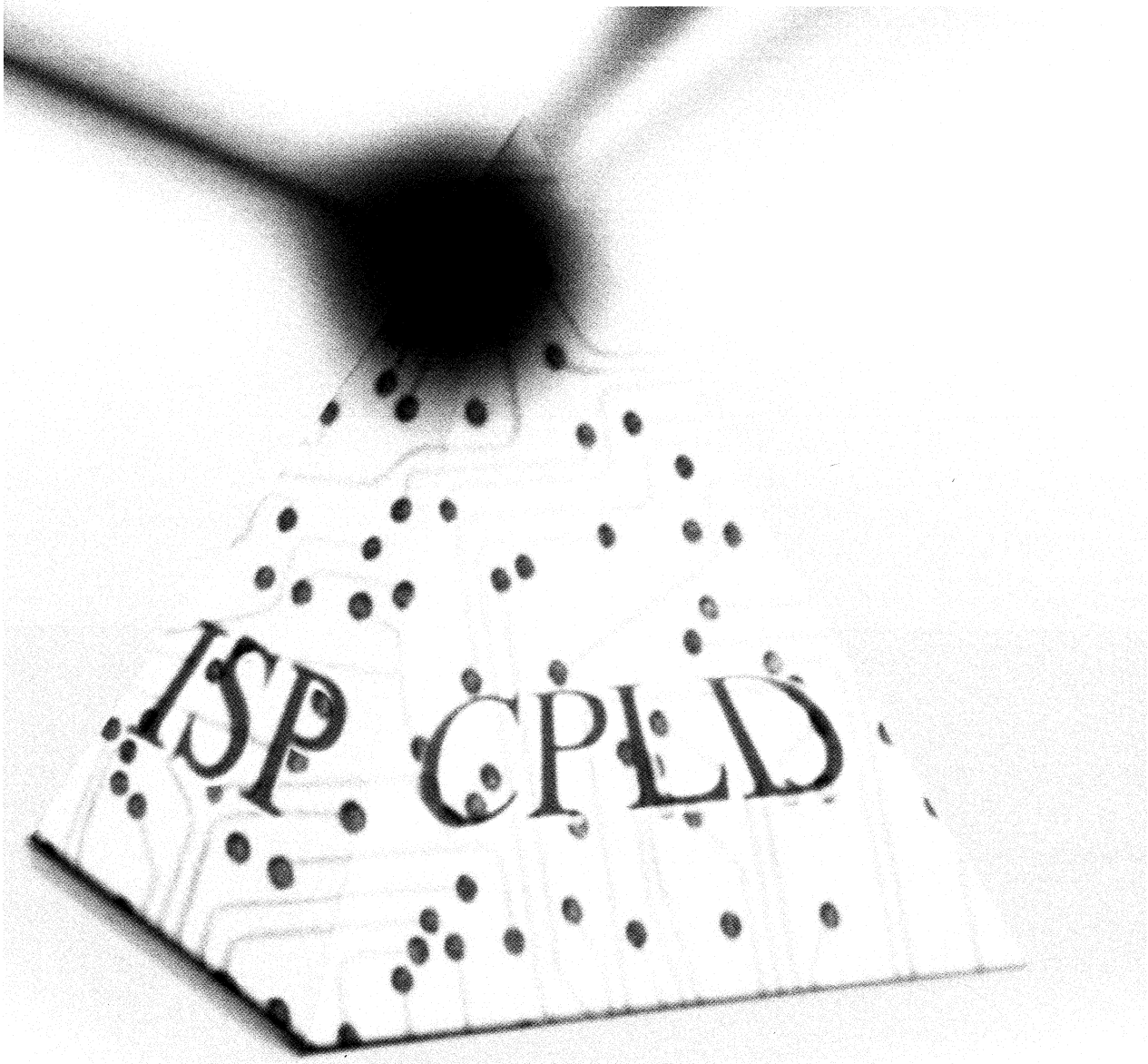
The ISP Application Guide
and CPLD Data Book



Success made simple.

XILINX®

The Xilinx logo shown above is a registered trademark of Xilinx, Inc.

The shadow X shown above is a trademark of Xilinx, Inc.

XILINX, XACT, XC2064, XC3090, XC4005, XC-DS501, FPGA Architect, FPGA Foundry, NeoCAD, NeoCAD EPIC, NeoCAD PRISM, NeoROUTE, Plus Logic, Plustran, P+, Timing Wizard, and TRACE are registered trademarks of Xilinx, Inc.

All XC-prefix product designations, XACT*step*, XACT*step* Advanced, XACT*step* Foundry, XACT-Floorplanner, XACT-Performance, XAPP, XAM, X-BLOX, X-BLOX plus, XChecker, XDM, XDS, XEPLD, XPP, XSI, BITA, Configurable Logic Cell, CLC, Dual Block, FastCLK, FastCONNECT, FastFLASH, FastMap, HardWire, LCA, Logic Cell, LogiCore, LogicProfessor, MicroVia, PLUSASM, PowerGuide, PowerMaze, Select-RAM, SMARTswitch, TrueMap, UIM, VectorMaze, VersaBlock, VersaRing, XABEL, Xilinx Foundation Series, and ZERO+ are trademarks of Xilinx, Inc. The Programmable Logic Company and The Programmable Gate Array Company are service marks of Xilinx, Inc.

IBM is a registered trademark and PC/AT, PC/XT, PS/2 and Micro Channel are trademarks of International Business Machines Corporation. DASH, Data I/O and FutureNet are registered trademarks and ABEL, ABEL-HDL and ABEL-PLA are trademarks of Data I/O Corporation. SimuCad and Silos are registered trademarks and P-Silos and P/C-Silos are trademarks of SimuCad Corporation. Microsoft is a registered trademark and MS-DOS is a trademark of Microsoft Corporation. Centronics is a registered trademark of Centronics Data Computer Corporation. PALASM is a registered trademark of Advanced Micro Devices, Inc. UNIX is a trademark of AT&T Technologies, Inc. CUPL, PROLINK, and MAKEPRG are trademarks of Logical Devices, Inc. Apollo and AEGIS are registered trademarks of Hewlett-Packard Corporation. Mentor and IDEA are registered trademarks and NETED, Design Architect, System Architect, QuickSim, QuickSim II, and EXPAND are trademarks of Mentor Graphics, Inc. Sun is a registered trademark of Sun Microsystems, Inc. SCHEMA II+ and SCHEMA III are trademarks of Omation Corporation. OrCAD is a registered trademark of OrCAD Systems Corporation. Viewlogic, Viewsim, and Viewdraw are registered trademarks of Viewlogic Systems, Inc. CASE Technology is a trademark of CASE Technology, a division of the Teradyne Electronic Design Automation Group. DECstation is a trademark of Digital Equipment Corporation. Synopsys is a registered trademark of Synopsys, Inc. Verilog is a registered trademark of Cadence Design Systems, Inc.

Xilinx, Inc. does not assume any liability arising out of the application or use of any product described or shown herein; nor does it convey any license under its patents, copyrights, or maskwork rights or any rights of others. Xilinx, Inc. reserves the right to make changes, at any time, in order to improve reliability, function or design and to supply the best product possible. Xilinx, Inc. will not assume responsibility for the use of any circuitry described herein other than circuitry entirely embodied in its products. Xilinx, Inc. devices and products are protected under one or more of the following U.S. Patents: 4,642,487; 4,695,740; 4,706,216; 4,713,557; 4,746,822; 4,750,155; 4,758,985; 4,820,937; 4,821,233; 4,835,418; 4,853,626; 4,855,619; 4,855,669; 4,902,910; 4,940,909; 4,967,107; 5,012,135; 5,023,606; 5,028,821; 5,047,710; 5,068,603; 5,140,193; 5,148,390; 5,155,432; 5,166,858; 5,224,056; 5,243,238; 5,245,277; 5,267,187; 5,291,079; 5,295,090; 5,302,866; 5,319,252; 5,319,254; 5,321,704; 5,329,174; 5,329,181; 5,331,220; 5,331,226; 5,332,929; 5,337,255; 5,343,406; 5,349,248; 5,349,249; 5,349,250; 5,349,691; 5,357,153; 5,360,747; 5,361,229; 5,362,999; 5,365,125; 5,367,207; 5,386,154; 5,394,104; 5,399,924; 5,399,925; 5,410,189; 5,410,194; 5,414,377; 5,422,833; 5,426,378; 5,426,379; 5,430,687; 5,432,719; 5,448,181; 5,448,493; 5,450,021; 5,450,022; 5,453,706; 5,466,117; 5,469,003; 5,475,253; 5,477,414; 5,481,206; 5,483,478; 5,486,707; 5,486,776; 5,488,316; 5,489,858; 5,489,866; 5,491,353; 5,495,196; 5,498,979; 5,498,989; 5,499,192; 5,500,608; 5,500,609; 5,502,000; 5,502,440; RE 34,363, RE 34,444, and RE 34,808. Other U.S. and foreign patents pending. Xilinx, Inc. does not represent that devices shown or products described herein are free from patent infringement or from any other third party right. Xilinx, Inc. assumes no obligation to correct any errors contained herein or to advise any user of this text of any correction if such be made. Xilinx, Inc. will not assume any liability for the accuracy or correctness of any engineering or software support or assistance provided to a user.

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

The ISP Application Guide and CPLD Data Book

Technical Support Telephone Hotline:	800-255-7778 (North America) 408-879-5199 (USA, Xilinx headquarters) (44) 1932 820821 (United Kingdom) (33) 1 3463 0100 (France) (49) 89 991 54930 (Germany) (81) 3-3297-9163 (Japan)
Technical Support Direct FAX:	408-879-4442 (USA, Xilinx headquarters) (44) 1932 828522 (United Kingdom) (33) 1 3463 0959 (France) (49) 89 904 4748 (Germany) (81) 3-3297-0067 (Japan)
Technical Support Hotline E-mail:	hotline@xilinx.com (USA, Xilinx headquarters) ukhelp@xilinx.com (United Kingdom) frhelp@xilinx.com (France) dlhelp@xilinx.com (Germany) jhotline@xilinx.com (Japan)
Xilinx BBS:	408-559-9327 (USA, Xilinx headquarters) (44) 1932 333540 (United Kingdom) 8 data bits, no parity, 1 stop
XDOCS E-mail Document Server:	xdocs@xilinx.com send E-mail with help in the header
XFACTS Automated FAX Server:	408-879-4400
Xilinx Home Page (WWW):	http://www.xilinx.com/

2100 Logic Drive
San Jose, California 95124
United States of America
Telephone: (408) 559-7778
Fax: (408) 559-7114

Welcome to our new ISP Application Guide and CPLD Databook!

This book will provide you with all the technical specifications and application information you'll need to fully utilize the advanced features of our exciting new XC9500 In-System Programmable™ CPLDs, as well as complete technical specifications on our industry standard, low cost, XC7300 family. Use these advanced devices to gain the high performance, time-to-market, and cost benefits that are necessary for your company's end products.

Our goal at Xilinx is to continually innovate new programmable logic devices and technologies, giving you the means to create superior, value-added products in your market. So, no matter what your design needs are, Xilinx has the right density with the right performance at the right price for all your programmable logic requirements.

We also provide the best customer support in the industry. You'll find our applications staff always ready and capable of helping you create the most efficient, highest performance designs possible. In addition, our web site (www.xilinx.com) is filled with the latest product and applications information; a quick, thorough, and up to date resource that can save you valuable time.

We are pleased to help you with all of your programmable logic requirements. Thank you for choosing Xilinx.

XILINX®

Section Titles

1 ISP and JTAG Support

2 Application Notes

3 XC9500 Data Sheets

4 XC7300 Data Sheets

5 Device Packaging

6 Quality Assurance

7 Technical Support

8 Sales Offices, Representatives, Distributors

Table of Contents

ISP and JTAG Support

Using the XC9500 JTAG Boundary-Scan Interface

Downloading a Design File 1-1
EZTag Download Software 1-2
Interfacing to Third-Party Boundary-Scan Test Tools 1-2
JTAG Details 1-4
JTAG Instructions Supported in XC9500 Parts 1-6
Device Operations 1-7
BSDL Description Summary 1-7
JEDEC Description Summary 1-7

Using In-System Programmability in Boundary-Scan Systems

XC9500 TAP Characteristics 1-9
Creating Boundary-Scan Chains 1-10
In-System Programming 1-13
System-Level Design Issues 1-13

XC9500 In-System Programming Using an Embedded Microcontroller

Programming XC9500 CPLDs 1-16
Hardware Design 1-17
Firmware Design 1-18
SVF Overview 1-27
SVF Commands 1-27
XSVF Commands 1-29
svf2xsvf File Conversion Utility 1-30
xsvf2ascii File Conversion Utility 1-30

Using Automatic Test Equipment to Program XC9500 Devices In-System

SVF Overview 1-47
SVF Specification 1-48
Using EZTag to generate an SVF file 1-49
SVF Interpretation 1-49
Predicted ATE Programming Time 1-52

In-System Programming Times

Device Programming Times 1-53

Application Notes

XC9500 Pin-Locking Capability and Benchmarks

Pin-Locking Issues 2-1
The Keys to Reliable Pin-Locking 2-2
Pin-Locking Benchmarks 2-2

Using the XC9500 Timing Model

Device Timing Overview 2-9
Timing Model 2-10

XC9500 Design Optimization

Optimizing Density 2-13
Optimizing Timing 2-15
Optimizing Schematic Designs 2-15
Optimizing ABEL Designs 2-15
Optimizing VHDL Designs 2-16
TSPEC and .CST File Format 2-17

Designing with XC9500 CPLDs

XC9500 Architecture 2-19
Automatic Software 2-22
Practical Considerations for XC9500 Designs 2-24

Pin Preassigning with XC9500 CPLDs

XC9500 Routing Resources 2-28
Datapath Estimation Guidelines 2-28
Control Path Estimation Guidelines 2-28
Pin Preassigning Guidelines 2-29
Manual Estimation Example 2-29

Design Migration with XC9500 CPLDs

The XC9500 Architecture 2-34
XC9500 Packaging 2-34
Design Migration Strategy 2-34
Design Migration Example 2-35

Using ABEL with Xilinx CPLDs

ABEL Language Structure 2-41
Using Truth Tables 2-45
Using State Diagrams 2-45
Using Property Statements 2-47

Embedded Instrumentation Using XC9500 CPLDs

Creating a Signature Analyzer 2-51
Creating a Logic Analyzer 2-53
Creating Checkpoint Probes 2-54

Metastability Considerations

Metastability Measurement 2-56
Design Considerations 2-57

FastFLASH: A New Electrically Erasable CPLD Technology

EEPROM Technology 2-60
FastFLASH Technology 2-61

XC9500 Data Sheets

XC9500 In-System Programmable CPLD Family

Family Overview 3-1
Architecture Description 3-1
Function Block 3-3
Macrocell 3-4
Product Term Allocator 3-6
FastCONNECT Switch Matrix 3-9
I/O Block 3-10
Pin-Locking Capability 3-11
In-System Programming 3-12
Endurance 3-12
IEEE 1149.1 Boundary-Scan (JTAG) 3-12
Design Security 3-12
Low Power Mode 3-13
Timing Model 3-13
Power-Up Characteristics 3-15
Development System Support 3-15
FastFLASH Technology 3-15

XC9536 3-17

XC9572 3-25

XC95108 3-33

XC95144 3-41

XC95180 3-47

XC95216 3-53

XC95288 3-63

XC95432 3-75

XC95576 3-77

XC7300 Data Sheets

XC7300 CMOS CPLD Family

Description 4-1
3.3 V or 5 V Interface Configuration 4-7
Power-On Characteristics/Master Reset 4-7
Power Management 4-7
Erasure Characteristics 4-7
Design Recommendations 4-7
Design Security 4-8
High-Volume Production Programming 4-8
XACTstep Development System 4-8
Timing Model 4-8

XC7318 4-11

XC7336/XC7336Q 4-19

XC7354 4-29

XC7372 4-37

XC73108 4-45

XC73144 4-55

Device Packaging

Packages and Thermal Characteristics

Number of Available I/O Pins 5-1
Package Options 5-2
Physical Dimensions 5-3

Quality Assurance

Xilinx Quality Assurance

Quality Assurance Program 6-1

Technical Support

Technical Support

Technical Support Hotlines 7-1
X-TALX: The Xilinx Network of Electronic Services 7-2
Technical Literature 7-3
XCELL Newsletter 7-3

HW-130 Programmer

HW-130 Programmer 7-5

Sales Offices

Sales Offices, Sales Representatives, and Distributors

Sales Offices, Representatives, and Distributors 8-1

XILINX®

ISP and JTAG Support

1 ISP and JTAG Support

2 Application Notes

3 XC9500 Data Sheets

4 XC7300 Data Sheets

5 Device Packaging

6 Quality Assurance

7 Technical Support

8 Sales Offices, Representatives, Distributors

Visit us on the WEB at:

www.xilinx.com

for the latest product information.

Using the XC9500 JTAG Boundary-Scan Interface

January, 1997 (Version 1.0) Application Note

Summary

This application note explains the XC9500 boundary-scan interface and demonstrates the software available for programming and testing XC9500 CPLDs. An appendix summarizes JTAG operations and overviews the additional operations supported by XC9500 CPLDs for in-system programming.

Xilinx Family

XC9500

1

Introduction

IEEE Boundary-Scan Standard 1149.1, also known as JTAG, is a testing standard that uses software to reduce costs. The primary benefit of the standard is its ability to transform difficult printed circuit board testing problems into well-structured, efficient solutions that are easily performed in software. The standard defines a hardware architecture and the mechanisms for its use.

The JTAG standard itself defines instructions that can be used to perform functional and interconnect tests as well as built-in self test procedures. Vendor-specific extensions to the standard allow execution of maintenance and diagnostic applications as well as permit programming algorithms for reconfigurable parts.

Connecting Devices in a Boundary-Scan Chain

All devices in the chain share the TCK and TMS signals. The system TDI signal is connected to the TDI input of the first device in the boundary-scan chain. The TDO signal from that first device is connected to the TDI input of the second device in the chain and so on. The last device in the chain has its TDO output connected to the system TDO pin. This configuration is illustrated in Figure 2.

Downloading a Design File

The JTAG Download Cable, shown in Figure 1, connects to the parallel printer port of any PC. The cable contains drivers to buffer the signals as they are driven into the system, and the power for the drivers is derived from the target system. The cable's V_{CC} and GND wires are connected to the corresponding signals on the target system, and the remaining four wires are connected to the corresponding TAP inputs on the target system. The cable pins are clearly labeled. TRST is not supported by the JTAG Download cable and if any parts in the system have a TRST, this pin should be attached to V_{CC} through a pullup resistor.

Figure 3 shows how the cable is connected to the printed circuit board for programming. Connect all six flying leads to the target board and observe the power sequencing recommendations.

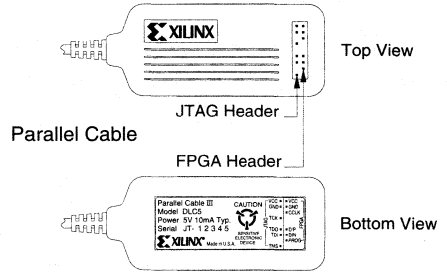


Figure 1: XC9500 JTAG Download Cable

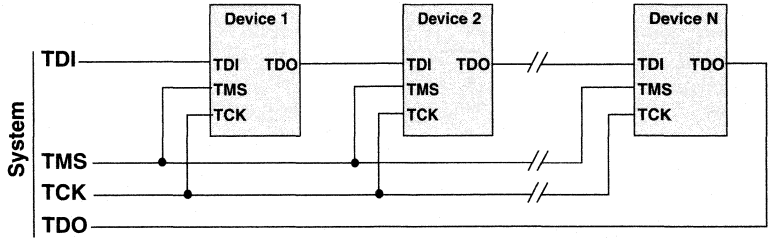


Figure 2: Single Port Serial Boundary-Scan Chain

Power sequencing

Cable protection ensures the parallel port cannot be damaged through normal cable operation. For increased safety, ensure that the PC is always powered up before the target system.

When powering down, turn off the target system first, and then turn off the host PC.

EZTag Download Software

Upon initiation of the EZTag Download Software, the parallel port is queried to verify the connection of the JTAG Download cable. The target system power must be on and the cable attached for proper verification. If an error message is returned, stating that the cable could not be found or indicating a cable other than the JTAG Download cable was identified, check the cable power connections. Figure 4 shows the EZTag software user interface.

Using EZTag

The following steps outline the downloading procedure:

1. Invoke EZTag.
2. Select the files for each device in the chain ordered from system TDI to TDO (use JEDEC files for XC9500 devices or BSDL files for other JTAG-compatible devices).
3. Select the operations desired for each XC9500 part.
4. Select the "execute" button and downloading will begin. Detailed information regarding the downloading progress and any failure conditions will be displayed in the system log file.

Using the XChecker Cable

The XChecker cable can also be used to program XC9500 devices. In this case, attach the TDI, TCK, TMS, V_{CC}, and GND pins to the target board with the flying leads, as shown in Figure 5. The TDO signal function will be performed by the XChecker signal labeled "RD". The EZTag software will automatically query the computer I/O ports and detect the existence of the XChecker cable.

See Appendix 1 for more details on the specific JTAG features supported.

Interfacing to Third-Party Boundary-Scan Test Tools

BSDL files are required for interfacing to third-party boundary-scan board test equipment (ATE), automatic test pattern generation software (ATPG) and JTAG-based development and de-bugging systems.

The BSDL files for all package variations of the available XC9500 devices can be found in the EZTag software "data" directory. The BSDL file names are shown in Table 1.

Table 1: BSDL Files

Part Type	Package	BSDL File Name
XC9536	PC44	xc3644p.bsd
XC9536	VQ44	xc3664v.bsd
XC95108	PQ100	xc108100.bsd
XC95108	PQ160	xc108100.bsd
XC95108	PC84	xc10884.bsd
XC95108	TQ100	xc108tq.bsd
XC95216	PQ160	xc216160.bsd
XC95216	PQ208	xc216208.bsd
XC9572	PQ100	xc72100p.bsd
XC9572	TQ100	xc72100t.bsd
XC9572	PC84	xc7284.bsd
XC95288	HQ208	xc288208.bsd

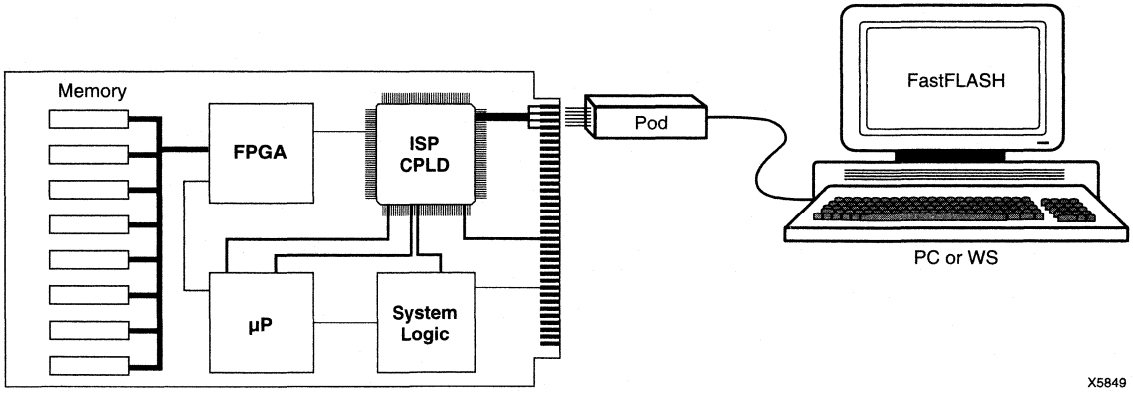


Figure 3: Target PCB Connected for Program/Test

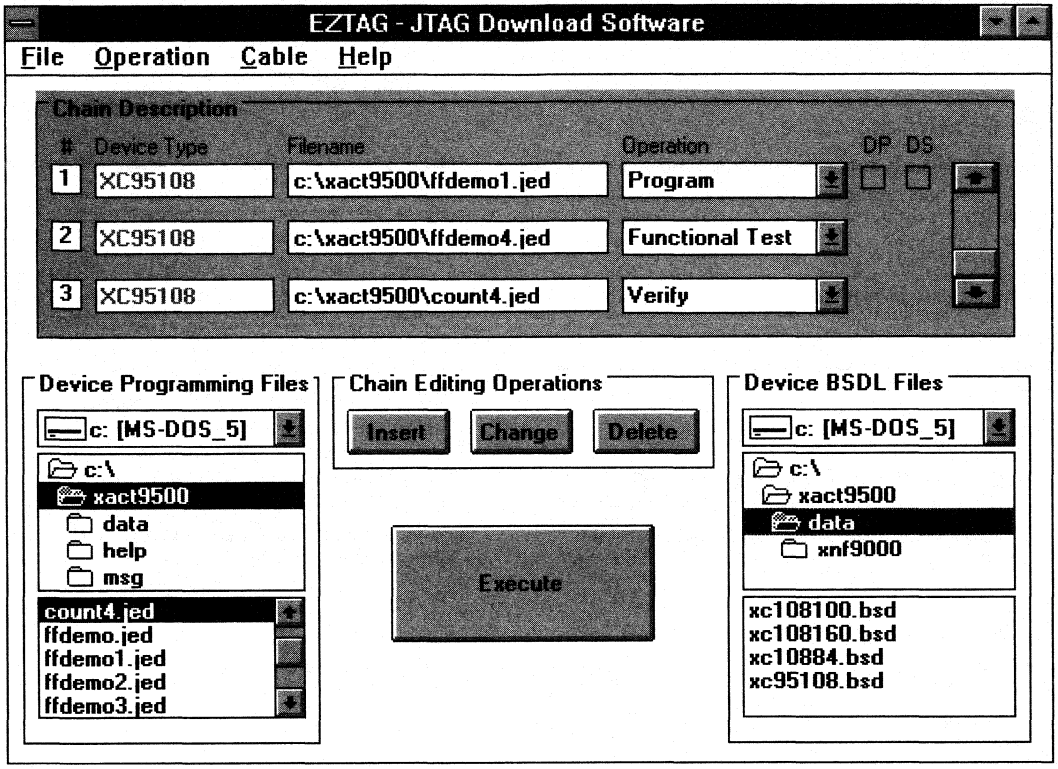


Figure 4: The EZTag Download Software User Interface

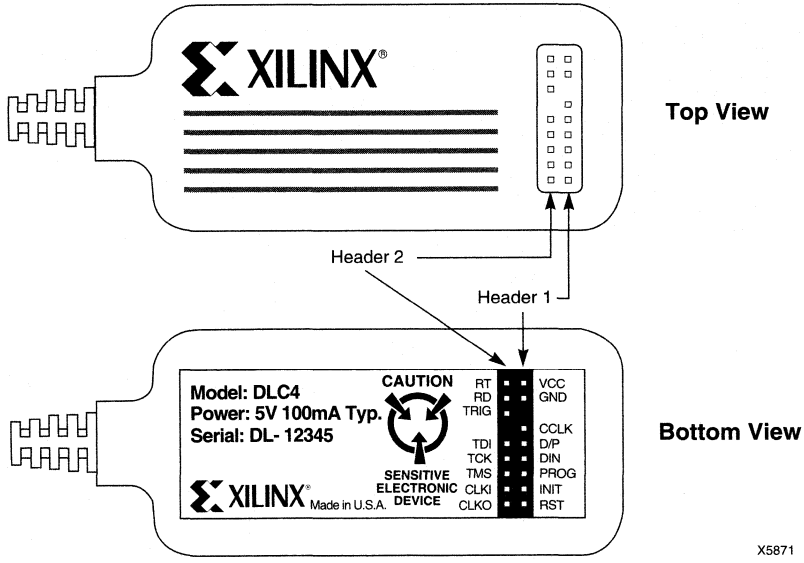


Figure 5: XChecker Cable

Appendix 1 - JTAG Details

The top level schematic of the test logic defined by IEEE Std. 1149.1 includes several key blocks as shown in Figure 6:

The TAP Controller

The TAP controller responds to control sequences supplied through the test access port (TAP) and generates the clocks and control signals required by the other circuit blocks.

The Instruction Register

The instruction register is a shift register-based circuit and is serially loaded with instructions that select an operation to be performed.

The Data Registers

The data registers are a bank of shift registers. The stimuli required by an operation are serially loaded into the data registers, selected by the current instruction. Following execution of the operation, results can be shifted out for examination.

The JTAG Test Access Port

The JTAG Test Access Port (TAP) has four pins that drive the circuit blocks and control specific operations. The TAP loads and unloads instructions and data. The four TAP pins are: TMS, TCK, TDI and TDO. The function of each TAP pin is:

- **TMS** - Test Mode Select is the mode input signal to the TAP Controller. The TAP controller is a 16-state finite state machine (FSM) that controls the JTAG engine. At the rising edge of TCK, TMS determines the TAP controller state sequence. TMS has an internal pull-up resistor to provide a logic 1 to the system if TMS is not driven.
- **TCK** - JTAG Test Clock sequences the TAP controller as well as all JTAG registers.
- **TDI** -Test Data Input is the serial data input to all JTAG instruction and data registers. The TAP controller state and instruction register contents determine which register is fed by TDI for any operation. TDI has an internal pull-up resistor to provide a logic 1 to the system if TDI is not driven. TDI is loaded into the JTAG registers on TCK's rising edge.
- **TDO** - Test Data Out is the serial data output for all JTAG instruction and data registers. The TAP controller state and instruction register contents determine which register feeds TDO for a specific operation. Only one register (instruction or data) is connected between TDI and TDO for any JTAG operation. TDO changes state on TCK's falling edge and is only active during the shifting of data through the device. TDO is in a 3-state condition at all other times.

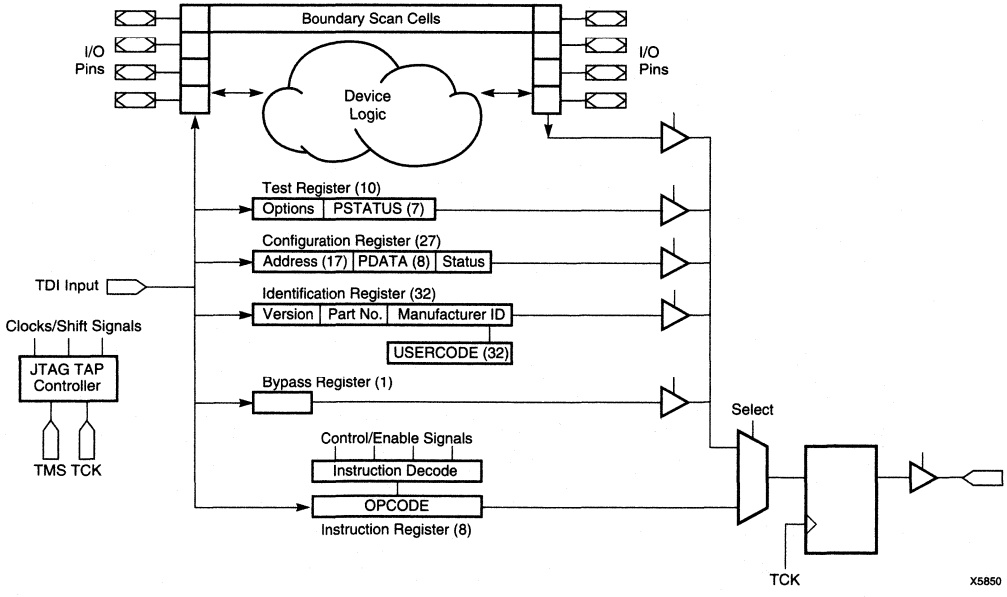


Figure 6: JTAG Architecture

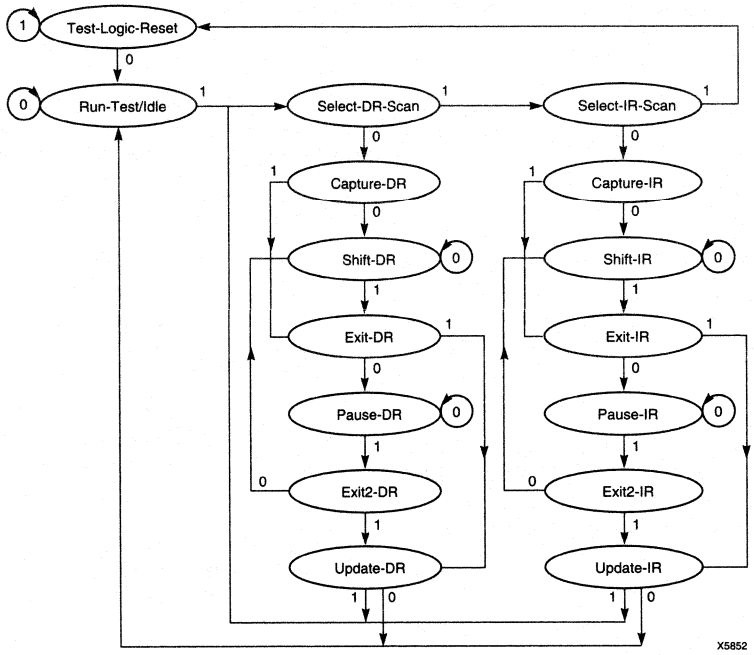


Figure 7: TAP Controller State Diagram

JTAG TAP Controller

The TAP Controller is a 16-state FSM, that controls the loading of data into the various JTAG registers. A state diagram of the TAP controller is shown in Figure 7.

The state of TMS at the rising edge of TCK determines the sequence of state transitions. There are basically two state transition paths for sampling the signal at TDI: one for shifting information to the instruction register and one for shifting data into the data register.

JTAG TAP Controller States

Test-Logic-Reset

This state is entered on device power-up when at least five TCK clocks occur with TMS held high. Entry into this state resets all JTAG logic so that it does not interfere with the normal component logic, and loads the IDCODE instruction into the instruction register.

Run-Test-Idle

In this state certain operations can occur depending on the current instruction. For the XC9500 family, "Run-Test-Idle" causes generation of the program, verify, erase, and POR (Power-On-Reset) pulses when the associated ISP instruction is active.

Select-DR-Scan

This is a transitional state entered prior to performing a scan operation on a data register or in passing to the Select-IR-Scan state.

Select-IR-Scan

This is a transitional state entered prior to performing a scan operation on the instruction register or in returning to the Test-Logic-Reset state.

Capture-DR

This state allows data to be loaded from parallel inputs into the data register selected by the current instruction at the rising edge of TCK. If the selected data register has no parallel inputs, the register retains its state.

Shift-DR

In this state data is shifted by one stage in the currently selected register from TDI towards TDO by on each rising edge of TCK.

Exit1-DR

This is a transitional state allowing the option of passing to the Pause- DR state or transitioning directly to the Update-DR state.

Pause-DR

This is a wait state that allows shifting of data to be temporarily halted.

Exit2-DR

This is a transitional state allowing the option of passing to the Update-DR state or returning to the Shift-DR state to continue accepting data.

Update-DR

In this state the data contained in the currently selected data register is loaded into a latched parallel output (for registers that have such a latch) on the falling edge of TCK after entering this state. The parallel latch prevents changes at the parallel register output from occurring during the shifting process.

Capture-IR

In this state data is loaded from parallel inputs into the instruction register on the rising edge of TCK. The least two significant bits of the parallel inputs must have the value 01, and the remaining 6 bits are either hard-coded or used for monitoring the security and data protect bits.

Shift-IR

In this state instruction register values are shifted one stage towards TDO on each rising TCK edge.

Exit1-IR

Exit1-IR is a transitional state allowing the option of transitioning to the Pause-IR state or the Update-IR state.

Pause-IR

Pause-IR allows shifting of the instruction to be temporarily halted.

Exit2-IR

Exit2-IR is a transitional state allowing the option of passing to the Update-IR state or returning to the Shift-IR state to continue shifting in data.

Update-IR

In this state instruction register values are parallel latched on the falling edge of TCK. The parallel latch prevents changes at the parallel output of the instruction register from occurring during the shifting process.

JTAG Instructions Supported in XC9500 Parts

Mandatory Boundary Scan Instructions

BYPASS

The BYPASS instruction configures the device to bypass the scan registers and pass immediately to TDO.

SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction allows a snapshot of the normal operation of a component to be taken and examined. It also allows data values to be loaded onto the latched parallel outputs of the boundary scan shift register prior to the selection of other boundary-scan test instructions.

EXTEST

The EXTEST instruction allows testing of off-chip circuitry and board level interconnections.

XC9500 Additional Boundary Scan Instructions

INTEST

The INTEST instruction allows testing of the on-chip system logic while the component is already on the board.

HIGHZ

HIGHZ permits automatic placement of all outputs on the XC9500 part to high impedance (3-state) mode. This condition can be beneficial for board testing strategies.

IDCODE

The IDCODE instruction allows blind interrogation of the components assembled onto a printed circuit board to determine what components exist in a system.

USERCODE

The USERCODE instruction allows a user-programmable identification code to be shifted out for examination. This allows the programmed function of the component to be determined.

XC9500 Reconfiguration Instructions

ISPEN

ISPEN activates the XC9500 device for In-System Programming.

FPGM

FPGM programs bits at specified addresses.

FERASE

FERASE erases a block of programming cells.

FVFY

FVFY verifies the programming at specified addresses.

ISPEX

ISPEX transfers the XC9500 memory cell contents to internal low power configuration latches.

Device Operations

The programming information is extracted from the JEDEC file generated by the fitter software. The JEDEC file name is defaulted to *design_name*.jed.

Device operation options available to users are:

Program & Verify

Download contents of the JEDEC file to the device programming registers. Configure the device and read back the contents of device programming registers and compare them to the JEDEC file. Report any differences to the user.

Verify

Read back contents of the device programming registers and compare them with the JEDEC file.

Erase

Clear the device configuration information.

Functional Test

Apply user-specified functional vectors from the JEDEC file to the device, comparing results obtained with expected values. Report any differences.

Read Device ID

Read and display the contents of the JTAG IDCODE register.

Read User Signature

The signature value is set by the user at programming time. It is valid only after programming. This function reads the contents of the JTAG USERCODE register and displays the result.

Bypass

Ignore this device when addressing devices in the JTAG boundary scan chain.

Readback

Extracts contents of device programming registers and creates a new JEDEC file with the results.

Checksum

Extract the contents of device programming registers and calculate a checksum for comparison with the expected value

BSDL Description Summary

The Boundary-Scan Description Language (BSDL) describes the boundary scan features of a component. The system looks for BSDL files along the XACT path and in the current working directory. A BSDL file must be specified for each non-XC9500 device in the JTAG chain.

The name of the BSDL file is assumed to be *device_name*.bsd.

JEDEC Description Summary

The JEDEC file is an ASCII file containing the configuration information and optionally the vectors that can be used to verify the functional behavior of the configured part. A JEDEC file must be specified for each XC9500 device in the JTAG chain; one JEDEC file is generated for each XC9500 device in the system by the fitter software.

The name of the JEDEC file is assumed to be *design_name*.jed.

References

1. IEEE Std. 1149.1a 1993 Standard Test Access Port & Boundary-Scan Architecture. 1993

2. The Boundary-Scan Handbook, Ken Parker, Klewer Academic Publishers, 1992

3. JEDEC Standard, Standard Data Transfer Format Between Data Preparation System and Programmable Logic Device Programmer JESD3-C, June 1994.

4. IEEE Std. 1149.1b Supplement (B) to Standard Test Access Port & Boundary-Scan Architecture, IEEE Std. 1149.1 - 1990, 1994.

Using In-System Programmability in Boundary-Scan Systems

January, 1997 (Version 1.0) Application Note

Summary

This application Note discusses basic design considerations for in-system programming of multiple XC9500 devices in a boundary-scan chain, and shows how to design systems that contain multiple XC9500 devices as well as other IEEE 1149.1-compatible devices.

Xilinx Family

XC9500

Introduction

The XC9500 family performs both in-system programming and IEEE 1149.1 boundary-scan (JTAG) testing via a single 4-wire Test Access Port (TAP). This simplifies system designs and allows standard Automatic Test Equipment to perform both functions. Xilinx also provides the EZTag™ software that automatically programs and tests XC9500 devices from the standard test vector and device programming files generated by most CPLD development tools.

XC9500 TAP Characteristics

The AC and DC characteristics of the XC9500 TAP are described as follows.

TAP Timing

Figure 1 shows the timing relationships of the TAP signals. These TAP timing characteristics are identical for both boundary-scan and ISP operations. The timing for the INPUT-I/O-CLK and I/O signals is relevant to boundary-scan operations (such as EXTEST) that activate or strobe the system pins.

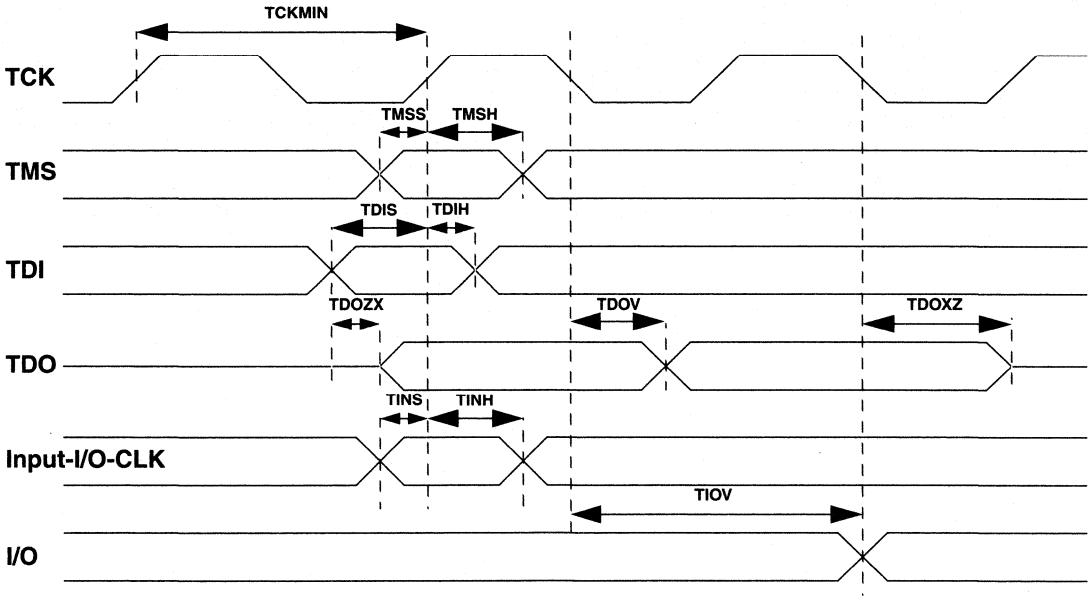


Figure 1: Test Access Port Timing

TAP AC Parameters

Table 1 shows the timing parameters for the TAP waveforms shown in Figure 1.

Table 1: Test Access Port Timing Parameters (ns)

Symbol	Parameter	Min	Max
TCKMIN	TCK Minimum Clock Period	100	
TMSS	TMS Setup Time	10	
TMSH	TMS Hold Time	10	
TDIS	TDI Setup Time	15	
TDIH	TDI Hold Time	25	
TDOZX	TDO Float to Valid Delay		35
TDOXZ	TDI Valid to Float Delay		35
TDOV	TDO Valid Delay		35
TINS	I/O Setup Time	15	
TINH	I/O Hold Time	30	
TIOV	EXTEST Output Valid Delay		55

Terminating TAP pins

The XC9500 TDI and TMS pins have internal 10Kohm pull-up resistors, which are required by the 1149.1 standard. Because these pins are internally terminated, no further termination is required on the TAP connections.

Capacitive Decoupling

Each XC9500 device should have a 0.1µf and a 1.0µf capacitor connected directly between its VCC and GND pins. This helps to provide stable, noise-free power.

Calculating Maximal Chain Lengths

The XC9500 TAP pins have approximately 5 pF of signal loading. Because each TDI input is driven by only one TDO output (or equivalent single drive) there are no signal limitations related to those connections beyond those of standard board interconnect design rules.

The maximum TDO frequency will be 1/2 of the maximum TCK frequency. Because TCK and TMS are parallel driven signals the maximum number of parts in a single boundary-scan chain is determined by the ability of the TCK and TMS drivers to deliver the signals at the appropriate frequencies to the parts in the boundary-scan chain. Standard board-layout design rules also apply here.

If the boundary-scan chain includes more than 16 devices, buffers for TMS and TCK are recommended.

Part Enable Ordering

The ISPEX instruction allows the flexibility to enable parts in an arbitrary order. In some systems the order in which parts are enabled is critical. For instance, if a slave device awakens before its controller, it may enter an error condition from which it cannot exit.

The EZTag software enables each part immediately after programming. In concurrent mode the parts are enabled in order from system TDI to system TDO.

Creating Boundary-Scan Chains

There are a number of possibilities for creating boundary scan chains, several of which are discussed in the following sections

Single Port Serial Chain

The most simple and widely-used boundary-scan configuration is the single port serial chain shown in Figure 2, and only this type of configuration is supported by the EZTag software. In this configuration, four pins are allocated in the system to facilitate connection of the TCK (clock), TMS (mode), TDI (Test Data Input), and TDO (Test Data Output) signals.

All devices in the chain share the TCK and TMS signals. The system TDI signal is connected to the TDI input of the first device in the boundary-scan chain. The TDO signal from that first device is connected to the TDI input of the second device in the chain and so on. The last device in the chain has its TDO output connected to the system TDO pin.

Other more complex chain variations are discussed later in this application note. Software supplied by third party developers supports these complex chain configurations.

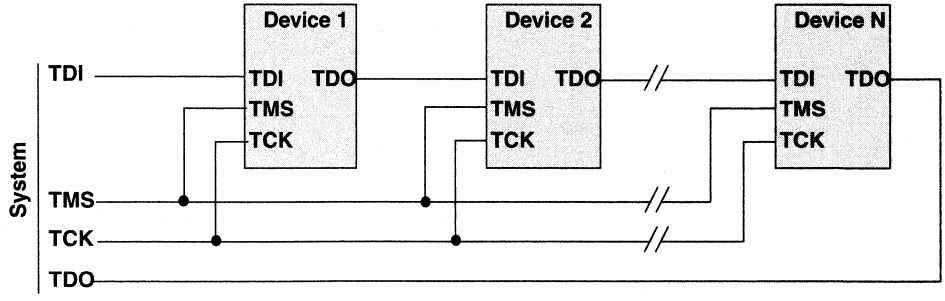


Figure 2: Single Port Serial Boundary-Scan Chain

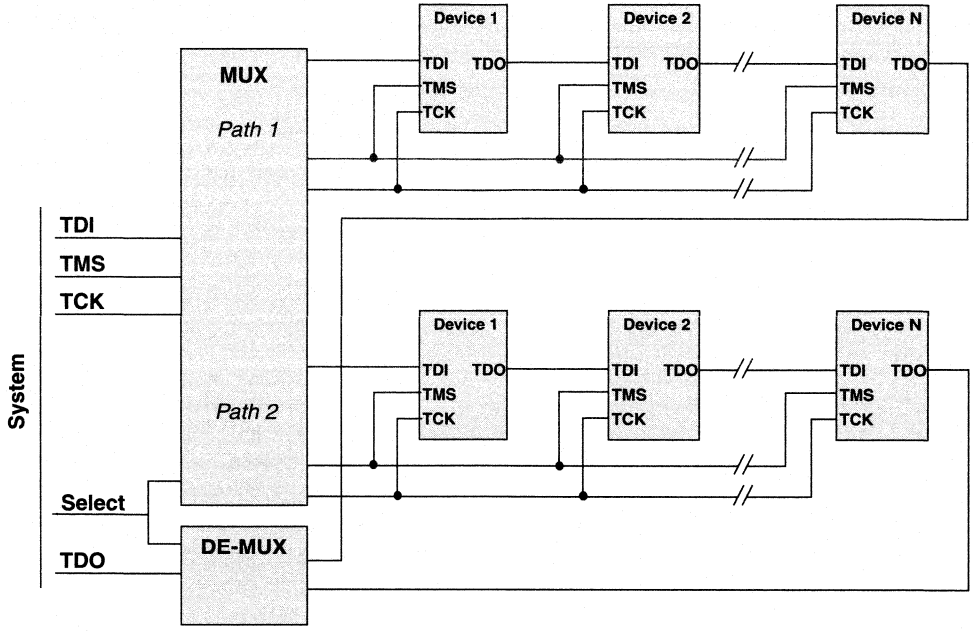


Figure 3: Star Configuration

Star Configuration

The single port serial chain, shown in Figure 3, configuration has a significant limitation due to the possibility that a defect in the backplane wiring or the removal of a board from the system will break the chain. This would make ISP and system testing impossible. In order to overcome this limitation and make the 1149.1 standard practical for very large systems, the standard allows the connection of boundary-scan chains in star configuration in which the four pins of the TAP are multiplexed. The costs of this approach are the additional overhead required to switch between scan paths, and the reduced TCK frequency due to TMS routing delays.

Multiple Independent Paths

In the topology shown in Figure 4, the TDI and TDO paths are independent allowing data to be streamed into and out of the portions of the system independently.

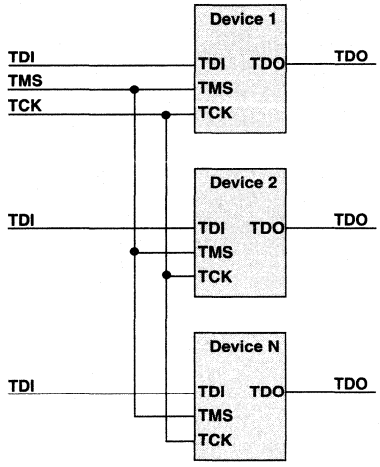


Figure 4: Multiple Independent Chains

Parallel Chains

In the topology shown in Figure 5, TDI and TMS inputs are independent but the TDO is shared. This means that although data can be streamed into portions of the system independently, data being streamed out is time multiplexed through TMS control.

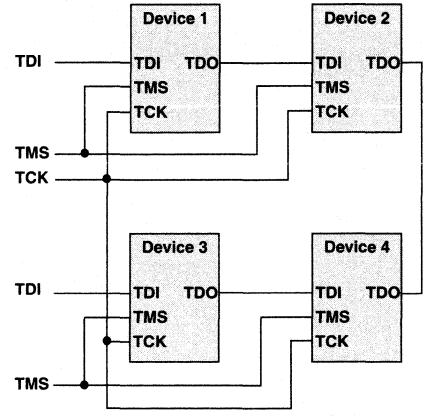


Figure 5: Parallel Chains

In-System Programming

Using the Xilinx Download Cables

The EZTag software can be used with either the Xilinx JTAG parallel download cable or the Xilinx serial XChecker cable.

IBM PC compatible systems can use the Xilinx high speed Parallel Cable III (part number DLC5). The cable pod includes port protection and drive circuitry which requires a 5V power supply that is usually is supplied by the target system. This cable operates TCK at a frequency of between 150KHz and 350Khz which is determined by the port speed of the host computer.

Sun and HP workstations (as well as IBM PC-compatible systems) use the Xilinx XChecker cable, which connects to the computer's serial port. The XChecker cable pod contains an XC3042 FPGA and 1Mbit of static RAM. The FPGA is configured to operate as a UART to facilitate host-cable communications. It also includes circuitry to enable high speed 1149.1 TAP signal processing the collecting TDO results in XChecker's static RAM. The TDO data can then be uploaded to the host. Like the parallel cable, the XChecker cable requires 5V to operate and this is usually provided by the target system.

When addressing the TAP the XChecker cable operates at TCK frequencies of approximately 1 MHz, which is controlled by a crystal in the XChecker cable pod. Although the cable TAP driver can operate at 1MHz, the overall speed of this cable is determined by the serial port throughput which is typically 38K baud.

The parallel and serial cable characteristics are:

- Power - A 5V power supply capable of providing 125 mA peak current and 60 mA steady state is required. The parallel cable requires a 5V power supply capable of providing 20 mA current.
- Drive Capabilities - The XChecker cable outputs are capable of sourcing or sinking up to 4 mA. The parallel cable outputs are capable of sourcing or sinking up to 20 mA.
- Special considerations - For both the XChecker and parallel download cables, the 1149.1 TAP drive electronics is in the cable pod. This should be taken into account when extending the signal reach from the pod to the system. When extending the cable from the port connection side, the drive capabilities of the host computer's port itself must be considered.

Concurrent Program and Erase Modes

One operating mode of the EZTag software performs concurrent erasing and programming. The advantage of this approach is speed; the overall programming time is dictated by the slowest part in the boundary-scan chain. Also, the total number of vectors required is optimized. The disadvantage of this approach is that the system must supply a peak operating current equal to that required by all parts being programmed or erased concurrently. For more information on how to use this feature in EZTag, please see the Xilinx *EZTag User's Guide*.

Note: although XC9500 parts can be programmed concurrently, the current EZTag software does not generate an SVF file that supports this operation. The current EZTag generates an SVF file that bypasses all parts except the one being programmed.

ISP Mode I/O Behavior

The functional pins of the device transition to a high-impedance state when ISP mode is entered using the ISPEN instruction. At the completion of an ISP programming or erase operation, the ISPEX instruction is executed. When leaving ISPEX mode (by shifting in an new boundary-scan instruction other than ISPEN), the device initializes to its programmed state; the functional pins take on their selected operations (input, output, or bidirectional) and the device registers take on their pre-selected initial values.

System-Level Design Issues

The normal operating mode of a system or a device in the system is known as mission mode which is different from test mode. When operating a device in boundary-scan test mode (such as when using either INTEST or EXTEST) as well as when performing ISP operations, the device is effectively disconnected from the overall system. When the operation is completed, the device is re-connected to the system. This can sometimes result in unpredictable system behavior. Additional discussion regarding this problem can be found in "The Boundary-Scan Handbook" by Ken Parker. Fortunately, the XC9500 family supplies two proprietary boundary-scan instructions that serve to alleviate this problem.

XC9500 Mission Mode Exit and Re-Entry Techniques

The XC9500 devices support two boundary-scan instructions that can be used to help alleviate the problems associated with exiting and re-entering mission mode. The instructions are ISPEN (ISP enable) and ISPEX (ISP exit).

- **ISPEN** - The ISPEN instruction is used at the beginning of every block of ISP operations that will attempt to access for alteration or read the device internal program memory (such as program, erase, verify, etc.). When the device is in ISPEN mode, the device I/O pins immediately enter a state in which they are floating with a weak pull-up resistor enabled on each pin. The device pins therefore neither drive nor sense external signal levels.
- **ISPEX** - The ISPEX instruction is used to conclude every block of ISP operations that have either been

read from or written to the device internal program memory. As long as the ISPEX instruction remains in the instruction register the functional pins remain in their lightly pulled-up high impedance state. Once the ISPEX instruction is replaced with any other boundary-scan instruction (except ISPEN), the device returns to its initial power state with the pins configured to their programmed states (input, output, or bidirectional) and with the device flip-flops taking on their initial states.

The ISPEX operation takes approximately 100 microseconds to complete. If the ISPEX instruction is held in the instruction register for longer than 100 microseconds, the ISPEX operation will not take effect until the ISPEX instruction is displaced from the instruction register.

In order to ensure safe operation, all INTEST,EXTEST, and ISP operations involving the XC9500 parts should be bracketed by ISPEN and ISPEX instructions.

The designer must also be careful to select an initial condition that is "system-safe" so that when the ISPEX instruction is released the XC9500 part in question will safely resume operation with the rest of the system.

Basic Boundary-Scan Design Guidelines

The following guidelines will help ensure a successful design.

- Make certain that all parts in the boundary-scan chain have 1149.1 compatible test access ports.
- Use simple buffering for TCK/TMS signals, to simplify test considerations for the boundary-scan TAP.
- Do not invert TCK or TMS pathways, to guarantee complete test software compatibility.
- Group similar device families, and have a single level converter interface between them, for TCK, TMS, TDI, TDO, and system pins.
- Check that the mission logic is safe from any possible errors that might arise while the boundary-scan data is being shifted through the boundary-scan chain. For example, pay close attention to bus enable or chip select signals that might be enabled simultaneously, causing unexpected bus contention.
- Provide the capability for the ATE to disable conventional (non boundary-scan) IC's whose run-time node values might introduce conflicts with boundary-scan logic values during test operations.
- Verify that the entire system is held in a benign state during boundary-scan test operations.
- Verify that the set-up and hold times of TDI and TMS with respect to TCK are met by the system.

Debugging Boundary-Scan Systems

The following guidelines and helpful information will help isolate potential problems.

- When traversing the IR states, the CAPTURE-IR value specified in the BSDL file is always shifted out on TDO at SHIFT-IR. This fact can be used to test boundary-scan chain continuity.
- After exit from Test-Logic-Reset, if the system transitions directly to Shift-DR, the values shifted out on TDO must be either the IDCODE (if implemented) or the BYPASS register contents. If all logic 0's are shifted in at TDI, then the first incidence of a logic 1 on TDO represents the first bit of an IDCODE. This fact can be used for blind interrogation of the boundary-scan chain and for further boundary-scan chain continuity checks.
- When entering ISP mode via the ISPEN instruction, all XC9500 function pins float to a weakly pulled-up high impedance state. The pins can easily be tested for this behavior.
- When ISPEX is shifted out of the instruction register, the XC9500 devices should take on their programmed values with the functional pins acting immediately as inputs or outputs, as programmed. The pins can easily be tested for this behavior.
- TDO assumes its defined value at the falling edge of TCK.
- When not in SHIFT-IR or SHIFT-DR, TDO exhibits high impedance.
- The last valid TDI bit clocks into the TAP with TMS high.
- In BYPASS mode, TDO equals the applied TDI data one TCK pulse earlier.

Conclusion

When designing ISP systems, common-sense rules related to electronic system design and board layout should be adhered to. In order to benefit from the synergies associated with the integration of test and programming operations the designer must consciously design with the entire system life cycle in mind.

References

IEEE 1149.1-1990 Std Test Access Port and Boundary-Scan Architecture

Colin Maunder and Rod Tulloss, The Test Access Port and Boundary-Scan Architecture, ISBN: 0-8186-9070-4.

Kenneth P. Parker, The Boundary-Scan Handbook, ISBN: 0-7923-9270-1.

Harry Bleeker et al., Boundary-Scan Test - A Practical Approach, ISBN: 0-792-9296-5.

Hideo Fujiwara, Logic Testing and Design for Testability, ISBN: 0-262-06096-5.

M. Montrose, Printed Circuit Board Design Techniques, ISBN: 0780311310.

XC9500 In-System Programming Using an Embedded Microcontroller

January, 1997 (Version 1.0) Application Note

Summary

The XC9500 high performance CPLD family provides in-system programmability, reliable pin locking, and JTAG boundary-scan test capability. This powerful combination of features allows designers to make significant changes and yet keep the original device pinouts, eliminating the need to re-tool PC boards. By using an embedded controller to program these CPLDs from an on-board RAM or EPROM, designers can easily upgrade, modify, and test designs, even in the field.

Xilinx Family

XC9500

Introduction

The XC9500 CPLD family combines superior performance with an advanced architecture to create new design opportunities that were previously impossible. The combination of in-system programmability, reliable pin locking, and JTAG test capability gives the following important benefits:

- Reduces device handling costs and time to market.
- Saves the expense of laying out new PC boards.
- Allows remote maintenance, modification, testing.
- Increases the life span and functionality of products.
- Enables unique, customer-specific features.

The ISP controller shown in Figure 1 can help designers achieve these unprecedented benefits by providing a simple means for automatically programming XC9500 CPLDs from design information stored in EPROM. This design is easily modified for remote downloading applications and the included C-code can be compiled for any microcontroller.

To create device programming files, Xilinx provides the EZTag™ software that automatically reads standard JEDEC device programming files and converts them to SVF format which contains both data and programming instructions for the CPLDs. These files are then converted to a compact binary format (XSVF) and can be stored in the on-board EPROM. The 8051 microcontroller interprets the XSVF information and generates the programming instructions, data, and control signals for the XC9500 CPLD.

By using a simple IEEE 1149.1 (JTAG) interface, XC9500 CPLDs are easily programmed and tested without using expensive hardware. Multiple devices can be daisy-chained, permitting a single 4-wire Test Access Port (TAP) to control any number of XC9500 CPLDs or other JTAG-compatible devices.

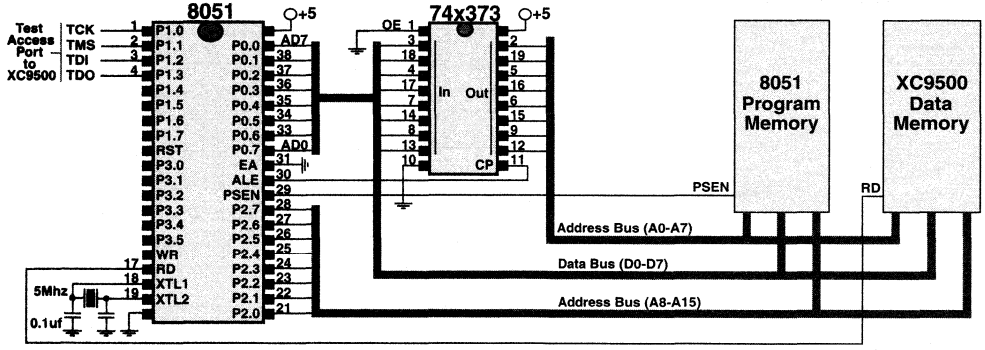


Figure 1: ISP Controller Schematic

Programming XC9500 CPLDs

Serial Vector Format (SVF) is a syntax specification for describing high level IEEE 1149.1 (JTAG) bus operations. SVF was developed by Texas Instruments and has been adopted as a standard for data interchange by JTAG test equipment and software manufacturers such as Teradyne, Tektronix, and others. XC9500 CPLDs accept programming and JTAG boundary-scan test instructions in SVF format, via the TAP. The timing for these TAP signals is shown in Figure 5.

The EZTag software automatically converts standard JEDEC programming files into SVF format. However, the SVF format is ASCII which is inefficient for embedded applications due to its memory requirements. Therefore, to minimize the memory requirements, SVF is converted into a more compact (binary) format called XSVF. In this design, an 8051 C-code algorithm interprets the XSVF file and provides the required JTAG TAP stimulus to the CPLD, performing the programming and (optional) test operations which were originally specified in the SVF file.

Note: For a description of the SVF and XSVF commands and file formats, see Appendix A and B.

The flow for creating the programming files that are used with this design, is shown in Figure 2.

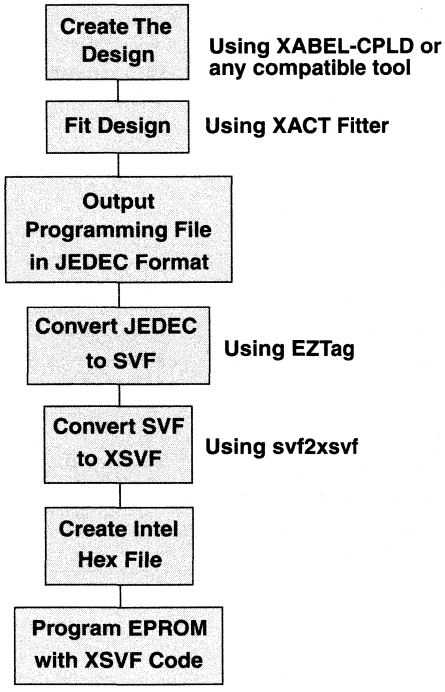


Figure 2: Program Flow

JTAG Instruction Summary

XC9500 devices accept both programming and test instructions via the JTAG TAP. The JTAG commands used for programming and functional test are:

- INTEST - Isolates the device from the system, applies test vectors to the device input pins, and captures the results from the device output pins.
- ISPEN - Enables the ISP function in the XC9500 device, floats all device function pins, and initializes the programming logic.
- FERASE - Erases a specified program memory block.
- FPGM - Programs specific bit values at specified addresses. An FPGMI instruction is used for the XC95216 and larger devices which have automatic address generation capabilities.
- FVFY - Reads the fuse values at specified addresses. An FVFYI instruction is used for the XC95216 and larger devices which have automatic address generation capabilities.
- ISPEX - Exits ISP Mode. The device is then initialized to its programmed function with all pins operable.
- SAMPLE/PRELOAD - Allows values to be loaded into the boundary scan register to drive the device output pins. Also captures the values on the input pins.
- BYPASS - Bypasses a device in a boundary scan chain by functionally connecting TDI to TDO.

The following instructions are also available in the XC9500 devices but are not used for programing or functional test:

- EXTEST - Isolates the device I/O pins from the internal device circuitry to enable connectivity tests between devices. It uses the device pins to apply test values and to capture the results.
- IDCODE - Returns a 32-bit hardwired identification code that defines the part type, manufacturer, and version number.
- USERCODE - Returns a 32-bit user-programmable code that can be used to store version control information or other user-defined variables.
- HIGHZ - Causes all device pins to float to a high impedance state.

Creating an SVF File Using EZTag

This procedure shows how to create an SVF file; it assumes that the Xilinx XACT version 6.0.0 software, or newer, is being used, which includes the XC9500 fitter and the EZTag software.

1. Create the design using XABEL-CPLD or any compatible third-party design entry tool.
2. Fit the design and save it to a JEDEC output file.
3. Invoke the EZTag software from the XACT command line using the following command:

```
eztag -svf
```

The following message appears:

```
Xilinx (R) EZTAG XC9500-CPLD-6.0.0 - JTAG
Boundary-Scan Download

Copyright (C) Xilinx Inc. 1991-1995. All
Rights Reserved.

-------------------------------------------------------------

SVF GENERATION MODE.
EZTAG ?
```

4. At the **EZTAG ?** prompt type the following command:

   ```
   part deviceType1:designName1
        deviceType2:designName2
        deviceTypeN:designNameN <CR>
   ```

   deviceType is the name of the BSDL file for that device and ***designName*** is the name of the design to translate into SVF. Multiple *deviceType:designName* pairs are separated by spaces. For example:

   ```
   part xc95108:abc12 xc95144:wxyz
   ```

   This command defines the JTAG device chain, from one to any number of devices. The parts specified in the **part** command should be arranged in order beginning with the first device to receive TDI and ending with the last device to output TDO.
   Note: For any non-XC9500 part in the JTAG chain make certain that the BSDL file for the specified part is available along the XACT path and is called *deviceType*.bsd.

5. Enter any one of the following commands:

- **erase *designName*** — generates an SVF file for the bit sequence needed to erase the specified part.
- **verify *designName* [-j *jedecFileName*]** — generates an SVF file that specifies the bit sequence to read back the device contents and compare it against the contents of the specified JEDEC file.
- **program [-v] *designName* [-j *jedecFileName*]** — generates an SVF file that specifies the bit sequence to program the specified part from a JEDEC file named *designName*.jed (or alternately, the JEDEC file name specified after the "-j"). The program command option add the following functionality:
  -v — Follow up the programming operation with a read-back verification against the contents of the JEDEC file.

6. Exit EZTag by entering the following command:

   ```
   quit
   ```

   NOTE: The SVF file will be named *designName*.svf, and will be created in the current working directory (the directory in which EZTAG is being run). Consecutive operations on the same *designName* file will overwrite the SVF file each time. The SVF file contains all data and commands necessary to perform the specified function.

7. Use the svf2xsvf tool to translate the ASCII SVF file into a more compact binary XSVF format. XSVF files are approximately 80% smaller than SVF files. To perform the conversion, enter the following command at the system prompt:

   ```
   svf2xsvf svf_file_name xsvf_file_name
   ```

   Note: The svf2xsvf translator supports only SDR, SIR, and RUNTEST instructions in the source SVF file.

The XSVF file now contains both the programming data and instructions, ready for use by the 8051 microcontroller.

EPROM Programming

To program an EPROM, the binary XSVF file must be converted to an Intel Hex or similar PROM format file. Most embedded processor development system software will automatically convert included binary files to the appropriate format. Public domain file conversion software is also available, as shown in Appendix D.

Software Limitations

EZTAG can generate SVF files only for devices for which JEDEC files can be created. Designers should verify that the development software they are using can create JEDEC files for the specific devices they intend to use.

The current software can only generate SVF files for operations on one part at a time. If there are several parts to be programmed, additional program commands must be executed — one for each part, creating multiple SVF files. In each SVF file, one device will be programmed while the others are held in bypass mode.

Hardware Design

As shown in Figure 1, this design requires only an 8051 microcontroller, an address latch, and enough EPROM or RAM to contain both the 8051 code and the CPLD programming data.

Hardware Design Description

The 8051 allows 64K of program and 64K of data space; much more than is needed in this application. However the ability to separate address and data space is used to simplify the addressing scheme.

The 8051 multiplexes port 0 for both data and addresses. The ALE signal causes the 74x373 to latch the low order address, and the high order address is output on port 2. Port 0 then floats, allowing the selected EPROM to drive the data inputs. Then the $\overline{PSEN}$ signal goes low to activate an 8051 program read operation, or the $\overline{RD}$ signal goes low to activate a CPLD programming data read operation.

Estimated EPROM Memory Requirements

Table 1 shows the estimated EPROM capacity needed to contain both the 8051 code and the XC9500 programming

data. The XSVF file sizes are shown for an erase and program operation.

Table 1: XSVF File Sizes

Device Type	XSVF File Size	C-Code	Total
XC9536	5194	7k	12K
XC9572	11674	7k	19K
XC95108	19598	7k	27K
XC95144	12960 (estimated)	7k	20K
XC95180	16200 (estimated)	7k	23K
XC95216	26390	7k	33K
XC95288	34560 (estimated)	7k	42K

The XSVF file sizes are dependent only on the device type, not on the design implementation. If further compression of the XSVF file is needed, a standard compression technique, such as Lempel-Ziv can be used.

Modifications for Other Applications

The design presented in this application note is for a standalone ISP controller. However, it is also possible to apply these techniques to microcontrollers that may already exist within a design. To implement this design in an already existing microcontroller, all that is needed is four I/O pins to drive the TAP, and enough storage space to contain both the controller program and the CPLD download data. In addition, care must be taken to preserve the JTAG port timing.

The TAP timing in this design is dependent on the 8051 clock. For other 8051 clock frequencies or for different microcontrollers, the timing must be calculated accordingly, in order to implement the timing specified in Figure 5.

Using a different microcontroller would require changing the I/O subroutine calls while preserving the correct TAP timing relationships. These subroutine calls are located in the ports.c file. All other C-code is independent of the microcontroller and will not need to be modified.

RAM can be used instead of the EPROM in this design. This would allow the XC9500 devices to be programmed and tested remotely via modem, using remote control software written by the user.

Debugging Suggestions

The following suggestions may be helpful in testing this design:

- View the contents of the XSVF file using the xsvf2ascii converter. This will decode the binary file and display the XSVF data and instructions. To run this converter, enter the following command at the system prompt:

  ```
  xsvf2ascii
  ```

- Change the **#define DEBUG_MODE 0** to **#define DEBUG_MODE 1** in the ports.h file to see the calculated values of the TDI and TMS ports on the rising edge of TCK, when the code is compiled. Use this to verify the functionality of the C-code if it is ported to a different microcontroller. (See Appendix C for more information.)
- Use the ASCII text output, generated by xsvf2ascii, to verify that the bit sequence output of the microcontroller is correct.
- Decrease the TCK frequency to test that the wait times for program and erase are sufficiently long.
- Make certain that the function pins go into a 3-state condition in ISP mode.
- Test that the function pins initialize when ISP mode is terminated with the ISPEX command.
- Verify that the devices which are not being programmed are in bypass mode. Bypass mode causes TDO to be the same as TDI, delayed by one TCK clock pulse.

Firmware Design

The flow chart for the C-Code is shown in Figure 3. This code continuously reads the instructions and arguments from the XSVF file contained in the XC9500 program data EPROM and branches in one of three ways based on the three possible XSVF instructions (XRUNTEST, XSIR, XSDR) as described in Appendix B.

When the C-Code reads an XRUNTEST instruction, it reads in the next four bytes of data that specify the number of microseconds for which the device will stay in the Run-Test/Idle state before the next XSIR or XSDR instruction is executed. The runTestTimes variable is used to store this value.

When the C-Code reads an XSIR instruction, it provides stimulus to the TMS and TCK ports until it arrives in the Shift-IR state. It then reads a byte that specifies the length of the data and the actual data itself, outputting the specified data on the TDI port. Finally, when all the data has been output to the TDI port, the TMS value is changed and successive TCK pulses are output until the Run-Test/Idle state is reached again.

When the C-Code reads an XSDR instruction, it reads the data specifying the values that will be output during the Shift-DR state. The code then toggles TMS and TCK appropriately to transition directly to the Shift-DR state. It then holds the TMS value at 0 in order to stay in the Shift-DR state and the data from the XSVF file is output to the TDI port while storing the data received from the TDO port. After all the data has been output to the TDI port, TMS is set to 1 in order to move to the Exit-1-DR state. Then, the TDO input value is compared to the TDO expected value. If the two values fail to match, the exception handling procedure is executed as shown in Figure 6. If the TDO input values match the expected values, the code returns to the Run-Test/Idle state and waits for the amount of time specified by the runTestTimes variable (which was originally set in the XRUNTEST instruction).

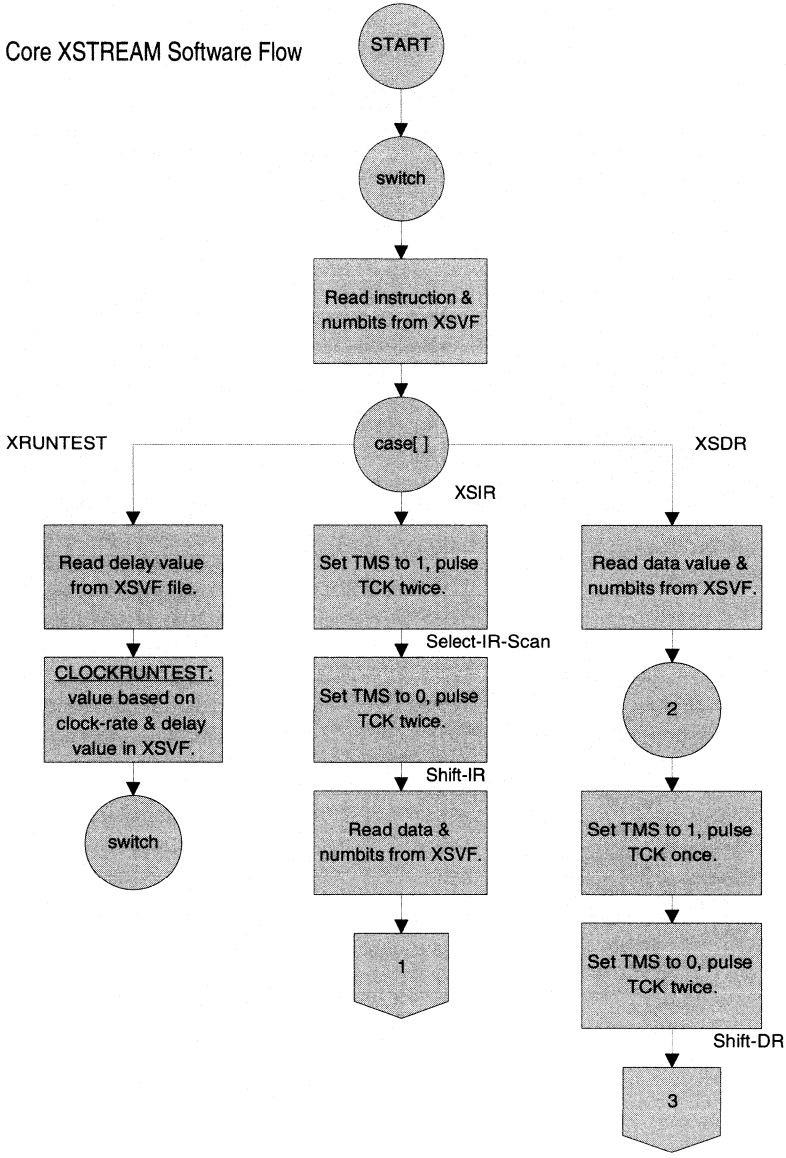


Figure 3: Flow Chart for the ISP Controller Code

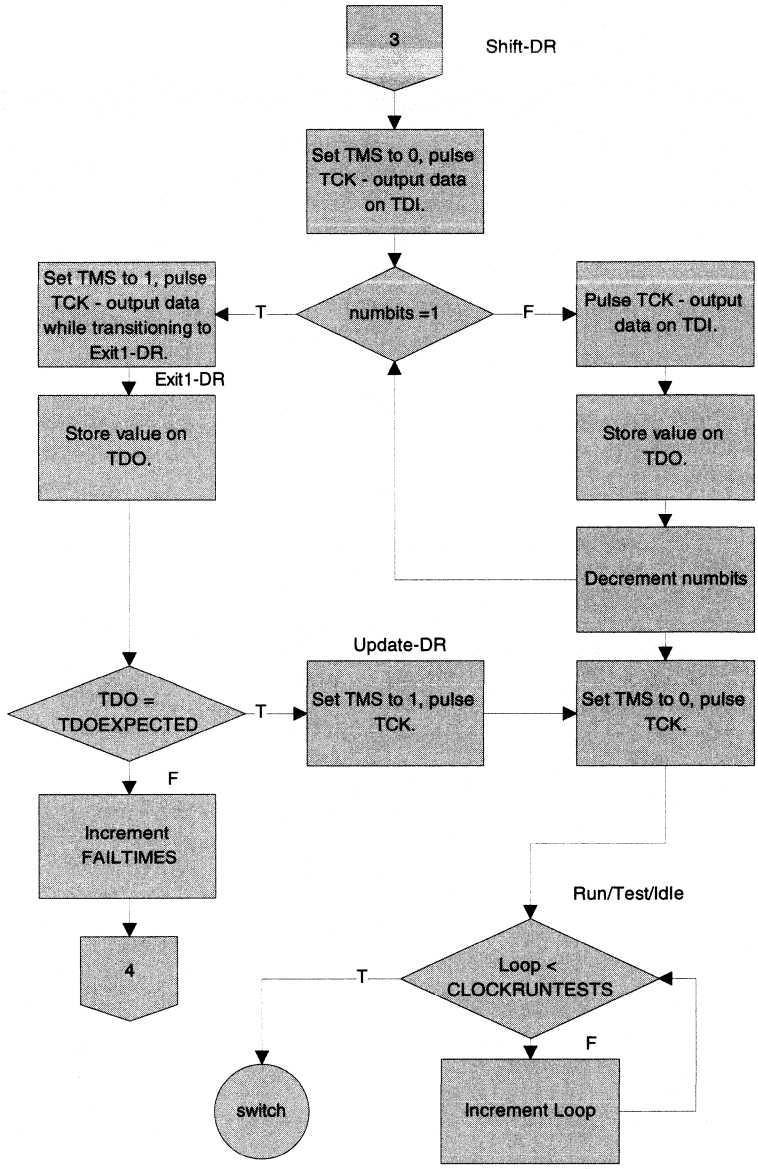


Figure 3: Continued

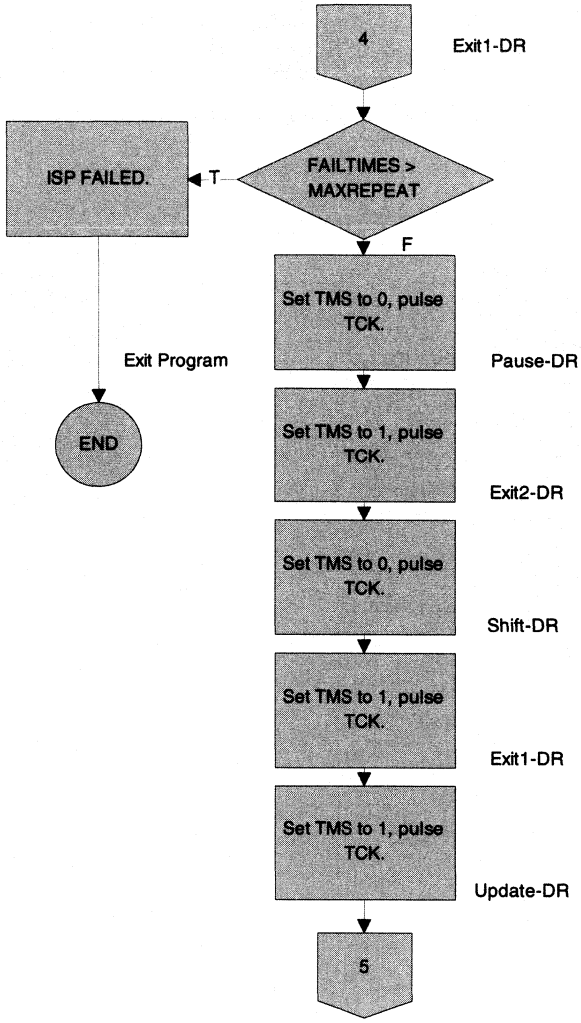


Figure 3: Continued

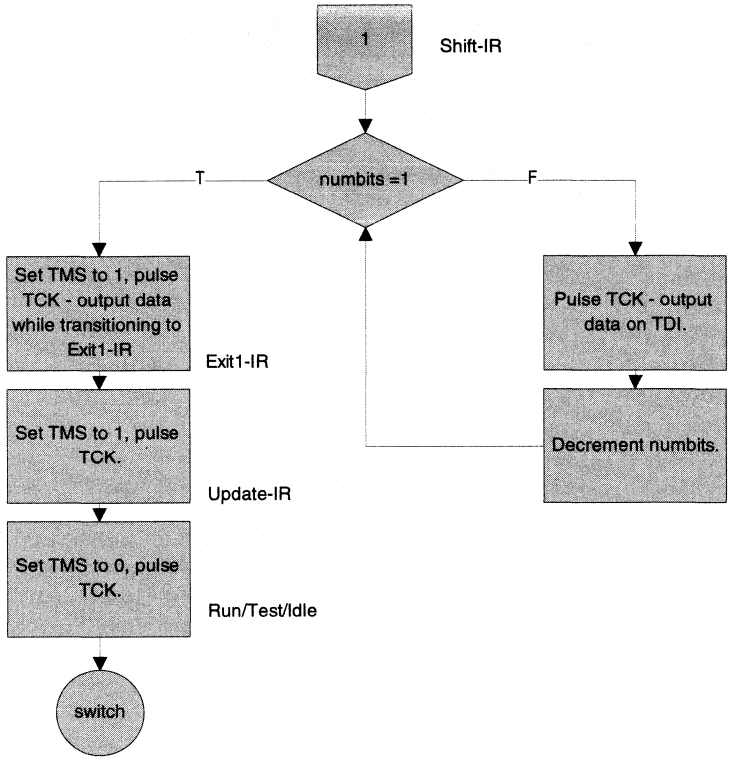


Figure 3: Continued

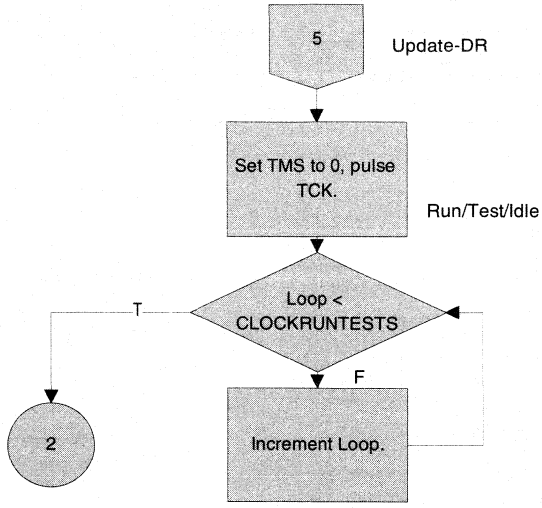


Figure 3: Continued

Memory Map

The 8051 memory map is divided into two 64K byte blocks: one for the 8051 program and one for data. The 8051 program memory resides in the 8051 program block and is enabled by the $\overline{PSEN}$ signal. The XC9500 CPLD program memory resides in the 8051 data block and is enabled by the $\overline{RD}$ signal.

Port Map

The 8051 I/O ports are used to generate the memory address and the TAP signals, as shown in Figure 1. Port 1 of the 8051 is used to control the TAP signals; Table 2 shows the port configuration.

Table 2: 8051 Port 1 Mapping

TAP Pin	Port1 Bit	Configured as
TCK	0	Input
TMS	1	Input
TDI	2	Input
TDO	3	Output

TAP Timing

Figure 5 shows the timing relationships of the TAP signals. The C-code running on the 8051 insures that the TDI and TMS values are driven at least two instruction cycles before asserting TCK. At that same time, TDO can be strobed.

Parts of the XSVF file specify wait times during which the device programs or erases the specified location or sector. Implementation of the wait timer can be accomplished either by software loops that depend on the processor's cycle time or by using the 8051's built-in timer function. In this design, timing is established through software loops in the ports.c file.

TAP AC Parameters

Table 3 shows the timing parameters for the TAP waveforms, shown in Figure 5.

Table 3: Test Access Port Timing Parameters (ns.)

Symbol	Parameter	Min	Max
TCKMIN	TCK Minimum Clock Period	100	
TMSS	TMS Setup Time	10	
TMSH	TMS Hold Time	10	
TDIS	TDI Setup Time	15	
TDIH	TDI Hold Time	25	
TDOZX	TDO Float to Valid Delay		35
TDOXZ	TDI Valid to Float Delay		35
TDOV	TDO Valid Delay		35
TINS	I/O Setup Time	15	
TINH	I/O Hold Time	30	
TIOV	EXTEST Output Valid Delay		55

XC9500 Programming Algorithm

This section describes the programming algorithm executed by the 8051 C-code that reads the XSVF file; this code is contained in the micro.c file in Appendix C. This information is valuable to users who want to modify the C-code for porting to other microcontrollers.

The XSVF file contains all XC9500 programming instructions and data. This allows the TAP driver code to be very simple. The 8051 interprets the XSVF instructions that describe the CPLD design and then outputs the TAP signals for programming (and testing) the XC9500 device. The command sequence for device programming is shown in Figure 4.

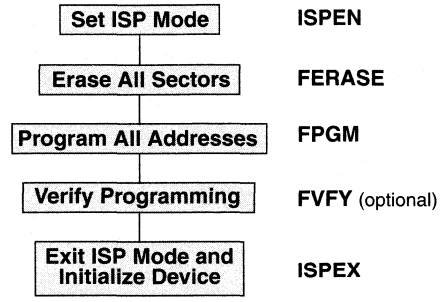


Figure 4: Device Programming Flow

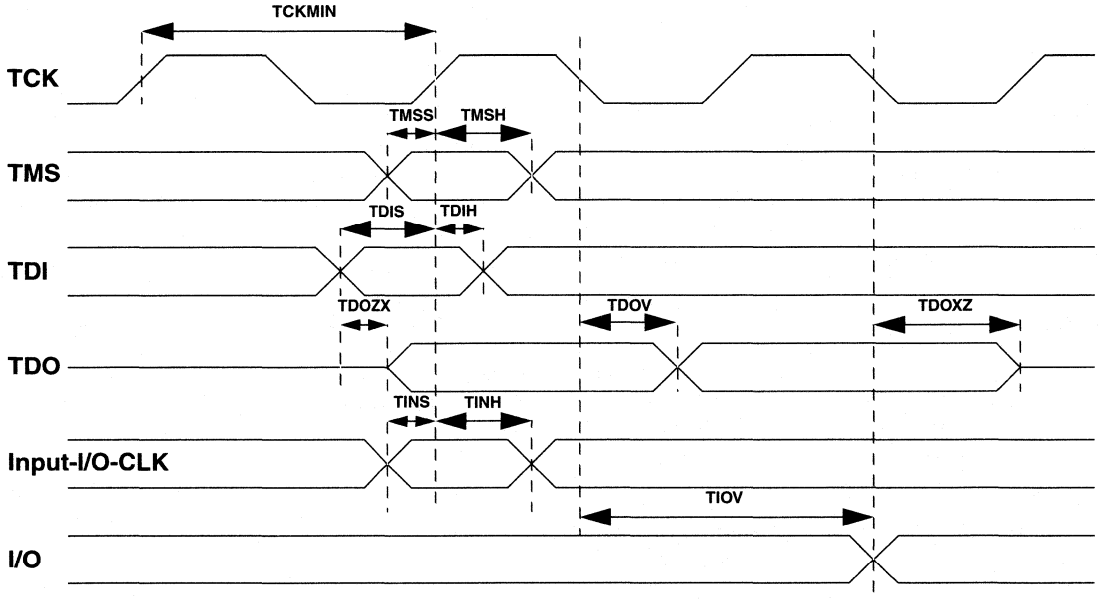


Figure 5: Test Access Port Timing

Exception Handling

Figure 6 shows the state diagram for the internal device programming state machine, as defined by the IEEE 1149.1 standard. The C-code drives the 1149.1 TAP controller through the state sequences to load data and instructions, and capture results. One of the key functions performed by the C-code is the TAP controller state transition sequence that is executed when a program or erase operation needs to be repeated, which may occur on a small percentage of addresses. If a sector or address needs to be re-programmed or re-erased, the device status bits return a value that is different from that which is predicted in the XSVF file. In order to retry the previous (failed) data, the following 1149.1 TAP state transition sequence is followed, if the TDO mismatch is identified at the EXIT1-DR state:

```
EXIT1-DR, PAUSE-DR, EXIT2-DR, SHIFT-DR,
EXIT1-DR, UPDATE-DR, RUN-TEST/IDLE
```

The application then waits for the amount of time that was previously specified by XRUNTEST. The effect of this state sequence is to re-apply the previous value rather than apply the new TDI value that was just shifted in.

This "exception handling loop" is attempted no more than 32 times. If the TDO value does not match after 32 attempts, the part is defective and a failure is logged. When the retry operation is successful, the algorithm shifts-in the next XSDR data.

Conclusion

XC9500 CPLDs are easily programmed by an embedded processor. And, because the XC9500 family is 1149.1 compliant, system and device test functions can also be controlled by the embedded processor, in addition to programming. This capability opens new possibilities for upgrading designs in the field, creating user-specific features, and remote downloading of CPLD programs.

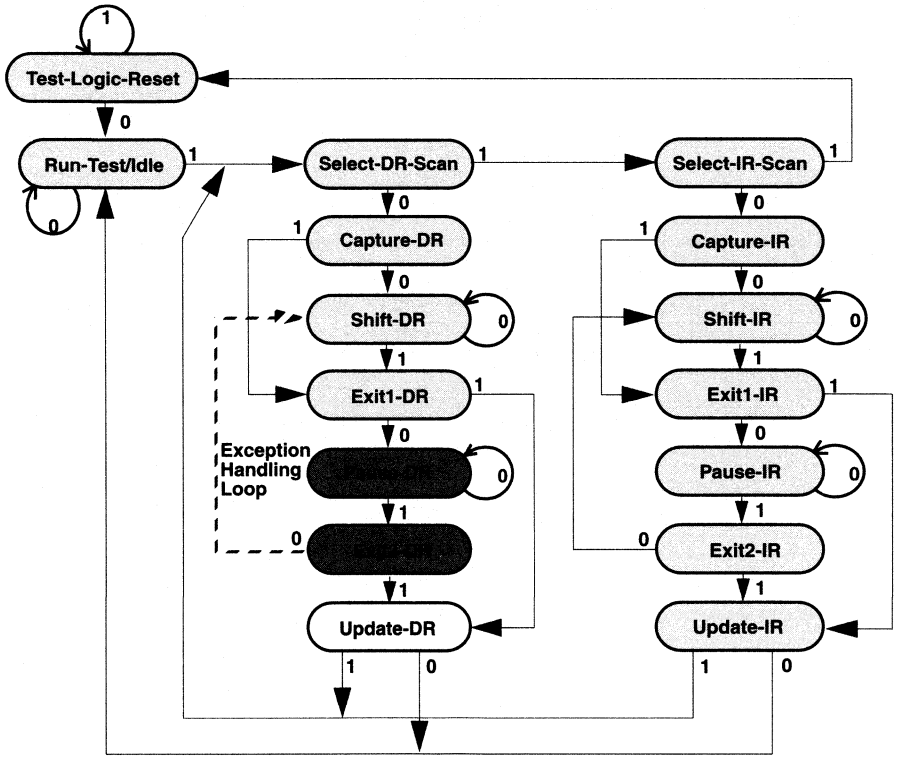


Figure 6: TAP State Machine Flow

Note: The values shown adjacent to each transition represent the signal present at TMS during the rising edge of TCK.

Appendix A

SVF File Format for the XC9500 Family

SVF Overview

This appendix describes the Serial Vector Format syntax, as it applies to the XC9500 family; only those commands and command options that apply to XC9500 devices are described. An SVF file is the media for exchanging descriptions of high-level IEEE 1149.1 bus operations which consist of scan operations and movements between different stable states on the 1149.1 state diagram (as shown in Figure 6). SVF does not explicitly describe the state of the 1149.1 bus at every Test Clock (TCK).

An SVF file contains a set of ASCII statements. Each statement consists of a command and its associated parameters, terminated by a semicolon. SVF is case sensitive, and comments are indicated by an exclamation point (!).

Scan data within a statement is expressed in hexadecimal and is always enclosed in parenthesis. The scan data cannot specify a data string that is larger than the specified bit length; the Most Significant Bit (MSB) zeros in the hex string are not considered when determining the string length. The bit order for scan data defines the LSB (rightmost bit) as the first bit scanned into the device for TDI and SMASK scan data, and is the first bit scanned out for TDO and MASK data.

SVF Commands

The following SVF Commands are supported by the XC9500 Family:

- SDR (Scan Data Register).
- SIR (Scan Instruction Register).
- RUNTEST.

For each of the following command descriptions:

- The parameters are mandatory.
- Optional parameters are enclosed in brackets ([ ]).
- Variables are shown in *italics*.
- Parenthesis "( )"are used to indicate hexadecimal values.
- A scan operation is defined as the execution of an SIR or SDR command and any associated header or trailer commands.

SDR, SIR

SDR *length* TDI (*tdi*) SMASK (*smask*)
[TDO (*tdo*) MASK (*mask*)];

SIR *length* TDI (*tdi*) TDO SMASK (*smask*);

These commands specify a scan pattern to be applied to the target scan registers. The SDR command (Scan Data Register) specifies a data pattern to be scanned into the target device Data Register. The SIR command (Scan Instruction Register) specifies a data pattern to be scanned into the target device Instruction Register.

Prior to scanning the values specified in these commands, the last defined header command (HDR or HIR) will be added to the beginning of the SDR or SIR data pattern and the last defined trailer command (TDR or TIR) will be appended to the end of the SDR or SIR data pattern.

Parameters:

length — A 32-bit decimal integer specifying the number of bits to be scanned.

[TDI (*tdi*)] — (optional) The value to be scanned into the target, expressed as a hex value. If this parameter is not present, the value of TDI to be scanned into the target device will be the TDI value specified in the previous SDR/SIR statement. If a new scan command is specified, which changes the length of the data pattern with respect to a previous scan, the TDI parameter must be specified, otherwise the default TDI pattern is undetermined and is an error.

