Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details	
Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	10368
Total RAM Bits	113664
Number of I/O	372
Number of Gates	643000
Voltage - Supply	1.425V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	680-BBGA
Supplier Device Package	680-FPBGA (35x35)
Purchase URL	https://www.e-xfl.com/product-detail/lattice-semiconductor/ort82g5-2f680c

Description

What is an FPSC?

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

FPSC Overview

Lattice's Series 4 FPSCs are created from Series 4 ORCA FPGAs. To create a Series 4 FPSC, several columns of Programmable Logic Cells (see FPGA Logic Overview section for FPGA logic details) are added to an embedded logic core. Other than replacing some FPGA gates with ASIC gates, at greater than 10:1 efficiency, none of the FPGA functionality is changed—all of the Series 4 FPGA capability is retained 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 Lattice 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, complied Verilog simulation models, 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 at www.latticesemi.com or contacting a local sales office.

FPGA Logic Overview

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

The hierarchical architecture of the logic, clocks, routing, RAM, and system-level blocks create a seamless merge of FPGA and ASIC designs. Modular hardware and software technologies enable System-on-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 block-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-block 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

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 ORT42G5 and 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 ORT42G5 and ORT82G5 to work in an independent system without an external microprocessor interface.

Backplane Transceiver Core Detailed Description

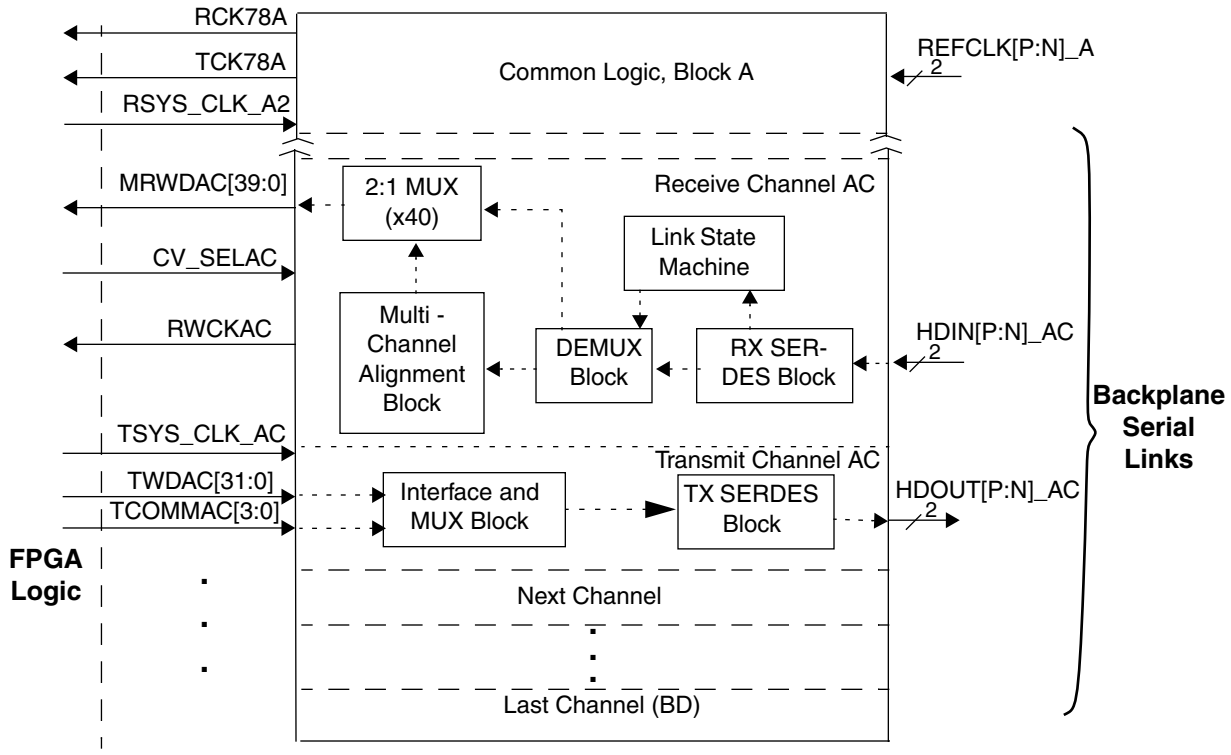
The following sections describe the various logic blocks in the Embedded Core portion of the FPSC. The FPGA section of the FPSC is identical to an ORCA OR4E04 FPGA except that the pads on one edge of the FPGA chip are replaced by the Embedded Core. For a detailed description of the programmable logic functions, please see the ORCA Series 4 FPGA Data Sheet and related application and technical notes.

The major functional blocks in the Embedded Core include:

- Two SERializer-DESerializer (SERDES) blocks and Clock and Data Recovery (CDR) circuitry
- 8b/10b encoder/decoders
- Transmit pre-emphasis circuitry
- 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 Table 41 and the signals at the Transceiver Embedded Core/FPGA interface for the ORT42G5 are listed in Table 8, Table 9 and Table 11; and for the ORT82G5, in Table 8, Table 10 and Table 12.

