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

Find the *Curtiss-Wright / Pentland Systems MPV991* at our website: ***Click HERE***

Product Manual

MPV991

Timer/Counter

Board

PENTLAND SYSTEMS LIMITED

MPV 991

Pentland Systems Limited
1 Cochrane Square
Brucefield Industrial Park
LIVINGSTON
West Lothian
Scotland
EH54 9DR

Telephone: **(0506) 464666**
Fax No: **(0506) 463030**

Additional copies of this manual are available at a nominal charge from Pentland Systems Limited at the above address. Please state clearly which manual(s) you require.

PENTLAND SYSTEMS LIMITED

MPV991 OPERATING MANUAL

Pentland Systems is committed to providing products of the highest quality that meet or exceed customers' requirements. We have taken great care to ensure that this product will perform satisfactorily. If you should have any difficulty, Pentland Systems Applications Engineers are available to assist you. Our address, fax and phone numbers are on the preceeding page.

The information provided in this manual is believed to be helpful, accurate and reliable; however Pentland Systems Limited assumes no responsibility for errors or omissions. Pentland Systems has no control over the use of this information, and all such use shall be entirely at the users own risk. Prices and specifiications are subject to change without notice. No patent rights or licences to any of the circuits described herein are implied or granted to any third party. Pentland Systems Limited does not authorise or warrant any Pentland Systems product for use in life support devices or systems.

Should it become necessary to return this product for repair, please refer to the return procedures at the end of this manual.

CHANGE HISTORY

Document Number: P60963

ECO No.	10289	10372	10524	10967	11023	11267	11833
ISSUE	DEC 88	JAN 89	MAY 89	SEPT 90	OCT 90	JUL 91	OCT 93
REVISION	4.A	4.B	4.C	4.D	4.E	4.F	04G

CONTENTS

CHAPTER 1 - INTRODUCTION

This manual is intended as an introduction to the Pentland Systems MPV991 Timer/Counter Board with Quadrature Detection Circuitry.

It consists of a general description of the board and its functions, and a guide to its configuration, installation and operation.

It does not include a detailed technical discussion of individual sections of the board. Readers requiring more information on the System Timing Controllers (STCs) and Universal Interrupt Controller (UIC) used are directed towards the Technical Manuals for these devices which accompany this manual.

CHAPTER 2 - GENERAL DESCRIPTION

2.1 INTRODUCTION

Chapter 2 includes a brief description of the MPV991 VMEbus Timer/Counter Board, a summary of its key features and a full technical specification.

2.2 KEY FEATURES

- **10 General Purpose 16 bit Timer/Counter Channels (2 x Am9513A)**
- **8 Digital Input Channels**
- **8 Gates**
- **8 Tri-state Output Channels**
- **Interrupt Capability via VME Priority Interrupt Bus (Am9519A)**
- **Quadrature Detection Circuitry**
- **5 Internal Frequency Sources**
- **Schmitt-trigger inputs with the facility to add low pass filtering avoids erroneous counting caused by glitches at input.**

2.3 DESCRIPTION OF MPV991

The MPV991 VMEbus Timer/Counter Board is supplied on a single 6U Card and complies with VMEbus specification Rev. C.1.

The MPV991 is based around two Am9513A System Timing Controllers (STCA and STCB) each of which provides 5 general purpose 16 bit Timer/Counters. These 10 counters may be individually programmed to count UP or DOWN in either BCD or BINARY.

In addition each counter may be programmed to select its input from either one of four internal frequency sources (based on the 4MHz clock frequency) or one of four external pins associated with each STC, and to select active HIGH or active LOW input polarity. Both hardware and software gating of each counter is available, and each gate input may control up to three of its associated counters simultaneously.

The counters can be individually programmed to provide a pulse or toggled output directed to the outside world via a software enabled tri-state buffer. Counter outputs are also applied to an AM9519A Universal Interrupt Controller (UIC) which may be enabled to interrupt the host CPU via the VME Priority Interrupt Bus.

The accumulated count for each counter may be read at any time without disturbing the counting process. Any combination of counters for a given STC may be concatenated to provide an effective counter length up to 80 bits.

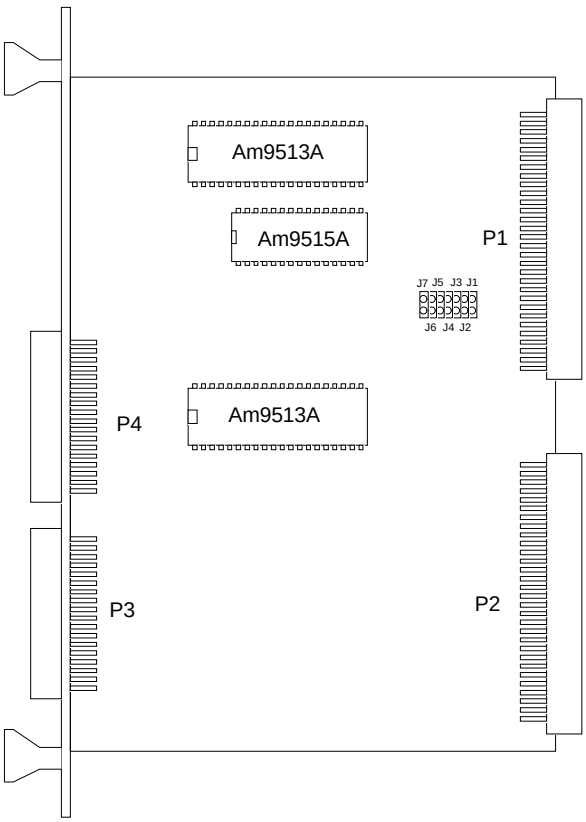
Quadrature detector circuits associated with each STC accept SINE and COSINE outputs from measurement devices such as optical shaft encoders and produce CW and CCW pulses which may be applied to one of the counters in order to determine the current device position. Up to four such devices may be connected to a single MPV991.

Communication with the MPV991 is achieved via the two 37 way D Type connectors P3 and P4 on the front panel (P3 associated with STCA and P4 associated with STCB) or alternatively via the P2 connector on the backplane. The pinouts for these connectors are detailed in Section 4.5.1 Connector Pinouts.

Two standard versions of the MPV991 exist. They differ in the Non VME Digital Input lines interface. An MPV991D has differential line receivers terminated by 100 ohm, 1/2W resistors as shown in Figure 2.2. The base model MPV991 employs standard 74LS14 hex schmidt trigger invertors to interface to the outside world. All other aspects of the boards are identical. Sections 2.4 and 4.5 detail the differences.

The MPV991D is recommended for connection to devices which may be remote or located in noisey environments. For best results , line to line termination impedance should be adjusted to match the cable impedance used.

FIGURE 2.1 - MPV991 Timer/Counter Board Layout



2.4 SPECIFICATION

2.4.1 Functional

8 Digital Inputs
8 Gates
8 Tri-state Outputs
10 Counter/Timer Channels
8 Interrupt Channels
4 Quadrature Detection Channels
8 8/16/32 Bit Interrupt Vectors
1K Block of Short I/O address space occupied
7 VME Interrupt Levels

2.4.2 Electrical

2.4.2.1 Non VME Digital Input Lines (MPV991)

	MIN.	TYP.	MAX.	
POSITIVE-GOING THRESHOLD VOLTAGE	1.4	1.6	1.9	V
NEGATIVE-GOING THRESHOLD VOLTAGE	0.5	0.8	1.0	V
HYSTERESIS	0.4	0.8		V
INPUT CURRENT AT POSITIVE GOING THRESHOLD		-0.14		mA
INPUT CURRENT AT NEGATIVE GOING THRESHOLD		-0.18		mA
HIGH LEVEL INPUT CURRENT			0.02	mA
LOW LEVEL INPUT CURRENT			-0.4	mA

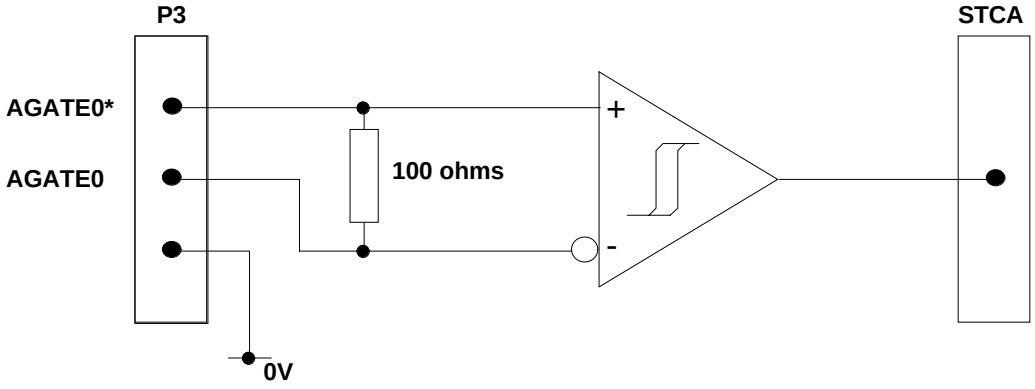
2.4.2.2 Balanced Differential Line Receiver

Part No. : 26LS32 ACN

Line to line termination impedence : 100Ω [±5%, 1/2W Metal Film (R15 - R30)]

	MIN.	TYP.	MAX.	
DIFFERENTIAL INPUT HIGH THRESHOLD VOLTAGE			0.2	V
DIFFERENTIAL INPUT LOW THRESHOLD VOLTAGE	-0.2			V
HYSTERESIS		50		mV

FIGURE 2.2 -Differential Line Receiver



2.4.2.3 Non VME Digital Output Lines

	MIN.	TYP.	MAX.	
HIGH LEVEL OUTPUT VOLTAGE	2	3.4		V
LOW LEVEL OUTPUT VOLTAGE			0.5	V
HIGH LEVEL OUTPUT CURRENT			-15	mA
LOW LEVEL OUTPUT CURRENT			24	mA

2.4.2.4 +5V Supply

+5V is available on P2, P3, and P4. This supply (+5V[f]) is fuse-protected at 1A and is obtained from the VME Backplane.

2.4.3 Quadrature Detection Circuits

Quadrature detector maximum rotation speed

1,000000/N revs/s

where N is the Line number of the encoder

Maximum count speed : 4MHz

Minimum Input Pulse Duration : 125ns

Power Supply +5 ±0.25V @ 3.0A maximum

2.4.4 Environmental

OPERATING TEMPERATURE	**0 to 60 degrees C**
STORAGE TEMPERATURE	**-25 to +85 degrees C**
RELATIVE HUMIDITY	**5 % to 95% non-condensing**

2.4.5 VMEbus Compliance

a) Complies with VMEbus Specification REV C.1

b) A16 : D08(0) DTB Slave

c) Release on Acknowledge (ROAK) Interrupter

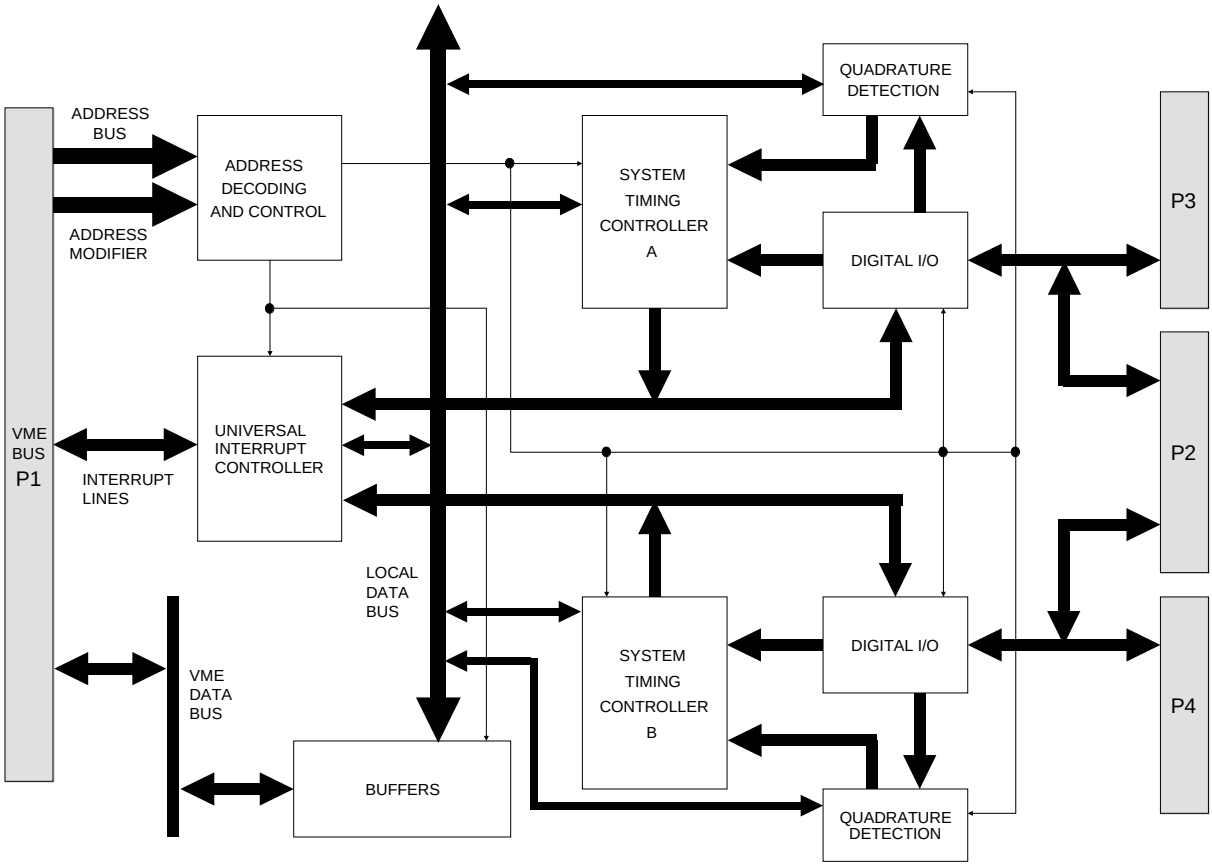
d) Form Factor - Double Height (6U) and Single Width

CHAPTER 3 - FUNCTIONAL DESCRIPTION

3.1 INTRODUCTION

Chapter 3 is an overview of the principles of operation of the MPV991. A detailed description of individual devices is beyond the scope of this manual, and the reader is directed towards the Technical Manuals for the Am 9513A System Timing Controller and the Am 9519A Universal Interrupt Controller for more information.

FIGURE 3.1 - Block Diagram of the MPV991

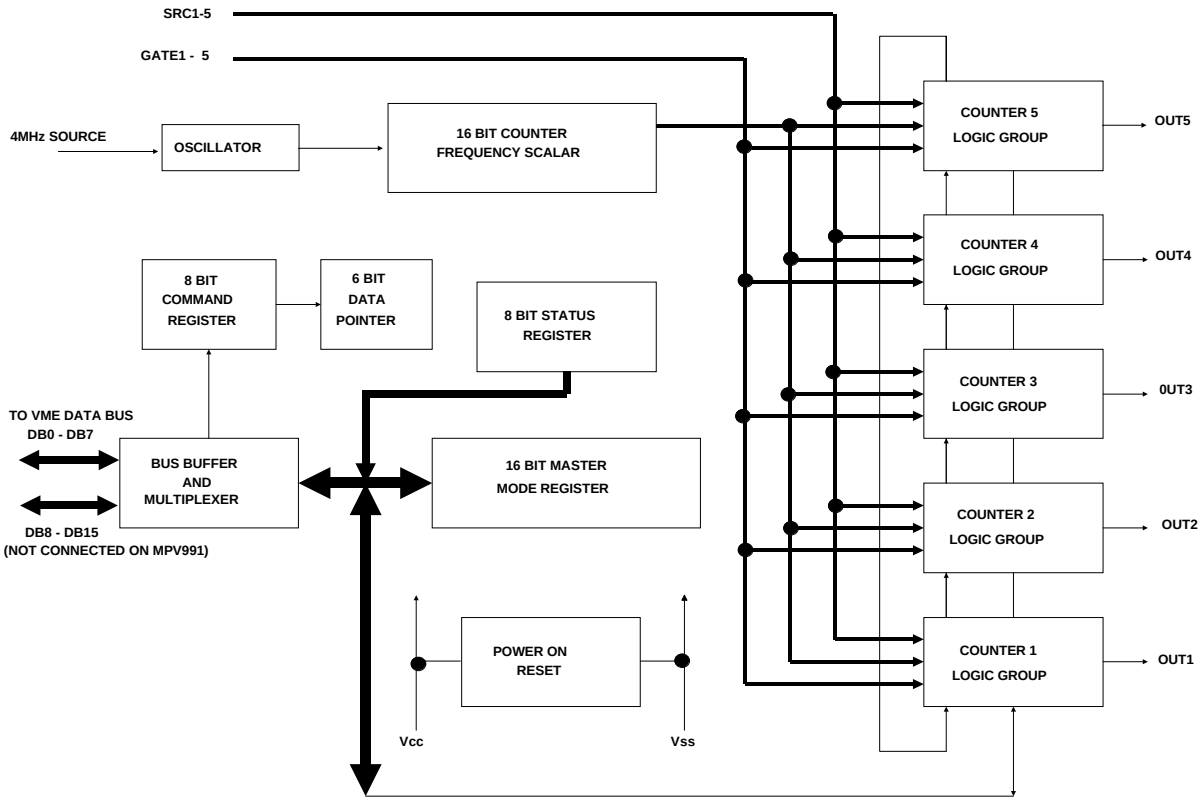


3.2 TIMER/COUNTER SECTION

The heart of the MPV991 is the two Am9513A System Timing Controllers (STC A and STC B). A detailed description of this device may be found in the Am 9513A Technical Manual supplied.