[TDO (*tdo*)] — (optional) The test values to be compared against the actual values scanned out of the target device, expressed as a hex string. If this parameter is not present, no comparison will be performed. If no TDO parameter is present, the MASK will not be used.

[MASK (*mask*)] — (optional) The mask to be used when comparing TDO values against the actual values scanned out of the target device, expressed as a hex string. A "0" in a specific bit position indicates a "don't care" for that position. If this parameter is not present, the mask will equal the previously specified MASK value specified for the SIR/SDR statement. If a new scan command is specified which changes the length of the data pattern with respect to a previous scan, the MASK parameter must be specified, otherwise the default MASK pattern is undefined and is an error. If no TDO parameter is present, the MASK will not be used.

[SMASK (*smask*)] — (optional) Specifies which TDI data is "don't care", expressed as a hex string. A "0" in a specific bit position indicates that the TDI data in that bit position is a "don't care". If this parameter is not present, the mask will equal the previously specified SMASK value specified for the SDR/SIR statement. If a new scan command is specified which changes the length of the data pattern with respect to a previous scan, the SMASK parameter must be specified, otherwise the default SMASK pattern used is undefined and is an error. The SMASK will be used even if the TDI parameter is not present.

Example:

```
SDR 27 TDI (008003fe) SMASK (07ffffff)
TDO (00000003) MASK (00000003) ;

SIR 16 TDO (ABCD);
```

RUNTEST

RUNTEST *run_count* TCK;

This command forces the target 1149.1 bus to the Run-Test/Idle state for a specific number of microseconds, then moves the target device bus to the IDLE state. This is used to control RUNBIST operations in the target device.

Parameters:

run_count — The number of TCK clock periods that the 1149.1 bus will remain in the Run Test/Idle state, expressed as a 32 bit unsigned number.

Example:

```
RUNTEST 1000 TCK;
```

```
! Begin Test Program

TRST OFF;                    !disable test reset line
ENDIR IDLE;                  !End IR scan in IDLE
HIR
HDR 16 TDI (FFFF) TDO (FFFF) MASK (FFFF); !16 bit DR Header
TIR
TDR
SIR
SDR
STATE
RUNTEST
!End test program
```

Figure 7: Sample SVF File

Appendix B

XSVF File Format and Conversion Utilities

This appendix includes the following reference information:

- **The XSVF Commands** — The instructions that are supported, their arguments, and definitions.
- **The svf2xsvf Utility** — Converts the standard SVF file format to the more compact binary XSVF format.
- **The xsvf2ascii Utility** — Converts the XSVF file format to ascii text for debugging purposes.

XSVF Commands

The following commands describe the 1149.1 operations in a way that is similar to the SVF syntax. The key difference between SVF and XSVF is that the XSVF file format affords better data compression and therefore produces smaller files.

The format of the XSVF file is a one byte instruction followed by a variable number of arguments (as described in the command descriptions below). The binary (hex) value for each instruction is shown in Table 4:

Table 4: Binary Encoding of XSVF Instructions

XSVF Instruction	Binary Encoding (hex)
XCOMPLETE	0x00
XTDOMASK	0x01
XSIR	0x02
XSDR	0x03
XRUNTEST	0x04
XREPEAT	0x07
XSDRSIZE	0x08
XSDRTDO	0x09
XSETSDRMASKS	0x0a
XSDRINC	0x0b

XTDOMASK

```
XTDOMASK value<"length" bits>
```

XTDOMASK sets the TDO mask which masks the value of all TDO values from the SDR instructions. Length is defined by the last XSDRSIZE instruction. XTDOMASK may be used multiple times in the XSVF file if the TDO mask changes for various SDR instructions.

Example:

```
XTDOMASK 0x00000003
```

This example defines that TDOMask is 32 bits long and equals 0x00000003

XREPEAT

```
XREPEAT times<1 byte>
```

Defines the number of times that TDO will be tested against the expected value before the ISP operation will be considered a failure. By default, a device may fail an XSDR instruction 32 times before the ISP operation is terminated as a failure. This instruction is optional.

Example:

```
XREPEAT 0x0f
```

This example sets the command repeat value to 15.

XRUNTEST

```
XRUNTEST time<4 bytes>
```

Defines the amount of time (in microseconds) the device should sit in the Run-Test/Idle state after each visit to the SDR state.

Example:

```
XRUNTEST 0x00000fa0
```

This example specifies an idle time of 4000 microseconds.

XSIR

```
XSIR length<1 byte> TDIValue<"length"
bits>
```

Go to the Shift-IR state and shift in the TDIValue.

Example:

```
XSIR 0x08 0xec
```

XSDR

```
XSDR TDIValue<"length" bits>
```

Go to the Shift-DR state and shift in TDIValue; compare the TDOExpected value from the last XSDR instruction against the TDO value that was shifted out (use the TDOMask which was generated by the last XTDOMASK instruction). Length comes from the XSDRSIZE instruction.

If the TDO value does not match TDOExpected, return to the Run-Test/Idle state again, and wait the amount of time last specified by the XRUNTEST command, then try the SIR instruction again. If TDO is wrong more than the maximum number of times specified by the XREPEAT instruction, then the ISP operation will be determined to have failed.

Example:

```
XSDR 02c003fe
```

XSDRSIZE

```
XSDRSIZE length<4 bytes>
```

Specifies the length of all XSDR/XSDRTDO records that follow.

Example:

```
XSDRSIZE 0x0000001b
```

This example defines the length of the following XSDR/XSDRTDO arguments to be 27 bits (4 bytes) in length.

XSDRTDO

```
TDIValue<"length" bits>
TDOExpected<"length" bits>
```

Go to the Shift-DR state and shift in TDIValue; compare the TDOExpected value against the TDO value that was shifted out (use the TDOMask which was generated by the last XTDOMASK instruction). Length comes from the XSDRSIZE instruction.

If the TDO value does not match TDOExpected, return to the Run-Test/Idle state again, and wait the amount of time last specified by the XRUNTEST command, then try the SIR instruction again. If TDO is wrong more than the maximum number of times specified by the XREPEAT instruction, then the ISP operation will be determined to have failed.

The TDOExpected Value will be used in all successive XSDR instructions until the next XSDR instruction is given.

Example:

```
XSDRTDO 0x000007fe 0x00000003
```

For this example, go to the Shift-DR state and shift in 0x000007fe. Perform a logical AND on the TDO shifted out and the TDOMASK from the last XTDOMASK instruction and compare this value to 0x00000003.

XSETSDRMASKS

```
XSETSDRMASKS addressMask<"length" bits>
dataMask<"length" bits>
```

Set SDR Address and Data Masks. The address and data mask of future XSDRINC instructions are indicated using the XSETSDRMASKS instructions. The bits that are 1 in addressMask indicate the address bits of the XSDR instruction; those that are 1 in dataMask indicate the data bits of the XSDR instruction. "Length" comes from the value of the last XSDRSize instruction.

Example:

```
XSETSDRMASKS  00800000 000003fc
```

XSDRINC

```
XSDRINC startAddress<"length" bits>
numTimes<1 byte> data[1]<"length2" bits>
...data[numTimes]<"length2" bits>
```

Do successive XSDR instructions. Length is specified by the last XSDRSIZE instruction. Length2 is specified as the number of 1 bits in the dataMask section of the last XSETSDRMASKS instruction.

The startAddress is the first XSDR to be read in. For numTimes iterations, increment the address portion (indicated by the addressMask section of the last XSETSDRMASKS instruction) by 1, and load in the next data portion into the dataMask section.

Note that an XSDRINC <start> 255 data0 data1 ... data255 actually does 256 SDR instruction since the start address also represents an SDR instruction

Example:

```
XSDRINC  004003fe 05 ff ff ff ff ff
```

XCOMPLETE

```
XCOMPLETE
```

End of XSVF file reached.

Example:

```
XCOMPLETE
```

svf2xsvf File Conversion Utility

This executable reads in an SVF file (generated by EZTag) and generates an XSVF file.

Usage:

```
svf2xsvf [-nc] [-r number] <file1>
<file2>
```

file1: SVF input file name.
file2: XSVF output file name.

Options:

-nc — No compression. Don't use the XSETSDRMASKS and XSDRINC instructions.

-r *number* — Set the XREPEAT value to number

xsvf2ascii File Conversion Utility

This executable reads in an XSVF file (generated by svf2xsvf) and outputs the XSVF commands contained in the file. It is useful for debugging.

Usage:

```
xsvf2ascii <file1> <file2>
```

file1: XSVF input file name.
file2: ascii output file name.

Appendix C

C-Code Listing

The following files contain the C source code used to read an XSVF file and output the appropriate Test Access Port control bits:

C-Code Files

- lenval.c — This file contains routines for using the lenVal data structure.
- micro.c — This file contains the main function call for reading in a file from an EPROM and driving the JTAG signals.
- ports.c — This file contains the routines to output values on the JTAG ports, to read the TDO bit, and to read a byte of data from the EPROM.

Header Files

- lenval.h — This file contains a definition of the lenVal data structure and extern procedure declarations for manipulating objects of type lenVal. The lenVal structure is a byte oriented type used to store an arbitrary length binary value.
- ports.h — This file contains extern declarations for providing stimulus to the JTAG ports.

To compile this C-code for a microcontroller other than the 8051, only four functions within the ports.c file need to be modified:

- setPort — Sets a specific port on the microcontroller to a specified value.
- readTDOBit — Reads the TDO port.
- readByte — Reads a byte of data from the XSVF file.
- waitTime — Pauses for a specified amount of time.

For help in debugging the code, a compiler switch called DEBUG_MODE is provided. This switch allows the designer to simulate the TAP outputs in a PC environment. If DEBUG_MODE is defined, the software reads from an XSVF file (which must be named prom.bit) and prints the calculated value of the microcontroller's I/O ports (TDI and TMS) on each rising edge of TCK. Because the TDO value cannot be read during DEBUG_MODE, the software assumes that the TDO value is correct. This function provides a simulation of the TAP signals that can be used to verify the actual operation.


```
/*****************************************************/
/* file: lenval.c                                    */
/* abstract:  This file contains routines for using  */
/*            the lenVal data structure.             */
/*****************************************************/
#include "lenval.h"
#include "ports.h" /* for DEBUG_MODE define */

/* return the value represented by this lenval */
long value(lenVal *x)
{
    int i;
    long result=0;          /* result to hold the accumulated result */
    for (i=0;i<x->len;i++)
    {
       result=result<<8;    /* shift the accumulated result */
       result+=x->val[i];   /* get the last byte first */
    }
    return result;
}

/* set x to value; assumes value<512 */
void initLenVal(lenVal *x, long value)
{
    x->len=1;
    x->val[0]=(unsigned char) value;
}
```

```
/* return TRUE iff actual=expected (after masking out some bits using mask */
short EqualLenVal(lenVal *expected, lenVal *actual, lenVal *mask)
{
    int i;
#ifdef DEBUG_MODE
    /* since we can't read TDO, just assume TDO matches whatever is expected */
    return 1;
#endif
    for (i=0;i<expected->len;i++)
    {
        unsigned char byteVal1=expected->val[i];    /* i'th byte of expected */
        unsigned char byteVal2=actual->val[i];      /* i'th byte of actual   */
        byteVal1 &= mask->val[i];                   /* mask out expected     */
        byteVal2 &=mask->val[i];                    /* mask out actual       */
        if (byteVal1!=byteVal2)
            return 0;                               /* values are not equal  */
    }
    return 1;                                        /* values are equal      */
}



/* return the (byte, bit) of lv (reading from left to right) */
short RetBit(lenVal *lv, int byte, int bit)
{
    int i;
    unsigned char ch=lv->val[byte];   /* get the correct byte of data */
    unsigned char mask=128; /* 10000000 */
    for (i=0;i<bit;i++)
        mask=mask>>1;        /* mask the correct bit of the byte       */
    return ((mask & ch) !=0);  /* return 1 if the bit is 1, 0 otherwise */
}

/* set the (byte, bit) of lv equal to val (e.g. SetBit("00000000",byte, 1)
   equals "01000000" */
void SetBit(lenVal *lv, int byte, int bit, short val)
{
    int i;
    unsigned char *ch=&(lv->val[byte]);
    unsigned char OrMask=1, AndMask=255;
    for (i=0;i<7-bit;i++)
        OrMask=OrMask<<1;
    AndMask-=OrMask;
    *ch = *ch & AndMask;  /* 0 out the bit */
    if (val)
        *ch = *ch | OrMask;   /* fill in the bit with the correct value */
}

/* add val1 to val2 and store in resVal;      */
/* assumes val1 and val2  are of equal length */
void addVal(lenVal *resVal, lenVal *val1, lenVal *val2)
{
    unsigned char carry=0;
    short i;
```

```
    resVal->len=val1->len;  /* set up length of result */
    /* start at least significant bit and add bytes    */
    for (i=val1->len-1;i>=0;i--)
    {
        unsigned char v1=val1->val[i];  /* i'th byte of val1 */
        unsigned char v2=val2->val[i];  /* i'th byte of val2 */
        /* add the two bytes plus carry from previous addition */
        unsigned char res=v1+v2+carry;
        /* set up carry for next byte */
        if (v1+v2+carry>255)
            carry=1;                    /* carry into next byte */
        else
            carry=0;
        resVal->val[i]=res;             /* set the i'th byte of the result */
    }
}

/* read from XSVF numBytes bytes of data into x */
void readVal(lenVal *x, short numBytes)
{
    int i;
    for (i=0;i<numBytes;i++)
        readByte(&(x->val[i]));  /* read a byte of data into the lenVal */
    x->len=numBytes;             /* set the length of the lenVal        */
}
```

```
/*****************************************************/
/* file: lenval.h                                    */
/* abstract:  This file contains a description of the  */
/*            data structure "lenval".                */
/*****************************************************/

#ifndef lenval_dot_h
#define lenval_dot_h

/* the lenVal structure is a byte oriented type used to store an */
/* arbitrary length binary value. As an example, the hex value   */
/* 0x0e3d is represented as a lenVal with len=2 (since 2 bytes   */
/* and val[0]=0e and val[1]=3d.  val[2-MAX_LEN] are undefined    */

/* maximum length (in bytes) of value to read in        */
/* this needs to be at least 4, and longer than the     */
/* length of the longest SDR instruction.  If there is, */
/* only 1 device in the chain, MAX_LEN must be at least */
/* ceil(27/8) == 4.  For 6 devices in a chain, MAX_LEN  */
/* must be 5, for 14 devices MAX_LEN must be 6, for 20  */
/* devices MAX_LEN must be 7, etc..                     */
/* You can safely set MAX_LEN to a smaller number if you*/
/* know how many devices will be in your chain.         */
#define MAX_LEN 4


typedef struct var_len_byte
{
    short len;   /* number of chars in this value */
    unsigned char val[MAX_LEN+1];  /* bytes of data */
} lenVal;


/* return the long representation of a lenVal */
extern long value(lenVal *x);

/* set lenVal equal to value */
extern void initLenVal(lenVal *x, long value);

/* check if expected equals actual (taking the mask into account) */
extern short EqualLenVal(lenVal *expected, lenVal *actual, lenVal *mask);

/* add val1+val2 and put the result in resVal */
extern void addVal(lenVal *resVal, lenVal *val1, lenVal *val2);

/* return the (byte, bit) of lv (reading from left to right) */
extern short RetBit(lenVal *lv, int byte, int bit);

/* set the (byte, bit) of lv equal to val (e.g. SetBit("00000000",byte, 1)
   equals "01000000" */
extern void SetBit(lenVal *lv, int byte, int bit, short val);

/* read from XSVF numBytes bytes of data into x */
extern void  readVal(lenVal *x, short numBytes);

#endif
```

```
/*****************************************************************/
/* file: micro.c                                                 */
/* abstract:  This file contains the main function               */
/*            call for reading in a file from a prom             */
/*            and pumping the JTAG ports.                        */
/*                                                               */
/* Notes: There is a compiler switch called DEBUG_MODE.          */
/*        If DEBUG_MODE is defined, the compiler will read       */
/*        the xsvf file from a file called "prom.bit".          */
/*        It will also enable debugging of the code              */
/*        by printing the TDI and TMS values on the              */
/*        rising edge of TCLK.                                   */
/*****************************************************************/

#include "lenval.h"
#include "ports.h"


#define CLOCK_RATE 150    /* set to be the clock rate of the system in kHz */

/* encodings of xsvf instructions */

#define XCOMPLETE        0
#define XTDOMASK         1
#define XSIR             2
#define XSDR             3
#define XRUNTEST         4
#define XREPEAT          7
#define XSDRSIZE         8
#define XSDRTDO          9
#define XSETSDRMASKS     10
#define XSDRINC          11

/* return number of bytes necessary for "num" bits */
#define BYTES(num) \
        (((num%8)==0) ? (num/8) : (num/8+1))

extern void doSDRMasking(lenVal *dataVal, lenVal *nextData,
            lenVal *addressMask, lenVal *dataMask);
extern short loadSDR(int numBits, lenVal *dataVal, lenVal *TDOExpected,
             lenVal *TDOMask);
extern void clockOutLenVal(lenVal *lv,long numBits,lenVal *tdoStore);
extern void gotoIdle();

lenVal TDOMask;     /* last TDOMask received */
lenVal maxRepeat;   /* max times tdo can fail before ISP considered failed */
lenVal runTestTimes; /* value of last XRUNTEST instruction */


#ifdef DEBUG_MODE
#include <stdio.h>
FILE *in;           /* for debugging */
#endif
```

```
/* clock out the bit onto a particular port */
void clockOutBit(short p, short val)
{
    setPort(p,val);  /* change the value of TMS or TDI */
    pulseClock();    /* set TCK to Low->High->Low      */
}

/* clock out numBits from a lenVal;  the least significant bits are */
/* output on the TDI line first; exit into the exit(DR/IR) state.   */
/* if tdoStore!=0, store the TDO bits clocked out into tdoStore.    */
void clockOutLenVal(lenVal *lv,long numBits,lenVal *tdoStore)
{
    int i;
    short j,k;

    /* if tdoStore is not null set it up to store the tdoValue */
    if (tdoStore)
        tdoStore->len=lv->len;

    for (i=0;i<lv->len;i++)
    {
        /* nextByte contains the next byte of lenVal to be shifted out */
        /* into the TDI port                                           */
        unsigned char nextByte=lv->val[lv->len-i-1];
        unsigned char nextReadByte=0;
        unsigned char tdoBit;
        /* on the last bit, set TMS to 1 so that we go to the EXIT DR */
        /* or to the EXIT IR state */
        for (j=0;j<8;j++)
        {
            /* send in 1 byte at a time */
            /* on last bit, exit SHIFT SDR */
            if (numBits==1)
              setPort(TMS,1);

            if (numBits>0)
            {
              tdoBit=readTDOBit();  /* read the TDO port into tdoBit */
              clockOutBit(TDI,nextByte & 0x1);  /* set TDI to last bit */
              nextByte=nextByte>>1;
              numBits--;
              /* first tdoBit of the byte goes to 0x00000001   */
              /* second tdoBit goes to 0x00000010, etc.        */
              /* Shift the TDO bit to the right location below */
              for (k=0;k<j;k++)
                 tdoBit=tdoBit<<1;

              /* store the TDO value in the nextReadByte */
              nextReadByte|=tdoBit;
            }
        }
        /* if storing the TDO value, store it in the correct place */
        if (tdoStore)
            tdoStore->val[tdoStore->len-i-1]=nextReadByte;
    }
}
```

```
/* parse the xsvf file and pump the bits */
int main()
{
    lenVal inst; /* instruction */
    lenVal bitLengths; /* hold the length of the arguments to read in */
    lenVal dataVal,TDOExpected;
    lenVal SDRSize,addressMask,dataMask;
    lenVal sdrInstructs;
    long i;

#ifdef DEBUG_MODE
    /* read from the file "prom.bit" instead of a real prom */
    in=fopen("prom.bit","rb");
#endif
    gotoIdle();
    while (1)
    {
        readVal(&inst,1);  /* read 1 byte for the instruction */
        switch (value(&inst))
        {
        case XTDOMASK:
            /* read in new TDOMask */
            readVal(&TDOMask,BYTES(value(&SDRSize)));
            break;
        case XREPEAT:
            /* read in the new XREPEAT value */
            readVal(&maxRepeat,1);
            break;
        case XRUNTEST:
            /* read in the new RUNTEST value */
            readVal(&runTestTimes,4);
            break;
        case XSIR:
            /* load a value into the instruction register */
            clockOutBit(TMS,1);  /* Select-DR-Scan state  */
            clockOutBit(TMS,1);  /* Select-IR-Scan state  */
            clockOutBit(TMS,0);  /* Capture-IR state      */
            clockOutBit(TMS,0);  /* Shift-IR state        */
            readVal(&bitLengths,1); /* get number of bits to shift in */
            /* store instruction to shift in */
            readVal(&dataVal,BYTES(value(&bitLengths)));
            /* send the instruction through the TDI port and end up   */
            /* dumped in the Exit-IR state                            */
            clockOutLenVal(&dataVal,value(&bitLengths),0);
            clockOutBit(TMS,1);  /* Update-IR state       */
            clockOutBit(TMS,0);  /* Run-Test/Idle state   */
            break;
        case XSDRTDO:
            /* get the data value to be shifted in */
            readVal(&dataVal,BYTES(value(&SDRSize)));
            /* store the TDOExpected value    */
            readVal(&TDOExpected,BYTES(value(&SDRSize)));
            /* shift in the data value and verify the TDO value against */
            /* the expected value                                       */
            if (!loadSDR(value(&SDRSize), &dataVal, &TDOExpected, &TDOMask))
            {
```

```
          /* The ISP operation TDOs failed to match expected */
          return 0;
            }
            break;
        case XSDR:
            readVal(&dataVal,BYTES(value(&SDRSize)));
            /* use TDOExpected from last XSDRTDO instruction */
            if (!loadSDR(value(&SDRSize), &dataVal, &TDOExpected, &TDOMask))
              return 0;  /* TDOs failed to match expected */
            break;
        case XSDRINC:
            readVal(&dataVal,BYTES(value(&SDRSize)));
            if (!loadSDR(value(&SDRSize), &dataVal, &TDOExpected, &TDOMask))
              return 0;  /* TDOs failed to match expected */
            readVal(&sdrInstructs,1);
            for (i=0;i<value(&sdrInstructs);i++)
            {
              lenVal nextData;
              int dataLength=8; /* fix to be number of 1's in dataMask */
              readVal(&nextData,BYTES(dataLength));
              doSDRMasking(&dataVal, &nextData, &addressMask, &dataMask);
              if (!loadSDR(value(&SDRSize), &dataVal,
                        &TDOExpected, &TDOMask))
                 return 0;  /* TDOs failed to match expected */
            }
            break;
        case XSETSDRMASKS:
            /* read the addressMask */
            readVal(&addressMask,BYTES(value(&SDRSize)));
            /* read the dataMask    */
            readVal(&dataMask,BYTES(value(&SDRSize)));
            break;
        case XCOMPLETE:
            /* return from subroutine */
            return 1;
            break;
         case XSDRSIZE:
            /* set the SDRSize value */
            readVal(&SDRSize,4);
            break;

        }
    }
}



/* determine the next data value from the XSDRINC instruction and store     */
/* it in dataVal.                                                           */
/* Example:  dataVal=0x01ff, nextData=0xab, addressMask=0x0100,             */
/*           dataMask=0x00ff, should set dataVal to 0x02ab                  */
void doSDRMasking(lenVal *dataVal, lenVal *nextData, lenVal *addressMask,
            lenVal *dataMask)
{
    int i,j,count=0;
    /* add the address Mask to dataVal and return as a new dataVal */
```

```
    addVal(dataVal, dataVal, addressMask);
    for (i=0;i<dataMask->len;i++)
    {
        /* look through each bit of the dataMask. If the bit is     */
        /* 1, then it is data and we must replace the corresponding*/
        /* bit of dataVal with the appropriate bit of nextData      */
        for (j=0;j<8;j++)
            if (RetBit(dataMask,i,j))  /* this bit is data */
            {
              /* replace the bit of dataVal with a bit from nextData */
              SetBit(dataVal,i,j,RetBit(nextData,count/8,count%8));
              count++;  /* count how many bits have been replaced */
            }
    }
}

/* goto the idle state by setting TMS to 1 for 5 clocks, followed by TMS */
/* equal to 0 */
void gotoIdle()
{
    int i;
    setPort(TMS,1);
    for (i=0;i<5;i++)
       pulseClock();
    setPort(TMS,0);
    pulseClock();
}


/* return 0 iff the TDO doesn't match what is expected */
short loadSDR(int numBits, lenVal *dataVal, lenVal *TDOExpected,
              lenVal *TDOMask)
{
    int failTimes=0;
    while (1)
    {
        lenVal actualTDO;
        int repeat;  /* to do RUNTESTS */
        clockOutBit(TMS,1);  /* Select-DR-Scan state  */
        clockOutBit(TMS,0);  /* Capture-DR state      */
        clockOutBit(TMS,0);  /* Shift-DR state        */
        /* output dataVal onto the TDI ports; store the TDO value returned */
        clockOutLenVal(dataVal,numBits,&actualTDO);
        /* compare the TDO value against the expected TDO value */
        if (EqualLenVal(TDOExpected,&actualTDO,TDOMask))
        {
            /* TDO matched what was expected */
            clockOutBit(TMS,1);  /* Update-DR state    */
            clockOutBit(TMS,0);  /* Run-Test/Idle state*/

            /* wait in Run-Test/Idle state */
            waitTime(value(&runTestTimes));
            break;
        }
        else
        {
```

```
            /* TDO did not match the value expected */
            failTimes++;        /* update failure count */
            if (failTimes>value(&maxRepeat))
                return   0;   /* ISP failed */
            clockOutBit(TMS,0); /* Pause-DR state      */
            clockOutBit(TMS,1); /* Exit2-DR state      */
            clockOutBit(TMS,0); /* Shift-DR state      */
            clockOutBit(TMS,1); /* Exit1-DR state      */
            clockOutBit(TMS,1); /* Update-DR state     */
            clockOutBit(TMS,0); /* Run-Test/Idle state */
            /* wait in Run-Test/Idle state */
            waitTime(value(&runTestTimes));
        }
    }
    return 1;
}
```

```
/*****************************************************/
/* file: ports.c                                     */
/* abstract:  This file contains the routines to     */
/*            output values on the JTAG ports, to read */
/*            the TDO bit, and to read a byte of data  */
/*            from the prom                          */
/*                                                   */
/* Notes: See the notes for micro.c for explanation of */
/*        the compiler switch  "DEBUG_MODE".         */
/*****************************************************/
#include "ports.h"

#ifdef DEBUG_MODE
#include "stdio.h"
extern FILE *in;
#endif

#ifdef DEBUG_MODE
/* if in debugging mode, use variables instead of setting the ports */
short pTCK,pTMS,pTDI;
#endif


/* if in debugging mode, then just set the variables */
void setPort(short p,short val)
{
#ifdef DEBUG_MODE
    if (p==TCK)
       pTCK=val;
    if (p==TMS)
       pTMS=val;
    if (p==TDI)
       pTDI=val;
#endif
}

#ifdef DEBUG_MODE
void printPorts()
{
    printf("%d   %d\n",pTMS,pTDI);
}
#endif

/* toggle tck LHL */
void pulseClock()
{
    setPort(TCK,0);  /* set the TCK port to low  */
    setPort(TCK,1);  /* set the TCK port to high */
#ifdef DEBUG_MODE
    /* if in debugging mode, print the ports on the rising clock edge */
    printPorts();
#endif;
    setPort(TCK,0);  /* set the TCK port to low  */

}
```

```
/* read in a byte of data from the prom */
void readByte(unsigned char *data)
{
#ifdef DEBUG_MODE
    /* pretend reading using a file */
    fscanf(in,"%c",data);
#endif
}

/* read the TDO bit from port */
unsigned char readTDOBit()
{
#ifdef DEBUG_MODE
    return 1;  /* garbage value for now; replace with real port read. */
#endif
}


/* Wait at least the specified number of microsec.                          */
/* Use a timer if possible; otherwise estimate the number of instructions   */
/* necessary to be run based on the microcontroller speed.  For this example */
/* we pulse the TCK port a number of times based on the processor speed.    */
void waitTime(long microsec)
{
    int repeat;
#define CLOCK_RATE 150 /* set to be the clock rate of the system in kHz */
    long clockRunTests=microsec*CLOCK_RATE/1000;
    for (repeat=0;repeat<clockRunTests;repeat++)
                pulseClock();
}
```

```
/*****************************************************/
/* file: ports.h                                     */
/* abstract:  This file contains extern declarations */
/*            for providing stimulus to the JTAG ports.*/
/*****************************************************/

#ifndef ports_dot_h
#define ports_dot_h

#if 0
#define DEBUG_MODE  /* this line can be enabled in order to read in the xsvf from */
             /* a file called "prom.bits" and output the                    */
             /* TMS and TDI values on the rising edge of                    */
             /* the clock                                                   */
#endif

/* these constants are used to send the appropriate ports to setPort */
/* they should be enumerated types, but some of the microcontroller  */
/* compilers don't like enumerated types */
#define TCK 0
#define TMS 1
#define TDI 2

/* set the port "p" (TCK, TMS, or TDI) to val (0 or 1) */
extern void setPort(short p, short val);

/* read the TDO bit and store it in val */
extern unsigned char readTDOBit();

/* make clock go down->up->down*/
extern void pulseClock();

/* read the next byte of data from the xsvf file */
extern void readByte(unsigned char *data);

#endif
```

Appendix D

Binary to Intel Hex Translator

This appendix contains C-code that can be used to convert XSVF files to Intel Hex format for downloading to an EPROM programmer. Most embedded processor code development systems can output Intel Hex for included binary files, and for those systems the following code is not needed. However, designers can use the following C-code if the development system they are using does not have Intel Hex output capability.

Overview

The ISP controller described in this application note allows designers to program and test XC9500 CPLDs from information stored in EPROM. This information is saved in a binary XSVF file that contains both device programming instructions as well as the device configuration data. The 8051 microcontroller reads the EPROM (or EPROMs) that contain the XSVF file, converts the binary information to XC9500 compatible instructions and data, and outputs the programming information to the XC9500 device through a 4-wire test access port.

After an XC9500 design has been converted to XSVF format, the XSVF information is converted to Intel hex format which is downloaded to an EPROM programmer. The resulting EPROMs, containing the CPLD programming information, can then be used in this ISP controller design.


```
/*
        This program is released to the public domain. It
prints a file to stdout in Intel HEX 83 format.
*/

#include <stdio.h>

#define RECORD_SIZE0x10/* Size of a record. */
#define BUFFER_SIZE 128

/*** Local Global Variables ***/

static char *line, buffer[BUFFER_SIZE];
static FILE *infile;

/*** Extern Functions Declarations ***/

extern char hex( int c );
extern void puthex( int val, int digits );

/*** Program Main ***/

main( int argc, char *argv[] )
{
        int c=1, address=0;
        int sum, i;
        i=0;
        /*** First argument - Binary input file ***/

        if (!(infile = fopen(argv[++i],"rb"))) {
          fprintf(stderr, "Error on open of file %s\n",argv[i]);
          exit(1);
        }

        /*** Read the file character by character ***/
```

```
        while (c != EOF) {
          sum = 0;
          line = buffer;
          for (i=0; i<RECORD_SIZE && (c=getc(infile)) != EOF; i++) {
            *line++ = hex(c>>4);
            *line++ = hex(c);
            sum += c; /* Checksum each character. */
          }
          if (i) {
            sum += address >> 8;/* Checksum high address byte.*/
            sum += address & 0xff;/* Checksum low address byte.*/
            sum += i;         /* Checksum record byte count.*/
            line = buffer;     /* Now output the line! */
            putchar(':');
            puthex(i,2);     /* Byte count. */
            puthex(address,4);     /* Do address and increment */
            address += i;     /*    by bytes in record.    */
            puthex(0,2);     /* Record type.    */
            for(i*=2;i;i--)     /* Then the actual data.    */
              putchar(*line++);
            puthex(0-sum,2);     /* Checksum is 1 byte 2's comp.*/
            printf("\n");
          }
        }
        printf(":00000001FF\n");/* End record. */
}



/* Return ASCII hex character for binary value. */

char
hex( int c )
{
        if((c &= 0x000f)<10)
          c += '0';
        else
          c += 'A'-10;
        return((char) c);
}

/* Put specified number of digits in ASCII hex. */

void
puthex( int val, int digits )
{
        if (--digits)
          puthex(val>>4,digits);
        putchar(hex(val & 0x0f));
}
```

Using Automatic Test Equipment to Program XC9500 Devices In-System

January, 1997 (Version 1.0) Application Note

Summary

This application note describes how to program XC9500 devices in-system, using standard automatic test equipment.

Xilinx Family

XC9500

1

Introduction

XC9500 devices use a standard 4-wire Test Access Port (TAP) for both In-System Programming (ISP) and IEEE 1149.1 boundary scan (JTAG) testing. Therefore, manufacturers can reduce their overall system cost and reduce device damage due to unnecessary handling by using automatic test equipment (ATE) to both program and test these devices. The XC9500 Boundary-scan architecture is shown in Figure 1.

The Xilinx EZTag™ software helps make this possible by automatically generating a Serial Vector Format (SVF) file describing the programming and test algorithms required by the XC9500 devices. Most ATE platforms accept SVF as a test vector input format. This application note describes the steps required to generate an SVF file and how the ATE uses this file to program and test a device.

SVF Overview

The original Serial Vector Format was developed jointly by Texas Instruments and Teradyne in response to a need for the exchange of boundary-scan test vectors between such tools as test generation software and ATE. At that time, usage of the IEEE standard 1149.1 was increasing but no common format or language existed to satisfy the need for a common data exchange.

The developers of SVF chose a format that did not use test vectors solely to provide TCK (clock) and TMS (mode control) signals to the IEEE 1149.1 TAP. Instead, the underlying models of the SVF format assume that all operations begin and end in stable states. This results in a much simpler and more concise description of the stimulus vectors.

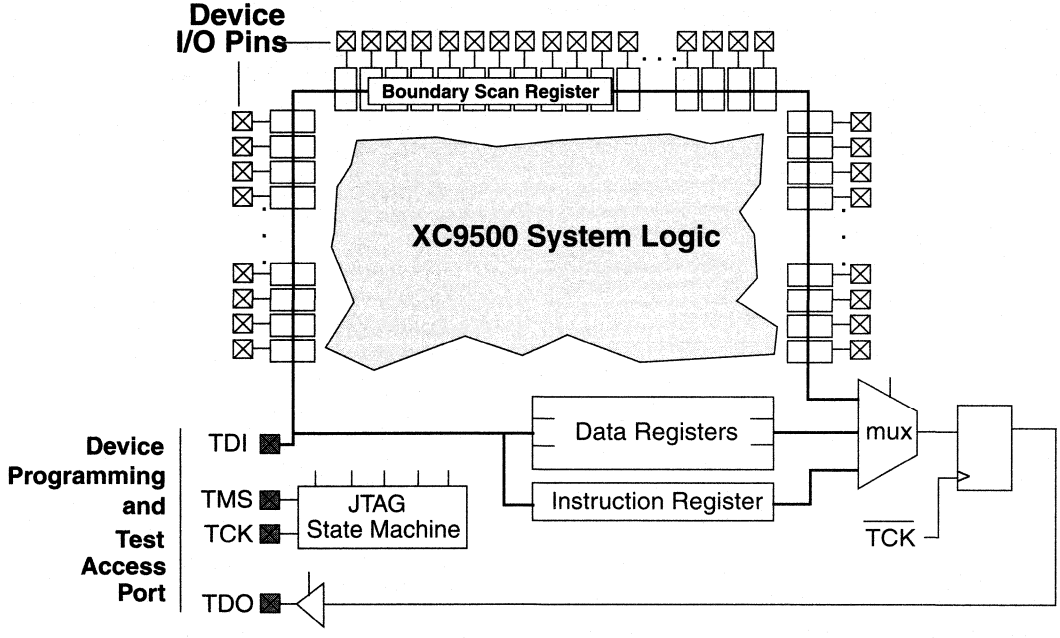


Figure 1: XC9500 Boundary Scan Architecture

Between mid-1991 and the autumn of 1994 three revisions of SVF were developed, with the goal of creating a format that was independent of the test application vehicle. By late 1994 over 100 companies had developed SVF-based tools and at least ten vendors of CAE tools and ATE were supporting SVF.

SVF has proven itself to be a useful and reliable format for exchanging data between ATE and the software that drives it.

SVF Specification

For the purposes of XC9500 ISP, only three records of the thirteen SVF records that describe the standard are needed. Those three records are discussed in this section.

An SVF file contains a set of ASCII statements. The maximum number of characters allowed on a line is 256, however one SVF statement can span more than one line. Each statement consists of a command and its associated parameters, terminated by a semicolon. SVF is not case sensitive and comments are indicated by an exclamation point (!) or a pair of slashes (//) at the beginning of a line, terminated by a carriage return.

Scan data within a statement is expressed in hexadecimal and is always enclosed in parenthesis. The scan data cannot specify a data string that is larger than the specified bit length; the Most Significant Bit (MSB) zeros in the hex string are not considered when determining the string length. The bit order for scan data defines the LSB (rightmost bit) as the first bit scanned into the device for scan data specified by the TDI and SMASK keywords, and is the first bit scanned out for data specified by the TDO and MASK keywords.

The following SVF Commands are supported by the XC9500 EZTag software:

- SDR (Scan Data Register).
- SIR (Scan Instruction Register).
- RUNTEST.

In each of the following command descriptions the parameters are mandatory. Optional parameters are enclosed in brackets ([ ]). Variables are shown in *italics*. Parenthesis "( )"are used to indicate hexadecimal values.

A scan operation is defined as the execution of an SIR or SDR command and any associated header or trailer commands.

SDR, SIR

SDR *length* [TDI (*tdi*)] [TDO (*TDO*)] [MASK (*msk*)] [SMASK (*smask*)][PIO (*pio*)];

SIR *length* [TDI (*tdi*)] [TDO (*TDO*)] [MASK (*msk*)] [SMASK (*smask*)][PIO (*pio*)];

These commands specify a scan pattern to be applied to the target scan registers. The SDR command (Scan Data Register) specifies a data pattern to be scanned into the target device Data Register. The SIR command (Scan Instruction Register) specifies a data pattern to be scanned into the target device Instruction Register.

Parameters:

length — A 32-bit decimal integer specifying the number of bits to be scanned.

[TDI (*tdi*)] — (optional) This specifies the value to be scanned into the target, expressed as a hex value. If this parameter is not present, the value of TDI to be scanned into the target device will be the TDI value specified in the previous SDR/SIR statement. If a new scan command is specified, which changes the length of the data pattern with respect to a previous scan, the TDI parameter must be specified, otherwise the default TDI pattern is undetermined and is an error.

[TDO (*tdo*)] — (optional) This specifies the test values to be compared against the actual values scanned out of the target device, expressed as a hex string. If this parameter is not present, no comparison will be performed. If no TDO parameter is present, the MASK will not be used.

[MASK (*mask*)] — (optional) This specifies the mask to be used when comparing TDO values against the actual values scanned out of the target device, expressed as a hex string. A "0" in a specific bit position indicates a "don't care" for that position. If this parameter is not present, the mask will equal the previously specified MASK value specified for the SIR/SDR statement. If a new scan command is specified which changes the length of the data pattern with respect to a previous scan, the MASK parameter must be specified, otherwise the default MASK pattern is undefined and is an error. If no TDO parameter is present, the MASK will not be used.

[SMASK (*smask*)] — (optional) This specifies which TDI data is "don't care", expressed as a hex string. A "0" in a specific bit position indicates that the TDI data in that bit position is a "don't care". If this parameter is not present, the mask will equal the previously specified SMASK value specified for the SDR/SIR statement. If a new scan command is specified which changes the length of the data pattern with respect to a previous scan, the SMASK parameter must be specified, otherwise the default SMASK pattern used is undefined and is an error. The SMASK will be used even if the TDI parameter is not present.

Example:

```
SDR 24 TDI (000010) TDO (818181) MASK
(FFFFFF) SMASK (0);

SIR 16 TDO (ABCD);
```

RUNTEST

RUNTEST *run_count* TCK;

This command forces the target IEEE 1149.1 bus to the Run- Test/Idle state for a specific number of TCK clock periods. This can be used to specify latency periods when operating the TAP.

Parameters:

run_count — The number of TCK clock periods that the 1149.1 bus will remain in the Run Test/Idle state, expressed as a 32 bit unsigned number.

Example:
RUNTEST 1000 TCK;

A Sample XVF File is shown as follows:

```
! Begin Test Program
TRST OFF;       !disable test reset line
ENDIR IDLE;     !End IR scan in IDLE
SIR 8 TDI (FE) MASK (FF)
SDR 14 TDI (3afe) MASK (3ff) TDO (0003)
    SMASK (3ff)
RUNTEST 100 TCK
!End test program
```

Using EZTag to generate an SVF file

This procedure shows how to create an SVF file; it assumes that the Xilinx XACT version 6.0.0 software, or newer, which includes the XC9500 fitter and the EZTag software, is being used.

1. Create the design using XABEL-CPLD or any compatible third-party design entry tool.
2. Fit the design and save it to a JEDEC output file.
3. Invoke the EZTag software from the XACT command line using the following command:

   ```
   eztag -svf
   The following message appears:
   Xilinx (R) EZTAG XC9500-CPLD-6.0.0 - JTAG
   Boundary-Scan Download
   Copyright (C) Xilinx Inc. 1991-1995. All
   Rights Reserved.
   ----------------------------------------
   SVF GENERATION MODE.
   EZTAG?
   ```

4. At the **EZTAG?** Prompt type the following command:

   part *deviceType1*:*designName1*
   deviceType2:*designName2*
   deviceTypeN:*designNameN* <CR>

   where ***designName*** is the name of the design to translate into SVF. Multiple *deviceType:designName* pairs are separated by spaces.

   This command defines the JTAG device chain, from one to any number of devices. The parts specified in the **part** command should be arranged in order beginning with the first device to receive TDI and ending with the last device to output TDO.

   Note: For any non-XC9500 part in the JTAG chain make certain that the BSDL file for the specified part is available along the XACT path and is called *deviceType*.bsd (e.g., 4003pc84.bsd for a XC4003 in the PC84 package).

5. Enter any one of the following commands:

   erase *designName* — generates an SVF file that specifies the bit sequence to erase the specified part.

   verify *designName* [-j *jedecFileName*] — generates an SVF file that specifies the bit sequence to read back the device contents and compares it against the contents of the specified JEDEC file.

   program [-b] *designName* [-j *jedecFileName*] — generates an SVF file that specifies the bit sequence to program the specified part from a JEDEC file named *designName*.jed (or alternately, the JEDEC file name specified after the "-j"). The program command options add the following functionality:

   -b — When using new devices shipped from the factory, this switch skips the erasure process that usually precedes programming. Erasure is not necessary for a device that has not been previously programmed.

6. Exit EZTag by entering the following command:

   quit

   NOTE: The SVF file will be named *designName*.svf, and will be created in the current working directory (the directory in which EZTAG is being run). Consecutive operations on the same *designName* file will overwrite the SVF file each time. The SVF file contains all data and commands necessary to perform the specified function.

SVF Interpretation

The simplicity of SVF is also one of its major weaknesses. Much of the behavior of SVF, while running, is left unspecified by the standard. In order to optimize SVF stimulus for an application, the interpretations of some operations must be defined more precisely.

RUNTEST TCK

Many ATE manufacturers prefer not to generate bursts of TCK activity because this results in significantly increased test vector file sizes. This increases the overall test cost and can cause the vector set to run inefficiently. Because the RUNTEST record is used to wait for something to happen, most ATE manufacturers interpret the TCK burst specified as a time value. The favored interpretation is that the

number represents a wait time in microseconds. This is how the EZTag-generated SVF files should be interpreted.

SDR predicted TDO values

The SVF specification describes a method for specifying predicted TDO values. It does not, however, specify actions to be taken when the predicted TDO value does not equal the expected values.

When using Xilinx XC9500 parts, the TDO values predicted reflect the status of the just completed operation (which could be an erase or a program operation). If the status is not the success status (which is the value predicted as the TDO value in the generated SVF file) then the following 1149.1 TAP controller state transition sequence should be followed (assuming the TDO validation failure is detected in the EXIT1-DR state):

1. EXIT1-DR
2. PAUSE-DR
3. EXIT2-DR
4. SHIFT-DR
5. EXIT1-DR
6. UPDATE-DR
7. RUN-TEST/IDLE

The above state transition sequence is illustrated in the 1149.1 TAP state diagram in Figure 2.

The net effect of the state transition sequence is to nullify the just-shifted-in programming or erase data and re-apply the previous program or erase data. Note that the ATE application interpreting the SVF must acknowledge this by not advancing beyond the current SVF record.

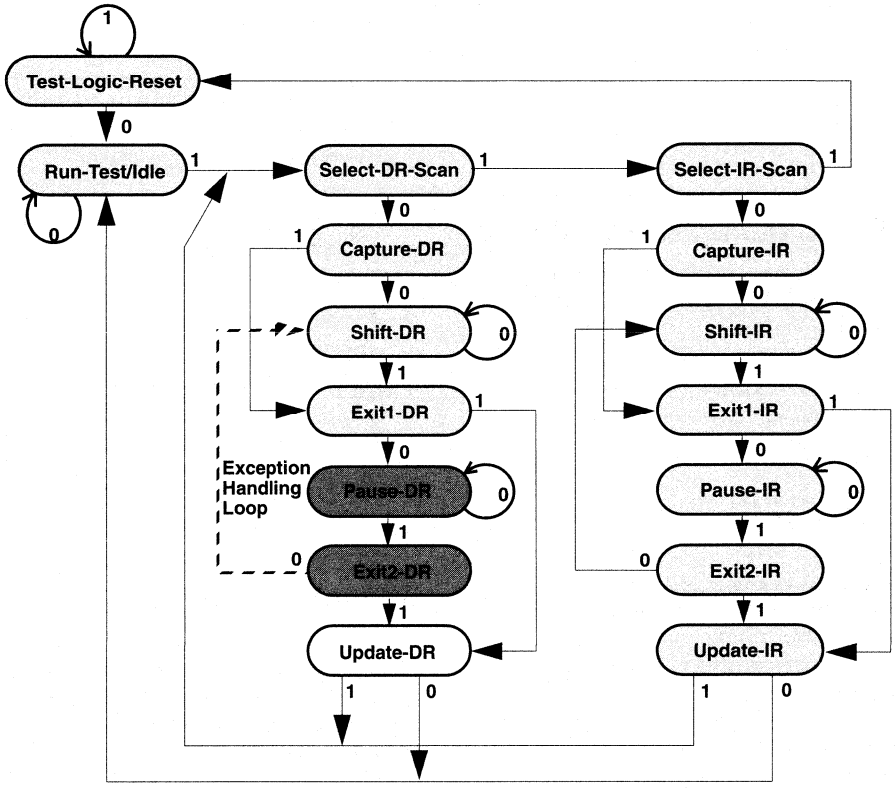


Figure 2: Test Access Port State Diagram

```
// First SDR record
SDR 27 TDI (000003fe) SMASK (07ffffff); // Just apply the value - no test for
TDO
RUNTEST 160000 TCK; // Wait for 160 msec.
// Second SDR record
SDR 27 TDI (008003fe) SMASK (07ffffff) TDO (00000003) MASK (00000003);
// Apply value to TDI read TDO test for concurrence
// if not as expected do "failure recovery loop" - hold
// at this SDR instruction.
RUNTEST 160000 TCK; // Wait for 160 msec
// Third SDR record
SDR 27 TDI (010003fe) SMASK (07ffffff) TDO (00000003) MASK (00000003);
RUNTEST 160000 TCK; // Wait for 160 msec
// Fourth SDR record
SDR 27 TDI (018003fe) SMASK (07ffffff) TDO (00000003) MASK (00000003);
RUNTEST 160000 TCK; // Wait for 160 msec
```

Figure 3: SVF File Fragment Illustrating ATE Flow

Using the SVF file as an example, as shown in Figure 3, the required ATE operation should then be as follows:

1. When reading an SDR instruction with a TDO specified (like the second one in Figure 3), the predicted TDO value must match the value output from the device on the tester. If it does not match, then the failure recovery loop is executed. In the RUN-TEST/IDLE state a pause is inserted for the amount of time specified in the previously applied RUNTEST instruction.
2. On exit from the RUNTEST instruction, re-apply that same SDR record (in this case the second one in the file) and test the TDO value again.
3. If the TDO matches the expected value, the TAP state machine is transitioned back to RUN-TEST/IDLE the normal way (via EXIT1-DR and UPDATE-DR) and is applied to the next SDR record.
4. This "recovery loop" is to be attempted no more than 32 times. If the TDO value does not match after 32 times, the part is considered defective and the process should abort with some failure indication supplied to the user.

Normally, less than 1% of the addresses fail the TDO check and require the additional erase or program time associated with execution of the failure recovery loop.

Pseudo-code ATE Algorithm

The following pseudo-code describes the sequence of operations that should be used in interpreting the SVF file on a generic ATE.

1. Go to Test-Logic-Reset state
2. Go to Run-Test Idle state
3. Read SVF record
4. if SIR record then
   go to Shift-IR state
   Scan in <TDI value>
5. else if SDR record then
   set <repeat count> to 0
   store <TDI value> as <current TDI value>
   store <TDO value> as <current TDO value>
6. go to Shift-DR state
   scan in <current TDI value>
   if <current TDO value> is specified then
      if <current TDO value> does not equal <actual TDO value> then
         if <repeat count> > 32 then
            LOG ERROR
            go to Run-Test Idle state
            go to Step 3
         end if
         go to Pause-DR
         go to Exit2-DR
         go to Shift-DR
         go to Exit1-DR
         go to Update-DR
         go to Run-Test/Idle
         increment <repeat count> by 1
         pause <current pause time> microseconds
         go to Step 6)
      end if
      else
      go to Run-Test Idle state
      go to Step 3
   endif
7. else if RUNTEST record then
   pause tester for <TCK value> microseconds
   store <TCK value> as <current pause time>
   end if

Predicted ATE Programming Time

Two components make up the overall programming time of the XC9500 part on ATE. The first component is the number of TAP vectors required to perform the specified operations. This is the number of TCK pulses that must be applied to the TAP to execute the specified operations (with TMS and TDI set appropriately). In the SVF file, this is described by the SDR and SIR records. The second component is the latency time associated with program and erase operations. During this period of time, TCK does not need to be pulsed (although it could be). Rather, the tester should simply pause. In the SVF file, this is described by the RUNTEST TCK record.

To determine the overall performance, the speed at which the ATE can stream the TAP stimulus vectors to the part must be considered. The equation below describes the typical programming time for a single XC9500 part.

Definition of Terms:

- FB = the number of function blocks = (the number of macrocells) / 18.
- Tvec = rate of application of ISP vectors = sec/vector, typically @ 10MHz = 0.1 usec/vector.
- Terase = flash erase time for each sector, typically 160 milliseconds.
- Tpgm = flash program time for each address, typically 160 usec.

Note: XC95144 and larger parts have auto-increment mode which significantly reduces number of ATE vectors that must be applied.

Operations:

Parts arriving from the factory are already erased. For these parts, the erase step may be skipped and therefore programming is described by the following equation:

1. Program Only

```
(2880*(FB)**2 + 34561*FB + 60)*Tvec +
(90*(FB)**2 + 1080*(FB))*Tpgm
```

2. Program (auto-inc.) Only

```
(1350*(FB)**2 + 16201*FB + 91)*Tvec +
(90*(FB)**2 + 1080*(FB))*Tpgm
```

Parts being re-programmed must first be erased and then programmed. The EZTag software defaults to using this approach:

3. Erase/Program

```
(2880*(FB)**2 + 34625*(FB) + 74)*Tvec +
(2*FB)*Terase + (90*(FB)**2 +
1080*(FB))*Tpgm
```

4. Erase/Program (auto-inc.)

```
(1350*(FB)**2 + 16265*(FB) + 105)*Tvec +
(2*FB)*Terase + (90*(FB)**2 +
1080*(FB))*Tpgm
```

Conclusion

By using the EZTag-generated SVF files it is possible to streamline manufacturing flows by programming XC9500 parts on automatic test equipment. This allows integration of the program and test steps of the system manufacturing process. This integration will result in higher system yields, better manufacturability, and simpler part inventory management.

References

Serial Vector Format Specification, Rev C., Texas Instruments.

The Boundary-Scan Handbook, Kenneth Parker, Klewer Academic Publishers, 1994.

IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-1990 (including IEEE Std 1149.1a-1993)

In-System Programming Times

January, 1997 (Version 1.0) Application Note

Summary

This application note discusses the in-system programming speed of the XC9500 devices.

Xilinx Family

XC9500

Introduction

XC9500 devices receive programming vectors and instructions via the JTAG Test Access Port. During programming, the address and data information is shifted in first and then a "burn-in" time is initiated to imprint the programming data into the selected flash cells. This is repeated for all flash memory addresses within the device.

Therefore, the time required to program an XC9500 device includes two components: the information download time and the flash memory burn-in time. In the typical application, where the JTAG clock operates from 1MHz to 10 MHz, the flash memory burn-in exceeds 90% of the total programming time.

Device Programming Times

Programming time varies greatly, depending on the programming environment.

Programming in a Production Environment

In a production environment, fast programming times translate to reduced costs, and most Automatic Test Equipment (ATE) used for board testing is capable of efficiently downloading information to the XC9500 devices at the maximum speed of 10MHz. In order to minimize the production programming costs, XC9500 devices are fully erased and ready for programming when shipped from the factory.

The device programming times for a production environment are shown in Table 1.

Programming in a Development Environment

The device programming times in a typical development environment are usually much longer than those for the production environment for several reasons; there is overhead time spent in the real-time generation of programming vectors from the JEDEC bitmap, bandwidth limitations for outputting JTAG vectors using a general purpose computer, and the time required to erase the device. Together, this overhead can cause a 10 to 50 times increase over the programming times shown in Table 1.

The development environment programming times vary depending on whether the download cable is parallel or serial and they depend on the configuration and type of base computer system.

Programming Multiple Devices Concurrently

Multiple devices in a JTAG chain can be programmed concurrently by downloading all devices with the programming information and then concurrently burning-in all devices in the chain. Therefore, because the download time is small, the total time required to program all devices in a chain is only a little longer than the time required to program the largest device in the chain.

Conclusion

XC9500 devices can be programmed very quickly in a production environment.

Table 1: Typical Device Programming Times

XC9500 Device Type	Programming Time Using ATE at 10MHz		
	1 Device	10 Devices	20 Devices
XC9536	4.8 seconds	5.0 seconds	5.3 seconds
XC9572	5.4 seconds	6.0 seconds	6.6 seconds
XC95108	4.8 seconds	5.7 seconds	6.6 seconds
XC95144	6.9 seconds	8.1 seconds	9.3 seconds
XC95180	9.6 seconds	11.4 seconds	13.2 seconds
XC95216	12.6 seconds	14.6 seconds	16.6 seconds
XC95288	18.3 seconds	20.8 seconds	23.3 seconds

XILINX®

Application Notes

1 ISP and JTAG Support

2 Application Notes

3 XC9500 Data Sheets

4 XC7300 Data Sheets

5 Device Packaging

6 Quality Assurance

7 Technical Support

8 Sales Offices, Representatives, Distributors

Visit us on the WEB at:

www.xilinx.com

for the latest application notes.

XC9500 Pin-Locking Capability and Benchmarks

January, 1997 (Version 1.3) Application Note

Summary

This application note presents benchmarks that demonstrate the superior pin-locking capability of the Xilinx XC9500 CPLDs. These benchmarks are based on typical applications and demonstrate the benefits of a highly routable switch matrix and wide function block fan-in when iterating pin-locked designs. The Xilinx results are compared to other vendors' CPLDs using their production fitters, proving that the Xilinx XC9500 family is the industry's best pin-locking CPLD.

Xilinx Family

XC9500

Introduction

The Xilinx XC9500 CPLD family provides the most advanced, most reliable pin-locking capability in the industry. This important feature allows designers to maintain pinouts after making design changes, eliminating costly, time consuming PC board re-work. CPLDs that do not have adequate pin-locking capability usually require new pinouts even after minor design changes, leaving no room for error and no possibility for field upgrades or field customization. Now, with the XC9500 family, designers can save time and money because they no longer need to modify PC boards every time they make a design change. In addition, this reliable pin-locking capability allows designers to use the in-system programmability features of the XC9500 family to upgrade or modify systems in the field.

This application note demonstrates the advanced pin-locking features of the XC9500 family and provides pin-locking performance comparisons for competing devices.

Pin-Locking Issues

In most CPLDs, each I/O pin is driven directly by a macrocell through an I/O block as shown in Figure 1. When the design is pin-locked, the fitter is forced to map logic into specific macrocells to maintain the pinout. If the device architecture is limited, with inadequate routing in the central switch matrix, the fitter may not be able to place and route the design when the pins are locked.

Some CPLDs use an output routing pool in an attempt to compensate for their primary routing deficiencies. However, output routing pools introduce additional delays and do not prevent the fitter from having to consume logic resources as routing feedthroughs, impacting both design performance and resource utilization.

Logic requirements also affect the ability of the fitter to place and route the design when the pinout is locked. Slow speed designs with simple, narrow logic functions requiring few inputs, feedbacks, and product terms are inherently easier to pinlock than high speed designs with wide fan-in and product term intensive logic functions.

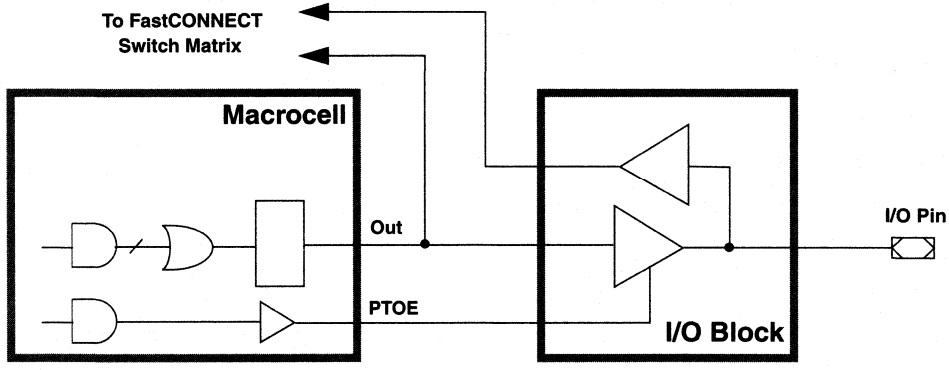


Figure 1: Simplified XC9500 I/O Architecture

The Keys to Reliable Pin-Locking

To address these pin-locking issues, Xilinx XC9500 CPLDs feature abundant routing resources, wide function block fan-in, and flexible product term allocation. The XC9500 fitter also optimizes the initial placement to maximize the design's pin-locking capability. Each of these factors is described as follows.

Routing Resources

Routability is a primary requirement for reliable pin-locking. The routing resources of a CPLD determine how much of the logic block resources (inputs, product terms, and registers) can be used to accommodate design changes after the pins are locked in a design. In a fully routable CPLD, buried logic can be moved without regard to routing restrictions, freeing function block resources that may be needed by the logic that drives the I/O pins.

The XC9500 family provides the most routing resources of any CPLD family currently available. The FastFLASH technology used in the XC9500 family uses smaller cell sizes than other technologies and therefore more routing switches can be packed into the same area. As a result, all devices in the XC9500 family are 100% routable; if there are enough function block resources to implement the design, it will route.

Pin-locking restricts the fitter's capability to place design resources and therefore good routability is crucial. With adequate routability, the constraints imposed by fixed pinouts can be overcome.

Function Block Fan-In Capability

Wide function block fan-in is another important requirement for pin-locking. Since CPLDs are typically used for high speed signal-intensive logic functions, wide function block fan-in is a requirement for implementing functions in a single logic level. The number of available function block inputs affects the fitter's ability to add more signals to any logic that must remain in that function block (because it drives I/O pins). Wide fan-in also helps the fitter implement that logic in a single pass though the device.

Each XC9500 function block has 36 inputs from the switch matrix. Other vendors' in-system programmable CPLDs have as few as 16 inputs.

Product Term Allocation

Product term allocation is important to pin-locking because it allows design changes that increase the product term requirement. All XC9500 devices allocate individual product terms from anywhere in the function block to the macrocell that needs them, accommodating logic changes when the design is pin-locked.

In the XC9500 family, up to 90 product terms can be allocated to any macrocell in the function block. This is in contrast to other vendors' CPLDs that restrict the product term availability (from 5 to 32 pterms) on the basis of macrocell location in the function block.

Fitter Strategy

Fitter software is a key component of any successful CPLD pin-locking solution. It must work in conjunction with the device architecture, spreading the outputs to accommodate design changes when the design is pin-locked.

The XC9500 fitter is optimized to take full advantage of the hardware resources of the XC9500 family. The fitting algorithms that determine how to place and route the design make full use of the abundant routing and product term allocation resources within an XC9500 device to give unparalleled pin-locking performance. The Xilinx fitter is capable of intelligently utilizing all available device resources to retain pinouts and still maintain the required performance, even after significant design changes.

Pin-Locking Benchmarks

The following benchmark data shows the relative pin-locking performance of Xilinx, Altera, Lattice, and AMD CPLDs. These benchmarks are based on typical applications such as address decoders, datapath designs, and address counters, in which reliable pin-locking is crucial. They illustrate the CPLD's capability to accommodate design changes while maintaining an acceptable level of design performance, because not only must the iterated design reroute when the pinout is maintained, it must do so with minimal impact on design performance. Therefore, all of the benchmark data presented in this application note is normalized to the design performance achieved when the fitters are free to choose the pinouts without restrictions.

Synario™ was used for design entry to support retargeting to multiple CPLD vendors using identical ABEL code. The following fitters from the CPLD vendors were used to implement the benchmark designs:

- XABEL-CPLD v6.1 for Xilinx
- pDS+ v2.2 for Lattice
- MAX+2 v6.2 for Altera
- MACH Device Kit v2.3 for AMD

Each design was initially compiled by allowing the fitter to freely choose the pinout. After changes were made to the design, it was re-compiled using the previously assigned pinout. Design performance was measured using t_{PD} and external f_{MAX} as true measures of system performance, where external f_{MAX} is defined as $1/(t_{CO} + t_{SU})$.

Software and Device Availability

Not all of the other vendors' announced devices were supported by their software and therefore not all of their device densities and packages could be evaluated, as indicated in the following charts. Updated benchmarks will be published when available.

Address Decoder Benchmark

This benchmark design, shown in Figure 2 and Figure 3, measures the effect of routing resources and function block fan-in on the CPLD's pin-locking capability. The design contains two 16, 32, or 36 bit buses which are decoded to generate two chip select outputs. A typical design change, involving the correction of a typographic error in which the outputs are decoded incorrectly, is illustrated in Figure 4.

The benchmark results in Figure 11 demonstrate that both the Xilinx XC9500 family and the Altera EPM7000S devices were able to accommodate the design changes without impact on design performance. The Lattice devices maintained the same pinout with a significant (up to 60%) performance penalty. Since the Lattice devices have 16 input logic blocks, the performance degradation of the 16-bit address decoder can be attributed to poor routing resources while the performance of the 32 and 36 bit decoders is degraded by both poor routing and narrow logic block fan-in.

The AMD MACH 5 devices exhibited a 33% performance degradation in the higher pin count packages when the designs were pin-locked. This degradation resulted from segment delays incurred during re-routing (but not incurred during the initial design compilation). Additionally, the MACH 5 software was unable to route the 36-bit wide decoder during the initial compile. This can be attributed to poor fitter performance, inadequate routing resources, or both.

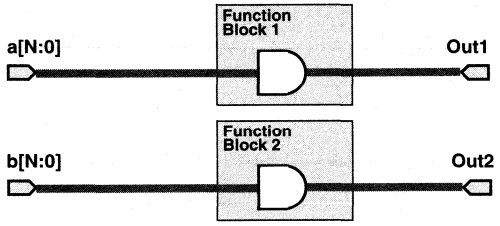


Figure 2: Address Decoder

```
MODULE SWAP
TITLE 'DECODER'
//inputs
a15..a0 pin; "A bus
b15...bo pin; "B bus

//variables
a_bus = [a15..a0];
b_bus = [b15..b0];

//outputs
out1 pin istype 'com';
out2 pin istype 'com';

equations

out1 = a_bus == 24;
out2 = b_bus == 24;

END
```

Figure 3: Address Decoder Code

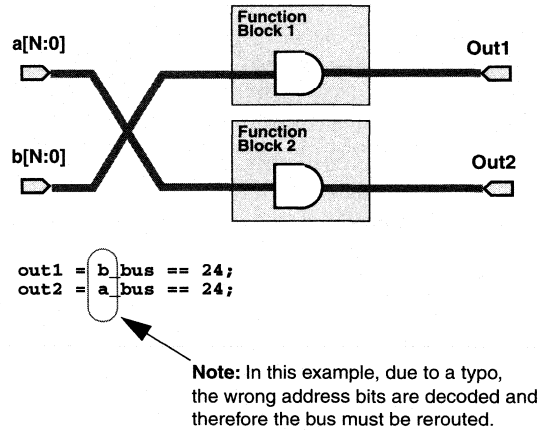


Figure 4: Address Decoder Design Iteration

Datapath Benchmark

This benchmark design, shown in Figure 5 and Figure 6, measures the affect of routing resources on the CPLD's pin-locking capability. This design contains a single 16, 32, or 36 bit wide data bus. A typical design change involving the reordering of data bits is illustrated in Figure 7.

The benchmark results shown in Figure 12 show that the Xilinx XC9500, AMD MACH 5, and Altera EPM7000S devices were able to accommodate the design changes without impact on design performance. Both the Lattice ispLSI1000 and ispLSI2000 devices sacrificed performance (up to 80%) to reroute the design when pinlocked. Since only one logic block input was required for each output, this performance degradation can be attributed to poor routing resources, or fitter performance, or both, but cannot be attributed to logic block fan-in.

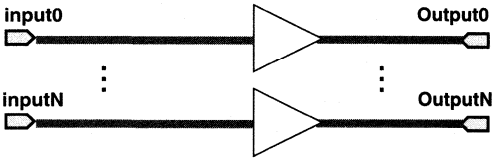
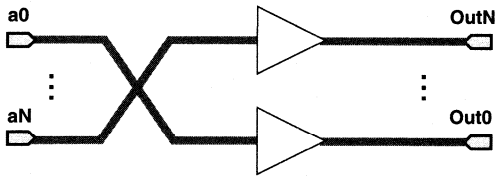


Figure 5: Data Path

```
MODULE REORDER
TITLE 'Datapath test'

//inputs
input15..input0 pin; "inputs

//outputs
output15..output0 pin istype 'com'; "outputs

equations

[output15..output0] = [input0..input15];

END
```

Figure 6: Data Path Code




```
[output15..output0] = [input0..input15];
```

Note: In this example, due to a typo, the data bits are ordered incorrectly therefore they must be reordered.

Figure 7: Datapath Design Iteration

Address Counter Benchmark

This benchmark design shown in Figure 8 and Figure 9, measures the effect of routing resources and function block fan-in on the CPLDs pin-locking capability when macrocell feedbacks and other high fan-out signals are involved. The design contains two 16, 24, or 32 bit loadable address counters loaded from separate buses but with common clock and hold signals. A typical design change correcting initial count load value is illustrated in Figure 10.

The benchmark results shown in Figure 13 demonstrate the superiority of the Xilinx pin-locking capability vs. Altera Lattice, and AMD. All Xilinx XC9500 devices were able to accommodate the design changes without impact on design performance. When the Altera EPM7000, EPM7000E and in-system-programmable EPM7000S routing resources were stressed, performance didn't just degrade, the devices completely failed to route. The Lattice ispLSI2000 devices used several layers of logic in the initial design, with correspondingly low f_{MAX}. This enabled the fitter to reroute the design using alternate routing paths, with less performance degradation (20%) than designs initially using only one logic level.

The MACH 5 devices were able to accommodate the design changes without incurring additional time delays for the 16- and 24-bit address counters. This was possible because segment delays were incurred during the initial design compilation and not just during the re-route. However, they completely failed to route the 32-bit wide counters during the initial design compilation. This can be attributed to poor fitter performance, inadequate routing resources, or both.

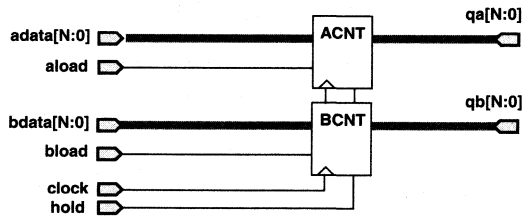


Figure 8: Address Counter

```
MODULE CNTSWAP
TITLE 'Counter Swapping'

//inputs
clock pin; "clock
hold pin; "counter hold
ain15..ain0, aload pin; a data bus
bin15..bin0, bload pin; b data bus

//outputs
qa15..qa0, qb15..qb0, pin istype 'reg';

//variables
acount = [qa15..qa0]; adata = [ain15..ain0];
bcount = [qb15..qb0]; bdata = [bin15..bin0];

equations

acount := adata & aload
   # acount & !aload & hold
   # (acount + 1) & !aload & !hold;
acount.clk = clock;

bcount := bdata & bload
   # bcount & !bload & hold
   # (bcount + 1) & !bload & !hold;
bcount.clk = clock;

END
```

Figure 9: Address Counter Code

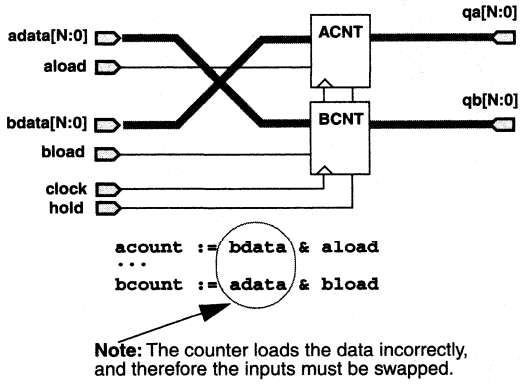


Figure 10: Address Counter Design Iteration

Conclusion

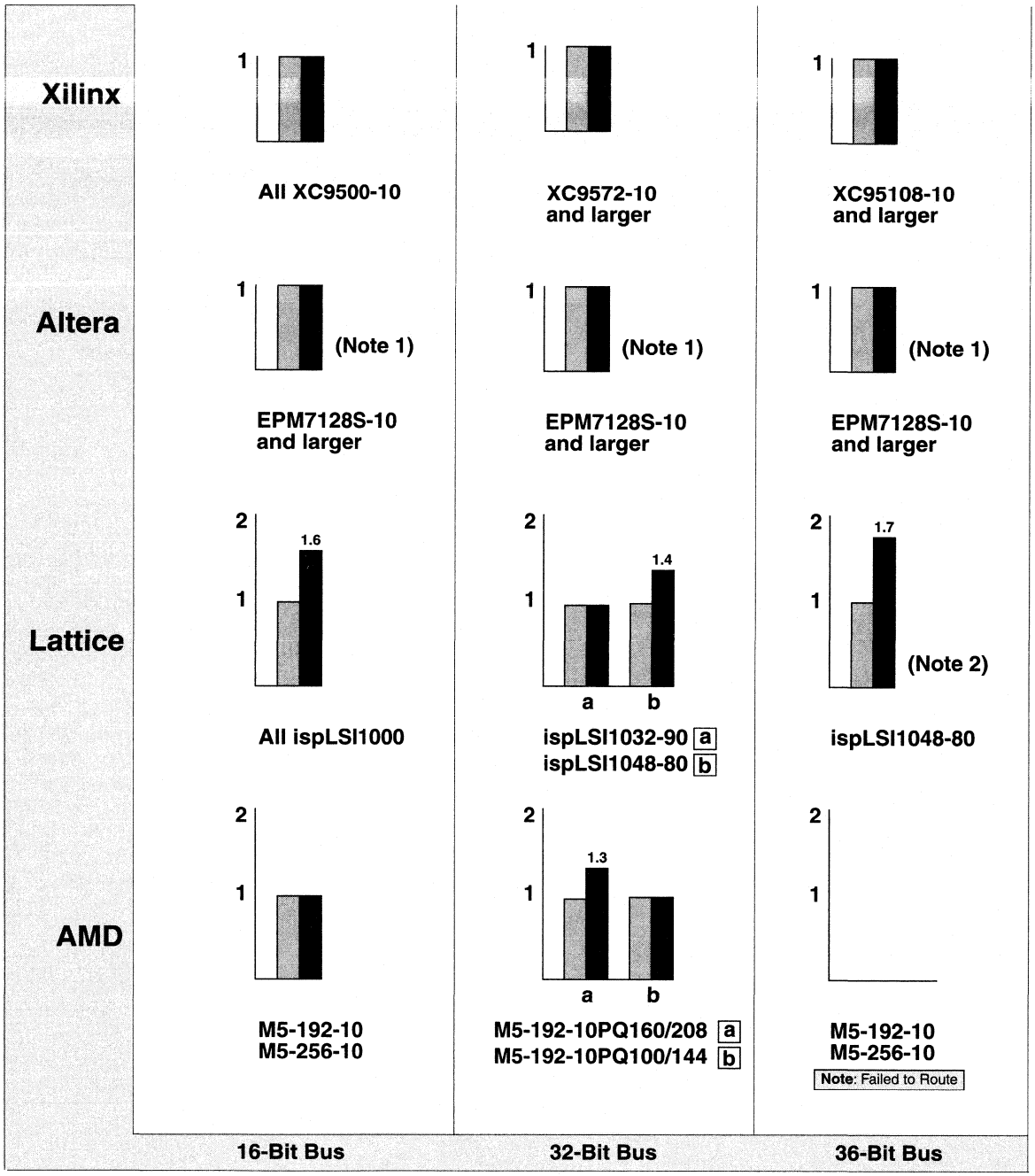
The benchmark results show the superior pin-locking performance of the Xilinx XC9500 family. This performance is consistent across all devices and package types. The wide function block fan-in enables pin-locking of wide, high speed logic functions. And, because feedthroughs are not needed for routing, there is no performance degradation due to routing congestion. This timing consistency is as important as routing ability for maintaining pin-locked designs.

Altera MAX7000, 7000E, and 7000S devices exhibit pin-locking problems due to sparse routing resources. This occurs when many macrocell feedbacks are used and these macrocells drive output pins. The problem is made worse in higher pin count versions of these Altera devices.

The current Altera software does not use logic feedthroughs to resolve routing congestion. Instead, when routing congestion occurs, the design fails to route. This failure can lead to unnecessary PC board re-work to accommodate the design change.

Lattice ispLSI devices suffer from poor routing resources and narrow function block fan-in. The Lattice fitter does use logic resources as feedthroughs in an effort to completely route the design. However, the impact on performance and utilization is significant, even for these very simple designs. In some cases t_{PD} slows as much as 80% and macrocell count increases 25%. The Lattice ispLSI devices employ a poor pin-locking architecture.

The AMD MACH 5 devices appear to suffer from a combination of inadequate routing resources and poor fitter performance. Narrow functions always re-routed after pin-locking, but with some performance degradation caused by segment delays. However, re-routing of wide functions is the strongest test of the affect of routing resources on pin-locking; in these tests, the AMD MACH 5 failed completely because it could not route the designs, even during the initial design compilation.



t_{PD} Performance — Initial Compile Before Pin-Locking

t_{PD} Performance — After Pin-Locking, with changes

Note 1: Lower density 7KS not avail., or would not generate pinout.

Note 2: Not enough I/O for design, using ispLSI1032.

Figure 11: Address Decoder Pin-Locking Performance

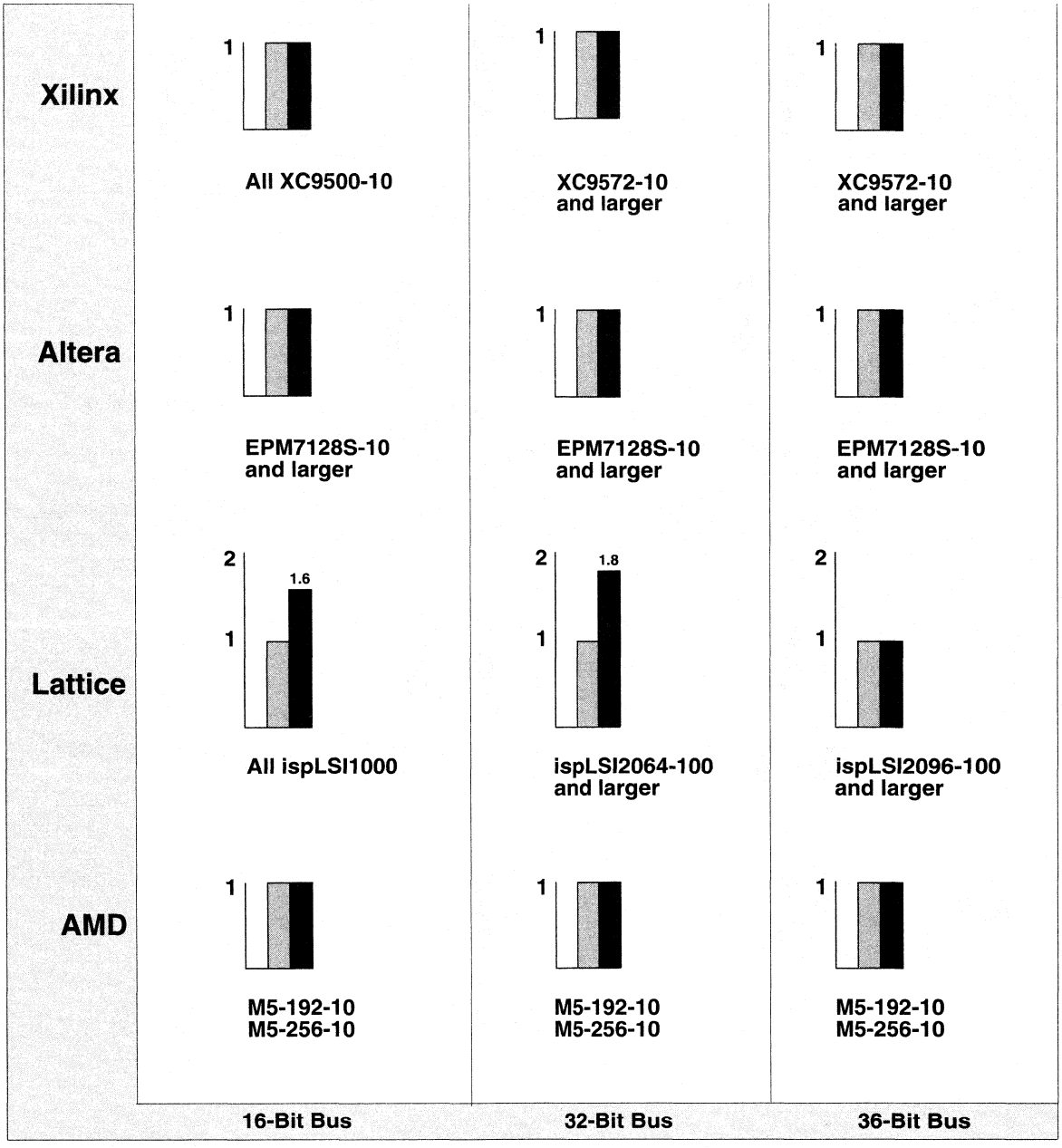


Figure 12: Datapath Pin-Locking Performance

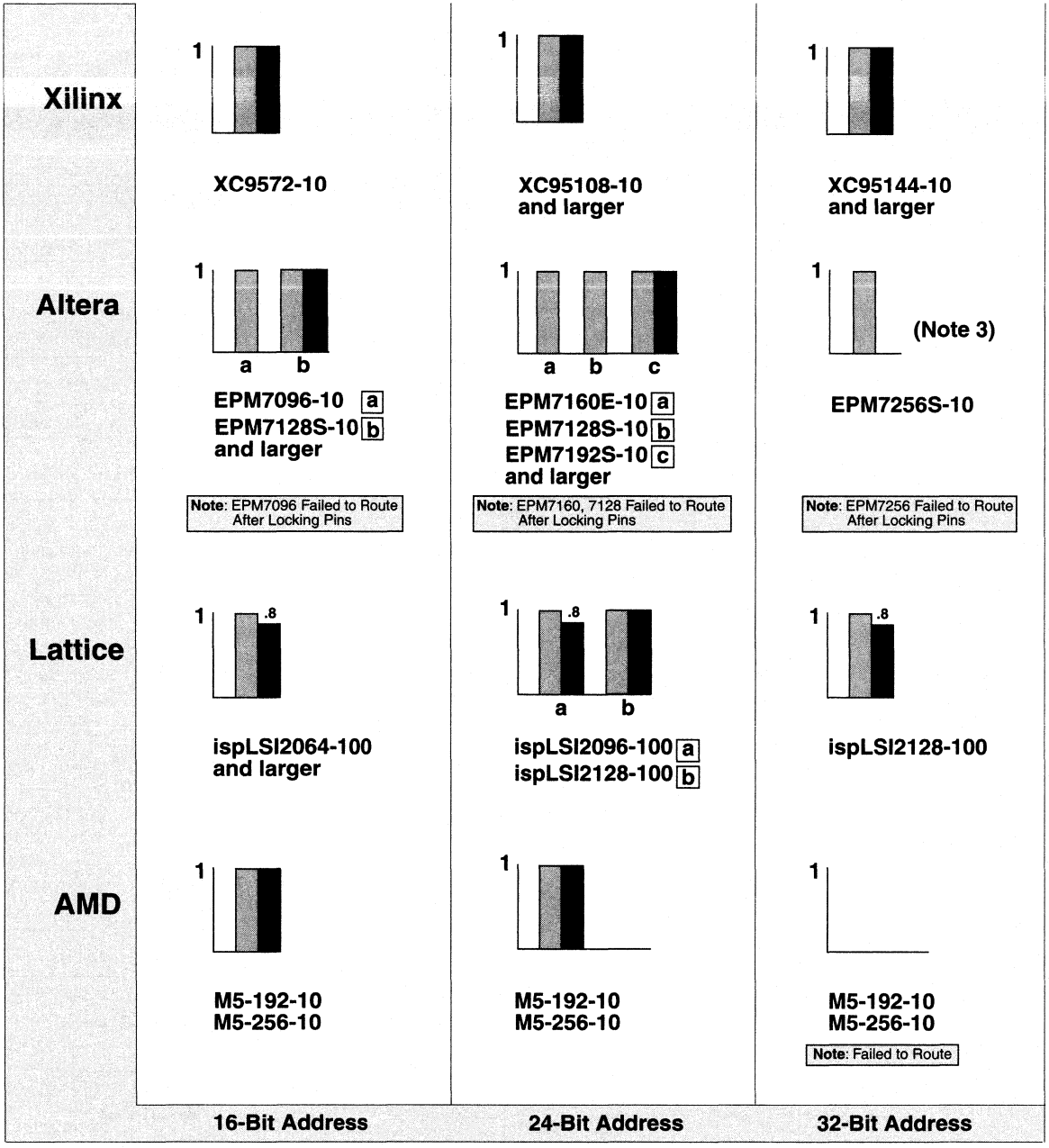


Figure 13: Address Counter Pin-Locking Performance

Using the XC9500 Timing Model

January, 1997 (Version 1.0) Application Note

Summary

This application note describes how to use the XC9500 timing model.

Xilinx Family

XC9500

Introduction

All XC9500 CPLDs have a uniform architecture and an identical timing model, making them very easy to use and understand. To determine specific timing details, users need only compare their paths of interest to the architectural diagrams and, using the timing model presented here, perform a simple addition of incremental time delays.

Device Timing Overview

External signals arrive at the pins and are delivered through the I/O block to the FastCONNECT Switch Matrix. From the switch matrix, they are dispatched to the various Function Blocks (FBs). As the signals enter the FBs, they incur incremental time delays depending on how the signals are used within the FB. For example, all logic signals must pass through the AND array where they encounter product terms which add a time delay as the signals pass through. Additional time delay may be encountered if the signals pass through the cascade logic and are redirected toward macrocells that are further away than those directly attached to the product terms.

There are additional timing requirements such as setup and clock-to-output times involved with passing signals through a flip-flop. As the signals exit flip-flops, they either pass to the outside world, through the I/O pins, or are fed back into the FastCONNECT switch matrix for additional logic operations.

Design timing can be manually analyzed as separate signals, each having unique timing parameters that are easily calculated. However, the Xilinx software provides a detailed timing report that tallies and summarizes all paths specified by the designer. The timing report is based on the model described here and is a convenient text based mechanism for isolating and displaying timing relationships.

The timing model shown in Figure 1 is used by the M1 release of the Xilinx XACT*step* development software which provides complete fitters for the XC9500 family as well as the timing models for simulation and detailed static timing reports.

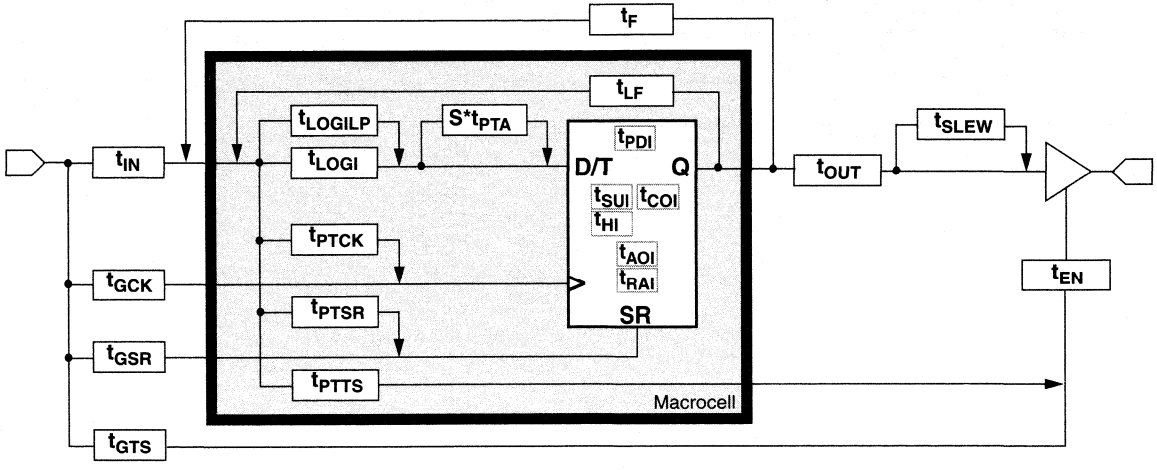


Figure 1: XC9500 Detailed Timing Model

Timing Model

The timing model shown in Figure 1 resembles the XC9500 macrocell with additional time delays included to account for the FastCONNECT Switch Matrix and the I/O buffers. As signals progress through an XC9500 device, they encounter each of these delays which are tallied to arrive at a cumulative time delay for that signal. Table 1 provides a detailed definition of each parameter contained in Figure 1. The exact values of these parameters for each device can be obtained from the data sheets.

Table 1: Key XC9500 Internal Timing Parameter Definitions

Symbol	Parameter
Buffer Delays	
t_{IN}	Input buffer delay
t_{GCK}	GCK buffer delay
t_{GSR}	GSR buffer delay
t_{GTS}	GTS buffer delay
t_{OUT}	Output buffer delay
t_{EN}	Output buffer enable/disable delay
Product Term Control Delays	
t_{PTCK}	Product term clock delay
t_{PTSR}	Product term set/reset delay
t_{PTTS}	Product term 3-state delay
Internal Register and Combinatorial Delays	
t_{PDI}	Combinatorial logic propagation delay
t_{SUI}	Register setup time
t_{HI}	Register hold time
t_{COI}	Register clock to output valid time
t_{AOI}	Register async. S/R to output delay
t_{RAI}	Register async. S/R recovery before clock
t_{LOGI}	Internal logic delay
t_{LOGILP}	Internal low power logic delay
Feedback Delays	
t_F	FastCONNECT matrix feedback delay
t_{LF}	Function Block local feedback delay
Time Adders	
t_{PTA}	Incremental product term allocator delay
t_{SLEW}	Slew rate limited delay

Timing Calculation Examples

Table 2 shows how various external timing parameters are derived from the internal timing parameters. For example, t_{PD} is the sum of the input buffer time delay (t_{IN}), the logic time delay (t_{LOGI}), the flip-flop pass through delay (t_{PDI}), and the output buffer time delay (t_{OUT}), as shown in Figure 2. Note that the input buffer delay is combined with the time delay accrued when the entering signal passes through the FastCONNECT switch matrix.

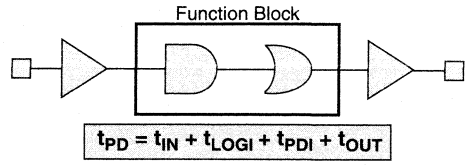


Figure 2: Simple t_{PD} Example

Table 2: Expressions for Key Timing Parameters Derived from Table 1

Symbol	Parameter	Calculation
t_{PD}	Propagation delay*	$t_{IN} + t_{LOGI} + t_{PDI} + t_{OUT}$
t_{SU}	Global clock setup time*	$t_{IN} + t_{LOGI} + t_{SUI} - t_{GCK}$
t_H	I/O hold time after GCK	$t_{GCK} + t_{HI} - t_{IN} - t_{LOGI}$
t_{CO}	Global clock-to-output*	$t_{GCK} + t_{COI} + t_{OUT}$
f_{CNT}	16-Bit counter frequency	$1/(t_{COI} + t_{LF} + t_{LOGI} + t_{SUI})$
f_{SYSTEM}	Internal system clock period*	$1/(t_{COI} + t_F + t_{LOGI} + t_{SUI})$
t_{PSU}	P-term Clock setup time*	$t_{IN} + t_{LOGI} + t_{SUI} - t_{IN} - t_{PTCK}$
t_{PH}	I/O hold time after p-term clock	$t_{IN} + t_{PTCK} + t_{HI} - t_{IN} - t_{LOGI}$
t_{PCO}	Product term clock-to-output	$t_{IN} + t_{PTCK} + t_{COI} + t_{OUT}$
t_{OE} t_{OD}	GTS to output enabled/disabled	$t_{GTS} + t_{EN}$
t_{POE} t_{POD}	P-term OE to output enabled/disabled	$t_{IN} + t_{PTTS} + t_{EN}$

* See AC Table Parameters

Figure 3 shows a variation on the simple t_{PD} example with the addition of cascaded product terms. The time delay from input A is slightly altered by the addition of one t_{PTA} value which accounts for the additional product terms. The XC9500 can accept and deliver product terms in either direction, so the t_{PTA} time delay will handle this factor. Also, product terms may arrive from non-adjacent macrocells, which would require an additional t_{PTA} to be added. Therefore, a single cascade delay may in some cases not be what the design software has chosen. This cascade timing can be controlled by using timing driven optimization, described in detail in the Design Optimization application note.

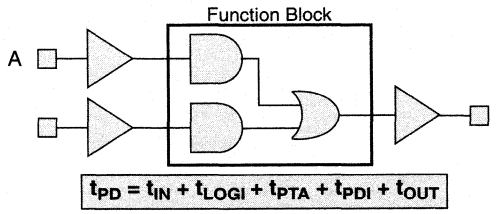


Figure 3: t_{PD} with Cascaded P-Terms

Figure 4 shows the results of supplementing single pass logic with an additional pass through another macrocell. In this case, there is a single pass through the input and output buffers, two passes through the macrocell logic, and a single pass through the feedback path. The feedback path can be either through the general feedback (t_F) or the local feedback (t_{LF}), depending on timing constraints supplied by the designer in a .CST file.

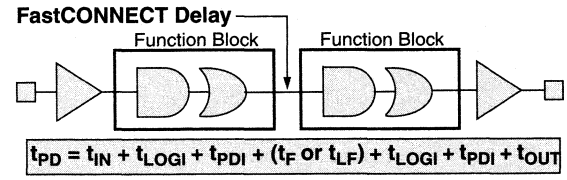


Figure 4: t_{PD} with Multiple Pass Logic

Figure 5 shows the situation for a simple flip-flop clocked by a global clock signal (GCK). The expressions for t_{CO}, t_H, and t_{SU} in Table 2 are valid for this arrangement.

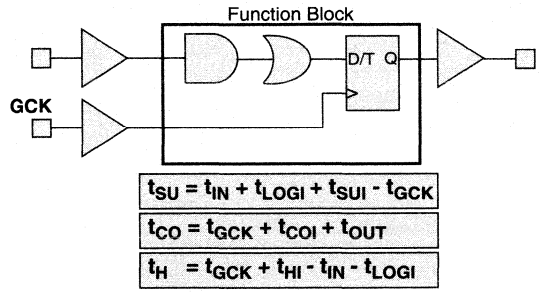


Figure 5: Simple Flip-Flop Path
(Note: Global clock)

Figure 6 shows the addition of another layer of macrocell logic into the situation described in Figure 5. The t_{CO} and t_H expressions remain the same, but the t_{SU} expression is increased by another ($t_{LOGI} + t_{PDI} + (t_F$ or $t_{LF})$) depending on timing constraints supplied by the designer in a .CST file.

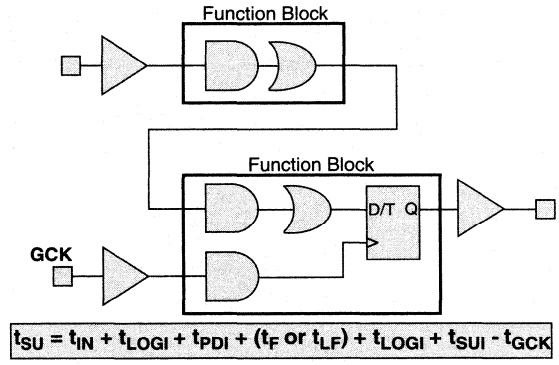


Figure 6: Flip-Flop with Multiple-Pass Logic
(Note: Global clock, t_{CO} and t_H are unchanged.)

Figure 7 shows two flip-flops connected by a single level of logic, clocked by a global clock. The t_{SU} and t_H for flip-flop A are identical to that of Figure 5.

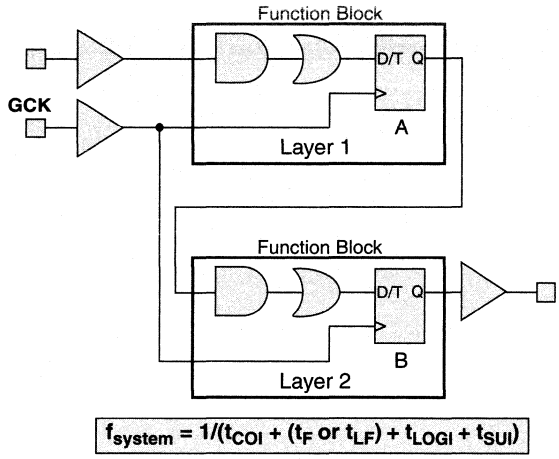


Figure 7: Multiple Flip-Flops with Single Level Logic (Note: Global Clock)

Figure 8 shows a single flip-flop with a product term clock. This arrangement differs from Figure 5 only in that the clock input comes from a product term clock. The entry for t_{PCO} in Table 2 reflects this variation. The timing for t_{PSU} and t_H is calculated using the product term clock timing parameters.

