XC8100 FPGA Family

June 1, 1996 (Version 1.0)

Preliminary Product Specification

Features

- Synthesis-targeted sea-of-gates architecture
  - Efficient results with top-down design
  - Design without architecture knowledge
  - Predictable pre-layout timing estimation
  - Accurate back-annotation
  - Very high routability
  - ASIC design flow
- Fine-grain architecture
  - High, predictable utilization: >95%
  - TrueMap logic mapping
- Innovative programmable cell
  - Combinatorial, synchronous, or three-state
  - High logic utilization for all designs
- Family of devices: 1K-45K usable gates
  - Same pinout as XC4000 and XC5200
  - MicroVia™ antifuse technology
- Low power CMOS
- System features
  - JTAG boundary scan
  - Fast, wide internal decode
  - On-chip three-state for internal bussing
  - I/O drive = 24 mA; PCI drive compliant
  - Slew-rate options to control ground bounce
  - Modular clock/buffer resources
  - 5 V, 3.3 V operation
- One-time programmable, single-chip solution
  - Design security
  - Xilinx and third party programmers
  - Self-test logic for 100% testability
  - Automatic post-programming test
- XACT*step*™ Series 8000 development system
  - Xilinx unified design entry libraries
  - Floorplanning, incremental design
  - High-speed PowerMaze™ router
  - PC: Windows 3.1/95/NT
  - WS: Sparcstation, HP PA, RS6000
- Supported by XACT*step* Foundation Series

Description

The XC8100 family of field programmable gate arrays (FPGAs) provides the same overall benefits as other Xilinx FPGAs: fast time-to-market, reduced design risk, low power, standard product availability, and the use of existing design methodologies. It combines the density of mask gate arrays with the flexibility of programmable logic.

The XC8100 family is targeted to be extremely efficient and cost effective when using top-down, technology-independent design methods. The XC8100 employs a new sea-of-gates FPGA architecture. The basic cell is small and was specifically architected for technology-independent design. A new process, the Xilinx MicroVia technology, minimizes the area taken up by the many interconnect elements used in a fine-grain structure. Programmable interconnect elements are stacked vertically between metal layers and are above the logic cells, using significantly less area than other programmable logic technologies. (Note that architectural diagrams in this data sheet do not necessarily show this internal structure). The result is that XC8100 devices have very rich interconnect resources while maintaining cost effectiveness. The XC8100 sea-of-gates architecture delivers high gate utilization, high routability, low cost, and fast design cycles. For high speed, the MicroVia antifuse has a typical on-resistance less than 50 Ω.

Like all true FPGAs, as shown in Figure 1, the XC8100 family consists of an array of logic cells and programmable routing resources surrounded by a ring of I/O connections. Unlike most FPGAs, which attempt to offer the "best" fixed cell, the XC8100 logic cells are themselves programmable. They can implement synchronous, combinatorial, or three-state functions. This means the XC8100 software has the flexibility to choose the best cell structure, depending on the logic to be implemented. A design does not have to be evaluated to see if it "fits", but instead can be implemented top-down.

Table 1: Product Line

Product	XC8100	XC8101	XC8103	XC8106	XC8109	XC8112*	XC8116*	XC8120*
Max Logic Gates	1K	2K	7K	13K	20K	27K	36K	45K
Typical Gate Range	.6 - 1K	1K - 2K	3K - 7K	6K - 13K	9K - 20K	12K - 27K	16K - 36K	20K - 45K
Cells	192	384	1024	1728	2688	3744	4800	6144
Flip-Flops (Max)	96	192	512	864	1344	1872	2400	3072
I/O	32	72	128	168	192	248	280	320

Note: * Future product plans

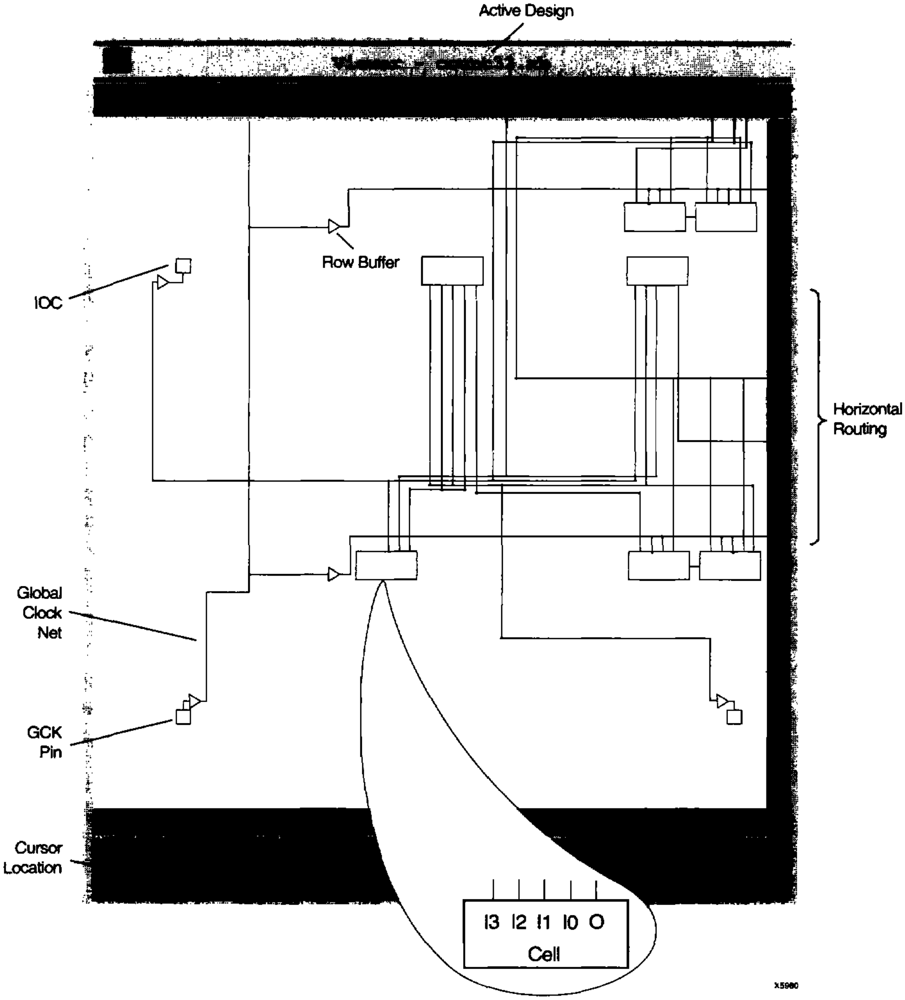


Figure 1: XC8100 Architecture in Viewer Window

Compared to Xilinx XC2000, XC3000, or XC4000 FPGAs, this architecture has many more cells, each with fewer gates. It also has fewer preconfigured resources — for example, no flip-flops in the logic cells or in the I/O cells. Routing resources are abundant and there is a very large ratio of interconnects to logic cell inputs and outputs. The architecture's objective is to achieve the highest gate utilization and routability across a diverse set of applications.

Unlike Xilinx SRAM FPGAs, which are infinitely reprogrammable, the XC8100 family is one-time programmable (OTP). XC8100 devices use MicroVia process technology, a combination of CMOS, a metal-to-metal antifuse and three layers of metal. Programming is done by Xilinx or third-party programmers, similar to OTP PLDs. There is no need for configuration storage in the target application and design security is very high.

The XC8100 design flow is exactly like that of an ASIC or gate array. Designs are entered and simulated with third-party CAE tools and then placed and routed by Xilinx XACT*step* Series 8000 tools. A key feature of the synthesis design flow is that technology mapping occurs in the synthesizer. All instance names, net names, and hierarchy in the EDIF netlist from the synthesizer are maintained by the Series 8000 software. Every logic gate in the design corresponds to a logic gate/CLC in the FPGA. Every net in the design corresponds to a net in the FPGA. Therefore, this information and the associated timings are available for backannotation, simulation and debugging. Figure 2 shows this mapping process – called TrueMap.

The XC8100 architecture maximizes the chance that a design will be completed automatically. Because of the architecture and new software written for it, device utiliza-

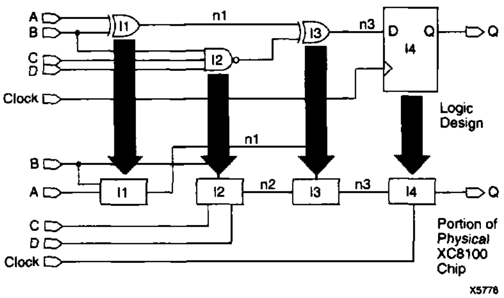


Figure 2: TrueMap™ Logic Mappings

tion does not degrade when technology independent design entry is used. Designs can be entered in HDLs with little concern for the IC architecture. With the very high probability of routing, different logic implementations can be accurately evaluated before place and route.

The Series 8000 software allows user control, primarily for maximizing the speed of the design. High level floorplanning works with constraints based on hierarchical block names. The fast PowerMaze router makes incremental design easier. The router can route a fully-utilized XC8106 in less than one minute – in most cases in only seconds.

Applications

The XC8100 family is targeted at three primary applications:

1. High-level design language (HDL) using logic synthesis. The architecture was developed to give very high gate utilization and low cost when designs are entered with a technology-independent methodology. Most synthesis algorithms were originally designed for gate arrays, and they are sub-optimal for the different architectures used in FPGAs and complex PLDs. The XC8100 architecture was specifically designed to fit the logic implementations that are produced by synthesis algorithms. During development, architectural simulations were run using actual synthesis products. The XC8100 programmable cell and design library were chosen to give cost-effective results when using logic synthesis. The XC8100 software is very ASIC-like and gives accurate estimations when synthesizing and evaluating architectural tradeoffs before place and route.
2. General logic applications, especially telecommunications and industrial control. The XC8100 offers high and predictable utilization over a wide range of logic functions. It can handle designs that are heavily synchronous or heavily combinatorial. Since the architecture delivers "usable gates" irrespective of the target application, designs can be quickly entered and results will be predictable. There is no need to analyze the target design for flip-flops, latches, three-states, etc., to determine how well the design "fits."

   XC8100 devices, especially the XC8100 and XC8101, can also implement logic traditionally done in CPLDs, with less than one-tenth the power.
3. Applications requiring a single-chip FPGA. These include high security and fast initial power-up operation.

Security

The XC8100 offers a high level of security for designs that are subject to reverse-engineering. The security strategy includes the software, the IC architecture, and the process.

The XC8100 software automatically stores design information (.xb file) in a compressed and encrypted file format. In addition, there is an optional password capability that only allows the design file to be used for programming, not for viewing or writing or manipulating. In other words, the password-protected design file can be given out for device programming while still maintaining security.

XC8100 devices receive programming information through the JTAG (boundary scan) port and data registers. While this information can be read back for factory testing purposes, these JTAG instructions are not documented. The user can program a bit which defeats the JTAG test/programming readback instructions, thereby eliminating this form of reverse engineering.

The third part of the security strategy involves the process. The physical implementation of a design occurs by programming the desired connection pattern. The connection is formed by creating a metal filament through a layer of amorphous silicon. Only a small percentage of the antifuses are programmed. In a typical design in the XC8106, about 2% of the approximately 700,000 antifuses will be programmed. The functionality of the design can only be copied by knowing exactly which of the antifuses were programmed.

The antifuses are located between the second and third layer of metal. This means that programmed and unprogrammed antifuses are covered by a layer of metal and a protective passivation layer. It is impossible to distinguish programmed antifuses from unprogrammed antifuses by inspection of the top of the chip. Attempts to remove the protection and metal layers will certainly result in damage or removal of the conductive filament between the metal layers.

Moreover, antifuses, whether programmed or unprogrammed, are difficult to distinguish from regular, permanent connections between the second and third layers of metal (vias). Note that this is different from a gate array, where vias only exist where connections are intended. In the XC8100, vias, programmed antifuses, and unprogrammed antifuses all exist simultaneously.

Another method of reverse engineering is hot spot analysis, which relies on distinguishing programmed from unprogrammed fuses by investigation of heat dissipation. Since the on-resistance of the programmed antifuses is very low, the heat dissipation is very low also. This makes this technique useless for determination of the state of the antifuse.

Performance Overview

The XC8100 family has been benchmarked with many actual customer designs running synchronous clock rates of 20-40 MHz (-1 speed grade). The performance of any design depends on the type of circuit implemented, including the delay through the combinatorial and sequential logic elements plus the delay in the interconnect routing. Figure 3 shows some performance numbers for representative circuits, using worst case timing parameters. A rough estimate of timing can be made by assuming 6 ns per logic level. This includes 3 ns for the CLC delay and 3 ns for the routing delays. More accurate estimations can be made using the information in the next section.

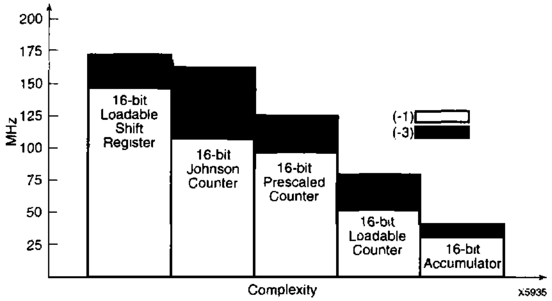


Figure 3: Representative Circuit Performance

	XC8100 Performance		
Function	**-1**	**-3***	
16-bit decoder from input pads (delay from strobe) (delay for full decode)	 7.6 11.7	 5.1 7.9	 ns ns
16-to-1 multiplexer	18.8	14.2	ns
16-bit loadable shift register	148	172	MHz
16-bit Johnson counter	109	161	MHz
16-bit prescaled counter	91	125	MHz
16-bit loadable counter	51	77	MHz
16-bit accumulator	28	41	MHz

*Advance Information

The XC8100 architecture and software are architected to deliver the maximum device performance when using synthesis CAE tools. This is possible due to:

- CLC designed to fit synthesis algorithms
- predictable prelayout timing for accurate synthesis
- block-level hierarchical floorplanning
- buffer resources for high fanout nets or critical signals
- flexible CLC - many flip-flops possible - allows pipelining
- cascade
- High drive outputs specified for 50 pF

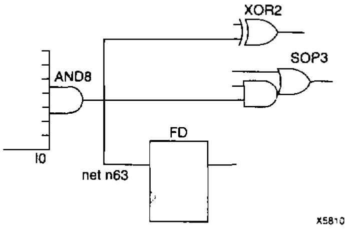


Figure 4: Example Path Delay Calculation

Estimating XC8100 Timing

Like an ASIC or FPGA, XC8100 circuit timings depend on the actual layout (placement and routing) of the design. XC8100 software calculates the timing using a very accurate SPICE-like timing model. However, prior to layout actual circuit performance can be estimated. This is done automatically by synthesis tools using the XC8100 library and can be done manually using information in this data sheet.

Accurate timing estimates prior to place and route are necessary when exploring a design space with synthesis tools. This allows the synthesizer to make the proper speed/area trade-offs without leaving the synthesis environment (i.e. without going through a full place and route cycle). The timing estimates for the XC8100 architecture are accurate to within 10% of the post place and route numbers. This high correlation between pre- and post- timing numbers is a result of the following architecture and design flow features:

- No logic mapping - the placed and routed design matches the synthesized design netlist exactly. There is no technology mapping phase where logic gates from the design netlist are mapped into different physical gates on the chip.
- All cell delays are built into the synthesis library. This, in conjunction with point 1, means that all cell delays used in the synthesis tool are preserved in the final chip.
- The wire load model is well characterized. The abundant routing in the architecture increases the likelihood of direct routes between CLCs rather than circuitous paths. This reduces the standard deviation of the estimated versus actual net delays.

The timing for the XC8100 is modeled with two elements: the cell delay and a fanout-dependent net delay (wireload model). Cell delay timings for most of the primitive library elements are shown in the AC Timing section of the data sheet. For example, the data sheet shows four numbers for the AND4 primitive. Worst-case numbers are worst-case voltage, worst-case temperature, worst-case process, and worst-case falling or rising edge. Note that the XC8100 library has input-to-output numbers, both rising and falling edges, for all pins. Synthesis software will automatically take advantage of the fastest pin if required, and schematic users can do this manually.

Pre-layout net delays are modeled as a function of fanout and can be estimated by using Table 2. Note that various buffers, as explained later, can be used to limit the delay of wide fanout nets.

Table 2: Pre-Layout Net Delays vs. Fanout

Fanout	Net Delay (ns) (-1)
1	1.2
2	2
3	3
4	4
n	n

For example, in Figure 4, net n63 has a fanout of 3, so the estimated delay (-1 speed grade) from the I0 input of the AND8 to the D input of FD is: 3.0 + 3 = 6.0 ns.

Architecture

Xilinx field-programmable gate arrays are comprised of three major configurable elements: configurable cells, input/output blocks and interconnections. The cells provide the functional elements for constructing the design's logic. The I/O cells provide the interface between the package pins and internal signal lines. The programmable interconnect resources provide routing paths to connect the inputs and outputs of the cells and I/O blocks onto the appropriate nets. In the XC8100, user configuration is established by one-time programming of MicroVia antifuses that determine the cell logic functions, I/O functions, and the interconnections.

