



Boundary-Scan Emulator for XC3000 Series

XAPP 007 March 11, 1997 (Version 1.1)

Application Note by Bernie New

Summary

CLBs are used to emulate IEEE1149.1 Boundary Scan. The FPGA is configured to test the board interconnect, and then reconfigured for operation.

Specifications

Tests Supported:	EXTEST
Number of CLBs:	11 Core Logic
	1/2 to 1-1/2 per IOB
	1 per 3-State Control

Xilinx Family

XC3000A/XC3100A

Demonstrates

State Machine Design

Introduction

With more complex integrated circuits and more densely packed PC boards, testability is a major issue. One solution to the testability problem is boundary scan. The XC4000/XC5200-Series Field Programmable Gate Arrays (FPGAs) include boundary scan registers that meet the requirements of the IEEE1149.1 standard. While this standard provides for diagnostic testing and supports built-in self-test (BIST), one of its primary objectives is the testing of the interconnections between ICs. This is achieved using a mandatory external test mode, called EXTEST.

Although the XC3000-Series FPGAs do not contain boundary-scan registers, it is possible to configure an XC3000-Series device to emulate the EXTEST. This emulation consumes a significant amount of the FPGA resources (almost all in an XC3020A), and it is not suggested that boundary scan be built into a working design. However, because the RAM-based FPGA device is reconfigurable, it can be configured for board testing, and then reconfigured for operation.

The second mandatory test mode, SAMPLE/PRELOAD, has no meaning because the FPGA must be reconfigured for testing. It is not, therefore, supported by the emulator. However, the minimum 2-bit Instruction Register provides four instructions to select between two choices, the Test Data and Output Registers. For consistency with other boundary-scanned parts, one of these instructions could be used to create a dummy SAMPLE/PRELOAD mode. Functionally, this would duplicate the EXTEST with the Test Data Register selected.

Four pins must be dedicated to the Test Access Port (TAP). Due to external interconnection requirements, these pins can probably not be reused in the actual design.

The TAP Controller, Instruction Register, Bypass Register and Test Data Output Buffer together with miscellaneous logic require 11 Configurable Logic Blocks (CLBs). The CLB requirement for the Test Data Register depends upon

the number of Input/Output Blocks (IOBs) used, and how they are configured. Each requires between 1/2 and 1-1/2 CLBs, plus 1 CLB for each distinct 3-state control. While this may not allow every IOB to be bidirectional with an independent 3-state control, it will accommodate most designs.

A specific boundary-scan emulation must be created for each FPGA design. This comprises the 11 CLBs of core logic, which is common to all emulations, and a Test Data Register concatenated from four standard cells according to the output usage in the design. The output pins must be tied to match the design.

