ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

**Artisan Technology Group**
(217) 352-9330 | sales@artisantg.com | artisantg.com

**Full-service, independent repair center**
with experienced engineers and technicians on staff.

**We buy your excess, underutilized, and idle equipment**
along with credit for buybacks and trade-ins.

**Custom engineering**
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

**Expert team | Trust guarantee | 100% satisfaction**

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *ICS Electronics ICS-330* at our website: ***Click HERE***

# ICS - 330

# OPERATING MANUAL

Interactive Circuits And Systems Ltd.
October 1997

The information in this manual has been carefully checked and is believed to be reliable; however, no responsibility is assumed for possible inaccuracies or omissions. Interactive Circuits and Systems Ltd. reserves the right to make changes to products herein described to improve reliability, function, or design. No patent rights are granted to any of the circuits described herein.

Extra copies of this manual are available from the factory.

ICS Ltd
5430 Canotek Road
Gloucester, Ontario
K1J 9G2 Canada

Tel: (613) 749-9241
USA: (800) 267-9794
Fax: (613) 749-9461

**DOC# E10375**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# TABLE OF CONTENTS


1 BOARD DESCRIPTION . . . . . 2
1.1 ARCHITECTURE . . . . . 2
1.2 SPECIFICATION . . . . . 4
1.3 MEMORY MAP . . . . . 4
1.3.1 Control Register . . . . . 4
1.3.2 Status Register . . . . . 4
1.3.3 Current Data Registers . . . . . 4
1.3.4 Latched Data Registers . . . . . 5
1.3.5 Test Data Registers . . . . . 5
1.3.7 High/Low Band Select Register . . . . . 5
. . . . . 6

2 INSTALLATION . . . . . 7
2.1 BOARD INSTALLATION . . . . . 7
2.1.1 Board Address Selection . . . . . 7
2.1.2 Interrupt Vector . . . . . 9
2.1.3 Interrupt Level Selection . . . . . 9
2.1.4 Address Modifier Code Response . . . . . 9
2.1.5 TCU/ACS Configuration . . . . . 10
2.2 CONNECTOR INTERFACE DEFINITION . . . . . 10
2.3 DRIVER INSTALLATION . . . . . 11

3 SUNOS DRIVER . . . . . 15
3.1 STARTING OUT AND FINISHING UP . . . . . 16
3.2 THE ioctl FUNCTION . . . . . 17
3.2.1 RESET . . . . . 18
3.2.2 RUN . . . . . 18
3.2.3 WORK . . . . . 18
3.2.4 TEST . . . . . 18
3.2.5 INT_ENABLE . . . . . 18
3.2.6 READ_STATUS . . . . . 19
3.2.7 READ_ED_REG . . . . . 19
3.2.8 WRITE_ED_REG . . . . . 19
3.2.9 READ_BAND_SEL . . . . . 19
3.2.10 WRITE_BAND_SEL . . . . . 19
3.2.11 WRITE_TEST . . . . . 19
3.2.12 READ_LATCHED . . . . . 19
3.2.13 READ_CURRENT . . . . . 20
3.2.14 FIND_CHANGE . . . . . 20
3.3 High Level Routines . . . . . 20
3.3.1 dio_start_pmfl_sampling () . . . . . 20
3.3.2 dio_wait_for_pmfl_change () . . . . . 21
3.3.3 dio_stop_pmfl_sampling () . . . . . 21
3.3.4 dio_perform_test () . . . . . 21

**APPENDIX 1 CIRCUIT DESCRIPTION** . . . . . 23
A1.1 DIGITAL CONTROL SIGNAL GENERATION . . . . . 23

APPENDIX 2 TROUBLESHOOTING . . . . . 26
A2.1 TROUBLESHOOTING PROCEDURES . . . . . 26
A2.2 TEST EQUIPMENT . . . . . 29




Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## DOCUMENT OVERVIEW

This manual provides complete instructions for installing and operating the ICS-330 Digital I/O card in a SUNOS environment. As appendices, it also contains a circuit description for the card, as well as troubleshooting information. The manual consists of the following sections:

1. Board Description

   - this section contains a description of the card's operating characteristics, specifications, and memory map.

2. Installation Instructions

   - this section contains installation information for both the ICS-330 card and its associated SUNOS driver. It also contains necessary I/O interface information.

3. Driver Description

   - this section discusses the use of the SUNOS driver's low and high level routines.

A1. Circuit Description

   - this section provides a more detailed technical description of the design.

A2. Troubleshooting

   - information needed for basic troubleshooting can be found in this section.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# 1 BOARD DESCRIPTION

## 1.1 ARCHITECTURE

The ICS-330 is an RS-422 level-compatible digital I/O board which provides 41 differential inputs and 24 differential control outputs. Performance Monitoring/Fault Location (PMFL) bits account for 40 of the differential input bits. The final input bit can be configured for one of two control inputs: the Accoustic Communications (ACS) Request To Transmit signal, or the Time Control Unit (TCU) Transmit Start Pulse. Each input is sampled and compared with the corresponding previously sampled states. Any change in state causes an on-board interrupt. Both present and previous states of each input can be read from VMEbus.

The control outputs include 10 Enable/Disable bits, 10 Hi/Lo Band Select bits, and 3 Low-Frequency Transmit Subsystem (LTS) control bits. All control outputs are differential. Fig. 1.1 shows a simplified block diagram of the ICS-330.

The differential inputs can be tri-stated, allowing a set of memory-mapped test registers to feed the comparator circuit. Also, the differential output bits are buffered back to VMEbus for diagnostic purposes.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

