



# **D0\_Muon Readout Controller (MRC) Design Specification**

**M.Larwill, W. Williams, B. Baldin, S. Hansen, S. Los, M.Matveev, V. Vaniev**

## **INTRODUCTION**

### ***Purpose***

The MUON Readout Card (MRC) is a part of D0 MUON Readout System and resides in the Moveable Counting House (MCH). It will be located in existing 9U X 280mm VME crates with custom back-planes. Having only sequential logic and no processor the MRC will perform the following functions:

- Receives data from front end electronics (FE) and buffers it in internal RAM;
- Sends Control signals to digital signal processor (DSP) in FE via universal asynchronous receiver/transmitter (UART);
- Transfers timing and control signals from MUON FANOUT CARD (MFC) to FE and status information from FE to MFC.

### ***Functional Block Description***

Each MRC connects to two FE data sources and consists of the following functional units:

1. Hot-Link Receiver CY7B933 and its Control Logic;
2. 8K byte Dual Port RAM;
3. Receivers/Transmitters from/to MFC;
4. Receivers/Transmitters from/to FE;
5. 2-channel High-Speed Serial Communication Controller AM85C30;
6. 32 bit Control/Status Register;
7. VME Slave Interface.

# DESCRIPTIONS OF SUBSECTIONS

## HOT-LINK RECEIVER CY7B933 and CONTROL LOGIC

HOT-Link Receiver CY7B933 in MRC and HOT-Link Transmitter CY7B923 in FE board are point-to-point serial link for transferring data at 160 M Bit/sec in one direction from FE to MRC.

Eight bits of data are loaded into the HOT-Link Transmitter on the FE then the serial data is shifted out of the differential Pseudo ECL (PECL) serial port at the bit rate and transmitted over the coaxial ASTRO cable. The distance between FE boards and MRC is about 280 ft.. At the MRC the signal is transformer coupled to a COMLINEAR CLCO14 receiver chip which then drives the HOT-Link receiver input.

A reference clock frequency for the clock/data synchronizer is 16 MHz. A LOW on the Data Output Ready (RDY) pin of HOT-Link Receiver indicates that new data has been received and is ready to be delivered. A missing pulse on RDY shows that the received data is the Null character (normally inserted by the transmitter as a pad between data inputs).

The data outputs (Q0-Q7, SC/D and RVS) all transition simultaneously, and are aligned with RDY and CKR within timing specification to interface directly with external memory. In our case this is a dual port RAM.

The HOT-Link Control Logic on the prototype is contained in one ALTERA EPX780\_84 for each FE but is implemented with ALTERA 7128S chips in the final design. The HOT-Link Control Logic contains the internal 8-bit register for Q0-Q7 data signals clocked on RDY pulse, 12-bit RAM Address Counter, 16-bit Time-out Counter and a combinatorial logic for HOT-Link control and writing data from this internal register to corresponding RAM chip.

The DSTREAM Trigger is set on receipt of a K28.0 character for all Data Transfer time and reset on receiving a K23.7 character. The Dual port RAM Address Counter is increased on the edge of the RDY pulse. It is reset on receipt of a K23.7 character, MRESET, RESET pulse from CSR and reset pulse INIT\_FR from INIT signal. Overflow of the RAM Counter (IAD13=1) disables the assertion of WR1-WR4 signals to RAM and further counter increasing. IAD13=1 also sets the OVF\_DSTR trigger bit in the CSR. Four RAM Write signals WR1-WR4 are caused from IAD0-IAD1 and RDY signal.

INIT\_FR pulse (62,5 Ns. width at 16 MHz clock) is generated on INIT signal from TFW only when current Data Transfer had finished (DSTREAM=0) or Data Transfer had not finished (DSTREAM=1) (erroneous situation), but time-out period had expired.

Upon generating INIT\_FR (and also upon generating REFI pulse from CSR by software) the REFRAM trigger is set for framing. The four delay elements (based on triggers) are designed to examine the RDY pin only after the 4 clock cycle delay (see HOT-Link description). The RDY signal coming after this delay sets the REFD trigger (separated bit to CSR indicating the reframe ended) and resets the REFRAM trigger.

In Built-In-Self-Test (BIST) Mode, upon receiving BISIN pulse from CSR the BISTEN trigger is reset and BISTEN signal to HOT-Link Receiver CY7B933 becomes active (LOW). This means that BIST is in progress. The BIST loop itself consists of 511 bytes transfer (see HOT-Link description). The procedure of REFD generating in BIST Mode is the same as discussed above on INIT\_FR. But in this case the BISTEN trigger is set (and BISTEN becomes inactive) after two clock cycles.

