TCPU Engineering User Manual:

Table of Contents

Change history

Version 4_4: Feb 7, 2008:

- Amended for current BNL firmware (J. Schambach)
- Deleted material inconsequential for operations at BNL (J. Schambach)

Version 4_2 : Sept 29, 2007

- New information on Aux data path usage.
- Additional examples of CAN bus messages as Appendix C.
- Included formerly separate firmware programming instructions (using programming headers) as Appendix B.

Version 4_1: Sept 24, 2007

- Updated configuration register table.
- Added “useful FPGA configuration bits” section, with an initial description of bunch reset signal.

Connector Summary

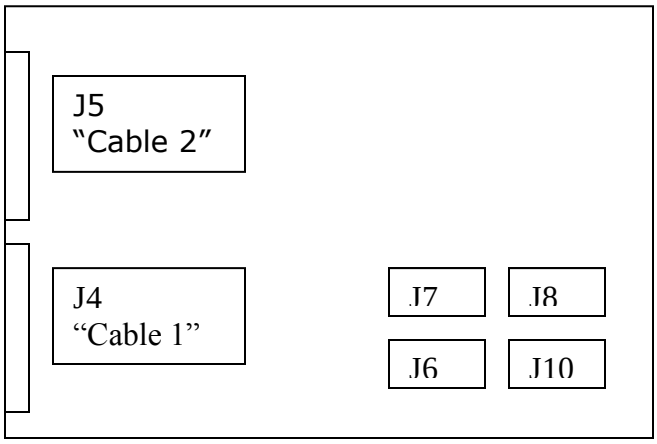
Power :

J26= +4.5V
J25 = GND

Jumper settings:

Jumper JU2: Short 1-2 = external oscillator
Open 1-2 = internal oscillator

Programming headers and locations (TOP VIEW):

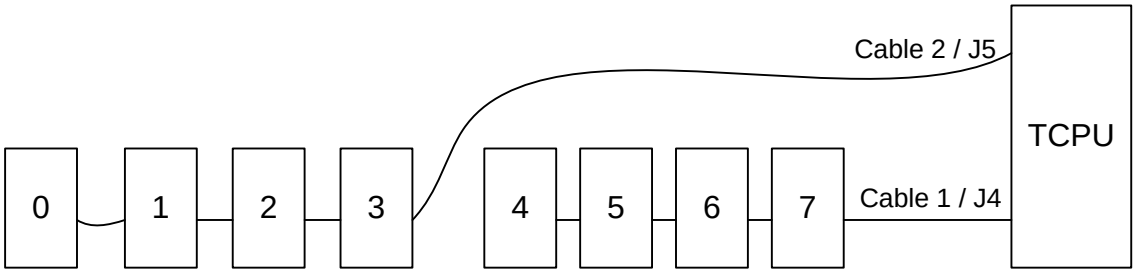


J7: Byteblaster JTAG programming of CPLD
J8: Byteblaster active serial programming of FPGA (EEPROM #1)
J6: Byteblaster JTAG debugging of FPGA
J10: ICD2 programming of MCU

J5 : Downstream cable 2
J4: Downstream cable 1

J11: Always jumper 1-2

TDIG / TDC numbering



	TDIG	TDIG	TDIG	TDIG	TDIG	TDIG	TDIG	TDIG	TCPU
Board position	0	1	2	3	4	5	6	7	
HPTDC ID	0, 1, 2	4,5,6	8,9,A	C,D,E	0, 1, 2	4,5,6	8,9,A	C,D,E	
Readout order	1	2	3	4	1	2	3	4	
Cable	2 (J5)	2	2	2	1 (J4)	1	1	1	

Useful FPGA configuration bits

Bunch reset CONFIG_14.4 This bit is mapped directly onto the output signals C1_BUNCH_RST and C2_BUNCH_RST. To issue a bunch reset signal, send a can bus message that will toggle the contents of this bit low-high-low. Default value is low. Note that CONFIG_0.3 must be LOW (default), otherwise test counter data will be sent to these signals.

Select Aux readout CONFIG_14.7 This bit selects between NORMAL (FPGA involvement) and AUX data readout paths. Value = 0 is default and selects NORMAL. Value = 1 selects AUX. To write a "1" to this bit (and "0" to all other bits in FPGA register 14, use "002 3 0E 0E 80" (assuming TCPU address is set to 0). Note that any other bits that need to be "1" must also be set in the value to be written to the resgister.

