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 | 216 |
| Number of Logic Elements/Cells | 972 |
| Total RAM Bits | 24576 |
| Number of I/O | 140 |
| Number of Gates | 30000 |
| Voltage - Supply | 2.375V ~ 2.625V |
| Mounting Type | Surface Mount |
| Operating Temperature | 0°C ~ 85°C (TJ) |
| Package / Case | 208-BFQFP |
| Supplier Device Package | 208-PQFP (28x28) |
| Purchase URL | https://www.e-xfl.com/product-detail/xilinx/xc2s30-6pq208c |

## Spartan-II Product Availability

Table 2 shows the maximum user I/Os available on the device and the number of user I/Os available for each device/package combination. The four global clock pins are usable as additional user I/Os when not used as a global clock pin. These pins are not included in user I/O counts.

*Table 2:* **Spartan-II FPGA User I/O Chart(1)**

| Device | Maximum User I/O | Available User I/O According to Package Type | | | | | |
|---|---|---|---|---|---|---|---|
| | | VQ100 VQG100 | TQ144 TQG144 | CS144 CSG144 | PQ208 PQG208 | FG256 FGG256 | FG456 FGG456 |
| XC2S15 | 86 | 60 | 86 | (Note 2) | - | - | - |
| XC2S30 | 92 | 60 | 92 | 92 | (Note 2) | - | - |
| XC2S50 | 176 | - | 92 | - | 140 | 176 | - |
| XC2S100 | 176 | - | 92 | - | 140 | 176 | (Note 2) |
| XC2S150 | 260 | - | - | - | 140 | 176 | 260 |
| XC2S200 | 284 | - | - | - | 140 | 176 | 284 |

**Notes:**
1. All user I/O counts do not include the four global clock/user input pins.
2. Discontinued by PDN2004-01.

Similarly, the F6 multiplexer combines the outputs of all four function generators in the CLB by selecting one of the F5-multiplexer outputs. This permits the implementation of any 6-input function, an 8:1 multiplexer, or selected functions of up to 19 inputs.

Each CLB has four direct feedthrough paths, one per LC. These paths provide extra data input lines or additional local routing that does not consume logic resources.

### Arithmetic Logic

Dedicated carry logic provides capability for high-speed arithmetic functions. The Spartan-II FPGA CLB supports two separate carry chains, one per slice. The height of the carry chains is two bits per CLB.

The arithmetic logic includes an XOR gate that allows a 1-bit full adder to be implemented within an LC. In addition, a dedicated AND gate improves the efficiency of multiplier implementation.

The dedicated carry path can also be used to cascade function generators for implementing wide logic functions.

### BUFTs

Each Spartan-II FPGA CLB contains two 3-state drivers (BUFTs) that can drive on-chip busses. See "Dedicated Routing," page 12. Each Spartan-II FPGA BUFT has an independent 3-state control pin and an independent input pin.

## Block RAM

Spartan-II FPGAs incorporate several large block RAM memories. These complement the distributed RAM Look-Up Tables (LUTs) that provide shallow memory structures implemented in CLBs.

Block RAM memory blocks are organized in columns. All Spartan-II devices contain two such columns, one along each vertical edge. These columns extend the full height of the chip. Each memory block is four CLBs high, and consequently, a Spartan-II device eight CLBs high will contain two memory blocks per column, and a total of four blocks.

*Table 5:* **Spartan-II Block RAM Amounts**

| Spartan-II Device | # of Blocks | Total Block RAM Bits |
|---|---|---|
| XC2S15 | 4 | 16K |
| XC2S30 | 6 | 24K |
| XC2S50 | 8 | 32K |
| XC2S100 | 10 | 40K |
| XC2S150 | 12 | 48K |
| XC2S200 | 14 | 56K |

Each block RAM cell, as illustrated in Figure 5, is a fully synchronous dual-ported 4096-bit RAM with independent control signals for each port. The data widths of the two ports can be configured independently, providing built-in bus-width conversion.

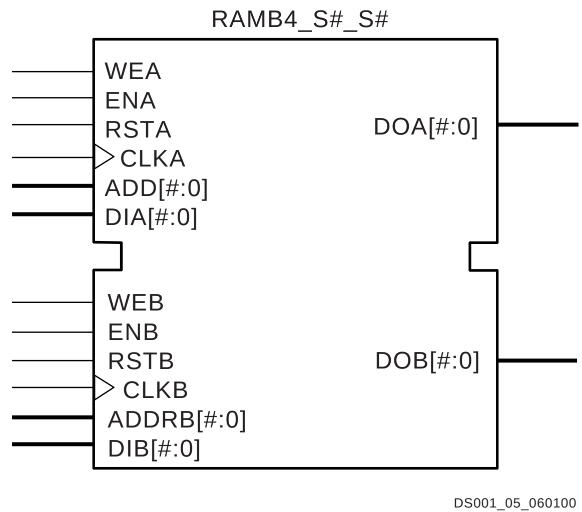



*Figure 5:* **Dual-Port Block RAM**

Table 6 shows the depth and width aspect ratios for the block RAM.

*Table 6:* **Block RAM Port Aspect Ratios**

| Width | Depth | ADDR Bus | Data Bus |
|---|---|---|---|
| 1 | 4096 | ADDR<11:0> | DATA<0> |
| 2 | 2048 | ADDR<10:0> | DATA<1:0> |
| 4 | 1024 | ADDR<9:0> | DATA<3:0> |
| 8 | 512 | ADDR<8:0> | DATA<7:0> |
| 16 | 256 | ADDR<7:0> | DATA<15:0> |

The Spartan-II FPGA block RAM also includes dedicated routing to provide an efficient interface with both CLBs and other block RAMs.

## Programmable Routing Matrix

It is the longest delay path that limits the speed of any worst-case design. Consequently, the Spartan-II routing architecture and its place-and-route software were defined in a single optimization process. This joint optimization minimizes long-path delays, and consequently, yields the best system performance.

The joint optimization also reduces design compilation times because the architecture is software-friendly. Design cycles are correspondingly reduced due to shorter design iteration times.

# Configuration

Configuration is the process by which the bitstream of a design, as generated by the Xilinx software, is loaded into the internal configuration memory of the FPGA. Spartan-II devices support both serial configuration, using the master/slave serial and JTAG modes, as well as byte-wide configuration employing the Slave Parallel mode.

## Configuration File

Spartan-II devices are configured by sequentially loading frames of data that have been concatenated into a configuration file. Table 8 shows how much nonvolatile storage space is needed for Spartan-II devices.

It is important to note that, while a PROM is commonly used to store configuration data before loading them into the FPGA, it is by no means required. Any of a number of different kinds of under populated nonvolatile storage already available either on or off the board (i.e., hard drives, FLASH cards, etc.) can be used. For more information on configuration without a PROM, refer to XAPP098, *The Low-Cost, Efficient Serial Configuration of Spartan FPGAs*.

*Table 8:* **Spartan-II Configuration File Size**

| Device | Configuration File Size (Bits) |
|---|---|
| XC2S15 | 197,696 |
| XC2S30 | 336,768 |
| XC2S50 | 559,200 |
| XC2S100 | 781,216 |
| XC2S150 | 1,040,096 |
| XC2S200 | 1,335,840 |

## Modes

Spartan-II devices support the following four configuration modes:

- Slave Serial mode
- Master Serial mode
- Slave Parallel mode
- Boundary-scan mode

The Configuration mode pins (M2, M1, M0) select among these configuration modes with the option in each case of having the IOB pins either pulled up or left floating prior to the end of configuration. The selection codes are listed in Table 9.

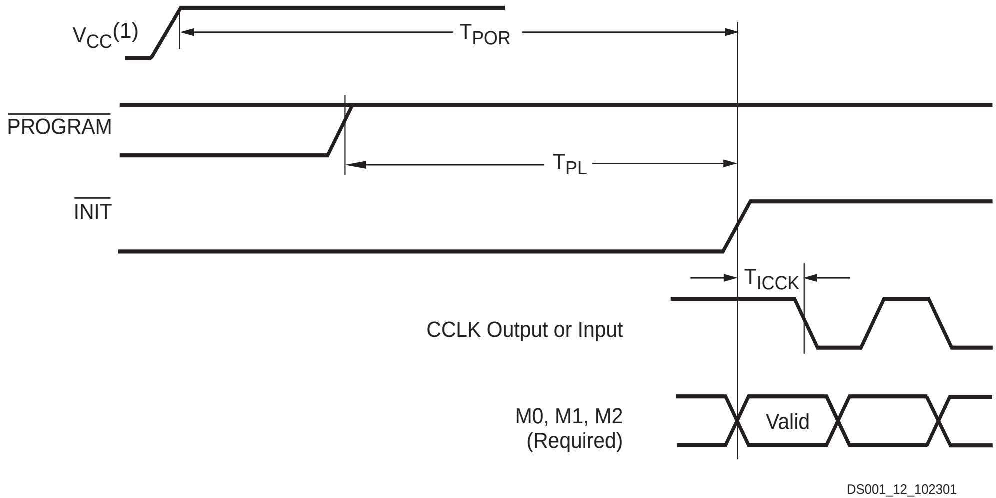
Configuration through the boundary-scan port is always available, independent of the mode selection. Selecting the boundary-scan mode simply turns off the other modes. The three mode pins have internal pull-up resistors, and default to a logic High if left unconnected.