If there were no violation symbols during BIST, the BISOK trigger is set on RDY transfer from LOW to HIGH at the end of BIST and active HIGH bit BISOK means that BIST had passed without errors.

The 16-bit Time-out Counter (time-out period is equal approximately 4 ms in our case) is incremented on CKR clock signal from CY7B933 when DSTREAM is active during reframe or BIST. This counter is reset after MRESET, RESET pulse from CSR, INIT\_FR pulse, K23.7 character receiving after successful Data Transfer, setting BISTEN inactive and setting REFD trigger. If Time-out Counter is not reset after normal termination of current operation, the TOUT Trigger is set in 4 ms. The separated condition bits of time-out sources (TOUT\_DSTR, TOUT\_REFR and TOUT\_BIST) are available in CSR.

If the violation symbol was received during DSTREAM or BIST (RVS signal from CY7B933 was active) the corresponding bit to CSR (RVS\_DSTR or RVS\_BIST) is set.

### ***Short CY7B933 Description***

For a detailed description of the HOT-Link devices refer to the CYPRESS data sheets. Here the devices is described only briefly. The HOT-Link Receiver accepts the serial bit stream at its differential line receiver, and, using a completely integrated PLL clock synchronizer, recovers the timing information necessary for data reconstruction. To properly align the incoming bit stream to the intended byte boundaries, the bit counter in Clock Sync block must be initialized. The Framer logic block in CY7B933 checks the incoming bit stream for the unique pattern that defines the byte boundaries. This logic looks for the symbol defined as “Special Character Comma” (K28.5). Once K28.5 is found, the free running bit counter in the Clock Sync block is synchronously reset to its initial state, thus “framing” the data to the correct byte boundaries. The bit stream is de-serialized, decoded, and checked for transmission errors. The recovered byte is presented in parallel form at a byte rate (16 MHz).

Framer Logic in the Receiver is controlled by the Reframe Enable (RF) input signal on the CY7B933. When RF is held HIGH, each SYNC (K28.5) symbol detected in the shifter will frame the data that follows. When RF held LOW, the reframing logic is disabled. The incoming data stream is then continuously deserialized and decoded using byte boundaries set by the internal byte counter. Bit errors in the data stream will not cause alias SYNC characters to reframe the data erroneously.

## **SERIAL COMMUNICATION CONTROLLER (SCC)**

The Dual-channel SCC formerly a Zilog Z16C30 now is an AM85C30 and is intended for direct communications between the VME Processor in Muon Crates and each DSPs in the FE. This connection is necessary for exchanging data between FE and MRC in testing and diagnostic operation modes.

### ***Brief Description of the AM85C30 SCC***

The AM85C30 is an enhanced serial communications controller designed for use with any conventional 8- and 16-bit bus. The device contains two independent 0 to 10 M-bit/sec, full duplex channels, two baud rate generators per channel, a digital phase-locked loop per channel, character counters for both receiver and transmitter in each channel, and 32-byte data FIFO for each receiver and transmitter.

The SCC is used in MRC in 16-bit non-multiplexed mode. In this case all internal registers in each channel are accessed indirectly using the address pointer in the Channel Command/Address Register (CCAR) in each channel. The address of the desired register is first written to the CCAR and then the selected register is accessed; the pointer in the CCAR is automatically cleared after this access. The Transmit Data Register (TDR, write only) and Receive Data Register (RDR, read only) are accessed directly using the separate D//C pin, without disturbing the contents of the pointer in the CCAR.

There are six sources of interrupts in each channel of the SCC: Receive Status, Receive Data, Transmit Status, Transmit Data, I/O Status and Device Status, arranged on the internal interrupt daisy chain in this priority order. The SCC doesn't require Interrupt Acknowledge Cycles (but supports this mode), so /SITACK and /PITACK input interrupt acknowledge pins are simply pulled up. Software can read the Interrupt Pending (IP) and Interrupt Under Service (IUS) bits in the Daisy Chain Control Register (DCCR).

### **DUAL PORT RAM**

The Dual Ported RAM holds data from one event from the each FE. The RAM has a maximum size of 8 Kbytes and is implemented with 2Kx8 Dual-Port Static RAM chips. There are four such chips for every FE interface section. The bytes of data from CY7B933 (HOT-Link) are written sequentially to the port A of memory chips. The port B of these chips is available for writing and reading from VME.

## **COMMUNICATIONS BETWEEN MRC AND FANOUT**

There will be one Fan-out card and 11 to 13 MRCs in each MUON Crate. Each MRC receives 15 signals from and transmits five signals to Fan-out Card (see Table 1).

### **FAST PECL SIGNALS FROM FANOUT**

Two fast signals from Fan-out Card are transmitted to MRCs using differential PECL lines. PECL differential transmitters are used in the Fan-out Card and CLC014 differential line receivers are used in MRCs for these signals.

