TTCrx Reference Manual

A Timing, Trigger and Control Receiver ASIC for LHC Detectors

J. Christiansen, A. Marchioro,
P. Moreira* and T. Toifl

CERN - EP/MIC, Geneva Switzerland

June 2002

Version 3.6

*Technical contact person e-mail: Paulo.Moreira@cern.ch

Summary of changes

Version 3.6

⇒ Introduction of a “TTCrx known bugs.”

⇒ Chapter 6, Level 1 Trigger Sequence extensively reviewed.

Version 3.5

I/O electrical information added to the manual.

Version 3.2

⇒ New package and pin-out: 144-pin flat pack ball grid array (fpBGA).

⇒ Improved immunity to Single Event Upsets (SEU).

⇒ Corrected bug in I2C logic. (When different chips were connected to the same I2C bus, which allowed multiple (>2) byte transfers, there was a risk of activating the I2C interface of the TTCrx although it was not actually addressed).

⇒ New TTCrx mezzanine board.

⇒ System integration guidelines.

⇒ Soldering guidelines.

Version 3.0

⇒ Move to the radiation-hard 0.8µm DMILL technology.

⇒ Extended supply voltage range (VDD: 3.3-5.0 V ± 10 %)

⇒ Biasing inputs Res and Res_b for the amplifier removed

⇒ A resistor-encoded ID makes the PROM optional for applications that require radiation hardness.

⇒ L1Accept trigger latency reduced by 1 cycle.

⇒ I2C interface added to access internal registers.

⇒ ERDUMP, CRDUMP and INIT Broadcast Commands replaced by individually-addressed commands

⇒ Updated Deskewing table

⇒ New pin-out

⇒ New TTCrx test board

⇒ Single Event Upset (SEU) Correction Logic

⇒ 8 bit SEU Counter

⇒ Double Hamming error counter reduced to 8 bits

Version 2.3

⇒ ClockL1Accept signal trigger latency.

Version 2.2

⇒ New TTCrx test board.

⇒ Deskew mapping table.

⇒ TTCrx new package – Ball Grid Array.

Chapter 1

Introduction

The TTCrx is a custom IC that was designed by the CERN EP Microelectronics group. This document is intended to provide a functional and physical description of the TTCrx IC from the user perspective.

The TTCrx acts as an interface between the Timing Trigger and Control distribution (TTC) system for LHC detectors and its receiving end users. The ASIC delivers the clock together with control and synchronisation information to the front-end electronics controllers in the detector. The TTCrx can be programmed to compensate for particle times of flight and for propagation delays associated with the detectors and their electronics. The IC delivers the 40.08 MHz LHC clock signal, the first level trigger decision signal, and its associated bunch and event numbers. In addition, it provides for the transmission of synchronised broadcast commands and individually-addressed commands and data.

TTC SYSTEM OVERVIEW

The Timing, Trigger and Control (TTC) system for LHC detectors has been specified and complete descriptions of the system and its functionality can be found in references [1] and [2]. However, a brief overview of the TTC system features that are most relevant for the understanding and utilisation of the TTCrx IC is given here.

The TTC system provides (1) all signals necessary to synchronise the detectors (the clock, event counter reset and bunch counter reset signals), (2) the level 1 trigger accept signal, and (3) arbitrary control data, which are all distributed on a single optical fibre.

Figure 1 illustrates the basic architecture of the TTC system: at the top of the TTC tree structure, two communication channels are Time Division Multiplexed (TDM),

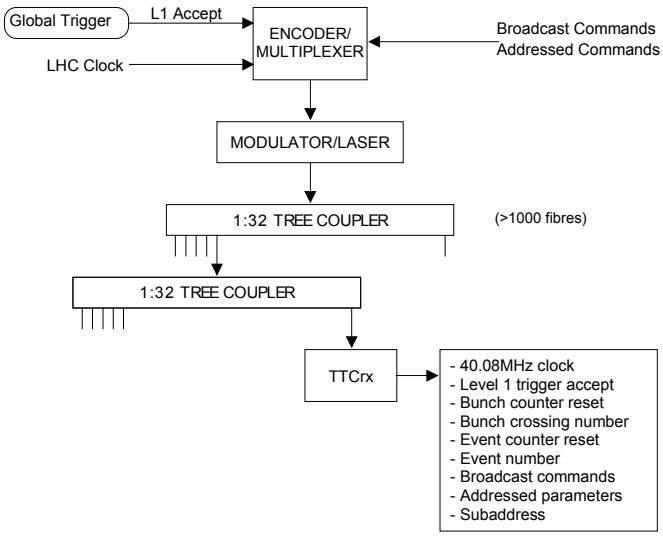


Figure 1 TTC optical distribution network

BiPhase Mark (BPM) encoded and transmitted over a passive optical fibre distribution network using a single laser source. One of the TDM channels (channel A) is exclusively dedicated to broadcast the first-level trigger-accept (L1A) decisions, delivering a one-bit decision for every bunch crossing. The other (channel B) is used to broadcast data to all or specific system destinations. The TTC system is also used to distribute the LHC 40.08 MHz bunch-crossing reference clock signal. This signal is not explicitly transmitted over the network and has to be recovered from the incoming data at each TTC system destination.

Data in channel B can be of two types [1], [2]: broadcast commands or individually-addressed commands/data. Broadcast commands are used to distribute messages to all TTC destinations in the system. When detected, these commands are executed by all the TTC receivers. The individually-addressed commands/data are implemented in the system to transmit user-defined data and commands over the network. These commands have two distinct modes of operation: in the first mode, they are aimed at the TTC receivers themselves and their user-defined content is used to control the receiver's operation. In the second mode, the data are intended for the external electronics. In this case, both the data and sub-address contents of the received commands are made externally available by the addressed TTC receiver.

Both the broadcast and the individually-addressed commands are transmitted over the TTC network using a frame format that has been specified in reference [1] and which is schematically represented in Figure 2. The frame structure contains several fields to control the transmission, and includes a field in which several redundant bits are inserted for error detection and correction. The coding scheme used is a standard Hamming code with the capability of double error detection and single bit error correction. The error correction coding covers the 8-bit data word in the case of a broadcast command/data frame and the 32-bit data in the case of an individually-addressed command/data frame[1]. The address space selection bit (E) instructs the addressed TTC receiver either to execute an internal operation or to make the received individually-addressed command/data externally available. Using this scheme it is possible to address up to 256 internal and external sub-addresses associated with up to 16K timing receivers in each timing distribution group.

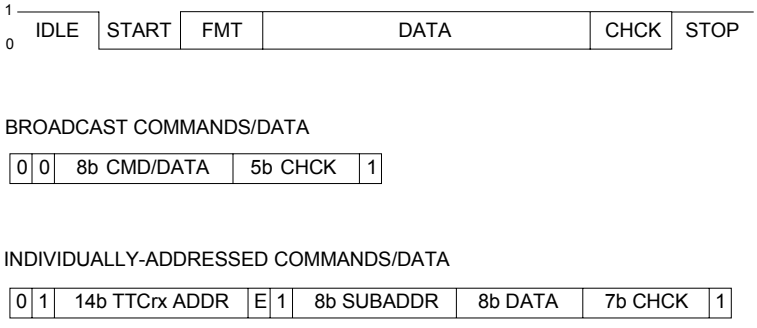


Figure 2 Data transmission frame format

Each frame is identified by a header bit (FMT) that indicates its type. Start (logical "0") and stop (logical "1") bits are always included at the beginning and end of the frame transmission to facilitate correct synchronisation.

As mentioned before, channels A and B are time division multiplexed and biphase mark encoded before transmission over the network. With this type of encoding, there is a fundamental phase ambiguity between the recovered clock and the two transmitted channels. This ambiguity is resolved automatically in the receivers by monitoring a constraint imposed on the data structure in channel A: Since the number of consecutive triggers is limited, the number of consecutive "ones" in channel A is

[1]Start, frame type and stop bits are not included in the error correction scheme.

not allowed to exceed 23 (Figure 3). There is no such limitation in channel B, so the TTCrx can identify the two channels without ambiguity.

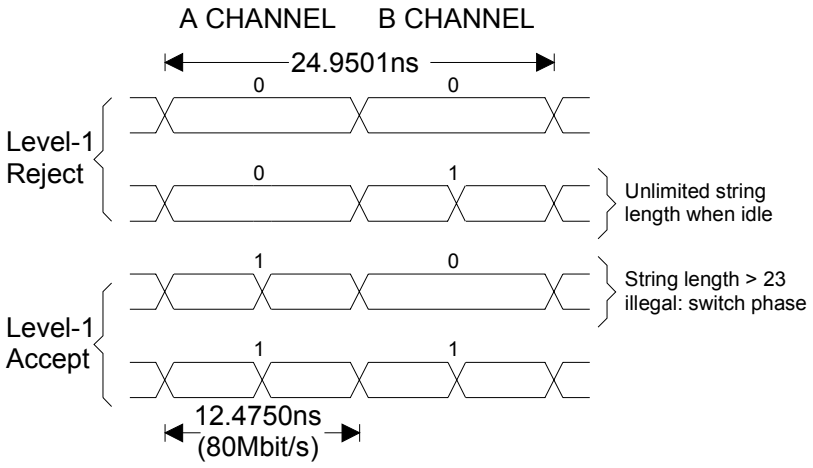


Figure 3 TDM biphase mark encoding

TTCrx OVERVIEW

A timing receiver is associated with each of the outputs of the TTC optical distribution network. Each receiver is composed of a commercial photo-detector with integrated preamplifier and of the special purpose custom IC (TTCrx) described in this manual [3].

The TTCrx recovers and distributes the 40.08 MHz LHC reference clock with minimum jitter. Secondly, the TTCrx ASIC receives, decodes, executes and distributes the commands and data broadcast over the TTC distribution network. It recognises individually-addressed commands for purposes of internal and external control and supports the transmission of synchronised broadcast commands.

The timing receiver also delivers the first-level trigger-accept decisions and their associated bunch and event identification numbers to the detector electronics. Each TTCrx IC is identified in the distribution network by a unique 14-bit channel Identification (ID) number.

TTCrx architecture

Figure 4 shows the architecture of the TTCrx. The core function of the circuit, enclosed by the dashed frame in the figure, is the recovery of the 40.08 MHz LHC clock and 80 Mbit/s serial data from a bi-phase mark encoded bit-stream, received over an optical link by a PIN-photodiode.

The differential signal coming from the photodiode[2] enters the chip on pins **In** and **In_b**. A limiting amplifier then restores the signal and converts it to full swing CMOS levels. From this, the clock and data recovery circuit extracts a 40 MHz clock signal and an 80 Mbit/s data stream.

The clock signal coming directly out of the clock and data recovery circuit (clk0 in Figure 4) is fed into two independent high-resolution phase shifters, providing a programmable delay. These devices can deskew the clock signal in steps of 104ps. The two resulting de-skewed clocks are denoted **Clock40Des1** and **Clock40Des2**.

[2] With integrated pre-amplifier

The 80 Mbit/s raw data stream is separated into two channels, denoted A and B, where channel A is exclusively reserved for the level 1 trigger accept signal, and channel B is used to transmit commands and data. Channel A is identified by the constraint that no more than 23 trigger accept decisions can occur consecutively.

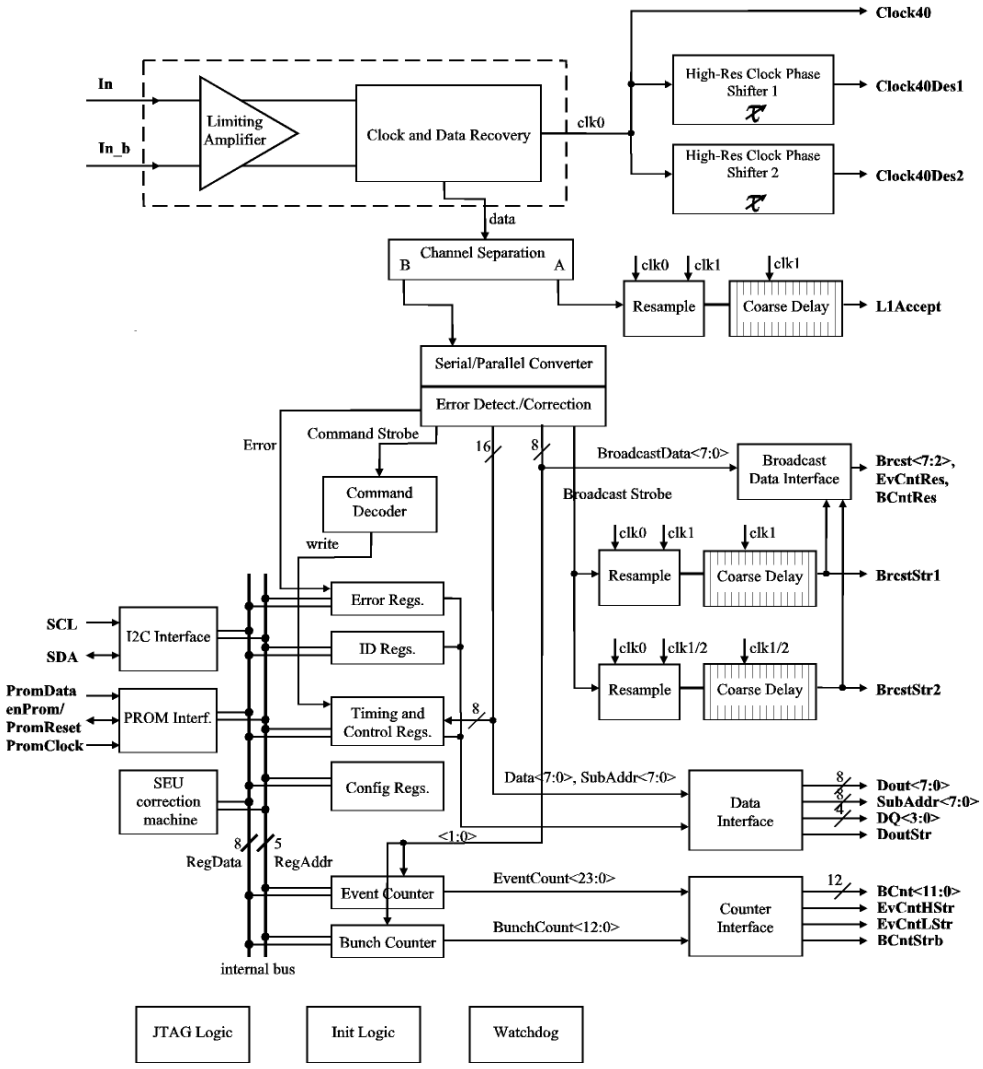


Figure 4 Timing receiver block diagram

The level 1 trigger-accept signal enters a delay pipeline, which provides coarse deskewing by delaying the signal by a programmable value in steps of 25ns.

The data in channel B are fed into a serial-to-parallel converter, which decodes the two supported data formats i.e. “Broadcast Commands” and “Individually Addressed Commands”. A Hamming error detection/correction unit checks the incoming data for transmission errors. For both formats, it can detect double bit errors, and correct single bit errors.

The broadcast command strobes **BrcstStr1** and **BrcstStr2,** which are validating newly received broadcast data, are coarse-delayed in two separate delay pipelines, each of a programmable length from 0 to 15.

An internal 8-bit bus connects 20 byte-wide internal registers. Three instances access this bus: the I2C interface, the PROM interface, and the Single Event Upset (SEU) error correction machine. The I2C interface allows to read and to write (or reset) all

internal registers. The PROM interface is only activated after a reset and if the PROM initialisation mode is chosen. It loads pre-defined register values, such as the chip ID number, from a serial PROM. The SEU check machine continuously monitors the registers for a radiation-induced single-event upset, thereby being able to correct single bit errors.

TTCrx internal registers

The TTCrx contains several internal registers used for the control and monitoring of its operation. These registers, which are described in detail in Chapter 3, are divided in the following groups:

- Timing registers
- Control register
- Error Counters
- ID and Configuration registers
- Bunch Counter and Event Counter register

The Timing and Control registers can be written through the optical link. In addition, all registers are accessible via the I2C interface (See Chapter 7.)

The Timing registers consist of two **Fine Delay** registers, and one **Coarse Delay** register. The Fine Delay registers control the delay generated in the high-resolution phase-shifters, the Coarse Delay register controls the pipeline delay for the First Level Trigger Accept (L1A) signal and the broadcast commands. The content of the Coarse Delay register in conjunction with that of the Fine Delay register affects the total amount of de-skewing.

The **Control** register is used to select different functional modes, and to minimise the power consumption of the IC by allowing to disable some of the chip functionality in applications that do not require it. For instance, the Event and Bunch counters and the Sub-Address and Data buses can be disabled if not required by the external electronics.

The **ID** and **Configuration** registers contain the configuration bits read during initialisation. They store the 14-bit chip ID, the 6-bit I2C_ID, parameters for the analogue part of the chip, and configuration bits to enable/disable specific blocks of the circuit for testing and debugging.

The **Bunch Counter** register contains a 12 bit wide free-running counter, incremented by the 40 MHz clock. The **Event Counter** register is 24 bit wide, and is incremented each time a level 1 trigger occurs. Both counters can be reset by specially defined broadcast commands. The Bunch counter register content is normally available to the outside logic on the **BCnt<11:0>** pins. However, during the two clock cycles following a trigger accept, the 24-bit Event Number register content can optionally be made available to the outside electronics on the same 12 output lines.