The features of the Am9513A available on the MPV991 are illustrated in Figure 3.2.

FIGURE 3.2 - Block Diagram of the Am9513A STC



Each STC supplies 5 of the 10 available general purpose counters, and has a number of inputs, outputs and registers associated exclusively with it. The operation of one of the two STC's is described in the following sections.

3.2.1 Clock Inputs (CLOCK0 - CLOCK3)

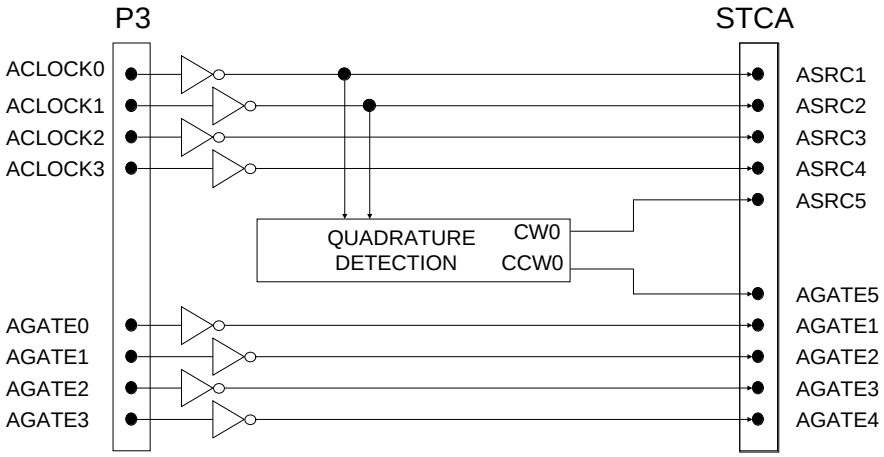
There are 4 Clock Inputs for each STC. When Quadrature Detection is disabled, these inputs are equivalent to SRC1-SRC4 of the STC chip through which external signals may be applied to the counters. Any single Source Input may be routed to any, or all, of the counters. The selected input and its active polarity for each counter is programmed via the COUNTER MODE REGISTER.

When Quadrature Detection is enabled, the Clock Inputs are used to route the COS, SIN and INDEX outputs from the transducer to the Quadrature detection circuits on the MPV991.

3.2.2 Internal Inputs (F1 - F5)

Five Internal frequency sources, based on the 4MHz clock, are available to each STC. These may be applied to any, or all, of the 5 counters. The selected input and its active polarity for each counter is programmed via the COUNTER MODE REGISTER (see Chapter 5).

FIGURE 3.3 - Clock Input Connections



3.2.3 Gate Inputs (GATE0 - GATE3)

Four Gate Inputs are associated with each STC. These may be used to control when and how the counting of an individual counter may proceed, or alternatively Gate pins may be used as count sources for other counters. Gate functions for each counter are programmed via the COUNTER MODE REGISTER.

3.2.4 Tri-state Outputs (OUT0 - OUT3)

Each output is directly associated with an individual counter. Outputs must be enabled by setting Bit 3 of the MPV991 CONTROL REGISTER.

An output reflects the occurrence of a TERMINAL COUNT (TC) from its associated counter. The Ouput may be a single pulse, train of pulses, squarewave or complex waveform depending on the COUNTER MODE REGISTER, the Input Source and the value in the LOAD register.

3.2.5 Counter Logic Groups

Each of the five Counter Logic Groups consists of a 16 bit general purpose counter and its associated control logic, a 16 bit LOAD REGISTER, a 16 bit HOLD REGISTER and a 16 bit COUNTER MODE REGISTER.

Access to these registers is detailed in Chapter 5.

3.2.5.1 Counter Mode Register

Each counter has its own COUNTER MODE REGISTER which is used to controls those functions which can be individually set for individual counters.

3.2.5.2 Load Register

The LOAD REGISTER contains the value loaded into the counter when it is initialised.

When the contents of the LOAD REGISTER is loaded into the counter and the counter is armed, the counter will count from this initial value in the manner defined by the COUNTER MODE REGISTER until one pulse before a zero state is reached.

ie. 0001 for down counting

9999(BCD) for BCD up counting

FFFF(HEX) for Binary up counting

The occurrence of the next pulse at the input produces a TERMINAL COUNT (TC) at the output. If the counter is configured for continuous counting (by the COUNTER MODE REGISTER) then the occurrence of a TERMINAL COUNT coincides with the reloading of the counter from the LOAD REGISTER and counting continues until the next TC, when the process is repeated. Thus the TERMINAL COUNT is said to occur when the counter would have reached a zero state had it not been reloaded, and the TERMINAL COUNT FREQUENCY is the input frequency divided by the value in the LOAD register.

3.2.5.3 Hold Register

The 16 bit HOLD REGISTER serves a dual purpose.

It can be used used to store the accumulated count without disturbing the counting process, which is an essential feature when the counter is to be used in event counting applications.

It is also used as an alternative LOAD source for the counter. In this case the counter is loaded both from the LOAD register and the HOLD register on the occurence of alternate TERMINAL COUNTS.

With different LOAD and HOLD values, complex duty waveforms can be produced.

3.3 INTERRUPT CONTROLLER SECTION

The Am 9519A Universal Interrupt Controller (UIC) provides interrupt capability via the VME Priority Interrupt Bus. A detailed description of the device can be found in the Am 9519A Technical Manual supplied.

The UIC provides any mix of one, two, three or four byte responses to the host CPU during the interrupt acknowledge process. The response bytes are all fully programmable so that any appropriate addressing, vectoring, instruction or other message protocol may be used.

Contention among multiple interrupts is managed internally using either fixed or rotating priority resolution circuitry. The direct vectoring capability of the UIC may be by-passed using the polled mode option.

3.3.1 Interrupt Request Register (IRR)

The 8 bit IRR is used to recognise and store active transitions on any of the Interrupt Request lines. A bit is set whenever the a valid request is received on the corresponding line and is cleared automatically when the interrupt is acknowledged.

3.3.2 Interrupt Service Register (ISR)

The 8 bit ISR is used to store the acknowledge status of individual interrupts. A bit is set when the corresponding interrupt is acknowledged. When the equivalent bit in the Auto Clear Register is set, the ISR bit is cleared before the end of the Interrupt Acknowledge, otherwise the ISR bit must be cleared as part of the service routine.

When fixed priority mode is selected, only higher priority interrupts will be serviced while a bit in the ISR is set. When rotating priority mode is selected, all ISR bits must be clear before an interrupt is serviced.

3.3.3 Interrupt Mask Register (IMR)

The 8 bit IMR is used to disable and enable individual interrupts. When an IMR bit is set it masks the corresponding IRR bit which cannot be serviced until the mask is cleared.

3.4 QUADRATURE DETECTION CIRCUITS.

Each STA has Quadrature Detection Circuits associated with it which can accept the output from up to two position transducers at one time.

This type of transducer typically produces 3 outputs :

SIN output - a squarewave output

COS output - a squarewave output

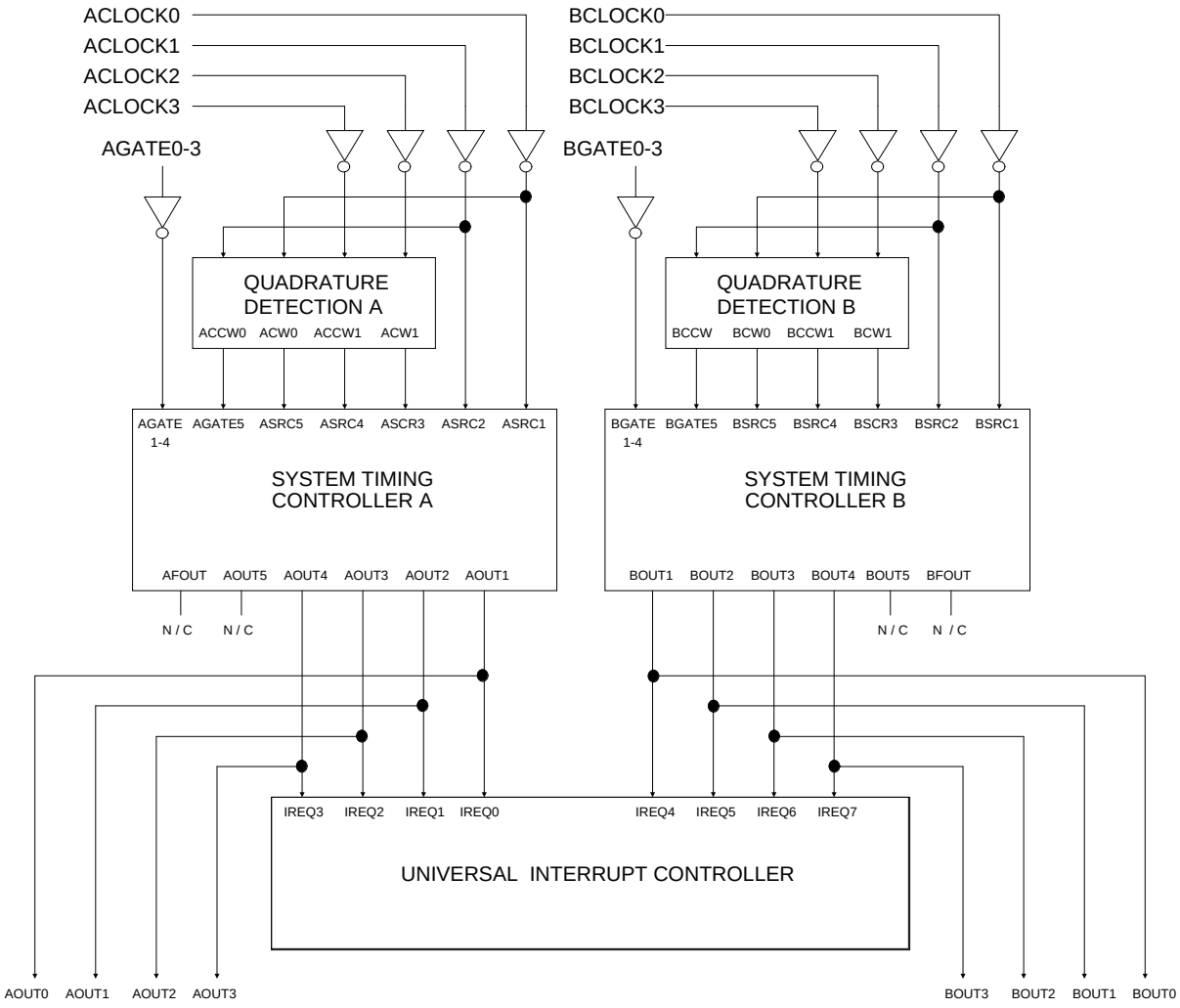
INDEX output - a single pulse

The SIN and COS outputs are Quadrature signals which are 90 degrees out of phase, and thus the direction of movement can be determined by which output lags the other. The positional movement is determined by the number of pulses.

The INDEX output is a single pulse which occurs when the transducer reaches a "zero" position. Not all transducers provide an INDEX pulse which is normally only of interest when using incremental devices.

The Quadrature Detection Circuits of the MPV991 decode the SIN and COS outputs from the transducer to provide a train of "clockwise" pulses for movement in one direction and a train of "counter-clockwise" pulses in the other.

FIGURE 3.4 - Input and Output Connections



When Quadrature Detection is enabled the input and output connections are as above.

When Quadrature Detection is disabled then : CLOCK2 --- SRC3 and CLOCK3 --- SRC4

Quadrature Detection cannot be disabled to SRC5 and GATE5

CHAPTER 4 - SETUP AND INSTALLATION

4.1 INTRODUCTION

Chapter 4 describes the preparation required to configure and connect the MPV991.

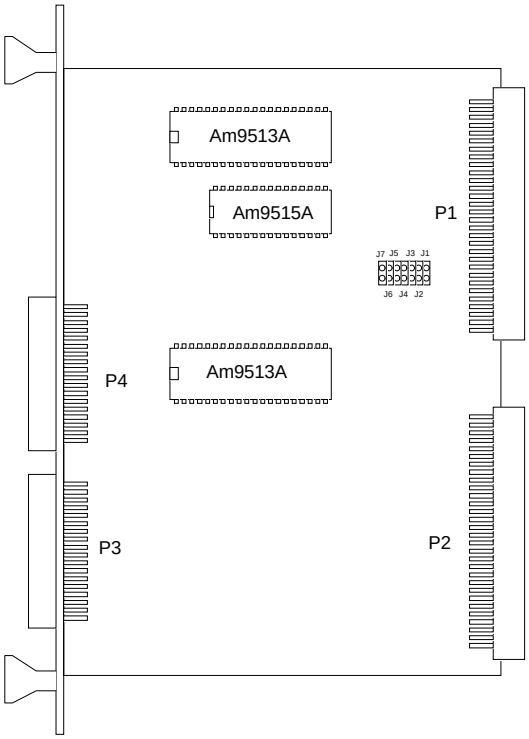
4.2 PREPARING THE BOARD

The MPV991 is sensitive to the build up of static electricity and normal anti-static precautions should be observed when unpacking or handling the board.

4.3 JUMPER CONFIGURABLE OPTIONS

There are 7 user selectable jumpers on the MPV991 and their location is shown in Figure 4.1.

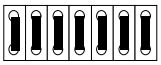
FIGURE 4.1 - Location of Jumpers



J1-J6 are used to select the base address of the MPV991, which always occupies 1K of memory within the VME short address space. J7 selects short supervisory access or short non-priviledged access address modifier.

The MPV991 is shipped from the factory with these jumpers set to default settings. The factory settings are shown in Table 4.1.

TABLE 4.1 - J1-J7 Factory Settings

J7	J6	J5	J4	J3	J2	J1	DEFAULT SETTING
IN	IN	IN	IN	IN	IN	IN	MPV991 BASE ADDRESS : VME SHORT I/O BASE ADDRESS + 0000H
 							SHORT SUPERVISORY ACCESS

4.4 CHANGING THE JUMPER SETTINGS

4.4.1 Base Address Jumpers (J1-J6)

The MPV991 occupies 1K of memory at any one of 64 offset locations within VME short I/O address space. The base address of the VME short I/O address space is dependent on the choice of CPU.

The base address of the MPV991 is the VME short I/O address space base address, plus the offset which is selected using jumpers J1-J6. The jumper settings and their corresponding address offsets are shown in Table 4.2.

ie. If the VME short I/O address space is located at FFFF0000H for a given CPU and jumpers J1 and J2 are removed then the base address of the MPV991 is FFFF0C00H.

