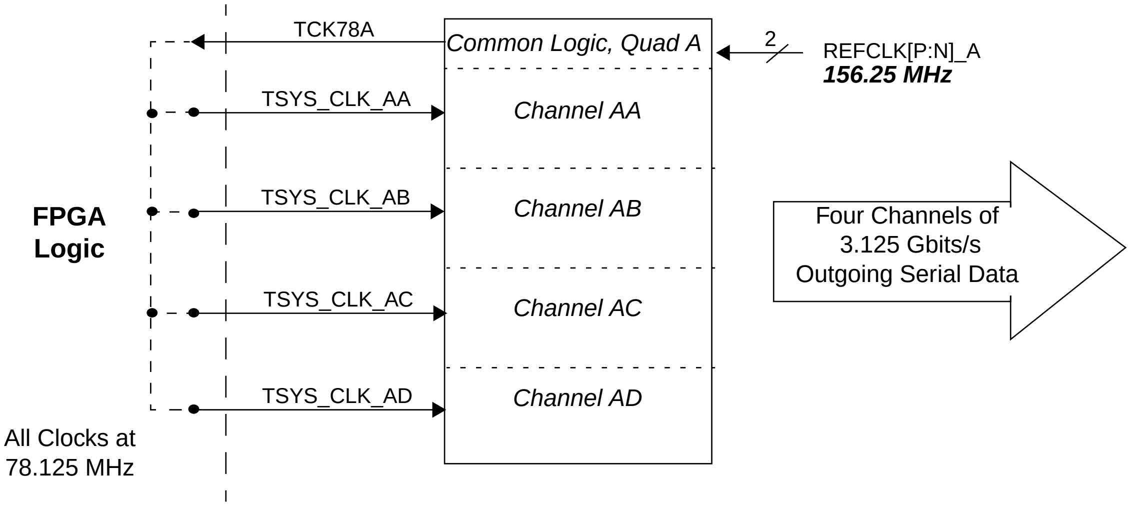
Table 11. TCK78[A:B] Source Selection

TCKSEL0	TCKSEL1	Clock Source
0	0	Channel A
1	0	Channel B
0	1	Channel C
1	1	Channel C

Recommended Transmit Clock Distribution

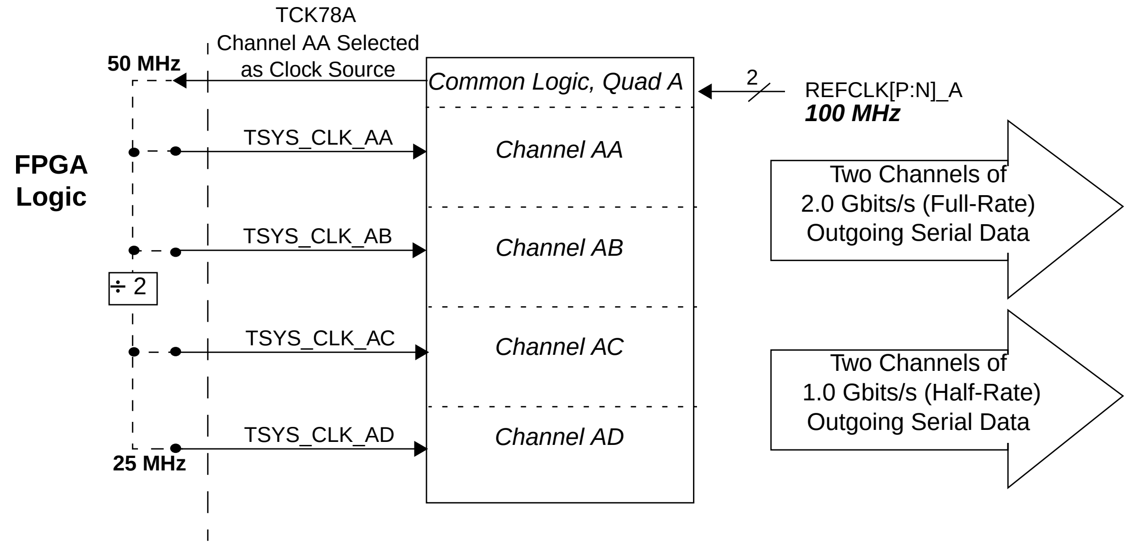
As an example of the recommended clock distribution approach, TSYS_CLK_A[A:D] can be sourced by TCK78A as shown in Figure 16 if the transmit line rate are common for all 4 channels in a quad. Similar clocking would be used for Quad B.

Figure 16. Transmit Clocking for a Single Quad (Similar Connections Would Be Used for Quad B)



If the transmit line rate is mixed between half and full rate among the channels, then the scheme shown in Figure 17 can be used. The figure shows TSYS_CLK_AA and TSYS_CLK_AB being sourced by TCK78A and TSYS_CLK_AC and TSYS_CLK_AD being sourced by TCK78A/2 (the division is done in FPGA logic). Similar clocking would be used for Quad B.

Figure 17. Mixed Rate Transmit Clocking for a Single Quad (Similar Connections Would Be Used for Quad B)



Receive Clock Source Selection and Recommended Clock Distribution

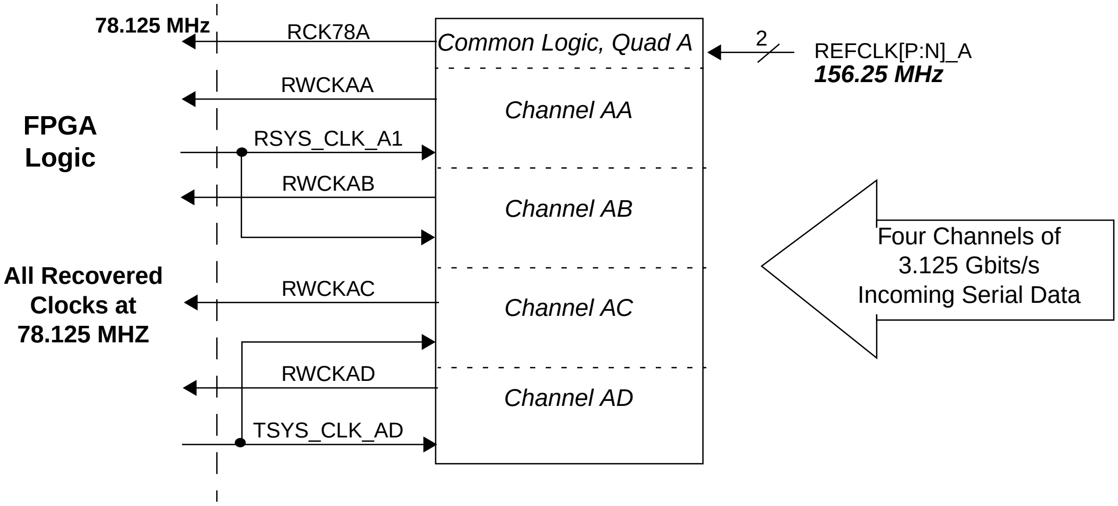
In the receive path, one clock per bank of 4 channels, called RCK78[A:B], is sent to the FPGA logic. The control register bits RCKSEL[0:1][A:B] are used to select the clock source for these clocks. The selection of the source for RCK78[A:B] is controlled by these bits as shown in Table 12.

Table 12. RCK78[A:B] Source Selection

RCKSEL0	RCKSEL1	Clock Source
0	0	Channel A
1	0	Channel B
0	1	Channel C
1	1	Channel C

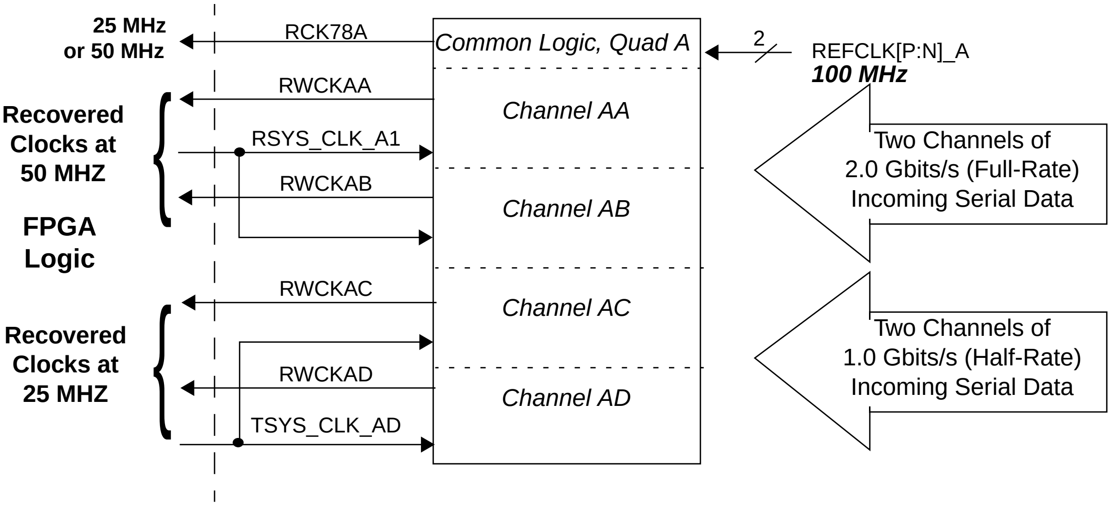
In the receive channel alignment bypass mode the data and recovered clocks for the 8 channels (4 per SERDES quad) are independent. The data for each channel are synchronized to the recovered clock from that channel.

Figure 18. - Receive Clocking for a Single Quad (Similar Connections Would Be Used for Quad B)



The receive channel alignment bypass mode allows mixing of half and full line rates among the channels, as shown in Figure 19. The figure shows channel pair AA and AB configured in full rate mode at 2.0 Gbits/s. Channel pair AC and AD are configured in half-rate mode at 1.0 Gbits/s.

Figure 19. Receive Clocking for Mixed Line Rates



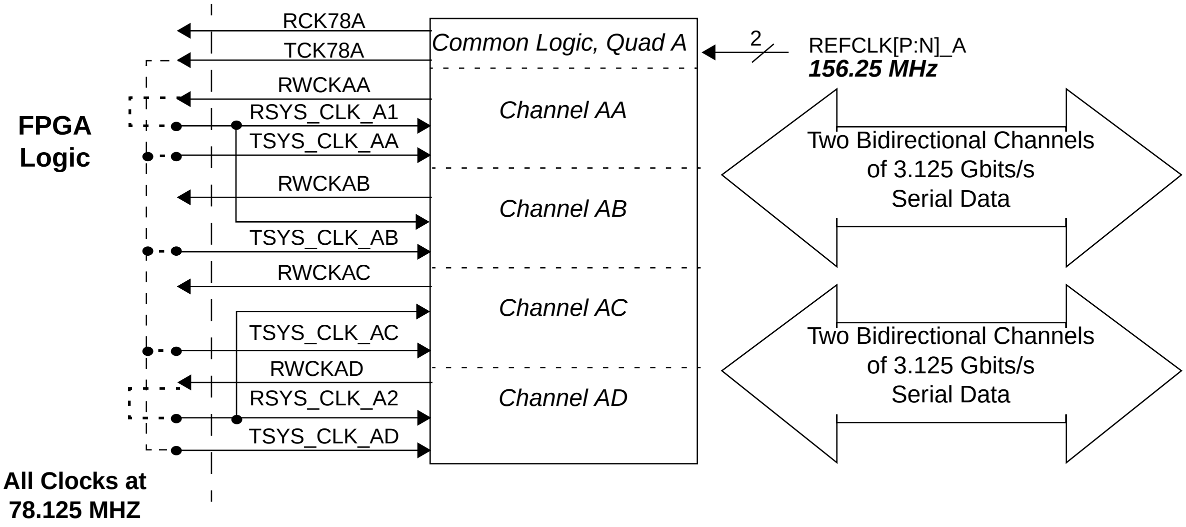
As noted in the caption of Figure 19, each quad can be configured in any line rate (1.0-3.7 Gbits/s), since each quad has its own reference clock input pins. The receive alignment FIFO per channel cannot be used in this mode.

Multi-Channel Alignment Clocking Strategies

The data on the 8 channels (4 per SERDES quad) in the ORT82G5 can be independent of each other or can be synchronized in several ways. For example, two channels within a SERDES can be aligned together; channel A and B and/or channel C and D. Alternatively, all four channels in a SERDES quad can be aligned together to form a communication channel with a bandwidth of 10 Gbits/s. Finally, the alignment can be extended across both SERDES quads to align all 8 channels. Individual channels within an alignment group can be disabled (i.e., powered down) without disrupting other channels. Clocking strategies for these various modes are described in the following paragraphs.

For dual alignment both twins within a quad can be sourced by clocks that are different from the other channels, however each pair of SERDES must have the same clock. The channel pair AA and AB is driven on the low speed side by RSYS_CLK_A1 and the channel pair AC and AD are driven on the low speed side by RSYS_CLK_A2. Either RWCKAA or RWCKAB can be connected to RSYS_CLK_A1 and either RWCKAC or RWCKAD can be connected to RSYS_CLK_A2. A clocking example for dual alignment is shown in Figure 20.

Figure 20. Receive Clocking for a Dual Alignment in a Single Quad (Similar Connections Would Be Used for Quad B)



For receive quad alignment, RSYS_CLK_[A1,B1] and RSYS_CLK_[A2,B2] can be tied together as shown for quad A and B in Figure 21. In receive eight-channel alignment, either RCK78A or RCK78B can be used to source RSYS_CLK_[A1,A2] and RSYS_CLK_[B1,B2] as shown in Figure 21.

Figure 21. Clocking for Quad Alignment in a Single Quad (Similar Connections Would Be Used for Quad B)