The **Single Bit Error** and the **Double Bit/Frame Error counters** are used to keep track of the number of errors occurring during data reception. Since the receiver Hamming decoder is capable of fully recovering from single bit errors, the data are accepted after correction and the Single Bit Error register incremented. When a double bit error is recognised by the receiver logic or a frame error is detected, the data are ignored and the contents of the Double Bit/Frame Error register incremented. An **SEU error counter** is incremented when a single event upset has been detected. The contents of the internal error counters are dumped on the external data bus when an error dump command is issued by the central TTC system.

Chapter 2

TTC System Frame Formats

This Chapter discusses the frame formats used for transmission of broadcast commands and Individually-Addressed Commands in the TTC system. This topic has already been introduced in section “TTC system overview” of Chapter 1.

FRAME FORMATS

Two basic frame formats are used to transmit commands/data to the TTC receivers: Broadcast Commands (BC) and Individually-Addressed Commands (IAC). The frames are sent with several redundant bits for single bit error correction and double bit error detection. The coding scheme used is a standard Hamming code with one additional even parity bit to detect double bit errors [4]. A start and a stop bit are included in each frame for correct frame synchronisation. The two frame formats are defined as follows:

Broadcast frame

The broadcast frame is used to distribute messages to all TTC receivers in the system (broadcast commands). This type of frame is identified by a “0” in its header bit (FMT). All TTCrx ASICs, after having performed appropriate checking on the received packet, execute the operation requested in the data part of the frame. For broadcast frames, error correction and detection is made on the eight data bits.

START	FMT	CMD/DATA <7:0>	CHCK <4:0>	STOP
0	0	dddddddd	eeeee	1

Table 1 The structure of the Broadcast Command (BC) frame

Individually-addressed commands/data frame

Individually-addressed commands/data frames are identified by a “1” in the header bit (FMT). This frame is used to address a single TTCrx in the system[3]. Data sent to a particular TTCrx are output to the **Dout<7:0>** and **SubAddr<7:0>** buses. The Data Qualifier bits **DQ<3:0>** are set to “0” for indicating IAC data, and **DoutStr** validates the data bus content. (See Chapter 9 for the signal timing.) The error correction coding covers the entire 32 data bits in the frame. Start, header and stop bits are not included in the error correction scheme.

START	FMT	TTCrxADDR <13:0>	E	1	SUBADDR <7:0>	DATA <7:0>	CHCK <6:0>	STOP
0	1	ttttttttttttt	i	1	ssssssss	dddddddd	eeeeeee	1

Table 2 The structure of the Individually-Addressed Command (IAC) frame

[3]See Chapter 4 for exception on the TTCrx ADDR “0”.

Chapter 3

Internal Registers

In this chapter, the TTCrx internal registers are described in detail. A brief summary of the TTCrx registers and their functionality can be found in Chapter 1.

TTCrx REGISTER FILE

The TTCrx contains 20 user-accessible registers, which are listed in Table 3. The I2C register address is given in the first column, followed by the PROM sequence number, which defines the order for reading predefined values during the initialisation procedure (see Chapter 8).

I2C reg. address	PROM seq. #	Register name	Default content (After reset)
Timing registers			
0	0	Fine Delay 1	00000000
1	1	Fine Delay 2	00000000
2	2	Coarse Delay	00000000
Control register			
3	3	Control	10010011
Error counter registers			
8	-	Single error count<7:0>	00000000
9	-	Single error count<15:8>	00000000
10	-	Double error count<7:0>	00000000
11	-	SEU error count <15:8>	00000000
ID registers			
16	4	ID<7:0>	00000000
17	5	MasterModeA<1:0>, ID<13:8>	00000000
18	6	MasterModeB<1:0>, I2C_ID <5:0>	00000000
Configuration registers			
19	7	Config 1	00000010
20	8	Config 2	10000100
21	9	Config 3	10100111
Status register			
22	-	Status	11100000
Bunch counter registers			
24	-	Bits <7:0>	00000000
25	-	Bits <15:8>	00000000
Event counter registers			
26	-	Bits <7:0>	00000000
27	-	Bits <15:8>	00000000
28	-	Bits <23:16>	00000000

Table 3 The TTCrx register file

Register access via I2C bus and optical link

The I2C interface can be used to read and write (or reset) all the registers in the table. The first four registers in the table, i.e. the timing and control registers can also be written by sending individually-addressed commands on the optical link.

Fine Delay register 1 and 2

Two deskewed clock outputs are provided by the high-resolution phase shifters of the TTCrx: **Clock40Des1** and **Clock40Des2**. The 25ns clock cycle is divided into 240 steps, equidistantly spaced by 104ps. The two fine delay registers encode the selected phase steps for the two independent clocks.

The fine delay registers are either loaded via the I2C bus or over the optical link by sending individually-addressed commands to internal sub-addresses “0” and “1” of a given TTCrx. Writing to the Fine delay registers thus allows the clock phase to be changed in steps of 104 ps between 0 and 25 ns.

Notice that due to the architecture of the phase shifter circuit, the byte-value of a Fine Delay register is not proportional to its corresponding phase delay, but has to be encoded first. Formulas and tables for encoding delay values are given in Appendix A.

Coarse Delay register

As shown in Figure 4, the TTCrx contains several coarse delay pipelines, providing the programmable delay of the Level 1 Trigger Accept and broadcast command signals in steps of 25ns. The Coarse Delay register holds the delay parameters for these signals. The content of this register in conjunction with that of the Fine Delay registers thus affects the total amount of deskewing.

The eight-bit **coarse delay** register holds two sets of four bits. Each determining the coarse deskewing in a range of [0:15] for two groups of registers: The Coarse delay register bits <3:0> control the amount of deskewing applied to the **L1Accept**, **BrcstStr1, BcntRes, EvCntRes, Brcst<5:2>** signals, whereas bits <7:4> determine the deskewing of the signals **BrcstStr2** and **Brcst<7:6>**.

Since the same deskewing is applied to both the L1A signal and the bunch/event counter reset strobes, the deskewing of the latter ones will also have to be performed at the source of the TTC system to compensate for the time necessary to transmit and decode these commands.

Bits	Name	Affected signals
<3:0>	Coarse delay 1	L1Accept, BrcstStr1, BcntRes, EvCntRes, Brcst<5:2>
<7:4>	Coarse delay 2	BrcstStr2, Brcst<7:6>.

Table 4 Bit assignment of the coarse delay register

Control register

The Control register is used to select the operational mode of the circuit, and to minimise the IC power consumption by allowing the disabling of some of the chip functionality in applications that do not require it. The bits of the **Control register** are allocated as follows (see Table 5):

Bits 0 and 1 specify the mode of operation for the bunch and the event counter. After the reception of a level 1 trigger accept signal, a trigger sequence is started, in which the contents of the different counters are multiplexed on the **BCnt<11:0>** bus. The settings of the control register specify the type of this trigger sequence, as described in detail in Chapter 6.

Bit 2 of the Control register (“SelClock40Des2”) determines if **Clock40Des1** or **Clock40Des2** is used for the synchronisation of the **BrcstStr2** signal and the associated upper two bits of the broadcast command byte **Brcst<7:6>**.

Bits 3 to 7 enable or disable various output signals. Switching off signals not in use significantly reduces the power consumption of the circuit.

	Function	Reset state
0	Enable Bunch Counter operation	1
1	Enable Event Counter operation	1
2	SelClock40Des2	0
3	Enable Clock40Des2 output	0
4	Enable ClockL1Accept output	1
5	Enable Parallel output bus[4]	0
6	Enable Serial B output	0
7	Enable (non-deskewed) Clock40 output	1

Table 5 Bit assignment of the Control register.

Single bit error counter

This 16-bit counter keeps track of the number of single bit errors recognised by the receiver's Hamming decoder. Since these errors are fully corrected, received commands and data are accepted by the TTCrx after correction by the receiver.

Double bit and frame error counter

This 8-bit counter counts the number of double bit Hamming errors and frame errors (i.e. received stop bit not equal to one). After such an error, the received data are ignored and no action is taken. The TTCrx tries to resynchronise to the next start bit. In the process of resynchronisation, errors can again occur.[5]

SEU error counter

The four timing and control registers, the three ID registers and the three Configuration registers are protected against the effect of a single event upset (SEU) by using a Hamming check sum. A SEU correction-machine continuously scans the contents of the registers and corrects them in the case that a single-event upset has occurred due to irradiation. Upon detection of an error, an automatic correction is performed by which the 8-bit SEU error counter is incremented.

ID registers

There are two different ID values occupying a total of three locations in the register file: The 14 bit chip ID, which is used for identifying Individually-Addressed Commands (IAC) sent over the optical link, and the 6 bit wide I2C_ID, serving as a base address for accessing the chip via the I2C interface. The bit assignment of the three ID registers is shown in Table 6.

After a reset (Chapter 8), the ID register latches the values on the SubAddr<7:0>, Data<7:0> buses in the 16 bit ID register, and the values of {SubAddr<7:6>, Data<5:0>} in the 8 bit wide I2C_ID register. By this, after a reset, the 6-bit I2C base address is then identical to the lowest six bits of the 14-bit chip ID. If the serial PROM is used for initialisation, these values are overwritten by the contents of the PROM.

[4] Setting this bit enables the following output pins: Dout<7:0>, DQ<3:0>, SubAddr<7:0> and DoutStr. All the other outputs function normally.

[5] Note that both the single error counter and double bit and frame error counter are disabled once their contents reach 65535 and 256, respectively. A re-initialisation sequence is necessary to reactivate and reset these counters.

Bit #	Name	Function	Init value Pin name	Change allowed?
ID<7:0>				
<7:0>	ID<7:0>	Chip ID (lower bits)	**Dout<7:0>**	YES
ID<15:8>				
<5:0>	ID<13:8>	Chip ID (upper bits)	**SubAddr<5:0>**	YES
<7:6>	MMA<1:0>	MasterModeA<1:0>	**SubAddr<7:6>**	NO
ID_I2C				
<5:0>	ID_I2C<5:0>	I2C base address	**Dout<5:0>**	YES
<7:6>	MMB<1:0>	MasterModeB<1:0>	**SubAddr<7:6>**	NO

Table 6 Bit assignments of the three ID registers.

The upper two bits of the ID register (bits <15:14>) and the upper two bits of the ID_I2C register (bits <7:6>) have a special meaning: They are not part of the identification number, but constitute two “master mode” bits, MasterModeA<1:0> and MasterModeB<1:0>, which determine the overall operational mode of the circuit. By hard-wiring their values via resistors, these two bits are guaranteed to be set to the correct value after a reset without the need for a radiation-hard PROM. They are replicated (MMA=MMB) for redundancy, in order to avoid that the chip, due to a single event upset (SEU), can get stuck in a mode where it could not get back to normal operation. The function of the master mode bits is explained in Chapter 8.

Configuration registers

In order to be able to test the circuit and to fine-tune certain parameters, three configuration registers are implemented on the circuit, denoted “Config1-3”. The following tables give an overview of the bit assignment of these registers. **Most bits are reserved for testing and debugging purposes, and are not supposed to be changed by the user.** Only the register bits that can be changed by the user are described in more detail.

Config 1

The bit allocation of the Config 1 register is displayed in the table below:

Config 1 (Reg.Adr. 19)				
Bit #	**Name**	**Function**	**Default value**	**Change allowed?**
<2:0>	dll_isel<2:0>	Selects DLL current	010	YES
<5:3>	pll_isel<2:0>	Selects PLL current	011	YES
6	dll_sel_aux_1	Selects test input for phase shifter 1	0	NO
7	dll_sel_aux_2	Selects test input for phase shifter 2	0	NO

Table 7 Bit assignment of Configuration register 1

The lower three bits, “dll_isel<2:0>” specify the charge-pumps current used in the delay-locked loops (DLLs) of the high-resolution phase shifters. Lower values, in general, lead to lower jitter on the clock lines. “pll_isel<2:0>” specifies the charge-pump current for the phase-locked-loop (PLL), which provides the function of clock recovery. Also in this case, lower values in general result in lower jitter. There are cases, however, e.g. when the transmitted data are highly random, in which jitter can be minimised by using higher values.

Config 2

"cf_en_check_machineA" of the Config 2 register (see Table 8) enables the internal Hamming check-machine, which constantly scans the internal registers and corrects them in case of a single event upset (SEU). For redundancy, the Config 3 register (see Table 9) contains a second bit with the same function, denoted "cf_en_check_machineB". For switching off the Hamming check machine, both bits have to be zero.

Config 2 (Reg.Adr. 20)				
Bit #	**Name**	**Function**	**Default value**	**Change allowed?**
<2:0>	mux_select<2:0>	Selects test outputs (see detailed map below)	101	NO
3	cf_sel_test_PD	Selects external test signal for enabling the PLL phase detector.	0	NO
4	cf_sel_inputA	When 1 selects inputs from optical link, otherwise test_in<3,4>	1	NO
5	cf_PLL_aux_ reset	Assert PLL test reset line	0	NO
6	cf_DLL_aux_ reset	Assert DLL test reset line	0	NO
7	cf_en_check_ machineA	Enables Hamming check machine	1	YES

Table 8. Bit assignment of Configuration register 2

Config 3

Config 3 (Reg. adr. 21)				
Bit #	**Name**	**Function**	**Default value**	**Change allowed?**
<2:0>	frequ_check_ period<2:0>	Stop frequency detection phase after 2^(n+4) cycles without "frequ_low" detected	111	NO
3	cf_dis_INITfaster	If 1 disables automatic frequ. increase after PLL reset	0	NO
4	cf_dis_watchdog	If 1 disables watchdog circuit	0	NO
5	cf_en_Hamming	Enables Hamming error detection/correction on incoming data stream	1	NO
6	cf_en_testIO	Enables Test Input/Outputs	0	NO
7	cf_en_check_ machineB	Enables Hamming check machine	1	YES

Table 9 Bit assignment of Configuration register 3

Bunch counter

The bunch counter is incremented by the received clock signal. This counter is 12 bit wide and is reset by sending a BCRST broadcast command, by writing to the register via the I2C interface, and by the chip initialisation procedure.

Event counter

The event counter is incremented upon reception of a trigger accept signal in channel A. This counter is 24 bit wide and is reset by sending an ECRST broadcast command, by writing to the register via the I2C interface, and by the chip initialisation procedure.

Status register

The status register allows to monitor some internal signals crucial for the operation of the circuit.

Status (Reg. adr. 22)			
Bit #	**Name**	**Function**	**Default value**
<3:0>	-	Always zero	0000
4	auto_reset_flag	A 1 indicates that an automatic reset has occurred due to a timeout condition in the watchdog circuit	0
5	frame_synch	A 1 indicates that channel B is synchronized to the data stream	1
6	dll_ready	A 1 indicates that the High-Resolution phase shifters are working properly	1
7	pll_ready	A 1 indicates that the clock and data recovery circuit is locked on the incoming data stream	1

During normal operation, bits <5:7> have to be one. If this is not the case, then, either the chip was reset or an error has occurred.

The status register cannot be written like an ordinary register since it does not contain any memory elements. A write access to this register has a special function, which is described in Chapter 7.

Chapter 4

Individually-Addressed Commands

As already described in the introduction, the TTCrx chip recognises two different data formats, broadcast commands and individually-addressed commands. Broadcast commands are decoded by all TTCrx's, whereas individually addressed commands are sent to specific chips with a certain identification number (ID).

This chapter discusses the "Individually-Addressed Command" (IAC) format, where a message is sent to a specific TTCrx in the system, identified by a 14-bit ID number. The net data contained in the IAC packet amounts to 16 bits. It is divided into an 8-bit DATA byte, and an 8-bit SUBADDR byte.

Individually-addressed commands can be sent to the outside world, such that their net 16-bit data content appears on the **Dout<7:0>** and **SubAddr<7:0>** pins and **DoutStr** validates the signal. Secondly, IACs can be used to write internal registers of the TTCrx and execute internal commands. One bit in the IAC data frame (the "E" bit in Table 2) signals if the command is internal or external.

TTCRX ADDRESSING

Each TTCrx IC is identified in the distribution network by a unique 14-bit channel Identification (ID) number. This number is read during the reset procedure (see Chapter 8), either from a serial PROM or - by using the hard-wired ID mechanism - from the **ID<15:0>** bus (which shares its pins with the **SubAddr<7:0>**, **Dout<7:0>** bus).

Individual addressing

The individually addressable space for each TTCrx is split into two: internal and external. The internal address space is used to write the TTCrx internal registers, while the external space allows commands and data to be transmitted to the detector electronics. When an individually-addressed command/data frame is received with the E bit equal to "0" the *internal* address space is assumed. A "1" received in the E bit indicates *external* addressing. Upon reception of an external command, the sub-address and data buses are set according to the data contents of the received command.