4.4.2 Address Modifier Jumper (J7)

A single jumper, J7, is used to select either short supervisory accesses (2D) or both supervisory accesses and non-priviledged accesses (29 and 2D). The jumper setting and its corresponding access response is shown in Table 4.3.

4.5 CONNECTING THE MPV991

Communication with the MPV991 is achieved via the two 37 way “D” Type connectors on the front panel. Mating connectors are Cannon DC37P/1A1N or equivalent.

P3 is associated with STCA and P4 is associated with STCB. Alternatively the MPV991 can be accessed via the P2 connector on the Backplane. The location of these connectors is shown in Figure 4.1.

Note

P3/P4 and P2 are hardwire interconnected. If connection to the MPV991 is via P3/P4, ensure that no interference signals are present on P2 prior to installation. P3/P4 signals present on the P2 connector during installation may cause damage to the MPV991 or sub-bus boards.

TABLE 4.2 - Base Address Settings

JUMPERS (J6 - J1)						ADDRESS OFFSET
J6	J5	J4	J3	J2	J1	
IN	IN	IN	IN	IN	IN	0000H
IN	IN	IN	IN	IN	OUT	0400H
IN	IN	IN	IN	OUT	IN	0800H
IN	IN	IN	IN	OUT	OUT	0C00H
IN	IN	IN	OUT	IN	IN	1000H
IN	IN	IN	OUT	IN	OUT	1400H
IN	IN	IN	OUT	OUT	IN	1800H
IN	IN	IN	OUT	OUT	OUT	1C00H
IN	IN	OUT	IN	IN	IN	2000H
IN	IN	OUT	IN	IN	OUT	2400H
IN	IN	OUT	IN	OUT	IN	2800H
IN	IN	OUT	IN	OUT	OUT	2C00H
IN	IN	OUT	OUT	IN	IN	3000H
IN	IN	OUT	OUT	IN	OUT	3400H
IN	IN	OUT	OUT	OUT	IN	3800H
IN	IN	OUT	OUT	OUT	OUT	3C00H
IN	OUT	IN	IN	IN	IN	4000H
IN	OUT	IN	IN	IN	OUT	4400H
IN	OUT	IN	IN	OUT	IN	4800H
IN	OUT	IN	IN	OUT	OUT	4C00H
IN	OUT	IN	OUT	IN	IN	5000H
IN	OUT	IN	OUT	IN	OUT	5400H
IN	OUT	IN	OUT	OUT	IN	5800H
IN	OUT	IN	OUT	OUT	OUT	5C00H
IN	OUT	OUT	IN	IN	IN	6000H
IN	OUT	OUT	IN	IN	OUT	6400H
IN	OUT	OUT	IN	OUT	IN	6800H
IN	OUT	OUT	IN	OUT	OUT	6C00H
IN	OUT	OUT	OUT	IN	IN	7000H
IN	OUT	OUT	OUT	IN	OUT	7400H
IN	OUT	OUT	OUT	OUT	IN	7800H
IN	OUT	OUT	OUT	OUT	OUT	7C00H

						ADDRESS OFFSET
J6	J5	J4	J3	J2	J1	
OUT	IN	IN	IN	IN	IN	8000H
OUT	IN	IN	IN	IN	OUT	8400H
OUT	IN	IN	IN	OUT	IN	8800H
OUT	IN	IN	IN	OUT	OUT	8C00H
OUT	IN	IN	OUT	IN	IN	9000H
OUT	IN	IN	OUT	IN	OUT	9400H
OUT	IN	IN	OUT	OUT	IN	9800H
OUT	IN	IN	OUT	OUT	OUT	9C00H
OUT	IN	OUT	IN	IN	IN	A000H
OUT	IN	OUT	IN	IN	OUT	A400H
OUT	IN	OUT	IN	OUT	IN	A800H
OUT	IN	OUT	IN	OUT	OUT	AC00H
OUT	IN	OUT	OUT	IN	IN	B000H
OUT	IN	OUT	OUT	IN	OUT	B400H
OUT	IN	OUT	OUT	OUT	IN	B800H
OUT	IN	OUT	OUT	OUT	OUT	BC00H
OUT	OUT	IN	IN	IN	IN	C000H
OUT	OUT	IN	IN	IN	OUT	C400H
OUT	OUT	IN	IN	OUT	IN	C800H
OUT	OUT	IN	IN	OUT	OUT	CC00H
OUT	OUT	IN	OUT	IN	IN	D000H
OUT	OUT	IN	OUT	IN	OUT	D400H
OUT	OUT	IN	OUT	OUT	IN	D800H
OUT	OUT	IN	OUT	OUT	OUT	DC00H
OUT	OUT	OUT	IN	IN	IN	E000H
OUT	OUT	OUT	IN	IN	OUT	E400H
OUT	OUT	OUT	IN	OUT	IN	E800H
OUT	OUT	OUT	IN	OUT	OUT	EC00H
OUT	OUT	OUT	OUT	IN	IN	F000H
OUT	OUT	OUT	OUT	IN	OUT	F400H
OUT	OUT	OUT	OUT	OUT	IN	F800H
OUT	OUT	OUT	OUT	OUT	OUT	FC00H

TABLE 4.3 - Access Response

J7	ADDRESSING MODE	MODIFIER
IN	SHORT SUPERVISORY ACCESS	2D
OUT	BOTH SUPERVISORY ACCESS AND NON-PRIVILEDGED ACCESS	29 & 2D

4.5.1 Connector Pinouts

FIGURE 4.2 - Pinout for Connector P2

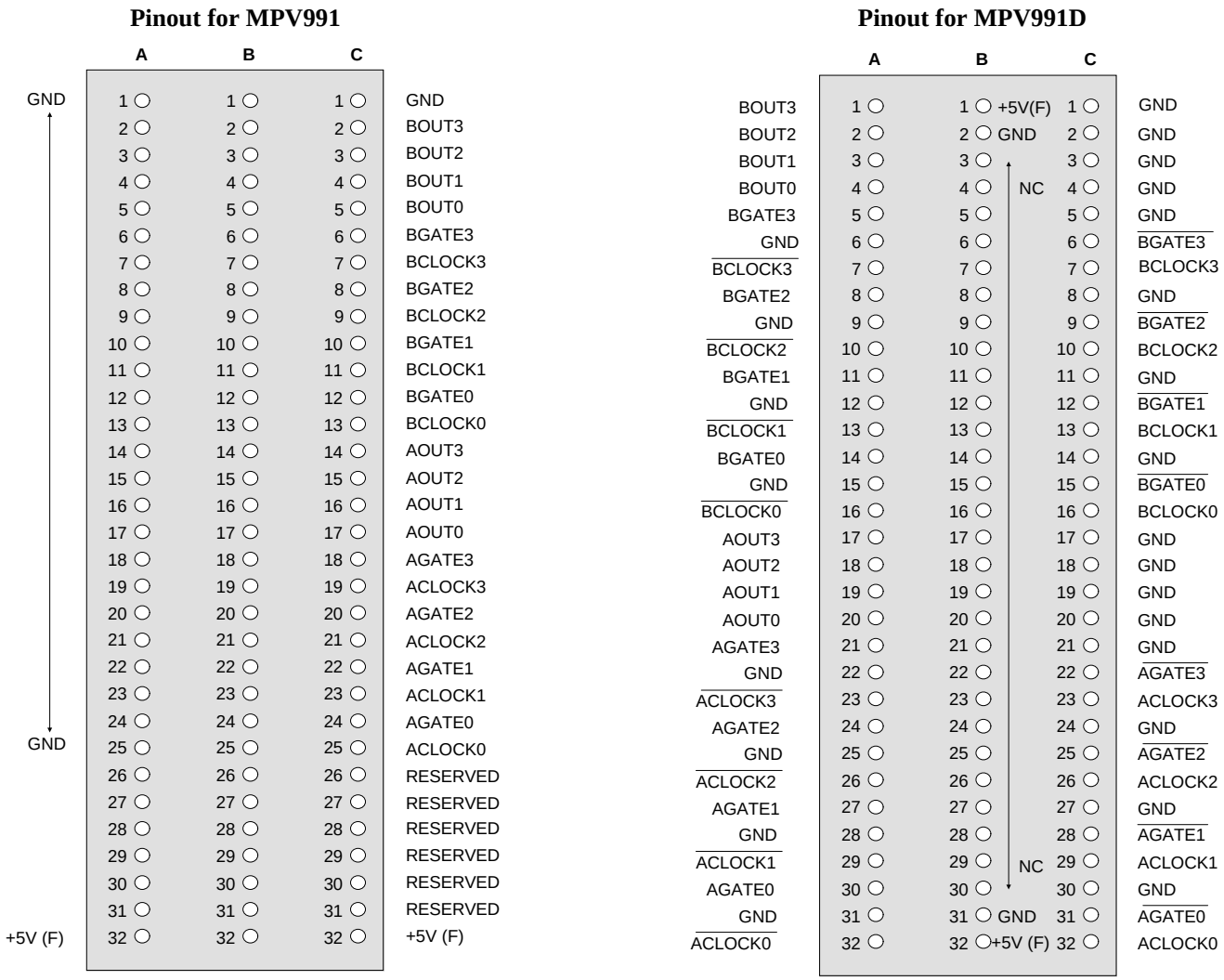
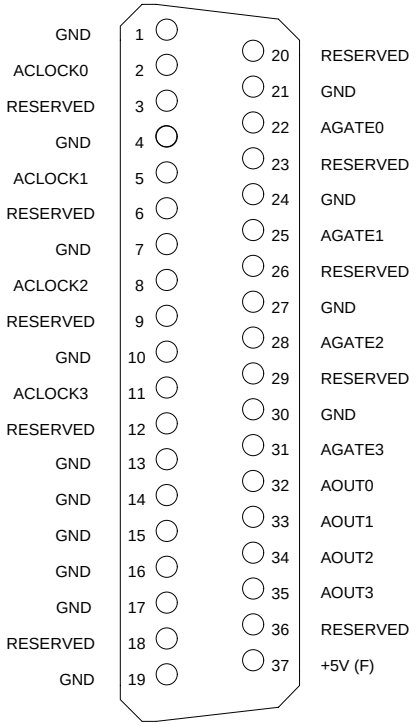


FIGURE 4.3 - Pinout for Connector P3

Pinout for MPV991



Pinout for MPV991D

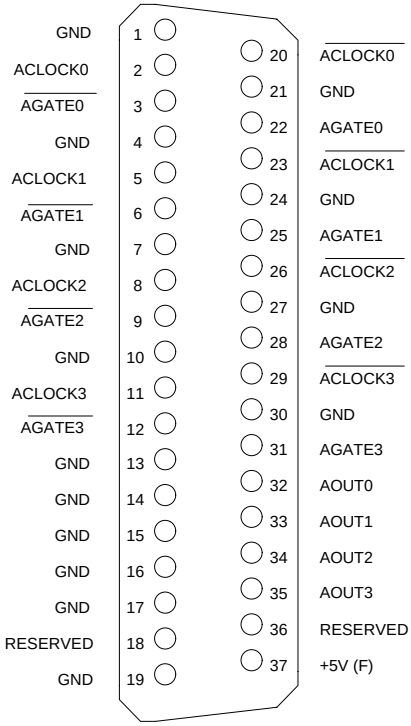
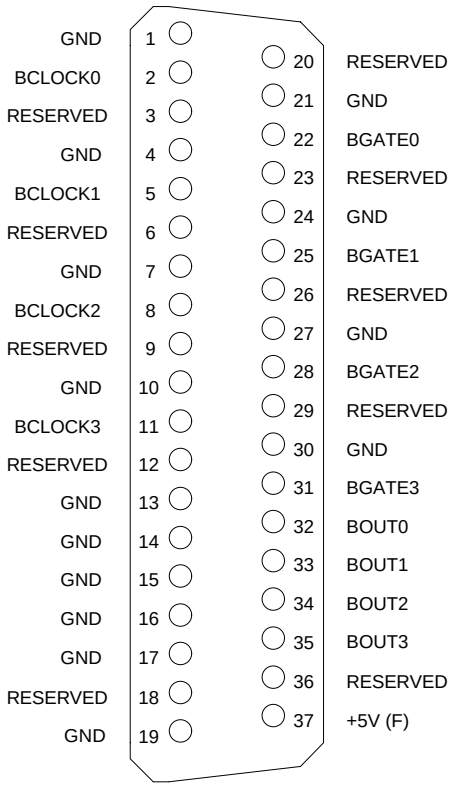
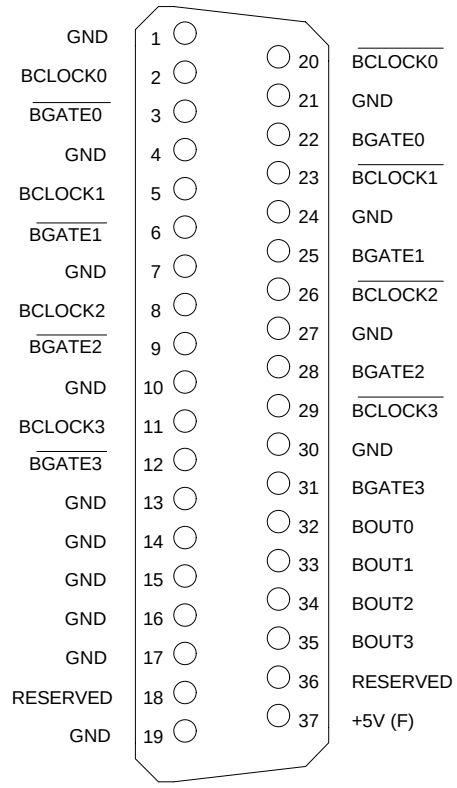


FIGURE 4.4 - Pinout for Connector P4

Pinout for MPV991



Pinout for MPV991D



4.5.2 Connecting a Transducer to the MPV991

When the MPV991 is to be used for position measurement, Quadrature detection should be enabled and the transducer connected as follows :

The SIN output from the transducer should be connected to one of the EVEN clock inputs on the MPV991 (ACLOCK0, ACLOCK2, BCLOCK0, BCLOCK2)

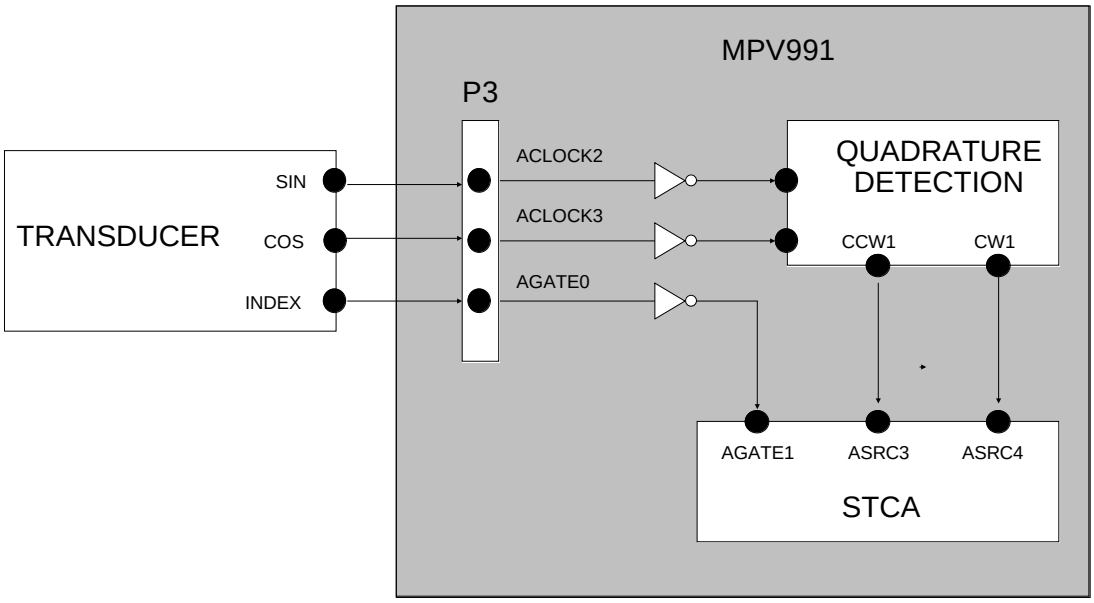
The COS output from the transducer should be connected to the ODD clock input to the MPV991 which corresponds to the clock input selected for the SIN input. (ACLOCK1 for ACLOCK0, ACLOCK3 for ACLOCK2, BCLOCK1 for BCLOCK0, BCLOCK3 for BCLOCK2).

