STEPPER MOTOR CONTROL LSI

PPMC 101C/102A

DATA MANUAL

Sil-Walker, Inc.
ROYAL BLDG. 1-1, SHINJUKU 5-CHOME, SHINJUKU-KU, TOKYO 160 JAPAN
PHONE: (03)341-3651 FAX: (03)341-3974 TELEX: 0232-3398 CABLE: SILWALKER

TABLE OF CONTENTS

INTRODUCTION

PPMC101C/102A is a unique one-chip LSI specially designed to interface a stepper motor to an 8-bit micro computer with no additional hardware. PPMC101C/102A provides 8 kinds of different operations by the command of master CPU including acceleration/deceleration and constant speed operation.

Operating frequency and number of phase for stepper motor are programmable. Distribution signal to excitation driving circuit can also be programmble for selection of 2-phase or 1-2 phase excitation (2-3 phase excitation for 5-phase motor) for 3, 4 and 5 phase motor.

In addition, PPMC101C/102A provides five kinds of "limit" switch input. Complete function necessary to control stepper motor is included in one chip LSI. The PPMC101C/102A can be easily interfaced with a microcomputer system.

1. PPMC101C/102A SPECIFICATIONS

Operation Command

- * *Emergency Stop*
- * *Decelerating Stop*
- * *Single Step*
- * *Acceleration & Deceleration*
- * *Constant Speed Operation*
- * *To move to the "Limit"*

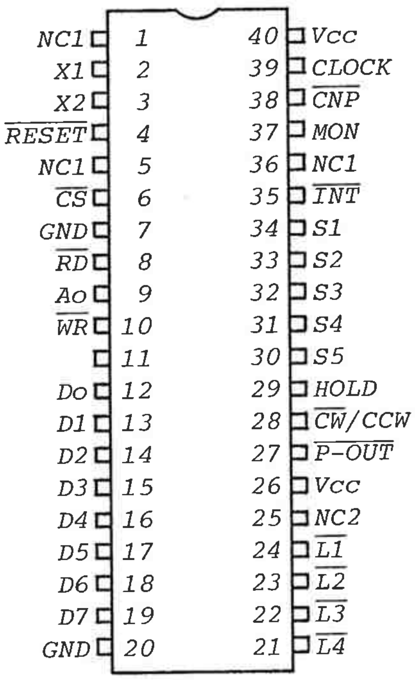
  (1) To move to the high speed limit
  (2) To move to the base point

Excitation Method

Motor	*Excitation*
3-phase	*2 phase* *1-2 phase*
4-phase	*2 phase* *1-2 phase*
5-phase	*2 phase* *2-3 phase*

Number of steps	:	16,777,216 max
Number of pulse for acceleration/deceleration	:	4 - 8,160
Maximum pulse rate	:	PPMC101C ... 5K pps (RA=20, fo=100KHz) PPMC102A ... 10K pps (RA=20, fo=200KHz)
Power supply	:	5V ± 5% 125mA max

2. TERMINAL ASSIGNMENT AND FUNCTIONS



(Top View)

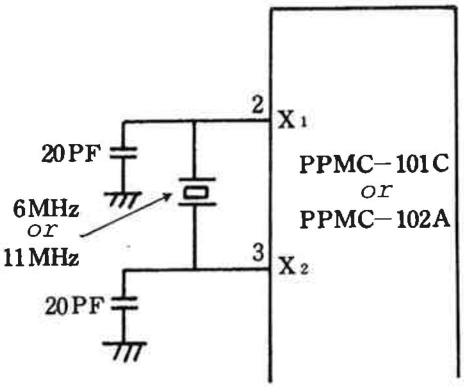
40 pin Dual-In-Line

Signal	Pin#	I/O	Description
X1, X2	2, 3	I	X-tal
$\overline{\text{RESET}}$	4	I	RESET input
$\overline{\text{CS}}$	6	I	Chip Select
$\overline{\text{RD}}$	8	I	READ strobe
Ao	9	I	Address 0
$\overline{\text{WR}}$	10	I	WRITE strobe
SYNC	11	O	Timing output
Do - D7	12-19	I/O	Data Bus 8-bit
$\overline{\text{L4}}$	21	I	Reverse high speed limit input
$\overline{\text{L3}}$	22	I	Forward " " " "
$\overline{\text{L2}}$	23	I	Reverse limit input
$\overline{\text{L1}}$	24	I	Forward " "
$\overline{\text{P-OUT}}$	27	O	Pulse output
CCW/$\overline{\text{CW}}$	28	O	Forward/Reverse status
			'0' = Forward
			'1' = Reverse
HOLD	29	O	Motor HOLD output
S5	30	O	Motor 5th phase output
S4	31	O	" 4th " "
S3	32	O	" 3rd " "
S2	33	O	" 2nd " "
S1	34	O	" 1st " "
$\overline{\text{INT}}$	35	O	Interrupt signal
MON	37	I	External control
			'0' = Motor ON
			'1' = Motor OFF
$\overline{\text{CNP}}$	38	I	Base point signal input
CLOCK	39	I	External clock input
Vcc	26,40	I	+5V DC
GND	7,20	I	0 V
NC1	1,5,36	I	pull up to Vcc with 3.3K ohm or open
NC2	25	O	OPEN

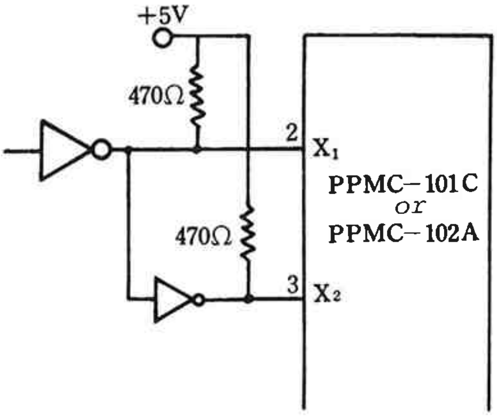
PIN DESCRIPTION

2-1 X1, X2

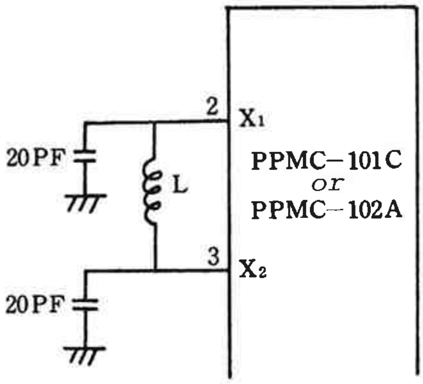
Inputs for a crystal. PPMC101C is normally operated with a 6 MHz crystal and PPMC102A with a 11 MHz crystal. You may also drive the clock inputs with an LC turned circuit or an external clock source (2 phases) as shown in Fig 2-1. 1 to 6 MHz for PPMC101C and 1 to 11 MHz for PPMC102A can also be used for driving frequency, but the operating speed slows down in accordance with the clock frequency.



X-tal Clock Driver



1 - 6 MHz Input Frequency External Clock Driver Circuit



LC Turned Circuit Clock Driver

Fig 2-1

PPMC-101C		PPMC-102A	
L=130μH	3 MHz	L=120μH	3.2 MHz
L= 40μH	5 MHz	L= 45μH	5.2 MHz

2-2 $\overline{\text{RESET}}$

Input used to reset status flip-flops and to set the program counter to zero. This pin should be connected to the RESET signal of a user's system. 50msec after the RESET signal rising edge the PPMC101C/102A is operative for initialization and operation command. The pulse width of the RESET signal must be no less than 2.5 μsec.

2-3 $\overline{\text{CS}}$

Input for chip select. To input the decorded signal from upper bits of ADDRESS. PPMC101C/102A is accessible at a low level 'O' on CS.

2-4 $\overline{\text{RD}}$

I/O read input which enables the master CPU to write data and status words from the PPMC101C/102A. The OUTPUT DATA BUS BUFFER or status register can be READ at a low level 'O' on $\overline{\text{CS}}$ and $\overline{\text{RD}}$.

2-5 $\overline{\text{WR}}$

I/O write input which enables the master CPU to write data and commands words to the PPMC101C/102A. Data on INPUT DATA BUS BUFFER can be written at a low level 'O' on $\overline{\text{CS}}$ and $\overline{\text{RD}}$.

2-6 AO

Address input used by the master processor to indicate whether the byte transfer is data or command as shown in the following table

Ao	$\overline{RD}$	$\overline{WR}$
0	Data Resister	Data Resister
1	Status Resister	Command Resister

Table 1

2-7 SYNC

Output signal which occurs once per execution of internal command in the PPMC101C/102A. It is also used to synchronize the single step operation. It is to be normally OPEN and used to check IC operation.

2-8 DO-D7

Tri-state, bidirectional DATA BUS BUFFER lines used to interface the PPMC101C/102A to an 8-bit master system data bus.

2-9 $\overline{\text{L1}}$ - $\overline{\text{L4}}$, $\overline{\text{CNP}}$

Inputs for external 'Limit' switches. Each signal is activated at a low level 'O'. Fig 2-2 shows the idea of 'Limit" switches.

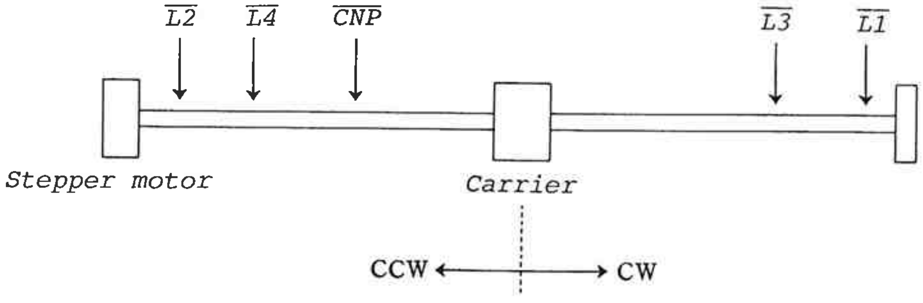


Fig 2-2

L1, L2 :

These switches are set at a maximum limit position where the carrier does not move further in CCW or CW. The motor will stop immediately when the carrier moves to these points regardless of the operation command. The carrier will no longer move further in the same direction even when it receives a command to move in the same direction. The carrier will start to move in the opposite direction only when it receives the command to move in reverse.

L3, L4 :

These switches must be positioned between L1 and L2, at a minimum distance corresponding to the number of deceleration steps. The stepper motor begins to decelerate at these positions (L3 or L4) in order to stop inside of L1 or L2.

$\overline{CNP}$:

Signal from $\overline{CNP}$ is used to establish a convenient reference point (Base point) with which the PPMC can monitor the position of the carrier. It does this by counting the number of steps in the data register. For example, in figure 2-2, in order to establish a convenient base point the command "move to base point" is used (see section 3-3-8). The motor will move the carrier to the position marked $\overline{CNP}$ and stop. Work can then proceed from this point.

2-10 $\overline{\text{P-OUT}}$, CCW/$\overline{\text{CW}}$

P-OUT is used for pulse output for other stepper motor driving modules without using PPMC phase output. It is useful for bipolar drive, switching drive and other special type of excitation method. It is recommended to use a decoder for CW or CCW pulse generation in combination with one-shot TIMER as shown in Fig 2-3 because driving module sometimes require 10 to 20 µsec pulse width. The signal from pulse output is a 5 µsec negative pulse and signal for direction is indicated by its level. In addition, these signals can be used for monitoring direction or number of pulses for rotation.

CCW/$\overline{\text{CW}}$ can be activated only when $\overline{\text{P-OUT}}$ is active.

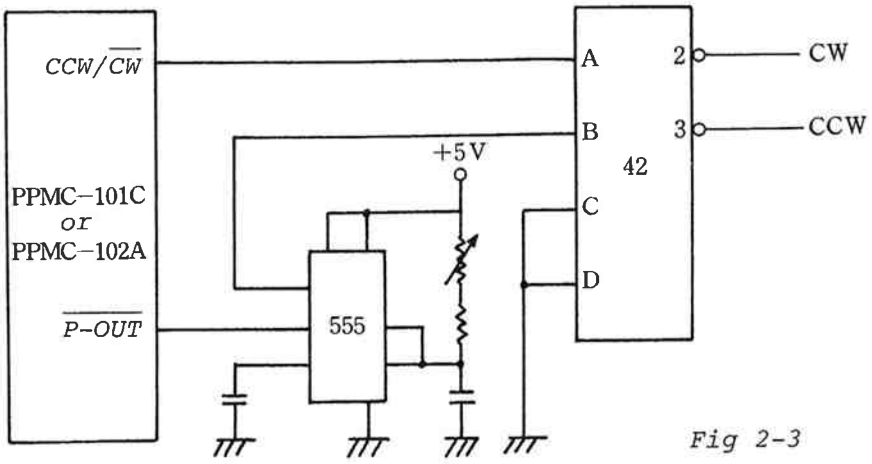


Fig 2-3

P-OUT is always available as well, in use of any type of phase output. (see page 13. motor code 01, 10, 11)

2-11 HOLD

HOLD output is high 3 msec after motor stops, but it is active only when bit 5 of the initialization command is set. (see page 13)

2-12 S1−S5

Provides signal for motor excitation drive.

Motor	*Control*
3 phase by	*S1 - S3*
4 " "	*S1 - S4*
5 " "	*S1 - S5*

Fig 2-4 shows the form of output.

The logic can be interchanged, positive to negative logic, and visa versa. Typical circuit is shown in Fig 2-5.

EXCITATION PULSE OUTPUT

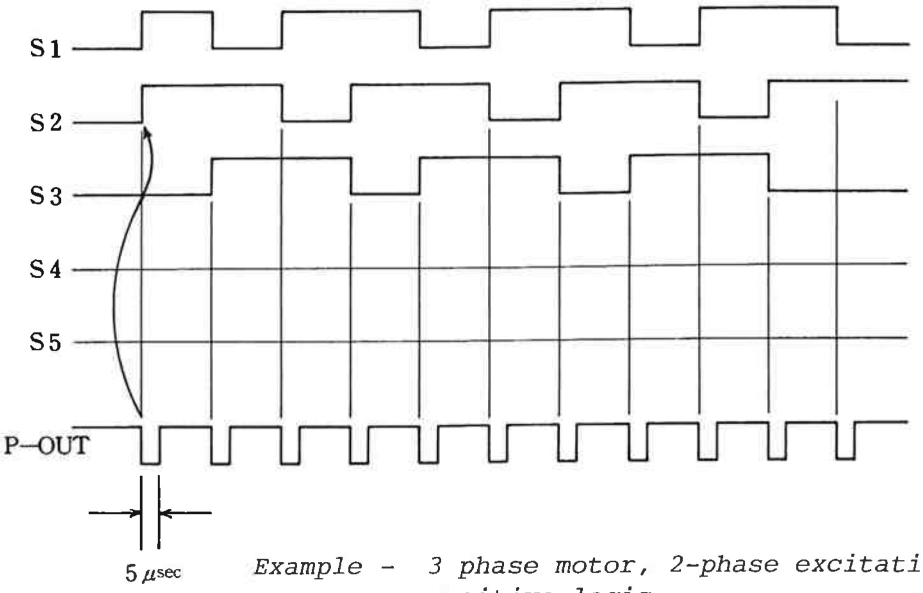
2-phase excitation

	S1	S2	S3
1	▨	▨	
2		▨	▨
3	▨		▨

(3-phase motor)

	S1	S2	S3	S4
1	▨	▨		
2		▨	▨	
3			▨	▨
4	▨			▨

(4-phase motor)



Example - 3 phase motor, 2-phase excitation positive logic

	S1	S2	S3	S4	S5
1	▨	▨			
2		▨	▨		
3			▨	▨	
4				▨	▨
5	▨				▨

(5-phase motor)

Fig 2-4

1-2 phase excitation

	S1	S2	S3
1	▨		
2	▨	▨	
3		▨	
4		▨	▨
5			▨
6	▨		▨

(3-phase motor)

	S1	S2	S3	S4
1	▨			
2	▨	▨		
3		▨		
4		▨	▨	
5			▨	
6			▨	▨
7				▨
8	▨			▨

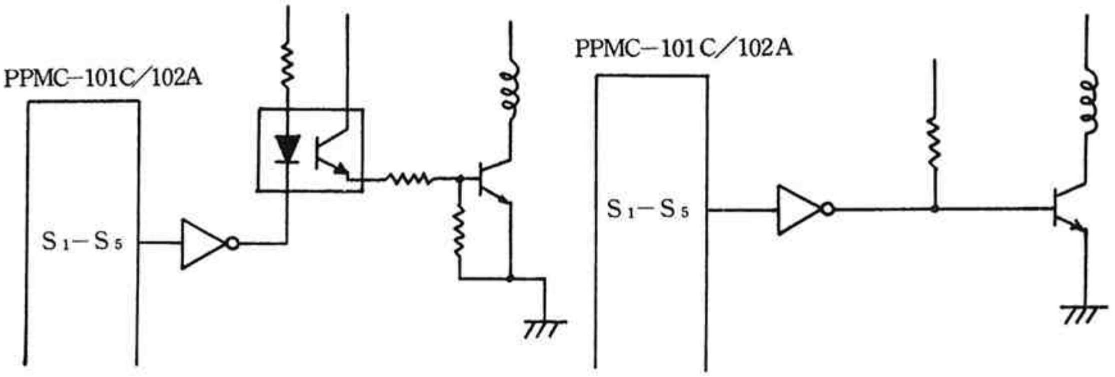
(4-phase motor)

2-3 phase excitation

	S1	S2	S3	S4	S5
1	▨	▨			
2	▨	▨	▨		
3		▨	▨		
4		▨	▨	▨	
5			▨	▨	
6			▨	▨	▨
7				▨	▨
8	▨			▨	▨
9	▨				▨
10	▨	▨			▨

(5-phase motor)

Output logic level can be switched by using positive or negative logic as shown in Fig. 2-5.



Output
Positive logic convention

Output
Negative logic convention

Fig. 2-5

2-13 $\overline{\text{INT}}$

Interrupt request is assertive 'O' when motor stops. $\overline{INT}$ can be cleared by reading the finish STATUS. This figure is not an open collector and OPEN COLLECTOR BUFFER is required as shown in Fig 2-6, when a multiple INTERUPT is expected.

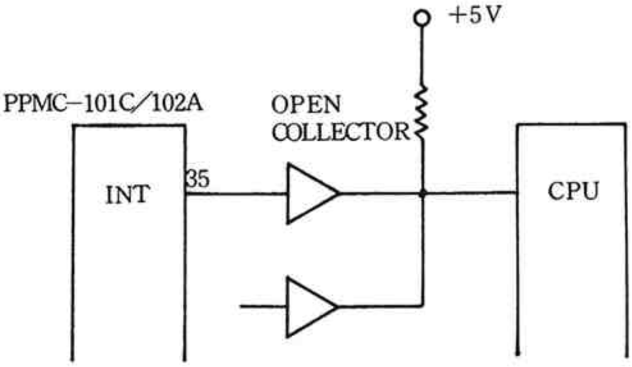


Fig 2-6

2-14 MON

When motor on input is 'O', PPMC does not output driving pulse. An example of an application is indicated in Fig 2-7, in which a thermal relay on the motor is used to protect overheating. MON input is ignored during operation of PPMC, and should be checked only before motor operation.

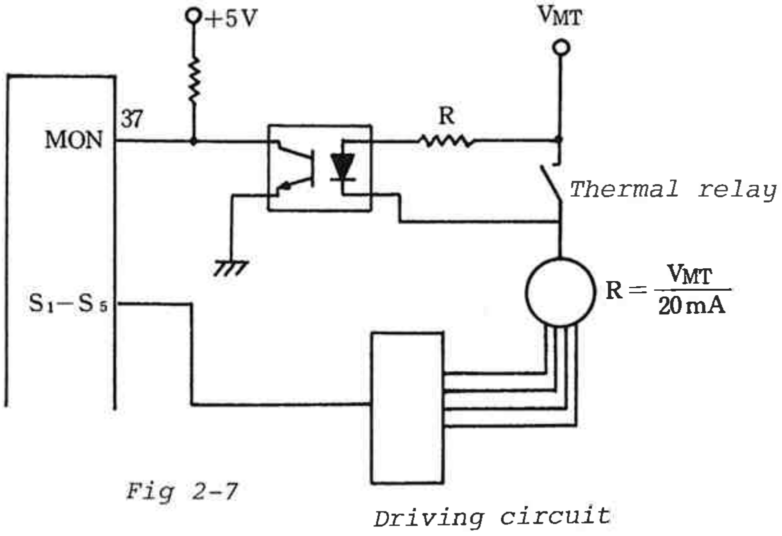


Fig 2-7

2-15 CLOCK-external clock

Basic signal to control speed of the stepper motor. The speed can be controlled between 400pps and 5K pps by the 100KHz clock input to PPMC101C and between 800pps and 10K pps by 200KHz clock input to PPMC102A. The clock frequency must be below 1/45 of X1, X2 clock. For example, when 6MHz X-tal is applied for X1 and X2, external clock input must be less than 133KHz (in case 11MHz is applied, external clock must be less than 244KHz). High level of the clock pulse must be more than 500 nsec.(250 nsec for 11MHz)

3. COMMUNICATION BETWEEN PPMC AND MASTER CPU

The communication between PPMC and master CPU consists of following 3 types of modes :

(1) Initialization

It designates type of motor, method of excitation, data for acceleration/deceleration and other parameters (see page 13 for details). After power 'ON', initialization is needed before operation command. Note : Some parameters cannot be changed once it is set. Re-initialization is not possible during operation.

(2) Operation Command

8 kinds of operation commands are available for stepper motor. The length of data to follow depends on the command.

(3) Register for PPMC

After completion of (2), master CPU reads the cause of operation finish, status of input/output terminal, and the number of remaining pulse.

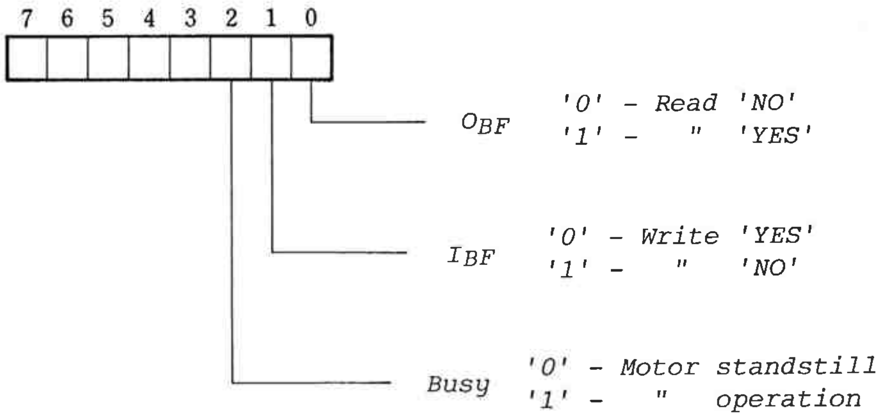
3–1 Register for PPMC

2 read only registers, and 2 write only registers are accessible to the uses.

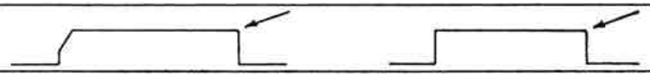
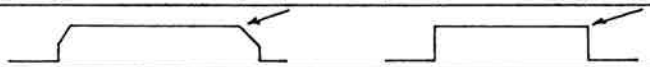
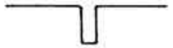
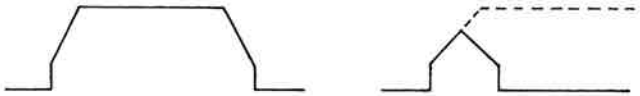
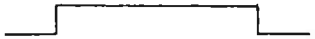
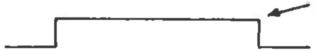
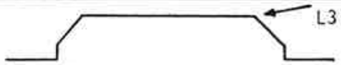
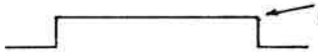
$\overline{CS}$	A_0	$\overline{RD}$	$\overline{WR}$	
1	×	×	×	DISABLE
0	0	0	1	READ DATA
0	1	0	1	READ STATUS
0	0	1	0	WRITE DATA
0	1	1	0	WRITE COMMAND

Table 3-1

3-1-1 Status Register



COMMAND TABLE

		COMMAND DATA		FUNCTION
INITIALIZATION		1 2 3 4 5	0 0 Self-starting pulse rate High speed pulse rate Accelerating/Decelerating pulse rate	
OPERATING COMMAND	Emergency Stop	1	0 1 0 0 0	
	Decelerating Stop	1	0 1 0 0 1	
	Single Step	1	0 1 0 1 0	 1 pulse
	Acceleration/ Deceleration	1 2 3 4	0 1 0 1 1 Number of operating step	
	Constant speed operation	1 2 3 4 5	0 1 1 0 0 Constant speed pulse rate Number of operating step	
	To move until the limit at constant speed	1 2	0 1 1 0 1 Constant speed pulse rate	 L1 or L2
	To move until high speed limit	1	0 1 1 1 0	 L3 or L4
	To move to the base point	1 2	0 1 1 1 1 Contant speed pulse rate	 CNP
READ REGISTER	Finish Data	1	1 0 0 0 0 0 0 0	To read data for reason of FINISH, etc. .... 1 byte
	Input signal	1	1 0 0 0 0 0 0 1	To read data for limit switch, etc. .... 1 byte
	Output signal	1	1 0 0 0 0 0 1 0	To read data for motor phase output and direction ... 1 byte
	Remaining step numbers	1	1 0 0 0 0 0 1 1	To read remaining number of steps .... 3 byte

3-1-1-1 OBF (Output Buffer Full)

This bit checks the status in order to read the data from PPMC. '0' indicates that there is no data in the buffer. It can only read the data when OBF is '1'.

3-1-1-2 IBF (Input Buffer Full)

This bit checks the status in order to write commands or input data to PPMC. '1' indicates that the data is full in the buffer and therefore, it is not possible to write new data. IBF must be '0' when you write data or give commands. If you were to write data at IBF '1' the former data would be erased.

3-1-1-3 BUSY (Motor Busy)

This flag outputs '1' during motor operation. It is only possible to input emergency stop and decelerating stop commands at that time. The IBF and BUSY bits must be checked before you input a command. This is '0' 2.5 μsec after INT output.

3-1-2 Read Register Data

Register data can be read out after checking OBF and input of READ REGISTER COMMAND.

3-1-3 Write Command

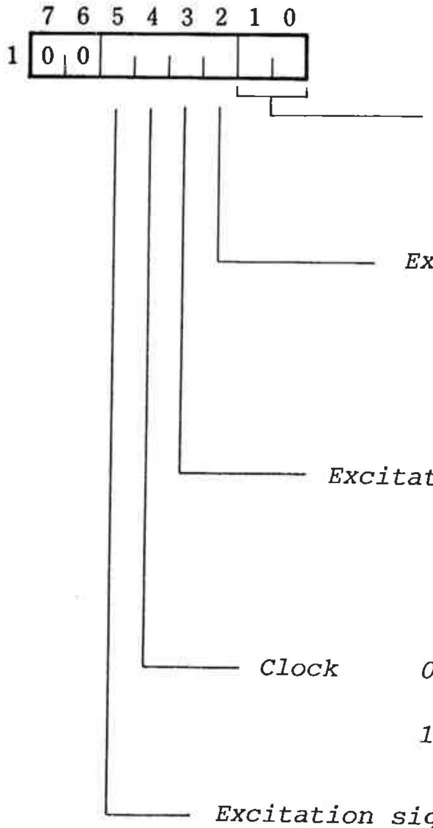
Before inputting initialization, operation command or read register command, check IBF and BUSY bit in the status register.

3-1-4 Write Data

Check IBF before writing data for pulse rate or number of steps. The order of input data must follow as indicated in command table (page 14). PPMC101C will start operating in accordance with the command 400 μsec and PPMC102A does 200 μsec after the data is written.

3-2 Initialization

Initialization Command



Motor Code
- 0 1 ... 3 phase motor
- 1 0 ... 4 " "
- 1 1 ... 5 " "

Excitation Method

0 ... 2 phase excitation

1 ... 1-2 phase excitation
(2-3 phase excitation for 5 phase motor)

Excitation output logic level (S1, S2, S3, S4, S5)

0 ... Negative logic convention

1 ... Positive " "

Clock

0 ... Internal clock (12.5KHz ... at 6MHz X1, X2)
(22.9KHz ... at 11MHz X1, X2)

1 ... External clock (Clock signal at pin#39)

Excitation signal switching output at motor standstill

0 ... Switching 'NO'

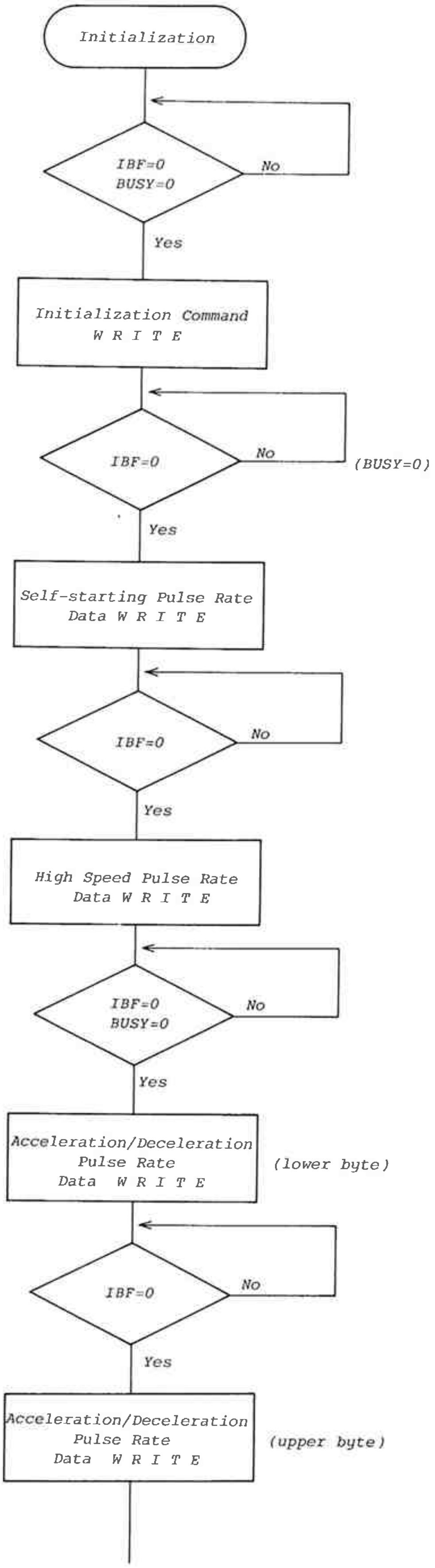
1 ... " 'YES'

Initialization Data

2		self-starting pulse rate .... RA max
3		high speed pulse rate .... RA min
4		Acc/Deceleration pulse rate .... lower byte
5		" " " .... upper byte

Initialization command to be input in the above order (1, 2, 3, 4, 5) after power 'ON'.

It can be shown in the following flow chart.


Initialization
IBF=0
BUSY=0
No
Yes
Initialization Command
WRITE
IBF=0
No
(BUSY=0)
Yes
Self-starting Pulse Rate
Data WRITE
IBF=0
No
Yes
High Speed Pulse Rate
Data WRITE
IBF=0
BUSY=0
No
Yes
Acceleration/Deceleration
Pulse Rate
Data WRITE
(lower byte)
IBF=0
No
Yes
Acceleration/Deceleration
Pulse Rate
Data WRITE
(upper byte)

3-2-1 Initialization Command

1) Motor code

The type of motor code used must match the spec of the motor.

2) Excitation method

The excitaion method used must match the spec of the motor.

3) Logic level of excitation output (S1, S2, S3, S4, S5)

In positive logic convention, the current will be flowing through the coil of the motor when output of PPMC is high. In negative logic convention, the current will be flowing through the coil of the motor when output of PPMC is low.

Fig 3-1 shows the circuits of logic level of excitation output.

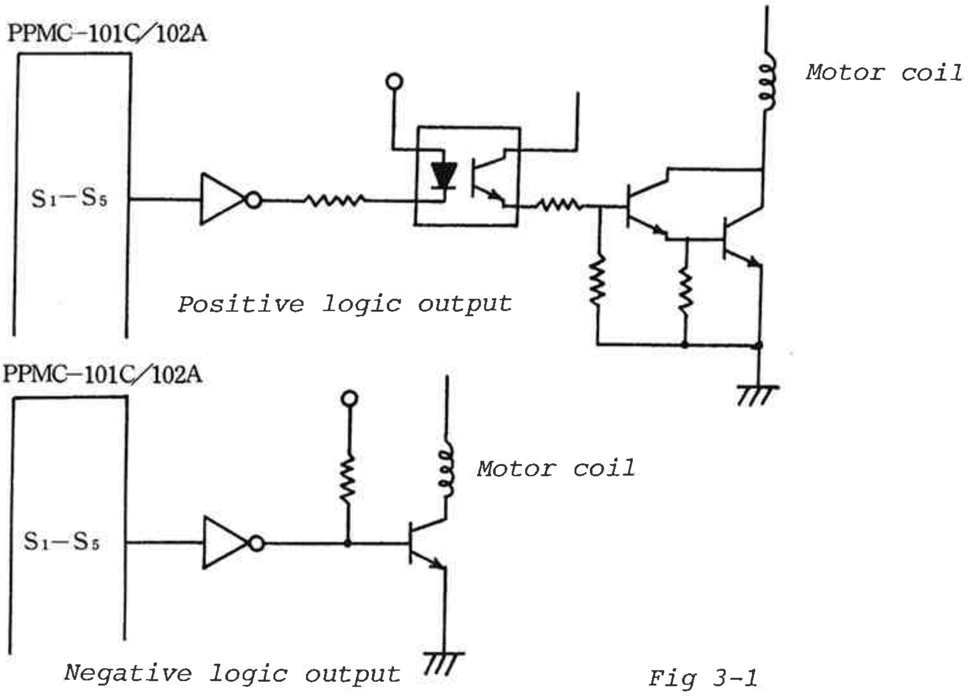


Fig 3-1

4) Clock

This is to select an internal or external basic clock for the motor. The internal clock will use frequency generated inside the PPMC. In the external clock mode the clock is provided by the input on pin#39.

It is possible for PPMC101C to control speed from 50pps to 600pps with the internal clock 12.5KHz and for PPMC102A to control speed from 100pps to 1,200pps. To control speed from 400pps to 5Kpps a 100KHz external clock should be used for PPMC101C. The external clock must be lower than 133KHz. In case of PPMC102A, speed can be controlled from 800pps to 10Kpps with 200KHz external clock which must be lower than 244KHz.

5) Excitation Signal Switching Output

Usually the current into the stepper motor remains at maximum current during stanstill. This maximum current which holds the motor can cause overheating. Bit 5 is used to prevent this type of problem by switching '1' or'0'. PPMC can switch the excitation output to minimize the excitation current. Switching frequency is about 2.2KHz for PPMC101C with a duty cycle of 30% and 4KHz for PPMC102A with a duty cycle of 35%.

About 100 msec after the output of a phase excitation signal, the motor will start operating from a standstill when "switching" is selected.

Type of motor code, excitation method and logic level of excitation pulse output cannot be changed once they are set after RESET, while clock, excitation pulse switching output and initialization data can be changed.

3-2-2 Initialization Data

3-2-2-1 Pulse Rate

PPMC applies the idea of pulse rate (RA) to decide speed of the stepper motor. The relationship between pps and RA is expressed in the following equations :

$$pps = \frac{fo}{RA}$$

fo : Basic clock frequency

RA : Pulse Rate

pps : Motor pulse per second

Basic clock is either a 12.5KHz for PPMC101C (22.9KHz for PPMC102A) clock generated inside PPMC or external clock applied to pin#39. Bit 4 (clock command bit) in initialization command is used to select either the internal/external clock.

PPS Table 3-1 shows various logical figure of RA and practical use.

RA fo		Logical (2 – 255)	Practical (20 – 255)	
Internal clock	12.5 KHz	49 Hz – 6.25KHz	49 Hz – 625Hz	(PPMC101C)
	22.9 KHz	89 Hz –11.4 KHz	89 Hz – 1,140Hz	(PPMC102A)
External clock	100 KHz	392 Hz – 50 KHz	392 Hz – 5 KHz	(PPMC101C)
	200 KHz	784 Hz –100 KHz	784 Hz – 10 KHz	(PPMC102A)

Table 3-1

The appropriate number of pulse for acceleration/deceleration should be decided by the customer's experience, because it depends primarily on type of motor, inertia moment of load, etc. In case of large inertia moment of load, a large number of pulse for acceleration/deceleration should be selected for slow operation. PPMC can be adapted quite well to the majority of the load. With 2 byte to store the number of pulse needed for acceleration/ deceleration, 4 - 8,160 steps can be set.

Some motors have a sympathetic point where there is no torque at certain frequency as shown in Fig 3-2. In such cases, the motor has to be started with a speed lower than the sympathetic point in order to fly into a higher speed area. To minimize the time to stay on the sympathetic point, higher speed for acceleration/deceleration must be applied. It is recommended that a damper should be used to increase the inertia moment if the motor goes into the sympathetic point with a small load.

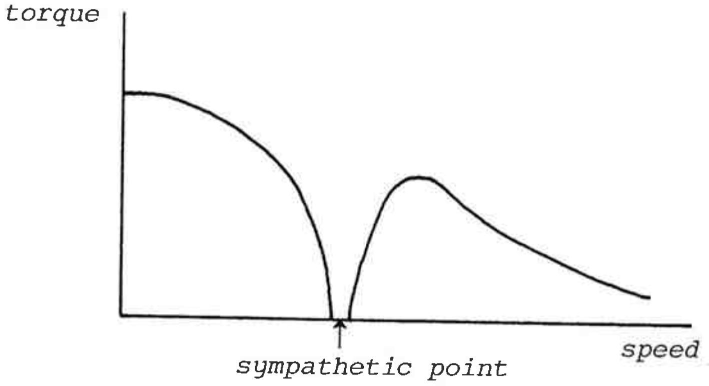


Fig 3-2

Initialization Data

Stepper motor has two types of operation as follows :

(A) Constant speed operation at lower speed of self-starting frequency.

(B) Ramp up/down operation.

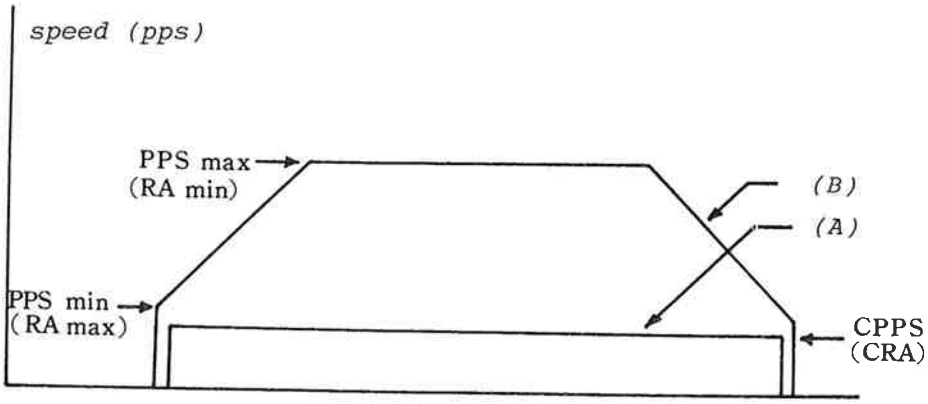


Fig 3-3

Self-starting frequency, parameters for high speed operation and acceleration/deceleration depend on the type of motor, excitation method, inertia or nature of load, etc. Relation among those are indicated below :

$$255 \geq CRA \geq RA\ max > RA\ min \geq 20$$

Pulse Rate		Symbol
At constant speed operation	.....	CRA
At self-starting	.....	RA max
At high speed operation	.....	RA min

CRA and RA max are to be set as large as possible within the above limit. The hardware limits the external clock rate to a maximum of 133KHz for PPMC101C (244KHz for PPMC102A). For example with 100KHz external clock for PPMC101C and 200KHz for PPMC102A the above relation can be converted into pps (pulse per second) as following.

PPMC101C	$392Hz \leq CPPS \leq PPS\ min < PPS\ max \leq 5KHz$
PPMC102A	$784Hz \leq CPPS \leq PPS\ min < PPS\ max \leq 10KHz$

In conclusing, with a 100KHz external clock for PPMC101C or 200KHz for PPMC102A, the stepper motor can be controlled from 400pps to 5Kpps for PPMC101C or 800pps to 10Kpps for PPMC102A. For lower speed operation, external clock frequency should be slowed down accordingly.

3-2-2-3 Aberration of motor speed

There are two major sources that cause the motor speed to deviate from the theoretical value (see Fig 3-4). The first source of error derives from the execution time of routine that outputs the excitation. A 50 µsec overhead time is needed in addition to the delay timing for the pulse output. Therefore for slow speed operations, the 50 µsec error is insignificant. The % error of the output speed will increase with an increase in motor speed.

The second source of error is the non-synchronization of the basic clock and the internal timer. A randam error corresponding to ±1 basic clock pulse in the timer counter is possible. Note the percentage error will be larger at a slower clock rate. For example at a clock rate of 20Hz, the randam error is ±5%, which may be acceptable in practical application.

Following is the curve that shows the differnce between theoretical value and practical speed at a basic clock frequency of 100KHz (PPMC101C) and 200KHz (PPMC102A). The graph shows that at RA=15, the motor speed is 5Kpps for PPMC101C and 10KHz for PPMC102A.

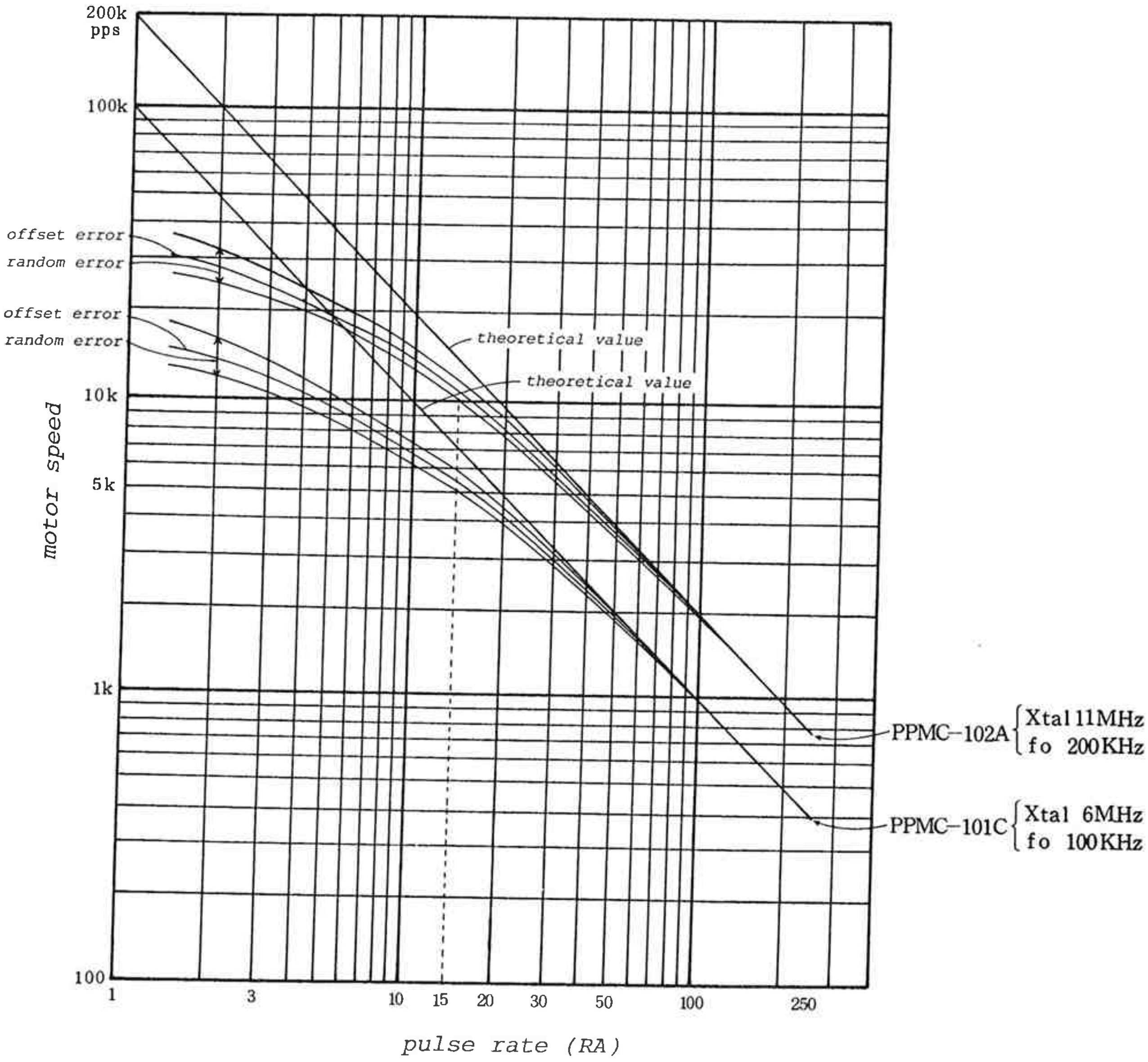
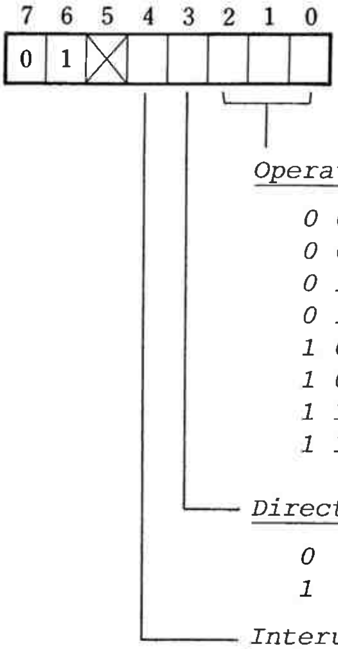


Fig 3-4

3-3 Operation Command



Operation command code

0 0 0 ..... Emergency stop
0 0 1 ..... Decelerating stop
0 1 0 ..... Single step
0 1 1 ..... Acceleration/Deceleration
1 0 0 ..... Constant speed operation
1 0 1 ..... To move until the limit
1 1 0 ..... To move until the high speed limit
1 1 1 ..... To move until the base point

Direction for rotation of motor

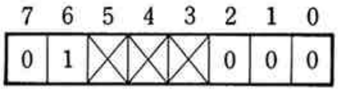
0 ..... C W
1 ..... C C W

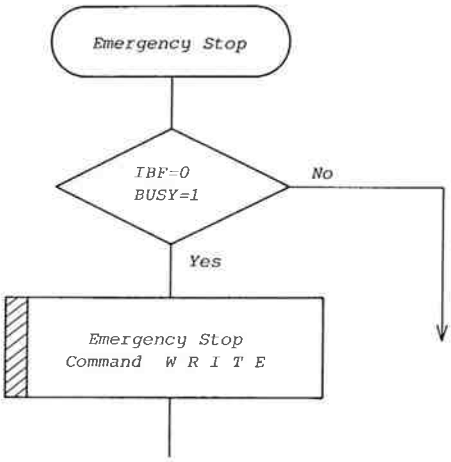
Interupt mask for FINISH signal

0 ..... Enable
1 ..... Disable

3-3-1 Emergency Stop

To stop rotation of motor instantaneously by inhibiting excitation output during any type of operation, whether it be acceleration/deceleration or constant speed operation. In high speed operation, the phase output stops instantaneously, but the motor will run off with inertia. Therefore, position data is no longer valid. It is necessary to reestablish the base point. During constant speed operation at self-starting frequency, motor can stop instantaneously and restarting is possible from that point by reading the number of operating pulse with the READ REGISTER COMMAND. Emergency stop requires only 1 byte operation command, and no data is necessary. Check whether the condition IBF=0, and BUSY=1 is satified before writing the emergency stop command as shown below.

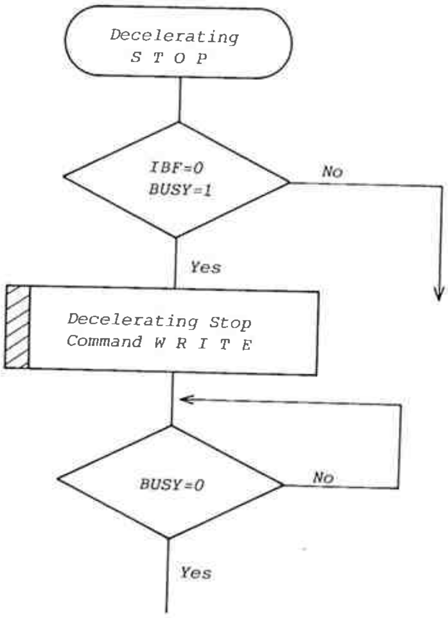




3-3-2 Decelerating Stop

When the decelerating stop command is input during acceleration/ deceleration, the motor will decelerate to stop. The motor will stop instantaneously during constant speed operation at self-starting frequency, the remaining number of pulse can be read, by the READ REGISTER COMMAND and the motor can be re-started from where it stopped. The bits for direction of motor rotation and FINISH INTERRUPT become assertive when the motor stops. Refer to the following flow chart for proper sequence of operation.

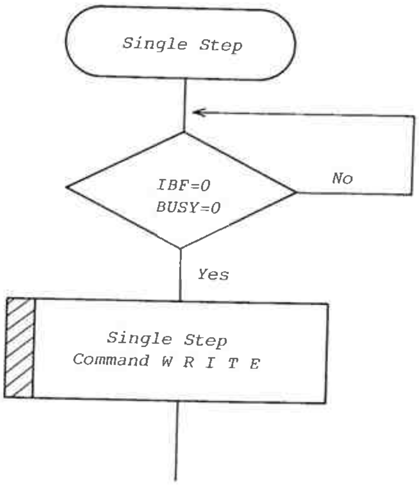
7	6	5	4	3	2	1	0
0	1	☒	☒	☒	0	0	1



3-3-3 Single Step

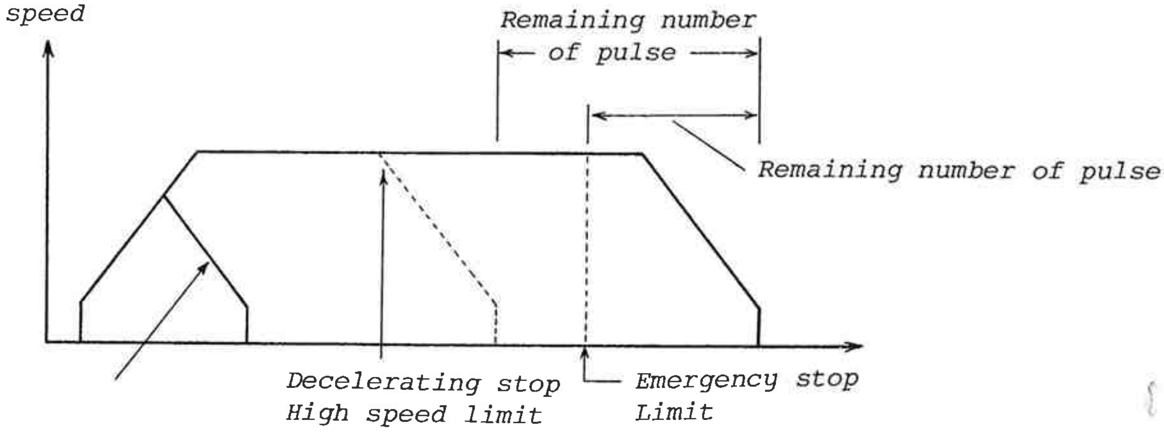
This is the command to move the stepper motor at a single step. It is useable when the master CPU needs to find out its position by itself. When this command is released continuously, timing must be controlled by the master CPU. All command modes are effective, this command consists of a single byte. No other data is necessary. Refer to the following flow chart for proper sequence of operation.

7	6	5	4	3	2	1	0
0	1	☒			0	1	0



3-3-4 Acceleration/Deceleration

This command for acceleration/deceleration in accordance with the data at the time of initialization. In addition to the command itself this operation requires 3 bytes of data, which store the total number of pulses to be output. For triangle operation, total number of pulse must be smaller than two times the acceleration/deceleration pulse number. The limiting switch input L3, L4 can be used to trigger the deceleration (see Fig 2-2) and L1 and L2 can be used to stop the motor. Note that irrelevant signals from L1-L4 will be ignored. For example in Fig 2-2, if the carrier is moving CW, the signals from L2 and L4 will be ignored.



The number of pulse (step number requested for operation - 1) can be got with 3 bytes.
FFFFFF (Hexa decimal) input moves 16,777,216 steps which is the maximum number of steps to move at a time.

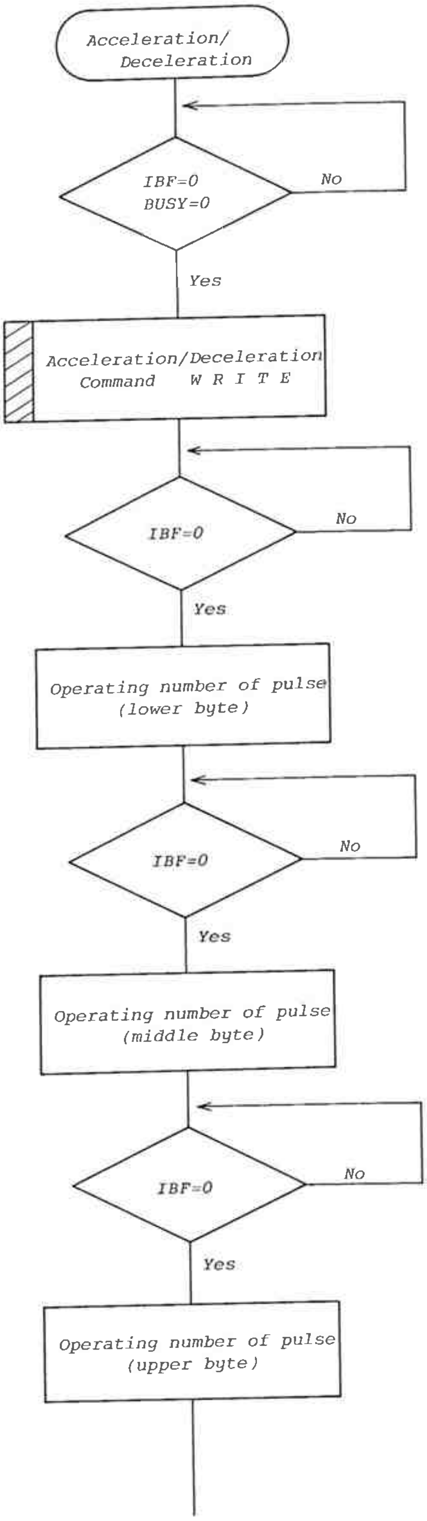
- Example -

To move 1,000 steps, 1,000 - 1 = 999 should be converted into Hexa decimal (0003E7) for input. Data must be input from the lower byte.

7	6	5	4	3	2	1	0
0	1	☒			0	1	1

Operating number of pulse

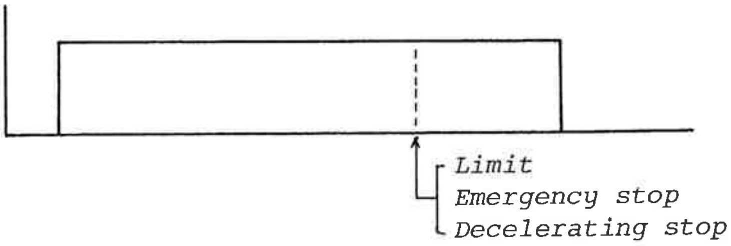
	7	6	5	4	3	2	1	0	
(E7)	1	1	1	0	0	1	1	1	*(lower byte)*
(03)	0	0	0	0	0	0	1	1	*(middle byte)*
(00)	0	0	0	0	0	0	0	0	*(upper byte)*



Acceleration/ Deceleration
IBF=0 BUSY=0
No
Yes
Acceleration/Deceleration Command W R I T E
IBF=0
No
Yes
Operating number of pulse (lower byte)
IBF=0
No
Yes
Operating number of pulse (middle byte)
IBF=0
No
Yes
Operating number of pulse (upper byte)

3-3-5 Constant Speed Operation

This command requires one byte of data for constant speed pulse rate as well as 3 bytes of data for the total number of pulse. The proper sequence of execution is shown in the following flow chart. The command causes the motor to rotate at a constant speed up to the designated distance. The speed is set by the pulse rate data, which has to be within the self-starting frequency of the motor. L1-L4 limit switch input can be used to decelerate and stop the motor. The READ REGISTER COMMAND can be used to readout the remaining number of pulse and the cause for stopping.

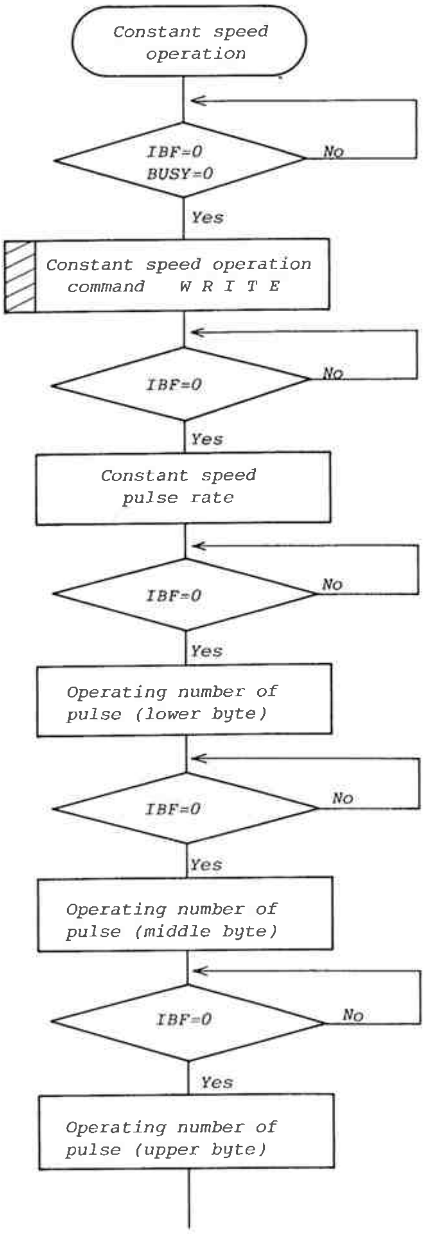


Refer to the following flow chart for proper sequence of operation.

7	6	5	4	3	2	1	0
0	1	☒			1	0	0

Operating number of pulse

	7	6	5	4	3	2	1	0
constant speed pulse rate								
(lower byte)								
(middle byte)								
(upper byte)								

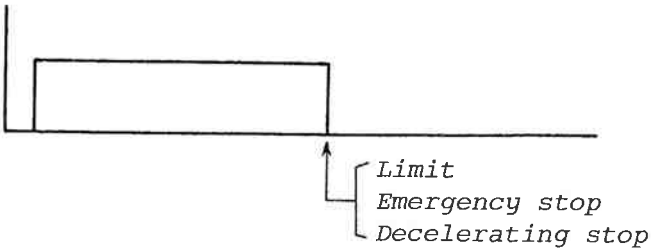

Constant speed operation
IBF=0
BUSY=0
No
Yes
Constant speed operation command WRITE
IBF=0
No
Yes
Constant speed pulse rate
IBF=0
No
Yes
Operating number of pulse (lower byte)
IBF=0
No
Yes
Operating number of pulse (middle byte)
IBF=0
No
Yes
Operating number of pulse (upper byte)

3-3-6 To Move At Constant Speed Until Limit Switch

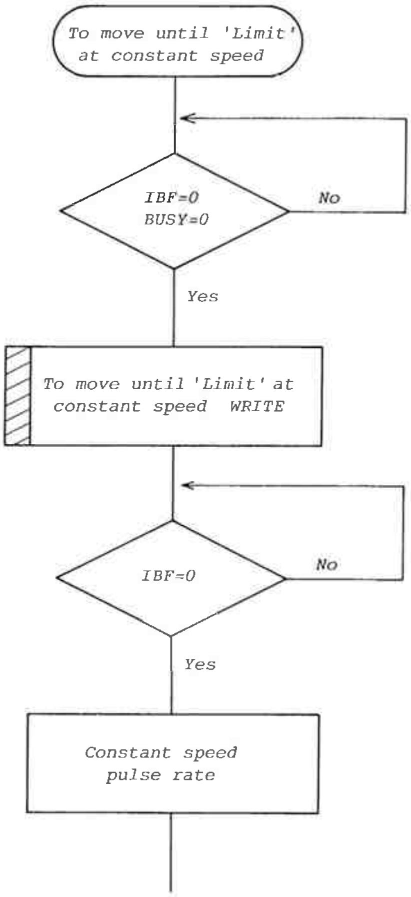
This mode of operation is similar to the previous one except that the data indicating the number of pulse is omitted. The motor will keep on moving until a signal from a limit switch (L1 or L2) is received. The signal from irrelevant L1 or L4 is ignored. Limit input in the same operating direction means as follows :

Operation	*Input*	
C W	*L1*	*limit input*
C C W	*L2*	*" "*

During a CW operation, L2 input will be ignored.



This command is normally used right after POWER ON or to re-start after motor run off.


To move until 'Limit' at constant speed
IBF=0
BUSY=0
No
Yes
To move until 'Limit' at constant speed WRITE
IBF=0
No
Yes
Constant speed pulse rate

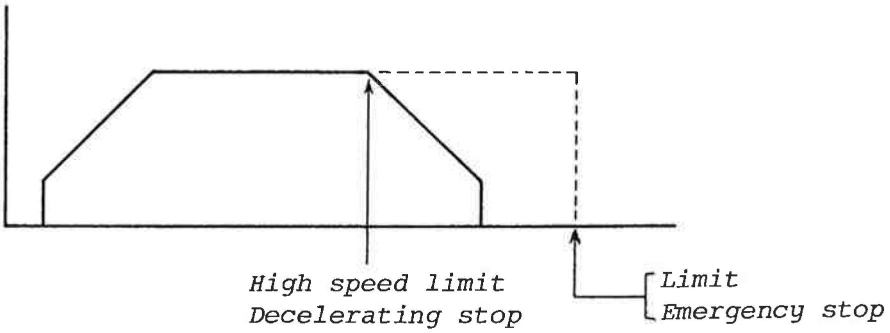
7	6	5	4	3	2	1	0
0	1	☒			1	0	1

7	6	5	4	3	2	1	0

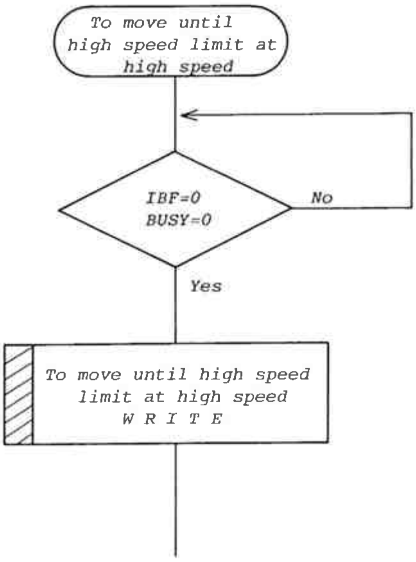
constant speed pulse rate

3-3-7 To Move At High Speed Until High Speed Operation Limit

This is the command to accelerate for high speed operation in accordance with the data at the time of initialization. Under this command the motor will rotate at high speed until a limiting signal is received. Then it will decelerate to stop according to the number of decelerating pulse. Once deceleration begins, it continues to decelerate even if limit input turns out '1'. Limit switch L1 for CW and L2 for CCW rotation can force the motor to stop, but inertia may cause the motor to overrun the desired stopping point.



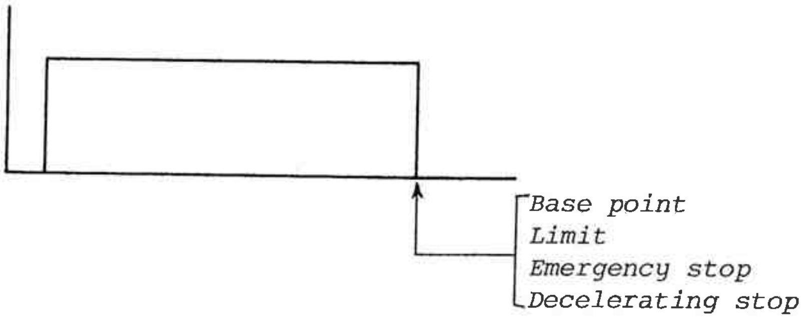
This command is also used right after POWER ON or to re-start motor after run-off. The decision whether to use this command or the command 'to move at constant speed until limit switch' (3-3-6) depends on the distance, time, accuracy of position, etc.


To move until high speed limit at high speed
IBF=0
BUSY=0
No
Yes
To move until high speed limit at high speed
W R I T E

7	6	5	4	3	2	1	0
0	1	☒			1	1	0

3-3-8 To Move To Base Point

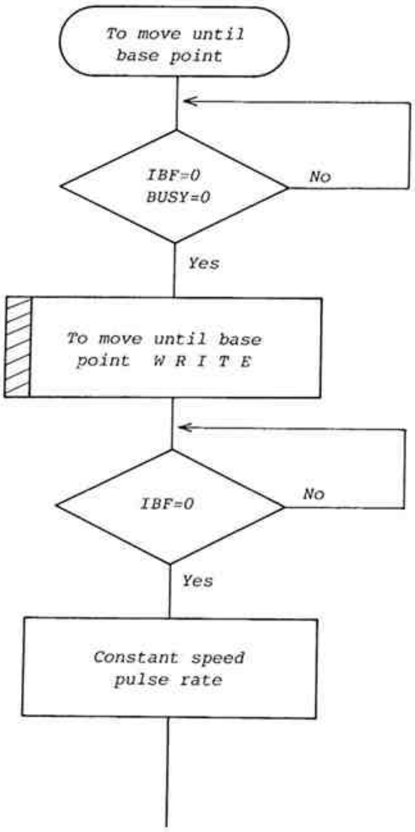
In this mode of operation, the motor rotates at constant speed until a $\overline{CNP}$ signal is detected. The motor can also be stopped by the relevant limit input L1-L4, emergency stop or decelerating stop.



7	6	5	4	3	2	1	0
0	1	☒			1	1	1

7	6	5	4	3	2	1	0

constant speed pulse rate



3-4 READ REGISTER

READ REGISTER COMMAND is used to read three kinds of status and a 3 byte data during standstill of motor. The proper format of the command is indicated below :

7	6	5	4	3	2	1	0
1	0	0	0	0	0		

Register code

0 0 ..... FINISH status

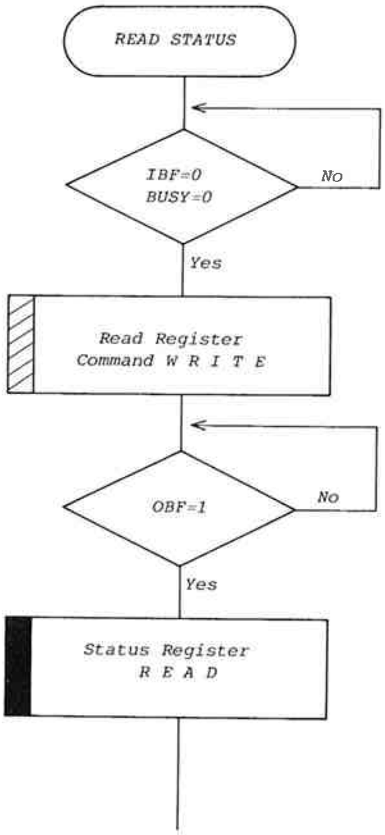
0 1 ..... Input signal status

1 0 ..... Output signal status

1 1 ..... Remaining number of pulse

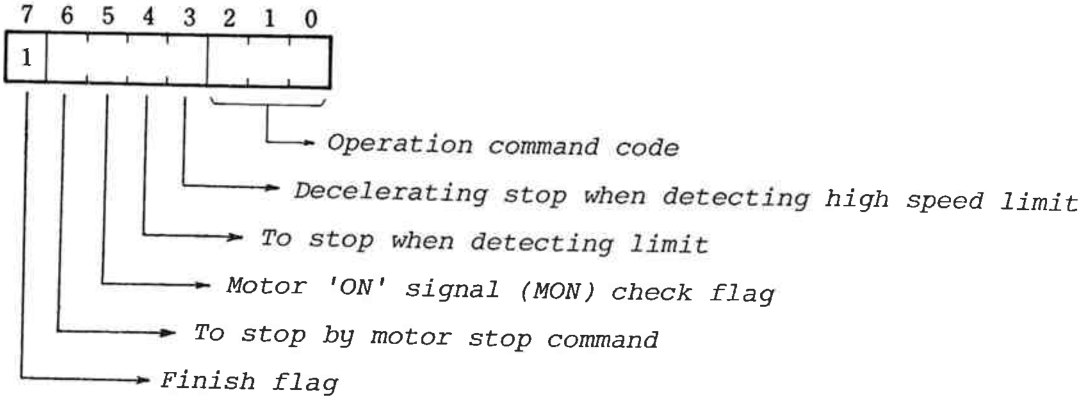
3-4-1 READ FINISH STATUS

The proper sequence to read the various status is shown in the following flow chart.



3-4-1-1 FINISH STATUS

The finish status register contains the following information :


7 6 5 4 3 2 1 0
1
Operation command code
Decelerating stop when detecting high speed limit
To stop when detecting limit
Motor 'ON' signal (MON) check flag
To stop by motor stop command
Finish flag

The lower 3 bytes contain the operation command code.

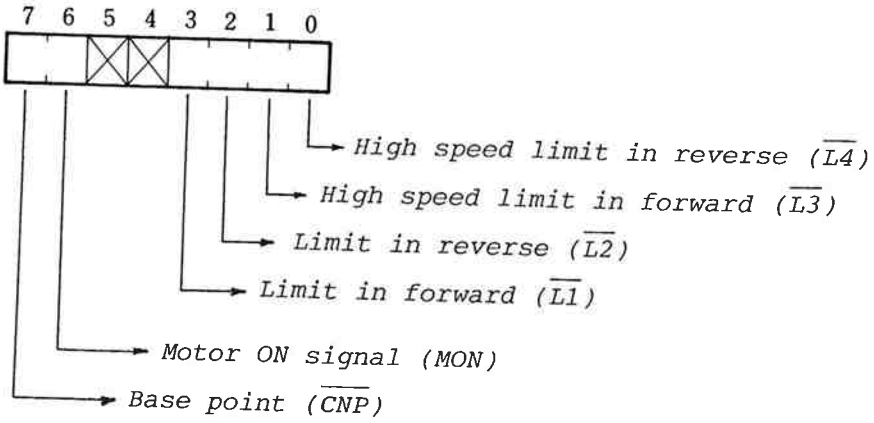
Bit 3 or 4 goes up to '1' after high speed limit ($\overline{L3}$, $\overline{L4}$) or limit (L1, L2) is input for motor stop. Bit 5 shows '1' when motor cannot operate with 'motor on signal' (MON) = 0.

A '1' in bit 6 indicates either an emergency stop or decelerating stop. When all number of pulses for acceleration/deceleration and constant speed operation are completely consumed to stop, all bits from 3 to 6 turns to '0'.

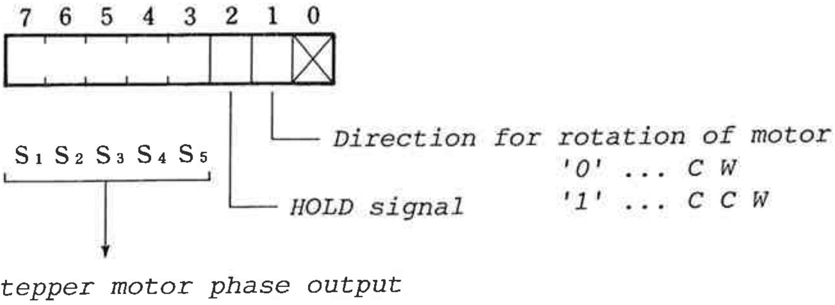
In the absence of finish interrupt mask, $\overline{INT}$ becomes assertive at the end of the operation, $\overline{INT}$ signal can be cleared to '1' by reading finish status and released.

3-4-1-2 INPUT signal

The input signal register reflects the state of various inputs shown below at the point where the motor stops.


7 6 5 4 3 2 1 0
High speed limit in reverse ($\overline{L4}$)
High speed limit in forward ($\overline{L3}$)
Limit in reverse ($\overline{L2}$)
Limit in forward ($\overline{L1}$)
Motor ON signal (MON)
Base point ($\overline{CNP}$)

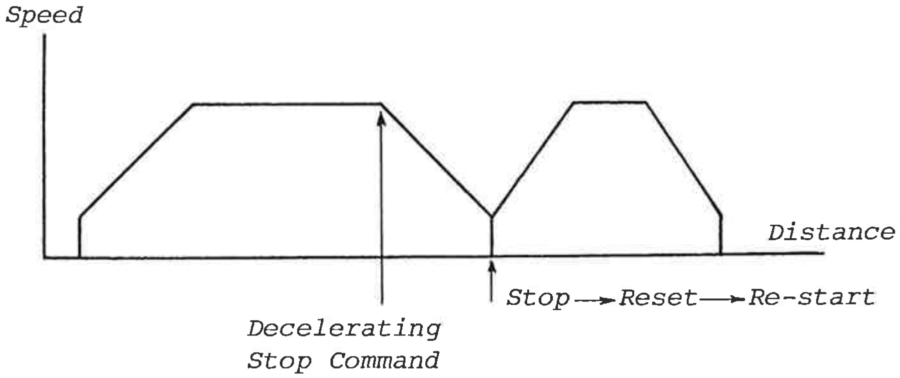
3-4-1-3 OUTPUT Signal



Phase outputs for stepper motor can be checked by bit 3 to 7

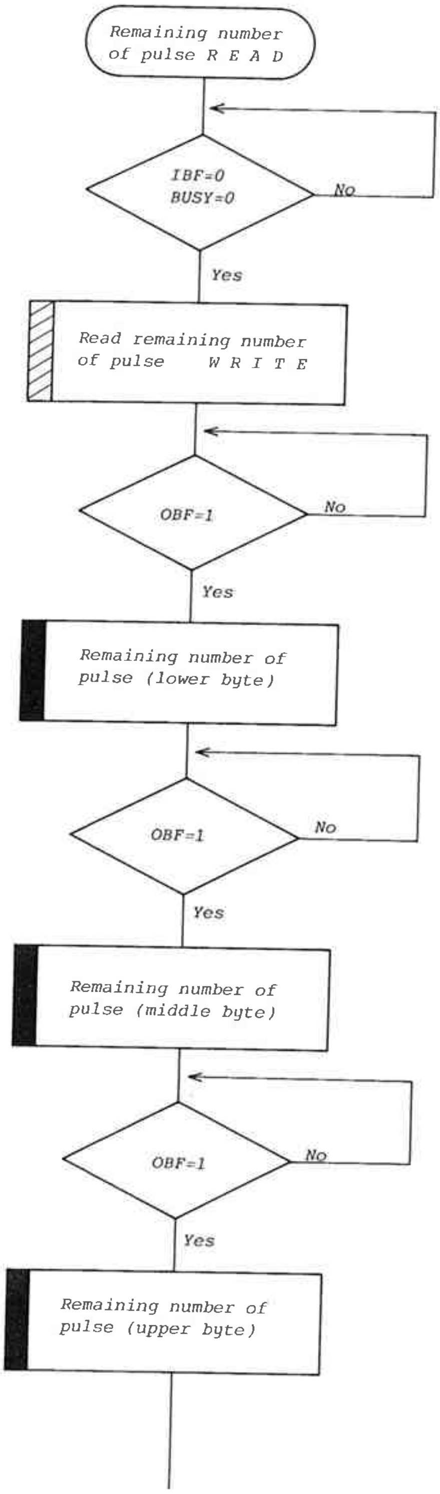
3-4-2 Remaining number of pulse

When the motor is stopped by limit switch or stop command during acceleration/deceleration or constant speed operation, the remaining number of pulse can be read by using this mode. If it is decided to finish the operation after the stop command, the original command and the remaining number of pulse can be input again to restart the mode.



The data becomes '0' when the operation has been successfully terminated.

The following flow chart shows the proper sequence of programming in order to read the data from PPMC.


Remaining number of pulse READ
IBF=0
BUSY=0
No
Yes
Read remaining number of pulse WRITE
OBF=1
No
Yes
Remaining number of pulse (lower byte)
OBF=1
No
Yes
Remaining number of pulse (middle byte)
OBF=1
No
Yes
Remaining number of pulse (upper byte)

7	6	5	4	3	2	1	0	
1	0	0	0	0	0	1	1	*Command*

Remaining number of pulse

7	6	5	4	3	2	1	0	
								(lower byte)
								(middle byte)
								(upper byte)

4. ELECTRICAL CHARACTERISTICS

4-1 PPMC101C CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (Ta=25°C)

Parameter	Symbol	Condition	Ratings	Unit
Supply Voltage	Vcc		-0.5 to +7.0	V
	V_{DD}		-0.5 to +7.0	V
Input Voltage	V_I		-0.5 to +7.0	V
Output Voltage	V_O		-0.5 to +7.0	V
Operating Temp.	T_{opt}		0 to +70	°C
Storage Temp.	T_{stg}		-65 to +150	°C

D.C. CHARACTERISTICS (Ta=0～+70°C, Vcc=+5V±5%, GND=0V)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Input High Voltage	V_{IH}	All except X1, X2, $\overline{RESET}$	2.0		Vcc	V
	V_{IH2}	X1, X2, $\overline{RESET}$	3.8		Vcc	V
Input Low Voltage	V_{IL}	All except X1, X2	-0.5		0.8	V
Output High Voltage	V_{OH}	D0-D7 ; I_{OH}=-400μA	2.4			V
	V_{OH1}	All other outputs ; I_{OH}=-50μA	2.4			V
Output Low Voltage	V_{OL}	D0-D7, SYNC ; I_{OL}=2.0mA			0.45	V
	V_{OL2}	All other outputs except D0-D7, SYNC ; I_{OL}=1.6mA			0.45	V
Low Input Load Current	I_{LI1}	Pin#21-24:L1-L4, Pin#27-34 V_{IL}=0.8V Pin#35-38:INT			0.4	mA
	I_{LI2}	$\overline{RESET}$, Pin#5:NC1 ; V_{IL}=0.8V			0.2	mA
Input Leakage Current	I_{IL}	Pin#1:NC1, Pin#39:CLOCK, $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, Ao Pin#7:GND Vss≦V_{IN}≦Vcc			±10	μA
Output Leakage Current	I_{OL}	D0-D7 High Z state Pin#20:GND+0.45≦V_{IN}≦Vcc			±10	μA
Supply Current	I_{DD}	V_{DD}		10	25	mA
	I_{DD} + I_{CC}	Total		65	135	mA

A.C. CHARACTERISTICS (Ta=0～+70°C, Vcc=VDD=+5V±5%, Vss=0V)

DBB READ

Parameter	Symbol	Condition	Min	Typ	Max	Unit
$\overline{CS}$, Ao setup to $\overline{RD}$	t_{AR}		0			ns
$\overline{CS}$, Ao hold after $\overline{RD}$	t_{RA}		0			ns
$\overline{RD}$ pulse width	t_{RR}	t_{CY}=2.5μs	250		2xt_{CY}	ns
$\overline{CS}$, Ao to Data out delay	t_{AD}				150	ns
$\overline{RD}$ to Data out delay	t_{RD}				150	ns
$\overline{RD}$ to Data float delay	t_{DF}		10		100	ns
Recovery time between $\overline{RD}$	t_{RV}		1			μs
Cycle time	t_{CY}	6 MHz X-tal	2.5			μs

DBB WRITE

Parameter	Symbol	Condition(*1)	Min	Typ	Max	Unit
$\overline{CS}$, Ao setup to $\overline{WR}$	t_{AW}		0			ns
$\overline{CS}$, Ao hold after $\overline{WR}$	t_{WA}		0			ns
$\overline{WR}$ pulse width	t_{WW}	t_{CY}=2.5μs	250		2xt_{CY}	ns
Data setup to $\overline{WR}$	t_{DW}		250			ns
Data hold after $\overline{WR}$	t_{WD}		0			ns

Note (*1) : D0 - D7 output C1=100pF

4-2 PPMC102A CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (Ta=25°C)

Parameter	Symbol	Condition	Ratings	Unit
Supply Voltage	Vcc		-0.5 to +7.0	V
	V_{DD}		-0.5 to +7.0	V
Input Voltage	V_I		-0.5 to +7.0	V
Output Voltage	V_O		-0.5 to +7.0	V
Operating Temp.	T_{opt}		0 to +70	°C
Storage Temp.	T_{stg}		-65 to +150	°C

D.C. CHARACTERISTICS (Ta=0～+70°C, Vcc=VDD=+5V±10%, Vss=0V)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Input High Voltage	V_{IH}	All except X1, X2, $\overline{RESET}$	2.0		Vcc	V
	V_{IH1}	X1, X2, $\overline{RESET}$	3.8		Vcc	V
Input Low Voltage	V_{IL}	All except X1, X2, $\overline{RESET}$	-0.5		0.8	V
	V_{IL1}	X1, X2, $\overline{RESET}$	-0.5		0.6	V
Output High Voltage	V_{OH}	D0-D7 ; $I_{OH}=-400\mu A$	2.4			V
	V_{OH1}	All other outputs ; $I_{OH}=-50\mu A$	2.4			V
Output Low Voltage	V_{OL}	D0-D7 ; $I_{OL}=2.0mA$			0.45	V
	V_{OL1}	P10-17, P20-27, SYNC ; $I_{OL}=1.6mA$			0.45	V
	V_{OL2}	PROG ; $I_{OL}=1.0mA$			0.45	V
Low Input Load Current	I_{LI}	P10-17, P20-27 ; $V_{IL}=0.8V$			0.5	mA
	I_{LI1}	$\overline{RESET}$, $\overline{SS}$; $V_{IL}=0.8V$			0.2	mA
Input Leakage Current	I_{IL}	T1, T2, $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, Ao, EA ; $Vss \leqq V_{IN} \leqq Vcc$			±10	μA
Output Leakage Current	I_{OFL}	D0-D7 High Z state ; $Vss+0.45 \leqq V_{IN} \leqq Vcc$			±10	μA
Supply Current	I_{DD}	V_{DD}			15	mA
	$I_{DD} + I_{CC}$	Total			125	mA

A.C. CHARACTERISTICS (Ta=0～+70°C, Vcc=VDD=+5V±10%, Vss=0V)

DBB READ

Parameter	Symbol	Condition	Min	Typ	Max	Unit
$\overline{CS}$, Ao setup to $\overline{RD}$	t_{AR}		0			ns
$\overline{CS}$, Ao hold after $\overline{RD}$	t_{RA}		0			ns
$\overline{RD}$ pulse width	t_{RR}		160			ns
$\overline{CS}$, Ao to Data out delay	t_{AD}	(*2)			130	ns
$\overline{RD}$ to Data out delay	t_{RD}	(*2)			130	ns
RD to Data float delay	t_{DF}				85	ns
Cycle time	t_{CY}		1.36		15	µs

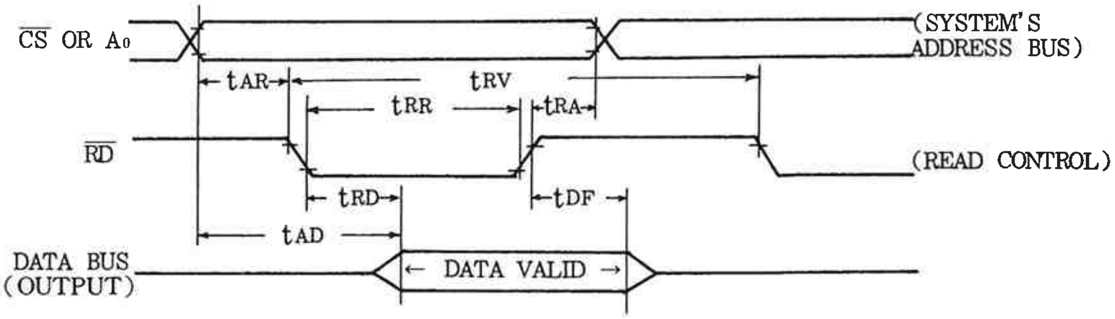
DBB WRITE

Parameter	Symbol	Condition	Min	Typ	Max	Unit
$\overline{CS}$, Ao setup to $\overline{WR}$	t_{AW}		0			ns
$\overline{CS}$, Ao hold after $\overline{WR}$	t_{WA}		0			ns
$\overline{WR}$ pulse width	t_{WW}		160			ns
Data setup to $\overline{WR}$	t_{DW}		130			ns
Data hold after $\overline{WR}$	t_{WD}		0			ns

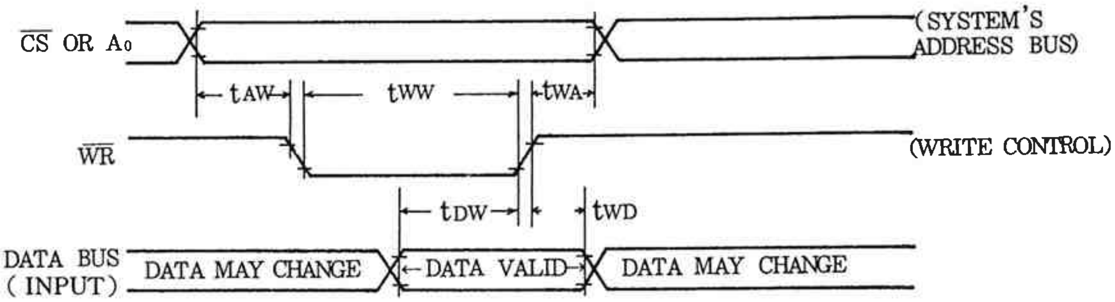
Note (*2) : CL = 150pF

TIMING

READ Operation (DBB Out Register)

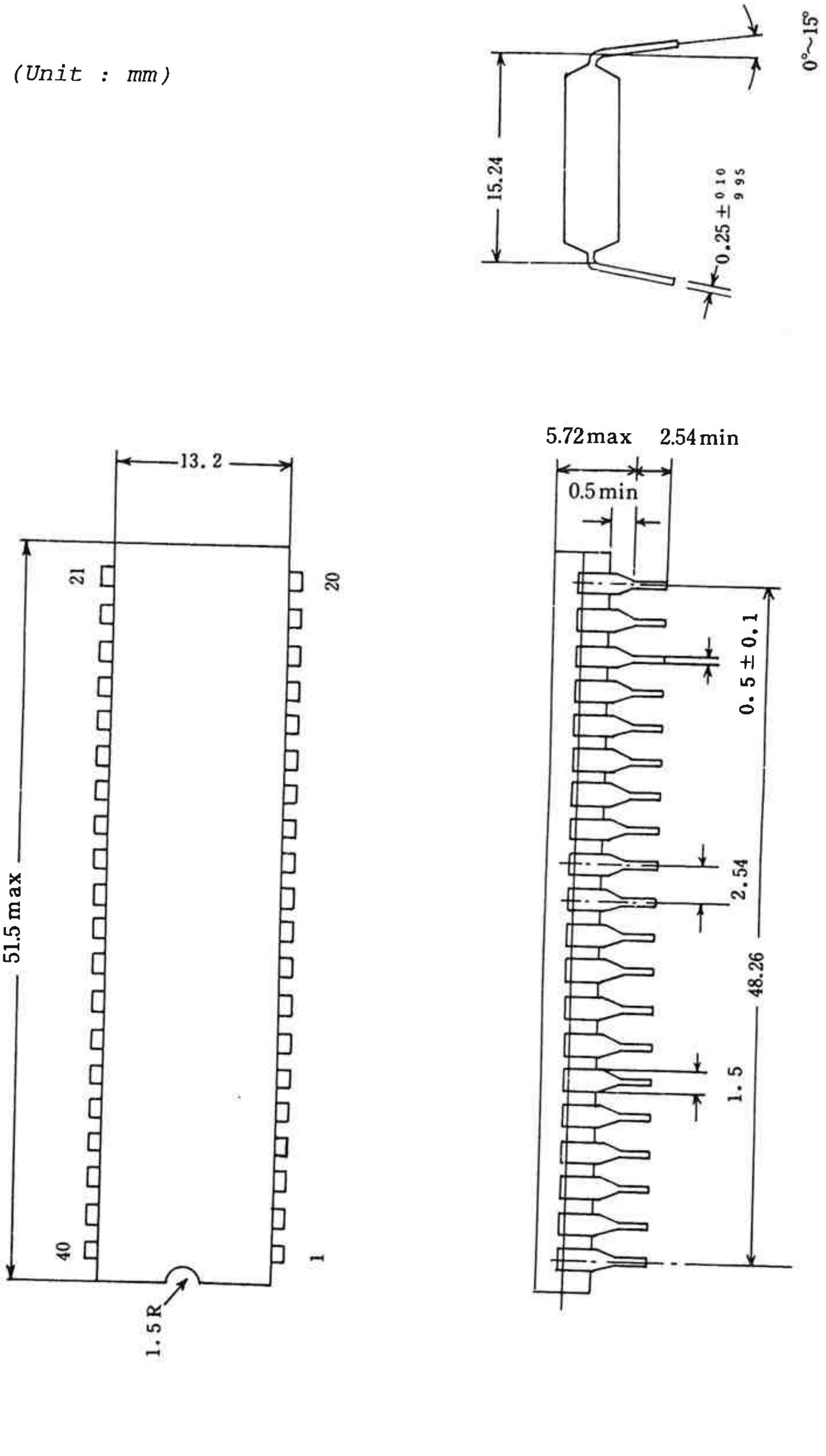


WRITE Operation (DBB In Register)



PACKAGE DIMENSION

(Unit : mm)

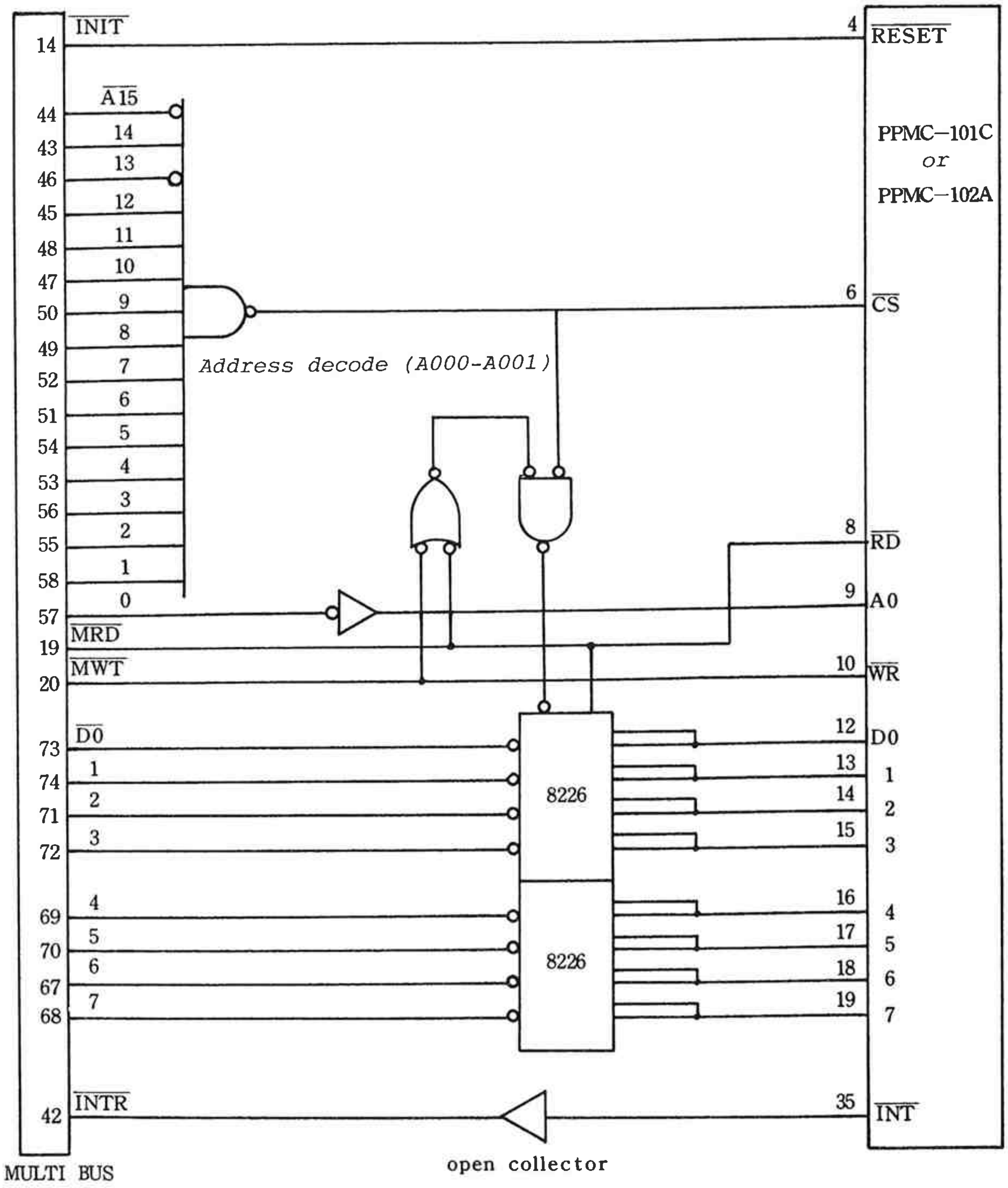


5. CONNECTION WITH PPMC101C/102A

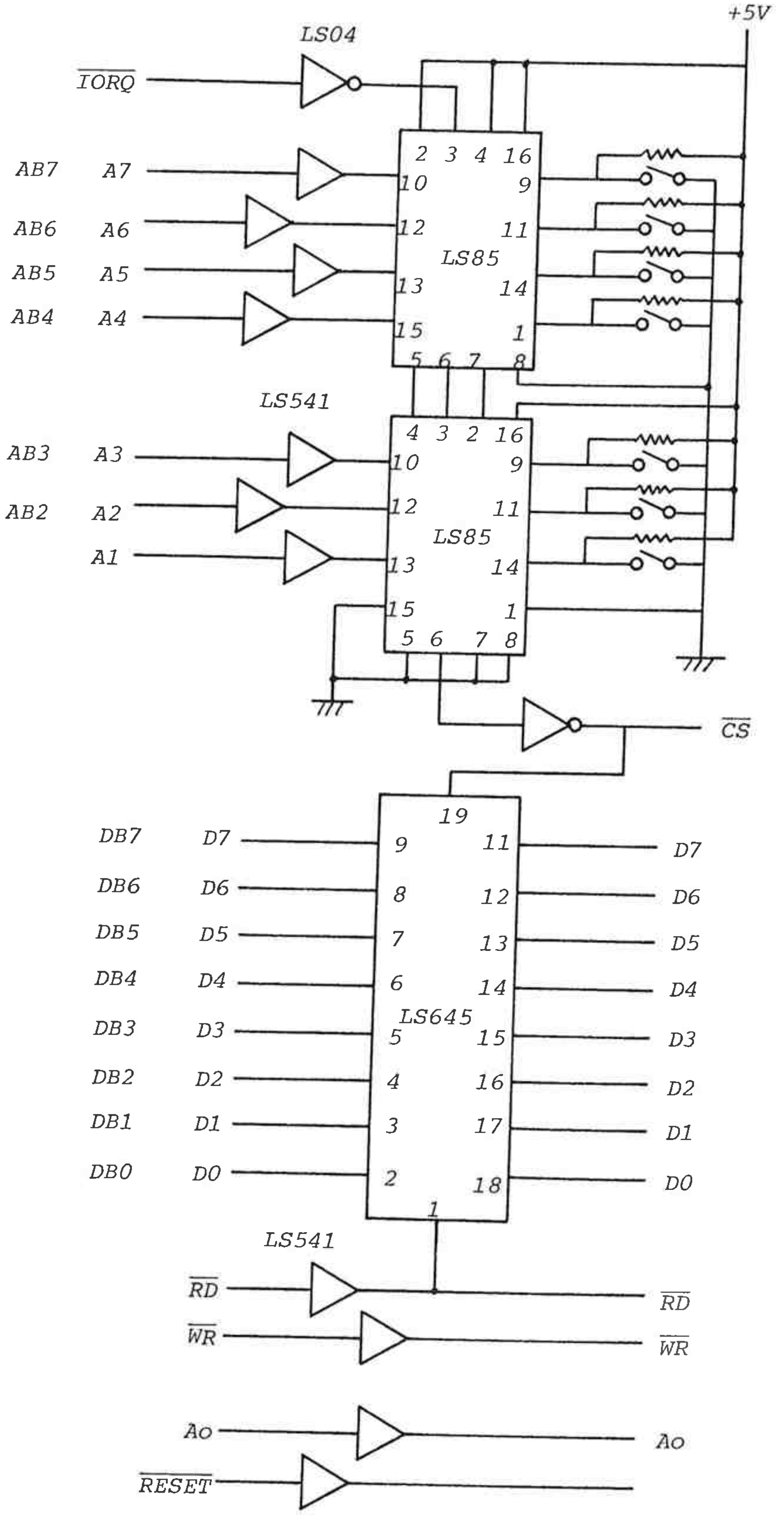
5-1 Interface to master CPU

PPMC101C/102A can be easily interfaced to INTEL 8080 as well as Motorola 6800 microprocessors by use of a simple additional circuits.

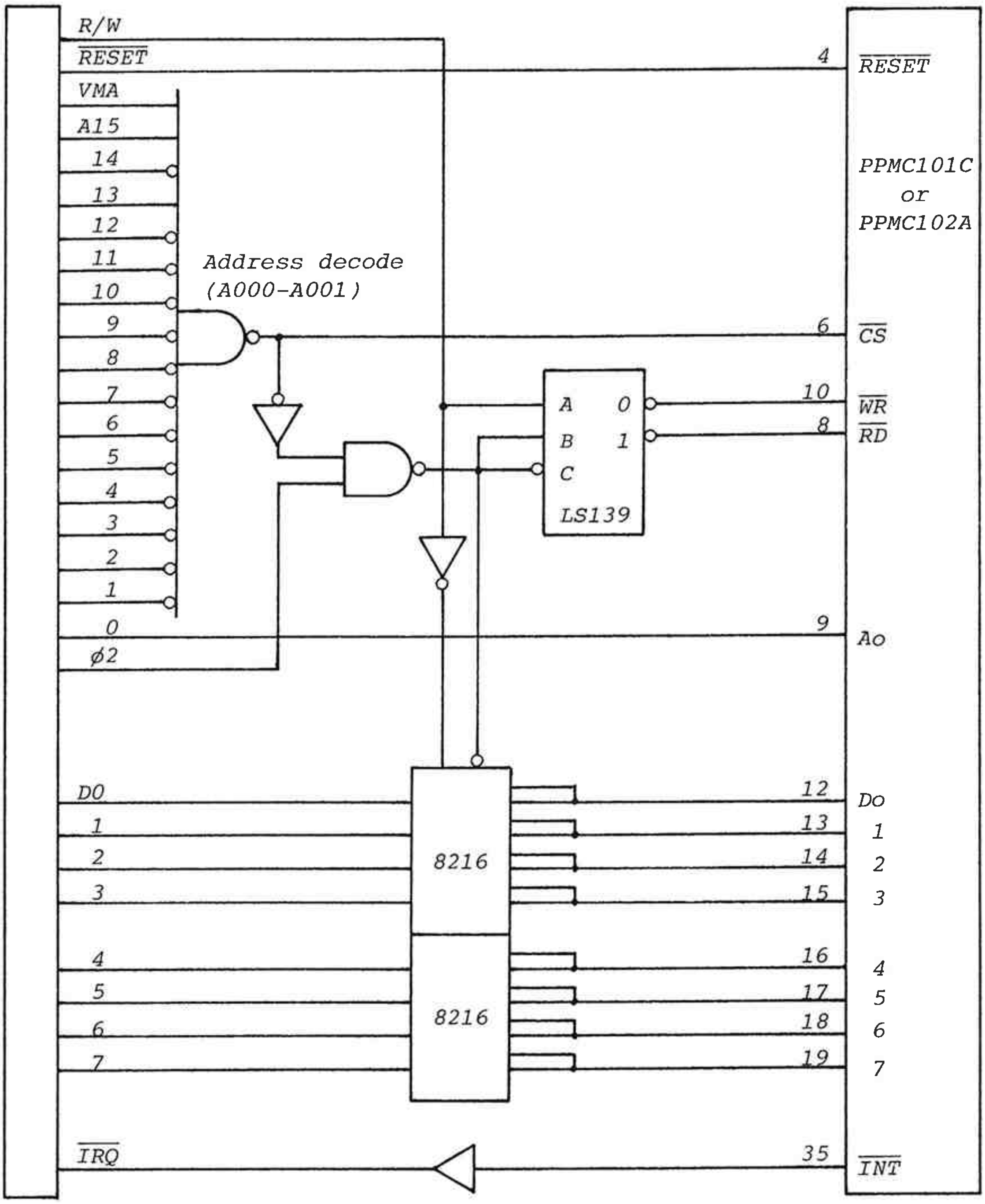
5-1-1 Interface to 8080 CPU



5-1-2 Interface to Z80 CPU



5-1-3 Interface to 6800 CPU

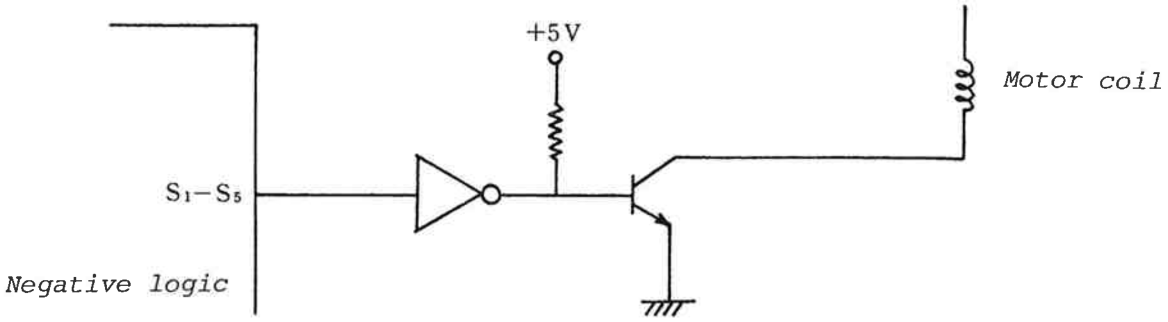


Open Collector

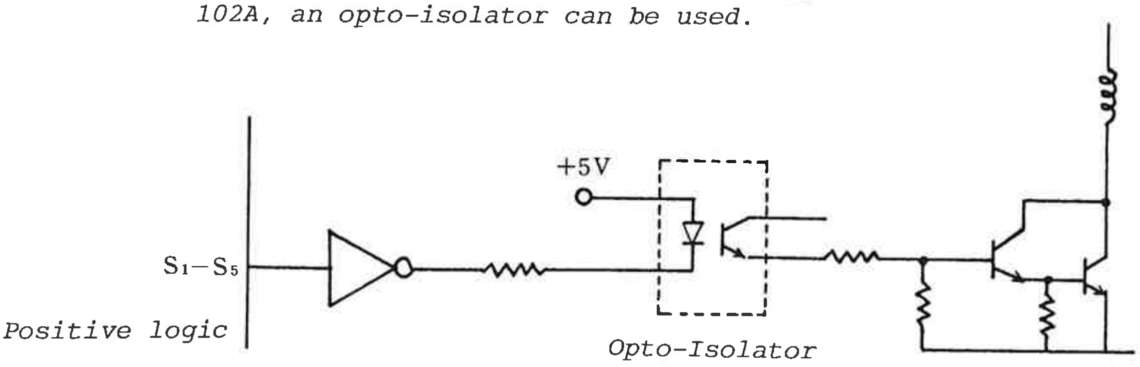
5-2 Interface to Stepper Motor

5-2-1 Phase Output

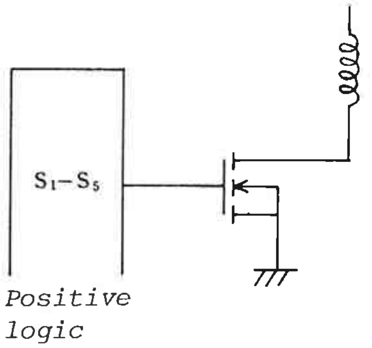
Each phase output can be set by positive logic or negative logic depending on the most suitable driving circuit. The driving capability of PPMC101C/102A is equal to one standard TTL Load. The following is a typical driving circuit.



Where isolation is needed between stepper motor and PPMC101C/102A, an opto-isolator can be used.



The following circuit shows how to use power MOS FET to drive a stepper motor.

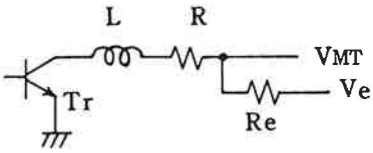


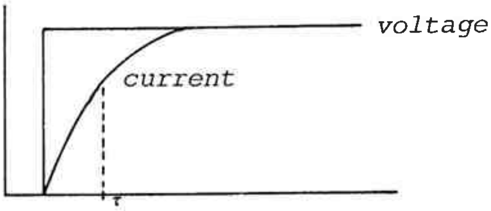
5-2-2 Excitation Circuit

When stepper motor rotates at high speed current change cannot keep up with high frequency, which reduces torque. But, it is possible to minimize change of torque by improving excitation circuit with idea mentioned below.

5-2-2-1 Improvement of Rising Characteristics

Excitation coil can be expressed by an equivalent circuit as shown below consisting of Resistor and Coil in series circuit.



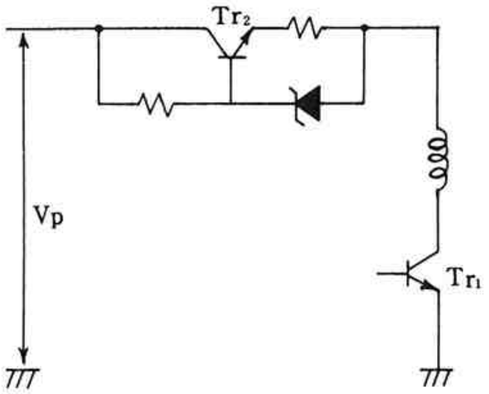


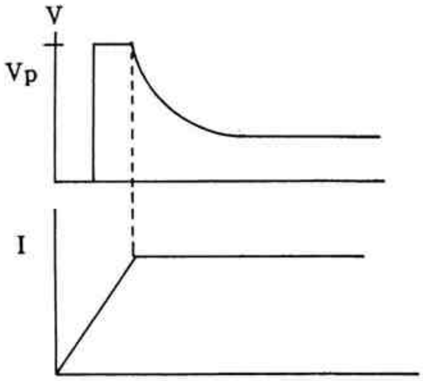
When Tr is ON in this circuit the current changes with the time constant $\tau = \frac{L}{R}$

By adding Resistor 'Re' in series, and proper increase of voltage with the same ratio of 'R' rising characteristics can be improved with the time constant $\tau e = \frac{L}{R+Re}$

This method is frequently used, but has disadvantage as well to increase of thermal loss by 'Re' with higher power requirement.

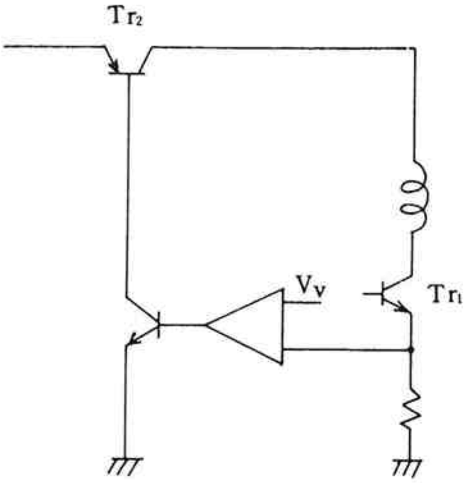
Another way to drive excitation coil is to generate constant current by the circuit shown below.

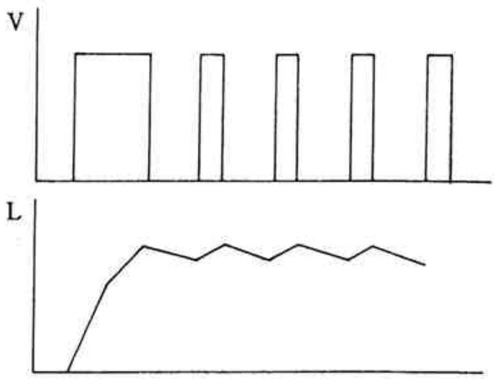




This is more efficient way but has disadvantage as well to consume high power by Tr2.

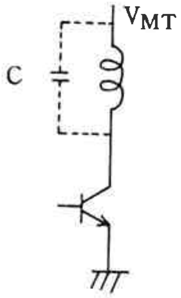
It is, therefore, recommended for best performance at high speed operation to detect the current and keep it constant by switching as shown below.





5-2-2-2 *Control Circuit*

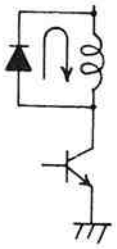
When excitation coil is turned "OFF", the running current must be off as soon as possible to avoid break torque. Supposing infinite VCED in transistor, following shows the best performance, but the transistor will be in danger to break up because of high voltage loaded into the "collctor" of transistor.



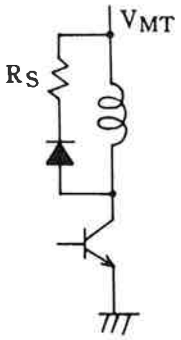
$$V = \sqrt{\frac{L}{C}}\, i + V_{MT}$$

i ·· *The current running through the coil 'L' right before "OFF"*

When operating at low speed, to use "diode" in the circuit would be the simplest way as shown below but it increases break torque by continuing current through the diode.



Higher performance can be achieved by adding resistor or zener diode in series with the diode (see below). Peak voltage, however must be carefully controlled no greater than VCEO of the transistor.



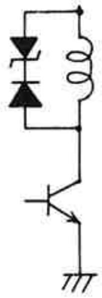
$V_{CEO} > V_{MT} + IR_S + V_{DF}$

V_{MT} ··· *Power voltage*

I ··· *The current running through the coil 'L' right before 'OFF"*

R_S ··· *Series resistor*

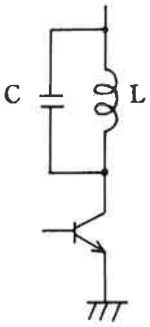
V_{DF} ··· *Diode forward voltage*



$V_{CEO} > V_Z + V_{DF} + V_{MT}$

V_Z ··· *Zener voltage*

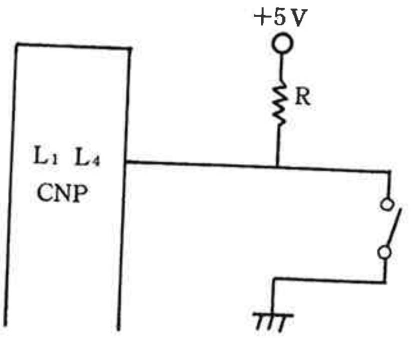
The voltage rising from LC can be also controlled within VCED by adding C (capaitor) in parallel with L as shown below.



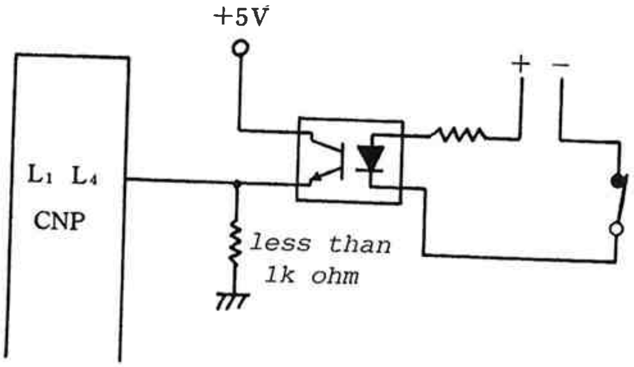
$$V_{CEO} > \sqrt{\frac{L}{C}i} + V_{MT}$$

5-2-3 Limit Switch

Limit switch can be connected to PPMC as shown in the following figure. The value of 'R' is recommended to be as small as possible to reduce noise.



The following figure shows how to isolate a normally closed limit switch electronically from PPMC101C/102A with inverted logic connection.


+5V
+ –
L1 L4
CNP
less than
1k ohm