



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



This document includes all four modules of the Spartan[®]-II FPGA data sheet.

**Module 1:
Introduction and Ordering Information****DS001-1 (v2.8) June 13, 2008**

- Introduction
- Features
- General Overview
- Product Availability
- User I/O Chart
- Ordering Information

**Module 2:
Functional Description****DS001-2 (v2.8) June 13, 2008**

- Architectural Description
 - Spartan-II Array
 - Input/Output Block
 - Configurable Logic Block
 - Block RAM
 - Clock Distribution: Delay-Locked Loop
 - Boundary Scan
- Development System
- Configuration
 - Configuration Timing
- Design Considerations

**Module 3:
DC and Switching Characteristics****DS001-3 (v2.8) June 13, 2008**

- DC Specifications
 - Absolute Maximum Ratings
 - Recommended Operating Conditions
 - DC Characteristics
 - Power-On Requirements
 - DC Input and Output Levels
- Switching Characteristics
 - Pin-to-Pin Parameters
 - IOB Switching Characteristics
 - Clock Distribution Characteristics
 - DLL Timing Parameters
 - CLB Switching Characteristics
 - Block RAM Switching Characteristics
 - TBUF Switching Characteristics
 - JTAG Switching Characteristics

**Module 4:
Pinout Tables****DS001-4 (v2.8) June 13, 2008**

- Pin Definitions
- Pinout Tables

IMPORTANT NOTE: *This Spartan-II FPGA data sheet is in four modules. Each module has its own Revision History at the end. Use the PDF "Bookmarks" for easy navigation in this volume.*

Introduction

The Spartan®-II Field-Programmable Gate Array family gives users high performance, abundant logic resources, and a rich feature set, all at an exceptionally low price. The six-member family offers densities ranging from 15,000 to 200,000 system gates, as shown in [Table 1](#). System performance is supported up to 200 MHz. Features include block RAM (to 56K bits), distributed RAM (to 75,264 bits), 16 selectable I/O standards, and four DLLs. Fast, predictable interconnect means that successive design iterations continue to meet timing requirements.

The Spartan-II family is a superior alternative to mask-programmed ASICs. The FPGA avoids the initial cost, lengthy development cycles, and inherent risk of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary (impossible with ASICs).

Features

- Second generation ASIC replacement technology
 - Densities as high as 5,292 logic cells with up to 200,000 system gates
 - Streamlined features based on Virtex® FPGA architecture
 - Unlimited reprogrammability
 - Very low cost
 - Cost-effective 0.18 micron process
- System level features
 - SelectRAM™ hierarchical memory:
 - 16 bits/LUT distributed RAM
 - Configurable 4K bit block RAM
 - Fast interfaces to external RAM
 - Fully PCI compliant
 - Low-power segmented routing architecture
 - Full readback ability for verification/observability
 - Dedicated carry logic for high-speed arithmetic
 - Efficient multiplier support
 - Cascade chain for wide-input functions
 - Abundant registers/latches with enable, set, reset
 - Four dedicated DLLs for advanced clock control
 - Four primary low-skew global clock distribution nets
 - IEEE 1149.1 compatible boundary scan logic
- Versatile I/O and packaging
 - Pb-free package options
 - Low-cost packages available in all densities
 - Family footprint compatibility in common packages
 - 16 high-performance interface standards
 - Hot swap Compact PCI friendly
 - Zero hold time simplifies system timing
- Core logic powered at 2.5V and I/Os powered at 1.5V, 2.5V, or 3.3V
- Fully supported by powerful Xilinx® ISE® development system
 - Fully automatic mapping, placement, and routing

Table 1: Spartan-II FPGA Family Members

Device	Logic Cells	System Gates (Logic and RAM)	CLB Array (R x C)	Total CLBs	Maximum Available User I/O ⁽¹⁾	Total Distributed RAM Bits	Total Block RAM Bits
XC2S15	432	15,000	8 x 12	96	86	6,144	16K
XC2S30	972	30,000	12 x 18	216	92	13,824	24K
XC2S50	1,728	50,000	16 x 24	384	176	24,576	32K
XC2S100	2,700	100,000	20 x 30	600	176	38,400	40K
XC2S150	3,888	150,000	24 x 36	864	260	55,296	48K
XC2S200	5,292	200,000	28 x 42	1,176	284	75,264	56K

Notes:

1. All user I/O counts do not include the four global clock/user input pins. See details in [Table 2, page 4](#).

General Overview

The Spartan-II family of FPGAs have a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), surrounded by a perimeter of programmable Input/Output Blocks (IOBs). There are four Delay-Locked Loops (DLLs), one at each corner of the die. Two columns of block RAM lie on opposite sides of the die, between the CLBs and the IOB columns. These functional elements are interconnected by a powerful hierarchy of versatile routing channels (see Figure 1).

Spartan-II FPGAs are customized by loading configuration data into internal static memory cells. Unlimited reprogramming cycles are possible with this approach. Stored values in these cells determine logic functions and interconnections implemented in the FPGA. Configuration data can be read from an external serial PROM (master

serial mode), or written into the FPGA in slave serial, slave parallel, or Boundary Scan modes.

Spartan-II FPGAs are typically used in high-volume applications where the versatility of a fast programmable solution adds benefits. Spartan-II FPGAs are ideal for shortening product development cycles while offering a cost-effective solution for high volume production.

Spartan-II FPGAs achieve high-performance, low-cost operation through advanced architecture and semiconductor technology. Spartan-II devices provide system clock rates up to 200 MHz. In addition to the conventional benefits of high-volume programmable logic solutions, Spartan-II FPGAs also offer on-chip synchronous single-port and dual-port RAM (block and distributed form), DLL clock drivers, programmable set and reset on all flip-flops, fast carry logic, and many other features.

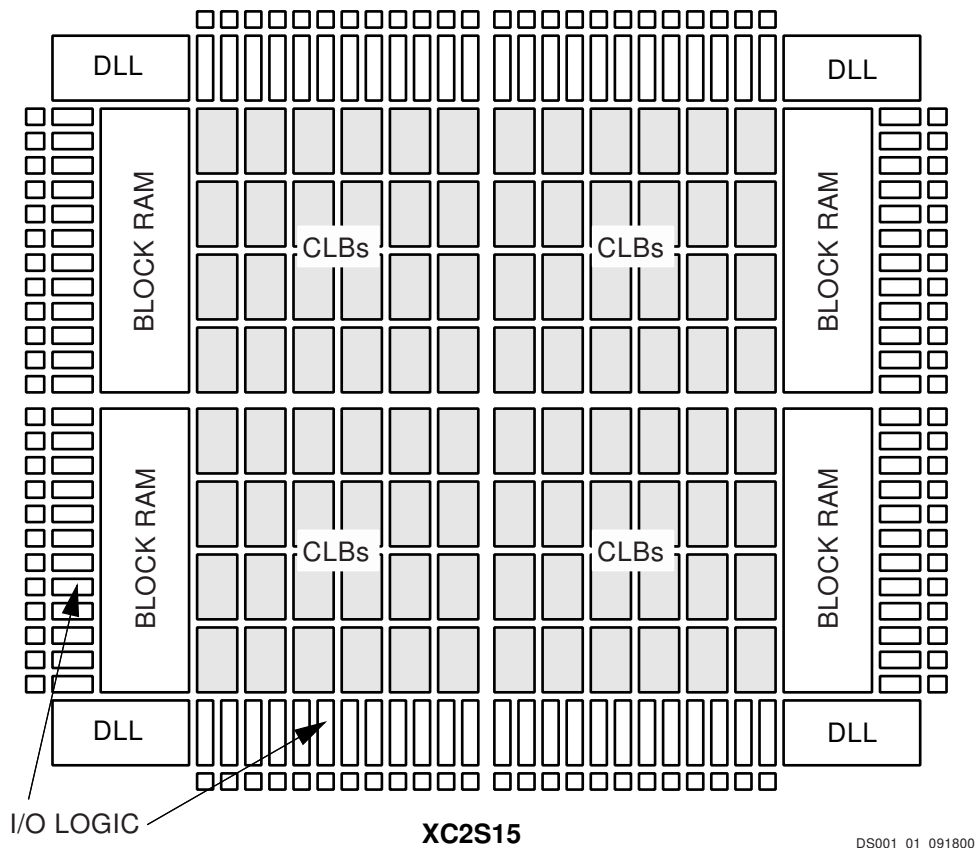


Figure 1: Basic Spartan-II Family FPGA Block Diagram

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

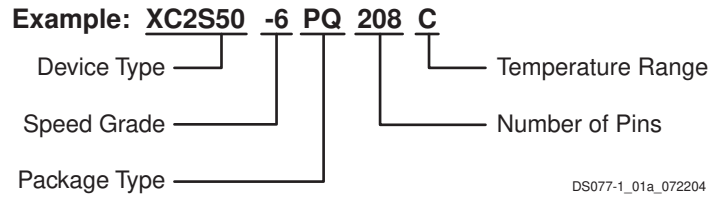
Notes:

1. All user I/O counts do not include the four global clock/user input pins.
2. Discontinued by [PDN2004-01](#).

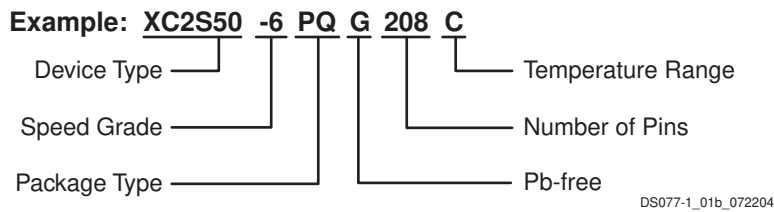
Ordering Information

Spartan-II devices are available in both standard and Pb-free packaging options for all device/package combinations. The Pb-free packages include a special "G" character in the ordering code.

