Introduction *1*

This manual describes the architecture of the IQ family of integrated circuit chips, which includes the IQ320, IQ240B, IQ160, IQ128B, IQ96, IQ64B, IQ48, and IQ32B part types.

Each IQ device implements a non-blocking switch matrix. Each signal line in the switch matrix is connected to a programmable I/O port. Each I/O port can be programmed for its I/O attributes as an input, output, or bidirectional buffer. By programming the switch matrix you can connect an I/O port to one or more other I/O ports. The number of I/Os in the IQ family range from 32 to 320.

The IQ family supports the JTAG standard for programming and testing these devices. JTAG comprises a serial five-signal bus for use during a boundary scan test of digital circuit boards. The five signals are

- Test Clock (TCK)
- Test Mode Select (TMS)
- Test Data In (TDI)
- Test Data Out (TDO)
- Test Reset (TRST*)

TCK clocks data synchronously in and out of TDI and TDO while TMS is used to control the system's state. TRST* resets all the chips on the JTAG bus.

For more details on the IQ family, refer to the IQ Family Data Sheet.

PAGE(S) INTENTIONALLY BLANK

IQ Family Architecture 2

All the IQ family devices are based on a common architecture. The main difference between the different members of the family is in the number of programmable I/O ports on the device.

A conceptual block diagram of a simplified IQ device with eight I/O ports is shown in Figure 2-1. The switch matrix of the IQ family can connect any I/O port to any other I/O port(s) on the chip. This array consists of a triangular array of transistor switches. Each transistor switch controls the connection between a unique pair of I/O ports.

Each transistor switch is controlled by a single SRAM cell. You can program each SRAM cell with a "1" or "0" to set switches ON or OFF. In addition, you can also set up IQ devices to connect multiple ports.

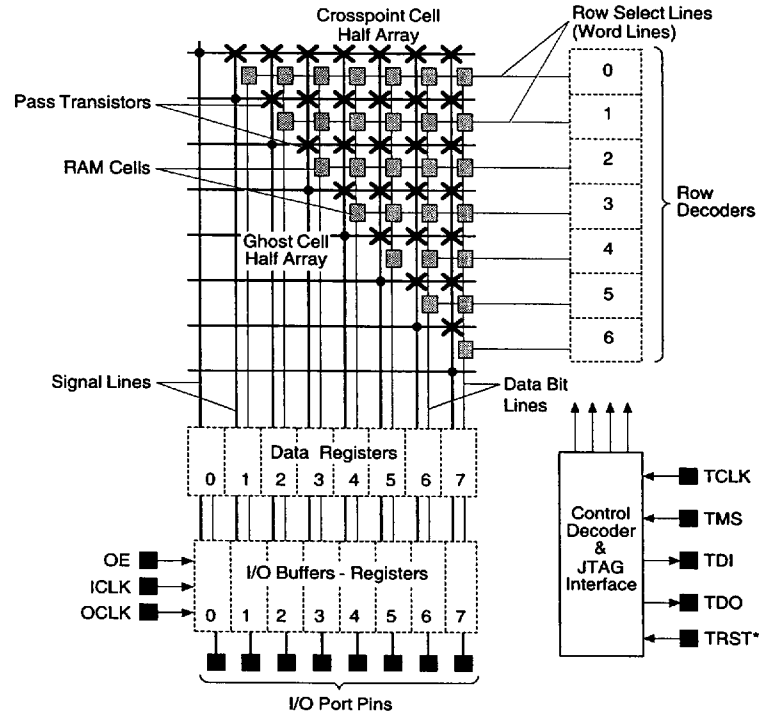

Figure 2-1 IQ Family Block Diagram

The switch matrix that establishes connections between I/O ports in this mode can be viewed as an N x N matrix. There are N horizontal and N vertical lines in this matrix. Each horizontal line in this matrix is permanently (hard) connected to a corresponding vertical line with a solid dot (•), as shown in Figure 2-2. It is also connected with a transistor switch to other vertical lines.

The SRAM cells controlling the transistor switches in the matrix are organized as a square array of N x N. There are N rows (words), each with N columns (bits per word). About 50% or $0.5*N*[N-1]$ locations in the array contain transistor switches and corresponding SRAM cells.

Two I/O ports can be connected (disconnected) by turning ON (OFF) the transistor switch that connects the corresponding horizontal and vertical lines. The switch and the corresponding SRAM cell required to connect I/O ports P_i and P_j has two possible locations:

- bit j of word i

or

- bit i of word j

The location is shown as hollow rectangles in Figure 2-2. Only one location contains a real switch and SRAM cell while the other is a ghost location and has no switch or SRAM cell. To affect a connection change, data must be written to the real SRAM cell.

"Programming the Crossbar Array (Xarray)" on page 20 provides details on how to determine the real SRAM cell location

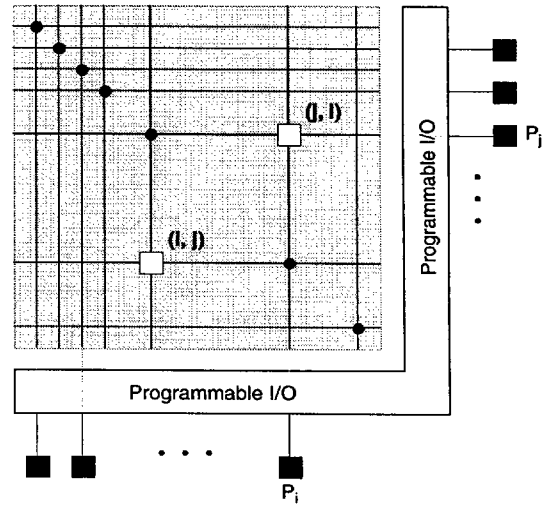

Figure 2-2 Switch Matrix Organization

Note – I-Cube devices whose part number has a B suffix are based on a larger device whose part number does not have a B suffix. For example, IQ240B is based on IQ320. For the purposes of Register Level programming, the B

devices should be viewed as the non-B devices they are based on. See the section titled "Programming IQ Devices with ..B Suffix (IQ240B, IQ128B, IQ64B,IQ32B)" on page 23 for more details.

Each IQ device provides a different number of external I/O ports. The ports are numbered from 0 to *M*-1. The value of *M* for each device is shown in Table 2-1.

Table 2-1 Number of I/O ports for Different IQ Devices

Port Type	I/O Ports on Package(M)	Words and Bits per word in Matrix(N)
IQ320	320	320
IQ240B	240	320
IQ160	160	160
IQ128B	160	160
IQ96	96	96
IQ64B	64	96
IQ48	48	48
IQ32B	32	48

The I/O ports have a number of programmable attributes. One of the attribute is FSEL (F3~F0, 4 bits, Function Select) which determines the mode of the I/O buffer. Table 2-2 describes all allowable values for this attribute.

Table 2-2 Values for FSEL

Binary Value	Mnemonic	Description
0000_2	NC	No Connect (Default)
0001_2	BR	Bus Repeater
0011_2	OP	Output
0100_2	F0	Force 0 Output
0101_2	F1	Force 1 Output
0110_2	IN	Input

Table 2-2 Values for FSEL

Binary Value	Mnemonic	Description
1000_2	A0	Array Side Force 0
1001_2	A1	Array Side Force 1
1011_2	RO	Registered Output
1100_2	NB	Non-buffered
1110_2	RI	Registered Input

Table 2-3 describes all values for the VLPU attribute (V1~V0, 2 bits, Output Voltage Level, and Additional Pull-Up Current). This function sets the output voltage level V_{OH} to TTL or CMOS. You can specify additional pull-up current when you set the CMOS output level.

Table 2-3 Values for VLPU

Binary Value	Mnemonic	Description
00_2	TTL	TTL output level
01_2	CLO	CMOS output level; low (1.5mA) additional pull-up current
10_2	CNO	CMOS output level; no additional pull-up current
11_2	CHI	CMOS output level; high (13.5mA) additional pull-up current

PAGE(S) INTENTIONALLY BLANK

JTAG Overview 3

JTAG is a five-wire serial bus for scan test of digital circuit boards. The JTAG bus is synchronously controlled by a host processor. The five JTAG signals are:

- Test Data In (TDI)
- Test Data Out (TDO)
- Test Mode Select (TMS)
- Test Clock (TCK)
- Test Reset (TRST*)

TCK is used to clock data synchronously in and out of TDI and TDO, while the TMS signal is used to control the system's states. TRST* is used to reset all the chips on the JTAG bus. Figure 3-1 shows a state diagram of the JTAG controller state machine.