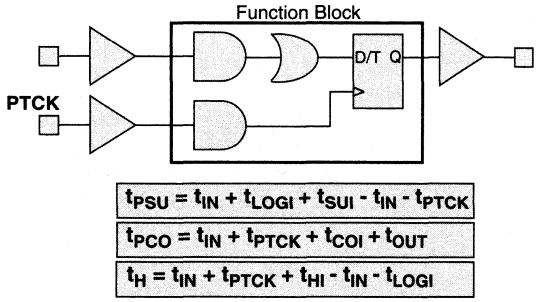


Figure 8: Single Flip-Flop with Product Term Clock

Figure 9 shows the timing for driving valid data onto a bus with respect to a rising clock edge, a common configuration that occurs in high speed buses. This is sometimes called t_{VAL}. In this example, it is assumed that Function Block feedback passes through the local feedback paths.

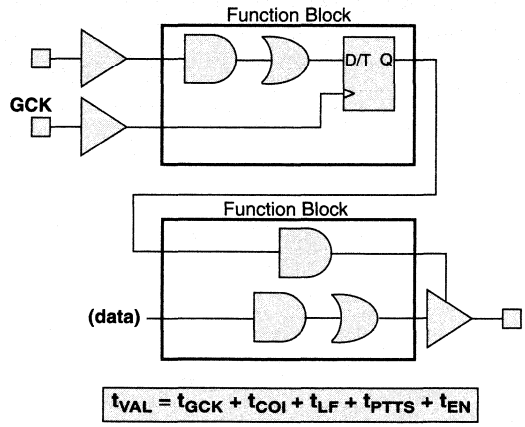


Figure 9: Flip-Flop-Controlled Output Enable

Conclusion

This set of examples is sufficient to describe a large number of design configurations, and other examples can easily be derived from the timing model. For manual calculations, other timing delays such as t_{SLEW} and t_{LOGILP} are easily added to the overall timing as required.

Summary

This application note shows the tradeoffs that can be made to gain the greatest possible densities and speeds for schematic, behavioral, and VHDL implementations.

Xilinx Family

XC9500

Introduction

The advanced architecture of the XC9500 CPLDs can implement a wide range of design densities and speeds, while allowing significant design changes to be made without modifying the original pinouts. The Xilinx XACT*step* software (fitter) makes full use of this flexibility, giving designers a number of easy to use options for optimizing design performance.

Optimizing Density

The following techniques can be used for those designs requiring the maximum amount of device resources.

Controlling the Macrocell Input and Pterm Limits

The Xilinx XACT*step* design software has options for setting the limits of the fitting algorithms. Two key options are input collapsing and product term collapsing. The defaults are 36 inputs and 15 p-terms.

The 36 input limit restricts the fitter from collapsing logic functions that will require more than 36 inputs. This is based on the actual number of input signals delivered into each function block (FB). Although this is a physical limit, it is possible to exceed it by permitting the software to wire AND signals in the FastCONNECT switch matrix. The software will detect and utilize wire ANDing whenever it is required. This will allow the product of several signals to be used as inputs to the FB on a single input line.

Similarly, the default of 15 p-terms restricts the fitter from collapsing logic functions that would use more than 15 product terms. This limits the size and the cascade timing penalty. Designers may adjust these two options to achieve improved fitting.

When a design fails to fit, a report is created, summarizing the unmapped logic and pins. This may be used as a guide for adjusting the input collapsing limit and the product term collapsing limit. The easiest approach is to experiment with different values. For example, when a design fails to fit, as shown in Figure 1, adjust the input collapsing limit and run another pass with the fitter. This will result in a successful fit or a new report of unmapped logic and pins.

```
************Resources Required by Unmapped Logic and Pins*****************
** Logic **
Signal                    Total     Signals              Pwr    Slew
Name                      Pt        Used                 Mode   Rate
$18N1345                  12        19                   STD
DPCS                      10        19                   STD
IO_CYC_DET                10        14                   STD
LBE1_N                    10        13                   STD
SPEC_CYC                  10        14                   STD
********************Function Block Resource Summary**********************
Function    # of          FB Inputs  Signals     Total       O/IO      IO
Block       Macrocells    Used       Used        Pt Used     Req       Avail
FB1         13            36         39          47          0/0       17
FB2         12            36         37          56          10/0      17
FB3         14            36         36          73          5/1       17
FB4         11            36         38          84          7/0       17
FB5         12            36         40          84          8/1       17
FB6         10            36         38          83          7/0       16
FB7         12            35         35          83          10/1      16
FB8         10            36         37          43          5/3       16
            ----                                  ----        ----      ----
            94                                    553         52/6      133
```

Figure 1: Report Showing a Design That Failed to Fit

If the list of unmapped resources appears to be growing, either raise or lower the limit for the next attempted fit. At some point, the list will not improve, indicating that the best ratio has been found. Then, adjust the product term collapsing limit and repeat the process to find the optimum ratio. Being systematic about the process ensures the quickest convergence. Figure 4 shows a report file for a design that fit after the input and pterm limits were adjusted according to this process.

On the PC platform, the global collapsing p-term limit and collapsing input limit can be set in the optimization template of the flow engine. The same limits are set on the workstation platforms by using the "-pterms" and the "-inputs" switch on the command line.

Preventing Logic Collapsing

Multiple levels of logic will frequently be collapsed to achieve the fastest pin-to-pin speeds or the greatest f_{MAX}, as shown in Figure 2.

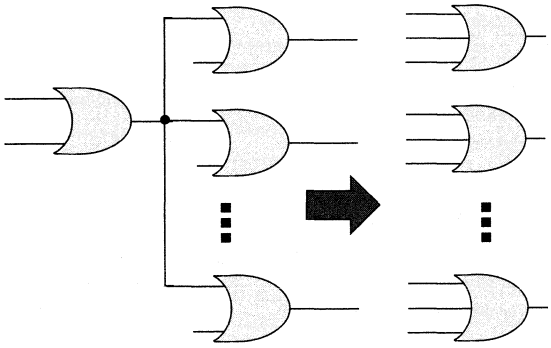


Figure 2: Logic Collapsing

Designers can easily control the collapsing of specific functions by specifying the fanout nodes that are to be exempt from the collapsing process. This can be accomplished in a schematic by using the **OPT=OFF** attribute. In HDL designs, this is accomplished by specifying the exempt nodes within the text of the HDL design file using properties or attributes. After flattening, the specified nodes are maintained as shown in Figure 3, resulting in greater density designs.

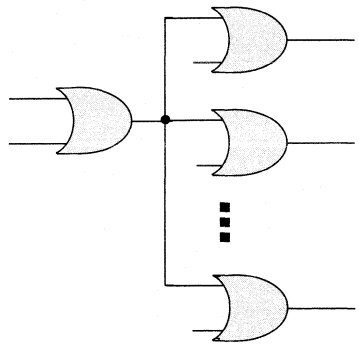


Figure 3: Uncollapsed Node

The increased density is achieved by implementing a high fanout function in a single macrocell versus collapsing the same function into each of the fanouts. The greater the number of fanouts an equation has, the greater the payoff.

Using Global Control Signals

By taking advantage of the global clocks (GCK), global set/reset signals (GSR), and global 3-state signals (GTS), which are available in every macrocell, designers can save the local product terms that would have been used for those same functions. These global signals also provide higher performance than local signals and they conserve the valuable FB inputs which can then be used for other logic signals.

```
********************Function Block Resource Summary**********************
Function    # of          FB Inputs  Signals    Total       O/IO      IO
Block       Macrocells    Used       Used       Pt Used     Req       Avail
FB1         11            36         41         28          0/0       17
FB2         14            36         43         45          0/0       17
FB3         18            36         37         46          12/1      17
FB4         11            36         41         47          8/1       17
FB5         14            36         36         54          8/1       17
FB6         10            35         37         46          8/1       16
FB7         13            36         39         45          7/2       16
FB8         18            36         36         78          9/0       16
            ----                                 ----         ----      ----
            109                                  389         52/6      133
```

Figure 4: Report Showing a Design That Fits

Optimizing Timing

XACT*step* provides three convenient means for timing optimization: Automatic timing optimization, TSPECs, and variable p-term collapse limits.

Using Automatic Timing Optimization

The automatic timing optimization minimizes the longest path in a design. It also improves the general timing of the whole design if detailed timing control (TSPECs) is not requested. Furthermore, it usually has no significant effect on design density or software run time. Timing optimization can be controlled in the optimization template on the PC, or by using the -notiming switch on workstation platforms.

The default for timing optimization is OFF.

Using TSPECs

Timing Specifications (TSPECs) define the required timing for specific paths in a design, causing the software to choose the optimum logic mapping to achieve the specified delays. TSPECs allow precise control over which paths are to be optimized. Specifically, this affects the collapsing order and the assignment of critical product terms near a macrocell output. When a signal has a **TSPEC** attribute, the fitter will use various methods to obtain the specified performance. These methods include using logic collapsing, limiting product term cascading, and using fast feedback paths within the FB.

See Appendix 1 for more information on using TSPECs.

Using P-Term Collapse Limits

Increasing the global p-term collapse limits increases the collapsing of combinatorial nodes into their fanouts. This usually increases the speed of the design but requires more device resources to implement.

Optimizing Schematic Designs

The following optimization techniques can be used in schematic designs.

Slew Rate Control

Assign the **FAST** attribute to output buffers or pads to specify the fastest slew rate. Unspecified outputs default to the slow slew rate.

Power Control

Assign the **LOWPWR** attribute to those signals that don't require the highest speed, in order to reduce power consumption. This attribute only affects macrocells, not the I/O blocks. All unspecified outputs default to the higher power, higher speed option.

Local Feedbacks

To use a local feedback path, use the **LOC** attribute to assign both the source and destination components to the same function block. Then, apply a TSPEC to that path such that it can only be met using a local feedback path.

Logic Optimization

Use the **OPT=OFF** attribute on selected nodes to inhibit the collapsing of those logic functions.

Optimizing ABEL Designs

The Xepld property statements are used within ABEL designs to control optimization. These property statements are passed directly to the Xilinx fitter and are not used by ABEL.

Slew Rate Control

The **FAST** property controls the output slew rate, and there can only be one **FAST** property used in each design. If there are only a few signals that require a fast slew rate, they can be listed individually after the property, and the remaining signals will be slew rate limited. Or, if there are only a few signals that need to be slew rate limited, then those signals can be listed.

```
xepld property 'fast on';
"all pins have fast slew rate

xepld property 'fast on x1 x2';
" only x1 and x2 are fast
" the remaining pins are slew limited

xepld property 'fast off x1 x2';
"only x1 and x2 are slew limited
"the remaining pins are fast
```

Local Feedbacks

The **PARTITION** property is used to force the specified signals into a specified FB. For example, the following statement will force signal A and B into function block 1.

```
xepld property 'PARTITION FB1 A B';
```

This xepld property statement, can be combined with a TSPEC in a .CST constraints file to specify timing. For example:

```
TIMESPEC = "TS01=FROM:FFS(A):TO:FFS(B)=3"
```

Power Control

The **PWR** property controls the power settings for individual macrocells.

```
xepld property 'pwr low';
"places all macrocells in low power mode

xepld property 'pwr low x1 x2'
```

```
"places x1 and x2 in low power mode
"the remaining in STD power mode

xepld property 'pwr std x1 x2'
"places x1 and x2 in STD power mode
"the remaining in low power mode
```

Logic Optimization

The **LOGIC_OPT** property allows the user to control the logic optimization done by the fitter. This should be used on selective nodes where collapsing those nodes would cause the design to become very large.

```
xepld property 'logic_opt off';
"Preserves all combinatorial nodes

xepld property 'logic_opt off x1';
"preserve x1 and collapse other nodes to
fitter limits
```

Optimizing VHDL Designs

VHDL support varies among the existing vendors in how the information is passed to the fitter. The summaries that follow describe the methods used with the Synopsys and Metamor VHDL tools.

Synopsis

- **Slew Rate Control** is accomplished with a script command:

```
-set_pad_type-slewrate NONE <port names>
```

- **Local Feedbacks** are accomplished by using the **LOC** attribute with a TSPEC (.CST file). **LOC** attributes are handled as follow (the italic text indicates additions to a standard script):

```
edifout_write_properties_list = loc
edifout_netlist_only = true
edifout_power_and_ground_representation =
cell
analyze-format vhdl<design_file_name>
elaborate <design_name>
compile
set_attribute <output_port> LOC <fb_name>
-type string

set_pad_type -slewrate NONE all_outputs()
set_port_is_pad "*"
insert_pads
write -format edif -hierarchy -output
<design_name>.edif
```

- **Power Control** is accomplished with the following script alterations, which pass low power properties into the design file (italic type indicates additions to a standard script):

```
edifout_write_properties_list = LOWPOWER
edifout_netlist_only = true
edifout_power_and_ground_representation =
cell
analyze-format vhdl <design_file_name>
elaborate <design_name>
compile

set_attribute <register_net>_REG LOWPWR
ON -type string

set_pad_type -slewrate NONE all_outputs()
set_port_is_pad "*"
insert_pads
write -format edif -hierarchy -output
<design_name>.edif
```

  Note that the register net will have appended a extra _REG to the name.

Metamor

- Slew rate control is managed with a Metamor library attribute. For a library attribute, include the metamor libraries. For example:

```
LIBRARY metamor;
USE metamor.attributes.all;
```

  Then, apply a property to the output signal as follows:

```
attribute property of B : signal is "Fast"
```

- Local feedbacks are handled with the **LOC** attribute and a TSPEC. To apply a **LOC** attribute, the designer must first define it. This is done with the following code.

```
attribute LOC : string;
```

  Then, apply the attribute to specific signals, as follows:

```
attribute LOC of B : signal is "FB2";
```

  A TSPEC can also be applied using a .CST constraints file.

- Power control is accomplished by defining a **LOWPWR** attribute and applying it to signals as well. This can be done with the following code:

```
attribute LOWPWR : string;
```

  Then, apply the attribute to specific signals as follows:

```
attribute LOWPWR of B : signal is "on";
```

Appendix 1 — TSPEC and .CST File Format

TSPECs can be applied to designs using constraint files. Constraint files can be used with all software flows, and there are several ways to use them - depending on the specific path taken to the fitter step. The constraint file is named "design.cst", where "design" is the same character string as the other design files. The .CST file is simply a text file comprised of several lines that dictate specific timing relationships must be met within the design.

There are only two types of signals defined in TSPECs. These are FFS (flip-flops) and PADS (primary I/Os). In general, four parameters need to be controlled:

- t_{PD}, the time a combinatorial output becomes valid after an input has changed is defined using a PADS to PADS definition.
- t_{SU}, the external setup time for a signal at the pin of the part before applying a clock is defined using a PADS to FFS definition.
- f_{MAX}, the cycle time at which a set of flops can be clocked internally is defined using FFS to FFS definition.
- t_{CO}, the clock to out time of the design is specified by using a FFS to PADS definition.

Specifying the estimated external clock speed requires the user to specify t_{SU}, f_{MAX}, and t_{CO}. The external clock speed will be calculated by using the longest delay of the three parameters.

The following examples will show the basic syntax of the constraint file and apply these definitions to control timing.

Example 1: Specifying t_{PD}

```
TIMESPEC = "TS01 = FROM:PADS:TO:PADS =
7.5";
```

This TSPEC will constrain all signals that originate and terminate at pins with a maximum time of 7.5 ns. When specifying tpd, however, the designer is usually only interested in certain paths in the design. The designer can specify groups of signals using the "timegrp" command. For instance, assume the design has a set of inputs A15 to A0, and a set of outputs X15 to X0. By using a "timegrp", the designer can limit the constraints to only those signals. Note that the designer can use the "*" for a wildcard match.

```
TIMEGRP = "AddressIn = PADS(A*)";
TIMEGRP = "AddressOut = PADS (X*)";
TIMESPEC = "TS01 = FROM: AddressIn:TO:Ad-
dressOut = 7.5";
```

The TIMESPEC command specifies that the delay between the inputs (A*) and the outputs (X*) is less than or equal to 7.5 nanoseconds.

Example 2: Specifying f_{MAX}

f_{MAX} is an extension of the TIMESPEC concept mentioned in Example 1. To specify the maximum required operating frequency for a flip flop or a group of flip flops, just use the following equation:

```
TIMESPEC_value = (1/fMAX)
```

where fMAX is the required operating frequency. To get 100 Mhz operation for a group of flip flops, 10 ns must be specified. Again, this can be applied over a group of flip flops using the TIMEGRP expression.

As a more concrete example, consider a design where multiple asynchronous chip select signals, Csel3 to Csel0, are being synchronized by a set of input registers. Both the input registers and the Csel registers are clocked by one source. First, the designer can define the destination registers, again using a wildcard match. Then, the designer can apply a TSPEC from all FFS to the timegroup, Csel.

```
TIMEGRP = "Csel=FFS(C*)";
TIMESPEC = "TS01 = FROM:FFS:TO:Csel=10";
```

This constraint limits the cycle time from the output of the input registers to the Csel registers to be a maximum of 10ns. This would allow us to clock the design at 100mhz without violating internal setup times.

Example 3: Specifying t_{SU}

To specify the worst case setup time for device inputs, use the following equation:

```
TIMESPEC_value = (desired setup time) +
(tGCK)
```

where t_{GCK} is the fast clock buffer time obtained from the target device data sheet.

For example, assume the desired setup time to clock delay is no more than 5.5 ns for the timespec design. The value of t_{GCK} for the XC95108-7 is 2.5 ns. Therefore the TIMESPEC value is (5.5 ns) + (2.5ns) = 8.0 ns.

In this example, assume that the Csel registers are being driven by inputs A15 to A0. the designer can now group the inputs and the registers together, and apply a TSPEC from the inputs to the flip-flops.

```
TIMEGRP = "Address=PADS(A*)";
TIMEGRP = "Csel=FFS(C*)";
TIMESPEC = "TS01=FROM:Ad-
dress:TO:Csel=8.0";
```

Example 4: Specifying t_{CO}

To specify the worst case clock-to-output time for a registered device output clocked by a global clock (t_{CO}), use the following equation:

TIMESPEC_value = (desired t_{CO}) - (t_{GCK})

In this example, assume that the Csel registers require a t_{CO} of 5.5 ns. Again, the designer can group the registers together, and apply a TSPEC from the flip-flops to the output pads.

```
TIMEGRP = "Csel=FFS(C*)";
TIMESPEC = "TS01=FROM:Csel:TO:PADS=3";
```

This TSPEC will constrain the signal path delay from the output of a flip-flop to the pin to 3 ns.

Summary

This application note will help designers understand the XC9500 architecture and how to get the best performance from these devices.

Xilinx Family

XC9500

Introduction

To get the best performance from any CPLD, the designer must be aware of its internal architecture and how the various device features work together. This application note provides useful examples and practical details for creating successful designs. These design techniques apply to all XC9500 devices because the architecture is uniform across the family.

XC9500 Architecture

The XC9500 architecture is comprised of multiple identical function blocks internally connected by a fully populated FastCONNECT switch matrix. The XC9500 function block has 18 macrocells per block and supports pin-to-pin speeds as fast as 5 ns, with clock rates up to 125 MHz. I/O signals can interface with 5 volt, 3.3 volt, or both levels.

Figure 1 shows the XC9500 architecture. Note the regular structure of high speed function blocks centrally connected by the FastCONNECT matrix and surrounded by pins. Signals enter and exit on the pins, form logic operations within the function blocks, and form connections and logic operations within FastCONNECT. Each of these features is discussed in the following sections to highlight key functionality.

Interconnect Within Function Blocks

Function blocks (FBs) have 36 input sites. The FBs receive signals from the FastCONNECT matrix and input pins. The logic blocks generate 18 signals per FB from the 18 macrocells in each block, and each macrocell signal can drive its own dedicated I/O pin or feedback by entering the FastCONNECT matrix. Additional high speed local paths exist within the FB.

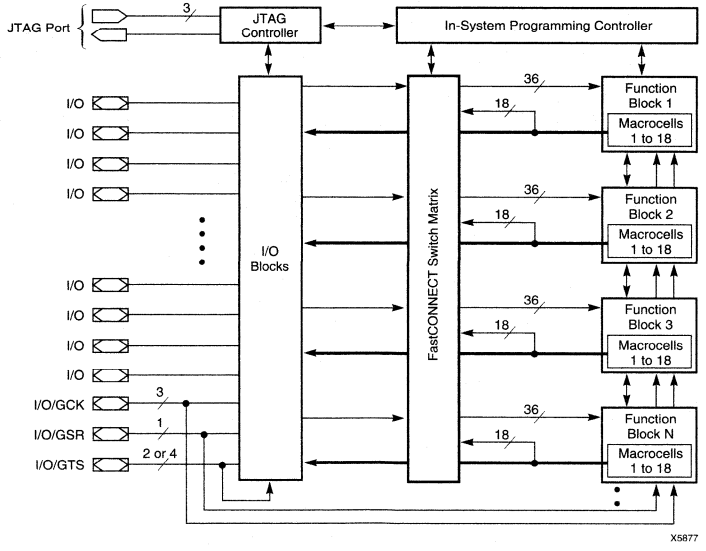


Figure 1: XC9500 Architecture

The FastCONNECT Switch Matrix

The FastCONNECT switch matrix attaches high speed signals to the function blocks. It also connects every macrocell output to the function blocks through a fully populated cross-point switch. This high degree of connectibility is a key factor that allows the designer to make design changes even after a device is mounted on a PC board.

Function Blocks

The function blocks, shown in Figure 2, are groups of 18 macrocells. Each FB has 90 product terms which can be assigned to any of the 18 macrocells. This provides optimum logic flexibility within the function block and supports pin-locked designs. The highest possible performance is attained by the software assigning a uniform five product terms per macrocell. The macrocell outputs can then drive output pins as well as provide inputs to both the FastCONNECT matrix and the FB in which it resides.

The Macrocell

In the default mode, there are 5 product terms that OR together driving the D input to the macrocell flip-flop, as shown in Figure 3. The most common arrangement includes an Exclusive-OR gate capable of performing parity, full addition, or logical inversion.

Another configuration exports product terms to a neighboring macrocell, increasing that macrocell's available product terms. Product term exporting is shown in Figure 3.

The XC9500 flip-flops can be configured as D- or T-type. This permits the building of efficient counters using only a few gates to drive the state transitions. Table 1 summarizes the number of product terms needed to build common logic functions; most datapath functions require one or fewer macrocells per bit.

Table 1: Macrocell/Product Term Allocation

Data Operation	P-Term Used
Shift Register	2
Counters	2-4
n:1 Mux	n
Adder	6
Exclusive-OR	2
Storage registers	1

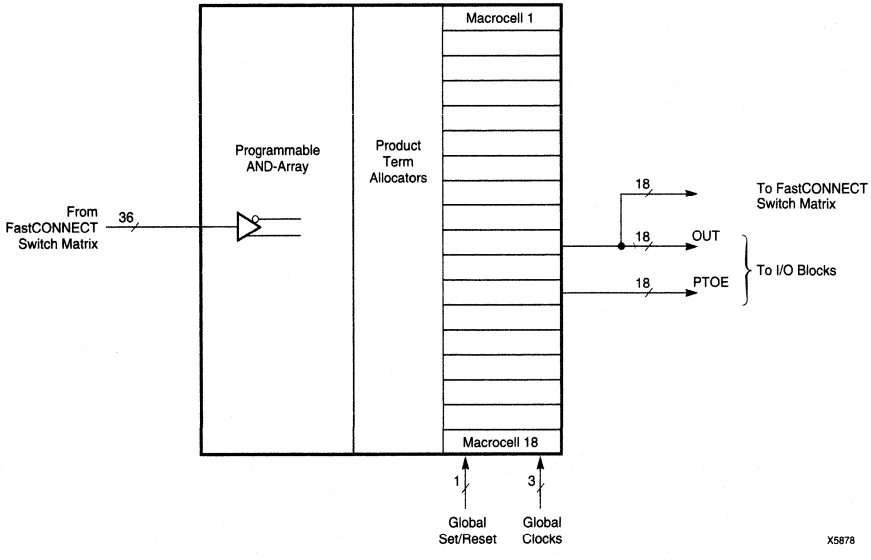


Figure 2: XC9500 Function Blocks

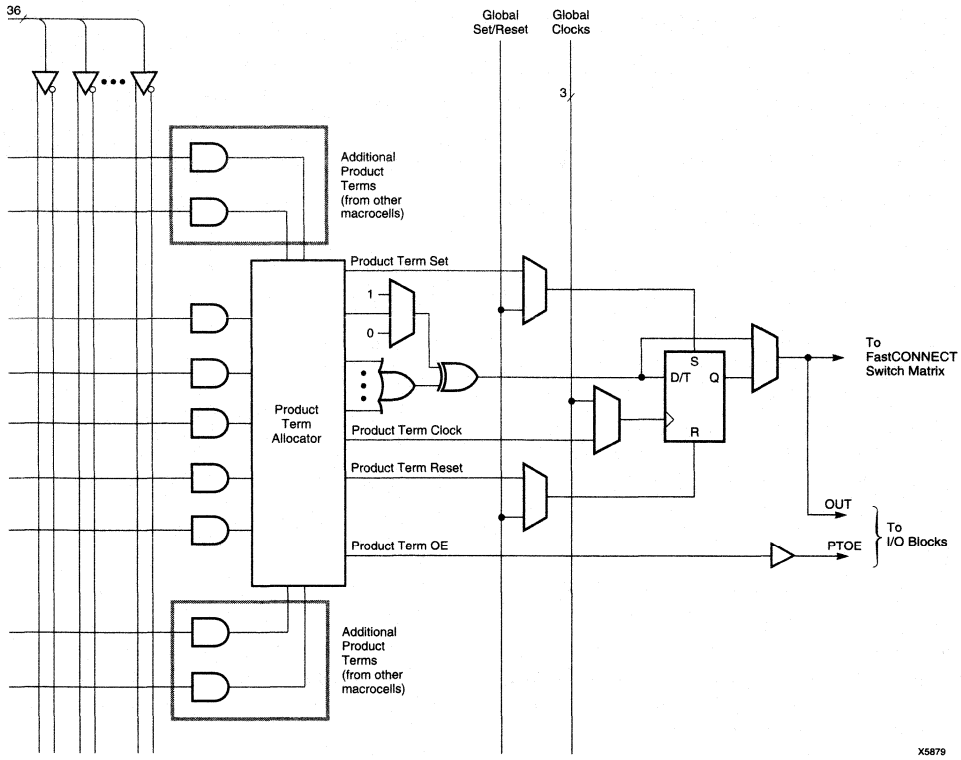


Figure 3: XC9500 Macrocells

Table 2 shows the pin compatibility of the XC9500 family. Designs can easily be migrated to larger or smaller devices. In many cases, greater density with equivalent speed can be gained by using larger parts. If a design is initially targeted to a smaller device, the same design can be moved into a larger device, if additional capability is required. This capability allows designers maintain their pin assignments, even when designs must be moved to a larger capacity device.

Moving designs from larger to smaller devices can also be accomplished, while keeping the original pinout, if the smaller device has enough resources to contain the design.

Table 2: XC9500 Available Packages and Device I/O Pins

Package	XC9536	XC9572	XC95108	XC95144	XC95180	XC95216	XC95288	XC95432	XC95576
44-Pin PLCC	34								
44-Pin VQFP	34								
84-Pin PLCC		69	69						
100-Pin PQFP		72	81	81					
100-Pin TQFP		72	81						
160-Pin PQFP			108	133	133	133			
208-Pin HQFP					166	166	168		
352-Pin BGA						166	192		
432-Pin BGA								232	232

Note: These numbers do not include the dedicated JTAG pins.

Automatic Software

The following design examples are created in ABEL. Typically, designers won't designate specific function mapping into XC9500 designs. However, designers occasionally like to control how a solution is implemented, and in that case, these methods may be of interest.

Boolean operators used by ABEL are **!** for invert, **#** for OR, and **&** for AND. Combinatorial logic expressions are formed with an equal sign, with operands and operators located on the right hand side of the expression.

Flip-flops are formed by writing expressions for the specific control pins of the flip-flop. The D-input is a special case, represented by the compound symbol ":=". Clock inputs are determined by the syntax *flip-flop_name*.clk, and reset inputs are designated by *flip-flop_name*.rst.

An ABEL design file contains a header section including optional documentation sections and mandatory declaration of inputs, outputs, global signals, and any user preferred arrangement of functions.

Logic AND

The FastCONNECT switch matrix is capable of combining signals with a wire-AND function. Signals entering the FastCONNECT matrix are assigned to function block inputs, and multiple signals, may form a wired-AND function, which reduces the macrocell logic requirements. This feature increases both the logic capacity and available signal inputs to the Function Blocks.

Gates

The following expressions show the basic logic operations.

```
ABAR = !A;
AORB = A#B;
AANDB = A&B;
ANORB = !(A#B);
ANANDB = !(A&B);
AEXORB = A$B;
AEXNORB = A!$B;
```

Multiplexers and Decoders

Using the above methods, compound expressions are formed to build logic functions. Using A0 to A3, B0 to B3, and SEL (select) as inputs, a multiplexer is described as follows:

```
DAT0 = SEL&A0 # !SEL&B0;
DAT1 = SEL&A1 # !SEL&B1;
DAT2 = SEL&A2 # !SEL&B2;
DAT3 = SEL&A3 # !SEL&B3;
```

The approach extends to larger multiplexers. The previous example uses one macrocell per data bit and leaves behind two unused product terms in each macrocell. To take advantage of four product terms per macrocell, the implementation expands as follows:

```
DAT0 = S1&S0&D0 # S1&!S0&C0 #
 !S1&S0&B0 #!S1&!S0&A0;
DAT1 = S1&S0&D1 # S1&!S0&C1 #
 !S1&S0&B1 #!S1&!S0&A1;
DAT2 = S1&S0&D2 # S1&!S0&C2 #
 !S1&S0&B2 #!S1&!S0&A2;
DAT3 = S1&S0&D3 # S1&!S0&C3 #
 !S1&S0&B3 #!S1&!S0&A3;
```

Very high speed decoders can be built in the macrocells to form SRAM select signals, but do not use all of the macrocell product terms or the flip-flop in most cases. Decoders are formed as follows:

```
DEC0 = !A3&!A2&!A1&!A0;
DEC1 = !A3&!A2&!A1&A0;
DEC2 = !A3&!A2&A1&!A0;
```

Registers

Simple registers are formed as follows:

```
A:= DATAINPUT;
A.CLK = CLOCK;
A.RST = RESET;
```

This describes a D-type flip-flop with its input tied to a signal named DATAINPUT, its clock tied to a signal called CLOCK, and its reset input tied to a signal called RESET.

Shift Registers

Cascading registers results in a shift register as follows:

```
A:=DATAINPUT;
B:=A;
C:=B;
D:=C;
A.CLK = CLOCK;
B.CLK = CLOCK;
C.CLK = CLOCK;
D.CLK = CLOCK;
A.RST = RESET;
B.RST = RESET;
C.RST = RESET;
D.RST = RESET;
```

This shift register uses four macrocells. If the signals designated A,B,C,D are declared as outputs, they will appear somewhere at the pins of an XC9500 device. If A,B,C, and D are declared as nodes (internal points), the software implements them within the macrocells.

Counters

Counters can be built in a number of ways. The most efficient method is to have the macrocell flip-flops configured as T-type flip-flops. The following equations form T-type flip-flops; they add logic to load, hold, increment, and clear the flip-flops. Note the compact vector notation:

```
module Tcount
title '4 bit counter with load and clear'
D0..D3pin;
Q3..Q0pin istype 'reg_T';
CLK, I0, I1pin;
Data = [D3..D0];
Count = [Q3..Q0];
Mode = [I1,I0];
Clear= [0,0];
Load = [1,0];
Inc= [1,1];
equations
Count.T = ((Count.q+1) & (Mode == Inc)
# (Data)& (Mode == Load)
# ( 0 )& (Mode == Clear))
$ Count.q
Count.C = CLK;
end
```

Comparators

Comparators are easily handled by the XC9500 macrocell. Single bit comparators do not use all available macrocell product terms, and therefor a more efficient method is to implement four bits at a time, to generate multiple compares per macrocell:

```
COMP = !B1&!B0&!A1&!A0 + B1&!B0&A1&!A0
!B1&B0&!A1&A0 + B1&B0&A1&A0
```

Several COMP signals can be gated together to detect equality across larger groups of bits. Each group of four bits uses 4 function block inputs and several four-bit comparisons can occur per function block. Another macrocell then forms the composite function of all the bit comparisons, as needed. Figure 4 shows this technique expanded to a 10-bit comparator which is commonly used on the most significant address lines of a 32-bit microprocessor's address lines.

Parity

Similar to comparisons, parity can be calculated with multiple data bits per macrocell. The first three bits are calculated using four product terms ORed together. This result is then delivered to the macrocell Exclusive-OR gate where a fourth data variable is introduced.

Latches

Occasionally, designers need a transparent latch within an XC9500 device. This latch is formed by feeding the macrocell combinatorial logic back upon itself as shown in the following equation:

```
Q = ENA&DATA + !ENA&Q + Q&DATA;
```

The signal Q&DATA is included to make the Q output glitch free.

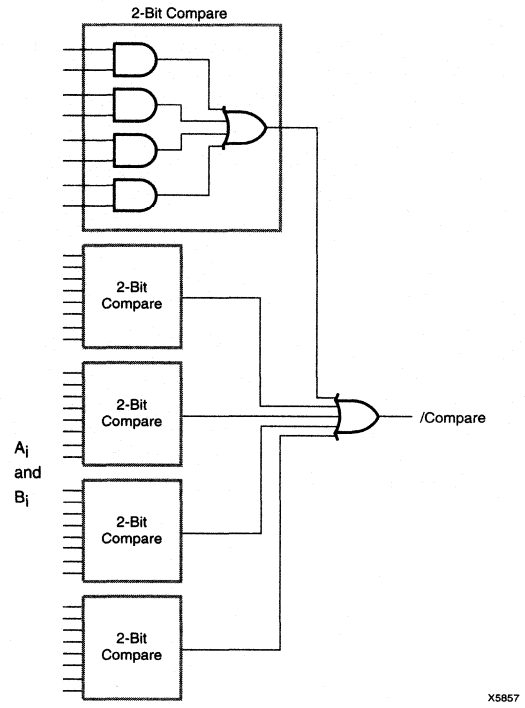


Figure 4: 10-Bit Comparator

Practical Considerations for XC9500 Designs

By following a few simple rules, XC9500 devices can easily interface with systems using 3.3 volt and 5 volt devices. Also, these devices behave much better if standard high performance printed circuit board techniques are used (as with all high-speed devices) so a small checklist is provided here for those rules.

Mixed Voltage Operation

XC9500 CPLDs support mixed voltage systems combining both 3.3 volt and 5 volt components as shown in Figure 6. The XC9500 family contains both logic and level shifting functions in a single programmable device. This eliminates the need for discrete level translation buffers. The XC9500 devices feature split power supply rails. The internal core logic always runs at 5 volts for the fastest possible performance. The output buffers can be powered by either 5 volts or 3.3 volts by connecting the I/O V_{CC} to a 3.3 volt or 5 volt supply. True TTL compatibility allows XC9500 CPLDs to drive and be driven by any combination of 3.3 and 5 volt logic without any performance penalty, even when the I/O V_{CC} pins are powered by 3.3 volts.

The XC9500 I/O structure is shown in Figure 5. Input protection diodes are connected to the internal 5 volt power supply rail and not to the output buffer supply rail. This allows the input to withstand a maximum voltage of >5 volts, even when the I/O power pins connect to 3.3 volts. Since both output transistors are N-channel devices, there is no parasitic diode to be forward biased if the output is in a 3-state condition and a 5 volt device is driving the XC9500 I/O pin. Therefore, the device can operate on a bus that includes both 3.3 volt and 5 volt devices.

Because the input protection circuitry is powered by the 5 volt core logic supply, the device pins should not be driven externally until the 5 volt V_{CCINT} supply is greater than 3 volts. In mixed 3.3/5 volt systems, where other 3.3 volt logic may be driving XC9500 devices, this requirement can be easily met by powering the 3.3 volt supply at the same time (or after) the 5 volt supply.

XC9500 devices are TTL-compatible with 3.3 and 5 volt logic as shown in Figure 7. The 5 volt TTL logic input thresholds are VIH = 2.0 V and VIL = 0.8 V. XC9500 CPLDs drive HIGH greater than 2.4 volts and LOW below 0.4 volts at the rated output drive currents, with at least 400 mV noise margin.

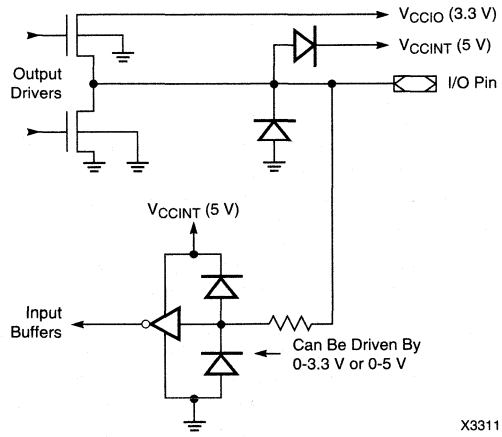


Figure 5: XC9500 I/O Architecture

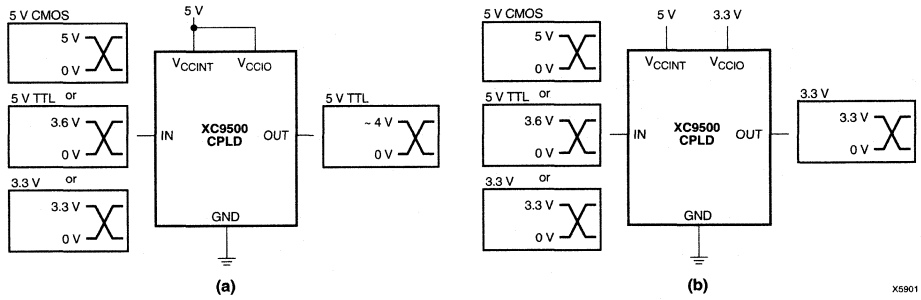


Figure 6: Typical Mixed Voltage System

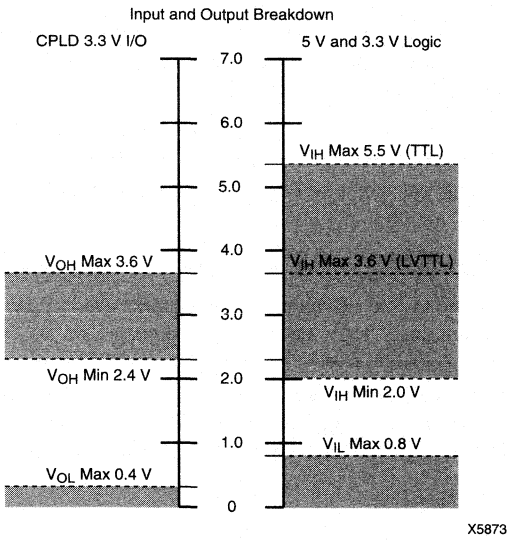


Figure 7: Driving 3.3 volt and 5.0 volt Components

High Speed Design Considerations

XC9500 CPLDs are offered with pin-to-pin delays as fast as 5 ns, and the actual speed may be faster. Therefore, additional care should be taken to minimize noise so that adjoining devices will operate properly.

Many high speed designs also require high current drive outputs for handling capacitive loads. XC9500 CPLDs provide 24 mA drivers to eliminate the need for additional buffering and therefore the designer needs to manage the total current being switched to minimize possible ground rise problems.

As with other high speed logic devices, XC9500 CPLDs should use low inductance capacitors located as close as possible to the V_{CC} and GND pins on a PC board. Care should be taken to mount the devices so that the PC board interconnect traces are as close as possible to the target signal destinations.

PC Board Layout Checklist:

Complying with the following checklist assures a successful design with XC9500 CPLDs:

1. Tie unused inputs to ground.
2. Locate XC9500 CPLDs near the devices they drive (or are driven by) to minimize transmission line effects.
3. Use wide spacing between fast signal lines (particularly clocks) to minimize crosstalk.
4. Place power pins (V_{CC} and GND) on separate printed circuit board planes. Fast signals should reside on a different plane.
5. Decouple the device V_{CC} input with a 0.1 µf capacitor. Directly connect each V_{CC} pin to the nearest ground plane. Low inductance, surface mounted capacitors are recommended.
6. Decouple the printed circuit board power inputs with 0.1 µµf ceramic (high frequency) and 100 µµf electrolytic (low frequency) filter capacitors.
7. Connect all device ground pins together.
8. Avoid using sockets to attach XC9500 CPLDs to the PCB. Direct soldered connection minimizes inductance and reduces ground rise. XC9500 CPLDs are specifically designed for direct PCB attachment.

Managing Ground Rise

Designers must also be aware of additional factors that can affect the performance of fast, high-current drive systems. Possible voltage rise on device ground pins can affect the driven output levels and be sensed by the switching CPLD.

Figure 8 shows how ground rise is typically observed. In this setup, multiple outputs are switched with a control variable, while one output is constantly being driven low and observed.

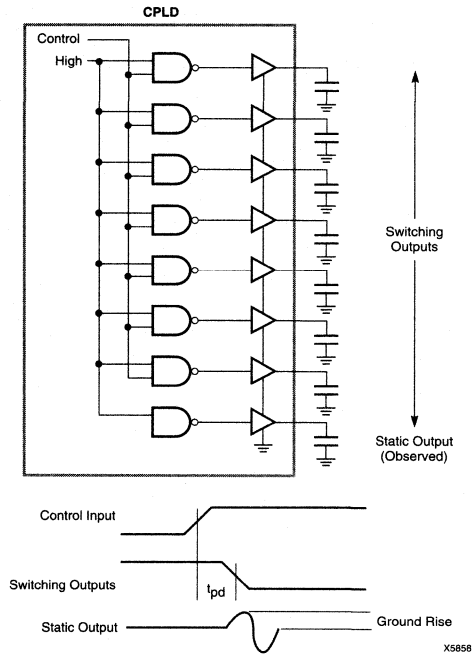


Figure 8: Ground Rise Test

As the multiple outputs switch, their in-rushing current converges at the ground pins of the device. Lead impedance causes the reference ground to develop a voltage higher than that which occurs before switching. The result is that

the static output being observed also develops an observable voltage swing.

All digital ICs have this property. No harm is caused to the system unless the voltage swing on the static output is capable of switching another circuit down the line. Problems can occur if the voltage swing is excessive. This effect is particularly significant if the static (quiet) signal is attached to another circuit's clock input.

Two factors contribute to this ground rise. First, the amount of capacitive load being driven is important because charge on this capacitance is the source of the in rushing current. Second, the number of simultaneous switching outputs is a factor since each switching output adds to the total capacitance being discharged.

XC9500 devices are in symmetric packages with multiple ground pins. However, some designs may need more grounding, and therefore the XC9500 family includes user programmable ground pins that allow the device I/O pins to be configured as additional grounds. Tying programmable ground pins to the external ground connection reduces system noise. The Xilinx XACT*step* software can be used to connect unused macrocell outputs to ground.

The following checklist will help reduce unnecessary ground noise:

1. Only connect the essential outputs to I/O pins. Intermediate shift register bits and counter bits that need not drive outputs should remain buried.
2. Minimize the number of outputs switching simultaneously.
3. Two global clock or GTS inputs can be managed by delaying one of the signals to gain signal skew.
4. Using additional ground pins can lower ground rise effects. Unused pins configured as grounds can be tied directly to the PC board ground plane. This splits the current driven into heavily loaded ground pins and lowers the voltage rise.
5. Signal skewing can also reduce ground rise. This can be achieved by mixing ordinary and fast slew rate outputs. Only assign the fast slew rate to signals that require it.

Conclusion

By using the techniques described in this application note, designers can achieve the highest-performance logic using the XC9500 family. The XC9500 family datasheet contains additional descriptions of the important system features.

Pin Preassigning with XC9500 CPLDs

January, 1997 (Version 1.0) Application Note

Summary

This application note describes the planning required for successful pin preassigning and gives a detailed example.

Xilinx Family

XC9500

Introduction

Reducing time to market is critical in today's highly competitive marketplace, and designers often need to prototype their products as swiftly as possible. Because PC board production is often the slowest part of the development process, it is often advantageous to begin PC board layout before the CPLD designs are complete. This requires designers to preassign device pins.

The flexibility of the XC9500 architecture permits pin preassignments, and a high degree of success can be achieved by following a few simple guidelines. The architecture also allows future edits to the design while maintaining the existing pinout. However, some design planning is necessary.

Many designs can be partitioned into datapath and control logic portions. Datapath operations are regular and bit-oriented, maintaining the relative positions of the bits. Datapath operations tend to retain their functionality after they are designed.

Control operations change as timing is adjusted during the design cycle while specifications are altered. These operations evolve as a design progresses. Control operations are much less structured than data operations and therefore the design strategy is different.

Obtaining a reasonable estimate of the design's logic requirements for both control and datapath operations is the key to preassigning pinouts. Once an estimate is obtained, pins are assigned with a high degree of confidence that the function block driving the pins will have the available resources to connect the logic needs of the corresponding macrocells. Because the design needs for datapath and control operations are different, the process of estimating their initial requirements will also differ.

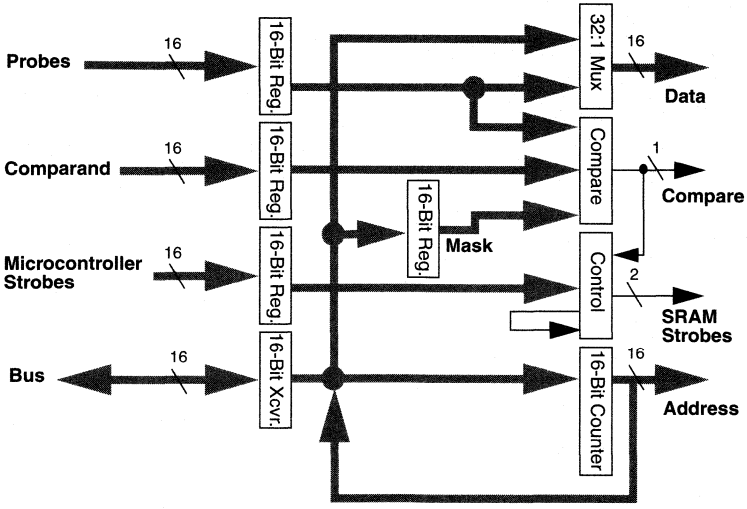


Figure 1: Pin Preassignment Design Example

XC9500 Routing Resources

XC9500 CPLDs combine a locally efficient logic block with a globally flexible interconnect structure to provide ideal connectability for a very large spectrum of designs. The key qualities of the logic block are:

- 36 input signals presented to the logic block.
- Automatic allocation of product terms as needed within the function block. The average is 5 product terms per macrocell, but up to 15 are easily obtained and up to 90 can be used when needed.
- Formation of efficient counters, multiplexers, shifters, and parity circuits with an efficiency of one macrocell or less per bit. The remaining logic is available for use by other functions.

The key properties of the interconnect structure are:

- Any input pin connects to any function block with constant high speed across the entire device.
- Any macrocell output can connect to its own or any other function block with no restriction.
- Macrocells can be internally bused with bit level independent 3-state control, to form internal data buses with global access to all logic blocks. (No other CPLD architecture offers this capability, which saves macrocell logic by using the routing resources to form multiplexers.)

These key features allow significant design changes to be made within CPLDs that are already attached to PC boards.

Datapath Estimation Guidelines

Datapath operations include storage, shifting, multiplexing, arithmetic, comparison, and boolean operations. For most designs, estimating the logic needs for a single bit is sufficient to quickly estimate the entire datapath. Table 1 gives the macrocell needs of various data operations per bit.

Table 1: Macrocell/Product Term Allocation

Data Operation	P-Terms Used
Shift Register	2
Counters	2-4
n:1 Mux	n
Adder	6
EX-OR	2
Storage registers	1

The Xilinx fitter software easily borrows the necessary product terms for denser functions from neighboring macrocells that have spare resources. As shown in Table 1, few functions require more than 4 product terms per macrocell. This suggests that all functions will easily fit into incremental macrocell units, except for the adder, which requires more than one macrocell, and the n:1 MUX which may need more product terms than a single macrocell can supply.

Control Path Estimation Guidelines

Control logic is more complicated to estimate than datapath logic because the logic needs vary and the timing is usually critical. Simple combinatorial control signals are easy to estimate, but state machines are not. However, one-hot encoded state machines, which are flip-flop rich with minimal input logic, may be used for estimation purposes.

One-hot state machine encoding works well for estimating pinouts because:

- No sophisticated state encoding is required.
- The input signals can be managed in a straightforward way.
- Performance degradation can be quantified at the initial estimation time.
- One-hot state machines are easily edited to add or delete states without upsetting the state encoding.
- One-hot solutions give a worst-case estimation, though encoded state machines usually produce logic that is more compact with no speed degradation (in CPLDs).

Editing one-hot machines without significantly altering the overall structure of the machine is easy. States may be inserted or deleted without major impact on the next state or output logic encoding. One-hot designs can also be implemented in either the Mealy or Moore versions as required. Figure 2 shows the basic one-hot encoding structure. The darker region of the figure shows additional logic added to alter the four state machine (A,B,C,D) to a five-state machine (A,B,C,D,E) without completely re-encoding.

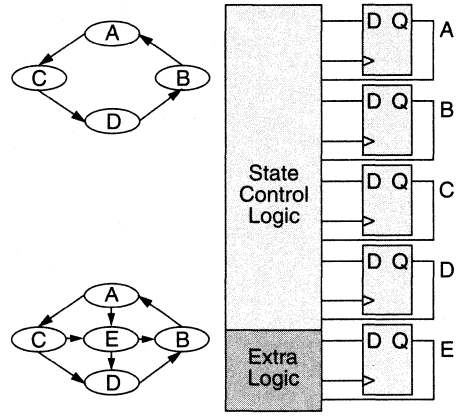


Figure 2: One-Hot Encoded State Machine Editing

Pin Preassigning Guidelines

The easiest and most successful method for preassigning pins is to do an estimated design and let the fitter software assign the pins. However, the following guidelines will help if a manual estimate is required:

- Partition the data and control sections, and estimate the needs of each section.
- Consider One-hot control unit designs to permit future editing and stable pin assignments.
- Use the following cell arrangement techniques:
  - Spread the outputs among all Function Blocks.
  - Within Function Blocks, spread macrocells evenly, with unused macrocells interleaved.
  - Assign outputs using common inputs to the same function blocks.
  - Don't use global pins (clocks, resets, 3-state, enables) for logic inputs or outputs unless the global pins are not needed.
- XC9500 CPLDs have common footprint packages for various density parts. By designing for the smallest capacity in a specific package footprint, there is always the option of a pin compatible density upgrade that matches the existing pinout. See Table 2 for package availability.
- Interleave output functions that need many product terms with functions that need few.

Manual Estimation Example

Figure 1 shows a block diagram of a proposed system that needs pinout assignments for early PCB delivery. The first step is to make a rough estimate of the required device capacity in order to target a particular XC9500 CPLD.

In this design, there are:

- 4 — 16-bit registers using 64 macrocells.
- 1 — 32:16 bit mux using 16 macrocells.
- 1 — 16 bit loadable counter using 16 macrocells.
- 1 — 16-bit transceiver using 16 pins (no macrocells).
- 1 — 16-bit maskable comparator using X macrocells.
- 1 — Control unit using Y macrocells.

To complete this estimate, more detail is required. First, the 16-bit transceiver will need 16 bi-directional pins and a global 3-state direction control signal. This partition will combine with the 32:16 mux and the 16-bit counter. One bit of this is shown in Figure 3.

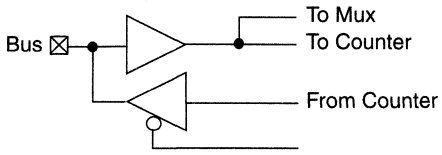


Figure 3: Transceiver Logic

The comparator will be created as an expansion of the diagram shown in Figure 4.

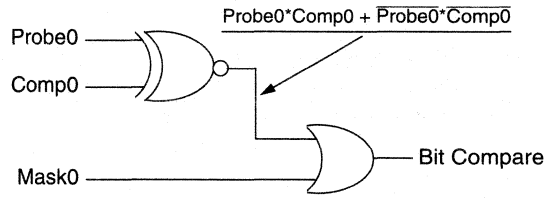


Figure 4: Bit Compare Logic

16-bit compares must be ANDed to create the overall signal "compare". Notice that each bit compare is easily computed with one bit per macrocell, but a total of 48 inputs are required. This exceeds the number of available inputs to an XC9500 function block. Also, 16 intermediate bit compares must be ANDed to permit ANDing a large number of such signals. Depending on the device, 12 bits may be assigned to a single FB using all FB inputs and still require making 4 more bit compares. This approach leaves 6 unusable macrocells remaining in an FB. A better approach is to assign 8 bit compares to an FB using 24 inputs with 12 inputs to support the remaining 10 macrocells. This is a natural way to interleave sparse resource macrocells (registers) with heavier resource macrocells (comparators). See Figure 5.

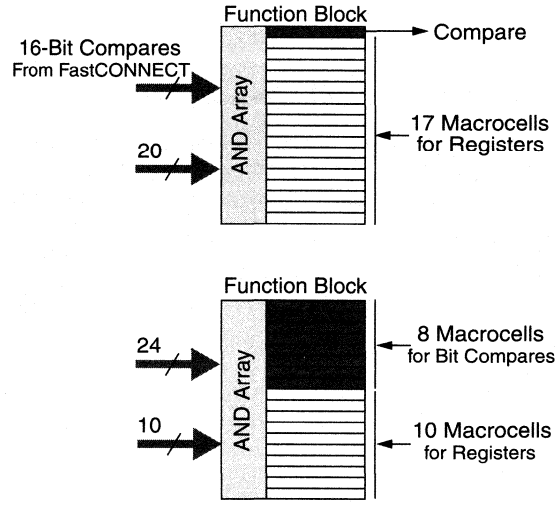


Figure 5: Possible Arrangement for Register and Compare Bits

Another version that may be more flexible is to interleave the registers with the compares in case additional variety (creeping elegance) may occur with the comparator. This approach makes single cascade groups of four product terms additionally available at each bit compare site. Com-

parator tally equals 17 macrocells. The 16-bit counter is loadable and synchronous. It needs 16 data inputs, 16 state feedbacks, and a load/count signal. This leaves only 3 available FB inputs, but clusters all macrocells within a single FB for faster speed.

The control unit will have a loadable counter used as an internal event timer. It will also require a simple state machine with about 4 states. Figure 6 estimates the Control Unit.

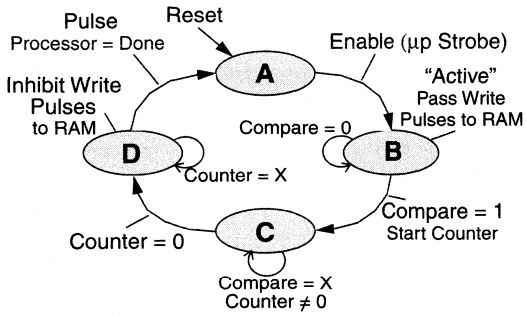


Figure 6: Control Unit State Machine

The loadable counter requires 33 inputs and 16 macrocells, so it makes sense to interleave this module among the registers which require few inputs and p-terms. As a one-hot state machine, the control unit state machine, with no simplification, requires 4 flip-flops and 4 inputs. Figure 7 gives an approximation of the design.

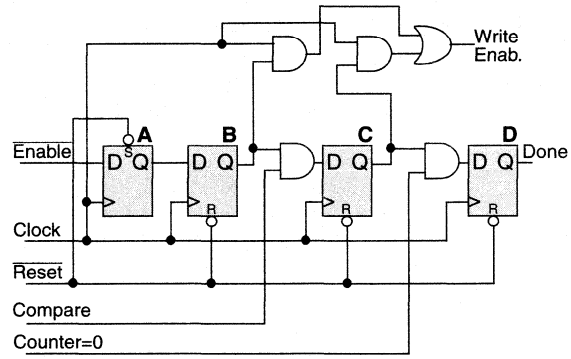


Figure 7: One-Hot State Machine Schematic

The tally can now be completed by adding in the maskable comparator which uses 17 macrocells and the control unit which uses 20 macrocells. The total is 96 + 37 = 133 macrocells. The smallest XC9500 device that can contain this is the XC95144. The I/O pins required are 108. Figure 8 suggests a reasonable pin assignment by associating macrocells to explicit pins in the PQ160 package.

Table 2: XC9500 Device Package Availability

Pins	XC9536	XC9572	XC95108	XC95144	XC95180	XC95216	XC95288	XC95432	XC95576
44	X								
84		X	X						
100		X	X	X					
160			X	X	X	X			
208					X	X	X		
352						X	X		
432								X	X

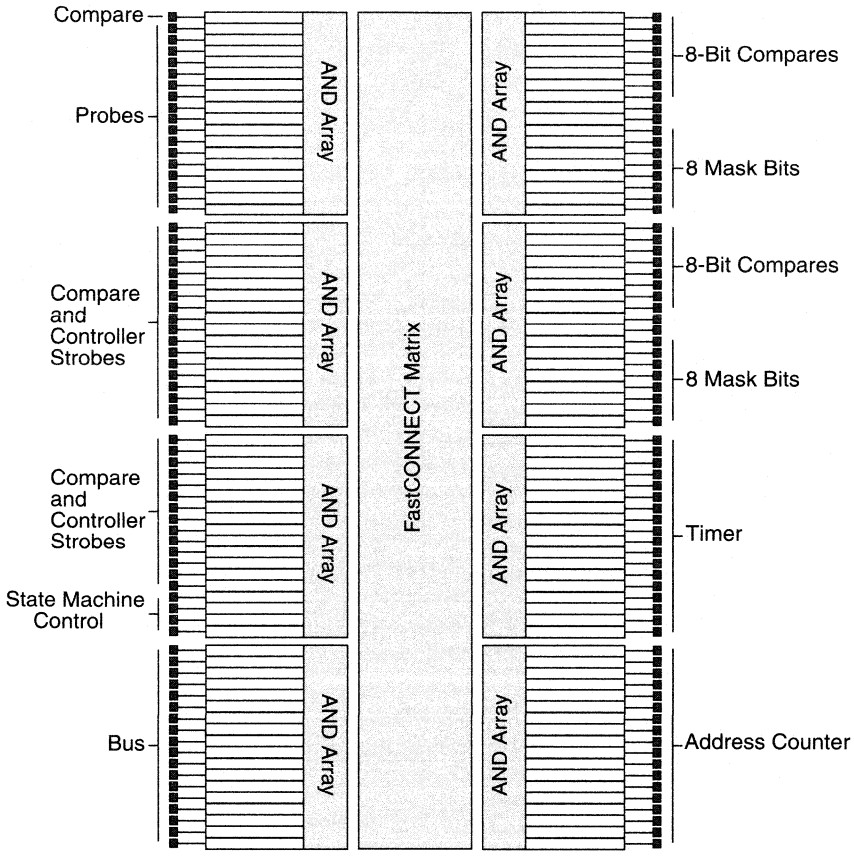


Figure 8: Pin Preassign Strategy for the XC95144

Conclusion

Pin preassigning is successfully achieved with XC9500 CPLDs by following simple guidelines. It is always recommended to do an estimated design and let the design software assign pins before committing to a printed circuit board.

The benefits of preassignment include faster time to prototype and ultimately faster time to market. Xilinx XC9500 CPLDs provide the speed and flexibility to make this goal a reality.

Design Migration with XC9500 CPLDs

January, 1997 (Version 1.0) Application Note

Summary

The advanced architecture of the XC9500 family, combined with consistent packaging options makes it easy to move an XC9500 design into a larger or smaller device and still keep the original footprint. This application note describes how to prepare designs for easy migration and provides examples demonstrating how to verify their functionality.

Xilinx Family

XC9500

2

Introduction

CPLD design requirements often change, even after PC boards are laid out. Therefore, it is often advantageous to move the modified design into a different device within the XC9500 family, a device that more efficiently meets the needs for macrocell resources. With a little planning, it is easy to migrate an XC9500 design within the family and still keep the original pinouts. Thus, expensive and time consuming PC board re-work can be avoided, decreasing time to market. In addition, moving to a smaller device when design requirements are reduced, reduces the overall component cost of the system.

The advanced design migration capability of the XC9500 family gives unprecedented flexibility for:

- **Maintaining Designs in the Field** - Significant changes can be made without redesigning PC boards.
- **Reducing Cost** - If the redesign uses less resources, it is possible to move to a smaller, lower cost device.
- **Prototyping with Larger Devices** - This can reduce product development time.
- **Using Available Devices** - Current CPLD inventory can be used more efficiently.

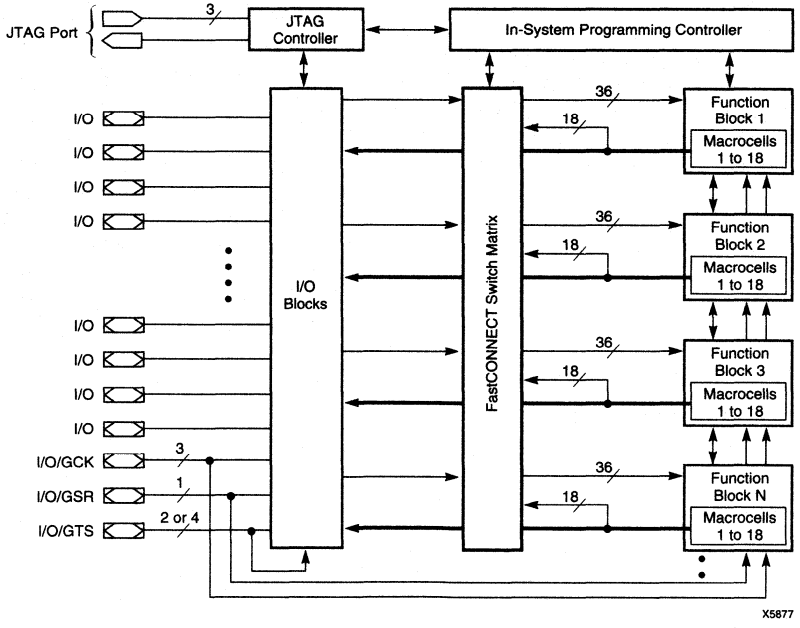


Figure 1: XC9500 Architecture

The XC9500 Architecture

Each device in the XC9500 family contains multiple uniform function blocks, connected by the FastCONNECT switch matrix, as shown in Figure 1. This architecture guarantees 100% routability, even for fully populated devices. This highly flexible routing capability along with the advanced features of the Xilinx XACT fitter software provide highly reliable pin locking; this is a key requirement for reliable design migration. The XC9500 advanced pin-locking capability allows significant design changes without reworking PC boards.

XC9500 Packaging

For the whole XC9500 family, all device types in the same package have the same pinouts. For example, an XC9572 die in a PC84 package has the same pinouts as an XC95108 in that same package. Therefore, migrated designs will fit the same footprint, they will function almost identically, and PC boards will not need to be modified. Package availability is shown in Table 1.

Design Migration Strategy

Moving an overcrowded design to a larger device is easy because the larger device will usually have an abundance of unused resources that make routing and pin-locking easier. Moving to a smaller device however, requires some previous planning. The following procedures show how to move designs into either a smaller or larger device.

Note1: These procedures only apply to different devices in the same package type. For example, upward design migration from an XC95108-PC84 design to an XC95144 cannot occur because the XC95144 is not available in the PC84 package. However it is possible to move an XC95108-PQ100 design into an XC95144 because the XC95144 is available in the PQ100 package.

Note2: These procedures and examples assume that the Xilinx XACT fitter, version 6.0.0 or later, is being used.

Migrating to a Larger Device

Moving a design to a larger device is very simple and straightforward. Usually this procedure is used when adding more functionality to an already full device or when using the larger devices in inventory.

1. Save the pinouts from the original, smaller, design. This is accomplished by selecting the "Lock the Pins" process in the XABEL-CPLD Project Navigator which creates a .GY2 (Pin-save) file that can later be used to specify pinouts.
2. Re-compile the original design, targeting the larger device, and using the original pinouts from the .GY2 file. To use the pinouts specified by the .GY2 file, highlight the "Fit Design" process in the XABEL-CPLD Project Navigator and click on "Properties"; "Fitter Options" is displayed. Select "Locked Pins" as the option for the "Pin Assignments" selection.
3. Review the Resource and Timing Reports generated by the fitter, to verify that the new design meets the specified pinout and timing requirements.

Migrating to a Smaller Device

Moving a design to a smaller device can also be accomplished, if the smaller device has enough I/O pins and macrocells to fit the design, and if the design is planned ahead. Usually this procedure is used if the smaller device is currently unavailable or if device resource requirements are likely to be reduced in the future.

Design migration into a smaller device is assured if the designer first targets the design to the smaller, harder to fit device, and allows the fitter to automatically assign pins. This will establish pinouts and timing based on the worst case device resource restrictions (the smaller device).

1. Compile the design, targeting the smaller device, without restricting pinouts. Save the pinouts into a Pin-Save file. This is accomplished by selecting the "Lock the Pins" process in the XABEL-CPLD Project Navigator which creates a .GY2 file that can later be used to specify pinouts. Verify that this design meets all of the functional and timing requirements.

Table 1: XC9500 Device Package Availability

Pins	XC9536	XC9572	XC95108	XC95144	XC95180	XC95216	XC95288	XC95432	XC95576
44	X								
84		X	X						
100		X	X	X					
160			X	X	X	X			
208					X	X	X		
352						X	X		
432								X	X

2. Re-compile the design, targeting the larger device, using the pinouts that were previously specified for the smaller device. To use the previous pinouts, highlight the "Fit Design" process in the XABEL-CPLD Project Navigator and click on "Properties"; "Fitter Options" is displayed. Select "Locked Pins" as the option for the "Pin Assignments" selection. Make sure this design meets all of the functional and timing requirements.
3. When both designs compile properly, and meet the design requirements, save the one targeted to the smaller device for later use.

Now, either the larger or smaller device can be used in the same socket on the PC board, and either design will function properly.

Design Migration Example

The following design example of a DRAM controller demonstrates the design migration procedure and shows how to compare the fitter reports to verify compatibility.

This design requires only 52 macrocells and therefore it will easily fit into an XC9572-7PC84. However, the design can also be targeted to an XC95108-7PC84 while maintaining both pinouts and performance.

This design, named "dram", was first compiled for the XC9572, allowing pinouts to be assigned by the fitter. After the design compiled successfully, the automatically generated pinouts were verified and saved in a Pin-Save file for later use. Then the design was re-compiled, targeting the XC95108, using the saved pinouts from the previous XC9572 compilation. The timing for both designs was determined to be acceptable and both designs were saved for later use.

These two designs were compiled using the Xilinx XACT-CPLD V6.0.0 fitter which generates two reports:

- **dram.rpt** - This report shows a resource usage summary and shows where each signal is mapped in the device. Primarily, use this report to verify that the pinouts match for both devices and that both devices have adequate resources to fit the design.
- **dram.tim** - This report shows design timing. Primarily, use this report to verify that the design performance is adequate for both devices.

The following report examples show selected comparisons that demonstrate the functional compatibility of the two designs; these reports are abbreviated for clarity.

Resource Summary Report Examples

The reports illustrated in Figure 2 show that both the XC9572 and the XC95108 have adequate resources to fit the DRAM controller design. The pinout diagrams for both designs were identical and are shown in Figure 3.

Timing Report Examples

The reports illustrated in Figure 4 and Figure 5 show selected differences in various timing parameters for the two devices. From this report comparison it can be seen that the timing for these two designs is almost identical. If this were a real design, the designer would need to verify that these minor timing differences are not significant.

The timing report example shows some of the timing parameters for the XC9572 device; a sample of the XC95108 timing parameters that do not match identically, are highlighted separately. Those parameters that match exactly are not highlighted.

```
ion XACT-CPLD-V6.0.0                          Xilinx Inc.
                              Fitter Report
   Design Name: dram
Fitting Status: Successful                    Date:  4-29-96, 12:58PM

*****************************  Resource Summary  *****************************

Design          Device            Macrocells     Product Terms    Pins
Name            Used              Used           Used             Used
dram            XC9572-7PC84      52 /72  ( 72%) 188/360 ( 52%)   67 /69   ( 97%)
```

Report 1

```
PIN RESOURCES:

Signal Type     Required     Mapped  | Pin Type            Used   Remaining
------------------------------------|-------------------------------------
Input         :    39          39   | I/O               :   61        2
Output        :    27          27   | GCK/IO            :    3        0
Bidirectional :     0           0   | GTS/IO            :    2        0
GCK           :     1           1   | GSR/IO            :    1        0
GTS           :     0           0   |
GSR           :     0           0   |
                   ----        ----
         Total     67          67

GLOBAL RESOURCES:

Signal 'CLOCK' mapped onto global clock net GCK1.
Global output enable net(s) unused.
Global set/reset net(s) unused.

POWER DATA:

There are 52 macrocells in high performance mode (MCHP).
There are 0 macrocells in low power mode (MCLP).
There are a total of 52 macrocells used (MC).

End of Resource Summary
```

Note: The Resource summaries for these two designs are almost identical, except for the added resources of the larger device.

```
XEPLD, Version XACT-CPLD-V6.0.0                Xilinx Inc.
                              Fitter Report
   Design Name: dram
Fitting Status: Successful                    Date:  4-29-96,  8:45AM

*****************************  Resource Summary  *****************************

Design          Device            Macrocells     Product Terms    Pins
Name            Used              Used           Used             Used
dram            XC95108-7PC84     52 /108 ( 48%) 188/540 ( 34%)   67 /69   ( 97%)
```

Report 2

```
PIN RESOURCES:

Signal Type     Required     Mapped  | Pin Type            Used   Remaining
------------------------------------|-------------------------------------
Input         :    39          39   | I/O               :   61        2
Output        :    27          27   | GCK/IO            :    3        0
Bidirectional :     0           0   | GTS/IO            :    2        0
GCK           :     1           1   | GSR/IO            :    1        0
GTS           :     0           0   |
GSR           :     0           0   |
                   ----        ----
         Total     67          67

GLOBAL RESOURCES:
Signal 'CLOCK' mapped onto global clock net GCK1.
Global output enable net(s) unused.
Global set/reset net(s) unused.

POWER DATA:
There are 52 macrocells in high performance mode (MCHP).
There are 0 macrocells in low power mode (MCLP).
There are a total of 52 macrocells used (MC).

End of Resource Summary
```

Figure 2: A Comparison of Resource Summary Reports

```
           *************************** Device Pin Out ***************************

           Device : XC9572-7PC84


                   C                                                                       C
           D       L           D       D       D               M                           A
           Q   A   O   G   B   Q   A   Q       Q   M   A   A   A           M   V   A   B   C
           M   1   C   N   E   M   1   M   A   M   A   3   1   1   N   A   A   C   2   E   H
           3   7   K   D   7   1   9   0   6   2   9   0   4   1   A   9   7   C   2   0   E
          -----------------------------------------------------------------------------------
         /11  10   9   8   7   6   5   4   3   2   1  84  83  82  81  80  79  78  77  76  75 \
    BE1 | 12                                                                              74 | A8
   DQM4 | 13                                                                              73 | VCC
    A27 | 14                                                                              72 | MA3
   BRDY | 15                                                                              71 | A4
    GND | 16                                                                              70 | MUX
    A13 | 17                                                                              69 | ADS
    A12 | 18                                                                              68 | DQM6
    MA0 | 19                                                                              67 | A20
   DQM7 | 20                                                                              66 | A29
  WRITE | 21                                  XC9572-7PC84                                65 | A3
    VCC | 22                                                                              64 | VCC
    MA4 | 23                                                                              63 | DQM5
   N.C. | 24                                                                              62 | BE4
    CAS | 25                                                                              61 | MA8
    MA2 | 26                                                                              60 | GND
    GND | 27                                                                              59 | TDO
    TDI | 28                                                                              58 | BE3
    TMS | 29                                                                              57 | BE5
    TCK | 30                                                                              56 | A31
    A21 | 31                                                                              55 | A16
     WE | 32                                                                              54 | A18
        \ 33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 /
          -----------------------------------------------------------------------------------
           M   B   M   R   A   V   A   B   A   G   M   A   A   C   R   A   G   M   N   B   A
           A   E   A   E   2   C   5   E   7   N   A   1   1   S   A   2   N   A   .   R   1
           6   2   1   F   3   C       6       D   1   0   1       S   8   D   5   C   D   5
                       R                           0                               .   Y
                       E                                                               _
                       S                                                               I
```

Figure 3: XC9572 Pinout Diagram (Identical for the XC95108)

Pinout Diagram

Figure 3 shows the pinout diagram for both reports, which were identical. Use this pinout diagram to verify that the pinouts are acceptable and identical for both designs.

```
Performance Summary:

Worst case Pad to Pad path delay          :          7.5ns (1 macrocell levels)
(Includes an external input margin of 0.0ns.)
(Includes an external output margin of 0.0ns.)
Pad 'A23' to Pad 'MA10'

Clock net 'CLOCK' path delays:

Worst case clock pad to output pad delay  :         14.0ns (2 macrocell levels)
(Includes an external input margin of 0.0ns.)
(Includes an external output margin of 0.0ns.)
Clock Pad 'CLOCK' to Output Pad 'MA10'                          (Fast Clock)

Clock to Setup path delay                 :         15.0ns (1 macrocell levels)
Clock to Q, net 'P0_BANKB0.Q' to DFF Setup(D) at 'CS_MC.D'      (Fast Clock)
Target FF drives output net 'CS_MC'

Clock to Pad path delay                   :         11.5ns (2 macrocell levels)
(Includes an external output margin of 0.0ns.)
Clock to Q, net 'MUX_MC.Q' to Pad 'MA10'                        (Fast Clock)

Pad to Setup path delay                   :         17.5ns (1 macrocell levels)
(Includes an external input margin of 0.0ns.)
Pad 'CLOCK' to DFF Setup(D) at 'CS_MC.D'                        (Fast Clock)
Target FF drives output net 'CS_MC'

Pad Ending at Clock Pin path delay        :          2.5ns (0 macrocell levels)
(Includes an external input margin of 0.0ns.)
Clock Pad 'CLOCK' to DFF Clock Pin at 'FCLKIO_0'                (Fast Clock)

Setup to Clock at the Pad                 :         15.0ns (1 macrocell levels)
Data signal 'CLOCK' to DFF D input Pin at 'CS_MC.D'
Clock pad 'CLOCK' to DFF Clock Pin at 'FCLKIO_0'                (Fast Clock)

                     Minimum Clock Period: 15.0ns
                Maximum Internal Clock Speed: 66.6Mhz
                       (Limited by Cycle Time)

            Estimated Maximum External Clock Speed: 66.6Mhz
                       (Limited by Cycle Time)
```

Figure 4: Timing Performance Summary (Identical for both devices)

Timing Performance Summary

Figure 4 shows the Timing Performance Summary. Use this part of the Timing Report to verify the overall timing for the design. This gives a quick overview of timing and reveals any major timing differences. The two Performance Summaries are identical for both designs in this example, and therefore only one report is shown here for clarity.

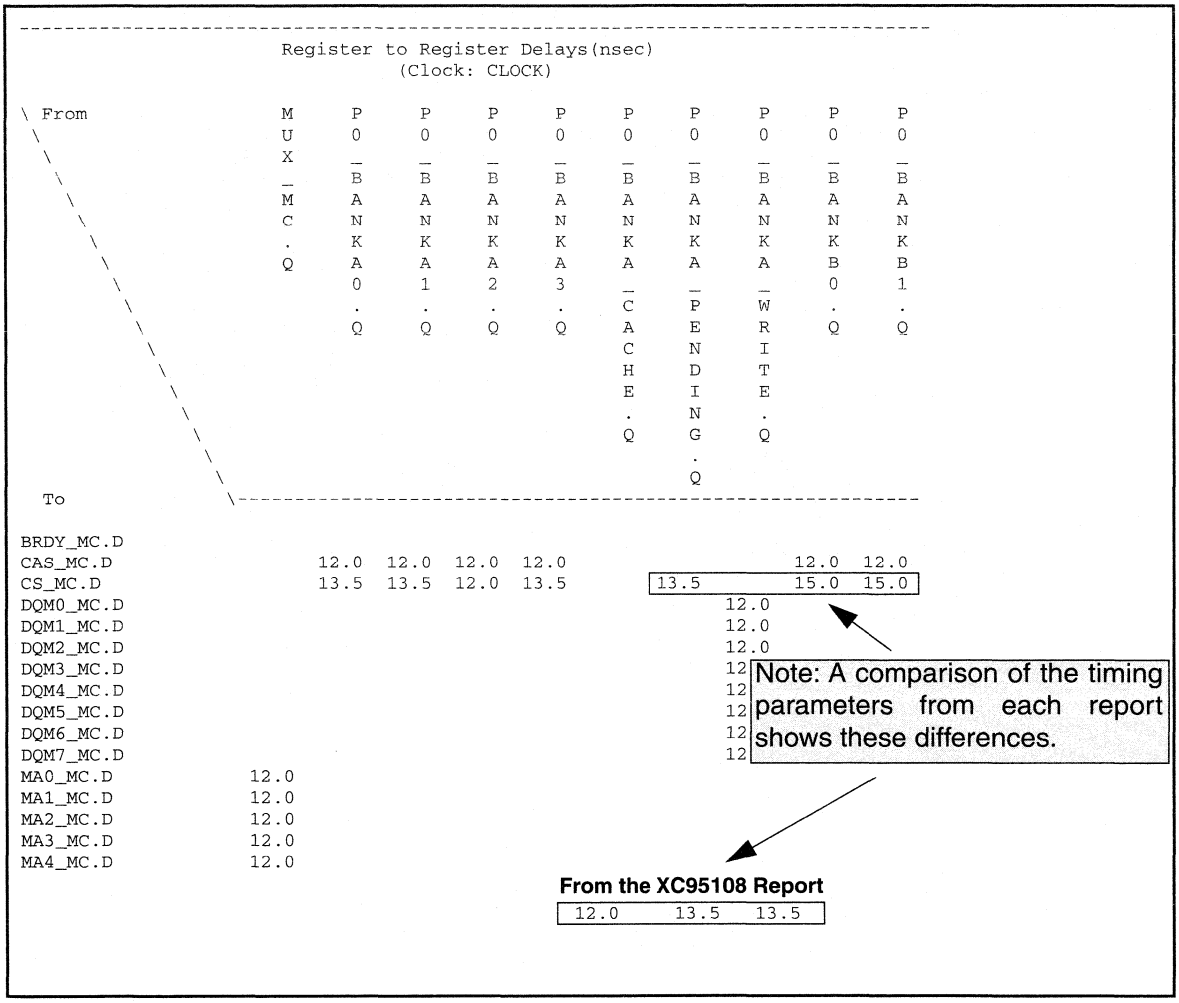


Figure 5: Timing Report Differences (XC9572 Report Shown Here)

Timing Report Differences

Figure 5 shows some of the timing differences from the two Timing Reports. Typically, the designer would compare each timing parameter from the two reports to be sure that all critical timing is within specification limits. In this example the timing difference is only 1.5ns for a register-to-register delay, which is well within the timing requirements for this design.

There were other timing differences between the two Timing Reports. However, the differences were very minor and are not illustrated here.

Conclusions

The flexibility of the XC9500 architecture, combined with the advanced features of the Xilinx fitter, give designers the capability to easily migrate designs into larger or smaller devices, while retaining the original pinouts. In addition, the timing differences between migrated designs are usually insignificant. This feature gives maximum design flexibility and allows designers to modify designs even after PC boards are laid out.

Using ABEL with Xilinx CPLDs

January, 1997 (Version 1.0) Application Note

Summary

This application note provides a basic overview of the ABEL language and gives examples showing how to use ABEL to fully utilize the specific features of Xilinx CPLDs.

Xilinx Families

XC9500, XC7300

Introduction

ABEL (Advanced Boolean Expression Language), combined with the Xilinx fitter software, provides a complete behavioral development environment for entering, simulating, and implementing designs for Xilinx CPLDs. And, because ABEL was developed specifically for programmable logic devices, it provides several important features that support the Xilinx CPLD architecture.

ABEL Language Structure

ABEL designs are organized into modules. Each module contains at least one set of declarations, logic descriptions, and an optional set of test vectors. Most designs are completely specified in a single module. However, using the hierarchal feature found in ABEL 6, multi-module designs can also be specified. Figure 1 shows the distinct sections of code that are necessary to completely specify a design.

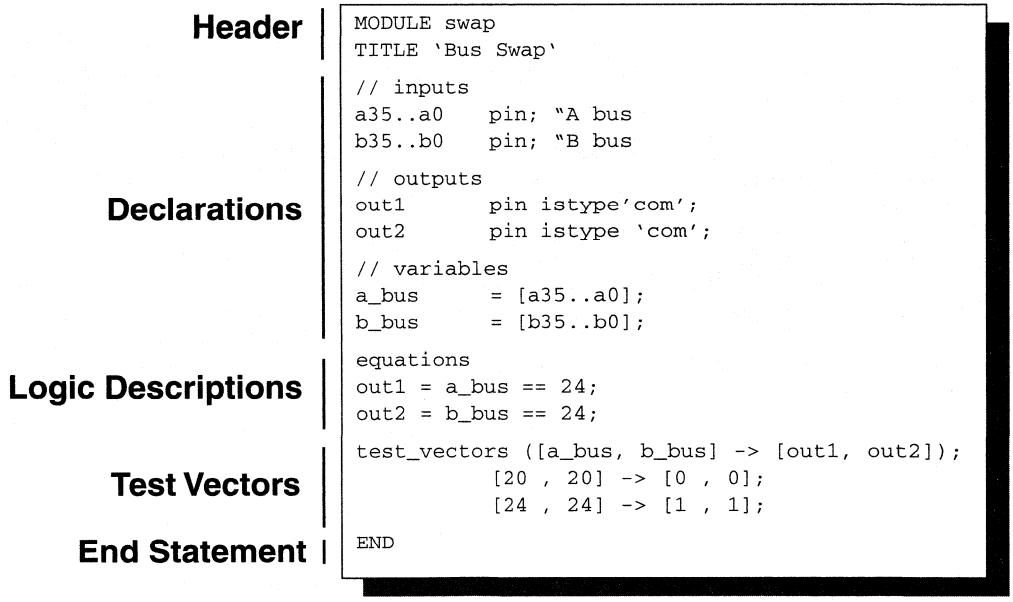


Figure 1: ABEL Module Structure

The Header Section

The module name is specified by the keyword **Module.** An optional **Title** may also be used after the module name to further describe the design. The double quote or the double slash can also be used to add comments. At the end of each module definition the keyword **End** is used to specify the end of the design.

For example:

```
MODULE mydesign

TITLE  'version 1 of mydesign'
"Additionally, comments can be added using
"the double quote
// or the double slash

END
```

The Declarations Section

Declarations are used to define constants, signals, and sets, and to pass property statements to the fitter for controlling device specific features which are not directly supported by ABEL. The various elements contained in the Declaration section are:

- Constants — Constants are declared by assigning a value to a constant name.
- Input pins — specified with the **pin** keyword without types.
- Output pins — specified by the **pin** keyword and either registered or combinatorial by using the **istype** keyword.
- Nodes or buried logic — specified by the **node** keyword and can be either registered or combinatorial by using the **istype** keyword.
- Arrays of signals — declared by ending the signal name with a number and using the double period (..) syntax.
- Sets — declared to make the ABEL code easier to read and write by replacing long redundant signals with a single reference. In addition, higher level operations performed on sets allow very powerful and complex designs to be specified very quickly.

For example:

```
"Declaring constants
On = 1;
Off = 0;

"Declaring input pins
my_input, my_clk pin;

"Declaring output pins
my_combinatorial_output pin istype 'com';
my_registered_ouput     pin istype 'reg';

"Declaring nodes
my_combinatorial_node  node isypte 'com';
my_registered_node     node istype 'reg';

"Declaring a 5-bit array of nodes
Count4..Count0  node istype 'reg';

"Declaring a set to reference all 5-bits
Counter = [Count4..Count0];

"Property statements
xepld PROPERTY 'fast on';"Sets all outputs
                          "to fast slew.
```

Note that in the declarations section, the order in which constants and sets are defined is important. If constant X is defined with constant Y, for example, then constant Y must be defined first.

The Logic Description Section

The logic description section specifies the functions of the design. This can be done in three ways: Equations, Truth Tables, and State Diagrams. Equations are useful for designs with regular patterns such as counters or multiplexors. Truth Tables are a good entry method for designs that do not have regular patterns, such as a 7-segment LED decoder. State Diagrams are useful for specifying designs with complex state machines.

Following the Declarations section, the design section is delimited by using the keyword **Equations**, **Truth_Table**, or **State_Diagram.** State_diagrams and truth tables using sequential logic will need an accompanying equations section to define clock signals as well. These keywords must be used when switching between the three design methods.

Equations

Equation design entry primarily consists of assignment statements. These can be combinatorial assignments (=), or registered assignments using the delay operator (:=). All registered equations using the delay operator (:=) will behave as being implemented as an edge triggered flip-flop. Therefore, they must also have a clock associated with the signal name. This is done by using the **.clk** dot extension.

For example:

```
my_registered_node := my_input;
my_registered_node.clk = my_clk;
```

This is logically equivalent to describing the same logic with detailed dot extensions as follows.

```
my_registered_node.d = my_input;
my_registered_node.clk = my_clk;
```

Using equations to specify a design is similar to programming in other languages, except the context of the equations are evaluated in parallel rather than sequentially. In

the following example, the order in which the equations are written is only important in programming languages.

In ABEL, all of the equations are evaluated concurrently, thus, the order in which they are presented is *not* important.

For example, in a normal programming language such as C, the code:

```
x = x + 1;
total = total +x;
```

is not the same as

```
total = total + x;
x = x+1;
```

However, in ABEL

```
x := x+1;
total :=  total + x;
```

is the same as

```
total := total + x;
x := x + 1;
```

In order to process information sequentially in digital logic, registers are used. In the previous example, note that the assignments are made with a ":=" . This means that in order for **x** or **total** to actually change values, a rising edge clock signal must be received by the register. The implementation for this design would look something like the following (line numbers are added to accompany the following explanation):

```
1  MODULE example1
2
3  my_clock pin;
4  x7..x0 node istype 'reg';
5  total7..total0 pin istype 'reg';
6
7  x = [x7..x0];
8  total = [total7..total0];
9
10 @carry 4;"Limit the carry chain to
           "4-bits
11
12 EQUATIONS"Signals the beginning of
           "an equation section
13 [x, total].clk = my_clock;
           " Set the clock signals for
           " registers to my_clock
14
15 x := x+1;  "This will implement an
              "counter counting by 1
16 total := total + x;
             "Note this is an adder that
             "uses the @carry directive
             "to implement the 8 bit
            "adder with two 4-bit adders
17
18 TEST_VECTORS ([my_clock] -> [x, total])
19
20 [.C.] -> [ 1 , 0];
21 [.C.] -> [ 2 , 1];
22 [.C.] -> [ 3 , 3];
23 [.C.] -> [ 4 , 6];
24 [.C.] -> [ 5 , 10];
25 [.C.] -> [ 6 , 15];
26 [.C.] -> [ 7 , 21];
27
28 END
```

The beginning of this design (as for all ABEL designs) defines the module name. Following the module name are the declarations of the module. On line 3, the clock input is defined. On line 4, the nodes that will contain the information for **x** are declared as an 8-bit register. Line 5 declares the output pins for **total**, an 8-bit register. On line 7 and 8, the 8 bits are renamed to a single variable name, allowing the equations to be written quickly and clearly. Line 10 is an ABEL compiler directive which limits the lookahead carry chain of the adder to 4 bits. On line 12, the **EQUATIONS** keyword signifies the end of the Declarations section and the beginning of the logic descriptions.

Every register must have a clock associated with it and in this design, line 13 specifies that the **x** and **total** registers are clocked by the input **myclock**. The actual logic assignments on lines 15 and 16 determine how the registers are affected when **myclock** goes high. At the rising clock edge, **x** (after clock) will get the value of **x** (before clock) **+ 1**, and **total** (after clock) will get the value of **total** (before clock) **+ x** (before clock). Figure 2 shows a block diagram of the circuit described by this code.

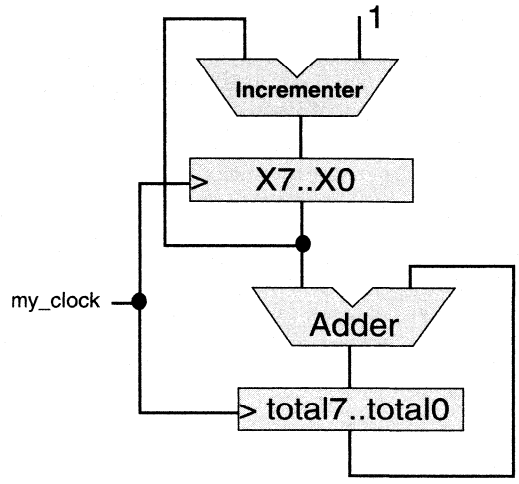


Figure 2: Block Diagram

Test Vectors

Test vectors are also included at the end of this design. These test vectors show the expected values of the output for each rising clock signal, and are extremely useful for verifying the design. Test vectors are also included in the JEDEC programming file which can be used in the XC9500 family to perform an **INTEST operation** through the **JTAG** port.

Dot Extensions

Dot extensions, illustrated in Figure 3, give more control over the implementation of the design. Dot extensions such as **.AP** or **.AR** are used to specify asynchronous preset and asynchronous reset for flip-flops. Other common dot extensions are **.OE** used to specify the output enable, and **.PIN** used with bi-directional signals. For a complete set of supported dot extensions, please refer to the documentation for the particular version of ABEL being used.

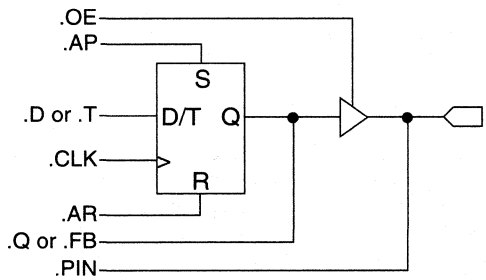


Figure 3: Directly Supported ABEL Dot Extensions

Logical Operators

Logical operators allow the user to manipulate signals logically and can be applied to both signals and sets. When applied to sets, they are applied bitwise, and the sets must be the same size.

The operators are shown in Table 1.

Arithmetic Operators

Arithmetic and relational operators can be used on sets to quickly generate adders, counters, and comparators. Large arithmetic functions, such as adders and magnitude comparators, will generate very wide equations to implement the carry look ahead signals. In order to control the width of these lookahead carry equations, the compiler directive, **@carry**, can be used. For example, **@carry 4**, would limit the carry chain to 4-bits. An 8-bit adder therefore, would be implemented as two 4 bit adders. Each adder would perform the carry lookahead in parallel for its own four bits. However, a carry signal will be generated by the lower four bits that will be cascaded to the higher order adder. Typically, carry chain lengths of 3 or 4 is a good trade-off between speed and density.

Relational Operators

Relational operators are also used for generating the boolean result for conditional equations. These are useful for making the comparisons used in address decoding. Relational operators are used in conjunction with the conditional equation, **when...then...else** statement. Using conditional equations simplifies the task of building components with control signals.

For example, using operators:

```
"Using the logical OR operator.
my_reg_output := my_input # my_reg_node;
my_reg_output.clk = my_clock;

"Using the arithmetic operator
sum := a + b;
sum.clk = my_clock;

"Using the relational operators with
"conditional equations
WHEN (a != b) THEN
     c := my_input;
ELSE
     c := my_reg_node;
c.clk = my_clock;
```

Conditional equations allow the designer to make certain assignments depending upon the result of a relational operator. In the previous example, **c** is assigned **my_input** if **a** did not equal **b**. Otherwise, **c** will be assigned to the value stored in **my_registered_node**. Multiple assignments are made by containing them within the **{}** symbols.

The **else** statement does not need to be included in every conditional equation. When there are multiple conditions, they can be specified one at a time, however, make sure all possibilities are covered, otherwise unexpected behavior may occur in the design. For example:

```
"Using multiple WHEN without ELSE
"statements, the description of a mux
"is done by conditionally setting Output
"to Data_A or Data_B. Note the {} are used
"to make multiple assignments for each
"conditional.
Output2.clk = myclock;
WHEN (select == 0) THEN
{Output = Data_A;
Output2 := ValueA;}
WHEN (select == 1) THEN
{Output = Data_B;
 Output2 := ValueB;}
WHEN (select == 2) THEN
{Output = Data_C;
 Output2 := ValueC;}
"Note, select may also equal 3 and not
"defining the behavior for 3 will make
"the condition a don't care.
```

Table 1: ABEL Operators

Logical		Arithmetic		Relational	
&	AND	-	Twos complement	==	Equal
#	OR	A-B	Subtraction	!=	Not equal
!	NOT	A+B	Addition	<	Less than
$	XOR	<<	Shift left	<=	Less than or equal
		>>	Shift right	>	Greater than
				>=	Greater than or equal

Using Truth Tables

Truth table design entry consists of specifying the input signals and the responding output signals, which can be registered or combinatorial. Truth tables are handy when specifying designs with irregular patterns, such as a 7-segment LED decoder.

A truth table starts with a header specifying all of the input and output signals. A transition specified with the **->** syntax is a combinatorial transition, while a **:>** specifies a registered transition. Registered truth tables must also include a clock equation to specify the clock input to the register. For example:

```
"Syntax for defining a truth table
TRUTH_TABLE ([ my_input ] ->
[my_combinatorial output] :>
[my_registered_output])

"A mux described with a truth table

EQUATIONS   "Note that we define the clock
            "with an equation first.
Output.clk = my_clock;

"Then, we specify the truth table input
"and outputs.
TRUTH_TABLE
([select]:> [ Output ])
  [ 0 ]  :> [ Data_A ];
"If select is 0,then Output gets Data_A
  [ 1 ]  :> [ Data_B ];
"If select is 1, then Output gets Data_B
```

Entering Test Vectors

Test vectors can optionally be added to perform a functional test of the logic descriptions. For the XC9500 family, they are included in the JEDEC file which can also be used with the JTAG INTEST feature to test the functionality of the physical device in the system.