Standard Packaging



Pb-Free Packaging



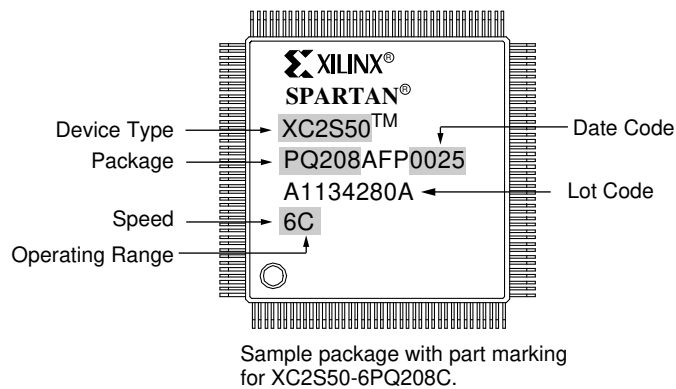
Device Ordering Options

Device	Speed Grade		Number of Pins / Package Type		Temperature Range (T _J)	
XC2S15	-5	Standard Performance	VQ(G)100	100-pin Plastic Very Thin QFP	C = Commercial	0°C to +85°C
XC2S30	-6	Higher Performance ⁽¹⁾	CS(G)144	144-ball Chip-Scale BGA	I = Industrial	-40°C to +100°C
XC2S50			TQ(G)144	144-pin Plastic Thin QFP		
XC2S100			PQ(G)208	208-pin Plastic QFP		
XC2S150			FG(G)256	256-ball Fine Pitch BGA		
XC2S200			FG(G)456	456-ball Fine Pitch BGA		

Notes:

- The -6 speed grade is exclusively available in the Commercial temperature range.

Device Part Marking



Revision History

Date	Version No.	Description
09/18/00	2.0	Sectioned the Spartan-II Family data sheet into four modules. Added industrial temperature range information.
10/31/00	2.1	Removed Power down feature.
03/05/01	2.2	Added statement on PROMs.
11/01/01	2.3	Updated Product Availability chart. Minor text edits.
09/03/03	2.4	Added device part marking.
08/02/04	2.5	Added information on Pb-free packaging options and removed discontinued options.
06/13/08	2.8	Updated description and links. Updated all modules for continuous page, figure, and table numbering. Synchronized all modules to v2.8.

Architectural Description

Spartan-II FPGA Array

The Spartan®-II field-programmable gate array, shown in [Figure 2](#), is composed of five major configurable elements:

- IOBs provide the interface between the package pins and the internal logic
- CLBs provide the functional elements for constructing most logic
- Dedicated block RAM memories of 4096 bits each
- Clock DLLs for clock-distribution delay compensation and clock domain control
- Versatile multi-level interconnect structure

As can be seen in [Figure 2](#), the CLBs form the central logic structure with easy access to all support and routing structures. The IOBs are located around all the logic and

memory elements for easy and quick routing of signals on and off the chip.

Values stored in static memory cells control all the configurable logic elements and interconnect resources. These values load into the memory cells on power-up, and can reload if necessary to change the function of the device.

Each of these elements will be discussed in detail in the following sections.

Input/Output Block

The Spartan-II FPGA IOB, as seen in [Figure 2](#), features inputs and outputs that support a wide variety of I/O signaling standards. These high-speed inputs and outputs are capable of supporting various state of the art memory and bus interfaces. [Table 3](#) lists several of the standards which are supported along with the required reference, output and termination voltages needed to meet the standard.

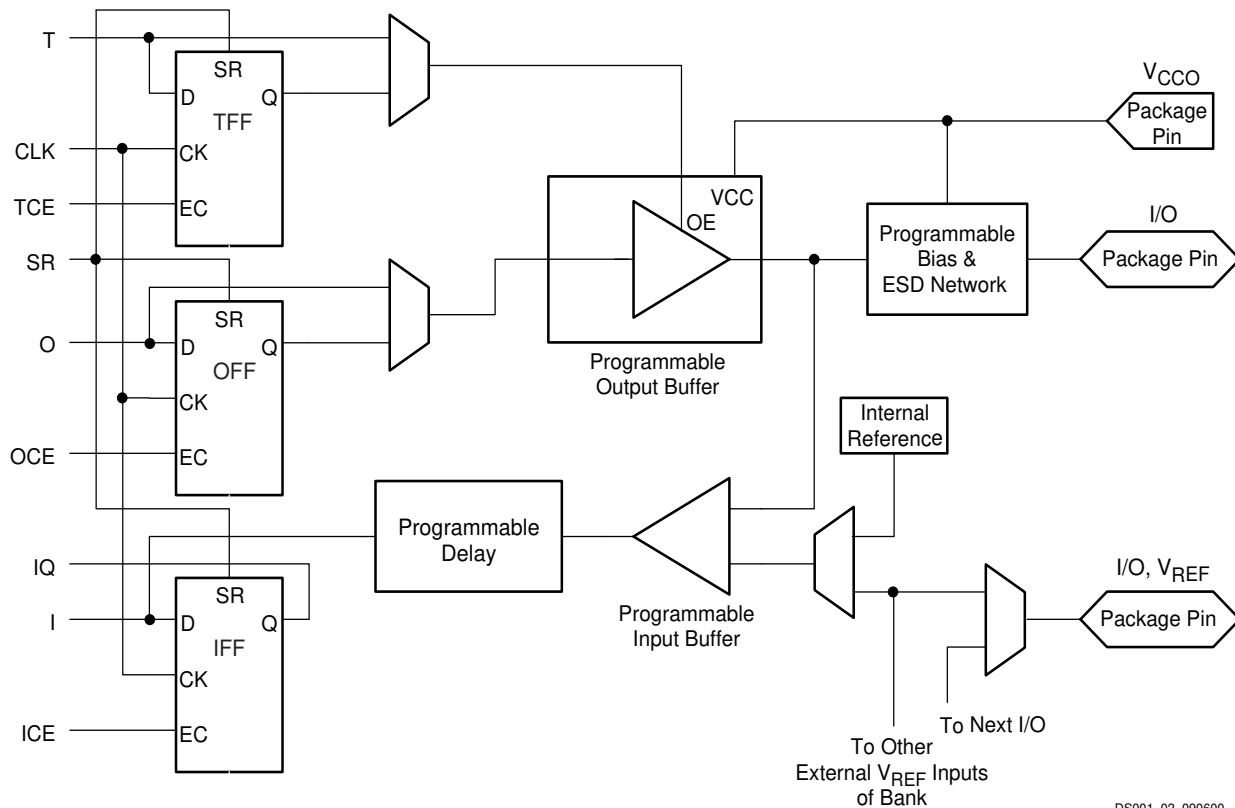


Figure 2: Spartan-II FPGA Input/Output Block (IOB)

DS001_02_090600

The three IOB registers function either as edge-triggered D-type flip-flops or as level-sensitive latches. Each IOB has a clock signal (CLK) shared by the three registers and independent Clock Enable (CE) signals for each register. In addition to the CLK and CE control signals, the three registers share a Set/Reset (SR). For each register, this signal can be independently configured as a synchronous Set, a synchronous Reset, an asynchronous Preset, or an asynchronous Clear.

A feature not shown in the block diagram, but controlled by the software, is polarity control. The input and output buffers and all of the IOB control signals have independent polarity controls.

Optional pull-up and pull-down resistors and an optional weak-keeper circuit are attached to each pad. Prior to configuration all outputs not involved in configuration are forced into their high-impedance state. The pull-down resistors and the weak-keeper circuits are inactive, but inputs may optionally be pulled up.

Table 3: Standards Supported by I/O (Typical Values)

I/O Standard	Input Reference Voltage (V_{REF})	Output Source Voltage (V_{CCO})	Board Termination Voltage (V_{TT})
LVTTTL (2-24 mA)	N/A	3.3	N/A
LVC MOS2	N/A	2.5	N/A
PCI (3V/5V, 33 MHz/66 MHz)	N/A	3.3	N/A
GTL	0.8	N/A	1.2
GTL+	1.0	N/A	1.5
HSTL Class I	0.75	1.5	0.75
HSTL Class III	0.9	1.5	1.5
HSTL Class IV	0.9	1.5	1.5
SSTL3 Class I and II	1.5	3.3	1.5
SSTL2 Class I and II	1.25	2.5	1.25
CTT	1.5	3.3	1.5
AGP-2X	1.32	3.3	N/A

The activation of pull-up resistors prior to configuration is controlled on a global basis by the configuration mode pins. If the pull-up resistors are not activated, all the pins will float. Consequently, external pull-up or pull-down resistors must be provided on pins required to be at a well-defined logic level prior to configuration.

All pads are protected against damage from electrostatic discharge (ESD) and from over-voltage transients. Two forms of over-voltage protection are provided, one that permits 5V compliance, and one that does not. For 5V compliance, a zener-like structure connected to ground turns on when the output rises to approximately 6.5V. When 5V compliance is not required, a conventional clamp diode may be connected to the output supply voltage, V_{CCO} . The type of over-voltage protection can be selected independently for each pad.

All Spartan-II FPGA IOBs support IEEE 1149.1-compatible boundary scan testing.

Input Path

A buffer in the Spartan-II FPGA IOB input path routes the input signal either directly to internal logic or through an optional input flip-flop.

An optional delay element at the D-input of this flip-flop eliminates pad-to-pad hold time. The delay is matched to the internal clock-distribution delay of the FPGA, and when used, assures that the pad-to-pad hold time is zero.