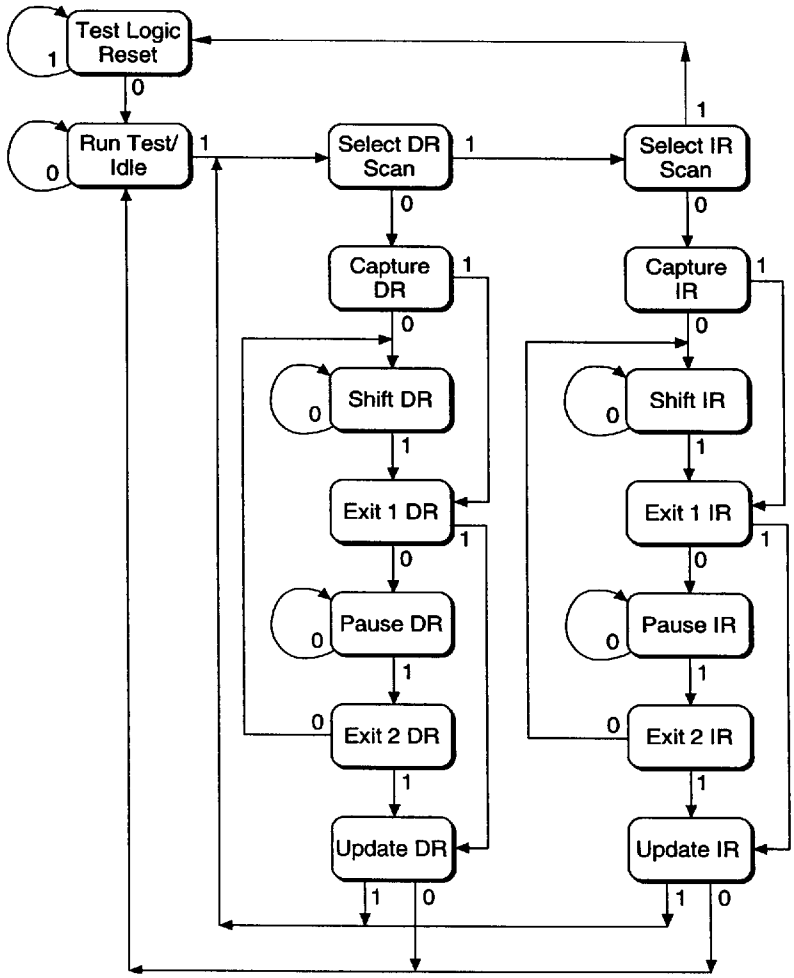

Figure 3-1 State Diagram of JTAG Controller State Machine

Properly sequencing the TMS line allows the system to access the instruction register (16-bit wide) or one of the data registers, as shown in Figure 4-2.

The data register is selected through the instruction register. Some instructions do not select a register but strobe some system action, such as resetting the device. Each register (instruction and data) is accessed by a read-modify-write process; the process steps are called

- CAPTURE
- SHIFT
- UPDATE

When shifting in new data using TDI, the previous data can be captured and shifted out using TDO, as appropriate. Regardless of the state of the JTAG controller, clocking the controller for five cycles and holding TMS at 1 returns it to the Test Logic Reset (TLR) state.

The TDIs and TDOs of all the chips on the board are usually connected in a series. The chips appear as one long shift register to the system controller. To keep data lengths manageable, individual chips can also be instructed to bypass by connecting the input to the output through a one bit BYPASS shift register resulting in shorter patterns and faster operations.

For more information, you can find detailed descriptions of JTAG in the *IEEE Standard Test Access Port and Boundary-Scan Architecture* document, IEEE Std. 1149.1-1990 and in "The Test Access Port and Boundary-Scan Architecture" by Colin M. Maunder and Rodham E. Tulloss (Eds), IEEE Computer Society Press.

Note – A common convention in standard JTAG documentation when specifying a literal bit string for TMS or TDI makes the right-most (the least significant) bit the one that is shifted in first. Similarly, in a TDO string, the right-most bit is the one that is shifted out first. For example, the TDI input string specification 0001111 shifts in four 1s first followed by three 0s.

PAGE(S) INTENTIONALLY BLANK

Programming IQ Devices 4

This chapter describes programming a chain of IQ devices by using four external signals, TDI, TDO, TMS, and TCK, as follows:

- TDI is the input that carries a sequence of 0s and 1s to be serially shifted into the chain.
- As a bit sequence is shifted in, another bit stream is shifted out to the TDO pin.
- TMS, which is connected in parallel (broadcast) to all devices, controls the state transitions for the JTAG controller inside each IQ device.
- TCK is the input clock that synchronizes the operation of the other three signals.

Programming IQ Devices in a Series

To program and test a series of IQ devices, you need to perform the following steps:

1. **Generate the sequence of 0s and 1s to be scanned into the TDI and TMS pins by using software.**

   The sequence shifted into TMS steers the controllers through various states. The sequence scanned into TDI represents either JTAG instructions or contents for the various data registers.

2. **As soon as the two input bit streams, one for TMS and the other for TDI, are generated, scan in the bit sequences into the device chain synchronized by the TCK clock.**

   While this task is taking place, another sequence of 0s and 1s is scanned out to TDO. Some hardware mechanism is needed to capture this output stream.

3. **Analyze the resulting TDO bit stream by using software to test and debug.**

   The steps of generating input patterns, downloading or uploading the patterns, and analyzing the output patterns can happen in a clock-by-clock fashion, in bursts of a small number of clocks, or in batch fashion wherein the entire collection of devices can be programmed at once.

Figure 4-1 illustrates programming a chain of IQ devices with the four external signals.

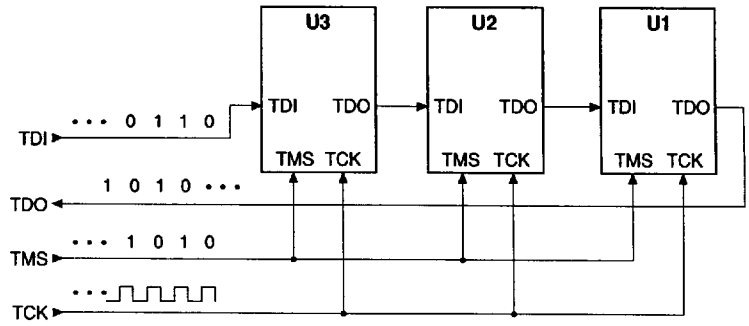

Figure 4-1 Programming a Chain of IQ Devices

Generating the Bit Sequence

You can generate bit sequences for TDI and TMS signals and analyze a TDO bit stream by:

- Using IDSComp or IDSPro
- Using the driver primitives library
- Writing software to program devices

Using IDSComp or IDSPro

This method is best suited to applications that require infrequent, off-line generation of configuration bit streams. It relies on using either IDSComp or IDSPro, executable files, that can run on a PC or workstation and generate a disk file containing bit streams for TDI and TMS.

The ASCII netlist file, an input file, specifies the number and order of chaining IQ devices in a design. For each IQ device, the file describes the desired connectivity to I/O ports and the attributes of each port. The output bit stream file is a fully packed, binary file that you can downloaded to your target system.

IDSComp or IDSPro are available as part of the I-Cube Development Systems Products IDS100 and IDS200.

Using the Driver Primitives Library

For embedded applications that require dynamic, incremental reconfigurations, I-Cube provides a primitive library of functions. The library is available in C source code as part of the IDS500 product.

You can customize the library for your application. The library source code is then compiled and linked to your final application software. You can call functions in the library to program the SRAM calls in the crossbar array and the registers in the I/O buffers.

The bit stream resulting from a series of such function calls is maintained in the memory of the on-board host processor. If hardware exists for shifting JTAG data in and out of a chain of IQ devices, the bit streams can be downloaded immediately. The bit stream can also be saved for later batch downloading.

Writing Software to Program Devices

This method relies on primitive interaction with a set of IQ devices in a design. You need to be familiar with the general operation of a JTAG-compliant device to use this method. Based on the information contained in this document, you can write your own code to generate the 0s and 1s

required to program the devices, as illustrated in the examples provided in Chapter 6, "Programming Examples." You do not need any software from I-Cube to perform this task.

You can interact with IQ devices using this method. If hardware is available for scanning bit sequences in and out of IQ devices, a two-way, interactive operation can be carried out. A typical sequence is to shift in a JTAG instruction followed by a bit stream for one of the data registers. The TDO pattern which was shifted out is then analyzed. Working at such a low level is similar to programming in machine language.

JTAG Registers

A programming view of an IQ device is shown in Figure 4-2. The device contains the following registers:

- Instruction register
- Bypass register
- Device ID register
- Crossbar array data register (Xarray DR)
- I/O buffer data registers (IOB DR)
- Mode control and status data register (MOD DR)

These registers are described in the following sections.Figure 4-2 illustrates the JTAG programming registers.

