# *User Manual for the CAMAC RS-485 Access Board Set*

**An interface module for testing the DYC3 readout controller**

*Principal Author:*
Richard Kwarciany

*Contributing Authors:*
Robert Forster
James Franzen

Fermilab, United States Department of Energy
On-line Support Department
Data Acquisition Hardware Group
December 15, 1995

# DISCLAIMER NOTICE

This material was prepared as a part of work sponsored by the United States Department of Energy. The Department of Energy, Universities Research Association, Inc., and their agents and employees, make no warranty, express or implied, and assume no legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, nor represent that its use would not infringe privately owned rights.

# 1. Overview

The CRABS module combines the functionality of a CAMAC Input/Output FIFO memory module with an output buffer that emulates FERA devices, an output buffer which emulates DART data sources, and an input buffer that emulates DART data buffers. The FERA output may be used to test FERA devices such as the 4301 readout controller, but it's main purpose is to provide data input to the DYC3 without requiring a full FERA system. On the RS-485 side, the CRABS acts as an RS-485 input FIFO module, which accepts data driven on RS-485 from the DYC3. The module can also output data in the RS-485 buffer through a DART style RS-485 port. The RS-485 input buffer function replaces, for test purposes, a DC-2/DM-115 combination, a VME/VSB dual-ported memory, a VME master, and an EBI. The RS-485 output buffer function replaces data sources such as the DYC3 and the FSCC.

Module functionality has been divided into data input, and data output modules, and are implemented in two separate CAMAC boards. The data input module (CRABS/INPUT or CRABI) can function as a stand-alone CAMAC data input buffer for RS-485 level data. The data output module (CRABS/OUTPUT or CRABO) is has no CAMAC interface, and must be attached to a CRABS/INPUT module to function.

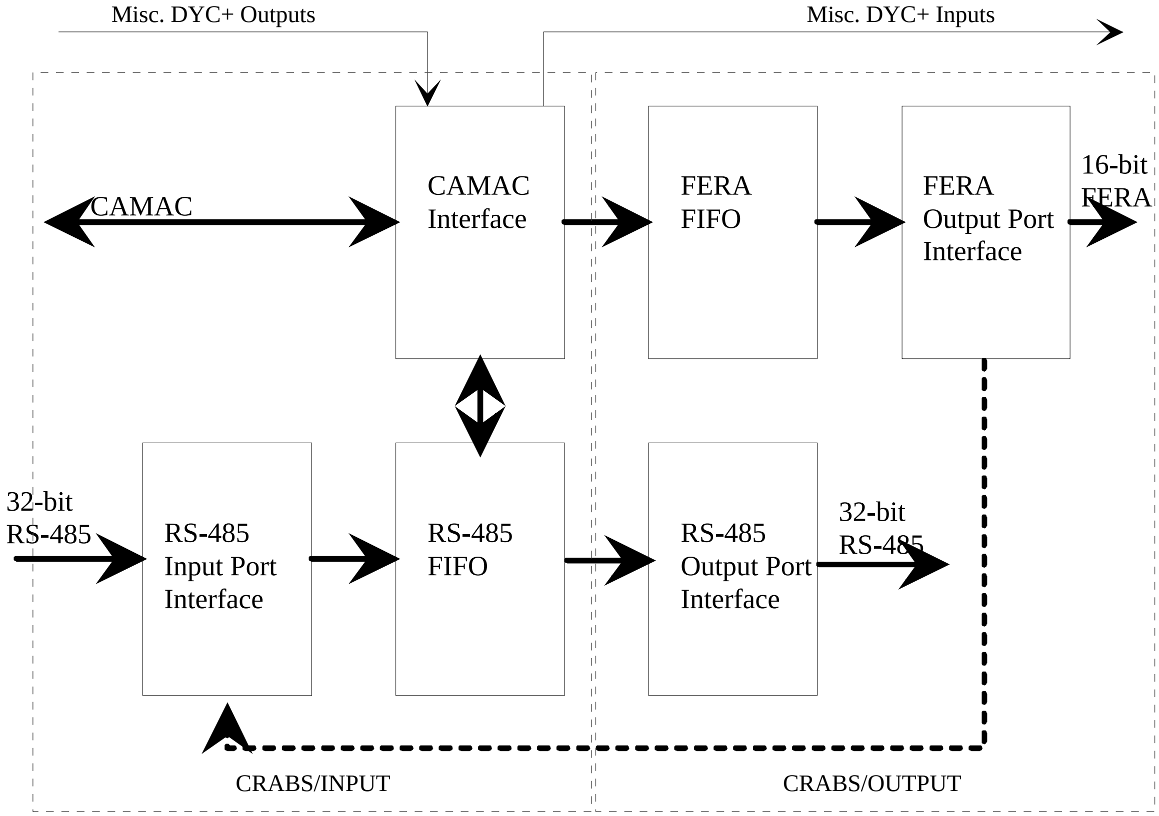
The CRABS module is designed to allow the efficient testing and repair of the DYC3 readout controller without requiring large investments of hardware and software merely to create the test stand. By reducing all external functions in to a simple CAMAC module, the design cost will be easily repaid in faster deployment and repair of DYC3 controllers in the fixed target run.

# 2. Design Requirements

The CRABS must provide the following functions:

1. An output port which provides known data on the differential ECL, FERA bus, for input to the DYC3.
2. An input port which takes as input RS-485 signals from the DYC3 and buffers the data for later readout over CAMAC, or output through the RS-485 output port.
3. An output port which buffers data from either the RS-485 input, or CAMAC, and drives RS-485 signals.
4. An input port which emulates the activity of the EBI interface, that pulls the RS-485 data out of the internal buffer of the DYC3, and is capable of reading the 'gas gauge' status bits (FF,AF,AE,EF).
5. A permit-in/permit-out port which can test the reaction of the DYC3 to the PERMIT IN signal, and which can determine that the DYC3 correctly asserts the PERMIT OUT signal.
6. A circuit which responds to the EOR signal as generated by the DYC3.
7. Various random I/O's to handle the miscellaneous level translators on the DYC3.

# 3. Functional Description



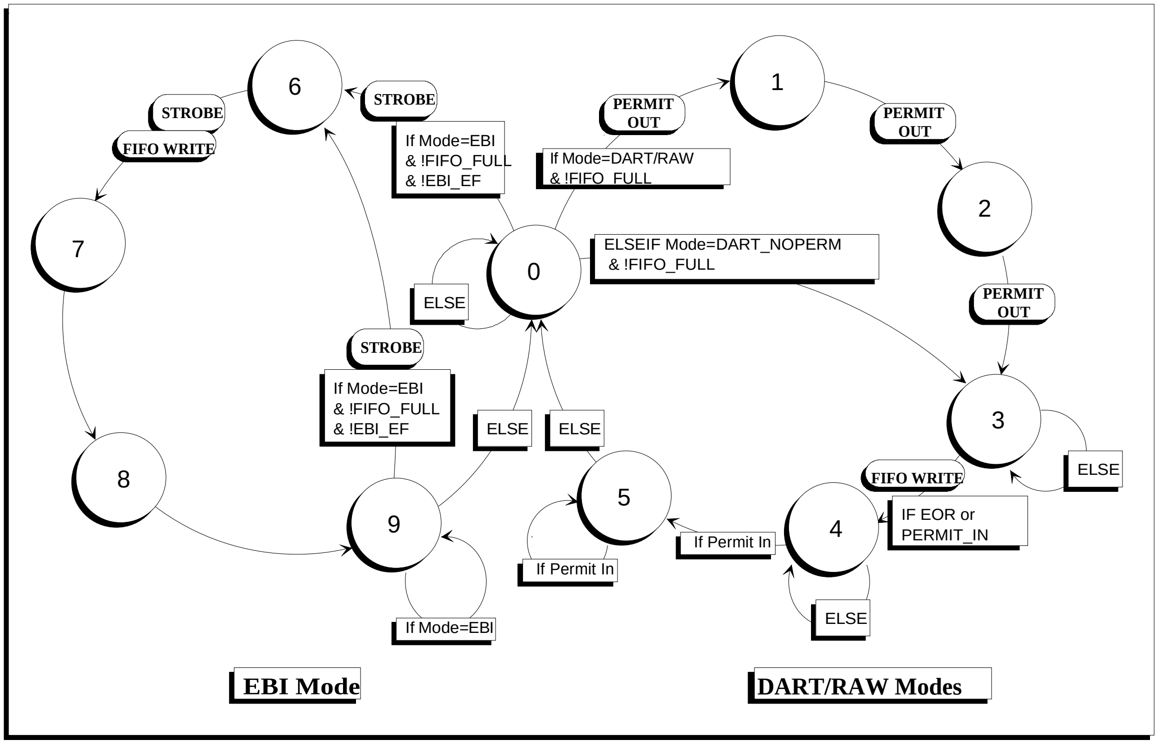

*CRABS Block Diagram*

## 3.1 RS-485 Input Port/RS-485 FIFO

### 3.1.1 RS-485 Input Port Description

In general, the RS-485 Input Port consists of differential RS-485 receivers, a 16 kilobyte (4 K x 32-bit) FIFO memory (the same FIFO as the RS-485 Output Port), and a state machine to control them. It is designed to emulate the 32-bit RS-485 level input ports of DART compliant data receivers, as well as EBI style data receivers. All control lines necessary to emulate both protocols are provided on the 50-pin, and 34-pin connectors of the CRABS/INPUT module. DART style data sources can be connected directly to the CRABS/INPUT module. EBI style data sources must use a QUADRUPOD cable adapter to connect the 50/34 style connectors to the 64/10 style connectors of the EBI. Data which has been written into the FIFO over the RS-485 Input Port can be read out of the FIFO through the CAMAC interface, or it can be output through the RS-485 Output Port. Data can also be written into the RS-485 FIFO through the CAMAC interface, or from the FERA Output Port when in Loopback mode. Modes for the RS-485 Input Port are set in the Mode register, and are described in the “RS-485 Input Port Operating Modes” section of this document. The input side of the RS-485 FIFO is controlled by the RS-485 Input Port Controller, and the output side of the RS-485 FIFO is controlled by the RS-485 Output Port Controller.

### 3.1.2 RS-485 Input Port Theory of Operation




***RS-485 Input Port Controller State Diagram***

A state machine is implemented in an EPLD to control the timing of the various control lines. In the DART modes, the state machine is idle (state 3) during data transfers. The active edge of the Strobe input from the data source is used to generate the FIFO Write signals. Upon receipt of an EOR signal, the state machine generates a FIFO Write pulse to clock the EOR in. If DART_PERM mode has been selected, the state machine then waits for Permit In before returning to state 0.

In EBI mode, the state machine drives the FIFO Write line as well as the RS-485 Strobe output to the data source. The State Machine continues clocking data into the FIFO until the EBI_EF “gas gauge” bit goes true.