The INDEX output (if available) should be connected to any one of the GATE inputs on the corresponding STC. (AGATE0-3 for STCA and BGATE0-3 for STCB)

Figure 4.6 shows one possible arrangement for a single transducer connected to the MPV991, with Quadrature Detection **enabled**.

When using an MPV991D, connection method is as above. Care should be taken when determining required polarity of signal. If the non inverted output from the transducer is connected to the non inverted input of the MPV991D then there **will be** an invertion between input and STC chip as shown in Figure 4.8. This was done to maintain consistency between MPV991 and MPV991D.

FIGURE 4.5 - Connecting a Transducer to the MPV991



CHAPTER 5 - OPERATING THE MPV991

5.1 INTRODUCTION

Chapter 5 describes the registers and commands which are used to control and operate the MPV991.

For a more detailed description of the registers associated with the STC's and UIC, please refer to the Technical manuals provided.

5.2 CHECKLIST

Before attempting to operate the MPV991, check that the following steps have been completed.

1. Select the desired base address for the MPV991 using Jumpers J1-J6.

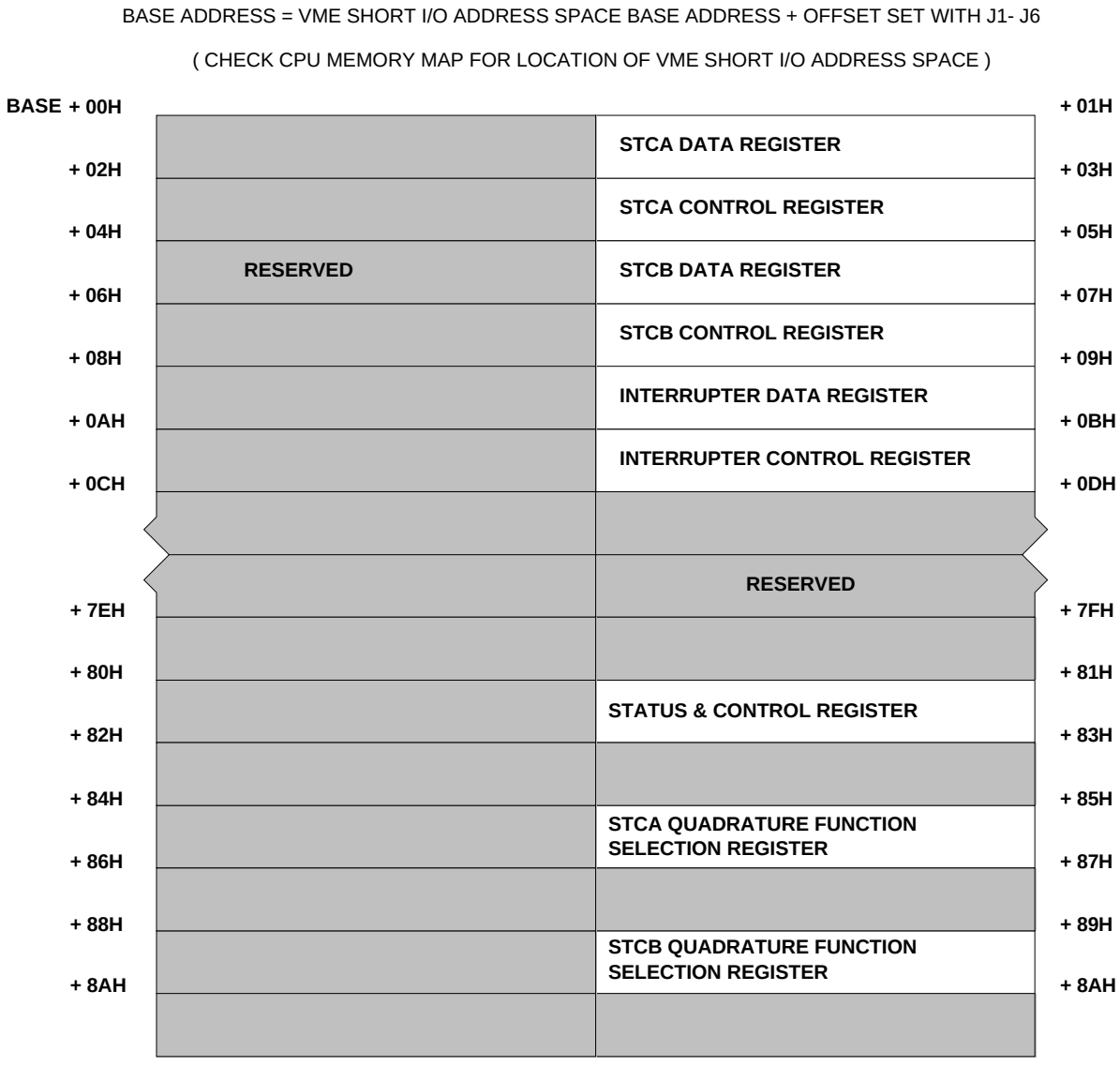
2. Select the desired access response using Jumper J7.

3. Install the MPV991 in the VMEbus Backplane with a host CPU.

4. If the MPV991 is to receive external inputs or provide external signals, it should be connected via connectors P2, P3 or P4.

5.3 MEMORY MAP

FIGURE 5.1 - Memory Map of the MPV991



5.4 MPV991 CONTROL/STATUS REGISTER

The MPV991 has a dual purpose 8 bit Control/Status Register which is located at the address BASE+81H. (see Figure 5.1.)

Figure 5.2 summarises the results of READ and WRITE operations on this register.

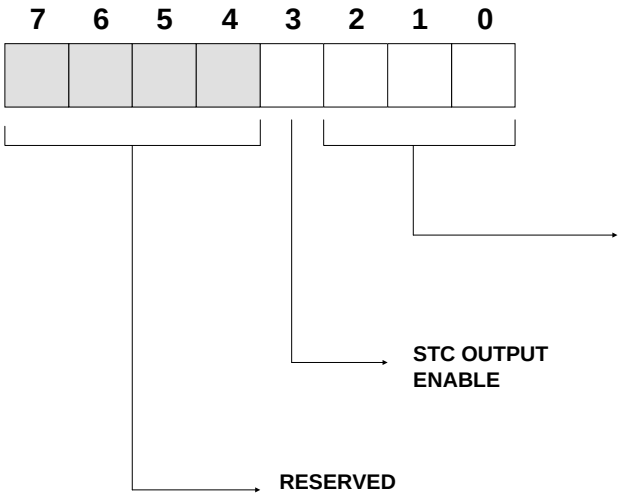
WRITING to this location accesses the CONTROL Register which is used to set the VME Interrupt Level (IRQ1 - IRQ7) and to enable the outputs of STCA and STCB.

FIGURE 5.2 - Summary of READ and WRITE Operations

READ	STATUS REGISTER	INDICATES DIRECTION DETECTED BY QUADRATURE DETECTION CIRCUITS
WRITE	CONTROL REGISTER	SET VME INTERRUPT LEVEL AND ENABLE STC OUTPUTS

READING from this location accesses the STATUS Register which indicates the direction of movement detected by each channel when the Quadrature Detection circuitry is in use.

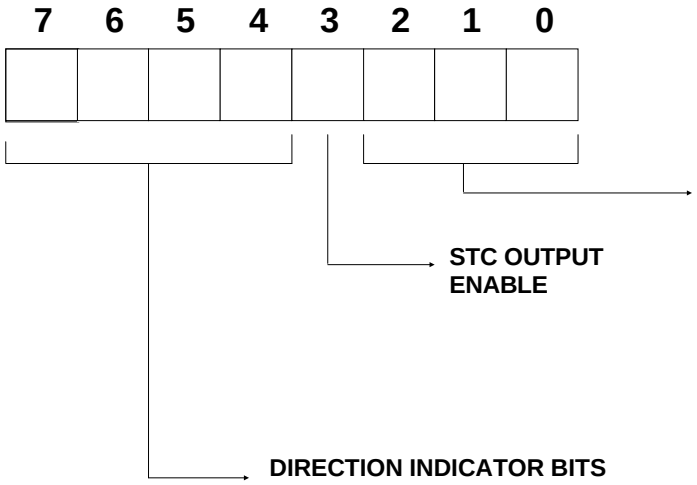
FIGURE 5.3 - CONTROL register (WRITE only)



BIT			INTERRUPT LEVEL
2	1	0	
0	0	0	NONE
0	0	1	IREQ1
0	1	0	IREQ2
0	1	1	IREQ3
1	0	0	IREQ4
1	0	1	IREQ5
1	1	0	IREQ6
1	1	1	IREQ7

5.5 PROGRAMMING THE SYSTEM TIMING CONTROLLERS

FIGURE 5.4 - STATUS Register (READ only)



VME INTERRUPT LEVEL SELECT

BIT 2	BIT 1	BIT 0	INTERRUPT LEVEL
0	0	0	NONE
0	0	1	IREQ1
0	1	0	IREQ2
0	1	1	IREQ3
1	0	0	IREQ4
1	0	1	IREQ5
1	1	0	IREQ6
1	1	1	IREQ7

	CHANNEL NUMBERS	BIT NUMBERS	CW BIT SETTING	CCW BIT SETTING
STC A	0 & 1	4	1	0
	2 & 3	5	1	0
STC B	0 & 1	6	1	0
	2 & 3	7	1	0

Each STC is addressed by the user as only two locations: a CONTROL port and a DATA port. (see Memory Map in Figure 5.1.)

These ports are used to access and control the operation of each of the 5 general counters and their own associated registers.

Detailed information concerning the operation of these registers can be found in the Am9513A Technical Manual supplied.

Each WRITE access to the CONTROL port transfers the data written to the COMMAND register.

Each READ access to the CONTROL port transfers the data read from the STATUS register.

5.5.1 Command Register

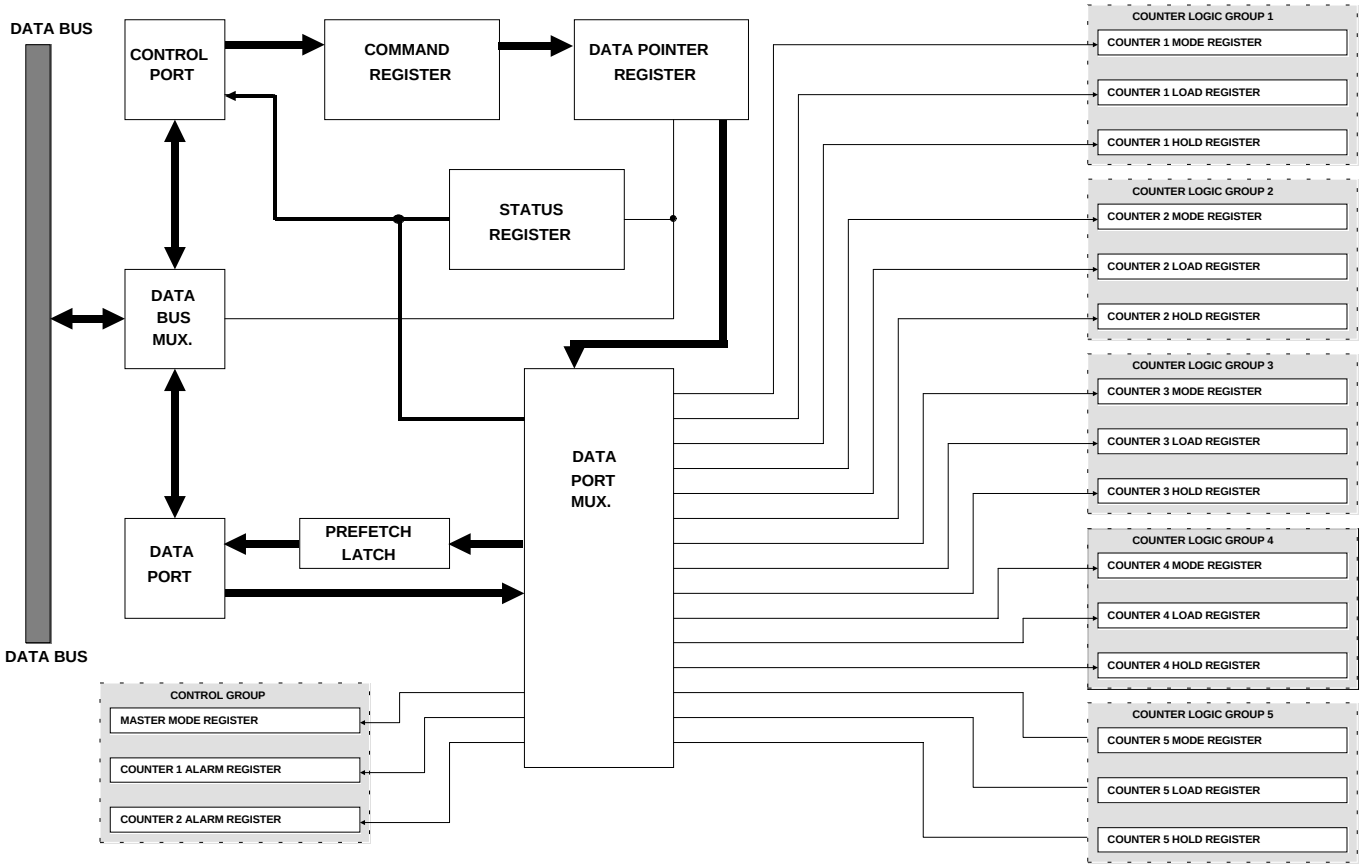
The COMMAND register provides direct control over each of the 5 general counters. It is also used to load the DATA POINTER register with the destination of the data next written to the DATA port.

5.5.2 Data Pointer Register

The destination of data written to the DATA port is controlled by the 6 bit DATA POINTER register.

Bit1-Bit3 is the GROUP POINTER which indicates the group of registers to be accessed.

FIGURE 5.5 - STC Register Access



Bit4-Bit5 is the ELEMENT POINTER which selects the individual register within the group which is accessed.

Bit0 of the DATA POINTER is the BYTE POINTER. As the 16 bit registers are accessed in 8 bit mode, two transfers are necessary to set each register. The BYTE POINTER indicates which byte will be transfered next and is toggled automatically after each transfer.

There are two methods of setting the destination in the DATA POINTER register.

1. Using the COMMAND Register

When Bit5 - Bit7 of the COMMAND register are set to 000, Bit0- Bit4 are used to set the destination stored in the DATA POINTER register.(see Table 5.1 and Figure 5.6)

2. Using Data Pointer Sequencing.

When Data Pointer Sequencing is enabled (Bit14 of the MASTER MODE register set to 0, see Section 5.5.3) the DATA POINTER register is automatically incremented after each data WRITE to point to the next destination register, as defined by the sequence cycles illustrated in Figure 5.7

cycle entered:

If the DATA POINTER is set to point to the MODE register of a counter group the DATA POINTER will subsequently sequence through the ELEMENT cycle.

If the DATA POINTER is set to point to the HOLD register of a counter group the DATA POINTER will subsequently sequence through the HOLD cycle.

