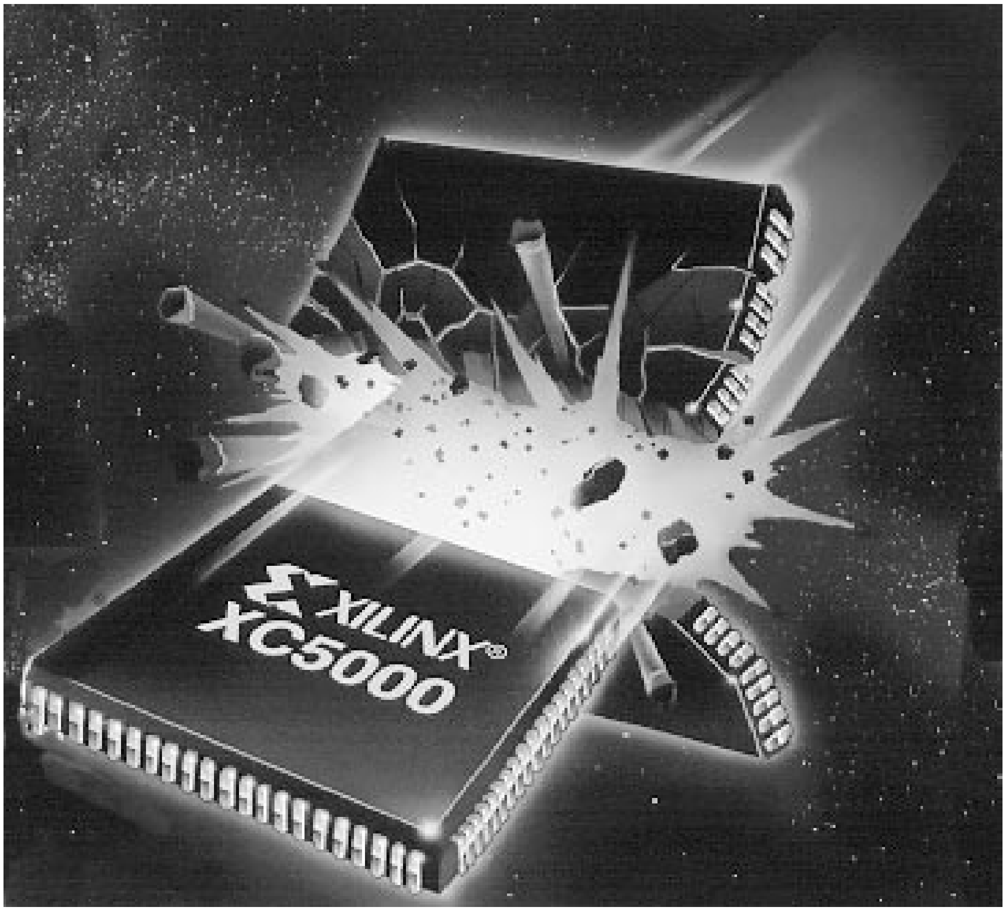
XILINX®

Technical
Data

XC5200
Logic Cell Array Family



Preliminary (v1.0) • April 1995

XILINX® and XACT are registered trademarks of Xilinx. All XC-prefix product designations, XACT-Performance, X-BLOX, XChecker, XDM, LCA, Logic Cell, Express, VersaBlock, and VersaRing are trademarks of Xilinx. The Programmable Logic Company and The Programmable Gate Array Company are service marks of Xilinx.

Mentor is a registered trademark and NETED, Design Architect, QuickSim, and QuickSim II are trademarks of Mentor Graphics, Inc. OrCAD is a registered trademark of OrCAD Systems Corporation. Viewlogic, Viewsim, and Viewdraw are registered trademarks of Viewlogic Systems, Inc. Synopsys is a registered trademark of Synopsys, Inc. Xilinx does not assume any liability arising out of the application or use of any product described herein; nor does it convey any license under its patents, copyrights, or maskwork rights or any rights of others.

Xilinx reserves the right to make changes, at any time, in order to improve reliability, function or design and to supply the best product possible. Xilinx will not assume responsibility for the use of any circuitry described herein other than circuitry entirely embodied in its products. Xilinx devices and products are protected under one or more of the following U.S. Patents: 4,642,487; 4,695,740; 4,706,216; 4,713,557; 4,746,822; 4,750,155; 4,758,985; 4,783,607; 4,820,937; 4,821,233; 4,835,418; 4,853,626; 4,855,619; 4,855,669; 4,902,910; 4,940,909; 4,967,107; 5,012,135; 5,023,606; 5,028,821; 5,047,710; 5,068,603; 5,140,193; 5,148,390; 5,155,432; 5,166,858; 5,224,056; 5,243,238; 5,267,187; 5,224,056; 5,245,277; 5,291,079; 5,295,090; 5,302,866; 5,319,252; 5,319,254; 5,321,704; 5,329,174; 5,329,181; 5,331,220; 5,331,226; 5,332,929; 5,337,255; 5,343,406; 5,349,248; 5,349,249; 5,349,250; 5,349,691; 5,357,153; 5,360,747; 5,361,229; 5,362,999; 5,365,125; 5,367,207; 5,386,154; RE 34,363; RE 34,444; and RE 34,808. Xilinx, Inc. does not represent that devices shown or products described herein are free from patent infringement or from any other third-party right. Xilinx assumes no obligation to correct any errors contained herein or to advise any user of this text of any correction if such be made. Xilinx will not assume any liability for the accuracy or correctness of any engineering or software support or assistance provided to a user.

Xilinx products are not intended for use in life support appliances, devices, or systems. Use of a Xilinx product in such applications without the written consent of the appropriate Xilinx officer is prohibited.

XILINX®

TABLE OF CONTENTS

XILINX®

XC5200
Logic Cell Array Family

Preliminary (v1.0)

Product Description

Features

- High-density family of Field-Programmable Gate Arrays (FPGAs)
- Design- and process-optimized for low cost
  - — 0.6-µm three-layer metal (TLM) process
- System performance up to 50 MHz
- SRAM-based, in-system reprogrammable architecture
- Flexible architecture with abundant routing resources
  - — VersaBlock™ logic module
  - — VersaRing™ I/O interface
  - — Dedicated cell-feedthrough path
  - — Hierarchical interconnect structure
  - — Extensive registers/latches
  - — Dedicated carry logic for arithmetic functions
  - — Cascade chain for wide input functions
  - — Dedicated IEEE 1149.1 boundary-scan logic
  - — Internal 3-state bussing capability
  - — Four global low-skew clock or signal distribution nets
  - — Globally selectable CMOS or TTL input thresholds
  - — Output slew-rate control
  - — 8-mA sink current per output
- Configured by loading binary file
  - — Unlimited reprogrammability
  - — Six programming modes, including high-speed Express™ mode
- 100% factory tested
- 100% footprint compatibility for common packages
- Fully supported by XACT® Development System
  - — Includes complete support for XACT-Performance™, X-BLOX™, Unified Libraries, Relationally Placed Macros (RPMs), XDelay, and XChecker™
  - — Wide selection of PC and workstation platforms
  - — Interfaces to more than 100 third-party CAE tools

Description

The XC5200 Field-Programmable Gate Array Family is engineered to deliver the lowest cost of any FPGA family. By optimizing the new XC5200 architecture for TLM technology and 0.6-µm CMOS SRAM process, dramatic advances have been made in silicon efficiency. These advances position the XC5200 family as a cost-effective, high-volume alternative to gate arrays.

Building on experiences gained with three previous successful SRAM FPGA families, the XC5200 family brings a robust feature set to high-density programmable logic design. The VersaBlock logic module, the VersaRing I/O interface, and a rich hierarchy of interconnect resources combine to enhance design flexibility and reduce time-to-market.

Complete support for the XC5200 family is delivered through the familiar XACT software environment. The XC5200 family is fully supported on popular workstation and PC platforms. Popular design entry methods are fully supported, including ABEL, schematic capture, and synthesis. Designers utilizing logic synthesis can use their existing Synopsys, Viewlogic, Mentor, and Exemplar tools to design with the XC5200 devices.

Table 1. Initial XC5200 Field-Programmable Gate Array Family Members

Device	XC5202	XC5204	XC5206	XC5210	XC5215
Typical Gate Range	2,200 - 2,700	3,900 - 4,800	6,000 - 7,500	10,000 - 12,000	14,000 - 18,000
VersaBlock Array	8 x 8	10 x 12	14 x 14	18 x 18	22 x 22
Number of CLBs	64	120	196	324	484
Number of Flip-Flops	256	480	784	1,296	1,936
Number of I/Os	84	124	148	196	244
TBUFs per Horizontal Longline	10	14	16	20	24

XC5200 Family Compared to XC4000 Family

For those readers already familiar with the XC4000 family of Xilinx Field-Programmable Gate Arrays, here is a concise description of the similarities and differences between the XC4000 and XC5200 families.

Superficially, the XC5200 family is quite similar to the XC4000 family. Both use CMOS SRAM technology. Both use 4-input lookup tables with unshared inputs. Both have a dedicated fast carry track, and dedicated boundary-scan logic in the input/output blocks (IOBs).

XC5200 and XC4000 devices are footprint and pin-out compatible; their pin names and pin locations are identical. XC5200 devices offer the same configuration options as XC4000 devices, and they can be intermixed with XC4000 devices in a configuration daisy chain.

There are also, however, significant differences between the two families:

- XC5200 lookup tables cannot be used as RAM.
- The XC5200 family offers dedicated carry logic, but differs from the XC4000 family in that the sum is generated in an additional function generator in the adjacent column. An XC5200 device thus uses twice as many function generators for adders, subtracters, accumulators, and some counters. Note, however, that a loadable up/down counter requires the same number of function generators in both families.
- XC5200 devices have no dedicated wide edge decoders. The XC5200 carry logic, unlike the XC4000 architecture, can be used to cascade function generators to implement wide AND and OR functions, for example.
- The XC5200 family contains a flexible coupling of logic and local routing resources called the VersaBlock. The XC5200 VersaBlock element includes the Configurable Logic Block (CLB), a Local Interconnect Matrix (LIM), and direct connects to neighboring VersaBlocks.
- XC5200 CLBs are roughly equivalent to two XC4000 CLBs. Each XC5200 CLB contains four 4-input function generators and four registers, which are configured as four independent Logic Cells™ (LCs). The output from each function generator can be brought out as a CLB output and/or drive the D input of a flip-flop. Pairs of logic cells can be combined to form a 5-input function generator.
- There are four direct feedthrough paths per CLB, one per LC. These paths can provide extra data input lines or serve as local routes without consuming any logic resources.
- The XC5200 family has a global reset, whereas the XC4000 family has both a global set and a global reset.
- Unlike the XC4000 family, each register can be configured as either an edge-triggered D flip-flop or a transparent, level-sensitive latch.
- There are no dedicated IOB flip-flops, but there are fast direct connects to adjacent CLBs.

Table 2. Four Generations of Xilinx Field-Programmable Gate Array Families

Parameter	XC5200	XC4000	XC3000A/XC3100A	XC2000
Function generators per CLB	4	3	2	2
Logic inputs per CLB	20	9	5	4
Logic outputs per CLB	12	4	2	2
Low-skew global buffers	4	8	2	2
User RAM	no	yes	no	no
Dedicated decoders	no	yes	no	no
Cascade chain	yes	no	no	no
Fast carry logic	yes	yes	no	no
Internal 3-state drivers	yes	yes	yes	no
IEEE boundary scan	yes	yes	no	no
Output slew-rate control	yes	yes	yes	no
Power-down option	no	no	yes	yes
Crystal oscillator circuit	no	no	yes	yes

Table 3. Routing Resource Comparison

Resource	XC5200	XC4000
Single-length Lines	10	8
Double-length Lines	4	4
Longlines	8	6
Direct Connects	8	0
VersaRing	yes	no

- The TLM process allows significant improvements in the routing structure. Each XC5200 VersaBlock element has complete intra-CLB routing, the LIM, and offers four direct routing connections to each of the four neighboring CLBs (North, South, East, and West). Any function generator or flip-flop thus has unrestricted connectivity to 19 other function generators or flip-flops: three in its own CLB, and 16 in the adjacent CLBs. These direct connects do not compete with the general routing resources (see Table 3).
- Each XC5200 3-state buffer (TBUF) can drive up to two horizontal Longlines; each XC4000 TBUF accesses only one horizontal Longline.
- There is a special racetrack, the VersaRing, between the outer edge of the core CLB array and the ring of IOBs, providing significant help in overcoming the problems caused by early locking of I/O pins.
- There are no internal pull-ups for XC5200 Longlines.

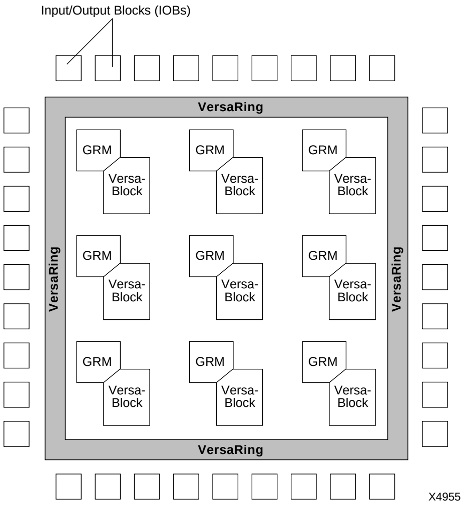


Figure 1. XC5200 Architectural Overview

Architectural Overview

Figure 1 presents a simplified, conceptual overview of the XC5200 architecture. Similar to conventional FPGAs, the XC5200 family consists of programmable IOBs, programmable logic blocks, and programmable interconnect. Unlike other FPGAs, however, the logic and local routing resources of the XC5200 family are combined in flexible VersaBlocks. General-purpose routing connects to the VersaBlock through the General Routing Matrix (GRM).

VersaBlock: Abundant Local Routing Plus Versatile Logic

The basic logic element in each VersaBlock structure is the Logic Cell, shown in Figure 2. Each LC contains a 4-input function generator (F), a storage device (FD), and control logic. There are five independent inputs and three outputs to each LC. The independence of the inputs and outputs allows the software to maximize the resource utilization within each LC. Each Logic Cell also contains a direct feedthrough path that does not sacrifice the use of either the function generator or the register; this feature is a first for FPGAs. The storage device is configurable as either a D flip-flop or a latch. The control logic consists of carry logic for fast implementation of arithmetic functions, which can also be configured as a cascade chain allowing decode of very wide input functions.

The XC5200 CLB consists of four LCs, as shown in Figure 3. Each CLB has 20 independent inputs and 12 independent outputs. The top and bottom pairs of LCs can be configured to implement 5-input functions. The challenge of FPGA implementation software has always been to maximize the usage of logic resources. The XC5200 family addresses this issue by surrounding each CLB with two types of local interconnect — the LIM and direct connects. These two interconnect resources, combined with the CLB, form the VersaBlock, represented in Figure 4.

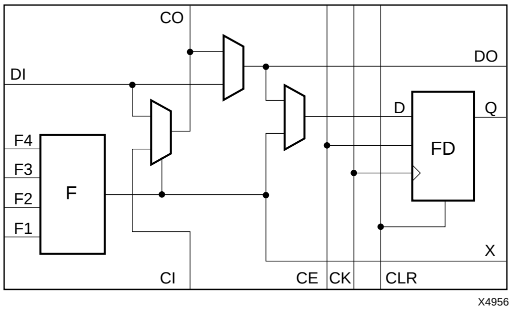


Figure 2. XC5200 Logic Cell (Four LCs per CLB)

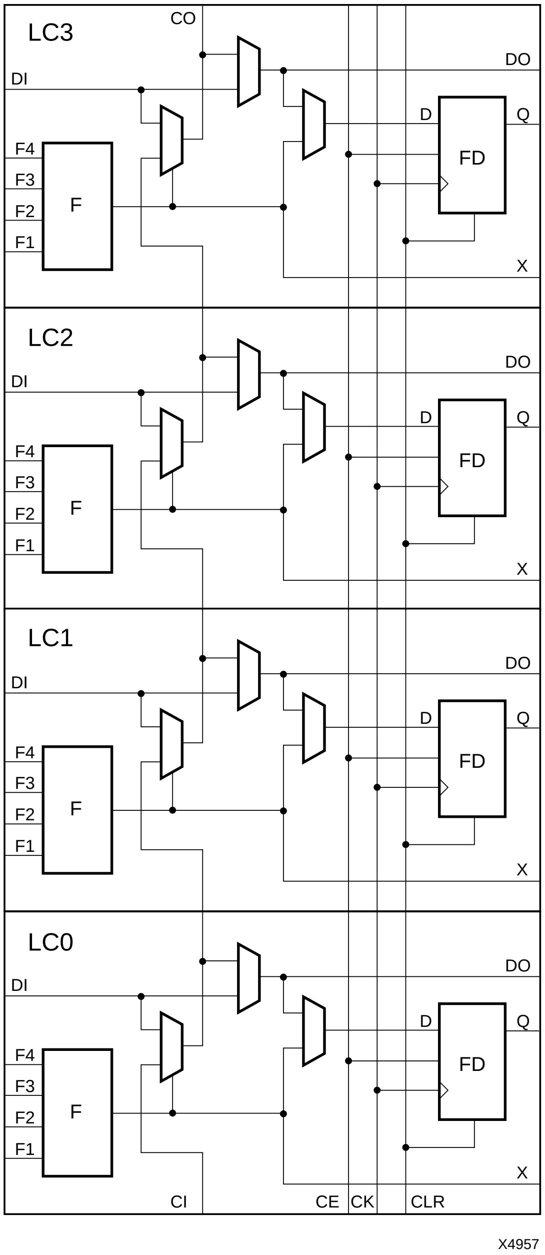


Figure 3. Configurable Logic Block

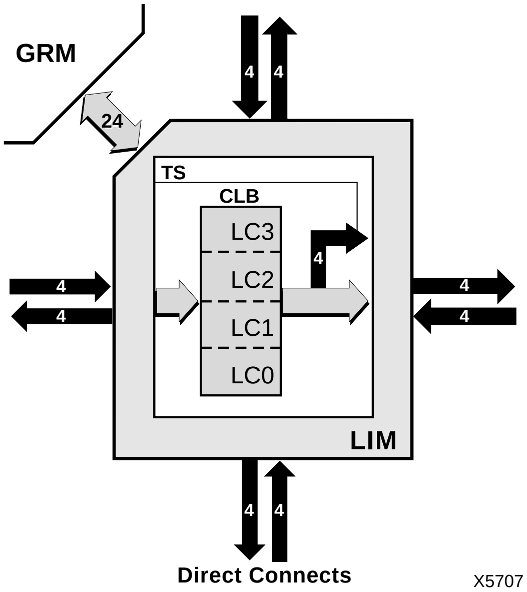


Figure 4. VersaBlock

The LIM provides 100% connectivity of the inputs and outputs of each LC in a given CLB. The benefit of the LIM is that no general routing resources are required to connect feedback paths within a CLB. The LIM connects to the GRM via 24 bidirectional nodes.