### 3.1.3 RS-485 Input Port Operating Modes

| *Mode* | *Mode Name* | *Description* |
|---|---|---|
| *0* | *CAMAC* | State Machine is disabled. Data can be written into the RS-485 FIFO through the CAMAC interface. RS-485 input receivers are disabled. |
| *1* | *DART_PERM* | State Machine allows data to be clocked into the RS-485 FIFO using the DART protocol. State Machine generates a Permit Out pulse before enabling the FIFO for input, and after receipt of an EOR, waits for a Permit In pulse before allowing further data to be clocked in. |
| *2* | *DART_NOPERM* | State Machine allows data to be clocked into the RS-485 FIFO using the DART protocol. State Machine does not generate Permit Out, and ignores Permit In. |
| *3* | *EBI* | State Machine clocks data into the RS-485 FIFO using EBI protocol. Gas gauge EF bit is monitored. |

| | | |
|---|---|---|
| *4* | *LOOPBACK* | State Machine allows data to be clocked into the RS-485 FIFO from the FERA Output Port. 16 bits of FERA data are duplicated on both the high and low 16 bits of the RS-485 Input Port. |
| *5* | *RAW* | State Machine functions as in DART_PERM mode. Input receivers are enabled for "RAW" data input. The WAIT output is disabled, and EOR is ignored by the State Machine, but the state of the EOR input and the remaining seven control bits can be read in Status register C register. |
| *6* | *reserved* | |
| *7* | *reserved* | |
| *8* | *DART_PERM_SINGLE* | In this mode, the State Machine operates in a similar manor to mode 1, except it allows more control over the transmission of PERMIT_OUT to the data source. Setting the input port into mode 8 causes the port to function similarly to mode 1 (DART_PERM mode) except that upon receipt a PERMIT_IN from the data source, the state machine pauses at state zero. To send another PERMIT_OUT to the data source, the mode is changed to mode 0, and then back to mode 8. When the mode is switched back to mode 8, PERMIT_OUT is again sent to the data source. |
| *8-15* | *reserved* | |

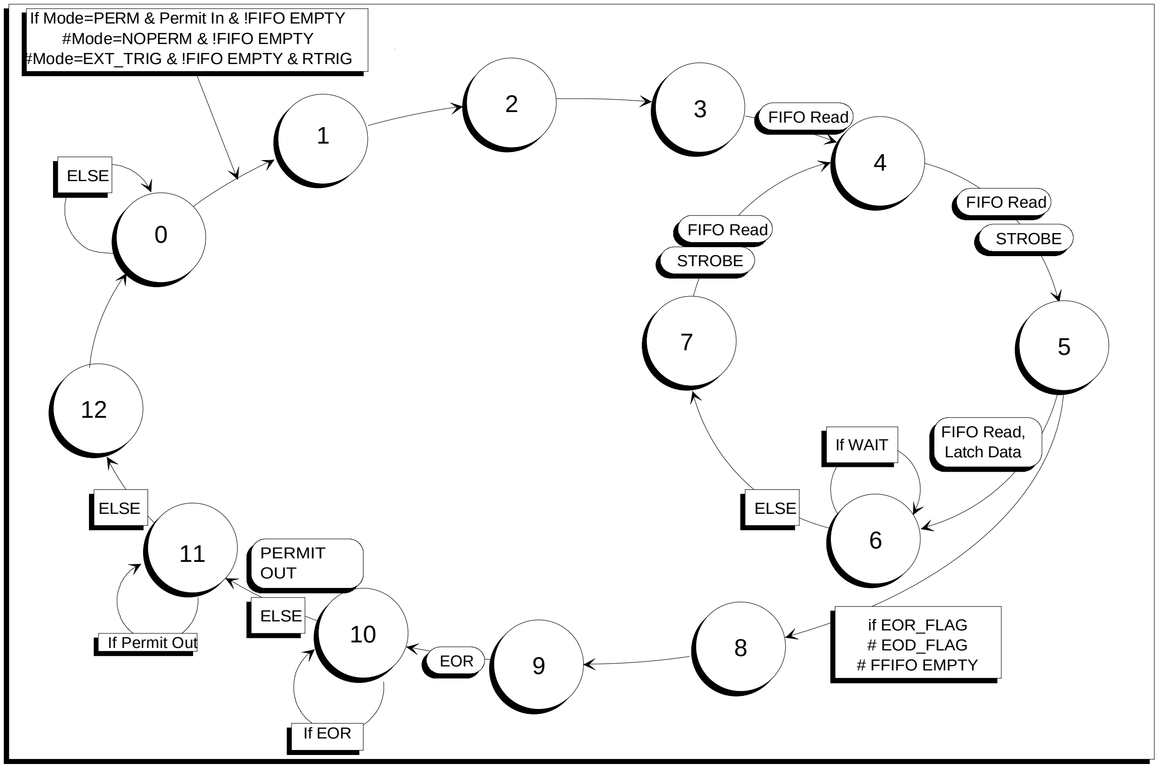
## 3.2 RS-485 Output Port/RS-485 FIFO

### 3.2.1 RS-485 Output Port Description

The RS-485 Output Port consists of differential RS-485 transmitters, a 16 kilobyte (4K x 32-bit) FIFO memory (the same FIFO as the RS-485 Input Port), and a state machine to control them. It is designed to emulate the 32-bit RS-485 level output ports of DART compliant data sources. The Output Port does *not* support EBI style data sources. Data in the RS-485 FIFO can be output through the RS-485 Output Port, or it can be read out through the CAMAC interface, once the RS-485 Output Controller has been placed in the proper mode. Modes for the RS-485 Output Port are set in the Mode register, and are described in the "RS-485 Output Port Operating Modes" section of this document. The input side of the RS-485 FIFO is controlled by the RS-485 Input Port Controller, and the output side of the RS-485 FIFO is controlled by the RS-485 Output Port Controller.

The RS-485 Output Port Controller is designed to support multiple blocks of data in the RS-485 FIFO. Data blocks are separated in the FIFO by inserting a word with either the EOB (End Of Block) or the EOR (End Of Record) bits set. These two bits function identically except that the EOR bit causes an EOR pulse to be output, whereby the EOB bit does not. As the Controller clocks the data out of the RS-485 FIFO, it checks the state of these two bits. If it sees either of them true, it halts data transmission, and continues to either start outputting the next block, or waits for Permit In (depending on the mode set). If the EOR bit caused the Controller to stop transmitting data, then an EOR pulse will be output. In either case, the word containing the flag bit true is not output.

### 3.2.2 RS-485 Output Port Theory of Operation




***RS-485 Output Port Controller State Machine Diagram***

### 3.2.3 RS-485 Output Port Operating Modes

| *Mode* | *Mode Name* | *Description* |
|---|---|---|
| *0* | *DISABLED* | State Machine is disabled. Data can be read out of the RS-485 FIFO (RFIFO) through the CAMAC interface. |
| *1* | *PERM* | Output data in RFIFO through the RS-485 Output Port using DART protocol. State Machine will wait for Permit In, then output data until the RFIFO is empty or a flag is seen. If an EOR (End Of Record) flag is seen, an EOR pulse is generated before Permit Out is generated. The state machine waits for Permit In before transmitting the next data block. If an EOB (End Of Block) flag is seen, no EOR pulse is generated, and Permit Out is generated. The state machine waits for Permit In before transmitting the next data block. If the RFIFO is emptied, Permit Out is generated. This mode is generally used for middle and last modules in the RS-485 token passing chain. |
| *2* | *PERM_RETRANS* | Output data in RFIFO through the RS-485 Output Port as in PERM mode, except that upon emptying the RFIFO, the state machine “retransmits” the RFIFO (reset the RFIFO’s data output pointer so that the same block of data may be output again) and outputs the data again. Data in the RFIFO will continuously be output until the mode is changed. |
| *3* | *PERM_TRIG* | State Machine idles until a front panel RS-485 Trigger Input pulse (NIM level) is seen. Data is then output as in PERM_RETRANS mode. |
| *4* | *NOPERM* | Output data in RFIFO through the RS-485 Output Port using DART protocol. The first data block is transmitted without waiting |

| | | |
|---|---|---|
| | | for Permit In. Subsequent data blocks are not transmitted until Permit In is received. State Machine will output data until the RFIFO is empty or a flag is seen. If an EOR flag is seen, an EOR pulse is generated before Permit Out is generated. The state machine waits for Permit In before transmitting the next data block. If an EOB flag is seen, no EOR pulse is generated, and Permit Out is generated. The state machine waits for Permit In before transmitting the next data block. If the RFIFO is emptied, Permit Out is generated. This mode is generally used for the first or only module in the RS-485 token passing chain. |
| *5* | *NOPERM_RETRANS* | Output data in RFIFO through the RS-485 Output Port as in NOPERM mode, except that upon emptying the RFIFO, the state machine "retransmits" the RFIFO (reset the RFIFO's data output pointer so that the same block of data may be output again) and outputs the data again. Data in the RFIFO will continuously be output until the mode is changed. |
| *6* | *NOPERM_TRIG* | State Machine idles until a front panel RS-485 Trigger Input pulse (NIM level) is seen. Data is then output as in NOPERM mode. |
| . *7* | *RAW* | State Machine functions as in PERM mode, except at a reduced data rate. A new data word is presented on the RS-485 data lines every 1.6μs. The data is accompanied by an 800ns Strobe pulse. The active (falling) edge of the Strobe pulse is approximately centered in each data word. The WAIT input is ignored in RAW mode, and EOR is not generated by the state machine. EOR can be output by setting the RAW EOR bit in the mode register. The remaining six control outputs of the RS-485 Output Port can be driven by setting the corresponding bits in the mode register. |
| *8* | *reserved* | |
| *9* | *PERM_FIRST* | Same as Mode 1 except the first event will be output without waiting for Permit In. This mode is used when first in Permit Chain. |
| *10* | *PERM_RETRANS_FIRST* | Same as Mode 2 except the first event will be output without waiting for Permit In. This mode is used when first in Permit Chain. |
| *11* | *PERM_TRIG_FIRST* | Same as Mode 3 except the first event will be output without waiting for Permit In. This mode is used when first in Permit Chain. |
| *12-15* | *reserved* | |

