CLOCK AND CONTROL BOARD FOR THE CSC EMU PERIPHERAL AND TRACK FINDER ELECTRONICS

CCB2004 Specification (Production Board)

Rice University

Version 3.4.2

04/02/2008

Abstract

This document describes the functionality of the production version of the Clock and Control Board (CCB2004) for the CMS EMU peripheral electronics and its communication with other modules in the peripheral crate. This CCB2004 can also be used in the CSC Track Finder crate.

Introduction

The CSC EMU electronics includes chamber-mounted Front End Boards (FEB), connected to electronics on the periphery of the detector. The peripheral Trigger/DAQ electronics resides in VME 9U crates and includes: the combined Cathode LCT/Trigger Motherboard cards (TMB), the Data Acquisition Motherboard (DAQMB), the Muon Port Card (MPC) and the Clock and Control Board (CCB). More details about the peripheral backplane and communications between modules are given in [1].

All elements of the EMU electronics will be synchronized with the LHC bunch crossing frequency. The Timing, Trigger and Control (TTC) system, which is used for this purpose for all LHC detectors, has been specified and a detailed description of this system and its functionality can be found, for example, in [2-3]. The TTC system is based on an optical fan-out system and provides for the distribution of the LHC timing reference signal, the first level trigger decisions and control commands to about 1000 destinations. At the lowest level of the TTC system, the TTCrx ASIC [3] receives the control and synchronization data from the central TTC system through the optical cable and outputs TTL-compatible TTC signals in parallel form. The TTCrx is mounted on a small mezzanine card that is designed at CERN and installed on the main Clock and Control Board.

Block diagram of the CCB2004 is shown on Figure 1. There are several new features added into design as compared to the previous version which we will refer to as CCB’2001 [4]. The new CCB2004 board can accommodate not only the TTCrm [3] mezzanine with the TTCrx ASIC, but also the most recent version of the TTC mezzanine card, the TTCrq [5], that has an additional 26-pin connector and provides low-jitter clock outputs of 40Mhz, 80Mhz and 160MHz. The CCB2004 may decode several commands not only from the broadcast lines Brcst[7..2] as the CCB2001, but also from the Data[7..0] bus that is used for the individually addressed commands.

The CCB2001 functionality was implemented using a mezzanine mounted Altera EPF10K100A PLD. In order to reduce the cost, improve radiation tolerance and decrease the reconfiguration time we decided to use a Xilinx Virtex-2 FPGA mounted directly on the main CCB2004 board. The most critical part of the CCB2004 functionality is implemented in discrete CMOS logic that is immune to single-event upsets (SEU) at the expected dosage. These functions include the VME address and command decoding, generation of Hard_reset commands, clock and command distribution to custom backplane. An FPGA may experience SEU's however, and in order to prevent malfunctioning, it can be reloaded from its EPROM upon common or CCB-specific Hard_reset command.

Another new feature is the ability to distribute an 80Mhz clock from the TTRrq to the Muon Port Card over dedicated LVDS lines. This low jitter clock can be used to drive the gigabit serializers residing on the MPC board.

Estimated power consumption: +3.3V - < 2 A, +5V - <1 A.

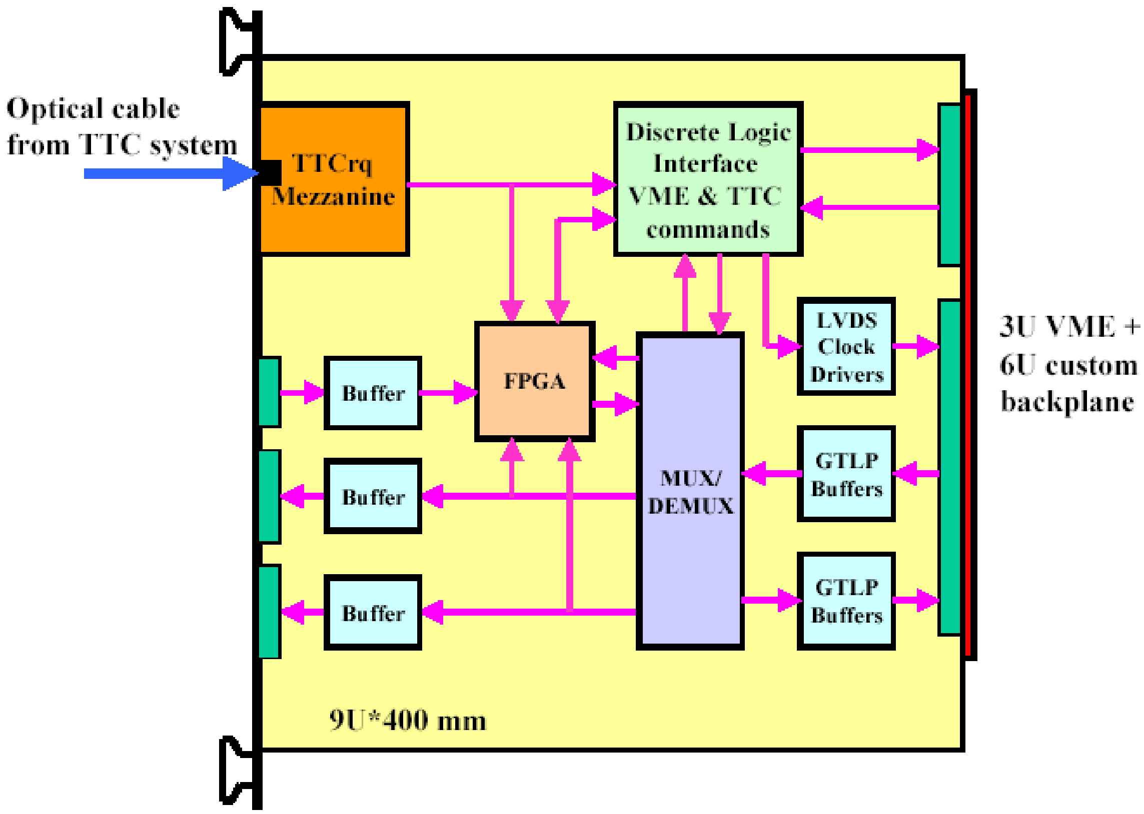

Figure 1: Block Diagram of the CCB'2004

1. Modes of Operations

The most critical functions of the CCB2004 are implemented in discrete logic. Among these functions are the following :

- VME A24D16 slave interface;
- translation of all basic signals from the TTCrq mezzanine to custom backplane;
- decoding of Hard_reset commands from Brcst[7..2] and Data[7..2] lines and generation of 500 ns Hard_Reset pulses to custom backplane;
- access to TTCrq board over I2C serial bus under VME control;
- JTAG access to FPGA/EPROM from VME ;
- monitoring of the Conf_Done signals from all connected peripheral or TF boards.

The FPGA can accept all the signals from the TTCrq board, and can generate all the backplane signals to peripheral boards, providing more flexibility than the discrete logic. The choice of interface between the TTCrq mezzanine and custom backplane (discrete logic or FPGA) is defined by CSRA1[0] which is implemented in a discrete logic (Fig.2). When CSRA1[0]=1, the interface is a discrete logic interface ("discrete logic" mode). When CSRA1[0]=0, the interface is an FPGA ("FPGA" mode). In "FPGA" mode two options are possible: when TTC commands are coming from the TTCrq (if CSRB1[0]=0) and when they are generated upon write into CSRB2 and CSRB3 (if CSRB1[0]=1). Also, independently from CSRB1[0], several commands (see Table 8, starting from Base+50 address) can be generated upon VME write to dedicated addresses. This option provides compatibility with the CCB'2001 board.

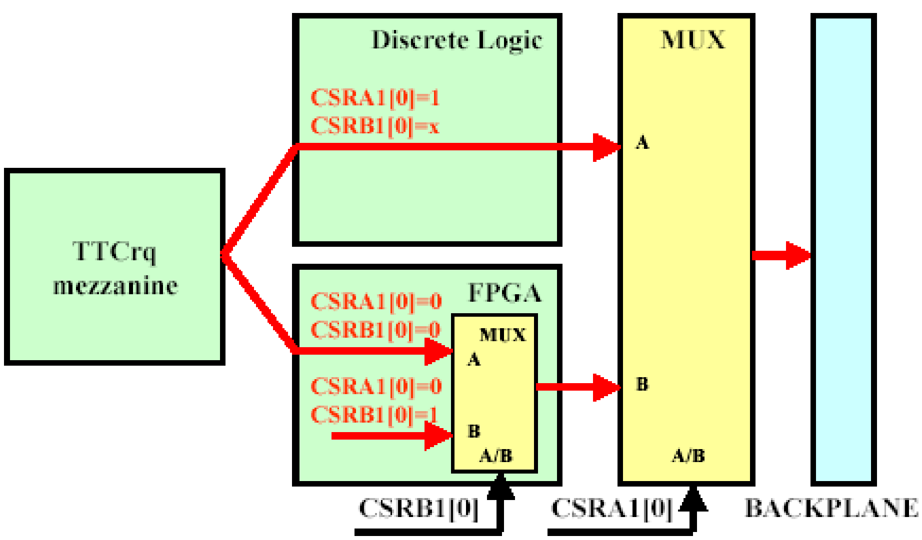

Figure 2: Modes of Operation

After power cycling and reloading from EPROM all the registers inside the FPGA are set to "0". This means all sources of L1ACC are enabled and the CCB'2004 will not hold the next L1ACC and Pretriggers after the first L1ACC has been sent to custom backplane.

2. TTC Interface and TTCrq Mezzanine Board

The list of signals that the CCB2004 accepts and transmits from/to the TTC mezzanine board is given in Table 1. Pin assignment of three mezzanine connectors is shown in Table 2. The board dimensions are given on Figure 2.

Table 1: Interface signals to/from TTC mezzanine card

Signal	Bits	Short Description
Signals to/from two 50-pin connectors backward compatible with TTCrm mezzanine card		
BCntRes	1	Bunch Counter Reset signal
BCntStr	1	Bunch Counter Strobe. Indicates that a bunch number is present on the output BCnt<11:0> bus
Brcst<7:2>	6	Broadcast commands/data output bus
BrcstStr<2:1>	2	Broadcast messages strobes
Clock40Des1	1	LHC 40.08 MHz deskewed reference clock signal
DbErStr	1	Indicates that a double error or a frame error has occurred
Dout<7:0>	8	Data bus. Normally used to output the data content of an individually-addressed commands/data
DQ<3:0>	4	Data qualifier bits. Indicate the type of data on the data bus register
BCnt<11..0>	12	Bunch or Event counter outputs
DoutStr	1	Data out strobe. Indicates valid data on the data bus
EvCntHStr	1	Event counter high word strobe
EvCntLStr	1	Event counter low word strobe
EvCntRes	1	Event counter reset signal
L1Accept	1	First level trigger accept signal
Reset_b	1	Reset TTCrx ASIC, active "low"
SinErrStr	1	Single error strobe
SubAddr<7:0>	8	Subaddress bus. Used to output the subaddress content of an individually address commands/data
TTCReady	1	Indicates that TTCrx ASIC is ready for normal operation
SDA	1	Data Line of I2C interface
SCL	1	Clock Line of I2C interface
Total	54	
Signals to/from an additional 26-pin connector (TTCrq mezzanine only)		
FoSelect<3..0>	4	Control inputs for the VCXO free running oscillation frequency (QPLL only)
Mode	1	QPLL multiplication mode control input
Restart	1	Enable/disable automatic restart of the PLL (control input for the QPLL)
ExternalControl	1	Control input for the VCXO (QPLL only)
Locked	1	QPLL output status signal
40Mhz clock	2	LVDS clock outputs
80/60Mhz clock	2	LVDS clock outputs
160/120Mhz clock	2	LVDS clock outputs
40Mhz_CMOS	1	CMOS 40Mhz clock output
~Reset	1	QPLL control input, active "low"
Error	1	QPLL status output
Total	17	

The QPLL ASIC residing on the TTCrq mezzanine card has several control inputs that are connected to the FPGA. All control inputs are set into default states using on-board pull-up and pull-down resistors.