The TTCrx **internal** addressing space is allocated as follows:

SUBADDR <7:0>	Register / Command
00000000	Fine Delay Register 1 <7:0>
00000001	Fine Delay Register 2 <7:0>
00000010	Coarse Delay Register <7:0>
00000011	Control Register <7:0>
00000100	Execute ERDUMP command
00000101	Execute CRDUMP command
00000110	Execute RESET command

Hence, some sub-addresses (0-3) are used for writing register values, whereas others (4,5,6) are used for executing internal commands. In the first case, the 8 bit DATA section of the IAC frame is written in the specified register, while in the second case the content of the DATA section is ignored. The function of the commands is explained later in this chapter.

Global addressing with generic address 0

If an individually-addressed command is to be received by all connected TTCrx's in a system, disregarding the individual values of their IDs, the command can be sent to the generic address 0.

Note that in this case IACs are similar to broadcast commands, with the difference that the net data content of the frame is 16 bits. Although the concept of *individually* addressed commands, of course, does not apply any more in the strict sense of the word, this feature is useful for programming internal register values for the whole system with a single transmission.

INTERNAL COMMANDS

This section describes the internal commands that can be executed by the TTCrx. The commands are issued by sending an internal individually-addressed command (IAC) as described above.

ERDUMP

Error dump (Sub-address = 4): the internal error counters are dumped on the external data bus **Dout<7:0>**. Data are output during four consecutive clock cycles. As for a normal data transfer, the data strobe line **DoutStr** signals the presence of valid data on the bus and the **DQ<3:0>** bits indicate the type of the data according to the following table:

DQ<3:0>	Data<7:0> bus content
0001	Single Bit Error Counter Low
0010	Single Bit Error Counter High
0011	Double Bit Error Counter Low
0100	SEU Error Counter

CRDUMP

Configuration register dump (Sub-address = 5): The internal configuration and control registers are dumped on the external data bus. For this operation the data qualifier bits are used as follows:

DQ<3:0>	Data<7:0> bus content
0101	Fine Delay register 1
0110	Fine Delay register 2
0111	Coarse Delay register
1000	Control register
1001	ID register <7:0>
1010	ID register <13:8>

RESET

Reset the TTCrx (Sub-address = 6)

This instruction initiates a complete reset procedure (see Chapter 8) of the TTCrx. It should therefore be used with care. The command can only be received, of course, if the clock-and data recovery circuit of the TTCrx is working correctly.

Chapter 5

Broadcast commands

Structure of the broadcast data packet

The TTCrx can receive up to 256 different broadcast messages, encoded in the 8-bit broadcast data packet. These 8 bits are divided into three groups: the two lowermost bits, the four middle bits, and the two uppermost bits. The lowermost bits <1:0> are reserved for the bunch counter reset signal (bit 0), and the event counter reset signal (bit 1), the middle four bits <5:2> are referred to as the **system** broadcast message, the uppermost bits <7:6> as the **user** broadcast message. An overview of the broadcast data packet is given in the table below. (The details about timing and synchronising clock are explained later in this chapter.)

Bit #	Signal name	Internal action when high	Coarse delay value 1=bits<3:0> 2=bits<7:4>	Output synchronised with 1: Clock40Des1 2: Clock40Des2	Output pin name
0	Bunch counter reset	Resets internal bunch counter	1	1	BcntRes
1	Event counter reset	Resets internal event counter	1	1	EvCntRes
<5:2>	System message	-	1	1	Brcst<5:2>
<7:6>	User message	-	2	1 or 2	Brcst <7:6>

The broadcast data are accessible at the outside in the form of the event and bunch counter reset strobes (**BCntRes** and **EvCntRes**), corresponding to bits <1:0> of the broadcast data byte sent, and of the broadcast data bus **Brcst<7:2>**, which is validated by the strobe signals **BrcstStr1** and **BrcstStr2**.

It has to be noted that **BcntRes** and **EvCntRes,** although in principle corresponding to Brcst<1:0>, actually function as strobe signals, i.e. they stay high for only one cycle. On the other hand, data sent in bits <7:2> will remain active on **Brcst<7:2>** until the next broadcast command is sent.

Coarse delay of the broadcast signals

The different parts of the broadcast message can be delayed in steps of 25ns. The following rules apply: Bits <5:0> of the broadcast message (thus including the reset strobes for the event and bunch counter and the system messages) are delayed by the value defined by bits <3:0> of the Coarse Delay register, whereas the user message bits <7:6> are delayed by a value stored in bits <7:4> of the Coarse Delay register. It should be noted that the broadcast message bits <5:2> are thus delayed by the same value as the **L1Accept** signal.

The delays affect both the strobe signals (**BrcstStr1, Brcstr2**) and the data signals (**Brcst<7:2>, EvCntRes, BcntRes**).

Synchronisation with either Clock40Des1 or Clock40Des2

Bits <5:0> of the broadcast data are always synchronised to **Clock40Des1**, whereas the bits of the ***user*** broadcast message (bits <7:6>) can be synchronised to either **Clock40Des1** or **Clock40Des2**, depending on Bit 2 (“SelClock40Des2”) of the

Control register. Hence, the strobe signals **BrcstStr1** and the lower data bits (**BcntRes, EvCntRes, Brcst<5:2>)** are always synchronised to **Clock40Des1**, whereas **BrcstStr2** and **Brcst<7:6>** are synchronised to either **Clock40Des1** or **Clock40Des2.**

Bunch and event counter reset commands

As described above, the function of the two lowest bits of the broadcast byte is predefined in the TTC system: Bit 0 carries the bunch counter reset signal, and bit 1 the event counter reset signal, corresponding to the following command table:

Command	Format uu ssssss	Function
NOP	uu ssss00	Do nothing
BCRST	uu ssss01	Bunch counter reset
ECRST	uu ssss10	Event counter reset
EBCRST	uu ssss11	Reset event and bunch counters

Table 10 Pre-defined broadcast commands

Note that sending a Bunch counter reset or an Event counter reset signal also resets the internal bunch or event counter of the TTCrx. The interpretation of the upper six bits in the broadcast data byte (two **user** and four **system** bits) is left to the user.

Chapter 6

Level 1 Trigger Sequences

L1Accept signal

After receiving an L1Accept signal on channel “A”, the TTCrx activates the **L1Accept** pin after a delay specified by the lower four bits of the Coarse Delay Register.

ClockL1Accept signal

The **ClockL1Accept** signal combines the **Clock40** signal with the trigger information: Upon reception of a trigger accept, the output is suppressed during one clock cycle. **ClockL1Accept** has lower trigger latency than **L1Accept**. But, unlike the latter, its phase is not programmable. In order to save power, it is possible to disable the **ClockL1Accept** signal by clearing Bit 4 of the Control register.

Counter access on the BCnt<11:0> bus

The 12-bit Bunch and the 24-bit Event counter values are multiplexed on the **BCnt<11:0>** counter output bus. The lowest two bits in the Control register determine what signals are available on **BCnt<11:0>** during a trigger sequence according to the following table:

Control Register, bit <1:0>	Trigger Cycle	Signal on BCnt<11:0> pins	Strobe signal
00	default	Event counter low	-
	0	Event counter low	EvCntLStr
01	default	Bunch counter	-
	0	Bunch counter	BCntStr
10	default	Event counter low	-
	0	Event counter low	EvCntLStr
	1	Event counter high	EvCntHStr
11	default	Event counter low	-
	0	Bunch counter	BCntStr
	1	Event counter low	EvCntLStr
	2	Event counter high	EvCntHStr

Table 11 Bunch counter bus trigger cycles.

Trigger cycle 0 is the cycle during which, after the specified coarse delay, the **L1Accept** signal is activated. Cycles 1 and 2 are the following two consecutive cycles.

The ‘default’ value in the table corresponds to the case that no trigger sequence is active. Although the specified signal is available on the **BCnt<11:0>** bus, none of the strobe signals (**BCntStr**, **EvCntLStr**, **EvCntHStr**) is activated.

Note that if bits <1:0> of the Control Register are set to ‘01’ then **BCnt<11:0>** is constantly changing, leading to higher power consumption.

The first event after an Event Counter Reset (**ECRST** or **EBCRST**) will be marked as event number zero.

Minimum trigger spacing

The minimum allowed trigger spacing, that is, the maximum trigger rate, depends on the programmed trigger mode (see Table 11).

Trigger mode "00"

In trigger mode "00", upon the reception of an L1 accept signal on channel "A", the TTCrx makes available on the bunch counter bus the content of the event counter low register.

This mode has been specified to support a minimum trigger spacing of one (that is, a maximum trigger rate of 40 MHz). However, if the trigger spacing is decreased to less than three, the bunch counter bus content becomes erroneous, although the **L1Accept** and **EvCntLStr** signals continue to operate correctly (see Figure 5). Consequently, if the bunch counter bus content is used in the system, the minimum trigger spacing becomes limited to 3. If only signals **L1Accept** and **EvCntLStr** are used, then a minimum trigger spacing of one is still valid. Due to the channel identification constraint (see TTC system overview) the maximum number of consecutive triggers allowed is 22.

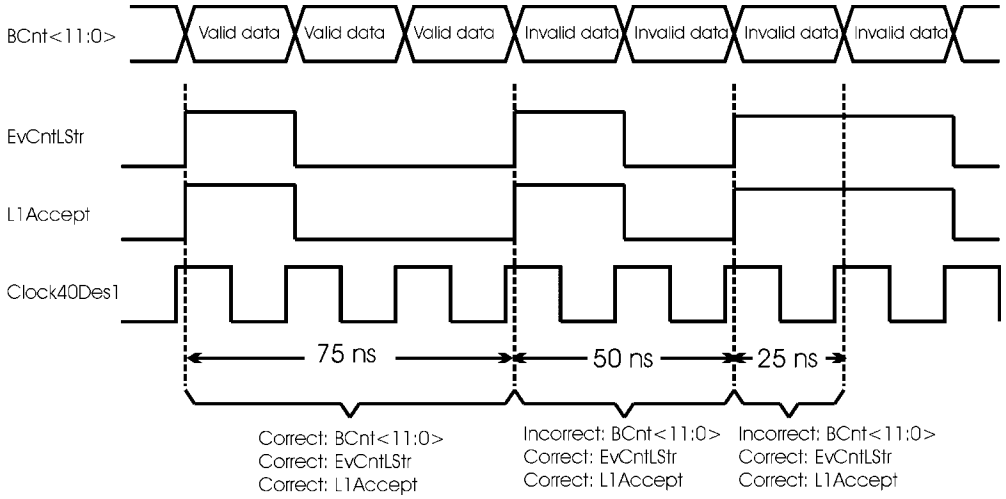


Figure 5 Trigger mode "00"

Trigger mode "01"

In this mode the minimum trigger spacing is one. That means that consecutive L1 accepts can be received in channel "A" resulting in a maximum trigger rate of 40 MHz. Due to the channel identification constraint (see TTC system overview) the maximum number of consecutive triggers allowed is 22. This mode works as specified under all conditions. See Figure 6 for an example of operation.

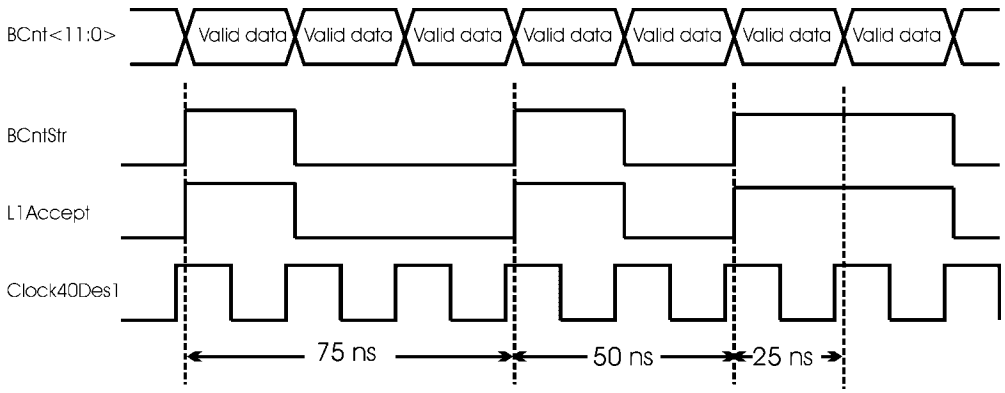


Figure 6 Trigger mode “01”

Trigger mode “10”

In trigger mode “10”, upon the reception of an L1 accept signal on channel “A”, the TTCrx makes available on the bunch counter bus the content of the event counter low followed by the content of the event counter high register.

This mode has been specified to support a minimum trigger spacing of two (that is, a maximum trigger rate of 20 MHz). However, if the trigger spacing is decreased to two, the bunch counter bus content becomes erroneous (see Figure 7). Nonetheless, the **L1Accept**, **EvCntLStr and EvCntHStr** signals continue to operate correctly (see Figure 7)**.** Consequently, if the bunch counter bus content is used in the system, the minimum trigger spacing becomes limited to 3. If only signals **L1Accept**, **EvCntLStr** and **EvCntHStr** are used, then a minimum trigger spacing of two is still valid.

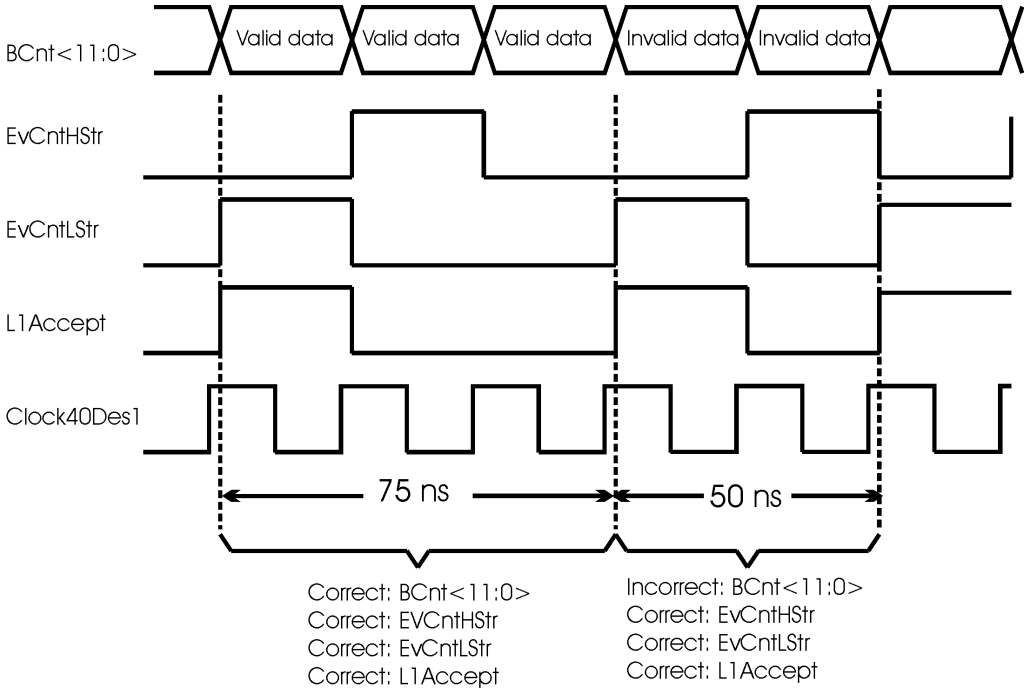


Figure 7 Trigger mode “10”

Trigger mode “11”

This is the default TTCrx trigger mode. In this mode, the minimum trigger spacing is three. That means that at least two L1 rejects must exist between two L1 accepts, resulting in a maximum trigger rate of 13.33 MHz. After the reception of an L1 accept on channel “A”, the TTCrx will output sequentially on the “Bunch Counter Bus”, the “Bunch Counter” content followed by the contents of the “Event Counter Low” and “Event Counter High” registers. This mode works as specified under all conditions. See Figure 8 for an example of operation.

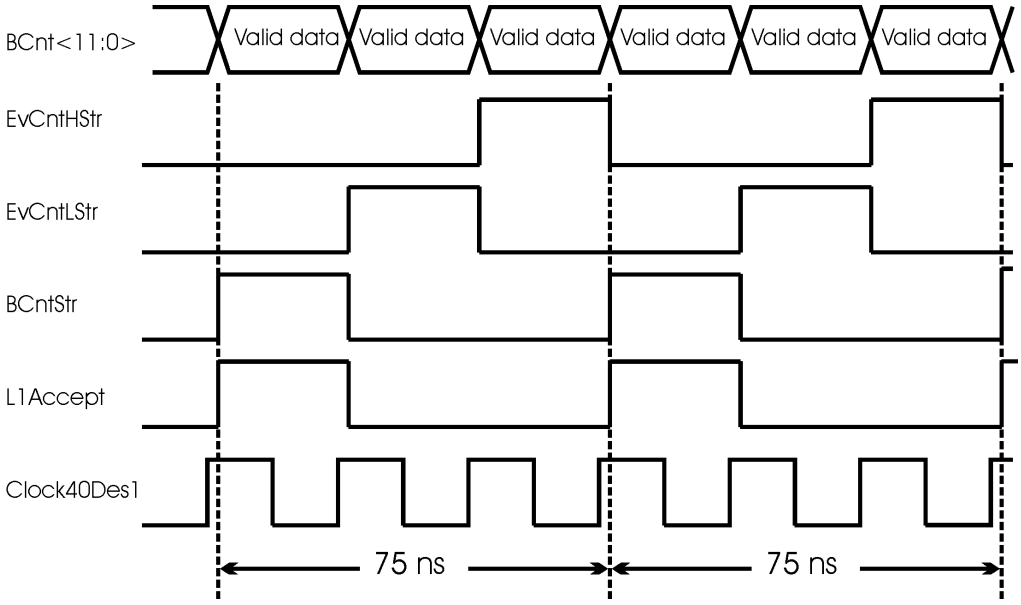


Figure 8 Trigger mode “11”

Chapter 7

Register Access via the I2C Bus

The I2C bus protocol defines a standard for an asynchronous serial bus with a maximum transfer rate of 1 Mbit/s [7].