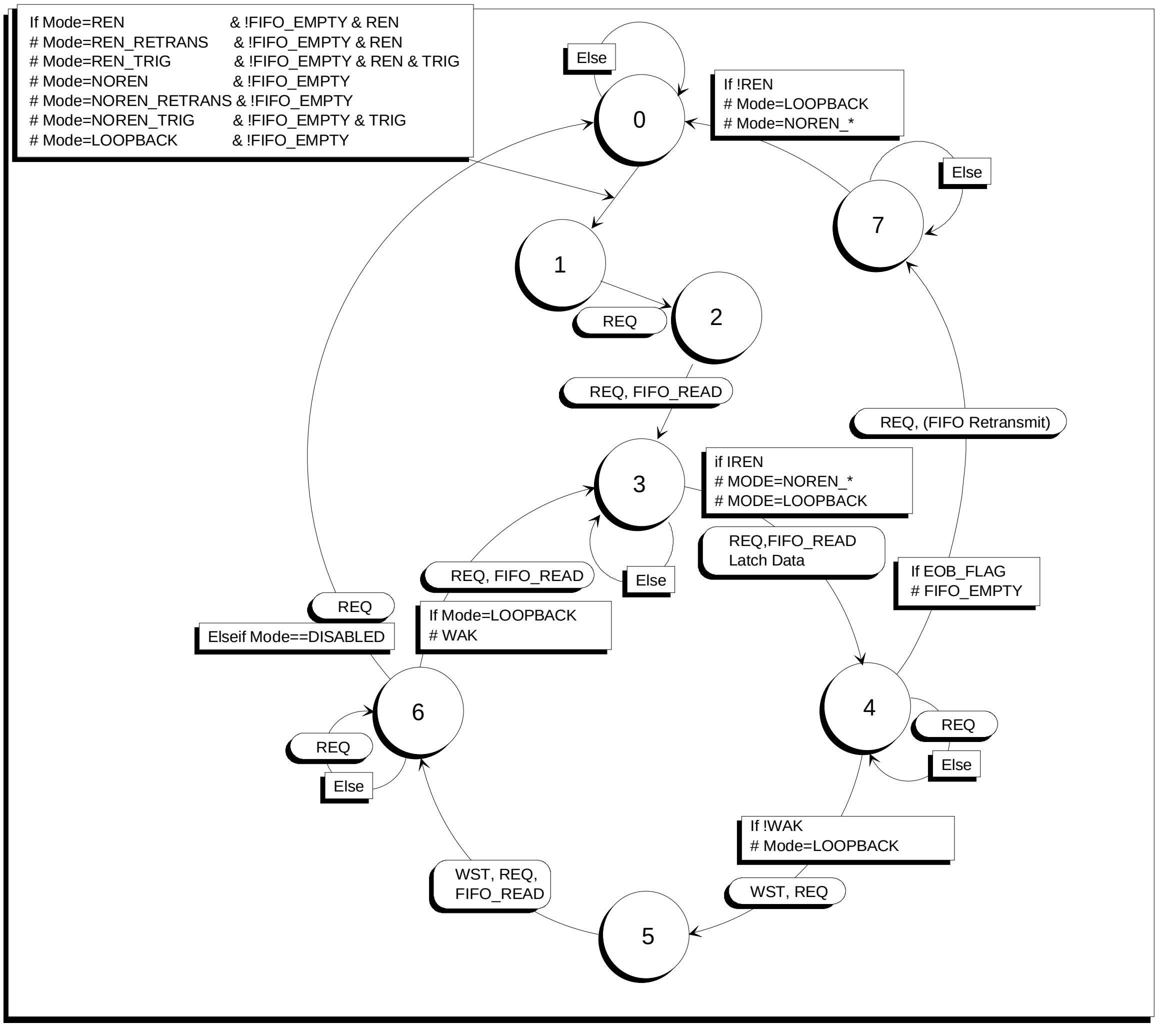
## 3.3 FERA Output Port/FERA FIFO

### 3.3.1 FERA Output Port Description

The FERA Output Port consists of differential ECL level drivers, an 16 kilobyte (8K x 16-bit) FIFO memory, and a state machine to control them and generate the FERA bus control signals. It is designed to emulate the FERA (LeCroy(tm) Model 4300 CAMAC Fast Encoding and Readout ADC) module's front panel ECL output port and Command Bus. Data in the FERA FIFO can be output through the FERA Output Port, or it can be internally connected to the RS-485 Input port and clocked into the RS-485 FIFO (in loopback mode). Data in the FERA FIFO *cannot* directly be read out through the CAMAC interface. Modes for the FERA Output Port are set in the Mode register, and are described in the "FERA Output Port Operating Modes" section of this document.

The FERA Output Port Controller is designed to support multiple blocks of data in the FERA FIFO. Data blocks are separated in the FIFO by inserting a word with the EOB (End Of Block) bit set. The FERA Output Port Controller is then placed into either REN_FLAG or NOREN_FLAG mode. Data will then be output until the EOB flag is seen or the FIFO is emptied. The word containing the flag bit true is not output.

### 3.3.2 FERA Output Port Theory of Operation




***FERA Output Port Controller State Machine Diagram***

### 3.3.3 FERA Output Port Operating Modes

| *Mode* | *Mode Name* | *Description* |
|---|---|---|
| *0* | *DISABLED* | State Machine is disabled. |
| *1* | *REN* | Output data in FERA FIFO through the FERA Output Port. Output Port will generate a REQ output, and wait for a REN response before starting the transfer. State Machine will output data until the EOB flag is reached, or the FFIFO becomes empty. REQ is negated, and PASS is generated. If the FFIFO has more data, REQ is asserted again. Upon receipt of the REN input the next block of data will be output. |
| *2* | *REN_RETRANS* | Output data in FERA FIFO through the FERA Output Port as in REN mode. Output Port will generate a REQ output, and wait for |

| | | |
|---|---|---|
| | | a REN response before starting the transfer. State Machine will output data until the FIFO is empty, drop REQ and drive the PASS output. The State Machine will then retransmit the FIFO, and raise REQ. In this way the same data block will continuously be output. |
| *3* | *REN_TRIG* | State Machine idles until a front panel RS-485 Trigger Input is seen. Data is then output as in REN_RETRANS mode. |
| *4* | *NOREN* | Output data in FERA FIFO through the FERA Output Port. Output Port will generate a REQ output, but will not wait for a REN response before starting the transfer. State Machine will output data until the EOB flag is reached, or the FFIFO becomes empty. REQ is negated, and PASS is generated. If the FFIFO has more data, REQ is asserted again. The next block of data will then be output. |
| *5* | *NOREN_RETRANS* | Output data in FERA FIFO through the FERA Output Port as in NOREN mode. Output Port will generate a REQ output, but will not wait for a REN response before starting the transfer. State Machine will output data until the FIFO is empty, drop REQ and drive the PASS output. The State Machine will then retransmit the FIFO, and raise REQ. In this way the same data block will continuously be output. |
| *6* | *NOREN_TRIG* | State Machine idles until a front panel RS-485 Trigger Input is seen. Data is then output as in NOREN_RETRANS mode. |
| *7* | *LOOPBACK* | Transfer data in FERA FIFO back to the RS-485 FIFO. RS-485 Input Port must be set to LOOPBACK mode first. 16-bits of FERA data is copied to low and high 16-bits of RS-485 FIFO. LOOPBACK mode is used to test the function of the two FIFO's, the FERA Output Port Controller, and the RS-485 Input Port Controller. Data is transferred until the FFIFO becomes empty. |
| *9-15* | *reserved* | |

## 3.4 Programmable Counter

### 3.4.1 Programmable Counter Description

A 22-bit programmable counter associated with the RS-485 Input Port has been implemented as a system debugging tool. The counter is configured by setting the three counter mode bits in the Mode Register. It can be configured to count inactive to active transitions of either the STROBE I/O line (STROBE is an input in DART mode, and an output in EBI mode), the WAIT output, or the EOR input. A Reset bit and a Count Enable bit are provided in the Control Register. For counting to occur, the Reset bit must be low, and the Count Enable bit must be high. A counter overflow bit is provided in Status Register B. The value of the counter can be read at any time via the CAMAC interface, but if it is read when counting is enabled, it is possible for a read to occur coincident with an increment of the counter, and produce an invalid result. Therefore, it is recommended that the Count Enable bit be cleared before the value of the counter is read. The maximum value for the Programmable Counter is 4,194,303. Incrementing the counter from 4,194,303 will cause the count value to wrap to 0, and set the Counter Overflow bit high.

### 3.4.2 Programmable Counter Operating Modes

| ***Mode*** | ***Mode Name*** | ***Description*** |
|---|---|---|
| *0* | *DISABLED* | Counting is disabled. Counter value is affected only by the Reset bit, or by a board reset. |
| *1* | *COUNT_DART_STROBES* | If the Count Enable bit is high, then the counter value is incremented by one count, each time an inactive to active transition of the DART STROBE input line occurs. |

| | | |
|---|---|---|
| | | Counter value is set to zero by the Reset bit, or by a board reset. |
| 2 | *COUNT_EBI_STROBES* | If the Count Enable bit is high, then the counter value is incremented by one count, each time an inactive to active transition of the EBI STROBE output line occurs. Counter value is set to zero by the Reset bit, or by a board reset. |
| 3 | *COUNT_WAITS* | If the Count Enable bit is high, then the counter value is incremented by one count, each time an inactive to active transition of the WAIT output line occurs. Counter value is set to zero by the Reset bit, or by a board reset. |
| 4 | *COUNT_EORS* | If the Count Enable bit is high, the counter value is incremented by one count, each time an inactive to active transition of the EOR input line occurs. Counter value is set to zero by the Reset bit, or by a board reset. |
| *5-7* | *Reserved* | Unused modes are equivalent to mode 0. |

# 4. Control and Monitor Functions

The CRABS is controlled through a control register and a mode register accessed via CAMAC. Each of its features is enabled and disabled by setting the appropriate bits in these registers. The input and output ports are configured via the mode register. Test data to be output through the FERA Output FIFO is written to the CRABS through CAMAC as well. Data can be written or read from the RS-485 FIFO either through CAMAC or over separate RS-485 ports. Since the test FERA output port of the CRABS buffers data using FIFO type memory, writing the output test data over CAMAC is done by serially accessing a single CAMAC Sub-address location in the module. Data buffered in the RS-485 FIFO can be read out of, and written into the CRABS in a similar manor, except that since the input FIFO is 32-bits wide, two CAMAC Sub-addresses have been allocated to the input and also to the output of the port.

The CRABS features a “loop-back” mode whereby the output FIFO can be internally routed to the Input FIFO for CRABS data path sanity checking.