Programmable Cell (CLC)

The XC8100 implements combinatorial and sequential logic by configuring and interconnecting identical Configurable Logic Cells (CLC). Each CLC is equivalent to 3.25 "LSI-Logic gates." This figure was derived by synthesizing a range of designs first to the XC8100, and then to an LSI Logic library.

In order to allow the widest range of logic - combinatorial, synchronous, and three-state - to map efficiently to the XC8100 architecture, CLCs are internally programmable. While this is also true of Xilinx SRAM FPGAs, other antifuse FPGA architectures have fixed cell structures with only *programmable interconnect*. A *programmable cell* ensures that logic will more frequently fit efficiently into a cell. The cell is programmable as to: input inversions, combinatorial function choice, synchronous logic internal feedback path, three-state, and cascade enable. The XC8100 software *automatically* configures the cell based on the user's design netlist. The variety of building blocks means a higher device utilization for a range of logic functions.

Figure 5 shows the different possible implementations of a single CLC. Each CLC has four logic inputs plus a cascade input, and one logic output plus a cascade output (cascade not shown). The logic output has three-state control, and each CLC is also connected to a global reset network. Each of the four inputs can be configured as inverted or non-inverted, so input signal inversions ("bubbling") are free and output bubbles don't matter. Cells do not have to be "wasted" as inverters, which would lower the gate utilization. Any input can be hardwired to logic zero or one.

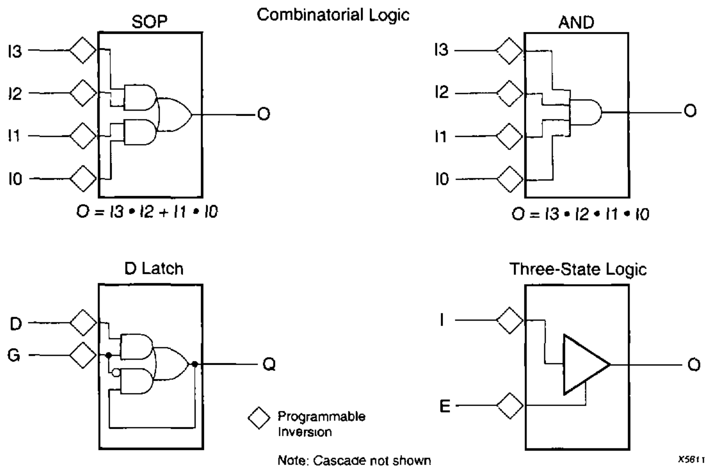


Figure 5: Programmable Cell

For combinatorial logic, the cell has two programmable functions, AND or Sum-of-Products (SOP). Pairs of CLC inputs are internally ANDed, and the two signals are then combined in a circuit programmable as either an OR or an AND gate. The SOP can also be considered a 2-input multiplexer, or a 2-input XOR and XNOR.

For synchronous logic, configuring the cell-internal feedback path makes any CLC a latch with D and G inputs (LD primitive). When G is high, the latch is in the pass-through state. Both inputs can be configured to be either active High or active Low. A two-cell latch (LDC primitive) adds an asynchronous CLR input. All latches are also controlled by the global Reset signal.

The XC8100 software automatically combines two adjacent latches through the Cascade connection to create a D-flip-flop. One CLC is the master latch and the second is the slave latch. The flip-flop has a global Reset which does not involve user routing and also has an overriding asynchronous Clear input (FDC primitive). It can also have an asynchronous Preset input that overrides the D input (FDP primitive). Note that neither the feedback connections of both latches nor the master-slave connection use general routing elements, or have programming elements in their paths. This gives the flip-flops excellent speed and metastability behavior. The XC8100 “semi-dedicated” flip-flops combine performance with flexibility in utilization.

For three-state logic, one of the CLC inputs controls a tristate buffer in the output path (BUFE primitive). The Data and Enable control inputs can be configured active High or active Low. Combining the latch and three-state functions gives a register file capability (LDE4 primitive) which can be used to build FIFOs, see Figure 6.

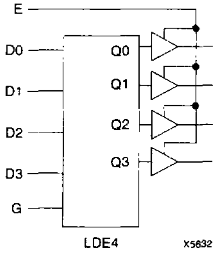


Figure 6: Quad Latch Bank with Three-State Output Uses Four CLCs

Global Reset

A global reset signal is automatically distributed to each CLC latch, and therefore each flip-flop. It is controlled internally by the GRST library primitive. It is activated automatically on power-up. When active, the net presets all asynchronous preset flip-flops (FDP primitive) and resets all other flip-flops. In the case of the latches, the clear only takes place when the latches are in the latching state. In other words, reset does not affect the output in pass-through mode. Global reset is typically complete within 100 ns. See AC timings.

Master Reset

A master reset of the chip is accomplished by pulling the $\overline{\text{MR}}$ pin low. This starts the power-on reset sequence, presetting all asynchronous preset flip-flops (FDP) and resetting all other flip-flops and latches. The master reset locks out the device for several ms. See the AC timings. The primary use of the master reset is as part of a system, board or module reset.

Cascade

Each CLC has a fifth input and second output for cascade. The cascade inputs and outputs use dedicated routing to the nearest cells, extending along an entire row. See Figure 8. Combining two adjacent CLCs through their cascade connection expands the functionality. For synchronous logic, cascade is used to automatically build flip-flops, as explained earlier. For combinatorial functions, the software cascades CLCs to build the wider ANDs and SOPS in the primitive library. Cascade can also be used (with ANDCC and SOPCC primitives) to build functions like wide decode and fast shift registers. The cascade connection has several advantages over conventional routing:

- It is faster because it doesn't need a programmable element and it avoids the delay of the cell output driver.
- It has known timing.
- It does not consume any general routing resources.

Address Decoders

The cascade can be used to build wide decode functions. See Figure 7. For example, five CLCs, each configured as a four-input AND gate, can feed a CLC configured as a four-input AND with cascade (ANDCX). The delay for this 20 input decoder is less than 12 ns. Using two-CLC AND8 primitives produces a forty-input decoder that has a delay less than 15 ns. If the address is available, the strobe delay is about 8 ns (-1).

Programmable Interconnect

Most programmable logic devices have to balance routing/interconnect resources with die size and cost. However, the XC8100 architecture offers a small die size with an abundant amount of routing, both metal-segment wires and interconnections. The routing software has many alternatives to make sure a design is fully routed.

Figure 8 is an example diagram of the interconnect, showing only the wires actually used for nets in a design. In the horizontal direction, there are 47 wires that each cell can connect to (not all shown in Figure 8). There are 5 separate types, three of which can also be connected horizontally:

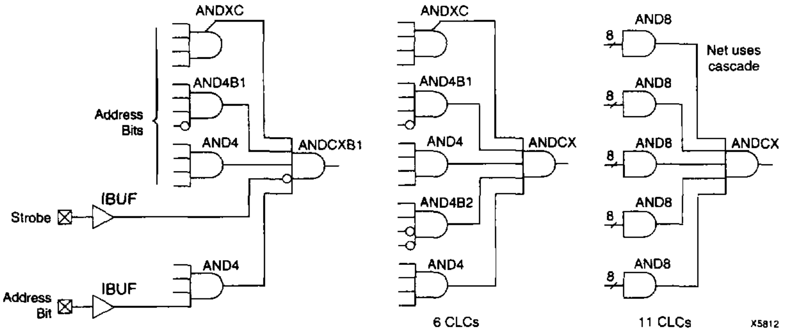


Figure 7: Wide Decode Using Cascade

- single-block (a block is 4 cells wide) length wires extend for one block only. They are used primarily to connect the cells within the block.
- double-block length wires connect left and right to adjacent blocks.
- quad-block length wires connect left and right for a length of four blocks. They can connect any cells that are spanned within the four-block length.
- horizontal long lines run the entire width of the device. They can connect any cells in a row. These wires are optionally used by the row buffers.
- constant 0 used for logic 0 and 1 inputs.

There are two wire types in the vertical direction:

- *double-block length wires, analogous to the horizontal* ones.
- quad-block length wires.

In addition to the wires that run along all the rows and columns of the device, there are twelve vertical long lines (VLLs) on each side of the device. They interconnect with any horizontal wire and make routing with preassigned I/Os easier.

Global Nets

Global nets are those needing optimized timing or low skew for the distribution of clock, time-critical and/or high fan-out control signals. The XC8100 has dedicated hardware resources for this purpose. The XC8100 system is very flexible since the various buffers can be used independently or together, for clocks or data. See Table 3, Figure 9, and Figure 10. Four package pins, GCK1-4, feed four dedicated high drive buffers (BUFEDGE) that drive four vertical long lines (not part of the general lines described in the Programmable Interconnect section) on two sides of the device. If the GCK pins are not used, they are available for user I/O. Each pair of rows of CLCs also has four dedicated buffers (BUFROW). To limit fanout, the XC8100 software can also use CLCs as buffers (BUF1X primitive). These buffers can be automatically selected by the synthesis software.

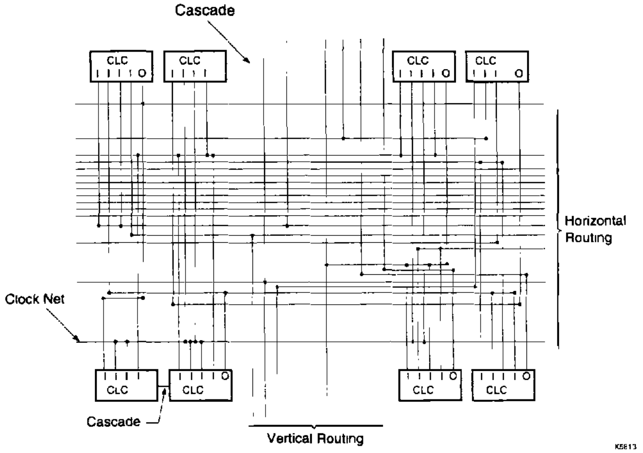


Figure 8: Example Routing Detail

Table 3: XC8100 Buffers

Name	Description
BUFGP	This global buffer is formed by a BUFEDGE driving some number of BUFROWs. The source for this buffer is one of four specific external pins, GCK1 - 4. The output of the buffer can drive all CLCs in a chip. The maximum number of BUFGP buffers is four.
BUFGS	This can be thought of as an internal BUFGP. It is accessible internally and can drive any number of BUFROWs.
BUF1X	This buffer is formed by using a single CLC. It has the same drive and fanout characteristics as the CLC. It can be automatically selected by synthesis tools to limit fanout in critical paths.
BUFROW	This is a dedicated high drive buffer whose output drives all CLCs in the driven row. The input to this buffer is from a CLC or from external pads through IBUF cells. Using this buffer automatically constrains the placement of the driven CLCs to the row. Each BUFROW drives two rows of CLC/FFs: XC8100 48 CLCs (24 FFs) XC8101 48 CLCs (24 FFs) XC8103 64 CLCs (32 FFs) XC8106 96 CLCs (48 FFs) XC8109 112 CLCs (56 FFs)

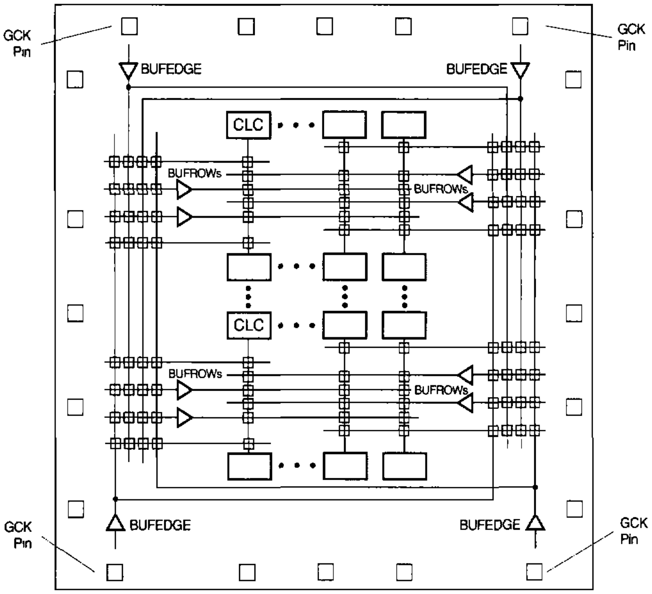
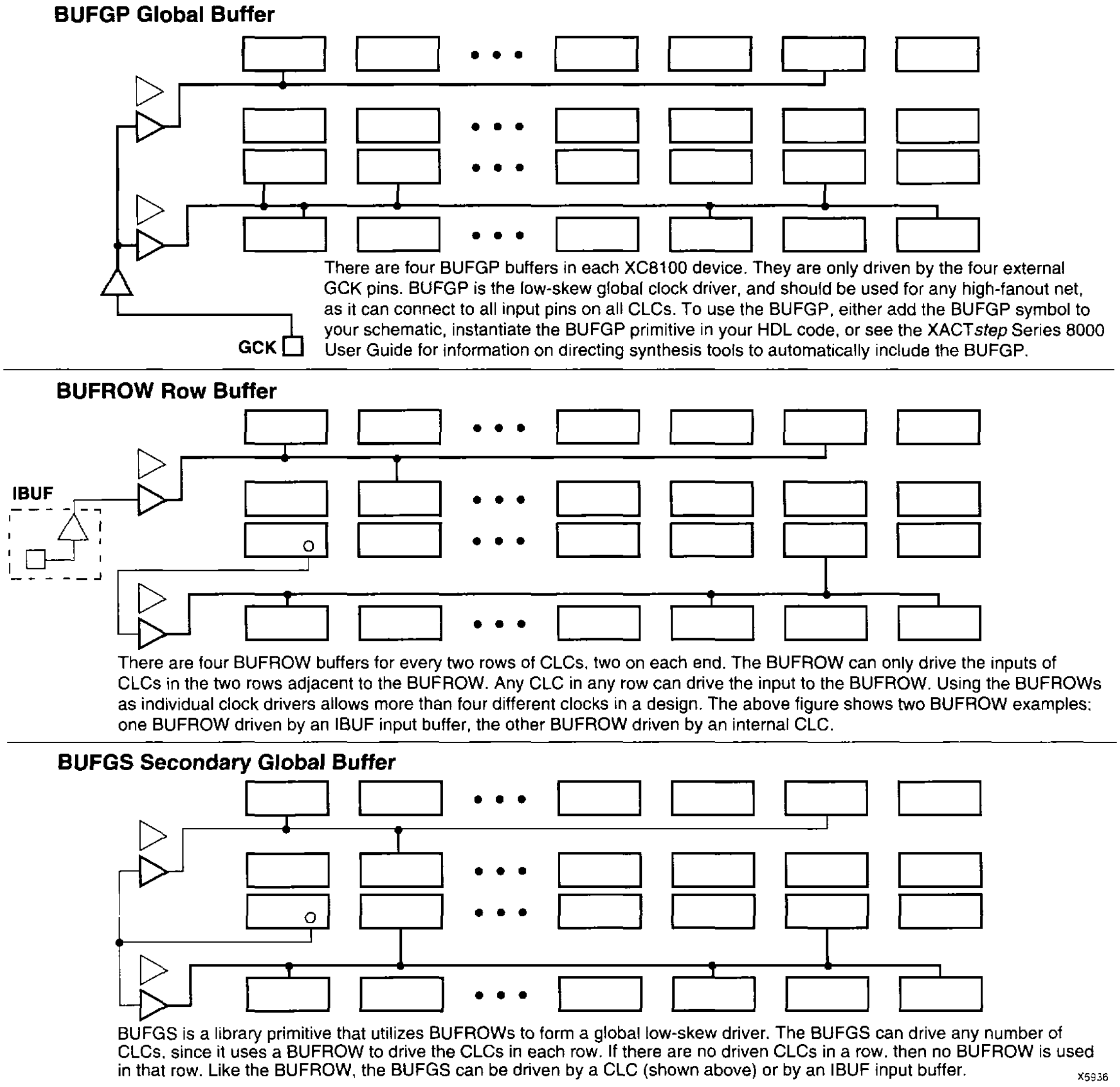


Figure 9: XC8100 Buffer Configurations



BUFGP Global Buffer

There are four BUFGP buffers in each XC8100 device. They are only driven by the four external GCK pins. BUFGP is the low-skew global clock driver, and should be used for any high-fanout net, as it can connect to all input pins on all CLCs. To use the BUFGP, either add the BUFGP symbol to your schematic, instantiate the BUFGP primitive in your HDL code, or see the XACT*step* Series 8000 User Guide for information on directing synthesis tools to automatically include the BUFGP.

BUFROW Row Buffer

There are four BUFROW buffers for every two rows of CLCs, two on each end. The BUFROW can only drive the inputs of CLCs in the two rows adjacent to the BUFROW. Any CLC in any row can drive the input to the BUFROW. Using the BUFROWs as individual clock drivers allows more than four different clocks in a design. The above figure shows two BUFROW examples: one BUFROW driven by an IBUF input buffer, the other BUFROW driven by an internal CLC.

