Model V510-BA11

ARINC-429 Interface

INSTRUCTION MANUAL

March, 1998

CONTENTS

Features 1
General Description 1
Specifications 2
Ordering Information 2

UNPACKING AND INSTALLATION 3
Logical Address Switches 3
Interrupt Switches 4
Module Insertion 4

FRONT PANEL INFORMATION 4
LEDs 4
Connectors 5

MODULE SETUP 5

CONFIGURATION REGISTERS 6
ID/Logical Address Register 6
Device Type 7
Status/Control Register 7
Offset Register 8
Attribute Register 8
Subclass Register 9

OPERATIONAL REGISTERS 9
Diagnostic Register 10
Interrupt Status/ID Register 10
Read Channel 0 Storage Memory 11
Read Channel 1 Storage Memory 11
Channel 0 Storage Address 11
Channel 1 Storage Address 11
Write ARINC-429 XMIT FIFO 11
Read Protocol Management Control Word 12
Write Protocol Management Control Word 12
Channel 0 Pointer Address 12
Channel 1 Pointer Address 12
Channel 0 Pointer Memory 12
Channel 1 Pointer Memory 12
Read Interrupt Mask Register 12
Write Interrupt Mask Register 12
Read Interrupt Status Register 13
Read Interrupt Request Register 14
Selectively Clear Interrupt Status 14

CONTROL REGISTERS 14
Disable/Clear Memory and Interrupt 15
Reset Channel 0 Pointer Memory 15
Reset Channel 1 Pointer Memory 15
Enable ARINC-429 Activity 15
Disable ARINC-429 Activity 15
Retransmit XMIT FIFO 15
Test ARINC-429 State 15
Test Memory Clear Operation 15

OVERVIEW 15

WORD FORMATS 16
ARINC-429 FORMAT 16
HS-3282 FORMATS 16

TRANSMITTER OPERATION 17

ARINC ACTIVITY 17

RECEIVER OPERATION 18

MESSAGE POINTER MEMORY 19

MEMORY CLEAR OPERATIONS 20

PROTOCOL MANAGEMENT CONTROL WORD 20

BLOCK READS 23

INTERRUPTS 24

APPENDIX A - Registers 25
APPENDIX B - Software Example 31

FIGURES

Figure 1 — V510 Switch Locations 3

TABLES

TABLE 1 - CONFIGURATION REGISTERS - SHORT I/O ADDRESS SPACE 6
TABLE 2 - OPERATIONAL REGISTERS - STANDARD ADDRESS SPACE 9

Warranty

ARINC-429 Interface

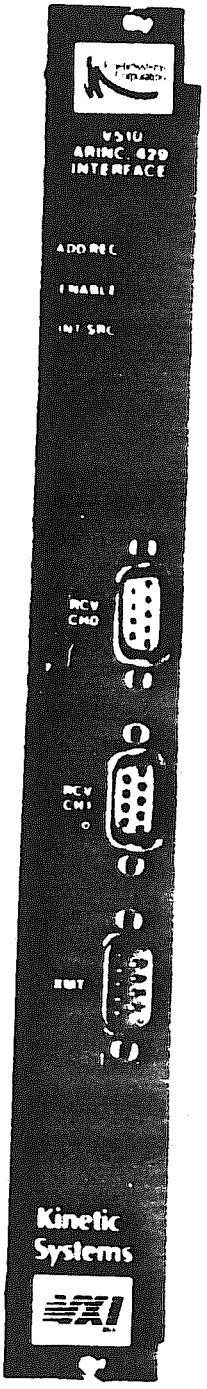
Transmitter and receiver interfaces to ARINC-429 aircraft busses **V510**

Features

- **Two ARINC-429 receive channels**
- **One ARINC-429 transmit channel**
- **Selectable transmit/receive data rates**
- **Computer-selectable list memory associated with each receive channel to read only messages associated with selected labels**
- **Retransmission of FIFO output data**

Typical Applications

- **Aircraft engine test cells**
- **Aircraft subsystem testing**



General Description

The V510 is a single-width, C-size, register-based, VXIbus module that provides an interface for one transmit and two receive channels compatible with the ARINC-429 Aircraft Data Bus (Mark 33 Digital Information Transfer System). ARINC-429 is the primary bus standard for modern civil air transports (such as the Boeing 747). A channel of this system, as specified by Aeronautical Radio, Inc., covers the transmission of avionics information in a digital format. Information is sent from a designated output port over a single shielded, twisted pair of wires to all other system elements having need for that information. Bidirectional data flow on a given twisted pair is not permitted. Transmission is made "open loop" (i.e., receive channels are not required to inform transmit channels that information has been received).

Each transmit and receive channel is completely independent. The channel data rate can be set to 12.5 kbits/s or 100 kbits/s. The transmit channel includes a FIFO, and each receive channel includes a 256 X 32 RAM memory so that data can be efficiently written and read. From a VXIbus perspective, the receive memory is accessed as if it were a FIFO.

The transmit output rate can be selected from 1 ms to 2.048 seconds in a binary fashion (1 ms, 2 ms, 4 ms....). Each ARINC message is generated by two 16-bit Write operations. If the transmit FIFO is written at a rate slower than the selected transmit rate, a message is transmitted at the first "tick" following the availability of data. If a Data Write operation is attempted when the transmit FIFO is full, the data is not accepted. For test purposes, the data at the "bottom" of the FIFO can be repeated at the selected rate.

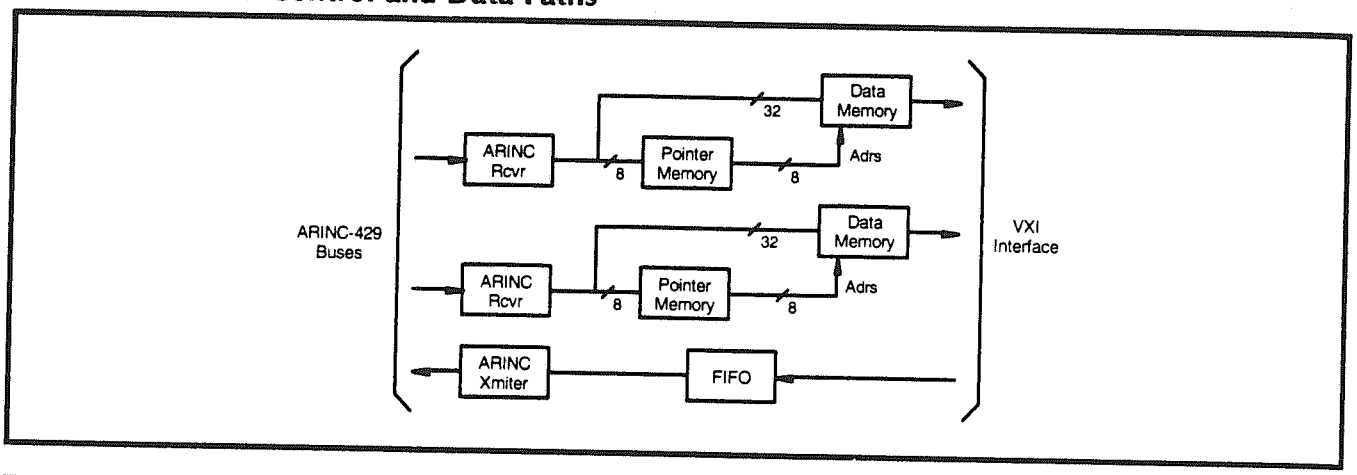
For many applications, reading all messages on a channel will result in the storage of more data than is necessary. The V510 contains a list memory for each channel which selects desired data labels to be read (from the 256 possible combinations). The "list" for each channel is a 256 x 9 memory that can be written from software. Data associated with a label will be the last-received ARINC word, even if no data word associated with that label was received since the last data read operation.

For test purposes, transmit and receive channels can be "looped back" under computer control. This allows testing of the module regardless of the external environment.

Connections to the ARINC-429 Bus are made through three 9-contact "D" type connectors mounted on the front panel of the V510. The receive channels are socket-type connectors, and the transmit channel is a pin-type connector.

The V510 supports both static and dynamic configuration. Access to the data is through memory locations indicated by the Offset Register within the VXIbus Configuration Register set, using A24/A16, D16 data transfers.

V510 ARINC-429 Control and Data Paths



Item	Specification
Number of Channels	One transmit, two receive
Data Format	32-bit ARINC-429
Transmission Rates	12.5 kbits/s or 100 kbits/s
Tansmitter Output Rates	12 rates: 1, 2, 4, 8, 16, 32, 64, 128, 256, and 512 ms; 1.024 and 2.048 s
Transmit Data FIFO Depth	256 32-bit words
Receive Data Memeory Size	256 32-bit words
Data Label List Memory Size	256 9-bit words
Loop Back Control	Programmable on/off
Input/Output Connector Types	
Transmit channel	9P "D"
Receive channels	9S "D"
Mating Connector	
Transmit channel	KineticSystems Model 5930-Z1A
Receive channels	KineticSystems Model 5931-Z1A
Power Requirements	
+5 V	300 mA, typical
Environmental and Mechanical	
Temperature range	
Operational	0°C to +50°C
Storage	-25°C to +75°C
Relative humidity	0 to 85%, non-condensing to +40°C
Cooling requirements	10 CFM
Dimensions	340 mm x 233.35 mm x 30.48 mm (C-size VXIBus)
Front-panel potential	Chassis ground

Ordering Information

Model V510-BA11 ARINC-429 Interface

Related Products

Model 5856-Axyz Cable—9P "D" to Unterminated
Model 5856-Bxyz Cable—9S "D" to Unterminated
Model 5856-Cxyz Cable—9P "D" to 9P "D"
Model 5930-Z1A Connector—9S "D"

UNPACKING AND INSTALLATION

The Model V510 is shipped in an antistatic bag within a styrofoam packing container. Carefully remove the module from its static-proof bag and set the various options to conform to the desired operating environment.

