Lattice Semiconductor Corporation - OR3T556BA256-DB Datasheet

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	2592
Total RAM Bits	43008
Number of I/O	223
Number of Gates	80000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C (TA)
Package / Case	256-BGA
Supplier Device Package	256-FPBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/lattice-semiconductor/or3t556ba256-db

Table of Contents

Programmable Logic Cells (continued)

BIDI Routing and SLIC Connectivity

The SLIC is connected to the rest of the PLC by the bidirectional (BIDI) routing segments and the PFU output switching segments coming from the PFU output multiplexer. The BIDI routing segments (xBID) are labeled as BL for BIDI-left and BR for BIDI-right. Each set of BR and BL xBID segments is composed of ten bidirectional lines (note that these lines are diagramed as ten input lines to the SLIC and ten output lines from the SLIC that can be used in a mutually exclusive fashion). Because the SLIC is connected directly to the outputs of the PFU, it provides great flexibility in routing via the xBID segments. The PFU routing segments, O[9:0], only connect to their respective line in the SLL, SUL, SUR, and SLR switching segment groups. That is, O9 only connects to SLL9, SUL9, SUR9, and SLR9. The BIDI lines provide the capability to connect to the other member of the routing set. That means, for example, that O9 can be routed to BR8 or BL8. This connectivity can be used as a means to distribute or gather signals on intra-PLC routing without disturbing inter-PLC resources. As described in the Switching Routing Segments subsection, the BIDI routing segments are also used for routes to a diagonally adjacent PFU.

In addition to the intra-PLC connections, the xBID and output switching segments also have connectivity to the x1, x5, and xL inter-PLC routing resources, providing an alternate routing path rather than using PLC xSW segments. These connections also provide a path to the 3-state buffers in the SLIC without encumbering the xSW segments. In this manner, buffering or 3-state control can be added to inter-PLC routing without disturbing local functionality within a PFU.

Control Signal and Fast-Carry Routing

PFU control signal and the fast-carry routing are performed using the FINS structure and several dedicated routing paths. The fast-carry (FC) routing resources consist of a dedicated bidirectional segment between each orthogonal pair of PLCs. This means that a fast-carry can go to or come from each PLC to the right or left, above or below the subject PLC. The FINS structure is used to control the switching of these fast-carry paths between the fast-carry input (FCIN) and fast-carry output (FCOUT) ports of the PFU.

The PFU control inputs (CE, SEL, LSR, ASWE) and CIN can be reached via the FINS by two special routing segments, E1 and E2. The E1 routing segment provides connectivity between all of the xBID routing segments and the FINS. It is unidirectional from the BIDI routing to the FINS. E1 also provides connectivity to the PFU clock input via FINS for a local clock signal. The E2 segment connects the SLIC DEC output to the FINS and to a group of CIPS that provide bidirectional connectivity with all of the BIDI routing segments. This allows the DEC signal to be used in the PFU and/or routed on the BIDI segments. It also allows signals to be routed to the PFU on the xBID segments if the SLIC DEC output is not used.

There is also a dedicated routing segment from the FINS to the SLIC TRI input used for BIDI buffer 3-state control.

Programmable Input/Output Cells

(continued)

Outputs

The PIC's output drivers have programmable drive capability and slew rates. Three propagation delays (fast, slewlim, sinklim) are available on output drivers. The sinklim mode has the longest propagation delay and is used to minimize system noise and minimize power consumption. The fast and slewlim modes allow critical timing to be met.

The drive current is 12 mA sink/6 mA source for the slewlim and fast output speed selections and 6 mA sink/3 mA source for the sinklim output. Two adjacent outputs can be interconnected to increase the output sink/source current to 24 mA/12 mA.

All outputs that are not speed critical should be configured as sinklim to minimize power and noise. The number of outputs that switch simultaneously in the same direction should be limited to minimize ground bounce. To minimize ground bounce problems, locate heavily loaded output buffers near the ground pads. Ground bounce is generally a function of the driving circuits, traces on the printed-circuit board, and loads and is best determined with a circuit simulation.

At powerup, the output drivers are in slewlim mode, and the input buffers are configured as TTL-level compatible (CMOS for OR3Txxx) with a pull-up. If an output is not to be driven in the selected configuration mode, it is 3-stated.

The output buffer signal can be inverted, and the 3-state control signal can be made active-high, active-low, or always enabled. In addition, this 3-state signal can be registered or nonregistered. Additionally, there is a fast, open-drain output option that directly connects the output signal to the 3-state control, allowing the output buffer to either drive to a logic 0 or 3-state, but never to drive to a logic 1. Because there is no explicit route required to create the open-drain output, its response is very fast. Like the input side of the PIO, there are two output connections from PIC routing to the output side of the PIO, OUT1, and OUT2. These connections provide for flexible routing and can be used in data manipulation in the PIO as described in subsequent paragraphs.

An FF has been added to the output path of the PIO. The register has a local set/reset and clock enable. The LSR has the option to be synchronous or asynchronous and have priority set as clock enable over LSR or LSR over clock enable. Clocking to the output FF can come from either the system clock or the ExpressCLK associated with the PIC. The input to the FF can come from either OUT1 or OUT2, or it can be tied to VDD or GND. Additionally, the input to the FF can be inverted.

Output Multiplexing

The Series 3 PIO output FF can be combined with the new PIO logic block to perform output data multiplexing with no PLC resources required. The PIO logic block has three multiplexing modes: OUT1OUTREG, OUT2OUTREG, and OUT1OUT2. OUT1OUTREG and OUT2OUTREG are equivalent except that either OUT1 or OUT2 is MUXed with the FF, where the FF data is output on the clock phase after the active edge. The simplest multiplexing mode is OUT1OUT2. In this mode, the signal at OUT1 is output to the pad while the clock is low, and the signal on OUT2 is output to the pad when the clock is high. Figure 25 shows a simple schematic of a PIO in OUT1OUT2 mode and a general timing diagram for multiplexing an address and data signal.

Often an address will be used to generate or read a data sample from memory with the goal of multiplexing the data onto a single line. In this case, the address often precedes the data by one clock cycle. OUT1OUTREG and OUT2OUTREG modes of the PIO logic can be used to address this situation.

Because OUT1OUTREG mode is equivalent to OUT2OUTREG, only OUT2OUTREG mode is described here. Figure 26 shows a simple PIO schematic in OUT2OUTREG mode and general timing for multiplexing data with a leading address. The address signal on OUT1 is registered in the PIO FF. This delays the address so that it aligns with the data signal. The PIO logic block then sends the OUTREG signal (address) to the pad when the clock is high and the OUT2 signal (data) to the pad when the clock is low, resulting in an aligned, multiplexed signal.

Programmable Input/Output Cells

(continued)

PIO Logic Function Generator

The PIO logic block can also generate logic functions based on the signals on the OUT2 and CLK ports of the PIO. The functions are AND, NAND, OR, NOR, XOR, and XNOR. Table 10 is provided as a summary of the PIO logic options.

Table 10. PIO Logic Options

Option	Description
OUT1OUTREG	Data at OUT1 output when clock low, data at FF out when clock high.
OUT2OUTREG	Data at OUT2 output when clock low, data at FF out when clock high.
OUT1OUT2	Data at OUT1 output when clock low, data at OUT2 when clock high.
AND	Output logical AND of signals on OUT2 and clock.
NAND	Output logical NAND of signals on OUT2 and clock.
OR	Output logical OR of signals on OUT2 and clock.
NOR	Output logical NOR of signals on OUT2 and clock.
XOR	Output logical XOR of signals on OUT2 and clock.
XNOR	Output logical XNOR of signals on OUT2 and clock.