The direct connects allow immediate connections to neighboring CLBs, once again without using any of the general interconnect. These two layers of local routing resource improve the granularity of the architecture, effectively making the XC5200 family a “sea of logic cells.” Each VersaBlock has four 3-state buffers that share a common enable line and directly drive horizontal Longlines, creating robust on-chip bussing capability. The VersaBlock allows fast, local implementation of logic functions, effectively implementing user designs in a hierarchical fashion. These resources also minimize local routing congestion and improve the efficiency of the general interconnect, which is used for connecting larger groups of logic. It is this combination of both fine-grain and coarse-grain architecture attributes that maximize logic utilization in the XC5200 family. This symmetrical structure takes full advantage of the third metal layer, freeing the placement software to pack user logic optimally with minimal routing restrictions.

VersaRing I/O Interface

The interface between the IOBs and core logic has been redesigned in the XC5200 family. The IOBs are completely decoupled from the core logic. The XC5200 IOBs contain dedicated boundary-scan logic for added board-level testability, but do not include input or output registers. This approach allows a maximum number of IOBs to be placed around the device, improving the I/O-to-gate ratio and decreasing the cost per I/O. A “freeway” of interconnect cells surrounding the device forms the VersaRing, which provides connections from the IOBs to the internal logic These incremental routing resources provide abundant connections from each IOB to the nearest VersaBlock, in addition to Longline connections surrounding the device. The VersaRing eliminates the historic trade-off between high logic utilization and pin placement flexibility. These incremental edge resources give users increased flexibility in preassigning (i.e., locking) I/O pins before completing their logic designs. This ability accelerates time-to-market, since PCBs and other system components can be manufactured concurrent with the logic design.

General Routing Matrix

The GRM is functionally similar to the switch matrices found in other architectures, but it is novel in its tight coupling to the logic resources contained in the VersaBlocks. Advanced simulation tools were used during the development of the XC5200 architecture to determine the optimal level of routing resources required. The XC5200 family contains six levels of interconnect hierarchy — a series of single-length lines, double-length lines, and Longlines all routed through the GRM. The direct connects, LIM, and logic-cell feedthrough are contained within each VersaBlock. Throughout the XC5200 interconnect, an efficient multiplexing scheme, in combination with TLM, was used to improve the overall efficiency of silicon usage.

Performance Overview

The XC5200 family has been benchmarked with many designs running synchronous clock rates up to 40 MHz. The performance of any design depends on the circuit to be implemented, and the delay through the combinatorial and sequential logic elements, plus the delay in the interconnect routing. Table 4 shows some performance numbers for representative circuits, using worst-case timing parameters for the Engineering Sample (ES) speed grade. A rough estimate of timing can be made by assuming 6 ns per logic level, which includes direct-connect routing delays. More accurate estimations can be made using the information in the Switching Characteristic Guideline section.

Table 4. Performance for Several Common Circuit Functions

Function	XC5200 Speed Grade		
	-6	-5	-4
16-bit Decoder from Input Pad	9 ns	8 ns	
24-bit Accumulator	32 MHz	39 MHz	
16-to-1 Multiplexer	16 ns	13 ns	
16-bit Unidirectional Loadable Counter	40 MHz	50 MHz	
16-bit U/D Counter	40 MHz	50 MHz	
16-bit Adder	24 ns	20 ns	
24-bit Loadable U/D Counter	36 MHz	42 MHz	

Development System

The powerful features of the XC5200 device family require an equally powerful, yet easy-to-use, set of development tools. Xilinx provides an enhanced version of the Xilinx Automatic CAE Tools (XACT), optimized for the XC5200 family.

As with other logic technologies, the basic methodology for XC5200 FPGA design consists of three interrelated steps: design entry, implementation, and verification. Popular generic tools are used for entry and simulation (for example, Viewlogic Systems's Viewdraw schematic editor and Viewsim simulator), but architecture-specific tools are needed for implementation.

All Xilinx development system software is integrated under the Xilinx Design Manager (XDM™), providing designers with a common user interface regardless of their choices of entry and verification tools. XDM simplifies the selection of command-line options with pull-down menus and online help text. Application programs ranging from schematic capture to Partitioning, Placement, and Routing (PPR) can be accessed from XDM, while the program-command sequence is generated and stored for documentation prior to execution. The XMAKE command, a design compilation utility, automates the entire implementation process, automatically retrieving the design's input files and performing all the steps needed to create configuration and report files.

Several advanced features of the XACT system facilitate XC5200 FPGA design. RPMs — schematic-based macros with relative location constraints to guide their placement within the FPGA — help to ensure an optimized implementation for common logic functions. An abundance of local routing permits RPMs to be contained within a single VersaBlock or to span across multiple VersaBlocks. XACT-Performance allows designers to enter the exact performance requirements during design entry, at the schematic level, to guide PPR.

Design Entry

Designs can be entered graphically, using schematic-capture software, or in any of several text-based formats (such as Boolean equations, state-machine descriptions, and high-level design languages).

Xilinx and third-party CAE vendors have developed library and interface products compatible with a wide variety of design-entry and simulation environments. A standard interface-file specification, Xilinx Netlist File (XNF), is provided to simplify file transfers into and out of the XACT development system.

Xilinx offers XACT development system interfaces to the following design environments:

- Viewlogic Systems (Viewdraw, Viewsim)
- Mentor Graphics V8 (NETED, QuickSim, Design Architect, QuickSim II)
- OrCAD (SDT, VST)
- Synopsys (Design Compiler, FPGA Compiler)
- Xilinx-ABEL (State Machine module generator)
- X-BLOX (Graphical Mode Generator)

Many other environments are supported by third-party vendors. Currently, more than 100 packages are supported.

The unified schematic library for the XC5200 FPGA reflects the wide variety of logic functions that can be implemented in these versatile devices. The 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.

Designing with macros is as easy as designing with standard SSI/MSI functions. The “soft macro” library contains detailed descriptions of common logic functions, but does not contain any partitioning or routing information. The performance of these macros depends, therefore, on how the PPR software processes the design. RPMs, on the other hand, do contain predetermined partitioning and relative placement information, resulting in an optimized implementation for these functions. Users can create their own library elements — either soft macros or RPMs — based on the macros and primitives of the standard library.

The X-BLOX design language is a graphics-based high-level description language (HDL) that allows designers to use a schematic editor to enter designs as a set of generic modules. The X-BLOX compiler synthesizes and optimizes the modules for the target device architecture, automatically choosing the appropriate architectural resources for each function.

The XACT design environment supports hierarchical design entry, with top-level drawings defining the major functional blocks, and lower-level descriptions defining the logic in each block. The implementation tools automatically combine the hierarchical elements of a design. Different hierarchical elements can be specified with different design entry tools, allowing the use of the most convenient entry method for each portion of the design.

Design Implementation

The design implementation tools satisfy the requirements for an automated design process. Logic partitioning, block placement, and signal routing are performed by the PPR program. The partitioner takes the logic from the entered design and maps the logic into the architectural resources of the FPGA (such as the logic blocks, I/O blocks, and 3-state buffers). The placer then determines the best locations for the blocks, depending on their connectivity and the required performance. The router finally connects the placed blocks together.

The PPR algorithms support fully automatic implementation of most designs. However, for demanding applications, the user may exercise various degrees of control over the automated implementation process. Optionally, user-designated partitioning, placement, and routing information can be specified as part of the design-entry process. The implementation of highly structured designs can benefit greatly from the basic floorplanning techniques familiar to designers of large gate arrays.

The PPR program includes XACT-Performance, a feature that allows designers to specify the timing requirements along entire paths during design entry. Timing path analysis routines in PPR then recognize and accommodate the user-specified requirements. Timing requirements can be entered on the schematic in a form directly relating to the system requirements (such as the targeted minimum clock frequency, or the maximum allowable delay on the data path between two registers). So, while the timing of each individual net is not predictable, the overall performance of the system along entire signal paths is automatically tailored to match user-generated specifications.

Design Verification

The high development cost associated with common mask-programmed gate arrays necessitates extensive simulation to verify a design. Due to the custom nature of masked gate arrays, mistakes or last-minute design changes cannot be tolerated. A gate-array designer must simulate and test all logic using simulation software. Simulation describes what happens in a system under worst-case situations. However, simulation can be tedious and slow, and simulation vectors must be generated. A few seconds of system time can take weeks to simulate.

Programmable-gate-array users, however, can use in-circuit debugging techniques in addition to simulation. Because Xilinx devices are reprogrammable, designs can be verified in real time without the need for extensive simulation vectors.

The XACT development system supports both simulation and in-circuit debugging techniques. For simulation, the system extracts the post-layout timing information from the design database. This data can then be sent to the simulator to verify timing-critical portions of the design database using XDELAY, the Xilinx static timing analyzer tool. Back-annotation — the process of mapping the timing information back into the signal names and symbols of the schematic — eases the debugging effort.

For in-circuit debugging, the XACT development system includes a serial download and readback cable (XChecker) that connects the FPGA in the system to the PC or workstation through an RS232 serial port. The engineer can download a design or a design revision into the system for testing. The designer can also single-step the logic, read the contents of the numerous flip-flops on the device, and observe internal logic levels. Simple modifications can be downloaded into the system in a matter of minutes.

Detailed Functional Description

CLB Logic

Figure 3 shows the logic in the XC5200 CLB, which consists of four Logic Cells (LC[3:0]). Each Logic Cell consists of an independent 4-input Lookup Table (LUT), and a D-Type flip-flop or latch with common clock, clock enable, and clear, but individually selectable clock polarity. Additional logic features provided in the CLB are:

- High-speed carry propagate logic.
- High-speed pattern decoding.
- High-speed direct connection to flip-flop D-inputs.
- Each flip-flop can be programmed individually as either a transparent, level-sensitive latch or a D flip-flop.
- Four 3-state buffers with a shared Output Enable.
- Two 4-input LUTs can be combined to form an independent 5-input LUT.

5-Input Functions

Figure 5 illustrates how the outputs from the LUTs from LC0 and LC1 can be combined with a 2:1 multiplexer (F5_MUX) to provide a 5-input function. The outputs from the LUTs of LC2 and LC3 can be similarly combined.

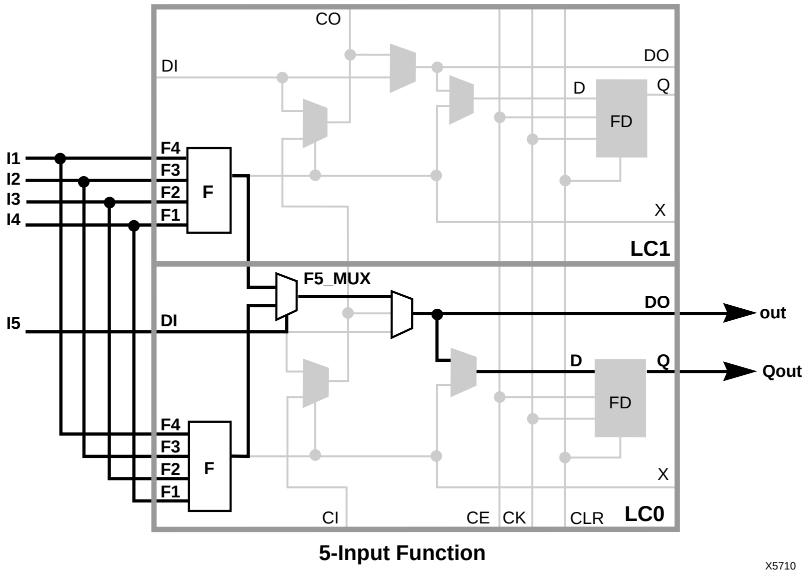


Figure 5. Two LUTs in Parallel Combined to Create a 5-input Function

Carry Function

The XC5200 family supports a carry-logic feature that enhances the performance of arithmetic functions such as counters, adders, etc. A carry multiplexer (CY_MUX) symbol on a schematic is used to indicate the XC5200 carry logic. This symbol represents the dedicated 2:1 multiplexer in each LC that performs the one-bit high-speed carry propagate per logic cell (four bits per CLB).

While the carry propagate is performed inside the LC, an adjacent LC must be used to complete the arithmetic function. Figure 6 represents an example of an adder function. The carry propagate is performed on the CLB shown, which also generates the half-sum for the four-bit adder. An adjacent CLB is responsible for XORing the half-sum with the corresponding carry-out. Thus an adder or counter requires two LCs per bit. Notice that the carry chain requires an initialization stage, which the XC5200 family accomplishes using the carry initialize (CY_INIT) macro and one additional LC.

The XC5200 library contains a set of RPMs and arithmetic functions designed to take advantage of the dedicated carry logic. Using and modifying these macros makes it much easier to implement customized RPMs, freeing the designer from the need to become an expert on architectures.

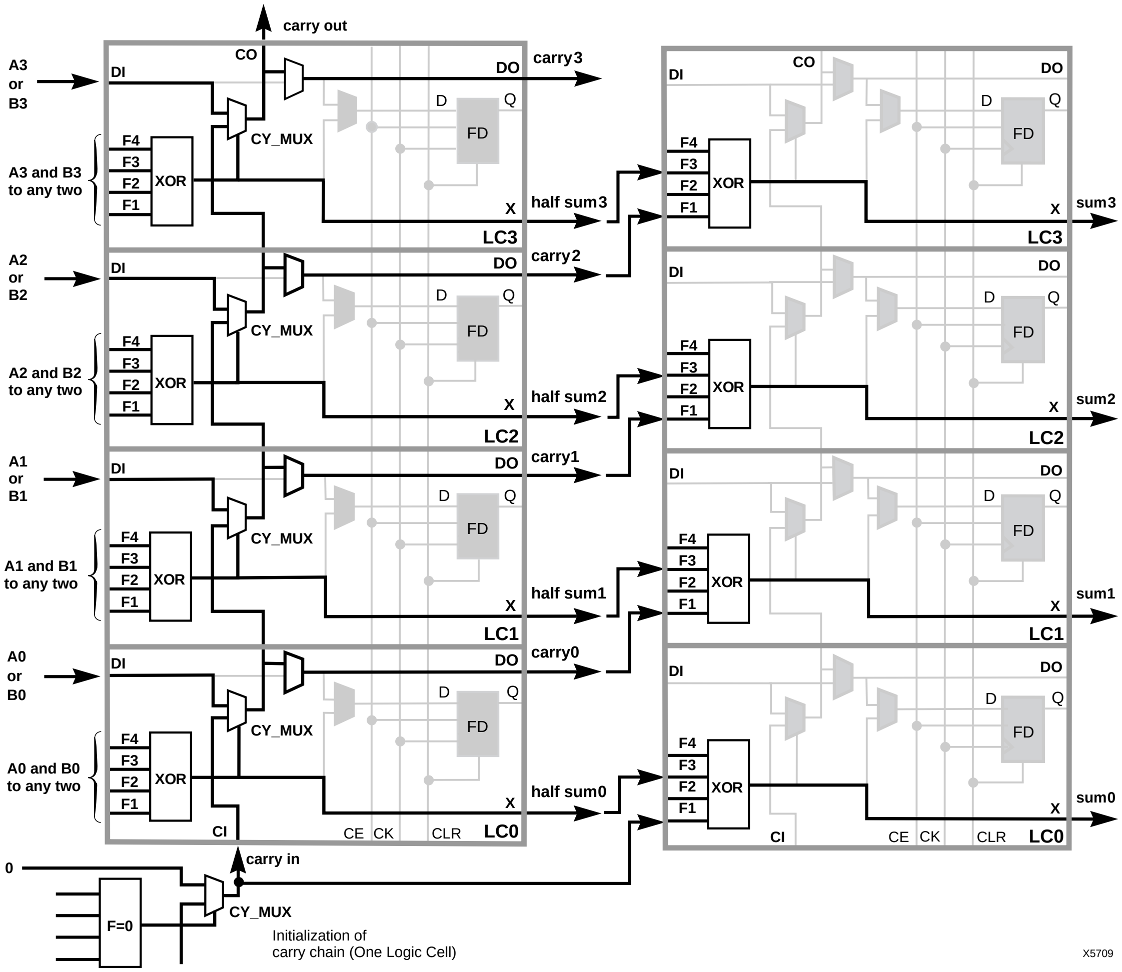


Figure 6. XC5200 CY_MUX Used for Adder Carry Propagate

Cascade Function

Each CY_MUX can be connected to the CY_MUX in the adjacent LC to provide cascadable decode logic. Figure 7 illustrates how the 4-input function generators can be configured to take advantage of these four cascaded CY_MUXes. Note that AND and OR cascading are specific cases of a general decode. In AND cascading all bits are decoded equal to logic one, while in OR cascading all bits are decoded equal to logic zero. The flexibility of the LUT achieves this result.