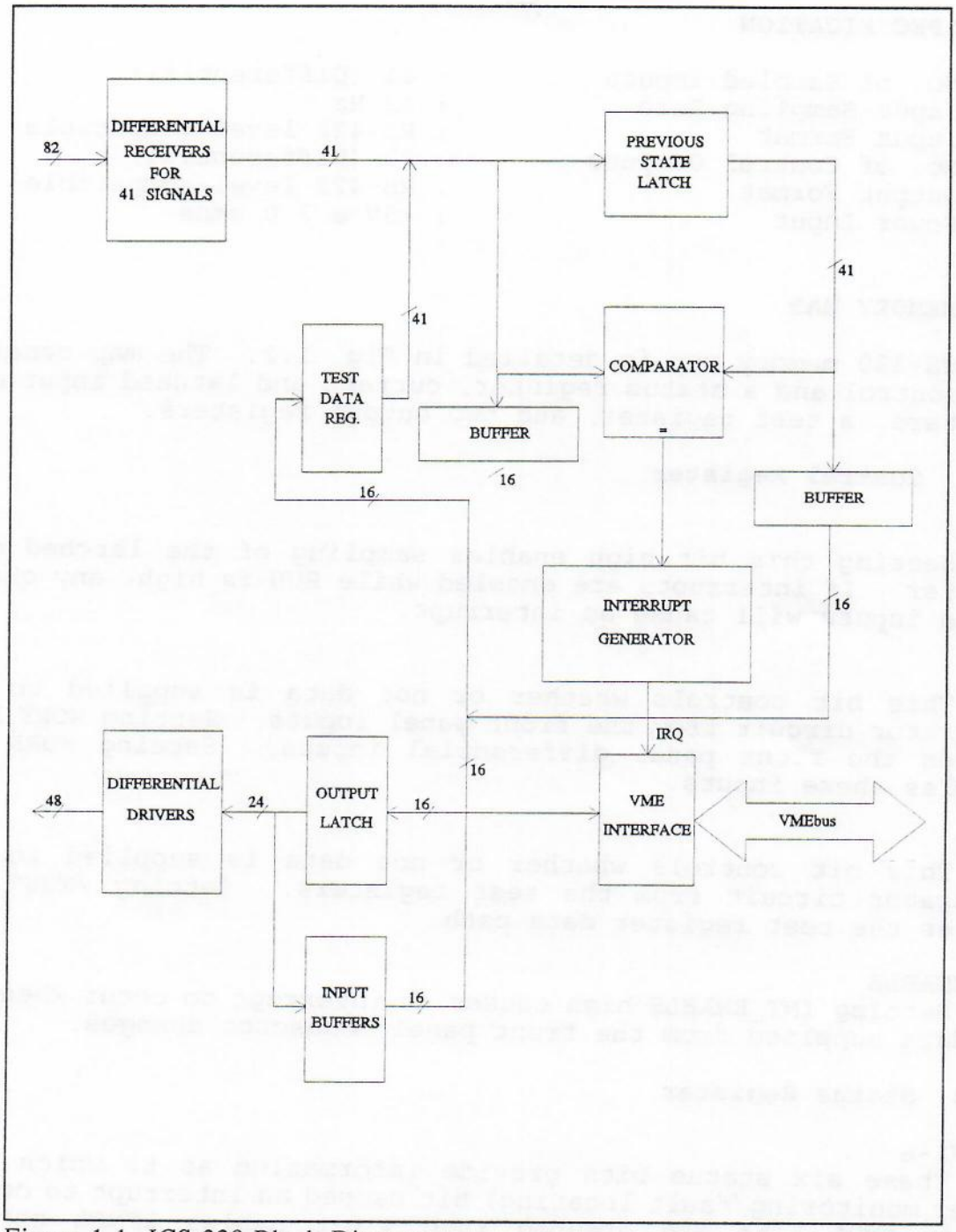



Figure 1.1 ICS-330 Block Diagram



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## 1.2 SPECIFICATION

| | |
|---|---|
| No. of Sampled Inputs | : 41 (Differential) |
| Input Sampling Rate | : 12 Hz |
| Input Format | : RS-422 level-compatible |
| No. of Control Outputs | : 23 (Differential) |
| Output Format | : RS-422 level-compatible |
| Power Input | : +5V @ 2.0 amps |

## 1.3 MEMORY MAP

The ICS-330 memory map is detailed in Fig. 1.2. The map consists of a control and a status register, current and latched input data registers, a test register, and two output registers.

### 1.3.1 Control Register

RUN

Setting this bit high enables sampling of the latched data register. If interrupts are enabled while RUN is high, any change in the inputs will cause an interrupt.

WORK

This bit controls whether or not data is supplied to the comparator circuit from the front panel inputs. Setting WORK high enables the front panel differential inputs. Setting WORK low disables these inputs.

/TEST

This bit controls whether or not data is supplied to the comparator circuit from the test registers. Setting /TEST low enables the test register data path.

INT_ENABLE

Setting INT_ENABLE high causes an interrupt to occur whenever the data supplied from the front panel connector changes.

### 1.3.2 Status Register

MATCH1-6

These six status bits provide information as to which PMFL (power monitoring/fault locating) bit caused an interrupt to occur. The 40 PMFL bits are grouped into 4-bit nybles 1PMFL through 10PMFL. MATCH1 set high indicates that a bit in either 1PMFL or 2PMFL changed state. Similarly, MATCH2 set high indicates that a bit in either 3PMFL or 4PMFL changed state. MATCH6 indicates that the ACS_RQ_TO_XMT/TCU_XMT_START bit changed state.

### 1.3.3 Current Data Registers



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The current data registers map the input data at the input connectors of the ICS-330 to VMEbus. These three registers contain the states of all 40 PMFL bits, as well as the ACS/TCU bit.

### 1.3.4 Latched Data Registers

The latched data registers map the sampled input data to VMEbus. These three registers contain the same format as the current data registers.

### 1.3.5 Test Data Registers

The test data registers feed data from VMEbus to the current and latched data registers, allowing the user to effect bit changes by overriding the front panel differential inputs. Again, these three registers contain the same format as the current data registers.

### 1.3.6 Enable/Disable Register

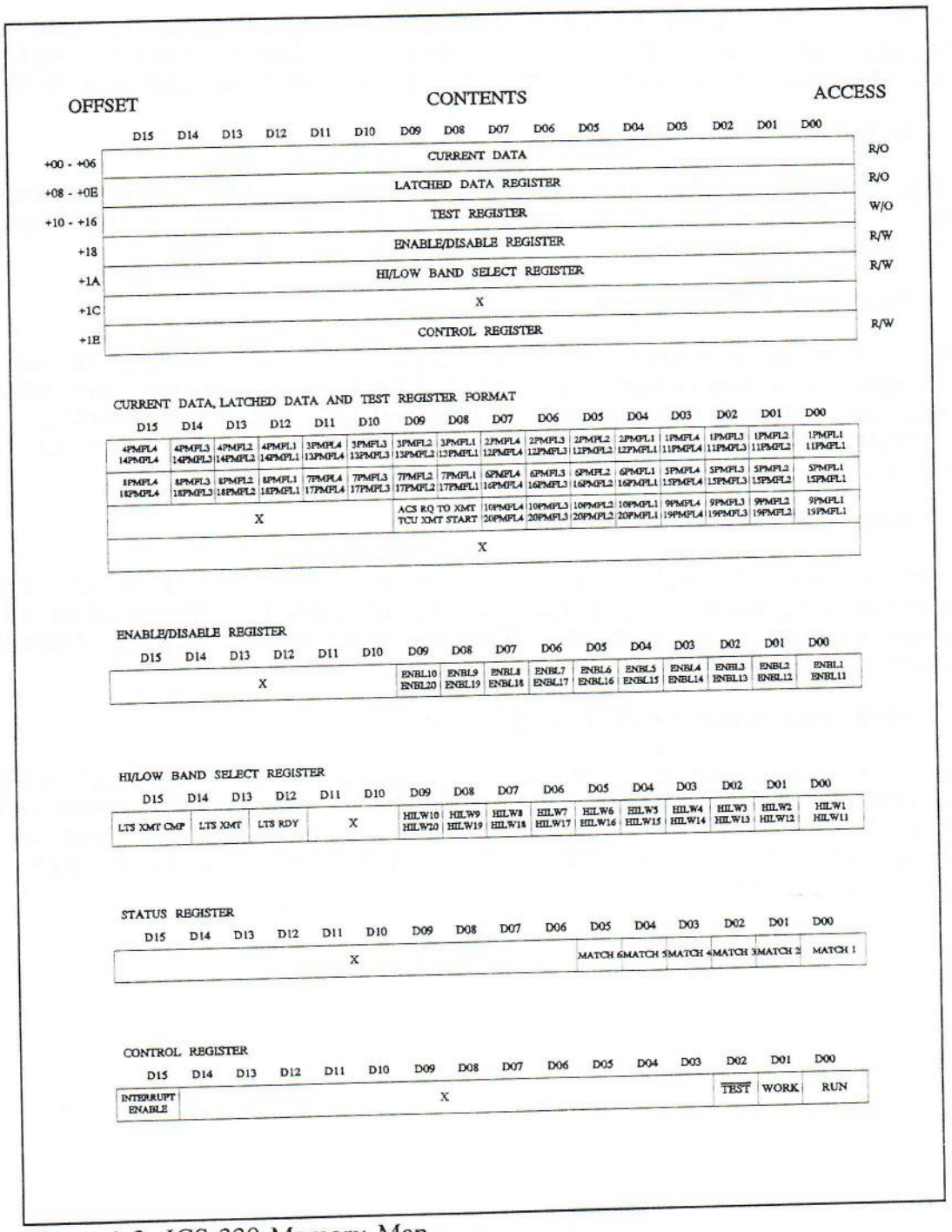
The enable/disable register maps 10 control bits to 10 differential outputs on the ICS-330 front panel. These bits can be addressed at any time and can also be read back, via differential receivers, for diagnostic purposes.