*Table 9:* **Configuration Modes**

| Configuration Mode | Preconfiguration Pull-ups | M0 | M1 | M2 | CCLK Direction | Data Width | Serial $D_{OUT}$ |
|---|---|---|---|---|---|---|---|
| Master Serial mode | No | 0 | 0 | 0 | Out | 1 | Yes |
| | Yes | 0 | 0 | 1 | | | |
| Slave Parallel mode | Yes | 0 | 1 | 0 | In | 8 | No |
| | No | 0 | 1 | 1 | | | |
| Boundary-Scan mode | Yes | 1 | 0 | 0 | N/A | 1 | No |
| | No | 1 | 0 | 1 | | | |
| Slave Serial mode | Yes | 1 | 1 | 0 | In | 1 | Yes |
| | No | 1 | 1 | 1 | | | |

**Notes:**

1. During power-on and throughout configuration, the I/O drivers will be in a high-impedance state. After configuration, all unused I/Os (those not assigned signals) will remain in a high-impedance state. Pins used as outputs may pulse High at the end of configuration (see Answer 10504).
2. If the Mode pins are set for preconfiguration pull-ups, those resistors go into effect once the rising edge of INIT samples the Mode pins. They will stay in effect until GTS is released during startup, after which the UnusedPin bitstream generator option will determine whether the unused I/Os have a pull-up, pull-down, or no resistor.




| Symbol | Description | Min | Max |
|---|---|---|---|
| $T_{POR}$ | Power-on reset | - | 2 ms |
| $T_{PL}$ | Program latency | - | 100 µs |
| $T_{ICCK}$ | CCLK output delay (Master Serial mode only) | 0.5 µs | 4 µs |
| $T_{PROGRAM}$ | Program pulse width | 300 ns | - |

**Notes: (referring to waveform above:)**

1. Before configuration can begin, $V_{CCINT}$ must be greater than 1.6V and $V_{CCO}$ Bank 2 must be greater than 1.0V.

*Figure 12:* **Configuration Timing on Power-Up**

### *Clearing Configuration Memory*

The device indicates that clearing the configuration memory is in progress by driving $\overline{\text{INIT}}$ Low. At this time, the user can delay configuration by holding either $\overline{\text{PROGRAM}}$ or $\overline{\text{INIT}}$ Low, which causes the device to remain in the memory clearing phase. Note that the bidirectional $\overline{\text{INIT}}$ line is driving a Low logic level during memory clearing. To avoid contention, use an open-drain driver to keep $\overline{\text{INIT}}$ Low.

With no delay in force, the device indicates that the memory is completely clear by driving $\overline{\text{INIT}}$ High. The FPGA samples its mode pins on this Low-to-High transition.

### *Loading Configuration Data*

Once $\overline{\text{INIT}}$ is High, the user can begin loading configuration data frames into the device. The details of loading the configuration data are discussed in the sections treating the configuration modes individually. The sequence of operations necessary to load configuration data using the serial modes is shown in Figure 14. Loading data using the Slave Parallel mode is shown in Figure 19, page 25.

#### CRC Error Checking

During the loading of configuration data, a CRC value embedded in the configuration file is checked against a CRC value calculated within the FPGA. If the CRC values do not match, the FPGA drives $\overline{\text{INIT}}$ Low to indicate that a frame error has occurred and configuration is aborted.

To reconfigure the device, the $\overline{\text{PROGRAM}}$ pin should be asserted to reset the configuration logic. Recycling power also resets the FPGA for configuration. See "Clearing Configuration Memory".

### *Start-up*

The start-up sequence oversees the transition of the FPGA from the configuration state to full user operation. A match of CRC values, indicating a successful loading of the configuration data, initiates the sequence.

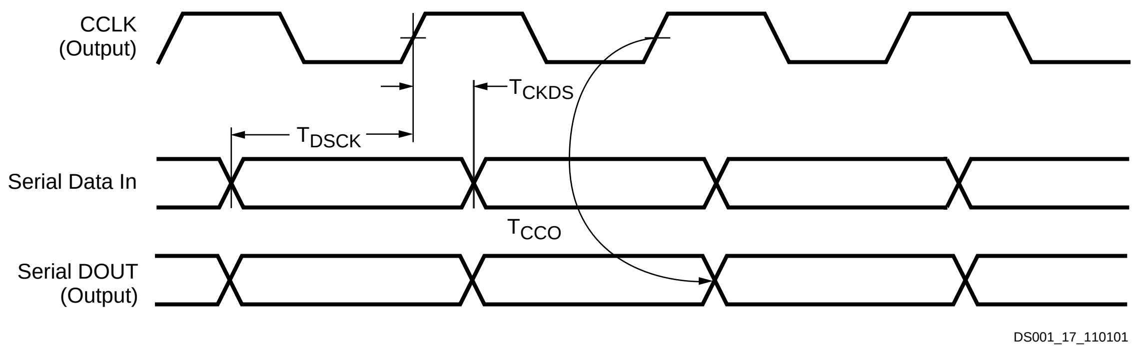
During start-up, the device performs four operations:

1. The assertion of DONE. The failure of DONE to go High may indicate the unsuccessful loading of configuration data.
2. The release of the Global Three State net. This activates I/Os to which signals are assigned. The remaining I/Os stay in a high-impedance state with internal weak pull-down resistors present.
3. Negates Global Set Reset (GSR). This allows all flip-flops to change state.
4. The assertion of Global Write Enable (GWE). This allows all RAMs and flip-flops to change state.

### *Master Serial Mode*

In Master Serial mode, the CCLK output of the FPGA drives a Xilinx PROM which feeds a serial stream of configuration data to the FPGA's DIN input. Figure 15 shows a Master Serial FPGA configuring a Slave Serial FPGA from a PROM. A Spartan-II device in Master Serial mode should be connected as shown for the device on the left side. Master Serial mode is selected by a <00x> on the mode pins (M0, M1, M2). The PROM RESET pin is driven by $\overline{\text{INIT}}$, and CE input is driven by DONE. The interface is identical to the slave serial mode except that an oscillator internal to the FPGA is used to generate the configuration clock (CCLK). Any of a number of different frequencies ranging from 4 to 60 MHz can be set using the ConfigRate option in the Xilinx software. On power-up, while the first 60 bytes of the configuration data are being loaded, the CCLK frequency is always 2.5 MHz. This frequency is used until the ConfigRate bits, part of the configuration file, have been loaded into the FPGA, at which point, the frequency changes to the selected ConfigRate. Unless a different frequency is specified in the design, the default ConfigRate is 4 MHz. The frequency of the CCLK signal created by the internal oscillator has a variance of +45%, –30% from the specified value.

Figure 17 shows the timing for Master Serial configuration. The FPGA accepts one bit of configuration data on each rising CCLK edge. After the FPGA has been loaded, the data for the next device in a daisy-chain is presented on the DOUT pin after the rising CCLK edge.




| Symbol | | Description | | Units |
|---|---|---|---|---|
| $T_{DSCK}$ | CCLK | DIN setup | 5.0 | ns, min |
| $T_{CKDS}$ | | DIN hold | 0.0 | ns, min |
| | | Frequency tolerance with respect to nominal | +45%, –30% | - |

*Figure 17:* **Master Serial Mode Timing**

## Slave Parallel Mode

The Slave Parallel mode is the fastest configuration option. Byte-wide data is written into the FPGA. A BUSY flag is provided for controlling the flow of data at a clock frequency $F_{CCNH}$ above 50 MHz.

Figure 18, page 24 shows the connections for two Spartan-II devices using the Slave Parallel mode. Slave Parallel mode is selected by a <011> on the mode pins (M0, M1, M2).

If a configuration file of the format .bit, .rbt, or non-swapped HEX is used for parallel programming, then the most significant bit (i.e. the left-most bit of each configuration byte, as displayed in a text editor) must be routed to the D0 input on the FPGA.

The agent controlling configuration is not shown. Typically, a processor, a microcontroller, or CPLD controls the Slave Parallel interface. The controlling agent provides byte-wide configuration data, CCLK, a Chip Select ($\overline{\text{CS}}$) signal and a Write signal ($\overline{\text{WRITE}}$). If BUSY is asserted (High) by the FPGA, the data must be held until BUSY goes Low.

After configuration, the pins of the Slave Parallel port (D0-D7) can be used as additional user I/O. Alternatively, the port may be retained to permit high-speed 8-bit readback. Then data can be read by de-asserting $\overline{\text{WRITE}}$. See "Readback," page 25.

If CCLK is slower than $F_{CCNH}$, the FPGA will never assert BUSY. In this case, the above handshake is unnecessary, and data can simply be entered into the FPGA every CCLK cycle.

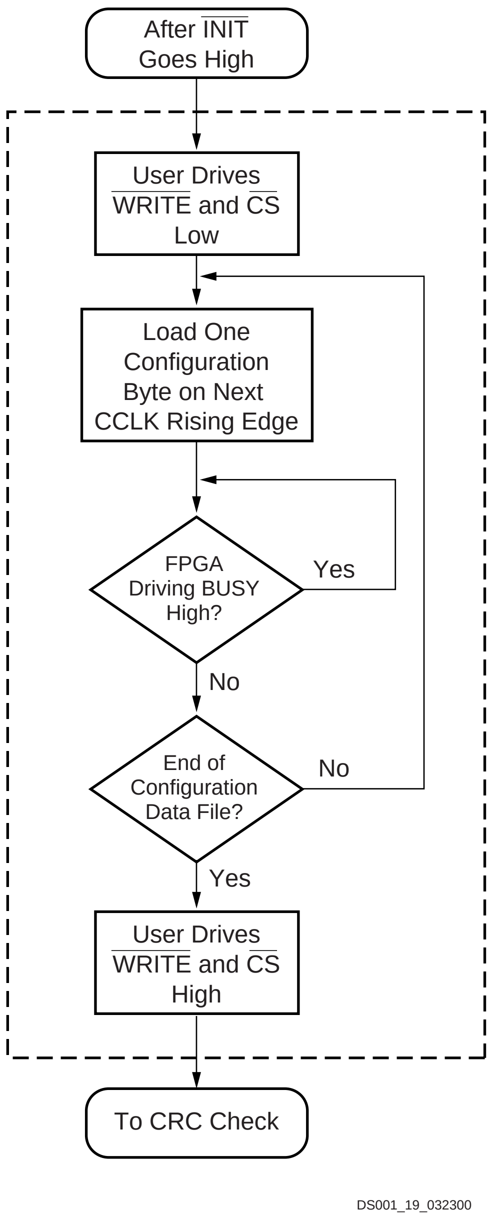



*Figure 19:* **Loading Configuration Data for the Slave Parallel Mode**

A configuration packet does not have to be written in one continuous stretch, rather it can be split into many write sequences. Each sequence would involve assertion of $\overline{\text{CS}}$.

In applications where multiple clock cycles may be required to access the configuration data before each byte can be loaded into the Slave Parallel interface, a new byte of data may not be ready for each consecutive CCLK edge. In such a case the $\overline{\text{CS}}$ signal may be de-asserted until the next byte is valid on D0-D7. While $\overline{\text{CS}}$ is High, the Slave Parallel interface does not expect any data and ignores all CCLK transitions. However, to avoid aborting configuration, $\overline{\text{WRITE}}$ must continue to be asserted while $\overline{\text{CS}}$ is asserted.

### *Abort*

To abort configuration during a write sequence, de-assert $\overline{\text{WRITE}}$ while holding $\overline{\text{CS}}$ Low. The abort operation is initiated at the rising edge of CCLK, as shown in Figure 21, page 26. The device will remain BUSY until the aborted operation is complete. After aborting configuration, data is assumed to be unaligned to word boundaries and the FPGA requires a new synchronization word prior to accepting any new packets.