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



The Embedded Core provides transceiver functionality for four or eight serial data channels and is organized into two blocks, each supporting two or four channels. Each channel is identified by both a block identifier [A:B] and a channel identifier [A:D]. In the ORT42G5 only the channel identifiers C and D are used. (This naming convention follows that of the ORT82G5).

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 and optionally 8b/10b decoded. The receiver also optionally recognizes the comma characters or code violations and aligns the bit stream to the proper word boundary. The resulting parallel data is optionally passed to the multi-channel alignment block before it is presented to the FPGA logic.

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 a negative disparity present state. 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.

- FMPU_SYNMODE_B = 11111111 (Register Location 30911)

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

- FMPU_STR_EN_xx = 1 enabled
- FMPU_STR_EN_xx = 0 disabled
- (Register Location 30810 and 30910, where xx is one of AC, AD, BC or BD.)

To resynchronize a multichannel alignment group set the following bit to zero, and then set it to one.

- FMPU_RESYNC4 for four channels, AC, AD, BC and BD. (Register Location 30A02, bit 2)
- FMPU_RESYNC2A for dual channels, AC and AD. (Register Location 30820, bit 5)
- FMPU_RESYNC2B for block channels, BC and BD. (Register Location 30920, bit 5)

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 one.

FMPU_RESYNC1_xx (Register Locations 30820 and 30920, bits 2 and 3, where xx is one of AC, AD, BC or BD).

ORT82G5 Configuration

Register settings for multi-channel alignment are shown in Table 7.

Table 7. Multi-channel Alignment Modes

Register Bits FMPU_SYNMODE_xx[0:1]	Mode
00	No multi-channel 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 multi-channel alignment group set the following bit to zero, and then set it to one:

- 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]

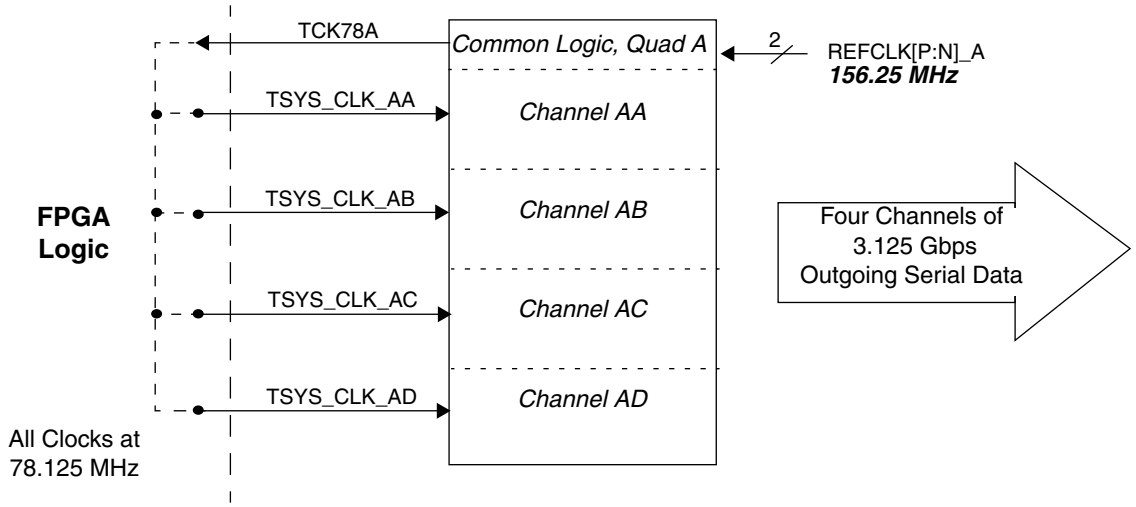
Table 17. TCK78[A:B] Source Selection

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

Recommended Transmit Clock Distribution for the ORT82G5

As an example of the recommended clock distribution approach, TSYS_CLK_A[A:D] can be sourced by TCK78A as shown in Figure 25 if the transmit line rate are common for all four channels in a quad. Similar clocking would be used for Quad B.

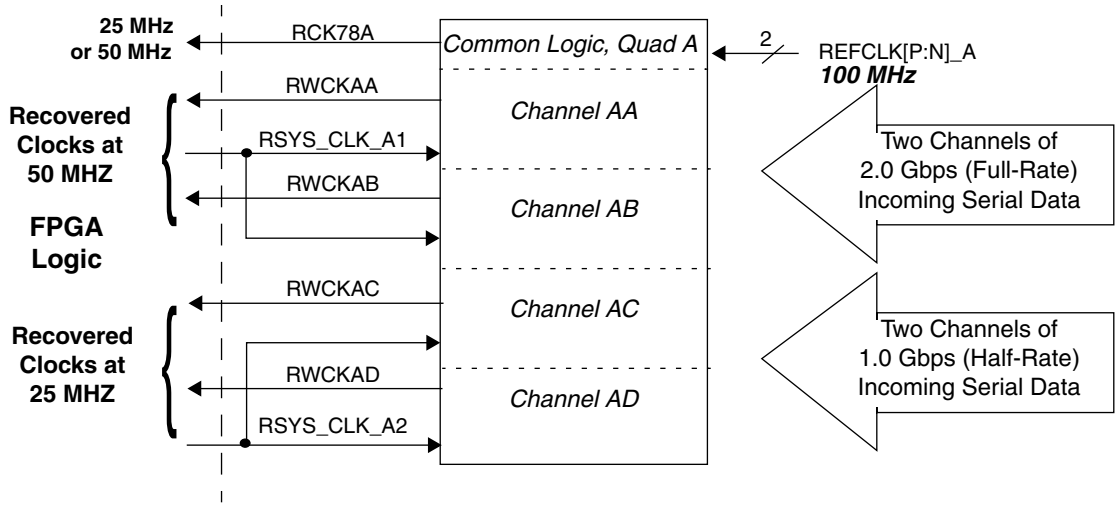
Figure 25. Transmit Clocking for a Single Block (Similar Connections Would Be Used for Block B)



