Clock Card Technical Description

1. Summary

This document describes the card-level functionality and implementation of the Multi-Channel Electronics (MCE) [11]. Where appropriate, recommendations are made in regards to the current state of the design. This document is meant to complement the Multichannel Electronics Requirements and Recommendations [6].

2. References

[1] 'Using the Jam Language for ISP & ICR via an Embedded Processor' Altera Application Note 088
[2] 'FPGA Configuration Block Diagram' SC2/ELE/S560/003
[3] 'JTAG Bus Switching Circuitry' SC2_ELE_S565_003_007
[4] 'Bus Backplane Connector Pinout' SC2_ELE_S565_001_004
[5] 'SCUBA 2 Protocol – Canada' SC2/SOF/S200/014
[6] 'Multichannel Electronics Requirements and Recommendations' SC2/ELE/S500/011
[7] 'Stratix Device Handbook'
[8] 'Bus Backplane Instruction Set Architecture' SC2_ELE_S565_001_002
[9] 'Bus Backplane ISA Description' SC2_ELE_S565_001_003
[10] 'Address Card DAC Circuitry' SC2_ELE_S565_104_001
[11] 'Multichannel Electronics Block Diagram' SC2/ELE/S560/001
[12] 'Configuring Stratix and Stratix GX Devices' Altera Application Note 208

3. Clock Card

3.1 CC Introduction

The CC will be the master card in each subrack. It will act as an interface between the cards inside the subrack and the world outside. Its interfaces are:

(a) FO communication with RTL computers
(b) FO receiver for RTS DV pulses
(c) BB communication with other cards in a subrack
(d) JTAG interface (reconfiguration of Altera devices in a subrack)

The role of the CC will be that of a slave to the real-time Linux computers, and that of a master to the cards in a subrack. All communication from the computers will be directed to the CC. No other cards in the subrack will directly see commands issued by the computers. The CC will take an active role in communicating with the other cards, by interpreting commands from the computers and issuing these commands to cards using a simple bus-backplane communications protocol. The CC will also send its own commands/data as necessary. Data/replies from other cards will be directed to the CC which will package these appropriately and send them to the correct destination (i.e. computers, power supply card, RS-232 interface, test-point headers).

The CC will act as a special data-switch that can add or remove information from packets it receives, and send out packets of its own. This will allow the CC to take an active role in: power shutdown, card resets, computer command error-checking, configuration control and communications control.

The CC controller will be a 780-pin Altera Stratix FPGA which will allow vertical migration between EP1S10-EP1S40 devices.

3.2 CC Engineering Requirements

3.2.1 Prevent Card Insertion in the Wrong Subrack Slot

The CC's front faceplate will be mechanically keyed to prevent insertion in the wrong slot of a subrack. The keying system will allow for the unique keying of at least 12 types of cards (1 PC, 1 CC, 2 AC, 2 RC, 6 BC).

3.2.2 Accept FPGA Configuration Files from the RTL Computers

The CC will accept JTAG configuration data in compressed .jbc byte-code files or pure binary files from the RTL computers. The CC will store this file temporarily in an on-board RAM.

The .jbc-file size will depend on the number of logic elements used in the FPGA design. The on-board RAM will also provide space for the Jam Player's heap, stack, symbol table and uncompressed .jbc-file data. Some of these are dynamically variable in size – however the uncompressed-data size is fixed, and can be determined [1] (p.18).

Pure binary files may be directly transmitted over the JTAG bus, without any decoding. Use of binary images would significantly reduce the design and computational intensity required in transferring the file from the RTL to the target FPGA or persistent configuration storage (see 3.2.4 below). Binary images contain all data required by an FPGA, and the FPGA contains a mechanism for verifying and reporting the integrity of the data.

3.2.3 Program all Altera Devices in a Subrack

The CC's FPGA will take an active role in reconfiguring the rest of the configuration devices in the subrack, using an embedded JTAG programmer (i.e. JAM Player/JRunner) to parse .jbc configuration files or a simpler parser for pure binary files. This type of configuration will be used while the subracks are mounted on the cryostat. The JTAG programmer or binary parser may only be available in the CC's factory configuration.

All configuration files downloaded from the RTL computers will be temporarily stored in an on-board RAM (refer to [2]).

The Jam Player is a C program that parses a .jbc file, interprets each Jam instruction, and reads and writes data to and from the JTAG chain. The player may be ported to be run by a NIOS processor on the FPGA. The Jam Player will program the Altera device that the .jbc file targets. The .jbc file may target any configuration device in the JTAG chain. The Jam Player is distributed as open-source code that is available at: https://www.altera.com/support/software/download/programming/jam/dnl-byte_code_player.jsp. The Jam Player I/O functions, which are contained in the jbistub.c file, will need to be modified for a given application. These functions include those that specify addresses to I/O pins, delay routines, operating system-specific functions, and routines for file I/O (refer to [1], p. 9).