## 4.1 Programmer’s Model

### 4.1.1 Register Map

**F0** (bits 24 … 1)

| | |
|---|---|
| **A0** | **RS-485 FIFO READ LOW** |
| **A1** | **RS-485 FIFO READ HIGH** |
| **A2** | **No Function** |
| **A3** | **COUNTER READ** |

**F16** (bits 24 … 1)

| | |
|---|---|
| **A0** | **RS-485 FIFO WRITE LOW** |
| **A1** | **RS-485 FIFO WRITE HIGH** |
| **A2** | **FERA FIFO WRITE** |
| **A3** | **No Function** |

**F1** (bits 24 … 1)

| | |
|---|---|
| **A0** | **CONTROL/STATUSA READ** |
| **A1** | **MODE READ** |
| **A2** | **STATUSB READ** |
| **A3** | **STATUSC READ** |

**F17** (bits 24 … 1)

| | |
|---|---|
| **A0** | **CONTROL WRITE** |
| **A1** | **MODE WRITE** |
| **A2** | **No Function** |
| **A3** | **No Function** |

### 4.1.2 Register Definitions

#### *4.1.2.1 RS-485 FIFO Write Low Word (F16 A0)*

| 24 | | | | | 19 | 18 | 17 | 16 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | EOB | EOR | D15 | D0 |

This register contains the low 16 bits of the RS-485 FIFO's (RFIFO) inputs, and the two data flags. Writing this register pushes the written value into the low half of the RFIFO, and increments the low word of the FIFO's internal write pointer by one. The FIFO can only be written through the CAMAC port when the RS-485 Input Controller is disabled (mode = 0). Writing this register when the RS-485 Input Controller is enabled (mode <> 0) has no effect. CAMAC writes to the RFIFO when the RFIFO is full cause a "Q=0" response on the CAMAC bus.

Inserting a word in the RFIFO with the EOR bit high causes the State Machine to stop outputting data when this flag is encountered, and generated an End Of Record pulse on the RS-485 cable. This flag is active high.

The EOB flag is used to separate blocks of data in the RFIFO without generating an End Of Record pulse. Function of this flag is identical to the EOR flag except that no EOR pulse is generated on the RS-485 cable.

#### *4.1.2.2 RS-485 FIFO Read Low Word (F0 A0)*

| 24 | | | | | 19 | 18 | 17 | 16 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | EOB | EOR | D15 | D0 |

This register contains the low 16 bits of the RS-485 FIFO's outputs and the two data flags. Reading this register returns the current word being output by the low half of the RFIFO, and increments the low word of the FIFO's internal read pointer by one. The RFIFO can only be read through the CAMAC port when the RS-485 Output Controller is disabled (mode = 0). Data returned when the RS-485 Output Controller is enabled (mode <>0) is undefined. CAMAC reads of the RFIFO when the RFIFO is empty cause a "Q=0" response on the CAMAC bus.

#### *4.1.2.3 RS-485 FIFO Write High Word (F16 A1)*

| 24 | | | | | 19 | 18 | 17 | 16 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | SPARE FLAG1 | SPARE FLAG0 | D31 | D16 |

This register contains the high 16 bits of the RS-485 FIFO's inputs, and the inputs to the RFIFO's two spare flag bits. Writing this register pushes the written value into the high half of the FIFO, and increments the high word of the FIFO's internal write pointer by one. The FIFO can only be written through the CAMAC port when the RS-485 Input Controller is disabled (mode = 0). Writing this register when the RS-485 Input Controller is enabled (mode <> 0) has no effect. CAMAC writes to the RS-485 FIFO when the FIFO is full cause a "Q=0" response on the CAMAC bus.

The spare flag bits are passed through the RFIFO, but have no effect on the operation of the RS-485 Output Port.

#### *4.1.2.4 RS-485 FIFO Read High Word (F0 A1)*

| 24 | | | | | 19 | 18 | 17 | 16 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | SPARE FLAG1 | SPARE FLAG0 | D31 | D16 |

This register contains the high 16 bits of the RS-485 FIFO's outputs, and the two spare flag bits. Reading this register returns the current word being output by the high half of the FIFO, and increments the high word of the FIFO's internal read pointer by one. The FIFO can only be read through the CAMAC port when the RS-485 Output Controller is disabled (mode = 0). Data returned when the RS-485 Output Controller is enabled (mode <> 0) is undefined. CAMAC reads of the RS-485 FIFO when the FIFO is empty cause a "Q=0" response on the CAMAC bus.

### *4.1.2.5 FERA FIFO Write (F16 A2)*

| 24 | | | | | 19 | 18 | 17 | 16 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | SPARE FLAG | EOB FLAG | D15 | D0 |

This register contains the 16 bits of the FERA Output FIFO's inputs, and the inputs to the two FERA FIFO flags. Writing this register pushes the written value into the FERA FIFO, and increments the FIFO's internal write pointer by one. The FIFO can be written at any time, however, it is not recommended that the FERA FIFO be written when the FERA Output Controller is enabled. CAMAC writes to the FERA FIFO when the FIFO is full cause a "Q=0" response on the CAMAC bus.

The Event Flag is used by the FERA Output Controller (in certain modes) as a marker to separate data blocks in the FIFO. Writing a word to the FIFO with the Event Flag set causes the FERA Output Controller to recognize the word as a separator between data blocks in the FIFO. The word containing the flag bit set is never output as data.

The Spare Flag is currently not used and has no effect on the operation of the FERA port.

### *4.1.2.6 Counter Read (F0 A3)*

| 24 | 23 | 22 | 21 | 20 | 1 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | OVR FLW | D19 | D0 |

The Counter Read Register returns the current value of the Programmable Counter, and the overflow bit. The counter can be read at any time, however, if it is read when the counter value is changing, the value returned may be erroneous. The overflow bit is set to a one when the counter increments from decimal 1,048,575 (hex FFFFF) to 1,048,576 (hex 100000), and cleared by the Counter Reset bit or a Board Reset.

### *4.1.2.7 Control Register Write (F17 A0)*

| 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLR EOR EN | CLR TSTRB OUT | CLR EOE OUT | CLR RESET OUT | CLR GATE OUT | CLR CLEAR OUT | CLR CNT ENAB | CLR CNT RESET | CLR FFIFO RE-X | CLR FFIFO RESET | CLR RFIFO RE-X | CLR RFIFO RESET | SET EOR EN | SET TSTRB OUT | SET EOE OUT | SET RESET OUT | SET GATE OUT | SET CLEAR OUT | SET CNT ENAB | SET CNT RESET | SET FFIFO RE-X | SET FFIFO RESET | SET RFIFO RE-X | SET RFIFO RESET |

The CRABS module's Control Bits are set and cleared by separate set and clear bits in the Control Register. The lower 12 bits of the Control Register contains the Set bits, and the Higher 12 bits contain the Clear bits. Writing zeros to any bit in the control register has no effect. Bit Definitions are as follows:

*Bit 1:* Set RFIFO Reset. Places the RS485 Input Controller, RS485 Output Controller, RS485 FIFO, and the Programmable Counter into reset. Counter value is set to zero. FIFO is cleared.

*Bit 13:* Clear RFIFO Reset.

*Bit 2:* Set RFIFO Retransmit. Sets the Retransmit input to the RS485 FIFO true. This allows retransmission of data previously read out of the FIFO. The Retransmit bit must be set, then cleared before data can be read from, or output from the FIFO. This bit can be set at any time, however, it is recommended that the FIFO be retransmitted only when the RS-485 Input Controller, and the RS-485 Output Controller are disabled or known to be idle.

*Bit 14:* Clear RFIFO Retransmit.

*Bit 3:* Set FFIFO Reset. Places the FERA Output Controller, and FERA FIFO into reset. FIFO is cleared.
*Bit 15:* Clear FFIFO Reset.
*Bit 4:* Set FFIFO Retransmit. Sets the Retransmit input to the FERA FIFO true. This allows retransmission of data previously readout of the FIFO. The Retransmit bit must be set, then cleared before data can be output from the FIFO. This bet can be set at any time, however, it is recommended that the FIFO be retransmitted only when the FERA Output Controller is disabled or known to be idle.
*Bit 16:* Clear FFIFO Retransmit.
*Bit 5:* Set Counter Reset. Sets the Counter Reset bit true, causing the count value to be set to zero, and counting to be disabled.
*Bit 17:* Clear Counter Reset.
*Bit 6:* Set Counter Enable. Enables the Programmable Counter. When disabled, the count value does not change unless it is reset.
*Bit 18:* Clear Counter Enable.
*Bit 7:* Set CLEAR Output. Sets the front panel NIM level CLEAR output on the CRABS Output board true.
*Bit 19:* Clear CLEAR Output.
*Bit 8:* Set GATE Output. Sets the front panel NIM level GATE output on the CRABS Output board true.
*Bit 20:* Clear GATE Output.
*Bit 9:* Set RESET Output. Sets the front panel NIM level RESET output on the CRABS Output board true.
*Bit 21:* Clear RESET Output.
*Bit 10:* Set EOE Output. Sets the front panel NIM level EOE output on the CRABS Output board true.
*Bit 22:* Clear EOE Output.
*Bit 11:* Set Trigger_Strobe Output. Sets the front panel RS-485 Trigger Strobe (bottom pins of 10-pin connector) true.
*Bit 23:* Clear Trigger Strobe Output.
*Bit 12:* Set EOR Driver Enable. Enables the RS-485 Output Port's EOR driver.
*Bit 24:* Clear EOR Driver Enable.

### *4.1.2.8 Control/StatusA Register Read (F1 A0)*

| 24–21 | 20–17 | 16–13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FERA OUTPUT CONTROLLER STATUS | RS-485 OUTPUT CONTROLLER STATUS | RS-485 INPUT CONTROLLER STATUS | EOR OUT | TSTRB OUT | EOE OUT | RESET OUT | GATE OUT | CLEAR OUT | CNT ENAB | CNT RESET | FFIFO RE-X | FFIFO RESET | RFIFO RE-X | RFIFO RESET |

Bits 1 through 12 of the Control/StatusA Register reflect the status of the corresponding bits in the control register. All bits are high true. Bits 13 through 24 reflect the status of the three data I/O controllers on the CRABS module. These bits are defined as follows:

*Bits 13-16:* RS-485 Input Controller Status. Status codes for the Input Controller are defined as follows:

| *Status Code* | *Description* |
|---|---|
| 0 | State Machine is idle, waiting for either a mode to be set, or the RS-485 FIFO to become not empty. |
| 1,2 | Driving Permit Out DART or RAW mode. |
| 3 | State machine is in DART or RAW data input mode. If in DART mode, it will stay in this state until EOR is received, or the mode is set to zero. If in RAW mode, the state machine will stay in this state until the mode is set to zero. |
| 4 | State machine is in DART mode, EOR has been received, and state machine is currently waiting for Permit In. |

| | |
|---|---|
| 5 | State machine is in DART mode, Permit In has been received, and state machine is currently waiting for the Permit In signal to negate. |
| 6,7,8 | State machine is in EBI mode, data input mode. |
| 9 | State machine is in EBI mode, data input mode. Reading state 9 for more than one sample in a row indicates that the state machine is waiting for the RS-485 FIFO to become not full, or waiting for EBI EF to negate. |

*Bits 17-20:* RS-485 Output Controller Status. Status codes for the RS-485 Output Controller are defined as follows:

| *Status Code* | *Description* |
|---|---|
| 0 | State machine is idle, waiting for either a mode to be set, or for the RS-485 FIFO to become not empty, or for Permit In. |
| 1,2,3,6,7,11,14,15 | State machine is outputting data. |
| 10 | State machine is outputting data. Reading status 10 for more than one sample in a row indicates that the state machine is waiting for WAIT from the data buffer to negate. |
| 9,12,13 | State machine is outputting EOR signal on DART cable. |
| 4,8 | State machine is outputting Permit Out. |