Aux data Readout

To use the AUX readout path, you must:

1. Set HPTDC configuration bit #44 "SELECT BYPASS INPUTS" to "1" in all TDCs in the readout chain.

2. Set TCPU FPGA configuration register bit CONFIG_14.7 = "1"
3. Unless power is cycled, a reset to the TCPU state machines, and possibly the HPTDCs, will probably be necessary when switching between modes (toggle CONFIG_1.0 HI then LO ("002 5 0E 01 00 01 00").
4. Set jumper J17 properly on each TDIG in the readout chain for Aux data path token routing – Jumper 2-3 if board is farthest from TCPU; otherwise jumper 1-2.

Configuration / Status Registers in FPGA:

MCU writes to:

Byte address	Register Name	Bit Position	Description	Default Value
0	CONFIG_0			
		0	SELECT TEST INPUT TO MCU FIFO	NOT IMPLEMENTED
		1	SELECT CABLE_2 (J5) INPUT TO MCU FIFO	0 = cable1 (J4) 1 = cable2 (J5) input NOT IMPLEMENTED
		2	SEL THUB DATA TO MCU	
		3	SEL TEST COUNTER DATA ON DOWNSTREAM CABLES	NOT IMPLEMENTED
		4	SEL TEST MODE FOR TDC TRIGGER	NOT IMPLEMENTED
		5	SEL TEST MODE FOR TDC TRIGGER	NOT IMPLEMENTED
		6	SEL TEST MODE FOR TDC BUNCH RESET	NOT IMPLEMENTED
		7	SEL TEST MODE FOR TDC EVENT RESET	NOT IMPLEMENTED
	CONFIG_1			
		0	RESET FPGA STATE MACHINES AND MCU FIFOS	0
		1	Clear FIFOs	0
		2		
		3		
		4		
		5		
		6		
		7		
	CONFIG_2	0	Enable Readout	0
		1	Trigger source	0 = test (def) 1 = Serdes
		3	Disable MCU FIFO write	0 = do write(def) 1 = don't write
		4	Enable Serdes Tx/Rx	0 = disabled 1 = enabled
		5		
		6		
		7		
	CONFIG_3	0	Serdes Hardware lock (active lo)	R/O
		1	Serdes Ready (firmware)	R/O

		2		
		3		
		4		
		5		
		6		
		7		
	CONFIG_14			
		2:0	SELECT ONE OF 8 FREQUENCIES FOR INTERNAL TEST PULSE GENERATOR (USED FOR CAN-BASED RDO)	000 = 40HZ (001 = 80 HZ, ETC)
		3	SELECT INTERNAL TEST PULSER	0 = external pulser from J9 input 1 = internal (FPGA) pulser
		4	BUNCH RESET – this register bit value is mapped directly to bunch reset outputs on cable 1 and cable 2	0 = DISABLED
		5	ENABLE TEST TRIGGER	0 = DISABLED
		6	ENABLE TEST "GET DATA" PULSE	0 = ENABLED
		7	SELECT AUX SERIAL READOUT	1 = aux readout 0 = normal readout

Appendix A : TCPU Interface Document

Version: 11.0

Changes:
11 May: Fixed J21 pinout definition

Interface Index

1. CAN Bus

TCPU has 2 independent CAN bus controllers with associated transcievers and cable connectors. The **external CAN** connects between trays and THUB. The **internal CAN** connects between TCPU and TDIG boards on the same tray.

External CAN network connection (J21):

Connector: 3 pin "Mate N Lock" AMP # 1-350943
Location: bottom side of board
Reference designator: J21
Cable plug AMP # 350766
Termination: Short jumper at JU4 or microcontroller closure of U59 switch

Pinout for J21:

PIN	SIGNAL
1	CANL
2	GND
3	CANH

Internal CAN network connection (J12, J4, J5):

Pinout for J12:

PIN	SIGNAL
1	CANL
2	GND
3	CANH

Test header connections
Connector 1: 3 pin vertical header
Location: Top side of board
Cable plug at J12 : straight pin header cable for monitoring and troubleshooting, not part of the installed configuration
Also connected to **tray cable connections** (also see section 7: "Tray interface")

1. J4/ pin 14 (CANH), pin 13 (CANL)
2. J5/ pin 14 (CANH), pin 13 (CANL)

Termination

2. Tray voltage Sense

3 Pin "Micro Mate-N- Lock" AMP # 1445098-3
Cable plug = AMP # 1445022-3

Pinout: 2 leads + shield To be connected before the fuse on TCPU,
Also, for safety reasons in case of cable damage, this "output" should be current limited (1/4 amp?) with a self-resetting fuse. A "pig-tail" section of cable and an additional in-line version of the above in-line connectors has been suggested to avoid having to make the "sense" section of the "integrated". cable longer than the L.V. section

