Model 2115

PCI Serial Highway Driver

INSTRUCTION MANUAL

February 5, 1999

(C) 1996,1997,1998, 1999
Copyright by
KineticSystems Company, LLC
Lockport, Illinois
All rights reserved

NOTICE

The 2x15 device may have trouble loading in some Pentium based computers.

If you experience trouble loading the KSC device driver:

1. Reboot
2. Enter the CMOS setup program
3. Enable the option asking if you are using a P&P operating system
4. Reboot the machine
5. Load the driver

This should take care of any loading problems.

KSC Support Staff

CONTENTS

PCI Enhanced Serial Highway Driver

Allows a computer PCI bus to host a Serial Highway

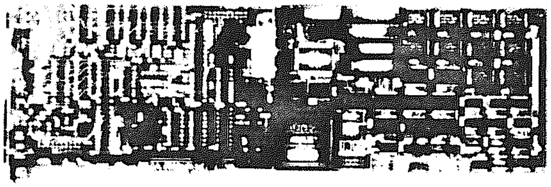
2115

Features

- **Provides a dedicated PCI interface to the CAMAC serial highway**
- **Supports the Enhanced highway mode with throughput of 3 Mbytes/s**
- **High-speed hardware list processing**
- **Capability of timer-initiated list processing for synchronous data acquisition**
- **Available with wire or fiber-optic transmission options**
- **Up to two kilometers between highway nodes at full throughput using fiber-optic transmission**
- **Supports up to 62 nodes**

Typical Applications

- **Realtime I/O interface for a personal computer or workstation**
- **Distributed data acquisition and control**
- **Systems requiring high throughput with minimum software overhead**



General Description *(Product specifications and descriptions subject to change without notice.)*

The 2115 Enhanced Serial Highway Driver (SHD) interfaces a host computer to the CAMAC Serial Highway using the Peripheral Component Interface (PCI) bus. This single card fits into any computer that can accept a full size PCI adapter card. The SHD operates the CAMAC Serial Highway at rates from 0.5 to 5 Mbytes/s in byte-serial mode.

The 2115 SHD communicates with the host through a 32-bit data path that is used to configure and initiate highway operations. A DMA (Direct Memory Access) mechanism is incorporated to transfer the data to and from the Serial Highway. The 2115 supports PCI burst transfers, yielding a maximum transfer burst rate of 15 Mbytes/s.

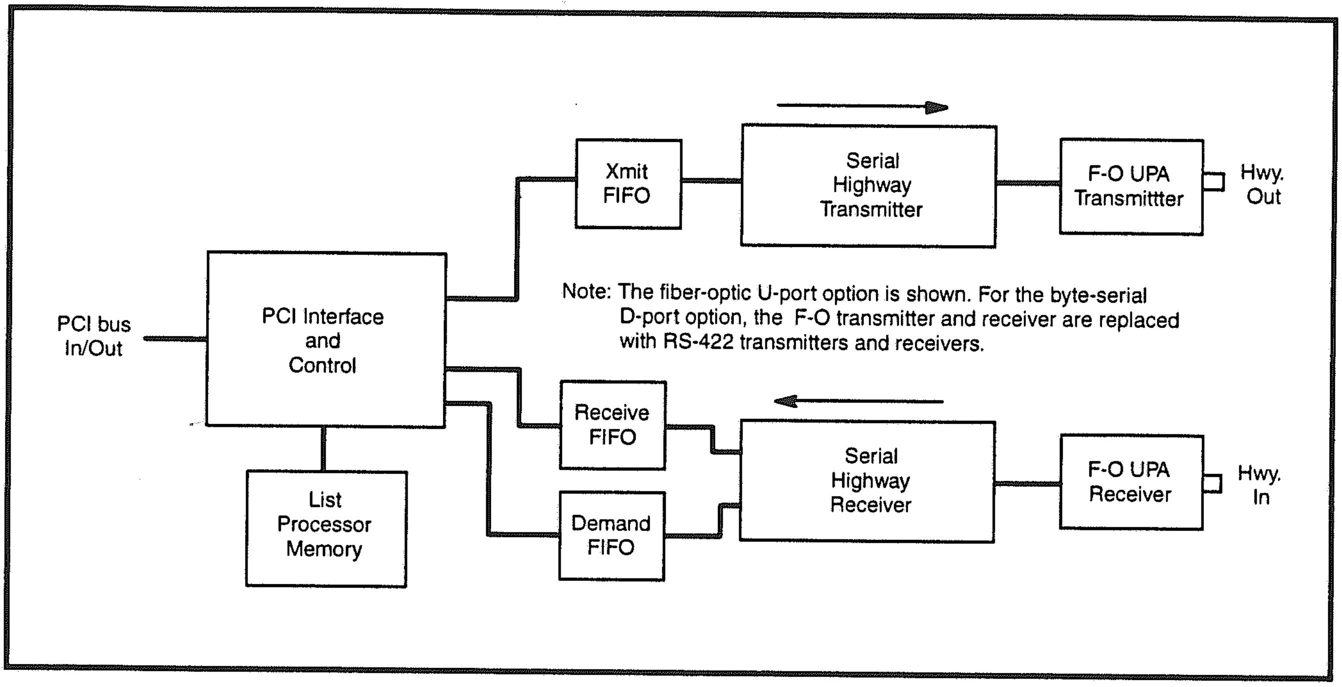
The SHD includes a list processor that allows execution of a preloaded list of highway operations. The list of operations is read from a 32K x 32 RAM memory and then executed. List processing may be initiated by software trigger or by an internal, crystal-controlled, programmable counter/timer. The timer provides for triggering rates that range from 0.06 Hz to 500 kHz in 1-microsecond increments. The 2115 uses host memory to implement multi-buffer functionality. Multi-buffer activity includes DMA data chaining and setting of buffer segment flags which in turn may generate PCI interrupts. This function allows the host computer to read a memory buffer previously filled by the 2115 while another segment of the buffer is filled.

A 2048-word FIFO is provided on the 2115 for storing asynchronous demand messages received from the highway. These messages are generated by Serial Crate Controllers (SCCs) in response to a LAM (Look-At-Me) in a chassis.

In addition to the Enhanced Mode, the 2115 includes software-selectable operating modes for Q-Ignore, Q-Stop, Q-Repeat, and Q-Scan operations, making it usable with any type L-2 SCC meeting the IEEE-595 specification. Enhanced Mode messages require usage of the 3952-Z1G or 3952-Z1H. Enhanced protocol allows "Enhanced" and "Standard" type L-2 SCCs to be mixed on the same highway, provided that the software directs enhanced block messages only to the enhanced SCCs.

The 2115-Z1A contains a D-port adapter which attaches to the serial highway by means of two 25-position D-type connectors. The 2115-Z1B connects to a serial highway that uses fiber-optic transmission. This module supports both 5 MHz and 2.5 MHz clock rates. Either version of the 2115 requires only one full-size PCI expansion slot.

Block Diagram


Xmit FIFO
Serial Highway Transmitter
F-O UPA Transmittter
Hwy. Out
PCI bus In/Out
PCI Interface and Control
Note: The fiber-optic U-port option is shown. For the byte-serial D-port option, the F-O transmitter and receiver are replaced with RS-422 transmitters and receivers.
Receive FIFO
Serial Highway Receiver
F-O UPA Receiver
Hwy. In
List Processor Memory
Demand FIFO

Ordering Information

Model 2115-Z1A	Enhanced Serial Highway Driver with D-port, for PCI host
Model 2115-Z1B	Enhanced Serial Highway Driver with 820 nm U-port, for PCI host

Associated Products

Model 3830-Z1A	List Sequencer for 3952 Serial Crate Controllers
Model 3939-Z2A	U-port Adapter, byte-serial, fiberoptic, 820 nm operation
Model 3952-Z1G	Enhanced Serial Crate Controller, type L-2, with standard relays
Model 3952-Z1H	Enhanced Serial Crate Controller, type L-2, with high-gauss relays
Model AD4Z-NPA1	Software, Device Driver, Windows NT
Model AD4Z-VAA1	Software, Device Driver, Open VMS/Alpha AXP

Introduction

The 2115 PCI Serial Highway Driver (SHD) is a full size PCI card which interfaces the PCI bus to the Serial Highway, IEEE Standard 595. This SHD operates the Serial Highway in byte mode at either 5 MHz, 2.5 MHz, 1 MHz or 500 kHz. The 2115 supports up to 62 remote crates for distributed data acquisition and control. This driver includes a block-data enhancement to the Serial Highway developed by KineticSystems, providing an effective data throughput rate of 3 Megabytes-per-second using the 5 MHz Serial Highway rate.

The CAMAC Serial Highway protocol, as defined by IEEE Standard 595, provides for one CAMAC 24-bit Dataway operation per Serial Highway command message. The highway block-mode enhancement contained in the 2115 allows multiple Dataway operations to occur within one highway message. These enhanced messages that use a pipelined approach for data transfers yield a data throughput that is actually five times as high as previous SHDs.

The 2115 executes highway operations provided by a list of instructions preloaded into a 32K x 32 list memory. This memory is loaded prior to initiating operations by programmed I/O transfers or by a DMA operation. Once list processing is initiated, data transfers to/from the 2115 may occur using either DMA or programmed transfers. DMA operations to/from the 2115 are executed using a 32-bit word format.

The list may contain such instructions as Single Operations, Block Transfer Operations, Single Inline Write Operations, Load Memory Address Register, Load Total Transfer Count Register and a Jump instruction. These instructions allow for a very versatile list. The format of the instruction varies depending on the type of instruction and are either one or two 32-bit words in length.

A Demand FIFO is provided to retain up to 2048 demand messages. These messages are generated by Serial Crate Controllers (SCCs) when a Look-At-Me (LAM) is asserted in a CAMAC chassis. When these messages are received by the 2115, the chassis address and a Serial Graded LAM (SGL) byte are stored in the FIFO. Optionally, the receipt of a demand message may assert an interrupt request to the PCI bus.

Installation

The Model 2115 is designed to fit into any full size PCI expansion slot. Before the 2115 is inserted into the computer backplane, several strap options must be configured. The following section shows the various strap select options and a description of changing the default values.

Before the installation is initiated, turn off the power to the computer and remove the power cord. Remove the cover to the computer and locate an empty expansion slot. Remove the blank plate from the mounting rail of the selected slot. Insert the 2115 into the slot and secure the mounting plate with the screw that was removed from the blank plate. Replace the cover on the computer and then plug the power cord back into the unit.

After the card is installed in the computer, connect the highway cables to the 2115. For the 2115-Z1A, the highway connections are made via two 25-pin D-type connectors on the 2115. The connector with the pins is the Defined Port In (DIN) to the driver and the socket connector is the Defined Port Out (DOUT) of the driver. The DIN connection is location P3 on the main card and P2 is the DOUT location. For the 2115-Z1B, the top connector is the Undefined Port In (UIN) to the driver and the bottom connector is the Undefined Port Out (UOUT) of the driver. The highway output connection of the 2115 connects to the highway input connection of the first SCC unit on the highway. The highway input connector of the 2115 connects to the highway output connection of the last SCC on the highway.

Strap Selections

The following chart shows the default values for each strap selection as set from the factory.

Strap Option	Default Configuration	Strap Designator
Serial Highway Clock Rate	5 Megahertz	STP7,STP8
Extended Space Bytes	Disabled	STP6
Q-Repeat Timeout	Enabled	STP4
Q-Repeat Timeout Value	15 Seconds	STP5,STP3,STP2
D-Port/U-Port	-Z1A D-port -Z1B U-Port	STP9,STP10, STP11

The following sections details information regarding the strap selections.

Serial Highway Clock Rate

The 2115 can operate the Serial Highway at either 5, 2.5, 1 or .5 Megahertz. This selection controls the rate at which the Serial Highway clock operates for byte-serial transfers. The 2115 does not support bit-serial operations.

Two straps located on the 2115 control the highway clock rate. The straps used to configure the clock rate are labeled STP7 and STP8. The binary combination of these two straps determine the clock rate as shown in the following chart. Appendix A of this manual shows the locations of these straps on the 2115.

Serial Highway Clock Rate	STP7 Position	STP8 Position
5 Megahertz	Right	Right
2.5 Megahertz	Right	Left
1 Megahertz	Left	Right
500 Kilohertz	Left	Left

The indications of LEFT and RIGHT in the above chart refer to the position of the strap jumper for the indicated strap. The orientation of the card is with the PCI connector facing toward the user when making the selections of right and left.

Extended Space Bytes

The 2115 normally inserts the minimum number of required space bytes during Serial Highway operations. Space bytes are generated by the SHD to allow the addressed Serial Crate Controller (SCC) to generate its reply message in place of some or all of the space bytes. Some SCC's require additional space bytes be generated by the SHD in order to accommodate the reply message. The 2115 provides a mechanism to insert a minimum of 16 space bytes for each command transmitted. The KineticSystems' SCCs do not require extended space bytes, the minimum number suffices for all messages.

The strap STP6 is used for enabling and disabling the extended space bytes. Placing the strap in the left-most position disables the extra space bytes and the right-most position enables the extra space bytes.

Q-Repeat Timeout Selection

The 2115 contains a strap selectable timer to control the timeout value for Q-Repeat transfers. If the 2115 transmits a message on the highway and does not receive a reply message within the selected timeout period, an error is generated and the highway is reset. This timeout value ranges from 3 seconds to 15 seconds. The timeout may also be disabled to allow for an infinite timeout value.

Please refer to Appendix A of this manual for the location and settings of the Q-Repeat Timeout. The following chart shows the various timeout values along with the strap number that selects the desired timeout.

Q-Repeat Timeout	STP4	STP5	STP2	STP3
Disabled	I	R	R	R
3 Seconds	R	I	R	R
7 Seconds	R	R	I	R
15 Seconds	R	R	R	I

For the above chart, I indicates a strap jumper installed and an R refers to a strap jumper removed. Note that only one strap jumper can be used for these selections.

D-Port / U-Port Options

The 2115 is shipped from the factory with the D-Port/U-port option straps set in the appropriate location for the option ordered. The 2115-Z1A option is configured for D-Port operation and the 2115-Z1B is configured for U-Port operation. Three straps on the 2115 are used to configure the 2115 for the various options. These straps are labeled STP9, STP10, and STP11. The following chart shows the position of the straps for the indicated 2115 option. Note that an R indicates a strap jumper connected to the rightmost two pins and an L indicates a strap jumper connected to the leftmost two pins.

2115 Option	Port Type	STP9	STP10	STP11
-Z1A	D-Port	L	R	R
-Z1B	U-Port	R	L	L

PCI Configuration Space

The PCI Specification mandates a 64-byte Configuration Header that describes the requirements of add-in cards. The data contained in this region uniquely identifies the device and allows for generic control of the device. The configuration data indicates the memory requirements of the device along with other device specific information.

This section describes the 64 bytes of configuration space implemented by the 2115. The following diagram is a composite chart showing the configuration header.

31	16	15	00	
Device Identification		Vendor Identification		00
Status		Command		04
Class Code			Revision	08
BIST	Header Type	Latency	Cache Size	0C
Base Address Register #1				10
Base Address Register #2				14
Base Address Register #3				18
Base Address Register #4				1C
Base Address Register #5				20
Base Address Register #6				24
Reserved				28
Reserved				2C
Expansion ROM Base Address				30
Reserved				34
Reserved				38
Maximum Latency	Minimum Latency	Interrupt Pin	Interrupt Line	3C

VENDOR IDENTIFICATION Field

The VENDOR IDENTIFICATION field contains read-only bits which identify the manufacturer of the device. The ID assigned to KineticSystems is 11F4 Hex.

DEVICE IDENTIFICATION Field

The DEVICE IDENTIFICATION field contains read-only bits which identify a particular device. The DEVICE ID field for this unit is 2115 Hex.

PCI COMMAND Register

The COMMAND field contains write/read bits used to configure basic PCI functions. The following diagram shows the COMMAND field as implemented by the 2115.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	FST BTB	SER ENA	0	PER ENA	0	0	0	MAS ENA	MEM ENA	I/O ENA

<15:10> Not Used. These bits are not used and read as zeros.

<9> FAST BACK-TO-BACK is a write/read bit used to enable or disable the 2115 from executing "fast" back-to-back bus master cycles after completing a write cycle. Setting this bit to a one enables fast transfers and a zero disables the mode.

<8> SYSTEM ERROR ENABLE is a write/read bit used to enable and disable the 2115 from driving the PCI SYSTEM ERROR (SERR) signal. This signal is used by the 2115 to inform the host CPU of a parity error during an address or control portion of a bus operation. Setting this bit to a one enables the 2115 to assert SERR and disabled with a zero.

<7> Not Used. These bits are not used and read as zeros.

<6> PARITY ERROR ENABLE is a write/read bit used to enable and disable the 2115 from driving the PCI PARITY ERROR (PERR) signal. This signal is asserted by the 2115 when a parity error is detected during a data transfer to/from the 2115. The PERR function is enabled by setting this bit to a one and disabled with a zero.

<5:3> Not Used. These bits are not used and read as zeros.

<2> BUS MASTER ENABLE is a write/read bit which enables and disables the 2115 from executing bus master operations. Setting this bit to a one enables the 2115 to function as a bus master and a zero disables the master operation.

<1> MEMORY SPACE ENABLE is a write/read bit that allows the 2115 to function in memory regions that may be defined in one of the base address registers. Since the 2115 is initially configured as an I/O device, this bit should be set to zero.

<0> I/O SPACE ENABLE is a write/read bit that allows the 2115 to function in I/O regions as defined in one of the base address registers. This bit is set to a one which allows the 2115 to function in I/O regions.

PCI STATUS Register

The PCI STATUS Register is used to record status information regarding PCI bus transfers. This register contains read-only bits and write/read bits. The following diagram shows the Status Register bits implemented by the 2115.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PAR DET	SIG SER	RCV MAB	RCV TAB	SIG TAB	DEV TM1	DEV TM0	DP RPT	1	0	0	0	0	0	0	0

<15> DETECTED PARITY ERROR is a write/read bit which is set whenever the 2115 detects a PCI parity error. This bit does not depend on the state of the PAR ENA bit in the PCI COMMAND Register. Once an error has been detected ,this may be cleared by writing a one to bit position 15.

<14> SIGNALED SYSTEM ERROR is a write/read bit which is set whenever the 2115 asserts the PCI SYSTEM ERROR (SERR) signal. Once this bit is set, it may be cleared by writing a one to bit position 14.

<13> RECEIVED MASTER ABORT is a write/read bit which is set when the 2115 is accessed as a target and the master aborts the transaction. This bit can be reset by writing a one to this bit position.

<12> RECEIVED TARGET ABORT is a write/read bit which is set when the 2115, acting as a bus master, has initiated a transfer and the addressed target aborts the transfer. This bit can be reset by writing a one to this bit position.

<11> SIGNALED TARGET ABORT is a write/read bit that is set when a bus master accesses the 2115 as a target and the 2115 aborts the cycle. This bit is reset when a one is written to this bit location.

<10:9> DEVICE SELECT TIMING 1 and 0 encode the timing of the PCI DEVSEL (Device Select) signal. This time reflects the slowest time that a device asserts DEVSEL for any bus command except Configuration Read and Configuration Write. Since the 2115 may be accesses in the fast mode, these bits are set to zero.

<8> DATA PARITY REPORTED is a write/read bit which is set when the 2115 detects a parity error when the 2115 is a bus master. This bit can be reset by writing a one to this bit position.

REVISION Field

The REVISION field contains read-only bits which reflect the current revision level of the 2115. The 2115 starts at revision one and subsequent revisions increment the number.

CLASS CODE Field

The CLASS CODE field actually contains three subfields that represent device characteristics. These three subfields are the BASE CLASS, the SUB-CLASS and the PROG I/F fields. These subfields define such parameters as network controllers, display controllers, video device, etc.

The 2115 does not fit into any of the defined class codes. Therefore, the class code that the 2115 uses is FF0000 Hex which indicates that the 2115 class code is undefined by the PCI specification.

CACHE LINE SIZE Field

The CACHE LINE SIZE field is used by the system to define the cache line size. The 2115 does not use Memory Write and Invalidate PCI bus cycles when operating as a bus master and therefore sets this field to zero.