If the transmit line rate is mixed between half and full rate among the channels, then the scheme shown in Figure 26 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.

The receive channel alignment bypass mode allows mixing of half and full line rates among the channels, as shown in Figure 28. The figure shows channel pair AA and AB configured in full rate mode at 2.0 Gbps. Channel pair AC and AD are configured in half-rate mode at 1.0 Gbps.

Figure 28. Receive Clocking for Mixed Line Rates



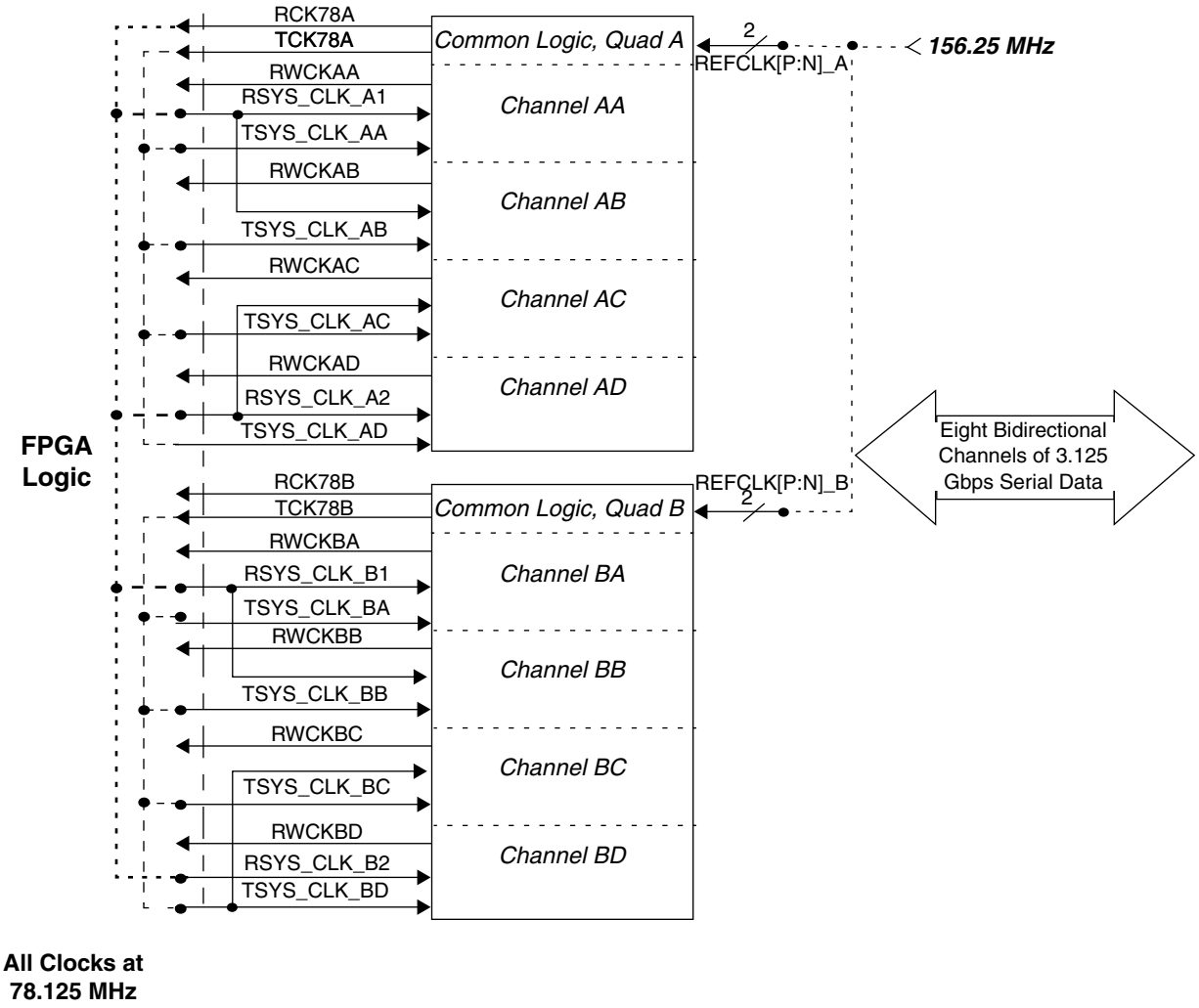
As noted in the caption of Figure 28, each quad can be configured in any line rate (0.6 to 3.7 Gbps), since each quad has its own reference clock input pins. The receive alignment FIFO per channel cannot be used in this mode.