Logical Address Switches

The V510 represents one of the 255 devices permitted in a VXIbus system (Logical Address 0 is reserved for Slot 0 devices). The module is shipped from the factory with it's address set for Logical Address 255. This address can be shared by mutiple devices in a system that supports dynamic configuration. In a system where static configuration of the module is desired, The Logical Address must be manually established. This is accomplished by manipulating the eight rocker switches located under the access hole in the module's right-side ground shield. Refer to FIGURE 1.

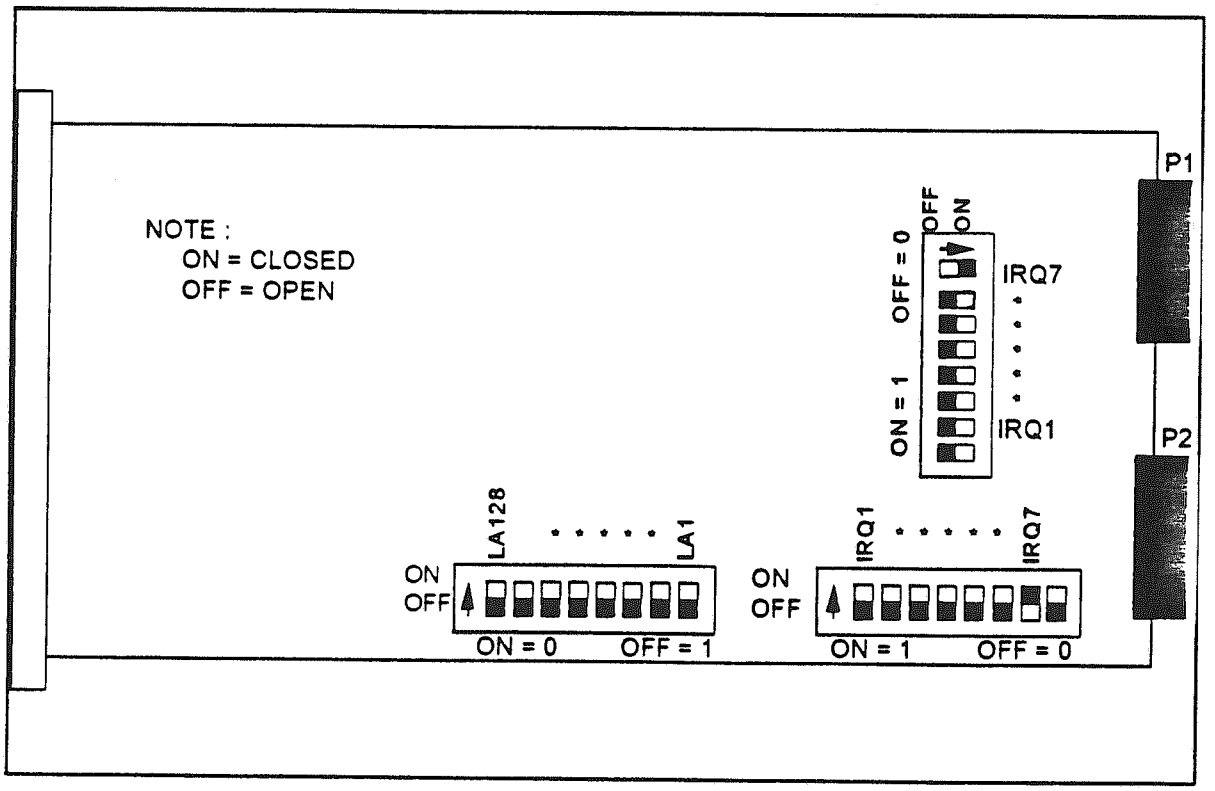


FIGURE 1 — V510 SWITCH LOCATIONS

The eight switches represent a binary combination of numbers that range from zero to 255. Use a scribe or other appropriate instrument to set the Logical Address to the desired value.

Model V510-BA11

The bit pattern for the base address is shown below:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	LA128	LA64	LA32	LA16	LA8	LA4	LA2	LA1	0	0	0	0	0	0

Bits 15 and 14 are set to "1" (VXI defined).
Bits 13 through 6 are user-selectable via the address switches LA128-LA1.
Bits 5 through 0 are set to "0" to indicate a block of 64 bytes.

Interrupt Switches

The V510 has two banks of eight-position switches to select one of seven Interrupt Request levels. Refer to Figure 1 for the switch locations and switch settings. Both banks of eight-position switches must be set to the same values. As shown in Figure 1, IRQ 7 is set to the same position in both banks.

Module Insertion

The V510 is a C-sized, single width VXIbus module. It requires 330 milliamperes of +5 volt power, and 10 cubic feet per minute of air flow to maintain stability. Except for Slot 0, the V510 can be mounted in any unpccupied slot in a C-size VXIbus main frame.

NOTE: TURN MAINFRAME POWER OFF WHEN INSERTING OR REMOVING MODULE

WARNING: REMEMBER TO REMOVE THE INTERRUPT ACKNOWLEDGE DAISY-CHAIN JUMPERS PRIOR TO INSERTING THIS MODULE IN THE BACKPLANE

FRONT PANEL INFORMATION

LEDs

ADD_REC	This LED is illuminated only when one of the operational registers (offsets 12_{16} through $6E_{16}$) is accessed. The operational registers must be enabled by setting bit #15 in the Status/Control register.
ACTIVE	When this LED is on, the V510 is enabled for ARINC activity.
INT SRC	This LED turns on when any of the Interrupt Request bits are set.

Connectors

Three 9-contact "D" type connectors are used to bring the signals to/from the module and the ARINC-429 bus. A pin-type is used for the Transmitter output, and socket type connectors are used for the Receiver inputs (one per channel). All connectors are located on the module's front panel.

Pins 1 and 2 of the 9-pin connectors are used for ARINC bus signal connection. The following chart shows the ARINC bus connections to each 9-pin connector.

RECEIVER 1 INPUT Connector J1		RECEIVER 0 INPUT Connector J2		TRANSMITTER OUTPUT Connector J3	
Pin 1	RCV Data High	Pin 1	RCV Data High	Pin 1	XMT Data High
Pin 2	RCV Data Low	Pin 2	RCV Data Low	Pin 2	XMT Data Low
Pin 3	Ground	Pin 3	Ground	Pin 3	Ground
Pin 4	No Connection	Pin 4	No Connection	Pin 4	No Connection
Pin 5	Ground	Pin 5	Ground	Pin 5	Ground
Pin 6 thru Pin 9	No Connection	Pin 6 thru Pin 9	No Connection	Pin 6 thru Pin 9	No Connection

MODULE SETUP

The V510 Configuration Registers reside in the short I/O address space of the VXIbus and respond to both Address Modifier Codes 29_{16} and $2D_{16}$. The switches LA128 through LA1 determine the base address of the V510 Configuration Registers. The default base address setting is 255, which enables dynamic addressing on the V510. Also, the default Interrupt Request level is set to Level #7. If the Base Address or the Interrupt Level need to be changed, refer to the section on Switch Selections (page 3).

In the dynamic mode, the Base Address for the Configuration Registers must be changed. This is done by writing to the ID/Logical Address Register with a value other than 00_{16} or FF_{16}. The formula for a new Base Address is given below:

$$\text{VXIbus\_Address} = (64 \times \text{Logical\_Address}) + 49152$$

where (Logical_Address) is in the range of 1 to 254. The new VXIbus_Address will be in the range of $C040_{16}$ to $FF80_{16}$. Before writing the new Logical_Address, the MODID line must be set to the slot location that the V510 resides in. (A Slot 0 controller is used to select or deselect the MODID lines.) The ID/Logical Address can then be written with the new Logical Address value. The Slot 0 controller should deselect the MODID line once the new Logical Address has been written.

The V510 Operational Registers reside in the Standard Address space of the VXIbus and responds to four Address Modifier Codes; 39_{16}, $3A_{16}$, $3D_{16}$ and $3E_{16}$. To configure and gain access to the Operational Registers, refer to the section on Operational Registers (page 9).

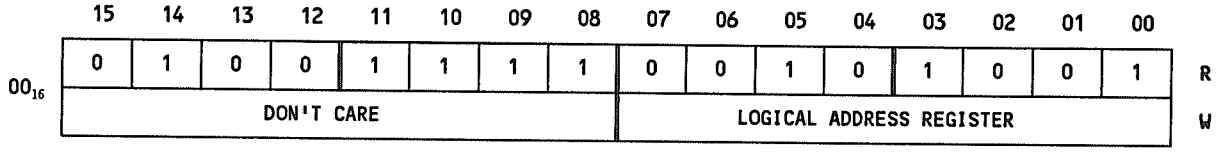
CONFIGURATION REGISTERS

The following six registers occupy the short I/O (A16) address space and conform to the VXI specifications for configuration registers. These registers are normally used once to configure the V510 at power-up. The operational registers are used the remainder of the time. The following chart shows the function, offset from the base Logical Address, and Read/Write capability of each of the Configuration registers within the V510.

TABLE 1 - CONFIGURATION REGISTERS - SHORT I/O ADDRESS SPACE

OFFSET	Write/Read MODE	REGISTER NAME
00_{16}	Write/Read	ID/Logical Address Register
02_{16}	Read Only	Device Type Register
04_{16}	Write/Read	Status/Control Register
06_{16}	Write/Read	Offset Register
08_{16}	Read Only	Attribute Register
$1E_{16}$	Read Only	Subclass Register

ID/Logical Address Register



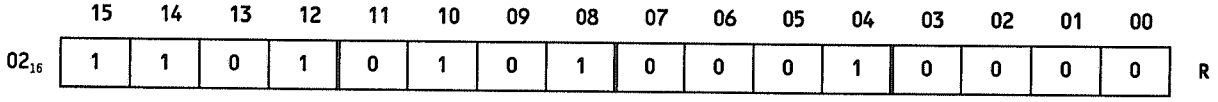
On READ transactions,

Bit	Mnemonic	Description
14,15	Device Class	This is a VXI-defined Extended device.
12,13	Address Space	The V510 uses bot A16 and A24 Address space.
0-11	Manufacturer's ID	3881 ($F29_{16}$) for KineticSystems.