Each input buffer can be configured to conform to any of the low-voltage signaling standards supported. In some of these standards the input buffer utilizes a user-supplied threshold voltage, V_{REF} . The need to supply V_{REF} imposes constraints on which standards can be used in close proximity to each other. See "[I/O Banking](#)," page 9.

There are optional pull-up and pull-down resistors at each input for use after configuration.

Output Path

The output path includes a 3-state output buffer that drives the output signal onto the pad. The output signal can be routed to the buffer directly from the internal logic or through an optional IOB output flip-flop.

The 3-state control of the output can also be routed directly from the internal logic or through a flip-flop that provides synchronous enable and disable.

Each output driver can be individually programmed for a wide range of low-voltage signaling standards. Each output buffer can source up to 24 mA and sink up to 48 mA. Drive strength and slew rate controls minimize bus transients.

In most signaling standards, the output high voltage depends on an externally supplied V_{CCO} voltage. The need to supply V_{CCO} imposes constraints on which standards can be used in close proximity to each other. See "[I/O Banking](#)".

An optional weak-keeper circuit is connected to each output. When selected, the circuit monitors the voltage on the pad and weakly drives the pin High or Low to match the input signal. If the pin is connected to a multiple-source signal, the weak keeper holds the signal in its last state if all

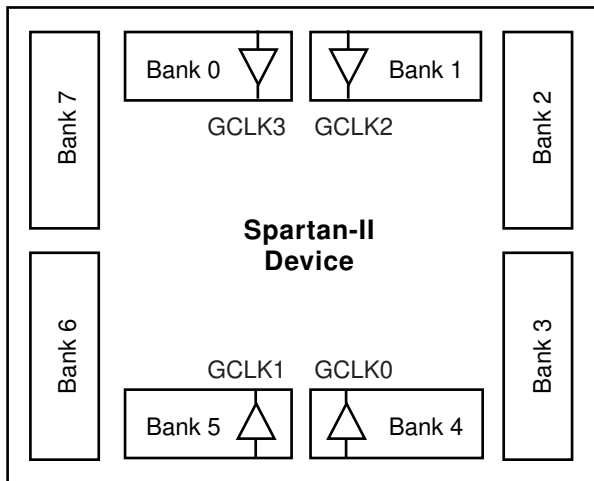
drivers are disabled. Maintaining a valid logic level in this way helps eliminate bus chatter.

Because the weak-keeper circuit uses the IOB input buffer to monitor the input level, an appropriate V_{REF} voltage must be provided if the signaling standard requires one. The provision of this voltage must comply with the I/O banking rules.

I/O Banking

Some of the I/O standards described above require V_{CCO} and/or V_{REF} voltages. These voltages are externally connected to device pins that serve groups of IOBs, called banks. Consequently, restrictions exist about which I/O standards can be combined within a given bank.

Eight I/O banks result from separating each edge of the FPGA into two banks (see Figure 3). Each bank has multiple V_{CCO} pins which must be connected to the same voltage. Voltage is determined by the output standards in use.



DS001_03_060100

Figure 3: Spartan-II I/O Banks

Within a bank, output standards may be mixed only if they use the same V_{CCO} . Compatible standards are shown in Table 4. GTL and GTL+ appear under all voltages because their open-drain outputs do not depend on V_{CCO} .

Table 4: Compatible Output Standards

V _{CCO}	Compatible Standards
3.3V	PCI, LVTTTL, SSTL3 I, SSTL3 II, CTT, AGP, GTL, GTL+
2.5V	SSTL2 I, SSTL2 II, LVCMOS2, GTL, GTL+
1.5V	HSTL I, HSTL III, HSTL IV, GTL, GTL+

Some input standards require a user-supplied threshold voltage, V_{REF} . In this case, certain user-I/O pins are

automatically configured as inputs for the V_{REF} voltage. About one in six of the I/O pins in the bank assume this role.

V_{REF} pins within a bank are interconnected internally and consequently only one V_{REF} voltage can be used within each bank. All V_{REF} pins in the bank, however, must be connected to the external voltage source for correct operation.

In a bank, inputs requiring V_{REF} can be mixed with those that do not but only one V_{REF} voltage may be used within a bank. Input buffers that use V_{REF} are not 5V tolerant. LVTTTL, LVCMOS2, and PCI are 5V tolerant. The V_{CCO} and V_{REF} pins for each bank appear in the device pinout tables.

Within a given package, the number of V_{REF} and V_{CCO} pins can vary depending on the size of device. In larger devices, more I/O pins convert to V_{REF} pins. Since these are always a superset of the V_{REF} pins used for smaller devices, it is possible to design a PCB that permits migration to a larger device. All V_{REF} pins for the largest device anticipated must be connected to the V_{REF} voltage, and not used for I/O.

Independent Banks Available

Package	VQ100 PQ208	CS144 TQ144	FG256 FG456
Independent Banks	1	4	8

Configurable Logic Block

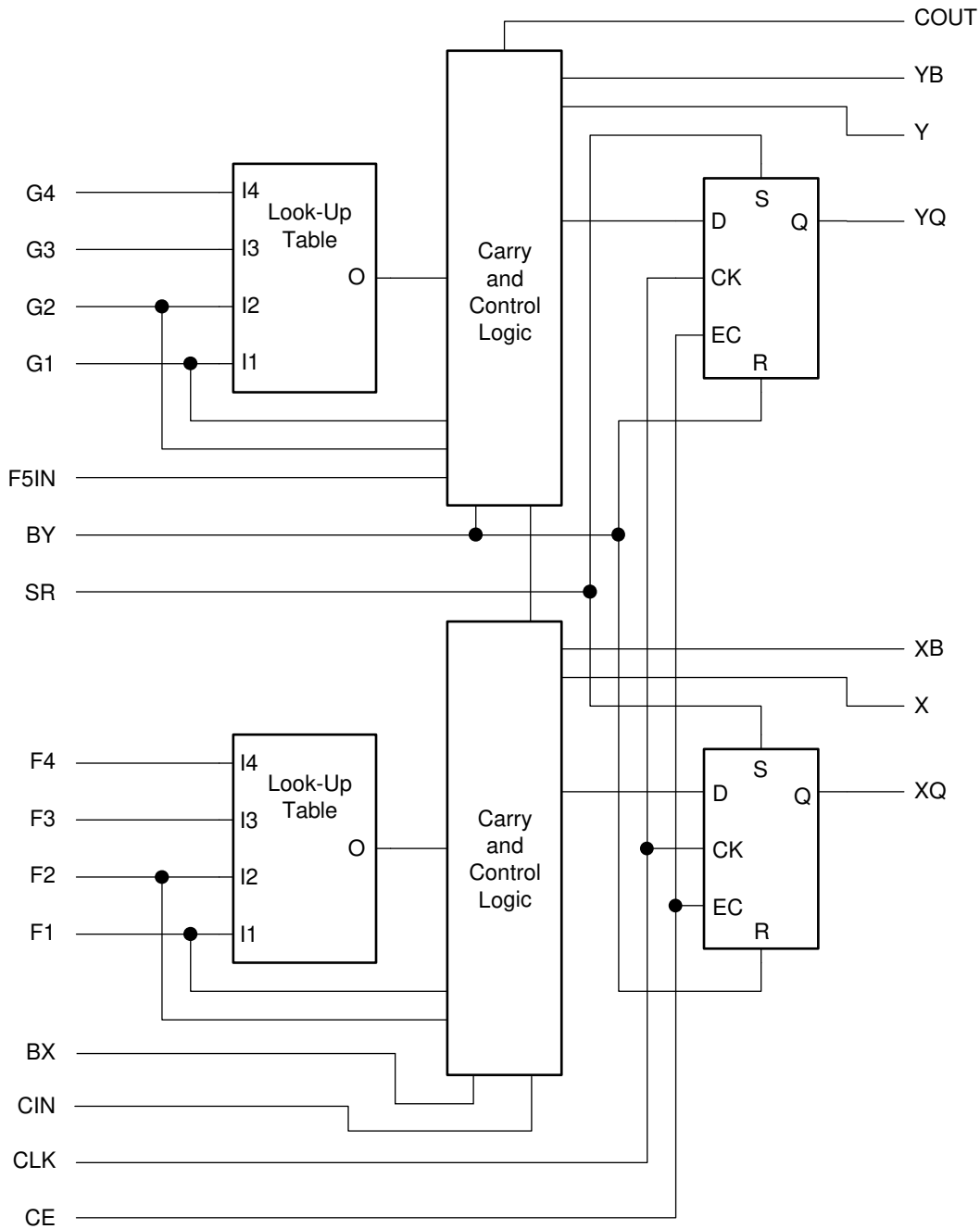
The basic building block of the Spartan-II FPGA CLB is the logic cell (LC). An LC includes a 4-input function generator, carry logic, and storage element. Output from the function generator in each LC drives the CLB output and the D input of the flip-flop. Each Spartan-II FPGA CLB contains four LCs, organized in two similar slices; a single slice is shown in Figure 4.

In addition to the four basic LCs, the Spartan-II FPGA CLB contains logic that combines function generators to provide functions of five or six inputs.

Look-Up Tables

Spartan-II FPGA function generators are implemented as 4-input look-up tables (LUTs). In addition to operating as a function generator, each LUT can provide a 16 x 1-bit synchronous RAM. Furthermore, the two LUTs within a slice can be combined to create a 16 x 2-bit or 32 x 1-bit synchronous RAM, or a 16 x 1-bit dual-port synchronous RAM.