Multi-Channel Alignment Clocking Strategies for the ORT82G5

The data on the eight channels (four 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 Gbps. 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. 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 29.

Figure 31. 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 1.5V. 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.

Test Modes

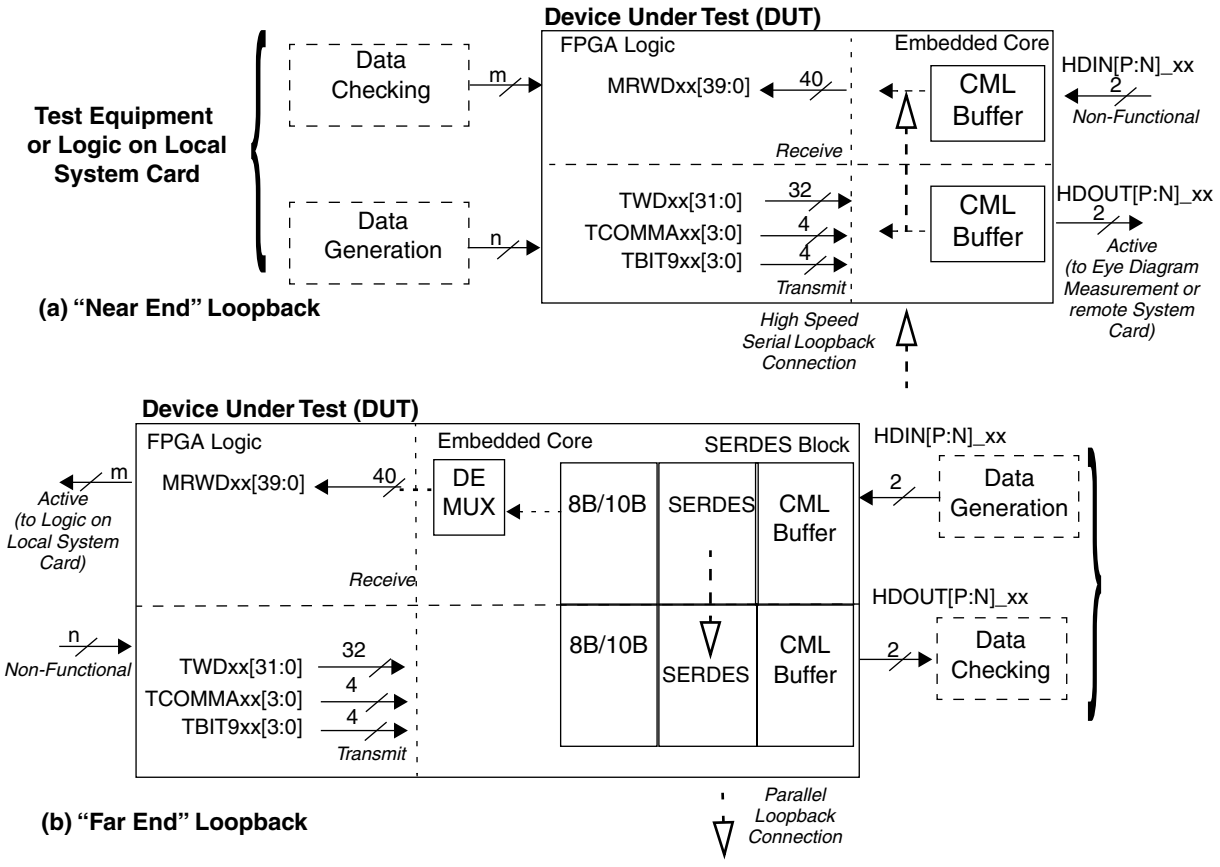
In addition to the operational logic described in the preceding sections, the Embedded Core contains logic to support various test modes - both for device validation and evaluation and for operating system level tests. The following sections discuss two of the test support logic blocks, supporting various loopback modes and SERDES characterization.

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 32(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 32(b)), the generating and checking functions are performed remotely, either by test equipment or a remote system card.

Figure 32. “Near End” vs. “Far End” Loopback



The loopback mode can also be characterized by the physical location of the loopback connection. There are three 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)

The three 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 32(a).

The data rate selection bits, TXHR and RXHR, in the channel configuration registers must be configured to carry the same value. Table 19 and Table 20 summarize the settings of the control interface register configuration bits for high-speed serial loopback.

Table 19. High-Speed Serial Loopback Configuration Bit Definitions for the ORT42G5

Register Address	Bit Value	Bit Name	Comments
30022, 30032, 30122, 30132	Bit 0 = 0 or 1	TXHR	Set to 0 or 1. TXHR and RXHR bits must be set to the same value.
	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.
30023, 30033, 30123, 30133	Bit 0 = 0 or 1	RXHR	Set to 0 or 1. TXHR and RXHR bits must be set to the same value.
	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.
30801, 30901	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.

Table 20. High-Speed Serial Loopback Configuration Bit Definitions for the ORT82G5

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.
	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.
	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.

Table 24. 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 25. The channel that is observed is also based on the decoding of SCHAR_CHAN as shown in Table 25.