*Bits 21-24:* FERA Output Controller Status. Status codes for the FERA Output Controller are defined as follows:

| *Status Code* | *Description* |
|---|---|
| 0 | State machine is idle, waiting for either REN to go true, or TRIG depending on mode set. |
| 1 | State machine is waiting for REN input to become true. |
| 2 | State machine is waiting for the FFIFO to become Not Empty. |
| 3 | State machine is waiting for FTRIG Trigger Input. |
| 4 | State machine is waiting for WAK. |
| 5 | State machine is waiting for WAK to negate. |
| 6 | State machine is transmitting data. |
| 7 | State machine is waiting for REN input to negate. |

### *4.1.2.9 Mode Register Write (F17 A1)*

| 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 – 13 | 12 – 9 | 8 – 5 | 4 – 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RAW SPARE 1 OUT | RAW RSBAF OUT | RAW 16/32* OUT | RAW EOR OUT | TRIG3 /RAW FEVN | TRIG2 /RAW WAIT | TRIG1 /RAW SSTB | TRIG0 | NOT USED | COUNTER MODE | FERA OUTPUT CONTROLLER MODE | RS-485 OUTPUT CONTROLLER MODE | RS-485 INPUT CONTROLLER MODE |
| (EBI AE) | | | (EBI FF) | | (EBI AF) | (EBI EF) | | | | | | |

The Mode register contains the mode control bits for the three I/O controller state machines, as well as the mode control bits for the Programmable Counter, and also eight RS-485 output bits. The eight bits allow the eight non-data RS-485 lines in the RS-485 Output Port to be driven when using RAW mode. Bits in the Mode register are defined as follows:

*Bits 1-4:* RS-485 Input Controller Mode. The following modes are defined for the RS-485 Input Controller:

| *Mode* | *Description* |
|---|---|
| 0 | CAMAC, State Machine Disabled |
| 1 | DART, use Permits and loop |
| 2 | DART, ignore Permits and loop |
| 3 | EBI |
| 4 | Internal Loop Back from FERA port |
| 5 | RAW |

| | |
|---|---|
| 6,7 | reserved |
| 8 | DART_PERM_SINGLE |
| 9-15 | reserved |

*Bits 5-8:* RS-485 Output Controller Mode. The following modes are defined for the RS-485 Output Controller:

| *Mode* | *Description* |
|---|---|
| 0 | State Machine Disabled. |
| 1 | Use Permits, Output until FIFO empty or Flag is seen. Middle or Last in permit chain. |
| 2 | Use Permits, Output until FIFO empty, then retransmit and repeat. Middle or Last in permit chain. |
| 3 | Wait for front panel RS-485 Trigger Input, then output as in mode 2. Middle or Last in permit chain. |
| 4 | Ignore Permits, Output until FIFO empty or Flag is seen. |
| 5 | Ignore Permits, Output until FIFO empty, then retransmit and repeat. |
| 6 | Wait for front panel RS-485 Trigger Input, then output as in mode 4. |
| 7 | Output data in RAW mode. Wait for Permit, then output data as in mode 1 except at a reduced rate (625Khz rate). WAIT is ignored and EOR is not generated. |
| 8 | reserved. |
| 9 | Same as Mode 1 except the first event will be output without waiting for Permit In. First in permit chain. |
| 10 | Same as Mode 2 except the first event will be output without waiting for Permit In. First in permit chain. |
| 11 | Same as Mode 3 except the first event will be output without waiting for Permit In. First in permit chain. |
| 12-14 | reserved. |
| 15 | Same as Mode 7 except the first event will be output without waiting for Permitn In. |

*Bits 9-12:* FERA Output Controller Mode. The following modes are defined for the FERA Output Controller:

| *Mode* | *Description* |
|---|---|
| 0 | State Machine Disabled. |
| 1 | Use REN/PASS, Output until FIFO empty, or Flag is reached. |
| 2 | Use REN/PASS, Output until empty, then retransmit and repeat. |
| 3 | Wait for FERA Trigger Input, then output as in mode 2. |
| 4 | Ignore REN/PASS, Output until FIFO empty, or Flag is reached. |
| 5 | Ignore REN/PASS, Output until empty, then retransmit and repeat. |
| 6 | Wait for FERA Trigger Input, then output as in mode 5. |
| 7 | Internal Loopback to RS-485 Input port. |
| 9-15 | reserved |

*Bits 13-15:* Programmable Counter Mode. The following modes are defined for the Programmable Counter:

| *Mode* | *Description* |
|---|---|
| 0 | Counter Disabled. |
| 1 | Count DART RS-485 Input Strobes. |
| 2 | Count EBI RS-485 Input Strobes. |
| 3 | Count RS-485 Input Waits. |
| 4 | Count RS-485 Input EORs. |
| 5-7 | Counter Disabled. |

*Bit 16:* Unused bit. This bit may be written to and read back, but it has no effect on the operation of the CRABS module.
*Bit 17:* TRIG0 Output. When high, the front panel Trigger ID bit 0 is driven true.
*Bit 18:* TRIG1/RAW SSTROBE Output. When high, the front panel Trigger ID bit 1 is driven true. The RS-485 SSTROBE output (RS-485 Output Port) is also driven true if the RS-485 Output Controller is disabled.
*Bit 19:* TRIG2/RAW WAIT Output. When high, the front panel Trigger ID bit 2 is driven true. The RS-485 WAIT line (normally an input on the RS-485 Output Port, but driven as an output in RAW mode) is also driven true if the RS-485 Output Controller is disabled.
*Bit 20:* TRIG3/RAW FILL-EVEN Output. When high, the front panel Trigger ID bit 3 is driven true. The RS-485 FILL-EVEN Output (RS-485 Output Port) is also driven true if the RS-485 Output Controller is disabled.
*Bit 21:* RAW EOR Output. When high, the End Of Record Output (RS-485 Output Port) is driven true if the RS-485 Output Controller is disabled.
*Bit 22:* RAW 16/32* Output. When high, the 16/32* Output (RS-485 Output Port) is driven true if the RS-485 Output Controller is disabled.
*Bit 23:* RAW RSBAF Output. When high, the RSBAF Output (RS-485 Output Port) is driven true if the RS-485 Output Controller is disabled.
*Bit 24:* RAW SPARE1 Output. When high, this output (RS-485 Output Port) is driven true if the RS-485 Output Controller is disabled.

### *4.1.2.10 Mode Register Read (F1 A1)*

| 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 13 | 12 9 | 8 5 | 4 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RAW SPARE1 OUT | RAW RSBAF OUT | RAW 16/32* OUT | RAW EOR OUT | TRIG3 /RAW FEVN | TRIG2 /RAW WAIT | TRIG1 /RAW SSTB | TRIG0 | NOT USED | COUNTER MODE | FERA OUTPUT CONTROLLER MODE | RS-485 OUTPUT CONTROLLER MODE | RS-485 INPUT CONTROLLER MODE |
| (EBI AE) | | | (EBI FF) | | (EBI AF) | (EBI EF) | | | | | | |

Returns the contents of the Mode Register previously written to it.

### *4.1.2.11 Status Register B Read (F1 A2)*

| 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | RSBAF IN | OUT PO | OUT PI | IN PO | IN PI | CNTR OVFL | FIFO HF IN | BUSY IN | FERA GATE IN | FERA CLEAR IN | 0 | 0 | EBI EF | EBI AE | EBI AF | EBI FF | FFIFO FF | FFIFO HF | FFIFO EF | RFIFO FF | RFIFO HF | RFIFO EF |

The state of several miscellaneous status bits in the CRABS module are accessible in Status Register B. Bits are defined as follows:
*Bit 1:* RFIFO_EF. Empty flag of the RS-485 FIFO. 1 = Not empty, 0 = Empty.
*Bit 2:* RFIFO_HF. Half full flag of the RS-485 FIFO. 1 = FIFO is less than half full, 0 = FIFO is greater than or exactly half full.
*Bit 3:* RFIFO_FF. Full flag of the RS-485 FIFO. 1 = Not full, 0 = Full.
*Bit 4:* FFIFO_EF. Empty flag of the FERA FIFO. 1 = Not empty, 0 = Empty.
*Bit 5:* FFIFO_HF. Half full flag of the FERA FIFO. 1 = FIFO is less than half full, 0 = FIFO is greater than of exactly half full.
*Bit 6:* FFIFO_FF. Full flag of the FERA FIFO. 1 = Not full, 0 = full.
*Bits 7-10:* EBI “gas gauge” bits as received on the RS-485 Input Port.
*Bit 11,12:* Unused status bits. Always read as zeros.
*Bit 13:* FERA CLEAR Input. Reflects the status of the CLEAR input on the FERA Output Port. 1 = active.
*Bit 14:* FERA GATE Input. Reflects the status of the GATE input on the FERA Output Port.
*Bit 15:* BUSY Input. Reflects the status of the front panel BUSY Input. 1 = active.
*Bit 16:* FIFO HF Input. Reflects the status of the front panel FIFO HF Input. 1 = active.
*Bit 17:* Counter Overflow. Overflow flag of the programmable counter. 1 = overflow. Overflow for the Programmable Counter occurs when the count is incremented from 4,194,303 to 0.
*Bit 18:* Input Permit In. When low, Permit In on the RS-485 Input Port is active. 1 = active

*Bit 19:* Input Permit Out. When low, Permit Out on the RS-485 Input Port is active. 1 = active
*Bit 20:* Output Permit In. When low, Permit In on the RS-485 Output Port is active. 1 = active
*Bit 21:* Output Permit Out. When low, Permit Out on the RS-485 Output Port is active. 1 = active
*Bit 22:* RSBAF Input. Reflects the status of the RSBAF Input on the RS-485 Output Port. 1 = active.
*Bit 23,24:* Unused status bits. Always read as zeros.

