3. Configuration and Testing

AGX51003-2.0

Introduction

All Arria® GX devices provide JTAG boundary-scan test (BST) circuitry that complies with the IEEE Std. 1149.1. You can perform JTAG boundary-scan testing either before or after, but not during configuration. Arria GX devices can also use the JTAG port for configuration with the Quartus® II software or hardware using either jam files (**.jam**) or jam byte-code files (**.jbc**).

This chapter contains the following sections:

IEEE Std. 1149.1 JTAG Boundary-Scan Support

Arria GX devices support I/O element (IOE) standard setting reconfiguration through the JTAG BST chain. The JTAG chain can update the I/O standard for all input and output pins any time before or during user-mode through the `CONFIG_IO` instruction. You can use this capability for JTAG testing before configuration when some of the Arria GX pins drive or receive from other devices on the board using voltage-referenced standards. Because the Arria GX device may not be configured before JTAG testing, the I/O pins may not be configured for appropriate electrical standards for chip-to-chip communication. Programming these I/O standards via JTAG allows you to fully test the I/O connections to other devices.

A device operating in JTAG mode uses four required pins, `TDI`, `TDO`, `TMS`, and `TCK`, and one optional pin, `TRST`. The `TCK` pin has an internal weak pull-down resistor, while the `TDI`, `TMS`, and `TRST` pins have weak internal pull-up resistors. The JTAG input pins are powered by the 3.3-V V_{CCPD} pins. The `TDO` output pin is powered by the V_{CCIO} power supply in I/O bank 4.

Arria GX devices also use the JTAG port to monitor the logic operation of the device with the SignalTap® II embedded logic analyzer. Arria GX devices support the JTAG instructions shown in Table 3–1.

Arria GX, Cyclone® II, Cyclone, Stratix®, Stratix II, Stratix GX , and Stratix II GX devices must be within the first 17 devices in a JTAG chain. All of these devices have the same JTAG controller. If any of the Stratix, Arria GX, Cyclone, and Cyclone II devices are in the 18th or further position, they will fail configuration. This does not affect the functionality of the SignalTap® II embedded logic analyzer.

Table 3–1. Arria GX JTAG Instructions

JTAG Instruction	Instruction Code	Description
SAMPLE/PRELOAD	00 0000 0101	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation and permits an initial data pattern to be output at the device pins. Also used by the SignalTap II embedded logic analyzer.
EXTEST *(1)*	00 0000 1111	Allows external circuitry and board-level interconnects to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
BYPASS	11 1111 1111	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation.
USERCODE	00 0000 0111	Selects the 32-bit USERCODE register and places it between the TDI and TDO pins, allowing the USERCODE to be serially shifted out of TDO.
IDCODE	00 0000 0110	Selects the IDCODE register and places it between TDI and TDO, allowing IDCODE to be serially shifted out of TDO.
HIGHZ *(1)*	00 0000 1011	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation, while tri-stating all of the I/O pins.
CLAMP *(1)*	00 0000 1010	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while holding I/O pins to a state defined by the data in the boundary-scan register.
ICR instructions	—	Used when configuring an Arria GX device via the JTAG port with a USB-Blaster™, MasterBlaster™, ByteBlasterMV™, EthernetBlaster™, or ByteBlaster II download cable, or when using a **.jam** or **.jbc** via an embedded processor or JRunner™.
PULSE_NCONFIG	00 0000 0001	Emulates pulsing the nCONFIG pin low to trigger reconfiguration even though the physical pin is unaffected.
CONFIG_IO *(2)*	00 0000 1101	Allows configuration of I/O standards through the JTAG chain for JTAG testing. Can be executed before, during, or after configuration. Stops configuration if executed during configuration. Once issued, the CONFIG_IO instruction holds nSTATUS low to reset the configuration device. nSTATUS is held low until the IOE configuration register is loaded and the TAP controller state machine transitions to the UPDATE_DR state.

Notes to Table 3–1:

(1) Bus hold and weak pull-up resistor features override the high-impedance state of HIGHZ, CLAMP, and EXTEST.

(2) For more information about using the CONFIG_IO instruction, refer to the *MorphIO: An I/O Reconfiguration Solution for Altera Devices* White Paper.

The Arria GX device instruction register length is 10 bits and the USERCODE register length is 32 bits. Table 3–2 and Table 3–3 show the boundary-scan register length and device IDCODE information for Arria GX devices.

