FERMILAB-TM-2458-PPD

VME Data Acquisition Modules for MINERvA Experiment

Boris Baldin

Fermi National Accelerator Laboratory, Batavia, IL 60510

Abstract

This document describes two VME modules developed for MINERvA experiment at Fermilab. The Chain ReadOut Controller (CROC) module has four serial data channels and can interface with up to 48 front-ends using standard CAT5e networking cable. The data transmission rate of each channel is 160 Mbit/s. The maximum data transmission rate via VME bus is ~18 MB/s. The Chain Readout Interface Module (CRIM) is designed to provide various interface functions for the CROC module. It is compatible with MINOS MTM timing module and can be used to distribute timing signals to four CROC modules. The CRIM module also has a data port compatible with the CROC serial data interface. The data port can be used for diagnostic purpose and can generate triggers from front-end events. The CRIM module is a standard D08(O) interrupter module.

I. MINERvA Chain Readout Controller

1. Introduction

MINERvA Chain Readout Controller (CROC) is dedicated to provide timing, slow control and data readout for front-end modules based on 64-channel multi-anode photomultiplier tube. The front-end modules are connected to each other in a chain using commercial networking cable. Both ends of the chain or DAQ loop are connected to the Chain Readout Controller. The Chain Readout Controller is a VME based module which resides in a VME crate controlled by a PC via commercial PCI-VME interface or by an embedded VME controller. The Chain Readout Controller receives a set of MINOS timing signals from a modified version of the MINOS MTM module [1].

2. DAQ loop

The DAQ loop consists of up to 12 front-ends. The number of front-ends in the loop and the total cable length are limited by the degradation of the timing signal which is common for all front-ends in the loop. The maximum length of the loop should not exceed 60 ft. A diagram of the DAQ loop signals is shown in Fig. 1.

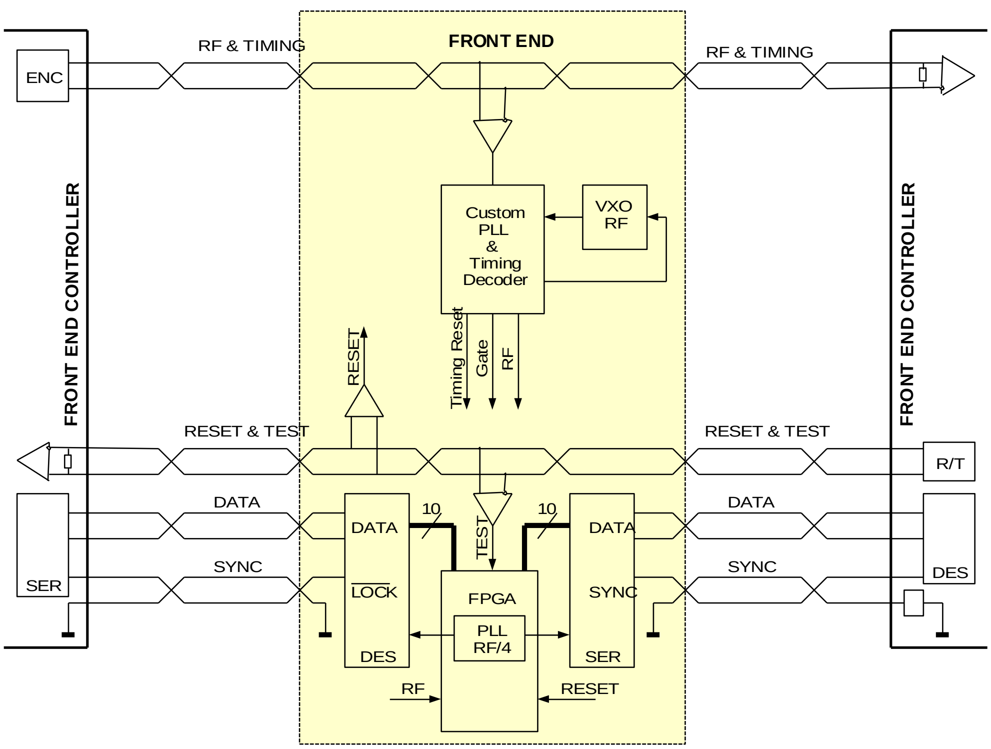


Fig. 1 Diagram of the DAQ loop signals

There are four signals in the DAQ loop. Two of the signals "RF & Timing" and "Reset & Test" are bussed (multi-drop) signals that propagate through the front-ends via continuous twisted pair. There is one driver and one receiver at the controller side and multiple receivers in the front-ends for these signals. The other two signals "Data" and "Sync" are chained (point-to-point) signals that are received and transmitted by each front-end. "RF & Timing" and "Data" propagate in one direction, and "Reset & Test" and "Sync" propagate in the opposite direction.

The "RF & Timing" signal is an accelerator RF clock signal with embedded encoded timing signals. This signal is generated by a standard LVDS driver and passes through all front-ends in a loop until it reaches an LVDS receiver at the controller. The implementation of the timing encoding is similar to the D0 muon timing encoding scheme developed by Sten Hansen [2]. A timing diagram of the encoded signals is shown in Fig. 2. The FCMND signal is an example of a software generated timing command.

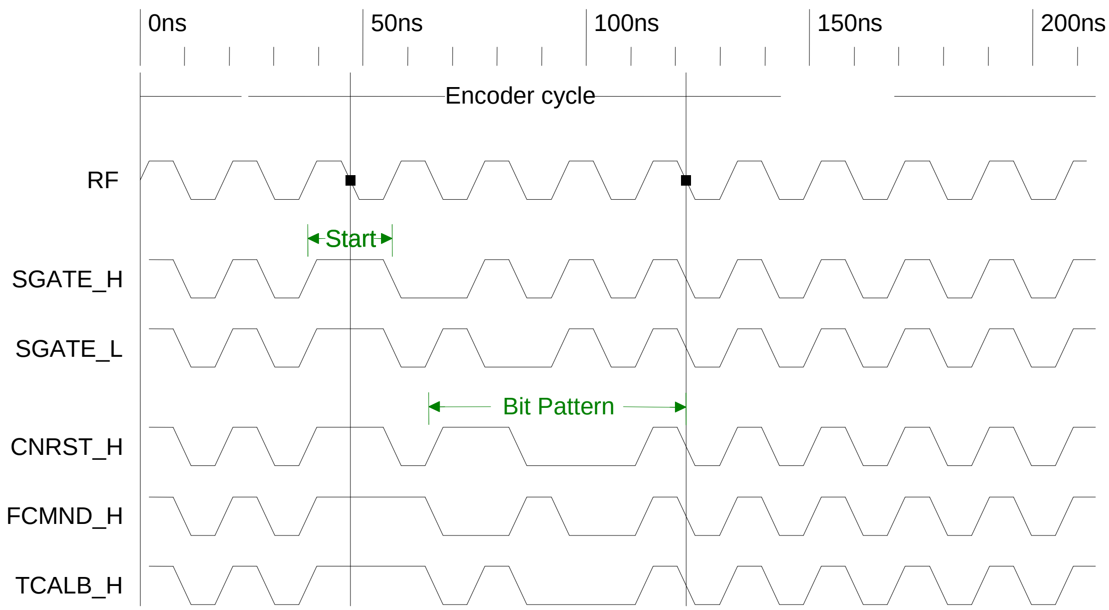


Fig. 2 Timing Encoder signals

The encoded timing signal always starts with two consecutive ones and ends with a one. A seven bit pattern follows the start bits. Note that not all bit combinations are acceptable. Table 1 shows actual binary vales of the encoded MTM signals.

