MPV955 Analog Output Board

PENTLAND SYSTEMS LIMITED

MPV 955

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

Telephone: **(01506) 464666**
Fax No: **(01506) 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

MPV955 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: P61491

ECO No.	11962	11986	12007	12049	12115	12274	12554/12560
ISSUE	SEPT 94	OCT 94	JAN 95	FEB 95	MAY 95	APR 96	NOV 97
REVISION	01H	01I	01J	01K	01L	01M	01N

Table of Contents

Chapter 1 - GENERAL INFORMATION

Chapter 2 - INSTALLATION INSTRUCTIONS

Chapter 3 - OPERATING INSTRUCTIONS

Chapter 4 - FUNCTIONAL DESCRIPTION

Chapter 5 - PROGRAMMING EXAMPLES

Chapter 1 - GENERAL INFORMATION

1.1 INTRODUCTION

This manual describes the installation and operation of the MPV955 analog output board.

1.2 GENERAL DESCRIPTION

The MPV955 analog output board is electrically and mechanically compatible with, and interfaces directly to, the VMEbus.

The features of the MPV955 include:

- 8 output channels
- 16 bit resolution
- "Watchdog" timer
- Bipolar/unipolar Output ranges per Channel
- Filtered/Unfiltered Outputs
- 16K words of memory
- Single +5V supply operation

The MPV955 provides eight channels of high-speed analog output on a single board, with two modes of operation - continuous or single-shot.

The MPV955 offers 3 triggering options, namely internal, external and event. The conversion period for both internal and event trigger can be set in software via an on-board programmable timer.

Any of the seven interrupt levels available on VMEbus can be user selected by software and generated by the MPV955. During an interrupt service the MPV955 supplies the user programmable status/ID byte to the CPU when requested.

The MPV955 VMEbus interface capability allows standard addressing with or without address modifier codes. Address modifier response is user selectable by appropriate programming an on-board PROM.

A front panel LED illuminates to indicate that the board is in a halted condition.

1.3 SPECIFICATION

1.3.1 Electrical

Typical at +25^{o}C and rated power supply unless otherwise stated.

1.3.2 Output Characteristics

Number of Channels	8 single ended
D/A output ranges	0 to +10V, 0 to +5, ±10V, ±5V
Output impedance	<1mΩ
Short circuit protection, duration	Indefinite on short to common
Output Drive Current	15mA Max per channel (Resistive Load)

1.3.3 Transfer Characteristic

Resolution	16 bits

1.3.4 Accuracy

Total error	±0.006% FSR[1]
System linearity	±0.006% FSR[1]
Gain error	±0.0075%
Offset error	Adjustable to zero
Monotonicity	Guaranteed (0^{o}C to 60^{o}C)
Temperature gain drift characteristic	20 ppm/°C maximum
Temperature offset drift characteristic	10 ppm/°C maximum

1.3.5 Reconstruction Filter

Cut-off frequency	48.8kHz
Roll-off	80dB/decade
Settling time (max)	1µs to 0.1% of the required final value
	4µs to 0.01% of the required final value

1.3.6 Digital Specification

Rate timer	
(i) Internal trigger	1.5µs to 127.5µs
(ii) External trigger	>1.5µs
(iii) Event trigger	>1.5µs once only
Timeout Control	
(i) Minimum value set be FEH	1.3 seconds
(ii) Maximum value set by 00H	347 seconds
(iii) User selectable resistor option for greater values	

1.3.7 Power Requirements

+5V ±5% at 2.5A typical

1.3.8 Environmental

Operating Temperature	0^{o}C to 60^{o}C
Storage Temperature	-25^{o}C to $+85^{o}$C
Relative Humidity	5% to 90% non-condensing

1.3.9 VME Interface

Board type	Slave
Addressing modes	Standard A24
Data transfers	Word D16

1.3.10 Rev B Compatible

DSA low to DTACK low	300ηs typical
DSA high to DTACK high	200ηs typical
Data transfer rate	4 MBytes/sec

NOTES:

[1] ±10V without filter options

1.3.11 Mechanical

Compatible with double height VMEbus card-racking systems.
Minimum card spacing 20.34mm (0.8") (VMEbus - specification compatible).
One 25 pin ’D’ type connector on board for analog outputs.
For mating connector Cannon ’D’ type connector part number DC25P/IAIN or equivalent.

Chapter 2 - INSTALLATION INSTRUCTIONS

2.1 INTRODUCTION

This chapter provides installation instructions and an overview of the user selectable options on the MPV955.

2.2 PREPARING THE BOARD

This section describes the hardware preparation of the MPV955 prior to installation in a VMEbus system. Following the instructions in this section will ensure that the board is configured as desired and will function as expected when installed.

The MPV955 board is factory calibrated and tested prior to shipment with jumpers in pre-determined positions.

Figure 2.1 indicates the physical position of the jumpers.

Table 2.1 lists each jumper, its function and the factory set position.

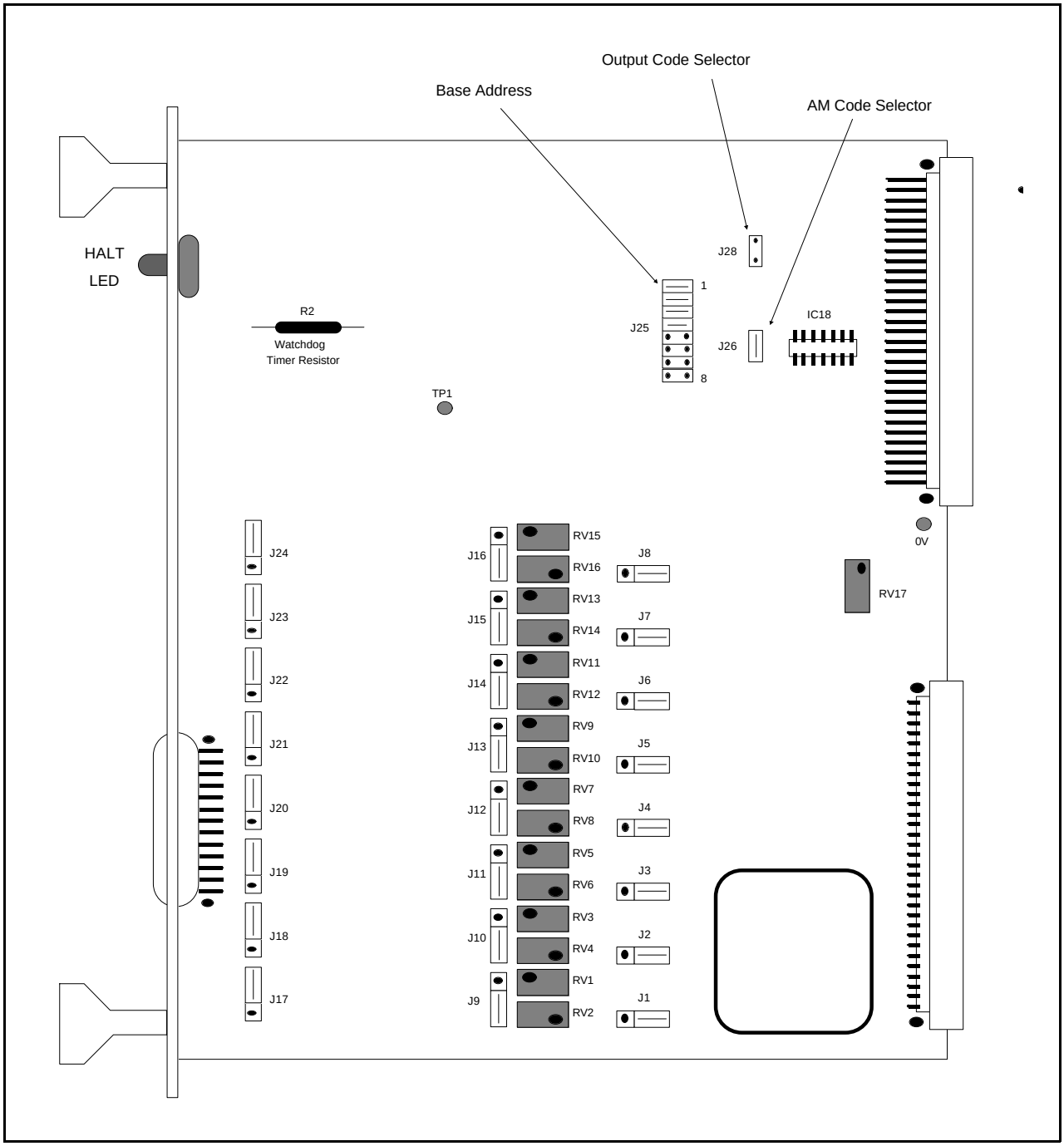
A detailed description of each jumper is given in sections 2.2.1 to 2.2.6.

To alter the address modifier code response from the factory set values, the AM decoding PROM should be programmed as described in section 2.2.6.1.

Before installation, configuration of the jumpers should be verified and, if necessary, altered to suit particular system requirements.

The board should not be installed or removed while power is applied to the VMEbus system.

Figure 2.1 Position of Jumpers and Pots



Channel	Gain Pot	Offset Pot	Voltage Range Jumper	Bipolar/unipolar Jumpers	Filtering Jumpers
0	RV1	RV2	J1	J9	J17
1	RV3	RV4	J2	J10	J18
2	RV5	RV6	J3	J11	J19
3	RV7	RV8	J4	J12	J20
4	RV9	RV10	J5	J13	J21
5	RV11	RV12	J6	J14	J22
6	RV13	RV14	J7	J15	J23
7	RV15	RV16	J8	J16	J24

Table 2.1 Jumper Functions and Factory Settings

JUMPER DESIGNATION	FUNCTION	FACTORY SET CONDITION
J1 to J8	D/A range selection	a 1 2 3
J9 to J16	Select bipolar or unipolar	a 1 2 3
J17 to J24	Filtered/Unfilitered Output Selection	a 1 2 3
J25	Base Address Selection	a 1 2 3 4 5 6 7 8 b
J26	AM Response	1 2 a
J28	AM Code Selector	1 2 a

2.2.1 Base Address Selection

The base address of the MPV955 can be set to any value between FF0000H and 000000H by selecting the appropriate combination of jumpers on J25. The factory set configuration is F00000H.

The MPV955 is memory mapped into 64Kbyte locations within the VMEbus system memory map.

Figure 2.2 shows the location of of jumper J25.

Figure 2.2 Base Address Selection Jumpers

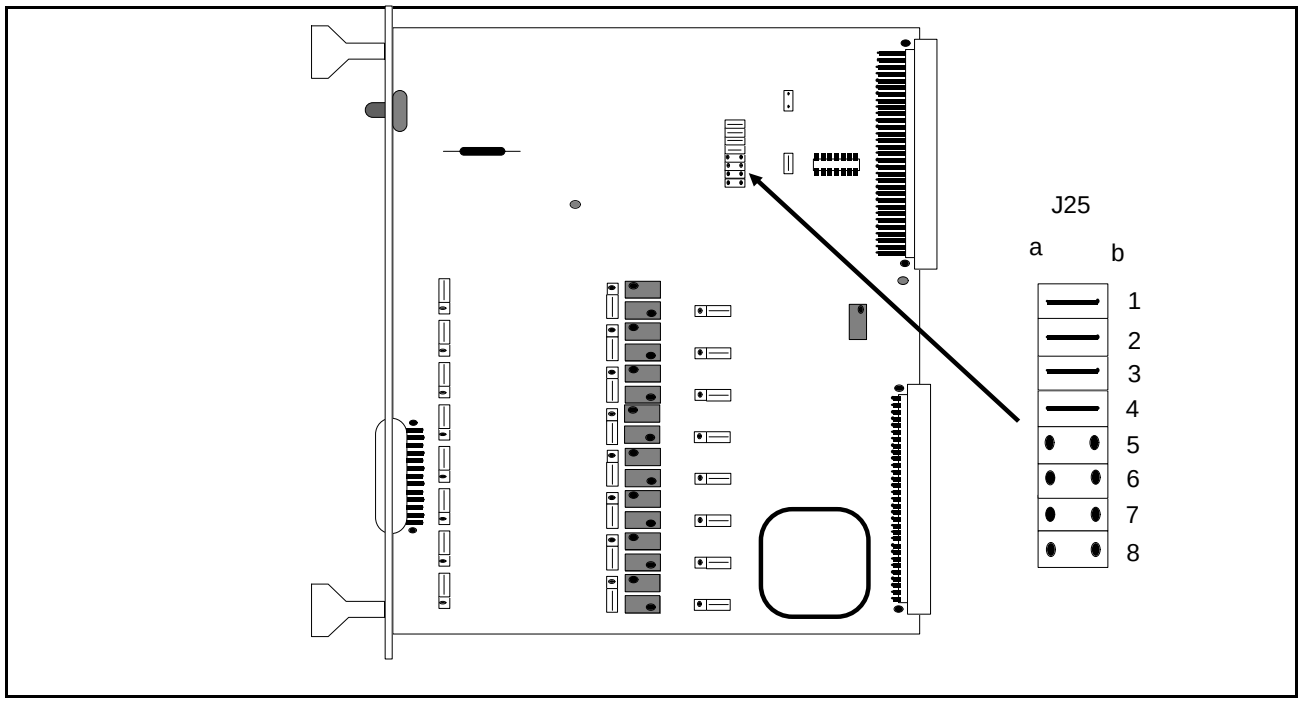


Table 2.2 lists the jumper positions that determine which base address is selected.

Table 2.2 Base Address Jumper Selection

ADDRESS LINE	JUMPER POSITION	FACTORY SET CONDITION
A16	J25 (a1-b1)	0 INSERTED
A17	J25 (a2-b2)	0 INSERTED
A18	J25 (a3-b3)	0 INSERTED
A19	J25 (a4-b4)	0 INSERTED
A20	J25 (a5-b5)	1 REMOVED
A21	J25 (a6-b6)	1 REMOVED
A22	J25 (a7-b7)	1 REMOVED
A23	J25 (a8-b8)	1 REMOVED

2.2.2 Output Range Selection

The range of the analog outputs of the MPV955 are jumper selectable. The jumpers which are used to select different output ranges for each channel are shown in figure 2.3 and table 2.3.