Table 2: Pin assignment of the TTCmr/TTCrq mezzanine connectors

Connector J1			Connector J2			Connector J3 (TTCrq only)		
Pin	Signal	Type	Pin	Signal	Type	Pin	Signal	Type
1	Clock40	Output	1	BrcstStr2	Output	1	F0Select<0>	Input
2	Clock40Des1	Output	2	ClockL1Accept	Output	2	Mode	Input
3	Brcst<5>	Output	3	Brcst<6>	Output	3	InLVDS+	Input
4	Brcst<4>	Output	4	Brcst<7>	Output	4	InLVDS-	Input
5	Brcst<3>	Output	5	EvCntRes	Output	5	GND	Power
6	Brcst<2>	Output	6	L1Accept	Output	6	ExternalClock	Input
7	Clock40Des2	Output	7	EvCntLStr	Output	7	AutoRestart	Input
8	BrcstStr1	Output	8	EvCntHStr	Output	8	ExternalContr	Input
9	DbErrStr	Output	9	BcntRes	Output	9	F0Select<3>	Input
10	SinErrStr	Output	10	GND	Power	10	~Reset	Input
11	SubAdd<0>	Bidir	11	BCnt<0>	Output	11	Locked	Output
12	SubAdd<1>	Bidir	12	BCnt<1>	Output	12	Error	Output
13	SubAdd<2>	Bidir	13	BCnt<2>	Output	13	GND	Power
14	SubAdd<3>	Bidir	14	BCnt<3>	Output	14	Lvds80Mhz-	Output
15	SubAdd<4>	Bidir	15	BCnt<4>	Output	15	Lvds80Mhz+	Output
16	SubAdd<5>	Bidir	16	BCnt<5>	Output	16	GND	Power
17	SubAdd<6>	Bidir	17	BCnt<6>	Output	17	F0Select<2>	Input
18	SubAdd<7>	Bidir	18	BCnt<7>	Output	18	GND	Power
19	DQ<0>	Output	19	BCnt<8>	Output	19	Lvds160Mhz+	Output
20	DQ<1>	Output	20	BCnt<9>	Output	20	Lvds160Mhz-	Output
21	DQ<2>	Output	21	BCnt<10>	Output	21	GND	Power
22	DQ<3>	Output	22	BCnt<11>	Output	22	Lvds40Mhz-	Output
23	DoutStr	Output	23	JTAGTMS	Input	23	Lvds40Mhz+	Output
24	GND	Power	24	JTAGTRST_b	Input	24	F0Select<1>	Input
25	Dout<0>	Bidir	25	JTAGTCK	Input	25	Cmos40Mhz	Output
26	Dout<1>	Bidir	26	JTAGTDO	Output	26	GND	Power
27	Dout<2>	Bidir	27	SDA	Bidir			
28	Dout<3>	Bidir	28	JTAGTDI	Input			
29	Dout<4>	Bidir	29	BCntStr	Output			
30	Dout<5>	Bidir	30	Serial_B_Chan	Output			
31	Dout<6>	Bidir	31	GND	Power			
32	Dout<7>	Bidir	32	GND	Power			
33	Reset_b	Input	33	GND	Power			
34	TTCReady	Output	34	GND	Power			
35	GND	Power	35	+5V	Power			
36	GND	Power	36	+5V	Power			
37	GND	Power	37	+5V	Power			
38	GND	Power	38	+5V	Power			
39	GND	Power	39	QPLL power	Passive			
40	GND	Power	40	SCL	Input			
41	GND	Power	41	GND	Power			
42	GND	Power	42	GND	Power			
43	GND	Power	43	+5V	Power			
44	GND	Power	44	+5V	Power			
45	GND	Power	45	+5V	Power			
46	GND	Power	46	+5V	Power			
47	GND	Power	47	GND	Power			
48	GND	Power	48	GND	Power			
49	GND	Power	49	GND	Power			
50	GND	Power	50	GND	Power			

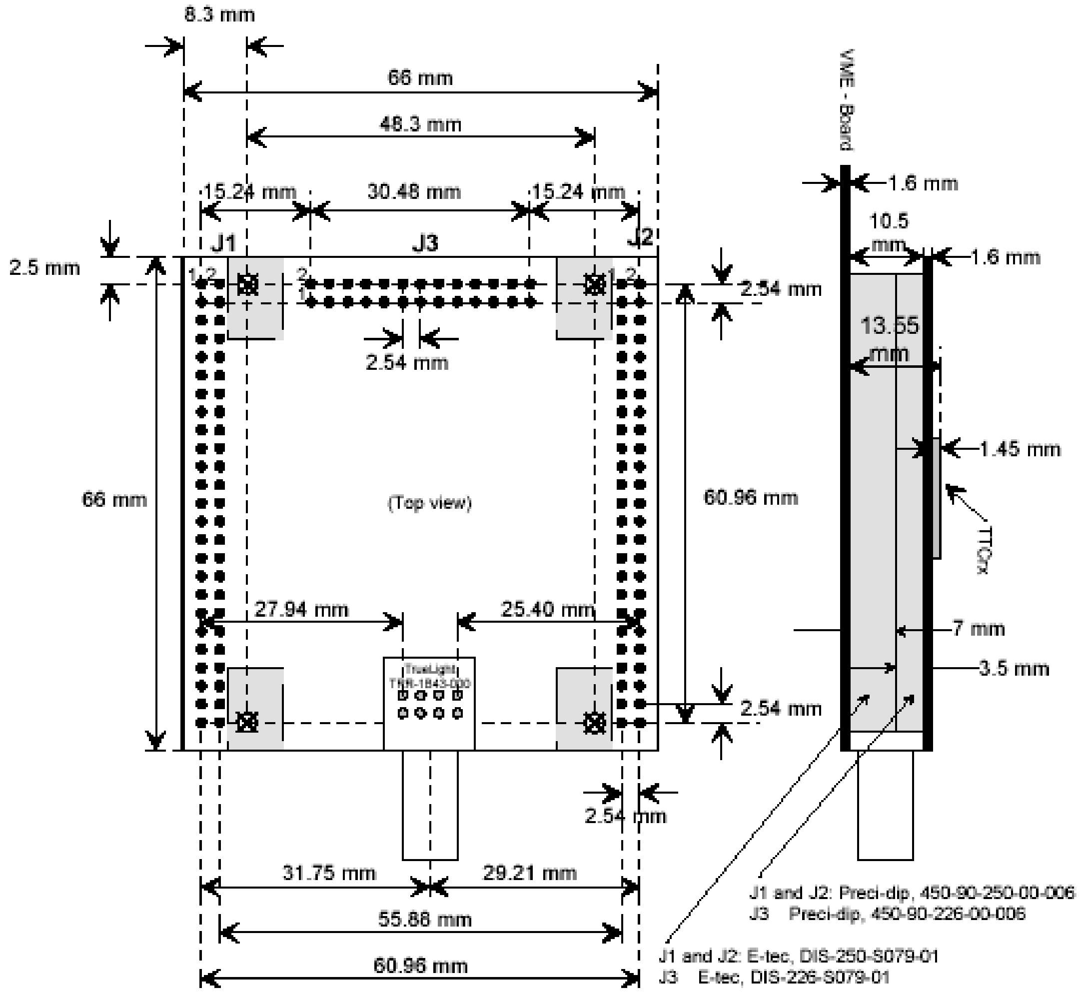

Figure 2: TTCrq layout and dimensions

Each TTCrq mezzanine board has a unique 14-bit number that is defined by on-board jumpers. For more details see chapter 8 of the TTCrx Reference Manual [3]. This number can be accessed via I2C bus (see section 6) or from CSRB18 (see section 4.21). The settings on TTCrq mezzanines installed on production CCB2004 boards are the following: SubAdd[7..0]=1 and Data[7..0]=1..255. As a rule, the code on Data[7..0] lines is equal to serial number of the CCB2004 board (marked on the front panel and on the top side of the main board). The other jumpers on TTCrq mezzanine described in [5] are:

- R53..R55 (termination resistors for 40/80/160Mhz LVDS clock outputs) are not installed (there are 100 Ohm termination resistors R15..R17 installed on CCB2004)
- ST19 is set to "TTCrx" clock source for clock pin 2 on J1 connector
- R59 is not mounted (on-board regulator provides the QPLL power)

2. Custom Backplane Interface

The lists of signals that the CCB2004 distributes to the custom peripheral and Track Finder backplanes are given in Tables 3A and 3B respectively. Pin assignment of the custom peripheral backplane connectors is based on [6] (both on-board connectors are AMP 100145-1 female 125-pin 25-row by 5 pins) and given in Tables 4A and 5A. The CCB2004 may also be used in the CSC Track Finder crate with another custom backplane [7]. In this case the CCB2004 distributes/receives a subset of signals described in Tables 4B and 5B.

Texas Instruments SN74GTLPH16912GR transceivers are used for the communication with other modules in a crate. All GTLP backplane signals are sent in "negative" logic as proposed in [1], i.e. active output signal corresponds to a "low" GTLP level. The same is expected for all incoming backplane signals. 28 point-to-point cfg_done signals (from TMB, DMB and MPC) are terminated on CCB2004 board with 56 Ohm resistors to +1.5V. All bussed signals are terminated on both ends of the backplane.

Table 3A: Peripheral Backplane Interface Signals

Signal	Bits	Source	Destination	Type	Logic	Duration/Level
Clock Bus: Clock Distribution & Bunch Crossing						
ccb_clock40	19	TTCrx, QPLL, osc	All 19 Slots	Point-to-point	LVDS	40.08MHz
Clk80_MPC	1	QPLL, oscillator	MPC	Point-to-point	LVDS	80.16MHz
Fast Control Bus						
ccb_clock40_enable	1	TTCrx, QPLL, oscillator	All 19 Slots	Bussed	GTLP	40.08Mhz
ccb_cmd[5..0]	6	TTCrx	All 19 Slots	Bussed	GTLP	Level
Ccb_ttcrx_ready (former ccb_reserved[0])	1	TTCrx	All 19 slots	Bussed	GTLP	Level
QPLL_locked (former ccb_reserved[1]	1	QPLL ASIC	All 19 Slots	Bussed	GTLP	Level
ccb_evcntres	1	TTCrx	All 19 Slots	Bussed	GTLP	25 ns
ccb_bcntres	1	TTCrx	All 19 Slots	Bussed	GTLP	25 ns
Ccb_L1Reset (former ccb_reserved[4])	1	DTTCrx	All 19 Slots	Bussed	GTLP	25 ns
ccb_cmd_strobe	1	TTCrx	All 19 Slots	Bussed	GTLP	25 ns
ccb_bx0	1	DTTCrx	All 19 Slots	Bussed	GTLP	25 ns
ccb_l1accept	1	TTCrx	All 19 Slots	Bussed	GTLP	25 ns
ccb_data[7..0]	8	TTCrx	All 19 Slots	Bussed	GTLP	Level
ccb_data_strobe	1	TTCrx	All 19 Slots	Bussed	GTLP	25 ns
ccb_reserved[3..2]	2	VME	All 19 Slots	Bussed	GTLP	Level
Total	26					
TMB Reload Bus: ALCT+CLCT+TMB FPGA Reload						
tmb_hard_reset	1	DTTCrx (*)	9 TMB	Bussed	GTLP	400 ns
tmb_cfg_done[8..0]	9	9 TMB	VME	Point-to-Point	GTLP	Level
alct_hard_reset	1	DTTCrx (*)	9 TMB	Bussed	GTLP	400 ns
alct_cfg_done[8..0]	9	9 TMB (from ALCT)	VME	Point-to-Point	GTLP	Level
Tmb_soft_reset (former tmb_reserved[1])	1	VME	9 TMB	Bussed	GTLP	Level
tmb_reserved[0]	1	VME	9 TMB	Bussed	GTLP	Level
Total	22					