On WRITE transactions,

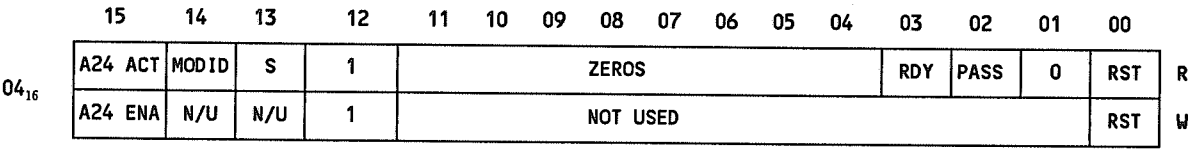
8-15	Don't Care	These bits are ignored on write operations.
0-7	Logical Address	If a dynamically confiured device, this field is written with the new Logical Address.

Device Type



Bit	Mnemonics	Description
12-15	Required Memory	The V510 requires one kilobyte of A24 space memory.
0-11	Model Code	510_{16} for this module.

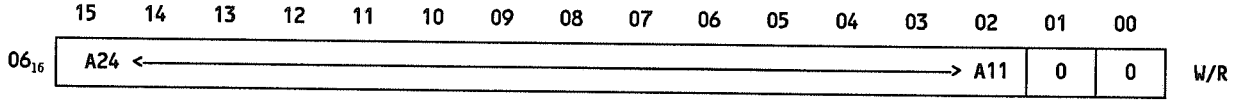
Status/Control Register



Bit	Mnemonics	Description
15	A24	Writing a "1" will enable A24 addressing and allow access to the Operational Registers. Reading a "1" indicates A24 is active. This bit is reset to a "0" on power-up or the assertion of SYSRESET*.
14	MODID	This Read-Only bit is set to a "1" if the module is <u>not</u> selected with the MODID line on VXIbus connector P2. A "0" will indicate that the device is selected by a high state on the P2 MODID line.
13	Status	This Read-Only bit indicates the status of the last operational transaction to the V510. A "1" indicates the transaction completed successfully.
12	BLK	Writing a "0" will enable BLOCK mode. Reading this bit as a "0" will indicate that BLOCK mode is active.
11-4	N/U	Not used. Read as zero.

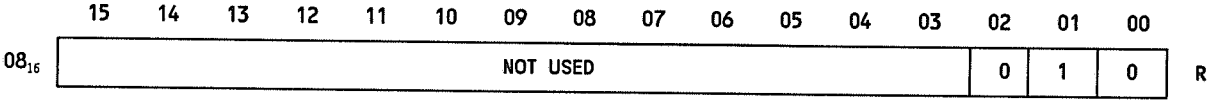
3	RDY	READY. The V510 is always ready. Read as a "1".
2	PASS	PASS. The V510 will always pass self tests. Read as a "1".
1	N/U	NOT USED. Read as a "0".
0	RST	RESET. This Read/Write bit controls the Soft Reset condition within the V510. While the Soft Reset condition is enabled (by writing a "1" to this bit position), any further access to the Operational Registers (see below), except the Diagnostic and Interrupt Status registers, is inhibited. The state of the module is maintained as it was just prior to the Soft Reset being enabled. This bit can be reset by writing a "0", on power-up or the assertion of SYSRESET*.

Offset Register



This 16-bit read/write register defines the base address of the A24 space Operational Registers. The Offset register is reset to an all "0" condition on power-up or the assertion of SYSRESET*, and is written with the appropriate value under program control.

Attribute Register



This Read-only register indicates the interrupt capabilities present on the V510.

Bit	Mnemonics	Description
3-15	Not Used.	Read as zeros.
2	Intr Cntrl	Indicates that the V510 has Interrupt Control capability.
1	Intr Hndlr	Indicates that the V510 is <u>not</u> an Interrupt Handler.
0	Intr Stat	Indicates that the V510 has Interrupt Status capability.

Subclass Register

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
$1E_{16}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	R

This Read-only register further defines the V510 as an Extended device, in accordance with specification VXI-2, Revision 1.0.

Bit 15 = 1 Indicates that this is a VXIbus defined Extended Device
Bits 14-0 = $7FFE_{16}$ Indicates that this is an Extended Register Based Device

OPERATIONAL REGISTERS

The operational registers allow access to — and control of — the functional aspects of the V510. To gain access to these registers, write the Configuration Offset Register to establish the A24 base address. Then, write to the Configuration Status/Control Register with Bit 15 set to a "1". Now the Operational Registers are enabled and set to the desired based address. Any access to the operational registers must be word (D16) transfers. The Operational Registers are described below:

TABLE 2 - V510 OPERATIONAL REGISTERS - STANDARD ADDRESS SPACE

A24 OFFSET	Write/Read MODE	REGISTER NAME
00_{16}	Write/Read	Diagnostic Register
02_{16}	Read Only	Interrupt Status/ID Register
12_{16}	Read Only	Channel 0 Storage Memory
16_{16}	Read Only	Channel 1 Storage Memory
$1A_{16}$	Write Only	Channel 0 Storage Address
$1E_{16}$	Write Only	Channel 1 Storage Address
22_{16}	Write Only	ARINC-429 XMIT FIFO
26_{16}	Read Only	Protocol Management Control
$2A_{16}$	Write Only	Protocol Management Control
$2E_{16}$	Write Only	Channel 0 Pointer Address
32_{16}	Write Only	Channel 1 Pointer Address
36_{16}	Write Only	Channel 0 Pointer Memory
$3A_{16}$	Write Only	Channel 1 Pointer Memory
$3E_{16}$	Read Only	Interrupt Mask Register
42_{16}	Write Only	Interrupt Mask Register
46_{16}	Read Only	Interrupt Status Register
$4A_{16}$	Read Only	Interrupt Request Register
$4E_{16}$	Write Only	Selectively Clear Interrupt Status Bits
52_{16}	Read Only	Disable/clear Memory and Interrupt
56_{16}	Read Only	Reset Channel 0 Pointer Memory
$5A_{16}$	Read Only	Reset Channel 1 Pointer Memory
$5E_{16}$	Read Only	Enable ARINC-429 Activity
62_{16}	Read Only	Disable ARINC-429 Activity
66_{16}	Read Only	Retransmits XMIT FIFO

$6A_{16}$	Read Only	Test ARINC-429 state
$6E_{16}$	Read Only	Test Memory Clear Operation

Diagnostic Register (Offset 00_{16})

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
00_{16}	BLOCK OFFSET								D	S	0	INT ENA	INT SRC	0	0	0	R
00_{16}	BLOCK OFFSET								0	0	0	INT ENA	0	0	0	INIT	W

Bit	Mnemonic	Description
15-8	BLKOFF	Block Offset: These eight bits contain the register offset to be accessed during a DMA (BLOCK) transfer. A value of 12_{16}, 16_{16}, 36_{16}, or $3A_{16}$ would be loaded before a DMA operation is started (see below).
7	Diagnostic	When this bit is set to a "1", the last register access to the operational registers (offsets 12_{16} through $6E_{16}$) was valid.
6	Status	When this bit is set to a "1", the last register access to the operational registers (offsets 12_{16} through $6E_{16}$) was accepted.
5	N/U	Not Used. Ignored on writes, read as "0".
4	INT ENA	Interrupt Enable: Setting this bit to a "1" will enable interrupts.
3	INT SRC	Interrupt Source: When this bit is set to a "1", the V510 has an interrupt pending.
2, 1	N/U	Not Used. Ignored on writes, read as "0"s.
0	INIT	Setting this bit to a "1" will only reset the operational registers (offsets 12_{16} through $6E_{16}$). The configuration registers and the Diagnostic register are unaffected.

Interrupt Status/ID Register (Offset 02_{16})

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
02_{16}	STATUS								ID								R

This is a read-only, 16-bit Interrupt Status Register. During an interrupt acknowledge cycle, this register will output a Status/ID value during a D8, D16, or a D32 data transfer. In a D32

data transfer, the upper 16-bits will be pulled to a logic "1" condition by the backplane termination networks. A read from this register will show the current Status/ID value.

Bit	Mnemonic	Description
15-8	STATUS	These eight bits will indicate a Request True or a Request False condition: Request True = FD_{16} Request False = FC_{16}
7-0	ID	These eight bits represent the Logical Address of the V510 Configuration Registers.

Read Channel 0 Storage Memory (Offset 12_{16})
Read Channel 1 Storage Memory (Offset 16_{16})

These two registers will read half of the ARINC-429 word from channel zero or one. To read the full 32-bit ARINC-429 word, two reads of the register are required. The registers at offsets 26_{16} and $2A_{16}$ must first be written with the storage address before a read to channel ero or one is executed. The Status bit in the diagnostic register should always equal "1".

Channel 0 Storage Address (Offset $1A_{16}$)
Channel 1 Storage Address (Offset $1E_{16}$)

The Channel Storage Address registers are Write-Only registers that are used for specifying the on-board memory addresses from which subsequent reads of the registers at offsests 12_{16} or 16_{16} are executed. The registers at offsets 12_{16} and 16_{16} read Channels 0 and 1 ARINC-429 Half-words respectively. The status bit in the diagnostic register will equal "1". The register format is shown below.

8	7	6	5	4	3	2	1
A7	A6	A5	A4	A3	A2	A1	A0

Write ARINC-429 XMIT FIFO (Offset 22_{16})

This register is used to write ARINC-429 half-words to the transmitter FIFO. Two 16-bit words must be written to make one 32-bit ARINC-429 word. A "0" for the status bit in the diagnostic register indicates that the FIFO is full, and the last write operation was not accepted.

Read Protocol Management Control Word (Offset 26_{16})
Write Protocol Management Control Word (Offset $2A_{16}$)

The Protocol Management Control Words control the module's overall ARINC operating characteristics. See the section on Protocol Management Control Words (page 20) for detailed information on the this register. Once ARINC activity has been enabled in the module, the user is prevented from loading the Control Words. The Status bit in the diagnostic register will equal "0" for any acces to these registers while ARINC activity is enabled.

