Muon g–2 Experiment

Clock & Controls Center (CCC) Manual

May 2016

This user guide is for the Clock & Controls Center (CCC) for the Muon $g-2$ experiment at Fermi National Accelerator Laboratory. The CCC was developed to distribute a high-precision clock and synchronization signals to the experiment's various subsystems. For information on the experimental requirements and design choices, please refer to the appropriate sections of the Technical Design Report.

For questions or suggestions, please contact the principle developer:

Sweigart, David (Cornell University)

Contents

***

Acronyms

***

AMC	Advanced Mezzanine Card
BME	Biphase Mark Encoding
CCC	Clock and Controls Center
DAQ	Data Acquisition
EEPROM	Electrically Erasable Programmable Read-Only Memory
FMC	FPGA Mezzanine Card
FPGA	Field-Programmable Gate Array
GbE	Gigabit Ethernet
IP	Internet Protocol
IPMI	Intelligent Platform Management Interface
JTAG	Joint Test Action Group
LED	Light-Emitting Diode
LSB	Least Significant Bit
LVCMOS	Low-Voltage Complementary Metal Oxide Semiconductor
LVDS	Low-Voltage Differential Signaling
MAC	Media Access Control
MCH	μTCA Carrier Hub
MIDAS	Maximum Integrated Data Acquisition System

MMC	μTCA Management Controller
MSB	Most Significant Bit
PLL	Phase-Locked Loop
PM	Power Module
SPI	Serial Peripheral Interface
TCA	Telecommunications Computing Architecture
TCP	Transmission Control Protocol
TTC	Timing, Trigger, and Control
TTS	Trigger Throttling System
WFD	Waveform Digitizer

Configuration Guide

***

FC7 Configuration

***

1.1 Configuration Parameters

The FC7 needs to switch automatically among different modes, according to a predetermined sequence. This sequence will be configurable via IPbus with a dedicated slave.

The switching among modes primarily refers to changing trigger modes, such as muon fills, laser runs, pedestal runs, and asynchronous crate readouts. For the Rider, this is accomplished by changing the TTC Channel B command with or prior to the TTC Channel A trigger. For the laser system, this is accomplished by outputting a trigger signal after a set delay for only laser runs.

A diagram of such a trigger sequence scheme is shown in Figure 1.1. Each step of the sequence is defined by a trigger index, trigger type, and pre-trigger gap. These three parameters are defined in Table 1.1. Note that each sequence is synchronous to the accelerator trigger, and, at the end of the sequence, the FC7 will wait for the next accelerator trigger.

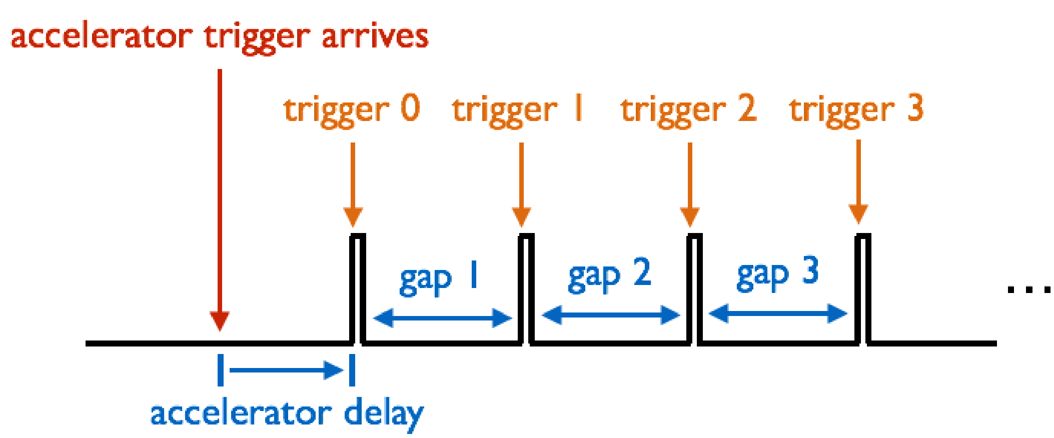

Figure 1.1: FC7 trigger sequence scheme.

Parameter	Description
Trigger Index	Index of trigger in sequence, starting at zero.
Trigger Type	Type of trigger (muon, laser, pedestal, asynchronous readout).
Pre-Trigger Gap	Number of 40-MHz TTC clock cycles to wait before issuing trigger.

Table 1.1: FC7 triggering parameters that are configurable via IPbus.