Table 25. 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 blocks (labeled A and B) built-into the Embedded ASIC Core (EAC). Each memory block has a capacity of 4K words by 36 bits. These two memory blocks (also called “slices”) are in addition to the block RAMs found in the FPGA portion of the ORT82G5.

Although the memory blocks/slices are in the EAC part of the chip, they do not interact with the rest of the EAC circuits, but are standalone memories designed specifically to increase RAM capacity in the ORT82G5 chip. They can be used by logic implemented in the FPGA portion of the FPSC. Figure 34 represents one of the two available memory slices built into the EAC. The index “x” refers to the memory slice (x=A for slice A, x=B for slice B). Each memory slice is organized into two sections, which are also labeled as A and B. In Figure 34, SDRAM A is one section of slice x, and SDRAM B is another section of slice x. Data can be written to both sections of a slice independently. However, a read access can access only one of sections A or B at any given time (CSR_x=0 selects section A, CSR_x=1 selects section B).

The 36 bits written to or read from the memory slice are composed of 32 bits of data (bits 31:24, 23:16, 15:8, 7:0), and 4 bits of parity (bits 35,34,33,32). The core performs no parity checking functions. The data read from the memory is registered so that it works as a pipelined synchronous memory block.

For illustration purposes, assuming that the memory slice in Figure 34 is slice A (x=A), then certain signals apply to both sections of slice A. These include D_A[35:0], CKW_A, AW_A[10:0], BYTEWN_A[3:0], Q_A[35:0], CKR_A, CSR_A, and AR_A[10:0]. The BYTEWN_A[3:0] are byte and parity write enable bits for each byte and parity bit of data being written.

BYTEWN_A[3] is associated with D_A[35,31:24].
BYTEWN_A[2] is associated with D_A[34,23:16].
BYTEWN_A[1] is associated with D_A[33,15:8].
BYTEWN_A[0] is associated with D_A[32,7:0].

The signals that are unique to each section of slice A are:

CSWA_A --enables writing to section A of slice A
CSWB_A -- enables writing to section B of slice A

External Reference Clock

The external reference clock selection and its interface are a critical part of system applications for this product. Table 38 specifies reference clock requirements, over the full range of operating conditions. The designer is encourage to read TN1040, *SERDES Reference Clock,* which discusses various aspects of this system element and its interconnection.

Table 38. Reference Clock Specifications (REFCLKP and REFCLKN)

Parameter	Min.	Typ.	Max.	Units
Frequency Range	60	—	185	MHz
Frequency Tolerance[1]	-350	—	350	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_{DDIB}	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_{DDIB}	mV_{p-p}
Input Capacitance (at REFCLKP)	—	—	5	pF
Input Capacitance (at REFCLKN)	—	—	5	pF

1. This specification indicates the capability of the high speed receiver CDR PLL to acquire lock when the reference clock frequency and incoming data rate are not synchronized.

Embedded Core Timing Characteristics

Table 39 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 39. 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.

Table 40. Pin Descriptions (Continued)

Symbol	I/O	Description
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.[1]
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.[1]
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.[1]
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.[1]
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.[1]
$\overline{\text{CS0}}$, 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.[1]
$\overline{\text{RD}}$/$\overline{\text{MPI\_STRB}}$	I	$\overline{\text{RD}}$ is used in the asynchronous peripheral configuration mode. A low on $\overline{\text{RD}}$ changes D[7:3] into a status output. $\overline{\text{WR}}$ and $\overline{\text{RD}}$ should not be used simultaneously. If they are, the write strobe overrides. This pin is also used as the MPI data transfer strobe. As a status indication, a high indicates ready, and a low indicates busy.
$\overline{\text{WR}}$/MPI_RW	I	$\overline{\text{WR}}$ is used in asynchronous peripheral mode. A low on $\overline{\text{WR}}$ transfers data on D[7:0] to the FPGA. In MPI mode, a high on MPI_RW allows a read from the data bus, while a low causes a write transfer to the FPGA.
	I/O	After configuration, if the MPI is not used, $\overline{\text{WR}}$/MPI_RW is a user-programmable I/O pin.[1]
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.
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.
MPI_BDIP	I	$\overline{\text{MPI\_BDIP}}$ is driven by the PowerPC processor in MPI mode. Assertion of this pin indicates that the second beat in front of the current one is requested by the master. Negated before the burst transfer ends to abort the burst data phase.
MPI_TSZ[0:1]	I	MPI_TSZ[0:1] signals are driven by the bus master in MPI mode to indicate the data transfer size for the transaction. Set 01 for byte, 10 for half-word, and 00 for word.
A[21:0]	O	During master parallel mode A[21:0] address the configuration EPROMs up to 4M bytes.
	I/O	If not used for MPI these pins are user-programmable I/O pins after configuration.[1]
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.[1]

Table 41. FPSC Function Pin Descriptions (Continued)

Symbol	I/O	Description
HDOUTP_AB (ORT82G5 only)	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 (ORT82G5 only)	O	High-speed CML transmit data output – SERDES quad B, channel A.
HDOUTP_BA (ORT82G5 only)	O	High-speed CML transmit data output – SERDES quad B, channel A.
HDOUTN_BB (ORT82G5 only)	O	High-speed CML transmit data output – SERDES quad B, channel B.
HDOUTP_BB (ORT82G5 only)	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 (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_AB (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_AC	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_AD	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_BA (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_BB (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_BC	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDIB_BD	—	1.8V/1.5V power supply for high-speed serial input buffers.
VDDOB_AA (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_AB (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_AC	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_AD	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_BA (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_BB (ORT82G5 only)	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_BC	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDOB_BD	—	1.8V/1.5V power supply for high-speed serial output buffers.
VDDGB_A	—	1.5V guard band power supply.
VDDGB_B	—	1.5V guard band power supply.
VDD_ANA	—	1.5V power supply for SERDES analog receive and transmit circuitry.