### **TTL SIGNALS FROM FANOUT**

The other signals from Fan-out to MRCs (INIT, L1ACC, L1REJ, L2ACC, L2REJ and Xing<0-7>) are transmitted in TTL levels. 74FCT16244 16 bit latches are used as receivers and transmitters for these signals.

### **SIGNALS TO FANOUT**

Each MRC can transmit five signals to Fan-out Card: BUSY1, BUSY2, ERROR1, ERROR2 and SRQ. Each output signal from the MRC notifies the Fan-out Card about particular MRC internal condition (interrupts from UART s, BEG\_DATA and END\_DATA HOT-Link Control Logic signals) and can cause an interrupt of Fan-out Interrupt Controller. All output signals driven with 74BCT760 buffers on the MRCs are open collector outputs are WIRE-ORed on the back-plane with the same signals of other MRCs in the Crate.

All the discussed above signals are connected to the A and C lines in J2 VME back-plane.

**TABLE 1 ( INTERCONNECTIONS IN VME CRATE )**

Number	Signal	Source	Dest.	Type	Transmitter	Receiver
1	RF 53 Mhz	FO	MRCs	PECL lines	SY10ELT22	CLC014
2	Encoded Gap/Reset/FC	FO	MRCs	PECL lines	SY10ELT22	CLC014
3	INIT	FO	MRCs	TTL line	FCT16244	FCT16244
4	L1 ACC	FO	MRCs	TTL line	FCT16244	FCT16244
5	Spare	FO	MRCs	TTL line	FCT16244	FCT16244
6	L2 ACC	FO	MRCs	TTL line	FCT16244	FCT16244
7	L2 REJ	FO	MRCs	TTL line	FCT16244	FCT16244
8-15	Crossing #	FO	MRCs	TTL lines	FCT16244	FCT16244
16	ERROR1	MRCs	FO	WIRE OR TTL OC	74BCT760 (OC)	FCT16244
17	ERROR 2	MRCs	FO	WIRE OR TTL OC	74BCT760 (OC)	FCT16244
18	BUSY 1	MRCs	FO	WIRE OR TTL OC	74BCT760 (OC)	FCT16244
19	BUSY 2	MRCs	FO	WIRE OR TTL OC	74BCT760 (OC)	FCT16244
20	SRQ	MRCs	FO	WIRE OR TTL OC	74BCT760 (OC)	FCT16244
21	STR	FO	MRCs	TTL line	FCT16244	FCT16244

## **COMMUNICATION BETWEEN MRC AND FRONT END BOARDS**

There are 21 communication signals between one section of the MRC and each of two Front End Sections (see Table 2).

16 of these signals (53MHz, RESET, GAP, INIT, L1ACC, L1REJ, L2ACC, L2REJ, DONE and UARTTRANSM) are outputs for MRC and inputs for FE; five signals (L2DATA, ERROR1, BUSY1, BUSY2 and UARTREC) are inputs for MRC and outputs for FE boards.

### **FAST SIGNALS TO FE FROM MRC**

Two fast signals (53MHz, Encoded GAP/Sync/RESET) are transmitted via coaxial ASTRO cables. These signals are transmitted by CLC006 drivers and are transformer coupled to convert differential outputs to single ended signal.

Receivers for ERROR1, BUSY1, BUSY2 and UARTREC signals are based on AD8002 Current Feedback Amplifier and MAX902 Voltage Comparator.

### **SIGNALS TO MRC FROM FE**

The L2DATA is transformer coupled on the MRC to a CLC014 differential receiver. The CLC014 drives the HOT-Link Receiver CY7B933. L2DATA arrives at the MRC on ASTRO cables.

All other signals (INIT, L1ACC, L1REJ, L2ACC, L2REJ and Xing<0-7>) are transmitted to FE using 75110A High Speed Current Drivers.

**TABLE 2 ( INTERCONNECTIONS BETWEEN FRONT END and MRC )  
(one section, twisted pairs)**

Number	Signal	Source	Dest.	Type	Transmitter	Receiver
1	INIT	MRC	FE	tw. pair	75110A	AD8002
2	L1 ACC	MRC	FE	tw. pair	75110A	AD8002
3	L2 ACC	MRC	FE	tw. pair	75110A	AD8002
4	L2 REJ	MRC	FE	tw. pair	75110A	AD8002
5-12	Crossing #	MRC	FE	tw. pair	75110A	AD8002
13	DONE	MRC	FE	tw. pair	75110A	AD8002
14	UART transm.	MRC	FE	tw. pair	75110A	AD8002
15	UART receiver	FE	MRC	tw. pair	75110A	AD8002
16	ERROR 1	FE	MRC	tw. pair	75110A	AD8002
17	ERROR 2	FE	MRC	tw. pair	75110A	AD8002
18	BUSY 1	FE	MRC	tw. pair	75110A	AD8002
19	BUSY 2	FE	MRC	tw. pair	75110A	AD8002
20	STRB	MRC	FE	tw. pair	75110A	AD8002

