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

2001 Prototype Specification

Rice University

Version 2.2

08/04/2006

Abstract

This document describes the functionality of the Clock and Control Board (CCB) Prototype (2001) for the CMS Cathode Strip Chamber (CSC) peripheral and Track Finder (TF) electronics and its communications with other modules in the peripheral and TF crates.

Introduction

The CSC Trigger Electronics consists of on-chamber mounted Front End Boards (FEB), electronics on the periphery of the detector, and a TF crate in the counting room. 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) described below. More details about the peripheral backplane and communications between modules are given in [1-2]. The TF crate [3] comprises 12 Sector Processors (SP), one Muon Sorter (MS), one CCB and four additional boards (3 MPC and one DDU) needed for testing. The CCB for the TF crate has slightly reduced functionality due to lower number of serviced modules. In both backplanes the CCB resides on slot 12.

All elements of the Muon Trigger System should be synchronized with respect to the LHC beam crossing. 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 [4-5]. The TTC system is based on an optical fan-out system and provides the distribution of the LHC timing reference signal, the first level trigger decisions and its associated bunch and event numbers from one source to about 1000 destinations. The TTC system also allows to adjust the timing of all of these signals. At the lowest levels of the TTC system, the TTCrx ASIC [5] receives control and synchronization information 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 installed on the main Clock and Control Board.

1. TTC Interface

The list of signals that the CCB accepts and transmits from/to the TTCrx mezzanine board is given in Table 1.

Table 1

Signal	Bits	Short Description
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
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	

2. Backplane Interface

The list of signals that the CCB distributes to the custom peripheral backplane is based on [1-2] and given in Table 2. More detailed description of these signals, timing etc is given in the following sections. Pin assignment of the custom backplane connectors for the peripheral CCB slot is based on [2] (both on-board connectors are AMP 100145-1 female 125-pin 25-row by 5 pins) is given in Tables 3 and 4.

The list of signals that the CCB distributes to the custom Track Finder crate is based on [3] and given in Table 5. Pin assignment of the custom backplane connectors for the TF CCB slot is given in Tables 6 and 7.

Table 2

Inputs and Outputs of the Peripheral CCB

Signal	Bits	Source	Destination	Type	Logic	Duration
Clock Bus: Clock Distribution & Bunch Crossing						
ccb_clock40[1]	19	TTCrx[2], FP[3],Xtal	All 19 Slots	Point-to-point	LVDS	40MHz
Fast Control Bus						
ccb_clock40_enable	1	VME[4], FP	All 19 Slots	Bussed	GTLP	Pulse, n counts
ccb_cmd[5..0]	6	TTCrx, VME	All 19 Slots	Bussed	GTLP	Level
Ccb_ttcrx_ready (former ccb_reserved[0])	1	TTCrx	All 19 slots	Bussed	GTLP	Level
ccb_evcntres	1	TTCrx, VME	All 19 Slots	Bussed	GTLP	25ns
ccb_bcntres	1	TTCrx, VME	All 19 Slots	Bussed	GTLP	25ns
Ccb_L1Reset (former ccb_reserved[4])	1	dTTCrx, VME	All 19 Slots	Bussed	GTLP	25 ns
ccb_cmd_strobe	1	TTCrx, VME	All 19 Slots	Bussed	GTLP	25ns
ccb_bc0	1	dTTCrx[5], VME, FP	All 19 Slots	Bussed	GTLP	25ns+ECL FP
ccb_l1accept	1	TTCrx, VME, FP	All 19 Slots	Bussed	GTLP	25ns+ECL FP
ccb_data[7..0]	8	dTTCrx, VME	All 19 Slots	Bussed	GTLP	Level
ccb_data_strobe	1	TTCrx, VME	All 19 Slots	Bussed	GTLP	25ns
ccb_reserved[3..1]	3	VME	All 19 Slots	Bussed	GTLP	25ns
Total	26					
TMB Reload Bus: ALCT+CLCT+TMB FPGA Reload						
tmb_hard_reset[6]	1	dTTCrx, VME	9 TMB	Bussed	GTLP	400ns
tmb_cfg_done[8..0]	9	9 TMB	VME	Point-to-Point	GTLP	Level
alct_hard_reset	1	dTTCrx, VME	9 TMB	Bussed	GTLP	400ns
alct_cfg_done[8..0]	9	9 TMB (fromALCT)	VME	Point-to-Point	GTLP	Level
Tmb_soft_reset (former tmb_reserved[1])	1	dTTCrx, VME	9 TMB	Bussed	GTLP	25 ns
tmb_reserved[0]	1	VME	9 TMB	Bussed	GTLP	25ns
Total	22					
MPC Reload Bus: MPC FPGA Reload						
mpc_hard_reset	1	dTTCrx, VME	MPC	Point-to-Point	GTLP	400ns
mpc_cfg_done	1	MPC	CCB	Point-to-Point	GTLP	Level
Mpc_soft_reset (former mpc_reserved[2])	1	dTTCrx, VME	MPC	Point-to-Point	GTLP	25 ns
mpc_reserved[1..0]	2	VME	MPC	Point-to-Point	GTLP	25ns
Total	5					

[1] Clock: CCB distributes Clock40Des1 from the TTCrx or a crystal or FP
[2] TTCrx: Signals derived unaltered by the CCB from the TTCrx chip
[3] FP: Signals input or output via CCB Front Panel (ECL levels)
[4] VME: Signals read or written via the CCB VME interface
[5] dTTCrx: Signals decoded by CCB from TTCrx Brcst[7..2]
[6] hard_reset: CCB decodes separate TTC commands for tmb_hard_reset, alct_hard_reset, dmb_hard_reset, and mpc_hard_reset. The TTC command "hard_reset" pulses all hard resets simultaneously.

