April 23 2003

Control program for the TTC setup at Heidelberg PI

R.Schwemmer

Abstract

The LHCb detectors Timing and Fast Control System relies on radiation hard on detector components and custom VME off detector electronics. The TTCrx Chip is a radhard Chip for on detector distribution of clock and trigger signals. In order to prototype the front end electronics, a VME interface for the TTCrx was developed by CERN. It comprises the TTCvi VME slave and the TTCvx clock generator and optical transmitter. The ttcvi-rx-ctrl program was written to be a user interface, which controls most of the functionality of the TTCvi. Additional routines where added to set the internal registers of the TTCrx chip via the optical fiber link. It was successfully used in a LHCb OT TDC setup.

Contents

1 Setup

A CES RIO 8062 Power PC, a TTCvi Mk II and an Atlas TTCvx where connected within a VME Crate. The Atlas TTCvx was used as fiber optics transceiver to send A and B-Channel commands to the TTCrm. The connection to the TTCrm used was a 50m suner fiber-optics optical fiber.

2 Usage

2.1 Installation

This program was tested and compiled on a CES RIO 8062 Power PC,running LYNX OS, using the CES VME libraries to communicate to the TTCvi.
Needed files/libraries to compile this program:

- ttcvi-rx-ctrl.c/ttcvi-rx-ctrl.h
- ttcvi.c, ttcvi.h
- f_getvme_addr.c, f_relvme_add.c,func_types.h
- menulib *http://hepwww.rl.ac.uk/Atlas-L1/Software/Tbdaq/MENU.HTML*
- libcurses
- libtermcap
- libces
- libvme
- libpthread

After you got all the needed libraries and files, do a *make* to compile the program.

2.2 Command Line Parameters

The program accepts one command line parameter, which is the base VME address of the TTCvi Interface(No preceding 0x).
If no Address is given, the standard address: 0x393A3D00 is used. [1]

[1]Rev. 1.6 of the TTCvi Manual stated this address to be 0x393A3D3E, which was found to be wrong.

2.3 TTCvi Control

2.3.1 L1A Trigger Select

In this section you can choose which source for the L1A trigger signals should be selected.
L1A <0..3> selects one of the four input channels on the front side of the TTCvi panel.
If you choose *Force Trigger*, an additional option will be shown, which allows you to send a L1 Accept signal by pressing the l key.
Random creates random L1A signals. The rate at which those signals are created can be set by choosing the Random Trigger Rate option in the Main Menu.
Calibr enables the generation of Calibration trigger pulses. For more information on how to use this mode, see the TTCvi Manual [1]
To disable L1A signal output, choose *disable*.

2.3.2 Random Trigger Rate

This option allows you to set the event rate of the TTCvi internal random event generator. Possible configurations are:
1 Hz, 100 Hz, 1 kHz, 5 kHz, 10 kHz, 25 kHz, 50 kHz, 100 kHz

2.3.3 Orbit Selection

The TTCvi can either use its internal orbit signal generator, or can use an orbit signal connected to one of the front panel NIM connectors. This option allows to select between *internal* and *external* orbit signals.¡Orbit rate¿

2.3.4 Event/Orbit Counter

The 24 bit internal Orbit/Event counter is displayed here. You can set it to either count *Orbits* or count *L1A Events* . Additionally the counter can be *reset*.

2.3.5 BC-Delay

The TTCvi normally receives its clock signal from an external source. In order to compensate different cable lengths, the phase of the input clock can be adjusted by a rotary switch on the front panel in 2 ns steps. This displays the actual delay of the received clock signal. You can use it to synchronize

A and B Channel output. Choosing this option will refresh all the currently displayed values in this menu.

2.3.6 Reset Module

This will reset the TTCvi Module and clear all non persistent counters and configurations. *Event/Orbit counter, the 4 B-Go fifos and their inhibit configurations*

2.4 TTCrx Control

This part of the program allows transmission of *predefined* and *arbitrary* commands via B channel cycles. All predefined Commands are sent using non synchronous B-Channel cycles.