### *4.1.2.12 Status Register C Read (F1 A3)*

| 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | NO OUT TERM | NO IN TERM | RAW SPARE IN | RAW RSBAF IN | RAW 16/32* IN | RAW EOR IN | RAW FEVN IN | RAW WAIT IN | RAW SSTB IN | RAW STRB IN |

The state of the eight non-data RS-485 RAW Mode inputs on the RS-485 Input port can be read in Status Register C. Bits are defined as follows:

*Bit 1:* RAW STROBE Input. Reflects the state of the STROBE input on the RS-485 Input Port.
*Bit 2:* RAW SSTROBE Input. Reflects the state of the SSTROBE input on the RS-485 Input Port.
*Bit 3:* RAW WAIT Input. Reflects the state of the WAIT input on the RS-485 Input Port. WAIT is driven as an output on this port when the Input Port Controller is enabled, but its output driver is disabled when the Input Controller is disabled.
*Bit 4:* RAW FILL-EVEN Input. Reflects the state of the FILL-EVEN input on the RS-485 Input Port.
*Bit 5:* RAW EOR Input. Reflects the state of the EOR input on the RS-485 Input Port.
*Bit 6:* RAW 16/32* Input. Reflects the state of the 16/32* input on the RS-485 Input Port.
*Bit 7:* RAW RSBAF Input. Reflects the state of the RSBAF input on the RS-485 Input Port.
*Bit 8:* RAW SPARE Input. Reflects the state of the SPARE input on the RS-485 Input Port.
*Bit 9:* When read as a zero, this bit indicates that the termination SIP resistor networks are installed on the RS-485 input port. A one read here indicates that the resistor networks are not present.
*Bit 10:* When read as a zero, this bit indicates that the termination SIP resistor networks are installed on the RS-485 output port. A one read here indicates that the resistor networks are not present.
*Bits 11-24:* These bits are unused, and are always read as zeros.

# 5. Required Hardware

Since the CRABS is a CAMAC slave module, a CAMAC crate with power supply, and a crate controller is necessary.

# 6. Required Software

Software for the test stand will be written in C, using the SPUDS diagnostic environment. The purpose of the tests will be to test production quantities of DYC3 modules to verify proper function, and to provide scope looping tools to aid in repair. Software will be divided into five modules.

## 6.1 Test Stand Configuration and Self Test.

All components of the system are configured as necessary, and test stand confidence tests are run. Depending on the hardware configuration chosen, as much of the test hardware that can be tested without the DYC3 module is tested. Only Solution D is completely self-testable.

## 6.2 Push Data Test

The DYC3 is tested in DART mode, by inputting FERA data to the DYC3, and then collecting the RS-485 data from the DYC3 and comparing it to the data sent. FIFO status flags are tested for correct timing (within one data word resolution).

## 6.3 Pull Data Test

The DYC3 is testing in EBI mode, by inputting FERA data to the DYC3, then reading the data out over the RS-485 link and comparing. FIFO status flags are tested for correct timing (within one data word resolution).

### 6.4 Miscellaneous I/O Test

Token passing, event ID, Busy generation, and FERA control signals are tested.

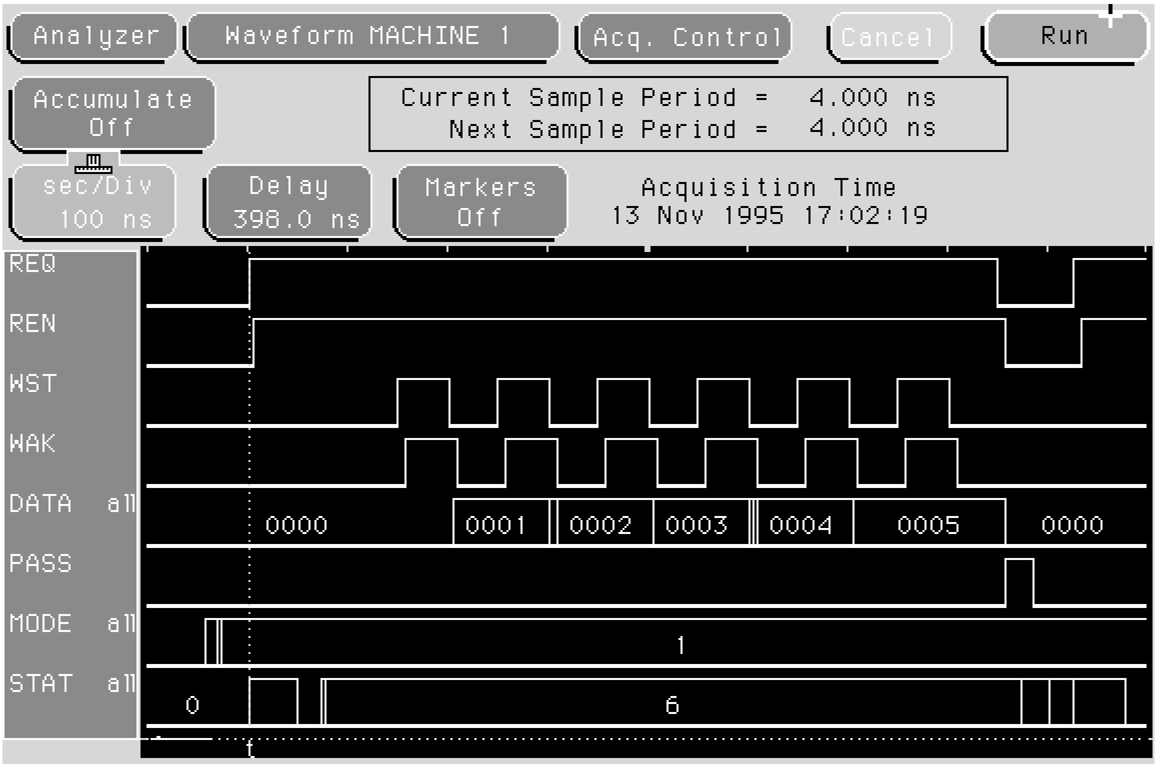
### 6.5 Scope Loop Utility

A simple menu driven scope loop builder program to generate repetitive patterns for oscilloscope probing of the DYC3. The utility will provide programmability of data patterns, and loop counts.

## 7. Data Flow Timing Details

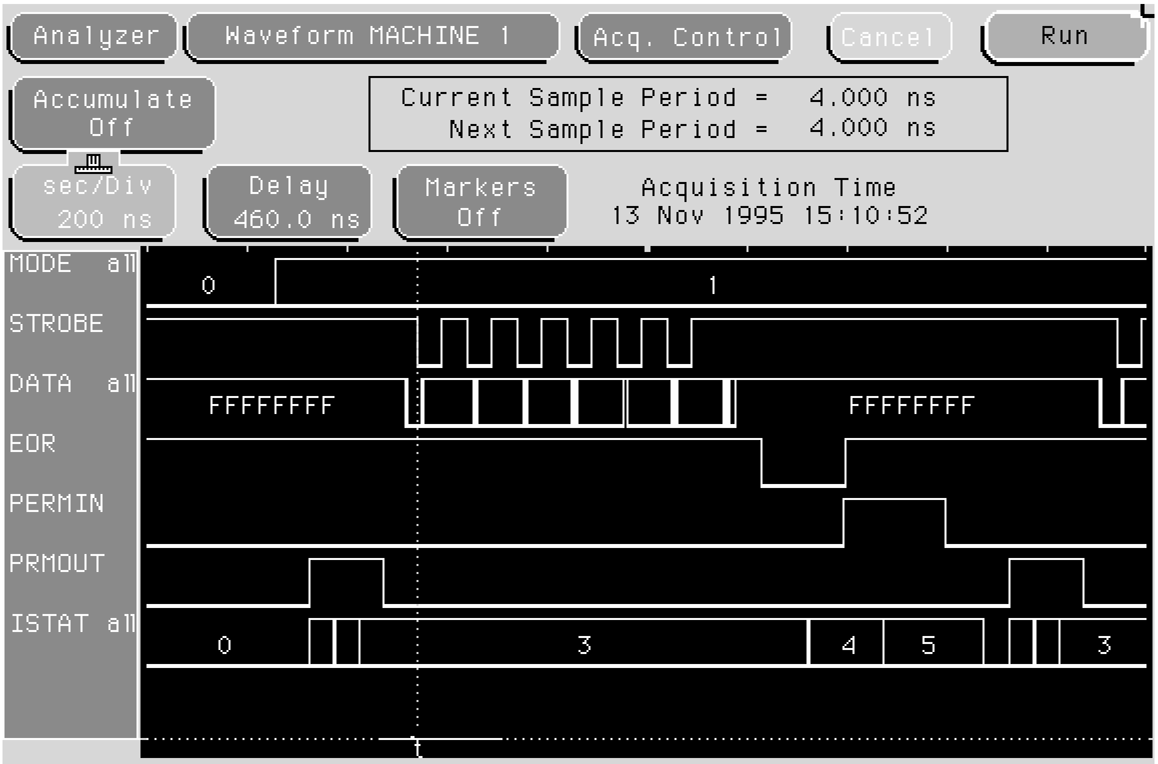
To test DYC3 modules, the desired data pattern is written into the CRABS Output FIFO over CAMAC. The FERA FIFO appears to CAMAC as one 24-bit sub-address location. The lowest 16-bits are defined as the FERA data bits, and bits 17 and 18 are control bits as defined in the following section. Bits 19 through 24 are undefined, and have no effect on module operation. The FERA FIFO is 18-bits wide by 8-K words deep. To simulate a FERA readout, the CRABS issues a REQ over the FERA Command bus. The DYC responds by asserting the REN control line. The CRABS then begins clocking data out of the Output FIFO, and transmitting it over the ECL output port. The CRABS then issues a Permit Out signal to the DYC3. The DYC3 responds by outputting the data it received through its FERA port, over it's RS-485 bus. This data is clocked into the CRABS RS-485 FIFO. The RS-485 FIFO is 32-bits wide by 8 K long words deep. The data in this FIFO is then read out over CAMAC. The RS-485 FIFO appears as two 24-bit CAMAC sub-address locations. Each location contains 16-bits of the FIFO's data in the lowest 16 bits. One of the locations contains the lowest order 16-bits of the RS-485 data, and one of the locations contains the highest order 16-bits of the RS-485 data.

# 8. Control Flow Details




***CRABS FERA Output Port Timing***

The CRABS module's Output Port, emulates a FERA (LeCroy(tm) Model 4300 CAMAC Fast Encoding and Readout ADC) module's front panel ECL output port and Command Bus. Control signal timing for the CRABS Output Port is intended to be as similar to the FERA module's as possible. In general, the Command Bus contains all of the control signals for the ECL Output Port which contains only data. Since FERA modules are intended to be chained together on the same bus, the CRABS must emulate any control signal timing which may occur due to switching of control of the ECL Port Bus from one FERA module to another. When the CRABS FERA port is enabled by setting the output mode in the mode register, it drives the REQ (Request) output true. The DYC3 should then respond by driving the REN (Readout Enable) line true. The CRABS then clocks the first word out of its FERA FIFO, and drives the WST (Write Strobe) line true. The DYC3 should respond to the WST by driving the WAK (Write Acknowledge) line true. The CRABS then negates the WST line, and clocks the next word out of its FIFO. The CRABS continues in this fashion until it either empties the FIFO, or the PASS flag (17th bit in the output FIFO) is clocked out of the FIFO. This flag allows the FIFO to be loaded with several simulated events worth of data separated by PASS flags. The CRABS FERA sequencer will drop the REQ line, and generate a PASS output whenever it sees the PASS bit true coming out of the Output FIFO.