TABLE 5.1 - STC Command Register

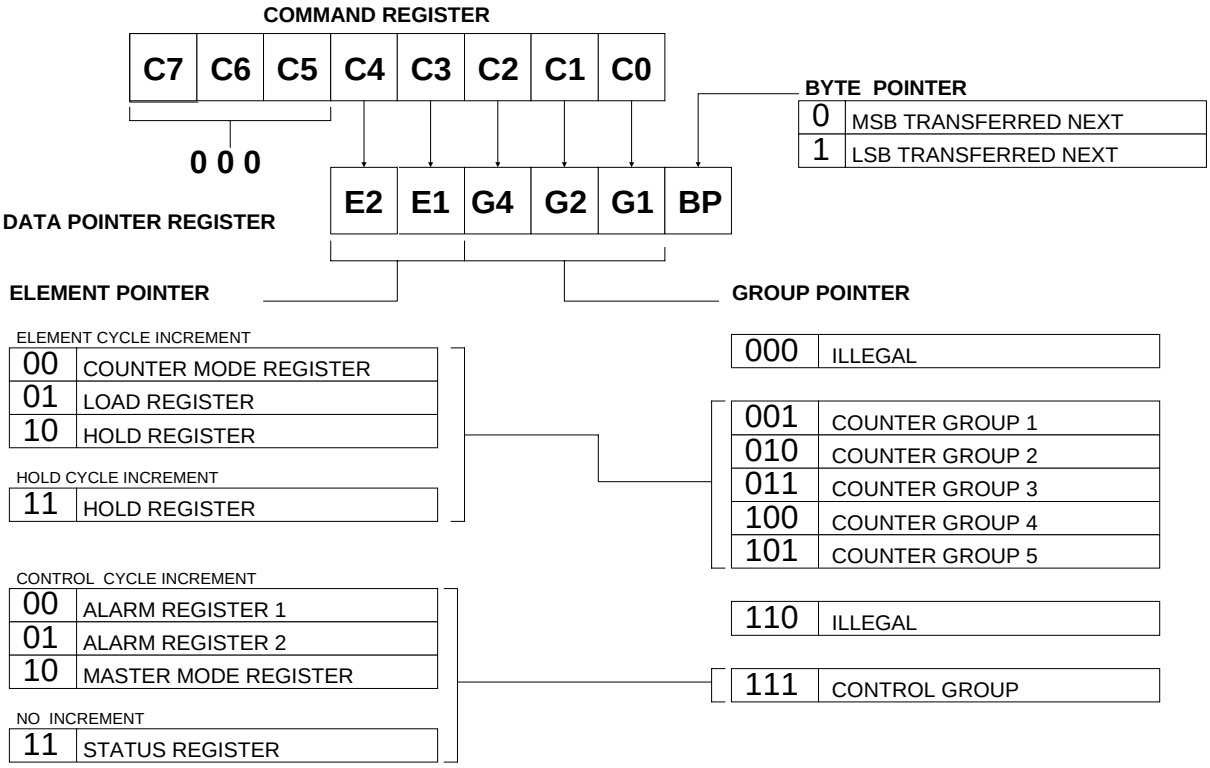
COMMAND CODE								
C7	C6	C5	C4	C3	C2	C1	C0	COMMAND DESCRIPTION
0	0	0	X	X	X	X	X	SET DATA POINTER REG. WITH C0-C4
0	0	1	S5	S4	S3	S2	S1	ARM ALL SELECTED REGISTERS
0	1	0	S5	S4	S3	S2	S1	LOAD SELECTED COUNTERS FROM SPECIFIED SOURCE
0	1	1	S5	S4	S3	S2	S1	LOAD AND ARM SELECTED COUNTERS
1	0	0	S5	S4	S3	S2	S1	DISARM AND SAVE SELECTED COUNTERS
1	0	1	S5	S4	S3	S2	S1	SAVE SELECTED COUNTERS IN HOLD REGISTER
1	1	0	S5	S4	S3	S2	S1	DISARM SELECTED COUNTERS
1	1	1	0	0	N4	N2	N1	CLEAR TOGGLE OUT (LOW) FOR COUNTER N
1	1	1	0	1	N4	N2	N1	SET TOGGLE OUT (HIGH) FOR COUNTER N
1	1	1	1	0	N4	N2	N1	STEP COUNTER N
1	1	1	0	1	0	0	0	DISABLE DATA POINTER SEQUENCING (set MM14)
1	1	1	0	1	1	1	0	GATE OFF FOUT (set MM12)
1	1	1	0	1	1	1	1	ENTER 16 BIT BUS MODE (set MM13) - NOT USED ON 991
1	1	1	0	0	0	0	0	ENABLE DATA POINTER SEQUENCING (clear MM14)
1	1	1	0	0	1	1	0	GATE ON FOUT (clear MM12)
1	1	1	0	0	1	1	1	ENTER 8 BIT BUS MODE (clear MM13)
1	1	1	1	1	0	0	0	ENABLE PREFETCH
1	1	1	1	1	0	0	1	DISABLE PREFETCH
1	1	1	1	1	1	1	1	MASTER RESET

S5	S4	S3	S2	S1	COUNTER
0	0	0	0	1	1
0	0	0	1	0	2
0	0	1	0	0	3
0	1	0	0	0	4
1	0	0	0	0	5

N4	N2	N1	COUNTER
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5

5.5.3 Master Mode Register

FIGURE 5.6 - Using the Command Reg. to set the DPR.



The 16 bit MASTER MODE register is used to control the functions which effect all five of the counters.

Data Pointer Sequencing

Data Bus Width

Scaler Control (BCD or Binary)

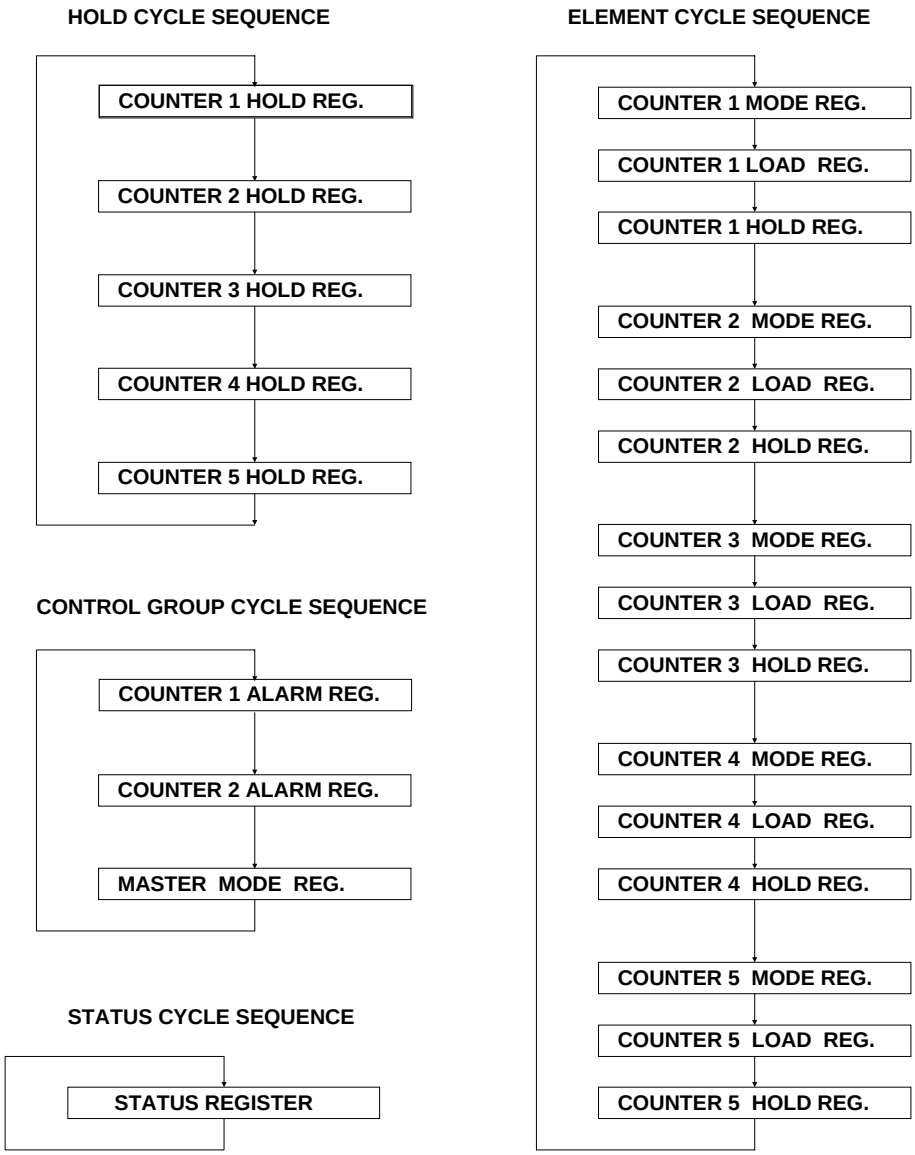
Enable Comparators for Counter 1 and Counter 2

Time-of-day Operation

Frequency control (not available on MPV991)

There are two methods of accessing the MASTER MODE register.

FIGURE 5.7 - Data Pointer Sequencing



1. Using the COMMAND register to access the MASTER MODE register

Certain Commands written to the COMMAND register are used to set or clear individual bits in the MASTER MODE register.

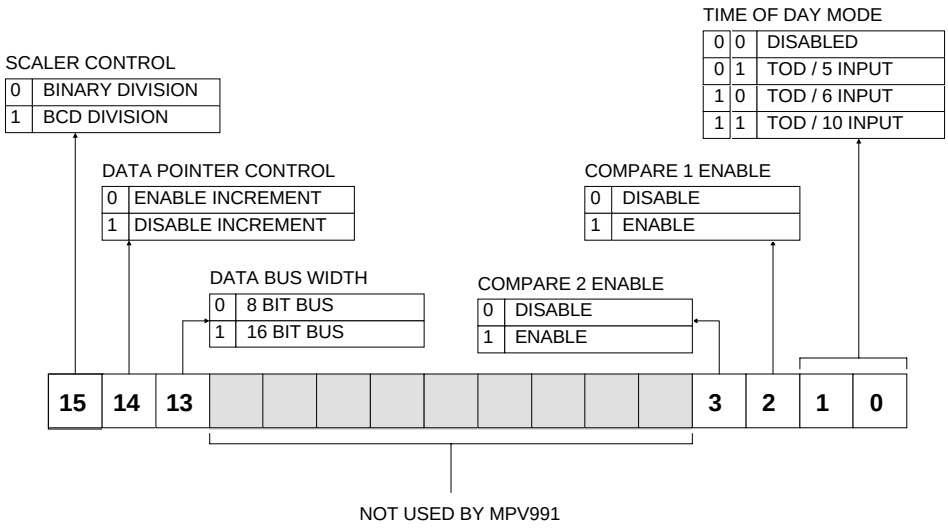
For convenience these are summarised again in Table 5.2.

2. Using the DATA POINTER register to access the MASTER MODE register

All bits of the MASTER MODE register may be accessed by loading its location into the DATA POINTER register. The required bit pattern is then written into the DATA port in two bytes.

After RESET the MASTER MODE register is left in the following condition :

FIGURE 5.8 - Master Mode Register



Time-of-Day disabled

Both Comparators disabled

Data Bus 8 Bits wide

Data Pointer Sequencing enabled

Frequency scaler divides in Binary

Note that the Data Bus **must** be used in 8 Bit mode.

TABLE 5.2 - Master Mode Commands

COMMAND CODE								COMMAND DESCRIPTION
1	1	1	0	1	1	1	0	GATE OFF FOUT (set MM12)
1	1	1	0	1	1	1	1	ENTER 16 BIT BUS MODE (set MM13) **N/A ON MPV991**
1	1	1	0	0	0	0	0	ENABLE DATA POINTER SEQUENCING (clear MM14)
1	1	1	0	0	1	1	0	GATE ON FOUT (clear MM12)
1	1	1	0	0	1	1	1	ENTER 8 BIT BUS MODE (clear MM13)

5.5.4 Counter Mode Register

The 16 bit COUNTER MODE register is used to control functions which effect individual counters only.

Output Control

Count Control

Count Source Selection

Gating Control

Source Edge Control

The COUNTER MODE register is accessed by loading its location into the DATA POINTER register and writing the desired bit pattern into the DATA port in two bytes.

After RESET the COUNTER MODE register is left in the following condition :

Output low impedance to ground
Count DOWN
Count in Binary
Count ONCE
Reload from LOAD register
No retrigger
F1 selected as Input
Positive input polarity
Gating Disabled

5.5.5 Load register

The 16 bit LOAD register is used to store the value loaded into the counter when it is initialised. (see Chapter 3).

The LOAD register is accessed by loading its location into the DATA POINTER register and writing the desired value into the DATA port in two bytes.

5.5.6 Hold Register

The 16 bit HOLD register has two functions.

It may be used as an alternative load source, offering the ability to produce complex duty waveforms. It is also used to store the accumulated counter contents without disturbing the counting process. (see Chapter 3)

There are two methods of accessing the HOLD register dependent on the way it is being used

1. Used as an alternative load source.

By loading its location into the DATA POINTER register and writing the desired value into the DATA port in two bytes.

2. Used to store the accumulated count.

By a command written into the COMMAND register.

5.5.7 Status Register

The 8 Bit STATUS register reflects the occurrence of a TC pulse (see Chapter 3) from each counter, and the condition of the BYTE POINTER used for data WRITES.

It is accessed by READING the CONTROL port. (see Section 5.4)