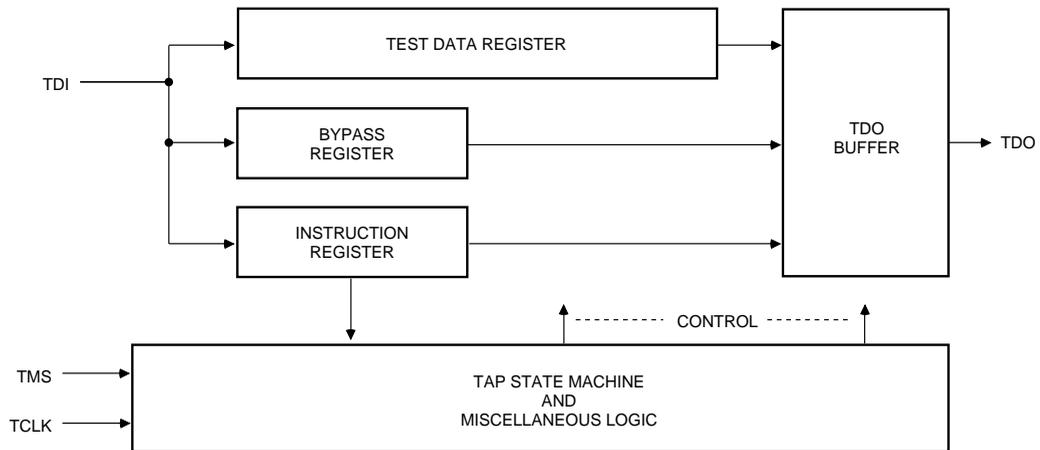
Operating Description

Overview

A block diagram of the IEEE1149.1 Boundary-Scan emulator is shown in [Figure 1](#). The four pins used are Test Data In (TDI), Test Data Out (TDO), Test Mode Select (TMS) and Test Clock (TCK). Operation of the emulator is controlled by the TAP state machine. This, in turn, is controlled by the serial TMS data stream.

Test data is shifted from TDI, through either the Instruction or Test Data/Bypass Registers, to the TDO. The choice between Instruction and Test Data/Bypass Registers is made according to the TMS bit-stream. The Test Data or Bypass Register is selected by the contents of the Instruction Register.

Before shifting commences, input data is captured by a parallel transfer into the appropriate shift register. After shifting is complete, new data is transferred in parallel into a second register where it is available to the outputs.



X1981A

Figure 1: IEEE 1149.1 Emulator Block Diagram

After configuration, the emulator automatically enters the power-up state required by the specification, and therefore, the Test Reset Signal (TRST) is not implemented. However, the polarity of all the registers is such that global reset may be used for this, if desired. The input pins used for TMS and TDI, and TRST if used, should be pulled up.

TAP Controller State Machine

The state diagram for the TAP Controller state machine is shown in Figure 2. This is implemented as two linked state machines, each using “one-hot” encoding.

The state-assignment table for this state machine is shown in Figure 3. Four state variables are used to create the states Test Logic Reset, Run Test/Idle, Select DR Scan, and Select IR Scan.

In the latter two states, the second state machine may be initiated. This has six state variables, and creates the states Capture (CAP), Shift (SH), Exit1 (E1), Pause (PAU), Exit2 (E2) and Update (UPD). These are qualified by the output of the first state machine to control the Test Data and Instruction Registers as necessary.

While this second state machine is operating, the first state machine is held in its current state. Following the Update state, the first state machine is forced to the appropriate state determined by TMS.

Figure 4 shows the schematic diagram of the state machine, and Figure 5 shows the equations that determine its next state. The only point of special interest is the use of clock enable in the first state machine. When the second state machine is in any of its first five states, the clock is dis-

abled in the first state machine, thereby saving complexity in the next-state logic.

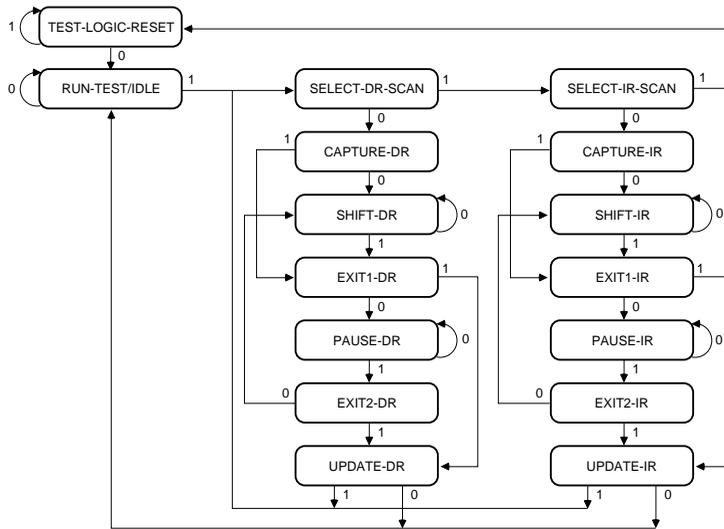
Note that the RTI flip-flop has inverters at its input and output. This causes the RTI state to be stored in active-Low form, such that this state is activated upon configuration or global reset. The pairs of flip-flops identified by circled numbers may be combined into single CLBs. The state machine requires six CLBs.