## Boundary-Scan Mode

In the boundary-scan mode, no nondedicated pins are required, configuration being done entirely through the IEEE 1149.1 Test Access Port.

Configuration through the TAP uses the special CFG_IN instruction. This instruction allows data input on TDI to be converted into data packets for the internal configuration bus.

The following steps are required to configure the FPGA through the boundary-scan port.

1. Load the CFG_IN instruction into the boundary-scan instruction register (IR)
2. Enter the Shift-DR (SDR) state
3. Shift a standard configuration bitstream into TDI
4. Return to Run-Test-Idle (RTI)
5. Load the JSTART instruction into IR
6. Enter the SDR state
7. Clock TCK through the sequence (the length is programmable)
8. Return to RTI

Configuration and readback via the TAP is always available. The boundary-scan mode simply locks out the other modes. The boundary-scan mode is selected by a <10x> on the mode pins (M0, M1, M2).

# Readback

The configuration data stored in the Spartan-II FPGA configuration memory can be readback for verification. Along with the configuration data it is possible to readback the contents of all flip-flops/latches, LUT RAMs, and block RAMs. This capability is used for real-time debugging.

For more detailed information see XAPP176, *Spartan-II FPGA Family Configuration and Readback.*

## Design Considerations

This section contains more detailed design information on the following features:

- Delay-Locked Loop . . . see page 27
- Block RAM . . . see page 32
- Versatile I/O . . . see page 36

## Using Delay-Locked Loops

The Spartan-II FPGA family provides up to four fully digital dedicated on-chip Delay-Locked Loop (DLL) circuits which provide zero propagation delay, low clock skew between output clock signals distributed throughout the device, and advanced clock domain control. These dedicated DLLs can be used to implement several circuits that improve and simplify system level design.

### Introduction

Quality on-chip clock distribution is important. Clock skew and clock delay impact device performance and the task of managing clock skew and clock delay with conventional clock trees becomes more difficult in large devices. The Spartan-II family of devices resolve this potential problem by providing up to four fully digital dedicated on-chip Delay-Locked Loop (DLL) circuits which provide zero propagation delay and low clock skew between output clock signals distributed throughout the device.

Each DLL can drive up to two global clock routing networks within the device. The global clock distribution network minimizes clock skews due to loading differences. By monitoring a sample of the DLL output clock, the DLL can compensate for the delay on the routing network, effectively eliminating the delay from the external input port to the individual clock loads within the device.

In addition to providing zero delay with respect to a user source clock, the DLL can provide multiple phases of the source clock. The DLL can also act as a clock doubler or it can divide the user source clock by up to 16.

Clock multiplication gives the designer a number of design alternatives. For instance, a 50 MHz source clock doubled by the DLL can drive an FPGA design operating at 100 MHz. This technique can simplify board design because the clock path on the board no longer distributes such a high-speed signal. A multiplied clock also provides designers the option of time-domain-multiplexing, using one circuit twice per clock cycle, consuming less area than two copies of the same circuit.

The DLL can also act as a clock mirror. By driving the DLL output off-chip and then back in again, the DLL can be used to de-skew a board level clock between multiple devices.

In order to guarantee the system clock establishes prior to the device "waking up," the DLL can delay the completion of the device configuration process until after the DLL achieves lock.

By taking advantage of the DLL to remove on-chip clock delay, the designer can greatly simplify and improve system level design involving high-fanout, high-performance clocks.

### Library DLL Primitives

Figure 22 shows the simplified Xilinx library DLL macro, BUFGDLL. This macro delivers a quick and efficient way to provide a system clock with zero propagation delay throughout the device. Figure 23 and Figure 24 show the two library DLL primitives. These primitives provide access to the complete set of DLL features when implementing more complex applications.

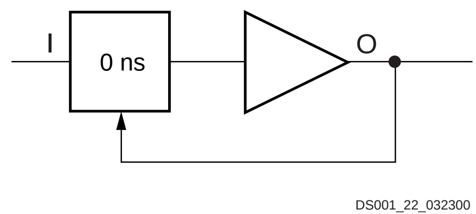



*Figure 22:* **Simplified DLL Macro BUFGDLL**

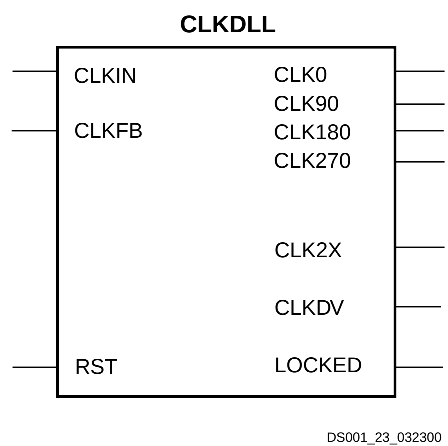



*Figure 23:* **Standard DLL Primitive CLKDLL**

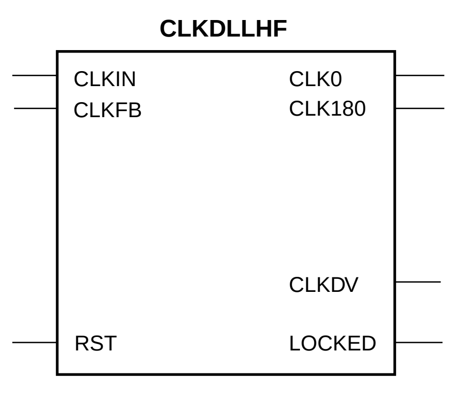



*Figure 24:* **High-Frequency DLL Primitive CLKDLLHF**

### *Startup Delay Property*

This property, STARTUP_WAIT, takes on a value of TRUE or FALSE (the default value). When TRUE the Startup Sequence following device configuration is paused at a user-specified point until the DLL locks. XAPP176: *Configuration and Readback of the Spartan-II and Spartan-IIE Families* explains how this can result in delaying the assertion of the DONE pin until the DLL locks.

### *DLL Location Constraints*

The DLLs are distributed such that there is one DLL in each corner of the device. The location constraint LOC, attached to the DLL primitive with the numeric identifier 0, 1, 2, or 3, controls DLL location. The orientation of the four DLLs and their corresponding clock resources appears in Figure 27.

The LOC property uses the following form.

LOC = DLL2

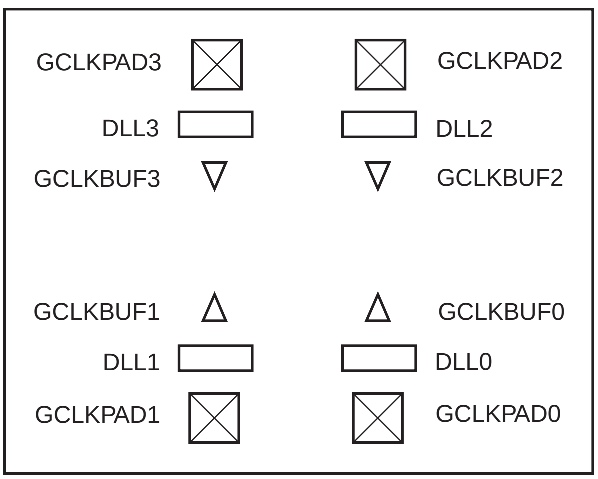



*Figure 27:* **Orientation of DLLs**

## Design Considerations

Use the following design considerations to avoid pitfalls and improve success designing with Xilinx devices.

### *Input Clock*

The output clock signal of a DLL, essentially a delayed version of the input clock signal, reflects any instability on the input clock in the output waveform. For this reason the quality of the DLL input clock relates directly to the quality of the output clock waveforms generated by the DLL. The DLL input clock requirements are specified in the "DLL Timing Parameters" section of the data sheet.

In most systems a crystal oscillator generates the system clock. The DLL can be used with any commercially available quartz crystal oscillator. For example, most crystal oscillators produce an output waveform with a frequency tolerance of 100 PPM, meaning 0.01 percent change in the clock period. The DLL operates reliably on an input waveform with a frequency drift of up to 1 ns — orders of magnitude in excess of that needed to support any crystal oscillator in the industry. However, the cycle-to-cycle jitter must be kept to less than 300 ps in the low frequencies and 150 ps for the high frequencies.

### *Input Clock Changes*

Changing the period of the input clock beyond the maximum drift amount requires a manual reset of the CLKDLL. Failure to reset the DLL will produce an unreliable lock signal and output clock.

It is possible to stop the input clock in a way that has little impact to the DLL. Stopping the clock should be limited to less than approximately 100 µs to keep device cooling to a minimum and maintain the validity of the current tap setting. The clock should be stopped during a Low phase, and when restored the full High period should be seen. During this time LOCKED will stay High and remain High when the clock is restored. If these conditions may not be met in the design, apply a manual reset to the DLL after re-starting the input clock, even if the LOCKED signal has not changed.

When the clock is stopped, one to four more clocks will still be observed as the delay line is flushed. When the clock is restarted, the output clocks will not be observed for one to four clocks as the delay line is filled. The most common case will be two or three clocks.

In a similar manner, a phase shift of the input clock is also possible. The phase shift will propagate to the output one to four clocks after the original shift, with no disruption to the CLKDLL control.

### *Output Clocks*

As mentioned earlier in the DLL pin descriptions, some restrictions apply regarding the connectivity of the output pins. The DLL clock outputs can drive an OBUF, a global clock buffer BUFG, or route directly to destination clock pins. The only BUFGs that the DLL clock outputs can drive are the two on the same edge of the device (top or bottom). One DLL output can drive more than one OBUF; however, this adds skew.

Do not use the DLL output clock signals until after activation of the LOCKED signal. Prior to the activation of the LOCKED signal, the DLL output clocks are not valid and can exhibit glitches, spikes, or other spurious movement.

## Creating Larger RAM Structures

The block RAM columns have specialized routing to allow cascading blocks together with minimal routing delays. This achieves wider or deeper RAM structures with a smaller timing penalty than when using normal routing channels.

## Location Constraints

Block RAM instances can have LOC properties attached to them to constrain the placement. The block RAM placement locations are separate from the CLB location naming convention, allowing the LOC properties to transfer easily from array to array.

The LOC properties use the following form:

LOC = RAMB4_R#C#

RAMB4_R0C0 is the upper left RAMB4 location on the device.

## Conflict Resolution

The block RAM memory is a true dual-read/write port RAM that allows simultaneous access of the same memory cell from both ports. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window. The following lists specifics of port and memory cell write conflict resolution.