MPC Reload Bus: MPC FPGA Reload						
mpc_hard_reset	1	DTTCrx (*)	MPC	Point-to-Point	GTLP	400 ns
mpc_cfg_done	1	MPC	CCB	Point-to-Point	GTLP	Level
Mpc_soft_reset (former mpc_reserved[2])	1	VME	MPC	Point-to-Point	GTLP	Level
mpc_reserved[1..0]	2	VME	MPC	Point-to-Point	GTLP	Level
Total	5					
DMB Reload Bus: DMB FPGA Reload						
dmb_hard_reset	1	DTTCrx (*)	9 DMB	Bussed	GTLP	400 ns
dmb_cfg_done[8..0]	9	9 DMB	CCB	Point-to-Point	GTLP	Level
Dmb_soft_reset (former dmb_reserved[2])	1	VME	9 DMB	Bussed	GTLP	Level
dmb_reserved[1..0]	2	VME	9 DMB	Bussed	GTLP	Level
Total	13					
DAQ Special Purpose Bus [Used by DMB and TMB]						
dmb_cfeb_calibrate[2..0]	3	VME	9 DMB 9TMB	Bussed	GTLP	Level
dmb_l1a_release	1	9 DMB, 9 TMB	VME	Bussed	GTLP	Level
dmb_reserved_out[4..0]	5	VME	9 DMB 9TMB	Bussed	GTLP	Level
dmb_reserved_in[2..0]	3	9 DMB, 9 TMB	VME	Bussed	GTLP	Level
Total	12					
Trigger Special Purpose Bus [Used by TMB only]						
alct_adb_pulse_sync	1	VME	9 TMB	Bussed	GTLP	Level
alct_adb_pulse_async	1	VME	9 TMB	Bussed	GTLP	Level
clct_external_trigger	1	VME	9 TMB	Bussed	GTLP	Level
alct_external_trigger	1	VME	9 TMB	Bussed	GTLP	Level
clct_status[8..0]	9	9 TMB	FP	Bussed	GTLP	Level
alct_status[8..0]	9	9 TMB	FP	Bussed	GTLP	Level
tmb_l1a_request	1	9 TMB	VME	Bussed	GTLP	Level
tmb_l1a_release	1	9 TMB	VME	Bussed	GTLP	Level
tmb_reserved_in[4..0]	5	9 TMB	VME	Bussed	GTLP	Level
tmb_reserved_out[2..0]	3	VME	9 TMB	Bussed	GTLP	Level
Total	32					

(*) – decoded from Brcst[7..2] or Data[7..2]

Table 3B: Track Finder Backplane Interface Signals

Signal	Bits	Source	Destination	Type	Logic	Duration/Level
Clock Bus: Clock Distribution & Bunch Crossing						
ccb_clock40	17	TTCrx, QPLL, oscillator	All 17 Slots	Point-to-point	LVDS	40.08MHz or 80.16Mhz, see section 2.1
Clk80_MPC	3	QPLL, oscillator	3 MPC	Point-to-point	LVDS	80.16MHz
Fast Control Bus						
ccb_clock40_enable	1	TTCrx, QPLL, oscillator	All 17 Slots	Bussed	GTLP	40.08Mhz
ccb_cmd[5..0]	6	TTCrx	All 17 Slots	Bussed	GTLP	Level
Ccb_ttcrx_ready (former ccb_reserved[0])	1	TTCrx	All 17 slots	Bussed	GTLP	Level
QPLL_locked (former ccb_reserved[1]	1	QPLL ASIC	All 17 Slots	Bussed	GTLP	Level
ccb_evcntres	1	TTCrx	All 17 Slots	Bussed	GTLP	25 ns
ccb_bcntres	1	TTCrx	All 17 Slots	Bussed	GTLP	25 ns
Ccb_L1Reset (former ccb_reserved[4])	1	DTTCrx	All 17 Slots	Bussed	GTLP	25 ns
ccb_cmd_strobe	1	TTCrx	All 17 Slots	Bussed	GTLP	25 ns
ccb_bx0	1	DTTCrx	All 17 Slots	Bussed	GTLP	25 ns
ccb_l1accept	1	TTCrx	All 17 Slots	Bussed	GTLP	25 ns
ccb_data[7..0]	8	TTCrx	All 17 Slots	Bussed	GTLP	Level
ccb_data_strobe	1	TTCrx	All 17 Slots	Bussed	GTLP	25 ns
ccb_reserved[3..2]	2	VME	All 17 Slots	Bussed	GTLP	Level
Total	26					
SP, MS, MPC, DDU Reload Bus						
sp_hard_reset	1	DTTCrx (*)	12 SP	Bussed	GTLP	400 ns
sp_cfg_done[12..1]	12	12 SP	VME	Point-to-Point	GTLP	Level
ms_hard_reset	1	DTTCrx (*)	MS	Point-to-Point	GTLP	400 ns
ms_cfg_done	1	MS	VME	Point-to-Point	GTLP	Level
mpc_hard_reset	1	DTTCtx(*)	3 MPC	Bussed	GTLP	400 ns
mpc_cfg_done	3	3 MPC	VME	Point-to-Point	GTLP	Level
ddu_cfg_done	1	DDU	VME	Point-to-Point	GTLP	Level
Total	20					
Special Purpose and Reserved Lines						
sp_l1a_request	1	12 SP	VME	Bussed	GTLP	Pulse
ms_l1a_request	1	MS	VME	Point-to-Point	GTLP	Pulse
ms_to_ccb[1..0]	2	MS	VME	Point-to-Point	GTLP	
sp_to-ccb[2..0]	3	12 SP	VME	Bussed	GTLP	
ccb_to_ms[3..0]	4	CCB	MS	Point-to-Point	GTLP	
Total	11					

(*) – decoded from Brcst[7..2] or Data[7..2]

Table 4A: X40 Backplane Connector, Peripheral Crate

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	CLK+DMB1	B1	CLK-DMB1	C1	GND	D1	CLK+TMB1	E1	CLK-TMB1
A2	CLK+TMB2	B2	CLK-TMB2	C2	GND	D2	CLK+DMB9	E2	CLK-DMB9
A3	CLK+DMB2	B3	CLK-DMB2	C3	GND	D3	CLK+TMB9	E3	CLK-TMB9
A4	CLK+TMB3	B4	CLK-TMB3	C4	GND	D4	CLK+DMB8	E4	CLK-DMB8
A5	CLK+DMB3	B5	CLK-DMB3	C5	GND	D5	CLK+TMB8	E5	CLK-TMB8
A6	CLK+TMB4	B6	CLK-TMB4	C6	GND	D6	CLK+DMB7	E6	CLK-DMB7
A7	CLK+DMB4	B7	CLK-DMB4	C7	GND	D7	CLK+TMB7	E7	CLK-TMB7
A8	CLK+TMB5	B8	CLK-TMB5	C8	GND	D8	CLK+DMB6	E8	CLK-DMB6
A9	CLK+DMB5	B9	CLK-DMB5	C9	GND	D9	CLK+TMB6	E9	CLK-TMB6
A10	CLK+MPC	B10	CLK-MPC	C10	GND	D10	alct_cfg_d_9	E10	dmb_cfg_d_9
A11	tmb_cfg_d_1	B11	alct_cfg_d_1	C11	GND	D11	dmb_cfg_d_8	E11	tmb_cfg_d_9
A12	dmb_cfg_d_1	B12	tmb_cfg_d_2	C12	GND	D12	tmb_cfg_d_8	E12	alct_cfg_d_8
A13	alct_cfg_d_2	B13	dmb_cfg_d_2	C13	GND	D13	alct_cfg_d_7	E13	dmb_cfg_d_7
A14	tmb_cfg_d_3	B14	alct_cfg_d_3	C14	GND	D14	dmb_cfg_d_6	E14	tmb_cfg_d_7
A15	dmb_cfg_d_3	B15	tmb_cfg_d_4	C15	GND	D15	tmb_cfg_d_6	E15	alct_cfg_d_6
A16	alct_cfg_d_4	B16	dmb_cfg_d_4	C16	GND	D16	Mpc_h_reset	E16	Mpc_rsv0
A17	Tmb_cfg_d_5	B17	alct_cfg_d_5	C17	GND	D17	Mpc_rsv1	E17	Mpc_softres
A18	dmb_cfg_d_5	B18		C18	GND	D18	mpc_c_done	E18	
A19	Clock_enable	B19	Ccb_rsv4	C19	GND	D19	Clk80+MPC	E19	Clk80-MPC
A20	Ccb_cmd0	B20	Ccb_cmd1	C20	GND	D20	Ccb_cmd2	E20	Ccb_cmd3
A21	Ccb_cmd4	B21	Ccb_cmd5	C21	GND	D21	Ccb_evcntres	E21	Ccb_bcntres
A22	Ccb_cmd_str	B22	Ccb_bx0	C22	GND	D22	Ccb_l1accept	E22	Ccb_data_st
A23	Ccb_data0	B23	Ccb_data1	C23	GND	D23	Ccb_data2	E23	Ccb_data3
A24	Ccb_data4	B24	Ccb_data5	C24	GND	D24	Ccb_data6	E24	Ccb_data7
A25	Ccb_rsv0	B25	Ccb_rsv1	C25	GND	D25	Ccb_rsv2	E25	Ccb_rsv3

Table 4B: X40 Backplane Connector, Track Finder Crate

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	CLK+DDU	B1	CLK-DDU	C1	GND	D1		E1	
A2	CLK+MPC1	B2	CLK-MPC1	C2	GND	D2		E2	
A3	CLK+MPC2	B3	CLK-MPC2	C3	GND	D3	CLK+SP12	E3	CLK-SP12
A4	CLK+MPC3	B4	CLK-MPC3	C4	GND	D4	CLK+SP11	E4	CLK-SP11
A5	CLK+SP1	B5	CLK-SP1	C5	GND	D5	CLK+SP10	E5	CLK-SP10
A6	CLK+SP2	B6	CLK-SP2	C6	GND	D6	CLK+SP9	E6	CLK-SP9
A7	CLK+SP3	B7	CLK-SP3	C7	GND	D7	CLK+SP8	E7	CLK-SP8
A8	CLK+SP4	B8	CLK-SP4	C8	GND	D8	CLK+SP7	E8	CLK-SP7
A9	CLK+SP5	B9	CLK-SP5	C9	GND	D9	CLK+MS	E9	CLK-MS
A10	CLK+SP6	B10	CLK-SP6	C10	GND	D10		E10	sp_c_don_12
A11		B11		C11	GND	D11		E11	sp_c_don_11
A12		B12		C12	GND	D12		E12	sp_c_don_10
A13	ddu_cfg_done	B13	mpc_c_don_1	C13	GND	D13		E13	sp_c_done_9
A14	mpc_c_done_2	B14	mpc_c_don_3	C14	GND	D14		E14	sp_c_done_8
A15	sp_c_done_1	B15	sp_c_done_2	C15	GND	D15		E15	sp_c_done_7
A16	sp_c_done_3	B16	sp_c_done_4	C16	GND	D16		E16	
A17	sp_c_done_5	B17	sp_c_done_6	C17	GND	D17		E17	
A18		B18		C18	GND	D18	ms_c_done	E18	
A19	Clock_enable	B19	Ccb_rsv4	C19	GND	D19		E19	
A20	Ccb_cmd0	B20	Ccb_cmd1	C20	GND	D20	Ccb_cmd2	E20	Ccb_cmd3
A21	Ccb_cmd4	B21	Ccb_cmd5	C21	GND	D21	Ccb_evcntres	E21	Ccb_bcntres
A22	Ccb_cmd_str	B22	Ccb_bx0	C22	GND	D22	Ccb_l1accept	E22	Ccb_data_st
A23	Ccb_data0	B23	Ccb_data1	C23	GND	D23	Ccb_data2	E23	Ccb_data3
A24	Ccb_data4	B24	Ccb_data5	C24	GND	D24	Ccb_data6	E24	Ccb_data7
A25	Ccb_rsv0	B25	Ccb_rsv1	C25	GND	D25	Ccb_rsv2	E25	Ccb_rsv3