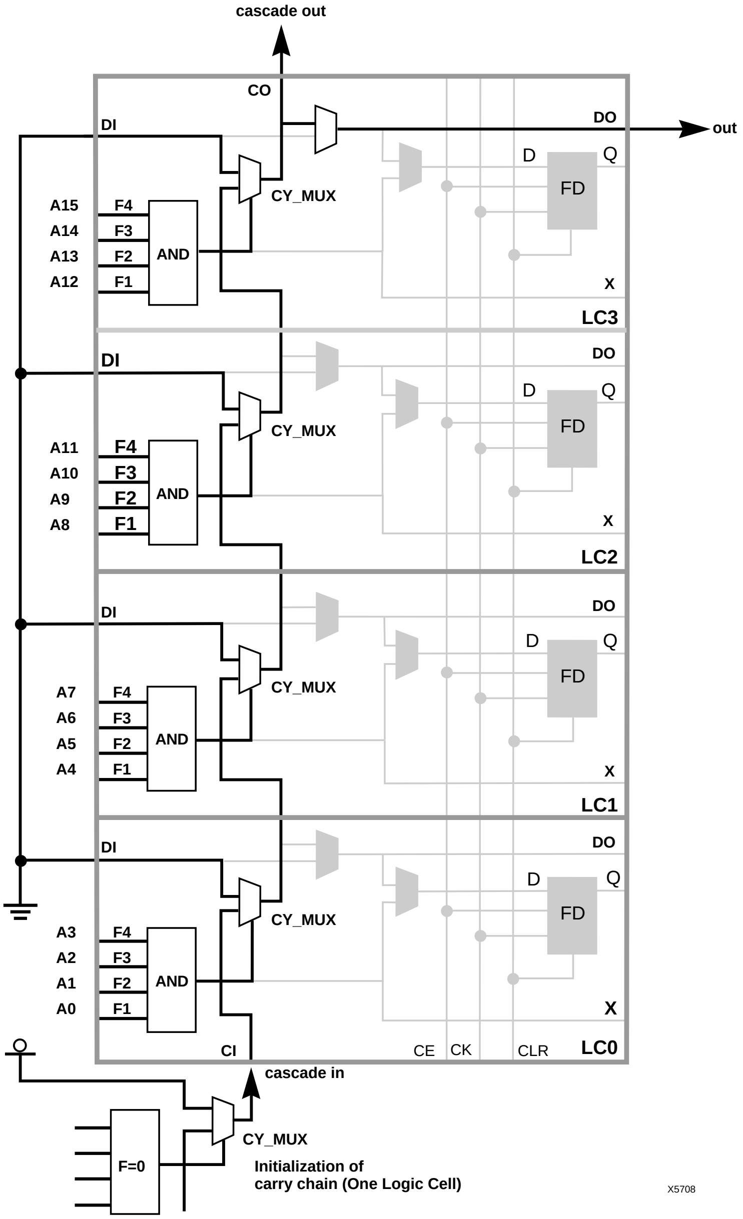


Figure 7. XC5200 CY_MUX Used for Decoder Cascade Logic

3-State Buffers

The XC5200 family has four dedicated TBUFs per CLB. The four buffers are individually configurable through four configuration bits to operate as simple non-inverting buffers or in 3-state mode. When in 3-state mode the CLB's output enable (TS) control signal drives the enable to all four buffers (see Figure 8). Each TBUF can drive up to two horizontal Longlines

Oscillator

The XC5200 oscillator (OSC52) divides the internal 16-MHz clock or a user clock that is connected to the "C" pin. The user then has the choice of dividing by 4, 16, 64, or 256 for the "OSC1" output and dividing by 2, 8, 32, 128, 1024, 4096, 16384, or 65536 for the "OSC2" output. The division is specified via a "DIVIDEn_BY=x" attribute on the symbol, where n=1 for OSC1, or n=2 for OSC2. The OSC5 macro is used where an internal oscillator is required. The CK_DIV macro is applicable when a user clock input is specified (see Figure 9).

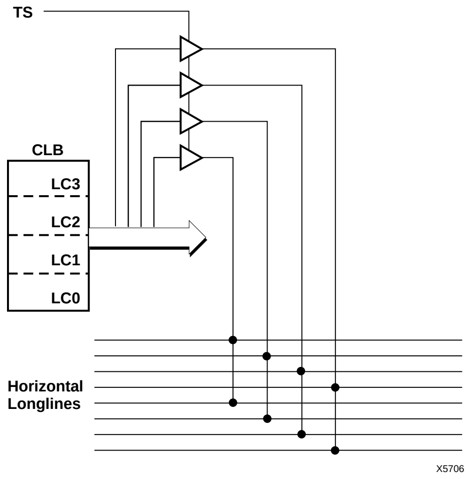


Figure 8. XC5200 3-State Buffer

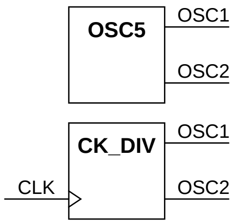


Figure 9. XC5200 Oscillator Macros

Start-Up

On start-up, all XC5200 internal flip-flops are reset. The XC5200 devices do not support the "INIT=" attribute. Thus, the XC5200 family has only a global reset (GR) signal. The user can assign the pin location for the GR signal and use it to reset asynchronously all of the flip-flops in the design without using general routing resources. The user can also assign a positive or negative polarity to GR.

Boundary Scan

XC5200 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 can also support two USERCODE instructions.

Boundary-scan operation is independent of individual IOB configuration and package type. All IOBs are treated as independently controlled bidirectional pins, including any unbonded IOBs. Retaining the bidirectional test capability after configuration provides flexibility for interconnect testing.

Also, internal signals can be captured during EXTEST by connecting them to unbonded IOBs, or to the unused outputs in IOBs used as unidirectional input pins. This technique partially compensates for the lack of INTEST support.

The public boundary-scan instructions are always available prior to configuration. After configuration, the public instructions and any USERCODE instructions are only available if specified in the design. While SAMPLE and BYPASS are available during configuration, it is recommended that boundary-scan operations not be performed during this transitory period.

In addition to the test instructions outlined above, the boundary-scan circuitry can be used to configure the Logic Cell Array (LCA™) device, and to read back the configuration data.

All of the XC4000 boundary-scan modes are supported in the XC5200 family. Three additional outputs for the User Register are provided (Reset, Update, and Shift), representing the decoding of the corresponding state of the boundary-scan internal state machine. For details on boundary scan, refer to "Boundary Scan in XC4000 Devices — Application Note" on pages 8-45 through 8-42 of the 1994 Xilinx Programmable Logic Data Book.

VersaBlock Routing

Local Interconnect Matrix

The GRM connects to the VersaBlock via 24 bidirectional ports (M0-M23). Excluding direct connections, global nets, and 3-statable Longlines, all VersaBlock inputs and outputs connect to the GRM via these 24 ports. Four 3-statable unidirectional signals (TQ0-TQ3) drive out of the VersaBlock directly onto the horizontal Longlines. Two horizontal global nets (GH0 and GH1) and two vertical global nets (GV0 and GV1) connect directly to every CLB clock pin; they can connect to other CLB inputs via the GRM. Each CLB also has four unidirectional direct connects to each of its four neighboring CLBs. These direct connects can also feed directly back to the CLB (see Figure 10).

In addition, each CLB has 16 direct inputs, four direct connections from each of the neighboring CLBs. These direct connections provide high-speed local routing that bypasses the GRM.

The 13 CLB outputs (12 LC outputs plus a V_{cc}/GND signal) connect to the eight VersaBlock outputs via the output multiplexers, which consist of eight fully populated 13-to-1 multiplexers. Of the eight VersaBlock outputs, four signals drive each neighboring CLB directly, and provide a direct feedback path to the input multiplexers. The four remaining multiplexer outputs can drive the GRM through four TBUFs (TQ0-TQ3). All eight multiplexer outputs can connect to the GRM through the bidirectional M0-M23 signals. All eight signals also connect to the input multiplexers and are potential inputs to that CLB.

CLB inputs have several possible sources: the 24 signals from the GRM, 16 direct connections from neighboring VersaBlocks, four signals from global, low-skew buffers (GH0, GH1, GV0, and GV1), and the four signals from the CLB output multiplexers. Unlike the output multiplexers, the input multiplexers are not fully populated; i.e., only a subset of the available signals can be connected to a given CLB input. The flexibility of LUT input swapping and LUT mapping compensates for this limitation. For example, if a 2-input NAND gate is required, it can be mapped into any of the four LUTs, and use any two of the four inputs to the LUT.

Direct Connects

The unidirectional direct-connect segments are connected to the logic input/output pins through the CLB’s input and output multiplexer array, and thus bypass the programmable routing matrix altogether. These lines are intended to increase the routing channel utilization where possible, while simultaneously reducing the delay incurred in speed-critical connections.

The direct connects also provide a high-speed path from the edge CLBs to the VersaRing input/output buffers, and thus reduce set-up time, clock-to-out, and combinational propagation delay.

The direct connects are ideal for developing customized RPM cells. Using direct connects improves the macro performance, and leaves the other routing channels intact for improved routing. Direct connects can also route through a CLB using one of the four cell-feedthrough paths.

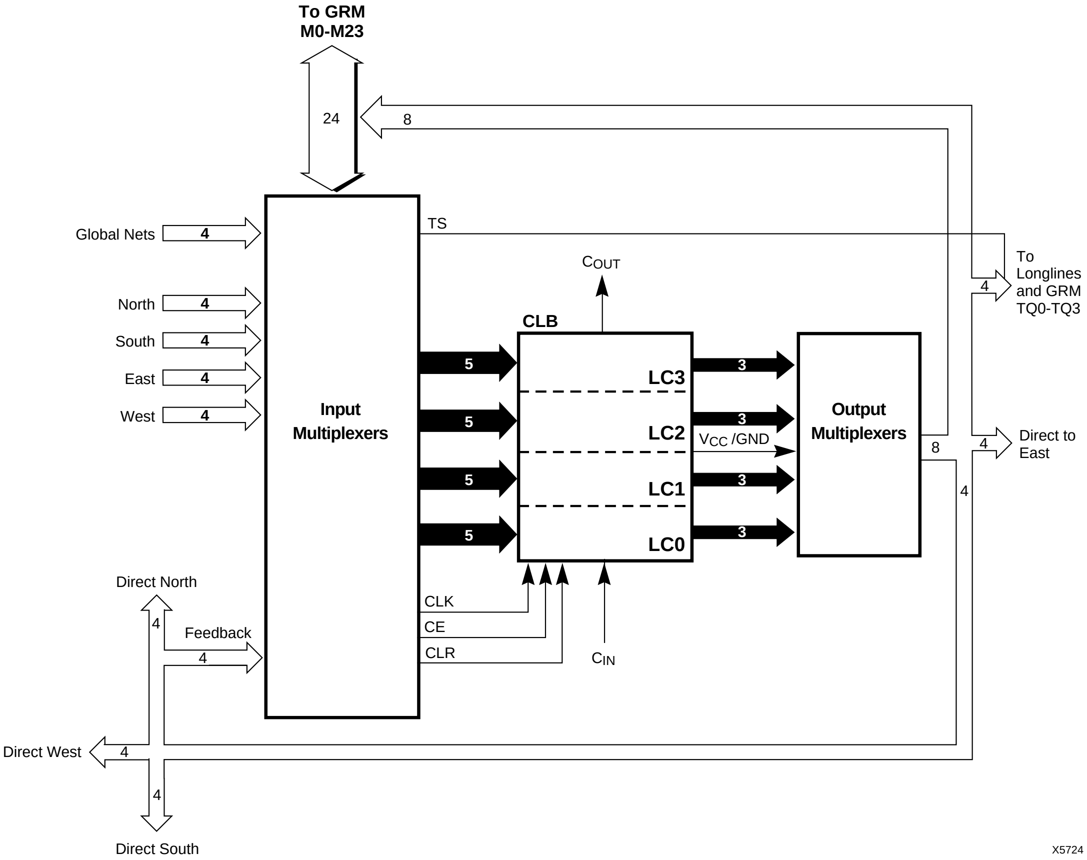


Figure 10. VersaBlock Details

General Routing Matrix

The General Routing Matrix, shown in Figure 11, provides flexible bidirectional connections to the Local Interconnect Matrix through a hierarchy of different-length metal segments in both the horizontal and vertical directions. A programmable interconnect point (PIP) establishes an electrical connection between two wire segments. The PIP, consisting of a pass transistor switch controlled by a memory element, provides bidirectional (in some cases, unidirectional) connection between two adjoining wires. A collection of PIPs inside the General Routing Matrix and in the Local Interconnect Matrix provides connectivity between various types of metal segments. A hierarchy of PIPs and associated routing segments combine to provide a powerful interconnect hierarchy:

- Forty bidirectional single-length segments per CLB provide ten routing channels to each of the four neighboring CLBs in four directions.
- Sixteen bidirectional double-length segments per CLB provide four routing channels to each of four other (non-neighboring) CLBs in four directions.
- Eight horizontal and eight vertical bidirectional Longline segments span the width and height of the chip, respectively.
- Two low-skew horizontal and vertical unidirectional global-line segments span each row and column of the chip, respectively.

Single- and Double-Length Lines

The single- and double-length bidirectional line segments make up the bulk of the routing channels. The double-length lines hop across every other CLB to reduce the propagation delays in speed-critical nets. Regenerating the signal strength is recommended after traversing three or four such segments. XACT place-and-route software automatically connects buffers in the path of the signal as necessary. Single- and double-length lines cannot drive onto Longlines and global lines; Longlines and global lines can, however, drive onto single- and double-length lines. As a general rule, Longline and global-line connections to the programmable routing matrix are unidirectional, with the signal direction from these lines toward the routing matrix.

Longlines

Longlines are used for high-fan-out signals, 3-state busses, low-skew nets, and faraway destinations. Row and column splitter PIPs in the middle of the array effectively double the total number of Longlines by electrically dividing them into two separated half-lines. The horizontal Longlines are driven by the 3-state buffers in each CLB, and are driven by similar buffers at the periphery of the array from the VersaRing I/O Interface.

Bus-oriented microprocessor designs are accommodated by using horizontal Longlines in conjunction with the 3-state buffers in the CLB and in the VersaRing. Additionally, programmable keeper cells at the periphery can be enabled to retain the last valid logic level on the Longlines when all buffers are in 3-state mode.

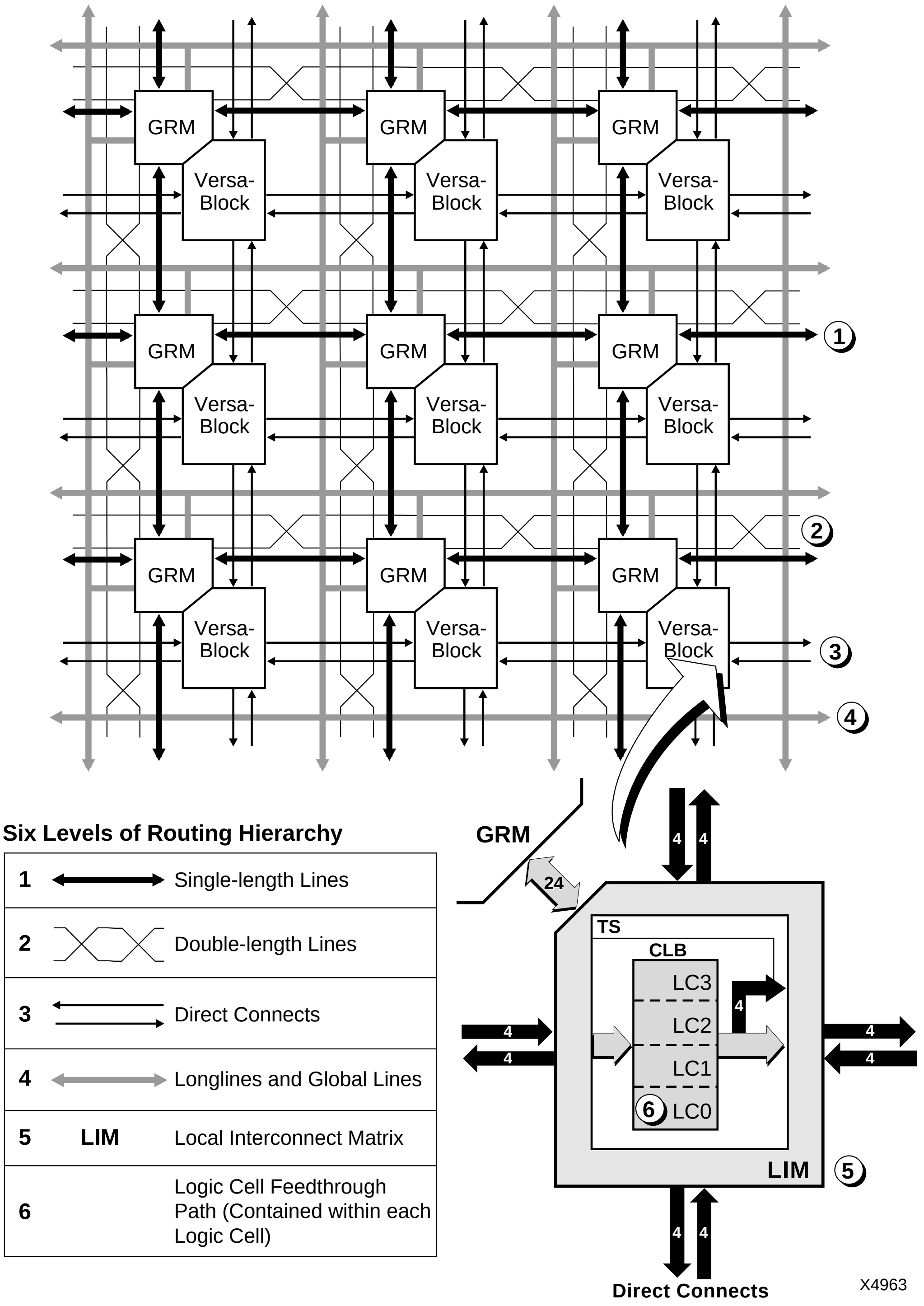
Longlines connect to the single-length or double-length lines, or to the logic inside the CLB, through the General Routing Matrix. The only manner in which a Longline can be driven is through the four 3-state buffers; therefore, a Longline-to-Longline or single-line-to-Longline connection through PIPs in the General Routing Matrix is not possible. Again, as a general rule, long- and global-line connections to the General Routing Matrix are unidirectional, with the signal direction from these lines toward the routing matrix.

The XC5200 family has no pull-ups on the ends of the Longlines sourced by TBUFs. Consequently, wired functions (i.e., WAND and WORAND) and wide multiplexing functions requiring pull-ups for undefined states (i.e., bus applications) must be implemented in a different way. In the case of the wired functions, the same functionality can be achieved by taking advantage of the carry/cascade logic described above, implementing a wide logic function in place of the wired function. In the case of 3-state bus applications, the user must insure that all states of the multiplexing function are defined. This process is as simple as adding an additional TBUF to drive the bus High when the previously undefined states are activated.