Instruction Register

The Instruction Register, shown in Figure 6, is two bits long, the minimum according to the specification. The shift register is enabled when the Instruction Register is selected by the state machine. In the Capture state, it is parallel loaded with 01(Binary), as required by the specification. It shifts data in the Shift state, and holds at other times.

Data from the shift register is clocked into the parallel register during the Update IR state. This parallel register is provided with a synchronous reset, which operates during the Test Logic/Reset state. The data in the parallel register is stored in inverted form, such that the Bypass Register (mandatory code: all ones) is selected after configuration or following a global reset.

For verification that the correct configuration has been loaded, additional bits could be added at the TDI end of the shift register. During Capture, these would be loaded with a code unique to the configuration. This would then be shifted out and become available as status bits. The parallel register need not be extended. Alternatively, the optional ID Code register could be implemented.



NOTE: The value shown adjacent to each state transition in this figure represents the signal present at TMS at the time of a rising edge at TCK.

X3208A

Figure 2: State Diagram for the TAP Controller

Test Data Register

The Test Data Register contains as many bits as there are used IOBs, plus one bit for each distinct 3-state control. This is concatenated from four types of 1-bit macros. Each of these is tied to a specific IOB, and the type of macro is determined by the function of the IOB.

The simplest macro, shown in Figure 7, is used for input pins. Data from the pad is loaded during Capture, when the Test Data Register is selected. This macro uses 1/2 CLB.

Figure 8 shows the second macro, which also requires 1/2 CLB. Although this macro may be used for 3-state and bidirectional outputs, it is most appropriate for simple outputs. Data from the shift register is clocked into the IOB output flip-flop by Update DR. During Capture, data from the pad is loaded into the shift register.

If the output is enabled, as is always the case in a non-3-state pin, the data captured is the contents of the parallel register, provided it is not corrupted by an interfering external signal. If the 3-state output is not enabled, data is always captured from an external source; or it is undetermined if an external source does not exist.

A better output macro is shown in Figure 9. The IOB flip-flop is replaced with a CLB flip-flop. During Capture, the parallel register is always read back into the shift register. However, this macro requires 1 CLB per output.

This macro should also be used to control 3-state outputs. When the design groups several outputs onto one 3-state