1. Should be externally connected on board to 3.3V pull-up resistor.

Table 44. ORT42G5 484-pin PBGAM (fpBGA) Pinout (Continued)

484-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	484-PBGAM
G9	-	-	VSS	VSS	-	-
L3	7 (CL)	5	IO	PL21D	A10/PPC_A24	L10C
L4	7 (CL)	5	IO	PL21C	A9/PPC_A23	L10T
L5	7 (CL)	5	IO	PL22D	A8/PPC_A22	-
F10	-	-	VDD15	VDD15	-	-
G10	-	-	VSS	VSS	-	-
M3	7 (CL)	6	IO	PL24D	PLCK1C	L11C
M4	7 (CL)	6	IO	PL24C	PLCK1T	L11T
N4	7 (CL)	6	IO	PL25C	A7/PPC_A21	-
M2	7 (CL)	6	IO	PL26D	A6/PPC_A20	L12C
M1	7 (CL)	6	IO	PL26C	A5/PPC_A19	L12T
N3	7 (CL)	7	IO	PL27D	WR_N/MPI_RW	-
F11	-	-	VDD15	VDD15	-	-
N5	7 (CL)	8	IO	PL28D	A4/PPC_A18	-
M5	7 (CL)	-	VDDIO7	VDDIO7	-	-
N2	7 (CL)	8	IO	PL29D	A3/PPC_A17	L13C
N1	7 (CL)	8	IO	PL29C	A2/PPC_A16	L13T
G11	-	-	VSS	VSS	-	-
P2	7 (CL)	8	IO	PL30D	A1/PPC_A15	L14C
P1	7 (CL)	8	IO	PL30C	A0/PPC_A14	L14T
F12	-	-	VDD15	VDD15	-	-
P3	7 (CL)	8	IO	PL31D	DP0	L15C
P4	7 (CL)	8	IO	PL31C	DP1	L15T
R4	6 (BL)	1	IO	PL32D	D8	L16C
R3	6 (BL)	1	IO	PL32C	VREF_6_01	L16T
R2	6 (BL)	1	IO	PL33D	D9	L17C
R1	6 (BL)	1	IO	PL33C	D10	L17T
G12	-	-	VSS	VSS	-	-
T3	6 (BL)	2	IO	PL34D	-	-
P5	6 (BL)	-	VDDIO6	VDDIO6	-	-
T2	6 (BL)	2	IO	PL34B	-	L18C
T1	6 (BL)	2	IO	PL34A	-	L18T
U1	6 (BL)	3	IO	PL35B	D11	L19C
U2	6 (BL)	3	IO	PL35A	D12	L19T
R5	6 (BL)	-	VDDIO6	VDDIO6	-	-
V1	6 (BL)	3	IO	PL36B	VREF_6_03	L20C
V2	6 (BL)	3	IO	PL36A	D13	L20T
G13	-	-	VSS	VSS	-	-
W2	6 (BL)	4	IO	PL37B	-	L21C
W1	6 (BL)	4	IO	PL37A	VREF_6_04	L21T
Y1	6 (BL)	4	IO	PL39D	PLL_CK7C/HPPLL	L22C
Y2	6 (BL)	4	IO	PL39C	PLL_CK7T/HPPLL	L22T
U3	-	-	I	PTEMP	PTEMP	-
F13	-	-	VDD15	VDD15	-	-

Table 44. ORT42G5 484-pin PBGAM (fpBGA) Pinout (Continued)

