ATLAS Level-1 Calorimeter Trigger 25 August 2005
Software Note 002 (EDMS document number 633166, EDMS ATL-DA-ES-0047)
Version 1.0

Use of TTC System

L1Calo Group [1]

1 Introduction

This document sets out some suggestions for the use of the TTC system [1] in the Level 1 Calorimeter Trigger [2] and in particular lists the TTC broadcast commands we may use. A scheme for assigning TTCrx chip addresses is also proposed.

The Calorimeter Trigger contains of a number of different types of crate connected to the TTC system. In all of them, the TTC fibre goes to a TCM which fans out the TTC signals electrically along a custom backplane. However in the present design, the TCM itself does not include a TTCrx chip.

- Preprocessor crates containing PPMs
- Cluster processor crates containing CPMs and CMMs
- Jet/energy processor crates containing JEMs and CMMs
- Readout driver crates containing RODs (and maybe a CERN BUSY module)
- The TTC crate containing the TTCvi, LTP and one or more BUSY modules

Additionally, in test setups, we may have:

- DSS crates containing DSS modules (which receive individual TTC signals, not via the TCM)
- LVDS source modules (LSMs) for testing CPMs and JEMs.

The TileCal (and LAr) receiver crates will probably not be connected to the TTC system.

Apart from the trigger crates themselves, the DAQ readout systems (ROS) connected to our RODs will also be connected to our TTC partition.

[1] Please send any comments and corrections to Murrough Landon.

2 TTCrx Addresses

Each TTCrx chip in one TTC partition should have a unique 14 bit address. We will probably only use TTC broadcasts and would therefore not need to address individual chips. Nevertheless it is probably wise that we still ensure that each TTCrx chip does indeed have a unique address.

It is proposed that we use geographic addressing within our VME crates to assign not only the VME address of a module, but also the TTC address of its TTCrx chip (and the CANbus address). The TTCrx chip reads its address from its data and command pins when a reset is issued to the chip.

However, the TTCrx chip also derives its I2C bus address at reset from the lowest six bits of the 14 bit address. These six bits are variously referred to as I2C_ID or ID_I2C in the TTCrx documentation [3]. They form the top six bits of the two consecutive seven bit I2C addresses used by the TTCrx. It seems desirable that all the TTCrx chips on a given type of module have the same I2C bus addresses – especially since there may be other devices on the same I2C bus.

The geographic addressing scheme, at least in the CP and JEP crates uses the minimum number of backplane pins and relies on each type of module "knowing" what type it is. Some of this extra module type information will need to be supplied to form the TTCrx address.

The main part of the trigger processor system consists of sixteen crates: 8 PP, 4 CP, 2 JEP, 2 ROD. The two TileCal receiver crates will probably not be connected to the TTC system. However the Readout Systems (ROS) to which our RODs are connected **will** need to be part of the same TTC partition. Hence we should allow for more than sixteen "crates" in the system. So the TTCrx address will need five rather than four bits for crate identification (may be a mixture of crate number and type).

We will generally have more than sixteen modules per crate, so again five bits of TTCrx address may be required, perhaps only in some types of crate.

Clearly ten bits of crate/module identification does not allow the low six bits to be fixed separately. There will have to be some overlap between the crate/module address bits and the I2C address bits.

After some discussion, the proposal is to use three of the lowest six bits (ie the ID_I2C part) to uniquely identify each type of module. The upper eight bits can then be used to identify the crate and module number within that crate. Given that the crate type is generally defined by the types of module it contains and since we dont actually expect more than 16 modules of one type per crate, four bits for the crate and four bits for the module could in practice be sufficient.

However there is no requirement that the usage of the eight bits should be the same for different types of module. The PPM and ROD crates will use standard VME64x geographic addressing with five bits per slot, presumably with some additional private crate addressing scheme using three bits. Whereas the CP and JEP crates use a private scheme with up to four bits per slot and three bits per crate.

Modules, such as the DSS, which live in standard 6U VME crates without geographic addressing cannot have their TTCrx address derived automatically from the VME backplane. The address has to be configured by an EPROM and in principle could be anything that doesnt clash with the geographically addressed module types. However a range for the DSS is suggested for definiteness.

An addressing scheme along these lines is shown in table 1.

TTCrx Address	Module Type and Addresses	ID_I2C
`ccc nnnnn 000001`	PPM: slot nnnnn=0–31 in crates ccc=0–7	1
`00cc nnnn 000010`	CPM: module nnnn=1–14 in crates cc=0–3	2
`010c nnnn 000011`	JEM: module nnnn=0–15 in crates c=0–1	3
`0ccc 000n 000100`	CMM: module n=0–1 in crates ccc=0–5	4
`xxxx xxxx 000101`	TCM (only for CAN, no TTCrx)	5
`00c nnnnn 000110`	ROD: slot nnnnn=0–31 in crates c=0–1	6
`xxxx xxxx 000111`	DSS: with arbitrary xxxx xxxx, eg serial no.	7
`xxxx xxxx 001000`	LSM: with arbitrary xxxx xxxx, eg serial no.	8
`xxxx xxxx xxxxxx`	Reserved for ROS etc	>8

Table 1: TTCrx address scheme: `c` are crate address bits; `n` are module address bits. ID_I2C is the top six bits of the I2C bus address. This is also used as the module type component of the CAN address.

The variable length crate address that table 1 assumes is compatible with the numbering of all the crates in the processor system as shown in table 2.