PIO Register Control Signals

As discussed in the Inputs and Outputs subsections, the PIO latches/FFs have various clock, clock enable (CE), local set/reset (LSR), and global set/reset (GSRN) controls. Table 11 provides a summary of these control signals and their effect on the PIO latches/FFs. Note that all control signals are optionally invertible.

Table 11. PIO Register Control Signals

Control Signal	Effect/Functionality
ExpressCLK	Clocks input fast-capture latch; optionally clocks output FF, or 3-state FF.
System Clock (SCLK)	Clocks input latch/FF; optionally clocks output FF, or 3-state FF.
Clock Enable (CE)	Optionally enables/disables input FF (not available for input latch mode); optionally enables/disables output FF; separate CE inversion capability for input and output.
Local Set/Reset (LSR)	Option to disable; affects input latch/FF, output FF, and 3-state FF if enabled.
Global Set/Reset (GSRN)	Option to enable or disable per PIO after initial configuration.
Set/Reset Mode	The input latch/FF, output FF, and 3-state FF are individually set or reset by both the LSR and GSRN inputs.

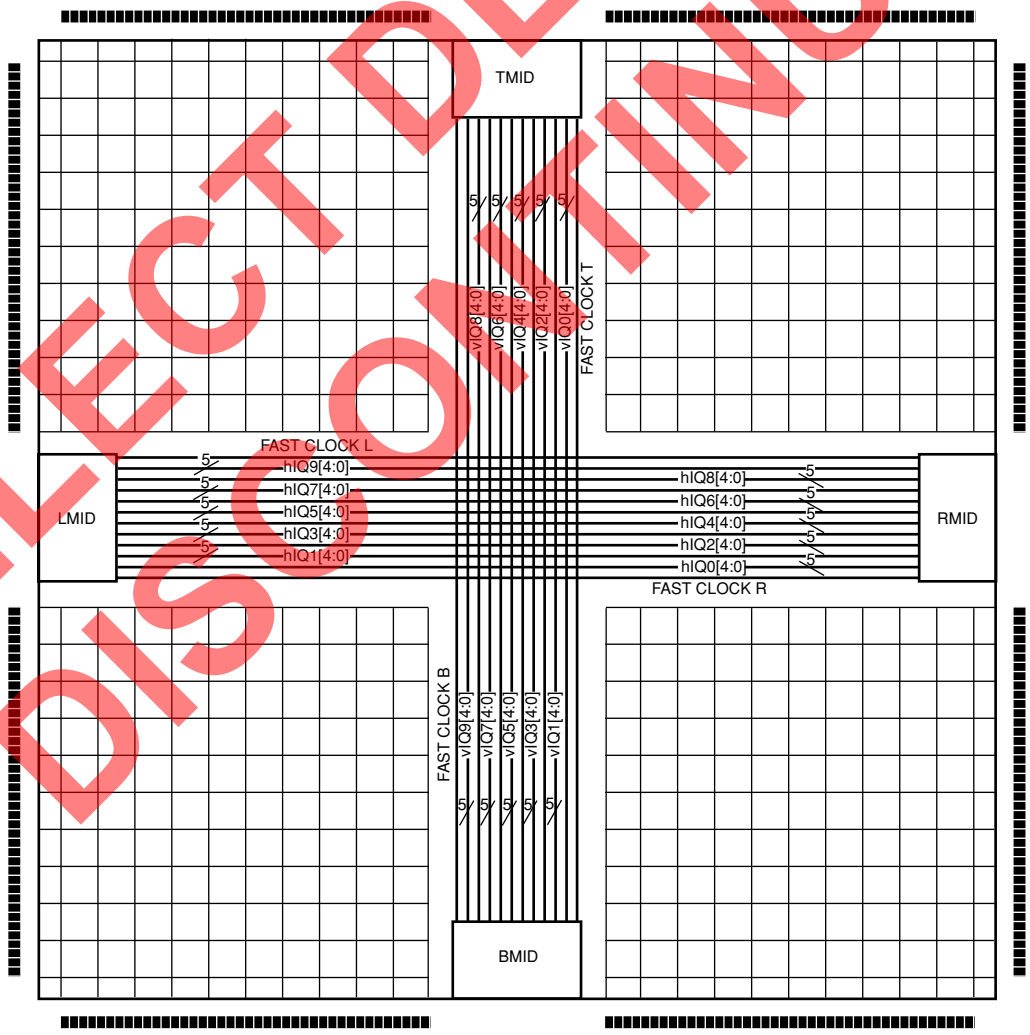
High-Level Routing Resources

The high-level routing resources in the *ORCA* Series 3 devices are interquad routing, corner cell routing, and PIC interquad routing. These resources and their related structures are discussed in the following subsections.

Interquad Routing

In the *ORCA* Series 3 devices, the PLC array is split into four equal quadrants. In between these quadrants, routing has been added to route signals between the quadrants and distribute clocks. In addition to general routing, there are four specialized clock routing spines. The general routing is discussed below, followed by the special clock routing.

One of the main purposes of interquad routing is to distribute internally generated signals, such as clocks and control signals. There are two types of interquad blocks: vertical and horizontal. Vertical interquad blocks (vIQ) run between quadrants on the left and right, while horizontal interquad blocks (hIQ) run between top and bottom quadrants. Interquad lines begin and end in the MID cells that are discussed later. Since hIQ and vIQ blocks have the same logic, only the hIQ block is described below. The interquad routing connects to x5 and xH segments. It does not affect other local routing (xsw, x1, fast carry), so local routing is the same, whether PLC-PLC connections cross quadrants or not. Figure 28 presents a (not to scale) view of interquad routing.



5-4538(F)

Figure 28. Interquad Routing

Special Function Blocks

Special function blocks in the Series 3 provide extra capabilities beyond general FPGA operation. These blocks reside in the corners and MIDs (middle inter-quad areas) of the FPGA array.

Single Function Blocks

Most of the special function blocks perform a specific dedicated function. These functions are data/configuration readback control, global 3-state control (TS_ALL), internal oscillator generation, global set/reset (GSRN), and start-up logic.

Readback Logic

The readback logic is located in the upper right corner of the FPGA and can be enabled via a bit stream option or by instantiation of a library readback component.

Readback is used to read back the configuration data and, optionally, the state of the PFU outputs. A readback operation can be done while the FPGA is in normal system operation. The readback operation cannot be daisy-chained. To use readback, the user selects options in the bit stream generator in the ispLEVER Development System.

Table 12 provides readback options selected in the bit stream generator tool. The table provides the number of times that the configuration data can be read back. This is intended primarily to give the user control over the security of the FPGA's configuration program. The user can prohibit readback (0), allow a single readback (1), or allow unrestricted readback (U).

Table 12. Readback Options

Option	Function
0	Prohibit Readback
1	Allow One Readback Only
U	Allow Unrestricted Number of Readbacks

Readback can be performed via the Series 3 microprocessor interface (MPI) or by using dedicated FPGA readback controls. If the MPI is enabled, readback via the dedicated FPGA readback logic is disabled. Readback using the MPI is discussed in the Microprocessor Interface (MPI) section.