FIGURE 5.9 - Counter Mode Register

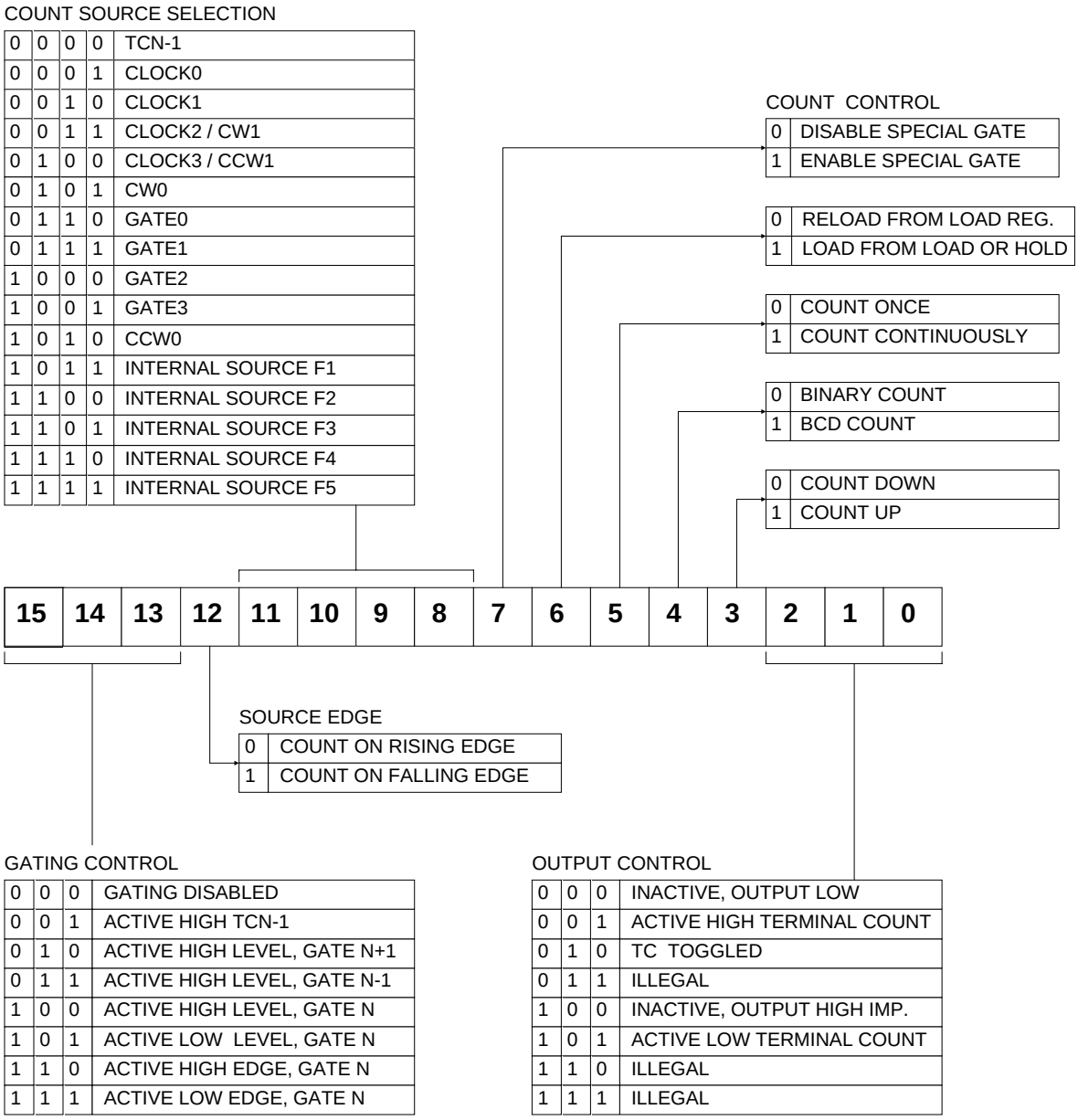
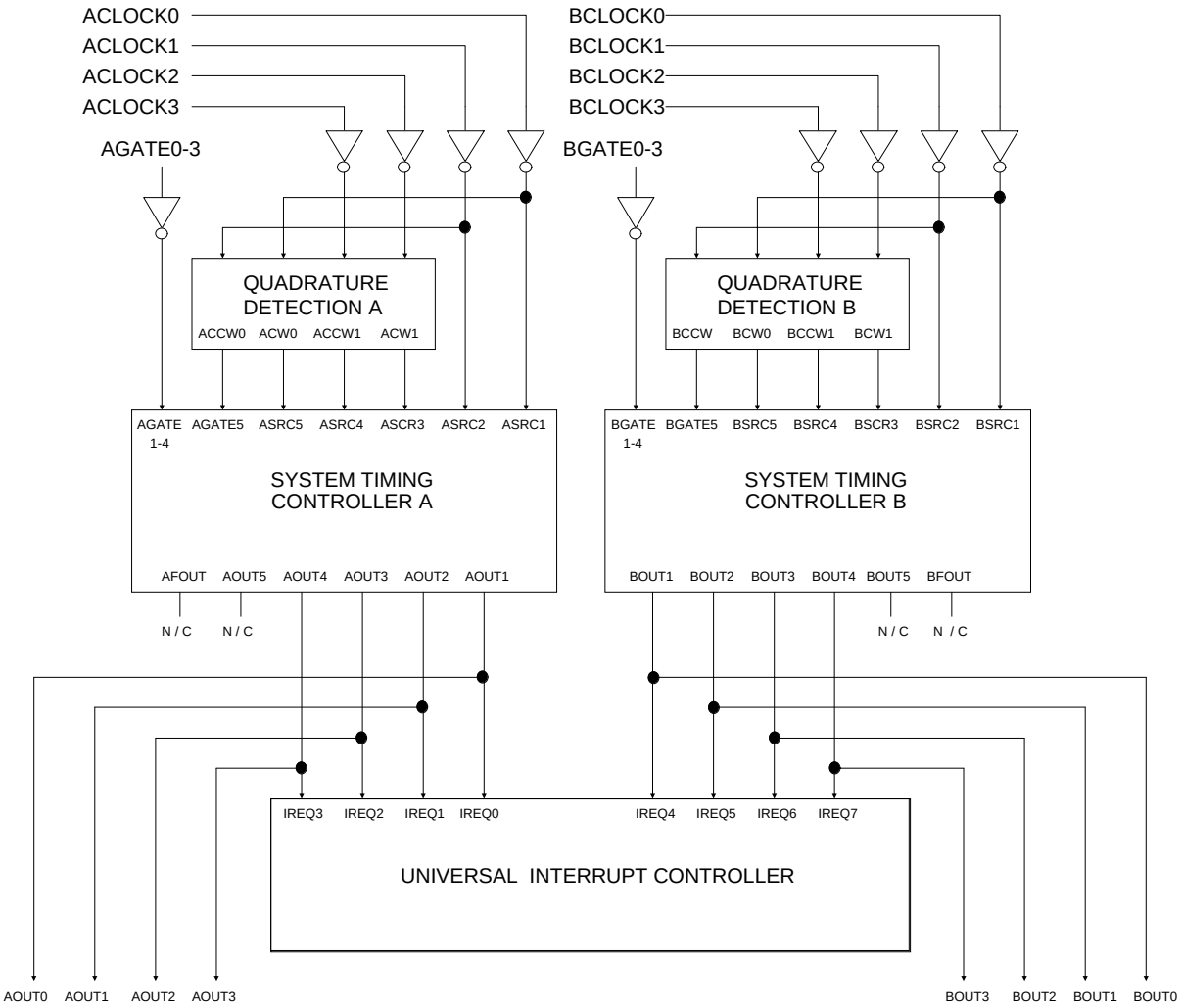


FIGURE 5.10 - Input and Output Connections



When Quadrature Detection is enabled the input and output connections are as above

When Quadrature Detection is disabled then : CLOCK2 --- SRC3 and CLOCK3 --- SRC4

Quadrature Detection cannot be disabled to SRC5 and GATE5

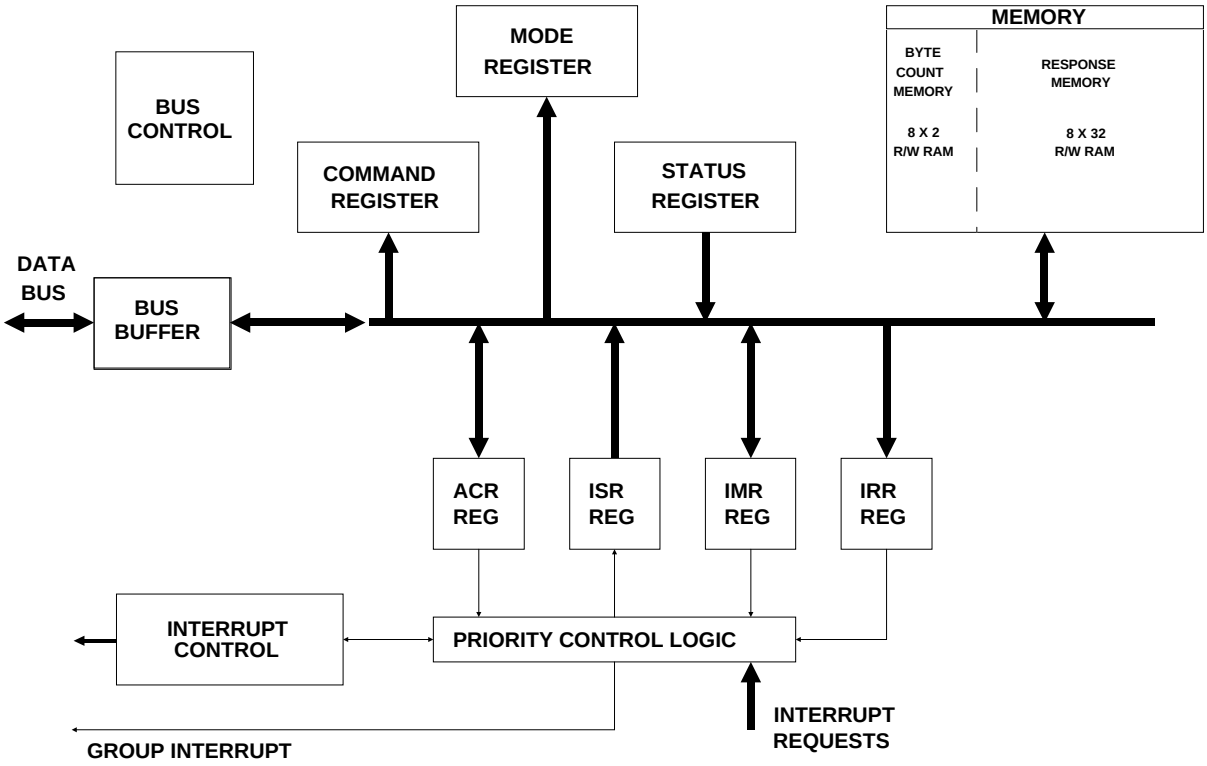
5.6 PROGRAMMING THE UNIVERSAL INTERRUPT CONTROLLER

The UIC is addressed by the user as only two locations: a CONTROL port and a DATA port. (see Memory Map in Figure 5.1.)

These ports are used to access and control the operation of registers associated with the UIC.

Detailed information concerning the operation of these registers can be found in the Am9519A Technical Manual supplied.

FIGURE 5.11 - UIC Register Access



Each WRITE access to the CONTROL port transfers the data written to the COMMAND register.

Each READ access to the CONTROL port transfers the data read from the STATUS register.

TABLE 5.3 - UIC Command Register Command Set

COMMAND CODE								
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**COMMAND DESCRIPTION**
0	0	0	0	0	0	0	0	RESET (clear IRR,ISR,ACR,set IMR)
0	0	0	1	0	X	X	X	CLEAR ALL IRR AND IMR BITS
0	0	0	1	1	B2	B1	B0	CLEAR IRR AND IMR BITS SPECIFIED BY B2,B1,B0
0	0	1	0	0	X	X	X	CLEAR ALL IMR BITS (ENABLE)
0	0	1	0	1	B2	B1	B0	CLEAR IMR BITS SPECIFIED BY B2,B1,B0
0	0	1	1	0	X	X	X	SET ALL IMR BITS (DISABLE)
0	0	1	1	1	B2	B1	B0	SET IMR BITS SPECIFIED BY B2,B1,B0
0	1	0	0	0	X	X	X	CLEAR ALL IRR BITS
0	1	0	0	1	B2	B1	B0	CLEAR IRR BITS SPECIFIED BY B2,B1,B0
0	1	0	1	0	X	X	X	SET ALL IRR BITS
0	1	0	1	1	B2	B1	B0	SET IRR BITS SPECIFIED BY B2,B1,B0
0	1	1	0	X	X	X	X	CLEAR HIGHEST PRIORITY ISR BIT
0	1	1	1	0	X	X	X	CLEAR ALL ISR BITS
0	1	1	1	1	B2	B1	B0	CLEAR ISR BIT SPECIFIED BY B2,B1,B0
1	0	0	M4	M3	M2	M1	M0	LOAD MODE REGISTER BITS 0-4 WITH PATTERN M0-M4
1	0	1	0	M6	M5	0	0	LOAD MODE REGISTER BITS 5 & 6 WITH PATTERN
1	0	1	0	M6	M5	0	1	LOAD MODE REG. BITS 5 & 6 AND SET BIT 7
1	0	1	0	M6	M5	1	0	LOAD MODE REG. BITS 5 & 6 AND CLEAR BIT 7
1	0	1	1	X	X	X	X	PRESELECT IMR FOR LOADING
1	1	0	0	X	X	X	X	PRESELECT ACR FOR LOADING
1	1	1	BY1	BY0	L2	L1	L0	LOAD BY1,BY0 INTO BYTE REGISTER AND SELECT LEVEL

B2	B1	B0	BIT No.
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

L2	L1	L0	LEVEL No.
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

BY1	BY0	COUNT
0	0	0
0	1	1
1	0	2
1	1	3

.6.1 Command Register

The 8 bit COMMAND register provides direct control over all the other associated registers.

.6.2 Mode Register

The 8 bit MODE register controls the operating mode of the UIC.

The MODE register is accessed by direct commands loaded into the COMMAND

TABLE 5.4 - UIC Mode Register Command Summary

COMMAND CODE								COMMAND DESCRIPTION
7	6	5	4	3	2	1	0	
1	0	0	M4	M3	M2	M1	M0	LOAD MODE REGISTER BITS 0-4 WITH PATTERN M0-M4
1	0	1	0	M6	M5	0	0	LOAD MODE REGISTER BITS 5 & 6 WITH PATTERN
1	0	1	0	M6	M5	0	1	LOAD MODE REG. BITS 5 & 6 AND SET BIT 7
1	0	1	0	M6	M5	1	0	LOAD MODE REG. BITS 5 & 6 AND CLEAR BIT 7

After RESET the MODE register is left in the following condition:

Fixed priority
Individual Vectoring
Interrupt operation
GINT active low
IREQ active low
ISR preselected for reading
UIC disarmed

5.6.3 Interrupt Request Register (IRR)

A bit in the 8 bit IRR is set whenever the corresponding IREQ input becomes active (see Section 3.3.1).

Individual bits may be cleared using commands written into the COMMAND register and these commands are summarised in Table 5.5 for convenience.

The contents of the IRR may be read at the DATA port by setting Bit5 and Bit6 of the MODE register to 10.

TABLE 5.5 - Interrupt Request Register Command Summary

COMMAND CODE								COMMAND DESCRIPTION
7	6	5	4	3	2	1	0	
0	1	0	0	0	X	X	X	CLEAR ALL IRR BITS
0	1	0	0	1	B2	B1	B0	CLEAR IRR BITS SPECIFIED BY B2,B1,B0
0	1	0	1	0	X	X	X	SET ALL IRR BITS
0	1	0	1	1	B2	B1	B0	SET IRR BITS SPECIFIED BY B2,B1,B0

5.6.4 Interrupt Service Register (ISR)

The 8 bit ISR is used to store the acknowledge status of individual interrupts. (see Section 3.3.2)

When the ISR is not programmed for automatic clearing it must be cleared by a command from the host CPU before the end of the interrupt.

The contents of the ISR may be read at the DATA port by setting bit5 and bit6 of the MODE register to 00.

TABLE 5.6 - Interrupt Service Register Command Summary

COMMAND CODE								
7	6	5	4	3	2	1	0	COMMAND DESCRIPTION
0	1	1	0	X	X	X	X	CLEAR HIGHEST PRIORITY ISR BIT
0	1	1	1	0	X	X	X	CLEAR ALL ISR BITS
0	1	1	1	1	B2	B1	B0	CLEAR ISR BIT SPECIFIED BY B2,B1,B0

5.6.5 Interrupt Mask Register (IMR)

The 8 bit IMR is used to disable/enable individual interrupts. Only unmasked IRR bits are serviced. (see Section 3.3.3)

There are two methods of altering bits in the IMR :

1. Direct commands from the COMMAND register

The commands written into the COMMAND register which affect the IMR are summarised in Table 5.7.

TABLE 5.7 - Interrupt Mask Register Command Summary

COMMAND CODE								
7	6	5	4	3	2	1	0	COMMAND DESCRIPTION
0	0	1	0	0	X	X	X	CLEAR ALL IMR BITS (ENABLE)
0	0	1	0	1	B2	B1	B0	CLEAR IMR BITS SPECIFIED BY B2,B1,B0
0	0	1	1	0	X	X	X	SET ALL IMR BITS (DISABLE)
0	0	1	1	1	B2	B1	B0	SET IMR BITS SPECIFIED BY B2,B1,B0

2. Pre-select IMR for DATA using MODE register

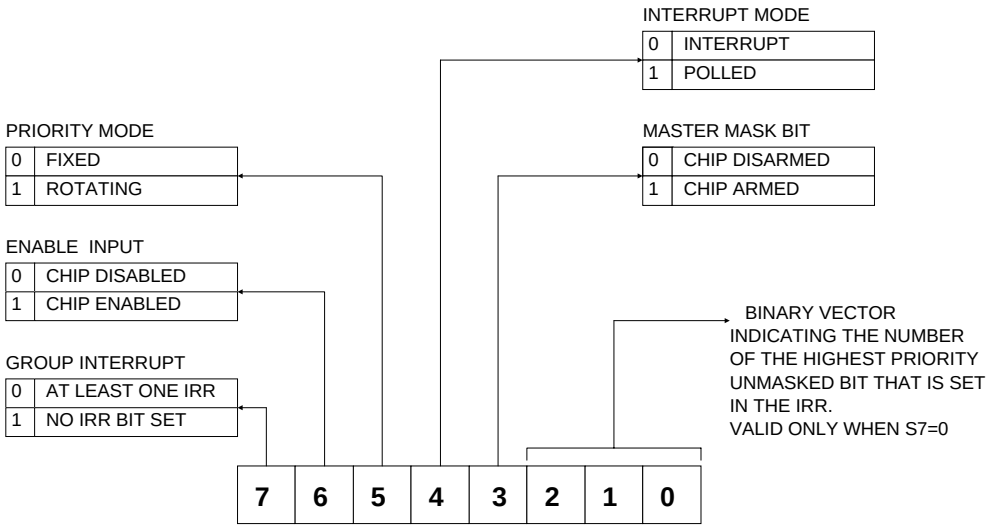
Bit5 and Bit6 of the MODE register are set to 01 to preselect the IMR. The desired bit pattern may then be entered into the DATA port.