BUFGS Secondary Global Buffer

BUFGS is a library primitive that utilizes BUFROWs to form a global low-skew driver. The BUFGS can drive any number of CLCs, since it uses a BUFROW to drive the CLCs in each row. If there are no driven CLCs in a row, then no BUFROW is used in that row. Like the BUFROW, the BUFGS can be driven by a CLC (shown above) or by an IBUF input buffer.

Figure 10: XC8100 Buffer Usage

Input/Output Cell (IOC)

The IOCs form the interface between the internal logic and the I/O pads of the device. Each IOC consist of a programmable output section that drives the pad, and a programmable input section that receives data from the pad. See Figure 11. Aside from being connected to the same pad, the input and output sections have nothing else in common.

Input

The input section receives data from the pad or the JTAG circuitry and passes it to the interconnect structure. Inputs can be globally programmed for TTL or CMOS input thresholds. TTL is the default. There are two input buffer timing specifications. The faster one is selected by using the IBUF library element, while a delayed timing is available with the DBUF library element. The DBUF is used for the data signal to guarantee zero hold time if data and clock signals are going from external pins to an internal flip-flop.

Pull-up Resistor

Each I/O pad can be programmed with or without a pull-up resistor. This selection is independent of the IOC usage. The default is for the software to automatically program the pull-up resistor to prevent a floating input. The pullup is about 50K ohms.

Output

The output section takes data and three-state control information from the interconnect structure. It can also take data from the JTAG scan circuitry. Output data can be inverted or non-inverted.

The output driver has four options: 1) it can be permanently disabled, making the pad an input-only pad (IBUF, DBUF primitives); 2) the driver can be permanently active, making the pad output only (OBUF, RBUF primitives); 3) it can be three-state controlled from either the internal logic or the JTAG circuitry (OBUFE, RBUFE primitives): or 4) a combination (IOBUF). The three-state control signal can be inverted, allowing the signal to be thought of as either active Low Output Enable, or active Low three-state. The outputs are CMOS compatible, which results in an unloaded rail-to-rail signal swing. Each output can be individually configured for either of two slew-rate options, which affect only the pull-down operation. See Figure 12.

When programmed for resistive mode (RBUF, RBUFE primitives), the pull-down transistor is driven hard, resulting in a practically constant on-resistance of < 20 Ω. This results in the fastest High-to-Low transition, and the capability to sink up to 24 mA. Resistive mode is required for driving terminated transmission lines with 4 to 24 mA of dc sink current and for highly capacitive loads (>200 pF). When many resistive mode outputs switch simultaneously High to Low, this configuration option may result in excessive ground bounce. The user must limit the number of simultaneous transitions per package ground pin - see the DC specification section.

The slew rate limited configuration is capacitive mode (OBUF, OBUFE). This mode uses a novel, patent-pending method of slew rate control that reduces ground bounce without any significant delay penalty. The High-to-Low transition starts as described above, but the drive to the pull-down transistor is reduced as soon as the output voltage reaches a value around 1V. This results in a higher resistance in the pull-down transistor, a slowing down of the falling edge, and significantly reduced ground bounce. In this mode the output driver sinks 4 mA at V_{OL}. This mode is recommended for outputs requiring less than 4 mA DC or for capacitive loads of less than 200 pF. See Figure 13 for typical output V/I characteristics.

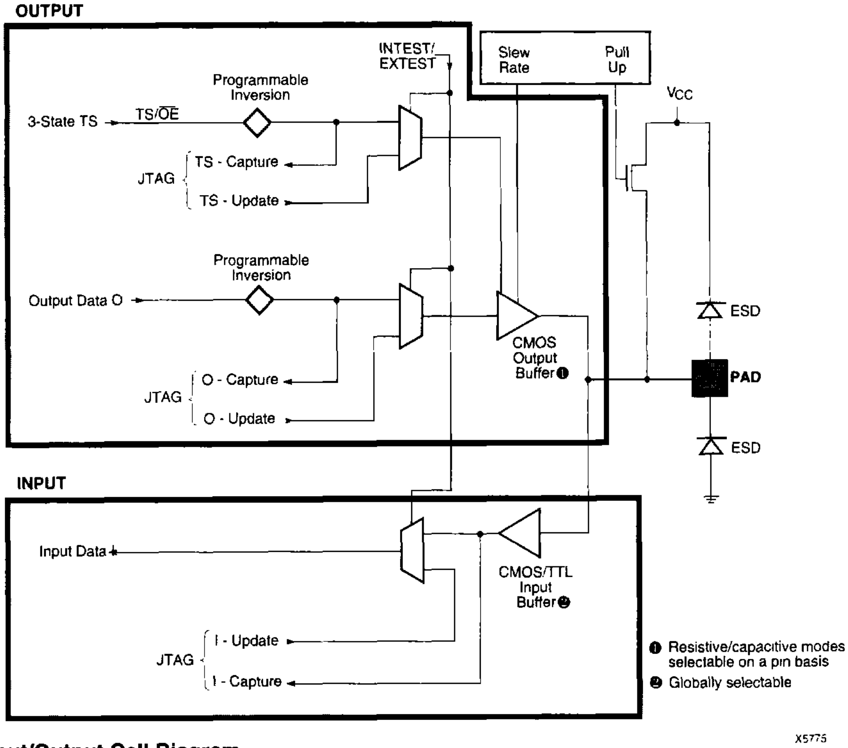


Figure 11: Input/Output Cell Diagram

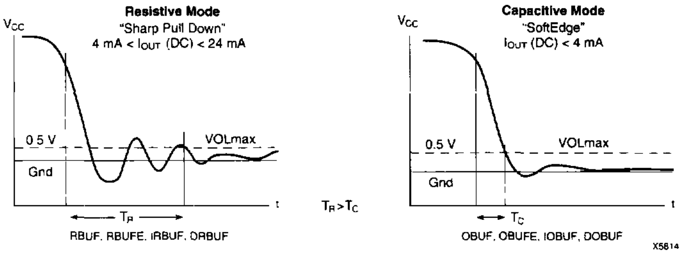


Figure 12: Output Slew Rate Options

IEEE 1149.1 Boundary Scan

The XC8100 has similar IEEE 1149.1/JTAG (Joint Test Action Group) capabilities as the XC4000. Three differences are that:

- The XC8100 JTAG pins TDI, TDO, TMS and TCK are dedicated.
- The XC8100 supports the JTAG instruction IDCODE.
- The XC8100 instruction word length is 10 bits.

JTAG is an industry-standard serial access interface designed to provide an efficient and convenient means of testing and monitoring integrated circuit components on a circuit board. It was designed to provide an alternative to the traditional bed-of-nails circuit board testers which have difficulty accommodating high pin count, fine pitch component packages, and which are unable to support surface mount boards with components on both sides. In the XC8100, the JTAG interface both serves its traditional board testing role and is also the primary test and antifuse programming interface. Virtually all factory test and programming operations are accessed via the JTAG interface.

The 1149.1 interface consists of four pins, TDI, TDO, TCK, and TMS. The basic concept is that the TDI and TDO pins form the beginning and end of serial shift registers within the device. See Figure 14. TCK and TMS provide a means to select a specific shift register and when and what to shift.

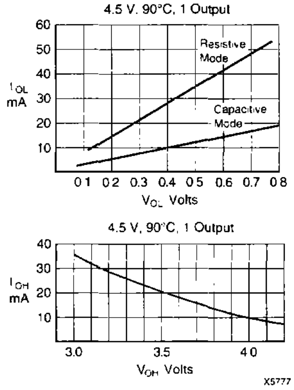


Figure 13: Typical Output V/I Characteristics

In order to provide control of the JTAG data registers, the JTAG specification includes a definition of what is called the test access port (TAP) controller. Since there are only two pins to control it, the TAP controller is designed as a state machine with one input, TMS. The XC8100 TAP controller is clocked by the TMS pin and uses the standard IEEE JTAG state diagram.The XC8100 provides the full set of IEEE 1149.1 boundary scan instructions, including the optional IDCODE. See the following Table:

Instruction Code	Instruction	Description
0000000000	INTEST/ EXTEST	Forces state of I/O cells
0000000001	SAMPLE	Samples state of I/O cells
1111111101	USERCODE	User-programmable register
1111111110	IDCODE	Samples existing device code
1111111111	BYPASS	Selects bypass register

INTEST/EXTEST

This instruction combines the mandatory JTAG EXTEST with the optional INTEST instruction. When selected, this instruction uses the JTAG boundary scan register values to:

- Drive the signal at the output of the input buffer (INTEST)
- Drive the data input of the output buffer (EXTEST)
- Drive the enable input of the output buffer (EXTEST)

To do this, the signals which normally drive these nodes are disabled via multiplexers.

SAMPLE

This instruction samples the logic values at each pin. For a bidirectional I/O pin, three values are sampled:

- The signal at the output of the input buffer
- The signal at the data input of the output buffer
- The signal at the enable input of the output buffer

IDCODE

This instruction provides access to a 32-bit data register that always captures a 32-bit word built into the device. The register can be read by the programming software. See Figure 15.

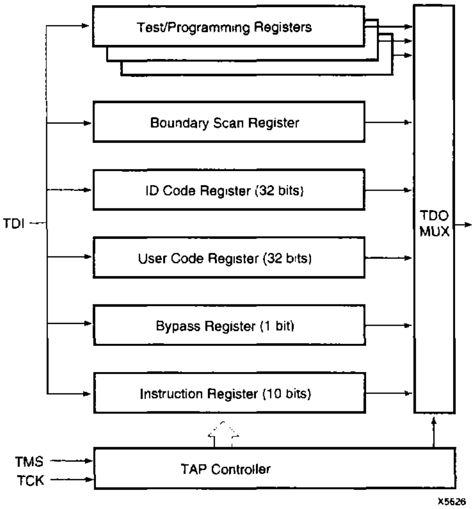

Test/Programming Registers
Boundary Scan Register
ID Code Register (32 bits)
User Code Register (32 bits)
Bypass Register (1 bit)
Instruction Register (10 bits)
TDI
TDO MUX
TMS
TCK
TAP Controller
X5626

Figure 14: XC8100 IEEE 1149.1 Architecture

In hex, the IDCODEs are:

XC8100	—	X7E90093
XC8101	1X	X7E94093
XC8101	2X	X7E95093
XC8103	1X	X7E9C093
XC8103	2X	X7E9D093
XC8106	1X	X7EA8093
XC8106	2X	X7EA9093
XC8109	2X	X7B58093

USERCODE

This instruction provides access to a 32-bit data register which can be programmed by the user, for example with the design version.

BYPASS

This instruction places a single flip-flop between TDI and TDO. Its capture value is 0.

The Boundary Scan Register bits are shown in the pinout tables (The pinout tables section begins on page 25). There are three bits for each I/O pin. The bits are TS, O, and I, as shown in Figure 11, with TS closest to the TDO end. There is a fourth bit for the four GCK/I/O pins.

Power Consumption

The XC8100 has the power consumption characteristics of Xilinx CMOS FPGAs. There are two components to the power. DC quiescent power is low and almost all of the power dissipation is a function of the design speed, the number of nodes toggling, and the capacitive loading on the outputs.

Quiescent current can be minimized by attention to its six sources:

1. Using the TTL input voltage mode draws current of about 8 mA to 30 mA, worst case, depending on the device, the package, and the voltage state of the input pins. There are two components to this current. First, the reference circuit draws 8 mA worst case, 4 mA typical, independent of device. Second, each I/O pin will draw about 100 µA worst-case if the input to the pin is held at DC low. Therefore this component depends on the circuit and the package (number of I/Os available). Both components are eliminated by selecting CMOS input levels (2.2 V trip point). At 3.3 V operation, the input level must be CMOS, so this source of current is not applicable.
2. Pulling down an I/O pin with the weak pull-up transistor enabled draws about 50 µA per pin at 3.3 V, 100 µA per pin at 5 V. The weak I/O pull-ups are automatically enabled by the design software (although they are automatically turned off when an I/O is driven by the FPGA). They can be disabled within the Series 8000 software by the command: set_attribute -port *my_input_pin* pullup false.
3. Leaving a non-pulled up I/O pin floating can produce a worst-case current of about 0.5 mA per pin at 3.3 V, 1mA per pin at 5 V. This occurs when the input voltage is at the I/O trip point. To eliminate this source of quiescent current, do not allow non-pulled up pins to float. Remember the default is that all I/Os have pull-ups. The JTAG TCK pin, which by the IEEE specification may not have a pullup or pulldown, must be connected to V_{CC} or GND to eliminate it as a source of quiescent current.
4. MicroVia leakage current depends on the design and is not under user control. It is typically very small and can be calculated after the design is placed and routed. Series 8000 software has a command (report -

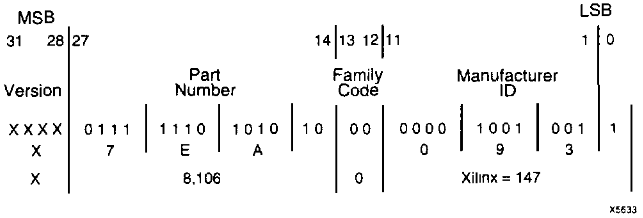


Figure 15: XC89106 IDCODE Register

stress_factor) that shows the number of antifuses to be programmed in the design and the number of "critical" antifuses. Critical antifuses are those which have the potential of being between two active and different nets. Only these antifuses can contribute to quiescent current. The worst-case theoretical MicroVia leakage current, assuming all critical antifuses have opposite logic potential on their terminals, is calculated by multiplying *each critical antifuse by 100 pA at 3.3 V, 10 nA at 5 V.* On small designs (XC8100, XC8101) this number is typically less than a few hundred µA, worst case temperature and 5.25 V. At room temperature, 5.0 V, the MicroVia leakage would be about an order of magnitude less.

5. CMOS leakage current is proportional to the device and is not under user control. It is typically a few µA. Current (May 96) test limits are a few mA for the XC8106 at 5 V with other devices proportionally less or more.

6. A small current is used by an internal voltage detector on the VPP pin. With VPP tied to VCC, this current is about 100 µA at 5 V, 30 µA at 3.3 V. A larger current exists if VPP is tied to ground. A minor enhancement is scheduled for 2H 1996 to eliminate this source of current.

Attention to these factors can result in a typical quiescent current of a few µAs for small designs at 3.3 V.

Almost all power is dynamic, and is determined by the number of nodes, their capacitance, and the frequency they are discharging and charging. This number is very design dependent, especially on the number and frequency of inputs that are toggling, not just the circuit implementation in the FPGA.

Figure 16 is a test design that illustrates the power consumption of an XC8106 at 5 V and 3.3 V. The design uses 1550 CLCs (90% of the XC8106), with 10% sequential logic and 90% combinational. All outputs are unloaded. The design consists of a 16-bit counter, a 16-bit 4-to-1 MUX, and a 16-bit multiplier. The select lines, by controlling one of the multiplier inputs, determine the number of nodes that switch.

3.3 V Operation

XC8100 devices can be operated at 5 V or 3.3 V. At 3.3, the *timing parameters are derated (see "Device Specifications"* on page 33). I/O pins cannot be directly driven above V_{CC}, although there are standard resistor solutions. The CMOS/ TTL input voltage choice should be set in the software to CMOS.

Programming

Programming XC8100 FPGAs is supported by several methods: 1) the Xilinx HW-130 Programmer; 2) selected third-party programmers; 3) distributors; and 4) Xilinx for volume designs. Series 8000 software has a Demo Mode where no key is needed. The software can be copied for programming XC8100 devices in parallel using multiple PCs or workstations. Another option is to run multiple Windows 95 sessions to multiple serial ports.

On the device, only 5 pins are required for the programming interface. The JTAG pins TDI, TDO, TCK, and TMS are used for addressing the elements to be programmed, for shifting the programming data in, for verification, and for testing. The Vpp pin provides a high voltage during programming and is used to measure the resistance of the programmed antifuse.

