CoolRunner-II Programmer Qualification Specification

Revision 1.3

February 3, 2005

1.0 Introduction

The CoolRunner^{TM}-II programming and verification procedures are similar to those used with standard non-volatile (NV) memories. Memory cells can be erased only as a single array, while programming of groups of individual cells is performed. After successful completion of an erase operation, all cells in the device are in the logical “1” state. All configuration operations are performed via the IEEE 1149.1 test access port (TAP) and its sixteen state TAP controller.

The Xilinx software generates device-programming files in JEDEC format. The JEDEC file also contains device identification information, which is used to validate the device being programmed. The manufacturer’s code (IDCODE) identifies Xilinx as the manufacturer and the product code of the device. It is read via the Device Identification Register of the IEEE Standard 1149.1. Please refer to tables 4 and 5 for these values.

2.0 Features

2.1 Erase

The device is electrically erasable. The programming algorithm includes a check of the device to ensure all bits are erased before allowing programming. Note that erased cells are verified to logical 1 state.

2.2 Addressing

The device is addressed with a 6,7 or 8-bit address depending on the family member (see Table 2). The address is indexed with a Gray code count starting from zero and counting sequentially up to a maximum Gray code value. However, some bits of some addresses are “don't cares”. The legal device addresses for each family member are listed in Table 10.

2.3 Operations

The device has several independent operations: Program, Erase, Verify, and Initialize. To use these instructions, first an enable command (ISC_ENABLE) must be executed via the JTAG controller. Then, after performing the desired operations, a disable command (ISC_DISABLE) must be executed to properly configure all device locations. Program is used for selectively changing groups of NV-cells within the device. Erase is the procedure used to erase all cells in the device in one operation. Verify is used to read NV-cell content. Initialize is used to force configuration of the device to the pattern loaded in the NV-memory without requiring power cycling.

2.4 Signature String (USERCODE)

The programmer or host computer can support a mode for reading and displaying the data stored in the user storage area by executing the USERCODE instruction. The size of the user storage is 32 bits. Because the signature string is not part of the JEDEC file, programming these bytes is an independent operation from the operation of programming configuration bits and can be performed at anytime. **See the Fuse Map for the particular family member for USERCODE address and bit locations.**

Note: The erase operation will erase the pattern memory **and** the USERCODE data.

2.5 Device DONE Bits

The device supports a program done feature, which protects the design from initializing if programming has not completed successfully. The done bits will be erased with the ISC_ERASE instruction and must be re-programmed at the end of the user array program operation and before the ISC_INIT instruction is executed. **The program done bits must be programmed correctly before the device can initialize successfully**.

2.6 Device Security

The Device supports a security feature, which protects the design from being read back or altered. A secured device may still be erased and then reprogrammed. The device's IDCODE and USERCODE data will still be readable if the device is secured.

2.7 1149.1 and 1532 Standards Compliance

The CoolRunner II complies with the IEEE Std 1149.1 Standard Test Access Port and Boundary-Scan Architecture , as well as IEEE Std 1532, Standard for In-System Configuration of Programmable Devices.

The addition of the ability to capture and shift out status from the device's instruction register provides a method of identifying the four modal states defined in IEEE Std 1532, Standard for In-System Configuration of Programmable Devices. Done bits indicate if the device is programmed successfully. The availability of the ISC_ENABLE and ISC_ DISABLE signals in the instruction register capture bits allows the 1532 modal state to be determined. (see Figure 12 for instruction register capture bit assignments)

3.0 Special Instructions

3.1 Device/ File Checksum Calculation

Each JEDEC file contains a fuse checksum (C-Field). The checksum is to be implemented according to JEDEC Standard JESD3-C. **The same method must be used to calculate the device checksum, and the programmed device checksum should match the JEDEC file fuse checksum.**

3.2 Compatibility Checks

3.2.1 Adapter Type

The device adapter should contain an electronically readable code for identification. The programming algorithm must check the adapter ID for compatibility with the target device. The JEDEC design file contains information specific to the device pin count and fuse size, both of which must be compared to the adapter in use. The supported packages for each device are shown in Table 8.

4.0 Programming Sequence

The device programming sequence, illustrated in Figure 8, begins by verifying that the device design file and programming algorithm match the installed programmer adapter. This check is accomplished by first comparing the programmer adapter ID to all acceptable adapter IDs in the programmer algorithm and next to the fuse count and pin count contained in the JEDEC file. If a mismatch occurs, display message **A: “Incompatible Adapter Or File**” and terminate the programming sequence.

4.1 Device Identification Code

In response to issuing an IDCODE instruction, the device provides a 32-bit identification code. This code format is defined by IEEE STD 1149.1. The first bit on TDO is always a “1,” and the following 11 bits are the manufacturer’s identity bit (see Table 4). The 16 bits following the manufacturer identity are indicated in Table 5. First is the 3-bit package code, followed by a 1, followed by the 6-bit macrocell count, followed by “011,” followed by the 3-bit architecture code. The final four bits shifted out are a version code used by Xilinx for internal purposes.

	Version	Architecture Identifier	Reserved	Macrocell Count	Reserved	Package Identifier	Manufacturer Identity	First Bit
Values		011	011		1		00001001001	1
Bits	31-28	27-25	24-22	21-16	15	14-12	11-1	0
	TDI							**TDO**

Figure 1. Device Identification Code

4.1.1 Manufacturer’s Identity Code

The manufacturer's identity code in the device (see Table 4) is contained in bits 1 through 11 of the Device Identification Register as illustrated in Figure 1. This register is read via the ISC port with the IEEE 1149.1 IDCODE instruction (see Table 38). The code translations are listed in Table 5.

The next step is to verify that the device part number listed in the JEDEC file is a valid part number for the manufacturer determined from the Device Identification Register. If the device part number is not valid for the device manufacturer, display the message **B:** “**Manufacturer’s Code Error”** and terminate the programming sequence.

4.1.2 Product Code

The product codes, both architecture and number of macrocells codes, of the device (see Table 5) are contained in bits 25 through 27, and 16 through 21 of the Device Identification Register as illustrated in Figure 1. This register is read via ISC port with the IEEE 1149.1 IDCODE instruction (see Table 38).

The next step is to verify that the device part and package number listed in the JEDEC file is a correct for the architecture and number of macrocells determined from the Device Identification Register. If the device part and package number is not correct for the device architecture and number of macrocells, display the message **C: “Product Code Error”** and terminate the programming sequence.

4.2 Device Erase Check

Verify that all NV-cells are in the unprogrammed state (‘1’). Blank check is performed with ISC verify command (ISC_READ) for each address (see Table 22). Recall that an ISC_ENABLE instruction must be executed prior to sequentially shifting in addresses, executing verify operation and shifting out data for actual data comparison. If any of the NV-cells are programmed (‘0’), display the message **E: “Device Not Erased”** and allow the option to erase the device.

4.3 Device Bulk Erase

The device has to be bulk erased before programming. If the device fails to erase, display message **F: “Device Failed To Erase”** and terminate the programming sequence. To erase the device, follow the flow in Figure 10, and the algorithm detailed in Table 20. Note that all NV-memory (user configuration bits , security bits , Done bits and signature string) are erased with this operation. Recall that an ISC_ENABLE instruction must be executed prior to executing the erase operation. If blank check (see Table 22) fails after performing the erase operation, then display the message **“F: Device Failed to Erase** and terminate the sequence with an ISC_DISABLE instruction.

4.4 Device Programming

Only after successful erase can the actual programming flow begin as illustrated in Figure 9. The address and data are first loaded before a short programming pulse is applied to the device. The programming algorithm itself is detailed in Table 24.

4.4.1 Assembling Data in the ISC Register for Programming

The programming data is contained in the JEDEC file generated by Xilinx software. The JEDEC file adheres to the format of JEDEC standard JESD3-C. The programming data is formatted in L records in the JEDEC file. Consider the following JEDEC file fragment:

```
L000000 01110111*
L000008 01111101*
L000016 01110111*
L000024 01111101*
L000032 01110111*
L000040 01110111*
L000048 01110111*
L000056 01111101*
```

The decimal number immediately following the L is the fuse number. Looking at the first line " L000000 01110111", the first fuse number for the first (leftmost) bit is 0. Fuse number 0 have a logic value of 0 associated with it. The fuse numbers increment by one as you proceed to the right. Fuse number 1 has a logic value of 1 associated with it, fuse 2 - logic 1, fuse 3 - logic 1, fuse 4 - logic 0 and so on. The end of line character is the asterisk (*).

The Excel fuse map file indicates the bit position of each bit of fuse data in the ISC Register. Each column in the fuse map represents a device address. Each row represents a bit in the ISC Register. The leftmost column is ordinal address 0. The uppermost row is ISC Register data bit location 0.

The fuse data from the location specified in the first row of the column is assigned to the first valid data bit in the ISC Shift Register. For the xc2c32, this would be the data from fuse location 5696. The first valid data bit is position 1 since position 0 holds a transfer bit for the xc2c32 (indicated by a blank row in the fuse map). This means that the fuse data from location 5696 in the JEDEC file is loaded into ordinal address 0's data bit location 1. This mapping continues for each bit in each address location. So data bit location 2 is filled with the fuse data from location 5697, data bit 3 is filled with fuse data 5698, etc. You must always remember to load the transfer bit locations with a logic 0 during programming.

Note: The XC2C64 and XC2C64A have no transfer bits.

You must also remember to translate address values from their ordinal position to their appropriate grey code value according to Table 10. Figures 2 and 3 illustrate the assembly of address and data information into the configuration register.

This programming procedure should be followed by the verify procedure. An ISC_ENABLE command is executed at the beginning of the programming procedure, and an ISC_DISABLE is executed at the end of the verify procedure. The number of address and data bits varies from device to device. Specific address information of all addresses and the number of data bits can be found in Tables 2 and 10.

4.4.2 Example of Assembling Data into the ISC Shift Register

In the following examples, the highest order bit is shifted in first.
NOTE: 0's must be inserted in the Transfer Bit locations to correctly activate those cells.

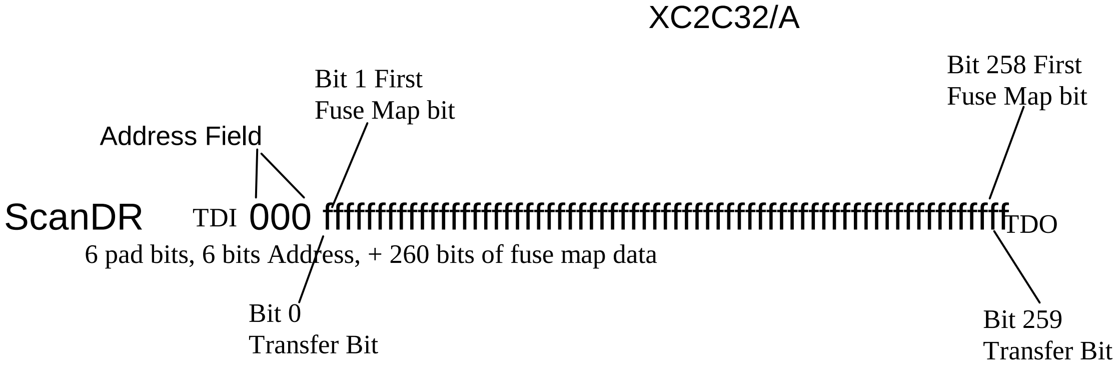

XC2C32/A
Bit 1 First
Fuse Map bit
Bit 258 First
Fuse Map bit
Address Field
ScanDR
TDI
000 ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
TDO
6 pad bits, 6 bits Address, + 260 bits of fuse map data
Bit 0
Transfer Bit
Bit 259
Transfer Bit

Figure 2. Assembling Data into the ISC Shift Register (XC2C32/A)

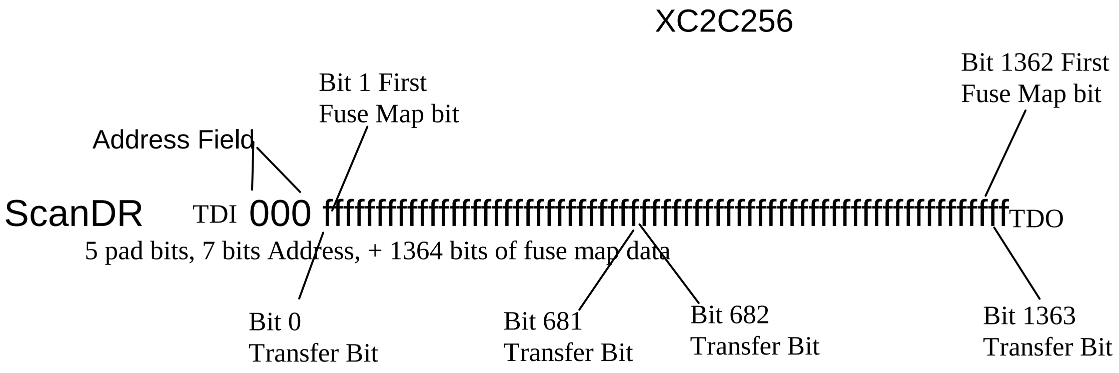


Figure 3. Assembling Data into the ISC Shift Register (XC2C256)

4.4.3 Programming XC2C32A and XC2C64A Devices

4.4.3.1 Programming the XC2C32A

The XC2C32A is an extension of the XC2C32. IO Banking has been added to the XC2C32A. The XC2C32 does not have IO Banking. This feature required the addition of four memory bits to control the input and output voltage standards of the two IO Banks of the new XC2C32A. As a result of these additional bits, the JEDEC file for the XC2C32A has four more fuses than the JEDEC file of the XC2C32. The Fuse Map for the XC2C32A also reflects this.

It is the intention of Xilinx to make the programming of the XC2C32A backward compatible to the XC2C32. **To be more specific, the XC2C32A should be programmable by a customer with either a XC2C32 JEDEC file or a XC2C32A JEDEC file**. Xilinx's intention is to avoid inconveniencing customers that have designed the XC2C32 into their systems. When they begin receiving XC2C32A 's, **these customers should be able to program the XC2C32A's just as they were programming the XC2C32's without IO Banking, using the same JEDEC file and not having to re-fit their patterns.** Obviously if a customer wants to take advantage of the XC2C32A's IO Banking feature, the XC2C32A JEDEC file will be required.

A versus Non-A JEDEC essentials:

The XC2C32A is treated exactly like a XC2C32 when using an XC2C32 JEDEC file; specifically treating it as having 12274 fuses instead of 12278 (12274+4). Specifically, this means that when using a JEDEC file targeted to an XC2C32 to configure an XC2C32A device, the following is true:

a) Programming is executed across 12274 fuses
b) The checksum is calculated across 12274 fuses
c) Verification is executed across 12274 fuses
d) Readback is executed across 12274 fuses
e) It should be noted that the device's erase function operates on all fuses, and the erased fuse value of fuse locations 12275 to 12278 ensures that their associated features are disabled

4.4.3.2 XC2C32A IDCODE

The IDCODEs are different to allow electrical identification of the two devices.

Bit 0 = Mandatory '1'
Bits 1-11 = Manufacture ID
Bits 12-14 = Package ID
Bit 15 = Voltage
Bits 16-21 = Macrocells
Bits 22-24 = Technology
Bits 25-27 = Architecture
Bits 28-31 = Version (These are X'ed out) in the BSDL files.

```
                     Bit     0  |  1        -     11 |12-14|15| 16  - 21| 22-24| 25-27|28 - 31
XC2C32    IDCODE =  1  |  10010010000  |  pkg | 1 | 100000 |  110  |  110  | XXXX

                     Bit    0  |  1        -     11 |12-14|15| 16  - 21| 22-24| 25-27|28 - 31
XC2C32A IDCODE =  1  |  10010010000  |  pkg | 1 | 100001 |  110  |  110  | XXXX
```

NOTE that the IDCODES differ only in Bits 16 – 21 (Macrocells). Bit 21 designates IO Banking in the XC2C32A.