### 1.3.7 High/Low Band Select Register

This register maps 13 control bits to 13 differential outputs on the ICS-330 front panel. These include 10 high/low band select bits and 3 LTS control bits. These bits can be addresed at any time and can also be read back, via differential receivers, for diagnostic purposes.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



| OFFSET | CONTENTS | ACCESS |
|---|---|---|
| +00 - +06 | CURRENT DATA | R/O |
| +08 - +0E | LATCHED DATA REGISTER | R/O |
| +10 - +16 | TEST REGISTER | W/O |
| +18 | ENABLE/DISABLE REGISTER | R/W |
| +1A | HI/LOW BAND SELECT REGISTER | R/W |
| +1C | X | |
| +1E | CONTROL REGISTER | R/W |

CURRENT DATA, LATCHED DATA AND TEST REGISTER FORMAT

| D15 | D14 | D13 | D12 | D11 | D10 | D09 | D08 | D07 | D06 | D05 | D04 | D03 | D02 | D01 | D00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4PMFL4<br>14PMFL4 | 4PMFL3<br>14PMFL3 | 4PMFL2<br>14PMFL2 | 4PMFL1<br>14PMFL1 | 3PMFL4<br>13PMFL4 | 3PMFL3<br>13PMFL3 | 3PMFL2<br>13PMFL2 | 3PMFL1<br>13PMFL1 | 2PMFL4<br>12PMFL4 | 2PMFL3<br>12PMFL3 | 2PMFL2<br>12PMFL2 | 2PMFL1<br>12PMFL1 | 1PMFL4<br>11PMFL4 | 1PMFL3<br>11PMFL3 | 1PMFL2<br>11PMFL2 | 1PMFL1<br>11PMFL1 |
| 8PMFL4<br>18PMFL4 | 8PMFL3<br>18PMFL3 | 8PMFL2<br>18PMFL2 | 8PMFL1<br>18PMFL1 | 7PMFL4<br>17PMFL4 | 7PMFL3<br>17PMFL3 | 7PMFL2<br>17PMFL2 | 7PMFL1<br>17PMFL1 | 6PMFL4<br>16PMFL4 | 6PMFL3<br>16PMFL3 | 6PMFL2<br>16PMFL2 | 6PMFL1<br>16PMFL1 | 5PMFL4<br>15PMFL4 | 5PMFL3<br>15PMFL3 | 5PMFL2<br>15PMFL2 | 5PMFL1<br>15PMFL1 |
| X | | | | | | ACS RQ TO XMT<br>TCU XMT START | | 10PMFL4<br>20PMFL4 | 10PMFL3<br>20PMFL3 | 10PMFL2<br>20PMFL2 | 10PMFL1<br>20PMFL1 | 9PMFL4<br>19PMFL4 | 9PMFL3<br>19PMFL3 | 9PMFL2<br>19PMFL2 | 9PMFL1<br>19PMFL1 |
| X | | | | | | | | | | | | | | | |

ENABLE/DISABLE REGISTER

| D15 | D14 | D13 | D12 | D11 | D10 | D09 | D08 | D07 | D06 | D05 | D04 | D03 | D02 | D01 | D00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | | | | | | ENBL10<br>ENBL20 | ENBL9<br>ENBL19 | ENBL8<br>ENBL18 | ENBL7<br>ENBL17 | ENBL6<br>ENBL16 | ENBL5<br>ENBL15 | ENBL4<br>ENBL14 | ENBL3<br>ENBL13 | ENBL2<br>ENBL12 | ENBL1<br>ENBL11 |

HI/LOW BAND SELECT REGISTER

| D15 | D14 | D13 | D12 | D11 | D10 | D09 | D08 | D07 | D06 | D05 | D04 | D03 | D02 | D01 | D00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LTS XMT CMP | LTS XMT | LTS RDY | X | | | HILW10<br>HILW20 | HILW9<br>HILW19 | HILW8<br>HILW18 | HILW7<br>HILW17 | HILW6<br>HILW16 | HILW5<br>HILW15 | HILW4<br>HILW14 | HILW3<br>HILW13 | HILW2<br>HILW12 | HILW1<br>HILW11 |

STATUS REGISTER

| D15 | D14 | D13 | D12 | D11 | D10 | D09 | D08 | D07 | D06 | D05 | D04 | D03 | D02 | D01 | D00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | | | | | | | | | | MATCH 6 | MATCH 5 | MATCH 4 | MATCH 3 | MATCH 2 | MATCH 1 |

CONTROL REGISTER

| D15 | D14 | D13 | D12 | D11 | D10 | D09 | D08 | D07 | D06 | D05 | D04 | D03 | D02 | D01 | D00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INTERRUPT ENABLE | X | | | | | | | | | | | | $\overline{\text{TEST}}$ | WORK | RUN |

Figure 1.2 ICS-330 Memory Map



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# 2 INSTALLATION

## 2.1 BOARD INSTALLATION

Prior to installation, it is necessary to make two decisions regarding the configuration of the ICS-330 board:

1. Physical Address: the address can be chosen from the standard 24-bit VMEbus address space. "Writing Device Drivers" of the SUN documentation claims 24-bit VMEbus address space 0x800000 to 0xdfffff is available for use. To avoid choosing a physical address which conflicts with memory space already in use, it is recommended that the addresses of any boards already installed in the SUN computer be identified. During re-boots, diagnostics are maintained in the file "/usr/adm/messages" which, among other things, identifies the devices recognized by the System. Reviewing this file will assist in identifying memory addresses already in use.

2. Interrupt Vector: "Writing Device Drivers" of the SUN documentation claims that vectors 0xc8 through 0xff are available for use. Again, it is strongly suggested that the interrupt vector assignments of other boards installed in the SUN computer be identified. The "/usr/adm/messages" file indicates the interrupt vectors already in use.

Once these two decisions have been made, the installation procedure can commence. Fig. 2.1 shows the locations of various DIP switches.

### 2.1.1 Board Address Selection

The base address of the ICS-330 can be set to any 512 byte block boundary in the 16Mbyte VME address space. The DIP switch set corresponds to VMEbus address lines as follows:

| VME address | DIP sw | VME address | DIP sw |
|---|---|---|---|
| A09 | H11 | A16 | H3 |
| A10 | H12 | A17 | H4 |
| A11 | H13 | A18 | H5 |
| A12 | H14 | A19 | H6 |
| A13 | H15 | A20 | H7 |
| A14 | H16 | A21 | H8 |
| A15 | H17 | A22 | H9 |
| /LWD | H18 (open) | A23 | H10 |

An open DIP switch sets the corresponding address bit HIGH. /LWD must always remain open.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