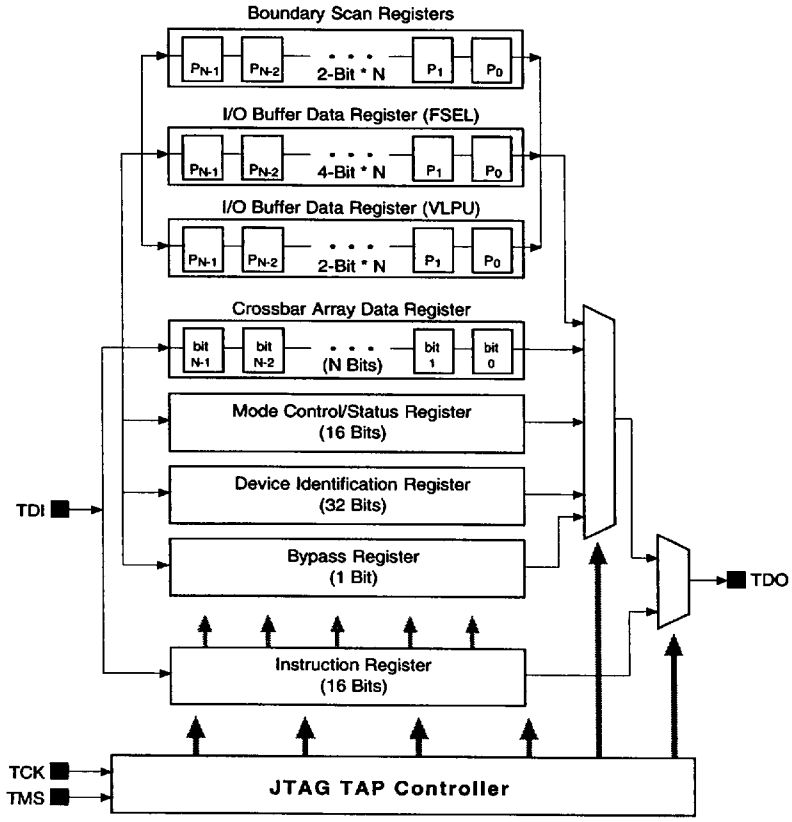

Figure 4-2 IQ Device JTAG Programming Registers

Instruction Register

The instruction register (IR) is 16-bit wide. It is used to select a data register (DR) to read and/or write and start various actions.

Bypass Register

The bypass register is a 1-bit register that connects TDI to TDO with a nearly direct connection. It is useful for accessing one specific device in a serial chain of IQ devices. This is a mandatory JTAG feature.

Device ID Register

The 32-bit device identification register identifies a particular device on the scan path, a standard JTAG feature. The following ID codes are for different devices in this family:

Table 4-1 Device ID Codes

Devices	ID Code
IQ320	$0000289F_{16}$
IQ240B	$0000289F_{16}$
IQ160	$0000389F_{16}$
IQ128B	$0000389F_{16}$
IQ96	$0000589F_{16}$
IQ64B	$0000589F_{16}$
IQ48	$0000689F_{16}$
IQ32B	$0000689F_{16}$

Crossbar Array Data Register (Xarray DR)

The crossbar array data register stores the bit pattern to be copied into the crossbar switch SRAM cells during the Update-DR state and also holds the bits copied out of the switch cells during the Capture-DR state. The specific word of the RAM array to write/read is contained in the instruction register. This data register is *N* bits long where *N* is defined for specific devices as follows:

Table 4-2 Data Register Sizes

IQ320	IQ240B	IQ160	IQ128B	IQ96B	IQ96	IQ64B	IQ48	IQ32B
320	320	160	160	160	96	96	48	48

I/O Buffer Data Registers (IOB DR)

There are two I/O buffer data registers of varying lengths. They store the bit streams to be copied into the I/O buffer control blocks during the Update-DR state, and hold the bits copied out of the I/O buffer control block during the Capture-DR state.

I/O buffer data registers shown in Table 4-3 are user-accessible.

Table 4-3 I/O Buffer Data Registers

Name	Function	Length
FSEL	Selects I/O Function	4 * N
VLPU	Selects I/O Output Voltage Level and Additional Pull-up Current	2 * N

The value of N is shown in Table 2-2.

Mode Control and Status Data Register (MOD DR)

The 16-bit register stores the settings of several mode control flags. Most of these controls are used for internal operation. You can only change the control bit that enables or disables the RapidConnect™ mode.

JTAG Instructions

The Instruction Register (IR) selects the data register and initiates any test operations. The available JTAG instructions are shown in the following:

0000_{16}	0 000	0000	0000	0000	**Extest**
8001_{16}	1 000	0000	0000	0001	**Extest** (Same instruction; Read Back)
0005_{16}	0 000	0000	0000	0101	**Sample / Preload**
$8C01_{16}$	1 000	1100	0000	0001	**FSEL**, Function Select, Read/Write
0009_{16}	0 000	0000	0000	1001	**VLPU**, V_{OH} Level & Pull-Up Current Read/Write
$xxxx_{16}$	P 010	aaaa	aaaa	aa01	Cross-Point Array DR, Read/Write
$xxxx_{16}$	P 011	aaaa	aaaa	aa01	Cross-Point Array DR, Write Only
$FFE1_{16}$	1 111	1111	1110	0001	Device ID Reg. (Set by **JTAG** Reset)
$FFD1_{16}$	1 111	1111	1101	0001	Control/Status Register
$FFFF_{16}$	1 111	1111	1111	1111	Bypass Register

The P bit is even parity for the instruction register. The bit is ignored for now. There are two instructions for writing the crossbar array DR. The first instruction does both a read (DR Capture) and a write (DR Update). The second instruction suppresses the read operation during the Capture state. In these instructions the 10-bit address (designated by **aaaa aaaa aa**) selects one of the *N Xarray* words, word 0 for I/O port 0, word 1 for I/O port 1, etc.[1]

Programming the Crossbar Array (Xarray)

The crossbar array consists of *N* lines running horizontally and *N* lines running vertically, as shown in Figure 2-1. Each pair of one horizontal line and one vertical line that intersect along the main diagonal is permanently connected at the main diagonal intersection and is connected to one external I/O port.

Each non-diagonal intersection can be connected using a transistor switch that is controlled by an SRAM cell. Thus the `Xarray` can be thought of as an *NxN* SRAM block of *N* words. Each word represents one horizontal line and is *N* bits long. The notation (i, j) is used to designate the SRAM cell at the

1. This is true for the IQ devices without the B suffix. See the section called "Programming the Crossbar Array (Xarray)" for details.

intersection of horizontal line i and vertical line j or bit j of word i. To connect line i with line j (where i is not equal to j), one needs to turn on either the bit (i, j) or the bit (j, i).

Only one of these locations contains a switch. The location without the switch is referred to as the ghost location. Figure 2-1 illustrates this by presenting the upper-right triangle as real switches and the lower-left triangle as ghost locations. In the actual implementation, the presence or absence of switches forms a complex pattern and is determined by using a table look-up process, explained in Appendix A, "Look-up and Port Mapping Tables".

The *Xarray* is programmed by writing *N* words of *N* bits each, programming (turning the bit to a 1 or a 0) both the real switches and the ghost switches, using the *N*-bit *Xarray* DR. In each DR scan, the address of the affected word is contained in the current contents of the IR register. The *Xarray* contents may also be read back as *N* words of *N* bits each. If location (i, j) is real (not a ghost location) and has the value 1, then ports p_i and p_j are connected. Writing to a ghost location has no effect, reading a ghost location always returns a 0.

Programming the I/O Buffer Control Registers

Two JTAG instructions are available for programming the two I/O buffer registers, FSEL and VLPU, respectively. Refer to Chapter 2, "IQ Family Architecture" for the allowable values of these attributes.

For example, to program the I/O function FSEL, the FSEL instruction is first scanned in. Then a DR scan of *N**4 bits is performed by supplying 4 bits of data for each I/O port on the die. Depending on the device, some or all I/O ports on the die are brought out to the package pins. See the section "Programming IQ Devices with ..B Suffix (IQ240B, IQ128B, IQ64B,IQ32B)" for more information.

To change the I/O function of one I/O port, all *N**4 bits of DR data must be shifted in. Those ports whose I/O functions are not changed should be rewritten using their original values.

4

IQ Device Initialization and Mode Control

When the system is first powered on, the programming bits in I/O buffers are cleared, i.e., all I/O Ports are set to FSEL = NC, VLPU = TTL attributes, and All bits in the Mode Control Register are reset to "0". The *Xarray* SRAM cells are **NOT** cleared and may contain arbitrary values.

To prepare the device(s) for proper operation, an initialization sequence, that clears all *Xarray* SRAM cells and sets the appropriate Mode Control Register bits must first be downloaded. At the end of the initialization sequence, the JTAG controller must be brought to the Run Test/Idle state. For the Mode Control Register, all bits except the RapidConnect and Turbo bits, indicated as "r" and "t" respectively, MUST be set as shown in Table 4-4.

Table 4-4 Mode Control Register Contents for After Initialization

Devices	Mode Register Contents
IQ320	0000 0000 0t00 0r00 (LSB)
IQ240B	0000 0000 0t00 0r00 (LSB)
IQ160	0000 0000 0t11 1r00 (LSB)
IQ128B	0000 0000 0t11 1r00 (LSB)
IQ96	0000 0000 0t00 0r00 (LSB)
IQ64B	0000 0000 0t00 0r00 (LSB)
IQ48	0000 0000 0t00 0r00 (LSB)
IQ32B	0000 0000 0t00 0r00 (LSB)

Depending on the desired mode of operation, the RapidConnect and Turbo bits should be set as shown in Table 4-5.