4.4.3.3 Programming the XC2C64A

The XC2C64A is an extension of the XC2C64. IO Banking has been added to the XC2C64A. The XC2C64 does not have IO Banking. This feature required the addition of four memory bits to control the input and output voltage standards of the two IO Banks of the new XC2C64A. As a result of these additional bits, the JEDEC file for the XC2C64A has four more fuses than the JEDEC file of the XC2C64. The Fuse Map for the XC2C64A also reflects this.

It is the intention of Xilinx to make the programming of the XC2C64A backward compatible to the XC2C64. **To be more specific, the XC2C64A should be programmable by a customer with either a XC2C64 JEDEC file or a XC2C64A JEDEC file**. Xilinx's intention is to avoid inconveniencing customers that have designed the XC2C64 into their systems. When they begin receiving XC2C64A 's, **these customers should be able to program the XC2C64A's just as they were programming the XC2C64's without IO Banking, using the same JEDEC file and not having to re-fit their patterns.** Obviously if a customer wants to take advantage of the XC2C64A's IO Banking feature, the XC2C64A JEDEC file will be required.

A versus Non-A JEDEC essentials:

The XC2C64A is treated exactly like a XC2C64 when using an XC2C64 JEDEC file; specifically treating it as having 25808 fuses instead of 25812 (25808+4). Specifically, this means that when using a JEDEC file targeted to an XC2C64 to configure an XC2C64A device, the following is true:

a) Programming is executed across 25808 fuses
b) The checksum is calculated across 25808 fuses
c) Verification is executed across 25808 fuses
d) Readback is executed across 25808 fuses
e) It should be noted that the device's erase function operates on all fuses, and the erased fuse value of fuse locations 25809 to 25812 ensures that their associated features are disabled

4.4.3.4 XC2C64A IDCODE

The IDCODEs are different to allow electrical indentification of the two devices.

Bit 0 = Mandatory ‘1’
Bits 1-11 = Manufacture ID
Bits 12-14 = package ID
Bit 15 = Voltage
Bits 16-21 = Macrocells
Bits 22-24 = Technology
Bits 25-27 = Architecture
Bits 28-31 = Version (These are X’ed out) in the BSDL files.

```
                  Bit      0  | 1        -      11 |12-14|15| 16  - 21| 22-24| 25-27|28 - 31
XC2C64    IDCODE =  1  |  10010010000  |  pkg | 1 | 101000 |  110  |  110  | XXXX

                  Bit      0  | 1        -      11 |12-14|15| 16  - 21| 22-24| 25-27|28 - 31
XC2C64A IDCODE =  1  |  10010010000  |  pkg | 1 | 101001 |  110  |  110  | XXXX
```

NOTE that the IDCODES differ only in Bits 16 – 21 (Macrocells). Bit 21 designates IO Banking in the XC2C64A.

4.5 Post-Program Verify

After programming all the addresses of the device, perform data verification, as illustrated in Figure 9, and detailed in Table 26. This verification procedure should follow the programming procedure.

4.5.1 Assembling Data in the ISC Shift Register for Verification

During verification, the ISC Shift Register serves as an address register when shifted in and a data retrieval register when shifted out of the device. During verification, the device only uses the address bits. The entire ISC Shift Register must be shifted in with each shift including the grey code value for the address location to be read. The data bits in the ISC Shift Register can be set to any value as they are shifted in as they are not used.

After the read is completed, and the ISC Shift Register is shifted out, the data in the ISC Shift Register must be reverse mapped using the fuse map files to effect comparison against the data contents of the source JEDEC file. Basically, you must use the reverse of the technique listed in section 4.4.1. In this case, you would map from the ISC Shift Register bit location, to the fuse number in the JEDEC file instead of the other way around.

If any cell fails to verify, display the message **G: "Device Failed To Program"** and terminate the programming sequence. Note that this operation can be performed at any time.

4.6 Secure Device / Program DONE Bits

Following the programming and post-program verify sequence, **the algorithm must program the DONE bits as directed in Table 28 (see truth table for done bits in Figure 8)** and optionally secure the device, as illustrated in Figure 11.

4.6.1 Done Bits

The DONE bits is a feature that is required and enables initialization of the device. Initialize forces configuration of the device to that loaded in the NV-memory without power cycling.

Only if the done bits are correctly programmed is the device considered programmed. See Figure 8 and Table 7 for DONE BITS decoded state meanings.

Note: The JEDEC file DOES NOT contain fuses for these DONE bits. The DONE bits must be set with an independent program operation to the address shown in Table 10 and must be the last address programmed during the programming operation, preferably after verify completes. See Table 28 for a detailed description of the done bit programming algorithm.

DONE bit programming allows successful completion of initialization. After DONE bit programming, the application will display the message **K: "Device Programming Done"**.
After programming, the DONE bit must be verified as programmed as detailed in Table 30. If the device bits fail to verify Done, display the message **M: "Device Done verify failed"** and the programming sequence should be terminated with an ISC_DISABLE.

4.6.1.1 XC2C32/A Done and Security Bit Programming Example

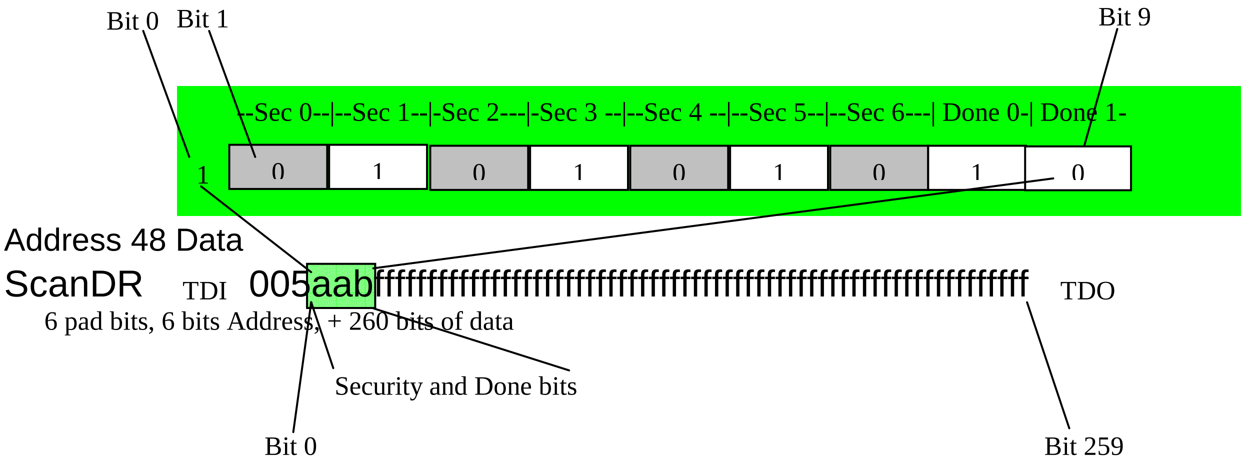

Bit 0
Bit 1
Bit 9
--Sec 0--|--Sec 1--|-Sec 2---|-Sec 3 --|--Sec 4 --|--Sec 5--|--Sec 6---| Done 0-| Done 1-
1
0
1
0
1
0
1
0
1
0
Address 48 Data
ScanDR
TDI
005aabffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
TDO
6 pad bits, 6 bits Address, + 260 bits of data
Security and Done bits
Bit 0
Bit 259

Figure 4. Programming Security and Done Bits in XC2C32/A CoolRunner-II Devices

The difference between the XC2C32/A and the 128, 256, 384 and 512 is the length of the data register and the address of the Security and Done bits. See Table 2 for details. The Security and Done bits are located at Bit 1 through Bit 9.

4.6.1.2 The XC2C64/A Done and Security Bit Programming Example

The XC2C64/A is unique from the other members of the CoolRunner-II family. Use the example below as a reference to program the Security and Done bits.

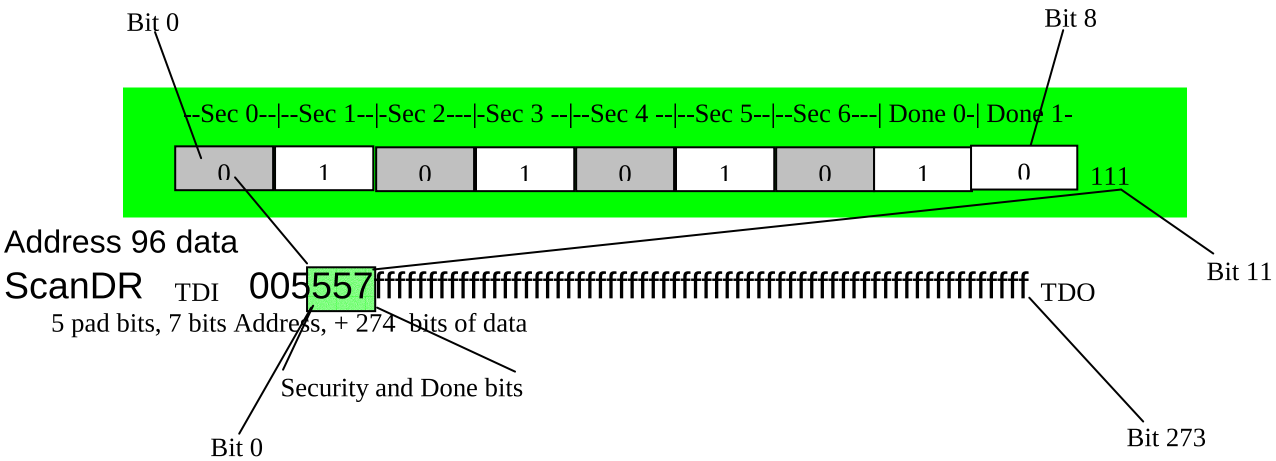

Bit 0
Bit 8
--Sec 0--|--Sec 1--|-Sec 2---|-Sec 3 --|--Sec 4 --|--Sec 5--|--Sec 6---| Done 0-| Done 1-
0
1
0
1
0
1
0
1
0
111
Bit 11
Address 96 data
ScanDR
TDI
005557ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
TDO
5 pad bits, 7 bits Address, + 274 bits of data
Security and Done bits
Bit 0
Bit 273

Figure 5. Programming Security and Done Bits in XC2C64/A CoolRunner-II Devices

NOTE: Bit 0 is not skipped for the XC2C64/A Security / Done bit locations.

4.7 Read Security Address

A security feature is available which when programmed, disables access to the information contained in the non-volatile EPROM array. Any operation reading the user array will result in unknown data returned and should not be allowed. A 'read secured' device can still be erased and only then, reprogrammed.

Note: The JEDEC file DOES NOT contain fuses for the security bits. The security bits must be set with an independent operation to the address shown in Table 10.

Figures 4 and 5 illustrate the configuration register data for Security / Done programming.

If you elect not to secure the device, display the message **H: "Device Not Secured"**.

If you choose to secure the device, you must program the security bit (see Figure 11), as directed in Table 32. Then, you must verify that the security bit is set as detailed in Table 34. While checking the device for read security before an operation, if the Security address is programmed, consider the device read secured. You will read all logical ones ('1,1,1,1,1,1,1') if read is permitted or ('0,1,0,1,0,1,0') if read secured. If the device fails to verify as secure, display the message **J: Device Failed To Secure"** and terminate the programming sequence with an ISC_DISABLE instruction. If successful, display the message **D: "Device Secured"** and terminate the programming sequence with an ISC_DISABLE instruction. This procedure should follow the programming sequence without an ISC_DISABLE command being included until completing the done bits and the security programming.

If a device is Secured, the signature string, manufacturer's code and product code, Security bits and Done bits can still be read. All other data can not be read. Table 1 shows which operations may be performed after a device is Secured.

Note that the device is not secured after programming the security bit until either an ISC_INIT instruction is executed or the power is cycled.

Table 1: Valid Operations on a Secured Device*

Operation	Valid
Program	No
Erase	Yes
Verify	No
Blank Check	No
Signature String (USERCODE)	Yes
Mfg/Product Code (IDCODE)	Yes
Security bits	Yes
Done bits	Yes

*JTAG instructions USERCODE and IDCODE will read these bits.

4.8 Signature String Address (USERCODE)

The signature string is a user definable code which can be programmed at any time until the security bit is set and initialized into SRAM. The programmer should allow reading and displaying of the string. These data are erased with erase operation and programmed with the same procedure used for ordinary configuration bits. The user should be able to define the string during a programming session. The address and bit positions that are used to program the USERCODE bits are detailed in the device fuse map files.

4.8.1 Example: Programming a USERCODE in XC2C32/A CoolRunner-II Devices

Suppose we are trying to load the following USERCODE value: “1 2 3 4 5 6 7 8”
The desired USERCODE in binary is: 0001 0010 0011 0100 0101 0110 0111 1000

The target USERCODE bit locations read from the Fuse Map files are: 249, 248, 247, 246, 245, 244, …. 218. In Address 49 (the Fuse Map for this family is available in Excel Spead Sheet files). The resulting configuration register contents are illustrated in Figure 6.

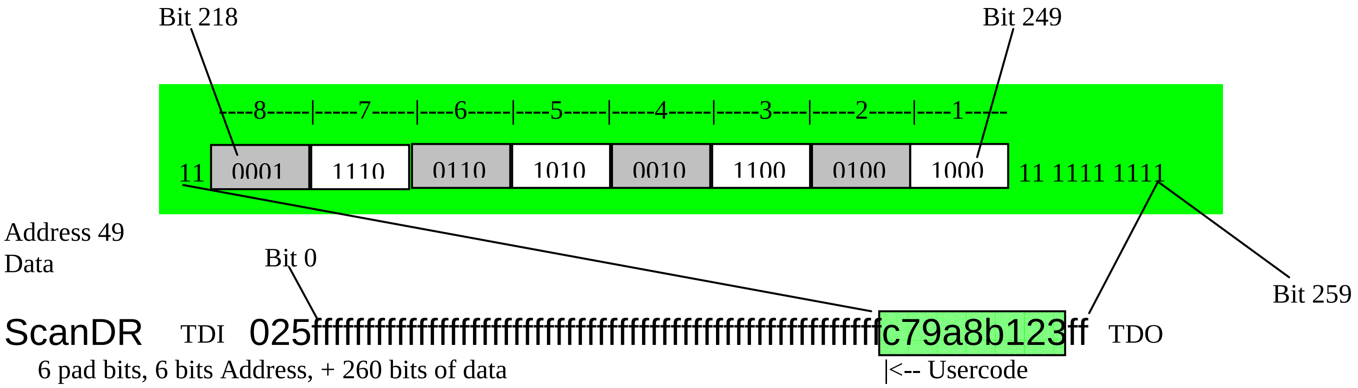


Figure 6. USERCODE in XC2C32/A

In summary, the procedure to set USERCODE bits is as follows:

Step 1: Convert USERCODE from Hex to Binary.
Step 2: Get USERCODE row address and Bit locations from map file.
Step 3: Convert USERCODE row address to Grey code. Set first six bits to the USERCODE row address.
Step 4: Set the rest of the bits (6-259, in the above example to 1).
Step 5: Overlay USERCODE bits at the locations specified in the map file (bits 249-218 in the above example).
Step 6: The USERCODE data can then be programmed to the device in a manner similar to any other program operation as detailed in Table 24.

