TTCrx Receiver User Manual

Document version: v0.1

Author: Johan Alme (jalm@hib.no)

Firmware version: v2.0

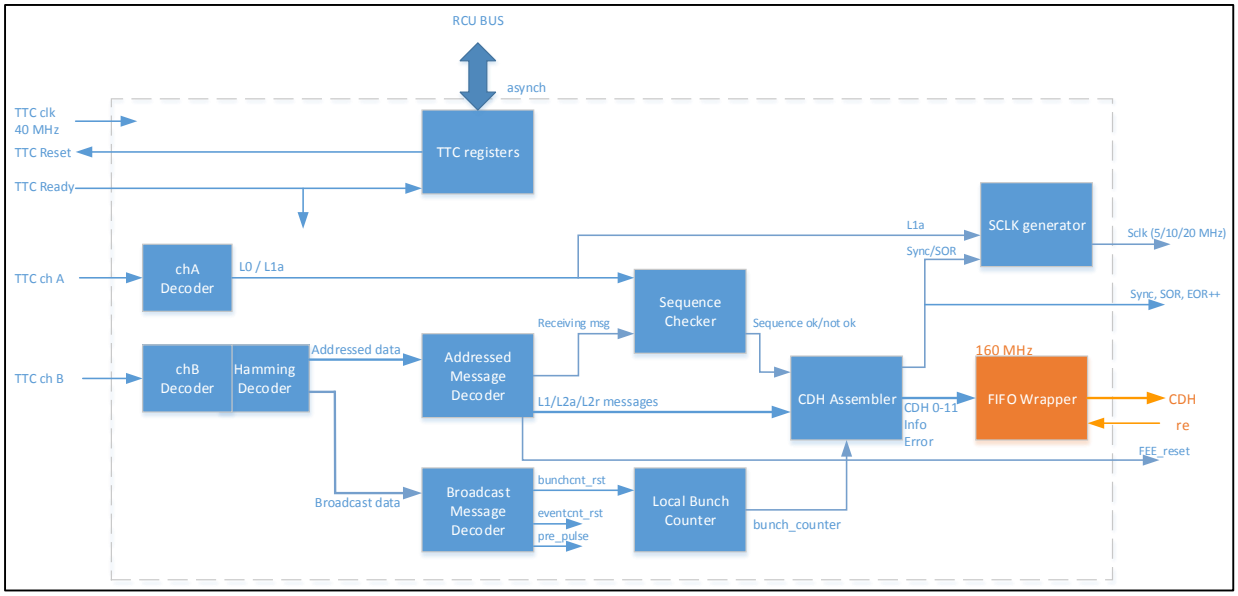

Figure 2-1: Block diagram of TTCrx Receiver Module

1 Innhold

2 Overview

2.1 IO description

2.1.1 Inputs

- ttcrx_clk: Assumed to be on a clock network. This must be handled extaernally
- chA: Sampled on negative edge of ttcrx_clk firt and then positive edge before being used.
- chB: Sampled on negative edge of ttcrx_clk firt and then positive edge before being used.
- TTCrx_ready: Used as asynch reset to the module. This is to ensure that the TTC module is in idle state if the LTU is not connected or is misbehaving.
- Areset: this is assumed coming from a different clock domain and is synched tobe used as synchronous system reset.
- Meb_full, ddl_rdy_rx: These are synched on the input – assuming to come from different clock domain.
- CDH_read_enable: in sys_clk domain. Used for reading CDH FIFO.

2.1.2 Outputs

- All outputs in the ttc_output record are in the TTC 40 MHz clock domain and must be synchronized to the clock domain they belong. False path constraints must be set accordingly
- CDH & buffered_events: These ports are in the sys _clk domain. The CDH is 33 bit since the MSB is an even parity bit of the 32 other bits.
- sclk: sampling clock output. To be forwarded to FECs as sampling clock. Derived from the ttc_clk

2.2 Generic Settings

Name	Possible Values	Explanation
GC_INCLUDE_HAMMING	0: No hamming decoding 1: Hamming encoding enabled	
GC_INCLUDE_PARITY	0: No parity encoding done 1: Parity encoding done	Parity encoding for the FIFO storage.
GC_IS_ASYNCH	0: Assuming full design running on 40 MHz clock 1: Assuming different clock for FIFO readout and RCU BUS	
GC_reg_base_addr	Any 16 bit vector (integer value)	Default Value: 16#0200#
GC_reg_addr_mask	Any 16 bit vector (integer value)	Default Value: 16#FF00#