LATENCY TIMER REGISTER

The LATENCY TIMER REGISTER is only used when the 2115 is operating as a bus master. The value loaded in this register is the minimum number of PCI bus clocks that the 2115 can be guaranteed as a master. After the 2115 becomes bus master and asserts the PCI FRAME signal, the Latency Timer is decremented for each PCI bus clock. Subsequent to the timer decrementing to zero, the 2115 ignores the PCI bus grant signal and continues to transfer data until the timer expires. The value loaded into this register is in multiples of eight clock cycles since the low 3 bits of this field are hardwired to zero. This register is loaded with a value of F8 hex at power-up. If this register is written to via application software, the value of F8 must be maintained.

HEADER Field

The HEADER field establishes whether a PCI device contains a single function or multi-function PCI bus agent. Since the 2115 contains only a single function, this field is set to zero.

BUILD IN SELF TEST Field

This field is used to present Built In Self Test diagnostic results to a bus master. The 2115 does not implement BIST and returns a zero for this field.

BASE ADDRESS REGISTERS

The BASE ADDRESS REGISTERS are used to specify the memory or I/O requirements of add-in devices and also to configure the base addresses of these devices.

After power-up, system software can determine how much address space a particular device requires by writing all ones to a base address register and then reading the value back. The device returns zeros in all address bit locations that do not define the base address.

The least significant bit in each of the base address registers is used for specifying the region of address space for which the device is to reside. A value of zero specifies a memory region and a value of one specifies an I/O region. The 2115 is configured to operate in the I/O region.

The 2115 implements two of the Base Address registers. The first Base Address register is used to communicate with the PCI Interface Operational Registers (IOR) and the second is used to communicate with the Highway Operational Registers (HOR). The Interface Operational Registers require 16 longwords (64 bytes) of address space. A read of the first base address register after a write of all ones returns the value FFFFFFC1 Hex, indicating a request for 16 longwords of I/O space.

The Highway Operational Registers require 16 longwords (64 bytes) of address space. A read of the second base address register after a write of all ones returns the value FFFFFFC1 Hex, indicating a request for 16 longwords in I/O space.

After the system software has determined the total address space requirements of the system, it assigns the base addresses to memory and I/O devices by writing their Base Address Registers.

EXPANSION ROM BASE ADDRESS Register

This field is used to assign a physical memory address to expansion ROM in a system. The 2115 does not contain an expansion ROM and therefore does not require use of this field.

INTERRUPT LINE Field

This write/read field is used for communicating interrupt routing information and is configured by the PCI BIOS after power-up. The value in this field informs the system interrupt controller which pin of the controller the interrupt is connected to.

INTERRUPT PIN Field

The INTERRUPT PIN field is read-only and specifies which PCI interrupt pin that the 2115 is connected to. The 2115 returns a value of one in this field indicating that it uses the INTA interrupt.

MINIMUM GRANT Register

This write/read register is used by bus masters to specify the minimum amount of time the device needs for a period of burst transfers. Since the 2115 does not have this requirement, a value of zero must be used.

2115 Operational Registers

The 2115 PCI Serial Highway Driver (SHD) uses several I/O addressable registers to control and monitor operations.

All data transfers to/from the 2115 must be executed using 32-bit data transfers. The 2115 does not accommodate byte or shortword accesses.

The 2115 incorporates a PCI Interface Controller manufactured by Applied Micro Circuits Corporation, the S5933. This device contains several PCI configuration registers and also some of the 2115 operational registers. The primary register that the 2115 uses is the bidirectional FIFO register. This FIFO is used in transferring data to/from CAMAC chassis on the Serial Highway. A Control/Status register provides access to FIFO flags.

PCI Interface Operational Registers

The PCI Interface Operational Registers are contained in the PCI interface chip used on the 2115. This chip is the S5933 and is manufactured by Applied Micro Circuits Corporation. The base address of these registers is loaded by power-on BIOS routines and is contained in the Base Address Register #1 location of the PCI Configuration Registers.

Note: All references to Incoming and Outgoing are referred to the host. An Outgoing operation is a write operation from the host and an Incoming operation is a read operation from the 2115.

The following chart shows the various PCI Interface Registers along with their offsets from the base address.

Offset	Register	Access	
0	Outgoing Mailbox #1	W/R	*
4	Outgoing Mailbox #2	W/R	*
8	Outgoing Mailbox #3	W/R	*
C	Outgoing Mailbox #4	W/R	*
10	Incoming Mailbox #1	R	*
14	Incoming Mailbox #2	R	*
18	Incoming Mailbox #3	R	*
1C	Incoming Mailbox #4	R	*
20	Data FIFO	W/R	
24	Master Write Address	R	
28	Master Write Transfer Count	R	
2C	Master Read Address	R	
30	Master Read Transfer Count	R	
34	Mailbox Empty/Full Status	W/R	*
38	Interrupt Control/Status	W/R	
3C	Bus Master Control/Status	W/R	

* These registers are found in the PCI Interface Controller but not used by the 2115.

Outgoing/Incoming Mailboxes

The Incoming and Outgoing Mailboxes are not used by the 2115.

Data FIFO Register

The Data FIFO Register is a write/read register located at an offset of 20 hex from the selected base address register #1 and is composed of two 8 x 32- bit FIFOs located in the PCI interface chip. When executing Serial Highway operations, all write and read data passes through these FIFOs. Data transfers to or from these FIFOs can be done by either programmed transfers executed by the host or by allowing the 2115 to become a bus master and transfer the data.

Several status indicators are provided which indicate the amount of data contained in the PCI interface chip FIFOs. The Bus Master Control/Status register contains 6 status bits that correspond to the PCI Interface chip internal FIFOs.

When executing CAMAC operations, write and read data transferred during the operation pass through this FIFO register. CAMAC data words may be either 16 or 24-bits wide. 24-Bit data words occupy one complete 32-bit FIFO word. When the selected CAMAC data word size is 16-bits, two data words are contained in each FIFO word. The following diagram shows how the CAMAC data words are packed in the FIFO register.

16-Bit CAMAC Data Word Storage

Offset 0 hex	CAMAC 16-Bit Data Word #2	CAMAC 16-Bit Data Word #1
Offset 4 hex	CAMAC 16-Bit Data Word #4	CAMAC 16-Bit Data Word #3
Offset 8 hex	CAMAC 16-Bit Data Word #6	CAMAC 16-Bit Data Word #5
Offset C hex	CAMAC 16-Bit Data Word #8	CAMAC 16-Bit Data Word #7

24-Bit Data Word Storage

Offset 0 hex	CAMAC 24-Bit Data Word #1
Offset 4 hex	CAMAC 24-Bit Data Word #2
Offset 8 hex	CAMAC 24-Bit Data Word #3
Offset C hex	CAMAC 24-Bit Data Word #4.

Mixed 16/24-Bit Data Storage

Offset 0 hex	CAMAC 16-Bit Data Word #2	CAMAC 16-Bit Data Word #1
Offset 4 hex	Reserved	CAMAC 16-Bit Data Word #3
Offset 8 hex	CAMAC 24-Bit Data Word #1	
Offset C hex	CAMAC 16-Bit Data Word #5	CAMAC 16-Bit Data Word #4

Master Write Address Register

The Master Write Address Register is a read-only register located at an offset of 24 hex from the selected base address #1 that contains the last address that was accessed by the 2115 during a bus master write operation executed by the 2115. This address is actually pointing to one longword beyond the last address accessed since the memory address is incremented after each execution of a data transfer.

The Master Write Address Register can be written by either the list processor or by programmed I/O through the Highway Operational Registers. When the Memory Address Register (MAR) of the Highway Operational Registers is written by programmed I/O or the list processor, a hardware mechanism loads both the Master Write Address Register and the Master Read Address Registers. Therefore, it is not necessary to load this register prior to a DMA operation, only the MAR in the Highway Operational Register space.

After a bus master operation completes, the current memory address minus 4 may be read from the Master Write Address Register. The initial memory address used for the bus master operation can be read through the Memory Address Register of the Highway Operational Registers.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MWAR 31	MWAR 30	MWAR 29	MWAR 28	MWAR 27	MWAR 26	MWAR 25	MWAR 24	MWAR 23	MWAR 22	MWAR 21	MWAR 20	MWAR 19	MWAR 18	MWAR 17	MWAR 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MWAR 15	MWAR 14	MWAR 13	MWAR 12	MWAR 11	MWAR 10	MWAR 09	MWAR 08	MWAR 07	MWAR 06	MWAR 05	MWAR 04	MWAR 03	MWAR 02	0	0

<31:2> MASTER WRITE ADDRESS 31 through 2 are read-only bits used to return the last address accessed during a DMA operation.

<1:0> These bits are not used since the DMA operations execute longword (32-bit) transfers.

Master Write Transfer Count Register

This register is not used by the 2115. When executing bus master operations, the 2115 uses the Total Transfer Count Register (TTCR) of the Highway Operational Registers to determine the number of bus operations to perform.

Master Read Address Register

The Master Read Address Register is a read-only register located at an offset of 2C hex from the selected base address #1 that contains the last address that was accessed by the 2115 during a bus master read operation executed by the 2115. This address is actually pointing to one longword beyond the last address accessed since the memory address is incremented after each execution of a data transfer.

The Master Read Address Register can be written by either the list processor or by programmed I/O through the Highway Operational Registers. When the Memory Address Register (MAR) of the Highway Operational Registers is written by programmed I/O or the list processor, a hardware mechanism loads both the Master Write Address Register and the Master Read Address Registers. Therefore, it is not necessary to load this register prior to a DMA operation, only the MAR in the Highway Operation Register space.

After a bus master operation completes, the current memory address minus 4 may be read from the Master Read Address Register. The initial memory address used for the bus master operation can be read through the Memory Address Register of the Highway Operational Registers.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MRAR 31	MRAR 30	MRAR 29	MRAR 28	MRAR 27	MRAR 26	MRAR 25	MRAR 24	MRAR 23	MRAR 22	MRAR 21	MRAR 20	MRAR 19	MRAR 18	MRAR 17	MRAR 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MRAR 15	MRAR 14	MRAR 13	MRAR 12	MRAR 11	MRAR 10	MRAR 09	MRAR 08	MRAR 07	MRAR 06	MRAR 05	MRAR 04	MRAR 03	MRAR 02	0	0

<31:2> MASTER READ ADDRESS 31 through 2 are read-only bits used to return the last address accessed during a DMA operation.

<1:0> These bits are not used since the DMA operations execute longword (32-bit) transfers.

Master Read Transfer Count Register

This register is not used by the 2115. When executing bus master operations, the 2115 uses the Total Transfer Count Register (TTCR) of the Highway Operational Registers to determine the number of bus operations to perform.

Mailbox Empty/Full Status Register

The register is not used by the 2115 but is shown here for completeness.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMB 4D	IMB 4C	IMB 4B	IMB 4A	IMB 3D	IMB 3C	IMB 3B	IMB 3A	IMB 2D	IMB 2C	IMB 2B	IMB 2A	IMB 1D	IMB 1C	IMB 1B	IMB 1A

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
OMB 4D	OMB 4C	OMB 4B	OMB 4A	OMB 3D	OMB 3C	OMB 3B	OMB 3A	OMB 2D	OMB 2C	OMB 2B	OMB 2A	OMB 1D	OMB 1C	OMB 1B	OMB 1A

PCI Interface Interrupt Control/Status Register

The PCI Interface Interrupt Control/Status Register of the PCI Interface Operational Registers is located at an offset of 38 hex from the selected base address #1 and used to monitor and control interrupts generated by the PCI interface chip. Before an interrupt is sourced from the interface chip to the PCI bus, it must be enabled with the PCI INTERRUPT ENABLE bit in the Interrupt Control/Status Register of the Highway Operational Registers.

The following shows the bit pattern for the Interrupt Control/Status Register of the PCI Interface Operational Registers.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	INT REQ	0	TAR ABT	MAS ABT	RTC	WTC	IMB SRC	OMB SRC

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RTC IE	WTC IE	0	IMB IE	IMB EMB	IMB EMA	IMB EBB	IMB EBA	0	0	0	OMB IE	OMB EMB	OMB EMA	OMB EBB	OMB EBA

<31:24> These 8 bits control endian conversion and must be set to zero for the 2115 to operate properly.

<23> INTERRUPT REQUEST is a read-only bit that is set when the PCI interface chip is requesting service. This bit reflects the interrupt output of the PCI chip and does not indicate that the PCI bus interrupt is asserted. If the PCI Interrupt Enable bit of the Interrupt Control/Status Register of the Highway Operational Registers is set to a one, this bit indicates that a PCI bus interrupt is requested.

<22> This bit is not used and read as a zero.

<21> TARGET ABORT is a read/write-to-clear bit that indicates when the 2115 executes a bus master transfer and the addressed target aborts the transfer. An interrupt source is generated when this bit is set. This bit is cleared by writing a one to this bit position.

<20> MASTER ABORT is a read/write-to-clear bit that indicates when the 2115 executes a bus master operation and the addressed target does not respond. An interrupt source is generated when this bit is set. A write operation to this register with this bit set to a one clears the bit.

<19> READ TRANSFER COMPLETE is a read/write-to-clear bit that is set when the Master Read Transfer Count Register is decremented to zero.

<18> WRITE TRANSFER COMPLETE is a read/write-to-clear bit that is set when the Master Write Transfer Count Register is decremented to zero.

<17> INCOMING MAILBOX INTERRUPT SOURCE is a read/write-to-clear bit that is set to a one when the mailbox selected by bits 12 through 8 of this register are written. Since the 2115 does not use the mailbox registers, this bit should not be set.

<16> OUTGOING MAILBOX INTERRUPT SOURCE is a read/write-to-clear bit that is set to a one when the mailbox selected by bits 4 through 0 of this register is read. Since the 2115 does not use the mailbox registers, this bit should not be set.

<15> READ TRANSFER COMPLETE INTERRUPT ENABLE is a write/read bit used to enable and disable the generation of an interrupt when the read transfer count is exhausted.

<14> WRITE TRANSFER COMPLETE INTERRUPT ENABLE is a write/read bit used to enable and disable the generation of an interrupt when the write transfer count is exhausted.

<13> This bit is not used and read as a zero.

<12> INCOMING MAILBOX INTERRUPT ENABLE is a write/read bit that is used to enable/disable the generation of an interrupt source when a preselected incoming mailbox register is written. Bits 11 through 8 of this register select which mailbox register write operation will generate the interrupt source.

<11:10> INCOMING INTERRUPT SELECT bits are used to select which incoming mailbox write operations are to generate an interrupt source. Since the 2115 does not used the mailbox registers, these bits should be set to zero.

<9:8> INCOMING MAILBOX BYTE INTERRUPT SELECT bits are used to select which byte of the mailbox, selected by bits 11 and 10 of this register, is actually to cause the interrupt source. Since the 2115 does not used the mailbox registers, these bits should be set to zero.

<7:5> These bits are not used and read as zeros.

<4> OUTGOING MAILBOX INTERRUPT ENABLE is a write/read bit that is used to enable/disable the generation of an interrupt when a preselected outgoing mailbox register is written. Since the 2115 does not used the mailbox registers, these bits should be set to zero.

<3:2> OUTGOING INTERRUPT SELECT bits are used to select which outgoing mailbox write operations are to generate an interrupt source. Since the 2115 does not used the mailbox registers, these bits should be set to zero.

<1:0> OUTGOING MAILBOX BYTE INTERRUPT SELECT bits are used to select which byte of the mailbox, selected by bits 3 and 2 of this register, is actually to cause the interrupt source. Since the 2115 does not used the mailbox registers, these bits should be set to zero.

Bus Master Control/Status Register

The Bus Master Control/Status Register is used to monitor/control bus master operations and to check status of the two PCI interface chip FIFOs. The following diagram shows the bit pattern for the Bus Master Control/Status Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	MBX RST	INF RST	OTF RST	AON RST	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	RDT ENA	0	0	0	WTT ENA	0	0	ITC ZERO	OTC ZERO	IFE MT	IFI 4+	IFI FUL	OTF MT	OTF 4+	OTF FUL

The bits shown as zeros in the preceding bit layout must be set to zeros when writing to the Bus Master Control/Status Register. The bits that have non-zero indications are the only useful bits in this register. The remaining bits are reserved for diagnostic purposes. Even though some of these bits are shown as zeros, they may eventually be read as ones. Note that the Inbound FIFO refers to the internal PCI controller interface chips' FIFO for executing CAMAC read operations and the Outbound FIFO refers to highway write operations.

<31:28> These bits are not used but must be written to zero.

<27> MAILBOX FLAG RESET is a write-only bit which resets all of the mailbox status flags.

<26> INBOUND FIFO RESET is a write-only bit which clears the inbound FIFO, Inbound FIFO +4 bit, Inbound FIFO Full and sets the Inbound FIFO Empty flag.

<25> OUTBOUND FIFO RESET is a write-only bit which clears the outbound FIFO, Outbound FIFO +4 bit, Outbound FIFO Full and sets the Outbound FIFO Empty bit.

<24> ADD-ON RESET is a write-only bit which, when set to a one, resets all of the Highway Operational Registers.

<23:15> These bits are not used but must be written to zero.

<14> READ TRANSFER ENABLE (RDT ENA) is a write/read bit used to enable DMA operations from PCI memory to the 2115 for CAMAC write operations. Setting this bit to a zero before the transfer count expires suspends the active transfer.

<13:11> These bits are not used but must be written to zero.

<10> WRITE TRANSFER ENABLE (WTT ENA) is a write/read bit used to enable DMA operations from the 2115 to PCI memory for CAMAC read operations. Setting this bit to a zero before the transfer count expires suspends the active transfer.

<09:08> These bits are not used but must be written to zero.

<07> INBOUND TERMINAL COUNT ZERO (ITC ZERO) is a read-only bit that is set to a one indicating that the Master Write Transfer Count is zero.

<07> OUTBOUND TERMINAL COUNT ZERO (OTC ZERO) is a read-only bit that is set to a one indicating that the Master Read Transfer Count is zero.

<5> INBOUND FIFO EMPTY is a read-only bit which is set when the inbound FIFO contains no data.

<4> INBOUND FIFO +4 is a read-only bit and is set to a one as long as there are at least 4 longwords of data in the inbound FIFO.

<3> INBOUND FIFO FULL is a read-only bit that is set when the incoming FIFO is full.

<2> OUTBOUND FIFO EMPTY is a read-only bit which is set when the outbound FIFO contains no data.

<1> OUTBOUND FIFO +4 SPACES is a read-only bit and is set to a one as long as there are at least 4 empty longwords in the outbound FIFO.

<0> OUTBOUND FIFO FULL is a read-only bit that is set when the outbound FIFO is full.

Highway Operational Registers

The following section describes the registers used to control operations directed toward the Serial Highway. The base address of these registers is set dynamically when the computer is powered-up and the PCI BIOS is executed. The PCI BIOS determines the resources required by each add-in card and allocates memory and I/O addresses accordingly. After the setup has been completed, the Base Address Register #2 from the PCI Interface Registers may be read to determine the address allocated to the 2115 by the PCI BIOS.

Address	Register Description	Mnemonic
Base + 00	Control/Status Register	CSR
Base + 04	Interrupt Control/Status Register	ICSR
Base + 08	Timer Control Register	TCR
Base + 0C	Command Memory Address Register	CMA
Base + 10	Command Memory Data Register	CMD
Base + 14	List Transfer Count Register	LTCR
Base + 18	Total Transfer Count Register	TTCR

Address	Register Description	Mnemonic
Base + 1C	Memory Address Register	MAR
Base + 20	Buffer Interval Counter	BIC
Base + 24	Multi-Buffer Control Register	MBMCT
Base + 28	Demand FIFO Register	DFR
Base + 2C	Reset Interface	RSTIFC
Base + 30	Reset Demand FIFO	RSTDFR
Base + 34 through Base + 3F	Reserved	

Appendix B contains a composite register layout chart for the 2115.

Control/Status Register

The Control/Status Register (CSR) is a write-read register located at an offset of 0 from the selected base address. This register is used to control and monitor various operations occurring within the 2115 and on the Serial Highway. Since this register contains read-only, write-only, and write/read bits, two bit patterns are shown. Those bits that are shown as zero for the write layout must be set to zero when writing to this register.

Control/Status Register(CSR)

Write Operations:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	RLD ENA	APND STAT	SUSP	DMA DIR	DMA ENA	HWY/ LIST	GO

Read Operations:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR CD3	ERR CD2	ERR CD1	ERR CD0	0	DERR	TMO	NO SYNC	ERR	N>23	LPE	TPE	ADNR	STE	NOX	NOQ

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	DMD OFLO	DMD PND	0	0	0	DONE	RLD ENA	APND STAT	SUSP	DMA DIR	DMA ENA	HWY/ LIST	0

<31:28> ERROR CODE3 through 0 are read-only bits which encode the source of a 2115 error. The following chart shows the error code hex pattern along with the source

of the error. The error sources are encoded by a priority encoder resulting in the highest priority error being displayed in the case where multiple errors are encountered. Individual bits in the CSR may be read to determine other error sources that may have occurred along with the highest priority error encoded..