Channel 0 Pointer Address (Offset $2E_{16}$)
Channel 1 Pointer Address (Offset 32_{16})

The Pointer Address register is used for specifying the address at which subsequent Pointer Memory Load operations are to occur. Before loading the Pointer Memory, the address must be specified. The Pointer Address is written to the register at offset $2E_{16}$ for Channel 0, and at offset 32_{16} for Channel 1. The Pointer Address is automatically incremented after each write to the Pointer Memory registers. This facilitates loading the Pointer Memory, since it is not necessary to reload the address for each entry. Once ARINC activity is enabled, the Pointer Address Load operations are ignored and the status bit in the diagnostic register will equal "0". The format of the Pointer Address register is shown below.

8	7	6	5	4	3	2	1
A7	A6	A5	A4	A3	A2	A1	A0

Channel 0 Pointer Memory (Offset 36_{16})
Channel 1 Pointer Memory (Offset $3A_{16}$)

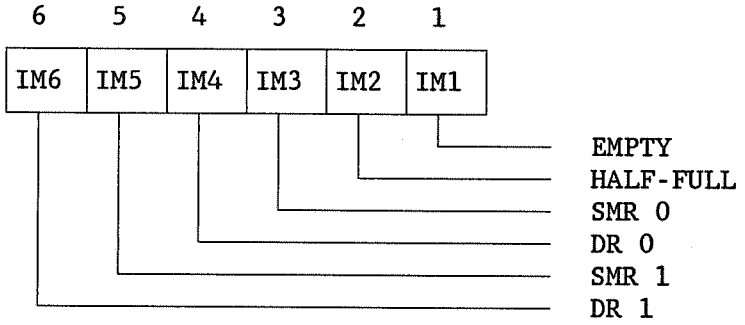
The Pointer Memory is written to the register at offset 36_{16} for channel zero, and at offset $3A_{16}$ for channel one. Once ARINC activity is enabled within the module, these commands are ignored and clear the status bit in the diagnostic register to "0". Refer to the section on the Message Pointer Memory (page 19) for more information. The format of the Pointer Memory data entry is shown below.

9	8	7	6	5	4	3	2	1
SMR	A7	A6	A5	A4	A3	A2	A1	A0

Read Interrupt Mask Register (Offset $3E_{16}$)
Write Interrupt Mask Register (Offset 42_{16})

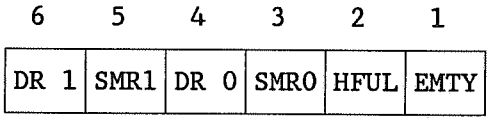
The Interrupt Mask Register is used for specifying which of the six Interrupt sources are to generate an Interrupt Request. If an Interrupt source is to generate an interrupt, it must first be masked "on" in the Interrupt Mask Register. An Interrupt is masked "on" by writing a "1"

into the bit position corresponding to the interrupt source. Each bit position in the Interrupt Mask Register corresponds to the same bit position in the Interrupt Status Register (see below). The Interrupt Mask Register is cleared on power-up and a SYSRESET* cycle. The following diagram shows the bit pattern in the Interrupt Mask Register.



Read Interrupt Status Register (Offset 46_{16})

A read from the register at offset 46_{16} will indicate the Interrupt Status condition of the V510. All bits in the Interrupt Status Register are cleared on power-up, the assertion of a SYSRESET* signal, or a selective-clear operation to the individual bits. The following diagram shows the bit pattern of the Interrupt Status Register.



Bit	Mnemonic	Function
6	DR 1	DATA RECEIVED 1. This bit is set when any message is received on channel 1.
5	SMR 1	SELECTED MESSAGE RECEIVED 1. This bit is set when a selected message is received on channel 1. To enable the SELECTED MESSAGE RECEIVED status bit, the pointer memory must have been previously written with bit 9 set to a "1". (See page 12.)
4	DR 0	DATA RECEIVED 0. This bit is set when any message is received on channel 0.
3	SMR 0	SELECTED MESSAGE RECEIVED 0. This bit is set when a selected message is received on channel 0. To enable the SELECTED MESSAGE RECEIVED status bit, the pointer memory must have been previously written with bit 9 set to a "1". (See page 12.)

2	HFUL	HALF FULL. This bit is set when the transmit FIFO goes less than HALF-FULL. This bit does not represent the static state of the FIFO half-full flag. HALF-FULL is set only when the FIFO makes the transition from a more than half-full to a less than half-full condition. The actual transition is made when the FIFO contains 257 words, and a word is read out of the FIFO and is sent out the transmit port.
1	EMTY	EMPTY. This bit is set when the transmit FIFO goes EMPTY. This bit does not represent the static state of the FIFO's empty flag. EMPTY is set only when the FIFO makes the transition from not-empty to empty (i.e., the last transmit data word has been read from the FIFO).

Read Interrupt Request Register (Offset $4A_{16}$)

The Interrupt Request Register is used for determining the source of an interrupt generated by the V510. An interrupt is generated when an Interrupt Status bit is true AND its associated Interrupt Mask bit is true. The Interrupt Request Register is cleared on power-up, the assertion of a SYSRESET* signal, or a Selective-clear operation to individual Interrupt Status bits. The following diagram shows the bit pattern for the Interrupt Request Register.

6	5	4	3	2	1
DR 1	SMR1	DR 0	SMR0	HFUL	EMTY

Selectively Clear Interrupt Status (Offset $4E_{16}$)

This register is available to selectively clear any Interrupt Status bit, and it will also clear the corresponding Interrupt Request bit. The selective clear operation is performed by writing to the register at offset $4E_{16}$ with write data set equal to the Interrupt Status bit(s) to be cleared (e.g., a write data pattern set to the value 5 clears the SMR0 and EMTY Interrupt Status bits). More than one Interrupt Status bit may be cleared with each write operation, depending on the write data used. The status bit in the Diagnostic register will always equal a "1" after writing this register.

CONTROL REGISTERS

The V510 has eight control registers at offsets 52_{16} through $6E_{16}$. These registers are read-only and return a one-bit status code. This bit has the same meaning as Bit 6 (the Status bit) in the Diagnostic Register. This status code will indicate that the command was accepted or a test condition is true when equal to "1". These eight control registers are described below:

Disable/Clear Memory and Interrupt (Offset 52_{16})

A read from this register offset will disable ARINC-429 activity in the module, clears all memory data and all interrupts. This register will always returns a data value of one.

Reset Channel 0 Pointer Memory (Offset 56_{16})
Reset Channel 1 Pointer Memory (Offset $5A_{16}$)

A read from these register offsets will clear either the Channel 0 or the Channel 1 Pointer Memory Address to zero. This register will always return a data value of one.

Enable ARINC-429 Activity (Offset $5E_{16}$)
Disable ARINC-429 Activity (Offset 62_{16})

A read from these two register offsets will enable or disable ARINC-429 activity in the module. Both of these registers will return a data value of one.

Retransmit XMIT FIFO (Offset 66_{16})

A read from this offset will re-transmit the message in the transmitter FIFO. This register will return a data value of one.

Test ARINC-429 State (Offset $6A_{16}$)

A read from this register will indicate if an ARINC-429 port is active. A data value of one indicates the module is busy.

Test Memory Clear Operation (Offset $6E_{16}$)

This register is used to check the memory-clear operation initiated by reading the register at offset 52_{16}. A data value of one will indicate the clear operation has completed.

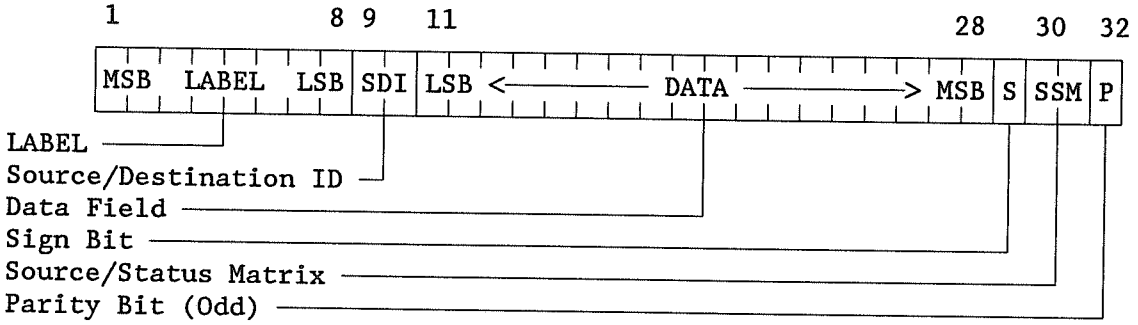
OVERVIEW

The Model V510 is capable of transmitting and receiving single-word messages (discrete, two's complement BNR, and BCD data per the numerical subset of ISO Alphabet No. 5) over the ARINC-429 bus. Multi-word messages containing AIM data (Acknowledgement, ISO Alphabet No. 5 and Maintenance information encoded in dedicated words) or file transfers are not supported.

The bulk of the ARINC-429 protocol management is performed by a Harris HS-3282 Bus Interface Circuit. The 32-bit word formats for both the bit-serial ARINC-429 output and the 16-bit parallel I/O ports of the Harris IC are shown below.