Table 4-5 RapidConnect and Turbo Bit Settings

Devices	RapidConnect Bit (r)	Turbo Bit (t)
IQ320	1 = Enable, 0 = Disable	1 = Enable, 0 = Disable
IQ240B	1 = Enable, 0 = Disable	1 = Enable, 0 = Disable
IQ160	1 = Enable, 0 = Disable	1 = Enable, 0 = Disable
IQ128B	1 = Enable, 0 = Disable	1 = Enable, 0 = Disable
IQ96	1 = Enable, 0 = Disable	1 = Enable, 0 = Disable
IQ64B	1 = Enable, 0 = Disable	1 = Enable, 0 = Disable
IQ48	1 = Enable, 0 = Disable	0 = Enable, 1 = Disable
IQ32B	1 = Enable, 0 = Disable	0 = Enable, 1 = Disable

Recommended Setting for Turbo Mode

The Turbo mode allows you to set the thresholds at which logic transitions are detected by the IQ device. When the Turbo mode is disabled, the transition thresholds are set roughly at the midpoint of the voltage range. When Turbo mode is enabled, the high-to-low detection point is set above the midpoint while the low-to-high detection point is set below the midpoint. This results in shorter propagation delays. In typical usage, the Turbo mode is enabled.

Note – If the device is brought to ***Test Logic Reset*** state, either by holding the TRST* pin low or by applying a "11111" sequence to TMS, all the programming bits in I/O buffers and Mode Control register are cleared. Once JTAG programming has begun, the JTAG state machine should **NOT** be brought back to the ***Test Logic Reset*** state unless complete resetting of the device is desired. The device can be brought back to the **Run Test/Idle** state between instructions.

Programming IQ Devices with ..B Suffix (IQ240B, IQ128B, IQ64B,IQ32B)

Devices with a ...B suffix are bond-out versions of another IQ device die with some I/O ports and crossbar array lines left unused.

- IQ240B is a bond-out of the IQ320 die
- IQ128B is a bond-out of the IQ160 die
- IQ64B is a bond-out of the IQ96 die

- IQ32B is a bond-out of the IQ48 die

For programming purposes, an I/O port of such a device must first be mapped to a corresponding port of the original die from which it was derived. You can do this using the mapping tables shown in Appendix A, "Look-up and Port Mapping Tables". All other bit stream generation programming follows the procedures described earlier in this chapter.

In addition, the I/O ports on the die that are not bonded to the package pin should be configured as "A0". All *Xarray* SRAM cells, including those corresponding to the non-bonded I/O ports must be cleared.

Bit Shifting Order for TDI data

The shifting of the TDI data bits for each instruction are shown below. The bit indicated on the *right* is shifted in first.

For Instruction Register

Bit_{15}, ..., Bit_2, Bit_1, Bit_0

For Bypass Register

Bit_0

For Device ID Register

Bit_{31} , ..., Bit_2, Bit_1, Bit_0

For Crossbar Array Data Register (Xarray)

Bit_{n-1} , ..., Bit_2, Bit_1, Bit_0

For I/O Buffer Data Registers (FSEL)

$Port_{n-1}Bit_3$, $Port_{n-1}Bit_2$, ..., $Port_0Bit_1$, $Port_0Bit_0$

For I/O Buffer Data Registers (VLPU)

$Port_{n-1}Bit_1$, $Port_{n-1}Bit_0$, ..., $Port_0Bit_1$, $Port_0Bit_0$

For Mode Control Data Register

Bit_{15} , ..., Bit_2, Bit_1, Bit_0

Reading Back Device Status

The JTAG instructions for programming the I/O buffers and Xarray are Read/Modify/Write instructions. This means that during the DR Scan, the DR-Update always takes place. If you are interested in only reading back the contents of the various IOB and Xarray data registers, you must ensure that the same contents that are captured during Capture-DR for shifting out over TDO are written back during Update-DR.

This can be achieved by capturing the data register contents during Capture-DR, shifting those out over the TDO pin during Shift-DR, and shifting the same data back into the device over the TDI pin while the state machine is still in Shift-DR. This will ensure that the data that is written into the data register during Update DR is the same as before, avoiding any inadvertent changes to the device configuration.

PAGE(S) INTENTIONALLY BLANK

Boundary Scan Testing 5

This chapter describes how the boundary scan features of the IQ devices can be used for board level testing. The boundary scan testing requires the use of four external signals, TDI, TDO, TMS, and TCK, as follows:

- TDI is the input that carries a sequence of 0s and 1s to be serially shifted into the device chain.
- As a bit sequence is shifted in, another bit stream is shifted out to the TDO pin.
- TMS, which is connected in parallel (broadcast) to all devices, controls the state transitions for the JTAG controller inside each IQ device.
- TCK is the input clock that synchronizes the operation of the other three signals.

Board-Level Testing

The boundary scan features on the IQ devices can be used for board-level testing. The board-level testing involves the following steps:

1. **Generating a sequence of 0s and 1s to be scanned into the TDI and TMS pins**

   The sequence shifted into TMS steers the controllers through various states. The sequence scanned into TDI represents either JTAG instructions or contents for the various data registers.

2. **The two input bit streams, one for TMS and the other for TDI, are scanned into the device chain synchronized by the TCK clock.**

   While this task is taking place, another sequence of 0s and 1s is scanned out to TDO. Some hardware mechanism is needed to capture this output stream.

3. **Analyzing the resulting TDO bit stream by using software to test and debug.**

   The steps of generating input patterns, downloading or uploading the patterns, and analyzing the output patterns can happen in a clock-by-clock fashion, in bursts of a small number of clocks, or in batch fashion.

Figure 5-1 illustrates how multiple IQ devices are connected together on a board.

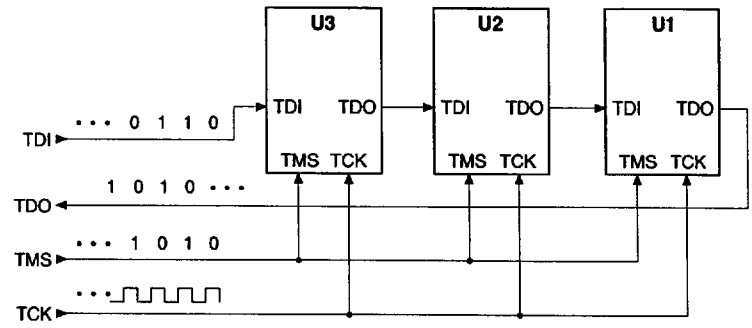

Figure 5-1 Programming Multiple IQ Devices in a Chain

Generating the Bit Sequence

Using the information provided in this chapter and elsewhere in the manual, the user can write the code that will generate the bit sequences for TDI and TMS signals, and analyze a TDO bit stream allowing the user to perform the necessary test functions. The user needs to be familiar with the general operation of a JTAG-compliant device to use this method.

If hardware is available for scanning bit sequences in and out of IQ devices, a two-way, interactive operation can be carried out. A typical sequence is to shift in a JTAG instruction followed by a bit stream for one of the data registers. The TDO pattern which was shifted out is then analyzed.

JTAG Registers

Only the following JTAG registers shown in Figure 5-2 get used in boundary scan testing.

- Instruction Register (IR)
- Bypass Register (Bypass)
- Device ID Register (ID)
- Boundary Scan Data Register (Bscan)

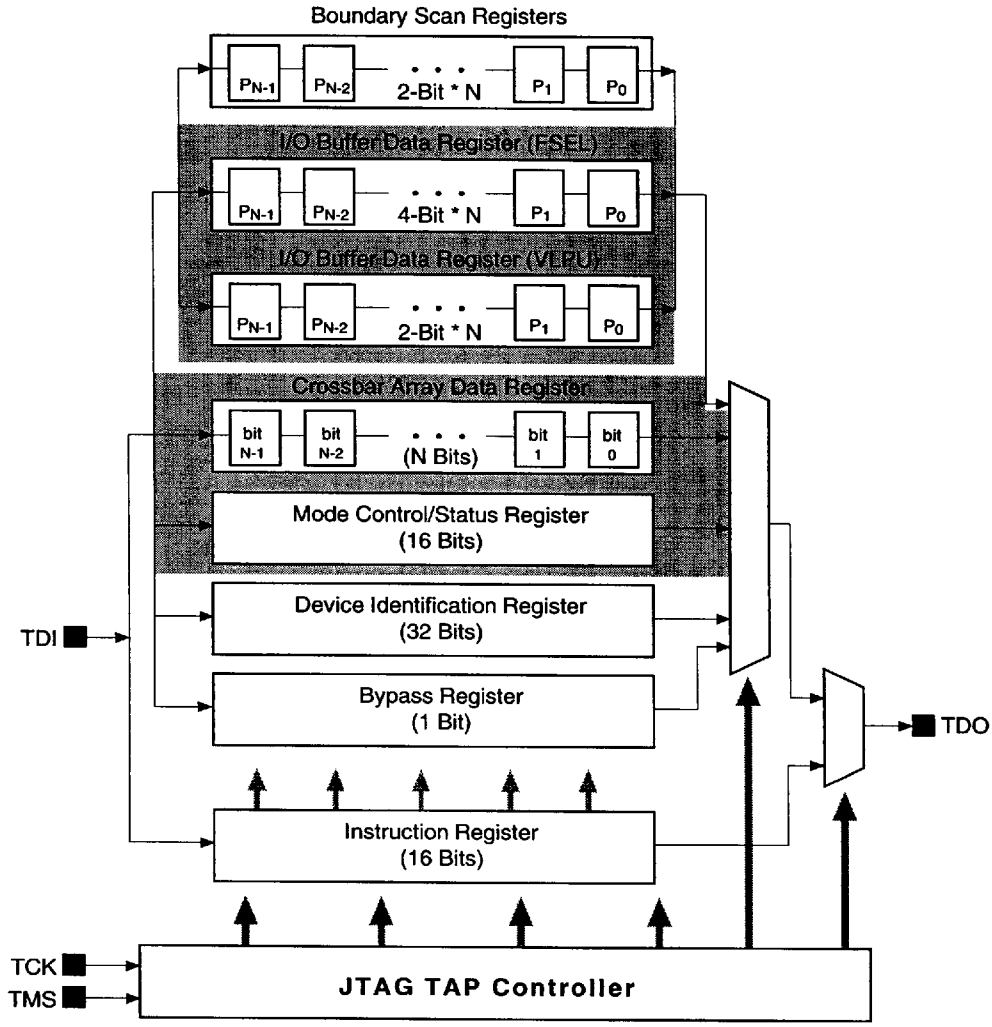

Figure 5-2 JTAG Registers Used for Boundary Scan Testing

Instruction Register

The Instruction Register (IR) is 16 bits wide. Boundary scan related JTAG instructions are loaded into this register.