Table 3–2. Arria GX Boundary-Scan Register Length

Device	Boundary-Scan Register Length
EP1AGX20	1320
EP1AGX35	1320
EP1AGX50	1668
EP1AGX60	1668
EP1AGX90	2016

Table 3–3. 2-Bit Arria GX Device IDCODE

Device	IDCODE (32 Bits)			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit)
EP1AGX20	0000	0010 0001 0010 0001	000 0110 1110	1
EP1AGX35	0000	0010 0001 0010 0001	000 0110 1110	1
EP1AGX50	0000	0010 0001 0010 0010	000 0110 1110	1
EP1AGX60	0000	0010 0001 0010 0010	000 0110 1110	1
EP1AGX90	0000	0010 0001 0010 0011	000 0110 1110	1

SignalTap II Embedded Logic Analyzer

Arria GX devices feature the SignalTap II embedded logic analyzer, which monitors design operation over a period of time through the IEEE Std. 1149.1 (JTAG) circuitry. You can analyze internal logic at speed without bringing internal signals to the I/O pins. This feature is particularly important for advanced packages, such as FineLine BGA (FBGA) packages, because it can be difficult to add a connection to a pin during the debugging process after a board is designed and manufactured.

Configuration

The logic, circuitry, and interconnects in the Arria GX architecture are configured with CMOS SRAM elements. Altera® FPGAs are reconfigurable and every device is tested with a high coverage production test program so you do not have to perform fault testing and can instead focus on simulation and design verification.

Arria GX devices are configured at system power up with data stored in an Altera configuration device or provided by an external controller (for example, a MAX® II device or microprocessor). You can configure Arria GX devices using the fast passive parallel (FPP), active serial (AS), passive serial (PS), passive parallel asynchronous (PPA), and JTAG configuration schemes. Each Arria GX device has an optimized interface that allows microprocessors to configure it serially or in parallel, and synchronously or asynchronously. The interface also enables microprocessors to treat Arria GX devices as memory and configure them by writing to a virtual memory location, making reconfiguration easy.

In addition to the number of configuration methods supported, Arria GX devices also offer decompression and remote system upgrade features. The decompression feature allows Arria GX FPGAs to receive a compressed configuration bitstream and decompress this data in real-time, reducing storage requirements and configuration time. The remote system upgrade feature allows real-time system upgrades from remote locations of Arria GX designs. For more information, refer to "Configuration Schemes" on page 3–5.

Operating Modes

The Arria GX architecture uses SRAM configuration elements that require configuration data to be loaded each time the circuit powers up. The process of physically loading the SRAM data into the device is called configuration. During initialization, which occurs immediately after configuration, the device resets registers, enables I/O pins, and begins to operate as a logic device. The I/O pins are tri-stated during power up, and before and during configuration. Together, the configuration and initialization processes are called command mode. Normal device operation is called user mode.

SRAM configuration elements allow you to reconfigure Arria GX devices in-circuit by loading new configuration data into the device. With real-time reconfiguration, the device is forced into command mode with a device pin. The configuration process loads different configuration data, re-initializes the device, and resumes user-mode operation. You can perform in-field upgrades by distributing new configuration files either within the system or remotely.

PORSEL is a dedicated input pin used to select power-on reset (POR) delay times of 12 ms or 100 ms during power up. When the PORSEL pin is connected to ground, the POR time is 100 ms. When the PORSEL pin is connected to V_{CC}, the POR time is 12 ms.

The nIO_PULLUP pin is a dedicated input that chooses whether the internal pull-up resistors on the user I/O pins and dual-purpose configuration I/O pins (nCSO, ASDO, DATA[7..0], nWS, nRS, RDYnBSY, nCS, CS, RUnLU, PGM[2..0], CLKUSR, INIT_DONE, DEV_OE, DEV_CLR) are on or off before and during configuration. A logic high (1.5, 1.8, 2.5, 3.3 V) turns off the weak internal pull-up resistors, while a logic low turns them on.

Arria GX devices also offer a new power supply, V_{CCPD}, which must be connected to 3.3 V in order to power the 3.3-V/2.5-V buffer available on the configuration input pins and JTAG pins. V_{CCPD} applies to all the JTAG input pins (TCK, TMS, TDI, and TRST) and the following configuration pins: nCONFIG, DCLK (when used as an input), nIO_PULLUP, DATA[7..0], RUnLU, nCE, nWS, nRS, CS, nCS, and CLKUSR. The V_{CCSEL} pin allows the V_{CCIO} setting (of the banks where the configuration inputs reside) to be independent of the voltage required by the configuration inputs. Therefore, when selecting the V_{CCIO} voltage, you do not have to take the VIL and VIH levels driven to the configuration inputs into consideration. The configuration input pins, nCONFIG, DCLK (when used as an input), nIO_PULLUP, RUnLU, nCE, nWS, nRS, CS, nCS, and CLKUSR, have a dual buffer design: a 3.3-V/2.5-V input buffer and a 1.8-V/1.5-V input buffer. The V_{CCSEL} input pin selects which input buffer is used. The 3.3-V/2.5-V input buffer is powered by V_{CCPD}, while the 1.8-V/1.5-V input buffer is powered by V_{CCIO}.