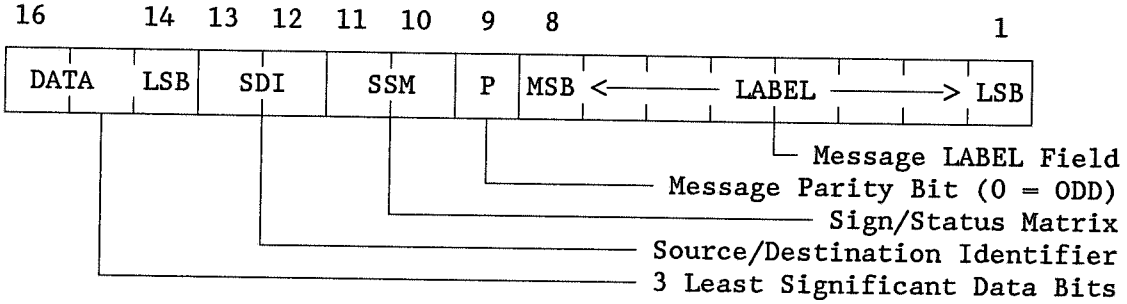
WORD FORMATS

ARINC-429 FORMAT

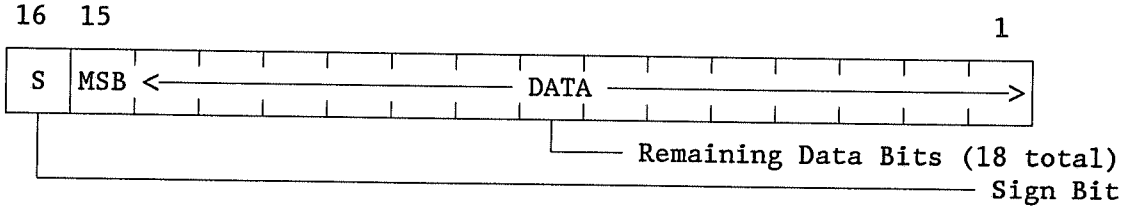


HS-3282 FORMATS

WORD 1



WORD 2



Note that, for the bit-serial ARINC-429 word, the most significant bit of the LABEL field (bit 1) is the first bit transmitted or received, and the parity bit (bit 32) is the last bit transmitted or received. On the parallel port side, bit position 1 is the LSB, and bit position 16 contains the most significant bit of the word. Single word (32-bit) ARINC messages written to or read

from the V510 must be accomplished with two 16-bit VME data transfers as shown above, with Word 1 being written or read first, and Word 2 last.

TRANSMITTER OPERATION

Messages that are to be transmitted over the ARINC-429 bus are stored in a 512-word (1K byte) deep FIFO. This permits the FIFO to contain up to 256 ARINC messages. Since the transmit FIFO is only 16-bits wide, two successive FIFO load operations are necessary to place each 32-bit message into the FIFO. Message halves are loaded into the FIFO by writing to the register at offset 22_{16} (ARINC-429 XMIT FIFO register) in the order shown under the Word Format section of this manual; WORD1 and then WORD2. The status bit in the Diagnostic register will equal a "1" for writes to offset 22_{16}, as long as the FIFO is not full. If the FIFO is full and the write command is executed, a status = 0 response is obtained and the command is ignored.

Once the V510 is enabled for ARINC activity, the transmission of messages begins when the transmission rate timer expires. The transmission rate timer controls the rate at which messages are sent out over the ARINC bus. There are 4 bits in the Protocol Management Control word for specifying the transmission rate. These rates range from one millisecond to 1.024 seconds in a binary progression (...,128mS, 256mS,...). Once the timer expires, two 16-bit words are read from the FIFO and stored in the ARINC interface chip. After the two data words are stored, the ARINC interface chip is enabled for message transmission. Subsequent messages are not transmitted until the transmission rate timer expires. This sequence continues as long as the module is enabled for ARINC activity and there are transmit data words in the FIFO. Should only the first half of the message be in the FIFO when the transmission rate timer expires (i.e., the FIFO goes empty after reading the first word), the transmitter is not enabled until the next transmission rate timer expiration, providing that the rest of the message has been loaded into the FIFO.

A message re-transmit option is also available for generating repetitive messages. This option may only be used after the initial transmission of the FIFO's contents. After the FIFO has been emptied, the re-transmit register (offset 66_{16}) may be read. This causes the FIFO's internal Read pointer to be reset to the beginning of the data list. Once the pointer has been reset, the transmission proceeds as described above.

ARINC ACTIVITY

The V510 will not transmit or receive any ARINC messages unless it is enabled to do so. ARINC activity within the V510 is enabled by reading the register at offset $5E_{16}$, and is disabled by reading the register at offset 62_{16}. Optionally, ARINC activity may be enabled by reading the register offset 66_{16}, which also re-transmits the contents of the transmit data FIFO. Refer to the Transmitter Operation section of this manual for further information regarding the Re-Transmit feature.

RECEIVER OPERATION

The HS-3282 is capable of receiving ARINC-429 messages from two separate and distinct buses, facilitated by two sets of receiver circuitry within the chip. On the VME side of the chip, duplicate circuitry handles messages from either port. For simplicity sake, only a single port's receiver is described.

Received ARINC-429 messages are stored in a 256-word, 32-bit wide Static RAM array. Since the data bus for this array is used both to load data into RAM and to read messages out to the backplane, the entire 32-bit wide message is transferred at once. This avoids the situation where a message readout over the VMEbus contains half old data and half new data (i.e., the same memory location was in the process of being updated while being read).

In response to the assertion of the D/R (Device Ready) flag on the HS-3282, data is gated out (via the EN and SEL lines) to a pair of 16-bit wide registers. Both registers together are then clocked into the storage memory as a 32-bit quantity.

The location in the memory where the messages are stored is determined by the combination of a pointer array and the contents of the message's LABEL field. As the message is transferred to the holding register, the LABEL field is applied to the address lines of a 256 X 9 RAM memory. The lower eight bits of the pointer contents are then used as the address at which the message is stored in the buffer memory. Using this scheme, the user can determine where in the memory array the incoming messages can be stored. Usually, memory allocation would be prioritized such that messages of most interest would be loaded at the front of memory (low order addresses), so that a DMA-type Block Transfer read could most expeditiously retrieve those messages. Optionally, a dedicated location in memory could be designated as a "garbage" location, to collect unwanted messages.

The ninth bit of this pointer array (the MSB) can be used to set an Interrupt source any time a message or messages with a label of interest is received.

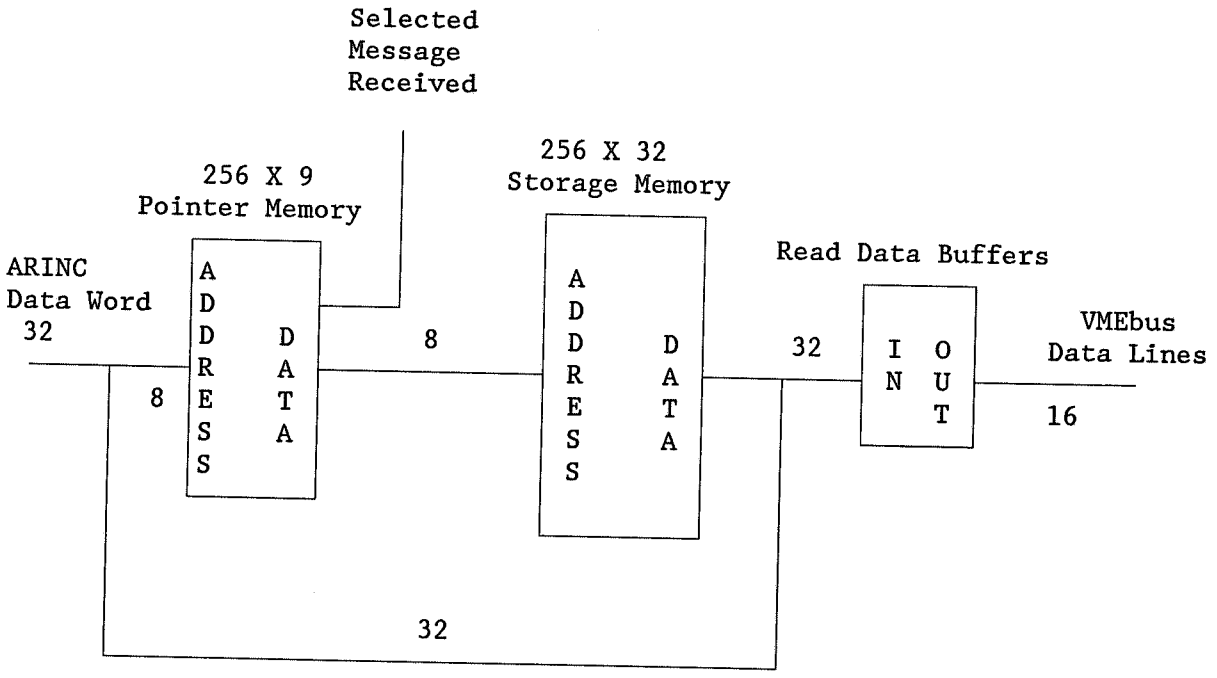
The address register for the pointer memory is loaded by writing to register offset $2E_{16}$ for Channel #0 and 32_{16} for Channel #1, and the memory itself is written to offset 36_{16} for Channel #0 and offset $3A_{16}$ for Channel #1. The pointer memory address is incremented after a write to the pointer memory register. The 8-bit address register is loaded by writing to register offset $1A_{16}$ for Channel #0 and offset $1E_{16}$ for Channel #1, and the memory itself is read by reading register offset 12_{16} for Channel #0 and 16_{16} for Channel #1. Note that two VME read cycles [register offsets 12_{16} or 16_{16}] are required to retrieve an entire ARINC-429 message. Word 1 and Word 2 of the message appear on the VMEbus as described in the Word Format section. The memory address is automatically incremented after a read from register offset 12_{16} for Channel #0 and 16_{16} for Channel #1, allowing a pre-fetch from memory of the next message word for the next read from offsets 12_{16} or 16_{16}. A read from register offset 56_{16} or $5A_{16}$ will reset the memory pointer for channel #0 or channel #1, respectively, so that it points to location 0. A pre-fetch of the first memory location occurs automatically with registers $1A_{16}$, $1E_{16}$, 56_{16} and $5A_{16}$.

Storage memory pre-fetches are also done in response to RAM clear operations. These operations include power-up, a SYSRESET* signal, setting bit one in the Diagnostic register,

or by reading register offset 52_{16}. Thus, if a message is received after a RAM clear operation and ARINC activity has been enabled in the V510, and Storage RAM address location 0 is to contain the message, the first ARINC word readout results in a message containing all zeros. So, if it desired to read all 256 current ARINC messages after a RAM clear operation, the user must execute either a read to the Reset Memory Pointer register or a write to the Channel Memory Address register, which will do a pre-fetch of the current messages.