The NIOS processor may require separate external program memory for its controlling software and Jam Player. The Jam Player itself will require about 50 kB of code space.

If pure binary configuration files will be used, the NIOS processor (and associated FPGA and program memory space) may be replaced by a simple state machine that transfers data between the RTL and on-board RAM, and then between the RAM and the JTAG bus. The size of the on-board RAM for the binary file approach would have to be sufficient to contain the largest configuration file, being 1550 kB for an EP1S40.

3.2.4 Provide Persistent Storage for FPGA Reconfiguration Data

This is to meet the requirement that specifies a maximum allowable subrack reconfiguration time of 5 minutes, on power-up. The CC will store its factory and application configurations in two separate Altera configuration devices. The application-configuration device will be part of the JTAG chain so that it is programmable in-circuit (refer to [2]). During subrack power-up, the CC FPGA will be automatically reconfigured by the factory configuration device. Using a configuration device in fast passive parallel mode (refer to [12]), the reconfiguration time of an Altera Stratix EP1S40 is estimated to be 1.2s. This is a good upper bound, as this is the largest FPGA that the CC may have on board. A Stratix EP1S40 configuration file requires 1550 kB max storage, so an EPC16 configuration device (2 MB) will be large enough for one file. In factory mode, some functionality may be available to users, which would not be available in application mode. Once a user has finished working in factory mode, a software command will be issued to trigger the FPGA to load its application configuration from the second configuration device.

Both of the CC FPGA's factory and application configurations may initiate reconfigurations into either factory or application mode. A 'configuration reset' (initiated by the Power Card via the BRst signal, see 3.2.19 below) will always reconfigure the CC FPGA from the factory configuration device. An INIT_CONFIG 'JTAG' command to the application configuration device will force the CC FPGA into application mode.

3.2.5 Accept Programming Data from a Byte-Blaster Header on Faceplate

As mentioned, a JTAG chain makes it possible to program all Altera devices in a subrack (refer to [2]). A front-panel 10-pin standard header on the CC will allow the subrack's Altera devices to be programmed and to return diagnostic information using a Byte-BlasterMD and the JTAG chain. The Byte-Blaster configuration interface will be faster then the in-circuit configuration with the Jam Stapl Player or binary parser. The Jam Player or binary parser will be implemented on the CC's FPGA, and will be the default controller of the JTAG chain. Therefore, if the Byte Blaster header is used, a manual switch will be require to control on-board buffers which will force the CC's FPGA to be a passive link in the JTAG chain. The Byte Blaster will be used during prototyping.

SN74ALVC125/126 LVCMOS buffers will be used in the on-board circuitry to toggle the FPGA from being an active component in the JTAG chain to one that is passive (i.e. from a device that configures to one that is configured). For a schematic representation of the switched-input JTAG circuitry on the CC, see [3].

3.2.6 Identify the CC's Own Slot, Serial #, and Firmware Version

The CC will determine its own slot number via four pins to the bus backplane ('SID0-SID3', [4]); its serial number from a unique on board Maxim silicon ID IC (DS18S20 – the same IC will be used for temperature sensing), and its firmware version from a hard-coded value in the firmware or one that has been assigned by Quartus II.

3.2.7 Identify the Subrack's Serial

The CC will logically identify the unique serial number of the subrack that it is inserted into. There with be a silicon ID IC on the BB. The CC-BB pin designated for this function is 'BoxID' (refer to [4]). A DS2401 will be used.

3.2.8 Identify the Sub-array

The CC will determine which sub-array of 8 it belongs to via an opto-mechanism that uniquely relates its position on the cryostat. Metal tongues on the cryostat faceplates that mate with subrack-mounted opto-interrupters are considered the best solution for this. The CC-BB pins designated for this function are 'ArryID0-ArryID2' (refer to [4]).

3.2.9 Receive Commands from RTL computers; and Return Acknowledgements

The CC will:

(a) Receive packaged commands from the RTL computers over FO link
(b) Unpackage them as per the SCUBA2 FO transmission protocol created by Xiaofeng Gao [5]
(c) Interpret them
(d) Carry out appropriate actions (i.e. query cards, assemble data/information)
(e) Package the reply (with possible data) to the RTL computers data in keeping with the SCUBA2 FO transmission protocol
(f) Send the reply on FO link to RTL computers

The CC will be capable of returning 'packaged' data to the RTL computers in at least three different modes. By 'packaged', it is meant that the data will be assembled into an agreed upon format for transmission via an FO-link to the RTL computers. In all modes, the CC will only send data to the RTL computer if it has received a request for them. Requests may be for a single frame, or for several consecutive frames.