The Spartan-II FPGA LUT can also provide a 16-bit shift register that is ideal for capturing high-speed or burst-mode data. This mode can also be used to store data in applications such as Digital Signal Processing.



DS001_04_091400

Figure 4: Spartan-II CLB Slice (two identical slices in each CLB)

Storage Elements

Storage elements in the Spartan-II FPGA slice can be configured either as edge-triggered D-type flip-flops or as level-sensitive latches. The D inputs can be driven either by function generators within the slice or directly from slice inputs, bypassing the function generators.

In addition to Clock and Clock Enable signals, each slice has synchronous set and reset signals (SR and BY). SR forces a storage element into the initialization state specified for it in the configuration. BY forces it into the

opposite state. Alternatively, these signals may be configured to operate asynchronously.

All control signals are independently invertible, and are shared by the two flip-flops within the slice.

Additional Logic

The F5 multiplexer in each slice combines the function generator outputs. This combination provides either a function generator that can implement any 5-input function, a 4:1 multiplexer, or selected functions of up to nine inputs.

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

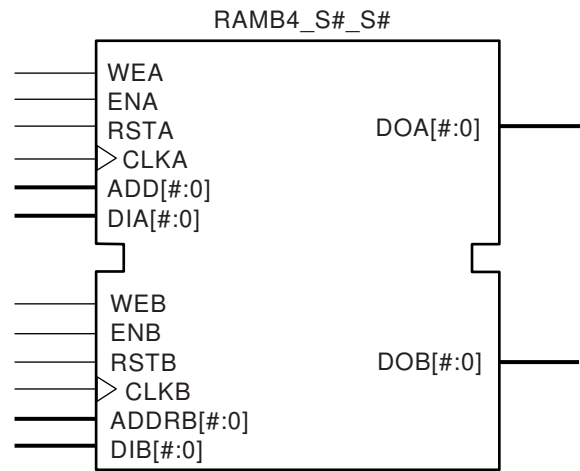
Spartan-II FPGAs incorporate several large block RAM memories. These complement the distributed RAM Look-Up Tables (LUTs) that provide shallow memory structures implemented in CLBs.

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.



DS001_05_060100

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

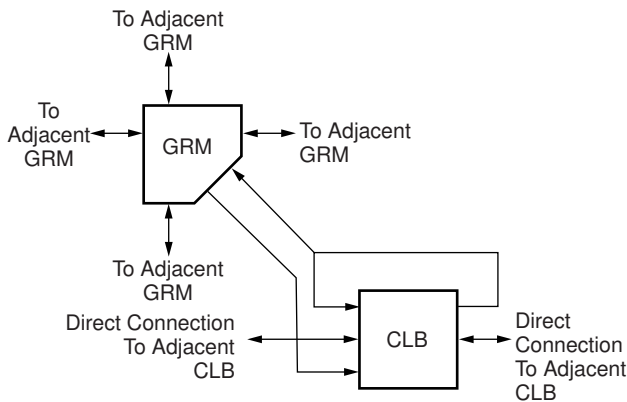
It is the longest delay path that limits the speed of any worst-case design. Consequently, the Spartan-II routing architecture and its place-and-route software were defined in a single optimization process. This joint optimization minimizes long-path delays, and consequently, yields the best system performance.

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.

Local Routing

The local routing resources, as shown in Figure 6, provide the following three types of connections:

- Interconnections among the LUTs, flip-flops, and General Routing Matrix (GRM)
- Internal CLB feedback paths that provide high-speed connections to LUTs within the same CLB, chaining them together with minimal routing delay
- Direct paths that provide high-speed connections between horizontally adjacent CLBs, eliminating the delay of the GRM



DS001_06_032300

Figure 6: Spartan-II Local Routing

General Purpose Routing

Most Spartan-II FPGA signals are routed on the general purpose routing, and consequently, the majority of interconnect resources are associated with this level of the routing hierarchy. The general routing resources are located in horizontal and vertical routing channels associated with the rows and columns CLBs. The general-purpose routing resources are listed below.

- Adjacent to each CLB is a General Routing Matrix (GRM). The GRM is the switch matrix through which horizontal and vertical routing resources connect, and is also the means by which the CLB gains access to the general purpose routing.
- 24 single-length lines route GRM signals to adjacent GRMs in each of the four directions.
- 96 buffered Hex lines route GRM signals to other GRMs six blocks away in each one of the four directions. Organized in a staggered pattern, Hex lines may be driven only at their endpoints. Hex-line signals can be accessed either at the endpoints or at the midpoint (three blocks from the source). One third of the Hex lines are bidirectional, while the remaining ones are unidirectional.
- 12 Longlines are buffered, bidirectional wires that distribute signals across the device quickly and

efficiently. Vertical Longlines span the full height of the device, and horizontal ones span the full width of the device.

I/O Routing

Spartan-II devices have additional routing resources around their periphery that form an interface between the CLB array and the IOBs. This additional routing, called the VersaRing, facilitates pin-swapping and pin-locking, such that logic redesigns can adapt to existing PCB layouts. Time-to-market is reduced, since PCBs and other system components can be manufactured while the logic design is still in progress.

Dedicated Routing

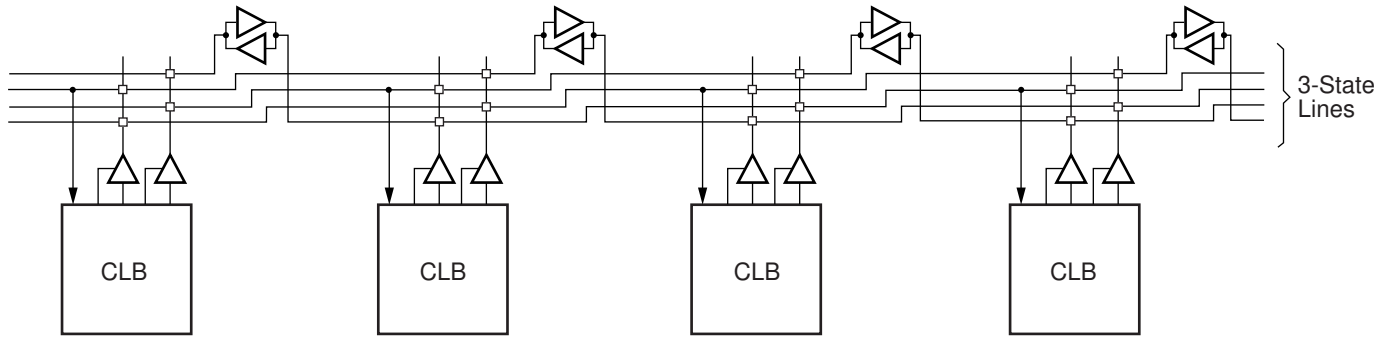
Some classes of signal require dedicated routing resources to maximize performance. In the Spartan-II architecture, dedicated routing resources are provided for two classes of signal.

- Horizontal routing resources are provided for on-chip 3-state busses. Four partitionable bus lines are provided per CLB row, permitting multiple busses within a row, as shown in Figure 7.
- Two dedicated nets per CLB propagate carry signals vertically to the adjacent CLB.

Global Routing

Global Routing resources distribute clocks and other signals with very high fanout throughout the device. Spartan-II devices include two tiers of global routing resources referred to as primary and secondary global routing resources.

- The primary global routing resources are four dedicated global nets with dedicated input pins that are designed to distribute high-fanout clock signals with minimal skew. Each global clock net can drive all CLB, IOB, and block RAM clock pins. The primary global nets may only be driven by global buffers. There are four global buffers, one for each global net.
- The secondary global routing resources consist of 24 backbone lines, 12 across the top of the chip and 12 across bottom. From these lines, up to 12 unique signals per column can be distributed via the 12 longlines in the column. These secondary resources are more flexible than the primary resources since they are not restricted to routing only to clock pins.



DS001_07_090600

Figure 7: BUFT Connections to Dedicated Horizontal Bus Lines

Clock Distribution

The Spartan-II family provides high-speed, low-skew clock distribution through the primary global routing resources described above. A typical clock distribution net is shown in Figure 8.

Four global buffers are provided, two at the top center of the device and two at the bottom center. These drive the four primary global nets that in turn drive any clock pin.

Four dedicated clock pads are provided, one adjacent to each of the global buffers. The input to the global buffer is selected either from these pads or from signals in the general purpose routing. Global clock pins do not have the option for internal, weak pull-up resistors.

networks. The DLL monitors the input clock and the distributed clock, and automatically adjusts a clock delay element. Additional delay is introduced such that clock edges reach internal flip-flops exactly one clock period after they arrive at the input. This closed-loop system effectively eliminates clock-distribution delay by ensuring that clock edges arrive at internal flip-flops in synchronism with clock edges arriving at the input.

In addition to eliminating clock-distribution delay, the DLL provides advanced control of multiple clock domains. The DLL provides four quadrature phases of the source clock, can double the clock, or divide the clock by 1.5, 2, 2.5, 3, 4, 5, 8, or 16. It has six outputs.

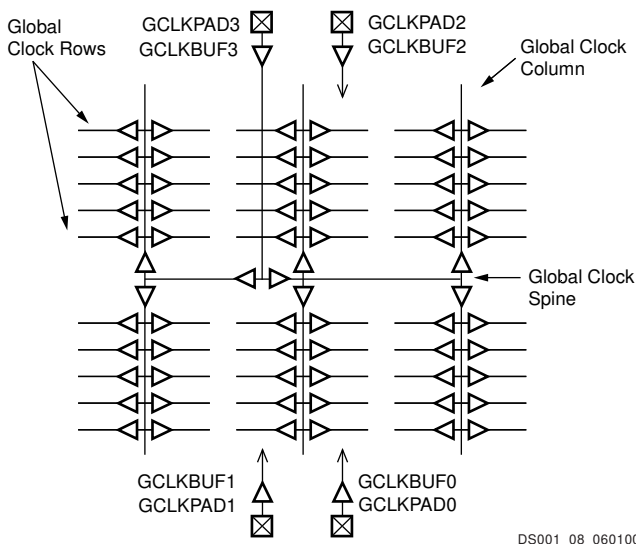
The DLL also operates as a clock mirror. By driving the output from a DLL off-chip and then back on again, the DLL can be used to deskew a board level clock among multiple Spartan-II devices.