- If both ports write to the same memory cell simultaneously, violating the clock-to-clock setup requirement, consider the data stored as invalid.
- If one port attempts a read of the same memory cell the other simultaneously writes, violating the clock-to-clock setup requirement, the following occurs.
  - The write succeeds
  - The data out on the writing port accurately reflects the data written.
  - The data out on the reading port is invalid.

Conflicts do not cause any physical damage.

### *Single Port Timing*

Figure 33 shows a timing diagram for a single port of a block RAM memory. The block RAM AC switching characteristics are specified in the data sheet. The block RAM memory is initially disabled.

At the first rising edge of the CLK pin, the ADDR, DI, EN, WE, and RST pins are sampled. The EN pin is High and the WE pin is Low indicating a read operation. The DO bus contains the contents of the memory location, 0x00, as indicated by the ADDR bus.

At the second rising edge of the CLK pin, the ADDR, DI, EN, WR, and RST pins are sampled again. The EN and WE pins are High indicating a write operation. The DO bus mirrors the DI bus. The DI bus is written to the memory location 0x0F.

At the third rising edge of the CLK pin, the ADDR, DI, EN, WR, and RST pins are sampled again. The EN pin is High and the WE pin is Low indicating a read operation. The DO bus contains the contents of the memory location 0x7E as indicated by the ADDR bus.

At the fourth rising edge of the CLK pin, the ADDR, DI, EN, WR, and RST pins are sampled again. The EN pin is Low indicating that the block RAM memory is now disabled. The DO bus retains the last value.

### *Dual Port Timing*

Figure 34 shows a timing diagram for a true dual-port read/write block RAM memory. The clock on port A has a longer period than the clock on Port B. The timing parameter $T_{BCCS}$, (clock-to-clock setup) is shown on this diagram. The parameter, $T_{BCCS}$ is violated once in the diagram. All other timing parameters are identical to the single port version shown in Figure 33.

$T_{BCCS}$ is only of importance when the address of both ports are the same and at least one port is performing a write operation. When the clock-to-clock set-up parameter is violated for a WRITE-WRITE condition, the contents of the memory at that location will be invalid. When the clock-to-clock set-up parameter is violated for a WRITE-READ condition, the contents of the memory will be correct, but the read port will have invalid data. At the first rising edge of the CLKA, memory location 0x00 is to be written with the value 0xAAAA and is mirrored on the DOA bus. The last operation of Port B was a read to the same memory location 0x00. The DOB bus of Port B does not change with the new value on Port A, and retains the last read value. A short time later, Port B executes another read to memory location 0x00, and the DOB bus now reflects the new memory value written by Port A.

At the second rising edge of CLKA, memory location 0x7E is written with the value 0x9999 and is mirrored on the DOA bus. Port B then executes a read operation to the same memory location without violating the $T_{BCCS}$ parameter and the DOB reflects the new memory values written by Port A.

### *PCI — Peripheral Component Interface*

The Peripheral Component Interface (PCI) standard specifies support for both 33 MHz and 66 MHz PCI bus applications. It uses a LVTTL input buffer and a push-pull output buffer. This standard does not require the use of a reference voltage ($V_{REF}$) or a board termination voltage ($V_{TT}$), however, it does require a 3.3V output source voltage ($V_{CCO}$). I/Os configured for the PCI, 33 MHz, 5V standard are also 5V-tolerant.

### *GTL — Gunning Transceiver Logic Terminated*

The Gunning Transceiver Logic (GTL) standard is a high-speed bus standard (JESD8.3). Xilinx has implemented the terminated variation of this standard. This standard requires a differential amplifier input buffer and an open-drain output buffer.

### *GTL+ — Gunning Transceiver Logic Plus*

The Gunning Transceiver Logic Plus (GTL+) standard is a high-speed bus standard (JESD8.3).

### *HSTL — High-Speed Transceiver Logic*

The High-Speed Transceiver Logic (HSTL) standard is a general purpose high-speed, 1.5V bus standard (EIA/JESD 8-6). This standard has four variations or classes. Versatile I/O devices support Class I, III, and IV. This standard requires a Differential Amplifier input buffer and a Push-Pull output buffer.

### *SSTL3 — Stub Series Terminated Logic for 3.3V*

The Stub Series Terminated Logic for 3.3V (SSTL3) standard is a general purpose 3.3V memory bus standard (JESD8-8). This standard has two classes, I and II. Versatile I/O devices support both classes for the SSTL3 standard. This standard requires a Differential Amplifier input buffer and an Push-Pull output buffer.

### *SSTL2 — Stub Series Terminated Logic for 2.5V*

The Stub Series Terminated Logic for 2.5V (SSTL2) standard is a general purpose 2.5V memory bus standard (JESD8-9). This standard has two classes, I and II. Versatile I/O devices support both classes for the SSTL2 standard. This standard requires a Differential Amplifier input buffer and an Push-Pull output buffer.

### *CTT — Center Tap Terminated*

The Center Tap Terminated (CTT) standard is a 3.3V memory bus standard (JESD8-4). This standard requires a Differential Amplifier input buffer and a Push-Pull output buffer.

### *AGP-2X — Advanced Graphics Port*

The AGP standard is a 3.3V Advanced Graphics Port-2X bus standard used with processors for graphics applications. This standard requires a Push-Pull output buffer and a Differential Amplifier input buffer.

## Library Primitives

The Xilinx library includes an extensive list of primitives designed to provide support for the variety of Versatile I/O features. Most of these primitives represent variations of the five generic Versatile I/O primitives:

- IBUF (input buffer)
- IBUFG (global clock input buffer)
- OBUF (output buffer)
- OBUFT (3-state output buffer)
- IOBUF (input/output buffer)

These primitives are available with various extensions to define the desired I/O standard. However, it is recommended that customers use a a property or attribute on the generic primitive to specify the I/O standard. See "Versatile I/O Properties".

### *IBUF*

Signals used as inputs to the Spartan-II device must source an input buffer (IBUF) via an external input port. The generic IBUF primitive appears in Figure 35. The assumed standard is LVTTL when the generic IBUF has no specified extension or property.

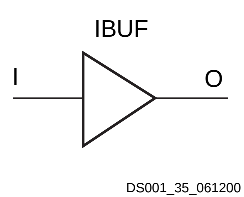



*Figure 35:* **Input Buffer (IBUF) Primitive**

When the IBUF primitive supports an I/O standard such as LVTTL, LVCMOS, or PCI33_5, the IBUF automatically configures as a 5V tolerant input buffer unless the $V_{CCO}$ for the bank is less than 2V. If the single-ended IBUF is placed in a bank with an HSTL standard ($V_{CCO}$ < 2V), the input buffer is not 5V tolerant.

The voltage reference signal is "banked" within the Spartan-II device on a half-edge basis such that for all packages there are eight independent $V_{REF}$ banks internally. See Figure 36 for a representation of the I/O banks. Within each bank approximately one of every six I/O pins is automatically configured as a $V_{REF}$ input.

IBUF placement restrictions require that any differential amplifier input signals within a bank be of the same standard. How to specify a specific location for the IBUF via the LOC property is described below. Table 16 summarizes the input standards compatibility requirements.

An optional delay element is associated with each IBUF. When the IBUF drives a flip-flop within the IOB, the delay element by default activates to ensure a zero hold-time requirement. The NODELAY=TRUE property overrides this default.

When the IBUF does not drive a flip-flop within the IOB, the delay element de-activates by default to provide higher performance. To delay the input signal, activate the delay element with the DELAY=TRUE property.

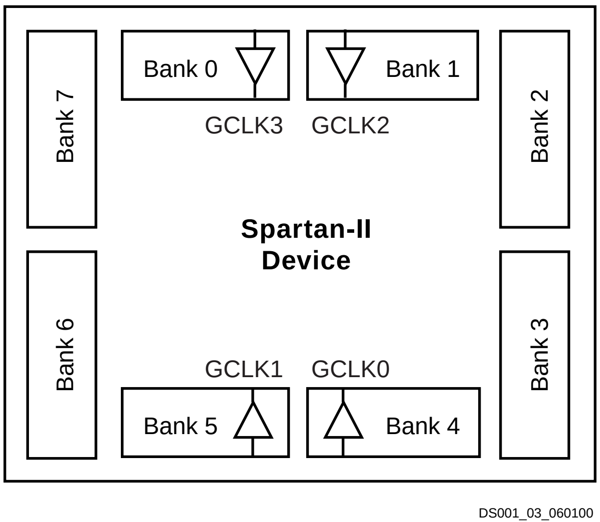



*Figure 36:* **I/O Banks**

*Table 16:* **Xilinx Input Standards Compatibility Requirements**

| | |
|---|---|
| Rule 1 | All differential amplifier input signals within a bank are required to be of the same standard. |
| Rule 2 | There are no placement restrictions for inputs with standards that require a single-ended input buffer. |

### IBUFG

Signals used as high fanout clock inputs to the Spartan-II device should drive a global clock input buffer (IBUFG) via an external input port in order to take advantage of one of the four dedicated global clock distribution networks. The output of the IBUFG primitive can only drive a CLKDLL, CLKDLLHF, or a BUFG primitive. The generic IBUFG primitive appears in Figure 37.

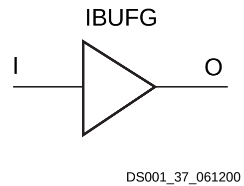



*Figure 37:* **Global Clock Input Buffer (IBUFG) Primitive**

With no extension or property specified for the generic IBUFG primitive, the assumed standard is LVTTL.

The voltage reference signal is "banked" within the Spartan-II device on a half-edge basis such that for all packages there are eight independent $V_{REF}$ banks internally. See Figure 36 for a representation of the I/O banks. Within each bank approximately one of every six I/O pins is automatically configured as a $V_{REF}$ input.

IBUFG placement restrictions require any differential amplifier input signals within a bank be of the same standard. The LOC property can specify a location for the IBUFG.

As an added convenience, the BUFGP can be used to instantiate a high fanout clock input. The BUFGP primitive represents a combination of the LVTTL IBUFG and BUFG primitives, such that the output of the BUFGP can connect directly to the clock pins throughout the design.

The Spartan-II FPGA BUFGP primitive can only be placed in a global clock pad location. The LOC property can specify a location for the BUFGP.

### OBUF

An OBUF must drive outputs through an external output port. The generic output buffer (OBUF) primitive appears in Figure 38.

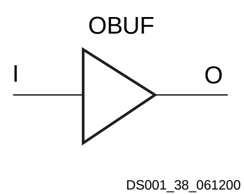



*Figure 38:* **Output Buffer (OBUF) Primitive**

With no extension or property specified for the generic OBUF primitive, the assumed standard is slew rate limited LVTTL with 12 mA drive strength.