5.6.6 Status Register

The 8 bit STATUS register reflects the internal state of the UIC and is read directly by a READ operation at the COMMAND port.

The STATUS register is illustrated in Figure 5.12.

FIGURE 5.12 - Status Register



5.6.7 Programming the Interrupt Response

The action taken by the host on the reception of a given interrupt is determined by the contents of the associated RESPONSE MEMORY location. (see Chapter 3)

The RESPONSE MEMORY is accessed via the COMMAND register. Bit5-Bit7 of the COMMAND register are set to 111 to point to RESPONSE MEMORY. The remaining bits select the desired Response Level and one of the four RESPONSE BYTES associated with it. Response Data (eg. an interrupt vector) is then written to that location via the DATA port.

FIGURE 5.13 - Programming the Interrupt Response

COMMAND CODE								
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**COMMAND DESCRIPTION**
1	1	1	B1	B0	L2	L1	L0	LOAD B1,B0 INTO BYTE REGISTER AND SELECT LEVEL

B1	B0	COUNT
0	0	1
0	1	2
1	0	3
1	1	4

L2	L1	L0	LEVEL
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

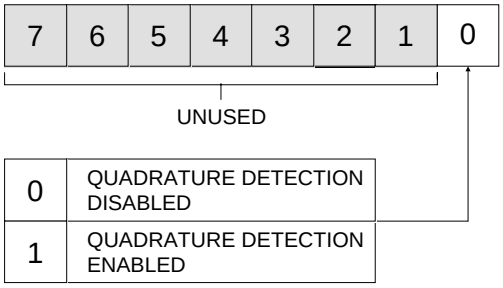
5.7 QUADRATURE DETECTION CIRCUITS

The Quadrature Detection Circuits associated with each STC accept the SIN and COS outputs from a transducer and decode these signals to provide a train of "clockwise" pulses for movement in one direction and a train of "counter-clockwise" pulses for movement in the other.

Quadrature Detection is selected by setting Bit0 of the Quadrature Function Selection Register. (Bit1-Bit7 are unused).

When Quadrature Detection is selected the CW and CCW pulses are available at the at the STC inputs as detailed in Figure 5.14.

FIGURE 5.14 - Quadrature Function Selection Register



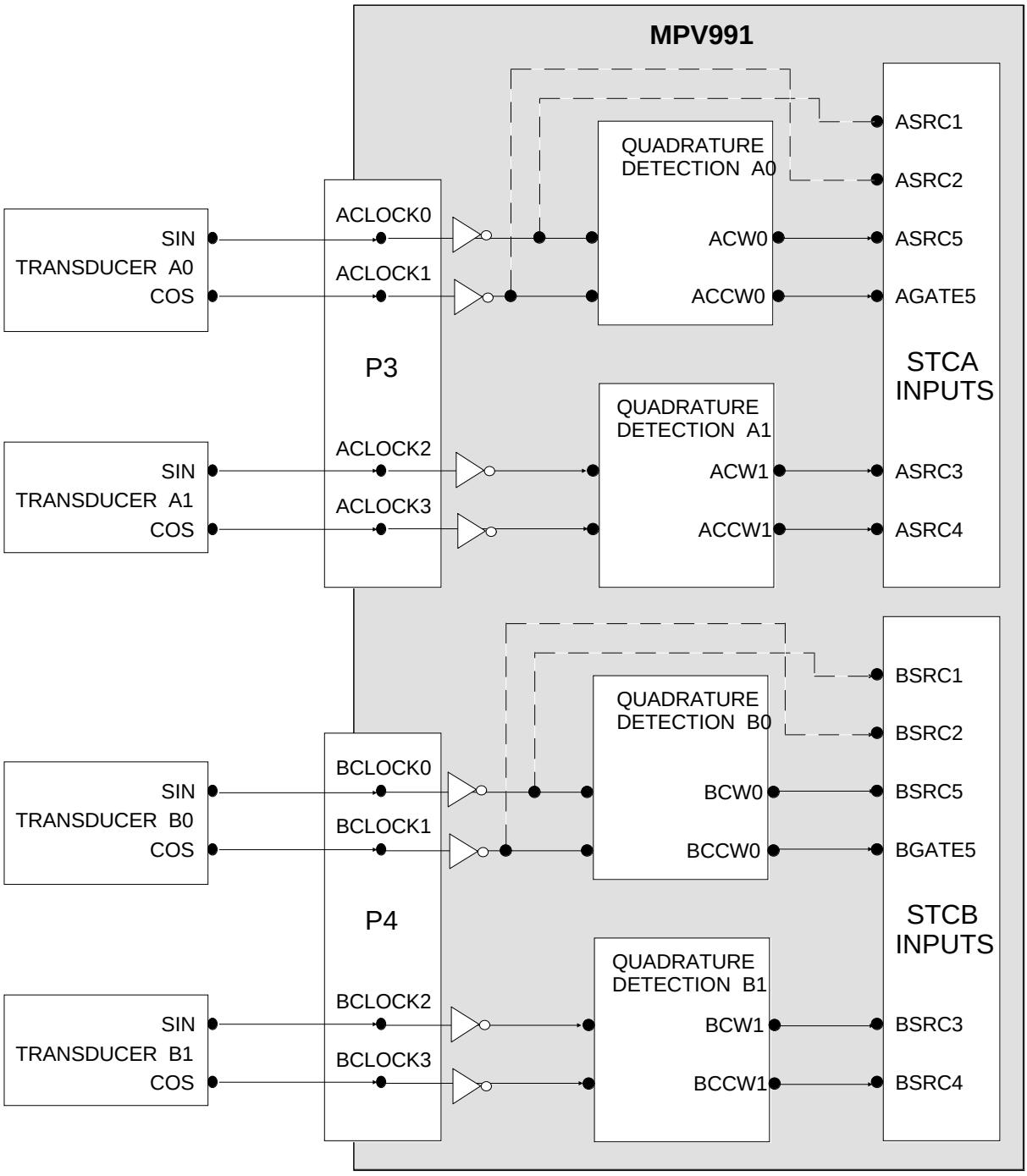
SRC5 and GATE5 on the STC are always connected to the Quadrature Detection Circuits whether or not Quadrature Detection is selected.

By counting the number of CW and CCW pulses associated with each transducer it is possible to calculate the position of the transducer. One method of doing this is discussed in the following section.

FIGURE 5.15 - Quadrature Detection, Inputs and Outputs

STC INPUT	QUADRATURE SELECTED	QUADRATURE NOT SELECTED
SRC1	CLOCK0	CLOCK0
SRC2	CLOCK1	CLOCK1
SRC3	CW1	CLOCK2
SRC4	CCW1	CLOCK3
SRC5	CW0	CW0
GATE1	GATE0	GATE0
GATE2	GATE1	GATE1
GATE3	GATE2	GATE2
GATE4	GATE3	GATE3
GATE5	CCW0	CCW0

FIGURE 5.16 - Quadrature Detection, Inputs and Outputs



.7.1 Position Measurement using Quadrature Detection

Position measurement using the MPV991 can be divided into four general operations:

1. Determine the direction of movement.

This operation is achieved by connecting the transducer to the MPV991 as described earlier, and enabling the Quadrature Detection circuits by setting Bit1 of the QFS Register to 1. The Quadrature Detection circuits then decode the transducer SIN and COS outputs to produce two pulse trains indicating movement in either the "clockwise" or "anti-clockwise" direction.

2. Record the number of pulses indicating movement in each direction.

The CW and CCW pulses are applied to two separate counters which have been configured as follows :

Count up

Count continuously

Reload from LOAD reg. (loaded with 0000H)

Gating enabled (either edge or level)

Special Gate enabled (Hardware retriggering selected)

3. Detect "Zero Position"

If an INDEX output, indicating when a "zero position" has been reached, is available from the transducer, then this should be used to zero and reset the counters.

This may be done by applying the INDEX pulse to the gate inputs of the two counters, provided they have been configured to support hardware retriggering. In this mode, on reception of the INDEX pulse the contents of both counters is transfered to their respective HOLD register. The next COUNT pulse (CW or CCW) received by each counter, after the INDEX pulse at the gate, causes the counter to be reloaded from its LOAD register and counting continues as futher pulses are received.

For applications where the number of INDEX pulses received is also of interest, the INDEX pulse may be applied to a third counter which is configured as a simple UP counter.

4. Calculate Actual Position.

Once the MPV991 has been configured as described and armed, it will record the movement in either direction continuously. The counters may be interrogated at any time without disturbing the counting process and the accumulated count value transfered to their associated HOLD registers. These count values may then be saved and used to calculate the Actual transducer position.

CHAPTER 6 - APPLICATIONS AND EXAMPLES

6.1 INTRODUCTION

Chapter 6 contains a brief summary of typical applications in which the MPV991 is used, and a number of simple Example Programs which demonstrate the operation of the board.

6.2 APPLICATIONS

Repetitive timing and counting activities are fundamentally involved in numerous applications, from simple count accumulation to complex control processes. These activities tie up a significant portion of CPU resources. The MPV991 releases these resources.

Applications in which the MPV991 is used can be divided into three general categories:

1. Count Accumulation

The MPV991 is used simply to record the counts applied to its inputs.

The accumulated count is available at any time without disturbing the counting process. The count is first saved in the HOLD register and then passed to a location in memory where it may be stored and operated upon.

APPLICATIONS :

- Event Counting
- Time-of-Day Generation
- Interval Measurement

2. Frequency Division and Generation

Output frequencies and waveforms are based on the range of input sources available. The output frequency is dependent on the LOAD value selected for each counter and using alternate load sources allows complex waveforms to be generated.

APPLICATIONS :

- Frequency Division
- Waveform Generation
- Baud Rate Generation

3. Measurement and Control

The Quadrature Detection Circuitry and the Universal Interrupt Controller provide the MPV991 with very powerful Measurement and Control capability.

APPLICATIONS :

- Position Measurement
- Gap Detection
- Periodic Interrupt Generation
- Machine Health Monitoring
- Motor Control
- Process Control
- Robotics

6.3 EXAMPLE PROGRAMS

6.3.1 EXAMPLE PROGRAM 1

This is a test program for the MPV991 COUNTER MODULE developed for a MOTOROLA MVME133XT, 25MHz CPU.

It demonstrates a very basic use of one of the counters on one of the two System Timing Controllers (STCs) available on the MPV991.

The program sets up COUNTER GROUP 1 on STCA to output a continuous square wave whose period is dependent on the source frequency and the value loaded into the associated LOAD register.

EXAMPLE : Source frequency F1 is selected, which is a 4MHz source. A square wave of frequency 1KHz is desired. A square wave can be achieved by setting the output to toggle on each TC and the time elapsed between each TC is controlled by the value loaded into the LOAD register. As two TCs must occur for each complete period, the frequency of the TC pulse train must be twice the desired frequency and the value written into the LOAD register is derived as follows ----

LOAD VALUE = (F1/f)/2 = (4MHz/1KHz)/2 = 2000 = 7D0H

For the purpose of this test program, the Data Pointer Sequencing is disabled. Therefore data pointer must be set to point to the desired destination register before each data write.

********** **PROGRAM CODE** **********

```
        ORG         $3000
---------- initialisation sequence ----------


        MOVE.B      #$FF,$FFFF0003    ;RESET STC A
        MOVE.B      #$E7,$FFFF0003    ;set to 8 bit mode
        MOVE.B      #$5F,$FFFF0003    ;"dummy" LOAD required as part of
                                      ;initialisation
---------- disable data pointer sequencing for this test program -------

        MOVE.B      #$E8,$FFFF0003    ;disable data pointer sequencing
```

(This could also be achieved by pointing directly to the MASTER MODE register and setting BIT 14 :

```
;       MOVE.B      #$17,$FFFF0003    ;point to MM register
;       MOVE.B      #$00,$FFFF0001    ;set LSB
;       MOVE.B      #$40,$FFFF0001    ;set MSB
```

```
---------- set up STCA, counter group 1----------------------------------------
        MOVE.B      #$01,$FFFF0003    ;point to STCA, CNTR 1, MODE register
        MOVE.B      #$22,$FFFF0001    ;set MODE (LSB)
        MOVE.B      #$0B,$FFFF0001    ;set MODE (MSB) :
                                      ;TOGGLE TC
                                      ;count down
                                      ;count binary
                                      ;count CONTINUOUS
                                      ;reload from LOAD REG
                                      ;disable special gate
                                      ;F1 input source
                                      ;count on rising edge
                                      ;no gating
---------- control period of waveform with LOAD reg ----------------------
        MOVE.B      #$09,$FFFF0003    ;point to LOAD register
        MOVE.B      #$D0,$FFFF0001    ;load LSB into LOAD reg
        MOVE.B      #$07,$FFFF0001    ;load MSB into LOAD reg
        MOVE.B      #$41,$FFFF0003    ;load contents of LOAD reg. into CNTR 1
---------- set up 991 CONTROL reg and arm timer ---------------------------
        MOVE.B      #$08,$FFFF0081    ;enable outputs on MPV991
        MOVE.B      #$61,$FFFF0003    ;arm counter 1
---------- end program, return to monitor----------------------------------------
        TRAP        #$0E
```

****************************END OF PROGRAM CODE********************

6.3.2 EXAMPLE PROGRAM 2

This is a test program for the MPV991 COUNTER MODULE developed for a MOTOROLA MVME133XT, 25MHz CPU.

It demonstrates a very basic use of one of the counters on one of the two System Timing Controllers (STCs) available on the MPV991.

The program sets up COUNTER GROUP 1 on STCA to output a continuous square wave whose period and MARK:SPACE ratio is dependent on the source frequency and the values loaded into the associated LOAD and HOLD registers.

EXAMPLE : Source F2 (4MHz/16 = 250KHz) is selected in order to produce a square wave of frequency 15KHz and mark:space ratio of 1:2. The counter is set to load from the LOAD and HOLD registers on alternate TC pulses. This means that the duration of the MARK and the SPACE can be controlled independently.

MARK pulse = f1 = 10KHz and SPACE pulse = f2 = 5KHz

LOAD value = F2/f1 = 250KHz/10KHz = 12.5 = 0CH

HOLD value = F2/f2 = 250KHz/5KHz = 25 = 19H

Data pointer sequencing is disabled for this test program

****************************PROGRAM CODE****************************

```
        ORG         $3000
----------------initialisation sequence ----------------------------------------------------
        MOVE.B      #$FF,$FFFF0003    ;RESET STC A
        MOVE.B      #$E7,$FFFF0003    ;set to 8 bit mode
        MOVE.B      #$5F,$FFFF0003    ;"dummy" LOAD required as part of
                                      ;initialisation
----------------disable data pointer sequencing for this test program--------
        MOVE.B      #$E8,$FFFF0003    ;disable data pointer sequencing
(This could also be achieved by pointing directly to the MASTER MODE register
and setting BIT 14 :
;       MOVE.B      #$17,$FFFF0003    ;point to MM register
;       MOVE.B      #$00,$FFFF0001    ;set LSB
;       MOVE.B      #$40,$FFFF0001    ;set MSB
----------------set up STCA, counter group 1 ------------------------------------------
        MOVE.B      #$01,$FFFF0003    ;point to STCA, CNTR 1, MODE register
        MOVE.B      #$62,$FFFF0001    ;set MODE (LSB)
        MOVE.B      #$0C,$FFFF0001    ;set MODE (MSB) :
                                      ;TOGGLE TC
```

```
                                        ;count down
                                        ;count binary
                                        ;count CONTINUOUS
                                        ;reload from LOAD and HOLD regs
                                        ;disable special gate
                                        ;F2 input source
                                        ;count on rising edge
                                        ;no gating
---------- control period of waveform with LOAD reg.----------
        MOVE.B      #$09,$FFFF0003      ;point to LOAD register
        MOVE.B      #$0C,$FFFF0001      ;load LSB into LOAD reg
        MOVE.B      #$00,$FFFF0001      ;load MSB into LOAD reg
        MOVE.B      #$11,$FFFF0003      ;point to HOLD register
        MOVE.B      #$19,$FFFF0001      ;load LSB into HOLD reg
        MOVE.B      #$00,$FFFF0001      ;load MSB into HOLD reg
        MOVE.B      #$41,$FFFF0003      ;load contents of LOAD reg. into COUNTER 1
---------- set up 991 CONTROL reg and arm timer----------
        MOVE.B      #$08,$FFFF0081      ;enable outputs on MPV991
        MOVE.B      #$61,$FFFF0003      ;arm counter 1
---------- end program, return to monitor ----------
        TRAP        #$0E

********** END OF PROGRAM CODE **********
```

6.3.3 EXAMPLE PROGRAM 3

This is a test program for the MPV991 COUNTER MODULE developed for a MOTOROLA MVME133XT, 25MHz CPU.