**TABLE 2 (INTERCONNECTIONS BETWEEN FRONT END and MCH )  
(coaxial cable)**

Number	Signal	Source	Dest.	Type	Transmitter	Receiver
1	53 Mhz	MRC	FE	coaxial	CLC006	CLC014
2	RESET(FC)	MRC	FE	coaxial	CLC006	CLC014
3	GAP	MRC	FE	coaxial	CLC006	CLC014
4	L2 DATA	FE	MRC	coaxial	CLC006	CLC014

**TABLE 2 (INTERCONNECTIONS BETWEEN FRONT END and L2 )  
(coaxial cable)**

1	L2 TRIG	FE	L2	coaxial	CLC006	CLC014
---	---------	----	----	---------	--------	--------

The high speed connections to the FE are made on micro-coax. The MRC sends and receives signals from to FE crates. The connector on the MRC has signals for two crates and must be split into two cables to feed two FE crates. The pin-outs for the connectors are shown below.

**AMP 103167-5 16 PIN HEADER CONNECTOR MOUNTED ON MRC BOARD**

Pin	Signal
1	DATAA+
2	DATAA-
3	CLKA+
4	CLKA-
5	GAPA+
6	GAPA-
7	RESETA+
8	RASETA-
9	DATAB+
10	DATAB-
11	CLKB+
12	CLKB-
13	GAPB+
14	GAPB-
15	RESETB+
16	RESETB-

**AMP 103167-1 8 PIN HEADER CONNECTOR MOUNTED ON FE BOARD**

Pin	Signal
1	DATA+
2	DATA-
3	CLK+
4	CLK-
5	GAP+
6	GAP-
7	RESET+
8	RASET-

## **VME INTERFACE**

The MRC has a 32-bit VME Slave Interface consisting of:

- Input buffers for VME address and control signals;
- VME address decoder;
- VME 32-bit Data Bus Receivers/Transmitters;
- VME command decoder.

A memory map of all RAM buffers and I/O ports in MRC is given in Table 4. It is possible to select Base address via DIP switches on MRC board (A16-A31 address bits). A step of selecting Base address is 64 Kbyte.

74FCT16244 chips are used as input buffers and 74FCT16500 as Data Bus Receivers/Transmitters. The VME address and command decoder for both sections of MRC is based on single ALTERA EMP7128s chip.

**TABLE 4 (MRC Memory Map)**

All addresses start from Base Address ( N \* 64K, N=0...255). In the following table the base address is assumed to be 220000.

Address	Starting Address Offset	Length (bytes)	Status Read/ Write	Assigned
220000.. .221FFF	0	8K	R/W	Dual-Port RAM Section A
224000	16K	1	W	Reset SCC AM85C30
214010	16K + 16	4	R/W	CSR Section A
228000.. .229FFF	32K	8K	R/W	Dual-Port RAM Section B
22C000	48K+3	2	R/W	SCC Channel B; access to all internal registers
22C004	48K+7	2	R/W	SCC Channel A; access to all internal registers
22C008	48K+11	2	R/W	SCC Channel B; direct access to TDR and RDR registers
22C00C	48K+15	2	R/W	SCC Channel A; direct access to TDR and RDR registers
22C010	48K + 16	4	R/W	CSR Section B

CSR data format (16 bit LOW)

vme	D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
csr	D07	D06	D05	D04	D03	D02	D01	D00	D15	D14	D13	D12	D11	D10	D09	D08

CSR data format (16 bit HIGH)

vme	D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
csr	D23	D22	D21	D20	D19	D18	D17	D00	D31	D30	D29	D28	D27	D26	D25	D24

Zilog Z16C30 Serial Communication Controller Data format

vme	D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
scc	D07	D06	D05	D04	D03	D02	D01	D00	D15	D14	D13	D12	D11	D10	D09	D08

### **32 BIT CONTROL/STATUS REGISTER**

There is one 32-bit Control/Status Register (CSR) in each MRC section (Table 3). It consists of:

- BUSY1, BUSY2 and ERROR1 bits indicating conditions of corresponding signals from FE, and ERROR2 bit (as a LOGICAL OR of error sources in MRC itself (OVF, RVS and TOUT signals from HOTLink Receivers with corresponding mask bits)
- mask BUSY1\_M, BUSY2\_M, ERROR1\_M, ERROR2\_M and SRQ\_M bits for corresponding signals, and ERROR1\_S bit for setting ERROR1 signal by software
- TOUT, RVS and OVF bits for imitation corresponding conditions of HOTLink Receiver by software
- RESET bit for resetting ERROR bits (TOUT, RVS and OVF) in CSR itself and in Hotlink Control Logic
- REF bit (Reframe Mode of CY7B933)
- DONE bit (sends DONE signal to FE)
- CONN bit indicating (when HIGH) that connection with Hotlink Transmitter in FE board is valid (LOW means that connection failed)
- SRQ bit for SRQ signal software setting/resetting
- SRQ\_R bit for resetting SRQ Trigger
- DSTREAM bit indicating (when HIGH) that Data Transfer from FE board is in progress
- BIST bit for setting Hotlink Built-In Self-Test (BIST)
- BISOK bit indicating BIST status
- TOUT\_DSTR, TOUT\_BIST, TOUT\_REFR, RVS\_DSTR, RVS\_BIST and OVF\_DSTR bits indicating the error status of different
  - (TOUT, RVS, OVF) sources during different modes (Data Transfer
  - (DSTR), Reframe (RFR) and Built-In Self-Test (BIST)
- INTSCC bit indicating the Interrupt Request from SCC
- CSR for each section is based on separate ALTERA EPM128S chip.

MUON READOUT CARD CONTROL/STATUS REGISTER

Bit #	Read/ Write	WRITE		READ	
		NAME	REMARKS	NAME	REMARKS
0	R	-	-	BUSY1	FE signal
1	R	-	-	BUSY2	FE signal
2	R	-	-	ERROR1	FE signal + Set Register
3	R	-	-	ERROR2	Error2 Sources OR * Mask Register
4	R/W	BUSY1_M	Busy1 Mask Register	BUSY1_M	Busy1 Mask Register
5	R/W	BUSY2_M	Busy2 Mask Register	BUSY2_M	Busy2 Mask Register
6	R/W	ERROR1_M	Error1 Mask Register	ERROR1_M	Error1 Mask Register
7	R/W	ERROR1_S	Error1 Set Register	ERROR1_S	Error1 Set Register
8	R/W	TOUT_S	Setting of Time-out Error	TOUT	Status of Time-out Error Trigger
9	R/W	RVS_S	Setting of RVS Error	RVS	Status of RVS Error Trigger
10	R/W	OVF_S	Setting of Overflow Error	OVF	Status of OVF Error Trigger
11	W	RESET	Reset TOUT, RVS, OVF Bits and CY7B933 Control Logic	-	-
12	R/W	ERROR2_M	Error2 Mask Register	ERROR2_M	Error2 Mask Register
13	R	-	-	CONN	MRC-FE Connection
14	R/W	SRQ_R	Reset SRQ Trigger	DSTREAM	CY7B933 Data Transfer Status ("1" when DSTREAM exists)
15	R/W	DONE	Send DONE signal to FE	SRQ_DSTR	"1" when DSTREAM ended
16	R/W	REF	Reframe the CY7B933	REFD	Has reframed
17	R/W	SRQ_M	Service Request Mask	SRQ_M	Service Request Mask
18	R/W	SRQ	SRQ to Fan-out	SRQ	SRQ Status
19	R/W	BIST	Set BIST Mode	BISOK	BIST Status from CY7B933 Control Logic
20	R	-	-	TOUT_DSTR	Time-out during Data Transfer
21	R	-	-	TOUT_BIST	Time-out during Self-Test
22	R	-	-	TOUT_REFR	Time-out during Reframe
23	R	-	-	RVS_DSTR	RVS during Data Transfer
24	R	-	-	RVS_BIST	RVS during Self-Test
25	R	-	-	OVF_DSTR	OVF during Data Transfer
26	R	-	-	INTSCC	Interrupt from SCC
27		-	-	-	-
28		-	-	-	-
29		-	-	-	-
30		-	-	-	-
31		-	-	-	-

## **MECHANICAL PARAMETERS AND FRONT PANEL**

The MRC is realized as one-slot VME 9U module on PCB 14.4'' x 11'' size. There are 16 LED on the front panel (seven for each channel) indicating the following conditions:

- disconnection with between FE board and MRC (DISCONN, red);
- Data Transfer from FE board is in progress (DSTREAM, green);
- ERROR1 is present (ERROR1, red);
- ERROR2 is present (ERROR2, red);
- BUSY1 is present (BUSY1, red);
- BUSY2 is present (BUSY2, red);
- SRQ is present (SRQ, green).
- +5 and -5,2 power supply are on.

There is also 16-pin AMP connector for coaxial cables on the front panel.