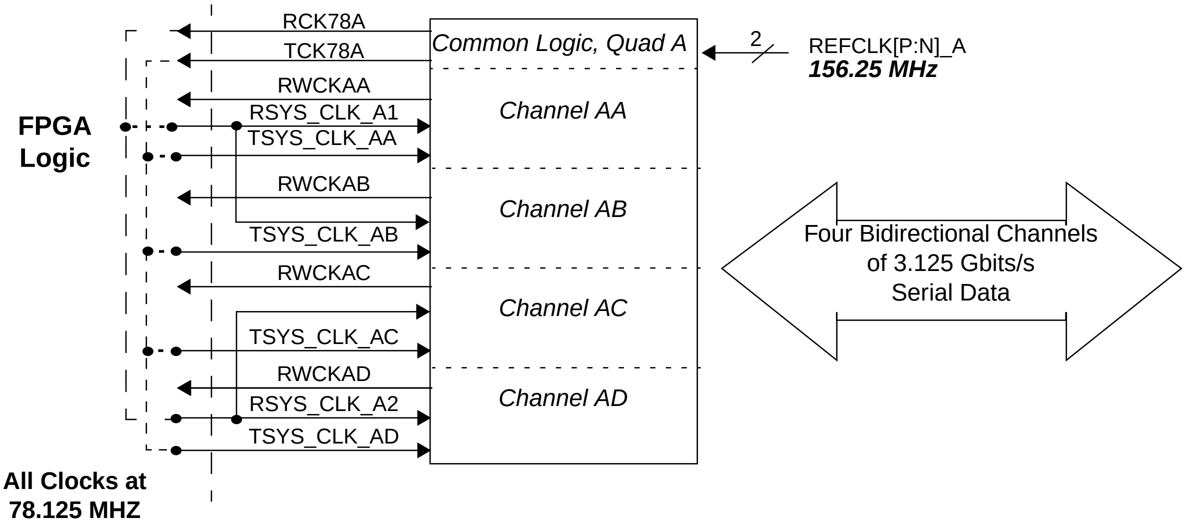
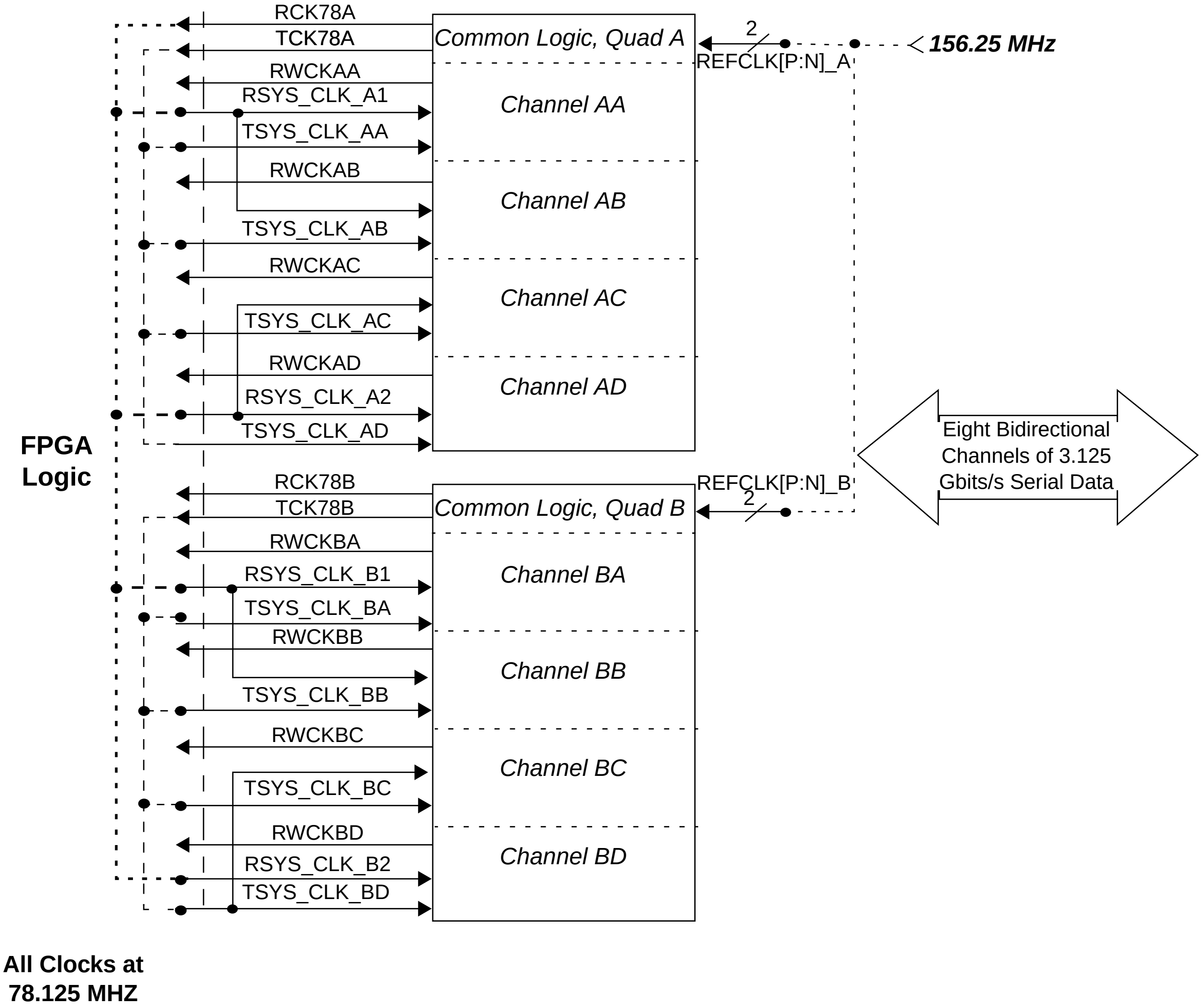


Figure 22. Clocking for Eight Channel Alignment



Reset Operation

The SERDES block can be reset in one of three different ways as follows: on power up, using the hardware reset, or via the microprocessor interface. The power up reset process begins when the power supply voltage ramps up to approximately 80% of the nominal value of 1.5 V. Following this event, the device will be ready for normal operation after 3 ms.

A hardware reset is initiated by making the PASB_RESETN low for at least two microprocessor clock cycles. The device will be ready for operation 3 ms after the low to high transition of the PASB_RESETN. This reset function affects all SERDES channels and resets all microprocessor and internal registers and counters.

Using the software reset option, each channel can be individually reset by setting SWRST (bit 2) to a logic 1 in the channel configuration register. The device will be ready 3 ms after the SWRST bit is deasserted. Similarly, all four channels per quad SERDES can be reset by setting the global reset bit GSWRST. The device will be ready for normal operation 3 ms after the GSWRST bit is deasserted. Note that the software reset option resets only SERDES internal registers and counters. The microprocessor registers are not affected. It should also be noted that the embedded block cannot be accessed until after FPGA configuration is complete.

Start Up Sequence

The following sequence is required by the ORT82G5 device. For information required for simulation that may be different than this sequence, see the ORT82G5 design kit.

1. Initiate a hardware reset by making PASB_RESETN low. Keep this low during FPGA configuration of the device. The device will be ready for operation 3 ms after the low to high transition of PASB_RESETN.

2. Configure the following SERDES internal and external registers. Note that after device initialization, all alarm and status bits should be read once to clear them. A subsequent read will provide the valid state. Set the following bits in register 30800:

   – Bits LCKREFN_[AD:AA] to 1, which implies lock to data.
   – Bits ENBYSYNC_[AD:AA] to 1 which enables dynamic alignment to comma.

   Set the following bits in register 30801:
   – Bits LOOPENB_[AD:AA] to 1 if high-speed serial loopback is desired.

   Set the following bits in register 30900:
   – Bits LCKREFN_[BD:BA] to 1 which implies lock to data.
   – Bits ENBYSYNC_[BD:BA] to 1 which enables dynamic alignment to comma.

   Set the following bits in register 30901:
   – Bits LOOPENB_[BD:BA] to 1 if high-speed serial loopback is desired.

   Set the following bits in registers 30002, 30012, 30022, 30032, 30102, 30112, 30122, 30132:
   – TXHR set to 1 if TX half-rate is desired.
   – 8B10BT set to 1

   Set the following bits in registers 30003, 30013, 30023, 30033, 30103, 30113, 30123, 30133:
   – RXHR Set to 1 if RX half-rate is desired.
   – 8B10BR set to 1.

   Assert GSWRST bit by writing two 1's. Deassert GSWRST bit by writing two 0's.
   Wait 3ms. If higher speed serial loopback has been selected, the receive PLLs will use this time to lock to the new serial data.

   Monitor the following alarm bits in registers 30000, 30010, 30020, 30030, 30110, 30120, 30130:
   – LKI-PLL lock indicator. 1 indicates that PLL has achieved lock.

3. If 8b/10b mode is enabled, enable link synchronization by sending the following sequence three times:
   – K28.5 D21.4 D21.5 D21.5

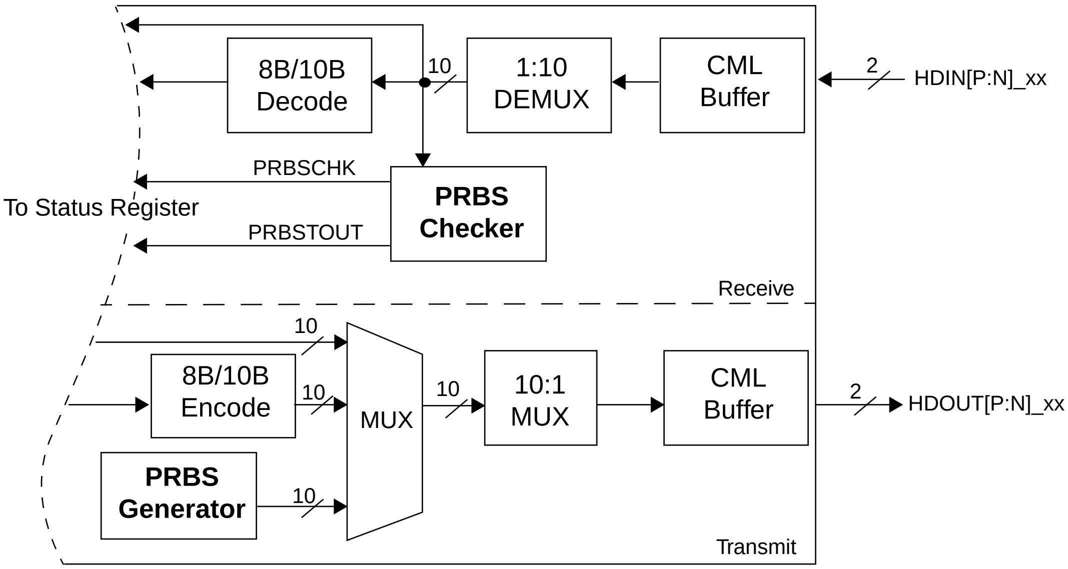
Test Modes

In addition to the operational logic described in the preceding sections, the Embedded Core contains logic to support various test modes - both for device validation and evaluation and for operating system level tests. The following sections discuss three of the test support logic blocks - the Pseudo-Random Bit Sequence (PRBS) generators and checkers, support for various loopback modes and support for SERDES characterization.

PRBS Generators and Checker Logic

The PRBS logic includes a stand-alone PRBS generator in the transmit path and a PRBS checker in the receive path, as shown in Figure 23 and is used in loopback testing. There is one generator and one checker for each SERDES channel. The transmit data is generated by a Linear Feedback Shift Register (LFSR) and thus is independent of the transmit data inputs to the Embedded core. If the PRBS logic is enabled (by setting the PRBS_xx bits in the control registers), the data is routed to the serializer and output buffer through a multiplexer.

Figure 23. Pseudo-Random Bit Sequence (PRBS) Generate and Check Logic



The receiver first deserializes the incoming serial data to regenerate the transmitted 10-bit word. The PRBS checker on the receiver compares the regenerated 10-bit word to an internally calculated 10-bit word on a word by word basis and signals a mismatch by asserting a PRBSCHK alarm status bit.

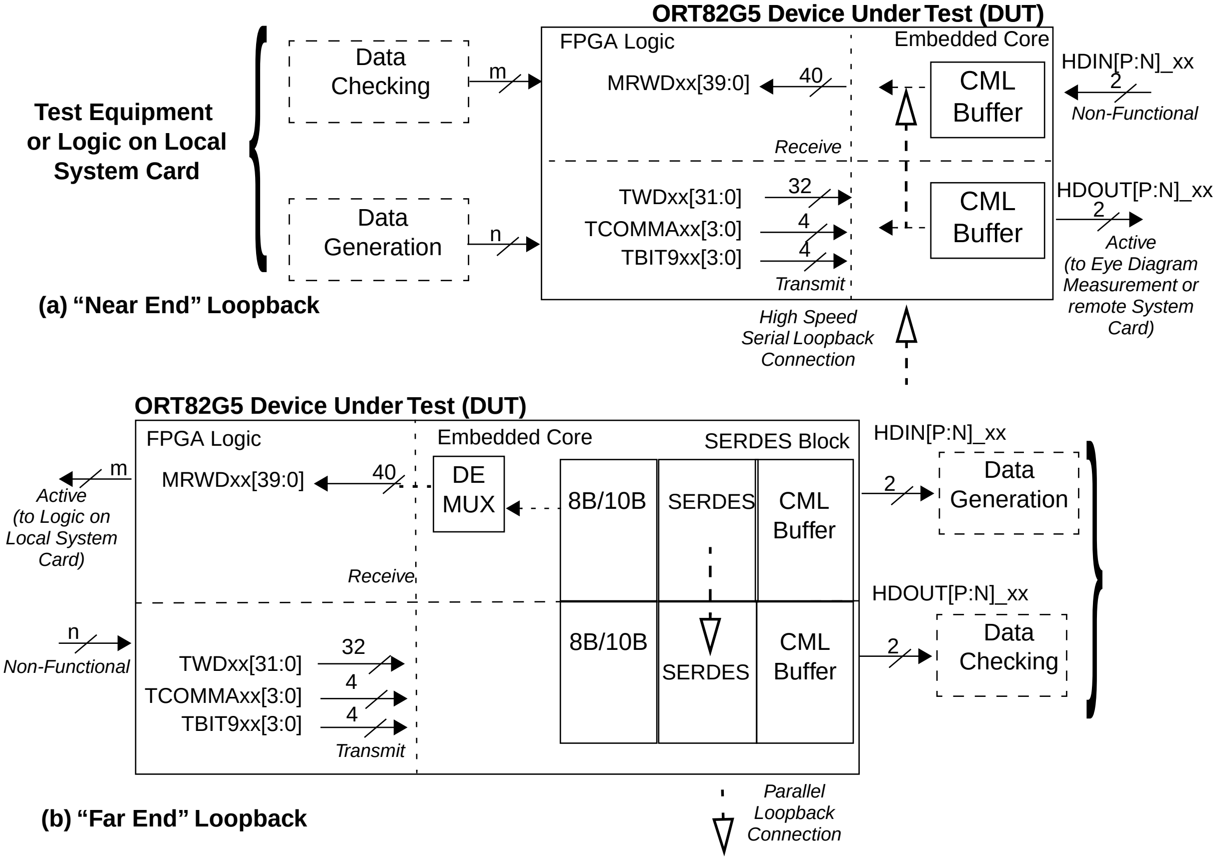
The PRBS checker also contains a watchdog timer which asserts the time-out alarm status bit, PRBSTOUT_xx, if the PRBS test cannot progress beyond its start state within a reasonable time interval. Both the PRBSCHK_xx and the PRBSTOUT_xx alarms can generate an interrupt if their corresponding masks are disabled (MPRBSCHK_xx = MPRBSTOUT_xx = 0).

Loopback Testing

Loopback testing is performed by looping back (either internal to the Embedded Core, by configuring the FPGA logic or by external connections) transmitted data to the corresponding receiver inputs, or received data to the transmitter output. The loopback path may be either serial or parallel.

In general, loopback tests can be classified as “near end” or “far end.” In “near end” loopback (Figure 24(a)), data is generated and checked locally, i.e. by logic on, or connection of, test equipment to the same card as the FPSC. In “far end” loopback (Figure 24(b)), the generating and checking functions are performed remotely, either by test equipment or a remote system card.

Figure 24. “Near End” vs. “Far End” Loopback



The loopback mode can also be characterized by the physical location of the loopback connection. There are four possible loopback modes supported by the Embedded Core logic:

- High-speed serial loopback at the CML buffer interface (near end)
- Parallel loopback at the SERDES boundary (far end)
- Parallel loopback at MUX/DEMUX boundary excluding SERDES (near end)
- Loopback testing using the PRBS generator/checker

The four loopback modes are described in more detail in the following sections. As noted earlier, other specialized loopback modes can be obtained by configuration of the FPGA logic or by connections external to the FPSC.

High-Speed Serial Loopback at the CML Buffer Interface

The high-speed serial loopback mode has the serial transmit signals looped back internally to the serial receive circuitry. The internal loopback path is from the input connection to the transmit CML buffer to the output connection from the receive CML buffer. The data are sourced on the TWDxx[31:0], TCOMMAxx[3:0] and TBIT9xx[3:0] signal lines and received on the MRWDxx[39:0] signal lines. The serial loopback path does not include the high-speed input and output buffers. If TESTEN_xx is set, the HDOUTP_xx and HDOUTN_xx outputs are active in this mode while the CML input buffers are powered down. The device is otherwise in its normal mode of operation. This mode is normally used for tests where the data source and destination are on the same card and is the basic loopback path shown earlier in Figure 24(a).

The data rate selection bits, TXHR and RXHR, in the channel configuration registers must be configured to carry the same value and the PRBS Generator and Checker are excluded by setting the PRBS configuration bit to 0. The 8b/10b encoder/decoder is not in the loopback path. Table 13 summarizes the settings of the control interface register configuration bits for high-speed serial loopback.