The **TEST_VECTORS** keyword signifies the end of the previous section and the start of the test vectors that are used to functionally simulate the design. Test vectors are entered in the same format as truth tables, and special signals are defined to help make the process easier. The three most common are **.C.** , **.Z.**, and **.X.**. The **.C.** signal represents a clock that starts from a logical low, goes to a logical high, and returns to a logical low. This is more convenient than entering three test vectors for each clock pulse. The **.Z.** is used to represent signals that are 3-stated, and the **.X.** signal represents a don't care signal. Don't care signals can be used both for inputs and outputs.

Using State Diagrams

State diagrams tend to produce very legible and easy to maintain code. The following steps are used in creating a design using state diagrams.

1. Declare the state bits.
2. Declare a name for the set of state bits.
3. Assign a value for each state.
4. Define the state machine clock signals with equations.
5. Define the state transitions using the name defined in step 2.

Declaring state bits, and defining how each state is represented by those state bits is done by using the sets and constant declarations presented earlier. After defining the state bits and states, an equations section is needed to specify clock signals and other control logic. Using the STATE_DIAGRAM keyword, the designer can now specify assignments and state transitions for each of those states. To specify an unconditional state transition, use the **GOTO** statement. For conditional transitions, use the **IF...THEN... ELSE** statement. Note that this is different from the **WHEN...THEN...ELSE** used in the equations section.

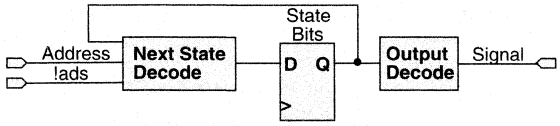
For example:

```
module traf

title 'Traffic Light Controler'

" A controller is needed to control the
" timing of a traffic light.
```

```
" The green light should be lit for thirty
" seconds.  Then a yellow light for two
" seconds, and a red light for
" thirty seconds We then repeat the entire
" cycle. This design must run on a clock
" which has a period of 1 sec.

"clock in with a period of 1 second.
clk             pin;

"reset to determine initial state.
reset           pin;

"Declare some nodes for a counter.
Count4..Count0  node istype 'reg';
Counter = [Count4..Count0];

"Step 1.  Declare the state bits.
"These bits are also our outputs
"in this example...
red             pin istype 'reg';
yellow          pin istype 'reg';
green           pin istype 'reg';

"Step 2.  Declare a name for the set of
"state bits.
"Step 3.  Assign a unique value for the
"for each state.
Light =   [green, yellow, red];
GO =      [ 1,      0,      0 ];
CAUTION = [ 0,      1,      0 ];
STOP =    [ 0,      0,      1 ];

Equations
"Step 4. Set up the clock and reset lines
"for the state machine.
green.ap     = reset;
red.ar       = reset;
yellow.ar    = reset;
Counter.ar   = reset;
Counter.clk = clk;
[green, yellow, red].clk = clk;

"Step 5.  Define the state transitions
"using the name defined in step 2.

State_Diagram Light

state GO:
IF (Counter < 30) then GO with
     Counter := Counter + 1;
ELSE goto CAUTION with
     Counter := Counter + 1;

state CAUTION:
IF (Counter != 0) then CAUTION with
     Counter := Counter + 1;
ELSE goto STOP with
     Counter := Counter + 1;

state STOP:
IF (Counter < 30) then STOP with
    Counter := Counter + 1;
ELSE goto GO with
    Counter := 1;

test_vectors
([clk,reset] -> [red, yellow, green])

    [0,1]    ->   [0,0,1];

  [.c., 0]   ->   [0,0,1];
@repeat 29
 {[.c., 0]   ->   [0,0,1];}

  [.c., 0]   ->   [0,1,0];
  [.c., 0]   ->   [0,1,0];

  [.c., 0]   ->   [1,0,0];
@repeat 29
 {[.c., 0]   ->   [1,0,0];}

  [.c., 0]   ->   [0,0,1];
@repeat 29
 {[.c., 0]   ->   [0,0,1];}

  [.c., 0]   ->   [0,1,0];
  [.c., 0]   ->   [0,1,0];

  [.c., 0]   ->   [1,0,0];
@repeat 29
 {[.c., 0]   ->   [1,0,0];}

end
```

Assignments can be made for state diagrams in two ways. If the assignment is made combinatorially in the state, then the outputs will be decoded from the state bits. This will improve the density of the final design, however it will add some delay. As shown in Figure 4, decoding from the present state adds a level of logic to achieve the desired output. For example:

```
zero_state:
output1 = 0;
GOTO one_state;

one_state:
output1 = 1;
GOTO zero_state;
```



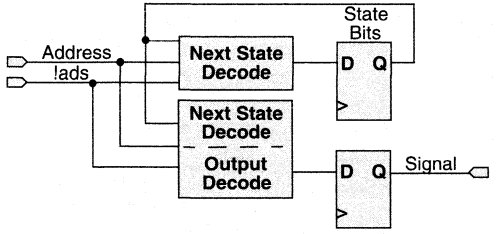
Output decoder may minimize product term requirement, but t_{CO} is slower.

Figure 4: Decoding Present State

If the signal is declared as a register type, then while the next state is determined, the output is decoded at the same time, and will transition at the next clock edge. This implementation is shown in Figure 5. To implement this in ABEL, we use the **with** statement in our state diagram.

For example:

```
zero_state :
GOTO one_state
with output1 := 1;

one_state :
GOTO zero_state
with output1 := 0;
```



Duplication of the next state decoder may increase product term count, but t_{CO} is faster.

Figure 5: Decoding Next State

Using Property Statements

Although ABEL was initially developed for PLDs, there are still device-specific features that ABEL does not support directly. Instead, ABEL provides a **property** statement allowing device specific commands to be passed to the fitter software. Property statements must be placed in the declarations section. These property statements allow the user to control the following:

- Slew rates
- Logic optimizations
- Logic placement
- Power settings
- Preload values

fast

The **fast** property controls the output slew rate, and there can only be one **fast** property used in each design. If there are only a few signals that require a fast slew rate, they can be listed individually after the property, and the remaining signals will be slew rate limited. Or, if there are only a few signals that need to be slew rate limited, then those signals can be listed.

```
xepld property 'fast on';
"all pins have fast slew rate

xepld property 'fast on x1 x2';
" only x1 and x2 are fast
" the remaining pins are slew limited

xepld property 'fast off x1 x2';
"only x1 and x2 are slew limited
"the remaining pins are fast
```

logic_opt

The **logic_opt** property allows the user to control the logic optimization done by the fitter. This should be used on selective nodes, where collapsing those nodes would cause the design to become very large.

```
xepld property 'logic_opt off';
"Preserves all combinatorial nodes

xepld property 'logic_opt off x1';
"preserve x1 and collapse other nodes to
fitter limits
```

minimize

The **minimize** property is used to prevent boolean minimization on equations, and is primarily used to prevent removal of redundant product terms in combinatorial logic.

```
xepld property 'minimize off x1 x2';
"keep redundant product terms for x1 & x2
```

partition

The **partition** property is used when specific placement of logic is desired.

```
xepld property 'partition fb1 x1 x2';
"place the functions of x1 and x2 in
"function block 1

xepld property 'partition fb1_2 x1';
"place the function x1 in
"function block 1, macrocell 2
```

pwr

The **pwr** property controls the power settings for individual macrocells.

```
xepld property 'pwr low';
"places all macrocells in low power mode

xepld property 'pwr low x1 x2'
"places x1 and x2 in low power mode
"the remaining in STD power mode

xepld property 'pwr std x1 x2'
"places x1 and x2 in STD power mode
"the remaining in low power mode
```

.prld

The **.prld** property controls the initial state of the registers at power up. Note that because this property is only passed on and used by the fitter, and is not used by ABEL, the preload value will not be reflected in test vectors. The default preload value is 0 for all XC9500 registers. Therefore, only registers that require a value of 1 need to be specified.

For example:

```
xepld property 'equation x1.prld = VCC';
"preload register x1 to a 1
```

DESIGN EXAMPLES

The following examples demonstrate some basic, specific design principles.

Bi-Directional pins

This example shows how to implement a bi-directional signal in ABEL. Bi-directional signals are commonly found whenever a bus is being used by several different devices. This usually involves some kind of control signal to allow only one device to drive the bus at a given time. In this example, the input pin **write** is used to control if data is being driven on to the data pins **D7..D0** from an outside source, such as a microprocessor, or if data is being driven from the CPLD to be read from the data pins by an external source

```
module bidi;

"This design will take a value from the
"pins D7..D0 and store it in a Register
"when the signal, write, is high. When
"write goes low, it will output the
"saved value at pins D7..D0

"inputs
write           pin;
myclock         pin;

"Bi-directional signal also has a register
"associated with it.
D7..D0          pin istype 'reg';

"Define my sets
Data = [D7..D0];

Equations;
Data.oe = !write;  "3-State the data lines
                   "when writing to register
Data.clk = myclock;

WHEN (write==1) THEN Data := Data.pin;
"When we are writing to the part, read the
"data pins and save in data register.
ELSE
Data := Data;
"Else, drive the data pins with the value
"saved in the register so we can read it
"back.
end;
```

Latches

Latches can be implemented in two ways. In the first example, latch_output utilizes the asynchronous set and reset of a flip-flop to implement a latch.

```
module ltest1

input,le        pin;
latch_output    pin istype 'reg';

equations

latch_output.ap = input & le;
latch_output.ar = !input & le;
latch_output.clk = 0; "Clock must be
                      "grounded
latch_output.d = 0; "D-input must be
                    "grounded

test_vectors    ([input, le] ->
[latch_output])
                [0,0] -> [0];
                [1,0] -> [0];
                [0,1] -> [0];
                [1,1] -> [1];
                [1,0] -> [1];
                [0,1] -> [0];

end;
```

Latches can also be implemented combinatorially by using a feedback path and providing a redundant product term to cover glitches. This will require the following code:

```
MODULE comlatch;

le          pin;
input       pin;
latch_out   pin  istype 'com,retain';
```

```
// The ABEL compiler will retain redundant
// logic for the latch_out output
// because they have the RETAIN attribute.
// However, the MINIMIZE OFF property
//statement is required to instruct the
// Xilinx fitter to also retain the
// redundant logic.

xepld property 'minimize off latch_out';
" The fitter will retain redundant logic
" for these nodes

EQUATIONS;

    latch_out = input & le
   " latch is transparent high
# latch_out & !le
" latch data on falling edge of le
# latch_out & input;
" Redundant product term

TEST_VECTORS
([ le, input] -> [latch_out]);
  [1 , 0  ]   -> [   0 ];  " transparent
  [1 , 1  ]   -> [   1 ];  " transparent
  [0 , 1  ]   -> [   1 ];  " latch a 1
  [0 , 0  ]   -> [   1 ];  " change input
  [1 , 0  ]   -> [   0 ];  " transparent
  [0 , 0  ]   -> [   0 ];  " latch a 0
  [0 , 1  ]   -> [   0 ];  " change data

END; " All modules must have an END state-
ment
```

Counters

Counters are useful in a variety of applications, such as memory interfaces, generating delay states, or simple state machines. This example shows how to build a loadable up/down counter with a count enable.

```
module counter

"32-bit Up/Down counter with parallel load
"and enable

"Outputs
Q31..Q0       pin istype 'reg';

"Inputs
D31..D0       pin;
Load          pin;              " Load Cmd
Count_Enable  pin;              " Count Cmd
UpDown        pin;             " Up/Down Cmd
myclk         pin;              " Clock

Counter = [Q31..Q0];
Input   = [D31..D0];

Equations
@carry 4;
Counter.clk = myclk;
WHEN (!Load & Count_Enable & UpDown)
     THEN Counter := Counter + 1
else
WHEN (!Load & Count_Enable & !UpDown)
     THEN Counter := Counter - 1
else
WHEN (Load)
     THEN Counter := Input;
else
Counter := Counter
end;
```

Multiplexers

Multiplexers can be used to control the data flow of a design. The following example demonstrates how to implement a 16 bit 4-1 registered multiplexer:

```
module mux2
A15..A0          pin; " Inputs
B15..B0          pin; " Inputs
C15..C0          pin; " Inputs
D15..D0          pin; " Inputs
Q15..Q0         pin istype 'reg'; " Output

Sel1..Sel0       pin;
clk              pin;

Output  = [Q20..Q0];
DataA = [A20..A0];
DataB = [B20..B0];
DataC = [C20..C0];
DataD = [D20..D0];
Select  = [Sel1..Sel0];

Equations

Output.clk = clk;

WHEN Select == 0 THEN Output := DataA
else WHEN Select == 1 THEN Output := DataB
else WHEN Select == 2 THEN Output := DataC
else Output := DataD;

end
```

Conclusion

ABEL allows complex behavioral designs to be easily implemented and simulated. In addition, the special features and capabilities of the device are easily accessed through ABEL property statements. ABEL is a simple yet powerful software tool that provides the designer with an efficient language for developing XC7300 and XC9500 designs.

Embedded Instrumentation Using XC9500 CPLDs

January, 1997 (Version 1.0) Application Note

Summary

This application note shows how to build embedded test instruments into XC9500 CPLDs.

Xilinx Family

XC9500

Introduction

Systems that use embedded remote diagnostic techniques can identify and isolate potential problems even before field personnel are assigned. This reduces equipment down time and gives field personnel the additional information that saves maintenance effort and money. Now, using the advanced features of the XC9500 CPLD family, designers can easily build advanced remote diagnostic capability into any system and achieve these remarkable benefits.

The XC9500 family has JTAG boundary scan capability built in, to quickly and easily perform a functional device or system test. However, in some cases it is beneficial to design-in other, more precise, diagnostic capabilities as well. This application note describes how to apply unused device resources, or even automatically and remotely reprogram XC9500 devices in the field, to create embedded signature analyzers, logic analyzers, and programmable test points to precisely pinpoint system failures.

Creating a Signature Analyzer

Various nodes (outputs or signal points) in a system will exhibit a repeatable, measurable pattern (signature) when stimulated appropriately. Therefore, failures can be diagnosed by comparing a measured signature with its expected value. Signature analysis identifies a set of critical "test points" in a system and then captures an encoded binary pattern using a polynomial encoding scheme similar to that of cyclic Linear Feedback Shift Registers (LFSRs). The circuitry consists of a shift register with XOR functions embedded at key positions which generate distinctive signature patterns. Figure 1 shows a typical schematic for a signature analyzer.

XC9500 CPLD Signature Analysis Support

Xilinx high performance CPLDs are frequently used in memory and system controllers. These controllers are centrally located within a system, and multiprocessing architectures may have several CPLDs on different cards, making them a natural place to embed diagnostic circuitry.

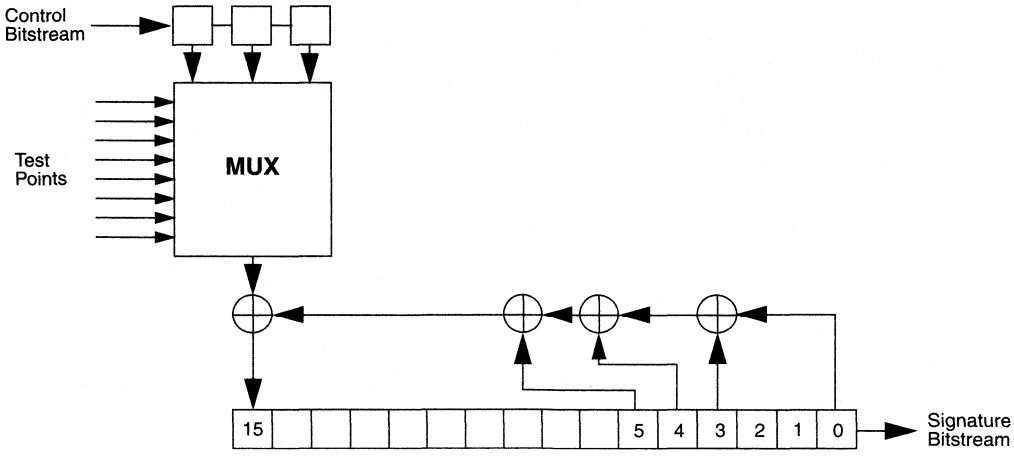


Figure 1: Signature Analysis Using a Cyclic Linear Feedback Shift Register

A typical embedded diagnostic strategy is shown below:

- Verify microprocessor operation at initialization by capturing signatures while transitioning out of reset.
- Verify more signatures as the processor steps through it's bootstrap ROM.
- Capture signatures as the processor tests the memory and subsequent I/O operations.

Figure 2 shows how such a configuration could be built.

Dedicated XC9500 CPLDs can easily implement any or all of the system shown in Figure 2. However, it is also possible to build the signature analyzer by reprogramming unused parts of the system.

For example, the DRAM controller CPLD may not be involved in the system diagnostics while the processor is booting from ROM. In this case, the DRAM controller logic can be erased from the CPLD and the signature analyzer can be constructed from it's logic resources. The DRAM controller is reprogrammed after the diagnostic routine is finished. Then, while the DRAM is being tested, unused data communication's or I/O circuitry can be reprogrammed to act as the signature analyzer. In this way, different system pieces can be reused or adapted to suit the diagnostic needs.

The ability of XC9500 devices to retain pinouts after making design changes, and their ability to perform a minimum of 10,000 program/erase cycles, makes embedded diagnostics and reprogrammability possible.

Signature Analysis Test Strategy Overview

The signature analyzer uses a multiplexed shift register which must be initialized with a pattern other than all zeros (which would make it fail). Therefore, the ability to initialize the register to a known pattern is necessary.

The test points shown in Figure 1 are selected with a three bit serial register and a multiplexer. The signatures must be unique, and each test point must have an identical starting point.

Before signature testing occurs, capture known good signatures and store them for future comparison using the procedure outlined below:

1. Aim the multiplexer at a chosen test point.
2. Initialize the Linear Feedback Shift Register (LFSR).
3. Enable the LFSR.
4. Enable the device under test (DUT).
5. Count operation cycles of DUT until complete.
6. Disable the LFSR and DUT.
7. Read back signature.
8. Compare this signature to the known good signature.

Managing the sequence of captured patterns is important. Once a board has passed all signature tests, the scope of the diagnostic can be increased. For example, analyzing data communication circuits with selected loopback tests. Additional tests may also be performed using the signature method as shown in Figure 2.

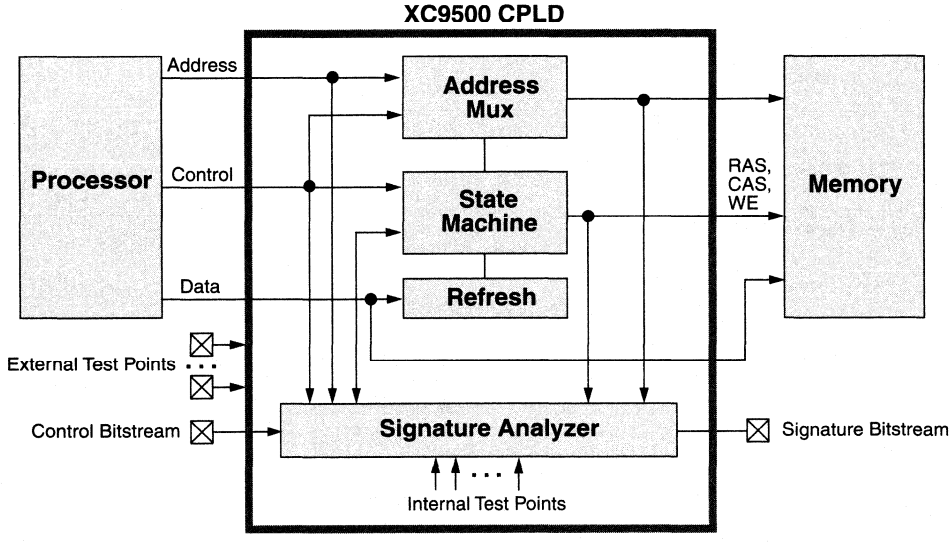


Figure 2: Embedded Signature Analyzer

Practical Considerations

Use the following guidelines when designing signature analysis experiments:

- Synchronous behavior is the most repeatable and the most desirable to use for signature analysis.
- Selection of test points should be done so their global influence is a factor for selection. Any processor status/ control lines which enable data transfers, generate interrupts or make bus requests are candidates.
- Most signatures are unique. However, there is a remote possibility of obtaining aliases. Check for aliases during the initial capture of base-line behavior.
- Table 1 supplies a set of feedback equations for building LFSRs which are the basic components for signature analyzers. Data entry can be exclusive ORed with the feedback input variable (n in this case).
- Avoid initializing the signature register with all 0s.
- Supplement signature analysis with other diagnostics such as memory tests, disk CRC diagnostics, and JTAG boundary scan.

Table 1: LSFR Feedback Equations

n	XOR Feedback Equation
2	X2 = X1 ⊕ X0
3	X3 = X1 ⊕ X0
4	X4 = X1 ⊕ X0
5	X5 = X2 ⊕ X0
6	X6 = X1 ⊕ X0
7	X7 = X3 ⊕ X0
8	X8 = X4 ⊕ X3 ⊕ X2 ⊕ X0
12	X12 = X6 ⊕ X4 ⊕ X1 ⊕ X0
16	X16 = X5 ⊕ X4 ⊕ X3 ⊕ X0
20	X20 = X3 ⊕ X0
24	X24 = X7 ⊕ X2 ⊕ X1 ⊕ X0
28	X28 = X3 ⊕ X0
32	X30 = X22 ⊕ X2 ⊕ X1 ⊕ X0

Note: n equals the number of shift register taps

Creating a Logic Analyzer

An embedded logic analyzer can capture and store a complete picture of system functionality at any number of simultaneous probe points. Figure 3 shows a typical implementation.

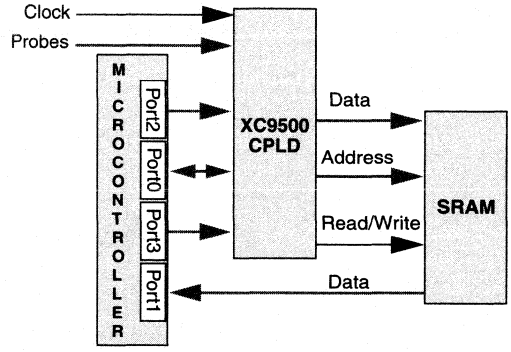


Figure 3: Three Chip Logic Analyzer

The microcontroller enables the logic analyzer and performs housekeeping activities as well as functional testing of the SRAM and CPLD connections. The CPLD drives the data to the SRAM, manages the SRAM addresses, generates memory strobes, and implements breakpoint logic to permit pre-triggering on patterns in the probe data. The SRAM stores the collected data in consecutive locations.

Figure 4 expands the detail for each block of the CPLD. 16-bit data is captured and passed through an output multiplexer to the SRAM data input port. A 16-bit mask register and a 16-Bit enable register combine with the 16-bit probe data to generate an equality compare for triggering the analyzer. The mask register provides the trigger value while the enable register allows for "don't care" bits.

The control unit automatically delivers memory read and write strobes when the analyzer is enabled by the microcontroller. The captured probe data is stored in the SRAM automatically. When the probe data pattern matches the mask register, a counter inside the control unit is enabled and decrements toward zero on each successive memory transfer. At zero, the control unit disables the storage process.

The counter shown in Figure 4, drives the SRAM address lines and increments on each transfer. The bus (shown in the lower half of Figure 4) permits the microcontroller to load the address counter and pass data through the data multiplexer. The data is read back through the microcontroller bus (not shown).

The microcontroller can enable, disable, and read back SRAM data which is forwarded to a personal computer or workstation for handling the housekeeping and display. The design can also be modified to perform as a simple digital oscilloscope by adding an analog to digital converter where the probes enter the CPLD.

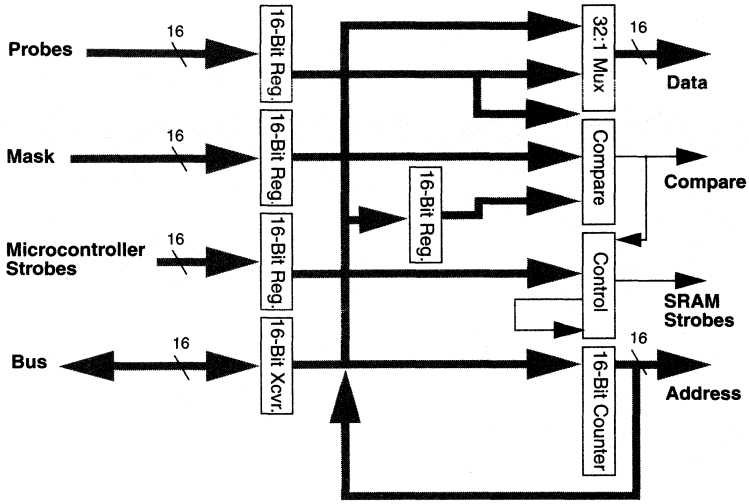


Figure 4: XC9500 Logic Analyzer Implementation

Creating Checkpoint Probes

Because of the small physical size of the pins used in high pin count packages, it is often difficult to attach oscilloscope or logic analyzer probes to a CPLD. In addition, its a good idea to not use physical probes due to the possibility of pin damage. By incrementally connecting a series of internal CPLD test points to dedicated, permanent probe points on the PC board, designers and technicians can access any number of internal nodes by using only one, or several, dedicated probe test points.

Figure 5 shows a CPLD with two dedicated "observation" output pins used for probing internal nodes. These pins can be connected to permanent test points that are soldered to the PC board allowing easy and reliable access for oscilloscopes or other test equipment.

XC9500 designs can be re-compiled to connect any internal node to the dedicated observation pins as needed.

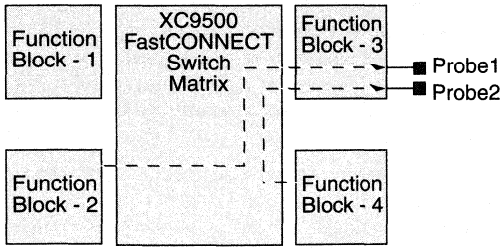


Figure 5: Creating Programmable Checkpoint Probes

Conclusion

XC9500 CPLDs provide the advanced technology for building effective yet inexpensive embedded test circuitry in a wide range of microprocessor systems. This capability is not limited to microprocessor diagnostics, and may be easily expanded to include a wider range of digital systems. With reliable pin-locking and a guaranteed minimum of 10,000 program/erase cycles, many new possibilities exist for creating systems with unsurpassed reliability and maintainability

References

1. Logic Design Principles (with emphasis on Testable Semicustom Circuits), E.J. McCluskey, Prentice-Hall, 1986
2. Design of Testable Logic Circuits, R.G. Bennets, Addison-Wesley, 1984
3. Digital Design (Principles and Practices), J. Wakerly, Prentice-Hall, 1994

Metastability Considerations

January, 1997 (Version 1.0) Application Note

Summary

Metastability is unavoidable in asynchronous systems. However, using the formulas and test measurements supplied here, designers can calculate the probability of failure. Design techniques for minimizing metastability are also provided.

Xilinx Families

XC7300, XC9500

Introduction

Metastability in digital systems can occur when the data input to a flip-flop is asynchronous to the clock, which can lead to setup or hold time violations. Metastability can appear as a flip-flop that switches late or doesn't switch at all. It can present a brief pulse at a flip-flop output (called a runt pulse) or cause flip-flop output oscillations. Any of these conditions can cause system failures.

The usual cause of metastability is a setup time violation, as demonstrated in Figure 1. If setup time violation is unavoidable, it is possible to calculate how frequently the flip-flop will fail. The industry standard formula for Mean Time Between Failures (MTBF) for a metastable flip-flop is given by:

$$MTBF = (e^{(-C2*tMET)}) / (C1*Fc*Fd)$$

where:

- e = 2.718281828...
- t_{MET} = time delay for the metastability to resolve itself
- F_C = the clocking frequency
- Fd = the data frequency
- C1 = a constant representing the metastability catching setup time window
- C2 = a constant describing the speed with which the metastable condition is resolved

This formula has been used over the last 25 years and is found to be accurate. The variables in the expression are functions of the flip-flop design, its process technology, the clocking rate, and the data switching speed, which are discussed in the following sections.

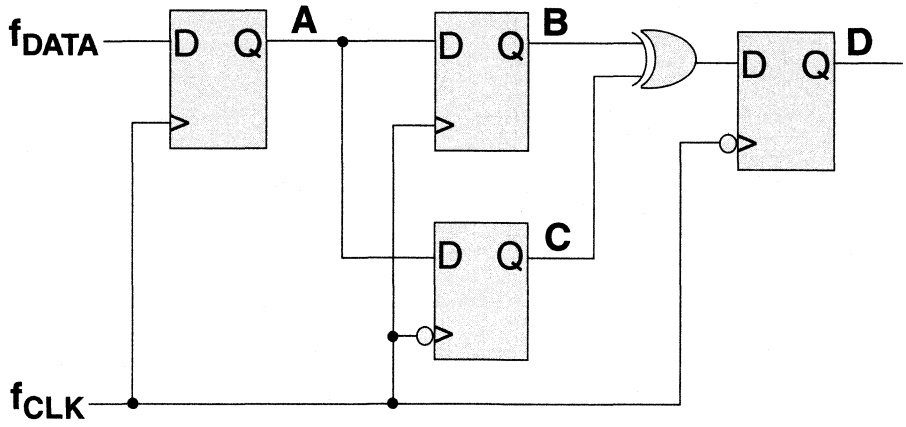


Figure 1: Metastability Measurement Circuit

Metastability Measurement

To test for metastability, a flip-flop is isolated within the CPLD and a clock is applied with asynchronous data input. The data is applied by an independent clocking source that is not related to the signal attached to the flip-flop clock input. The flip-flop eventually encounters a metastable state, which is observed by comparing the state of the flip-flop with it's state at a subsequent time, before the state should have changed again. If the state samples do not match, a metastable condition has occurred and a counter is incremented.

Two other questions must also be answered, and are given time parameters corresponding to their longevity:

- How often does metastability occur (related to C1)?
- How long does the metastable state persist when it does occur (related to C2)?

MTBF is inversely proportional to the clock rate (Fc) and the data rate (Fd). In designs having asynchronous data, most designers do not know their data rate, so it is difficult to estimate the MTBF accurately. Usually, a small time period is considered (10 seconds, for example) and the number of clocks and data transitions during the small time is used to define Fc and Fd. As the time delay is increased, the number of failures decreases dramatically.

By counting the number of failures over time, MTBF can be directly calculated. The values are derived by a formula which includes counts of the number of failures and the time delays for sampling.

Metastability Constants for Xilinx CPLDs

As shown in Figure 1, data is applied to flip-flop A asynchronously with respect to the clock input. The output of flip-flop A passes to two other flip-flops and a simple comparison of the two outputs is made. Note that flip-flop C and D are clocked by the inverted clock. If flip-flop B and C are not identical, a logical one will be captured by flip flop D, indicating a metastable event has occurred.

At 25 degrees C, with Vcc = 5.0 volts, several XC7300 and XC9500 devices were repeatedly measured. By knowing two MTBF values and two t_{MET} times, the constants C2 and C1 are obtained through the following expressions:

- $C1 = e^{(-C2 * tMET)} / (MTBF * Fc * Fd)$
- $C2 = Ln (\blacktriangle MTBF) / \blacktriangle t_{MET}$
- Fc = Frequency of the clock (10Mhz for these tests)
- Fd = Frequency of the data (1Mhz for these tests)
- $t_{MET} = (Ln(MTBF * Fc * Fd * C1)) / C2$

For the XC7300 family:

- $C2 = 3.49 * 10^{9}$
- $C1 = 1.0238 * 10^{-15}$

For the XC9500 family

- $C2 = 6.1172 * 10^{9}$
- $C1 = 9.554 * 10^{-18}$

As shown in Figure 2, the MTBF goes up dramatically as additional time delay for sampling the outputs increases. As a point of reference, 1 year is about 31.5 million seconds.

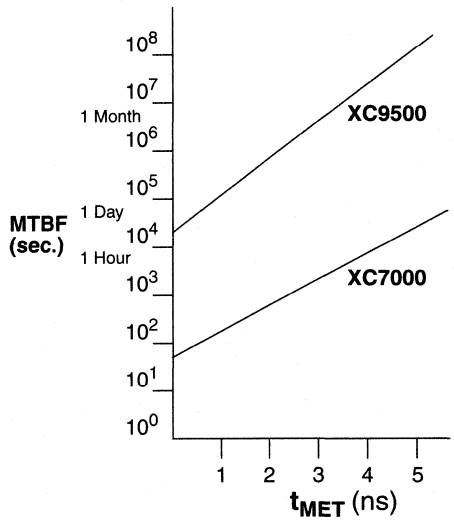


Figure 2: LOG MTBF versus t_{MET}

Design Considerations

To determine how to safely use a flip-flop, using the previous equation:

1. Determine a desired MTBF.
2. Insert the C1 and C2 values into the equation for the chosen flip-flop.
3. Determine whether data transitions are asynchronous or synchronous with respect to the clock. If they are asynchronous, use the average data switching rate calculated in step 4, as follows. If they are synchronous, use the quoted setup and hold times.
4. Calculate t_{MET} using the formula:

   $t_{MET} = (Ln(MTBF * Fc * Fd * C1)) / C2$

5. If the flip-flop passes through any output that causes it to have delays, add that delay to the t_{MET} expression.

Another way to decrease the effects of metastability is to cascade multiple flip-flops. Because metastability is a statistical effect, the possibility of metastability diminishes for cascaded flip-flops. Figure 3 shows a typical application.

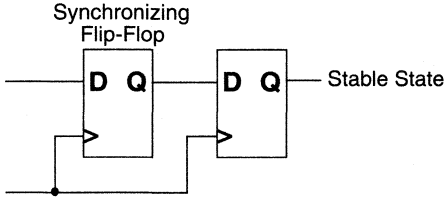


Figure 3: Synchronizing with a Cascaded Flip-Flop

Also, if setup and hold time violations are unavoidable, additional time delay may be added to provide more settling time.

Conclusion

Metastability is unavoidable in asynchronous systems but careful attention to design can usually prevent the problem of violating setup and hold times. Other design techniques exist for improving metastability performance and are described in the following references.

References

1. ANSI/IEEE Std. 1014-1987, IEEE Standard for a Versatile Backplane Bus: VMEbus, Appendix D (Metastability and Synchronization), pg. 281-295
2. Metastable behavior in digital systems, L. Kleeman and A. Cantoni, IEEE Design & Test of Computers, Dec. 1987
3. Measured Flip-Flop Responses to Marginal Triggering. IEEE Transaction on Computers, vol. C-32, no 12, Dec. 1983, pp 1207-1209.
4. High Speed Digital Design (A Handbook of Black Magic), H.W. Johnson and M. Graham, Prentice-Hall, 1993, pp 120 - 131

FastFLASH: A New Electrically Erasable CPLD Technology

January, 1997 (Version 1.0) Application Note

Summary

The Xilinx FastFLASH technology, used in the XC9500 family, provides key advantages in reliability, density, and performance. This overview describes the FastFLASH process technology and compares it with EEPROM technology.

Xilinx Family

XC9500

Introduction

The Xilinx FastFLASH technology is a double-polysilicon, 2-layer metal CMOS flash technology for CPLDs. This technology allows 5 V in-system programming. FastFLASH technology is capable of producing the high cell density needed for pin-locking, with a high endurance level of 10,000 program/erase cycles. Compared to typical CPLD EEPROM technology, FastFLASH technology provides advantages in reliability, density, and performance. The compatibility of the technology with industry-leading flash processes ensures the scalability of the basic process and the availability of foundry capacity.

Background

CPLDs are programmable by designers to implement desired logic functions. As non-volatile devices, CPLDs retain the programmed information even after removing power. The underlying non-volatile process technology supports this ability.

The process technology for CPLDs traditionally follows non-volatile memory (NVM) technologies. EPROM memory technology offers excellent memory cell density at a low process cost, although it is not electrically erasable. EEPROM technology offers electrical erasability at a reasonable process cost. However, it requires a relatively large memory cell size and offers limited endurance. Flash technology is an electrically erasable extension of EPROM technology. FastFLASH technology offers the best long term technology solution by providing the cell density of EPROM, the electrical erasability of EEPROM, excellent endurance characteristics, and long term process cost benefits.

Compared to EEPROM technology, the proprietary FastFLASH technology supports the needs of CPLDs by providing:

- high-performance logic capability
- high memory cell density
- electrical erasability
- 5 V program and erase
- high reliability and endurance
- process scalability
- fast device programming times

Figure 1 shows the relative cell sizes for FastFLASH and conventional EEPROM technologies.

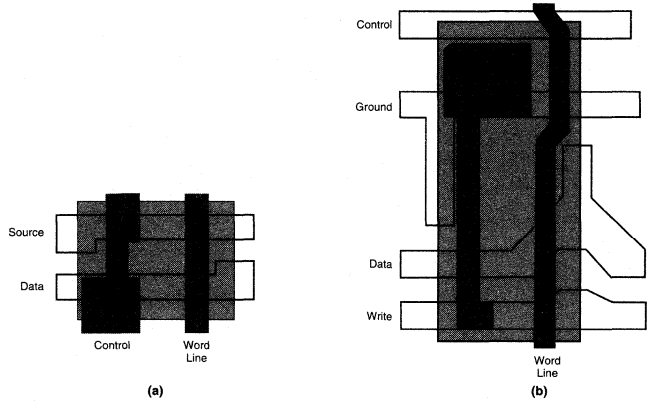


Figure 1: Layout Comparison of a FastFLASH Cell (a) and an EEPROM Cell (b)

EEPROM Technology

EEPROM technology was the first electrically erasable technology used for CPLDs. The programmable element is a special thin oxide capacitor that conducts a small current when a sufficient voltage is applied across the oxide. The tunnel oxide, approximately 80 Angstroms thick, is used to inject or extract charge from a floating gate via Fowler-Nordheim (FN) tunneling. The floating gate is connected to the gate of a sense transistor in order to sense the programming state. In addition to the tunnel oxide capacitor and sense transistor, two more transistors and an additional control capacitor are required to create a single EEPROM cell that can be programmed and erased in a CPLD application.

Figure 2 shows the schematic and cross-section of an EEPROM cell used in CPLDs. The tunnel oxide capacitor transports charge to and from the floating gate, which controls the sense transistor. Two additional transistors are used for the program and read operations. A control gate capacitor transfers voltage to the floating node for program and erase operations. Compared to standard CMOS logic processes, three additional device structures are created for the EEPROM cell: the tunnel oxide capacitor, the control gate capacitor and the high-voltage transistor. The resulting process complexity makes the scalability of the process and the EEPROM cell more difficult in future technology generations.

The EEPROM cell density is inferior because each cell is a circuit consisting of five separate device structures. The EEPROM cell area in a typical 0.6 micron technology is 75 to 100 square microns. The inadequate cell density directly affects the pin-locking ability of the architecture because more routing switches significantly increases die cost. The large cell size also introduces parasitic capacitances with the potential to limit the overall performance.

In general, EEPROM technology is developed and enhanced by individual companies without the benefit of an industry-wide development interest. This further increases the difficulty of long-term process migration.

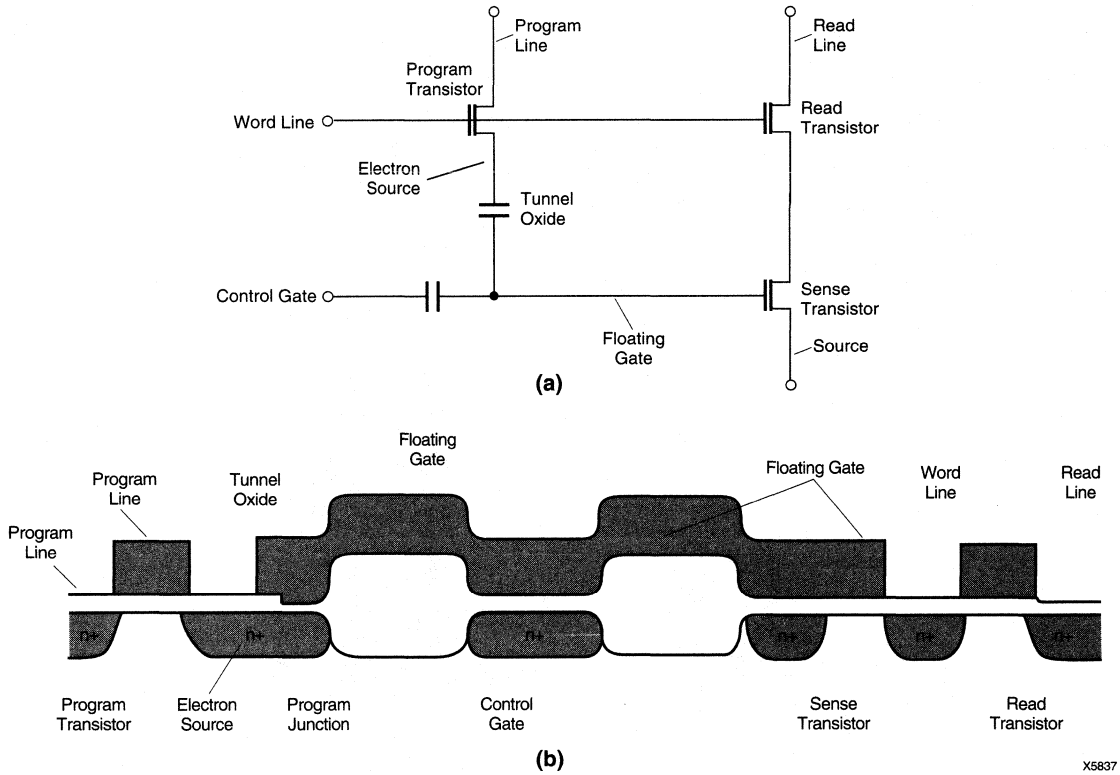


Figure 2: Schematic and Cross Section of an EEPROM Cell

FastFLASH Technology

FastFLASH technology is compatible with the industry-leading flash processes. The fundamental programmable element is the flash transistor. It incorporates the floating gate into the device structure for improved cell area as shown in Figure 3. By the addition of an NMOS transistor in series, as shown in Figure 4, the flash transistor is incorporated into the FastFLASH cell.

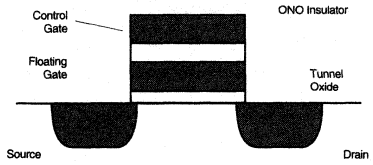


Figure 3: FastFLASH Transistor

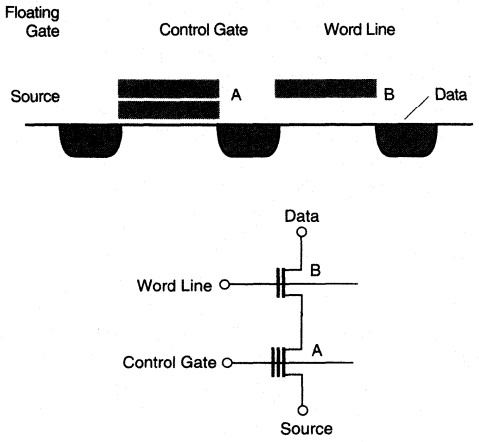


Figure 4: FastFLASH Cell with Series Transistor

The behavior of an individual flash transistor is changed with a program or an erase operation. When a flash transistor is in the erased state, the threshold voltage (V_{ta}) is approximately 1 V. During programming, the threshold voltage (V_{tb}) increases sufficiently above 5.5 V so the transistor does not turn on for a logic operation, as shown in Figure 5.

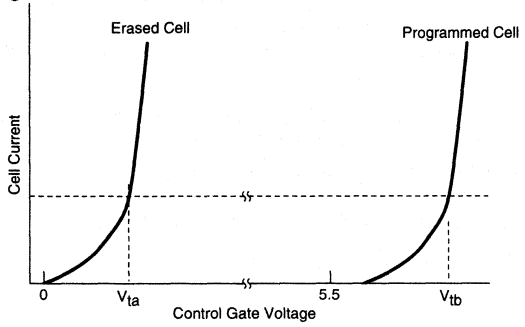


Figure 5: Cell Characteristics

The physical structure of the flash transistor includes a floating gate polysilicon layer that is isolated from the silicon substrate by a thin oxide approximately 100 Angstroms thick. Above the floating gate is the control gate polysilicon layer, with an insulating oxide-nitride-oxide layer between them. The control gate is driven by internal logic circuits, while the floating gate is unconnected. When the flash transistor is in the erased state, there is no net charge on the floating gate. By modifying the net electrical charge on the floating gate, the threshold voltage may be increased to 6 V or more.

The flash transistor is programmed by applying approximately 12 volts to the control gate, 5.5 volts to the drain, and 0 volts to the source as shown in Figure 6. The voltages are supplied by internal voltage pumps or externally by a device programmer. During the programming operation, channel hot electrons (CHE) are created near the pinch-off region. Some CHEs have sufficient thermal energy to pass through the thin oxide and remain on the floating gate. The collected electrons create a net negative voltage on the floating gate that opposes the electric field emanating from the control gate. The result is a net increase in the threshold voltage.

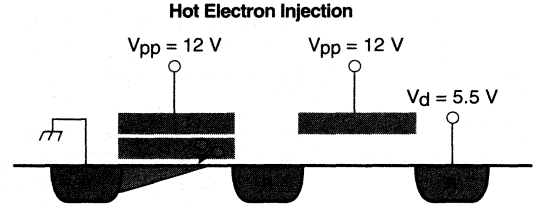


Figure 6: Programming a FastFLASH Cell

The flash transistor is erased by applying 0 volts to the control gate and approximately 10 volts to the source with the drain left floating. In Figure 7, the electric field between the floating gate and the source node is increased to the point where Fowler-Nordheim tunneling takes place. Excess electrons are transported from the floating gate to the source. The transistor is designed to make the erase process self-limiting. The electric field decreases as electrons are removed from the floating gate. FN tunneling effectively stops when the floating gate is electrically neutral.

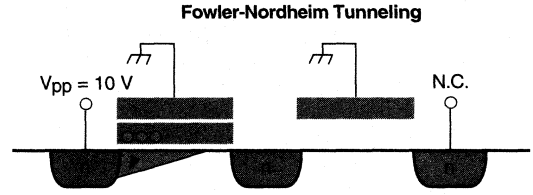


Figure 7: Erasing a FastFLASH Cell

Reliability and Endurance

The reliability of the programmable cell relates to data retention and endurance. Data retention is how long the cell can keep its programmed and erased states. Endurance is the number of times a cell can be programmed and erased without error.

When a cell is programmed and electrons are injected into the floating cell, the net charge is expected to remain indefinitely. In practice, charge leaks off -- typically tens of years under normal storage and operating conditions. The charge transport can occur due to direct tunneling and thermal leakage. Direct tunneling is exponentially dependent on electric field. Generally, the direct tunneling leakage current is very small and largely negligible. Thermal leakage is the dominant leakage component. The leakage characteristics are thoroughly characterized and modeled.

Endurance is determined by the magnitude of the applied electric field and the quality of the thin oxide used for the program and erase operations. When program and erase operations are performed in FastFLASH cells, electrons are transported across the oxide in the direction of the applied electric field.

Endurance failures occur in the alteration of the tunnel oxide characteristics where charge is repeatedly transported across the oxide. In FastFLASH technology, the maximum electric field is approximately 10 MV/cm. Consequently, the major cause of endurance failure is electron trapping in the tunnel oxide. After a large number of reprogramming cycles (typically over 100,000 and up to 1 million cycles), traps can be created within the oxide. When a sufficient number of electrons become trapped in the oxide, the localized electric field is distorted. The result is reduced program and erase efficiency, and reduced voltage margins between programmed and erased states. The FastFLASH technology endurance failure is a gradual degradation rather than an abrupt failure.

In EEPROM cells, the program and erase fields are 15 to 20 MV/cm. The substantially higher field strength makes the EEPROM cell susceptible to destructive tunnel oxide breakdown in addition to electron trapping in the oxide. Therefore, the oxide is physically ruptured and the cell is permanently damaged. The EEPROM endurance failure mechanism is a hard failure that is probabilistic in nature and difficult to screen.

In both flash and EEPROM technologies, oxide quality control becomes important. In typical CPLD EEPROM technologies, the tunnel oxide is approximately 80 Angstroms, among the thinnest tunneling oxides in use today. In contrast, FastFLASH technology uses industry-standard process steps to achieve a high-quality tunnel oxide of approximately 100 Angstroms.

There is a difference between the endurance of a single cell, and the overall endurance of the CPLD product using 100,000 cells or more. For the CPLD, failure statistics of a large number of cells suggest that the product endurance is typically 1.5 orders of magnitude lower than the endurance of an individual cell.

By using a significantly thicker tunnel oxide and lower electric fields than EEPROM technologies, the FastFLASH technology offers an endurance level of 10,000 cycles today, with plans for 1,000,000 cycles in the future. Comparatively, EEPROM CPLD technologies offer 10,000 cycles for the older, 0.8 micron, geometries and only 100 cycles for leading edge 0.6 micron (and smaller) technologies.

Cell Density

The FastFLASH cell, with its flash transistor and series read transistor, is considerably smaller than the comparable EEPROM cell with five device structures. Figure 1 shows the relative layouts of the FastFLASH cell and an EEPROM cell. The FastFLASH cell is approximately 25 square microns in 0.6 micron technology. In contrast, the EEPROM cell is approximately 75 to 100 square microns, or three to four times larger. Therefore, FastFLASH offers three or more times cell density than EEPROM technology.

Cell density is important in CPLD technology, because a programmable cell controls each routing switch. The routability and pin-locking capability are directly impacted by EEPROM-based CPLD architectures using fewer programmable cells for routing. These architectures typically devote 10 to 20% of the chip area to programmable cells. Therefore, increasing cell count by a factor of three would substantially increase the production costs in these devices.

Program and Erase Times

The CHE mechanism used in programming the FastFLASH cell is very fast, typically on the order of 20 microseconds for 8 bits. This is an important advantage for production device programming where devices can be programmed from the erased state in as few as two seconds using automatic test equipment. In contrast, FN tunneling used in EEPROM cells is several orders of magnitude slower, at approximately 40 milliseconds for several hundred cells. The net result is slower device programming.

For both technologies, FN tunneling is used in device erasure. In either case, the device is erased one sector at a time, with a typical erase time of several hundred milliseconds. In a production environment the devices are received in an erased state and therefore the erase time is not a cost factor in production programming.

Process Scalability

Process scalability is defined as the ease of shrinking the process and chips into future generations of technology with smaller chip sizes, lower chip costs, and faster chips. It is important to ensure that a migration path for cost and speed improvements, as well as continued availability of the current process to meet production requirements.

Factors that contribute to process scalability include:

- Number of different device structures
- Scalability of the programmable cell
- Compatibility with main-stream memory processes

The leading flash memory technologies are double-polysilicon, stacked gate technologies with CHE injection for programming and FN tunneling for erase. These technologies offer superior process scaling and continued development improvements in both cost and speed. Since the FastFLASH technology is compatible with these process technologies, it will continue to enjoy continued process migration into tighter geometries.

In contrast, the EEPROM technology used in CPLDs is not used in high-volume memory technologies. Combined with a complex cell structure, EEPROM technologies are substantially more difficult to migrate to tighter geometries.

Conclusion

FastFLASH technology offers significant advantages in reliability, density, and performance over comparable EEPROM technologies. Utilizing a flash transistor structure that is compatible with the leading non-volatile memory technology ensures process scalability and future foundry availability.

References

1. C.Hu, "Nonvolatile Semiconductor Memories: Technology, Design, and applications" IEEE press, New York, 1991.
2. C.Hu, "Thin oxide reliability", IEDM Tech. Digest pp 368-371, 1985.
3. S.K.Lai, V.K.Dham and D.Guterman, "Comparison and trends in today's dominant E^2 technologies", IEDM Tec. Dig., pp 580-583, 1986.
4. B.Dipert and L.Hebert, "Flash memory goes mainstream", IEEE spectrum, pp 48-52, October 1993.

XILINX®

XC9500 Data Sheets

1 ISP and JTAG Support

2 Application Notes

3 XC9500 Data Sheets

4 XC7300 Data Sheets

5 Device Packaging

6 Quality Assurance

7 Technical Support

8 Sales Offices, Representatives, Distributors

Visit us on the WEB at:

www.xilinx.com

for the latest data sheets.

XC9500 In-System Programmable CPLD Family

January, 1997 (Version 1.1)

Preliminary Product Information

Features

- High-performance
  - 5 ns pin-to-pin logic delays on all pins
  - f_{CNT} to 125 MHz
- Large density range
  - 36 to 576 macrocells with 800 to12,800 usable gates
- 5 V in-system programmable
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- PCI compliant (-5, -7, -10 speed grades)
- Advanced 0.6µm CMOS 5V FastFLASH technology
- Supports parallel programming of multiple XC9500 devices

Family Overview

The XC9500 CPLD family provides advanced in-system programming and test capabilities for high performance, general purpose logic integration. All devices are in-system programmable for a minimum of 10,000 program/erase cycles. Extensive IEEE 1149.1 (JTAG) boundary-scan support is also included on all family members.

As shown in Table 1, logic density of the XC9500 devices ranges from 800 to over 12,800 usable gates with 36 to 576 registers, respectively. Multiple package options and associated I/O capacity are shown in Table 2. The XC9500 family is fully pin-compatible allowing easy design migration across multiple density options in a given package footprint.

The XC9500 architectural features address the requirements of in-system programmability. Enhanced pin-locking capability avoids costly board rework. An expanded JTAG instruction set allows version control of programming patterns and in-system debugging. In-system programming throughout the full device operating range and a minimum of 10,000 program/erase cycles provide worry-free reconfigurations and system field upgrades.

Advanced system features include output slew rate control and user-programmable ground pins to help reduce system noise. I/Os may be configured for 3.3 V or 5 V operation. All outputs provide 24 mA drive.

Architecture Description

Each XC9500 device is a subsystem consisting of multiple Function Blocks (FBs) and I/O Blocks (IOBs) fully interconnected by the FastCONNECT switch matrix. The IOB provides buffering for device inputs and outputs. Each FB provides programmable logic capability with 36 inputs and 18 outputs. The FastCONNECT switch matrix connects all FB outputs and input signals to the FB inputs. For each FB, 12 to 18 outputs (depending on package pin-count) and associated output enable signals drive directly to the IOBs. See Figure 1.

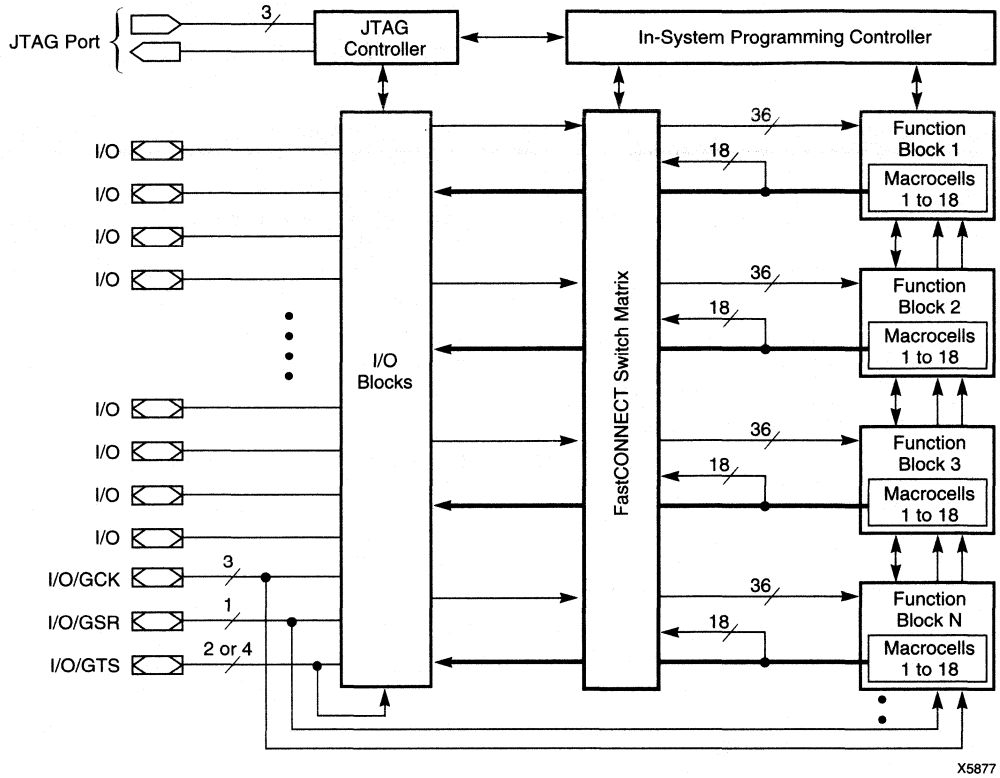


Figure 1: XC9500 Architecture

Note: Function Block outputs (indicated by the bold line) drive the I/O Blocks directly.

Table 1: XC9500 Device Family

	XC9536	XC9572	XC95108	XC95144	XC95180	XC95216	XC95288	XC95432	XC95576
Macrocells	36	72	108	144	180	216	288	432	576
Usable Gates	800	1,600	2,400	3,200	4,000	4,800	6,400	9600	12,800
Registers	36	72	108	144	180	216	288	432	576
t_{PD} (ns)	5	7.5	7.5	7.5	10	10	10	10	12
t_{SU} (ns)	4.5	5.5	5.5	5.5	6.5	6.5	6.5	6.5	9.5
t_{CO} (ns)	4.5	5.5	5.5	5.5	6.5	6.5	6.5	6.5	9.5
f_{CNT} (MHz)	100	125	125	125	111	111	111	111	100
f_{SYSTEM} (MHz)	100	83	83	83	67	67	67	67	67
	Preliminary								

Note: f_{CNT} = Operating frequency for 16-bit counters
f_{SYSTEM} = Internal operating frequency for general purpose system designs spanning multiple FBs.

Figure 2: Available Packages and Device I/O Pins (not including dedicated JTAG pins)

	XC9536	XC9572	XC95108	XC95144	XC95180	XC95216	XC95288	XC95432	XC95576
44-Pin VQFP	34								
44-Pin PLCC	34								
84-Pin PLCC		69	69						
100-Pin TQFP		72	81						
100-Pin PQFP		72	81	81					
160-Pin PQFP			108	133	133	133			
208-Pin HQFP					166	166	168		
352-Pin BGA						166	192		
432-Pin BGA								232[1]	232[1]

Note 1: Subject to change

Function Block

Each Function Block, as shown in Figure 3, is comprised of 18 independent macrocells, each capable of implementing a combinatorial or registered function. The FB also receives global clock, output enable, and set/reset signals. The FB generates 18 outputs that drive the FastCONNECT switch matrix. These 18 outputs and their corresponding output enable signals also drive the IOB.

Logic within the FB is implemented using a sum-of-products representation. Thirty-six inputs provide 72 true and complement signals into the programmable AND-array to form 90 product terms. Any number of these product terms, up to the 90 available, can be allocated to each macrocell by the product term allocator.

Each FB (except for the XC9536) supports local feedback paths that allow any number of FB outputs to drive into its own programmable AND-array without going outside the FB. These paths are used for creating very fast counters and state machines where all state registers are within the same FB.

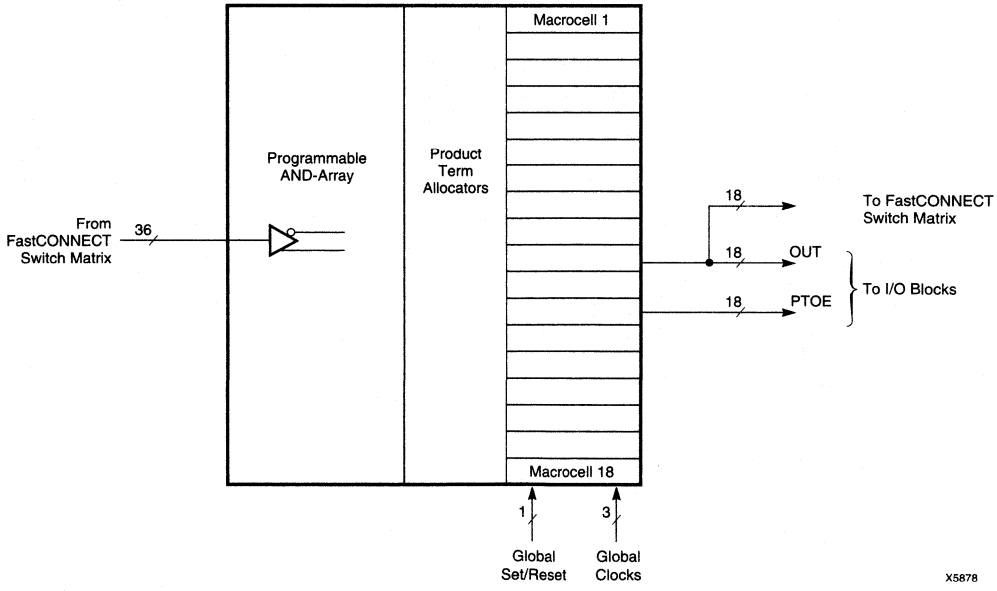


Figure 3: XC9500 Function Block

Macrocell

Each XC9500 macrocell may be individually configured for a combinatorial or registered function. The macrocell and associated FB logic is shown in Figure 4.

Five direct product terms from the AND-array are available for use as primary data inputs (to the OR and XOR gates) to implement combinatorial functions, or as control inputs including clock, set/reset, and output enable. The product term allocator associated with each macrocell selects how the five direct terms are used.

The macrocell register can be configured as a D-type or T-type flip-flop, or it may be bypassed for combinatorial operation. Each register supports both asynchronous set and reset operations. During power-up, all user registers are initialized to the user-defined preload state (default to 0 if unspecified).

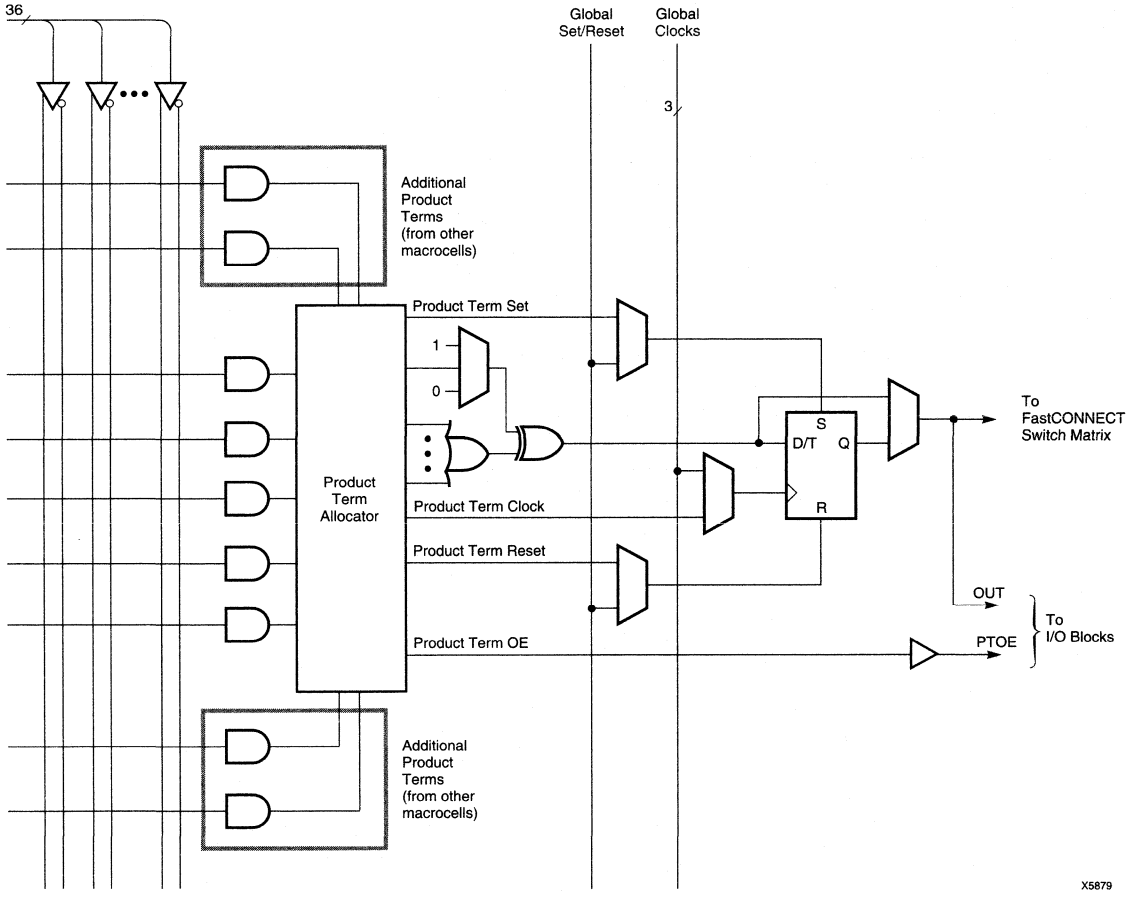


Figure 4: XC9500 Macrocell Within Function Block

All global control signals are available to each individual macrocell, including clock, set/reset, and output enable signals. As shown in Figure 5, the macrocell register clock originates from either of three global clocks or a product term clock. Both true and complement polarities of a GCK pin can be used within the device. A GSR input is also provided to allow user registers to be set to a user-defined state.

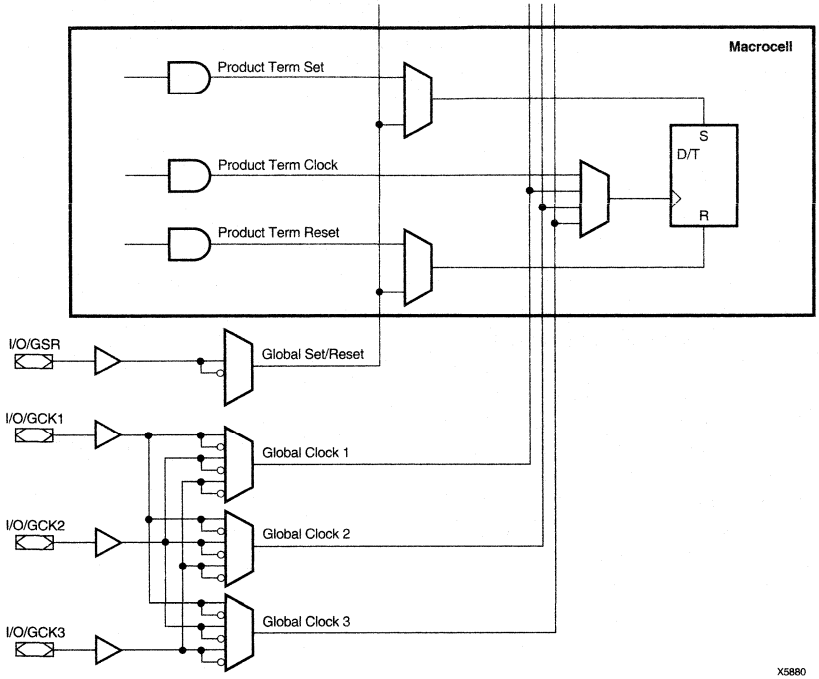


Figure 5: Macrocell Clock and Set/Reset Capability

Product Term Allocator

The product term allocator controls how the five direct product terms are assigned to each macrocell. For example, all five direct terms can drive the OR function as shown in Figure 6.

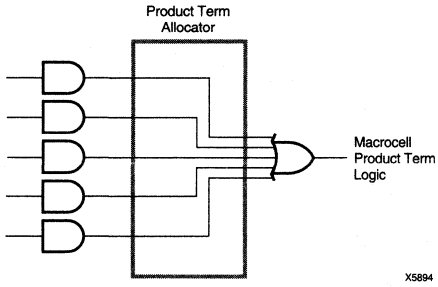


Figure 6: Macrocell Logic Using Direct Product Term

The product term allocator can re-assign other product terms within the FB to increase the logic capacity of a macrocell beyond five direct terms. Any macrocell requiring additional product terms can access uncommitted product terms in other macrocells within the FB. Up to 15 product terms can be available to a single macrocell with only a small incremental delay of t_{PTA}, as shown in Figure 7.

Note that the incremental delay affects only the product terms in other macrocells. The timing of the direct product terms is not changed.

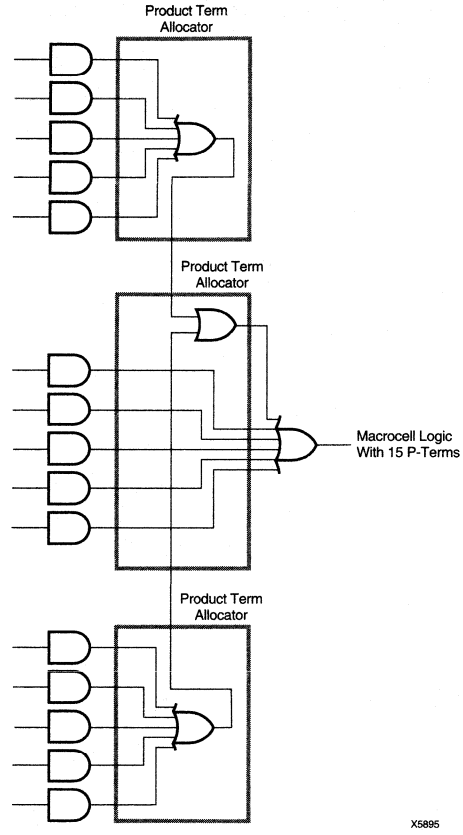


Figure 7: Product Term Allocation With 15 Product Terms

The product term allocator can re-assign product terms from any macrocell within the FB by combining partial sums of products over several macrocells, as shown in Figure 8.

In this example, the incremental delay is only $2*t_{PTA}$. All 90 product terms are available to any macrocell, with a maximum incremental delay of $8*t_{PTA}$.

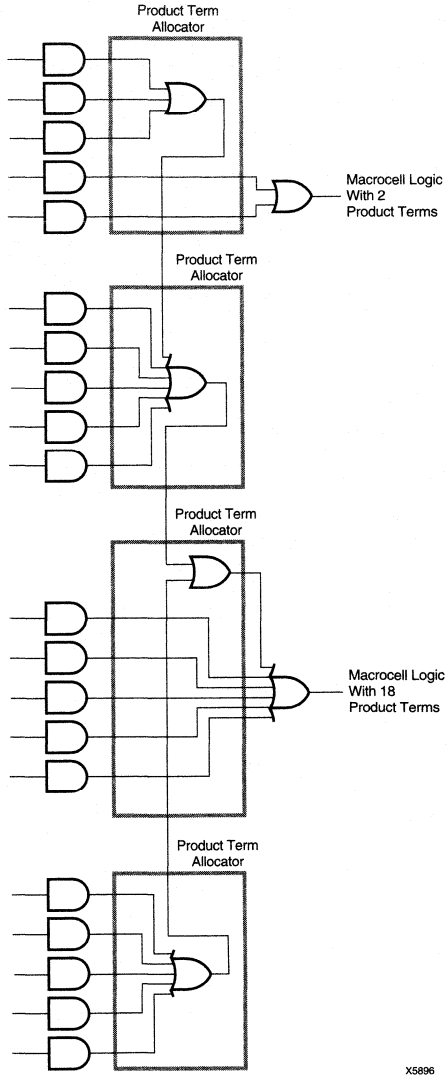


Figure 8: Product Term Allocation Over Several Macrocells

The internal logic of the product term allocator is shown in Figure 9.

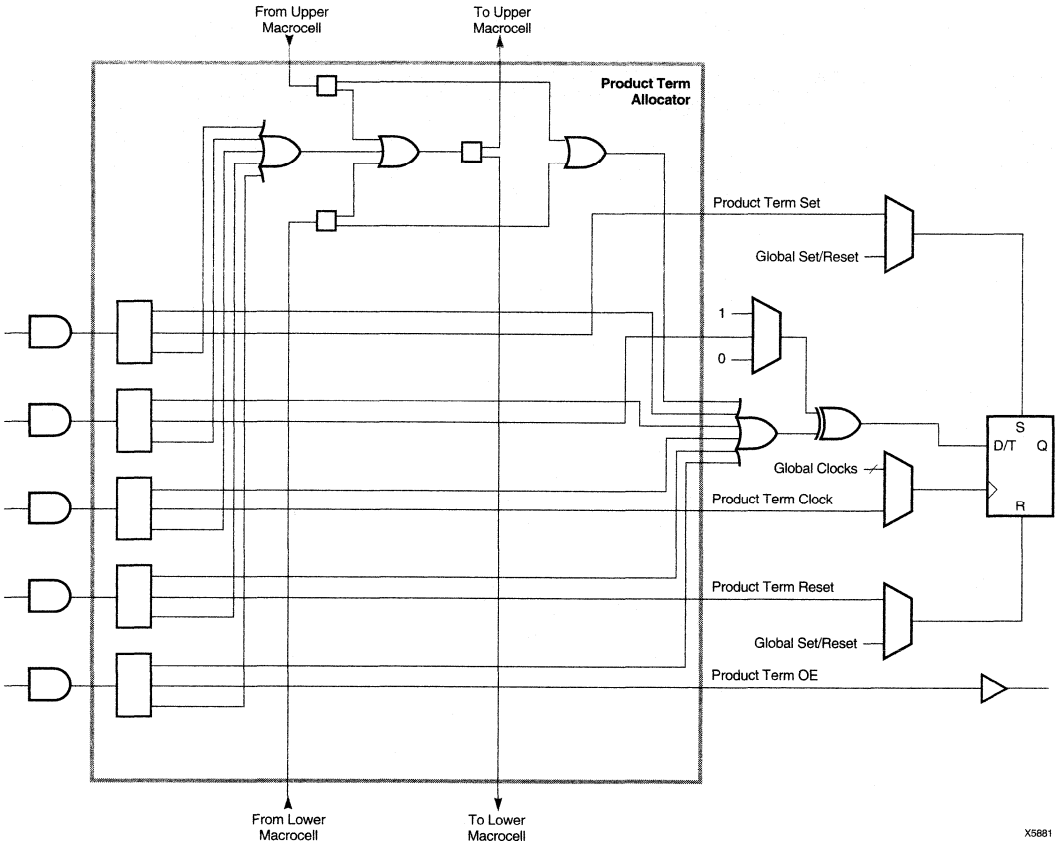


Figure 9: Product Term Allocator Logic

FastCONNECT Switch Matrix

The FastCONNECT switch matrix connects signals to the FB inputs, as shown in Figure 10. All IOB outputs (corresponding to user pin inputs) and all FB outputs drive the FastCONNECT matrix. Any of these (up to a FB fan-in limit of 36) may be selected, through user programming, to drive each FB with a uniform delay.

The FastCONNECT switch matrix is capable of combining multiple internal connections into a single wired-AND output before driving the destination FB. This provides additional logic capability and increases the effective logic fan-in of the destination FB without any additional timing delay. This capability is available for internal connections originating from FB outputs only. It is automatically invoked by the development software where applicable.

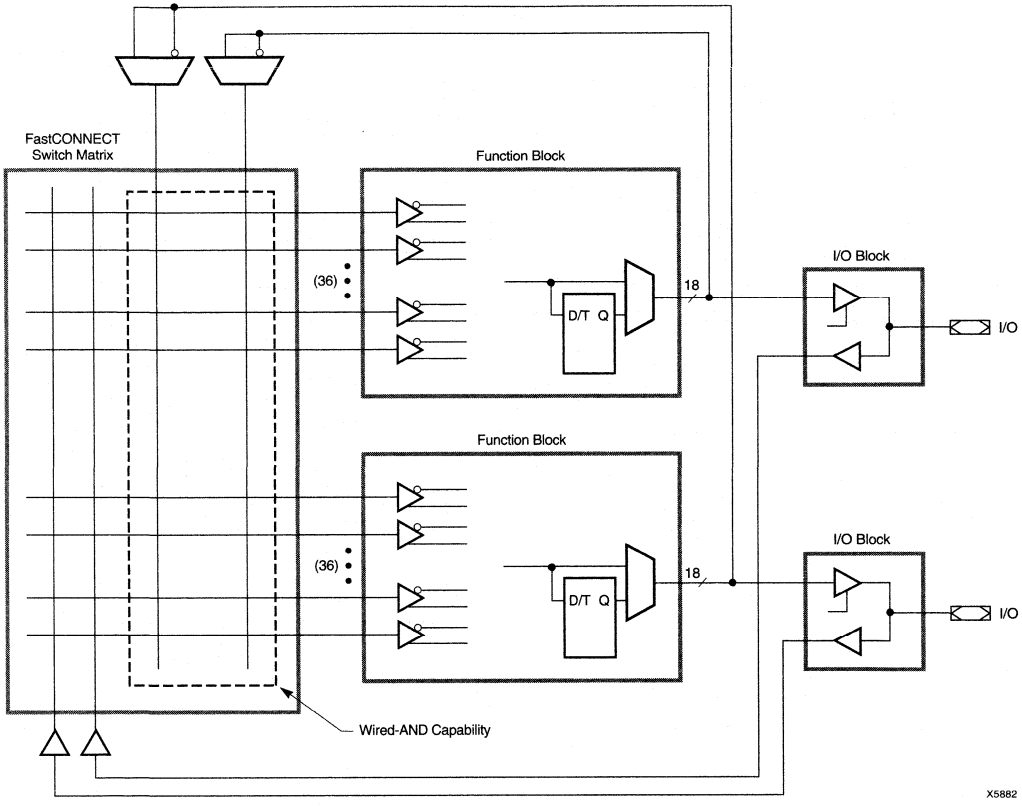


Figure 10: FastCONNECT Switch Matrix

I/O Block

The I/O Block (IOB) interfaces between the internal logic and the device user I/O pins. Each IOB includes an input buffer, output driver, output enable selection multiplexer, and user programmable ground control. See Figure 11 for details.

The input buffer is compatible with standard 5 V CMOS, 5 V TTL and 3.3 V signal levels. The input buffer uses the internal 5 V voltage supply (V_{CCINT}) to ensure that the input thresholds are constant and do not vary with the V_{CCIO} voltage.

The output enable may be generated from one of four options: a product term signal from the macrocell, any of the global OE signals, always "1", or always "0". There are two global output enables for devices with up to 144 macrocells, and four global output enables for devices with 180 or more macrocells. Both polarities of any of the global 3-state control (GTS) pins may be used within the device.

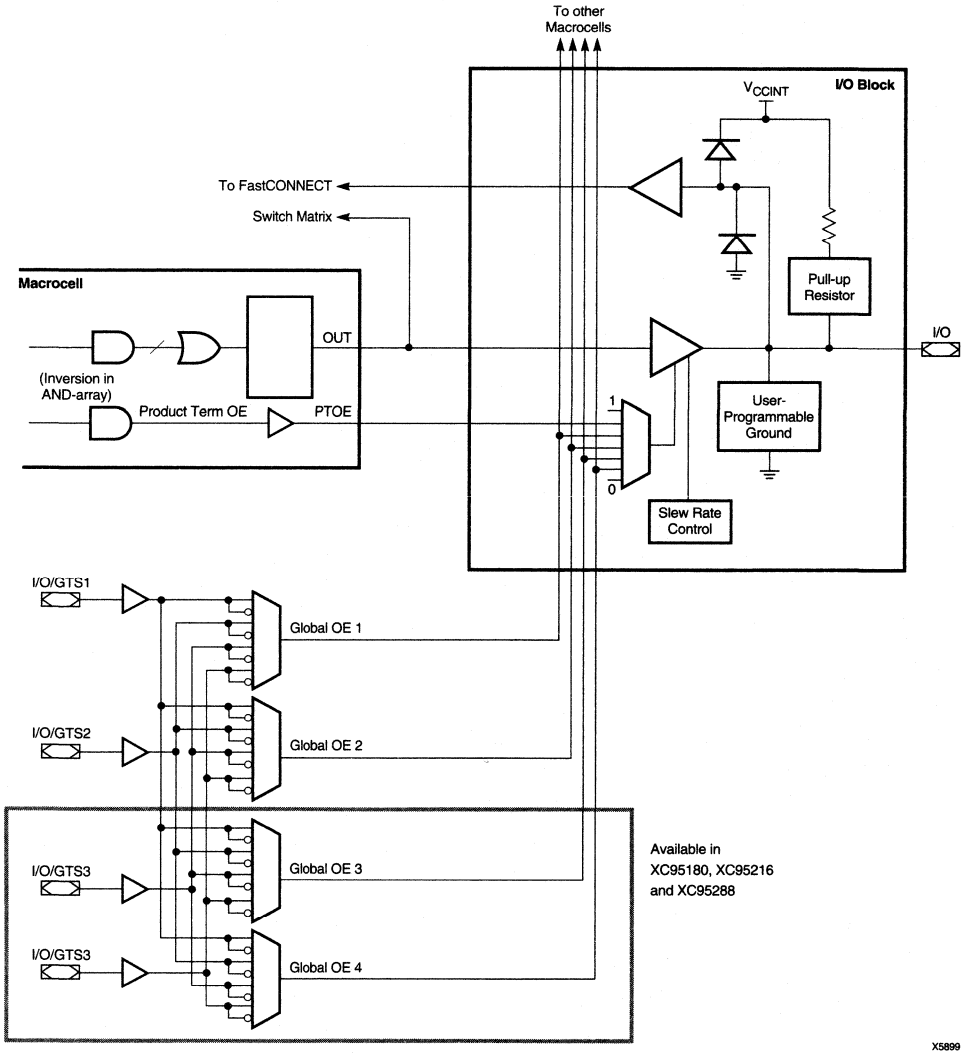


Figure 11: I/O Block and Output Enable Capability

Each output has independent slew rate control. Output edge rates may be slowed down to reduce system noise (with an additional time delay of t_{SLEW}) through programming. See Figure 12.

Each IOB provides user programmable ground pin capability. This allows device I/O pins to be configured as additional ground pins. By tying strategically located programmable ground pins to the external ground connection, system noise generated from large numbers of simultaneous switching outputs may be reduced.

A control pull-up resistor (typically 10K ohms) is attached to each device I/O pin to prevent them from floating when the device is not in normal user operation. This resistor is active during device programming mode and system power-up. It is also activated for an erased device. The resistor is deactivated during normal operation.

The output driver is capable of supplying 24 mA output drive. All output drivers in the device may be configured for either 5 V TTL levels or 3.3 V levels by connecting the device output voltage supply (V_{CCIO}) to a 5 V or 3.3 V voltage supply. Figure 13 shows how the XC9500 device can be used in 5 V only and mixed 3.3 V/5 V systems.

Pin-Locking Capability

The capability to lock the user defined pin assignments during design changes depends on the ability of the architecture to adapt to unexpected changes. The XC9500 devices have architectural features that enhance the ability to accept design changes while maintaining the same pinout.

The XC9500 architecture provides 100% routing within the FastCONNECT switch matrix, and incorporates a flexible Function Block that allows block-wide allocation of available product terms. This provides a high level of confidence of maintaining both input and output pin assignments for unexpected design changes.

For extensive design changes requiring higher logic capacity than is available in the initially chosen device, the new design may be able to fit into a larger pin-compatible device using the same pin assignments. The same board may be used with a higher density device without the expense of board rework.

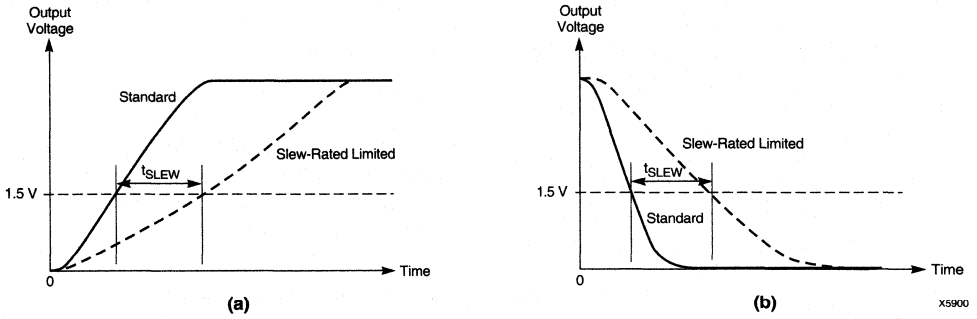


Figure 12: Output Slew-Rate Control For (a) Rising and (b) Falling Outputs

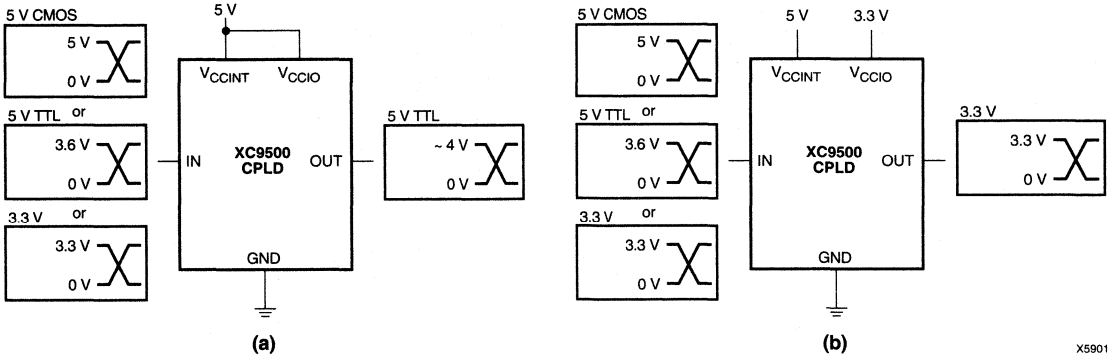


Figure 13: XC9500 Devices in (a) 5 V Systems and (b) Mixed 3.3 V/5 V Systems

In-System Programming

XC9500 devices are programmed in-system via a standard 4-pin JTAG protocol, as shown in Figure 14. In-system programming offers quick and efficient design iterations and eliminates package handling. The Xilinx development system provides the programming data sequence using a Xilinx download cable, a third-party JTAG development system, JTAG-compatible board tester, or a simple microprocessor interface that emulates the JTAG instruction sequence.

All I/Os are 3-stated and pulled high by the IOB resistors during in-system programming. If a particular signal must remain low during this time, then a pulldown resistor may be added to the pin.

External Programming

XC9500 devices can also be programmed by the Xilinx HW130 device programmer as well as third-party programmers. This provides the added flexibility of using pre-programmed devices during manufacturing, with an in-system programmable option for future enhancements.

In applications where in-system programmability is unnecessary or undesirable, the XC9500F family is available. The XC9536F, XC9572F, and XC95108F are 100% plug-in compatible with the standard XC9500 devices, supporting identical JEDEC bitmap, timing, and functionality. The XC9500F devices support full JTAG functionality as in the standard XC9500 family, with the exception of special in-system programming instructions. The XC9500F devices have the in-system programming circuit deactivated to ensure the programming information cannot be changed via the JTAG chain. All XC9500F devices must be programmed using the Xilinx HW130 or other third-party device programmers. The XC9500F series is supported for 44-pin VQFP, 84-pin PLCC, and 100-pin and 160-pin PQFP packages.

Endurance

All XC9500 CPLDs provide a minimum endurance level of 10,000 in-system program/erase cycles. Each device meets all functional, performance, and data retention specifications within this endurance limit.

IEEE 1149.1 Boundary-Scan (JTAG)

XC9500 devices fully support IEEE 1149.1 boundary-scan (JTAG). EXTEST, SAMPLE/PRELOAD, BYPASS, USERCODE, INTEST, IDCODE, and HIGHZ instructions are supported in each device. For ISP operations, five additional instructions are added; the ISPEN, FERASE, FPGM, FVFY, and ISPEX instructions are fully compliant extensions of the 1149.1 instruction set.

Design Security

XC9500 devices incorporate advanced data security features which fully protect the programming data against unauthorized reading or inadvertent device erasure/reprogramming. Table 2 shows the four different security settings available.

The read security bits can be set by the user to prevent the internal programming pattern from being read or copied. Erasing the entire device is the only way to reset the read security bit.

The write security bits provide added protection against accidental device erasure or reprogramming when the JTAG pins are subject to noise, such as during system power-up. Once set, the write-protection may be deactivated when the device needs to be reprogrammed with a valid pattern.

Table 2: Data Security Options

		Read Security: Default	Read Security: Set
Write Security	Default	Read Allowed Program/Erase Allowed	Read Inhibited Program/Erase Allowed
	Set	Read Allowed Program/Erase Inhibited	Read Inhibited Program/Erase Inhibited

X5905

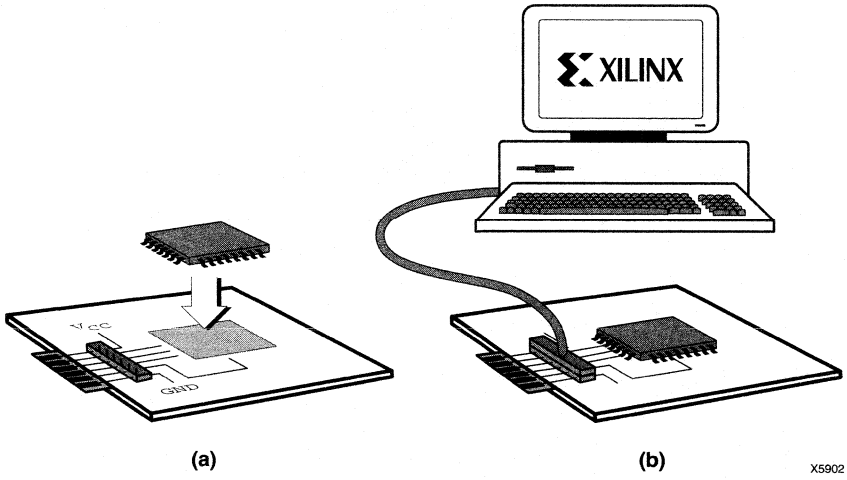


Figure 14: In-System Programming Operation (a) Solder Device to PCB and (b) Program Using Download Cable

Low Power Mode

All XC9500 devices offer a low-power mode for individual macrocells or across all macrocells. This feature allows the device power to be significantly reduced.

Each individual macrocell may be programmed in low-power mode by the user. Performance-critical parts of the application can remain in standard power mode, while other parts of the application may be programmed for low-power operation to reduce the overall power dissipation. Macrocells programmed for low-power mode incur additional delay (t_{LP}) in pin-to-pin combinatorial delay as well as register setup time. Product term clock to output and product term output enable delays are unaffected by the macrocell power-setting.

Timing Model

The uniformity of the XC9500 architecture allows a simplified timing model for the entire device. The basic timing model, shown in Figure 15, is valid for macrocell functions that use the direct product terms only, with standard power setting, and standard slew rate setting. Table 3 shows how each of the key timing parameters is affected by the product term allocator (if needed), low-power setting, and slew-limited setting.

The product term allocation time depends on the logic span of the macrocell function, which is defined as one less than the maximum number of allocators in the product term path. If only direct product terms are used, then the logic span is 0. The example in Figure 6 shows that up to 15 product terms are available with a span of 1. In the case of Figure 8, the 18 product term function has a span of 2.

Detailed timing information may be derived from the full timing model shown in Figure 16. The values and explanations for each parameter are given in the individual device data sheets. Refer to the application brief *Using the XC9500 Timing Model*, in this manual, for additional information.

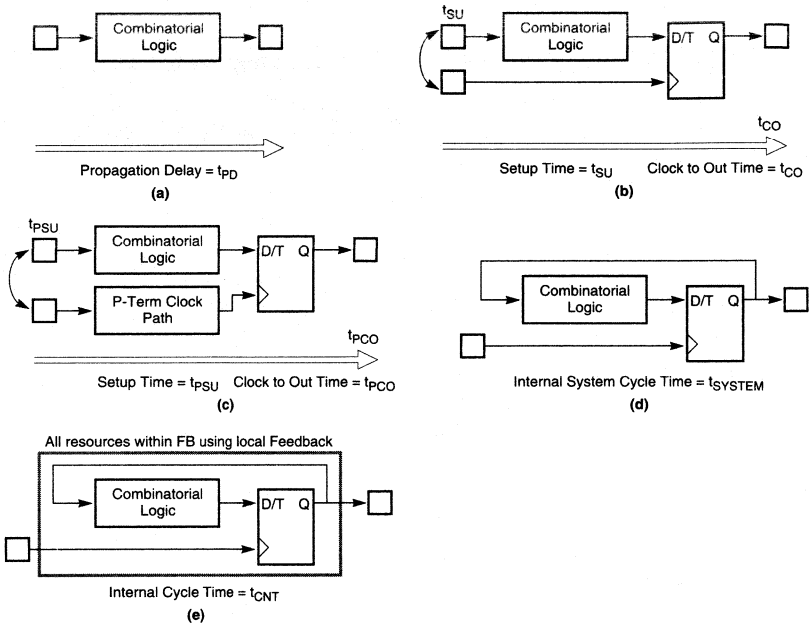


Figure 15: Basic Timing Model

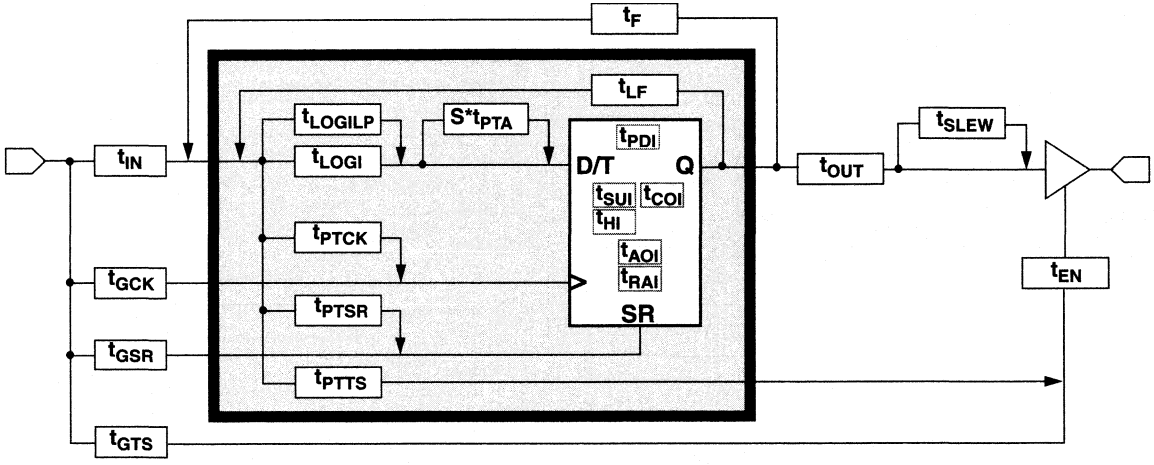


Figure 16: Detailed Timing Model

Power-Up Characteristics

The XC9500 devices are well behaved under all operating conditions. During power-up each XC9500 device employs internal circuitry which keeps the device in the quiescent state until the V_{CCINT} supply voltage is at a safe level (approximately 3.8 V). During this time, all device pins and JTAG pins are disabled and all device outputs are disabled with the IOB pull-up resistors (~ 10K ohms) enabled, as shown in Table 4. When the supply voltage reaches a safe level, all user registers become initialized (typically within 100 µs), and the device is immediately available for operation, as shown in Figure 17.