The pins used for dedicated readback are readback data (RD_DATA), read configuration ($\overline{\text{RD\_CFG}}$), and configuration clock (CCLK). A readback operation is initiated by a high-to-low transition on $\overline{\text{RD\_CFG}}$. The $\overline{\text{RD\_CFG}}$ input must remain low during the readback operation. The readback operation can be restarted at frame 0 by driving the $\overline{\text{RD\_CFG}}$ pin high, applying at least two rising edges of CCLK, and then driving $\overline{\text{RD\_CFG}}$ low again. One bit of data is shifted out on RD_DATA at the rising edge of CCLK. The first start bit of the readback frame is transmitted out several cycles after the first rising edge of CCLK after $\overline{\text{RD\_CFG}}$ is input low (see the Readback Timing Characteristics table in the Timing Characteristics section). To be certain of the start of the readback frame, the data can be monitored for the 01 frame start bit pair.

Readback can be initiated at an address other than frame 0 via the new microprocessor interface (MPI) control registers (see the Microprocessor Interface (MPI) section for more information). In all cases, readback is performed at sequential addresses from the start address.

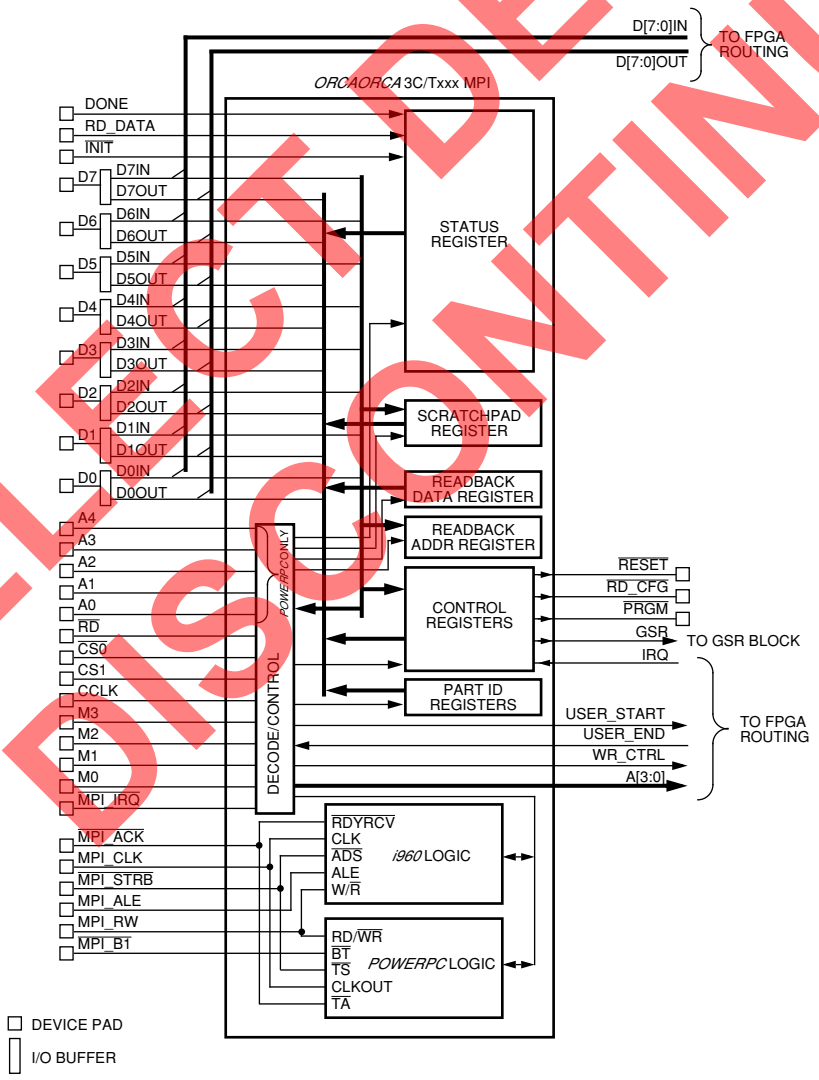
It should be noted that the RD_DATA output pin is also used as the dedicated boundary-scan output pin, TDO. If this pin is being used as TDO, the RD_DATA output from readback can be routed internally to any other pin desired. The $\overline{\text{RD\_CFG}}$ input pin is also used to control the global 3-state (TS_ALL) function. Before and during configuration, the TS_ALL signal is always driven by the $\overline{\text{RD\_CFG}}$ input and readback is disabled. After configuration, the selection as to whether this input drives the readback or global 3-state function is determined by a set of bit stream options. If used as the $\overline{\text{RD\_CFG}}$ input for readback, the internal TS_ALL input can be routed internally to be driven by any input pin.

Microprocessor Interface (MPI)

The Series 3 FPGAs have a dedicated synchronous microprocessor interface function block (see Figure 42). The MPI is programmable to operate with *PowerPC* MPC800 series microprocessors and *Intel* i960** J core processors; see Table 16 and Table 17, respectively, for compatible processors. The MPI implements an 8-bit interface to the host processor (*PowerPC* or *i960*) that can be used for configuration and readback of the FPGA as well as for user-defined data processing and general monitoring of FPGA function. In addition to dedicated-function registers, the microprocessor interface allows for the control of up to 16 user registers (RAM or flip-flops) in the FPGA logic. A synchronous/asynchronous handshake procedure is used to control transactions with user logic in the FPGA array. There is also capability for the FPGA logic to interrupt the host processor either by a hard interrupt or by having the host processor poll the microprocessor interface.

The control portion of the microprocessor interface is available following powerup of the FPGA if the mode pins specify MPI mode, even if the FPGA is not yet configured. The mode pin (M[2:0]) settings can be found in the FPGA Configuration Modes section of this data sheet, and the setup and use of the MPI for configuration is discussed in the MPI Setup and Control subsection. For postconfiguration use, the MPI must be included in the configuration bit stream by using an MPI library element in your design from the *ORCA* macro library, or by setting the MP_USER bit of the MPI configuration control register prior to the start of configuration (MPI registers are discussed later).

* *Intel* and *i960* are registered trademarks of Intel Corporation.



5-5806(F)

Figure 42. MPI Block Diagram

Microprocessor Interface (MPI) (continued)

MPI Setup and Control

The MPI has a series of addressable registers that provide MPI control and status, configuration and readback data transfer, FPGA device identification, and a dedicated user scratchpad register. All registers are 8 bits wide. The address map for these registers and the user-logic address space are shown in Table 19, followed by descriptions of the register and bit functions. Note that for all registers, the most significant bit is bit 7, and the least significant bit is bit 0.

Table 19. MPI Setup and Control Registers

Address (Hex)	Register
00	Control Register 1.
01	Control Register 2.
02	Scratchpad Register.
03	Status Register.
04	Configuration/Readback Data Register.
05	Readback Address Register 1 (bits [7:0]).
06	Readback Address Register 2 (bits [15:8]).
07	Device ID Register 1 (bits [7:0]).
08	Device ID Register 2 (bits [15:8]).
09	Device ID Register 3 (bits [23:16]).
0A	Device ID Register 4 (bits [31:24]).
0B—0F	Reserved.
10—1F	User-definable Address Space.

Control Register 1

The MPI control register 1 is a read/write register. The host processor writes a control byte to configure the MPI. It is readable by the host processor to verify the status of control bits previously written.