4.8.2 Example: Programming a USERCODE in XC2C64/A CoolRunner-II Devices

The XC2C64/A USERCODE is unique from the other members of the CoolRunner II family.
Please Use This Example for Programming the XC2C64/A Only

Suppose we are trying to load the following USERCODE value: "1 2 3 4 5 6 7 8"
The desired USERCODE in binary is: 0001 0010 0011 0100 0101 0110 0111 1000

The target USERCODE bit locations read from the fuse map file are 242, 243, 244, 245, 246, 247, …. 273 in address 97 (the Fuse Map for this member is available in Excel Spead Sheet files). The resulting configuration data register contents are illustrated in Figure 7. **NOTE: this is backwards from the other members of the family.**

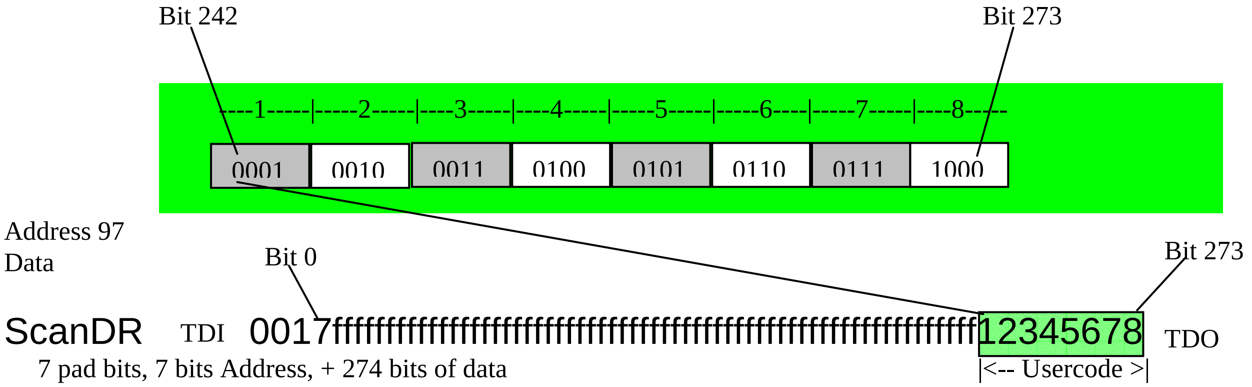


Figure 7. USERCODE in XC2C64/A

In summary, the procedure to set USERCODE bits in the XC2C64/A is:

Step 1: Convert USERCODE from Hex to Binary.
Step 2: Get USERCODE row address and Bit locations from map file.
Step 3: Convert USERCODE row address to Grey code. Set first seven bits to the USERCODE row address.
Step 4: Set the rest of the bits , in the above example to 1).
Step 5: Overlay USERCODE bits at the locations specified in the map file (bits 242-273 in the above example).
Step 6: The USERCODE data can then be programmed to the device in a manner similar to any other program operation as detailed in Table 24.

Table 2. Number of Addresses, Shift Data Register, Length and Transfer Bit Locations for CoolRunner-II Family

Part Number	Number of Address bits	Number of Addresses	Number of Shift Register bits	Tranfer Bit location
XC2C32/A	6	0-47 user, 48 sec/done, 49 UES	260	0,259
XC2C64/A	7	0-95 user, 96 sec/done, 97 UES	274	
XC2C128	7	0-79 user, 80 sec/done, 81 UES	752	0,375,376,751
XC2C256	7	0-95 user, 96 sec/done, 97 UES	1364	0,681,682,1363
XC2C384	7	0-119 user, 120 sec/done, 121 UES	1868	0,933,934,1867
XC2C512	8	0-159 user, 160 sec/done, 161 UES	1980	0,989,990,1979

Code	Messages
A	Incompatible Adapter or File for Current Algorithm
B	Manufacturer's Code Error
C	Product Code Error
D	Device Secured
E	Device Not Blank
F	Device Failed To Erase
G	Device Failed To Program
H	Device Not Secured
I	Device Failed To Verify
J	Device Failed To Secure
K	Device programming Done
L	Device programming not Done
M	Device Done verify failed

DONE BITS

Done 1	Done 0	Function
0	0	NA
0	1	Programed
1	0	NA
1	1	Blank

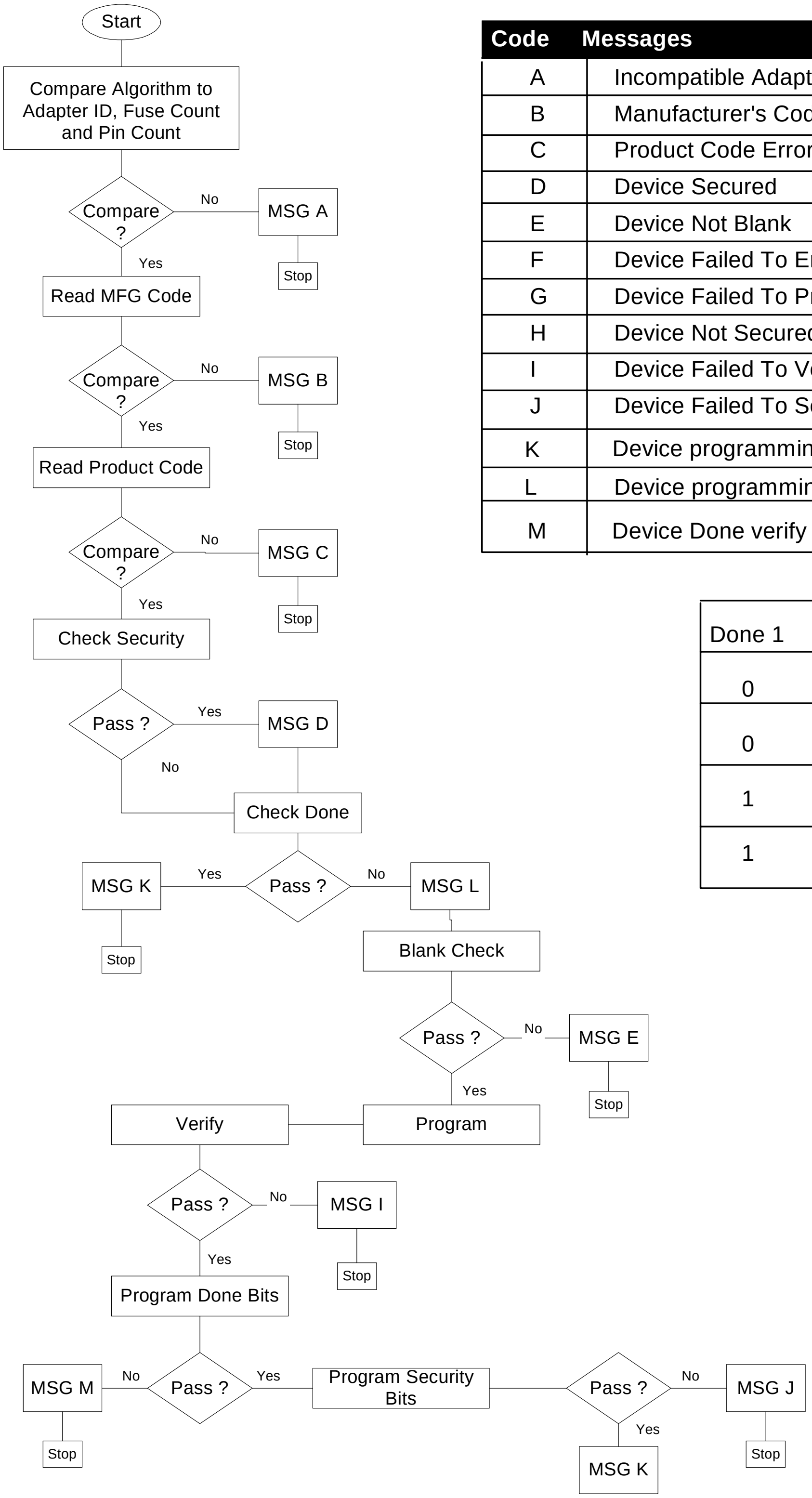


Figure 8. Overall Programming Sequence

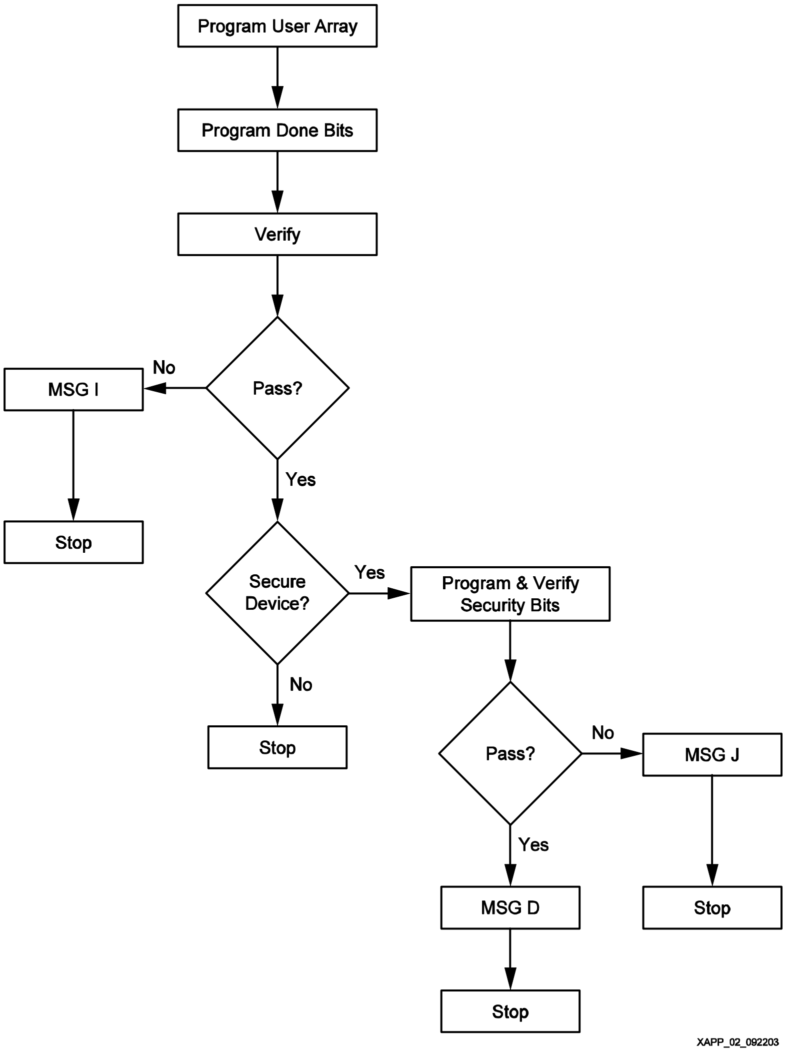


Figure 9. Programming Flow

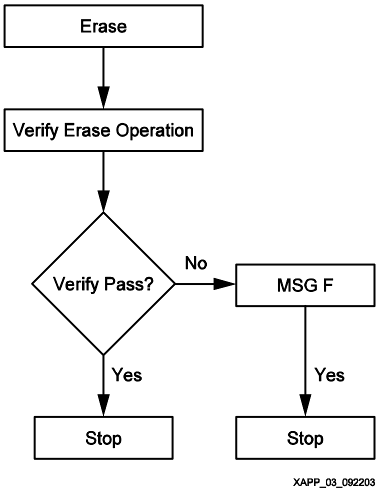


Figure 10. Erase Flow

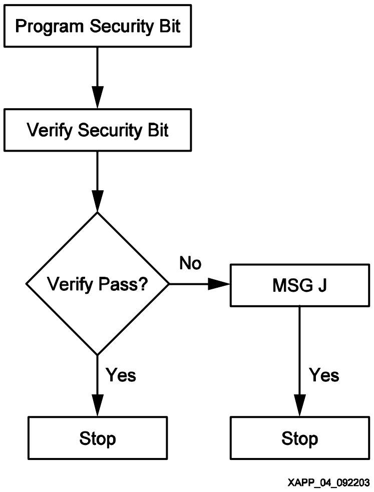


Figure 11. Security Bit Programming

Table 3. Common DC Programming and Erase Characteristics T_A = 25°C ± 5°C

Symbol	Description	Limits		
		Min	Max	Units
I_{IL}	Input Leakage		10	µA
I_{CC}	V_{CC} Supply Current		10	mA
V_{IL}	Low-Level Input Voltage	0	0.3 x Vcc	V
V_{IH}	High-Level Input Voltage	0.7 x Vcc	Vcc	V
V_{OL}	Low-Level Output Voltage		x	V
V_{OH}	High-Level Output Voltage	x		V
V_{CCBNK}	V_{CC}	1.71	1.89	V

Note: Xilinx recommends that the mean be used whenever possible.
Note: ISC operations guaranteed only over the Commercial Temperature operating range of Ta = 0 C – 70 C

Table 4. Manufacturer ID Codes

Manufacturer	**ID code** (Bit Position 21-1)
Xilinx	00001001001

Table 5. Device ID Codes

Product	Package	**Architecture Code** (Bit Position 27-25)	**Macrocell Count Code** (Bit Position 21-16)	**Package Code** (Bit Position 14-12)
XC2C32	VQ44	011	000001	100
XC2C32	PC44	011	000001	101
XC2C32	CP56	011	000001	011
XC2C32A	QFG32	011	100001	001
XC2C32A	VQ44	011	100001	100
XC2C32A	PC44	011	100001	101
XC2C32A	CP56	011	100001	011

XC2C64	VQ44	011	000101	110
XC2C64	PC44	011	000101	010
XC2C64	CP56	011	000101	101
XC2C64	VQ100	011	000101	100
XC2C64A	QFG48	011	100101	001
XC2C64A	VQ44	011	100101	110
XC2C64A	PC44	011	100101	010
XC2C64A	CP56	011	100101	101
XC2C64A	VQ100	011	100101	100
XC2C128	VQ100	011	011000	010
XC2C128	CP132	011	011000	011
XC2C128	TQ144	011	011000	100
XC2C128	FT256	011	011000	110
XC2C256	VQ100	011	010100	010
XC2C256	PQ208	011	010100	101
XC2C256	TQ144	011	010100	100
XC2C256	CP132	011	010100	011
XC2C256	FT256	011	010100	110
XC2C384	FG324	011	010101	010
XC2C384	TQ144	011	010101	100
XC2C384	PQ208	011	010101	101
XC2C384	FT256	011	010101	110
XC2C512	FG324	011	010111	010
XC2C512	PQ208	011	010111	100
XC2C512	FT256	011	010111	110

Table 6. AC Programming Specifications

Symbol	Description	Limits		
		Min	Max	Units
T_{PWPGM}	Program Pulse Width	10	-	ms
T_{ERASE}	Erase Pulse Width	100	-	ms

Table 7. IEEE STD 1532 Modal States

Modal State	ISC_DISABLE	ISC_ENABLE	DONE1	DONE0
Operational Modal State	0	0	0	1
ISC Accessed Modal State	0	1	X	X
Unprogrammed Modal State (BLANK)	0	0	1	1
ISC Complete Modal State	1	0	X	X

The security bit is also reflected in the instruction register capture bits. Each time the TAP controller is clocked through the CIR (Capture Instruction Register) state the Instruction Register will be loaded with the bits indicated in Figure 12. These bits can then be shifted out on TDO for examination.

TDI | ”0” | ”0” | ISC_DIS | ISC_EN| SEC | DONE | “0” | “1” | TDO

Figure 12. Capture Bits in Instruction Register

5.0 Package Information

Table 8. Device Packages

Packages	XC2C32	XC2C32A	XC2C64	XC2C64A	XC2C128	XC2C256	XC2C384	XC2C512
QFG32		X						
VQ44	X	X	X	X				
PC44	X	X	X	X				
QFG48				X				
CP56	X	X	X	X				
VQ100			X	X	X	X		
CP132					X	X		
TQ144					X	X	X	
PQ208						X	X	X
FT256						X	X	X
FG324							X	X

6.0 Address Information

6.1 Erase Addresses (for Bulk Erase)

No address required for bulk erase; erase is performed with the ISC_ERASE instruction.

Note: When reading an erased device, logical one ('1') is the erased bit logical value. Logical zero ('0') is the programmed bit logical value.

6.2 Grey Code Addressing

When programming a CoolRunner II device, each address requires programming pulse be applied to the device (also see programming specification). All data must be loaded in the shift register (bits D0-273 for 64MC) for each address; however, don't cares (X) exist in some addresses. Tabulated below is the address decimal sequence in the first column, and the binary Gray code count address sequence in the second column. The “don’t cares” can be identified from the fuse map for each family member.