If the device is in the erased state (before any user pattern is programmed), the device outputs remain disabled with the IOB pull-up resistors enabled. The JTAG pins are enabled to allow the device to be programmed at any time.

If the device is programmed, the device inputs and outputs take on their configured states for normal operation. The JTAG pins are enabled to allow device erasure or boundary-scan tests at any time.

Development System Support

The XC9500 CPLD family is fully supported by the development systems available from Xilinx and the Xilinx Alliance Program vendors.

The designer can create the design using ABEL, schematics, equations, VHDL, or other HDL languages in a variety of software front-end tools. The development system can be used to implement the design and generate a JEDEC bitmap which can be used to program the XC9500 device. Each development system includes JTAG download software that can be used to program the devices via the standard JTAG interface and a download cable.

FastFLASH Technology

An advanced 0.6 µm CMOS Flash process is used to fabricate all XC9500 devices. Specifically developed for Xilinx in-system programmable CPLDs, the FastFLASH process provides high performance logic capability, fast programming times, and endurance of 10,000 program/erase cycles.

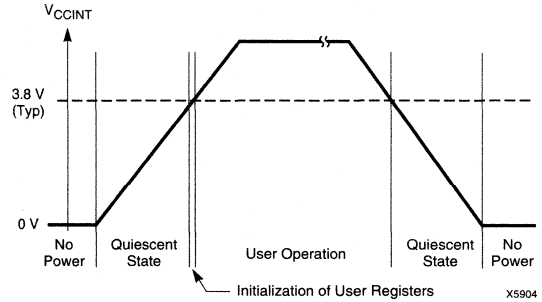


Figure 17: Device Behavior During Power-up

Table 3: Timing Model Parameters

Description	Parameter	Product Term Allocator[1]	Macrocell Low-Power Setting	Output Slew-Limited Setting
Propagation Delay	t_{PD}	+ t_{PTA} * S	+ t_{LP}	+ t_{SLEW}
Global Clock Setup Time	t_{SU}	+ t_{PTA} * S	+ t_{LP}	–
Global Clock-to-output	t_{CO}	–	–	+ t_{SLEW}
Product Term Clock Setup Time	t_{PSU}	+ t_{PTA} * S	+ t_{LP}	–
Product Term Clock-to-output	t_{PCO}	–	–	+ t_{SLEW}
Internal System Cycle Period	t_{SYSTEM}	+ t_{PTA} * S	+ t_{LP}	–

Note: 1. S = the logic span of the function, as defined in the text.

Table 4: XC9500 Device Characteristics

Device Circuitry	Quiescent State	Erased Device Operation	Valid User Operation
IOB Pull-up Resistors	Enabled	Enabled	Disabled
Device Outputs	Disabled	Disabled	As Configured
Device Inputs and Clocks	Disabled	Disabled	As Configured
Function Block	Disabled	Disabled	As Configured
JTAG Controller	Disabled	Enabled	Enabled

XC9536 In-System Programmable CPLD

January, 1997 (Version 1.0)

Preliminary Product Specification

Features

- 5 ns pin-to-pin logic delays on all pins
- f_{CNT} to 100 MHz
- 36 macrocells with 800 usable gates
- Up to 34 user I/O pins
- 5 V in-system programmable (ISP)
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- PCI compliant (-5, -7, -10 speed grades)
- Advanced 0.6 µm CMOS 5V FastFLASH technology
- Supports parallel programming of more than one XC9500 concurrently
- Available in 44-pin PLCC and 44-pin VQFP packages
- Plug-in compatible non-ISP XC9536F available in 44-pin PLCC and 44-pin VQFP packages

Description

The XC9536 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration. It is comprised of two 36V18 Function Blocks, providing 800 usable gates with propagation delays of 5 ns. See Figure 2 for the architecture overview.

Power Management

Power dissipation can be reduced in the XC9536 by configuring macrocells to standard or low-power modes of operation. Unused macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

I_{CC} (mA) =

MC_{HP} (1.7) + MC_{LP} (0.9) + MC (0.006 mA/MHz) f

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of macrocells used

f = Clock frequency (MHz)

Figure 1 shows a typical calculation for the XC9536 device.

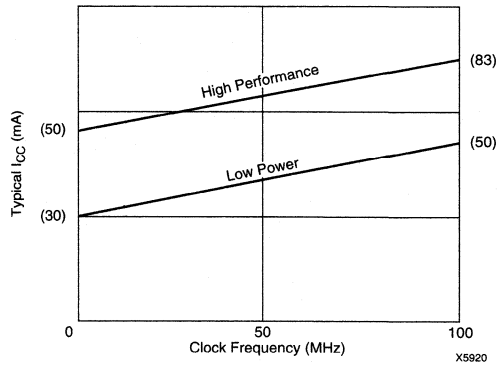


Figure 1: Typical I_{CC} vs. Frequency For XC9536

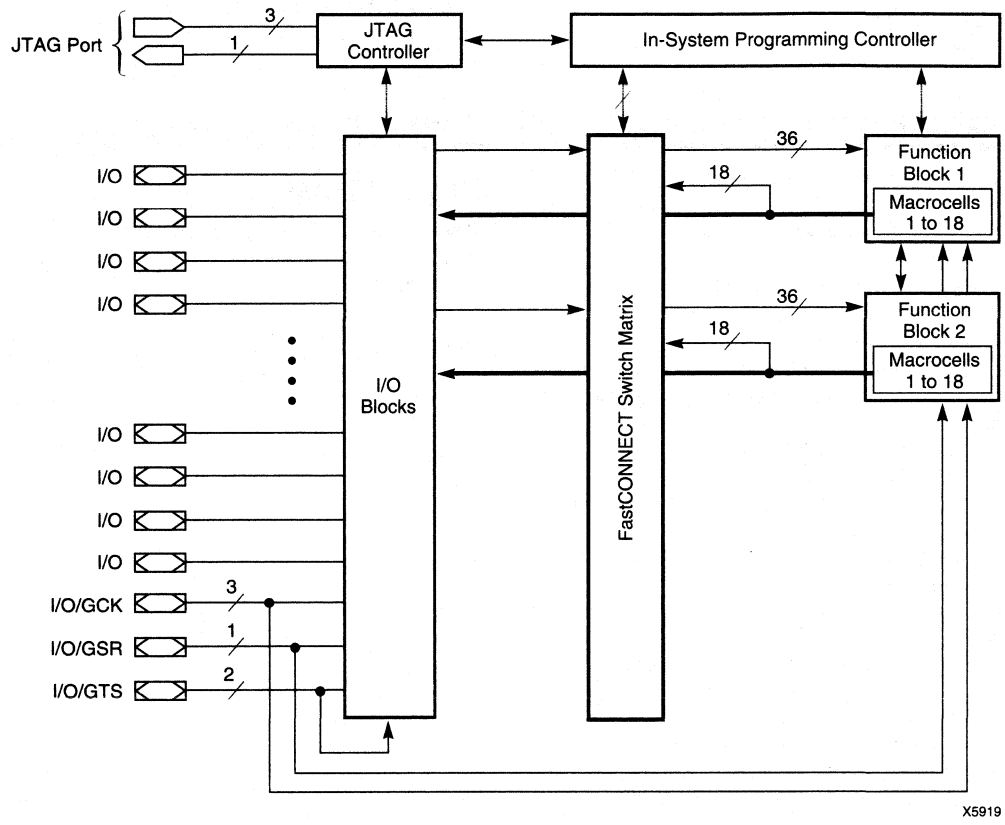


Figure 2: XC9536 Architecture

Note: Function Block outputs (indicated by the bold line) drive the I/O Blocks directly

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage relative to GND	-0.5 to 7.0	V
V_{IN}	DC input voltage relative to GND	-0.5 to V_{CC} + 0.5	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to V_{CC} + 0.5	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Max soldering temperature (10 ns @ 1/16 in = 1.5 mm)	+260	°C

Warning: 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 Rating conditions for extended periods of time may affect device reliability.

Recommended Operating Conditions[1]

Symbol	Parameter	Min	Max	Units
V_{CCINT}	Supply voltage for internal logic and input buffer	4.75 (4.5)	5.25 (5.5)	V
V_{CCIO}	Supply voltage for output drivers for 5 V operation	4.75 (4.5)	5.25 (5.5)	V
	Supply voltage for output drivers for 3.3 V operation	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.80	V
V_{IH}	High-level input voltage	2.0	V_{CCINT} +0.5	V
V_O	Output voltage	0	V_{CCINT} + 0.5	V

Note 1. Numbers in parenthesis are for industrial-temperature range versions.

Endurance Characteristics

Symbol	Parameter	Min	Max	Units
t_{DR}	Data Retention	20	-	Years
N_{PE}	Program/Erase Cycles	10,000	-	Cycles

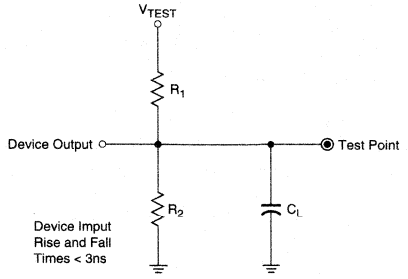
DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	Output high voltage for 5 V operation	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	Output high voltage for 3.3 V operation	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	Output low voltage for 5 V operation	I_{OL} = 24 mA V_{CC} = Min		0.5	V
	Output low voltage for 3.3 V operation	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
I_{IH}	I/O high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
C_{IN}	I/O capacitance	V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}	Operating Supply Current (low power mode, active)	V_I = GND, No load f = 1.0 MHz	30 (Typ)		mA

AC Characteristics

Symbol	Parameter	XC9536-5		XC9536-7		XC9536-10		XC9536-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{PD}	I/O to output valid		5.0		7.5		10.0		15.0	ns
t_{SU}	I/O setup time before GCK	4.5		5.5		6.5		8.0		ns
t_{H}	I/O hold time after GCK	0.0		0.0		0.0		0.0		ns
t_{CO}	GCK to output valid		4.5		5.5		6.5		8.0	ns
f_{CNT}[1]	16-bit counter frequency	100		83		67		56		MHz
f_{SYSTEM}[2]	Multiple FB internal operating frequency	100		83		67		56		MHz
t_{PSU}	I/O setup time before p-term clock input	0.5		1.5		2.5		4.0		ns
t_{PH}	I/O hold time after p-term clock input	4.0		4.0		4.0		4.0		ns
t_{PCO}	P-term clock to output valid		8.5		9.5		10.5		12.0	ns
t_{OE}	GTS to output valid		6.0		7.0		10.0		15.0	ns
t_{OD}	GTS to output disable		6.0		7.0		10.0		15.0	ns
t_{POE}	Product term OE to output enabled		10.5		13.0		15.5		18.0	ns
t_{POD}	Product term OE to output disabled		10.5		13.0		15.5		18.0	ns
t_{WLH}	GCK pulse width (High or Low)		4.0		4.0		4.5		5.5	ns
		Preliminary								

Note: 1. f_{CNT} is the fastest 16-bit counter frequency available.
f_{CNT} is also the Export Control Maximum flip-flop toggle rate, f_{TOG}.
2. f_{SYSTEM} is the internal operating frequency for general purpose system designs spanning multiple FBs.



V_{CCIO} Level	V_{TEST}	R_1	R_1	C_L
5 V	5.0 V	160 Ω	120 Ω	35 pF
3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X5222

Figure 3: AC Load Circuit

Internal Timing Parameters

Symbol	Parameter	XC9536-5		XC9536-7		XC9536-10		XC9536-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
Buffer Delays										
t_{IN}	Input buffer delay		1.5		2.5		3.5		4.5	ns
t_{GCK}	GCK buffer delay		2.0		2.5		3.0		3.0	ns
t_{GSR}	GSR buffer delay		4.0		4.5		6.0		7.5	ns
t_{GTS}	GTS buffer delay		6.0		7.0		10.0		15.0	ns
t_{OUT}	Output buffer delay		2.0		2.5		3.0		4.5	ns
t_{EN}	Output buffer enable/disable delay		0.0		0.0		0.0		0.0	ns
Product Term Control Delays										
t_{PTCK}	Product term clock delay		4.5		4.0		3.5		2.5	ns
t_{PTSR}	Product term set/reset delay		1.0		2.0		2.5		3.0	ns
t_{PTTS}	Product term 3-state delay		9.0		10.5		12.0		13.5	ns
Internal Register and Combinatorial delays										
t_{PDI}	Combinatorial logic propagation delay		0.5		0.5		1.0		3.0	ns
t_{SUI}	Register setup time	4.0		3.5		3.5		3.5		ns
t_{HI}	Register hold time	0.5		2.0		3.0		4.5		ns
t_{COI}	Register clock to output valid time		0.5		0.5		0.5		0.5	ns
t_{AOI}	Register async. S/R to output delay		6.0		6.5		7.0		8.0	ns
t_{RAI}	Register async. S/R recovery before clock	5.0		7.5		10.0		15.0		ns
t_{LOGI}	Internal logic delay		1.0		2.0		2.5		3.0	ns
t_{LOGILP}	Internal low power logic delay		9.0		10.0		11.0		11.5	ns
Feedback Delays										
t_F	FastCONNECT matrix feedback delay		4.5		6.0		8.5		11.0	ns
t_{LF}	Function Block local feeback delay		4.5		6.0		8.5		11.0	ns
Time Adders										
t_{PTA}[3]	Incremental Product Term Allocator delay		1.0		1.0		1.0		1.5	ns
t_{SLEW}	Slew-rate limited delay		3.5		4.0		4.5		5.0	ns
		Preliminary								

Note: 3. t_{PTA} is multiplied by the span of the function as defined in the family data sheet.

XC9536 I/O Pins

Function Block	Macrocell	PC44	VQ44	BScan Order	Notes
1	1	2	40	105	
1	2	3	41	102	
1	3	5	43	99	[1]
1	4	4	42	96	
1	5	6	44	93	[1]
1	6	8	2	90	
1	7	7	1	87	[1]
1	8	9	3	84	
1	9	11	5	81	
1	10	12	6	78	
1	11	13	7	75	
1	12	14	8	72	
1	13	18	12	69	
1	14	19	13	66	
1	15	20	14	63	
1	16	22	16	60	
1	17	24	18	57	
1	18	–	–	54	

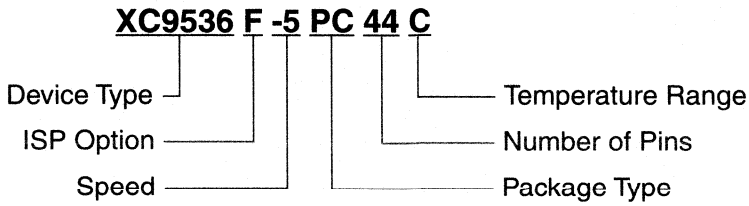
Function Block	Macrocell	PC44	VQ44	BScan Order	Notes
2	1	1	39	51	
2	2	44	38	48	
2	3	42	36	45	[1]
2	4	43	37	42	
2	5	40	34	39	[1]
2	6	39	33	36	[1]
2	7	38	32	33	
2	8	37	31	30	
2	9	36	30	27	
2	10	35	29	24	
2	11	34	28	21	
2	12	33	27	18	
2	13	29	23	15	
2	14	28	22	12	
2	15	27	21	9	
2	16	26	20	6	
2	17	25	19	3	
2	18	–	–	0	

Note: [1] Global control pin

XC9536 Global, JTAG and Power Pins

Pin Type	PC44	VQ44
I/O/GCK1	5	43
I/O/GCK2	6	44
I/O/GCK3	7	1
I/O/GTS1	42	36
I/O/GTS2	40	34
I/O/GSR	39	33
TCK	17	11
TDI	15	9
TDO	30	24
TMS	16	10
V_{CCINT} 5 V	21,41	15,35
V_{CCIO} 3.3 V/5 V	32	26
GND	23,10,31	17,4,25
No Connects	—	—

Ordering Information



ISP Options

__ With ISP (blank)
F Without ISP

Speed Options

-15 15 ns pin-to-pin delay
-10 10 ns pin-to-pin delay
-7 7.5 ns pin-to-pin delay
-5 5 ns pin-to-pin delay

Packaging Options

PC44 44-Pin Plastic Leaded Chip Carrier (PLCC)
VQ44 44-Pin Thin Quad Pack (VQFP)

Temperature Options

C Commercial 0°C to 70°C
I Industrial –40°C to 85°C

3

Component Availability

Pins		44	
Type		Plastic PLCC	Plastic VQFP
Code		PC44	VQ44
XC9536	–15	C(I)	C(I)
	–10	C(I)	C(I)
	–7	C	C
	–5	C	C
XC9536F	–15	C(I)	C(I)
	–10	C(I)	C(I)
	–7	C	C
	–5	C	C

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

XC9572 In-System Programmable CPLD

January, 1997 (Version 1.0)

Preliminary Product Specification

Features

- 7.5 ns pin-to-pin logic delays on all pins
- f_{CNT} to 125 MHz
- 72 macrocells with 1,600 usable gates
- Up to 72 user I/O pins
- 5 V in-system programmable (ISP)
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- PCI compliant (-7, -10 speed grades)
- Advanced 0.6 µm CMOS 5V FastFLASH technology
- Supports parallel programming of more than one XC9500 concurrently
- Available in 84-pin PLCC, 100-pin PQFP and 100-pin TQFP packages
- Plug-in compatible, non-ISP XC9572F available in 84-pin PLCC and 100-pin PQFP packages

Description

The XC9572 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration. It is comprised of four 36V18 Function Blocks, providing 1,600 usable gates with propagation delays of 7.5 ns. See Figure 2 for the architecture overview.

Power Management

Power dissipation can be reduced in the XC9572 by configuring macrocells to standard or low-power modes of operation. Unused macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

I_{CC} (mA) =

MC_{HP} (1.7) + MC_{LP} (0.9) + MC (0.006 mA/MHz) f

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of macrocells used

f = Clock frequency (MHz)

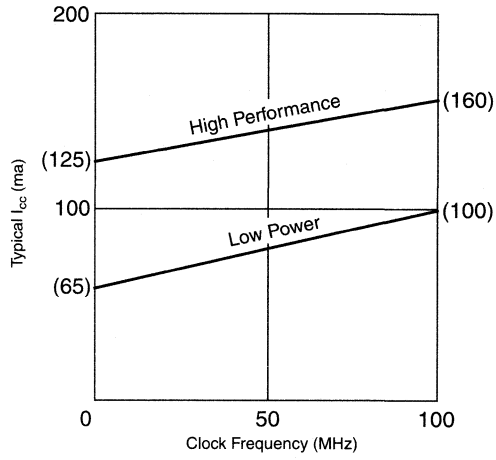


Figure 1: Typical I_{CC} vs. Frequency for XC9572

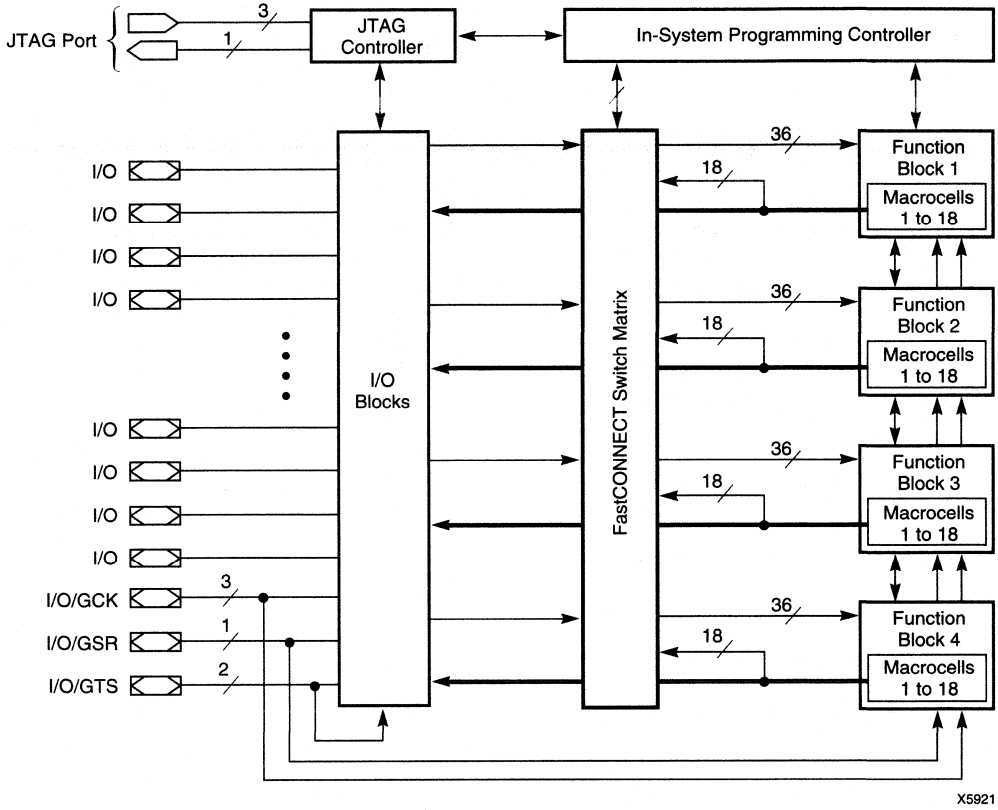


Figure 2: XC9572 Architecture

Note: Function Block outputs (indicated by the bold line) drive the I/O Blocks directly

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage relative to GND	-0.5 to 7.0	V
V_{IN}	DC input voltage relative to GND	-0.5 to V_{CC} + 0.5	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to V_{CC} + 0.5	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Max soldering temperature (10 ns @ 1/16 in = 1.5 mm)	+260	°C

Warning: 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 Rating conditions for extended periods of time may affect device reliability.

Recommended Operation Conditions[1]

Symbol	Parameter	Min	Max	Units
V_{CCINT}	Supply voltage for internal logic and input buffer	4.75 (4.5)	5.25 (5.5)	V
V_{CCIO}	Supply voltage for output drivers for 5 V operation	4.75 (4.5)	5.25 (5.5)	V
	Supply voltage for output drivers for 3.3 V operation	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.80	V
V_{IH}	High-level input voltage	2.0	V_{CCINT} +0.5	V
V_O	Output voltage	0	V_{CCINT} + 0.5	V

Note: 1. Numbers in parenthesis are for industrial temperature range versions.

Endurance Characteristics

Symbol	Parameter	Min	Max	Units
t_{DR}	Data Retention	20	-	Years
N_{PE}	Program/Erase Cycles	10,000	-	Cycles

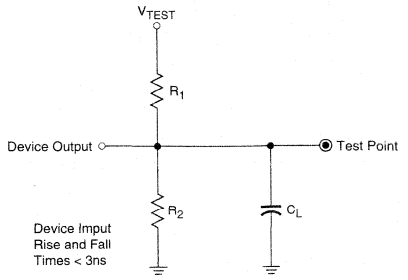
DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	Output high voltage for 5 V operation	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	Output high voltage for 3.3 V operation	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	Output low voltage for 5 V operation	I_{OL} = 24 mA V_{CC} = Min		0.5	V
	Output low voltage for 3.3 V operation	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
I_{IH}	I/O high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
C_{IN}	I/O capacitance	V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}	Operating Supply Current (low power mode, active)	V_I = GND, No load f = 1.0 MHz	65 (Typ)		ma

AC Characteristics

Symbol	Parameter	XC9572-7		XC9572-10		XC9572-15		Units
		Min	Max	Min	Max	Min	Max	
t_{PD}	I/O to output valid		7.5		10.0		15.0	ns
t_{SU}	I/O setup time before GCK	5.5		6.5		8.0		ns
t_{H}	I/O hold time after GCK	0.0		0.0		0.0		ns
t_{CO}	GCK to output valid		5.5		6.5		8.0	ns
f_{CNT}[1]	16-bit counter frequency	125		111		95		MHz
f_{SYSTEM}[2]	Multiple FB internal operating frequency	83		67		56		MHz
t_{PSU}	I/O setup time before p-term clock input	1.5		2.5		4.0		ns
t_{PH}	I/O hold time after p-term clock input	4.0		4.0		4.0		ns
t_{PCO}	P-term clock to output valid		9.5		10.5		12.0	ns
t_{OE}	GTS to output valid		7.0		10.0		15.0	ns
t_{OD}	GTS to output disable		7.0		10.0		15.0	ns
t_{POE}	Product term OE to output enabled		13.0		15.5		18.0	ns
t_{POD}	Product term OE to output disabled		13.0		15.5		18.0	ns
t_{WLH}	GCK pulse width (High or Low)		4.0		4.5		5.5	ns
		Preliminary						

Note: 1. f_{CNT} is the fastst 16-bit counter frequency available, using the local feedback when applicable. f_{CNT} is also the Export Control Maximum flip-flop toggle rate, f_{TOG}.
2. f_{SYSTEM} is the internal operating frequency for general purpose system designs spanning multiple FBs.



V_{CCIO} Level	V_{TEST}	R_1	R_1	C_L
5 V	5.0 V	160 Ω	120 Ω	35 pF
3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X5222

Figure 3: AC Load Circuit

Internal Timing Parameters

Symbol	Parameter	XC9572-7		XC9572-10		XC9572-15		Units
		Min	Max	Min	Max	Min	Max	
Buffer Delays								
t_{IN}	Input buffer delay		2.5		3.5		4.5	ns
t_{GCK}	GCK buffer delay		2.5		3.0		3.0	ns
t_{GSR}	GSR buffer delay		4.5		6.0		7.5	ns
t_{GTS}	GTS buffer delay		7.0		10.0		15.0	ns
t_{OUT}	Output buffer delay		2.5		3.0		4.5	ns
t_{EN}	Output buffer enable/disable delay		0.0		0.0		0.0	ns
Product Term Control Delays								
t_{PTCK}	Product term clock delay		4.0		3.5		2.5	ns
t_{PTSR}	Product term set/reset delay		2.0		2.5		3.0	ns
t_{PTTS}	Product term 3-state delay		10.5		12.0		13.5	ns
Internal Register and Combinatorial delays								
t_{PDI}	Combinatorial logic propagation delay		0.5		1.0		3.0	ns
t_{SUI}	Register setup time	3.5		3.5		3.5		ns
t_{HI}	Register hold time	2.0		3.0		4.5		ns
t_{COI}	Register clock to output valid time		0.5		0.5		0.5	ns
t_{AOI}	Register async. S/R to output delay		6.5		7.0		8.0	ns
t_{RAI}	Register async. S/R recovery before clock	7.5		10.0		15.0		ns
t_{LOGI}	Internal logic delay		2.0		2.5		3.0	ns
t_{LOGILP}	Internal low power logic delay		10.0		11.0		11.5	ns
Feedback Delays								
t_F	FastCONNECT matrix feedback delay		6.0		8.5		11.0	ns
t_{LF}	Function Block local feeback delay		2.0		2.5		3.5	ns
Time Adders								
t_{PTA}[3]	Incremental Product Term Allocator delay		1.0		1.0		1.5	ns
t_{SLEW}	Slew-rate limited delay		4.0		4.5		5.0	ns
		Preliminary						

Note: 3. t_{PTA} is multiplied by the span of the function as defined in the family data sheet.

XC9572 I/O Pins

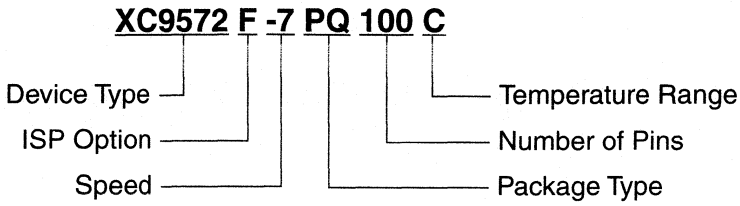
Function Block	Macrocell	PC84	PQ100	TQ100	BScan Order	Notes
1	1	4	18	16	213	
1	2	1	15	13	210	
1	3	6	20	18	207	
1	4	7	22	20	204	
1	5	2	16	14	201	
1	6	3	17	15	198	
1	7	11	27	25	195	
1	8	5	19	17	192	
1	9	9	24	22	189	[1]
1	10	13	30	28	186	
1	11	10	25	23	183	[1]
1	12	18	35	33	180	
1	13	20	38	36	177	
1	14	12	29	27	174	[1]
1	15	14	31	29	171	
1	16	23	41	39	168	
1	17	15	32	30	165	
1	18	24	42	40	162	
2	1	63	89	87	159	
2	2	69	96	94	156	
2	3	67	93	91	153	
2	4	68	95	93	150	
2	5	70	97	95	147	
2	6	71	98	96	144	
2	7	76	5	3	141	[1]
2	8	72	99	97	138	
2	9	74	1	99	135	[1]
2	10	75	3	1	132	
2	11	77	6	4	129	[1]
2	12	79	8	6	126	
2	13	80	10	8	123	
2	14	81	11	9	120	
2	15	83	13	11	117	
2	16	82	12	10	114	
2	17	84	14	12	111	
2	18	–	94	92	108	
3	1	25	43	41	105	
3	2	17	34	32	102	
3	3	31	51	49	99	
3	4	32	52	50	96	
3	5	19	37	35	93	
3	6	34	55	53	90	
3	7	35	56	54	87	
3	8	21	39	37	84	
3	9	26	44	42	81	
3	10	40	62	60	78	
3	11	33	54	52	75	
3	12	41	63	61	72	
3	13	43	65	63	69	
3	14	36	57	55	66	
3	15	37	58	56	63	
3	16	45	67	65	60	
3	17	39	60	58	57	
3	18	–	61	59	54	
4	1	46	68	66	51	
4	2	44	66	64	48	
4	3	51	73	71	45	
4	4	52	74	72	42	
4	5	47	69	67	39	
4	6	54	78	76	36	
4	7	55	79	77	33	
4	8	48	70	68	30	
4	9	50	72	70	27	
4	10	57	83	81	24	
4	11	53	76	74	21	
4	12	58	84	82	18	
4	13	61	87	85	15	
4	14	56	80	78	12	
4	15	65	91	89	9	
4	16	62	88	86	6	
4	17	66	92	90	3	
4	18	–	81	79	0	

Notes: [1] Global control pin

XC9572 Global, JTAG and Power Pins

Pin Type	PC84	PQ100	TQ100
I/O/GCK1	9	24	22
I/O/GCK2	10	25	23
I/O/GCK3	12	29	27
I/O/GTS1	76	5	3
I/O/GTS2	77	6	4
I/O/GSR	74	1	99
TCK	30	50	48
TDI	28	47	45
TDO	59	85	83
TMS	29	49	47
V_{CCINT} 5 V	38,73,78	7,59,100	5,57,98
V_{CCIO} 3.3 V/5 V	22,64	28,40,53,90	26,38,51,88
GND	8,16,27,42,49,60	2,23,33,46,64,71,77,86	100,21,31,44,62,69,75,84
No Connects	—	4,9,21,26,36,45,48,75,82	2,7,19,24,34,43,46,73,80

Ordering Information



ISP Options

__ With ISP (blank)
F Without ISP

Speed Options

-15 15 ns pin-to-pin delay
-10 10 ns pin-to-pin delay
-7 7.5 ns pin-to-pin delay

Packaging Options

PC84 84-Pin Plastic Leaded Chip Carrier (PLCC)
PQ100 100-Pin Plastic Quad Flat Pack (PQFP)
TQ100 100-Pin Very Thin Quad Flat Pack (TQFP)

Temperature Options

C Commercial 0°C to 70°C
I Industrial –40°C to 85°C

Component Availability

Pins		84	100	
Type		Plastic PLCC	Plastic PQFP	Plastic TQFP
Code		PC84	PQ100	TQ100
XC9572	–15	C(I)	C(I)	C(I)
	–10	C(I)	C(I)	C(I)
	–7	C	C	C
XC9572F	–15	C(I)	C(I)	-
	–10	C(I)	C(I)	-
	–7	C	C	-

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

XC95108 In-System Programmable CPLD

January, 1997 (Version 1.0)

Preliminary Product Specification

Features

- 7.5 ns pin-to-pin logic delays on all pins
- f_{CNT} to 125 MHz
- 108 macrocells with 2400 usable gates
- Up to 108 user I/O pins
- 5 V in-system programmable (ISP)
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- PCI compliant (-7, -10 speed grades)
- Advanced 0.6 µm CMOS 5V FastFLASH technology
- Supports parallel programming of more than one XC9500 concurrently
- Available in 84-pin PLCC, 100-pin PQFP, 100-pin TQFP and 160-pin PQFP packages
- Plug-in compatible, non-ISP XC95108F available in 84-pin PLCC, 100-pin PQFP, and 160-pin PQFP packages

Description

The XC95108 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration. It is comprised of six 36V18 Function Blocks, providing 2,400 usable gates with propagation delays of 7.5 ns. See Figure 2 for the architecture overview.

Power Management

Power dissipation can be reduced in the XC95108 by configuring macrocells to standard or low-power modes of operation. Unused macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

I_{CC} (mA) =

MC_{HP} (1.7) + MC_{LP} (0.9) + MC (0.006 mA/MHz) f

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of macrocells used

f = Clock frequency (MHz)

Figure 1 shows a typical calculation for the XC95108 device.

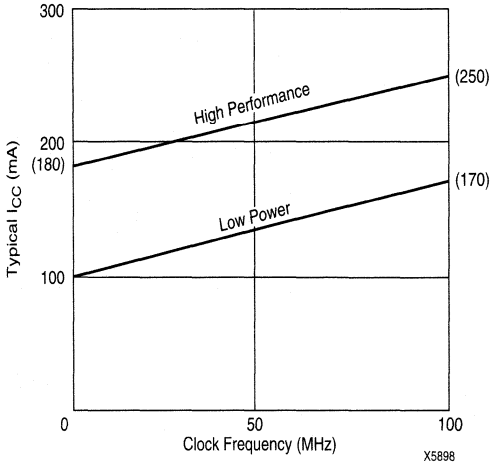


Figure 1: Typical I_{CC} vs. Frequency for XC95108

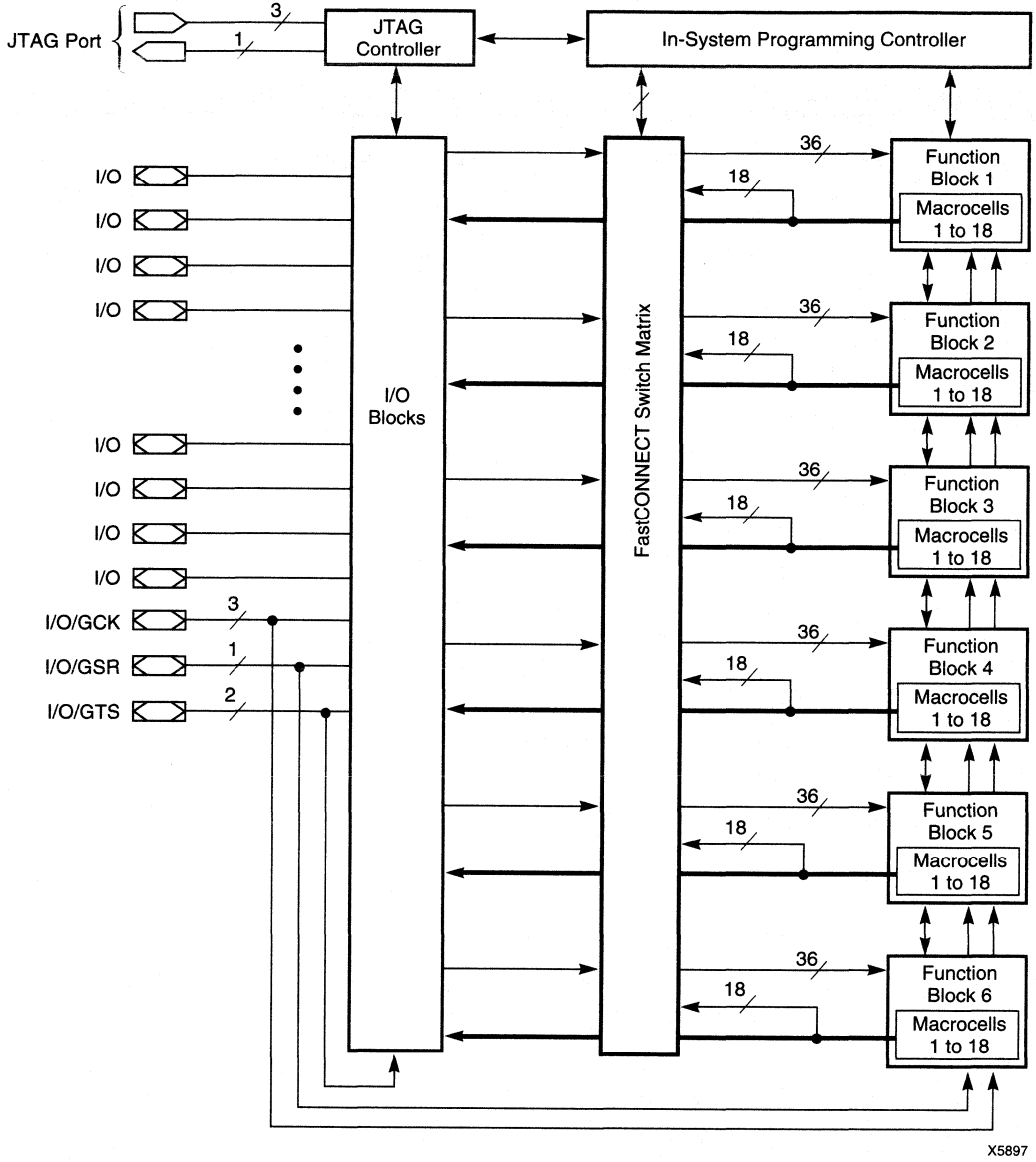


Figure 2: XC95108 Architecture

Note: Function Block outputs (indicated by the bold line) drive the I/O Blocks directly

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage relative to GND	-0.5 to 7.0	V
V_{IN}	DC input voltage relative to GND	-0.5 to V_{CC} + 0.5	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to V_{CC} + 0.5	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Max soldering temperature (10 ns @ 1/16 in = 1.5 mm)	+260	°C

Warning: 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 Rating conditions for extended periods of time may affect device reliability.

Recommended Operation Conditions[1]

Symbol	Parameter	Min	Max	Units
V_{CCINT}	Supply voltage for internal logic and input buffer	4.75 (4.5)	5.25 (5.5)	V
V_{CCIO}	Supply voltage for output drivers for 5 V operation	4.75 (4.5)	5.25 (5.5)	V
	Supply voltage for output drivers for 3.3 V operation	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.80	V
V_{IH}	High-level input voltage	2.0	V_{CCINT} +0.5	V
V_O	Output voltage	0	V_{CCINT} + 0.5	V

Note: 1. Numbers in parenthesis are for industrial-temperature range versions.

Endurance Characteristics

Symbol	Parameter	Min	Max	Units
t_{DR}	Data Retention	20	-	Years
N_{PE}	Program/Erase Cycles	10,000	-	Cycles

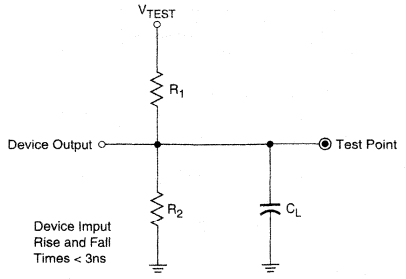
DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	Output high voltage for 5 V operation	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	Output high voltage for 3.3 V operation	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	Output low voltage for 5 V operation	I_{OL} = 24 mA V_{CC} = Min		0.5	V
	Output low voltage for 3.3 V operation	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
I_{IH}	I/O high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
C_{IN}	I/O capacitance	V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}	Operating Supply Current (low power mode, active)	V_I = GND, No load f = 1.0 MHz	100 (Typ)		ma

AC Characteristics

Symbol	Parameter	XC95108-7		XC95108-10		XC95108-15		XC95108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{PD}	I/O to output valid		7.5		10.0		15.0		20.0	ns
t_{SU}	I/O setup time before GCK	5.5		6.5		8.0		10.0		ns
t_H	I/O hold time after GCK	0.0		0.0		0.0		0.0		ns
t_{CO}	GCK to output valid		5.5		6.5		8.0		10.0	ns
f_{CNT}[1]	16-bit counter frequency	125		111		95		83		MHz
f_{SYSTEM}[2]	Multiple FB internal operating frequency	83		67		56		50		MHz
t_{PSU}	I/O setup time before p-term clock input	1.5		2.5		4.0		4.0		ns
t_{PH}	I/O hold time after p-term clock input	4.0		4.0		4.0		6.0		ns
t_{PCO}	P-term clock to output valid		9.5		10.5		12.0		16.0	ns
t_{OE}	GTS to output valid		7.0		10.0		15.0		20.0	ns
t_{OD}	GTS to output disable		7.0		10.0		15.0		20.0	ns
t_{POE}	Product term OE to output enabled		13.0		15.5		18.0		22.0	ns
t_{POD}	Product term OE to output disabled		13.0		15.5		18.0		22.0	ns
t_{WLH}	GCK pulse width (High or Low)		4.0		4.5		5.5		5.5	ns
		Preliminary								

Note: 1. f_{CNT} is the fastest 16-bit counter frequency available, using the local feedback when applicable. f_{CNT} is also the Export Control Maximum flip-flop toggle rate, f_{TOG}.
2. f_{SYSTEM} is the internal operating frequency for general purpose system designs spanning multiple FBs.



V_{CCIO} Level	V_{TEST}	R_1	R_1	C_L
5 V	5.0 V	160 Ω	120 Ω	35 pF
3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X5222

Figure 3: AC Load Circuit

Internal Timing Parameters

Symbol	Parameter	XC95108-7		XC95108-10		XC95108-15		XC95108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
Buffer Delays										
t_{IN}	Input buffer delay		2.5		3.5		4.5		6.5	ns
t_{GCK}	GCK buffer delay		2.5		3.0		3.0		3.0	ns
t_{GSR}	GSR buffer delay		4.5		6.0		7.5		9.5	ns
t_{GTS}	GTS buffer delay		7.0		10.0		15.0		20.0	ns
t_{OUT}	Output buffer delay		2.5		3.0		4.5		6.5	ns
t_{EN}	Output buffer enable/disable delay		0.0		0.0		0.0		0.0	ns
Product Term Control Delays										
t_{PTCK}	Product term clock delay		4.0		3.5		2.5		2.5	ns
t_{PTSR}	Product term set/reset delay		2.0		2.5		3.0		3.0	ns
t_{PTTS}	Product term 3-state delay		10.5		12.0		13.5		15.5	ns
Internal Register and Combinatorial delays										
t_{PDI}	Combinatorial logic propagation delay		0.5		1.0		3.0		4.0	ns
t_{SUI}	Register setup time	3.5		3.5		3.5		3.5		ns
t_{HI}	Register hold time	2.0		3.0		4.5		6.5		ns
t_{COI}	Register clock to output valid time		0.5		0.5		0.5		0.5	ns
t_{AOI}	Register async. S/R to output delay		6.5		7.0		8.0		9.0	ns
t_{RAI}	Register async. S/R recovery before clock	7.5		10.0		15.0		20.0		ns
t_{LOGI}	Internal logic delay		2.0		2.5		3.0		3.0	ns
t_{LOGILP}	Internal low power logic delay		10.0		11.0		11.5		11.5	ns
Feedback Delays										
t_F	FastCONNECT matrix feedback delay		6.0		8.5		11.0		13.0	ns
t_{LF}	Function Block local feeback delay		2.0		2.5		3.5		5.0	ns
Time Adders										
t_{PTA}[3]	Incremental Product Term Allocator delay		1.0		1.0		1.5		1.5	ns
t_{SLEW}	Slew-rate limited delay		4.0		4.5		5.0		5.5	ns
		Preliminary								

Note: 3. t_{PTA} is multiplied by the span of the function as defined in the family data sheet.

XC95108 I/O Pins

Function Block	Macrocell	PC84	PQ100	TQ100	PQ160	BScan Order	Notes	Function Block	Macrocell	PC84	PQ100	TQ100	PQ160	BScan Order	Notes
1	1	–	–	–	25	321		3	1	–	–	–	45	213	
1	2	1	15	13	21	318		3	2	14	31	29	47	210	
1	3	2	16	14	22	315		3	3	15	32	30	49	207	
1	4	–	21	19	29	312		3	4	–	36	34	57	204	
1	5	3	17	15	23	309		3	5	17	34	32	54	201	
1	6	4	18	16	24	306		3	6	18	35	33	56	198	
1	7	–	–	–	27	303		3	7	–	–	–	50	195	
1	8	5	19	17	26	300		3	8	19	37	35	58	192	
1	9	6	20	18	28	297		3	9	20	38	36	59	189	
1	10	–	26	24	36	294		3	10	–	45	43	69	186	
1	11	7	22	20	30	291		3	11	21	39	37	60	183	
1	12	9	24	22	33	288	[1]	3	12	23	41	39	62	180	
1	13	–	–	–	34	285		3	13	–	–	–	52	177	
1	14	10	25	23	35	282	[1]	3	14	24	42	40	63	174	
1	15	11	27	25	37	279		3	15	25	43	41	64	171	
1	16	12	29	27	42	276	[1]	3	16	26	44	42	68	168	
1	17	13	30	28	44	273		3	17	31	51	49	77	165	
1	18	–	–	–	43	270		3	18	–	–	–	74	162	
2	1	–	–	–	158	267		4	1	–	–	–	123	159	
2	2	71	98	96	154	264		4	2	57	83	81	134	156	
2	3	72	99	97	156	261		4	3	58	84	82	135	153	
2	4	–	4	2	4	258		4	4	–	82	80	133	150	
2	5	74	1	99	159	255	[1]	4	5	61	87	85	138	147	
2	6	75	3	1	2	252		4	6	62	88	86	139	144	
2	7	–	–	–	9	249		4	7	–	–	–	128	141	
2	8	76	5	3	6	246	[1]	4	8	63	89	87	140	138	
2	9	77	6	4	8	243	[1]	4	9	65	91	89	142	135	
2	10	–	9	7	12	240		4	10	–	–	–	147	132	
2	11	79	8	6	11	237		4	11	66	92	90	143	129	
2	12	80	10	8	13	234		4	12	67	93	91	144	126	
2	13	–	–	–	14	231		4	13	–	–	–	153	123	
2	14	81	11	9	15	228		4	14	68	95	93	146	120	
2	15	82	12	10	17	225		4	15	69	96	94	148	117	
2	16	83	13	11	18	222		4	16	–	94	92	145	114	
2	17	84	14	12	19	219		4	17	70	97	95	152	111	
2	18	–	–	–	16	216		4	18	–	–	–	155	108	

Notes: [1] Global control pin

XC95108 I/O Pins (continued)

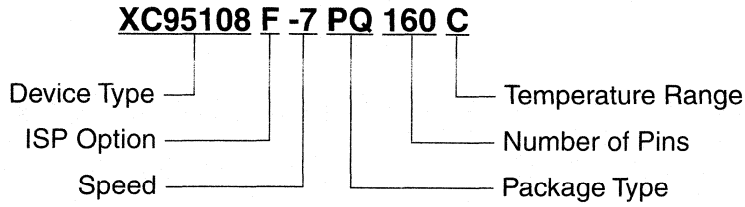
Function Block	Macrocell	PC84	PQ100	TQ100	PQ160	BScan Order	Notes
5	1	–	–	–	76	105	
5	2	32	52	50	79	102	
5	3	33	54	52	82	99	
5	4	–	48	46	72	96	
5	5	34	55	53	86	93	
5	6	35	56	54	88	90	
5	7	–	–	–	78	87	
5	8	36	57	55	90	84	
5	9	37	58	56	92	81	
5	10	–	–	–	84	78	
5	11	39	60	58	95	75	
5	12	40	62	60	97	72	
5	13	–	–	–	87	69	
5	14	41	63	61	98	66	
5	15	43	65	63	101	63	
5	16	–	61	59	96	60	
5	17	44	66	64	102	57	
5	18	–	–	–	89	54	

Function Block	Macrocell	PC84	PQ100	TQ100	PQ160	BScan Order	Notes
6	1	–	–	–	91	51	
6	2	45	67	65	103	48	
6	3	46	68	66	104	45	
6	4	–	75	73	116	42	
6	5	47	69	67	106	39	
6	6	48	70	68	108	36	
6	7	–	–	–	105	33	
6	8	50	72	70	111	30	
6	9	51	73	71	113	27	
6	10	–	–	–	107	24	
6	11	52	74	72	115	21	
6	12	53	76	74	117	18	
6	13	–	–	–	112	15	
6	14	54	78	76	122	12	
6	15	55	79	77	124	9	
6	16	–	81	79	129	6	
6	17	56	80	78	126	3	
6	18	–	–	–	114	0	

XC95108 Global, JTAG and Power Pins

Pin Type	PC84	PQ100	TQ100	PQ160
I/O/GCK1	9	24	22	33
I/O/GCK2	10	25	23	35
I/O/GCK3	12	29	27	42
I/O/GTS1	76	5	3	6
I/O/GTS2	77	6	4	8
I/O/GSR	74	1	99	159
TCK	30	50	48	75
TDI	28	47	45	71
TDO	59	85	83	136
TMS	29	49	47	73
V_{CCINT} 5 V	38,73,78	7,59,100	5,57,98	10,46,94,157
V_{CCIO} 3.3 V/5 V	22,64	28,40,53,90	26,38,51,88	1,41,61,81,121,141
GND	8,16,27,42,49,60	2,23,33,46,64,71,77,86	100,21,31,44,62,69,75,84	20,31,40,51,70,80,99
GND	–	–	–	100,110,120,127,137
GND	–	–	–	160

Ordering Information



ISP Options

__ With ISP (blank)
F Without ISP

Speed Options

- 20 20 ns pin-to-pin delay
-15 15 ns pin-to-pin delay
-10 10 ns pin-to-pin delay
-7 7 ns pin-to-pin delay

Packaging Options

PC84 84-Pin Plastic Leaded Chip Carrier (PLCC)
PQ100 100-Pin Plastic Quad Flat Pack (PQFP)
TQ100 100-Pin Very Thin Quad Flat Pack (TQFP)
PQ160 160-Pin Plastic Quad Flat Pack (PQFP)

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	–40°C to 85°C

Component Availability

Pins		84	100		160
Type		Plastic PLCC	Plastic PQFP	Plastic TQFP	Plastic PQFP
Code		PC84	PQ100	TQ100	PQ160
XC95108	–20	C(I)	C(I)	C(I)	C(I)
	–15	C(I)	C(I)	C(I)	C(I)
	–10	C(I)	C(I)	C(I)	C(I)
	–7	C	C	C	C
XC95108F	–20	C(I)	C(I)	-	C(I)
	–15	C(I)	C(I)	-	C(I)
	–10	C(I)	C(I)	-	C(I)
	–7	C	C	-	C

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

XC95144 In-System Programmable CPLD

January, 1997 (Version 1.0)

Advanced Product Specification

Features

- 7.5 ns pin-to-pin logic delays on all pins
- f_{CNT} to 111 MHz
- 144 macrocells with 3,200 usable gates
- Up to 133 user I/O pins
- 5 V in-system programmable
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- PCI compliant (-7, -10 speed grades)
- Advanced 0.6 µm CMOS 5V FastFLASH technology
- Supports parallel programming of more than one XC9500 concurrently
- Available in 100-pin PQFP, and 160-pin PQFP packages

Description

The XC95144 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration. It is comprised of eight 36V18 Function Blocks, providing 3,200 usable gates with propagation delays of 7.5 ns. See Figure 1 for the architecture overview.

Power Management

Power dissipation can be reduced in the XC95144 by configuring macrocells to standard or low-power modes of operation. Unused macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

I_{CC} (mA) =

MC_{HP} (1.7) + MC_{LP} (0.9) + MC (0.006 mA/MHz) f

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of macrocells used

f = Clock frequency (MHz)

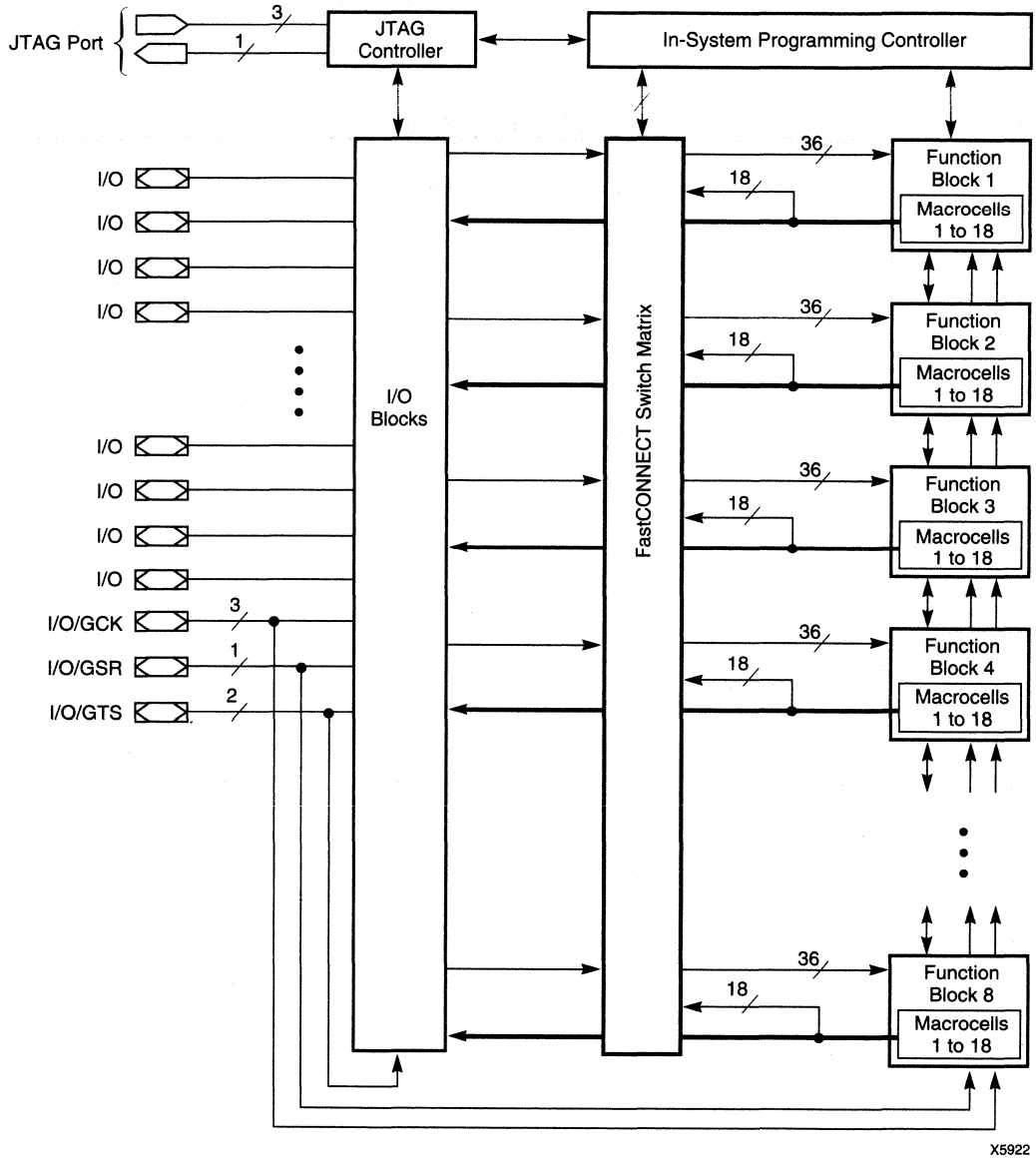


Figure 1: XC95144 Architecture

Function Block outputs (indicated by the bold line) drive the I/O Blocks directly

XC95144 I/O Pins

Function Block	Macrocell	PQ100	PQ160	BScan Order	Notes
1	1	–	38	429	
1	2	15	21	426	
1	3	16	22	423	
1	4	–	25	420	
1	5	17	23	417	
1	6	18	24	414	
1	7	–	32	411	
1	8	19	26	408	
1	9	20	28	405	
1	10	–	74	402	
1	11	21	29	399	
1	12	22	30	396	
1	13	–	39	393	
1	14	24	33	390	[1]
1	15	25	35	387	[1]
1	16	–	78	384	
1	17	26	36	381	
1	18	–	–	378	
2	1	–	3	375	
2	2	4	4	372	[1]
2	3	–	147	369	
2	4	–	158	366	
2	5	5	6	363	[1]
2	6	6	8	360	[1]
2	7	–	7	357	
2	8	8	11	354	
2	9	9	12	351	
2	10	–	155	348	
2	11	10	13	345	
2	12	11	15	342	
2	13	–	5	339	
2	14	12	17	336	
2	15	13	18	333	
2	16	–	105	330	
2	17	14	19	327	
2	18	–	–	324	

Function Block	Macrocell	PQ100	PQ160	BScan Order	Notes
3	1	–	53	321	
3	2	27	37	318	
3	3	–	84	315	
3	4	–	45	312	
3	5	29	42	309	[1]
3	6	30	44	306	
3	7	–	48	303	
3	8	31	47	300	
3	9	32	49	297	
3	10	–	89	294	
3	11	34	54	291	
3	12	35	56	288	
3	13	–	55	285	
3	14	36	57	282	
3	15	37	58	279	
3	16	–	34	276	
3	17	38	59	273	
3	18	–	–	270	
4	1	–	149	267	
4	2	92	143	264	
4	3	–	107	261	
4	4	–	123	258	
4	5	93	144	255	
4	6	94	145	252	
4	7	–	151	249	
4	8	95	146	246	
4	9	96	148	243	
4	10	–	114	240	
4	11	97	152	237	
4	12	98	154	234	
4	13	–	150	231	
4	14	99	156	228	
4	15	1	159	225	[1]
4	16	–	14	222	
4	17	3	2	219	[1]
4	18	–	–	216	

Notes: [1] Global control pin
Macrocell outputs to package pins subject to change, contact factory for latest information. Power, GND, JTAG and Global Signals are fixed.

XC95144 I/O Pins (continued)

Function Block	Macrocell	PQ100	PQ160	BScan Order	Notes
5	1	–	65	213	
5	2	39	60	210	
5	3	–	27	207	
5	4	–	76	204	
5	5	41	62	201	
5	6	42	63	198	
5	7	–	67	195	
5	8	43	64	192	
5	9	44	68	189	
5	10	–	93	186	
5	11	45	69	183	
5	12	48	72	180	
5	13	–	66	177	
5	14	51	77	174	
5	15	52	79	171	
5	16	–	52	168	
5	17	54	82	165	
5	18	–	–	162	
6	1	–	–	159	
6	2	79	124	156	
6	3	–	9	153	
6	4	–	91	150	
6	5	80	126	147	
6	6	81	129	144	
6	7	–	131	141	
6	8	82	133	138	
6	9	83	134	135	
6	10	–	130	132	
6	11	84	135	129	
6	12	87	138	126	
6	13	–	132	123	
6	14	88	139	120	
6	15	89	140	117	
6	16	–	153	114	
6	17	91	142	111	
6	18	–	–	108	

Function Block	Macrocell	PQ100	PQ160	BScan Order	Notes
7	1	–	–	105	
7	2	55	86	102	
7	3	–	50	99	
7	4	–	43	96	
7	5	56	88	93	
7	6	57	90	90	
7	7	–	83	87	
7	8	58	92	84	
7	9	60	95	81	
7	10	–	109	78	
7	11	61	96	75	
7	12	62	97	72	
7	13	–	85	69	
7	14	63	98	66	
7	15	65	101	63	
7	16	–	87	60	
7	17	66	102	57	
7	18	–	–	54	
8	1	–	–	51	
8	2	67	103	48	
8	3	–	128	45	
8	4	–	16	42	
8	5	68	104	39	
8	6	69	106	36	
8	7	–	118	33	
8	8	70	108	30	
8	9	72	111	27	
8	10	–	125	24	
8	11	73	113	21	
8	12	74	115	18	
8	13	–	119	15	
8	14	75	116	12	
8	15	76	117	9	
8	16	–	112	6	
8	17	78	122	3	
8	18	–	–	0	

XC95144 Global, JTAG and Power Pins

Pin Type	PQ100	PQ160
I/O/GCK1	24	33
I/O/GCK2	25	35
I/O/GCK3	29	42
I/O/GTS1	5	6
I/O/GTS2	6	8
I/O/GTS3	3	2
I/O/GTS4	4	4
I/O/GSR	1	159
TCK	50	75
TDI	47	71
TDO	85	136
TMS	49	73
V_{CCINT} 5 V	7,59,100	10,46,94,157
V_{CCIO} 3.3 V/5 V	28,40,53,90	1,41,61,81,121,141
GND	2,23,33,46,64,71, 77,86	20,31,40,51,70,80, 99,100,110,120,127, 137,160
No Connects	–	–

XC95180 In-System Programmable CPLD

January, 1997 (Version 1.0)

Advanced Product Specification

Features

- 10 ns pin-to-pin logic delays on all pins
- f_{CNT} to 111 MHz
- 180 macrocells with 4,000 usable gates
- Up to 166 user I/O pins
- 5 V in-system programmable
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- PCI compliant (-10 speed grade)
- Advanced 0.6 µm CMOS 5V FastFLASH technology
- Supports parallel programming of more than one XC9500 concurrently
- Available in 160-pin PQFP, and 208-pin HQFP packages

Description

The XC95180 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration. It is comprised of ten 36V18 Function Blocks, providing 4,000 usable gates with propagation delays of 10 ns. See Figure 1 for the architecture overview.

Power Management

Power dissipation can be reduced in the XC95180 by configuring macrocells to standard or low-power modes of operation. Unused macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

I_{CC} (mA) =

MC_{HP} (1.7) + MC_{LP} (0.9) + MC (0.006 mA/MHz) f

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of macrocells used

f = Clock frequency (MHz)

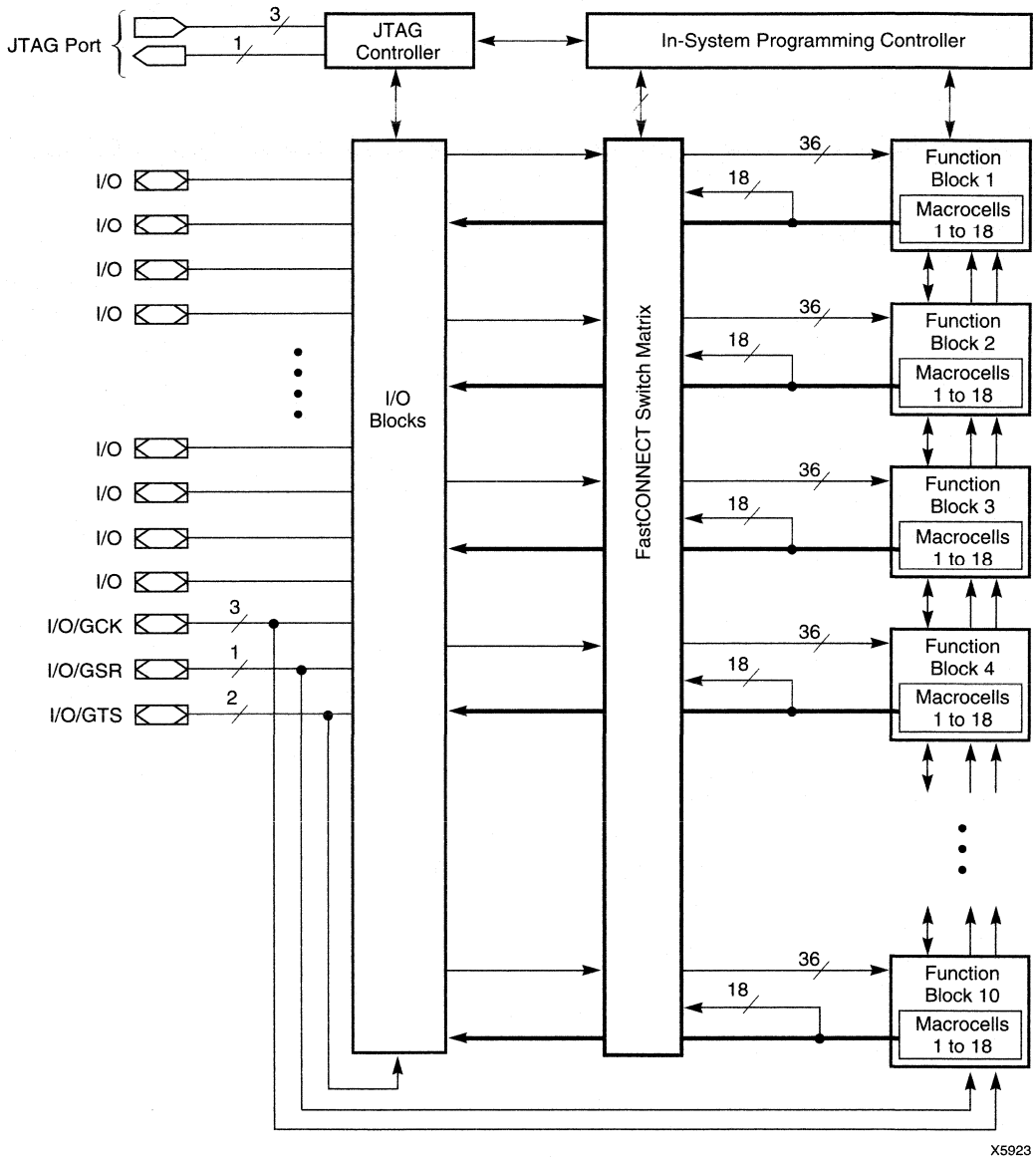


Figure 1: XC95180 Architecture

Function Block outputs (indicated by the bold line) drive the I/O Blocks directly

XC95180 I/O Pins

Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
1	1	–	39	537	
1	2	22	30	534	
1	3	23	31	531	
1	4	24	32	528	
1	5	25	33	525	
1	6	26	34	522	
1	7	–	40	519	
1	8	27	35	516	
1	9	28	36	513	
1	10	29	37	510	
1	11	30	38	507	
1	12	32	43	504	
1	13	–	41	501	
1	14	33	44	498	[1]
1	15	34	45	495	
1	16	35	46	492	[1]
1	17	36	47	489	
1	18	–	–	486	
2	1	–	14	483	
2	2	6	7	480	[1]
2	3	7	8	477	
2	4	8	9	474	[1]
2	5	9	10	471	
2	6	11	15	468	
2	7	–	28	465	
2	8	12	16	462	
2	9	13	17	459	
2	10	14	18	456	
2	11	15	19	453	
2	12	16	20	450	
2	13	–	29	447	
2	14	17	21	444	
2	15	18	22	441	
2	16	19	23	438	
2	17	21	25	435	
2	18	–	–	432	
3	1	–	48	429	
3	2	37	49	426	
3	3	38	50	423	
3	4	39	51	420	
3	5	42	55	417	[1]
3	6	43	56	414	
3	7	–	54	411	
3	8	44	57	408	
3	9	45	58	405	
3	10	47	60	402	
3	11	48	61	399	
3	12	49	63	396	
3	13	–	62	393	
3	14	50	64	390	
3	15	52	70	387	
3	16	53	71	384	
3	17	56	74	381	
3	18	–	–	387	
4	1	–	196	375	
4	2	150	194	372	
4	3	151	197	369	
4	4	152	198	366	
4	5	153	199	363	
4	6	154	200	360	
4	7	–	203	357	
4	8	155	201	354	
4	9	156	202	351	
4	10	–	208	348	
4	11	158	205	345	
4	12	159	206	342	[1]
4	13	–	12	339	
4	14	2	3	336	[1]
4	15	3	4	333	
4	16	4	5	330	[1]
4	17	5	6	327	
4	18	–	–	324	

Notes: [1] Global control pin
Macrocell outputs to package pins subject to change, contact factory for latest information. Power, GND, JTAG and Global Signals are fixed.

XC95180 I/O Pins (continued)

Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
5	1	–	–	321	
5	2	54	72	318	
5	3	55	73	315	
5	4	57	75	312	
5	5	58	76	309	
5	6	59	77	306	
5	7	–	67	303	
5	8	60	78	300	
5	9	62	82	297	
5	10	–	–	294	
5	11	63	83	291	
5	12	64	84	288	
5	13	–	80	285	
5	14	65	85	282	
5	15	66	86	279	
5	16	67	87	276	
5	17	68	88	273	
5	18	–	–	270	
6	1	–	169	267	
6	2	134	174	264	
6	3	135	175	261	
6	4	138	178	258	
6	5	139	179	255	
6	6	140	180	252	
6	7	–	183	249	
6	8	142	182	246	
6	9	143	185	243	
6	10	–	189	240	
6	11	144	186	237	
6	12	145	187	234	
6	13	–	195	231	
6	14	146	188	228	
6	15	147	191	225	
6	16	148	192	222	
6	17	149	193	219	
6	18	–	–	216	

Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
7	1	–	90	213	
7	2	69	89	210	
7	3	72	95	207	
7	4	74	97	204	
7	5	76	99	201	
7	6	77	100	198	
7	7	–	91	195	
7	8	78	102	192	
7	9	79	103	189	
7	10	–	101	186	
7	11	82	110	183	
7	12	83	111	180	
7	13	–	106	177	
7	14	84	112	174	
7	15	85	113	171	
7	16	86	114	168	
7	17	87	115	165	
7	18	–	–	162	
8	1	–	144	159	
8	2	118	154	156	
8	3	119	155	153	
8	4	122	158	150	
8	5	123	159	147	
8	6	124	160	144	
8	7	–	151	141	
8	8	125	161	138	
8	9	126	162	135	
8	10	–	165	132	
8	11	128	164	129	
8	12	129	166	126	
8	13	–	168	123	
8	14	130	167	120	
8	15	131	170	117	
8	16	132	171	114	
8	17	133	173	111	
8	18	–	–	108	

XC95180 I/O Pins (continued)

Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes	Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
9	1	–	–	105		10	1	–	–	51	
9	2	88	116	102		10	2	104	135	48	
9	3	89	117	99		10	3	105	136	45	
9	4	90	118	96		10	4	106	137	42	
9	5	91	121	93		10	5	107	138	39	
9	6	92	122	90		10	6	108	139	36	
9	7	–	107	87		10	7	–	120	33	
9	8	93	123	84		10	8	109	140	30	
9	9	95	125	81		10	9	111	145	27	
9	10	–	109	78		10	10	–	142	24	
9	11	96	126	75		10	11	112	146	21	
9	12	97	127	72		10	12	113	147	18	
9	13	–	119	69		10	13	–	143	15	
9	14	98	128	66		10	14	114	148	12	
9	15	101	131	63		10	15	115	149	9	
9	16	102	133	60		10	16	116	150	6	
9	17	103	134	57		10	17	117	152	3	
9	18	–	–	54		10	18	–	–	0	

XC95180 Global, JTAG and Power Pins

Pin Type	PQ160	HQ208
I/O/GCK1	33	44
I/O/GCK2	35	46
I/O/GCK3	42	55
I/O/GTS1	6	7
I/O/GTS2	8	9
I/O/GTS3	2	3
I/O/GTS4	4	5
I/O/GSR	159	206
TCK	75	98
TDI	71	94
TDO	136	176
TMS	73	96
V_{CCINT} 5 V	10,46,94,157	11,59,124,153,204
V_{CCIO} 3.3 V/5 V	1,41,61,81,121,141	1,26,53,65,79,92,105, 132,157,172,181,184
GND	20,31,40,51,70,80, 99,100,110,120,127, 137,160	2,13,24,27,42,52,66,68, 69,81,93,104,108,129, 130,141,156,163,177, 190,207
No Connects	–	–

XC95216 In-System Programmable CPLD

January, 1997 (Version 1.0)

Preliminary Product Specification

Features

- 10 ns pin-to-pin logic delays on all pins
- f_{CNT} to 111 MHz
- 216 macrocells with 4800 usable gates
- Up to 166 user I/O pins
- 5 V in-system programmable
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- PCI compliant (-10 speed grade)
- Advanced 0.6 µm CMOS 5V FastFLASH technology
- Supports parallel programming of more than one XC9500 concurrently
- Available in 160-pin PQFP and 208-pin HQFP packages

Description

The XC95216 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration. It is comprised of twelve 36V18 Function Blocks, providing 4,800 usable gates with propagation delays of 10 ns. See Figure 2 for the architecture overview.

Power Management

Power dissipation can be reduced in the XC95216 by configuring macrocells to standard or low-power modes of operation. Unused macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

I_{CC} (mA) =

MC_{HP} (1.7) + MC_{LP} (0.9) + MC (0.006 mA/MHz) f

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of macrocells used

f = Clock frequency (MHz)

Figure 1 shows a typical calculation for the XC95216 device.

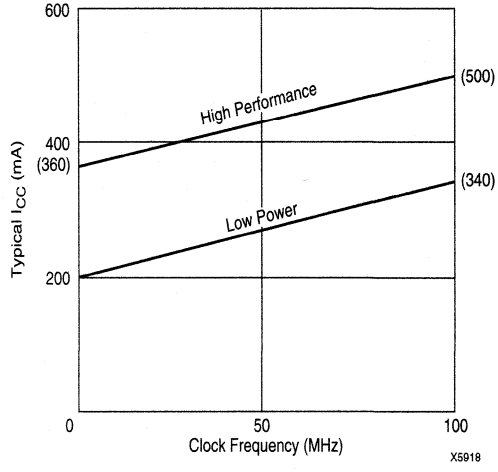


Figure 1: Typical I_{CC} vs. Frequency For XC95216

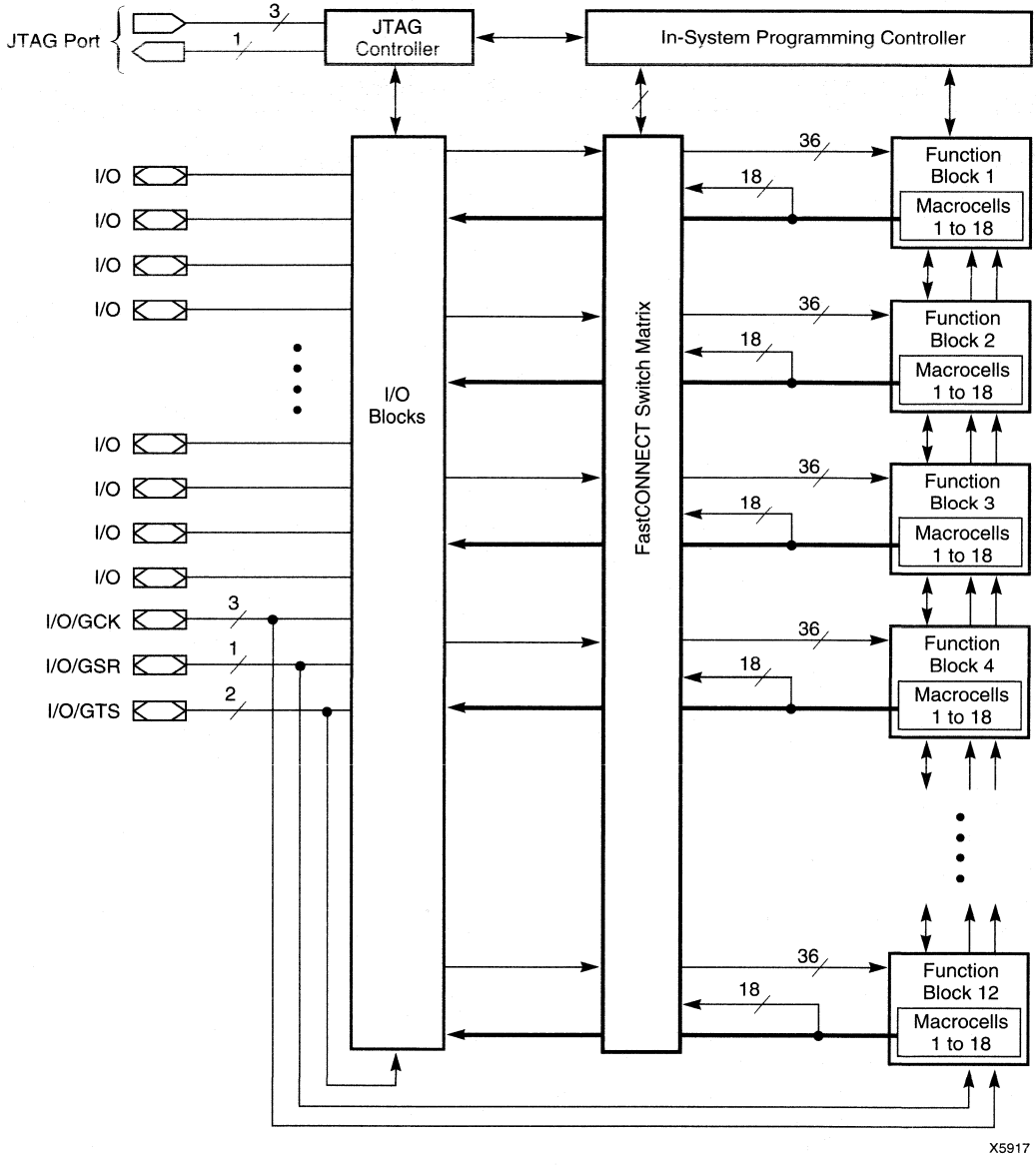


Figure 2: XC95216 Architecture

Function Block outputs (indicated by the bold line) drive the I/O Blocks directly

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage relative to GND	-0.5 to 7.0	V
V_{IN}	DC input voltage relative to GND	-0.5 to V_{CC} + 0.5	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to V_{CC} + 0.5	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Max soldering temperature (10 ns @ 1/16 in = 1.5 mm)	+260	°C

Warning: 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 Rating conditions for extended periods of time may affect device reliability.

Recommended Operating Conditions[1]

Symbol	Parameter	Min	Max	Units
V_{CCINT}	Supply voltage for internal logic and input buffer	4.75 (4.5)	5.25 (5.5)	V
V_{CCIO}	Supply voltage for output drivers for 5 V operation	4.75 (4.5)	5.25 (5.5)	V
	Supply voltage for output drivers for 3.3 V operation	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.80	V
V_{IH}	High-level input voltage	2.0	V_{CCINT} +0.5	V
V_O	Output voltage	0	V_{CCINT} + 0.5	V

Note: 1. Numbers in parenthesis are for industrial-temperature range versions.

Endurance Characteristics

Symbol	Parameter	Min	Max	Units
t_{DR}	Data Retention	20	-	Years
N_{PE}	Program/Erase Cycles	10,000	-	Cycles

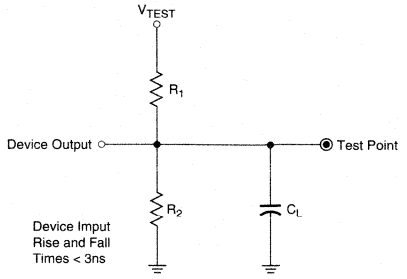
DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	Output high voltage for 5 V operation	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	Output high voltage for 3.3 V operation	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	Output low voltage for 5 V operation	I_{OL} = 24 mA V_{CC} = Min		0.5	V
	Output low voltage for 3.3 V operation	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
I_{IH}	I/O high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
C_{IN}	I/O capacitance	V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}	Operating Supply Current (low power mode, active)	V_I = GND, No load f = 1.0 MHz	200 (typ)		ma

AC Characteristics

Symbol	Parameter	XC95216-10		XC95216-15		XC95216-20		Units
		Min	Max	Min	Max	Min	Max	
t_{PD}	I/O to output valid		10.0		15.0		20.0	ns
t_{SU}	I/O setup time before GCK	6.5		8.0		10.0		ns
t_H	I/O hold time after GCK	0.0		0.0		0.0		ns
t_{CO}	GCK to output valid		6.5		8.0		10.0	ns
f_{CNT}[1]	16-bit counter frequency	111		95		83		MHz
f_{SYSTEM}[2]	Multiple FB internal operating frequency	67		56		50		MHz
t_{PSU}	I/O setup time before p-term clock input	2.5		4.0		4.0		ns
t_{PH}	I/O hold time after p-term clock input	4.0		4.0		6.0		ns
t_{PCO}	P-term clock to output valid		10.5		12.0		16.0	ns
t_{OE}	GTS to output valid		10.0		15.0		20.0	ns
t_{OD}	GTS to output disable		10.0		15.0		20.0	ns
t_{POE}	Product term OE to output enabled		15.5		18.0		22.0	ns
t_{POD}	Product term OE to output disabled		15.5		18.0		22.0	ns
t_{WLH}	GCK pulse width (High or Low)		4.5		5.5		5.5	ns
		Preliminary						

Note: 1. f_{CNT} is the fastest 16-bit counter frequency available, using the local feedback when applicable.
f_{CNT} is also the Export Control Maximum flip-flop toggle rate, f_{TOG}.
2. f_{SYSTEM} is the internal operating frequency for general purpose system designs spanning multiple FBs.



V_{CCIO} Level	V_{TEST}	R_1	R_1	C_L
5 V	5.0 V	160 Ω	120 Ω	35 pF
3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X5222

Figure 3: AC Load Circuit

Internal Timing Parameters

Symbol	Parameter	XC95216-10		XC95216-15		XC95216-20		Units
		Min	Max	Min	Max	Min	Max	
Buffer Delays								
t_{IN}	Input buffer delay		3.5		4.5		6.5	ns
t_{GCK}	GCK buffer delay		3.0		3.0		3.0	ns
t_{GSR}	GSR buffer delay		6.0		7.5		9.5	ns
t_{GTS}	GTS buffer delay		10.0		15.0		20.0	ns
t_{OUT}	Output buffer delay		3.0		4.5		6.5	ns
t_{EN}	Output buffer enable/disable delay		0.0		0.0		0.0	ns
Product Term Control Delays								
t_{PTCK}	Product term clock delay		3.5		2.5		2.5	ns
t_{PTSR}	Product term set/reset delay		2.5		3.0		3.0	ns
t_{PTTS}	Product term 3-state delay		12.0		13.5		15.5	ns
Internal Register and Combinatorial delays								
t_{PDI}	Combinatorial logic propagation delay		1.0		3.0		4.0	ns
t_{SUI}	Register setup time	3.5		3.5		3.5		ns
t_{HI}	Register hold time	3.0		4.5		6.5		ns
t_{COI}	Register clock to output valid time		0.5		0.5		0.5	ns
t_{AOI}	Register async. S/R to output delay		7.0		8.0		9.0	ns
t_{RAI}	Register async. S/R recovery before clock	10.0		15.0		20.0		ns
t_{LOGI}	Internal logic delay		2.5		3.0		3.0	ns
t_{LOGILP}	Internal low power logic delay		11.0		11.5		11.5	ns
Feedback Delays								
t_F	FastCONNECT matrix feedback delay		8.5		11.0		13.0	ns
t_{LF}	Function Block local feeback delay		2.5		3.5		5.0	ns
Time Adders								
t_{PTA}[3]	Incremental Product Term Allocator delay		1.0		1.5		1.5	ns
t_{SLEW}	Slew-rate limited delay		4.5		5.0		5.5	ns
		Preliminary						

Note: 3. t_{PTA} is multiplied by the span of the function as defined in the family data sheet.

XC95216 I/O Pins

Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
1	1	–	–	645	
1	2	18	22	642	
1	3	19	23	639	
1	4	–	28	636	
1	5	21	25	633	
1	6	22	30	630	
1	7	–	–	627	
1	8	23	31	624	
1	9	24	32	621	
1	10	–	12	618	
1	11	25	33	615	
1	12	26	34	612	
1	13	–	–	609	
1	14	27	35	606	
1	15	28	36	603	
1	16	29	37	600	
1	17	30	38	597	
1	18	–	–	594	
2	1	–	–	591	
2	2	6	7	588	[1]
2	3	7	8	585	
2	4	–	29	582	
2	5	8	9	579	[1]
2	6	9	10	576	
2	7	–	–	573	
2	8	11	15	570	
2	9	12	16	567	
2	10	–	-	564	
2	11	13	17	561	
2	12	14	18	558	
2	13	–	–	555	
2	14	15	19	552	
2	15	16	20	549	
2	16	–	14	546	
2	17	17	21	543	
2	18	–	–	540	

Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
3	1	–	–	537	
3	2	32	43	534	
3	3	33	44	531	[1]
3	4	–	39	528	
3	5	34	45	525	
3	6	35	46	522	[1]
3	7	–	–	519	
3	8	36	47	516	
3	9	37	49	513	
3	10	–	67	510	
3	11	38	50	507	
3	12	39	51	504	
3	13	–	–	501	
3	14	42	55	498	[1]
3	15	43	56	495	
3	16	–	80	492	
3	17	44	57	489	
3	18	–	–	486	
4	1	–	–	483	
4	2	152	198	480	
4	3	153	199	477	
4	4	–	196	474	
4	5	154	200	471	
4	6	155	201	468	
4	7	–	–	465	
4	8	156	202	462	
4	9	158	205	459	
4	10	–	-	456	
4	11	159	206	453	[1]
4	12	2	3	450	[1]
4	13	–	–	447	
4	14	3	4	444	
4	15	4	5	441	[1]
4	16	–	203	438	
4	17	5	6	435	
4	18	–	–	432	

XC95216 I/O Pins (continued)

Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
5	1	–	–	429	
5	2	45	58	426	
5	3	47	60	423	
5	4	–	41	420	
5	5	48	61	417	
5	6	49	63	414	
5	7	–	–	411	
5	8	50	64	408	
5	9	52	70	405	
5	10	–	109	402	
5	11	53	71	399	
5	12	54	72	396	
5	13	–	–	393	
5	14	55	73	390	
5	15	56	74	387	
5	16	–	40	384	
5	17	57	75	381	
5	18	–	–	378	
6	1	–	–	375	
6	2	140	180	372	
6	3	142	182	369	
6	4	–	208	366	
6	5	143	185	363	
6	6	144	186	360	
6	7	–	–	357	
6	8	145	187	354	
6	9	146	188	351	
6	10	–	183	348	
6	11	147	191	345	
6	12	148	192	342	
6	13	–	–	339	
6	14	149	193	336	
6	15	150	194	333	
6	16	–	169	330	
6	17	151	197	327	
6	18	–	–	324	

Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
7	1	–	–	321	
7	2	58	76	318	
7	3	59	77	315	
7	4	–	54	312	
7	5	60	78	309	
7	6	62	82	306	
7	7	–	–	303	
7	8	63	83	300	
7	9	64	84	297	
7	10	–	91	294	
7	11	65	85	291	
7	12	66	86	288	
7	13	–	–	285	
7	14	67	87	282	
7	15	68	88	279	
7	16	–	48	276	
7	17	69	89	273	
7	18	–	–	270	
8	1	–	–	267	
8	2	126	162	264	
8	3	128	164	261	
8	4	–	143	258	
8	5	129	166	255	
8	6	130	167	252	
8	7	–	–	249	
8	8	131	170	246	
8	9	132	171	243	
8	10	–	195	240	
8	11	133	173	237	
8	12	134	174	234	
8	13	–	–	231	
8	14	135	175	228	
8	15	138	178	225	
8	16	–	189	222	
8	17	139	179	219	
8	18	–	–	216	

XC95216 I/O Pins (continued)

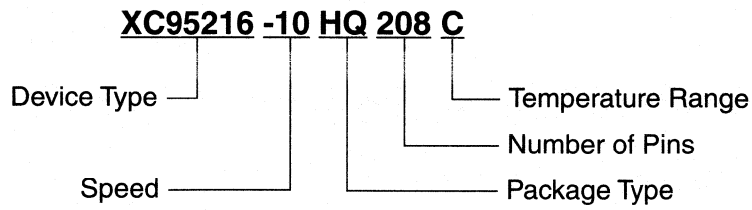
Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
9	1	–	–	213	
9	2	72	95	210	
9	3	74	97	207	
9	4	–	101	204	
9	5	76	99	201	
9	6	77	100	198	
9	7	–	–	195	
9	8	78	102	192	
9	9	79	103	189	
9	10	–	90	186	
9	11	82	110	183	
9	12	83	111	180	
9	13	–	–	177	
9	14	84	112	174	
9	15	85	113	171	
9	16	–	62	168	
9	17	86	114	165	
9	18	–	–	162	
10	1	–	–	159	
10	2	113	147	156	
10	3	114	148	153	
10	4	–	144	150	
10	5	115	149	147	
10	6	116	150	144	
10	7	–	–	141	
10	8	117	152	138	
10	9	118	154	135	
10	10	–	168	132	
10	11	119	155	129	
10	12	122	158	126	
10	13	–	–	123	
10	14	123	159	120	
10	15	124	160	117	
10	16	–	165	114	
10	17	125	161	111	
10	18	–	–	108	

Function Block	Macrocell	PQ160	HQ208	BScan Order	Notes
11	1	–	–	105	
11	2	87	115	102	
11	3	88	116	99	
11	4	–	119	96	
11	5	89	117	93	
11	6	90	118	90	
11	7	–	–	87	
11	8	91	121	84	
11	9	92	122	81	
11	10	–	107	78	
11	11	93	123	75	
11	12	95	125	72	
11	13	–	–	69	
11	14	96	126	66	
11	15	97	127	63	
11	16	–	120	60	
11	17	98	128	57	
11	18	–	–	54	
12	1	–	–	51	
12	2	101	131	48	
12	3	102	133	45	
12	4	–	106	42	
12	5	103	134	39	
12	6	104	135	36	
12	7	–	–	33	
12	8	105	136	30	
12	9	106	137	27	
12	10	–	151	24	
12	11	107	138	21	
12	12	108	139	18	
12	13	–	–	15	
12	14	109	140	12	
12	15	111	145	9	
12	16	–	142	6	
12	17	112	146	3	
12	18	–	–	0	

XC95216 Global, JTAG and Power Pins

Pin Type	PQ160	HQ208
I/O/GCK1	33	44
I/O/GCK2	35	46
I/O/GCK3	42	55
I/O/GTS1	6	7
I/O/GTS2	8	9
I/O/GTS3	2	3
I/O/GTS4	4	5
I/O/GSR	159	206
TCK	75	98
TDI	71	94
TDO	136	176
TMS	73	96
V_{CCINT} 5 V	10,46,94,157	11, 59, 124, 153, 204
V_{CCIO} 3.3 V/5 V	1,41,61,81,121,141	1, 26, 53, 65, 79, 92, 105, 132, 157, 172, 181, 184
GND	20, 31, 40, 51, 70, 80, 99, 100, 110, 120, 127, 137, 160	2, 13, 24, 27, 42, 52, 66, 68, 69, 81, 93, 104, 108, 129, 130, 141, 156, 163, 177, 190, 207
No Connects	–	–

Ordering Information



Speed Options

- 20 20 ns pin-to-pin delay
-15 15 ns pin-to-pin delay
-10 10 ns pin-to-pin delay

Packaging Options

PQ160 160-Pin Plastic Quad Flat Pack (PQFP)
HQ208 208-Pin Heat Sink Quad Flat Pack (HQFP)

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	–40°C to 85°C

Component Availability

Pins		160	160
Type		Plastic PQFP	Power QFP
Code		PQ160	HQ208
XC95216	–20	C(I)	C(I)
	–15	C	C
	–10	C	C

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

XC95288 In-System Programmable CPLD

January, 1997 (Version 1.0)

Advanced Product Specification

Features

- 10 ns pin-to-pin logic delays on all pins
- f_{CNT} to 111 MHz
- 288 macrocells with 6,400 usable gates
- Up to 192 user I/O pins
- 5 V in-system programmable
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- PCI compliant (-10 speed grade)
- Advanced 0.6 µm CMOS 5V FastFLASH technology
- Supports parallel programming of more than one XC9500 concurrently
- Available in a 208-pin HQFP and 352-pin BGA packages

Description

The XC95288 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration. It is comprised of sixteen 36V18 Function Blocks, providing 6,400 usable gates with propagation delays of 10 ns. See Figure 1 for the architecture overview.

Power Management

Power dissipation can be reduced in the XC95288 by configuring macrocells to standard or low-power modes of operation. Unused macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

I_{CC} (mA) =

MC_{HP} (1.7) + MC_{LP} (0.9) + MC (0.006 mA/MHz) f

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of macrocells used

f = Clock frequency (MHz)

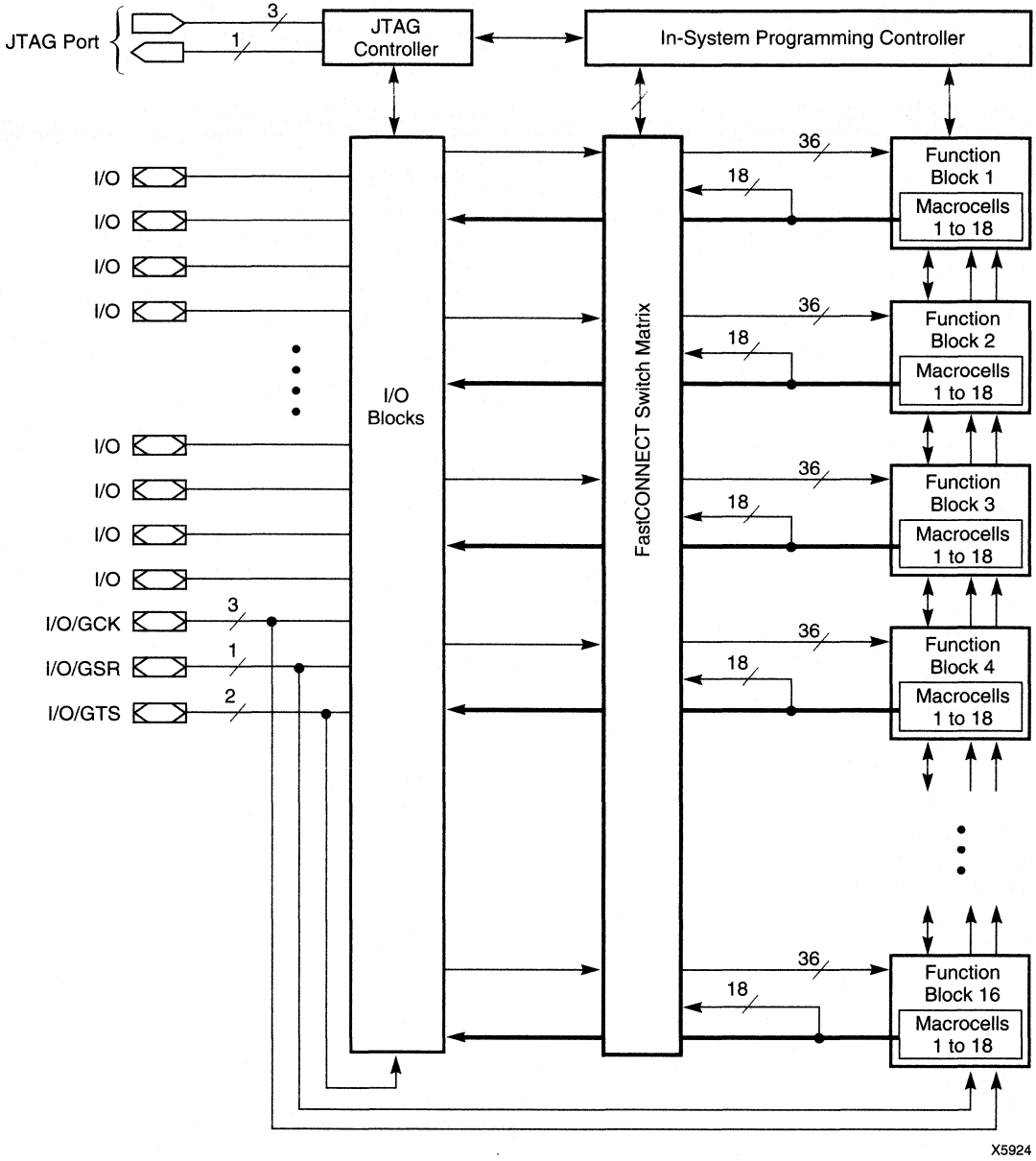


Figure 1: XC95288 Architecture

Note: Function Block outputs (indicated by the bold line) drive the I/O Blocks directly

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage relative to GND	-0.5 to 7.0	V
V_{IN}	DC input voltage relative to GND	-0.5 to V_{CC} + 0.5	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to V_{CC} + 0.5	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Max soldering temperature (10 ns @ 1/16 in = 1.5 mm)	+260	°C

Warning: 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 Rating conditions for extended periods of time may affect device reliability.