Global Clock Buffers

Global buffers in Xilinx FPGAs are special buffers that drive a dedicated routing network called Global Lines, as shown in Figure 12. This network is intended for high-fan-out clocks or other control signals, to maximize speed and minimize skewing while distributing the signal to many loads.

The XC5200 family has a total of four global buffers (BUFG symbol in the library), each with its own dedicated routing channel. Two are distributed vertically and two horizontally throughout the LCA.



Six Levels of Routing Hierarchy

1	⟷	Single-length Lines
2		Double-length Lines
3	⇄	Direct Connects
4	⟷	Longlines and Global Lines
5	LIM	Local Interconnect Matrix
6		Logic Cell Feedthrough Path (Contained within each Logic Cell)

Figure 11. XC5200 Interconnect Structure

Global Lines

Two pairs of horizontal and vertical global lines provide low-skew clock signals to the CLBs. Global lines are driven by low-skew buffers inside the VersaRing. The global lines provide direct input only to the CLB clock pins. The global lines also connect to the General Routing Matrix to provide access from these lines to the function generators and other control signals.

Four clock input pads at the corners of the chip, as shown in Figure 12, provide a high-speed, low-skew clock network to each of the four global-line buffers. In addition to the dedicated pad, the global lines can be sourced by internal logic. PIPs from several routing channels within the VersaRing, inside the IOI cell, can also be configured to drive the global-line buffers.

VersaRing Input/Output Interface

The VersaRing, shown in Figure 13, is positioned between the core logic and the pad ring; it has all the routing resources of a VersaBlock without the CLB logic. The VersaRing decouples the pad ring's pitch from the core's pitch. Each VersaRing Cell provides up to four pad-cell connections on one side, and connects directly to the CLB ports on the other side. Depending on placement and pad-cell pitch, any number of pad cells to a maximum of four can be connected to a VersaRing cell. Note: there are no direct connects from the Pads on top and bottom edges.

Input/Output Pad

The I/O pad, shown in Figure 14, consists of an input buffer and an output buffer. The output driver is an 8-mA full-rail CMOS buffer with 3-state control. Two slew-rate control modes are supported to minimize bus transients. Both the output buffer and the 3-state control are invertible.

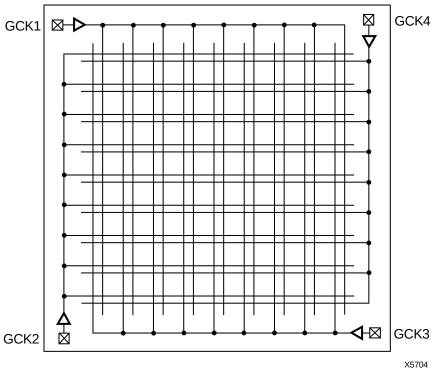


Figure 12. Global Lines

The input buffer has globally selected CMOS and TTL input thresholds. The input buffer is invertible and also provides a programmable delay line to assure reliable chip-to-chip set-up and hold times. Minimum ESD protection is 5 KV using the Human Body Model.

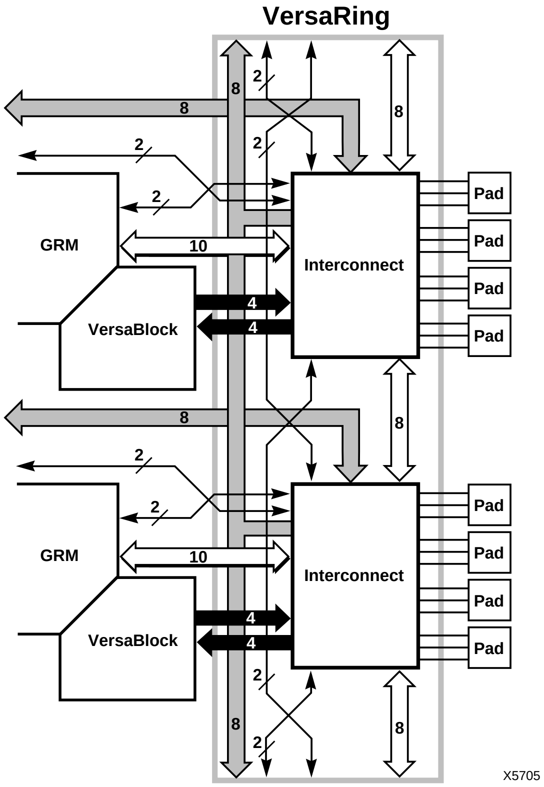


Figure 13. VersaRing I/O Interface

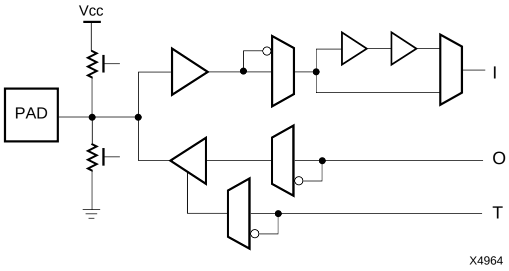


Figure 14. XC5200 I/O Block

Configuration

Configuration is the process of loading design-specific programming data into one or more LCA devices to define the functional operation of the internal blocks and their interconnections. This is somewhat like loading the command registers of a programmable peripheral chip. Each configuration bit defines the state of a static memory cell that controls either a function LUT bit, a multiplexer input, or an interconnect pass transistor. The XACT development system translates the design into a netlist file. It automatically partitions, places, and routes the logic and generates the configuration data in PROM format.

Modes

The XC5200 family has seven modes of configuration, selected by a 3-bit input code applied to the LCA mode pins (M0, M1, and M2). There are three self-clocking Master modes, two Peripheral modes, a Slave serial mode, and a new high-speed Slave parallel mode called the Express. See Table 5.

Brief descriptions of the seven modes are provided below. For details on all modes except Express, see pages 2-32 through 2-41 of the 1994 Xilinx Programmable Logic Data Book.

Master Modes

The Master modes use an internal oscillator to generate CCLK for driving potential slave devices, and to generate address and timing for external PROM(s) containing the configuration data. Master Parallel (up or down) modes generate the CCLK signal and PROM addresses, and receive byte parallel data, which is internally serialized into the LCA data-frame format. The up and down selection generates starting addresses at either zero or 3FFFF, to be compatible with different microprocessor addressing conventions. The Master Serial Mode generates CCLK and receives the configuration data in serial form from configuration data in serial form from a Xilinx serial-configuration PROM.

Peripheral Modes

The two Peripheral modes accept byte-wide data from a bus. A READY/BUSY status is available as a handshake signal. In the asynchronous mode, the internal oscillator generates a CCLK burst signal that serializes the byte-wide data. In the synchronous mode, an externally supplied clock input to CCLK serializes the data.

Slave Serial Mode

In the Slave Serial mode, the LCA device receives serial-configuration data on the rising edge of CCLK and, after loading its configuration, passes additional data out, resynchronized on the next falling edge of CCLK. Multiple slave devices with identical configurations can be wired with parallel DIN inputs so that the devices can be configured simultaneously.

Daisy Chaining

Multiple devices may be daisy-chained together so that they may be programmed using a single bitstream. The first device in the chain may be set to operate in any mode. All devices except the first device in the chain must be set to operate in Slave Serial mode.

All CCLK pins are tied together and the data chain passes from DOUT to DIN of successive devices along the chain.

Table 5. Configuration Modes

Mode	M2	M1	M0	CCLK	Data
Master Serial	0	0	0	output	Bit-Serial
Slave Serial	1	1	1	input	Bit-Serial
Master Parallel up	1	0	0	output	Byte-Wide, 00000 ↑
Master Parallel down	1	1	0	output	Byte-Wide, 3FFFF ↓
Peripheral Synchronous *	0	1	1	input	Byte-Wide
Peripheral Asynchronous	1	0	1	output	Byte-Wide
Express	0	1	0	input	Byte-Wide
Reserved	0	0	1	—	—

* Peripheral Synchronous can be considered byte-wide Slave Parallel

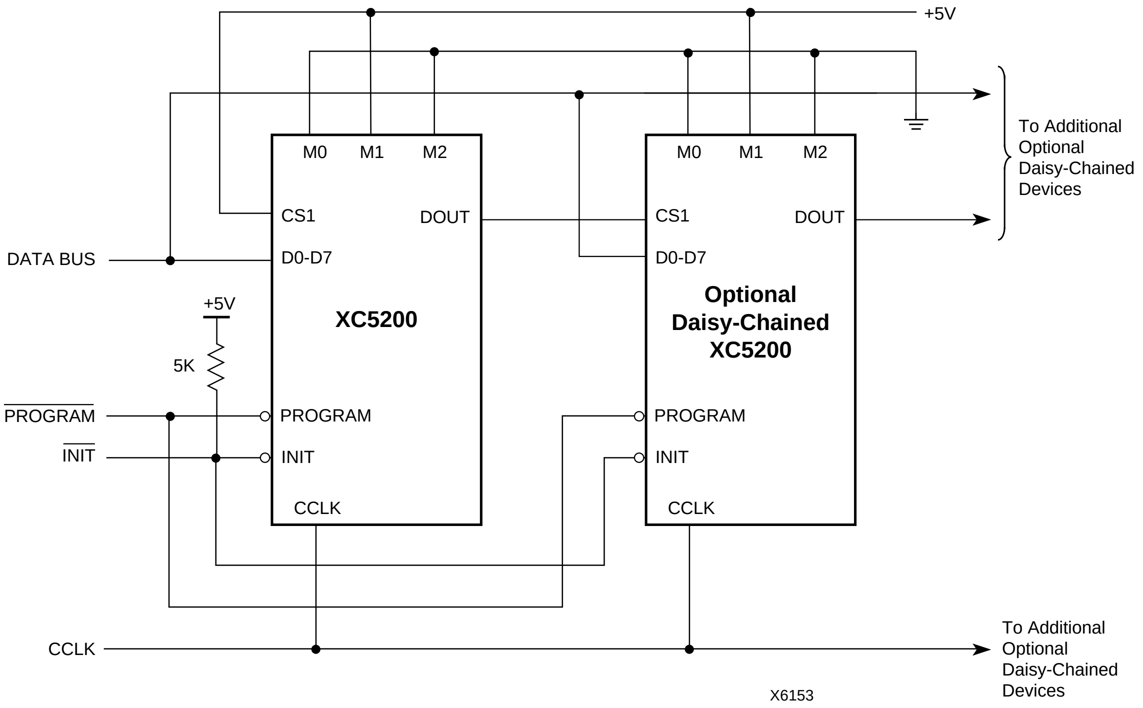


Figure 15. Express Mode

Express Mode

The Express mode (see Figure 15) is similar to the Slave serial mode, except that data is processed one byte per CCLK cycle instead of one bit per CCLK cycle. An external source is used to drive CCLK while byte-wide data is loaded directly into the configuration data shift registers. In this mode the XC5200 family is capable of supporting a CCLK frequency of 10 MHz, which is equivalent to an 80-MHz serial rate, because eight bits of configuration data are being loaded per CCLK cycle. An XC5210 in the Express mode, for instance, can be configured in about 2 ms. The Express mode does not support CRC error checking, but does support constant-field error checking.

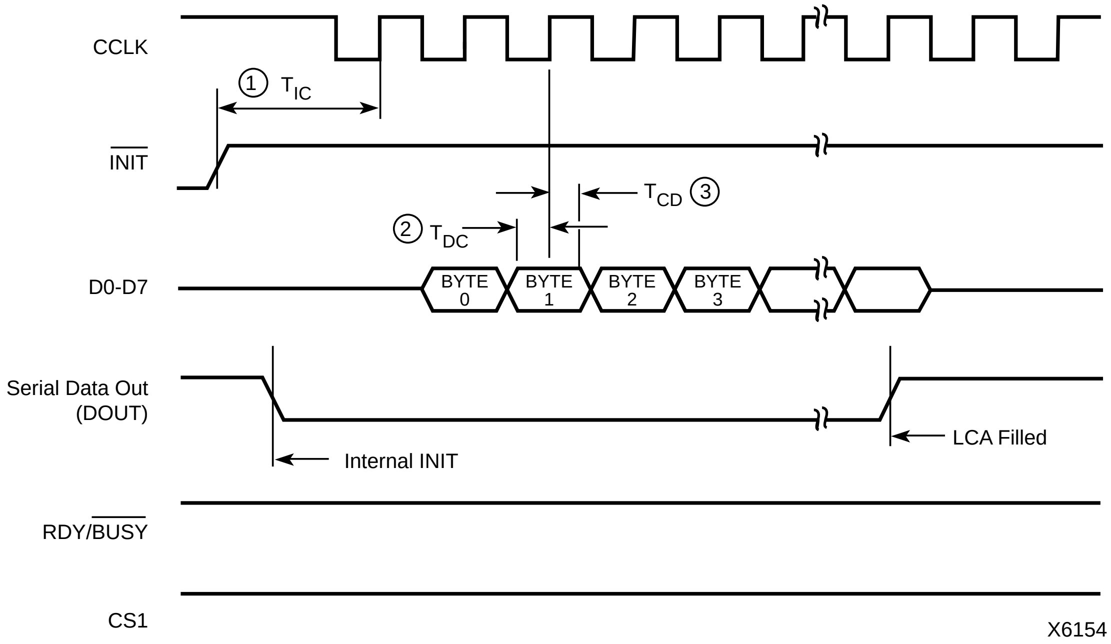
In the Express configuration mode, an external signal drives the CCLK input(s) of the LCA device(s). The first bytes of parallel configuration data must be available at the D inputs of the LCA devices a short set-up time before each rising CCLK edge. Subsequent data bytes are clocked in on each consecutive rising CCLK edge. See Figure 16.

The Express mode is only supported by the XC5200 family. It may not be used, therefore, when an XC5200 device is daisy-chained with devices from other Xilinx families.

If the first device is configured in the Express mode, additional devices may be daisy-chained only if every device in the chain is also configured in the Express mode. CCLK pins are tied together and D7-D0 pins are tied together for all devices along the chain. A status signal is passed from DOUT to CS1 of successive devices along the chain. The lead device in the chain has its CS1 input tied High (or floating, since there is an internal pull-up). All devices receive and recognize the preamble and length count, but frame data is accepted only when CS1 is High and the device's configuration memory is not already full. The status pin DOUT is pulled LOW two internal-oscillator cycles (nominally 1 MHz per cycle) after INIT is recognized as High, and remains Low until the device's configuration memory is full. Then DOUT is pulled High to signal the next device in the chain to accept the configuration data on the D7-D0 bus.

How to Delay Configuration After Power-Up

For details on how to delay configuration after power-up, refer to page 2-32 of the 1994 Xilinx Programmable Logic Data Book.



	Description	Symbol		Min	Max	Units
CCLK	$\overline{\text{INIT}}$ (High) Setup time required	1	T_{IC}	5		µs
	DIN Setup time required	2	T_{DC}	50		ns
	DIN Hold time required	3	T_{CD}	0		ns
	CCLK High time		T_{CCH}	50		ns
	CCLK Low time		T_{CCL}	50		ns
	CCLK Frequency		F_{CC}		10	MHz
				Preliminary		

Figure 16. Express Mode Programming Switching Characteristics

Format

Table 6 describes the XC5200 configuration data stream. Table 7 provides details of the internal configuration data structure.

Configuration Sequence

Figure 17 illustrates the XC5200 start-up sequence. It is described in detail in the sections below.

Clear Internal Logic

When reprogramming the XC5200 chip, a contention-free state must be reached before memory initialization can begin. In this state internal control lines sequence activities in the following order: long lines are disabled, output drivers are forced Low, and interconnect lines are discharged. Each of these operations requires one cycle of the 1-MHz Initialization clock. This sequencing is important only when reprogramming, because the contention-free state is immediately entered when configuring from a power-on state.

Table 6. Unified XC5200 Bitstream Format

Data Type	Value
Fill Byte	11111111
Preamble	11110010
Length Counter	COUNT(23:0)
Fill Byte	11111111
Start Byte	11111110
Data Frame *	DATA(N-1:0)
Cyclic Redundancy Check or Constant Field Check	CRC(3:0) or 0110
Fill Nibble	1111
Extend Write Cycle	FFFFFF
Postamble	11111110
Fill Bytes (30)	FFFF…FF

Legend:

(unshaded)	Only once per bitstream
(light)	Once per data frame
(dark)	Once per device

Table 7. Internal Configuration Data Structure

Device	VersaBlock Array	PROM Size (bits)	Xilinx Serial Prom Needed
XC5202	8 x 8	42,448	XC1765
XC5204	10 x 12	70,736	XC1728
XC5206	14 x 14	106,320	XC17128
XC5210	18 x 18	165,520	XC17256
XC5215	22 x 22	237,776	XC17256

Bits per Frame = (34 x number of Rows) + 28 for the top + 28 for the bottom + 4 splitter bits + 8 start bits + 8 error check bits + 4 fill bits + 4 extended write bits

Number of Frames = (12 x number of Columns) + 7 for the left edge + 8 for the right edge + 1 splitter bit

Program Data = (Bits per Frame x Number of Frames) + 48 header bits + 8 postamble bits + 280 fill bits

PROM Size = Program Data

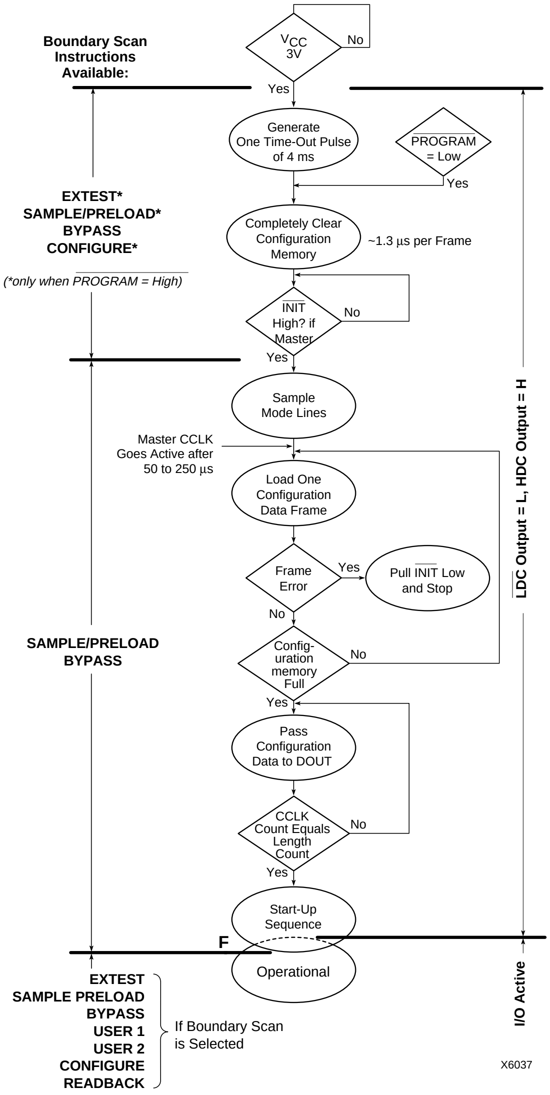

Boundary Scan Instructions Available:
VCC 3V
No
Yes
Generate One Time-Out Pulse of 4 ms
PROGRAM = Low
Yes
EXTEST*
SAMPLE/PRELOAD*
BYPASS
CONFIGURE*
(*only when PROGRAM = High)
Completely Clear Configuration Memory
~1.3 µs per Frame
INIT High? if Master
No
Yes
Sample Mode Lines
Master CCLK Goes Active after 50 to 250 µs
Load One Configuration Data Frame
Frame Error
Yes
Pull INIT Low and Stop
No
SAMPLE/PRELOAD
BYPASS
Config-uration memory Full
No
Yes
Pass Configuration Data to DOUT
CCLK Count Equals Length Count
No
Yes
Start-Up Sequence
F
Operational
LDC Output = L, HDC Output = H
I/O Active
EXTEST
SAMPLE PRELOAD
BYPASS
USER 1
USER 2
CONFIGURE
READBACK
If Boundary Scan is Selected
X6037

Figure 17. Start-up Sequence

Clear Address Registers

During this phase the configuration address registers are cleared to ensure that they will contain at most a single token at all times. Prior to memory initialization, the XC5200 device eliminates the possibility of multiple tokens within the address register, as is typically the case when powering on.