Table 10. Grey Code Addressing

Address	512MC -------	---------	---------	----------	----------	----------	512MC
	64-384MC	---------	64MC	128MC	256MC	384MC	
	32MC	32MC					
0	0 0 000000	6 bits	7 Bits	7 Bits	7 Bits	7 Bits	8 Bits
1	0 0 000001						
2	0 0 000011						
3	0 0 000010						
4	0 0 000110						
5	0 0 000111						
6	0 0 000101						
7	0 0 000100						
8	0 0 001100						
9	0 0 001101						
10	0 0 001111						
11	0 0 001110						
12	0 0 001010						
13	0 0 001011						
14	0 0 001001						
15	0 0 001000						
16	0 0 011000						
17	0 0 011001						
18	0 0 011011						
19	0 0 011010						
20	0 0 011110						
21	0 0 011111						
22	0 0 011101						
23	0 0 011100						

24	0 0 010100						
25	0 0 010101						
26	0 0 010111						
27	0 0 010110						
28	0 0 010010						
29	0 0 010011						
30	0 0 010001						
31	0 0 010000						
32	0 0 110000						
33	0 0 110001						
34	0 0 110011						
35	0 0 110010						
36	0 0 110110						
37	0 0 110111						
38	0 0 110101						
39	0 0 110100						
40	0 0 111100						
41	0 0 111101						
42	0 0 111111						
43	0 0 111110						
44	0 0 111010						
45	0 0 111011						
46	0 0 111001						
47	0 0 111000						
48	0 0 101000	Sec, Done					
49	0 0 101001	UES					
50	0 0 101011						
51	0 0101010						
52	0 0101110						
53	0 0101111						
54	0 0101101						
55	0 0101100						
56	0 0100100						
57	0 0100101						
58	0 0100111						
59	0 0100110						
60	0 0100010						
61	0 0100011						
62	0 0100001						
63	0 0100000						
64	0 1100000						
65	0 1100001						
66	0 1100011						
67	0 1100010						
68	0 1100110						
69	0 1100111						
70	0 1100101						
71	0 1100100						
72	0 1101100						

73	0 1101101		▒	░	▓	▒	░
74	0 1101111		▒	░	▓	▒	░
75	0 1101110		▒	░	▓	▒	░
76	0 1101010		▒	░	▓	▒	░
77	0 1101011		▒	░	▓	▒	░
78	0 1101001		▒	░	▓	▒	░
79	0 1101000		▒	░	▓	▒	░
80	0 1111000	---------	▒	Sec, Done	▓	▒	░
81	0 1111001		▒	UES	▓	▒	░
82	0 1111011		▒		▓	▒	░
83	0 1111010		▒		▓	▒	░
84	0 1111110		▒		▓	▒	░
85	0 1111111		▒		▓	▒	░
86	0 1111101		▒		▓	▒	░
87	0 1111100		▒		▓	▒	░
88	0 1110100		▒		▓	▒	░
89	0 1110101		▒		▓	▒	░
90	0 1110111		▒		▓	▒	░
91	0 1110110		▒		▓	▒	░
92	0 1110010		▒		▓	▒	░
93	0 1110011		▒		▓	▒	░
94	0 1110001		▒		▓	▒	░
95	0 1110000		▒		▓	▒	░
96	0 1010000	---------	Sec, Done	----------	Sec, Done	▒	░
97	0 1010001		UES		UES	▒	░
98	0 1010011					▒	░
99	0 1010010					▒	░
100	0 1010110					▒	░
101	0 1010111					▒	░
102	0 1010101					▒	░
103	0 1010100					▒	░
104	0 1011100					▒	░
105	0 1011101					▒	░
106	0 1011111					▒	░
107	0 1011110					▒	░
108	0 1011010					▒	░
109	0 1011011					▒	░
110	0 1011001					▒	░
111	0 1011000					▒	░
112	0 1001000					▒	░
113	0 1001001					▒	░
114	0 1001011					▒	░
115	0 1001010					▒	░
116	0 1001110					▒	░
117	0 1001111					▒	░
118	0 1001101					▒	░
119	0 1001100					▒	░
120	0 1000100	---------	---------	----------	----------	Sec,	░

						Done	
121	0 1000101					UES	
122	0 1000111						
123	01000110						
124	01000010						
125	01000011						
126	01000001						
127	01000000						
128	11000000						
129	11000001						
130	11000011						
131	11000010						
132	11000110						
133	11000111						
134	11000101						
135	11000100						
136	11001100						
137	11001101						
138	11001111						
139	11001110						
140	11001010						
141	11001011						
142	11001001						
143	11001000						
144	11011000						
145	11011001						
146	11011011						
147	11011010						
148	11011110						
149	11011111						
150	11011101						
151	11011100						
152	11010100						
153	11010101						
154	11010111						
155	11010110						
156	11010010						
157	11010011						
158	11010001						
159	11010000						
160	11110000	---------	---------	----------	----------	----------	Sec, Done
161	11110001						UES
162	11110011						

Table 11. Instruction Op Codes

Code	Instruction	Register used
00000000	EXTEST	Boundary-Scan
00000001	IDCODE	Device ID
11111101	USERCODE	Device ID

00000011	Sample/Preload	Boundary-Scan
00000010	INTEST	Boundary-Scan
11111100	HIGHZ	Bypass
11111010	CLAMP	Bypass
11111111	BYPASS	Bypass
11100110	ISC_Sram_Write	ISC Shift
11100100	ISC_Enable_OTF	ISC Shift
11101000	ISC_Enable	ISC Shift
11101001	ISC_Enable_Clamp	ISC Shift
11101101	ISC_Erase	ISC Shift
11101010	ISC_Program	ISC Shift
11100000	ISC_NOOP	Bypass
11101110	ISC_READ	ISC Shift
11110000	ISC_Init	ISC Shift
11100111	ISC_Sram_Read	ISC Shift
11000000	ISC_Disable	ISC Shift
	PRIVATE Instructions	
00010001	Private1	
00010010	Private2	
00010011	Private3	
00010100	Private4	
00010101	Private5	
00010110	Private6	

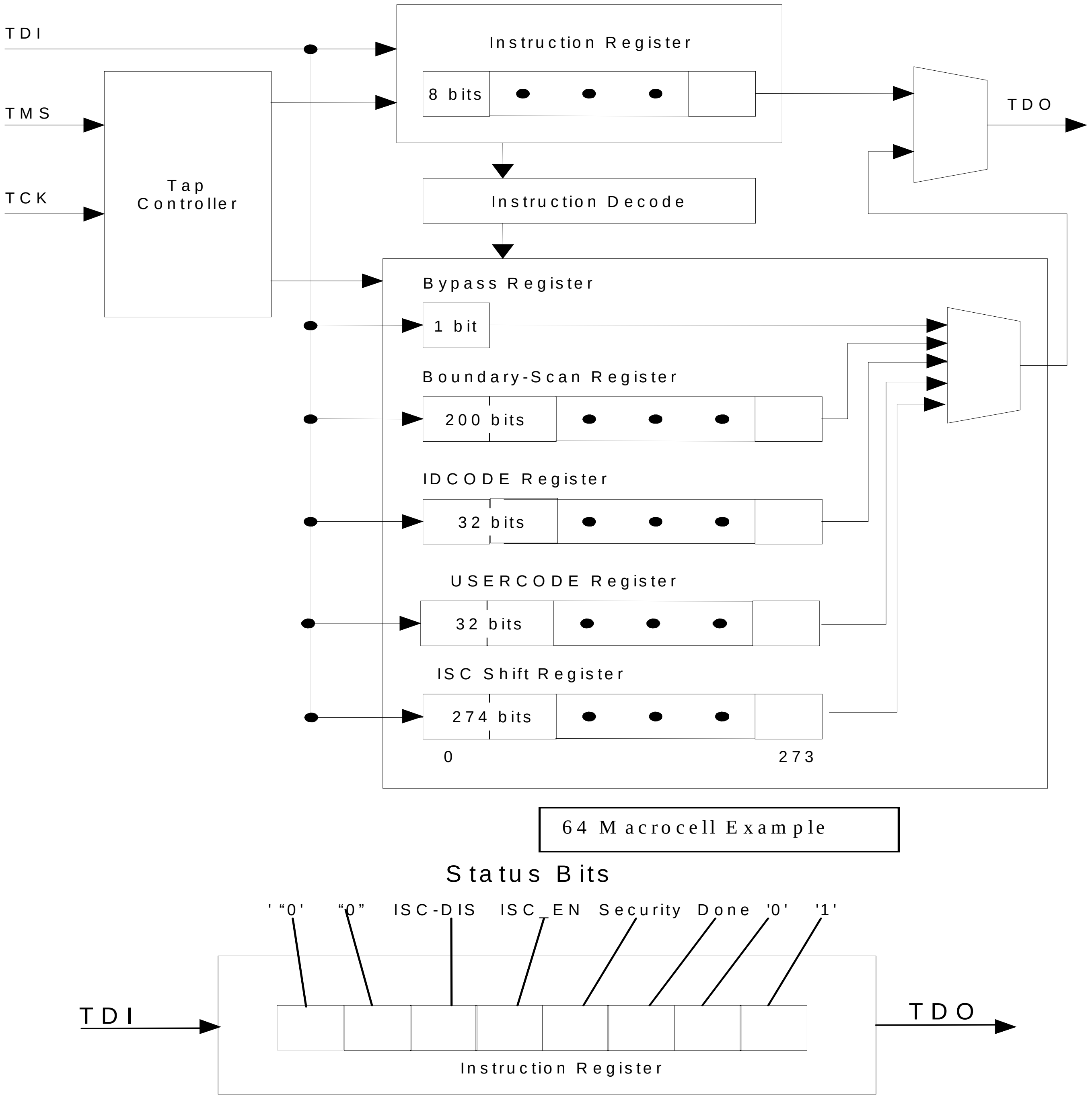


Figure 13. JTAG Block Diagram

Table 12. JTAG Register Description (64MC Example)

Register	# of bits	Register Description
Instruction Register	8	The Instruction Register is a shift-register-based design which allows an instruction to be shifted into a device. The instruction shifted into the register is latched at the completion of the shifting process. The instruction is used to select the BST or ISC operation and/or the data register to be accessed. The parallel output from the Instruction Register is latched to ensure that the BST and ISC logic is protected from the transient data patterns that will occur in its shift-register stages as new instruction data is entered.
Bypass Register	1	The Bypass Register contains a single shift-register stage and is used to provide a minimum length serial path between the TDI and TDO pins of a component when no test or program operation of that component is required. This allows more rapid movement of test/program data to and from other components on a circuit pack that are required to perform test/program operations.
Boundary Scan Register	192	The Boundary-Scan Register allows testing of circuitry external to the CPLD and also permits the system signals flowing into and out of the CPLD logic to be sampled and examined without causing interference with the normal operation of the CPLD logic. The Boundary-Scan Register is a long shift register composed of all the Boundary-Scan cells at the pins of the device.
IDCODE Register	32	This is a 32 bit shift-register, parallel-in and serial out. The register contains the following information: Bit(s) Usage 0 1 - pre-defined 1-11 Manufacturing Identity 12-27 Part Number 28-31 Version
USERCODE Register	32	This is a 32 bit shift-register, parallel-in and serial out. The register contains user information.
ISC Shift Register	7 address bits 274 data bits 281 bits total	Used to address the EEPROM row and contains the data that is being written into or read from the EEPROM array.

Table 13. JTAG Pin Description

Pins	Name	Description
TCK	Test Clock Input	Clock pin to shift the serial data and instructions in and out of the TDI and TDO pins, respectively. TCK is also used to clock the TAP Controller state machine.
TMS	Test Mode Select	Test mode select pin selects the JTAG instruction mode. TMS should be driven high during user mode operation.
TDI	Test Data Input	Serial input pin for instructions and test data. Data is shifted in on the rising edge of TCK.
TDO	Test Data Output	Serial output pin for instructions and test data. Data is shifted out on the falling edge of TCK. The signal is tri-stated if data is not being shifted out of the device.

Table 14. Supported Low Level JTAG Boundary-Scan Commands

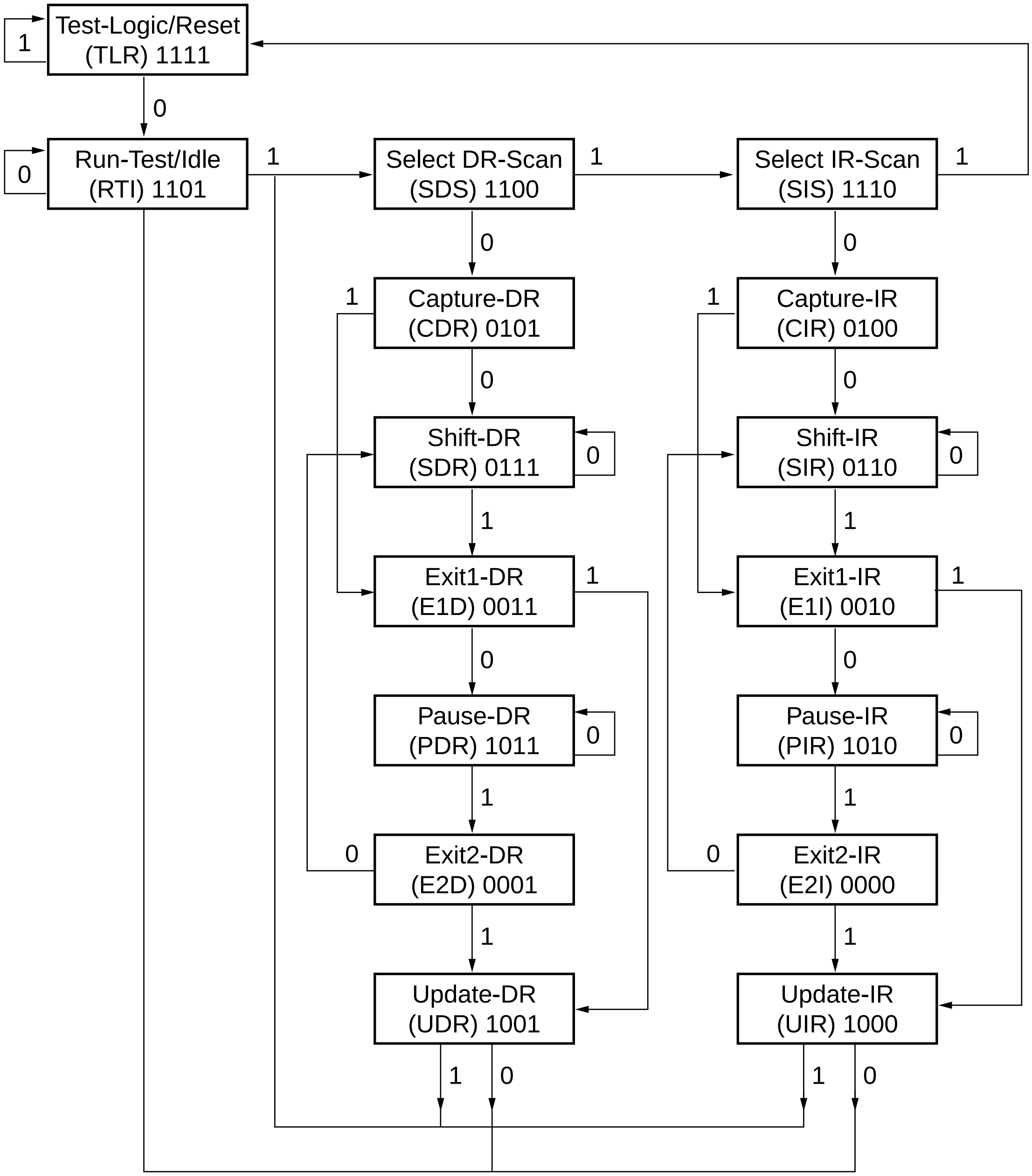
Instruction **(Instr. Code)** ***Register Used***	**Description**
BYPASS (11111111) *Bypass Register*	Places the 1 bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through the selected device to adjacent devices during normal device operation without affecting the operation of the IC.
EXTEST (00000000) *Boundary-Scan Register*	Puts the IC into external mode. Forces data externally from the boundary scan outputs and receives data externally to the boundary scan inputs.
IDCODE (00000001) *IDCODE Register*	Selects the IDCODE register and places it between TDI and TDO, allowing the IDCODE to be serially shifted out of TDO. The IDCODE instruction permits blind interrogation of the components assembled onto a circuit pack. Thus, in circumstances where the component population may vary, it is possible to determine what components exist in a product.
USERCODE (11111101) *USERCODE Register*	Allows IC to remain in normal mode and Selects the USERCODE register and places it between TDI and TDO, allowing the USERCODE to be serially shifted out of TDO.
SAMPLE/PRELOAD (00000011) *Boundary-Scan Register*	Allows IC to remain in normal mode and selects the boundary scan register to be connected between TDI and TDO. Allows sampling data entering and leaving the device. It allows preload data into the boundary scan register before executing the EXTEST instruction.
INTEST (00000010) *Boundary-Scan Register*	Puts the IC in internal mode and selects the boundary scan register to be connected between TDI and TDO. It allows to drive data into the CORE logic of the IC from the boundary scan inputs and to receive CORE logic output data into the boundary scan outputs.