Table 13. High-Speed Serial Loopback Configuration Bit Definitions

Register Address	Bit Value	Bit Name	Comments
30002, 30012, 30022, 30032, 30102, 30112, 30122, 30132	Bit 0 = 0 or 1	TXHR	Set to 0 or 1. TXHR and RXHR bits must be set to the same value.
30002, 30012, 30022, 30032, 30102, 30112, 30122, 30132	Bit 7 = 0 or 1	8B10BT	Set to 0 or 1. If set to 0, the 8b/10b encoder is excluded from the loopback path. The 8b/10b encoder and decoder selection control bits must both be set to the same value.
30003, 30013, 30023, 30033, 30103, 30113, 30123, 30133	Bit 0 = 0 or 1	RXHR	Set to 0 or 1. TXHR and RXHR bits must be set to the same value.
30003, 30013, 30023, 30033, 30103, 30113, 30123, 30133	Bit 3 = 0 or 1	8B10BR	Set to 0 or 1. If set to 0, the 8b/10b decoder is excluded from the loopback path. The 8b/10b encoder and decoder selection control bits must both be set to the same value.
30004, 30014, 30024, 30034, 30104, 30114, 30124, 30134	Bit 0 = 0	PRBS	Set to 0.
30801, 30901	Bit 0 =1 (Channel A) Bit 1 = 1 (Channel B) Bit 2 = 1 (Channel C) Bit 3 = 1 (Channel D)	LOOPENB_xx	Set any of the bits 0-3 to 1 to do serial loopback on the corresponding channel.* The high speed serial outputs will not be active.

*This test mode can also be set using TESTEN_xx in place of LOOPENB_xx. In that case, Test Mode must be set to 00000.

Parallel Loopback at the SERDES Boundary

In this parallel loopback differential data are received at the HDINP_xx and HDINN_xx pins and are retransmitted at the HDOUTP_xx and HDOUTN_xx pins. The loopback path is at the interface between the SERDES blocks and the MUX and DEMUX blocks and uses the parallel 10-bit buses at these interfaces (see Figure 25 and Figure 26). The loopback connection is made such that the input signals to the TX SERDES block is the same as the output signals from the RX SERDES block. In this parallel loopback mode, the MRWDxx[39:0] signal lines remain active and the TWDxx[31:0], TCOMMAxx[3:0] and TBIT9xx[3:0] signal lines are not used. This mode is normally used for tests where serial test data is received from and transmitted to either test equipment or via a serial backplane to a remote card and is the basic loopback path shown earlier in Figure 24(b).

The data rate selection bits TXHR and RXHR in the channel configuration registers must be configured to carry the same value and the PRBS generator and checker are excluded by setting the PRBS configuration bit to 0. Also, the 8b/10b encoder and decoder are excluded from the loopback path by setting the 8b10bT and 8b10bR configuration bits to 0. Table 14 illustrates the control interface register configuration for the parallel loopback.

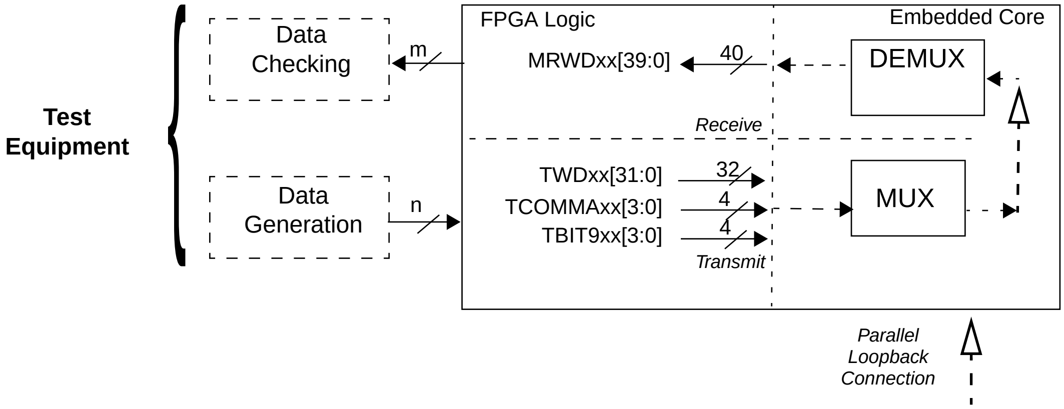
Table 14. Parallel Loopback at the SERDES Boundary Configuration Bit Definitions

Register Address (Hex)	Bit Value	Bit Name	Comments
300002	Bit 0 = 0 or 1	TXHR	Set to 0 or 1. TXHR and RXHR bits must be set to the same value.
30002	Bit 7 = 0	8b10bT	Set to 0 The 8b/10b encoder is excluded from the loopback path. The 8b/10b encoder and decoder selection control bits must both be set to 0.
30003	Bit 0 = 0 or 1	RXHR	Set to 0 or 1. TXHR and RXHR bits must be set to the same value.
30003	Bit 3 = 0	8b10bR	Set to 0.The 8b/10b decoder is excluded from the loopback path. The 8b/10b encoder and decoder selection control bits must both be set to 0.
30004	Bit 0 = 0	PRBS	Set to 0.
30005	Bit 7 = 1	GTESTEN	SET to 1 if the loopback is Done Globally on All Four Channels
30006	Bits[4:0]	Testmode	Set to 00001

Parallel Loopback at MUX/DEMUX Boundary, Excluding SERDES

This is a low-frequency test mode used to test the MUX/DEMUX logic block. As with the mode described in the previous section, the loopback path is at the interface between the SERDES blocks and the MUX and DEMUX blocks and uses the parallel 10-bit buses at these interfaces (see Figure 25 and Figure 26). However, the loopback connection is made such that the output signals from the TX MUX block are used as the input signals to the RX SERDES block. In this loopback mode the MRWDxx[39:0], TWDxx[31:0], TCOMMAxx[3:0] and TBIT9xx[3:0] signal lines function normally and the high-speed serial input and output buffers are not used. Use of this mode also requires configuration of the FPGA logic to connect the MRWDxx[39:0], TWDxx[31:0], TCOMMAxx[3:0] and TBIT9xx[3:0] signal lines to external pins. The basic loopback path is shown in Figure 25.

Figure 25. Parallel Loopback at MUX/DEMUX Boundary, Excluding SERDES



This test mode is enabled by setting the pin PLOOP_TEST_ENN to 1. PASB_TESTCLK must be running in this mode at 4x frequency of RSYS_CLK[A1,A2,B1,B2] or TSYS_CLK_[AA, AB . . . BD].

Loopback Testing Using The Pseudo-Random Bit Stream (PRBS) Generator/Checker

The PRBS logic can be tested using loopback. The loopback can be either internal to the device or external to it. To enable PRBS testing using internal loopback, use the following sequence:

1. To preform test with internal loopback, set the TESTEN_xx bit to 1 (registers 30801, 30901). Alternately, a LOOPENB_xx bit may be set to 1 and Test Mode set to 00000.
2. Set ENBSYNC register bit(s) to 1, depending on the channel(s) being tested (registers 30800, 30900).
3. Lock receiver to data by setting LCKREFN register bits to 1 (registers 30800, 30900).

4. Enable PRBS by setting PRBS_xx register bits (30004,30014, 30024, 30034, 30104, 30114, 30124, 30134). Alternately, the GPRBS_[A,B] bits can be used to enable PRBS test for all 4 SERDES channels within a bank (registers 30005, 30105).

5. Assert GSWRST_[A,B] bits (registers 30005, 30105) by writing two ones to reset the transmit and receive logic. Then clear the reset bits by writing two 0s.

6. Monitor the PRBSCHK_xx and PRBSTOUT_xx (30000,30010, 30020, 30030, 30100, 30120, 30130, 30140) alarm bits.

SERDES Characterization Test Mode

The SERDES characterization mode is a test mode that allows for direct control and observation of the transmit and receive SERDES interfaces at chip ports. With these modes the SERDES logic and I/O can be tested one channel at a time in either the receive or transmit modes. The SERDES characterization mode is available for only one quad (quad B) of the ORT82G5.

The characterization test mode is configured by setting bits in the control registers via the system bus. There are 4 bits that set up the test mode. The transmit characterization test mode is entered when SCHAR_ENA=1 and SCHAR_TXSEL=1. Entering this mode will cause chip port inputs to directly control the SERDES low-speed transmit ports of one of the channels as shown in Table 15.

Table 15. SERDES Transmit Characterization Mode

Chip Port	SERDES Input
PSCHAR_CKIO0	TBCBx
PSCHAR_LDIO[9:0]	LDINBx[9:0]

The x in the table will be a single channel in SERDES quad B, selected by the SCHAR_CHAN control bits. The decoding of SCHAR_CHAN is shown in Table 16.

Table 16. Decoding of SCHAR_CHAN

SCHAR_CHAN0	SCHAR_CHAN1	Channel
0	0	BA
1	0	BB
0	1	BC
1	1	BD

The receive characterization test mode is entered when SCHAR_ENA=1 and SCHAR_TXSEL=0, In this mode, one of the channels of SERDES outputs is observed at chip ports as shown in Table 17. The channel that is observed is also based on the decoding of SCHAR_CHAN as shown in Table 17.

Table 17. SERDES Receive Characterization Mode

SERDES Output	Chip Port
BYTSYNCBx	PSCHAR_BYTSYNC
WDSYNCBx	PSCHAR_WDSYNC
CVOBx	PSCHAR_CV
LDOUTBx[9:0]	PSCHAR_LDIO[9:0]
RBC0Bx	PSCHAR_CKIO0
RBC1Bx	PSCHAR_CKIO1

Embedded Core Block RAM

There are two independent memory slices (labeled A and B)in the embedded core. Each memory slice has a capacity of 4K words by 36 bits. These are in addition to the block RAMs found in the FPGA portion of the ORT82G5. Although the memory slices are in the embedded core part of the chip, they do not interact with the rest of the embedded core circuits, but are standalone memories designed specifically to increase RAM capacity in the ORT82G5 chip. They can be used by the soft IP cores implemented in the FPGA portion of the FPSC.

A block diagram of a memory slice is shown in Figure 26. Each memory slice is organized into two sections (labeled SRAM A and SRAM B) and has one read port, one write port and four byte-write-enable (active-low) signals. Each byte has eight data bits and a control/parity bit. The control/parity bit responds to the same byte enable (BYTEWN_x[x]) as it's corresponding data. No special logic such as parity checking is performed on this bit by the core. The read data from the memory is registered so that it works as a pipelined synchronous memory block. The minimum timing specifications are shown in Figure 27 and Figure 28. Signal names and functions are summarized later in Table 18 and follow the general Series 4 naming conventions.

Figure 26. Block Diagram, Embedded Core Memory Slice

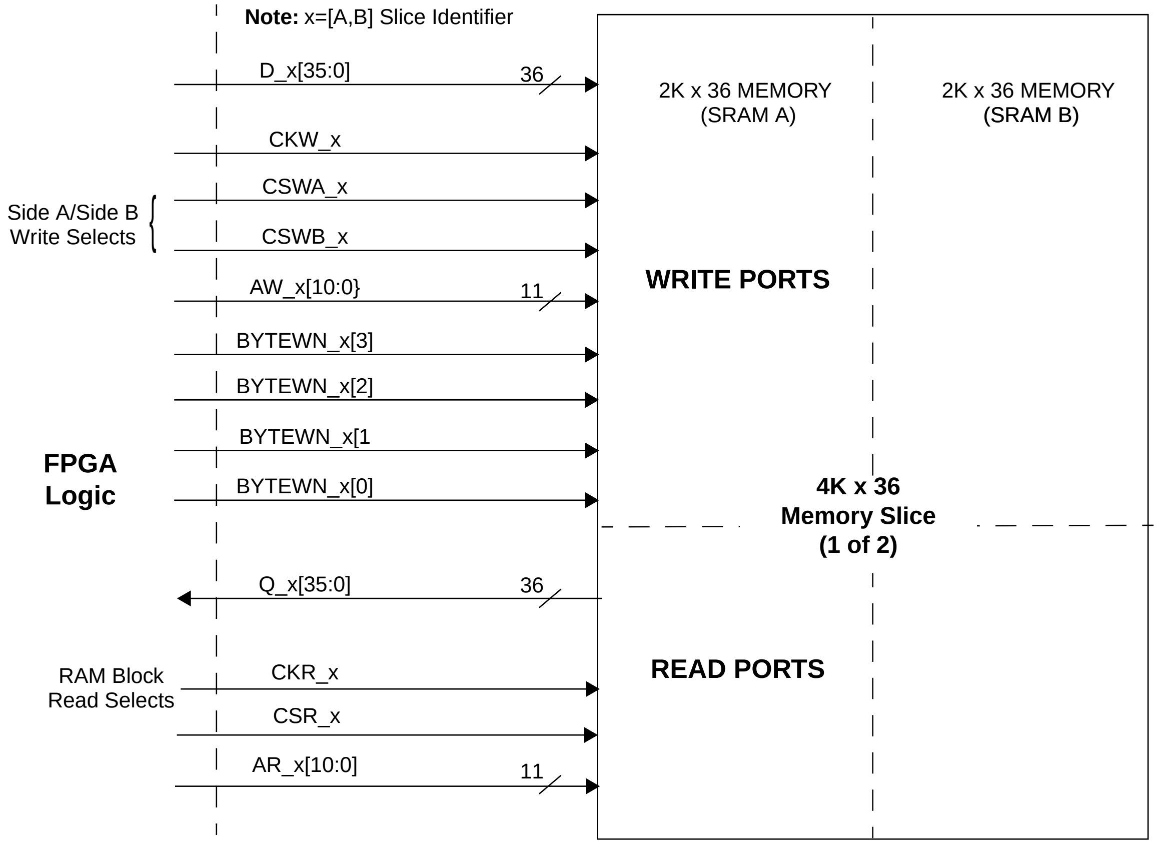


Figure 27. Minimum Timing Specs for Memory Blocks-Write Cycle (-1 Speed Grade)

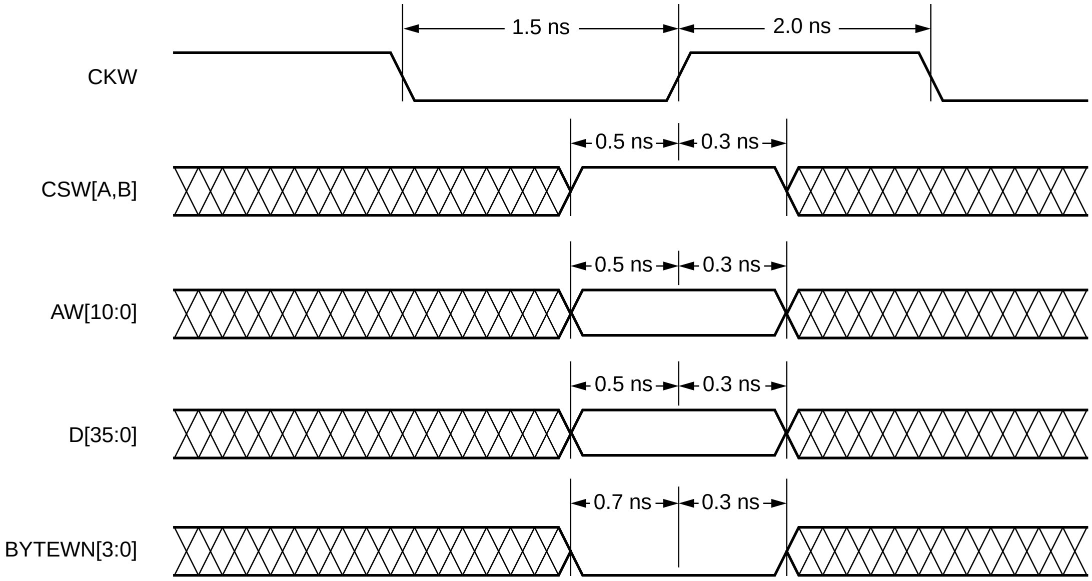
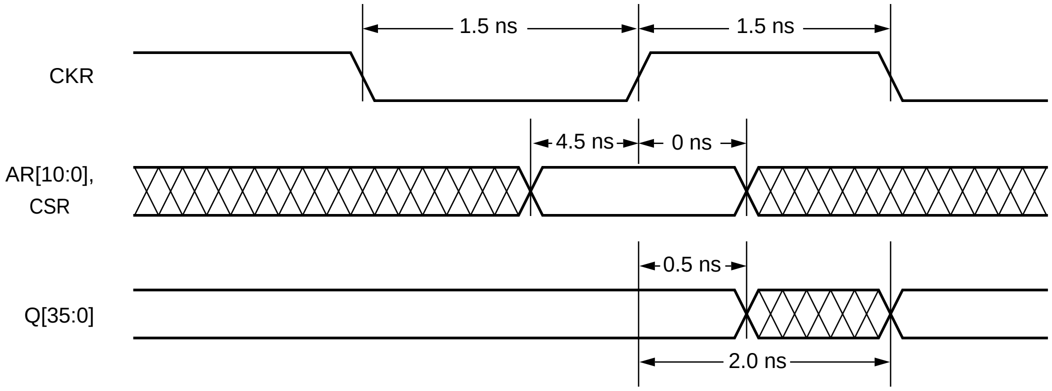


Figure 28. Minimum Timing Specs for Memory Blocks-Read Cycle (-1 Speed Grade)



In Table 18, an input refers to a signal flowing into the embedded core and an output refers to a signal flowing out of the embedded core.