# **CHANGES FROM FIRST PROTOTYPE**

There are a number of small changes which are to be made from the first prototype. The main changes are in the clock receiver logic, and the user of max7128s where ever there was a EPX780.

## **EPM7128s substitution.**

The EPM7128s is not pin for pin compatible with the EPX780. Since the EPX780 is no longer in production we are forced to make the substitution. The following shows the changes which will need to be made.

## **HOT-LINK COMLINEAR substitution.**

In the first prototype the data from the FE came via ASTRO cable and fed into a transformer coupled bias network on the MRC. This network coupled the MC10H116 receiver with cable. The output of the MC10H116 then fed the CY7B933 HOT-Link chip. In the new design the bias network has been replaced with a COMLINEAR (CLC014) cable receiver chip which requires far less parts although transformer coupling is still used.

## **ENCODED RF CHANGES**

The new design has the SYNC gap encoded. The high speed signals are still received differentially on the MRC from the back-plane and are then transmitted to the FE via astro cable.

## D0 Muon Upgrade Backplane

### J7 and J8 Connectors Pin Assignment

Pin	Signal	Pin	Signal
1	Xing1+	2	Xing1-
3	Xing2+	4	Xing2-
5	Xing3+	6	Xing3-
7	Xing4+	8	Xing4-
9	Xing5+	10	Xing5-
11	Xing6+	12	Xing6-
13	Xing7+	14	Xing7-
15	Xing8+	16	Xing8-
17	INIT+	18	INIT-
19	L1ACC+	20	L1ACC-
21	ERR2+	22	ERR2-
23	L2ACC+	24	L2ACC-
25	L2REJ+	26	L2REJ-
27	TxDAT+	28	TxDAT-
29	DONE+	30	DONE-
31	STRB+	32	STRB-
33	RxDAT+	34	RxDAT-
35	ERR1+	36	ERR1-
37	BUSY1+	38	BUSY1-
39	BUSY2+	40	BUSY2-
41	GND	42	GND
43	GND	44	GND
45	GND	46	GND
47	GND	48	GND
49	GND	50	GND

#### Notes

1. J7 is referred to Section A of MRC; J8 is referred to Section B of MRC.
2. 10 pins (41-50) of J8 are connected to GND at the backplane; 10 pins (41-50) of J7 are connected to GND within VME module.
3. TxDAT denotes MRC's UART transmitter output; RxDAT denotes MRC's UART receiver input.

## D0 MUON BACKPLANE J1 CONNECTOR

Pin	A	B	C
1	D00	BBSY/	D08
2	D01	BCLR/	D09
3	D02	ACFAIL/	D10
4	D03	BG0IN/	D11
5	D04	BG0OUT/	D12
6	D05	BG1IN/	D13
7	D06	BG1OUT/	D14
8	D07	BG2IN/	D15
9	GND	BG2OUT/	GND
10	SYSCLK	BG3IN/	SYSFAIL/
11	GND	BG3OUT/	BERR/
12	DS1/	BR0/	SYSRESET/
13	DS0/	BR1/	LWORD/
14	WRITE/	BR2/	AM5
15	GND	BR3/	A23
16	DTACK/	AM0	A22
17	GND	AM1	A21
18	AS/	AM2	A20
19	GND	AM3	A19
20	IACK/	GND	A18
21	IACKIN/	SERCLK	A17
22	IACKOUT/	SERDAT	A16
23	AM4	GND	A15
24	A07	IRQ7/	A14
25	A06	IRQ6/	A13
26	A05	IRQ5/	A12
27	A04	IRQ4/	A11
28	A03	IRQ3/	A10
29	A02	IRQ2/	A09
30	A01	IRQ1/	A08
31	-12V	+5VSTDBY	+12V
32	+5V	+5V	+5V

## D0 MUON BACKPLANE J2 CONNECTOR

	A	B	C
1	RF+	+5V	RF-
2	GND	GND	GND
3	SYNC GAP+	RESERVED	SYNC GAP-
4	GND	A24	GND
5	RESET+	A25	RESET-
6	GND	A26	X1
7	X2	A27	GND
8	GND	A28	Pin
9	X4	A29	GND
10	GND	A30	X5
11	X6	A31	GND
12	GND	GND	X7
13	X8	+5V	GND
14	GND	D16	L1ACC
15	L1REJ	D17	GND
16	GND	D18	L2ACC
17	L2REJ	D19	GND
18	GND	D20	INIT
19	BUSY1	D21	GND
20	GND	D22	BUSY2
21	ERR1	D23	GND
22	GND	GND	ERR2
23	SRQ	D24	GND
24		D25	
25		D26	
26		D27	
27		D28	
28		D29	
29		D30	
30		D31	
31		GND	
32		+5V	