MESSAGE POINTER MEMORY

The Pointer Memory is a 256 X 9 static RAM which is used for specifying the message storage address within the 256 X 32 Storage RAM. When an ARINC message is received, the LABEL field of the message is applied to the address inputs of the Pointer Memory. The addressed location of the Pointer Memory is read to determine the address of the Storage RAM where the entire ARINC message is to be stored. The following is a simplified block diagram of the message storage technique.



Along with providing the message mapping protocol, the Pointer Memory also allows for the generation of an Interrupt source. Each entry in the Pointer Memory contains a bit for enabling the Interrupt source. When a message is received and the Pointer Memory is read, the ninth bit is checked to see if an Interrupt source is to be generated. If the bit is set, an Interrupt source is asserted. After the message is stored, the Selected Message Received (SMR) bit is set. The Interrupt Status Register may now be read to determine which channel generated the interrupt.

The Pointer Address data is written to the register at offset $2E_{16}$ for Channel #0 and to offset 32_{16} for Channel #1. Once ARINC activity is enabled within the module, these commands are ignored and return a "0" response for the status bit in the diagnostic register. The format of the Pointer Memory data entry is shown below.

9	8	7	6	5	4	3	2	1
SMR	A7	A6	A5	A4	A3	A2	A1	A0

MEMORY CLEAR OPERATIONS

All locations within both the Pointer Memory and the Storage Memory can be cleared by several means. Clearing the RAM refers to writing zeros to all locations in the memory. The memories are cleared by the following:

a.) Power-up cycle
b.) SYSRESET* signal
c.) Reading the register at offset 52_{16} (Clear Memory)
d.) Setting bit #1 (INIT) in the Diagnostic register

The clear operation takes approximately 170 microseconds. A test command to the V510 is used to verify that the clear operation is complete. The test command is a read of the register at offset $6E_{16}$. If the returned data pattern equals zero, the RAM clear operation is still in progress; otherwise, a "1" will confirm the completion of the operation.

After the RAM clear operation has completed, the Pointer Address and Storage Address registers are cleared. Once the addresses are cleared, a pre-fetch of Storage Address location zero is performed. This actually leaves the Storage Address register pointing to Storage RAM location 1, but a read of the Storage RAM [register 12_{16} or 16_{16}] results in the first ARINC message set to all zeros.

PROTOCOL MANAGEMENT CONTROL WORD

Interfacing to the ARINC-429 bus is provided by the Harris HS-3282 interface chip. The clock source for the interface chip is a one Megahertz crystal-controlled oscillator. The setting of the Control Word in the module determines the overall operating characteristics of the ARINC interface. The Control Word is written through the register at offset $2A_{16}$, and is read through the register at offset 26_{16}. Once ARINC activity has been enabled in the module, the user is prevented from loading a new Control Word. The status bit in the diagnostic register will equal "0" for reads of offset $2A_{16}$ if the module is enabled for ARINC activity.

Bits 15 through 5 of the Control Word are contained in the HS-3282 interface chip and control the following characteristics :

a.) ARINC Word Length
b.) Receiver Data Rate

c.) Transmitter Data Rate
d.) Parity
e.) Message Compare Field
f.) Self-Test Mode

Bits 4 through 1 control the rate at which messages are sent out over the ARINC bus. Note that this rate does not control the ARINC data bit-rate, but rather controls the rate at which ARINC messages are loaded into the HS-3282 interface chip before transmission.

All bits in the Control Word are reset to "0" on power-up, SYSRESET*, or setting bit one (INIT) in the diagnostic register. Bits 4 through 1 are used for controlling the counter circuitry which derives the various transmitter message rates. The format of the Control Word is shown in the following diagram.

16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01
WLN	RBR	TBR	PIN	Y1	X1	SD1	Y0	X0	SD0	STM	N/U	XR8	XR4	XR2	XR1

Bit	Mnemonic	Function
15	WLN	WORD LENGTH. This bit is used to specify the length of the ARINC message. WLN is reset to a "0" for the 32-bit message format and set to a "1" for a 25-bit message format.
14	RBR	RECEIVER BIT RATE. This bit is used for specifying the low receiver bit data rate. If RBR is reset to "0", then the receiver bit data rate is 100 Kbps (1 Mhz ÷ 10). If RBR is set to a "1", then the receiver bit data rate is 12.5 Kbps (1 Mhz ÷ 80).
13	TBR	TRANSMITTER BIT RATE. This bit is used for specifying either the high or low transmitter bit data rate. If TBR is reset to "0", then the transmitter bit data rate is 100 Kbps (1 Mhz ÷ 10). If TBR is set to a "1", then the transmitter bit data rate is 12.5 Kbps (1 Mhz ÷ 80).
12	PIN	PARITY. This bit is used to invert the transmitter parity bit for parity error testing. If PIN is reset to "0", normal (odd) parity is selected. If PIN is set to a "1", even parity is selected.
11	Y1	This bit is used along with X1 and SD1, bits 10 and 9 of the Control Word, to enable the reception of selected ARINC messages on receiver channel 1. If SD1 = "1", then the Y1 bit is compared with ARINC data bit 10. If X1 also matches (see X1), the word will be accepted by the receiver. If SD1 = "0", then the comparison is ignored.

0	X1	This bit is used along with Y1 and SD1, bits 10 and 9 of the Control Word, to enable the reception of selected ARINC messages on receiver channel 1. If SD1 = "1", then the X1 bit is compared with ARINC data bit 9. If Y1 also matches (see Y1), the word will be accepted by the receiver. If SD1 = "0", the comparison is ignored.
9	SD1	This bit is used to enable or disable the Source/Destination decoder for receiver channel 1. If this bit is set to "1", received ARINC messages are only stored if ARINC data bit 9 matches the X1 bit (see X1) and ARINC data bit 10 matches Y1 (see Y1). If SD1 is reset to "0", the Source Destination decoder is disabled.
8	Y0	This bit is used along with X0 and SD0, bits 10 and 9 of the Control Word, to enable the reception of selected ARINC messages on receiver channel 0. If SD0 = "1", then the Y0 bit is compared with ARINC data bit 10. If X0 also matches (see X0), the word will be accepted by the receiver. If SD0 = "0", the comparison is ignored.
7	X0	This bit is used along with Y0 and SD0, bits 10 and 9 of the Control Word, to enable the reception of selected ARINC messages on receiver channel 0. If SD0 = "1", then the X0 bit is compared with ARINC data bit 9. If Y0 also matches (see Y0), the word will be accepted by the receiver. If SD0 = "0", the comparison is ignored.
6	SD0	This bit is used to enable or disable the Source/Destination decoder for receiver channel 0. If this bit is set to "1", received ARINC messages are only stored if ARINC data bit 9 matches the X0 bit (see X0) and ARINC data bit 10 matches Y0 (see Y0). If SD0 is reset to "0", the source/ Destination decoder is disabled.
5	STM	SELF-TEST MODE. This bit is used to enable or disable the self-test feature of the HS-3282 interface chip. The self-test mode is enabled by writing this bit to a "0", and disabled by a "1" (normal operation). When enabled for the self-test mode, the interface chip connects the self-test signal from the transmitter directly to the receiver shift registers. This connection bypasses the input receivers. This mechanism provides a means for verifying correct operation of the V510 module without regard for the actual connection to the ARINC bus.

4	N/U	NOT USED. This bit is not used and should be written as a "0" when writing to the Control Word. This bit is read back as a "0".
3 - 0	XR8-XR1	TRANSMITTER RATE 8 thru TRANSMITTER RATE 1 These bits are used to specify the rate at which ARINC messages are loaded into the HS-3282 interface chip and sent out onto the ARINC bus. The binary combination of these bits determines the transmit message rate as shown below :

XR8	XR4	XR2	XR1	MESSAGE INTERVAL
0	0	0	0	1 milliseconds
0	0	0	1	2 milliseconds
0	0	1	0	4 milliseconds
0	0	1	1	8 milliseconds
0	1	0	0	16 milliseconds
0	1	0	1	32 milliseconds
0	1	1	0	64 milliseconds
0	1	1	1	128 milliseconds
1	0	0	0	256 milliseconds
1	0	0	1	512 milliseconds
1	0	1	0	1024 milliseconds
1	0	1	1	
	through			2048 milliseconds
1	1	1	1	

BLOCK READS

To read the memories of the V510 in DMA mode, the Diagnostic register (in the Operational register set) and the Status/Control register (in the Configuration register set) must be set to the proper state, as follows:

1) Write the Diagnostic register (Offset 00_{16}) with the BLOCK OFFSET value of 12_{16} (Channel 0 Storage Memory), 16_{16} (Channel 1 Storage Memory), 36_{16} (Channel 0 Pointer Memory) or $3A_{16}$ (Channel 1 Pointer Memory).

2) Write the BLKENA bit with a "0" in the configuration Status/Control register.

   Once this bit is set, all of the operational registers are disabled except for the register offset written to the Diagnostic register. Any access to the one kilobyte A24 address space of the V510 will only access the Block Offset value stored in the Diagnostic Register.

3) When the specified memory is read, clear the BLKENA bit in the Configuration Status/Control register. The normal operational registers will be enabled again.

VXIbus Block mode may be used to read the memory using Address Modifier Code $3B_{16}$ and $3F_{16}$.

INTERRUPTS

The V510 has the ability to set an interrupt when any one of the conditions in the Interrupt Status register become true.

To enable the V510 for interrupting the VXIbus, the following must be done:

1) Write to the Interrupt Mask register to enable the appropriate interrupt status bits.

2) Enable Interrupts to the VXIbus by writing Bit 4 in the Diagnostic register to a "1".