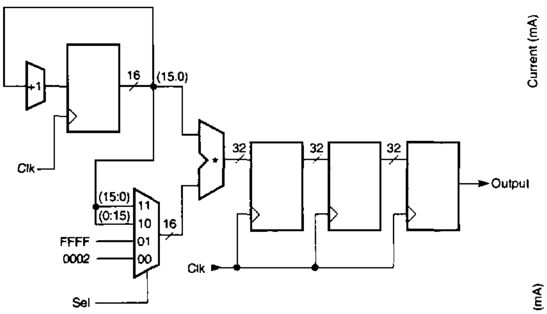
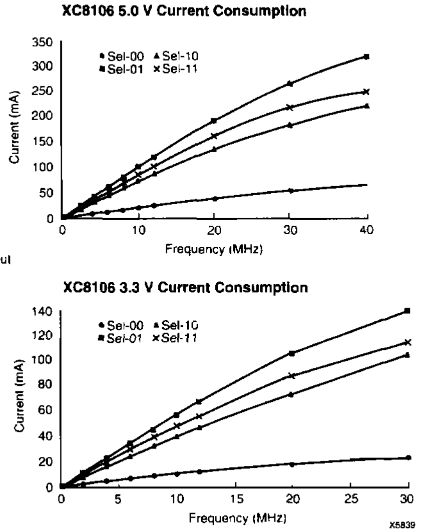
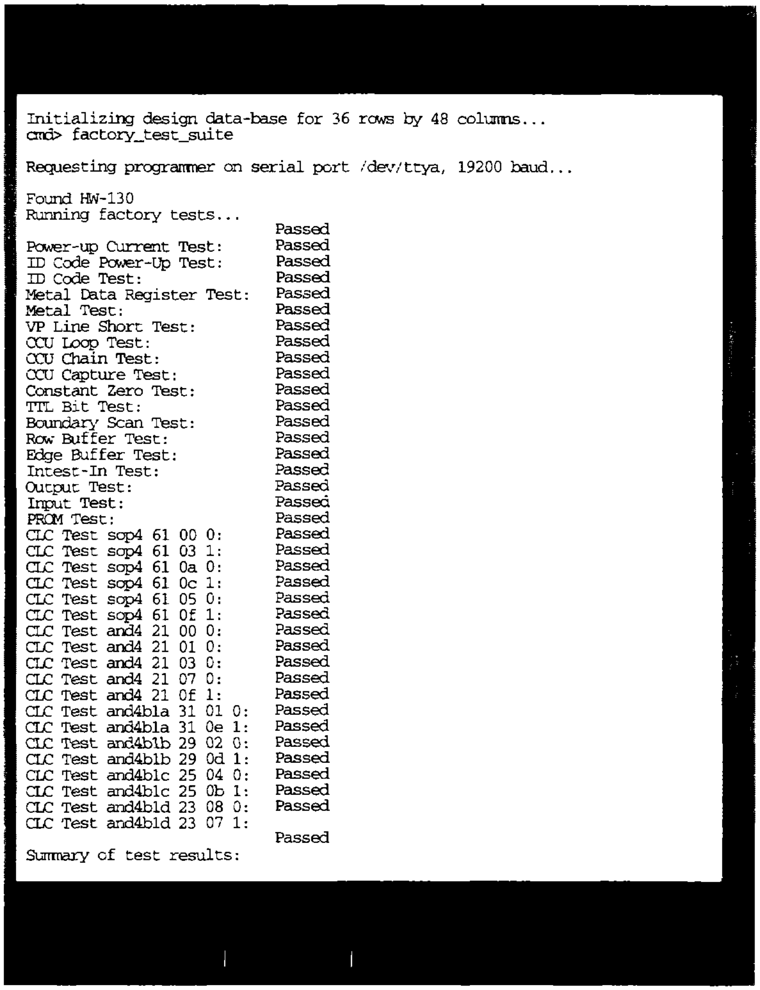


Figure 16: XC8106 Example Design





```
Initializing design data-base for 36 rows by 48 columns...
cmd> factory_test_suite

Requesting programmer on serial port /dev/ttya, 19200 baud...

Found HW-130
Running factory tests...
                              Passed
Power-up Current Test:        Passed
ID Code Power-Up Test:        Passed
ID Code Test:                 Passed
Metal Data Register Test:     Passed
Metal Test:                   Passed
VP Line Short Test:           Passed
CCU Loop Test:                Passed
CCU Chain Test:               Passed
CCU Capture Test:             Passed
Constant Zero Test:           Passed
TTL Bit Test:                 Passed
Boundary Scan Test:           Passed
Row Buffer Test:              Passed
Edge Buffer Test:             Passed
Intest-In Test:               Passed
Output Test:                  Passed
Input Test:                   Passed
PROM Test:                    Passed
CLC Test sop4 61 00 0:        Passed
CLC Test sop4 61 03 1:        Passed
CLC Test sop4 61 0a 0:        Passed
CLC Test sop4 61 0c 1:        Passed
CLC Test sop4 61 05 0:        Passed
CLC Test sop4 61 0f 1:        Passed
CLC Test and4 21 00 0:        Passed
CLC Test and4 21 01 0:        Passed
CLC Test and4 21 03 0:        Passed
CLC Test and4 21 07 0:        Passed
CLC Test and4 21 0f 1:        Passed
CLC Test and4b1a 31 01 0:     Passed
CLC Test and4b1a 31 0e 1:     Passed
CLC Test and4b1b 29 02 0:     Passed
CLC Test and4b1b 29 0d 1:     Passed
CLC Test and4b1c 25 04 0:     Passed
CLC Test and4b1c 25 0b 1:     Passed
CLC Test and4b1d 23 08 0:     Passed
CLC Test and4b1d 23 07 1:
                              Passed
Summary of test results:
```

X5937

Figure 17: Series 8000 Software Testing a Device in an HW-130

The programming algorithm includes measuring the resistance of the programmed MicroVia antifuse to guarantee the speed and functionality of the part. Since this is the one aspect that cannot be verified during factory test, it completes the 100% functional testing of the device.

There are several unique capabilities in the XC8100 programming architecture. They are aimed at addressing the issues of accurate programming and 100% post-programming yield. Series 8000 programming software takes advantage of a unique XC8100 feature that each and every antifuse can be randomly addressed. The capabilities include:

- Series 8000 software can run a complete functional factory test on an unprogrammed part in a programmer using the JTAG port. Figure 17 is a screen shot of the Series 8000 software running the test suite on an unprogrammed device
- As noted above, during programming the actual resistance of each programmed fuse is measured;
- To guarantee post-programming yield, after programming all nets are checked to see that only intended nets have been programmed, i.e. no unintended antifuses have been programmed. This is done without user test vectors;
- While these capabilities ensure that XC8100 devices are 100% testable without vectors, it is possible to apply post-programming test vectors using the programmer hardware. The XC8100 software can take simulation vectors and apply them to the device and read back the results through the JTAG port.

Testing

The functionality of gate arrays and older one-time programmable (OTP) PLDs has to be verified with test vectors after personalization. Even then, fault coverage is often below the standards of off-the-shelf devices. XC8100 FPGAs address this problem through the following:

- An architecture designed for testability.
- The use of extensive on-chip test circuitry.
- A novel method of post-programming net verification.

The test circuitry is used in conjunction with a set of special test instructions which are written to the dedicated JTAG port. This circuitry allows each device function — except the actual antifuses to be programmed — to be separately isolated and 100% tested before shipping. The programmed antifuses are later verified by the programmer. The result is the XC8100 family achieves the 100% tested level of Xilinx SRAM-based reprogrammable devices.

In concept, XC8100 FPGAs are tested in four stages. The first three are done at the factory before programming, the last by the programming hardware. First, both the functionality and speed performance of all CMOS logic on the device are verified. This includes programming circuitry, the CLC logic, the IOC logic, and the long line and clock buffers. The inputs and outputs of all the internal logic are accessible for 100% testing. Second, all metal interconnect lines are tested. Special patented circuitry allows all metal lines to be accessible. Third, all antifuses are stressed and tested to be in the correct state (off) before programming. Fourth, during programming the programmer checks that all interconnect elements meet their resistance specification after programming. The current through each programmed antifuse is measured so that the resistance meets the speed specification. Moreover, all nets are tested to ensure no antifuses have been inadvertently programmed.

Metastability Calculation

Whenever a clocked flip-flop synchronizes an asynchronous input, there is a small probability that the flip-flop will exhibit an unpredictable clock-to-Q delay. This occurs when the input transition not only violates the setup and hold time specifications, but actually occurs within the tiny timing window where the flip-flop accepts the new input. This results in the flip-flop's output being between a logic zero and logic one – a "metastable" state. The time required to transition from the metastable state to a valid logic one or zero is the delay. The Mean Time Between Failure (MTBF) for metastability is defined statistically. Figure 19 shows the data for the XC8100 using a 1 MHz data rate and a 10 MHz clock.

The XC8100 offers two options in designing for metastability. The standard XC8100 flip-flops (FD, FDC, FDP) provide MTBFs that are in the range of programmable logic devices today. An additional option is the four-CLC, double-synchronized versions of the flip-flop elements, (FD_SYNC), which automatically use two flip-flops connected through cascade. This option has one additional clock cycle of latency. This may make sense for certain signals given the large number of CLCs/FFs on XC8100 FPGAs. Figure 19 shows that for these flip-flops, a delay of 3 ns has a MTBF over 100 years.

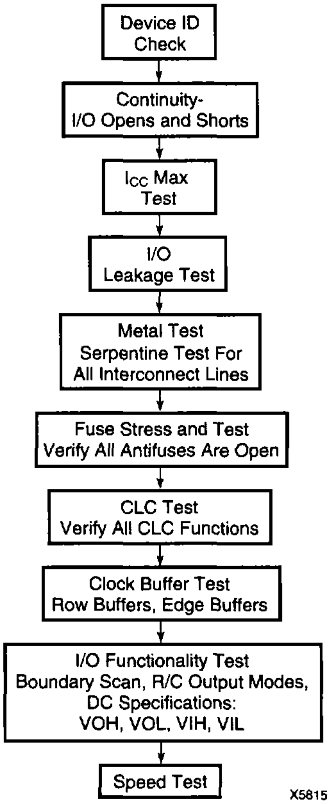

Device ID Check
Continuity- I/O Opens and Shorts
I_{CC} Max Test
I/O Leakage Test
Metal Test Serpentine Test For All Interconnect Lines
Fuse Stress and Test Verify All Antifuses Are Open
CLC Test Verify All CLC Functions
Clock Buffer Test Row Buffers, Edge Buffers
I/O Functionality Test Boundary Scan, R/C Output Modes, DC Specifications: VOH, VOL, VIH, VIL
Speed Test
X5815

Figure 18: XC8100 Factory Test Flow Chart Used for the Factory Test of all XC8100 FPGAs

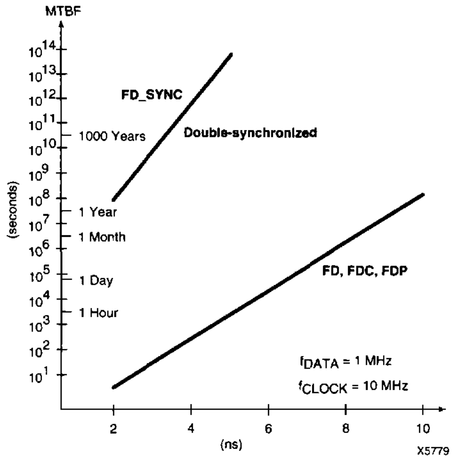


Figure 19: Metastable MTBF

XACT*step* Series 8000 Development System

The XC8100 development system, part of the Xilinx XACT-step system, employs many new features to take advantage of the new architecture:

- command shell and unified database structure
- new placement and routing software
- ASIC-like design flow
- 1-to-1 mapping between input netlist and CLCs in the device: TrueMap Logic
- maintains hierarchical information throughout the design flow
- maintains original net and instance names throughout the flow
- EDIF and SDF for CAE tool interfacing

Similar to other Xilinx FPGAs, the XC8100 FPGA design flow is based on a four-step process:

1. Design entry and functional verification using standard CAE tools.
2. Design implementation using XC8100 tools.
3. Design verification using standard CAE tools.
4. For the XC8100, programming is the fourth step.

Design Entry

The design is first described using a variety of methods: Verilog, VHDL, schematic, equations, or state machine. The designer uses standard CAE tools in this phase — Xilinx supplies a design kit (synthesis models, schematic library) for the third-party CAE vendor's tool. Some CAE vendors supply the design kit themselves. The flexibility of the XC8100 architecture means that the user can design with little concern for the architecture or implementation details. In most cases, the software automatically chooses all the capabilities described in the Architecture section.

Design Implementation

The second step in the process takes an EDIF netlist output from the design entry stage and:

- reads the design netlist, including hierarchy.
- places the netlist primitives on specific CLCs without changing any of the logic. This insures a 1-1 correspondence between design and implementation.
- routes all nets. The router uses rip-up-and-retry algorithms to shift congestion.
- calculates actual timing delays for all nets. This information is used for timing analysis and by third party timing/simulation tools.
- shows a view of the design at the physical level. Figure 21 is an output of the Viewer, showing the dense routing used to implement a 6502 VHDL model in an XC8106.
- generates fusemap and files for the programmer hardware.

Design Verification

The third step lets the designer use third-party CAE tools for analysis or simulation of the timings exported by Series 8000 software.

Programming

All programming software is included in XACT*step* Series 8000. The software connects to the programmer hardware through a serial port.

Platforms

The XC8100 software runs under Windows 3.1, Windows 95, Windows NT, SunOS 4.1.1 or later, Solaris 2.4 or later, HPUX 9.05 or later, and AIX 3.2 or later. Platforms include PC, Sparcstation, HP PA, and RS6000.

Workstation General Requirements

- minimum 32 MB of memory
- hard disk with at least 40 MB available for XC8100 programs, symbol libraries and data files
- minimum swap space of 100 MB
- color console monitor (any text terminal is sufficient when using only the XC8100 command shell)
- two or three-button mouse (no mouse is required when using only XC8100 command shell)
- postscript printer such as the Apple Laser Writer II

Sun Sparcstation

- SunOS 4.1.1 or later, Solaris 2.4 or later
- access to CD ROM drive
- Motif Window Manager or OpenLook Window

HP PA series

- access to CD ROM drive
- HPUX 9.0 or later
- HP VUE 3.0 or later

RS 6000 series

- access to CD ROM drive
- AIX 3.2 or later

IBM Compatible PC's

- Run on 386 or 486, Pentium recommended
- 8 MB RAM (small designs), 16 MB recommended
- hard disk space varies with the selected installation: 8 MB min, 13 MB typ, and 57 MB full
- color monitor capable of running in VGA mode or better
- two or three button Windows-compatible mouse
- Windows driver program for graphics adapter and display
- access to CD ROM drive

HW-130 Programmer

- PC or workstation serial port

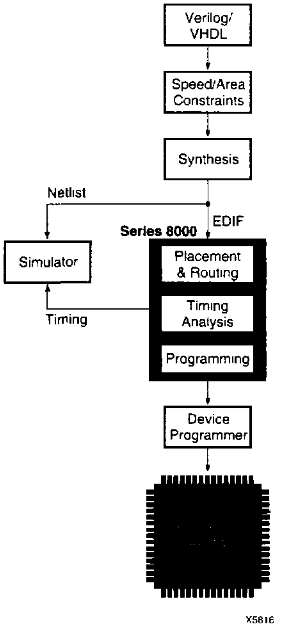


Figure 20: XC8100 Synthesis Design Flow

Series 8000 Software Ordering Information

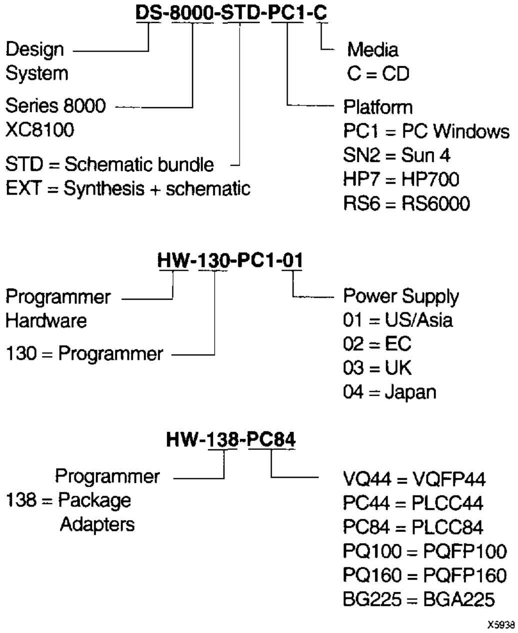
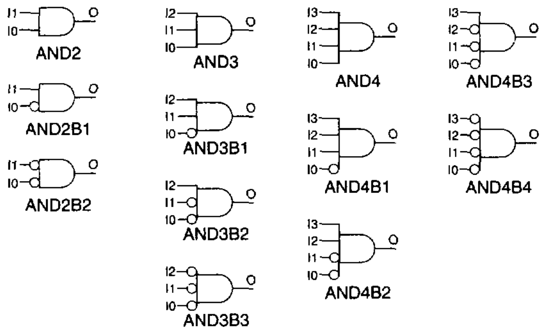


Figure 21: XC8100 Architecture in Software Viewer

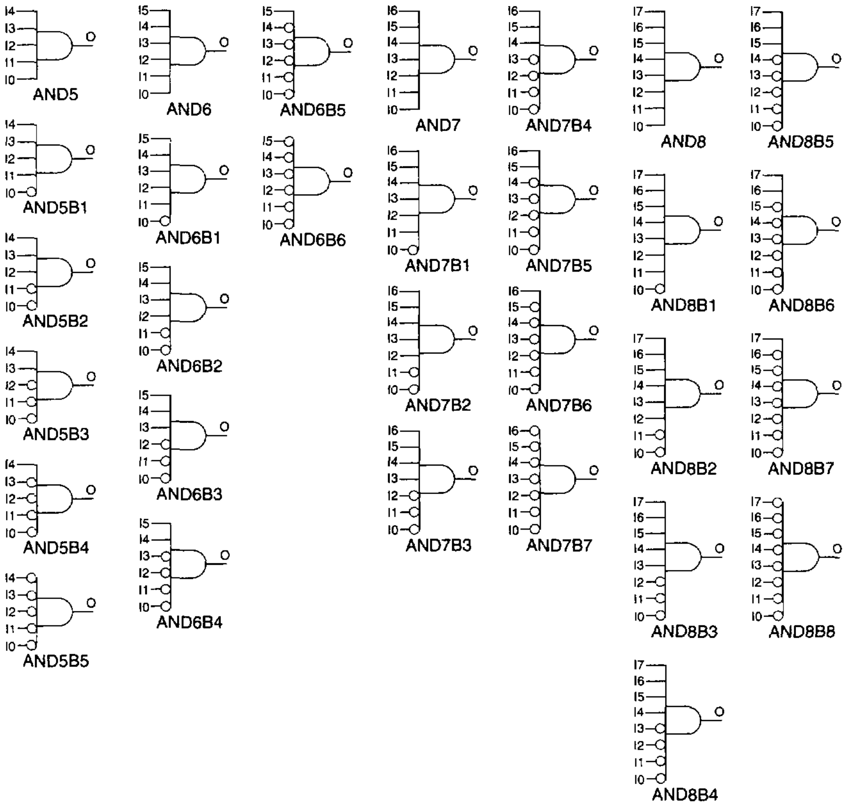
XC8100 Synthesis Library

The XC8100 synthesis library is the set of primitives used by synthesis CAE tools to generate the gate-level EDIF net list for the XC8100 software tools. It does not include macros used for schematic design entry. The library uses the standardized conventions of the Xilinx unified library.