Error Code	Error Source
F	Reserved.
E	Reserved.
D	NO SYNC. The 2115 is not receiving a clock from the highway receive port.
C	ADDRESS_NOT_RECOGNIZED. An addressed highway command message was transmitted by the 2115 and then received by the 2115; indicating that the addressed slave did not accept the message.
B	TIMEOUT. A highway command message was transmitted by the 2115 and a reply message was not received within the preselected timeout period.
A	SERIAL TRANSMISSION ERROR is set when the 2115 detects either a Longitudinal or Transverse Parity Error on an incoming message.
9	N GREATER THAN 23. N>23 indicates that a Q-Scan operation in a CAMAC chassis resulted in the station number incrementing beyond 23.
8	NO-X. A highway operation to an addressed CAMAC chassis resulted in a CAMAC X-response of zero.
7	NO-Q. A highway operation to an addressed CAMAC chassis resulted in a CAMAC Q-response of zero.
6	Reserved.
5	Reserved.
4	SERIAL ERROR is set whenever the ERR bit in a highway reply message is set. This bit is set in response to a parity error on the command message to the SCC.
3	Reserved.
2	Reserved.

1 ILLEGAL COMMAND. This error code is generated when the list processor encounters an instruction that it can not execute.

0 NO ERROR. There are no errors to report from the 2115.

<27> This bit is not used and read as a zero.

<26> DELAYED ERROR is a read-only bit which is set when the DERR bit in a highway reply status byte is set. This bit is set after an operation that caused an ERR. Refer to bit <23> of this register.

<25> TIMEOUT is a read-only bit that is set when the 2115 transmits a command message to a slave and the slave does not respond within the selectable timeout period. This timeout may also be generated when a Q-repeat operation is executed and a Q=1 response is not received within the timeout period. This period can range from 3 to 15 seconds.

<24> NO SYNC is a read-only bit that is set when the 2115 highway driver is not receiving a highway clock on the 2115 receive port.

<23> ERR is a read-only bit that is set when the ERR bit is set in a highway reply message status byte. This bit is set when a No-X is encountered or the SCC detects a parity error on the incoming message.

<22> N GREATER THAN 23 is a read-only bit that is set when a Q-Scan operation in an addressed CAMAC chassis terminated due to the CAMAC station number incrementing beyond 23.

<21> LONGITUDINAL PARITY ERROR is a read-only bit that is set when the 2115 detects a longitudinal parity error on the incoming message.

<20> TRANSVERSE PARITY ERROR is a read-only bit that is set when the 2115 detects a transverse parity error on the incoming message.

<19> ADDRESS_NOT_RECOGNIZED is a read-only bit that is set when the 2115 transmits a command message to a SCC and receives the message back; indicating that the addressed SCC did not respond.

<18> SERIAL TRANSMISSION ERROR is a read-only bit which is set when the 2115 detects a parity error or the ERR bit is set in a Serial Highway status byte. The following equation describes the STE bit.

$$STE = TPE + LPE + ERR$$

<17> NO-X is a read-only bit which is set when an addressed CAMAC chassis operation resulted with a CAMAC X-response of zero.

<16> NO-Q is a read-only bit which is set when an addressed CAMAC chassis operation resulted with a CAMAC Q-response of zero.

<15:13> These bits are not used and read as zeros.

<12> DEMAND OVERFLOW is a read-only bit that is set when the 2115 has 2048 demands pending in the Demand FIFO Register and a subsequent demand message is received. This bit may be cleared by a write operation to the Clear Demand FIFO Register address.

<11> DEMAND PENDING is a read-only bit which is set when at least one demand message is contained in the Demand FIFO Register.

<10:8> These bits are not used and read as zeros.

<7> DONE is a read-only bit which indicates when the 2115 is not executing any list operations. Once a list operation is initiated, the DONE bit is set to a zero until the operation is completed.

<6> RELOAD ENABLE a write/read bit used to control the automatic reloading of the Memory Address and Total Transfer Count Registers once the transfer count register is exhausted. Setting this bit to a one enables reloading and a zero disables reloading.

<5> APPEND STATUS is a write/read bit that enables and disables the appending of a status word after a list execution terminates. Setting this bit to a one enables the status append and a zero disables the function. Refer to the Appending Status section of this manual for additional information.

<4> SUSPEND is a write/read bit used to suspend list processing operations in the 2115. After SUSPEND is set to a one, the host must wait for the DONE bit to be set before using the 2115 for other operations. After the DONE bit has been set, the SUSPEND bit may then be written back to a zero.

<3> DMA DIRECTION is a write/read bit that specifies the direction of DMA transfers. A direction of zero specifies DMA transfers from host memory to the 2115 (write operations). A one specifies DMA transfers from the 2115 to host memory (read operations).

<2> DMA ENABLE is a write/read bit that is used to enable/disable DMA activity. Setting this bit to a one enables DMA operations to occur when the list processor requests data transfers. A zero allows for programmed I/O transfers and disables DMA operations.

<1> HIGHWAY/LIST is a write/read bit used to specify whether DMA operations are to load the Command Memory or transfer data for a CAMAC operation. Setting this bit to a one enables the loading of the Command Memory using DMA.

Setting this bit to a zero enables the transfer of data to/from the 2115 for CAMAC operations.

<0> GO is a write-only bit that is used to initiate list processing operations at the address specified in the Command Memory Address Register. This bit is also used to start a DMA Load Command Memory Data operation.

Interrupt Control/Status Register

The Interrupt Configuration Register (ICSR) is a write/read register located at an offset of 4 from the selected base address. This register is used to enable/disable the various sources of interrupts on the 2115. Status bits are also in this register reflecting the state of the interrupt sources. Once enabled and sourced, the 2115 asserts the PCI INTA signal to the host.

The 2115 has four sources of interrupts as follows:

1.) List Processing Done
2.) Demand
3.) Multi-Buffer Interrupt
4.) A PCI Interface Controller Interrupt

The ICSR contains individual enables and status bits for each of the four sources. To allow an interrupt status bit to be enabled to generate an interrupt, the corresponding enable bit must be set to a one. Interrupt sources are disabled by setting the bit to a zero. After an interrupt source bit is set, it may be cleared by writing a one in the bit location to be cleared.

The following diagram shows the bit layout of the Interrupt Control/Status Register.

Interrupt Control/Status Register (ICSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	INFC INT	MBM INT	DMD INT	DONE INT	INFC IE	MBM IE	DMD IE	DONE IE

<31:8> These bits are not used and read as zero.

<7> PCI INTERFACE INTERRUPT SOURCE is a read only bit which reflects the status of the PCI interface chips' interrupt source. The actual source of these interrupts may be found in the PCI Interrupt Control/Status Register.

<6> MULTIBUFFER MEMORY INTERRUPT SOURCE is a read/write-to-clear bit that is set when the Buffer Interval Counter overflows and this interrupt source

is enabled . A write of this bit with data set to one clears the source. The MBM IE bit in this register must be set to a one to enable this source.

<5> DEMAND INTERRUPT SOURCE is a read/write-to-clear bit that is set whenever a demand message is received by the 2115 and this interrupt source is enabled. A write of this bit with data set to one clears the source. The DMD IE bit in this register must be set to a one to enable this interrupt source.

<4> DONE INTERRUPT SOURCE is a write/read bit that is set when the 2115 completes execution of a list and this interrupt source is enabled. A write of this bit with data set to a one clears the source. The DONE IE bit in this register must be set to a one to enable this source.

<3> PCI INTERFACE INTERRUPT ENABLE is a write/read bit that is used to enable/disable the generation of a PCI interrupt request when the PCI Interface chip requires service. Setting this bit to a one enables the interrupt source and a zero disables the source.

<2> MULTIBUFFER MEMORY INTERRUPT SOURCE ENABLE is a write/read bit that enables/disables the assertion of an interrupt source when the Buffer Interval Counter overflows during multibuffer operations. Setting this bit to a one enables this interrupt source and disabled with a zero.

<1> DEMAND INTERRUPT SOURCE ENABLE is a write/read bit that enables/disables the generation of an interrupt source when the 2115 receives a demand message from the highway. Setting this bit to a one enables the interrupt source and a zero disables the source.

<0> DONE INTERRUPT SOURCE ENABLE is a write/read bit used to enable/disable the generation of an interrupt source when a list processing operation completes. Setting this bit to a one enables the interrupt source and a zero disables the source.

Timer Control Register

The Timer Control Register (TCR) is a write/read register located at an offset of 08 hex from the selected base address. This register is used to specify the frequency at which list execution is initiated during timer initiated list processing operations.

The timer frequency can range from 250 Kilohertz to .059 hertz. This range yields 'tic' rates (periods) from 40 microseconds to 16.777 seconds in 1 microsecond increments. This timer rate specification does not refer to the interval at which each element in the list is executed but the rate at which the entire list is initiated. This automatic mode is enabled with bit 24 in this register.

The 2115 provides for an external clock input that controls the rate at which the list is started. This input is routed through the external clock LEMO connector mounted on the front panel of the 2115. The clock period may range from 250 KHz to DC. The input signal is low true and must have a minimum pulse width of 200 nanoseconds and a maximum of 1 microsecond.

If the timer rate selected for list execution is faster than the time it takes to execute the entire list, the clock transitions that occur while the list is executing are ignored.

The data for this register is calculated as follows:

$$((1/\text{Desired Frequency})/1 \text{ X } 10^{-6})\text{-}1$$

The following diagram shows the bit layout of the Timer Control Register.

Timer Control Register (TCR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	CLK SRC	TMR ENA	TCR 23	TCR 22	TCR 21	TCR 20	TCR 19	TCR 18	TCR 17	TCR 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TCR 15	TCR 14	TCR 13	TCR 12	TCR 11	TCR 10	TCR 9	TCR 8	TCR 7	TCR 6	TCR 5	TCR 4	TCR 3	TCR 2	TCR 1	TCR 0

<31:26> These bits are not used and read as zeros.

<25> CLOCK SOURCE is a write/read bit that selects the source of the timing signal that initiates list execution when the TMR ENA bit is set to a one. Setting the CLK SRC bit to a one selects the external clock input and a zero selects the internal timer.

<24> TIMER ENABLE is a write/read bit that is used to enable/disable timer initiated list processing operations. Setting this bit to a one enables timer initiated operations and a zero disables timer operations.

<23:0> TIMER CONTROL RATE23 through 0 are write/read bits that are used to select the rate at which list processing operations are initiated when the timer operations are enabled. The least significant bit of this field corresponds to a 1 microsecond interval. The data loaded in this register is actually one less than the desired interval. For example, to select a 250 microsecond interval, load the value 249 into this register.

Command Memory Address Register

The Command Memory Address Register (CMA) is a write/read register located at an offset of 0C Hex from the selected base address. This register is for several purposes. The CMA is used to specify the start address for list processing, specify the memory address at which subsequent write/read operations to the Command Memory Data Register occur, and to read the last address that the list processor accessed during a list processing operation.

The Command Memory Data Register (CMD) is a 32K X 32 memory that is used to hold the list processing instructions to be executed. The CMD is loaded by executing programmed I/O write operations to the Command Memory Data at offset 10 Hex. Prior to executing the initial write to the memory, the CMA must first be loaded with the first address location to access. The valid address range is 0 to 7FFF Hex. After the CMA is loaded, the first CMD data word may then be written. After a write operation to the CMD, the CMA is automatically incremented to the next address location. This allows the CMD to be loaded without having to write the CMA for every CMD write operation.

If it is necessary to read the CMD, the CMA must first be loaded with the initial address to be accessed. After a word of data is read from the CMD, the CMA is automatically incremented to the next address location.

A bit in this register is also provided which initiates a single list processing operation. Setting this bit causes the same operation as setting the GO bit in the Control/Status Register.

The format of the Command Memory Address Register is shown in the following diagram.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LIST GO	CMA 14	CMA 13	CMA 12	CMA 11	CMA 10	CMA 9	CMA 8	CMA 7	CMA 6	CMA 5	CMA 4	CMA 3	CMA 2	CMA 1	CMA 0

<31:16> These bits are not used and read as zeros.

<15> LIST GO is a write-only bit which is set to a one to initiate a list processing operation. Setting this bit has the same effect as setting the GO bit in the CSR. This bit is not latched.

<14:0> COMMAND MEMORY ADDRESS14 through 0 are used for specifying the initial address for CMD write/read operations, the initial address for list execution, and for determining where the list processor halted after list processing.

Command Memory Data Register

The Command Memory Data Register (CMD) is a write/read register located at an offset of 10 Hex from the selected base address. This register is used to load the instructions for the list processor. The format of these instructions/commands may be found in the Command Memory Instructions section of this manual.

The Command Memory is a 32K X 32 memory which is accessed by programmed I/O or loaded via DMA. Before initial words may be written to the CMD, the Command Memory Address Register (CMA) must be loaded. After the CMA is loaded with the first address of the CMD to access, the CMD may then be written. After a write operation to the CMD, the CMA is automatically incremented to the next sequential address location. This eliminates the need to reload the CMA for every access to the CMD.

The CMD may also be loaded using DMA. To load the command memory using DMA, first load the CMA with the initial address to access. Then, set the HWY/LIST bit in the Control/Status register to a one when setting the GO bit. When the operation terminates, the DONE bit in the Control/Status register is set to a one.

If necessary, the CMD may be read to verify its contents. As with write operations to the CMD, the CMA must be loaded prior to the initial access to the CMD. After a read operation is executed to the CMD, the CMA is automatically incremented.

The format of the Command Memory Data Register is shown in the following diagram.

Command Memory Data Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMD 31	CMD 30	CMD 29	CMD 28	CMD 27	CMD 26	CMD 25	CMD 24	CMD 23	CMD 22	CMD 21	CMD 20	CMD 19	CMD 18	CMD 17	CMD 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMD 15	CMD 14	CMD 13	CMD 12	CMD 11	CMD 10	CMD 9	CMD 8	CMD 7	CMD 6	CMD 5	CMD 4	CMD 3	CMD 2	CMD 1	CMD 0

<31:0> COMMAND MEMORY DATA31 through 0 are write/read bits used to load/read the 32K X 32 Command Memory.

List Transfer Count Register

The List Transfer Count Register (LTCR) is a read-only register located at an offset of 14 Hex from the selected base address. This register returns the two's complement of the number of transfers remaining during a block transfer mode of operation. When the list processor encounters a block transfer instruction, it loads the 32-bit List Transfer Count Register with the count specified in the instruction. This value is the two's complement of the maximum number of 16-bit transfers that are to occur for the block transfer operation. When this counter is incremented to zero, the block transfer operation terminates and the next instruction in the list is interpreted. If an error occurs during the block operation, the LTCR may then be read to determine the number of transfers remaining.

The following diagram shows the bit pattern for the List Transfer Count Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LTC 31	LTC 30	LTC 29	LTC 28	LTC 27	LTC 26	LTC 25	LTC 24	LTC 23	LTC 22	LTC 21	LTC 20	LTC 19	LTC 18	LTC 17	LTC 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LTC 15	LTC 14	LTC 13	LTC 12	LTC 11	LTC 10	LTC 9	LTC 8	LTC 7	LTC 6	LTC 5	LTC 4	LTC 3	LTC 2	LTC 1	LTC 0

<31:0> LIST TRANSFER COUNT31 through 0 are read-only bits which reflect the number of transfers remaining after a block transfer operation concludes. This data is in two's complement format.

Total Transfer Count Register

The Total Transfer Count Register (TTCR) is a write/read register located at an offset of 18 Hex from the selected base address. This register is used to specify the maximum number of DMA transfers that are to occur. Before a list operation is executed where data transfers to/from the highway occur using Direct Memory Access (DMA), the Total Transfer Count Register must be loaded with the two's complement of the maximum number of 16-bit words to transfer. In case that the list terminates prematurely, the TTCR may be read to determine the number of 16-bit words remaining to be transferred.

Along with write access to the TTCR from the host by programmed I/O, the TTCR may also be loaded by using a list instruction. When the list processor encounters this instruction, the contents of the count specification are loaded into the TTCR.

The TTCR is incremented twice for every DMA write or read access to PCI. When the counter increments to zero, the transfer is considered complete.

The following diagram shows the bit layout for the Total Transfer Count Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TTCR 31	TTCR 30	TTCR 29	TTCR 28	TTCR 27	TTCR 26	TTCR 25	TTCR 24	TTCR 23	TTCR 22	TTCR 21	TTCR 20	TTCR 19	TTCR 18	TTCR 17	TTCR 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TTCR 15	TTCR 14	TTCR 13	TTCR 12	TTCR 11	TTCR 10	TTCR 9	TTCR 8	TTCR 7	TTCR 6	TTCR 5	TTCR 4	TTCR 3	TTCR 2	TTCR 1	TTCR 0

<31:0> TOTAL TRANSFER COUNT31 through 0 are write/read bits which specify the maximum number of DMA transfers to execute. A read of this register returns the two's complement of the number of transfers remaining.

Memory Address Register

The Memory Address Register (MAR) is a write/read register located at an offset of 1C Hex from the selected base address. This register is used to specify the initial Direct Memory Access (DMA) address. Before any DMA operations are executed by the 2115, the MAR must be loaded with the first address to be accessed during DMA write or read operations. After the 2115 executes the first DMA operation, the MAR is automatically incremented to the next sequential longword address. Note that all DMA MAR specifications must be longword aligned.

When the Reload Enable (RLD ENA) bit in the CSR is enabled, the MAR may automatically reload itself after the Total Transfer Count Register is incremented to zero. The value that is reloaded is contained in the MAR. This feature allows the 2115 to DMA data to a circular buffer.

Along with write access to the MAR from the host by programmed I/O, the MAR may also be loaded by using a list instruction. When the list processor encounters this instruction, the contents of the address specification are loaded into the MAR.

DMA operations to/from the 2115 are always 32-bits in length. Therefore, the MAR specification must be on a longword boundary.

The following diagram shows the bit pattern for the Memory Address Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR 31	MAR 30	MAR 29	MAR 28	MAR 27	MAR 26	MAR 25	MAR 24	MAR 23	MAR 22	MAR 21	MAR 20	MAR 19	MAR 18	MAR 17	MAR 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MAR 15	MAR 14	MAR 13	MAR 12	MAR 11	MAR 10	MAR 9	MAR 8	MAR 7	MAR 6	MAR 5	MAR 4	MAR 3	MAR 2	0	0

<31:2> MEMORY ADDRESS31 through 2 are write/read bits used to specify the initial DMA Memory Address.

<1:0> These bits are not used and read as zeros.

Buffer Interval Counter

The Buffer Interval Counter (BIC) is a write/read register located at an offset of 20 Hex from the selected base address. This counter is used to define the number of DMA transfers that the 2115 executes before a multibuffer flag bit is set and an interrupt is generated. The BIC allows the 2115 to use host computer memory to simulate a multibuffer memory. When using this mode, the 2115 is continuously acquiring data and transferring the read data to memory via DMA. When a predetermined number of DMA transfers have occurred, a multibuffer flag is set and optionally an interrupt generated. When the host sees the flag set, it may then read the buffer of data received. As additional words are transferred to memory, subsequent multibuffer flags are set.

When the Total Transfer Count Register is incremented to zero, the Memory Address Register and Total Transfer Register are reloaded, creating a circular buffer. The reloading of the Memory Address Register and the Total Transfer Count Register is controlled through the Reload Enable bit in the Control/Status Register. As long as the Reload Enable bit is set, this data acquisition sequence continues until the 2115 is disabled.

Refer to the Multibuffered Data Acquisition section of this manual for additional information.

The following diagram shows the bit pattern for the Buffer Interval Counter.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BIC 31	BIC 30	BIC 29	BIC 28	BIC 27	BIC 26	BIC 25	BIC 24	BIC 23	BIC 22	BIC 21	BIC 20	BIC 19	BIC 18	BIC 17	BIC 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
BIC 15	BIC 14	BIC 13	BIC 12	BIC 11	BIC 10	BIC 9	BIC 8	BIC 7	BIC 6	BIC 5	BIC 4	BIC 3	BIC 2	BIC 1	BIC 0

<31:0> BUFFER INTERVAL COUNTER31 through 0 are write/read bits used to specify the number of DMA transfers that the 2115 executes before a multibuffer flag is set during multibuffered data acquisition.