3. Multiplicity

"Right angle shrouded pin header" AMP # 499913-2
Ribbon cable plug = 3M #3385-6000 (14 pin)

Voltage level: LVPECL
Chipset: Texas Instruments TB5D1M (Quad Differential PECL Driver)

Pinout (suggestion):

PIN	SIGNAL
1	Bit1+
2	Bit1-
3	Bit2+
4	Bit2-
5	Bit3+
6	Bit3-
7	Bit4+

8	Bit4-
9	Bit5+
10	Bit5-
11	RHIC Clock +
12	RHIC Clock -
13	GND
14	GND

4. Clock Distribution:

In case SERDES clock distribution doesn't work will use this second means as clock input to TCPU:
Clock frequency: 40MHz
"RJ45" Molex # 95001-2881
LVPECL signal levels

Pinout (suggestion):

PIN	**SIGNAL**
7	Clock+
8	Clock-
9	GND (shield)
10	GND (shield)

Clock distribution cable: CAT-6 shielded wire
What chip set do we want to specify here (transmitter and receiver)?

Alternatively: use differential coax? How do we make this decision?

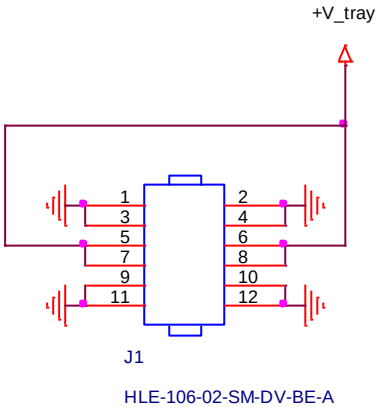
5. Low Voltage power input

Two L.V. "Screw Terminals" Keystone # 1202 or 8195
Wire Lugs = MNG10-10-FBX

6. Aux L.V. Out

for Alternate Multiplicity "TTRIG" board

Proposed connector is Samtec: HLE-106-02-SM-DV-BE-A to be placed on top side of TCPU, to accept pins from below . See Pinout and Footprint attached separately



7. Tray Interface

J4 and J5 are each 50 pin ribbon cable connectors which carry signals between TCPU and 2 sets of TDIG cards.

The physical location and location labeling of boards on a detector tray is as follows:

TDIG0 TDIG1 TDIG2 TDIG3 TDIG4 TDIG5 TDIG6 TDIG7 TCPU

Daisy-chained tray cabling is arranged physically and logically as follows:

TDIG0 == TDIG1 == TDIG2 == TDIG3 == TCPU / J5

TDIG4 == TDIG5 == TDIG6 == TDIG7 == TCPU / J4

Each TDIG board has

- An “upstream” connector TDIG / J5, physically closest to TCPU (on the right hand side of each TDIG in the diagrams above)
- A “downstream” connector TDIG / J4, physically farthest from TCPU (on the left hand side of each TDIG in the diagrams above)

Each TCPU board has 2 “downstream” connectors, TCPU / J4 and TCPU / J5 (both on the left hand side of TCPU in the diagrams above). The signal assignments for TCPU / J4 and TCPU / J5 are given in the tables below.

CABLE SIGNAL	CABLE PINS	BOARD SIGNAL	DIRECTION
C1_DCLK_40MHZ	3,4	C1_DCLK_40MHZ	DOWNSTREAM
TRAY_CAN	13,14	TRAY_CAN	BIDIRECTIONAL
C1_DLV_0	7,8	C1_SER_OUT	UPSTREAM
C1_DLV_1	9,10	C1_TOKEN_OUT	UPSTREAM
C1_DLV_2	11,12	C1_STROBE_OUT	UPSTREAM
C1_DLV_3	15,16	C1_DDAISY_DATA	UPSTREAM
C1_DLV_4	17,18	C1_DDAISY_TOK_OUT	UPSTREAM
C1_DLV_5	19,20	C1_DSTATUS0	UPSTREAM
C1_DLV_6	21,22	C1_DSTATUS1	UPSTREAM
C1_DLV_7	23,24	C1_DDAISY_CLK	DOWNSTREAM
C1_DLV_8	25,26	C1_DDAISY_TOK_IN	DOWNSTREAM
C1_DLV_9	27,28	C1_MCU_RESETB	DOWNSTREAM
C1_DLV_10	29,30	C1_FLEX_RESET_OUT	DOWNSTREAM
C1_DLV_11	31,32	C1_TRAY_TOKEN	DOWNSTREAM
C1_DLV_12	33,34	C1_DSPARE_OUT1	DOWNSTREAM
C1_DLV_13	35,36	C1_BUNCH_RST	DOWNSTREAM
C1_DLV_14	37,38	DCONFIG_OUT	DOWNSTREAM
C1_DLV_15	39,40	C1_TRIGGER	DOWNSTREAM
C1_DLV_16	41,42	C1_CLK_10MHZ	DOWNSTREAM
C1_DLV_17	43,44	C1_DMULT0	UPSTREAM
C1_DLV_18	45,46	C1_DMULT1	UPSTREAM
C1_DLV_19	47,48	C1_DMULT2	UPSTREAM
C1_DLV_20	49,50	C1_DMULT3	UPSTREAM