Table 5A: X41 Backplane Connector, Peripheral Crate

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	Tmb_h_reset	B1	Alct_h_reset	C1	GND	D1	tmb_rsv_0	E1	tmb_softres
A2	Alct_adb_p_s	B2	alct_adb_p_a	C2	GND	D2	clct_ext_trig	E2	alct_ext_trig
A3	Clct_status0	B3	Clct_status1	C3	GND	D3	Clct_status2	E3	Clct_status3
A4	Clct_status4	B4	Clct_status5	C4	GND	D4	Clct_status6	E4	Clct_status7
A5	Clct_status8	B5	Alct_status0	C5	GND	D5	alct_status1	E5	alct_status2
A6	Alct_status3	B6	Alct_status4	C6	GND	D6	Alct_status5	E6	Alct_status6
A7	Alct_status7	B7	alct_status8	C7	GND	D7	tmb_l1a_req	E7	tmb_l1a_rel
A8	Tmb_rsv_in0	B8	Tmb_rsv_in1	C8	GND	D8	Tmb_rsv_in2	E8	Tmb_rsv_in3
A9	Tmb_rsv_in4	B9	tmb_rsv_o0	C9	GND	D9	Tmb_rsv_o1	E9	Tmb_rsv_o2
A10	dmb_h_reset	B10	dmb_rsv_0	C10	GND	D10	dmb_rsv_1	E10	dmb_softres
A11	dmb_cfeb_c0	B11	dmb_cfeb_c1	C11	GND	D11	dmb_cfeb_c2	E11	dmb_l1a_rel
A12	dmb_rsv_o0	B12	dmb_rsv_o1	C12	GND	D12	dmb_rsv_o2	E12	dmb_rsv_o3
A13	dmb_rsv_o4	B13	dmb_rsv_in0	C13	GND	D13	dmb_rsv_in1	E13	
A14		B14		C14	+1.5V	D14		E14	
A15		B15		C15	GND	D15		E15	
A16		B16		C16	+1.5V	D16		E16	
A17		B17		C17	GND	D17		E17	
A18		B18		C18	+1.5V	D18		E18	
A19		B19		C19	GND	D19		E19	
A20		B20		C20	+1.5V	D20		E20	
A21		B21		C21	GND	D21		E21	
A22		B22		C22	+1.5V	D22		E22	
A23		B23		C23	GND	D23		E23	
A24		B24		C24	+1.5V	D24		E24	
A25		B25		C25	GND	D25		E25	

Table 5B: X41 Backplane Connector, Track Finder Crate

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	sp_hard_reset	B1	ms_h_reset	C1	GND	D1		E1	
A2		B2		C2	GND	D2		E2	
A3		B3		C3	GND	D3		E3	
A4		B4		C4	GND	D4		E4	
A5		B5		C5	GND	D5		E5	
A6		B6		C6	GND	D6		E6	
A7		B7		C7	GND	D7	sp_l1a_req	E7	ms_l1a_req
A8	ms_to_ccb_0	B8	ms_to_ccb_1	C8	GND	D8		E8	
A9		B9		C9	GND	D9		E9	
A10	mpc_h_reset	B10		C10	GND	D10		E10	
A11		B11		C11	GND	D11		E11	
A12	ccb_to_ms_0	B12	ccb_to_ms_1	C12	GND	D12	ccb_to_ms_2	E12	ccb_to_ms_3
A13		B13	sp_to_ccb_0	C13	GND	D13	sp_to_ccb_1	E13	sp_to_ccb_2
A14		B14		C14	+1.5V	D14	Clk80+Mpc1	E14	Clk80-Mpc1
A15		B15		C15	GND	D15	Clk80+Mpc2	E15	Clk80-Mpc2
A16		B16		C16	+1.5V	D16	Clk80+Mpc3	E16	Clk80-Mpc3
A17		B17		C17	GND	D17		E17	
A18		B18		C18	+1.5V	D18		E18	
A19		B19		C19	GND	D19		E19	
A20		B20		C20	+1.5V	D20		E20	
A21		B21		C21	GND	D21		E21	
A22		B22		C22	+1.5V	D22		E22	
A23		B23		C23	GND	D23		E23	
A24		B24		C24	+1.5V	D24		E24	
A25		B25		C25	GND	D25		E25	

3.1. Clock Distribution

There are four sources of the 40.078Mhz clock on a CCB2004 board: Clock40Des1 from the TTCrx ASIC (1), CMOS40Mhz output from the QPLL ASIC (2), QPLL_40Mhz after LVDS-to-LVTTL conversion (3) and Clock40Osc (4) that is derived from an on-board oscillator (80.157Mhz divided by 2). One source must be selected using switch S5.

There are two sources of the 80.157Mhz: Clock80Mhz from the QPLL ASIC and Clock80Osc from an on-board oscillator. One source must be selected using switch S7.

All modules in the peripheral crate require a point-to-point 40MHz LVDS clock from the CCB2004. Some modules in the Track Finder crate may require either 40Mhz or 80Mhz clocks over the same lines from the CCB'2004. In order to allow the CCB2004 design to be compatible with both peripheral and Track Finder crates the CCB2004 has a switch that allows to chose 40/80Mhz source for a several slots (see Table 6).

Table 6: Clocking Options in the Peripheral and Track Finder Crates

Slot	Peripheral Crate	Clock Source	Track Finder Crate	Clock Source
1	Crate Master	-	Crate Master	-
2	TMB1	40Mhz	DDU	40Mhz + 80Mhz
3	DMB1	40Mhz	MPC1	40Mhz + 80Mhz
4	TMB2	40Mhz	MPC2	40Mhz + 80Mhz
5	DMB2	40Mhz	MPC3	40Mhz
6	TMB3	40Mhz	SP1	40Mhz or 80Mhz
7	DMB3	40Mhz	SP2	40Mhz or 80Mhz
8	TMB4	40Mhz	SP3	40Mhz or 80Mhz
9	DMB4	40Mhz	SP4	40Mhz or 80Mhz
10	TMB5	40Mhz	SP5	40Mhz or 80Mhz
11	DMB5	40Mhz	SP6	40Mhz or 80Mhz
12	MPC	40Mhz + 80Mhz	CCB	TTC/internal
13	CCB	internal	-	-
14	TMB6	40Mhz	Muon Sorter	40Mhz
15	DMB6	40Mhz	-	-
16	TMB7	40Mhz	SP7	40Mhz or 80Mhz
17	DMB7	40Mhz	SP8	40Mhz or 80Mhz
18	TMB8	40Mhz	SP9	40Mhz or 80Mhz
19	DMB8	40Mhz	SP10	40Mhz or 80Mhz
20	TMB9	40Mhz	SP11	40Mhz or 80Mhz
21	DMB9	40Mhz	SP12	40Mhz or 80Mhz

All clocks are distributed over the peripheral backplane using individual point-to-point LVDS lines. There are fine and coarse delays for the 40Mhz clock and command signals incorporated into TTCrx ASIC. They are programmable via the I2C interface or optical path. In addition to individual clock per each slot there is a common GTLP line (formerly assigned for ccb_clock_enable signal) that carries a continuous 40.078Mhz signal selected from one of four sources described above. This signal can be used by Sector Processors in conjunction with the 80Mhz clock.

A dedicated 80Mhz LVDS clock in addition to 40Mhz clock is transmitted to MPC slot in the peripheral crate and to every (out of 3) MPC slot in the Track Finder crate.

3.2. Fast Control and Data Buses

The TTCrx ASIC can distribute broadcast commands over Brcst[7..2] bus and individually addressed commands over Data[7..0] and Subadd[7..0] buses. Six backplane lines ccb_cmd[5..0] and ccb_cmd_strobe lines represent the Brcst[7..2] and BrcstStr1 outputs of the TTCrx respectively. The ccb_data[7..0] and ccb_data_strobe lines represent the data[7..0] and data_strobe outputs of the TTCrx respectively. Peripheral EMU electronics may need to decode both broadcast (all crates in a system) and individual (only one crate in a system) commands. The same decoding scheme (Table 7) is being used on both fast control and data buses. Subadd[7..0] lines are not provided the peripheral backplane. Note that after power up several outputs of TTCrx, including Dout[7..0], DQ[3..0], SubAdd[7..0] and DoutStr are disabled (chapter 3 of [3]). In order to enable them, bit 5 in the control register of the TTCrx should be set “1”.

According to [3], both Clk40Des1 and Clk40Des2 can be used for the synchonization of BrcstStr2 and Brcst[7..6]. The BrcstStr1 and Brcst[5..2] are always synchronized with the Clk40des1. Also there are separate settings for coarse delays for BrcsStr1 and BrcstStr2. So, in order to make a broadcast scheme work in the CCB’2004, the coarse settings for BrcstStr1 and BrcstStr2 should be the same and BrcsStr2 and Brcst[7..6] should be synchronized with the Clk40Des1. This can be done by control access to the TTCrx ASIC. This is a default state after power up and reset.

Several Hard_Reset commands (from either Brcst[7..2] or Data[7..2] busses) are decoded inside the CCB2004 (they are marked red in Table 7). All other commands must be decoded by targeted peripheral boards from the ccb_cmd[5..0]+ccb_cmd_strobe AND ccb_data[7..2]+ccb_data_strobe lines.

Table 7: Fast Control Bus Ccb_cmd[5..0] and Ccb_data[7..2] (*) Commands

Signal	Code (hex)	Description
BC0	01	Bunch Crossing Zero
OC0	02	Orbit Counter Reset
L1 Reset	03	Reset L1 readout buffers
Hard_reset (**)	04	Reload all FPGAs from EPROMs
Start Trigger	06	
Stop Trigger	07	
Test Enable	08	
Private Gap	09	
Private Orbit	0A	
CCB_hard_reset (***)	0F	Reload Xilinx FPGA from its EPROM on CCB’2004 board
Tmb_hard_reset (**)	10	Reload TMB FPGA’s from EPROM
Alct_hard_reset (**)	11	Reload ALCT FPGA’s from EPROM
Dmb_hard_reset (**)	12	Reload DMB FPGA’s from EPROM
Mpc_hard_reset (**)	13	Reload MPC FPGA’s from EPROM
Dmb_cfeb_calibrate0	14	CFEB Calibrate Pre-Amp Gain
Dmb_cfeb_calibrate1	15	CFEB Trigger Pattern Calibration

Dmb_cfeb_calibrate2	16	CFEB Pedestal Calibration
Dmb_cfeb_initiate	17	Initiate CFEB calibration (Hold next L1ACC and Pretriggers)
Alct_adb_pulse_sync	18	Pulse Anode Discriminator, synchronous
Alct_adb_pulse_async	19	Pulse Anode Discriminator, asynchronous
Clct_external_trigger	1A	External Trigger All CLCTs
Alct_external_trigger	1B	External Trigger All ALCTs
Soft_reset	1C	Initializes the FPGA on DMB, TMB and MPC boards
DMB_soft_reset	1D	Initializes the FPGA on a DMB
TMB_soft_reset	1E	Initializes the FPGA on a TMB
MPC_soft_reset	1F	Initializes the FPGA on a MPC
Inject patterns from TMBs	24	Inject test patterns from the TMB's internal RAM to MPC
Alct_adb_pulse	25	Generate both synchronous and asynchronous anode discriminator pulses
Inject patterns from SP	2F	Inject test patterns from the SP's internal FIFO to MS
Inject patterns from MPC	30	Inject test patterns from the MPC's internal FIFO to SP
Inject patterns from MS	31	Inject test patterns from the MS's internal FIFO to GMT
Bunch Counter Reset	32	Resets bunch counters

(*) – data[1..0] can be any values

(**) – decoded by CCB

(***) –decoded only by discrete logic interface of the CCB from the TTCrq outputs

3.3. L1ACC and Pretrigger Sources and Control

There are ten possible sources for L1ACC and two Pretriggers (ALCT_external_trigger and CLCT_external_trigger) in "FPGA" mode:

- TTCrx_L1ACC (25 ns pulse),
- External_L1ACC (LVDS input coming through the front panel),
- Tmb_l1a_request (25 ns pulse coming over custom backplane),
- Tmb_l1a_release (25 ns pulse coming over custom backplane)
- Artificial L1ACC generated upon VME write to (base + 54) address (see Table 8),
- Any source of ALCT_adb_pulse_sync (VME, TTC command),
- Any source of ALCT_adb_pulse_async (VME, TTC command)
- DMB_cfeb_calibrate[2..0] pulses (VME, TTC command); logical "OR" of three pulses.

Each of these sources can be masked using dedicated CSRB1 bits (disabled if "1" and enabled if "0"). Before distribution to backplane both Pretriggers can be delayed for a number of 40Mhz clocks (from 1 to 255, defined by CSRB5[15..8]). Independently from Pretriggers, the L1ACC can also be delayed for n=1..255 clock cycles where (n) is defined by CSRB5[7..0]. The ALCT_adb_pulse_sync and ALCT_adb_pulse_async backplane signals are sent to backplane without delay. These sources may produce L1ACC to backplane when CSRA1[0]=0 independently from CSRB1[0]. In "discrete logic" mode, when the CSRA1[0]=1, only L1ACC from TTCrx may unconditionally propagate to custom backplane with a delay of ~50 ns.

There is a 32-bit counter of L1ACC requests. The counter counts the L1ACC requests from any selected source in the FPGA even if the transmission to backplane is disabled. The counter is disabled after power up and the CCB2004 internal reset.