The LVTTL OBUF additionally can support one of two slew rate modes to minimize bus transients. By default, the slew rate for each output buffer is reduced to minimize power bus transients when switching non-critical signals.

IOBUF_<slew_rate>_<drive_strength>

<slew_rate> can be either F (Fast), or S (Slow) and <drive_strength> is specified in milliamps (2, 4, 6, 8, 12, 16, or 24).

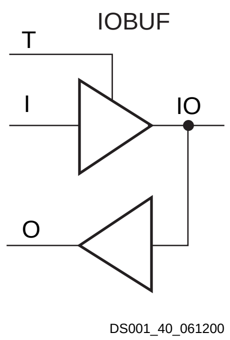



*Figure 40:* **Input/Output Buffer Primitiveprimitive (IOBUF)**

When the IOBUF primitive supports an I/O standard such as LVTTL, LVCMOS, or PCI33_5, the IBUF automatically configures as a 5V tolerant input buffer unless the $V_{CCO}$ for the bank is less than 2V. If the single-ended IBUF is placed in a bank with an HSTL standard ($V_{CCO}$ < 2V), the input buffer is not 5V tolerant.

The voltage reference signal is "banked" within the Spartan-II device on a half-edge basis such that for all packages there are eight independent $V_{REF}$ banks internally. See Figure 36, page 39 for a representation of the Spartan-II FPGA I/O banks. Within each bank approximately one of every six I/O pins is automatically configured as a $V_{REF}$ input.

Additional restrictions on the Versatile I/O IOBUF placement require that within a given $V_{CCO}$ bank each IOBUF must share the same output source drive voltage. Input buffers of any type and output buffers that do not require $V_{CCO}$ can be placed within the same $V_{CCO}$ bank. The LOC property can specify a location for the IOBUF.

An optional delay element is associated with the input path in each IOBUF. When the IOBUF drives an input flip-flop within the IOB, the delay element activates by default to ensure a zero hold-time requirement. Override this default with the NODELAY=TRUE property.

In the case when the IOBUF does not drive an input flip-flop within the IOB, the delay element de-activates by default to provide higher performance. To delay the input signal, activate the delay element with the DELAY=TRUE property.

3-state output buffers and bidirectional buffers can have either a weak pull-up resistor, a weak pull-down resistor, or a weak "keeper" circuit. Control this feature by adding the appropriate primitive to the output net of the IOBUF (PULLUP, PULLDOWN, or KEEPER).

## Versatile I/O Properties

Access to some of the Versatile I/O features (for example, location constraints, input delay, output drive strength, and slew rate) is available through properties associated with these features.

### *Input Delay Properties*

An optional delay element is associated with each IBUF. When the IBUF drives a flip-flop within the IOB, the delay element activates by default to ensure a zero hold-time requirement. Use the NODELAY=TRUE property to override this default.

In the case when the IBUF does not drive a flip-flop within the IOB, the delay element by default de-activates to provide higher performance. To delay the input signal, activate the delay element with the DELAY=TRUE property.

### *IOB Flip-Flop/Latch Property*

The I/O Block (IOB) includes an optional register on the input path, an optional register on the output path, and an optional register on the 3-state control pin. The design implementation software automatically takes advantage of these registers when the following option for the Map program is specified:

map -pr b <filename>

Alternatively, the IOB = TRUE property can be placed on a register to force the mapper to place the register in an IOB.

### *Location Constraints*

Specify the location of each Versatile I/O primitive with the location constraint LOC attached to the Versatile I/O primitive. The external port identifier indicates the value of the location constrain. The format of the port identifier depends on the package chosen for the specific design.

The LOC properties use the following form:

LOC=A42

LOC=P37

### *Output Slew Rate Property*

In the case of the LVTTL output buffers (OBUF, OBUFT, and IOBUF), slew rate control can be programmed with the SLEW= property. By default, the slew rate for each output buffer is reduced to minimize power bus transients when switching non-critical signals. The SLEW= property has one of the two following values.

SLEW=SLOW

SLEW=FAST

### *Output Drive Strength Property*

For the LVTTL output buffers (OBUF, OBUFT, and IOBUF, the desired drive strength can be specified with the DRIVE=

### CTT

A sample circuit illustrating a valid termination technique for CTT appear in Figure 51. DC voltage specifications appear in Table 29 for the CTT standard. See "DC Specifications" in Module 3 for the actual FPGA characteristics .

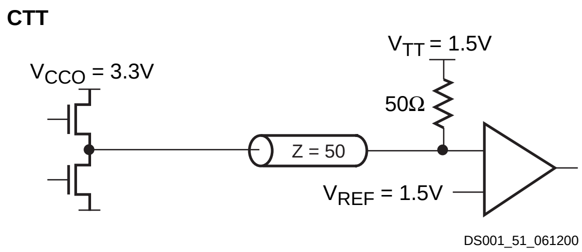



*Figure 51:* **Terminated CTT**

*Table 29:* **CTT Voltage Specifications**

| Parameter | Min | Typ | Max |
|---|---|---|---|
| $V_{CCO}$ | 2.05(1) | 3.3 | 3.6 |
| $V_{REF}$ | 1.35 | 1.5 | 1.65 |
| $V_{TT}$ | 1.35 | 1.5 | 1.65 |
| $V_{IH} \geq V_{REF} + 0.2$ | 1.55 | 1.7 | - |
| $V_{IL} \leq V_{REF} - 0.2$ | - | 1.3 | 1.45 |
| $V_{OH} \geq V_{REF} + 0.4$ | 1.75 | 1.9 | - |
| $V_{OL} \leq V_{REF} - 0.4$ | - | 1.1 | 1.25 |
| $I_{OH}$ at $V_{OH}$ (mA) | –8 | - | - |
| $I_{OL}$ at $V_{OL}$ (mA) | 8 | - | - |

**Notes:**
1. Timing delays are calculated based on $V_{CCO}$ min of 3.0V.

### PCI33_3 and PCI66_3

PCI33_3 or PCI66_3 require no termination. DC voltage specifications appear in Table 30 for the PCI33_3 and PCI66_3 standards. See "DC Specifications" in Module 3 for the actual FPGA characteristics.

*Table 30:* **PCI33_3 and PCI66_3 Voltage Specifications**

| Parameter | Min | Typ | Max |
|---|---|---|---|
| $V_{CCO}$ | 3.0 | 3.3 | 3.6 |
| $V_{REF}$ | - | - | - |
| $V_{TT}$ | - | - | - |
| $V_{IH} = 0.5 \times V_{CCO}$ | 1.5 | 1.65 | $V_{CCO}$+ 0.5 |
| $V_{IL} = 0.3 \times V_{CCO}$ | –0.5 | 0.99 | 1.08 |
| $V_{OH} = 0.9 \times V_{CCO}$ | 2.7 | - | - |
| $V_{OL} = 0.1 \times V_{CCO}$ | - | - | 0.36 |
| $I_{OH}$ at $V_{OH}$ (mA) | Note 1 | - | - |
| $I_{OL}$ at $V_{OL}$ (mA) | Note 1 | - | - |

**Notes:**
1. Tested according to the relevant specification.

### PCI33_5

PCI33_5 requires no termination. DC voltage specifications appear in Table 31 for the PCI33_5 standard. See "DC Specifications" in Module 3 for the actual FPGA characteristics.

*Table 31:* **PCI33_5 Voltage Specifications**

| Parameter | Min | Typ | Max |
|---|---|---|---|
| $V_{CCO}$ | 3.0 | 3.3 | 3.6 |
| $V_{REF}$ | - | - | - |
| $V_{TT}$ | - | - | - |
| $V_{IH}$ | 1.425 | 1.5 | 5.5 |
| $V_{IL}$ | –0.5 | 1.0 | 1.05 |
| $V_{OH}$ | 2.4 | - | - |
| $V_{OL}$ | - | - | 0.55 |
| $I_{OH}$ at $V_{OH}$ (mA) | Note 1 | - | - |
| $I_{OL}$ at $V_{OL}$ (mA) | Note 1 | - | - |

**Notes:**
1. Tested according to the relevant specification.

## Power-On Requirements

Spartan-II FPGAs require that a minimum supply current $I_{CCPO}$ be provided to the $V_{CCINT}$ lines for a successful power-on. If more current is available, the FPGA can consume more than $I_{CCPO}$ minimum, though this cannot adversely affect reliability.

A maximum limit for $I_{CCPO}$ is not specified. Therefore the use of foldback/crowbar supplies and fuses deserves special attention. In these cases, limit the $I_{CCPO}$ current to a level below the trip point for over-current protection in order to avoid inadvertently shutting down the supply.

| Symbol | Description | Conditions | | New Requirements[1] For Devices with Date Code 0321 or Later | | Old Requirements[1] For Devices with Date Code before 0321 | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Junction Temperature[2] | Device Temperature Grade | Min | Max | Min | Max | |
| $I_{CCPO}$[3] | Total $V_{CCINT}$ supply current required during power-on | $-40°C \leq T_J < -20°C$ | Industrial | 1.50 | - | 2.00 | - | A |
| | | $-20°C \leq T_J < 0°C$ | Industrial | 1.00 | - | 2.00 | - | A |
| | | $0°C \leq T_J \leq 85°C$ | Commercial | 0.25 | - | 0.50 | - | A |
| | | $85°C < T_J \leq 100°C$ | Industrial | 0.50 | - | 0.50 | - | A |
| $T_{CCPO}$[4,5] | $V_{CCINT}$ ramp time | $-40°C \leq T_J \leq 100°C$ | All | - | 50 | - | 50 | ms |

**Notes:**

1. The date code is printed on the top of the device's package. See the "Device Part Marking" section in Module 1.
2. The expected $T_J$ range for the design determines the $I_{CCPO}$ minimum requirement. Use the applicable ranges in the junction temperature column to find the associated current values in the appropriate new or old requirements column according to the date code. Then choose the highest of these current values to serve as the minimum $I_{CCPO}$ requirement that must be met. For example, if the junction temperature for a given design is $-25°C \leq T_J \leq 75°C$, then the new minimum $I_{CCPO}$ requirement is 1.5A. If $5°C \leq T_J \leq 90°C$, then the new minimum $I_{CCPO}$ requirement is 0.5A.
3. The $I_{CCPO}$ requirement applies for a brief time (commonly only a few milliseconds) when $V_{CCINT}$ ramps from 0 to 2.5V.
4. The ramp time is measured from GND to $V_{CCINT}$ max on a fully loaded board.
5. During power-on, the $V_{CCINT}$ ramp must increase steadily in voltage with no dips.
6. For more information on designing to meet the power-on specifications, refer to the application note **XAPP450 "Power-On Current Requirements for the Spartan-II and Spartan-IIE Families"**