The CC FO interface will use Cypress 8/10-encoded 200/250 Mbps serializers (CY7B923) and deserializers (CY7B933) which are compatible with the same Hotlink protocol as the SDSU PCI FO interface. They will interface with the Agilent, 1300 nm, 260 MBd FO transmitter (HFBR-1119T), and receiver (HFBR-2119T) using a 5V differential PECL interface. Both the transmitter and receiver will be equipped with FO ST-connectors. NC74LVX541 buffers will be used as 5-to-3.3 LVCMOS interfaces between the transmitter/receiver and FPGA to avoid 5V power-sequencing (refer to [7]).

3.2.10 Compile Readout Data Received

The CC's FPGA will compile data-frames as they arrive in fragments from the RCs. Different data modes, and thus different types of frames will be assembled at different rates, depending on which data-reporting mode the multi-channel electronics are in (i.e. SQUID characterization, scientific or engineering mode).

The on-board RAM used as temporary storage for .jbc/binary files could also provide storage for buffered frames that will be transmitted to the RTL computers. It is assumed that the CC RAM will never be used concurrently for .jbc/binary file storage and frame buffering. At 5.12 KB per scientific frame, a 2 MB RAM could hold a maximum of 390 frames, or provide about 1s of frame-buffering, before data is lost. An optional second 2 MB device would, if populated, provide additional buffering, with the ability to alternate between the two RAM devices for extended elastic storage of streaming data.

3.2.11 Receive a DV Signal from the RTS

DV pulses will enable the transmission of data frames to the RTL computers. In scientific mode, the CC FPGA will receive DV pulses from the RTS FO receiver and integrate that information into byte-aligned transmissions to all cards over the LVDS Cmd line. The Cmd line will also be used for a variety of commands to the cards.

DV pulses will be received by the CC FPGA, de-bounced, and synchronized to the 25 MHz reference clock. When a DV pulse is recognized by the FPGA, it remains to be determined if it will be communicated to the RCs for the start of the following line-scan (800 kHz) or the start of the next frame-scan (20 kHz). At this time, it is thought that either choice meets the maximum jitter requirements for frame capture.

The RTS FO interface will use an Agilent, 820 nm, 5 MBd receiver (HFBR-2412T) with ST-type-connector and threaded bushing. The receiver's output will be an open collector, terminated with a pull-up resistor to 3.3v. Optical power 'on' will correspond to '0' output logic (active low). The bandwidth of the RTS FO interface will be sufficient for short packets of information to be embedded into the DV signalling scheme.

3.2.12 Interface with the BB

The CC's BB connector will be compatible with the slot '8' BB pinout. The CC pinout is outlined in [4]. A description of the pins is given below:

Power:
+5VD: +5V regulated, for the FO interface ICs
+3.3VDun: +3.3V unregulated; regulated on board for the FO interface ICs, I/O
+2.5VDun: +2.5 unregulated; regulated on board for the Altera Stratix power

LVDS Signals:
Tx?A+/Tx?A-, Tx?B+/Tx?B-: LVDS point-to-point data lines. The CC will receive information from each card over a set of two LVDS pairs. LVDS receivers, external to the FPGA, will be used for these signals in order to protect the FPGA against backplane faults.
Clk+/Clk-: LVDS 25 MHz clock signal from the CC; multi-tapped by all other cards from the BB. LVDS repeaters will be used on all other cards to minimize the effects of stubs on the multi-tapped line. Low-propagation delay National DS90LV001 LVDS repeaters will be used for this and all other multi-tapped LVDS lines below.
Sync+/Sync-: LVDS frame synchronization from the CC; multi-tapped by all other cards from the BB.
Cmd+/Cmd-: LVDS command line controlled by the CC; multi-tapped by all other cards from the BB. Commands will be sent as serial byte streams. Some commands will integrate DV-pulse information to be used by the AC and RCs.
TxSpare+/TxSpare-: A single LVDS pair to be used as a spare. This pair will originate from the CC and be multi-tapped by all other cards from the BB.