Figure 2.3 D/A Range Selection Jumpers

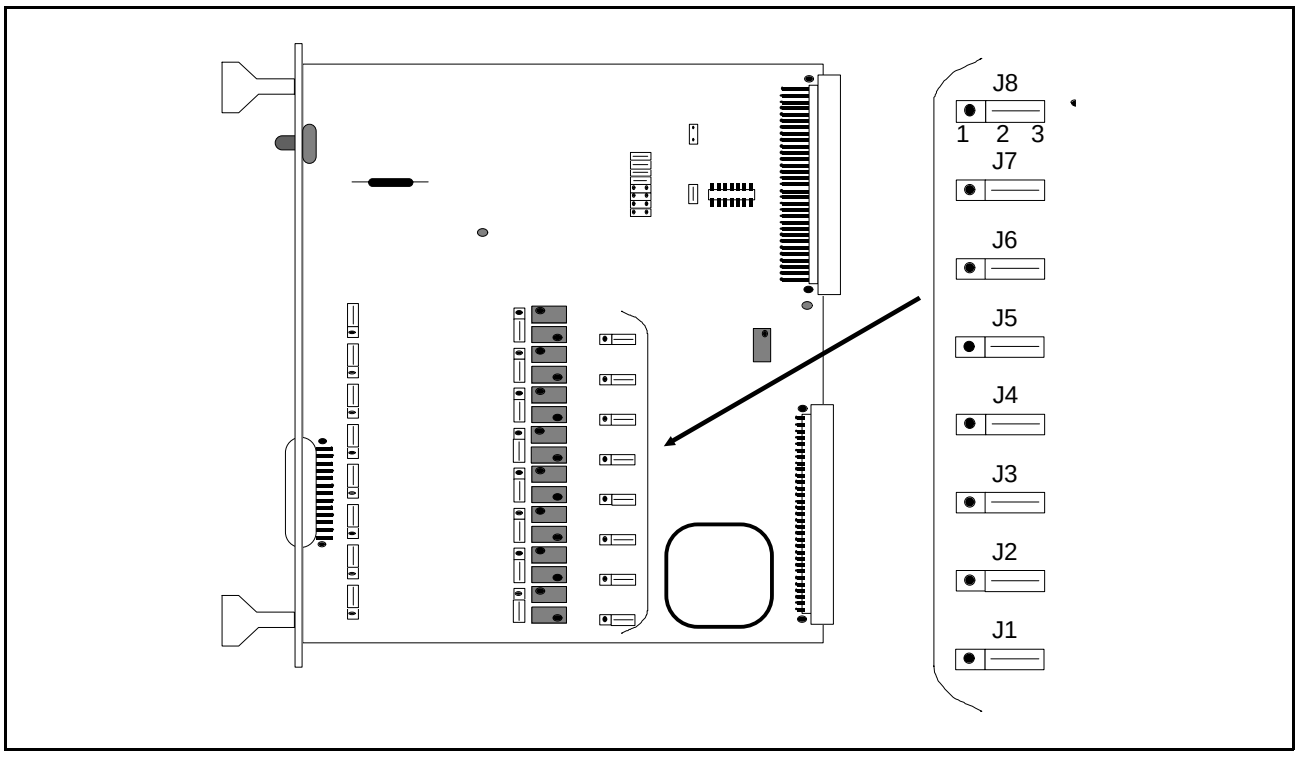


Table 2.3 Analog Output Range Selection

RANGE Unipolar/Bipolar	CHANNEL	JUMPER	INSERT	REMOVE
+10V/±10V	0	J1	2-3	1-2
+5V/±5V	0	J1	1-2	2-3
+10V/±10V	1	J2	2-3	1-2
+5V/±5V	1	J2	1-2	2-3
+10V/±10V	2	J3	2-3	1-2
+5V/±5V	2	J3	1-2	2-3
+10V/±10V	3	J4	2-3	1-2
+5V/±5V	3	J4	1-2	2-3
+10V/±10V	4	J5	2-3	1-2
+5V/±5V	4	J5	1-2	2-3
+10V/±10V	5	J6	2-3	1-2
+5V/±5V	5	J6	1-2	2-3
+10V/±10V	6	J7	2-3	1-2
+5V/±5V	6	J7	1-2	2-3
+10V/±10V	7	J8	2-3	1-2
+5V/±5V	7	J8	1-2	2-3

2.2.3 Unipolar/Bipolar Range Selection

The analog outputs of the board can be either unipolar (0V to +5V or 0V to +10V) or bipolar (-5V to +5V or -10V to +10V), selectable by jumpers J9 to J16. The position of the jumpers is shown in figure 2.4 and the output ranges corresponding to the jumper settings are shown in table 2.4.

Figure 2.4 Unipolar/Bipolar Selection Jumpers

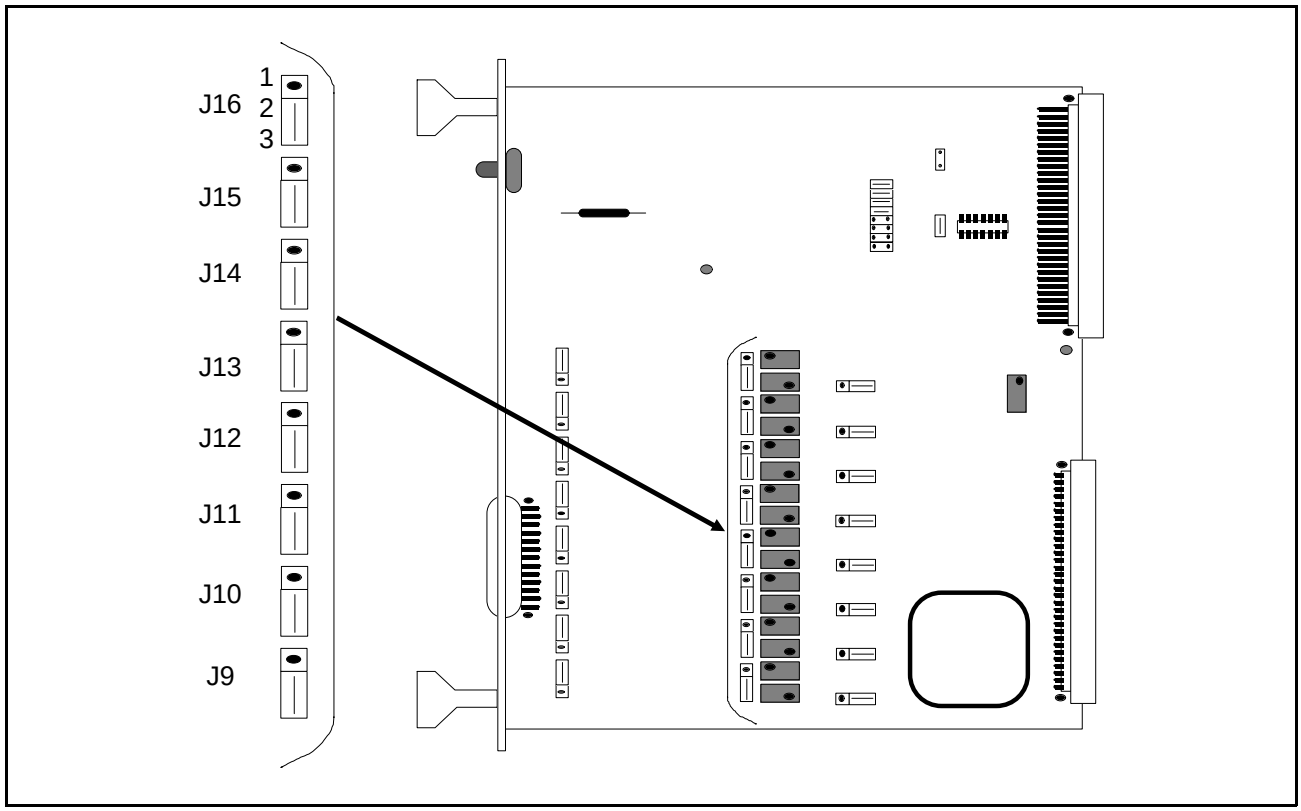


Table 2.4 Bipolar/Unipolar Selection

CHANNEL	OPTION	JUMPER	INSERT	REMOVE
0	Bipolar	J9	2-3	1-2
	Unipolar	J9	1-2	2-3
1	Bipolar	J10	2-3	1-2
	Unipolar	J10	1-2	2-3
2	Bipolar	J11	2-3	1-2
	Unipolar	J11	1-2	2-3
3	Bipolar	J12	2-3	1-2
	Unipolar	J12	1-2	2-3
4	Bipolar	J13	2-3	1-2
	Unipolar	J13	1-2	2-3
5	Bipolar	J14	2-3	1-2
	Unipolar	J14	1-2	2-3
6	Bipolar	J15	2-3	1-2
	Unipolar	J15	1-2	2-3
7	Bipolar	J16	2-3	1-2
	Unipolar	J16	1-2	2-3

2.2.4 Filtered/Unfiltered Output Selection

Filtered and unfiltered output options are available using jumpers J17- J24. The jumpers which are used to select filtered or unfiltered outputs for each channel are shown in figure 2.5 and table 2.5.

Figure 2.5 Filter Selection Jumpers

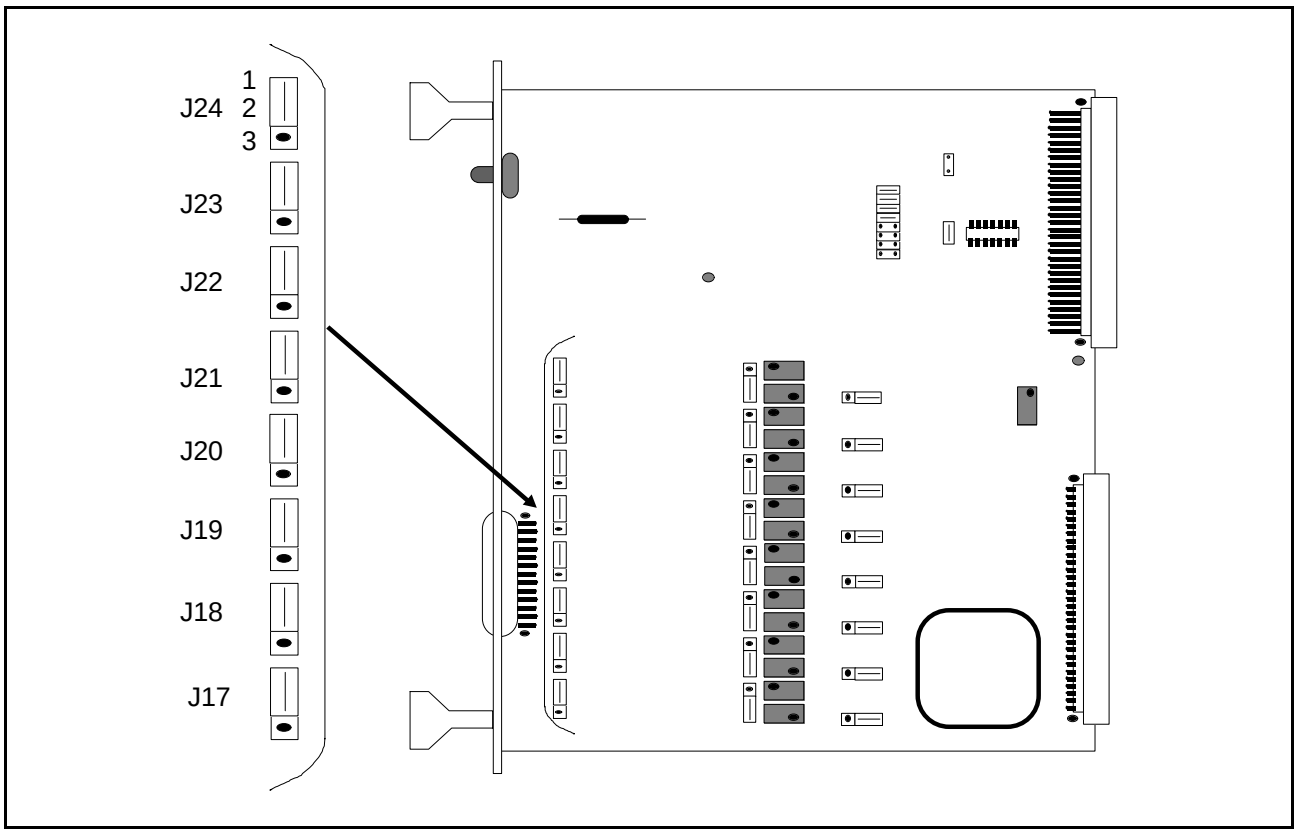


Table 2.5 Filtered/Unfiltered Output Selection

OPERATION	CHANNEL	JUMPER	INSERT	REMOVE
Filtered	0	J17	2-3	1-2
Unfiltered	0	J17	1-2	2-3
Filtered	1	J18	2-3	1-2
Unfiltered	1	J18	1-2	2-3
Filtered	2	J19	2-3	1-2
Unfiltered	2	J19	1-2	2-3
Filtered	3	J20	2-3	1-2
Unfiltered	3	J20	1-2	2-3
Filtered	4	J21	2-3	1-2
Unfiltered	4	J21	1-2	2-3
Filtered	5	J22	2-3	1-2
Unfiltered	5	J22	1-2	2-3
Filtered	6	J23	2-3	1-2
Unfiltered	6	J23	1-2	2-3
Filtered	7	J24	2-3	1-2
Unfiltered	7	J24	1-2	2-3

2.2.5 Digital to Analog Converter Input Code Selection

The input code of the DACs on the MPV955 is selectable via jumper J28. The position of J28 and the options available are given in figure 2.6 and table 2.6.