AND Gates (1 CLC)

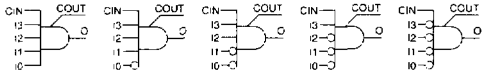


AND Gates (2 CLCs)



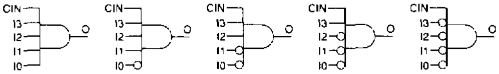
ANDCC – AND with Cascade In and Out

1 CLC



ANDCC ANDCCB1 ANDCCB2 ANDCCB3 ANDCCB4

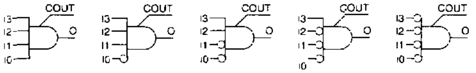
ANDCX – AND with Cascade In



ANDCX ANDCXB1 ANDCXB2 ANDCXB3 ANDCXB4

ANDXC – AND with Cascade Out

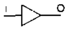
1 CLC



ANDXC ANDXCB1 ANDXCB2 ANDXCB3 ANDXCB4

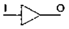
BUF – Clock/Net Buffers

0 CLCs

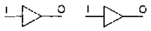


BUFROW

1 CLC



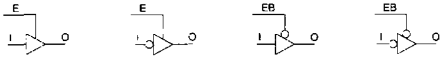
BUF1X



BUFGS BUFGP

BUFE – Three-State Buffers

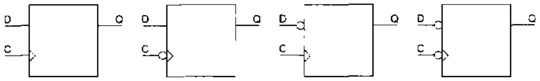
1 CLC



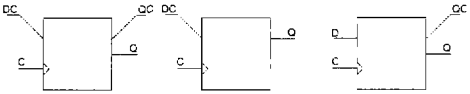
BUFE BUFEB1A BUFEB1B BUFEB2

FD – D Flip-Flops

2 CLCs



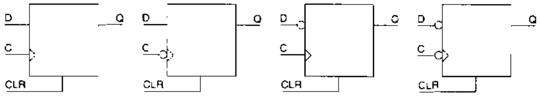
FD FD_1 FD_2 FD_3



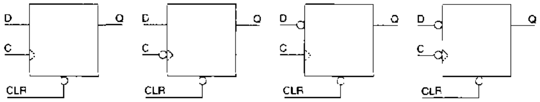
FD_CC FD_CX FD_XC

FDC – D Flip-Flop with Asynchronous Clear

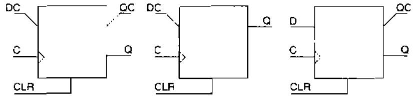
2 CLCs



FDC FDC_1 FDC_2 FDC_3



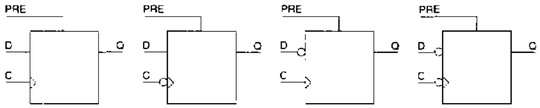
FDC_4 FDC_5 FDC_6 FDC_7



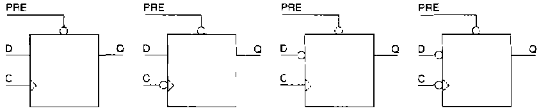
FDC_CC FDC_CX FDC_XC

FDP – D Flip-Flop with Asynchronous Preset

2 CLCs



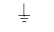
FDP FDP_1 FDP_2 FDP_3



FDP_4 FDP_5 FDP_6 FDP_7

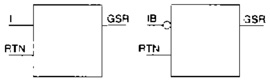
GND – Ground Signal Tag

0 CLCs



GRST

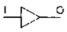
0 CLCs



GRST GRSTB1

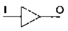
IBUF, DBUF – Input Pad Buffers

0 CLCs



IBUF

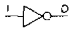
0 CLCs



DBUF

INV – Inverting Buffers

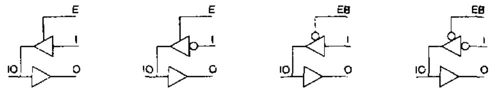
1 CLC



INV1X

IOBUF, DBUF, IRBUF, DRBUF – Bidirectional Three-State Pad Buffers

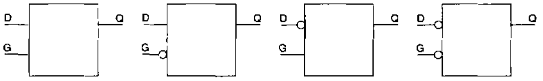
0 CLCs



IOBUF	IOBUFB1A	IOBUFB1B	IOBUFB2
DOBUF	DOBUFB1A	DOBUFB1B	DOBUFB2
IRBUF	IRBUFB1A	IRBUFB1B	IRBUFB2
DRBUF	DRBUFB1A	DRBUFB1B	DRBUFB2

LD – Transparent Data Latches

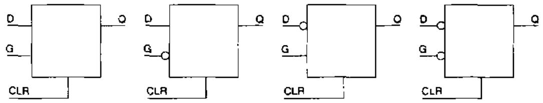
1 CLC



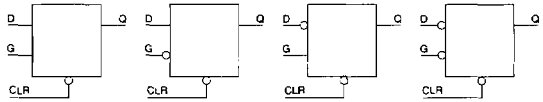
LD LD_1 LD_2 LD_3

LDC – Transparent Data Latch with Asynchronous Clear

2 CLCs



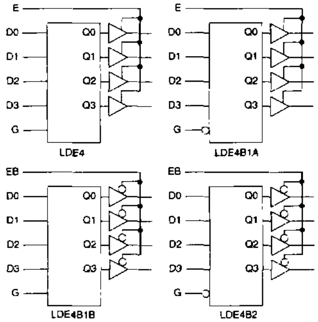
LDC LDC_1 LDC_2 LDC_3



LDC_4 LDC_5 LDC_6 LDC_7

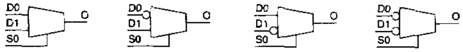
LDE4 – Quad Latch Bank with Three-State Output

4 CLCs



M2_1 – 2 to 1 Multiplexers

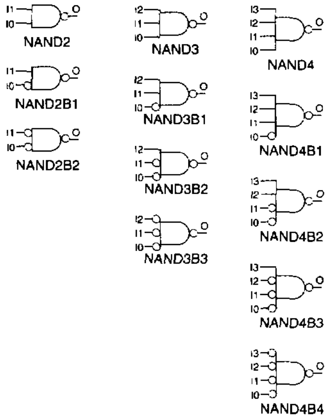
1 CLC



M2_1 M2_1B1A M2_1B1B M2_1B2

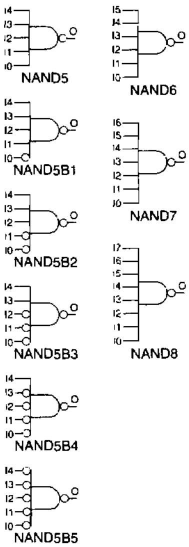
NAND Gates (1 CLC)

1 CLC



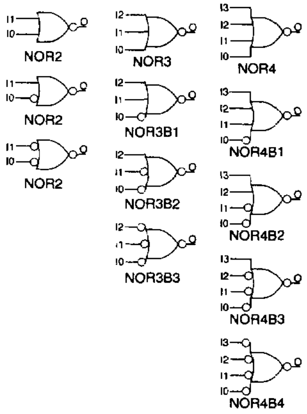
NAND Gates (2 CLCs)

2 CLCs



NOR Gates (1 CLC)

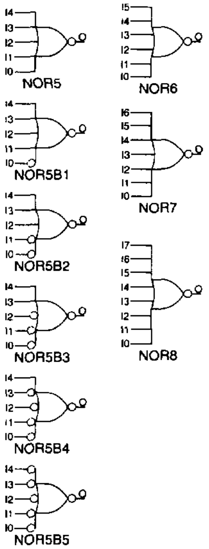
1 CLC



NOR2 NOR3 NOR4
NOR2 NOR3B1 NOR4B1
NOR2 NOR3B2 NOR4B2
NOR3B3 NOR4B3
NOR4B4

NOR Gates (2 CLCs)

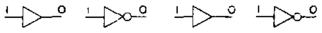
2 CLCs



NOR5 NOR6
NOR5B1 NOR7
NOR5B2 NOR8
NOR5B3
NOR5B4
NOR5B5

OBUF, RBUF – Output Pad Buffers (C, R modes)

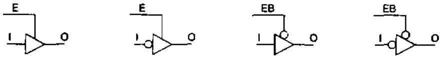
0 CLCs



OBUF OBUFB1 RBUF RBUFB1

OBUFE, RBUFE – Output Pad Three-State Buffers

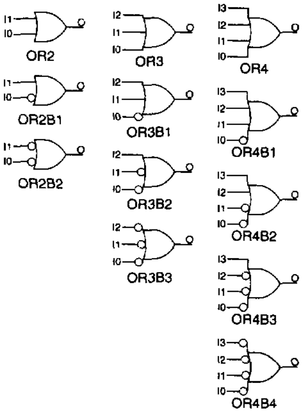
0 CLCs



OBUFE RBUFE
OBUFEB1A RBUFEB1A
OBUFEB1B RBUFEB1B
OBUFEB2 RBUFEB2

OR Gates (1 CLC)

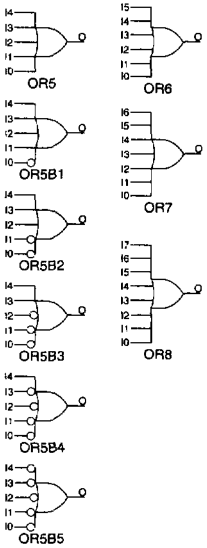
1 CLC



OR2 OR3 OR4
OR2B1 OR3B1 OR4B1
OR2B2 OR3B2 OR4B2
OR3B3 OR4B3
OR4B4

OR Gates (2 CLCs)

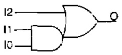
2 CLCs



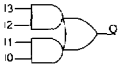
OR5 OR6
OR5B1 OR7
OR5B2 OR8
OR5B3
OR5B4
OR5B5

SOP – Sum of Products (1 CLC)

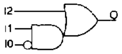
1 CLC



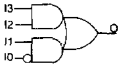
SOP3



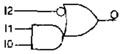
SOP4



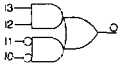
SOP3B1A



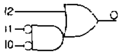
SOP4B1



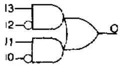
SOP3B1B



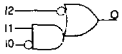
SOP4B2A



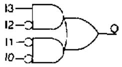
SOP3B2A



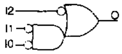
SOP4B2B



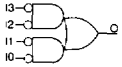
SOP3B2B



SOP4B3



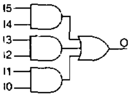
SOP3B3



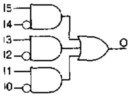
SOP4B4

SOP – Sum of Products (2 CLCs)

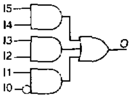
2 CLCs



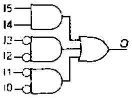
SOP6



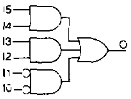
SOP6B3B



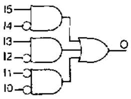
SOP6B1



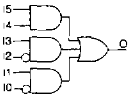
SOP6B4A



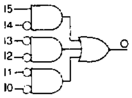
SOP6B2A



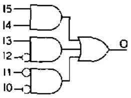
SOP6B4B



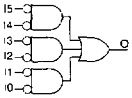
SOP6B2B



SOP6B5



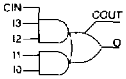
SOP6B3A



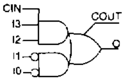
SOP6B6

SOPCC – Sum of Products with Cascade In and Out

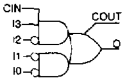
1 CLC



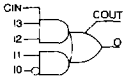
SOPCC



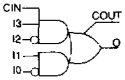
SOPCCB2A



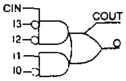
SOPCCB3A



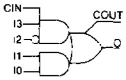
SOPCCB1A



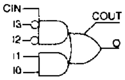
SOPCCB2B



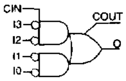
SOPCCB3B



SOPCCB1B



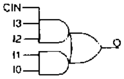
SOPCCB2C



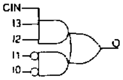
SOPCCB4

SOPCX – Sum of Products with Cascade In

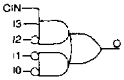
1 CLC



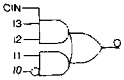
SOPCX



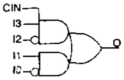
SOPCXB2A



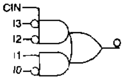
SOPCXB3A



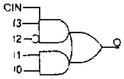
SOPCXB1A



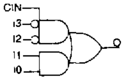
SOPCXB2B



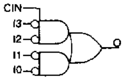
SOPCXB3B



SOPCXB1B



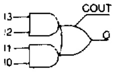
SOPCXB2C



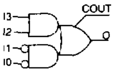
SOPCXB4

SOPXC – Sum of Products with Cascade Out

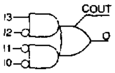
1 CLC



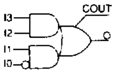
SOPXC



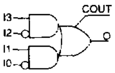
SOPXCB2A



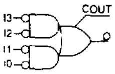
SOPXCB3



SOPXCB1



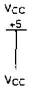
SOPXCB2B



SOPXCB4

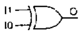
VCC – V_{CC} Signal Tag

0 CLCs

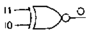


XOR2, XNOR2 – Two-Input Exclusion OR/NOR

1 CLC



XOR2



XNOR2

Pin Descriptions

V_{CC}

Two or more, depending on the package type. All must be connected to the +5 V/3.3 V supply voltage.

V_{PP}

V_{pp} is the programming voltage. This pin can be left floating, but will draw slightly less ICCO if connected to V_{CC} during operation.

GND

Four or more, depending on the package type. All must be connected to ground.

GCK1 - GCK4 - I/O

Four global clock inputs each connect to a dedicated internal global buffer (bufedge) with short delay and minimal skew. If not used for this purpose, these pins are user I/O.

TDO

Test Data Output pin for JTAG operation and testing. This is a dedicated pin and is not available for user I/O. It has no pull-up or pull-down.

TDI, TCK, TMS

Test Data In, Test Clock, and Test Mode Select inputs for JTAG boundary scan, programming, and testing. These are dedicated pins and are not available for user I/O. TMS and TDI have pull-ups, TCK does not.

$\overline{\text{MASTER RESET}}$

This active-low pin has the same functionality as removing V_{CC} and then reapplying power. It resets all synchronous logic. Master Reset three-states all I/O pins while held low, and can be useful for board testing. The pin has an internal pull-up resistor, so no external resistor is needed if the pin is unused. All synchronous logic is reset time TMRQ after $\overline{MR}$ is raised high. It is recommended that $\overline{MR}$ on the XC8100 (625 gate) device have a 1kΩ resistor in series to limit current from signals that violate the V_{IN} specification of -0.5V.

I/O

These pins are configured by the user to be either input or output. Programmable options include input voltage levels (CMOS or TTL on a per-chip basis) and output slew rate (resistive or capacitive mode on a per-pin basis). If an I/O is not used, an internal pull-up resistor is automatically enabled, so no external resistor is required.