In a special CFEB calibration mode, the CCB2004 may receive a dmb_cfeb_initiate command from the TTCrx (this command can be also send upon write to a specific VME address, see Table 8). In this mode, the CCB2004 “holds” the next L1ACC and both Pretriggers (by preventing them from being broadcast on the backplane). In a similar way, if the CSRB1[13]=1, then the CCB2004 holds the next L1ACC and Pretriggers after the first L1ACC has been sent to backplane. The CCB’2004 can exit from this “hold” mode when a dmb_l1a_release or tmb_l1a_release signals have been received from DMB or TMB or when VME write to (base address + 58) has been performed.

3.4. Reloading Signals

The Xilinx FPGA residing on CCB2004 board can be reloaded from its EPROM on common and CCB-specific “Hard_reset” commands (Table 7) if switch S10-8 is “on” (see Section 7). Reconfiguration time is ~60 ms. Note that when common “Hard_Reset” or “CCB_Hard_reset” commands are decoded inside the FPGA, they in fact do not force reconfiguration of the FPGA itself. These commands are effective for FPGA reconfiguration only after decoding by discrete logic.

Hard_reset reloading signals (from either Brcst[7..2] or Data[7..2] busses) are decoded by the CCB2004 and expanded to 500 ns before distribution to the backplane. There is also a common Hard_reset command (Table 7), that causes generation of all reloading signals to all boards in a crate simultaneously.

An ALCT, TMB, DMB and MPC cards inform the CCB2004 when they have completed the reconfiguration process by asserting their cfg_done signals. These static signals are available for reading from CSRA2-3 (see 3.2-3.3).

4. VME Interface and Control and Status Registers (CSR)

The CCB2004 can be accessed in the VME crate using geographical addressing that utilizes the address pins GA<4-0> available on the VME64x backplane. In this mode the CCB2004 recognizes its address space when the code on address lines A<23-19> is equal to the 5-bit geographical code of its slot. The base address is 600000(hex) in the Track Finder crate (slot 12) and 680000(hex) in the peripheral crate (slot 13). The board recognizes an AM codes 39(hex), 3D(hex) and supports A24D16 slave operations. The CCB2004 does not respond to byte-addressing modes, so all valid addresses must be even numbers. The list of available VME addresses is given in Table 8.

There are two groups of Control and Status Registers (CSR): CSRA and CSRB. The group A is implemented in a discrete logic and the group B in Xilinx FPGA.

Table 8

Address (hex)	Function
Group A, discrete logic	
Base + 0	CSRA1 (control/status register), discrete logic
Base + 2	CSRA2 (status register), discrete logic. Write to CSRA2 causes generation of

	Hard_reset for the FPGA. See 4.2. for more details.
Base + 4	CSRA3 (status register), discrete logic. Write to CSRA3 causes generation of Soft_reset for the FPGA See 4.3. for more details.
Group B, FPGA	
Base + 20	CSRB1 (control register), FPGA
Base + 22	CSRB2 (command bus), FPGA
Base + 24	CSRB3 (data bus), FPGA
Base + 26	CSRB4 (general purpose R/W register, future use), FPGA
Base + 28	CSRB5 (delay register), FPGA
Base + 2a	CSRB6 (control register), FPGA
Base + 2c	CSRB7 (QPLL control register), FPGA
Base + 2e	CSRB8 (general purpose R/W register, future use), FPGA
Base + 30	CSRB9 (status register, serial ID chip), FPGA
Base + 32	CSRB10, not implemented, FPGA
Base + 34	CSRB11 (status register), FPGA
Base + 36	CSRB12 (status register), FPGA
Base + 38	CSRB13 (status register), FPGA
Base + 3a	CSRB14 (status register), FPGA
Base + 3c	CSRB15 (status register), FPGA
Base + 3e	CSRB16 (status register), FPGA
Base + 40	CSRB17 (status register, date of firmware revision), FPGA
Base + 42	CSRB18 (TTCrx hardwired ID number). Read only. Valid after TTCrx reset. CSRB18[7..0] = Data[7..0] and CSRB18[15..8] = SubAdr[7..0]
Base + 44	CSRB19 (16-bit counter of BRCSTSTR1 strobes from the TTCrx; write to this address resets the counter)
Base + 46	CSRB20 (16-bit counter of BRCSTSTR2 strobes from the TTCrx; write to this address resets the counter)
Base + 48	CSRB21 (16-bit counter of DOUTSTR strobes from the TTCrx; write to this address resets the counter)
Base + 4a	
Base + 4c	
Base + 4e	
Base + 50	Generate L1Reset 25 ns pulse (write only), FPGA (*)
Base + 52	Generate BC0 25 ns pulse (write only), FPGA (*)
Base + 54	Generate L1ACC 25 ns pulse (write only) and external triggers (if enabled), FPGA (*)
Base + 56	Generate "DMB_cfeb_initiate" pulse (write only) to hold the next L1ACC and Pretriggers, FPGA
Base + 58	Generate "Release HOLD L1ACC Mode" pulse to enable transmission of the L1ACC and Pretriggers (write only), FPGA
Base + 5a	Reset CSRB11 Error Bits (write only), FPGA
Base + 5c	Reset TTCrx ASIC (write only). Generates 50 us pulse of negative polarity to TTCrx.
Base + 5e	
Base + 60	Generate "Hard_Reset" 500 ns pulse to all modules in the crate (write only), FPGA (*)
Base + 62	Generate "TMB_Hard_Reset" 500 ns pulse to all TMB boards in the crate (write only), FPGA (*)
Base + 64	Generate "DMB_Hard_Reset" 500 ns pulse to all DMB boards in the crate (write only), FPGA (*)
Base + 66	Generate "ALCT_Hard_Reset" 500 ns pulse to all ALCT boards (write only), FPGA (*)
Base + 68	Generate "MPC_Hard_Reset" 500 ns pulse to MPC (write only), FPGA (*)
Base + 6a	Generate "Soft_Reset" 25 ns pulse to TMB, DMB, MPC boards in the crate (write only), FPGA (*)
Base + 6c	Generate "TMB_Soft_Reset" 25 ns pulse to TMB boards (write only), FPGA (*)
Base + 6e	Generate "DMB_Soft_Reset" 25 ns pulse to TMB boards (write only), FPGA (*)

Base + 70	Generate “MPC_Soft_Reset” 25 ns pulse to TMB boards (write only), FPGA (*)
Base + 72	
Base + 74	
Base + 76	
Base + 78	
Base + 7a	
Base + 7c	
Base + 7e	
Base + 80	Generate both “ALCT_adb_pulse_sync” and “ALCT_adb_pulse_async” (write only), FPGA (*)
Base + 82	Generate “ALCT_adb_pulse_sync” 500 ns pulse (write only), FPGA (*)
Base + 84	Generate “ALCT_adb_pulse_async” pulse (write only), FPGA (*)
Base + 86	Generate “CLCT_external_trigger” 25 ns pulse (write only), FPGA (*)
Base + 88	Generate “ALCT_external_trigger” 25 ns pulse (write only), FPGA (*)
Base + 8a	Generate “DMB_cfeb_calibrate[0]” 25 ns pulse (write only), FPGA (*)
Base + 8c	Generate “DMB_cfeb_calibrate[1]” 25 ns pulse (write only), FPGA (*)
Base + 8e	Generate “DMB_cfeb_calibrate[2]” 25 ns pulse (write only), FPGA (*)
Base + 90	Read L1ACC Counter[15..0] (read only), FPGA
Base + 92	Read L1ACC Counter[31..16] (read only), FPGA
Base + 94	Reset L1ACC Counter to 0 (write only), FPGA
Base + 96	Enable L1ACC Counter to count (write only), FPGA
Base + 98	Disable L1ACC Counter to count (write only), FPGA
Base + 9a	Generate 800 us “Reset pulse” on 1-Wire bus to initialize the serial ID chip (write only), FPGA
Base + 9c	Generate 3 us “Read pulse” on 1-Wire bus to read data from the serial ID chip (write only), FPGA
Base + 9e	Reset CSRB9 (write only), FPGA
Base + a0	Generate “Write-zero” 50 us pulse on 1-Wire bus to send a command to serial ID chip (write only), FPGA
Base + a2	Generate “Write-one” 12 us pulse on 1-Wire bus to send a command to serial ID chip (write only), FPGA

(*) These optional write commands cause generation of pulses of specific length onto dedicated backplane lines. Ccb_cmd[5..0], ccb_cmd_strobe, ccb_data[7..0], ccb_data_strobe lines remain inactive during these commands. These commands are independent from CSRB1[0].

4.1. CSRA1 (base + 00)

Bit	Access	Function
0	R/W	Source of signals to custom backplane (FPGA if “0” and discrete logic interface if “1”).
1	R/W	Enable Read operations over I2C bus from TTCrx ASIC (active “0”)
2	R/W	I2C_SDA line (active “0”)
3	R/W	I2C_SCL line (active “0”)
4	R	I2C_SDA line (read only)
5	R/W	FPGA_TDI pin of the JTAG interface to FPGA/EPROM
6	R/W	FPGA_TMS pin of the JTAG interface to FPGA/EPROM
7	R/W	FPGA_TCK pin of the JTAG interface to FPGA/EPROM
8	R	FPGA_TDO pin of the JTAG interface from FPGA/EPROM (read only)
9	R/W	-
10	R/W	-
11	R/W	-
12	R/W	-
13	R/W	-
14	R/W	-
15	R/W	-

4.2. CSRA2 (base + 02)

Write operation to CSRA2 (any data) causes unconditional "Hard Reset" of the FPGA. Data format for read operations is shown in a table below.

Bit	Access	Function
0	R	Mpc_cfg_done (peripheral crate) (active "0") or MS_cfg_done (TF crate) (active "0")
1	R	Alct_cfg_done_1 (peripheral crate) (active "0")
2	R	Alct_cfg_done_2 (peripheral crate) (active "0")
3	R	Alct_cfg_done_3 (peripheral crate) (active "0")
4	R	Alct_cfg_done_4 (peripheral crate) (active "0") or SP_cfg_done_3 (TF crate) (active "0")
5	R	Alct_cfg_done_5 (peripheral crate) (active "0") or SP_cfg_done_6 (TF crate) (active "0")
6	R	Alct_cfg_done_6 (peripheral crate) (active "0") or SP_cfg_done_7 (TF crate) (active "0")
7	R	Alct_cfg_done_7 (peripheral crate) (active "0")
8	R	Alct_cfg_done_8 (peripheral crate) (active "0") or SP_cfg_done_10 (TF crate) (active "0")
9	R	Alct_cfg_done_9 (peripheral crate) (active "0")
10	R	Tmb_cfg_done_1 (peripheral crate) (active "0")
11	R	Tmb_cfg_done_2 (peripheral crate) (active "0")
12	R	Tmb_cfg_done_3 (peripheral crate) (active "0")
13	R	Tmb_cfg_done_4 (peripheral crate) (active "0") or SP_cfg_done_2 (TF crate) (active "0")
14	R	Tmb_cfg_done_5 (peripheral crate) (active "0") or SP_cfg_done_5 (TF crate) (active "0")
15	R	Tmb_cfg_done_6 (peripheral crate) (active "0")

4.3. CSRA3 (base + 04)

Write operation to CSRA3 (any data) causes "Soft Reset" of the FPGA logic, including reset of CSRB11[10..8], reset of L1ACC counter, disable L1ACC counter, enable unconditional propagation of L1ACC from selected source(s) to custom backplane. Data format for read operations is shown in a table below.

