Firmware Overview

1. Summary

This document provides an overview of the firmware design for the SCUBA-2 Multichannel Electronics. It should be read in conjunction with the top level firmware block diagrams (SC2_ELE_S563_100, SC2_ELE_S563_200, SC2_ELE_S563_201, SC2_ELE_S563_300, SC2_ELE_S563_400, and SC2_ELE_S563_500)

2. References

‘Subrack Top Level Block Diagram’, SC2_ELE_S563_100.

‘Clock Card Block Diagram’, SC2_ELE_S563_200.

‘Clock Card Configuration Block Diagram’, SC2_ELE_S563_201.

‘Address Card Block Diagram’, SC2_ELE_S563_300.

‘Bias Card Top Level Block Diagram’, SC2_ELE_S563_400.

‘Readout Card Top Level Block Diagram’, SC2_ELE_S563_500.

‘Multichannel Electronics FPGA Configuration’, SC2/ELE/S500/22.

‘Multichannel Electronics Requirements and Recommendations’, SC2/ELE/S500/11.

‘Multichannel Electronics Block Diagram’, SC2_ELE_S560_001.

‘Stratix FPGA Family Data Sheet’, DS-STXFAMLY-3.0, Altera Corp.

SCUBA -2 Data Acquisition Software Overview *Part Two Protocols* , SC2/SOF/S200/014

3. Introduction

The firmware for SCUBA-2 multichannel electronics consists of five separate designs, two for the clock card (boot and instrument designs), and one each for the address, bias and readout cards. The electronics consists of a subrack containing a clock card, an address card, three bias cards, and four readout cards connected via a bus backplane. This is reflected in the subrack top level firmware block diagram (SC2_ELE_S563_100).

The clock card in the electronics is connected via a duplex fibre to the data acquisition PC. The PC sends command packets over the fibre to the electronics, and the electronics then sends back a reply packet and in some cases further data packet(s). In all cases the PC acts as the master and the electronics must only respond when requested to by the PC.

Additionally there is a further fibre input to the clock card providing a copy of the Data Valid (DV) signal from the telescope Real Time Sequencer (RTS). The DV signal is used by the electronics to define frames containing valid data which can be processed and returned as a data packet to the PC.

The firmware block diagrams were generated using HDL Designer from Mentor Graphics. Note that ports which connect to LVDS lines on the bus backplane are shown as singled ended on the diagrams. These signals will be converted to differential format when the design is synthesised.

4. Communications protocols

4.1 General

The communications protocol between the data acquisition PC and the multichannel electronics is defined in document SC2/SOF/S200/014. The basic process is that the PC sends a command packet to the clock card, the appropriate card(s) processes the command, and the clock card returns a reply packet to the PC. In addition for one command, “GO”, one or more data packets are returned by the clock card following the reply packet.

The communications protocol is handled in the clock card firmware as described in the following sections (refer to drawing SC2_ELE_S563_200).

4.2 Command/reply mechanism

Command packets are sent over the fibre optic link from the PC. The command packet is received as a series of bytes which are written into a FIFO (RX_FIFO, I0) under control of a Finite State Machine (FSM) (RX_control_FSM, I1).The byte stream is monitored by the RX_protocol_FSM (I2) which looks for valid command packets from the PC. When a valid command packet is identified it is passed to the Command_FSM (I3) on the clock card and also to all the other cards in the subrack via the LVDS Cmd line on the bus backplane. The Bus_cmd_tx (I4) block converts the command packet into a suitable format for transmission over the Cmd line via the LVDS_tx1 (I28) block.

The Command_FSM blocks on each card check to see if the command is for them, and if so the command is executed and a reply packet is returned to the Reply_data_FSM (I16) block on the clock card. Each card has a dedicated LVDS point-to-point link on the bus backplane for returning the reply data packet. These links are connected via the LVDS_rxN blocks to the Bus_reply_rx blocks (N=1 to 8) and then to the Reply_data_FSM (I16) block. The Reply_data_FSM (I16) block waits for the appropriate card(s) to return a reply packet which is then formatted according to the communications protocol and sent as a byte stream via the TX_FIFO (I18) block over the fibre optic link to the PC. The Tx_control_FSM (I19) block monitors the TX_FIFO and passes the data to the fibre optic transmitter.

If required the clock card firmware can be modified to extend the command /reply mechanism described above so that the clock card can independently send commands to the other cards in the subrack. This could be used, for example, for monitoring the operation of the other cards.

4.3 Data packets

Data packets are sent back to the PC under control of the Reply_data_FSM (I16) block, following the corresponding reply packet. The timing of the data packets is controlled by the BDV signal (DV_FSM (I6) on the clock card). BDV can be generated either from a copy of the DV signal from the RTS (scientific data), or internally by the clock card (engineering data).