HIGHZ (11111100) *Bypass Register*	The HIGHZ instruction places the component in a state in which all of its system logic outputs are placed in an inactive drive state (e.g., high impedance). In this state, an in-circuit test system may drive signals onto the connections normally driven by a component output without incurring the risk of damage to the component. The HIGHZ instruction also forces the Bypass Register between TDI and TDO.
CLAMP (11111010) *Bypass Register*	Forces data externally from the boundary scan outputs and selects the one bit bypass register to be connected between TDI and TDO. The data into the boundary scan register can be preloaded using sample/preload.

Table 15. Supported Low Level ISC Commands

ISC_SRAM_WRITE (11100110) *ISC Shift Register*	To configure a specified address of the SRAM. Selects ISC register to be connected between TDI and TDO.
ISC_ENABLE_OTF (11100100) *ISC Shift Register*	Enters ISC mode in on-the-fly mode, which allows the IC to be in normal operating mode while the IC EEPROM is re-programmed.
ISC_ENABLE (11101000) *ISC Shift Register*	Enters normal ISC mode, puts IC in disable mode. Tri-states Outputs
ISC_ENABLE_CLAMP (11101001) *ISC Shift Register*	Requires preloading of Boundary-Scan Register.Enters ISC mode, puts IC in disable mode. Outputs clamp to state determined by data in Boundary-Scan Reg.
ISC_ERASE (11101101) *ISC Shift Register*	Bulk erases the IC EEPROM. Including DONE bits and Security bits
ISC_NOOP (11100000) *Bypass Register*	No Operation in RTI
ISC_PROGRAM (11101010) *ISC Shift Register*	Program the data in the ISC shift register into the addressed EEPROM row.
ISC_READ (11101110) *ISC Shift Register*	Reads the contents of a specified address of the IC EEPROM.
ISC_INIT (11110000) *ISC Shift Register*	Starts initialization process, moves data from the EEPROM to SRAM.
ISC_SRAM_READ (11100111) *ISC Shift Register*	Reads a specific address from the SRAM. If security bit is programmed, SRAM access is blocked.

ISC_DISABLE (11000000) *ISC Shift Register*	Leaves the ISC mode. Device is not Operational untill TLR State is entered.



CR2_10_112603

Figure 14. IEEE STD 1149.1 TAP Controller State Diagram

7.0 TAP Controller Commands

7.1 Test-Logic/Reset: The BST and ISC logic is disabled so that normal operation of the on-chip system logic (i.e. in response to stimuli received through the system pins only) can continue unimpeded. This is achieved by initializing the instruction register to contain the IDCODE instruction. No matter what the original state of the controller, it will enter *Test-Logic-Reset* when TMS is held high for at least 5 rising edges of TCK. The controller remains in this state while TMS is high. Note that the TAP controller will be forced to the *Test-Logic-Reset* controller state at power-up.

7.2 Run-Test/Idle: All of the instructions supported by Philips CPLDs do not cause functions to execute in the *Run-Test/Idle* controller state. Thus, all BST and ISC data registers selected by the current instruction shall retain their previous state (i.e. Idle). The instruction does not change while the TAP controller is in this state.

7.3 Select-DR-Scan: This is a transitional controller state in which all BST and ISC data registers selected by the current instruction retain their previous state. The instruction does not change while the TAP controller is in this state.

7.4 Select-IR-Scan: This is a transitional controller state in which all BST and ISC data registers selected by the current instruction retain their previous state. The instruction does not change while the TAP controller is in this state.

7.5 Capture-DR: In this controller state, data may be parallel loaded into the BST data registers selected by the current instruction on the rising edge of TCK. If a BST data register selected by the current instruction does not have parallel input, or if capturing is not required for the selected test, then the register retains its previous state. The instruction does not change while the TAP controller is in this state.

7.6 Shift-DR: In this controller state, the BST or ISC data register connected between TDI and TDO as a result of the current instruction shifts data one stage towards its serial output on each rising edge of TCK. BST or ISC data registers that are selected by the current instruction, but are not placed in the serial path, retain their previous state. The instruction does not change while the TAP controller is in this state.

7.7 Exit1-DR: This is a transitional state. All BST or ISC data registers selected by the current instruction retain their previous state. The instruction does not change while the TAP controller is in this state.

7.8 Pause-DR: This controller state allows shifting of the BST or ISC data register in the serial path between TDI and TDO to be temporarily halted. All BST or ISC data registers selected by the current instruction retain their previous state. The instruction does not change while the TAP controller is in this state.

7.9 Exit2-DR: This is a transitional state. All BST or ISC data registers selected by the current instruction retain their previous state. The instruction does not change while the TAP controller is in this state.

7.10 Update-DR: Some BST data registers may be provided with a latched parallel output to prevent changes at the parallel output while data is shifted in the associated shift-register path in response to certain instructions (e.g. EXTEST). Data is latched onto the parallel output of these BST data registers from the shift-register path on the falling edge of TCK in the *Update-DR* controller state. The data held at the latched parallel outputs should not change. The instruction does not change while the TAP controller is in this state.

7.11 Capture-IR: In this controller state, the shift-register contained in the instruction register loads a pattern of fixed logic values on the rising edge of TCK. In addition, design-specific data may be loaded into shift-register stages that are not required to be set to fixed values. The CoolRunner II will load STATUS bits into the Instruction register in this state. This STATUS information will be shifted out of TDO and can be viewed to determine the modal state of the device. Figure 13 shows the bits and their location in the Instruction Register. BST or ISC data registers selected by the current instruction retain their previous state. The instruction does not change while the TAP controller is in this state.

7.12 Shift-IR: In this controller state, the shift-register contained in the instruction register is connected between TDI and TDO and shifts data one stage towards its serial output on each rising edge of TCK. BST or ISC data registers that are selected by the current instruction, but are not placed in the serial path, retain their previous state. The instruction does not change while the TAP controller is in this state.

7.13 Exit1-IR: This is a transitional state. BST or ISC data registers selected by the current instruction retain their previous state. The instruction does not change while the TAP controller is in this state and the instruction register retains its state.

7.14 Pause-IR: This controller state allows shifting of the instruction register to be temporarily halted. BST or ISC data registers selected by the current instruction retain their previous state. The instruction does not change while the TAP controller is in this state and the instruction register retains its state.

7.15 Exit2-IR: This is a transitional state. BST or ISC data registers selected by the current instruction retain their previous state. The instruction does not change while the TAP controller is in this state and the instruction register retains its state.

7.16 Update-IR: The instruction shifted into the instruction register is latched onto the parallel output from the shift-register path on the falling edge of TCK in the *Update-IR* controller state. Once the new instruction has been latched, it becomes the current instruction. BST or ISC data registers selected by the current instruction retain their previous state.

8.0 Supported High Level ISC and JTAG Instructions

Support the following High Level ISC and JTAG Boundary-Scan Instructions:

Table 16. Supported High Level ISC Commands

Bulk_Erase	Erase the EEPROM array.
Blank_Check	Verify that the EEPROM array has been erased.
Program	Program the EEPROM array.
Verify	Verify the EEPROM array.
Pr_Security	Program the Security Bits.
Rd_Security	Read the Security Bits.
Pr_DONE	Program the DONE Bit.
Rd_DONE	Read the DONE Bit.
Pr_UES	Program the UES.
USERCODE	Read the UES.
Bulk_Erase_OTF	Erase the EEPROM array.(Device continues to run)
Blank_Check_OTF	Verify that the EEPROM array has been erased. .(Device continues to run)
Program_OTF	Program the EEPROM array. .(Device continues to run)
Verify_OTF	Verify the EEPROM array. .(Device continues to run)
ReConfigure_OTF	Reconfigure the EEPROM array into the SRAM array and begin operation with the new pattern
SRAM_READ	The SRAM READ comand reads the user's data from the SRAM array.
SRAM_WRITE	The SRAM WRITE command writes the user's data into the SRAM array.

Table 17. JTAG Commands

BYPASS	Connect TDI to TDO (1bit)
IDCODE	Read the device's ID code
USERCODE	Read the device's Usercode
Preconditon Outputs	Allows the user to specify outputs

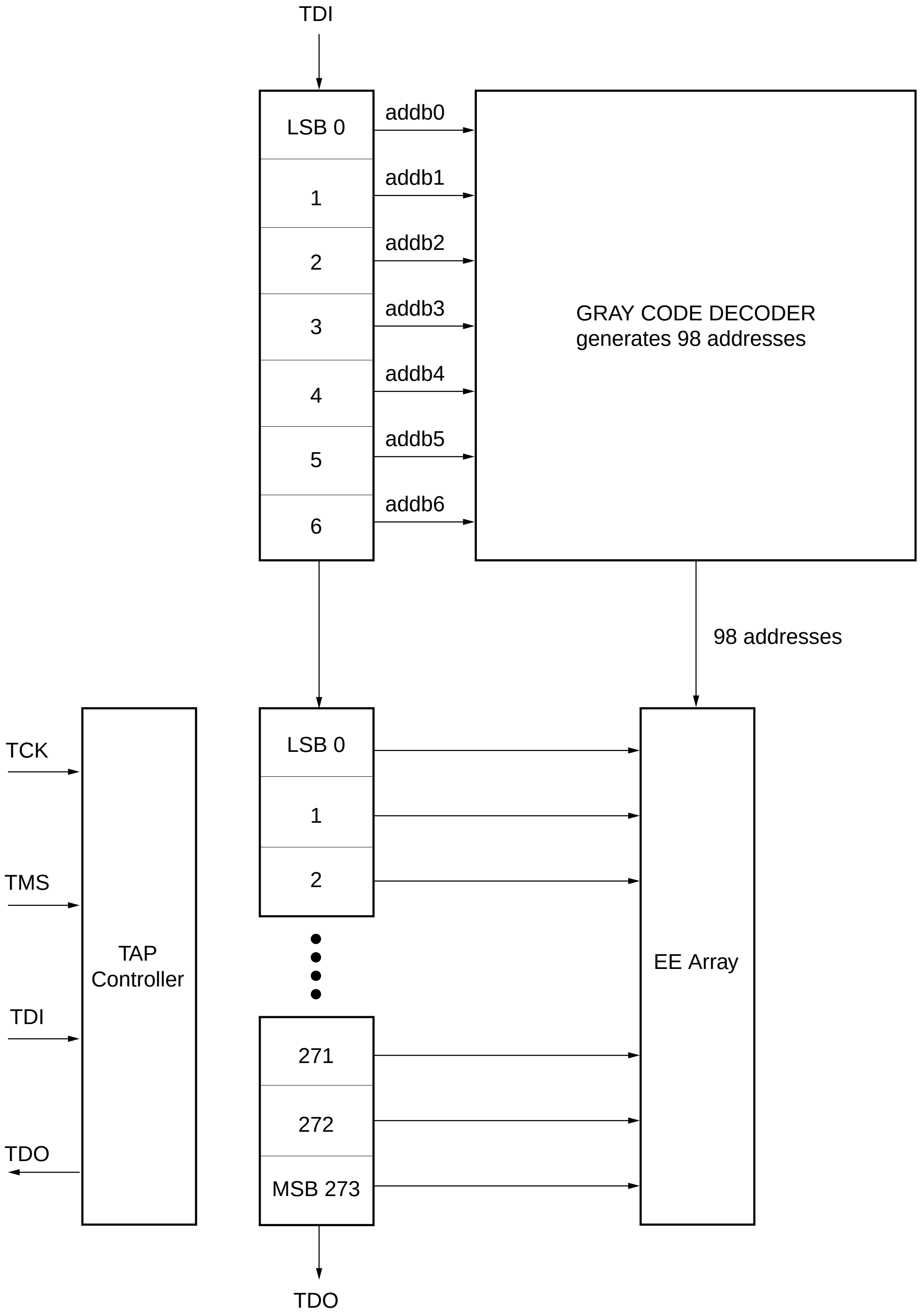


Figure 15. 64 Macrocell Device Shift Register Architecture

9.0 Initialization Times

Maximum initialization time for the entire family of devices is 800 us over operating conditions.

10.0 Detailed Programming Algorithms

The following pages detail high level ISC and JTAG instructions. **Note: the following commands use the XC2C64 as an example. Please refer to Table 2 and Table 10 for device specific data, such as the number of Address bits, number of Shift Register bits, address location of the Done bits, etc.**

Bulk Erase

The Bulk_Erase command erases the entire EEPROM array.

Table 19. Bulk_Erase Command Flow

Command	Sequence
Bulk_Erase	1. Ensure device is in test-logic/reset state 2. Shift in the ENABLE instruction 3. Shift in the ERASE instruction 4. Execute the instruction (erase the contents of the EEPROM array) 5. Shift in the DISCHARGE instruction 6. Execute the instruction (discharge high voltage) 7. Shift in the INIT instruction 8. Execute the instruction (activate the contents of the EEPROM array) 9. Shift in the DISABLE instruction 10. Execute the instruction (activate the contents of the EEPROM array) 11. Shift in the BYPASS instruction

Table 20. Bulk_Erase Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register ; Set enable flip-flop	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)

8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bit (0-6)	TDI = 1011011 (Erase)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Erase MSB)
13	TMS =0 , TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Erase the device	Loop for 100 ms
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 1, TCK = ↑	Select IR-Scan		
16	TMS = 0, TCK = ↑	Capture-IR		
17	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (DISCHARGE)
18	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
19	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Discharge high voltage	Loop for 20 us
20	TMS = 1, TCK = ↑	Select DR-Scan		
21	TMS = 1, TCK = ↑	Select IR-Scan		
22	TMS = 0, TCK = ↑	Capture-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (Init)
23	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Init MSB)
24	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
25	TMS = 1, TCK = ↑	Select DR-Scan		
26	TMS = 0, TCK = ↑	Capture-DR		
27	TMS = 1, TCK = ↑	Exit1-DR		
28	TMS = 1, TCK = ↑	Update-DR		ISP_INIT pulse
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Initialize device with new program data	Loop for Initialization time (800 us)
29	TMS = 1, TCK = ↑	Select DR-Scan		
30	TMS = 1, TCK = ↑	Select IR-Scan		
31	TMS = 0, TCK = ↑	Capture-IR		
32	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
33	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Disable MSB)
34	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
35	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
36	TMS = 1, TCK = ↑	Select DR-Scan		
37	TMS = 1, TCK = ↑	Select IR-Scan		
38	TMS = 0, TCK = ↑	Capture-IR		
39	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
40	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
41	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
42	TMS = 1, TCK = ↑	Select DR-Scan		
43	TMS = 1, TCK = ↑	Select IR-Scan		
44	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
45	TMS = 1,	Test-Logic/Reset	In user mode	DONE

Blank Check

The Blank_Check command verifies that the entire EEPROM array has been erased.