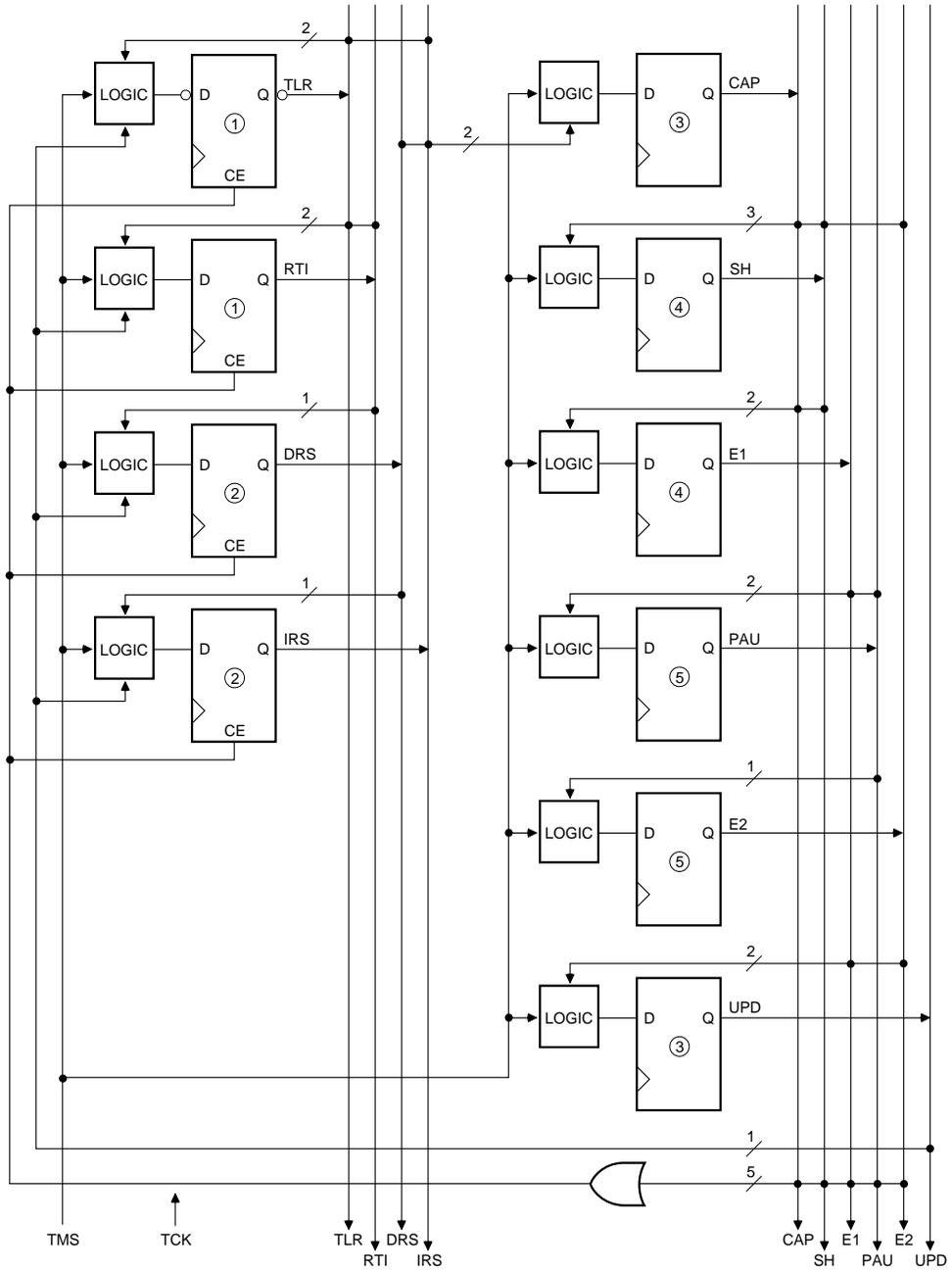
control, only one of these macros need be used to control the outputs.

The last macro, shown in Figure 10, is an enhanced macro for bidirectional pins. This macro operates in the same way as the enhanced 3-state output macro, but has an additional multiplexer that selects between the input data and the parallel register according to the 3-state control. This macro uses 1-1/2 CLBs.

	TLR	RT	DRS	IRS	CAP	SH	E1	PAU	E2	UPD
TEST LOGIC RESET	1	0	0	0	0	0	0	0	0	0
RUN TEST/IDLE	0	1	0	0	0	0	0	0	0	0
SELECT DR SCAN	0	0	1	0	0	0	0	0	0	0
CAPTURE DR	0	0	1	0	1	0	0	0	0	0
SHIFT DR	0	0	1	0	0	1	0	0	0	0
EXIT 1 DR	0	0	1	0	0	0	1	0	0	0
PAUSE DR	0	0	1	0	0	0	0	1	0	0
EXIT 2 DR	0	0	1	0	0	0	0	0	1	0
UPDATE DR	0	0	1	0	0	0	0	0	0	1
SELECT IR SCAN	0	0	0	1	0	0	0	0	0	0
CAPTURE IR	0	0	0	1	1	0	0	0	0	0
SHIFT IR	0	0	0	1	0	1	0	0	0	0
EXIT 1 IR	0	0	0	1	0	0	1	0	0	0
PAUSE IR	0	0	0	1	0	0	0	1	0	0
EXIT 2 IR	0	0	0	1	0	0	0	0	1	0
UPDATE IR	0	0	0	1	0	0	0	0	0	1