When BDV is asserted all cards which have information to be included in the data packet send their information to the Reply_data_FSM (I16) block. The information returned may either eventually be included in the file header on disk, e.g. the current heater setting from the appropriate bias card, or be the actual data from the readout cards.

The clock card collates all the information for the various cards and stores it in the Frame_buffer (I5) memory block. Once a complete frame of data, including header information, is assembled it is packaged according to the communications protocol and sent to the PC via the TX_FIFO (I18) block in the same way as a reply packet. The exact content and format of the data transferred to the clock card from the other cards is still to be defined.

Additional frame buffer memory is available in external SRAM controlled by the SRAM_FSM (I17) block on the clock card. This will be most useful in engineering mode where it is known that the data transfer rate is limited by the fibre optic link speed.

5. Command execution

5.1 General

The command packets from the PC contain one of five different commands as detailed in document SC2/SOF/S200/014. These commands are executed by the multichannel electronics as described in the following sections.

5.2 Write_memory and Read_memory

The Write_memory (WM) and Read_memory (RM) commands are the primary means of controlling the operation of the electronics from the PC. The command packet includes a card address, register address and data to be written. Data read is returned as part of the reply packet associated with a RM command. The card address specifies which card the command is intended for. A card address of -1 means the command is intended for all cards.

A memory block on each card, e.g. Register_memory (I27) on the clock card, is used to store the data value from each WM command, and in some cases provides the data for the corresponding RM command. There are five signals associated with the memory: r_addr is the register address, r_dout is data to be written, r_din is data to be read, r_wr controls when data is read, and r_rd controls when data is read. In addition to the memory block there are a series of registers on each card connected to the memory signals. WM commands write data to the corresponding register where it is used to perform the appropriate operation, for example, setting a bias voltage. RM commands read data either from the register memory block or from the appropriate register, for example, the card ID register (CardID_FSM (I39) on the clock card).

5.3 Start_application and Stop_application

The Start_application (GO) and Stop_application (ST) commands are used to control the return of data packets to the PC. The address parameter of the GO command specifies which type of data is returned, e.g. scientific or engineering, and is sent to all cards (card address=-1). The ST command is used to abort a data collection process which is running.

The GO commands prepares the cards for a data acquisition sequence, with frames returned under control of the BDV signal. The details of exactly what each card does in response to the GO command and associated BDV pulses is yet to be detailed.

5.4 Reset

The Reset command (RS) is used to reset the multichannel electronics. Three levels of software controlled reset are possible: power reset is the equivalent of cycling the power to the system, register reset which resets registers and FSMs, and configuration reset which forces FPGA reconfiguration. It is not clear at this point whether all reset options are needed. The reset level is selectable using the parameter for the RS command.

6. Synchronisation

The sync signal on the bus backplane is used to synchronise the various cards in the subrack, in particular address and readout cards. The Sync_FSM (I7) block on the clock card is used to generate the sync signal.

The exact functionality of the sync signal still needs to be finalised. The options are line sync, frame sync or both.

7. Configuration scheme

The configuration scheme to be used in the multichannel electronics is detailed in document SC2/ELE/S500/22. In essence the clock card FPGA has two configurations: a boot configuration which is used to control configuration of the EPC16 devices associated with each FPGA, and an instrument configuration which is used for normal operation. Clearly the two configurations must use the same physical pinout. Unused pins in a particular configuration must have an associated firmware block which ties the pins to an appropriate level.

The boot configuration (see drawing SC2_ELE_S563_201) uses a NIOS processor running the JAM player software to control configuration of the EPC16 devices associated with each FPGA. The boot code for the NIOS processor is stored in the bottom 64Kof the clock card EPC16. Application code for the NIOS processor, if needed, can be stored in free FLASH memory above the FPGA configuration space. The FLASH_FSM (I20) on the clock card controls the FLASH memory interface. The RS232_FSM (I41) provides control of an RS232 interface for use during debugging of the NIOS code.

Configuration files are transferred from the PC into external SRAM on the clock card before being transferred by the JAM player over the JTAG bus to the appropriate EPC16 device(s). The SRAM_FSM (I17) and JTAG_FSM (I21) are used to control the SRAM and JTAG interfaces, respectively.

The Remote_update (I44) on the clock card is used to control the operation of the remote configuration mode on the Stratix device. In the boot configuration it is used to select the appropriate FLASH page in the EPC16 containing the instrument configuration, and to set up a user watchdog timer. In the instrument configuration the block is used to reset the user watchdog timer and control switching back to the boot configuration.