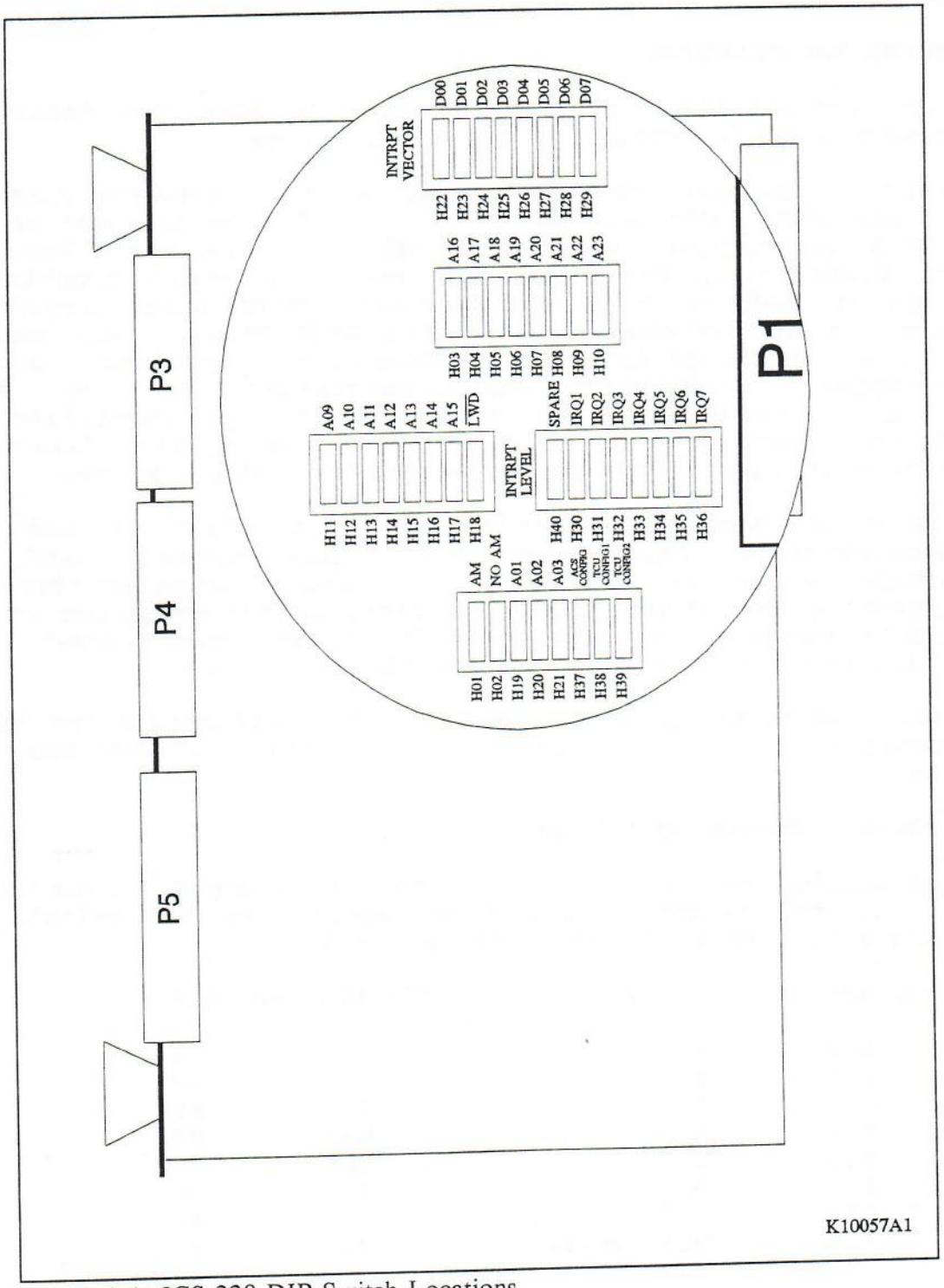



Figure 2.1 ICS-330 DIP Switch Locations



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### 2.1.2 Interrupt Vector

When the ICS-330 is enabled for interrupts, an 8-bit interrupt vector code is sent to the processor indicating what device has generated the interrupt. This code is selectable using DIP switches H22 through H29 as follows:

| Vector Address | DIP sw |
|---|---|
| D00 | H22 |
| D01 | H23 |
| D02 | H24 |
| D03 | H25 |
| D04 | H26 |
| D05 | H27 |
| D06 | H28 |
| D07 | H29 |

An open DIP switch sets the corresponding vector address bit HIGH.

### 2.1.3 Interrupt Level Selection

Any one of seven interrupt levels can be selected by connecting one of the seven request lines IRQ1-IRQ7. Corresponding to a selected interrupt level, a 3-bit interrupt level match must also be set. The possible choices are listed below:

| Interrupt Level | IRQ DIP | Level DIP's | | |
|---|---|---|---|---|
| 1 | H30 | **H19** | H20 | H21 |
| 2 | H31 | H19 | **H20** | H21 |
| 3 | H32 | **H19** | **H20** | H21 |
| 4 | H33 | H19 | H20 | **H21** |
| 5 | H34 | **H19** | H20 | **H21** |
| 6 | H35 | H19 | **H20** | **H21** |
| 7 | H36 | **H19** | **H20** | **H21** |

The IRQ DIP column indicates which of DIP switches H30-H36 should be closed. The Level DIP column indicates which of DIP switches H19-H21 should be open (BOLD = open).

### 2.1.4 Address Modifier Code Response

The ICS-330 can be configured to respond to one of two address modifier conditions using DIP switches H1-H2. DIP switch H1 closed selects standard addressing (A24) with address modifiers $39, $3D, and $3F. These refer to non-priviledged, supervisory, and block I/O access, respectively. DIP switch H2 closed selects addressing without AM code decoding. DIP switches H1 and H2 should never both be closed when the card is powered up.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### 2.1.5 TCU/ACS Configuration

One input on the ICS-330 (P5-49&50) can be configured as a differential input (ACS REQUEST TO TRANSMIT), or a 50 ohm terminated single-ended TTL input (TCU TRANSMIT START). DIP switches H37 through H39 select one of the two above configurations:

ACS REQUEST TO TRANSMIT - H37 closed
H38 open
H39 open

TCU TRANSMIT START - H37 open
H38 closed
H39 closed

## 2.2 CONNECTOR INTERFACE DEFINITION

**P3 PMFL Input Data**

| | |
|---|---|
| 1,2 | PMFL 1.0 |
| 3,4 | PMFL 1.1 |
| 5,6 | PMFL 1.2 |
| 7,8 | PMFL 1.3 |
| 9,10 | PMFL 2.0 |
| 11,12 | PMFL 2.1 |
| 13,14 | PMFL 2.2 |
| 15,16 | PMFL 2.3 |
| 17,18 | PMFL 3.0 |
| 19,20 | PMFL 3.1 |
| 21,22 | PMFL 3.2 |
| 23,24 | PMFL 3.3 |
| 25,26 | PMFL 4.0 |
| 27,28 | PMFL 4.1 |
| 29,30 | PMFL 4.2 |
| 31,32 | PMFL 4.3 |
| 33,34 | PMFL 5.0 |
| 35,36 | PMFL 5.1 |
| 37,38 | PMFL 5.2 |
| 39,40 | PMFL 5.3 |

-paired pins are +,-

**P4 PMFL Input Data**

| | |
|---|---|
| 1,2 | PMFL 6.0 |
| 3,4 | PMFL 6.1 |
| 5,6 | PMFL 6.2 |
| 7,8 | PMFL 6.3 |
| 9,10 | PMFL 7.0 |
| 11,12 | PMFL 7.1 |
| 13,14 | PMFL 7.2 |
| 15,16 | PMFL 7.3 |
| 17,18 | PMFL 8.0 |
| 19,20 | PMFL 8.1 |
| 21,22 | PMFL 8.2 |
| 23,24 | PMFL 8.3 |
| 25,26 | PMFL 9.0 |
| 27,28 | PMFL 9.1 |
| 29,30 | PMFL 9.2 |
| 31,32 | PMFL 9.3 |
| 33,34 | PMFL 10.0 |
| 35,36 | PMFL 10.1 |
| 37,38 | PMFL 10.2 |
| 39,40 | PMFL 10.3 |



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**P5 Control Outputs & ACS/TCU Input**

| | | | |
|---|---|---|---|
| 1,2 | Enable/Disable 1 | 27,28 | Hi/Low Select 4 |
| 3,4 | Enable/Disable 2 | 29,30 | Hi/Low Select 5 |
| 5,6 | Enable/Disable 3 | 31,32 | Hi/Low Select 6 |
| 7,8 | Enable/Disable 4 | 33,34 | Hi/Low Select 7 |
| 9,10 | Enable/Disable 5 | 35,36 | Hi/Low Select 8 |
| 11,12 | Enable/Disable 6 | 37,38 | Hi/Low Select 9 |
| 13,14 | Enable/Disable 7 | 39,40 | Hi/Low Select 10 |
| 15,16 | Enable/Disable 8 | 41,42 | LTS READY |
| 17,18 | Enable/Disable 9 | 43,44 | LTS XMITTING |
| 19,20 | Enable/Disable 10 | 45,46 | LTS XMIT COMPLETE |
| 21,22 | Hi/Low Select 1 | 47,48 | Spare |
| 23,24 | Hi/Low Select 2 | 49,50 | ACS RQ TO XMIT/ |
| 25,26 | Hi/Low Select 3 | | TCU XMIT START |

-paired pins are +,-

## 2.3 DRIVER INSTALLATION

Once the board is successfully installed and the system rebooted, the device driver may be loaded into the SUN computer's kernel. The following instructions are applicable to SUNOS Version 4.1.