Table 1 : J4 Signal Assignments

CABLE SIGNAL	CABLE PINS	BOARD SIGNAL	DIRECTION
C2_DCLK_40MHZ	3,4	C2_DCLK_40MHZ	DOWNSTREAM
TRAY_CAN	13,14	TRAY_CAN	BIDIRECTIONAL
C2_DLV_0	7,8	C2_SER_OUT	UPSTREAM
C2_DLV_1	9,10	C2_TOKEN_OUT	UPSTREAM
C2_DLV_2	11,12	C2_STROBE_OUT	UPSTREAM
C2_DLV_3	15,16	C2_DDAISY_DATA	UPSTREAM
C2_DLV_4	17,18	C2_DDAISY_TOK_OUT	UPSTREAM
C2_DLV_5	19,20	C2_DSTATUS0	UPSTREAM
C2_DLV_6	21,22	C2_DSTATUS1	UPSTREAM
C2_DLV_7	23,24	C2_DDAISY_CLK	DOWNSTREAM
C2_DLV_8	25,26	C2_DDAISY_TOK_IN	DOWNSTREAM
C2_DLV_9	27,28	C2_MCU_RESETB	DOWNSTREAM
C2_DLV_10	29,30	C2_FLEX_RESET_OUT	DOWNSTREAM
C2_DLV_11	31,32	C2_TRAY_TOKEN	DOWNSTREAM
C2_DLV_12	33,34	C2_DSPARE_OUT1	DOWNSTREAM
C2_DLV_13	35,36	C2_BUNCH_RST	DOWNSTREAM
C2_DLV_14	37,38	DCONFIG_OUT	DOWNSTREAM
C2_DLV_15	39,40	C2_TRIGGER	DOWNSTREAM
C2_DLV_16	41,42	C2_CLK_10MHZ	DOWNSTREAM
C2_DLV_17	43,44	C2_DMULT0	UPSTREAM
C2_DLV_18	45,46	C2_DMULT1	UPSTREAM
C2_DLV_19	47,48	C2_DMULT2	UPSTREAM
C2_DLV_20	49,50	C2_DMULT3	UPSTREAM

Table 2: J5 Signal Asignments

8. Tray geographic ID

Each TCPU has two rotary switches for manual setting of tray geographic ID: SW5 and SW4. Each switch has 10 positions, 0 to 9, which should be used to set the tray geographic ID as a 2 digit decimal number – SW5 is the 10's digit and SW4 is the 1's digit.