Multibuffer Memory Control Register

The Multibuffer Memory Control Register (MBMCT) is a write/read register located at an offset of 24 Hex from the selected base address. This register is used to enable the multibuffer flags and to monitor the status of the flags.

Once multibuffer activity has been enabled, the multibuffer flags are incrementally set in this register as the Buffer Interval Counter (BIC) expires. The BIC is decremented as the 2115 executes each DMA operation. When the BIC is exhausted, the next sequential flag bit is set. After the fourth flag is set, the next expiration of the BIC causes the first flag to be set. If the host computer has not cleared a flag, via programmed I/O, before the 2115 loops around to set it again, the flag overflow bit is set. Refer to the Multibuffered Data Acquisition section of this manual for additional information.

The following diagram shows the bit pattern for the Multibuffer Control Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	MBM ENA	FLG OFLO	FLG 3	FLG 2	FLG 1	FLG 0

<31:6> These bits are not used and read as zeros.

<5> MULTIBUFFER ENABLE is a write/read bit used to enable/disable the generation of the flag bits that correspond to the expiration of the Buffer Interval Counter (BIC). Setting this bit to a one enables the flags and disabled by a zero.

<4> FLAG OVERFLOW is a read/write-to-clear bit which is used to signify an overflow condition. This bit is set whenever the 2115 needs to set a multibuffer

flag and the flag is already set, indicating that a buffer of data is being overwritten before the host could read the buffer and clear the flag. A write operation with this bit set to a one causes the overflow condition to be cleared.

<3:0> MULTIBUFFER FLAG3 through 0 are read/write-to-clear bits that are sequentially set as the Buffer Interval Counter expires during multibuffer data acquisition. A write operation to these bits with any bit set to a one causes the corresponding flag to be cleared.

Demand FIFO Register

The Demand FIFO Register (DFR) is a read-only register located at an offset of 28 Hex from the selected base address. This register is loaded with demand message data received from the highway. When a CAMAC chassis requires service, it may generate an asynchronous demand message on the highway. When the 2115 receives these demands, they are placed in a 2K X 16 FIFO. Optionally, the 2115 may assert a PCI interrupt when the 2115 receives a demand message. The FIFO data contains the crate address of the chassis that generated the demand message along with an 5-bit identifier.

As long as there is at least one demand message contained in the Demand FIFO Register the DEMAND PENDING (DMD PND) bit in the CSR is set to a one. If the DFR is full (2048 demands) and an additional demand message is received from the highway, the DEMAND OVERFLOW (DMD OFL) bit in the CSR is set to a one. The Demand FIFO Register and the DEMAND OVERFLOW bit are cleared by power-up, a write to the Reset Interface address, and by a write to the Reset Demand FIFO address.

The following diagram shows the bit pattern for the Demand FIFO Register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	SGL 16	SGL 8	SGL 4	SGL 2	SGL 1	0	0	C 32	C 16	C 8	C 4	C 2	C 1

<31:13> These bits are not used and read as zeros.

<12:8> SGL16 through 1 are read-only bits which indicate the SGL pattern received from the demand message.

<7:0> These bits are not used and read as zeros.

<5:0> CRATE ADDRESS32 through 1 are read-only bits which indicates the crate address that generated the demand message.

Reset Interface

The Reset Interface (RSTIFC) is a write-only address location located at an offset of 2C Hex from the selected base address. A write operation to this address with any data pattern causes the 2115 to be reset to an initial state.

Reset Demand FIFO Register

The Reset Demand FIFO Register is a write-only address location located at an offset of 30 Hex from the selected base address. A write operation to this address with any data pattern causes the Demand FIFO Register and the DEMAND OVERFLOW bit to be cleared.

List Processing Operation

The 2115 provides a mechanism to execute a list of CAMAC operations from a preloaded memory. The memory is a 32K x 32 memory and is called the Command Memory. The Command Memory can hold up to 32768 CAMAC commands that are to be executed by the list processor. The CAMAC operations include Single Transfers, Standard Block Transfers, Enhanced Block transfers and Single Inline Write operations. The Command Memory can be loaded to contain many individual lists. Each individual list must be terminated with a Halt instruction to signify its end. Before a list is executed, the starting address for list processing is preloaded in the Command Memory Address Register.

The Command Memory can be loaded by either programmed I/O or by Direct Memory Access (DMA). Prior to loading the Command Memory, the Command Memory Address Register (CMA) must be loaded with the initial address to be loaded. After the CMA is loaded, the Command Memory Data (CMD) may then be written. To load the CMD by programmed I/O, simply write the desired data to the CMD register address. Each write operation to the CMD causes the CMA to be incremented to the next location. To load the CMD using DMA, the Total Transfer Count Register (TTCR) must be loaded with the number of elements to load into the CMD. This value must be in two's complement format. Once the TTCR is loaded, the HWY/LIST bit and the GO bit in the Control/Status Register (CSR) must be set to one. After the operation is initiated, the DONE bit in the CSR is negated until the CMD is loaded.

Once the CMD is loaded with the instruction(s) to execute, the list may be executed. Prior to list execution, the initial address for list processing must be loaded in the CMA. The list may be initiated by either programmed I/O, an internal automatic timer, or by and external timer/event. To start the list by programmed I/O, the GO bit in the CSR may be set or bit 15 in the CMA register may be set. Both bit locations provide the same function.

To initiate list execution by timer or external timer/event, the Timer Control Register (TCR) must be loaded appropriately. For internal timer operations, the TCR is loaded with the interval at which the list is executed. For external triggering, the TCR must be setup to enable the external event. Please refer to the Timer Control Register section of this manual for additional information.

After list processing is started, the List Processor reads the CMD at the address pointed to be the CMA. The List Processor then decodes the instruction type and executes the appropriate operation. As elements of the list are read by the List Processor, the CMA is incremented. The List Processor continues processing list elements until an error is encountered or a Halt instruction is found. When list processing stops, the DONE bit in the CSR is then set to a one indicating that the list operation is complete.

The following sections describe the various List Processing Instructions available for the 2115.

Command Memory Instruction Formats

The 2115 contains a 32K x 32 word memory used to hold the list processing instructions to be executed by the hardware list processor. When list processing is initiated, the hardware list processor examines the header instruction of the first list instruction. If the header indicates that the instruction is a data transfer instruction to a CAMAC chassis, the hardware processor forms the necessary command to transmit on the highway. After the highway transaction is complete, the Command Memory Address is incremented and the next instruction interpreted.

The lower 16-bits of the first word of a command instruction specifies the type of the instruction. This 16-bit word is referred to as the instruction header. The following diagram shows the format of the instruction header.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CM 0	0	C 32	C 16	C 8	C 4	C 2	C 1	0	TM 2	TM 1	QM 2	QM 1	0	WS 1	AD

Bit 15 of the instruction header specifies whether the instruction is a CAMAC operation or a non-CAMAC operation. When CM0=0, the instruction executes a CAMAC operation. Setting CM0=1 indicates a "special" or non-CAMAC instruction.

If the CM0 bit is a zero, indicating that highway data transfers are to occur, bits 13 through 8 specify the crate address at which the addressed command is directed. For special instructions, these bits may take on other definitions. Bits 13 through 8 for data transfer instructions allow a crate address value from zero to 63. Only crate addresses in the range of 1 to 62 are valid. All other crate addresses are reserved.

The definitions of the remaining bits in the instruction header vary depending on the type of command. The next two sections fully define bits 6 through 0 of the instruction header.

CAMAC Instructions

For all CAMAC instructions, the instruction header has bits 15 set to zero. Bits 13 through 8 are used to specify the crate address of the CAMAC chassis to access during the operation. Other bits in this word define the Transfer Mode, Q-Mode, and Data Word Size. The following diagram shows the instruction header for CAMAC instructions.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	C 32	C 16	C 8	C 4	C 2	C 1	0	TM 2	TM 1	AM 2	AM 1	0	WS 1	AD

Bits 15 and 14 of the CAMAC instruction header must be set to zeros.

Bits 13 through 8 are used to specify the node address of the CAMAC chassis to access.

Bits 6 and 5 are the Transfer Mode bits and specify the CAMAC transfer protocol as follows:

TM2	TM1	Transfer Mode
0	0	Single Operation
0	1	Standard Block Transfer Operation
1	0	Enhanced Block Transfer Operation
1	1	Single Inline Write Operation

The Single Operation Transfer Mode simply transfers one data word to/from the CAMAC chassis for the specified CAMAC command. These operations may include write, read and control operations.

Block Transfer Operations move blocks of data to/from a CAMAC chassis for each block instruction. The number of data words to transfer during Block Transfer operations are found in additional words accompanying this instruction. These operations may include either CAMAC write or read operations.

Single Inline Write Operations are single transfer CAMAC write operations that have the CAMAC write data embedded in the list. These commands are useful for initializing modules and also allow CAMAC write operations to occur in a read command list.

The Q-Mode bits, QM2 and QM1, specify the Q-Mode to be used during the requested transfer. The mode entered also depends on the type of transfer mode selected. The following chart shows the Q-Modes supported for Single, Standard Block and Single Inline Write transfer modes.

Non-Enhanced Transfer Mode Q-Mode selections.

QM2	QM1	Q- Mode
0	0	Q-Stop Mode
0	1	Q-Ignore Mode
1	0	Q-Repeat Mode
1	1	Q-Scan Mode

When the Enhanced Block Transfer mode is selected, the Q-Mode supported are as follows:

Enhanced Transfer Mode Q-Mode selections.

QM2	QM1	Q- Mode
0	0	Q-Stop Mode
0	1	Q-Ignore Mode
1	0	Q-Repeat (CAMAC Reads Only)
1	1	Q-Ignore List Sequencer Module Mode *

* Requires use of Model 3830 List Sequencer Module

The Q-Modes for CAMAC provide a mechanism for controlling the data transfer base on the CAMAC Q-response received after every Dataway cycle. These modes include stopping when a Q=0 response is found, repeating a CAMAC command until a Q=1 response is found, or scanning through a CAMAC chassis. Please refer to the CAMAC Q-Modes section of this manual for additional information.

The Word Size bit, WS1, is used to specify the size of the CAMAC data word accessed in the CAMAC chassis. Setting the WS1 bit to a zero selects 24-bit CAMAC data words and a one selects 16-bit CAMAC data words.

The last bit in the instruction header, ABORT DISABLE, is used to enable or disable the termination of an operation when an X-response of zero occurs. Setting this bit to a zero enables the termination due to a X=0 response from a CAMAC module and a one enables the X=0 termination. This bit has no effect on Q-Scan operations that normally encounter X=0 responses.

CAMAC NAF Word

The second 16-bit word for a CAMAC instruction contains the CAMAC Station Number (N), Subaddress (A), and the Function Code (F). This second 16-bit word is the high 16-bits of the first 32-bit Command Memory Data word. The following diagram shows the format of the second instruction word containing the CAMAC NAF specification.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1

Bits 31 and 30 are not used and must be set to zero.

The Station Number 16 through 1 bits are used to specify the CAMAC Station Number to be used for the addressed operation. This allows for a CAMAC Station Number specification in the range of 0 through 31. Valid CAMAC Station Numbers that specify actual slots within the chassis are 1 through 23. Station Number 30 is a pseudo-address and is used to access the internal registers of the SCC. Station Numbers 24 through 29 and 0 are reserved. Please refer to the SCC manual for additional information on internal controller registers.

The Subaddress 8 through 1 bits are used to specify the CAMAC Subaddress to be accessed during the addressed CAMAC operation. Valid Subaddresses range from 0 to 15.

The Function Code 16 through 1 bits are used to specify the CAMAC Function Code to be used during the addressed CAMAC operation. The binary combination of the F16 and F8 bits determine the type of operation as follows.

F16	F8	Operation
0	0	CAMAC Read
0	1	CAMAC Control *
1	0	CAMAC Write
1	1	CAMAC Control *

* CAMAC Control operations are dataless operations. A Dataway cycle is executed for these commands but no data is transferred.

CAMAC Single Transfer Instructions

The Header Word and CAMAC NAF specification are common for all CAMAC instructions. This information occupies an entire 32-bit Command Memory Data word. For CAMAC Single Operations, this is the only data required. Single Transfer instructions are selected by setting both the TM2 and TM1 bits to zero. For each Single Transfer instruction encountered in the list, only one data word is transferred. The direction of the transfer is based on the CAMAC Function Code specified for the command. This instruction format may also be used to execute CAMAC control operations. To transfer multiple data words to/from a CAMAC chassis with one instruction, the CAMAC Block Transfer instruction must be used.

The following diagram shows the composite format for the CAMAC Single Operation instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	C 32	C 16	C 8	C 4	C 2	C 1	0	0	0	QM 2	QM 1	0	WS 1	AD

CAMAC Standard Block Transfer Instructions

CAMAC Standard Block Transfer instructions allow multiple data words to be transferred to/from a CAMAC chassis with one instruction. Standard Block Transfers are selected by setting the TM2 bit to a zero and the TM1 bit to a one. During Standard Block Transfers, a complete Serial Highway command/reply sequence is executed for each CAMAC data word. This allows operations such as Q-Scan modes that require the station number and subaddress be altered at the 2115 based on the received CAMAC Q and X-responses. A faster method of transferring data to/from a CAMAC crate is by using the Enhanced Block Transfer modes. These modes transfer data in a pipelined fashion by not waiting for a reply for each data word before subsequent data words are transferred. Refer to the Enhanced Block Transfer section of this manual for additional details.

Associated with each Block Transfer instruction is a Transfer Count. The Transfer Count, which is 32-bits is length, is the two's complement of the maximum number of CAMAC words to transfer during the block operation. This transfer count specifies the number of 16-bit data words to transfer. Therefore, each 16-bit CAMAC data word requires a count of one and each 24-bit data word requires a count of 2.

When the list processor finds a Block Transfer instruction in the Command Memory, it loads the List Transfer Count Register on the 2115 with the Transfer Count data found in the list. The List Transfer Count Register is then incremented as each data word is transferred to/from the CAMAC chassis. The Block Transfer operation continues until the List Transfer Count is exhausted or an error occurs. If the transfer terminates prematurely, the List Transfer Count Register may then be read to determine the number of data words remaining to transfer.

The format of the CAMAC Standard Block Transfer instruction is shown in the following diagram.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	C 32	C 16	C 8	C 4	C 2	C 1	0	0	1	QM 2	QM 1	0	WS 1	AD

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LTCR 31	LTCR 30	LTCR 29	LTCR 28	LTCR 27	LTCR 26	LTCR 25	LTCR 24	LTCR 23	LTCR 22	LTCR 21	LTCR 20	LTCR 19	LTCR 18	LTCR 17	LTCR 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LTCR 15	LTCR 14	LTCR 13	LTCR 12	LTCR 11	LTCR 10	LTCR 9	LTCR 8	LTCR 7	LTCR 6	LTCR 5	LTCR 4	LTCR 3	LTCR 2	LTCR 1	LTCR 0

CAMAC Enhanced Block Transfer Instructions

CAMAC Enhanced Block Transfer instructions allow multiple data words to be transferred to/from a CAMAC chassis with one instruction. Enhanced Block Transfers are selected by setting the TM2 bit to a one and the TM1 bit to a zero. The enhanced modes transfer data in a pipelined fashion by not waiting for a reply for each data word before subsequent data words are transferred. Note that several of the Q-Modes are not supported when selecting the Enhanced Block Transfer operations. The modes not supported are Q-Scan write, Q-Scan reads, and Q-Repeat writes. Since these modes require the CAMAC Q and X-responses to determine the next command or data word to execute, they cannot be used due to the pipelined technique.

Associated with each Block Transfer instruction is a Transfer Count. The Transfer Count, which is 32-bits is length, is the two's complement of the maximum number of CAMAC words to transfer during the block operation. This transfer count specifies the number of 16-bit data words to transfer. Therefore, each 16-bit CAMAC data word requires a count of one and each 24-bit data word requires a count of 2.

When the list processor finds a Block Transfer instruction in the Command Memory, it loads the List Transfer Count Register on the 2115 with the Transfer Count data found in the list. The List Transfer Count Register is then incremented as each data word is transferred to/from the CAMAC chassis. The Block Transfer operation continues until the List Transfer Count is exhausted or an error occurs. If the transfer terminates prematurely, the List Transfer Count Register may then be read to determine the number of data words remaining to transfer.

The format of the CAMAC Enhanced Block Transfer instruction is shown in the following diagram.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	C 32	C 16	C 8	C 4	C 2	C 1	0	1	0	QM 2	QM 1	0	WS 1	AD

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LTCR 31	LTCR 30	LTCR 29	LTCR 28	LTCR 27	LTCR 26	LTCR 25	LTCR 24	LTCR 23	LTCR 22	LTCR 21	LTCR 20	LTCR 19	LTCR 18	LTCR 17	LTCR 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LTCR 15	LTCR 14	LTCR 13	LTCR 12	LTCR 11	LTCR 10	LTCR 9	LTCR 8	LTCR 7	LTCR 6	LTCR 5	LTCR 4	LTCR 3	LTCR 2	LTCR 1	LTCR 0

CAMAC Single Inline Write Instruction

The Single Inline Write instruction allows a predetermined data word to be placed in the list and written to the specified NAF. Most often, this type of instruction is used for module initialization or as a mechanism to execute a CAMAC write operation in a read list.

The format of the CAMAC Single Inline Write instruction is shown in the following diagram.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	C 32	C 16	C 8	C 4	C 2	C 1	0	1	1	QM 2	QM 1	0	WS 1	AD

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	W 23	W 22	W 21	W 20	W 19	W 18	W 17	W 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
W 15	W 14	W 13	W 12	W 11	W 10	W 9	W 8	W 7	W 6	W 5	W 4	W 3	W 2	W 1	W 0

If it is necessary to embed a CAMAC control operation in the list, the Single Inline Write format may be used. Since CAMAC control operations are dataless, the 2nd Word may be filled with any data pattern. Even though the data portion is not used, it must still be included.

Special Instructions

There are several special instructions implemented by the 2115. The only required special instruction that must be used is the HALT instruction. The HALT must be the last list instruction loaded following a valid list.

Other instructions include loading of the DMA Total Transfer Count Register and the Memory Address Registers, and changing the direction of a DMA transfer. Care must be taken when using these instructions to ensure that the host has allocated appropriate data space for the buffers.

The following chart shows the special instructions implemented by the 2115. The hex data shown corresponds to the first 16-bit word of the special instruction (header).

Special Instruction (hex)	Function
8000	Halt
8010	Load Memory Address Register
8011	Load Total Transfer Count Register
8012	Set DMA Direction To One
8013	Set DMA Direction To Zero
8014	Jump To List Address
8015	Write Reply FIFO

Halt Instruction

The Halt instruction has a value of 8000 hex and must be placed at the end of a list sequence. This special instruction informs the list processor to cease processing until retriggered. The following diagram shows the bit pattern for the Halt instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Load Memory Address Register Instruction

The Load Memory Address Register instruction has an opcode of 8010 hex and is followed by a longword that contains the data to be loaded. This instruction loads the Memory Address Register used for Direct Memory Accesses (DMA) through the list processor. When the list processor encounters this instruction, it makes sure that all the current DMA activity is complete before loading the data into the Memory Address Register.

The following shows the format for the Load Memory Address Register instruction.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR 31	MAR 30	MAR 29	MAR 28	MAR 27	MAR 26	MAR 25	MAR 24	MAR 23	MAR 22	MAR 21	MAR 20	MAR 19	MAR 18	MAR 17	MAR 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MAR 15	MAR 14	MAR 13	MAR 12	MAR 11	MAR 10	MAR 9	MAR 8	MAR 7	MAR 6	MAR 5	MAR 4	MAR 3	MAR 2	0	0

MEMORY ADDRESS 31 through 2 is the data to be loaded into the Memory Address Register.

Load Total Transfer Count Register Instruction

The Load Total Transfer Count Register instruction has an opcode of 8011 hex and is followed by a longword that contains the data to be loaded. This instruction loads the Total Transfer Count Register used for Direct Memory Accesses (DMA) through the list processor. When the list processor encounters this instruction, it makes sure that all the current DMA activity is complete before loading the data into the Total Transfer Count Register.

The following shows the format for the Load Total Transfer Count Register instruction.