X1983

Figure 3: State Assignment for the TAP State Machine



X3209A

Figure 4: TAP State Machine (6 CLBs)

Bypass Register

The Bypass Register, shown in **Figure 11**, operates when the Data Register is selected. A zero is loaded during Capture, and data is shifted through the register during Shift. Otherwise, the register holds the data. The Bypass Register uses 1/2 CLB.

$$\begin{aligned} \text{TLR} &= \text{CE} \cdot [(\text{IRS} \cdot \overline{\text{UPD}} + \text{TLR}) \cdot \text{TMS}] + \overline{\text{CE}} \cdot \text{TLR} \\ \text{RTI} &= \text{CE} \cdot [(\text{TLR} + \text{RTI} + \text{UPD}) \cdot \text{TMS}] + \overline{\text{CE}} \cdot \text{RTI} \\ \text{DRS} &= \text{CE} \cdot [\text{TMS} \cdot (\text{RTI} + \text{UPD}) + \overline{\text{TMS}} \cdot \text{DRS}] + \overline{\text{CE}} \cdot \text{DRS} \\ \text{IRS} &= \text{CE} \cdot [\text{TMS} \cdot (\text{DRS} \cdot \overline{\text{UPD}}) + \overline{\text{TMS}} \cdot \text{IRS} \cdot \overline{\text{UPD}}] + \overline{\text{CE}} \cdot \text{IRS} \\ \text{CAP} &= (\text{DRS} + \text{IRS}) \cdot \overline{\text{TMS}} \cdot \text{CE} \cdot \overline{\text{UPD}} \\ \text{SH} &= (\text{CAP} + \text{E2} + \text{SH}) \cdot \overline{\text{TMS}} \\ \text{E1} &= (\text{CAP} + \text{SH}) \cdot \text{TMS} \\ \text{PAU} &= (\text{E1} + \text{PAU}) \cdot \text{TMS} \\ \text{E2} &= \text{PAU} \cdot \text{TMS} \\ \text{UPD} &= (\text{E1} + \text{E2}) \cdot \text{TMS} \\ \text{CE} &= \sim (\text{CAP} + \text{SH} + \text{E1} + \text{PAU} + \text{E2}) \end{aligned}$$

X6064

Figure 5: TAP State Machine Logic Equations

TDO Buffer

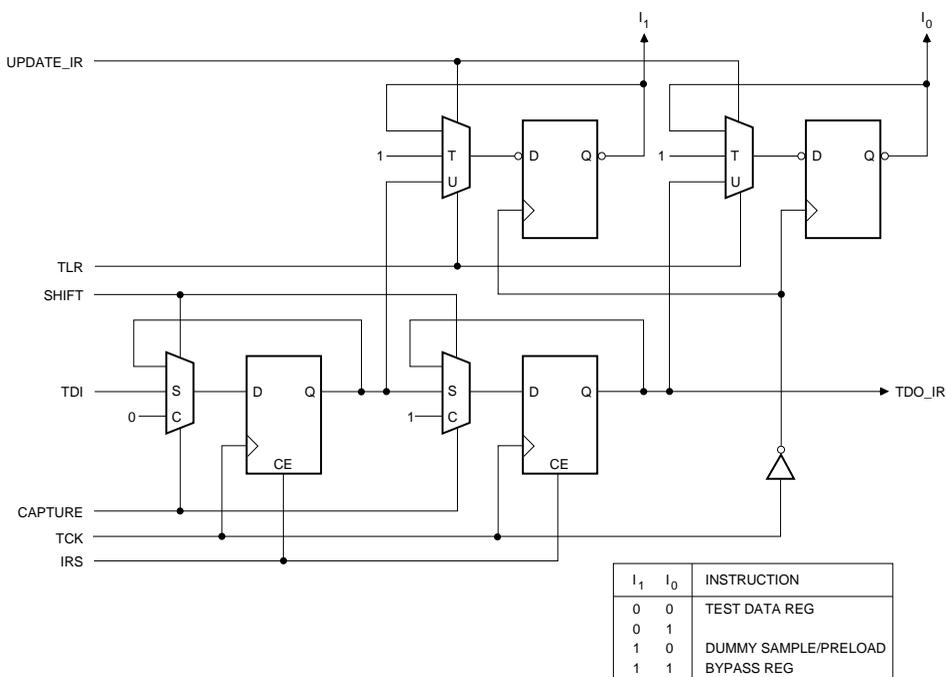
Figure 12 shows the Test Data Output Buffer. Data is selected from the Instruction Register, the Test Data Register or the Bypass Register, and clocked out on the negative edge of TCK. The 3-state output is only enabled during Shift. The TDO Buffer uses 1 CLB.