Bypass Register

The Bypass Register is a 1-bit register that connects TDI to TDO with a nearly direct connection. It is useful for accessing one specific device in a chain of IQ devices.

Device ID Register

The 32-bit Device Identification Register identifies a particular device in the device chain. Table 5-1 shows ID codes used for the IQ devices:

Table 5-1 Device ID Codes

Devices	ID Code
IQ320	$0000289F_H$
IQ240B	$0000289F_H$
IQ160	$0000389F_H$
IQ128B	$0000389F_H$
IQ96	$0000589F_H$
IQ64B	$0000589F_H$
IQ48	$0000689F_H$
IQ32B	$0000689F_H$

Boundary Scan Register

All the I/O Port pins are included in the boundary scan chain. There are a total of N boundary scan cells on the IQ devices. Table 5-3 shows the count. The ordering of the boundary scan cells in the scan chain for IQ devices is shown in Table 5-3, Table 5-4, Table 5-5, and Table 5-6.

Table 5-2 Total number of boundary scan cells (N)

Device	N
IQ320	320
IQ240B	320
IQ160	160
IQ128B	160
IQ128B	160
IQ96	96
IQ64B	96
IQ48	48
IQ32B	48

Table 5-3 Boundary Scan Cells for IQ320 and IQ240B

Scan Cell	IQ320 Pin Name	Scan Cell	IQ320 Pin Name	Scan Cell	IQ320 Pin Name	Scan Cell	IQ320 Pin Name
0	Port 000	40	Port 040	80	Port 080	120	Port 120
1	Port 001	41	Port 041	81	Port 081	121	Port 121
2	Port 002	42	Port 042	82	Port 082	122	Port 122
3	Port 003	43	Port 043	83	Port 083	123	Port 123
4	Port 004	44	Port 044	84	Port 084	124	Port 124
5	Port 005	45	Port 045	85	Port 085	125	Port 125
6	Port 006	46	Port 046	86	Port 086	126	Port 126
7	Port 007	47	Port 047	87	Port 087	127	Port 127
8	Port 008	48	Port 048	88	Port 088	128	Port 128
9	Port 009	49	Port 049	89	Port 089	129	Port 129
10	Port 010	50	Port 050	90	Port 090	130	Port 130
11	Port 011	51	Port 051	91	Port 091	131	Port 131
12	Port 012	52	Port 052	92	Port 092	132	Port 132
13	Port 013	53	Port 053	93	Port 093	133	Port 133
14	Port 014	54	Port 054	94	Port 094	134	Port 134
15	Port 015	55	Port 055	95	Port 095	135	Port 135
16	Port 016	56	Port 056	96	Port 096	136	Port 136
17	Port 017	57	Port 057	97	Port 097	137	Port 137
18	Port 018	58	Port 058	98	Port 098	138	Port 138
19	Port 019	59	Port 059	99	Port 099	139	Port 139
20	Port 020	60	Port 060	100	Port 100	140	Port 140
21	Port 021	61	Port 061	101	Port 101	141	Port 141
22	Port 022	62	Port 062	102	Port 102	142	Port 142
23	Port 023	63	Port 063	103	Port 103	143	Port 143
24	Port 024	64	Port 064	104	Port 104	144	Port 144
25	Port 025	65	Port 065	105	Port 105	145	Port 145
26	Port 026	66	Port 066	106	Port 106	146	Port 146
27	Port 027	67	Port 067	107	Port 107	147	Port 147
28	Port 028	68	Port 068	108	Port 108	148	Port 148
29	Port 029	69	Port 069	109	Port 109	149	Port 149
30	Port 030	70	Port 070	110	Port 110	150	Port 150
31	Port 031	71	Port 071	111	Port 111	151	Port 151
32	Port 032	72	Port 072	112	Port 112	152	Port 152
33	Port 033	73	Port 073	113	Port 113	153	Port 153
34	Port 034	74	Port 074	114	Port 114	154	Port 154
35	Port 035	75	Port 075	115	Port 115	155	Port 155
36	Port 036	76	Port 076	116	Port 116	156	Port 156
37	Port 037	77	Port 077	117	Port 117	157	Port 157
38	Port 038	78	Port 078	118	Port 118	158	Port 158
39	Port 039	79	Port 079	119	Port 119	159	Port 159

Table 5-3 Boundary Scan Cells for IQ320 and IQ240B

Scan Cell	IQ320 Pin Name	Scan Cell	IQ320 Pin Name	Scan Cell	IQ320 Pin Name	Scan Cell	IQ320 Pin Name
160	Port 160	200	Port 200	240	Port 240	280	Port 280
161	Port 161	201	Port 201	241	Port 241	281	Port 281
162	Port 162	202	Port 202	242	Port 242	282	Port 282
163	Port 163	203	Port 203	243	Port 243	283	Port 283
164	Port 164	204	Port 204	244	Port 244	284	Port 284
165	Port 165	205	Port 205	245	Port 245	285	Port 285
166	Port 166	206	Port 206	246	Port 246	286	Port 286
167	Port 167	207	Port 207	247	Port 247	287	Port 287
168	Port 168	208	Port 208	248	Port 248	288	Port 288
169	Port 169	209	Port 209	249	Port 249	289	Port 289
170	Port 170	210	Port 210	250	Port 250	290	Port 290
171	Port 171	211	Port 211	251	Port 251	291	Port 291
172	Port 172	212	Port 212	252	Port 252	292	Port 292
173	Port 173	213	Port 213	253	Port 253	293	Port 293
174	Port 174	214	Port 214	254	Port 254	294	Port 294
175	Port 175	215	Port 215	255	Port 255	295	Port 295
176	Port 176	216	Port 216	256	Port 256	296	Port 296
177	Port 177	217	Port 217	257	Port 257	297	Port 297
178	Port 178	218	Port 218	258	Port 258	298	Port 298
179	Port 179	219	Port 219	259	Port 259	299	Port 299
180	Port 180	220	Port 220	260	Port 260	300	Port 300
181	Port 181	221	Port 221	261	Port 261	301	Port 301
182	Port 182	222	Port 222	262	Port 262	302	Port 302
183	Port 183	223	Port 223	263	Port 263	303	Port 303
184	Port 184	224	Port 224	264	Port 264	304	Port 304
185	Port 185	225	Port 225	265	Port 265	305	Port 305
186	Port 186	226	Port 226	266	Port 266	306	Port 306
187	Port 187	227	Port 227	267	Port 267	307	Port 307
188	Port 188	228	Port 228	268	Port 268	308	Port 308
189	Port 189	229	Port 229	269	Port 269	309	Port 309
190	Port 190	230	Port 230	270	Port 270	310	Port 310
191	Port 191	231	Port 231	271	Port 271	311	Port 311
192	Port 192	232	Port 232	272	Port 272	312	Port 312
193	Port 193	233	Port 233	273	Port 273	313	Port 313
194	Port 194	234	Port 234	274	Port 274	314	Port 314
195	Port 195	235	Port 235	275	Port 275	315	Port 315
196	Port 196	236	Port 236	276	Port 276	316	Port 316
197	Port 197	237	Port 237	277	Port 277	317	Port 317
198	Port 198	238	Port 238	278	Port 278	318	Port 318
199	Port 199	239	Port 239	279	Port 279	319	Port 319