Power-On Time-Out

An internal power-on reset circuit is triggered when power is applied. When V_{CC} reaches the voltage at which portions of the LCA begin to operate (i.e., performs a write-and-read test of a sample pair of configuration memory bits), the programmable I/O buffers are 3-stated with active high-impedance pull-up resistors. A time-out delay — nominally 4 ms — is initiated to allow the power-supply voltage to stabilize. For correct operation the power supply must reach V_{CC}(min) by the end of the time-out, and must not dip below it thereafter.

There is no distinction between master and slave modes with regard to the time-out delay. Instead, the $\overline{\text{INIT}}$ line is used to ensure that all daisy-chained devices have completed initialization. Since XC2000 devices do not have this signal, extra care must be taken to guarantee proper operation when daisy-chaining them with XC5200 devices. For proper operation with XC3000 devices, the $\overline{\text{RESET}}$ signal, which is used in XC3000 to delay configuration, should be connected to $\overline{\text{INIT}}$.

If the time-out delay is insufficient, configuration should be delayed by holding the $\overline{\text{INIT}}$ pin Low until the power supply has reached operating levels.

During all three phases — Power-on, Initialization, and Configuration — DONE is held Low; HDC, $\overline{\text{LDC}}$, and $\overline{\text{INIT}}$ are active; DOUT is driven; and all I/O buffers are disabled.

Initialization

This phase clears the configuration memory and establishes the configuration mode.

The configuration memory is cleared at the rate of one frame per internal clock cycle (nominally 1 MHz). An open-drain bidirectional signal, $\overline{\text{INIT}}$, is released when the configuration memory is completely cleared. The device then tests for the absence of an external active-low level on $\overline{\text{INIT}}$. The mode lines are sampled two internal clock cycles later (nominally 2 µs).

The master device waits an additional 32 µs to 256 µs (nominally 64-128 µs) to provide adequate time for all of the slave devices to recognize the release of $\overline{\text{INIT}}$ as well. Then the master device enters the Configuration phase.

Configuration

The length counter begins counting immediately upon entry into the configuration state. In slave-mode operation it is important to wait at least two cycles of the internal 1-MHz clock oscillator after $\overline{\text{INIT}}$ is recognized before toggling CCLK and feeding the serial bitstream. Configuration will not begin until the internal configuration logic reset is released, which happens two cycles after $\overline{\text{INIT}}$ goes High. A master device's configuration is delayed from 32 to 256 µs to ensure proper operation with any slave devices driven by the master device.

A preamble field at the beginning of the configuration data stream indicates that the next 24 bits represent the length count. The length count equals the total number of configuration bits needed to load the complete configuration data to all daisy-chained devices. Once the preamble and length-count values have been passed through to the next device in the daisy-chain, DOUT is held High to prevent start bits from reaching any daisy-chained devices. After fully configuring itself, the device passes serial data to downstream daisy-chained devices via DOUT until the full length count is reached.

Errors in the configuration bitstream are checked at the end of a frame of data. The device does not check the preamble or length count for errors. In a daisy-chained configuration, configuration data for downstream devices are not checked for errors. If an error is detected after reading a frame, the $\overline{\text{ERR}}$ pin (also known as $\overline{\text{INIT}}$) is immediately pulled Low and all configuration activity ceases. However, a master or Peripheral Asynchronous device will continue outputting a configuration clock and incrementing the PROM address indefinitely even though it will never complete configuration. A reprogram or power-on must be applied to remove the device from this state.

Start-Up and Operation

The XC5200 start-up sequence is identical to that of the XC4000 family. Each of these events may occur in any order: (a) DONE is pulled High; and/or (b) user I/Os become active; and/or (c) Internal Reset is deactivated. As a configuration option, the three events may be triggered by a user clock rather than by CCLK, or the start-up sequence may be delayed by externally holding the DONE pin Low.

In any mode, the clock cycles of the start-up sequence hare not included in the length count. The length of the bitstream is greater than the length count.

Pin Functions During Configuration

CONFIGURATION MODE: <M2:M1:M0>						**USER OPERATION**
SLAVE <1:1:1>	**MASTER-SER <0:0:0>**	**SYN.PERIPH <0:1:1>**	**ASYN.PERIPH <1:0:1>**	**MASTER-HIGH <1:1:0>**	**MASTER-LOW <1:0:0>**	
				A16	A16	GCK1-I/O
				A17	A17	I/O
TDI	TDI	TDI	TDI	TDI	TDI	TDI-I/O
TCK	TCK	TCK	TCK	TCK	TCK	TCK-I/O
TMS	TMS	TMS	TMS	TMS	TMS	TMS-I/O
						I/O
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	I/O
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	I/O
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	I/O
						GCK2-I/O
HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	I/O
LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	I/O
INIT-ERROR	INIT-ERROR	INIT-ERROR	INIT-ERROR	INIT-ERROR	INIT-ERROR	I/O
						I/O
DONE	DONE	DONE	DONE	DONE	DONE	DONE
PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	I/O
						GCK3-I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	I/O
			CSO (I)			I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	I/O
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	I/O
			RS (I)			I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	I/O
		RDY/BUSY	RDY/BUSY	RCLK	RCLK	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	DOUT	I/O
CCLK (I)	CCLK (O)	CCLK (I)	CCLK (O)	CCLK (O)	CCLK (O)	CCLK (I)
TDO	TDO	TDO	TDO	TDO	TDO	TDO-I/O
			WS (I)	A0	A0	I/O
				A1	A1	GCK4-I/O
			CS1 (I)	A2	A2	I/O
				A3	A3	I/O
				A4	A4	I/O
				A5	A5	I/O
				A6	A6	I/O
				A7	A7	I/O
				A8	A8	I/O
				A9	A9	I/O
				A10	A10	I/O
				A11	A11	I/O
				A12	A12	I/O
				A13	A13	I/O
				A14	A14	I/O
				A15	A15	I/O
						ALL OTHERS

▢ Represents a 50-kΩ to 100-kΩ pull-up before and during configuration

* INIT is an open-drain output during configuration

(I) Represents an input (O) Represents an output

Before and during configuration, all outputs that are not used for the configuration process are 3-stated with a 50-kΩ to 100-kΩ pull-up resistor.

Pin Descriptions

Permanently Dedicated Pins

V_{CC}

Eight or more (depending on package type) connections to the nominal +5-V supply voltage. All must be connected.

GND

Eight or more (depending on package type) connections to ground. All must be connected.

CCLK

During configuration, Configuration Clock is an output of the LCA in master modes or Asynchronous Peripheral mode, but is an input to the LCA in Slave Serial mode and Synchronous Peripheral mode.

After configuration, CCLK has a weak pull-up resistor and can be selected as Readback Clock.

DONE

This is a bidirectional signal with optional pull-up resistor.

As an output, it indicates the completion of the configuration process. The configuration program determines the exact timing, the clock source for the Low-to-High transition, and enable of the pull-up resistor.

As an input, a Low level on DONE can be configured to delay the global logic initialization or the enabling of outputs.

$\overline{\text{PROGRAM}}$

This is an active-Low input, held Low during configuration, that forces the LCA to clear its configuration memory.

When $\overline{\text{PROGRAM}}$ goes High, the LCA executes a complete clear cycle, before it goes into a WAIT state and releases $\overline{\text{INIT}}$. After configuration, it has an optional pull-up resistor.

User I/O Pins That Can Have Special Functions

RDY/$\overline{\text{BUSY}}$

During peripheral modes, this pin indicates when it is appropriate to write another byte of data into the LCA device. The same status is also available on D7 in Asynchronous Peripheral mode, if a read operation is performed when the device is selected. After configuration, this is a user-programmable I/O pin.

$\overline{\text{RCLK}}$

During Master Parallel configuration, each change on the A0-15 outputs is preceded by a rising edge on $\overline{\text{RCLK}}$, a redundant output signal. After configuration, this is a user-programmable I/O pin.

M0, M1, M2

As mode inputs, these pins are sampled before the start of configuration to determine the configuration mode to be used.

After configuration, M0, M1, and M2 become user-programmable I/O.

TDO

If boundary scan is used, this is the Test Data Output.

If boundary scan is not used, this pin becomes user-programmable I/O.

TDI, TCK, TMS

If boundary scan is used, these pins are Test Data In, Test Clock, and Test Mode Select inputs, respectively, coming directly from the pads, bypassing the IOBs. These pins can also be used as inputs to the CLB logic after configuration is completed.

If the boundary scan option is not selected, all boundary scan functions are inhibited once configuration is completed. These pins become user-programmable I/O.

HDC

High During Configuration is driven High until configuration is completed. It is available as a control output indicating that configuration is not yet completed. After configuration, this is a user-programmable I/O pin.

Pin Descriptions

$\overline{\text{LDC}}$

Low During Configuration is driven Low until configuration completes. It is available as a control output indicating that configuration is not yet completed. After configuration, this is a user-programmable I/O pin.

$\overline{\text{INIT}}$

Before and during configuration, this is a bidirectional signal. An external pull-up resistor is recommended.

As an active-Low open-drain output, $\overline{\text{INIT}}$ is held Low during the power stabilization and internal clearing of the configuration memory. As an active-Low input, it can be used to hold the LCA device in the internal WAIT state before the start of configuration. Master-mode devices stay in a WAIT state an additional 30 to 300 µs after $\overline{\text{INIT}}$ has gone High.

During configuration, a Low on this output indicates that a configuration data error has occurred. After configuration, this is a user-programmable I/O pin.

GCK1 - GCK4

Four Global Inputs each drive a dedicated internal global net with short delay and minimal skew. If not used for this purpose, any of these pins is a user-programmable I/O pin.

$\overline{\text{CS0}}$, CS1, $\overline{\text{WS}}$, $\overline{\text{RS}}$

These four inputs are used in peripheral modes. The chip is selected when $\overline{\text{CS0}}$ is Low and CS1 is High. While the chip is selected, a Low on Write Strobe ($\overline{\text{WS}}$) loads the data present on the D0 - D7 inputs into the internal data buffer; a Low on Read Strobe ($\overline{\text{RS}}$) changes D7 into a status output: High if Ready, Low if Busy, and D0…D6 are active Low. $\overline{\text{WS}}$ and $\overline{\text{RS}}$ should be mutually exclusive, but if both are Low simultaneously, the Write Strobe overrides. After configuration, these are user-programmable I/O pins.

A0 - A17

During Master Parallel mode, these 18 output pins address the configuration EPROM. After configuration, these are user-programmable I/O pins.

D0 - D7

During Master Parallel and peripheral configuration modes, these eight input pins receive configuration data. After configuration, they are user-programmable I/O pins.

DIN

During Slave Serial or Master Serial configuration modes, this is the serial configuration data input receiving data on the rising edge of CCLK.

During parallel configuration modes, this is the D0 input. After configuration, DIN is a user-programmable I/O pin.

DOUT

During configuration in any mode, this is the serial configuration data output that can drive the DIN of daisy-chained slave LCA devices. DOUT data changes on the falling edge of CCLK, 1.5 CCLK periods after it was received at the DIN input. After configuration, DOUT is a user-programmable I/O pins.

Unrestricted User-Programmable I/O Pins

I/O

A pin that can be configured to be input and/or output after configuration is completed. Before configuration is completed, these pins have an internal high-value pull-up resistor that defines the logical level as High.

Before and during configuration, all outputs that are not used for the configuration process are 3-stated with a 50-kΩ to 100-kΩ pull-up resistor.

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	-0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	-0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	-0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	-65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in. = 1.5 mm)	+260	°C
T_J	Junction temperature in plastic packages	+125	°C
	Junction temperature in ceramic packages	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Description	Min	Max	Units
V_{CC}	Supply voltage relative to GND Commercial: 0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial: -40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND Military: -55°C to 125°C	4.5	5.5	V
V_{IHT}	High-level input voltage — TTL configuration	2.0	V_{CC}	V
V_{ILT}	Low-level input voltage — TTL configuration	0	0.8	V
V_{IHC}	High-level input voltage — CMOS configuration	70%	100%	V_{CC}
V_{ILC}	Low-level input voltage — CMOS configuration	0	20%	V_{CC}
T_{IN}	Input signal transition time		250	ns

DC Characteristics Over Operating Conditions

Symbol	Description	Min	Max	Units
V_{OH}	High-level output voltage @ I_{OH} = -8.0 mA, V_{CC} min	3.86		V
V_{OL}	Low-level output voltage @ I_{OL} = 8.0 mA, V_{CC} max (Note 1)		0.4	V
I_{CCO}	Quiescent LCA supply current (Note 1)		15	mA
I_{IL}	Leakage current	-10	+10	µA
C_{IN}	Input capacitance (sample tested)		15	pF
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0V (sample tested)	0.02	0.25	mA

Note: 1. With no output current loads, all package pins at V_{CC} or GND, and the LCA configured with a MakeBits tie option.

Global Buffer Switching Characteristic Guidelines

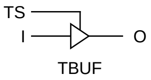
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		**-6**	**-5**	**-4**
Description	**Symbol**	**Device**	**Max (ns)**	**Max (ns)**	**Max (ns)**
Global Signal Distribution					
From pad through global buffer, to any clock (CK)	T_{BUFG}	XC5202			
		XC5204			
		XC5206	9.4	8.8	
		XC5210	9.4	8.8	
		XC5215			
Internal Clock to Output Pad Delay					
From clock (CK) to output pad (fast), using direct connect between Q and output (O)	T_{OKPOF}	XC5202			
		XC5204			
		XC5206	9.9	8.9	
		XC5210	9.9	8.9	
		XC5215			
From clock (CK) to output pad (slew-limited), using direct connect between Q and output (O)	T_{OKPOS}	XC5202			
		XC5204			
		XC5206	14.8	12.7	
		XC5210	14.8	12.7	
		XC5215			

Note: 1. Die-size-dependent parameters are based upon XC5210 characterization. Production specifications will vary with array size.

Longline Switching Characteristic Guidelines

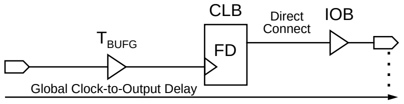
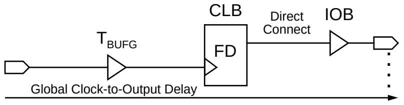
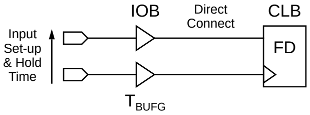
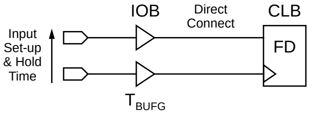
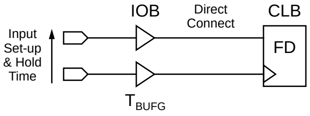
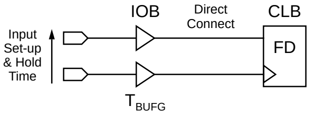
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-6	-5	-4
Description	**Symbol**	**Device**	**Max (ns)**	**Max (ns)**	**Max (ns)**
TBUF driving a Longline 					
I to Longline, while TS is Low; i.e., buffer is constantly active	T_{IO}	XC5202			
		XC5204			
		XC5206	4.0	3.6	
		XC5210	4.0	3.6	
		XC5215			
TS going Low to Longline going from floating High or Low to active Low or High	T_{ON}	XC5202			
		XC5204			
		XC5206	5.3	4.8	
		XC5210	5.3	4.8	
		XC5215			
TS going High to TBUF going inactive, not driving Longline	T_{OFF}	XC5202			
		XC5204			
		XC5206	2.4	2.2	
		XC5210	2.4	2.2	
		XC5215			

Note: 1. Die-size-dependent parameters are based upon XC5210 characterization. Production specifications will vary with array size.

Guaranteed Input and Output Parameters (Pin-to-Pin)

All values listed below are tested directly, and guaranteed over the operating conditions. The same parameters can also be derived indirectly from the Global Buffer specifications. The XACT delay calculator uses this indirect method, and may overestimate because of worst-case assumptions. When there is a discrepancy between these two methods, the values listed below should be used, and the derived values should be considered conservative overestimates.