In order to guarantee that the system clock is operating correctly prior to the FPGA starting up after configuration, the DLL can delay the completion of the configuration process until after it has achieved lock.

Boundary Scan

Spartan-II devices support all the mandatory boundary-scan instructions specified in the IEEE standard 1149.1. A Test Access Port (TAP) and registers are provided that implement the EXTEST, SAMPLE/PRELOAD, and BYPASS instructions. The TAP also supports two USERCODE instructions and internal scan chains.

The TAP uses dedicated package pins that always operate using LVTTTL. For TDO to operate using LVTTTL, the V_{CCO} for Bank 2 must be 3.3V. Otherwise, TDO switches rail-to-rail between ground and V_{CCO}. TDI, TMS, and TCK have a default internal weak pull-up resistor, and TDO has no default resistor. Bitstream options allow setting any of the four TAP pins to have an internal pull-up, pull-down, or neither.



DS001_08_060100

Figure 8: Global Clock Distribution Network

Delay-Locked Loop (DLL)

Associated with each global clock input buffer is a fully digital Delay-Locked Loop (DLL) that can eliminate skew between the clock input pad and internal clock-input pins throughout the device. Each DLL can drive two global clock

Boundary-scan operation is independent of individual IOB configurations, and unaffected by package type. All IOBs, including unbonded ones, are treated as independent 3-state bidirectional pins in a single scan chain. Retention of the bidirectional test capability after configuration facilitates the testing of external interconnections.

Table 7 lists the boundary-scan instructions supported in Spartan-II FPGAs. Internal signals can be captured during EXTEST by connecting them to unbonded or unused IOBs. They may also be connected to the unused outputs of IOBs defined as unidirectional input pins.

Table 7: Boundary-Scan Instructions

Boundary-Scan Command	Binary Code[4:0]	Description
EXTEST	00000	Enables boundary-scan EXTEST operation
SAMPLE	00001	Enables boundary-scan SAMPLE operation
USR1	00010	Access user-defined register 1
USR2	00011	Access user-defined register 2
CFG_OUT	00100	Access the configuration bus for Readback
CFG_IN	00101	Access the configuration bus for Configuration
INTEST	00111	Enables boundary-scan INTEST operation
USRCODE	01000	Enables shifting out USER code
IDCODE	01001	Enables shifting out of ID Code
HIZ	01010	Disables output pins while enabling the Bypass Register
JSTART	01100	Clock the start-up sequence when StartupClk is TCK
BYPASS	11111	Enables BYPASS
RESERVED	All other codes	Xilinx® reserved instructions

The public boundary-scan instructions are available prior to configuration. After configuration, the public instructions remain available together with any USERCODE instructions installed during the configuration. While the SAMPLE and BYPASS instructions are available during configuration, it is recommended that boundary-scan operations not be performed during this transitional period.

In addition to the test instructions outlined above, the boundary-scan circuitry can be used to configure the FPGA, and also to read back the configuration data.

To facilitate internal scan chains, the User Register provides three outputs (Reset, Update, and Shift) that represent the corresponding states in the boundary-scan internal state machine.

Figure 9 is a diagram of the Spartan-II family boundary scan logic. It includes three bits of Data Register per IOB, the IEEE 1149.1 Test Access Port controller, and the Instruction Register with decodes.

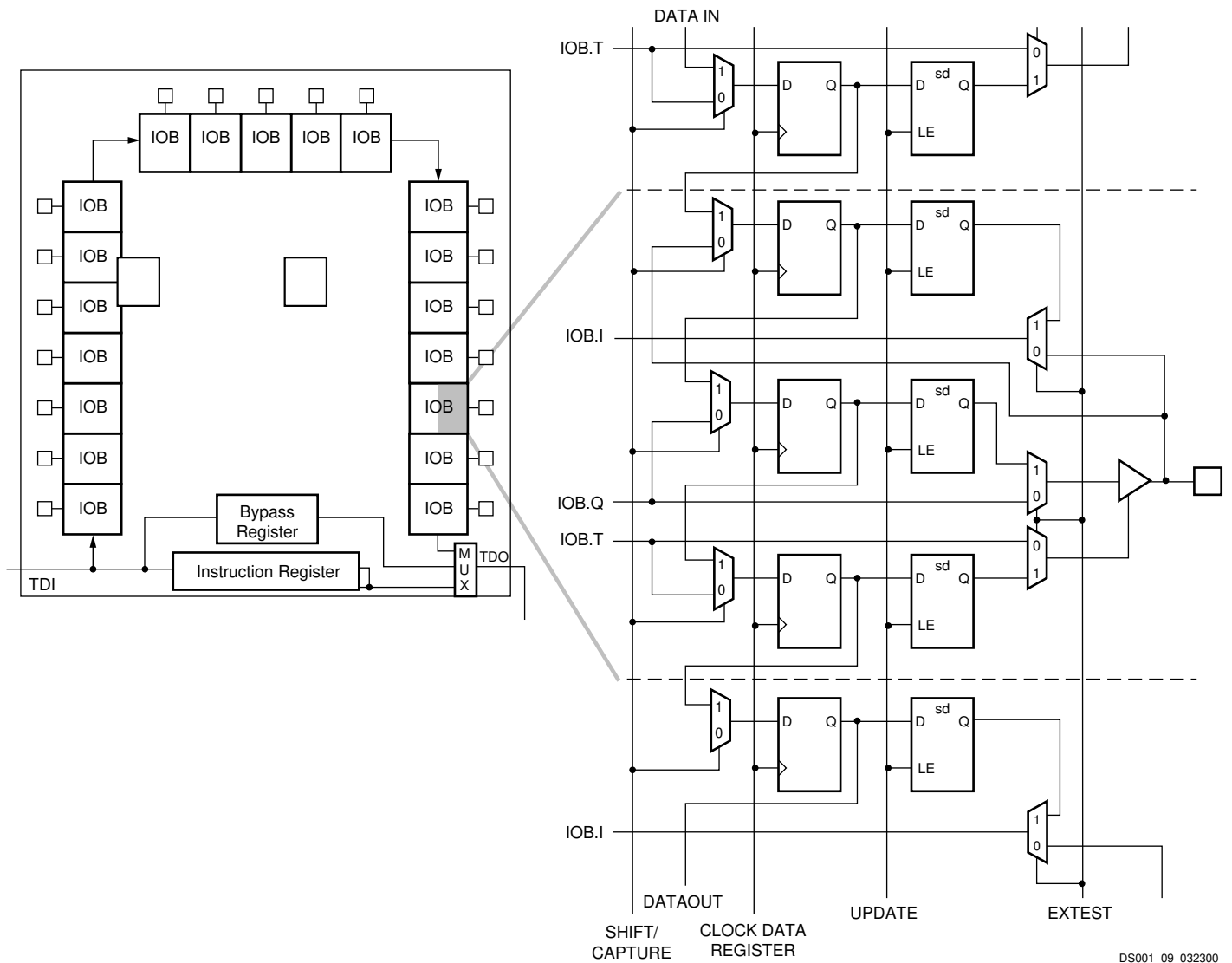


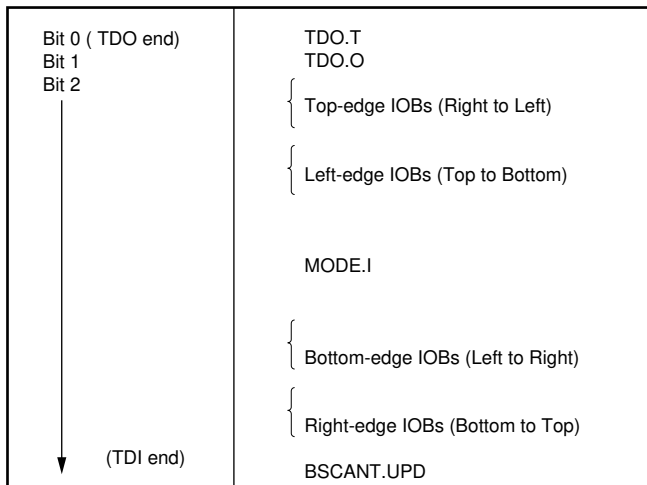
Figure 9: Spartan-II Family Boundary Scan Logic

Bit Sequence

The bit sequence within each IOB is: In, Out, 3-State. The input-only pins contribute only the In bit to the boundary scan I/O data register, while the output-only pins contributes all three bits.

From a cavity-up view of the chip (as shown in the FPGA Editor), starting in the upper right chip corner, the boundary scan data-register bits are ordered as shown in Figure 10.

BSDL (Boundary Scan Description Language) files for Spartan-II family devices are available on the Xilinx website, in the Downloads area.



DS001_10_032300

Figure 10: Boundary Scan Bit Sequence

Development System

Spartan-II FPGAs are supported by the Xilinx ISE® development tools. The basic methodology for Spartan-II FPGA design consists of three interrelated steps: design entry, implementation, and verification. Industry-standard tools are used for design entry and simulation, while Xilinx provides proprietary architecture-specific tools for implementation.