Table 5-4 Boundary Scan Cells for IQ160 and IQ128B

Scan Cell	IQ160 Pin Name	Scan Cell	IQ160 Pin Name	Scan Cell	IQ160 Pin Name	Scan Cell	IQ160 Pin Name
0	Port 000	40	Port 040	80	Port 080	120	Port 120
1	Port 001	41	Port 041	81	Port 081	121	Port 121
2	Port 002	42	Port 042	82	Port 082	122	Port 122
3	Port 003	43	Port 043	83	Port 083	123	Port 123
4	Port 004	44	Port 044	84	Port 084	124	Port 124
5	Port 005	45	Port 045	85	Port 085	125	Port 125
6	Port 006	46	Port 046	86	Port 086	126	Port 126
7	Port 007	47	Port 047	87	Port 087	127	Port 127
8	Port 008	48	Port 048	88	Port 088	128	Port 128
9	Port 009	49	Port 049	89	Port 089	129	Port 129
10	Port 010	50	Port 050	90	Port 090	130	Port 130
11	Port 011	51	Port 051	91	Port 091	131	Port 131
12	Port 012	52	Port 052	92	Port 092	132	Port 132
13	Port 013	53	Port 053	93	Port 093	133	Port 133
14	Port 014	54	Port 054	94	Port 094	134	Port 134
15	Port 015	55	Port 055	95	Port 095	135	Port 135
16	Port 016	56	Port 056	96	Port 096	136	Port 136
17	Port 017	57	Port 057	97	Port 097	137	Port 137
18	Port 018	58	Port 058	98	Port 098	138	Port 138
19	Port 019	59	Port 059	99	Port 099	139	Port 139
20	Port 020	60	Port 060	100	Port 100	140	Port 140
21	Port 021	61	Port 061	101	Port 101	141	Port 141
22	Port 022	62	Port 062	102	Port 102	142	Port 142
23	Port 023	63	Port 063	103	Port 103	143	Port 143
24	Port 024	64	Port 064	104	Port 104	144	Port 144
25	Port 025	65	Port 065	105	Port 105	145	Port 145
26	Port 026	66	Port 066	106	Port 106	146	Port 146
27	Port 027	67	Port 067	107	Port 107	147	Port 147
28	Port 028	68	Port 068	108	Port 108	148	Port 148
29	Port 029	69	Port 069	109	Port 109	149	Port 149
30	Port 030	70	Port 070	110	Port 110	150	Port 150
31	Port 031	71	Port 071	111	Port 111	151	Port 151
32	Port 032	72	Port 072	112	Port 112	152	Port 152
33	Port 033	73	Port 073	113	Port 113	153	Port 153
34	Port 034	74	Port 074	114	Port 114	154	Port 154
35	Port 035	75	Port 075	115	Port 115	155	Port 155
36	Port 036	76	Port 076	116	Port 116	156	Port 156
37	Port 037	77	Port 077	117	Port 117	157	Port 157
38	Port 038	78	Port 078	118	Port 118	158	Port 158
39	Port 039	79	Port 079	119	Port 119	159	Port 159

Table 5-5 Boundary Scan Cells for IQ96 and IQ64B

Scan Cell	IQ96 Pin Name	Scan Cell	IQ96 Pin Name	Scan Cell	IQ96 Pin Name	Scan Cell	IQ96 Pin Name
0	Port 000	24	Port 024	48	Port 048	72	Port 072
1	Port 001	25	Port 025	49	Port 049	73	Port 073
2	Port 002	26	Port 026	50	Port 050	74	Port 074
3	Port 003	27	Port 027	51	Port 051	75	Port 075
4	Port 004	28	Port 028	52	Port 052	76	Port 076
5	Port 005	29	Port 029	53	Port 053	77	Port 077
6	Port 006	30	Port 030	54	Port 054	78	Port 078
7	Port 007	31	Port 031	55	Port 055	79	Port 079
8	Port 008	32	Port 032	56	Port 056	80	Port 080
9	Port 009	33	Port 033	57	Port 057	81	Port 081
10	Port 010	34	Port 034	58	Port 058	82	Port 082
11	Port 011	35	Port 035	59	Port 059	83	Port 083
12	Port 012	36	Port 036	60	Port 060	84	Port 084
13	Port 013	37	Port 037	61	Port 061	85	Port 085
14	Port 014	38	Port 038	62	Port 062	86	Port 086
15	Port 015	39	Port 039	63	Port 063	87	Port 087
16	Port 016	40	Port 040	64	Port 064	88	Port 088
17	Port 017	41	Port 041	65	Port 065	89	Port 089
18	Port 018	42	Port 042	66	Port 066	90	Port 090
19	Port 019	43	Port 043	67	Port 067	91	Port 091
20	Port 020	44	Port 044	68	Port 068	92	Port 092
21	Port 021	45	Port 045	69	Port 069	93	Port 093
22	Port 022	46	Port 046	70	Port 070	94	Port 094
23	Port 023	47	Port 047	71	Port 071	95	Port 095

Table 5-6 Boundary Scan Cells for IQ48 and IQ32B

Scan Cell	IQ48 Pin Name	Scan Cell	IQ48 Pin Name	Scan Cell	IQ48 Pin Name	Scan Cell	IQ48 Pin Name
0	Port 000	12	Port 012	24	Port 024	36	Port 036
1	Port 001	13	Port 013	25	Port 025	37	Port 037
2	Port 002	14	Port 014	26	Port 026	38	Port 038
3	Port 003	15	Port 015	27	Port 027	39	Port 039
4	Port 004	16	Port 016	28	Port 028	40	Port 040
5	Port 005	17	Port 017	29	Port 029	41	Port 041
6	Port 006	18	Port 018	30	Port 030	42	Port 042
7	Port 007	19	Port 019	31	Port 031	43	Port 043
8	Port 008	20	Port 020	32	Port 032	44	Port 044
9	Port 009	21	Port 021	33	Port 033	45	Port 045
10	Port 010	22	Port 022	34	Port 034	46	Port 046
11	Port 011	23	Port 023	35	Port 035	47	Port 047

Each scan cell is two bits wide. The functions performed by the two bits vary depending on the operation. This is shown in Table 5-7 and is explained later.

Table 5-7 Boundary Scan Cell Bits & Function

During...	Bit Location	Bit Name	Function
Read (Capture DR)	0 (LSB)	AS	Array Side value for I/O Port pins
	1 (MSB)	PS	Pin Side value for I/O Port pins
Write (Update DR)	0 (LSB)	JE	Control Bit; 0 = Tristate, 1 = Force
	1 (MSB)	JD	Data Bit

Bit Shifting Order for TDI data

The shifting of the TDI data bits for the Instruction Register and the different Data Registers are shown below. The bit indicated on the *right* is shifted in first.

For Instruction Register

Bit_{15}, ..., Bit_2, Bit_1, Bit_0

For Bypass Register

Bit_0

For Device ID Register

Bit_{31}, ..., Bit_2, Bit_1, Bit_0

For Boundary Scan Register

$ScanCell_{N-1}Bit_1$, $ScanCell_{N-1}Bit_0$, ... , $ScanCell_1Bit_1$, $ScanCell_1Bit_0$, $ScanCell_0Bit_1$, $ScanCell_0Bit_0$

JTAG Instructions

The Instruction Register (IR) selects the test operation to be performed. The JTAG instructions available to the user for performing the boundary scan functions are:

0000_H	0000	0000	0000	0000	Extest
0005_H	0000	0000	0000	0101	Sample / Preload
$FFE1_H$	1111	1111	1110	0001	Device ID Register
$FFFF_H$	1111	1111	1111	1111	Bypass Register

Using Boundary Scan

The boundary scan features on the IQ devices can be used for either forcing signal values onto the device pins or for reading the signal values appearing on the device pins.

The Extest instruction is used to internally force the pins of the device to a logic value. This is accomplished by shifting in the Extest instruction during the IR Scan followed by the data for the Boundary Scan Data Register during DR Scan. The two bits of data are shifted in for each boundary scan cell - the data bit (JD) is the data to be put out onto the corresponding pin, while the control bit (JE) specifies if the pin is to be driven or tristated. The data is loaded into the Boundary Scan Register during *Update DR*.

In addition, the Extest instruction captures the data values at the device pins into the Boundary Scan Data Register during *Capture DR* for reading out over the TDO pin. The two bits of data that are captured are called AS and PS. For I/O Port pins, Bit 0 (AS) contains the data value from the array side, i.e., from the corresponding line in the Switch Matrix, of the I/O Port pin while Bit 1 (PS) contains the data value present on the I/O Port pin. For non I/O Port pins,

Bit 0 is to be ignored while Bit 1 (PS) contains the data value present on the pin. Since the JTAG controller passes through the *Capture DR* state before it reaches the *Update DR* state, the data captured for readback is the result of the data forced on the (output) pins during the previous *Update DR*.

During Extest, the internal logic that is normally driving the device pins is disconnected. Instead, the data read into the Boundary Scan Register during *Update DR* is applied to the pins. The data that is loaded into the AS and PS bits for readback during *Capture DR* is either the boundary scan data (JD) driven onto the pins (when JE = 1) or is the external signal value connected to the pins (when JE = 0). The JE bit is reset to "0" during device reset.

The Sample / Preload is a read-only instruction and is also used for reading data appearing on the device pins. The pin values are read into Bit 0 and Bit 1 of the Boundary Scan Data Register during *Capture DR*. The two read bits are called AS and PS. For I/O Port pins, Bit 0 (AS) contains the data from the array side, i.e., from the corresponding line in the Switch Matrix, of the I/O Port pin, while Bit 1 (PS) contains the data value present on the I/O Port pin. For non I/O Port pins, Bit 0 is to be ignored while Bit 1 (PS) contains the data value present on the pin. In the case of Sample / Preload instruction, the data values are the ones driven on to the (output) pins by the internal device logic or by external signals connected to the (input) pins.

Using Boundary Scan on IQ Devices with ..B Suffix

Devices with a ...B suffix are bond-out versions of another IQ device die with some I/O Port s and Switch Matrix lines left unused.

- IQ240B, IQ128B, IQ64B and IQ32B are a bond-out of the IQ320, IQ160, IQ96 and IQ48 dice respectively.