Table 1 Encoded timing signals

No.	Signal Name	Binary Value	Hex Value	Comment
1	SGATE_H	10110001	0xB1	Rising edge
2	SGATE_L	11010001	0xD1	Falling edge
3	CNRST_H	11000101	0xC5	Rising edge
4	TCALB_H	10001001	0x89	Rising edge

The proposed scheme allows encoding up to 34 different signals and has a fixed delay of 5.5 RF clock cycles. Additional timing signals can be generated by issuing a VME write command (Fast command) to an internal register of the module.

The duration of the “Reset & Test” LVDS signal is different for these two signals. The test pulse duration is ~20 nS and the reset pulse duration is ~100 µS. The front-ends detect the reset pulse and perform cold start reset. The test signal is used to measure propagation delay of the cable loop and for testing front-end’s functionality. It can be applied at the logic level or used to fire an analog test pulser located at the front-end board. The reset signal is used explicitly for resetting front-ends to a default state.

The “Data” signal is an LVDS output of the Texas Instruments SN65LV1023A serializer chip. A matching SN65LV1224B deserializer chip is used in the front-end circuitry and receiving part of the controller. The clock frequency selected for data transmission is RF/4 or 13.28 MHz which lowers the power consumption of both chips to a reasonable level (<60 mW) and sets the bit frequency at 159.31 MHz which allows to extend the length of the cable loop without a risk of degrading the performance.

Table 2 DAQ loop signal specification

No.	Signal	Level	Direction	Termination	Comment
1	RF & Timing	LVDS	Downstream	Controller	Bussed
2	Reset & Test	LVDS	Upstream	Controller	Bussed
3	Data	LVDS	Downstream	Front-End	Chained
4	Sync	LVTTL	Upstream	N/A	GND

Table 3. Serializer port pinouts

No.	Signal Name	Pin	Comment
1	SRFTM+	1	LVDS output
2	SRFTM-	2	LVDS output
3	SRTST+	4	LVDS input, terminated
4	SRTST-	5	LVDS input, terminated
5	DOUT+	7	LVDS output
6	DOUT-	8	LVDS output
7	SYNC	3	LVTTL input, no termination
8	GND	6	Connected to GND via 100 ohms

Note: RFTIM+, RFTIM-, RSTST+ and RSTST- are connected to same name pins on the front-end’s connectors.

The “Sync” signal is a deserializer LVTTL “Lock” output of the PLL. This connection allows for automatic re-synchronization of the deserializer. If the lock is lost, serializer will generate synchronization pattern until deserializer is locked again. The controller and front-ends have status bits of the connection upstream and downstream. Summary of four DAQ loop signals is presented in Table 2 and pinouts of the dual port RJ-45 jack in Table 3 and Table 4.

Table 4 Deserializer port pinouts

No.	Signal Name	Pin	Comment
1	DRFTM+	1	LVDS input, terminated
2	DRFTM-	2	LVDS input, terminated
3	DRTST+	4	LVDS output
4	DRTST-	5	LVDS output
5	DATIN+	7	LVDS input, terminated
6	DATIN-	8	LVDS input, terminated
7	LOCKB	3	LVTTL output active low
8	GND	6	Connected to GND via 100 ohms

The Chain Readout Controller receives timing signals from the MINOS Master Clock System. A modified version of the MINOS Minder Timing Module (MTM) is used to distribute timing signals within MINERvA readout crates. Each CROC has an input CAT5e RJ-45 connector compatible with a set of standard MINOS timing signals. The pinouts of the connector are shown in Table 5.

Table 5 Timing input connector pinouts

No.	Signal Name	Pin	Comment
1	RF-	1	LVDS input, terminated
2	RF+	2	LVDS input, terminated
3	SGATE-	3	LVDS input, terminated
4	SGATE+	4	LVDS input, terminated
5	CNRST-	5	LVDS input, terminated
6	CNRST+	6	LVDS input, terminated
7	TCALB-	7	LVDS input, terminated
8	TCALB+	8	LVDS input, terminated

Note: The pinouts of the connector correspond to CAT3 straight pinouts

A simplified block-diagram of the data transmission logic is shown in Fig. 3. The contents of the message first must be loaded into a 16-bit wide FIFO memory via VME bus. After the message is loaded, it can be transmitted by sending a VME command to the controller. After the front-end has responded, its message can be read out from a Dual Port Memory (DPM). There is a status register for each DAQ loop channel with various status information including FIFO Empty flag, FIFO Full flag, Message Sent, Message Received bits etc.

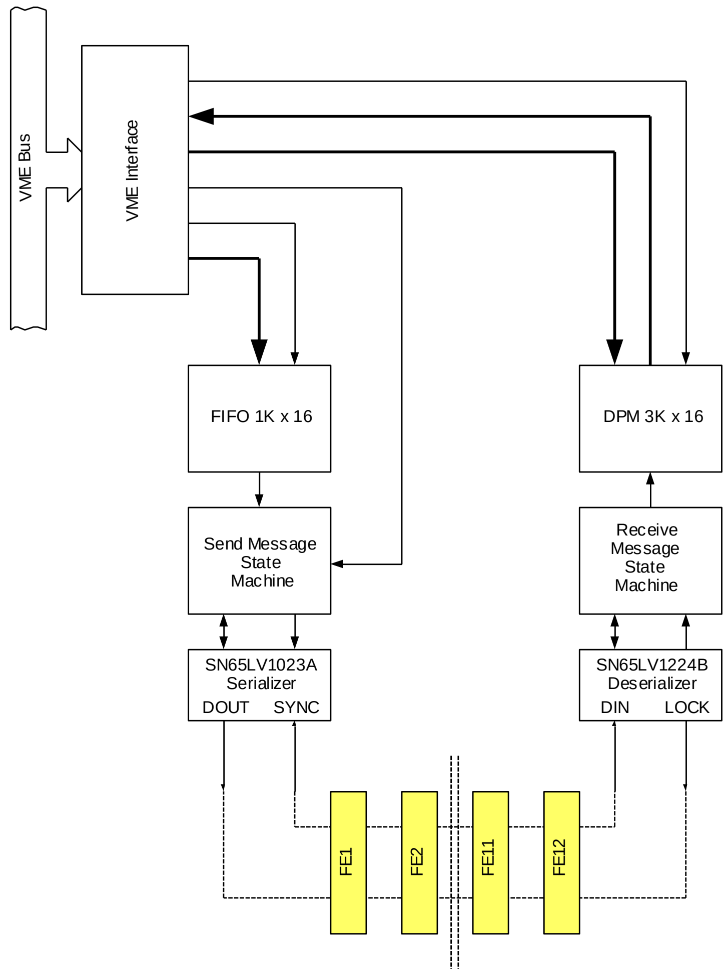


Fig. 3 Data transmission logic

3. Data formats

A data from the controller is transferred to front-ends in a form of messages. Each message consists of several groups of 10-bit words [3]. Two upper bits of the word are control bits used to indicate the beginning and the end of a message and lower eight bits represent actual byte of data. Currently there are four groups in a message. First group consists of one word. This group also indicates the beginning of a message and includes an address of the front-end. Second group consists of eight words. Third group is actual data send to the front-end. The number of words in this group is always *odd*. The readout controller also attaches a CRC byte as a fourth group at the end of the message for data verification. This group also indicates the end of a message. The total number of bytes that controller receives from the VME master as a message data is always *even*. The order of bytes in a 16-bit word follows Big Endian VME byte ordering. In the DAQ loop the most significant byte is transmitted *first*.