Figure 2.6 Output Code Selection Jumper

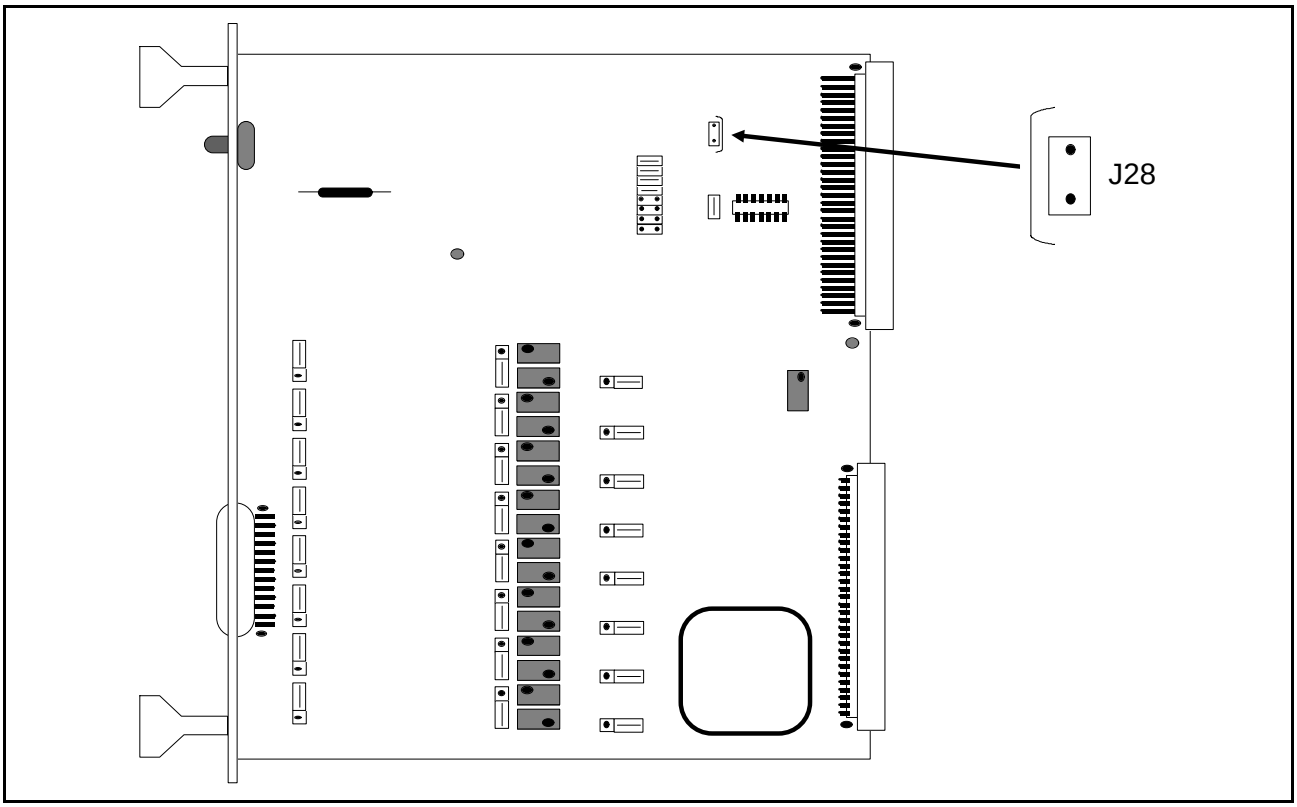


Table 2.6 Output Code Selection

RANGE	JUMPER	CODE	POSITION
UNIPOLAR	J28	Complementary Straight Binary	Removed
BIPOLAR	J28 J28	Offset Binary 2's Complement	Removed Inserted

2.2.6 Address Modifier Code Response

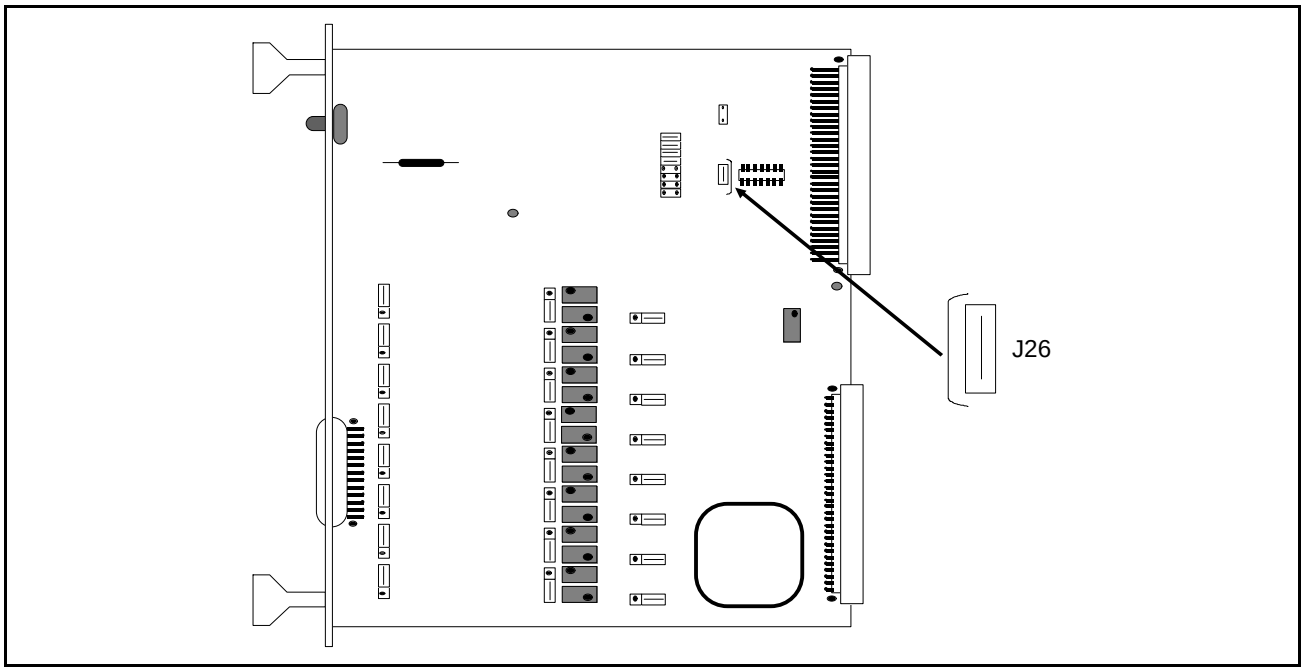
IC18 is a PROM, which in conjunction with J26 positions 1 and 2, allows the response to address modifier codes to be varied to suit system requirements. The prom/jumper combination offers the options shown in table 2.7.

The positions of IC18 and jumper J26 are shown in figure 2.7.

Table 2.7 Address/Address Modifier (AM) Response

JUMPER	POSITION	AM CODES	ADDRESS DECODING
J26	Inserted	N/A	A01-A23
J26	Removed	3D, 39	A01-A23

Figure 2.7 AM PROM and Response Selector Jumper



A new PROM can be programmed to provide an alternative AM code response. The PROM is a 256x4 bit open-collector 82S126 type. The following section explains the interaction between the jumper J26 and the codes stored in the PROM. An understanding of this section should allow the correct jumper/PROM code combination to be selected for a given response.

2.2.6.1 Programming the PROM

The board allows the user to change between any of the two addressing modes simply by changing the position of the jumper J26 (see table 2.7).

Both options are described as follows:

1. Standard Addressing and No Address Modifier.
   The response of the PROM is jumpered out.
2. Standard Addressing With Address Modifier

All the locations in the PROM are set to 03H except where an AM code response is required, in which case the locations are set to 02H. The AM code chosen must be a valid VMEbus specified value. Thus, for example, for option 2 in table 2.7 every location in the PROM will contain 03H except addresses 39H and 3DH which will contain 02H.

2.2.7 The Watchdog Timer Clock

The clock used by the watchdog timer is generated by IC5 (ICM 7555) and it is possible to alter this clock rate by changing the value of resistor R2. The clock period resulting from a given resistor value is given by the formula:

$$T = \frac{R}{5} \text{ (s)}$$

where R is the value of the resistor R2 in MΩ

The factory and supplied resistor R2 is 6.8MΩ, therefore the watchdog clock period is approximately 1.36 seconds.

Chapter 3 - OPERATING INSTRUCTIONS

3.1 INTRODUCTION

This chapter indicates the way in which the MPV955 is used in a VMEbus system. The various control registers and operating modes of the board are individually described.

The board is shipped from the factory in a calibrated condition and ready for use. Installation requires plugging the card into an empty slot in the VMEbus card cage and wiring the analog connector. The factory set condition of the board is described in section 3.9. If a configuration other than the factory set condition is required, details of all jumper settings can be found in Chapter 2.

Many of the boards parameters are software programmable, and their operation is described in this chapter.

3.1.1 Jumper Setting Checklist

Before inserting the board into a VMEbus system and applying power, to ensure correct system operation check that the following options have been correctly set:

1. Base Address - has the base address been set to suit system requirements? Refer to section 2.2.1.
2. Address Modifier Codes - has the Address Modifier response been correctly set? Refer to section 2.2.6.
3. Output Range and Mode - has the correct voltage output range and mode been selected? Refer to section 2.2.2, 2.2.3 and 2.2.5.

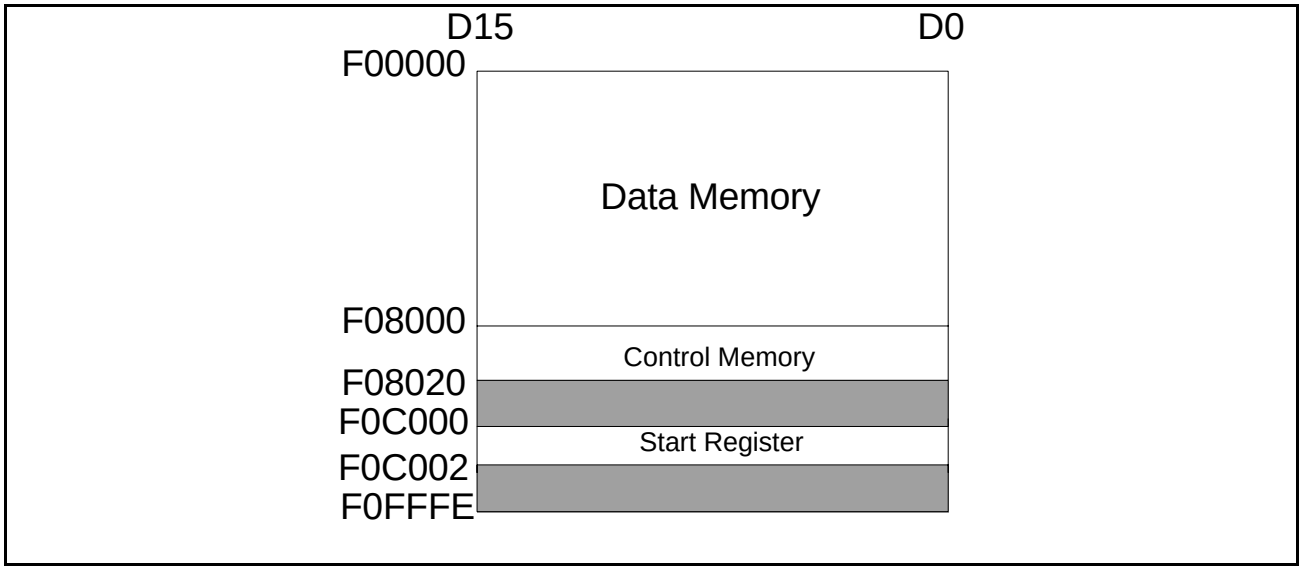
3.2 PROGRAMMING MODEL

The MPV955 appears to the VMEbus as a 64Kbyte block, whose base address is jumper selectable in the 16Mbyte address range of the VMEbus on 64Kbyte boundaries. The analog output has a resolution of 16 bits, therefore word accesses are required when writing data to the board.

3.2.1 Address Map

The address map (memory map) of the MPV955 is shown in figure 3.1 as set at the factory. The base address is jumper selectable (refer to section 2.2.1).

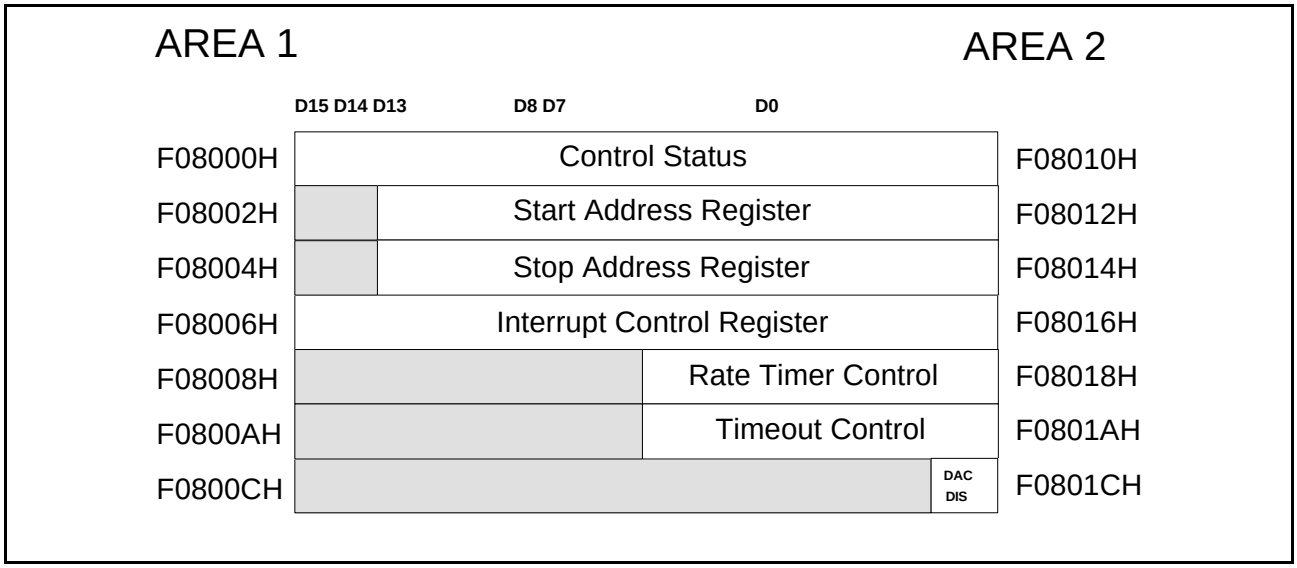
Figure 3.1 Address Map of the MPV955



In most circumstances, the MPV955 will be read from and written to by a CPU board on the VMEbus. The CPU will write data words to be converted to analog voltages into the MPV955 data memory. The CPU will also write to the control registers of the MPV955 to set up the required mode in which the data will be output. Finally, to initiate the output sequence, the CPU will execute a READ or WRITE operation to the last quarter of the memory map of the MPV955. This last quarter is called the START REGISTER, even although no data is stored in this part of the board.