Table 2-1: Generic variables

2.3 Clocking Strategy

40 MHz TTC clock used as system clock for all module, except for the output FIFO. The output FIFO is clocked by any clock with a higher frequency (sys_clk)

2.4 Reset Strategy

Three reset signals:

- TTCrx_ready used as asynch reset (negative polarity) to the module, resetting all but configuration settings
- *Areset:* System reset from other clock domain (positive polarity). Synchronized to TTC clock domain. Full reset
- Reset in module (see register map). Resets all status and counters, but not configuration

3 Register Map

3.1 Base address

For the current RCU implementation the base address is set to 0x200. All addresses listed below are given with respect to that.

3.2 Register map

Name	Address offset	Access	Comment
CONTROL[3:0]	0x00	RW	[0] ChA & chB input enable *(not used in current version)* [1] Mask not legal triggers [3:2] SCLK clock speed: "00": 10 MHz "01": 2.5 MHz "10": 5 MHz "11": 20 MHz [4] Generate CDH on L2r [5] Generate CDH on L2 Timeout Default value: X"33"
STATUS[23:0]	0x04	R	[0] Bunchcount Overflow [1] Run Active [2] Busy [3] TTC Ready [7:4] Buffered Events [11:8] CDH Version = x3 [23:16] Module Version = x20
ARM_SYNCH	0x08	RW	Set to '1' to initiate a resynchronizing of sampling clock upon arrival of first L0 trigger. Resets automatically on this incident.
STATUS_RESET	0x0C	T	Resets all status and counters. Does *not* reset configuration.
NA	0x10		For Future use
L1_LATENCY[15:0]	0x14	RW	[11:0] L1a latency with respect to L0 [15:12] L1a boundary region Default value: X"2117"
L2_LATENCY[31:0]	0x18	RW	[15:0] Min clock cycles since L0 [31:16] Max clock cycles since L0 Default value: X"4e200C80"
L1_MSG_LATENCY[31:0]	0x1C	RW	[15:0] Min clock cycles since L0 [31:16] Max clock cycles since L0 Default value: X"4e200028"
BCID_LOCAL[11:0]	0x20	R	Local BCID value. Sampled on valid L1a trigger
NA	0x24		For Future use
L0_COUNTER[15:0]	0x28	R	Number of L0 triggers received
L1A_COUNTER[15:0]	0x2C	R	Number of L1A triggers received
L1R_COUNTER[15:0]	0x30	R	Number of L1R situations
L1_MSG_COUNTER[15:0]	0x34	R	Number of complete L1 messages received

Name	Address offset	Access	Comment
L2A_COUNTER[15:0]	0x38	R	Number of complete L2a messages received
L2R_COUNTER[15:0]	0x3C	R	Number of L2r messages received
L2TIMEOUT_COUNTER[15:0]	0x40	R	Number of L2 timeout situations
HAMMINGERROR_COUNTER[31:0]	0x44	R	[0:15] Single Hamming Errors [31:16] Double Hamming Errors
MSGSEQ_ERROR_COUNTER[31:0]	0x48	R	[0:15] Msg Decoding Error Counter [31:16] Seq Decoding Error Counter
NA	0x4C		For Future use
CDH00[31:0]	0x50	R	Last received CDH word 0: Always X"FFFFFFFF"
CDH01[31:0]	0x54	R	Last received CDH word 1
CDH02[31:0]	0x58	R	Last received CDH word 2
CDH03[31:0]	0x5C	R	Last received CDH word 3
CDH04[31:0]	0x60	R	Last received CDH word 4
CDH05[31:0]	0x64	R	Last received CDH word 5
CDH06[31:0]	0x68	R	Last received CDH word 6
CDH07[31:0]	0x6C	R	Last received CDH word 7
CDH08[31:0]	0x70	R	Last received CDH word 8
CDH09[31:0]	0x74	R	Last received CDH word 9
EVENT_INFO[31:0]	0x78	R	Last received event information word: [0] RoI announced (ESR) [1] Sw trigger: L1_sw|L2a_sw [2] Cal. trigger: CIT_L1|CIT_L2a [3] 0 [7:4] RoC (SW trigger type) [8] L2 reject received [9] L2 accept received [10] CDH + payload is sent [11] 0 [15:12] SCLK phase upon L1a trigger [19:16] Num of words in Event (xB=d11) [31:20] X"A95" – Info identifier
EVENT_ERROR[20:0]	0x7C	R	Last received event error word: [0] ch B stop bit error [1] Single Hamming error, Addressed Message [2] Double Hamming error, Addressed Message [3] Single Hamming error, Broadcast Message [4] Double Hamming error, Broadcast Message [5] Unknown Message address [6] Incomplete L1 Message [7] Incomplete L2a Message [8] '0' [9] TTCrx Address error [10] L1_msg_content_err