1st Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

2nd Longword:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TTCR 31	TTCR 30	TTCR 29	TTCR 28	TTCR 27	TTCR 26	TTCR 25	TTCR 24	TTCR 23	TTCR 22	TTCR 21	TTCR 20	TTCR 19	TTCR 18	TTCR 17	TTCR 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TTCR 15	TTCR 14	TTCR 13	TTCR 12	TTCR 11	TTCR 10	TTCR 9	TTCR 8	TTCR 7	TTCR 6	TTCR 5	TTCR 4	TTCR 3	TTCR 2	TTCR 1	TTCR 0

TOTAL TRANSFER COUNT 31 through 0 is the data to be loaded into the Total Transfer Count Register.

Set DMA Direction Instruction

The Set DMA Direction instruction has an opcode of 8012 hex and is used to set the DMA DIRECTION in the Control/Status Register to a one. Setting this bit to a one sets the direction of DMA transfer from the 2115 to computer memory (reads). When the list processor encounters this instruction, it makes sure that all the current DMA activity is complete before setting the direction bit.

The following shows the format for the Set DMA Direction instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Clear DMA Direction Instruction

The Clear DMA Direction instruction has an opcode of 8013 hex and is used to clear the DMA DIRECTION in the Control/Status Register. Clearing this bit to a zero sets the direction of DMA transfer from the computer memory to the 2115 (writes). When the list processor encounters this instruction, it makes sure that all the current DMA activity is complete before clearing the direction bit.

The following shows the format for the Clear DMA Direction instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Jump To List Address Instruction

The Jump to List Address Instruction has an opcode of 8014 hex and is used to transfer list processing control to another address in a list. This command is useful for executing repeated list operations in a timer initiated list application. After the timer initially expires, the list is executed until a halt is found. The next instruction after the halt is a Jump to List Address instruction. When the timer expires the next time, the jump instruction executes returning the list processor to the beginning of the list.

The following shows the format for the Jump To List Address instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CMA 14	CMA 13	CMA 12	CMA 11	CMA 10	CMA 09	CMA 08	CMA 07	CMA 06	CMA 05	CMA 04	CMA 03	CMA 02	CMA 01	CMA 00

Write Reply FIFO Instruction

The Write Reply FIFO Short instruction has an opcode of 8015 hex and is used to write 16-bits of data to the Reply FIFO on the 2115. The second longword associated with this command contains the data to write to the FIFO. The Reply FIFO holds all the data received from the highway during read operations on the highway. If desired, the list processor may be instructed to insert a data pattern into the read data path. Since the 2115 forms 32-bit read data words before transferring the data to the output FIFO, it may be necessary to include a Write Reply FIFO Short instruction in a list to make sure longword alignment is maintained. For example, if a list operation resulted in an odd number of 16-bit words being transferred, the user must include a Write Reply FIFO Short instruction to ensure that the last 16-bit word is not "stuck" in the 2115.

The following diagram shows the format for the Write Reply FIFO Short instruction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
RFD 15	RFD 14	RFD 13	RFD 12	RFD 11	RFD 10	RFD 9	RFD 8	RFD 7	RFD 6	RFD 5	RFD 4	RFD 3	RFD 2	RFD 1	RFD 0

Please refer to Appendix C which contains a list processor instruction summary.

Standard Block Transfer Operations

The 2115 provides four modes of transferring multiple data words to/from a CAMAC crate using Standard Block Transfers. Standard Block Transfers use the conventional command/reply sequence of the Serial Highway for each data word transferred to/from a CAMAC crate. Each block transfer instruction has an associated transfer count. This transfer count specifies the number of 16-bit words that are to be transferred during the block. This initial value is loaded into the List Transfer Count Register (LTCR) when the instruction is processed. As data words are transferred to/from a CAMAC module, this register is incremented until it overflows. When the LTCR overflows, the block transfer is complete.

During Standard Block Transfer Read operations, the initial CAMAC read operation is transmitted onto the highway and the 2115 waits for the reply to this command. Once the reply is received it is examined for the validity of data. Data validity depends on such parameters as the Q-Mode selected and the CAMAC Q and X-responses. If the 2115 receives valid data without any transmission errors, the read data is stored in a 2K x 32 FIFO for transfer to the host. Data from this FIFO can be read by the host using programmed I/O or by DMA. After the data is stored in the FIFO, the List Transfer Count Register (LTCR) is incremented and is checked for an overflow. If the LTCR has not overflowed, another highway read command is transmitted and the 2115 again waits for the reply. This sequence continues until the LTCR overflows or an error is encountered; at which time the DONE bit in the Control/Status Register is set to a one.

When Standard Block Transfer Write operations are executed, the 2115 waits for the CAMAC write to be written from the host into the 2K x 32 write data FIFO. Once write data is available, the 2115 increments the LTCR, forms a command message, transmits it onto the highway and waits for a reply. If the 2115 receives a valid reply message the status is checked for validity. Response validity depends on such parameters as the Q-Mode selected and the returned CAMAC Q and X-responses. If the response is valid, the 2115 checks the LTCR. If the LTCR has overflowed, the DONE bit is set to a one in the CSR and the operation terminates. If the LTCR has not overflowed, another write command message is transmitted onto the highway and the sequence continues. If an error is encountered during the block transfer, the DONE bit is set and the operation ceases.

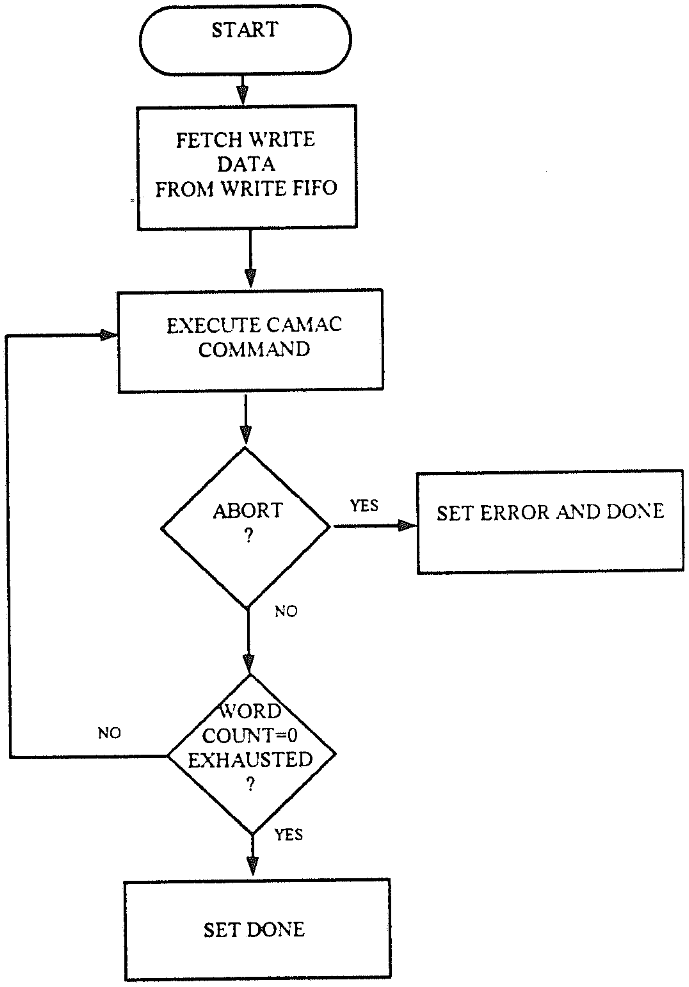
These various Q-Modes include Q-Stop, Q-Ignore, Q-Repeat and Q-Scan. The block transfer data may be transferred to/from the 2115 using programmed transfers or direct memory access (DMA). For programmed transfers, all data transferred to/from the 2115 is under control of the host processor. A faster and more efficient mechanism to transfer CAMAC write/read data is using Direct Memory Access (DMA). Once DMA operations are initialized, all data transfers to/from the 2115 during a block transfer operation are executed autonomously. The host processor merely initiates the operation and the 2115 moves the data to and from the host memory.

Q-Stop Standard Block Transfers

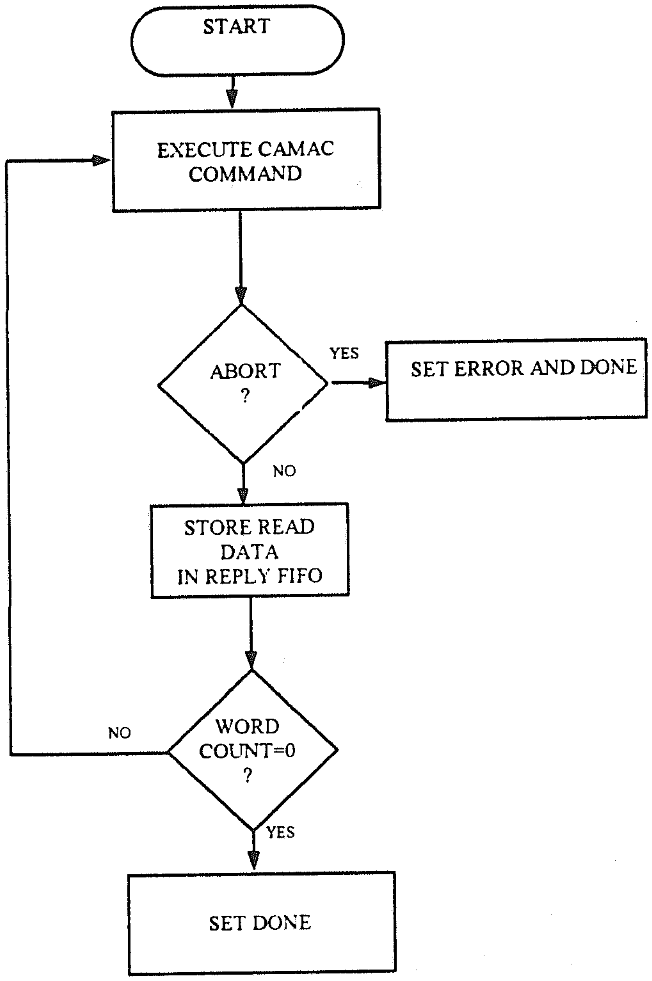
The Q-Stop block transfer mode is selected by setting the QM2 and QM1 to zero when writing the instruction header. During Q-Stop operations, the CAMAC command specified is repeated until a Q-response of zero is received or the transfer count is exhausted. The block transfer also terminates in an error if the an X-response of zero is received with the abort enabled. The following equation describes ERROR for the Q-Stop block transfer mode.

```
ERROR = NO-Q
        # NO-X & !AD
        # ADNR
        # LPE & !AD
        # TPE & !AD
        # TMO
```

The following two diagrams are simplified flow diagrams illustrating Q-Stop block transfer write and read operations.



Q-IGNORE /Q-STOP WRITE COMMANDS



Q-IGNORE /Q-STOP READ COMMANDS

Q-Ignore Standard Block Transfers

The Q-Ignore block transfer mode is selected by setting the QM2 bit to a zero and the QM1 bit to a one when setting up the instruction header. During Q-Ignore operations, the CAMAC command specified is repeated until the transfer count is exhausted. The block transfer also terminates in an error if the an X-response of zero is received with the abort enabled. The following equation describes ERROR for the Q-Ignore block transfer mode.

```
ERROR = NO-X & !AD
        # ADNR
        # LPE
        # TPE
        # TMO
```

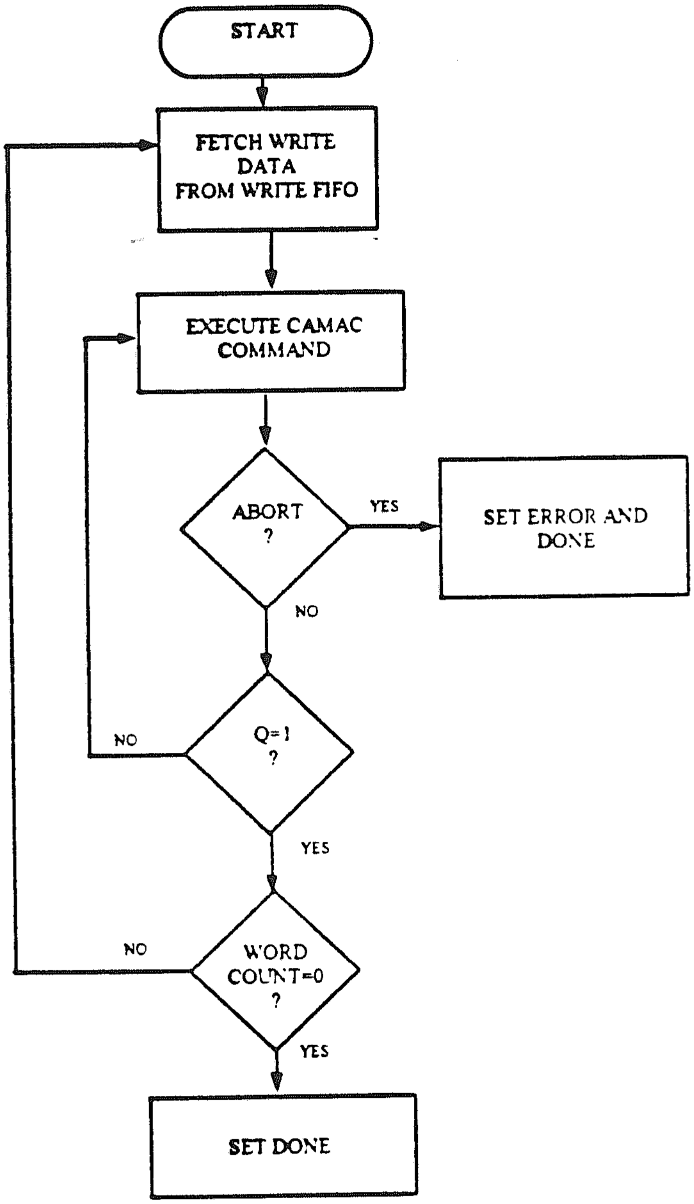
The two diagrams on page 44 are simplified flow diagrams illustrating Q-Ignore transfer write and read operations.

Q-Repeat Standard Block Transfers

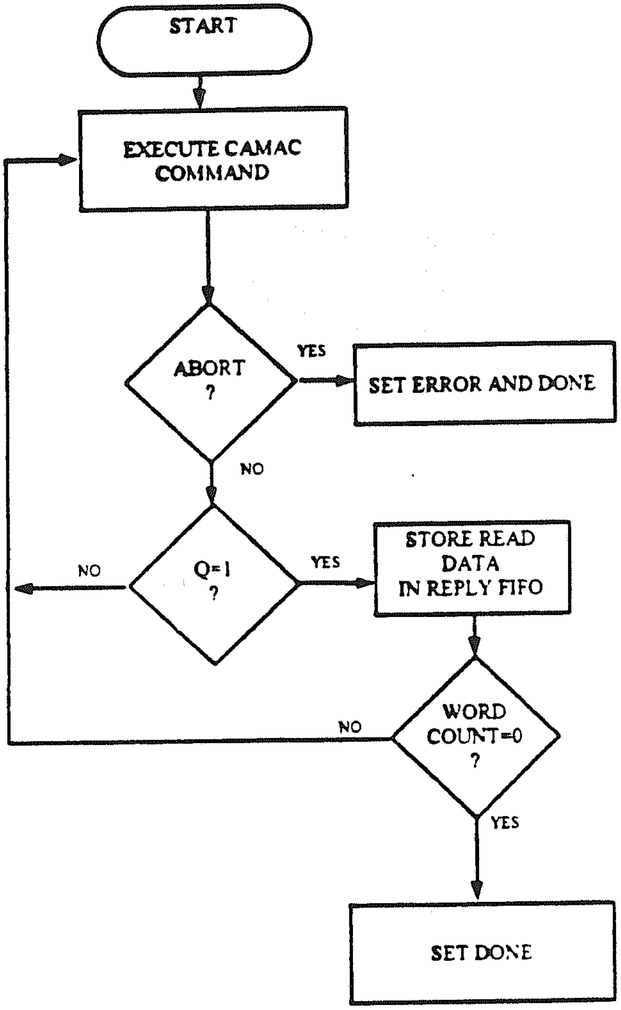
The Q-Repeat block transfer mode is selected by setting the QM2 bit to a one and the QM0 bit to a zero when setting up the instruction header. During Q-Repeat operations, the CAMAC command specified is repeated for each data word until a CAMAC Q-response of one is received. A Q-response of one either causes new write data to be fetched or read data to be stored. the command is repeated for each data word until the transfer count is exhausted. If a Q-response of one is not received within the selected timeout window, an ERROR is set and the block transfer terminates. The block transfer also terminates in an error if the an X-response of zero is received with the abort enabled. The following equation describes ERROR for the Q-Repeat block transfer mode.

```
ERROR = NO-X & !AD
        # ADNR
        # LPE
        # TPE
        # TMO
```

The following diagram is a flow diagram for Q-Repeat block transfer write and read operations.



Q-REPEAT WRITE COMMANDS



Q -REPEAT READ COMMANDS

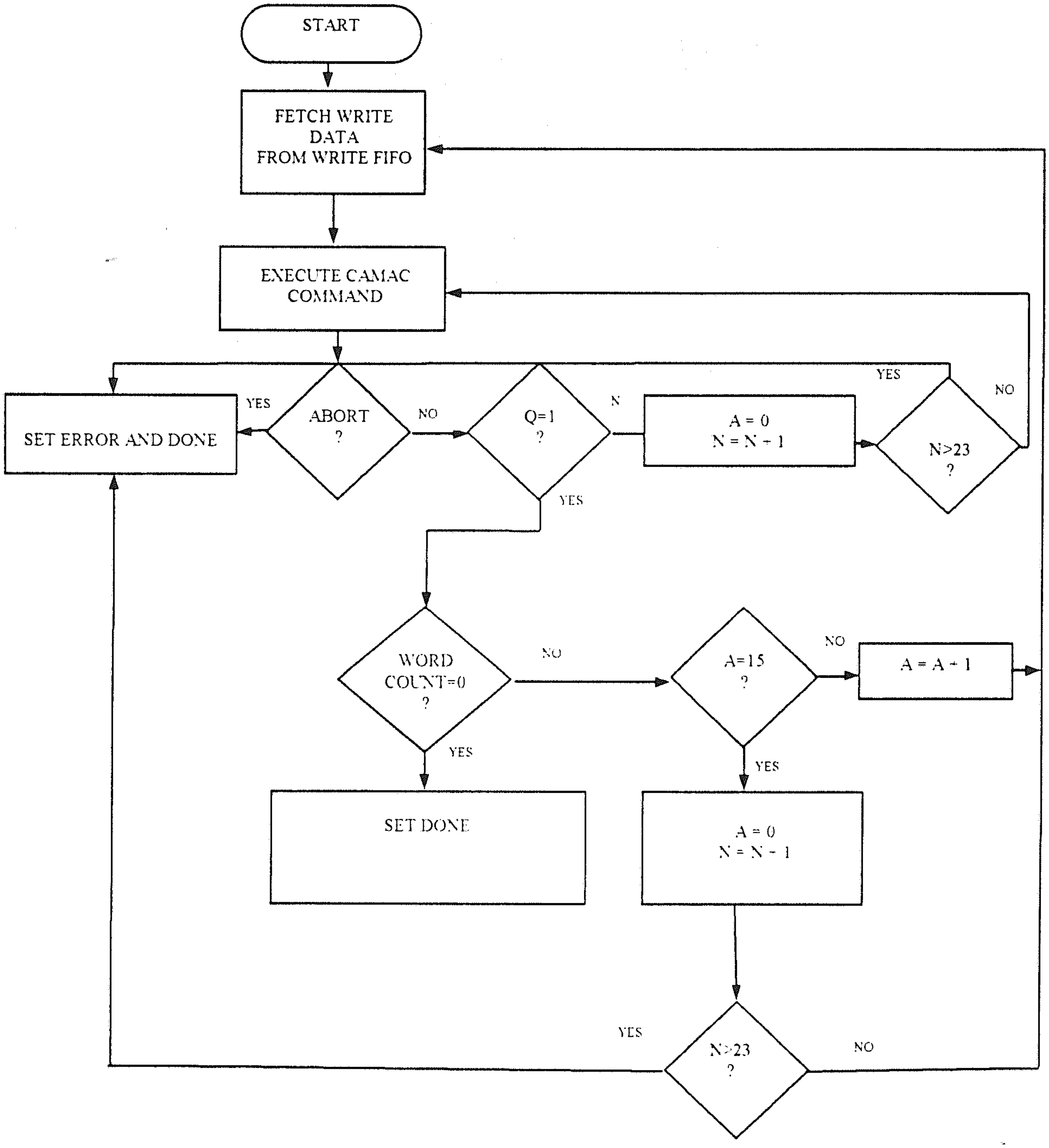
Q-Scan Standard Block Transfers

The Q-Scan block transfer mode is selected by setting the QM2 and QM1 bits to a one when setting up the instruction header. During Q-Scan operations, the 2115 uses the Q-response from the previous operation to determine the station number (N) and subaddress (A) for the next operation. A Q-response of zero indicates that the last valid subaddress of the current station number has been accessed. The 2115 responds to a Q-response of zero by resetting the subaddress and incrementing the station number, and continuing the scan. A Q-response of one indicates that the last command was executed to a valid subaddress. The 2115 responds to the Q-response of one by either storing the read data or fetching the next write data. After a Q-response of one is received, the CAMAC address is incremented as follows: the subaddress is incremented or if the subaddress was 15, it is reset to zero and the station number is incremented.The following flow chart illustrates the updating of the CAMAC station number and subaddress based on the CAMAC Q-response.