## DC Input and Output Levels

Values for $V_{IL}$ and $V_{IH}$ are recommended input voltages. Values for $V_{OL}$ and $V_{OH}$ are guaranteed output voltages over the recommended operating conditions. Only selected standards are tested. These are chosen to ensure that all standards meet their specifications. The selected standards are tested at minimum $V_{CCO}$ with the respective $I_{OL}$ and $I_{OH}$ currents shown. Other standards are sample tested.

| Input/Output Standard | $V_{IL}$ | | $V_{IH}$ | | $V_{OL}$ | $V_{OH}$ | $I_{OL}$ | $I_{OH}$ |
|---|---|---|---|---|---|---|---|---|
| | V, Min | V, Max | V, Min | V, Max | V, Max | V, Min | mA | mA |
| LVTTL[1] | –0.5 | 0.8 | 2.0 | 5.5 | 0.4 | 2.4 | 24 | –24 |
| LVCMOS2 | –0.5 | 0.7 | 1.7 | 5.5 | 0.4 | 1.9 | 12 | –12 |
| PCI, 3.3V | –0.5 | 44% $V_{CCINT}$ | 60% $V_{CCINT}$ | $V_{CCO} + 0.5$ | 10% $V_{CCO}$ | 90% $V_{CCO}$ | Note (2) | Note (2) |
| PCI, 5.0V | –0.5 | 0.8 | 2.0 | 5.5 | 0.55 | 2.4 | Note (2) | Note (2) |
| GTL | –0.5 | $V_{REF} - 0.05$ | $V_{REF} + 0.05$ | 3.6 | 0.4 | N/A | 40 | N/A |
| GTL+ | –0.5 | $V_{REF} - 0.1$ | $V_{REF} + 0.1$ | 3.6 | 0.6 | N/A | 36 | N/A |
| HSTL I | –0.5 | $V_{REF} - 0.1$ | $V_{REF} + 0.1$ | 3.6 | 0.4 | $V_{CCO} - 0.4$ | 8 | –8 |
| HSTL III | –0.5 | $V_{REF} - 0.1$ | $V_{REF} + 0.1$ | 3.6 | 0.4 | $V_{CCO} - 0.4$ | 24 | –8 |
| HSTL IV | –0.5 | $V_{REF} - 0.1$ | $V_{REF} + 0.1$ | 3.6 | 0.4 | $V_{CCO} - 0.4$ | 48 | –8 |
| SSTL3 I | –0.5 | $V_{REF} - 0.2$ | $V_{REF} + 0.2$ | 3.6 | $V_{REF} - 0.6$ | $V_{REF} + 0.6$ | 8 | –8 |
| SSTL3 II | –0.5 | $V_{REF} - 0.2$ | $V_{REF} + 0.2$ | 3.6 | $V_{REF} - 0.8$ | $V_{REF} + 0.8$ | 16 | –16 |
| SSTL2 I | –0.5 | $V_{REF} - 0.2$ | $V_{REF} + 0.2$ | 3.6 | $V_{REF} - 0.6$ | $V_{REF} + 0.6$ | 7.6 | –7.6 |
| SSTL2 II | –0.5 | $V_{REF} - 0.2$ | $V_{REF} + 0.2$ | 3.6 | $V_{REF} - 0.8$ | $V_{REF} + 0.8$ | 15.2 | –15.2 |

## IOB Input Switching Characteristics[1]

Input delays associated with the pad are specified for LVTTL levels. For other standards, adjust the delays with the values shown in "IOB Input Delay Adjustments for Different Standards," page 57.

| Symbol | Description | Device | Speed Grade -6 Min | Speed Grade -6 Max | Speed Grade -5 Min | Speed Grade -5 Max | Units |
|---|---|---|---|---|---|---|---|
| **Propagation Delays** | | | | | | | |
| $T_{IOPI}$ | Pad to I output, no delay | All | - | 0.8 | - | 1.0 | ns |
| $T_{IOPID}$ | Pad to I output, with delay | All | - | 1.5 | - | 1.8 | ns |
| $T_{IOPLI}$ | Pad to output IQ via transparent latch, no delay | All | - | 1.7 | - | 2.0 | ns |
| $T_{IOPLID}$ | Pad to output IQ via transparent latch, with delay | XC2S15 | - | 3.8 | - | 4.5 | ns |
| | | XC2S30 | - | 3.8 | - | 4.5 | ns |
| | | XC2S50 | - | 3.8 | - | 4.5 | ns |
| | | XC2S100 | - | 3.8 | - | 4.5 | ns |
| | | XC2S150 | - | 4.0 | - | 4.7 | ns |
| | | XC2S200 | - | 4.0 | - | 4.7 | ns |
| **Sequential Delays** | | | | | | | |
| $T_{IOCKIQ}$ | Clock CLK to output IQ | All | - | 0.7 | - | 0.8 | ns |
| **Setup/Hold Times with Respect to Clock CLK[2]** | | | | | | | |
| $T_{IOPICK}$ / $T_{IOICKP}$ | Pad, no delay | All | 1.7 / 0 | - | 1.9 / 0 | - | ns |
| $T_{IOPICKD}$ / $T_{IOICKPD}$ | Pad, with delay[1] | XC2S15 | 3.8 / 0 | - | 4.4 / 0 | - | ns |
| | | XC2S30 | 3.8 / 0 | - | 4.4 / 0 | - | ns |
| | | XC2S50 | 3.8 / 0 | - | 4.4 / 0 | - | ns |
| | | XC2S100 | 3.8 / 0 | - | 4.4 / 0 | - | ns |
| | | XC2S150 | 3.9 / 0 | - | 4.6 / 0 | - | ns |
| | | XC2S200 | 3.9 / 0 | - | 4.6 / 0 | - | ns |
| $T_{IOICECK}$ / $T_{IOCKICE}$ | ICE input | All | 0.9 / 0.01 | - | 0.9 / 0.01 | - | ns |
| **Set/Reset Delays** | | | | | | | |
| $T_{IOSRCKI}$ | SR input (IFF, synchronous) | All | - | 1.1 | - | 1.2 | ns |
| $T_{IOSRIQ}$ | SR input to IQ (asynchronous) | All | - | 1.5 | - | 1.7 | ns |
| $T_{GSRQ}$ | GSR to output IQ | All | - | 9.9 | - | 11.7 | ns |

**Notes:**
1. Input timing for LVTTL is measured at 1.4V. For other I/O standards, see the table "Delay Measurement Methodology," page 60.
2. A zero hold time listing indicates no hold time or a negative hold time.

## Calculation of $T_{IOOP}$ as a Function of Capacitance

$T_{IOOP}$ is the propagation delay from the O Input of the IOB to the pad. The values for $T_{IOOP}$ are based on the standard capacitive load ($C_{SL}$) for each I/O standard as listed in the table "Constants for Calculating TIOOP", below.

For other capacitive loads, use the formulas below to calculate an adjusted propagation delay, $T_{IOOP1}$.

$$T_{IOOP1} = T_{IOOP} + Adj + (C_{LOAD} - C_{SL}) * F_L$$

Where:

Adj is selected from "IOB Output Delay Adjustments for Different Standards", page 59, according to the I/O standard used

$C_{LOAD}$ is the capacitive load for the design

$F_L$ is the capacitance scaling factor

## Delay Measurement Methodology

| Standard | $V_L$(1) | $V_H$(1) | Meas. Point | $V_{REF}$ Typ(2) |
|---|---|---|---|---|
| LVTTL | 0 | 3 | 1.4 | - |
| LVCMOS2 | 0 | 2.5 | 1.125 | - |
| PCI33_5 | Per PCI Spec | | | - |
| PCI33_3 | Per PCI Spec | | | - |
| PCI66_3 | Per PCI Spec | | | - |
| GTL | $V_{REF} - 0.2$ | $V_{REF} + 0.2$ | $V_{REF}$ | 0.80 |
| GTL+ | $V_{REF} - 0.2$ | $V_{REF} + 0.2$ | $V_{REF}$ | 1.0 |
| HSTL Class I | $V_{REF} - 0.5$ | $V_{REF} + 0.5$ | $V_{REF}$ | 0.75 |
| HSTL Class III | $V_{REF} - 0.5$ | $V_{REF} + 0.5$ | $V_{REF}$ | 0.90 |
| HSTL Class IV | $V_{REF} - 0.5$ | $V_{REF} + 0.5$ | $V_{REF}$ | 0.90 |
| SSTL3 I and II | $V_{REF} - 1.0$ | $V_{REF} + 1.0$ | $V_{REF}$ | 1.5 |
| SSTL2 I and II | $V_{REF} - 0.75$ | $V_{REF} + 0.75$ | $V_{REF}$ | 1.25 |
| CTT | $V_{REF} - 0.2$ | $V_{REF} + 0.2$ | $V_{REF}$ | 1.5 |
| AGP | $V_{REF} - (0.2xV_{CCO})$ | $V_{REF} + (0.2xV_{CCO})$ | $V_{REF}$ | Per AGP Spec |

**Notes:**

1. Input waveform switches between $V_L$ and $V_H$.
2. Measurements are made at $V_{REF}$ Typ, Maximum, and Minimum. Worst-case values are reported.
3. I/O parameter measurements are made with the capacitance values shown in the table, "Constants for Calculating TIOOP". See Xilinx application note XAPP179 for the appropriate terminations.
4. I/O standard measurements are reflected in the IBIS model information except where the IBIS format precludes it.

## Constants for Calculating $T_{IOOP}$

| Standard | $C_{SL}$(1) (pF) | $F_L$ (ns/pF) |
|---|---|---|
| LVTTL Fast Slew Rate, 2 mA drive | 35 | 0.41 |
| LVTTL Fast Slew Rate, 4 mA drive | 35 | 0.20 |
| LVTTL Fast Slew Rate, 6 mA drive | 35 | 0.13 |
| LVTTL Fast Slew Rate, 8 mA drive | 35 | 0.079 |
| LVTTL Fast Slew Rate, 12 mA drive | 35 | 0.044 |
| LVTTL Fast Slew Rate, 16 mA drive | 35 | 0.043 |
| LVTTL Fast Slew Rate, 24 mA drive | 35 | 0.033 |
| LVTTL Slow Slew Rate, 2 mA drive | 35 | 0.41 |
| LVTTL Slow Slew Rate, 4 mA drive | 35 | 0.20 |
| LVTTL Slow Slew Rate, 6 mA drive | 35 | 0.100 |
| LVTTL Slow Slew Rate, 8 mA drive | 35 | 0.086 |
| LVTTL Slow Slew Rate, 12 mA drive | 35 | 0.058 |
| LVTTL Slow Slew Rate, 16 mA drive | 35 | 0.050 |
| LVTTL Slow Slew Rate, 24 mA drive | 35 | 0.048 |
| LVCMOS2 | 35 | 0.041 |
| PCI 33 MHz 5V | 50 | 0.050 |
| PCI 33 MHZ 3.3V | 10 | 0.050 |
| PCI 66 MHz 3.3V | 10 | 0.033 |
| GTL | 0 | 0.014 |
| GTL+ | 0 | 0.017 |
| HSTL Class I | 20 | 0.022 |
| HSTL Class III | 20 | 0.016 |
| HSTL Class IV | 20 | 0.014 |
| SSTL2 Class I | 30 | 0.028 |
| SSTL2 Class II | 30 | 0.016 |
| SSTL3 Class I | 30 | 0.029 |
| SSTL3 Class II | 30 | 0.016 |
| CTT | 20 | 0.035 |
| AGP | 10 | 0.037 |

**Notes:**

1. I/O parameter measurements are made with the capacitance values shown above. See Xilinx application note XAPP179 for the appropriate terminations.
2. I/O standard measurements are reflected in the IBIS model information except where the IBIS format precludes it.

## Block RAM Switching Characteristics

| Symbol | Description | Speed Grade -6 Min | -6 Max | -5 Min | -5 Max | Units |
|---|---|---|---|---|---|---|
| **Sequential Delays** | | | | | | |
| $T_{BCKO}$ | Clock CLK to DOUT output | - | 3.4 | - | 4.0 | ns |
| **Setup/Hold Times with Respect to Clock CLK(1)** | | | | | | |
| $T_{BACK}$ / $T_{BCKA}$ | ADDR inputs | 1.4 / 0 | - | 1.4 / 0 | - | ns |
| $T_{BDCK}$/ $T_{BCKD}$ | DIN inputs | 1.4 / 0 | - | 1.4 / 0 | - | ns |
| $T_{BECK}$/ $T_{BCKE}$ | EN inputs | 2.9 / 0 | - | 3.2 / 0 | - | ns |
| $T_{BRCK}$/ $T_{BCKR}$ | RST input | 2.7 / 0 | - | 2.9 / 0 | - | ns |
| $T_{BWCK}$/ $T_{BCKW}$ | WEN input | 2.6 / 0 | - | 2.8 / 0 | - | ns |
| **Clock CLK** | | | | | | |
| $T_{BPWH}$ | Minimum pulse width, High | - | 1.9 | - | 1.9 | ns |
| $T_{BPWL}$ | Minimum pulse width, Low | - | 1.9 | - | 1.9 | ns |
| $T_{BCCS}$ | CLKA -> CLKB setup time for different ports | - | 3.0 | - | 4.0 | ns |

**Notes:**
1. A zero hold time listing indicates no hold time or a negative hold time.

## TBUF Switching Characteristics

| Symbol | Description | Speed Grade -6 Max | -5 Max | Units |
|---|---|---|---|---|
| **Combinatorial Delays** | | | | |
| $T_{IO}$ | IN input to OUT output | 0 | 0 | ns |
| $T_{OFF}$ | TRI input to OUT output high impedance | 0.1 | 0.2 | ns |
| $T_{ON}$ | TRI input to valid data on OUT output | 0.1 | 0.2 | ns |

## JTAG Test Access Port Switching Characteristics

| Symbol | Description | Speed Grade -6 Min | -6 Max | -5 Min | -5 Max | Units |
|---|---|---|---|---|---|---|
| **Setup and Hold Times with Respect to TCK** | | | | | | |
| $T_{TAPTCK}$ / $T_{TCKTAP}$ | TMS and TDI setup and hold times | 4.0 / 2.0 | - | 4.0 / 2.0 | - | ns |
| **Sequential Delays** | | | | | | |
| $T_{TCKTDO}$ | Output delay from clock TCK to output TDO | - | 11.0 | - | 11.0 | ns |
| FTCK | Maximum TCK clock frequency | - | 33 | - | 33 | MHz |

## XC2S50 Device Pinouts *(Continued)*

| XC2S50 Pad Name | | | | | |
|---|---|---|---|---|---|
| Function | Bank | TQ144 | PQ208 | FG256 | Bndry Scan |
| I/O | 3 | - | - | J14 | 503 |
| I/O | 3 | P56 | P127 | K15 | 506 |
| $V_{CCINT}$ | - | P55 | P128 | $V_{CCINT}$* | - |
| I/O, TRDY(1) | 3 | P54 | P129 | J15 | 512 |
| $V_{CCO}$ | 3 | P53 | P130 | $V_{CCO}$ Bank 3* | - |
| $V_{CCO}$ | 2 | P53 | P130 | $V_{CCO}$ Bank 2* | - |
| GND | - | P52 | P131 | GND* | - |
| I/O, IRDY(1) | 2 | P51 | P132 | H16 | 515 |
| I/O | 2 | - | P133 | H14 | 518 |
| I/O | 2 | P50 | P134 | H15 | 521 |
| I/O | 2 | - | - | J13 | 524 |
| I/O (D3) | 2 | P49 | P135 | G16 | 527 |
| I/O, $V_{REF}$ | 2 | P48 | P136 | H13 | 530 |
| GND | - | - | P137 | GND* | - |
| I/O | 2 | - | P138 | G14 | 533 |
| I/O | 2 | - | P139 | G15 | 536 |
| I/O | 2 | - | P140 | G12 | 539 |
| I/O | 2 | - | - | F16 | 542 |
| I/O | 2 | P47 | P141 | G13 | 545 |
| I/O (D2) | 2 | P46 | P142 | F15 | 548 |
| $V_{CCINT}$ | - | - | P143 | $V_{CCINT}$* | - |
| $V_{CCO}$ | 2 | - | P144 | $V_{CCO}$ Bank 2* | - |
| GND | - | P45 | P145 | GND* | - |
| I/O (D1) | 2 | P44 | P146 | E16 | 551 |
| I/O | 2 | P43 | P147 | F14 | 554 |
| I/O | 2 | P42 | P148 | D16 | 557 |
| I/O | 2 | - | - | F12 | 560 |
| I/O | 2 | - | P149 | E15 | 563 |
| I/O, $V_{REF}$ | 2 | P41 | P150 | F13 | 566 |
| GND | - | - | - | GND* | - |
| I/O | 2 | - | P151 | E14 | 569 |
| I/O | 2 | - | - | C16 | 572 |
| I/O | 2 | P40 | P152 | E13 | 575 |
| I/O | 2 | - | - | B16 | 578 |
| I/O (DIN, D0) | 2 | P39 | P153 | D14 | 581 |
| I/O (DOUT, BUSY) | 2 | P38 | P154 | C15 | 584 |
| CCLK | 2 | P37 | P155 | D15 | 587 |
| $V_{CCO}$ | 2 | P36 | P156 | $V_{CCO}$ Bank 2* | - |

## XC2S50 Device Pinouts *(Continued)*

| XC2S50 Pad Name | | | | | |
|---|---|---|---|---|---|
| Function | Bank | TQ144 | PQ208 | FG256 | Bndry Scan |
| $V_{CCO}$ | 1 | P35 | P156 | $V_{CCO}$ Bank 1* | - |
| TDO | 2 | P34 | P157 | B14 | - |
| GND | - | P33 | P158 | GND* | - |
| TDI | - | P32 | P159 | A15 | - |
| I/O ($\overline{CS}$) | 1 | P31 | P160 | B13 | 0 |
| I/O ($\overline{WRITE}$) | 1 | P30 | P161 | C13 | 3 |
| I/O | 1 | - | - | C12 | 6 |
| I/O | 1 | P29 | P162 | A14 | 9 |
| I/O | 1 | - | - | D12 | 12 |
| I/O | 1 | - | P163 | B12 | 15 |
| GND | - | - | - | GND* | - |
| I/O, $V_{REF}$ | 1 | P28 | P164 | C11 | 18 |
| I/O | 1 | - | P165 | A13 | 21 |
| I/O | 1 | - | - | D11 | 24 |
| I/O | 1 | - | P166 | A12 | 27 |
| I/O | 1 | P27 | P167 | E11 | 30 |
| I/O | 1 | P26 | P168 | B11 | 33 |
| GND | - | P25 | P169 | GND* | - |
| $V_{CCO}$ | 1 | - | P170 | $V_{CCO}$ Bank 1* | - |
| $V_{CCINT}$ | - | P24 | P171 | $V_{CCINT}$* | - |
| I/O | 1 | P23 | P172 | A11 | 36 |
| I/O | 1 | P22 | P173 | C10 | 39 |
| I/O | 1 | - | P174 | B10 | 45 |
| I/O | 1 | - | P175 | D10 | 48 |
| I/O | 1 | - | P176 | A10 | 51 |
| GND | - | - | P177 | GND* | - |
| I/O, $V_{REF}$ | 1 | P21 | P178 | B9 | 54 |
| I/O | 1 | - | P179 | E10 | 57 |
| I/O | 1 | - | - | A9 | 60 |
| I/O | 1 | P20 | P180 | D9 | 63 |
| I/O | 1 | P19 | P181 | A8 | 66 |
| I, GCK2 | 1 | P18 | P182 | C9 | 72 |
| GND | - | P17 | P183 | GND* | - |
| $V_{CCO}$ | 1 | P16 | P184 | $V_{CCO}$ Bank 1* | - |
| $V_{CCO}$ | 0 | P16 | P184 | $V_{CCO}$ Bank 0* | - |
| I, GCK3 | 0 | P15 | P185 | B8 | 73 |
| $V_{CCINT}$ | - | P14 | P186 | $V_{CCINT}$* | - |
| I/O | 0 | P13 | P187 | A7 | 80 |

## XC2S100 Device Pinouts *(Continued)*

| XC2S100 Pad Name | | | | | | Bndry Scan |
|---|---|---|---|---|---|---|
| Function | Bank | TQ144 | PQ208 | FG256 | FG456 | |
| I/O | 2 | - | - | F12 | G20 | 695 |
| I/O | 2 | - | P149 | E15 | F19 | 701 |
| I/O, $V_{REF}$ | 2 | P41 | P150 | F13 | F21 | 704 |
| $V_{CCO}$ | 2 | - | - | $V_{CCO}$ Bank 2* | $V_{CCO}$ Bank 2* | - |
| GND | - | - | - | GND* | GND* | - |
| I/O | 2 | - | P151 | E14 | F20 | 707 |
| I/O | 2 | - | - | C16 | F18 | 710 |
| I/O | 2 | - | - | - | E21 | 713 |
| I/O | 2 | P40 | P152 | E13 | D22 | 716 |
| I/O | 2 | - | - | B16 | E20 | 719 |
| I/O (DIN, D0) | 2 | P39 | P153 | D14 | D20 | 725 |
| I/O (DOUT, BUSY) | 2 | P38 | P154 | C15 | C21 | 728 |
| CCLK | 2 | P37 | P155 | D15 | B22 | 731 |
| $V_{CCO}$ | 2 | P36 | P156 | $V_{CCO}$ Bank 2* | $V_{CCO}$ Bank 2* | - |
| $V_{CCO}$ | 1 | P35 | P156 | $V_{CCO}$ Bank 1* | $V_{CCO}$ Bank 1* | - |
| TDO | 2 | P34 | P157 | B14 | A21 | - |
| GND | - | P33 | P158 | GND* | GND* | - |
| TDI | - | P32 | P159 | A15 | B20 | - |
| I/O ($\overline{CS}$) | 1 | P31 | P160 | B13 | C19 | 0 |
| I/O ($\overline{WRITE}$) | 1 | P30 | P161 | C13 | A20 | 3 |
| I/O | 1 | - | - | C12 | D17 | 9 |
| I/O | 1 | P29 | P162 | A14 | A19 | 12 |
| I/O | 1 | - | - | - | B18 | 15 |
| I/O | 1 | - | - | D12 | C17 | 18 |
| I/O | 1 | - | P163 | B12 | D16 | 21 |
| GND | - | - | - | GND* | GND* | - |
| $V_{CCO}$ | 1 | - | - | $V_{CCO}$ Bank 1* | $V_{CCO}$ Bank 1* | - |
| I/O, $V_{REF}$ | 1 | P28 | P164 | C11 | A18 | 24 |
| I/O | 1 | - | P165 | A13 | B17 | 27 |
| I/O | 1 | - | - | D11 | D15 | 33 |
| I/O | 1 | - | P166 | A12 | C16 | 36 |
| I/O | 1 | - | - | - | D14 | 39 |
| I/O, $V_{REF}$ | 1 | P27 | P167 | E11 | E14 | 42 |
| I/O | 1 | P26 | P168 | B11 | A16 | 45 |
| GND | - | P25 | P169 | GND* | GND* | - |

## XC2S100 Device Pinouts *(Continued)*

| XC2S100 Pad Name | | | | | | Bndry Scan |
|---|---|---|---|---|---|---|
| Function | Bank | TQ144 | PQ208 | FG256 | FG456 | |
| $V_{CCO}$ | 1 | - | P170 | $V_{CCO}$ Bank 1* | $V_{CCO}$ Bank 1* | - |
| $V_{CCINT}$ | - | P24 | P171 | $V_{CCINT}$* | $V_{CCINT}$* | - |
| I/O | 1 | P23 | P172 | A11 | C15 | 48 |
| I/O | 1 | P22 | P173 | C10 | B15 | 51 |
| I/O | 1 | - | - | - | F12 | 54 |
| I/O | 1 | - | P174 | B10 | C14 | 57 |
| I/O | 1 | - | P175 | D10 | D13 | 63 |
| I/O | 1 | - | P176 | A10 | C13 | 66 |
| GND | - | - | P177 | GND* | GND* | - |
| I/O, $V_{REF}$ | 1 | P21 | P178 | B9 | B13 | 69 |
| I/O | 1 | - | P179 | E10 | E12 | 72 |
| I/O | 1 | - | - | A9 | B12 | 75 |
| I/O | 1 | P20 | P180 | D9 | D12 | 78 |
| I/O | 1 | P19 | P181 | A8 | D11 | 84 |
| I, GCK2 | 1 | P18 | P182 | C9 | A11 | 90 |
| GND | - | P17 | P183 | GND* | GND* | - |
| $V_{CCO}$ | 1 | P16 | P184 | $V_{CCO}$ Bank 1* | $V_{CCO}$ Bank 1* | - |
| $V_{CCO}$ | 0 | P16 | P184 | $V_{CCO}$ Bank 0* | $V_{CCO}$ Bank 0* | - |
| I, GCK3 | 0 | P15 | P185 | B8 | C11 | 91 |
| $V_{CCINT}$ | - | P14 | P186 | $V_{CCINT}$* | $V_{CCINT}$* | - |
| I/O | 0 | P13 | P187 | A7 | A10 | 101 |
| I/O | 0 | - | - | D8 | B10 | 104 |

## XC2S150 Device Pinouts

| XC2S150 Pad Name | | | | | |
|---|---|---|---|---|---|
| Function | Bank | PQ208 | FG256 | FG456 | Bndry Scan |
| GND | - | P1 | GND* | GND* | - |
| TMS | - | P2 | D3 | D3 | - |
| I/O | 7 | P3 | C2 | B1 | 221 |
| I/O | 7 | - | - | E4 | 224 |
| I/O | 7 | - | - | C1 | 227 |
| I/O | 7 | - | A2 | F5 | 230 |
| GND | - | - | GND* | GND* | - |
| I/O | 7 | P4 | B1 | D2 | 233 |
| I/O | 7 | - | - | E3 | 236 |
| I/O | 7 | - | - | F4 | 239 |
| I/O | 7 | - | E3 | G5 | 242 |
| I/O | 7 | P5 | D2 | F3 | 245 |
| GND | - | - | GND* | GND* | - |
| $V_{CCO}$ | 7 | - | $V_{CCO}$ Bank 7* | $V_{CCO}$ Bank 7* | - |
| I/O, $V_{REF}$ | 7 | P6 | C1 | E2 | 248 |
| I/O | 7 | P7 | F3 | E1 | 251 |
| I/O | 7 | - | - | G4 | 254 |
| I/O | 7 | - | - | G3 | 257 |
| I/O | 7 | - | E2 | H5 | 260 |
| I/O | 7 | P8 | E4 | F2 | 263 |
| I/O | 7 | - | - | F1 | 266 |
| I/O, $V_{REF}$ | 7 | P9 | D1 | H4 | 269 |
| I/O | 7 | P10 | E1 | G1 | 272 |
| GND | - | P11 | GND* | GND* | - |
| $V_{CCO}$ | 7 | P12 | $V_{CCO}$ Bank 7* | $V_{CCO}$ Bank 7* | - |
| $V_{CCINT}$ | - | P13 | $V_{CCINT}$* | $V_{CCINT}$* | - |
| I/O | 7 | P14 | F2 | H3 | 275 |
| I/O | 7 | P15 | G3 | H2 | 278 |
| I/O | 7 | - | - | H1 | 284 |
| I/O | 7 | - | F1 | J5 | 287 |
| I/O | 7 | P16 | F4 | J2 | 290 |
| I/O | 7 | - | - | J3 | 293 |
| I/O | 7 | P17 | F5 | K5 | 299 |
| I/O | 7 | P18 | G2 | K1 | 302 |
| GND | - | P19 | GND* | GND* | - |
| $V_{CCO}$ | 7 | - | $V_{CCO}$ Bank 7* | $V_{CCO}$ Bank 7* | - |
| I/O, $V_{REF}$ | 7 | P20 | H3 | K3 | 305 |
| I/O | 7 | P21 | G4 | K4 | 308 |
| I/O | 7 | - | H2 | L6 | 311 |

## XC2S150 Device Pinouts *(Continued)*

| XC2S150 Pad Name | | | | | |
|---|---|---|---|---|---|
| Function | Bank | PQ208 | FG256 | FG456 | Bndry Scan |
| I/O | 7 | P22 | G5 | L1 | 314 |
| I/O | 7 | - | - | L5 | 317 |
| I/O | 7 | P23 | H4 | L4 | 320 |
| I/O, IRDY(1) | 7 | P24 | G1 | L3 | 323 |
| GND | - | P25 | GND* | GND* | - |
| $V_{CCO}$ | 7 | P26 | $V_{CCO}$ Bank 7* | $V_{CCO}$ Bank 7* | - |
| $V_{CCO}$ | 6 | P26 | $V_{CCO}$ Bank 6* | $V_{CCO}$ Bank 6* | - |
| I/O, TRDY(1) | 6 | P27 | J2 | M1 | 326 |
| $V_{CCINT}$ | - | P28 | $V_{CCINT}$* | $V_{CCINT}$* | - |
| I/O | 6 | - | - | M6 | 332 |
| I/O | 6 | P29 | H1 | M3 | 335 |
| I/O | 6 | - | J4 | M4 | 338 |
| I/O | 6 | P30 | J1 | M5 | 341 |
| I/O, $V_{REF}$ | 6 | P31 | J3 | N2 | 344 |
| $V_{CCO}$ | 6 | - | $V_{CCO}$ Bank 6* | $V_{CCO}$ Bank 6* | - |
| GND | - | P32 | GND* | GND* | - |
| I/O | 6 | P33 | K5 | N3 | 347 |
| I/O | 6 | P34 | K2 | N4 | 350 |
| I/O | 6 | - | - | N5 | 356 |
| I/O | 6 | P35 | K1 | P2 | 359 |
| I/O | 6 | - | K3 | P4 | 362 |
| I/O | 6 | - | - | R1 | 365 |
| I/O | 6 | P36 | L1 | P3 | 371 |
| I/O | 6 | P37 | L2 | R2 | 374 |
| $V_{CCINT}$ | - | P38 | $V_{CCINT}$* | $V_{CCINT}$* | - |
| $V_{CCO}$ | 6 | P39 | $V_{CCO}$ Bank 6* | $V_{CCO}$ Bank 6* | - |
| GND | - | P40 | GND* | GND* | - |
| I/O | 6 | P41 | K4 | T1 | 377 |
| I/O, $V_{REF}$ | 6 | P42 | M1 | R4 | 380 |
| I/O | 6 | - | - | T2 | 383 |
| I/O | 6 | P43 | L4 | U1 | 386 |
| I/O | 6 | - | M2 | R5 | 389 |
| I/O | 6 | - | - | V1 | 392 |
| I/O | 6 | - | - | T5 | 395 |
| I/O | 6 | P44 | L3 | U2 | 398 |
| I/O, $V_{REF}$ | 6 | P45 | N1 | T3 | 401 |
| $V_{CCO}$ | 6 | - | $V_{CCO}$ Bank 6* | $V_{CCO}$ Bank 6* | - |
| GND | - | - | GND* | GND* | - |