To install the device driver, log into the system as root, and follow the procedure outlined below.

***Step 1*** Make a new directory for the driver and related files. As an example, the directory "/sys/dio" might be used. To create such a directory, enter the following command:

```
mkdir /sys/dio
```

***Step 2*** Change to the new directory by using the 'cd' command. To do so, enter the following command:

```
cd /sys/dio
```

***Step 3*** Put the ICS-330 Driver Distribution tape into the tape drive and use the 'tar' command to load the contents of the tape into the directory created in Step 1. If the tape drive is designated as /dev/rst0, the command to be given to extract the driver from the tape would be as follows:

```
tar xvf /dev/rst0
```

***Step 4*** Put the driver source and include files into appropriate directories:

```
cp dio.c /sys/sundev
cp dioregs.h /usr/include/sys
cp diocmds.h /usr/include/sys
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

***Step 5*** The Kernel configuration file must now be modified to reflect the addition of the ICS-330 driver. The configuration file is located in "/sys/sun". Move to this directory by entering the following command:

```
cd /sys/sun
```

***Step 6*** Edit the file "conf.c" file and make the changes indicated below. NOTE: It may be necessary to change the file protection of "conf.c". If so, this can be done by issuing the command:

```
chmod u+w conf.c
```

prior to editing.

1. Immediately preceding the lines defining the character device switch (i.e. "struct cdevsw cdevsw[]=") and/or the line "extern int seltrue();", add the following text:

```
#include "dio.h"
#if NDIO > 0
extern  int  dioopen(),  dioclose(),  dioread(),  diowrite(),
          dioioctl();
#else
#define dioopen     nodev
#define dioclose    nodev
#define dioioctl    nodev
#endif
```

2. Find the end of the current structure definition, which should resemble the following:

```
    {
        taacopen, nulldev, nodev, nodev,   /*62*/
        nodev, nodev, seltrue, taacmmap
        0,
    },
};
```

Insert the following lines before the closing brace:

```
    {
        dioopen, dioclose, dioread, diowrite  /*63*/
        dioioctl, nodev, seltrue, 0,
        0,
    },
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The result of the modification should resemble the following:

```
        {
                taacopen, nulldev, nodev, nodev,   /*62*/
                nodev, nodev, seltrue, taacmmap,
                0,
        },
        {
                dioopen, dioclose, dioread, diowrite,   /*63*/
                dioioctl, nodev, seltrue, 0,
                0,
        },
};
```

**NOTE**: Each set of lines preceding the one added ends with a numbered comment corresponding to the major device number. Add a comment with the next numeric value to the end of newly added line and make a note of it (in this example the major device number is /*63*/). The board's major device number will be required later on in the installation process.

***Step 7*** The device must now be made evident to the System. To do so, begin by changing directories to "/sys/sun4/conf" as follows:

```
cd /sys/sun4/conf
```

***Step 8*** Edit the current config file as follows:

```
vi GENERIC
```

where "GENERIC" is the current config file. Again, prior to editing, it may be necessary to change the file protection of the new file.

***Step 9*** Append the following lines to the end of the file:

```
#
# Support for the ICS-330 DIO board
#
device dio0 at vme24d16 ? csr 0xd00000 priority 3 vector diointr
0xf0
```

**NOTE**: This example is for a 24-bit VME bus board base address of 0xd00000 and an interrupt vector of 0xf0. The address and interrupt vector chosen at Step 3 of the ICS-330 Board Installation procedure would be so indicated in the above line. For a 16-bit VMEbus base address of 0x3300, the appended line would be:

```
device dio0 at vme16d16 ? csr 0x3300 priority 3 vector diointr 0xf0
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

***Step 10*** Add the following line to the end of the file "files" in the current directory:

```
sundev/dio.c    optional dio device-driver
```

**NOTE**: Don't forget to add the "optional dio device-driver" after sundev/dio.c

***Step 11*** Give the following command, using the name given to the config file:

```
config GENERIC
```

***Step 12*** A directory will be created containing all the necessary object modules to build a new kernel. Move to this directory by giving the following command, using the name given to the config file:

```
cd ../GENERIC
```

***Step 13*** Generate a new kernel by typing:

```
make
```

A successful generation will be concluded with text resembling the following being issued on the display:

```
loading vmunix
rearranging symbols
text    date    bss     dec     hex
63596   115496  160912 912344 debd8
```

***Step 14*** Save the original "/vmunix" file and move the new kernel to the root directory with the following set of commands:

```
mv /vmunix /oldvmunix
mv vmunix /vmunix
```

***Step 15*** Re-boot the system by using the command:

```
sync; sync; reboot
```

If everything has been done correctly, the following line will be evident on the display as part of the boot-up's report:

```
dio0 installed
```

If this line cannot be found in the report, or the line:

```
dio0 missing
```

is present, review the preceding steps and verify that the board



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

has been correctly installed and that the jumpers match the desired choices.

***Step 16*** Change to the "/dev" directory as follows:

```
cd /dev
```

***Step 17*** Issue the following commands, using the major device number (63) in the comment from Step 6 above:

```
/etc/mknod /dev/dio0 c 63 0
chmod ugo+rw dio0
```

The ICS-330 driver is now installed and ready for use. Note that if another board is to be added, it would be named /dev/dio1 and the steps from 9 to 17 would be repeated for the new board.

## 3 SUNOS DRIVER

The ICS-330 device driver has been implemented to allow the ICS-330 board to function as a file from the viewpoint of User processes. User programs make accesses to the board in the same manner they would access any other file (or device) in a UNIX environment.

Three driver functions are available:

open () - open the device for exclusive access;

close () - free up the device to other processes;

ioctl () - place the device in various modes of operation.

Each of these functions will be detailed further in a succeeding section.

In addition to the above standard UNIX functions, there are also four specialized high level routines:

dio_start_pmfl_sampling () - start pmfl sampling;

dio_wait_for_pmfl_change () - read the pmfl which changed;

dio_stop_pmfl_sampling () - stop pmfl sampling;

dio_perform_test () - perform ICS-330 diagnostics;

These functions should be used in preference to the ioctl call since these functions were specially tailored for the ICS-330 and are easier to use. Each of these functions will be detailed further in a succeeding section.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## 3.1 STARTING OUT AND FINISHING UP

Before a program can begin to control an installed ICS-330 board, there are a few things it needs to do. First, it must know about the constants needed to communicate with the ICS-330 driver. These can be found in three header files:

| | |
|---|---|
| fcntl.h | which contains some standard UNIX constants; |
| dioreg.h | which contains constants for the dio driver; and, |
| diocmds.h | which contains ioctl requests for the dio driver. |

Since these header files all live in /usr/include/sys, they can be included using angle brackets, just as stdio.h is included (see below for examples). If any one of the files is not included, the C compiler will fail to compile your program and will display error messages saying that some variables (the constants defined in these header files) have not been defined.

Next, the file that is attached to the ICS-330 must be opened. This is done using the UNIX open function, which takes two parameters: the name of the file to be opened, and the access mode. The name of the file in most cases will be /dev/dio0; if more than one ICS-330 is installed on the system, then the file name will be /dev/dion, where n ranges from 0 to the number of boards installed minus one. The access mode can be one of:

| | |
|---|---|
| O_RDONLY | read only; |
| O_WRONLY | write only; or, |
| O_RDWR | read and write. |

Regardless of what the access mode is, only one process is allowed access to the ICS-330 board at a time; that is, any attempt to open an already open ICS-330 file will generate an error.

When an ICS-330 is successfully opened, the open routine will return an integer (the file descriptor) that is used to refer to the active ICS-330 in all subsequent actions. If the file descriptor that was returned is -1, then an error has occurred during the open; if this is the case, then the global variable errno will be set to error number of the error that occurred. Thus, a typical application program would use the following code to open the ICS-330 attached to the file /dev/dio0:

```
int dio_handle;
dio_handle = open("/dev/dio0", O_RDWR);
if (dio_handle == -1)
```



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
{
perror("Error opening /dev/dio0 ");
exit(errno)
}
```

That is, if the open was unsuccessful, perror (see the manual entry for a description) uses errno to print out an appropriate error message, then the program exists with a status indicating that an error has occurred. Possible errors include the device driver being not installed, the ICS-330 being not present, and the ICS-330 being used by another user or process.

Now that the ICS-330 is open, you can use the ioctl function to control the board. When you are done with the ICS-330, you should use the close function to free it up for use by other users or processes by using the following call:

```
close(dio_handle);
```

This call should not generate any errors.

### 3.2 THE ioctl FUNCTION

The ioctl (i/o control) function is the UNIX function that is used to set up i/o devices. When using the ICS-330, the user may have to call ioctl several times, with different requests, to correctly setup the ICS-330. The format for an ioctl (which cannot be called until after the ICS-330 has been opened) is:

```
status = ioctl (dio_handle, request, arg);
```

where the parameters are:

```
int dio_handle;     /* file descriptor of the open ICS-330 */
int request;        /* one of the requests in diocmds.h */
short *arg;         /* pointer to an argument.  The type of
                       the argument depends on the request */
```

The ioctl command returns 0 if the call was successful, and -1 otherwise. If -1 was returned, then the externally defined variable "errno" will be set to one of:

| | |
|---|---|
| ENOTTY | parameter not defined for the device; and |
| EINVAL | option is out of range. |

These error return values should help to determine what caused the error.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

An example of an ioctl() call is as follows:

```
short arg = 1;
if (ioctl (dio_handle, RUN, &arg) == -1)
     {
     perror("Error setting RUN bit ");
     exit (errno);
     }
```

Note that the following call is NOT correct and can cause your program to bomb:

```
if (ioctl (dio_handle, RUN, 1) == -1)
     {
     perror("Error setting RUN bit ");
     exit (errno);
     }
```

and is a frequent source of problems.

The following sub-sections will discuss each request and the range of its arguments.

### 3.2.1 RESET

The RESET request disables the ability for the ICS-330 to generate an interrupt and resets the ICS-330 registers. It does not take an argument.

### 3.2.2 RUN

The RUN request sets the RUN bit on the ICS-330 to the value held in the short argument. The argument can have only two possible values: 0 and 1.

### 3.2.3 WORK

The WORK request sets the WORK bit on the ICS-330 to the value held in the short argument. If the WORK bit is set, the data from the receivers can be examined; otherwise they cannot be examined. The argument can only have two possible values: 0 and 1.

### 3.2.4 TEST

The TEST request sets the TEST bit on the ICS-330 to the value held in the short argument. The argument can have only two possible values: 0 and 1.

### 3.2.5 INT_ENABLE

The INT_ENABLE request sets the INTERRUPT ENABLE bit on the ICS-330



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

to the value held in the short argument. If the bit is cleared, interrupts are disabled; otherwise, interrupts are enabled. The argument can have only two possible values: 0 and 1.

### 3.2.6 READ_STATUS

The READ_STATUS request returns the value in the ICS-330 status register. The argument must be a pointer to a short. The value of the register will be returned in the argument.

### 3.2.7 READ_ED_REG

The READ_ED_REG request returns the value in the ICS-330 ENABLE/DISABLE register. The argument must be a pointer to a short. The value of the register will be returned in the argument.

### 3.2.8 WRITE_ED_REG

The WRITE_ED_REG request writes the value pointed to by the argument into the ICS-330 ENABLE/DISABLE register. The argument must be a pointer to a short.

### 3.2.9 READ_BAND_SEL

The READ_BAND_SEL request returns the value in the ICS-330 BAND SELECT register. The argument must be a pointer to a short. The value of the register will be returned in the argument.

### 3.2.10 WRITE_BAND_SEL

The WRITE_BAND_SEL request writes the value pointed to by the argument into the ICS-330 BAND SELECT register. The argument must be a pointer to a short.

### 3.2.11 WRITE_TEST

The WRITE_TEST request writes the values pointed to by the argument into the ICS-330 TEST registers. The argument must be a pointer to an array of three shorts. The first short will be written into the low test register, the second into the mid test register, and the third into the high test register.

### 3.2.12 READ_LATCHED

The READ_LATCHED request returns the values in the ICS-330 LATCHED register. The argument must be a pointer to an array of three shorts. The value in the low latched register will be returned in the first short, the value of the mid latched register in the second, and the value of the high latched register in the third.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### 3.2.13 READ_CURRENT

The READ_CURRENT request returns the values in the ICS-330 CURRENT register. The argument must be a pointer to an array of three shorts. The value in the low current register will be returned in the first short, the value of the mid current register in the second, and the value of the high current register in the third.

### 3.2.14 FIND_CHANGE

The FIND_CHANGE request determines if any of the PMFLs on either of the boards have changed. It returns the current value of the PMFL which has changed, and the number of the PMFL which has changed.

The pmfl number will have the following possible values: 1-20, 21, and 22. The values 1-20 correspond to the 20 pmfls. The value 21 corresponds to the ACS RQ TO XMT bit on the first ICS-330 and the value 22 corresponds to the TCU XMT START bit on the second ICS-330.

This call will block until a pmfl has changed on either of the cards.

## 3.3 High Level Routines

There are four specialized high level routines for the ICS-330. To be able to use them, the following conditions must be met:

1) two ICS-330s must already be opened;

2) the global variables "dio_handle0" and "dio_handle1" must be declared in the main user program;

3) the global variables "dio_handle0" and "dio_handle1" must be initialized to the file descriptors for the two ICS-330s;

4) the file "diohigh.o" must be linked in with your program.

The following sub-sections will discuss each of the high level routines.

### 3.3.1 dio_start_pmfl_sampling ()

This routine starts pmfl sampling on both cards by de-activating the TEST bit, and activating the RUN, WORK and interrupt enable bits. On success, this routine will return a zero; otherwise an integer error code is returned.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

### 3.3.2 dio_wait_for_pmfl_change ()

```
dio_wait_for_pmfl_change (buffer, amp)
short *buffer, *amp
```

This routine blocks the caller until an interrupt is received on either of the ICS-330s. The pointer "buffer" will point to a 16-bit word where bits 0 to 3 will be set to the value of the pmfl bits in the current register which provoked the interrupt. The pointer "amp" will point to the number of the corresponding pmfl. The number will take on a value between 1 and 20, or one of two special values, 21 and 22. The numbers between 1 and 20 correspond to the 20 pmfls. The number 21 corresponds to the ACS RQ TO XMT bit on the first card, and the number 22 corresponds to the TCU XMT START bit on the second card.

On success, this routine will return a zero; otherwise an integer error code is returned.

### 3.3.3 dio_stop_pmfl_sampling ()

This routine stops pmfl sampling on both cards by disabling the interrupt and RUN bits. On success, this routine will return a zero; otherwise an integer error code is returned.