Table 20. MPI Setup and Control Registers Descriptions

Bit #	Description
Bit 0	**GSR Input.** Setting this bit to a 1 invokes a global set/reset on the FPGA. The host processor must return this bit to a 0 to remove the GSR signal. GSR does not affect the registers at MPI addresses 0 through F hexadecimal or any configuration registers. Default state = 0.
Bit 1	**Reserved.**
Bit 2	**Reserved.**
Bit 3	**Reserved.**
Bit 4	**Reserved.**
Bit 5	**$\overline{\text{RD\_CFG}}$ Input.** Changing this bit to a 0 after configuration will initiate readback. The host processor must return this bit to a 1 to remove the $\overline{\text{RD\_CFG}}$ signal. Since this bit works exactly like the $\overline{\text{RD\_CFG}}$ input pin, please see the FPGA pin descriptions for more information on this signal. Default state = 1.
Bit 6	**Reserved.**
Bit 7	**$\overline{\text{PRGM}}$ Input.** Setting this bit to a 0 causes the FPGA to begin configuration and resets the boundary-scan circuitry. The host processor must return this bit to a 1 to remove the $\overline{\text{PRGM}}$ signal. Since this bit works exactly like the $\overline{\text{PRGM}}$ input pin (except that it does not reset the MPI), please see the FPGA pin descriptions for more information on this signal. Default state = 1.

Configuration Data Format

The ispLEVER Development System interfaces with front-end design entry tools and provides tools to produce a fully configured FPGA. This section discusses using the ispLEVER Development System to generate configuration RAM data and then provides the details of the configuration frame format.

The *ORCA* OR3Cxx and OR3Txxx Series FPGAs are bit stream compatible.

Using ispLEVER to Generate Configuration RAM Data

The configuration data bit stream defines the I/O functionality, logic, and interconnections within the FPGA. The bit stream is generated by the development system. The bit stream created by the bit stream generation tool is a series of 1s and 0s used to write the FPGA configuration RAM. It can be loaded into the FPGA using one of the configuration modes discussed later.

In the bit stream generator, the designer selects options that affect the FPGA's functionality. Using the output of the bit stream generator, **circuit_name.bit**, the development system's download tool can load the configuration data into the *ORCA* series FPGA evaluation board from a PC or workstation.

Alternatively, a user can program a PROM (such as a Serial ROM or a standard EPROM) and load the FPGA from the PROM. The development system's PROM programming tool produces a file in .mks or .exo format.

Configuration Data Frame

Configuration data can be presented to the FPGA in two frame formats: autoincrement and explicit. A detailed description of the frame formats is shown in Figure 52, Figure 53, and Table 32. The two modes are similar except that autoincrement mode uses assumed address incrementation to reduce the bit stream size, and explicit mode requires an address for each data frame. In both cases, the header frame begins with a series of 1s and a preamble of 0010, followed by a 24-bit length count field representing the total number of configuration clocks needed to complete the loading of the FPGAs.

Following the header frame is a mandatory ID frame. (Note that the ID frame was optional in the *ORCA* 2C and 2C/TxxA Series.)

The ID frame contains data used to determine if the bit stream is being loaded to the correct type of *ORCA* FPGA (i.e., a bit stream generated for an OR3T55 is being sent to an OR3T55). Error checking is always enabled for Series 3 devices, through the use of an 8-bit checksum. One bit in the ID frame also selects between the autoincrement and explicit address modes for this load of the configuration data.

A configuration data frame follows the ID frame. A data frame starts with a 01-start bit pair and ends with enough 1-stop bits to reach a byte boundary. If using autoincrement configuration mode, subsequent data frames can follow. If using explicit mode, one or more address frames must follow each data frame, telling the FPGA at what addresses the preceding data frame is to be stored (each data frame can be sent to multiple addresses).

Following all data and address frames is the postamble. The format of the postamble is the same as an address frame with the highest possible address value with the checksum set to all ones.

FPGA Configuration Modes (continued)

Asynchronous Peripheral Mode

Figure 56 shows the connections needed for the asynchronous peripheral mode. In this mode, the FPGA system interface is similar to that of a microprocessor-peripheral interface. The microprocessor generates the control signals to write an 8-bit byte into the FPGA. The FPGA control inputs include active-low $\overline{CS0}$ and active-high CS1 chip selects and $\overline{WR}$ and $\overline{RD}$ inputs. The chip selects can be cycled or maintained at a static level during the configuration cycle. Each byte of data is written into the FPGA's D[7:0] input pins. D[7:0] of the FPGA can be connected to D[7:0] of the microprocessor only if a standard prom file format is used. If a .bit or .rbt file is used from ispLEVER, then the user must mirror the bytes in the .bit or .rbt file OR leave the .bit or .rbt file unchanged and connect D[7:0] of the FPGA to D[0:7] of the microprocessor.

The FPGA provides an RDY/$\overline{BUSY}$ status output to indicate that another byte can be loaded. A low on RDY/$\overline{BUSY}$ indicates that the double-buffered hold/shift registers are not ready to receive data, and this pin must be monitored to go high before another byte of data can be written. The shortest time RDY/$\overline{BUSY}$ is low occurs when a byte is loaded into the hold register and the shift register is empty, in which case the byte is immediately transferred to the shift register. The longest time for RDY/$\overline{BUSY}$ to remain low occurs when a byte is loaded into the holding register and the shift register has just started shifting configuration data into configuration RAM.

The RDY/$\overline{BUSY}$ status is also available on the D7 pin by enabling the chip selects, setting $\overline{WR}$ high, and applying $\overline{RD}$ low, where the $\overline{RD}$ input provides an output enable for the D7 pin when $\overline{RD}$ is low. The D[6:0] pins are not enabled to drive when $\overline{RD}$ is low and, therefore, only act as input pins in asynchronous peripheral mode. Optionally, the user can ignore the RDY/$\overline{BUSY}$ status and simply wait until the maximum time it would take for the RDY/$\overline{BUSY}$ line to go high, indicating the FPGA is ready for more data, before writing the next data byte.

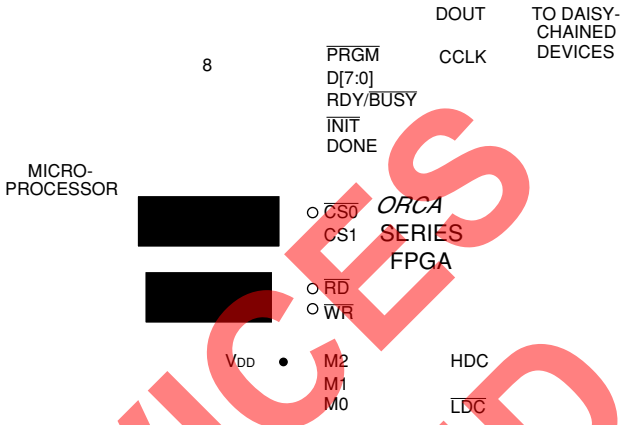


Figure 56. Asynchronous Peripheral Configuration

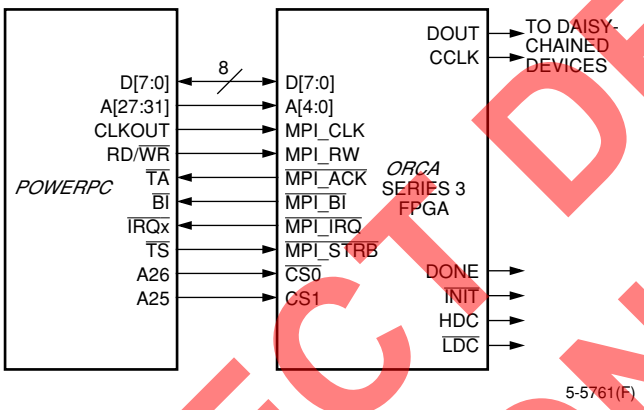
Microprocessor Interface (MPI) Mode

The built-in MPI in Series 3 FPGAs is designed for use in configuring the FPGA. Figure 57 and Figure 58 show the glueless interface for FPGA configuration and readback from the *PowerPC* and *i960* processors, respectively. When enabled by the mode pins, the MPI handles all configuration/readback control and handshaking with the host processor. For single FPGA configuration, the host sets the configuration control register $\overline{PRGM}$ bit to zero then back to a one and, after reading that the $\overline{INIT}$ signal is high in the MPI status register, transfers data 8 bits at a time to the FPGA's D[7:0] input pins.

