Multichannel Electronics FPGA Configuration Scheme

1. Summary

This document discusses the scheme to be adopted for configuring the FPGAs in the Multichannel Electronics (MCE). This version has been updated to take account of changes from the previously proposed scheme.

2. References

[1] ‘Multichannel Electronics Block Diagram’, SC2_ELE_S560_001.
[2] ‘Stratix FPGA Configuration’, SC2/ELE/S500/21.
[3] ‘MCE Engineering Description’, SC2/ELE/S565/000
[4] ‘AN 217: Using Remote System Configuration with Stratix & Stratix GX Devices’, AN-217-2.1, Altera Corp.
[5] ‘Multichannel Electronics Requirements and Recommendations’, SC2/ELE/S500/11.
[6] ‘FPGA Configuration Block Diagram’, SC2_ELE_S560_003.
[7] ‘Clock Card Configuration Scheme Schematic Capture’ SC2/ELE/S565/103/001
[8] ‘AN 208: Configuring Stratix & Stratix GX Devices’, AN-208-2.2, Altera Corp
[9] ‘Stratix FPGA Family Data Sheet’, DS-STXFAMLY-3.0, Altera Corp.
[10] ‘ByteBlasterMV Parallel Port Download Cable Data Sheets’, DS-BYTBLMV-3.3, Altera Corp.
[11] ‘Using the JAM Language for ISP & ICR via an Embedded Processor’ Altera Application Note 088
[12] ‘CC Dual-EPC16 Configuration Block Diagram’, SC2_ELE_S565_003_000

3. Introduction

The SCUBA-2 multichannel electronics consists of a number of different cards each of which is fitted with an Altera Stratix FPGA. Drawing [1] shows a block digram of the multichannel electronics. The Stratix FPGAs can be configured in a variety of different ways as discussed in [2].

This document answers some of the problems with the currently proposed FPGA configuration scheme, and details the particular configuration scheme which has been chosen for the multichannel electronics.

4. Requirements

There are two main requirements affecting the FPGA configuration scheme:

1) All FPGAs must be configured in less than 5 minutes (requirement (e) in section 4.8.1 of [5]).
2) It must be possible to perform remote configuration updates from the data acquisition PC (requirement (k) in section 4.8.1 of [5]).

There are some additional engineering requirements which are not stated in [5], and which affected the decision not to adopt the previous configuration scheme:

3) The Clock Card (CC) factory configuration must not be accessible or altered from the ByteBlasterMV interface on the CC faceplate or fibre optic interface.
4) The ByteBlasterMV interface and fibre optic interface must have configuration privileges on the ‘application’ configuration devices and FPGAs on all cards (with the possible exception of the CC’s FPGA).

5. Previous Configuration Scheme Issues

5.1 Scheme A

In this scheme, the CC FPGA would be set in remote configuration mode. This would allow the CC's EPC16 to independently store and load two configuration files: factory and application configurations. During startup, the factory configuration would be loaded into the CC FPGA first. In this configuration, the CC could interrogate the JTAG chain to determine the firmware versions on all Stratix devices, and initiate configuration updates as required by the real-time Linux PCs. If a configuration update request was received by the CC, then the configuration data would be downloaded from the PC into the SRAM on the clock card and the CC FPGA would take control of the JTAG bus with a JAM Player to configure the relevant device(s) using the data stored in the SRAM. Once all the configuration requests were complete, the PC would send a start request which the CC FPGA would use to force the EPC16 to load the main application configuration into itself. The system would then be ready for normal operation.

As noted in the previous version of this document, using a single EPC16 in Remote Configuration Mode [4] on the CC introduced several problems:

1) It would not be possible to update or add application configurations to an EPC16 without overwriting part, or all, of the boot configuration in the process. Thus, any errors during the reconfiguration, e.g. a power failure or glitch, would likely result in the corruption of the boot configuration and render the system useless.
2) A single EPC16 has just enough storage to contain two EP1S25 configuration files (factory and application configurations), so it would not be possible to use a larger Stratix device on the CC.

5.2 Scheme B

These two problems outlined above prompted the suggestion of an alternate configuration scheme in which the EPC16 on the CC would be replaced by a larger FLASH memory in conjunction with an Altera MAX device, which would act as the FLASH/configuration controller. This scheme had two advantages:

1) The boot configuration could be separated from the instrument configuration and would never be overwritten.
2) The size of the FLASH memory could be chosen to give sufficient storage for all the required configurations.