V_{CCSEL} is sampled during power up. Therefore, the V_{CCSEL} setting cannot change on-the-fly or during a reconfiguration. The V_{CCSEL} input buffer is powered by V_{CCINT} and must be hard-wired to V_{CCPD} or ground. A logic high V_{CCSEL} connection selects the 1.8-V/1.5-V input buffer, and a logic low selects the 3.3-V/2.5-V input buffer. V_{CCSEL} should be set to comply with the logic levels driven out of the configuration device or MAX II microprocessor.

If the design must support configuration input voltages of 3.3 V/2.5 V, set V_{CCSEL} to a logic low. You can set the V_{CCIO} voltage of the I/O bank that contains the configuration inputs to any supported voltage. If the design must support configuration input voltages of 1.8 V/1.5 V, set V_{CCSEL} to a logic high and the V_{CCIO} of the bank that contains the configuration inputs to 1.8 V/1.5 V.

For more information about multi-volt support, including information about using `TDO` and `nCEO` in multi-volt systems, refer to the *Arria GX Architecture* chapter.

Configuration Schemes

You can load the configuration data for an Arria GX device with one of five configuration schemes (refer to Table 3–4), chosen on the basis of the target application. You can use a configuration device, intelligent controller, or the JTAG port to configure an Arria GX device. A configuration device can automatically configure an Arria GX device at system power up.

You can configure multiple Arria GX devices in any of the five configuration schemes by connecting the configuration enable (`nCE`) and configuration enable output (`nCEO`) pins on each device. Arria GX FPGAs offer the following:

- Configuration data decompression to reduce configuration file storage
- Remote system upgrades for remotely updating Arria GX designs

Table 3–4 lists which configuration features can be used in each configuration scheme.

For more information about configuration schemes in Arria GX devices, refer to the *Configuring Arria GX Devices* chapter.

Table 3–4. Arria GX Configuration Features (Part 1 of 2)

Configuration Scheme	Configuration Method	Decompression	Remote System Upgrade
FPP	MAX II device or microprocessor and flash device	✓ *(1)*	✓
	Enhanced configuration device	✓ *(2)*	✓
AS	Serial configuration device	✓	✓ *(3)*
PS	MAX II device or microprocessor and flash device	✓	✓
	Enhanced configuration device	✓	✓
	Download cable *(4)*	✓	—
PPA	MAX II device or microprocessor and flash device	—	✓

Table 3–4. Arria GX Configuration Features (Part 2 of 2)

Configuration Scheme	Configuration Method	Decompression	Remote System Upgrade
JTAG	Download cable (4)	—	—
	MAX II device or microprocessor and flash device	—	—

Notes for Table 3–4:

(1) In these modes, the host system must send a DCLK that is 4× the data rate.

(2) The enhanced configuration device decompression feature is available, while the Arria GX decompression feature is not available.

(3) Only remote update mode is supported when using the AS configuration scheme. Local update mode is not supported.

(4) The supported download cables include the Altera USB-Blaster universal serial bus (USB) port download cable, MasterBlaster™ serial/USB communications cable, ByteBlaster II parallel port download cable, ByteBlasterMV parallel port download cable, and the EthernetBlaster download cable.

Device Configuration Data Decompression

Arria GX FPGAs support decompression of configuration data, which saves configuration memory space and time. This feature allows you to store compressed configuration data in configuration devices or other memory and transmit this compressed bitstream to Arria GX FPGAs. During configuration, the Arria GX FPGA decompresses the bitstream in real time and programs its SRAM cells. Arria GX FPGAs support decompression in the FPP (when using a MAX II device or microprocessor and flash memory), AS, and PS configuration schemes. Decompression is not supported in the PPA configuration scheme nor in JTAG-based configuration.

Remote System Upgrades

Shortened design cycles, evolving standards, and system deployments in remote locations are difficult challenges faced by system designers. Arria GX devices can help effectively deal with these challenges with their inherent re programmability and dedicated circuitry to perform remote system updates. Remote system updates help deliver feature enhancements and bug fixes without costly recalls, reduce time to market, and extend product life.

Arria GX FPGAs feature dedicated remote system upgrade circuitry to facilitate remote system updates. Soft logic (Nios® processor or user logic) implemented in the Arria GX device can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, recovers from any error condition by reverting back to a safe configuration image, and provides error status information. This dedicated remote system upgrade circuitry avoids system downtime and is the critical component for successful remote system upgrades.

Remote system configuration is supported in the following Arria GX configuration schemes: FPP, AS, PS, and PPA. You can also implement remote system configuration in conjunction with Arria GX features such as real-time decompression of configuration data for efficient field upgrades.