Table 18. Embedded Memory Slice Core/FPGA Interface Signal Description

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

Memory Map

Definition of Register Types

The registers in ORT82G5 are 8-bit memory locations, which in general can be classified into the following types: Status Register and Control Register.

Status Register
Read-only register to convey the status information of various operations within the FPSC core. An example is the state of the XAUI link-state-machine.

Control Register
Read-write register to set up the control inputs that define the operation of the FPSC core.

The SERDES block within the ORT82G5 core has a set of status and control registers for its operation. There is another group of status and control registers which are implemented outside the SERDES, which are related to the SERDES and other functional blocks in the FPSC core. Reserved addresses for register blocks are shown in Table 19. The registers will be described in detail here.

Each SERDES has four independent channels, which are named A, B, C, or D. Using this nomenclature, the SERDES A channels are named as AA, AB, AC, and AD, while SERDES B channels will be BA, BB, BC, and BD.

Table 19. Structural Register Elements

Address (0x)	Description
300xx	SERDES A, internal registers.
301xx	SERDES B, internal registers.
308xx	Channel A [A:D] registers (external to SERDES blocks).
309xx	Channel B [A:D] registers (external to SERDES blocks).
30A0x	Global registers (external to SERDES blocks).

A full memory map is included in Table 20.

Table 20 details the memory map for the FPSC portion of the ORT82G5 device.

Addresses for the control registers for the FPGA portion of the device are detailed in the ORCA Series 4 data sheet. This table shows the databus oriented for the PPC interface. DB0 is the MSB, while DB7 is the LSB. If the user master interface is used to perform operations to the ASIC core then the databus must be used in the opposite notation, where DB7 is the MSB and DB0 is the LSB.

Table 20. Memory Map

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
SERDES Alarm Registers (Read Only), xx=[AA, ...,BD]				
30000 - AA 30010 - AB 30020 - AC 30030 - AD 30100 - BA 30110 - BB 30120 - BC 30130 - BD	[0]	Reserved	00	Reserved
	[1]	LKI_xx		Receive PLL Lock Indication, Channel xx. LKI_xx = 1 indicates the receive PLL is locked.
	[2]	PRBSCHK_xx		PRBS Check Pass/Fail Indication, Channel xx. When PRBSCHK_xx = 0, it is a pass indication. PRBSCHK=1 indicates a failure. PRBSCHK_xx can be reset only if PRBS block is enables.
	[3]	PRBSTOUT_xx		PRBS Checker Watchdog Timer Time-Out Alarm, Channel xx. PRBSTOUT_xx = 1indicates a time out has occurred.
	[4:7]	Not Used		Not Used
SERDES Alarm Mask Registers (Read/Write), xx=[AA, ...,BD]				
30001 - AA 30011 - AB 30021 - AC 30031 - AD 30101 - BA 30111 - BB 30121 - BC 30131 - BD	[0]	Reserved	FF	Reserved, must be set to 1. Set to 1 on device reset.
	[1]	MLKI_xx		Mask Receive PLL Lock Indication, Channel xx.
	[2]	MPRBSCHK_xx.		Mask PRBS Check Pass/Fail Indication, Channel xx.
	[3]	MPRBSTOUT_xx		Mask PRBS Checker Watchdog Timer Time-Out Alarm, Channel xx.
	[4:7]	Not Used		Not Used
	[4]	HAMP_xx		Transmit Half Amplitude Selection Bit, Channel xx. When HAMP_xx = 1, the transmit output buffer voltage swing is limited to half its normal amplitude. Otherwise, the transmit output buffer maintains its full voltage swing. HAMP_xx = 0 on device reset.
	[5]	Reserved		Reserved. Must be set to 0 Set to 0 on device reset.
	[6]	Reserved		Reserved
	[7]	8b10bT_xx		Transmit 8b/10b Encoder Enable Bit, Channel xx. When 8b10bT_xx = 1, the 8b/10b encoder in the transmit path is enabled. Otherwise, the data is passed unencoded. 8b10bT_xx = 0 on device reset.
	[5:7]	Not Used		Not Used

Table 20. Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
SERDES Common Transmit and Receive Channel Configuration Registers (Read/Write), xx=[AA, ...,BD]				
30002 - AA 30012 - AB 30022 - AC 30032 - AD 30102 - BA 30112 - BB 30122 - BC 30132 - BD	[0]	TXHR_xx	00	Transmit Half Rate Selection Bit, Channel xx. When TXHR_xx = 1, HDOUT_xx's baud rate = (REFCLK[A:B]*10) and TCK78[A:B] =(REFCLK[A:B]/4); when TXHR_xx=0, HDOUT_xx's baud rate = (REFCLK[A:B]*20) and TCK78[A:B]=(REFCLK[A:B]/2). TXHR_xx = 0 on device reset.
	[1]	PWRDNT_xx		Transmit Powerdown Control Bit, Channel xx. When PWRDNT_xx = 1, sections of the transmit hardware are powered down to conserve power. PWRDNT_xx = 0 on device reset.

	[2]	PE0_xx		Transmit Preemphasis Selection Bit 0, Channel xx. PE0_xx and PE1_xx select one of three preemphasis settings for the transmit section.
				PEO_xx=PE1_xx = 0, Preemphasis is 0%
				PEO_xx=1, PE1_xx = 0 or PEO_xx=0, PE1_xx = 1, Preemphasis is 12.5%
				PEO_xx=PE1_xx = 1, Preemphasis is 25%.
				PEO_xx=PE1_xx = 0 on device reset.

	[3]	PE1_xx		
	[4]	HAMP_xx		Transmit Half Amplitude Selection Bit, Channel xx. When HAMP_xx = 1, the transmit output buffer voltage swing is limited to half its normal amplitude. Otherwise, the transmit output buffer maintains its full voltage swing. HAMP_xx = 0 on device reset.
	[5]	Reserved		Reserved. Must be set to 0 Set to 0 on device reset.
	[6]	Reserved		Reserved
	[7]	8b10bT_xx		Transmit 8b/10b Encoder Enable Bit, Channel xx. When 8b10bT_xx = 1, the 8b/10b encoder in the transmit path is enabled. Otherwise, the data is passed unencoded. 8b10bT_xx = 0 on device reset.
30003 - AA 30013 - AB 30023 - AC 30033 - AD 30103 - BA 30113 - BB 30123 - BC 30133 - BD	[0]	RXHR_xx	20	Receive Half Rate Selection Bit, Channel xx. When RXHR_xx =1, HDIN_xx's baud rate = (REFCLK[A:B]*10) and RCK78[A:B]=(REFCLK[A:B]/4); when RXHR_xx=0, HDIN_xx's baud rate = (REFCLK[A:B]*20) and RCK78[A:B]=(REFCLK/2). RXHR_xx = 0 on device reset.
	[1]	PWRDNR_xx		Receiver Power Down Control Bit, Channel xx. When PWRDNR_xx = 1, sections of the receive hardware are powered down to conserve power. PWRDNR_xx = 0 on device reset.
	[2]	Reserved		Reserved. Must be set to 1. Set to 1 on device reset.
	[3]	8b10bR_xx		Receive 8b/10b Decoder Enable Bit, Channel xx. When 8b10bR = 1, the 8b/10b decoder in the receive path is enabled. Otherwise, the data is passed undeocded. 8b10bR_xx = 0 on device reset.
	[4]	LINKSM_xx		Link State Machine Enable Bit, Channel xx. When LINKSM_xx = 1, the receiver Fiber Channel link state machine is enabled. Otherwise, the Fibre Channel link state machine is disabled. NOTE: LINKSM_xx is ignored when XAUI_MODE_xx=1. LINKSM_xx = 0 on device reset.
	[5:7]	Not Used		Not Used

Table 20. Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
SERDES Common Transmit and Receive Channel Configuration Registers (Read/Write), xx=[AA, ...,BD]				
30004 - AA 30014 - AB 30024 - AC 30034 - AD 30104 - BA 30114 - BB 30124 - BC 30134 - BD	[0]	PRBS_xx	40	Transmit and Receive PRBS Enable Bit, Channel xx. When PRBS_xx = 1, both the transmitter's PRBS generator and the receiver's PRBS checker are enabled. GRBS_[A:B] overrides these bits if set. NOTE: Due to timing, PRBS mode is intended for use with internally or externally looped-back data only. PRBS_xx = 0 on device reset.
	[1]	MASK_xx		Transmit and Receive Alarm Mask Bit, Channel xx. When MASK_xx = 1, the transmit and receive alarms of a channel are prevented from generating an interrupt (i.e., they are masked or disabled). The MASK_xx bit overrides the individual alarm mask bits in the Alarm Mask Registers. MASK_xx = 1 on device reset.
	[2]	SWRST_xx		Transmit and Receive Software Reset Bit, Channel xx. When SWRST_ss = 1, this bit provides the same function as the hardware reset, except that all configuration register settings are unaltered. This is not a self-clearing bit. Once set, this bit must be manually set and cleared. SWRST = 0 on device reset.
	[3:6]	Not Used		Not Used
	[7]	TESTEN_xx		Transmit and Receive Test Enable Bit, Channel xx. When TESTEN_xx = 1, the transmit and receive sections are placed in test mode. The TestMode_[A:B][4:0] bits in the Global Control Registers specify the particular test, and must also be set. NOTE: When the global test enable bit GTESTEN_[A:B] = 0, the individual channel test enable bits are used to selectively place a channel in test or normal mode. When GTESTEN_[A:B] = 1, all channels are set to test mode regardless of their TESTEN setting TESTEN_xx = 0 on device reset.

Table 20. Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
SERDES Global Control Registers (ReadWrite) Acts on all four Channels in SERDES Quad A or SERDES Quad B.				
30005 - A 30105 - B	[0]	GPRBS_[A:B]	44	Global PRBS Enable. The GPRBS_[A:B] bit globally enables the PRBS generators and checkers of all four channels when GPRBS_[A:B] = 1. GRBS_[A:B] overrides the PRBS_xx bits for the individual channels. GPRBS_[A:B] = 0 on device reset.
	[1]	GMASK_[A:B]		Global Mask. When GMASK_[A:B] = 1, the transmit and receive alarms of all channel in the SERDES quad are prevented from generating an interrupt (i.e., they are masked or disabled). The GMASK_[A:B] bit overrides the individual MASK_xx bits. GMASK_[A:B] = 1 on device reset.
	[2]	GSWRST_[A:B]		Software reset bit. The GSWRST_[A:B] bit provides the same function as the hardware reset for the transmit and receive sections of all four channels, except that the device configuration settings are not affected when GSWRST_[A:B] is asserted. This is not a self-clearing bit. Once set, this bit must be manually set and cleared. The GSWRST_[A:B] bit overrides the individual SWRST_xx bits. GSWRST_[A:B] = 0 on device reset.
	[3]	GPWRDNT_[A:B]		Powerdown Transmit Function. When GPWRDNT_[A:B] = 1, sections of the transmit hardware for all four channels of are powered down to conserve power. The GPWRDNT_[A:B] bit overrides the individual PWRDNT_xx bits. GPWRDNT_[A:B] = 0 on device reset.
	[4]	GPWRDNR_[A:B]		Powerdown Receive Function. When GPWRDNR_[A:B] = 1, sections of the receive hardware for all four channels are powered down to conserve power. The GPWRDNR_[A:B] bit overrides the individual PWRDNR_xx bits. GPWRDNR_[A:B] = 0 on device reset.
	[5]	Reserved		Reserved, 1 on device reset.
	[6]	Not Used		Not Used
	[7]	GTESTEN_[A:B]		Test Enable Control. When GTESTEN_[A:B] = 1, the transmit and receive sections of all four channels are placed in test mode. The GTESTEN_[A:B] bit overrides the individual TESTEN_xx bits. GTESTEN_[A:B] = 0 on device reset.
30006 - A 30106 - B	[0:4]	TestMode[A:B]	00	TestMode - See Test Mode section for settings
	[5]	Not Used		Not Used
	{6:7]	Reserved		Reserved

Table 20. Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
Control Registers (Read/Write), xx=[AA, ...,BD]				

30800 - Ax
30900 - Bx
	[0]xA
	[1]xB
	[2]xC
	[3]xD
		ENBYSYNC_xx	00	ENBYSYNC_xx = 1 Enables Receiver Byte Synchronization for Channel xx.
				ENBYSYNC_xx = 0 on device reset.

	[4]xA [5]xB [6]xC [7]xD	LCKREFN_xx		LCKREFN_xx = 0 Locks the receiver PLL to ref reference clock for Channel xx. LCKREFN_xx =1 = Locks the receiver to data for Channel xx. NOTE: When LCKREFN_xx = 0, the corresponding LKI_xx bit is also zero. LCKREFN_xx = 0 on device reset.
30801 - Ax 30901 - Bx	[0]xA [1]xB [2]xC [3]xD	LOOPENB_xx		Enable Loopback Mode for Channel xx. When LOOPEN_xx=1, the transmitter high-speed output is looped back to the receiver high-speed input. This mode is similar to high-speed loopback mode enabled by TESTMODE_xx except that LOOPEN_xx disables the high-speed serial output. LOOPEN_xx=0 on device reset.
	[4]xA [5]xB [6]xC [7]xD	NOWDALIGN_xx		Word Align Disable Bit. When NOWDALIGN_xx=1, receiver word alignment is disabled for Channel xx. NOWDALIGN_xx=0 on device reset.
30802 30902	[0:7]	Reserved for future use		
30803 30903	[0:7]	Reserved for future use		
30810 - Ax 30910 - Bx	[0]xA [1]xB [2]xC [3]xD	DOWDALIGN_xx	00	Word Realign Bit. When DOWDALIGN_xx transitions from 0 to 1, the receiver realigns on the next comma character for Channel xx. NOWDALIGN_xx=0 on device reset.
	[4]xA [5]xB [6]xC [7]xD	FMPU_STR_EN_xx		Enable multi-channel alignment for Channel xx. When FMPU_STR_EN_xx=1, the corresponding channel participates in multi-channel alignment. FMPU_STR_EN_xx=0 on device reset.
30811 - Ax 30911 - Bx	[0:1] xA [2:3] xB [4:5] xC [6:7] xD	FMPU_SYNMODE_xx[0:1]	00	Sync mode for xx 00 = No channel alignment 10 = Twin channel alignment 01 = Quad channel alignment 11 = 8 channel alignment
30812 30912	[0:7]	Reserved for future use		
30813 30912	[0:7]	Reserved for future use		
30823 30923	[0:7]	Reserved for future use		
30830 30930	[0:7]	Reserved for future use		