2.4.1 Set TTCrx address

Every TTCrx responds to an unique 14 bit address. Here you can enter to which address predefined commands should be sent. The predefined commands are those, that allow you to manipulate the internal registers of the TTCrx and are summarized in *Set Internal Registers*. If you want to quickly configure all the TTCrx of your network, you can set the receiver address to 0000 to broadcast those commands.
When you change the receiver address, the locally buffered TTCrx settings are *not cleared* so you might want to issue a reset command to either upload your local data to the new address, or to clear the locally stored configuration. [2]
Note: there are only 14 bit for addresses, so the highest possible address is 0x3fff, which would be entered as 3fff, i.e. without the preceding 0x. You can enter higher numbers, but they will automatically be converted into the highest possible.

2.4.2 Set Internal Registers

For synchronization purposes the TTCrx contains 2 *fine delay registers*,which allow to adjust the phase of the recovered clock and can be accessed via 2 pins on the TTCrm board. Each signal can be delayed in 104.17 ps steps up to one full 25 ns clock cycle. [3]

[2]Possible bug: the TTCrx chip rev. 3.1 was found to need a power cycle reset, after this soft reset was issued

[3]The delay steps where found to be too big. Approx. 2-3 ns

Coarse Delay registers are used to delay the propagation of L1 Accept and Broadcast commands in steps of 25 ns. Coarse Delay 1 applies to the *L1Accept, BrcstStr1, BcntRes, EvCntRes, Brcst<5:2>*signals, while Coarse Delay 2 delays the signals *BrcstStr2* and *Brcst<7:6>*.
For more information about delaying clock cycles, coarse delays and how output signals are affected see Chapter 3 and 9 of the TTCrx Reference Manual [2]
Control Register:
This register allows the activation and deactivation of certain chip functionalities, which might not be used and can be turned off to reduce power consumption. When the program is started, the standard values for TTCrx rev. 3.1 are loaded into the local buffer. Those features are:

- Enable Bunch Counter Operation
- Enable Event Counter Operation
- SelClock40Des2 [4]
- Enable Clock40Des2 output
- Enable ClockL1Accept output
- Enable Parallel output bus [5]
- Enable Serial B Output
- Enable (non-deskewed) Clock40 output

2.4.3 Sending data

You can choose between uploading single commands, which are sent not synchronous to clock cycles using the *send single* option, or you can use the TTCvi's internal FIFO buffers:*Send Multiple Commands*, to program a command chain, then upload them into the TTCvi and have them sent synchronously to the TTCrx.

[4]Determines if Clock40Des1 or Clock40Des2 is used for synchronization of the BrcstStr2 signal

[5]This enables the Dout$< 7:0 >$, DQ$< 3:0 >$, SubAddr$< 7:0 >$ and DoutStr output pins

2.4.4 Sending a single command

Here you can enter any TFC Frame compatible command, which will then be sent asynchronously.
See section 2.4.6 for the syntax of those commands.

2.4.5 Issuing multiple commands

After choosing *Send Multiple Commands* you have an overview over the 4 TTCvi integrated FIFO buffers plus an additional program buffer, which you can also use to send preprogrammed, asynchronous command chains using the VME bus.
FIFOS <0..3>:
The TTCvi contains 4 32 bit wide, 256 commands deep FIFO buffers, which can be preprogrammed and then activated.
At first you should select a mode for the FIFO you want to use. See the TTCvi Reference Manual [1] for exact descriptions of the modes which can be applied to the different buffers. *It is mandatory, that you set at least an inhibit delay and duration, which are not zero, or the FIFOs will not be sent. Note: during the testing of this program, the Repetitive Mode didn't seem to work as anticipated. The contents of the FIFO were sent once, and then only empty commands i.e zeros where sent, after the read-pointer reached the end of the buffer.* After you have set the mode you want to use, you can start to enter commands using the *add commands* option. You can enter both short and long cycle commands.
See section 2.4.6 for the syntax of those commands.
After you are finished entering commands, you have to *Transfer* them to the TTCvi. If you choose to *don't look at the FIFO state*, everything will be sent with the next clock cycle, as soon as you transfer it to the TTCvi. Else the data is stored within the FIFO of the TTCvi and will not be sent unless you issue a B-GO signal.