If configuring multiple FPGAs through daisy-chain operation is desired, the MP_DAISY bit must be set in the configuration control register of the MPI. Because of the latency involved in a daisy-chain configuration, the MP_HOLD_BUS bit may be set to zero rather than one for daisy-chain operation. This allows the MPI to acknowledge the data transfer before the configuration information has been serialized and transferred on the FPGA daisy-chain. The early acknowledgment frees the host processor to perform other system tasks. Configuring with the MP_HOLD_BUS bit at zero requires that the host microprocessor poll the RDY/$\overline{BUSY}$ bit of the MPI status register and/or use the MPI interrupt capability to confirm the readiness of the MPI for more configuration data.

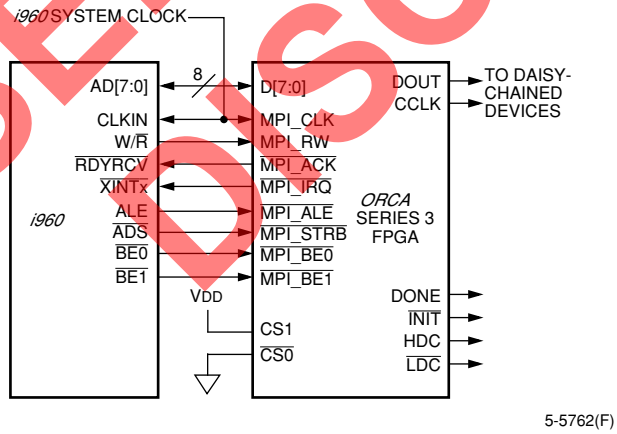
FPGA Configuration Modes (continued)

There are two options for using the host interrupt request in configuration mode. The configuration control register offers control bits to enable the interrupt on either a bit stream error or to notify the host processor when the FPGA is ready for more configuration data. The MPI status register may be used in conjunction with, or in place of, the interrupt request options. The status register contains a 2-bit field to indicate the bit stream error status. As previously mentioned, there is also a bit to indicate the MPI's readiness to receive another byte of configuration data. A flow chart of the MPI configuration process is shown in Figure 59. The MPI status and configuration register bit maps can be found in the Special Function Blocks section and MPI configuration timing information is available in the Timing Characteristics section of this data sheet.



Note: FPGA shown as a memory-mapped peripheral using CS0 and CS1. Other decoding schemes are possible using CS0 and/or CS1.

Figure 57. *PowerPC*/MPI Configuration Schematic



Note: FPGA shown as only system peripheral with fixed chip select signals. For multiperipheral systems, address decoding and/or latching can be used to implement chip selects.

Figure 58. *i960*/MPI Configuration Schematic

Configuration readback can also be performed via the MPI when it is in user mode. The MPI is enabled in user mode by setting the MP_USER bit to 1 in the configuration control register prior to the start of configuration or through a configuration option. To perform readback, the host processor writes the 14-bit readback start address to the readback address registers and sets the RD_CFG bit to 0 in the configuration control register. Readback data is returned 8 bits at a time to the readback data register and is valid when the DATA_RDY bit of the status register is 1. There is no error checking during readback. A flow chart of the MPI readback operation is shown in Figure 60. The RD_DATA pin used for dedicated FPGA readback is invalid during MPI readback.

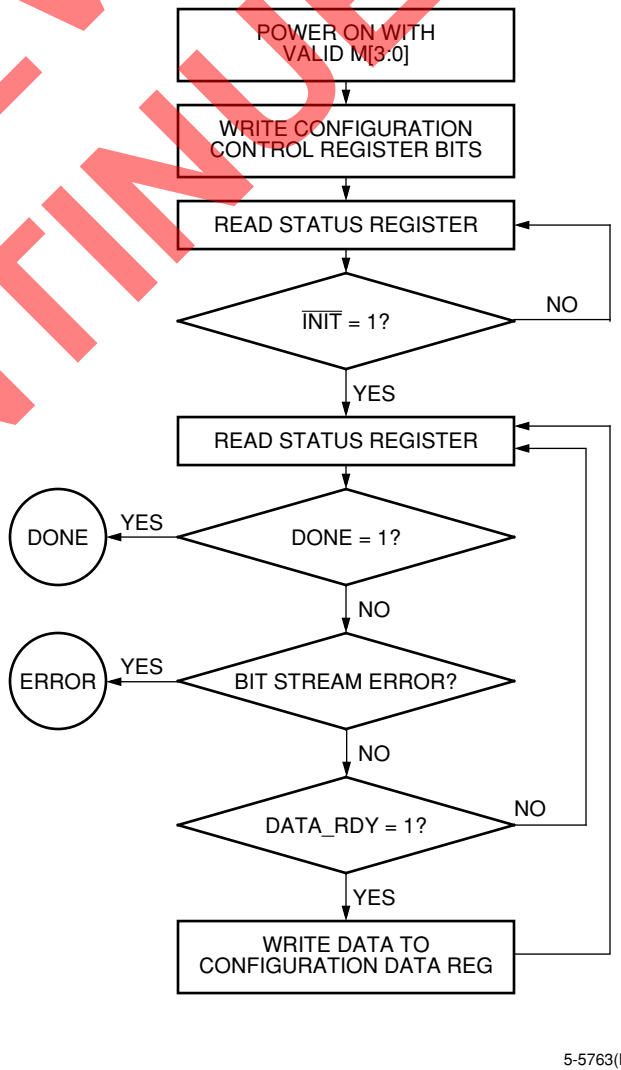


Figure 59. Configuration Through MPI

Timing Characteristics (continued)

Special Function Blocks Timing

Table 49. Microprocessor Interface (MPI) Timing Characteristics

OR3Cxx Commercial: V_{DD} = 5.0 V ± 5%, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C < T_A < +85 °C.
OR3Txxx Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C < T_A < +85 °C.