Recommended Operation Conditions[1]

Symbol	Parameter	Min	Max	Units
V_{CCINT}	Supply voltage for internal logic and input buffer	4.75 (4.5)	5.25 (5.5)	V
V_{CCIO}	Supply voltage for output drivers for 5 V operation	4.75 (4.5)	5.25 (5.5)	V
	Supply voltage for output drivers for 3.3 V operation	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.80	V
V_{IH}	High-level input voltage	2.0	V_{CCINT} +0.5	V
V_O	Output voltage	0	V_{CCINT} + 0.5	V

Note: 1. Numbers in parenthesis are for industrial-temperature range versions.

Endurance Characteristics

Symbol	Parameter	Min	Max	Units
t_{DR}	Data Retention	20	-	Years
N_{PE}	Program/Erase Cycles	10,000	-	Cycles

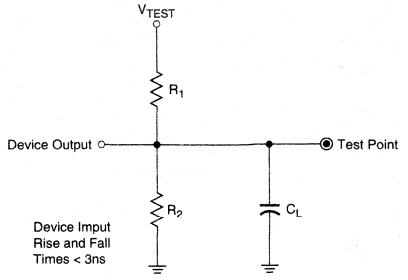
DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	Output high voltage for 5 V operation	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	Output high voltage for 3.3 V operation	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	Output low voltage for 5 V operation	I_{OL} = 24 mA V_{CC} = Min		0.5	V
	Output low voltage for 3.3 V operation	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
I_{IH}	I/O high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CC}		±10.0	µA
C_{IN}	I/O capacitance	V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}	Operating Supply Current (low power mode, active)	V_I = GND, No load f = 1.0 MHz	260 (Typ)		ma

AC Characteristics

Symbol	Parameter	XC95288-10		XC95288-15		XC95288-20		Units
		Min	Max	Min	Max	Min	Max	
t_{PD}	I/O to output valid		10.0		15.0		20.0	ns
t_{SU}	I/O setup time before GCK	6.5		8.0		10.0		ns
t_H	I/O hold time after GCK	0.0		0.0		0.0		ns
t_{CO}	GCK to output valid		6.5		8.0		10.0	ns
f_{CNT}[1]	16-bit counter frequency	111		95		83		MHz
f_{SYSTEM}[2]	Multiple FB internal operating frequency	67		56		50		MHz
t_{PSU}	I/O setup time before p-term clock input	2.5		4.0		4.0		ns
t_{PH}	I/O hold time after p-term clock input	4.0		4.0		6.0		ns
t_{PCO}	P-term clock to output valid		10.5		12.0		16.0	ns
t_{OE}	GTS to output valid		10.0		15.0		20.0	ns
t_{OD}	GTS to output disable		10.0		15.0		20.0	ns
t_{POE}	Product term OE to output enabled		15.5		18.0		22.0	ns
t_{POD}	Product term OE to output disabled		15.5		18.0		22.0	ns
t_{WLH}	GCK pulse width (High or Low)		4.5		5.5		5.5	ns
		Advanced						

Note: 1. f_{CNT} is the fastest 16-bit counter frequency available, using the local feedback when applicable. f_{CNT} is also the Export Control Maximum flip-flop toggle rate, f_{TOG}.
2. f_{SYSTEM} is the internal operating frequency for general purpose system designs spanning multiple FBs.



V_{CCIO} Level	V_{TEST}	R_1	R_1	C_L
5 V	5.0 V	160 Ω	120 Ω	35 pF
3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X5222

Internal Timing Parameters

Symbol	Parameter	XC95288-10		XC95288-15		XC95288-20		Units
		Min	Max	Min	Max	Min	Max	
Buffer Delays								
t_{IN}	Input buffer delay		3.5		4.5		6.5	ns
t_{GCK}	GCK buffer delay		3.0		3.0		3.0	ns
t_{GSR}	GSR buffer delay		6.0		7.5		9.5	ns
t_{GTS}	GTS buffer delay		10.0		15.0		20.0	ns
t_{OUT}	Output buffer delay		3.0		4.5		6.5	ns
t_{EN}	Output buffer enable/disable delay		0.0		0.0		0.0	ns
Product Term Control Delays								
t_{PTCK}	Product term clock delay		3.5		2.5		2.5	ns
t_{PTSR}	Product term set/reset delay		2.5		3.0		3.0	ns
t_{PTTS}	Product term 3-state delay		12.0		13.5		15.5	ns
Internal Register and Combinatorial delays								
t_{PDI}	Combinatorial logic propagation delay		1.0		3.0		4.0	ns
t_{SUI}	Register setup time	3.5		3.5		3.5		ns
t_{HI}	Register hold time	3.0		4.5		6.5		ns
t_{COI}	Register clock to output valid time		0.5		0.5		0.5	ns
t_{AOI}	Register async. S/R to output delay		7.0		8.0		9.0	ns
t_{RAI}	Register async. S/R recovery before clock	10.0		15.0		20.0		ns
t_{LOGI}	Internal logic delay		2.5		3.0		3.0	ns
t_{LOGILP}	Internal low power logic delay		11.0		11.5		11.5	ns
Feedback Delays								
t_F	FastCONNECT matrix feedback delay		8.5		11.0		13.0	ns
t_{LF}	Function Block local feeback delay		2.5		3.5		5.0	ns
Time Adders								
t_{PTA}[3]	Incremental Product Term Allocator delay		1.0		1.5		1.5	ns
t_{SLEW}	Slew-rate limited delay		4.5		5.0		5.5	ns
		Advanced						

Note: 3. t_{PTA} is multiplied by the span of the function as defined in the family data sheet.

XC95288 I/O Pins

Function Block	Macrocell	HQ208	BScan Order	Notes
1	1	–	861	
1	2	28	858	
1	3	29	855	
1	4	–	852	
1	5	30	849	
1	6	31	846	
1	7	–	843	
1	8	32	840	
1	9	–	837	
1	10	33	834	
1	11	–	831	
1	12	34	828	
1	13	–	825	
1	14	35	822	
1	15	36	819	
1	16	–	816	
1	17	37	813	
1	18	–	810	
2	1	–	807	
2	2	15	804	
2	3	16	801	
2	4	–	798	
2	5	17	795	
2	6	18	792	
2	7	–	789	
2	8	19	786	
2	9	–	783	
2	10	20	780	
2	11	–	777	
2	12	21	774	
2	13	–	771	
2	14	22	768	
2	15	23	765	
2	16	–	762	
2	17	25	759	
2	18	–	756	

Function Block	Macrocell	HQ208	BScan Order	Notes
3	1	–	753	
3	2	38	750	
3	3	39	747	
3	4	–	744	
3	5	40	741	
3	6	41	738	
3	7	–	735	
3	8	43	732	
3	9	–	729	
3	10	44	726	[1]
3	11	–	723	
3	12	45	720	
3	13	–	717	
3	14	46	714	[1]
3	15	47	711	
3	16	–	708	
3	17	48	705	
3	18	–	702	
4	1	–	699	
4	2	3	696	[1]
4	3	4	693	
4	4	–	690	
4	5	5	687	[1]
4	6	6	684	
4	7	–	681	
4	8	7	678	[1]
4	9	–	675	
4	10	8	672	
4	11	–	669	
4	12	9	666	[1]
4	13	–	663	
4	14	10	660	
4	15	12	657	
4	16	–	654	
4	17	14	651	
4	18	–	648	

Notes: [1] Global control pin
Macrocell outputs to package pins subject to change, contact factory for latest information. Power, GND, JTAG and Global Signals are fixed.

XC95288 I/O Pins (continued)

Function Block	Macrocell	HQ208	BScan Order	Notes
5	1	–	645	
5	2	49	642	
5	3	50	639	
5	4	–	636	
5	5	51	633	
5	6	54	630	
5	7	–	627	
5	8	55	624	[1]
5	9	–	621	
5	10	56	618	
5	11	–	615	
5	12	57	612	
5	13	–	609	
5	14	58	606	
5	15	60	603	
5	16	–	600	
5	17	61	597	
5	18	–	594	
6	1	–	591	
6	2	197	588	
6	3	198	585	
6	4	–	582	
6	5	199	579	
6	6	200	576	
6	7	–	573	
6	8	201	570	
6	9	–	567	
6	10	202	564	
6	11	–	561	
6	12	203	558	
6	13	–	555	
6	14	205	552	
6	15	206	549	[1]
6	16	–	546	
6	17	208	543	
6	18	–	540	

Function Block	Macrocell	HQ208	BScan Order	Notes
7	1	–	537	
7	2	62	534	
7	3	63	531	
7	4	–	528	
7	5	64	525	
7	6	66	522	
7	7	–	519	
7	8	67	516	
7	9	–	513	
7	10	69	510	
7	11	–	507	
7	12	70	504	
7	13	–	501	
7	14	71	498	
7	15	72	495	
7	16	–	492	
7	17	73	489	
7	18	–	486	
8	1	–	483	
8	2	186	480	
8	3	187	477	
8	4	–	474	
8	5	188	471	
8	6	189	468	
8	7	–	465	
8	8	191	462	
8	9	–	459	
8	10	192	456	
8	11	–	453	
8	12	193	450	
8	13	–	447	
8	14	194	444	
8	15	195	441	
8	16	–	438	
8	17	196	435	
8	18	–	432	

Note: [1] Global control pin

XC95288 I/O Pins (continued)

Function Block	Macrocell	HQ208	BScan Order	Notes
9	1	–	429	
9	2	74	426	
9	3	75	423	
9	4	–	420	
9	5	76	417	
9	6	77	414	
9	7	–	411	
9	8	78	408	
9	9	–	405	
9	10	80	402	
9	11	82	399	
9	12	83	396	
9	13	–	393	
9	14	84	390	
9	15	85	387	
9	16	–	384	
9	17	86	381	
9	18	–	378	
10	1	–	375	
10	2	170	372	
10	3	171	369	
10	4	–	366	
10	5	173	363	
10	6	174	360	
10	7	–	357	
10	8	175	354	
10	9	–	351	
10	10	178	348	
10	11	179	345	
10	12	180	342	
10	13	–	339	
10	14	182	336	
10	15	183	333	
10	16	–	330	
10	17	185	327	
10	18	–	324	

Function Block	Macrocell	HQ208	BScan Order	Notes
11	1	–	321	
11	2	87	318	
11	3	88	315	
11	4	–	312	
11	5	89	309	
11	6	90	306	
11	7	–	303	
11	8	91	300	
11	9	–	297	
11	10	95	294	
11	11	97	291	
11	12	99	288	
11	13	–	285	
11	14	100	282	
11	15	101	279	
11	16	–	276	
11	17	102	273	
11	18	–	270	
12	1	–	267	
12	2	158	264	
12	3	159	261	
12	4	–	258	
12	5	160	255	
12	6	161	252	
12	7	–	249	
12	8	162	246	
12	9	–	243	
12	10	164	240	
12	11	165	237	
12	12	166	234	
12	13	–	231	
12	14	167	228	
12	15	168	225	
12	16	–	222	
12	17	169	219	
12	18	–	216	

XC95288 I/O Pins (continued)

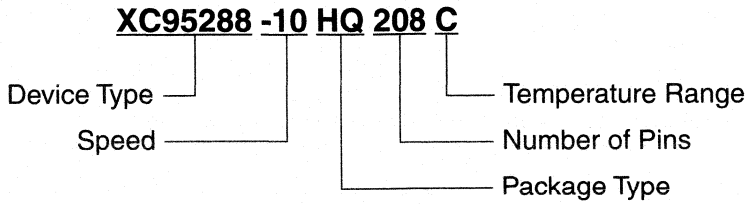
Function Block	Macrocell	HQ208	BScan Order	Notes
13	1	–	213	
13	2	103	210	
13	3	106	207	
13	4	–	204	
13	5	107	201	
13	6	109	198	
13	7	–	195	
13	8	110	192	
13	9	–	189	
13	10	111	186	
13	11	112	183	
13	12	113	180	
13	13	–	177	
13	14	114	174	
13	15	115	171	
13	16	–	168	
13	17	116	165	
13	18	–	162	
14	1	–	159	
14	2	144	156	
14	3	145	153	
14	4	–	150	
14	5	146	147	
14	6	147	144	
14	7	–	141	
14	8	148	138	
14	9	–	135	
14	10	149	132	
14	11	150	129	
14	12	151	126	
14	13	–	123	
14	14	152	120	
14	15	154	117	
14	16	–	114	
14	17	155	111	
14	18	–	108	

Function Block	Macrocell	HQ208	BScan Order	Notes
15	1	–	105	
15	2	117	102	
15	3	118	99	
15	4	–	96	
15	5	119	93	
15	6	120	90	
15	7	–	87	
15	8	121	84	
15	9	–	81	
15	10	122	78	
15	11	123	75	
15	12	125	72	
15	13	–	69	
15	14	126	66	
15	15	127	63	
15	16	–	60	
15	17	128	57	
15	18	–	54	
16	1	–	51	
16	2	131	48	
16	3	133	45	
16	4	–	42	
16	5	134	39	
16	6	135	36	
16	7	–	33	
16	8	136	30	
16	9	–	27	
16	10	137	24	
16	11	138	21	
16	12	139	18	
16	13	–	15	
16	14	140	12	
16	15	142	9	
16	16	–	6	
16	17	143	3	
16	18	–	0	

XC95288 Global, JTAG and Power Pins

Pin Type	HQ208
I/O/GCK1	44
I/O/GCK2	46
I/O/GCK3	55
I/O/GTS1	7
I/O/GTS2	9
I/O/GTS3	3
I/O/GTS4	5
I/O/GSR	206
TCK	98
TDI	94
TDO	176
TMS	96
V_{CCINT} 5 V	11,59,124,153,204
V_{CCIO} 3.3 V/5 V	1,26,53,65,79,92,105, 132,157,172,181,184
GND	2,13,24,27,42,52,68,81, 93,104,108,129,130, 141,156,163,177, 190,207
No Connects	–

Ordering Information



Speed Options

- 20 20 ns pin-to-pin delay
-15 15 ns pin-to-pin delay
-10 10 ns pin-to-pin delay

Packaging Options

HQ208 208-Pin Heat Sink Quad Flat Pack (HQFP)
BG352 352-Pin Plastic Ball Grid Array (BGA)

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	–40°C to 85°C

3

Component Availability

Pins		208	352
Type		Plastic HQFP	Plastic BGA
Code		HQ	BG
XC95288	–20	C(I)	C(I)
	–15	C	C
	–10	C	C

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

XC95432 In-System Programmable CPLD

January, 1997 (Version 1.0)

Advanced Product Specification

Features

- 10 ns pin-to-pin logic delays on all pins
- f_{CNT} to 111 MHz
- 432 macrocells with 9,600 usable gates
- Up to 232 user I/O pins
- 5 V in-system programmable
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- Advanced 0.6 µm CMOS 5V FastFLASH technology
- Supports parallel programming of more than one XC9500 concurrently
- Available in a 432-pin BGA package

Description

The XC95432 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration. It is comprised of twenty-four 36V18 Function Blocks, providing 9,600 usable gates with propagation delays of 10 ns.

XC95576 In-System Programmable CPLD

January, 1997 (Version 1.0)

Advanced Product Specification

Features

- 12 ns pin-to-pin logic delays on all pins
- f_{CNT} to 100 MHz
- 576 macrocells with 12,800 usable gates
- Up to 232 user I/O pins
- 5 V in-system programmable
  - Endurance of 10,000 program/erase cycles
  - Program/erase over full voltage and temperature range
- Enhanced pin-locking architecture
- Flexible 36V18 Function Block
  - 90 product terms drive any or all of 18 macrocells within Function Block
  - Global and product term clocks, output enables, set and reset signals
- Extensive IEEE Std 1149.1 boundary-scan (JTAG) support
- Programmable power reduction mode in each macrocell
- Slew rate control on individual outputs
- User programmable ground pin capability
- Extended pattern security features for design protection
- High-drive 24 mA outputs with 3.3 V or 5 V I/O capability
- Advanced 0.6 µm CMOS 5V FastFLASH technology
- Supports parallel programming of more than one XC9500 concurrently
- Available in a 432-pin BGA package

Description

The XC95576 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration. It is comprised of thirty-two 36V18 Function Blocks, providing 12,800 usable gates with propagation delays of 12 ns.

XILINX®

XC7300 Data Sheets

1 ISP and JTAG Support

2 Application Notes

3 XC9500 Data Sheets

4 XC7300 Data Sheets

5 Device Packaging

6 Quality Assurance

7 Technical Support

8 Sales Offices, Representatives, Distributors

Visit us on the WEB at:

www.xilinx.com

for the latest data sheets.

XILINX®

XC7300 CMOS CPLD Family

January, 1997 (Version 1.0)

Product Specification

Features

- High-performance Complex Programmable Logic Devices (CPLDs)
  - 5.0 / 7.5 ns pin-to-pin speeds on all fast inputs
  - Up to 167 MHz maximum clock frequency
- 100% PCI compliant
- High-drive 24 mA output
- I/O operation at 3.3 V or 5 V
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- 100% interconnect matrix
  - Maximizes resource utilization
  - Wire-AND capability via SMARTswitch
- High-speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 43 to 61 MHz 18-bit accumulators
- Multiple independent clocks
- Each input programmable as direct, latched, or registered
- Power management options
- Multiple security bits for design protection
- Supported by industry standard design and verification tools
- Advanced Dual-Block architecture
  - Fast Function Blocks
  - High-Density Function Blocks (XC7354, XC7372, XC73108, XC73144)
- 0.8 µ CMOS EPROM technology

Description

The XC7300 family employs a unique Dual-Block architecture that provides high speed operations via Fast Function Blocks and/or high density capability via High Density Function Blocks.

Fast Function Blocks (FFBs) provide fast, pin-to-pin speed and logic throughput for critical decoding and ultra-fast state machine applications. High-Density Function Blocks (FBs) provide maximum logic density and system-level features to implement complex functions with predictable timing for adders and accumulators, wide functions and state machines requiring large numbers of product terms, and other forms of complex logic. See Figure 1.

In addition, the XC7300 architecture employs the Universal Interconnect Matrix (UIM) which guarantees 100% interconnect of all internal functions. This interconnect scheme provides constant, short interconnect delays for all routing paths through the UIM. Constant interconnect delays simplify device timing and guarantee design performance, regardless of logic placement within the chip.

The UIM provides an intrinsic wire-AND capability called SMARTswitch. Transferring functions into the UIM conserves macrocell logic. This increases the total logic capacity of the device. The wire-AND capability also significantly increases the signal fan-in of each function block. All Xilinx-supported CPLD design tools automatically implement SMARTswitch.

The XC7300 Family

	XC7318	XC7336	XC7354	XC7372	XC73108	XC73144
Typical 22V10 Equivalent	1.5 – 2	3 – 4	6	8	12	16
Number of Macrocells	18	36	54	72	108	144
Number of Function Blocks	2	4	6	8	12	16
Number of Flip-Flops	18	36	108	126	198	276
Number of Fast Inputs	12	12	12	12	12	12
Number of Signal Pins	38	38	58	84	120	156

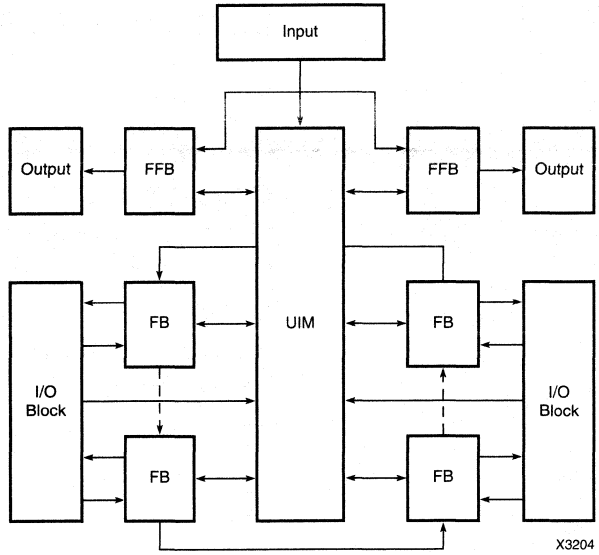


Figure 1: XC7300 Device Block Diagram

All XC7300 Dual-Block CPLDs include programmable power management features to specify high-performance or low-power operation on an individual macrocell-by-macrocell basis. Unused macrocells are automatically turned off to minimize power dissipation. Designers can operate speed-critical paths at maximum performance, while non-critical paths dissipate less power.

Fast Function Blocks

The FFB has 24 inputs that can be individually selected from the UIM, 12 fast input pins, or the nine macrocell feedbacks from the FFB. The programmable AND array in each FFB generates 45 product terms to drive the nine macrocells in each FFB. Each macrocell can be configured for registered or combinatorial logic. See Figure 2.

Five product terms from the programmable AND array are allocated to each macrocell. Four of these product terms are ORed together and may be optionally inverted before driving the input of a programmable D-type flip-flop. The fifth product term drives the asynchronous active-High programmable Reset or Set Input to the macrocell flip-flop. The flip-flop can be configured as a D-type or Toggle flip-flop, or transparent for combinatorial outputs.

Two FFB macrocell differences exist between the XC7318/XC7336/XC73144 and the XC7354/XC7372/XC73108.

In the XC7318, XC7336 and XC73144, five product terms from the programmable AND array are allocated to each macrocell. Four of these product terms are OR'd together and may be optionally inverted before driving the input of a programmable D-type flip-flop. The fifth product term drives the asynchronous active High programmable Set or Reset input to the macrocell flip-flop. The flip-flop can be configured as a D-type or Toggle flip-flop, or transparent for combinatorial outputs. See Figure 2.

In the XC7354, XC7372 and XC73108, five product terms from the programmable AND array are allocated to each macrocell. Four of these product terms are OR'd together, inverted and drive the input of a programmable D-type flip-flop. The fifth product term drives the asynchronous active High programmable Set input to the macrocell flip-flop. The flip-flop can be configured as a D-type flip-flop or transparent for combinatorial outputs. See Figure 3.

The programmable clock source is one of two global Fast-Clock signals (FCLK0 or FCLK1) that are distributed with short delay and minimal skew over the entire chip.

The FFB macrocells drive chip outputs directly through 3-state output buffers. Each output buffer can be individually controlled by one of two dedicated Fast Output Enable inputs or permanently enabled or disabled. The macrocell output can also be routed back as an input to the FFB and the UIM.

Each FFB output is capable of sinking 24 mA when V_{CCIO} = 5 volts. These include all outputs on the XC7318 and XC7336 devices and all Fast Outputs (FOs) on the XC7354, XC7372, XC73108, and XC73144 devices.

Unlike other I/Os, the FFB inputs do not have an input register.

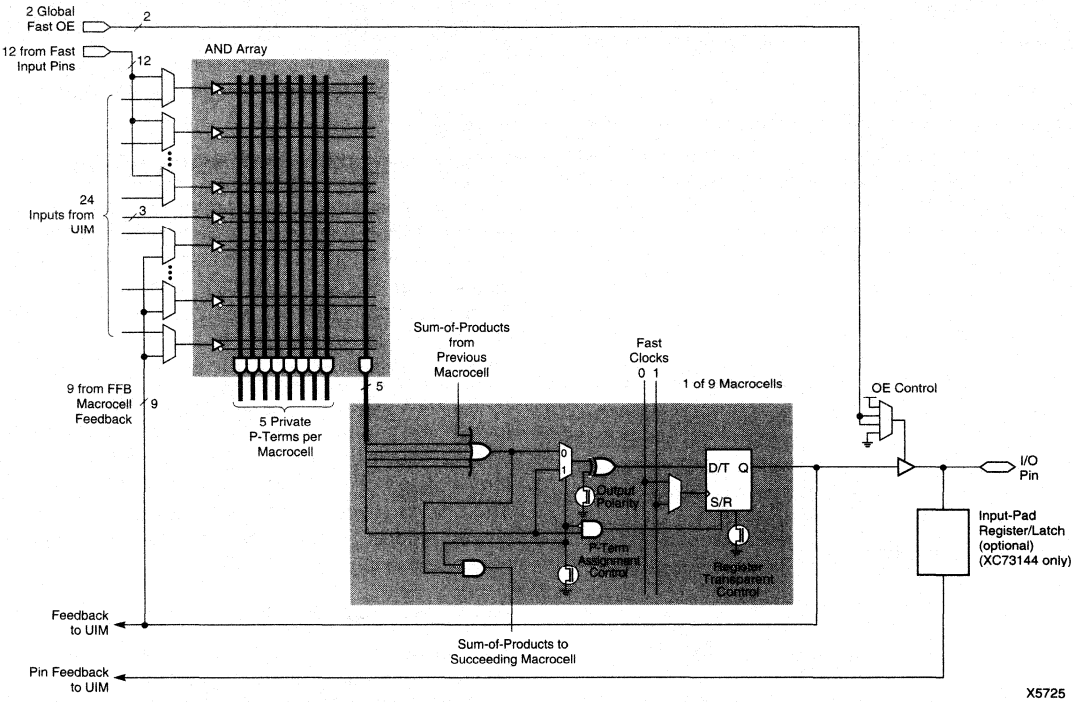


Figure 2: Fast Function Block and Macrocell Schematic for the XC7318, XC7336, and XC73144

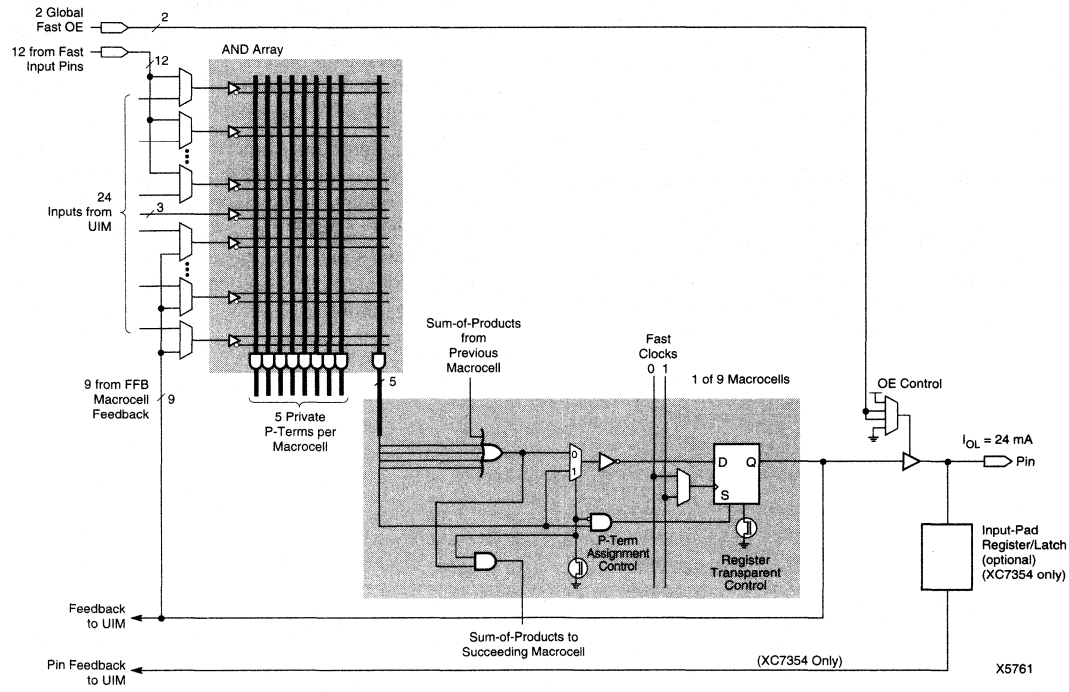


Figure 3: Fast Function Block and Macrocell Schematic for the XC7354, XC7372, and XC73108

Product Term Assignment

Each macrocell sum-of-product OR gates can be expanded using the FFB product term assignment scheme. Product term assignment transfers product terms in increments of four product terms from one macrocell to the neighboring macrocell (Figure 4). Complex logic functions requiring up to 36 product terms can be implemented using all nine macrocells within the FFB. When product terms are assigned to adjacent macrocells, the product term normally dedicated to the Set or Reset function becomes the input to the macrocell register.

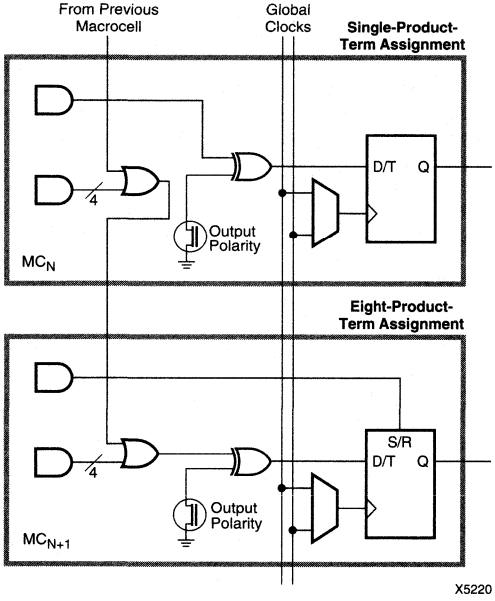


Figure 4: Fast Function Block Product Term Assignment

High-Density Function Blocks

The XC7354, XC7372, XC73108 and XC73144 devices contain multiple, High-Density FBs linked though the UIM. Each FB contains nine macrocells. Each macrocell can be configured for either registered or combinatorial logic. A detailed block diagram of the FB is shown in Figure 5.

Each FB receives 21 signals and their complements from the UIM and an additional three inputs from the Fast Input (FI) pins.

Shared and Private Product Terms

Each macrocell contains five private product terms that can be used as the primary inputs for combinatorial functions implemented in the Arithmetic Logic Unit (ALU), or as individual Reset, Set, Output-Enable, and Clock logic functions for the flip-flop. Each FB also provides an additional 12 shared product terms, which are uncommitted product terms available for any of the nine macrocells within the FB.

Four private product terms can be ORed together with up to four shared product terms to drive the D1 input to the ALU. The D2 input is driven by the OR of the fifth private product term and up to eight of the remaining shared product terms. The shared product terms add no logic delay, and each shared product term can be connected to one or all nine macrocells in the FB.

Arithmetic Logic Unit

The functional versatility of each macrocell in the FB is enhanced through additional gating and control functions available in the ALU. A detailed block diagram of the XC7300 ALU is shown in Figure 6.

The ALU has two programmable modes; *logic* and *arithmetic*. In logic mode, the ALU functions as a 2-input function generator using a 4-bit look-up table that can be programmed to generate any Boolean function of its D1 and D2 inputs as illustrated in Table 1.

The function generator can OR its inputs, widening the OR function to a maximum of 17 inputs. It can AND them, which means that one sum-of-products can be used to mask the other. It can also XOR them, toggling the flip-flop or comparing the two sums of products. Either or both of the sum-of-product inputs to the ALU can be inverted, and either or both can be ignored.

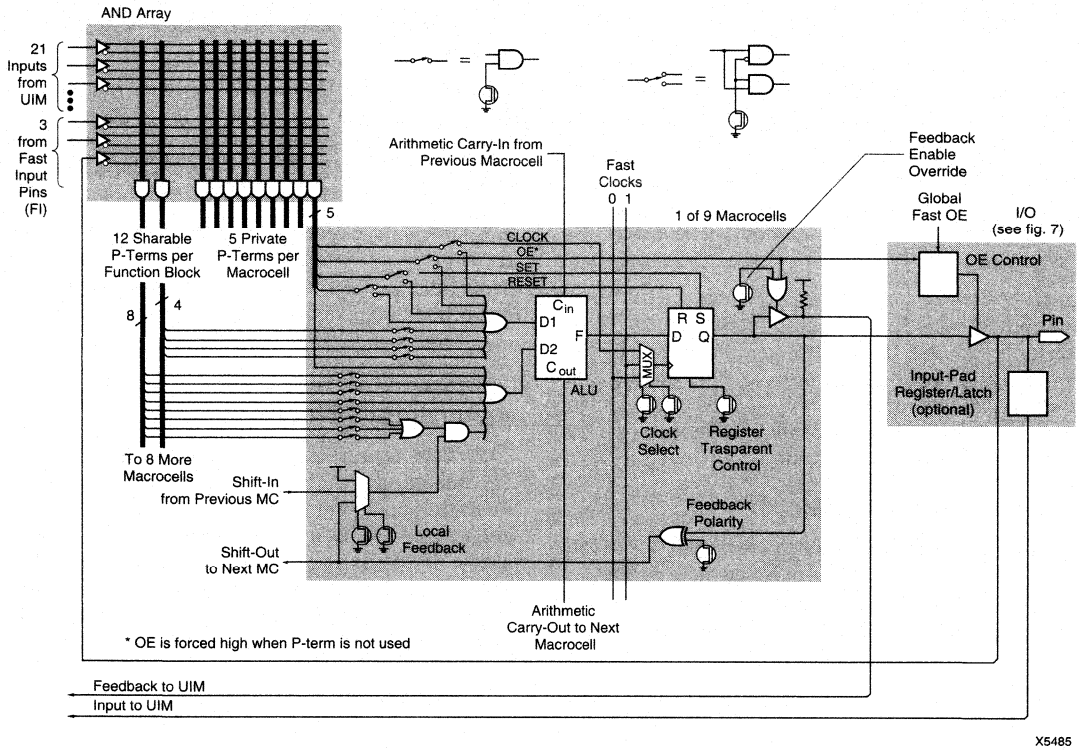


Figure 5: High-Density Function Block and Macrocell Schematic

Table 1: Function Generator Logic Operations

Function	
D1:+: D2	$\overline{D1}$:+: $\overline{D2}$
D1 * D2	$\overline{D1 * D2}$
D1 + D2	$\overline{D1 + D2}$
D1	D2
D1	D2
D1 * $\overline{D2}$	$\overline{D1}$ * D2
D1 + $\overline{D2}$	$\overline{D1}$ + D2

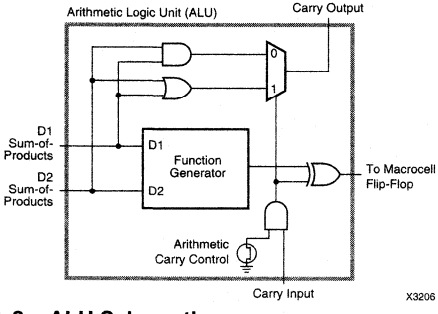


Figure 6: ALU Schematic

Therefore, the ALU can implement one additional layer of logic without any speed penalty.

In arithmetic mode, the ALU block can be programmed to generate the arithmetic sum or difference of the D1 and D2 inputs. Combined with the carry input from the next lower macrocell, the ALU operates as a 1-bit full adder generating a carry output to the next higher macrocell. The carry chain propagates between adjacent macrocells and also crosses the boundaries between FBs. This dedicated carry chain overcomes the inherent speed and density problems of the traditional CPLD architecture when trying to perform arithmetic functions.

Carry Lookahead

Each FB provides a carry lookahead generator capable of anticipating the carry across all nine macrocells. The carry lookahead generator reduces the ripple-carry delay of wide arithmetic functions such as add, subtract, and magnitude compare to that of the first nine bits, plus the carry lookahead delay of the higher-order FBs.

Macrocell Flip-Flop

The ALU block output drives the input of a programmable D-type flip-flop. The flip-flop is triggered by the rising edge of the clock input, but it can be configured as transparent,

making the Q output identical to the D input, independent of the clock, or as a conventional flip-flop.

The macrocell clock source is programmable and can be one of the private product terms or one of two global FastCLK signals (FCLK0 and FCLK1). Global FastCLK signals are distributed to every macrocell flip-flop with short delay and minimal skew.

The asynchronous Set and Reset product terms override the clocked operation. If both asynchronous inputs are active simultaneously, Reset overrides Set.

In addition to driving the chip output buffer, the macrocell output is routed back as an input to the UIM. One private product term can be configured to control the Output Enable of the output buffer and/or the feedback to the UIM. If it is configured to control UIM feedback, the Output Enable product term forces the UIM feedback line High when the macrocell output is disabled.

Universal Interconnect Matrix

The UIM receives inputs from each macrocell output, I/O pin, and dedicated input pin. Acting as fully connected crossbar switch, the UIM generates 21 output signals to each FB and 24 output signals to each FFB.

Each UIM input can be connected to any UIM output. The UIM delay is constant, regardless of the routing distance, fan-out, or fan-in.

When multiple UIM inputs are connected to the same output, their wire-AND is formed by using internally available inversions. This AND logic can also be used to implement wide NAND, OR or NOR functions. This offers an additional level of logic without any speed penalty.

A macrocell feedback signal that is disabled by the output enable product term represents a High input to the UIM. Programming several such macrocell outputs onto the same UIM output emulates a 3-state bus line. If one of the macrocell outputs is enabled, the UIM output assumes the enabled output's level.

Input/Output Blocks

Macrocells drive chip outputs directly through 3-state output buffers, each individually controlled by the Output Enable product term mentioned above. The macrocell output can be inverted. An additional configuration option allows the output to be disabled permanently. Two dedicated FastOE inputs can also be configured to control any of the chip outputs instead of, or in conjunction with, the individual Output Enable product term. See Figure 7.

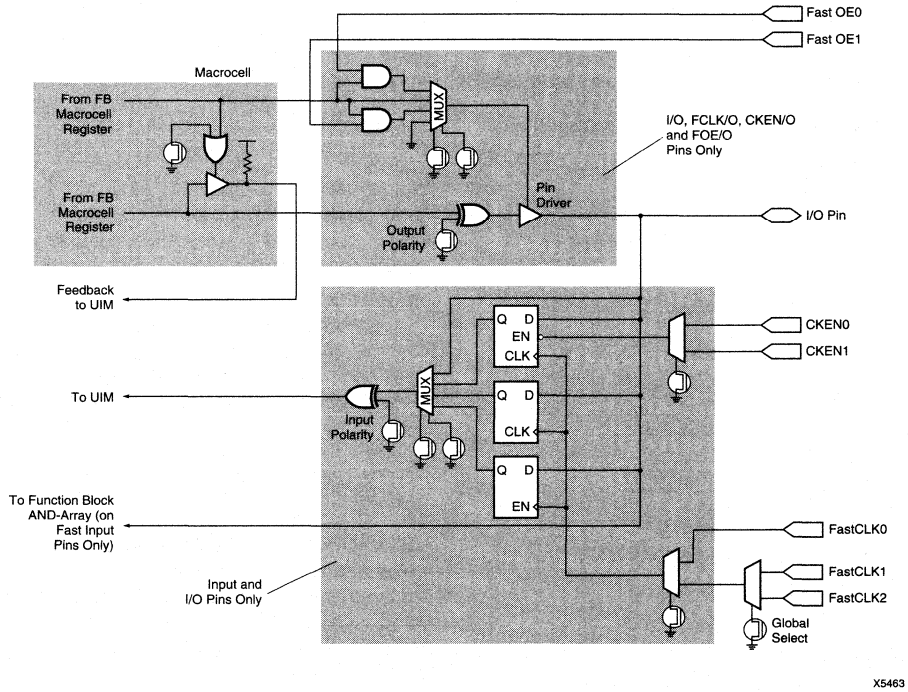


Figure 7: Input/Output Schematic (except XC7318/XC7336 which do not include I/O flip-flops)

Output buffers, except those connected to FFBs, can sink 12 mA when V_{CCIO} = 5 V. FFB outputs can sink 24 mA when V_{CCIO} = 5 V.

Each signal input to the chip is connected to a programmable input structure that can be configured as direct, latched, or registered. The latch and flip-flop can use one of two FastCLK signals as latch enable or clock. The two FastCLK signals are FCLK0 and a global choice of either FCLK1 or FCLK2. Latches are transparent when FastCLK is High, and flip-flops clock on the rising edge of FastCLK. The flip-flop includes an active-low clock enable, which when High, holds the present state of the flip-flop and inhibits response to the input signal. The clock enable source is one of two global Clock Enable signals ($\overline{CE0}$ and $\overline{CE1}$). An additional configuration option is polarity inversion for each input signal.

3.3 V or 5 V Interface Configuration

XC7300 devices can be used in systems with two different supply voltages: 3.3 V and 5 V. Each XC7300 device has separate V_{CC} connections to the internal logic and input buffers (V_{CCINT}) and to the I/O drivers (V_{CCIO}). V_{CCINT} must always be connected to a nominal 5 V supply, while V_{CCIO} may be connected to either 3.3 V or 5 V, depending on the output interface requirement.

When V_{CCIO} is connected to 5 V, the input thresholds are TTL levels, compatible with 3.3 V and 5 V logic. The output High levels are also TTL compatible. When V_{CCIO} is connected to 3.3 V, the input thresholds are still TTL levels, and the outputs pull up to the 3.3 V rail. This makes the XC7300 family ideal for interfacing directly to 3.3 V components. In addition, the output structure is designed so the I/O can also safely interface to a mixed 3.3 V and 5 V bus.

Power-On Characteristics/Master Reset

Each XC7300 device undergoes a short internal initialization sequence upon device powerup. During this time (t_{RESET}), the outputs remain 3-stated while the device is configured from its internal EPROM array and all registers are initialized. If the $\overline{MR}$ pin is tied to V_{CCINT}, the initialization sequence is completely transparent to the user and is completed in t_{RESET} after V_{CCINT} has reached 4.75 V. If $\overline{MR}$ is held low while the device is powering up, the internal initialization sequence begins and outputs will remain 3-stated until the sequence is complete and $\overline{MR}$ is brought High. V_{CC} rise must be monotonic to ensure the initialization sequence is performed correctly.

For additional flexibility, the $\overline{MR}$ pin is provided so the device can be reinitialized after power is applied. On the falling edge of $\overline{MR}$, all outputs become 3-stated and the initialization sequence begins. The outputs remain 3-stated until the internal initialization sequence is complete and $\overline{MR}$ is brought High. The minimum $\overline{MR}$ pulse with is t_{WMR}. If $\overline{MR}$ is brought high after t_{WMR}, but before t_{RESET}, the outputs will become active after t_{RESET}. It is essential that the $\overline{MR}$ pin remain static during power on reset (t_{RESET}).

During the initialization sequence, all input registers or latches are preloaded High and all FB and FFB macrocell registers are preloaded to a known state. For FFB macrocell registers where the Set/Reset product term is defined, the preload is accomplished by asserting the product term shortly before the end of the initialization sequence. When the Set/Reset product term is configured as Reset, the register preload value is Low. When the Set/Reset product term is configured as Set, the register preload value is High. For FFB macrocell registers where the Set/Reset product term is not used, the register preload value is High.

For FB macrocell registers, the preload value is defined by a separate preload configuration bit, independent of the Set and Reset product terms. The value of this preload configuration bit may be determined by the user. If unspecified, the register preload value is Low.

Power Management

The XC7300 family features a power-management scheme permitting non-speed-critical paths of a design to be operated at reduced power. Overall power dissipation is often reduced significantly, since, in most systems only a small portion is speed critical.

Macrocells can individually be specified for high performance or low power operation by adding attributes to the logic schematic, or declaration statements to the behavioral description. To further reduce power dissipation, unused FBs are turned off and unused macrocells in used FBs are configured for low power operation.

Erasure Characteristics

In windowed packages, the EPROM array can be erased by exposure to UV light with wavelengths of approximately 4000 Å. The recommended erasure time is approximately 1 hr. when the device is placed within 1 in. of an UV lamp with 12,000 $\mu W/cm^2$ power rating. To prevent unintentional exposure, place opaque labels over the device window.

When the device is exposed to high intensity UV light for much longer periods, permanent damage can occur. The maximum integrated dose the XC7300 CPLDs can be exposed to without damage is 7000 W • s/cm^2, or approximately one week at 12,000 $\mu W/cm^2$.

Design Recommendations

For proper operation, all unused input and I/O pins must be connected to a valid logic level (High or Low). The recommended decoupling for all V_{CC} pins should total 1 μF using high-speed (tantalum or ceramic) capacitors.

Use electrostatic discharge (ESD) handling procedures with the XC7300 CPLDs to prevent damage to the device during programming, assembly, and test.

Design Security

Each member of the XC7300 family has a multibit security system that controls access to the configuration programmed into the device. This security scheme uses multiple EPROM bits at various locations within the EPROM array to offer a higher degree of design security than other EPROM and fused-based devices. Programmed data within EPROM cells is invisible–even when examined under a microscope–and cannot be selectively erased. The EPROM security bits, and the device configuration data, reset when the device is erased.

High-Volume Production Programming

The XC7300 family is available as a factory programmed product. For factory programming procedures, contact your local Xilinx representative.

XACT*step* Development System

The XC7300 CPLD family is fully supported by the Xilinx XACT*step* development system. The designer can create the design using ABEL, schematics, equations, VHDL or other HDL languages in a variety of software front-end tools. The XACT*step* development system can be used to implement the design and generate a bitmap which can be used to program the XC7300 devices.

Timing Model

Timing within the XC7300 family is accurately determined using external timing parameters from the device data sheet, a variety of CAE simulators, or with the timing model shown in Figure 8.

The timing model is based on the fixed internal delays of the XC7300 architecture which consists of four basic parts: I/O Blocks, the UIM, FFBs and FBs. The timing model identifies the internal delay paths and their relationships to ac characteristics. Using this model and the ac characteristics, designers can calculate the timing information for a particular device.

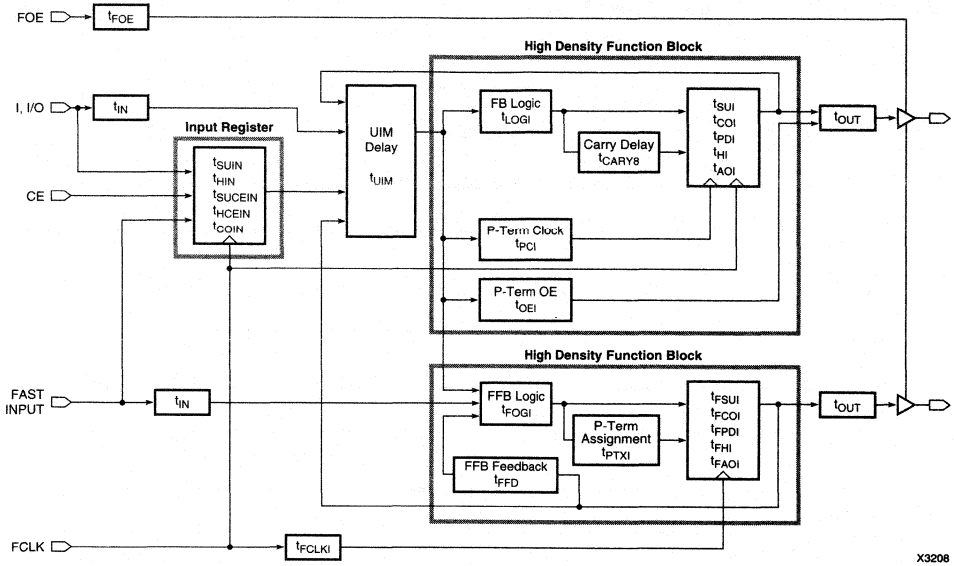
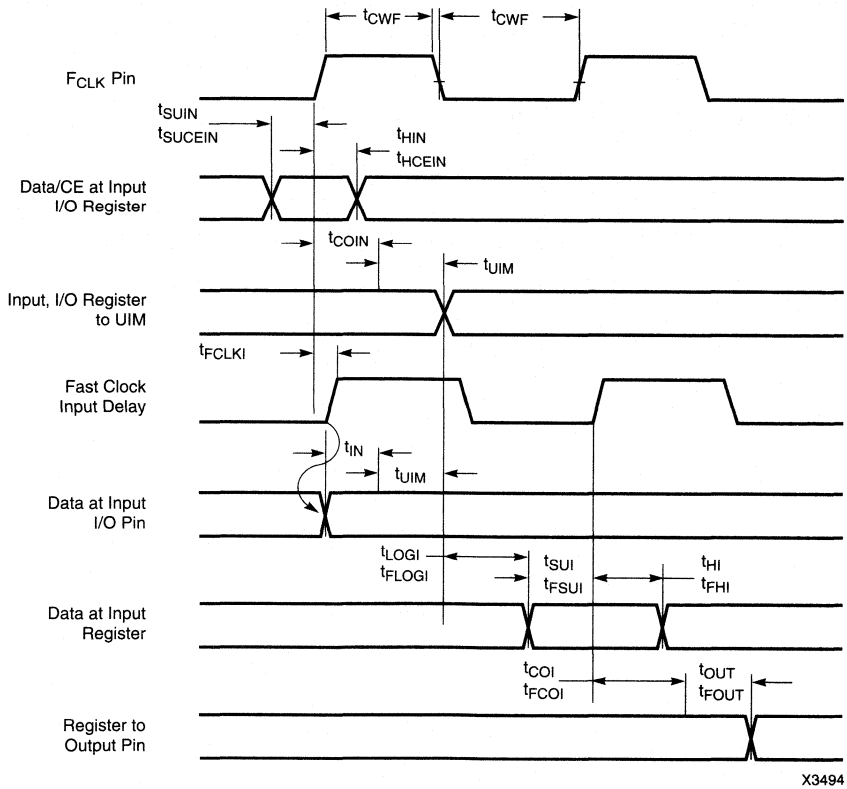


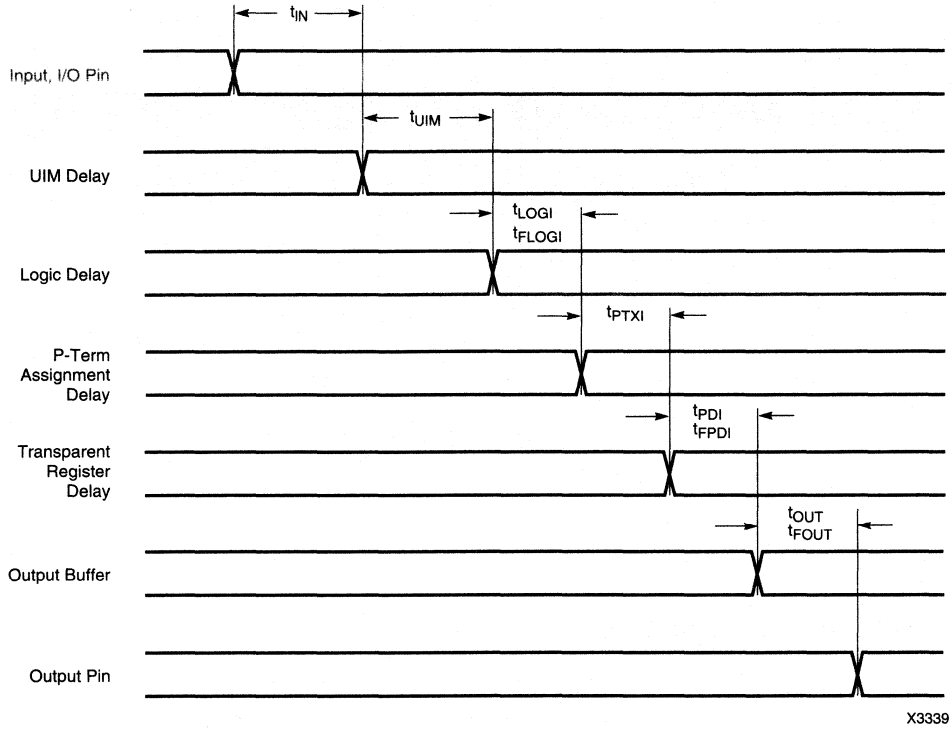
Figure 8: XC7300 Timing Model

4

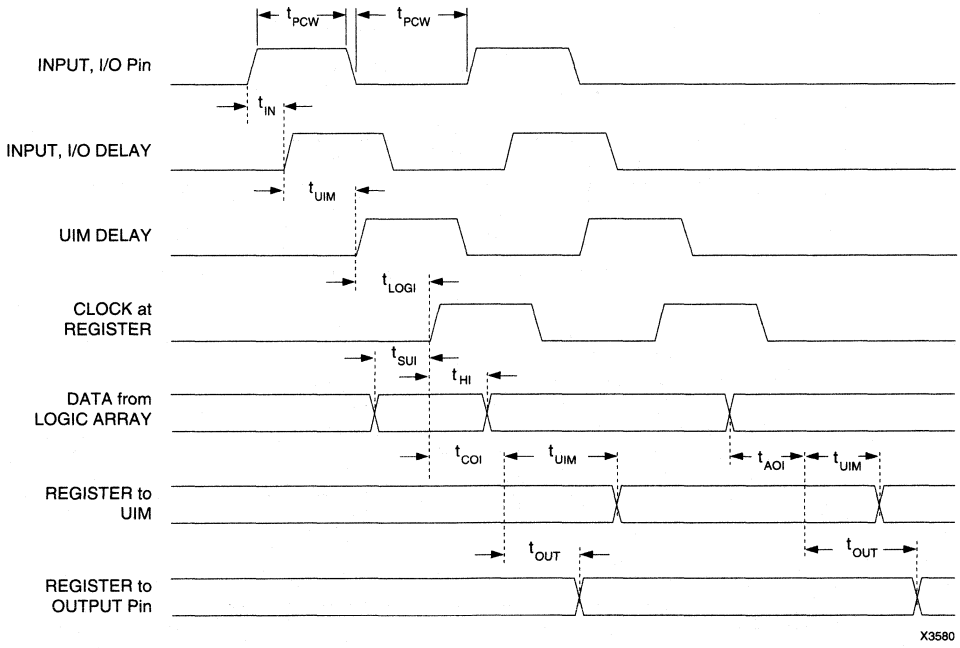
Synchronous Switching Characteristics



Combinational Switching Characteristics



Asynchronous Switching Characteristics



XC7318
18-Macrocell CMOS CPLD

January, 1997 (Version 1.0) | Product Specification

Features

- Ultra high-performance Complex Programmable Logic Devices (CPLDs)
  - 5 ns pin-to-pin speeds on all fast inputs
  - Up to 167 MHz maximum clock frequency
- 100% PCI compliant
- High-drive 24 mA output
- I/O operation at 3.3 V or 5 V
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- 100% interconnect matrix
  - Maximizes resource utilization
  - Wire-AND capability via SMARTswitch
- Multiple security bits for design protection
- Incorporates two PAL-like 24V9 Fast Function Blocks
- 0.8 µ CMOS EPROM technology
- Available in 44-pin PQFP and PLCC packages

General Description

The XC7318 is a high performance CPLD providing general purpose logic integration. It consists of two PAL-like 24V9 Fast Function Blocks interconnected by the 100%-populated Universal Interconnect Matrix (UIM™). See Figure 1 for the architecture overview.

4

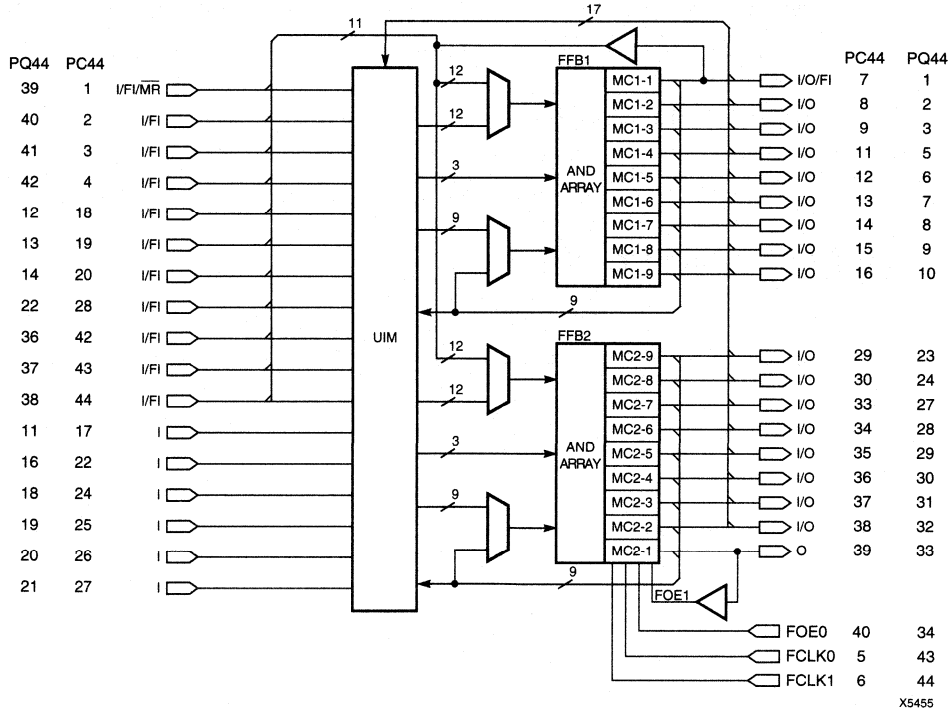


Figure 1: XC7318 Architecture

Power Estimation

Figure 2 shows a typical power estimation for the XC7318 device, programmed as a 16-bit counter and operating at the indicated clock frequency.

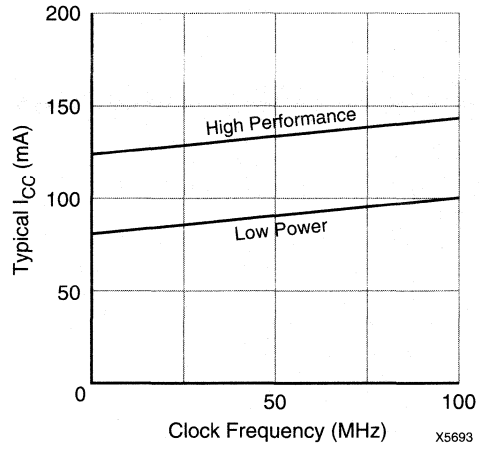


Figure 2: Typical I_{CC} vs. Frequency for XC7318

Absolute Maximum Ratings

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

Warning: 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.

Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units
V_{CCINT} V_{CCIO}	Supply voltage relative to GND Commercial T_A = 0°C to 70°C	4.75	5.25	V
V_{CCIO}	I/O supply voltage relative to GND	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{IH}	High-level input voltage	2.0	V_{CC} +0.5	V
V_O	Output voltage	0	V_{CCIO}	V
T_{IN}	Input signal transition time		50.0	ns

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V TTL Low-level output voltage	I_{OL} = 24 mA V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage	I_{OL} = 24 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins	V_{IN} = GND f = 1.0 MHz		6.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FOE0, FOE1)	V_{IN} = GND f = 1.0 MHz		8.0	pF
C_{OUT}[1]	Output capacitance	V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}[2]	Supply current	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5V f = 1.0 MHz @ 25°C	90 Typ		mA

Notes: 1. Sample tested.
2. Measured with device programmed as a 16-bit counter.

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time		80	160	µs

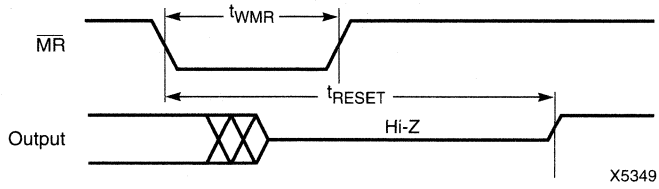


Figure 3: Global Reset Waveform

Fast Function Block (FFB) External AC Characteristics [1]

Symbol	Parameter	XC7318-5		XC7318-7		Units
		Min	Max	Min	Max	
t_{PD}	Fast input to output valid [2]		5.0		7.5	ns
	I/O or input to output valid [2]		8.5		12.0	ns
t_{SU}	Fast input setup time before FCLK	4.5		5.0		ns
	I/O or input setup time before FCLK	7.0		8.5		ns
t_H	Fast, I/O or input hold time after FCLK	0		0		ns
t_{CO}	FCLK input to output valid		4.5		4.5	ns
t_{FOE}	FOE input to output valid		7.0		7.5	ns
t_{FOD}	FOE input to output disable		7.0		7.5	ns
f_{MAX}	Max count frequency [2, 3]	167.0		125.0		MHz
t_{WLH}	Fast Clock pulse width (High or Low)	3.0		4.0		ns

Notes: 1. All appropriate ac specifications tested using Figure 5 as test load circuit.
2. Assumes four product terms per output.
3. Export Control Max. flip-flop toggle rate.

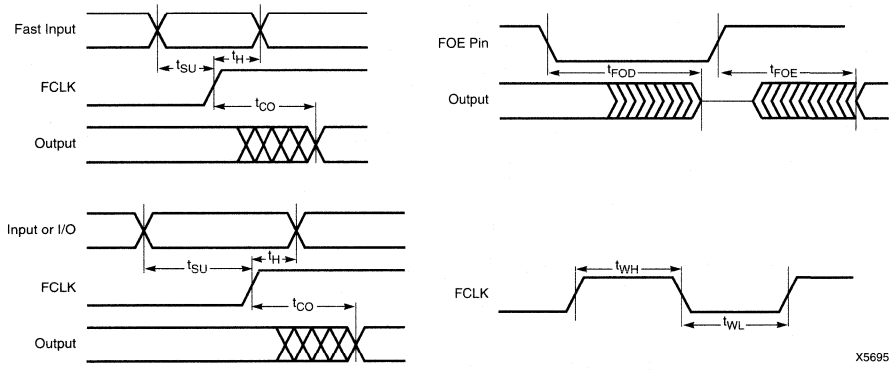
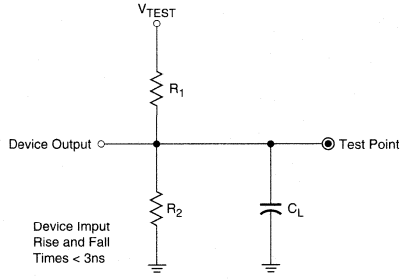


Figure 4: Switching Waveform



V_{CCIO} Level	V_{TEST}	R_1	R_1	C_L
5 V	5.0 V	160 Ω	120 Ω	35 pF
3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X5222

Figure 5: AC Load Circuit

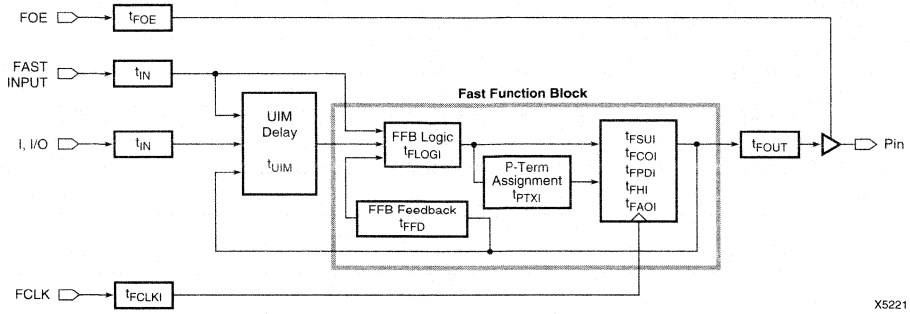


Figure 6: XC7318 Timing Model

Timing Model

Timing within the XC7318 is accurately determined using external timing parameters from the device data sheet, using a variety of CAE simulators, or with the timing model shown in Figure 6.

The timing model is based on the fixed internal delays of the XC7318 architecture that consists of three basic parts: I/O Blocks, the UIM and Fast Function Blocks. The timing model identifies the internal delay paths and their relationships to ac characteristics. Using this model and the ac characteristics, designers can easily calculate the timing information for the XC7318.

4

Fast Function Block (FFB) Internal AC Characteristics

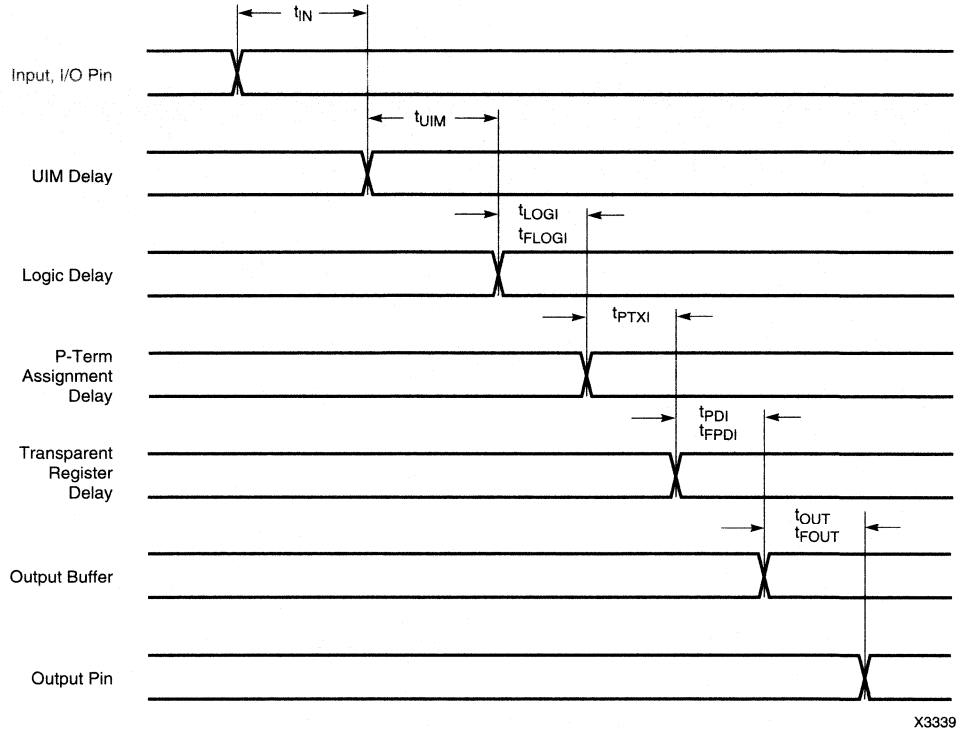
Symbol	Parameter	XC7318-5		XC7318-7		Units
		Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay [1]		1.0		1.5	ns
$t_{FLOGILP}$	Low-power FFB logic array delay [1]		2.0		3.5	ns
t_{FSUI}	FFB register setup time	2.5		1.5		ns
t_{FHI}	FFB register hold time	1.0		2.5		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		0.5	ns
t_{FAOI}	FFB register async. set delay		2.0		2.0	ns
t_{PTXI}	FFB p-term assignment delay		0.6		0.8	ns
t_{FFD}	FFB feedback delay		0.5		4.0	ns

Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.

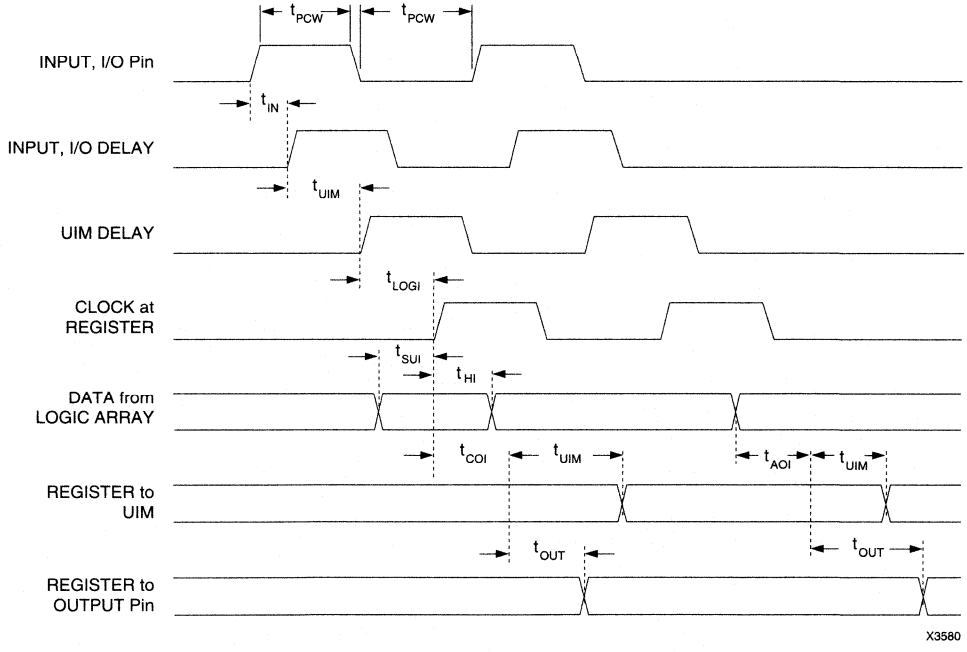
Internal AC Characteristics

Symbol	Parameter	XC7318-5		XC7318-7		Units
		Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		1.5		2.5	ns
t_{FOUT}	FFB output buffer and pad delay		2.0		3.0	ns
t_{UIM}	Universal Interconnect Matrix delay		3.5		4.5	ns
t_{FCLKI}	Fast clock buffer delay		1.5		1.5	ns

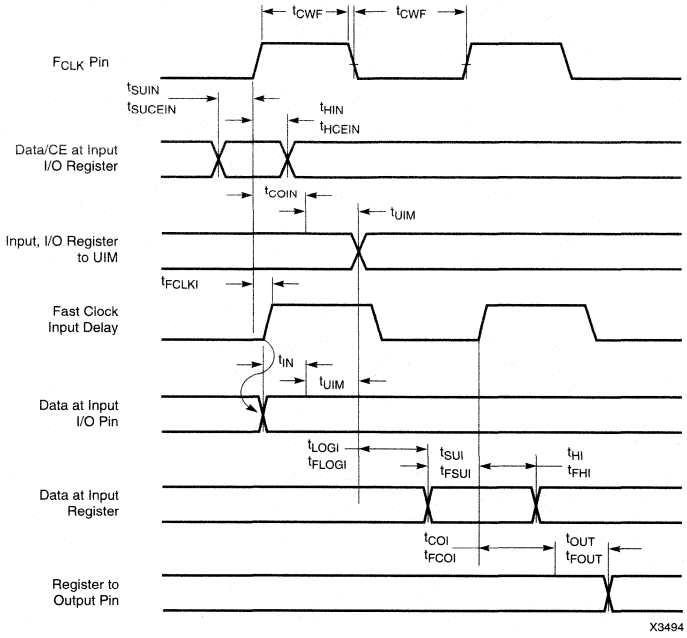
Combinational Switching Characteristics



Asynchronous Switching Characteristics



Synchronous Switching Characteristics

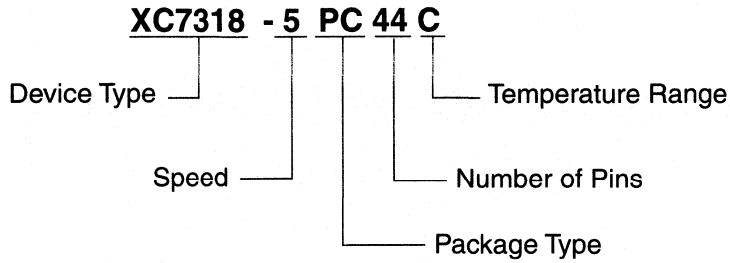


XC7318 Pinouts

PQ44	PC44	Input	XC7318	Output
39	1	I/FI	$\overline{MR}$	
40	2	I/FI		
41	3	I/FI		
42	4	I/FI		
43	5	FCLK0		
44	6	FCLK1		
1	7	I/FO/FI		MC1-1
2	8	I/FO		MC1-2
3	9	I/FO		MC1-3
4	10		**GND**	
5	11	I/FO		MC1-4
6	12	I/FO		MC1-5
7	13	I/FO		MC1-6
8	14	I/FO		MC1-7
9	15	I/FO		MC1-8
10	16	I/FO		MC1-9
11	17	I		
12	18	I/FI		
13	19	I/FI		
14	20	I/FI		
15	21		V_{CCINT}	
16	22	I		

PQ44	PC44	Input	XC7318	Output
17	23		**GND**	
18	24	I		
19	25	I		
20	26	I		
21	27	I		
22	28	I/FI		
23	29	I/FO		MC2-9
24	30	I/FO		MC2-8
25	31		**GND**	
26	32		V_{CCIO}	
27	33	I/FO		MC2-7
28	34	I/FO		MC2-6
29	35	I/FO		MC2-5
30	36	I/FO		MC2-4
31	37	I/FO		MC2-3
32	38	I/FO		MC2-2
33	39	FOE1/FO		MC2-1
34	40	FOE0		
35	41		V_{CCINT}/V_{PP}	
36	42	I/FI		
37	43	I/FI		
38	44	I/FI		

Ordering Information



Speed Options

-7	7.5 ns pin-to-pin delay (commercial only)
-5	5 ns pin-to-pin delay (commercial only)

Packaging Options

PC44	44-Pin Plastic Leaded Chip Carrier
PQ44	44-Pin Plastic Quad Flat Pack

Temperature Options

C	Commercial	$0^{o}C$ to $70^{o}C$

Component Availability

Pins		44	
Type		Plastic PLCC	Plastic PQFP
Code		PC44	PQ44
XC7318	–7	C	C
	–5	C	C

C = Commercial = 0° to +70°C

XC7336/XC7336Q
36-Macrocell CMOS CPLD

January, 1997 (Version 1.0) | Product Specification

Features

- Ultra high-performance Complex Programmable Logic Devices (CPLDs)
  - 5 ns pin-to-pin speeds on all fast inputs
  - Up to 167 MHz maximum clock frequency
- New low power XC7336Q
- 100% PCI compliant
- High-drive 24 mA output
- I/O operation at 3.3 V or 5 V
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- 100% interconnect matrix
  - Maximizes resource utilization
  - Wire-AND capability via SMARTswitch
- Multiple security bits for design protection
- Incorporates four PAL-like 24V9 Fast Function Blocks
- 0.8 µ CMOS EPROM technology
- Available in 44-pin PQFP and PLCC/CLCC packages (The 7336Q is also available in 44-pin VQFP package)

General Description

The XC7336 is a high performance CPLD providing general purpose logic integration. It consists of four PAL-like 24V9 Fast Function Blocks interconnected by the 100%-populated Universal Interconnect Matrix (UIM™). See Figure 1 for the architecture overview.

The XC7336 is designed in 0.8 µ CMOS EPROM technology, in speed grades ranging from 5 to 15 ns. The XC7336Q is also available now, providing lower power consumption in -10, -12 and -15 ns speed grades.

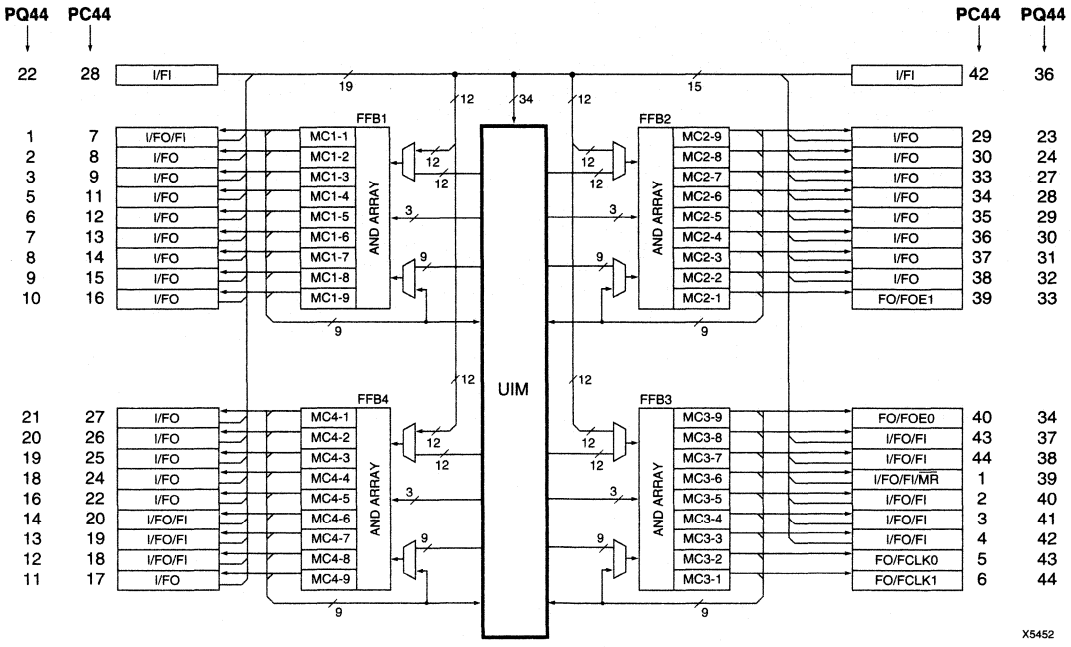


Figure 1: XC7336 Architecture

Power Estimation

Figure 2 shows a typical power estimation for the XC7336 and the XC7336Q device, programmed as two 16-bit counters and operating at the indicated clock frequency.

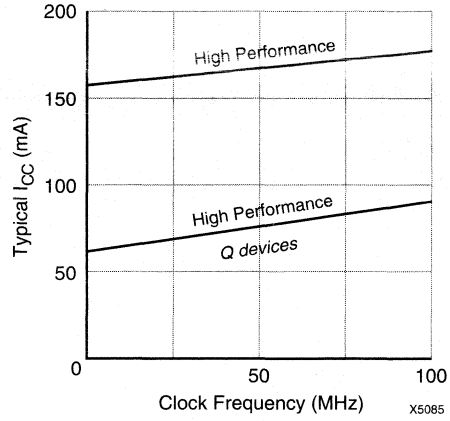


Figure 2: Typical I_{CC} vs. Frequency for XC7336

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage with respect to GND	-0.5 to 7.0	V
V_{IN}	DC Input voltage with respect to GND	-0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Maximum soldering temperature (10s @ 1/16 in. = 1.5 mm)	+250	°C

Warning: 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.

Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units
V_{CCINT}	Supply voltage relative to GND Commercial T_A = 0°C to 70°C	4.75	5.25	V
V_{CCIO}	Supply voltage relative to GND Industrial T_A = –40°C to 85°C	4.50	5.50	V
V_{CCIO}	I/O supply voltage relative to GND	3.0	3.60	V
V_{IL}	Low-level input voltage	0	0.80	V
V_{IH}	High-level input voltage	2.0	V_{CC} +0.5	V
V_O	Output voltage	0	V_{CCIO}	V
T_{IN}	Input signal transition time		50	ns

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter		Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage		I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage		I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V TTL Low-level output voltage		I_{OL} = 24 mA V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage		I_{OL} = 24 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current		V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current		V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins		V_{IN} = GND f = 1.0 MHz		6.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FOE0, FOE1)		V_{IN} = GND f = 1.0 MHz		8.0	pF
C_{IN}	Input capacitance for Fast Inputs		V_{IN} = GND f = 1.0 MHz		12.0	pF
C_{OUT}[1]	Output capacitance		V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}[2]	Supply current	(Non Q)	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5V f = 1.0 MHz @ 25°C	126 Typ		mA
		(Q)		55 Typ		

Notes: 1. Sample tested.
2. Measured with device programmed as two 16-bit counters.

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time		80	160	µs

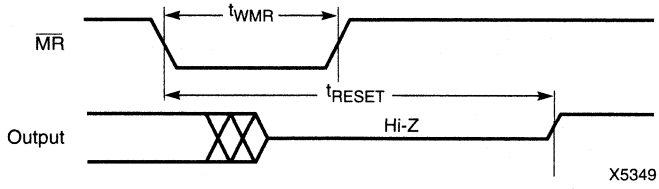


Figure 3: Global Reset Waveform

Fast Function Block (FFB) External AC Characteristics [1]

Symbol	Parameter	XC7336-5		XC7336-7		XC7336-10		XC7336-12		XC7336-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
t_{PD}	Fast input to output valid [2]		5.0		7.5		10.0		12.0		15.0	ns
	I/O or input to output valid [2]		8.5		12.0		15.0		19.0		23.0	ns
t_{SU}	Fast input setup time before FCLK	4.5		5.0		5.0		6.0		7.0		ns
	I/O or input setup time before FCLK	7.0		8.5		10.0		13.0		15.0		ns
t_H	Fast, I/O or input hold time after FCLK	0		0		0		0		0		ns
t_{CO}	FCLK input to output valid		4.5		4.5		8.0		9.0		12.0	ns
t_{FOE}	FOE input to output valid		7.0		7.5		10.0		12.0		15.0	ns
t_{FOD}	FOE input to output disable		7.0		7.5		10.0		12.0		15.0	ns
f_{MAX}	Max count frequency [2, 3]	167.0		125.0		100.0		80.0		66.7		MHz
t_{WLH}	Fast Clock pulse width (High or Low)	3.0		4.0		5.0		5.5		6.0		ns

Symbol	Parameter	XC7336Q-10		XC7336Q-12		XC7336Q-15		Units
		Min	Max	Min	Max	Min	Max	
t_{PD}	Fast input to output valid [2]		10.0		12.0		15.0	ns
	I/O or input to output valid [2]		15.0		19.0		23.0	ns
t_{SU}	Fast input setup time before FCLK	6.5		6.5		7.0		ns
	I/O or input setup time before FCLK	11.5		13.5		15.0		ns
t_H	Fast, I/O or input hold time after FCLK	0		0		0		ns
t_{CO}	FCLK input to output valid		6.5		8.5		12.0	ns
t_{FOE}	FOE input to output valid		10.0		12.0		15.0	ns
t_{FOD}	FOE input to output disable		10.0		12.0		15.0	ns
f_{MAX}	Max count frequency [2, 3]	100.0		80.0		66.7		MHz
t_{WLH}	Fast Clock pulse width (High or Low)	5.0		5.5		6.0		ns

Notes: 1. All appropriate ac specifications tested using Figure 5 as test load circuit.
2. Assumes four product terms per output.
3. Export Control Max. flip-flop toggle rate.

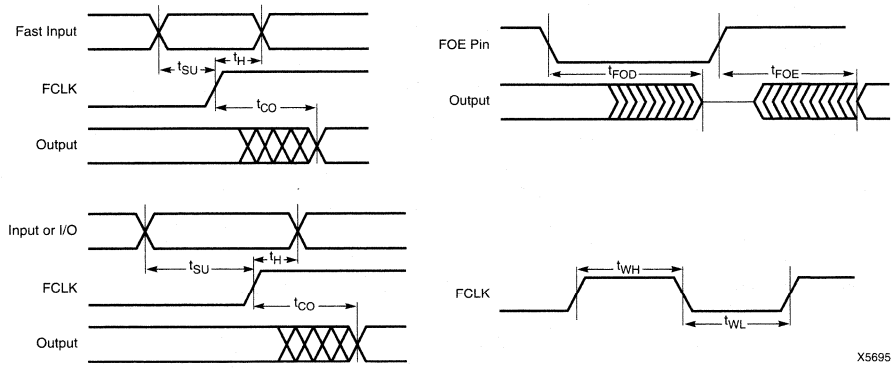
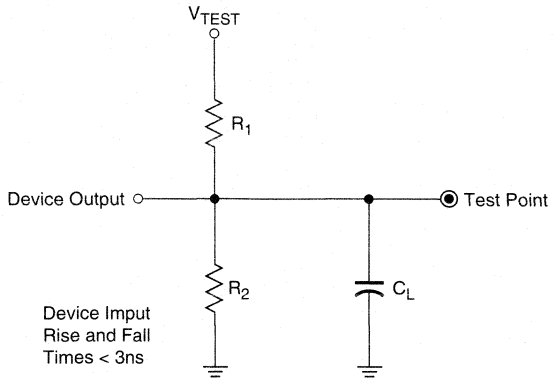


Figure 4: Switching Waveform



V_{CCIO} Level	V_{TEST}	R_1	R_1	C_L
5 V	5.0 V	160 Ω	120 Ω	35 pF
3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X5222

Figure 5: AC Load Circuit

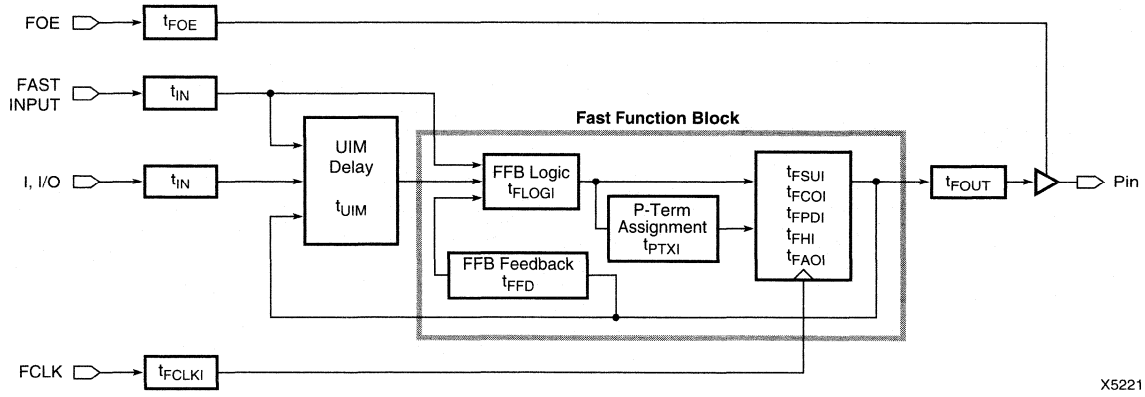


Figure 6: XC7336 Timing Model

Timing Model

Timing within the XC7336 is accurately determined using external timing parameters from the device data sheet, using a variety of CAE simulators, or with the timing model shown in Figure 6.

The timing model is based on the fixed internal delays of the XC7336 architecture that consists of three basic parts: I/O Blocks, the UIM and Fast Function Blocks. The timing model identifies the internal delay paths and their relationships to ac characteristics. Using this model and the ac characteristics, designers can easily calculate the timing information for the XC7336.

Fast Function Block (FFB) Internal AC Characteristics

Symbol	Parameter	XC7336-5		XC7336-7		XC7336-10		XC7336-12		XC7336-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay [1]		1.0		1.5		1.5		2.0		2.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay [1]		2.0		3.5		5.5		7.0		8.0	ns
t_{FSUI}	FFB register setup time	2.5		1.5		2.5		3.0		4.0		ns
t_{FHI}	FFB register hold time	1.0		2.5		2.5		3.0		3.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		0.5		0.5		1.0		1.0	ns
t_{FAOI}	FFB register async. set delay		2.0		2.0		2.5		3.0		4.0	ns
t_{PTXI}	FFB p-term assignment delay		0.6		0.8		1.0		1.2		1.5	ns
t_{FFD}	FFB feedback delay		0.5		4.0		5.0		6.5		8.0	ns

Symbol	Parameter	XC7336Q-10		XC7336Q-12		XC7336Q-15		Units
		Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay [1]		3.0		3.0		2.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay [1]		5.5		7.0		8.0	ns
t_{FSUI}	FFB register setup time	2.5		3.0		4.0		ns
t_{FHI}	FFB register hold time	2.5		3.0		3.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		1.0		1.0	ns
t_{FAOI}	FFB register async. set delay		2.5		3.0		4.0	ns
t_{PTXI}	FFB p-term assignment delay		1.0		1.2		1.5	ns
t_{FFD}	FFB feedback delay		5.0		6.5		8.0	ns

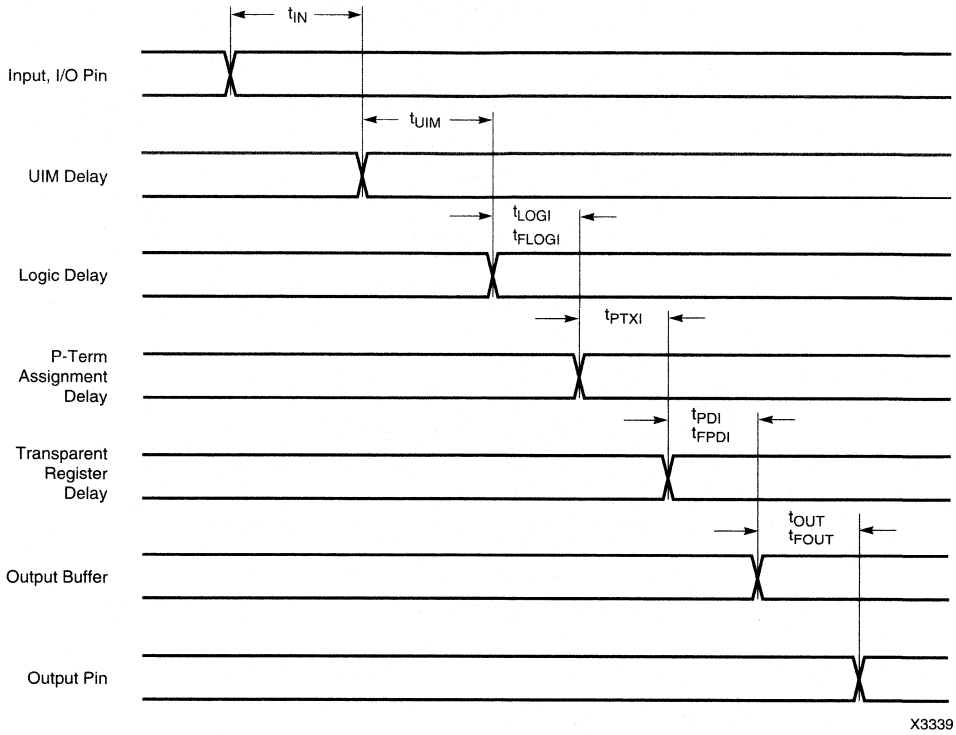
Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

Internal AC Characteristics

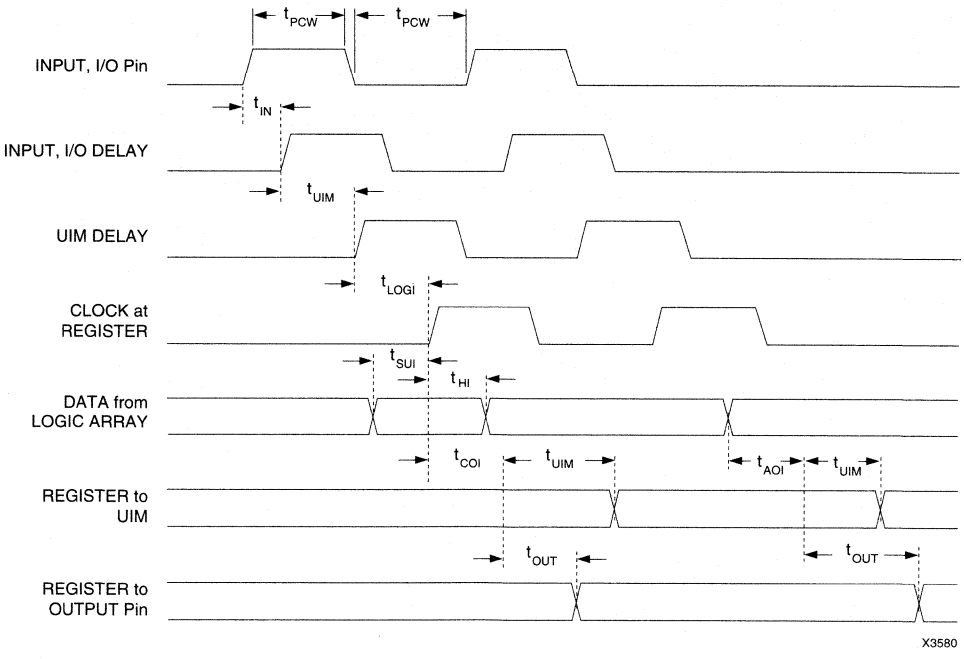
Symbol	Parameter	XC7336-5		XC7336-7		XC7336-10		XC7336-12		XC7336-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		1.5		2.5		3.5		4.0		5.0	ns
t_{FOUT}	FFB output buffer and pad delay		2.0		3.0		4.5		5.0		7.0	ns
t_{UIM}	Universal Interconnect Matrix delay		3.5		4.5		5.0		7.0		8.0	ns
t_{FCLKI}	Fast clock buffer delay		1.5		1.5		2.5		3.0		4.0	ns

Symbol	Parameter	XC7336Q-10		XC7336Q-12		XC7336Q-15		Units
		Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		3.5		4.0		5.0	ns
t_{FOUT}	FFB output buffer and pad delay		3.0		4.5		7.0	ns
t_{UIM}	Universal Interconnect Matrix delay		5.0		7.0		8.0	ns
t_{FCLKI}	Fast clock buffer delay		2.5		3.0		4.0	ns

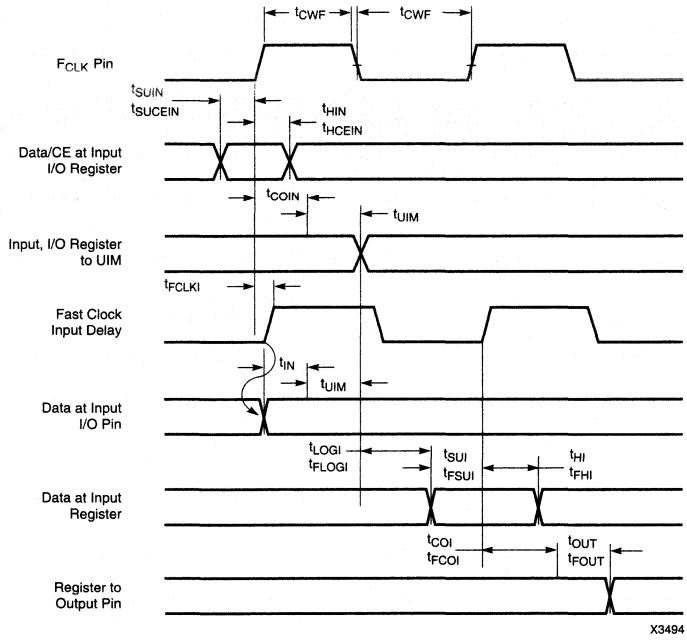
Combinational Switching Characteristics



Asynchronous Switching Characteristics



Synchronous Switching Characteristics

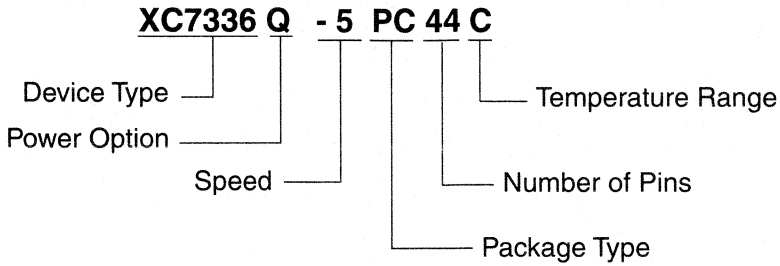


XC7336 Pinouts

VQ44/PQ44	PC44	Input	XC7336	Output
39	1	I/FO/FI	$\overline{MR}$	MC3-6
40	2	I/FO/FI		MC3-5
41	3	I/FO/FI		MC3-4
42	4	I/FO/FI		MC3-3
43	5	FO/FCLK0		MC3-2
44	6	FO/FCLK1		MC3-1
1	7	I/FO/FI		MC1-1
2	8	I/FO		MC1-2
3	9	I/FO		MC1-3
4	10		**GND**	
5	11	I/FO		MC1-4
6	12	I/FO		MC1-5
7	13	I/FO		MC1-6
8	14	I/FO		MC1-7
9	15	I/FO		MC1-8
10	16	I/FO		MC1-9
11	17	I/FO		MC4-9
12	18	I/FO/FI		MC4-8
13	19	I/FO/FI		MC4-7
14	20	I/FO/FI		MC4-6
15	21		V_{CCINT}	
16	22	I/FO		MC4-5

VQ44/PQ44	PC44	Input	XC7336	Output
17	23		**GND**	
18	24	I/FO		MC4-4
19	25	I/FO		MC4-3
20	26	I/FO		MC4-2
21	27	I/FO		MC4-1
22	28	I/FI		
23	29	I/FO		MC2-9
24	30	I/FO		MC2-8
25	31		**GND**	
26	32		V_{CCIO}	
27	33	I/FO		MC2-7
28	34	I/FO		MC2-6
29	35	I/FO		MC2-5
30	36	I/FO		MC2-4
31	37	I/FO		MC2-3
32	38	I/FO		MC2-2
33	39	FO/FOE1		MC2-1
34	40	FO/FOE0		MC3-9
35	41		V_{CCINT}/V_{PP}	
36	42	I/FI		
37	43	I/FO/FI		MC3-8
38	44	I/FO/FI		MC3-7

Ordering Information



Power Options

Q Low Power -10, -12, -15 speeds

Speed Options

-15 15 ns pin-to-pin delay
-12 12 ns pin-to-pin delay
-10 10 ns pin-to-pin delay
-7 7.5 ns pin-to-pin delay (commercial only)
-5 5 ns pin-to-pin delay (commercial only)

Packaging Options

PC44 44-Pin Plastic Leaded Chip Carrier
WC44 44-Pin Windowed Ceramic Leaded Chip Carrier
PQ44 44-Pin Plastic Quad Flat Pack
VQ44 44-Pin Thin Quad Pack

Temperature Options

C Commercial0°C to 70°C
I Industrial –40°C to 85°C

4

Component Availability

Pins		44			
Type		Plastic PLCC	Ceramic CLCC	Plastic PQFP	Plastic VQFP
Code		PC44	WC44	PQ44	VQ44
XC7336	–15	CI	CI	CI	
	–12	CI	CI	C	
	–10	CI	CI	C	
	–7	C	C	C	
	–5	C	C	C	
XC7336Q	–15	CI	CI	C	C
	–12	CI	CI	C	C
	–10	C	C	C	C

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

Features

- High-performance Complex Programmable Logic Devices (CPLDs)
  - 7.5 ns pin-to-pin speeds on all fast inputs
  - Up to 125 MHz maximum clock frequency
- 100% PCI compliant
- 18 outputs with 24 mA drive
- I/O operation at 3.3 V or 5 V
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- 100% interconnect matrix
  - Maximizes resource utilization
  - Wire-AND capability via SMARTswitch
- High-speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 61 MHz 18-bit accumulators
- Multiple independent clocks
- Up to 54 inputs programmable as direct, latched, or registered
- Power management options
- Multiple security bits for design protection
- 54 macrocells with programmable I/O architecture
- Advanced Dual-Block architecture
  - 2 Fast Function Blocks
  - 4 High-Density Function Blocks
- 0.8 µ CMOS EPROM technology
- Available in 44-pin and 68-pin PLCC and CLCC packages

General Description

The XC7354 is a high performance CPLD providing general purpose logic integration. It consists of two PAL-like 24V9 Fast Function Blocks and four High Density Function Blocks interconnected by the 100%-populated Universal Interconnect Matrix (UIM™).