### 3.3.4 dio_perform_test ()

This routine will initiate and perform testing on both ICS-330s and return the status of the cards' operational capability. A returned value of zero indicates that all tests passed successfully. Any other return value indicates that an error was detected.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

# APPENDICES

## APPENDIX 1

CIRCUIT DESCRIPTION

## APPENDIX 2

TROUBLESHOOTING



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## APPENDIX 1 CIRCUIT DESCRIPTION

### A1.1 DIGITAL CONTROL SIGNAL GENERATION

The following discussion assumes that a cicuit schematic of the ICS-330 is available.

The board timing is derived from a 2.4576 MHz crystal oscillator which is divided down to 12 Hz (CLKIN) using a divider circuit consisting of U50 (divide by 2), counters U62 and U63 (divide by 50,000), and PAL U57 (divide by 2).

PM/FL words are input via front panel connectors P3 and P4, which are then latched at a 12 Hz rate by PALs U51 through U56. These comparators test for a match (MATCH1-MATCH6) between the sampled PM/FL words on the latched outputs of U51 through U56, and the inputs of U51 through U56 which represent the current PM/FL status on P3 and P4.

The 5 MATCHx control signals, representing 8 PM/FL bits each, are fed to PAL U57, which monitors them on a continuous basis. An interrupt is produced when any one of the six MATCHx bits becomes false, which asserts LIRQ and creates an interrupt. When a mismatch is detected, the CLK_IN signal is also disabled and remains high to allow reading of the status register and the PM/FL word change.

Besides providing monitoring of 40 PM/FL bits, an additional input is available on the P5 connector for either a differential or 50 ohm terminated single-ended TTL signal. This input is sampled and represented by MATCH6.

The ICS-330 also allows VMEbus access to a set of registers which are output on front panel connector P5 in differential format. These latching registers consist of U32 through U37, and differential drivers U38 through U43. U35 and U36 implement ENABLE/DISABLE0-9 bits, while U34, U35, U36, and U37 implement HI/LO_BAND_SELECT0-9 bits, as well as three additional outputs: LTS_READY, LTS_TRANSMITTING, and LTS_TRANSMIT_COMPLETE.

The important control signals are discussed below, with reference to Fig. A1.1:

CLKIN - 12 Hz sampling clock for PM/FL words when RUN is asserted. When RUN is not asserted, CLKIN is a 614.4 KHz clock. This provides for faster updating of latched PM/FL words for running diagnostics, or for clearing a previous PM/FL state that caused an interrupt and has already been acknowledged.

MATCH1-MATCH6 - Active high match bits which are asserted as long as successively sampled PM/FL words are the same.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

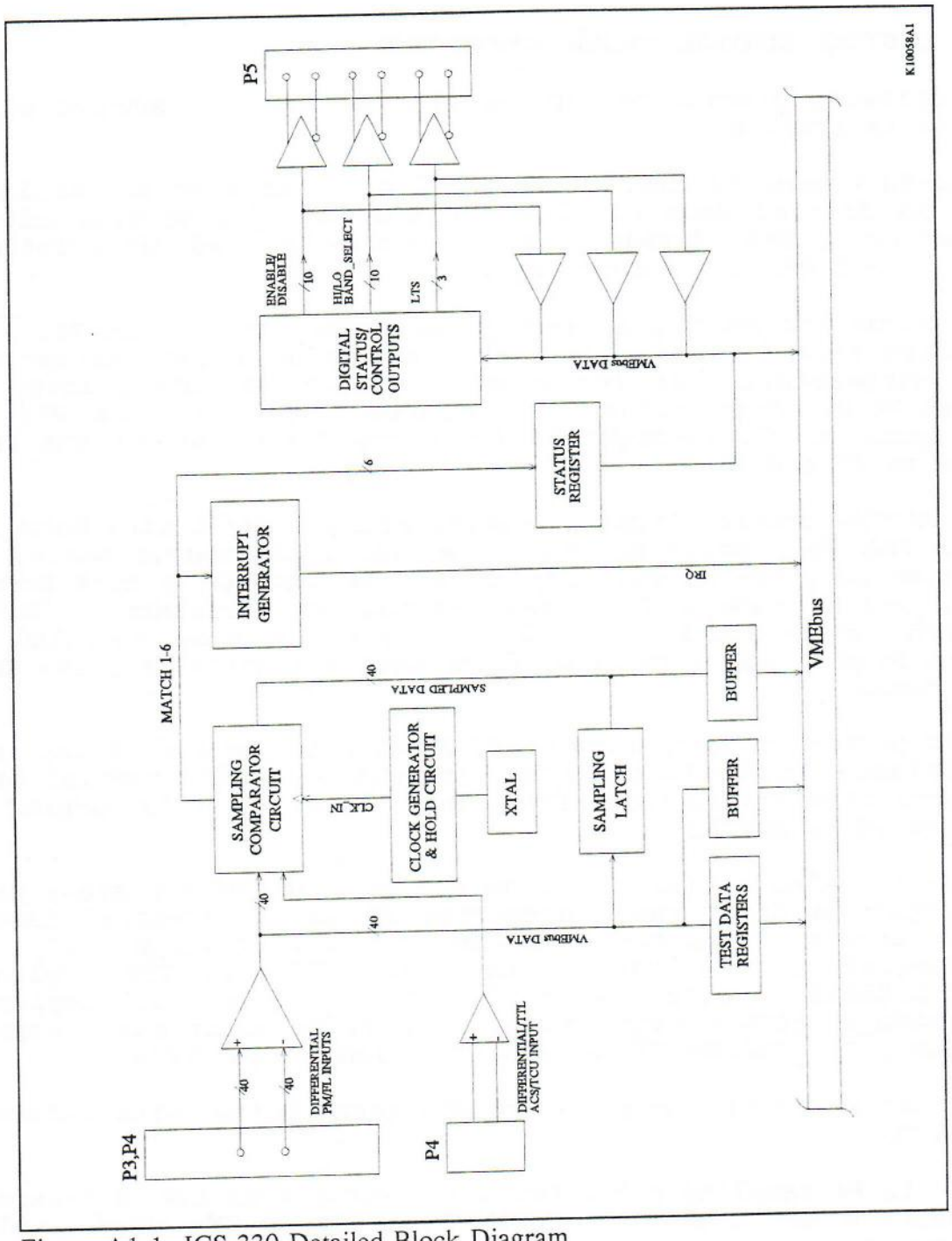



Figure A1.1 ICS-330 Detailed Block Diagram



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

/MATCH - Active low logical NAND of MATCH1-MATCH6. If any of MATCH1-MATCH6 becomes a logical zero, then a mismatch has been detected on at least one of the PM/FL words, signaling an interrupt.

/HOLD - Active low signal which is used to gate CLKIN. When any of MATCH1-MATCH6 becomes invalid, /HOLD is asserted, and causes CLKIN to remain high. This prevents the condition that triggers an interrupt from changing until the changed PM/FL word has been identified. To re-enable CLKIN, RUN must be cleared and then re-asserted.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## APPENDIX 2 TROUBLESHOOTING

### A2.1 TROUBLESHOOTING PROCEDURES

A failure of the BIT (built-in-test) is likely to result from a hardware failure of the ICS-330 card. The following discussion provides a brief guide for emergency diagnosis of any problem with the ICS-330, and should be used in conjunction with the circuit description of Appendix 1 to expedite any repair.

Before trying to troubleshoot the board, it should be ensured that VMEbus DIP switches which control board address, IRQ level, interrupt vector and address modifier code response are properly set.

A number of test points are available for monitoring some of the more important signals. These test points, referred to in the accompanying text, can be located using Fig. A2.1. In order to gain easy access to the board's test points, use of a VMEbus extender card is strongly recommended.