Parameter	Symbol	Speed								Unit
		–4		–5		–6		–7		
		Min	Max	Min	Max	Min	Max	Min	Max	
***PowerPC* Interface Timing** (T_J = 85 °C, V_{DD} = min)										
Transfer Acknowledge Delay (CLK to $\overline{TA}$)	TA_DEL	—	11.6	—	9.3	—	8.0	—	6.8	ns
Burst Inhibit Delay (CLK to BIN)	BI_DEL	—	11.6	—	9.3	—	8.0	—	6.8	ns
Transfer Acknowledge Delay to High Impedance	TA_DELZ	—	(2)	—	(2)	—	(2)	—	(2)	ns
Burst Inhibit Delay to High Impedance	BI_DELZ	—	(2)	—	(2)	—	(2)	—	(2)	ns
Write Data Setup Time (data to $\overline{TS}$)	WD_SET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Write Data Hold Time (data from CLK while $\overline{MPI\_ACK}$ low)	WD_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Address Setup Time (addr to $\overline{TS}$)	A_SET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Address Hold Time (addr from CLK while $\overline{MPI\_ACK}$ low)	A_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Read/Write Setup Time (R/W to $\overline{TS}$)	RW_SET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Read/Write Hold Time (R/W from CLK while $\overline{MPI\_ACK}$ low)	RW_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Chip Select Setup Time ($\overline{CS0}$, CS1 to $\overline{TS}$)	CS_SET	0.3	—	.25	—	.14	—	.12	—	ns
Chip Select Hold Time ($\overline{CS0}$, CS1 from CLK)	CS_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
User Address Delay (pad to UA[3:0])	UA_DEL	—	3.3	—	2.6	—	2.3	—	1.9	ns
User Read/Write Delay (pad to URDWR_DEL)	URDWR_DEL	—	7.0	—	5.4	—	4.2	—	3.6	ns
***i960* Interface Timing** (T_J = 85 °C, V_{DD} = min)										
Addr/Data Select to ALE ($\overline{ADS}$, to ALE low)	ADSN_SET	2.0	—	1.8	—	1.6	—	1.4	—	ns
Addr/Data Select to ALE ($\overline{ADS}$, from ALE low)	ADSN_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Ready/Receive Delay (CLK to $\overline{RDYRCV}$)	RDYRCV_DEL	—	11.6	—	9.3	—	8.0	—	6.8	ns
Ready/Receive Delay to High Impedance	RDYRCV_DELZ	—	(2)	—	(2)	—	(2)	—	(2)	ns
Write Data Setup Time	WD_SET	(3)	—	(3)	—	(3)	—	(3)	—	ns
Write Data Hold Time	WD_HLD	(4)	—	(4)	—	(4)	—	(4)	—	ns
Address Setup Time (addr to ALE low)	A_SET	2.0	—	1.8	—	0.50	—	—	0.42	ns
Address Hold Time (addr from ALE low)	A_HLD	2.0	—	1.8	—	0.51	—	—	0.44	ns
Byte Enable Setup Time ($\overline{BE0}$, $\overline{BE1}$ to ALE low)	BE_SET	2.0	—	1.8	—	0.50	—	—	0.42	ns
Byte Enable Hold Time ($\overline{BE0}$, $\overline{BE1}$ from ALE low)	BE_HLD	2.0	—	1.8	—	0.51	—	—	0.44	ns
Read/Write Setup Time	RW_SET	(3)	—	(3)	—	(3)	—	(3)	—	ns
Read/Write Hold Time	RW_HLD	(4)	—	(4)	—	(4)	—	(4)	—	ns
Chip Select Setup Time ($\overline{CS0}$, CS1 to CLK)[1]	CS_SET	2.0	—	1.8	—	0.45	—	—	0.38	ns
Chip Select Hold Time ($\overline{CS0}$, CS1 from CLK)[1]	CS_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
User Address Delay (CLK low to UA[3:0])	UA_DEL	—	6.6	—	4.3	—	4.1	—	3.5	ns
User Read/Write Delay (pad to URDWR_DEL)	URDWR_DEL	—	7.0	—	5.4	—	4.2	—	3.6	ns

1. For user system flexibility, $\overline{CS0}$ and CS1 may be set up to any one of the three rising clock edges, beginning with the rising clock edge when $\overline{MPI\_STRB}$ is low. If both chip selects are valid and the setup time is met, the MPI will latch the chip select state, and $\overline{CS0}$ and CS1 may go inactive before the end of the read/write cycle.
2. 0.5 MPI_CLK.
3. Write data and W/$\overline{R}$ have to be valid starting from the clock cycle after both $\overline{ADS}$ and $\overline{CS0}$ and CS1 are recognized.
4. Write data and W/$\overline{R}$ have to be held until the microprocessor receives a valid $\overline{RDYRCV}$.

Notes:
Read and write descriptions are referenced to the host microprocessor; e.g., a read is a read by the host (*PowerPC*, *i960*) from the FPGA.

PowerPC and *i960* timings to/from the clock are relative to the clock at the FPGA microprocessor interface clock pin (MPI_CLK).

Timing Characteristics (continued)

Table 53. General-Purpose Clock Timing Characteristics (Internally Generated Clock)

OR3Cxx Commercial: V_{DD} = 5.0 V ± 5%, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C < T_A < +85 °C.
OR3Txxx Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C < T_A < 70 °C; Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C < T_A < +85 °C.

Device (T_J = 85 °C, V_{DD} = min)	Symbol	Speed								Unit
		-4		-5		-6		-7		
		Min	Max	Min	Max	Min	Max	Min	Max	
OR3T20	CLK_DEL	—	—	—	4.22	—	3.46	—	2.84	ns
OR3T30	CLK_DEL	—	—	—	4.29	—	3.48	—	2.87	ns
OR3T55	CLK_DEL	—	5.34	—	4.41	—	3.53	—	2.93	ns
OR3C/T80	CLK_DEL	—	5.49	—	4.52	—	3.57	—	2.98	ns
OR3T125	CLK_DEL	—	—	—	4.80	—	3.71	—	3.13	ns

Notes:
This table represents the delay for an internally generated clock from the clock tree input in one of the four middle PICs (using pSW routing) on any side of the device which is then distributed to the PFU/PIO clock inputs. If the clock tree input used is located at any other PIC, see the results reported by ispLEVER.

This clock delay is for a fully routed clock tree that uses the general clock network. The delay will be reduced if any of the clock branches are not used. See pin-to-pin timing in Table 56 for clock delays of clocks input on general I/O pins.

Estimating Power Dissipation (continued)

OR3Txxx

The total operating power dissipated is estimated by summing the standby (I_{DDSB}), internal, and external power dissipated. The internal and external power is the power consumed in the PLCs and PICs, respectively. In general, the standby power is small and may be neglected. The total operating power is as follows:

$$P_T = \Sigma P_{PLC} + \Sigma P_{PIC}$$

The internal operating power is made up of two parts: clock generation and PFU output power. The PFU output power can be estimated based upon the number of PFU outputs switching when driving an average fan-out of two:

$$P_{PFU} = 0.068 \text{ mW/MHz}$$

For each PFU output that switches, 0.068 mW/MHz needs to be multiplied times the frequency (in MHz) that the output switches. Generally, this can be estimated by using one-half the clock rate, multiplied by some activity factor; for example, 20%.

The power dissipated by the clock generation circuitry is based upon four parts: the fixed clock power, the power/clock branch row or column, the clock power dissipated in each PFU that uses this particular clock, and the power from the subset of those PFUs configured as synchronous memory. Therefore, the clock power can be calculated for the four parts using the following equations.

OR3T20 Clock Power

P = [0.38 mW/MHz
+ (0.045 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T20 clock power ≈ 2.92 mW/MHz.

OR3T30 Clock Power

P = [0.53 mW/MHz
+ (0.061 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T30 clock power ≈ 3.98 mW/MHz.

OR3T55 Clock Power

P = [0.88 mW/MHz
+ (0.102 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T55 clock power ≈ 6.58 mW/MHz.

OR3T80 Clock Power

P = [0.107 mW/MHz
+ (0.124 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T80 clock power ≈ 9.47 mW/MHz.

OR3T125 Clock Power

P = [0.167 mW/MHz
+ (0.193 mW/MHz/Branch) (# Branches)
+ (0.015 mW/MHz/PFU) (# PFUs)
+ (0.004 mW/MHz/PIO (# PIOs)]

For a quick estimate, the worst-case (typical circuit) OR3T125 clock power ≈ 15.44 mW/MHz.

The power dissipated in a PIC is the sum of the power dissipated in the four PIOs in the PIC. This consists of power dissipated by inputs and ac power dissipated by outputs. The power dissipated in each PIO depends on whether it is configured as an input, output, or input/output. If a PIO is operating as an output, then there is a power dissipation component for P_{IN}, as well as P_{OUT}. This is because the output feeds back to the input.