Using the programs FIFO (Async VME Cycles):
This program part was added, after it was found, that the TTCvi FIFO didn't work as advertised. The used command syntax is the same as the one for the TTCvi FIFOs and can be found in section 2.4.6. The only mode you can choose here is how often your command chain should be sent(0 for infinite). After selecting how often they should be sent, enter the desired commands, and start the transfer using the s key. The program will spawn a new thread, which will then send the entered data using VME asynchronous cycles. The loop can be stopped anytime by pressing the s key again.

2.4.6 Long and short command syntax

short commands:
0<Data(2)>

- *0:* marks this as a short frame.
- *data <00-ff>:* the actual command.

Example: 012 sends a broadcast with HEX value 0x12.

long commands:
1<TTCrx address(4)>< E ><sub-address(2)><data(2)>

- *1:* marks this as a long frame.
- *TTCrx address <0000-3fff>:* a 14 bit identifier for the TTCrx you want to talk to, entered in hexadecimal format without preceding 0x.
- $E < 1|0 >$*:* decides if this is an external or internal command. Use 1 for external and 0 for internal commands.
- *sub-address <00-ff>:* the sub-address of the register you want to access.
- *data <00-ff>:* the actual command.

For example a command that would switch on all functionalities of the control register on all TTCrx would be:
10000030f
There is no syntax check, so you can enter everything. The parser however will decide from the first number entered, if it is a long or a short frame and then interpret all following numbers, corresponding to that bit as short or long frames and ignore everything else.

2.4.7 TTCrx Reset

[6] Since there is no way for the TTCrx to communicate back to the TTCvi the program uses an internal buffer, to keep track of the changes that have occurred within the TTCrx registers. To keep synchronized with the TTCrx chips, you can issue various reset commands to the program and the TTCrx. Here, local means the running ttcvi-rx-ctrl program, and remote denotes the TTCrx chips.

[6]Possible bug: The TTCrx chip rev. 3.1 was found to need a power cycle reset, after this soft reset was issued

Complete Reset will send a reset signal to the TTCrx address given in the TTCrx Control menu, and will also reset the locally buffered settings to the standard settings.
Reset and upload local settings will send a reset signal to the chip and upload the locally stored settings afterwards. This is useful, if you want to upload a certain configuration to different, but not all of your TTCrx.
Reset local settings only loads the defaults into the local buffer. It is also useful if you changed the receiver address, and don't want to keep the local buffered settings.

2.5 TFC Broadcasts

Following the TFC Broadcast Format proposal[3] , additional options have been added, to send the broadcasts specified. Those commands can also be sent under the arbitrary commands section, but are also implemented as menu options.

2.5.1 L1 Decision Broadcasts

These broadcasts are used to convey the results of the Level-1 trigger to all the Level-1 electronics[3]
In Order to supply the Broadcast with the needed lower order 2 bits trigger number for the event, the TTCvi's internal Event/Orbit Counter is used as trigger number here. Two additional options allow to set the Event and Bunch Counter reset bits in the broadcast package. The reserved options haven't been defined yet, the corresponding bit combination however is already available and can be sent by this program.

2.5.2 Synchronous Commands

Again, the Event and Bunch Counter bits can be set additionally, also the reserved options are not yet defined, but the corresponding bit combinations will be sent.
When issuing a *Calibration Pulse Injection*, a *Calibration Trigger* is automatically sent afterwards.

References

[1] Ph. Farthouat, TTC-VMEbus Interface TTCvi - MkII Manual

[2] Jorgen Christiansen et al., TTCrx Reference Manual, A Timing, Trigger and Control Receiver ASIC for LHC Detectors

[3] Beat Jost, TFC Broadcast Format (Proposal) LHCb 2001-17