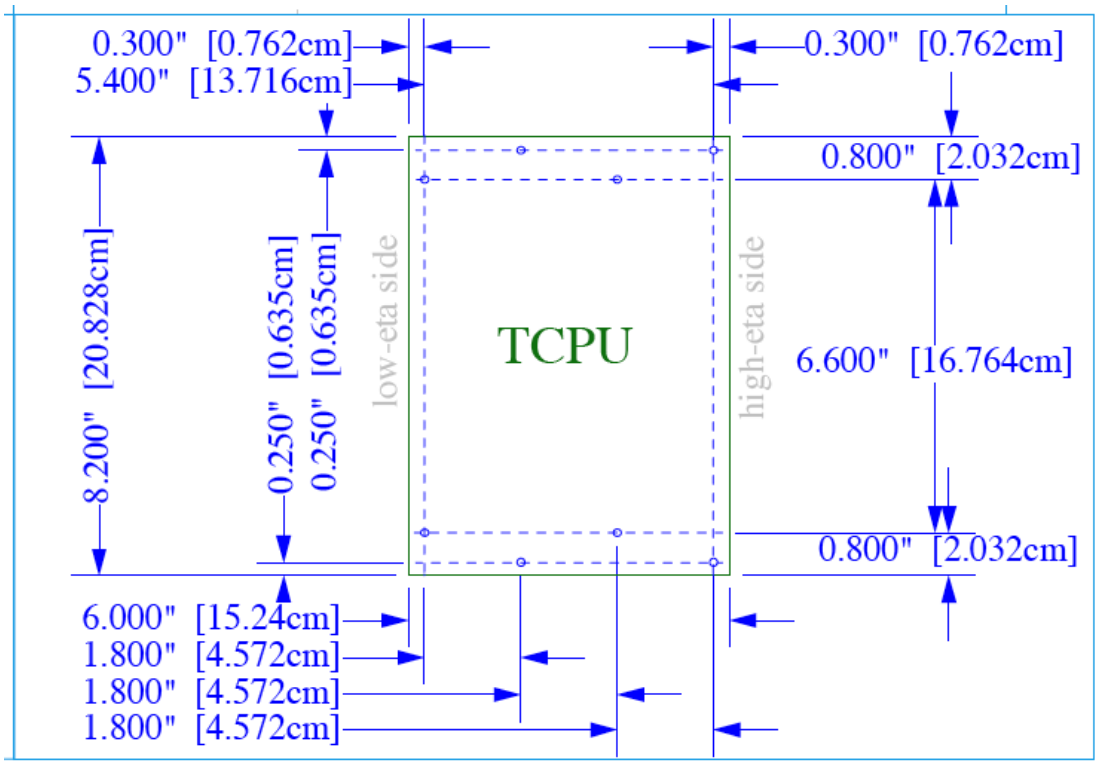
Example: For tray geographic ID = 25, set SW5 = 2 and SW4 = 5.

The binary coded decimal value from SW4 and SW5 is transferred to the MCU and is available for use in the CAN and data transfer protocols, in the format specified in those protocols. Will be used to determine TCPU CANbus nodeID.

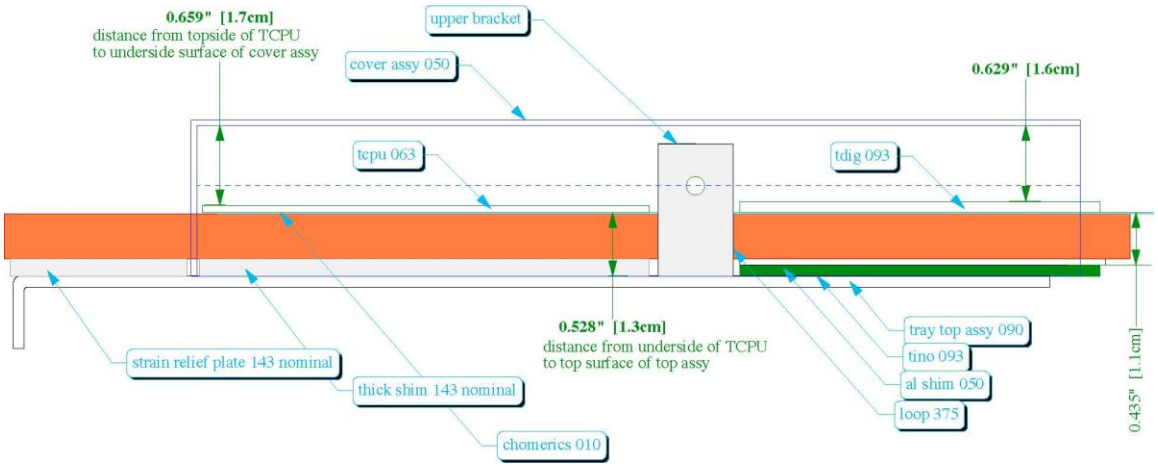
Tray geographic ID needs to be available to FPGA for inclusion in geographic data words defined below.

9. Mechanical Dimensions

Tray integration dimensions:



Clearance from top of TCPU to underside surface of cover assembly = 0.629 inches.