DMB Reload Bus: DMB FPGA Reload						
dmb_hard_reset	1	dTTCrx, VME	9 DMB	Bussed	GTLP	400ns
dmb_cfg_done[8..0]	9	9 DMB	CCB	Point-to-Point	GTLP	Level
Dmb_soft_reset (former dmb_reserved[2])	1	dTTCrx, VME	9 DMB	Bussed	GTLP	25 ns
dmb_reserved[1..0]	2	VME	9 DMB	Bussed	GTLP	25ns
Total	13					
DAQ Special Purpose Bus [Used by DMB and TMB]						
dmb_cfeb_calibrate[2..0]	3	VME, FP, dTTCrx	9 DMB 9TMB	Bussed	GTLP	25ns ECL FP
dmb_l1a_release	1	9 DMB, 9 TMB	CCBL1ALogic	Bussed	GTLP	25ns
dmb_reserved_out[4..0]	5	VME	9 DMB 9TMB	Bussed	GTLP	-
dmb_reserved_in[2..0]	3	9 DMB, 9 TMB	VME	Bussed	GTLP	-
Total	12					
Trigger Special Purpose Bus [Used by TMB only]						
alct_adb_pulse_sync	1	VME, FP, dTTCrx	9 TMB	Bussed	GTLP	25ns ECL FP
alct_adb_pulse_async	1	dTTCrx,VME, FP	9 TMB	Bussed	GTLP	25ns ECL FP
clct_external_trigger	1	VME, FP, dTTCrx	9 TMB	Bussed	GTLP	25ns ECL FP
alct_external_trigger	1	VME, FP, dTTCrx	9 TMB	Bussed	GTLP	25ns ECL FP
clct_status[8..0]	9	9 TMB	VME, FP	Bussed	GTLP	25ns ECL FP
alct_status[8..0]	9	9 TMB	VME, FP	Bussed	GTLP	25ns ECL FP
tmb_l1a_request	1	9 TMB, FP	CCBL1ALogic	Bussed	GTLP	25ns ECL FP
tmb_l1a_release	1	9 TMB	CCBL1ALogic	Bussed	GTLP	25ns
tmb_reserved_in[4..0]	5	9 TMB	VME	Bussed	GTLP	-
tmb_reserved_out[2..0]	3	VME	9 TMB	Bussed	GTLP	-
Total	32					

Table 3. X40 Peripheral Backplane Connector (CCB slot)

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_c_d_9	E10	dmb_c_d_9
A11	Tmb_c_d_1	B11	alct_c_d_1	C11	GND	D11	dmb_c_d_8	E11	Tmb_c_d_9
A12	dmb_c_d_1	B12	Tmb_c_d_2	C12	GND	D12	Tmb_c_d_8	E12	alct_c_d_8
A13	alct_c_d_2	B13	dmb_c_d_2	C13	GND	D13	alct_c_d_7	E13	dmb_c_d_7
A14	Tmb_c_d_3	B14	alct_c_d_3	C14	GND	D14	dmb_c_d_6	E14	Tmb_c_d_7
A15	dmb_c_d_3	B15	Tmb_c_d_4	C15	GND	D15	Tmb_c_d_6	E15	alct_c_d_6
A16	alct_c_d_4	B16	dmb_c_d_4	C16	GND	D16	Mpc_h_reset	E16	Mpc_rsv0
A17	Tmb_c_d_5	B17	alct_c_d_5	C17	GND	D17	Mpc_rsv1	E17	Mpc_softres
A18	dmb_c_d_5	B18		C18	GND	D18	mpc_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_ttcrx_rd	A25	Ccb_rsv1	C25	GND	D25	Ccb_rsv2	E25	Ccb_rsv3

Table 4. X41 Peripheral Backplane Connector (CCB slot)

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	Clct_status3	B6	Alct_status4	C6	GND	D6	Alct_status5	E6	Alct_status6
A7	Clct_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_in0
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		A25		C25	GND	D25		E25	

Table 5

Inputs and Outputs of the Track Finder CCB

Signal	Bits	Source	Destination	Type	Logic	Duration
Clock Bus: Clock Distribution & Bunch Crossing						
ccb_clock40	17	TTCrx, FP,Xtal	All 17 Slots	Point-to-point	LVDS	40.08MHz
ccb_clock40_enable	1	VME, FP	All 17 Slots	Bussed	GTLP	Pulse, n counts
Fast Control Bus						
ccb_cmd[5..0]	6	TTCrx, VME	All 17 Slots	Bussed	GTLP	Level
Ccb_ttcrx_ready (former ccb_reserved[0])	1	TTCrx	All 17 slots	Bussed	GTLP	Level
ccb_evcntres	1	TTCrx, VME	All 17 Slots	Bussed	GTLP	25ns
ccb_bcntres	1	TTCrx, VME	All 17 Slots	Bussed	GTLP	25ns
Ccb_L1Reset (former ccb_reserved[4])	1	DTTCrx, VME	All 17 Slots	Bussed	GTLP	25 ns
ccb_cmd_strobe	1	TTCrx, VME	All 17 Slots	Bussed	GTLP	25ns
ccb_bc0	1	DTTCrx, VME, FP	All 17 Slots	Bussed	GTLP	25ns+ECL FP
ccb_l1accept	1	TTCrx, VME, FP	All 17 Slots	Bussed	GTLP	25ns+ECL FP
ccb_data[7..0]	8	TTCrx, VME	All 17 Slots	Bussed	GTLP	Level
ccb_data_strobe	1	TTCrx, VME	All 17 Slots	Bussed	GTLP	25ns
ccb_reserved[3..1]	3	VME	All 17 Slots	Bussed	GTLP	25ns
Total	25					
SP and MS Reload Signals						
SP_hard_reset	1	dTTCrx, VME	12 SP	Bussed	GTLP	400ns
SP_cfg_done[12..1]	12	12 SP	VME	Point-to-Point	GTLP	Level
MS_hard_reset	1	dTTCrx, VME	MS	Point-to-point	GTLP	400ns
MS_cfg_done	1	MS	VME	Point-to-Point	GTLP	Level
MPC_hard_reset	1	dTTCrx, VME	3 MPC+DDU	Bussed	GTLP	400 ns
MPC_cfg_done[3..1]	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 Signals						
SP_l1a_request	1	12 SP	CCB L1A logic	Bussed	GTLP	25 ns
MS_l1a_request	1	MS	CCB L1A logic	Bussed	GTLP	25 ns
SP_to-CCB[2..0]	3	12 SP	CCB	Bussed	GTLP	25 ns
MS-to-CCB[1..0]	2	MS	CCB	Bussed	GTLP	25 ns
CCB-to-MS[2..0]	3	CCB	MS	Point-to-Point	GTLP	25 ns
Total	10					

Table 6. X40 Track Finder Backplane Connector (CCB slot)