The power dissipated by an input buffer ($V_{IH} = V_{DD} –$ 0.3 V or higher) is estimated as:

$$P_{IN} = 0.09 \text{ mW/MHz}$$

The ac power dissipation from an output or bidirectional is estimated by the following:

$$P_{OUT} = (C_L + 8.8 \text{ pF}) \times V_{DD}^2 \times F \text{ Watts}$$

where the unit for C_L is farads, and the unit for F is Hz.

Table 74. OR3T55, OR3C/T80, and OR3T125 352-Pin PBGA Pinout

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
B1	PL1D	PL1D	PL1D	I/O
C2	PL1C	PL1C	PL1C	I/O
C1	PL1B	PL1B	PL1B	I/O
D2	PL1A	PL1A	PL1A	I/O
D3	PL2D	PL2D	PL2D	I/O-A0/$\overline{\text{MPI\_BE0}}$
D1	PL2C	PL2A	PL2A	I/O
E2	PL2B	PL3D	PL3D	I/O
E4	—	PL3B	PL3B	I/O
E3	PL2A	PL3A	PL3A	I/O
E1	PL3D	PL4D	PL4D	I/O
F2	PL3C	PL4C	PL4C	I/O
G4	PL3B	PL4B	PL4B	I/O
F3	PL3A	PL4A	PL5D	I/O
F1	PL4D	PL5D	PL6D	I/O
G2	PL4C	PL5C	PL6C	I/O
G1	PL4B	PL5B	PL6B	I/O
G3	PL4A	PL5A	PL7D	I/O-A1/$\overline{\text{MPI\_BE1}}$
H2	PL5D	PL6D	PL8D	I/O
J4	PL5C	PL6C	PL8C	I/O
H1	PL5B	PL6B	PL8B	I/O
H3	PL5A	PL6A	PL8A	I/O-A2
J2	PL6D	PL7D	PL9D	I/O
J1	PL6C	PL7C	PL9C	I/O
K2	PL6B	PL7B	PL9B	I/O
J3	PL6A	PL7A	PL9A	I/O-A3
K1	PL7D	PL8D	PL10D	I/O
K4	PL7C	PL8A	PL10A	I/O
L2	PL7B	PL9D	PL11D	I/O
K3	PL7A	PL9B	PL11A	I/O-A4
L1	PL8D	PL9A	PL12D	I/O-A5
M2	PL8C	PL10C	PL12A	I/O
M1	PL8B	PL10B	PL13D	I/O
L3	PL8A	PL10A	PL13A	I/O-A6
N2	PECKL	PECKL	PECKL	I-ECKL
M4	PL9C	PL11C	PL14C	I/O
N1	PL9B	PL11B	PL14B	I/O
M3	PL9A	PL11A	PL14A	I/O-A7/MPI_CLK
P2	PL10D	PL12D	PL15D	I/O
P4	PL10C	PL12C	PL15C	I/O
P1	PL10B	PL12B	PL15B	I/O
N3	PL10A	PL12A	PL15A	I/O-A8/MPI_RW
R2	PL11D	PL13D	PL16D	I/O-A9/$\overline{\text{MPI\_ACK}}$
P3	PL11C	PL13B	PL16A	I/O
R1	PL11B	PL13A	PL17D	I/O
T2	PL11A	PL14C	PL17A	I/O-A10/$\overline{\text{MPI\_BI}}$

Pin	OR3T55 Pad	OR3C/T80 Pad	OR3T125 Pad	Function
R26	PR11B	PR13A	PR17D	I/O
T24	PR11C	PR13B	PR16A	I/O
P25	PR11D	PR13D	PR16D	I/O
R23	PR10A	PR12A	PR15A	I/O
P26	PR10B	PR12B	PR15B	I/O
R24	PR10C	PR12C	PR15C	I/O
N25	PR10D	PR12D	PR15D	I/O
N23	PECKR	PECKR	PECKR	I-ECKR
N26	PR9B	PR11B	PR14B	I/O
P24	PR9C	PR11C	PR14C	I/O
M25	PR9D	PR11D	PR14D	I/O
N24	PR8A	PR10A	PR13A	I/O
M26	PR8B	PR10C	PR13D	I/O
L25	PR8C	PR10D	PR12A	I/O
M24	PR8D	PR9B	PR12D	I/O
L26	PR7A	PR9C	PR11A	I/O-CS1
M23	PR7B	PR9D	PR11D	I/O
K25	PR7C	PR8A	PR10A	I/O
L24	PR7D	PR8D	PR10D	I/O
K26	PR6A	PR7A	PR9A	I/O-$\overline{CS0}$
K23	PR6B	PR7B	PR9B	I/O
J25	PR6C	PR7C	PR9C	I/O
K24	PR6D	PR7D	PR9D	I/O
J26	PR5A	PR6A	PR8A	I/O
H25	PR5B	PR6B	PR8B	I/O
H26	PR5C	PR6C	PR8C	I/O
J24	PR5D	PR6D	PR8D	I/O
G25	PR4A	PR5A	PR7A	I/O-$\overline{RD}$/MPI_$\overline{STRB}$
H23	PR4B	PR5B	PR6A	I/O
G26	PR4C	PR5C	PR6C	I/O
H24	PR4D	PR5D	PR5A	I/O
F25	PR3A	PR4A	PR4A	I/O
G23	PR3B	PR4B	PR4B	I/O
F26	PR3C	PR4C	PR4C	I/O
G24	PR3D	PR4D	PR4D	I/O
E25	PR2A	PR3A	PR3A	I/O-$\overline{WR}$
E26	PR2B	PR3B	PR3B	I/O
F24	—	PR3D	PR3D	I/O
D25	PR2C	PR2A	PR2A	I/O
E23	PR2D	PR2D	PR2D	I/O
D26	PR1A	PR1A	PR1A	I/O
E24	PR1B	PR1B	PR1B	I/O
C25	PR1C	PR1C	PR1C	I/O
D24	PR1D	PR1D	PR1D	I/O
C26	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{PRD\_CFGN}$	$\overline{RD\_CFG}$
A25	PT18D	PT22D	PT28D	I/O-SECKUR

Pin	OR3C/T80 Pad	OR3T125 Pad	Function
B19	PT9B	PT11D	I/O
D18	PT9C	PT12A	I/O-D0/DIN
A19	PT9D	PT12C	I/O
C18	PT10A	PT12D	I/O
B18	PT10B	PT13A	I/O
A18	PT10C	PT13C	I/O
C17	PT10D	PT13D	I/O-D1
B17	PT11A	PT14A	I/O-D2
A17	PT11B	PT14B	I/O
B16	PT11C	PT14C	I/O
D16	PT11D	PT14D	I/O
C16	PT12A	PT15A	I/O-D3
A15	PT12B	PT15B	I/O
B15	PT12C	PT15C	I/O
C15	PECKT	PECKT	I-ECKT
A14	PT13A	PT16A	I/O-D4
B14	PT13B	PT16B	I/O
C14	PT13C	PT16D	I/O
A13	PT13D	PT17A	I/O
D14	PT14A	PT17B	I/O
B13	PT14B	PT17D	I/O
C13	PT14C	PT18A	I/O-D5
B12	PT14D	PT18B	I/O
D13	PT15A	PT18D	I/O
C12	PT15B	PT19A	I/O
A11	PT15D	PT19D	I/O
B11	PT16A	PT20A	I/O
D12	PT16B	PT20D	I/O-D6
C11	PT16C	PT21A	I/O
A10	PT16D	PT21D	I/O
B10	PT17A	PT22D	I/O
C10	PT17B	PT23B	I/O
A9	PT17C	PT23C	I/O
B9	PT17D	PT23D	I/O
D10	PT18A	PT24A	I/O
C9	PT18B	PT24B	I/O
B8	PT18C	PT24C	I/O
C8	PT18D	PT24D	I/O-D7
D9	PT19A	PT25A	I/O
A7	PT19B	PT25B	I/O
B7	PT19C	PT25C	I/O
C7	PT19D	PT25D	I/O
D8	PT20A	PT26A	I/O
A6	PT20B	PT26B	I/O
B6	PT20C	PT26C	I/O
C6	PT20D	PT26D	I/O
A5	PT21A	PT27A	I/O-RDY/RCLK/MPI_ALE
B5	PT21B	PT27B	I/O
C5	PT21C	PT27C	I/O
D6	PT21D	PT27D	I/O