Crate	Range	Crate Type
`0ccc`	`ccc=0-7`	PP crates 0–7
`10cc`	`cc=0-3`	CP crates 8–11 (0–3 in CP/JEP GEOADD scheme)
`110c`	`c=0-1`	JEP crates 12–13 (4–5 in CP/JEP GEOADD scheme)
`111c`	`c=0-1`	ROD crates 14–15

Table 2: Crate numbering for the whole processor

3 Broadcast Commands

The TTC system can either address individual TTCrx chips, or else it can send broadcasts to the whole system. There are no intermediate subsets, so we can't subdivide our system into the three component processors – without using three TTCvi modules which seems excessive.

There are two broadcast command formats: long and short. The short format sends a single 8-bit word. Two of these bits are reserved for the event and bunch counter resets. The long format sends an 8-bit word together with an additional 8-bit subaddress.

In the list below we can identify requirements for only a few TTC commands so the short format would appear to be sufficient. However as the long and short commands appear on separate pins on the TTCrx, modules should track both sets of pins for future proofing.

The upper two command bits come with a separate strobe (and separately configurable delay) from the lower six bits. As the DSS only uses the two upper bits, any commands to the DSS may use only these two. Commands to other modules should set the upper two bits to zero.

3.1 Summary of Requirements

The required commands are summarised as follows:

- Starting and stopping synchronous playback of data from various playback memories. The modules must already have been loaded with playback data. For most modules (except the DSS) only the start command is important and it just resets the read pointer to the playback memories. For the DSS, the command switches the module into or out of playback mode.
  Mandatory.

- Start of synchronous event fragment capture in the 9U RODs. The RODs must already all have been configured with the same criteria for selecting events and all buffers cleared.
  Mandatory.

- One JEM specific command for enabling writing into the spy memories. In the JEM, unlike other modules, the spy memories do not wrap and are not overwritten once filled.
  Mandatory

The following commands have been discussed and might be useful:

- Starting and stopping the synchronisation of the LVDS links. This tells the PPMs to start sending the AA55 synchronisation pattern from their LVDS transmitters. It may not be **essential** that this is done synchronously, but it would be very convenient. It may also be useful for the LVDS source modules (LSMs).
  Probably desirable

Several other possible commands were discussed in the past but are not considered useful or appropriate and have been removed from the document.

3.2 Command Assignments

Given the restrictions imposed by the DSS, the TTC command assignments are as shown in table 3. The two upper bits identify DSS or global commands. The `mmmm` and `xx` bit fields are independent and can be combined arbitrarily. The two `xx` bits are reserved by the TTC system for bunch and event counter reset commands. These two bits will normally, but not necessarily, be zero when L1Calo specific TTC commands are sent. The four `mmmm` bits are discussed below.

Command bits	Command meaning
`00mmmmxx`	Reserved for PPM/CMM/CPM/JEM/ROD, ignored by DSS
`01mmmmxx`	Global start playback (reset playback pointer)
`10mmmmxx`	Global stop playback (only used by DSS?)
`11mmmmxx`	Reserved for future DSS or global command

Table 3: Bit fields in TTC commands: `mmmm` = command to module, `xx` = reserved (ECR,BCR)

There are 15 possible non-zero `mmmm` values available. Zero needs to be reserved for bunch and event counter reset commands to be sent on their own. The non-zero values are used for commands to one or more specific module types and will be ignored by the DSS. Some values have been provisionally assigned according to the requirements in section 3.1 and are shown in table 4.

4 Implementation

In order for all the boards with TTCrx chips to respond to these TTC commands it is clearly essential that the TTCrx command pins be wired to some control FPGA

mmmm bits	Command meaning
0000	Reserved for separate BCR and ECR commands
0001	ROD: start synchronous event fragment capture
0010	PPM (and LSM): stop LVDS sync pattern
0011	PPM (and LSM): start LVDS sync pattern
0100	JEM: clear and enable spy memories
0101	Unused
0110	Unused
0111	Unused
1000	Unused
1001	Unused
1010	Unused
1011	Unused
1100	Unused
1101	Unused
1110	Unused
1111	Unused

Table 4: Commands to specific module types

on the board which can then take the relevant actions. For flexibility all the eight short format command output pins should be available. And for future proofing, all eight (separate) long format command output pins should also be tracked. *NB the DSS only uses two pins of the short format and will not be changed in future. The JEM0 doesnt receive short broadcasts but will soon be obsolete.*

All the actions that can be initiated by a TTC broadcast should also be available as VME commands, eg through some "pulse" type control register. It is suggested that in case of a conflict, the TTC command should take precedence.

From the software point of view, it would be useful if all the modules in the system presented the same programming model for access to the 20 internal registers on the TTCrx chips. A scheme [4] for doing this, requiring a command access followed by polling a status register for completion, has been proposed by Richard Staley. Direct memory mapping is undesirable as the I2C transfer takes 380μs to complete which is a long time to hold the VME bus.

References

[1] TTC home page
http://ttc.web.cern.ch/TTC/intro.html

[2] ATLAS Level 1 Calorimeter Trigger home page
http://hepwww.pp.rl.ac.uk/Atlas-L1

[3] TTCrx specification
http://www.cern.ch/TTC/TTCrx_manual3.2.pdf

[4] Proposed programming model for TTCrx
http://www.ep.ph.bham.ac.uk/user/staley/I2C_Controller.pdf