The program sets up four counters on STCB to output continuous square waves of various frequencies and M/S ratios based on the sources F1 to F4.

Data Pointer Sequencing is disabled for this program, however the following program is identical except for the fact that data pointer sequencing is enabled. The commands which are made redundant by the use of data pointer sequencing are highlighted here.

****************************PROGRAM CODE******************************

```
        ORG         $3000
---------- initialisation sequence ----------------------------------------------------
        MOVE.B      #$FF,$FFFF0007    ;RESET STCB
        MOVE.B      #$E7,$FFFF0007    ;set to 8 bit mode
        MOVE.B      #$5F,$FFFF0007    ;"dummy" LOAD required as part of initialisatior
---------- disable autoincrement for this test program ----------------------
        MOVE.B      #$E8,$FFFF0007    ;disable data pointer sequencing
---------- set up STCB, counter group 1----------------------------------------
;Output square wave of frequency 4KHz, M/S 1:1
;Select input source F1 = 4MHz
;mark:space = 1:1 therefore reload from load register only.
;LOAD value = (4MHz/4KHz)/2 = 1F4H
        MOVE.B      #$01,$FFFF0007    ;point to STCB, COUNTER 1, MODE register
        MOVE.B      #$22,$FFFF0005    ;set MODE (LSB)
        MOVE.B      #$0B,$FFFF0005    ;set MODE (MSB) : TOGGLE TC
                                      ;count down
                                      ;count binary
                                      ;count CONTINUOUS
                                      ;reload from LOAD register
                                      ;disable special gate
                                      ;F1 input source
                                      ;count on rising edge
                                      ;no gating
---------- control period of waveform with LOAD 1reg----------------------
        MOVE.B      #$09,$FFFF0007    ;point to STCB, CNTR 1, LOAD register
        MOVE.B      #$F4,$FFFF0005    ;load LSB into LOAD reg 1
        MOVE.B      #$01,$FFFF0005    ;load MSB into LOAD reg 1
---------- set up STCB, counter group 2----------------------------------------
output square wave of frequency 4kHz, M/S 1:3
select iput source F1 = 4MHz
;reload from both LOAD and Hold registers
```

```
;MARK pulse = f1 = 1KHz, LOAD value = 4MHz/1KHz = 4000 = FA0H
;SPACE pulse = f2 = 3KHz, HOLD value = 4MHz/3KHz = 1333.3 = 535H
        MOVE.B      #$02,$FFFF0007    ;point to STCB, CNTR 2, MODE register
        MOVE.B      #$62,$FFFF0005    ;set MODE (LSB)
        MOVE.B      #$0B,$FFFF0005    ;set MODE (MSB) : TOGGLE TC
                                      ;count down
                                      ;count binary
                                      ;count CONTINUOUS
                                      ;reload from LOAD and HOLD regs
                                      ;disable special gate
                                      ;F1 input source
                                      ;count on rising edge
                                      ;no gating
----------control period of waveform with LOAD 2 reg ----------

        MOVE.B      #$0A,$FFFF0007    ;point to STCB, CNTR 2, LOAD register
        MOVE.B      #$A0,$FFFF0005    ;load LSB into LOAD reg 2
        MOVE.B      #$0F,$FFFF0005    ;load MSB into LOAD reg 2
        MOVE.B      #$12,$FFFF0007    ;point to STCB, CNTR 2, HOLD register
        MOVE.B      #$35,$FFFF0005    ;load LSB into HOLD reg 2
        MOVE.B      #$05,$FFFF0005    ;load MSB into HOLD reg 2
----------set up STCB, counter group 3 ----------------------------------------

Output 25KHz square wave, mark:space = 1:4
select input source F2 = F1/16 = 250KHz
reload from LOAD and HOLD registers
MARK pulse = f1 = 5KHz, HOLD value = 250KHz/5KHz = 50 = 32H
;SPACE pulse = f2 = 20KHz, LOAD value = 250KHz/20KHz = 12.5 =0CH
        MOVE.B      #$03,$FFFF0007    ;point to STCB, CNTR 3, MODE register
        MOVE.B      #$62,$FFFF0005    ;set MODE (LSB)
        MOVE.B      #$0C,$FFFF0005    ;set MODE (MSB) :
                                      ;TOGGLE TC
                                      ;count down
                                      ;count binary
                                      ;count CONTINUOUS
                                      ;reload from LOAD and HOLD regs
                                      ;disable special gate
                                      ;F2 input source
                                      ;count on rising edge
                                      ;no gating
----------control period of waveform with LOAD 3 reg ----------

        MOVE.B      #$0B,$FFFF0007    ;point to STCB, CNTR 3, LOAD register
        MOVE.B      #$32,$FFFF0005    ;load LSB into LOAD reg 3
        MOVE.B      #$00,$FFFF0005    ;load MSB into LOAD reg 3
        MOVE.B      #$13,$FFFF0007    ;point to STCB, CNTR 3, HOLD register
        MOVE.B      #$0C,$FFFF0005    ;load LSB into HOLD reg 3
```

```
        MOVE.B      #$00,$FFFF0005    ;load MSB into HOLD reg 3
----------set up STCB, counter group 4----------

;Output F4 frequency square wave, mark:space = 1:1
;select input source F4

;reload from LOAD only
;LOAD value = 2H
        MOVE.B      #$04,$FFFF0007    ;point to STCB, CNTR 4, MODE register
        MOVE.B      #$22,$FFFF0005    ;set MODE (LSB)
        MOVE.B      #$0E,$FFFF0005    ;set MODE (MSB) :
                                      ;TOGGLE TC
                                      ;count down
                                      ;count binary
                                      ;count CONTINUOUS
                                      ;reload from LOAD and HOLD regs
                                      ;disable special gate
                                      ;F4 input source
                                      ;count on rising edge
                                      ;no gating
----------control period of waveform with LOAD 4 reg----------

        MOVE.B      #$0C,$FFFF0007    ;point to STCB, COUNTER 4, LOAD register
        MOVE.B      #$02,$FFFF0005    ;load LSB into LOAD reg 4
        MOVE.B      #$00,$FFFF0005    ;load MSB into LOAD reg 4
----------load contents of LOAD registers----------

        MOVE.B      #$4F,$FFFF0007    ;load contents of LOAD 1,2,3,4 registers
----------set up 991 CONTROL reg and arm timer----------

        MOVE.B      #$08,$FFFF0081    ;enable outputs on MPV991
        MOVE.B      #$2F,$FFFF0007    ;arm counters
----------end program, return to monitor----------

        TRAP        #$0E

********** END OF PROGRAM CODE **********
```

6.3.4 EXAMPLE PROGRAM 3A

This is a test program for the MPV991 COUNTER MODULE developed for a MOTOROLA MVME133XT, 25MHz CPU.

This program is identical in function to that in EXAMPLE PROGRAM 3, however it illustrates the use of the Data Pointer Sequencer to reduce the number of instructions required.

****************************PROGRAM CODE****************************

```
        ORG         $3000
---------------- initialisation sequence ----------------------------------------------------
        MOVE.B      #$FF,$FFFF0007    ;RESET STCB
        MOVE.B      #$E7,$FFFF0007    ;set to 8 bit mode
        MOVE.B      #$5F,$FFFF0007    ;"dummy" LOAD required as part of
                                      ;initialisation
---------------- set up STCB, counter group 1-----------------------------------------
Output square wave of frequency 4KHz, M/S 1:1
Select input source F1 = 4MHz
mark:space = 1:1 therefore reload from load register only.
LOAD value = (4MHz/4KHz)/2 = 1F4H
        MOVE.B      #$01,$FFFF0007    ;point to STCB, CNTR 1, MODE register
        MOVE.B      #$22,$FFFF0005    ;set MODE (LSB)
        MOVE.B      #$0B,$FFFF0005    ;set MODE (MSB) : TOGGLE TC
                                      ;count down
                                      ;count binary
                                      ;count CONTINUOUS
                                      ;reload from LOAD register
                                      ;disable special gate
                                      ;F1 input source
                                      ;count on rising edge
                                      ;no gating
control period of waveform with value loaded into LOAD 1reg
        MOVE.B      #$F4,$FFFF0005    ;load LSB into LOAD reg 1
        MOVE.B      #$01,$FFFF0005    ;load MSB into LOAD reg 1

---------------- set up STCB, counter group 2 ----------------------------------------
output square wave of frequency 4kHz, M/S 1:3
;select iput source F1 = 4MHz
;reload from both LOAD and Hold registers
;MARK pulse = f1 = 1KHz, LOAD value = 4MHz/1KHz = 4000 = FA0H
;SPACE pulse = f2 = 3KHz, HOLD value = 4MHz/3KHz = 1333.3 = 535H
        MOVE.B      #$02,$FFFF0007    ;point to STCB, CNTR 2, MODE register
        MOVE.B      #$62,$FFFF0005    ;set MODE (LSB)
        MOVE.B      #$0B,$FFFF0005    ;set MODE (MSB) : TOGGLE TC
```

```
                                        ;count down
                                        ;count binary
                                        ;count CONTINUOUS
                                        ;reload from LOAD and HOLD regs
                                        ;disable special gate
                                        ;F1 input source
                                        ;count on rising edge
                                        ;no gating
---------- control period of waveform with LOAD 2 reg ----------
        MOVE.B      #$A0,$FFFF0005      ;load LSB into LOAD reg 2
        MOVE.B      #$0F,$FFFF0005      ;load MSB into LOAD reg 2
        MOVE.B      #$35,$FFFF0005      ;load LSB into HOLD reg 2
        MOVE.B#$05,$FFFF0005            ;load MSB into HOLD reg 2
---------- set up STCB, counter group 3  ----------

;Output 25KHz square wave, mark:space = 1:4
;select input source F2 = F1/16 = 250KHz
;reload from LOAD and HOLD registers
;MARK pulse = f1 = 5KHz, HOLD value = 250KHz/5KHz = 50 = 32H
;SPACE pulse = f2 = 20KHz, LOAD value = 250KHz/20KHz = 12.5 =0CH

        MOVE.B      #$62,$FFFF0005      ;set MODE (LSB)
        MOVE.B      #$0C,$FFFF0005      ;set MODE (MSB) : TOGGLE TC
                                        ;count down
                                        ;count binary
                                        ;count CONTINUOUS
                                        ;reload from LOAD and HOLD regs
                                        ;disable special gate
                                        ;F2 input source
                                        ;count on rising edge
                                        ;no gating
---------- control period of waveform with LOAD 3 reg ----------
        MOVE.B      #$32,$FFFF0005      ;load LSB into LOAD reg 3
        MOVE.B      #$00,$FFFF0005      ;load MSB into LOAD reg 3
        MOVE.B      #$0C,$FFFF0005      ;load LSB into HOLD reg 3
        MOVE.B      #$00,$FFFF0005      ;load MSB into HOLD reg 3
---------- set up STCB, counter group 4  ----------

Output F4 frequency square wave, mark:space = 1:1
;select input source F4
;reload from LOAD only
;LOAD value = 2H
        MOVE.B      #$04,$FFFF0007      ;point to STCB, CNTR , MODE register
        MOVE.B      #$22,$FFFF0005      ;set MODE (LSB)
        MOVE.B      #$0E,$FFFF0005      ;set MODE (MSB) : TOGGLE TC
```

```
                                        ;count down
                                        ;count binary
                                        ;count CONTINUOUS
                                        ;reload from LOAD and HOLD regs
                                        ;disable special gate
                                         ;F4 input source
                                        ;count on rising edge
                                        ;no gating
----------control period of waveform with LOAD 4 reg ----------
        MOVE.B      #$02,$FFFF0005    ;load LSB into LOAD reg 4
        MOVE.B      #$00,$FFFF0005    ;load MSB into LOAD reg 4
----------load contents of LOAD registers----------
        MOVE.B      #$4F,$FFFF0007    ;load contents of LOAD 1,2,3,4 registers
----------set up 991 CONTROL reg and arm timer ----------
        MOVE.B      #$08,$FFFF0081    ;enable outputs
        MOVE.B      #$2F,$FFFF0007    ;arm counters
----------end program, return to monitor  ----------
        TRAP        #$0E

********** END OF PROGRAM CODE **********
```

;This is a test program for the MPV991 COUNTER MODULE developed for a MOTOROLA MVME133XT, 25MHz CPU.

;The program sets up COUNTER GROUP 1 on STCA to output a continuous square wave of frequency 10KHz and COUNTER GROUP 1 on STCB to output a continuous square wave of low frequency. The output from STCB is then used to GATE the output from STCA. The result is a train of 10KHz pulses repeating at 100Hz intervals.

PROGRAM CODE

```
        ORG         $3000
---------- initialisation sequence ----------------------------------------------------

        MOVE.B      #$FF,$FFFF0003  ;RESET STCA
        MOVE.B      #$E7,$FFFF0003  ;set to 8 bit mode
        MOVE.B      #$5F,$FFFF0003  ;"dummy" LOAD required as part of
                                    ;initialisation
        MOVE.B      #$FF,$FFFF0007  ;RESET STCB
        MOVE.B      #$E7,$FFFF0007  ;set to 8 bit mode
        MOVE.B      #$5F,$FFFF0007  ;"dummy" LOAD required as part of
                                    ;initialisation
---------- set up STCB, counter group 1 ----------------------------------------------

        MOVE.B      #$01,$FFFF0007  ;point to STCB, CNTR 1, MODE register
        MOVE.B      #$22,$FFFF0005  ;set MODE (LSB)
        MOVE.B      #$0D,$FFFF0005  ;set MODE (MSB) :
                                    ;TOGGLE TC
                                    ;count down
                                    ;count binary
                                    ;count CONTINUOUS
                                    ;reload from LOAD reg
                                    ;disable special gate
                                    ;F4 input source
                                    ;count on rising edge
                                    ;no gating

---------- control period of waveform with LOAD 1reg ----------------------

        MOVE.B      #$50,$FFFF0005  ;load LSB into LOAD reg 1
        MOVE.B      #$C3,$FFFF0005  ;load MSB into LOAD reg 1
```

set up STCA, counter group 1

Output continuous square wave of 10KHz, LOAD value = 200 = C8H
Active high gating

```
        MOVE.B      #$01,$FFFF0003  ;point to STCA, COUNTER 1, MODE register
        MOVE.B      #$22,$FFFF0001  ;set MODE (LSB)
        MOVE.B      #$8B,$FFFF0001  ;set MODE (MSB) : TOGGLE TC
                                    ;count down
```

```
                                        ;count binary
                                        ;count CONTINUOUS
                                        ;reload from LOAD
                                        ;disable special gate
                                        ;F1 input source
                                        ;count on rising edge
                                        ;active high GATE
----------control period of waveform with LOAD 1 reg----------

        MOVE.B      #$C8,$FFFF0005      ;load LSB into LOAD reg 1
        MOVE.B      #$00,$FFFF0005      ;load MSB into LOAD reg 1
----------load contents of LOAD registers ----------------------------------

        MOVE.B      #$41,$FFFF0003      ;load contents of STCA LOAD 1 reg
        MOVE.B      #$41,$FFFF0007      ;load contents of STCB LOAD 1 reg
----------set up 991 CONTROL reg and arm timer----------------------------

        MOVE.B      #$0F,$FFFF0081      ;select IRQ7 and enable outputs
        MOVE.B      #$21,$FFFF0003      ;arm STCA (counters start)
        MOVE.B      #$21,$FFFF0007      ;arm STCB (counters start)
----------end program, return to monitor ----------------------------------

        TRAP        #$0E

**************************END OF PROGRAM CODE *****************
```