8. Address card

The address card (see drawing SC2_ELE_S563_300) provides a clock to each of the 41 DACs on the card via the Enable_DAC_Clks_FSM (I2). Two separate 14-bit data busses are provided which connect to the DACs controlling odd and even row addresses, respectively (DAC_FSM (I0)). Using two busses allows adjacent row addresses to be changed simultaneously.

In order to reduce crosstalk between one row of pixels and the next it may be desirable to use a non-sequential addressing scheme. Two data busses allows this option as long as the addresses follow an odd/even, or even/odd, sequence , e.g. 1, 4, 7, 10, 13 etc.

Note that non-sequential addressing only affects crosstalk between rows on the actual detector. It does nothing to reduce crosstalk between adjacent column outputs on the cryocables.

An interface to the FLASH memory on the EPC16 is included (FLASH_FSM (I21)) which can be used if necessary.

9. Bias card

The bias card (see drawing SC2_ELE_S563_400) provides 32 single-ended outputs controlled by DAC_FSM_0..3 (I4, I5, I6, I7), and a single differential channel controlled by DIFF_FSM (I8). A serial clock for the DACs is provided by Serial_clock (I3).

The single ended DACs have, for the moment, been split into four banks of eight with each bank having common clock, data and clear lines, and separate chip selects. The details of how many DACs there are in a bank will be driven by the hardware design requirements.

An interface to the FLASH memory on the EPC16 is included (FLASH_FSM1 (I22)). This might be useful for storing coefficients or look up tables for the filter algorithm.

An interface to the FLASH memory on the EPC16 is included (FLASH_FSM (I21)) which can be used if necessary.

10. Readout card

The readout card has eight identical channels represented as components (readout_inner) on the top level firmware diagram (SC2_ELE_S563_500). Each channel takes the output from the corresponding ADC, processes this data, and provides an output to the corresponding DAC. The data from each channel is fed to a multiplexer (data_mux_FSM (I11)) which passes the data back to the clock card when requested.

Each channel also has a DAC providing the SQUID series array bias current, and another DAC providing compensation for the cryocable resistance. The bias and cable DACs are controlled by the SSA_DAC_control (I2) and the Cable_offset_control (I23) blocks, respectively. Each of these blocks provides common clock, data and clear lines and individual chip selects for each channel.

The detailed operation of the readout_inner blocks for each channel is beyond the scope of this document and will be covered in a separate technical note.

An interface to the FLASH memory on the EPC16 is included (FLASH_FSM1 (I22)). This might be useful for storing coefficients or look up tables for the filter algorithm.

11. Miscellaneous functions

11.1 Identification

11.1.1 Box ID

Each subrack has a silicon ID IC fitted on the backplane which provides a unique 64-bit ID. The BoxID_FSM (I22) on the clock card reads this ID. The RM command is used to transfer this information to the PC.

11.1.2 Card ID and temperature

Each card has a silicon ID IC fitted to it which provides a unique 64-bit ID and the temperature of the IC. The CardID_FSM (I39 on clock card) is used to read the ID and temperature. The RM command is used to transfer this information to the PC.

11.1.3 Slot ID

Each slot has a set of four pins which cards can read to determine which slot they are in. The SID_FSM (I23 on the clock card) is used to read the slot ID. The RM command is used to transfer this information to the PC.

11.1.4 Array ID

The subarray to which each subrack is connected can be determined by three lines from the instrument backplane. The ArryID_FSM (I24) on the clock card reads this ID. The RM command is used to transfer this information to the PC.

11.2 Front panel LEDs

The four front panel LEDs on each card are controlled by the LED_FSM (I42 on the clock card).

11.3 Power card control

The clock card is connected to the power card by a two way SPI/MICROWARE serial bus. This is used to control power supply sequencing and both power and configuration reset modes. The PS_FSM (I25) on the clock card controls this interface.

11.4 Power monitoring

The status of the power supplies on each card is monitored by the Power_monitor_FSM (I43 on the clock card). The power supply status is transferred to the LED_FSM which controls the appropriate LED on the front panel of the card.

11.5 FPGA configuration state

The ConfRdy line on the bus backplane is a wire-OR of the CONF_DONE line from each FPGA. The CfgRdy_FSM (I26) on the clock card monitors this line. The RM command can be used to read the ConfRdy status.

11.6 Diagnostic header

Ten diagnostics lines, connected to a header, are provided on each card. These lines are controlled by the Diag_FSM (I40 on the clock card) and can be used as required during development.

12. Conclusions

This document presents an overview of a first draft of the firmware design. It is intended to be starting point for discussions to refine the block diagrams such that the firmware and hardware for each design match. Once this is achieved an ICD can be put in place to control changes and allow the hardware and firmware designs to proceed in parallel.