Each front-end in the loop has a unique address. The front-end re-transmits the message if the address in the message does not match its pre-assigned address. The addressed front-end responds to a controller's message by transmitting its response along the chain back to the controller. The format of the response message follows the same rules as described in the previous paragraph. There is always *odd* number of words in the third group, so the total number of data bytes received by the controller excluding CRC byte is always *even*. This convention simplifies formatting data to 16-bit VME words. The controller calculates the number of data bytes it receives and writes it at a starting address of the DPM as a 16-bit data word. The length of the data in bytes includes itself. Consequent messages are written starting at the current value of the address pointer with the data length first. The overflow bit is set when the address pointer reaches the last available address. The DPM memory is accessible to the VME master in 16-bit and 32-bit data formats. The received CRC byte is compared to the calculated checksum of the data, but the DPM address pointer *is not incremented*. Several messages can be stored in the DPM for the consequent readout. The size of the DPM for each DAQ loop is 6 KB. The size of the FIFO memory for each channel is 2 KB.

4. VME module implementation

The current version of the readout controller is implemented as a standard 6Ux160 mm **A24D32** VME slave module. It has the following front panel connectors (see Fig. 4):

- A single RJ-45 input timing connector
- Four dual RJ-45 DAQ loop connectors. Note that the top connector of the pair has to be connected to the first front-end in a chain

There are five bi-level LEDs on the front panel:

- "VME" (green) – VME command decoded
- "NO RF" (red) – missing external RF input signal
- "RFD" (green) – RF signal detected in the loop
- "ERR" (red) – CE, TO or DF error bit is set

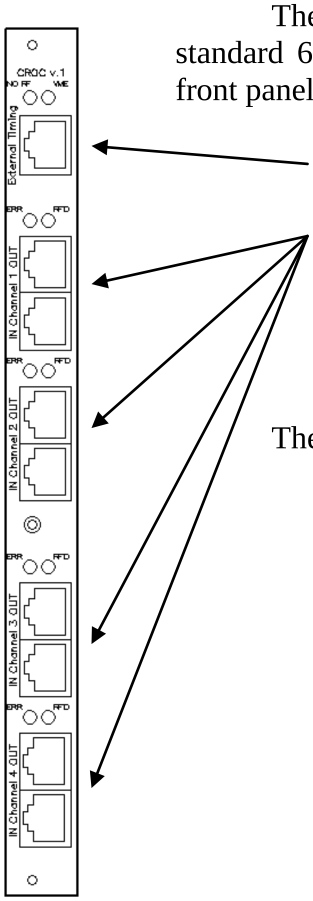


Fig. 4 CROC front panel

5. Specification

- Power consumption - +5V/1.2A
- External timing inputs - LVDS (see Table 5)
- DAQ loop signals - LVDS/LVTTL (see Table 2)
- Main clock frequency - 53.1 MHz (± 200 ppm)
- VME address modifiers - 39, 3d, 3e, 3a, 3b, 3f
- VME address/data mode - A24D16
- DPM read address/data mode - A24D16, A24D32, A24BLT
- Number of front-ends per loop - 12 max
- Channel FIFO size - 1Kx16 bit
- Channel DPM size - 3Kx16 bit
- DAQ loop cable type - unshielded CAT5e
- Maximum loop length - 60 ft.
- DAQ loop data rate - 160 Mbit/s (12 bits @13.3 MHz)

6. VME registers

Each controller module has four DAQ loops or channels. Each channel has its own unique VME addresses for memories and data registers. An eight bit on-board switch selects 64K address space for each controller. A general VME memory map is shown in Table 7. The description of the readout controller channel data registers listed below is the same for all channels (n = 1 - 4).

Readout Controller channel input data register MData*n*

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYTE(0)								BYTE(1)							

Note: BYTE(0) – first byte of the message, BYTE(1) – second byte of the message.

In order to send a message to the front-end, a VME master has to write all data bytes to a CROC's FIFO memory using a single 16-bit data register. After all data is written, the Send Message command has to be issued by writing 0x0101 to the Send Message register. The controller will wait for the front-end response and indicate that it has received it by changing the Message Received status bit. If the CRC sum was incorrect, the CRC error bit will be set as well. If the controller does not receive a response from a front-end within ~480 µS, the TimeOut error bit is set. If the address pointer reaches *17FF*, the DPM Full error bit is set.

Readout Controller channel Send Message register SM*n*

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

The output data format is similar to the input data format. First word of the data is data length in bytes (inclusive). The most significant byte of the message is received *first*.

Readout Controller channel received data Dual Port Memory RData*n*

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL(0)								DL(1)							
BYTE(0)								BYTE(1)							
…								…							
BYTE(2k-1)								BYTE(2k)							

Note: DL1, DL0 – data length, BYTE(0) – first byte of the message, BYTE(1) – second byte of the message, BYTE(2k) – last (even) byte of the message.

The received data can be read out from the controller Dual Port Memory using specified address range for each channel.

Readout Controller channel Status Register SR*n*

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	PL1	PL0	0	LS	SY	RF	0	DF	FF	EF	TO	CE	MR	MS

Note: MS – Message sent,
MR – Message received,
CE – CRC error,
TO – Timeout error,
EF – FIFO not empty flag (0 - empty),
FF – FIFO full flag (1 – full),
DF – DPM full flag (1 – full),
RF – RF loop monitor (1 – RF present),
SY – Serializer SYNC status (1 - synchronized),
LS – Deserializer LOCK status (1 - locked),
PL0 – CPLD PLL0 status (1 - locked)
PL1 – CPLD PLL1 status (1 - locked).

A separate VME command is used to clear some status register bits. Only seven bits: MS, MR, CE, TO, FF, EF and DF are cleared by this command. The DPM write address pointer has to be reset after each front-end message is read out, otherwise the next message will be stored at the next available address at the end of the previous message. Both commands are using the same VME address, but different data bits as shown below.