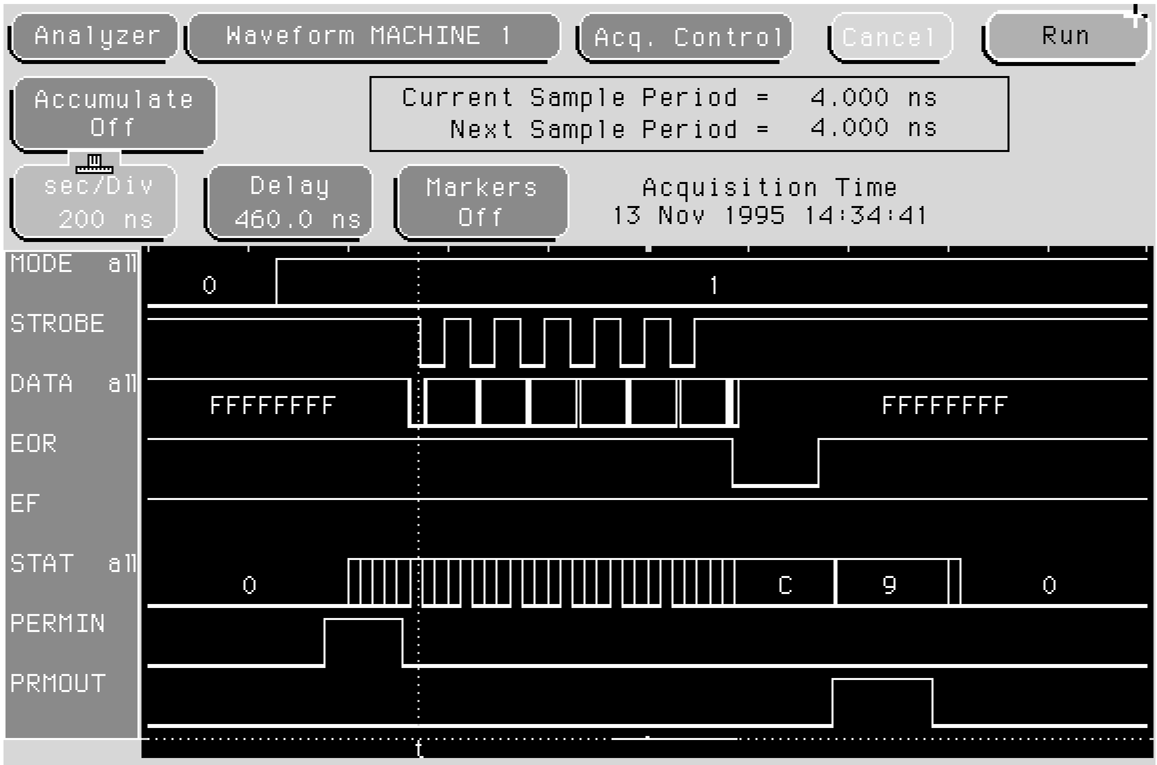



***CRABS RS-485 Input Port Timing in DART Mode***

*(Permits shown inverted)*

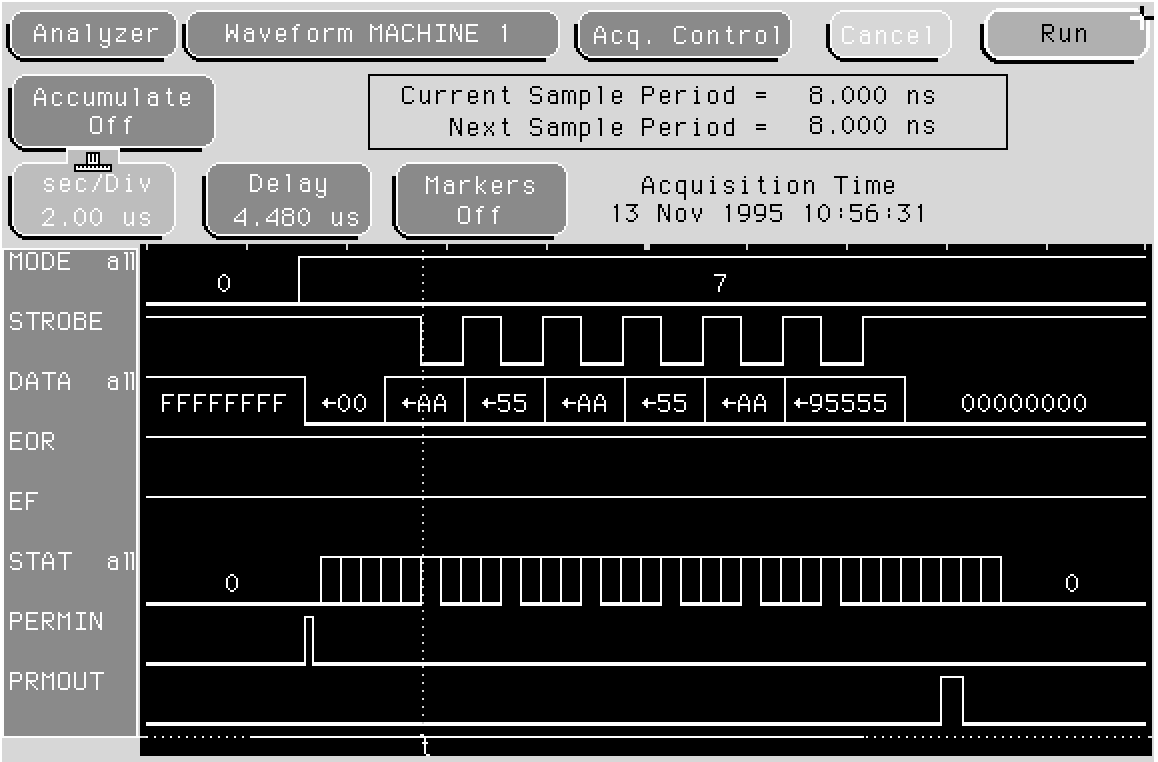
The 32-bit RS-485 input port of the CRABS can be software configured to emulates either a DART system input buffer, or an EBI. The timing of the Input PORT is, therefore, intended to be as close to the timing of these two devices as possible. The port can also be configured to test DART Fiber modules in RAW mode.

In DART mode, the CRABS RS-485 Input Port issues a Permit Out pulse, which should be interpreted by the DYC3 as the signal to start outputting data. Data on the RS-485 Data lines is clocked into the CRABS RS-485 FIFO by the RS-485 Strobe pulse. When the CRABS input FIFO's become half full, it drives the RS-485 Wait line true. The DYC3 should then pause its output.




***CRABS RS-485 Output Port Timing in DART Mode***

*(Permits shown inverted)*

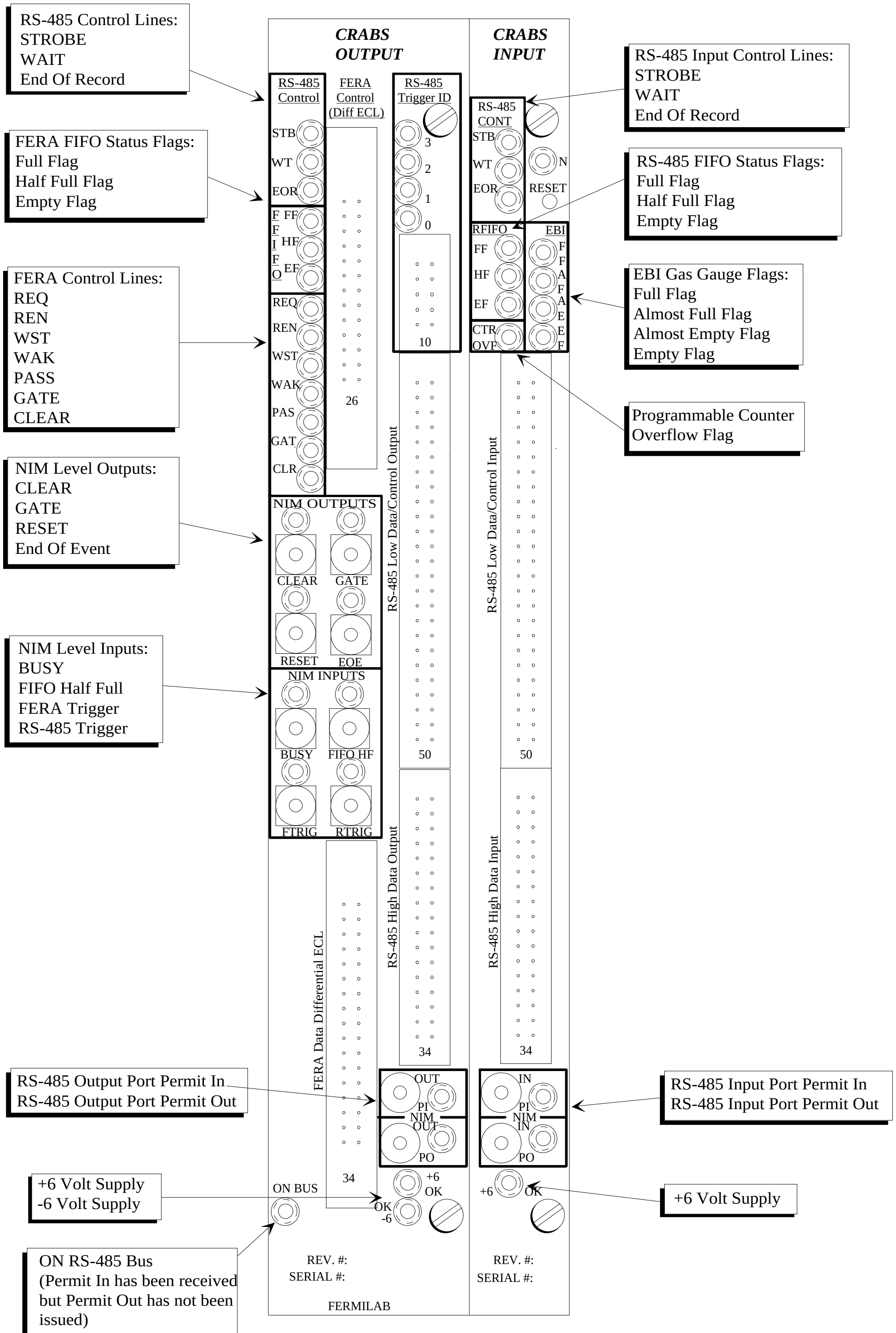



***CRABS RS-485 Output Port Timing in RAW Mode***
*(Permits shown inverted)*

In RAW mode, the RS-485 Output Port outputs data at a reduced rate to ensure that signal transitions will not be missed by the DART Fiber module's RAW mode 400ns sample rate. The data are also accompanied by a Strobe pulse as in DART mode. DART Fiber Optic modules do not need the Strobe pulse to pass data in RAW mode, but the CRABS RS-485 Input Port does. The Fiber modules pass the Strobe similarly to any other data bit in RAW mode.