TTL Signals:
TCK, TDI, TDO, TMS: JTAG configuration pins.
ArryID0-2: 3 pins routed to a header on the BB. The header will be used to connect to 3 open-collector opto-electrical devices mounted on the rear of a subrack, that mate with an arrangement of tabs to uniquely associate a subrack with a specific quadrant of the cold arrays. The CC must provide pull-up resistors on these signals. The arrays will be identified in the following manner:
Arrangement 000: 450μm quadrant 1
Arrangement 001: 450μm quadrant 2
Arrangement 010: 450μm quadrant 3
Arrangement 011: 450μm quadrant 4
Arrangement 100: 850μm quadrant 1
Arrangement 101: 850μm quadrant 2
Arrangement 110: 850μm quadrant 3
Arrangement 111: 850μm quadrant 4

BoxID: Pin connected to a silicon ID on the BB; used to uniquely identify the subrack.
nEXTND: Pin used to determine whether the card is on an extender card.

PSDO: Power supply data out.
PSCSO: Power supply chip-select out.
PSCLKO: Power supply clock out.
PSDI: Power supply data in.
PSCSI: Power supply chip-select in.
PSCLKI: Power supply clock in.
SID0-3: Four pins to open-circuit ("0") or ground ("1") on the BB, to uniquely identify the slot that the card is inserted into. The CC must provide pull-up resistors on these pins. The voltage levels of the SID pins will reflect the following numbering scheme:
Slot 0: AC (leftmost)
Slot 1: BC0
Slot 2: BC1
Slot 3: BC2
Slot 4: RC0
Slot 5: RC1
Slot 6: RC2
Slot 7: RC3
Slot 8: CC
Slot 9: PC (rightmost)

SpTTL0-3: 4 spare 3V TTL signals that emanate from the CC will be multi-tapped by all other cards from the BB.
BRst: A line from the Power Card will be multi-tapped by the configuration and reset circuitry of all Altera FPGAs in a subrack. A low-to-high transition on this line will cause all devices to reset and begin reconfiguration.

3.2.13 Provide a Reference Clock and Synchronization Commands for the Subrack

The CC will provide a multi-tapped 25 MHz reference clock signal to all other cards in the multi-channel electronics. All other clocks in the subrack will be derived from that signal. A 25 MHz clock distributed over the BB implies that communication over the BB will occur at 25 Mbits /s (this makes the BB the bandwidth limiting step in the system if a Gigabit Ethernet FO link is used).

The 25 MHz reference clock will be derived from a 100 MHz plastic surface mount crystal oscillator (ECS-3953M-500-B, 50ppm). The crystal's combined accuracy and stability should be better than 100 ppm to satisfy the Cypress Tx/Rx REFCLK tolerances (CY7B923/CY7B933 data sheets). 2:1 clock-division with a SN74LVC74AD dual flip-flop will achieve a 50% duty cycle on the 25 MHz reference clock. The clock will be distributed to

(a) The Cypress FO transmitter and receiver
(b) The FPGA
(c) The BB, for multi-dropped LVDS distribution

During operation, the AC, RCs and BCs will be co-ordinated. For this, CC will control a multi-tapped LVDS 'Sync' line which will transmit short co-ordination/DV commands to those cards. Co-ordination will allow address lines to settle before SQUID values are read (NIST has estimated the address settling time to be 360 ns).

3.2.14 Send Commands/Data to Cards, and Receive Acknowledgements/Data from Cards

The CC will support a master/slave protocol with other cards in the multi-channel electronics. The CC will send commands (which may include parameters or data) to all other cards, and expect to receive acknowledgements from the cards that were targeted. These acknowledgements may contain data that the CC is querying for. The command protocol is outlined in [8] and [9]. Other than 'observational' communication, the CC will accomplish the following:

(a) The CC will identify what types of cards are on the bus backplane, which slots they are in, what serial number they have been assigned, and what version of firmware they are running.
(b) The CC will query for any error conditions that occur on other cards, including high temperatures, board-regulated voltage deviations, addressing errors, readout errors, biasing

errors, frame errors, and FO or BB transmission errors. The CC will learn of errors either through routine housekeeping queries, or through flags/checksums that can be set in the acknowledgements that cards return to other commands. If a transmission error occurs, the CC may re-query the card if the BB bandwidth does permits it to do so.

(c) The CC will communicate with the PC using a two-way SPI/MICROWIRE (small peripheral interface)

3.2.15 Provide Diagnostic Information

This will require the inclusion of the CC's FPGA in the JTAG chain, and the inclusion of boundary-scan circuitry in the FPGA. The JTAG chain will be used to extract diagnostic information from the Altera devices in the multi-channel electronics. The chain will be routed to a header on the faceplate of CC.

Three LEDs (as per MultiChannel Electronics Requirements and Recommendations Draft Version 0.7) will be machined into the CC's faceplate, to allow visual verification of the CC FPGA's functionality.