484-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	484-PBGAM
V9	5 (BC)	1	IO	PB17A	-	-
W9	5 (BC)	1	IO	PB17C	-	L35T
Y9	5 (BC)	1	IO	PB17D	-	L35C
U9	5 (BC)	1	IO	PB18A	-	-
AA9	5 (BC)	1	IO	PB18C	VREF_5_01	L36T
AB9	5 (BC)	1	IO	PB18D	-	L36C
G16	-	-	VDD15	VDD15	-	-
H13	-	-	VSS	VSS	-	-
AB10	5 (BC)	2	IO	PB19A	-	L37T
AA10	5 (BC)	2	IO	PB19B	-	L37C
W10	5 (BC)	2	IO	PB19C	PBCK0T	L38T
Y10	5 (BC)	2	IO	PB19D	PBCK0C	L38C
V10	5 (BC)	2	IO	PB20A	-	-
U13	5 (BC)	-	VDDIO5	VDDIO5	-	-
AB11	5 (BC)	2	IO	PB20C	VREF_5_02	L39T
AA11	5 (BC)	2	IO	PB20D	-	L39C
U10	5 (BC)	2	IO	PB21A	-	-
H6	-	-	VDD15	VDD15	-	-
Y11	5 (BC)	3	IO	PB21C	-	L40T
W11	5 (BC)	3	IO	PB21D	VREF_5_03	L40C
U11	5 (BC)	3	IO	PB22A	-	-
J7	-	-	VSS	VSS	-	-
AB12	5 (BC)	3	IO	PB22C	-	L41T
AA12	5 (BC)	3	IO	PB22D	-	L41C
U12	5 (BC)	3	IO	PB23A	-	-
Y12	5 (BC)	3	IO	PB23C	PBCK1T	L42T
W12	5 (BC)	3	IO	PB23D	PBCK1C	L42C
V11	5 (BC)	3	IO	PB24A	-	-
J8	-	-	VSS	VSS	-	-
AB13	5 (BC)	4	IO	PB24C	-	L43T
AA13	5 (BC)	4	IO	PB24D	-	L43C
V12	5 (BC)	4	IO	PB25A	-	-
U14	5 (BC)	-	VDDIO5	VDDIO5	-	-
AB14	5 (BC)	4	IO	PB25C	-	L44T
AA14	5 (BC)	4	IO	PB25D	VREF_5_04	L44C
J9	-	-	VSS	VSS	-	-
Y13	5 (BC)	5	IO	PB26C	-	L45T
W13	5 (BC)	5	IO	PB26D	VREF_5_05	L45C
U15	5 (BC)	-	VDDIO5	VDDIO5	-	-
AB15	5 (BC)	5	IO	PB27C	-	L46T
AA15	5 (BC)	5	IO	PB27D	-	L46C
AB16	5 (BC)	6	IO	PB28C	-	L47T
AA16	5 (BC)	6	IO	PB28D	VREF_5_06	L47C
H14	-	-	VDD15	VDD15	-	-

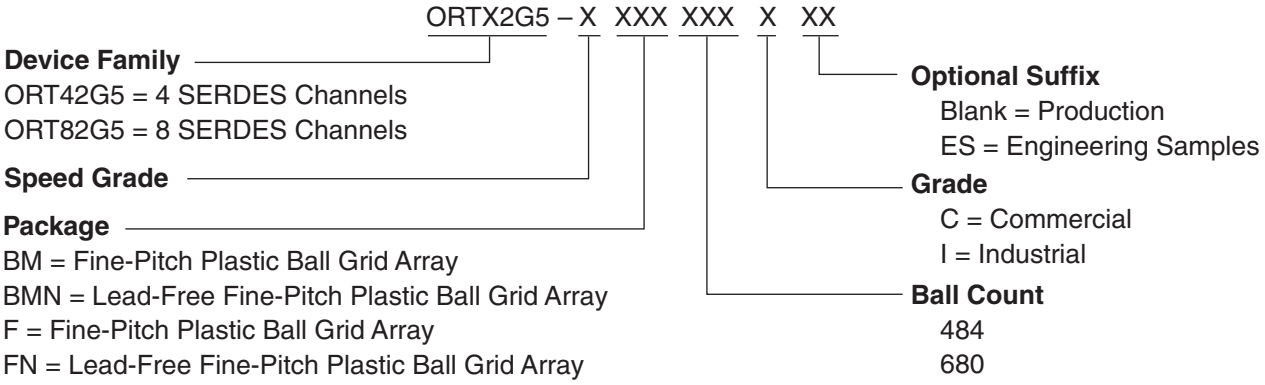
Table 45. ORT82G5 680-Pin PBGAM (fpBGA) Pinout

680-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	680-PBGAM
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
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

Table 45. ORT82G5 680-Pin PBGAM (fpBGA) Pinout (Continued)

680-PBGAM	VDDIO Bank	VREF Group	I/O	Pin Description	Additional Function	680-PBGAM
W31	—	—	VSS	VSS	—	—
V30	—	—	VDDOB	VDDOB_BD	—	—
W33	—	—	O	HDOUTN_BD	—	—
H33	—	—	VSS	VSS	—	—
W34	—	—	O	HDOUTP_BD	—	—
V31	—	—	VDDOB	VDDOB_BD	—	—
H34	—	—	VSS	VSS	—	—
J32	—	—	VSS	VSS	—	—
U31	—	—	VDDOB	VDDOB_AD	—	—
T34	—	—	O	HDOUTP_AD	—	—
M32	—	—	VSS	VSS	—	—
T33	—	—	O	HDOUTN_AD	—	—
U30	—	—	VDDOB	VDDOB_AD	—	—
T31	—	—	VSS	VSS	—	—
R34	—	—	I	HDINP_AD	—	—
N32	—	—	VSS	VSS	—	—
R33	—	—	I	HDINN_AD	—	—
T30	—	—	VDDIB	VDDIB_AD	—	—
U32	—	—	VSS	VSS	—	—
R31	—	—	VDDOB	VDDOB_AC	—	—
P34	—	—	O	HDOUTP_AC	—	—
U33	—	—	VSS	VSS	—	—
P33	—	—	O	HDOUTN_AC	—	—
R30	—	—	VDDOB	VDDOB_AC	—	—
P31	—	—	VSS	VSS	—	—
N34	—	—	I	HDINP_AC	—	—
U34	—	—	VSS	VSS	—	—
N33	—	—	I	HDINN_AC	—	—
P30	—	—	VDDIB	VDDIB_AC	—	—
V32	—	—	VSS	VSS	—	—
M34	—	—	O	HDOUTP_AB	—	—
V33	—	—	VSS	VSS	—	—
M33	—	—	O	HDOUTN_AB	—	—
N31	—	—	VDDOB	VDDOB_AB	—	—
M31	—	—	VSS	VSS	—	—
L34	—	—	I	HDINP_AB	—	—
V34	—	—	VSS	VSS	—	—
L33	—	—	I	HDINN_AB	—	—
N30	—	—	VDDIB	VDDIB_AB	—	—
K34	—	—	O	HDOUTP_AA	—	—
K33	—	—	O	HDOUTN_AA	—	—
M30	—	—	VDDOB	VDDOB_AA	—	—
L32	—	—	VDD_ANA	VDD_ANA	—	—
L31	—	—	VSS	VSS	—	—