Pin	OR3C/T80 Pad	OR3T125 Pad	Function
A4	PT22A	PT28A	I/O
B4	PT22B	PT28B	I/O
C4	PT22C	PT28C	I/O
D5	PT22D	PT28D	I/O-SECKUR
A12	VSS	VSS	VSS
A16	VSS	VSS	VSS
A2	VSS	VSS	VSS
A20	VSS	VSS	VSS
A24	VSS	VSS	VSS
A29	VSS	VSS	VSS
A3	VSS	VSS	VSS
A30	VSS	VSS	VSS
A8	VSS	VSS	VSS
AD1	VSS	VSS	VSS
AD31	VSS	VSS	VSS
AJ1	VSS	VSS	VSS
AJ2	VSS	VSS	VSS
AJ30	VSS	VSS	VSS
AJ31	VSS	VSS	VSS
AK1	VSS	VSS	VSS
AK29	VSS	VSS	VSS
AK3	VSS	VSS	VSS
AK31	VSS	VSS	VSS
AL12	VSS	VSS	VSS
AL16	VSS	VSS	VSS
AL2	VSS	VSS	VSS
AL20	VSS	VSS	VSS
AL24	VSS	VSS	VSS
AL29	VSS	VSS	VSS
AL3	VSS	VSS	VSS
AL30	VSS	VSS	VSS
AL8	VSS	VSS	VSS
B1	VSS	VSS	VSS
B29	VSS	VSS	VSS
B3	VSS	VSS	VSS
B31	VSS	VSS	VSS
C1	VSS	VSS	VSS
C2	VSS	VSS	VSS
C30	VSS	VSS	VSS
C31	VSS	VSS	VSS
H1	VSS	VSS	VSS
H31	VSS	VSS	VSS
M1	VSS	VSS	VSS
M31	VSS	VSS	VSS
T1	VSS	VSS	VSS
T31	VSS	VSS	VSS
Y1	VSS	VSS	VSS
Y31	VSS	VSS	VSS
A1	VDD	VDD	VDD
A31	VDD	VDD	VDD

Pin	OR3C/T80 Pad	OR3T125 Pad	Function
AA28	VDD	VDD	VDD
AA4	VDD	VDD	VDD
AE28	VDD	VDD	VDD
AE4	VDD	VDD	VDD
AH11	VDD	VDD	VDD
AH15	VDD	VDD	VDD
AH17	VDD	VDD	VDD
AH21	VDD	VDD	VDD
AH25	VDD	VDD	VDD
AH28	VDD	VDD	VDD
AH4	VDD	VDD	VDD
AH7	VDD	VDD	VDD
AJ29	VDD	VDD	VDD
AJ3	VDD	VDD	VDD
AK2	VDD	VDD	VDD
AK30	VDD	VDD	VDD
AL1	VDD	VDD	VDD
AL31	VDD	VDD	VDD
B2	VDD	VDD	VDD
B30	VDD	VDD	VDD
C29	VDD	VDD	VDD
C3	VDD	VDD	VDD
D11	VDD	VDD	VDD
D15	VDD	VDD	VDD
D17	VDD	VDD	VDD
D21	VDD	VDD	VDD
D25	VDD	VDD	VDD
D28	VDD	VDD	VDD
D4	VDD	VDD	VDD
D7	VDD	VDD	VDD
G28	VDD	VDD	VDD
G4	VDD	VDD	VDD
L28	VDD	VDD	VDD
L4	VDD	VDD	VDD
R28	VDD	VDD	VDD
R4	VDD	VDD	VDD
U28	VDD	VDD	VDD
U4	VDD	VDD	VDD

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:

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

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 Θ_{JA} measurements described above, besides the other parameters measured, an additional temperature reading, T_C, is made with a thermocouple attached at top-dead-center of the case. ψ_{JC} is also expressed in units of °C/watt.

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:

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

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 so as 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 Θ_{JC} from ψ_{JC}. Θ_{JC} is a true thermal resistance and is expressed in units of °C/watt.

ΘJB

This is the thermal resistance from junction to board (a.k.a. Θ_{JL}). It is defined by:

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

where T_B is the temperature of the board adjacent to a lead measured with a thermocouple. The other parameters on the right-hand side have been defined above. This is considered a true thermal resistance, and the measurement is made with a water-cooled heat sink pressed against the board so as to draw most of the heat out of the leads. Note that Θ_{JB} is expressed in units of °C/watt, and that this parameter and the way it is measured is still in JEDEC committee.

Commercial

Device Family	Part Number	Speed Grade	Package Type	Pin/Ball Count	Grade	Packing Designator
OR3T55	OR3T557PS208-DB[1]	7	SQFP2	208	C	DB
	OR3T557S208-DB	7	SQFP	208	C	DB
	OR3T557PS240-DB[3]	7	SQFP2	240	C	DB
	OR3T557BA256-DB	7	PBGA	256	C	DB
	OR3T557BA352-DB	7	PBGA	352	C	DB
	OR3T556PS208-DB[1]	6	SQFP2	208	C	DB
	OR3T556S208-DB	6	SQFP	208	C	DB
	OR3T556PS240-DB[3]	6	SQFP2	240	C	DB
	OR3T556BA256-DB	6	PBGA	256	C	DB
	OR3T556BA352-DB	6	PBGA	352	C	DB
OR3T80	OR3T807PS208-DB[1]	7	SQFP2	208	C	DB
	OR3T807S208-DB	7	SQFP	208	C	DB
	OR3T807PS240-DB[3]	7	SQFP2	240	C	DB
	OR3T807BA352-DB	7	PBGA	352	C	DB
	OR3T807BC432-DB	7	EBGA	432	C	DB
	OR3T806PS208-DB[1]	6	SQFP2	208	C	DB
	OR3T806S208-DB	6	SQFP	208	C	DB
	OR3T806PS240-DB[3]	6	SQFP2	240	C	DB
	OR3T806BA352-DB	6	PBGA	352	C	DB
	OR3T806BC432-DB	6	EBGA	432	C	DB
OR3T125	OR3T1257PS208-DB[3]	7	SQFP2	208	C	DB
	OR3T1257PS240-DB[3]	7	SQFP2	240	C	DB
	OR3T1257BA352-DB	7	PBGA	352	C	DB
	OR3T1257BC432-DB	7	EBGA	432	C	DB
	OR3T1256PS208-DB[3]	6	SQFP2	208	C	DB
	OR3T1256PS240-DB[3]	6	SQFP2	240	C	DB
	OR3T1256BA352-DB	6	PBGA	352	C	DB
	OR3T1256BC432-DB	6	EBGA	432	C	DB