It is recommended that prototype CC boards have a number of FPGA pins tied to diagnostic I/O (i.e. test-points [line-select sync, frame sync, etc], LEDs, RS-232 interface for debugging a NIOS processor). In addition, headers to signals from the FO transmitter and receiver will allow the capture of 8-bit wide data by a logic analyser. An impedance-controlled Mictor header will be provided to allow monitorring of high-speed FPGA signals. The Maxim MAX6301 reset/watchdog chip will be used to qualify and monitor the 1.5V and 3.3V digital power supplies; the FPGA will be reset should either supply fail to meet minimum voltage requirements. As the clock card does not use any analogue supply voltages, they will not be monitorred.

3.2.16 Sense CC Temperature

The on-board temperature will be sensed with a Maxim temperature-sensor (DS18S20 – this IC will be used simultaneously as a silicon ID), and the results will be used to determine whether the card is operating within an appropriate temperature range. If it isn't, the operator will be notified if the RTL computer requests 'housekeeping' data.

Additionally, the package temperature of the CC FPGA will be monitorred using the FPGA's on-board diode-connected transistor and a helper chip (MAX1618).

3.2.17 Distribute Power on the CC

The CC requires the following power levels:

(a) 1.5VD to the CC FPGA (regulated on-board)
(b) 3.3VD to the LVTTL and LVC components, and LVDS buffers (regulated on-board)
(c) 5VD to the FO components and Cypress FO transceivers

The 1.5V will be regulated with a 3A, low dropout regulator (LT1587-1.5). The 3.3V supplies will be regulated with a 3A low-dropout regulator (LT1764-3.3).

3.2.18 Monitor 5V, 3.3V and 1.5V Levels on All Power Supply Lines

It may be necessary for a card to monitor the voltage levels on some power supply lines that emanate from the PC. A Maxim power supply voltage monitor (MAX6339PUT) may be used to monitor 5, 3.3 and 1.5 voltage levels. The output of the IC would be routed to an on-board LED and the FPGA.

3.2.19 Initiate Resets

The CC will co-ordinate power-shutdown/resets in a subrack. Users will be able to initiate resets through both hardware and software. There will be three types of 'resets': power reset, configuration reset and register reset.

A '**power reset**' will turn off the discrete voltage levels from all brick regulators on the PC to the other cards in a well-defined sequence. A software 'power reset' may be initiated by the CC if it receives a command to do so. It will send a command to the PC to begin power-down. A hardware 'power reset' may be initiated using a switch/breaker embedded in the power supply lines to cut power to the

subracks. Note that some preparation may be necessary on the part of the FPGAs in a subrack, before power can be removed.

A '**configuration reset**' will trigger the reconfiguration of all FPGAs in a subrack. A software 'configuration reset' will be initiated by the PC if the CC receives a command to do so and forwards it to the PC. A hardware 'configuration reset' will be initiated using a pinhole button on the faceplate of the PC routed to one of the PC-processor's I/O pins. The PC may only be issued a shutdown command after the FPGAs in a subrack are prepared for a shutdown. The PC will then assert the BRst line, which will cause a low-to-high transition on the nCONFIG pins of all FPGAs in the subrack. BRst will only be asserted upon power on, or if the CC issues a reset command. Note also that the JTAG Stapl player may be used to artificially assert the nCONFIG pin on a single FPGA within the subrack. However, it will be impossible for the CC to reset itself using the Stapl player.

A '**register reset**' will cause certain FPGA-registers in the subrack to be initialized to pre-specified values. A software 'register reset' will be initiated by the CC if it receives a command to do so. A hardware 'register reset' will also be initiated by the CC if one of it's I/O pins is asserted by a pinhole button on the CC's faceplate. To reset concerned registers in other FPGAs in the subrack, the CC will issue a command over the LVDS Cmd line.

For development and debugging purposes, it would be desirable to include on-board reset buttons for to trigger (a) a local (CC only) configuration reset, and (b) a local non-reconfiguration reset (via the FPGA's nDEVCLR signal).

3.3 CC Non-Requirements

3.3.1 Determine the FO BER

The CC will not determine the FO bit-error rate. It will be desirable to characterize the reliability of FO link during diagnostic and operational modes. The original SCUBA2 requirements document states that "a programmable random number generator must be provided in the Altera firmware to allow the BER of the fibre optic link to be measured." However, it is thought that this requirement should be applied to the RTL computers instead, because they control the FO link and the CC will never initiate communication with the RTL computers – it will only respond to commands it receives. If the FO BER is to be measured, the RTL computer or PCI card or FO-interface chip will send a special checksum-less 'echo' command that includes a random bit sequence which the CC will replicate in its reply.