	Speed Grade		-6		-5		-4
Description	**Symbol**	**Device**	**Max (ns)**		**Max (ns)**		**Max (ns)**
Global Clock to Output Pad (fast)  CLB Direct Connect IOB T_{BUFG} FD Global Clock-to-Output Delay	T_{ICKOF} (Max)	XC5202					
		XC5204					
		XC5206	17.2		15.4		
		XC5210	17.2		15.4		
		XC5215					
Global Clock to Output Pad (slew-limited)  CLB Direct Connect IOB T_{BUFG} FD Global Clock-to-Output Delay	T_{ICKO} (Max)	XC5202					
		XC5204					
		XC5206	21.7		19.0		
		XC5210	21.7		19.0		
		XC5215					
Input Set-up Time (no delay) to CLB Flip-Flop  IOB Direct Connect CLB Input Set-up & Hold Time FD T_{BUFG}	T_{PSUF} (Min)	XC5202					
		XC5204					
		XC5206	1.2		0.5		
		XC5210	1.2		0.5		
		XC5215					
Input Hold Time (no delay) to CLB Flip-Flop  IOB Direct Connect CLB Input Set-up & Hold Time FD T_{BUFG}	T_{PHF} (Min)	XC5202					
		XC5204					
		XC5206	3.0		2.5		
		XC5210	3.0		2.5		
		XC5215					
Input Set-up Time (with delay) to CLB Flip-Flop  IOB Direct Connect CLB Input Set-up & Hold Time FD T_{BUFG}	T_{PSU} (Min)	XC5202					
		XC5204					
		XC5206	7.5		6.4		
		XC5210	7.5		6.4		
		XC5215					
Input Hold Time (with delay) to CLB Flip-Flop  IOB Direct Connect CLB Input Set-up & Hold Time FD T_{BUFG}	T_{PH} (Min)	XC5202					
		XC5204					
		XC5206	0		0		
		XC5210	0		0		
		XC5215					

Note:
1. These measurements assume that the flip-flop has a direct connect to or from the IOB. XACT-Performance can be used to assure that direct connects are used.
2. When testing outputs (fast or slew-limited), half of the outputs on one side of the device are switching.
3. Die-size-dependent parameters are based upon XC5210 characterization. Production specifications will vary with array size.

IOB Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

Speed Grade		**-6**	**-5**	**-4**
Description	**Symbol**	**Max (ns)**	**Max (ns)**	**Max (ns)**
Input				
Propagation Delays from CMOS or TTL Levels				
Pad to I (no delay)	T_{PI}	5.4	4.9	
Pad to I (with delay)	T_{PID}	11.1	10.2	
Output				
Propagation Delays to CMOS or TTL Levels				
Output (O) to Pad (fast)	T_{OPF}	4.6	4.5	
Output (O) to Pad (slew-limited)	T_{OPS}	9.4	8.3	
3-state to Pad active (fast)	T_{TSONF}	6.9	6.6	
3-state to Pad active (slew-limited)	T_{TSONS}	11.6	10.4	
Internal GTS to Pad active (fast)	T_{GTSF}	17.7	15.9	
Internal GTS to Pad active (slew-limited)	T_{GTSS}	22.3	19.7	

Note: 1. Timing is measured at pin threshold, with 50-pF external capacitance loads. **Slew-limited** output rise/fall times are approximately two times longer than **fast** output rise/fall times. For the effect of capacitive loads on ground bounce, see pages 8-8 through 8-10 of the 1994 Xilinx Programmable Logic Data Book.

2. Unused and unbonded IOBs are configured by default as inputs with internal pull-up resistors.

CLB Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade	**-6**		**-5**		**-4**	
Description	**Symbol**	**Min (ns)**	**Max (ns)**	**Min (ns)**	**Max (ns)**	**Min (ns)**	**Max (ns)**
Combinatorial Delays							
F inputs to X output	T_{ILO}		5.5		4.5		
DI inputs to DO output (Logic-Cell Feedthrough)	T_{IDO}		4.2		3.3		
F inputs via F5_MUX to DO output	T_{IMO}		7.1		5.7		
Carry Delays							
Incremental delay per bit	T_{CY}		0.7		0.6		
Carry-in overhead from DI	T_{CYDI}		1.7		1.5		
Carry-in overhead from F	T_{CYL}		3.6		3.2		
Carry-out overhead to DO	T_{CYO}		3.9		3.1		
Sequential Delays							
Clock (CK) to out (Q) (Flip-Flop)	T_{CKO}		5.4		4.4		
Gate (Latch enable) going active to out (Q)	T_{GO}		8.6		6.8		
Set-up Time Before Clock (CK)							
F inputs	T_{ICK}	2.1		1.5			
F inputs via F5_MUX	T_{MICK}	3.6		2.7			
DI input	T_{DICK}	0.5		0.3			
CE input	T_{EICK}	1.2		0.9			
Hold Times After Clock (CK)							
F inputs	T_{CKI}	0		0			
F inputs via F5_MUX	T_{CKMI}	0		0			
DI input	T_{CKDI}	0		0			
CE input	T_{CKEI}	0		0			
Clock Widths							
Clock High Time	T_{CH}	6.0		6.0			
Clock Low Time	T_{CL}	6.0		6.0			
Reset Delays							
Width (High)	T_{CLRW}	6.0		6.0			
Delay from CLR to Q (Flip-Flop)	T_{CLR}		7.3		5.8		
Delay from CLR to Q (Latch)	T_{CLRL}		6.1		4.8		
Global Reset Delays (see Note 2)							
Width (High)	T_{GCLRW}	6.0		6.0			
Delay from internal GCLR to Q	T_{GCLR}		12.4		10.2		

Note: 1. The CLB K to Q output delay (T_{CKO}) of any CLB, plus the shortest possible interconnect delay, is always longer than the Data In hold-time requirement (T_{CKDI}) of any CLB on the same die.

2. Timing is based upon the XC5210 device. For other devices, see XACT Timing Calculator.

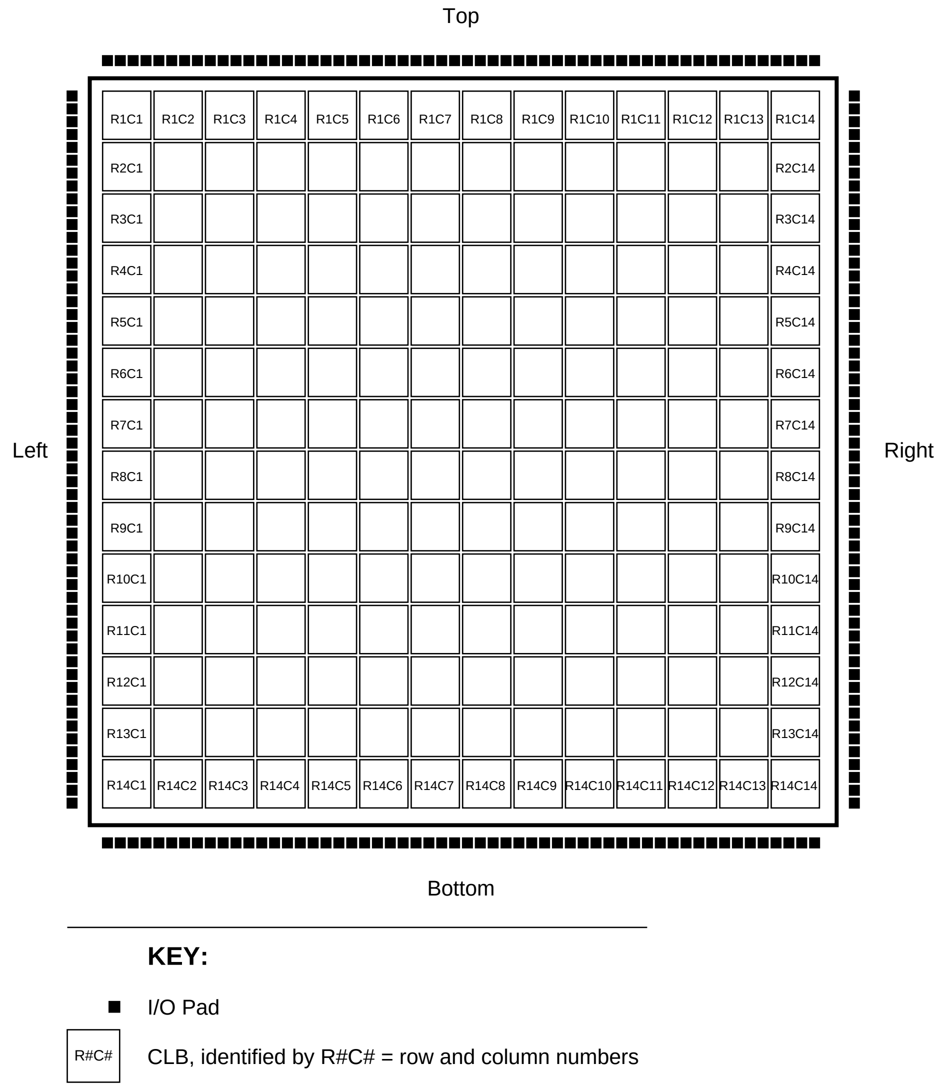
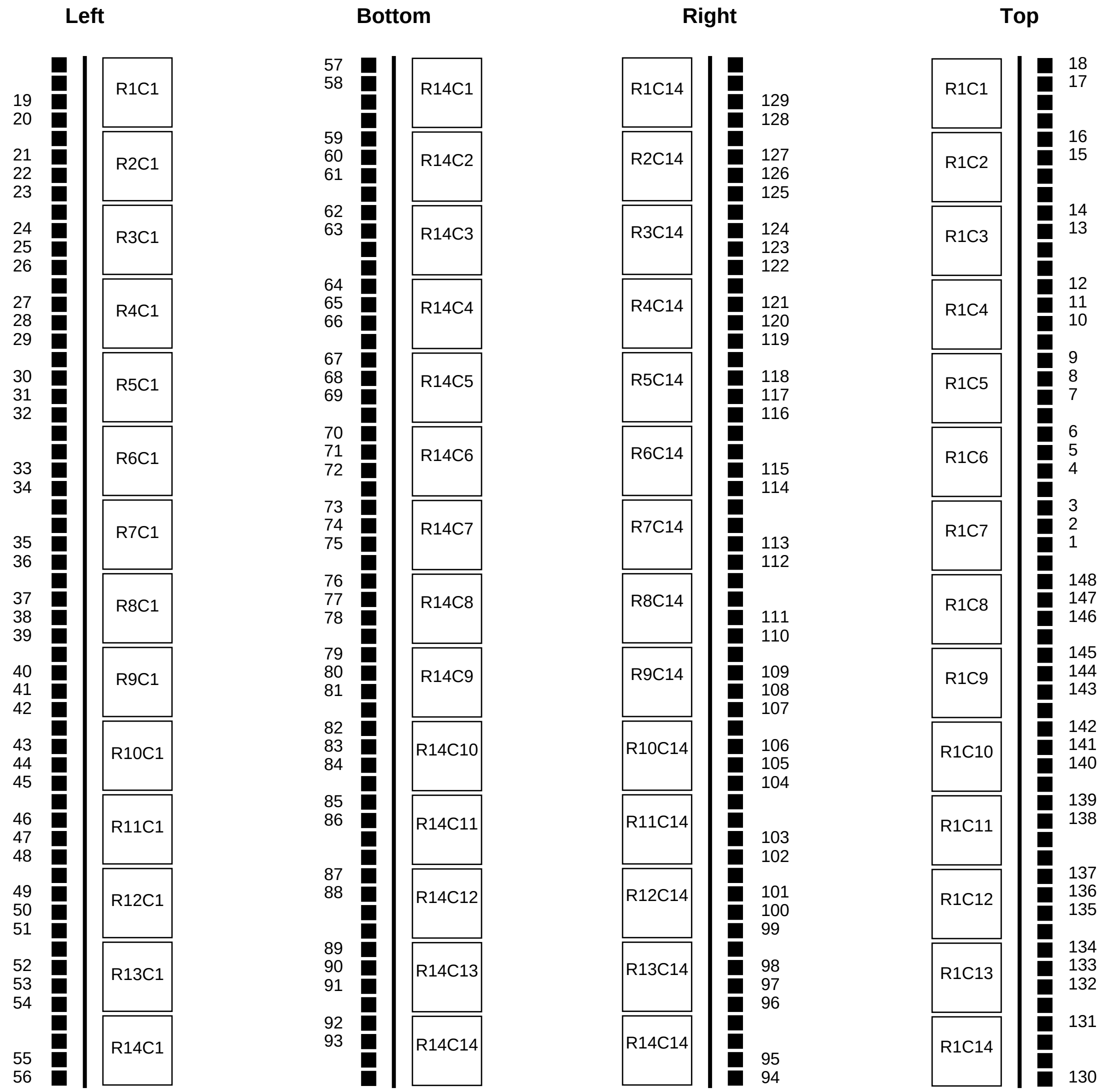
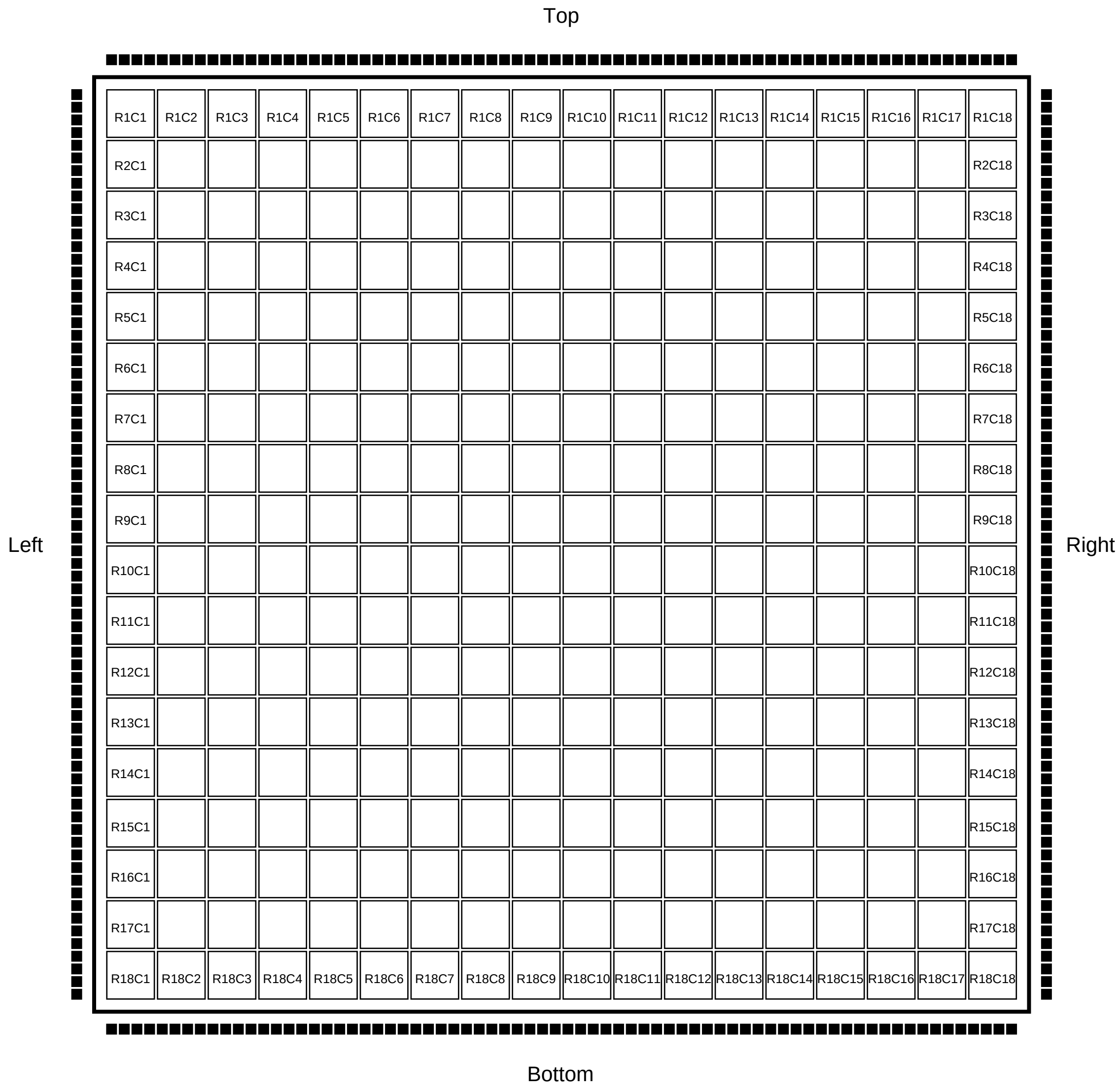

Top
R1C1 R1C2 R1C3 R1C4 R1C5 R1C6 R1C7 R1C8 R1C9 R1C10 R1C11 R1C12 R1C13 R1C14
R2C1 R2C14
R3C1 R3C14
R4C1 R4C14
R5C1 R5C14
R6C1 R6C14
R7C1 R7C14
Left
Right
R8C1 R8C14
R9C1 R9C14
R10C1 R10C14
R11C1 R11C14
R12C1 R12C14
R13C1 R13C14
R14C1 R14C2 R14C3 R14C4 R14C5 R14C6 R14C7 R14C8 R14C9 R14C10 R14C11 R14C12 R14C13 R14C14
Bottom
KEY:
■ I/O Pad
R#C# CLB, identified by R#C# = row and column numbers

Figure 18. XC5206 CLB-to-Pad Relationship



Note: Pad numbers (1, 2, …, 148) refer to die pads, not external device pins. Also see the XC5206 pinout table on pages 36-37.