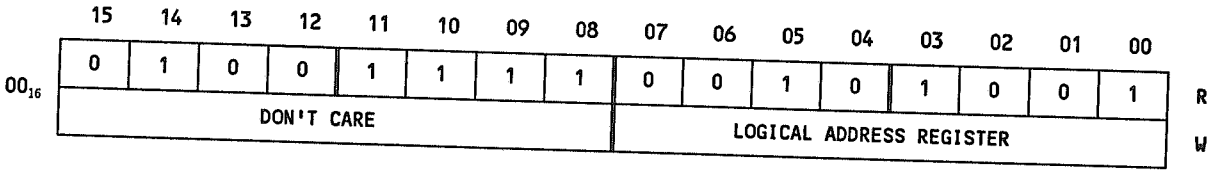
The V510 is now able to cause an interrupt on the VXIbus. Once the V510 sets an interrupt and an interrupt handler reads the Status/ID register, the Enable Interrupt bit in the Diagnostic Register is cleared automatically. Before the INT ENA bit (Bit 4) in the Diagnostic register can be set again, check the DONE INT bit (Bit 3) of the Diagnostic register. If this bit is a "1", read the Interrupt Request register to determine which Interrupt status bit is setting the interrupt. Then selectively clear the appropriate interrupt status bit(s) to clear Bit 3 in Diagnostic register. Otherwise, false interrupts will occur.

APPENDIX A

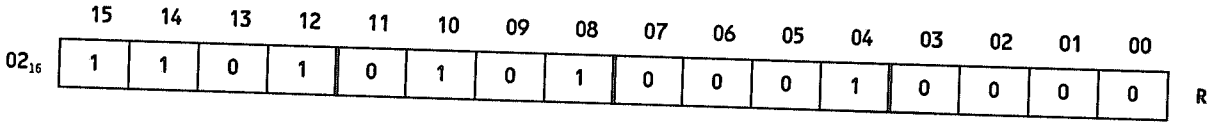
TABLE 1
CONFIGURATION REGISTERS - SHORT I/O ADDRESS SPACE

OFFSET	Write/Read MODE	REGISTER NAME
00_{16}	Write/Read	ID/Logical Address Register
02_{16}	Read Only	Device Type Register
04_{16}	Write/Read	Status/Control Register
06_{16}	Write/Read	Offset Register
08_{16}	Read Only	Attribute Register
$1E_{16}$	Read Only	Subclass Register

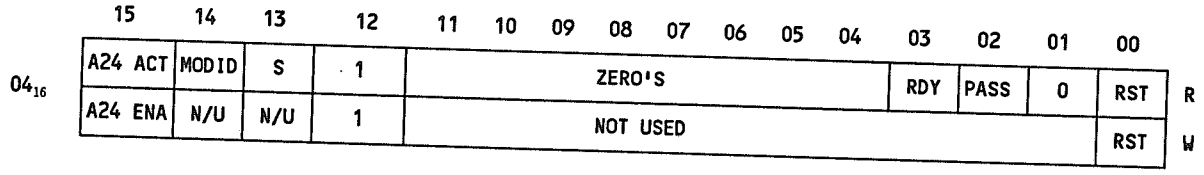
ID/Logical Address Register



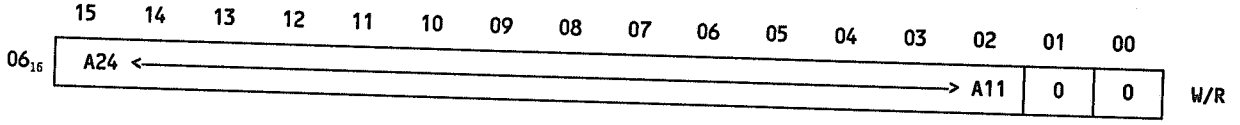
Device Type



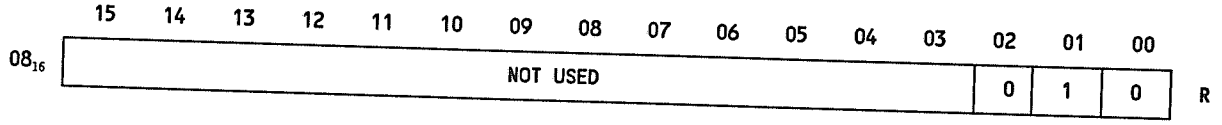
Status/Control Register



Offset Register



Attribute Register



Subclass Register

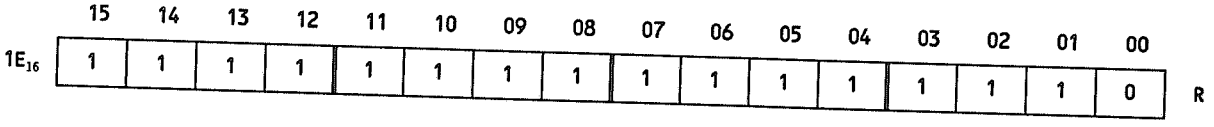


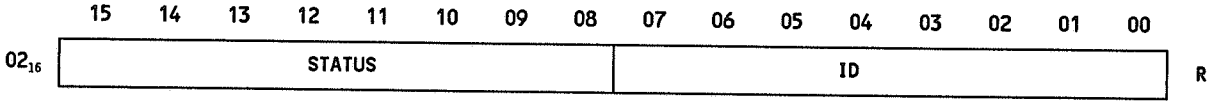
TABLE 2
V510 OPERATIONAL REGISTERS - STANDARD ADDRESS SPACE

A24 OFFSET	Write/Read MODE	REGISTER NAME
00_{16}	Write/Read	Diagnostic Register
02_{16}	Read Only	Interrupt Status/ID Register
12_{16}	Read Only	Channel 0 Storage Memory
16_{16}	Read Only	Channel 1 Storage Memory
$1A_{16}$	Write Only	Channel 0 Storage Address
$1E_{16}$	Write Only	Channel 1 Storage Address
22_{16}	Write Only	ARINC-429 XMIT FIFO
26_{16}	Read Only	Protocol Management Control
$2A_{16}$	Write Only	Protocol Management Control
$2E_{16}$	Write Only	Channel 0 Pointer Address
32_{16}	Write Only	Channel 1 Pointer Address
36_{16}	Write Only	Channel 0 Pointer Memory
$3A_{16}$	Write Only	Channel 1 Pointer Memory
$3E_{16}$	Read Only	Interrupt Mask Register
42_{16}	Write Only	Interrupt Mask Register
46_{16}	Read Only	Interrupt Status Register
$4A_{16}$	Read Only	Interrupt Request Register
$4E_{16}$	Write Only	Selectively Clear Interrupt Status Bits
52_{16}	Read Only	Disable/clear Memory and Interrupt
56_{16}	Read Only	Reset Channel 1 Pointer Memory
$5A_{16}$	Read Only	Reset Channel 2 Pointer Memory
$5E_{16}$	Read Only	Enable ARINC-429 Activity
62_{16}	Read Only	Disable ARINC-429 Activity
66_{16}	Read Only	Retransmits XMIT FIFO
$6A_{16}$	Read Only	Test ARINC-429 state
$6E_{16}$	Read Only	Test Memory Clear Operation

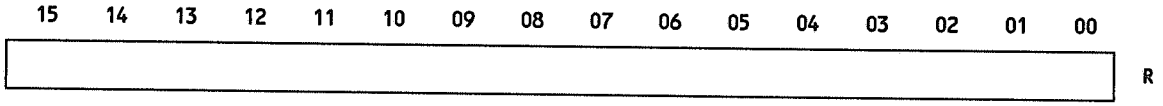
Diagnostic Register

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
00_{16}	BLOCK OFFSET								D	S	0	INT ENA	INT SRC	0	0	0	R
00_{16}	BLOCK OFFSET								0	0	0	INT ENA	0	0	0	INIT	W

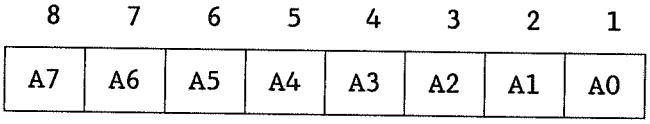
Interrupt Status/ID Register



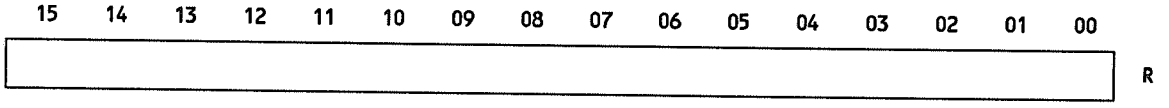
Read Channel 0 Storage Memory (Offset 12_{16})
Read Channel 1 Storage Memory (Offset 16_{16})



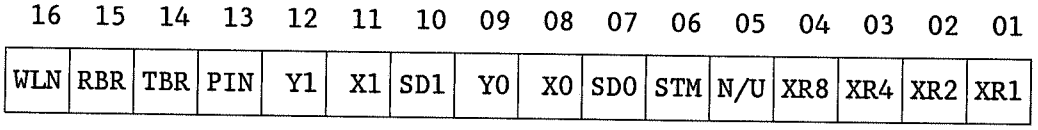
Channel 0 Storage Address (Offset $1A_{16}$)
Channel 1 Storage Address (Offset $1E_{16}$)



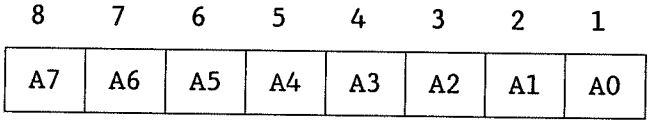
Write ARINC-429 XMIT FIFO (Offset 22_{16})



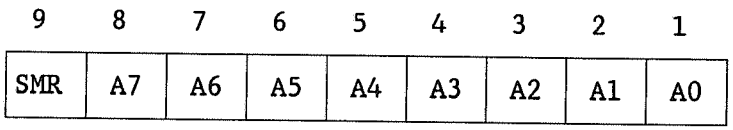
Read Protocol Management Control Word (Offset 26_{16})
Write Protocol Management Control Word (Offset $2A_{16}$)



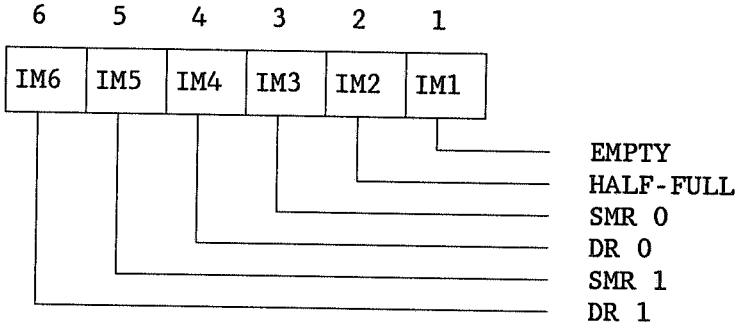
Channel 0 Pointer Address (Offset $2E_{16}$)
Channel 1 Pointer Address (Offset 32_{16})