Table 21. Blank_Check Command Flow

Command	Sequence
Blank_Check	1. Ensure device is in Test-Logic/Reset state 2. Shift in BYPASS instruction to flush out Status Register and check for Security and Done status 3. Shift in the ENABLE instruction 4. Shift in the VERIFY instruction 5. Shift in the address of the EEPROM row being verified. 6. Execute the command (this transfers the row data into the ISC Shift Register) 7. Shift out the data from the ISC Shift Register 8. Compare the shifted-out data to the expected data 9. Repeat step 4 though 8 until all EEPROM rows have been checked 10. The DONE bits will be erased and if not re-programmed the device will initialize but the outputs 11. Will remain tri-stated and the ZIA will not be enabled which is ok because Blank is not a valid pattern. 12. Shift in the DISCHARGE instruction 13. Execute the instruction (discharge high voltage) 14. Shift in the INIT instruction 15. Execute the instruction (activate the contents of the EEPROM array) 16. Shift in the DISABLE instruction 17. Execute the instruction (activate the contents of the EEPROM array) 18. Shift in the BYPASS instruction

Table 22. Blank_Check Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 11111111 (Bypass)
Entering the Bypass Instruction will flush out the STATUS Register; and a check for Security and DONE can be performed.				
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register ;	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bit (0-6)	TDI = 0111011 (Verify)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Verify MSB)
13	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 0, TCK = ↑	Capture-DR		
16	TMS = 0, TCK = ↑	Shift-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift first address bits (6-1)	TDI = address (MSB, ...)

17	TMS = 1, TCK = ↑	Exit1-DR	Shift last address bit 0	TDI = address (LSB)
18	TMS = 1, TCK = ↑	Update-DR		
Loop4	TMS = 0, TCK = ↑	Run-Test/Idle	Load the Shift Register.	Loop for 20us
Loop4	TMS = 1, TCK = ↑	Select DR-Scan		
Loop4	TMS = 0, TCK = ↑	Capture-DR		
Loop4	TMS = 0, TCK = ↑	Shift-DR	Bit 0 is available at TDO	TD0 = data bit 0; TDI=1
Loop4	TMS = 0, TCK = ↑	Shift-DR	Shift out data bits (1-273)	TD0 = data bits (1-273);TDI=1
Loop4	TMS = 0, TCK = ↑	Shift-DR	Shift in next address bits (6-1)	
Loop4	TMS = 1, TCK = ↑	Exit1-DR	Shift in LSB address bit 0	Compare Data
Loop4	TMS = 1, TCK = ↑	Update-DR		
19	**Execute loop4 until all addresses have been loaded. Load dummy address after last data read.**			
20	TMS = 1, TCK = ↑	Select DR-Scan		
21	TMS = 1, TCK = ↑	Select IR-Scan		
22	TMS = 0, TCK = ↑	Capture-IR		
23	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (DISCHARGE)
24	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
25	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Discharge high voltage	Loop for 20 us
26	TMS = 1, TCK = ↑	Select DR-Scan		
27	TMS = 1, TCK = ↑	Select IR-Scan		
28	TMS = 0, TCK = ↑	Capture-IR		
29	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (Init)
30	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Init MSB)
31	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
32	TMS = 1, TCK = ↑	Select DR-Scan		
33	TMS = 0, TCK = ↑	Capture-DR		
34	TMS = 1, TCK = ↑	Exit1-DR		
35	TMS = 1, TCK = ↑	Update-DR		ISP_INIT pulse
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Initialize device with new program data	Loop for Initialization time (800 us)
36	TMS = 1, TCK = ↑	Select DR-Scan		
37	TMS = 1, TCK = ↑	Select IR-Scan		
38	TMS = 0, TCK = ↑	Capture-IR		
39	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
40	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
41	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
42	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
43	TMS = 1, TCK = ↑	Select DR-Scan		
44	TMS = 1, TCK = ↑	Select IR-Scan		
45	TMS = 0, TCK = ↑	Capture-IR		
46	TMS = 0, TCK = ↑	Shift-IR		
loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
47	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
48	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
49	TMS = 1, TCK = ↑	Select DR-Scan		
50	TMS = 1, TCK = ↑	Select IR-Scan		
51	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
52	TMS = 1,	Test-Logic/Reset	In user mode	DONE

Program

The Program command programs the user’s data into the EEPROM array.

Table 23. Program Command Sequence

Command	Sequence
Program	1. Ensure device is in Test-Logic/Reset state 2. Shift in the ENABLE instruction 3. Shift in the PROGRAM instruction 4. Shift in the address and data for the EEPROM row being programmed. 5. Execute the command (Program the data into the selected EEPROM row) 6. Repeat steps 4 and 6 until all EEPROM rows have been programmed 7. Shift in the address and data for the EEPROM row 96 1010000 8. Execute the command (Program the DONE bits) 9. Shift in the DISCHARGE instruction 10. Execute the instruction (discharge high voltage) 11. Shift in the INIT instruction 12. Execute the instruction (activate the contents of the EEPROM array) 13. Shift in the DISABLE instruction 14. Execute the instruction (activate the contents of the EEPROM array) 15. Shift in the BYPASS instruction

Table 24. Program Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register; Set Enable Flip-flop	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0101011 (Program)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Program MSB)
13	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 0, TCK = ↑	Capture-DR		
16	TMS = 0, TCK = ↑	Shift-DR		
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift in Data bits 273-0	TDI = data bits (273 - 0)
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift in Address bit 6-1	TDI = address bits (6-1)
Loop3	TMS = 1, TCK = ↑	Exit1-DR	Shift in Address bit 0	TDI = address bit 0 (LSB)
Loop3	TMS = 1, TCK = ↑	Update-DR		
Loop3	TMS = 0, TCK = ↑	Run-Test/Idle	Program data in EEPROM	Wait 10 ms.
	Execute loop3 96 times to PROGRAM the entire device.			
	PROGRAM the DONE Bits (Note; location of DONE bits DONE BIT0=D8=”1”, DONE BIT 1=D9=”0”.			

17	TMS = 1, TCK = ↑	Select DR-Scan		
18	TMS = 0, TCK = ↑	Capture-DR		
19	TMS = 0, TCK = ↑	Shift-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in Data bits 273-0	TDI = data bits (273 - 0)
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in Address bit 6-1(101000)	TDI = address bits (6-1)
20	TMS = 1, TCK = ↑	Exit1-DR	Shift in Address bit 0 (0)	TDI = address bit 0 (LSB)
21	TMS = 1, TCK = ↑	Update-DR		
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Program data in EEPROM	Wait 10 ms.
22	TMS = 1, TCK = ↑	Select DR-Scan		
23	TMS = 1, TCK = ↑	Select IR-Scan		
24	TMS = 0, TCK = ↑	Capture-IR		
25	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (DISCHARGE)
26	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
27	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Discharge high voltage	Loop for 20 us
28	TMS = 1, TCK = ↑	Select DR-Scan		
29	TMS = 1, TCK = ↑	Select IR-Scan		
30	TMS = 0, TCK = ↑	Capture-IR		
31	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (Init)
32	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Init MSB)
33	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
34	TMS = 1, TCK = ↑	Select DR-Scan		
35	TMS = 0, TCK = ↑	Capture-DR		
36	TMS = 1, TCK = ↑	Exit1-DR		
37	TMS = 1, TCK = ↑	Update-DR		
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Initialize device with new program data	Loop for Initialization time (800 us)
38	TMS = 1, TCK = ↑	Select DR-Scan		
39	TMS = 1, TCK = ↑	Select IR-Scan		
40	TMS = 0, TCK = ↑	Capture-IR		
41	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
42	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
43	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
44	TMS = 1, TCK = ↑	Select DR-Scan		
45	TMS = 1, TCK = ↑	Select IR-Scan		
46	TMS = 0, TCK = ↑	Capture-IR		
47	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
48	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
49	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
50	TMS = 1, TCK = ↑	Select DR-Scan		
51	TMS = 1, TCK = ↑	Select IR-Scan		
52	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
53	TMS = 1,	Test-Logic/Reset	In user mode	DONE

Verify

The Verify command verifies that the user's data in the EEPROM array matches the data contained in the JEDEC file.

Table 25. Verify Command Sequence

Command	Sequence
Verify	1. Ensure device is in Test-Logic/Reset state 2. Shift in BYPASS instruction to flush out Status Register and check for Security and Done status 3. Shift in the ENABLE instruction 4. Shift in the VERIFY instruction 5. Shift to RTI and loop 20us for voltages to settle 6. Shift in the address of the EEPROM row being verified. 7. Execute the command (this transfers the row data into the ISC Shift Register) 8. Shift out the data from the ISC Shift Register 9. Compare the shifted-out data to the expected data 10. Repeat step 4 though 8 until all EEPROM rows have been verified 11. Make sure the DONE bits are verifyed (Done0=Bit8=1, Done1=Bit9=0)= programming DONE 12. Shift in the DISCHARGE instruction 13. Execute the instruction (discharge high voltage) 14. Shift in the INIT instruction 15. Execute the instruction (activate the contents of the EEPROM array) 16. Shift in the DISABLE instruction 17. Execute the instruction (activate the contents of the EEPROM array) 18. Shift in the BYPASS instruction

Table 26. Verify Command Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-7)	TDI = 11111111 (Bypass)
Entering the Bypass Instruction will flush out the STATUS Reg. And a check for Security, and DONE can be performed.				
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register ;	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bit (0-6)	TDI = 0111011 (Verify)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Verify MSB)
13	TMS = 0, TCK = ↑	Update-IR	Load the Instruction Register	
15	TMS = 1, TCK = ↑	Select DR-Scan		
16	TMS = 0, TCK = ↑	Capture-DR		
17	TMS = 0, TCK = ↑	Shift-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift first address bits (6 -1)	TDI = address (MSB, ...)

18	TMS = 1, TCK = ↑	Exit1-DR	Shift first address bit 0	TDI = address (LSB)
19	TMS = 1, TCK = ↑	Update-DR	Load the Shift Register.	
Loop3	TMS = 0, TCK = ↑	Run-Test/Idle	Wait for data to settle	Loop for 20us
Loop3	TMS = 1, TCK = ↑	Select DR-Scan		
Loop3	TMS = 0, TCK = ↑	Capture-DR		
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift out data bit 273	TD0 = data bit 273; TDI=1
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift out data bits (272 - 0)	TD0 = data bits (272-0);TDI=1
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift in next address bits (6-1)	
Loop3	TMS = 1, TCK = ↑	Exit1-DR	Shift in next address bit 0	Compare Data
Loop3	TMS = 1, TCK = ↑	Update-DR	Load the Shift Register.	
20	**Execute loop3 until all addresses have been loaded. Load dummy address after last data read.**			
21	TMS = 1, TCK = ↑	Select DR-Scan		
22	TMS = 1, TCK = ↑	Select IR-Scan		
23	TMS = 0, TCK = ↑	Capture-IR		
24	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (DISCHARGE)
25	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
26	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Discharge high voltage	Loop for 20 us
27	TMS = 1, TCK = ↑	Select DR-Scan		
28	TMS = 1, TCK = ↑	Select IR-Scan		
29	TMS = 0, TCK = ↑	Capture-IR		
30	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (Init)
31	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Init MSB)
32	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
33	TMS = 1, TCK = ↑	Select DR-Scan		
34	TMS = 0, TCK = ↑	Capture-DR		
35	TMS = 1, TCK = ↑	Exit1-DR		
36	TMS = 1, TCK = ↑	Update-DR		
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Initialize device with new program data	Loop for Initialization time (800 us)
37	TMS = 1, TCK = ↑	Select DR-Scan		
38	TMS = 1, TCK = ↑	Select IR-Scan		
39	TMS = 0, TCK = ↑	Capture-IR		
40	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
41	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
42	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
43	TMS = 1, TCK = ↑	Select DR-Scan		
44	TMS = 1, TCK = ↑	Select IR-Scan		
45	TMS = 0, TCK = ↑	Capture-IR		
46	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
47	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
48	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
49	TMS = 1, TCK = ↑	Select DR-Scan		
50	TMS = 1, TCK = ↑	Select IR-Scan		
51	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
52	TMS = 1,	Test-Logic/Reset	In user mode	DONE

Program Done

The Program Done command programs the Done cells of the device and is mandatory before Initialization

Table 27. Program Done Command Sequence

Command	Sequence
Pr_Done	1. Ensure device is in the Test-Logic/Reset state 2. Shift in the ENABLE instruction 3. Shift in the PROGRAM instruction 4. Shift in the address of the EEPROM row containing the done bits and shift in the data with the corresponding done bits set to the programming state and the other bits set to the non-programming state. NOTE: The SECURITY bits are also located in this same row. 5. Addr. 1010000,(sec0=D1=”1”, sec1=D=’1’, sec2=D=’1’, sec3=D=’1’, sec4=D=’1’ 6. Sec5=D=’1’, sec6=D=’1’) **DONE0=D=’1’, DONE1=D=’0’ Represent Program Done** 7. Execute the command (Program the data into the selected EEPROM row) 8. Shift in the DISCHARGE instruction 9. Execute the command (Discharge high voltage) 10. Shift in the INIT instruction 11. Execute the command (Activate the done bits) 12. Shift in the DISABLE instruction 13. Execute the instruction 14. Shift in the BYPASS instruction
DIS	15.

Table 28. Program Done Bit Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register; Set Enable Flip-flop	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0101011 (Program)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Program MSB)
13	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 0, TCK = ↑	Capture-DR		
16	TMS = 0, TCK = ↑	Shift-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in Data bits 273-0	TDI = data bits (273 - 0)
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in Address bit 6-1	TDI = address bits (6-1)
17	TMS = 1, TCK = ↑	Exit1-DR	Shift in Address bit 0	TDI = address bit 0 (LSB)
18	TMS = 1, TCK = ↑	Update-DR		

Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Program data in EEPROM	Wait 10 ms.
19	TMS = 1, TCK = ↑	Select DR-Scan		
20	TMS = 1, TCK = ↑	Select IR-Scan		
21	TMS = 0, TCK = ↑	Capture-IR		
22	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (DISCHARGE)
23	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
24	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Discharge high voltage	Loop for 20 us
25	TMS = 1, TCK = ↑	Select DR-Scan		
26	TMS = 1, TCK = ↑	Select IR-Scan		
27	TMS = 0, TCK = ↑	Capture-IR		
28	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (Init)
29	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Init MSB)
30	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
31	TMS = 1, TCK = ↑	Select DR-Scan		
32	TMS = 0, TCK = ↑	Capture-DR		
33	TMS = 1, TCK = ↑	Exit1-DR		
34	TMS = 1, TCK = ↑	Update-DR		
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Initialize device with new program data	Loop for Initialization time (800 us)
35	TMS = 1, TCK = ↑	Select DR-Scan		
36	TMS = 1, TCK = ↑	Select IR-Scan		
37	TMS = 0, TCK = ↑	Capture-IR		
38	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
39	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
40	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
41	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
42	TMS = 1, TCK = ↑	Select DR-Scan		
43	TMS = 1, TCK = ↑	Select IR-Scan		
44	TMS = 0, TCK = ↑	Capture-IR		
45	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
46	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
47	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
48	TMS = 1, TCK = ↑	Select DR-Scan		
49	TMS = 1, TCK = ↑	Select IR-Scan		
50	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
51	TMS = 1,	Test-Logic/Reset	In user mode	DONE

Read Done

The Read Done command reads the Done bits of the device.