Readout Controller channel Clear Status/Reset DPM register CS*n*

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	RP	0	CS	0	0	0	0	0	RP	0	CS	0

Note: CS = 1 for Clear Status command, otherwise 0
RP = 1 for Reset DPM pointer command, otherwise 0

In order to facilitate cable loop delay measurement, which is necessary for a correction of the SGATE propagation delay to the front-end the test pulse is used. The loop delay counter is running at 8RF frequency, which provides a timing resolution of ~2.35 nS. Each time the test pulse is generated by the CROC logic, the counter is loaded with the time difference between its departure and arrival. The cable delay value is stored until Clear Status command is issued. It is possible to perform multiple measurements by issuing test pulse several times and dividing the result by the number of pulses. The result of this measurement combined with the measured by the front-end arrival time of the test pulse can be used to calculate individual propagation delay of the timing signals to each front-end in the DAQ loop. Assuming that the front-end is capable of measuring a time difference between rising edge of the SGATE signal and arrival time of the test pulse, the following equation is true:

$$\mathbf{T_f = (T0 + T_d + T_l - M0)/2}, \text{ where}$$

T_f – propagation delay of SGATE to the front-end,
T0 – delay setting of the test pulse relative to SGATE,
T_d – internal logic delay (will be measured with the front-end),
T_l – cable loop delay,
M0 – test pulse arrival time relative to SGATE.

Readout Controller channel Loop Delay register LD*n*

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0

Note: D6..D0 – result of the loop delay measurement in 2.35 nS steps

It is possible to accumulate several messages in the DPM, assuming they will require less than 6K bytes of storage. In order to read the whole array of messages in a single step one need to know what is the size of the array. There is a separate VME command to read current value of the DPM write pointer. The actual size of the array is equal to the value of the DPM pointer minus two bytes.

Readout Controller channel DPM pointer register MP*n*

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0	P15	P14	P13	P12	P11	P10	P9	P8

Note: P0..P15 – current value of the DPM write pointer (bytes)

The following control registers are common for all four channels. The Timing Setup register allows selecting internal or external clock source and time delay of the test pulse. In the internal mode main clock signal is generated using 53 MHz quartz crystal. In the external mode the MINOS MTM module can be used as an external clock source. The test pulse is generated every time when positive transition of the SGATE signal is detected and the Test Pulse delay is enabled. A relative delay of the pulse can be selected using bits D0-D9 of the Timing Setup register. Writing to the Test Pulse register also generates a test pulse. Writing to the Channel Reset register generates a channel reset pulse. Both test pulse and channel reset pulse will be sent only to the channels with the corresponding mask bit set to one. Writing to the Fast Command register an 8-bit value generates an encoded timing command. The following additional timing commands are currently used in the front-end:

Table 6 Additional encoded timing commands

No.	Signal Name	Binary Value	F7..F0 hex value
5	FPGA_RST	10001101	0x8D
6	LOAD_TIMER	11001001	0xC9

The MTM signals listed in the Table 1 can be simulated using this command as well. One should keep in mind that in this case they will not be synchronous to the MINOS timing. Generating SGATE using software timing commands will *not generate* the delayed test pulse even if the Test Pulse delay is enabled. The Test Pulse delay enable bit can be used to disable the test pulse while running with the external SGATE signal.

Readout Controller Timing Setup register TS0

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CM	0	0	TE	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Note: TE – Test Pulse delay enable (1 – enabled)
D0-D9 – Test Pulse delay value relative to SGATE in 18.9 ns steps
CM – Clock mode (1 – external, 0 – internal)

Readout Controller Reset and Test pulse mask register RT0

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	R4	R3	R2	R1	0	0	0	0	T4	T3	T2	T1

Note: T1-T4 – Test pulse enable for channels 1-4
R1-R4 – Reset enable for channels 1-4

Readout Controller Channel Reset register CR0

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Readout Controller Fast Command register FC0

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NC	NC	NC	NC	NC	NC	NC	NC	F7	F6	F5	F4	F3	F2	F1	F0

Readout Controller Test Pulse register TP0

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

Table 7 Readout controller VME map

Starting Address	Size	Read/Wri	Comment
Base + 000000	6K	R	Channel 1 DPM memory, RData1
Base + 002000	2	W	Channel 1 FIFO input data register, MData1
Base + 002010	2	W	Channel 1 Send Message register, SM1
Base + 002020	2	R	Channel 1 Status Register, SR1
Base + 002030	2	W	Channel 1 Clear Status/Reset DPM register, CS1
Base + 002040	2	R	Channel 1 Loop Delay register, LD1
Base + 002050	2	R	Channel 1 DPM Pointer register, MP1

Base + 004000	6K	R	Channel 2 DPM memory, RData2
Base + 006000	2	W	Channel 2 FIFO input data register, MData2
Base + 006010	2	W	Channel 2 Send Message register, SM2
Base + 006020	2	R	Channel 2 Status Register, SR2
Base + 006030	2	W	Channel 2 Clear Status/Reset DPM register, CS2
Base + 006040	2	R	Channel 2 Loop Delay register, LD2
Base + 006050	2	R	Channel 2 DPM Pointer register, MP2
Base + 008000	6K	R	Channel 3 DPM memory, RData3
Base + 00A000	2	W	Channel 3 FIFO input data register, MData3
Base + 00A010	2	W	Channel 3 Send Message register, SM3
Base + 00A020	2	R	Channel 3 Status Register, SR3
Base + 00A030	2	W	Channel 3 Clear Status/Reset DPM register, CS3
Base + 00A040	2	R	Channel 3 Loop Delay register, LD3
Base + 00A050	2	R	Channel 3 DPM Pointer register, MP3
Base + 00C000	6K	R	Channel 4 DPM memory, RData4
Base + 00E000	2	W	Channel 4 FIFO input data register, MData4
Base + 00E010	2	W	Channel 4 Send Message register, SM4
Base + 00E020	2	R	Channel 4 Status Register, SR4
Base + 00E030	2	W	Channel 4 Clear Status/Reset DPM register, CS4
Base + 00E040	2	R	Channel 4 Loop Delay register, LD4
Base + 00E050	2	R	Channel 4 DPM Pointer register, MP4
Base + 00F000	2	R/W	Timing Setup register, TS0
Base + 00F010	2	R/W	Reset and Test pulse mask register, RT0
Base + 00F020	2	W	Channel Reset register, CR0
Base + 00F030	2	W	Fast Command register, FC0
Base + 00F040	2	W	Test Pulse register, TP0

II. MINERvA CROC Interface Module

1. Introduction

MINERvA Chain Readout Interface Module (CRIM) is designed to provide various interface functions for the CROC (Chain Readout Controller [4]) VME module used in the DAQ system and various test stand setups. The need for this module is dictated by a wide variety of applications the CROC module must be used for. Implementation of a full set of requirements in the CROC design would unavoidably complicate it and increase its cost as well as potentially decrease the ease of its use and the reliability. On the other hand, additional features needed for the test stand configurations are mostly useless in a standard DAQ configuration and, therefore, present additional overhead for the CROC design.