Although some I/O Port s and some other pins from the original device are not available on these bond-out devices, the scan cells corresponding to these pins are still a part of the Boundary Scan Data Register and must be properly accounted for.

PAGE(S) INTENTIONALLY BLANK

Programming Examples 6

This chapter provides several examples that illustrate programming a chain of IQ devices.

Tables in this chapter list the values required for programming. Within the tables, the following symbols are used:

- An x in the TDI or the TDO column indicates that the data is "don't care" or unknown.
- A ? in the Controller State column indicates that the TAP controller is in an unknown state. If the system was just powered on, any IQ devices were probably initialized during the initialization sequence.

Reset the IQ Device and Shift Out Chip ID

Table 6-1 Resetting an IQ Device and Shift Out Chip ID

JTAG Cycle #	TMS	TDI	TDO	Controller State	Comments
1	1[1]	x	x	?	
2	1[1]	x	x	?	
3	1[1]	x	x	?	
•	1[1]	x	x	?	
•	1[1]	x	x	Test Logic Reset	This cycle can be repeated as long as desired. The IQ Family chip and the Instruction Register are reset to zero, which is the instruction for chip ID.
•	0	x	x	Run Test/Idle	Again, this cycle can be repeated as long as desired.
•	1	x	x	Select DR Scan	
•	0	x	x	Capture DR	
•	0	x	x	Shift DR	
•	0	x	id_0	Shift DR	Shift out the device ID, starting with bit 0.
•	0	x	id_1	Shift DR	
•	⋮				
•	0	x	id_{30}	Shift DR	
•	1	x	id_{31}	Exit 1 DR	If you have several devices, stay in the Shift DR state and continue to scan the ID of the second device, etc. The most significant bit of the ID of the last device coincides with the Exit 1 DR transition.
•	1	x	x	Update DR	
•	0	x	x	Run Test/Idle	Back in the Idle state.

1. Assuming an unknown initial state, five 1s brings the controller to a known state and resets the chip.

Read/Write an Xarray Word (word 141_{10}) of a Chain of IQ320s

Table 6-2 Read/Write an Xarray Word

JTAG Cycle #	TMS	TDI	TDO	Controller State	Comments
1	0	x	x	Run Test/Idle	Default state.
2	1	x	x	Select DR Scan	
3	1	x	x	Select IR Scan	
•	0	x	x	Capture IR	
•	0	x	x	Shift IR	
•	0	I_0	i_0	Shift IR	Shift in the instruction bit pattern 0010 0010 0011 0101 (2235_{16}. bit 0 is the first bit.) which is the Xarray RW instruction with a row address of 141_{10}.
•	⋮				
•	0	I_{14}	i_{14}	Shift IR	The old instruction $i_{15}i_{14}...i_0$ gets shifted out.
•	1	I_{15}	i_{15}	Exit 1 IR	If there is more than one device, remain in the Shift IR state and continue to shift the same instruction bit pattern to the IR of the second device, etc. The most significant instruction bit of the last device coincides with the `Exit 1 IR` transition.
•	1	x	x	Update IR	
•	1	x	x	Select DR Scan	
•	0	x	x	Capture DR	
•	0	x	x	Shift DR	
•	0	D_0	d_0	Shift DR	Load the Xarray DR with new data and shift out old word contents.
•	⋮				
•	0	D_{318}	d_{318}	Shift DR	

Table 6-2 Read/Write an Xarray Word *(Continued)*

JTAG Cycle #	TMS	TDI	TDO	Controller State	Comments
•	1	D_{319}	d_{319}	Exit 1 DR	If you have more than one device, remain in the Shift DR state and continue to shift in the Xarray DR contents for the second device, etc. In a multi-IQ device case in which a device is put in BYPASS mode (the device's IR was loaded with the BYPASS instruction above), the DR pattern consists of one bit.
•	1	x	x	Update DR	
•	0	x	x	Run Test/Idle	Back in the Idle state.

Set Up I/O Buffers for a Chain of IQ320s

Table 6-3 Programming I/O Buffers for a Chain of IQ320s

JTAG Cycle #	TMS	TDI	TDO	Controller State	Comments
1	0	x	x	Run Test/Idle	Default state.
2	1	x	x	Select DR Scan	
3	1	x	x	Select IR Scan	
•	0	x	x	Capture IR	
•	0	x	x	Shift IR	
•	0	I_0	i_0	Shift IR	Shift in instruction bit pattern 1000 1100 0000 0001 ($8C01_{16}$), which is the Function Select (FSEL) instruction.
•	⋮				
•	0	I_{14}	i_{14}	Shift IR	
•	1	I_{15}	i_{15}	Exit 1 IR	If you have more than one device, stay in the Shift IR state and continue to shift the same instruction bit pattern to the IR of the second device, and so on. Notice that the most significant instruction bit of the last device coincides with the Exit 1 IR transition.
•	1	x	x	Update IR	
•	1	x	x	Select DR Scan	
•	0	x	x	Capture DR	
•	0	x	x	Shift DR	
•	0	$F_{0,0}$	$d_{0,0}$	Shift DR	Load the Function Select DR with new data and shift out old register contents, starting with Port 0 of device 0. Each port takes four cycles.
•	0	$F_{0,1}$	$f_{0,1}$	Shift DR	
•	0	$F_{0,2}$	$f_{0,2}$	Shift DR	
•	0	$F_{0,3}$	$f_{0,3}$	Shift DR	
•	0	$F_{1,0}$	$f_{1,0}$	Shift DR	Port 1
•	•	•	•	•	•

Table 6-3 Programming I/O Buffers for a Chain of IQ320s *(Continued)*

JTAG Cycle #	TMS	TDI	TDO	Controller State	Comments
•	•	•	•	•	•
•	1	$F_{319,3}$	$f_{319,3}$	Exit 1 DR	Port 319 of device 0. If you have more than one device, stay in the Shift DR state and continue to shift in the FSEL contents for all the ports of the second device, and so on. In a multi-IQ device case in which a device is put in BYPASS mode (the device's IR was loaded with the BYPASS instruction above), the DR pattern consists of one bit.
•	1	x	x	Update DR	
•	0	x	x	Run Test/Idle	Back in the Idle state.

Look-up and Port Mapping Tables A

The tables in this appendix specify the die to package pin mapping for the I/O Ports and the look-up values for determining real and ghost SRAM cell locations in the Switch Matrix.

The following tables help determine the locations of the real SRAM cells in the Switch Matrix. The SRAM cell controlling the connection between I/O Port "i" and I/O Port "j" is determined as follows:

Get the *Index* values corresponding to I/O Port numbers "i" and "j". If the index value for "i" is greater than index value for "j", then the SRAM cell has the row (word) address i* and column (bit) address j*, otherwise it has row address of j* and column address of i*. The numbers i* and j* represent the I/O Port locations on the die.

Example 1: On the IQ240B, the SRAM cell controlling the connection between I/O Port 80 and I/O Port 180 is at location: Row address = 241, Column Address = 110. Note that the IQX240B is a bondout version of IQX320 die, and the I/O Ports 80 and 180 on the device package are the I/O Ports 110 and 241 respectively on the device die.

Example 2: On the IQ160, the SRAM cell controlling the connection between I/O Port 20 and I/O Port 100 is at location: Row address = 100, Column Address = 20

.

Table A-1 IQ320 and IQ240B Table

Die Port, IQ320 Port #	IQ240B Port #	Index	Die Port, IQ320 Port #	IQ240B Port #	Index
0	-	2	40	30	162
1	0	6	41	31	166
2	-	10	42	32	170
3	1	14	43	33	174
4	-	18	44	34	178
5	2	22	45	35	182
6	-	26	46	36	186
7	3	30	47	37	190
8	-	34	48	38	194
9	4	38	49	39	198
10	-	42	50	40	202
11	5	46	51	41	206
12	-	50	52	42	210
13	6	54	53	43	214
14	-	58	54	44	218
15	7	62	55	45	222
16	8	66	56	-	226
17	9	70	57	46	230
18	10	74	58	-	234
19	11	78	59	47	238
20	12	82	60	-	242
21	13	86	61	48	246
22	14	90	62	-	250
23	15	94	63	49	254
24	-	98	64	-	258
25	16	102	65	50	262
26	17	106	66	-	266
27	18	110	67	51	270
28	19	114	68	-	274
29	20	118	69	52	278
30	21	122	70	-	282
31	22	126	71	53	286
32	23	130	72	-	290
33	24	134	73	54	294
34	-	138	74	-	298
35	25	142	75	55	302
36	26	146	76	56	306
37	27	150	77	57	310
38	28	154	78	58	314
39	29	158	79	59	318