Miscellaneous Logic

The Miscellaneous Logic, shown in **Figure 13**, uses 1-1/2 CLBs. Its function is to combine states from the state machine to enable various registers.

Most registers in the emulator are clocked by TCK (or its inverse) and controlled by enables. The only exception is the IOB flip-flop used in the simple output macro of the Test Data Register. Since IOB flip-flops have no clock enable, a gated clock must be used.

Rather than ANDING the clock with a gating signal, a flip-flop is used. During Update when the Data Register is selected, Update DR is clocked High on the negative edge of TCK. The state machine can only remain in this state for one period, and this defines the length of the update clock. The ACLK buffer is used to distribute the Update DR clock.



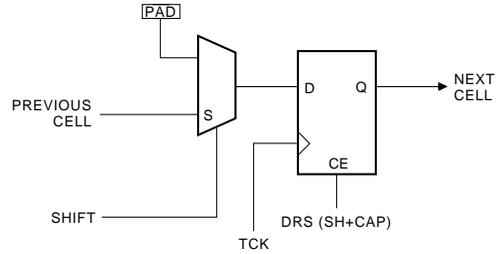
X3210A

Figure 6: Instruction Register (2 CLBs)

Implementation Notes

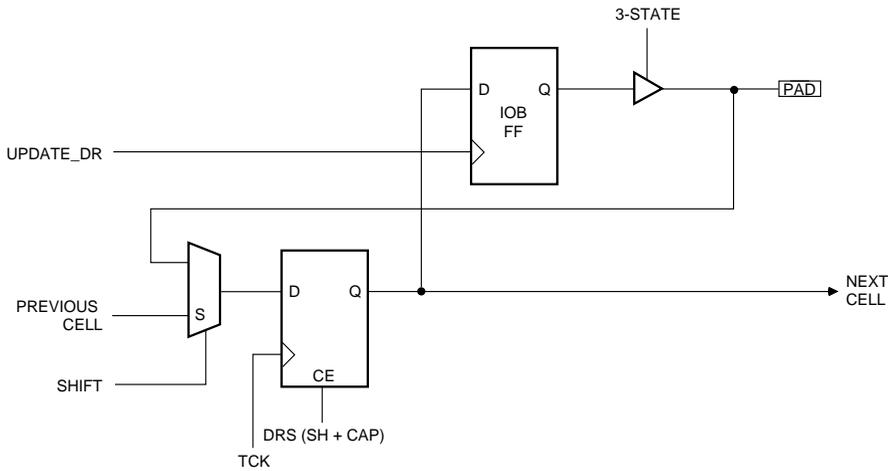
The design support for the XC3000-Series Boundary-Scan Emulator comprises five soft macros. The first of these contains the 11 CLBs of core logic, including the Test Access Port. Location constraints must be added to the schematic to specify the desired location of the TAP input and output pins.

The remaining macros support different types of input/output pins. These macros need to be selected according to the input/output utilization, and connected to form a shift register between the data pins of the first macros. Again, a location constraint must be added to each macro, specifying the pin with which it is associated.