## D0 MUON BACKPLANE J3 CONNECTOR

Pin	A	B	C
1	GND	GND	GND
2	X1B- (J8:2)	X1B+ (J8:1)	X2B- (J8:4)
3	X2B+ (J8:3)	X3B- (J8:6)	X3B+ (J8:5)
4	X4B- (J8:8)	X4B+ (J8:7)	X5B- (J8:10)
5	X5B+ (J8:9)	X6B- (J8:12)	X6B+ (J8:11)
6	X7B- (J8:14)	X7B+ (J8:13)	X8B- (J8:16)
7	X8B+ (J8:15)	INITB- (J8:18)	INITB+ (J8:17)
8	L1ACCB- (J8:20)	L1ACCB+ (J8:19)	L1REJB- (J8:22)
9	L1REJB+ (J8:21)	L2ACCB- (J8:24)	L2ACCB+ (J8:23)
10	L2REJB- (J8:26)	L2REJB+ (J8:25)	SCCTRB- (J8:28)
11	SCCTRB+ (J8:27)	DONEB- (J8:30)	DONEB+ (J8:29)
12	RESB- (J8:32)	RESB+ (J8:31)	SCCRCB- (J8:34)
13	SCCRCB+ (J8:33)	ERR1B- (J8:36)	ERR1B+ (J8:35)
14	BUSY1B- (J8:38)	BUSY1B+ (J8:37)	BUSY2B- (J8:40)
15	BUSY2B+ (J8:39)	X1A- (J7:2)	X1A+ (J7:1)
16	X2A- (J7:4)	X2A+ (J7:3)	X3A- (J7:6)
17	X3A+ (J7:5)	X4A- (J7:8)	X4A+ (J7:7)
18	X5A- (J7:10)	X5A+ (J7:9)	X6A- (J7:12)
19	X6A+ (J7:11)	X7A- (J7:14)	X7A+ (J7:13)
20	X8A- (J7:16)	X8A+ (J7:15)	INITA- (J7:18)
21	INITA+ (J7:17)	L1ACCA- (J7:20)	L1ACCA+ (J7:19)
22	L1REJA- (J7:22)	L1REJA+ (J7:21)	L2ACCA- (J7:24)
23	L2ACCA+ (J7:23)	L2REJA- (J7:26)	L2REJA+ (J7:25)
24	SCCTRA- (J7:28)	SCCTRA+ (J7:27)	DONEA- (J7:30)
25	DONEA+ (J7:29)	RESA- (J7:32)	RESA+ (J7:31)
26	SCCRCA- (J7:34)	SCCRCA+ (J7:33)	ERR1A- (J7:36)
27	ERR1A+ (J7:35)	BUSY1A- (J7:38)	BUSY1A+ (J7:37)
28	BUSY2A- (J7:40)	BUSY2A+ (J7:39)	GND
29	GND	GND	GND
30	GND	GND	GND
31	GND	GND	GND
32	-5V	-5V	-5V

# DOWNLOADING, CONTROLLING & MONITORING

The operation of the system will require downloading, controlling, and monitoring of data and registers within the D0\_muon system. This will be done via the VME based 68020 modules at system start-up, at system failure, or when the system needs new information. The normal data path is through the VBD. The 68020 VME crate controller has a program which will be written in assembly code which will respond to interrupts from the MRC via the MFC. The VBD will then be informed that there is data to be readout.

## DIAGNOSTICS AND TEST FIXTURE DESCRIPTION

The D0 MUON MRC module will attempt to provide as much testing facility on-board as possible. The functions of the board which require external detection or generation are the input receivers and the custom back-plane and connections to the FE.

The new test stand software will run diagnostic tests on incoming and suspect or broken cards which have been removed from the D0 MUON system. There are two types of diagnostic tests: single module diagnostics and system or multiple module diagnostics. For some test a VME processor, an FE and a MFC will be needed to communicate with the MRC module. The following are some expected test:

- writing to module registers
- reading from module status registers
- reading module data
- mode setups
- triggers tests
- buffers tests
- data validity tests

There are two type of data validity tests. The slower yet more flexible test allows the VME to generate patterns for reading and writing control and data registers.

In a faster test the FE system will receive data at full speed through the data input connectors. The MRC modules will not know if the data originated from the FE, or from a test board. Such multi-module test might only occur in a D0 teststand.

Some of the test stand diagnostics software will run from a remote terminal and will be implemented with TCL. This will permit the code to be ported to the actual D0 system. The connection to the MRC will be made through the VME crate controller.

The current test stand software has the following features.