10. Serial THUB link

J18
RJ45 connector part #AMP 5555153-1

PIN	SIGNAL
1	RIN+ (receiver)
2	RIN- (receiver)
3	DO+ (transmitter
4	No connect
5	No connect
6	DO- (transmitter
7	Clock + (LVPECL) “CLK1/CLK1B” signal pair
8	Clock – (LVPECL) “CLK1/CLK1B” signal pair
9	GND (shield)
10	GND (shield)

SERDES chip used will be National DS92LV18.
Receiver needs 100 Ohm termination across the receiver lines
Clock used will be 20 MHz. Use PLL to clean up and multiply to 40 MHz on RCLK. Can possibly be used then as input clock to TCPU.
Rev 2: Tests with the THUB prototype showed that the clock is not constant in phase between power-cycles, so it can’t be used. So we added a 40MHz LVPECL clock on pins 7 and 8 instead for this revision to test as possible candidate of TCPU master clock.
Separate clock inputs:

J19
RJ45 connector part #AMP 5555153-1

PIN	SIGNAL
1	ext test 1
2	ext test 2
3	ext test 3
4	ext test 4
5	No connect
6	No connect
7	Clock + (LVPECL) "CLK3/CLK3B" signal pair
8	Clock – (LVPECL) "CLK3/CLK3B" signal pair
9	GND (shield)
10	GND (shield)

Ext test [4:1] are routed directly to the FPGA

J23, J22

SMB connector
J23 Clock+ (LVPECL) "CLK2/CLK2B" signal pair
J22 Clock- (LVPECL) "CLK2/CLK2B" signal pair

Only 1 of the 3 clock signal pairs can be active at a given time. Input pairs can be permanently disabled by removing the AC coupling capacitors from their signal path.

11. Data Communication : TCPU <-> THUB

TCPU I/O using FPGA and DS92LV18T (U66 ser/deser device)

TCPU outputs (to DIN[17:0] on U66) are labeled as “TX” for transmit.
TCPU inputs (from ROUT[17:0] on U66) are labeled as “RX” for receive.

All operations on U66 ser/deser data words (DIN / TX [17:0] and ROUT / RX [17:0]) assume the following bit ordering conventions:

LOGICAL BIT NAME on TCPU	BIT #	PIN #	BIT ORDER
TX_D0	0	21	LSB
TX_D1	1	22	
TX_D2	2	23	
TX_D3	3	24	
TX_D4	4	25	
TX_D5	5	26	
TX_D6	6	27	
TX_D7	7	28	
TX_D8	8	33	
TX_D9	9	34	
TX_D10	10	35	
TX_D11	11	36	
TX_D12	12	37	
TX_D13	13	38	
TX_D14	14	39	
TX_D15	15	40	
TX_D16	16	18	
TX_D17	17	3	MSB

LOGICAL BIT NAME on TCPU	BIT #	PIN #	NOTES
RX_D0	0	45	LSB
RX_D1	1	46	
RX_D2	2	47	
RX_D3	3	48	
RX_D4	4	54	
RX_D5	5	55	
RX_D6	6	56	
RX_D7	7	57	
RX_D8	8	64	
RX_D9	9	65	
RX_D10	10	66	
RX_D11	11	67	
RX_D12	12	70	
RX_D13	13	71	
RX_D14	14	72	
RX_D15	15	73	
RX_D16	16	62	
RX_D17	17	80	MSB

Messages from THUB transmitter to TCPU reciever

Messages from THUB to TCPU shall consist of 18-bit words.

The format for these **THUB command words** is shown in the table below:

Bit position	Field
17	Data type 1
16	Data type 0
15	Data ID 3
14	Data ID 2
13	Data ID 1
12	Type ID 0
11	Data 11
10	Data 10
9	Data 9
8	Data 8
7	Data 7
6	Data 6
5	Data 5
4	Data 4
3	Data 3
2	Data 2
1	Data 1
0	Data 0

Field Definitions:

Data type [1:0] values may be

[00] invalid data
[11] valid data, odd parity
[10] valid data, even parity
[01] control word

Data ID [3:0] value definitions depend on Data type values:

1. For Data type = [11] or [10] (data word),

Data ID values may be:

[0000] = trigger token data, phase A (and issue trigger)
[0001] = trigger token data, phase B (and issue trigger)
[1100] = test data 0
[1101] = test data 1

2. For Data type = [01] (command word),

Data ID values may be:

[0000] = go to test mode 0
[0001] = go to test mode 1
[0010] = issue bunch reset
[0011] = reset
[0100] = abort upload

Data [11:0] is valid for data words only (Data ID = [11] or [10]) and otherwise should be set to 0.

Messages from TCPU transmitter to THUB receiver

Messages from TCPU to THUB shall consist of groups of at least 3 18-bit words.

The first word(s) shall function as a synchronization word (invalid data) and may be repeated. This word shall have [00] in bit positions 17 and 16 to indicate invalid data.

Following the synchronization word(s) shall always be exactly 2 TCPU data words.