Bit	Access	Function
0	R	Tmb_cfg_done_7 (peripheral crate) (active "0") or SP_cfg_done_8 (TF crate) (active "0")
1	R	Tmb_cfg_done_8 (peripheral crate) (active "0")
2	R	Tmb_cfg_done_9 (peripheral crate) (active "0") or SP_cfg_done_11 (TF crate) (active "0")
3	R	Dmb_cfg_done_1 (peripheral crate) (active "1")
4	R	Dmb_cfg_done_2 (peripheral crate) (active "1")
5	R	Dmb_cfg_done_3 (peripheral crate) (active "1") or SP_cfg_done_1 (TF crate) (active "0")
6	R	Dmb_cfg_done_4 (peripheral crate) (active "1") or SP_cfg_done_4 (TF crate) (active "0")
7	R	Dmb_cfg_done_5 (peripheral crate) (active "1")
8	R	Dmb_cfg_done_6 (peripheral crate) (active "1")
9	R	Dmb_cfg_done_7 (peripheral crate) (active "1") or SP_cfg_done_9 (TF crate) (active "0")
10	R	Dmb_cfg_done_8 (peripheral crate) (active "1")
11	R	Dmb_cfg_done_9 (peripheral crate) (active "1") or SP_cfg_done_12 (TF crate) (active "0")
12	R	CCB'2004 FPGA configuration from EPROM done successfully (active"1")
13	R	TTCrx_Ready status line (active "1" when "Ready")
14	R	QPLL_lock output of the QPLL ASIC (active "0" when "Locked")
15	R	All connected peripheral or TF boards (including CCB'2004 itself) have been configured from their EPROM's successfully (active "0)

4.4. CSRB1 (base + 20)

Bit	Access	Function
0	R/W	Defines the source of ccb_cmd[5..0], ccb_cmd_strobe, ccb_data[7..0], ccb_data_stobe signals distibuted to backplane when CSRA1[0]=0. The source is TTCrx if « 0 » and CSRB2 (for command bus) and CSRB3 (for data bus) if « 1 »
1	R/W	-
2	R/W	Mask L1ACC source from logical « OR » of three DMB_cfeb_calibrate[2..0] sources (disabled if « 1 » and enabled if « 0 »)
3	R/W	Mask L1ACC source from TTCrx (disabled if « 1 » and enabled if « 0 »)
4	R/W	Mask L1ACC from VME command (disabled if « 1 » and enabled if « 0 »)
5	R/W	Mask L1ACC from TMB_L1A_REQ backplane line (disabled if « 1 » and enabled if « 0 »)
6	R/W	Mask L1ACC from TMB_L1A_REL backplane line (disabled if « 1 » and enabled if « 0 »)
7	R/W	Mask L1ACC from the front panel (disabled if « 1 » and enabled if « 0 »)
8	R/W	Enable (when « 1 ») or Disable (when « 0 ») all inputs from the front panel
9	R/W	Mask generation of delayed ALCT_external_trigger signal from any L1A source (disabled if « 1 » and enabled if « 0 »)
10	R/W	Mask generation of delayed CLCT_external_trigger signal from any L1A source (disabled if « 1 » and enabled if « 0 »)
11	R/W	Mask generation of delayed Pretriggers and delayed L1ACC from any of ALCT_adb_sync_pulse sources (disabled if « 1 » and enabled if « 0 »)
12	R/W	Mask generation of delayed Pretriggers and delayed L1ACC from any of ALCT_adb_async_pulse sources (disabled if « 1 » and enabled if « 0 »)
13	R/W	If « 1 », then the CCB disables sending delayed L1ACC and both delayed Pretriggers to backplane after the first L1ACC has been translated to backplane. If « 0 », then L1ACC and Pretriggers are enabled unconditionally
14	R/W	When « 0 », the Tmb_l1A_Release signal enables propagation of delayed L1ACC and both delayed Pretriggers to custom backplane
15	R/W	When « 0 », the Dmb_l1A_Release signal enables propagation of delayed L1ACC and both delayed Pretriggers to custom backplane

4.5. CSRB2 (base + 22)

Bit	Access	Function
0	R/W	BcntRes *
1	R/W	EvCntRes *
2	R/W	Data transmitted to ccb_cmd[0] when CSRA1[0]=0 and CSRB1[0]=1 *
3	R/W	Data transmitted to ccb_cmd[1] when CSRA1[0]=0 and CSRB1[0]=1 *
4	R/W	Data transmitted to ccb_cmd[2] when CSRA1[0]=0 and CSRB1[0]=1 *
5	R/W	Data transmitted to ccb_cmd[3] when CSRA1[0]=0 and CSRB1[0]=1 *
6	R/W	Data transmitted to ccb_cmd[4] when CSRA1[0]=0 and CSRB1[0]=1 *
7	R/W	Data transmitted to ccb_cmd[5] when CSRA1[0]=0 and CSRB1[0]=1 *
8	R/W	-
9	R/W	-
10	R/W	-
11	R/W	-
12	R/W	-
13	R/W	-
14	R/W	-
15	R/W	-

* Ccb_cmd_strobe (25 ns pulse) is generated upon write data into CSRB2

4.6. CSRB3 (base + 24)

Bit	Access	Function
0	R/W	Data transmitted to ccb_data[0] when CSRA1[0]=0 and CSRB1[0]=1 *
1	R/W	Data transmitted to ccb_data[1] when CSRA1[0]=0 and CSRB1[0]=1 *
2	R/W	Data transmitted to ccb_data[2] when CSRA1[0]=0 and CSRB1[0]=1 *
3	R/W	Data transmitted to ccb_data[3] when CSRA1[0]=0 and CSRB1[0]=1 *
4	R/W	Data transmitted to ccb_data[4] when CSRA1[0]=0 and CSRB1[0]=1 *
5	R/W	Data transmitted to ccb_data[5] when CSRA1[0]=0 and CSRB1[0]=1 *
6	R/W	Data transmitted to ccb_data[6] when CSRA1[0]=0 and CSRB1[0]=1 *
7	R/W	Data transmitted to ccb_data[8] when CSRA1[0]=0 and CSRB1[0]=1 *
8	R/W	-
9	R/W	-
10	R/W	-
11	R/W	-
12	R/W	-
13	R/W	-
14	R/W	-
15	R/W	-

* Ccb_data_strobe (25 ns pulse) is generated upon write data into CSRB3

4.7. CSRB4 (base + 26)

Bit	Access	Function
0	R/W	-
1	R/W	-
2	R/W	-
3	R/W	-
4	R/W	-
5	R/W	-
6	R/W	-
7	R/W	-
8	R/W	-
9	R/W	-
10	R/W	-
11	R/W	-
12	R/W	-
13	R/W	-
14	R/W	-
15	R/W	-

4.8. CSRB5 (base + 28)

Bit	Access	Function
0	R/W	Delay of L1A (any of 7 sources) before distribution to backplane, LSB (*)
1	R/W	Delay of L1A (any of 7 sources) before distribution to backplane (*)
2	R/W	Delay of L1A (any of 7 sources) before distribution to backplane (*)
3	R/W	Delay of L1A (any of 7 sources) before distribution to backplane (*)
4	R/W	Delay of L1A (any of 7 sources) before distribution to backplane (*)
5	R/W	Delay of L1A (any of 7 sources) before distribution to backplane (*)
6	R/W	Delay of L1A (any of 7 sources) before distribution to backplane (*)
7	R/W	Delay of L1A (any of 7 sources) before distribution to backplane, MSB (*)
8	R/W	Delay of ALCT and CLCT external triggers before distribution to backplane, LSB (*)
9	R/W	Delay of ALCT and CLCT external triggers before distribution to backplane (*)
10	R/W	Delay of ALCT and CLCT external triggers before distribution to backplane (*)

11	R/W	Delay of ALCT and CLCT external triggers before distribution to backplane (*)
12	R/W	Delay of ALCT and CLCT external triggers before distribution to backplane (*)
13	R/W	Delay of ALCT and CLCT external triggers before distribution to backplane (*)
14	R/W	Delay of ALCT and CLCT external triggers before distribution to backplane (*)
15	R/W	Delay of ALCT and CLCT external triggers before distribution to backplane, MSB (*)

(*) Delay step = 1BX (25 ns)

4.9. CSRB6 (base + 2a)

Bit	Access	Function
0	R/W	CCB_reserved[2] line (active « 1 »)
1	R/W	CCB_reserved[3] line (active « 1 »)
2	R/W	TMB_reserved[0] line (active « 1 »)
3	R/W	MPC_reserved[0] line (active « 1 »)
4	R/W	MPC_reserved[1] line (active « 1 »)
5	R/W	DMB_reserved[0] line (active « 1 »)
6	R/W	DMB_reserved[1] line (active « 1 »)
7	R/W	TMB_reserved_out[0] line (active « 1 »)
8	R/W	TMB_reserved_out[1] line (active « 1 »)
9	R/W	TMB_reserved_out[2] line (active « 1 »)
10	R/W	DMB_reserved_out[0] line (active « 1 »)
11	R/W	DMB_reserved_out[1] line (active « 1 »)
12	R/W	DMB_reserved_out[2] line (active « 1 »)
13	R/W	DMB_reserved_out[3] line (active « 1 »)
14	R/W	DMB_reserved_out[4] line (active « 1 »)
15	R/W	-

4.10. CSRB7 (base + 2c)

Bit	Access	Function
0	R/W	QPLL_mode line
1	R/W	QPLL_reset line
2	R/W	QPLL_restart line
3	R/W	QPLL_excnt line
4	R/W	QPLL_fsel0 line
5	R/W	QPLL_fsel1 line
6	R/W	QPLL_fsel2 line
7	R/W	QPLL_fsel3 line
8	R/W	-
9	R/W	-
10	R/W	-
11	R/W	-
12	R/W	-
13	R/W	-
14	R/W	-
15	R/W	-

4.11. CSRB8 (base + 2e)

Bit	Access	Function
0	R/W	-
1	R/W	-
2	R/W	-
3	R/W	-
4	R/W	-

5	R/W	-
6	R/W	-
7	R/W	-
8	R/W	-
9	R/W	-
10	R/W	-
11	R/W	-
12	R/W	-
13	R/W	-
14	R/W	-
15	R/W	-

4.12. CSRB9 (base + 30)

Bit	Access	Function
0	R	« 0 » indicates the « Presence pulse » from serial ID after the « Reset pulse »
1	R	Data bit from serial ID chip
2	R	Status of the initialization. When « 1 » after the « Reset pulse » , the CSRB9[0] is valid
3	R	Status of the read cycle. When « 1 » after the « Read pulse », the CSRB9[1] is valid
4	R	Status of the command cycle. When « 1 » after the « Write-one » or « Write-zero » command, the next command can be sent
5	R	« 0 »
6	R	« 0 »
7	R	« 0 »
8	R	« 0 »
9	R	« 0 »
10	R	« 0 »
11	R	« 0 »
12	R	« 0 »
13	R	« 0 »
14	R	« 0 »
15	R	« 0 »

4.13. CSRB10 (base + 32)

Bit	Access	Function
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

4.14. CSRB11 (base + 34)

Bit	Access	Function
0	R	DMB_reserved_in[0] line (« 1 » if backplane line is active)
1	R	DMB_reserved_in[1] line (« 1 » if backplane line is active)
2	R	DMB_reserved_in[2] line (« 1 » if backplane line is active)
3	R	TMB_reserved_in[0] line (« 1 » if backplane line is active)
4	R	TMB_reserved_in[1] line (« 1 » if backplane line is active)
5	R	TMB_reserved_in[2] line (« 1 » if backplane line is active)
6	R	TMB_reserved_in[3] line (« 1 » if backplane line is active)
7	R	TMB_reserved_in[4] line (« 1 » if backplane line is active)
8	R	TTC_SINERRSTR line (*)
9	R	TTC_DBERRSTR line (*)
10	R	QPLL_error line (*)
11	R	QPLL_lock line (active « 1 » when locked)
12	R	TTCrx_ready line (active « 1 » when ready)
13	R	« 0 »
14	R	« 0 »
15	R	« 0 »

(*) – any error from these sources will set these bits "1". They can be reset by sending "Reset FPGA logic" or "Reset CSRB11 error bits" commands.

4.15. CSRB12 (base + 36)

Bit	Access	Function
0	R	TTC_BCNT[0] line, latched on TTC_BCNTSTR strobe
1	R	TTC_BCNT[1] line, latched on TTC_BCNTSTR strobe
2	R	TTC_BCNT[2] line, latched on TTC_BCNTSTR strobe
3	R	TTC_BCNT[3] line, latched on TTC_BCNTSTR strobe
4	R	TTC_BCNT[4] line, latched on TTC_BCNTSTR strobe
5	R	TTC_BCNT[5] line, latched on TTC_BCNTSTR strobe
6	R	TTC_BCNT[6] line, latched on TTC_BCNTSTR strobe
7	R	TTC_BCNT[7] line, latched on TTC_BCNTSTR strobe
8	R	TTC_BCNT[8] line, latched on TTC_BCNTSTR strobe
9	R	TTC_BCNT[9] line, latched on TTC_BCNTSTR strobe
10	R	TTC_BCNT[10] line, latched on TTC_BCNTSTR strobe
11	R	TTC_BCNT[11] line, latched on TTC_BCNTSTR strobe
12	R	-
13	R	-
14	R	-
15	R	-

4.16. CSRB13 (base + 38)

Bit	Access	Function
0	R	TTC_BCNT[0] line, latched on TTC_EVCNTLSTR strobe
1	R	TTC_BCNT[1] line, latched on TTC_EVCNTLSTR strobe
2	R	TTC_BCNT[2] line, latched on TTC_EVCNTLSTR strobe
3	R	TTC_BCNT[3] line, latched on TTC_EVCNTLSTR strobe
4	R	TTC_BCNT[4] line, latched on TTC_EVCNTLSTR strobe
5	R	TTC_BCNT[5] line, latched on TTC_EVCNTLSTR strobe
6	R	TTC_BCNT[6] line, latched on TTC_EVCNTLSTR strobe
7	R	TTC_BCNT[7] line, latched on TTC_EVCNTLSTR strobe
8	R	TTC_BCNT[8] line, latched on TTC_EVCNTLSTR strobe
9	R	TTC_BCNT[9] line, latched on TTC_EVCNTLSTR strobe

10	R	TTC_BCNT[10] line, latched on TTC_EVCNTLSTR strobe
11	R	TTC_BCNT[11] line, latched on TTC_EVCNTLSTR strobe
12	R	-
13	R	-
14	R	-
15	R	-

4.17. CSRB14 (base + 3a)

Bit	Access	Function
0	R	TTC_BCNT[0] line, latched on TTC_EVCNTHSTR strobe
1	R	TTC_BCNT[1] line, latched on TTC_EVCNTHSTR strobe
2	R	TTC_BCNT[2] line, latched on TTC_EVCNTHSTR strobe
3	R	TTC_BCNT[3] line, latched on TTC_EVCNTHSTR strobe
4	R	TTC_BCNT[4] line, latched on TTC_EVCNTHSTR strobe
5	R	TTC_BCNT[5] line, latched on TTC_EVCNTHSTR strobe
6	R	TTC_BCNT[6] line, latched on TTC_EVCNTHSTR strobe
7	R	TTC_BCNT[7] line, latched on TTC_EVCNTHSTR strobe
8	R	TTC_BCNT[8] line, latched on TTC_EVCNTHSTR strobe
9	R	TTC_BCNT[9] line, latched on TTC_EVCNTHSTR strobe
10	R	TTC_BCNT[10] line, latched on TTC_EVCNTHSTR strobe
11	R	TTC_BCNT[11] line, latched on TTC_EVCNTHSTR strobe
12	R	-
13	R	-
14	R	-
15	R	-

4.18. CSRB15 (base + 3c)

Bit	Access	Function
0	R	-
1	R	-
2	R	TTC_BRCST[2]
3	R	TTC_BRCST[3]
4	R	TTC_BRCST[4]
5	R	TTC_BRCST[5]
6	R	TTC_BRCST[6]
7	R	TTC_BRCST[7]
8	R	-
9	R	-
10	R	TTC_DQ[0]
11	R	TTC_DQ[1]
12	R	TTC_DQ[2]
13	R	TTC_DQ[3]
14	R	-
15	R	-

4.19. CSRB16 (base + 3e)

Bit	Access	Function
0	R	TTC_DOUT[0] line latched on TTC_DOUTSTR strobe
1	R	TTC_DOUT[1] line latched on TTC_DOUTSTR strobe
2	R	TTC_DOUT[2] line latched on TTC_DOUTSTR strobe
3	R	TTC_DOUT[3] line latched on TTC_DOUTSTR strobe

4	R	TTC_DOUT[4] line latched on TTC_DOUTSTR strobe
5	R	TTC_DOUT[5] line latched on TTC_DOUTSTR strobe
6	R	TTC_DOUT[6] line latched on TTC_DOUTSTR strobe
7	R	TTC_DOUT[7] line latched on TTC_DOUTSTR strobe
8	R	TTC_SUBAD[0] line latched on TTC_DOUTSTR strobe
9	R	TTC_SUBAD[1] line latched on TTC_DOUTSTR strobe
10	R	TTC_SUBAD[2] line latched on TTC_DOUTSTR strobe
11	R	TTC_SUBAD[3] line latched on TTC_DOUTSTR strobe
12	R	TTC_SUBAD[4] line latched on TTC_DOUTSTR strobe
13	R	TTC_SUBAD[5] line latched on TTC_DOUTSTR strobe
14	R	TTC_SUBAD[6] line latched on TTC_DOUTSTR strobe
15	R	TTC_SUBAD[7] line latched on TTC_DOUTSTR strobe

4.20. CSRB17 (base + 40)

Bit	Access	Function
0	R	Date of the current firmware version, Day (LSB)
1	R	Date of the current firmware version, Day
2	R	Date of the current firmware version, Day
3	R	Date of the current firmware version, Day
4	R	Date of the current firmware version, Day (MSB)
5	R	Date of the current firmware version, Month (LSB)
6	R	Date of the current firmware version, Month
7	R	Date of the current firmware version, Month
8	R	Date of the current firmware version, Month (MSB)
9	R	Date of the current firmware version, Year (LSB) *
10	R	Date of the current firmware version, Year *
11	R	Date of the current firmware version, Year *
12	R	Date of the current firmware version, Year (MSB) *
13	R	« 0 »
14	R	« 0 »
15	R	« 0 »

* add to 2000 to obtain a year

4.21. CSRB18 (base + 42)

Bit	Access	Function
0-7	R	Represents the TTCrx hardwired ID on Data[7..0] lines. Valid after TTCrx reset command (write to base+5c then wait 65 us)
15-8	R	Represents the TTCrx hardwired ID on SubAdr[7..0] lines. Valid after TTCrx reset command (write to base+5c then wait 65 us)

4.22. CSRB19 (base + 44)

Bit	Access	Function
15-0	R	16-bit counter of BRCSTSTR1 pulses from the TTCrx ASIC
	W	Resets the counter to « 0 » (write any data)

4.23. CSRB20 (base + 46)

Bit	Access	Function
15-0	R	16-bit counter of BRCSTSTR2 pulses from the TTCrx ASIC
	W	Resets the counter to « 0 » (write any data)

4.24. CSRB21 (base + 48)

Bit	Access	Function
15-0	R	16-bit counter of DOUTSTR pulses from the TTCrx ASIC
	W	Resets the counter to « 0 » (write any data)

5. JTAG Access to FPGA and EPROM

One Xilinx XC2V250-4FG456 FPGA that requires one XC18V02 EPROM is being used on CCB2004 board. Both chips can be accessed over JTAG bus; the EPROM is the first device in JTAG chain, and the FPGA is the second one. JTAG signals can be emulated using write and read operations directed to bits CSRA1[8..5]. In addition to that, JTAG access from an external computer is possible using a Xilinx Parallel Cable IV. An on-board jumper S10-1 defines which of these two options is activated.

6. I2C Programming

The I2C access to TTCrx ASIC is supported by CSRA1[4..1] bits. During read operations the CSRA1[1] must be "0". During write operations CSRA1[1] must be "1".

For the TTCrx ASIC, all data transfers over the I2C bus are performed using only two registers: the I2C_pointer register and the I2C_data register. The I2C_pointer register is 5-bit wide and contains the address of the internal register as defined in Table 3 on page 13 in [3]. Hence, each I2C access is performed in two steps:

1. Write to register number in the I2C_pointer register
2. Read or write the I2C_data register.

Each TTCrx ASIC occupies two addresses in the 7-bit I2C address space. The 7-bit I2C address is derived from the content of the ID_I2C <5..0> base address register as described in Table 9 based on [3]. ID_I2C<5..0> bits are defined by the jumpers on Dout<5..0>lines on the TTCrq mezzanine board.

Table 9: I2C address calculation

I2C access register	Resulting 7-bit I2C address
I2C_pointer	ID_I2C<5..0> * 2
I2C_data	ID_I2C<5..0> * 2 + 1

Note that the correct operation of the I2C bus requires the TTCrx to be locked to the TTC signal ("TTC_Ready"=1).

7. Serial Number Access

There is a Silicon Serial Number DS2401 [8] chip that consists of a factory-lasered, 64-bit ROM that includes a unique 48-bit serial number, an 8-bit CRC, and an 8-bit Family Code (01h). Data is transferred serially via the 1-Wire protocol, read and write least significant bit first. The protocol details and timing diagrams are given in [8]. An access to serial number chip consists of three phases: Initialization, ROM Function Command, and Read Data.

The Initialization sequence consists of a "Reset pulse" transmitted by the master followed by a "Presence pulse" transmitted by the DS2401. The "Presence pulse" lets the bus master know that the DS2401 is on the bus and ready to operate. For the Initialization, the "Reset pulse" should be sent, then CSRB9[2] should be checked, and, when the CSRB9[2]=1, the CSRB9[0] should be checked. If CSRB9[0]=0 at this moment, that means that the "Presence pulse" was sent and the next step can be performed.

The ROM Function Command phase consists of sending a Read ROM command [33h] or [0Fh] to DS2401. The first bit ("Write-one") should be sent, then CSRB9[4] scanned, and, when CSRB9[4]=1, the next bit of command should be sent. Since all commands are 8-bit long, eight write operations are necessary.

The Read Data phase consists of 64 read cycles. Each cycle starts with sending a "Read pulse", then CSRB9[3] is scanned, and, when CSRB9[3]=1, the valid data bit should be received from CSRB9[1]. Note the first data bit should be "1" and the next seven bits should be "0" (they represent the Family Code 01h). Bits 49-56 are also "0" and bits 57-64 represent the CRC code.

8. On-board switches, jumpers and fuses

Switch S1 provides either +3.3V (S1-1 is "on" and S1-2 is "off") or +5.0V (S1-1 is "off" and S1-2 is "on") to all components (except pin-preamplifier) on TTCrq mezzanine. The jumper S1-2 should be installed. Pin-preamplifier is powered from +5V permanently.

Switch S2 provides control inputs for the TTCrx and QPLL ASICs. When S2-1 is "on", the "mode" input of QPLL can be controlled from CSRB7[0]. When S2-1 is "off", the mode="1" due to internal pull-up resistor. When S2-2 is "on", the "externalControl" input of QPLL can be controlled from CSRB7[3]. When S2-2 is "off", the externalControl="1" due to internal pull-up resistor. When S2-3 is "on", the "autoRestart" input of the QPLL can be controlled from CSRB7[2]. When S2-3 is "off", the autoRestart="1" due to internal pull-up resistor. When S2-4 is "on", the "reset" input of QPLL can be controlled from CSRB7[1]. The S2-5 allows to connect (when "on") a pull-up 1Kohm resistor to "reset" input of the QPLL. Either S2-4 or S2-5 should be "on". When S2-6 is "on", the "reset" input of the TTCrx can be controlled from CSRA1[9].

8-position switches S3, S4, S6 and S9 are needed to provide a "crate_done" signal indicating that all peripheral boards and the CCB'2004 itself are in a correct initial state after power up and/or reconfiguration. When any of these switches is set to "on", the corresponding signal (see Table 10) will be participating in a decision. When "off", the signal is not involved in a decision. Particularly, for all unplugged peripheral boards the corresponding switches should be set "off". The resulting "crate_done" signal is active when all lines participating in a decision are active.

Table 10

Switch	Source	Switch	Source	Switch	Source	Switch	Source
S3-1	TMB1	S4-1	TMB9	S6-1	DMB8	S9-1	ALCT7
S3-2	TMB2	S4-2	DMB1	S6-2	DMB9	S9-2	ALCT8
S3-3	TMB3	S4-3	DMB2	S6-3	ALCT1	S9-3	ALCT9
S3-4	TMB4	S4-4	DMB3	S6-4	ALCT2	S9-4	MPC
S3-5	TMB5	S4-5	DMB4	S6-5	ALCT3	S9-5	FPGA_done *
S3-6	TMB6	S4-6	DMB5	S6-6	ALCT4	S9-6	TTCrx_ ready
S3-7	TMB7	S4-7	DMB6	S6-7	ALCT5	S9-7	QPLL_lock
S3-8	TMB8	S4-8	DMB7	S6-8	ALCT6	S9-8	FPGA lock*

* Indicate that Xilinx FPGA on CCB'2004 board was configured from its EPROM and its internal DLL was locked successfully.