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_d_12
A11		B11		C11	GND	D11		E11	SP_c_d_11
A12		B12		C12	GND	D12		E12	SP_c_d_10
A13	Ddu_c_done	B13	Mpc1_c_don	C13	GND	D13		E13	SP_c_d_9
A14	Mpc2_c_don	B14	Mpc3_c_don	C14	GND	D14		E14	SP_c_d_8
A15	SP_c_done_1	B15	SP_c_done_2	C15	GND	D15		E15	SP_c_d_7
A16	SP_c_done_3	B16	SP_c_done_4	C16	GND	D16		E16	MS_c_done
A17	SP_c_done_5	B17	SP_c_done_6	C17	GND	D17		E17	
A18		B18		C18	GND	D18		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_ttcrx_rd	A25	Ccb_rsv1	C25	GND	D25	Ccb_rsv2	E25	Ccb_rsv3

Table 7. X41 Track Finder Backplane Connector (CCB slot)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	SP_hard_res	B1	MS_hard_res	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_ccb0	B8	MS_to_ccb1	C8	GND	D8		E8	
A9		B9		C9	GND	D9		E9	
A10	Mpc_hard_r	B10		C10	GND	D10		E10	
A11		B11		C11	GND	D11		E11	
A12	CCB_to_ms0	B12	CCB_to_ms1	C12	GND	D12	CCB_to_ms2	E12	
A13		B13	SP_to_ccb0	C13	GND	D13	SP_to_ccb1	E13	Sp_to_ccb2
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	

2.1. Clock Distribution

The CCB board normally distributes to all modules in crate the Clock40Des1 deskewed clock signal provided by the TTCrx. In addition to that, there are three non-TTC clock options available: (1) external clock from the front panel connector, (2) continuous 40Mhz clock, generated by CCB on-board quartz (80Mhz divided by 2) and (3) single 25 ns pulse generated upon VME command. Two CSR1 bits are responsible for the clock source selection. The ccb_clock40 signal is distributed to every slot in the crate over individual point-to-point LVDS lines.

There are fine and coarse delays for the clock and command signals incorporated into TTCrx ASIC. They are programmable via the I2C interface or optical path. The CCB can generate a ccb_clock40_enable signal (pulse of programmable length, equal to 1 to 255 clock periods of 40Mhz clock frequency). This signal would allow pausing the clocking of some critical logic parts on a specific boards for debugging purposes. CSR4[7..0] defines the length of ccb_clock40_enable pulse.

2.2. Fast Control Bus

Six lines of ccb_cmd[5..0] command bus and one strobe ccb_cmd_strobe line would allow CCB and other boards in crate to decode up to 63 separate commands. There are two possible sources for data bus and strobe: either TTCrx_Brcst[7..2] and TTCrx_BrcstStr1 (in "TTCrx" mode) or CSR2[7..2] bits and Wr_CSR2 strobe ("VME" mode). In "TTCrx" mode there are also BCntRes and EvCntRes signals that are translated directly from the TTCrx chip. In "VME" mode these signals may be generated by writing data into CSR2[0] and CSR2[1] respectively. Timing on ccb_cmd[5..0], ccb_evcntres, ccb_bcntres and ccb_cmd_strobe lines is similar to the TTCrx timing (see fig.7 in [5]). The decoding scheme of the ccb_cmd[5..0] is shown in Table 8.

In a similar fashion, the ccb_data[7..0] bus and corresponding ccb_data_strobe strobe line may have two sources: TTCrx or CSR3[7..0]. There are two options in a "TTCrx" mode. The TTCrx_Dout[7..0] bits are translated to the backplane only when DQ[3..0]=0 (this means an individually addressed command data mode, see table on Page 31 in [5]). But if command codes 20..23(hex) are recognized from the TTC, then a content of the corresponding 8-bit bunch counter or event counter registers is transmitted to ccb_data[7..0] lines. Timing on the ccb_data[7..0] and ccb_data_strobe lines is similar to TTCrx timing (see fig.8 in [5]).

Table 8

Fast Control Bus Ccb_cmd[5..0] Decoding Scheme

Signal	Code (hex)	Description
BC0 (*)	1	Bunch Crossing Zero
L1 Reset (*)	3	Reset L1 readout buffers and resynchronize optical links
Hard_reset (*)	4	Reload all FPGAs from EPROMs
Start Trigger	6	
Stop Trigger	7	
Test Enable	8	
Private Gap	9	
Private Orbit	A	
Tmb_hard_reset (*)	10	Reload TMB FPGAs from EPROM
Alct_hard_reset (*)	11	Reload ALCT FPGAs from EPROM
Dmb_hard_reset (*)	12	Reload DMB FPGAs from EPROM
Mpc_hard_reset (*)	13	Reload MPC FPGAs 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
Send_bcnt[7..0] (*)	20	Send Bunch_Counter[7..0] to ccb_data[7..0] bus
Send_evcnt[7..0] (*)	21	Send Event_Counter[7..0] to ccb_data[7..0] bus
Send_evcnt[15..8] (*)	22	Send Event_Counter[15..8] to ccb_data[7..0] bus
Send_evcnt[23..16] (*)	23	Send Event_Counter[23..16] to ccb_data[7..0] bus
Inject patterns from TMBs	24	Injects patterns from TMB's internal RAM to MPC
Alct_adb_pulse (*)	25	Generate both synchronous and asynchronous anode discriminator pulses
Inject patterns from MPCs	30	Injects patterns from MPC's input FIFO to SP
Inject patterns from MS	31	Injects patterns from MS input FIFO to Global Muon Trigger

(*) – decoded by CCB

Three ccb_reserved[3..1] signals (25 ns pulses) can be generated upon write to a specific addresses in a CCB address space (see Table 9).

2.3. Reloading Bus

Hard_reset reloading signals are decoded by the CCB and expanded to 400 ns before distribution to the backplane. They also can be generated upon write to specific addresses in the CCB address space (see Table 9). There is also a common Hard_reset command (Table 8) which causes generation of reloading signals to all boards in a crate simultaneously. This signal can also be obtained from the front panel.

Soft_reset signals are also decoded inside the CCB either from TTCrx or VME write. They will be used for FPGA initialization individually on TMB, DMB and MPC boards. There is also a common Soft_Reset command for all boards in a crate.