The Xilinx development system is integrated under a single graphical interface, providing designers with a common user interface regardless of their choice of entry and verification tools. The software simplifies the selection of implementation options with pull-down menus and on-line help.

For HDL design entry, the Xilinx FPGA development system provides interfaces to several synthesis design environments.

A standard interface-file specification, Electronic Design Interchange Format (EDIF), simplifies file transfers into and out of the development system.

Spartan-II FPGAs supported by a unified library of standard functions. This library contains over 400 primitives and macros, ranging from 2-input AND gates to 16-bit accumulators, and includes arithmetic functions, comparators, counters, data registers, decoders, encoders, I/O functions, latches, Boolean functions, multiplexers, shift registers, and barrel shifters.

The design environment supports hierarchical design entry. These hierarchical design elements are automatically combined by the implementation tools. Different design entry tools can be combined within a hierarchical design, thus allowing the most convenient entry method to be used for each portion of the design.

Design Implementation

The place-and-route tools (PAR) automatically provide the implementation flow described in this section. The partitioner takes the EDIF netlist for the design and maps the logic into the architectural resources of the FPGA (CLBs and IOBs, for example). The placer then determines the best locations for these blocks based on their interconnections and the desired performance. Finally, the router interconnects the blocks.

The PAR algorithms support fully automatic implementation of most designs. For demanding applications, however, the user can exercise various degrees of control over the process. User partitioning, placement, and routing information is optionally specified during the design-entry process. The implementation of highly structured designs can benefit greatly from basic floorplanning.

The implementation software incorporates timing-driven placement and routing. Designers specify timing requirements along entire paths during design entry. The timing path analysis routines in PAR then recognize these user-specified requirements and accommodate them.

Timing requirements are entered in a form directly relating to the system requirements, such as the targeted clock frequency, or the maximum allowable delay between two registers. In this way, the overall performance of the system along entire signal paths is automatically tailored to user-generated specifications. Specific timing information for individual nets is unnecessary.

Design Verification

In addition to conventional software simulation, FPGA users can use in-circuit debugging techniques. Because Xilinx devices are infinitely reprogrammable, designs can be verified in real time without the need for extensive sets of software simulation vectors.

The development system supports both software simulation and in-circuit debugging techniques. For simulation, the system extracts the post-layout timing information from the design database, and back-annotates this information into the netlist for use by the simulator. Alternatively, the user can verify timing-critical portions of the design using the static timing analyzer.

For in-circuit debugging, the development system includes a download cable, which connects the FPGA in the target system to a PC or workstation. After downloading the design into the FPGA, the designer can read back the contents of the flip-flops, and so observe the internal logic state. Simple modifications can be downloaded into the system in a matter of minutes.

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.

Signals

There are two kinds of pins that are used to configure Spartan-II devices: Dedicated pins perform only specific configuration-related functions; the other pins can serve as general purpose I/Os once user operation has begun.

The dedicated pins comprise the mode pins (M2, M1, M0), the configuration clock pin (CCLK), the $\overline{\text{PROGRAM}}$ pin, the DONE pin and the boundary-scan pins (TDI, TDO, TMS, TCK). Depending on the selected configuration mode, CCLK may be an output generated by the FPGA, or may be generated externally, and provided to the FPGA as an input.

Note that some configuration pins can act as outputs. For correct operation, these pins require a V_{CCO} of 3.3V to drive an LVTTTL signal or 2.5V to drive an LVC MOS signal. All the relevant pins fall in banks 2 or 3. The $\overline{\text{CS}}$ and $\overline{\text{WRITE}}$ pins for Slave Parallel mode are located in bank 1.

For a more detailed description than that given below, see "Pinout Tables" in Module 4 and [XAPP176](#), *Spartan-II FPGA Series Configuration and Readback*.

The Process

The sequence of steps necessary to configure Spartan-II devices are shown in Figure 11. The overall flow can be divided into three different phases.

- Initiating Configuration
- Configuration memory clear
- Loading data frames
- Start-up

The memory clearing and start-up phases are the same for all configuration modes; however, the steps for the loading of data frames are different. Thus, the details for data frame loading are described separately in the sections devoted to each mode.

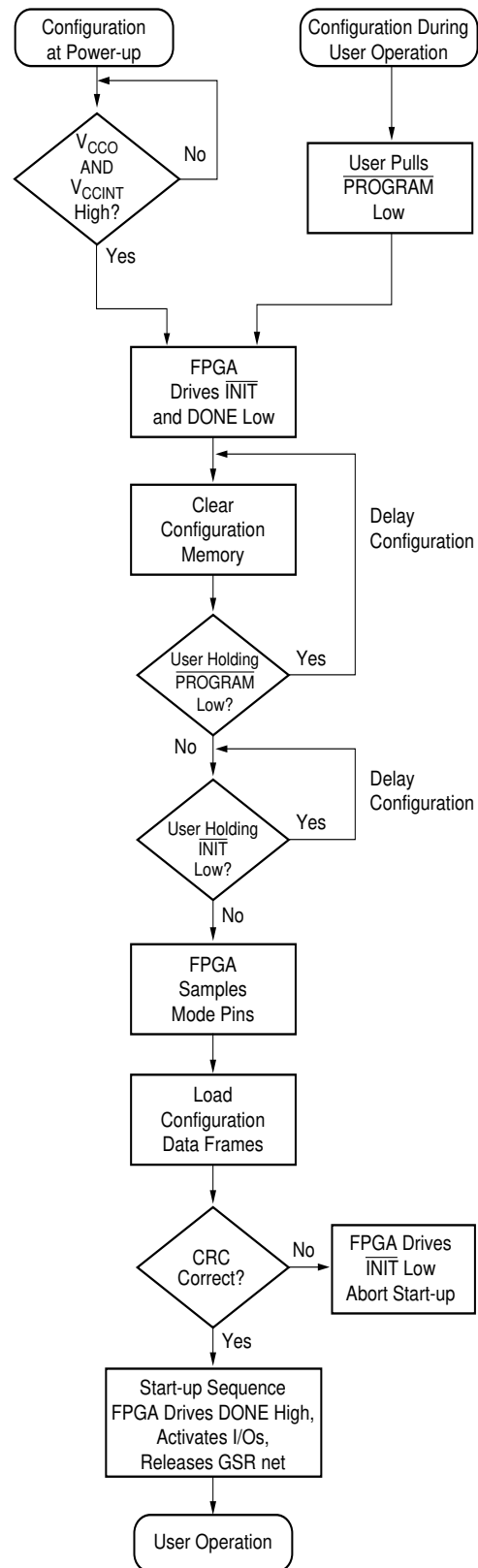
Initiating Configuration

There are two different ways to initiate the configuration process: applying power to the device or asserting the $\overline{\text{PROGRAM}}$ input.

Configuration on power-up occurs automatically unless it is delayed by the user, as described in a separate section below. The waveform for configuration on power-up is shown in Figure 12, page 19. Before configuration can begin, V_{CCO} Bank 2 must be greater than 1.0V. Furthermore, all V_{CCINT} power pins must be connected to a 2.5V supply. For more information on delaying configuration, see "Clearing Configuration Memory," page 19.

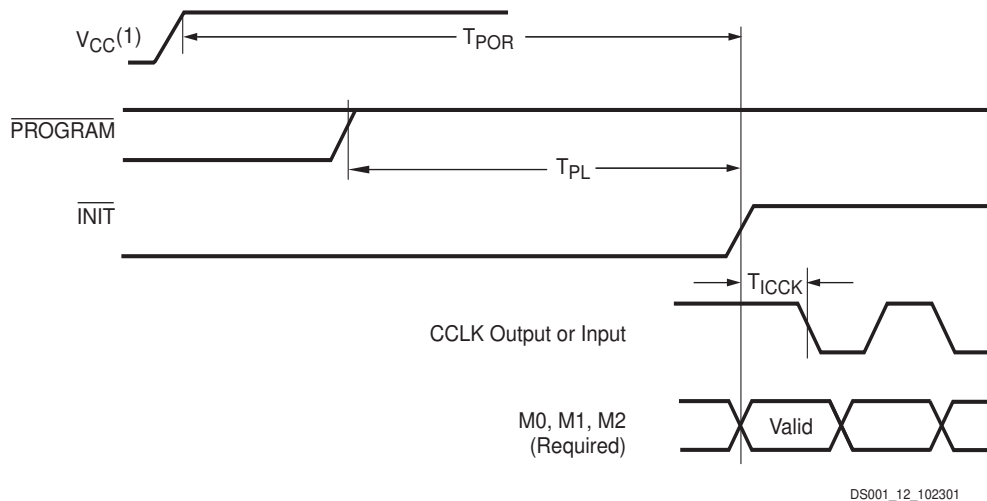
Once in user operation, the device can be re-configured simply by pulling the $\overline{\text{PROGRAM}}$ pin Low. The device acknowledges the beginning of the configuration process

by driving DONE Low, then enters the memory-clearing phase.



DS001_11_111501

Figure 11: Configuration Flow Diagram



DS001_12_102301

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.

By default, these operations are synchronized to CCLK. The entire start-up sequence lasts eight cycles, called C0-C7, after which the loaded design is fully functional. The default timing for start-up is shown in the top half of Figure 13. The four operations can be selected to switch on any CCLK cycle C1-C6 through settings in the Xilinx software. Heavy lines show default settings.