The format for these **TCPU data words** is shown in the table below:

First TCPU data word	
Bit position	Field
17	Data type 1
16	Data type 0
15	Data ID 3
14	Data ID 2
13	Data ID 1
12	Data ID 0
11	Data 27
10	Data 26
9	Data 25
8	Data 24
7	Data 23
6	Data 22
5	Data 21
4	Data 20
3	Data 19
2	Data 18
1	Data 17
0	Data 16

SECOND WORD	
Bit position	Field
17	Data Type 1
16	Data Type 0
15	Data 15
14	Data 14
13	Data 13
12	Data 12
11	Data 11
10	Data 10
9	Data 9
8	Data 8
7	Data 7
6	Data 6
5	Data 5
4	Data 4
3	Data 3
2	Data 2
1	Data 1
0	Data 0

Field Definitions:

Data type [1:0] values may be

[11] valid data, odd parity
[10] valid data, even parity
[00] synchronizer word – data ID and data values are ignored
[01] control word – TBD (to preserve symmetry)

Data ID [3:0] values may be

[0 to 7] = HPTDC data types

[1000] = status/error data
[1001] = multiplicity data
[1010] = trigger token return
[1011] = test data
[1100] = geographic data
[1101] = tag data (not used in TCPU-THUB communications)
[1110] = separator data

Data [23:0] values are formatted according to Data ID

For Data ID from 0 to 7, data bits are defined according to the HPTDC reference manual.

Trigger token return:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0																	token											

Empty fields are reserved for higher level processing in the THUB and should be set to all 0's.

Geographic Data:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0																					tray ID							H

Table 3: Geographical Data Word

where:

- "*tray ID"* is a number between 1 and 120, uniquely identifying each tray according to the document "STAR Geometry" (STAR Note 229),
- "*H"* is either 0 or 1, identifying a *half tray* corresponding to 4 TDIG boards; TDIG boards 0 – 3 correspond to *H = 0,* while TDIG boards 4 – 7 correspond to *H = 1.*
- Empty fields are reserved and should be set to all 0's
- For Run 8, we just send "H", "tray ID" is set to "0" and filled in by THUB

Separator Word:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0									number of items												trigger counter							

where:

- *"number of items"* is the number of preceding HPTDC words from a readout chain (currently limited to 10 bits).
- *"trigger counter"* are the 8 least significant bits of an internal counter on TCPU that keeps track of how many trigger commands were received from THUB.

Example Communications:

At the beginning, sometime after startup, the first message will be a bunch reset message from THUB to all TCPUs:

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	C	C	C	C	X	X	X	X	X	X	X	X	X	X	X	X

where:
DATA TYPE = 01 = CONTROL DATA
CCCC = DATA ID = bunch reset command
XXxx = 0000 0000 0000

The run can then start, and whenever THUB receives a trigger, it sends a trigger word to TCPU as follows:

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT1 = 1	DT0 = PARITY	C	C	C	C	T	T	T	T	T	T	T	T	T	T	T	T

where:
DATA TYPE = 10 | 11 = [11] valid data, odd parity **OR** [10] valid data, even parity

CCCC = 0000 | 0001 indicating phase
TTTT = 12 bit TCD Token

The TCPU then issues a trigger to all TDIG boards either in the 40Mhz phase that coincides with the incoming 20Mhz rising clock edge, or in the 40Mhz phase that coincides with the incoming 20Mhz trailing edge, according to the phase information contained in this packet.

After issuing the trigger, the TCPU will attempt to readout the two serial readout chains from the tray, and then start sending data to THUB according to a sequence defined below, two 18 bit words at a time:

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	p	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	p	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

where:
1p = DATA TYPE
D[31:0] = data word bit
00 = invalid data

The sequence of (32-bit) data words sent from TCPU to THUB is then defined as follows:

Word #:	**Content:**
1	Header Trigger Data (containing TCD Token) (not in run 8)
2	Geographical Data for first half tray
3	HPTDC data words from 12 HPTDCs (TDIG 0 – 3)
…	…
3+n	HPTDC data words from 12 HPTDCs (TDIG 0 – 3)
3+n+1	TDIG separator word containing # of preceding HPTDC data
3+n+2	Geographical Data for second half tray
3+n+3	HPTDC data words from 12 HPTDCs (TDIG 4 – 7)
…	…
m	HPTDC data words from 12 HPTDCs (TDIG 4 -7)
m+1	TDIG separator word containing # of preceding HPTDC data
m+2	Multiplicity data bits 0 – 27
m+3	Multiplicity data bits 28 – 56
m+4	Multiplicity data bits 56 – 85
m+5	Multiplicity data bits 86 – 113
m+6	Multiplicity data bits 114 – 141
m+7	Multiplicity data bits 142 – 169
m+8	Multiplicity data bits 170 – 192, left justified with 0's

The format of the Multiplicity data is simple: one bit per hit MRPC channel, as inferred from the HPTDC data. In version 0 of the firmware, the multiplicity data will not be sent by the TCPU and the data transmission terminates after word "m+1" in the above table. The sequence above shows where future versions of the firmware will insert this multiplicity information.