ALCT, TMB, DMB and MPC cards inform the CCB when they have completed the reconfiguration process by asserting cfg_done signals. These static signals are available for reading from CSR6..CSR8.

The tmb_reserved[0], mpc_reserved[1..0], dmb_reserved[1..0] signals are reserved for future use. In the present design the 25 ns pulses can be generated on these lines upon write to specific addresses in CCB address space (see Table 9). They are synchronized with the selected 40MHz clock source.

2.4. DAQ Special Purpose Bus

Dmb_cfeb_calibrate[2..0] 25 ns pulses to the backplane can be generated upon TTC commands (Table 5), write operation to a specific CCB addresses (Table 9) or on the rising edge of the external pulses coming through the CCB front panel. This scheme works in both "TTCrx" and "VME" modes and CSR1<0> doesn't affect it. These signals are synchronized with the chosen 40Mhz clock.

Dmb_reserved_out[4..0] signals (25 ns pulses in the present scheme) may be generated upon write to a specific CCB addresses (see Table 9). They are synchronized with the selected source of 40Mhz clock.

Dmb_reserved_in[2..0] signals (presumably pulses) cause setting to "1" specific bits in CSR11 (see below). These bits can be set to "0" upon sending "reset CSR11" write command to a specific CCB address (see Table 9).

2.5. Trigger Special Purpose Bus

ALCT_adb_pulse_sync (25 ns) and ALCT_adb_pulse_async (asynchronous in respect to selected 40Mhz clock) can be generated upon TTC commands (see Table 9), write operation to a specific VME address (see Table 9) or on the rising edge of the external pulses coming through the CCB front panel. This scheme works in both "TTCrx" and "VME" modes and CSR1[0] doesn't affect it. The ALCT_adb_pulse_sync is synchronized with the chosen 40Mhz clock and expanded to 500 ns before distribution to backplane. The ALCT_adb_pulse_async signal is the same length as its original source. ALCT_adb_pulse_sync and ALCT_adb_pulse_async also cause generation of L1ACC and Pretriggers (ALCT_external_trigger and CLCT_external_trigger) to backplane as described in more details in 2.6.

The Pretrigger signals ALCT_external_trigger and CLCT_external_trigger can be generated from any source of L1ACC (see 2.6), front panel, TTC system or upon VME command. There are individual masks for them in CSR1. Before distribution to

backplane they can be delayed for 1..255 ticks of 40MHz clock (defined by CSR5[15..8]).

Clct_status[8..0] and alct_status[8..0] signals (presumably pulses) cause setting to "1" specific bits in CSR9 and CSR10 respectively. These bits can be set to "0" upon sending "reset CSR9" and "reset CSR10" write commands to specific CCB addresses (Table 6).

Tmb_reserved_out[2..0] signals (25 ns pulses in the present scheme) may be generated upon write to specific CCB addresses (see Table 6). They are synchronized with the selected source of 40Mhz clock.

Tmb_reserved_in[4..0] signals (presumably pulses) cause setting to "1" specific bits in CSR11 (see below). These bits can be set to "0" upon sending "reset CSR11" write command to a specific CCB address (see Table 6).

2.6. L1ACC and Pretrigger Sources and Control

There are several possible sources for L1ACC and two Pretriggers (ALCT_external_trigger and CLCT_external_trigger):

- TTCrx_L1ACC (25 ns pulse),
- Ext_L1ACC (ECL pulse coming through the CCB front panel),
- Tmb_l1a_request (25 ns pulse coming over backplane),
- artificial L1ACC generated upon VME write to (base + 2a) address (see Table 9),
- any source of ALCT_adb_pulse_sync (VME, front panel, TTC command),
- any source of ALCT_adb_pulse_async (VME, front panel, TTC command).

Each of these sources can be masked using dedicated CSR1 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 CSR5[15..8]). Independently from Pretriggers, the L1ACC can also be delayed for n=1..255 clock cycles where (n) is defined by CSR5[7..0]. The starting point for the delay counting is the same. The ALCT_adb_pulse_sync and ALCT_adb_pulse_async backplane signals are sent to backplane without delay. This logic works in both "TTCrx" and "VME" modes and CSR1[0] doesn't affect it.

There is a 32-bit counter of L1ACC requests available for read, reset, and enable/disable operations over VME. The counter counts the L1ACC requests from any selected source even if the transmission to backplane is disabled. The counter is disabled after power up and CCB internal reset. The 32-bit content of the counter is latched into two 16-bit output registers on read command to (base + 96) address.

In a special CFEB calibration mode, the CCB 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 9). In this mode, the CCB "holds" the next L1ACC and both Pretriggers (by preventing them from being broadcast on the backplane). In a similar way, if the CSR1[13]=0, then the CCB holds the next L1ACC and Pretriggers after the first L1ACC has been sent to backplane. The CCB 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 + 5c) has been initiated.

2.7. Access to TTCrx Signals

Several internal CSR registers provide access to TTCrx output signals for testing and debugging purposes. During the trigger sequence (upon reception of a trigger signal), the contents of the bunch counter and the event counter are made available on the BCnt<11..0> lines. The content of a bunch counter is latched into CSR12<11..0> upon BCntStr strobe. Content of an event counter (12 LSB and 12 MSB) is latched into CSR13<11..0> and CSR14<11..0> upon EvCntLStr and EvCntHStr respectively. In addition to that, the Brcst<7-2>, BrcstStr1, BrcstStr2, BcntRes, EvCntRes and DQ<3..0> bits from the TTCrx outputs are directly latched into CSR15. Content of the Dout<7..0> and SubAd<7..0> are latched into CSR16 upon the DoutStr strobe. Registers CSR12..CSR16 are available for read only.

2.8. Electrical Interface

Fairchild GTLP16612 (or compatible TI SN74GTLP18612) transceivers are used for the communications with other modules in a crate (Table 2). All GTLP signals to backplane 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 a CCB board with 100 Ohm resistors to 1.5V. All bussed signals are terminated on both ends of the backplane.

3. Control and Status Registers (CSR)

3.1. CSR1

CSR1[0] - defines the source of signals distributed to ccb_cmd[5..0] and ccb_cmd_str lines and the source of "Hard reset" signals. If CSR1<0>=1, the source is TTCrx ASIC. If CSR1<0>=0, the source is CSR2.