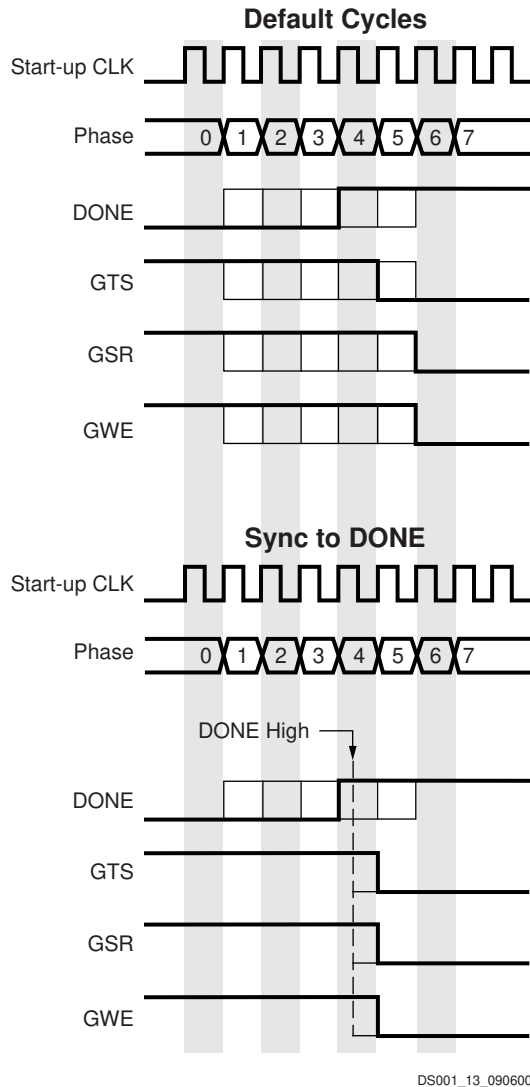


Figure 13: Start-Up Waveforms

The bottom half of Figure 13 shows another commonly used version of the start-up timing known as Sync-to-DONE. This version makes the GTS, GSR, and GWE events conditional upon the DONE pin going High. This timing is important for a daisy chain of multiple FPGAs in serial mode, since it ensures that all FPGAs go through start-up together, after all their DONE pins have gone High.

Sync-to-DONE timing is selected by setting the GTS, GSR, and GWE cycles to a value of DONE in the configuration options. This causes these signals to transition one clock cycle after DONE externally transitions High.

Serial Modes

There are two serial configuration modes: In Master Serial mode, the FPGA controls the configuration process by driving CCLK as an output. In Slave Serial mode, the FPGA passively receives CCLK as an input from an external agent (e.g., a microprocessor, CPLD, or second FPGA in master mode) that is controlling the configuration process. In both modes, the FPGA is configured by loading one bit per CCLK cycle. The MSB of each configuration data byte is always written to the DIN pin first.

See Figure 14 for the sequence for loading data into the Spartan-II FPGA serially. This is an expansion of the "Load Configuration Data Frames" block in Figure 11. Note that CS and WRITE normally are not used during serial configuration. To ensure successful loading of the FPGA, do not toggle WRITE with CS Low during serial configuration.

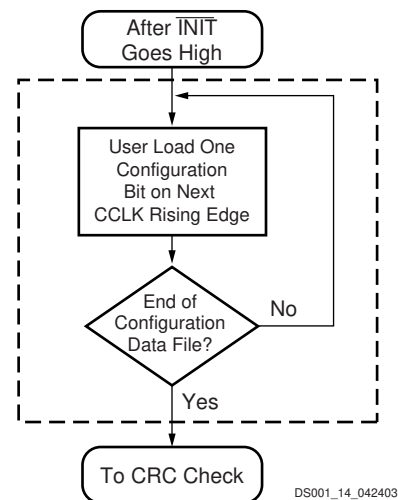


Figure 14: Loading Serial Mode Configuration Data

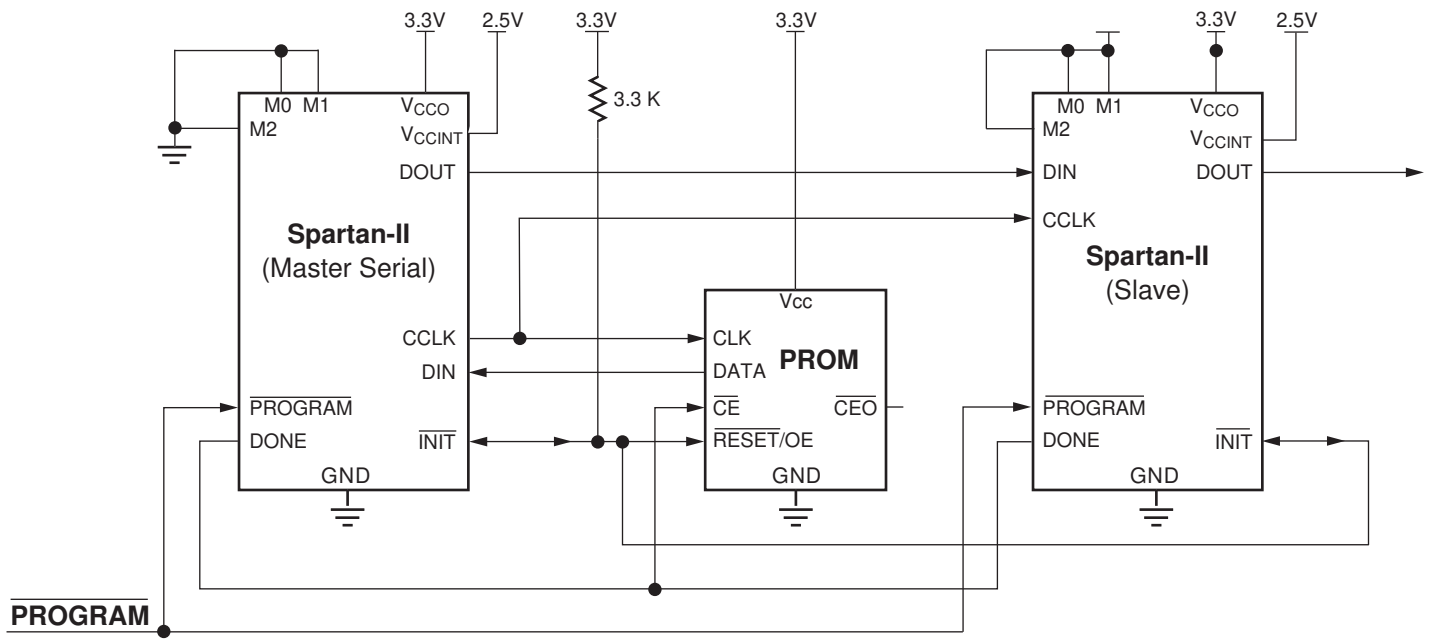
Slave Serial Mode

In Slave Serial mode, the FPGA's CCLK pin is driven by an external source, allowing FPGAs to be configured from other logic devices such as microprocessors or in a daisy-chain configuration. Figure 15 shows connections for a Master Serial FPGA configuring a Slave Serial FPGA from a PROM. A Spartan-II device in slave serial mode should be connected as shown for the third device from the left. Slave Serial mode is selected by a <11x> on the mode pins (M0, M1, M2).

Figure 16 shows the timing for Slave Serial configuration. The serial bitstream must be setup at the DIN input pin a short time before each rising edge of an externally generated CCLK.

Multiple FPGAs in Slave Serial mode can be daisy-chained for configuration from a single source. The maximum amount of data that can be sent to the DOUT pin for a serial daisy chain is $2^{20}-1$ (1,048,575) 32-bit words, or 33,554,400 bits, which is approximately 25 XC2S200 bitstreams. The configuration bitstream of downstream devices is limited to this size.

After an FPGA is configured, data for the next device is routed to the DOUT pin. Data on the DOUT pin changes on the rising edge of CCLK. Configuration must be delayed until \overline{INIT} pins of all daisy-chained FPGAs are High. For more information, see "Start-up," page 19.

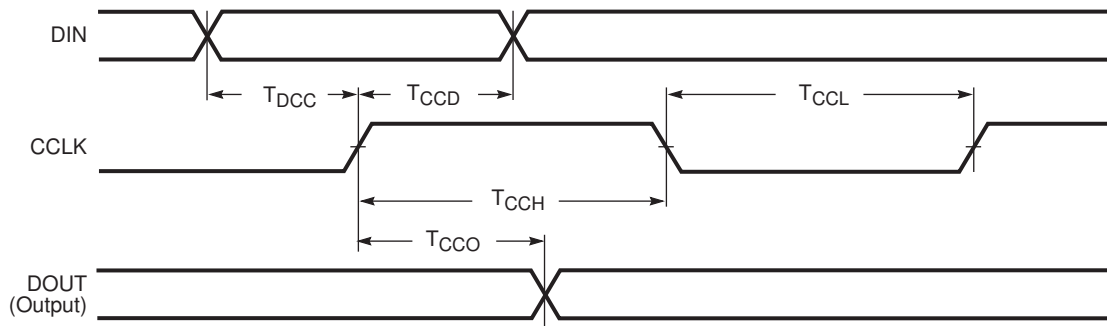


DS001_15_060608

Notes:

1. If the DriveDone configuration option is not active for any of the FPGAs, pull up DONE with a 330Ω resistor.

Figure 15: Master/Slave Serial Configuration Circuit Diagram



DS001_16_032300

Symbol		Description		Units
T_{DCC}	CCLK	DIN setup	5	ns, min
T_{CCD}		DIN hold	0	ns, min
T_{CCO}		DOUT	12	ns, max
T_{CCH}		High time	5	ns, min
T_{CCL}		Low time	5	ns, min
F_{CC}		Maximum frequency	66	MHz, max

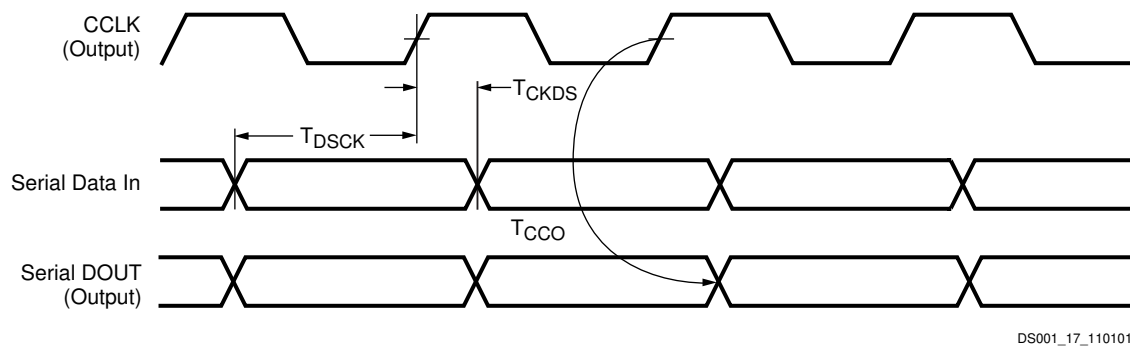
Figure 16: Slave Serial Mode Timing

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{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.