It would also be possible to define multiple trigger sequences. This would require similar parameters to determine the order and rate that each sequence is executed. A multi-sequence behavior could be advantageous for certain triggers, such as the asynchronous crate readout, that need to be issued but not very often.

In addition, multiple trigger sequences will be integral to running the $g-2$ experiment, simply because of the bunch structure within each super-cycle. In each super-cycle, there will be two groups of eight muon bunches that are separated by 10 ms. There is a 197 ms gap after the first bunch group and a 1063 ms gap after the second bunch group. We will definitely want to execute a difference set of sequences during the three different gaps (10 ms, 197 ms, 1063 ms).

1.2 Configuration Logic

The implementation of trigger sequences should be straightforward. A dedicated IPbus slave should be created that holds a memory of 32 32-bit registers. These registers will hold the trigger type and pre-trigger gap for each index of the sequence. The total number of indices in the sequence will also need to be specified in one of the registers.

These IPbus registers will then be used in a finite state machine. An idea of what this state machine might look like is pictured in Figure 1.2. The state machine will be initialized to the `IDLE` state, where it waits for the (already delayed) front panel trigger. When it sees the trigger, it will transition to the `SEND_TRIGGER` state, which sets the TTC Channel A to issue a trigger.

If that was the last trigger in the sequence, the state machine transitions back to `IDLE`. If there is another index in the sequence, it will instead transition to the `SEND_COMMAND` state. It stays here until the proper TTC Channel B command is issued, setting the next trigger type, and, when done, moves to `WAIT`. The state machine will then pause until the pre-trigger gap period has expired. At that point, it will move back to the `SEND_TRIGGER` state and repeat the logical process.

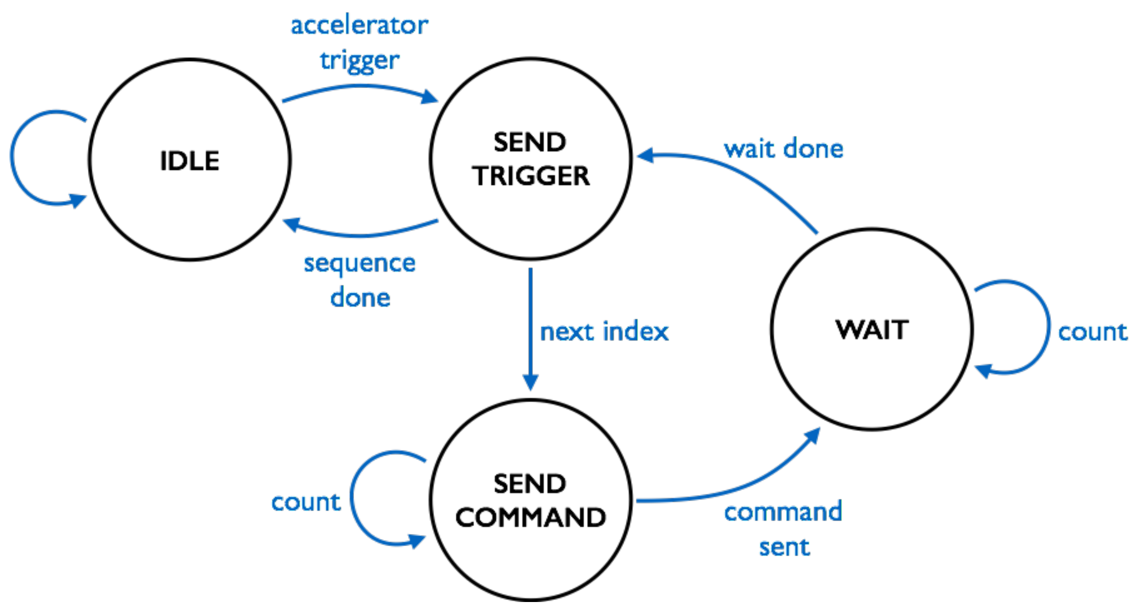

Figure 1.2: State machine diagram for implementing FC7 trigger sequence.

1.2.1 Timing Specifications

Recall that the TTC data is encoded in two alternating channels. The Channel A encodes the 1-bit trigger signal, while the Channel B encodes the 8-bit broadcast command with an 8-bit overhead. In each 40-MHz clock cycle, one bit of each channel is sent. This means that 16 clock cycles are required to send the command to switch fill types and, therefore, the minimum delay between different trigger types is 425 ns, after accounting for the one clock cycle enforced by the `WAIT`.