For more information about remote configuration in Arria GX devices, refer to the *Remote System Upgrades with Arria GX Devices* chapter.

Configuring Arria GX FPGAs with JRunner

The JRunner software driver configures Altera FPGAs, including Arria GX FPGAs, through the ByteBlaster™ II or ByteBlasterMV cables in JTAG mode. The programming input file supported is in Raw Binary File (**.rbf**) format. JRunner also requires a Chain Description File (**.cdf**) generated by the Quartus II software. JRunner is targeted for embedded JTAG configuration. The source code is developed for the Windows NT operating system (OS), but can be customized to run on other platforms.

For more information about the JRunner software driver, refer to the *AN414: JRunner Software Driver: An Embedded Solution for PLD JTAG Configuration* and the source files on the Altera website.

Programming Serial Configuration Devices with SRunner

You can program a serial configuration device in-system by an external microprocessor using SRunner™. SRunner is a software driver developed for embedded serial configuration device programming that can be easily customized to fit into different embedded systems. SRunner software driver reads a raw programming data file (**.rpd**) and writes to serial configuration devices. The serial configuration device programming time using SRunner software driver is comparable to the programming time when using the Quartus II software.

For more information about SRunner, refer to the *AN418: SRunner: An Embedded Solution for Serial Configuration Device Programming* and the source code on the Altera website.

For more information about programming serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS64, and EPCS128) Data Sheet* in the *Configuration Handbook*.

Configuring Arria GX FPGAs with the MicroBlaster Driver

The MicroBlaster™ software driver supports a raw binary file (RBF) programming input file and is ideal for embedded FPP or PS configuration. The source code is developed for the Windows NT operating system, although it can be customized to run on other operating systems.

For more information about the MicroBlaster software driver, refer to the *Configuring the MicroBlaster Fast Passive Parallel Software Driver White Paper* or the *AN423: Configuring the MicroBlaster Passive Serial Software Driver*.

PLL Reconfiguration

The phase-locked loops (PLLs) in the Arria GX device family support reconfiguration of their multiply, divide, VCO-phase selection, and bandwidth selection settings without reconfiguring the entire device. You can use either serial data from the logic array or regular I/O pins to program the PLL's counter settings in a serial chain. This option provides considerable flexibility for frequency synthesis, allowing real-time variation of the PLL frequency and delay. The rest of the device is functional while reconfiguring the PLL.

For more information about Arria GX PLLs, refer to the *PLLs in Arria GX Devices* chapter.

Automated Single Event Upset (SEU) Detection

Arria GX devices offer on-chip circuitry for automated checking of single event upset (SEU) detection. Some applications that require the device to operate error free at high elevations or in close proximity to Earth's North or South Pole requires periodic checks to ensure continued data integrity. The error detection cyclic redundancy check (CRC) feature controlled by the **Device and Pin Options** dialog box in the Quartus II software uses a 32-bit CRC circuit to ensure data reliability and is one of the best options for mitigating SEU.

You can implement the error detection CRC feature with existing circuitry in Arria GX devices, eliminating the need for external logic. Arria GX devices compute CRC during configuration. The Arria GX device checks the computed-CRC against an automatically computed CRC during normal operation. The `CRC_ERROR` pin reports a soft error when configuration SRAM data is corrupted, triggering device reconfiguration.

Custom-Built Circuitry

Dedicated circuitry is built into Arria GX devices to automatically perform error detection. This circuitry constantly checks for errors in the configuration SRAM cells while the device is in user mode. You can monitor one external pin for the error and use it to trigger a reconfiguration cycle. You can select the desired time between checks by adjusting a built-in clock divider.

Software Interface

Beginning with version 7.1 of the Quartus II software, you can turn on the automated error detection CRC feature in the **Device and Pin Options** dialog box. This dialog box allows you to enable the feature and set the internal frequency of the CRC between 400 kHz to 50 MHz. This controls the rate that the CRC circuitry verifies the internal configuration SRAM bits in the Arria GX FPGA.

For more information about CRC, refer to *AN 357: Error Detection Using CRC in Altera FPGAs*.

Document Revision History

Table 3–5 lists the revision history for this chapter.

Table 3–5. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
December 2009, v2.0	■ Document template update. ■ Minor text edits.	—
May 2009 v1.4	■ Removed "Temperature Sensing Diode" section. ■ Updated Table 3–1 and Table 3–4.	—
May 2008 v1.3	Updated note in "Introduction" section.	
	Minor text edits.	—
August 2007 v1.2	Added the "Referenced Documents" section.	—
June 2007 v1.1	Deleted Signal Tap II information from Table 3–1.	—
May 2007 v1.0	Initial Release	—