2. Main features

The CRIM has the following front panel features (Fig. 8):

- External timing input RJ-45 connector compatible with MINOS MTM [2] output
- Two dual port timing output RJ-45 connectors compatible with the CROC external timing input
- Four dual LEMO connectors for four LVTTL input and four LVTTL output signals (1K or 50 ohm input termination, 1K output load)
- One dual port RJ-45 test connector compatible with the CROC DAQ loop signals
- LED indicators for External RF, VME cycle, LVTTL inputs, LVTTL outputs and DAQ loop signals

The following timing modes of operation are implemented:

- MTM timing mode – timing signals are received from the MTM module and some are distributed via four RJ-45 ports to CROC modules
- External timing mode - external timing signals SGATE, CNRST and TCALB, applied to the LVTTL inputs, are synchronized to the internal 53.1047 MHz quartz oscillator and distributed via four RJ-45 ports
- Internal timing mode – all output timing signals are generated by an internal sequencer in sync with a free running on-board 53.1047 MHz quartz oscillator
- External DAQ loop timing mode – the on-board quartz oscillator is locked to the DAQ loop RF clock with encoded timing signals

The following functionality is implemented in the on-board CPLD firmware:

- VME A24D16 slave interface
- VME D08(O) programmable interrupter
- PLL circuit to control a 53.1 MHz VCOX

- Timing sequencer triggerable by MTM or LVTTL input signals

Using functionality described above, the following configurations are possible:

- LED test stand with external LED pulser synchronized to the event data readout by the CROC either using CRIM LVTTL trigger output signal to fire the pulser or using pulser logic level output connected to the CRIM LVTTL input to start readout
- Small data acquisition system running in external or internal timing mode with up to four CROC modules
- Full data acquisition system with MINERvA MTM module and Chain Readout Interface Module to generate VME interrupts using MTM signals
- A self-triggered DAQ system using CRIM trigger capability to generate VME interrupts using a trigger word received from a Front-End
- CROC test setup simulating front-end response using DAQ loop test connector

In all described above configurations synchronization of the data readout to detector events can be done either by using VME interrupts or by polling a VME register.

3. CRIM Functional blocks

The CRIM has several independent functional blocks along with a common VME A24D16 slave interface:

- D08(O) interrupter module
- Timing module
- DAQ loop test module

The following sections describe each module separately.

Interrupter module

The interrupter module has eight inputs assigned in the following order:

1. Input 0 - External trigger from input connector “T”
2. Input 1 - Rising edge of SGATE signal
3. Input 2 - Falling edge of SGATE signal
4. Input 3 - CNRST signal from any source
5. Input 4 - TCALB signal from any source
6. Input 5 - Front-End trigger word received
7. Input 6 - Reserved for future expansion

8. Input 7 - Reserved for future expansion

Input 0 has the highest priority and Input 7 has the lowest. If there are two or more interrupts pending, they are processed in the order of priority. In order to work with VME interrupts, the VME master has to have an interrupt handler.

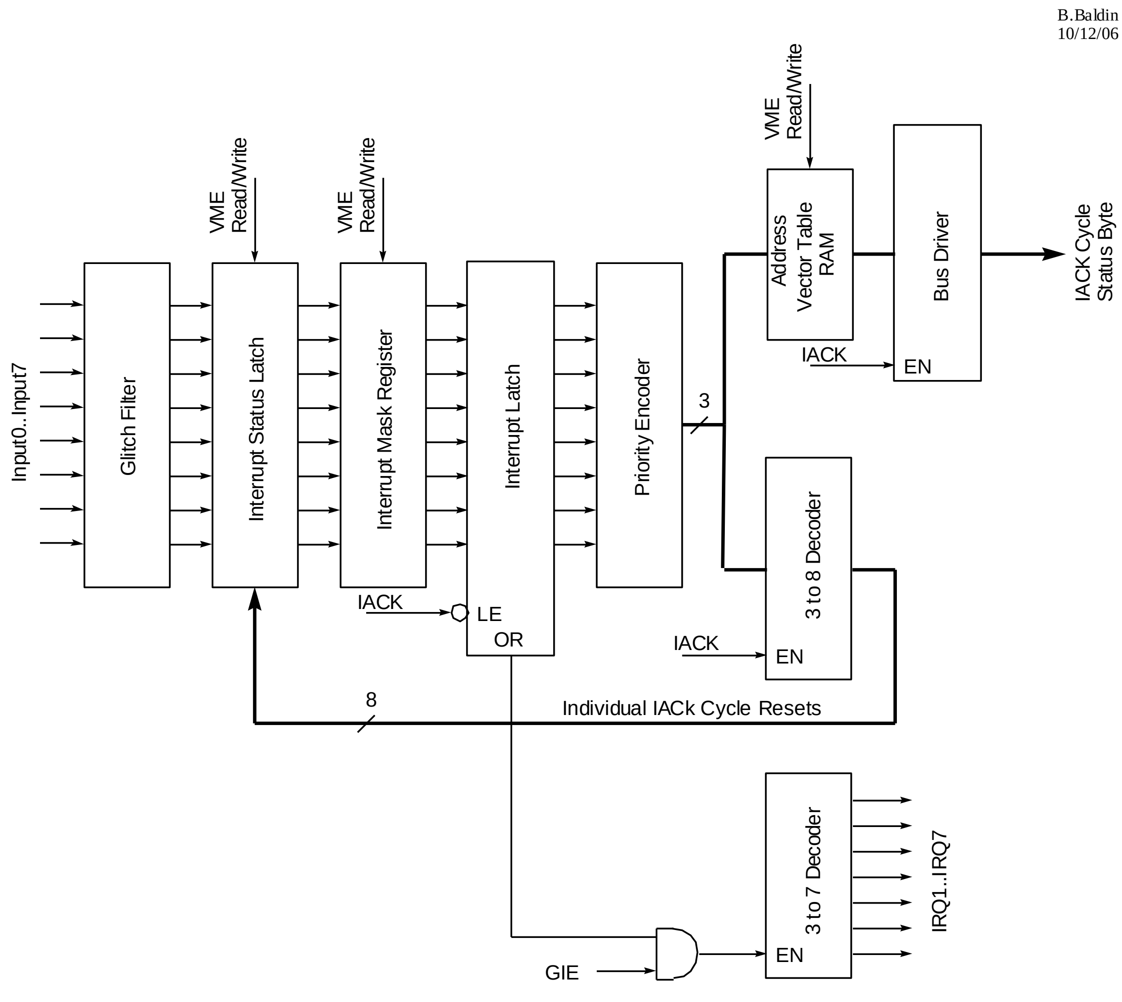


Fig. 5 Block Diagram of the CRIM Interrupter Logic

A block diagram of the CRIM Interrupter Logic is shown in Fig. 5. On power up, interrupts are disabled, interrupt level is set to five and interrupt vector table is loaded with default values (see Table 8 below). When GIE bit is set to one, interrupts are enabled. If any of the interrupt mask register bits is set to one, the corresponding interrupt input is enabled. When input signal sets an interrupt status bit to one, the IRQ line corresponding to the selected interrupt level is driven to zero. If corresponding software driver is active, an interrupt acknowledge cycle (IACK cycle) is triggered. During IACK cycle the interrupter logic compares three bit vector on address lines set by the interrupt handler with its programmed interrupt level. If there is a match, the interrupter puts eight bit status byte (status ID) on the data bus and clears corresponding interrupt status latch.