Table 29. Read Done Bit Command Sequence

Command	Sequence
Rd_Done	1. Ensure device in is Test-Logic/Reset state 2. Shift in the BYPASS instruction to flush out the Status Register and check for Security and Done status 3. Shift in the ENABLE instruction 4. Shift in the VERIFY instruction 5. Shift in the address of the EEPROM row containing the done bits. 6. Execute the command 7. Addr. 1010000,(sec0=D1=”1”, sec1=D2=’1’, sec2=D3=’1’, sec3=D4=’1’, sec4=D5=’1’ 8. Sec5=D6=’1’, sec6=D7=’1’) **DONE0=D8=’1’, DONE1=D9=’0’ Represent Program Done** 9. Execute the command (Discharge high voltage) 10. Shift in the INIT instruction 11. Execute the command 12. Shift in the DISABLE instruction 13. Execute the instruction 14. Shift in the BYPASS instruction

Table 30. Read Done Bit Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-7)	TDI = 11111111 (Bypass)
Entering the Bypass Instruction will flush out the STATUS Reg. And a check for Security, and DONE can be performed.				
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register ; Set Enable Flip-flop	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0111011 (Verify LSB)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Verify MSB)
13	TMS = 0, TCK = ↑	Update-IR	Load the Instruction Register	
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 0, TCK = ↑	Capture-DR		
16	TMS = 0, TCK = ↑	Shift-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in data bits (273-0)	TDI = All data bits = 0’s
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in address bits (6-1)	TDI = 101000 (Addr.96)
17	TMS = 1, TCK = ↑	Exit1-DR	Shift in address bit 0	TDI = 0 (LSB)
18	TMS = 1, TCK = ↑	Update-DR	.	

Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Load the Shift Register	Loop for 20us
19	TMS = 1, TCK = ↑	Select DR-Scan		
20	TMS = 0, TCK = ↑	Capture-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift out data	1111111 01 rest ‘1’s
21	TMS = 1, TCK = ↑	Exit1-DR		
22	TMS = 1, TCK = ↑	Update-DR		
23	TMS = 1, TCK = ↑	Select DR-Scan		
24	TMS = 1, TCK = ↑	Select IR-Scan		
25	TMS = 0, TCK = ↑	Capture-IR		
26	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (DISCHARGE)
27	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
28	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Discharge high voltage	Loop for 20 us
29	TMS = 1, TCK = ↑	Select DR-Scan		
30	TMS = 1, TCK = ↑	Select IR-Scan		
31	TMS = 0, TCK = ↑	Capture-IR		
32	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (Init)
33	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Init MSB)
34	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
35	TMS = 1, TCK = ↑	Select DR-Scan		
36	TMS = 0, TCK = ↑	Capture-DR		
37	TMS = 1, TCK = ↑	Exit1-DR		
38	TMS = 1, TCK = ↑	Update-DR		ISP_INIT
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Initialize device with new program data	Loop for Initialization time (800 us)
39	TMS = 1, TCK = ↑	Select DR-Scan		
40	TMS = 1, TCK = ↑	Select IR-Scan		
41	TMS = 0, TCK = ↑	Capture-IR		
42	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
43	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
44	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
45	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
46	TMS = 1, TCK = ↑	Select DR-Scan		
47	TMS = 1, TCK = ↑	Select IR-Scan		
48	TMS = 0, TCK = ↑	Capture-IR		
49	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
50	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
51	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
52	TMS = 1, TCK = ↑	Select DR-Scan		
53	TMS = 1, TCK = ↑	Select IR-Scan		
54	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
55	TMS = 1,	Test-Logic/Reset	In user mode	DONE

Program Security

The Program Security command programs the security cell of the device.

Table 31. Program Security Command Sequence

Command	Sequence
Pr_Security	1. Ensure device is in the Test-Logic/Reset state 2. Shift in the ENABLE instruction 3. Shift in the PROGRAM instruction **4.** Shift in the address of the EEPROM row containing the security bit and shift in the data with the corresponding security bits set to the programming state and the other bits set to the non-programming state. NOTE: The DONE bits are also located in this same row. **5.** Addr. 1010000**, (sec0=D1=”0”, sec1=D2=’1’, sec2=D3=’0’, sec3=D4=’1’, sec4=D5=’0’** 6. **Sec5=D6=’1’, sec6=D7=’0’)** DONE0=D8=’1’, DONE1=D9=’0’ Represent Program Done/Secured 7. Execute the command (Program the data into the selected EEPROM row) 8. Shift in the DISCHARGE instruction 9. Execute the command (Discharge high voltage) 10. Shift in the INIT instruction 11. Execute the command (Activate the security bit) 12. Shift in the DISABLE instruction 13. Execute the instruction 14. Shift in the BYPASS instruction

Table 32. Program Security Bit Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register; Set Enable Flip-flop	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0101011 (Program)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Program MSB)
13	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 0, TCK = ↑	Capture-DR		
16	TMS = 0, TCK = ↑	Shift-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in Data bits 273-0	TDI = data bits (273 - 0)
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in Address bit 6-1	TDI = address bits (6-1)
17	TMS = 1, TCK = ↑	Exit1-DR	Shift in Address bit 0	TDI = address bit 0 (LSB)
18	TMS = 1, TCK = ↑	Update-DR		

Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Program data in EEPROM	Wait 10 ms.
19	TMS = 1, TCK = ↑	Select DR-Scan		
20	TMS = 1, TCK = ↑	Select IR-Scan		
21	TMS = 0, TCK = ↑	Capture-IR		
22	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (DISCHARGE)
23	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
24	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Discharge high voltage	Loop for 20 us
25	TMS = 1, TCK = ↑	Select DR-Scan		
26	TMS = 1, TCK = ↑	Select IR-Scan		
27	TMS = 0, TCK = ↑	Capture-IR		
28	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (Init)
29	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Init MSB)
30	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
31	TMS = 1, TCK = ↑	Select DR-Scan		
32	TMS = 0, TCK = ↑	Capture-DR		
33	TMS = 1, TCK = ↑	Exit1-DR		
34	TMS = 1, TCK = ↑	Update-DR		
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Initialize device with new program data	Loop for Initialization time (800 us)
35	TMS = 1, TCK = ↑	Select DR-Scan		
36	TMS = 1, TCK = ↑	Select IR-Scan		
37	TMS = 0, TCK = ↑	Capture-IR		
38	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
39	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
40	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
41	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
42	TMS = 1, TCK = ↑	Select DR-Scan		
43	TMS = 1, TCK = ↑	Select IR-Scan		
44	TMS = 0, TCK = ↑	Capture-IR		
45	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
46	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
47	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
48	TMS = 1, TCK = ↑	Select DR-Scan		
49	TMS = 1, TCK = ↑	Select IR-Scan		
50	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
51	TMS = 1,	Test-Logic/Reset	In user mode	DONE

Read Security

The Read Security command reads the security bit of the device.

Table 33. Read Security Bit Command Sequence

Command	Sequence
Rd_Security	1. Ensure device in is Test-Logic/Reset state 2. Shift in the BYPASS instruction to flush out the Status Register and check for Security and Done status 3. Shift in the ENABLE instruction 4. Sshift in the VERIFY instruction 5. Shift in the EEPROM row address containing the security bits 6. Execute the command **7.** Addr. 1010000**, (sec0=D1="0", sec1=D2='1', sec2=D3='0', sec3=D4='1', sec4=D5='0'** 8. **Sec5=D6='1', sec6=D7='0')** DONE0=D8='1', DONE1=D9='0' Represent Program Done/Secured 9. Shift in the DISCHARGE instruction 10. Execute the command 11. Shift in the INIT instruction 12. Execute the command 13. Shift in the DISABLE instruction 14. Execute the instruction 15. Shift in the BYPASS instruction

Table 34. Read Security Bit Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-7)	TDI = 11111111 (Bypass)
Entering the Bypass Instruction will flush out the STATUS Reg. And a check for Security, and DONE can be performed.				
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register ; Set Enable Flip-flop	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0111011 (Verify LSB)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Verify MSB)
13	TMS = 0, TCK = ↑	Update-IR	Load the Instruction Register	
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 0, TCK = ↑	Capture-DR		
16	TMS = 0, TCK = ↑	Shift-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in data bits (273-0)	TDI = All data bits = 0's
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in address bits (6-1)	TDI = 101000 (Addr.96)

17	TMS = 1, TCK = ↑	Exit1-DR	Shift in address bit 0	TDI = 0 (LSB)
18	TMS = 1, TCK = ↑	Update-DR	.	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Load the Shift Register	Loop for 20us
19	TMS = 1, TCK = ↑	Select DR-Scan		
20	TMS = 0, TCK = ↑	Capture-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift out data bit 274	0101010 10 rest ‘1’s
21	TMS = 1, TCK = ↑	Exit1-DR		
22	TMS = 1, TCK = ↑	Update-DR		
23	TMS = 1, TCK = ↑	Select DR-Scan		
24	TMS = 1, TCK = ↑	Select IR-Scan		
25	TMS = 0, TCK = ↑	Capture-IR		
26	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (DISCHARGE)
27	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
28	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Discharge high voltage	Loop for 20 us
29	TMS = 1, TCK = ↑	Select DR-Scan		
30	TMS = 1, TCK = ↑	Select IR-Scan		
31	TMS = 0, TCK = ↑	Capture-IR		
32	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6	TDI = 0000111 (INIT)
33	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1NIT MSB)
34	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
35	TMS = 1, TCK = ↑	Select DR-Scan		
36	TMS = 0, TCK = ↑	Capture-DR		
37	TMS = 1, TCK = ↑	Exit1-DR		
38	TMS = 1, TCK = ↑	Update-DR		
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Initialize device with new program data	Loop for Initialization time (800 us)
39	TMS = 1, TCK = ↑	Select DR-Scan		
40	TMS = 1, TCK = ↑	Select IR-Scan		
41	TMS = 0, TCK = ↑	Capture-IR		
42	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
43	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
44	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
45	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
46	TMS = 1, TCK = ↑	Select DR-Scan		
47	TMS = 1, TCK = ↑	Select IR-Scan		
48	TMS = 0, TCK = ↑	Capture-IR		
49	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
50	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Bypass MSB)
51	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
52	TMS = 1, TCK = ↑	Select DR-Scan		
53	TMS = 1, TCK = ↑	Select IR-Scan		
54	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
55	TMS = 1,	Test-Logic/Reset	In user mode	DONE

11.0 JTAG Instructions

The JTAG commands supported by the ISC PC Parallel Port Programmer are described in the following subsections. These commands are basic JTAG commands which are required by the ISC programmer to control multiple JTAG/ISC devices in a JTAG chain.

Bypass

The Bypass command passes data from TDI to TDO with a one half clock cycle delay. The outputs of the device are not affected by this operation.

Table 35. BYPASS Command Sequence

Command	Sequence
BYPASS	1. Shift in the BYPASS instruction 2. Shift in/out data while in the DR-Shift TAP Controller State

Table 36. BYPASS Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
loop1	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (BYPASS MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 0, TCK = ↑	Capture-DR		
10	TMS = 0, TCK = ↑	Shift-DR		
loop2	TMS = 0, TCK = ↑	Shift-DR	Shift in/out Data bits	TDI=TDO (1/2 TCK delay)
11	TMS = 1, TCK = ↑	Exit1-DR	Shift in/out last Data bit	TDI=TDO (1/2 TCK delay)
12	TMS = 1, TCK = ↑	Update-DR		
13	TMS = 1, TCK = ↑	Select DR-Scan		
14	TMS = 1, TCK = ↑	Select IR-Scan		
15	TMS = 1, TCK = ↑	Test-Logic/Reset		
	TMS = 1	Test-Logic/Reset	DONE	DONE

Please note that the device must remain in loop2, "*Shift-DR State*", to remain in the BYPASS mode.

IDCODE

The IDCODE command returns device identification data on TDO. The outputs of the device are not affected by this operation.

Table 37. IDCODE Command Sequence

Command	Sequence
IDCODE	1. Shift in the IDCODE instruction 2. Shift in/out data while in the DR-Shift TAP Controller State

Table 38. IDCODE Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
loop1	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1000000 (IDCODE)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 0 (IDCODE MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 0, TCK = ↑	Capture-DR		
10	TMS = 0, TCK = ↑	Shift-DR		
loop2	TMS = 0, TCK = ↑	Shift-DR	Shift in/out Data bits (0-31)	Device ID data
11	TMS = 1, TCK = ↑	Exit1-DR	Shift in/out last Data bit (32)	Device ID data (MSB)
12	TMS = 1, TCK = ↑	Update-DR		
13	TMS = 1, TCK = ↑	Select DR-Scan		
14	TMS = 1, TCK = ↑	Select IR-Scan		
15	TMS = 1, TCK = ↑	Test-Logic/Reset		
	TMS = 1	Test-Logic/Reset	DONE	DONE

USERCODE

The USERCODE command loads and shifts out the 32 bit user provided signature data. The outputs of the device are not affected by this operation.

Table 39. USERCODE Command Sequence

Command	Sequence
USERCODE	1. Shift in the USERCODE instruction 2. Shift in/out data while in the DR-Shift TAP Controller State

Table 40. USERCODE Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
loop1	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111101 (USERCODE)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (USERCODE MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 0, TCK = ↑	Capture-DR		
10	TMS = 0, TCK = ↑	Shift-DR		
loop2	TMS = 0, TCK = ↑	Shift-DR	Shift in/out USERCODE data	USERCODE value (0-31)
11	TMS = 1, TCK = ↑	Exit1-DR	Shift in/out last USERCODE data bit	(USERCODE MSB)
12	TMS = 1, TCK = ↑	Update-DR		
13	TMS = 1, TCK = ↑	Select DR-Scan		
14	TMS = 1, TCK = ↑	Select IR-Scan		
15	TMS = 1, TCK = ↑	Test-Logic/Reset		
	TMS = 1	Test-Logic/Reset	DONE	DONE

12.0 Advanced Functionality

The CoolRunner-II family of devices supports two advanced configuration capabilities. The first, known as On-The-Fly programming allows end users to configure the NV-RAM while the device is active without affecting the device operation. The stored configuration is then made active under end-user control.

The second, known as SRAM programming, allows access directly to the SRAM pattern memory for both programming and verification.

12.1 Programming On-The-Fly

Programming On-The-Fly is a feature on the CoolRunner II that allows a new pattern to be programmed into the EEPROM without interrupting the operation of the device on the present pattern. Another way to look at this is reprogramming in the background. To begin operation on the new pattern just execute the ISC_INIT instruction and in microseconds the part will re-configure to the new pattern and begin operation.

ISC_PROGRAM and ISC_READ instructions are the same instructions as those used for regular programming, the only difference is that the Initialization instruction is not included in the High Level Commands for these operations.

The following High level commands are defined for use when the ISC_ENABLE_OTF instruction is executed in place of the ISC_ENABLE instruction.

On-The-Fly Bulk Erase

The Bulk Erase command erases the entire EEPROM array.

Table 41. Bulk_Erase Command Flow

Command	Sequence
Bulk_Erase	1. Ensure device is in test-logic/reset state 2. Shift in the ISC_ENABLE_OTF instruction **(Device will continue to operate)** 3. Shift in the ERASE instruction 4. Execute the instruction (erase the contents of the EEPROM array) 5. Shift in the ISC_DISABLE instruction 6. Execute the instruction.

Table 42. Bulk_Erase Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0010011 (Enable_OTF)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register ; Set enable flip-flop	
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bit (0-6)	TDI = 1011011 (Erase)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Erase MSB)
13	TMS =0 , TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Erase the device	Loop for 100 ms
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 1, TCK = ↑	Select IR-Scan		
16	TMS = 0, TCK = ↑	Capture-IR		
17	TMS = 0, TCK = ↑	Shift-IR		
Loop5	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
18	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
19	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
20	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
21	TMS = 1, TCK = ↑	Select DR-Scan		
22	TMS = 1, TCK = ↑	Select IR-Scan		
23	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
24	TMS = 1,	Test-Logic/Reset	In user mode	DONE

The Initialization has been removed to avoid disturbing the device from operating on the present pattern which is stored in SRAM.

On-The-Fly Programming

The Program command programs the user’s data into the EEPROM array.