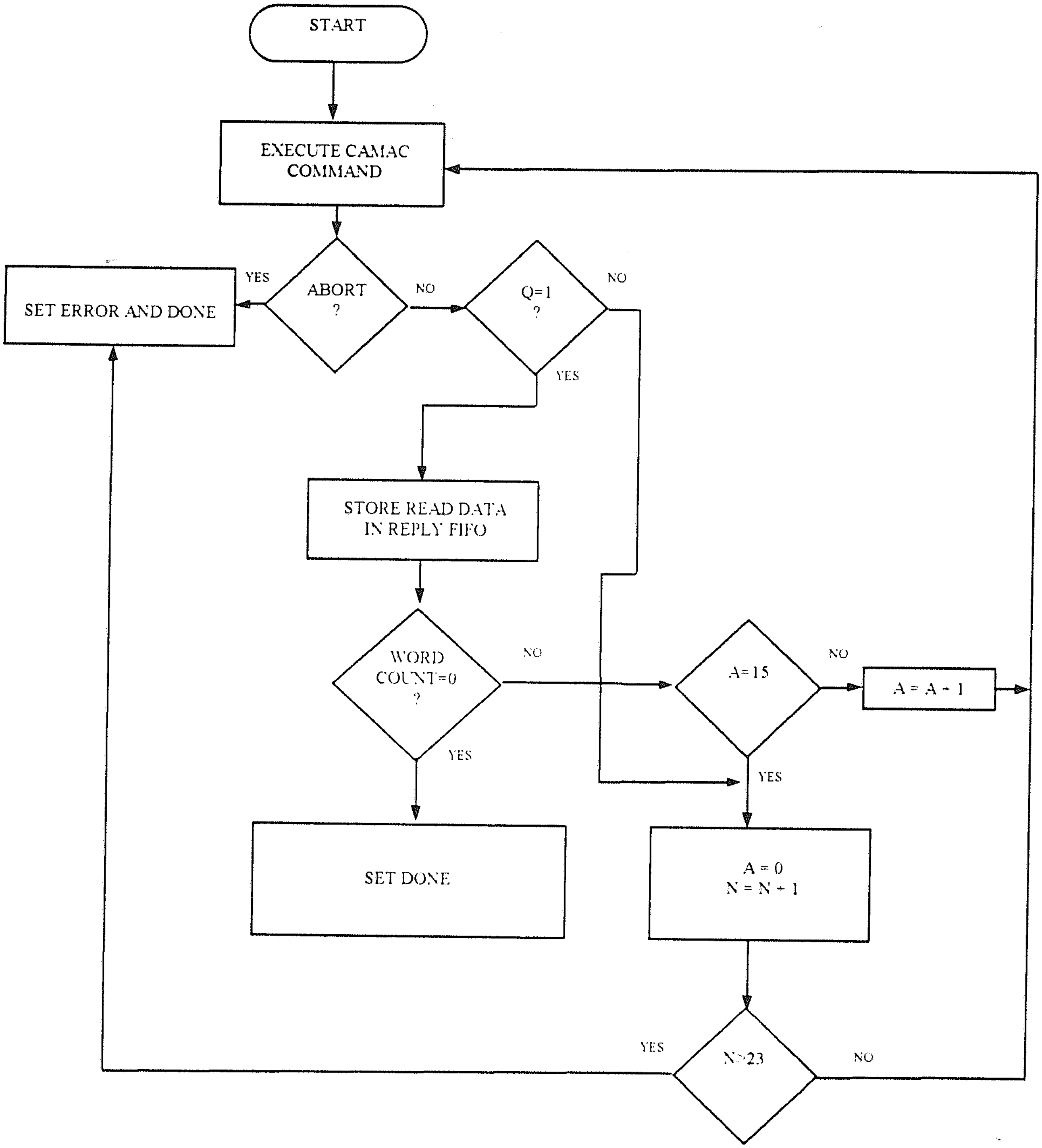
If, due to a programming error, the operation causes the station number to increment beyond station number 23, the block transfer is terminated and the ERROR bit is set. The following equation describes ERROR for the Q-Scan block transfer mode.

```
ERROR = ADNR
        # LPE
        # TPE
        # TMO
        # N>23
```

The following flow diagrams illustrate the Q-Scan write and read operations.



Q-SCAN WRITE OPERATIONS



Q-SCAN READ OPERATIONS

Enhanced Block Transfer Operations

The 2115 provides for five modes of Enhanced Serial Highway Block Transfers. These modes include Single NAF Q-Stop, Single NAF Q-Ignore, Single NAF Q-Repeat (read-only), List Sequence Q-Repeat (read-only), and List Sequence Q-Ignore. The two List Sequence modes require use of the KSC Model 3830-Z1A List Sequencer Module (LSM). When the LSM is in use, the 2115 provides the data transfer mechanism to/from CAMAC and the LSM provides the NAFs for the Dataway in a preloaded memory.

In the Enhanced Serial Highway mode of operation, transfers to/from the CAMAC crate are done in a pipelined fashion. Data transmitted onto or received from the Serial Highway are FIFO buffered. The 2115 provides two 2K x 32 FIFOs. One FIFO is used for CAMAC write operations and the other for CAMAC reads. When an Enhanced Block write operation is executed, the data for the operation may already be contained in the transmit (write) FIFO. In this case, data is transmitted onto the highway at its maximum rate, which is 3 Megabytes-per-second. If during the long block transfers the 2115 cannot maintain the 3 Megabyte data rate, the 2115 inserts "filler" words into the data stream to maintain data flow on the highway. The enhanced crate controllers are able to detect these "filler" words, discard them, and reports this in the status byte of the reply message. The 2115 receiver checks the status bytes returned for each word. If the status byte indicates a "filler" word, both the status and the following data are discarded. If the status byte indicates a valid word, the status byte is stored in the status register and the read data (if a read command) is stored in the read reply FIFO.

Enhanced Serial Highway Command/Reply Sequence

The block write operation on the following page shows the bytes associated with an Enhanced Serial Highway Command/Reply sequence for a CAMAC write operation. This byte stream is a logical extension of the CAMAC standard for single transactions. A key element in the sequence is the CONTROL byte. The two LSBs of the CONTROL byte are decoded by the crate controller and have the following definitions:

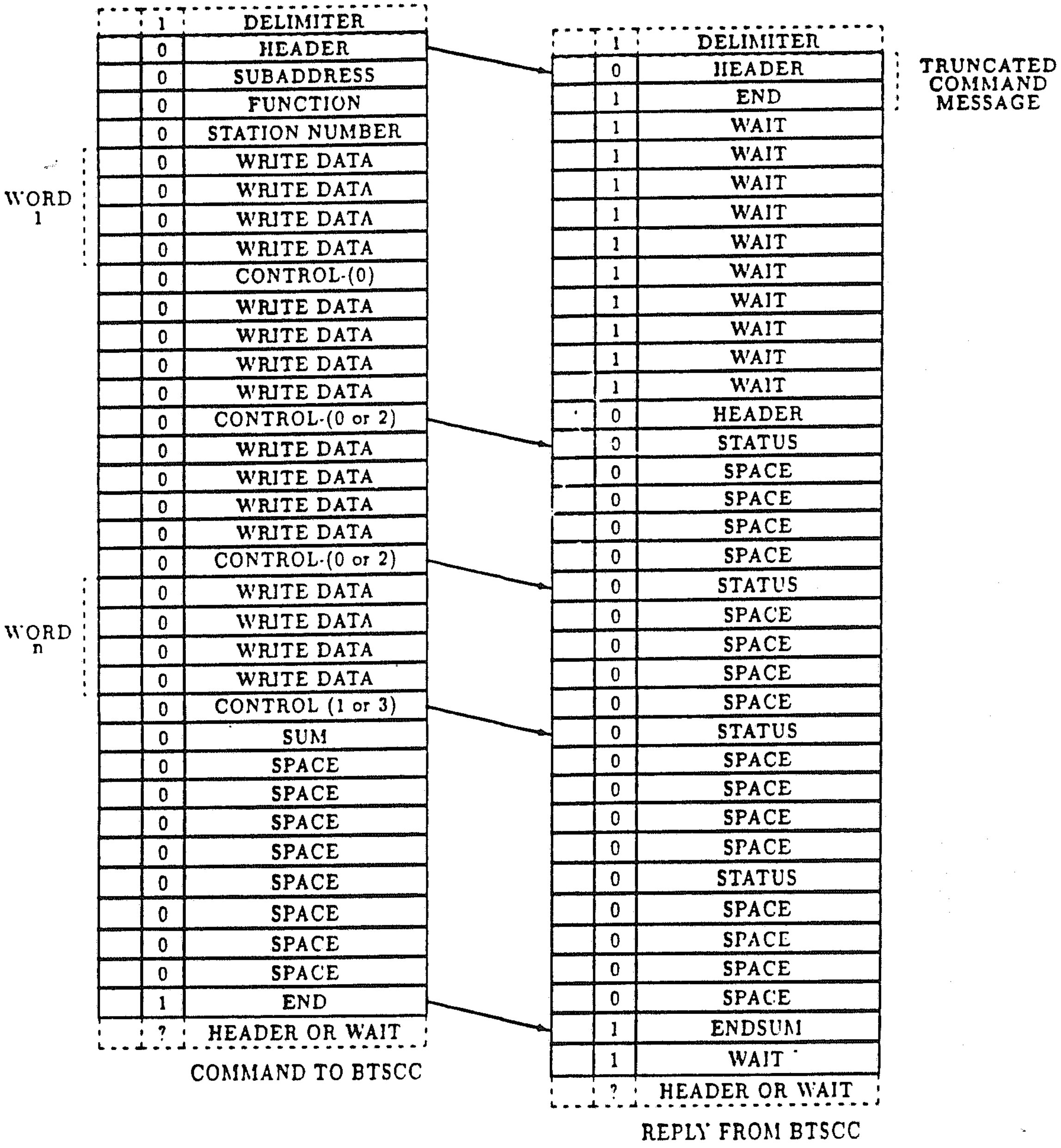
D1	D0	Definition
0	0	The received word is not a "filler" word and it is not the last data word in the block.
0	1	The received word is not a "filler" word and it is the last data word in the block.
1	0	The received word is a "filler" word and is not the last data word in the block.
1	1	The received word is a "filler" word and is the last data word in the block.

During CAMAC write operations, the 2115 inserts "filler" words whenever the transmit (write) data FIFO is empty. As soon as additional CAMAC write data words are loaded into the FIFO, the 2115 stop sending "filler" words and replaces them with actual write data.

During read operations, the 2115 inserts "filler" words whenever the reply (read) FIFO exceeds the half-full indicator so that read reply data already in the reply data stream is not lost. when the crate controller sees these "filler" words, is suspends CAMAC read activity until non-"filler" words resume. As soon as the reply FIFO is less than half-full, the 2115 stop sending "filler" words and the crate controller resumes CAMAC activity.

The status byte in the reply for each data word provides CAMAC Q and X information as well as error information. After the first word in a block is received, bit 5 of the status byte, normally set to a one to indicate a reply, is set to 0 by the crate controller to indicate a "filler" word. If the 2115 receiver encounters a "filler" word, it discards the status byte and the data word.

The following diagram shows the Command/Reply sequence for an Enhanced Serial Highway write operation.



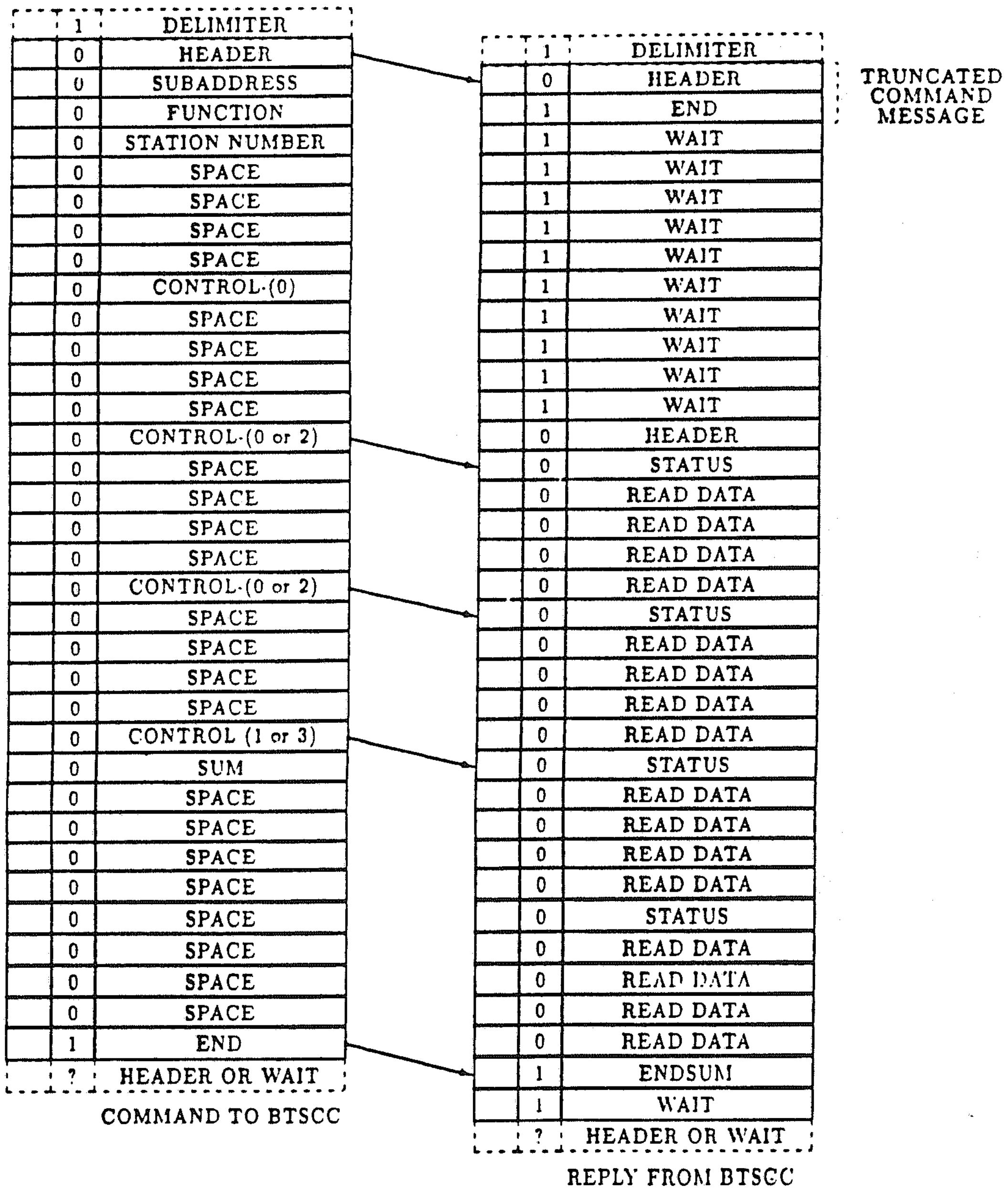
1	DELIMITER
0	HEADER
0	SUBADDRESS
0	FUNCTION
0	STATION NUMBER
0	WRITE DATA
0	WRITE DATA
0	WRITE DATA
0	WRITE DATA
0	CONTROL-(0)
0	WRITE DATA
0	WRITE DATA
0	WRITE DATA
0	WRITE DATA
0	CONTROL-(0 or 2)
0	WRITE DATA
0	WRITE DATA
0	WRITE DATA
0	WRITE DATA
0	CONTROL-(0 or 2)
0	WRITE DATA
0	WRITE DATA
0	WRITE DATA
0	WRITE DATA
0	CONTROL (1 or 3)
0	SUM
0	SPACE
0	SPACE
0	SPACE
0	SPACE
0	SPACE
0	SPACE
0	SPACE
0	SPACE
1	END
?	HEADER OR WAIT

COMMAND TO BTSCC

1	DELIMITER
0	HEADER
1	END
1	WAIT
1	WAIT
1	WAIT
1	WAIT
1	WAIT
1	WAIT
1	WAIT
1	WAIT
1	WAIT
1	WAIT
0	HEADER
0	STATUS
0	SPACE
0	SPACE
0	SPACE
0	SPACE
0	STATUS
0	SPACE
0	SPACE
0	SPACE
0	SPACE
0	STATUS
0	SPACE
0	SPACE
0	SPACE
0	SPACE
0	STATUS
0	SPACE
0	SPACE
0	SPACE
0	SPACE
1	ENDSUM
1	WAIT
?	HEADER OR WAIT

REPLY FROM BTSCC

The block read operation figure on the following page shows an Enhanced Serial Highway Command/Reply Sequence for a CAMAC read operation. This block mode transaction is very similar to the write operation except that groups of four SPACE bytes are provided in the command message to allow space (6-bits per byte) for the read reply data.



COMMAND TO BTSCC

	1	DELIMITER
	0	HEADER
	0	SUBADDRESS
	0	FUNCTION
	0	STATION NUMBER
	0	SPACE
	0	SPACE
	0	SPACE
	0	SPACE
	0	CONTROL-(0)
	0	SPACE
	0	SPACE
	0	SPACE
	0	SPACE
	0	CONTROL-(0 or 2)
	0	SPACE
	0	SPACE
	0	SPACE
	0	SPACE
	0	CONTROL-(0 or 2)
	0	SPACE
	0	SPACE
	0	SPACE
	0	SPACE
	0	CONTROL (1 or 3)
	0	SUM
	0	SPACE
	0	SPACE
	0	SPACE
	0	SPACE
	0	SPACE
	0	SPACE
	0	SPACE
	0	SPACE
	1	END
	?	HEADER OR WAIT

REPLY FROM BTSCC

	1	DELIMITER
	0	HEADER
	1	END
	1	WAIT
	1	WAIT
	1	WAIT
	1	WAIT
	1	WAIT
	1	WAIT
	1	WAIT
	1	WAIT
	1	WAIT
	1	WAIT
	0	HEADER
	0	STATUS
	0	READ DATA
	0	READ DATA
	0	READ DATA
	0	READ DATA
	0	STATUS
	0	READ DATA
	0	READ DATA
	0	READ DATA
	0	READ DATA
	0	STATUS
	0	READ DATA
	0	READ DATA
	0	READ DATA
	0	READ DATA
	0	STATUS
	0	READ DATA
	0	READ DATA
	0	READ DATA
	0	READ DATA
	1	ENDSUM
	1	WAIT
	?	HEADER OR WAIT

TRUNCATED COMMAND MESSAGE

The following describes the five Enhanced Serial Highway Transfer Modes.

Q-Stop Enhanced Single NAF Block Transfers

The Q-Stop block transfers are selected by setting the QM2 and QM1 bits to zero when setting up the instruction header. During Q-Stop block transfers, the CAMAC command specified is repeated until a Q-response of zero is received or the transfer count is exhausted. The block transfer will also terminate is an error is encountered. Care should be taken when using this mode for CAMAC read operations. Since the enhanced operations are pipelined, several spaces for read operations may be present on the command path from the 2115 when the transfer count expires. This causes the crate controller to execute "extra" read commands in the CAMAC chassis. If this is not acceptable, defer to using Standard Block Transfers.

The following equation describes the ERROR source.

```
ERROR = NO-Q
        # NO-X & !AD
        # ADNR
        # LPE & !AD
        # TPE & !AD
        # TMO
```

Q-Ignore Enhanced Single NAF Block Transfers

The Q-Ignore block transfers are selected by setting the QM2 bit to a zero and the QM1 bit to a one when setting up the instruction header. During Q-Ignore block transfers, the CAMAC command specified is repeated until the transfer count is exhausted. The block transfer will also terminate if an error is encountered. Care should be taken when using this mode for CAMAC read operations. Since the enhanced operations are pipelined, several spaces for read operations may be present on the command path from the 2115 when the transfer count expires. This causes the crate controller to execute "extra" read commands in the CAMAC chassis. If this is not acceptable, defer to using Standard Block Transfers.