After that the GIE bit is set to zero and has to be re-enabled by the software. All interrupter features are programmable via VME bus using the following registers:

Interrupt mask register IM, 0xF000

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	M7	M6	M5	M4	M3	M2	M1	M0

Note: M0...M7 are individual mask bits corresponding Input 0...7 signals (1 - enable)

Interrupt status register IS, 0xF010

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	I7	I6	I5	I4	I3	I2	I1	I0

Note: I0...I7 are pending interrupts corresponding Input 0...7 signals (1 - active),
writing a one to any of I0...I7 bits will clear corresponding pending interrupt

Clear pending interrupts register CP, 0xF020

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

Note: Writing **0x81** to this register will clear all pending interrupts

Interrupt configuration register IC, 0xF040

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	GIE	0	0	0	0	IL2	IL1	IL0

Note: IL2 - IL0 is 3-bit VME interrupt level (1...7 - valid, default - 5),
GIE - global interrupt enable (1 - enable, 0 - disable, default - 0)

Table 8 Status ID vector table VT, Base address 0xF800

Input #	Memory Address	Default Status ID Vector
0	Base + 0x0010	0x08
1	Base + 0x0012	0x09

2	Base + 0x0014	0x0A
3	Base + 0x0016	0x0B
4	Base + 0x0018	0x0C
5	Base + 0x001A	0x0D
6	Base + 0x001C	0x0E
7	Base + 0x001E	0x0F

Note: Value of the status ID vector can be anything from 1 to 255, 0 - invalid

- Timing module

The timing module provides four sets of CROC compatible LVDS output signals and has four modes of operation:

- MTM – output clock is synchronous to the MTM clock, other output signals are generated by internal sequencer triggered by TCALB signal from the MTM or by a software command. The sequencer will also be triggered by an external signal at the trigger input (T). SGATE and CNRST can be used as interrupts for the DAQ system. A signal at the trigger input T is also propagated to the trigger output.
- INT - output signals are generated by internal sequencer synchronously with 53.1047 MHz. SGATE, CNRST and TCALB signals are generated periodically at the selected repetition frequency (0.5 Hz. to 52 kHz) or by a software command in the single sequence mode. SGATE and TCALB have programmable settings.
- EXT - SGATE, CNRST and TCALB signals can be provided externally using LVTTL inputs or generated using VME commands.
- DAQ - special mode which simulates MINERvA front-end and detects front-end trigger. In this mode the CRIM does not generate any output timing signals.

A block diagram of the timing module is shown in Fig. 6. In the MTM mode the PLL circuit is locked to the incoming RF signal from the MINOS timing system. In this mode timing sequencer triggered by the MTM TCALB signal generates internal CNRST, SGATE and TCALB signals. The CNRST and SGATE signals are propagated to the CRIM RJ-45 outputs. The TCALB signal is not propagated, but available as LVTTL signal at the Lemo output marked "C controlled by the bit 15 of the SGATE width register GW (**0xC020**). When bit 15 is set to one, the TCALB is generated during the sequence. By default this bit is set to zero.

In the DAQ mode encoded timing signal from the DAQ loop is supplied to the PLL reference input. In EXT and INT modes internal clock frequency is provided by a free running quartz oscillator (53.1047 MHz). The external input trigger signal TRIG (labeled T on the front panel) is permanently connected to the interrupter Input 0 and can be used in any timing mode. Inputs 1...4 of the interrupter are connected to the signal multiplexer output, and, therefore, they depend on the timing mode selected. Input 5 is connected to the DAQ loop test module and can be used in a self-triggering DAQ system. The LVTTL output labeled T provides a trigger signal generated by the VME commands (fast trigger or start sequencer) or a copy of the T input signal in the MTM mode. The LVTTL outputs labeled G, R and C correspond to SGATE, CNRST and TCALB signals of the signal multiplexer respectively. The LVTTL inputs labeled T, G, R and C

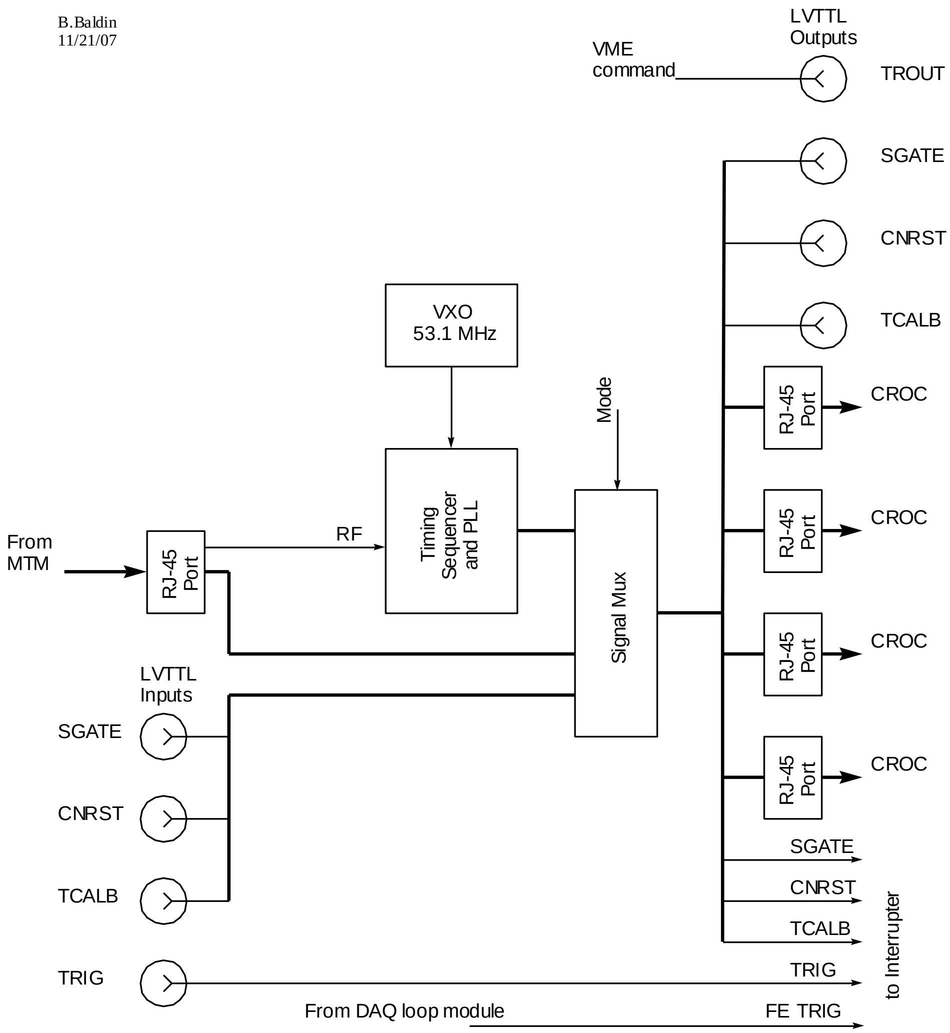


Fig. 6 Timing module

can be used to supply external trigger, SGATE, CNRST and TCAL signals respectively. Note that the input labeled T has a termination resistor of 1K and the other three inputs are terminated with a 50 ohm resistor.