CSR1[1..2] - defines the source of 40MHz clock distributed to backplane

CSR1[2]	CSR1[1]	
0	0	CCB on-board oscillator (80MHz divided by 2)
0	1	Clock40Des1 from TTCrx
1	0	CCB front panel connector (ECL input)
1	1	Single 25 ns pulse upon write to (Base +38) address

CSR1[3] - mask L1ACC source from TTCrx (disabled if "1" and enabled if "0")

CSR1[4] - mask L1ACC source from VME (disabled if "1" and enabled if "0")

CSR1[5] - mask L1ACC source from TMB_L1AREQ backplane signal (disabled if "1" and enabled if "0")

CSR1[6] - mask FP_BX0 source from the front panel signal (disabled if "1" and enabled if "0")

CSR1[7] - mask "External_L1ACC" signal from the front panel (disabled if "1" and

enabled if "0")
CSR1[8] - "1" enables all (except Clock40) inputs from the front panel. "0" disables all Inputs from the front panel. Clock40 is always enabled.
CSR1[9] - masks generation of delayed ALCT_external_trigger signal to backplane from any L1ACC source (disabled if "1" and enabled if "0")
CSR1[10] - masks generation of delayed CLCT_external_trigger signal to backplane from any L1ACC source (disabled if "1" and enabled if "0")
CSR1[11] - masks generation of delayed Pretriggers and delayed L1ACC from any of ALCT_adb_sync_pulse sources (disabled if "1" and enabled if "0")
CSR1[12] - masks generation of delayed Pretriggers and delayed L1ACC from any of ALCT_adb_async_pulse sources (disabled if "1" and enabled if "0")
CSR1[13] - if "0", then the CCB disables sending delayed L1ACC and both delayed Pretriggers to backplane after the first L1ACC has been translated to backplane. If "1", then L1ACC and pretriggers are enabled unconditionally.
CSR1[14] - mask FP_BCNTRES signal from the front panel (disabled if "1" and enabled if "0")
CSR1[15] - mask FP_Hard_Reset signal from the front panel (disabled if "1" and enabled if "0")

3.2. CSR2

CSR2[0] - BcntRes
CSR2[1] - EvCntRes
CSR2[7..2] - data transmitted to ccb_cmd[5..0] bus when CCB is in "VME" mode (see CSR1). Ccb_cmd_strobe (25 ns pulse) is generated upon write data into CSR2.
CSR2[8..15] - not used

3.3. CSR3

CSR3[7..0] - data transmitted to ccb_data[7..0] bus when CCB is in "VME" mode (see CSR1). Ccb_data_strobe (25 ns pulse) is generated upon write data into CSR3.
CSR2[8..15] - not used

3.4. CSR4

CSR4[7..0] – Delay of the L1A_Request signal from the backplane source (specified by CSR4[11..10]) before distribution to front panel connector P11-27/28 **(implemented only in firmware for the TF crate)**
CSR4[8] - disables (if "1") propagation of the TMB_L1A_Release from the backplane line to internal CCB logic
CSR4[9] - disables (if "1") propagation of the DMB_L1A_Release from the backplane line to internal CCB logic
CSR4[10] – enables (if "0") propagation of the TMB_L1A_Request (peripheral crate) or SP_L1A_Request (Track Finder crate) to front panel **(implemented only in firmware for the TF crate)**
CSR4[11] – enables (if "0") propagation of the TMB_L1A_Release (peripheral crate) or

MS_L1A_Request (Track Finder crate) to front panel **(implemented only in firmware for the TF crate)**

CSR4[15..12] – not used

3.5. CSR5

CSR5[7..0] - delay of L1ACC (any of 7 possible sources) before distribution to backplane. See 2.6. for more details.
CSR5[15..8] - delay of ALCT_external_trigger and CLCT_external_trigger sources before distribution to backplane. See 2.5 and 2.6 for more details.

3.6. CSR6

CSR6[8..0] - TMB_cfg_done[8..0] lines. Read only.
CSR6[9..15] - always "0".

3.7. CSR7

CSR7[8..0] - ALCT_cfg_done[8..0] lines. Read only.
CSR7[9..15] - always "0".

3.8. CSR8

CSR8[8..0] - DMB_cfg_done[8..0] lines. Read only.
CSR8[9] - MPC_cfg_done line. Read only.
CSR8[10..15] - always "0".

3.9. CSR9

CSR9[8..0] - CLCT_status[8..0] lines. Read only.
CSR9[9] - TTCrx_ready line. Read only.
CSR9[10 TTCrx_SingleErrStr line. Read only.
CSR9[11] TTCrx_DoubleErrStr line. Read only.
CSR9[12..15] - always "0".

3.10. CSR10

CSR10[8..0] - ALCT_status[8..0] lines. Read only.
CSR10[9..15] - always "0".

3.11. CSR11

CSR11[2..0] - DMB_reserved_in[2..0] lines. Read only.
CSR11[7..3] - TMB_reserved_in[4..0] lines. Read only.
CSR11[8..9] - FP_RSV<1..2> CCB front panel inputs. Read only.

CSR11[10..15] - always “0”.

3.12. CSR12

CSR12[11..0] - TTCrx_BCNT[11..0] lines (latched into flip-flops upon TTC_BCNTSTR strobe). Read only.
CSR12[12] - TTC_BCNTSTR. Read only.
CSR12[13..15] - always “0”.

3.13. CSR13

CSR13[11..0] - TTCrx_BCNT[11..0] lines (latched into flip-flops upon TTC_EVCNTLSTR strobe). Read only.
CSR13[12] - TTC_EVCNTLSTR. Read only.
CSR13[13..15] - always “0”.

3.14. CSR14

CSR14[11..0] - TTCrx_BCNT[11..0] lines (latched into flip-flops upon TTC_EVCNTHSTR strobe). Read only.
CSR14[12] - TTC_EVCNTHSTR. Read only.
CSR14[13..15] - always “0”.

3.15. CSR15

CSR15[3..0] - TTCrx_BRCST[5..2] lines (latched into flip-flops upon TTC_BRCSTSTR1 strobe). Read only.
CSR15[5..4] - TTCrx_BRCST[7..6] lines (latched into flip-flops upon TTC_BRCSTSTR2 strobe). Read only.
CSR15[6] - TTCrx_BRCSTSTR1 line (latched into flip-flops upon TTC_Clock40Des1). Read only.
CSR15[7] - TTCrx_BRCSTSTR2 line (latched into flip-flops upon TTC_Clock40Des1). Read only.
CSR15[8] - TTCrx_EVCNTRES line (latched into flip-flops upon TTC_BRCSTSTR1 strobe). Read only.
CSR15[9] - TTCrx_BCNTRES line (latched into flip-flops upon TTC_BRCSTSTR1 strobe). Read only.
CSR15[13..10] - TTCrx_DQ[3..0] lines (latched into flip-flops upon TTC_DOUTSTR strobe). Read only.
CSR15[14] - TTCrx_DOUTSTR line (latched into flip-flops upon TTC_Clock40Des1). Read only.
CSR15[15] - always “0”.

3.16. **CSR16**

CSR16[7..0] - TTCrx_DOUT[7..0] lines (latched into flip-flops upon TTC_DOUTSTR strobe). Read only.
CSR16[15..8] - TTCrx_SUBAD[7..0] lines (latched into flip-flops upon TTC_DOUTSTR strobe). Read only.

3.17. **CSR17** (date of the current firmware version)

CSR17[4..0] - Day (CSR17<0> is LSB and CSR17<4> is MSB). Read only.
CSR17[8..5] - Month (CSR17<5> is LSB and CSR17<8> is MSB). Read only.
CSR17[11..9] - Year (Add to 2000) (CSR17<9> is LSB, CSR17<11> is MSB).
CSR17[15..12] - Always "0". Read only.

4. **VME Interface**

The CCB can be accessed in the VME crate using either geographical or logical addressing mode (mode is selected by an on-board DIP switch). It recognizes AM codes 39(hex), 3A(hex), 3D(hex) and 3E(hex) and supports A24D16 slave operations. Geographical mode utilizes the geographical address pins GA[4-0] available on the VME64x backplane. In this mode the CCB recognizes its address space when the code on address lines A[23-19] is equal to the 5-bit geographical code of its slot. If S3-1 and S3-4 are connected, the logical address mode is selected. If S3-1 and S3-4 are disconnected, the geographical mode is selected. Base logical address C00000(hex) is used for initial testing and debugging (fixed in firmware). All decoded addresses are listed in Table 9.

Table 9

Address (hex)	Function
Base + 0	CSR1 (read/write)
Base + 2	CSR2 (read/write)
Base + 4	CSR3 (read/write)
Base + 6	CSR4 (read/write)
Base + 8	CSR5 (read/write)
Base + a	CSR6 (read only)
Base + c	CSR7 (read only)
Base + e	CSR8 (read only)
Base + 10	CSR9 (read only)
Base + 12	CSR10 (read only)
Base + 14	CSR11 (read only)
Base + 16	CSR12 (read only)
Base + 18	CSR13 (read only)
Base + 1a	CSR14 (read only)
Base + 1c	CSR15 (read only)
Base + 1e	CSR16 (read only)
Base + 20	I2C controller (read/write)
Base + 22	I2C controller (read/write)
Base + 24	Reset I2C controller (write only)
Base + 26	Reset TTCrx ASIC (write only)

Base + 28	Reset CCB internal logic (write only)
Base + 2a	Generate L1ACC 25 ns pulse (write only) and external triggers (if enabled)
Base + 2c	Generate TMB "Hard reset" 400 ns pulse (write only)
Base + 2e	Generate DMB "Hard reset" 400 ns pulse (write only)
Base + 30	Generate ALCT "Hard reset" 400 ns pulse (write only)
Base + 32	Generate MPC "Hard reset" 400 ns pulse (write only)
Base + 34	Generate "Hard Reset" 400 ns pulses to all modules in crate (write only)
Base + 36	Generate BC0 25 ns pulse (write only)
Base + 38	Generate 25 ns pulse to ccb_clock40 line (write only) if CSR1<1>=CSR1<2>=1
Base + 3a	
Base + 3c	Generate "Soft Reset" 25 ns pulse to all modules in crate (write only)
Base + 3e	Generate both "ALCT_adb_pulse_sync" and "ALCT_adb_pulse _async" (write only)
Base + 40	Generate "ALCT_adb_pulse_sync" 500 ns pulse (write only)
Base + 42	Generate "ALCT_adb_pulse_async" pulse (write only)
Base + 44	Generate "CLCT_external_trigger" 25 ns pulse (write only)
Base + 46	Generate "ALCT_external_trigger" 25 ns pulse (write only)
Base + 48	Generate "DMB_cfeb_calibrate[0]" 25 ns pulse (write only)
Base + 4a	Generate "DMB_cfeb_calibrate[1]" 25 ns pulse (write only)
Base + 4c	Generate "DMB_cfeb_calibrate[2]" 25 ns pulse (write only)
Base + 4e	Generate CSR9 reset (write only)
Base + 50	Generate CSR10 reset (write only)
Base + 52	Generate CSR11 reset (write only)
Base + 54	
Base + 56	
Base + 58	Generate "CCB_clock40_enable" pulse, (n) counts, defined by CSR4<7-0> (write only)
Base + 5a	Generate "DMB_cfeb_initiate" pulse (write only) (Hold next L1ACC and Pretriggers)
Base + 5c	Generate "Release HOLD L1ACC Mode" pulse (write only)
Base + 5e	CSR17 (Read only)
Base + 60	Generate "MPC_reserved[0]" 25 ns pulse (write only)
Base + 62	Generate "MPC_reserved[1]" 25 ns pulse (write only)
Base + 64	Generate "Soft Reset" 25 ns pulse to MPC (write only)
Base + 66	Generate "DMB_reserved[0]" 25 ns pulse (write only)
Base + 68	Generate "DMB_reserved[1]" 25 ns pulse (write only)
Base + 6a	Generate "Soft Reset" 25 ns pulse to all DMB (write only)
Base + 6c	Generate "DMB_reserved_out[0]" 25 ns pulse (write only)
Base + 6e	Generate "DMB_reserved_out[1]" 25 ns pulse (write only)
Base + 70	Generate "DMB_reserved_out[2]" 25 ns pulse (write only)
Base + 72	Generate "DMB_reserved_out[3]" 25 ns pulse (write only)
Base + 74	Generate "DMB_reserved_out[4]" 25 ns pulse (write only)
Base + 76	Generate "TMB_reserved_out[0]" 25 ns pulse (write only)
Base + 78	Generate "TMB_reserved_out[1]" 25 ns pulse (write only)
Base + 7a	Generate "TMB_reserved_out[2]" 25 ns pulse (write only)
Base + 7c	Generate "TMB_reserved[0]" 25 ns pulse (write only)
Base + 7e	Generate "Soft Reset" 25 ns pulse to all TMB (write only)
Base + 80	
Base + 82	Generate "CCB_reserved[1]" 25 ns pulse (write only)
Base + 84	Generate "CCB_reserved[2]" 25 ns pulse (write only)
Base + 86	Generate "CCB_reserved[3]" 25 ns pulse (write only)
Base + 88	Generate L1 Reset 25 ns pulse (write only)
Base + 8a	Generate "FP_RSV1_OUT" 25 ns pulse to front panel (write only)
Base + 8c	
Base + 8e	
Base + 90	

Base + 92	
Base + 94	
Base + 96	Read L1ACC Counter[15..0]
Base + 98	Read L1ACC Counter[31..16]
Base + 9a	Reset L1ACC Counter to 0 (write only)
Base + 9c	Enable L1ACC Counter to count (write only)
Base + 9e	Disable L1ACC Counter to count (write only)

5. I2C Programming

There is one device (TTCrx mezzanine card) in an I2C chain on the CCB board **(Note: as of June 2002, it was decided to remove all PHOS4 chips from the CCB board for more precise clock distribution over backplane, so the description of the PHOS4 programming over I2C bus was removed from this manual)**. The Philips PCF8584 [6] I2C controller provides an access to all these devices over two signal lines: SDA (data) and SCL (clock). Controller itself can be programmed via VME (see Table 9). An 8 Mhz reference clock for the PCF8584 controller is provided from the main PLD. PCF8584 controller operates in a "68000" mode (see [6], page 6).

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 at [5]. 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 10, based on Chapter 7 of document [5]. ID_I2C[5..0] bits are defined by the switches on Dout<5..0>lines on the TTCrx board.

The TTCrx is accessible over I2C bus only if optical connection to TTC transmitter is established and the ttcrx_ready = "1".

Table 10

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

6. Front Panel

There are three 34-pin connectors (0.1 x 0.1 shrouded headers) on the CCB front panel that provide differential ECL inputs and outputs to/from the CCB (see Tables 11-13) . All input signals are expected in positive logic. All output signals are in negative logic.

Table 11. Input connector P10

Pin	Signal	Pin	Signal
1	External_clock40+	2	External_clock40-
3	External_clock40_enable+	4	External_clock40_enable-
5	External_l1accept+	6	External_l1accept-
7	Dmb_cfeb_calibrate[0]+	8	Dmb_cfeb_calibrate[0]-
9	Dmb_cfeb_calibrate[1]+	10	Dmb_cfeb_calibrate[1]-
11	Dmb_cfeb_calibrate[2]+	12	Dmb_cfeb_calibrate[2]-
13	Alct_adb_pulse_sync+	14	Alct_adb_pulse_sync-
15	Alct_adb_pulse_async+	16	Alct_adb_pulse_async-
17	Clct_external_trigger+	18	Clct_external_trigger-
19	Alct_external_trigger+	20	Alct_external_trigger-
21	FP_BC0+	22	FP_BC0-
23	FP_BCNTRES+	24	FP_BCNTRES-
25	FP_Hard_Reset+	26	FP_Hard_Reset-
27		28	
29		30	
31		32	
33		34	

Table 12. Output connector P11

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	Ccb_bc0+	22	Ccb_bc0-
23	Ccb_l1accept+	24	Ccb_l1accept-
25	Ccb_cmdstr+ (BC0+ from SP in the Track Finder Crate)	26	Ccb_cmdstr- (BC0- from SP in the Track Finder Crate)
27	Ccb_fp_reserved_out[0]+ (L1A_Request+ in the Track Finder Crate)	28	Ccb_fp_reserved_out[0]- (L1A_Request- in the Track Finder Crate)
29		30	
31		32	
33		34	

Table 13. Output connector P12

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	

There are 12 LEDs on the front panel:

- +3.3V (D9), +5V (D11), -5.2V (D7) Power (green)
- "L1A" (D3) L1ACCEPT (red, with one-shot, from any of 7 possible sources. Active if L1ACC was sent to backplane)
- "BX0" (D4) BX0 (red, with one-shot)
- "HR" (D5) Hard Reset (any of TMB_HR, DMB_HR, ALCT_HR, MPC_HR, general
- HR) (red, with one-shot)
- "I2C" (D6) indicates access to I2C controller (red, with one-shot)
- "Mode" (D8) indicates CCB operation mode or state of CSR1<0> ("on" if "TTCrx" mode, "off" if "VME" mode), yellow, static signal
- "VME" (D2) indicates VME access to CCB (yellow, with one-shot)
- "TTC_Ready" (D10) indicates that TTCrx is in normal operation mode (green)
- "SinEr" (D13) indicates single error signal from TTCrx (red)
- "DbEr" (D14) indicates double error signal from TTCrx (red)

7. Board Initialization

After power up the following procedures are necessary:

1. Reset CCB internal logic.
2. Program all CSR's with required values. If CSR4 will be used, it should have a non-zero value.

8. Mechanical Parameters

The CCB is designed as a VME 9U*400 mm board (see block diagram on Fig.1). A TTCrx mezzannine card is mounted on two 50-pin headers. The main CCB logic (including VME interface) is designed in a single PLD (Altera EPF10K100ABC356-1). This PLD as well as a configuration EPROM (Altera EPC2LC20) and a 10-pin header for ByteBlaster downloading cable are located on a separate mezzanine card ~10*10 cm in size. This mezzanine card is attached to four Samtec OPC-150-T-D connectors mounted on the main CCB board. Such a configuration will easily allow us to replace the mezzanine card by another one with radiation hard version of PLD if necessary.

Both PLD and EPROM can be programmed/configured over JTAG cable compatible with Altera Bit- or ByteBlaster. Two DIP switches S1 and S2 define the configuration of the JTAG chain (see Table 14).

Table 14. JTAG chain configuration

S1 1-4	S1 2-3	S2 1-4	S2 2-3	Configuration
on	off	on	off	EPROM
off	on	off	on	PLD
All others				prohibited

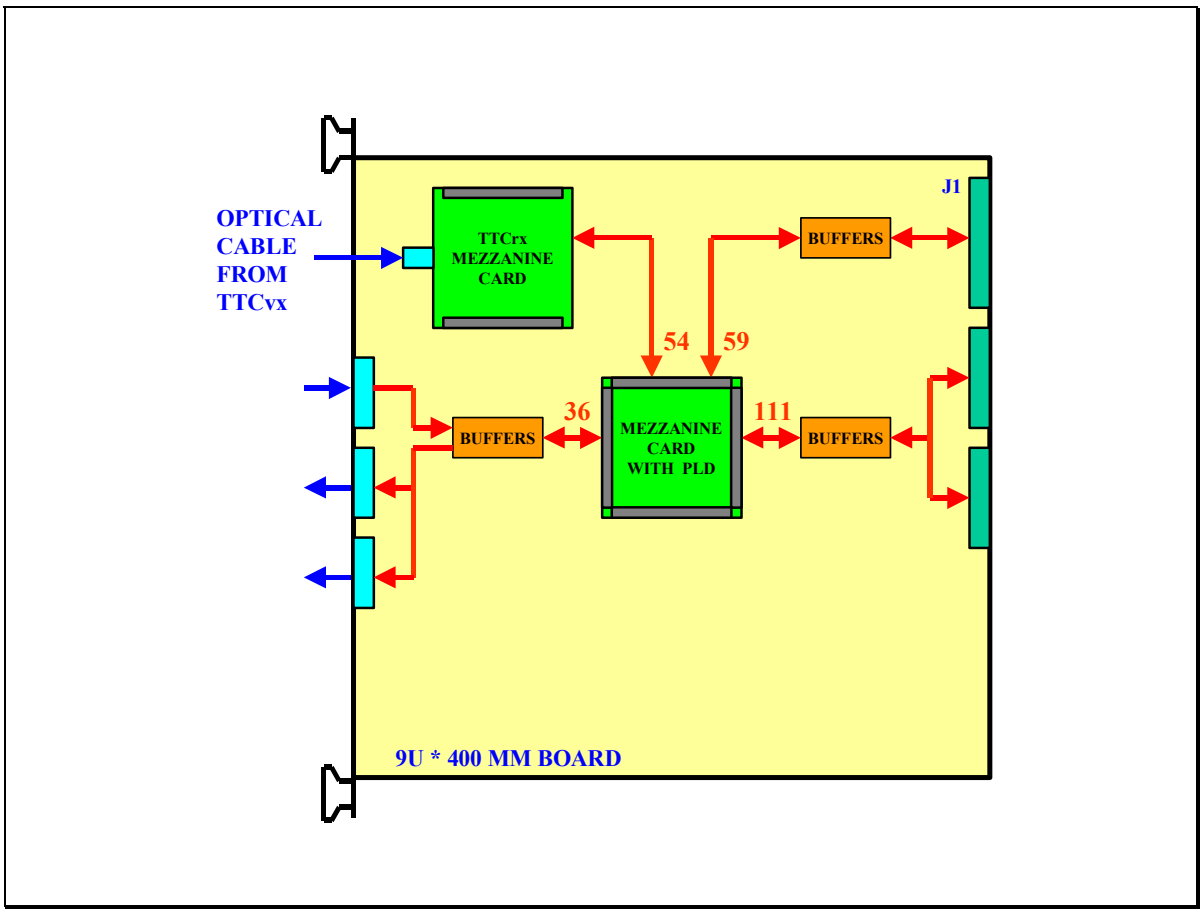

Figure 1: Block Diagram of the Clock and Control Board

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]. http://www.phys.ufl.edu/~madorsky/backplane/
[3]. http://www.phys.ufl.edu/~madorsky/TrackFinder/TF backplane v4.doc
[4]. B.G.Taylor. Timing, Trigger and Control (TTC) System for LHC Detectors. CERN/ECP http://www.cern.ch/TTC/intro.html
[5]. J.Christiansen, A.Marchioro, P.Moreira and T.Toifl. TTCrx Reference Manual. February 2003. Version 3.8. January 2003. CERN-ECP/MIC, Geneva, Switzerland.
[6]. Philips Data Handbook IC12. 1997. I2C Peripherals.

History

03/05/2001: Table 3 and section "I2C Programming" were added.
03/08/2001: Addition to 2.6 and 2.8.
03/16/2001: Table 5 is completed.
04/24/2001: Table 9 is fixed. Addition to 2.8.
05/11/2001: Changes in Table 9. Minor changes in CSR12-CSR14.
08/29/2001: Update of Table 9.
09/18/2001: Table 13 was added.
11/26/2001: ccb_rsv[0] signal is assigned to TTCrx_ready static signal. Programming details related to TTCrx were added.
12/07/2001: clarification on I2C programming (example in Section 5).
12/12/2001: An error in I2C programming (section 5) was fixed.
01/03/2002: CSR1<9-10> bits were defined.
02/07/2002: Changes in Table 5. Soft_reset was added. Appendix A was added.
02/15/2002: Ccb_reserved[4] line was renamed to ccb_l1reset line.
03/25/2002: Minor changes in Section 7.
05/15/2002: Changes in 2.6. More options added to generate anode and cathode Pretriggers and L1ACC. Added the independent programmable delay for Pretriggers. Added the 32-bit counter of L1ACC requests.
07/16/2002: CSR17 was added.
08/05/2002: PHOS4 programming information was removed. Three signals were renamed in Table 9.
08/27/2002: CSR4<8-9> were added.
09/18/2002: The ALCT_adb_pulse_sync was expanded to 500 ns before distribution to backplane. The content of the L1ACC counter was latched into two 16-bit temporary registers before reading over VME.
10/24/2002: New TTC and VME commands were added to generate both "ALCT_adb_pulse_sync" and "ALCT_adb_pulse_async".
11/13/2002: The CCB functionality for the Track Finder crate was added.
12/06/2002: Update of the CCB functionality for the Track Finder crate.
04/03/2003: BX0 replaced by BC0.
04/13/2003: Update of section 5 (I2C Programming).
11/13/2003: Base+80 VME write command was removed from Table 9.
08/04/2006: Description of bits CSR4[7-0] and CSR4[11-10] was added in Section 3.4