Name	Address offset	Access	Comment
			[11] L2_msg_content_err [12] spurious_L0 [13] missing_L0 [14] spurious_L1a [15] boundary_L1a [16] missing_L1a [17] spurious_L1_msg [18] missing_L1_msg [19] spurious_L2_msg [20] missing_L2_msg

Table 3-1: TTCrx Register Map. Legend: R= Read Only, RW = Read and Write, T = trigger (write anything will cause an action)

3.3 BusyBox Register Map

The BusyBox has a 16 bit datawidth on the bus, hence is a wrapper written to accommodate the BusyBox bus standard. The register map for the BusyBox is given below. Note that the busybox trigger module has a base address of 0x3000, and the addresses in the table are given with respect to that.

Name	Address offset	Access	Comment
CONTROL[5:0]	0x00	RW	
STATUS[15:0]	0x02	R	
STATUS[23:16]	0x03	R	
ARM_SYNCH	0x04	RW	
STATUS_RESET	0x06	T	
L1_LATENCY[15:0]	0x0A	RW	
L2_LATENCY[15:0]	0x0C	RW	
L2_LATENCY[31:16]	0x0D	RW	
L1_MSG_LATENCY[15:0]	0x0E	RW	
L1_MSG_LATENCY[31:16]	0x0F	RW	
BCID_LOCAL[11:0]	0x10	R	
L0_COUNTER[15:0]	0x14	R	
L1A_COUNTER[15:0]	0x16	R	
L1R_COUNTER[15:0]	0x18	R	
L1_MSG_COUNTER[15:0]	0x1A	R	
L2A_COUNTER[15:0]	0x1C	R	
L2R_COUNTER[15:0]	0x1E	R	
L2TIMEOUT_COUNTER[15:0]	0x20	R	
HAMMINGERROR_COUNTER[15:0]	0x22	R	
HAMMINGERROR_COUNTER[31:16]	0x23	R	
MSGSEQ_ERROR_COUNTER[15:0]	0x24	R	
MSGSEQ_ERROR_COUNTER[31:16]	0x25	R	
CDH00[15:0]	0x28	R	
CDH00[31:16]	0x29	R	
CDH01[15:0]	0x2A	R	
CDH01[31:16]	0x2B	R	
CDH02[15:0]	0x2C	R	
CDH02[31:16]	0x2D	R	
CDH03[15:0]	0x2E	R	
CDH03[31:16]	0x2F	R	
CDH04[15:0]	0x30	R	
CDH04[31:16]	0x31	R	
CDH05[15:0]	0x32	R	
CDH05[31:16]	0x33	R	
CDH06[15:0]	0x34	R	
CDH06[31:16]	0x35	R	
CDH07[15:0]	0x36	R	
CDH07[31:16]	0x37	R	
CDH08[15:0]	0x38	R	
CDH08[31:16]	0x39	R	
CDH09[15:0]	0x3A	R	
CDH09[31:16]	0x3B	R	
EVENT_INFO[15:0]	0x3C	R	

Name	Address offset	Access	Comment
EVENT_INFO[31:16]	0x3D	R	
EVENT_ERROR[15:0]	0x3E	R	
EVENT_ERROR[20:16]	0x3F	R	

Table 3-2: TTCrx Register Map BusyBox. Legend: R= Read Only, RW = Read and Write, T = trigger (write anything will cause an action)

4 Design Details

To be added.