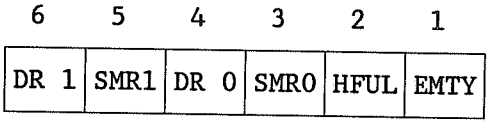
Channel 0 Pointer Memory (Offset 36_{16})
Channel 1 Pointer Memory (Offset $3A_{16}$)



Read Interrupt Mask Register (Offset $3E_{16}$)
Write Interrupt Mask Register (Offset 42_{16})



Read Interrupt Status Register (Offset 46_{16})
Read Interrupt Request Register (Offset $4A_{16}$)
Selectively Clear Interrupt Status (Offset $4E_{16}$)

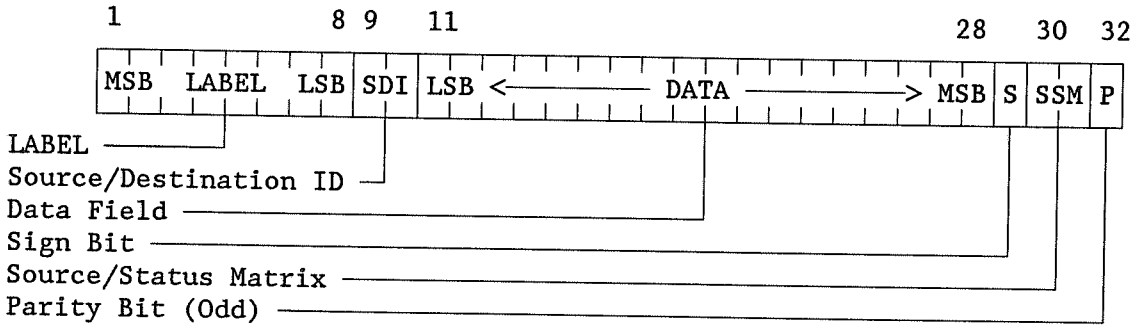


CONTROL REGISTERS

Disable/Clear Memory and Interrupt (Offset 52_{16})
Reset Channel 0 Pointer Memory (Offset 56_{16})
Reset Channel 1 Pointer Memory (Offset $5A_{16}$)
Enable ARINC-429 Activity (Offset $5E_{16}$)
Disable ARINC-429 Activity (Offset 62_{16})
Retransmit XMIT FIFO (Offset 66_{16})
Test ARINC-429 State (Offset $6A_{16}$)
Test Memory Clear Operation (Offset $6E_{16}$)

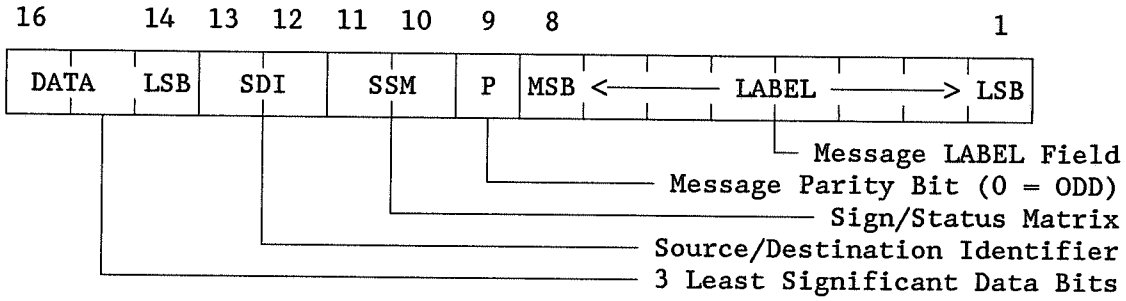
WORD FORMATS

ARINC-429 FORMAT

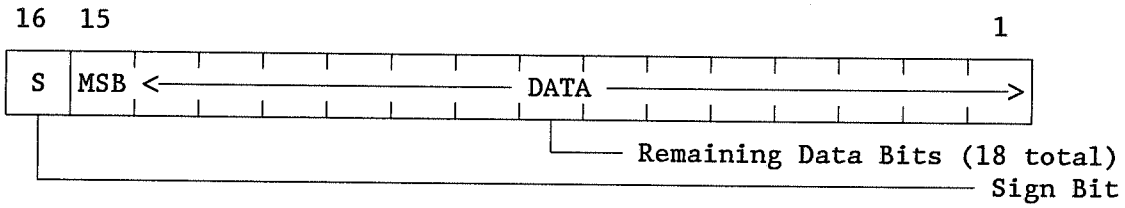


HS-3282 FORMATS

WORD 1



WORD 2



APPENDIX B

```
#define       VXIDOS 1
#include      "nivxi.h"

short  xmit[512];
short  point[256];

main ()
{
        short logical_id, status;
        unsigned short value;
        unsigned long offset;
        short   SMR, i, inc, x;


        system("RESMAN -o");

        /* Initialize VXI library */
        status = InitVXIlibrary ();
        if (status != 0 )
        {
                printf ("\nError Opening VXI Library!!\n");
                printf ("Error Status = %d", status);
                exit(-1);
        }

        /* Find the Logical Address of the V510 */
        status = FindDevLA ("", -1, 0x510, -1, -1, -1, -1, &logical_id);
        if ( status != 0 )
        {
                printf ("\nCould Not Find Instrument In Crate!!\n");
                printf ("Error Status = %d", status);
                exit(-1);
        }

        /* Read the Offset register from the instrument */
        status = VXIinReg ( logical_id, 6, &value );
        if ( status != 0 )
        {
                printf ("Error Reading Instrument Offset Register\n");
                exit (-1);
        }

        /* Shift the A24 Base offset over 8 */
        offset = (unsigned long)value << 8;
        printf ("\nChecking all of the registers....\n\n\n\n");
        printf ("\nTHIS EXAMPLE WILL RUN UNTIL A KEY IS HIT\n");

/*****************************************************************/
        SMR = 0;
        /*write to the diagnostic register*/
        status = VXIout(2, offset + 0, 2, 1);
        if ( status != 0 )
```

```
        {
                printf ("\nError writing the diagnotic register");
                exit(-1);
        }

/*      disable ARINC Activity   */
        status = VXIin( 2, offset + 0x62, 2, &value);
        if ( status != 0 )
        {
                printf ("\nError Disabling ARINC Activity");
                exit(-1);
        }

    do
    {

/*      Write LAM Request Register.     */
        status = VXIout( 2, offset + 0x42, 2, 4);
        if ( status != 0 )
        {
                printf ("Error Writing the LAM Request Register");
                exit(-1);
        }

/*      Write configuration word.       */
        status = VXIout( 2, offset + 0x2A, 2, 0x20);
        if ( status != 0 )
        {
                printf ("Error writing the configuration word");
                exit(-1);
        }

        inc = 0;
/*      Build loop for data here. .....         */
        for ( i = 0; i < 512; i+=2 )
        {
                xmit[i] = inc;
                xmit[i+1] = 255 - inc;
                inc++;
        }

/*      load ARIC data words in transmit fifo.  */
        for ( i = 0; i < 512; i++ )
        {
                status = VXIout( 2, offset + 0x22, 2, xmit[i]);
                if ( status != 0 )
                {
                        printf ("Error Writing transmit data ");
                        exit(-1);
                }
        }
```

```
/*    Set pointer memory address here.        */
      status = VXIout( 2, offset + 0x2E, 2, 0);
      if ( status != 0 )
      {
            printf ("Error resetting pointer memory address.");
            exit(-1);
      }

/*    setup array for pointer memory.         */
      for ( i = 0 ; i < 256 ; i++ )
            point[i] = i;

/*    set SMR bit in array.                   */
      point[SMR] = point[SMR] + 256;

/*    load pointer memory.                    */
      for ( i = 0; i < 256; i++ )
      {
            status = VXIout( 2, offset + 0x36, 2, point[i]);
            if ( status != 0 )
            {
                  printf ("Error writing pointer memory. ");
                  exit(-1);
            }
      }

/*    enable activity.                        */
      status = VXIin( 2, offset + 0x5E, 2, &value);
      if ( status != 0 )
      {
            printf ("Error enabling activity. ");
            exit(-1);
      }

/*    check for status before we continue.    */
      do
      {
            status = VXIin( 2, offset + 0x46, 2, &value);
            if ( status != 0 )
            {
                  printf ("Error checking for status.");
                  exit(-1);
            }

      } while ( value != 15 && !kbhit() );

/*    clear status word.                      */
      status = VXIout( 2, offset + 0x4E, 2, value);
      if ( status != 0 )
      {
            printf ("Error clearing status register.");
            exit(-1);
```

```
        }

/*      set address for memory readout.         */
        status = VXIout( 2, offset + 0x1A, 2, 0);
        if ( status != 0 )
        {
                printf ("Error writing rec. memory address register. ");
                exit(-1);
        }

/*      read block of data out of receiver memory. */
        for ( i = 0 ; i < 511 ; i++ )
        {
                status = VXIin( 2, offset + 0x12, 2, &value);
                if ( status != 0 )
                {
                printf ("Error reading receiver memory. ");
                exit(-1);
                }
        }

/*      disable ARINC activity.                 */
        status = VXIin( 2, offset + 0x62, 2, &value);
        if ( status != 0 )
        {
                printf ("Error disabling ARINC activity. ");
                exit(-1);
        }

        SMR++;
        if (SMR == 256 )
                SMR = 0;

    } while (!kbhit());

        printf ("\n\nAll register have either been written too or read from");
        printf ("\nThis module has passed!!");

        status = CloseVXIlibrary ();
        if ( status != 0 )
        {
                printf ("\nError Closing VXI Library!!\n");
                printf ("Error Status = %d", status);
                exit(-1);
        }
}
```