In the FC7 firmware, caution must be taken to ensure proper synchronization of signals. This is especially important the signals cross clock domains, which is the case for all signals going from the 125-MHz IPbus clock domain and to the 40-MHz trigger logic clock domain.

1.2.2 Broadcast Commands

The Channel B of the TTC protocol encodes an 8-bit broadcast command. In the calorimeter system, the broadcast commands are used to switch trigger types (muon fill, laser, pedestal, asynchronous readout) on the waveform digitizers, i.e., Riders. The valid broadcast commands that are recognized by the Riders are listed and described in Table 1.2. Note that the *Event-Count Reset* and *Counter Reset* commands are also recognized by the AMC13.

Command	Broadcast [7:0]	Meaning
Event-Count Reset	`xxxxxx1x`	Reset event counter
Counter Reset	`001x1xxx`	Reset 44-bit counter used for timestamps
Async. Readout Trigger Type	`100x0xxx`	Switch to “Async. Readout” trigger type
Muon Trigger Type	`101x0xxx`	Switch to “Muon Fill” trigger type
Laser Trigger Type	`110x0xxx`	Switch to “Laser” trigger type
Pedestal Trigger Type	`111x0xxx`	Switch to “Pedestal” trigger type
Start Async. Pulse Storage	`100x1xxx`	Start accepting front panel triggers
Stop Async. Pulse Storage	`101x1xxx`	Stop accepting front panel triggers

Table 1.2: Valid broadcast commands recognized by the Riders.

Data Collection Procedures

***

Data Formats

***

2.1 AMC13 Data Format

The FC7 outputs its data to the AMC13 over the μTCA backplane link. For each TTC trigger, the AMC13 collects the “data payloads” from all enabled FC7 boards and packages them into the format shown in Figure 2.1. This procedure basically prepends and appends so-called CDF and Payload Block header and trailer words, respectively.

Note that, if the FC7 data blocks become sufficiently large, the BU firmware will divide the payload into blocks of 4096 64-bit words. When this division occurs, each individual block will be flanked by a payload block header, an AMC header, and a payload block trailer. See the AMC13 specifications for more detailed information.

It is also important to note that the AMC13 is big-endian device. That is, bits are stored in its memory and transmitted with the smallest address to the left, such as in the figures shown in this guide. All data words sent to thru the AMC13 will to adhere to this convention.

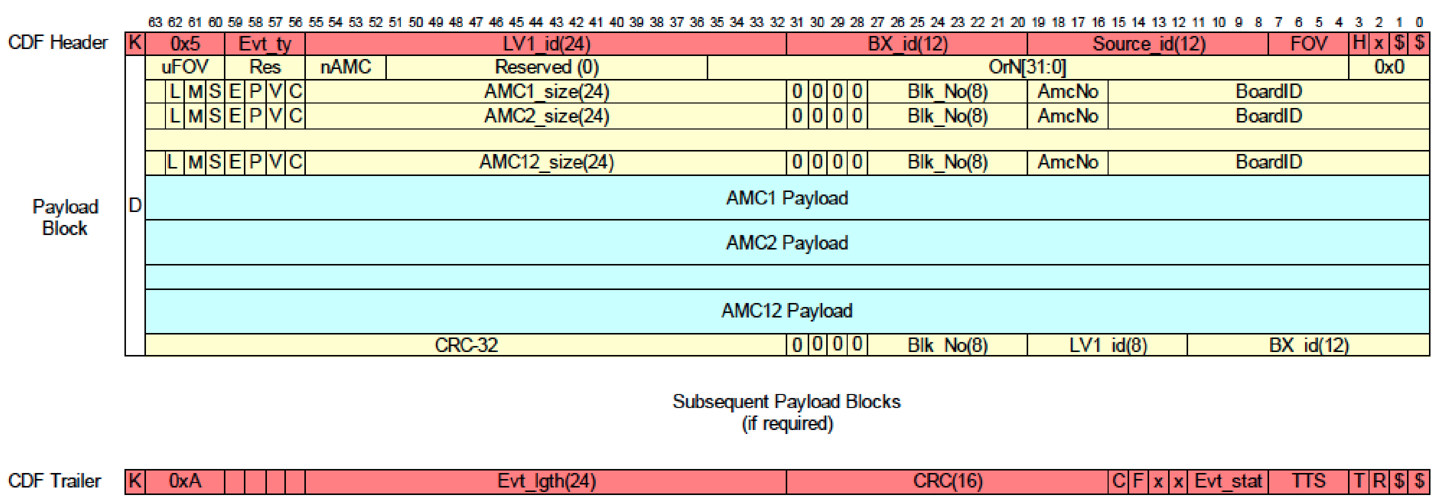

Figure 2.1: AMC13 to DAQ data format.