Note that in the EXT timing mode input signals are synchronized to the internal 53.1047 MHz clock. This will cause variation of the delay between LVTTL input and corresponding LVTTL output signal. The duration of the output signal will be multiple of 18.9 ns. The following registers provide settings for the timing module:

Timing setup register TS, 0xC010

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M3	M2	M1	M0	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0

Note: F11...F0 frequency select bits (F11- F8 ~0.5 Hz to 4 Hz range, F7 - F0 ~400 Hz to 52 kHz range, all zero - single sequence mode); M3...M0 - mode select bits (**0x8** - MTM, **0x4** - INT, **0x2**- Ext, **0x1** - DAQ)

SGATE width register GW, 0xC020

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC	X	X	X	X	X	X	X	X	G6	G5	G4	G3	G20	G1	G0

Note: G0...G6 - gate width select bits in 150.6 ns steps in INT mode, X - Don't care
TC – sequencer control bit (1 – enable TCALB, 0 – disable TCALB)

TCALB pulse delay register TP, 0xC030

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	X	X	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Note: D9...D0 - TCALB pulse delay select bits in 18.9 ns steps, , X - Don't care;
The delay counter starts and stops with the gate signal in INT mode

Software trigger register ST, 0xC040

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

Note: Writing **0x0404** to this register generates 18.9 ns trigger output pulse

Software TCALB register SP, 0xC050

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

Note: Writing **0x0404** to this register generates 18.9 ns TCALB pulse in EXT mode

Software SGATE register GS0, 0xC060

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	F	R

Note: Writing **0x0401** to this register starts SGATE pulse; writing **0x0402** stops SGATE pulse in EXT mode

Software CNRST register CR, 0xC080

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	SS	0	CR	0	0	0	0	0	SS	0	CR	0

Note: Writing **0x0202** to this register generates counter reset pulse in EXT mode;
Writing **0x0808** to this register starts single sequence of CNRST, SGATE and TCALB in INT or MTM mode when no frequency is selected (F11…F0 are set to zero)

General purpose software register GR, 0xC0A0

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Note: This is a general purpose read/write register. Default value – **0x0000**

The value of a 28-bit timing counter is latched by the rising edge of the MINOS SGATE signal in two 16-bit registers. This counter is reset approximately every second by the MINOS CNRST signal and provides a timing reference to the MINOS detector timing. The counter is running at the MINERvA MTM clock frequency and its value is valid only in the MTM mode.

Lower 16 bits of the MINOS gate arrival time GT0, 0xC0B0

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

Upper 12 bits of the MINOS gate arrival time GT1, 0xC0C0

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	T27	T26	T25	T24	T23	T22	T21	T20	T19	T18	T17	T16

Note: T27…T0 bits are latched by the rising edge of the external gate signal

- DAQ loop test module

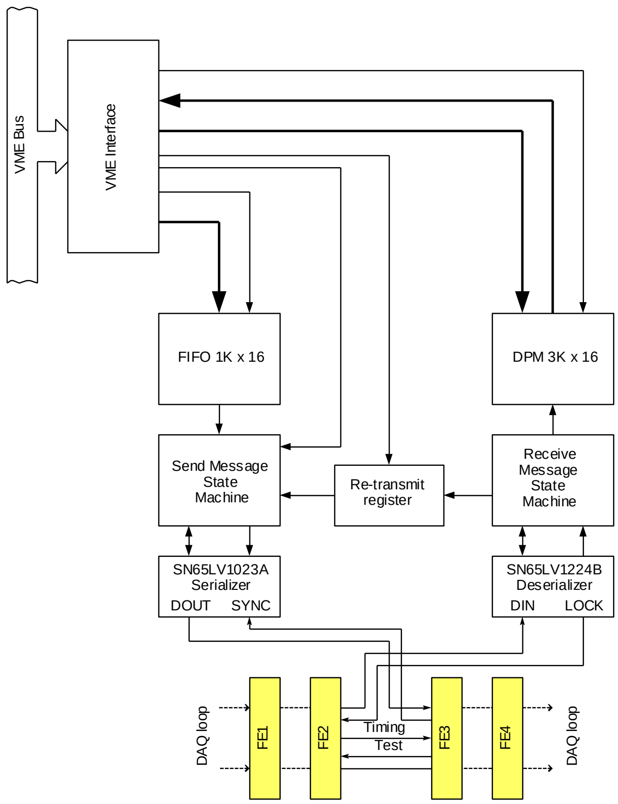


Fig. 7 DAQ loop test module

In the DAQ mode a connection to the external DAQ loop driven by the MINERvA CROC module is required. When inserted in a DAQ loop, this module can be used as a front-end simulator or DAQ loop monitor passing through all messages and storing them in the internal memory. This module also can generate an interrupt when a trigger word is detected in the data stream. A block diagram of the DAQ loop test module is shown in Fig. 7.

The DAQ loop test module has two main modes of operation: pass trough mode and front-end mode. The desired mode is selected by setting to one bit 15 or bit 14 of the DAQ mode control register. In the pass through mode (TR = 1) any message received by the deserializer is stored in the DPM and re-transmitted downstream the DAQ loop via serializer. Usual diagnostics of CRC error and DPM overflow is provided. If the front-end trigger is enabled in the pass through mode (FE = 1), the module will generate a signal for the interrupter when a trigger word is received. Each trigger word will be stored in the DPM instead of regular messages. In the front-end mode (SM = 1), any message received by the deserializer is also stored in the DPM. But, after the CRC byte is received, the module generates a response message using the data stored in the FIFO memory. The FIFO is setup in such a way that after the message is sent, the FIFO read pointer is reset. This allows for repeated responses without loading FIFO after each transmission. The DAQ loop test module has the following registers and memory:

Message data Dual Port Memory RD, 0x0000

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL(0)								DL(1)							
BYTE(0)								BYTE(1)							
…								…							
BYTE(2k-1)								BYTE(2k)							

Note: DL1, DL0 – data length, BYTE(0) – first byte of the message, BYTE(1) – second byte of the message, BYTE(2k) – last (even) byte of the message.

All messages passing through the DAQ loop test module are stored sequentially in the dual port RAM. The FIFO memory is loaded using a single register.

Trigger data Dual Port Memory RD, 0x0000

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIG(0)								TRIG(1)							
TRIG(2)								TRIG(3)							
…								…							
TRIG(2k-1)								TRIG(2k)							

All trigger bytes are stored sequentially in the DPM. An interrupt handling routine has to read trigger data after each interrupt and reset DPM address pointer otherwise the DPM overflow flag will eventually be set.

FIFO data register FD, 0x2000

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYTE(0)								BYTE(1)							

Note: BYTE(0) – first byte of the message, BYTE(1) – second byte of the message.