The following equation describes the ERROR source.

```
ERROR =  NO-X & !AD
        # ADNR
        # LPE & !AD
        # TPE & !AD
        # TMO
```

Q-Repeat Enhanced Single NAF Block Transfer

The Q-Repeat block transfers are selected by setting the QM2 bit to a one and the QM1 bit to a zero when setting up the instruction header. This mode may only be used for CAMAC read operations. When executing Q-Repeat reads, the 2115 places valid read commands on the Serial

Highway as long as the read reply FIFO is not half-full. When valid read data with a CAMAC Q-response of 1 is returned to the 2115, it is stored in the read reply FIFO and the transfer count is incremented. The block transfer is complete when the transfer count is exhausted or an error occurs. Care should be taken when using this mode for CAMAC read operations. Since the enhanced operations are pipelined, several spaces for read operations may be present on the command path from the 2115 when the transfer count expires. This causes the crate controller to execute "extra" read commands in the CAMAC chassis. If this is not acceptable, defer to using Standard Block Transfers.

The following equation describes the ERROR source.

```
ERROR =  NO-X & !AD
         # ADNR
         # LPE & !AD
         # TPE & !AD
         # TMO
```

Enhanced List Sequencer Modes

List sequencing allows Enhanced Serial Highway operations to take place to a predetermined list of CAMAC commands. This is accomplished with the use of the KSC Model 3830-Z1A List Sequencer Module. Each CAMAC data word transferred has a corresponding NAF loaded in the List Sequencer Module (LSM). Before a List Sequencer operation begins, the LSM must be loaded with the CAMAC commands that are to be used during the block transfer. The CAMAC commands contained in any one LSM list must be unidirectional. CAMAC read and write operations cannot be mixed within a single list. CAMAC control commands may reside in any list as long as data buffer space is allocated to them.

There are four lists available within the LSM, two read and two write. Each of these lists can contain up to 512 CAMAC commands. It is also possible to load the LSM in such a way that only two lists are available, one for reads and one for writes. In this case, each list may contain up to 1024 commands.

List 0 and List 1 of the LSM contain sets of CAMAC read NAFs to be executed whereas List 2 and List 3 contain sets of CAMAC write NAFs to be executed. When loading the 2115 to execute list sequencing operations, a special CAMAC command must be specified. These special commands inform the 3952 Block Transfer Serial Crate Controller (BTSCC)/LSM combination as to which list to use for the block transfer.

Command	**NAF List**	**Data Direction**
N(30) F(12) A(0)	List 0	CAMAC Read
N(30) F(12) A(1)	List 1	CAMAC Read
N(30) F(12) A(2)	List 2	CAMAC Write
N(30) F(12) A(3)	List 3	CAMAC Write

When the BTSCC receives on of the above commands, it enables the list mode transfers. For the block transfer, the NAF received as part of the command message is ignored and the Dataway NAFs are supplied by the LSM.

Before the list transfer can begin, the 3830 LSM must be loaded.

Programmed I/O Block Transfers

A Block Transfer Mode of operation transfers multiple CAMAC data words to/from a CAMAC crate. These modes provide a more efficient mechanism for transferring a block of data to/from a CAMAC module. This section describes the steps necessary to execute a block transfer operation using programmed I/O transfers to move the data to/from the interface. Another mechanism is also available for transferring data to/from the interface during block transfer operations. This mode is Direct Memory Access (DMA). Using DMA transfers, initial conditions are loaded by programmed I/O and then all subsequent data transfers to/from the interface are executed without host processor intervention. Please refer to the DMA Block Transfer section of this manual for additional information.

During programmed I/O block transfers, the host processor must transfer all data to/from the FIFO Data Register. Care must be taken when writing data to the FIFO or reading data from the FIFO. Before an access attempt is made, the PCI Interface Bus Master Control/Status Register should be read to determine the amount of data contained in the FIFO's. When writing to the FIFO, ensure that there is room available for the FIFO to accept the write data. For FIFO read operations, ensure that the FIFO contains data to be read. Failure to check the FIFO flags before accessing the FIFO Data Register could result in the PCI bus timing out due to a large number of PCI retry and disconnects. It should also be noted that a CAMAC read that results in an error does not deposit any read data into the FIFO.

Programmed Block Transfer Read Operation (F16 = 0 ; F8 = 0)

1.) Load the Command Memory Address with the initial list address.

2.) Load the Commany Memory Data with the NAFs to execute.

3.) Set the Command Memory Address back to the initial address.

4.) Load the Transfer Count Register with the two's complement of the number of transfers to execute.

5.) Load the Control/Status Register (CSR) with GO set to a 1.

6.) Check the Inbound FIFO Flag in the Bus Master CSR to see is CAMAC read data is available for reading.

7.) If CAMAC read data is available, retrieve it from the FIFO Data Register.

8.) Check the DONE bit in the CSR to see if the interface has completed the CAMAC accesses.

9.) If DONE is not set, jump back to step 4.

10.) If DONE is set, check the FIFO flag to ensure all read data has been read from the FIFO Data Register.

11.) Check the ERR, No-X and No-Q bits in the CSR to verify data validity.

Programmed Block Transfer Write Operation (F16 = 1 ; F8 = 0)

1.) Load the Command Memory Address with the initial list address.

2.) Load the Commany Memory Data with the NAFs to execute.

3.) Set the Command Memory Address back to the initial address.

4.) Load the Transfer Count Register with the two's complement of the number of transfers to execute.

5.) Load the Control/Status Register (CSR) with GO set to a 1.

6.) Check the Outbound FIFO Flag in the Bus Master CSR to see if there is room in the FIFO for the CAMAC write data.

7.) If room exists for the write data, write the CAMAC write data into the FIFO Data Register.

8.) Check the DONE bit in the CSR to see if the interface has completed the CAMAC accesses or an error has occurred.

9.) If DONE is not set, jump back to step 4.

10.) If DONE is set check the ERR, No-X and No-Q bits in the CSR to verify data validity.

DMA I/O Block Transfers

A Block Transfer Mode of operation transfers multiple CAMAC data words to/from a CAMAC crate. These modes provide a more efficient mechanism for transferring a block of data to/from a CAMAC module. This section describes the steps necessary to execute a block transfer operation using direct memory access (DMA) to move the data to/from the interface. This has an inherent speed advantage over programmed I/O transfers since the host processor is not involved with the actual moving of CAMAC write/read data from the interface.

When setting up DMA operations, the following PCI Interface Registers are accessed:

1.) Master Write Address Register	Offset 24
2.) Master Write Transfer Count Register	Offset 28
3.) Master Read Address Register	Offset 2C
4.) Master Read Transfer Count Register	Offset 30
5.) Bus Master Control/Status Register	Offset 3C

The Master Write Address Register and Master Write Transfer Count Register are used to specify the DMA initial address and transfer byte count for CAMAC Read operations The WRITE references in the name of these registers refers to the actual direction of DMA, not the CAMAC access. The Master Read Address Register and Master Read Transfer Count Register are used to specify the DMA initial address and transfer byte count for CAMAC Write operations.

The Bus Master Control/Status Register is used to enable/disable the DMA transfers. The DMA transfers should not be enabled until after the GO bit has been set in the Control/Status Register. After the DMA operation and the CAMAC transfers are complete, the DMA transfers should then be disable to prevent inadvertent data transfers to/from the FIFO Data Register for other operations.

To enable DMA transfers for CAMAC write operations, the READ TRANSFER ENABLE bit in the Bus Master Control/Status Register must be set to a one. For CAMAC read operations, the WRITE TRANSFER ENABLE bit in the Bus Master Control/Status Register must be set to a one.

The following sections illustrate the sequences necessary to setup and execute CAMAC block transfer write and read operations using DMA.

DMA Block Transfer Read Operation (F16 = 0 ; F8 = 0)

1.) Load the Command Memory Address with the initial list address.

2.) Load the Commany Memory Data with the NAFs to execute.

3.) Set the Command Memory Address back to the initial address.

4.) Load the Transfer Count Register with the two's complement of the number of transfers to execute.

5.) Load the Master Write Address Register with the initial memory address where CAMAC read data is to be stored.

6.) Load the Master Write Transfer Count Register with the maximum number of bytes that are to be transferred during the block transfer. Note that this is a byte count. Since the 2115 always executed 32-bit DMA operations, this value must be a multiple of 4.

7.) Load the Control/Status Register (CSR) with DMA DIR set to 1, DMA ENA set to 1 and GO set to a 1.

8.) Wait for the DONE bit in the CSR to be asserted.

9.) Check the ERR, No-X and No-Q bits in the CSR to verify data validity.

DMA Block Transfer Write Operation (F16 = 1 ; F8 = 0)

1.) Load the Command Memory Address with the initial list address.

2.) Load the Commany Memory Data with the NAFs to execute.

3.) Set the Command Memory Address back to the initial address.

4.) Load the Transfer Count Register with the two's complement of the number of transfers to execute.

5.) Load the Master Read Address Register with the initial memory address where CAMAC write data is stored.

6.) Load the Master Read Transfer Count Register with the maximum number of bytes that are to be transferred during the block transfer. Note that this is a byte count. Since the 2115 always executed 32-bit DMA operations, this value must be a multiple of 4.

7.) Load the Control/Status Register (CSR) with DMA DIR set to 0, DMA ENA set to 1 and GO set to a 1.

8.) Wait for the DONE bit in the CSR to be asserted.

9.) Check the ERR, No-X and No-Q bits in the CSR to verify data validity.

The following two functions written in 'C' are samples that illustrate setting up and executing DMA block transfers.

The first section shows a routine written to execute CAMAC block transfer read operations.

```
/**********************************************************************/
/*                                          */
/* Filename : dmablkrd.c                              */
/*                                          */
/**********************************************************************/
/* Parameters:

                base1           -       base address register #1
                base2           -       base address register #2
                crate           -       CAMAC chassis to access
                n               -       CAMAC Station Number
                a               -       CAMAC Subaddress
                f               -       CAMAC Function Code
                q               -       returned CAMAC Q-response
                x               -       returned CAMAC X-response
                data            -       CAMAC read data array
                count           -       Transfer count
                qmode           -       Block Transfer Mode (1-4)
                word_size       -       CAMAC Data Word Size
                abort_disable   -       abort on NO-X bit                          */

#define ushort unsigned short
#define ulong unsigned long
unsigned long inpl(unsigned short address);

ushort dma_block_read(ushort base1,
                      ushort base2,
                      ushort crate,
                      ushort n,
                      ushort a,
                      ushort f,
                      ushort *q,
                      ushort *x,
                      ulong huge *data,
                      ulong count,
                      ushort qmode,
                      ushort word_size,
                      ushort abort_disable)
 {
 ushort csr = base2;                      /* define interface register addresses */
 ushort naf = base2 + 0x04;
 ushort tcr = base2 + 0x08;

 ushort fifo = base1 + 0x20;              /* define PCI interface registers      */
 ushort bmcsr = base1 + 0x3c;
 ushort mwar = base1 + 0x24;
 ushort mwtcr = base1 + 0x28;

 ulong  mardata,byte_count,llp,ldata,stmo,csrdata,done;
 ushort sdatalo;
 short err;

 ldata = ((ulong)(crate) << 16L) +          /* form CNAF register data              */
       ((ulong)(n) << 9L) +
       ((ulong)(a) << 5L) +
       ((ulong)(f));
 outl(naf,ldata);                           /* write CNAF register                  */
 ldata = ~(count-1);                        /* form 2's complement of transfer count */
 outl(tcr,ldata);                           /* write transfer count                 */

 ldata = (ulong *)data;                     /* get physical memory address          */
 mardata = ((ldata & 0xffff0000) >> 12) + (ldata & 0xffff);
 outl(mwar,mardata);                        /* load write address register          */
 if (word_size) {                           /* get actual byte counts               */
  byte_count = count*2;
  }
 else {
  byte_count = count*4;
  }
 outl(mwtcr,byte_count);                    /* write transfer count register        */
 ldata = 0x6000000;                         /* data for FIFO reset                  */
 outl(bmcsr,ldata);                          /* reset fifo's                         */
 ldata = 0x400;
 outl(bmcsr,ldata);                         /* enable pci memory reads               */

 ldata = ((ulong)(word_size) << 13L) +      /* form CSR data                        */
       ((ulong)(abort_disable) << 12L) +
       ((ulong)(qmode) << 1) + 1;
 outl(csr,ldata);                           /* write csr                            */
 stmo=0;                                    /* initialize timeout counter           */
 while((stmo != 0x7ffff) && ((inpl(csr) & 0x80) != 0x80)) {    /* wait for done or timeout  */
  stmo++;
  }
 if (stmo == 0x7ffff) {
  printf("\n\nTimed out waiting for DONE ..  dma_block_read\n");
  }
 stmo=0;
 while((stmo != 0x7ffff) && ((inpl(bmcsr) & 0x20) != 0x20)) {          /* wait for all read data to be DMAed */
  stmo++;
  }
 if (stmo == 0x7ffff) {
  printf("\n\nTimed out waiting for FIFO empty ..  dma_block_read\n");
  }
 ldata = 0x6000000;                         /* write data              */
 outl(bmcsr,ldata);                         /* reset fifo's & stop dma */
 csrdata = inpl(csr);                       /* get status and return */
```

```
*q=1;
*x=1;
if ((csrdata & 0x10000) != 0) {                    /* return CAMAC Q and X              */
  *q = 0;
  }
if ((csrdata & 0x20000) != 0) {
  *x = 0;
  }
sdatalo = (ushort)((csrdata & 0x80000000) >> 31);
return (sdatalo);
}
```

This section shows a routine written to execute CAMAC block transfer write operations.

```
/*************************************************************************/
/*                                                                       */
/* Filename : dmablkwt.c                                                 */
/*                                                                       */
/*************************************************************************/
/* Parameters:

                base1           -       base address register #1
                base2           -       base address register #2
                crate           -       CAMAC chassis to access
                n               -       CAMAC Station Number
                a               -       CAMAC Subaddress
                f               -       CAMAC Function Code
                q               -       returned CAMAC Q-response
                x               -       returned CAMAC X-response
                data            -       CAMAC write data array
                count           -       Transfer count
                qmode           -       Block Transfer Mode (1-4)
                word_size       -       CAMAC Data Word Size
                abort_disable   -       abort on NO-X bit                      */

#define ushort unsigned short
#define ulong unsigned long

unsigned long inpl(unsigned short address);

ushort dma_block_write(ushort base1,
                       ushort base2,
                       ushort crate,
                       ushort n,
                       ushort a,
                       ushort f,
                       ushort *q,
                       ushort *x,
                       ulong huge *data,
                       ulong count,
                       ushort qmode,
                       ushort word_size,
                       ushort abort_disable)
 {
 ushort csr = base2;                        /* define interface register addresses */
 ushort naf = base2 + 0x04;
 ushort tcr = base2 + 0x08;

 ushort fifo = base1 + 0x20;                /* define pci interface register addresses */
 ushort bmcsr = base1 + 0x3c;
 ushort mrar = base1 + 0x2c;
 ushort mrtcr = base1 + 0x30;
ulong mardata,byte_count,llp,ldata,stmo,csrdata,done;
ushort sdatalo;
short err;

ldata = ((ulong)(crate) << 16L) +                   /* form CNAF data         */
      ((ulong)(n) << 9L) +
      ((ulong)(a) << 5L) +
      ((ulong)(f));
outl(naf,ldata);                                    /* write cnaf */
ldata = ~(count-1);                                 /* form 2's complement of transfer count */
outl(tcr,ldata);                                    /* write transfer count   */

ldata = (ulong *)data;                              /* get physical buffer address       */
mardata = ((ldata & 0xffff0000) >> 12) + (ldata & 0xffff);
outl(mrar,mardata);                                 /* load read memory address          */
if (word_size) {                                    /* get actual byte count  */
  byte_count = count*2;
  }
else {
  byte_count = count*4;
  }
outl(mrtcr,byte_count);                             /* write read byte transfer count    */
ldata = 0x6000000;                                  /* write data                        */
outl(bmcsr,ldata);                                  /* reset fifo's */

ldata = ((ulong)(word_size) << 13L) +               /* form csr data                     */
      ((ulong)(abort_disable) << 12L) +
      ((ulong)(qmode) << 1L) + 1;
outl(csr,ldata);                                    /* write csr                         */
ldata = 0x4000;                                     /* write data                        */
outl(bmcsr,ldata);                                  /* enable pci memory reads           */
exit=0;
stmo=0;                                             /* wait for done or timeout          */
while((stmo != 0x7ffff) && ((inpl(csr) & 0x80) != 0x80)) {
  stmo++;
  }
if (stmo == 0x7ffff) {
  printf("\n\nTimed out waiting for DONE ..  dma_block_write\n");
  }
ldata = 0x6000000;                                  /* write data                        */
outl(bmcsr,ldata);                                  /* reset fifo's & stop dma           */
csrdata = inpl(csr);                                /* read csr                          */
*q=1;                                               /* return CAMAC Q and X              */
*x=1;
if ((csrdata & 0x10000) != 0) {
  *q = 0;
  }
if ((csrdata & 0x20000) != 0) {
  *x = 0;
  }
sdatalo = (ushort)((csrdata & 0x80000000) >> 31);
return (sdatalo);
}
```

CAMAC List Processing Example

As an example, assume it is desired to read a two channel analog-to-digital converter (ADC) located in chassis number 3 slot 6. 1024 samples of the analog data are to be taken from each channel. The ADC module is accessed by the following commands:

F(2) A(0) - Reads the digitized data. Q-response of 1 indicates valid data.

F(17) A(0) - Selects the channel to read.
Data = 1 selects channel 1.
Data = 2 selects channel 2.

F(24) A(0) - Disables ADC conversions.

F(26) A(0) - Enables ADC conversions.

For this application example, an F(17) A(0) with data = 1 is first executed to select the desired channel. An F(26) A(0) is then executed to enable conversions. A Standard CAMAC Block Transfer operation is executed in the Q-Repeat mode to read the 1024 converted samples. After the samples are read, an F(24) A(0) command is executed to disable the conversions. These steps are then repeated for channel number 2.

For this example, the following commands are executed :

Single Inline CAMAC Write	Crate(3) N(6) F(17) A(0) DATA(1)
Single Inline CAMAC Write	Crate(3) N(6) F(26) A(0) DATA(not required for Control)
CAMAC Block Trasnfer	Crate(3) N(6) F(2) A(0) WORD COUNT(1024)
Single Inline CAMAC Write	Crate(3) N(6) F(24) A(0) DATA(not required for Control)
Single Inline CAMAC Write	Crate(3) N(6) F(17) A(0) DATA(2)
Single Inline CAMAC Write	Crate(3) N(6) F(26) A(0) DATA(not required for Control)
CAMAC Block Transfer	Crate(3) N(6) F(2) A(0) WORD COUNT(1024)
Single Inline CAMAC Write	Crate(3) N(6) F(24) A(0) DATA(not required for Control)

After the list is formed with the above parameters, the actual list is as follows:

Hex Instruction Data	**List Entry Contents**
$0C110368_{16}$	N(6) A(0) F(17) Single Inline Write Crate (3) Q-Ignore 24-Bit
00000001_{16}	DATA(1)
$0C1A0368_{16}$	N(6) A(0) F(26) Single Inline Write Crate (3) Q-Ignore 24-Bit
00000000_{16}	DATA (0)

Hex Instruction Data	List Entry Contents
$0C030330_{16}$	N(6) A(0) F(2) Block Transfer Crate(3) Q-Repeat 24-Bit
$FFFFF800_{16}$	Transfer Count (-1024 * 2)
$0C180368_{16}$	N(6) A(0) F(24) Single Inline Write Crate (3) Q-Ignore 24-Bit
00000000_{16}	DATA (0)
$0C110368_{16}$	N(6) A(0) F(17) Single Inline Write Crate(3) Q-Ignore 24-Bit
00000002_{16}	DATA (2)
$0C1A0368_{16}$	N(6) A(0) F(26) Single Inline Write Crate(3) Q-Ignore 24-Bit
00000000_{16}	DATA (0)
$0C020330_{16}$	N(6) A(0) F(2) Block Transfer Crate(3) Q-Repeat 24-Bit
$FFFFF800_{16}$	Transfer Count (-1024)
$0C180368_{16}$	N(6) A(0) F(24) Single Inline Write Crate(3) Q-Ignore 24-Bit
00000000_{16}	DATA (0)
00008000_{16}	HALT

Timer Initiated List Processing

The 2115 provides an operating mode whereby a list of commands can be repetitively executed at a predetermined rate. This rate is referred to as the Tic Rate. The rate can range from a minimum of 10 microseconds to a maximum of 16.77 seconds.