Part Number Description



Device Type Options

Device	Voltage
ORT42G5	1.5V internal 3.3/2.5/1.8/ 1.5V I/O
ORT82G5	1.5V internal 3.3/2.5/1.8/ 1.5V I/O

Ordering Information

Conventional Packaging

Commercial[1]

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade
ORT42G5	ORT42G5-3BM484C	3	PBGAM	484	C
	ORT42G5-2BM484C	2	PBGAM	484	C
	ORT42G5-1BM484C	1	PBGAM	484	C
ORT82G5	ORT82G5-3F680C	3	PBGAM (No Heat Spreader)	680	C
	ORT82G5-2F680C	2	PBGAM (No Heat Spreader)	680	C
	ORT82G5-1F680C	1	PBGAM (No Heat Spreader)	680	C
	ORT82G5-3BM680C2[2]	3	PBGAM (With Heat Spreader)	680	C
	ORT82G5-2BM680C[2]	2	PBGAM (With Heat Spreader)	680	C
	ORT82G5-1BM680C[2]	1	PBGAM (With Heat Spreader)	680	C

1. For all but the slowest commercial speed grade, the speed grades on these devices are dual marked. For example, the commercial speed grade -2XXXXXC is also marked with the industrial grade -1XXXXXI. The commercial grade is always one speed grade faster than the associated dual mark industrial grade. The slowest commercial speed grade is marked as commercial grade only.
2. BM680 package was converted to F680 via PCN#09A-08.

Industrial[1]

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade
ORT42G5	ORT42G5-2BM484I	2	PBGAM	484	I
	ORT42G5-1BM484I	1	PBGAM	484	I
ORT82G5	ORT82G5-2F680I	2	PBGAM (No Heat Spreader)	680	I
	ORT82G5-1F680I	1	PBGAM (No Heat Spreader)	680	I
	ORT82G5-2BM680I[2]	2	PBGAM (With Heat Spreader)	680	I
	ORT82G5-1BM680I[2]	1	PBGAM (With Heat Spreader)	680	I

1. For all but the slowest commercial speed grade, the speed grades on these devices are dual marked. For example, the commercial speed grade -2XXXXXC is also marked with the industrial grade -1XXXXXI. The commercial grade is always one speed grade faster than the associated dual mark industrial grade. The slowest commercial speed grade is marked as commercial grade only.
2. BM680 package was converted to F680 via PCN#09A-08.

Lead-Free Packaging

Commercial[1]

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade
ORT42G5	ORT42G5-3BMN484C	3	Lead-Free PBGAM	484	C
	ORT42G5-2BMN484C	2	Lead-Free PBGAM	484	C
	ORT42G5-1BMN484C	1	Lead-Free PBGAM	484	C
ORT82G5	ORT82G5-3FN680C	3	Lead-Free FPGA (No Heat Spreader)[2]	680	C
	ORT82G5-2FN680C	2	Lead-Free FPGA (No Heat Spreader)[2]	680	C
	ORT82G5-1FN680C	1	Lead-Free FPGA (No Heat Spreader)[2]	680	C

1. For all but the slowest commercial speed grade, the speed grades on these devices are dual marked. For example, the commercial speed grade -2XXXXXC is also marked with the industrial grade -1XXXXXI. The commercial grade is always one speed grade faster.
2. Refer to the Thermal Management document at www.latticesemi.com for Θ_{JA} and Θ_{JC} information.

Industrial[1]

Device Family	Part Number	Speed Grade	Package Type	Ball Count	Grade
ORT42G5	ORT42G5-2BMN484I	2	Lead-Free PBGAM	484	I
	ORT42G5-1BMN484I	1	Lead-Free PBGAM	484	I
ORT82G5	ORT82G5-2FN680I	2	Lead-Free FPGA (No Heat Spreader)[2]	680	I
	ORT82G5-1FN680I	1	Lead-Free FPGA (No Heat Spreader)[2]	680	I

1. For all but the slowest commercial speed grade, the speed grades on these devices are dual marked. For example, the commercial speed grade -2XXXXXC is also marked with the industrial grade -1XXXXXI. The commercial grade is always one speed grade faster.
2. Refer to the Thermal Management document at www.latticesemi.com for Θ_{JA} and Θ_{JC} information.