Transferring data by using the I2C access registers

All data transfer over the I2C bus is performed using only two registers: The **I2C_pointer** register and the **I2C_data** register. The **I2C_pointer** register is five bits wide and contains the address of the internal register as defined in Table 3 (page 15). When reading the **I2C_data** register, the content of the TTCrx register *being addressed by the pointer register* is transferred. Conversely, writing a byte to the **I2C_data** register in fact writes to the TTCrx register *addressed by* the **I2C_pointer** register. Hence, each I2C access is performed in two steps:

1) Write the register number in the **I2C_pointer** register

2) Read or write the **I2C_data** register

According to the I2C bus specification, each device on the bus is addressed by a 7-bit wide I2C device address. Each TTCrx chip occupies two consecutive positions in the 7-bit I2C address space. Hence, it is possible to address 64 devices in the system. The 7-bit I2C address is derived from the content of the ID_I2C<5:0> base address register in the following way:

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

Table 12 I2C address calculation.

Reading and writing register values

All the registers shown in Table 3 (page 15) can be accessed over the I2C bus. After a write access, the corresponding register is in general set to the value of the transmitted data byte. However, a write access on some special registers i.e. the counter registers and the status register, has a different meaning:

Writing to the counter registers

Any write access to one of the counter registers (error counters, bunch counter, event counter) resets the respective counter.

Writing to the status register

Writing the value 5 to the status register initiates a reset procedure.

Writing the value 0 the status register deletes the watchdog-reset flag (See also Chapter 8).

Chapter 8

Reset procedure

A reset initialises all parts of the TTCrx, and can be initiated either

a) By a low on the **Reset_b** pin;

b) By sending an (individually-addressed) RESET command via the optical link;

c) By sending a reset command on the I2C interface;

d) By a timeout condition in the watchdog circuit.

Hardwired ID and MasterMode bits

This section describes how, after a reset, the TTCrx reads in the value of its ID, encoded with connected resistors, as seen in Figure 9.

During a reset, the output drivers on the **SubAddr<7:0>, Data<7:0>** buses are deactivated. Resistors, which connect the different pins of the bus to either VDD or GND, encode a 16 bit value, which pull the values on **SubAddr<7:0>, Data<7:0>** to a logic zero or one.

This number is then latched at the rising edge of the **Reset_b** signal. After the latching, the output drivers are switched on again, and **SubAddr<7:0>, Data<7:0>** act as outputs.

By this, the serial PROM, which was the unique option to set the chip ID in early versions, is not necessary any more. Initialising the chip with a PROM, which is still supported as an option, can however be useful to fine-tune certain chip parameters, whose predefined values were set conservatively. (e.g. the currents used in the DLLs and the PLL). Since the definition of the configuration registers has changed, it is not possible to transfer any programmed PROM from previous versions.

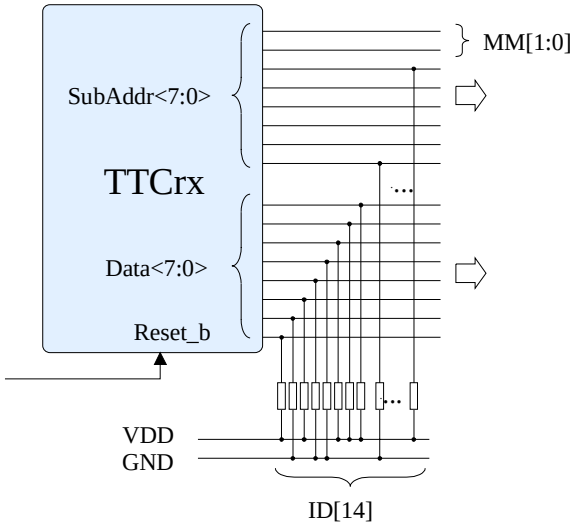


Figure 9 Using the SubAddr<7:0> and Data<7:0> bus for reading the 14-bit chip ID and the two MasterMode<1:0> bits after a reset

The two uppermost bits of **SubAddr<7:0>** are not used for the ID. They define the “MasterMode<1:0>” bits. These bits are crucial for the operation of the TTCrx, since

they encode the basic mode of operation, and were therefore included in the hard-wired initialisation procedure.

The function of the **MasterMode<1:0>** bits is described in the following table:

Bit #	Pin	Function	default value
0	SubAddr<6>	Disable Serial/Parallel Converter when 1	0
1	SubAddr<7>	Test Mode when 1	0

Table 13 Definition of the two MasterMode bits. They are read in after a reset from SubAddr<7:6>.

Master mode bit 0 determines if serial/parallel conversion shall be performed on the data in channel B. For applications where the TTCrx is to be used solely as a serial receiver, all the internal command decoding can hence be switched off by setting this bit to one.

Master mode bit 1 enables the test/debugging mode, which is never used during normal operation.

For choosing the standard mode of operation, SubAddr<6> and <7> must be resistor-connected to GND.

Minimum Width of the Reset Pulse

If the hardwired ID option is used, the external reset pulse has to have a minimum duration such that the resistors can safely pull the voltage to the desired level. Let *R* be the value of the pull-up / pull-down resistor, and *C* the overall capacitance on a pin, then the reset pulse should have a minimum width of

$$t_{\min} = 10 \cdot R \cdot C \,.$$

For example, values of *R* = 100 kΩ and *C* = 50 pF result in a minimum width of the reset pulse of 50 µs. Smaller values of *R* lead to smaller minimum reset pulses, but at the same time increase the static current consumption in the resistors after the reset.

Enabling/Disabling the PROM

In the case that a serial PROM[6] is used to set the chip ID and other register contents, the value read from the **SubAddr<7:0>, Data<7:0>** bus is overwritten by the PROM content.

In order to signal that a PROM should be used to initialise the TTCrx, the **enProm/PromReset** pin has to be connected to VDD with a pull-up resister. The pin is both input and output, using a similar technique as for reading the hardwired ID: When the TTCrx reset is active, the **enProm/PromReset** output driver is deactivated, allowing the pin to acquire the value of the resistor-connected voltage. At the falling edge of the internal reset line the value at the **enProm/PromReset** pin is latched internally, and the output driver is activated.

Note that in the case that no PROM is present the **enProm/PromReset** pin has to be connected to GND with a pull-down resistor.

[6] Serial PROM type XC1736D from Xilinx [5].

PROM Data format

As shown in Table 3 (Chapter 3), 10 eight-bit registers are initialised from the serial PROM. Hence, the first 80 bits are read by the TTCrx and have to be programmed. The data sequence goes from lower bits to higher bits and from PROM register address 0 to 9 (Table 3). Hence, the first bit of the PROM corresponds to bit 0 of register 0 (="Fine delay 1" register), the last (80^{th}) bit corresponds to bit 7 of register 9 (= "Config 3 " register).

The Xilinx XC1736D PROM has to be programmed to use an active-high reset.

Automatic Reset due to Watchdog circuit

The TTCrx incorporates a watchdog circuit which monitors whether the phase-locked loop (PLL) of the clock and data-recovery circuit is properly locked to the Biphase Mark input signal. If the chip finds out that no lock is achieved for a certain time, then an automatic reset is initiated. After a reset caused by the watchdog circuit, bit <4> of the Status register is set to one. The user thus has the possibility of knowing whether an automatic reset has occurred by reading the Status register over the I2C bus.

Chapter 9

TTCrx Signals and Timing

This chapter describes the TTCrx external signals and the most important timing relations among these signals.

TTCrx EXTERNAL SIGNALS

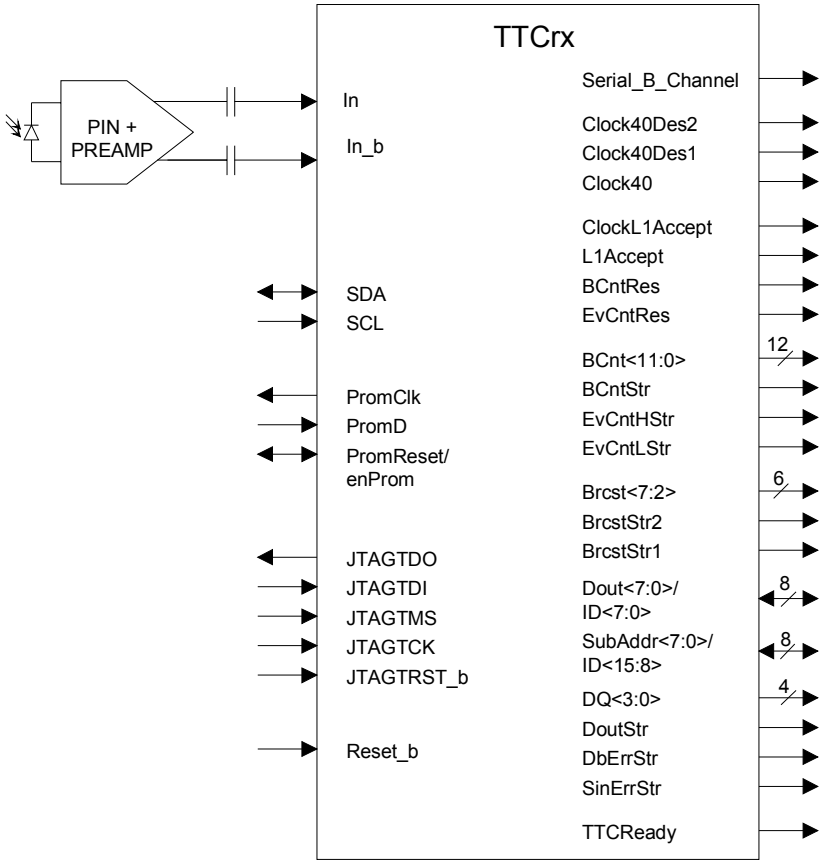


Figure 10 TTCrx external signals overview

The TTCrx signals available to the user are represented in Figure 10. Note that the signals on some pins (**enProm/PromReset**, **Dout<7:0>/ID<7:0>**, **SubAddr<7:0>/ ID<15:8>**) are defined differently during a reset (see Chapter 8). A description of the functionality of the individual signal pins is given next.

BCnt<11:0>

Counter output bus. This bus reflects the content of the bunch, event low or event high counter register, depending on the value of bits <1:0> of the control register, and the cycle of the trigger sequence (See Chapter 6). Its data type is validated by the

signals **BCntStr**, **EvCntHStr** and **EvCntLStr**. For electrical specifications, see Table 14 – I/O cell: IOF3.

BCntRes

Bunch counter reset signal. Indicates a bunch counter reset as a consequence of a broadcast command (See Chapter 5). As in the case of the **L1Accept** signal, deskewing is controlled by bits <3:0> in the coarse delay register and by the content of Fine delay 1 register. For electrical specifications, see Table 14 – I/O cell: OB33.

BCntStr

Bunch counter strobe. Indicates that a bunch number is present on the output BCnt<11:0> bus. See Chapter 6. For electrical specifications, see Table 14– I/O cell: OB33

Brcst<7:6>

Broadcast commands/data output bus. User defined part of a broadcast message. See Chapter 5. For electrical specifications, see Table 14– I/O cell: OB33.

Brcst<5:2>

Broadcast commands/data output bus. System wide part of a broadcast message. See Chapter 5. For electrical specifications, see Table 14– I/O cell: OB33.

BrcstStr1

Broadcast messages strobe 1. The total amount of deskewing applied to this strobe signal is controlled by bits <3:0> of the coarse delay register and the Fine delay register 1 (see Chapter 5). For electrical specifications, see Table 14– I/O cell: OB33.

BrcstStr2

Broadcast messages strobe 2. The total amount of deskewing applied to this strobe signal is controlled by bits <7:4> of the coarse delay register and the Fine delay register 1 or 2 (See Chapter 5). For electrical specifications, see Table 14– I/O cell: OB33.

Clock40

LHC 40.08 MHz non-deskewed reference clock signal. This output can be enabled/disabled by changing bit 7 of the Control register. (See “Control register” in Chapter 3). For electrical specifications, see Table 14 – I/O cell: OB33.

Clock40Des1

LHC 40.08 MHz deskewed reference clock 1. The deskewing factor is controlled by writing into the TTCrx subaddress “0” (see “Fine Delay register 1 and 2 ” in Chapter 3). For electrical specifications, see Table 14 – I/O cell: OB33.

Clock40Des2

LHC 40.08 MHz deskewed reference clock 2. The deskewing factor is controlled by writing into the TTCrx subaddress “1” (see “Fine Delay register 1 and 2 ” in Chapter

3). The pin is enabled by a one on bit 3 of the Control register. For electrical specifications, see Table 14 – I/O cell: OB33.

ClockL1Accept

This signal combines the non–deskewed clock and the first level trigger-accept information. The signal is coded such that, in absence of a trigger-accept decision it is identical to the Clock40 signal. When a valid first level trigger-accept decision occurs the clock signal is suppressed (logic level “0”) during a clock cycle. This signal is enabled by bit number 4 in the Control register. For electrical specifications, see Table 14 – I/O cell: OB33.

DbErrStr

Double error or frame error strobe. Indicates that a double error or a frame error has occurred. For electrical specifications, see Table 14 – I/O cell: OB33.

Dout<7:0> / ID<7:0>

Data bus. This bus is normally used to output the data content of an individually-addressed commands/data. However, it is also used for dumping the contents of the internal error counters and of the configuration register (”ERDUMP” and “CRDUMP”). The type of data present on the bus is validated by signals DQ<3:0>. Bus operation can be enable/disabled by writing into the control register. During a reset, the outputs are put in high-impedance state. This allows that connected resistors define a logic state, corresponding to the upper bits of the chip ID number (and of the master mode bits, see Chapter 8.) For electrical specifications, see Table 14 – I/O cell: IOF3.

DQ<3:0>

Data qualifier bits. This bus indicates the type of data present on the data bus register, corresponding to the table below. (See also “ERDUMP” and “CRDUMP” commands in Chapter 4). For electrical specifications, see Table 14 – I/O cell: OB33.

DQ<3:0>	Data<7:0> bus content
0000	Individually-addressed command data
0001	Single Bit Error Counter Low
0010	Single Bit Error Counter High
0011	Double Bit Error Counter Low
0100	SEU Error Counter
0101	Fine Delay register 1
0110	Fine Delay register 2
0111	Coarse Delay register
1000	Control register
1001	ID register <7:0>
1010	ID register <13:8>

DoutStr

Data out strobe. Indicates valid data on the data bus. For electrical specifications, see Table 14 – I/O cell: OB33.

EvCntHStr

Event counter high word strobe. Indicates that the counter output bus **BCnt<11:0>** contains the high word of the event number. For electrical specifications, see Table 14 – I/O cell: OB33.

EvCntLStr

Event counter low word strobe. Indicates that the counter output bus **BCnt<11:0>** contains the low word of the event number. For electrical specifications, see Table 14 – I/O cell: OB33.

EvCntRes

Event counter reset signal. Indicates an event counter reset. As the **L1Accept** signal, its deskewing is controlled by bits <3:0> in the coarse delay register and by the content of the fine delay register 1. See “Coarse Delay register” and “Fine Delay register 1 and 2 “. For electrical specifications, see Table 14 – I/O cell: OB33.

In and In_b

Differential analogue input. Correct operation of the TTCrx IC requires the peak to peak amplitude of the input differential signal to be within 20 mV_{pp} and 1 V_{pp}. The input signal has to be BiPhase Mark encoded (see Figure 3 on page 11) and the frame formats specified in sections “TTC system overview” (Chapter 1) have to be respected for correct receiver operation.

JTAGTCK

JTAG test clock. For electrical specifications, see Table 14 – I/O cell: IB15.

JTAGTDI

JTAG test data in. For electrical specifications, see Table 14 – I/O cell: IB15.

JTAGTDO

JTAG test data out. For electrical specifications, see Table 14 – I/O cell: OB93.

JTAGTMS

JTAG test mode select. For electrical specifications, see Table 14 – I/O cell: IB15.

JTAGTRST_b

JTAG test reset For electrical specifications, see Table 14 – I/O cell: IB15.

L1Accept

First level trigger-accept signal. (See Chapter 6.) The total amount of deskewing applied to this signal is controlled by bits <3:0> in the coarse delay register and by the content of the fine delay register 1. For electrical specifications, see Table 14 – I/O cell: OB33.

PromClock

Serial configuration PROM clock signal. See Chapter 8, and reference [5]. For electrical specifications, see Table 14 – I/O cell: OB33.

PromD

Serial configuration PROM data output. See Chapter 8, and reference [5]. For electrical specifications, see Table 14 – I/O cell: IB15.

EnProm / PromReset

This pin is switched to high-impedance state during a reset, allowing a resistor to encode either zero or one. In the latter case, the TTCrx reads the serial PROM for initialising its registers. After a reset, the pin is switched to output and then carries the reset pulse for the PROM generated on the TTCrx. See Chapter 8, and reference [5]. For electrical specifications, see Table 14 – I/O cell: IOF3.