Figure 19. XC5206 CLB-to-Pad Relationship (Detail)

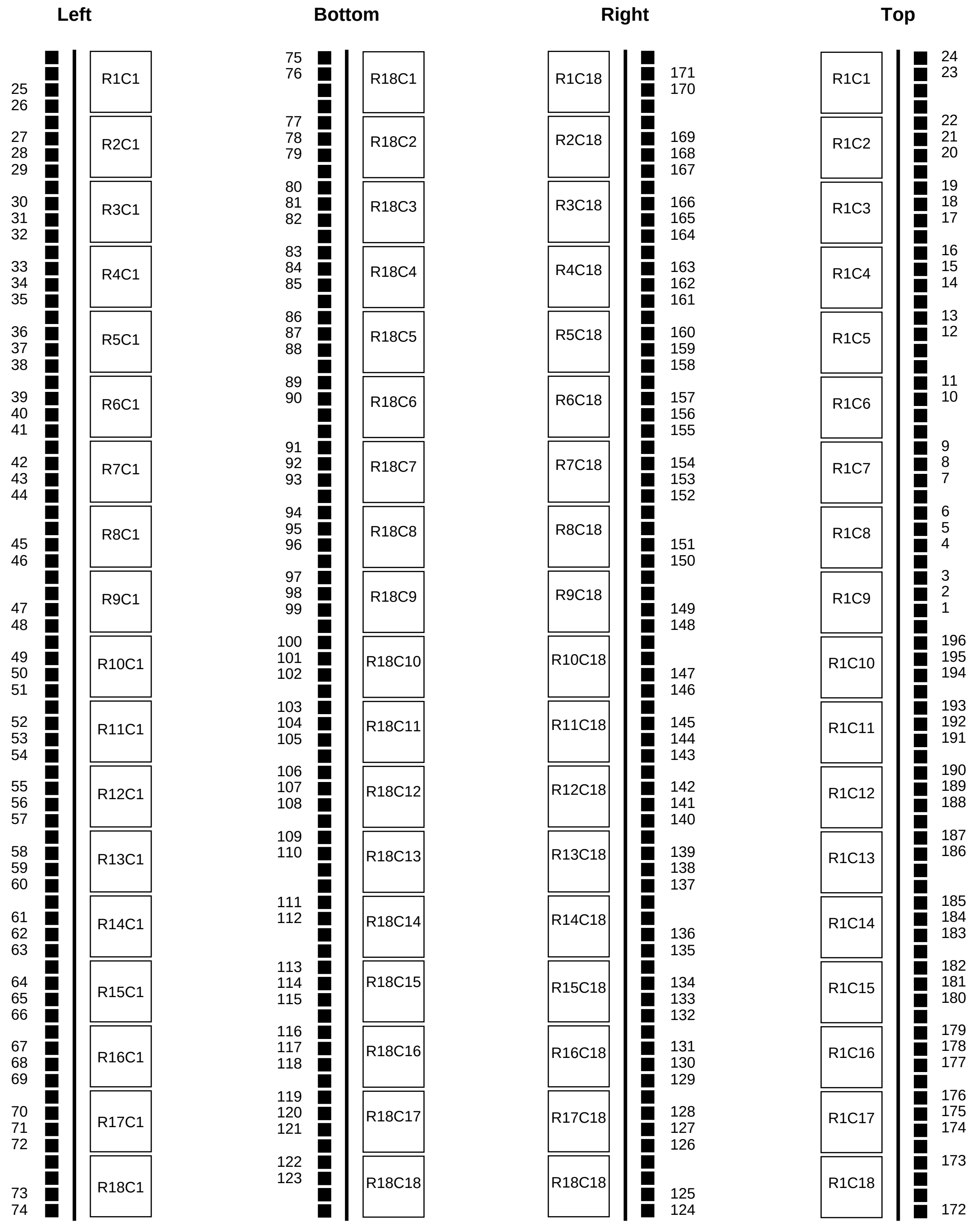


KEY:

■ I/O Pad

R#C# CLB, identified by R#C# = row and column numbers

Figure 20. XC5210 CLB-to-Pad Relationship



Note: Pad numbers (1, 2, …, 196) refer to die pads, not external device pins. Also see the XC5210 pinout table on pages 38-39.

Figure 21. XC5210 CLB-to-Pad Relationship (Detail)

XC5206 Pinouts

Pin Description †	PC84	PQ160	PQ208	PG191	Boundary Scan Order
VCC	2	142	183	J4	-
1. I/O (A8)	3	143	184	J3	87
2. I/O (A9)	4	144	185	J2	90
3. I/O	-	145	186	J1	93
4. I/O	-	146	187	H1	99
5. I/O	-	-	188	H2	102
6. I/O	-	-	189	H3	105
7. I/O (A10)	5	147	190	G1	111
8. I/O (A11)	6	148	191	G2	114
9. I/O	-	149	192	F1	117
10. I/O	-	150	193	E1	123
GND	-	151	194	G3	-
-	-	-	195*	-	-
-	-	-	196*	-	-
11. I/O	-	152	197	C1	126
12. I/O	-	153	198	E2	129
13. I/O (A12)	7	154	199	F3	138
14. I/O (A13)	8	155	200	D2	141
15. I/O	-	156	201	B1	150
16. I/O	-	157	202	E3	153
17. I/O (A14)	9	158	203	C2	162
18. I/O (A15)	10	159	204	B2	165
VCC	11	160	205	D3	-
-	-	-	206*	-	-
-	-	-	207*	-	-
-	-	-	208*	-	-
-	-	-	1*	-	-
GND	12	1	2	D4	-
-	-	-	3*	-	-
19. GCK1 (A16, I/O)	13	2	4	C3	174
20. I/O (A17)	14	3	5	C4	177
21. I/O	-	4	6	B3	183
22. I/O	-	5	7	C5	186
23. I/O (TDI)	15	6	8	A2	189
24. I/O (TCK)	16	7	9	B4	195
25. I/O	-	8	10	C6	198
26. I/O	-	9	11	A3	201
-	-	-	12*	-	-
-	-	-	13*	-	-
GND	-	10	14	C7	-
27. I/O	-	11	15	A4	207
28. I/O	-	12	16	A5	210
29. I/O (TMS)	17	13	17	B7	213
30. I/O	18	14	18	A6	219
31. I/O	-	-	19	C8	222
32. I/O	-	-	20	A7	225
33. I/O	-	15	21	B8	234
34. I/O	-	16	22	A8	237
35. I/O	19	17	23	B9	246
36. I/O	20	18	24	C9	249
GND	21	19	25	D9	-
VCC	22	20	26	D10	-

* Indicates unconnected package pins.

Pin Description †	PC84	PQ160	PQ208	PG191	Boundary Scan Order
37. I/O	23	21	27	C10	255
38. I/O	24	22	28	B10	258
39. I/O	-	23	29	A9	261
40. I/O	-	24	30	A10	267
41. I/O	-	-	31	A11	270
42. I/O	-	-	32	C11	273
43. I/O	25	25	33	B11	279
44. I/O	26	26	34	A12	282
45. I/O	-	27	35	B12	285
46. I/O	-	28	36	A13	291
GND	-	29	37	C12	-
-	-	-	38*	-	-
-	-	-	39*	-	-
47. I/O	-	30	40	A15	294
48. I/O	-	31	41	C13	297
49. I/O	27	32	42	B14	303
50. I/O	-	33	43	A16	306
51. I/O	-	34	44	B15	309
52. I/O	-	35	45	C14	315
53. I/O	28	36	46	A17	318
54. I/O	29	37	47	B16	321
55. M1 (I/O)	30	38	48	C15	330
GND	31	39	49	D15	-
56. M0 (I/O)	32	40	50	A18	333
-	-	-	51*	-	-
-	-	-	52*	-	-
-	-	-	53*	-	-
-	-	-	54*	-	-
VCC	33	41	55	D16	-
57. M2 (I/O)	34	42	56	C16	336
58. GCK2 (I/O)	35	43	57	B17	339
59. I/O (HDC)	36	44	58	E16	348
60. I/O	-	45	59	C17	351
61. I/O	-	46	60	D17	354
62. I/O	-	47	61	B18	360
63. I/O (LDC)	37	48	62	E17	363
64. I/O	-	49	63	F16	372
65. I/O	-	50	64	C18	375
GND	-	51	67	G16	-
66. I/O	-	52	68	E18	378
67. I/O	-	53	69	F18	384
68. I/O	38	54	70	G17	387
69. I/O	39	55	71	G18	390
70. I/O	-	-	72	H16	396
71. I/O	-	-	73	H17	399
72. I/O	-	56	74	H18	402
73. I/O	-	57	75	J18	408
74. I/O	40	58	76	J17	411
75. I/O ($\overline{\text{ERR}}$, $\overline{\text{INIT}}$)	41	59	77	J16	414
VCC	42	60	78	J15	-
GND	43	61	79	K15	-
76. I/O	44	62	80	K16	420

† Leading number refers to bonded pad, shown in Figure 19.

XC5206 Pinouts (continued)

Pin Description †	PC84	PQ160	PQ208	PG191	Boundary Scan Order
77. I/O	45	63	81	K17	423
78. I/O	-	64	82	K18	426
79. I/O	-	65	83	L18	432
80. I/O	-	-	84	L17	435
81. I/O	-	-	85	L16	438
82. I/O	46	66	86	M18	444
83. I/O	47	67	87	M17	447
84. I/O	-	68	88	N18	450
85. I/O	-	69	89	P18	456
GND	-	70	90	M16	-
-	-	-	91*	-	-
-	-	-	92*	-	-
86. I/O	-	71	93	T18	459
87. I/O	-	72	94	P17	468
88. I/O	48	73	95	N16	471
89. I/O	49	74	96	T17	480
90. I/O	-	75	97	R17	483
91. I/O	-	76	98	P16	486
92. I/O	50	77	99	U18	492
93. I/O	51	78	100	T16	495
GND	52	79	101	R16	-
-	-	-	102*	-	-
DONE	53	80	103	U17	-
-	-	-	104*	-	-
-	-	-	105*	-	-
VCC	54	81	106	R15	-
-	-	-	107*	-	-
PROG	55	82	108	V18	-
94. I/O (D7)	56	83	109	T15	504
95. GCK3 (I/O)	57	84	110	U16	507
96. I/O	-	85	111	T14	516
97. I/O	-	86	112	U15	519
98. I/O (D6)	58	87	113	V17	522
99. I/O	-	88	114	V16	528
100. I/O	-	89	115	T13	531
101. I/O	-	90	116	U14	534
-	-	-	117*	-	-
-	-	-	118*	-	-
GND	-	91	119	T12	-
102. I/O	-	92	120	U13	540
103. I/O	-	93	121	V13	543
104. I/O (D5)	59	94	122	U12	552
105. I/O ($\overline{\text{CS0}}$)	60	95	123	V12	555
106. I/O	-	-	124	T11	558
107. I/O	-	-	125	U11	564
108. I/O	-	96	126	V11	567
109. I/O	-	97	127	V10	570
110. I/O (D4)	61	98	128	U10	576
111. I/O	62	99	129	T10	579
VCC	63	100	130	R10	-
GND	64	101	131	R9	-

Pin Description †	PC84	PQ160	PQ208	PG191	Boundary Scan Order
112. I/O (D3)	65	102	132	T9	588
113. I/O ($\overline{\text{RS}}$)	66	103	133	U9	591
114. I/O	-	104	134	V9	600
115. I/O	-	105	135	V8	603
116. I/O	-	-	136	U8	612
117. I/O	-	-	137	T8	615
118. I/O (D2)	67	106	138	V7	618
119. I/O	68	107	139	U7	624
120. I/O	-	108	140	V6	627
121. I/O	-	109	141	U6	630
GND	-	110	142	T7	-
-	-	-	143*	-	-
-	-	-	144*	-	-
122. I/O	-	111	145	U5	636
123. I/O	-	112	146	T6	639
124. I/O (D1)	69	113	147	V3	642
125. I/O ($\overline{\text{RCLK}}$-$\overline{\text{BUSY}}$/RDY)	70	114	148	V2	648
126. I/O	-	115	149	U4	651
127. I/O	-	116	150	T5	654
128. I/O (D0, DIN)	71	117	151	U3	660
129. I/O (DOUT)	72	118	152	T4	663
CCLK	73	119	153	V1	-
VCC	74	120	154	R4	-
-	-	-	155*	-	-
-	-	-	156*	-	-
-	-	-	157*	-	-
-	-	-	158*	-	-
130. (I/O) TDO	75	121	159	U2	-
GND	76	122	160	R3	-
131. I/O (A0, $\overline{\text{WS}}$)	77	123	161	T3	9
132. I/O (GCK4, A1)	78	124	162	U1	15
133. I/O	-	125	163	P3	18
134. I/O	-	126	164	R2	21
135. I/O (CS1, A2)	79	127	165	T2	27
136. I/O (A3)	80	128	166	N3	30
137. I/O	-	129	167	P2	33
138. I/O	-	130	168	T1	42
-	-	-	169*	-	-
-	-	-	170*	-	-
GND	-	131	171	M3	-
139. I/O	-	132	172	P1	45
140. I/O	-	133	173	N1	51
141. I/O (A4)	81	134	174	M2	54
142. I/O (A5)	82	135	175	M1	57
143. I/O	-	-	176	L3	63
144. I/O	-	136	177	L2	66
145. I/O	-	137	178	L1	69
146. I/O	-	138	179	K1	75
147. I/O (A6)	83	139	180	K2	78
148. I/O (A7)	84	140	181	K3	81
GND	1	141	182	K4	-

Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 666 = BSCAN.UPD

XC5210 Pinouts

Pin Description [†]	PC84	PQ160	PQ208	PG223	PQ240	Boundary Scan Order
VCC	2	142	183	J4	212	-
1. I/O (A8)	3	143	184	J3	213	111
2. I/O (A9)	4	144	185	J2	214	114
3. I/O	-	145	186	J1	215	117
4. I/O	-	146	187	H1	216	123
5. I/O	-	-	188	H2	217	126
6. I/O	-	-	189	H3	218	129
-	-	-	-	-	219*	-
7. I/O (A10)	5	147	190	G1	220	135
8. I/O (A11)	6	148	191	G2	221	138
VCC	-	-	-	-	222	-
9. I/O	-	-	-	H4	223	141
10. I/O	-	-	-	G4	224	150
11. I/O	-	149	192	F1	225	153
12. I/O	-	150	193	E1	226	162
GND	-	151	194	G3	227	-
13. I/O	-	-	195	F2	228	165
14. I/O	-	-	196	D1	229	171
15. I/O	-	152	197	C1	230	174
16. I/O	-	153	198	E2	231	177
17. I/O (A12)	7	154	199	F3	232	183
18. I/O (A13)	8	155	200	D2	233	186
19. I/O	-	-	-	F4	234	189
20. I/O	-	-	-	E4	235	195
21. I/O	-	156	201	B1	236	198
22. I/O	-	157	202	E3	237	201
23. I/O (A14)	9	158	203	C2	238	210
24. I/O (A15)	10	159	204	B2	239	213
VCC	11	160	205	D3	240	-
-	-	-	206*	-	-	-
-	-	-	207*	-	-	-
-	-	-	208*	-	-	-
-	-	-	1*	-	-	-
GND	12	1	2	D4	1	-
-	-	-	3*	-	-	-
25. GCK1 (A16, I/O)	13	2	4	C3	2	222
26. I/O (A17)	14	3	5	C4	3	225
27. I/O	-	4	6	B3	4	231
28. I/O	-	5	7	C5	5	234
29. I/O (TDI)	15	6	8	A2	6	237
30. I/O (TCK)	16	7	9	B4	7	243
31. I/O	-	8	10	C6	8	246
32. I/O	-	9	11	A3	9	249
33. I/O	-	-	12	B5	10	255
34. I/O	-	-	13	B6	11	258
35. I/O	-	-	-	D5	12	261
36. I/O	-	-	-	D6	13	267
GND	-	10	14	C7	14	-
37. I/O	-	11	15	A4	15	270
38. I/O	-	12	16	A5	16	273
39. I/O (TMS)	17	13	17	B7	17	279
40. I/O	18	14	18	A6	18	282
VCC	-	-	-	-	19	-
41. I/O	-	-	-	D7	20	285
42. I/O	-	-	-	D8	21	291
-	-	-	-	-	22*	-
43. I/O	-	-	19	C8	23	294
44. I/O	-	-	20	A7	24	297
45. I/O	-	15	21	B8	25	306
46. I/O	-	16	22	A8	26	309
47. I/O	19	17	23	B9	27	318
48. I/O	20	18	24	C9	28	321
GND	21	19	25	D9	29	-
VCC	22	20	26	D10	30	-
49. I/O	23	21	27	C10	31	327
50. I/O	24	22	28	B10	32	330
51. I/O	-	23	29	A9	33	333
52. I/O	-	24	30	A10	34	339
53. I/O	-	-	31	A11	35	342
54. I/O	-	-	32	C11	36	345
-	-	-	-	-	37*	-
55. I/O	-	-	-	D11	38	351
56. I/O	-	-	-	D12	39	354
VCC	-	-	-	-	40	-
57. I/O	25	25	33	B11	41	357
58. I/O	26	26	34	A12	42	363
59. I/O	-	27	35	B12	43	366
60. I/O	-	28	36	A13	44	369
GND	-	29	37	C12	45	-
61. I/O	-	-	-	D13	46	375
62. I/O	-	-	-	D14	47	378
63. I/O	-	-	38	B13	48	381
64. I/O	-	-	39	A14	49	387
65. I/O	-	30	40	A15	50	390
66. I/O	-	31	41	C13	51	393
67. I/O	27	32	42	B14	52	399
68. I/O	-	33	43	A16	53	402
69. I/O	-	34	44	B15	54	405
70. I/O	-	35	45	C14	55	411
71. I/O	28	36	46	A17	56	414
72. I/O	29	37	47	B16	57	417
73. M1 (I/O)	30	38	48	C15	58	426
GND	31	39	49	D15	59	-
74. M0 (I/O)	32	40	50	A18	60	429
-	-	-	51*	-	-	-
-	-	-	52*	-	-	-
-	-	-	53*	-	-	-
-	-	-	54*	-	-	-
VCC	33	41	55	D16	61	-
75. M2 (I/O)	34	42	56	C16	62	432
76. GCK2 (I/O)	35	43	57	B17	63	435
77. I/O (HDC)	36	44	58	E16	64	444
78. I/O	-	45	59	C17	65	447
79. I/O	-	46	60	D17	66	450
80. I/O	-	47	61	B18	67	456
81. I/O (LDC)	37	48	62	E17	68	459
82. I/O	-	49	63	F16	69	462
83. I/O	-	50	64	C18	70	468
84. I/O	-	-	65	D18	71	471
85. I/O	-	-	66	F17	72	474
86. I/O	-	-	-	E15	73	480
87. I/O	-	-	-	F15	74	483
GND	-	51	67	G16	75	-
88. I/O	-	52	68	E18	76	486
89. I/O	-	53	69	F18	77	492
90. I/O	38	54	70	G17	78	495
91. I/O	39	55	71	G18	79	504
VCC	-	-	-	-	80	-
92. I/O	-	-	72	H16	81	507
93. I/O	-	-	73	H17	82	510
-	-	-	-	-	83*	-
94. I/O	-	-	-	G15	84	516
95. I/O	-	-	-	H15	85	519
96. I/O	-	56	74	H18	86	522
97. I/O	-	57	75	J18	87	528
98. I/O	40	58	76	J17	88	531
99. I/O ($\overline{\text{ERR}}$, $\overline{\text{INIT}}$)	41	59	77	J16	89	534
VCC	42	60	78	J15	90	-
GND	43	61	79	K15	91	-
100. I/O	44	62	80	K16	92	540

* Indicates unconnected package pins.