Table A-1 IQ320 and IQ240B Table

Die Port, IQ320 Port #	IQ240B Port #	Index	Die Port, IQ320 Port #	IQ240B Port #	Index
80	-	0	120	90	160
81	60	4	121	91	164
82	-	8	122	92	168
83	61	12	123	93	172
84	-	16	124	94	176
85	62	20	125	95	180
86	-	24	126	96	184
87	63	28	127	97	188
88	-	32	128	98	192
89	64	36	129	99	196
90	-	40	130	100	200
91	65	44	131	101	204
92	-	48	132	102	208
93	66	52	133	103	212
94	67	56	134	104	216
95	68	60	135	105	220
96	-	64	136	106	224
97	69	68	137	107	228
98	-	72	138	108	232
99	70	76	139	109	236
100	-	80	140	-	240
101	71	84	141	110	244
102	72	88	142	-	248
103	73	92	143	111	252
104	74	96	144	-	256
105	75	100	145	112	260
106	76	104	146	-	264
107	77	108	147	113	268
108	78	112	148	-	272
109	79	116	149	114	276
110	80	120	150	-	280
111	81	124	151	115	284
112	82	128	152	-	288
113	83	132	153	116	292
114	84	136	154	-	296
115	85	140	155	117	300
116	86	144	156	-	304
117	87	148	157	118	308
118	88	152	158	-	312
119	89	156	159	119	316

Table A-1 IQ320 and IQ240B Table

Die Port, IQ320 Port #	IQ240B Port #	Index	Die Port, IQ320 Port #	IQ240B Port #	Index
160	-	319	200	149	159
161	120	315	201	150	155
162	-	311	202	151	151
163	121	307	203	152	147
164	-	303	204	153	143
165	122	299	205	154	139
166	-	295	206	155	135
167	123	291	207	156	131
168	-	287	208	157	127
169	124	283	209	158	123
170	-	279	210	159	119
171	125	275	211	160	115
172	-	271	212	161	111
173	126	267	213	162	107
174	-	263	214	163	103
175	127	259	215	164	99
176	-	255	216	165	95
177	128	251	217	166	91
178	-	247	218	167	87
179	129	243	219	168	83
180	-	239	220	169	79
181	130	235	221	170	75
182	131	231	222	-	71
183	132	227	223	171	67
184	133	223	224	-	63
185	134	219	225	172	59
186	135	215	226	-	55
187	136	211	227	173	51
188	137	207	228	-	47
189	138	203	229	174	43
190	139	199	230	-	39
191	140	195	231	175	35
192	141	191	232	-	31
193	142	187	233	176	27
194	143	183	234	-	23
195	144	179	235	177	19
196	145	175	236	-	15
197	146	171	237	178	11
198	147	167	238	-	7
199	148	163	239	179	3

Table A-1 IQ320 and IQ240B Table

Die Port, IQ320 Port #	IQ240B Port #	Index	Die Port, IQ320 Port #	IQ240B Port #	Index
240	-	317	280	-	157
241	180	313	281	208	153
242	181	309	282	-	149
243	182	305	283	209	145
244	-	301	284	-	141
245	183	297	285	210	137
246	-	293	286	-	133
247	184	289	287	211	129
248	-	285	288	-	125
249	185	281	289	212	121
250	-	277	290	-	117
251	186	273	291	213	113
252	-	269	292	-	109
253	187	265	293	214	105
254	188	261	294	-	101
255	189	257	295	215	97
256	190	253	296	216	93
257	191	249	297	217	89
258	192	245	298	218	85
259	193	241	299	219	81
260	194	237	300	220	77
261	195	233	301	221	73
262	-	229	302	222	69
263	196	225	303	223	65
264	-	221	304	224	61
265	197	217	305	225	57
266	-	213	306	226	53
267	198	209	307	227	49
268	199	205	308	228	45
269	200	201	309	229	41
270	201	197	310	230	37
271	202	193	311	231	33
272	203	189	312	232	29
273	204	185	313	233	25
274	-	181	314	234	21
275	205	177	315	235	17
276	-	173	316	236	13
277	206	169	317	237	9
278	-	165	318	238	5
279	207	161	319	239	1

Table A-2 IQ160 and IQ128B Table

Die Port, IQ160 Port #	IQ128B Port #	Index	Die Port, IQ160 Port #	IQ128B Port #	Index
0	0	2	40	-	0
1	1	6	41	-	4
2	2	10	42	32	8
3	3	14	43	33	12
4	4	18	44	34	16
5	5	22	45	35	20
6	6	26	46	36	24
7	7	30	47	-	28
8	8	34	48	37	32
9	9	38	49	38	36
10	10	42	50	39	40
11	-	46	51	40	44
12	-	50	52	-	48
13	11	54	53	41	52
14	12	58	54	42	56
15	13	62	55	43	60
16	14	66	56	44	64
17	-	70	57	45	68
18	-	74	58	46	72
19	15	78	59	-	76
20	16	82	60	47	80
21	17	86	61	48	84
22	18	90	62	49	88
23	-	94	63	50	92
24	-	98	64	51	96
25	19	102	65	-	100
26	20	106	66	52	104
27	21	110	67	53	108
28	22	114	68	54	112
29	-	118	69	55	116
30	-	122	70	56	120
31	23	126	71	-	124
32	24	130	72	57	128
33	25	134	73	58	132
34	26	138	74	59	136
35	27	142	75	60	140
36	28	146	76	61	144
37	29	150	77	62	148
38	30	154	78	63	152
39	31	158	79	-	156

Table A-2 IQ160 and IQ128B Table

Die Port, IQ160 Port #	IQ128B Port #	Index	Die Port, IQ160 Port #	IQ128B Port #	Index
80	64	159	120	96	157
81	65	155	121	97	153
82	66	151	122	98	149
83	-	147	123	-	145
84	67	143	124	-	141
85	68	139	125	99	137
86	69	135	126	100	133
87	70	131	127	101	129
88	71	127	128	102	125
89	-	123	129	-	121
90	72	119	130	-	117
91	73	115	131	103	113
92	74	111	132	104	109
93	75	107	133	105	105
94	76	103	134	106	101
95	-	99	135	-	97
96	77	95	136	-	93
97	78	91	137	107	89
98	79	87	138	108	85
99	80	83	139	109	81
100	81	79	140	110	77
101	-	75	141	111	73
102	82	71	142	112	69
103	83	67	143	113	65
104	84	63	144	114	61
105	85	59	145	115	57
106	86	55	146	116	53
107	-	51	147	-	49
108	87	47	148	117	45
109	88	43	149	118	41
110	89	39	150	119	37
111	90	35	151	120	33
112	91	31	152	121	29
113	-	27	153	122	25
114	92	23	154	123	21
115	93	19	155	-	17
116	94	15	156	124	13
117	95	11	157	125	9
118	-	7	158	126	5
119	-	3	159	127	1

Table A-3 IQ96 and IQ64B Table

Die Port, IQ96 Port #	IQ64B Port #	Index	Die Port, IQ96 Port #	IQ64B Port #	Index
0	0	2	24	-	0
1	1	6	25	-	4
2	-	10	26	-	8
3	-	14	27	16	12
4	2	18	28	-	16
5	3	22	29	17	20
6	-	26	30	-	24
7	-	30	31	-	28
8	-	34	32	-	32
9	-	38	33	-	36
10	-	42	34	18	40
11	-	46	35	19	44
12	4	50	36	20	48
13	5	54	37	21	52
14	6	58	38	22	56
15	7	62	39	23	60
16	8	66	40	24	64
17	9	70	41	25	68
18	10	74	42	26	72
19	11	78	43	27	76
20	12	82	44	28	80
21	13	86	45	29	84
22	14	90	46	30	88
23	15	94	47	31	92

Table A-3 IQ96 and IQ64B Table

Die Port, IQ96 Port #	IQ64B Port #	Index	Die Port, IQ96 Port #	IQ64B Port #	Index
48	32	95	72	-	93
49	33	91	73	-	89
50	34	87	74	-	85
51	35	83	75	-	81
52	36	79	76	-	77
53	37	75	77	-	73
54	38	71	78	48	69
55	39	67	79	49	65
56	40	63	80	50	61
57	41	59	81	51	57
58	42	55	82	52	53
59	43	51	83	53	49
60	44	47	84	-	45
61	45	43	85	54	41
62	46	39	86	55	37
63	47	35	87	56	33
64	-	31	88	57	29
65	-	27	89	58	25
66	-	23	90	59	21
67	-	19	91	-	17
68	-	15	92	60	13
69	-	11	93	61	9
70	-	7	94	62	5
71	-	3	95	63	1

Table A-4 IQ48 and IQ32B Table

Die Port, IQ48 Port #	IQ32B Port #	Index	Die Port, IQ48 Port #	IQ32B Port #	Index
0	-	2	24	-	47
1	0	6	25	15	43
2	1	10	26	16	39
3	2	14	27	17	35
4	-	18	28	-	31
5	-	22	29	-	27
6	-	26	30	-	23
7	-	30	31	-	19
8	3	34	32	18	15
9	4	38	33	19	11
10	5	42	34	20	7
11	6	46	35	21	3
12	7	0	36	22	45
13	8	4	37	23	41
14	9	8	38	-	37
15	-	12	39	24	33
16	10	16	40	25	29
17	-	20	41	26	25
18	-	24	42	27	21
19	-	28	43	28	17
20	11	32	44	-	13
21	12	36	45	29	9
22	13	40	46	30	5
23	14	44	47	31	1

Revision History

Revision	Date	Comments
1.00	August 1992	First Release
2.00	June 1994	Support for the IQ family devices
3.00	March 1995	Support for new IQ devices: IQ96 and IQ64B
3.20	January 1996	Support for new IQ devices: IQ48 and IQ32B
3.40	January 1997	Added a chapter for boundary scan testing