Table 20. Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
30820 - Ax 30920 - Bx	[0]xA [1]xB [2]xC [3]xD	FMPU_RESYNC1_xx	00	Resync a Single Channel. When FMPU_RESYNC1_xx transitions from 0 to 1, the corresponding channel is resynchronized (the write and read pointers are reset). FMPU_STR_EN_xx=0 on device reset.
	[4] xA & xB [5] xC & xD	FMPU_RESYNC2_A1	00	Resync a Pair of Channels. When FMPU_RESYNC2_[A:B][1:2] transitions from a 0 to a 1, the corresponding channel pair is resynchronized. FFMPU_RESYNC2_[A:B][1:2=0 on device reset.
	[6]	FMPU_RESYNC4[A:B]	00	Resync a Four-Channel Group. When FMPU_RESYNC4[A:B] transitions from a 0 to a 1, the corresponding four-channel group is resynchronized. FMPU_RESYNC4[A:B]=0 on device reset.
	[7]	XAUI_MODE[A:B]	00	Controls use of XAUI link state machine in place of Fibre-Channel state machine. When XAUI_MODE[A:B]=1, all four channels in the SERDES quad enable their XAUI link state machines. (LINKSM_xx bits are ignored). XAUI_MODE[A:B]=0 on device reset.
30821 - A 30921 - B	[0]	NOCHALGN [A:B]		Bypass channel alignment. NOCHALGN [A:B] =1 causes bypassing of multi-channel alignment FIFOs for the corresponding SERDES quad. NOCHALGN [A:B] =0 on device reset.
	[1:7]	Reserved for future use		
30822	[0:7]	Reserved for future use		
30832	[0:7]	Reserved for future use		
30833	[0:7]	Reserved for future use		
30933	[0:3]	Reserved for future use		
	[4:5]	SCHAR_CHAN[0:1]	00	Select channel to test 00 = Channel BA 10 = Channel BB 01 =Channel BC 11 = Channel BD
	[6]	SCHAR_TXSEL		1=Select TX option 0=Select RX option
	[7]	SCHAR_ENA		1=Enable Characterization of SERDES B
Status Registers (Read Only), xx=[AA, ...,BD]				
30804 - Ax 30904 - Bx	[0:1] xA [2:3] xB [4:5] xC [6:7] xD	XAUISTAT_xx[0:1]	00	XAUI Status Register. Status of XAUI link state machine for Channel xx 00—No synchronization. 10—Synchronization done. 01,11—Not used. XAUISTAT_xx[0:1] = 00 on device reset.
30805 - Ax 30905 - Bx	[0]xA [1]xB [2]xC [3]xD	DEMUXWAS_xx	00	Status of Word Alignment. When DEMUX_WAS_xx=1, word alignment is achieved for Channel xx. DEMUX_WAS_xx=0 on device reset.
	[4]xA [5]xB [6]xC [7]xD	CH248_SYNC_xx		Status of Channel Alignment. When CH248_SYNC_xx=1, multi-channel alignment is achieved for Channel xx. CH248_SYNC_xx=0 on device reset.
30806 30906	[0:7]	Reserved for future use		

Table 20. Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
30807 - 30907	[0:7]	Reserved for future use		
30814 - Ax 30914 - Bx	[0] xA & xB [1] xC & xD	SYNC2_[A:B][1:2] OVFL		Multi-Channel Overflow Status. When SYNC2_[A:B][1:2]OVFL=1, dual-channel synchronization FIFO overflow has occurred. SYNC2_[A:B][1:2]OVFL=0 on device reset.
	[2]	SYNC4_ [A:B]OVFL		Multi-Channel Overflow Status. When SYNC4_[A:B]OVFL=1, quad-channel synchronization FIFO overflow has occurred. SYNC4_[A:B]OVFL=0 on device reset.
	[3] xA & xB [4] xC & xD	SYNC2_[A:B][1:2] OOS		Multi-Channel Out-Of-Sync Status. When SYNC2_[A:B][1:2] OOS=1, dual-channel synchronization has failed. SYNC2_[A:B][1:2] OOS on device reset.
	[5]	SYNC4_[A:B]_OOS		Multi-Channel Out-Of-Sync Status. When SYNC4_[A:B]_OOS=1, quad-channel synchronization has failed. SYNC4_[A:B]_OOS=0 on device reset.
	[6:7]	Reserved for future use		
30815 30915	[0:7]	Reserved for future use		
30816 30916	[0:7]	Reserved for future use		
30817 30917	[0:7]	Reserved for future use		
30824 30924	[0:7]	Reserved for future use		
Status Registers (Read Only), xx=[AA, ...,BD]				
30804 - Ax 30904 - Bx	[0:1] xA [2:3] xB [4:5] xC [6:7] xD	XAUISTAT_xx[0:1]	00	XAUI Status Register. Status of XAUI link state machine for Channel xx 00—No synchronization. 10—Synchronization done. 01,11—Not used. XAUISTAT_xx[0:1] = 00 on device reset.
30805 - Ax 30905 - Bx	[0]xA [1]xB [2]xC [3]xD	DEMUXWAS_xx	00	Status of Word Alignment. When DEMUX_WAS_xx=1, word alignment is achieved for Channel xx. DEMUX_WAS_xx=0 on device reset.
	[4]xA [5]xB [6]xC [7]xD	CH248_SYNC_xx		Status of Channel Alignment. When CH248_SYNC_xx=1, multi-channel alignment is achieved for Channel xx. CH248_SYNC_xx=0 on device reset.
30806 30906	[0:7]	Reserved for future use		
30807 30907	[0:7]	Reserved for future use		

Table 20. Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
30814 - Ax 30914 - Bx	[0] xA & AB [1] xC & xD	SYNC2_[A:B][1:2] OVFL	00	Multi-Channel Overflow Status. When SYNC2_[A:B][1:2]OVFL=1, dual-channel synchronization FIFO overflow has occurred. SYNC2_[A:B][1:2]OVFL=0 on device reset.
	[2]	SYNC4_ [A:B]OVFL		Multi-Channel Overflow Status. When SYNC4_[A:B]OVFL=1, quad-channel synchronization FIFO overflow has occurred. SYNC4_[A:B]OVFL=0 on device reset.
	[3] xA & AB [4] xC & xD	SYNC2_[A:B][1:2] OOS		Multi-Channel Out-Of-Sync Status. When SYNC2_[A:B][1:2] OOS=1, dual-channel synchronization has failed. SYNC2_[A:B][1:2] OOS on device reset.
	[5]	SYNC4_[A:B]_OOS		Multi-Channel Out-Of-Sync Status. When SYNC4_[A:B]_OOS=1, quad-channel synchronization has failed. SYNC4_[A:B]_OOS=0 on device reset.
	[6:7]	Reserved for future use		
30815 30915	[0:7]	Reserved for future use		
30816 30916	[0:7]	Reserved for future use		
30817 30917	[0:7]	Reserved for future use		
30824 30924	[0:7]	Reserved for future use		
30825 30925	[0:7]	Reserved for future use		
30826 30926	[0:7]	Reserved for future use		
30827 30927	[0:7]	Reserved for future use		
30834 30934	[0:7]	Reserved for future use		
30835 30935	[0:7]	Reserved for future use		
30836 30936	[0:7]	Reserved for future use		
30837 30937	[0:7]	Reserved for future use		

Table 20. Memory Map (Continued)

(0x) Absolute Address	Bit	Name	Reset Value (0x)	Description
Common Control Registers (Read/Write)				
30A000 30A000	[0:1]	TCKSELA	00	Transmit Clock Select. Controls source of 78 MHz TCK78 for SERDES quad A 00 = Channel AA 10 = Channel AB 01 = Channel AC 11 = Channel AD
	[2:3	RCKSELA		Receive Clock Select. Controls source of 78 MHz RCK78 for SEDRES quad A 00 = Channel AA 10 = Channel AB 01 = Channel AC 11 = Channel AD
	[4:5]	TCKSELB		Transmit Clock Select. Controls source of 78 MHz TCK78 for SERDES quad B 00 = Channel BA 10 = Channel BB 01 = Channel BC 11 = Channel BD
	[6:7}	RCKSELB		Receive Clock Select. Controls source of 78 MHz RCK78 for SERDES quad B 00 = Channel BA 10 = Channel BB 01 = Channel BC 11 = Channel BD
30A01 30A01	[0:4]	Reserved for future use		
	[5:7]	RX_FIFO_MIN	00	Threshold for low address in RX_FIFO's RX_FIFO_MIN Bits x, y, z.*
30A02 30A02	[0:1]	RX_FIFO_MIN	00	Threshold for low address in RX_FIFO's RX_FIFO_MIN Bits u, w.*
	[2]	FMPU_RESYNC8		Resynchronize and eight-channel group. When FPMPU_RESYNC8 transitions from A0 to A1, the entire eight-channel group is resynchronized. FMPU_RESYNC0 = 0 on device reset
	[3:7]	Reserved for future use		
Common Status Registers xx=[AA, ...,BD]				
30A04 30A04	[0]	SYNC8_OVFL	00	Read-Only Multi-Channel Overflow Status. When SYNC8_OVFL=1, 8-channel synchronization FIFO overflow has occurred. SYNC8_OVFL=0 on device reset.
	[1]	SYNC8_OOS		Read-Only Multi-Channel Out-Of-Sync Status. When SYNC8_OOS=1, 8-channel synchronization has failed. SYNC8_OOS=0 on device reset.
	[2:7]	Reserved for future use		
30A05 30A05		Reserved for future use		

* RX_FIFO_MIN {0:4} = Bits [w, v, z, y, x], Useful Values: 0:17(decimal)

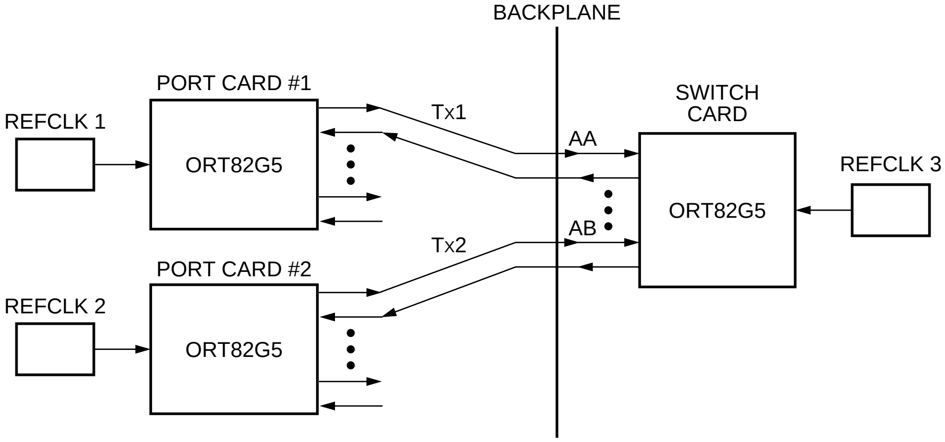
Recommended Board-level Clocking for the ORT82G5

Option 1: Asynchronous Reference Clocks Between Rx and Tx Devices

Each board that uses the ORT82G5 as a transmit or receive device will have its own local reference clock as shown in Figure 29. Figure 29 shows the ORT82G5 device on the switch card receiving data on two of its channels from a separate source. Data tx1 is transmitted from a tx device with refclk1 as the reference clock and Data tx2 is transmitted from a tx device with refclk2 as the reference clock. Receive channel AA locks to the incoming data tx1 and receive channel AB locks to the incoming data tx2.

The advantage of this clocking scheme is the fact that it is not necessary to distribute a reference clock (typically 156 MHz for 10GE and 155.52 MHz for OC-192 applications) across a backplane.

Figure 29. Asynchronous Clocking Between Rx and Tx Devices



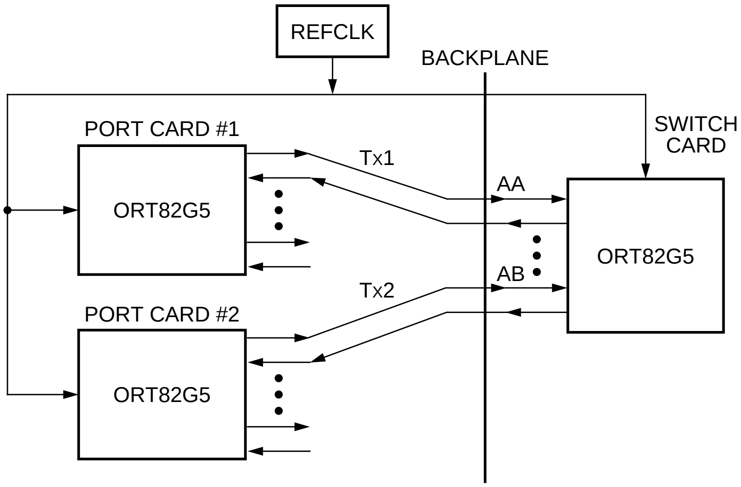
Option 2: Synchronous Reference Clocks to Rx and Tx Devices

In this type of clocking, a single reference clock is distributed to all receive and transmit devices in a system (Figure 30). This distributed clocking scheme will permit maximum flexibility in the usage of transmit and receive channels in the current silicon such as:

- All transmit and receive channels can be used within any quad in receive channel alignment or alignment bypass mode.
- In channel alignment mode, each receive channel operates on its own independent clock domain.

The disadvantage with this scheme is the fact that it is difficult to distribute a 156 MHz reference clock across a backplane. This may require expensive clock driver chips on the board to drive clocks to different destinations within the specified jitter limits for the reference clock.

Figure 30. Distributed Reference Clock to Rx And Tx Devices



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.

Parameter	Symbol	Min.	Max.	Unit
Storage Temperature	T_{STG}	– 65	150	°C
Power Supply Voltage with Respect to Ground	V_{DD33}	– 0.3	4.2	V
	V_{DDIO}	– 0.3	4.2	V
	V_{DD15}	—	2.0	V
Input Signal with Respect to Ground	V_{IN}	V_{SS} – 0.3	V_{DD}IO + 0.3	V
Signal Applied to High-impedance Output	—	V_{SS} – 0.3	V_{DD}IO + 0.3	V
Maximum Package Body (Soldering) Temperature	—	—	220	°C

Recommended Operating Conditions

Parameter	Symbol	Min.	Max.	Unit
Power Supply Voltage with Respect to Ground[1]	V_{DD33}	3.0	3.6	V
	V_{DD15}	1.425	1.575	V
Input Voltages	V_{IN}	V_{SS} – 0.3	V_{DDIO} + 0.3	V
Junction Temperature[2]	T_J	– 40	125	°C

1. 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 (OR4E04) and the ORCA Series 4 I/O Buffer Technical Note. FPSC Standby Currents (IDDSB15 and IDDSB33) are tested with the Embedded Core in the powered down state.
2. Designed for greater than 10 year electromigration life at 3.125 Gbits/s at 100 ˚C junction temperature.

SERDES Electrical and Timing Characteristics

Table 21. Absolute Maximum Ratings

Parameter	Conditions	Min.	Typ.	Max.	Unit
Power Dissipation	SERDES, MUX/DEMUX, Align FIFO and I/O (per channel), 1.25 Gbit/s	—	—	195	mW
	SERDES, MUX/DEMUX, Align FIFO and I/O (per channel), 2.50 Gbit/s	—	—	210	mW
	SERDES, MUX/DEMUX, Align FIFO and I/O (per channel), 3.25 Gbit/s	—	—	225	mW
	8b/10b Encoder/Decoder (per channel)	—	—	50	mW

Table 22. Recommended Operating Conditions

Parameter	Conditions	Min.	Typ.	Max.	Unit
VDD15 Supply Voltage (VDD15, VDDRX, VDDTX, VDDAUX, VDDGB)	—	1.425	—	1.575	V
CML I/O Supply Voltage (VDDIB, VDDOB)	—	1.425	—	1.890	V

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

Figure 31. Receive Data Eye-Diagram Template (Differential)

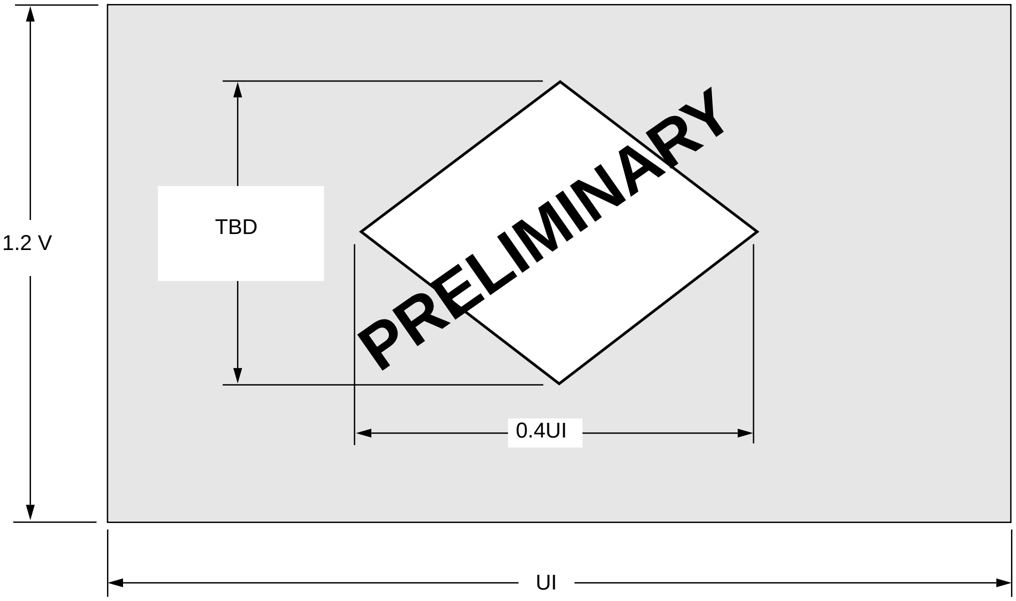


Figure 31 provides a graphical characterization of the SERDES receiver input requirements. It provides guidance on a number of input parameters, including signal amplitude and rise time limits, noise and jitter limits, and P and N input skew tolerance. it is believed that incoming data patterns falling within the shaded region of the template will be received without error (BER < 10E-12), over all specified operating conditions.

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

Signal jitter is of special interest to system designers. It is often the primary limiting characteristic of long digital links and of systems with high noise level environments. An interesting characteristic of the Clock and Data Recovery (CDR) portion of the ORT82G5 SERDES receiver is its ability to filter incoming signal jitter that is below the clock recovery PLL bandwidth (estimated to be about 3 MHz). For signals with high levels of low frequency jitter the receiver can detect incoming data, error-free, with eye-openings significantly less than that of Figure 31. . This phenomenon has been observed in the laboratory.

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

Table 23. Receiver Specifications

Parameter	Conditions	Min	Typ	Max	Unit
Input Data					
Stream of Nontransitions	—	—	—	72	Bits
Eye Opening Interval	—	0.4	—	—	UIP-P
Eye Opening Voltage	—	200	—	—	mVP-P

HSI Electrical and Timing Characteristics

Table 24. Reference Clock Specifications (REFINP and REFINN)

Parameter	Min	Typ	Max	Unit
Frequency Range	100	—	185	MHz
Frequency Tolerance	– 100	—	100	ppm
Duty Cycle (Measured at 50% Amplitude Point)	40	50	60	%
Rise Time	—	500	1000	ps
Fall Time	—	500	1000	ps
P–N Input Skew	—	—	75	ps
Differential Amplitude	500	800	2 x V_{DD}	mV_{p-p}
Common Mode Level	$V_{single-ended}/2$	0.75	V_{DD15} – ($V_{single-ended}/2$)	V
Single-Ended Amplitude	250	400	V_{DD15}	mV_{p-p}
Input Capacitance (at REFINP)	—	—	5	pF
Input Capacitance (at REFINPIT)	—	—	3	pF
Inband (< 10 MHz) Jitter (2.5 Gbits/s)	—	—	30	ps_{p-p}
Inband (< 10 MHz) Jitter (1.25 Gbits/s)	—	—	60	ps_{p-p}

Note: Additional (<10 MHz) REFCLK jitter will increase the total transmit output jitter.
Note: The frequency tolerance is important when separate reference clocks are used for the transmitter that is driving the receive

Table 25. Channel Output Jitter (1.25 Gbits/s) - Half Rate Mode

Parameter	Min	Typ	Max	Unit
Deterministic	—	—	0.08	UI_{p-p}
Random	—	—	0.12	UI_{p-p}
Total	—	—	0.20	UI_{p-p}

Table 26. Channel Output Jitter (2.5 Gbits/s) Full Rate Mode

Parameter	Min	Typ	Max	Unit
Deterministic	—	—	0.10	UI_{p-p}
Random	—	—	0.14	UI_{p-p}
Total	—	—	0.24	UI_{p-p}

Table 27. Serial Output Timing and Levels (CML I/O)

Parameter	Min	Typ	Max	Unit
Rise Time (20%—80%)	50	80	110	ps
Fall Time (80%—20%)	50	80	110	ps
Common Mode	V_{DDOB} –0.30	V_{DDOB} –0.25	V_{DDOB} –0.15	V
Differential Swing (Full Amplitude)	800	900	1100	mV_{p-p}
Differential Swing (Half Amplitude)	400	500	600	mV_{p-p}
Output Load	—	50	—	Ω

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

HSI Electrical and Timing Characteristics

Table 28. Serial Input Timing and Levels (CML I/O)

Parameter	Min	Typ	Max	Unit
Rise Time	See Eye Diagram, Figure 31.	—	—	ps
Fall Time	See Eye Diagram, Figure 31.	—	—	ps
Differential Swing	200	—	—	mVp-p
Common-mode Level	0.5	—	VDD15	V
Internal Buffer Resistance (Each input to VDDIB)	40	50	60	Ω
PLL Lock Time			50	µs

Embedded Core Timing Characteristics

Table 29 summarizes the end-to-end latencies through the embedded core for the various modes. All latencies are given in clock cycles for system clocks at half the REFCLK_[A:B] frequency. For a REFCLK_[A:B] of 156.25 MHz, a system clock cycle is 6.4 ns.

Table 29. Signal Latencies, Embedded Core

Operating Mode	Signal Latency (max.)
Transmit Path	5 clock cycles
Receive Path	
Multi-Channel Alignment Bypassed[1]	4.5 clock cycles
With Multi-Channel Alignment[1]	13.5-22.5 clock cycles

[1] With multi-channel alignment, the latency is largest when the skew between channels is at the maximum that can be correctly compensated for (18 clock cycles). The latency specified in the table is for data from the channel received first.

Pin Descriptions

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

Symbol	I/O	Description
Dedicated Pins		
VDD33	—	3.3V 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.
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.

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

Symbol	I/O	Description
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.*

* 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 30. FPSC Function Pin Descriptions

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

Table 30. FPSC Function Pin Descriptions (Continued)

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

Table 30. FPSC Function Pin Descriptions (Continued)

Symbol	I/O	Description
VSSIB_AA	—	High-speed input receive buffer ground (no external pin).
VSSIB_AB	—	High-speed input receive buffer ground.
VSSIB_AC	—	High-speed input receive buffer ground.
VSSIB_AD	—	High-speed input receive buffer ground.
VSSIB_BA	—	High-speed input receive buffer ground.
VSSIB_BB	—	High-speed input receive buffer ground.
VSSIB_BC	—	High-speed input receive buffer ground.
VSSIB_BD	—	High-speed input receive buffer ground.
VSSOB_AA	—	High-speed output transmit buffer ground (no external pin).
VSSOB_AB	—	High-speed output transmit buffer ground.
VSSOB_AC	—	High-speed output transmit buffer ground.
VSSOB_AD	—	High-speed output transmit buffer ground.
VSSOB_BA	—	High-speed output transmit buffer ground.
VSSOB_BB	—	High-speed output transmit buffer ground.
VSSOB_BC	—	High-speed output transmit buffer ground.
VSSOB_BD	—	High-speed output transmit buffer ground.
VSSTX_AA	—	SERDES analog transmit circuitry ground (no external pin).
VSSTX_AB	—	SERDES analog transmit circuitry ground.
VSSTX_AC	—	SERDES analog transmit circuitry ground.
VSSTX_AD	—	SERDES analog transmit circuitry ground.
VSSTX_BA	—	SERDES analog transmit circuitry ground.
VSSTX_BB	—	SERDES analog transmit circuitry ground.
VSSTX_BC	—	SERDES analog transmit circuitry ground.
VSSTX_BD	—	SERDES analog transmit circuitry ground.
VDDRX_AA	—	1.5 V Power supply for SERDES analog transmit and receive circuitry.
VDDRX_AB	—	1.5 V Power supply for SERDES analog transmit and receive circuitry.
VDDRX_AC	—	1.5 V Power supply for SERDES analog transmit and receive circuitry.
VDDRX_AD	—	1.5 V Power supply for SERDES analog transmit and receive circuitry.
VDDRX_BA	—	1.5 V Power supply for SERDES analog transmit and receive circuitry.
VDDRX_BB	—	1.5 V Power supply for SERDES analog transmit and receive circuitry.
VDDRX_BC	—	1.5 V Power supply for SERDES analog transmit and receive circuitry.
VDDRX_BD	—	1.5 V Power supply for SERDES analog transmit and receive circuitry.
VDDAUX_A	—	1.5 V power supply for SERDES auxiliary circuit.
VDDAUX_B	—	1.5 V power supply for SERDES auxiliary circuit.

Power Supplies for ORT82G5

Power Supply Descriptions

Table 31 shows the ORT82G5 embedded core power supply connection groupings. The Tx-Rx digital power supplies are used for transmit and receive digital logic including the microprocessor logic. The Tx-Rx analog power supplies are used for high-speed analog circuitry between the I/O buffers and the digital logic. The Rx input buffer power supplies are used to power the input (receive) buffers. The Tx output buffer supplies are used to power the output (transmit) buffers. The Rx and Tx buffer power supplies can be independently set to 1.5 V or 1.8 V, depending on the end application. The auxiliary and guard band supplies are independent connection brought out to pins. In the ORT82G5, many of the VDD pins shown in Table 31 are connected together at the package substrate level.

The same also applies for various VSS pins. At the package ball level in Table , the following names appear instead of the names in Table 31: VDDT, VDDR, VDDOB, VDDIB, VSST, VSSRX.

Table 31. Power Supply Pin Groupings

Tx-Rx Digital 1.5 V	Tx-Rx Analog 1.5 V (VDDT, VDDR)	Tx Output Buffers 1.5/1.8 V (VDDOB)	Rx Input Buffers 1.5 V/1.8 V (VDDIB)	Auxiliary 1.5 V (VDDAUX)	Guard Band 1.5 V (VDDGB)
VDD15	VDDRX_AA	VDDOB_AA	VDDIB_AA	VDDAUX_A	VDDGB_A
—	VDDTX_AA	VDDOB_AB	VDDIB_AB	VDDAUX_B	VDDGB_B
—	VDDRX_AA	VDDOB_AC	VDDIB_AC	—	—
—		VDDOB_AD	VDDIB_AD	—	—
—		VDDOB_BA	VDDIB_BA	—	—
—		VDDOB_BB	VDDIB_BB	—	—
—		VDDOB_BC	VDDIB_BC	—	—
—		VDDOB_BD	VDDIB_BD	—	—
—	VDDRX_BA	—	—	—	—
—	VDDTX_BA	—	—	—	—
—	VDDRX_BA	—	—	—	—
—		—	—	—	—
—		—	—	—	—
—		—	—	—	—
—		—	—	—	—
—		—	—	—	—

Recommended Power Supply Connections

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

- Tx-Rx digital auxiliary supplies.
- The Tx-Rx digital and auxiliary power supply nodes should be supplied by a 1.5 V source. A single 1.5 V source can supply power to Tx-Rx digital and auxiliary nodes.
- Tx-Rx analog, guardband supplies.
- A dedicated 1.5 V power supply should be provided to the analog power pins. This will allow the end user to minimize noise. The guard band pins can also be sourced from the analog power supplies.
- Tx output buffers.
- The power supplies to the Tx output buffers should be isolated from the rest of the board power supplies. Special care must be taken to minimize noise when providing board level power to these output buffers. The power supply can be 1.5 V or 1.8 V depending on the end application.
- Rx input buffers.

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

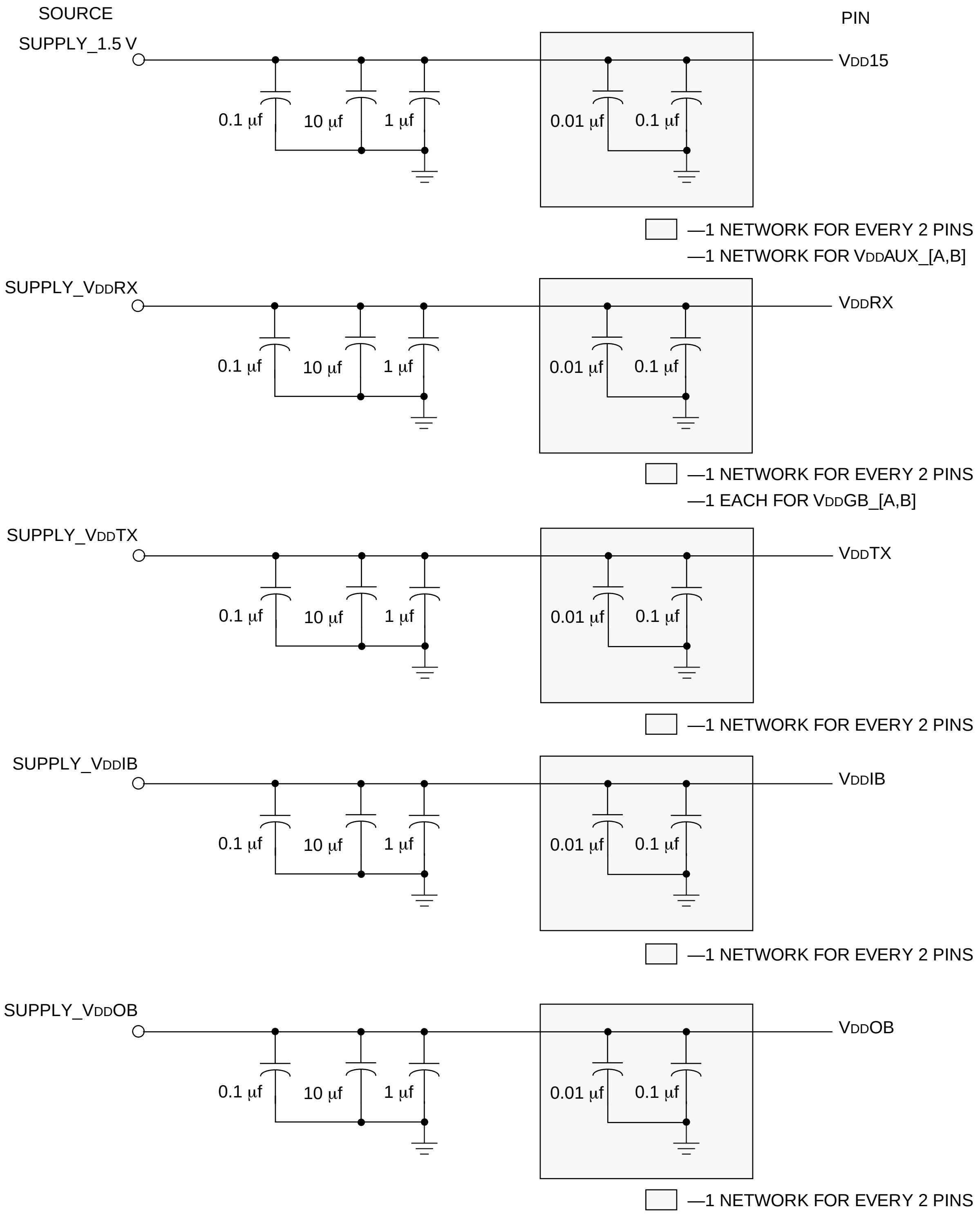
Recommended Power Supply Filtering Scheme

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

Power supply filtering is in the form of:

- A parallel bypass capacitor network consisting of 10 µf, 0.1 µf, and 1.0 µf caps close to the power source.
- A parallel bypass capacitor network consisting of 0.01 µf and 0.1 µf close to the pin on the ORT82G5.
- Example connections are shown in Figure 31. The naming convention for the power supply sources shown in the figure are as follows:
  - Supply_1.5 V—Tx-Rx digital, auxiliary power pins.
  - Supply_VDDRX—Rx analog power pins, guard band power pins.
  - Supply VDDTX—Tx analog power pins.
  - Supply VDDIB—Input Rx buffer power pins.
  - Supply_VDDOB—Output Tx buffer power pins.

Figure 32. Power Supply Filtering



Package Information

Package Pinouts

Table 32 provides the package pin and pin function for the ORT82G5 FPSC and packages. The bond pad name is identified in the PIO nomenclature used in the ispLEVER System software design editor. The Bank column provides information as to which output voltage level bank the given pin is in. The Group column provides information as to the group of pins the given pin is in. This is used to show which VREF pin is used to provide the reference voltage for single-ended limited-swing I/Os. If none of these buffer types (such as SSTL, GTL, HSTL) are used in a given group, then the VREF pin is available as an I/O pin.

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

As shown in the pair columns in Table 32, differential pairs and physical locations are numbered within each bank (e.g., L19C-A0 is the nineteenth pair in an associated bank). A ‘C’ indicates complementary differential, whereas a ‘T’ indicates true differential. An _A0 indicates the physical location of adjacent balls in either the horizontal or vertical direction. Other physical indicators are as follows:

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

VREF pins, shown in the Pin Description column in Table 32, are associated to the bank and group (e.g., VREF_TL_01 is the VREF for group one of the Top Left (TL) bank.t

Table 32. ORT82G5 680-Pin PBGAM Pinout

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

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

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

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

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

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

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

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

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

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

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

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

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

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

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

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AP33	—	—	IO	PSCHAR_LDIO8	—	—
AN32	—	—	IO	PSCHAR_LDIO7	—	—
AM31	—	—	IO	PSCHAR_LDIO6	—	—
AA17	—	—	VDD15	VDD15	—	—
AM32	—	—	VDD33	VDD33	—	—
AL31	—	—	IO	PSCHAR_LDIO5	—	—
AM33	—	—	IO	PSCHAR_LDIO4	—	—
AA18	—	—	VDD15	VDD15	—	—
AK30	—	—	IO	PSCHAR_LDIO3	—	—
AL32	—	—	IO	PSCHAR_LDIO2	—	—
AA19	—	—	VDD15	VDD15	—	—
AB16	—	—	VDD15	VDD15	—	—
AK31	—	—	VDD33	VDD33	—	—
AJ30	—	—	IO	PSCHAR_LDIO1	—	—
AK33	—	—	IO	PSCHAR_LDIO0	—	—
AK34	—	—	IO	PSCHAR_CKIO1	—	—
AJ31	—	—	IO	PSCHAR_CKIO0	—	—
AJ33	—	—	IO	PSCHAR_XCK	—	—
AJ34	—	—	IO	PSCHAR_WDSYNC	—	—
AH30	—	—	IO	PSCHAR_CV	—	—
AH31	—	—	IO	PSCHAR_BYTSYNC	—	—
AH32	—	—	O	ATMOUT_B (no connect)	—	—
AH33	—	—	VSSGB_B	VSSGB_B	—	—
AH34	—	—	VDDGB_B	VDDGB_B	—	—
AA32	—	—	VDDR	VDDAUX_B	—	—
AF30	—	—	—	REXT_B	—	—
AF31	—	—	—	REXTN_B	—	—
AE30	—	—	I	REFCLKN_B	—	—
AE31	—	—	I	REFCLKP_B	—	—
AB32	—	—	VSST	VSSAUX_B	—	—
AD30	—	—	VDDIB	VDDIB_BA	—	—
AD32	—	—	VDDR	VDDRX_BA	—	—
AF33	—	—	I	HDINN_BA	—	—
AC32	—	—	VSST	VSSIB_BA	—	—
AF34	—	—	I	HDINP_BA	—	—
AE32	—	—	VDDR	VDDRX_BA	—	—
AD31	—	—	VSSRX	VSSRX_BA	—	—
K32	—	—	VDDT	VDDTX_BA	—	—
AC30	—	—	VDDOB	VDDOB_BA	—	—
AE33	—	—	O	HDOUTN_BA	—	—
AF32	—	—	VSST	VSSOB_BA	—	—
AE34	—	—	O	HDOUTP_BA	—	—
AC30	—	—	VDDOB	VDDOB_BA	—	—
AG30	—	—	VSST	VSSTX_BA	—	—

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
AB30	—	—	VDDIB	VDDIB_BB	—	—
AD33	—	—	I	HDINN_BB	—	—
AG31	—	—	VSST	VSSIB_BB	—	—
AD34	—	—	I	HDINP_BB	—	—
AC31	—	—	VSSRX	VSSRX_BB	—	—
AB31	—	—	VDDOB	VDDOB_BB	—	—
AC33	—	—	O	HDOUTN_BB	—	—
AG32	—	—	VSST	VSSOB_BB	—	—
AC34	—	—	O	HDOUTP_BB	—	—
AB31	—	—	VDDOB	VDDOB_BB	—	—
AG33	—	—	VSST	VSSTX_BB	—	—
AA30	—	—	VDDIB	VDDIB_BC	—	—
AB33	—	—	I	HDINN_BC	—	—
AG34	—	—	VSST	VSSIB_BC	—	—
AB34	—	—	I	HDINP_BC	—	—
AA31	—	—	VSSRX	VSSRX_BC	—	—
Y30	—	—	VDDOB	VDDOB_BC	—	—
AA33	—	—	O	HDOUTN_BC	—	—
H30	—	—	VSST	VSSOB_BC	—	—
AA34	—	—	O	HDOUTP_BC	—	—
Y31	—	—	VDDOB	VDDOB_BC	—	—
H31	—	—	VSST	VSSTX_BC	—	—
W30	—	—	VDDIB	VDDIB_BD	—	—
Y33	—	—	I	HDINN_BD	—	—
H32	—	—	VSST	VSSIB_BD	—	—
Y34	—	—	I	HDINP_BD	—	—
W31	—	—	VSSRX	VSSRX_BD	—	—
V30	—	—	VDDOB	VDDOB_BD	—	—
W33	—	—	O	HDOUTN_BD	—	—
H33	—	—	VSST	VSSOB_BD	—	—
W34	—	—	O	HDOUTP_BD	—	—
V31	—	—	VDDOB	VDDOB_BD	—	—
H34	—	—	VSST	VSSTX_BD	—	—
J32	—	—	VSST	VSSTX_AD	—	—
U31	—	—	VDDOB	VDDOB_AD	—	—
T34	—	—	O	HDOUTP_AD	—	—
M32	—	—	VSST	VSSOB_AD	—	—
T33	—	—	O	HDOUTN_AD	—	—
U30	—	—	VDDOB	VDDOB_AD	—	—
T31	—	—	VSSRX	VSSRX_AD	—	—
R34	—	—	I	HDINP_AD	—	—
N32	—	—	VSST	VSSIB_AD	—	—
R33	—	—	I	HDINN_AD	—	—
T30	—	—	VDDIB	VDDIB_AD	—	—

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
U32	—	—	VSST	VSSTX_AC	—	—
R31	—	—	VDDOB	VDDOB_AC	—	—
P34	—	—	O	HDOUTP_AC	—	—
U33	—	—	VSST	VSSOB_AC	—	—
P33	—	—	O	HDOUTN_AC	—	—
R30	—	—	VDDOB	VDDOB_AC	—	—
P31	—	—	VSSRX	VSSRX_AC	—	—
N34	—	—	I	HDINP_AC	—	—
U34	—	—	VSST	VSSIB_AC	—	—
N33	—	—	I	HDINN_AC	—	—
P30	—	—	VDDIB	VDDIB_AC	—	—
V32	—	—	VSST	VSSTX_AB	—	—
N31	—	—	VDDOB	VDDOB_AB	—	—
M34	—	—	O	HDOUTP_AB	—	—
V33	—	—	VSST	VSSOB_AB	—	—
M33	—	—	O	HDOUTN_AB	—	—
N31	—	—	VDDOB	VDDOB_AB	—	—
M31	—	—	VSSRX	VSSRX_AB	—	—
L34	—	—	I	HDINP_AB	—	—
V34	—	—	VSST	VSSIB_AB	—	—
L33	—	—	I	HDINN_AB	—	—
N30	—	—	VDDIB	VDDIB_AB	—	—
M30	—	—	VDDOB	VDDOB_AA	—	—
K34	—	—	O	HDOUTP_AA	—	—
K33	—	—	O	HDOUTN_AA	—	—
M30	—	—	VDDOB	VDDOB_AA	—	—
L32	—	—	VDDT	VDDTX_AA	—	—
L31	—	—	VSSRX	VSSRX_AA	—	—
P32	—	—	VDDR	VDDRX_AA	—	—
J34	—	—	I	HDINP_AA	—	—
J33	—	—	I	HDINN_AA	—	—
R32	—	—	VDDR	VDDRX_AA	—	—
L30	—	—	VDDIB	VDDIB_AA	—	—
K31	—	—	—	REFCLKP_A	—	—
K30	—	—	—	REFCLKN_A	—	—
J31	—	—	O	REXTN_A	—	—
J30	—	—	O	REXT_A	—	—
Y32	—	—	VDDR	VDDAUX_A	—	—
G34	—	—	VDDGB_A	VDDGB_A	—	—
G33	—	—	VSSGB_A	VSSGB_A	—	—
G32	—	—	O	ATMOUT_A (no connect)	—	—
G31	—	—	I	PRESERVE01	—	—
F33	—	—	I	PRESERVE02	—	—
G30	—	—	I	PRESERVE03	—	—

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
F31	—	—	O	PSYS_RSSIG_ALL	—	—
F30	—	—	I	PSYS_DOBISTN	—	—
E31	—	—	VDD33	VDD33	—	—
AB17	—	—	VDD15	VDD15	—	—
AB18	—	—	VDD15	VDD15	—	—
D32	—	—	I	PBIST_TEST_ENN	—	—
E30	—	—	I	PLOOP_TEST_ENN	—	—
AB19	—	—	VDD15	VDD15	—	—
D31	—	—	I	PASB_PDN	—	—
C32	—	—	I	PMP_TESTCLK	—	—
C31	—	—	VDD33	VDD33	—	—
AJ32	—	—	VDD15	VDD15	—	—
B32	—	—	I	PASB_RESETN	—	—
A33	—	—	I	PASB_TRISTN	—	—
B31	—	—	I	PMP_TESTCLK_ENN	—	—
A32	—	—	I	PASB_TESTCLK	—	—
AK32	—	—	VDD15	VDD15	—	—
AB21	—	—	VSS	VSS	—	—
A31	—	—	VDD33	VDD33	—	—
B30	1 (TC)	7	IO	PT36D	—	—
AB22	—	—	VSS	VSS	—	—
C30	1 (TC)	7	IO	PT36B	—	—
D30	1 (TC)	7	IO	PT35D	—	—
B13	1 (TC)	—	VDDIO1	VDDIO1	—	—
E29	1 (TC)	7	IO	PT35B	—	—
E28	1 (TC)	7	IO	PT34D	VREF_1_07	—
AN33	—	—	VSS	Vss	—	—
D29	1 (TC)	8	IO	PT34B	—	—
B29	1 (TC)	8	IO	PT33D	—	L1C_A0
C29	1 (TC)	8	IO	PT33C	VREF_1_08	L1T_A0
B15	1 (TC)	—	VDDIO1	VDDIO1	—	—
E27	1 (TC)	8	IO	PT32D	—	L2C_A0
E26	1 (TC)	8	IO	PT32C	—	L2T_A0
AP34	—	—	Vss	Vss	—	—
A30	1 (TC)	8	IO	PT32B	—	—
A29	1 (TC)	9	IO	PT31D	—	L3C_D3
E25	1 (TC)	9	IO	PT31C	VREF_1_09	L3T_D3
B17	1 (TC)	—	VDDIO1	VDDIO1	—	—
E24	1 (TC)	9	IO	PT31A	—	—
B28	1 (TC)	9	IO	PT30D	—	L4C_A0
C28	1 (TC)	9	IO	PT30C	—	L4T_A0
B2	—	—	Vss	Vss	—	—
D28	1 (TC)	9	IO	PT30A	—	—
C27	1 (TC)	9	IO	PT29D	—	L5C_A0

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
D27	1 (TC)	9	IO	PT29C	—	L5T_A0
E23	1 (TC)	9	IO	PT29B	—	L6C_A0
E22	1 (TC)	9	IO	PT29A	—	L6T_A0
D26	1 (TC)	1	IO	PT28D	—	L7C_A0
D25	1 (TC)	1	IO	PT28C	—	L7T_A0
B33	—	—	Vss	Vss	—	—
D24	1 (TC)	1	IO	PT28B	—	L8C_A0
D23	1 (TC)	1	IO	PT28A	—	L8T_A0
C26	1 (TC)	1	IO	PT27D	VREF_1_01	L9C_A0
C25	1 (TC)	1	IO	PT27C	—	L9T_A0
D11	1 (TC)	—	VDDIO1	VDDIO1	—	—
E21	1 (TC)	1	IO	PT27B	—	L10C_A0
E20	1 (TC)	1	IO	PT27A	—	L10T_A0
D22	1 (TC)	2	IO	PT26D	—	L11C_A0
D21	1 (TC)	2	IO	PT26C	VREF_1_02	L11T_A0
E34	—	—	Vss	Vss	—	—
A28	1 (TC)	2	IO	PT26B	—	—
B26	1 (TC)	2	IO	PT25D	—	L12C_A0
B25	1 (TC)	2	IO	PT25C	—	L12T_A0
D13	1 (TC)	—	VDDIO1	VDDIO1	—	—
B27	1 (TC)	2	IO	PT25B	—	—
A27	1 (TC)	3	IO	PT24D	—	L13C_A0
A26	1 (TC)	3	IO	PT24C	VREF_1_03	L13T_A0
N13	—	—	Vss	Vss	—	—
C24	1 (TC)	3	IO	PT24B	—	—
C22	1 (TC)	3	IO	PT23D	—	L14C_A0
C23	1 (TC)	3	IO	PT23C	—	L14T_A0
D15	1 (TC)	—	VDDIO1	VDDIO1	—	—
B24	1 (TC)	3	IO	PT23B	—	—
D20	1 (TC)	3	IO	PT22D	—	L15C_A0
D19	1 (TC)	3	IO	PT22C	—	L15T_A0
N14	—	—	Vss	Vss	—	—
E19	1 (TC)	3	IO	PT22B	—	L16C_A0
E18	1 (TC)	3	IO	PT22A	—	L16T_A0
C21	1 (TC)	4	IO	PT21D	—	L17C_A0
C20	1 (TC)	4	IO	PT21C	—	L17T_A0
A25	1 (TC)	4	IO	PT21B	—	L18C_A0
A24	1 (TC)	4	IO	PT21A	—	L18T_A0
B23	1 (TC)	4	IO	PT20D	—	L19C_A0
A23	1 (TC)	4	IO	PT20C	—	L19T_A0
N15	—	—	Vss	Vss	—	—
E17	1 (TC)	4	IO	PT20B	—	L20C_A0
E16	1 (TC)	4	IO	PT20A	—	L20T_A0
B22	1 (TC)	4	IO	PT19D	—	L21C_A0

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
B21	1 (TC)	4	IO	PT19C	VREF_1_04	L21T_A0
C18	1 (TC)	4	IO	PT19B	—	L22C_A0
C19	1 (TC)	4	IO	PT19A	—	L22T_A0
N20	—	—	Vss	Vss	—	—
A22	1 (TC)	5	IO	PT18D	PTCK1C	L23C_A0
A21	1 (TC)	5	IO	PT18C	PTCK1T	L23T_A0
N21	—	—	Vss	Vss	—	—
D17	1 (TC)	5	IO	PT18B	—	L24C_A0
D18	1 (TC)	5	IO	PT18A	—	L24T_A0
B20	1 (TC)	5	IO	PT17D	PTCK0C	L25C_A0
B19	1 (TC)	5	IO	PT17C	PTCK0T	L25T_A0
A20	1 (TC)	5	IO	PT17B	—	L26C_A0
A19	1 (TC)	5	IO	PT17A	—	L26T_A0
A18	1 (TC)	5	IO	PT16D	VREF_1_05	L27C_A0
B18	1 (TC)	5	IO	PT16C	—	L27T_A0
Y21	—	—	Vss	Vss	—	—
C17	1 (TC)	5	IO	PT16B	—	L28C_D0
D16	1 (TC)	5	IO	PT16A	—	L28T_D0
A17	1 (TC)	6	IO	PT15D	—	L29C_D0
B16	1 (TC)	6	IO	PT15C	—	L29T_D0
E15	1 (TC)	6	IO	PT15B	—	L30C_A0
E14	1 (TC)	6	IO	PT15A	—	L30T_A0
A16	1 (TC)	6	IO	PT14D	—	L31C_A0
A15	1 (TC)	6	IO	PT14C	VREF_1_06	L31T_A0
Y22	—	—	Vss	Vss	—	—
D14	1 (TC)	6	IO	PT14B	—	—
C16	0 (TL)	1	IO	PT13D	MPI_RTRY_N	L1C_A0
C15	0 (TL)	1	IO	PT13C	MPI_ACK_N	L1T_A0
D7	0 (TL)	—	VDDIO0	VDDIO0	—	—
C14	0 (TL)	1	IO	PT13B	—	L2C_A0
B14	0 (TL)	1	IO	PT13A	VREF_0_01	L2T_A0
A14	0 (TL)	1	IO	PT12D	M0	L3C_A0
A13	0 (TL)	1	IO	PT12C	M1	L3T_A0
AA20	—	—	Vss	Vss	—	—
E12	0 (TL)	2	IO	PT12B	MPI_CLK	L4C_A0
E13	0 (TL)	2	IO	PT12A	A21/MPI_BURST_N	L4T_A0
C13	0 (TL)	2	IO	PT11D	M2	L5C_A0
C12	0 (TL)	2	IO	PT11C	M3	L5T_A0
B12	0 (TL)	2	IO	PT11B	VREF_0_02	L6C_A0
A12	0 (TL)	2	IO	PT11A	MPI_TEA_N	L6T_A0
D12	0 (TL)	3	IO	PT10D	—	L7C_D0
C11	0 (TL)	3	IO	PT10C	—	L7T_D0
B11	0 (TL)	3	IO	PT10B	—	—
A11	0 (TL)	3	IO	PT9D	VREF_0_03	L8C_A0