FIFO reset register FR, 0x2008

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Note: Writing **0x0808** to this register resets the FIFO flag

Send message register SM, 0x2010

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Note: Writing **0x0101** to this register sends a message if FIFO is not empty

Status Register SR, 0x2020

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EC	RS	DS	PL0	CM	LS	SY	RF	0	DF	FF	EF	0	CE	MR	MS

Note: MS – Message sent,
MR – Message received,
CE – CRC error,
EF – FIFO not empty flag (0 - empty),
FF – FIFO full flag (1 – full),
DF – DPM full flag (1 – full),
RF – RF loop monitor (1 – RF present),
SY – Serializer SYNC status (1 - synchronized),
LS – Deserializer LOCK status (1 - locked),
CM – Common mode level violation
PL0 – CPLD PLL0 status (1 - locked),
DS – Differential control signal received (1 - received),
RS – Reset signal received (1 - received),
EC – Encoded timing command received (1 - received).

Clear status/Reset DPM register CS, 0x2030

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	RP	0	CS	0	0	0	0	0	RP	0	CS	0

Note: CS = 1 for Clear Status command, otherwise 0
RP = 1 for Reset DPM pointer command, otherwise 0

Send SYNC to serializer register SS, 0x2040

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Note: Writing **0x0101** to this register sends SYNC pulse to the serializer

Read DPM pointer register MP, 0x2050

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0	P15	P14	P13	P12	P11	P10	P9	P8

Note: P0...P15 – current value of the DPM write pointer (bytes)

Read decoded timing command register DT, 0x2060

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	C7	C6	C5	C4	C3	C2	C1	C0

Note: C7...C0 – bits of the last timing command decoded, start bit not included

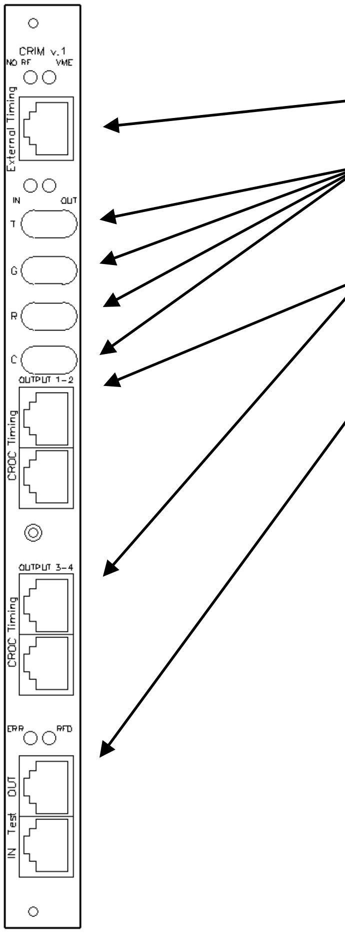
DAQ mode control register CR, 0x2070

VME Data bits															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR	SM	CE	FE	0	0	0	0	0	0	0	0	0	0	0	0

Note: TR – re-transmit enable (1 - enable)
SM - send message enable (1 - enable)
CE - CRC error enable (1 - enable)
FE - front-end trigger enable (1 - enable)

4. VME module implementation

The current version of the chain readout interface module is implemented as a standard 6Ux160 mm **A24D16** VME slave module. It has the following front panel connectors (see Fig. 8):



- A single RJ-45 input connector labeled External Timing
- Four double LEMO LVTTL input/output connectors labeled T - Trigger, G - SGATE, R - CNRST and C - TCALB
- Two dual RJ-45 connectors labeled CROC Timing outputs 1 - 4
- One dual RJ-45 DAQ loop connector labeled Test IN/OUT

There are three bi-level LEDs on the front panel:

- "VME" (green) – VME command decoded
- "NO RF" (red) – missing external RF input signal
- "IN" (green) – LVTTL input signal received
- "OUT" (green) – LVTTL output signal generated
- "RFD" (green) – RF signal detected in the loop
- "ERR" (red) – CE or DF error bit is set

Fig. 8 CRIM front panel

5. Specification

- Power consumption - +5V/0.6A
- Main clock frequency - 53.1047 MHz (± 200 ppm)
- External timing inputs - LVDS (see [4])
- CROC timing outputs - LVDS (see [4])
- LVTTL inputs impedance - 1K, 50 ohm
- LVTTL outputs load - 1K
- DAQ loop signals - LVDS/LVTTL
- VME interrupter type - D08(O)
- VME address modifiers - 39, 3d, 3e, 3a
- VME address/data mode - A24D16
- Test channel FIFO size - 1Kx16 bit
- Test channel DPM size - 3Kx16 bit
- DAQ loop cable type - CAT5e
- Maximum loop length - 60 ft.
- DAQ loop data rate - 13.3 MByte/s

An eight bit on-board switch selects 64K address space for the module. A general VME memory map is shown in Table 9.

Table 9. Chain Readout Interface Module VME map

Starting Address	Size	Read/	Comment
Base + 000000	6K	R	DPM memory, RD
Base + 002000	2	W	FIFO input data register, FD
Base + 002008	2	W	FIFO reset register, FR
Base + 002010	2	W	Send Message register, SM
Base + 002020	2	R	Status Register, SR
Base + 002030	2	W	Clear Status/Reset DPM register, CS
Base + 002040	2	W	Send sync register, SS
Base + 002050	2	R	Read DPM Pointer register, MP
Base + 002060	2	R	Read decoded timing command, DT
Base + 002070	2	R/W	Control register, CR
Base + 00C010	2	R/W	Timing setup register, TS
Base + 00C020	2	R/W	SGATE width register, GW
Base + 00C030	2	R/W	TCALB delay register, TP
Base + 00C040	2	W	Software trigger register, ST
Base + 00C050	2	W	Software TCALB register, SP
Base + 00C060	2	W	Software SGATE start/stop register, GS0
Base + 00C080	2	W	Software CNRST register, CR
Base + 00C0A0	2	R/W	General purpose software register, GR
Base + 00C0B0	2	R/W	Lower 16 bit of MINOS gate timing register, GT0
Base + 00C0C0	2	R/W	Upper 12 bit of MINOS gate timing register, GT1
Base + 00F000	2	R/W	Interrupt mask register, IM
Base + 00F010	2	R/W	Interrupt status register, IS
Base + 00F020	2	W	Clear pending interrupts register, CP
Base + 00F040	2	R/W	Interrupt configuration register, IC
Base + 00F800	16	R/W	Status ID vector table memory, VT

III. References

[1] T.Fitzpatrick, C.Rotolo. “MINOS Master Clock System Preliminary Design Specification,” NUMI-NOTR-ELEC-827, Fermilab, November 2000.

[2] B.Baldin, et al. "D0 Muon Readout Electronics Design," IEEE Trans. on Nuclear Science, Vol.44, No.3, pp 363-369, June 1997.

[3] C.Gingu, P.Rubinov. “Minerva Master/Slave Data Acquisition Board Description and Measurement Result,” MINERvA Document Database #1569-v1, http://minerva-docdb.fnal.gov,Fermilab, June 2005.

[4] B.Baldin. "MINERvA Chain Readout Controller," MINERvA Document Database #773-v2, http://minerva-docdb.fnal.gov, Fermilab, May 2006.