XC8100 Pin Assignments

The XC8100 pinout is based on the XC4000 pinouts. This means that, for any package, power and control pins are on the same pins as in the XC4000. However, not all the control pins have the same function. Typically the XC4000 M0/M1/M2 pins are fixed. If JTAG is not used on the XC8100, this is compatible. DONE can be an input or an output. If it's an input, the XC8100 trace would have to be cut. The V_{PP} pin is compatible with either CCLK as an input or outputs, although V_{PP} has a larger input capacitance. Following is the mapping:

Type	XC8100	XC4000
V_{CC}	Same as XC4000	–
GND	Same as XC4000	–
Control	TMS	M0
	TCK	M1
	TDI	M2
	TDO	TDO
	V_{PP}	CCLK
	MR	DONE

Number of Available I/O Pins

Device	Packages						
	Max I/O	PC44	VQ44	PC84	PQ100	PQ160	BG225
XC8100	32	32	32	–	–	–	–
XC8101	72	(32)	(32)	61	72	–	–
XC8103	128	32	32	61	64	(128)	–
XC8106	168	–	–	61	76	(129)	(168)
XC8109	192	–	–	61	–	129	192

Note: Parentheses indicates future products

XC8100 Pinouts

Pin Description	PC44	VQ44	Bound Scan
VCC	2	40	-
I/O	3	41	69
I/O	4	42	72
I/O	5	43	75
I/O	6	44	78
I/O	7	1	81
I/O	8	2	84
GCK1-I/O	9	3	87
I/O	10	4	91
GND	11	5	–
I/O	12	6	94
I/O	13	7	97
TCK	14	8	–
TMS	15	9	–
TDI	16	10	–
GCK2-I/O	17	11	0
I/O	18	12	4
I/O	19	13	7
I/O	20	14	10
I/O	21	15	13
I/O	22	16	16
VCC	23	17	–
GND	24	18	–
I/O	25	19	19
I/O	26	20	22
I/O	27	21	25
I/O	28	22	28
I/O	29	23	31
I/O	30	24	34
MR	31	25	–
I/O	32	26	38
GCK3-I/O	33	27	41
GND	34	28	–
I/O	35	29	44
I/O	36	30	47
VPP	37	31	–
TDO	38	32	–
I/O	39	33	51
GCK4-I/O	40	34	54
I/O	41	35	57
I/O	42	36	60
I/O	43	37	63
I/O	44	38	66
GND	1	39	–

XC8101 Pinouts

Pin Description	PC84	PQ100	Bound Scan
VCC	2	92	-
I/O	3	93	159
I/O	4	94	162
I/O	-	95	165
I/O	-	96	168
I/O	5	97	171
I/O	6	98	174
I/O	7	99	177
I/O	8	100	180
I/O	-	-	193
I/O	-	-	196
I/O	9	1	189
I/O	10	2	192
VCC	11	3	-
GND	12	4	-
GCK1-I/O	13	5	195
I/O	14	6	199
I/O	15	7	202
I/O	16	8	205
I/O	17	9	208
I/O	18	10	211
NC	-	11	-
I/O	19	12	214
I/O	20	13	217
GND	21	14	-
VCC	22	15	-
I/O	23	16	220
I/O	24	17	223
NC	-	18	-
I/O	25	19	226
I/O	26	20	229
I/O	27	21	232
I/O	-	22	235
I/O	28	23	238
I/O	29	24	241
TCK	30	25	-
GND	31	26	-
TMS	32	27	-
VCC	33	28	-
TDI	34	29	-
GCK2-I/O	35	30	0
I/O	36	31	4
I/O	-	-	7
I/O	-	-	10
I/O	-	32	13
I/O	37	33	16
I/O	38	34	19
I/O	39	35	22
I/O	-	36	25
I/O	-	37	28
I/O	40	38	31
I/O	41	39	34
VCC	42	40	-
GND	43	41	-
I/O	44	42	37
I/O	45	43	40
I/O	-	44	43
I/O	-	45	46
I/O	46	46	49
I/O	47	47	52
I/O	48	48	55
I/O	49	49	58
I/O	-	-	61
I/O	-	-	64
I/O	50	50	67
I/O	51	51	70
GND	52	52	-
MR	53	53	-
VCC	54	54	-
NC	55	55	-
I/O	56	56	74
GCK3-I/O	57	57	77
I/O	58	58	80
I/O	-	59	83
I/O	59	60	86
I/O	60	61	89
NC	-	62	-
NC	-	63	-
I/O	61	64	92
I/O	62	65	95
VCC	63	66	-
GND	64	67	-
I/O	65	68	98
I/O	66	69	101
NC	-	70	-
I/O	67	71	104
I/O	68	72	107
I/O	69	73	110
I/O	70	74	113
I/O	71	75	116
I/O	72	76	119
VPP	73	77	-
VCC	74	78	-
TDO	75	79	-
GND	76	80	-
I/O	77	81	123
GCK4-I/O	78	82	126
I/O	-	-	129
I/O	-	-	132
I/O	79	83	135
I/O	80	84	138
I/O	81	85	141
I/O	82	86	144
I/O	-	87	147
I/O	-	88	150
I/O	83	89	153
I/O	84	90	156
GND	1	91	-

XC8103 Pinouts

Pin Description	PC84	PQ100	Bound Scan
VCC	2	92	-
I/O	3	93	123
I/O	4	94	126
NC	-	95	-
NC	-	96	-
I/O	5	97	129
I/O	6	98	132
-	-	-	-
-	-	-	-
GND	-	-	-
-	-	-	-
-	-	-	-
I/O	7	99	135
I/O	8	100	138
I/O	-	-	-
I/O	-	-	-
I/O	9	1	141
I/O	10	2	144
VCC	11	3	-
GND	12	4	-
GCK1-I/O	13	5	147
I/O	14	6	151
I/O	-	-	-
I/O	-	-	-
I/O	15	7	154
I/O	16	8	157
-	-	-	-
-	-	-	-
GND	-	-	-
-	-	-	-
-	-	-	-
I/O	17	9	160
I/O	18	10	163
NC	-	11	-
I/O	-	-	-
I/O	19	12	166
I/O	20	13	169
GND	21	14	-
VCC	22	15	-
I/O	23	16	172
I/O	24	17	175
NC	-	18	-
I/O	-	-	-
I/O	25	19	178
I/O	26	20	181
-	-	-	-
-	-	-	-
GND	-	-	-
-	-	-	-
-	-	-	-
I/O	27	21	184
I/O	-	22	187
I/O	-	-	-
I/O	-	-	-
I/O	28	23	190
I/O	29	24	193
TCK	30	25	-
GND	31	26	-
TMS	32	27	-
VCC	33	28	-
TDI	34	29	-
GCK2-I/O	35	30	0
I/O	36	31	4
I/O	-	-	-
I/O	-	-	-
I/O	-	32	7
I/O	37	33	10
-	-	-	-
-	-	-	-
GND	-	-	-
-	-	-	-
-	-	-	-
I/O	38	34	13
I/O	39	35	16
NC	-	36	-
NC	-	37	-
I/O	40	38	19
I/O	41	39	22
VCC	42	40	-
GND	43	41	-
I/O	44	42	25
I/O	45	43	28
NC	-	44	-
NC	-	45	-
I/O	46	46	31
I/O	47	47	34
-	-	-	-
-	-	-	-
GND	-	-	-
-	-	-	-
-	-	-	-
I/O	48	48	37
I/O	49	49	40
I/O	-	-	-
I/O	-	-	-
I/O	50	50	43
I/O	51	51	46
GND	52	52	-
MR	53	53	-
VCC	54	54	-
NC	55	55	-
I/O	56	56	50
GCK3-I/O	57	57	53
I/O	-	-	-
I/O	-	-	-
I/O	58	58	56
I/O	-	59	59
-	-	-	-
-	-	-	
GND	-	-	
-	-	-	
-	-	-	
I/O	59	60	62
I/O	60	61	65
NC	-	62	-
NC	-	63	-
I/O	61	64	68
i/O	62	65	71
VCC	63	66	-
GND	64	67	-
I/O	65	68	74
I/O	66	69	77
NC	-	70	-
I/O	-	-	-
I/O	67	71	80
I/O	68	72	83
-	-	-	-
-	-	-	-
GND	-	-	-
-	-	-	-
-	-	-	-
I/O	69	73	86
I/O	70	74	89
I/O	-	-	-
I/O	-	-	-
I/O	71	75	92
I/O	72	76	95
VPP	73	77	-
VCC	74	78	-
TDO	75	79	-
GND	76	80	-
I/O	77	81	99
GCK4-I/O	78	82	102
I/O	-	-	-
I/O	-	-	-
I/O	79	83	105
I/O	80	84	108
-	-	-	-
-	-	-	-
GND	-	-	-
-	-	-	-
-	-	-	-
I/O	81	85	111
I/O	82	86	114
-	-	-	-
NC	-	87	-
NC	-	88	-
I/O	83	89	117
I/O	84	90	120
GND	1	91	-

XC8106 Pinouts

Pin			Bound		Pin			Bound		Pin			Bound	
Description	PC84	PQ100	Scan	PQ160	Description	PC84	PQ100	Scan	PQ160	Description	PC84	PQ100	Scan	PQ160
VCC	2	92	-	142	I/O	28	23	250	36	I/O	-	-	-	90
I/O	3	93	165	143	I/O	29	24	253	37	GND	-	-	-	91
I/O	4	94	168	144	TCK	30	25	-	38	I/O	-	-	-	92
I/O	-	95	171	145	GND	31	26	-	39	I/O	-	-	-	93
I/O	-	96	174	146	TMS	32	27	-	40	I/O	59	60	86	94
I/O	5	97	177	147	VCC	33	28	-	41	I/O	60	61	89	95
I/O	6	98	180	148	TDI	34	29	-	42	I/O	-	-	-	96
I/O	-	-	-	149	GCK2-I/O	35	30	0	43	NC	-	62	-	-
I/O	-	-	-	150	I/O	36	31	4	44	I/O	-	63	92	97
GND	-	-	-	151	I/O	-	-	7	45	I/O	61	64	95	98
I/O	-	-	-	152	I/O	-	-	10	46	I/O	62	65	98	99
I/O	-	-	-	153	I/O	-	32	13	47	VCC	63	66	-	100
I/O	7	99	183	154	I/O	37	33	16	48	GND	64	67	-	101
I/O	8	100	186	155	I/O	-	-	-	49	I/O	65	68	101	102
I/O	-	-	189	156	I/O	-	-	-	50	I/O	66	69	104	103
I/O	-	-	192	157	GND	-	-	-	51	I/O	-	70	107	104
I/O	9	1	195	158	I/O	-	-	-	52	I/O	-	-	-	105
I/O	10	2	198	159	I/O	-	-	-	53	I/O	67	71	110	106
VCC	11	3	-	160	I/O	38	34	19	54	I/O	68	72	113	107
GND	12	4	-	1	I/O	39	35	22	55	I/O	-	-	-	108
GCK1-I/O	13	5	201	2	I/O	-	36	25	56	I/O	-	-	-	109
I/O	14	6	205	3	I/O	-	37	28	57	GND	-	-	-	110
I/O	-	-	-	4	I/O	40	38	31	58	I/O	-	-	-	111
I/O	-	-	-	5	I/O	41	39	34	59	I/O	-	-	-	112
I/O	15	7	208	6	VCC	42	40	-	60	I/O	69	73	116	113
I/O	16	8	211	7	GND	43	41	-	61	I/O	70	74	119	114
I/O	-	-	-	8	I/O	44	42	37	62	I/O	-	-	-	115
I/O	-	-	-	9	I/O	45	43	40	63	I/O	-	-	-	116
GND	-	-	-	10	I/O	-	44	43	64	I/O	71	75	122	117
I/O	-	-	-	11	I/O	-	45	46	65	I/O	72	76	125	118
I/O	-	-	-	12	I/O	46	46	49	66	VPP	73	77	-	119
I/O	17	9	214	13	I/O	47	47	52	67	VCC	74	78	-	120
I/O	18	10	217	14	I/O	-	-	-	68	TDO	75	79	-	121
I/O	-	-	-	15	I/O	-	-	-	69	GND	76	80	-	122
I/O	-	11	220	16	GND	-	-	-	70	I/O	77	81	129	123
I/O	19	12	223	17	I/O	-	-	-	71	GCK4-I/O	78	82	132	124
I/O	20	13	226	18	I/O	-	-	-	72	I/O	-	-	135	125
GND	21	14	-	19	I/O	48	48	55	73	I/O	-	-	138	126
VCC	22	15	-	20	I/O	49	49	58	74	I/O	79	83	141	127
I/O	23	16	229	21	I/O	-	-	61	75	I/O	80	84	144	128
I/O	24	17	232	22	I/O	-	-	64	76	I/O	-	-	-	129
I/O	-	18	235	23	I/O	50	50	67	77	I/O	-	-	-	130
I/O	-	-	-	24	I/O	51	51	70	78	GND	-	-	-	131
I/O	25	19	238	25	GND	52	52	-	79	I/O	-	-	-	132
I/O	26	20	241	26	MR	53	53	-	80	I/O	-	-	-	133
I/O	-	-	-	27	VCC	54	54	-	81	I/O	81	85	147	134
I/O	-	-	-	28	NC	55	55	-	82	I/O	82	86	150	135
GND	-	-	-	29	I/O	56	56	74	83	I/O	-	-	-	136
I/O	-	-	-	30	GCK3-I/O	57	57	77	84	I/O	-	87	153	137
I/O	-	-	-	31	I/O	-	-	-	85	I/O	-	88	156	138
I/O	27	21	244	32	I/O	-	-	-	86	I/O	83	89	159	139
I/O	-	22	247	33	I/O	58	58	80	87	I/O	84	90	162	140
I/O	-	-	-	34	I/O	-	59	83	88	GND	1	91	-	141
I/O	-	-	-	35	I/O	-	-	-	89					