In case of errors or timeouts we will use "Control" packets (which I haven't defined, yet) instead of the data words shown above. Those "Control" packets could again be single 18 bit transmissions (instead of two 18 bit transmissions, as is the case for "Data"), where the upper 2 bits are "01". We can then use the same definition that we use for the THUB-TCPU communication, i.e.:

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	C	C	C	C	A	A	A	A	A	A	A	A	A	A	A	A

where:
01 = "Control word"
CCCC = "Status" or "Error" code (e.g. "timeout", "no data", "PLD SEU", etc)
AAaaa = Whatever argument is needed to further specify the "Status" or "Error".

We could discuss what we do in case there is no data (but no error condition either) from a whole half tray. Do we still send the Geographical data and TDIG separator words? Or maybe just one of those? Or both with the # of preceding HPTDC data in the separator word set to 0?
Let's decide here that version 0 of the firmware sends all of the framing packets no matter if there are data or not for debugging purposes. We can then decide in the future to omit the framing information when there is no data, if so desired.

Version 0 of the firmware will ignore any further THUB-TCPU communication while processing the initial trigger command from THUB, until all data for the corresponding event has been transferred to THUB according to the above indicated sequence. In the future, we will define "Abort" control packets that will cause the TCPU to abort the current event and get ready for receipt of the next trigger. In future versions of the firmware we will also allow additional triggers to occur while the current event is being processed.

Appendix B: Programming TCPU (or TDIG) board using pre-compiled code files.

1. **CPLD:**
    a. Start Quartus; select “run with existing license”
    b. Select “Tools”; select “Programmer”, a blank programming screen opens titled “QuartusII – [chain1.cdf].
    c. Select “File”; select “Open”; select “Files of type: Programming Files...”;
    d. Browse to the location containing “SmallCPLD.CDF”; click “Open”.
    e. Select “Open a new Programmer window listing this file”; click “OK”.
    f. The chain will open.
    g. The display will show “Mode JTAG”; “Progress 0%”.
    h. The release programming filename; “Device EPM3064AL44”; Checksum as documented in the release document; “Usercode” all F’s should display.
    i. The two boxes labeled “Program/Configure” and “Verify” must be checked.
    j. Connect ByteBlaster cable to board header J7; apply power to board.
    k. Click the button labeled “Start”. The program/verify sequence will run and programming progress will be indicated in the “Progress” bar.

2. **FPGA**
    a. Start Quartus; select “run with existing license”
    b. Select “Tools”; select “Programmer”, a blank programming screen opens titled “QuartusII – [chain1.cdf].
    c. Select “File”; select “Open”; select “Files of type: Programming Files...”;
    d. Browse to the location containing “TDIG_pins1.CDF”; click “Open”.
    e. Select “Open a new Programmer window listing this file”; click “OK”.
    f. The chain will open.
    g. The display will show “Mode Active Serial Programming”; “Progress 0%”.
    h. The release programming filename; “Device EPCS4”; Checksum as documented in the release document; “Usercode” all 0’s should display.
    i. The boxes labeled “Program/Configure” and “Verify” must be checked for both the EPCS4 and the sub-device “Page 0” (4 check-marks total).
    j. Connect ByteBlaster cable to board header J8; apply power to board.
    k. Click the button labeled “Start”. The program/verify sequence will run and programming progress will be indicated in the “Progress” bar.

3. **MCU:**
    a. Start MPLAB
    b. Select “Programmer”;
    c. Select “MPLAB ICD2”. If the MPLAB module is connected to the board several messages will appear, ending with “MPLAB ICD2 Ready”.
    d. Select “File”; select “Import”.
    e. Browse to the location containing the release filename.hex
    f. Select the filename and click “open”. The message “Loaded should appear in the MPLAB “Output” window.
    g. Connect Blue Sky Electronics programming cable from MPLAB ICD2 to board header J10; apply power to board.
    h. Select “Programmer”; then select “Program”. MPLAB will erase, program, and verify the MCU code image. The message “Programming succeeded will appear in the “Output” window. (The yellow “Program target device” performs the same action as the “programmer/program” selection sequence).