2.2 FC7 Data Format

The AMC13 imposes further specifications on the format of the data sent from the FC7. In particular, it requires its own two headers and one trailer to surround the FC7 data.

The format of the FC7 data block differs slightly depending on whether the FC7 is for TTC encodement or fanout, as shown in Figures 2.2 and 2.3, respectively. A brief description of the main parameters is given in Table 2.1.

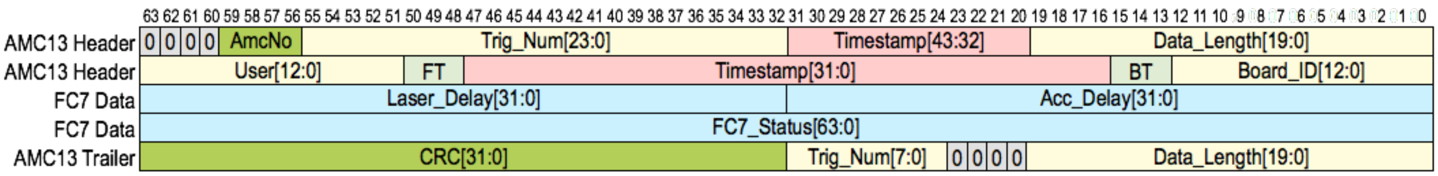

Figure 2.2: Encoder FC7 to AMC13 data format.

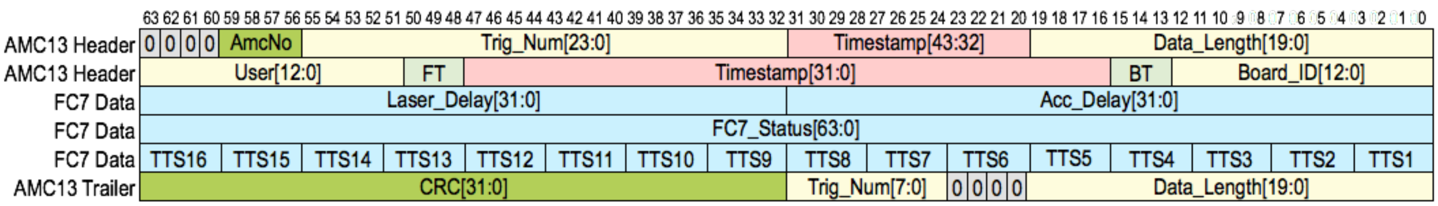

Figure 2.3: Fanout FC7 to AMC13 data format.

Parameter	Description
AmcNo	AMC slot number occupied by the FC7, indexed from 1 to 12. Inserted by the AMC13.
Trig_Num	TTC trigger number. This is not the same as a "muon fill number" because laser and pedestal triggers will also be included in the overall count. If needed, a separate fill counter for muon fills could incorporated and stored in the User parameter.
Timestamp	There is a 44-bit counter that is incremented in the FC7 on the rising edge of the 40-MHz TTC clock. This field is the value of that counter when each TTC trigger is received off the μTCA backplane. Note that, with a 44-bit counter incrementing every 25 ns, it will not need to be reset for up to 5 days. If needed, this counter can be setup to be reset for every run.
Data_Length	Number of 64-bit words in the event, including the two AMC13 headers and one AMC13 trailer.
User	13 bits left available for the user.
FT	3-bit fill type, which will be 001 for a muon fill, 010 for a laser fill, and 011 for a pedestal fill.
BT	3-bit board type, which will be 001 for the Rider, 010 for the CCC FC7, and 011 for the tracker FC7.
Board_ID	Unique serial number for the FC7 baseboard. This number should be stored in an EEPROM, if available, and used to assign the MAC address. The MSB is reserved to distinguish between the TTC encodement (0) and the TTC fanout (1) FC7 boards.
Laser_Delay	Delay set for the output trigger to the laser system, in units of the number of 40-MHz TTC clock cycles.
Acc_Delay	Delay set between the received accelerator trigger and the output TTC trigger, in units of the number of 40-MHz TTC clock cycles.
FC7_Status	64 bits available for the important FC7 status signals. This will include internal errors, such as from the SFP transceivers, and the trigger sequence index number.
TTS#	Latest TTS state received from the given SFP receiver.
CRC	Checksum inserted by the AMC13 for an offline data integrity check.

Table 2.1: FC7 parameters that appear in its output data format.