Reset_b

Active low reset signal. See Chapter 8. For electrical specifications, see Table 14 – I/O cell: OB33. For electrical specifications, see Table 14 – I/O cell: IBD5.

Serial_B_Channel

This signal is used to make available to the users the serial data received on channel B (including frame, start and stop bits). The bit rate is 40.08 Mbit/s. This output can be enabled/disabled by writing into the control register. For electrical specifications, see Table 14 – I/O cell: OB33.

SinErrStr

Single error strobe. Indicates that a single error has occurred. For electrical specifications, see Table 14 – I/O cell: OB33.

SubAddr<7:0> / ID<15:8>

Sub-address bus. Used to output the sub-address content of an individually-addressed command/data. Bus operation can be enabled/disabled by changing Bit 5 of the control register. During a reset, the outputs are put in high-impedance state. This allows connected resistors to define a logic state, corresponding to the upper bits of the chip ID number (and of the master mode bits, see Chapter 8). For electrical specifications, see Table 14 – I/O cell: IOF3.

TTCReady

TTCrx ready. The TTC is only working correctly when this signal is high. For electrical specifications, see Table 14 – I/O cell: OB33.

I/O Cell	I/O type	Fanin	Fanout	Delay sensitivity
IB15	input	1.2 pF	-	-
IBD5	Inut Schmitt trigger	1.3 pF	-	-
IOF3	Bidirectional, tristate out	1.3 pF	10 pF	0.24 ns/pF
OB33	Output	-	16.7 pF	0.05 ns/pF
OB93	Output, tristate	-	10.2 pF	0.08 ns/pF

Table 14 I/O Cells electrical specifications

Signal Timing

The general timing relations among the TTCrx output signals are illustrated in the following figures. The timing relations among some of the signals can be modified by the user. The internal registers that control the TTCrx timing are: the Coarse Delay register and the Fine Delay registers 1 and 2. The contents of these registers can be modified using Individually-Addressed Commands/data as explained in sections TTCrx registers.

Signals **L1Accept, EvCntRes, BcntRes, BrcstStr1 and BrcstStr2** are used for purposes of system synchronisation. This signals do not convey any precise timing information on their own. Accurate timing information is only obtained when these signals are used in combination with the clock signal **Clock40Des1** or **Clock40Des2**. The rising edge of this clock signal marks the instant when those signals are valid.

Most signals delivered by the TTCrx are time-aligned with **Clock40Des1**. The upper two bits of the Broadcast command, together with **BrcstStr2**, can optionally be aligned with **Clock40Des2**.

Broadcast Command timing

Figure 11 shows the timing relationship of the signals used for transmitting broadcast commands. The displayed situation corresponds to the case that bit 2 of the Control register (“SelClock40Des2”) is set to 1, by which **Clock40Des2** is selected for aligning **Brcst<7:6>** and **BrcsStr2**. All the other signals are aligned to **Clock40Des1**. In the depicted case, coarse delay 1 (=coarse_delay_register<3:0>) is 0, whereas coarse delay 2 (=coarse_delay_register<7:4>) is 1, resulting in an additional delay cycle for **Brcst<7:6>** and **BrcstStr2**.

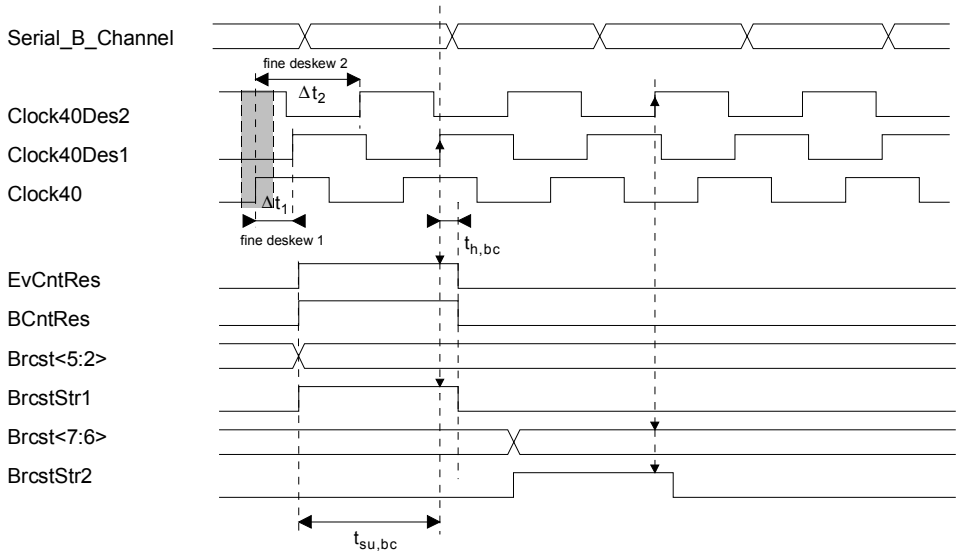


Figure 11 TTCrx Broadcast command timing

Individually Addressed Command timing

Individually addressed commands are output on the **Dout<7:0>**, **SubAddr<7:0>** and **DQ<3:0>** buses, as shown in Figure 12. The strobe signal **DoutStr**, which is aligned to **Clock40Des1**, signals the arrival of a new data value. The 16 bit net data is contained in **Dout<7:0>** and **SubAddr<7:0>,** whereas the data qualifier bits **DQ<3:0>** specify the type of data.

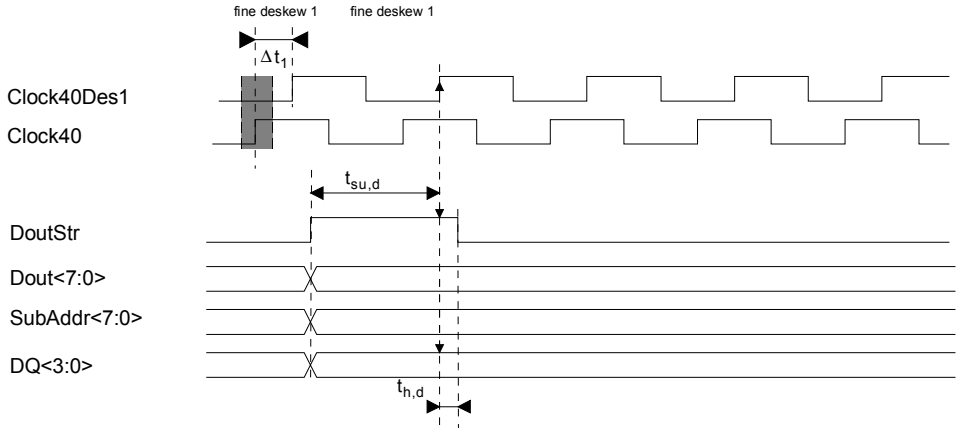


Figure 12 Individually addressed command timing

Trigger sequence

Upon reception of a trigger signal, a trigger sequence is initiated. The type of trigger sequence depends on the values of bits <1:0> of the Control register. During the sequence, the contents of the bunch counter and the event counter are made available on the **BCnt<11:0>** pins. In the situation shown in Figure 13, bits <1:0> of the Control register are set to '11', i.e. both the event counter and the bunch counter are transmitted.

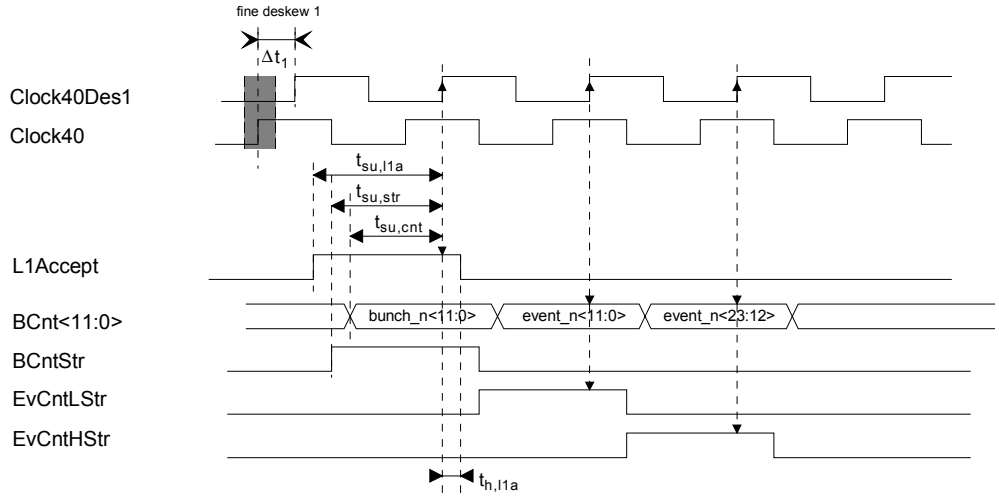


Figure 13 Trigger sequence, Bits <1:0> of the Control register = '11'

Trigger latency

Figure 14 illustrates the definition used to measure the trigger latency on the **L1Accept** pin. The value does not take into account other contributions to the trigger latency from the other system components such as the delay in the optical fibres, the optical-preamplifier delay or the TTC transmitter delay.

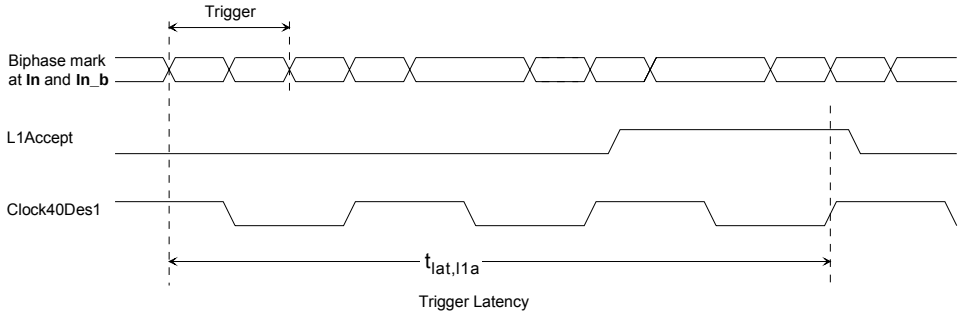


Figure 14 Trigger latency.

The timing definition for measuring trigger latency on **ClockL1Accept** is shown in Figure 15.

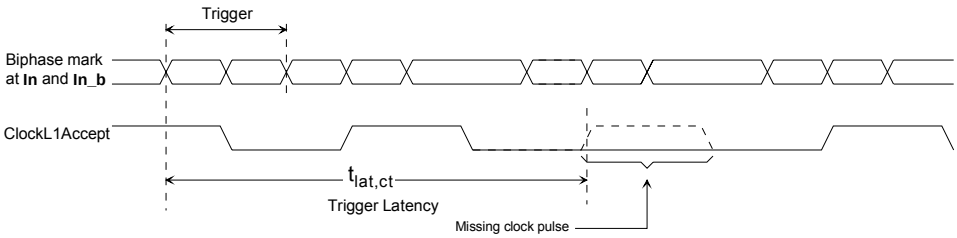


Figure 15 ClockL1Accept signal trigger latency

Recommended operating conditions[7]

		MIN	TYP	MAX	UNIT
V_{DD}	Supply voltage	3.3		5.0	V
V_{IH}	High-level input voltage	V_{DD}-0.5			V
V_{IL}	Low-level input voltage			0.5	V
T_A	Operation free-air temperature	-10	27	75	°C

Timing characteristics

In the following timing tables, Clock refers to Clock40Des1 except in the cases of BrcstStr2 and Brcst<7:2>. In this case, Clock40Des1 or Clock40Des2 must be used depending on the user selection (please see: Coarse Delay register and Control register).

Note that the signal Clock40 has no guarantied timing.

The timing of the I2C-Bus signals (SDA and SCL) is according to the I2C-Bus specification (see, for example, reference [7]).

Timing characteristics, VDD=5.0 ± 0.5 V

		MIN	TYP	MAX	UNIT
$t_{su,bc}$	Broadcast data[8] valid to Clock	20	23		ns
$t_{h,bc}$	Broadcast data hold	1.2	2		ns
$t_{su,d}$	IAC data[9] valid to Clock	17	21		ns
$t_{h,d}$	IAC data hold	2.1	3		ns
$t_{su,l1a}$	L1Accept to Clock	18	21		ns
$t_{h,l1a}$	L1Accept hold	2.3	3.5		ns
$t_{su,str}$	Counter strobes[10] to Clock	16	17		ns
$t_{su,cnt}$	BCnt<11:0> valid to Clock	9	15		ns
$t_{lat,l1a}$	L1Accept latency	64	71	83	ns
$t_{lat,ct}$	ClockL1Accept latency	35	37	43	ns
$t_{r,I2C}$	Rise time for I2C signals (SCL, SDA)			20	ns
$t_{f,I2C}$	Fall time for I2C signals (SCL, SDA)			20	ns

[7] Both the analogue (A_VDD) and the digital (D_VDD) power supplies should be set to the same value. They must be merged outside the ASIC. Care must be taken to keep supply noise to a minimum.
[8] Brcst<7:2>, BCntRes, EvCntRes, BrcstStr1, BrcstStr2.
[9] Dout<7:0>, SubAddr<7:0>, DQ<3:0>, DoutStr.
[10] BCntStr, EvCntLStr, EvCntHStr.

Timing characteristics, VDD=3.3 ± 0.3 V

		MIN	TYP	MAX	UNIT
$t_{su,bc}$	Broadcast data valid to Clock	18	21		ns
$t_{h,bc}$	Broadcast data hold	2.2	2.8		ns
$t_{su,d}$	IAC data valid to Clock	15	18		ns
$t_{h,d}$	IAC data hold	3.0	4.4		ns
$t_{su,l1a}$	L1Accept to Clock	16	19		ns
$t_{h,l1a}$	L1Accept hold	3.8	5.3		ns
$t_{su,str}$	Counter strobes to Clock	16	18		ns
$t_{su,cnt}$	BCnt<11:0> valid to Clock	4	11		ns
$t_{lat,l1a}$	L1Accept latency	66	74	86	ns
$t_{lat,ct}$	ClockL1Accept latency	37	41	47	ns
$t_{r,I2C}$	Rise time for I2C signals (SCL, SDA)			20	ns
$t_{f,I2C}$	Fall time for I2C signals (SCL, SDA)			20	ns

Chapter 10

Radiation effects

The TTCrx is now fabricated in the radiation-hard DMILL technology, which completely eliminates the possibility of a single-event latch-up, and should show a high immunity to single-event upset (SEU).

Hamming correction machine

In addition, the chip protects its most important control and configuration registers with a Hamming check sum, allowing to correct single bit errors in one of the registers caused by single event upset (SEU). The Hamming correction machine checks one memory byte per beam revolution time (88.9924µs). A total of 10 such cycles are needed to check and correct the complete set of registers.

Watchdog circuit

A watchdog circuits constantly monitors the correct operation of the circuit, and initiates a reset in case that the TTCrx has lost lock to the incoming data stream. Hence, if the chip operation gets disturbed by an SEU then it will regain operation after some time.

Possibility of blockage in the I2C interface

There is one pathological case of a single event upset in the I2C interface, which – by pulling down the I2C data line “SDA” – would block the whole bus. This condition should be noticed by the I2C bus controller. There are two possibilities to restore operation: Either the TTCrx is reset (via the Reset_b pin), or the I2C clock line “SCL” is clocked until the blockage on “SDA” disappears. The latter can be achieved by writing a dummy byte on the I2C bus.

Chapter 11

TTCrx Packaging and Pin Assignments

The TTCrx has been packaged in a 144-pin BGA 13 mm side package. The BGA package physical outline and pin assignments are specified in this chapter.

TTCrx PACKAGE

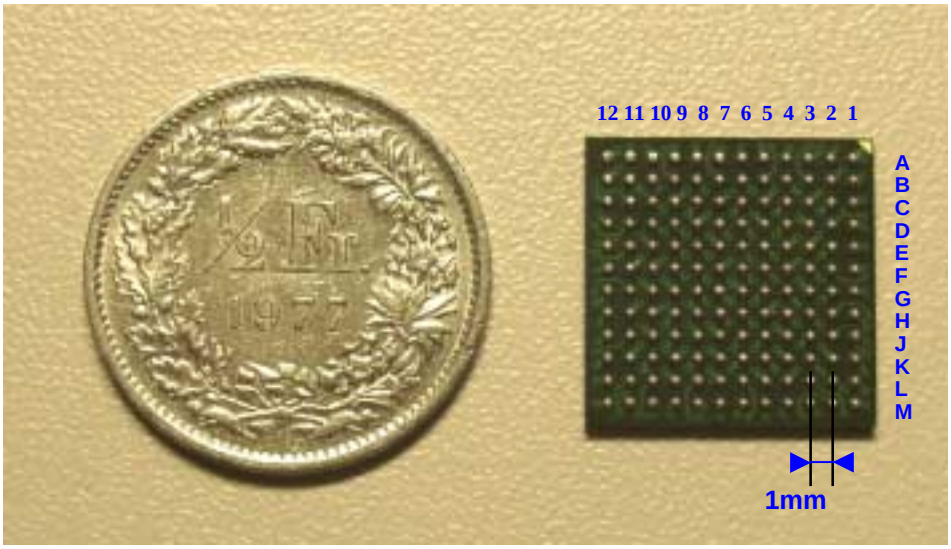


Figure 16 Photograph of 144 pin fpBGA package

PIN ASSIGNMENTS

The following two tables list the IC pin names and numbers sorted by bond pad number on the chip and by module ball position on the package.