However, this scheme introduced another problems:

1) The CC flash could no longer be part of the JTAG chain. Thus, it would be impossible to directly program the 'application' portion of the configuration flash using the faceplate JTAG header. This effectively would mean that the only way to alter the application configuration on the CC would be via the fibre optic interface from one of the real-time Linux computers.

6. Current Configuration Scheme

A block diagram of the latest FPGA configuration scheme is shown in diagram [6]. It depicts each card with a FPGA and an 'application' configuration device on board (this is to meet the requirement in [5], for configuration time). All configuration devices are set up to transfer data in Fast Passive Parallel (FPP) mode. On each card, both the application configuration device and the FPGA will be part of the MCE JTAG chain, and the configuration device will be the first link in the chain as seen by a JTAG TDI signal input on the card's Bus Backplane (BB) connector. Active bus switches (SN74CBTLV126) on the BB will be used to bypass empty card slots and maintain the continuity of the JTAG chain. When a card is plugged into the backplane the JTAG bypass switch will be disabled and the JTAG chain will flow through the devices on the card.

The JTAG chain will be routed to configure cards in furthest-to-nearest order from the CC. Thus, the CC FPGA will always be the last link in the JTAG chain. A variation on this will occur when the CC FPGA will act as a JAM Player or bit-stream player to configure the other devices in the MCE. In this case, the CC FPGA will remove itself from the JTAG chain using external circuitry (refer to [6]), and the CC configuration device will become the last link in the JTAG chain.

The JTAG bus can be controlled in two ways: (i) by means of a ByteBlasterMV connected to a JTAG header on the face plate of the clock card, and (ii) by a Jam/bit-stream Player in the Stratix device on the clock card. (One implementation of a configuration engine uses a NIOS processor to run JAM player software to decode .JBC files. A simpler implementation might use binary image files with a simple memory-to-JTAG bit-stream player.) The JAM/bit-stream Player is the default JTAG controller. However, a manual bus switch will allow the Stratix device on the clock card to disable the JAM/bit-stream player and the ByteBlasterMV to take control of the JTAG bus via the faceplate header. Option (i) will generally be used during development and will allow the configuration of any device(s) on any card to be updated using the Altera Quartus software. Option (ii) is used for updating the configuration information stored in each EPC16 device via the fibre optic link connected to the data acquisition PC. This mode utilizes the remote configuration capability of the Stratix device, as described earlier.

As mentioned, every card will have an 'application' configuration device. The only exception to this scheme is on the CC, where a second configuration device will contain the CC 'factory' configuration. It will not be connected to the MCE JTAG chain – instead, it will be pre-programmed using an exclusive on-board JTAG header.

The factory configuration will implement essential functionality (i.e. fibre optic communications, the JAM/bit-stream Player, some Bus Backplane communications, etc), and will be a stable configuration that the CC can revert to if there are problems with its 'application' firmware. The 'application' configuration will implement all required functionality. JAM Player commands may be required in the factory and application configurations to trigger reconfiguration of the CC FPGA into application mode. The JAM Player can trigger reconfiguration on any card by issuing a command which artificially asserts the nINIT_CONF pin on the FPGA of interest.

The scheme outlined above is in response to 'Scheme B' outlined in §5. This scheme offers the following advantages:

1) The CC application configuration device is part of the MCE JTAG chain
2) Hardware/firmware for a new CPLD does not have to be implemented.
3) Factory and Application configurations are in physically separate devices, and the factory configuration can not be inadvertently altered through the front-panel JTAG interface.

Depending on the resources needed for the factory and application configurations, the two designs could be combined into a single configuration if desired. However, it is beneficial to keep the two configurations separate as they can then be treated as independent firmware tasks.

It is also worth noting that EPC16s have a 64K byte area of FLASH memory which will be made use of as required. This storage may be enough space for the NIOS processor application code used in the factory configuration.

7. Power Up and Configuration

During power up sequencing done by the Power Card (PC), the 'BRst' line (Backplane Reset) on the Bus Backplane (BB) will be asserted by the PC until all power supplies are active. When this line is released, each FPGA will begin its normal configuration sequence. On the Address Card (AC), Bias Cards (BC) and Readout Cards (RC), the configuration devices will upload their 'application' configurations using a Standard FPP link to their respective FPGAs. The Clock Card (CC), however, will always load a 'factory' configuration from a write-protected memory device. As shown in diagram [6], the Clock Card contains two EPC16s, one for the factory configuration and one for the 'application' configuration.

Depending on the implementation details of the CC firmware, the CC may benefit from use of Remote/Local Configuration Mode [4]. All of the other Stratix devices use the Standard FPP configuration mode. However, it would be sensible to provide hardware links such the configuration mode of all the Stratix devices could be changed if necessary.