1. The graphics and human interface part of the code is independent of the VME code.
2. The VME processor code can execute quickly and can use the VME DEBUG commands.
3. The VME code can be down loaded such that each diagnostic routine can be executed independently without having to build a large program. This may mean it will employ the standard debug command set as well as additional custom code.
4. The VME code will communicate to the human interface in short cryptic messages. These messages will be designed to interface with the human interface software which will exist on a remote machine. These messages can be interpreted as text but are not designed as the best visual interface. The current version runs on a PC under NT or Windows 95.
5. The code has built in looping for each routine and will report responses at a rate determined by the users request. That is if you select a mode that reports only on error the VME code will execute until an error occurs or until the routine completes without error. If the user selects mode report all actions the routine will give detailed reports of all actions as they occur. Other modes will be described later.
6. For now the graphic interface is TCL/TK.
7. The code on the PC is broken into the TCL/TK interface and C code executables which are called by a mailbox type interface to the TCL/TK.
8. The software that interfaces to the VME code from the TCL/TK interface will be transparent to the user and is dependent on the VME processor. For the processor with the RS232 interface only we will use a connection through a terminal server or a direct connection. If the processor has an Ethernet connection (i.e. VME167) we will also connect via rlogin through the terminal server. Each processor has its own startup mode. Each will require an interface to download the programs.
9. The programs that run on the VME processor will run to completion and exit for each command. This will aid implementing them in the field.

## **SUBROUTINES IN PC BASED CODE**

### **Change\_params**

Currently a piece of code which modifies files to setup the parameters for the next operation.

### **access\_vme**

This section of routines form a standard set of operation which can be used to access the VME. Looping, word size and masking are included.

### **d0\_test\_loop**

This is currently the main test loop. Many of the changes for customizing test are anticipated to be made to this section of code. When the VME processor is connected with RS232 it will not have access to any files directly. All values will remain internal to its memory or are passed as text between it and the host processor.

## **d0\_test\_code**

This is a program callable from d0\_test\_loop. It can be used for testing any piece of code. It has no purpose except to test prototype code. In the current implementation the user first edits a text file to include the actual DEBUG commands which are to be sent to the VME processor. The lines are then typed to the processor and executed just as if they were being entered at the console.

## **d0\_test\_init**

This routine performs the initial download of parameters.

## **d0\_key**

This file is the place where RS232 I/O is controlled for all other code.

## **d0\_event**

Currently this code meant to be test which involve more than one module.

## **d0\_messages**

In previous versions we used this routine as a central program to send all messages to the host. In future modifications the decoding of the message into text will be done on the host computer via TCL/TK.

## **d0\_params.h**

Defines the structures and values for items such that they can be set and modified in one place. All frequently used constants are defined here and given a symbolic name.

## **d0\_vars.c**

This program was used to make all params defined before they were called. This was a trick that worked well with the VME167 which down loads each of the subprograms without linking them. In this routine the d0params.h include file and d0vars.h file are used to set up the globals for all other programs.

## **d0\_vars.h**

This piece of code was used primarily for making all variables and programs global.

## STRUCTURE OF COMMANDS

*Current COMMAND file words as defined in D0test and d0\_params.h used as parameters for each VME instruction.*

- 1 address
- 2 mrc\_address1
- 3 mrc\_address2
- 4 offset
- 5 data
- 6 single\_data
- 7 data\_mask
- 8 single\_offset
- 9 loop\_count
- 10 test\_mode
- 11 address\_mode
- 12 block\_size
- 13 b\_per\_wrd
- 14 inputfile
- 15 outputfile
- 16 configfile
- 17 ev\_config\_file

## CURRENT COMMANDS

Read2Screen  
Read2file  
WriteBlock  
ReadSingle  
WriteSingle  
WriteFromFile  
SystemTests  
ChangeTestAdd  
ChangeBlockSize  
ChangeInputFile  
ChangeOutputFile  
ChangeLoopCount  
ChangeTestMode  
ChangeAddressMode  
ChangeDataMask  
WriteBufferMenu  
DisplayReatBuffer  
DisplayVMecycle  
CheckAdd  
ResetErrors

ChangeAll  
LoadConfig  
SaveConfig  
ChangeTestEnables  
EnableAllTests  
DisableAllTests  
Readfailsuccess  
Changetemp\_errs\_max  
ChangeModuleCount  
DoTrig  
EnableTestGroup  
Changecrates\_in\_stream  
ChangeMTM\_mode  
TestCode  
EVENTS test  
AUTO\_test  
CLR\_ERRS  
REFRESH  
EVENT\_CFG\_load

## **D0GSE code**

Some testing of code which could be downloaded to the VME 68020 processor has been done. Initial testing has been done with simple calls to the DEBUG command of the processor.