XC8109 Pinouts

Pin Description	PC 84	PQ 160	PG 223	BG 225	Bound Scan
VCC	2	142	J4	D8	-
I/O	3	143	J3	E8	399
I/O	4	144	J2	B7	402
I/O	-	145	J1	A7	405
I/O	-	146	H1	C7	408
I/O	-	-	H2	D7	411
I/O	-	-	H3	E7	414
I/O	5	147	G1	A6	417
I/O	6	148	G2	B6	420
I/O	-	-	-	-	423
I/O	-	-	-	-	426
I/O	-	-	-	-	429
I/O	-	-	H4	C6	432
I/O	-	-	G4	F7	435
I/O	-	149	F1	A5	438
I/O	-	150	E1	B5	441
GND	-	151	G3	**	-
I/O	-	-	-	-	444
I/O	-	-	F2	D6	447
I/O	-	-	D1	C5	450
I/O	-	152	C1	A4	453
I/O	-	153	E2	E6	456
I/O	7	154	F3	B4	459
I/O	8	155	D2	D5	462
I/O	-	-	F4	A3	465
I/O	-	-	E4	C4	468
I/O	-	156	B1	B3	471
I/O	-	157	E3	F6	474
I/O	9	158	C2	A2	477
I/O	10	159	B2	C3	480
VCC	11	160	D3	B2	-
GND	12	1	D4	A1	-
GCK1-I/O	13	2	C3	D4	483
I/O	14	3	C4	B1	487
I/O	-	4	B3	C2	490
I/O	-	5	C5	E5	493
I/O	15	6	A2	D3	496
I/O	16	7	B4	C1	499
I/O	-	8	C6	D2	502
I/O	-	9	A3	G6	505
I/O	-	-	B5	E4	508
I/O	-	-	B6	D1	511
I/O	-	-	D5	E3	514
I/O	-	-	D6	E2	517
GND	-	10	C7	**	-
I/O	-	11	A4	F5	520
I/O	-	12	A5	E1	523
I/O	17	13	B7	F4	526
I/O	18	14	A6	F3	529
I/O	-	-	D7	F2	532
I/O	-	-	D8	F1	535
I/O	-	-	C8	G4	538
I/O	-	-	A7	G3	541
I/O	-	15	B8	G2	544
I/O	-	16	A8	G1	547
I/O	19	17	B9	G5	550
I/O	20	18	C9	H3	553
GND	21	19	D9	H2	-
VCC	22	20	D10	H1	-
I/O	23	21	C10	H4	556
I/O	24	22	B10	H5	559
I/O	-	23	A9	J2	562
I/O	-	24	A10	J1	565
I/O	-	-	A11	J3	568
I/O	-	-	C11	J4	571
I/O	-	-	D11	J5	574
I/O	-	-	D12	K1	577
I/O	25	25	B11	K2	580
I/O	26	26	A12	K3	583
I/O	-	27	B12	J6	586
I/O	-	28	A13	L1	589
GND	-	29	C12	**	-
I/O	-	-	D13	L2	592
I/O	-	-	D14	K4	595
I/O	-	-	B13	L3	598
I/O	-	-	A14	M1	601
I/O	-	30	A15	K5	604
I/O	-	31	C13	M2	607
I/O	27	32	B14	L4	610
I/O	-	33	A16	N1	613
I/O	-	34	B15	M3	616
I/O	-	35	C14	N2	619
I/O	28	36	A17	K6	622
I/O	29	37	B16	P1	625
TCK	30	38	C15	N3	-
GND	31	39	D15	**	-
TMS	32	40	A18	P2	-
VCC	33	41	D16	R1	-
TDI	34	42	C16	M4	-
GCK2-I/O	35	43	B17	R2	0
I/O	36	44	E16	P3	4
I/O	-	45	C17	L5	7
I/O	-	46	D17	N4	10
I/O	-	47	B18	R3	13
I/O	37	48	E17	P4	16
I/O	-	49	F16	K7	19
I/O	-	50	C18	M5	22
I/O	-	-	D18	R4	25
I/O	-	-	F17	N5	28
I/O	-	-	E15	P5	31
I/O	-	-	F15	L6	34
I/O	-	-	-	-	37
GND	-	51	G16	**	-
I/O	-	52	E18	R5	40
I/O	-	53	F18	M6	43
I/O	38	54	G17	N6	46
I/O	39	55	G18	P6	49
I/O	-	-	-	-	52
I/O	-	-	-	-	55
I/O	-	-	-	-	58
I/O	-	-	H16	R6	61
I/O	-	-	H17	M7	64
I/O	-	-	G15	N7	67
I/O	-	-	H15	P7	70
I/O	-	56	H18	R7	73
I/O	-	57	J18	L7	76
I/O	40	58	J17	N8	79
I/O	41	59	J16	P8	82
VCC	42	60	J15	R8	-
GND	43	61	K15	M8	-
I/O	44	62	K16	L8	85
I/O	45	63	K17	P9	88
I/O	-	64	K18	R9	91
I/O	-	65	L18	N9	94
I/O	-	-	L17	M9	97
I/O	-	-	L16	L9	100
I/O	-	-	L15	R10	103
I/O	-	-	M15	P10	106
I/O	-	-	-	-	109
I/O	-	-	-	-	112
I/O	-	-	-	-	115
I/O	46	66	M18	N10	118
I/O	47	67	M17	K9	121
I/O	-	68	N18	R11	124
I/O	-	69	P18	P11	127
GND	-	70	M16	**	-
I/O	-	-	-	-	130
I/O	-	-	N15	M10	133
I/O	-	-	P15	N11	136
I/O	-	-	N17	R12	139
I/O	-	-	R18	L10	142
I/O	-	71	T18	P12	145
I/O	-	72	P17	M11	148
I/O	48	73	N16	R13	151
I/O	49	74	T17	N12	154
I/O	-	75	R17	P13	157
I/O	-	76	P16	K10	160
I/O	50	77	U18	R14	163
I/O	51	78	T16	N13	166
GND	52	79	R16	**	-
MR	53	80	U17	P14	-
VCC	54	81	R15	R15	-
NC	53	82	V18	M12	-
I/O	56	83	T15	P15	170
GCK3-I/O	57	84	U16	N14	173
I/O	-	85	T14	L11	176
I/O	-	86	U15	M13	179
I/O	-	-	R14	N15	182
I/O	-	-	R13	M14	185
I/O	58	87	V17	J10	188
I/O	-	88	V16	L12	191
I/O	-	89	T13	M15	194
I/O	-	90	U14	L13	197
I/O	-	-	V15	L14	200
I/O	-	-	V14	K11	203
GND	-	91	T12	**	-
I/O	-	-	R12	L15	206
I/O	-	-	R11	K12	209
I/O	-	92	U13	K13	212
I/O	-	93	V13	K14	215
I/O	59	94	U12	K15	218
I/O	60	95	V12	J12	221
I/O	-	-	T11	J13	224
I/O	-	-	U11	J14	227
I/O	-	96	V11	J15	230
I/O	-	97	V10	J11	233
I/O	61	98	U10	H13	236
I/O	62	99	T10	H14	239
VCC	63	100	R10	H15	-
GND	64	101	R9	**	-
I/O	65	102	T9	H12	242
I/O	66	103	U9	H11	245
I/O	-	104	V9	G14	248
I/O	-	105	V8	G15	251
I/O	-	-	U8	G13	254
I/O	-	-	T8	G12	257
I/O	67	106	V7	G11	260
I/O	68	107	U7	F15	263
I/O	-	108	V6	F14	266
I/O	-	109	U6	F13	269
I/O	-	-	R8	G10	272
I/O	-	-	R7	E15	275
GND	-	110	T7	**	-
I/O	-	-	R6	E14	278
I/O	-	-	R5	F12	281
I/O	-	-	V5	E13	284
I/O	-	-	V4	D15	287
I/O	-	111	U5	F11	290
I/O	-	112	T6	D14	293
I/O	69	113	V3	E12	296
I/O	70	114	V2	C15	299
I/O	-	115	U4	D13	302
I/O	-	116	T5	C14	305
I/O	71	117	U3	F10	308
I/O	72	118	T4	B15	311
VPP	73	119	V1	C13	-
VCC	74	120	R4	B14	-
TDO	75	121	U2	A15	-
GND	76	122	R3	D12	-
I/O	77	123	T3	A14	315
GCK4-I/O	78	124	U1	B13	318
I/O	-	125	P3	E11	321
I/O	-	126	R2	C12	324
I/O	79	127	T2	A13	327
I/O	80	128	N3	B12	330
I/O	-	-	P4	F9	333
I/O	-	-	N4	D11	336
I/O	-	129	P2	A12	339
I/O	-	130	T1	C11	342
I/O	-	-	R1	B11	345
I/O	-	-	N2	E10	348
I/O					351
GND	-	131	M3	**	-
I/O	-	132	P1	A11	354
I/O	-	133	N1	D10	357
I/O	-	-	M4	C10	360
I/O	-	-	L4	B10	363
I/O	-	-	-	-	366
I/O	-	-	-	-	369
I/O	-	-	-	-	372
I/O	81	134	M2	A10	375
I/O	82	135	M1	D9	378
I/O	-	-	L3	C9	381
I/O	-	136	L2	B9	384
I/O	-	137	L1	A9	387
I/O	-	138	K1	E9	390
I/O	83	139	K2	C8	393
I/O	84	140	K3	B8	396
GND	1	141	K4	A8	-

Note: ** These BGA225 balls are connected to ground: F8, G7, G8, G9, H6, H7, H8, H9, H10, J7, J8, J9, K8

Package Pinout Diagrams

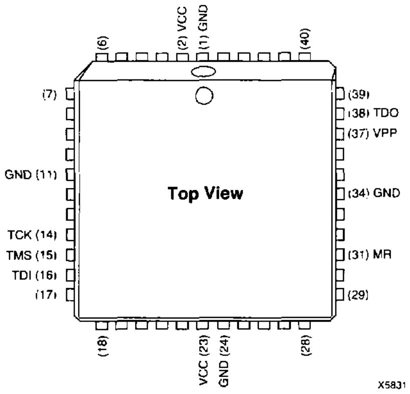


Figure 22: PC44 PLCC with 44 Leads, 50 mil Lead Pitch

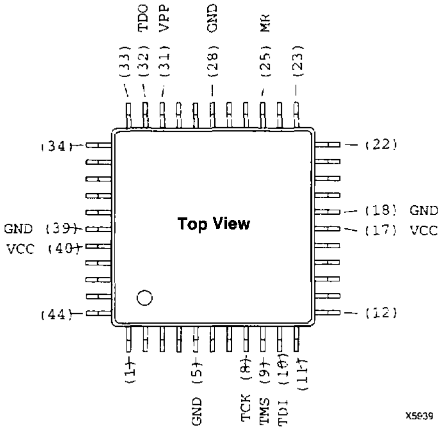


Figure 23: VQ44 VQFP with 44 Leads, 0.8 mm Lead Pitch

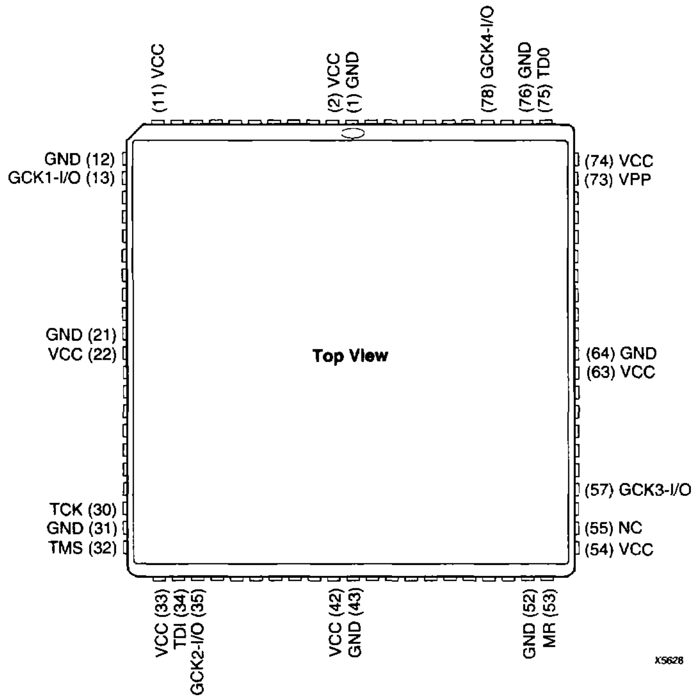


Figure 24: PC84 PLCC with 84 Leads, 50 mil Lead Pitch

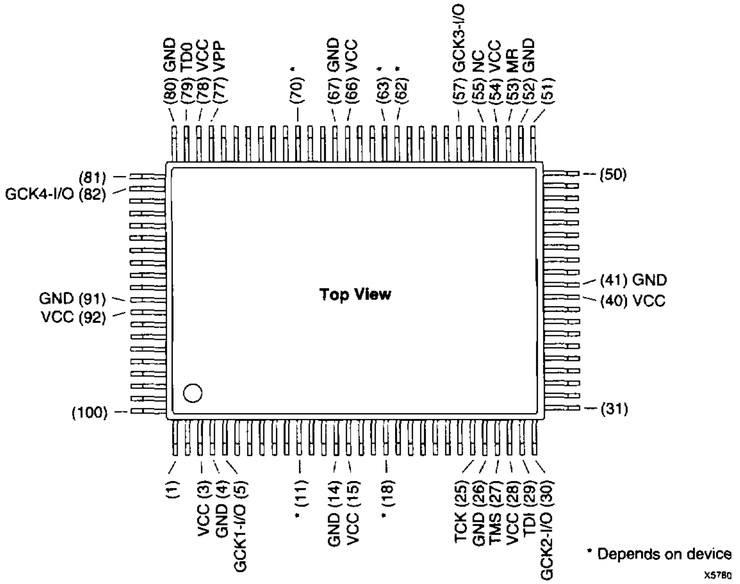


Figure 25: PQ100 PQFP with 100 Leads, .65 mm Lead Pitch

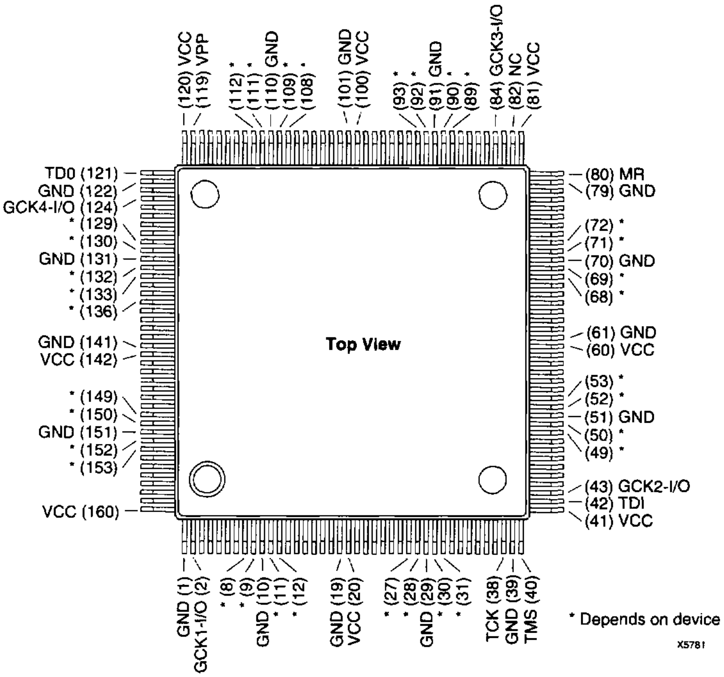
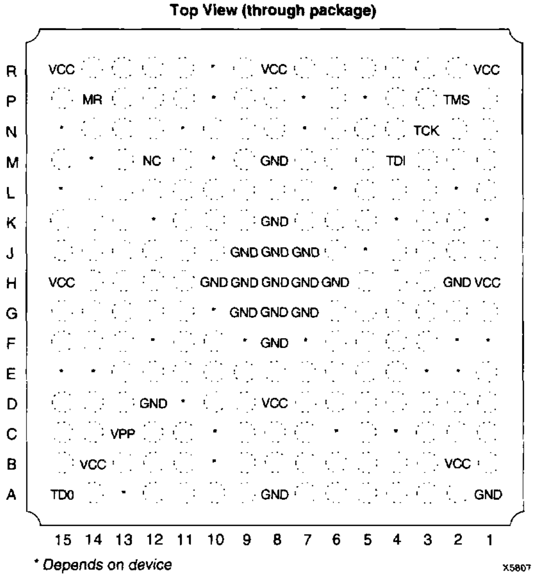


Figure 26: PQ160 PQFP with 160 Leads, .65mm Lead Pitch



Top View (through package)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
R	VCC					*		VCC							VCC
P		MR				*			*		*			TMS	
N	*				*				*				TCK		
M		*		NC		*		GND				TDI			
L	*									*				*	
K				*				GND				*			*
J							GND	GND	GND		*				
H	VCC					GND	GND	GND	GND	GND				GND	VCC
G						*	GND	GND	GND						
F				*			*	GND	*					*	*
E	*	*											*	*	
D				GND	*			VCC							
C			VPP			*				*		*			
B		VCC				*								VCC	
A	TD0		*					GND							GND

** Depends on device*

X5807

Figure 27: BG225 Plastic Ball Grid Array with 225 P_bS_n Balls, 15 x 15 Array, 1.50 mm Lead Pitch

Device Specifications

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 three-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 sec @ 1/16 inch)	+ 260	°C
T_J	Junction temperature - Ceramic	+ 150	°C
	Junction temperature - Plastic	+ 125	°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.

5 V 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
V_{IHT}	High-level input voltage for TTL threshold	2.0	V_{CC}	V
V_{IHC}	High-level input voltage for CMOS threshold	70%	100%	V_{CC}
V_{ILT}	Low-level input voltage for TTL threshold	0	0.8	V
V_{ILC}	Low-level input voltage for CMOS threshold	0	20%	V_{CC}
T_{IN}	Input signal transition time		250	ns

DC Characteristics Over 5 V Operating Conditions

Symbol	Description		Min	Max	Units
V_{OH}	High-level output voltage, I_{OH} = -4 mA, V_{CC} min		3.86		V
V_{OLR}	Low-level output voltage, I_{OL} = 24 mA, V_{CC} min resistive mode (Note 1)			0.50	V
V_{OLC}	Low-level output voltage, I_{OL} = 4 mA, V_{CC} min capacitive mode (Note 1)			0.40	V
I_{CCO}	Quiescent supply current, CMOS mode, (Note 2) – actual I_{CCO} depends on the design – TTL mode adds 8-30 mA – see Power Consumption section	XC8100 XC8101 XC8103 XC8106 XC8109		0.8 1.5 3.0 6.0 10.0	mA mA mA mA mA
I_I	Input leakage current		- 10	+ 10	µA
I_{RIN}	Pad pull-up current, V_{IN} = 0 V (sample tested)		0.02	0.20	mA
C_{IN}	Input capacitance (sample tested)			15	pF

Notes: 1. Total continuous output sink current may not exceed 100 mA per ground pin. The number of ground pins varies by package.
2. With no output current loads, no active inputs, all package pins at VCC or GND. Typical quiescent supply current at room temperature is less than 50% of the maximum.

DC Characteristics Over 3.3 V Operating Conditions

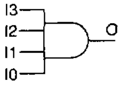
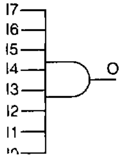
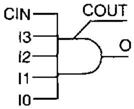
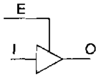
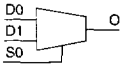
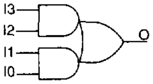
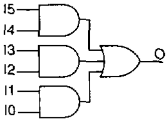
Symbol	Description		Min	Max	Units
V_{CC}	Supply voltage relative to GND Commercial	0°C to 70°C	3.0	3.6	V
V_{IH}	High-level input voltage (Note 1)		2.0	V_{CC} + 0.3	V
V_{IL}	Low-level input voltage (Note 1)		- 0.3	0.8	V
V_{OH}	High-level output voltage, I_{OH} = -4.0 mA, VCC min I_{OH} = -100 µA, VCC min		2.4 V_{CC} -0.2		V V
V_{OLR}	Low-level output voltage, I_{OL} = 12.0 mA, VCC min resistance mode			0.4	V
V_{OLC}	Low-level output voltage, I_{OL} = 4.0 mA, VCC min capacitive mode			0.4	V
I_{CCO}	Quiescent supply current, CMOS mode, (Note 2)	XC8100		250	µA
	– actual I_{CCO} depends on the design	XC8101		500	µA
	– see Power Consumption section	XC8103		800	µA
		XC8106		1500	µA
		XC8109		3000	µA

Notes: 1. Set CMOS/TTL input threshold to CMOS mode for use at 3.3 V.