Pin assignment sorted by IC bond pad number

PIN	Chip pad#	Package pad #	Name	Type
C02	1	4	Reset_b	in
C01	2	5	PromD	in
E05	3	6	PromClk	out
D03	4	7	PromReset	out
D02	5	8	TTCReady	out
D01	6	9	D_VDD	Digital supply (I/O)
E04	7	10	D_GND	Digital ground (I/O)
E03	8	11	D_VDD	Digital supply (core)
E01	9	12	D_GND	Digital ground (core)
E02	10	13	GND	Ground
F05	11	14	VDD	Analogue supply

F01	12	16	In	Input from PIN +
G01	13	19	In_b	Input from PIN -
G02	14	21	G_GND	Galvanic ground
G03	15	22	A_VDD	Analogue supply
G04	16	23	A_VDD	Analogue supply
H02	17	24	GND	Ground
H01	18	25	D_VDD	Digital supply (I/O)
G05	19	26	D_GND	Digital ground (I/O)
H03	20	27	SCL	I2C clock
J02	21	28	SDA	I2C data
J01	22	29	JTAGTDO	out
H04	23	30	JTAGTDI	in
J03	24	31	JTAGTMS	in
K01	25	32	JTAGTCK	in
M02	26	42	JTAGTRST_b	in
K04	27	43	Serial_B_Channel	out
H06	28	44	BCnt<11>	out
L04	29	45	BCnt<10>	out
M04	30	46	D_VDD	Digital supply (I/O)
K05	31	47	D_GND	Ground (I/O)
L05	32	48	BCnt<9>	out
M05	33	49	BCnt<8>	out
J05	34	50	BCnt<7>	out
J06	35	51	BCnt<6>	out
L06	36	52	BCnt<5>	out
M06	37	53	BCnt<4>	out
K06	38	54	BCnt<3>	out
G07	39	55	BCnt<2>	out
M07	40	56	BCnt<1>	out
L07	41	57	BCnt<0>	out
K07	42	59	D_VDD	Digital supply (core)
M08	43	60	GND	Ground (core)
L08	44	61	BCntStr	out
J07	45	62	EvCntHStr	out
K08	46	63	EvCntLStr	out
M09	47	64	BCntRes	out
L09	48	65	EvCntRes	out
J08	49	66	L1Accept	out
K09	50	67	Brcst<7>	out
K11	51	77	Brcst<6>	out
K12	52	78	BrcstStr2	out
J10	53	79	Brcst<5>	out
J11	54	80	Brcst<4>	out
J12	55	81	D_VDD	Digital supply (core)
H09	56	82	D_VDD	Digital supply (I/O)
H10	57	83	D_GND	Ground (I/O)
H12	58	84	D_VDD_C	Special I/O power
H11	59	85	Clock40	40.08 MHz clock
G08	60	86	D_GND_C	Special I/O ground
G09	61	87	D_VDD_C	Special I/O power
G12	62	88	Clock40Des1	Deskewed clock 1
G11	63	89	D_GND_C	Special I/O ground
G10	64	90	D_VDD_C	Special I/O power
F12	65	93	Clock40Des2	Deskewed clock 2
F08	66	95	ClockL1Accept	Clock & L1A
E11	67	96	D_GND_C	Special I/O ground

E12	68	97	D_VDD	Digital supply (I/O)
E10	69	98	D_GND	Ground (I/O)
F07	70	99	D_GND	Ground (core)
D11	71	100	Brcst<3>	out
D12	72	101	Brcst<2>	out
D10	73	102	BrcstStr1	out
E09	74	103	SinErrStr	out
C12	75	104	DbErrStr	out
E08	76	114	SubAddr<0>	out
E07	77	115	SubAddr<1>	out
B09	78	116	SubAddr<2>	out
A09	79	117	SubAddr<3>	out
C09	80	118	SubAddr<4>	out
C08	81	119	D_GND	Ground (I/O)
B08	82	120	D_VDD	Digital supply (I/O)
A08	83	121	SubAddr<5>	out
D08	84	122	SubAddr<6>	out
F06	85	123	SubAddr<7>	out
B07	86	124	DQ<0>	out
A07	87	125	DQ<1>	out
C07	88	126	DQ<2>	out
D07	89	127	DQ<3>	out
A06	90	128	DoutStr	out
B06	91	129	Dout<0>	out
C06	92	130	Dout<1>	out
D06	93	131	D_GND	Ground (I/O)
A05	94	132	D_VDD	Digital supply (I/O)
B05	95	133	Dout<2>	out
C05	96	134	Dout<3>	out
D05	97	135	Dout<4>	out
A04	98	136	Dout<5>	out
B04	99	137	Dout<6>	out
C04	100	138	Dout<7>	out
A01	-	1	-	N.C.
B01	-	2	-	N.C.
C03	-	3	-	N.C.
F04	-	15	-	N.C.
F02	-	17	-	N.C.
F03	-	18	-	N.C.
G06	-	20	-	N.C.
K02	-	33	-	N.C.
J04	-	34	-	N.C.
L01	-	35	-	N.C.
M01	-	36	-	N.C.
L02	-	37	-	N.C.
H05	-	38	-	N.C.
K03	-	39	-	N.C.
L03	-	40	-	N.C.
M03	-	41	-	N.C.
H07	-	58	-	N.C.
M10	-	68	-	N.C.
L10	-	69	-	N.C.
K10	-	70	-	N.C.
L11	-	71	-	N.C.
M11	-	72	-	N.C.
M12	-	73	-	N.C.

J09	-	74	-	N.C.
H08	-	75	-	N.C.
L12	-	76	-	N.C.
F09	-	91	-	N.C.
F11	-	92	-	N.C.
F10	-	94	-	N.C.
C11	-	105	-	N.C.
D09	-	106	-	N.C.
B11	-	107	-	N.C.
B12	-	108	-	N.C.
A12	-	109	-	N.C.
A11	-	110	-	N.C.
C10	-	111	-	N.C.
B10	-	112	-	N.C.
A10	-	113	-	N.C.
E06	-	139	-	N.C.
B03	-	140	-	N.C.
A03	-	141	-	N.C.
D04	-	142	-	N.C.
A02	-	143	-	N.C.
B02	-	144	-	N.C.

Pin assignments: sorted by pin

PIN	Chip pad#	Package pad #	Name	Type
A01	-	1	-	N.C.
A02	-	143	-	N.C.
A03	-	141	-	N.C.
A04	98	136	Dout<5>	out
A05	94	132	D_VDD	Digital supply (I/O)
A06	90	128	DoutStr	out
A07	87	125	DQ<1>	out
A08	83	121	SubAddr<5>	out
A09	79	117	SubAddr<3>	out
A10	-	113	-	N.C.
A11	-	110	-	N.C.
A12	-	109	-	N.C.
B01	-	2	-	N.C.
B02	-	144	-	N.C.
B03	-	140	-	N.C.
B04	99	137	Dout<6>	out
B05	95	133	Dout<2>	out
B06	91	129	Dout<0>	out
B07	86	124	DQ<0>	out
B08	82	120	D_VDD	Digital supply (I/O)
B09	78	116	SubAddr<2>	out
B10	-	112	-	N.C.
B11	-	107	-	N.C.
B12	-	108	-	N.C.
C01	2	5	PromD	in
C02	1	4	Reset_b	in
C03	-	3	-	N.C.
C04	100	138	Dout<7>	out
C05	96	134	Dout<3>	out

C06	92	130	Dout<1>	out
C07	88	126	DQ<2>	out
C08	81	119	D_GND	Ground (I/O)
C09	80	118	SubAddr<4>	out
C10	-	111	-	N.C.
C11	-	105	-	N.C.
C12	75	104	DbErrStr	out
D01	6	9	D_VDD	Digital supply (I/O)
D02	5	8	TTCReady	out
D03	4	7	PromReset	out
D04	-	142	-	N.C.
D05	97	135	Dout<4>	out
D06	93	131	D_GND	Ground (I/O)
D07	89	127	DQ<3>	out
D08	84	122	SubAddr<6>	out
D09	-	106	-	N.C.
D10	73	102	BrcstStr1	out
D11	71	100	Brcst<3>	out
D12	72	101	Brcst<2>	out
E01	9	12	D_GND	Digital ground (core)
E02	10	13	GND	Ground
E03	8	11	D_VDD	Digital supply (core)
E04	7	10	D_GND	Digital ground (I/O)
E05	3	6	PromClk	out
E06	-	139	-	N.C.
E07	77	115	SubAddr<1>	out
E08	76	114	SubAddr<0>	out
E09	74	103	SinErrStr	out
E10	69	98	D_GND	Ground (I/O)
E11	67	96	D_GND_C	Special I/O ground
E12	68	97	D_VDD	Digital supply (I/O)
F01	12	16	In	Input from PIN +
F02	-	17	-	N.C.
F03	-	18	-	N.C.
F04	-	15	-	N.C.
F05	11	14	VDD	Analogue supply
F06	85	123	SubAddr<7>	out
F07	70	99	D_GND	Ground (core)
F08	66	95	ClockL1Accept	Clock & L1A
F09	-	91	-	N.C.
F10	-	94	-	N.C.
F11	-	92	-	N.C.
F12	65	93	Clock40Des2	Deskewed clock 2
G01	13	19	In_b	Input from PIN -
G02	14	21	G_GND	Galvanic ground
G03	15	22	A_VDD	Analogue supply
G04	16	23	A_VDD	Analogue supply
G05	19	26	D_GND	Digital ground (I/O)
G06	-	20	-	N.C.
G07	39	55	BCnt<2>	out
G08	60	86	D_GND_C	Special I/O ground
G09	61	87	D_VDD_C	Special I/O power
G10	64	90	D_VDD_C	Special I/O power
G11	63	89	D_GND_C	Special I/O ground
G12	62	88	Clock40Des1	Deskewed clock 1
H01	18	25	D_VDD	Digital supply (I/O)

H02	17	24	GND	Ground
H03	20	27	SCL	I2C clock
H04	23	30	JTAGTDI	in
H05	-	38	-	N.C.
H06	28	44	BCnt<11>	out
H07	-	58	-	N.C.
H08	-	75	-	N.C.
H09	56	82	D_VDD	Digital supply (I/O)
H10	57	83	D_GND	Ground (I/O)
H11	59	85	Clock40	40.08 MHz clock
H12	58	84	D_VDD_C	Special I/O power
J01	22	29	JTAGTDO	out
J02	21	28	SDA	I2C data
J03	24	31	JTAGTMS	in
J04	-	34	-	N.C.
J05	34	50	BCnt<7>	out
J06	35	51	BCnt<6>	out
J07	45	62	EvCntHStr	out
J08	49	66	L1Accept	out
J09	-	74	-	N.C.
J10	53	79	Brcst<5>	out
J11	54	80	Brcst<4>	out
J12	55	81	D_VDD	Digital supply (core)
K01	25	32	JTAGTCK	in
K02	-	33	-	N.C.
K03	-	39	-	N.C.
K04	27	43	Serial_B_Channel	out
K05	31	47	D_GND	Ground (I/O)
K06	38	54	BCnt<3>	out
K07	42	59	D_VDD	Digital supply (core)
K08	46	63	EvCntLStr	out
K09	50	67	Brcst<7>	out
K10	-	70	-	N.C.
K11	51	77	Brcst<6>	out
K12	52	78	BrcstStr2	out
L01	-	35	-	N.C.
L02	-	37	-	N.C.
L03	-	40	-	N.C.
L04	29	45	BCnt<10>	out
L05	32	48	BCnt<9>	out
L06	36	52	BCnt<5>	out
L07	41	57	BCnt<0>	out
L08	44	61	BCntStr	out
L09	48	65	EvCntRes	out
L10	-	69	-	N.C.
L11	-	71	-	N.C.
L12	-	76	-	N.C.
M01	-	36	-	N.C.
M02	26	42	JTAGTRST_b	in
M03	-	41	-	N.C.
M04	30	46	D_VDD	Digital supply (I/O)
M05	33	49	BCnt<8>	out
M06	37	53	BCnt<4>	out
M07	40	56	BCnt<1>	out
M08	43	60	GND	Ground (core)
M09	47	64	BCntRes	out

M10	-	68	-	N.C.
M11	-	72	-	N.C.
M12	-	73	-	N.C.

Chapter 12

JTAG Boundary-Scan

The TTCrx implements a subset of the JTAG/IEEE 1149.1 standard (see for instance [6]) providing the capability for board-level connectivity tests.

JTAG Device ID

The JTAG logic includes a Device Identification Register and the device identification number is:

ID = “0545408F” (HEX)

Boundary Scan Register

The Boundary Scan Register (BSR) includes all the I/O signals with exception of the analogue signals. Interface signals between the full custom part and the standard cells part of the design are also included in the BSR.

Boundary scan register read out order

Order	PIN #	Name	Type	Description
1	D 02	TTCReady	out	TTCrx is ready and stable
2	J 02	SDA	out	I2C data output
3	E 05	PromClk	out	Serial Prom clock
4	D 03	PromReset	out	Serial Prom CE* and reset
5	K 04	Serial_B_Channel	out	Serial B channel
6	E 09	SinErrStr	out	Single bit error strobe
7	C 12	DbErrStr	out	Double bit error strobe
8	H 11	Clock40	out	40.08 MHz clock
9	G 12	Clock40Des1	out	Deskewed 40.08 MHz clock 1
10	F 12	Clock40Des2	out	Deskewed 40.08 MHz clock 2
11	F 08	ClockL1Accept	out	Clock/Trigger output
12	D 10	BrcstStr1	out	Strobe for system broadcast bus
13	K 12	BrcstStr2	out	Strobe for user defined broadcast
14	M 09	BCntRes	out	Bunch counter reset
15	L 09	EvCntRes	out	Event counter reset strobe
16	D 12	Brcst<2>	out	User defined broadcast bus
17	D 11	Brcst<3>	out	User defined broadcast bus
18	J 11	Brcst<4>	out	System broadcast bus
19	J 10	Brcst<5>	out	System broadcast bus
20	K 11	Brcst<6>	out	System broadcast bus
21	K 09	Brcst<7>	out	System broadcast bus
22	J 08	L1Accept	out	L1 accept strobe
23	J 07	EvCntHStr	out	Event counter high strobe
24	K 08	EvCntLStr	out	Event counter low strobe
25	L 08	BCntStr	out	Bunch counter strobe
26	L 07	BCnt<0>	out	Bunch counter / Ev Counter bus
27	M 07	BCnt<1>	out	Bunch counter / Ev Counter bus
28	G 07	BCnt<2>	out	Bunch counter / Ev Counter bus
29	K 06	BCnt<3>	out	Bunch counter / Ev Counter bus

30	M 06	BCnt<4>	out	Bunch counter / Ev Counter bus
31	L 06	BCnt<5>	out	Bunch counter / Ev Counter bus
32	J 06	BCnt<6>	out	Bunch counter / Ev Counter bus
33	J 05	BCnt<7>	out	Bunch counter / Ev Counter bus
34	M 05	BCnt<8>	out	Bunch counter / Ev Counter bus
35	L 05	BCnt<9>	out	Bunch counter / Ev Counter bus
36	L 04	BCnt<10>	out	Bunch counter / Ev Counter bus
37	H 06	BCnt<11>	out	Bunch counter / Ev Counter bus
38	A 06	DoutStr	out	Data strobe
39	E 08	SubAddr<0>	out	External subaddress bus
40	E 07	SubAddr<1>	out	External subaddress bus
41	B 09	SubAddr<2>	out	External subaddress bus
42	A 09	SubAddr<3>	out	External subaddress bus
43	C 09	SubAddr<4>	out	External subaddress bus
44	A 08	SubAddr<5>	out	External subaddress bus
45	D 08	SubAddr<6>	out	External subaddress bus
46	F 06	SubAddr<7>	out	External subaddress bus
47	B 06	Dout<0>	out	Data output bus
48	C 06	Dout<1>	out	Data output bus
49	B 05	Dout<2>	out	Data output bus
50	C 05	Dout<3>	out	Data output bus
51	D 05	Dout<4>	out	Data output bus
52	A 04	Dout<5>	out	Data output bus
53	B 04	Dout<6>	out	Data output bus
54	C 04	Dout<7>	out	Data output bus
55	B 07	DQ<0>	out	Data qualifier
56	A 07	DQ<1>	out	Data qualifier
57	C 07	DQ<2>	out	Data qualifier
58	D 07	DQ<3>	out	Data qualifier
59	C 02	Reset_b	in	General reset input
60	C 01	PromD	in	Serial Prom data
61	H 03	SCL	in	I2C clock
62	J02	SDA	In/out	I2C data

Appendix A

Programming Fine deskewing values

Due to the Vernier principle used in the fine de-skew DLL there is no direct correspondence between the value programmed in the fine de-skew register and the resulting delay. In this appendix, an unfolding mapping table for fine de-skew programming is given.