The S5 and S7 switches define the sources of the 40Mhz and 80Mhz clocks.
When S5-1 is "on", the source of the main system clock is Clock40Des1 from TTCrx.
When S5-2 is "on", the source is CMOS40Mhz clock from the TTCrq mezzanine.
When S5-3 is "on", the source is on-board oscillator (80.157Mhz divided by 2).
When S5-4 is "on" the clock is from QPLL ASIC after LVDS-to-LVTTL translator residing on CCB2004 board.
Note that S5-2 and S5-4 provide essentially the same clock source, but the clock from S5-2 is delayed ~2 ns in respect to S5-4 due to additional buffer on CCB'2004 board.
Only one S5 switch should be set "on". **The default setting is S5-4 "on".**

Switch S7 provides either 40Mhz or 80Mhz clock sources for selected slots (Table 6).
When S7-1 is "on", the source is 40Mhz from one of the sources specified by S5.
The S7-2 should be used only when S7-1 is "off'. S7-2 defines the 80Mhz clock source from S7-3 or S7-4 to be distributed to backplane.
When S7-3 is "on", the source of 80Mhz clock is QPLL.
When S7-4 is "on", the source of 80Mhz clock is on-board oscillator.
Only one S7-3 or S7-4 should be set "on".
The default setting is when both S7-1 and S7-3 are set "on" (all clocks are 40Mhz).

When S8-1 is "on", the on-board oscillator turns off. S8-2 is not used.

Switch S10 is being used to configure/program the FPGA and EPROM over JTAG.
When S10-1 is "on", both FPGA and EPROM are accessible over JTAG connector on the front panel. When S10-1 is "off", they are accessible from VME using CSRA1[8..5].
S10-2 and S10-3 define the configuration mode of the FPGA (see Table 11 below).
Master SelectMAP mode should be set by default.

Table 11

S10-2	S10-3	Mode
off	off	Slave Serial
on	off	Boundary Scan
off	on	Master SelectMAP
on	on	Not used

S10-4, 5, 6 allow to change the order of EPROM and FPGA in a JTAG chain (Table 12).

Table 12

S10-4	S10-5	S10-6	JTAG chain
on	off	on	XC18V02VQ44C EPROM + XC2V250-FG456 FPGA
off	on	off	XC18V02VQ44C EPROM
All others			Not effective

S10-7 controls the HSWAP_EN pin of the FPGA. If "on", it activates the internal pull-up for user i/o in the device prior to configuration. By default, HSWAP_EN is tired "1" with internal pull-up resistor. S10-8 (when "on") allows to reconfigure the FPGA from EPROM on "Hard_reset" commands (either dedicated TTC "CCB_Hard_Reset" or common TTC "Hard_Reset" (see Table 7) or VME write to CSRA2 register) .

The default settings are: S10-1 "off", S10-2 "off", S10-3 "on", S10-4 "on", S10-5 "off" and S10-6 "on", S10-7 "off", S10-8 "off".

S11 should be used only for debugging purposes. It allows to select the geographical addresses GA[4..0] when VME64x backplane is not available. **The default state is "off" for all S11 switches.**

Jumpers JP1 and JP2 defines the mode of GTLP transmitters to custom backplane: either transparent (JP2 installed) or clocked (JP1 installed). Jumper JP3 (when installed) allows to turn all GTLP transmitters off (isolate them from backplane) when the FPGA is being configured. Jumper JP4 allows the FPGA to control the VME DACK line in addition to discrete logic interface. Jumper JP5, when installed, allows to reset the FPGA on common TTC "Hard_Reset" command (if S10-8 is "on"). If JP5 is not installed, but S10-8 is "on", then only the dedicated TTC "CCB_Hard_Reset" command and VME write to CSRA2 will cause reloading of the FPGA. **By default, JP2 and JP5 are installed, and JP1, JP3 and JP4 are not installed.**

Fuse F4 must be installed at any time, it provides +5V power from VME backplane. Fuse F3 provides +3.3V power from the VME64x backplane, while F5 provides +3.3V from on-board voltage regulator U95. Only one (by default, F3) should be installed. Fuse F2 is needed to provide +1.5V for the FPGA core. It should be installed at any time. Fuse F1 is needed to provide +1.5V for on board GTLP receivers from custom backplane and should be installed as well.

9. Front Panel

Two 34-pin connectors (Tables 13-14) provide several status outputs for monitoring purposes. The 18 CLCT and ALCT status signals are provided from the peripheral custom backplane directly. The 40Mhz clock output is the same as selected by S5 (see Section 8). Four signals BC0, Cmd_strobe, Data_Strobe, L1_Reset (Table 13) are the ones that the CCB2004 translates to backplane in the "FPGA" mode. The fifth signal in Table 13, the L1A is a logical OR of 10 sources described in Section 3.3. These sources can be masked out with the CSRB1 and the result can be delayed with the CSRB5. These five outputs are valid in any ("Discrete Logic" or "FPGA") mode of the CCB2004 operation. Texas Instruments SN75LVDS387 output LVDS drivers are being used. An optional 100 Ohm termination networks are provided (socketed SIP packages).

Table 13: Connector P4 (outputs)

Pin	Signal	Pin	Signal
1	Clct_status[0]+	2	Clct_status[0]-
3	Clct_status[1]+	4	Clct_status[1]-
5	Clct_status[2]+	6	Clct_status[2]-
7	Clct_status[3]+	8	Clct_status[3]-
9	Clct_status[4]+	10	Clct_status[4]-
11	Clct_status[5]+	12	Clct_status[5]-
13	Clct_status[6]+	14	Clct_status[6]-
15	Clct_status[7]+	16	Clct_status[7]-
17	Clct_status[8]+	18	Clct_status[8]-
19	Ccb_clock40+	20	Ccb_clock40-
21	BC0+	22	BC0-
23	L1A+	24	L1A-
25	Cmd_Strobe+	26	Cmd_Strobe-
27	Programmable Output_1+ (Data_Strobe+)	28	Programmable Output_1- (Data_Strobe-)
29	Programmable_Output_2+ (L1Reset+)	30	Programmable Output_2- (L1Reset-)
31		32	
33		34	

Table 14: Connector P6 (outputs)

Pin	Signal	Pin	Signal
1	Alct_status[0]+	2	Alct_status[0]-
3	Alct_status[1]+	4	Alct_status[1]-
5	Alct_status[2]+	6	Alct_status[2]-
7	Alct_status[3]+	8	Alct_status[3]-
9	Alct_status[4]+	10	Alct_status[4]-
11	Alct_status[5]+	12	Alct_status[5]-
13	Alct_status[6]+	14	Alct_status[6]-
15	Alct_status[7]+	16	Alct_status[7]-
17	Alct_status[8]+	18	Alct_status[8]-
19		20	
21		22	
23		24	
25		26	
27		28	
29		30	
31		32	
33		34	

Table 15: Connector P7 (LVDS inputs)

Pin	Signal	Pin	Signal
1	External_l1accept+	2	External_l1accept-
3	CLCT_external_trigger+	4	CLCT_external_trigger-
5	ALCT_external_trigger+	6	ALCT_external_trigger-
7		8	
9		10	
11		12	
13		14	

15		16	
17		18	
19		20	

There are 22 LEDs on the front panel:

- "+3.3" from VME64x backplane (if F3 is installed) or from on-board voltage regulator U95 (if F5 is installed) (green, D23)
- "+5.0" from VME backplane (green, D19)
- "+1.5" for FPGA core, from on-board voltage regulator U93 (green, D24)
- "+1.5B" for GTLP terminators from custom backplane powers (green, D25)
- "L1A" L1ACCEPT (red, with one-shot, if L1ACC was sent to backplane, D1)
- "BC0" BC0 (red, with one-shot, if BC0 signal was sent to backplane, D3)
- "HRESET" Hard_Reset (any Hard_Reset except the CCB_Hard_reset) that has been sent to backplane) (red, with one-shot, D5)
- "CCBHR" CCB_Hard_Reset (red, with one-shot, D7)
- "L1RES" L1 Reset (red, with one-shot, D10)
- "EVCRES" Event Counter Reset (red, with one-shot, D6)
- "BCRES" Bunch Counter Reset (red, with one-shot, D8)
- "CMDSTR" Command Strobe (red, with one-shot, D9)
- "DATSTR" Data Strobe (red, with one-shot, D11)
- "DACK" indicates VME access to CCB2004 board (yellow, with one-shot, D12)
- "TTCRDY" TTCrx_Ready indicates that the TTCrx is in normal operation mode (green)
- "JTAG" indicates JTAG access from VME to FPGA using CSRA1[8..5] bits (yellow, D2)
- "I2C" indicates access to TTCrx ASIC over I2C bus using CSRA1[4..1] bits (yellow, D4)
- "DISLOG" Discrete_Logic indicates that commands are sent from the TTC system to custom backplane through the discrete logic interface (when CSRA1[0]=1) (green, D18)
- "DONE" configuration of the Xilinx FPGA from its EPROM was done successfully (green, D14).
- "CRDONE" Crate_Done configuration of all connected boards installed in a crate from their EPROM's was done successfully AND TTCrq board is READY AND an FPGA was configured from its EPROM successfully AND QPLL on a TTCrq board was locked (green, D15)
- "QLOCK" QPLL_Locked status (green, D16).
- "CLK40" CLOCK indicates that the main 40Mhz clock from the selected source defined by S5 is provided for FPGA and the DLL in FPGA is locked (green, counter output running at ~10Hz, D17).

References

[1]. EMU Peripheral Crate Specification. UCLA High Energy Physics. Version 3.3. February 9, 2001. Available at: ftp://www-collider.physics.ucla.edu/pub/cms/peripheral_crate/documentation/

[2]. B.G.Taylor. Timing, Trigger and Control (TTC) System for LHC Detectors. CERN/ECP http://www.cern.ch/TTC/intro.html
[3]. J.Christiansen, A.Marchioro, P.Moreira and T.Toifl. TTCrx Reference Manual. August 2005. Version 3.10. CERN-ECP/MIC, Geneva, Switzerland. Available at http://bonner-ntserver.rice.edu/cms/TTCrx_manual3.10.pdf
[4]. Clock and Control Board for the CSC Peripheral and Track Finder Electronics. Specification. Available at http://bonner-ntserver.rice.edu/CMS/CCB'2001_20.pdf
[5]. http://proj-qpll.web.cern.ch/proj-qpll/
[6]. Alex Madorsky. University of Florida. Peripheral backplane pin assignment. Available at: http://www.phys.ufl.edu/~madorsky/backplane/PinAssignment.doc
[7]. CSC Track Finder Crate Specification. Draft 1/09/2003. Available at: http://www.phys.ufl.edu/~madorsky/TrackFinder/TF_backplane_v4.doc
[8]. DS2401 Silicon Serial Number Specification. Available at http://pdfserv.maxim-ic.com/en/ds/DS2401.pdf

History

10/13/2003: Initial release
10/15/2003: Change in decoding scheme: use data[7..2] instead of data[5..0] (Table 7).
11/12/2003: Tables 4B and 5B were added.
12/11/2003: QPLL_locked signal was added to custom backplane (former ccb_reserved[1]).
12/16/2003: Table 11 was added.
01/16/2004: Table 3 was renamed to Table 3A. Table 3B was added.
01/20/2004: Input connector (Table 12) was changed to 20-pin.
02/27/2004: Table 8 (VME addresses) and CSRB registers were added.
03/12/2004: Section 7 (Serial Number Access) was added.
03/19/2004: Figure 2 was added.
03/22/2004: Correction in Table 12, sections 4.5 and 4.6.
03/31/2004: Figure 1 was updated.
08/13/2004: Fixes in Section 8.
10/13/2004: CSRB18 was added. “Reset TTCrx” command was added.
11/10/2004: Corrections is Tables 4B, 5A, 5B.
11/17/2004: Change from version 2.0 to version 3.0 (production board)
Update of 4.1, 4.2, 4.3.
12/24/2004: Changes in section 4.18.
04/06/2005: Section 2 was expanded. Update of Section 8.
04/12/2005: Section 8 was updated.
07/28/2006: Minor corrections throughout the text.
09/01/2006: Changes in sections 3.3 (DMB_cfeb_calibrate[2..0] pulses were added to a list of the L1ACC sources) and 4.4 (bit CSRB1[2] was defined). Changes in register format, sections 4.15, 4.16, 4.17, 4.18.
06/27/2007: Command OC0 (Orbit Counter Reset) was added (Table 7).
02/15/2008: Registers CSRB19, CSRB20 and CSRB21 were added.
04/02/2008: Minor addition in Section 9.