The Timer Control Register is used to control the Tic Rate. To enable the list to be triggered from the timer, the TIMER ENABLE bit must be set to a one and the CLOCK SOURCE bit must be set to zero in the Timer Control Register. After these bits are set as described, the timer is enabled and list processing is initiated at the predetermined frequency.

The format of the timer initiated list is similar to what has been described for synchronous triggering with one exception. If a single list of instructions is to be executed at the Tic Rate, the list HALT instruction must be followed by JUMP instruction. The jump instruction should direct the list processor to jump to the beginning of the list. When a HALT instruction is encountered in the list, list execution ceases and the List Memory Address (LMA) is incremented which points to the next list instruction. After the first iteration of the list, the LMA is pointing to the JUMP instruction. After the timer expires for subsequent triggers, the list processor sees the JUMP instruction as the first command in the list. The list processor then takes the jump and executes the commands until the HALT instruction is again reached.

Timer Initiated List Processing Example

As a simple example to demonstrate Timer Initiated List Processing, assume a list that resides at list memory location 100_{16} and consists of two Single Inline Write operations. This first single inline is a write to N(1) F(16) A(0) with data of 123456_{16} and the second inline is a write to N(2) F(16) A(0) with data of $ABCDEF_{16}$. All these operations are directed at crate address 1.

The timer initiated list would appear as follows:

List Memory Address	**List Memory Data**	**Instruction Description**
100_{16}	02100168_{16}	C(1)N(1) A(0) F(16) Single Inline Write Q-Ignore 24-Bit
101_{16}	00123456_{16}	DATA(123456_{16})
102_{16}	04100168_{16}	C(1)N(2) A(0) F(16) Single Inline Write Q-Ignore 24-Bit
103_{16}	$00ABCDEF_{16}$	DATA ($ABCDEF_{16}$)
104_{16}	00008000_{16}	HALT
105_{16}	00008014_{16}	JUMP Instruction
106_{16}	00000100_{16}	Command Memory Address Jump Location (100_{16})

Multibuffer Operations

The 2115 may be configured to execute highway read operations automatically and notify the host computer when a buffer a data is available. As the host is reading data from one buffer segment, the 2115 may be filling another. This is referred to as multibuffering.

The highway operations can be executed automatically by enabling the on-board timer through the Timer Control Register. This register controls the frequency at which the list is executed. This rate is referred to as the 'tic' rate. After selecting the frequency in this register, the timer must be enabled.

When multibuffer operations are executed, the read data received from the highway is transferred to the host computer memory via Direct Memory Access (DMA). The DMA operation is initially set up through programmed I/O transfers to the 2115. This establishes the initial DMA address as well as the total number of PCI transfers to executed before reloading the address and transfer count registers. The reloading of the memory address and transfer count registers is enable with the RELOAD ENABLE bit in the Control/Status Registers. When the reloading is enabled, the host computer memory is used as a circular buffer. When the entire DMA buffer has been filled, the 2115 reloads its memory address and transfer count registers.

Before the multibuffer operation is initiated, the Buffer Interval Counter must be loaded with the count indicating the number of PCI transfers that are to occur before setting a multibuffer flag. The number loaded into the Buffer Interval Counter contains the number of 16-bit data transfers to PCI. As an example, assume that the transfer count specification for the 2115 is 1000. This creates a circular buffer in the host memory of 1000 16-bit words (500 32-bit words). The buffer is to be split into four equal segments. Therefore, the Buffer Interval Counter (BIC) is loaded with 1000/4. Every time the BIC is exhausted, a multibuffer flag bit is set. The flag bits may be read through the Multibuffer Control Register or, if enabled, a PCI interrupt may be generated when a flag is set. The flag bits are incrementally set until the transfer count expires and the internal pointer is set to the first flag bit.

When the host sees a flag bit on, it reads out the corresponding segment of data from the buffer and clears the appropriate multibuffer flag by executing a write to the Multibuffer Flag Register. The data for this write must correspond with the flag to clear.

If a flag is set and readout of the buffer segment is not complete by the time the 2115 loops around to the buffer segment again, the FLAG OVERFLOW bit in the Multibuffer Control Register. This is typically caused by the memory buffer being readout too slow.

Append Status

The 2115 may be enabled to append a status word onto the end of a highway read operation by enabling the APPEND STATUS bit in the Control/Status Register. Setting this bit to a one enables the feature and a zero disables the function.

After a read list operation is ended by the list processor encountering a HALT instruction, the APPEND STATUS bit is examined to see if an additional word of data is to be written into the highway read reply FIFO. If the bit is set to a one, the list processor reads a status word from internal registers and writes the data to the read reply FIFO. If append status is used, care must be taken to allocate the additional 16-bit data word in the read data block.

The following diagram shows the bit layout for the returned appended status word. The Control/Status Register section of this manual should be consulted for a detailed description of these bits.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR CD4	ERR CD3	ERR CD2	ERR CD1	0	DERR	TMO	NO SYNC	ERR	N> 23	LPE	TPE	ADNR	STE	NO X	NO Q

External Clock Input

If the clock rate selections on the 2115 do not meet an application requirement for timer initiated list processing, an external clock source may be connected to the 2115. This input is fed into the 2115 through the LEMO connector located on the rear panel. This input provided to the 2115 must be TTL compatible. The list is triggered by the high to low transition of the input, and must have a minimum pulse width of 300 nanoseconds. The actual input circuit consists of a 74LS14 gate with a 180Ω resistor to +5 volts and a 390 Ω resistor to ground.

To enable the external clock input, the Timer Control Register must have both the CLOCK SOURCE and TIMER ENABLE bits set to a one. Refer to the Timer Control Register section of this manual for additional information.

Interrupts

The 2115 may generate interrupts to the PCI bus from any of four sources. These sources include the assertion of the DONE bit in the Control/Status Register, the presence of a Demand Message in the Demand FIFO Register, the PCI interface chip, or by the Buffer Interval Counter overflowing during multibuffer read operations. These sources are individually enabled in the Interrupt Control/Status Register. The level at which the 2115 interrupts the host computer is based on the computer architecture and the mapping of the 2115 PCI interrupt to the host computer bus interrupt. The 2115 interrupts on the PCI bus by asserting the INTA signal.

The DONE interrupt source is generated when a list processing operation is completed and the DONE, bit 7, in the Control/Status Register is set to a one. The DONE INTERRUPT ENABLE, bit 0, in the Interrupt Control/Status Register must be set to a one in order to generate an PCI interrupt.

If enabled, the receipt of a Demand Message from the highway can generate an PCI interrupt. The interrupt is enabled in the Interrupt Control/Status Register using the DEMAND INTERRUPT ENABLE bit, bit 1. When this interrupt source is generated, the entire contents of the Demand FIFO Register should be read in order to allow additional interrupts to occur.

Another interrupt source from the 2115 is generated during multibuffer read operation. During multibuffer read operations, the Buffer Interval Counter is used to indicate when a segment of the read data buffer has been filled. When this occurs, an interrupt may be generated. This interrupt source is enabled using the MULTIBUFFER INTERRUPT ENABLE bit in the Interrupt Control/Status Register. Please refer to the Multibuffer Operations section of this manual for additional information.

The last interrupt source from the 2115 may be generated from the PCI interface chip. To enable this source, the PCI INTERRUPT ENABLE bit in the Interrupt/Control/Status Register must be set to a one. Refer to the PCI Interface Interrupt Control/Status Register section of this manual for additional information on the PCI interface chip interrupts.

Once an interrupt is asserted, it remains asserted until cleared by writing to the Interrupt Control/Status Register with the data that matches the interrupt sources to clear.

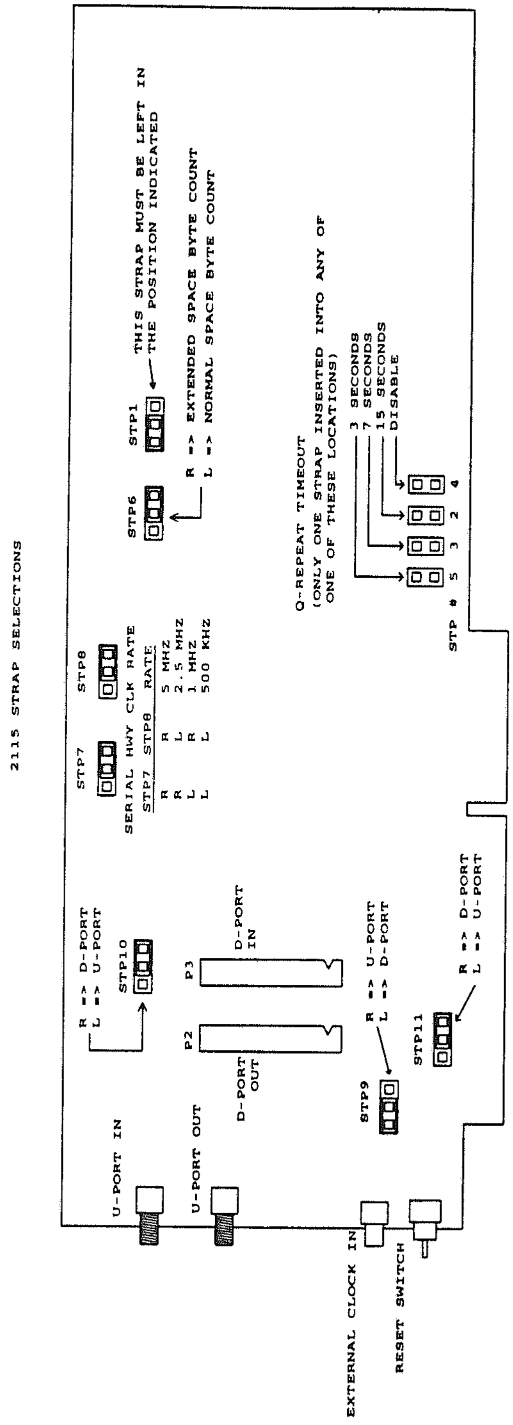
Demands

Demands are asynchronous messages received from devices on the highway. These messages provide a means of informing the 2115 that the device sourcing the message requires attention. The Demand Message contains the chassis address of the SCC requesting service along with an 5-bit identifier. These identifier bits are Serial-graded-LAM bits and are used to determine the source of a demand message within a CAMAC chassis.

After a Demand Message is received by the 2115, it is stored in a 2048 word FIFO. As long as there is at least one demand in the FIFO, the DEMAND PENDING bit in the Control/Status Register is set to a one. After all demand words are read from the FIFO, the bit is reset to a zero.

If the Demand FIFO is full and another Demand Message is received by the 2115, the message is lost and the DEMAND OVERFLOW bit in the Control/Status Register is set to a one. After the overflow occurs, the DEMAND FIFO OVERFLOW bit can be cleared by executing a write to the Demand FIFO Reset Register address.

Appendix A



SERIAL HWY CLK RATE

STP7	STP8	RATE
R	R	5 MHZ
R	L	2.5 MHZ
L	R	1 MHZ
L	L	500 KHZ

NOTES:

R INDICATES STRAP LOADED ONTO THE RIGHMOST STRAP POSTS
L INDICATES STRAP LOADED ONTO THE LEFTMOST STRAP POSTS

U-PORT / D-PORT OPTION STRAPS CONFIGURED AT THE FACTORY FOR THE APPROPRIATE 2115 OPTION

D-PORT CONNECTOR PINS ONLY AVAILABLE ON THE 2115-Z1A OPTION
U-PORT CONNECTOR PINS ONLY AVAILABLE ON THE 2115-Z1B OPTION

NOTCH IN P2 AND P3 CONNECTORS ARE PIN 1 INDICATORS

DEFAULT STRAP LOCATIONS SHOWN ARE FOR D-PORT OPTION

Appendix B

2115 Register Layout

Control/Status Register: Offset 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR CD4	ERR CD3	ERR CD2	ERR CD1	0	DERR	TMO	NO SYNC	ERR	N> 23	LPE	TPE	ADNR	STE	NO X	NO Q
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	DMD OFL	DMD PND	0	0	0	DONE	RLD ENA	APND STAT	SUSP	DMA DIR	DMA ENA	HWY/ LIST	GO

Interrupt Control/Status Register: Offset 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	INFC INT	MBM INT	DMD INT	DONE INT	INFC I.E.	MBM I.E.	DMD I.E.	DONE I.E.

Timer Control Register Offset 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	CLK SRC	TMR ENA	TCR 23	TCR 22	TCR 21	TCR 20	TCR 19	TCR 18	TCR 17	TCR 16
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TCR 15	TCR 14	TCR 13	TCR 12	TCR 11	TCR 10	TCR 09	TCR 08	TCR 07	TCR 06	TCR 05	TCR 04	TCR 03	TCR 02	TCR 01	TCR 00

Command Memory Address Offset 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
GO	CMA 14	CMA 13	CMA 12	CMA 11	CMA 10	CMA 09	CMA 08	CMA 07	CMA 06	CMA 05	CMA 04	CMA 03	CMA 02	CMA 01	CMA 00

Command Memory Data Offset 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMD 31	CMD 30	CMD 29	CMD 28	CMD 27	CMD 26	CMD 25	CMD 24	CMD 23	CMD 22	CMD 21	CMD 20	CMD 19	CMD 18	CMD 17	CMD 16
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CMD 15	CMD 14	CMD 13	CMD 12	CMD 11	CMD 10	CMD 09	CMD 08	CMD 07	CMD 06	CMD 05	CMD 04	CMD 03	CMD 02	CMD 01	CMD 00

List Transfer Count Register Offset 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LTCR 31	LTCR 30	LTCR 29	LTCR 28	LTCR 27	LTCR 26	LTCR 25	LTCR 24	LTCR 23	LTCR 22	LTCR 21	LTCR 20	LTCR 19	LTCR 18	LTCR 17	LTCR 16
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
LTCR 15	LTCR 14	LTCR 13	LTCR 12	LTCR 11	LTCR 10	LTCR 09	LTCR 08	LTCR 07	LTCR 06	LTCR 05	LTCR 04	LTCR 03	LTCR 02	LTCR 01	LTCR 00

Total Transfer Count Register Offset 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TTCR 31	TTCR 30	TTCR 29	TTCR 28	TTCR 27	TTCR 26	TTCR 25	TTCR 24	TTCR 23	TTCR 22	TTCR 21	TTCR 20	TTCR 19	TTCR 18	TTCR 17	TTCR 16
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TTCR 15	TTCR 14	TTCR 13	TTCR 12	TTCR 11	TTCR 10	TTCR 09	TTCR 08	TTCR 07	TTCR 06	TTCR 05	TTCR 04	TTCR 03	TTCR 02	TTCR 01	TTCR 00

Memory Address Register Offset 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAR 31	MAR 30	MAR 29	MAR 28	MAR 27	MAR 26	MAR 25	MAR 24	MAR 23	MAR 22	MAR 21	MAR 20-	MAR 19	MAR 18	MAR 17	MAR 16
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MAR 15	MAR 14	MAR 13	MAR 12	MAR 11	MAR 10	MAR 09	MAR 08	MAR 07	MAR 06	MAR 05	MAR 04	MAR 03	MAR 02	0	0

Buffer Interval Counter Offset 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BIC 31	BIC 30	BIC 29	BIC 28	BIC 27	BIC 26	BIC 25	BIC 24	BIC 23	BIC 22	BIC 21	BIC 20	BIC 19	BIC 18	BIC 17	BIC 16

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
BIC 15	BIC 14	BIC 13	BIC 12	BIC 11	BIC 10	BIC 09	BIC 08	BIC 07	BIC 06	BIC 05	BIC 04	BIC 03	BIC 02	BIC 01	BIC 00

MultiBuffer Control Register Offset 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	MB ENA	FLG OFL	FLAG 3	FLAG 2	FLAG 1	FLAG 0

Demand FIFO Register Offset 0x28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	SGL 16	SGL 8	SGL 4	SGL 2	SGL 1	0	0	C 32	C 16	C 8	C 4	C 2	C 1

Reset Interface Offset 0x2C

Reset Demand FIFO Register Offset 0x30

Appendix C

List Processor Command Summary

2115 List Processing Command Summary

Single CAMAC Operation

0	0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1
CMO	0	C 32	C 16	C 8	C 4	C 2	C 1	0	TM 2	TM 1	QM 2	QM 1	0	WS 1	AD

Normal CAMAC Block Transfer

0	0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1
CMO	0	C 32	C 16	C 8	C 4	C 2	C 1	0	0	1	QM 2	QM 1	0	WS 1	AD
TTCR 31	TTCR 30	TTCR 29	TTCR 28	TTCR 27	TTCR 26	TTCR 25	TTCR 24	TTCR 23	TTCR 22	TTCR 21	TTCR 20	TTCR 19	TTCR 18	TTCR 17	TTCR 16
TTCR 15	TTCR 14	TTCR 13	TTCR 12	TTCR 11	TTCR 10	TTCR 09	TTCR 08	TTCR 07	TTCR 06	TTCR 05	TTCR 04	TTCR 03	TTCR 02	TTCR 01	TTCR 00

Enhanced CAMAC Block Transfer

0	0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1
CMO	0	C 32	C 16	C 8	C 4	C 2	C 1	0	1	0	QM 2	QM 1	0	WS 1	AD
TTCR 31	TTCR 30	TTCR 29	TTCR 28	TTCR 27	TTCR 26	TTCR 25	TTCR 24	TTCR 23	TTCR 22	TTCR 21	TTCR 20	TTCR 19	TTCR 18	TTCR 17	TTCR 16
TTC R15	TTCR 14	TTCR 13	TTCR 12	TTCR 11	TTCR 10	TTCR 09	TTCR 08	TTCR 07	TTCR 06	TTCR 05	TTCR 04	TTCR 03	TTCR 02	TTCR 01	TTCR 00

Single Inline Write CAMAC Transfer

0	0	N 16	N 8	N 4	N 2	N 1	A 8	A 4	A 2	A 1	F 16	F 8	F 4	F 2	F 1
CMO	0	C 32	C 16	C 8	C 4	C 2	C 1	0	1	1	QM 2	QM 1	0	WS 1	AD
0	0	0	0	0	0	0	0	W 24	W 23	W 22	W 21	W 20	W 19	W 18	W 17
W 16	W 15	W 14	W 13	W 12	W 11	W 10	W 09	W 08	W 07	W 06	W 05	W 04	W 03	W 02	W 01

HALT Instruction

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Load Memory Address Register Instruction

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
MAR 31	MAR 30	MAR 29	MAR 28	MAR 27	MAR 26	MAR 25	MAR 24	MAR 23	MAR 22	MAR 21	MAR 20	MAR 19	MAR 18	MAR 17	MAR 16
MAR 15	MAR 14	MAR 13	MAR 12	MAR 11	MAR 10	MAR 09	MAR 08	MAR 07	MAR 06	MAR 05	MAR 04	MAR 03	MAR 02	0	0

Load Total Transfer Count Register Instruction

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1
TTCR 31	TTCR 30	TTCR 29	TTCR 28	TTCR 27	TTCR 26	TTCR 25	TTCR 24	TTCR 23	TTCR 22	TTCR 21	TTCR 20	TTCR 19	TTCR 18	TTCR 17	TTCR 16
TTCR 15	TTCR 14	TTCR 13	TTCR 12	TTCR 11	TTCR 10	TTCR 09	TTCR 08	TTCR 07	TTCR 06	TTCR 05	TTCR 04	TTCR 03	TTCR 02	TTCR 01	TTCR 00

Set DMA Direction Instruction

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0

Clear DMA Direction Instruction

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1

Jump To List Address Instruction

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	CMA 14	CMA 13	CMA 12	CMA 11	CMA 10	CMA 09	CMA 08	CMA 07	CMA 06	CMA 05	CMA 04	CMA 03	CMA 02	CMA 01	CMA 00

Write Reply Instruction

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RPY 15	RPY 14	RPY 13	RPY 12	RPY 11	RPY 10	RPY 09	RPY 08	RPY 07	RPY 06	RPY 05	RPY 04	RPY 03	RPY 02	RPY 01	RPY 00