FINE-DESKEWING PRINCIPLE

In order to obtain a sub-gate delay resolution an architecture based on two staggered delay locked loops is used in the TTCrx. Its principle of operation can be easily understood with reference to Figure 17. In this scheme, a first DLL generates N replicas of the recovered clock each one of them delayed by Δt_N = T/N seconds from the previous one, where T is the recovered clock period. One of these signals is selected as the input to the following delay locked loop. A second DLL generates N-1 copies of the clock signal but this time $\Delta t_{(N-1)}$ = T/(N-1) seconds apart. By appropriate output tap selection in each DLL the clock signal can be shifted with a time resolution given by $\Delta t = \Delta t_{(N-1)} - \Delta t_N$

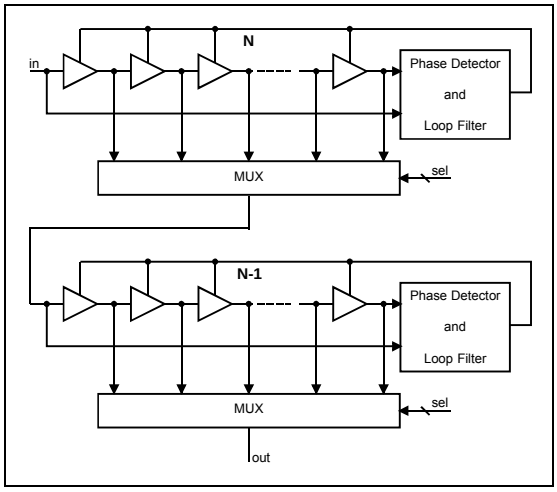


Figure 17 Fine de-skew delay generator architecture

In the TTCrx N=16 was used resulting in a minimum time step of 104.17 ps. The overall delay of the output clock signal with respect to input clock is given by

$$\tau = m \cdot \frac{T_{cycle}}{16} + n \cdot \frac{T_{cycle}}{15},$$

where Tcycle=24.95ns, corresponding to the LHC clock frequency of 40.08MHz. The variables *m* and *n* denote the delay tap chosen in the multiplexers, with *m* going from 0 to 15 and *n* from 0 to 14. The value of *m* is given to the upper four bits of the Fine Delay 1/2 register, and *n* corresponds to the lower four bits.

Since the clock signal is periodic, the delay t is mapped on the interval [0..24.95ns] and thus given by

$$\tau = T_{cycle} \cdot [(m \cdot 15 + n \cdot 16) \bmod 240],$$

where *mod* denotes the modulo operator.

If the desired fine delay is $K \times \Delta t$ (where K is an integer between 0 and 239) then the corresponding values for m and n have to be calculated and written to the Fine Delay register. Formulas for converting from K to (m,n) and vice versa are given below together with a conversion table. Note that the smallest delay (K=0) does *not* occur for (m=0,n=0), but for (m=14,n=0), which is due to timing requirements in the data-re-sampling units[11].

Conversion formulas

$$K = [m \cdot 15 + n \cdot 16 + 30] \bmod 240,$$

$$n = K \bmod 15,$$

$$m = [(K \operatorname{div} 15) - n + 14] \bmod 16.$$

with *div* denoting the integer division operator (with truncation.)

The values of n and m are then combined to an 8-bit value nm

$$nm = 16 \cdot n + m,$$

which can be written to one of the Fine Delay registers.

Conversion Table

K	nm
0	14
1	29
2	44
3	59
4	74
5	89
6	104
7	119
8	134
9	149
10	164
11	179
12	194
13	209
14	224
15	15
16	30
17	45
18	60
19	75
20	90
21	105
22	120
23	135
24	150
25	165
26	180
27	195
28	210
29	225
30	0
31	31
32	46
33	61
34	76
35	91
36	106
37	121
38	136
39	151
40	166
41	181
42	196
43	211
44	226
45	1
46	16
47	47
48	62
49	77
50	92
51	107
52	122
53	137

[11] This has also been the case in the previous versions of the TTCrx.

54	152
55	167
56	182
57	197
58	212
59	227
60	2
61	17
62	32
63	63
64	78
65	93
66	108
67	123
68	138
69	153
70	168
71	183
72	198
73	213
74	228
75	3
76	18
77	33
78	48
79	79
80	94
81	109
82	124
83	139
84	154
85	169
86	184
87	199
88	214
89	229
90	4
91	19
92	34
93	49
94	64
95	95
96	110
97	125
98	140
99	155
100	170
101	185
102	200
103	215
104	230
105	5
106	20
107	35
108	50
109	65
110	80
111	111
112	126
113	141
114	156
115	171
116	186
117	201
118	216
119	231
120	6
121	21
122	36
123	51
124	66
125	81
126	96
127	127
128	142
129	157
130	172
131	187
132	202
133	217
134	232
135	7
136	22
137	37
138	52
139	67
140	82
141	97
142	112
143	143
144	158
145	173
146	188
147	203
148	218
149	233
150	8
151	23
152	38
153	53
154	68
155	83
156	98
157	113
158	128
159	159
160	174
161	189
162	204
163	219
164	234
165	9
166	24
167	39
168	54
169	69

170	84
171	99
172	114
173	129
174	144
175	175
176	190
177	205
178	220
179	235
180	10
181	25
182	40
183	55
184	70
185	85
186	100
187	115
188	130
189	145
190	160
191	191
192	206
193	221
194	236
195	11
196	26
197	41
198	56
199	71
200	86
201	101
202	116
203	131
204	146
205	161

206	176
207	207
208	222
209	237
210	12
211	27
212	42
213	57
214	72
215	87
216	102
217	117
218	132
219	147
220	162
221	177
222	192
223	223
224	238
225	13
226	28
227	43
228	58
229	73
230	88
231	103
232	118
233	133
234	148
235	163
236	178
237	193
238	208
239	239

Appendix B

Integrating the TTCrx in the System

This appendix presents guidelines for implementing the TTCrx chip on a system's PCB, and describes the TTCrm mezzanine test card, which was developed and manufactured to facilitate testing of the current TTCrx version. It is available to designers wishing to integrate the TTCrx IC in their systems.

GENERAL GUIDELINES

In order to achieve low clock jitter and a high dynamic range of the optical input power, some basic rules have to be followed when the TTCrx is mounted on the PCB. These rules include

- Provide power and ground plane layers on the PCB.
- Use decoupling capacitances between all power and ground pins. Place the decoupling capacitances as close as possible to the supply pins.
- If possible, have a separate supply for the analogue part of the circuit (all pins denoted A_VDD, see Chapter 11.)
- Minimize the length of the signal path from the pin-photodiode to the TTCrx.
- Minimize any possible coupling to the signal from the photodiode.
- Use a symmetric structure with equal length for both inputs **In** and **In_b**. (1 mm of wiring corresponds to ~6 ps in timing.)
- Use the equalizer (high-pass filter) structure shown in Figure 18 (for a differential photodiode, e.g. TRR-1B43) and Figure 19 (for a single-ended photodiode, e.g. HFBR 2316) in order to minimize jitter for random data transfer.
- If more TTCrx's in the same system receive the signal from a single photo-diode, their outputs should be buffered by an additional post-amplifier. Do not distribute the weak signal from the diode across a board.
- If the JTAG signals are not used, make sure that the **JTAGTRST_b** pin is connected to ground with a pull-down resistor.
- The **enProm** pin has to be connected to either ground or Vdd with a 100k pull-up/down resistor. If the Prom option is not used, the SubAddr<7:6> bits (=MasterMode bits) have to be connected to GND with a pull-down resistor. (See Chapter 8, "Reset Procedure").
- The I2C pull-up resistors on the I2C clock and data lines (pins **SCL** and **SDA**) should be chosen small enough in order to provide the timing specified in Chapter 9, "Signal Timing".

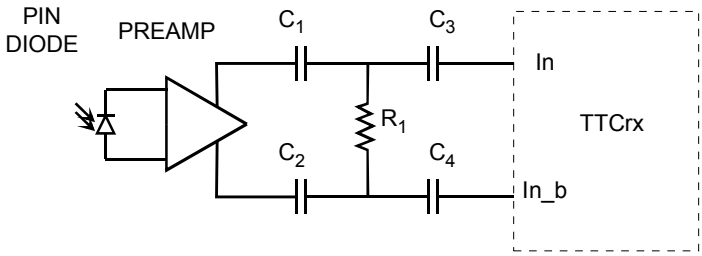


Figure 18 Equalizer structure to be used with a differential pin-photo diode+ preamplifier. (C1, C2=10 nF, C3, C4=100 nF, R1 = 200 Ohms).

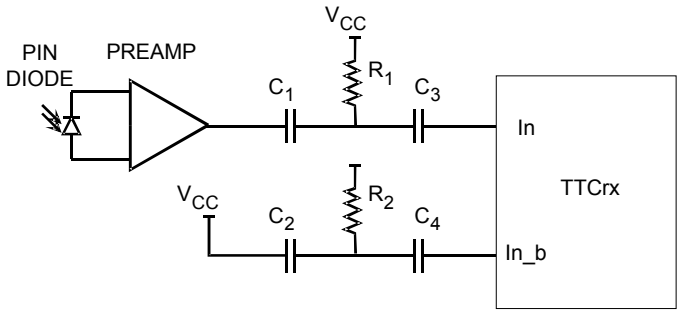


Figure 19 Equalizer structure to be used with a single-ended pin-photodiode + pre-amplifier, with output referenced to V_{CC}. (C1, C2=10 nF, C3, C4=100 nF, R1, R2 = 100 Ohms).

SOLDERING GUIDELINES

Packaged devices are very sensitive to moisture. Soldering the devices after the package has accumulated a significant amount of water can lead to the destruction of the chip. The devices should therefore be soldered within 168 hours after taking them out of the sealed bag. If the devices have been in contact with moister air for a longer period (> 196 h), they have to be pre-baked before undergoing an infrared soldering step.

MEZZANINE TEST BOARD

The schematic of the TTCrx mezzanine card (TTCrm) is shown in Figure 21. The present test board for the 144 pin BGA package (ECP 680-1102-630) contains the TTCrx IC, an integrated detector/preamplifier and a serial configuration PROM (XC1736D) The mezzanine card supports the use of pull up/down resistors to set the TTCrx individual address. The address is programmed by setting jumpers ST1 to ST6 and ST9 to ST16. The test board is prepared for soldering either the Agilent HFBR-2316T or the Honeywell HFD 8005 photodiode.

The jumper denoted "ST17" connects the enProm pin to either VDD or ground, thereby choosing if the Prom should be used to initialise the circuit.

For applications that do not require the use of JTAG, the JTAG reset pin (**JTAGTRST_B)** is connect to ground by a pull-down resistor in order ensure the correct operation of the circuit, The ASIC operation mode is set by pins **SubAddr<7:6>** through the use of pull-down/up resistors. These resistors are provided on the test board and the setting is done with jumpers ST7 and ST8.

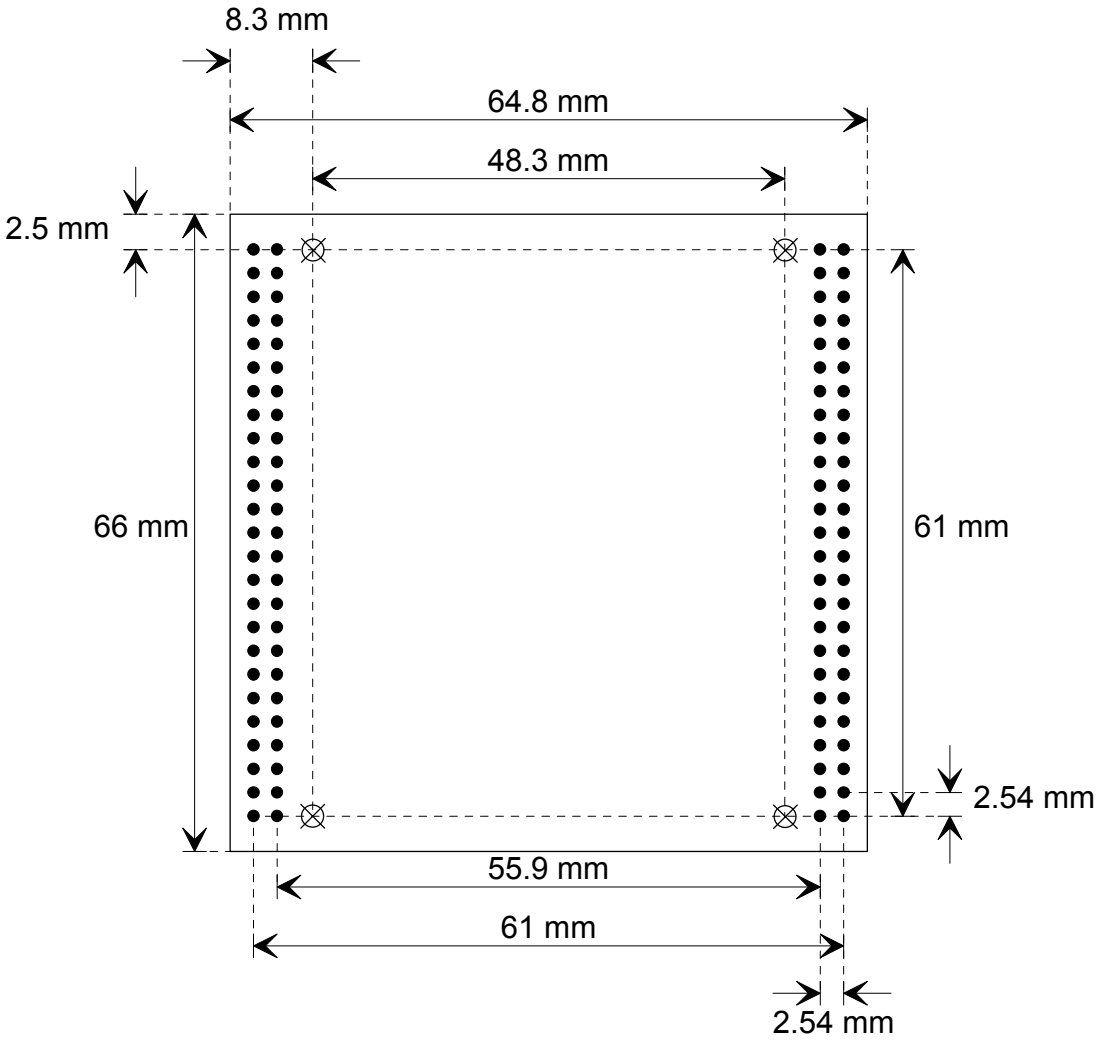


Figure 20 Mezzanine test board mechanical data.

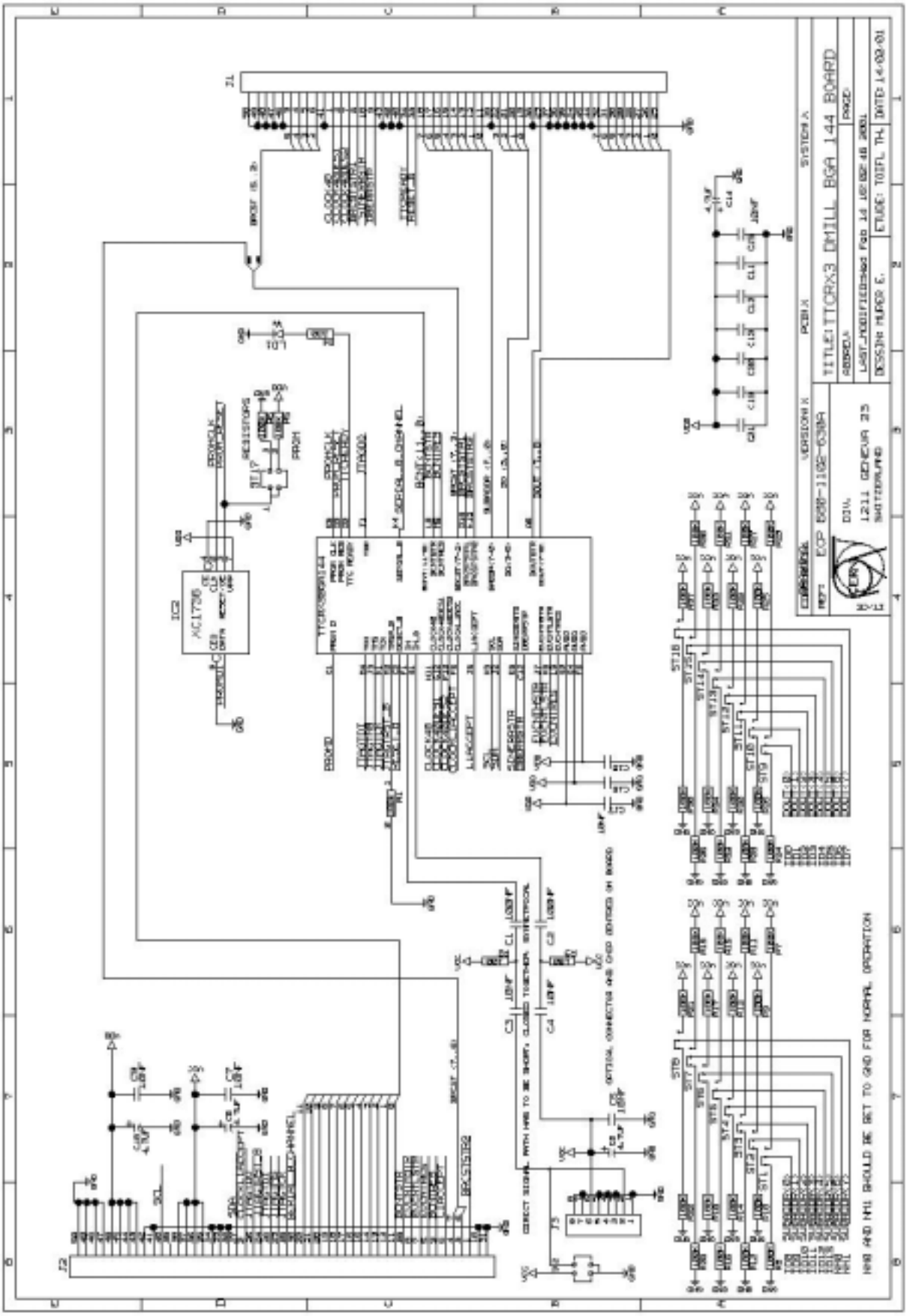


Figure 21 TTCrx test board schematic

Test Board Pin Assignment

Connector J1

Pin #	Name
1	Clock40
2	Clock40Des1
3	Brcst<5>
4	Brcst<4>
5	Brcst<3>
6	Brcst<2>
7	Clock40Des2
8	BrcstStr1
9	DbErrStr
10	SinErrStr
11	SubAddr<0>
12	SubAddr<1>
13	SubAddr<2>
14	SubAddr<3>
15	SubAddr<4>
16	SubAddr<5>
17	SubAddr<6>
18	SubAddr<7>
19	DQ<0>
20	DQ<1>
21	DQ<2>
22	DQ<3>
23	DoutStr
24	GND
25	Dout<0>
26	Dout<1>
27	Dout<2>
28	Dout<3>
29	Dout<4>
30	Dout<5>
31	Dout<6>
32	Dout<7>
33	Reset_b
34	TTCReady
35	GND
36	GND
37	GND
38	GND
39	GND
40	GND
41	GND
42	GND
43	GND
44	GND
45	GND
46	GND
47	GND
48	GND
49	GND
50	GND