# 9. Links and Data Paths

## 9.1 CRABS Front Panel

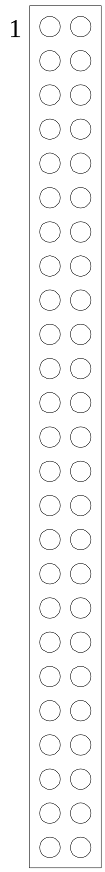



CRABS front panel 8-31-94 Kwarciany

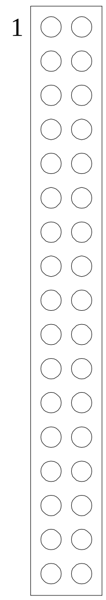
## 9.2 RS-485 Input Port

The 32-bit RS-485 Input Port of the CRABS is designed to accept data from both the EBI and DART types of DYC3 modules. It is, however, equipped only with connectors matching those of the DART system's DDD. Connection to the EBI style DYC3 can be made using the QUADRUPOD adapter.

### 9.2.1 DART System RS-485 Connectors




50-pin DART Connector




34-pin DART Connector

### *50 pin Connector Pinout*

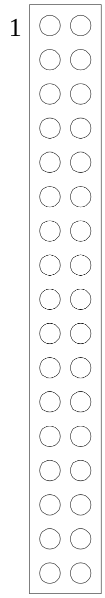
| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | +DATA 0 | 2 | -DATA 0 |
| 3 | +DATA 1 | 4 | -DATA 1 |
| 5 | +DATA 2 | 6 | -DATA 2 |
| 7 | +DATA 3 | 8 | -DATA 3 |
| 9 | +DATA 4 | 10 | -DATA 4 |
| 11 | +DATA 5 | 12 | -DATA 5 |
| 13 | +DATA 6 | 14 | -DATA 6 |
| 15 | +DATA 7 | 16 | -DATA 7 |
| 17 | +DATA 8 | 18 | -DATA 8 |
| 19 | +DATA 9 | 20 | -DATA 9 |
| 21 | +DATA 10 | 22 | -DATA 10 |
| 23 | +DATA 11 | 24 | -DATA 11 |
| 25 | +DATA 12 | 26 | -DATA 12 |
| 27 | +DATA 13 | 28 | -DATA 13 |
| 29 | +DATA 14 | 30 | -DATA 14 |
| 31 | +DATA 15 | 32 | -DATA 15 |
| 33 | +DATA STROBE | 34 | -DATA STROBE |
| 35 | +DART_SSTROBE/EBI_EF | 36 | -DART_SSTROBE/EBI_EF |
| 37 | +DART_WAIT*/EBI_AF | 38 | -DART_WAIT*/EBI_AF |
| 39 | +FEVEN | 40 | -FEVEN |
| 41 | +DART_EOR*/EBI_FF | 42 | -DART_EOR*/EBI_FF |
| 43 | +16/32 | 44 | -16/32 |
| 45 | +RSBAF | 46 | -RSBAF |
| 47 | +DART_SPARE/EBI_AE | 48 | -DART_SPARE/EBI_AE |
| 49 | N/C | 50 | N/C |

### *34 pin Connector Pinout*

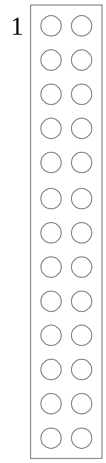
| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | +DATA 16 | 2 | -DATA 16 |
| 3 | +DATA 17 | 4 | -DATA 17 |
| 5 | +DATA 18 | 6 | -DATA 18 |
| 7 | +DATA 19 | 8 | -DATA 19 |
| 9 | +DATA 20 | 10 | -DATA 20 |
| 11 | +DATA 21 | 12 | -DATA 21 |
| 13 | +DATA 22 | 14 | -DATA 22 |
| 15 | +DATA 23 | 16 | -DATA 23 |
| 17 | +DATA 24 | 18 | -DATA 24 |
| 19 | +DATA 25 | 20 | -DATA 25 |
| 21 | +DATA 26 | 22 | -DATA 26 |
| 23 | +DATA 27 | 24 | -DATA 27 |
| 25 | +DATA 28 | 26 | -DATA 28 |
| 27 | +DATA 29 | 28 | -DATA 29 |
| 29 | +DATA 30 | 30 | -DATA 30 |
| 31 | +DATA 31 | 32 | -DATA 31 |
| 33 | N/C | 34 | N/C |

## 9.3 FERA Output Port

All FERA signals are differential ECL. Connectors are defined as follows.




34-pin FERA
ECL Port Connector




26-pin FERA
Command Bus
Connector

***34-pin FERA ECL Port Connector Pinout***

| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | +DATA 0 | 2 | -DATA 0 |
| 3 | +DATA 1 | 4 | -DATA 1 |
| 5 | +DATA 2 | 6 | -DATA 2 |
| 7 | +DATA 3 | 8 | -DATA 3 |
| 9 | +DATA 4 | 10 | -DATA 4 |
| 11 | +DATA 5 | 12 | -DATA 5 |
| 13 | +DATA 6 | 14 | -DATA 6 |
| 15 | +DATA 7 | 16 | -DATA 7 |
| 17 | +DATA 8 | 18 | -DATA 8 |
| 19 | +DATA 9 | 20 | -DATA 9 |
| 21 | +DATA 10 | 22 | -DATA 10 |
| 23 | +DATA 11 | 24 | -DATA 11 |
| 25 | +DATA 12 | 26 | -DATA 12 |
| 27 | +DATA 13 | 28 | -DATA 13 |
| 29 | +DATA 14 | 30 | -DATA 14 |
| 31 | +DATA 15 | 32 | -DATA 15 |
| 33 | N/C | 34 | N/C |

***26-pin FERA Command Bus Connector Pinout***

| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | N/C | 2 | N/C |
| 3 | +WST | 4 | -WST |
| 5 | +REQ | 6 | -REQ |
| 7 | +CLR | 8 | -CLR |
| 9 | +GATE | 10 | -GATE |
| 11 | +WAK | 12 | -WAK |
| 13 | N/C | 14 | N/C |
| 15 | N/C | 16 | N/C |
| 17 | N/C | 18 | N/C |
| 19 | N/C | 20 | N/C |
| 21 | N/C | 22 | N/C |
| 23 | +REN | 24 | -REN |
| 25 | +PASS | 26 | -PASS |

## 9.4 Input Permit Out (IN PO) Output

Front panel LEMO type connector. NIM level, single ended output. The purpose of this output is to allow testing of data sources on the RS-485 cable. The IN PO output is connected to the Permit In input of the tested device. When the RS-485 Input Port is initialized, the CRABS module then drives this 150-250ns pulse to trigger the data source to start sending data.

## 9.5 Output Permit Out (OUT PO) Output

Front panel LEMO type connector. NIM level, single ended output. This output allows for token passing when the CRABS module is used in data source daisy chaining tests. The OUT PO and OUT PI connectors have the equivalent function to the Permit Out and Permit In connectors on DART style data sources.

## 9.6 Clear Output

Front panel LEMO type connector. NIM level output. This output is driven to a NIM 1 by setting the CLEAR bit in the control register. It is driven to a NIM 0 by clearing the CLEAR bit.

## 9.7 Gate Output

Front panel LEMO type connector. NIM level output. This output is driven to a NIM 1 by setting the GATE bit in the control register. It is driven to a NIM 0 by clearing the GATE bit.

## 9.8 Reset Output

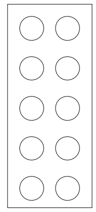
Front panel LEMO type connector. NIM level output. This output is driven to a NIM 1 by setting the RESET bit in the control register. It is driven to a NIM 0 by clearing the RESET bit.

## 9.9 EOE Output

Front panel LEMO type connector. NIM level output. This output is driven to a NIM 1 by setting the EOE bit in the control register. The output is driven to a NIM 0 by clearing the EOE bit.

## 9.10 Event ID Output

Four bits of differential RS-485. These four bits are set by setting the corresponding bits in the mode register.

1


10-pin Trigger ID
Connector

| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | +TRIG0 | 2 | -TRIG0 |
| 3 | +TRIG1 | 4 | -TRIG1 |
| 5 | +TRIG2 | 6 | -TRIG2 |
| 7 | +TRIG3 | 8 | -TRIG3 |
| 9 | +TSTRB | 10 | -TSTRB |

## 9.11 Input Permit In (IN PI) Input

Front panel LEMO type connector. NIM level, single ended input. This input is provided to allow testing of data sources on the RS-485 cable. The IN PI input is connected to the Permit Out of the tested device. If Permits are enabled (by setting the appropriate mode), the state machine will expect to see a Permit_In active after an EOR has been received. The RS-485 Input Port Controller's status lines can be examined to see if the tested device has passed the token out or not.

## 9.12 Output Permit In (OUT PI) Input

Front panel LEMO type connector. NIM level, single ended input. This input allows for token passing when the CRABS module is used in data source daisy chaining tests. The OUT PO and OUT PI connectors have the equivalent function to the Permit Out and Permit In connectors on DART style data sources. This input is edge triggered by a low going edge. The state of the internal latch of IN_PI is cleared by state machine state 1, Reset, and setting the RS-485 Output controller mode to zero.

## 9.13 Busy Input

Front panel NIM level LEMO connector. This input has no effect on the function of the CRABS module, but its state can be read in the Status B Register. A NIM 1 will cause the BUSY bit in Status B to be a 1, and a NIM 0 will cause the BUSY bit in the Status B register to be a 0. This input is provided specifically for the testing of DYC3 modules.

## 9.14 FIFO HF/AF Input

Front panel LEMO type connector with NIM level input. This input has no effect on the function of the CRABS module, but its state can be read in the Status B Register. A NIM 1 will cause the FIFO HF/AF bit in Status B to be a 1, and a NIM 0 will cause the FIFO HF/AF bit in the Status B register to be a 0. This input is provided specifically for the testing of DYC3 modules.

## 9.15 FERA Trigger (FTRIG) Input

Front panel NIM level LEMO connector. This input can be used to trigger the FERA Output Port Controller to start a FERA data block transmission. It is edge triggered by a low going edge (NIM 0 to 1 transition). The state of the internal latch of FTRIG is cleared by FERA Output Port state machine state 2, Reset, and setting the FERA Output Port controller mode to zero. The FERA Output Port Controller must be initialized to the proper mode to enable this input.

## 9.16 RS-485 Trigger (RTRIG) Input

Front panel NIM level LEMO connector. This input can be used to trigger the RS-485 Output Port Controller to start an RS-485 data block transmission. It is edge triggered by a low going edge (NIM 0 to 1 transition). The state of the internal latch of RTRIG is cleared by RS-485 Output Port state machine state 1, Reset, and setting the RS-485 Output Port controller mode to zero. The RS-485 Output Port Controller must be initialized to the proper mode to enable this input.