2. With no output current loads, no active inputs, all package pins at V_{CC} or GND. Typical quiescent supply current at room temperature is less than 25% of the maximum.

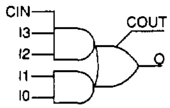
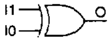
CLC Combinatorial Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% factory tested except for the programmed resistance of the fuses that will be used in the end application. 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. See the XC8100 software for complete timing information

		Speed Grade	-1				-2				
		Supply Voltage	5 V		3.3 V		5 V		3.3 V		
Description		Sym	Min	Max	Min	Max	Min	Max	Min	Max	Units
AND4 	Input I0 to output O	i0r		3.0		5.3					ns
	Input I1 to output O	i1r		2.9		5.2					ns
	Input I2 to output O	i2r		3.2		5.8					ns
	Input I3 to output O	i3r		2.9		5.2					ns
AND8 	Input I0 to output O	i0r		3.0		5.4					ns
	Input I4 to output O	i4r		5.4		9.9					ns
	Input I7 to output O	i7r		5.3		9.7					ns
	Note: Inputs I0 to I3 are faster than the cascaded inputs I4 to I7										
ANDCC 	Input I0 to output O	i0r		3.0		5.4					ns
	Input I3 to output O	i3r		3.3		6.1					ns
	Input I0 to output COUT	i0rc		2.3		4.2					ns
	Input I3 to output COUT	i3rc		2.6		4.9					ns
	Input CIN to output COUT	circ		2.4		4.6					ns
BUFE 	Input I to output O flow-through	ir		2.8		4.8					ns
	Three-state E to output O begin hi-Z	ehz		4.6		7.3					ns
	Three-state E to output O active	ezh		4.6		7.3					ns
M2_1 	Input D0 to output O	d0r		3.0		5.1					ns
	Input D1 to output O	d1r		2.8		5.0					ns
	Select input S to output O	s0r		3.1		5.6					ns
SOP4 	Input I0 to output O	i0r		3.1		5.5					ns
	Input I1 to output O	i1r		3.0		5.3					ns
	Input I2 to output O	i2r		3.1		5.5					ns
	Input I3 to output O	i3r		2.8		4.9					ns
SOP6 	Input I0 to output O	i0r		3.1		5.5					ns
	Input I2 to output O	i2r		4.8		8.4					ns
	Input I5 to output O	i5r		4.6		7.9					ns
	Note: Inputs I0 and I1 are faster and are approximately the same delay										

CLC Combinatorial Switching Characteristic Guidelines (continued)

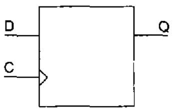
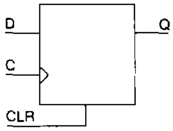
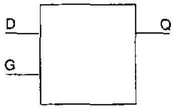
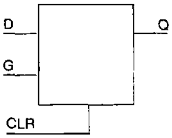
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% factory tested except for the programmed resistance of the fuses that will be used in the end application. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. See the XC8100 software for complete timing information.

		Speed Grade	**-1**				**-2**				
		Supply Voltage	**5 V**		**3.3 V**		**5 V**		**3.3 V**		
Description		**Sym**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
SOPCC											
	Input I0 to output O	i0r		3.1		5.5					ns
	Input I3 to output O	i3r		3.2		5.8					ns
	Input I0 to output COUT	i0r		2.3		4.3					ns
	Input I3 to output COUT	i3r		2.4		4.7					ns
	Input CIN to output COUT	circ		2.2		4.2					ns
XOR2											
	Input I0 to output O	i0r		3.3		5.6					ns
	Input I1 to output O	i1r		3.0		5.1					ns

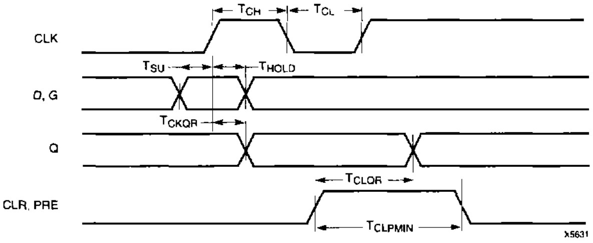
Notes:
1. Symbol names are those used in timing parameters in the XC8100 software, e.g., and4_i0r.
2. CLC combinatorial worst-case timings use rising edges. Falling edge signals are typically 0.6 ns faster.
3. Using inverted inputs ("bubbled") has minimal change on rising edge combinatorial timings. Inverting falling edge signals typically adds 0.4 - 0.5 ns.

CLC Sequential Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% factory tested except for the programmed resistance of the fuses that will be used in the end application. 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. See the XC8100 software for complete timing information.

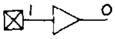
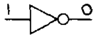
		Speed Grade	-1				-2				
		Supply Voltage	5 V		3.3 V		5 V		3.3 V		
Description		Sym	Min	Max	Min	Max	Min	Max	Min	Max	Units
FD 	Clock C to output Q delay	ckqr		3.5		6.1					ns
	Data set-up time before clock C	su	2.3		3.3						ns
	Data hold time after clock C	hold	0.0		0.0						ns
FDC 	Clock C to output Q delay	ckqr		3.5		6.1					ns
	Data set-up time before clock C	su	2.3		3.3						ns
	Data hold time after clock C	hold	0.0		0.0						ns
	Asynchronous CLR to output Q	clqf		3.0		5.9					ns
	CLR width	clpmin	1.7		3.3						ns
	Note: timings are similar for FDP (D FF with Reset)										
LD 	Data D to output Q delay	dqr		2.8		5.1					ns
	Latch G to output Q delay	gqr		3.0		5.4					ns
	Data set-up time to G	su	2.2		4.5						ns
	Data hold time from G	hold	0.0		0.0						ns
	G width	gpmin	2.5		5.3						ns
	Note: when G is high, data flows through. G latches on the falling edge										
LDC 	Data D to output Q delay	dqr		3.2		5.8					ns
	Latch G to output Q delay	gqr		3.4		6.4					ns
	Data set-up time to G	su	2.2		4.5						ns
	Data hold time from G	hold	0.0		0.0						ns
	Asynchronous CLR to output Q	clqr		5.1		9.1					ns
	CLR width	clpmin	2.2		3.9						ns
Clock	High time	T_{CH}	2.6		5.2						ns
	Low time	T_{CL}	2.6		5.2						ns
	Toggle Frequency	F_{TOG}		144							MHz

Note: Sequential worst-case timings delays use rising edges. Falling edge signals are typically 0.5 ns faster for single-CLC functions (e.g. LD) and 1.0 ns faster for double-CLC functions (e.g. FD).Toggle frequency based on worst-case data from XC8100 software.



Buffer Switching Characteristic Guidelines

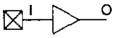
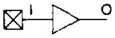
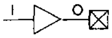
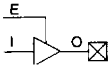
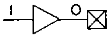
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% factory tested except for the programmed resistance of the fuses that will be used in the end application. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. See the XC8100 software for complete timing information.

		Speed Grade	-1				-2				
		Supply Voltage	5 V		3.3 V		5 V		3.3 V		
Description		Sym	Min	Max	Min	Max	Min	Max	Min	Max	Units
BUF1X 	Input I to output O delay	T_{IO} ir		2.8		4.8					ns
BUFGP 	Input I to output O delay Note: Timing is for buffer only, and does not include device-dependent wire delays	T_{IO} ir		4.6		7.6					ns
BUFROW 	Input I to output O delay Notes: 1. Same delay for BUFGS 2. Timing is for buffer only, and does not include device-dependent wire delays	T_{IO} ir		2.0		3.3					ns
INV1X 	Input I to output O delay	T_{IO} ir		3.0		5.1					ns

Note: 1. Symbols are Xilinx standard names and timing parameters used in XC8100 software, e.g., ibuf_ir.

I/O Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% factory tested except for the programmed resistance of the fuses that will be used in the end application. 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. See the XC8100 software for complete timing information.

	Speed Grade		-1				-2				
	Supply Voltage		5 V		3.3 V		5 V		3.3 V		
Description		Sym	Min	Max	Min	Max	Min	Max	Min	Max	Units
IBUF 	Input propagation delay	T_{PID}									
		ir		1.8		3.0					ns
	Note: same timing for IOBUF B-to-O										
DBUF 	Input propagation delay (with delay)	T_{PID}									
		ir		14.5		26.3					ns
	Note: same timing for DOBUF B-to-O										
OBUF 	Output delay to pin (capacitive mode)	T_{OP}									
		ir		3.4		6.7					ns
		if		5.4		7.2					ns
OBUFE 	Output delay to pin (capacitive mode)	ir		3.4		6.8					ns
		if		5.4		7.3					ns
	Three-state input E to pad O begin hi-Z	ehz		3.9		7.0					
	Three-state input E to pad O active	ezl		5.7		9.2					ns
											ns
	Note: same timing for IOBUF E-to-O										
RBUF 	Output delay to pin (resistive mode)	T_{OPR}									
		ir		3.5		6.1					ns
	Note: same timing for RBUFE	if		4.5		6.0					ns

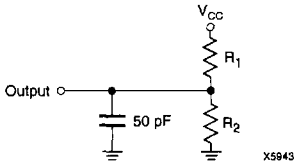
Notes: 1. Symbols are Xilinx standard names (e.g. TPID) and timing parameters used in XC8100 software, e.g., ibuf_ir.

2. Timing is measured at pin threshold, with 50 pF load.

3.

RBUF loading:

	5 V	3.3 V
R_1	177	217 Ω
R_2	370	294 Ω



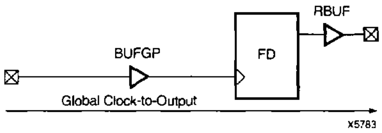
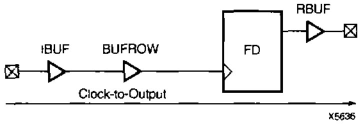
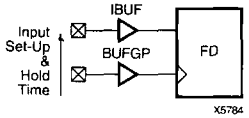
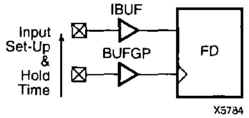
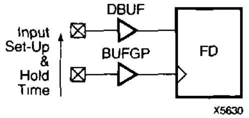
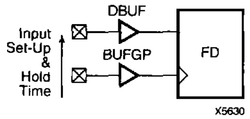
4. Output delays (except input E to pad O begin hi Z) change with capacitive load according to the following table. Delays above are calculated for 50 pF. As an example, obuf_ir at 20 pF is: 3.4 -(30 x 0.21) = 2.77 ns. XC8100 software can make these calculations automatically.

	5 V		3.3 V		
	Rise	Fall	Rise	Fall	Units
Resistive Mode	0.017	0.030	0.033	0.026	ns/pF
Capacitive Mode	0.021	0.046	0.054	0.049	ns/pF

5. Unused (bonded or unbonded) pads are automatically pulled-up internally with a ~50K Ω resistor.

Input and Output Parameters (Pin-to-Pin)

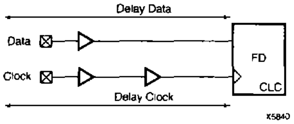
The following values reflect worst-case values over the recommended operating conditions. The exact timing depends on placement and routing so results may vary from design to design. See the XC8100 software for complete timing information.

		Speed Grade	-1				-2				
		Supply Voltage	5 V		3.3 V		5 V		3.3 V		
Description		Sym	Min	Max	Min	Max	Min	Max	Min	Max	Units
Global clock pin to output pin (resistive), 24 FF per row, all rows  	T_{ICKO}	XC8100		18.8		28.0					ns
		XC8101		19.0		29.0					ns
	(Max)	XC8103		20.0		30.1					ns
		XC8106		21.5		31.6					ns
		XC8109		21.9		32.0					ns
I/O pin to row output pin (resistive) 24 FF, 1 row  	T_{ICKO}	XC8100		15.5		24.0					ns
		XC8101		15.6		24.1					ns
	(Max)	XC8103		16.0		24.2					ns
		XC8106		16.7		25.1					ns
		XC8109		17.4		25.8					ns
Input set-up time, no delay, 1 FF  	T_{PSUF}	XC8100	0.1		0						ns
		XC8101	0.3		0						ns
	(Min)	XC8103	0.3		0						ns
		XC8106	0.1		0						ns
		XC8109	0		0						ns
Input hold time, no delay, 24 FF per row, all rows  	T_{PHF}	XC8100	7.8		11.5						ns
		XC8101	7.9		11.6						ns
	(Min)	XC8103	8.0		11.6						ns
		XC8106	8.3		12.1						ns
		XC8109	9.3		13.1						ns
Input set-up time, with DBUF delay, 1 FF  	T_{PSU}	XC8100	12.9		22.9						ns
		XC8101	13.1		23.2						ns
	(Min)	XC8103	13.0		23.1						ns
		XC8106	12.9		23.1						ns
		XC8109	13.0		23.0						ns
Input hold time, with DBUF delay, 24 FF per row, all rows  	T_{PH}	XC8100	0		0						ns
		XC8101	0		0						ns
	(Min)	XC8103	0		0						ns
		XC8106	0		0						ns
		XC8109	0		0						ns

Note: The external pin-to-pin setup and hold times for synchronous elements depend on the intrinsic setup and hold requirements of the appropriate CLC and the relative delays of the data and clock input nets. Specifically:

$T_{setup} = Delay_{data} + T_{setupCLC} - Delay_{clock}$

$T_{hold} = Delay_{clock} + T_{holdCLC} - Delay_{data}$



The data and clock delays consist of the delays associated with any blocks in the path (e.g. input buffers) and the net or wire delays. Once the design has been placed, routed and timed, the Series 8000 timing tool can be used to calculate the maximum possible delay for all paths. The actual delay will be less than these values particularly under non-worst case conditions. In most cases the delays on the clock and data paths will track. However not all nets depend identically on the same physical properties (resistance, capacitance, threshold voltage etc.) and hence not all delays will track perfectly.

Extensive simulation and characterization has shown that tracking between nets within one device will be better than 70%. To guarantee actual worst case setup and hold times, the worst possible tracking should be assumed.

Therefore the formulae become:

$$T_{setup} = Delay_{data} + T_{setupCLC} - 0.7^{*} (Delay_{clock})$$

$$T_{hold} = Delay_{clock} + T_{holdCLC} - 0.7^{*} (Delay_{data})$$

The user should use the 70% tracking factor when analyzing and interpreting timing information.

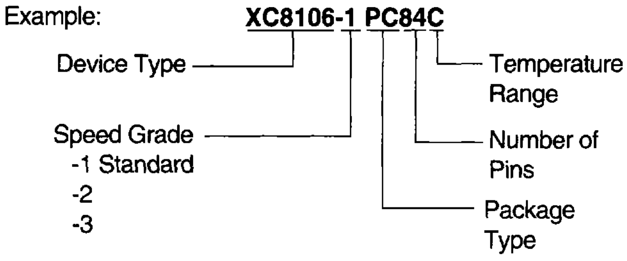
Achieving a 0 ns hold time at the pin level is a common requirement for FPGA designs. This entails having a data path which is slower than the clock path, which may be difficult if the clock has high fanout. To facilitate 0 ns hold times, an input buffer with additional delay (DBUF) is provided. This element has been designed to guarantee a 0 ns hold time even for a very heavily loaded clock signal.

When using DBUF to achieve 0 ns, the clock signal must be driven by a BUFGP element and should drive no more than 24 FFs per row. All rows may be driven provided each row contains a maximum of 24 FFs. If these conditions are met then the 0 ns hold time is guaranteed and this guideline supersedes the timing requirements indicated by the Series 8000 tool.

If the extra setup delay needed for DBUF cannot be tolerated, then IBUF should be used. In this case, the user may need to guarantee a positive hold time for data with respect to the clock for correct operation. The Series 8000 timing *requirements combined with the tracking factor should be* followed.

Description	Symbol	Device	Min	Max	Units
Reset Switching Characteristics Guidelines					
Delay from Master Reset pin high to device active	T_{MRQ}			3	ms
Low width on external MR pin	T_{MRW}		5		µs
Delay from internal GRST to FF reset -5 V	T_{GRIQ}			90	ns
Delay from internal GRST to FF reset -3.3 V	T_{GRIQ}			120	ns
GRST input I width (High or Low) -5 V	T_{GRW}		45		ns
GRST input I width (High or Low) -3.3 V	T_{GRW}		60		ns
Power-On Reset and Initalization Time					
V_{CC} above 2.5 V to device active. V_{CC} must rise monotonically.	T_{POR}			3	ms

Ordering Information



Note: Each Speed Grade can operate at 5 V or 3.3 V

Product Availability (5/96)

Pins	44	44	84	100	160	225
Type	**Plastic VQFP**	**Plastic PLCC**	**Plastic PLCC**	**Plastic PQFP**	**Plastic PQFP**	**Plastic BGFP**
Code	**VQ44**	**PC44**	**PC84**	**PQ100**	**PQ160**	**BG225**
XC8100-1	C	C				
XC8101-1	(C)	(C)	C	C		
XC8103-1	(C)	C	C	C	(C)	
XC8106-1			C	C	(C)	(C)
XC8109-1			C		C	C

Notes: Parentheses indicate future product plans