Table 43. Program Command Sequence

Command	Sequence
Program	1. Ensure device is in Test-Logic/Reset state 2. Shift in the ISC_ENABLE_OTF instruction 3. Shift in the ISC_PROGRAM instruction 4. Shift in the address and data for the EEPROM row being programmed. 5. Execute the command (Program the data into the selected EEPROM row) 6. Repeat steps 4 and 6 until all EEPROM rows have been programmed 7. Shift in the address and data for the EEPROM row 96 1010000 8. Execute the command (Program the DONE bits) 9. Shift in the DISABLE instruction 10. Execute the instruction (activate the contents of the EEPROM array)

Table 44. Program Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0010011 (Enable_OTF)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register ; Set enable flip-flop	
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0101011 (Program)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Program MSB)
13	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop3	TMS = 1, TCK = ↑	Select DR-Scan		
Loop3	TMS = 0, TCK = ↑	Capture-DR		
Loop3	TMS = 0, TCK = ↑	Shift-DR		
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift in Data bits 273-0	TDI = data bits (273 - 0)
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift in Address bit 6-1	TDI = address bits (6-1)
Loop3	TMS = 1, TCK = ↑	Exit1-DR	Shift in Address bit 0	TDI = address bit 0 (LSB)
Loop3	TMS = 1, TCK = ↑	Update-DR		
Loop3	TMS = 0, TCK = ↑	Run-Test/Idle	Program data in EEPROM	Wait 10 ms.
	Execute loop3 96 times to PROGRAM the entire device.			
	PROGRAM the DONE Bits			
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 0, TCK = ↑	Capture-DR		
16	TMS = 0, TCK = ↑	Shift-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in Data bits 273-0	TDI = data bits (273 - 0)
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift in Address bit 6-1(101000)	TDI = address bits (6-1)
17	TMS = 1, TCK = ↑	Exit1-DR	Shift in Address bit 0 (0)	TDI = address bit 0 (LSB)
18	TMS = 1, TCK = ↑	Update-DR		

Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Program data in EEPROM	Wait 10 ms.
19	TMS = 1, TCK = ↑	Select DR-Scan		
20	TMS = 1, TCK = ↑	Select IR-Scan		
21	TMS = 0, TCK = ↑	Capture-IR		
22	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
23	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
24	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
25	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
26	TMS = 1, TCK = ↑	Select DR-Scan		
27	TMS = 1, TCK = ↑	Select IR-Scan		
28	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
29	TMS = 1,	Test-Logic/Reset	In user mode	DONE

The Initialization has been removed to avoid disturbing the device from operating on the present pattern.

On-The-Fly Verification

The Verify command verifies that the user's data in the EEPROM array matches the data contained in the JEDEC file.

Table 45. Command Sequence

Command	Sequence
Verify	1. Ensure device is in Test-Logic/Reset state 2. Shift in the BYPASS instruction to flush out the Status Register and check for Security and Done status 3. Shift in the ISC_ENABLE_OTF instruction 4. Shift in the ISC_READ instruction 5. Shift in the address for the EEPROM row being verified.. 6. Execute the command (verify the data in the selected EEPROM row) 7. Repeat steps 4 and 6 until all EEPROM rows have been verified 8. Shift in the address and data for the EEPROM row 96 9. Execute the command (Program the DONE bits) 10. Shift in the DISABLE instruction 11. Execute the instruction (activate the contents of the EEPROM array)

Table 46. Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-7)	TDI = 11111111 (Bypass)
Entering the Bypass Instruction will flush out the STATUS Reg. And a check for Security, and DONE can be performed.				
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0010011 (Enable_OTF)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register ; Set enable flip-flop	
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bit (0-6)	TDI = 0111011 (Verify)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Verify MSB)
13	TMS = 0, TCK = ↑	Update-IR	Load the Instruction Register	
15	TMS = 1, TCK = ↑	Select DR-Scan		
16	TMS = 0, TCK = ↑	Capture-DR		
17	TMS = 0, TCK = ↑	Shift-DR		
Loop	TMS = 0, TCK = ↑	Shift-DR	Shift first address bits (6 -1)	TDI = address (MSB, ...)
18	TMS = 1, TCK = ↑	Exit1-DR	Shift first address bit 0	TDI = address (LSB)
19	TMS = 1, TCK = ↑	Update-DR	Load the Shift Register.	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Wait for data to settle	Loop for 20us
Loop3	TMS = 1, TCK = ↑	Select DR-Scan		
Loop3	TMS = 0, TCK = ↑	Capture-DR		
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift out data bit 273	TD0 = data bit 273; TDI=1
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift out data bits (272 - 0)	TD0 = data bits (272-0);TDI=1
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift in next address bits (6-1)	
Loop3	TMS = 1, TCK = ↑	Exit1-DR	Shift in next address bit 0	Compare Data

Loop3	TMS = 1, TCK = ↑	Update-DR		
Loop3	TMS = 0, TCK = ↑	Run-Test/Idle	Wait for data to settle	Loop for 20us
	Execute loop3 until all addresses have been loaded. Load dummy address after last data read.			
20	TMS = 1, TCK = ↑	Select DR-Scan		
21	TMS = 1, TCK = ↑	Select IR-Scan		
22	TMS = 0, TCK = ↑	Capture-IR		
23	TMS = 0, TCK = ↑	Shift-IR		
Loop5	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
24	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
25	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
26	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
27	TMS = 1, TCK = ↑	Select DR-Scan		
28	TMS = 1, TCK = ↑	Select IR-Scan		
29	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
30	TMS = 1,	Test-Logic/Reset	In user mode	DONE

NOTE: The device will still be runnung the old pattern.

On-The-Fly Re-Configuration

The following command sequence can be used to Re-Configure the device to the new pattern and begin operation.

Table 47. Re-Configuration Command Sequence

Command	Sequence
Verify	1. Ensure device is in Test-Logic/Reset state 2. Shift in the ISC_ENABLE_OTF instruction 3. Shift in the INIT instruction 4. Execute the instruction (Device will tri-state during initialization, then begin running the new pattern) 5. Shift in the DISABLE instruction 6. Execute the instruction (activate the contents of the EEPROM array)

Table 48. Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0010011 (Enable_OTF)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register ; Set enable flip-flop	
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (DISCHARGE)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
13	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Discharge high voltage	Loop for 20 us
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 1, TCK = ↑	Select IR-Scan		
16	TMS = 0, TCK = ↑	Capture-IR		
17	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000111 (Init)
18	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Init MSB)
19	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
20	TMS = 1, TCK = ↑	Select DR-Scan		
21	TMS = 0, TCK = ↑	Capture-DR		
22	TMS = 1, TCK = ↑	Exit1-DR		
23	TMS = 1, TCK = ↑	Update-DR		
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Initialize device with new program data	Loop for Initialization time (800 us)
24	TMS = 1, TCK = ↑	Select DR-Scan		
25	TMS = 1, TCK = ↑	Select IR-Scan		
26	TMS = 0, TCK = ↑	Capture-IR		
27	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)

28	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
29	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
30	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
31	TMS = 1, TCK = ↑	Select DR-Scan		
32	TMS = 1, TCK = ↑	Select IR-Scan		
33	TMS = 0, TCK = ↑	Capture-IR		
34	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
35	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
36	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
37	TMS = 1, TCK = ↑	Select DR-Scan		
38	TMS = 1, TCK = ↑	Select IR-Scan		
39	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
40	TMS = 1,	Test-Logic/Reset	In user mode	DONE

12.2 SRAM READ and WRITE Commands

SRAM READ for XC2C64/A Only

The SRAM READ command reads the user’s data from the SRAM array.
This SRAM READ command uses the Exit2-DR and Pause-DR states, this is not IEEE 1532 compliant.
Use the next SRAM READ command for all other family members. For verification of SRAM programming, the data returned during SRAM read back is compared to the data in the JEDEC file but all vacant SRAM locations are read back as ‘1’s.

Table 49. READ Command Sequence

Command	Sequence
ISC_SRAM_READ	1. Ensure device is in Test-Logic/Reset state 2. Shift in the ISC_ENABLE instruction 3. Shift in the ISC_SRAM_READ instruction 4. Shift in the address for the SRAM row being read. 5. Execute the command (Read the data from the selected SRAM row) 6. Repeat steps 4 and 6 until all SRAM rows have been read 7. Shift in the DISABLE instruction 8. Execute the instruction (activate the contents of the EEPROM array) 9. Shift in the BYPASS instruction

Table 50. SRAM READ Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register; Set Enable Flip-flop	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Enable Instruction	Loop for Initialization Times (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1110011 (ISC_SRAM_READ)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
13	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop3	TMS = 1, TCK = ↑	Select DR-Scan		
Loop3	TMS = 0, TCK = ↑	Capture-DR		
Loop3	TMS = 0, TCK = ↑	Shift-DR	First data bit is valid on TDO	Ignore this bit first pass
Loop4	TMS = 0, TCK = ↑	Shift-DR	Shift out remaining bits	Ignore these bits on first pass
Loop4	TMS = 0, TCK = ↑	Shift-DR	Shift in Address bit 6-1	TDI = address bits (6-1)
Loop3	TMS = 1, TCK = ↑	Exit1-DR	Shift in Address bit 0	TDI = address bit 0 (LSB)

Loop3	TMS = 0, TCK = ↑	Pause-DR		Equalize BitLines
Loop3	TMS = 1, TCK = ↑	Exit2-DR		SRAM to BitLines
Loop3	TMS = 1, TCK = ↑	Update-DR		Load Shift Reg.
Loop3	TMS = 0, TCK = ↑	Run-Test/Idle		
	Execute loop3 96 times to read the entire device.			
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 1, TCK = ↑	Select IR-Scan		
16	TMS = 0, TCK = ↑	Capture-IR		
17	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
18	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
19	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
20	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
21	TMS = 1, TCK = ↑	Select DR-Scan		
22	TMS = 1, TCK = ↑	Select IR-Scan		
23	TMS = 0, TCK = ↑	Capture-IR		
24	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
25	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
26	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
27	TMS = 1, TCK = ↑	Select DR-Scan		
28	TMS = 1, TCK = ↑	Select IR-Scan		
29	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
30	TMS = 1,	Test-Logic/Reset	In user mode	DONE

SRAM READ

The SRAM READ command reads the user's data from the SRAM array. For verification of SRAM programming, the data returned during SRAM read back is compared to the data in the JEDEC file but all vacant SRAM locations are read back as '1's.

Table 51. READ Command Sequence

Command	Sequence
ISC_SRAM_READ	1. Ensure device is in Test-Logic/Reset state 2. Shift in the ISC_ENABLE instruction 3. Shift in the ISC_SRAM_READ instruction 4. Shift in the address for the SRAM row being read. 5. Execute the command (Read the data from the selected SRAM row) 6. Repeat steps 4 and 6 until all SRAM rows have been read 7. Shift in the DISABLE instruction 8. Execute the instruction (activate the contents of the EEPROM array) 9. Shift in the BYPASS instruction

Table 52. SRAM READ Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register; Set Enable Flip-flop	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1110011 (ISC_SRAM_READ)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
13	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop3	TMS = 1, TCK = ↑	Select DR-Scan		
Loop3	TMS = 0, TCK = ↑	Capture-DR		
Loop3	TMS = 0, TCK = ↑	Shift-DR	First data bit is valid on TDO	Ignore this bit first pass
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift out remaining bits	Ignore these bits on first pass
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift in Address bit 6-1	TDI = address bits (6-1)
Loop3	TMS = 1, TCK = ↑	Exit1-DR	Shift in Address bit 0	Equalize BitLines
Loop3	TMS = 1, TCK = ↑	Update-DR		SRAM to BitLines
Loop3	TMS = 0, TCK = ↑	Run-Test/Idle		Load Shift Reg
	Execute loop3 96 times to read the entire device.			
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 1, TCK = ↑	Select IR-Scan		
16	TMS = 0, TCK = ↑	Capture-IR		
17	TMS = 0, TCK = ↑	Shift-IR		

Loop5	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
18	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)
19	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
20	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
21	TMS = 1, TCK = ↑	Select DR-Scan		
22	TMS = 1, TCK = ↑	Select IR-Scan		
23	TMS = 0, TCK = ↑	Capture-IR		
24	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
25	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
26	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
27	TMS = 1, TCK = ↑	Select DR-Scan		
28	TMS = 1, TCK = ↑	Select IR-Scan		
29	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
30	TMS = 1,	Test-Logic/Reset	In user mode	DONE

SRAM WRITE

The SRAM WRITE command writes the user's data into the SRAM array.

Table 53. WRITE Command Sequence

Command	Sequence
ISC_SRAM_WRITE	1. Ensure device is in Test-Logic/Reset state 2. Shift in the ENABLE instruction 3. Shift in the ISC_SRAM_WRITE instruction 4. Shift in the address for the SRAM row being written. 5. Execute the command (Write the data into the selected SRAM row) 6. Repeat steps 4 and 6 until all SRAM rows have been read 7. Shift in the DISABLE instruction 8. Execute the instruction (activate the contents of the EEPROM array) 9. Shift in the BYPASS instruction

Table 54. SRAM WRITE Detailed Algorithm

Step	Transition Conditions	TAP State	CPLD Event Description	Programmer Action
0	TMS = 1	Test-Logic/Reset	BEGIN	BEGIN
Loop0	TMS = 1, TCK = ↑	Test-Logic/Reset	Ensure device in Test-Logic/Reset State	Loop 5 times
1	TMS = 0, TCK = ↑	Run-Test/Idle		
2	TMS = 1, TCK = ↑	Select DR-Scan		
3	TMS = 1, TCK = ↑	Select IR-Scan		
4	TMS = 0, TCK = ↑	Capture-IR		
5	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0001011 (Enable)
6	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Enable MSB)
7	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register; Set Enable Flip-flop	
Loop	TMS = 0, TCK = ↑	Run-Test/Idle	Execute: Enable instruction	Loop for Initialization time (800 us)
8	TMS = 1, TCK = ↑	Select DR-Scan		
9	TMS = 1, TCK = ↑	Select IR-Scan		
10	TMS = 0, TCK = ↑	Capture-IR		
11	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0110011 (ISC_SRAM_WRITE)
12	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (MSB)
13	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
Loop3	TMS = 1, TCK = ↑	Select DR-Scan		
Loop3	TMS = 0, TCK = ↑	Capture-DR		
Loop3	TMS = 0, TCK = ↑	Shift-DR		
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift in Data bits to write to SRAM	TDI = data
Loop3	TMS = 0, TCK = ↑	Shift-DR	Shift in Address bit 6-1	TDI = address bits (6-1)
Loop3	TMS = 1, TCK = ↑	Exit1-DR	Shift in Address bit 0	TDI = address bit 0 (LSB)
Loop3	TMS = 1, TCK = ↑	Update-DR		
Loop3	TMS = 0, TCK = ↑	Run-Test/Idle		
	Execute loop3 96 times to Write the entire device.			
14	TMS = 1, TCK = ↑	Select DR-Scan		
15	TMS = 1, TCK = ↑	Select IR-Scan		
16	TMS = 0, TCK = ↑	Capture-IR		
17	TMS = 0, TCK = ↑	Shift-IR		
Loop5	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 0000001 (Disable)
18	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1 (Disable MSB)

19	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
20	TMS = 0, TCK = ↑	Run-Test/Idle	Execute Disable	
21	TMS = 1, TCK = ↑	Select DR-Scan		
22	TMS = 1, TCK = ↑	Select IR-Scan		
23	TMS = 0, TCK = ↑	Capture-IR		
24	TMS = 0, TCK = ↑	Shift-IR		
Loop	TMS = 0, TCK = ↑	Shift-IR	Shift in instruction bits (0-6)	TDI = 1111111 (BYPASS)
25	TMS = 1, TCK = ↑	Exit1-IR	Shift in instruction bit 7	TDI = 1(Bypass MSB)
26	TMS = 1, TCK = ↑	Update-IR	Load the Instruction Register	
27	TMS = 1, TCK = ↑	Select DR-Scan		
28	TMS = 1, TCK = ↑	Select IR-Scan		
29	TMS = 1, TCK = ↑	Test-Logic/Reset	One negative edge clock	
30	TMS = 1,	Test-Logic/Reset	In user mode	DONE