[†] Leading number refers to bonded pad, shown in Figure 21.

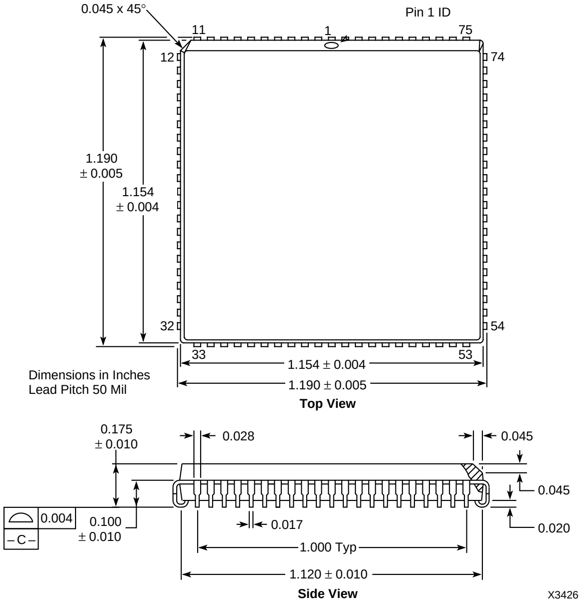
XC5210 Pinouts (continued)

Pin Description †	PC84	PQ160	PQ208	PG223	PQ240	Boundary Scan Order
101. I/O	45	63	81	K17	93	543
102. I/O	-	64	82	K18	94	546
103. I/O	-	65	83	L18	95	552
104. I/O	-	-	84	L17	96	555
105. I/O	-	-	85	L16	97	558
-	-	-	-	-	98*	-
106. I/O	-	-	-	L15	99	564
107. I/O	-	-	-	M15	100	567
VCC	-	-	-	-	101	-
108. I/O	46	66	86	M18	102	570
109. I/O	47	67	87	M17	103	576
110. I/O	-	68	88	N18	104	579
111. I/O	-	69	89	P18	105	588
GND	-	70	90	M16	106	-
112. I/O	-	-	-	N15	107	591
113. I/O	-	-	-	P15	108	600
114. I/O	-	-	91	N17	109	603
115. I/O	-	-	92	R18	110	606
116. I/O	-	71	93	T18	111	612
117. I/O	-	72	94	P17	112	615
118. I/O	48	73	95	N16	113	618
119. I/O	49	74	96	T17	114	624
120. I/O	-	75	97	R17	115	627
121. I/O	-	76	98	P16	116	630
122. I/O	50	77	99	U18	117	636
123. I/O	51	78	100	T16	118	639
GND	52	79	101	R16	119	-
-	-	-	102*	-	-	-
DONE	53	80	103	U17	120	-
-	-	-	104*	-	-	-
-	-	-	105*	-	-	-
VCC	54	81	106	R15	121	-
-	-	-	107*	-	-	-
PROG	55	82	108	V18	122	-
124. I/O (D7)	56	83	109	T15	123	648
125. GCK3 (I/O)	57	84	110	U16	124	651
126. I/O	-	85	111	T14	125	660
127. I/O	-	86	112	U15	126	663
128. I/O	-	-	-	R14	127	666
129. I/O	-	-	-	R13	128	672
130. I/O (D6)	58	87	113	V17	129	675
131. I/O	-	88	114	V16	130	678
132. I/O	-	89	115	T13	131	684
133. I/O	-	90	116	U14	132	687
134. I/O	-	-	117	V15	133	690
135. I/O	-	-	118	V14	134	696
GND	-	91	119	T12	135	-
136. I/O	-	-	-	R12	136	699
137. I/O	-	-	-	R11	137	708
138. I/O	-	92	120	U13	138	711
139. I/O	-	93	121	V13	139	714
VCC	-	-	-	-	140	-
140. I/O (D5)	59	94	122	U12	141	720
141. I/O (CS0)	60	95	123	V12	142	723
-	-	-	-	-	143*	-
142. I/O	-	-	124	T11	144	726
143. I/O	-	-	125	U11	145	732
144. I/O	-	96	126	V11	146	735
145. I/O	-	97	127	V10	147	738
146. I/O (D4)	61	98	128	U10	148	744
147. I/O	62	99	129	T10	149	747
VCC	63	100	130	R10	150	-
GND	64	101	131	R9	151	-
148. I/O (D3)	65	102	132	T9	152	756

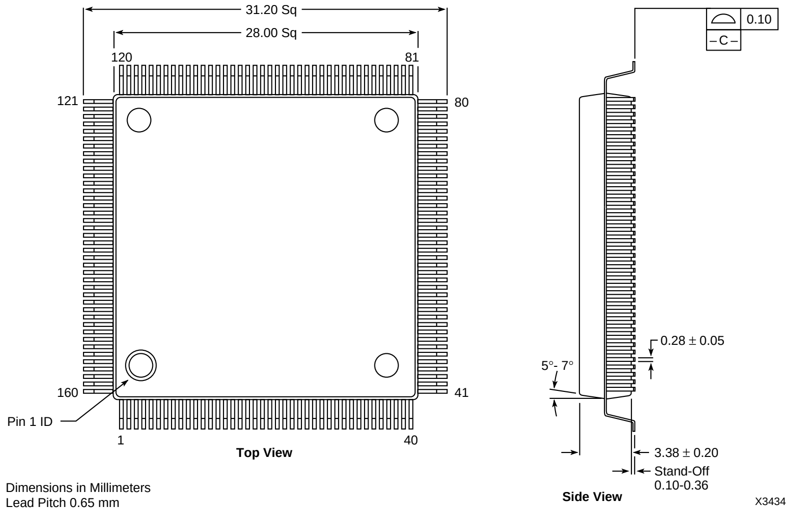
Pin Description †	PC84	PQ160	PQ208	PG223	PQ240	Boundary Scan Order
149. I/O (RS)	66	103	133	U9	153	759
150. I/O	-	104	134	V9	154	768
151. I/O	-	105	135	V8	155	771
152. I/O	-	-	136	U8	156	780
153. I/O	-	-	137	T8	157	783
-	-	-	-	-	158*	-
154. I/O (D2)	67	106	138	V7	159	786
155. I/O	68	107	139	U7	160	792
VCC	-	-	-	-	161	-
156. I/O	-	108	140	V6	162	795
157. I/O	-	109	141	U6	163	798
158. I/O	-	-	-	R8	164	804
159. I/O	-	-	-	R7	165	807
GND	-	110	142	T7	166	-
160. I/O	-	-	-	R6	167	810
161. I/O	-	-	-	R5	168	816
162. I/O	-	-	143	V5	169	819
163. I/O	-	-	144	V4	170	822
164. I/O	-	111	145	U5	171	828
165. I/O	-	112	146	T6	172	831
166. I/O (D1)	69	113	147	V3	173	834
167. I/O ($\overline{\text{RCLK-}}$ $\overline{\text{BUSY}}$/RDY)	70	114	148	V2	174	840
168. I/O	-	115	149	U4	175	843
169. I/O	-	116	150	T5	176	846
170. I/O (D0, DIN)	71	117	151	U3	177	855
171. I/O (DOUT)	72	118	152	T4	178	858
CCLK	73	119	153	V1	179	-
VCC	74	120	154	R4	180	-
-	-	-	155*	-	-	-
-	-	-	156*	-	-	-
-	-	-	157*	-	-	-
-	-	-	158*	-	-	-
172. I/O (TDO)	75	121	159	U2	181	-
GND	76	122	160	R3	182	-
173. I/O (A0, $\overline{\text{WS}}$)	77	123	161	T3	183	9
174. GCK4 (I/O, A1)	78	124	162	U1	184	15
175. I/O	-	125	163	P3	185	18
176. I/O	-	126	164	R2	186	21
177. I/O (CS1, A2)	79	127	165	T2	187	27
178. I/O (A3)	80	128	166	N3	188	30
179. I/O	-	-	-	P4	189	33
180. I/O	-	-	-	N4	190	39
181. I/O	-	129	167	P2	191	42
182. I/O	-	130	168	T1	192	45
183. I/O	-	-	169	R1	193	51
184. I/O	-	-	170	N2	194	54
-	-	-	-	-	195*	-
GND	-	131	171	M3	196	-
185. I/O	-	132	172	P1	197	57
186. I/O	-	133	173	N1	198	66
187. I/O	-	-	-	M4	199	69
188. I/O	-	-	-	L4	200	75
VCC	-	-	-	-	201	-
189. I/O (A4)	81	134	174	M2	202	78
190. I/O (A5)	82	135	175	M1	203	81
-	-	-	-	-	204*	-
191. I/O	-	-	176	L3	205	87
192. I/O	-	136	177	L2	206	90
193. I/O	-	137	178	L1	207	93
194. I/O	-	138	179	K1	208	99
195. I/O (A6)	83	139	180	K2	209	102
196. I/O (A7)	84	140	181	K3	210	105
GND	1	141	182	K4	211	-

Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 666 = BSCAN.UPD

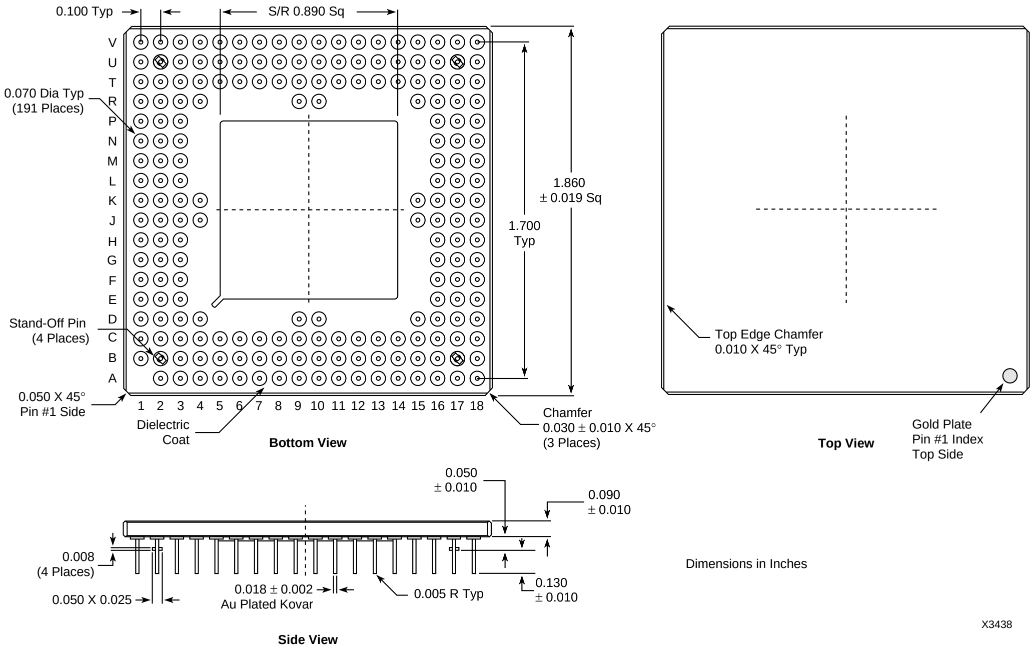
Physical Dimensions



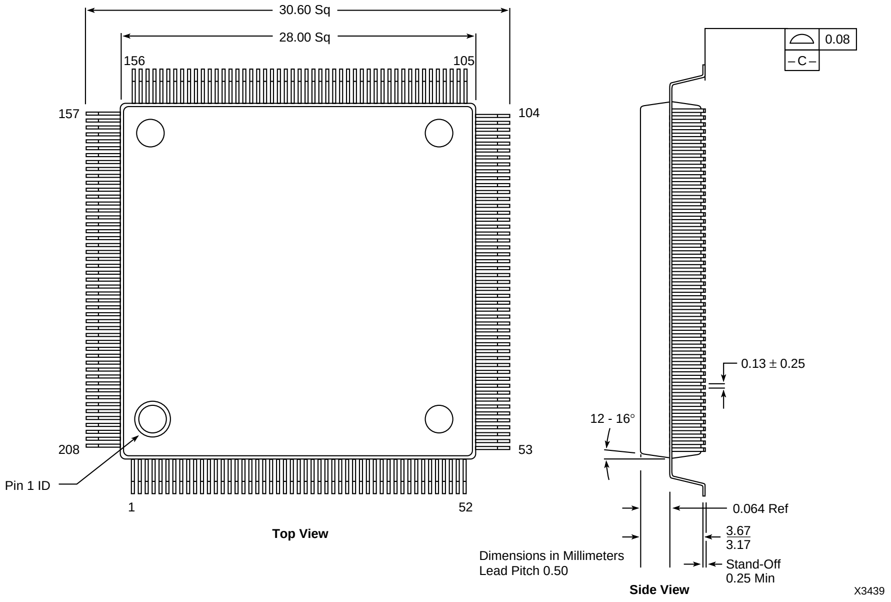
84-Pin Plastic PLCC (PC84)



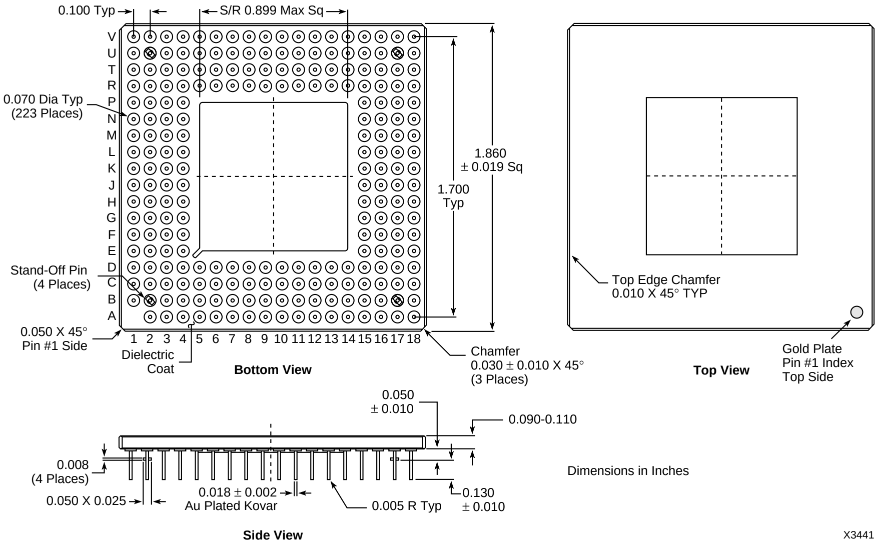
160-Pin Plastic PQFP (PQ160)



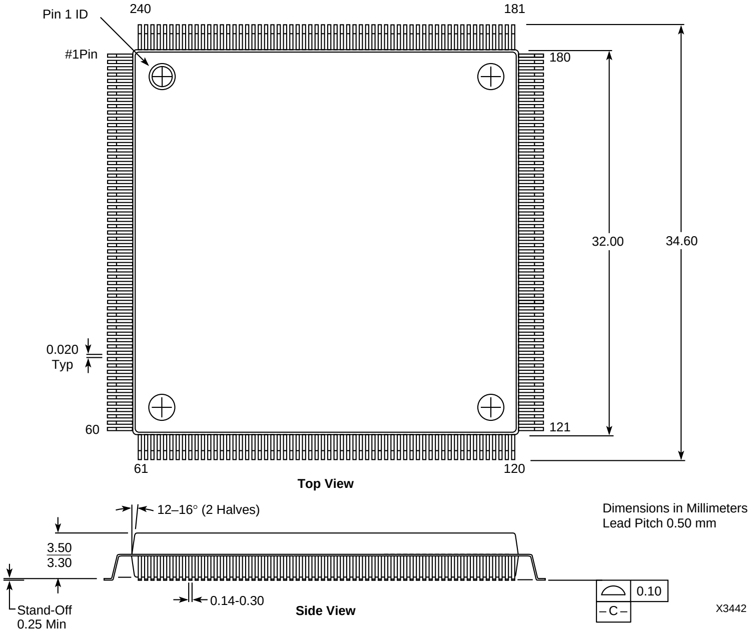
191-Pin Ceramic PGA (PG191)



208-Pin Plastic PQFP (PQ208)

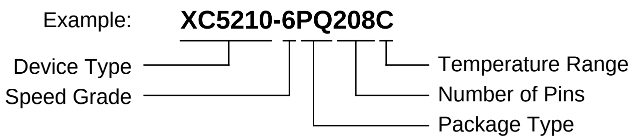


223-Pin Ceramic PGA (PG223)



240-Pin Plastic PQFP (PQ240)

Ordering Information



Component Availability

PINS		84	160	191	208	223	240
TYPE		PLASTIC PLCC	PLASTIC PQFP	CERAMIC PGA	PLASTIC PQFP	CERAMIC PGA	PLASTIC PQFP
CODE		PC84	PQ160	PG191	PQ208	PG223	PQ240
XC5206	-7	CI	CI	CI	CI		
	-6	CI	CI	CI	CI		
	-5	C	C	C	C		
XC5210	-7	CI	CI		CI	CI	CI
	-6	CI	CI		CI	CI	CI
	-5	CI	CI		CI	CI	CI

C = Commercial = 0° to +70°C **I = Industrial = -40° to +85°C**

Number of Available I/O Pins

Device	Max I/O	Package Type					
		PC84	PQ160	PG191	PQ208	PG223	PQ240
XC5206	148	65	133	148	148		
XC5210	196	65	133		164	196	196

XILINX®

The Programmable Logic Company℠

2100 Logic Drive, San Jose CA 95124-3400
Tel: (408) 559-7778 FAX: (408) 559-7114

0401300

Printed in U.S.A.

P/N 0401300 E2