The control registers of the MPV955 appear between the factory-set addresses of FO8000H and FO8020H. NOTE: No VME bus Cycles accessing the reserved areas of the MPV955 are premitted. These areas are reserved for future development. The address map of the MPV955 control registers is indicated in figure 3.2.

Figure 3.2 Control Registers



The function of each of these control registers is described in the following sections 3.2.2 to 3.2.8 inc. The function of Address Area 1 and Area 2 is described in para 3.2.9

3.2.2 Control/Status Register

This register incorporates the control and status bits which determine the mode of operation of the MPV955 The order of these bits is shown in Figure 3.3. Write operations are carried out on data bits D0-D7 to set up control of the board. READ operations can be carried out on data bits D0-D12 to verify the status of the board.

The function of these bits is as follows:

Figure 3.3 Control/Status Register

D15	D14	D13	D12	D11	D10	D9	D8
N/A				OVER SAMP	CYCFIN	TIMEOUT	HALT

D7	D6	D5	D4	D3	D2	D1	D0
N/A	CHANNEL SELECT			TIMEOUT ENABLE	CONT/ ONE	NORMAL/ EVENT	INT/ EXT

D0 - $\overline{\text{INTERNAL}}$/EXTERNAL Trigger Select

When this bit is reset to "0", data is output from the board at a rate set by the on-board timer. When D0 is set to "1", data is clocked out by a negative-going edge on the external trigger input (pin 12 on P3).

D1 - $\overline{\text{NORMAL}}$/EVENT Trigger

When this bit is set to "1", a single negative-going edge on the external trigger input will initiate a series of outputs under control of the on- board timer. Note that if this bit is set to "1" while external trigger is selected (D0 = "1"), then the board will be in an undefined state.

D2 - $\overline{\text{CONTINUOUS}}$/ONE-SHOT Mode Select

When this bit is reset to "0", data contained in the data memory will be continuously output through the DAC channels until the board is stopped by the CPU. With D2 set to "1", the board will output the data in the memory once, and then stop.

D3 - $\overline{\text{TIMEOUT ENABLE}}$

If this bit is reset to "0", then the on-board watchdog timer will be enabled. Conversely, the watchdog timer will be disabled if D3 is set to "1".

D4-D6 - OUTPUT CHANNEL Select

The bit pattern written to these bits selects the channel(s) on which data is output, according to the table in figure 3.4.

Figure 3.4 Channel Select Truth Table

D6	D5	D4	CHANNEL(S) SELECTED
0	0	0	Ch0
0	0	1	Ch0, 1
0	1	0	Ch0, 1, 2
0	1	1	Ch0, 1, 2, 3
1	0	0	Ch0, 1, 2, 3, 4
1	0	1	Ch0, 1, 2, 3, 4, 5
1	1	0	Ch0, 1, 2, 3, 4, 5, 6
1	1	1	Ch0, 1, 2, 3, 4, 5, 6, 7

D7 - Unused

Reading or writing to this bit has no effect on the MPV955.

D8 - $\overline{\text{HALT}}$

This bit cannot be written to, but it will be reset to "0" by the board on power-up, application of a SYSRESET signal, or if the board stops updating the DAC output channels because of a watchdog time-out. The Halt bit will be set to "1" (ie. inactive) when the MPV955 starts an output sequence.

D9 - TIMEOUT

This bit cannot be written to. It is set to "1" when the watchdog timer "times-out" ie. the timeout interval set has elapsed. This bit will be reset to "0" when the Start Register is written to. Operation of the watchdog timer is explained further in section 3.3.2.

D10 - CYCLE FINISHED

This bit cannot be written to, but it will be set to "1" by the MPV955 once a complete set of data has been output through the DAC channels. This bit will be reset to "0" when the MPV955 is restarted, ie. the Start Register has been written to.

D11 - OVER-SAMPLING

This bit cannot be written to. It will be set to "1" by the MPV955 if the DAC channels are supplied with "convert" commands faster than their highest output rate (ie. greater than 666kHz). This means that either the on board-timer or the external trigger input is running too fast.

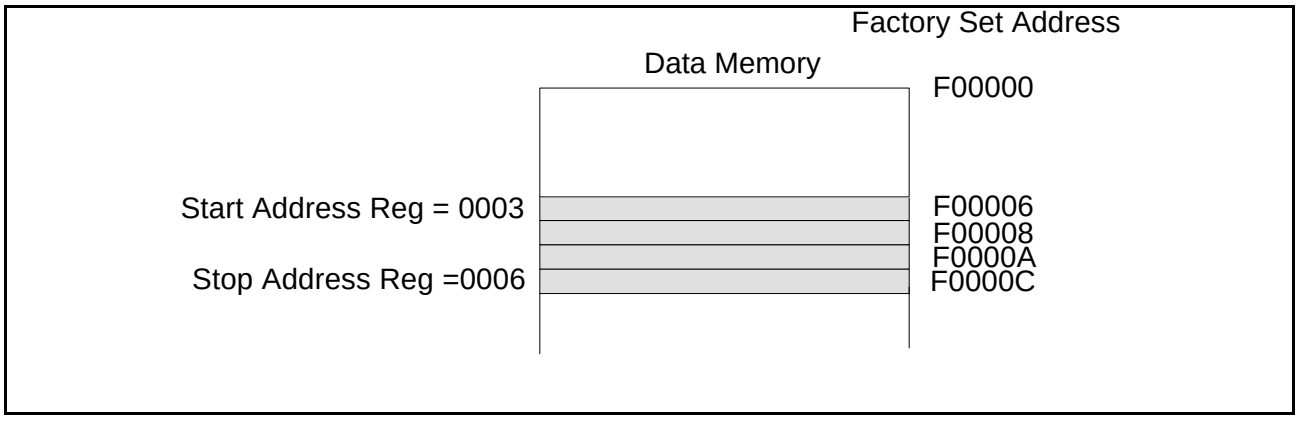
D12-D15 - Unused

Reading or writing to these bits has no effect on the MPV955.

3.2.3 Start Address Register and Stop Address Register

The MPV955 has 16kwords of data memory. READ or WRITE operations may be performed anywhere in this memory at any time. However, the Start Address Register and the Stop Address Register define the section of data memory from which the D/A converters read the data to be output from the board. These registers perform in the manner that their names suggest. The Start Address Register contains the address of the first data word to be output by the D/A converter, and the Stop Address Register contains the address of the last data word to be output. It is important to note that the address contained in these registers are word addresses, not byte addresses, as on the VMEbus. For example, if it is desired that four data words be output starting from the third data memory location, then the value 0003H should be written to the Start Address Register, and the values 0006H should be written to the Stop Address Register. This is illustrated in figure 3.5.

Figure 3.5 Start and Stop Address Operation



It should be noted also that with 16k words of data memory, the value of the most significant 2 bits of both registers has no effect on the MPV955.

In normal operation, the value of the contents of the Stop Address Register will be greater than the value of the contents of the Start Address Register. However, if the two values are equal, then the board will output the data word at that address but no other data words.

If the stop address is smaller than the start address, then the addresses of the data words output from the board will "wrap-round" through "0" and increment until the stop address is reached. For example, if the start address is set to the value 0FF8H and the stop address is set to the value 0003H then the data at addresses 00FF8H - 3FFFH inclusive will be output followed by the data at addresses 0000H - 0003H inclusive.

Note: Both the Start Address Register and the Stop Address Register are 14 bit wide. On reading either of these registers Bits D14 and D15 should be masked.

3.2.4 Interrupt Control Register

The MPV955 incorporates circuitry to generate an interrupt on the VMEbus. The interrupt priority, the interrupt status/ID byte, and the event generating the interrupt are all programmable by this register. The layout of the various control bits is shown in figure 3.6.

Figure 3.6 Interrupt Control Register

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	IOS	ICY	ITI		INTERRUPT PRIORITY			STATUS/ID BYTE							

READ and WRITE operations can be performed on all fields in the Interrupt Control Register.

D0-D7 - STATUS/ID Byte

This field is the byte supplied to the VMEbus interrupt handler during an Interrupt Acknowledge Cycle. (For a detailed description on the function of this byte, see the VMEbus specification, revision C, section 4).

D8-D10 - INTERRUPT PRIORITY

These 3 bits determine which one of the seven interrupt lines the MPV955 uses to request an interrupt, according to the table in figure 3.7.

Figure 3.7 Interrupt Priority Truth-Table

D10	D9	D8	INTERRUPT REQUEST LINE
0	0	0	Interrupts disabled
0	0	1	IRQ1 *
0	1	0	IRQ2 *
0	1	1	IRQ3 *
1	0	0	IRQ4 *
1	0	1	IRQ5 *
1	1	0	IRQ6 *
1	1	1	IRQ7 *

D11 - Unused

Reading or writing to this bit has no affect on the MPV955.

D12-D14 - INTERRUPT ENABLE Bits

On the MPV955, there are 3 possible sources of interrupts ie. an over- sampling indication, a watchdog timeout, or an output cycle complete indication. These 3 states correspond to the setting to “1" of bits D14, D12 and D13 respectively, of the Control/Status Register. The generation of an interrupt on entering one or several of these states is individually programmable via bits D12, D13 and D14 of the Interrupt Control Register.

When bit D12 of the Interrupt Control Register (ITI), is set to “1", an interrupt will be generated by the watchdog timer timing-out.

If bit D13 (ICY) is set to “1" then an interrupt will be generated when an output cycle is complete.

Similarly, an interrupt will be generated when the DACs are over- sampled, and bit D14 (IOS) is set to “1".

Note that if more than one of these bits is set to “1", then an interrupt will be generated on entering any of the states which have interrupts enabled. The generation of an interrupt can be written as a boolean equation:

Interrupt = ITI.(Timeout) + ICY.(Cycle Finished) + IOS.(Over-Sampling)

D15 - Not used.

Reading from or writing to this bit has no effect on the MPV955

3.2.5 Rate Timer Register

The rate at which fresh data is output from the MPV955 analog outputs is controllable using this write-only register. Note, however that to use this feature the internal trigger must be enabled, ie. bit D0 of the Control/Status Register is set to "0".

The output rate is programmable in 500 ηs increments from 0 to 127.5 µs. However, since the maximum update rate of the DACs is 1/1.166 usec (857kHz), setting the output rate to 0, 0.5 or 1 usec will produce an over-sampling indication.

The rate timer is implemented on the MPV955 as an 8-bit up-counter. The 8 bits of the counter are mapped into the least significant byte of the register.

The counter produces a DAC trigger pulse on reaching FFH. Therefore writing FFFBH to the register will result in an output rate of 2 µs (500kHz). The timer counts 4 times, outputs a pulse, is reloaded with FBH and repeats, again and again. Similarly, writing FFF0H will result in an output rate of 133.33kHz. The formula for calculating the output period for a certain value in the Rate Timer Register is:

Output rate = (255 - (Register Value)) x 500 ηs

Writing the value FFFFH in the Rate Timer Register is illegal, since it represents a sampling period of 0 seconds!

Since this register is write only, the register cannot be read to find out the value last written to it. Separate note of this value must be kept in the VMEbus system.

3.2.6 Timeout Control Register

A novel feature of the MPV955 is the watchdog timer. Its function is described fully in later sections of this manual, but briefly the principle of operation is as follows. When enabled i.e. bit D3 of the Control/Status Register is set to "0", the watchdog timer will count up over an interval. If this interval is complete before the VMEbus CPU resets the counter by accessing the Start Register, then a timeout occurs and the analog outputs are forced to zero volts. This is intended to enhance the reliability and fault tolerance of the VMEbus system.

Like the Rate Timer, the Watchdog Timer is implemented as an 8-bit up- counter. The 8 bits of the counter are mapped to the least significant byte of the register. The counter increments in units set by the watchdog timer clock (section 2.2.8). The factory set condition is units of 1.3 seconds. Thus, loading the Timeout Control Register with a value of FFFEH, will result in a timeout after 1.3 seconds. Similarly writing FFF0H, will result in a timeout interval 19.5 seconds. The formula for calculating the timeout interval for a certain value in the Timeout Control Register is:

Timeout Interval = (255 - (Register Value)) x 1.3 seconds

Note this register is write-only, and the value last loaded into it cannot be read back.

3.2.7 DAC Disable Register

Bit D0 of the register allows the user to disable the analog outputs, by setting this bit to a 1. The power up condition is D0=0

3.2.8 Start Register

An access of the start register will initiate conversions on the MPV955. Although termed a register, no data is stored - it merely starts conversions.

Upon receipt of the start command the timeout counter loads the timeout value into the timeout register. If another start command is not received during the time the board is active then timeout will occur. It is therefore important to regularly access the start register within the active period of the board at which point the timeout value will be reloaded and the counter will start to count up again from the timeout value originally loaded.

The Rate Timer, as discussed previously is an 8 bit up counter and timeout will occur (if enabled) upon reaching FFH, i.e. RCO will be asserted LOW for one clock period of the watchdog timer clock. Since the frequency of this clock can be altered by the user (2.2.8) it is important to realise that for the duration of RCO being asserted LOW (1 clock period), accessing the start register will have no effect on the MPV955 - it will remain HALTED.

If timeout is not important then this feature can be disabled and only one start command is necessary.

3.2.9 Address Area 1 And Area 2

Write accesses to the MPV955 control registers at Area 1 will automatically HALT output operation. Output channels shall remain at their previously loaded values.

Write Accesses to control registers using Area 2 addresses shall not HALT the output operation of the MPV955

Accessing the START ADDRESS REGISTER and STOP REGISTER at Area 2 addresses whilst the board is outputting, facilities the implemation of a software controlled swinging buffer mechanic using MPV955 data memory.

This feature is particularly valuable where large amounts of output data (greater than 16K words) must be output continuously. An example program implementing a swinging buffer is shown in section 5.4.

3.3 ENABLING THE ANALOG OUTPUTS

One of the features of the MPV955 is the way in which the analog outputs are forced to zero volts on either power-up, application of a SYSRESET* signal to the board, or timeout of the watchdog timer. This is intended to provide a certain amount of protection for the devices being driven by the MPV955 outputs from any possible transients during power-up of the MPV955 or failure of the VMEbus system.

The front panel LED illuminates when the DACs are not being updated from the memory. The LED will therefore switch on when the board has completed an output sequence (e.g. in one-shot mode) or has been halted by an exception condition (e.g. watchdog timeout).