Power Management

The XC7354 features a power-management scheme that permits non-speed-critical paths of a design to be operated at reduced power. Overall power dissipation is often reduced significantly, since, in most systems only a few paths are speed critical.

Macrocells can individually be specified for high performance or low power operation by adding attributes to the logic schematic, or declaration statements to the behavioral description. To minimize power dissipation, unused Function Blocks are turned off and unused macrocells in used Function Blocks are configured for low power operation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

$$I_{CC}\ (mA) = MC_{HP}\ (3.0) + MC_{LP}\ (2.6) + MC\ (0.006\ mA/MHz)\ f$$

Where:

MC_{HP} = Macrocells in high-performance mode
MC_{LP} = Macrocells in low-power mode
MC = Total number of macrocells used
f = Clock frequency (MHz)

Figure 1 shows a typical power calculation for the XC7354 device, programmed as three 16-bit counters and operating at the indicated clock frequency.

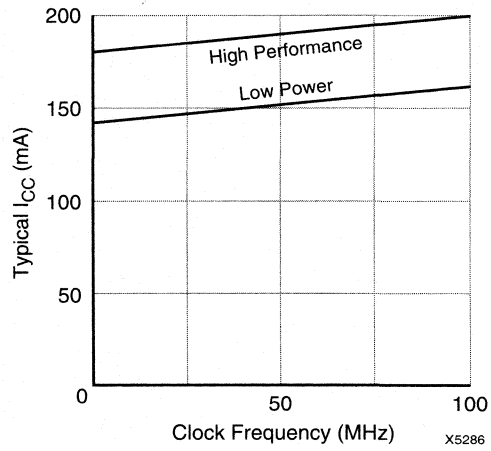


Figure 1: Typical I_{CC} vs. Frequency for XC7354

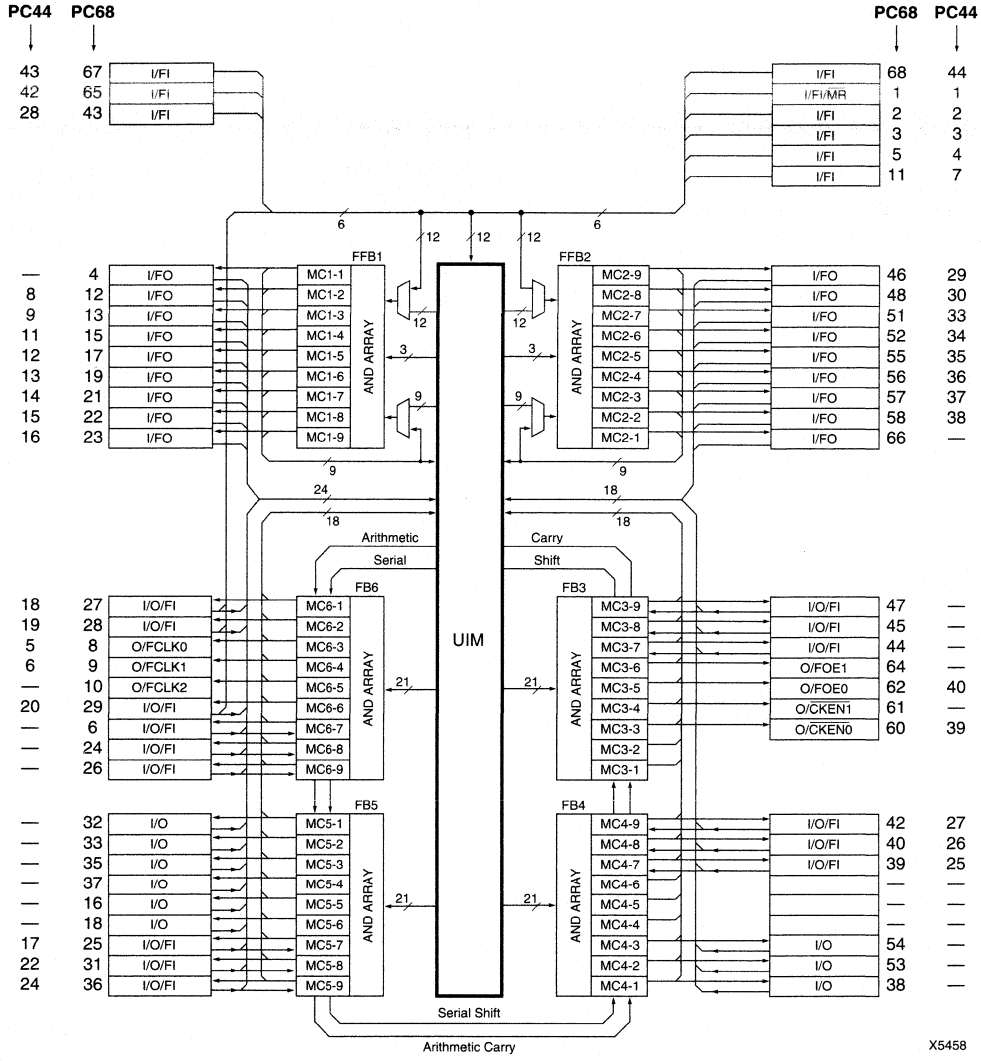


Figure 2: XC7354 Architecture

Absolute Maximum Ratings

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

Warning: 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.

Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units
V_{CCINT} V_{CCIO}	Supply voltage relative to GND Commercial T_A = 0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial T_A = –40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND Military T_A = –55°C to T_C + 125°C	4.5	5.5	V
V_{CCIO}	I/O supply voltage relative to GND	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{IH}	High-level input voltage	2.0	V_{CC} +0.5	V
V_O	Output voltage	0	V_{CCIO}	V
T_{IN}	Input signal transition time		50	ns

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V TTL Low-level output voltage	I_{OL} = 24 mA (FO) I_{OL} = 12 mA (I/O) V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins	V_{IN} = GND f = 1.0 MHz		8.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FCLK2, FOE0, FOE1)	V_{IN} = GND f = 1.0 MHz		12.0	pF
C_{OUT}[1]	Output capacitance	V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}[2]	Supply current (low power mode)	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5V f = 1.0 MHz @ 25°C	140 Typ		mA

Notes: 1. Sample tested.
2. Measured with device programmed as three 16-bit counters.

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time		80	160	µs

Fast Function Block (FFB) External AC Characteristics[3]

		XC7354-7 (Com only)		XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		
Symbol	Parameter	Min	Max	Min	Max	Min	Max	Min	Max	Units
f_{CF}	Max count frequency [1, 2, 4]	125.0		100.0		80.0		66.7		MHz
t_{SUF}	Fast input setup time before FCLK ↑ [1]	4.0		5.0		6.0		7.0		ns
t_{HF}	Fast input hold time after FCLK ↑	0		0		0		0		ns
t_{COF}	FCLK ↑ to output valid		5.5		8.0		9.0		12.0	ns
t_{PDFO}	Fast input to output valid [1, 2]		7.5		10.0		12.0		15.0	ns
t_{PDFU}	I/O to output valid [1, 2]		12.0		16.0		19.0		23.0	ns
t_{CWF}	Fast clock pulse width	4.0		5.0		5.5		6.0		ns

Notes: 1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.
3. All appropriate AC specifications tested using Figure 3 as the test load circuit.
4. Export Control Max. flip-flop toggle rate.

High-Density Function Block (FB) External AC Characteristics

		XC7354-7 (Com only)		XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		
Symbol	Parameter	Min	Max	Min	Max	Min	Max	Min	Max	Units
f_C	Max count frequency [1, 2]	95.2		76.9		66.7		55.6		MHz
t_{SU}	I/O setup time before FCLK ↑ [1, 2]	10.5		13.0		15.0		18.0		ns
t_H	I/O hold time after FCLK ↑	0		0		0		0		ns
t_{CO}	FCLK ↑ to output valid		7.0		10.0		12.0		15.0	ns
t_{PSU}	I/O setup time before p-term clock ↑ [2]	4.0		6.0		7.0		9.0		ns
t_{PH}	I/O hold time after p-term clock ↑	0		0		0		0		ns
t_{PCO}	P-term clock ↑ to output valid		13.5		17.0		20.0		24.0	ns
t_{PD}	I/O to output valid [1, 2]		16.5		22.0		27.0		32.0	ns
t_{CW}	Fast clock pulse width	4.0		5.0		5.5		6.0		ns
t_{PCW}	P-term clock pulse width	5.0		6.0		7.5		8.5		ns

Notes: 1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

Fast Function Block (FFB) Internal AC Characteristics

Symbol	Parameter	XC7354-7 (Com only)		XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay [1]		1.5		1.5		2.0		2.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay [1]		3.5		5.5		7.0		8.0	ns
t_{FSUI}	FFB register setup time	1.5		2.5		3.0		4.0		ns
t_{FHI}	FFB register hold time	2.5		2.5		3.0		3.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		0.5		1.0		1.0	ns
t_{FAOI}	FFB register async. set delay		2.0		2.5		3.0		4.0	ns
t_{PTXI}	FFB p-term assignment delay		0.8		1.0		1.2		1.5	ns
t_{FFD}	FFB feedback delay		4.0		5.0		6.5		8.0	ns

Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

High-Density Function Block (FB) Internal AC Characteristics

Symbol	Parameter	XC7354-7 (Com only)		XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{LOGI}	FB logic array delay [1]		3.5		3.5		4.0		5.0	ns
t_{LOGILP}	Low power FB logic delay [1]		7.0		7.5		9.0		11.0	ns
t_{SUI}	FB register setup time	1.5		2.5		3.0		4.0		ns
t_{HI}	FB register hold time	3.5		3.5		4.0		5.0		ns
t_{COI}	FB register clock-to-output delay		1.0		1.0		1.0		1.0	ns
t_{PDI}	FB register pass through delay		1.5		2.5		4.0		4.0	ns
t_{AOI}	FB register async. set/reset delay		2.5		3.0		4.0		5.0	ns
t_{RA}	Set/reset recovery time before FCLK ↑	13.5		16.0		18.0		21.0		ns
t_{HA}	Set/reset hold time after FCLK ↑	0		0		0		0		ns
t_{PRA}	Set/reset recovery time before p-term clock ↑	7.5		10.0		12.0		15.0		ns
t_{PHA}	Set/reset hold time after p-term clock ↑	5.0		6.0		8.0		9.0		ns
t_{PCI}	FB p-term clock delay		1.0		0		0		0	ns
t_{OEI}	FB p-term output enable delay		3.0		4.0		5.0		7.0	ns
t_{CARY8}	ALU carry delay within 1 FB [2]		5.0		6.0		8.0		12.0	ns
t_{CARYFB}	Carry lookahead delay per additional Functional Block [2]		1.0		1.5		2.0		3.0	ns

Notes: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.
2. Arithmetic carry delays are measured as the increase in required set-up time to adjacent macrocell(s) for adder with registered outputs.

4

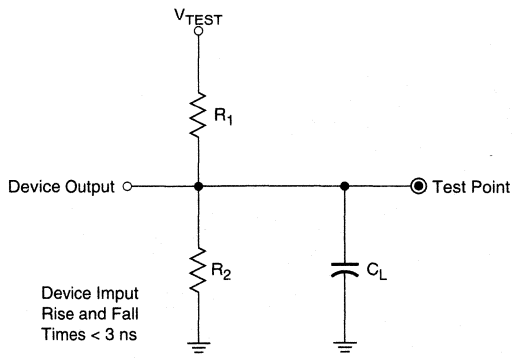
I/O Block External AC Characteristics

Symbol	Parameter	XC7354-7 (Com only)		XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{IN}	Max pipeline frequency (input register to FFB or FB register) [1]	95.2		76.9		66.7		55.6		MHz
t_{SUIN}	Input register/latch setup time before FCLK ↑	4.0		5.0		6.0		7.0		ns
t_{HIN}	Input register/latch hold time after FCLK ↑	0		0		0		0		ns
t_{COIN}	FCLK ↑ to input register/latch output		2.5		3.5		4.0		5.0	ns
t_{CESUIN}	Clock enable setup time before FCLK ↑	5.0		7.0		8.0		10.0		ns
t_{CEHIN}	Clock enable hold time after FCLK ↑	0		0		0		0		ns
t_{CWHIN}	FCLK pulse width high time	4.0		5.0		5.5		6.0		ns
t_{CWLIN}	FCLK pulse width low time	4.0		5.0		5.5		6.0		ns

Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

Internal AC Characteristics

Symbol	Parameter	XC7354-7 (Com only)		XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		2.5		3.5		4.0		5.0	ns
t_{FOUT}	FFB output buffer and pad delay		3.0		4.5		5.0		7.0	ns
t_{OUT}	FB output buffer and pad delay		4.5		6.5		8.0		10.0	ns
t_{UIM}	Universal Interconnect Matrix delay		4.5		6.0		7.0		8.0	ns
t_{FOE}	FOE input to output valid		7.5		10.0		12.0		15.0	ns
t_{FOD}	FOE input to output disable		7.5		10.0		12.0		15.0	ns
t_{FCLKI}	Fast clock buffer delay		1.5		2.5		3.0		4.0	ns



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
FO	5.0 V	5.0 V	160 Ω	120 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X3491

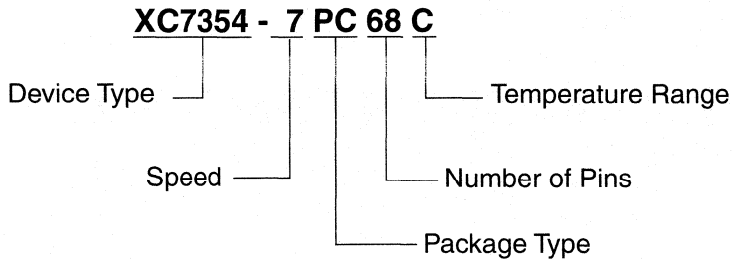
Figure 3: AC Load Circuit

XC7354 Pinouts

PC68	PC44	Input	XC7354	Output
1	1	I/FI/ $\overline{MR}$		
2	2	I/FI		
3	3	I/FI		
4	–	I/FO		MC1-1
5	4	I/FI		
6	–	I/O/FI		MC6-7
7	–		**GND**	
8	5	O/FCLK0		MC6-3
9	6	O/FCLK1		MC6-4
10	–	O/FCLK2		MC6-5
11	7	I/FI		
12	8	I/FO		MC1-2
13	9	I/FO		MC1-3
14	10		**GND**	
15	11	I/FO		MC1-4
16	–	I/O		MC5-5
17	12	I/FO		MC1-5
18	–	I/O		MC5-6
19	13	I/FO		MC1-6
20	–		V_{CCIO}	
21	14	I/FO		MC1-7
22	15	I/FO		MC1-8
23	16	I/FO		MC1-9
24	–	I/O/FI		MC6-8
25	17	I/O/FI		MC5-7
26	–	I/O/FI		MC6-9
27	18	I/O/FI		MC6-1
28	19	I/O/FI		MC6-2
29	20	I/O/FI		MC6-6
30	21		V_{CCINT}	
31	22	I/O/FI		MC5-8
32	–	I/O		MC5-1
33	–	I/O		MC5-2
34	23		**GND**	

PC68	PC44	Input	XC7354	Output
35	–	I/O		MC5-3
36	24	I/O/FI		MC5-9
37	–	I/O		MC5-4
38	–	I/O		MC4-1
39	25	I/O/FI		MC4-7
40	26	I/O/FI		MC4-8
41	–		**GND**	
42	27	I/O/FI		MC4-9
43	28	I/FI		
44	–	I/O/FI		MC3-7
45	–	I/O/FI		MC3-8
46	29	I/FO		MC2-9
47	–	I/O/FI		MC3-9
48	30	I/FO		MC2-8
49	31		**GND**	
50	32		V_{CCIO}	
51	33	I/FO		MC2-7
52	34	I/FO		MC2-6
53	–	I/O		MC4-2
54	–	I/O		MC4-3
55	35	I/FO		MC2-5
56	36	I/FO		MC2-4
57	37	I/FO		MC2-3
58	38	I/FO		MC2-2
59	–		V_{CCINT}	
60	39	O/$\overline{CKEN0}$		MC3-3
61	–	O/$\overline{CKEN1}$		MC3-4
62	40	O/FOE0		MC3-5
63	41		V_{CCINT}/V_{PP}	
64	–	O/FOE1		MC3-6
65	42	I/FI		
66	–	I/FO		MC2-1
67	43	I/FI		
68	44	I/FI		

Ordering Information



Speed Options

-15	15 ns pin-to-pin delay
-12	12 ns pin-to-pin delay
-10	10 ns pin-to-pin delay (commercial and industrial only)
-7	7.5 ns pin-to-pin delay (commercial only)

Packaging Options

PC44	44-Pin Plastic Leaded Chip Carrier
WC44	44-Pin Windowed Ceramic Leaded Chip Carrier
PC68	68-Pin Plastic Leaded Chip Carrier
WC68	68-Pin Windowed Ceramic Leaded Chip Carrier

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	-40°C to 85°C
M	Military	-55°C (Ambient) to 125°C (Case)

Component Availability

Pins		44		68	
Type		Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC
Code		PC44	WC44	PC68	WC68
XC7354	–15	CI	CI	CI	CIM
	–12	CI	CI	CI	CIM
	–10	CI	CI	CI	CI
	–7	C	C	C	C

C = Commercial = 0° to +70°C I = Industrial = –40° to 85°C M = Military = –55°C(A) to 125°C (C)

Features

- High-performance Complex Programmable Logic Devices (CPLDs)
  - 7.5 ns pin-to-pin speeds on all fast inputs
  - Up to 125 MHz maximum clock frequency
- 100% PCI compliant
- 18 outputs with 24 mA drive
- I/O operation at 3.3 V or 5 V
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- 100% interconnect matrix
  - Maximizes resource utilization
  - Wire-AND capability via SMARTswitch
- High-speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 61 MHz 18-bit accumulators
- Multiple independent clocks
- Up to 84 inputs programmable as direct, latched, or registered
- Power management options
- Multiple security bits for design protection
- 72 macrocells with programmable I/O architecture
- Advanced Dual-Block architecture
  - 2 Fast Function Blocks
  - 6 High-Density Function Blocks
- 0.8 µ CMOS EPROM technology
- Available in 68-pin and 84-pin PLCC/CLCC and 100-pin PQFP packages

General Description

The XC7372 is a high performance CPLD providing general purpose logic integration. It consists of two PAL-like 24V9 Fast Function Blocks and six High Density Function Blocks interconnected by the 100%-populated Universal Interconnect Matrix (UIM™). See Figure 2 for the architecture overview.

Power Management

The XC7372 features a power-management scheme that permits non-speed-critical paths of a design to be operated at reduced power. Overall power dissipation is often reduced significantly, since, in most systems only a few paths are speed critical.

Macrocells can individually be specified for high performance or low power operation by adding attributes to the logic schematic, or declaration statements to the behavioral description. To minimize power dissipation, unused Function Blocks are turned off and unused macrocells in used Function Blocks are configured for low power operation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

$$I_{CC}\ (mA) = MC_{HP}\ (3.1) + MC_{LP}\ (2.6) + MC\ (0.012\ mA/MHz)\ f$$

Where:

MC_{HP} = Macrocells in high-performance mode
MC_{LP} = Macrocells in low-power mode
MC = Total number of macrocells used
f = Clock frequency (MHz)

Figure 1 shows a typical power calculation for the XC7372 device, programmed as four 16-bit counters and operating at the indicated clock frequency.

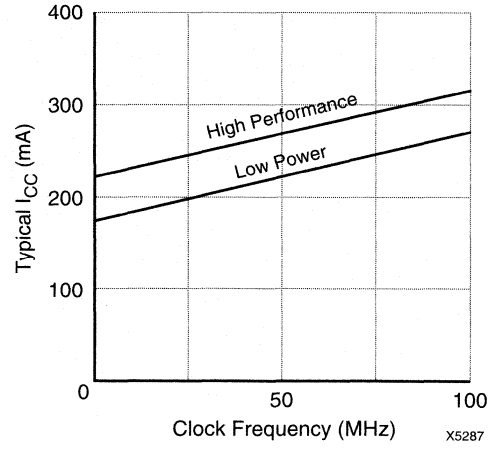


Figure 1: Typical I_{CC} vs. Frequency for XC7372

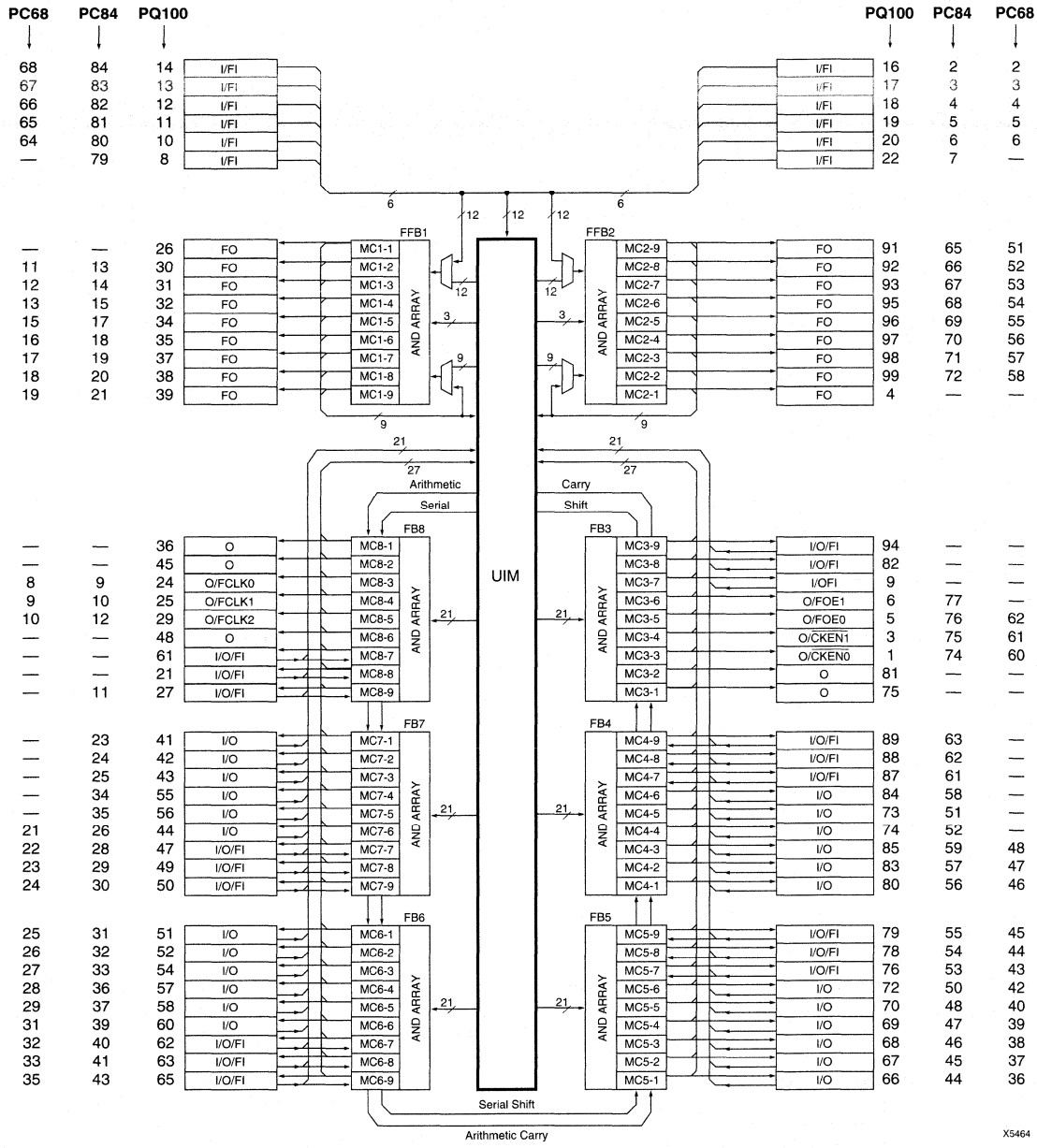


Figure 2: XC7372 Architecture

Absolute Maximum Ratings

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

Warning: 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.

Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units
V_{CCINT}/ V_{CCIO}	Supply voltage relative to GND Commercial T_A = 0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial T_A = –40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND Military T_A = –55°C to T_C + 125°C	4.5	5.5	V
V_{CCIO}	I/O supply voltage relative to GND	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{IH}	High-level input voltage	2.0	V_{CC} +0.5	V
V_O	Output voltage	0	V_{CCIO}	V
T_{IN}	Input signal transition time		50	ns

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V TTL Low-level output voltage	I_{OL} = 24 mA (FO) I_{OL} = 12 mA (I/O) V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins	V_{IN} = GND f = 1.0 MHz		8.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FCLK2, FOE0, FOE1)	V_{IN} = GND f = 1.0 MHz		12.0	pF
C_{OUT}[1]	Output capacitance	V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}[2]	Supply current (low power mode)	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5V f = 1.0 MHz @ 25°C	187 Typ		mA

Notes: 1. Sample tested.
2. Measured with device programmed as four 16-bit counters.

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time		80	160	µs

Fast Function Block (FFB) External AC Characteristics[3]

Symbol	Parameter	XC7372-7 (Com only)		XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{CF}	Max count frequency [1, 2, 4]	125.0		100.0		80.0		66.7		MHz
t_{SUF}	Fast input setup time before FCLK ↑ [1]	4.0		5.0		6.0		7.0		ns
t_{HF}	Fast input hold time after FCLK ↑	0		0		0		0		ns
t_{COF}	FCLK ↑ to output valid		5.5		8.0		9.0		12.0	ns
t_{PDFO}	Fast input to output valid [1, 2]		7.5		10.0		12.0		15.0	ns
t_{PDFU}	I/O to output valid [1, 2]		14.0		17.0		20.0		24.0	ns
t_{CWF}	Fast clock pulse width (High or Low)	4.0		5.0		5.5		6.0		ns

Notes: 1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.
3. All appropriate specifications tested using Figure 3 as the test load circuit.
4. Export Control Max. flip-flop toggle rate.

High-Density Function Block (FB) External AC Characteristics

Symbol	Parameter	XC7372-7 (Com only)		XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_C	Max count frequency [1, 2]	95.2		71.4		62.5		52.6		MHz
t_{SU}	I/O setup time before FCLK ↑ [1, 2]	12.5		14.0		16.0		19.0		ns
t_H	I/O hold time after FCLK ↑	0		0		0		0		ns
t_{CO}	FCLK ↑ to output valid		7.0		10.0		12.0		15.0	ns
t_{PSU}	I/O setup time before p-term clock ↑ [2]	6.0		6.0		7.0		9.0		ns
t_{PH}	I/O hold time after p-term clock ↑	0		0		0		0		ns
t_{PCO}	P-term clock ↑ to output valid		13.5		18.0		21.0		25.0	ns
t_{PD}	I/O to output valid [1, 2]		18.5		23.0		28.0		33.0	ns
t_{CW}	Fast clock pulse width	4.0		5.0		5.5		6.0		ns
t_{PCW}	P-term clock pulse width	5.0		6.0		7.5		8.5		ns

Notes: 1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

Fast Function Block (FFB) Internal AC Characteristics

Symbol	Parameter	XC7372-7 (Com only)		XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay [1]		1.5		1.5		2.0		2.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay [1]		3.5		5.5		7.0		8.0	ns
t_{FSUI}	FFB register setup time	1.5		2.5		3.0		4.0		ns
t_{FHI}	FFB register hold time	2.5		2.5		3.0		3.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		0.5		1.0		1.0	ns
t_{FAOI}	FFB register async. set delay		2.0		2.5		3.0		4.0	ns
t_{PTXI}	FFB p-term assignment delay		0.8		1.0		1.2		1.5	ns
t_{FFD}	FFB feedback delay		4.0		5.0		6.5		8.0	ns

Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

High-Density Function Block (FB) Internal AC Characteristics

Symbol	Parameter	XC7372-7 (Com only)		XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{LOGI}	FB logic array delay [1]		3.5		3.5		4.0		5.0	ns
t_{LOGILP}	Low power FB logic delay [1]		7.0		7.5		9.0		11.0	ns
t_{SUI}	FB register setup time	1.5		2.5		3.0		4.0		ns
t_{HI}	FB register hold time	3.5		3.5		4.0		5.0		ns
t_{COI}	FB register clock-to-output delay		1.0		1.0		1.0		1.0	ns
t_{PDI}	FB register pass through delay		1.5		2.5		4.0		4.0	ns
t_{AOI}	FB register async. set/reset delay		2.5		3.0		4.0		5.0	ns
t_{RA}	Set/reset recovery time before FCLK ↑	13.5		17.0		19.0		22.0		ns
t_{HA}	Set/reset hold time after FCLK ↑	0		0		0		0		ns
t_{PRA}	Set/reset recovery time before p-term clock ↑	7.5		10.0		12.0		15.0		ns
t_{PHA}	Set/reset hold time after p-term clock ↑	5.0		6.0		8.0		9.0		ns
t_{PCI}	FB p-term clock delay		1.0		0		0		0	ns
t_{OEI}	FB p-term output enable delay		3.0		4.0		5.0		7.0	ns
t_{CARY8}	ALU carry delay within 1 FB [2]		5.0		6.0		8.0		12.0	ns
t_{CARYFB}	Carry lookahead delay per additional Functional Block [2]		1.0		1.5		2.0		3.0	ns

Notes: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.
2. Arithmetic carry delays are measured as the increase in required set-up time to adjacent macrocell(s) for adder with registered outputs.

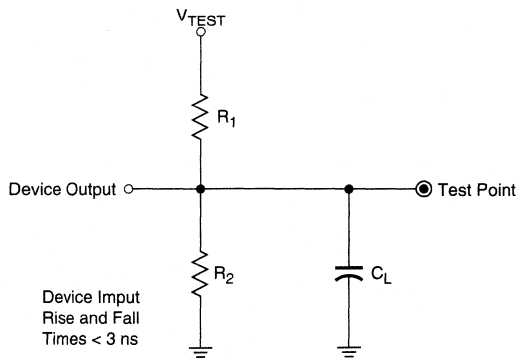
I/O Block External AC Characteristics

Symbol	Parameter	XC7372-7 (Com only)		XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{IN}	Max pipeline frequency (input register to FFB or FB register) [1]	95.2		71.4		62.5		52.6		MHz
t_{SUIN}	Input register/latch setup time before FCLK ↑	4.0		5.0		6.0		7.0		ns
t_{HIN}	Input register/latch hold time after FCLK ↑	0		0		0		0		ns
t_{COIN}	FCLK ↑ to input register/latch output		2.5		3.5		4.0		5.0	ns
t_{CESUIN}	Clock enable setup time before FCLK ↑	5.0		7.0		8.0		10.0		ns
t_{CEHIN}	Clock enable hold time after FCLK ↑	0		0		0		0		ns
t_{CWHIN}	FCLK pulse width high time	4.0		5.0		5.5		6.0		ns
t_{CWLIN}	FCLK pulse width low time	4.0		5.0		5.5		6.0		ns

Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

Internal AC Characteristics

Symbol	Parameter	XC7372-7 (Com only)		XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		2.5		3.5		4.0		5.0	ns
t_{FOUT}	FFB output buffer and pad delay		3.0		4.5		5.0		7.0	ns
t_{OUT}	FB output buffer and pad delay		4.5		6.5		8.0		10.0	ns
t_{UIM}	Universal Interconnect Matrix delay		6.5		7.0		8.0		9.0	ns
t_{FOE}	FOE input to output valid		7.5		10.0		12.0		15.0	ns
t_{FOD}	FOE input to output disable		7.5		10.0		12.0		15.0	ns
t_{FCLKI}	Fast clock buffer delay		1.5		2.5		3.0		4.0	ns



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
FO	5.0 V	5.0 V	160 Ω	120 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF

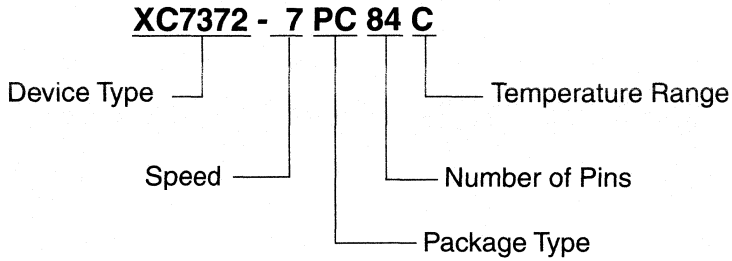
X3491

Figure 3: AC Load Circuit

XC7372 Pinouts

PQ100	PC84	PC68	Input	XC7372	Output
15	1	1		$\overline{MR}$	
16	2	2	I/FI		
17	3	3	I/FI		
18	4	4	I/FI		
19	5	5	I/FI		
20	6	6	I/FI		
21	–	–	I/O/FI		MC8-8
22	7	–	I/FI		
23	8	7		**GND**	
24	9	8	O/FCLK0		MC8-3
25	10	9	O/FCLK1		MC8-4
26	–	–	FO		MC1-1
27	11	–	I/O/FI		MC8-9
28	–	–		V_{CCIO}	
29	12	10	O/FCLK2		MC8-5
30	13	11	FO		MC1-2
31	14	12	FO		MC1-3
32	15	13	FO		MC1-4
33	16	14		**GND**	
34	17	15	FO		MC1-5
35	18	16	FO		MC1-6
36	–	–	O		MC8-1
37	19	17	FO		MC1-7
38	20	18	FO		MC1-8
39	21	19	FO		MC1-9
40	22	20		V_{CCIO}	
41	23	–	I/O		MC7-1
42	24	–	I/O		MC7-2
43	25	–	I/O		MC7-3
44	26	21	I/O		MC7-6
45	–	–	O		MC8-2
46	27	–		**GND**	
47	28	22	I/O/FI		MC7-7
48	–	–	O		MC8-6
49	29	23	I/O/FI		MC7-8
50	30	24	I/O/FI		MC7-9
51	31	25	I/O		MC6-1
52	32	26	I/O		MC6-2
53	–	–		V_{CCIO}	
54	33	27	I/O		MC6-3
55	34	–	I/O		MC7-4
56	35	–	I/O		MC7-5
57	36	28	I/O		MC6-4
58	37	29	I/O		MC6-5
59	38	30		V_{CCINT}	
60	39	31	I/O		MC6-6
61	–	–	I/O/FI		MC8-7
62	40	32	I/O/FI		MC6-7
63	41	33	I/O/FI		MC6-8
64	42	34		**GND**	
65	43	35	I/O/FI		MC6-9
66	44	36	I/O		MC5-1
67	45	37	I/O		MC5-2
68	46	38	I/O		MC5-3
69	47	39	I/O		MC5-4
70	48	40	I/O		MC5-5
71	49	41		**GND**	
72	50	42	I/O		MC5-6
73	51	–	I/O		MC4-5
74	52	–	I/O		MC4-4
75	–	–	O		MC3-1
76	53	43	I/O/FI		MC5-7
77	–	–		**GND**	
78	54	44	I/O/FI		MC5-8
79	55	45	I/O/FI		MC5-9
80	56	46	I/O		MC4-1
81	–	–	O		MC3-2
82	–	–	I/O/FI		MC3-8
83	57	47	I/O		MC4-2
84	58	–	I/O		MC4-6
85	59	48	I/O		MC4-3
86	60	49		**GND**	
87	61	–	I/O/FI		MC4-7
88	62	–	I/O/FI		MC4-8
89	63	–	I/O/FI		MC4-9
90	64	50		V_{CCIO}	
91	65	51	FO		MC2-9
92	66	52	FO		MC2-8
93	67	53	FO		MC2-7
94	–	–	I/O/FI		MC3-9
95	68	54	FO		MC2-6
96	69	55	FO		MC2-5
97	70	56	FO		MC2-4
98	71	57	FO		MC2-3
99	72	58	FO		MC2-2
100	73	59		V_{CCINT}	
1	74	60	O/$\overline{CKEN0}$		MC3-3
2	–	–		**GND**	
3	75	61	O/$\overline{CKEN1}$		MC3-4
4	–	–	FO		MC2-1
5	76	62	O/FOE0		MC3-5
6	77	–	O/FOE1		MC3-6
7	78	63		V_{CCINT}/ V_{PP}	
8	79	–	I/FI		
9	–	–	I/O/FI		MC3-7
10	80	64	I/FI		
11	81	65	I/FI		
12	82	66	I/FI		
13	83	67	I/FI		
14	84	68	I/FI		

Ordering Information



Speed Options

-15	15 ns pin-to-pin delay
-12	12 ns pin-to-pin delay
-10	10 ns pin-to-pin delay (commercial and industrial only)
-7	7.5 ns pin-to-pin delay (commercial only)

Packaging Options

PC68	68-Pin Plastic Leaded Chip Carrier
WC68	68-Pin Windowed Ceramic Leaded Chip Carrier
PC84	84-Pin Plastic Leaded Chip Carrier
WC84	84-Pin Windowed Ceramic Leaded Chip Carrier
PQ100	100-Pin Plastic Quad Flat Pack

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	-40°C to 85°C
M	Military	-55°C (Ambient) to 125°C (Case)

Component Availability

Pins		68		84		100
Type		Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Plastic PQFP
Code		PC68	WC68	PC84	WC84	PQ100
XC7372	–15	CI	CIM	CI	CIM	CI
	–12	CI	CIM	CI	CI	CI
	–10	CI	CI	CI	CI	CI
	–7	C	C	C	C	C

C = Commercial = 0° to +70°C I = Industrial = –40° to 85°C M = Military = –55°C(A) to 125°C (C)

XC73108
108-Macrocell CMOS CPLD

January, 1997 (Version 1.0) | Product Specification

Features

- High-performance Complex Programmable Logic Devices (CPLDs)
  - 7.5 ns pin-to-pin speeds on all fast inputs
  - Up to 125 MHz maximum clock frequency
- 100% PCI compliant
- 18 outputs with 24 mA drive
- I/O operation at 3.3 V or 5 V
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- 100% interconnect matrix
  - Maximizes resource utilization
  - Wire-AND capability via SMARTswitch
- High-speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 56 MHz 18-bit accumulators
- Multiple independent clocks
- Up to 120 inputs programmable as direct, latched, or registered
- Power management options
- Multiple security bits for design protection
- 108 macrocells with programmable I/O architecture
- Advanced Dual-Block architecture
  - 2 Fast Function Blocks
  - 10 High-Density Function Blocks
- 0.8 µ CMOS EPROM technology
- Available in 84-pin PLCC/CLCC, 144-pin PGA, 100-pin and 160-pin PQFP, and 225-pin BGA packages

General Description

The XC73108 is a high performance CPLD providing general purpose logic integration. It consists of two PAL-like 24V9 Fast Function Blocks and ten High Density Function Blocks interconnected by the 100%-populated Universal Interconnect Matrix (UIM™).

Power Management

The XC73108 features a power-management scheme that permits non-speed-critical paths of a design to be operated at reduced power. Overall power dissipation is often reduced significantly, since, in most systems only a few paths are speed critical.

Macrocells can individually be specified for high performance or low power operation by adding attributes to the logic schematic, or declaration statements to the behavioral description. To minimize power dissipation, unused Function Blocks are turned off and unused macrocells in used Function Blocks are configured for low power operation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

$$I_{CC}\ (mA) = MC_{HP}\ (2.4) + MC_{LP}\ (2.1) + MC\ (0.015\ mA/MHz)\ f$$

Where:

MC_{HP} = Macrocells in high-performance mode
MC_{LP} = Macrocells in low-power mode
MC = Total number of macrocells used
f = Clock frequency (MHz)

Figure 1 shows a typical power calculation for the XC73108 device, programmed as six 16-bit counters and operating at the indicated clock frequency.

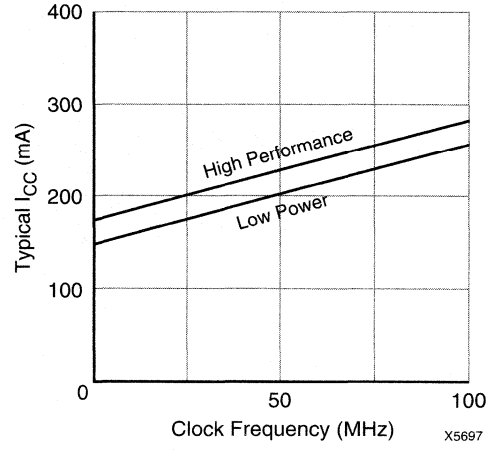


Figure 1: Typical I_{CC} vs. Frequency for XC73108

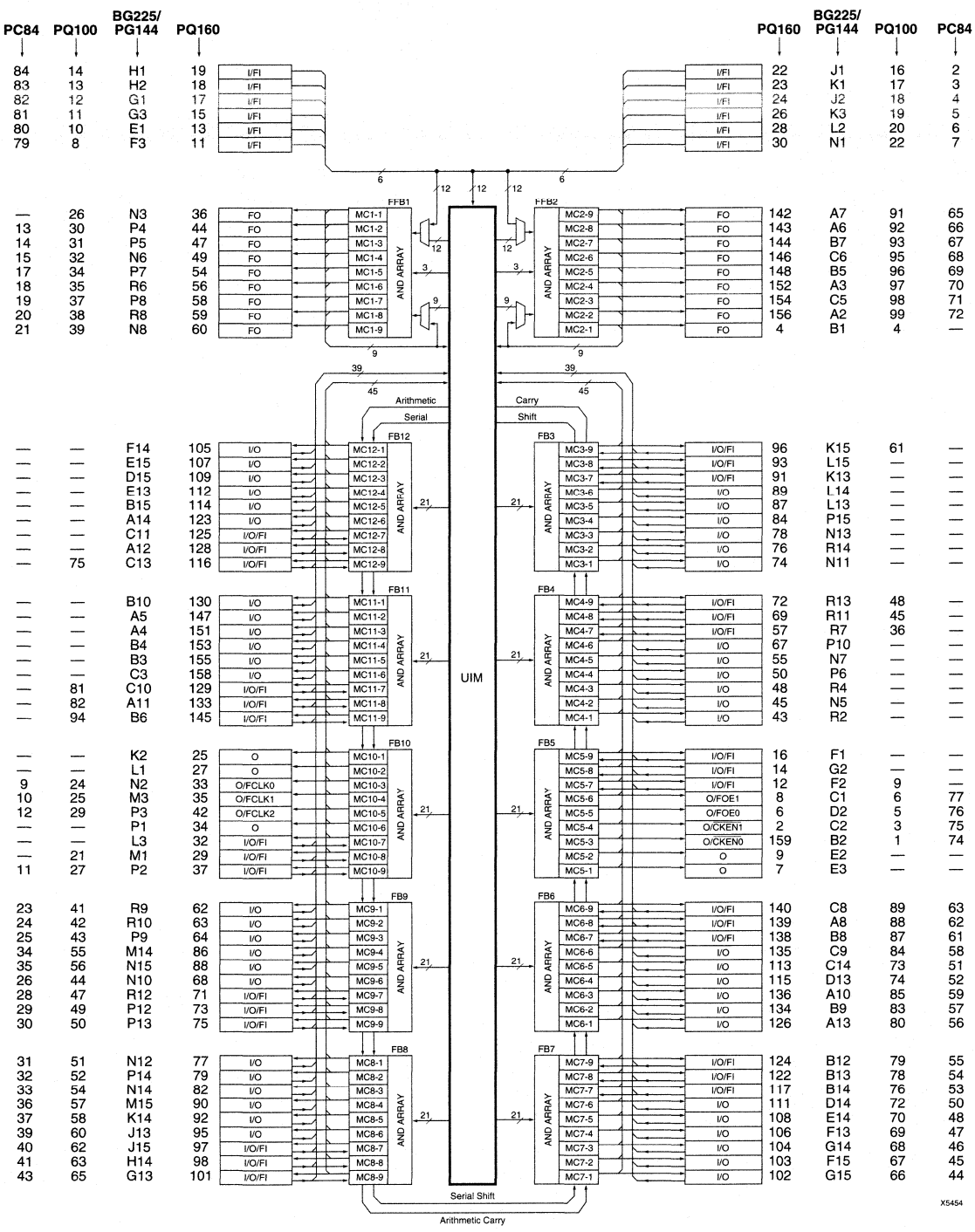


Figure 2: XC73108 Architecture

Absolute Maximum Ratings

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

Warning: 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.

Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units
V_{CCINT} V_{CCIO}	Supply voltage relative to GND Commercial T_A = 0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial T_A = –40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND Military T_A = –55°C to T_C + 125°C	4.5	5.5	V
V_{CCIO}	I/O supply voltage relative to GND	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{IH}	High-level input voltage	2.0	V_{CC} +0.5	V
V_O	Output voltage	0	V_{CCIO}	V
T_{IN}	Input signal transition time		50	ns

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V TTL Low-level output voltage	I_{OL} = 24 mA (FO) I_{OL} = 12 mA (I/O) V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins	V_{IN} = GND f = 1.0 MHz		8.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FCLK2, FOE0, FOE1)	V_{IN} = GND f = 1.0 MHz		12.0	pF
C_{OUT}[1]	Output capacitance	V_{IN} = GND f = 1.0 MHz		20.0	pF
I_{CC}[2]	Supply current (low power mode)	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5V f = 1.0 MHz @ 25°C	227 Typ		mA

Notes: 1. Sample tested.
2. Measured with device programmed as six 16-bit counters.

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time		80	160	µs

Fast Function Block (FFB) External AC Characteristics[3]

Symbol	Parameter	XC73108-7 (Com only)		XC73108-10 (Com only)		XC73108-12 (Com/Ind only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
f_{CF}	Max count frequency [1, 2, 4]	125.0		100.0		80.0		66.7		50.0		MHz
t_{SUF}	Fast input setup time before FCLK ↑ [1]	4.0		5.0		6.0		7.0		10.0		ns
t_{HF}	Fast input hold time after FCLK ↑	0		0		0		0		0		ns
t_{COF}	FCLK ↑ to output valid		5.5		8.0		9.0		12.0		15.0	ns
t_{PDFO}	Fast input to output valid [1, 2]		7.5		10.0		12.0		15.0		20.0	ns
t_{PDFU}	I/O to output valid [1, 2]		13.5		19.0		22.0		27.0		35.0	ns
t_{CWF}	Fast clock pulse width (High or Low)	4.0		5.0		5.5		6.0		6.0		ns

Notes: 1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.
3. All appropriate AC specifications tested using Figure 3 as the test load circuit.
4. Export Control Max. flip-flop toggle rate.

High-Density Function Block (FB) External AC Characteristics

Symbol	Parameter	XC73108-7 (Com only)		XC73108-10 (Com only)		XC73108-12 (Com/Ind only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
f_C	Max count frequency [1, 2]	83.3		62.5		55.6		45.5		35.7		MHz
t_{SU}	I/O setup time before FCLK ↑ [1, 2]	12.0		16.0		18.0		22.0		28.0		ns
t_H	I/O hold time after FCLK ↑	0		0		0		0		0		ns
t_{CO}	FCLK ↑ to output valid		7.0		10.0		12.0		15.0		20.0	ns
t_{PSU}	I/O setup time before p-term clock ↑ [2]	4.0		6.0		7.0		9.0		12.0		ns
t_{PH}	I/O hold time after p-term clock ↑	0		0		0		0		0		ns
t_{PCO}	P-term clock ↑ to output valid		15.0		20.0		23.0		28.0		36.0	ns
t_{PD}	I/O to output valid [1, 2]		18.0		25.0		30.0		36.0		45.0	ns
t_{CW}	Fast clock pulse width	4.0		5.0		5.5		6.0		6.0		ns
t_{PCW}	P-term clock pulse width	5.0		6.0		7.5		8.5		12.0		ns

Notes: 1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

Fast Function Block (FFB) Internal AC Characteristics

Symbol	Parameter	XC73108-7 (Com only)		XC73108-10 (Com only)		XC73108-12 (Com/Ind only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay [1]		1.5		1.5		2.0		2.0		3.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay [1]		3.5		5.5		7.0		8.0		11.0	ns
t_{FSUI}	FFB register setup time	1.5		2.5		3.0		4.0		6.0		ns
t_{FHI}	FFB register hold time	2.5		2.5		3.0		3.0		4.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		0.5		1.0		1.0		2.0	ns
t_{FAOI}	FFB register async. set delay		2.0		2.5		3.0		4.0		6.0	ns
t_{PTXI}	FFB p-term assignment delay		0.8		1.0		1.2		1.5		2.0	ns
t_{FFD}	FFB feedback delay		4.0		5.0		6.5		8.0		10.0	ns

Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

High-Density Function Block (FB) Internal AC Characteristics

Symbol	Parameter	XC73108-7 (Com only)		XC73108-10 (Com only)		XC73108-12 (Com/Ind only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
t_{LOGI}	FB logic array delay [1]		3.5		3.5		4.0		5.0		6.0	ns
t_{LOGILP}	Low power FB logic delay [1]		7.0		7.5		9.0		11.0		14.0	ns
t_{SUI}	FB register setup time	1.5		2.5		3.0		4.0		6.0		ns
t_{HI}	FB register hold time	3.5		3.5		4.0		5.0		6.0		ns
t_{COI}	FB register clock-to-output delay		1.0		1.0		1.0		1.0		1.0	ns
t_{PDI}	FB register pass through delay		1.5		2.5		4.0		4.0		4.0	ns
t_{AOI}	FB register async. set/reset delay		2.5		3.0		4.0		5.0		7.0	ns
t_{RA}	Set/reset recovery time before FCLK ↑	15.0		19.0		21.0		25.0		31.0		ns
t_{HA}	Set/reset hold time after FCLK ↑	0		0		0		0		0		ns
t_{PRA}	Set/reset recovery time before p-term clock ↑	7.5		10.0		12.0		15.0		20.0		ns
t_{PHA}	Set/reset hold time after p-term clock ↑	5.0		6.0		8.0		9.0		12.0		ns
t_{PCI}	FB p-term clock delay		1.0		0		0		0		0	ns
t_{OEI}	FB p-term output enable delay		3.0		4.0		5.0		7.0		9.0	ns
t_{CARY8}	ALU carry delay within 1 FB [2]		5.0		6.0		8.0		12.0		15.0	ns
t_{CARYFB}	Carry lookahead delay per additional Functional Block [2]		1.0		1.5		2.0		3.0		4.0	ns

Notes: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.
2. Arithmetic carry delays are measured as the increase in required set-up time to adjacent macrocell(s) for adder with registered outputs.

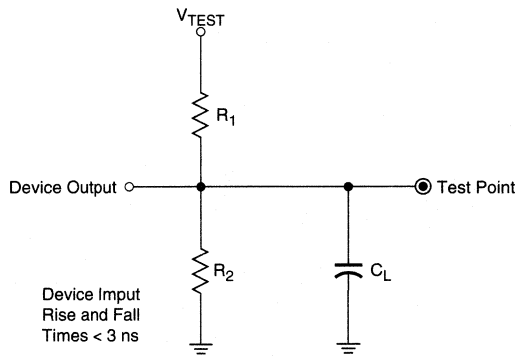
I/O Block External AC Characteristics

Symbol	Parameter	XC73108-7 (Com only)		XC73108-10 (Com only)		XC73108-12 (Com/Ind only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
f_{IN}	Max pipeline frequency (input register to FFB or FB register) [1]	83.3		62.5		55.6		45.5		35.7		MHz
t_{SUIN}	Input register/latch setup time before FCLK ↑	4.0		5.0		6.0		7.0		10.0		ns
t_{HIN}	Input register/latch hold time after FCLK ↑	0		0		0		0		0		ns
t_{COIN}	FCLK ↑ to input register/latch output		2.5		3.5		4.0		5.0		6.0	ns
t_{CESUIN}	Clock enable setup time before FCLK ↑	5.0		7.0		8.0		10.0		12.0		ns
t_{CEHIN}	Clock enable hold time after FCLK ↑	0		0		0		0		0		ns
t_{CWHIN}	FCLK pulse width high time	4.0		5.0		5.5		6.0		6.0		ns
t_{CWLIN}	FCLK pulse width low time	4.0		5.0		5.5		6.0		6.0		ns

Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

Internal AC Characteristics

Symbol	Parameter	XC73108-7 (Com only)		XC73108-10 (Com only)		XC73108-12 (Com/Ind only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		2.5		3.5		4.0		5.0		6.0	ns
t_{FOUT}	FFB output buffer and pad delay		3.0		4.5		5.0		7.0		9.0	ns
t_{OUT}	FB output buffer and pad delay		4.5		6.5		8.0		10.0		14.0	ns
t_{UIM}	Universal Interconnect Matrix delay		6.0		9.0		10.0		12.0		15.0	ns
t_{FOE}	FOE input to output valid		7.5		10.0		12.0		15.0		20.0	ns
t_{FOD}	FOE input to output disable		7.5		10.0		12.0		15.0		20.0	ns
t_{FCLKI}	Fast clock buffer delay		1.5		2.5		3.0		4.0		5.0	ns



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
FO	5.0 V	5.0 V	160 Ω	120 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X3491

Figure 3: AC Load Circuit

XC73108 Pinouts

PQ160	PG144 BG225	PQ100	PC84	Input	XC73108	Output
1	D3	–	–		V_{CCIO}	
2	C2	3	75	O/CKEN1		MC5-4
3	–	–	–		N/C	
4	B1	4	–	FO		MC2-1
5	–	–	–		N/C	
6	D2	5	76	O/FOE0		MC5-5
7	E3	–	–	O		MC5-1
8	C1	6	77	O/FOE1		MC5-6
9	E2	–	–	O		MC5-2
10	D1	7	78		V_{CCINT}/V_{PP}	
11	F3	8	79	I/FI		
12	F2	9	–	I/O/FI		MC5-7
13	E1	10	80	I/FI		
14	G2	–	–	I/O/FI		MC5-8
15	G3	11	81	I/FI		
16	F1	–	–	I/O/FI		MC5-9
17	G1	12	82	I/FI		
18	H2	13	83	I/FI		
19	H1	14	84	I/FI		
20	H3	–	–		**GND**	
21	J3	15	1		MR	
22	J1	16	2	I/FI		
23	K1	17	3	I/FI		
24	J2	18	4	I/FI		
25	K2	–	–	O		MC10-1
26	K3	19	5	I/FI		
27	L1	–	-	O		MC10-2
28	L2	20	6	I/FI		
29	M1	21	–	I/O/FI		MC10-8
30	N1	22	7	I/FI		
31	M2	23	8		**GND**	
32	L3	–	–	I/O/FI		MC10-7
33	N2	24	9	O/FCLK0		MC10-3
34	P1	–	–	O		MC10-6
35	M3	25	10	O/FCLK1		MC10-4
36	N3	26	–	FO		MC1-1
37	P2	27	11	I/O/FI		MC10-9
38	–	–	–		N/C	
39	–	–	–		N/C	
40	R1	–	–		**GND**	

PQ160	PG144 BG225	PQ100	PC84	Input	XC73108	Output
41	N4	28	–		V_{CCIO}	
42	P3	29	12	O/FCLK2		MC10-5
43	R2	–	–	I/O		MC4-1
44	P4	30	13	FO		MC1-2
45	N5	–	–	I/O		MC4-2
46	R3	–	–		V_{CCINT}	
47	P5	31	14	FO		MC1-3
48	R4	–	–	I/O		MC4-3
49	N6	32	15	FO		MC1-4
50	P6	–	–	I/O		MC4-4
51	R5	33	16		**GND**	
52	–	–	–		N/C	
53	–	–	–		N/C	
54	P7	34	17	FO		MC1-5
55	N7	–	–	I/O		MC4-5
56	R6	35	18	FO		MC1-6
57	R7	36	–	I/O/FI		MC4-7
58	P8	37	19	FO		MC1-7
59	R8	38	20	FO		MC1-8
60	N8	39	21	FO		MC1-9
61	N9	40	22		V_{CCIO}	
62	R9	41	23	I/O		MC9-1
63	R10	42	24	I/O		MC9-2
64	P9	43	25	I/O		MC9-3
65	–	–	–		N/C	
66	–	–	–		N/C	
67	P10	–	–	I/O		MC4-6
68	N10	44	26	I/O		MC9-6
69	R11	45	–	I/O/FI		MC4-8
70	P11	46	27		**GND**	
71	R12	47	28	I/O/FI		MC9-7
72	R13	48	–	I/O/FI		MC4-9
73	P12	49	29	I/O/FI		MC9-8
74	N11		–	I/O		MC3-1
75	P13	50	30	I/O/FI		MC9-9
76	R14	–	–	I/O		MC3-2
77	N12	51	31	I/O		MC8-1
78	N13	–	–	I/O		MC3-3
79	P14	52	32	I/O		MC8-2
80	R15	–	–		**GND**	

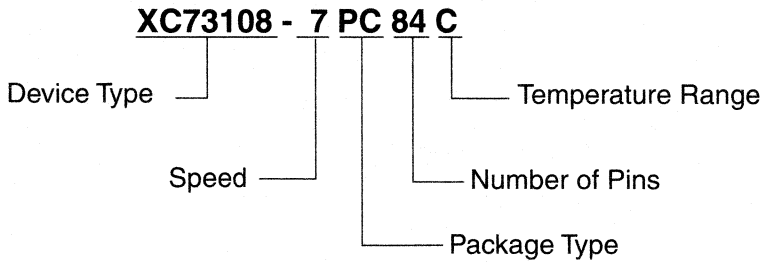
Note: With the XC73108 in the 225-pin ball grid array package, only 144 of the solder balls are connected, the remaining solder balls should be left unconnected.

XC73108 Pinouts (continued)

PQ160	PG144 BG225	PQ100	PC84	Input	XC73108	Output
81	M13	53	–		V_{CCIO}	
82	N14	54	33	I/O		MC8-3
83	–	–	–		N/C	
84	P15	–	–	I/O		MC3-4
85	–	–	–		N/C	
86	M14	55	34	I/O		MC9-4
87	L13	–	–	I/O		MC3-5
88	N15	56	35	I/O		MC9-5
89	L14	–	–	I/O		MC3-6
90	M15	57	36	I/O		MC8-4
91	K13	–	–	I/O/FI		MC3-7
92	K14	58	37	I/O		MC8-5
93	L15	–	–	I/O/FI		MC3-8
94	J14	59	38		V_{CCINT}	
95	J13	60	39	I/O		MC8-6
96	K15	61	–	I/O/FI		MC3-9
97	J15	62	40	I/O/FI		MC8-7
98	H14	63	41	I/O/FI		MC8-8
99	H15	–	–		**GND**	
100	H13	64	42		**GND**	
101	G13	65	43	I/O/FI		MC8-9
102	G15	66	44	I/O		MC7-1
103	F15	67	45	I/O		MC7-2
104	G14	68	46	I/O		MC7-3
105	F14	–	–	I/O		MC12-1
106	F13	69	47	I/O		MC7-4
107	E15	–	–	I/O		MC12-2
108	E14	70	48	I/O		MC7-5
109	D15	–	–	I/O		MC12-3
110	C15	71	49		**GND**	
111	D14	72	50	I/O		MC7-6
112	E13	–	–	I/O		MC12-4
113	C14	73	51	I/O		MC6-5
114	B15	–	–	I/O		MC12-5
115	D13	74	52	I/O		MC6-4
116	C13	75	–	I/O/FI		MC12-9
117	B14	76	53	I/O/FI		MC7-7
118	–	–	–		N/C	
119	–	–	–		N/C	
120	A15	77	–		**GND**	

PQ160	PG144 BG225	PQ100	PC84	Input	XC73108	Output
121	C12	–	–		V_{CCIO}	
122	B13	78	54	I/O/FI		MC7-8
123	A14	–	–	I/O		MC12-6
124	B12	79	55	I/O/FI		MC7-9
125	C11	–	–	I/O/FI		MC12-7
126	A13	80	56	I/O		MC6-1
127	B11	–	–		**GND**	
128	A12	–	–	I/O/FI		MC12-8
129	C10	81	–	I/O/FI		MC11-7
130	B10	–	–	I/O		MC11-1
131	–	–	–		N/C	
132	–	–	–		N/C	
133	A11	82	–	I/O/FI		MC11-8
134	B9	83	57	I/O		MC6-2
135	C9	84	58	I/O		MC6-6
136	A10	85	59	I/O		MC6-3
137	A9	86	60		**GND**	
138	B8	87	61	I/O/FI		MC6-7
139	A8	88	62	I/O/FI		MC6-8
140	C8	89	63	I/O/FI		MC6-9
141	C7	90	64		V_{CCIO}	
142	A7	91	65	FO		MC2-9
143	A6	92	66	FO		MC2-8
144	B7	93	67	FO		MC2-7
145	B6	94	–	I/O/FI		MC11-9
146	C6	95	68	FO		MC2-6
147	A5	–	–	I/O		MC11-2
148	B5	96	69	FO		MC2-5
149	–	–	–		N/C	
150	–	–	–		N/C	
151	A4	–	–	I/O		MC11-3
152	A3	97	70	FO		MC2-4
153	B4	–	–	I/O		MC11-4
154	C5	98	71	FO		MC2-3
155	B3	–	–	I/O		MC11-5
156	A2	99	72	FO		MC2-2
157	C4	100	73		V_{CCINT}	
158	C3	–	–	I/O		MC11-6
159	B2	1	74	O/CKEN0		MC5-3
160	A1	2	–		**GND**	

Ordering Information



Speed Options

-20	20 ns pin-to-pin delay
-15	15 ns pin-to-pin delay
-12	12 ns pin-to-pin delay
-10	10 ns pin-to-pin delay (commercial and industrial only)
-7	7.5 ns pin-to-pin delay (commercial only)

Packaging Options

PC84	84-Pin Plastic Leaded Chip Carrier
WC84	84-Pin Windowed Ceramic Leaded Chip Carrier
PQ100	100-Pin Plastic Quad Flat Pack
PG144	144-Pin Windowed Pin-Grid-Array
PQ160	160-Pin Plastic Quad Flat Pack
BG225	225-Pin Plastic Ball-Grid-Array

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	-40°C to 85°C
M	Military	-55°C (Ambient) to 125°C (Case)

Component Availability

Pins		84		100	144	160	225
Type		Plastic PLCC	Ceramic CLCC	Plastic PQFP	Ceramic PGA	Plastic PQFP	Plastic BGA
Code		PC84	WC84	PQ100	PG144	PQ160	BG225
XC73108	–20	CI	CI	CI	CIM	CI	CI
	–15	CI	CI	CI	CIM	CI	CI
	–12	CI	CI	CI	CI	CI	CI
	–10	C	C	C	C	C	C
	–7	C	C	C	C	C	C

C = Commercial = 0° to +70°C I = Industrial = –40° to 85°C M = Military = –55°C(A) to 125°C (C)

XC73144
144-Macrocell CMOS CPLD

January, 1997 (Version 1.0) | Product Specification

Features

- High-performance Complex Programmable Logic Devices (CPLDs)
  - 7.5 ns pin-to-pin speeds on all fast inputs
  - Up to 100 MHz maximum clock frequency
- 100% PCI compliant
- 18 outputs with 24 mA drive
- I/O operation at 3.3 V or 5 V
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- 100% interconnect matrix
  - Maximizes resource utilization
  - Wire-AND capability via SMARTswitch
- High-speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 43 MHz 16-bit accumulators
- Multiple independent clocks
- Up to 132 inputs programmable as direct, latched, or registered
- Power management options
- Multiple security bits for design protection
- 144 macrocells with programmable I/O architecture
- Advanced Dual-Block architecture
  - 4 Fast Function Blocks
  - 12 High-Density Function Blocks
- Programmable slew rate
- Programmable ground control
- 0.8 µ CMOS EPROM technology
- Available in 160-pin PQFP and 225-pin BGA packages

General Description

The XC73144 is a high performance CPLD providing general purpose logic integration. It consists of four PAL-like 24V9 Fast Function Blocks and twelve High Density Function Blocks interconnected by the 100%-populated Universal Interconnect Matrix (UIM™).

Power Management

The XC73144 features a power-management scheme that permits non-speed-critical paths of a design to be operated at reduced power. Overall power dissipation is often reduced significantly, since, in most systems only a few paths are speed critical.

Macrocells can individually be specified for high performance or low power operation by adding attributes to the logic schematic, or declaration statements to the behavioral description. To minimize power dissipation, unused Function Blocks are turned off and unused macrocells in used Function Blocks are configured for low power operation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

$$I_{CC}\ (mA) = MC_{HP}\ (2.4) + MC_{LP}\ (2.1) + MC\ (0.015\ mA/MHz)\ f$$

Where:

MC_{HP} = Macrocells in high-performance mode
MC_{LP} = Macrocells in low-power mode
MC = Total number of macrocells used
f = Clock frequency (MHz)

Figure 1 shows a typical power calculation for the XC73144 device, programmed as eight 16-bit counters and operating at the indicated clock frequency.

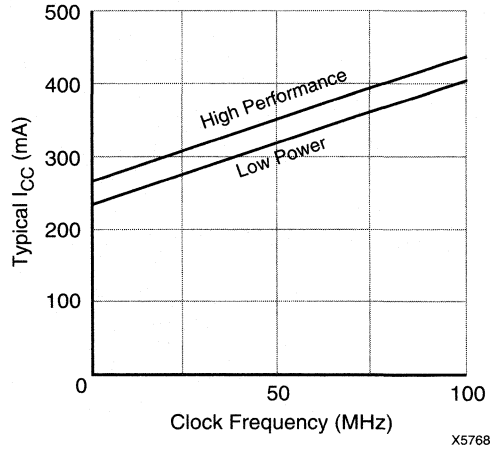


Figure 1: Typical I_{CC} vs. Frequency for XC73144

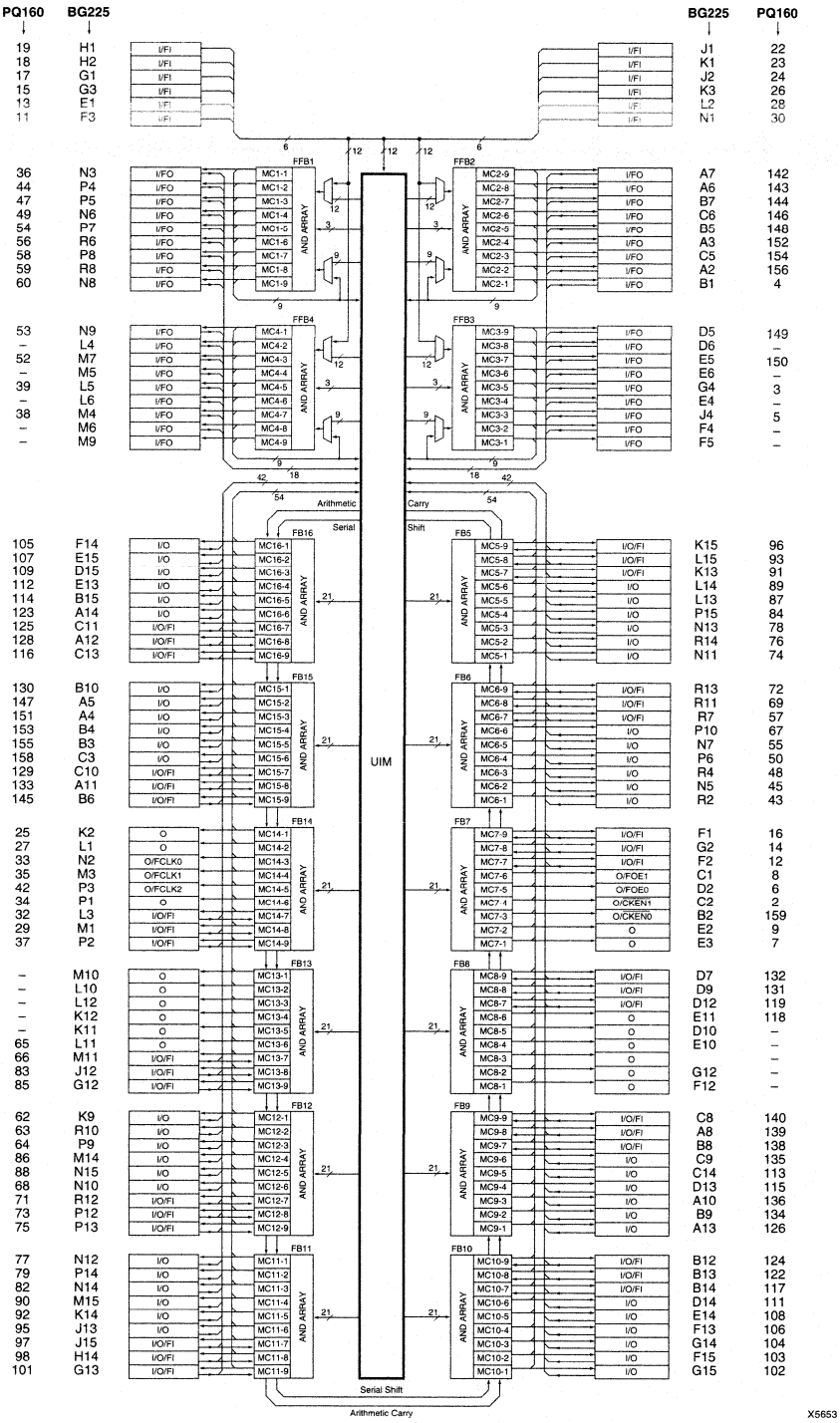


Figure 2: XC73144 Architecture

Absolute Maximum Ratings

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

Warning: 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.

Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units
V_{CCINT} V_{CCIO}	Supply voltage relative to GND Commercial T_A = 0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial T_A = –40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND Military T_A = –55°C to T_C + 125°C	4.5	5.5	V
V_{CCIO}	I/O supply voltage relative to GND	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{IH}	High-level input voltage	2.0	V_{CC} +0.5	V
V_O	Output voltage	0	V_{CCIO}	V
T_{IN}	Input signal transition time		50	ns

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V TTL Low-level output voltage	I_{OL} = 24 mA V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins	V_{IN} = GND f = 1.0 MHz		8.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FCLK2, FOE0, FOE1)	V_{IN} = GND f = 1.0 MHz		12.0	pF
C_{OUT}[1]	Output capacitance	V_{IN} = GND f = 1.0 MHz		10.0	pF
I_{CC}[2]	Supply current (low power mode)	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5V f = 1.0 MHz @ 25°C	250 Typ		mA

Notes: 1. Sample tested.
2. Measured with device programmed as eight 16-bit counters.

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time		80	160	µs

Slew Rate and Programmable Ground Control

Due to the large number of high current drivers available on the XC73144, two programmable signal management features have been included – slew rate control (SRC) and ground control (GC). Slew rate control is primarily for external system benefit, to reduce ringing and other coupling phenomenon. SRC permits designers to select either 1 V/ns or 1.5 V/ns slew rate on a pin-by-pin basis for any output or I/O signal. This can be done with PLUSASM or schematically, as needed. The default slew rate is 1 V/ns. To assign the pins with equations (PLUSASM), the designer needs to only declare them as follows:

FAST ON <signal name list>

This will assign the signals in the list to have a 1.5 V/ns slew rate. Omitting the signal name list will globally set all signals to be 1.5 V/ns. Specific signals therefore can be declared with 1 V/ns slew rate as follows:

FAST OFF <signal name list>

Schematic control of SRC is also straightforward. Again, the default is 1 V/ns, but to assign specific pins fast, the designer need only attach the "FAST" attribute to the I/O or output buffer or the corresponding pin.

Programmable ground control is useful for internal chip signal management. The output buffers of the Fast Function Blocks have an impedance of approximately 7 Ω when switching high to low, where the High Density Function Blocks impedance is around 14 Ω. Since this low impedance is negligible compared to the impedance of the pin inductance when output current transients occur, a reasonable ground connection can be made by driving unused output pins low and physically attaching them to external ground. The XC73144 architecture permits the automatic assignment of external ground signals to all macrocells that are not declared as primary outputs or I/Os. Note that the logical function of the buried macrocell is fully preserved, while its output driver is driving low and physically attached to ground. Should designers not wish to employ programmable ground control, they need only declare all such pins as primary I/Os whether they will be attached externally or not.

Fast Function Block (FFB) External AC Characteristics[3]

Symbol	Parameter	XC73144-7 (Com Only)		XC73144-10 (Com Only)		XC73144-12 (Com/Ind Only)		XC73144-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{CF}	Max count frequency [1, 2, 4]	105.0		100.0		80.0		66.7		MHz
t_{SUF}	Fast input setup time before FCLK ↑ [1]	4.0		5.0		6.0		7.0		ns
t_{HF}	Fast input hold time after FCLK ↑	0		0		0		0		ns
t_{COF}	FCLK ↑ to output valid		5.5		7.0		9.0		12.0	ns
t_{PDFO}	Fast input to output valid [1, 2]		7.5		9.0		12.0		15.0	ns
t_{PDFU}	I/O to output valid [1, 2]		13.5		17.0		22.0		27.0	ns
t_{CWF}	Fast clock pulse width (High or Low)	4.0		5.0		5.5		6.0		ns

Notes: 1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.
3. All appropriate AC specifications tested using Figure 3 as the test load circuit.
4. Export Control Max. flip-flop toggle rate.

High-Density Function Block (FB) External AC Characteristics

Symbol	Parameter	XC73144-7 (Com Only)		XC73144-10 (Com Only)		XC73144-12 (Com/Ind Only)		XC73144-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_C	Max count frequency [1, 2]	83.3		62.5		55.6		45.5		MHz
t_{SU}	I/O setup time before FCLK ↑ [1, 2]	12.0		13.5		18.0		22.0		ns
t_H	I/O hold time after FCLK ↑	0		0		0		0		ns
t_{CO}	FCLK ↑ to output valid		7.0		9.0		12.0		15.0	ns
t_{PSU}	I/O setup time before p-term clock ↑ [2]	4.0		6.0		7.0		9.0		ns
t_{PH}	I/O hold time after p-term clock ↑	0		0		0		0		ns
t_{PCO}	P-term clock ↑ to output valid		15.0		19.0		23.0		28.0	ns
t_{PD}	I/O to output valid [1, 2]		18.0		22.0		30.0		36.0	ns
t_{CW}	Fast clock pulse width	4.0		5.0		5.5		6.0		ns
t_{PCW}	P-term clock pulse width	5.0		6.0		7.5		8.5		ns

Notes: 1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

Fast Function Block (FFB) Internal AC Characteristics

Symbol	Parameter	XC73144-7 (Com Only)		XC73144-10 (Com Only)		XC73144-12 (Com/Ind Only)		XC73144-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay [1]		1.5		1.5		2.0		2.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay [1]		3.5		5.5		7.0		8.0	ns
t_{FSUI}	FFB register setup time	1.5		2.5		3.0		4.0		ns
t_{FHI}	FFB register hold time	2.5		2.5		3.0		3.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		0.5		1.0		1.0	ns
t_{FAOI}	FFB register async. set delay		2.0		2.5		3.0		4.0	ns
t_{PTXI}	FFB p-term assignment delay		0.8		1.0		1.2		1.5	ns
t_{FFD}	FFB feedback delay		4.0		5.0		6.5		8.0	ns

Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

High-Density Function Block (FB) Internal AC Characteristics

Symbol	Parameter	XC73144-7 (Com Only)		XC73144-10 (Com Only)		XC73144-12 (Com/Ind Only)		XC73144-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{LOGI}	FB logic array delay [1]		3.5		3.5		4.0		5.0	ns
t_{LOGILP}	Low power FB logic delay [1]		7.0		7.5		9.0		11.0	ns
t_{SUI}	FB register setup time	1.5		2.5		3.0		4.0		ns
t_{HI}	FB register hold time	3.5		3.5		4.0		5.0		ns
t_{COI}	FB register clock-to-output delay		1.0		1.0		1.0		1.0	ns
t_{PDI}	FB register pass through delay		1.5		2.5		4.0		4.0	ns
t_{AOI}	FB register async. set/reset delay		2.5		3.0		4.0		5.0	ns
t_{RA}	Set/reset recovery time before FCLK ↑	15.0		19.0		21.0		25.0		ns
t_{HA}	Set/reset hold time after FCLK ↑	0		0		0		0		ns
t_{PRA}	Set/reset recovery time before p-term clock ↑	7.5		10.0		12.0		15.0		ns
t_{PHA}	Set/reset hold time after p-term clock ↑	5.0		6.0		8.0		9.0		ns
t_{PCI}	FB p-term clock delay		1.0		0		0		0	ns
t_{OEI}	FB p-term output enable delay		3.0		4.0		5.0		7.0	ns
t_{CARY8}	ALU carry delay within 1 FB [2]		5.0		6.0		8.0		12.0	ns
t_{CARYFB}	Carry lookahead delay per additional Functional Block [2]		1.0		1.5		2.0		3.0	ns

Notes: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.
2. Arithmetic carry delays are measured as the increase in required set-up time to adjacent macrocell(s) for adder with registered outputs.

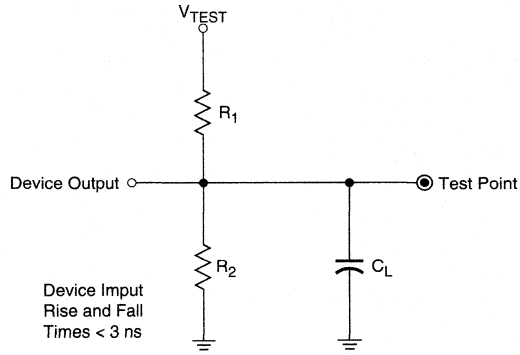
I/O Block External AC Characteristics

Symbol	Parameter	XC73144-7 (Com Only)		XC73144-10 (Com Only)		XC73144-12 (Com/Ind Only)		XC73144-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{IN}	Max pipeline frequency (input register to FFB or FB register) [1]	83.3		62.5		55.6		45.5		MHz
t_{SUIN}	Input register/latch setup time before FCLK ↑	4.0		5.0		6.0		7.0		ns
t_{HIN}	Input register/latch hold time after FCLK ↑	0		0		0		0		ns
t_{COIN}	FCLK ↑ to input register/latch output		2.5		3.5		4.0		5.0	ns
t_{CESUIN}	Clock enable setup time before FCLK ↑	5.0		7.0		8.0		10.0		ns
t_{CEHIN}	Clock enable hold time after FCLK ↑	0		0		0		0		ns
t_{CWHIN}	FCLK pulse width high time	4.0		5.0		5.5		6.0		ns
t_{CWLIN}	FCLK pulse width low time	4.0		5.0		5.5		6.0		ns

Note: 1. Specifications account for logic paths that use the maximum number of available product terms for a given macrocell.

Internal AC Characteristics

Symbol	Parameter	XC73144-7 (Com Only)		XC73144-10 (Com Only)		XC73144-12 (Com/Ind Only)		XC73144-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		2.5		3.5		4.0		5.0	ns
t_{FOUT}	FFB output buffer and pad delay		3.0		4.5		5.0		7.0	ns
t_{OUT}	FB output buffer and pad delay		4.5		6.5		8.0		10.0	ns
t_{UIM}	Universal Interconnect Matrix delay		6.0		9.0		10.0		12.0	ns
t_{FOE}	FOE input to output valid		7.5		10.0		12.0		15.0	ns
t_{FOD}	FOE input to output disable		7.5		10.0		12.0		15.0	ns
t_{FCLKI}	Fast clock buffer delay		1.5		2.5		3.0		4.0	ns



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
FO	5.0 V	5.0 V	160 Ω	120 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X3491

Figure 3: AC Load Circuit

XC73144 Pinouts

BG225	PQ160	Input	XC73144	Output
D3	1		V_{CCIO}	
E4	-	I/FO		MC3-4
F4	-	I/FO		MC3-2
C2	2	O/CKEN1		MC7-4
F5	-	I/FO		MC3-1
G4	3	I/FO		MC3-5
B1	4	I/FO		MC2-1
J4	5	I/FO		MC3-3
D2	6	O/FOE0		MC7-5
E3	7	O		MC7-1
C1	8	O/FOE1		MC7-6
E2	9	O		MC7-2
D1	10		V_{CCINT}/V_{PP}	
F3	11	I/FI		
F2	12	I/O/FI		MC7-7
E1	13	I/FI		
G2	14	I/O/FI		MC7-8
G3	15	I/FI		
F1	16	I/O/FI		MC7-9
G1	17	I/FI		
H2	18	I/FI		
H1	19	I/FI		
H3	20		**GND**	
J3	21	I/FI	**MR**	
K5	-		V_{CCIO}	
J1	22	I/FI		
K1	23	I/FI		
J2	24	I/FI		
K2	25	O		MC14-1
K3	26	I/FI		
L1	27	O		MC14-2
L2	28	I/FI		
M1	29	I/O/FI		MC14-8
N1	30	I/FI		
M2	31		**GND**	
L3	32	I/O/FI		MC14-7
N2	33	O/FCLK0		MC14-3
P1	34	O		MC14-6
M3	35	O/FCLK1		MC14-4
N3	36	I/FO		MC1-1
K4	-	I/FO		MC4-1
L4	-	I/FO		MC4-2
P2	37	I/O/FI		MC14-9
M4	38	I/FO		MC4-3
L5	39	I/FO		MC4-5
R1	40		**GND**	

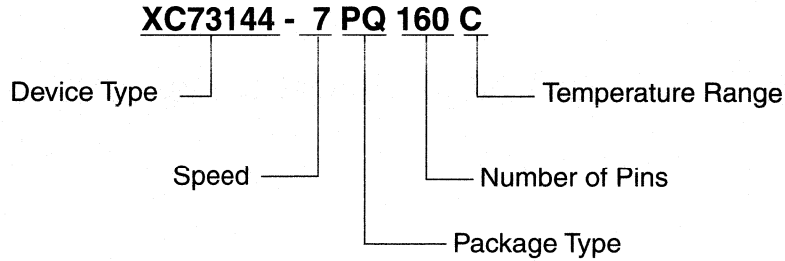
BG225	PQ160	Input	XC73144	Output
N4	41		V_{CCIO}	
P3	42	O/FCLK2		MC14-5
R2	43	I/O		MC6-1
P4	44	I/FO		MC1-2
N5	45	I/O		MC6-2
R3	46		V_{CCINT}	
M5	-	I/FO		MC4-4
P5	47	I/FO		MC1-3
R4	48	I/O		MC6-3
L6	-	I/FO		MC4-6
M6	-	I/FO		MC4-8
N6	49	I/FO		MC1-4
P6	50	I/O		MC6-4
R5	51		**GND**	
M7	52	I/FO		MC4-7
M9	53	I/FO		MC4-9
P7	54	I/FO		MC1-5
N7	55	I/O		MC6-5
R6	56	I/FO		MC1-6
R7	57	I/O/FI		MC6-7
P8	58	I/FO		MC1-7
R8	59	I/FO		MC1-8
N8	60	I/FO		MC1-9
N9	61		V_{CCIO}	
M10	-	O		MC13-1
L10	-	O		MC13-2
R9	62	I/O		MC12-1
R10	63	I/O		MC12-2
P9	64	I/O		MC12-3
L11	65	O		MC13-6
M11	66	I/O/FI		MC13-7
M12	-		**GND**	
P10	67	I/O		MC6-6
N10	68	I/O		MC12-6
R11	69	I/O/FI		MC6-8
P11	70		**GND**	
R12	71	I/O/FI		MC12-7
R13	72	I/O/FI		MC6-9
P12	73	I/O/FI		MC12-8
N11	74	I/O		MC5-1
P13	75	I/O/FI		MC12-9
R14	76	I/O		MC5-2
N12	77	I/O		MC11-1
N13	78	I/O		MC5-3
P14	79	I/O		MC11-2
R15	80		**GND**	

XC73144 Pinouts (continued)

BG225	PQ160	Input	XC73144	Output
M13	81		V_{CCIO}	
L12	-	O		MC13-3
K12	-	O		MC13-4
N14	82	I/O		MC11-3
K11	-	O		MC13-5
J12	83	I/O/FI		MC13-8
P15	84	I/O		MC5-4
G12	85	I/O/FI		MC13-9
M14	86	I/O		MC12-4
L13	87	I/O		MC5-5
N15	88	I/O		MC12-5
L14	89	I/O		MC5-6
M15	90	I/O		MC11-4
K13	91	I/O/FI		MC5-7
K14	92	I/O		MC11-5
L15	93	I/O/FI		MC5-8
J14	94		V_{CCINT}	
J13	95	I/O		MC11-6
K15	96	I/O/FI		MC5-9
J15	97	I/O/FI		MC11-7
H14	98	I/O/FI		MC11-8
H15	99		**GND**	
H13	100		**GND**	
F11	-		V_{CCINT}	
G13	101	I/O		MC11-9
G15	102	I/O		MC10-1
F15	103	I/O		MC10-2
G14	104	I/O		MC10-3
F14	105	I/O		MC16-1
F13	106	I/O		MC10-4
E15	107	I/O		MC16-2
E14	108	I/O		MC10-5
D15	109	I/O		MC16-3
C15	110		**GND**	
D14	111	I/O		MC10-6
E13	112	I/O		MC16-4
C14	113	I/O		MC9-5
B15	114	I/O		MC16-5
D13	115	I/O		MC9-4
C13	116	I/OFI		MC16-9
F12	-	O		MC8-1
E12	-	O		MC8-2
B14	117	I/O/FI		MC10-7
E11	118	O		MC8-6
D12	119	I/O/FI		MC8-7
A15	120		**GND**	

BG225	PQ160	Input	XC73144	Output
C12	121		V_{CCIO}	
B13	122	I/O/FI		MC10-8
A14	123	I/O		MC16-6
B12	124	I/O/FI		MC10-9
C11	125	I/O/FI		MC16-7
A13	126	I/O		MC9-1
D11	-	O		MC8-3
B11	127		**GND**	
A12	128	I/O/FI		MC16-8
E10	-	O		MC8-4
D10	-	O		MC8-5
C10	129	I/O/FI		MC15-7
B10	130	I/O		MC15-1
D9	131	I/O/FI		MC8-8
D7	132	I/O/FI		MC8-9
A11	133	I/O/FI		MC15-8
B9	134	I/O		MC9-2
C9	135	I/O		MC9-6
A10	136	I/O		MC9-3
A9	137		**GND**	
B8	138	I/O/FI		MC9-7
A8	139	I/O/FI		MC9-8
C8	140	I/O/FI		MC9-9
C7	141		V_{CCIO}	
A7	142	I/FO		MC2-9
A6	143	I/FO		MC2-8
B7	144	I/FO		MC2-7
B6	145	I/O/FI		MC15-9
C6	146	I/FO		MC2-6
D6	-	I/FO		MC3-8
E6	-	I/FO		MC3-6
A5	147	I/O		MC15-2
B5	148	I/FO		MC2-5
D5	149	I/FO		MC3-9
E5	150	I/FO		MC3-7
A4	151	I/O		MC15-3
A3	152	I/FO		MC2-4
B4	153	I/O		MC15-4
C5	154	I/FO		MC2-3
D4	-		**GND**	
B3	155	I/O		MC15-5
A2	156	I/FO		MC2-2
C4	157		V_{CCINT}	
C3	158	I/O		MC15-6
B2	159	O/CKEN0		MC7-3
A1	160		**GND**	

Ordering Information



Speed Options

-15	15 ns pin-to-pin delay
-12	12 ns pin-to-pin delay
-10	10 ns pin-to-pin delay (commercial and industrial only)
-7	7.5 ns pin-to-pin delay (commercial only)

Packaging Options

PQ160	160-Pin Plastic Quad Flat Pack
BG225	225-Pin Plastic Ball-Grid-Array

Temperature Options

C	Commercial 0°C to 70°C
I	Industrial -40°C to 85°C

Component Availability

Pins		160	225
Type		Plastic PQFP	Plastic BGA
Code		PQ160	BG225
XC73144	–15	CI	CI
	–12	CI	CI
	–10	C	C
	–7	C	C

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

XILINX®

Device Packaging

1 ISP and JTAG Support

2 Application Notes

3 XC9500 Data Sheets

4 XC7300 Data Sheets

5 Device Packaging

6 Quality Assurance

7 Technical Support

8 Sales Offices, Representatives, Distributors

Visit us on the WEB at:

www.xilinx.com

for the latest packaging information.