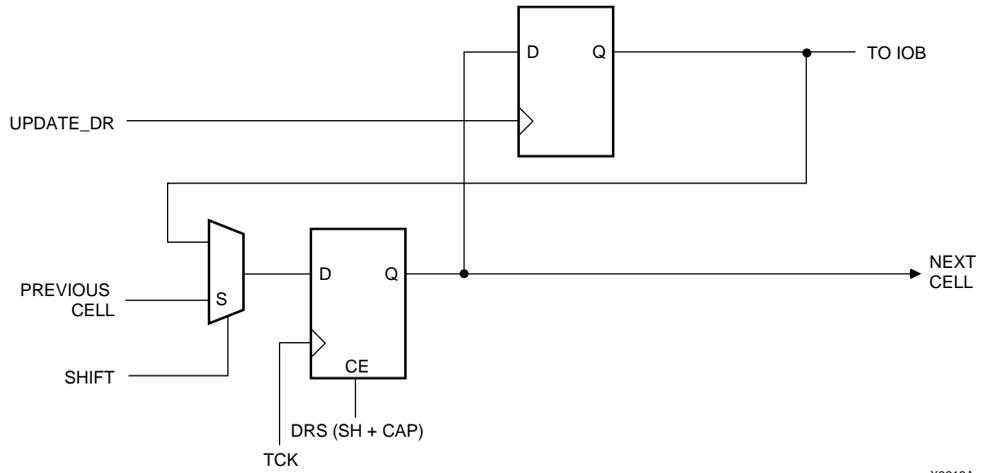
X3211A

Figure 7: Data Register Input Cell (1/2 CLB)



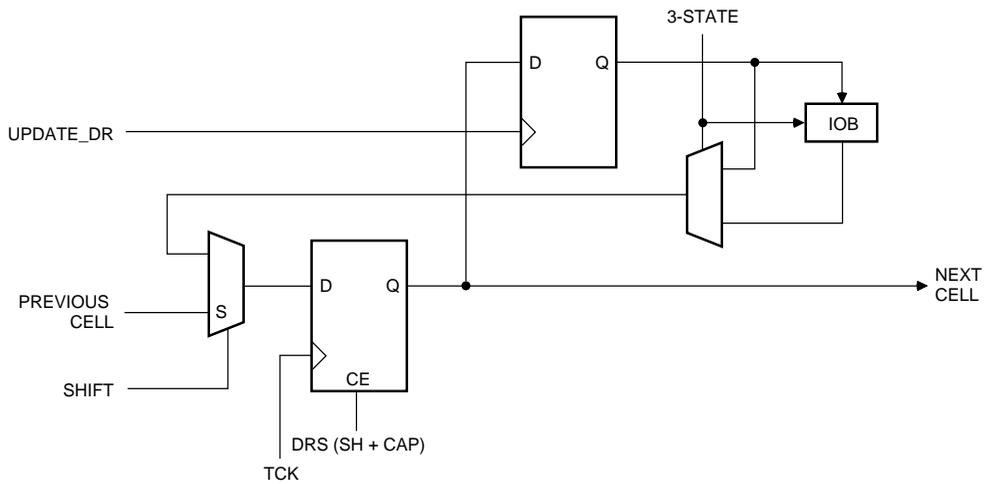
X3212A

Figure 8: Data Register Output/Bidirectional Cell (1/2 CLB + IOB Flip-Flop)



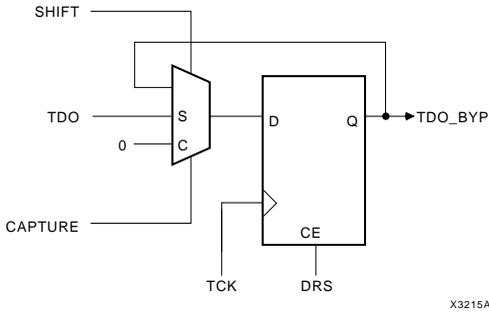
X3213A

Figure 9: Data Register Enhanced Output/3-state Cell (2 bits/2 CLBs)