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
A10	0 (TL)	3	IO	PT9C	—	L8T_A0
AA21	—	—	Vss	Vss	—	—
B10	0 (TL)	3	IO	PT9B	—	—
E11	0 (TL)	3	IO	PT8D	D0	L9C_D0
D10	0 (TL)	3	IO	PT8C	TMS	L9T_D0
C10	0 (TL)	3	IO	PT8B	—	—
A9	0 (TL)	4	IO	PT7D	A20/MPI_BDIP_N	L10C_A0
B9	0 (TL)	4	IO	PT7C	A19/MPI_TSZ1	L10T_A0
AA22	—	—	Vss	Vss	—	—
E10	0 (TL)	4	IO	PT7B	—	—
A8	0 (TL)	4	IO	PT6D	A18/MPI_TSZ0	L11C_A0
B8	0 (TL)	4	IO	PT6C	D3	L11T_A0
D9	0 (TL)	4	IO	PT6B	VREF_0_04	L12C_D0
C8	0 (TL)	4	IO	PT6A	—	L12T_D0
E9	0 (TL)	5	IO	PT5D	D1	L13C_D0
D8	0 (TL)	5	IO	PT5C	D2	L13T_D0
AB13	—	—	Vss	Vss	—	—
A7	0 (TL)	5	IO	PT5B	—	L14C_A0
A6	0 (TL)	5	IO	PT5A	VREF_0_05	L14T_A0
C7	0 (TL)	5	IO	PT4D	TDI	L15C_D0
B6	0 (TL)	5	IO	PT4C	TCK	L15T_D0
E8	0 (TL)	5	IO	PT4B	—	L16C_A0
E7	0 (TL)	5	IO	PT4A	—	L16T_A0
A5	0 (TL)	6	IO	PT3D	—	L17C_A0
B5	0 (TL)	6	IO	PT3C	VREF_0_06	L17T_A0
AB14	—	—	Vss	Vss	—	—
C6	0 (TL)	6	IO	PT3B	—	L18C_A0
D6	0 (TL)	6	IO	PT3A	—	L18T_A0
C4	0 (TL)	6	IO	PT2D	PLL_CK1C/PPLL	L19C_A0
B4	0 (TL)	6	IO	PT2C	PLL_CK1T/PPLL	L19T_A0
A4	0 (TL)	6	IO	PT2B	—	L20C_A0
A3	0 (TL)	6	IO	PT2A	—	L20T_A0
D5	—	—	O	PCFG_MPI_IRQ	CFG_IRQ_N/MPI_IRQ_N	—
E6	—	—	IO	PCCLK	CCLK	—
D4	—	—	IO	PDONE	DONE	—
E5	—	—	VDD33	VDD33	—	—
AB15	—	—	Vss	Vss	—	—
AL33	—	—	VDD15	VDD15	—	—
AL34	—	—	VDD15	VDD15	—	—
AM34	—	—	VDD15	VDD15	—	—
AN34	—	—	VDD15	VDD15	—	—
B34	—	—	VDD15	VDD15	—	—
C33	—	—	VDD15	VDD15	—	—
C34	—	—	VDD15	VDD15	—	—

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
D33	—	—	VDD15	VDD15	—	—
D34	—	—	VDD15	VDD15	—	—
E32	—	—	VDD15	VDD15	—	—
E33	—	—	VDD15	VDD15	—	—
F32	—	—	VDD15	VDD15	—	—
F34	—	—	VDD15	VDD15	—	—
N16	—	—	VDD15	VDD15	—	—
N17	—	—	VDD15	VDD15	—	—
N18	—	—	VDD15	VDD15	—	—
N19	—	—	VDD15	VDD15	—	—
P16	—	—	VDD15	VDD15	—	—
P17	—	—	VDD15	VDD15	—	—
P18	—	—	VDD15	VDD15	—	—
P19	—	—	VDD15	VDD15	—	—
R16	—	—	VDD15	VDD15	—	—
R17	—	—	VDD15	VDD15	—	—
R18	—	—	VDD15	VDD15	—	—
R19	—	—	VDD15	VDD15	—	—
T13	—	—	VDD15	VDD15	—	—
T14	—	—	VDD15	VDD15	—	—
T15	—	—	VDD15	VDD15	—	—
T20	—	—	VDD15	VDD15	—	—
T21	—	—	VDD15	VDD15	—	—
T22	—	—	VDD15	VDD15	—	—
U13	—	—	VDD15	VDD15	—	—
U14	—	—	VDD15	VDD15	—	—
U15	—	—	VDD15	VDD15	—	—
U20	—	—	VDD15	VDD15	—	—
U21	—	—	VDD15	VDD15	—	—
U22	—	—	VDD15	VDD15	—	—
V13	—	—	VDD15	VDD15	—	—
V14	—	—	VDD15	VDD15	—	—
V15	—	—	VDD15	VDD15	—	—
V20	—	—	VDD15	VDD15	—	—
V21	—	—	VDD15	VDD15	—	—
V22	—	—	VDD15	VDD15	—	—
W13	—	—	VDD15	VDD15	—	—
W14	—	—	VDD15	VDD15	—	—
W15	—	—	VDD15	VDD15	—	—
W20	—	—	VDD15	VDD15	—	—
W21	—	—	VDD15	VDD15	—	—
W22	—	—	VDD15	VDD15	—	—
Y16	—	—	VDD15	VDD15	—	—
Y17	—	—	VDD15	VDD15	—	—

Table 32. ORT82G5 680-Pin PBGAM Pinout (Continued)

BM680	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	BM680 Pair
Y18	—	—	VDD15	VDD15	—	—
Y19	—	—	VDD15	VDD15	—	—
T32	—	—	NC	NC	—	—
W32	—	—	NC	NC	—	—

Package Thermal Characteristics Summary

There are three thermal parameters that are in common use: ΘJA, ψJC, and ΘJC. 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.):

$$\Theta_{JA} = \frac{T_J - T_A}{Q} \tag{1}$$

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/W.

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:

$$\psi_{JC} = \frac{T_J - T_C}{Q} \tag{2}$$

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:

$$\Theta_{JC} = \frac{T_J - T_C}{Q} \tag{3}$$

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. It is defined by:

$$\Theta_{JB} = \frac{T_J - T_B}{Q} \qquad (4)$$

where TB 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, TAmax, and the power dissipated by the device, Q (expressed in °C), the maximum junction temperature is approximated by:

$$T_{Jmax} = T_{Amax} + (Q \cdot \Theta_{JB}) \qquad (5)$$

Table 33. lists the thermal characteristics for all packages used with the ORCA ORT82G5 Series of FPSCs.

Package Thermal Characteristics

Table 33. ORCA ORT82G5 Plastic Package Thermal Guidelines

	ΘJA (°C/W)			Maximum Power
Package	**0 fpm**	**200 fpm**	**500 fpm**	**T = 70 °C Max, TJ = 125 °C Max, 0 fpm**
680-Pin PBGAM*	9.8	7.8	6.8	4.1

Note: The 680-pin PBGAM package for the ORT82G5 includes a heat spreader.

Package Coplanarity

The coplanarity limits of the Lattice packages are as follows:

- PBGAM: 8.0 mils

Heat Sink Vendors for BGA Packages

The estimated worst-case power requirements for the ORT82G5 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 34. Heat Sink Vendors

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

Package Parasitics

The electrical performance of an IC package, such as signal quality and noise sensitivity, is directly affected by the package parasitics. Table 35 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.

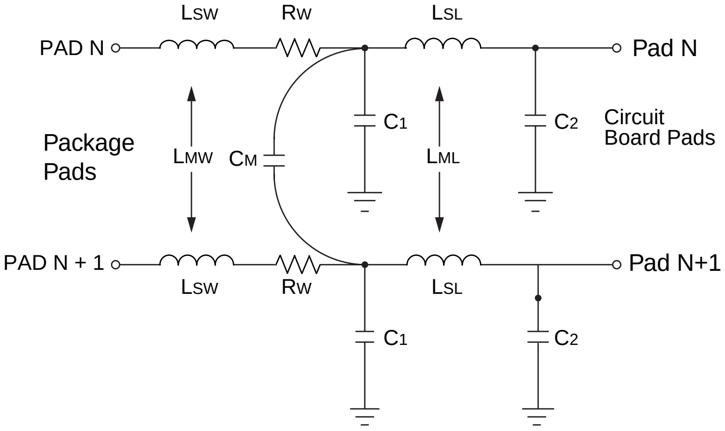
Four inductances in nH are listed: L_{SW} and L_{SL}, the self-inductance of the lead; and L_{MW} and L_{ML}, the mutual inductance to the nearest neighbor lead. These parameters are important in determining ground bounce noise and inductive crosstalk noise. Three capacitances in pF are listed: C_M, 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 35 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 35. ORCA ORT82G5 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	0.5 -1

Figure 33. Package Parasitics



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.

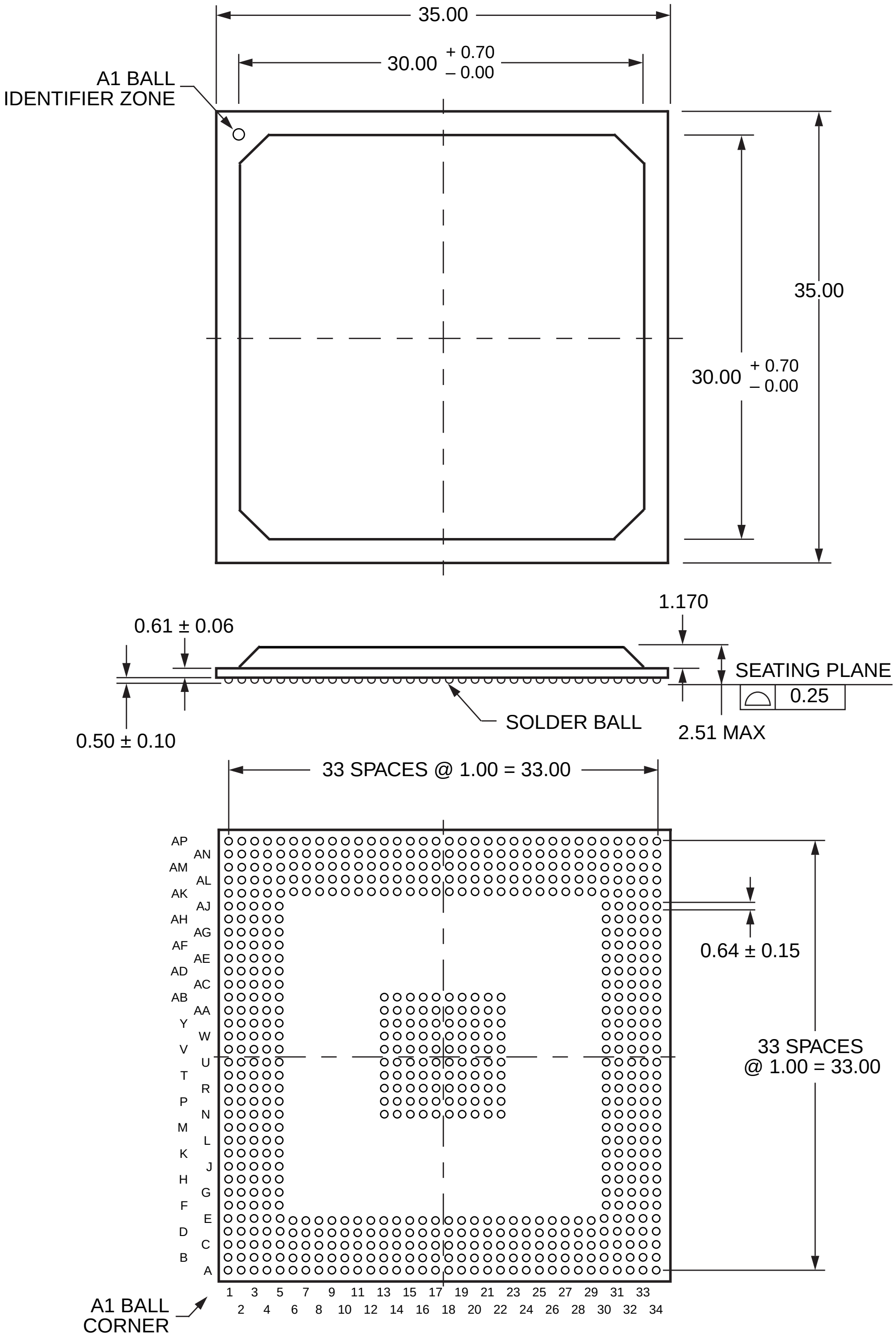
Typical (TYP): When specified after a dimension, this indicates the repeated design size if a tolerance is specified or repeated basic size if a tolerance is not specified.

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.

Figure 34. 680-Pin PBGAM

Dimensions are in millimeters.



Ordering Information

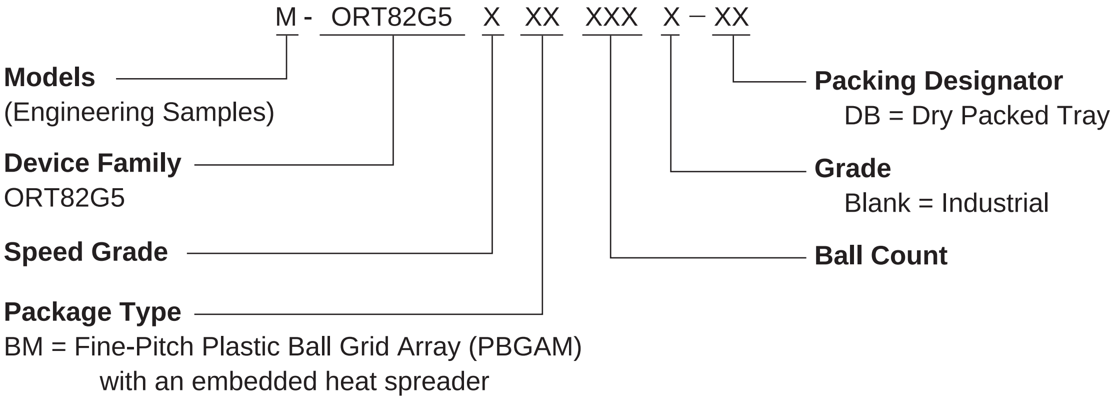


Table 36. Device Type Options

Device	Voltage
ORT82G5	1.5 V core 3.3 V/2.5 V/1.8 V/ 1.5 V I/O

Table 37. Temperature Range

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

Note: Device junction temperature of –40 ˚C to +125 ˚C are recommended

Table 38. Ordering Information

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade	Packing Designator
ORT82G5	M-ORT82G52BM680-DB	2	PBGAM	680	I	DB
	M-ORT82G51BM680-DB	1	PBGAM	680	I	DB

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