Connector J2

Pin #	Name
1	BrcstStr2
2	ClockL1Accept
3	Brcst<6>
4	Brcst<7>
5	EvCntRes
6	L1Accept
7	EvCntLStr
8	EvCntHStr
9	BcntRes
10	GND
11	BCnt<0>
12	BCnt<1>
13	BCnt<2>
14	BCnt<3>
15	BCnt<4>
16	BCnt<5>
17	BCnt<6>
18	BCnt<7>
19	BCnt<8>
20	BCnt<9>
21	BCnt<10>
22	BCnt<11>
23	JTAGTMS
24	JTAGTRST_b
25	JTAGTCK
26	JTAGTDO
27	SDA
28	JTAGTDI
29	BCntStr
30	Serial_B_Channel
31	GND
32	GND
33	GND
34	GND
35	PIN_Preamp_VCC
36	PIN_Preamp_VCC
37	PIN_Preamp_VCC
38	PIN_Preamp_VCC
39	N.C.
40	SCL
41	GND
42	GND
43	TTCrx_VDD
44	TTCrx_VDD
45	TTCrx_VDD
46	TTCrx_VDD
47	GND
48	GND
49	GND
50	GND

Appendix C

Measurement results

This Chapter contains measurement results of the TTCrx.

MEASUREMENT SETUP

The TTCrx IC was tested using the test board described in the previous appendix. A TTC transmitter crate, controlled by a VME module, was used to deliver the optical signal to the test board. During the tests random data and triggers were generated and continuously sent to the IC. The experimental set-up is schematically represented in Figure 22.

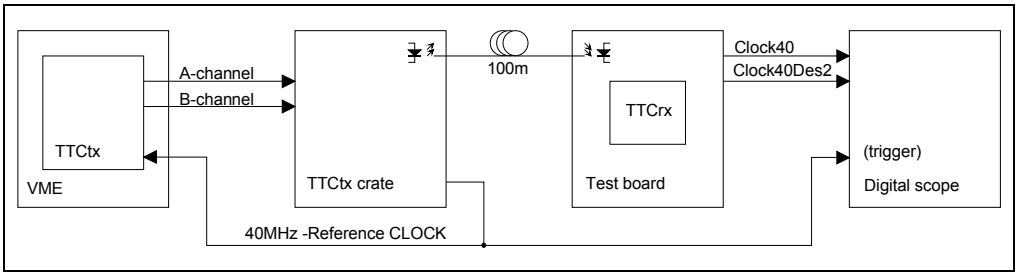


Figure 22 TTCrx test set-up.

Jitter

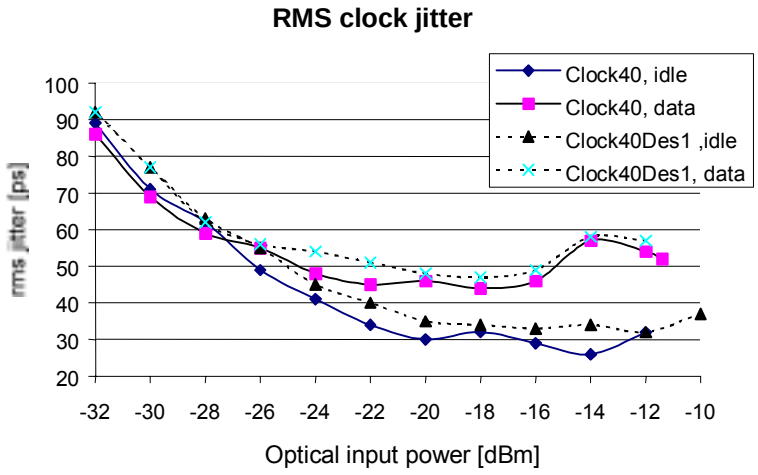


Figure 23 RMS Clock jitter versus input optical power.

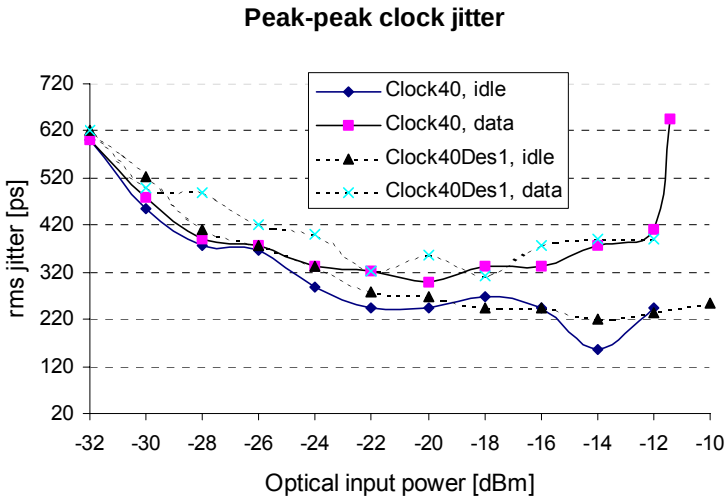


Figure 24 Peak to peak clock jitter versus input optical power.

Deskew function linearity

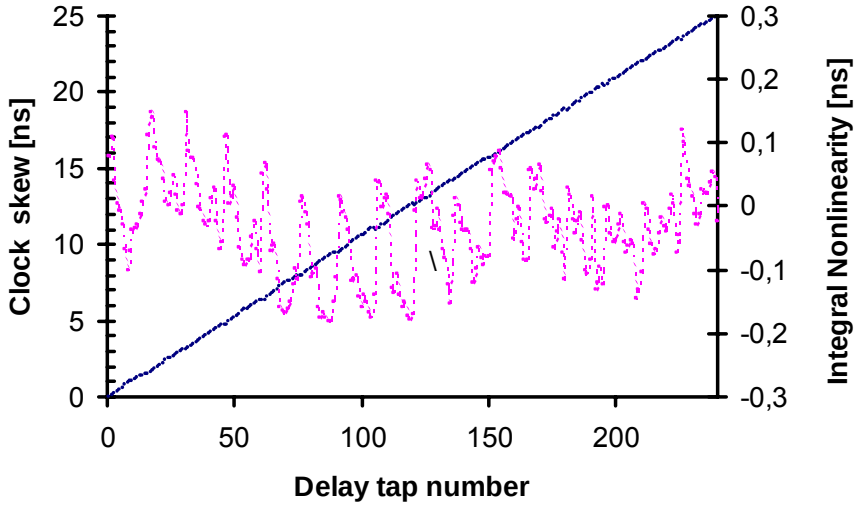


Figure 25 Measured delay as function of the programmed delay

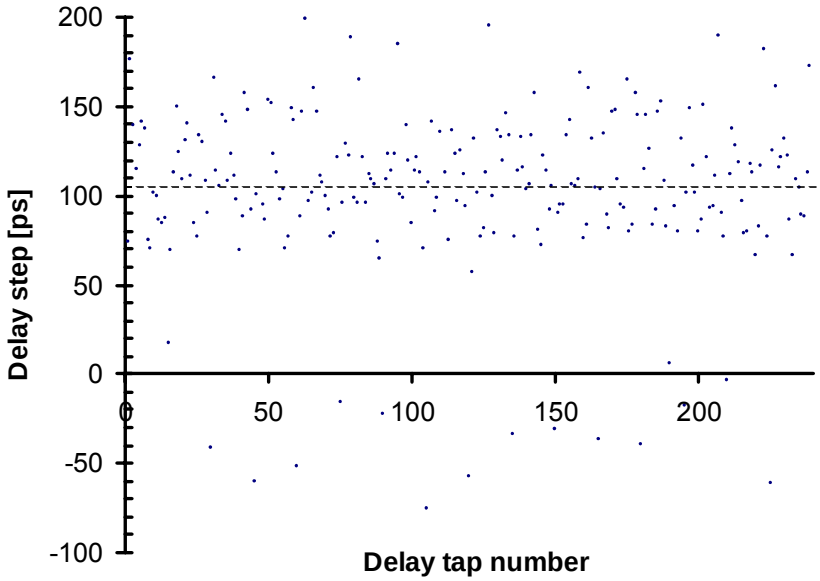


Figure 26 Measured differential non-linearity

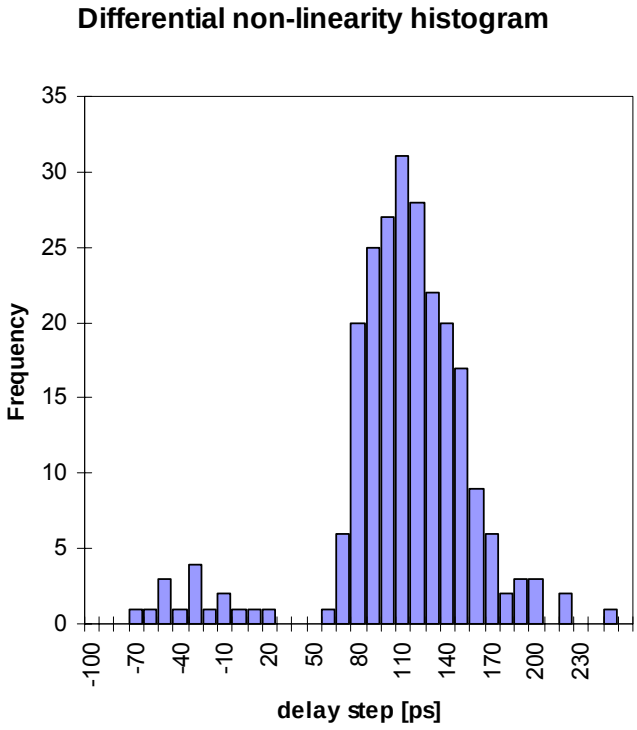
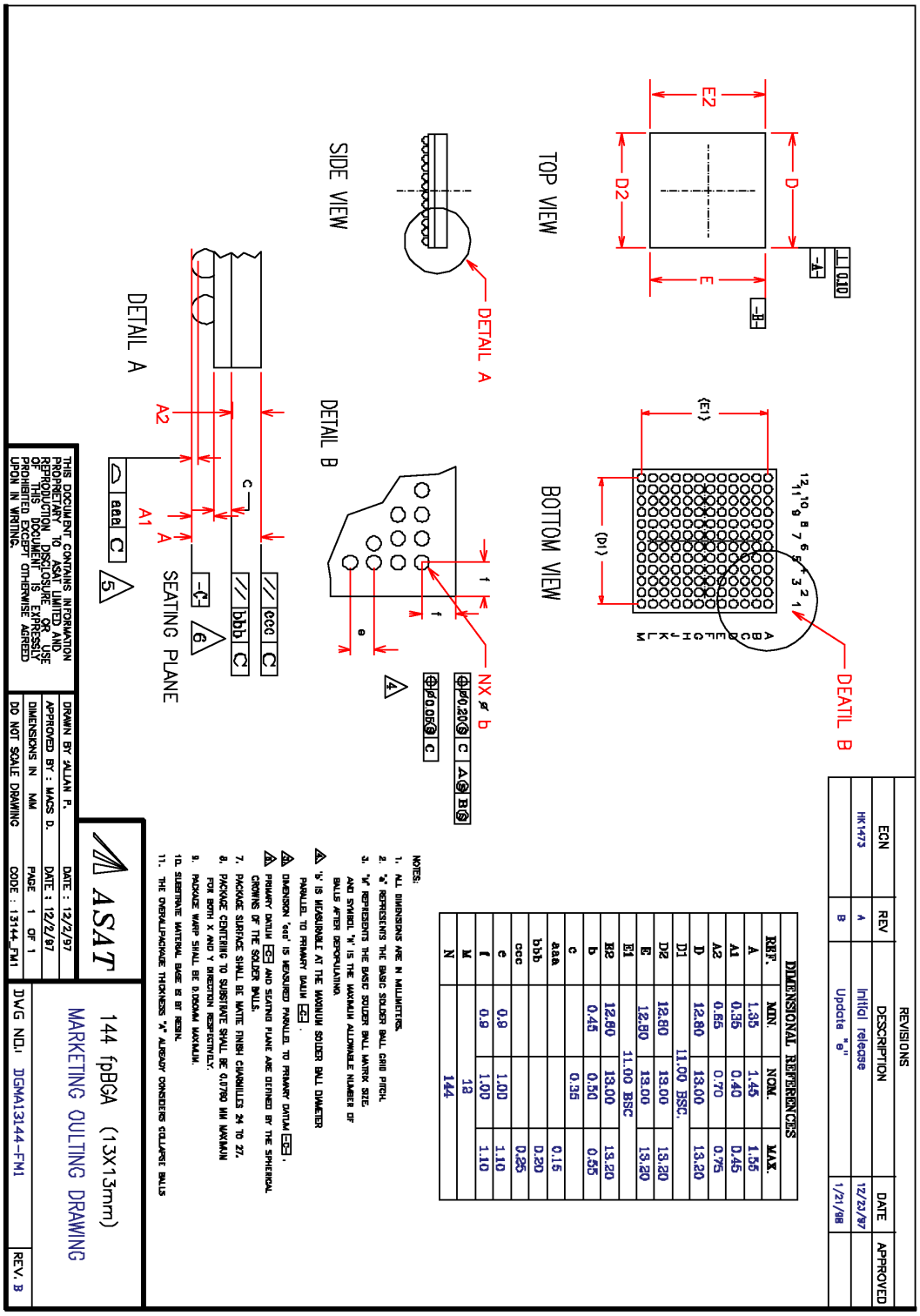


Figure 27 Measured differential non-linearity histogram

Appendix D

BGA 144 mechanical data



Appendix E

TTCrx known bugs

To date the following TTCrx bugs are known:

1. Trigger modes “00” and “10” do not work as specified. Please see Chapter 6 for a complete description of these operation modes and their associated problems.

References

[1] B. G. Taylor, “Timing, Trigger and Control (TTC) Systems for LHC Detectors,” CERN/EP: http://www.cern.ch/TTC/intro.html

[2] B.G. Taylor, "TTC Distribution for LHC Detectors," IEEE Trans. Nuclear Science, Vol. 45, No. 3, June 1998, pp. 821-828.

[3] J. Christiansen, A. Marchioro, P. Moreira and A. Sancho, “Receiver ASIC for Timing, Trigger and Control Distribution in LHC Experiments,” Presented at the IEEE Nuclear Science Symposium, San Francisco, 21-28 October 1995.

[4] H. Nussbaumer, “Computer Communication Systems: Data Circuits Error Detection Data Links,” Vol. 1, John Wiley & Sons, 1990.

[5] Xilinx, “The Programmable Logic Data Book,” 1993

[6] C. M. Maunder and R. E. Tulloss, “The Test Access Port and Boundary-Scan Architecture,” IEEE Computer Society Press, 1990

[7] Signetics Inc., I2C-Bus Specification, Application Note, Philips-Signetics, Jan. 1992

[8] T. Toifl, P. Moreira and A. Marchioro, "Measurements of Radiation Effects on the Timing, Trigger and Control Receiver (TTCrx) ASIC", T, Proceedings of the Sixth Workshop on Electronics for LHC Experiments, Kraków, 11-15 September 2000.