3.3.1 Power-up and SYSRESET

The MPV955 should not be installed or removed while power is applied to the VMEbus system. After insertion of the board and on power being applied to the VMEbus system, all 8 analog output channels will be forced to 0 Volts. On accessing the start register all output channels become active so if it is desired that unselected channels remain at 0V then the appropriate channels must be written to before accessing the start register.

Similarly the analog outputs are also forced to 0V on application of the VMEbus global reset signal SYSRESET*. The outputs are enabled in the same manner as before i.e. access the start register.

To prevent spikes due to indeterminate data in the DAC internal latches, the follow procedure is recommended.

- Store data for 0V in the first 16 memory locations
- Set bit D0 of DAC disable to 1
- Set the board for 8 channels, one shot mode, internal trigger
- Write to start address register 0000H
- Write to stop address register 000FH
- Write to start register to initiate output sequence
- Wait for cycle finished bit in status register to be set
- Reset bit D0 of DAC disable register

3.3.2 Using Watchdog Timer

The watchdog timer is located at the factory set address of F0800BH, i.e. the least significant byte of the Timeout Control Register. To use this feature effectively the timeout clock frequency must be known before hand. Section 3.2.6 describes this and section 2.2.8 provides a means to change the clock frequency simply by altering a resistor value.

When the clock frequency is known, a meaningful value can now be loaded into the timeout register. Since this register is an up-counter then for a given clock frequency 00H will give the longest time before a timeout and FEH will give the shortest time before a time-out.

Once a value has been loaded into this register it is important to check that the Timeout Enable Bit of the control register is logic "0" to enable the timer. The board can now be started by writing to the start register. The time out counter will begin to increment at a rate defined by the timeout clock. When the counter reaches FFH, the board will timeout. This can be prevented by accessing the start register within the timeout period specified, and the count up starts again.

If a timeout occurs, the timeout bit in the status register wil set to logic "1". The board can be restarted by a READ or WRITE operation to the start register.

This feature can be disabled by applying a "1" to bit D3 of the control register which disables the clock input to the timeout register.

The time out clock is not accurate (±25%) due to tolerances of the devices used to set the time period. The timeout control should not be used for timing applictions and is only intended as a means of resetting the analog outputs to 0V if a fault on the VMEbus occurs. (i.e. system hang-up).

3.4 TRIGGERING MODES AND OPERATION

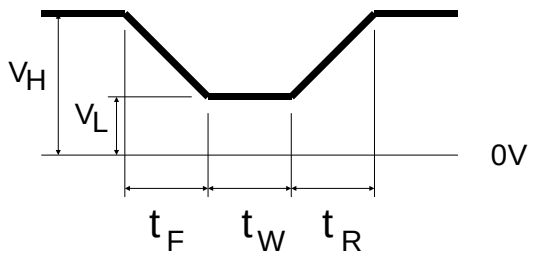
A new value will be output on the falling edge of the trigger signal (no matter if the trigger signal is internal, external or event). This trigger signal is available on the P2 and P3 connectors. There are, as previously mentioned, a number of different modes of triggering available as described in the following sections.

3.4.1 Internal Trigger

The internal trigger mode is set-up by writing a "0" into bits D0 and D1 of the control register. If, on internal trigger, a value is loaded into the rate timer which results in a trigger to the DACs before the previous DAC cycle has been completed, an oversampling indication in the status register results.

3.4.2 External Trigger

Figure 3.8 External Trigger Signal



Parameter	MAX	MIN
V_L	+0.5V	-10V
V_H	+25V	+3.5V
t_R	20µs	0µs
t_F	20µs	0µs
t_W	-	300ηs

A TTL compatible input signal can be used to externally trigger the board. This enables synchronisation of conversions with external or off- board events. Details of the trigger signal are shown in figure 3.8 below.

The data acquisition and conversion process is initiated on the falling edge of the trigger signal.

As with internal trigger the maximum trigger repetition rate must not exceed the rate at which digital output data can be converted to an analog signal since this will also cause an oversampling indication in the status register.

After the falling edge of the trigger there is a delay before the converted data appears at the analog output. This will be no longer than 2.5µs. The following sequence of operations should be performed when using external trigger:

1. Set bit D0 of control register "1" ie. enable external trigger.
2. Ensure that bit D1 is "0", since if this bit were logic "1" when D0 is "1" the board is in an undefined state.
3. Select the number of channels required for output.
4. Fill the appropriate memory locations with the data to be converted.
5. Write to the start register.
6. Apply external trigger.
7. Monitor the status register oversampling bit to determine if external trigger is too fast.

3.4.3 Event Trigger

Event trigger is set by writing a "1" to bit D1 of the control register and ensuring that bit D0 of the control register is "0". In event trigger mode a single external trigger pulse initiates D/A conversion. However, the rate at which D/A conversion occurs is set not by the external trigger used to initiate the conversions but by the value loaded into the rate timer. It is important that event trigger occurs only once since any other event trigger signals will cause the rate timer value to be loaded at random. This prevents a trigger pulse from occuring and thus the output cannot be updated.

After the intial external trigger pulse is applied there will be a short delay until the converted data appears at the analog outputs. This will be no longer than 2.5µs

The following sequence of operations should be performed when using event trigger:

1. Set bit D0 to "0" (internal trigger)
2. Set bit D1 to "1" (event trigger)
3. Select the number of channels required for output.
4. Fill the appropriate memory locations with the data to be converted.
5. Load the internal rate timer with the desired triggering rate.
6. Write to the start register.
7. Apply external trigger once only.
8. Monitor the status register for an oversampling indication.

3.5 ANALOG OUTPUT MODES

Through appropriate programming of the control registers, the MPV955 can be set up to operate in a number of different modes. However in all cases the procedure for using the board is the same.

After installing the MPV955 in a VMEbus system, and checking all the jumpers are correctly set (see chapter 2), a WRITE operation to the board's control registers will be carried out to set up the required mode of operation. Then, the data words which are to be converted to an analog output should be loaded into the MPV955 data memory. Finally, the Start Register will be accessed to initiate the conversion of the data to analog outputs. The MPV955 may be accessed while conversions are taking place, either to inspect status bits in the control registers, or to update data in the data memory. By writing to the data memory, the analog waveforms being output will be changed. The MPV955 will stop converting new data words when told to do so (i.e. one-shot mode), or by writing to any one of the control registers in area 1.

The MPV955 will always stop conversion if a WRITE operation is carried out to any of the control registers in area 1. This is because writing to the control registers may change the mode of operation of the board and there would be uncertainty as to what exactly happens to the outputs during this change.

Note that while the board is stopped, the analog outputs will not decay to zero. Instead, the outputs will remain at whatever level was being output when the board was stopped. This is due to the architecture of the digital to analog converters. The DACs used on the MPV955 are double-buffered devices (MP7636). That is to say, there are two data registers on the device. One register is loaded from the input pins of the

DAC, and its outputs drive the inputs of the second register. It is from the outputs of this second register that the analog output is derived. It is clear from this that the DAC will hold its analog output constant as long as the contents of the registers remain constant and power to the DACs is maintained.

One further consequence of the doubled buffered DACs is the indeterminate value of the analog output on the first trigger signal after writing to the Start Register.

When a trigger signal is given to a DAC on the MPV955, the contents of the first DAC register are transferred to the second register, and an analog signal corresponding to these contents appears on the DAC output. This means that on each channel selected, the first output appearing after starting the board will be some unknown value originating either from previous use of the board, or power-up of the board.

See section 3.3.1 for details of how to prevent spikes due to indeterminate data contained in the DAC internal latches. Also see the short program at the back of the manual which clears the DAC internal latches.

If the MPV955 has stopped conversion, and the board is restarted at the address by accessing the Start Register, data conversion will resume using the data pointed to by the Start Address Register. The MPV955 will not pick up from where it left off when it was stopped.

3.5.1 Single Channel Output

The main features of the MPV955 have been described. The following sections illustrate various operating modes and how different output waveforms can be obtained from the board.

3.5.1.1 Output a D.C. Value

1. Set up the control register for one channel on single-shot mode by writing FF04H to the control registers address. **Note** that only the least significant byte is loaded into the register i.e. 04H.
2. Write the same value into the start and stop address register (0000H in this example).
   Start Address Register value = 0000H
   Stop Address Register value = 0000H
3. Write a value into the data memory at the location specified by the start/stop address (i.e. 0000H).
   e.g. Location 0000H = 8000H
4. Read or write to the Start Register.
5. Repeat 4.

Due to the architecture of the DACs it is important that the first data value converted is ignored. This is because of the double-buffering technique which is explained in section 3.5.

On the second write to the start register the board will convert the data word held at location 0000H. Since 8000H is 1000 0000 0000 0000 in binary, the d.c. value is 5V if the board is set up for complementary straight binary in a +10V range.

If the control/status register is now examined it is found that D10 has been set to logic “ 1" signifying that the cycle has been completed and the board has now stopped data output.

3.5.1.2 Square Wave

A square wave can be achieved very simply by loading two memory locations with different digital words and altering the control register for continuous output.

1. Set up control/status register for one channel, continuous output i.e. write FF00H to the control register.
2. Set up the Start and Stop Address Registers.
   Start address register value = 0000H
   Stop address register value = 0001H
3. Write values into data memory
   e.g. Location 0000H = FFFFH
   Location 0002H = 8000H
4. One WRITE operation to the Start Register will initiate conversion.

This set of instructions will initiate conversion to channel 0 and will alternate between the data held in the two memory locations previously specified (0000H and 0002H). The result is a square wave of 5V amplitude using complementary straight binary in a +10V range.

Note that the amplitude and d.c. offset of the square wave can be changed at any time simply by writing different values to the data memory, even while the DAC is outputting the waveform.

3.5.1.3 Staircase Waveform

The staircase waveform is simply an extension of the previous examples making greater use of data memory. The steps taken are as follows:

1. Control word = XX08H (disable timeout)
2. Start address register Value = 0000H
   Stop address register Value = 000FH
3. Data Memory Locations:

   Location 0000 = 0000
   0002 = 1000
   0004 = 2000
   0006 = 3000
   0008 = 4000
   000A = 5000
   000C = 6000
   000E = 7000
   0010 = 8000
   0012 = 9000
   0014 = A000
   0016 = B000
   0018 = C000
   001A = D000
   001C = E000
   001E = F000
4. Write to the Start Register.

This simple extension of the previous examples shows a staircase waveform which can be observed. Note that there are 16 different values of data and that the stop address is not 0016 but 000F since these values are hexadecimal. **Note** also that the Stop Address Register value is not 001E (the last place data is stored), but 000FH i.e. 16 memory locations.

By using more data memory locations it is clear that greater resolution of this waveform is possible. Similarily it is easy to create a triangular waveform by extending the data locations and repeating the data.

3.5.2 Multi-Channel Output

Multi-channel output is achieved by writing a value (other than zero) into the control register bits D6, D5 and D4 as previously defined in section 3.2.2.

Due to the functional behaviour of the DACS it is important to ensure that the very first conversion data ignored. This is because of the double buffering technique which makes it impossible to know the data value initially stored in the latches on power-up. It should also be remembered that in multi-channel output mode, the data to each DAC is interleaved in the memory block.

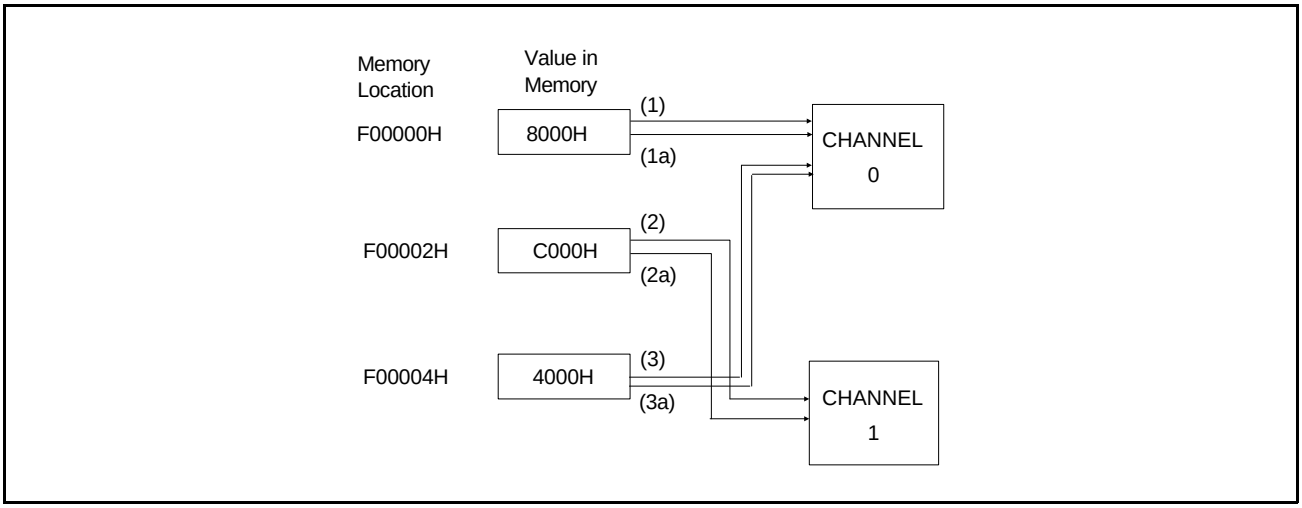
An example of two-channel output follows:

1. Control register = XX18H (2 channels, disable timeout)
2. Start Register = 0000H
   Stop Register = 0001H
3. Location 0000H = 8000H (5V analog)
   0002H = C000H (7.5V analog)
4. Write to the Start Register.

If the output channel 0 is examined it is found to be 5V d.c. and the output of channel 1 is 7.5V d.c.

Consider, now, the situation where the number of channels on which data is to be output is not an integral number of data values, for example two channels with three data values to be output. The output sequence is shown in Figure 3.9.

Figure 3.9 Output Sequence Example



(1) The first data value is output to channel 0

(2) The second data value is output to channel 1

(3) The third data value is output to channel 0

The channel selector is then reset and cycled around the same channels.

(1a) The first data value is output to channel 0

(2a) The second data value is output to channel 1

(3a) The third data value is output to channel 0

The result is a square wave on channel 0 and a d.c. level on channel 1.

3.6 ANALOG OUTPUT SECTION

The analog output section offers three output coding schemes - complementary straight binary (for unipolar ranges) and offset binary or 2's complement (for bipolar ranges).

3.6.1 Analog Output & Converter Input Codes

Tables 3.1, 3.2 and 3.3 illustrate the full-scale negative, midscale and full-scale positive output voltages for the output ranges with the corresponding digital input codes of the digital to analog converter and the value of one least significant bit (LSB).

Figures 3.10, 3.11 and 3.12 show, in graphical form, the relationship between the digital input codes and the output voltages.

For example, in table 3.2 a digital input code of 0000H corresponds to an output voltage of -9.99969V in the ±10V range. In the same range a value of FFFFH will produce an output voltage of +10V.

Table 3.1 Complementary Straight Binary

0V to +5V Scale

	VOLTAGE	CODE	RESOLUTION
Negative Full Scale	0.000V	FFFFH	76µV
Midscale	+2.500V	8000H	
Positive Full Scale	+4.99992488V	0000H	

0V to +10V Scale

	VOLTAGE	CODE	RESOLUTION
Negative Full Scale	0.000V	FFFFH	152µV
Midscale	+5.000V	8000H	
Positive Full Scale	+9.999848V	0000H	

Figure 3.10 Complementary Straight Binary

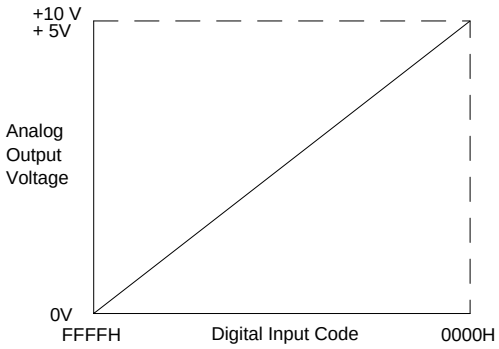


Table 3.2 Offset Binary

±5V Scale

	VOLTAGE	CODE	RESOLUTION
Negative Full Scale	-4.999848V	0000H	152µV
Midscale	0V	7FFFH	
Positive Full Scale	+5V	FFFFH	

±10V Scale

	VOLTAGE	CODE	RESOLUTION
Negative Full Scale	-9.99969V	0000H	310µV
Midscale	0V	7FFFH	
Positive Full Scale	+10V	FFFFH	

Figure 3.11 Offset Binary

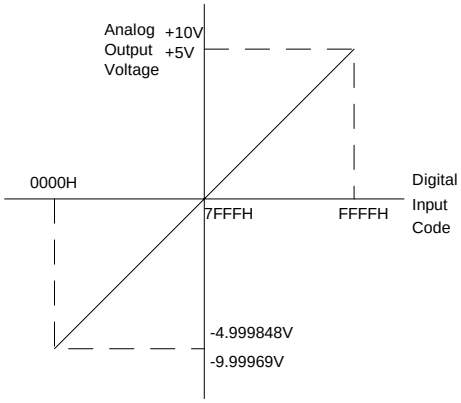


Table 3.3 2's Complement

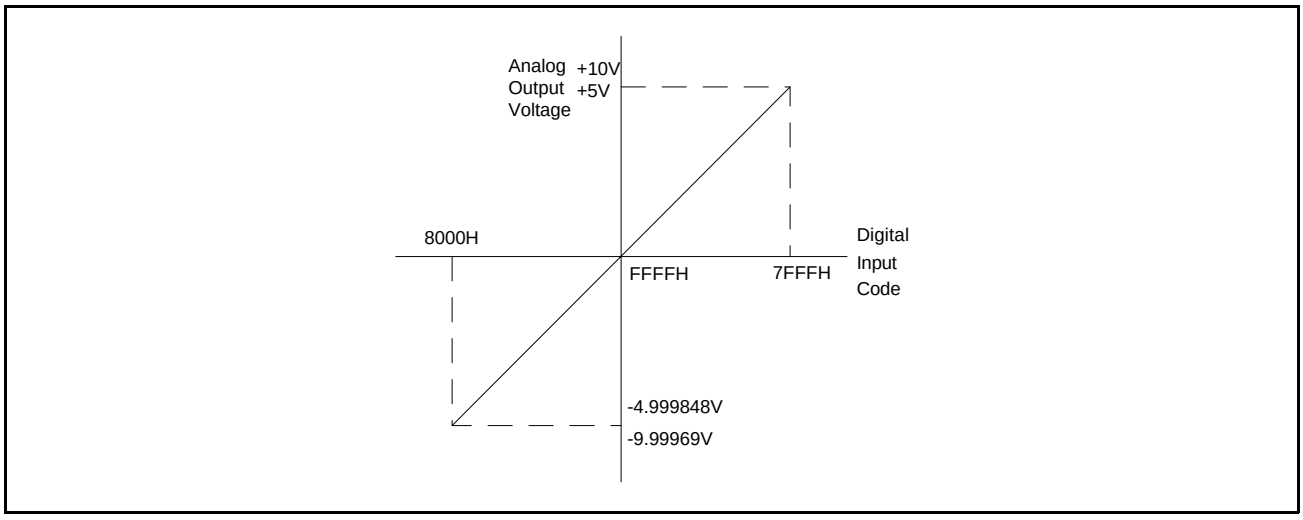
±5V Scale

	VOLTAGE	CODE	RESOLUTION
Negative Full Scale	-4.999848V	8000H	152µV
Midscale	0V	FFFFH	
Positive Full Scale	+5V	7FFFH	

±10V Scale

	VOLTAGE	CODE	RESOLUTION
Negative Full Scale	-9.99969V	8000H	310µV
Midscale	0V	FFFFH	
Positive Full Scale	+10V	7FFFH	

Figure 3.12 2's Compliment



3.6.2 Calculating Output Voltage from Input codes

For each of the three possible input codes an equation exists that allows the analog output voltage to be calculated from the digital input code.

(i) Complementary binary

$$\text{output voltage} = \text{range} \times \frac{(65535\text{-code})}{(65536)} \quad \text{——— 1}$$

(ii) Offset binary

$$\text{output voltage} = -\text{range} \times \frac{(32767\text{-code})}{(32768)} \quad \text{——— 2}$$

(iii) 2's complement

The appropriate equation for 2's complement code is dependant on the value of the most significant bit (msb) of the digital input word.

(a) msb = 1

$$\text{output voltage} = -\frac{\text{range} \times (65535\text{-code})}{32768} \quad \text{——— 3}$$

(b) msb = 0

$$\text{output voltage} = \frac{\text{range} \times \text{code}}{32768} + \frac{\text{range}}{32768} - 4$$

where

range = 5 for +5V D/A output range
= 10 for +10V D/A output range

code = DECIMAL value of code written to the DAC channel.

The following examples are intended to clarify the use of these equations.

CASE 1

Output range = 0 to +10V range = 10v
Input code = 7FFFH complementary binary -> equation 1

$$\text{output voltage} = \frac{10 \times (65535 - 32767)}{(65536)} = +5V$$

CASE 2

Output range = ±10V range = 10V
input code = 7FFFH Offset binary -> equation 2

$$\text{output voltage} = \frac{-10 \times (32767 - 32767)}{32768} = 0V$$

CASE 3

Output range = ±5V range = 5
input code = FFFFH offset binary - > equation 2

$$\text{Output voltage} = \frac{-5 \times (32767 - 65535)}{32768} = +5V$$

CASE 4

Output range =±10V range = 10V
input code = 8000H 2's complement -> equation 3(a)

$$\text{Output voltage} = \frac{-10 \times (65535 - 32768)}{32768} = -9.999695V$$

CASE 5

Output range = +5V range = 5
input code = 7FFFH 2's complement - > equation 3(b)

$$\text{Output voltage} = \frac{5 \times 32767}{32768} + \frac{5}{32768} = +5V$$

The above answers can be checked against the output voltages given in tables 3.2, 3.3 and 3.4.

3.6.3 Output Range Selection

All eight analog outputs are factory set for a range of +/- 10V with offset binary coding. Other output ranges can be selected as described in section 2.2.2.

3.6.4 Output Check

Static checks of the analog outputs can be performed by loading the memory locations with output data words. The base address XXXXX0H is factory set to F00000H. (refer to address map figure 3.1) The ideal values for positive and negative full scale are shown in tables 3.1, 3.2 and 3.3.

A program which performs a simple dynamic check of the analog outputs is listed in section 5.3.

3.7 CALIBRATING ANALOG OUTPUTS

If the output range of the MPV955 is changed, gain and offset should be adjusted. The gain and offset errors of the board may be optimized for any one range by calibrating the unit on that range. The board is factory calibrated on the +/- 10V range. Before making the adjustments allow the board to reach thermal equilibrium (15-20 minutes under power with outputs running and enabled).

Three adjustments are necessary to calibrate the analog outputs, these are:

1. Calibrate voltage reference
2. Calibrate board for zero offset error (bipolar ranges only)
3. Calibrate board for zero gain error.

3.7.1 Voltage Reference Calibration

The start register must be accessed before the voltage references become operational. The reference voltages can now be checked and adjusted as necessary using a digital 6½ digit DVM.

The -10V reference can be calibrated at J1 pin 3 using RV17. (See Figure 2.3 for position). RV17 should be adjusted to give -10V ±0.005V at J1 Pin 3.

3.7.2 Offset and Gain Calibration

(i) Set up the board for all 8 channels output, disable timeout and give continuous internal trigger mode.

(ii) Load the start register with the first memory location data is intended to be taken from

(iii) Load the STOP register with the start register location + 7H
i.e. STOP ADDRESS = START ADDRESS + 7H

Table 3.4 Offset Calibration

CHANNEL	OFFSET ADJUSTMENT POTENTIOMETER
0	RV2
1	RV4
2	RV6
3	RV8
4	RV10
5	RV12
6	RV14
7	RV16

(iv) Load 8 consecutive data memory locations with FFFFH

(v) Access Start Register

Offset adjustment can now be made by examining the analog output on the P3 connector. Table 3.5 details the values required for offset calibration.

Table 3.4 details the potentiometers used for this adjustment.

Table 3.5 Offset Calibration

RANGE	CODE	DATA	ANALOG OUTPUT
0 to +5V	Complementary Binary	FFFFH	0V
0 to +10V	Complementary Binary	FFFFH	0V
-5V to +5V	Offset Binary	FFFFH	+5V
-5V to +5V	2's Complement	7FFFH	+5V
-10V to +10V	Offset Binary	FFFFH	+10V
-10V to +10V	2's Complement	7FFFH	+10

Fullscale gain error can now be adjusted.

Table 3.6 details the potentiometers used for this adjustment, and table 3.7 details the values required for gain calibration.

Table 3.6 Gain Calibration

CHANNEL	OFFSET ADJUSTMENT POTENTIOMETER
0	RV1
1	RV3
2	RV5
3	RV7
4	RV9
5	RV11
6	RV13
7	RV15

Table 3.7 Gain Calibration

RANGE	CODE	DATA	ANALOG OUTPUT	1 LSB
0 to +5V	Complementary Binary	0000H	+4.999924	76µV
0 to +10V	Complementary Binary	0000H	+9.999848	152µV
-5V to +5V	Offset Binary	0000H	-4.999848	152µV
-5V to +5V	Two's Complement	8000H	-4.999848	152µV
-10V to +10V	Offset Binary	0000H	-9.99969	310µV
-10V to +10V	Two's Complement	8000H	-9.99969	310µV

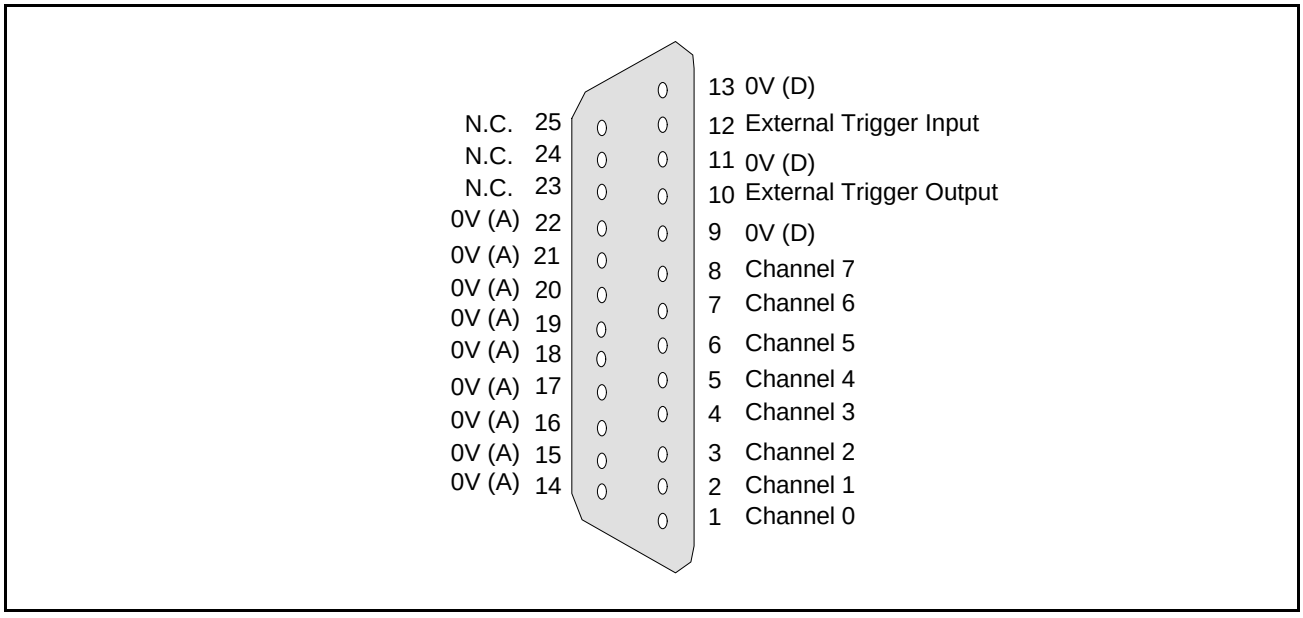
3.8 ANALOG OUTPUT CONNECTIONS

3.8.1 P3 Connector

The analog front panel connector, P3, is a 25 way "D" type socket (female). The mating connector should be a 25 way "D" type plug (male).

Use Cannon type DC25P/1A1N or equivalent.

Figure 3.13 "D" Type Connector Pinout



3.8.2 P2 Connector

The front panel connector signals (P3) are also available on the P2 connector for use in a customised back plane system where the user does not want connections to the front panel.

Table 3.8 J2/P2 Pin Assignments

PIN NUMBER	ROW A SIGNAL MNEMONIC	ROW B SIGNAL MNEMONIC	ROW C SIGNAL MNEMONIC
1	Channel 7 O/P	+5 Volts	User I/O
2	0V (A)	GND	User I/O
3	Channel 6 O/P	RESERVED	User I/O
4	0V (A)	A24	User I/O
5	Channel 5 O/P	A25	User I/O
6	0V (A)	A26	User I/O
7	Channel 4 O/P	A27	User I/O
8	0V (A)	A28	User I/O
9	Channel 3 O/P	A29	User I/O
10	0V (A)	A30	User I/O
11	Channel 2 O/P	A31	User I/O
12	0V (A)	GND	User I/O
13	Channel 1 O/P	+5Volts	User I/O
14	0V (A)	D16	User I/O
15	Channel 0 O/P	D17	User I/O
16	N/C	D18	User I/O
17	N/C	D19	User I/O
18	N/C	D20	User I/O
19	N/C	D21	User I/O
20	N/C	D22	User I/O
21	N/C	D23	User I/O
22	N/C	GND	User I/O
23	OV (D)	D24	User I/O
24	Ext. Trigger In	D25	User I/O

25	OV (D)	D26	User I/O
26	N/C	D27	User I/O
27	N/C	D28	User I/O
28	OV (D)	D29	User I/O
29	Ext. Trigger Out	D30	User I/O
30	OV (D)	D31	User I/O
31	N/C	GND	User I/O
32	N/C	+5Votls	User I/O

3.8.3 Factory Set Configuration

All boards are shipped from the factory ready for immediate use, however, they do have a number of user selectable options. These options and their factory set conditions, which have already been described, are summarized below:

BASE ADDRESS	F00000 H
ADDRESS/AM RESPONSE	Standard addressing no A.M. codes
Output RANGE	±10V
Output MODE	Bipolar
Output SELECTION	Unfiltered
CODING	Offset binary
MEMORY SIZE	16K x 16
Frequency Setting Resistor	6M8 ohms

Chapter 4 - FUNCTIONAL DESCRIPTION

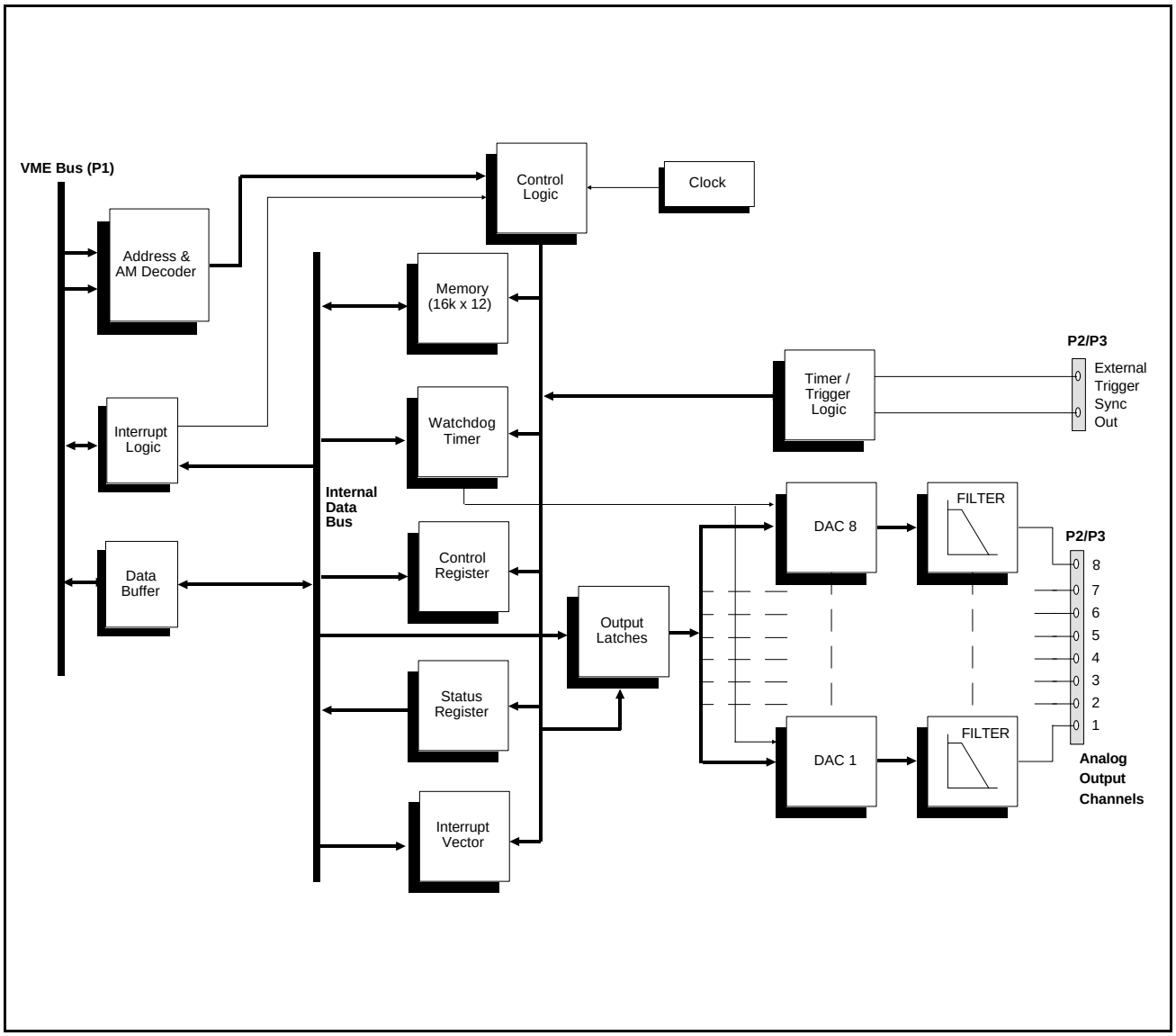
4.1 INTRODUCTION

This chapter is intended to explain the theory of operation of the MPV955 analog output board. It should be read while referring to the other chapters in this manual.

4.2 THEORY OF OPERATION

The MPV955 is an 8 channel analog voltage output board with 16 bit resolution which interfaces directly to the VMEbus. Figure 4.1 shows a block diagram of the board.

Figure 4.1 MPV955 Block Diagram



4.2.1 Address Decoder

The MPV955 address decoder decodes the address lines of the VMEbus to decide whether or not the CPU is accessing the board.

This is achieved in the following manner:

An 8-bit comparator compares the VMEbus address bits A23 to A16 with the jumper selectable value as specified by J25. If both bytes are the same then a "0" from the P=Q output indicates that the base address of the board is correct. This "0" is gated with a "0" from the correctly decoded address modifier (AM) PROM IC18 producing a "1" indicating that the board is being addressed.

4.2.2 Control/Status Register

The control register is a 8-bit latch/register with readback. This register controls the operation of the board, i.e. internal/external trigger, normal event trigger, continuous/one-shot mode, enable/disable timeout counter, and the number of channels selected. The status register is a PAL which is arranged in such a way as to inspect the 4 status bits at the same time as the 8 control bits are on the data bus. Due to the nature of the status register these bits are read-only. The organisation of these bits is shown in figure 4.2.

Figure 4.2 Organisation of Control/Status bits

D15	D8	D7	D0
	READ	READ/WRITE	

4.2.3 Start and Stop Address Registers

The start address register is a pointer to data memory and can hold a pointer to any location in the 16K data memory. The start address is loaded in one operation from the data bus and is held there until a change of the start address is required. This two-byte value is then loaded into 14 bit counter in LCA whose output is incremented from the start address value on each clock cycle.

Similarly, the stop address register can hold a pointer to any location in the 16K words of data memory, and is loaded into in one operation. This value is held until it is required to be changed. The value held is then compared to the incremented value of the start address. When the start address has been incremented until it equals the stop address, the output of a 14 bit comparator within the LCA indicates to the rest of the board that the stop address has been reached.

4.2.4 Interrupt Control Register

The interrupt control register supports READ/WRITE operations. and stores the interrupt priority level and the interrupt enable bits. When the MPV955 has been configured for an interrupt, the interrupt type generated by the board is compared with the type(s) of interrupts which it desired to service, i.e. interrupts on timeout, cycle finish or oversampling. During the interrupt acknowledge cycle the level of interrupt being serviced is indicated by address lines A01, A02 and A03.

This address information is compared to the generated interrupt level by a four-bit magnitude comparator and, if the service levels are the same, the interrupt control word held is enabled onto bits D00 to D07 of the data bus.

4.2.5 Rate Timer Register

The rate timer register is an 8-bit up counter which produces a trigger pulse to the DACS each time it reaches FFH. It then re-loads the rate timer value and begins counting again. This can only be accomplished when internal trigger is enabled by the control register. When this bit is disabled, the rate timer value is not loaded into the counter, so the only way to ensure trigger pulses for the DACS is via external trigger on the P3 connector.

4.2.6 Timeout Control Register

The timeout counter is an 8-bit up counter with a clock being supplied at approximately 0.77 Hz. The timeout can be controlled in two ways.

(i) The timeout value is contained in the least significant byte of the timeout register. This value represents the number of clock pulses to be counted. If another value is loaded into the timeout counter, this will mean a different period of time elapsing before time-out occurs.

(ii) Timeout can also be changed by altering the clock rate. Thus for a given timeout value, if the clock rate is altered the period before time-out will change accordingly.

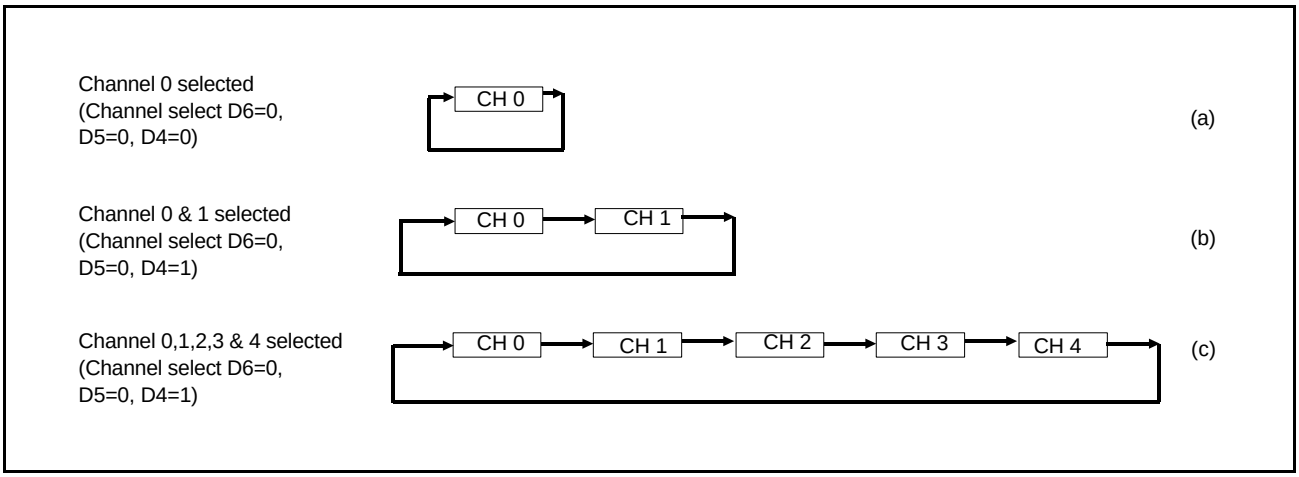
When the board is running, timeout can be prevented by regularly accessing the start register within the period before time-out. Every time a START command is given the value initially loaded into the timeout counter is re-loaded. This process can go on indefinitely.

If the watchdog timer facility is not required the clock can be disabled by simply writing a "1" into bit D3 of the control register. This action disables the clock input of IC13 ensuring that timeout will never occur.

4.2.7 Channel Selection

The number of channels selected for output is selected by the control register bits D04, D05 and D06. The values input to the control register are given in section 3.2.2. The manner of output is shown in figure 4.3.

Figure 4.3 Output Channel and Selection

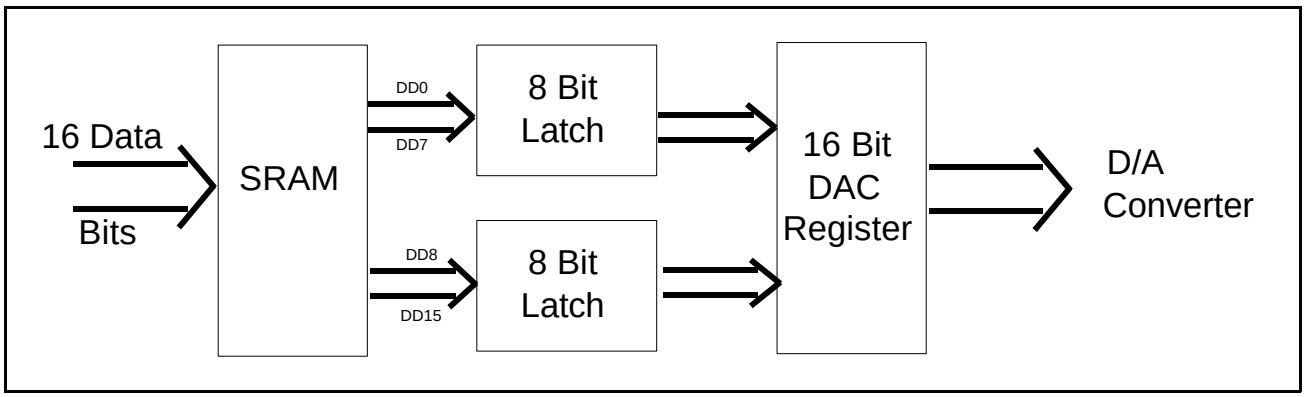


As can be seen from figure 4.3 the channels are selected in a cyclic manner always beginning at channel 0 and going sequentially through the channels. This continues until the maximum number of channels as defined by the channel select bits in the control register is reached. Once all channels have been selected the process will repeat if the board is in continuous mode. If not (i.e. one-shot mode),the board will stop.

4.2.8 Latch

The sixteen data bits are latched from the SRAMS into the DACs. The manner in which data is input to the DACs is as shown below in figure 4.4.

Figure 4.4 DAC Loading Configuration



First Byte D15..D8

2nd Byte D7..D0

4.2.9 Digital-to-Analog Converter

The D/A converter employed in this uses double buffering and a 16-bit multiplying D-to-A converter to achieve its 16-bit resolution. The D/A converter uses a R-2R resistor ladder network with binary weighted current switches which ensure a constant current in each ladder leg independent of the switch state.

One of the major problems in binary weighted current switches is differential non-linearity (unequal step size) or non-monotonicity due to the fact that currents associated with individual bits are more or less independent of each other. This is not a problem on the MPV955 since each DAC is guaranteed 16-bit monotonic. The convertor delivers an output voltage

Vout = -Vref x D x Rf where D = of all logic '1' currents

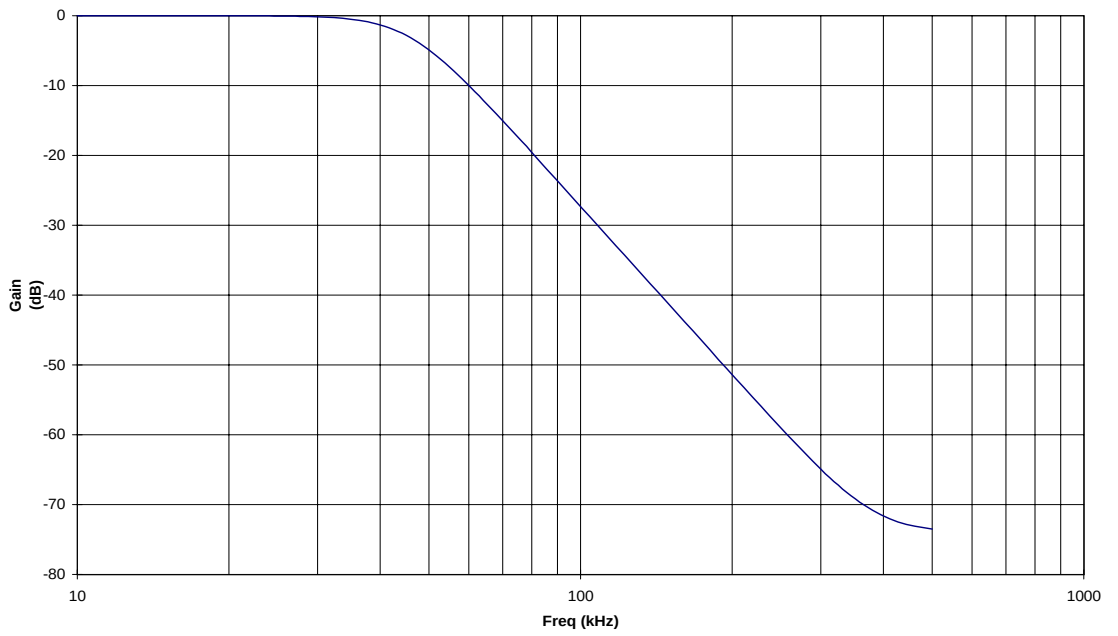
Thus for a positive output to be realised in unipolar mode the reference voltage should be negative.

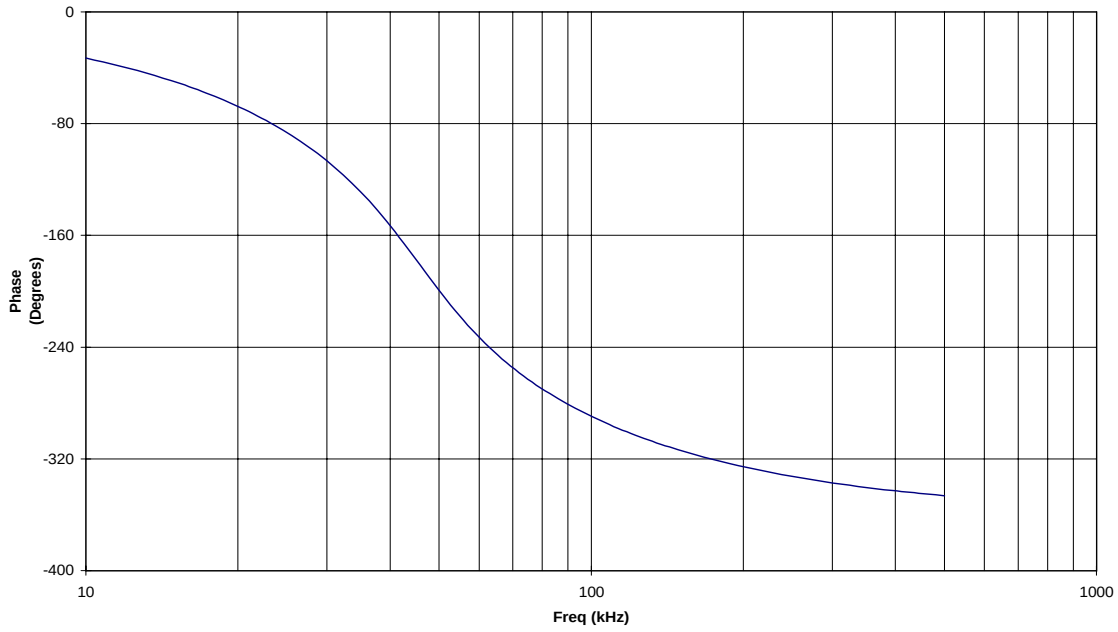
4.2.9.1 Reconstruction Filters

On each output channel of the board a reconstruction filter (LOW-PASS), to reduce possible glitches caused by the switches in the DAC having unequal turn-on and turn-off times. This can give rise to high frequency transients.

The filters are of the general biquad type and are configured to give low ripple (i.e. Butterworth) in the passband with a roll-off of -80dB/ decade. The typical gain and phase responses of these filters are as shown in figure 4.5.

Figure 4.5 Reconstruction Filter Characteristics





Chapter 5 - PROGRAMMING EXAMPLES

5.1 INTRODUCTION

This section is intended to assist the MPV955 user to quickly gain an understanding of the operation of the board using small, easily understood, applications programs.

The following three programs were written assuming that the board is still in its factory set condition and therefore the memory map in figure 3.1 applies. The programs are written in Motorola MC68020 assembly language on a MOTOROLA 133 system, and stored in RAM starting at location 20000H. Depending upon the type of assembler and monitor programs used, the following programs may require modification.

5.2 PROGRAM A

This program outputs a static value of voltages to the eight channels of the MPV955. As the board is setup to output the set values Area 1 has been selected. The voltages which appear on the outputs are as follows:

Channel 0	-10V
Channel 1	-7.5V
Channel 2	-5V
Channel 3	-2.5V
Channel 4	0V
Channel 5	+2.5V
Channel 6	+5V
Channel 7	+7.5V

```
            ORG         $20000;
DACDIS:     EQU         $F0800C;        DAC disable address
CONADD:     EQU         $F08001;        Control register address
STARTADD:   EQU         $F08002;        Start register address
STOPADD:    EQU         $F08004;        Stop register address
TIMEADD:    EQU         $F08008;        Rate timer register
GOADD:      EQU         $F0C000;        Start address register
MEMADD:     EQU         $F00000;        Base address of memory

CONBYTE:    EQU         #$78;           7 - enable 8 channels
                                        8 - disable watchdog timer
STARTWD:    EQU         #$0000;         First memory location
STOPWD:     EQU         #$0007;         Last memory location
GOWORD:     EQU         #$FFFF;         Value to start register
STEP:       EQU         #$0200;         Step size between channels,
                                        0200H = 2.5V
TICKRATE:   EQU         #$00CD;         Output sampling frequency
MINVAL:     EQU         #$F000;         First value to be output

            MOVE.W      #01,DACDIS      Disable all DAC outputs to 0V
            MOVE.W      #8,D1;          Initialise counter.
            LEA.L       MEMADD,A0;      Load base address of memory into
                                        A0
```

```
                MOVE.B      CONBYTE,CONADD;    Load control register with
                                               control byte
                MOVE.W      #TICKRATE,TIMEADD  Load sample frequency
                MOVE.W      STARTWD,STARTADD;  Load start register with start
                                               word
                MOVE.W      STOPWD,STOPADD;    Load stop register with stop word
                MOVE.W      MINVAL,D0;         Load the first value to be output
                                               into D0
LOOP            MOVE.W      D0,(A0)+;          Output value and increment memory
                ADD.W       STEP,D0;           Increment the output data by'STEP'
                SUBQ.W      1,D1;              Decrement counter
                BNE         LOOP;              All channels drive? No then LOOP
                MOVE.W      #00, DACDIS;       Enable all DAC outputs
                MOVE.W      GOWORD, GOADD;     Initiate conversions
                TRAP        14;                Return to monitor
```

5.3 PROGRAM B

This program outputs a "sawtooth" waveform in the range -10V to +7.5V on all channels. Again area 1 has been selected.

```
                ORG         $20000;
DACDIS:         EQU         $F0800C;           DAC disable address
CONADD:         EQU         $F08001;           Control register address
STARTADD:       EQU         $F08002;           Start register address
STOPADD:        EQU         $F08004;           Stop register address
TIMEADD:        EQU         $F08008:           Rate timer register
GOADD:          EQU         $F0C000;           Start address register
MEMADD:         EQU         $F00000;           Base address of memory

CONBYTE:        EQU         #$78;              7 - enable 8 channels
                                               8 - disable watchdog timer
STARTWD:        EQU         #$0000;            First memory location
STOPWD:         EQU         #$0007;            Last memory location
GOWORD:         EQU         #$FFFF;            Value to start register
STEP:           EQU         #$0200;            Step size between channels,
                                               0200H = 2.5V
TICKRATE:       EQU         #$00CD;            Output sampling frequency
MINVAL:         EQU         #$F000;            First value to be output

                MOVE.W      #01,DACDIS         Disable all DAC outputs to 0V
                MOVE.W      #8,D1;             Initialise counter.
                LEA.L       MEMADD,A0;         Load base address of memory into
                                               A0
                MOVE.B      CONBYTE,CONADD;    Load control register with
                                               control byte
```

```
                MOVE.W      #TICKRATE,TIMEADD; Load sample frequency
                MOVE.W      STARTWD,STARTADD;  Load start register with start
                                               word
                MOVE.W      STOPWD,STOPADD;    Load stop register with stop word
                MOVE.W      MINVAL,D0;         Load the first value to be output
                                               into D0
LOOP2:          MOVE.W      #8, D2;            Initialise loop 2 counter
LOOP1:          MOVE.W      D0, (A0)+;
                SUBQ.W      #1, D2;
                BNE         LOOP1;             8 memory location been filled
                                               with same value?
                ADD.W       STEP, D0;          Increment o/p data by 'STEP'
                SUBQ.W      #1, D1;            Decrement counter
                BNE         LOOP2;             Have all 8 blocks of memory been
                                               filled?
                MOVE.W      #00, DACDIS;       Enable all DAC outputs
                MOVE.W      GOWORD, GOADD;     Initiate conversions
                TRAP        #14;
```

5.4 PROGRAM C

This program implements the software swinging buffer when operating the MPV954 in the interrupt mode. Area 2 has been selected so that writes to the MPV954 start and stop address registers will not force a halt in output operation. Note, other control block writes will halt the MPV954. Processor interrupts on level IRQ6 must be enabled prior to program execution.

```
DACDIS          EQU         $F0800C         ;DAC disable address
CTRLADD         EQU         $F08000         ;Control/status register
STARTADD        EQU         $F08012         ;Start address register
STOPADD         EQU         $F08014         ;Stop address register
INTADD          EQU         $F08006         ;Interrupt control register
TIMEADD         EQU         $F08008         ;Rate timer register
STARTREG        EQU         $F0C000         ;Start register

BASERAM         EQU         $30000          ;Beginning of PROC memory
ENDRAM          EQU         $3FFFE          ;End of PROC memory

BUFFA           EQU         $E00000         ;Buffer A to swing with
BUFFB           EQU         $E10000         ;Buffer B

BUFFLTH         EQU         $0800           ;Buffer length in bytes
CTRLWRD         EQU         $0008           ;Continuous mode, normal, int
INTWRD          EQU         $2640           ;Enable interrupts (IR1, ID 40)
TICKRATE        EQU         $00CD           ;Output sampling frequency
```

```
            ORG         $25000
            MOVE.W      #01, DACDIS;        ;Disable all DAC outputs
            MOVE.W      #CTRLWRD,CTRLADD;
            MOVE.W      #INTWRD,INTADD;
            MOVE.W      #TICKRATE,TIMEADD;

BEGIN       LEA.L       BUFFA,A2            ;Start with buffer A
            MOVE.L      #ISR,$100           ;Load interrupt vector address
                                             with ISR
            LEA.L       BASERAM,A1          ;Load start address of PROC memory
            JSR         FIRST               ;Fill buffer A
            MOVE.W      #00,DACDIS          ;Enable all DAC Outputs
            JSR         SWING               ;Swing and fill
LOOP        JMP         LOOP                ;End of mainline code
```

*Output of MPV955 is controlled by swinging buffer mechanism under interrupt control

```
SWING       CMPA.L      #ENDRAM,A1
            BLE         VALID
            LEA.L       BASERAM,A1
VALID       LSR.W       #1,D0               ;Convert bytes to words for stop
            SUBQ.W      #1,D0               ;Less 1
            MOVE.W      D0, STOPPADD        ;Write stop address
            MOVE.W      #$FFFF,STARTREG     ;clear CYCFIN and start output
FIRST       MOVE.W      A2,D2               ;D0 to contain L.S. word of start
            LSR.W       #1,D2               ;Convert from bytes to words
            MOVE.W      D2,STARTADD         ;Write start adddress
BUFFER      MOVE.W      (A1)+(A2)+          ;Begin filling inactive buffer
            CMPA.L      #BUFFA+BUFFLTH,A2   ;Is buffer full
            BNE         CONT                ;Might be buffer B thats inactive
            MOVE.W      A2,D0               ;YES and buffer A was inactive
            LEA.L       BUFFB,A2            ;Set buffer B inactive
            RTS
CONT        CMPA.L      #BUFFB+BUFFLTH,A2   ;Is buffer full
            BNE         BUFFER              ;No then fill some more
            MOVE.W      A2,D0               ;YES and buffer B was inactive
            LEA.L       BUFFA,A2            ;Set buffer A inactive for next
            RTS

ISR         JSR         SWING               ;Interrupt service routine
            RTE

            END
```