DS001_17_110101

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

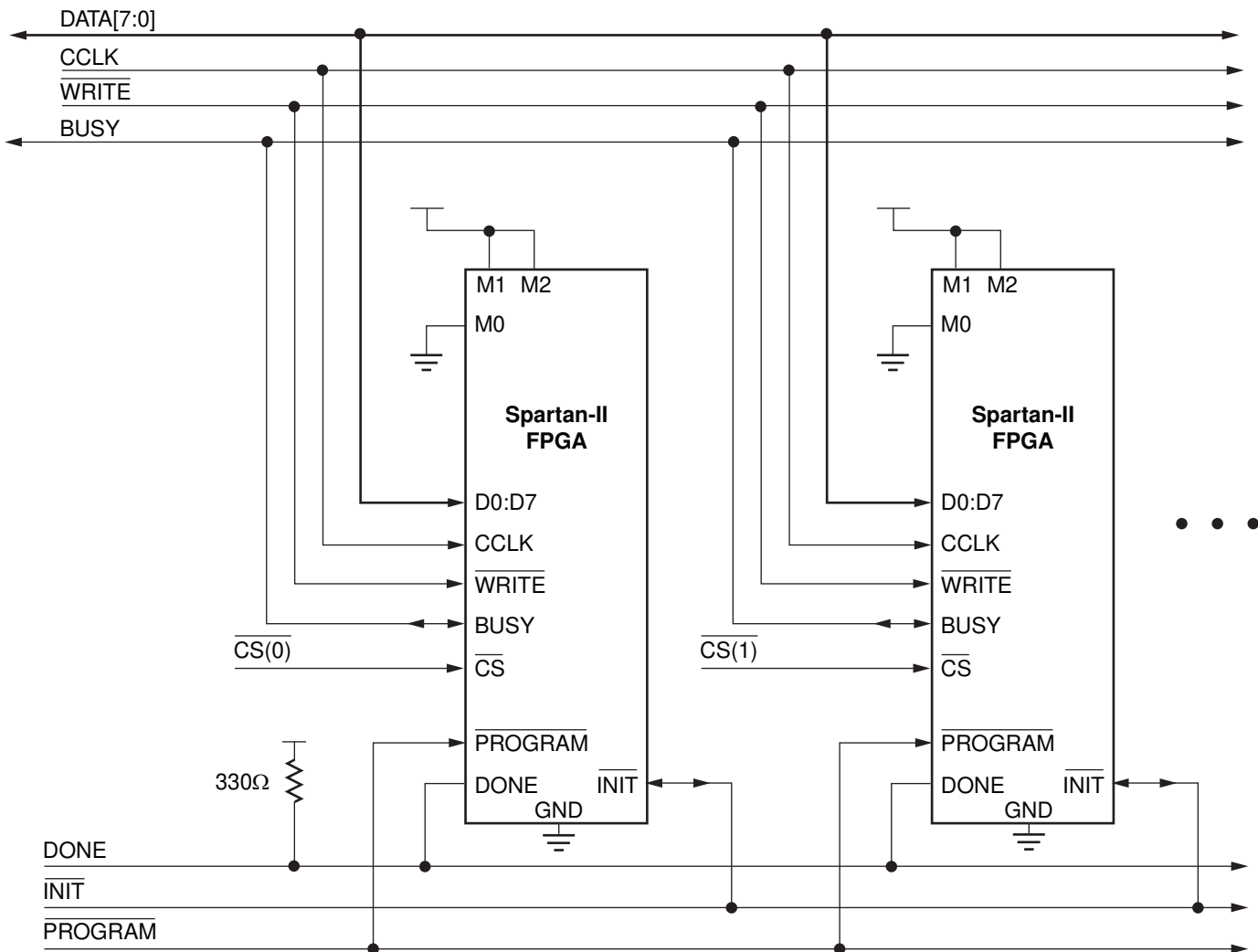
The Slave Parallel mode is the fastest configuration option. Byte-wide data is written into the FPGA. A BUSY flag is provided for controlling the flow of data at a clock frequency F_{CCNH} above 50 MHz.

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{CS}) signal and a Write signal (\overline{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{WRITE} . See "Readback," page 25.



DS001_18_060608

Figure 18: Slave Parallel Configuration Circuit Diagram

Multiple Spartan-II FPGAs can be configured using the Slave Parallel mode, and be made to start-up simultaneously. To configure multiple devices in this way, wire the individual CCLK, Data, $\overline{\text{WRITE}}$, and BUSY pins of all the devices in parallel. The individual devices are loaded separately by asserting the $\overline{\text{CS}}$ pin of each device in turn and writing the appropriate data. Sync-to-DONE start-up timing is used to ensure that the start-up sequence does not begin until all the FPGAs have been loaded. See "Start-up," page 19.

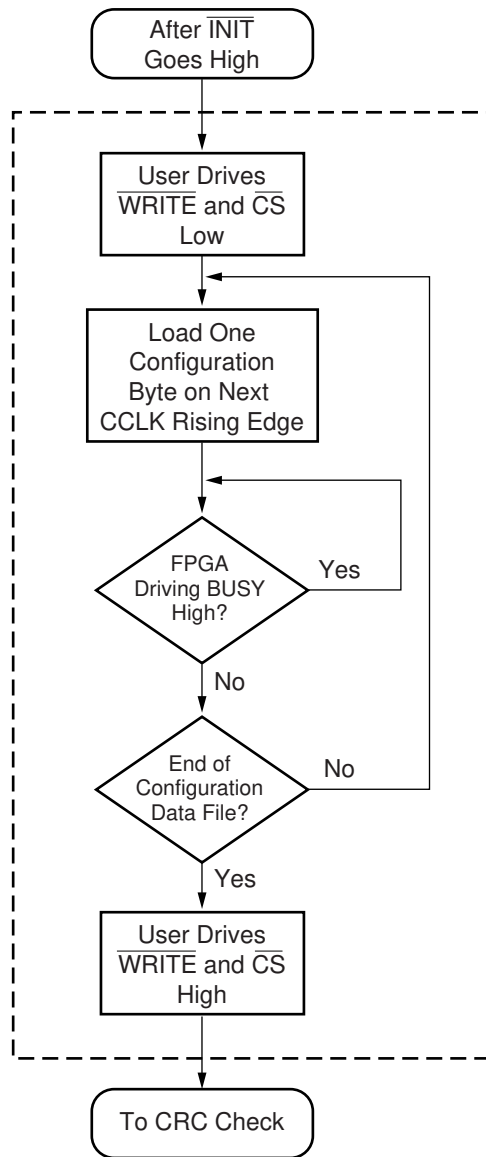
Write

When using the Slave Parallel Mode, write operations send packets of byte-wide configuration data into the FPGA. Figure 19, page 25 shows a flowchart of the write sequence used to load data into the Spartan-II FPGA. This is an expansion of the "Load Configuration Data Frames" block in Figure 11, page 18. The timing for write operations is shown in Figure 20, page 26.

For the present example, the user holds $\overline{\text{WRITE}}$ and $\overline{\text{CS}}$ Low throughout the sequence of write operations. Note that when $\overline{\text{CS}}$ is asserted on successive CCLKs, $\overline{\text{WRITE}}$ must remain either asserted or de-asserted. Otherwise an abort will be initiated, as in the next section.

1. Drive data onto D0-D7. Note that to avoid contention, the data source should not be enabled while $\overline{\text{CS}}$ is Low and $\overline{\text{WRITE}}$ is High. Similarly, while $\overline{\text{WRITE}}$ is High, no more than one device's $\overline{\text{CS}}$ should be asserted.
2. On the rising edge of CCLK: If BUSY is Low, the data is accepted on this clock. If BUSY is High (from a previous write), the data is not accepted. Acceptance will instead occur on the first clock after BUSY goes Low, and the data must be held until this happens.
3. Repeat steps 1 and 2 until all the data has been sent.
4. De-assert $\overline{\text{CS}}$ and $\overline{\text{WRITE}}$.

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.



DS001_19_032300

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{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{CS} signal may be de-asserted until the next byte is valid on D0-D7. While \overline{CS} is High, the Slave Parallel

interface does not expect any data and ignores all CCLK transitions. However, to avoid aborting configuration, \overline{WRITE} must continue to be asserted while \overline{CS} is asserted.

Abort

To abort configuration during a write sequence, de-assert \overline{WRITE} while holding \overline{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*.