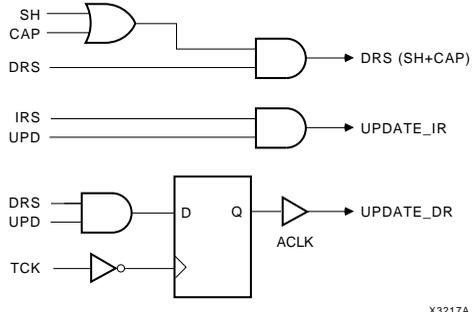
X3214A

Figure 10: Data Register Enhanced Bidirectional Cell (1-1/2 CLBs)



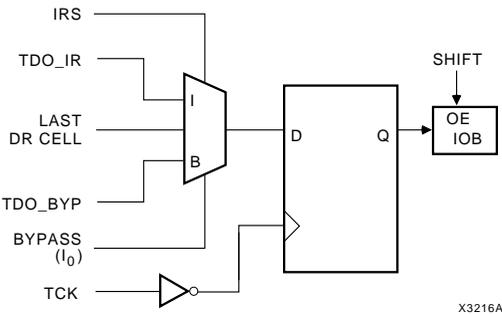
X3215A

Figure 11: Bypass Register (1/2 CLB)



X3217A

Figure 13: Miscellaneous Logic (1-1/2 CLBs)



X3216A

Figure 12: TDO Buffer (1 CLB)



Headquarters

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
U.S.A.
Tel: 1 (800) 255-7778
or 1 (408) 559-7778
Fax: 1 (800) 559-7114
Net: hotline@xilinx.com
Web: http://www.xilinx.com

North America

Irvine, California
(714) 727-0780

Englewood, Colorado
(303)220-7541
Sunnyvale, California
(408) 245-9850
Schaumburg, Illinois
(847) 605-1972
Nashua, New Hampshire
(603) 891-1098
Raleigh, North Carolina
(919) 846-3922
West Chester, Pennsylvania
(610) 430-3300
Dallas, Texas
(214) 960-1043

Europe

Xilinx Sarl
Jouy en Josas, France
Tel: (33) 1-34-63-01-01
Net: frhelp@xilinx.com
Xilinx GmbH
Aschheim, Germany
Tel: (49) 89-99-1549-01
Net: dlhelp@xilinx.com
Xilinx, Ltd.
Byfleet, United Kingdom
Tel: (44) 1-932-349401
Net: ukhelp@xilinx.com

Japan

Xilinx, K.K.
Tokyo, Japan
Tel: (03) 3297-9191

Asia Pacific

Xilinx Asia Pacific
Hong Kong
Tel: (852) 2424-5200
Net: hongkong@xilinx.com

© 1996 Xilinx, Inc. All rights reserved. The Xilinx name and the Xilinx logo are registered trademarks, all XC-designated products are trademarks, and the Programmable Logic Company is a service mark of Xilinx, Inc. All other trademarks and registered trademarks are the property of their respective owners.

Xilinx, Inc. does not assume any liability arising out of the application or use of any product described herein; nor does it convey any license under its patent, copyright 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. cannot assume responsibility for the use of any circuitry described other than circuitry entirely embodied in its products. Products are manufactured under one or more of the following U.S. Patents: (4,847,612; 5,012,135; 4,967,107; 5,023,606; 4,940,909; 5,028,821; 4,870,302; 4,706,216; 4,758,985; 4,642,487; 4,695,740; 4,713,557; 4,750,155; 4,821,233; 4,746,822; 4,820,937; 4,783,607; 4,855,669; 5,047,710; 5,068,603; 4,855,619; 4,835,418; and 4,902,910. Xilinx, Inc. cannot assume responsibility for any circuits shown nor represent that they are free from patent infringement or of 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.