Number of Available I/O Pins

Device	Max. I/O	44	68	84	100	144	160	208	225	352	432
XC7236A	36	36									
XC7272A	72		56	72							
XC7318	38	38									
XC7336	38	38									
XC7336Q	38	38									
XC7354	58	38	58								
XC7372	84		57	72	84						
XC73108	120			72	84	120	120		120		
XC73144	156						136		156		
XC9536	34	34									
XC9572	72			69	72						
XC95108	108			69	81		108				
XC95144	133				81		133				
XC95180	168						133	166			
XC95216	168						133	166		166	
XC95288	192							168		192	
XC95432	232										232
XC95576	232										232

5

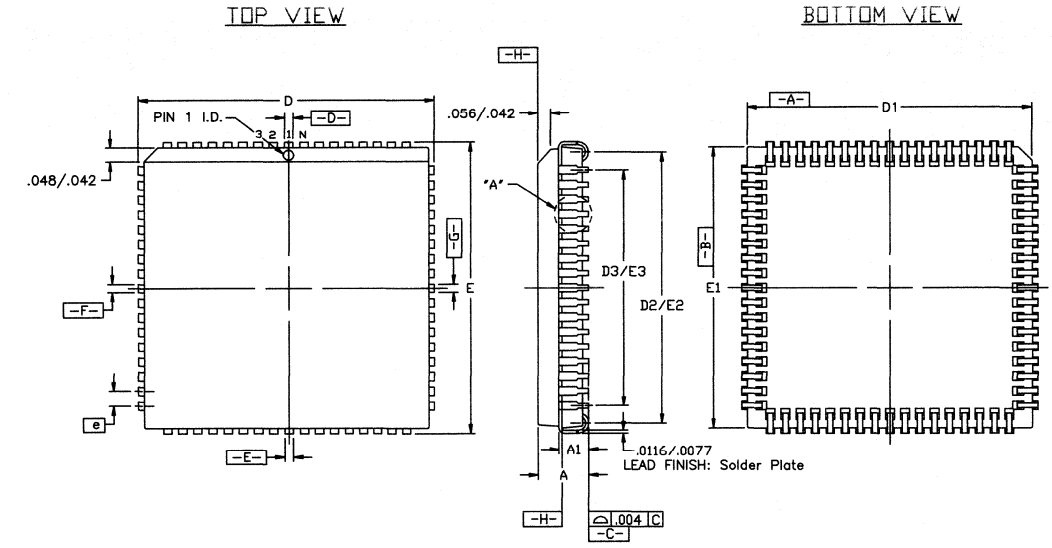
Package Options

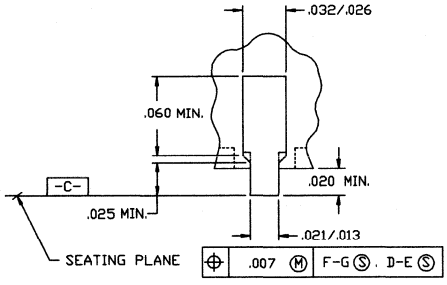
	Surface Mount						Through-hole
	PLCC	**PQFP**	**HQFP**	**TQFP**	**VQFP**	**BGA**	**PGA**
Standard	JEDEC	EIAJ	EIAJ	EIAJ	EIAJ	JEDEC	JEDEC
Lead Pitch	50 mil	0.65/0.5 mm	0.65/0.5 mm	0.5 mm	0.5 mm	1.5 mm	100 mil
Body	Plastic	Plastic	Plastic/ Metal	Plastic	Plastic	FR4	Ceramic/ Plastic
Temperature Options	C, I	C, I	C, I	C, I	C, I	C	C, I, M, B
Ordering Code	PC	PQ	HQ	TQ	VQ	BG	PG
XC7236A	44						
XC7272A	68, 84						84
XC7318	44	44					
XC7336	44	44					
XC7336Q	44	44			44		
XC7354	44, 68						
XC7372	68, 84	100					
XC73108	84	100,160				225	144
XC73144		160				225	
XC9536	44				44		
XC9572	84	100		100			
XC95108	84	100, 160		100			
XC95144		100, 160					
XC95180		160	208				
XC95216		160	208				
XC95288			208			352	
XC95432						352	
XC95576						352	

Physical Dimensions

PLCC Packages — PC20, PC28, PC44, PC68, PC84 5-4
PQFP Packages — PQ44, PQ100, PQ160, HQ208 5-5
TQFP Packages — TQ100 5-9
VQFP Packages — VQ44 5-10
BGA Packages — BG225, BG352 5-11
Ceramic PGA Packages — PG84, PG144 5-13
Windowed CLCC Packages — WC44, WC68, WC84 5-15

PLCC Packages — PC20, PC28, PC44, PC68, PC84





DETAIL "A"

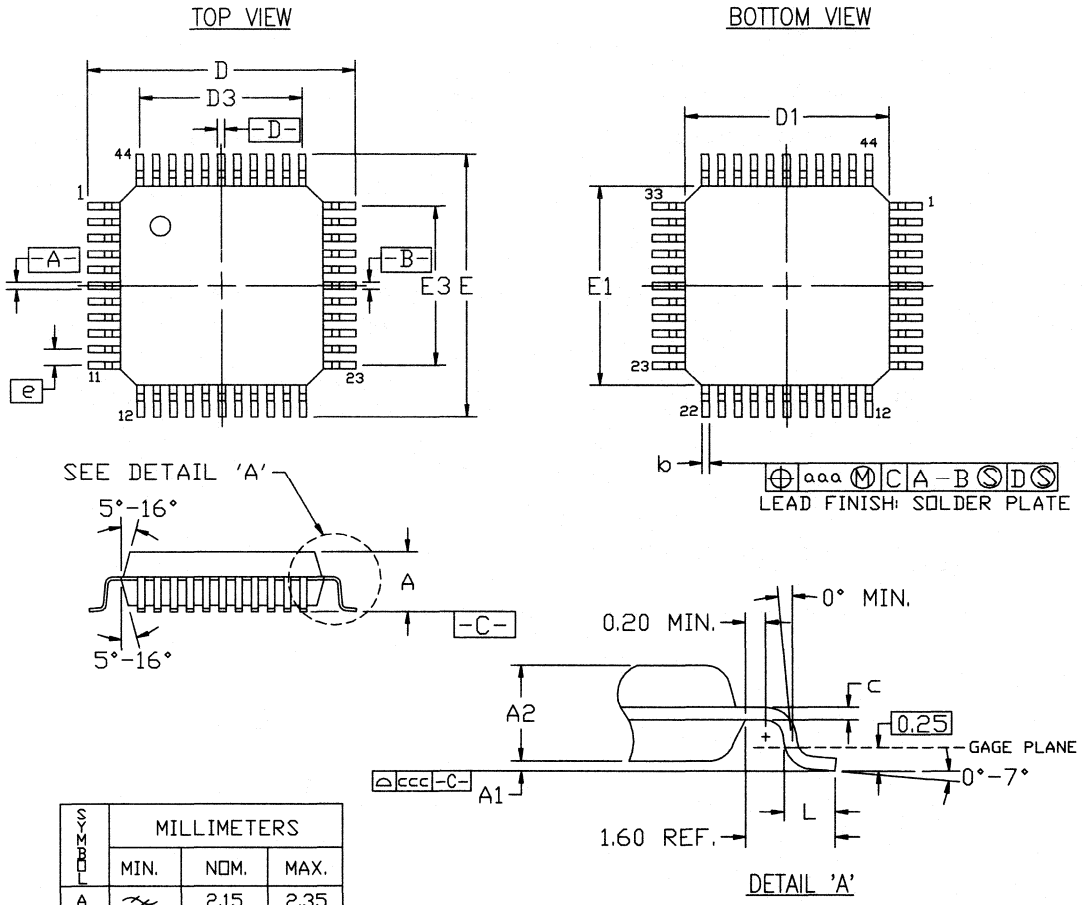
SYMBOL	INCHES									
	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.
A	.165	.180	.165	.180	.165	.180	.165	.200	.165	.200
A_1	.090	.120	.099	.110	.090	.120	.090	.130	.090	.130
D/E	.385	.395	.485	.495	.685	.695	.985	.995	1.185	1.195
D_1/E_1	.350	.356	.450	.456	.650	.656	.950	.958	1.150	1.158
D_2/E_2	.290	.330	.390	.430	.590	.630	.890	.930	1.090	1.130
D_3/E_3	.200 REF.		.300 REF.		.500 REF.		.800 REF.		1.000 REF.	
e	.050 BSC		.050 BSC		.050 BSC		.050 BSC		.050 BSC	
N	20		28		44		68		84	

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982.
2. DIMENSIONS "D1" AND "E1" DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED .010 PER SIDE.
3. "N" IS NUMBER OF TERMINALS.
4. CONFORM TO JEDEC MO-047
5. TOP OF PACKAGE MAY BE SMALLER THAN BOTTOM BY .010".

20, 28, 44, 68 and 84–PIN PLCC (PC20 THRU PC84)

PQFP Packages — PQ44, PQ100, PQ160, HQ208

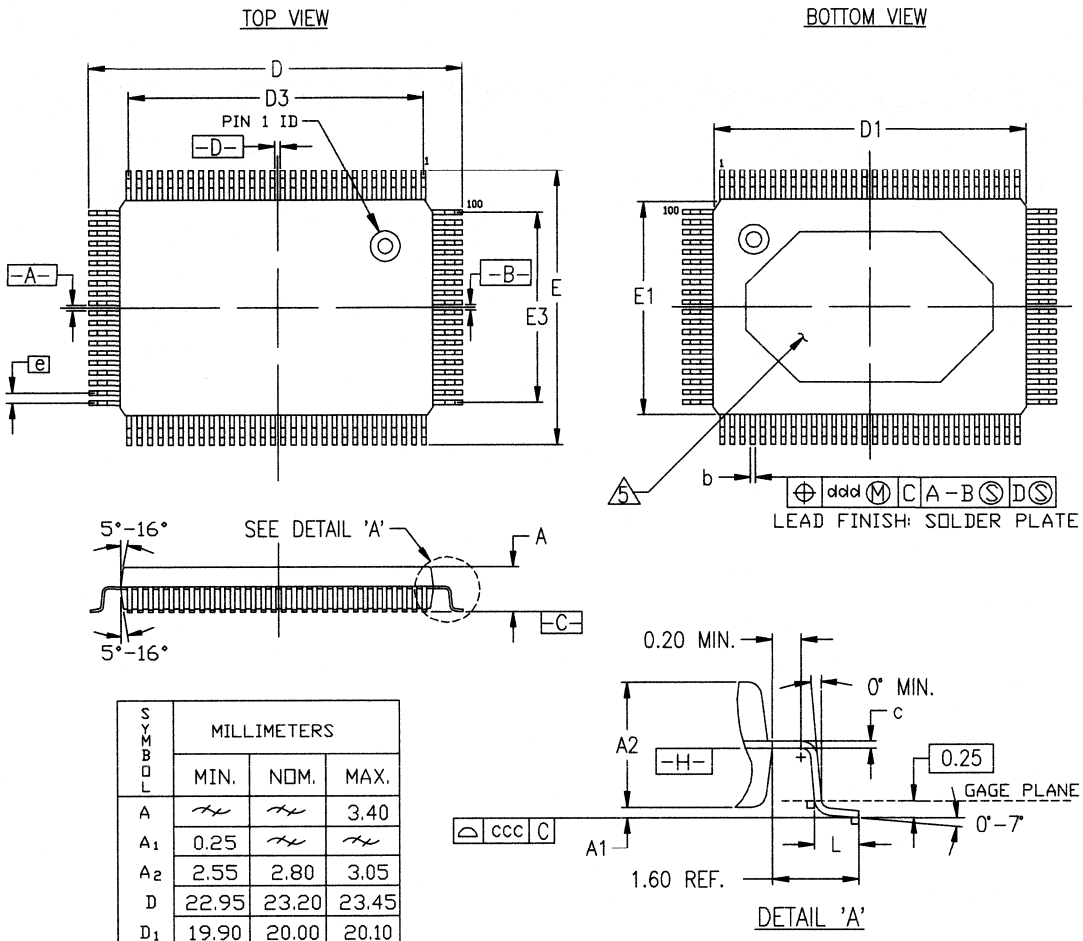


SYMBOL	MILLIMETERS		
	MIN.	NOM.	MAX.
A		2.15	2.35
A_1	0.05		0.25
A_2	1.95	2.00	2.10
D/E	12.95	13.20	13.45
D_1/E_1	9.90	10.00	10.10
D_3/E_3		8.00 BSC	
L	0.73	0.88	1.03
e		0.80 BSC.	
b	0.30		0.45
c	0.13		0.23
aaa		0.20	
ccc			0.10

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982.
2. DIMENSIONS "D1" AND "E1" DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION SHALL NOT EXCEED 0.25mm PER SIDE.
3. THE TOP OF PACKAGE MAY BE SMALLER THAN THE BOTTOM OF PACKAGE BY 0.15mm.
4. CONFORMS TO JEDEC MO-108-AA2

44-PIN PLASTIC PQFP (PQ44)

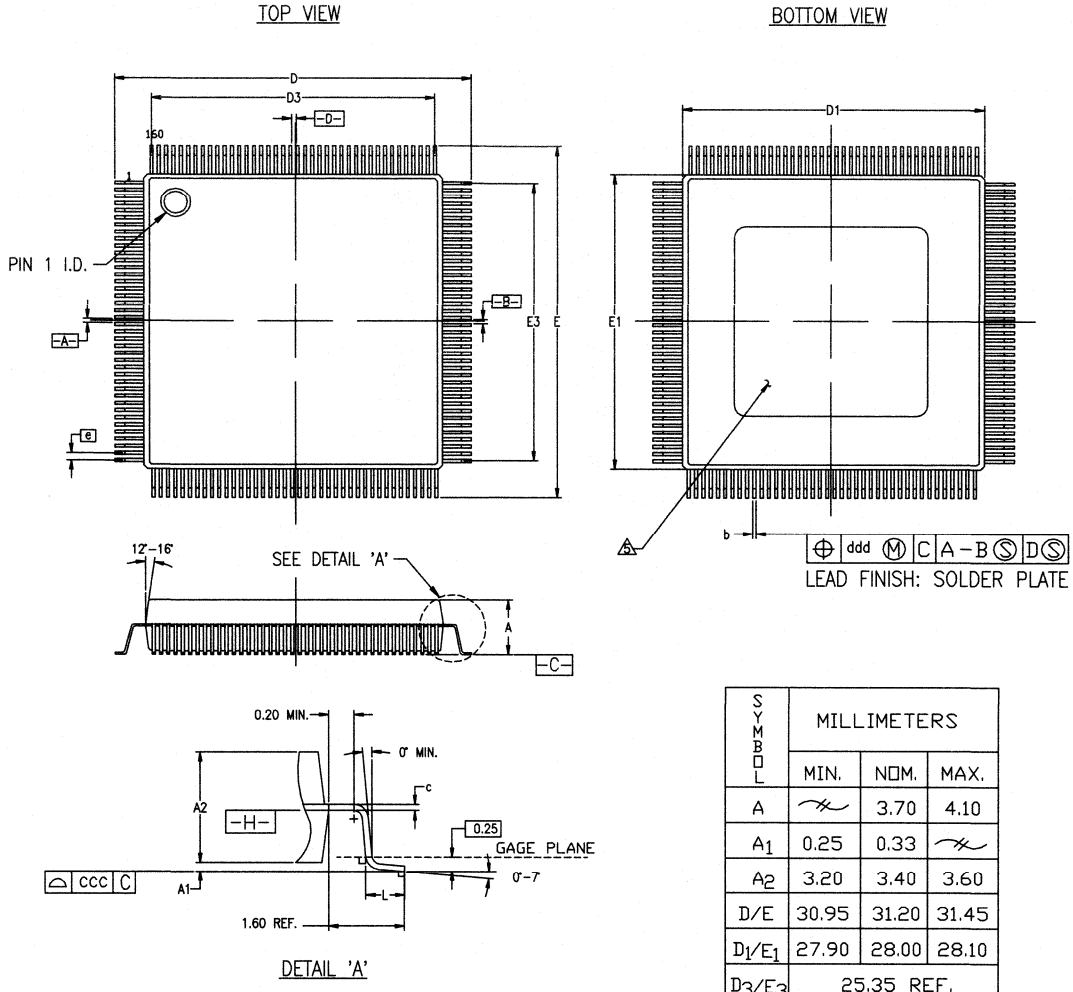


SYMBOL	MILLIMETERS		
	MIN.	NOM.	MAX.
A	---	---	3.40
A_1	0.25	---	---
A_2	2.55	2.80	3.05
D	22.95	23.20	23.45
D_1	19.90	20.00	20.10
D_3	18.85 REF.		
E	16.95	17.20	17.45
E_1	13.90	14.00	14.10
E_3	12.35 REF.		
L	0.73	0.88	1.03
e	0.65 BSC		
b	0.22	---	0.38
c	0.13	---	0.23
ccc	---	---	0.10
ddd	---	---	0.12

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982.
2. DIMENSIONS "D1" AND "E1" DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION IS 0.25mm PER SIDE.
3. THE TOP OF PACKAGE MAY BE EQUAL TO OR SMALLER THAN THE BOTTOM OF PACKAGE BY 0.15 MILLIMETERS.
4. PACKAGE CONFORMS TO JEDEC OUTLINE MO-108-CC1
5. THE SAME PACKAGE DIMENSIONS APPLY FOR THERMALLY ENHANCED PRODUCTS. HEAT SINK IS ADDED. THE PACKAGE CODE IS "HQ".

100-PIN PQFP (PQ100)
100-PIN HEAT SINK PQFP (HQ100)

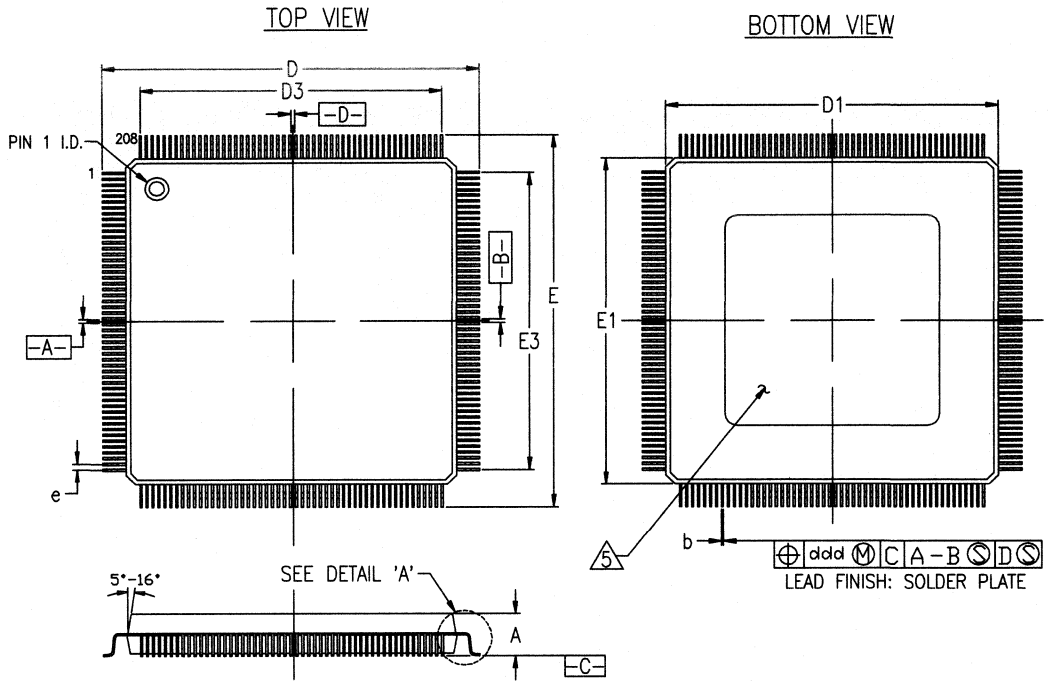


SYMBOL	MILLIMETERS		
	MIN.	NOM.	MAX.
A		3.70	4.10
A_1	0.25	0.33	
A_2	3.20	3.40	3.60
D/E	30.95	31.20	31.45
D_1/E_1	27.90	28.00	28.10
D_3/E_3	25.35 REF.		
L	0.73	0.88	1.03
e	0.65 BSC.		
b	0.22		0.38
c	0.13		0.23
ccc		0.10	
ddd		0.12	

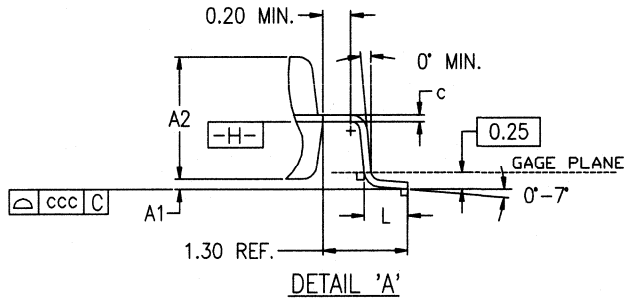
NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982.
2. DIMENSIONS "D1" AND "E1" DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION SHALL NOT EXCEED 0.25mm PER SIDE.
3. PACKAGE TOP DIMENSIONS MAY BE SMALLER THAN THE BOTTOM DIMENSIONS BY 0.20mm.
4. PACKAGE CONFORMS TO JEDEC MO-108-DD1
5. THE SAME PACKAGE DIMENSIONS APPLY FOR THERMALLY ENHANCED PRODUCTS. HEAT SINK IS ADDED. THE PACKAGE CODE IS "HQ".

160-PIN PQFP (PQ160)
160-PIN HEAT SINK PQFP (HQ160)



SYMBOL	MILLIMETERS		
	MIN.	NOM.	MAX.
A	--	3.70	4.10
A_1	0.25	0.33	--
A_2	3.20	3.40	3.60
D/E	30.60 BSC		
D_1/E_1	28.00 BSC		
D_3/E_3	25.50 REF.		
L	0.50	0.60	0.75
e	0.50 BSC.		
b	0.17	0.22	0.27
c	0.09	--	0.20
ccc	--	--	0.08
ddd	--	--	0.08

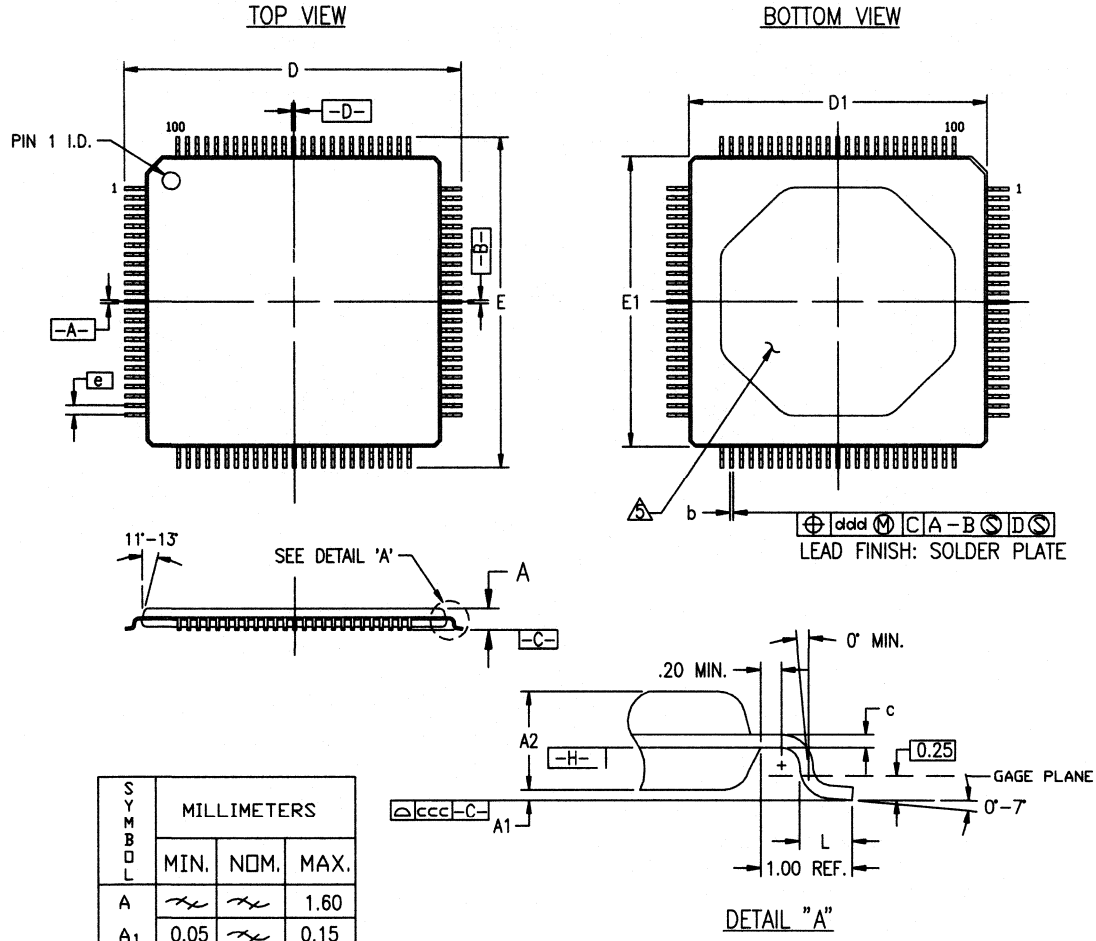


NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982.
2. DIMENSIONS "D1" AND "E1" DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE MOLD PROTRUSION SHALL NOT EXCEED 0.25mm PER SIDE.
3. PACKAGE TOP DIMENSIONS MAY BE SMALLER THAN THE BOTTOM DIMENSIONS BY 0.20mm.
4. DRAWING CONFORMS TO JEDEC MO-143-FA-1
5. THE SAME PACKAGE DIMENSIONS APPLY FOR THERMALLY ENHANCED PRODUCTS. HEAT SINK IS ADDED. THE PACKAGE CODE IS "HQ".

208-PIN PQFP (PQ208)
208-PIN HEAT SINK PQFP (HQ208)

TQFP Packages — TQ100



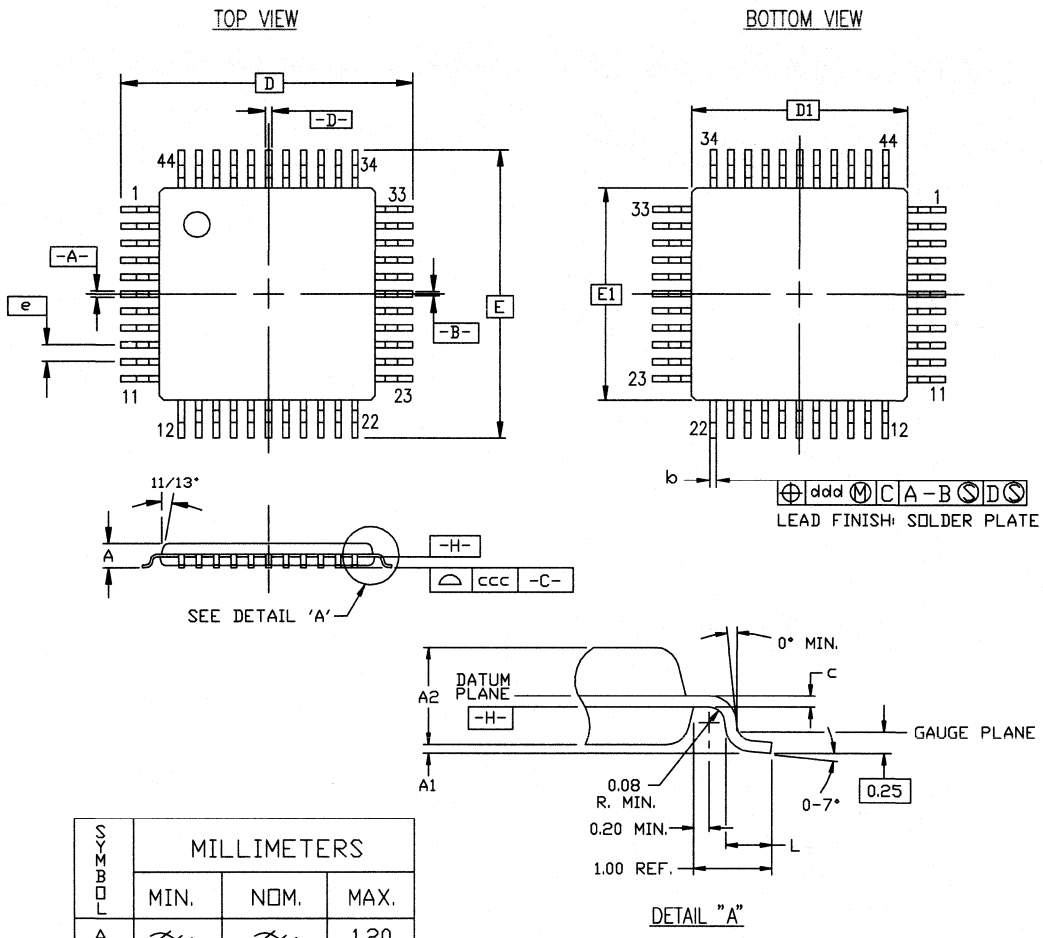
SYMBOL	MILLIMETERS		
	MIN.	NOM.	MAX.
A			1.60
A_1	0.05		0.15
A_2	1.35	1.40	1.45
D/E	16.00 BSC		
D_1/E_1	14.00 BSC		
L	0.45	0.60	0.75
e	0.50 BSC		
b	0.17	0.22	0.27
c	0.09		0.20
ccc			0.08
ddd			0.08

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982
2. DIMENSIONS "D1" AND "E1" DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION SHALL NOT EXCEED 0.25mm PER SIDE.
3. PACKAGE TOP DIMENSION MAY BE SMALLER THAN THE BOTTOM DIMENSION BY 0.15mm.
4. THIS OUTLINE CONFORMS TO JEDEC PUBLICATION 95 REGISTRATION MS-026-BED
5. THE SAME PACKAGE DIMENSIONS APPLY FOR THERMALLY ENHANCED PRODUCTS. HEAT SINK IS ADDED. THE PACKAGE CODE IS "HT".

100-PIN TQFP (TQ100)
100-PIN HEAT SINK TQFP (HT100)

VQFP Packages — VQ44



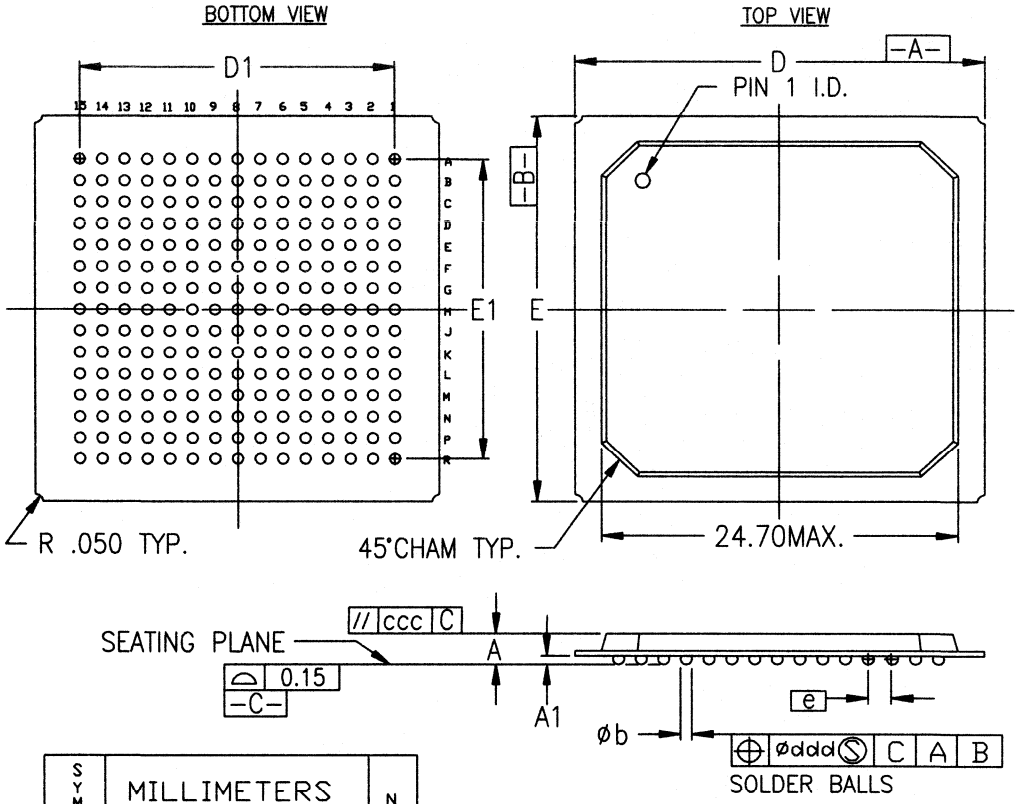
SYMBOL	MILLIMETERS		
	MIN.	NOM.	MAX.
A	—	—	1.20
A_1	0.05	—	0.15
A_2	0.95	1.00	1.05
D/E	12.00 BSC		
D_1/E_1	10.00 BSC		
L	0.45	0.60	0.75
e	0.80 BSC.		
b	0.30	0.37	0.45
c	0.09	—	.020
ccc	—	—	0.10
ddd	—	—	0.20

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M–1982.
2. DIMENSIONS "D1" AND "E1" DO NOT INCLUDE MOLD PROTRUSION ALLOWABLE MOLD PROTRUSION SHALL NOT EXCEED 0.25mm PER SIDE.
3. THE TOP OF PACKAGE MAY BE SMALLER THAN THE BOTTOM OF PACKAGE BY 0.15mm.
4. CONFORMS TO JEDEC MS-026-ACB

44-PIN PLASTIC VQFP (VQ44)

BGA Packages — BG225, BG352

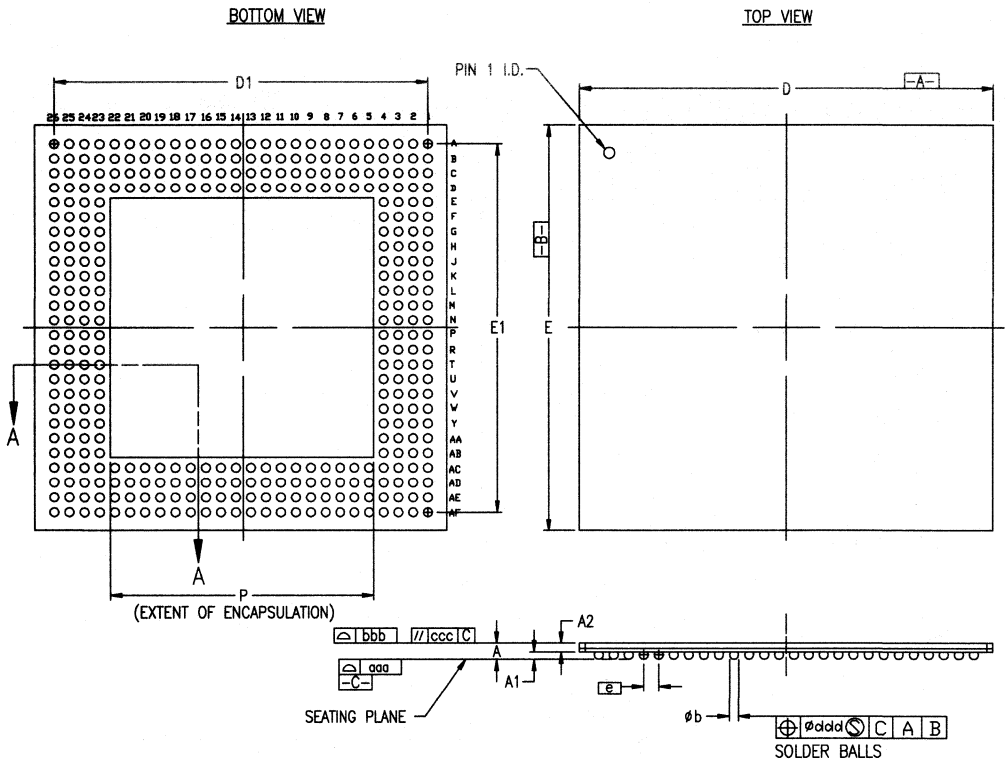


SYMBOL	MILLIMETERS			NOTE
	MIN.	NOM.	MAX.	
A	---	2.15	3.50	
A_1	0.50	0.60	0.70	
D/E	26.80	27.00	27.20	
D_1/E_1	---	21.00	---	
e	1.50 BSC			
øb	0.60	0.75	0.90	
ccc	---	---	0.35	
ddd	---	---	0.30	
M	15			

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M–1982
2. SYMBOL "M" IS THE PIN MATRIX SIZE.
3. CONFORMS TO JEDEC MO–151–CAL (DEPOPULATED)

225–BALL PLASTIC BGA (BG225)



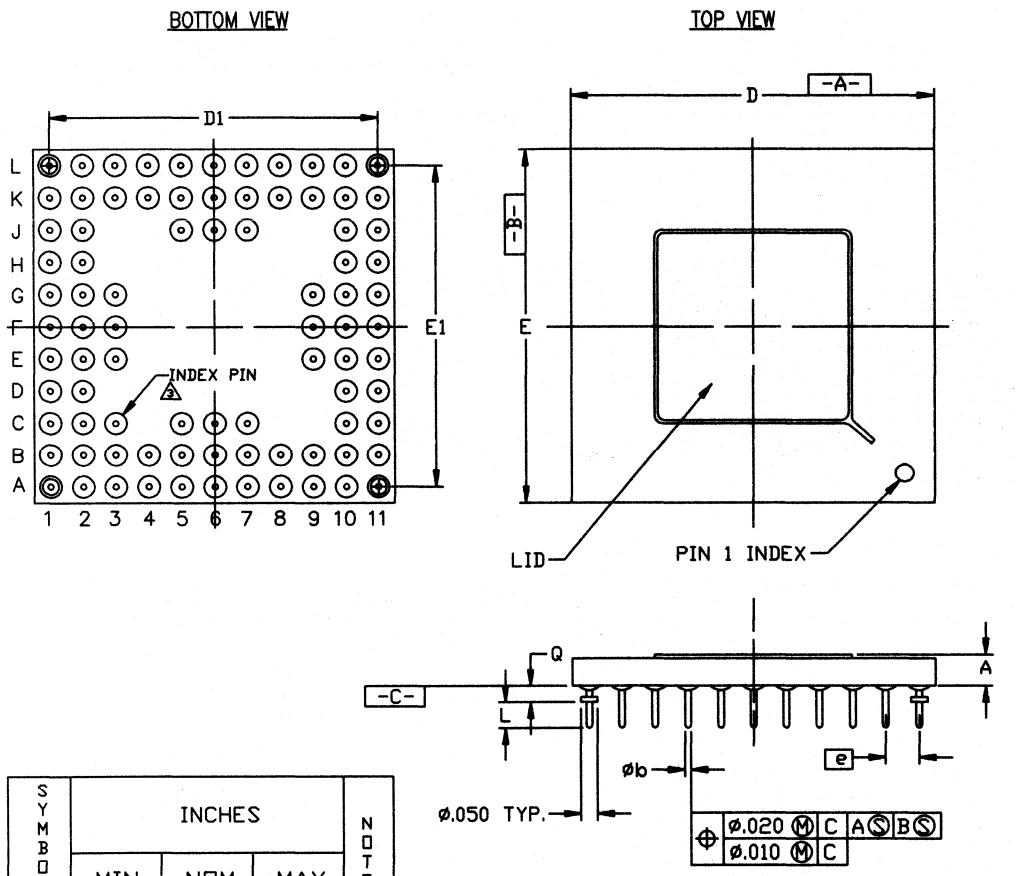
SYMBOL	MILLIMETERS			NOTE
	MIN.	NOM.	MAX.	
A	1.10	1.38	1.65	
A_1	0.50	0.60	0.70	
A_2	0.60	~~	0.95	
A_3	0.25	~~	~~	
D/E	34.80	35.00	35.20	
D_1/E_1	~~	31.75	~~	
e	1.27 BSC			
øb	0.60	0.75	0.90	
P	~~	~~	25.70	
aaa	~~	~~	0.15	
bbb	~~	~~	0.20	
ccc	~~	~~	0.25	
ddd	~~	~~	0.30	
M	26			

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M–1982
2. SYMBOL "M" IS THE PIN MATRIX SIZE.
3. CONFORMS TO JEDEC MO–151–BAR (DEPOPULATED)

352–BALL PLASTIC BGA (BG352)
CAVITY DOWN

Ceramic PGA Packages — PG84, PG144



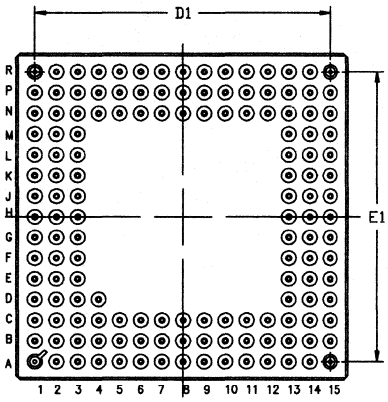
SYMBOL	INCHES			NOTE
	MIN.	NOM.	MAX.	
A			.145	
D/E	1.090	1.100	1.115	
D_1/E_1	1.000 BSC			
L	.120	.130	.140	
Q	.045		.060	
e	.100 BSC			
øb	.016	.018	.020	
M	11			

NOTES:

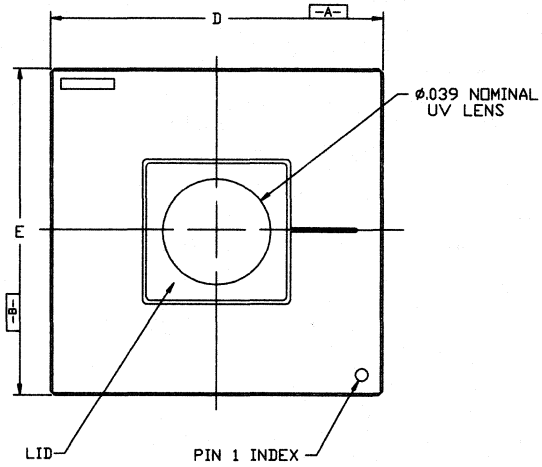
1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M–1982
2. SYMBOL "M" IS THE PIN MATRIX SIZE.
3. PIN C3 MAY OR MAY NOT BE ELECTRICALLY CONNECTED.
4. CONFORMS TO JEDEC MO–066–AC
5. LEAD FINISH: GOLD PLATED
   - COMMERCIAL (35 MICROINCHES MIN.)
   - MILITARY (50 MICROINCHES MIN.)

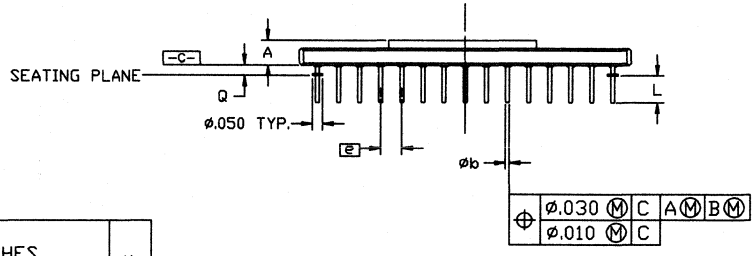
84–PIN CERAMIC PGA (PG84)

BOTTOM VIEW



TOP VIEW





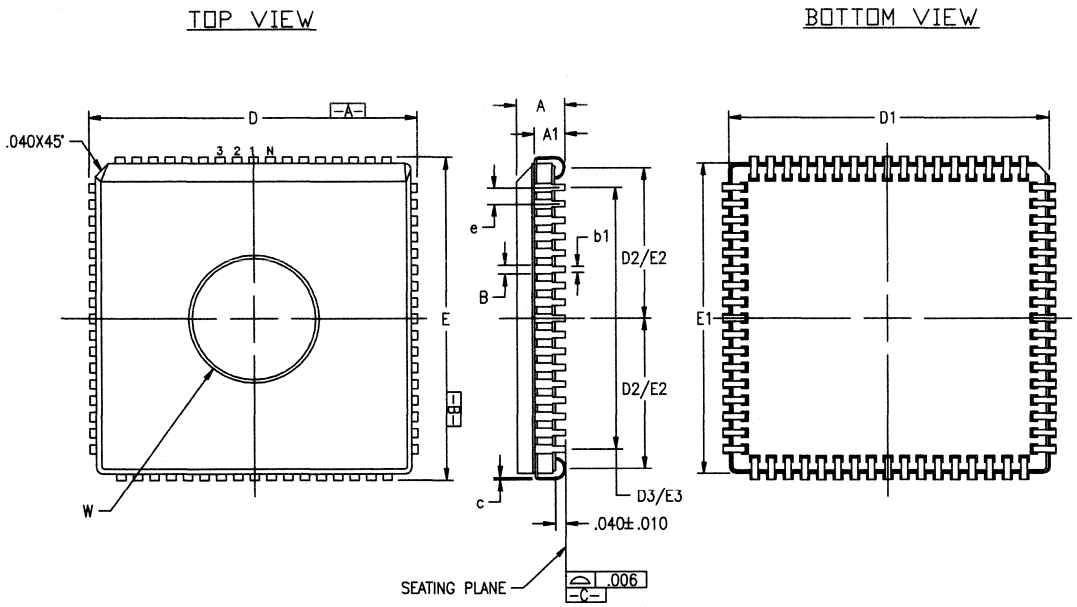
SYMBOL	INCHES			NOTE
	MIN.	NOM.	MAX.	
A			.145	
D/E	1.540	1.560	1.580	
D_1/E_1	1.400 BSC			
L	.120	.130	.140	
Q	.050		.060	
e	.100 BSC			
øb	.016	.018	.020	
M	15			

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M–1982
2. SYMBOL "M" IS THE PIN MATRIX SIZE.
3. CONFORMS TO JEDEC MO–067–AG
4. LEAD FINISH: GOLD PLATED
   - COMMERCIAL (35 MICROINCHES MIN.)
   - MILITARY (50 MICROINCHES MIN.)

144–PIN WINDOWED PGA (PG144)

Windowed CLCC Packages — WC44, WC68, WC84



SYMBOL	DIMENSION IN INCHES								
	MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX
A	.155	.172	.190	.155	.172	.190	.155	.172	.190
A1	.090	---	.120	.090	---	.120	.090	---	.120
B	.026	.028	.032	.026	.028	.032	.026	.028	.032
b1	.017	.019	.022	.017	.019	.022	.017	.019	.022
c	.006	.007	.012	.006	.007	.012	.006	.007	.012
D/E	.685	.690	.695	.985	.990	.995	1.185	1.190	1.195
D1/E1	.630	.650	.665	.930	.950	.965	1.130	1.150	1.165
D2/E2	.290	.305	.320	.440	.455	.470	.540	.555	.570
D3/E3	.500 REF.			.800 REF.			1.000 REF.		
e	.050 BSC			.050 BSC			.050 BSC		
N	44			68			84		
W	ø.350			ø.390 REF.			.450 SQ. REF		

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M–1982.
2. LEAD WIDTH DIMENSION INCLUDE LEAD TRIM OFFSET and LEAD FINISH, LEAD FINISH: (HOT SOLDER DIP)
3. SYMBOL 'N' IS THE NUMBER OF TERMINALS.
4. SYMBOL 'W' IS THE DIMENSION OF THE EPROM WINDOW.
5. THESE PACKAGES MEET DIMENSIONAL REQUIREMENTS OF JEDEC MO–087, VARIATIONS – AB(WC44); AD(WC68); AE(WC84).

44, 68, and 84–PIN WINDOWED CLCC (WC44, 68 and 84)

1 ISP and JTAG Support

2 Application Notes

3 XC9500 Data Sheets

4 XC7300 Data Sheets

5 Device Packaging

6 Quality Assurance

7 Technical Support

8 Sales Offices, Representatives, Distributors

Visit us on the WEB at:

www.xilinx.com

for the latest product information.

Quality Assurance Program

Quality Assurance encompasses all aspects of company business. Xilinx continually strives to improve quality to meet customer's changing needs and expectations. To do this, the company is dedicated to the following.

- To provide a broad range of products and services that satisfy both the expectations of customers and the company's stringent quality standards.

- To emphasize open communication with customers and suppliers, supported with the necessary statistical data.

- To continually improve the quality of Xilinx products, services, and company efficiency

- To maintain a work environment that fosters quality and reliability leadership and excellence.

From its inception, Xilinx has been committed to delivering the highest quality, most reliable programmable logic available. A strong Quality Assurance and Reliability program begins at the initial design stages and is carried through to final shipment. An extensive, on-going reliability-testing program is used to predict the field performance of all Xilinx devices.

These tests provide an accelerated method of emulating long-term system operation in severe field environments. From the performance of the devices during these tests, predictions of actual field performance under a variety of conditions can be easily calculated.

Xilinx is committed to customer satisfaction. By adhering to the highest quality standards, the company has achieved leadership in the CPLD and FPGA manufacturing areas.

Quarterly reports describing the nature and purpose of the various reliability tests performed on finished devices are available. Please contact the Quality Assurance and Reliability Department at Xilinx.

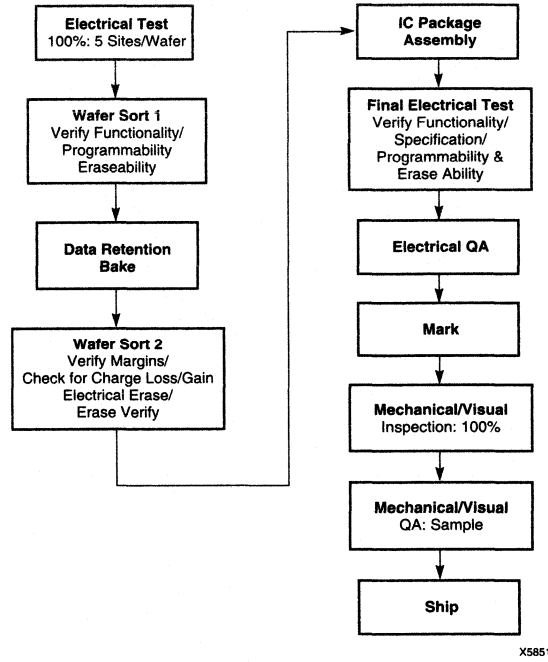


Figure 1: Wafer-Sort, Assembly, and Final Flow for Xilinx XC9500 Dvices

XILINX®

Technical Support

1 ISP and JTAG Support

2 Application Notes

3 XC9500 Data Sheets

4 XC7300 Data Sheets

5 Device Packaging

6 Quality Assurance

7 Technical Support

8 Sales Offices, Representatives, Distributors

Visit us on the WEB at:

www.xilinx.com

for the latest product information.

A complete and uniquely accessible offering of worldwide technical support services is available to Xilinx users.

Xilinx Field Application Engineers, located at sales offices and technical support centers worldwide, provide local engineering support, including design evaluation of new projects, close consultation throughout the design process, special training assignments, and new product presentations. Because their role as advisors and troubleshooters keeps them constantly on the go, they are best used, not for general questions, but for more targeted queries such as those related to architectural recommendations. The worldwide network of Xilinx sales representatives and distributors also provide local technical support for Xilinx users.

More general queries can be directed to the telephone "hotlines". Permanent teams of expert Technical Support Engineers located in the United States, United Kingdom, France, Germany, and Japan can handle problems and answer questions right on the spot, ensuring that the design process keeps moving forward.

In addition, Xilinx has several automated services, collectively referred to as X-TALX, to provide answers to user's queries 24 hours a day. These include a world wide web site, E-mail server, automated FAX system, bulletin board system, and special interest E-mail groups.

Many different publications assist users in completing designs quickly and efficiently, including technical manuals, data sheets, the AppLINX CD-ROM (a regularly-updated collection of the latest application notes and design hints), and the quarterly XCell newsletter.

For more in-depth support and instruction, a dedicated training organization conducts technical training classes worldwide. Courses geared for both novice and experienced users are available.

The following Technical Support Services are discussed in more detail in this chapter:

- Technical Support Hotline
- X-TALX: The Xilinx Network of Electronic Services
  - WebLINX World Wide Web Site
  - XDOCs E-mail document server
  - XFACTS document server
  - Xilinx Technical Bulletin Board Service
- Technical Literature
- AppLINX CD-ROM
- XCELL newsletter

Technical Support Hotlines

The technical support hotlines give Xilinx users direct telephone access to Xilinx Technical Support Engineers worldwide, providing a quick resolution to any problem that occurs during the design process. Technical questions also may be submitted via FAX or E-mail.

Hotline Support, U.S.

Customer Support Hotline	800-255-7778 Hrs: 8:00 a.m. - 5:00 p.m. Pacific time
Customer Support Fax Number	408-879-4442 Avail: 24 hrs/day-7 days/week
E-mail Address	hotline@xilinx.com
Electronic Technical Bulletin Board	408-559-9327
Customer Service (Call for software updates, authorization codes, documentation updates, etc.)	408-559-7778, Ask for customer service

Hotline Support, Japan

telephone: (81) 3-3297-9163
fax: (81) 3-3297-0067
e-mail: jhotline@xilinx.com

Hotline Support, Europe

UK, London Office

telephone: (44) 1932 820821
fax: (44) 1932 828522
Bulletin Board Service: (44) 1932 333540
e-mail: ukhelp@xilinx.com

France, Paris Office

telephone: (33) 1 3463 0100
fax: (33) 1 3463 0959
e-mail: frhelp@xilinx.com

Germany, Munich Office

telephone: (49) 89 991 54930
fax: (49) 89 904 4748
e-mail: dlhelp@xilinx.com

X-TALX: The Xilinx Network of Electronic Services

WebLINX World Wide Web Site (www.xilinx.com)

Our World Wide Web site provides access to current information, including product data sheets, application notes, press releases, financial status, employment opportunities, and an on-line technical support database. *Smart*Search™, our industry-wide search engine, is the definitive resource for programmable logic information. *Smart*Search™ Agents will watch the Web for you and inform you, via e-mail, when new or updated information is found. An FTP site also is available to facilitate the quick and easy transfer of design and data files (ftp.xilinx.com).

XDOCs E-mail Document Server

The XDOCS E-mail system provides 24-hour a day, 7 days a week access to the same database that the Technical Support Engineers use. This database is updated regularly with information on bugs, workarounds, and helpful hints. Via E-mail, users can search for a specific record, or supply keywords to trigger a search of the database; XDOCS will send the requested information by return E-mail. Automated updates also can be sent on a periodic basis notifying users of new additions to the system. To subscribe to XDOCS, send an E-mail to xdocs@xilinx.com with "help" as the only word in the subject header.

XFACTS Document Server

The XFACTS automated FAX system provides the same information as XDOCS, but uses a phone/FAX interface instead of E-mail. Using a touch-tone telephone, users can request documents that are sent to their FAX machine. Located in San Jose, California, the XFACTS system can be reached at 408-879-4400.

Xilinx Technical Bulletin Board Service (408) 559-9327

To provide users with up-to-date information and software support, Xilinx provides a 24-hour electronic bulletin board system (BBS). The Xilinx Technical Support BBS is available to all registered Xilinx development system users. Users with full privileges can browse files on the bulletin board, download those of interest, or upload files to Technical Support Engineers.

All BBS files can be accessed through the Xilinx Web and FTP locations.

New bulletin board users must answer a questionnaire when they first access the BBS. After answering the questionnaire, callers can browse through the file areas or upload files. A caller with a valid XACT protection key or valid host ID will be given full user privileges within 24 hours.

The software and hardware requirements for accessing the BBS are as follows:

Baud Rate	28.8K or less bps
Character Format	8 data bits, no parity, 1 stop bit
Transfer Protocols	ASCII, Xmodem, Ymodem, Zmodem

The Xilinx Technical Support BBS is a menu-driven system. To choose a menu command, simply type the highlighted first letter of the command. Most commands are "hot keys" and do not require you to press the return key. Here is a quick description of the available menu commands:

Main

U)pload	Upload a file to the Technical Support group.
D)ownload	Download a file. This assumes you already know the filename, otherwise select the File Manager.
F)ile Manager	Takes you to the File Manager menu. This menu is for locating files.
S)ystem Folder	Takes you to the System menu. This menu is for changing your password, display options, etc.

File Manager

F)lag	Flag files for download.
L)ocate Files	Use wildcards to search for files.
N)ew Files	Lists recently added files.
Z)ippy DIR scan	Searches for text in file descriptions.
#'s	Chooses a file area to browse.

System

M)ode of display	Toggles between text and graphics display
P)age length	Changes the number of printed lines between "More?" prompts.
T)ransfer Protocol	Changes the default transfer protocol.
V)iew Settings	Shows current settings and user information.
W)rite User Info	Changes current user settings.

E-mail addresses for questions related to specific applications

Digital Signal Processing applications	dsp@xilinx.com
PCI-bus applications	pci@xilinx.com
Plug and Play ISA applications	PnP@xilinx.com
PCMCIA card applications	pcmcia@xilinx.com
Asynchronous Transfer Mode applications	atm@xilinx.com
Reconfigurable Computing applications	reconfig@xilinx.com

Technical Support E-mail addresses

hotline@xilinx.com	USA, Xilinx Headquarters
ukhelp@xilinx.com	United Kingdom
frhelp@xilinx.com	France
dlhelp@xilinx.com	Germany
jhotline@xilinx.com	Japan

Technical Literature

Xilinx offers many different publications to assist users in completing designs quickly and efficiently. These include technical manuals, Data Books, data sheets, application notes, AppLINX CD, and the XCELL newsletter. Many of these publications are available on-line at the Xilinx WebLINX World Wide Web site.

As part of the development system products, Xilinx provides manuals and supporting documents for the development system tools, libraries, CAE tool interfaces, and related software tools. Many of these manuals are available on the CD that holds the software as well as hardcopy format. On-line help facilities also are an integral part of the development system products.

AppLINX

AppLINX is a collection of current application notes and other new technical documentation provided on a CD-ROM for easy reference by the design engineer. All the material on the CD is provided in Adobe Acrobat format for easy viewing and printing. The AppLINX CD is updated regularly as new material becomes available.

XCELL Newsletter

XCELL, the quarterly journal for Xilinx programmable logic users, is dedicated to supplying up-to-date information for system designers. A typical issue includes descriptions of new products, updates on component and software availability and revision levels, application ideas, design hints and techniques, and answers to frequently-asked questions.

To add your name to the XCELL subscription list, please send your name, company affiliation, and mailing address to Brad Fawcett, XCELL editor, via FAX at 408-879-4676 or via e-mail sent to brad.fawcett@xilinx.com.

HW-130 Programmer

January, 1997 (Version 1.1)

Device and Package Support

- XC1700 Serial PROMs
- XC7000 CPLDs
- XC9500 CPLDs
- Supports all Xilinx package types

Programmer Accessories

- Universal international power supply
- Power cord options for US/Asia, UK, EU and J standards.
- Serial download cable and adapters
- Users manual

Interface Software and System Requirements

The programmer software operates on a variety of different platforms. Table 1 indicates the minimum system requirements for each of the supported platforms. In all cases, a 3.5" disk drive or a CD-ROM drive and an RS-232 serial port are required. A mouse is recommended.

Programmer Functional Specifications

- Device programming and verification
- CPLD security control
- PROM reset polarity control
- Checksum calculation and comparison
- Blank check and signature ID tests
- Master device program upload
- File transfer and comparison
- Self check and auto calibration

Programming Socket Adapters

- Supports all package styles: PLCC, PQFP, BGA, SOIC, VOIC, PGA and DIP
- CPLD adapters for the HW-120 may be used on the HW-130 for the XC7000 devices.

Electrical Requirements and Physical Specifications

- Operating voltage: 100-250 VAC, 50-60 Mhz
- Power consumption: 1.0 Amps
- Dimensions: 6" x 7.75" x 2"
- Weight: 1 lb.
- Safety standards: approved by UL, CSA, TUV

New Programming Algorithm Support

The new programmer algorithms are available via the Xilinx BBS and e-mail:

- Send e-mail to xdocs@Xilinx. com with "search hw130" in the subject field
- When accessing the Xilinx bulletin board, type "F" and select directory #3. Select either Programming Support or type "Zhw130" to view all HW-130 related files. Refer to Section 8 of this Application Guide for instructions on how to access the Xilinx bulletin board.

Table 1: Interface Software and System Requirements

Requirements	DOS	Windows 3.1	Windows 95	Windows NT	Sun OS	Solaris	HP9000/700	IBM RS6000
Memory Needed	500KB	4MB	8MB	16MB	—	—	—	—
Hard Disk Space	2MB	2MB	2MB	2MB	6MB	6MB	6MB	6MB
System Software	3.3 or greater	3.1.x.	4.00	3.1 or greater	SunOS 4.1.3 or greater	SunOS 5.3 or greater, (Solaris 2.3 or higher)	HP-UX A09.05 or greater	AIX 3.2.5 or greater

Adapter Selection Table

Product Family	Package Types	Adapter P/N
XC7200A	PLCC/CLCC 44	HW-132-PC44
XC7200A	PLCC/CLCC 68	HW-132-PC68
XC7200A	PLCC/CLCC 84	HW-132-PC84
XC7200A	PGA 84	HW-132-PG84
XC7300/XC9500	PLCC/CLCC 44	HW-133-PC44
XC7300	PQFP 44	HW-133-PQ44
XC7300/XC9500	VQFP 44	HW-133-VQ44
XC7300	PLCC/CLCC 68	HW-133-PC68
XC7300/XC9500	PLCC/CLCC 84	HW-133-PC84
XC7300/XC9500	PQFP 100	HW-133-PQ100
XC7300	PGA 144	HW-133-PG144
XC7300[1]	PQFP 160	HW-133-PQ160
CPLD (XC7300/XC9500)[1]	PQFP 160	HW-133-PQ160
XC7300	BGA 225	HW-133-BG225
XC1700	DIP 8	HW-137-DIP8
XC1700	PLCC20/SO8/VO8	HW-137-PC20/SO8
Calibration Adapter		HW-130-CAL

1) Xilinx manufactures two versions of the HW-133-PQ160 adapter. The correct adapter for programming XC9500 devices has "CPLD" written on the front label, at the top left side, under the Xilinx logo.

Sales Offices

1 ISP and JTAG Support

2 Application Notes

3 XC9500 Data Sheets

4 XC7300 Data Sheets

5 Device Packaging

6 Quality Assurance

7 Technical Support

8 Sales Offices, Representatives, Distributors

Visit us on the WEB at:

www.xilinx.com

for the latest product information.

Sales Offices, Sales Representatives, and Distributors

January, 1997 (Version 1.03)

Headquarters

XILINX, Inc.
2100 Logic Drive
San Jose, CA 95124
(408) 559-7778
FAX: 408-559-7114

Xilinx Sales Offices

NORTH AMERICA

XILINX, Inc.
1281 Oakmead Pkwy.
Suite 202
Sunnyvale, CA 94086
(408) 245-9850
FAX: 408-245-9865

XILINX, Inc.
5690 DTC Blvd.
Suite 490W
Englewood, CO 80111
(303) 220-7541
FAX: 303-220-8641

XILINX, Inc.
15615 Alton Parkway
Suite 280
Irvine, CA 92718
(714) 727-0780
FAX: 714-727-3128

XILINX, Inc.
61 Spit Brook Rd.
Suite 403
Nashua, NH 03060
(603) 891-1098
FAX: 603-891-0890

XILINX, Inc.
905 Airport Rd.
Suite 200
West Chester, PA 19380
(610) 430-3300
FAX: 610-430-0470

XILINX, Inc.
939 North Plum Grove Road
Suite H
Schaumburg, IL 60173
(847) 605-1972
FAX: 847-605-1976

XILINX, Inc.
6080-C Six Forks Road
Raleigh, NC 27609
(919) 846-3922
FAX: 919-846-8316

XILINX, Inc.
4100 McEwen, Suite 237
Dallas, TX 75244
(214) 960-1043
FAX: 214-960-0927

EUROPE

XILINX, Ltd.
Benchmark House
203 Brooklands Road
Weybridge, Surrey KT13 ORH
United Kingdom
Tel: (44) 1-932-349401
FAX: (44) 1-932-349499

XILINX SARL
Espace Jouy Technology
21, rue Albert Calmette, Bt. C
78353 Jouy En Josas, Cedex
France
Tel: (33)-1-3463-01-01
FAX: (33)-1-3463-01-09

XILINX GmbH
Dorfstr. 1
85609 Aschheim
Germany
Tel: (49) 89-991549-0
FAX: (49) 89-904-4748

XILINX, AB
c/o Dipcom Electronics
Torshamnsgatan 35
Box 1230
S-164 28 Kista
Sweden
Tel: +46-8-752-24-70
FAX: +46-8-750-62-60

JAPAN

XILINX K. K.
Daini-Nagaoka Bldg. 2F
2-8-5, Hatchobori, Chuo-ku
Tokyo 104
Japan
Tel: (81) 3-3297-9191
FAX: (81) 3-3297-9189

ASIA PACIFIC

XILINX Asia Pacific
Unit 4312, Tower II
Metroplaza
Hing Fong Road
Kwai Fong, N.T.
Hong Kong
Tel: 852-2-424-5200
FAX: 852-2494-7159

North American Distributors

Hamilton Hallmark
Locations throughout
the U.S. and Canada.
1-800-332-8638
FAX: 1-800-257-0568

Marshall Industries
Locations throughout
the U.S. and Canada.
1-800-522-0084
FAX: 818 -307-6297

Insight Electronics
Locations throughout
the U.S.
1-800-677-7716
FAX: 619-587-1380

Nu Horizons
Electronics Corp.
Locations throughout
the U.S.
(516) 226-6000
FAX: 516-226-6262

U.S. Sales Representatives

ALABAMA

Electro Source
4835 University Sq., Ste.11
Huntsville, AL 35816
Tel: 205-830-2533
FAX: 205-830-5567

ARIZONA

Quatra Associates
4645 S. Lakeshore Dr.
Suite 1
Tempe, AZ 85282
(602) 820-7050
FAX: 602-820-7054

ARKANSAS

Bonser-Philhower Sales
689 W. Renner Road
Suite 101
Richardson, TX 75080
(214) 234-8438
FAX: 214-437-0897

CALIFORNIA

Quest-Rep Inc.
6494 Weathers Pl, Suite 200
San Diego, CA 92121
(619) 622-5040
FAX: 619-622-5047

Norcomp
1267 Oakmead Pkwy
Sunnyvale, CA 94086
(408) 733-7707
FAX: 408-774-1947

Norcomp
8880 Wagon Way
Granite Bay, CA 95746
(916) 791-7776
FAX: 916-791-2223

Norcomp
30101 Agoura Ct. #234
Agoura, CA 91301
(818) 865-8330
FAX: 818-865-2167

Norcomp
30 Corporate Park #200
Irvine, CA 92714
(714) 260-9868
FAX: 714-260-9659

COLORADO

Luscombe Engineering, Inc.
1500 Kansas Ave. Suite 1B
Longmont, CO 80501
(303) 772-3342
FAX: 303-772-8783

CONNECTICUT

John E. Boeing, Co., Inc.
101 Harvest Park, Bldg. 1A
No. Plains Industrial Road
Wallingford, CT, 06492
(203) 265-1318
FAX: 203-265-0235

DELAWARE

Delta Technical Sales, Inc.
122 N. York Rd., Suite 9
Hatboro, PA 19040
(215) 957-0600
FAX: 215-957-0920

FLORIDA

Semtronic Assoc., Inc.
657 Maitland Avenue
Altamonte Springs, FL 32701
(407) 831-8233
FAX: 407-831-2844

Semtronic Assoc., Inc.
3471 N. W. 55th Street
Ft. Lauderdale, FL 33309
(305) 731-2484
FAX: 305-731-1019

Semtronic Assoc., Inc.
1467 South Missouri Avenue
Clearwater, FL 34616
(813) 461-4675
FAX: 813-442-2234

GEORGIA

Electro Source
3280 Pointe Parkway, #1500
Norcross, GA 30092
(770)-734-9898
FAX: 770-734-9977

IDAHO (Southwest)

Thorson Pacific, Inc.
14575 Bel-Red Road #102
Bellevue, WA 98007
(206) 603-9393
FAX: 206-603-9380

Luscombe Engineering, Inc.
670 East 3900 South #103
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

ILLINOIS

Beta Technology Sales, Inc.
1009 Hawthorn Drive
Itasca, IL 60143
(708) 250-9586
FAX: 708-250-9592

Advanced Technical Sales
13755 St. Charles Rock Rd.
Bridgeton, MO 63044
(314) 291-5003
FAX: 314-291-7958

INDIANA

Gen II Marketing,Inc.
31 E. Main St.
Carmel, IN 46032
(317) 848-3083
FAX: 317-848-1264

Gen II Marketing, Inc.
1415 Magnavox Way
Sutie 130
Ft. Wayne, IN 46804
(219) 436-4485
FAX: 219-436-1977

IOWA

Advanced Technical Sales
375 Collins Road N.E.
Cedar Rapids, IA 52402
(319) 393-8280
FAX: 319-393-7258

KANSAS

Advanced Technical Sales
2012 Prairie Cir. Suite A
Olathe, KS 66062
(913) 782-8702
FAX: 913-782-8641

KENTUCKY

Gen II Marketing, Inc.
861 Corporate Dr. #210
Lexington, KY 40503
(606) 223-9181
FAX: (606) 223-2864

LOUISIANA (Northern)

Bonser-Philhower Sales
689 W. Renner Rd., Suite 101
Richardson, TX 75080
(214) 234-8438
FAX: 214-437-0897

LOUISIANA (Southern)

Bonser-Philhower Sales
10700 Richmond, Suite 150
Houston, TX 77042
(713) 782-4144
FAX: 713-789-3072

MAINE

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

MARYLAND

Micro Comp, Inc.
1421 S. Caton Avenue
Baltimore, MD 21227-1082
(410) 644-5700
FAX: 410-644-5707

MASSACHUSETTS

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

MICHIGAN

Miltimore Sales Inc.
22765 Heslip Drive
Novi, MI 48375
(810) 349-0260
FAX: 810-349-0756

Miltimore Sales Inc.
3684 44th St., Suite 100-J
Kentwood, MI 49512
(616)-554-9292
FAX: 616-554-9210

MINNESOTA

Beta Technology
18283 Minnetonka Blvd.
Suite C
Deephaven, MN 55391
(612) 473-2680
FAX: (612)473-2690

MISSISSIPPI

Electro Source
4835 University Sq., Ste.11
Huntsville, AL 35816
(205)-830-2533
FAX: 205-830-5567

MISSOURI

Advanced Technical Sales
2012 Prairie Cir. Suite A
Olathe, KS 66062
(913) 782-8702
FAX: 913-782-8641

Advanced Technical Sales
13755 St. Charles Rock Rd.
Bridgeton, MO 63044
(314) 291-5003
FAX: 314-291-7958

MONTANA

Luscombe Engineering, Inc.
670 East 3900 South #103
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

NEBRASKA

Advanced Technical Sales
375 Collins Road N.E.
Cedar Rapids, IA 52402
(319) 393-8280
FAX: 319-393-7258

NEVADA

Norcomp
8880 Wagon Way
Granite Bay, CA 95748
(916) 791-7776
FAX: 916-791-2223

Quatra Associates
(Las Vegas)
4645 S. Lakeshore Dr., Suite 1
Tempe, AZ 85282
(602) 820-7050
FAX: 602-820-7054

NEW HAMPSHIRE

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

NEW JERSEY (Northern)

Parallax
734 Walt Whitman Road
Melville, NY 11747
(516) 351-1000
FAX: 516-351-1606

NEW JERSEY (Southern)

Delta Technical Sales, Inc.
122 N. York Road, Suite 9
Hatboro, PA 19040
(215) 957-0600
FAX: 215-957-0920

NEW MEXICO

Quatra Associates
600 Autumwood Place, S. E.
Albuquerque, NM 87123
(505) 296-6781
FAX: 505-292-2092

NEW YORK (Metro)

Parallax
734 Walt Whitman Road
Melville, NY 11747
(516) 351-1000
FAX: 516-351-1606

NEW YORK

Electra Sales Corp.
333 Metro Park
Suite M103
Rochester, NY 14623
(716) 427-7860
FAX: 716-427-0614

Electra Sales Corp.
6700 Old Collamer Rd.
E. Syracuse, NY 13057
(315) 463-1248
FAX: 315-463-1717

NORTH CAROLINA

Electro Source
6050 C Six Forks Rd.
Raleigh, NC 27609
(919)-846-5888
FAX: 919-846-0408

NORTH DAKOTA

Beta Technology
18283 Minnetonka Blvd.
Suite C
Deepaven, MN 55391
(612) 473-2680
FAX: (612) 473-2690

OHIO

Bear Marketing, Inc.
3554 Brecksville Road
PO Box 427
Richfield, OH 44286-0427
(216) 659-3131
FAX: 216-659-4823

Bear Marketing, Inc.
270 Regency Ridge Drive
Suite 115
Dayton, OH 45459
(513) 436-2061
FAX: 513-436-9137

OKLAHOMA

Bonser-Philhower Sales
689 W. Renner Rd., Suite 101
Richardson, TX 75080
(214) 234-8438
FAX: 214-437-0897

OREGON

Thorson Pacific, Inc.
9600 S.W. Oak Street,
Suite 320
Portland, OR 97223
(503) 293-9001
FAX: 503-293-9007

PENNSYLVANIA

Delta Technical Sales, Inc.
122 N. York Rd., Suite 9
Hatboro, PA 19040
(215) 957-0600
FAX: 215-957-0920

Bear Marketing, Inc.
4284 Rt. 8, Suite 211
Allison Park, PA 15101
(412) 492-1150
FAX: 412-492-1155

PUERTO RICO

Semtronic Assoc., Inc
Crown Hills
125 Carite ST.
Esq. Avenue Parana
Rio Piedras, P.R. 00926
(809) 766-0700/0701
FAX: 809-763-8071

RHODE ISLAND

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

SOUTH CAROLINA

Electro Source
6050 C Six Forks Rd.
Raleigh NC 27609
(919)-846-5888
FAX: 919-846-0408

SOUTH DAKOTA

Beta Technology
18283 Minnetonka Blvd.
Suite C
Minnetonka, MN 55345
(612) 473-2680
FAX: (612) 473-2690

TENNESSEE

Electro Source
6050 C Six Forks Rd.
Raleigh, NC 27609
Tel: (919) 846-5888
FAX: 919-846-0408

TEXAS

Bonser-Philhower Sales
8240 MoPac Expwy.
Suite 295
Austin, TX 78759
(512) 346-9186
FAX: 512-346-2393

Bonser-Philhower Sales
10700 Richmond, Suite 150
Houston, TX 77042
(713) 782-4144
FAX: 713-789-3072

Bonser-Philhower Sales
689 W. Renner Rd., Suite 101
Richardson, TX 75080
(214) 234-8438
FAX: 214-437-0897

TEXAS (El Paso County)

Quatra Associates
600 Autumwood Place SE
Albuquerque, NM 87123
(505) 296-6781
FAX: 505-292-2092

UTAH

Luscombe Engineering Co.
670 East 3900 South #103
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

VERMONT

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

VIRGINIA

Micro Comp, Inc.
8811Timberlake Rd.
Suite 107
Lynchburg, VA 24502
(804) 239-2626
FAX: 804-239-1333

WASHINGTON

Thorson Pacific, Inc.
14575 Bel-Red Rd.
Suite 102
Bellevue, WA 98007
(206) 603-9393
FAX: 206-603-9380

WASHINGTON

(Vancouver, WA only)
Thorson Pacific, Inc.
9600 S.W. Oak Street
Suite 320
Portland, OR 97223
(503) 293-9001
FAX: 503-993-9007

WASHINGTON D.C.

Micro Comp, Inc.
1421 S. Caton Avenue
Baltimore, MD 21227-1082
(410) 644-5700
FAX: 410-644-5707

WEST VIRGINIA

Bear Marketing, Inc.
4284 Rt. 8 Suite 211
Allison Park, PA 15101
(412) 492-1150
FAX:412-492-1155

WISCONSIN (Western)

Beta Technology
18283 Minnetonka Blvd.
Suite C
Minnetonka, MN 55345
(612) 473-2680
FAX: 612-473-2690

WISCONSIN (Eastern)

Beta Technology Sales, Inc.
9401 N. Beloit, Suite 409
Mllwaukee, WI 53227
(414) 543-6609
FAX: 414-543-9288

WYOMING

Luscombe Engineering, Inc.
670 East 3900 South #103
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

International Sales Representatives

ASEAN

(Singapore, Malaysia,
Indonesia, Thailand,
Phillippines, Brunei)
MEMEC Asia Pacific Ltd.
10 Anson Rd. #14-02
International Plaza
Singapore 0207
Tel: (65) 222-4962
FAX: (65) 222-4939

AUSTRALIA

Advanced Component Dist.
Suite 5, Level 1, "Metro Centre"
124 Forest Rd. Hurstville 2220
P.O.Box 574 Hurstville 2220
Australia
Tel: (61) 2-9585-5030
FAX: (61) 2-9580-1095

Advanced Component Dist.
Unit 1, 14 Melrich Road
Bayswater VIC 3153
Melbourne, Australia
Tel: (61) 3-9762-4244
FAX: (61) 3-9761-1754

Advanced Component Dist.
Suite 8, 328 Albany Highway
Victoria Park, WA 6100
Australia
Tel: (61) 9-472-3232
FAX: (61) 9-470-2303

Advanced Component Dist.
20D William Street
Norwood SA 5067
Australia
Tel: (61)8-364-2844
FAX: (61)8-364-2811

Advanced Component Dist.
Ste. 1, 1048 Beaudesert Rd.
Cooper Plains
Queensland, 4108
Australia
Tel: (61) 7-246-5214
FAX: (61) 7-275-3662

AUSTRIA

SEI Elbatex GmbH
Eitnergasse 6
1230 Vienna
Austria
Tel: (43) 1-866-4220
FAX: (43) 1-866-42201

BELGIUM & LUXEMBURG

SEI Rodelco NV
Limburg Stirum 243
1780 Wemmel
Belgium
Tel: (32) 2-460-0560
FAX: (32) 2-460-0271

CANADA (BRITISH COLUMBIA)

Thorson Pacific, Inc.
4170 Still Creek Dr. #200Burnaby BC V6C 6C6
Canada
Tel: (604) 294-3999
FAX: (604) 473-7755

CANADA (ALBERTA)

Electro Source
6875 Royal Oak Ave.
Burnaby BC V5J 4J3
Canada
Tel: (604) 435-2533
FAX: (604)-435-2538

CANADA (OTTAWA)

Electro Source, Inc.
300 March Road, Suite 203
Kanata, Ontario K2K 2E4
Canada
Tel: (613) 592-3214
FAX: (613)-592-4256

CANADA (QUEBEC)

Electro Source
6600 TransCanada Hwy
Suite 420
Point Claire Quebec H9R 4S2
Canada
Tel: (514) 630-7486
FAX: 514-630-7421

CANADA (TORONTO)

Electro Source, Inc.
230 Galaxy Blvd.
Rexdale Ontario M9W 5R8
Canada
Tel: (416) 675-4490
FAX: (416)-675-6871

CHINA PEOPLE'S REPUBLIC

MEMEC (Beijing Rep Office)
Rm 5851, Blk 8, Xiynan Hotel
No. 1 Sarliha Rd.
Beijing 100046
Tel: (86) 10-8313388 X 5851
FAX: (86) 10-2564176

MEMEC Asia Pacific Ltd.
Shanghai Rep Office
Rm. 2363, West Building
Jim Jiang Hotel
59 Mao Ming Rd.
Shanghai 200020
Tel: (86) 21-2582582 x2362
FAX: (86) 21-4723388

CZECH REPUBLIC

SEI Eljapex/Elbatex GmbH
Prechodni 11
CZ-140 00 Praha 4
Czech Republic
Tel: (02)-692-8087
FAX: (02)-471-82-03

DENMARK

Micronor A/S
P.O.Box 929
Torvet 1
DK-8600 Silkeborg
Denmark
Tel: (45) 8681-6522
FAX: (45) 8681-2827

FINLAND

Memec Finland Oy
Vernissakatu 6
01300 Vantaa
Finland
Tel: (358) 0-7001-9830
FAX: (358) 0-7001-9839

FRANCE

Rep'tronic
1 Bis, rue Marcel Paul
Z.I. La Bonde
91742 Massy
France
Tel: (33) 1-69536720
FAX: (33) 1-60139198

Axess Technology
49, rue de l'Esterl
Silic 600
94663 Rungis Cedex
France
Tel: (33) 1-49-78-94-94
FAX: (33)1-49-78-03-24

Axess Technology
Route Joseph Coynel
38360 Engins
France
Tel: (33) 76-94-49-72
FAX: (33) 76-94-49-06

Axess Technology
23 le Grand Francois
La Courbatiere
38140 Rives
France
Tel: (33)76-91-45-30
FAX: (33)76-91-45-34

AVNET/EMG.
79 Rue Pierre Sémard
92 320 Chatillon
France
Tel: (33) 1 49 65 25 00
FAX : (33) 1 49 65 27 39

AVNET Composant
Sud-Ouest
Technoparc
Bat.4, Voie 5-BP404
31314 Labège Cédex
France
Tel: (33) 61 39 21 12
FAX: (33) 61 39 21 40

AVNET Composant
Aquitaine
16 Rue François Arago
Zi du Phare
33700 Mérignac
France
Tel: (33) 56 55 92 92
FAX: (33) 56 34 39 99

AVNET Composant
Rhône-Auvergne
Parc Club du Moulin à Vent
Bât 32-33, rue du Dr-G.Levy
69693 Vénissieux Cedex
France
Tel: (33) 7800-1280
FAX: (33) 7875-9597

AVNET Composant
Saint Etienne
Le Chatelet-5 place Carnot
42000 Saint etienne
France
Tel: (33) 77 92 77 66
FAX: (33) 77 92 77 30

AVNET Composant
Ouest
Technoparc-Bât.E
4 Av. des Peupliers, B.P. 43
35511 Césson Sévigné Cedex
France
Tel: (33) 99 83 84 85
FAX: (33) 99 83 80 83

AVNET Composant
Rhône-Alpes
Zac des Béalières
23 Av. de Granier
38240 Meylan
France
Tel: (33) 76 90 11 88
FAX: (33) 76 41 04 90

AVNET Composant
Nantes
Le Sillon de Bretagne
23e étage-Aile C
8 Av.des Thébaudières
44800 Saint Herblain
France
Tel: (33) 40 63 23 00
FAX: (33) 40 63 22 88

AVNET Composant
Marsielle
17, bd. Andri Aune
13006 Marseille
France
Tel: (33) 91 54 15 28

GERMANY

Avnet E2000
Stahlgruberring 12
81829 München
Germany
Tel: (49) 89-45110-01
FAX: (49) 89-45110-209

Avnet E2000
Kurfürstenstr. 130
10785 Berlin
Germany
Tel: (49) 30-214882-0
FAX: (49) 30-2141728

Avnet E2000
Friedrich-Ebert-Damm 145
22047 Hamburg
Germany
Tel: (49) 40-696-9520
FAX: (49) 40-696-2787

Avnet E2000
Benzstr. 1
70826 Gerlingen
Stuttgart Germany
Tel: (49) 7156-356-0
FAX: (49) 7156-28084

Avnet E2000
Heinrich-Hertz-Str. 52
40699 Erkrath
Düsseldorf Germany
Tel: (49) 211-92003-0
FAX: (49) 211-92003-99

Avnet E2000
Schmidtstr. 49
60326 Frankfurt/M.
Germany
Tel: (49) 69-973804-0
FAX: (49) 69-7380712

Avnet E2000
Fûrther Str. 212
90429 Nürnberg
Germany
Tel: (49) 911-93149-0
FAX: (49) 911-320821

Memec Central Europe
c/o Memec Component
Logistik GmbH
Lötscher Weg 66
D-41334 Nettetal
Germany
Tel: (49) 2153 733 92
FAX: (49) 2153 733 1 64

Metronik GmbH
Leonhardsweg 2
82008 Unterhaching
München Germany
Tel: (49) 89-611080
FAX: (49) 89-61108110

Metronik GmbH
Zum Lonnenhohl 38
44319 Dortmund
Germany
Tel: (49) 231-927110-0
FAX: (49) 231-927110-99

Metronik GmbH
Carl-Zeiss Str.6
25451 Quickborn
Hamburg Germany
Tel: (49) 4106-773050
FAX: (49) 4106-773052

Metronik GmbH
Osmiastr. 9
69221 Dossenheim
Mannheim Germany
Tel: (49) 6221-87044
FAX: (49) 6221-87046

Metronik GmbH
Pilotystr. 27/29
90408 Nürnberg
Germany
Tel: (49) 911-363536
FAX: (49) 911-353986

Metronik GmbH
Löwenstr. 37
70597 Stuttgart
Germany
Tel: (49) 711-7640333
FAX: (49) 711-7655181

Metronik GmbH
Schönauer Str. 113
04207 Leipzig
Germany
Tel: (49) 341-4239413
FAX: (49) 341-4239424

Metronik GmbH
Bahnstrasse 9
65205 Wiesbaden
Germany
Tel: (49) 611-97384-0
FAX: (49) 611-97384-18

Metronik GmbH
Franz-Schubert-Str.41
16548 Glienicke
Berlin Germany
Tel: (49) 33056-845-0
FAX: (49) 33056-845-50

Intercomp
Am Hochwald 42
D-82319
Starnberg Germany
Tel: (49) 8151-16044
FAX: (49) 8151-79270

Intercomp
Kniebisstr. 40/1
78628 Rottweil
Stuttgart Germany
Tel: (49) 741-1 48 45
FAX: (49) 741-1 52 20

Intercomp
Schustergasse 35
55278 Köngernheim
Frankfurt Germany
Tel: (49) 6737-9881
FAX: (49) 6737-9882

Intercomp
Hans Ackermann Str.23
91322 Grâfenberg
Germany
Tel: (49) 9192-998917
FAX: (49) 9192-998918

GREECE

Peter Caritato & Assoc. S. A.
Llia Iliot, 31
Athens 11743 Greece
Tel: (30) 1-9020115
FAX: (30) 1-9017024

Semicon
104 Aeolu Str.
10564 Athens
Greece
Tel: (30)-1-325 3126
FAX: (30) 1-321-6063

HONG KONG

MEMEC Asia Pacific Ltd.
Unit No. 2308-2319, Tower I
Metroplaza, Hing Fong Road,
Kwai Fong, N.T.
Hong Kong
Tel: (852) 2410 2780
FAX: (852) 2401 2518

HUNGARY

SEI Dataware KFT/Elbatex GmbH
Angol Utca 22
1149 Budapest
Hungary
Tel: (36) 1-163-5081
FAX: (36) 1-251-5517

SEI Eljapex/Elbatex GmbH
Vaci u 202
1138 Budapest
Hungary
Tel: (36) 1-269-9093/95
FAX: (36) 1-269-9096

INDIA

Core EI Micro Systems
45131 Manzanita Court
Fremont, California 94539
USA
Tel:(510) 770-1066
FAX: 510-657-1525

Core EI Logic System
ETD, Crompton Greaves Ltd.
First Floor, Surya Bhawan
Fergusson College Rd.
Pune 411005 India
Tel: (91) 212-323982
FAX: (91) 212-332838

IRELAND

Memec Ireland Ltd.
Gardner House
Bank Place
Limerick
Ireland
Tel: (353) 61-411842
FAX: (353) 61-411888

ISRAEL

E.I.M International Ltd.
9 Hashiloach Street
P.O. Box 4000
Kiryat Matalon
Petach Tikva 49130
Israel
Tel: (972) 3-92 33257
FAX: (972) 3-924 4857

ITALY

ACSIS S.R.L.
Via Alberto Mario. 26
20149 Milano, Italy
Tel: (39) 2-48022522
FAX: (39) 2-48012289

Silverstar-Celdis
Viale Fulvio Testi 280
20126 Milano, Italy
Tel: (39) 2-66125 1
FAX: (39) 2-66101359

Silverstar-Celdis
Via Collamarini, 22
40138 Bologna, Italy
Tel: (39) 51-538500
FAX: (39) 538831

Silverstar-Celdis
Via Paisiello,30
00198 Roma, Italy
Tel: (39) 6-8848841
FAX: (39) 6-8553228

Silverstar-Celdis
Centro Direzionale Benelli
Via Del Monaco, 16
61100 Pesaro, Italy
Tel: (39) 721-26560
FAX: (39) 721-400896

Silverstar-Celdis
Via A.da Noli 6
50127 Firenze, Italy
Tel: (39) 55-43 5125
FAX: (39) 55-43 77184

Silverstar-Celdis
Centro Piero Della Francesca
Corso Svizzera, 185 Bis
10149 Torino, Italy
Tel: (39) 11 77 10082
FAX: (39) 11 77 64921

Silverstar-Celdis
Via G. Antonio Resti 63
00143 Roma, Italy
Tel: (39) 6 519 57527
FAX: (39) 6 504 3330

Silverstar-Celdis
Via.delle Indus.13
35010 Limena
Padova, Italy
Tel: (39) 49 88 40044
FAX: (39) 49 88 41079

Silverstar-Celdis
Via Famagosta 1/5
17100 Savona, Italy
Tel: (39) 19 81 5090
FAX: (39) 19 81 5091

JAPAN

Okura Electronics Co., Ltd.
3-6, Ginza 2-chome,
Chuo-ku,
Tokyo, 104 Japan
Tel: (81) 3-3564-6871
FAX: (81) 3-3564-6870

Okura Electronics
Service Co., Ltd.
Yokota Bldg
Ginza 2-11-5
Chuo-Ku,
Tokyo, 104 Japan
Tel: (81) 3-3545-2360
FAX: (81) 3-3545-2351

Tokyo Electron Ltd.
TBS Broadcasting Center
5-3-6- Akasaka, Mingto-Ku
Tokyo, 163 Japan
Tel: (81) 3-5561-7212
FAX: (81) 3-5561-7389

Towa Elex Co., Ltd.
Unity Bldg. 5F
6-5 Nihonbashi Tomisawa-cho
Chuo-Ku, Tokyo 103
Japan
Tel: (81) 3-5640-1241
FAX: (81) 3-5640-1240

Varex Co., Ltd.
Nippo Shin-Osaka No. 2 Bldg.
1-8-33, Nishimiyahara,
Yodogawa-ku,
Osaka, 532 Japan
Tel: (81) 6-394-5201
FAX: (81) 6-394-5449

Marubun Corporation
8-1 Odenma-cho, Nihonbashi
Chuo-ku Tokyo, 103
Japan
Tel: (81) 3-3639-5210
FAX: (81) 3-3639-3727

Kaga Electronics Co., Ltd.
1-26-1 Otowa
Bunkyo-ky
Tokay 112, Japan
Tel: (81) 3-3942-6224
FAX: (81) 3-3942-6215

THE NETHERLANDS

SEI Rodelco BV
P.O.Box 6824
Takkebijsters 2
4802 HV Breda
The Netherlands
Tel: (31) 76-5722700
FAX: (31) 76-5710029

NEW ZEALAND

Advanced Component Dist.
P.O.Box 24-500
Royal Oak, Auckland
New Zeland
Tel: (64) 9-636-5984
FAX: (64) 9-636-5985

Advanced Component Dist
2 Heath St.
Wainviomata, Wellington
New Zealand
Tel: (64) 4-564-4902
FAX: (64) 4-564-2243

Advanced Component Dist.
Worchester Chambers
69 Worcester St.
Christchurch
New Zeland
Tel: (64) 3-379-3889
FAX: (64) 3-379-3072

NORWAY

BIT Elektronikk AS
Smedsvingen 4
P.O. Box 194
1360 Nesbru
Norway
Tel: (47) 66-98 13 70
FAX: (47) 66-98 13 71

POLAND

SEI NEljapex/Elbatex GmbH
Ul. Wilcza 50/52
PL-00679 Warszawa
Poland
Tel: (48) 2621 7122
FAX: (48) 2623 0605

PORTUGAL

TECMIC
Taguspark
Edificio Inovacao 1, no 222
2780 Oeiras
Portugal
Tel: (351) 1-422-8800
FAX: (351) 1-422-8809

RUSSIA

Scan
10/32 "B" Druzhby St.
117330 Moskva
Russia
Tel: (7) 095-1436641
FAX: (7) 095-9382247

Scan
42 Ordjonikidze
196143 St. Petersburg
Russia
Tel: (7) 812 299 7028
FAX: (7) 812 264 6000

Vostorg
3 Rue des Acacias
91370 Verrieres le Buisson
France
Tel: (33) 1-6920-4613
FAX: (33) 1-6011-5543

SINGAPORE

MEMEC Asia Pacific Ltd.
Singapore
Representative Office
10 Anson Road #14-02
International Plaza
Singapore 079903
Tel: (65)-222-4962
FAX: (65)-222-4939

Frontline Technologies Ptc. Ltd
2 Science Park Dr. #57
The Faraday
Singapore Science Park
Singapore 118222
Tel: (65)-779111
FAX: (65) 779-4455

SLOVAK REPUBLIC

Elbatex Slovensko
Svrcia ul
SK 84104 Bratislava
Tel: (42) 7 722 137
FAX: (42) 7 722 139

SLOVENIA/CROATIA

SEI Eljapex/Elbatex GmbH
Stegne 19, PO Box 19
SLO-61-117 Ljubljana
Tel: (61) 191-126-507
FAX: (61) 192-398-507

SOUTH AFRICA

Avnet - ASD
P.O. Box 3853
Rivonia 2128
South Africa
Tel: (27) 11-444-2333
FAX: (27) 11-444-1706

SOUTH AMERICA

DTS Ltda.
Rosas 1444
Santiago
Chile
Tel: (56) 2-697-0991
FAX: (56) 2-699-3316

Reycom Electronica S.A.
Bdo. de Irigoyen
972 Piso 2do "B"
1304 Buenos Aires
Argentina
Tel: (54) 1-304-2018
FAX: (54) 1-304-2010

Hitech El. Indl. Coml. Ltda.
Rau Branco de Moraes 489
Ch. Santo Antônio São Paulo
04718-010 - SP - Brazil
Tel: (55) 11-882-4000
FAX: (55) 11-882-4100

SOUTH KOREA

MEMEC Asia Pacific Ltd.
4FL, Je Woong Bldg.,176-11
Nonhyun-dong
Kangnam-ku
Seoul,135-010, South Korea
Tel: (82) 2-518-8181
FAX: (82) 2-518-9419

Hyunmyung Electronics Co. Ltd.
Dukwha Bld. 401
44-17, Seogyu-dong
Mapo-Gu Seoul
South Korea
Tel: (02) 2-277-7061
FAX: (02) 2-277-8470

SOUTHEAST ASIA

MEMEC Asia Pacific Ltd.
Unit No.2308-2319, Tower I
Metroplaza, Hing Fong Road,
Kwai Fong, N.T.
Hong Kong
Tel: (852) 410-2780
FAX: (852) 401-2518

SPAIN

SEI ADM Electronica, SA
Calle Tomas Breton,
no 50, 3-2 Planta
28045 Madrid
Spain
Tel: (34) 1-5304121
FAX: (34) 1-5300164

SEI ADM Electronica, SA
Mallorca 1
08014 Barcelona
Spain
Tel: (34) 3 426 6892
FAX: (34) 3-4250544
email:amdelec@lander.es

SEI ADM Electronica, SA
Herriko Gudarien, 8-1B
48200 Durango (Vizcaya)
Spain
Tel: (34)-4-6201572
FAX: (34)-4- 6202331

SWEDEN

DipCom Electronics AB
Box 1230
Torshamnsgatan 35
S-164 28 Kista
Sweden
Tel: (46) 8 752 24 80
FAX: (46) 8 751 3649

SWITZERLAND

Fenner Elektronik AG
Abteilung Bauteile
Gewerbestr. 10
CH-4450 Sissach
Switzerland
Tel: (41) 61 975 00 00
FAX: (41) 61 971 5608

TAIWAN

MEMEC Asia Pacific Ltd.
14F, No. 46, Lane 11
Kuang Fu North Road
Taipei
Taiwan R.O.C.
Tel: (886) 2-760-2028
FAX: (886) 2-765-1488w

TURKEY

Aztech Electronics Corp
524 42nd St. #200
Union City,
New Jersey 07087
Tel: (201) 867-2271
FAX: (201) 867-2162

Aztech Electronik
Farabi Sokak No.44/4
06690 Cankaya
Ankara
Turkey
Tel: (90) 312 467-9952
FAX: (90) 312467-9828

UK

Microcall Ltd.
17 Thame Park Road
Thame
Oxon OX9 3XD
UK
Tel: (44) 1-844-261939
FAX: (44) 1-844-261678

Cedar Technologies
Unit One Old Barns
Roycote Lane Farm
Milton Common
Oxfordshire OX9 2NZ
UK
Tel: (44) 1-844-278278
FAX: (44)1-844-278-378

Cedar Technologies
32 Enterprise House
Springkerse Business Park
Striling SK7 7UF
UK
Tel: (44) 1786-446221
FAX: (44) 1786-446223

Avnet EMG Ltd.
Jubilee House
Jubilee Road
Letchworth
Hertfordshire SG6 1QH
UK
Tel: (44) 1-462-480888
FAX: (44) 1-462-488567

MEMEC PLC
17 Thame Park Road
Thame
Oxfordshire OX9 3XD
UK
Tel: (44) 1-1844 261919
FAX: (44) 1-1844 261683

UNITED ARAB EMIRATES

Rezwan Trading Est.
P.O. Box 51973
Dubai
United Arab Emirates
Tel: (971) 4 279420
FAX: (971) 4 275741

The Programmable Logic Company℠

2100 Logic Drive
San Jose, CA 95124-3400
Tel: 1-408-559-7778
Fax: 1-408-559-7114
e-mail: hotline@xilinx.com
web: www.xilinx.com