Troubleshooting the ICS-330 should commence by first testing some of the digital control signals. All testing should be done with the board in RUN mode, with interrupts disabled.

After power-up, the PM/FL inputs should first be checked by applying levels to the P3, P4, and P5 connector inputs and then reading the current data and latched data registers. This should be done with the RUN bit still low.

While performing this test, the following test points should be checked with a logic analyzer:

1. TP6, TP7, TP8 (/RD_CUR1, /RD_CUR2, /RD_CUR3)

   These three signals control reading of the three current data registers from VMEbus. Performing a read operation on the appropriate register should bring the corresponding control signal low for a few hundred nanoseconds.

2. TP9, TP10, TP11 (/RD_D1, /RD_D2, /RD_D3)

   These three signals control reading of the three latched data registers from VMEbus. Performing a read operation on the appropriate register should bring the corresponding control signal low for a few hundred nanoseconds.

3. TP12 - TP17 (MATCH1-MATCH6)

   As described in section 1.3.2, the 6 MATCHx bits monitor changes at the PM/FL inputs, and cause an interrupt when such a change causes the current and latched data registers to



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

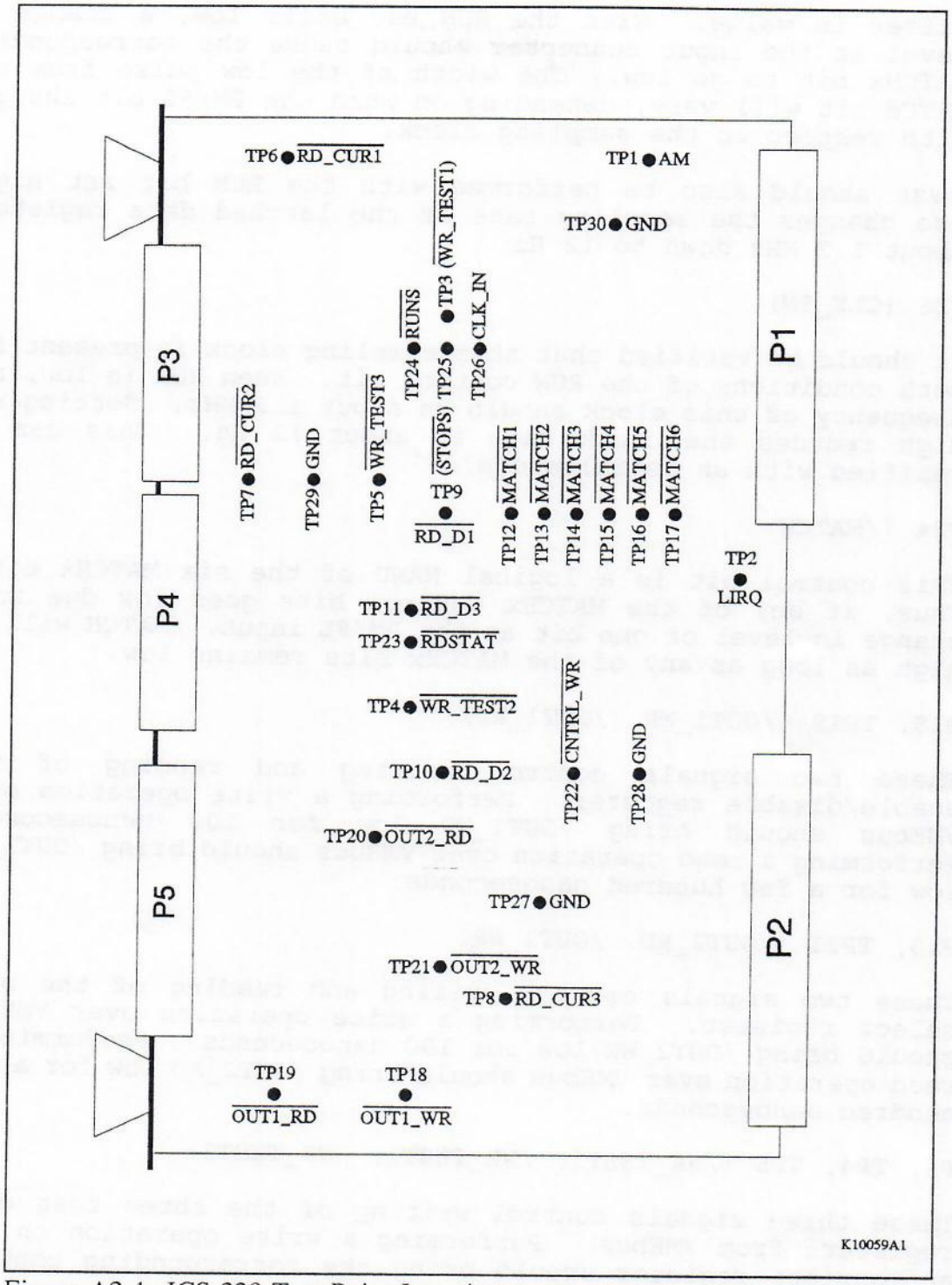



Figure A2.1 ICS-330 Test Point Locations



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

differ in value. With the RUN bit still low, a change in level at the input connector should cause the corresponding MATCHx bit to go low. The width of the low pulse from the MATCH bit will vary, depending on when the PM/FL bit changed with respect to the sampling clock.

This test should also be performed with the RUN bit set high. Doing so changes the sampling rate of the latched data registers from about 1.2 MHz down to 12 Hz.

4. TP26 (CLK_IN)

It should be verified that this sampling clock is present for both conditions of the RUN control bit. When RUN is low, the frequency of this clock should be about 1.2 MHz. Setting RUN high reduces the clock rate to about 12 Hz. This can be verified with an oscilloscope.

5. TP24 (/MATCH)

This control bit is a logical NAND of the six MATCHx bits. Thus, if any of the MATCHx control bits goes low due to a change in level of one bit at the PM/FL input, /MATCH will go high as long as any of the MATCHx bits remains low.

6. TP18, TP19 (/OUT1_WR, /OUT1_RD)

These two signals control writing and reading of the enable/disable register. Performing a write operation over VMEbus should bring /OUT1_WR low for 100 nanoseconds. Performing a read operation over VMEbus should bring /OUT1_RD low for a few hundred nanoseconds.

7. TP20, TP21 (/OUT2_RD, /OUT2_WR)

These two signals control writing and reading of the band select register. Performing a write operation over VMEbus should bring /OUT2_WR low for 100 nanoseconds. Performing a read operation over VMEbus should bring /OUT2_RD low for a few hundred nanoseconds.

8. TP3, TP4, TP5 (/WR_TEST1, /WR_TEST2, /WR_TEST3)

These three signals control writing of the three test data registers from VMEbus. Performing a write operation on the appropriate register should bring the corresponding control signal low for 100 nanoseconds.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

## A2.2 TEST EQUIPMENT

The following is a list of test equipment needed for diagnosing and repairing the ICS-330 card. It is assumed that the card is installed in a SUN computer (or any VMEbus computer) and appropriate software is available to perform the necessary board functions.

1. 100 MHz 2-channel oscilloscope

   Many of the control signals are of a repetitive nature and can be viewed with an oscilloscope. All of the analog signals should be viewed with an oscilloscope.

2. 10 nsec sampling logic analyzer

   Some of the control signals are generated by VMEbus accesses and must be viewed by triggering on a logic level with a logic analyzer. A logic analyzer is also useful for comparing some of the control signals with the clock signal.

3. VMEbus extender card

   An appropriate extender card is necessary for gaining access to the test points and other signals while the card is powered-up in a VMEbus chassis.

4. Digital Multimeter

   A multimeter is useful for verifying both digital and analog supply voltages and other fixed levels.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

**Artisan Technology Group** is an independent supplier of quality pre-owned equipment

## Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

## We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

## Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP