

2-Axes Motor Control IC with Interpolation Function

MCX312 User's Manual

| | |
|------------|----------|
| 2008-12-18 | Ver. 1.7 |
| 2009-03-10 | Ver. 1.9 |
| 2010-03-25 | Ver. 2.0 |
| 2011-07-27 | Ver. 2.1 |
| 2011-08-22 | Ver. 2.2 |

NOVA electronics

Introduction

Before using the MCX312, please read this manual thoroughly to ensure correct usage within the scope of the specification such as the signal voltage, signal timing, and operation parameter values.

In general, semiconductor products sometimes malfunction or fail to function. When incorporating this IC in a system, make sure that a safe system is designed to avoid any injuries or property damage caused by malfunctioning of this IC.

This IC is designed for application in general electronic devices (industrial automation devices, industrial robots, measuring instruments, computers, office equipment, household electrical goods, and so on). This IC is not intended for the use in high-performance high-reliability equipment whose failure or malfunctioning may directly cause death or injuries (atomic energy control equipment, aerospace equipment, transportation equipment, medical equipment, and various safety devices) and the operation for such use is not guaranteed. The customer shall be responsible for the use of this IC in any such high-performance and high-reliability equipment.

Notes on S-curve acceleration/deceleration driving

This IC is equipped with a function that performs decelerating stop for a fixed pulse drive with S-curve deceleration of the symmetrical acceleration/deceleration. However, when the initial speed is set to an extremely low speed (10 or less), slight premature termination or creep may occur. Before using a S-curve deceleration drive, make sure that your system allows premature termination or creep.

| | |
|---|----|
| 1.OUTLINE ----- | 1 |
| 2.TheDescriptions of Functions ----- | 6 |
| 2.1 Pulse Output Command ----- | 6 |
| 2.1.1 Fixed Driving Output ----- | 6 |
| 2.1.2 Continuous Driving Output ----- | 7 |
| 2.2 Acceleration and Deceleration ----- | 7 |
| 2.2.1 Constant Speed Driving ----- | 7 |
| 2.2.2 Trapezoidal Driving [Symmetrical] ----- | 8 |
| 2.2.3 Non-Symmetrical Trapezoidal Acceleration ----- | 9 |
| 2.2.4 S-curve Acceleration/Deceleration Driving ----- | 11 |
| 2.2.5 Pulse Width and Speed Accuracy ----- | 14 |
| 2.3 Position Control ----- | 15 |
| 2.3.1 Logic Position Counter and Real position Counter ----- | 15 |
| 2.3.2 Compare Register and Software Limit ----- | 15 |
| 2.3.3 Position Counter Variable Ring ----- | 16 |
| 2.3.4 Clearing a Real Position Counter Using an External Signal ----- | 16 |
| 2.4 Interpolation ----- | 18 |
| 2.4.1 Linear Interpolation ----- | 18 |
| 2.4.2 Circular Interpolation ----- | 19 |
| 2.4.3 The Bit Pattern Interpolation ----- | 21 |
| 2.4.4 Constant Vector Speed ----- | 25 |
| 2.4.5 Continuous Interpolation ----- | 25 |
| 2.4.6 The Acceleration / Deceleration Control in Interpolation ----- | 28 |
| 2.4.7 Single-step interpolation (from Command or External Signal) ----- | 31 |
| 2.4.8 Multichip Axes Interpolation ----- | 33 |
| 2.5 Interrupt ----- | 37 |
| 2.6 Other Functions ----- | 39 |
| 2.6.1 Driving By External Pulses ----- | 39 |
| 2.6.2 Pulse Output Type Selection ----- | 40 |
| 2.6.3 Pulse Input Type Selection ----- | 41 |
| 2.6.4 Hardware Limit Signals ----- | 41 |
| 2.6.5 Interface to Servo Motor Drivers ----- | 41 |
| 2.6.6 Emergency Stop ----- | 42 |
| 2.6.7 Status Output ----- | 42 |
| 2.6.8 General Purpose Input / Output Signal ----- | 43 |
| 2.6.9 Input Signal Filter ----- | 43 |
| 3. Pin Assignments and Signal Description ----- | 45 |
| 4. Register ----- | 50 |
| 4.1 Register Address by 16-bit Data Bus ----- | 50 |
| 4.2 Register Address by 8-bit Data Bus ----- | 51 |
| 4.3 Command Register: WR0 ----- | 52 |
| 4.4 Mode Register1: WR1 ----- | 52 |
| 4.5 Mode Register2: WR2 ----- | 53 |
| 4.6 Mode Register3: WR3 ----- | 55 |
| 4.7 Output Register: WR4 ----- | 57 |
| 4.8 Interpolation Mode Register: WR5 ----- | 57 |

| | |
|---|-----------|
| 4.9 Data Register: WR6/WR7 ----- | 58 |
| 4.10 Main Status Register: RR0 ----- | 58 |
| 4.11 Status Register 1: RR1 ----- | 59 |
| 4.12 Status Register 2: RR2 ----- | 60 |
| 4.13 Status Register 3: RR3 ----- | 61 |
| 4.14 Input Register: RR4 / RR5 ----- | 61 |
| 4.15 Data-Read Register: RR6 / RR7 ----- | 62 |
| | |
| 5. Command Lists ----- | 63 |
| | |
| 6. Commands for Data Writing ----- | 65 |
| 6.1 Range Setting ----- | 65 |
| 6.2 Jerk Setting ----- | 66 |
| 6.3 Acceleration Setting ----- | 66 |
| 6.4 Deceleration Setting ----- | 67 |
| 6.5 Initial Speed Setting ----- | 67 |
| 6.6 Drive Speed Setting ----- | 67 |
| 6.7 Output Pulse Number / Interpolation Finish Point Setting ----- | 68 |
| 6.8 Manual Decelerating Point Setting ----- | 68 |
| 6.9 Circular Center Setting ----- | 68 |
| 6.10 Logical Position Counter Setting ----- | 69 |
| 6.11 Real position Counter Setting ----- | 69 |
| 6.12 COMP+ Register Setting ----- | 69 |
| 6.13 COMP- Register Setting ----- | 69 |
| 6.14 Acceleration Counter Offsetting ----- | 70 |
| 6.15 NOP (For Axis Switching) ----- | 70 |
| | |
| 7. Commands for Reading Data ----- | 71 |
| 7.1 Logical Position Counter Reading ----- | 71 |
| 7.2 Real position Counter Reading ----- | 71 |
| 7.3 Current Drive Speed Reading ----- | 71 |
| 7.4 Current Acceleration / Deceleration Reading ----- | 72 |
| 7.5 Maximum Finish Point Reading For Multichip Linear Interpolation ----- | 72 |
| | |
| 8. Driving Commands ----- | 73 |
| 8.1 +Direction Fixed Driving ----- | 73 |
| 8.2 -Direction Fixed Driving ----- | 74 |
| 8.3 +Direction Continuous Driving ----- | 74 |
| 8.4 -Direction Continuous Driving ----- | 74 |
| 8.5 Drive Status Holding ----- | 75 |
| 8.6 Drive Status Holding Release / Finishing Status Clear ----- | 75 |
| 8.7 Decelerating Stop ----- | 75 |
| 8.8 Sudden Stop ----- | 75 |
| | |
| 9. Interpolation Commands ----- | 76 |
| 9.1 2-Axis Linear Interpolation ----- | 76 |
| 9.2 CW Circular Interpolation ----- | 76 |
| 9.3 CCW Circular Interpolation ----- | 77 |
| 9.4 2-Axis Bit Pattern Interpolation ----- | 77 |
| 9.5 BP Register Data Writing Enabling ----- | 77 |
| 9.6 BP Register Data Writing Disabling ----- | 77 |
| 9.7 BP Data Stack ----- | 78 |
| 9.8 BP Data Clear ----- | 78 |
| 9.9 Single Step Interpolation ----- | 78 |

| | |
|--|-----------|
| 9.10 Deceleration Enabling | 78 |
| 9.11 Deceleration Disabling | 79 |
| 9.12 Interpolation Interrupt Clear | 79 |
| 9.13 Maximum Finish Point Clear For Multichip Linear Interpolation | 79 |
| 10. Connection Examples | 80 |
| 10.1 Connection Example for 68000 CPU | 80 |
| 10.2 Connection Example for Z80 CPU | 80 |
| 10.3 Connection Example for H8 CPU | 81 |
| 10.4 Connection Example | 82 |
| 10.5 Pulse Output Interface | 82 |
| 10.6 Connection Example for Input Signals | 83 |
| 10.7 Connection Example for Encoder | 83 |
| 11. Example Program | 84 |
| 12. Electrical Characteristics | 90 |
| 12.1 DC Characteristics | 90 |
| 12.2 AC Characteristics | 91 |
| 12.2.1 Clock | 91 |
| 12.2.2 Read / Write Cycle | 91 |
| 12.2.3 BUSYN Signal | 92 |
| 12.2.4 SCLK/Output Signal Timing | 92 |
| 12.2.5 Input Pulses | 92 |
| 12.2.6 General Purpose Input / Output Signals | 93 |
| 13. Timing of Input / Output Signals | 94 |
| 13.1 Power-On Reset | 94 |
| 13.2 Fixed Pulse or Continuous Pulse Driving | 94 |
| 13.3 Interpolation | 95 |
| 13.4 Start Driving after Hold Command | 95 |
| 13.5 Sudden Stop | 95 |
| 13.6 Decelerating Stop | 96 |
| 14. Package Dimensions | 97 |
| 15. Specifications | 99 |
| Appendix A Speed Profile of Acceleration/Deceleration Drive | A1 |
| Appendix B Important Notice | B1 |

Update history

Aug/22/2011 Ver. 2.2

P7 ■ Changing a Drive speed During Driving and Fig.2.5 has been deleted.

P6 ~ 44 Chapter 2

The figure number of Fig.2.6 ~ Fig.2.42 is carried one because of Fig.2.5 having been deleted.

P58 The low-word data-writing 16-bit (WD15~WD0) is for register RR6 setting, and the high-word data-writing 16-bit (WD31~WD16) is for register RR7 setting. ← The low-word data-writing 16-bit (WD15~WD0) is for register WR6 setting, and the high-word data-writing 16-bit (WD31~WD16) is for register WR7 setting.

P90 12.1 DC Characteristics Reservation Temperature → Preservation Temperature

P92 12.2.3 BUSYN Signal

It is low when BUSYN is active. And BUSYN is low after 2 SCLK cycles when WRN ↑ active. → BUSYN becomes low active for maximum 2 SCLK cycles from WRN ↑. During the time, IC cannot accept Read/Write.

12.2.4 SCLK/Output Signal Timing

The following output signal is synchronized with SCLK output signal. The level at ACLK ↑ will be changed. Output signals : nPP/PLS, nPM/DIR, nDRIVE, nASND, nCNST, nDSND, nCMPP, nCMPM, nACASND and nACDSND. → The following output signals are synchronized with SCLK output signal. The level will be changed at SCLK ↑. Output signals : nPP/PLS, nPM/DIR, nDRIVE, nASND, nDSND, nCMPP, and nCMPM.

12.2.5 Input Pulses

a. In A/B quadrature pulse input mode, when nECA and nECB input pulses are changed, the value of real position counter will be changed to the value of those input pulses changed after the period of longest SCLK4 is passed. → In quadrature pulses input mode, when nECA and nECB input pulses are changed, the value of real position counter will be reflected in maximum 4 SCLK cycles.

b. In UP/DOWN pulse input mode, the real position counter will become the value of those input pulses changed, after the period between the beginning of nPPIN, nPMIN ↑ and the time of SCLK 4 cycle is passed. → In UP/DOWN pulse input mode, the value of real position counter will be reflected in maximum 4 SCLK cycles from nPPIN and nPMIN input ↑.

P94 13.2 Fixed or Continuous Driving

a. This first driving pulses (nPP, nPM, and nPLS) will be output after 3 SCLK cycles when BUSYN is ↑. → Driving pulses (nPP, nPM, and nPLS) shown as above are positive logic pulses. And the first driving pulse will be output after 3 SCLK cycles from BUSYN ↑.

b. The nDIR (direction) signal is valid after 1 SCLK cycle when BUSYN is ↑. → nDIR (direction) signal is valid after 1 SCLK cycle from BUSYN ↑.

c. The nDRIVE becomes Hi level when BUSYN is ↑. → dDRIVE becomes Hi level from BUSYN ↑.

d. The nASND and nDSND are on invalid level after 3 SCLK cycles when BUSYN is ↑. → nASND and nDSND are on valid level after 3 SCLK cycles from BUSYN ↑ and they return to low level when the cycle of final pulse output has finished.

P95 13.3 Interpolation

a. The first pulses (nPP, nPM, and nPLS) of interpolation driving will be output after 4 SCLK cycles when BUSYN is ↑. → The first pulses (nPP, nPM, and nPLS) of interpolation driving will be output after 4 SCLK cycles from BUSYN ↑

b. nDRIVE will become Hi level after 1 SCLK cycle when BUSYN is ↑. → nDRIVE will become Hi level after 1 SCLK cycle from BUSYN ↑.

c. DIR signal keeps the active level in 1 SCLK cycle before and after the Hi level pulse outputting. → When 1 pulse 1 direction type, nDIR signal is on valid level while Hi level pulse outputting and 1 SCLK cycle before and after Hi level pulse outputting. (When drive pulse is positive logic pulse.)

14.4 Start Driving after Hold Command

a. The pulses (nPP, nPM, and nPLS) of each axis will start outputting after 3 SCLK cycles when BUSYN is ↑. → a. The pulses (nPP, nPM, and nPLS) of each axis will start outputting after 3 SCLK cycles from BUSYN ↑.

b. nDRIVE will become Hi level when BUSYN is ↑ for each axis. → b. nDRIVE will become Hi level when BUSYN ↑ for each axis.

Jul/27/2011 Ver. 2.1

B9 iv Notice for continuous interpolation has been added.

Mar/25/2010 Ver. 2.0

P9 [Note] When continuous driving or circular interpolation driving is performed after fixed driving, WR3 /D5 bit must be reset 0 in advance.

P12 Added " e. When the fixed S-curve acceleration / deceleration driving is performed, the driving speed does not seldom reach the setting value."

P63 CP -1,073,741,824 ~ +1,073,741,824
CM -1,073,741,824 ~ +1,073,741,824

P99 Comparison Register
###(((COMP + Register Position comparison range -1,073,741,824 ~ +1,073,741,824
###(((COMP - Register Position comparison range -1,073,741,824 ~ +1,073,741,824

PB8 Our email address

10/19/2009 Ver. 1.9

P12 Added "SV must be set as more than 100" to the constraint of S-curve Acceleration / Deceleration Driving

P67 Separated two cases such as Trapezoidal Acceleration / Deceleration Driving and S-curve Acceleration / Deceleration Driving more clearly and added "SV must be set as more than 100" to 6.5 Initial Speed Setting.

10/03/2009 Ver. 1.8

P54, 55 WR2 D9 Descriptions

18/12/2008 Ver. 1.7

PB1~B2 Added Appendix B Technical Information I • II

6/8/2008 Ver. 1.6

PB1~B6 Added Appendix B Technical Information

3/7/2006 Ver. 1.5

P91~93 (the following items in the table)

| | | |
|------------------|---|------------|
| Wavelength | → | Width |
| Reservation Time | → | Hold Time |
| Established Time | → | Setup Time |

1/6/2006 Ver. 1.4

P53 line 37 the start → the end
P53 line 39 the end → the start

11/17/2004 Ver. 1.3

P12 line 46 $4000 \times 10 = 40000\text{PPS}$ → $4000 \times 10 = 40000\text{PPS}$

P14 line 29 tolerance → jitter

P37 line 6 During the power resetting, → When resetting,

P39 line 14 nEXPP → nEXPM

P39 line 19-22 (Corrected a paragraph.)

P46 line 4 n○○○ → ○○○N

P47 line 13 HKMT+ → HLMT+

P47 line 20 HKMT- → HLMT-

P53 line 46-47 (Added a paragraph, "Each axis is with...".)

P65 line 32 Acceleration/Deceleration and jerk is → Acceleration/Deceleration is

P71 line 11-12 (Corrected a paragraph.)

P73 line 23 real position → logical position

P74 line 8 real position → logical position

P74 line 22 real position → logical position

P74 line 36 real position → logical position

P91 line 10 Delay Time 21nS → Delay Time

P91 line 11 Delay Time 23nS → Delay Time

P93 line 10 WRN→nOUT7~0 → WRN ↑ →nOUT7~0

P94 line 21

dDRIVE → nDRIVE

P99 line 28-35

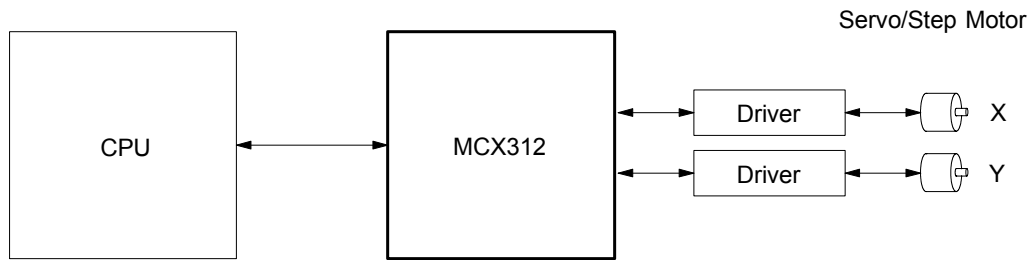
(Added descriptions of “multiple...” to the end of each line.)

1. OUTLINE

MCX312 is a 2-axis motion control IC which can control 2 axes of either stepper motor or pulse type servo drivers for position, speed, and interpolation controls. All of the MCX312's function are controlled by specific registers. There are command registers, data registers, status registers and mode registers. This motion control IC has the following built-in functions:

■ Individual Control for 2 Axes

MCX312 controls motors through pulse string driving. The IC can control motors of two axes independently with a single chip. Each of the two axes has identical function capabilities, and is controlled by the same method of operation with constant speed, trapezoidal or S-curve driving.



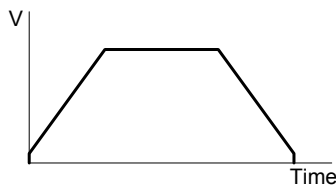
■ Speed Control

The speed range of the pulse output is from 1PPS to 4MPPS for constant speed, trapezoidal or S-curve acceleration/deceleration driving. Speed accuracy of the pulse output is less than $\pm 0.1\%$ (at CLK=16MHz). The speed of driving pulse output can be freely changed during the driving.

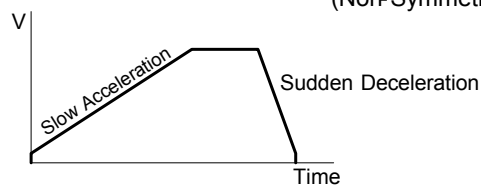
■ Acceleration/deceleration driving

The IC can control each axis for acceleration/deceleration of constant speed driving, trapezoidal acceleration/deceleration driving (symmetry/non-symmetry), and S-curve acceleration/deceleration. Automatic acceleration/deceleration of linear acceleration fixed speed pulse driving is available and no need to set deceleration starting point by manual. Since a primary linear increase/decrease method is applied for S-curve acceleration/deceleration, the speed curve forms a secondary parabola acceleration/deceleration curve. In S-curve acceleration and deceleration fixed driving, automatic deceleration is available for symmetrical S-curve only and triangle waveforms during S-curve acceleration/deceleration are prevented by a special method.

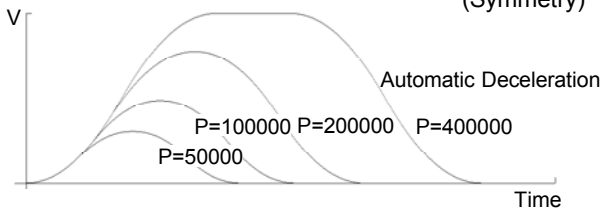
Trapezoidal Acceleration/Deceleration Driving (Symmetry)



Trapezoidal Acceleration/Deceleration Driving (Non-Symmetry)

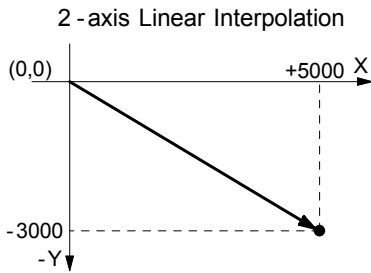


Parabola S -curve Acceleration/Deceleration Driving (Symmetry)



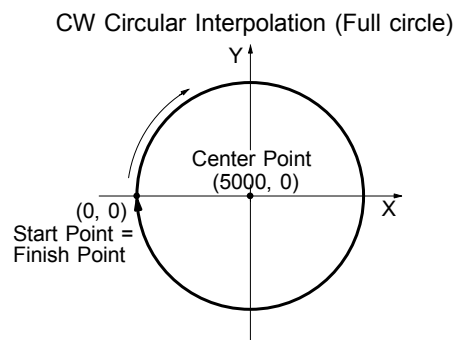
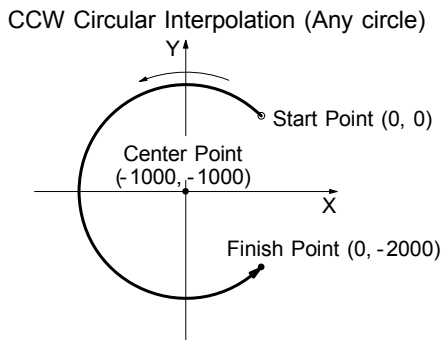
Linear Interpolation

2-axis linear interpolation can be performed. The position boundary is between coordinates $-8,388,607$ and $+8,388,607$, and the positioning error is within ± 0.5 LSB (Least Significant Bit). The interpolation speed range is from 1 PPS to 4 MPPS.



Circular Interpolation

Circular interpolation can be performed. The position boundary is between coordinates $-8,388,608$ and $+8,388,607$, and the positioning error is within ± 1 LSB. The interpolation speed range is from 1 PPS to 4 MPPS.

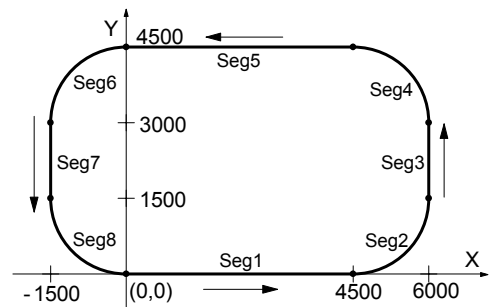


Bit Pattern Interpolation

This interpolation driving receives, for each axis in pulses, interpolation data that was converted to packet (a block of a predetermined amount of data) through the operation by the upper-level CPU and outputs interpolation pulses consecutively at the specified drive speed. This function enables drawing of various loci created by the upper-level CPU.

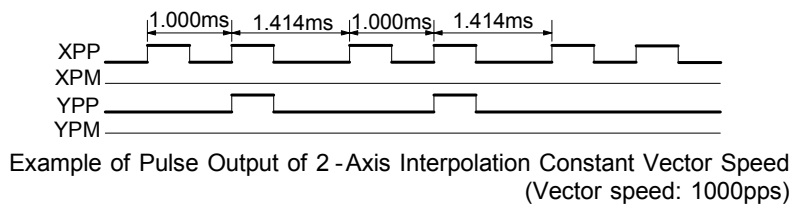
Continuous Interpolation

Different interpolation methods can be used continuously, linear interpolation \rightarrow circular interpolation \rightarrow linear interpolation The maximum drive speed of performing continuous interpolation is 2 MHz.



Constant Vector Speed Control

This function performs a constant vector speed. During the interpolation driving, MCX312 can set a 1.414 times pulse cycle for 2-axis simultaneous pulse output.



Example of Pulse Output of 2-Axis Interpolation Constant Vector Speed (Vector speed: 1000pps)

■ Position Control

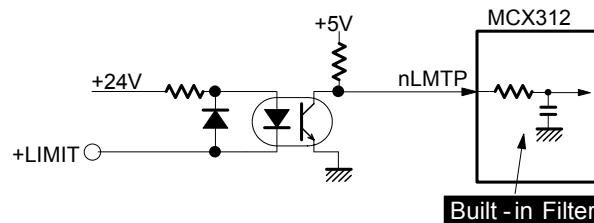
Each axis has a 32-bit logic position counter and a 32-bit real position counter. The logic position counter counts the number of output pulse, and the real position counter counts the feedback number of pulse from the external encoder or linear scale.

■ Compare Register and Software Limit

Each axis has two 32-bit compare registers for logical position counter and real position counter. The comparison result can be read from the status registers. The comparison result can be notified by an interrupt signal. These registers can be also functioned as software limits.

■ Input Signal Filter

The IC is equipped with an integral type filter in the input step of each input signal. It is possible to set for each input signal whether the filter function is enabled or the signal is passed through. A filter time constant can be selected from eight types.



■ Driving by External Signal

It is possible to control each axis by external signals. The +/- direction fixed driving, continuous driving or in manual pulsar mode can be also performed through the external signals. This function is used for JOG or teaching modes, and will share the CPU load.

■ Input for Home Search

Each axis has three external input signals to deceleration-stop during driving. Applying those input signals can perform high speed near home search, home search and encoder Z-signal search.

■ Servo Motor Feedback Signals

Each axis includes input pins for servo feedback signals such as in positioning.

■ Interrupt Signals

Interrupt signals can be generated when: (1). the start / finish of a constant speed drive during the acceleration/deceleration driving, (2). the end of driving, and (3). the compare result once higher / lower the border-lines of the position counter range. An interrupt signal can be also generated during the interpolation driving.

■ Real Time Monitoring

During the driving, the present status such as logical position, real position, drive speed, acceleration / deceleration, status of accelerating / decelerating and constant driving can be read.

■ 8 or 16 Bits Data Bus Selectable

MCX312 can be connected to either 8-bit or 16-bit CPU. Fig. 1.1 is the IC functional block diagram.

It consists of same functioned X and Y axes control sections and interpolation counting sections. Fig. 1.2 is the functional block diagram of each axis control section.

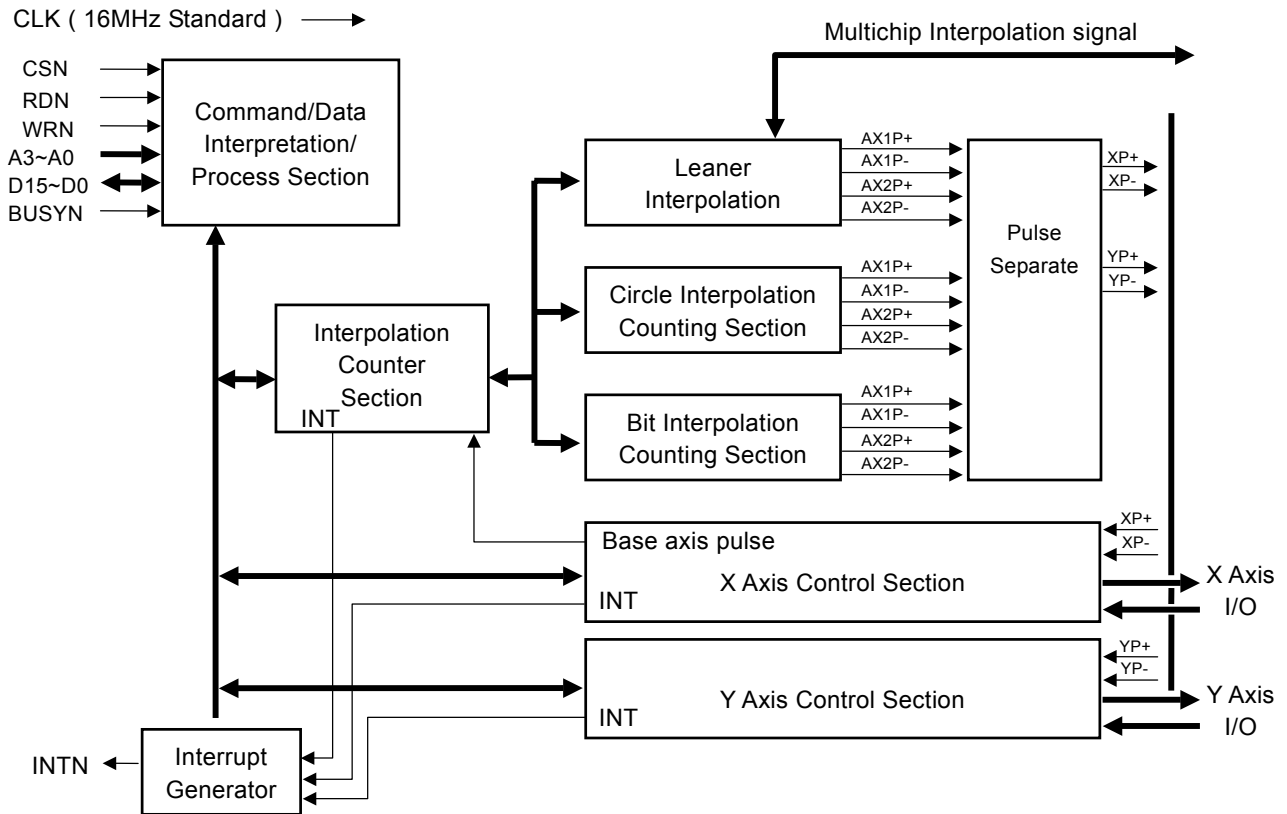


Fig. 1-1 MCX312 Functional Block Diagram

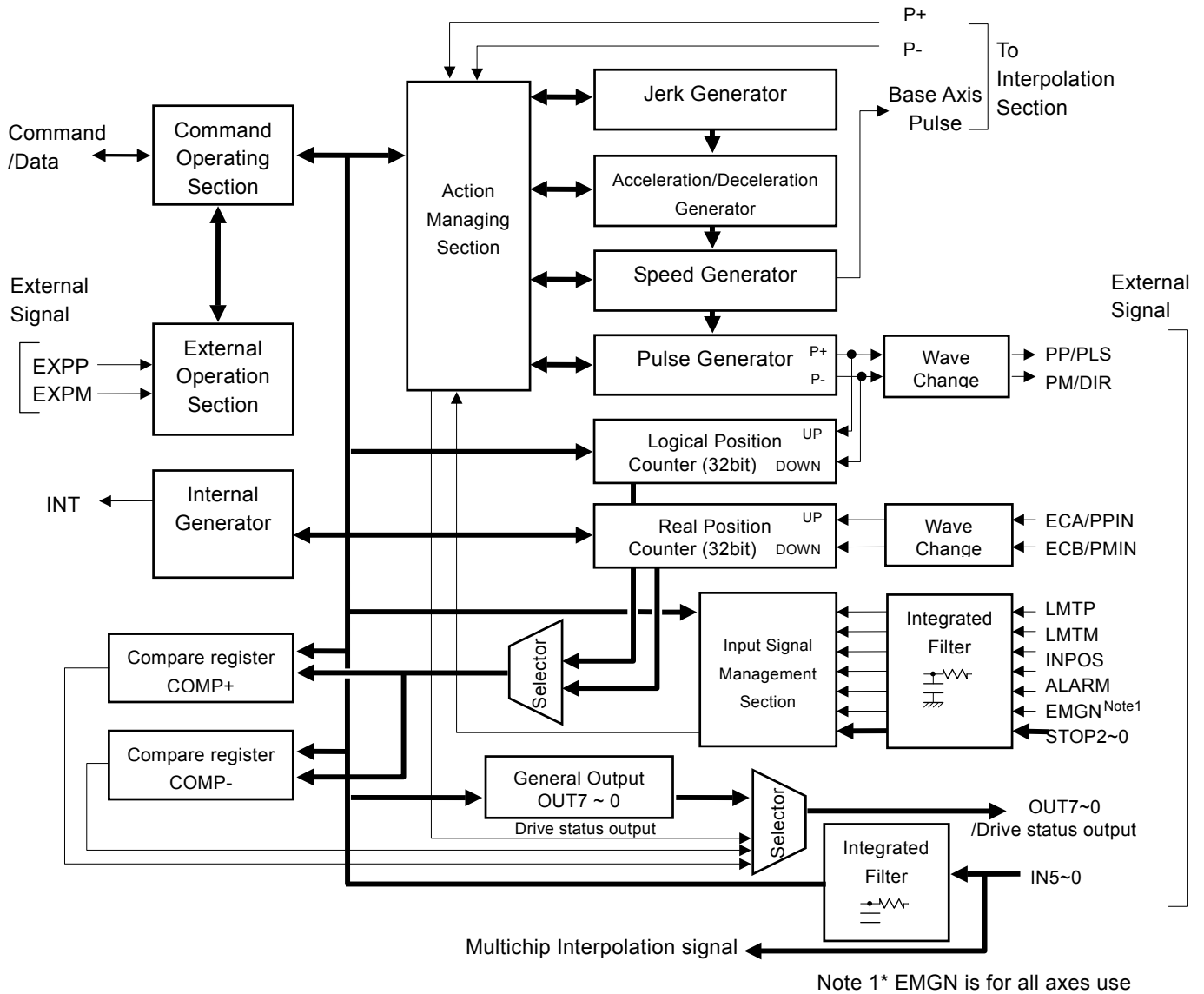


Fig. 1-2 Functional Block Diagram of Axis Control Section

2. The Descriptions of Functions

2.1 Pulse Output Command

There are two kinds of pulse output commands: fixed driving output and continuous driving output.

2.1.1 Fixed Driving Output

When host CPU writes a pulse numbers into MCX312 for fixed driving and configures the performance such as acceleration / deceleration and speed, MCX312 will generate the pulses and output them automatically. Fixed driving operation is performed at acceleration/deceleration, As shown in Fig. 2.1, automatic deceleration starts when the number of pulses becomes less than the number of pulses that were utilized at acceleration, and driving terminates at completion of the output of the specified output pulses. For fixed driving in acceleration / deceleration, the following parameters must be set.

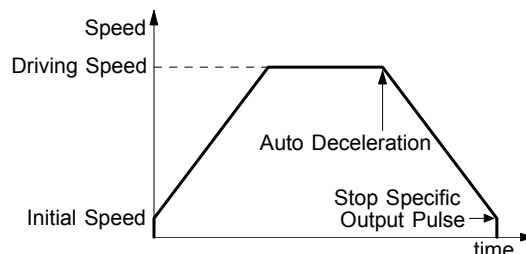


Fig.2.1 Fixed Driving

| Parameter name | Symbol | Comment |
|---------------------------|--------|--|
| Range | R | |
| Acceleration/Deceleration | A/D | When acceleration and deceleration are equal, the setting of deceleration is not required. |
| Initial Speed | SV | |
| Drive Speed | V | |
| Number of Output Pulse | P | |

■ Changing the Number of Output Pulse in Driving

The number of output pulse can be changed in the fixed driving. If the command is for increasing the output pulse, the pulse output profile is shown as Fig. 2.2 or 2.3. If the command is for decreasing the output pulses, the output pulse will be stopped immediately as shown in Fig. 2.4. Furthermore, when in the S-curve acceleration/deceleration driving mode, the output pulse number change will occur to an incomplete deceleration S-curve.

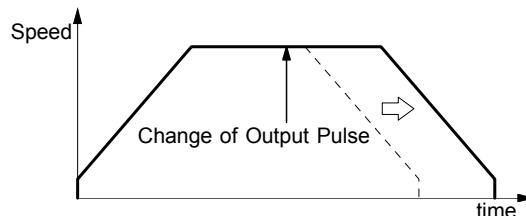


Fig.2.2 Change of Output Pulse Number in Driving

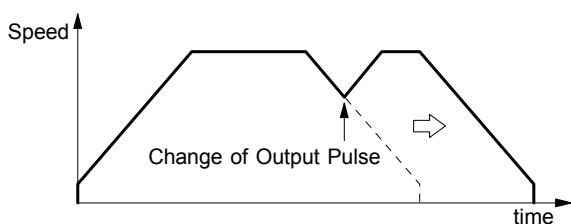


Fig. 2.3 Changing The Number of Output Pulse During Deceleration

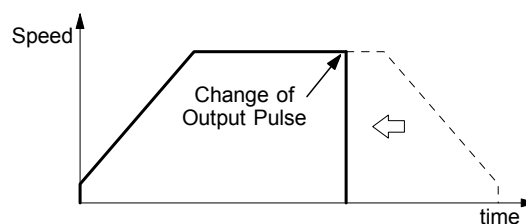


Fig. 2.4 Changing The Pulse Number Less Than Output Pulse Number

■ Manual Setting Deceleration for fixed Acceleration/Deceleration Driving

As shown in Fig. 2.1, generally the deceleration of fixed acceleration /deceleration driving is controlled automatically by MCX312. However, in the following situations, it should be preset the deceleration point by the users.

- The change of speed is too often in the trapezoidal fixed acceleration/deceleration driving.
- When use circular interpolation, bit pattern interpolation and continuous interpolation for acceleration and deceleration.

In case of manual deceleration, please set D0 bit of register WR3 to 1, and use command (07h) for presetting deceleration point. As to the other operation, the setting is as same as that of fixed driving.

■ Offset Setting for Acceleration/Deceleration Driving

The offset function can be used for compensating the pulses when the decelerating speed does not reach the setting initial speed during the S-curve fixed driving. MCX312 will calculate the acceleration / deceleration point automatically, and will arrange the pulse numbers in acceleration equal to that in deceleration. The method is calculating the output acceleration pulses and comparing them with the remaining pulses. When the remaining pulses are equal to or less the pulses in acceleration, it starts the deceleration.

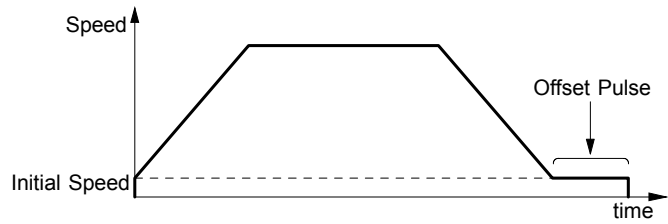


Fig. 2.5 Offset for Deceleration

When setting the offset for deceleration, MCX312 will start deceleration early for the offset. The greater is the positive value set for the offset, the closer the automatic declaration point becomes, increasing the creep pulses at the initial speed at deceleration termination. If a negative value is set for the offset value, output may stop prematurely before the speed reaches the initial speed (see Fig. 2.6).

The default value for offset is 8 when MCX312 power-on reset. It is not necessary to change the shift pulse value in the case of acceleration/deceleration fixed driving. As for fixed driving in non-symmetrical trapezoidal acceleration/deceleration or S-curve acceleration/deceleration, if creep pulses or premature termination occurs at termination of driving due to the low initial speed setting, correct the speed by setting the acceleration counter offset to an appropriate value.

2.1.2 Continuous Driving Output

When the continuous driving is performed, MCX312 will drive pulse output in a specific speed until stop command or external stop signal is happened. The main application of continuous pulse driving is: home searching, teaching or speed control. The drive speed can be changed freely during continuous driving.

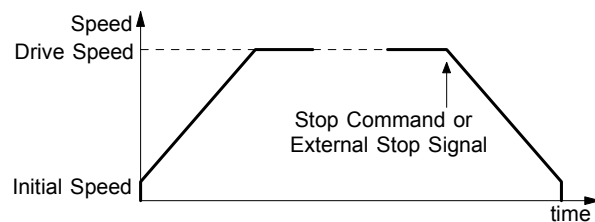


Fig. 2.6 Continuous Driving

Two stop commands are for stopping the continuous driving. One is “decelerating stop”, and the other is “sudden stop”. Three input pins, STOP2~STOP0, of each axis can be connected for external decelerating and sudden stop signals. Enable / disable, active levels and mode setting are possible.

■ Stop Condition for External Input STOP2 to STOP0 in Continuous Driving

Assign an encoder Z-phase signal, a home signal, and a near home signal in nSTOP2 to nSTOP0. (Assign an encoder Z phase signal in nSTOP2.) Enable / disable and logical levels can be set by bit from D5 to 0 of WR1 register of each axis. For the application of high-speed searching, the user can set MCX312 in the acceleration/deceleration continuous driving mode and enable STOP2,1,0 in WR1. And then, MCX312 will perform the decelerating stop when the external signal STOP2,1,0 is active.

For the application of low-speed searching, the user can set MCX312 in the constant-speed continuous driving and enable STOP2,1,0. Then, MCX312 will perform the sudden stop when STOP1 is active.

Except the parameter of the number of output pulse, the other three parameters for the fixed drive must be set to execute the acceleration/deceleration continuous driving.

2.2 Acceleration and Deceleration

Basically, driving pulses of each axis are output by a fixed driving command or a continuous driving command of the + direction or – direction. These types of driving can be performed with a speed curve of constant speed, linear acceleration, non-symmetrical linear acceleration, S-curve acceleration/deceleration according to the mode that is set or the operation parameter value.

2.2.1 Constant Speed Driving

When the drive speed set in MCX312 is lower than the initial, the acceleration / deceleration will not be performed, instead, a constant speed driving starts.

If the user wants to perform the sudden stop when the home sensor or encoder Z-phase signal is active, it is better not to perform

the acceleration / deceleration driving, but the low-speed constant driving from the beginning.
 For processing constant speed driving, the following parameters will be preset accordingly.

| Parameter name | Symbol | Comment |
|------------------------|--------|--|
| Range | R | |
| Initial Speed | SV | Set a value higher than the drive speed (V). |
| Drive Speed | V | |
| Number of Output Pulse | P | Not required for continuous driving. |

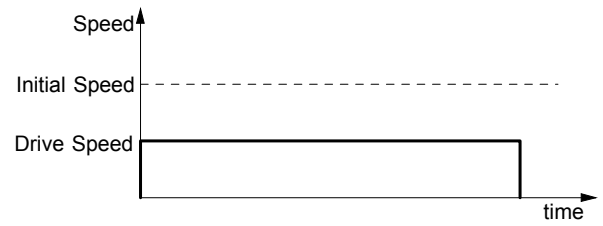
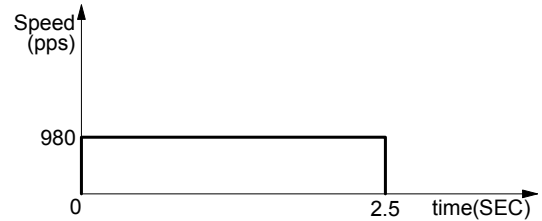


Fig. 2.7 Constant Speed Driving

■ Example for Parameter Setting of Constant Speed

The constant speed is set 980 PPS as shown in the right Figure.

- Range R = 8,000,000 ; Multiple = 1
- Initial Speed SV=980 ; Initial Speed ≥ Drive Speed
- Drive Speed V=980 ; Should be less than initial speed



Please refer each parameter in Chapter 6.

2.2.2 Trapezoidal Driving [Symmetrical]

In linear acceleration driving, the drive speed accelerates in a primary linear form with the specified acceleration slope from the initial speed at the start of driving. When the acceleration and the deceleration are the same (symmetrical trapezoid) in fixed driving, the pulses utilized at acceleration are counted. When the remaining number of output pulses becomes less than the number of acceleration pulses, deceleration starts. Deceleration continues in the primary line with the same slope as that of acceleration until the speed reaches the initial speed and driving stops, at completion of the output of all the pulses (automatic deceleration).

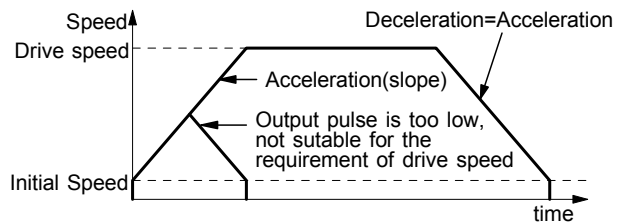


Fig. 2.8 Trapezoidal Driving (Symmetrical)

When the decelerating stop command is performed during the acceleration, or when the pulse numbers of the fixed drive do not reach the designated drive speed, the driving will be decelerating during acceleration, as show in Fig. 2.9. By setting a triangle prevention mode, such triangle form can be transformed to a trapezoid form even if the number of output pulses low. See the section of triangle prevention of fixed driving.

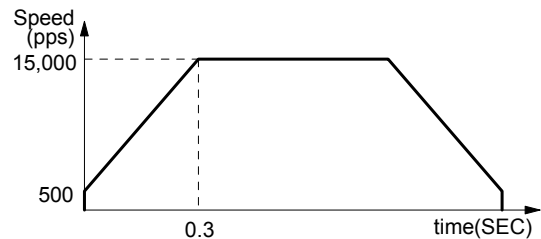
To perform symmetrical linear acceleration driving, the following parameters must be set, parameters marked by ○ will be set when needed.

| | Parameter name | Symbol | Comment |
|---|------------------------|--------|---|
| | Range | R | |
| | Acceleration | A | Acceleration and deceleration. |
| ○ | Deceleration | D | Deceleration when acceleration and deceleration are set individually. |
| | Initial Speed | SV | |
| | Drive Speed | V | |
| ○ | Number of Output Pulse | P | Not required for continuous driving. |

■ The example of setting Trapezoidal Driving

Shown in the figure right hand side, acceleration is form the initial speed 500 PPS to 15,000 PPS in 0.3 sec.

Range R = 4,000,000 ; Multiple= 2
 Acceleration A=193 ; (15,000-500)/0.3 =48,333
 ; 48,333/125/M = 193
 Initial Speed SV = 250 ; 500/M = 250
 Drive Speed V = 7,500 ; 15,000/M = 7,500



Please refer Chapter 6.

■ Triangle Prevention of Fixed Driving

The triangle prevention function prevents a triangle form in linear acceleration fixed driving even if the number of output pulses is low. When the number of pulses that were utilized at acceleration and deceleration exceeds 1/2 of the total number of output pulses during acceleration, this IC stops acceleration and enters a constant speed mode.

The triangle prevention function is disabled at resetting. The function can be enabled by setting bit D5 to 1 of the WR3 register.

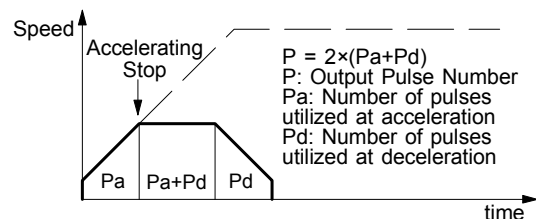


Fig. 2.9 Triangle Prevention of Linear Acceleration Driving

[Note] When continuous driving or circular interpolation driving is performed after fixed driving, WR3 /D5 bit must be reset 0 in advance.

2.2.3 Non-Symmetrical Trapezoidal Acceleration

When an object is to be moved using stacking equipment, the acceleration and the deceleration of vertical transfer need to be changed since a gravity acceleration is applied to the object.

This IC can perform automatic deceleration in fixed driving in non-symmetrical linear acceleration where the acceleration and the deceleration are different. It is not necessary to set a manual deceleration point by calculation in advance. Fig. 2.11 shows the case where the deceleration is greater than the acceleration and Fig. 2.12 shows the case where the acceleration is greater than the deceleration. In such non-symmetrical linear acceleration also, the deceleration start point is calculated within the IC based on the number of output pulses P and each rate parameter.

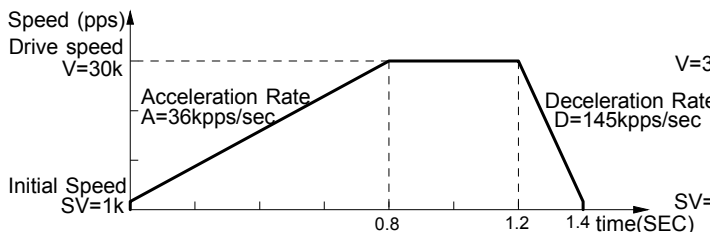


Fig.2.10 Non - Symmetrical Linear Acceleration Driving (acceleration<deceleration)

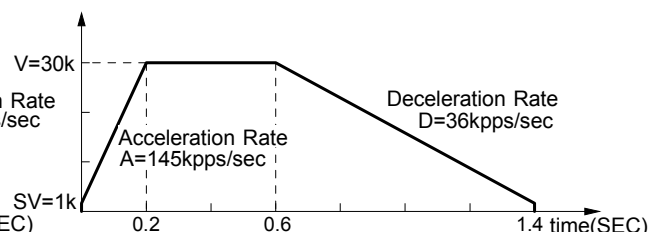


Fig.2.11 Non - Symmetrical Linear Acceleration Driving (acceleration>deceleration)

To perform automatic deceleration for fixed driving of non-symmetrical linear acceleration, bit D1 (DSNDE) to 1 of the WR3 register must be set to apply deceleration-setting value, and bit D0 (MANLD) to 0 of the WR3 register must be set to enable automatic deceleration during acceleration/deceleration driving.

| Mode setting bit | Symbol | Setting value | Comment |
|------------------|--------|---------------|--|
| WR3/D1 | DSNDE | 1 | The deceleration setting value is applied at deceleration. |
| WR3/D0 | MANLD | 0 | Automatic deceleration |

The following parameters must be set.

| Parameter name | Symbol | Comment |
|-------------------------|--------|------------------------------------|
| Range | R | |
| Acceleration | A | |
| Deceleration | D | |
| Initial speed | SV | |
| Drive speed | V | |
| Number of output pulses | P | Not required at continuous driving |

[Note]

- In the case of acceleration > deceleration (Fig. 2.12), the following condition is applied to the ratio of the acceleration and the deceleration.

$$D > A \times \frac{V}{4 \times 10^6}$$

D: Deceleration rate (pps/sec)
 A: Acceleration rate (pps/sec) Where CLK=16MHz
 V: Drive Speed (pps)

For instance, if the driving speed V = 100kps, deceleration D must be greater than 1/40 of acceleration A. The value must not be less than 1/40 of the acceleration.

- If acceleration > deceleration (Fig. 2.12), the greater the ratio of acceleration A to deceleration D becomes, the greater the number of creep pulses becomes (about maximum of 10 pulse when A/D=10 times). When creep pulses cause a problem, solve the problem by ①increasing the initial speed or ②setting a minus value to the acceleration counter offset.

2.2.4 S-curve Acceleration/Deceleration Driving

This IC creates an S curve by increasing/reducing acceleration/decelerations in a primary line at acceleration and deceleration of drive speed. Figure 2.13 shows the operation of S-curve acceleration/deceleration. When driving starts, the acceleration increases on a straight line at the specified jerk (K). In this case, the speed data forms a secondary parabolic curve (section a). When acceleration reaches designation value (A), acceleration is maintained. In this case, the speed data forms an increase on a straight line (section b).

If the difference between the specified drive speed (V) and the current speed becomes less than the speed that was utilized at the increase of acceleration, the acceleration starts to decrease towards 0. The decrease ratio is the same as the increase ratio and the acceleration decreases in a linear form of the specified jerk (K). In this case, the speed data forms a secondary parabolic curve (section c). Thus, the case that acceleration has a constant part in its acceleration, this book calls it The Partial S curve Acceleration.

On the other hand, if the difference between the specified drive speed (V) and the current speed becomes less than the speed that was utilized at the increase of acceleration before acceleration reaches designation value (A), section shifts from a to c without b section. Thus, the case that acceleration does not have a constant part in its acceleration, it calls The Perfect S curve Acceleration.

Please refer to example of parameter settings described later and appendix regarding cases of the partial S curve acceleration and the perfect S curve acceleration.

Also at the deceleration, the speed forms an S curve by increasing/decreasing the deceleration in a primary linear form (sections d, e and f).

The same operation is performed in acceleration/deceleration where the drive speed is changed during continuous driving.

To perform S curve acceleration/deceleration driving, set bit D2 to 1 of the nW3 register and parameters as follows, parameters marked by ○ will be set when needed.

| Parameter name | Symbol | Comment |
|--------------------------|--------|---|
| Range | R | |
| Jerk | K | |
| Acceleration | A | Acceleration/deceleration increases from 0 to the value linearly. |
| ○ Deceleration | D | Deceleration when acceleration and deceleration are set individually. |
| Initial Speed | SV | |
| Drive Speed | V | |
| ○ Number of Output Pulse | P | Not required for continuous driving. |

■The Prevention of Triangle Driving Profile

For fixed driving of linear acceleration/deceleration, the speed curve forms the triangle form when the output pulses do not reach the pulses required for accelerating to the drive speed or deceleration stop is applied during acceleration. In the case of S curve acceleration/deceleration driving, the following method is applied to maintain a smooth speed curve.

If the initial speed is 0, and if the rate of acceleration is a, then the speed at time t in acceleration region can be described as following.

$$v(t) = at^2$$

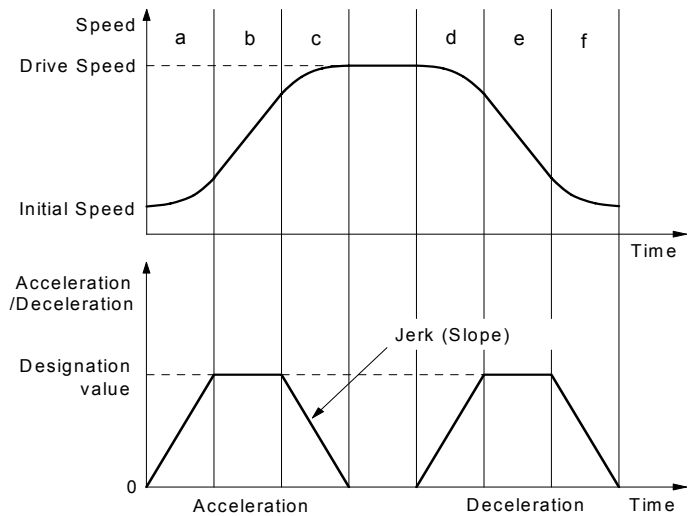


Fig.2.12 S-Curve Acceleration/Deceleration Driving

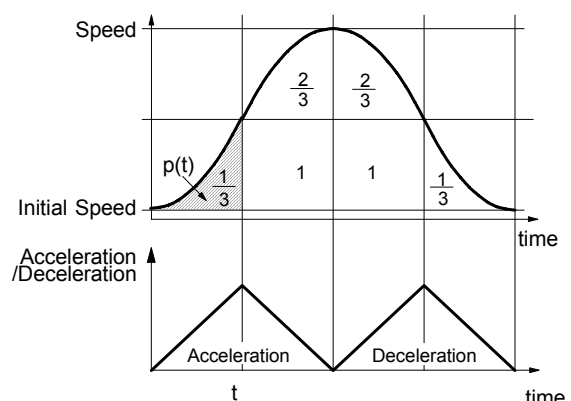


Fig. 2.13 The rule of 1/12 of Parabolic Acceleration/Deceleration

Therefore, the total the number of pulse p(t) from time 0 to t is the integrated of speed.

$$p(t) = 1/3 \times at^3$$

The total output pulse is

$$(1/3+2/3+1+2/3+1+1/3) \times at^3 = 4 at^3$$

so

$$p(t) = 1/12 \text{ (total pulse output)}$$

Therefore, when the output pulse in acceleration of S-curve is more than 1/12 of total output pulse, MCX312 will stop increasing acceleration and start to decrease the acceleration value.

In the constant acceleration part, when the output pulse in acceleration reaches 4/1 of total output pulse, MCX312 will start to decrease the acceleration value.

■ The Decelerating Stop for Preventing the Triangle Driving Profile

When the decelerating stop is commanded during the acceleration / deceleration driving, the acceleration is decreasing, then the deceleration starts when the acceleration reaches 0.

■ Constraints for S-curve Acceleration / Deceleration Driving

- a. The drive speed cannot be changed during the fixed S-curve acceleration / deceleration driving.
- b. When the fixed S-curve acceleration / deceleration driving is performed, the change of the numbers of output pulse during the deceleration will not result a normal S-curve driving profile.
- c. In case of executing circular interpolation, bit pattern interpolation and continuous interpolation, S-curve acceleration/deceleration cannot be executed normally.
- d. If an extremely low value is set as the initial speed for fixed driving of S-curve acceleration/deceleration, premature termination (output of the specified driving pulses is completed and terminated before the speed reaches the initial speed) or creep (output of specified driving pulses is not completed even if the speed reaches the initial speed and the remaining driving pulses are output at the initial speed) may occur. Set initial speed value (SV) more than 100.
- e. When the fixed S-curve acceleration / deceleration driving is performed, the driving speed does not seldom reach the setting value.

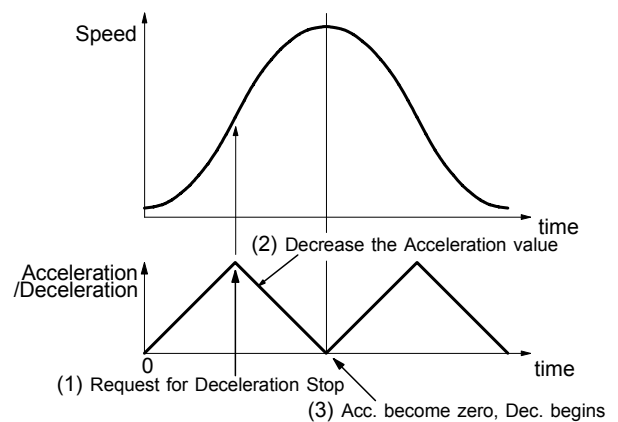


Fig. 2.14 The rule of 1/12 of Parabolic Acceleration/Deceleration

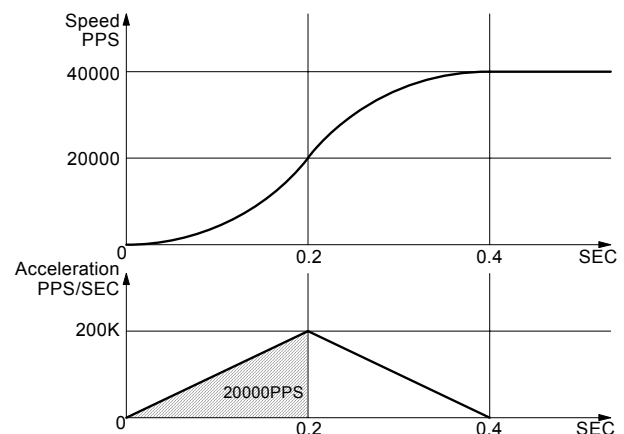
■ Example of Parameter Setting 1 (Perfect S-Curve Acceleration/Deceleration)

As shown in the diagram, in this example, the perfect S curve acceleration is applied to reach from the initial speed of 0 to 40KPPS in 0.4 seconds.

The speed must be 20,000PPS (half of 40,000PPS) in 0.2 sec (half of 0.4 sec) and then must reach to 40,000PPS in rest of 0.2 sec. At this time, the acceleration increases on a straight line in 0.2 sec and the integral value is equal to the starting speed 20,000PPS.

Therefore, the acceleration at 0.2 sec is $20,000 \times 2 / 0.2 = 200KPPS/SEC$ and the jerk is $200K / 0.2 = 1,000KPP/SEC^2$.

For the perfect S curve, the speed curve only depends on the jerk so that the value of acceleration/deceleration must be set greater than 200KPPS/SEC not to be the partial S curve.



Range R = 800000 ; Multiple=10
 Jerk K =625 ; $((62.5 \times 10^5) / 625) \times 10 = 1000 \times 10^3 \text{ PPS/SEC}^2$

Acceleration A = 160 ; $125 \times 160 \times 10 = 200 \times 10^3$ PPS/SEC
 Initial Speed SV = 100 ; $100 \times 10 = 1000$ PPS
 Drive Speed V = 4000 ; $4000 \times 10 = 40000$ PPS

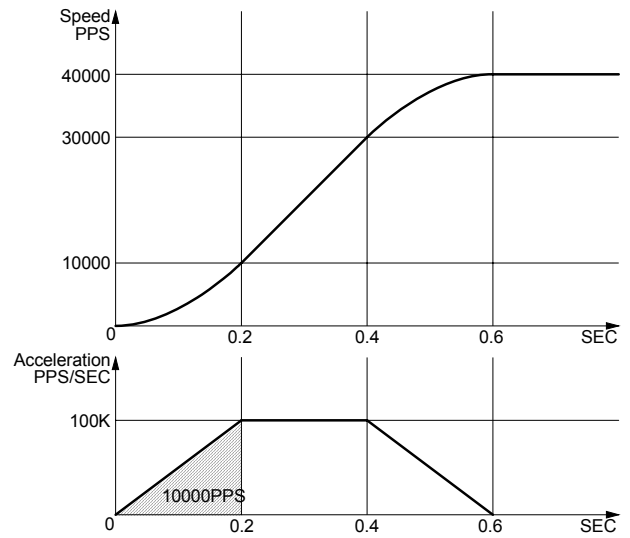
Please refer each parameter in Chapter 6.

■ Example of Parameter Setting 2 (Partial S-Curve Acceleration/Deceleration)

As shown in the diagram, in this example, the partial S curve acceleration is applied, firstly it reaches from initial speed of 0 to 10KPPS in 0.2 seconds by parabolic acceleration and then reaches from 10KPPS to 30KPPS in 0.2 sec by acceleration on a straight line, finally reaches from 30KPPS to 40KPPS in 0.2 sec by parabolic acceleration.

The first acceleration must increase up to 10,000PPS in 0.2 sec on a straight line. At this time, the integral value is equal to the rising speed 10,000PPS. Therefore, the acceleration at 0.2 sec is $10,000 \times 2 / 0.2 = 100\text{KPPS/SEC}$ and the jerk is $100\text{K} / 0.2 = 500\text{KPP/SEC}^2$.

Range R = 800000 ; Multiple=10
 Jerk K =1250 ; $((62.5 \times 10^6) / 1250) \times 10 = 500 \times 10^3$ PPS/SEC²
 Acceleration A = 80 ; $125 \times 80 \times 10 = 100 \times 10^3$ PPS/SEC
 Initial Speed SV = 100 ; $100 \times 10 = 1000$ PPS
 Drive Speed V = 4000 ; $4000 \times 10 = 40000$ PPS



2.2.5 Pulse Width and Speed Accuracy

■ Duty Ratio of Drive Pulse

The period time of + /- direction pulse driving of each axis is decided by system clock SCLK. The tolerance is within $\pm 1SCLK$ (For $CLK=16MHz$, the tolerance is $\pm 125nSEC$). Basically, the duty ratio of each pulse is 50% as show in Fig. 2.16. When the parameter setting is $R=8,000,000$ and $V=1000$ (Multiple=1, $V=1000PPS$), the driving pulse is 500uSEC on its Hi level and 500uSEC on its Low level and the period is 1mSEC.

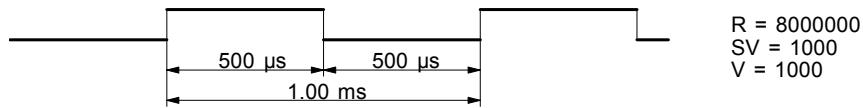


Fig. 2.15 High/Low Level Width of Driving Pulse Output ($V=1000PPS$)

However, during the acceleration / deceleration driving, the Low level pulse length is shorter than that of Hi level pulse during the acceleration; the Low level pulse is longer than that of Hi level pulse during the deceleration. See Fig. 2.17.

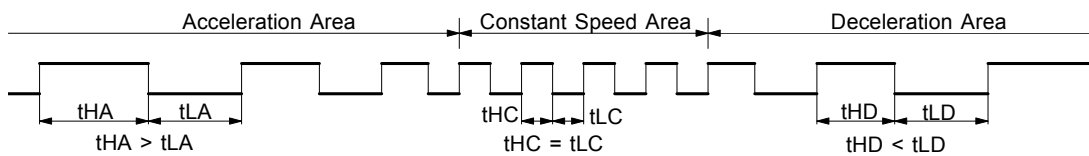


Fig. 2.16 Comparison of Drive Pulse Length in Acceleration/Deceleration

■ The Accuracy of Drive Speed

The clock (SCLK) running in MCX312 is half of external input clock (CLK). If CLK input is standard 16MHz, SCLK will be 8MHz. Therefore, the user had better driving the pulse speed in an exact multiple of SCLK period (125nSEC). Otherwise, the driving pulse will not very stable. The frequency (speed) of driving pulse of MCX312 can be, there are all exact the multiple of 125nSEC. For instance, the only frequencies that can be output are, double: 4.000 MHz, triple: 2.667 MHz, quadruple: 2.000 MHz, five times: 1.600 MHz, six times: 1.333 MHz, seven times: 1.143 MHz, eight times: 1.000 MHz, nine times: 889 KHz, 10 times: 800 KHz, Any fractional frequencies cannot be output. It is not very stable to set any desired drive speed. However, MCX312 can make any drive speed in using the following method.

For instance, in the case of the range setting value: $R=80,000$ (magnification = 100) and drive speed setting value: $V=4900$, the speed of driving pulses of $4900 \times 100 = 490 KPPS$ is output. Since this period is not a multiple integer of the SCLK period, pulses of 490KPPS cannot be output under a uniform frequency. Therefore, as shown in Fig. 2.18, MCX312 combines 16 times and 17 times of SCLK period in a rate of 674:326 to generate an average 490KPPS.



Fig. 2.17 The Driving Pulse of 490KPPS

According to this method, MCX312 can generate a constant speed driving pulse in a very high accuracy. In general, the higher of the drive speed, the lower of the accuracy. But for MCX312, it still can maintain relative accuracy when the drive speed is high. Actually, the accuracy of driving pulse is still within $\pm 0.1\%$.

Using oscilloscope for observing the driving pulse, we can find the jitter about 1SCLK (125nSEC). This is no matter when putting the driving to a motor because the jitter will be absorbed by the inertia of motor system.

2.3 Position Control

Fig 2.19 is 1-axis position control block diagram. For each axis, there are two 32 bit up-and-down counters for counting present positions and two comparison registers for comparing the present positions.

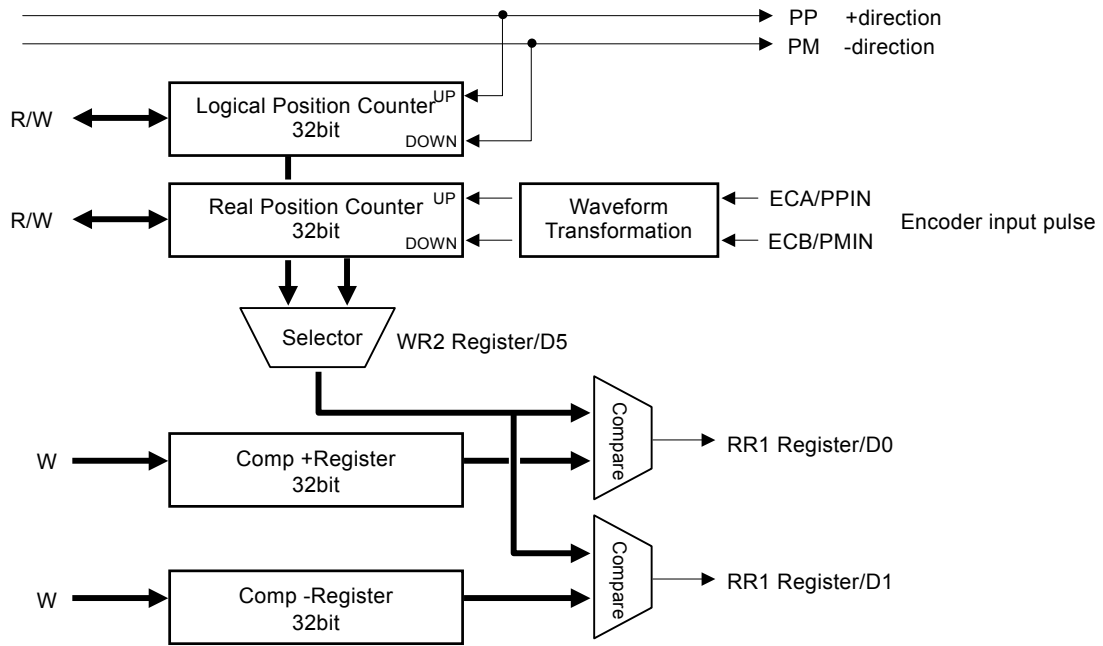


Fig. 2.18 Position Control Block Diagram

2.3.1 Logic Position Counter and Real position Counter

As shown above in Fig. 2.19, the logic position counter is counting the driving pulses in MCX312. When one + direction pulse is outputting, the counter will count-up 1; when one - direction pulse is outputting, the counter will count-down 1. The real position counter will count input pulse numbers from external encoder. The type of input pulse can be either A/B quadrature pulse type or Up / Down pulse (CW/CCW) type (See Chapter 2.6.3).

Host CPU can read or write these two counters any time. The counters are signed 32 bits, and the counting range is between -2,147,483,648 ~ + 2,147,483,647. The negative is in 2’s complement format. The counter value is random while resetting.

2.3.2 Compare Register and Software Limit

Each axis has, as shown in Fig. 2.19, two 32-bit registers which can compare the logical positions with the real positions. The logical position and real position counters are selected by bit D5 (CMPSL) of WR2 register.

The main function of COMP+ Register is to check out the upper limit of logical / real position counter. When the value in the logical / real position counters are larger than that of COMP+ Register, bit D0 (CMP+) of register RR1 will become 1. On the other hand, COMP- Register is used for the lower limit of logical / real position counter. When the value of logical / real position counter become smaller than that of COMP+ Register, bit D1 (CMP-) of register RR1 will become 1. Fig. 2.20 is an example for COMP+ = 10000, COMP- = -1000, COMP+ and COMP- registers can be used as software +/- limit.

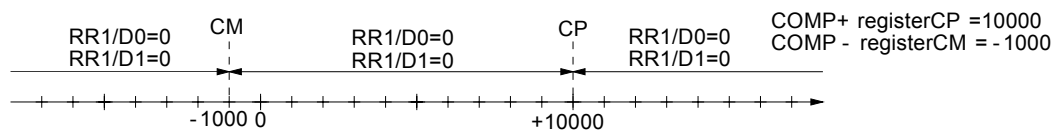


Fig. 2.19 Example of COMP+/- Register Setting

When D0 and D1 bits of WR2 register are set to 1, it enables the software limit. In driving, if the value of logical / real counter is larger than COMP+, the decelerating stop will be performed, and D0 (SLMT+) of RR2 register will change to 1. If the value of logical / actual counter is smaller than that of COMP+, the D0 bit of RR2 register will change to 0 automatically.

Host CPU can write the COMP+ and COMP- registers any time. However, when MCX312 is reset, the register values are random.

2.3.3 Position Counter Variable Ring

A logical position counter and a real position counter are 32-bit up/down ring counters. Therefore, normally, when the counter value is incremented in the + direction from FFFFFFFFh, which is the maximum value of the 32-bit length, the value is reset to the value 0. When the counter value is decremented in the – direction from the value 0, the value is reset to FFFFFFFFh. The variable ring function enables the setting of any value as the maximum value. This function is useful for managing the position of the axis in circular motions that return to the home position after one rotation, rather than linear motions.

To enable the variable ring function, set the D6 (RING) bit of the WR3 register to 1 and set the maximum value of the logical position counter in the COMP+ register and the maximum value of the real position counter in the COMP– register.

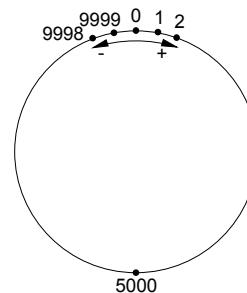


Fig. 2.20 Operation of Position Counter Ring Maximum Value 9999

For instance, set as follows for a rotation axis that rotates one cycle with 10,000 pulses.

- ① To enable the variable ring function, set 1 in the D6 bit of the WR3 register.
- ② Set 9,999 (270Fh) in the COMP+ register as the maximum value of the logical position counter.
- ③ Set 9,999 (270Fh) in the COMP– register when using a real position counter also.

The count operation will be as follows.

Increment in the + direction ...→9998→9999→0→1→...
 Decrement in the - direction ...→1→0→9999→9998→...

[Notes]

- The variable ring function enable/disable is set for each axis, however, a logical position counter and a real position counter cannot be enabled/disabled individually.
- If a variable ring function is enabled, a software limit function cannot be used.

2.3.4 Clearing a Real Position Counter Using an External Signal

This function clears a real position counter at rising of the Z-phase active level when Z-phase search is applied in home search.

Normally, home search is performed by assigning a near home signal, a home signal, and an encoder Z-phase signal to nSTOP0 to nSTOP2 signals and executing continuous driving. When the specified signal is activated, driving will stop and then the logical position/real position counters are cleared by the CPU. This function is useful for solving the problem of Z-phase detection position slippage that occurs due to a delay of the servo system or the mechanical system even if a low Z-phase search drive speed is set.

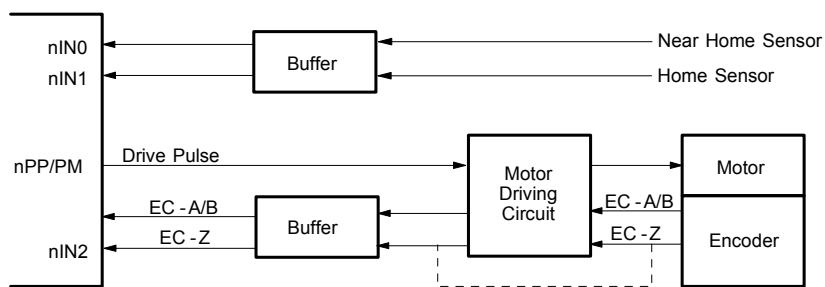


Fig. 2.21 Example of Signal Connection for Clearing The Real Position Counter by The STOP2 Signal

To clear a real position counter with a Z-phase signal in encoder Z-phase search, assign the Z-phase signal to nSTOP2 signal as shown Fig. 2.22. See below for the procedure for setting a mode or a command for Z-phase search accompanied by clearing of the real position counter.

- ① Set a range and an initial speed.
- ② Set a Z-phase search drive speed.
 If the value set for the drive speed is lower than the initial speed, acceleration/deceleration driving is not performed. If a Z-phase is detected, the driving pulse stops immediately.

- ③ Validate the STOP2 signal and set an active level.
WR1/D5(SP2-E) : 1, D4(SP2-L) : 0(Low active) 1(Hi active)
- ④ Enable the clearing of the real position counter using the STOP2 signal.
Set WR1/D6 to 1
- ⑤ Issue the + direction or - direction continuous driving command.

As a result of the operations described above, driving starts in the specified direction as shown in Fig. 2.23. When the Z-phase signal reaches an active level, the driving pulses stop and the real position counter is cleared at the rising of the Z-phase signal active level.

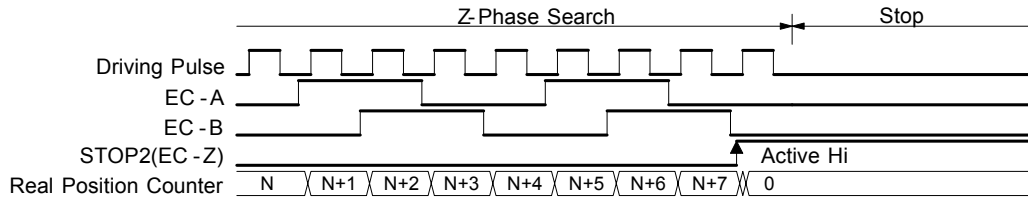


Fig. 2.22 Example of Operation of Clearing The Real Position Counter Using The STOP2 Signal

[Notes]

- Only the nSTOP2 signal can clear the real position counter. The nSTOP1 and nSTOP0 signals cannot clear the counter.
- When the input signal filter is invalid, an active level width of more than 4CLK cycles is necessary. When the input signal filter is valid, a time more than double the input signal delay time is necessary.
- It is recommended to perform Z-phase search from the one direction to enhance the position detection precision.
- When the nSTOP2 signal is already set to an active level at setting WR1/D6, 5, 4, the real position counter is cleared even if WR1/D6, 5, 4 is set.

[Notes] Input signal filter must be enabled to nSTOP2 signal when the real position counter uses. If input signal is passed through, the real position counter may be not cleared.

2.4 Interpolation

This 2-axis motion control IC can perform linear interpolation, circular interpolation and bit pattern interpolation.

In the process of interpolation driving, all the calculations will follow the X axis. So, the user has to set the parameters such as initial speed and drive speed of the X axis before performing the interpolation.

After setting all of the parameters for interpolations, and writing the interpolation driving commands to command register WR0, the user can start the interpolation driving. During the interpolation driving, D8 (I-DRV) of main status register RR0 will become 1 during the interpolation, and it will become 0 when the interpolation is finished. Also, during the interpolation driving, the bit D1 (Y-DRV) and D0 (X-DRV) will become 1.

The maximum drive speed is 4MPPS for linear, circular or bit pattern interpolation. For continuous interpolation, the maximum drive speed is 2MPPS.

Over Limit Error of Interpolation

When the hardware limit or the software limit of each axis is active during the interpolation driving, the interpolation will stop. If the stop is occurred by errors, RR0 (main status register) will confirm the error bit in D5 (Y-ERR) and D4 (X-EWW). RR0 will become 1, and RR2 (error register) of this axis will be read out.

[Note] In case of circular or bit patter interpolation, the “ active” of hardware or software limit, in either + or - direction, will stop the interpolation driving.

In-position Signal for Servo Motor

During the interpolation driving, when the in-position signal (nINPOS) of each X and Y axis is active, and also when the interpolation is finished, the INPOS signal of the axis is stand-by at its active level, and D8 (I-DRV) of RR0 register returns to 0.

2.4.1 Linear Interpolation

To execute the linear interpolation, the user can, according to the present point coordinates, set the finish point coordinates and the interpolation segment(s) for 2 axes. Fig. 2.24 shows an example of axis interpolation where linear interpolation is performed from the current coordinates to the finish point coordinates.

For individual axis control, the command pulse number is unsigned, and it is controlled by + direction command or – direction command. For interpolation control, the command pulse number is signed.

The resolution of linear interpolation is within ± 0.5 LSB, as shown in Fig. 2.25

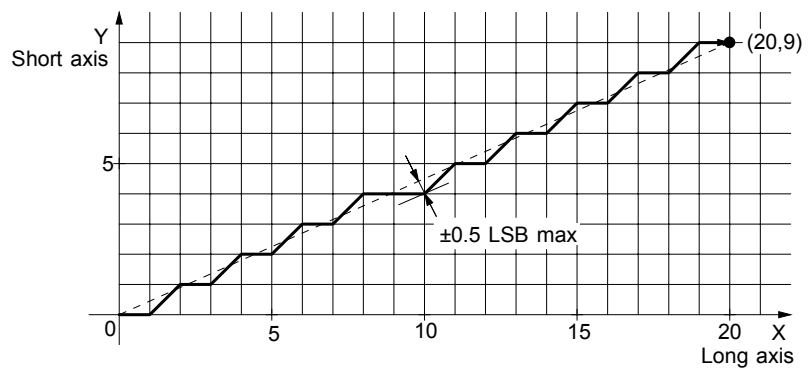


Fig. 2.23 The Position Accuracy for Linear Interpolation

As shown in Fig. 2.25, it is an example for pulse output of the linear interpolation driving. We define the longest distance movement in interpolation is the “long axis”. And the other is “short axis”. The long axis outputs an average pulse train. The driving pulse of the short axis depends on the long axis and the relationship of the two axes.

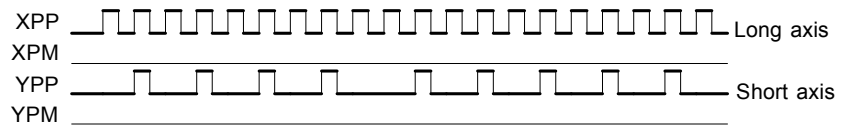


Fig. 2.24 The Example for Pulse Output at Finish Point (X=20, Y=9)

The range for each axis is a 24-bit signed counter, from -8,388,607 ~ + 8,388,607. (Notes: -8,388,607 cannot be set.)

■ The example of linear interpolation for 2 axes

Executing linear interpolation drives in X and Y axes from the current position to the finish position (X: +30,000, Y: -20,000). The initial speed = 500PPS, acceleration / deceleration = 40,000PPS/SEC, drive speed = 5,000PPS.

```

WR6 ← 1200h write      ; range: 8,000,000 (Multiple = 1)
WR7 ← 007Ah write
WR0 ← 0100h write

WR6 ← 0140h write      ; accel./decel. speed :40,000PPS/SEC
WR0 ← 0102h write      ; 40,000 / 125 / 1 = 320

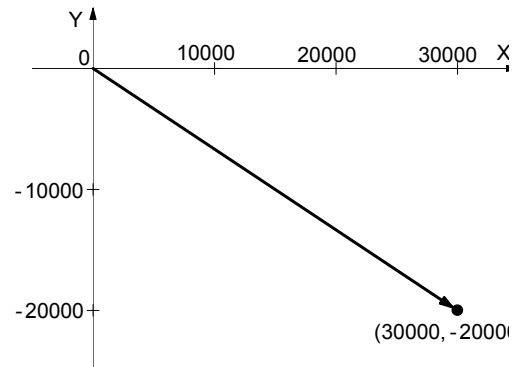
WR6 ← 01F4h write      ; initial speed :500PPS
WR0 ← 0104h write

WR6 ← 1388h write      ; drive speed :5,000PPS
WR0 ← 0105h write

WR6 ← 7530h write      ; finish point of X axis: 30,000
WR7 ← 0000h write
WR0 ← 0106h write

WR6 ← B1E0h write      ; finish point of Y axis: -20,000
WR7 ← FFFFh write
WR0 ← 0206h write

WR0 ← 003Bh write      ; deceleration enabling
WR0 ← 0030h write      ; linear interpolation driving for 2 axes enabling
    
```



2.4.2 Circular Interpolation

2 axes X and Y are for circular interpolation.

The circular interpolation is starting from the current position (start point). After setting the center point of circular, the finish position and the CW or CCW direction, the user can start the circular interpolation.

Note: The coordinates setting value is the relative value of the start point coordinates.

In Fig. 2.26, it explains the definition of CW and CCW circular interpolations. The CW circular interpolation is starting from the start point to the finish position with a clockwise direction; the CCW circular interpolation is with a counter-clockwise direction. When the finish point is set to (0, 0), a circle will come out.

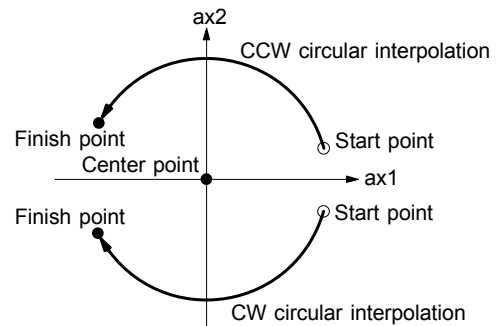


Fig. 2.25 CW / CCW Circular Interpolation

In Fig. 2.27, it explains the long axis and the short axis. First, we define 8 quadrants in the X-Y plane and put the numbers 0~7 to each quadrant. We find the absolute value of X is always larger than that of Y in quadrants 0, 3, 4 and 7, so we call X is the long axis (Y is the short axis) in these quadrants; in quadrants 1, 2, 5 and 6, Y is the long axis (X is the short axis). The short axis will output pulses regularly, and the long axis will output pulses depending on the interpolation calculation.

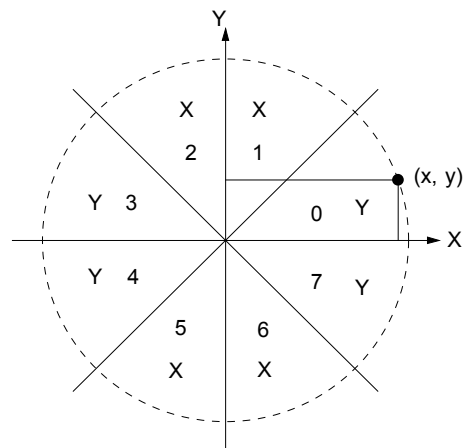


Fig. 2.26 The 0-7 Quadrants And Short

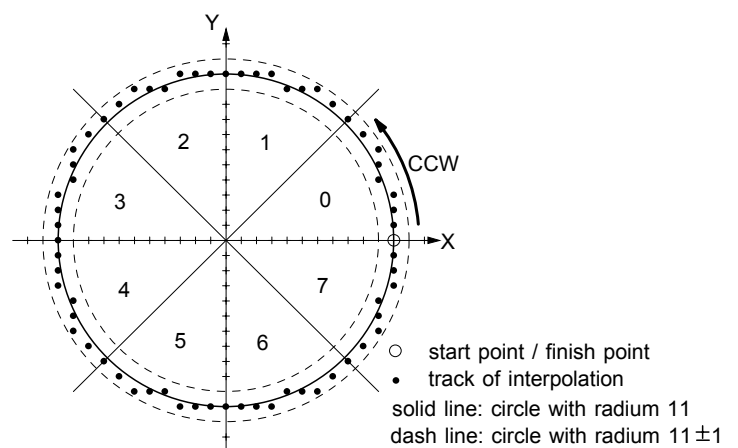


Fig. 2.27 The Example of Circular Interpolation

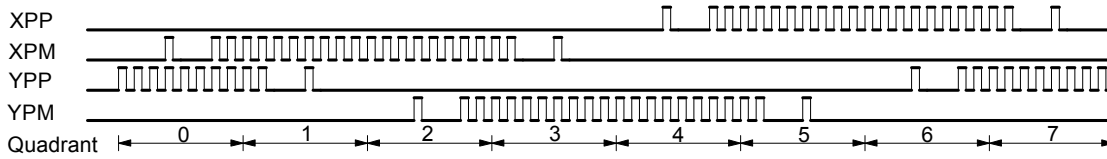


Fig. 2.28 The Example of Pulse Output in Circular Interpolation Driving

■ The Finish Point Checking of Circular Interpolation

In the circular interpolation, it assumes that the current position (start point) is (0,0). After the coordinates of the center point is set, the radius will be decided, and the circular tracking will start. The maximum error range of interpolation is within ± 1 LSB. Because of the ± 1 LSB error range, the designated finish point may not be on the circular track. When the value of finish point is same as that of short axis, this circular interpolation is finished.

Fig. 2.30 shows an example of CCW interpolation with the start point (0,0), center point (-200, 500) and finish point (-702, 299). The finish point is in quadrant 4, and Y axis is the short axis in quadrant 4. So the interpolation is finished when Y axis is 299.

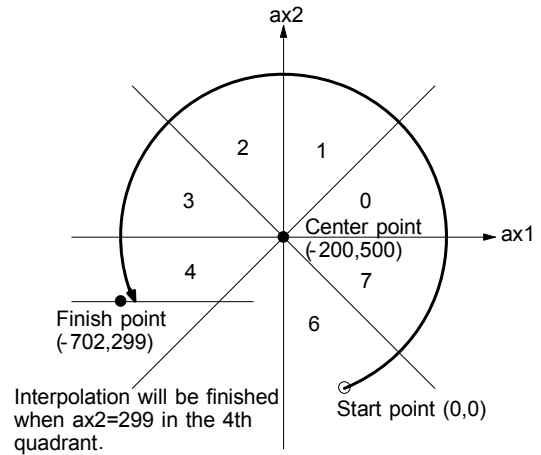


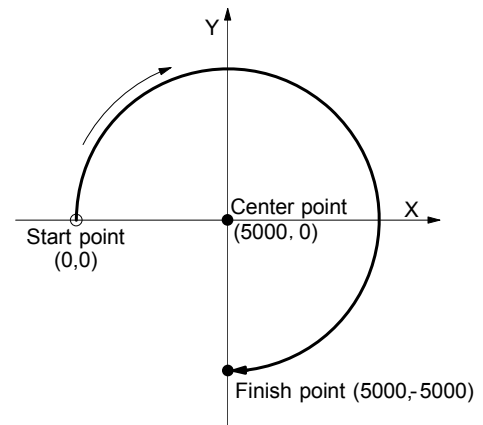
Fig. 2.29 Example of The Finish Point Checking of Circular Interpolation

The position boundary is between coordinates $-8,388,608$ and $+8,388,608$. The position tolerance for the specified circular curve is ± 1 within the entire interpolation range. The interpolation speed is within the range from 1PPS to 4MPPS.

■ The Example for CW Circular Interpolation

This CW circular interpolation starts from the current point (start point: 0, 0) to the finish point (X: 5000, Y: -5000); the center point is X: 5000, Y: 0. The interpolating speed is constant at 1000PPS in a constant vector speed driving.

- WR5 ← 0100h write ; constant linear speed
- WR6 ← 0900h write ; range : 4,000,000 (Multiple: 2)
- WR7 ← 003Dh write
- WR0 ← 0100h write
- WR6 ← 4DC0h write ; range of constant vector speed for 2 axes
- WR7 ← 0056h write ; $4,000,000 \times 1.414 = 5,656,000$
- WR0 ← 0200h write
- WR0 ← 01F4h write ; initial speed : $500 \times 2 = 1000$ PPS
- WR0 ← 0104h write
- WR6 ← 01F4h write ; drive speed : $500 \times 2 = 1000$ PPS
- WR0 ← 0105h write
- WR6 ← 1388h write ; center point of X : 5,000
- WR7 ← 0000h write
- WR0 ← 0108h write
- WR6 ← 0000h write ; center point of Y : 0
- WR7 ← 0000h write
- WR0 ← 0208h write
- WR6 ← 1388h write ; finish point of X : 5,000
- WR7 ← 0000h write
- WR0 ← 0106h write
- WR6 ← EC78h write ; finish point of Y : -5,000
- WR7 ← FFFFh write
- WR0 ← 0206h write
- WR0 ← 0032h write ; CW circular interpolation enabling



2.4.3 The Bit Pattern Interpolation

This interpolation driving receives interpolation data that is created by upper-level CPU and transformed to bit patterns in a block of a predetermined size, and outputs interpolation pulses consecutively at the specified drive speed.

Every axis has 2 bit-data buffers for host CPU: one for + direction and the other for - direction. When performing the bit pattern interpolation, the host CPU will write the designated interpolation data, for each axis X and Y, into MCX312.

If a bit in the bit pattern data from CPU is “1”, MCX312 will output a pulse at the time unit; if it is “0”, MCX312 will not output any pulse at the time unit.

For example, if the user want to generate the X-Y profile (see Fig. 2.31), the host CPU must write a set of pattern into those specific registers --- XPP: the + direction register for X axis, XPM: the - direction register for X axis, YPP and YPM: the + and - directions registers. With in the time unit, MCX312 will check the registers once and decide to output a pulse or not depending on the bit pattern.

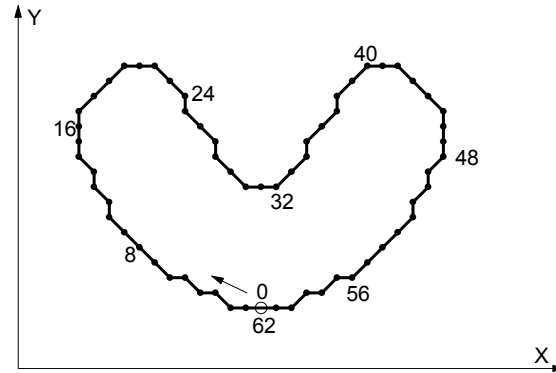


Fig. 2.30 Example for Bit Pattern Interpolation

```

←56 ←48 ←40 ←32 ←24 ←16 ←8 ←0
01000000 00000000 00011111 11011011 11110110 11111110 00000000 00000000 :XPP(X+direction)
01111111 11110101 00000000 00000000 00000000 00000000 00101011 11111111 :XPM(X-direction)
00000000 00000000 00000000 11111111 00000000 00001111 11111111 11010100 :YPP(Y+direction)
00001010 11111111 11111100 00000000 00111111 11000000 00000000 00000000 :YPM(Y-direction)
    
```

Fig. 2.32 shows the register configuration of the X axis and movements of bit data of bit pattern interpolation in this IC. BP1P register and BP1M register are 16 bit-data buffers for bit pattern data form the host CPU. (If the system uses 8-bit data bus, the host CPU has to write the data by low byte and high byte.) The + direction data should be written into BP1P, and the - direction data into BP1M. Once starting the bit pattern interpolation, the pulse outputting is in the order from D0.

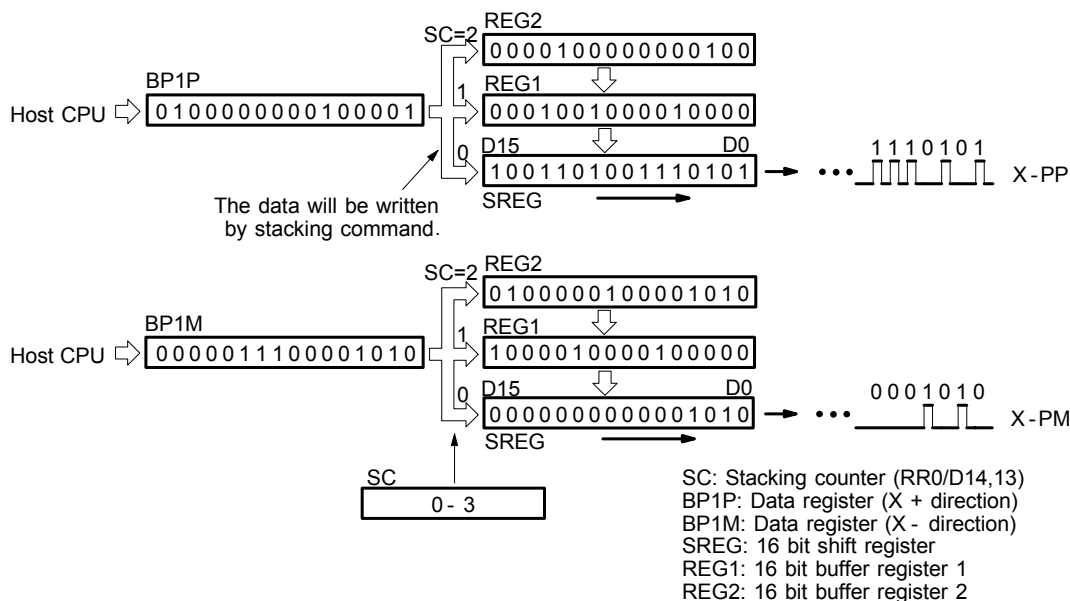


Fig. 2.31 Register Configuration and Movements of Bit Patterns of Bit Pattern Interpolation (for X axis)

Stacking counter (SC) is a 2-bit counter. Its value is between 0 and 3, which can be read from D14,13 of RR0 register. SC will decide which register for the data from the host CPU. The initial value of SC is 0. So, when host CPU writes bit pattern data into BP1P or BP1M, the data will be stored in SREG, and then, SC will count up to 1, and the next data from the host CPU will be written into REG1. By this way, the REG2 becomes the register when SC=2. The host CPU is not able to write any bit pattern data into MCX312 when SC=3.

When the bit pattern interpolation pulse is outputting, D0 in SREG will be shifted output first, and then in the order of D1, D...

When all of SREGs have been shifted output, the data in REG1 will be shifted to SREG, the data in REG2 will be shifted to REG1, and the SC will count down to 2. Then, the host CPU is able to write a new data into MCX312 again.

In order to make MCX312 output the bit pattern data continuously, the host CPU should write the data into MCX312 before SC counts down to 0. MCX312 will output a interrupt requirement signal to host CPU when SC counts down from 2 to 1.

■ The limitation for the speed of bit pattern interpolation driving

The maximum pulse output speed of MCX312 is 4MHz in bit pattern interpolation mode. However, the maximum speed will depend on the data update rate of host CPU if the bit pattern data are more than 48bits.

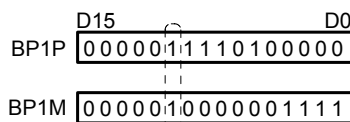
For example of the X and Y axes bit pattern interpolation, if the host CPU needs 100µsec to update new 16-bit data for X and Y axes. The maximum speed is $16/100\mu\text{SEC}=160\text{KPPS}$.

■ The ending of bit pattern interpolation

There are 2 ways can terminate the bit pattern interpolation.

- ① Write a ending code into buffer register of X axis.

The bit pattern interpolation mode will be finished, and stopped if the host CPU write "1" into both + and - directions buffer registers.



The Interpolation Stops Once The + And - Direction Are "1".

When the ending code is executed, the SC will become 0 automatically.

- ② The host CPU stops writing any command into MCX312.

When SC=0, and when no other data is updated, MCX312 will stop outputting pulse. Then, the bit pattern interpolation is finished.

■ Utilizing the stop command to pause the interpolation

The interpolation driving will be paused if a sudden stop or decelerating stop command is written into X axis which is executing the bit pattern interpolation. MCX312 will continue the bit pattern interpolation if the host CPU enables the bit pattern interpolation again. If the host CPU wants to finish the interpolation after writing stop command, all of the interpolation bit data in MCX312 must be cleared in using BP register (3Dh).

■ Utilizing hardware limit to interrupt the interpolation

The interpolation driving will be terminated when any hardware limit of any axis is active.

And, if host CPU wants to finish the interpolation, all of the interpolation data in MCX312 must be cleared.

■ Writing the bit pattern data into the register in MCX312

Either by 16-bit data bus or by 8-bit data bus, the address map of the command buffer for bit pattern interpolation data is show as follows:

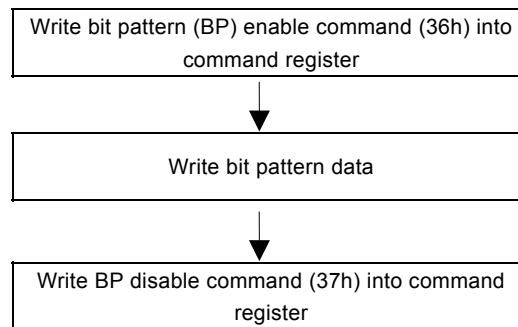
The addresses map of register for 16-bit data bus in bit pattern interpolation

| Address | | | Name of register | Content | The register with the same address |
|---------|----|----|------------------|-------------------|------------------------------------|
| A2 | A1 | A0 | | | |
| 0 | 0 | 0 | | | WR0 |
| 0 | 0 | 1 | | | nWR1 |
| 0 | 1 | 0 | BP1P | X +direction data | nWR2 |
| 0 | 1 | 1 | BP1M | X -direction data | nWR3 |
| 1 | 0 | 0 | BP2P | Y +direction data | WR4 |
| 1 | 0 | 1 | BP2M | Y -direction data | WR5 |

The addresses map of register for 8-bit data bus in bit pattern interpolation

| Address | | | | Name of register | Content |
|---------|----|----|----|------------------|---------------------------------|
| A3 | A2 | A1 | A0 | | |
| 0 | 1 | 0 | 0 | BP1PL | X +direction data the low byte |
| 0 | 1 | 0 | 1 | BP1PH | X +direction data the high byte |
| 0 | 1 | 1 | 0 | BP1ML | X -direction data the low byte |
| 0 | 1 | 1 | 1 | BP1MH | X -direction data the high byte |
| 1 | 0 | 0 | 0 | BP2PL | Y +direction data the low byte |
| 1 | 0 | 0 | 1 | BP2PH | Y +direction data the high byte |
| 1 | 0 | 1 | 0 | BP2ML | Y -direction data the low byte |
| 1 | 0 | 1 | 1 | BP2MH | Y -direction data the high byte |

For some addresses of bit pattern data registers are as same as nWR2 ~ nWR5, the host CPU can not write any data into the bit pattern data register since MCX312 has been reset. To write the bit pattern command, the host CPU should be with the following sequence.



[Note] If the host CPU doesn't disable the BP data register, the data in nWR2 ~ nWR5 registers cannot be assessed.

■ The example of bit pattern interpolation

The bit interpolation example is shown in Fig. 2.31. We set a constant speed: 1000PPS in a constant vector speed interpolation driving.

```

WR5 ← 0100h write      ; Constant linear speed

WR6 ← 0900h write      ; setting the speed parameter
WR7 ← 003Dh write      ; range: 4,000,000 (multiple rate: 2)
WR0 ← 0100h write

WR6 ← 4DC0h write      ; range of constant line speed
WR7 ← 0056h write      ; 4,000,000x1.414=5,656,000
WR0 ← 0200h write

WR6 ← 01F4h write      ; Initial speed: 500x2=1000PPS
WR0 ← 0104h write

WR6 ← 01F4h write      ; drive speed: 500x2=1000PPS
WR0 ← 0105h write

WR0 ← 0039h write      ; clear BP data

WR0 ← 0036h write      ; enable to write into BP register

BP1P ← 0000h write     ; command of bit 0~15th
                        ; X axis +direction
BP1M ← 2BFFh write     ; X axis -direction
BP2P ← FFD4h write     ; Y axis +direction
BP2M ← 0000h write     ; Y axis -direction
WR0 ← 0038h write     ; BP data stacking, SC=1

BP1P ← F6FEh write     ; command of bit 16~31th
                        ; X axis +direction
BP1M ← 0000h write     ; X axis -direction
BP2P ← 000Fh write     ; Y axis +direction
BP2M ← 3FC0h write     ; Y axis -direction
WR0 ← 0038h write     ; BP data stacking, SC=2

BP1P ← 1FDBh write     ; command of bit 32~47th
                        ; X axis +direction
BP1M ← 0000h write     ; X axis -direction
BP2P ← 00FFh write     ; Y axis +direction
BP2M ← FC00h write     ; Y axis -direction
WR0 ← 0038h write     ; BP data stacking, SC=3

WR0 ← 0034h write     ; enable 2 axis bit pattern
                        ; interpolation, because SC=3

J1  RR0/D14,13 read    ; until the SC is less than 2
    If D14=D13=1 Jump to J1

    BP1P ← 4000h write  ; command of bit 48~62th
                        ; X axis +direction
    BP1M ← 7FF5h write  ; X axis -direction
    BP2P ← 0000h write  ; Y axis +direction
    BP2M ← 0AFFh write  ; Y axis -direction
    WR0 ← 0038h write  ; BP data stacking, SC=3

    WR0 ← 0037h write  ; disable to write into BP register

J2  RR0/D8 read        ; until ending interpolation drive
    If D8=1 Jump to J2

```

(Note 1: If there is more BP data coming then repeat this process)

■ Executing bit pattern interpolation by interrupt

During the bit pattern interpolation, MCX312 will generate an interrupt request signal to the host CPU while SC changes the value from 2 to 1. To enable the interrupt, the host CPU must set D15 of register WR5 to 1. Then, INTN of MCX312 will go low once SC changes the value from 2 to 1. The host CPU will check the SC value, and write bit pattern command into the register. The interrupt signal will be released if the host CPU writes the SC stacking command (38h) into MCX312.

The interrupt signal will be released when the clear command (3Dh) is written into the command register. If the interrupt status is keeping on the Low level, it will return to high-Z level after MCX312 finishes the bit pattern interpolation.

2.4.4 Constant Vector Speed

MCX312 is with the constant vector speed control function which can control the resultant speed of two axes to keep the speed in constant.

Fig. 2.33 shows the profile of 2 axes interpolation driving. The vector speed reflects 1.414 times of the individual axis drive speed. So, we have to set the speed of 1.414 times to keep the vector speed for 2-axis driving.

■ Constant Vector Speed Set-up

The user should first set the values of D9 and D8 of register WR5 to 0 and 1. Then, set the range R of Y axis to be 1.414 times of the value of X axis. Therefore, MCX312 will use the range parameter of X axis if only 1 axis outputs pulses. However, when 2 axes output pulses simultaneously, MCX312 will use the range parameter of Y axis to implement the pulse period to 1.414 times.

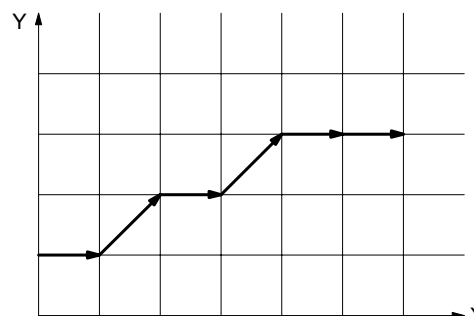


Fig. 2.32 Example of 2 -Axis Interpolation

■ The Example of Constant Vector Speed Interpolation for 2 Axes

As shown below, X axis, Y axis, and the interpolation is at a constant vector speed: 1000PPS. The result of driving pulse output is shown in Fig. 2.34.

| | | | |
|-------------------|----------------------------------|-------------------|--|
| WR5 ← 0100h write | ; constant vector speed | WR6 ← 03E8h write | ; finish point of X |
| WR6 ← 0900h write | ; setting parameter of X axis | WR7 ← 0000h write | |
| WR7 ← 003Dh write | ; range: 4,000,000 (multiple=2) | WR0 ← 0106h write | |
| WR0 ← 0100h write | | WR6 ← 0190h write | ; finish point of Y |
| WR6 ← 4DC0h write | ; range of constant vector speed | WR7 ← 0000h write | |
| WR7 ← 0056h write | ; 4,000,000x1.414=5,656,000 | WR0 ← 0206h write | |
| WR0 ← 0200h write | | WR0 ← 0030h write | ; 2-axis linear interpolation starting |
| WR6 ← 01F4h write | ; initial speed: 500x2=1000PPS | | |
| WR0 ← 0104h write | | | |
| WR6 ← 01F4h write | ; drive speed: 500x2=1000PPS | | |
| WR0 ← 0105h write | | | |

(continue) ↗

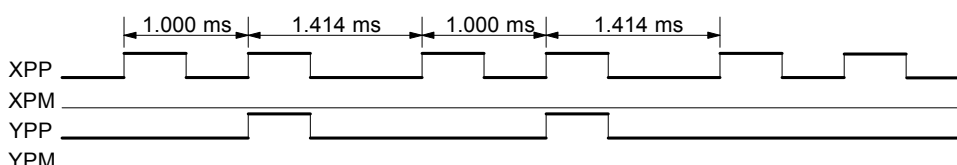


Fig. 2.33 Example of 2 -Axis Interpolation at A Constant Vector Speed (Speed =1000PPS)

[Caution] In the process of constant vector speed, the pulse width of high level of output waveform will not be changed, yet kept in the same width. The pulse cycle will be changed to 1.414 times.

2.4.5 Continuous Interpolation

The continuous interpolation is executing a series of interpolation processes such as linear interpolation → circular interpolation → linear interpolation → ...

During the continuous interpolation, the driving will not stop; contrarily, the pulses are output continuously. When executing the continuous interpolation, the host CPU has to write the next interpolation segment into MCX312 before the previous interpolation segment is finished.

■ Continuous Interpolation in Using D9 of RR0

If D9 (CNEXT) of register RR0 is 1, MCX312 is ready to accept the next interpolation segment. If D9 is 0, the host CPU is not able to write the next interpolation segment into MCX312. The D9 will become 1 only when the present command is executed. MCX312 will not accept the next command, and the D9 is 0 if the present command has not been executed.

So, the standard procedure of continuous interpolation is first to write, and enable the interpolation data and command, then check if D9 of RR0 is 1 or 0. And then, repeat writing commands and checking D9. The flow chart is shown at the right side.

■ Interrupt

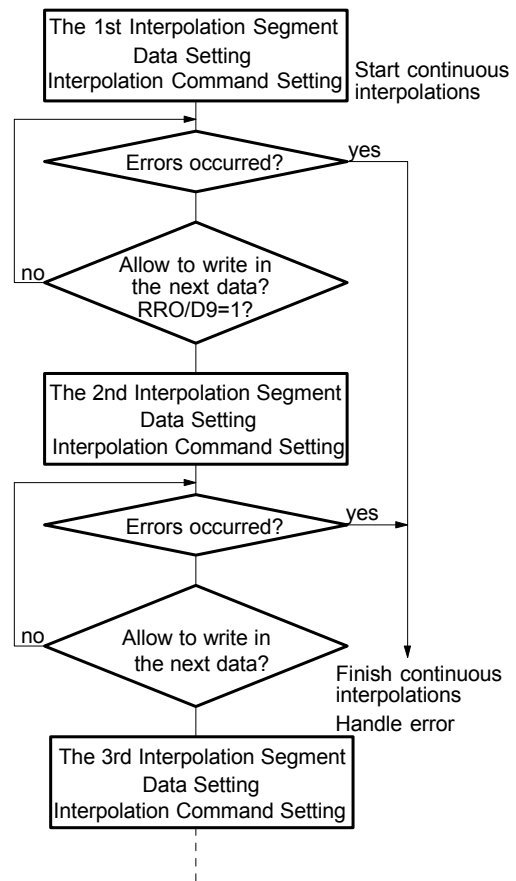
D14 of register WR5 is used for enable or disable the interrupt during the continuous interpolation. After setting D14 of register WR5 to 1, the interrupt occurs. Pin INTN of MCX312 will be on the Low level to interrupt the host CPU when D9 of register RR0 become 1. The INTN will be on the Hi level if the host CPU writes the next interpolation segment to MCX312. If the interrupt clear command (3Dh) is written to command register, the INTN signal will return to high-Z level from the Low level.

During the ending of the interpolation, it is forced to be “interrupt disable”, and the INTN signal will return to the high-Z level.

■ Errors Occurring in the Process of Continuous Interpolation

If an error such as over-traveling occurs in the process of continuous interpolation, the drive will stop at the present interpolation segment. The following interpolation segment is still in the command register, but will not be executed. The host CPU has to reload the next command again and enable it.

As shown in the flow chart above, the host CPU has to check the error message before loading the following command. If not, this command will not be executed and will be jumped. So, the user should assure, and check if any error status will occur before the following interpolation segment is loaded.



■ Attentions for Continuous Interpolation

- a. Before setting the interpolation segment, the user should first set other data such as center point, finish point... for each segment.
- b. The maximum speed for the continuous interpolation is 2MHz.
- c. The following interpolation segment must be loaded before the previous interpolation segment is finished.
- d. The segment driving time should be longer than the time for error checking and the command setting of next segment during the interpolation.
- e. In continuous interpolation, if one of 2/3 axes is 0, interpolation is performed correctly, otherwise, 0 cannot be set to the finish point of all axes in 2/3 axes linear interpolation, or to the center point of both axes in circular interpolation, any axis cannot set the data that drive pulse is not output. If suchlike data it set, interpolation cannot be performed correctly.

■ The Example of Continuous Interpolation

Fig. 2.35 shows an example of executing continuous interpolation beginning at point (0,0) from segment 1, 2, 3...to the segment 8. In segment 1, 3, 5, and 7, the linear interpolation will be executed; in segment 2, 4, 6, and 8, the circular interpolation will be executed, and the track is a quadrant circle with radius 1500. The interpolation driving is at a constant vector speed: 1000PPS.

```

WR5 ← 0100h write ; constant vector speed

WR6 ← 0900h write ; setting the parameter
WR7 ← 003Dh write ; range:4,000,000 (multiple: 2)
WR0 ← 0100h write

WR6 ← 4DC0h write ; 2-axis constant vector speed
WR7 ← 0056h write ; 4,000,000x1.414=5,656,000
WR0 ← 0200h write

WR6 ← 01F4h write ; Initial speed: 500x2=1000PPS
WR0 ← 0104h write

WR6 ← 01F4h write ; drive speed: 500x2=1000PPS
WR0 ← 0105h write

WR6 ← 1194h write ; finish point X: 4500
WR7 ← 0000h write
WR0 ← 0106h write

WR6 ← 0000h write ; finish point Y: 0
WR7 ← 0000h write
WR0 ← 0206h write

WR0 ← 0030h write ; 2-axis linear interpolation
    
```

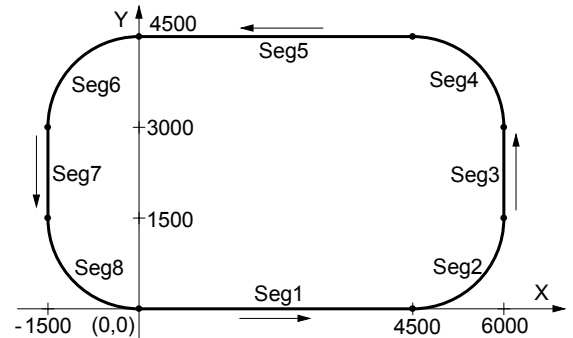


Fig. 2.34 The Example of Continuous Interpolation Axis

Seg1

Procedure A

```

J1 RR0(D4, D5) read ; if error occurs
If D4 or D5=1 ; jump to handle error
Jump to Error

RR0(D9) read ; waiting for next segment' s
enable signal

If D9=0 Jump to J1 ;
    
```

```

WR6 ← 0000h write ; center X: 0
WR7 ← 0000h write
WR0 ← 0108h write

WR6 ← 05DCh write ; center Y: 1500
WR7 ← 0000h write
WR0 ← 0208h write

WR6 ← 05DCh write ; finish point X:1500
WR7 ← 0000h write
WR0 ← 0106h write

WR6 ← 05DCh write ; finish point Y:1500
WR7 ← 0000h write
WR0 ← 0206h write

WR0 ← 0033h write ; CCW circular interpolation
    
```

Procedure A

```

WR6 ← 0000h write ; finish point X: 0
WR7 ← 0000h write
WR0 ← 0106h write

WR6 ← 05DCh write ; finish point Y: 1500
WR7 ← 0000h write
WR0 ← 0206h write

WR0 ← 0030h write ; 2-axis linear interpolation
    
```

Procedure A

⋮
(same procedure for segments 4 ~ 8.)

2.4.6 The Acceleration / Deceleration Control in Interpolation

Different from other IC chips only allowing constant speed for executing the interpolations, MCX312 supports the user to use trapezoidal and S-curve driving (for linear interpolation only).

In the process of interpolation, for executing acceleration / deceleration in continuous interpolation process, the user can enable the deceleration by command (3Bh), or disable deceleration by command (3Ch). The purpose for the deceleration command is to enable the automatic deceleration or manual deceleration function; the purpose of the disable deceleration command is to disable both of them. It will be disable while power-on reset. During the driving, the deceleration enable command cannot be executed.

■ The Acceleration / Deceleration for 2-axis Interpolation

It is possible to perform trapezoidal and S-curve acceleration/deceleration driving during the execution of 2-axis linear interpolation. Either automatic or manual deceleration can be used for decelerating.

When the manual deceleration is executed, the user can set the maximum absolute value of the axes to be the setting value of X axis decelerating point. For instance, while executing 2-axis linear interpolation to the finish point : (X: -20000, Y: 60000), and the assumed pulse numbers needed for deceleration are 5000. In such situation, the absolute value of Y axis is the largest, so we can set up 60000-5000=55000 to be the manual deceleration point of X axis.

Please refer to the example of 2-axis linear interpolation in 2.4.1.

■ The Acceleration / Deceleration for Circular Interpolation and Bit Pattern Interpolation

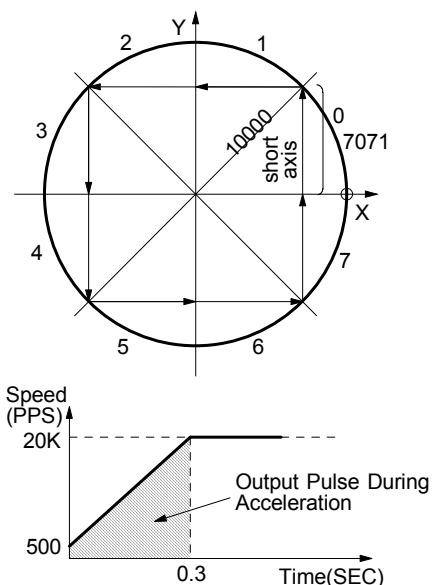
In circular interpolation and bit pattern interpolation, only manual deceleration in trapezoidal driving is available; the automatic deceleration in S-curve driving is not available.

The Figure on the right side shows the circular interpolation of a real circle with radius 1000 in a trapezoidal driving.

The user should calculate the decelerating point before driving because the automatic deceleration will not be active.

In the figure, the circle tracks through all the 8 quadrants: 0~7. In quadrant 0, Y axis is the short axis and it's displacement is about $10000 / \sqrt{2} = 7071$. The total output pulses numbers of the short axis are $7010 \times 8 = 56568$.

Furthermore, if the initial speed is 500PPS, and will be accelerated to 20KPPS after 0.3 SEC, the acceleration will be $(20000 - 500) / 0.3 = 65000 \text{ PPS/SEC}$. And the output pulses during acceleration will be $(500 + 20000) \times 0.3 / 2 = 3075$. Thus, if we set the deceleration as same as the acceleration, the manual decelerating point will be $56568 - 3075 = 53493$. [Note] this formula cannot be used in the constant vector speed driving.



```

WR3 ← 0001h write      ; manual deceleration enabling

WR6 ← 8480h write      ; range: 2,000,000; (multiple: 4)
WR7 ← 001Eh write      ;
WR0 ← 0100h write

WR6 ← 0082h write      ; acceleration :
WR0 ← 0102h write      ; 130x125x4=65000 PPS/SEC

WR6 ← 007Dh write      ; Initial speed:125x4=500PPS
WR0 ← 0104h write

WR6 ← 1388h write      ; drive speed
WR0 ← 0105h write      ; 5000x4=20000PPS

WR6 ← D8F0h write      ; center point X : -10000
WR7 ← FFFFh write      ;
WR0 ← 0108h write

WR6 ← 0000h write      ; center point Y : 0
WR7 ← 0000h write      ;
WR0 ← 0208h write

WR6 ← 0000h write      ; finish point X : 0
WR7 ← 0000h write      ;
WR0 ← 0106h write

WR6 ← 0000h write      ; finish point Y : 0
WR7 ← 0000h write      ;
WR0 ← 0206h write

WR6 ← D0F5h write      ; manual deceleration point setting : 53493
WR7 ← 0000h write      ;
WR0 ← 0107h write

WR0 ← 003Bh write      ; deceleration enabling
WR0 ← 0033h write      ; CCW circular interpolation starting

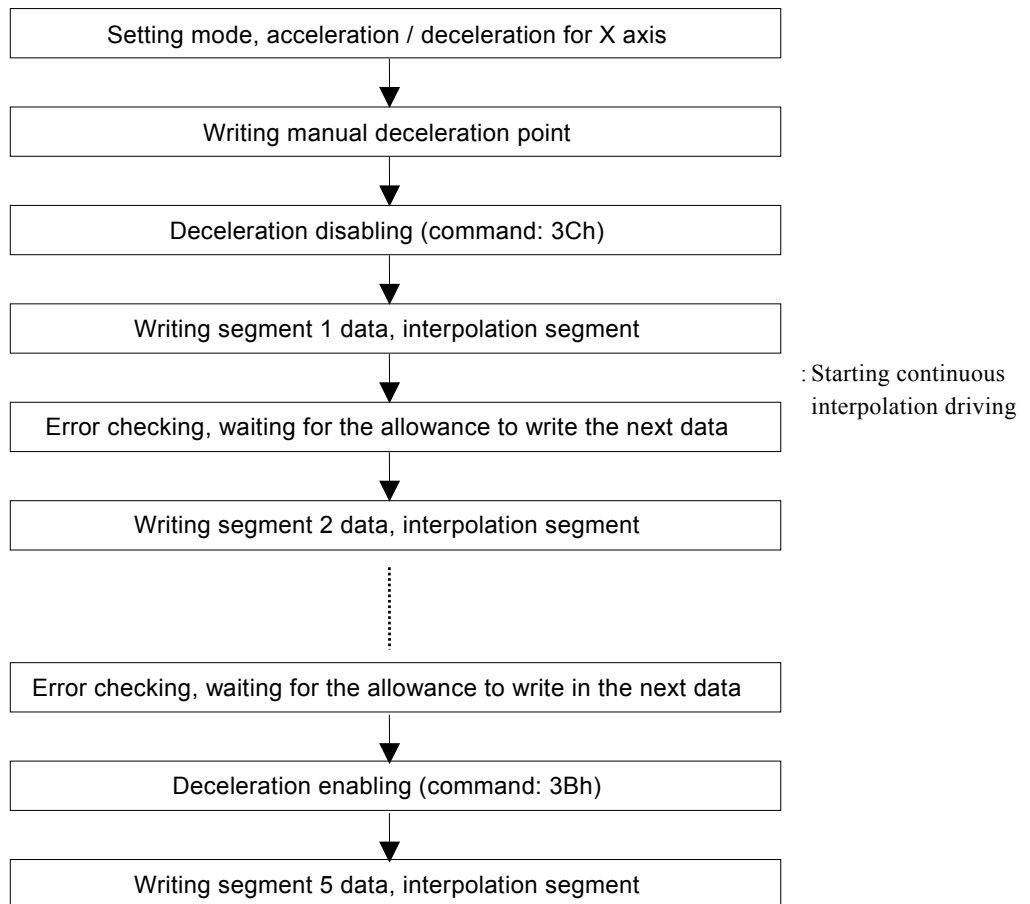
```

■ The Acceleration / Deceleration for Continuous Interpolation

In continuous interpolation, same as in circular and bit pattern interpolations, only manual deceleration in the trapezoidal driving is available; The automatic deceleration in S-curve driving is not available.

Before performing the continuous interpolation, it is necessary to preset the manual decelerating point; however, this setting point is related to X axis executing the deceleration in the last segment. The user should disable the deceleration, then start the interpolation driving. Before writing the interpolation command to the final segment which will execute the deceleration, the user should enable the deceleration at first. The deceleration will start if the output pulses are larger than X axis based pulses in the final segment.

For instance, there are 5 interpolation segments in the process of continuous interpolation. In case, the manual deceleration has to be executed in the last segment, segment 5, the procedure is shown as follows:



The manual deceleration point is related to X axis' driving pulses which comes from segment 5. For instance, assumed that it needs 2000 pulses for decelerating stop, and the total amount of pulse output form segment 5 is 5000. So, the manual deceleration point will be $5000 - 2000 = 3000$.

The cycle of deceleration should be started and finished within the same segment.

2.4.7 Single-step interpolation (from Command or External Signal)

Single-step is defined as: pulse by pulse outputting. Either command or external signal can execute the single-step interpolation. When one pulse is outputting, X axis interpolation will be set in the constant speed driving.

The Hi level width of each axis' s output pulse is 1/2 of the pulse cycle which is decided by the interpolating X axis' s drive speed. The Low level width is kept until next command or external signal comes. Fig. 2.36 is the example showing the execution of single-step interpolation from an external signal. X axis' s initial speed is 500PPS, the drive speed is at 500PPS constant speed driving. The Hi level width of output pulse is 1msec.

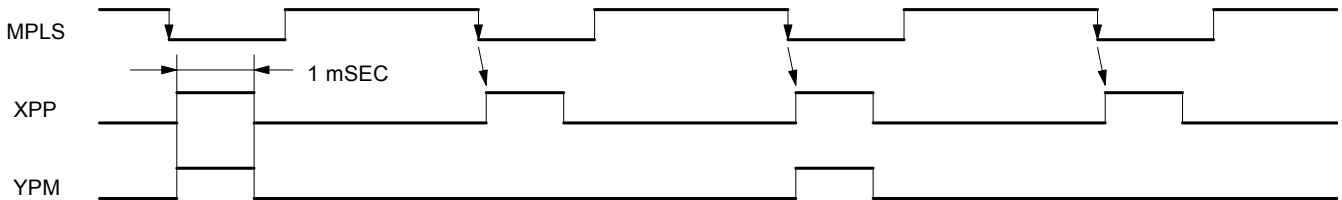


Fig. 2.35 Example of Single Step Interpolation (500PPS) by External Signal (MPLS)

■ Command Controlled Single-step Interpolation

The command: 3Ah is for single-step interpolation. The user can set D13 of register WR5 to 1 to enable the command controlled single-step interpolation. The operating procedure is shown as follow.

a. Set D13 of register WR5 to 1.

It will enable the command controlled single-step interpolation.

b. Set the initial and drive speeds of X axis in the interpolation process with the same value, and the driving becomes constant speed.

If the host CPU writes single step command into MCX312 at most 1mSEC, the user should set the drive speed more than 1000PPS.

c. Set interpolation data. (finish point, center point...)

d. Write interpolation command.

Although the interpolation segment is enabled, there is no pulse output because the single-step is command controlled.

e. Write the single-step interpolation command (3Ah).

The driving pulses according to the interpolation calculation will be output for each axis. The user may use command 3Ah for single step until the interpolation driving is finished.

If the user wants to stop sending single-steps during the interpolation, he can use the sudden stop command (27h), then wait for more than 1 pulse cycle, and then write the command (3Ah) again to stop the driving.

After this, all the following (3Ah) commands will not be active.

■ External Signal Controlled Single-step Interpolation

The MPLS pin (70) is used for the single-step interpolation from the external signal. This signal shares the pin with general-purpose input signal XIN5. The user can set D12 of register WR5 to 1 to enable the external signal controlled single-step interpolation. Normally, the MPLS input signal is on the Hi level. When it changes to Low, the interpolation step will be output.

The operating procedure is shown as follows.

a. Set D12 of register WR5 to 1.

It will enable the external signal controlled single-step interpolation.

b. Set the initial and drive speeds of X axis in the interpolation process to be the same value, and the driving becomes constant speed which should be higher than the Low pulse cycle of MPLS.

This is necessary for this controlled mode. And it will set the MCX312 into a constant speed mode.

c. Set interpolation data. (start point, center point...)

d. Write interpolation command.

Although the interpolation segment is enabled, there is no pulse output because the single-step is command controlled.

e. MPLS input on Low level

The interpolation pulse will be output from each axis after 2~5 CLK the pulse falling down (when the filer is invalid).

The Low level pulse width of MPLS has to be longer than 4CLK. Furthermore, the pulse cycle of MPLS has to be longer than the setting speed cycle of X axis.

The user may repeat the Low level of MPLS before the interpolation is finished.

If the user wants to stop sending single-steps during the interpolation, he can use the sudden stop command (27h), then wait for more than 1 pulse cycle, and then input pulse on MPLS Low level again to stop the driving (the user may try software reset also).

After this, all the following input pulses on MPLS Low level will not be active.

[Note] When generating Low pulses of MPLS at a mechanical contact point, prevent the occurrence of chattering by enabling the input signal filter of the MPLS signal.

2.4.8 Multichip axes interpolation

This function performs a linear interpolation of multi-axes using several chips this IC. Fig 2.37 shows the connection example where 6 –axes linear interpolation is performed by 3 chips. Main chip has a role to output synchronous pulses of interpolation drive to sub chip.

As shown in Figure, 8 signals (MPLS, MCLK, MERR, MINP, MDT3~0) are connected among chips respectively, and pulled up with 680Ω impedance. These signals share the general input signal (XIN5~2, YIN5~2) so that they cannot be used as general input signal. Each signal works as follows.

- MDT3 ~ 0: Transfer finish point data of each chip. (Main ↔ Sub)
- MCLK: Clock of data transfer for MD3 ~ 0. (Main ↔ Sub)
- MPLS: Synchronous pulse of interpolation drive. (Main → Sub)
- MERR: Inform error occurring. (Sub → Main)
- MINP: Inform in-position waiting. (Sub → Main)

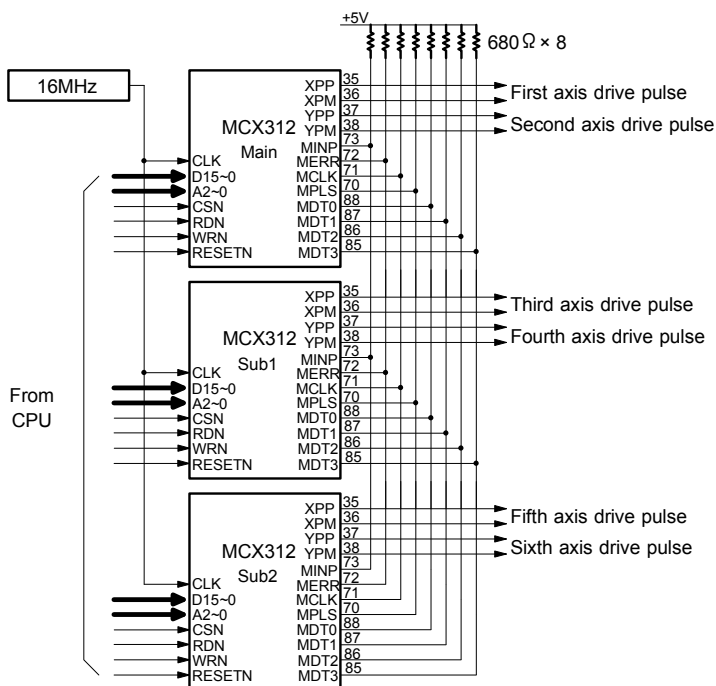


Fig 2.36 Example of signal connection for multichip axes interpolation

Procedures for linear interpolation by multi chips are described below.

① Mode setting of multichip interpolation

Assign main / sub chip by D11, 10 bit of WR5 register.

| WR5/D11 | WR5/D10 | Setting |
|---------|---------|--|
| 0 | 0 | Do not perform multichip interpolation |
| 0 | 1 | Main chip |
| 1 | 0 | Sub chip X and Y (both X and Y are used) |
| 1 | 1 | Sub chip X (only X is used) |

If sub chip X is assigned, Y axis of that chip is used independently at multichip interpolation. However, YIN5~2 signal cannot be used as general input.

Set mode setting of limit signal and servo motor-enabled signal for each axis the same as normal independent drive.

② Speed parameter setting

Set interpolation drive speed, which becomes the drive speed of the longest length axis of finish point in all axes.

And set speed parameter to main chip and X axis of each sub chip as follows. There is no need to set speed parameter for Y axis of each chip. When acceleration / deceleration is not performed, no need to set acceleration and deceleration but initial speed and drive speed should be set the same value.

| Speed parameter | | Main chip X axis | Sub chip X axis |
|-----------------|----|--|---|
| Range | R | Required | Set the same value as main range |
| Jerk | K | Required when S curve acceleration/deceleration *1 | Not required |
| Acceleration | A | Required when acceleration/deceleration | Not required |
| Deceleration | D | Required when acceleration/deceleration | Not required |
| Initial speed | SV | Required (the same value as drive speed when constant speed drive) | Set the same value as main drive speed |
| Drive speed | V | Required | Set the same value as main drive speed |

*1: WR3/D2 (SACC) bit should be set to 1 when S curve acceleration.

Drive speed of multichip interpolation must be 2Mpps or less.

③ Finish point data setting of each axis

Writes finish point data to each axis by relative value from the current position.

When multi –axes linear interpolation is performed, generally, the maximum value of finish point data in all axes is required in calculating linear interpolation for each axis. In order to enable high –speed continuous linear interpolation, this IC generates the maximum value automatically when finish point of each axis is set. There is no need to calculate the maximum value by CPU and set the maximum value to each axis.

When finish point data is written in some axis, it is transferred from written chip to another chip through multichip interpolation signal (MCLK, MDT3~0). It takes about 2 μ sec (CLK=16MHz). Therefore an interval of writing of finish point for each axis cannot be shortened than that time. In normal CPU, it takes more time so that it is unnecessary to input delay in the software. When each chip receives finish point data from multichip interpolation signal (MCLK, MDT3~0), it compares the value with the finish point maximum value register in own chip by absolute value, and if received value is larger than its own value, then updates the value of the finish point maximum register by absolute value.

The finish point maximum value register is cleared to 0 when resetting or immediately after starting interpolation drive command. Also it can be cleared by the finish point maximum value clear command (3Eh). The finish point maximum value can be read by 14h command (X axis assignment), user can confirm whether the maximum value is correctly generated after writing finish point data of all axes. Please note that the maximum value is cleared to 0 once interpolation drive starts.

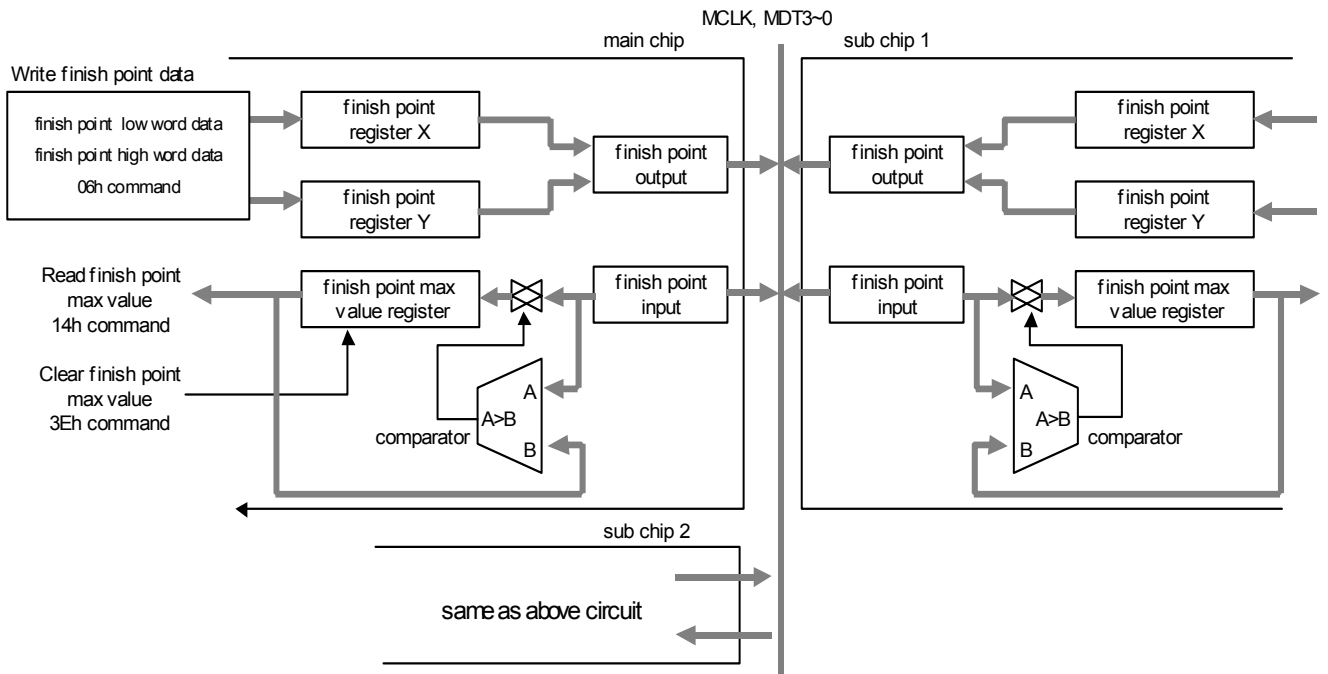
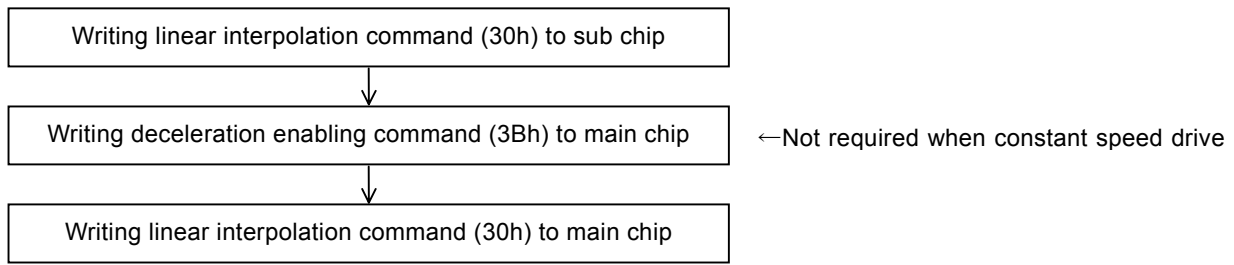


Fig 2.37 Multichip interpolation finish point data passing

The finish point maximum value must be set even though next finish point of interpolation is the same value as the previous one because it is cleared every interpolation driving.

④ Writing linear interpolation drive command

Linear interpolation command (30h) is written to each sub chip, and then linear interpolation command (30h) is written to main chip. If the command (30h) is written to main chip before sub chip, the multichip interpolation does not work properly. When acceleration / deceleration is performed, deceleration enabling command (3Bh) must be written to main chip before interpolation drive command is written.



Once 30h command is written to main chip, interpolation drive starts immediately and main chip starts to output synchronous pulse of interpolation drive from MPLS signal to each sub chip.

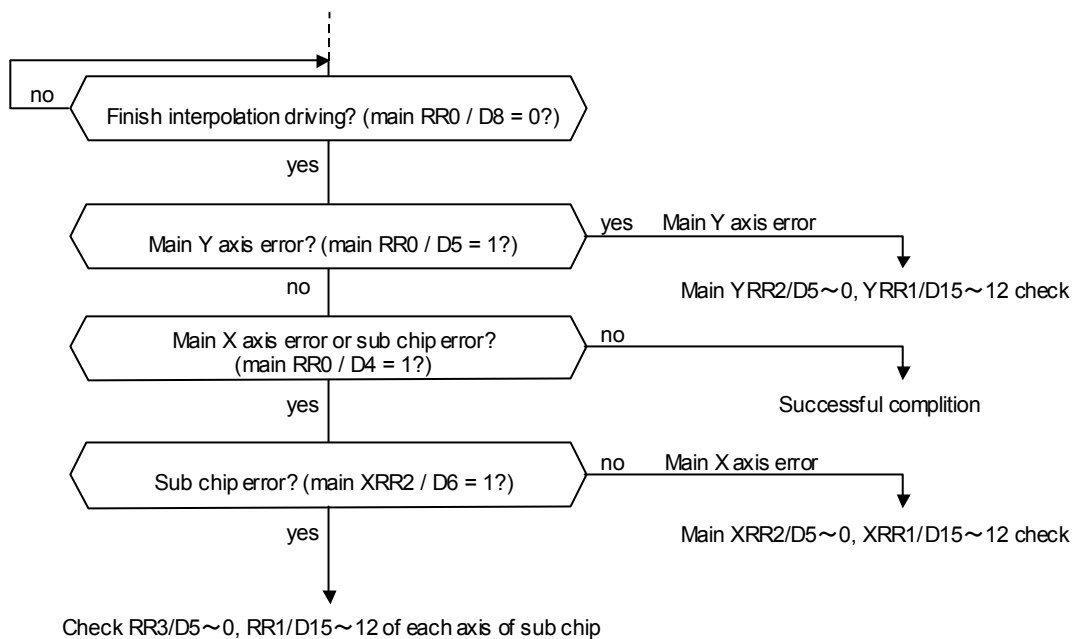
⑤ Finish of interpolation

RR0/D8 (I-DRV) bit of main chip becomes 1 during interpolation driving, and it returns to 0 when interpolation drive finishes. If in-position of each axis is enabled, RR0/D8 (I-DRV) bit of main chip returns to 0 after INPOS signal of every enabled axis becomes active level.

⑥ Errors occurring and check

When an error occurs in any axis of main chip during interpolation driving, one of bit D5~0/RR2 register of the axis becomes 1 and bit D5, 4 (Y-ERR, X-ERR) /RR0 register becomes 1. And when an error occurs in any axis of sub chip, sub chip makes MERR signal of multichip interpolation signal Low Active and informs main chip about an error occurring. In main chip, when errors are received, bit D6 (MULT) /RR2 register of X axis becomes 1 and bit D4 (X-ERR) /RR0 register becomes 1. Main chip stops outputting synchronous pulses of interpolation drive to sub chip if an error occurs, as a result, all axes stop immediately.

Example of the error check after interpolation driving is shown as follows.



When an error occurs in sub chip, synchronous pulses of interpolation drive from main stops on the sub chip side and keep it driving. Be sure to revert to initial status for each sub chip by soft reset after an error check.

In multichip interpolation, an error occurs even limit that is the opposite side of the direction of sub chip axis becomes active,

then interpolation drive stops. When a limit error occurs, escape from limit over by individual drive.

■ Continuous interpolation of multichip linear interpolation

In the case of multichip, linear interpolation can be performed continuously as well as single chip of 2-axis interpolation. Once interpolation drive starts, bit RR0/D9 (CNEXT) of main chip becomes 1. If D9 (CNEXT) bit = 1 is confirmed, finish point data of each axis in next segment will be written and linear interpolation command (30h) will be written to each axis.

■ Continuous interpolation setting example of multichip linear interpolation

Please refer to chapter 11 for program example.

[Note for multichip interpolation]

- Do not cross the wiring path of multichip interpolation signal (MPLS, MCLK, MERR, MINP, MDT3~0) with other signals, and connect them as short as possible, and cannot share the general input signal by jumper switching in customer's circuit system.
- In multichip interpolation, constant vector speed can be performed only with 2-axis of main chip, in this case, user should set the range of Y axis to be 1.414 of X axis value.
- In-position should be set disabled in continuous interpolation.

2.5 Interrupt

The interrupt is generated from X, Y axis, bit pattern interpolation or continuous interpolation.

There is only one interrupt signal, INTN (33), to the host CPU. So, the signal will be OR calculated, then output, as shown in Fig. 2.39.

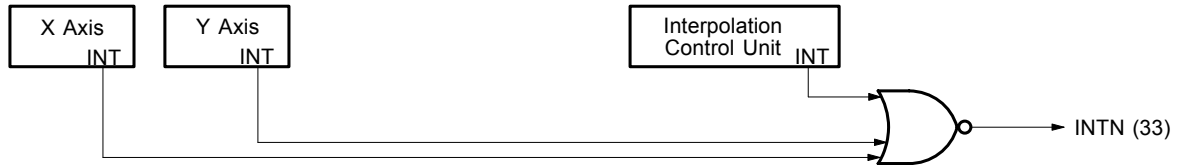


Fig. 2.38 Interrupt Signal Path in IC

Every interrupt can be enabled or disabled individually. When resetting, all interrupt signals are disabled.

■ Interrupt of X and Y Axes

The following table shows the interrupt factors generated by X and Y axes.

| Enable / Disable nWR1 Register | Status nRR3 Register | The Factors of Interrupt Happening |
|-----------------------------------|-------------------------|---|
| D9 (P≥C-) | D1 (P≥C-) | once the value of logical / real position counter is larger than or equal to the value of COMP- register (CM)... |
| D10 (P<C-) | D2 (P<C-) | once the value of logical/real position counter is smaller than the value of COMP- register (CM)... |
| D11 (P<C+) | D3 (P<C+) | once the value of logical / real position counter is larger than the value of COMP+ register (CM)... |
| D12 (P≥C+) | D4 (P≥C+) | once the value of logical / real position counter is smaller than or equal to the value of COMP+ register (CM)... |
| D13 (C-END) | D5 (C-END) | in the acceleration / deceleration driving, when the driving changes from the constant speed region into the decelerating region... |
| D14 (C-STA) | D6 (C-STA) | in the acceleration / deceleration driving, when the driving changes from the accelerating region into the constant speed region... |
| D15 (D-END) | D7 (D-END) | when the driving is finished... |

Each factor of interrupt can be masked by setting levels in nWR1 register bits: 1- enable and 0 - disable. When interrupt is generated during the driving, and if the interrupt is generated, each bit in nRR3 will be set to 1; INTN will be on the Low level. After the nRR3 status has been read from the host CPU, nRR3 will be cleared from 1 to 0, and INTN will return to the High-Z level.

■ Interrupt from Interpolations

| Enable / Disable WR5 Register | Status Check RR0 Register | The Factors of Interrupt Happening *Interrupt Clearing |
|----------------------------------|------------------------------|--|
| D14 (CIINT) | D9 (CNEXT) | in continuous interpolation, when MCX312 is available for the interpolation data of next node... *after next interpolation command is written, the interrupt will be cleared. |
| D15 (BPINT) | D14, 13 (BPS1,0) | In bit pattern interpolation, when the value of stack connector (SC) is changed from 2 to 1, and the stack is available for next BP command writing... *after a BP command for the stack is written, the interrupt will be cleared. |

When an interrupt is generated during interpolations, this interrupt can be cleared by writing the interrupt clear command (3Dh) INTN will return to the High-Z level automatically once the interpolation is finished.

See Bit pattern /Continuous Interpolation section for executing interpolation by interrupt.

2.6 Other Functions

2.6.1 Driving By External Pulses

Fixed driving and continuous driving can be controlled by either commands or external signals, which can reduce the load of host CPU. By inputting an encoder 2-phase signal of a manual pulsar, jog feed of each axis is enabled.

Each axis has two input signals, nEXPP and nEXPM. In fixed drive mode and a continuous drive mode, the nEXPP signal triggers driving in the + direction and the nEXPM signal in the – direction. nEXPP controls + direction pulse output, and nEXPM controls – direction command. D3 and D4 bits of register WR3 are for the setting in driving. The user should preset the parameters and commands. The default level of nEXPP and nEXPM is normally set on Hi. In manual pulsar mode, the A-phase signal is connected to nEXPP input and the B-phase signal to nEXPM input.

■ Fixed Driving Mode

Set bits D4 and D3 of register WR3 to 1 and 0 respectively, and set all the parameters of fixed driving. Once nEXPP is falling down to the Low level (↓), the + direction fixed driving will start; once nEXPM is raising to the Hi level (↑), the – direction fixed driving will start. The width of Low level signal must be larger than 4 CLK-cycle. Before this driving is finished, a new Hi-to-Low level falling down of the signal is invalid.

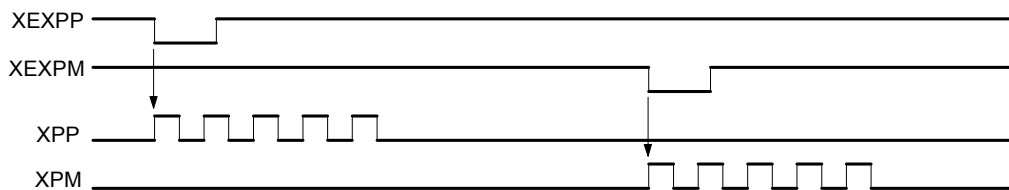


Fig. 2.39 Example of The Constant Driving by External Signal

■ Continuous Driving Mode

Set bits D4 and D3 of WR3 register to be 0 and 1 respectively, and set all the parameters of continuous driving. Once nEXPP is falling down to the Low level (↓), the + direction continuous driving will start; once nEXPM is raising to the Low level (↓), the – direction continuous driving will start. When nEXPP and nEXPM returns to the Hi level from the Low level, the decelerating stop will be performed in trapezoidal driving, and the sudden stop in constant speed driving.

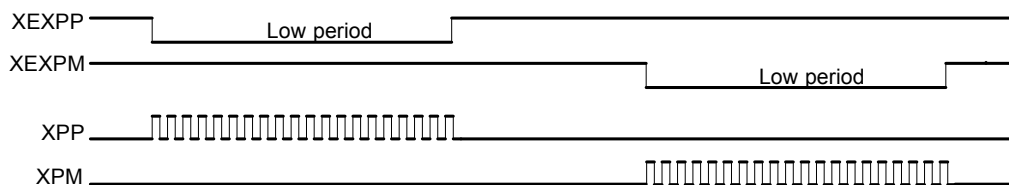


Fig. 2.40 Example of Continuous Driving by External Signal

■ Manual pulsar mode

Set the bits D4 and D3 of the WR3 register to 1 and set the necessary speed parameter for driving and the output pulse number. Connect the A-phase signal of the encoder to nEXPP input and the B-phase signal to nEXPM input. The – fixed driving is activated when the nEXPM signal is at a Low level and the nEXPP signal is at the rising edge. When the output pulse number is set to 1, one drive pulse is output at each of the rising edge and falling edge of the nEXPP signal. If the output pulse number is set to P, the P number of drive pulses is output.

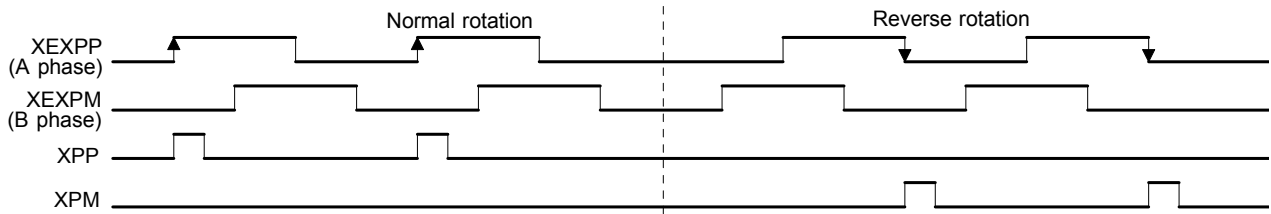


Fig. 2.41 Example Output Pulse 1 Driving by A Manual Pulsar

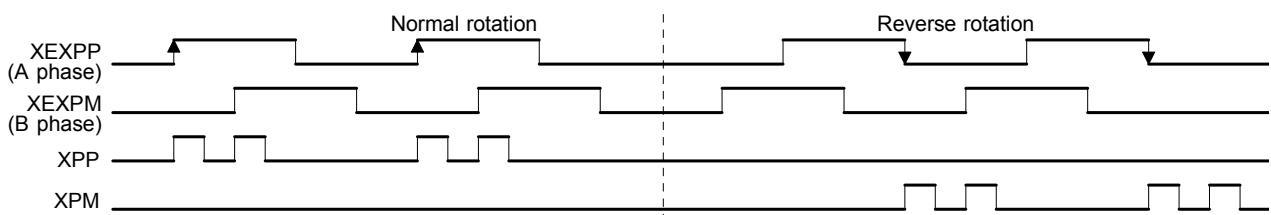


Fig. 2.42 Example of Output Pulse 2 Driving by A Manual Pulsar

Set the speed parameter in the following conditions to complete output of all the P number of drive pulses with a period from the rising edge/falling edge of the nEXPP signal to the next rising edge/falling edge.

$$V \geq F \times P \times 2$$

- V : Drive speed (pps)
- P : Output pulses
- F : Frequency (Hz) at the maximum speed of the manual pulsar encoder

For instance, under the condition where the maximum frequency of the manual pulsar is F=500Hz and the output pulse is P=1, the drive speed must be V=1000PPS or greater. Since acceleration/deceleration driving is not applied, set the initial speed SV to the same value as the drive speed. However, when a stepping motor is used for driving, the drive speed must not exceed the automatic activation frequency of the motor.

2.6.2 Pulse Output Type Selection

There are two types of pulse output--independent 2-pulse type: when the driving is in + direction, the pulse output is from nPP/PLS; when the driving is in – direction, the pulse output is from nPM/DIR; 1-pulse 1- direction type: nPP/PLS is for pulse outputting, and nPM/DIR is for direction signal outputting.

| Pulse Output Type | Drive Direction | (pulse / direction is set on the positive logical level) | |
|---------------------|-----------------|--|----------------|
| | | Pulse Output Waveform | |
| | | nPP/PLS Signal | nPM/DIR Signal |
| Independent 2-pulse | +Direction | | Low level |
| | -Direction | Low level | |
| 1-pulse 1-direction | +Direction | | Low level |
| | -Direction | | Hi level |

Bit D6 (PLSMD) of register WR2 is used for the selection of pulse output type.

Additionally, bits D7 (PLS-L) and D8 (DIR-L) of register WR2 can be used for pulse outputting, direction and logical level setting.

[Note] Please refer to Chapter 13.2, 13.3 for the pulse signal (nPLS) and direction signal (nDIR) in 1-pulse 1-direction pulse outputting.

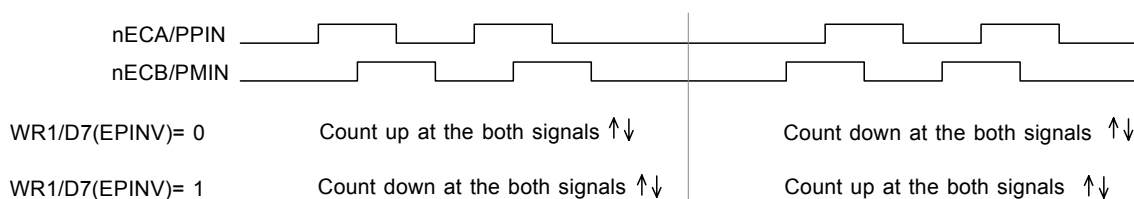
2.6.3 Pulse Input Type Selection

For real position counter, A/B quadrature pulse type and Up / Down pulse type can be selected for pulse input.

■ A/B quadrature pulse input mode

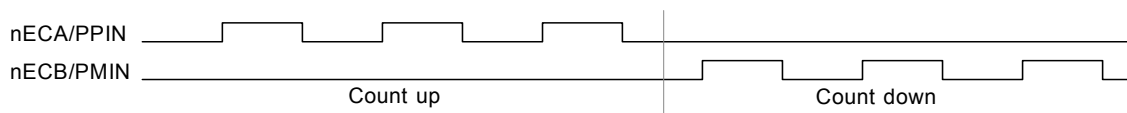
A/B quadrature pulse input mode can be set by setting the D9 (PINMD) bit of the WR2 register to 0. In this mode at the time of reset (WR1/D7=0), when A phase is advancing with positive logical pulses, the count is incremented and when the B-phase is advancing, the count is decremented. The count is incremented and decremented at the rising edge and falling edge of both signals. When the real position counter up/down reverse bit (WR1/D7) is set to 1, the up/down operation of the real counter is reversed.

In A/B quadrature pulse input mode, the input pulses can be divided into 1/2 or 1/4.



■ Up/down pulse input mode

By setting the D9 (PINMD) bit of the WR2 register to 1, a counter up/down pulse input mode can be set. nECA/PPIN is count up input and nECB/PMIN is count down input. The counter counts at the rising edge of the positive pulse.



Use the D9 (PINMD) bit of the WR2 register for selecting a pulse input mode and the D11 and D10 (PIND1,0) bits to set the division ratio of encoder 2-phase pulse input.

[Note] Time specification is applied to the pulse width and pulse cycle of input pulses. See Section 12.2.5 Input Pulse of Chapter 12.

2.6.4 Hardware Limit Signals

Hardware limit signals, nLMTP and nLMTM, are used for stopping the pulse output if the limit sensors of + and – directions are triggered.

When the limit signal and also the logical level are active, the command of sudden stop or decelerating stop can be set by bits D3 and D4 (HLMT+, HLMT–), and D2 (LMTMD) of register WR2.

2.6.5 Interface to Servo Motor Drivers

Enable / Disable and logical levels of the input signals for connecting servo motor drivers such as nINPOS (in-position input signal) and nALARM (alarm input signal) can be set by D15~12 bits of register WR2. nINPOS input signal responds to the in-position signal of servo motor driver.

When “enable” is set, and when the driving is finished, nINPOS will wait for the “active”. Then, the n-DRV bit of main status register RRO will return to 0.

nALARM input signal receives the alarm signal from servo motor drivers. When “enable” is set, nALARM signal will be monitored, and the D4 (alarm) bit of RR2 register is 1 when nALARM is active. The sudden stop will occur in the driving when this signal is active.

These input signals from servo motor drivers can be read by RR5 and RR6 registers.

A deviation counter clear and alarm reset signals are available as a servo motor driver output signal, assign nOUT7~0 to perform them.

2.6.6 Emergency Stop

Signal EMGN is able to perform the emergency stop function for both X and Y axes during the driving. Normally, this signal is kept on the Hi level. When it is falling to the Low level, all axes will stop immediately, and the D5 (EMG) bit of register RR2 (each axis) becomes 1. Please be noted that there is no way to select the logical level of EMGN signal.

Please check the following methods to perform the emergency stop function from the host CPU.

- a. Execute the sudden stop command for both of 2 axes at the same time...
Appoint both X and Y axes, then write the sudden stop command (27h) to register WR0.
- b. Reset software limit...
Write 800h to register WR0 to reset software limit.

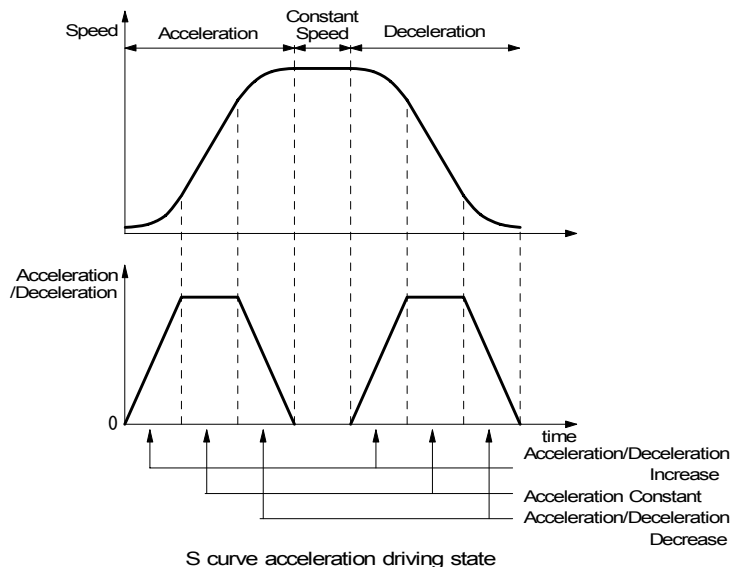
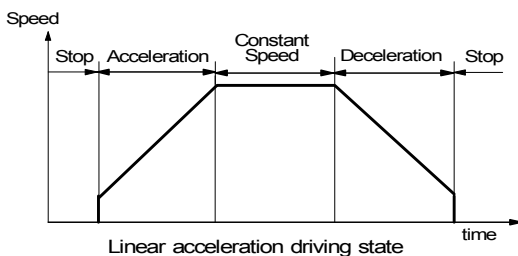
2.6.7 Status Output

The driving status of each axis will be output to main status register RR0 and status register nRR1 of each axis, also it can be output as signal that shares the same terminal with general purpose output signals. D7 (OUTSL) bit of register WR3 should be set 1 for drive status output. This bit will return 0 by reset and then turn to the terminal for general purpose output (nOUT 7~0).

nOUT4/DRIVE output signals and bits D1, 0 (n-DRV) of register RR0 can be used for drive / stop status output.

The driving status of acceleration / constant speed / deceleration will be output to bits D2 (ASND), D3 (CNST) and D4 (DSND), and also the signals TS/ASND, nOUT6/CNST and nOUT7/DSND will show the levels. Moreover in S curve acceleration / deceleration driving, the state of acceleration / constant speed / deceleration will be also shown to bits D5 (AASND), D6 (ACNST) and D7 (ADSND) and the signals nOUT0/ACASND and nOUT1/ACDSN.

| Drive Status | Status Register (Active : 1) | Output Signal (Active : Hi) |
|--------------------------------------|---------------------------------|--------------------------------|
| Drive | RR0 / D1, 0 (n-DRV) | nDRIVE |
| Acceleration | nRR1 / D2 (ASND) | nASND |
| Constant Speed | nRR1 / D3 (CNST) | nCNST |
| Deceleration | nRR1 / D4 (DSND) | nDSND |
| Acceleration / Deceleration Increase | nRR1 / D5 (AASND) | nACASND |
| Acceleration / Deceleration Constant | nRR1 / D6 (ACNST) | - |
| Acceleration / Deceleration Decrease | nRR1 / D7 (ADSND) | nACDSND |



2.6.8 General Purpose Input / Output Signal

In MCX312, there are 6 general purpose input pins, nIN5~0 and 8 general output pins, nOUT7~0, for each axis. However, during the multichip interpolation, nIN5~2 cannot be used cause they share the same terminals with the multichip interpolation signals. Also during the outputting, nOUT7~0 cannot be used cause they share the same terminals with the position comparison output and drive status output.

The status of XIN5~0 signals for X axis are displayed in D13~8 of RR4 register and YIN5~0 signals for Y axis are displayed in D13~8 of RR5 register respectively. Low level becomes 0 and Hi level becomes 1. Moreover, nIN5~0 signals are equipped with a function of input signal filter inside IC, please refer to 2.6.9.

Signals XOUT7~0 of the X axis can be output by setting the output level in each bit D7~0 of WR4 register and signals YOUT7~0 of the Y axis can be output by setting the output level in D15~8 of WR4 register, 0 set to Low level and 1 set to Hi level. When resetting, each bit of WR4 register will be cleared, then their output level will be set to Low level.

It is possible to use the general purpose output signals for motor driver current-OFF, deviation counting clear and alarm reset...

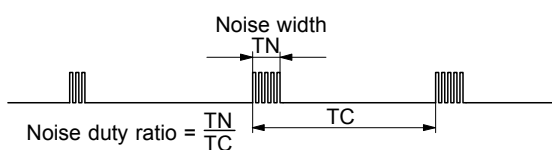
2.6.9 Input Signal Filter

This IC is equipped with an integral type filter in the input stage of each input signal. Figure 2.44 shows the filter configuration of each input signal of the X axis. The same circuit is provided to the Y axis also. The time constant of the filter is determined by the T oscillation circuit in the diagram. One time constant can be selected from eight time constants using the bits D15 to D13 (FL2 to FL0) of the nWR3 register. Using the bits D12 to D8 (FE4 to 0) of the nWR3 register, it is possible to set whether the filter function is enabled or the signal is passed through for a number of input signals. At reset, all the bits in the nWR3 register are cleared to 0 so that the filter function is disabled for all the input signals and the signals pass.

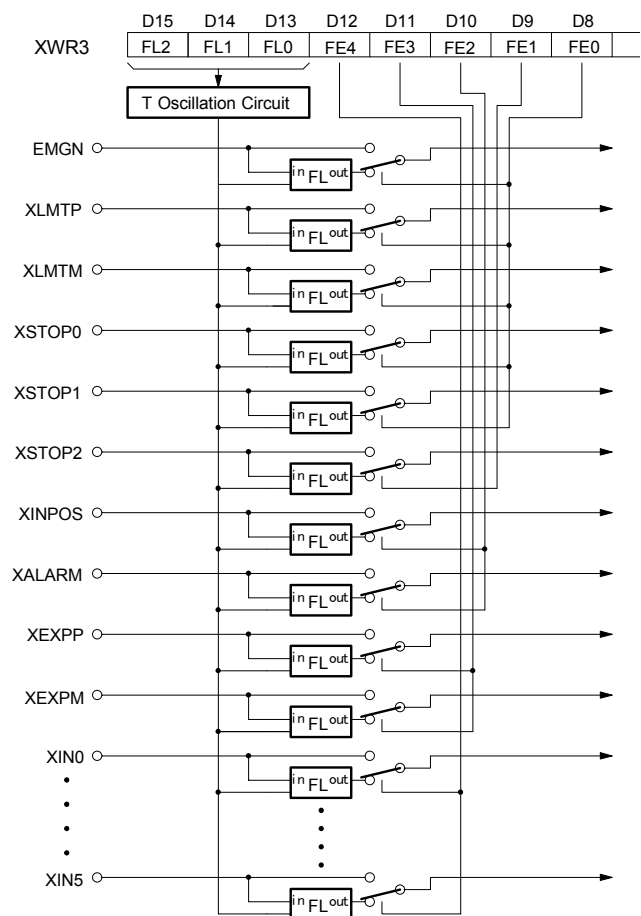
Select a filter time constant from eight stages as shown in the table below. When a time constant is increased, the removable maximum noise width increases, however, the signal delay time also increases. Therefore, set an appropriate value. Normally, set 2 or 3 for FL2 to FL0.

| FL2 ~ 0 | Removable maximum. noise width *1 | Input signal delay time |
|---------|-----------------------------------|-------------------------|
| 0 | 1.75μSEC | 2μSEC |
| 1 | 224μSEC | 256μSEC |
| 2 | 448μSEC | 512μSEC |
| 3 | 896μSEC | 1.024mSEC |
| 4 | 1.792mSEC | 2.048mSEC |
| 5 | 3.584mSEC | 4.096mSEC |
| 6 | 7.168mSEC | 8.192mSEC |
| 7 | 14.336mSEC | 16.384mSEC |

*1: Noise width



As the condition, the noise duty ratio (time ratio under which noise is generated in the signal) must be 1/4 or less.

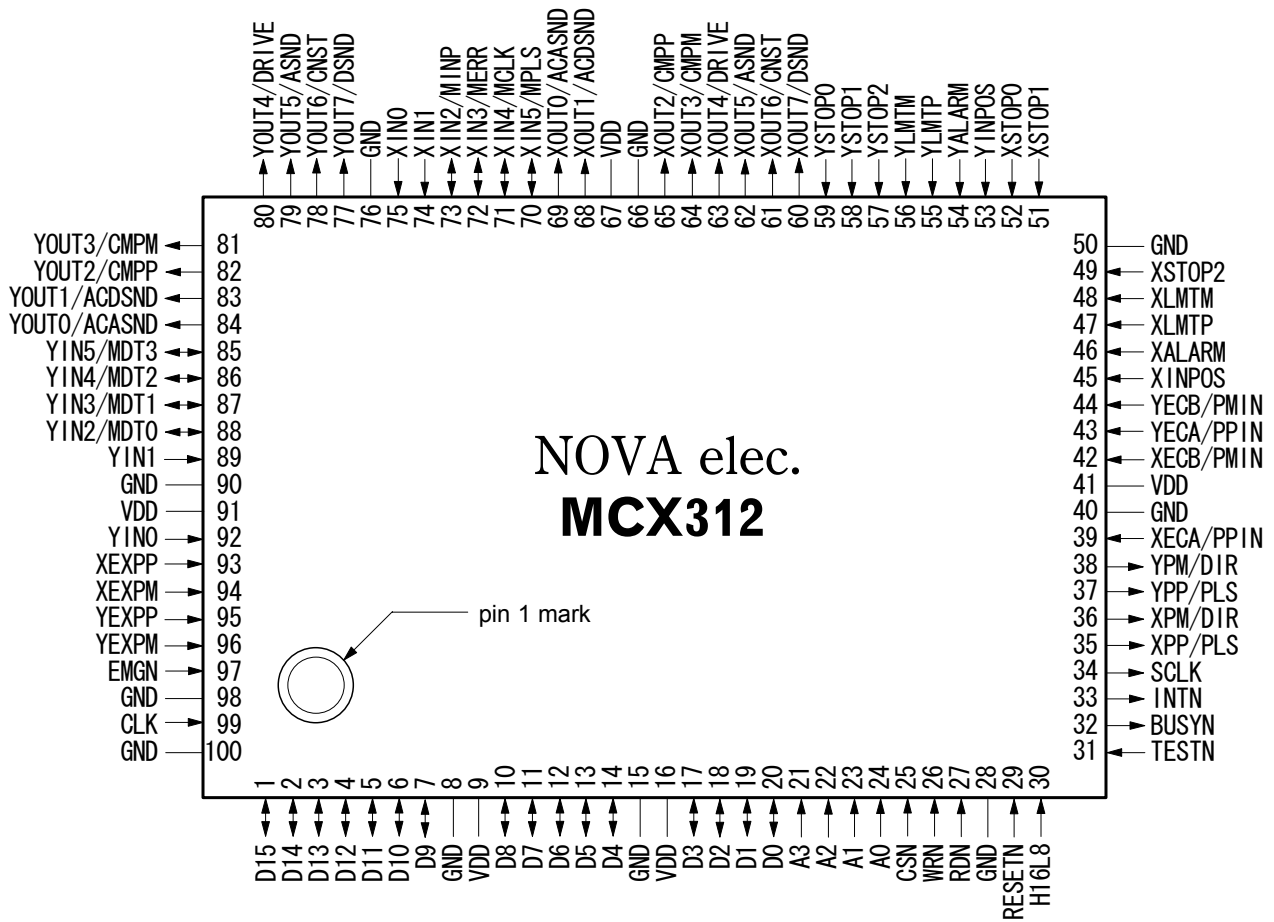


In bits D12 to D8 (FE4 to FE0) of the nWR3 register, set whether the filter function of each input signal is enabled or signals are passed through as shown below. When 1 is set in each bit, the filter function of the signal is enabled.

| Specification bit | Filter Enable signal |
|-------------------|---|
| NWR3/D8 (FE0) | EMGN ^{*2} , nLMTP, nLMTM, nSTOP0, nSTOP1 |
| D9 (FE1) | nSTOP2 |
| D10 (FE2) | nINPOS, nALARM |
| D11 (FE3) | nEXPP, nEXPM |
| D12 (FE4) | nIN0, nIN1, nIN2, nIN3, nIN4, nIN5 |

*2: The EMGN signal is set using the D8 bit of the WR3 register of the X axis.

3. Pin Assignments and Signal Description



See Chapter 14 for the 100-pin QFP package: 23.8×17.8mm, pin pitch: 0.65mm

■ Signal Description

Signals X○○○ and Y○○○ are input / output signals for X and Y axes, where n stands for X and Y. If the signals are named ○○○N, they are negative-active or low-active.

See the end of this chapter for description of input/output circuits. An integral filter circuit is available in the internal input column of this IC for the input signals with – F – symbol. See Section 2.6.9 for the filter function.

| Signal Name | Pin No. | Input/Output | Signal Description |
|------------------------|---------------------|------------------|--|
| CLK | 99 | Input A | Clock: clock signal for internal synchronous loop of MCX312 The standard frequency is 16 MHz. This signal is for drive speed, acceleration / deceleration and jerk. If the frequency setting is not 16 MHz, the setting values of speed and acceleration / deceleration are different. |
| D15~D0 | 1~7,10~14, 17~20 | Bi-directional A | DATA BUS: 3-state bi-direction 16-bit data bus When CSN=Low and RDN=Low, these signals are for outputting. Otherwise, they are high impedance inputs. If 8-bit data bus is used, D15~D8 can not be used, and D15~D8 should be pull up to + 5V through high impedance (about 100 kΩ). |
| A3~A0 | 21,22,23,24 | Input A | Address: address signal for host CPU to access the write / read registers A3 is used only when the 8-bit data bus is used. |
| CSN | 25 | Input A | Chip Select: input signal for selecting I/O device for MCX312. Set CSN to the Low level for data reading and writing. |
| WRN | 26 | Input A | Write Strobe: its level is Low while data is being written to MCX312. When WRN is Low, CSN and A3-A0 must be assured. When WRN is up (↑), the data will be latched in the write register, and while WRN is up (↑), the levels of D15~D0 should be assured. |
| RDN | 27 | Input A | Read Strobe: its level is Low while data is being read from MCX312. Only when CSN is on the low level, the selected read register data from A3~A0 address signals can be output from the data bus. |
| RESETN | 29 | Input A | Reset: reset (return to the initial setting) signal for MCX312. Setting RESETN to Low for more than 4 CLK cycles will reset MCX312. The RESETN setting is necessary when the power is on. [Note] If there is no clock input to MCX312, setting the RESETN to Low still cannot reset this IC. |
| H16L8 | 30 | Input A | Hi=16-bit, Low=8-bit: data bus width selection for 16-bit / 8-bit When the setting is Hi, 16-bit data bus is selected for processing the 16-bit data reading / writing in IC; when the setting is Low, 8-bit data bus (D7~D0) is active for data reading / writing. |
| TESTN | 31 | Input A | Test: terminal for internal-circuit test Please open, or connect it to + 5V. |
| BUSYN | 32 | Output B | Busy: reflecting the execution of the input command at this moment Once the command is written to MCX312, the process will take 2 CLK to 4CLK (250nsec for 16MHz) on the Low level. When BUSYN is on the Low level, the other written commands cannot be executed. |
| INTN | 33 | Output B | Interrupt: outputting an interrupt signal to the host CPU. If any interrupt factor occurs the interrupt, the level is Low; when the interrupt is released, it will return to the Hi-Z level. |
| SCLK | 34 | Output A | System Clock: SCLK=CLK/2 All the signals in MCX312 are controlled and synchronized by internal SCLK. When the output signal of each axis is latched, it can be used as an external signal source. [Note] There is no SCLK output when RESETN is on the Low level. |
| XPP/PLS YPP/PLS | 35 37 | Output A | Pulse +/Pulse: + direction dive pulse outputting When the reset is on the Low level, and while the driving is starting, DUTY 50% (at constant speed) of the plus drive pulses are outputting. + or – pulse mode is selectable. When the 1-pulse 1-direction mode is selected, this terminal is for drive output. |
| XPM/DIR YPM/DIR | 36 38 | Output A | Pulse –/Pulse: – direction dive pulse outputting When the reset is on the Low level, and while the driving is starting, DUTY 50% (at constant speed) of the plus drive pulses are outputting. + or – pulse mode is selectable. When the 1-pulse 1-direction mode is selected, this terminal is direction signal. |
| XECA/PPIN YECA/PPIN | 39 43 | Input A | Encoder-A/Pulse +in: signal for encoder phase-A input This input signal, together with phase-B signal, will make the Up / Down pulse transformation to be the input count of real position counter. When the Up / Down pulse input mode is selected, this terminal is for UP pulses input. Once the input pulse is up (↑), the real position counter is counting up. |
| XECB/PMIN YECB/PMIN | 42 44 | Input A | Encoder-B/Pulse -in: signal for encoder phase-B input This input signal, together with phase-A signal, will make the Up / Down pulse transformation to be the input count of real position counter. When the Up / Down pulse input mode is selected, this terminal is for DOWN pulses input. Once the input pulse is up (↑), the real position counter is counting down. |

| Signal Name | Pin No. | Input/Output | Signal Description |
|-------------------------------|--------------------------|------------------|--|
| XINPOS YINPOS | 45 53 | Input A - F - | In-position: input signal for servo driver in-position Enable / disable and logical levels can be set as commands. When "enable" is set, and after the driving is finished, this signal is active and standby. n-DVR bit of main status register returns to 0. |
| XALARM YALARM | 46 54 | Input A - F - | Servo Alarm: input signal for servo driver alarm Enable / disable and logical levels can be set as commands. When it is enable and when this signal is in its active level, the ALARM bit of RR2 register becomes 1. |
| XLMT+ YLMT+ | 47 55 | Input A - F - | OVER Limit +: signal of + direction over limit During the + direction drive pulse outputting, decelerating stop or sudden stop will be performed once this signal is active. When the filter function is disabled, the active pulse width must be 2CLK or more. When it is enable, and when this signal is in its active level, the HLMT+ of RR2 register becomes 1. |
| XLMT- YLMT- | 48 56 | Input A - F - | OVER Limit -: signal of - direction over limit During the - direction drive pulse outputting, decelerating stop or sudden stop will be performed once this signal is active. The active pulse width should be more than 2CLK. Decelerating stop / sudden stop and logical levels can be set during the mode selection. When it is enable, and when this signal is in its active level, the HLMT- of RR2 register becomes 1. |
| XSTOP2~0 YSTOP2~0 | 49, 51, 52 57, 58, 59 | Input A - F - | STOP 2~0: input signal to perform decelerating / sudden stop for each axis These signals can be used for HOME searching. The active pulse width should be more than 2CLK. Enable / disable and logical levels can be set for STOP2~STOP0. The signal status can be read from register RR4 / RR5. And nSTOP2 signal is equipped with a function that clears a real position counter value by (↑) signal depends on mode setting so that it is appropriate to input an encoder Z-phase signal. |
| XOUT7/DSND YOUT7/DSND | 60 77 | Output A | General Output 7 / Descend: general purpose output signals XOUT7~0 can output the 1/0 data of D7~0 in WR4 register and YOUT7~0 can output the 1/0 data of D15~8 in WR4 register to Hi / Low. They become Low when the IC is reset. When the drive status output mode is engaged, this signal can be used for reflecting the status of deceleration. While the driving command is executed and during the deceleration, it becomes Hi. |
| XOUT6/CNST YOUT6/CNST | 61 78 | Output A | General Output 6 / Constant: general purpose output signals (the operation is as same as nOUT7) When the drive status output mode is engaged, this signal can be used for reflecting the status of constant speed drive. While the driving command is executed and during the constant speed drive, it becomes Hi. |
| XOUT5/ASND YOUT5/ASND | 62 79 | Output A | General Output 5 / Ascend: general purpose output signals (the operation is as same as nOUT7) When the drive status output mode is engaged, this signal can be used for reflecting the status of acceleration. While the driving command is executed and during the acceleration, it becomes Hi. |
| XOUT4/DRIVE YOUT4/DRIVE | 63 80 | Output A | General Output 4 / Drive: general purpose output signals (the operation is as same as nOUT7) When the drive status output mode is engaged, this signal can be used for reflecting the status of drive. During outputting drive pulses, it becomes Hi. The DRIVE signal is set to a High level until nINPOS becomes active, while the nINPOS signal for the serve motor is enabled by mode selection. |
| XOUT3/CMPM YOUT3/CMPM | 64 81 | Output A | General Output 3 / Compare-: general purpose output signals (the operation is as same as nOUT7) When the drive status output mode is engaged, it becomes Hi if the value of logical / real position counter is smaller than that of COMP-; it becomes Low if the value of logical / real position counter is larger than that of COMP-. |
| XOUT2/CMPP YOUT2/CMPP | 65 82 | Output A | General Output2 / Compare+: general purpose output signals (the operation is as same as nOUT7) When the drive status output mode is engaged, it becomes Hi if the value of logical / real position counter is larger than that of COMP+; it becomes Low if the value of logical / real position counter is smaller than that of COMP+. |
| XOUT1/ACDSND YOUT1/ ACDSND | 68 83 | Output A | General Output 1 / Acceleration Descend: general purpose output signals (the operation is as same as nOUT7) When the drive status output mode is engaged, it becomes Hi if acceleration / deceleration of S curve acceleration / deceleration decreases. |
| XOUT0/ACASND YOUT0/ ACASND | 69 84 | Output A | General Output 0 / Acceleration Ascend: general purpose output signals (the operation is as same as nOUT7) When the drive status output mode is engaged, it becomes Hi if acceleration / deceleration of S curve acceleration / deceleration increases. |

| Signal Name | Pin No. | Input/Output | Signal Description |
|------------------------|--|---------------------------|--|
| XIN5/MPLS YIN5/MDT3 | 70 85 | Bi-directional B - F - | General Input 5 / : general purpose input signals The status of signal is displayed in a register, XIN5~0 of the X axis is displayed in D13~8 of RR4 register and YIN5~0 of the Y axis is displayed in D13~8 of RR5 register. The signal should be connected among chips and pulled up to +5V with 680Ω impedance when multichip axes interpolation is performed. MPLS input signal becomes pulse input signal for external pulse interpolation, The interpolation calculation starts by this signal (↓), and one pulse for each axis interpolation is output. |
| YIN4/MCLK YIN4/MDT2 | 71 86 | Bi-directional B - F - | General Input 4 / : general purpose input signals Reading operation is as same as nIN5) The signal should be connected among chips and pulled up to +5V with 680Ω impedance when multichip axes interpolation is performed. |
| XIN3/MERR YIN3/MDT1 | 72 87 | Bi-directional B - F - | General Input 3 / : general purpose input signals (Reading operation is as same as nIN5) The signal should be connected among chips and pulled up to +5V with 680Ω impedance when multichip axes interpolation is performed. |
| XIN2/MINP YIN2/MDT0 | 73 88 | Bi-directional B - F - | General Input 2 / : general purpose input signals (Reading operation is as same as nIN5) The signal should be connected among chips and pulled up to +5V with 680Ω impedance when multichip axes interpolation is performed. |
| XIN1 YIN1 | 74 89 | Input A - F - | General Input 1 / : general purpose input signals (Reading operation is as same as nIN5) |
| XIN0 YIN0 | 75 92 | Input A - F - | General Input 0 / : general purpose input signals (Reading operation is as same as nIN5) |
| XEXPP YEXPP | 93 95 | Input A - F - | External Operation +: + direction drive starting signal from external source When the fixed driving is commanded from an external source, +direction driving will start if this signal is down (↓). Otherwise, when the continuous driving is commanded from an external source, + driving will start if this signal is on the Low level. |
| XEXPM YEXPM | 94 96 | Input A - F - | External Operation -: - direction drive starting signal from external source When the fixed driving is commanded from an external source, - direction driving will start if this signal is down (↓). Otherwise, when the continuous driving is commanded from an external source, - driving will start if this signal is on the Low level. |
| EMGN | 97 | Input A - F - | Emergency Stop: input signal to perform the emergency stop for all axes When this signal is on the Low level, including the interpolation driving, every axis will stop the operation immediately. EMG bit of register RR2, of each axis, will become 1. The low level pulse width should be more than 2CLK. [Note] For this signal, its logical levels cannot be selected. |
| GND | 8, 15, 28, 40, 50, 66, 76, 90, 98, 100 | | Ground (0V) Terminal All of the 10 pins must be connected to 0V. |
| VDD | 9, 16, 41, 67, 91 | | + 5V Power Terminal. All of the 5 pins must be connected to +5V. |

■ Input/ Output Circuit

| | |
|------------------|---|
| Input A | Smith trigger input in TTL level, which is pulled up to VDD with high impedance. (dozens of kΩ ~ hundreds of kΩ) CMOS and TTL can be connected. The user should open, or pull up with + 5V if the input is not used. The signal with - F - symbol has an integral filter circuit in the internal input column of this IC. See Section 2.6.9 for the filter function. |
| Output A | It is CMOS level output, 4mA driving buffer (Hi level output current IOH=-4mA, VOH=2.4Vmin, Low level output current IOL=4mA, VOL=0.4Vmax). Up to 10 LSTTL can be driven. |
| Output B | It is open drain type output, 4mA driving buffer, (Low level output current IOL=4mA, VOL=0.4Vmax). Pull up to +5V with high impedance if this output is used. |
| Bi-directional A | Input side is TTL Smith trigger. Because there is no pull high resistor for those signals in this IC, the user should pull up the data bus with high impedance. The user should pull up to +5V with high impedance (about 100 kΩ) when bits D15~D8 are not used. Output side is CMOS level output, 8mA driving buffer (Hi level output current IOH=-8mA, VOH=2.4Vmin, Low level output current IOL=8mA, VOL=0.4Vmax). |
| Bi-directional B | Input side is Smith trigger input in TTL level, which is pulled up to VDD with high impedance. (dozens of kΩ ~ hundreds of kΩ) Output side is open drain type output, generally OFF and will be activated during multichip axes interpolation. When signals are connected among chips in multichip axes interpolation, please shorten the length of wiring as far as possible and do not cross other signal path. The signal with - F - symbol has an integral filter circuit in the internal input column of this IC. See Section 2.6.9 for the filter function. |

■ Notes for the Design of Circuitry**a. De-coupling Capacitor**

Please connect VDD and GND with one or two De-coupling capacitors (about 0.1 μ F).

b. Noise Generated by Terminal Induction

The noise will exist because the inductance is in these pins. The user can add a capacitor (10-100pF) to pins to reduce the noise.

c. Reflection on Transfer Path

The load capacity for outputting types A, B, and bi-direction type A and B are 20-50pf. So, the reflection will happen if the PCB wiring is more than 60cm.

4. Register

This chapter indicates the user how to access all the registers in MCX312, and what are the mapping addresses of these registers. Please refer to Chapter 2.4.3 for the registers (BP1P/M, BP2P/M) of bit pattern interpolation.

4.1 Register Address by 16-bit Data Bus

As shown is the table below, when 16-bit data bus is used, the access address of read / write register is 8-bit.

■ Write Register in 16-bit Data Bus

All registers are 16-bit length.

| Address A2 A1 A0 | Symbol | Register Name | Contents |
|---------------------|------------------------|-----------------------------|--|
| 0 0 0 | WR0 | Command Register | for setting axis assignment and command |
| 0 0 1 | XWR1 | X axis mode register 1 | for setting the logical levels of external decelerating stop, enable / disable, the valid / invalid of interrupt for each axis, and the mode in the real position counter. |
| | YWR1 | Y axis mode register 1 | |
| 0 1 0 | XWR2 | X axis mode register 2 | for setting the limit signal mode, driving pulse mode, encoder input signal mode, the logical levels and enable / disable of servo motor signal for each axis, and the action mode in the real position counter. |
| | YWR2 | Y axis mode register 2 | |
| 0 1 1 | BP1P | BP1P register | for setting the + direction bit data of the X axis in bit pattern interpolation |
| | XWR3 | X axis mode register 3 | for setting the manual deceleration, individually decelerating, and S-curve acceleration/ deceleration mode for each axis, external operation mode, and input signal filter. |
| YWR3 | Y axis mode register 3 | | |
| 1 0 0 | BP1M | BP1M register | for setting the - direction bit data of the X axis in bit pattern interpolation |
| | WR4 | Output register | for setting the general output nOUT7 ~ 0 |
| 1 0 1 | BP2P | BP2P register | for setting the + direction bit data of the Y axis in bit pattern interpolation |
| | WR5 | Interpolation mode register | for setting axis assignment and the constant vector speed mode, step output mode and interrupt |
| 1 1 0 | BP2M | BP2M register | for setting the - direction bit data of the Y axis in bit pattern interpolation |
| | WR6 | Data writing register 1 | for setting the low word 16-bit (D15-D0) for data writing. |
| 1 1 1 | WR7 | Data writing register 2 | for setting the high word 16-bit (D31-D16) for data writing. |

- Each axis is with WR1, WR2 and WR3 mode registers. Each register is for 2-axis data writing (at the same address). Before those registers have been accessed, the host CPU should specify which axis is going to be accessed by writing a NOP command into WR0.
- The register for bit pattern interpolation is BP1P, BP2P, BP1M, BP2M. After the resetting, the data writing cannot be performed, until the enable command (36h) is engaged by BP register. After the command 36h is enabled, the data writing cannot be performed in nWR2~3. So, the disable command (37h) should be engaged after the bit pattern interpolation data is written.
- The bits of nWR1, nWR2, nWR3, nWR4 and nWR5 will be cleared to 0 after the resetting. It will be unknown for other registers.

■ Read Register in 16-bit Data Bus

All registers are 16-bit length.

| Address A2 A1 A0 | Symbol | Register Name | Contents |
|---------------------|--------|--------------------------|--|
| 0 0 0 | RR0 | Main status register | error status, driving status, ready for interpolation, quadrant for circle interpolation and the stack of BP |
| 0 0 1 | XRR1 | X axis status register 1 | comparison result, acceleration state, and jerk state. finishing status |
| | YRR1 | Y axis status register 1 | |
| 0 1 0 | XRR2 | X axis status register 2 | error message |
| | YRR2 | Y axis status register 2 | |
| 0 1 1 | XRR3 | X axis status register 3 | interrupt message |
| | YRR3 | Y axis status register 3 | |

| | | | |
|-------|-----|-------------------------|--|
| 1 0 0 | RR4 | Input register 1 | I/O input for X axis |
| 1 0 1 | RR5 | Input register 2 | I/O input for Y axis |
| 1 1 0 | RR6 | Data reading register 1 | low word of data register (D15 ~ D0) |
| 1 1 1 | RR7 | Data reading register 2 | high word of data register (D31 ~ D16) |

- Each axis is with RR1, RR2 and RR3 mode registers. Each register is for 2-axis data writing (at the same address). Before those registers have been accessed, the host CPU should specify which axis is going to be accessed by writing a NOP command into WR0.

4.2 Register Address by 8-bit Data Bus

In case of the 8-bit data bus access, the 16-bit data bus can be divided into high and low word byte. As shown in the table below, xxxxL is the low word byte (D7~D0) of 16-bit register xxxx, xxxxH is the high word byte (D15~8) of 16-bit register xxxx. Only for the command register (WR0L, WR0H), the user must write to the high word byte (WR0H), then to the low word byte (WR0L).

■ Write Register in 8-bit Data Bus

| Address A3 A2 A1 A0 | Write Register |
|------------------------|--------------------|
| 0 0 0 0 | WR0L |
| 0 0 0 1 | WR0H |
| 0 0 1 0 | XWR1L,YWR1L |
| 0 0 1 1 | XWR1H,YWR1H |
| 0 1 0 0 | XWR2L,YWR2L, BP1PL |
| 0 1 0 1 | XWR2H,YWR2H, BP1PH |
| 0 1 1 0 | XWR3L,YWR3L, BP1ML |
| 0 1 1 1 | XWR3H,YWR3H, BP1MH |
| 1 0 0 0 | WR4L,BP2PL |
| 1 0 0 1 | WR4H,BP2PH |
| 1 0 1 0 | WR5L,BP2ML |
| 1 0 1 1 | WR5H,BP2MH |
| 1 1 0 0 | WR6L,BP3PL |
| 1 1 0 1 | WR6H,BP3PH |
| 1 1 1 0 | WR7L,BP3ML |
| 1 1 1 1 | WR7H,BP3MH |

■ Read Register in 8-bit Data Bus

| Address A3 A2 A1 A0 | Read Register |
|------------------------|---------------|
| 0 0 0 0 | RR0L |
| 0 0 0 1 | RR0H |
| 0 0 1 0 | XRR1L,YRR1L |
| 0 0 1 1 | XRR1H,YRR1H |
| 0 1 0 0 | XRR2L,YRR2L |
| 0 1 0 1 | XRR2H,YRR2H |
| 0 1 1 0 | XRR3L,YRR3L |
| 0 1 1 1 | XRR3H,YRR3H |
| 1 0 0 0 | RR4L |
| 1 0 0 1 | RR4H |
| 1 0 1 0 | RR5L |
| 1 0 1 1 | RR5H |
| 1 1 0 0 | RR6L |
| 1 1 0 1 | RR6H |
| 1 1 1 0 | RR7L |
| 1 1 1 1 | RR7H |

4.3 Command Register: WR0

Command register is used for the axis assignment and command registration for each axis in MCX312. The register consists of the bit for axis assignment, bit for setting command code, and bit for command resetting.

After the axis assignment and command code have been written to the register, this command will be executed immediately. The data such as drive speed setting and data writing command must be written to registers WR6 and WR7 first. Otherwise, when the reading command is engaged, the data will be written and set, through IC internal circuit, to registers RR6 and RR7.

When using the 8-bit data bus, the user should write data into the high word byte (H), then low word byte (L).

It requires 250 nSEC (maximum) to access the command code when CLK=16MHz. The input signal BUSYN is on the Low level at this moment. Please don't write the next command into WR0 before BUSYN return to the Hi level.

| | | | | | | | | | | | | | | | | |
|-----|-----------------|-----|-----|------------------|-----|-----|----|----|--------------|----|----|-----------------|----|----|----|----|
| WR0 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | RESET | 0 | 0 | 0 | 0 | 0 | Y | X | 0 | 0 | | | | | | |
| | Axis Assignment | | | | | | | | Command Code | | | | | | | |

D5 ~ 0 Command code setting
Please refer to chapter 5 and the chapters following for further description of command codes.

D9 ~ 8 Axis assignment
When the bits of the axis are set to 1, the axis is assigned. The assignment is not limited only for one axis, but for multi-axes simultaneously. It is possible to write the same parameters also. However, the data reading is only for one assigned axis.
Whenever the interpolation is commanded, the bits of the assigned axis (axes) should be set 0.

D15 RESET IC command resetting
When this bit is set to 1, but others are 0, the IC will be reset after command writing. After command writing, the BUSYN signal will be on the Low level within 875 nSEC (When CLK=16 MHz) maximum.

When 8-bit data bus is used, the reset is activated when the command (80h) is written to register WR0H.

RESET bit should be set to 0 when the other commands are written.

4.4 Mode Register1: WR1

Each axis is with mode register WR1. The axis specified by NOP command or the condition before decides which axis' s register will be written.

The register consists of the bit for setting enable / disable and enable logical levels of input signal STOP2~STOP0 (decelerating stop / sudden stop during the driving) and bit for occurring the interrupt enable / disable.

Once SP2~SP0 are active, when the fixed / continuous driving starts, and also when STOP signal becomes the setting logical level, the decelerating stop will be performed during the acceleration / deceleration driving and the sudden stop will be performed during the constant speed driving.

| | | | | | | | | | | | | | | | | |
|-----|--------------------------|-------|-------|------------------|------|------|------|------|---------------------------|-------|-------|-------|-------|-------|-------|-------|
| WR1 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| | D-END | C-STA | C-END | P≧C+ | P<C+ | P<C- | P≧C- | SMOD | EPINV | EPCLR | SP2-E | SP2-L | SP1-E | SP1-L | SP0-E | SP0-L |
| | Interrupt Enable/Disable | | | | | | | | Driving Stop Input Signal | | | | | | | |

- D5,3,1 SPM-E The bit for setting enable / disable of driving stop input signal STOPm 0: disable, 1: enable
- D4,2,0 SPM-L The bit for setting enable logical levels for input signal STOPm 0: stop on the Low level, 1: stop on the Hi level
- D6 EPCLR When driving stops triggered by the nSTOP2 signal, the real position counter is cleared. When the nSTOP2 signal is changed to the Active level while this bit is set to 1, the driving stops and the real position counter (EP) is cleared. The WR1/D5(SP2-E) bit must be set to 1 and the Enable level must be set in the WR1/D4(SP2-L) bit.
- D7 EPINV Reverse increase / decrease of real position counter.

| D7 (EPINV) | Input pulse mode | Increase / Decrease of real position counter |
|------------|--------------------|--|
| 0 | A / B -phase mode | Count up when A -phase is advancing Count down when B -phase is advancing |
| | Up-Down pulse mode | Count up when PPIN pulse input Count down when PMIN pulse input |
| 1 | A / B -phase mode | Count up when B -phase is advancing Count down when A -phase is advancing |
| | Up-Down pulse mode | Count up when PMIN pulse input Count down when PPIN pulse input |

- D8 SMOD Setting for prioritizing to reach specified drive speed during S curve acceleration / deceleration driving. 1: enable

For the following bits, the interrupt is set: 1: enable, 0: disable

- D9 $P \geq C-$ Interrupt occurs when the value of logical / real position counter is larger than or equal to that of COMP- register
- D10 $P < C-$ Interrupt occurs when the value of logical / real position counter is smaller than that of COMP- register
- D11 $P < C+$ Interrupt occurs when the value of logical / real position counter is smaller than that of COMP+ register
- D12 $P \geq C+$ Interrupt occurs when the value of logical / real position counter is larger than or equal to that of COMP+ register
- D13 C-END Interrupt occurs at the end of the constant speed drive during an acceleration / deceleration driving
- D14 C-STA Interrupt occurs at the start of the constant speed drive during an acceleration / deceleration driving
- D15 D-END Interrupt occurs when the driving is finished

D15~D0 will be set to 0 while resetting.

4.5 Mode Register2: WR2

Each axis is with mode register WR2. The axis specified by NOP command or the condition before decides which axis' s register will be written.

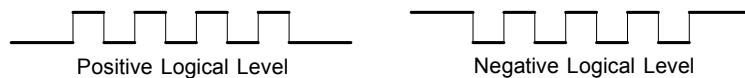
WR2 can be used for setting: (1). external limit inputs, (2). driving pulse types, (3). encoder signal types, and (4). the feedback signals from servo drivers.

| WR2 | H | | | | | | | | L | | | | | | | |
|-----|--------|--------|--------|--------|-------|-------|-------|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| | INP -E | INP -L | ALM -E | ALM -L | PIND1 | PIND0 | PINMD | DIR -L | PLS -L | PLSMD | CMPSL | HLMT- | HLMT+ | LMTMD | SLMT- | SLMT+ |

- D0 SLMT+ Enable / disable setting for COMP+ register which is used as the + direction software limit 1: enable, 0: disable
 Once it is enabled during the + direction driving, if the value of logical / real position counter is larger than that of COMP+, the decelerating stop will be performed. The D0 (SLMT+) bit of register RR2 will become 1. Under this situation, further written + direction driving commands will not be executed.
Note: When a position counter variable ring is used, a software over run limit cannot be used.
- D1 SLMT- Enable / disable setting for COMP- register which is used as the - direction software limit 1: enable, 0: disable
 Once it is enabled during the - direction driving, if the value of logical / real position counter is smaller than that of COMP-, the decelerating stop will be performed. The D1 (SLMT-) bit of register RR2 will become 1. Under this situation, further written - direction driving commands will not be executed.
- D2 LMTMD The bit for controlling stop type when the hardware limits (nLMTP and nLMTM input signals) are active 0: sudden stop, 1: decelerating stop
- D3 HLMT+ Setting the logical level of + direction limit input signal (nLMTP) 0: active on the Low level, 1: active on the Hi level
- D4 HLMT- Setting the logical level of - direction limit input signal (nLMTM) 0: active on the Low level, 1: active on the Hi level
- D5 CMPSL Setting if real position counter or logical position counter is going to be compared with COMP +/- register 0: logical position counter, 1 : real position counter
- D6 PLSMD Setting output pulse type 0: independent 2-pulse type, 1: 1-pulse 1-direction type

When independent 2-pulse type is engaged, + direction pulses are output through the output signal nPP/PLS, and - direction pulses through the output signal nPM/DIR.
 When 1-pulse 1-direction type is engaged, + and - directions pulses are output through the output signal nPP/PLS, and nPM/DIR is for direction signals.
 [Note] Please refer to Chapter 13.2 and 13.3 for the output timing of pulse signal (nPLS) and direction signal (nDIR) when 1-pulse 1-direction type is engaged.

- D7 PLS-L Setting logical level of driving pulses 0: positive logical level, 1: negative logical level

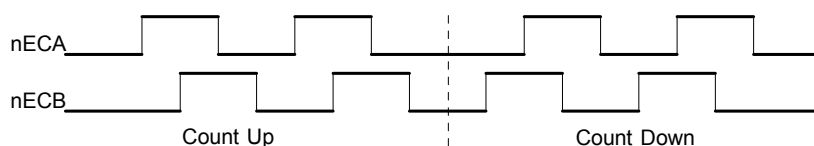


- D8 DIR-L Setting logical level of the direction (nPM/DIR) output signal for 1-pulse mode DIR-L

| D8 (DIR-L) | + direction | - direction |
|------------|-------------|-------------|
| 0 | Low | Hi |
| 1 | Hi | Low |

- D9 PINMD Setting the type of encoder input signals (nECA/PPIN and nECB/PMIN) 0: quadrature pulse input type 1: Up / Down pulse input type

When quadrature pulse input type is engaged and nECA signal goes faster 90 degree phase than nECB signal does, it's "count up" and nECB signal goes faster 90 degree phase than nECA signal does, it's "count down"



When Up / Down pulse input type is engaged, nECA/PPIN is for “count up” input, and nECB/PMIN is for “count down” input. So, it will count up when the positive pulses go up (↑).

D11,10 PIND1,0 The division setting for quadrature encoder input.

| D11 | D10 | Division |
|-----|-----|----------|
| 0 | 0 | 1/1 |
| 0 | 1 | 1/2 |
| 1 | 0 | 1/4 |
| 1 | 1 | Invalid |

Up / down pulse input is not available.

D12 ALM-L Setting active level of input signal nALARM 0: active on the Low level, 1: active on the Hi level

D13 ALM-E Setting enable / disable of servo alarm input signal nALARM 0: disable, 1: enable
When it is enabled, MCX312 will check the input signal. If it is active, D14 (ALARM) bit of RR2 register will become 1. The driving stops.

D14 INP-L Setting logical level of nINPOS input signal 0: active on the Low level, 1: active on the Hi level

D15 INP-E Setting enable/disable of in-position input signal nINPOS from servo driver 0: disable, 1: enable
When it is enabled, bit n-DRV of RR0 (main status) register does not return to 0 until nINPOS signal is active after the driving is finished.

D15~D0 will be set to 0 while resetting.

4.6 Mode Register3: WR3

Each axis is with mode register WR3. The axis specified by NOP command or the condition before decides which axis' s register will be written.

WR3 can be used for manual deceleration, individual deceleration, S-curve acceleration / deceleration, the setting of external operation mode, the setting of input signal filter, and so on.

| WR3 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
|-----|-----|-----|-----|------------------|-----|-----|-----|-----|----|----|----|-----------------|-------|------|-------|-------|
| | FL2 | FL1 | FL0 | FE4 | FE3 | FE2 | FE1 | FE0 | | | | EX0P1 | EX0P0 | SACC | DSNDE | MANLD |

D0 MANLD Setting manual / automatic deceleration for the fixed acceleration / deceleration driving
0: automatic deceleration, 1: manual deceleration
The decelerating point should be set if the manual deceleration mode is engaged.

D1 DSNDE Setting decelerating rate which is in accordance with the rate of the acceleration or an individual decelerating rate
0: acceleration, 1: deceleration

When 0 is set, acceleration value is used as acceleration / deceleration during acceleration / deceleration driving. When 1 is set, acceleration value is used during acceleration driving and deceleration value is used during deceleration driving.

1 should be set for non-symmetrical trapezoidal acceleration / deceleration driving.

D2 SACC Setting trapezoidal driving / S-curve driving
0: trapezoidal driving, 1: S-curve driving
Before S-curve driving is engaged, jerk (K) should be set.

D4,3 EXOP1,0 Setting the external input signals (nEXPP, nEXPM) for driving

| D4 (EXOP1) | D3 (EXOP0) | |
|------------|------------|---------------------------|
| 0 | 0 | external signals disabled |
| 0 | 1 | continuous driving mode |
| 1 | 0 | fixed driving mode |
| 1 | 1 | manual pulsar mode |

When the continuous driving mode is engaged, the + direction drive pulses will be output continuously once the nEXPP signal is on the Low level; the – direction pulses will be output continuously once the nEXPM signal is on the Low level. When the fixed driving mode is engaged, the + direction fixed driving starts once the nEXPP signal is falling to the Low level from the Hi level; the – direction fixed driving starts once the nEXPM signal is falling to the Low level from the Hi level.

In manual pulsar mode, fixed driving in the + direction is activated at \uparrow of the nEXPP signal when the nEXPM signal is at the Low level. The fixed driving is activated at \downarrow of the nEXPP signal when the nEXPM signal is at the Low level.

D5 AVTRI Prevent triangle waveforms during fixed driving at the trapezoidal acceleration / deceleration. 0: disable, 1: enable.

[Note] WR3/D5 bit should be reset to 0 when continuous driving is performed after fixed driving.

D6 VRING Enable the variable ring function of logical position and real position counter. 0: disable, 1: enable.

D7 OUTSL Driving status outputting or used as general purpose output signals (nOUT7~0)

0: nOUT7~0: general purpose output

The setting of each bit in WR4 register will be output to nOUT7~0.

1: nOUT7~0: driving status output (see the table below)

| Signal Name | Output Description |
|--------------|---|
| nOUT0/ACASND | When acceleration or deceleration of S curve acceleration / deceleration increases, the level becomes Hi. |
| nOUT1/ACDSND | When acceleration or deceleration of S curve acceleration / deceleration decreases, the level becomes Hi. |
| nOUT2/CMPP | Hi: if logical / real position counter \geq COMP+ register Low : if logical / real position counter $<$ COMP+ register |
| nOUT3/CMPM | Hi: if logical / real position counter $<$ COMP– register Low: if logical / real position counter \geq COMP– register |
| nOUT4/DRIVE | When drive pulse is outputting, the level becomes Hi. |
| nOUT5/ASND | When the driving command is engaged, the level becomes Hi once the driving status is in acceleration. |
| nOUT6/CNST | When the driving command is engaged, the level becomes Hi once the driving status is in constant speed driving. |
| nOUT7/DSND | When the driving command is engaged, the level becomes Hi once the driving status is in deceleration. |

D12~8 FE4~0 Set whether the input signal filter function enables or signal passes through. 0: through, 1: enable.

| Specification bit | Filter Enable signal |
|-------------------|---|
| D8 FE0 | EMGN ^{*2} , nLMTP, nLMTM, nSTOP0, nSTOP1 |
| D9 FE1 | nSTOP2 |
| D10 FE2 | nINPOS, nALARM |
| D11 FE3 | nEXPP, nEXPM |
| D12 FE4 | nIN0, nIN1, nIN2, nIN3, nIN4, nIN5 |

*2: The EMGN signal is set using the D8 bit of the WR3 register of the X axis.

D15~13 FL2~0 Set a time constant of the filter.

| FL2 ~ 0 | Removable maximum noise width | Input signal delay time |
|---------|-------------------------------|-------------------------|
| 0 | 1.75μSEC | 2μSEC |
| 1 | 224μSEC | 256μSEC |
| 2 | 448μSEC | 512μSEC |
| 3 | 896μSEC | 1.024mSEC |
| 4 | 1.792mSEC | 2.048mSEC |
| 5 | 3.584mSEC | 4.096mSEC |
| 6 | 7.168mSEC | 8.192mSEC |
| 7 | 14.336mSEC | 16.384mSEC |

D15~D0 will be set to 0 while resetting.

4.7 Output Register: WR4

This register is used for setting the general purpose output signals nOUT7~0. This 16-bit register locates 8 output signals of each axis. It can be also used as a 16-bit general purpose output. It is Low level output when the bit is set 0, and Hi level output when the bit is set 1.

| | | | | | | | | | | | | | | | | |
|-----|-------|-------|-------|------------------|-------|-------|-------|-------|-------|-------|-------|-----------------|-------|-------|-------|-------|
| WR4 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | YOUT7 | YOUT6 | YOUT5 | YOUT4 | YOUT3 | YOUT2 | YOUT1 | YOUT0 | XOUT7 | XOUT6 | XOUT5 | XOUT4 | XOUT3 | XOUT2 | XOUT1 | XOUT0 |

D15~D0 will be set to 0 while resetting, and nOUT7~0 signals become Low level.

4.8 Interpolation Mode Register: WR5

This register is used for setting constant vector speed mode, multichip interpolation mode, 1-step interpolation mode and interrupt during the interpolation.

| | | | | | | | | | | | | | | | | |
|-----|-----------|-------|-------|------------------|-----------|------|-----------------------|-------|----|----|----|-----------------|----|----|----|----|
| WR5 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | BPINT | CIINT | CMPLS | EXPLS | MLT1 | MLT0 | LSPD1 | LSPD0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Interrupt | | Step | | multichip | | Constant Vector Speed | | | | | | | | | |

D9,8 LSPD1,0 Constant vector speed mode setting of interpolation driving

| D9 | D8 | Code (Binary) |
|----|----|-------------------------------|
| 0 | 0 | constant vector speed invalid |
| 0 | 1 | 2-axis constant vector speed |
| 1 | 0 | (setting not available) |
| 1 | 1 | (setting not available) |

When 2-axis constant vector speed mode is engaged, the user should set the range (R) of the Y axis to be 1.414 times of the range (R) of the X axis.

D11,10 MLT1,0 Multichip interpolation mode setting

| D11 | D10 | Setting |
|-----|-----|---------------------------------------|
| 0 | 0 | disable multichip interpolation |
| 0 | 1 | main chip |
| 1 | 0 | sub chip X, Y (Both X and Y axes use) |
| 1 | 1 | sub chip (Only X axis) |

D12 EXPLS When it is 1, the external (MPLS) controlled single step interpolation mode is engaged.

- D13 CMPLS When it is 1, the command controlled single step interpolation mode is engaged.
- D14 CIINT Interrupt enable / disable setting during interpolation 0: disable 1: enable
- D15 BPINT Interrupt enable / disable setting during bit-pattern interpolation 0: disable 1: enable

D15~D0 will be set to 0 while resetting.

4.9 Data Register: WR6/WR7

Data registers are used for setting the written command data. The low-word data-writing 16-bit (WD15~WD0) is for register WR6 setting, and the high-word data-writing 16-bit (WD31~WD16) is for register WR7 setting.

| | | | | | | | | | | | | | | | | |
|-----|------|------|------|------------------|------|------|-----|-----|-----|-----|-----|-----------------|-----|-----|-----|-----|
| WR6 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | WD15 | WD14 | WD13 | WD12 | WD11 | WD10 | WD9 | WD8 | WD7 | WD6 | WD5 | WD4 | WD3 | WD2 | WD1 | WD0 |

| | | | | | | | | | | | | | | | | |
|-----|------|------|------|------------------|------|------|------|------|------|------|------|-----------------|------|------|------|------|
| WR7 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | WD31 | WD30 | WD29 | WD28 | WD27 | WD26 | WD25 | WD24 | WD23 | WD22 | WD21 | WD20 | WD19 | WD18 | WD17 | WD16 |

The user can write command data with a designated data length into the write register. It does not matter to write WR6 or WR7 first (when 8-bit data bus is used, the registers are WR6L, WR6H, WR7L and WR7H).

The written data is binary formatted; 2' complement is for negatives.

For command data, the user should use designated data length. For instance, the circular interpolation of the finish point should be set by a signed 32-bit format with the data length of 4 bytes, although its calculatable data range is from -8,388,608 to +8,388,607 signed 24-bit format.

The contents of WR6 and WR7 are unknown while resetting.

4.10 Main Status Register: RR0

This register is used for displaying the driving and error status of each axis. It also displays interpolation driving, ready signal for continuous interpolation, quadrant of circular interpolation and stack counter of bit pattern interpolation.

| | | | | | | | | | | | | | | | | |
|-----|-----|-------|-------|------------------|-------|-------|-------|-------|----|----|-------|-----------------|----|----|-------|-------|
| RR0 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | - | BPSC1 | BPSC0 | ZONE2 | ZONE1 | ZONE0 | CNEXT | I-DRV | 0 | 0 | Y-ERR | X-ERR | 0 | 0 | Y-DRV | X-DRV |

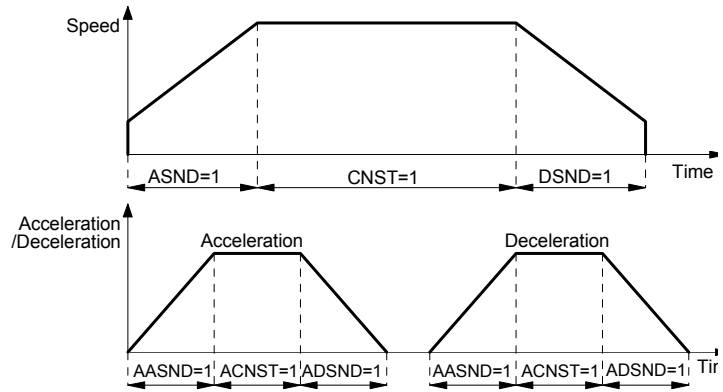
Error Status of Each Axis
Driving Status of Each Axis

- D1, 0 n-DRV Displaying driving status of each axis
When the bit is 1, the axis is an outputting drive pulse. ; when the bit is 0, the driving of the axis is finished. Once the in-position input signal nINPOS for servomotor is active, nINPOS will return to 0 after the drive pulse output is finished.
- D5, 4 n-ERR Displaying error status of each axis
If any of the error bits (D6~D0) of each axis's RR2 register and any of the error-finish bits (D15~D12) of each axis' s RR1 register becomes 1, this bit will become 1.
- D8 I-DRV Displaying interpolation driving status
While the interpolation drive pulses are outputting, the bit is 1.
- D9 CNEXT Displaying the possibility of continuous interpolation data writing

D6 ACNST In S-curve, it becomes 1 when acceleration / deceleration keeps constant speed.

D7 ADSND In S-curve, it becomes 1 when acceleration / deceleration decreases.

D10~8 STOP2~0 If the driving is stopped by one of external decelerating stop signals (nSTOP2 ~ 0), it will become 1.



D12 LMT+ If the driving is stopped by +direction limit signal (nLMTP), it will become 1.

D13 LMT- If the driving is stopped by -direction limit signal (nLMTM), it will become 1.

D14 ALARM If the driving is stopped by nALARM from servo drivers, it will become 1.

D15 EMG If the driving is stopped by external emergency signal (EMGN), it will become 1.

■ The Status Bits of Driving Finishing

These bits are keeping the factor information of driving finishing. The factors for driving finishing in fixed driving and continuous driving are shown as follows:

- a. when all the drive pulses are output in fixed driving,
- b. when deceleration stop or sudden stop command is written,
- c. when software limit is enabled, and is active,
- d. when external deceleration signal is enabled, and active,
- e. when external limit switch signals (nLMTP, nLMTM) become active,
- f. when nALARM signal is enabled, and active, and
- g. when EMGN signal is on the Low level.

Above factors “a.” and “b.” can be controlled by the host CPU, and factor “c.” can be confirmed by register RR2 even the driving is finished. As for factors “d.” ~ “g.”, the error status is latched in RR2 until next driving command or a clear command (25h) is written.

After the driving is finished, if the error factor bits D15~D12 become 1, n-ERR bit of main status register RRO will become 1.

Status bit of driving finishing can be cleared when next driving command is written, or when the finishing status clear command (25h) is used.

4.12 Status Register 2: RR2

Each axis is with status register 2. The axis specified by NOP command or the condition before decides which axis’ s register will be read.

This register is for reflecting the error information. When an error occurs, the error information bit is set to 1. When one or more of D6 to D0 bits of RR2 register are 1, n-ERR bits of main status register RR0 become 1.

| | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|---|---|---|---|----|------|-----|-------|-------|-------|-------|-------|----|----|----|
| RR2 | D15 | D14 | D13 | D12 | H | | | | D7 | D6 | D5 | L | | | | D3 | D2 | D1 | D0 |
| | - | - | - | - | - | - | - | - | - | MULT | EMG | ALARM | HLMT- | HLMT+ | SLMT- | SLMT+ | | | |

D0 SLMT+ During the + direction driving, when logical / real position counter ≥ COMP+ (COMP+ enabled, and used as software limit)

| | | |
|----|-------|---|
| D1 | SLMT- | During the – direction driving, when logical / real position counter \leq COMP- (COMPEnabled, and used as software limit) |
| D2 | HLMT+ | When external +direction limit signal (nLMTP) is on its active level |
| D3 | HLMT- | When external –direction limit signal (nLMTM) is on its active level |
| D4 | ALARM | When the alarm signal (nALARM) for servo motor is on its active level |
| D5 | EMG | When emergency stop signal (EMGN) becomes Low level. |
| D6 | MULT | This bit is only for the X axis of main chip at the multichip interpolation. When an error occurs in any axis of sub chip during multichip interpolation, it will become 1. |

In driving, when hardware / software limit is active, the decelerating stop or sudden stop will be executed.

Bit SLMT+ / – will not become 1 during the reverse direction driving.

4.13 Status Register 3: RR3

Each axis is with status register 3. The axis specified by NOP command or the condition before decides which axis' s register will be read.

This register is for reflecting the interrupt factor. When interrupt happens, the bit with the interrupt factor becomes 1. The user should set the interrupt factor through register WR1 to perform the interrupt.

To generate an interrupt, interrupt enable must be set for each factor in the WR1 register.

| | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|----|----|-------|-------|-------|-------------|------|------|-------------|----|
| RR3 | H | | | | | | | | L | | | | | | | |
| | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| | – | – | – | – | – | – | – | – | D-END | C-STA | C-END | P \geq C+ | P<C+ | P<C- | P \geq C- | – |

| | | |
|----|-------------|---|
| D1 | P \geq C- | Once the value of logical / real position counter is larger than that of COMP- register |
| D2 | P < C- | Once the value of logical / real position counter is smaller than that of COMP- register |
| D3 | P < C+ | Once the value of logical / real position counter is smaller than that of COMP+ register |
| D4 | P \geq C+ | Once the value of logical / real position counter is larger than that of COMP+ register |
| D5 | C-END | When the pulse output is finished in the constant speed drive during an acceleration / deceleration driving |
| D6 | C-STA | When the pulse output is started in the constant speed drive during an acceleration / deceleration driving |
| D7 | D-END | When the driving is finished |

When one of the interrupt factors occurs an interrupt, the bit of the register becomes 1, and the interrupt output signal (INTN) will become the Low level. The host CPU will read register RR3 of the interrupted axis, the bit of RR3 will be cleared to 0, and the interrupt signal will return to the non-active level. For a 8-bit data bus, all the bits are cleared when the RR3L register is read.

4.14 Input Register: RR4 / RR5

RR4 and RR5 are used for displaying the input signal status. The bit is 0 if the input is on the Low level; the bit is 1 if the input is on the Hi level.

These input signals can be used as general input signal when they are not used as function except for nLMTP/M signal.

| | | | | | | | | | | | | | | | | |
|-----|-------|-------|-------|------------------|-------|-------|-------|-------|-------|-------|-------|-----------------|-----|-------|-------|-------|
| RR4 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | X-LM- | X-LM+ | X-IN5 | X-IN4 | X-IN3 | X-IN2 | X-IN1 | X-IN0 | X-ALM | X-INP | X-EX- | X-EX+ | ENG | X-ST2 | X-ST1 | X-ST0 |

| | | | | | | | | | | | | | | | | |
|-----|-------|-------|-------|------------------|-------|-------|-------|-------|-------|-------|-------|-----------------|----|-------|-------|-------|
| RR5 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | Y-LM- | Y-LM+ | Y-IN5 | Y-IN4 | Y-IN3 | Y-IN2 | Y-IN1 | Y-IN0 | Y-ALM | Y-INP | Y-EX- | Y-EX+ | — | Y-ST2 | Y-ST1 | Y-ST0 |

| Bit Name | Input Signal | Bit Name | Input Signal |
|----------|--------------|----------|--------------|
| n-ST0 | n-STOP0 | n-IN0 | nIN0 |
| n-ST1 | n-STOP1 | n-IN1 | nIN1 |
| n-ST2 | n-STOP2 | n-IN2 | nIN2 |
| EMG | EMGN | n-IN3 | nIN3 |
| n-EX+ | nEXPP | n-IN4 | nIN4 |
| n-EX- | nEXPM | n-IN5 | nIN5 |
| n-INP | nINPOS | n-LM+ | nLMTP |
| n-ALM | nALARM | n-LM- | nLMTM |

4.15 Data-Read Register: RR6 / RR7

According to the data-read command, the data of internal registers will be set into registers RR6 and RR7. The low word 16 bits (D15 ~ D0) is set in RR6 register, and the high word 16 bits (D31 ~ D16) is set in RR7 register for data reading.

| | | | | | | | | | | | | | | | | |
|-----|------|------|------|------------------|------|------|-----|-----|-----|-----|-----|-----------------|-----|-----|-----|-----|
| RR6 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | RD15 | RD14 | RD13 | RD12 | RD11 | RD10 | RD9 | RD8 | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 |

| | | | | | | | | | | | | | | | | |
|-----|------|------|------|------------------|------|------|------|------|------|------|------|-----------------|------|------|------|------|
| RR7 | D15 | D14 | D13 | D12 ^H | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 ^L | D3 | D2 | D1 | D0 |
| | RD31 | RD30 | RD29 | RD28 | RD27 | RD26 | RD25 | RD24 | RD23 | RD22 | RD21 | RD20 | RD19 | RD18 | RD17 | RD16 |

The data is binary formatted; 2' s complement is for negatives.

5. Command Lists

■ Write Commands

| Code | Command | Symbol | Data Range | Data Length |
|------|--|--------|--|-------------|
| 00h | Range setting | R | R 8,000,000(multiple=1) ~ 16,000(multiple=500) | 4 bytes |
| 01 | Jerk setting (Acceleration increasing rate) | K | 1 ~ 65,535 | 2 |
| 02 | Acceleration setting | A | 1 ~ 8,000 | 2 |
| 03 | Deceleration setting | D | 1 ~ 8,000 | 2 |
| 04 | Initial speed setting | SV | 1 ~ 8,000 | 2 |
| 05 | Drive speed setting | V | 1 ~ 8,000 | 2 |
| 06 | Output pulse numbers / finish point | P | Output pulse numbers: 0~268,435,455 /finish point: -8,388,608 ~ +8,388,607 | 4 |
| 07 | Manual deceleration point setting | DP | 0 ~ 268,435,455 | 4 |
| 08 | Circular center point setting | C | -8,388,608 ~ +8,388,607 | 4 |
| 09 | Logical position counter setting | LP | -2,147,483,648 ~ +2,147,483,647 | 4 |
| 0A | Real position counter setting | EP | -2,147,483,648 ~ +2,147,483,647 | 4 |
| 0B | COMP + register setting | CP | -1,073,741,824 ~ +1,073,741,824 | 4 |
| 0C | COMP - register setting | CM | -1,073,741,824 ~ +1,073,741,824 | 4 |
| 0D | Acceleration counter offsetting | AO | -32,768 ~ +32,767 | 2 |
| 0F | NOP (For axis switching) | | | |

[Note] When those parameters are written, the total data length should be completely filled.

[Formula Calculation for Parameters]

$$\text{Multiple} = \frac{8,000,000}{R}$$

$$\text{Deceleration Increasing Rate (PPS/SEC}^2\text{)} = \frac{62.5 \times 10^6}{L} \times \frac{8,000,000}{\text{Multiple}}$$

$$\text{Jerk (PPS/SEC}^2\text{)} = \frac{62.5 \times 10^6}{K} \times \frac{8,000,000}{\text{Multiple}}$$

$$\text{Deceleration (PPS/SEC)} = D \times 125 \times \frac{8,000,000}{\text{Multiple}}$$

$$\text{Acceleration (PPS/SEC)} = A \times 125 \times \frac{8,000,000}{\text{Multiple}}$$

$$\text{Initial Speed (PPS)} = SV \times \frac{8,000,000}{\text{Multiple}}$$

$$\text{Drive Speed (PPS)} = V \times \frac{8,000,000}{\text{Multiple}}$$

■ Data Reading Commands

| Code | Command | Symbol | Data Range | Data Length |
|------|---|--------|---------------------------------|-------------|
| 10h | Logical position counter reading | LP | -2,147,483,648 ~ +2,147,483,647 | 4 bytes |
| 11 | Real position counter reading | EP | -2,147,483,648 ~ +2,147,483,647 | 4 |
| 12 | Current drive speed reading | CV | 1 ~ 8,000 | 2 |
| 13 | Acceleration / deceleration reading | CA | 1 ~ 8,000 | 2 |
| 14 | Maximum finish point reading for multichip linear interpolation | MX | 0 ~ 8,388,607 | 4 |

■ Driving Commands

| Code | Command |
|------|---|
| 20h | + direction fixed driving |
| 21 | - direction fixed driving |
| 22 | + direction continuous driving |
| 23 | - direction continuous driving |
| 24 | Drive start holding |
| 25 | Drive start holding release / stop status clear |
| 26 | Decelerating stop |
| 27 | Sudden stop |

■ Interpolation Commands

| Code | Command |
|------|---|
| 30h | 2-axis leaner interpolation |
| 31 | |
| 32 | CW circular interpolation |
| 33 | CCW circular interpolation |
| 34 | 2-axis bit pattern interpolation |
| 35 | |
| 36 | BP register writing enabled* |
| 37 | BP register writing disabled |
| 38 | BP data stack |
| 39 | BP data clear |
| 3A | 1-step interpolation |
| 3B | Deceleration valid |
| 3C | Deceleration invalid |
| 3D | Interpolation interrupt clear |
| 3E | Maximum finish point clear for multichip linear interpolation |

*BP = bit pattern

[Note] Please do not write the codes not mentioned above. The unknown situation could happen due to IC internal circuit test.

6. Commands for Data Writing

Data writing is used for setting driving parameters such as acceleration, drive speed, output pulse numbers...

It is possible to write the same data for more than one axis simultaneously if more those axes are assigned.

If the data length is two bytes, WR6 register can be used. If the data is 4 bytes, WR6/7 register can be used. Then, the axis assignment and command code will be written into register WR0 for execution.

Writing data for registers WR6 and WR7 is binary and 2' s complement for negatives. Each data should be set within the permitted data range. If the setting data out of range, the driving can not be done.

[Note]

- a. It requires 250 nSEC (maximum) to access the command code when CLK=16MHz. Please don' t write the next command or data into WR0 when the present command is written.
- b. Except acceleration offset (AO), the other parameters are unknown while resetting. So, please per-set proper values for those driving related parameters before the driving starts.

6.1 Range Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|---------------|--------|---|-------------|
| 00h | Range setting | R | 8,000,000 (multiple:1) ~ 16,000 (multiple:500) | 4 bytes |

“R” is the parameter determining the drive speed, acceleration / deceleration and jerk. The multiple can be calculated as follows where the range setting value is R.

$$\text{Multiple} = \frac{8,000,000}{R}$$

For the parameter setting range of drive speed, acceleration / deceleration is 1~8000, if the higher value is needed, the user should have a larger multiple.

In case of increasing the multiple, although the high speed driving is possible, the speed resolution will be decreased. So, the user can set the multiple as small as possible if the setting speed has covered the desired speed. For example, the maximum value of parameter for setting the drive speed (V) is 8000, and the drive speed is set 40KPPS. The user can set V=8000 and R=1,600,000. Because 40K is 5 times of 8000, we set the R=8,000,000/5=1,600,000.

The Range (R) cannot be changed during the driving. The speed will be changed discontinuously.

6.2 Jerk Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|--------------|--------|------------|-------------|
| 01h | Jerk setting | K | 1 ~ 65,535 | 2 bytes |

A jerk setting value is a parameter that determines the acceleration increase/decrease rate per unit in S-curve acceleration/deceleration.

“K” is the parameter determining the jerk. The jerk calculation is shown in the following formula:

$$\text{Jerk (PPS/SEC}^2\text{)} = \frac{62.5 \times 10^6}{K} \times \frac{8,000,000}{\text{Multiple}}$$

Because the setting range of jerk is 1 ~ 65,535, the jerk range is shown as follows:

$$\begin{aligned} \text{When Multiple} = 1, & \quad 954 \text{ PPS/SEC}^2 \sim 62.5 \times 10^6 \text{ PPS/SEC}^2 \\ \text{When Multiple} = 500, & \quad 477 \times 10^3 \text{ PPS/SEC}^2 \sim 31.25 \times 10^9 \text{ PPS/SEC}^2 \end{aligned}$$

[Note] This book uses the word “jerk” to express increase / decrease of acceleration / deceleration and increase / decrease rate per unit.

6.3 Acceleration Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|----------------------|--------|------------|-------------|
| 02h | Acceleration setting | A | 1 ~ 8,000 | 2 bytes |

In linear acceleration / deceleration driving, “A” is the parameter determining the acceleration at acceleration and deceleration at deceleration.

Acceleration / deceleration increases on a straight line from 0 to the specified value when S-curve acceleration / deceleration driving. Please refer to Fig. 2.13

The acceleration calculation is shown in the following formula:

$$\text{Acceleration (PPS/SEC)} = A \times 125 \times \frac{8,000,000}{\text{Multiple}}$$

For the range of A is from 1 ~ 8,000, the actual acceleration range is shown as follows:

$$\begin{aligned} \text{When Multiple} = 1, & \quad 125 \text{ PPS/SEC} \sim 1 \times 10^6 \text{ PPS/SEC} \\ \text{When Multiple} = 500, & \quad 62.5 \times 10^3 \text{ PPS/SEC} \sim 500 \times 10^6 \text{ PPS/SEC} \end{aligned}$$

6.4 Deceleration Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|----------------------|--------|------------|-------------|
| 03h | Deceleration setting | D | 1 ~ 8,000 | 2 bytes |

In linear acceleration / deceleration driving, “D” is the parameter determining the deceleration at deceleration in the acceleration / deceleration individual settings mode (WR3register D1=1).

Deceleration increases on a straight line from 0 to the specified value when S-curve acceleration / deceleration driving in this mode.

The deceleration calculation is shown in the following formula:

$$\text{Deceleration (PPS/SEC)} = D \times 125 \times \frac{8,000,000}{\text{Multiple}}$$

6.5 Initial Speed Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|-----------------------|--------|------------|-------------|
| 04h | Initial speed setting | SV | 1 ~ 8,000 | 2 bytes |

“SV” is the parameter determining the speed of initial speed. The initial speed calculation is shown in the following formula:

$$\text{Initial Speed (PPS)} = SV \times \frac{8,000,000}{\text{Multiple}}$$

In trapezoidal(Linear) acceleration/deceleration driving, for stepper motors, the user should set the initial speed smaller than the self-starting frequency of stepper motors. Also for a servo motor, if the value that is set is too low, creep or premature termination may occur. In this case, it is appropriate to set the value larger than $\sqrt{\text{acceleration}}$ (acceleration). For instance, when acceleration=125000 PPS/SEC, the value should be larger than $\sqrt{125000}=354$ PPS.

In fixed pulse S curve acceleration / deceleration driving, if the value that is set is too low such as SV is set less than 100, creep or premature termination may occur as well. Set the initial speed value (SV) as more than 100.

6.6 Drive Speed Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|---------------------|--------|------------|-------------|
| 05h | Drive speed setting | V | 1 ~ 8,000 | 2 bytes |

“V” is the parameter determining the speed of constant speed period in trapezoidal driving. In constant speed driving, the drive speed is the initial speed. The drive speed calculation is shown in the following formula:

$$\text{Drive Speed (PPS)} = V \times \frac{8,000,000}{\text{Multiple}}$$

If the setting drive speed is lower than the initial speed, the acceleration / deceleration will not be performed, and the driving is constant speed. During the encoder Z-phase searching (at a low-speed driving), if the user want to perform the sudden stop once the Z-phase is detected, the drive speed should be set lower than the initial speed.

Drive speed can be altered during the driving. When the drive speed of next constant speed period is set, the acceleration / deceleration will be performed to reach the new setting drive speed, then a constant speed driving starts.

[Note]

- a. In fixed S-curve acceleration / deceleration driving, there is no way to change the drive speed during the driving. In continuous S-curve acceleration / deceleration driving, the S-curve profile cannot be exactly tracked if the speed alterations during the acceleration / deceleration. it is better to change the drive speed in the constant speed period.
- b. In fixed trapezoidal driving, the frequent changes of drive speed may occur residual pulses at the end of deceleration.

6.7 Output Pulse Number / Interpolation Finish Point Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|--|--------|--|-------------|
| 06h | Output pulse number / interpolation finish point setting | P | Output pulse numbers: 0 ~ 268,435,455 Finish point: -8,388,608 ~ +8,388,607 | 4 bytes |

The number of output pulses indicates the total number of pulses that are output in fixed driving.

Set the finish point of each axis for linear and circular Interpolations, set the finish point by relative numbers with a signed 24-bit.

The number of output pulses is an unsigned but interpolation end-point is a signed, both the number of output pulses and the interpolation end-point should be set in 4 bytes data length.

The output pulse numbers can be changed during the driving.

6.8 Manual Decelerating Point Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|-----------------------------------|--------|-----------------|-------------|
| 07h | Manual decelerating point setting | DP | 0 ~ 268,435,455 | 4 bytes |

“DP” is the parameter setting the manual deceleration point in fixed acceleration / deceleration driving when the manual deceleration mode is engaged.

In manual deceleration mode, the user can set the bit D0 of WR3 register to 1. The decelerating point can be set:

$$\text{Manual Decelerating Point} = \text{Output Pulse Number} - \text{Pulse Number for Deceleration}$$

6.9 Circular Center Point Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|-------------------------------|--------|-------------------------|-------------|
| 08h | Circular center point setting | C | -8,388,608 ~ +8,388,607 | 4 bytes |

“C” is the parameter setting the center point in circular interpolation. The coordinates of center point should be set the relative number related to the current position.

6.10 Logical Position Counter Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|----------------------------------|--------|---------------------------------|-------------|
| 09h | Logical position counter setting | LP | -2,147,483,648 ~ +2,147,483,647 | 4 bytes |

“LP” is the parameter setting the value of logic position counter.

Logical position counter counts Up / Down according to the +/- direction pulse output.

The data writing and reading of logical position counter is possible anytime.

6.11 Real position Counter Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|-------------------------------|--------|---------------------------------|-------------|
| 0Ah | Real position counter setting | EP | -2,147,483,648 ~ +2,147,483,647 | 4 bytes |

“EP” is the parameter setting the value of real position counter.

Real position counter counts Up / Down according to encoder pulse input.

The data writing and reading of real position counter is possible anytime.

6.12 COMP+ Register Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|------------------------|--------|---------------------------------|-------------|
| 0Bh | COMP+ register setting | CP | -2,147,483,648 ~ +2,147,483,647 | 4 bytes |

“CP” is the parameter setting the value of COM+ register.

COMP+ register is used to compare with logical / real position counter, and the comparison result will be output to bit D0 of register RR1 or nOUT2/CMPP signal. Also, it can be used as the + direction software limit.

The value of COMP+ register can be written anytime.

6.13 COMP- Register Setting

| Code | Command | Symbol | Data Range | Data Length |
|------|------------------------|--------|---------------------------------|-------------|
| 0Ch | COMP- register setting | CM | -2,147,483,648 ~ +2,147,483,647 | 4 bytes |

“CM” is the parameter setting the value of COMP – register.

COMP- register is used to compare with logical / real position counter, and the comparison result will be output to bit D0 of RR1 register or nOUT3/CMPP signal. Also, it can be used as the - direction software limit.

The value of COMP- register can be written anytime.

6.14 Acceleration Counter Offsetting

| Code | Command | Symbol | Data Range | Data Length |
|------|---------------------------------|--------|-------------------|-------------|
| 0Dh | Acceleration Counter Offsetting | AO | -32,768 ~ +32,767 | 2 bytes |

“AO” is the parameter executing acceleration counter offset.

The offset value of acceleration counter will be set 8 while resetting.

6.15 NOP (for Axis Switching)

| Code | Command | Symbol | Data Range | Data Length |
|------|--------------------------|--------|------------|-------------|
| 0Fh | NOP (for axis switching) | | | |

No execution is performed.

Use this command for switching the axis for selecting the registers from WR1 ~ WR3 registers and RR1 ~ RR3 registers.

7. Commands for Reading Data

Data reading commands are used to read the register contents of each axis.

After a data reading command is written into register WR0, this data will be set in registers RR6 and RR7.

The host CPU can reach the data through reading registers RR6 and RR7. Reading data for registers WR6 and WR7 is binary and 2' s complement for negatives.

[Note]

a. It requires 250 nSEC (maximum) to access the command code of data reading where CLK = 16MHz. After the command is written and passed that time, read registers RR6 and 7.

b. The axis assignment is for one axis. If both X and Y axes are assigned, the data reading priority is X.

7.1 Logical Position Counter Reading

| Code | Command | Symbol | Symbol | Data length |
|------|----------------------------------|--------|---------------------------------|-------------|
| 10h | Logical position counter reading | LP | -2,147,483,648 ~ +2,147,483,647 | 4 bytes |

The current value of logical position counter will be set in read registers RR6 and RR7.

7.2 Real position Counter Reading

| Code | Command | Symbol | Data range | Data length |
|------|-------------------------------|--------|---------------------------------|-------------|
| 11h | Real position counter reading | EP | -2,147,483,648 ~ +2,147,483,647 | 4 bytes |

The current value of real position counter will be set in read registers RR6 and RR7.

7.3 Current Drive Speed Reading

| Code | Command | Symbol | Data range | Data length |
|------|-----------------------------|--------|------------|-------------|
| 12h | Current drive speed reading | CV | 1 ~ 8,000 | 2 bytes |

The value of current drive speed will be set in read registers RR6 and RR7.

When the driving stops, the value becomes 0. The data unit is as same as the setting value of drive speed (V).

7.4 Current Acceleration / Deceleration Reading

| Code | Command | Symbol | Data range | Data length |
|------|---|--------|------------|-------------|
| 13h | Current acceleration / deceleration reading | CA | 1 ~ 8,000 | 2 bytes |

The value of current acceleration / deceleration will be set in read registers RR6 and RR7.

When the driving stops, the read data is random number. The data unit is as same as the setting value of acceleration (A).

7.5 Maximum finish point Reading for multichip linear interpolation

| Code | Command | Symbol | Data range | Data length |
|------|---|--------|------------|-------------|
| 14h | Maximum finish point reading for multichip linear interpolation | MX | 1 ~ 8,000 | 4 bytes |

The maximum finish point value is read in multichip linear interpolation. X axis should be selected.

The maximum finish point value is cleared to 0 when resetting or immediately after interpolation drive command starts.

8. Driving Commands

Driving commands include the commands for each axis' s drive pulse output and other related commands. After the command code and axis assignment are written in command register WR0, the command will be executed immediately. It is possible to assign more than one axis with same command at the same time.

In driving, bit n-DRV of each axis' s main status register RR0 becomes 1. When the driving is finished, the bit n-DRV will return to 0.

If nINPOS input signal for servo drivers is enabled, bit n-DRV of main status register RR0 will not return to 0 until nINPOS signal is on its active level.

[Note] It requires 250 nSEC (maximum) to access the command code when CLK=16MHz. Please write the next command after this period of time.

8.1 +Direction Fixed Driving

| Code | Command |
|------|--------------------------|
| 20h | +Direction Fixed Driving |

The setting pulse numbers will be output through the output signal nPP.

In driving, logical position counter will count-up 1 when one pulse is output.

Before writing the driving command, the user should set the parameters for the outputting speed curve and the correct output pulse numbers (see the table below).

| | Range (R) | Jerk (K) | Acceleration (A) | Deceleration (D) | Initial Speed (SV) | Drive Speed (V) | Output pulses (P) |
|--|--------------|-------------|------------------|---------------------|-----------------------|--------------------|----------------------|
| Constant speed driving | ○ | | | | ○ | ○ | ○ |
| Linear acceleration / deceleration driving | ○ | | ○ | △ | ○ | ○ | ○ |
| Non-symmetrical linear acceleration / deceleration driving | ○ | | ○ | ○ | ○ | ○ | ○ |
| S curve acceleration / deceleration driving | ○ | ○ | ○ | △ | ○ | ○ | ○ |

△ will be set when needed

8.2 –Direction Fixed Driving

| Code | Command |
|------|--------------------------|
| 21h | –Direction Fixed Driving |

The setting pulse numbers will be output through the output signal nPM.

In driving, logical position counter will count-down 1 when one pulse is output.

Before writing the driving command, the user should set the parameters for the outputting speed curve and the correct output pulse numbers.

8.3 +Direction Continuous Driving

| Code | Command |
|------|-------------------------------|
| 22h | +Direction Continuous Driving |

Before the stop command or external signal is active, the pulse numbers will be continuously output through the output signal nPP.

In driving, logical position counter will count-up 1 when one pulse is output.

Before writing the driving command, the user should set the parameters for the outputting speed curve and the correct output pulse numbers.

8.4 –Direction Continuous Driving

| Code | Command |
|------|-------------------------------|
| 23h | –Direction Continuous Driving |

Before the stop command or external signal is active, the pulse numbers will be continuously output through the output signal nPM.

In driving, logical position counter will count-down 1 when one pulse is output.

Before writing the driving command, the user should set the parameters for the outputting speed curve and the correct output pulse numbers.

8.5 Drive Status Holding

| Code | Command |
|------|------------------------------|
| 24h | Holding for driving starting |

This command is to hold-on the start of driving.

When this command is used for starting multi-axis driving simultaneously, the user may write other commands after the drive status holding command is registered. The drive start holding release command (25h) can be written to start the driving.

In driving, even this command is written, the driving will not be stopped. The next command will be held.

8.6 Drive Status Holding Release / Finishing Status Clear

| Code | Command |
|------|--|
| 25h | Drive status holding release / finishing status clearing |

This command is to release the drive status holding (24h), and start the driving.

Also, this command can clear the finishing status bits D15 ~ 8 of register RR1.

8.7 Decelerating Stop

| Code | Command |
|------|------------------------------|
| 26h | Decelerating stop in driving |

This command performs the decelerating stop when the drive pulses are outputting.

If the drive speed is lower than the initial speed, the driving will be suddenly stopped when this command is engaged. In interpolation driving, for X axis, the decelerating stop and sudden stop commands can be written to stop the driving. Once the driving stops, this command will not work.

8.8 Sudden Stop

| Code | Command |
|------|------------------------|
| 27h | Sudden stop in driving |

This command performs the sudden stop when the drive pulses are output. Also, the sudden stop can be performed in acceleration / deceleration driving.

Once the driving stops, this command will not work.

9. Interpolation Commands

Interpolation commands consist of the commands for 2 axes linear interpolation, CW / CCW circular interpolation, 2 axes bit pattern interpolation and other related commands. There is no need to make the axis assignment in setting bits D9, 8 of command register WR0. Please set 0 in those bits.

Speed parameter setting for X axis is necessary before interpolation driving starts in any interpolation.

In interpolation driving, bit D8 (I-DRV) of main status register RR0 becomes 1, and will return to 0 when the driving is finished. In interpolation, the n-DRV bit of interpolating axis becomes 1.

[Note] It requires 250 nSEC (maximum) to access the command code when CLK=16MHz. Please write the next command within this period of time.

9.1 2-Axis Linear Interpolation

| Code | Command |
|------|-----------------------------|
| 30h | 2-axis linear interpolation |

This command performs interpolation between X and Y axes from present point to finish point.

Before driving, the finish point of X and Y axes should be set by incremental value respectively.

9.2 CW Circular Interpolation

| Code | Command |
|------|---------------------------|
| 32h | CW circular interpolation |

This command performs 2-axis clockwise circular interpolation, based on center point, from present point to finish point.

Before driving, the finish point of the X and Y axes should be set by incremental value respectively.

A full circle will come out if the finish position is set (0, 0).

9.3 CCW Circular Interpolation

| Code | Command |
|------|------------------------------|
| 33h | CCW circulator interpolation |

This command performs 2-axis counterclockwise circular interpolation, based on center point, from present point to finish point.

Before driving, the finish point of X and Y axes should be set by incremental value respectively.

A full circle will come out If the finish position is set (0, 0).

9.4 2-Axis Bit Pattern Interpolation

| Code | Command |
|------|----------------------------------|
| 34h | 2-axis bit pattern interpolation |

This command performs 2-axis bit pattern interpolation.

Before driving, the +/- direction bit data of X and Y axes should be set, and the setting bit data of each axis (each direction) is at most $16 \times 3 = 48$ -bit. Once the data is over than 48-bit, those remaining data can be filled during the driving.

9.5 BP Register Data Writing Enabling

| Code | Command |
|------|-----------------------------------|
| 36h | BP register data writing enabling |

This command enables the bit pattern data writing registers BP1P/M and BP2P/M.

After this command is issued, the data writing to register nWR2~nWR5 becomes disabled.

The data written to the bit pattern data writing registers is disabled while resetting.

9.6 BP Register Data Writing Disabling

| Code | Command |
|------|------------------------------------|
| 37h | BP register data writing disabling |

This command disables the bit pattern data writing registers BP1P/M and BP2P/M.

After this command is issued, the data writing to register nWR2~nWR5 becomes enabled.

9.7 BP Data Stack

| Code | Command |
|------|------------------|
| 38h | BP data stacking |

This command stacks the data of bit pattern data writing registers BP1P/M and BP2P/M.

After this command is issued, stack counter (SC) will plus 1. When stack counter (SC) is 3, this command cannot be issued again.

9.8 BP Data Clear

| Code | Command |
|------|------------------|
| 39h | BP data clearing |

This command clears all the bit pattern data, and sets the stack counter (SC) to 0.

9.9 Single Step Interpolation

| Code | Command |
|------|---------------------------|
| 3Ah | Single step interpolation |

This command performs 1-pulse (each step) output in interpolation driving.

When D13 bit of register WR5 is set 1, the single step interpolation can be performed. After this command is issued, single step interpolation starts.

9.10 Deceleration Enabling

| Code | Command |
|------|-----------------------|
| 3Bh | Deceleration enabling |

This command enables the automatic and manual decelerations.

In case of the individual interpolation, the user can issue this command before the driving. However, in continuous interpolation, the user should disable the deceleration than start the driving. This command should be put in the final node, and written before the interpolation command of the final node is written. If each axis has to decelerate individually, execute this command before driving. But for continuous interpolation, disable the deceleration first and enable it until the last node.

The deceleration is disabled while resetting. When the deceleration enabling command is issued, the enabling status is kept until the deceleration disabling command (3C) is written, or the reset happens.

Deceleration enabling / disabling is active in interpolation; automatic and manual decelerations are always active when individual axis is in driving.

9.11 Deceleration Disabling

| Code | Command |
|------|------------------------|
| 3Ch | Deceleration disabling |

This command disables the automatic or manual deceleration in interpolation.

9.12 Interpolation Interrupt Clear

| Code | Command |
|------|-------------------------------|
| 3Dh | Interpolation interrupt clear |

This command clears the interrupt in bit pattern or continuous interpolation.

After the bit D15 of WR5 is set to 1 in bit pattern interpolation, the stack counter (SC) is changed from 2 to 1, and the interrupt will be generated. In continuous interpolation, when the bit D14 of WR5 is set to 1, the interrupt will be generated when it is ready to write the interpolation data for next node.

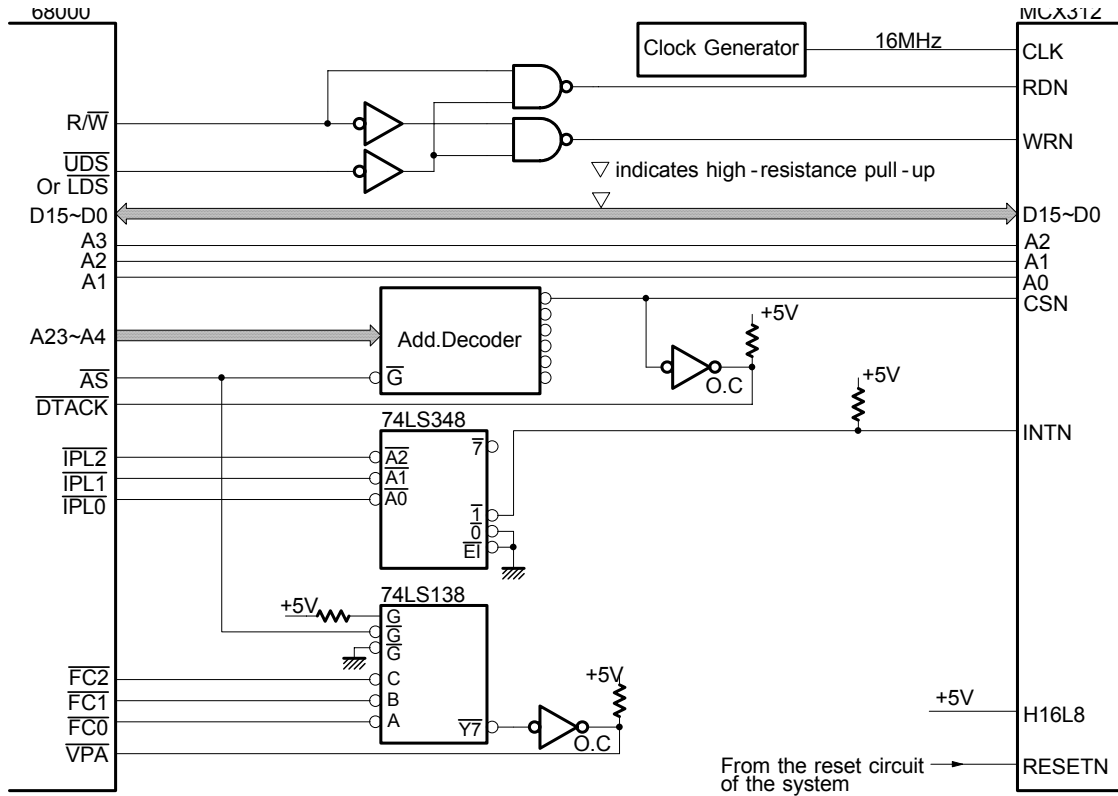
9.13 Maximum finish point Clear for multichip linear interpolation

| Code | Command |
|------|---|
| 3Eh | Maximum finish point clear for multichip linear interpolation |

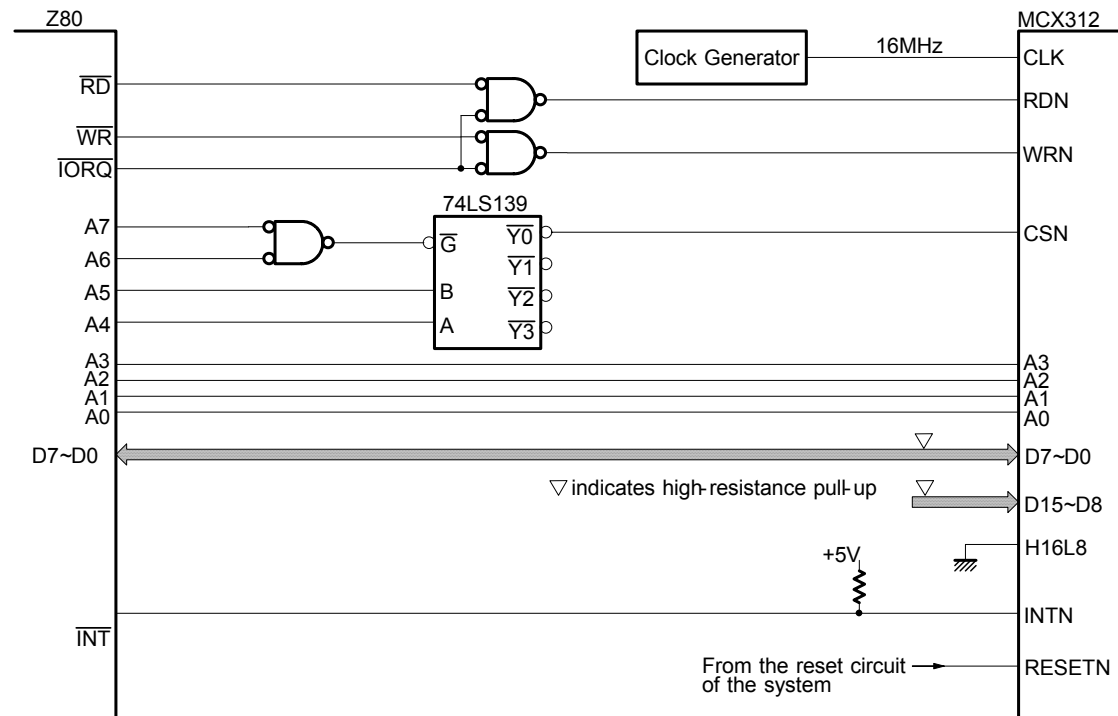
This command clears the maximum value of interpolation finish point in multichip linear interpolation.

10. Connection Examples

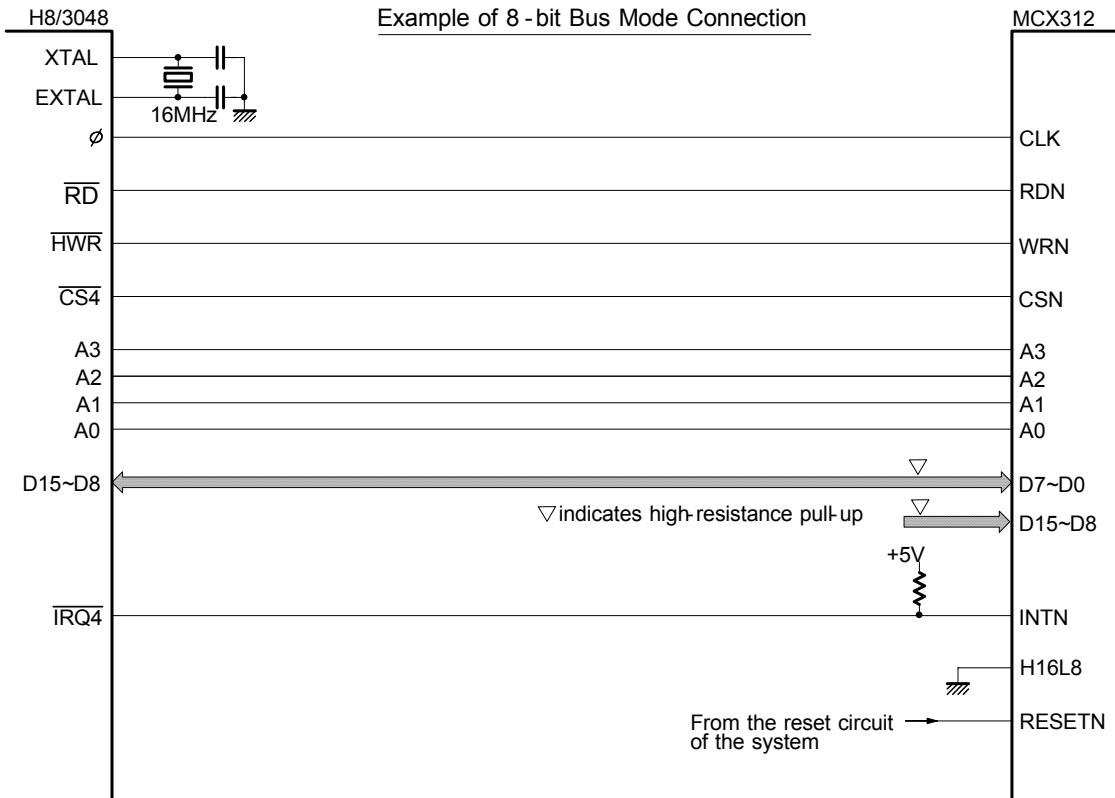
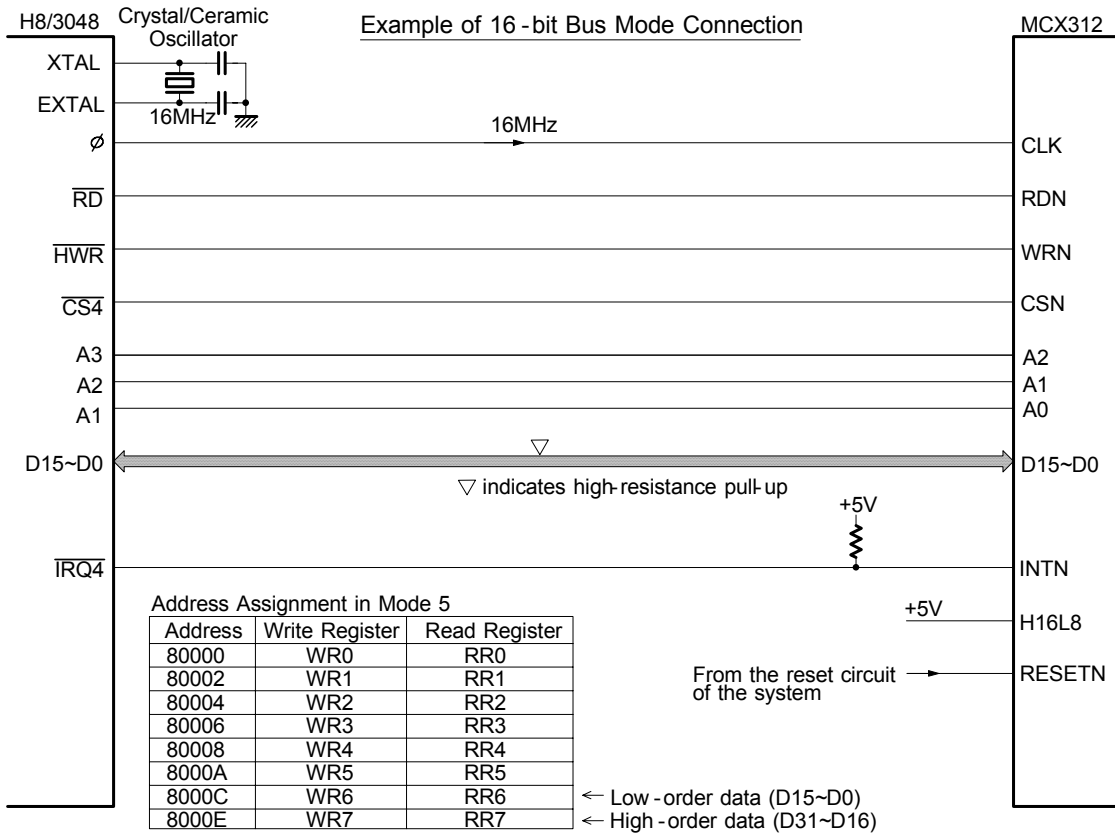
10.1 Connection Example for 68000 CPU



10.2 Connection Example for Z80 CPU

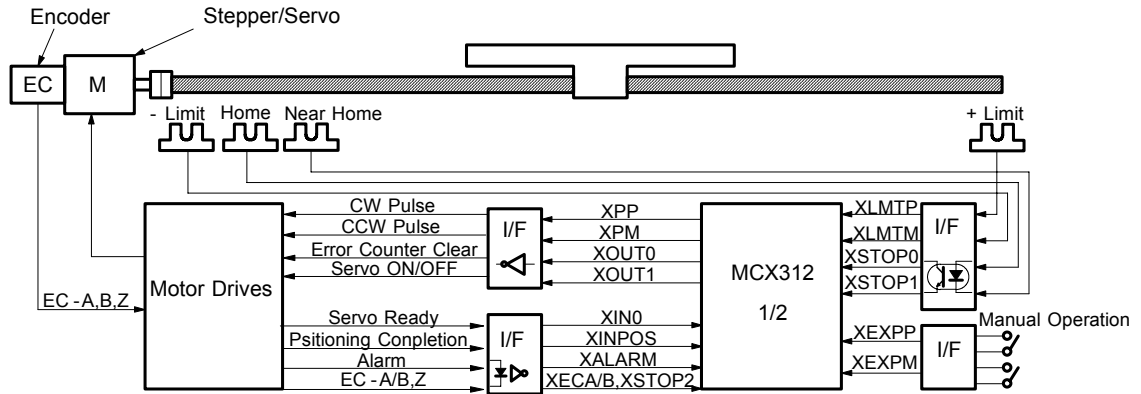


10.3 Example of Connection with H8 CPU



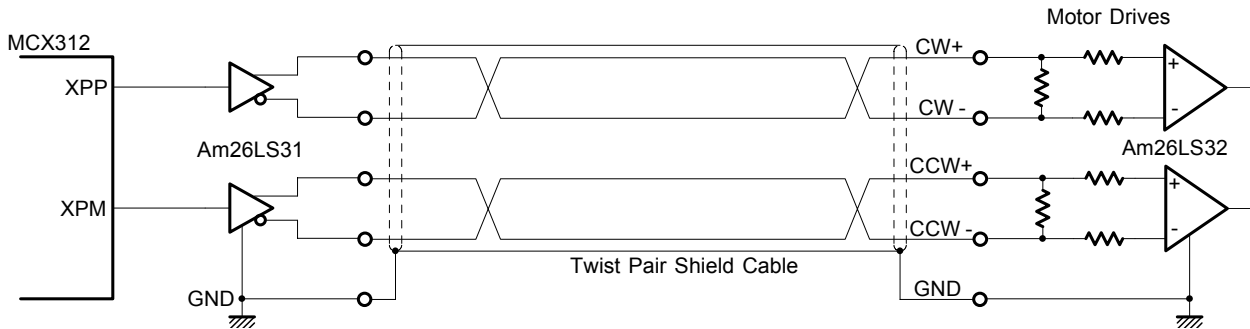
10.4 Connection Example

The figure shown below illustrates the example of X axis driving system. Y axis can be assigned in the same way.

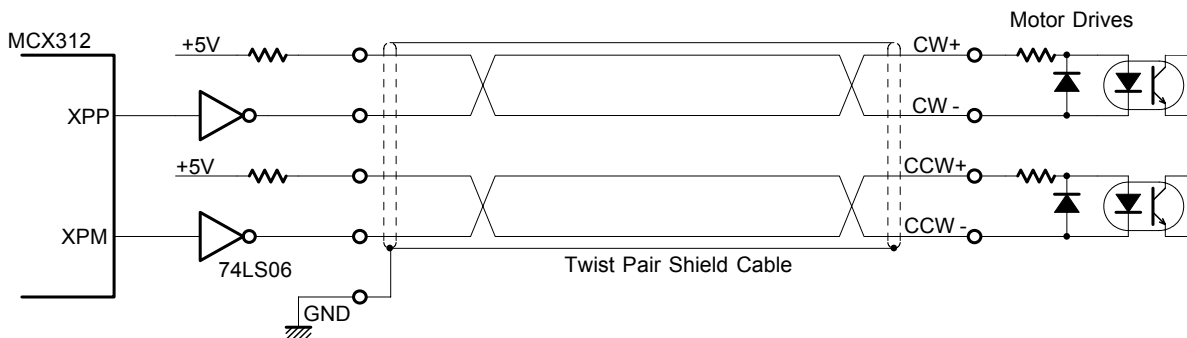


10.5 Pulse Output Interface

■ Output to Motor Drivers in Differential Circuit



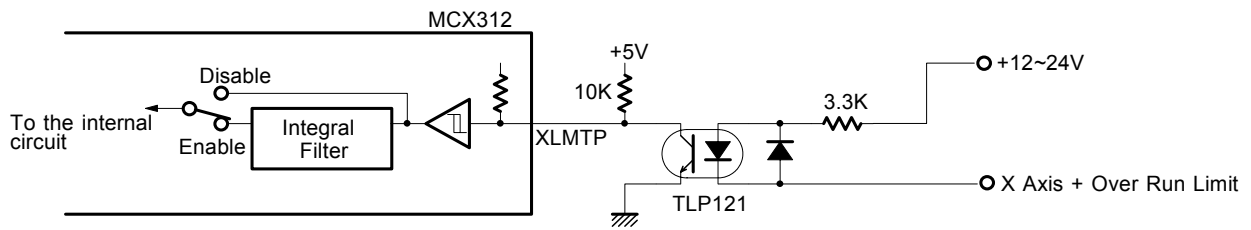
■ Open Collector TTL Output



For drive pulse output signals, we recommend the user to use twist pair shield cable due to the concern of EMC.

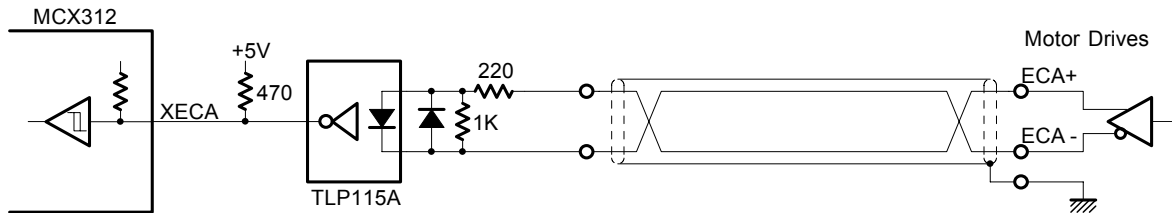
10.6 Connection Example for Input Signals

Limit signals often pick up some noise since complicated cabling is normally involved. A photo coupler alone may not be able to absorb this noise. Enable the filter function in the IC and set an appropriate time constant (FL=2,3).



10.7 Connection Example for Encoder

The following diagram is the example for the encoder signal which is differential line-drive output, Then, this signal can be received through the high speed photo coupler IC which can direct it to MCX312.



11. Example Program

The example of C program for MCX312 is shown in this section. This is a 16-bit bus configuration program.

```
#include <stdio.h>
#include <conio.h>

// ---- MCX312 register address definition ----

#define adr 0x0280 // Basic address

#define wr0 0x0 //Command register
#define wr1 0x2 //Mode register 1
#define wr2 0x4 //Mode register 2
#define wr3 0x6 //Mode register 3
#define wr4 0x8 //Output register
#define wr5 0xa //Interpolation mode register
#define wr6 0xc //Low word bits data writing register
#define wr7 0xe //High word bits data writing register

#define rr0 0x0 //Main status register
#define rr1 0x2 //Status register 1
#define rr2 0x4 //Status register 2
#define rr3 0x6 //Status register 3
#define rr4 0x8 //Input register 1
#define rr5 0xa //Input register 2
#define rr6 0xc //Low word bits data reading register
#define rr7 0xe //High word bits data reading register

#define bp1p 0x4 //BP + direction data register for the X axis control
#define bp1m 0x6 //BP - direction data register for the X axis control
#define bp2p 0x8 //BP + direction data register for the Y axis control
#define bp2m 0xa //BP - direction data register for the Y axis control

// wreg 1 (axis assignment, data) ----Write register 1 setting
void wreg1(int axis,int wdata)
{
    outpw(adr+wr0, (axis << 8) + 0xf); //axis assignment
    outpw(adr+wr1, wdata);
}

// wreg 2 (axis assignment, data) ----Write register 2 setting
void wreg2(int axis,int wdata)
{
    outpw(adr+wr0, (axis << 8) + 0xf); //axis assignment
    outpw(adr+wr2, wdata);
}

// wreg 3 (axis assignment, data) ----Write register 3 setting
void wreg3(int axis,int wdata)
{
    outpw(adr+wr0, (axis << 8) + 0xf); //axis assignment
    outpw(adr+wr3, wdata);
}

// command (axis assignment, data) ----For writing commands
void command(int axis,int cmd)
{
    outpw(adr+wr0, (axis << 8) + cmd);
}

// range(axis assignment, data) ----For range (R) setting
void range(int axis,long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x00);
}

// acac(axis assignment, data) ----For Jerk (K) setting
void acac(int axis,int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x01);
}
```

```

// acc(axis assignment, data) -----For acceleration/deceleration (A) setting
void acc(int axis,int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x02);
}

// dec( axis assignment, data) -----For deceleration (D) setting
void dec(int axis,int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x03);
}

// startv(axis assignment, data) -----For initial speed (SV) setting
void startv(int axis,int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x04);
}

// speed(axis assignment, data) -----For drive speed (V) setting
void speed(int axis,int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x05);
}

// pulse( axis assignment, data) -----For output pulse output/finish point (P) setting
void pulse(int axis,long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x06);
}

// decp(axis assignment, data) -----For manual deceleration (DP) setting
void decp(int axis,long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x07);
}

// center(axis assignment, data) -----For circular center point (C) setting
void center(int axis,long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x08);
}

// lp(axis assignment, data) -----For logical position counter (LP ) setting
void lp(int axis,long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x09);
}

// ep(axis assignment, data) -----For real position counter (EP) setting
void ep(int axis,long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x0a);
}

// compp(axis assignment, data) -----For COMP+ (CP) setting
void compp(int axis,long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x0b);
}

```

```

// compm(axis assignment, data) -----For COMP – (CM) setting
void compm(int axis,long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x0c);
}

// accofst(axis assignment, data) ----For acceleration counter shift (AO) setting
void accofst(int axis,long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x0d);
}

// readlp(axis assignment) -----For logical position counter (LP) reading
long readlp(int axis)
{
    long a;long d6;long d7;
    outpw(adr+wr0, (axis << 8) + 0x10);
    d6 = inpw(adr+rr6);d7 = inpw(adr+rr7);
    a = d6 + (d7 << 16);
    return(a);
}

// readep(axis assignment) -----For real position counter (EP) reading
long readep(int axis)
{
    long a;long d6;long d7;
    outpw(adr+wr0, (axis << 8) + 0x11);
    d6 = inpw(adr+rr6);d7 = inpw(adr+rr7);
    a = d6 + (d7 << 16);
    return(a);
}

// wait(axis assignment) -----For waiting for drive stop
void wait(int axis)
{
    while(inpw(adr+rr0) & axis);
}

// next_wait() -----Next data setting of waiting for continuous interpolation
void next_wait(void)
{
    while((inpw(adr+rr0) & 0x0200) == 0x0);
}

// bp_wait() ----- Next data setting of waiting for BP interpolation
void bp_wait(void)
{
    while((inpw(adr+rr0) & 0x6000) == 0x6000);
}

// home search() ----- All axes home search
void homesrch(void)
{
    {
        wreg1(0x3,0x0008);
        speed(0x3,2000);
        if((inpw(adr+rr4) & 0x2) == 0x2)
        {
            command(0x1,0x23);
        }
        if((inpw(adr+rr5) & 0x2) == 0x2);
        {
            command(0x2,0x23);
        }
        wait(0x3);

        wreg1(0x3,0x000c);
        speed(0x3,50);
        command(0x3,0x22);
        wait(0x3);

        wreg1(0x3,0x0000);
        speed(0x3,4000);
        pulse(0x3,100);
        command(0x3,0x21);
        wait(0x3);

        lp(0x3,0);
    }
    // [Action] (1)~(3) for both X, Y
    // (1) When home signal(STOP1) is OFF,
    // continuous drive in – direction at 2000PPS.
    // Deceleration stop when STOP1 is ON

    // (2) Continuous drive in + direction at 500PPS
    // Sudden stop when STOP1 is OFF

    // (3) 100 pulse offset drive in –direction at 4000PPS

    // (4) X, Y axes LP = 0

```

```

wreg2(0x3,0x0003);          // X, Y axes software limit: ON
compp(0x1,100000);          // X: -1000 ~ +100000
compm(0x1,-1000);           //
compp(0x2,50000);          // Y: -500 ~ +50000
compm(0x2,-500);           //
}

// ***** Main routine *****

void main(void)
{
    int count;
    outpw(adr+wr0, 0x8000);    //Software reset
    for(count = 0; count < 2; ++count);

    command(0x3,0xf);        //----- X and Y axes mode setting -----

    outpw(adr+wr1, 0x0000);    //Mode register 1:          00000000 00000000
    outpw(adr+wr2, 0x0000);    //Mode register 2:          00000000 00000000
    outpw(adr+wr3, 0x0000);    //Mode register 3:          00000000 00000000
    outpw(adr+wr4, 0x0000);    //General output register: 00000000 00000000
    outpw(adr+wr5, 0x0000);    //Interpolation mode register: 00000000 00000000

    //----- X and Y axes operation parameter initial setting -----
    accfst(0x3,0);            // AO = 0
    range(0x3,800000);        // R = 800000 ( Multiple = 10)
    acac(0x3,1010);           // K = 1010 (Jerk = 619KPPS/SEC2)
    acc(0x3,100);             // A = 100 (Acceleration/deceleration = 125KPPS/SEC)
    dec(0x3,100);             // D = 100 (Deceleration = 125KPPS/SEC)
    startv(0x3,100);          // SV = 100 (Initial speed = 1000PPS)
    speed(0x3,4000);          // V = 4000 (Drive speed = 4000PPS)
    pulse(0x3,100000);        // P = 100000 (Output pulses = 100000)
    lp(0x3,0);                // LP = 0 (Logical position counter = 0)

    homesrch();               //----- Home search for all axes -----

    //----- X and Y axes linear acceleration driving ----
    acc(0x3,200);             // A = 200 (Acceleration/deceleration = 250KPPS/SEC)
    speed(0x3,4000);          // V = 4000 (Drive speed = 4000PPS)
    pulse(0x1,80000);         // xP = 80000
    pulse(0x2,40000);         // yP = 40000
    command(0x3,0x20);        // + fixed driving
    wait(0x3);                // Waits for termination of driving

    //----- X axis non-symmetrical linear acceleration driving ----
    wreg3(0x1,0x0002);        // Acceleration/deceleration individual mode
    acc(0x1,200);             // xA = 200 (Acceleration/deceleration = 250KPPS/SEC)
    dec(0x1,50);              // xD = 50 (Deceleration = 62.5KPPS/SEC)
    speed(0x1,4000);          // xV = 4000 (Drive speed = 4000PPS)
    pulse(0x1,80000);         // xP = 80000
    command(0x1,0x20);        // + fixed driving
    wait(0x1);                // Waits for termination of driving
    wreg3(0x1,0x0000);        // Release of acceleration/deceleration individual mode

    //----- X and Y axes S-curve acceleration/deceleration driving ----
    wreg3(0x3,0x0004);        // S curve mode
    acac(0x3,1010);           // K = 1010 (Jerk = 619KPPS/SEC2)
    acc(0x3,200);             // A = 200 (Acceleration/deceleration = 250KPPS/SEC)
    speed(0x3,4000);          // V = 4000 (Drive speed = 4000PPS)
    pulse(0x1,50000);         // xP = 50000
    pulse(0x2,25000);         // yP = 25000
    command(0x3,0x21);        // - fixed driving
    wait(0x1);                //
    wreg3(0x3,0x0000);        // Release of S curve acceleration/deceleration mode

    //----- 2 -axis linear interpolation driving ----
    outpw(adr+wr5,0x0100);    // Constant vector speed
    range(0x1,800000);         // xR = 800000 (Multiple = 10)
    range(0x2,1131371);       // yR = 800000 × 1.414
    startv(0x1,100);          // SV = 100 (Initial speed = 1000PPS)
    speed(0x1,100);           // V = 100 (Drive speed = 1000PPS constant speed)
    pulse(0x1,5000);          // xP = +5000 (Finish point X = +5000)
    pulse(0x2,-2000);         // yP = -2000 (Finish point Y = -2000)
    command(0x0, 0x30);       // 2 -axis linear interpolation
    wait(0x3);

    //----- CCW circular interpolation driving
    outpw(adr+wr5,0x0100);    // Constant vector speed
    range(0x1,800000);         // xR = 800000 (Multiple = 10)
    range(0x2,1131371);       // yR = 800000 × 1.414
    startv(0x1,100);          // SV = 100 (Initial speed = 1000PPS)
    speed(0x1,100);           // V = 100 (Drive speed = 1000PPS)
    center(0x1,-5000);         // xC = -5000 (Center X = -5000)
    center(0x2,0);            // yC = 0 (Center Y = 0)
    pulse(0x1,0);             // xP = 0 (Finish point X = 0) Perfect circle
    pulse(0x2,0);             // yP = 0 (Finish point Y = 0)
    command(0x0,0x33);        // CCW circular interpolation
    wait(0x3);
}

```

```

speed(0x1,1);
command(0,0x36);

outpw(adr+bp1p,0x0000);
outpw(adr+bp1m,0x2bff);
outpw(adr+bp2p,0xffd4);
outpw(adr+bp2m,0x0000);
command(0,0x38);

outpw(adr+bp1p,0xf6fe);
outpw(adr+bp1m,0x0000);
outpw(adr+bp2p,0x000f);
outpw(adr+bp2m,0x3fc0);
command(0,0x38);

outpw(adr+bp1p,0x1fdb);
outpw(adr+bp1m,0x0000);
outpw(adr+bp2p,0x00ff);
outpw(adr+bp2m,0xfc00);
command(0,0x38);

command(0,0x34);

bp_wait();

outpw(adr+bp1p,0x4000);
outpw(adr+bp1m,0x7ff5);
outpw(adr+bp2p,0x0000);
outpw(adr+bp2m,0x0aff);
command(0x0x38);

command(0,0x37);

wait(0x3);

//----- X and Y axes bit pattern interpolation (Figure 2.31 Example) ----
// V = 1 (Drive speed = 10PPS constant speed)
// Enables bit pattern data write

// 0~15 writes bit data

// Stack

// 16~31 writes bit data

// 32~47 writes bit data

// Starts 2-axes BP interpolation driving

// Waits for data write

// 48~63 writes bit data

// Disables bit pattern data write

// Waits for termination of driving

//----- X and Y axes continuous interpolation (Figure 2.35 Example) -----

speed(0x1,100);

pulse(0x1,4500);
pulse(0x2,0);
command(0,0x30);

next_wait();
center(0x1,0);
center(0x2,1500);
pulse(0x1,1500);
pulse(0x2,1500);
command(0,0x33);

next_wait();
pulse(0x1,0);
pulse(0x2,1500);
command(0,0x30);

next_wait();
center(0x1,-1500);
center(0x2,0);
pulse(0x1,-1500);
pulse(0x2,1500);
command(0,0x33);

next_wait();
pulse(0x1,-4500);
pulse(0x2,0);
command(0,0x30);

next_wait();
center(0x1,0);
center(0x2,-1500);
pulse(0x1,-1500);
pulse(0x2,-1500);
command(0,0x33);

next_wait();
pulse(0x1,0);
pulse(0x2,-1500);
command(0,0x30);

next_wait();
center(0x1,1500);
center(0x2,0);
pulse(0x1,1500);
pulse(0x2,-1500);
command(0,0x33);

wait(0x3);

```


}

■ Example of continuous interpolation program for multichip linear interpolation

```

command(0x3,0xf); //----- Chip 1 mode setting -----
outpw(adr+wr1,0x0000); // Mode register 1
outpw(adr+wr2,0x0000); // Mode register 2
outpw(adr+wr3,0x0000); // Mode register 3
outpw(adr+wr4,0x0000); // General output register
outpw(adr+wr5,0c0400); // Interpolation mode register : Specify main chip
c2command(0x3,0xf); //----- Chip 2 mode setting -----
outpw(adr+wr1,0x0000); // Mode register 1
outpw(adr+wr2,0x0000); // Mode register 2
outpw(adr+wr3,0x0000); // Mode register 3
outpw(adr+wr4,0x0000); // General output register
outpw(adr+wr5,0x0800); // Interpolation mode register : Specify sub X and Y

accfst(0x1,0); //----- Chip 1 parameter setting -----
range(0x1,800000); // = 0
acc(0x1,300); // R = 800000 (Multiple = 10)
startv(0x1,100); // A = 100 (Acceleration/deceleration = 125KPPS/SEC)
speed(0x1,6000); // SV = 100 (Initial speed = 1000PPS)
// V = 6000 (Drive speed = 60000PPS)

c2accfst(0x1,0); //----- Chip 2 parameter setting -----
c2range(0x1,800000); // AO = 0
c2startv(0x1,6000); // R = 800000 (Multiple = 10)
c2speed(0x1,6000); // SV = 6000
// V = 6000

command(0x0,0x3e); // max clear
c2command(0x0,0x3e); // max clear
pulse(0x1,10000); // Seg 1x1 termination = 10000
pulse(0x2,-20000); // Seg 1y1 termination = -20000
c2pulse(0x1,-40000); // Seg 1x2 termination = -40000
c2pulse(0x2,40000); // Seg 1y2 termination = 40000
c2command(0x0,0x30);
command(0x0,0x30);

next_wait(); // Seg 2x1 termination = 10000
pulse(0x1,10000); // Seg 2y1 termination = -20000
pulse(0x2,-20000); // Seg 2x2 termination = -40000
c2pulse(0x1,-40000); // Seg 2y2 termination = 40000
c2pulse(0x2,40000);
c2command(0x0,0x30);
command(0x0,0x30);

next_wait(); // Seg 3x1 termination = 10000
pulse(0x1,10000); // Seg 3y1 termination = -20000
pulse(0x2,-20000); // Seg 3x2 termination = -40000
c2pulse(0x1,-40000); // Seg 3y2 termination = 40000
c2pulse(0x2,40000);
c2command(0x0,0x30);
command(0x0,0x3b); // Enable deceleration
command(0x0,0x30);

wait(0x1);

```

12. Electrical Characteristics

12.1 DC Characteristics

■ Absolute Maximum Rated

| Item | Symbol | Value | Unit |
|--------------------------|-----------|---------------------|------|
| Power Voltage | V_{DD} | -0.3 ~ +7.0 | V |
| Input voltage | V_{IN} | -0.3 ~ $V_{DD}+0.3$ | V |
| Input Current | I_{IN} | ±10 | mA |
| Preservation Temperature | T_{STG} | -40 ~ +125 | °C |

■ Recommend Operation Environment

| Item | Symbol | Value | Unit |
|---------------------|----------|-------------|------|
| Power Voltage | V_{DD} | 4.75 ~ 5.25 | V |
| Ambient Temperature | T_a | 0 ~ +85 | °C |

If the user wishes to operate the IC below 0°C, please make contact with our R&D engineer.

■ DC Characteristics

($T_a = 0 \sim +85^\circ\text{C}$, $V_{DD} = 5\text{V} \pm 5\%$)

| Item | Mark | Condition | Min. | Typ. | Max. | Unit | Remark |
|---------------------------|----------|--|---------------|------|------|------|------------------------------|
| High level input voltage | V_{IH} | | 2.2 | | | V | |
| Low level input voltage | V_{IL} | | | | 0.8 | V | |
| High level input current | I_{IH} | $V_{IN} = V_{DD}$ | -10 | | 10 | μA | |
| Low level input current | I_{IL} | $V_{IN} = 0\text{V}$ | -10 | | 10 | μA | D15~D0 Input signal |
| | | $V_{IN} = 0\text{V}$ | -200 | | -10 | μA | Input signal besides D15~D0 |
| High level output voltage | V_{OH} | $I_{OH} = -1\mu\text{A}$ | $V_{DD}-0.05$ | | | V | Note 1 |
| | | $I_{OH} = -4\text{mA}$ | 2.4 | | | V | Output signal besides D15~D0 |
| | | $I_{OH} = -8\text{mA}$ | 2.4 | | | V | D15~D0 Output signal |
| Low level output voltage | V_{OL} | $I_{OL} = 1\mu\text{A}$ | | | 0.05 | V | |
| | | $I_{OL} = 4\text{mA}$ | | | 0.4 | V | Output signal besides D15~D0 |
| | | $I_{OL} = 8\text{mA}$ | | | 0.4 | V | D15~D0 Output signal |
| Output leakage current | I_{OZ} | $V_{OUT}=V_{DD}$ or 0V | -10 | | 10 | μA | D15 ~ D0,BUSYN,INTN |
| Smith hysteresis voltage | V_H | | | 0.3 | | V | |
| Consuming current | I_{DD} | $I_{IO}=0\text{mA}, \text{CLK}=16\text{MHz}$ | | 28 | 50 | mA | |

Note 1 : BUSYN and INTN output signals have no items for high level output voltage due to the open drain output.

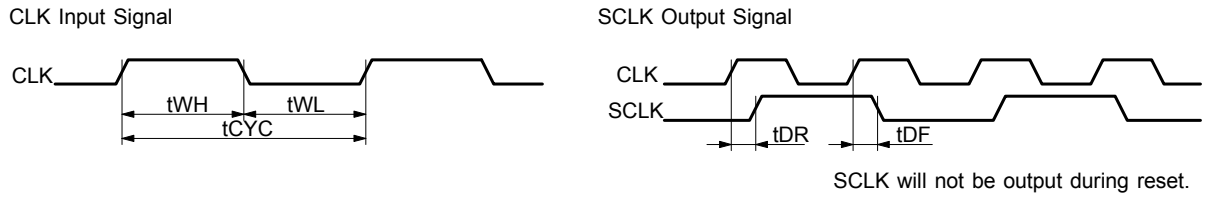
■ Pin Capacity

| Item | Mark | Condition | Min. | Typ. | Max. | Unit | Remark |
|------------------------|----------|--|------|------|------|------|------------------|
| Input/ Output capacity | C_{IO} | $T_a=25^\circ\text{C}$, $f=1\text{MHz}$ | | | 10 | pF | D15 ~ D0 |
| Input capacity | C_I | | | | 10 | pF | Other input pins |

12.2 AC Characteristics

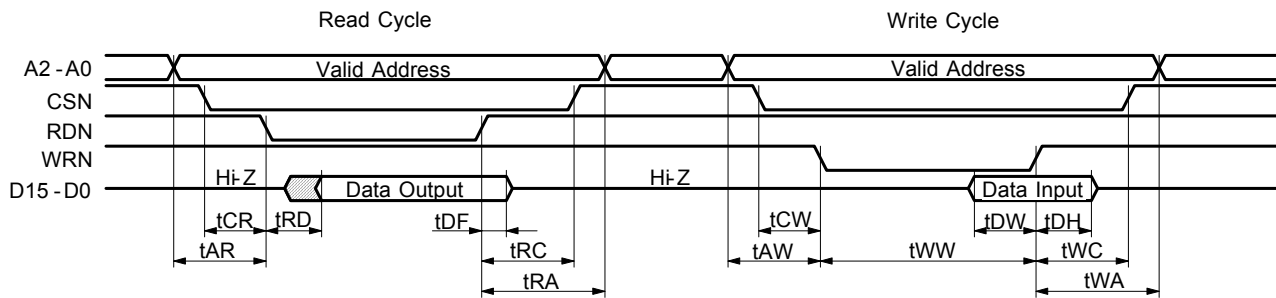
($T_a = 0 \sim +85^\circ\text{C}$, $V_{DD} = 5V \pm 5\%$, Output load condition: $85\text{ pF} + 1\text{ TTL}$)

12.2.1 Clock



| Symbol | Item | Min. | Max. | Unit |
|--------|---|------|------|------|
| tCYC | CLK Cycle | 62.5 | | nS |
| tWH | CLK Hi Level Width | 20 | | nS |
| tWL | CLK Low Level Width | 20 | | nS |
| tDR | CLK \uparrow \rightarrow SCLK \uparrow Delay Time | | 21 | nS |
| tDF | CLK \uparrow \rightarrow SCLK \downarrow Delay Time | | 23 | nS |

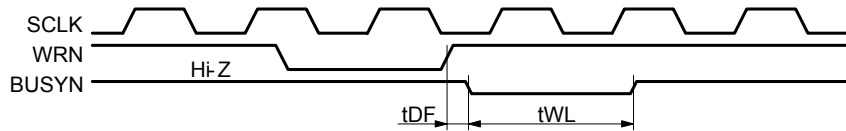
12.2.2 Read / Write Cycle



- The figure shown above is used for 16-bit data bus accessing ($H16L8 = \text{Hi}$). For 8-bit data bus ($H16L8 = \text{Low}$), the address signals shown in the figure become $A3\sim A0$, and data signals become $D7\sim D0$.
- At a read cycle, the data signal ($D15\sim D0$) becomes an output state as soon as both RDN and CSN become low and stays in the output state during tDF even if RDN is reset to High. Avoid the occurrence of bus conflict (collision).

| Symbol | Item | Min. | Max. | Unit |
|--------|---|------|------|------|
| tAR | Address Setup Time (to $RDN \downarrow$) | 0 | | nS |
| tCR | CSN Setup Time (to $RDN \downarrow$) | 0 | | nS |
| tRD | Output Data Delay Time (from $RDN \downarrow$) | | 29 | nS |
| tDF | Output Data Hold Time (from $RDN \uparrow$) | 0 | 30 | nS |
| tRC | CSN Hold Time (from $RDN \uparrow$) | 0 | | nS |
| tRA | Address Hold Time (from $RDN \uparrow$) | 0 | | nS |
| tAW | Address Setup Time (to $WRN \downarrow$) | 0 | | nS |
| tCW | CSN Setup Time (to $WRN \downarrow$) | 0 | | nS |
| tWW | WRN Low Level Width | 50 | | nS |
| tDW | Setup Time of Input Data (to $WRN \uparrow$) | 32 | | nS |
| tDH | Hold Time of Input Data (from $WRN \uparrow$) | 0 | | nS |
| tWC | CSN Hold Time (from $WRN \uparrow$) | 0 | | nS |
| tWA | Address Hold Time (from $WRN \uparrow$) | 5 | | nS |

12.2.3 BUSYN Signal



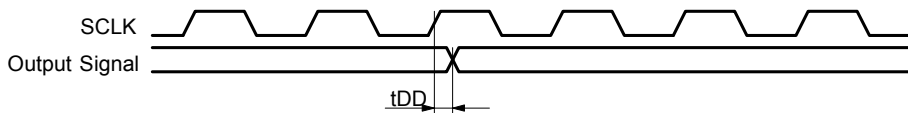
BUSYN becomes low active for maximum 2 SCLK cycles from WRN ↑. During the time, IC cannot accept Read/Write.

| Symbol | Item | Min. | Max. | Unit |
|--------|----------------------------|------|------------|------|
| tDF | WRN ↑ → BUSYN ↓ Delay Time | | 32 | nS |
| tWL | BUSYN Low Level Width | | tCYC×4 +30 | nS |

tCYC is a cycle of CLK.

12.2.4 SCLK/Output Signal Timing

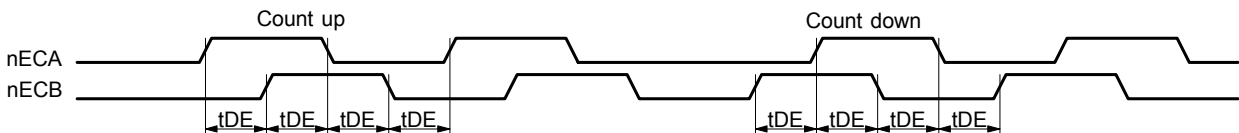
The following output single is synchronized with SCLK output signal. The level at ACLK ↑ will be changed.
Output signals : nPP/PLS, nPM/DIR, nDRIVE, nASND, nCNST, nDSND, nCMPP, nCMPM, nACASND and nACDSND.



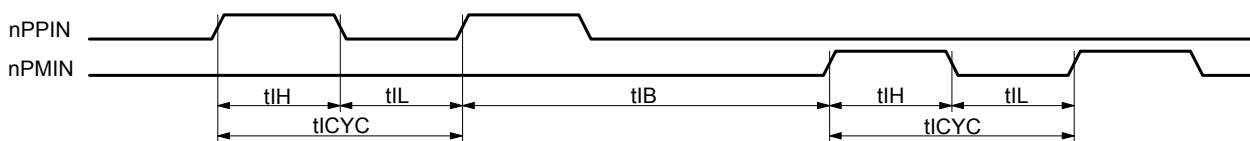
| Symbol | Item | Min. | Max. | Unit |
|--------|---------------------------------------|------|------|------|
| tDD | SCLK ↑ → Output Signal ↑ ↓ Delay Time | 0 | 20 | nS |

12.2.5 Input Pulses

■ Quadrature Pulses Input Mode (A/B phases)



■ Up / Down Pulses Input Mode



- a. In quadrature pulses input mode, when nECA and nECB input pulses are changed, the value of real position counter will be reflected in maximum 4 SCLK cycles.
- b. In UP/DOWN pulse input mode, the value of real position counter will be reflected in maximum 4 SCLK cycles from nPPIN and nPMIN input ↑.

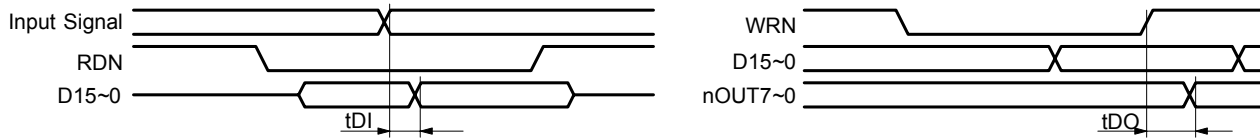
| Symbol | Item | Min. | Max. | Unit |
|--------|-------------------------------------|------------|------|------|
| tDE | nECA and nECB Phase Difference Time | tCYC×2 +20 | | nS |
| tIH | nPPIN and nPMIN Hi Level Width | tCYC×2 +20 | | nS |
| tIL | nPPIN and nPMIN Low Level Width | tCYC×2 +20 | | nS |
| tICYC | nPPIN and nPMIN Cycle | tCYC×4 +20 | | nS |
| tIB | nPPIN ↑ ↔ nPMIN ↑ between Time | tCYC×4 +20 | | nS |

tCYC is a cycle of CLK.

12.2.6 General Purpose Input / Output Signals

The figure shown at the lower left hand side illustrates the delay time when input signals nIN5 ~ 0, nEXPP, nEXPM, nINPOS, and nALARM are read through RR4 and RR5 registers.

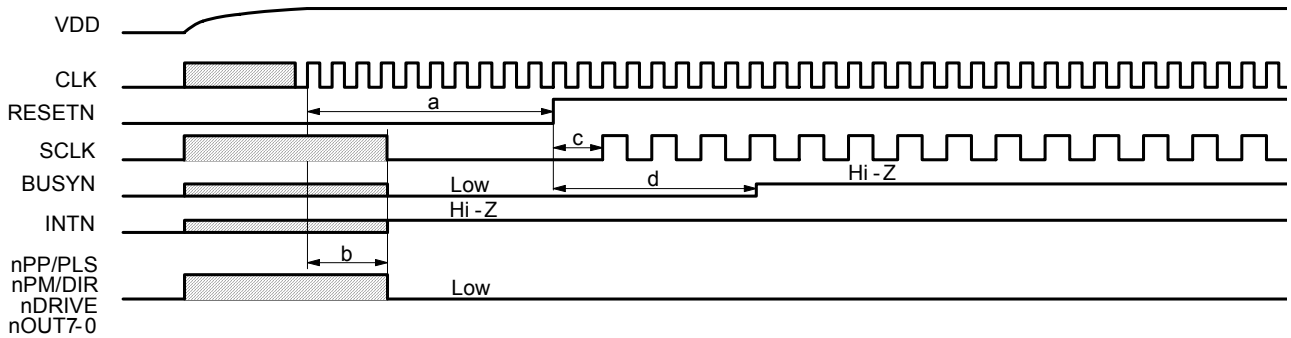
The figure shown at the lower right hand side illustrates the delay time when writing general output signal data into WR4.



| Symbol | Item | Min. | Max. | Unit |
|--------|--------------------------------|------|------|------|
| tDI | Input Signal → Data Delay Time | | 32 | nS |
| tDO | WRN ↓ → nOUT7~0 Setup Time | | 32 | nS |

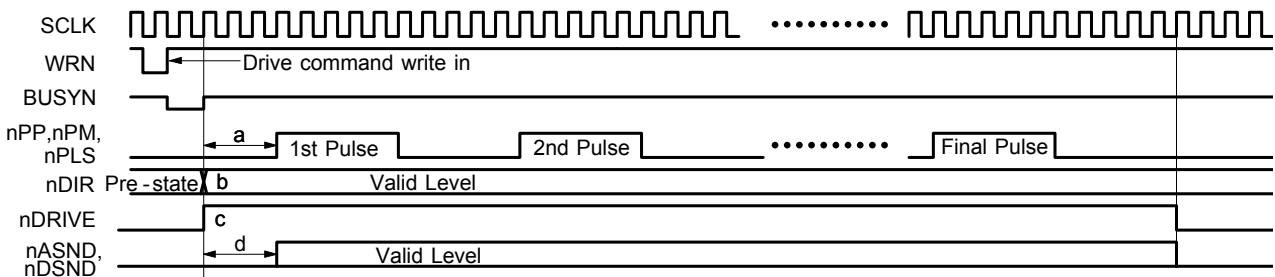
13. Timing of Input / Output Signals

13.1 Power-On Reset



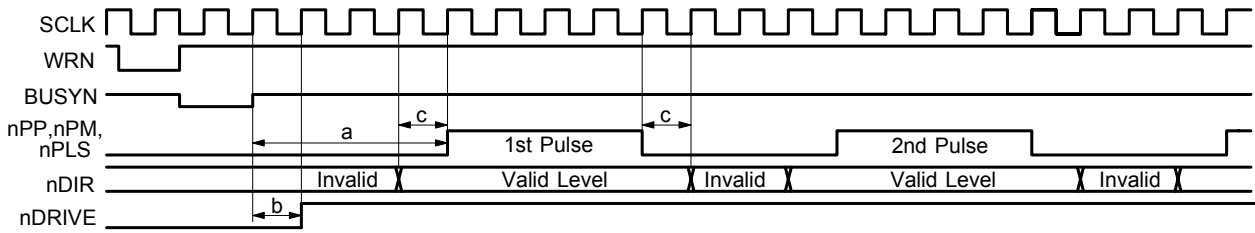
- a. The reset signal input to pin RESETN will keep on the Low level for at least 4 CLK cycles.
- b. When RESETN is on the Low level for 4 CLK cycles maximum, the output signals of MCX312 are decided.
- c. SCLK will be output after 2 CLK cycles when RESTN return to the Hi level.
- d. BUSYN keeps on the Low level for 8 CLK cycles when RESTN is on the Hi level.

13.2 Fixed or Continuous Driving



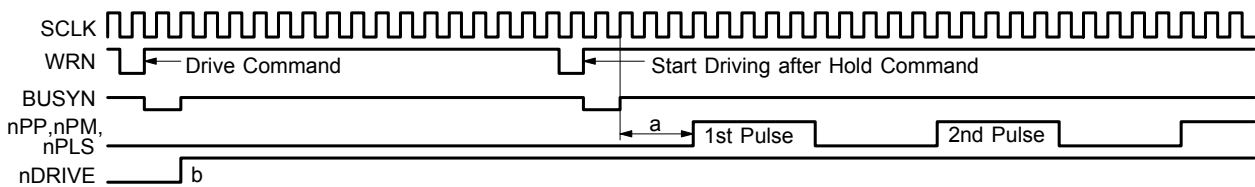
- a. Driving pulses (nPP, nPM, and nPLS) shown as above are positive logic pulses. And the first driving pulse will be output after 3 SCLK cycles from BUSYN ↑.
- b. nDIR (direction) signal is valid after 1 SCLK cycle from BUSYN ↑.
- c. dDRIVE becomes Hi level from BUSYN ↑.
- d. nASND and nDSND are on valid level after 3 SCLK cycles from BUSYN ↑ and they return to low level when the cycle of final pulse output has finished.

13.3 Interpolation



- a. The first pulses (nPP, nPM, and nPLS) of interpolation driving will be output after 4 SCLK cycles from BUSYN ↑.
- b. nDRIVE will become Hi level after 1 SCLK cycle from BUSYN ↑.
- c. When 1 pulse 1 direction type, nDIR signal is on valid level while Hi level pulse outputting and 1 SCLK cycle before and after Hi level pulse outputting. (When drive pulse is positive logic pulse.)

13.4 Start Driving after Hold Command



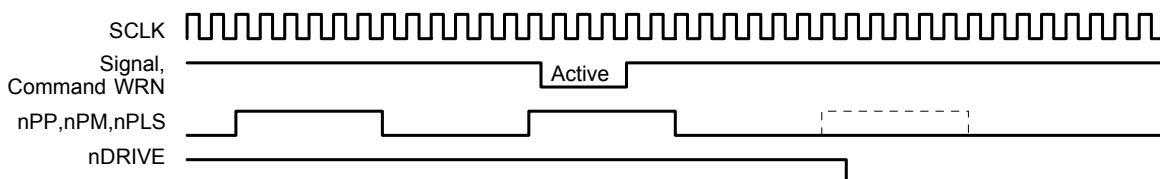
- a. The pulses (nPP, nPM, and nPLS) of each axis will start outputting after 3 SCLK cycles from BUSYN ↑.
- b. nDRIVE will become Hi level when BUSYN ↑ for each axis.

13.5 Sudden Stop

The following figure illustrates the timing of sudden stop. The sudden stop input signals are EMGN, nLMTP/M (When the sudden stop mode is engaged), and nALARM.

When sudden stop input signal becomes active, or the sudden stop command is written, it will stop the output of pulses immediately.

The width of external signals input for sudden stop must be more than 1 SCLK cycle. The stop function will not be active if the width is less 1 SCLK cycle.



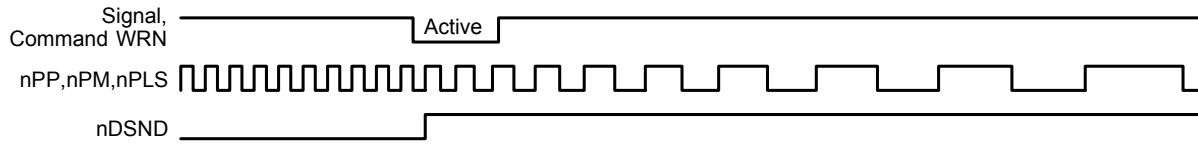
An instant stop input signal is loaded into IC by SCLK ↑.

When the input signal filter is enabled, the input signal will be delayed according to the constant value at filtering.

13.6 Decelerating Stop

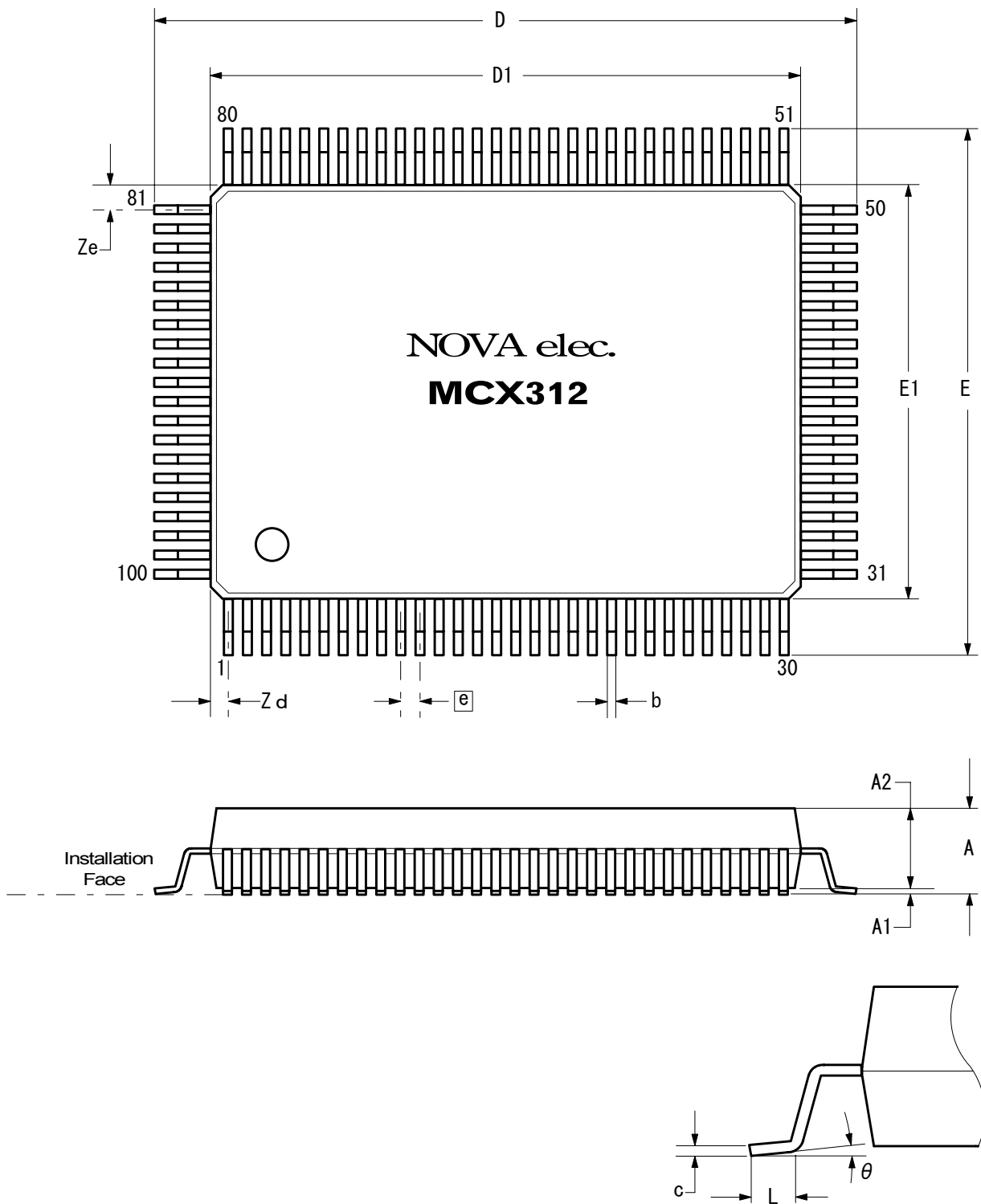
The following figure illustrates the timing of decelerating stop input signal and decelerating commands. The decelerating stop signal are nSTOP2 ~ 0 and nLMTP/M (When the decelerating mode is engaged).

When speed decelerating signals become active, or the decelerating stop command is written, the decelerating stop function will be performed.



If the input signal filter is disabled, the input signal is delayed according to the value of the time constant of the filter.

14. Package Dimensions



| Symbol | Size mm (inch) | | | Description |
|----------|-----------------|------------------|-----------------|---|
| | Minimum | Standard | Maximum | |
| A | — | — | 3.05 (0.12) | Height from the installation face to the top end of the package main unit |
| A1 | 0.09 (0.004) | 0.19 (0.007) | 0.29 (0.011) | Height from the installation face to the bottom end of the package main unit |
| A2 | 2.5 (0.098) | 2.7 (0.106) | 2.9 (0.114) | Height from the top to the bottom of the package main unit |
| b | 0.2 (0.008) | 0.3 (0.012) | 0.4 (0.016) | Pin width |
| c | 0.10 (0.004) | 0.15 (0.006) | 0.25 (0.010) | Pin thickness |
| D | 23.5 (0.925) | 23.8 (0.937) | 24.1 (0.949) | Maximum length in the package length direction including pins |
| D1 | 19.8 (0.780) | 20.0 (0.787) | 20.2 (0.795) | Length of the package main unit excluding pins |
| E | 17.5 (0.689) | 17.8 (0.701) | 18.1 (0.713) | Maximum length in the package width direction including pin |
| E1 | 13.8 (0.543) | 14.0 (0.551) | 14.2 (0.559) | Width of the package main unit excluding pins |
| e | 0.65 (0.026) | | | Pin pitch standard size |
| L | 0.6 (0.024) | 0.8 (0.031) | 1.0 (0.039) | Length of the flat section of the pins that contacts the installation face |
| Zd | — | 0.575 (0.023) | — | Length from the center of the outer-most pin to the outer-most pin section of the package main unit in the length direction |
| Ze | — | 0.825 (0.032) | — | Length from the center of the outer-most pin to the outer-most pin section of the package main unit in the width direction |
| θ | 0° | — | 10° | Angle of the pin flat section for the installation face |

15. Specifications

- **Control Axis** 2 axes
- **Data Bus** 16/8 bits selectable

Interpolation Functions

■ 2-axes Linear Interpolation

- Interpolation Range Each axis $-8,388,607 \sim +8,388,607$
- Interpolation Speed 1 ~ 4 MPPS
- Interpolation Accuracy ± 0.5 LSB (Within the range of whole interpolation)

■ Circular Interpolation

- Interpolation Range Each axis $-8,388,607 \sim +8,388,607$
- Interpolation Speed 1 ~ 4 MPPS
- Interpolation Accuracy ± 1 LSB (Within the range of whole interpolation)

■ 2 axes Bit Pattern Interpolation

- Interpolation Speed 1 ~ 4 MPPS (Dependent on CPU data writing time)

■ Related Functions of Interpolation

- Constant vector speed
- Single step interpolation (Command/external signals)
- Continuous interpolation
- Multichip axes linear interpolation

Common Specifications of Each Axis

■ Drive Pulses Output (When CLK = 16 MHz)

- Pulse Output Speed Range 1 ~ 4MPPS
- Pulse Output Accuracy within $\pm 0.1\%$ (according to the setting speed)
- S-curve Jerk $954 \sim 62.5 \times 10^6$ PPS/S² (Multiple = 1)
 $477 \times 10^3 \sim 31.25 \times 10^9$ PPS/S² (Multiple = 500)
- Accelerating / Decelerating Speed $125 \sim 1 \times 10^6$ PPS/S (Multiple = 1)
 $62.5 \times 10^3 \sim 500 \times 10^6$ PPS/S (Multiple = 500)
- Initial Speed 1 ~ 8,000PPS (Multiple = 1)
500PPS ~ 4×10^6 PPS (Multiple = 500)
- Drive Speed 1 ~ 8,000PPS (Multiple = 1)
500PPS ~ 4×10^6 PPS (Multiple = 500)
- Output-pulse Number 0 ~ 268,435,455 (fixed drive)
- Speed Curve Constant speed, linear acceleration,
parabola S-curve acceleration/deceleration drive
- Fixed Drive Deceleration Mode auto (non-symmetrical trapezoidal acceleration is also allowed) / manual
- Output-pulse numbers and drive speeds changeable during the driving
- Independent 2-pulse system or 1-pulse 1-direction system selectable
- Logical levels of drive pulse selectable

■ Encoder Input

- A/B quadrature pulse style or Up/Down pulse style selectable
- Pulse of 1, 2 and 4 divisions selectable (A/B quadrature pulse style)

■ Position Counter

- Logic Position Counter (for output pulse) range $-2,147,483,648 \sim +2,147,483,647$
 - Real Position Counter (for feedback pulse) range $-2,147,483,648 \sim +2,147,483,647$
- Data read and write possible

■ Comparison Register

- COMP + Register Position comparison range $-1,073,741,824 \sim +1,073,741,824$
- COMP – Register Position comparison range $-1,073,741,824 \sim +1,073,741,824$
- Status and signal outputs for the comparisons of position counters
- Software limit functioned

■ Interrupt (Interpolations Excluded)

- The factors of occurring interrupt:
 - ..the drive-pulse outputting
 - ..the start / finish of a constant-speed drive during the acceleration / deceleration driving
 - ..the end of the driving
 - ..the volume of position counter \geq the volume of COMP–
 - ..the volume of position counter $<$ the volume of COMP–
 - ..the volume of position counter \geq the volume of COMP+
 - ..the volume of position counter $<$ the volume of COMP+

Enable / disable for these factors selectable

■ External Signal for Driving

- EXPP and EXPM signals for fixed pulse / continuous drive
- Driving in manual pulsar mode (encoder input)

■ External Deceleration / Sudden Stop Signal

- STOP0 ~ 2 3 points for each axis
- Enable / disable and logical levels selectable

■ Servo Motor Input Signal

- ALARM (Alarm), INPOS (In Position Check)
- Enable / disable and logical levels selectable

■ General Input / Output Signal

- IN0 ~ 5 6 points for each axis (wherein 4 points use with multichip interpolation signal pin)
- OUT0 ~ 7 8 points for each axis (wherein it uses with drive status output pin)

■ Driving Status Signal Output

- DRIVE (drive pulse outputting), ASND (speed accelerating), CNST (constant driving), DSND (speed decelerating), CMPP (position \geq COMP+), CMPM (position $<$ COMP-), ACASND (acceleration/deceleration speed increase), ACDSND (acceleration/deceleration speed decrease)
- Drive status and status registers readable

■ Limit Signals Input

- 1 point, for each + and - side
- Logical levels and decelerating / sudden stop selectable

■ Emergency Stop Signal Input

- EMG, 1 point in all axes
- Sudden stop the drive pulse of all axes when on Low level

■ Contents of integral type filters

- Equipped with integral filters in the input column of each input signal. One time constant can be selected from eight types.

■ Electrical Characters

- Temperature Range for Driving 0 ~ + 85°C (32°F ~185°F)
- Power Voltage for Driving +5V \pm 5 %
- Input / Output Signal Level CMOS, TTL connectable
- Input Clock Pulse 16,000 MHz (Standard)

■ Package

100-pin plastic QFP, pitch = 0.65mm
Dimension : 23.8 × 17.8 × 3.05 mm

Appendix A Profile of Speed curve

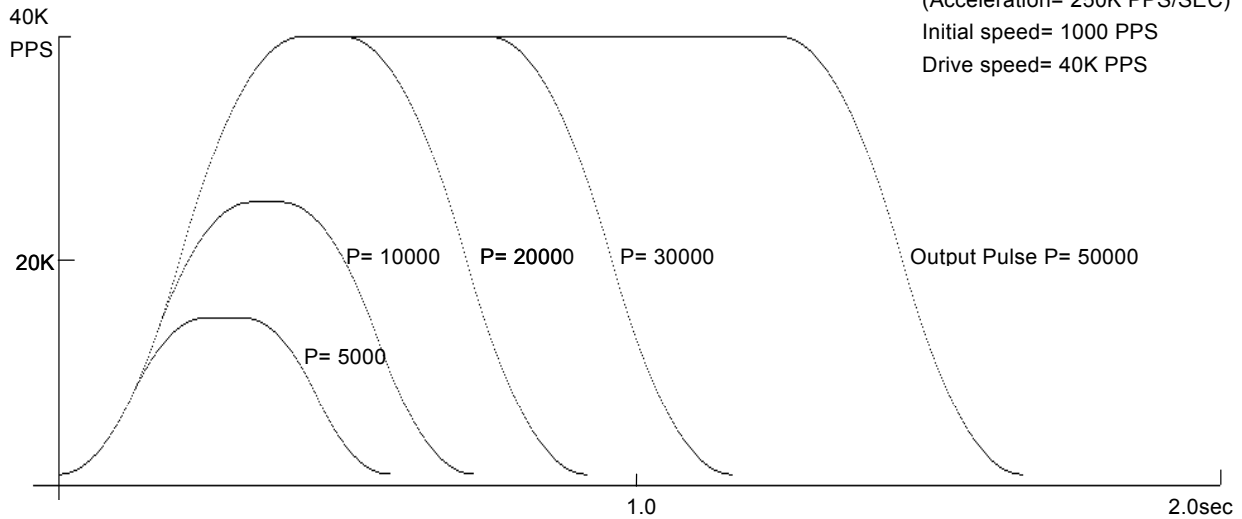
The following curves are based on the test records from MCX312 output drive pulses and speed curve traces.

The perfect S curve acceleration / deceleration is the curve drive, without linear acceleration / deceleration, before the appointed drive speed is reached. Partial S curve acceleration / deceleration is with a period of linear acceleration / deceleration before the appointed drive speed is reached.

■ 40KPPS Perfect S curve acceleration/deceleration

R=800000 (Multiple: 10), K=700, (A=D=200), SV=100, V=4000, A0=0
Auto Deceleration mode

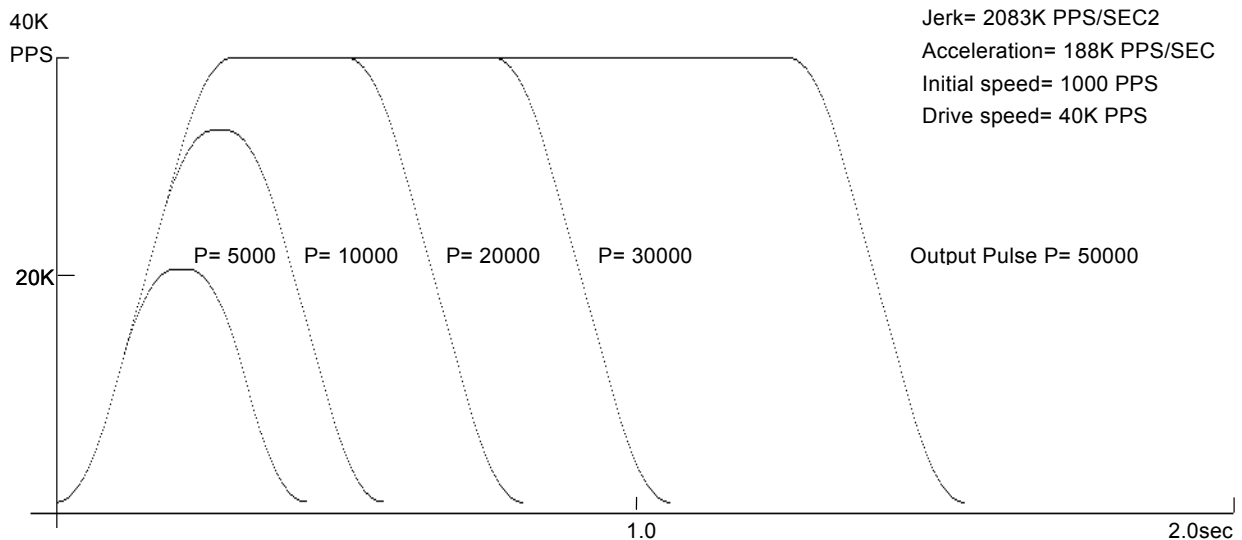
Jerk= 893K PPS/SEC2
(Acceleration= 250K PPS/SEC)
Initial speed= 1000 PPS
Drive speed= 40K PPS



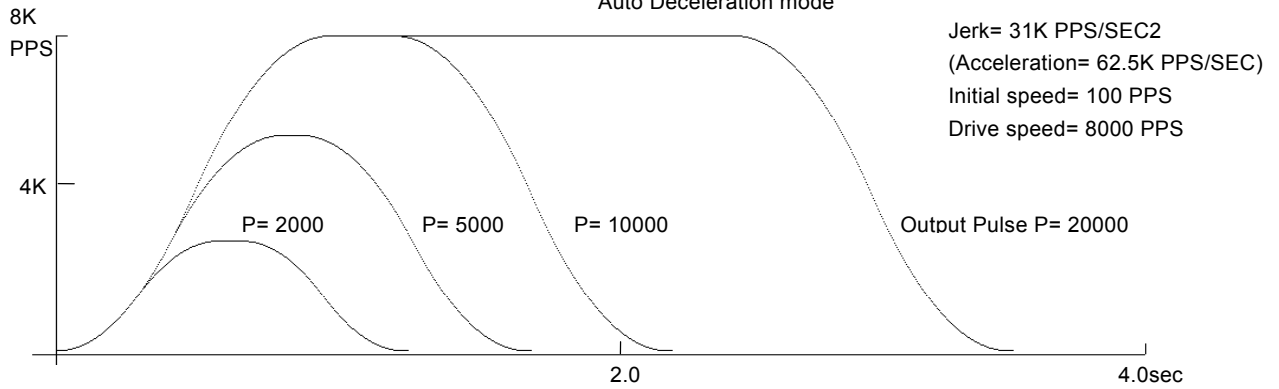
■ 40KPPS Partial S curve acceleration/deceleration

R=800000 (Multiple: 10), K=300, (A=D=150), SV=100, V=4000, A0=0
Auto Deceleration mode

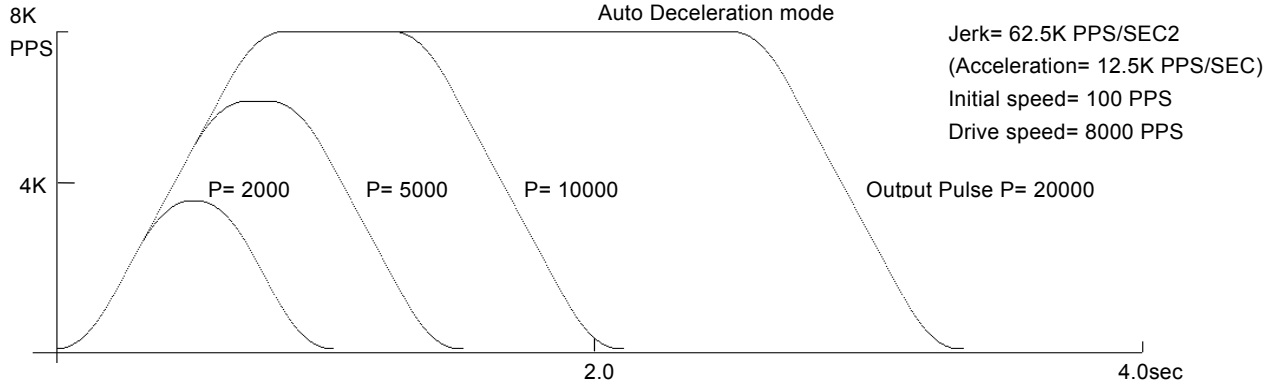
Jerk= 2083K PPS/SEC2
Acceleration= 188K PPS/SEC
Initial speed= 1000 PPS
Drive speed= 40K PPS



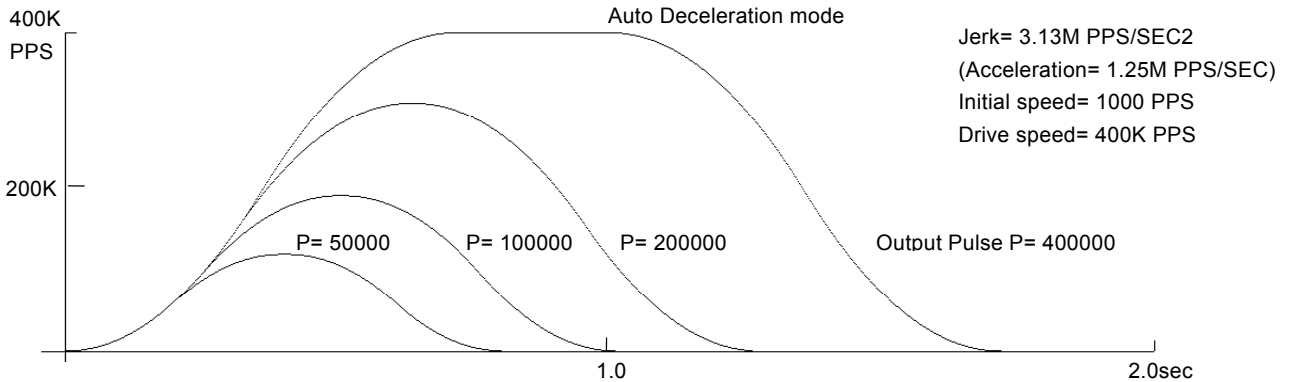
■ 8000PPS Perfect S curve acceleration/deceleration R=8000000 (Multiple: 1), K=2000, (A=D=500), SV=100, V=8000, A0=0
Auto Deceleration mode



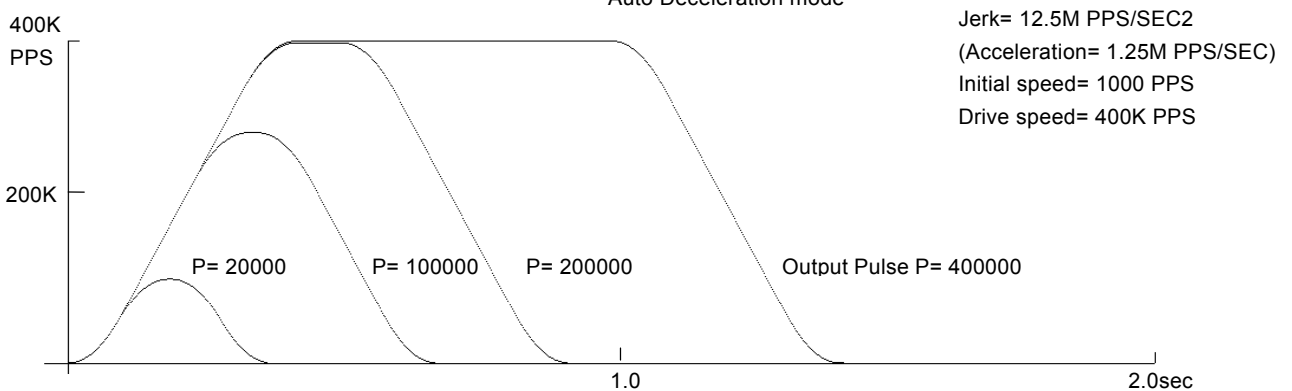
■ 8000PPS Partial S curve acceleration/deceleration R=8000000 (Multiple: 1), K=1000, A=D=100, SV=100, V=8000, A0=0
Auto Deceleration mode



■ 400KPPS Perfect S curve acceleration/deceleration R=80000 (Multiple: 100), K=2000, (A=D=100), SV=10, V=4000, A0=0
Auto Deceleration mode



■ 400KPPS Partial S curve acceleration/deceleration R=80000 (Multiple: 100), K=500, (A=D=100), SV=10, V=4000, A0=0
Auto Deceleration mode

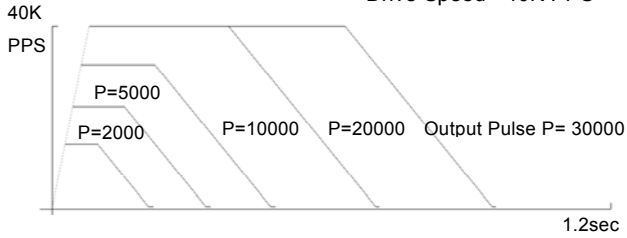


■ 40KPPS Non-symmetrical Trapezoidal acceleration/deceleration

Individual acceleration/deceleration: WR3/D1=1,
Triangle form prevention ON : WR3/D5=1

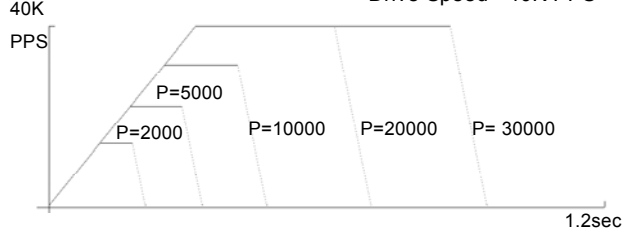
R=800000 (Multiple: 10), A=400, D=100, SV=50, V=4000, A0=0

Acceleration = 500K PPS/SEC
Deceleration= 125K PPS/SEC
Initial Speed= 500PPS
Drive Speed= 40K PPS



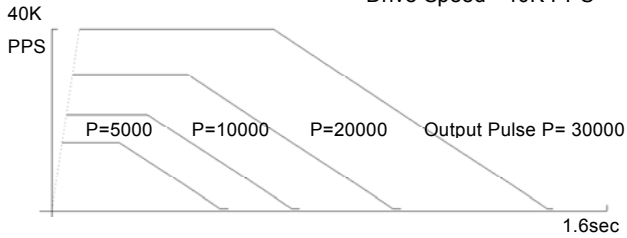
R=800000 (Multiple: 10), A=100, D=400, SV=50, V=4000, A0=0

Acceleration = 125K PPS/SEC
Deceleration= 500K PPS/SEC
Initial Speed= 500PPS
Drive Speed= 40K PPS



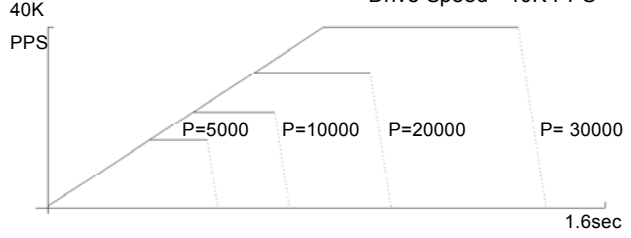
R=800000 (Multiple: 10), A=400, D=40, SV=50, V=4000, A0=0

Acceleration= 500K PPS/SEC
Deceleration= 50K PPS/SEC
Initial Speed= 500PPS
Drive Speed= 40K PPS



R=800000 (Multiple: 10), A=40, D=400, SV=50, V=4000, A0=0

Acceleration = 50K PPS/SEC
Deceleration= 500K PPS/SEC
Initial Speed= 500PPS
Drive Speed= 40K PPS



Appendix B Important notice

I Notice for fixed pulse or continuous pulse driving in S-curve acceleration/deceleration

[Symptom]

When using fixed pulse driving or continuous pulse driving in S-curve acceleration/deceleration, driving is performed at an initial speed without acceleration.

[Occurrence Condition]

1. When fixed pulse driving or continuous pulse driving is performed in S-curve acceleration / deceleration mode (WR3/D2=1),
2. the value of (Drive speed V - Initial speed SV) is lower than the half value of (V-SV) which is driven just before.
3. And in the driving just before, when ACC counter (which is used in S-curve acceleration / deceleration driving in the IC) is not returned to 0 at the end of driving.

[Workaround]

Before start of driving, use following steps to workaround this behavior, with an inspection command to clear the internal ACC counter. This command is not described in the manual.

When fixed pulse driving or continuous pulse driving is performed in S-curve acceleration/ deceleration mode (WR3/D2=1), write 44h command just before all the drive commands are written.

(Example)

Set mode for S-curve acceleration/deceleration

Set Range (R)

Set Jerk (K)

Set Acceleration (A)

Set Initial speed (SV)

Set Drive speed (V)

Set Output pulse number (P)

WR0 ← Axis assignment+44h ;Command for workaround

WR0 ← Axis assignment+20h ;Fixed pulse drive in the +direction

Waits for termination of driving

|

Set Output pulse number (P)

WR0← Axis assignment+44h ;Command for workaround

WR0 ← Axis assignment+21h ;Fixed pulse drive in the –direction

Waits for termination of driving

|

Change Drive speed (V)

WR0 ← Axis assignment+44h ;Command for workaround

WR0 ← Axis assignment+22h ;Continuous pulse drive in the +direction

Waits for termination of driving

II Notice for Compare Register

[Symptom]

Although Range of Position Comparison between Position Counter and Compare Register(COMP+,-) is shown as -2,147,483,648 to +2,147,483,647 (signed 32-bit)on User's Manual, actually it is -1,073,741,824 to +1,073,741,823(signed 31-bit)due to the defect of IC.

[Workaround]

Don't compare the values over the range such as -1,073,741,824 to +1,073,741,823.

III Notice for Fixed pulse driving in S-curve acceleration/deceleration

[Symptom]

When using fixed pulse driving in S-curve acceleration/deceleration, if one of the following occasions ① to ④ is taken just before finishing driving, pulse may be continuously outputted depending on setting value of parameters.

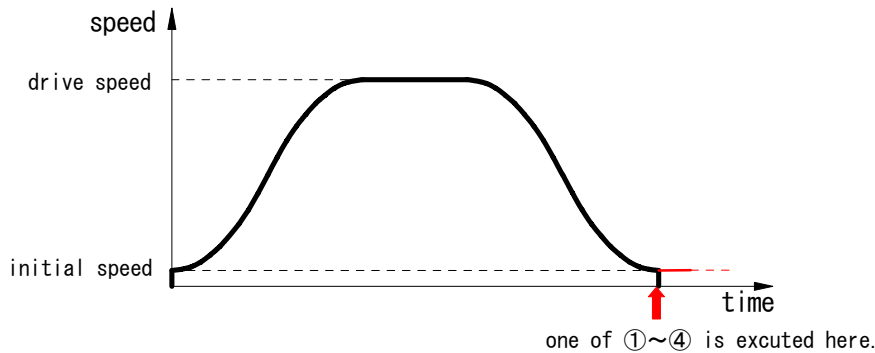


Figure 1. Speed profile of fixed pulse driving in S-curve acceleration/deceleration

- ① When decelerating stop command (26h) was given just before finishing driving
 - ② When setting stop mode of hardware limit(nLMTP/M signal) as decelerating stop(WR2/D2=1), driving is started and hardware limit of progress direction becomes active just before finishing driving
 - ③ When software limit is enabled(WR2/D0,1=1), driving is started and hardware limit of progress direction becomes active just before finishing driving
 - ④ When nSTOP(2~0)signals are enabled(WR1/D5,3,1), fixed pulse driving is started and those signals become active just before finishing driving
- This trouble won't happen when trapezoidal(liner) acceleration/deceleration drive or constant speed drive is performed.
 - This trouble won't be happened when continuous pulse driving in S-curve acceleration/deceleration.
 - This trouble won't happen when any of Sudden stop command, EMGN signal, LMT signal of Sudden stop and ALARM signal is outputted.

Fixed pulse driving in S-curve acceleration/deceleration is performed as that driving speed and initial speed becomes equal and acceleration becomes zero(0) when driving finished, which means all pulse outputted. But it's impossible to make driving speed & initial speed equal and acceleration zero(0) in all combinations of parameters because of the problem of calculation accuracy. This trouble will happen if it is in the above mentioned occasion ①, ②, ③ or ④ as the factors of deceleration stop accidentally and driving speed hasn't reached initial speed yet but acceleration has become zero(0) showing on Figure 2.

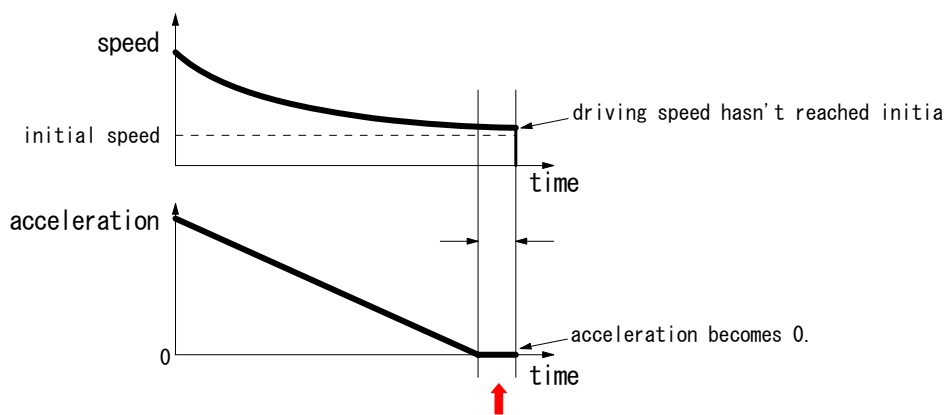


Figure 2. Driving speed and acceleration just before driving completion

Deceleration status of accelerating(ASND), constant speed driving(CNST) and decelerating(DSND) can be read out according to RR1 register of IC. The status is shown as Figure 3. as below ;

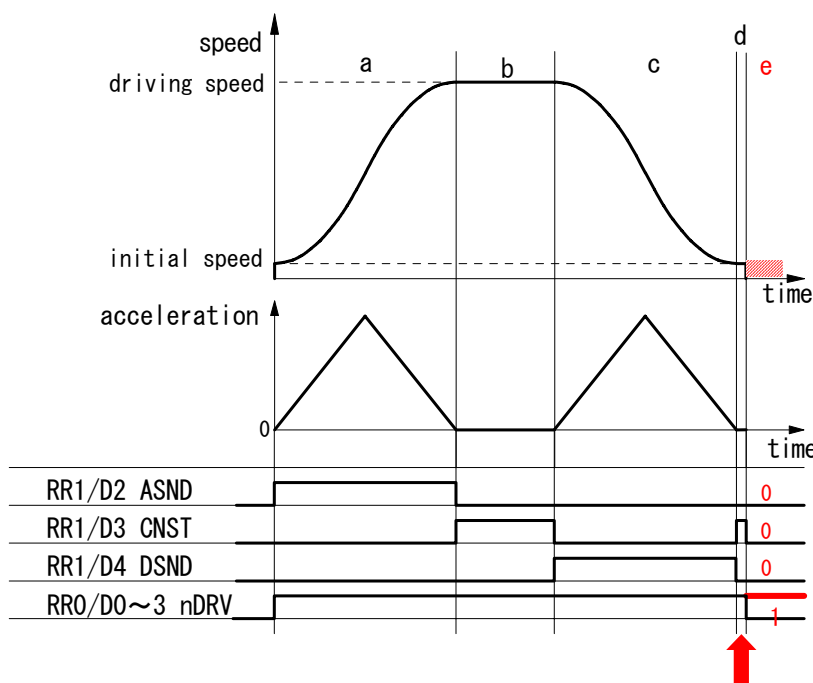


Figure 3. Status of deceleration shown by RR1 register

The timing when the trouble may possibly happen is shown as “d” on Figure 3. At this time, status of deceleration is constant speed driving(CNST=1). Besides in case the trouble such as outputting pulse continuously happens, all of ASND, CNST and DSND will become 0 even though it's driving(RR0/nDRV=1).

[Workaround]

1 When deceleration stop command(26h) is executed 【Case ①】

Basically once deceleration starts, it isn't necessary to execute deceleration stop command. Therefore deceleration stop command should be prohibit. See nRR1/D4(DSND) normally to know it's decelerating or not. Timing when the trouble happens is in “d” as shown on Figure 3 if deceleration stop command is executed. In “d” DSND status bit is 0 and CNST status bit is 1. Therefore the following two(2) solutions will be proposed.

(1) When interruption from IC can be used

When deceleration starts, interruption will be occurred to prohibit execution of deceleration stop command(26h) until driving completion. Flag of prohibition against deceleration stop command is prepared and it will be clear before driving starts. Enable constant speed area completion interruption(WR1/D13(C-END)=1). Fixed pulse driving starts and if interruption is occurred, read RR3/D5(C-END) within interruption process routine and status bit is 1, constant speed area is completed which is same as deceleration starts. So make flag of prohibition against deceleration stop command 1. Besides it is possible that CNST(constant speed area) will occur just before driving completion. Then return WR1/D13 status bit from 1 to 0 in order to prevent interruption. On the other hand, within task, see flag and if it's 1, don't execute deceleration stop command.

(2) When interruption can't be used

Terms when deceleration stop command should be executed in accelerating and constant speed driving such as “a” and “b” on Figure 3. Status of deceleration for both “d” (timing when the trouble happens) and “b” (constant speed driving) is same as constant speed driving as shown on Figure 3. But there is difference of driving speed between two. Driving speed in “b” is near that of setting driving speed. Driving speed in “d” is near that of initial speed. Therefore before driving starts, prepare judgmental speed which is middle one between initial speed and setting speed ($(\text{Driving speed} - \text{initial speed})/2 + \text{initial speed}$). When execute deceleration stop command during driving, make sure status is accelerating(ASND=1) or constant speed driving(CNST=1) and driving speed is same or faster than judgmental speed.

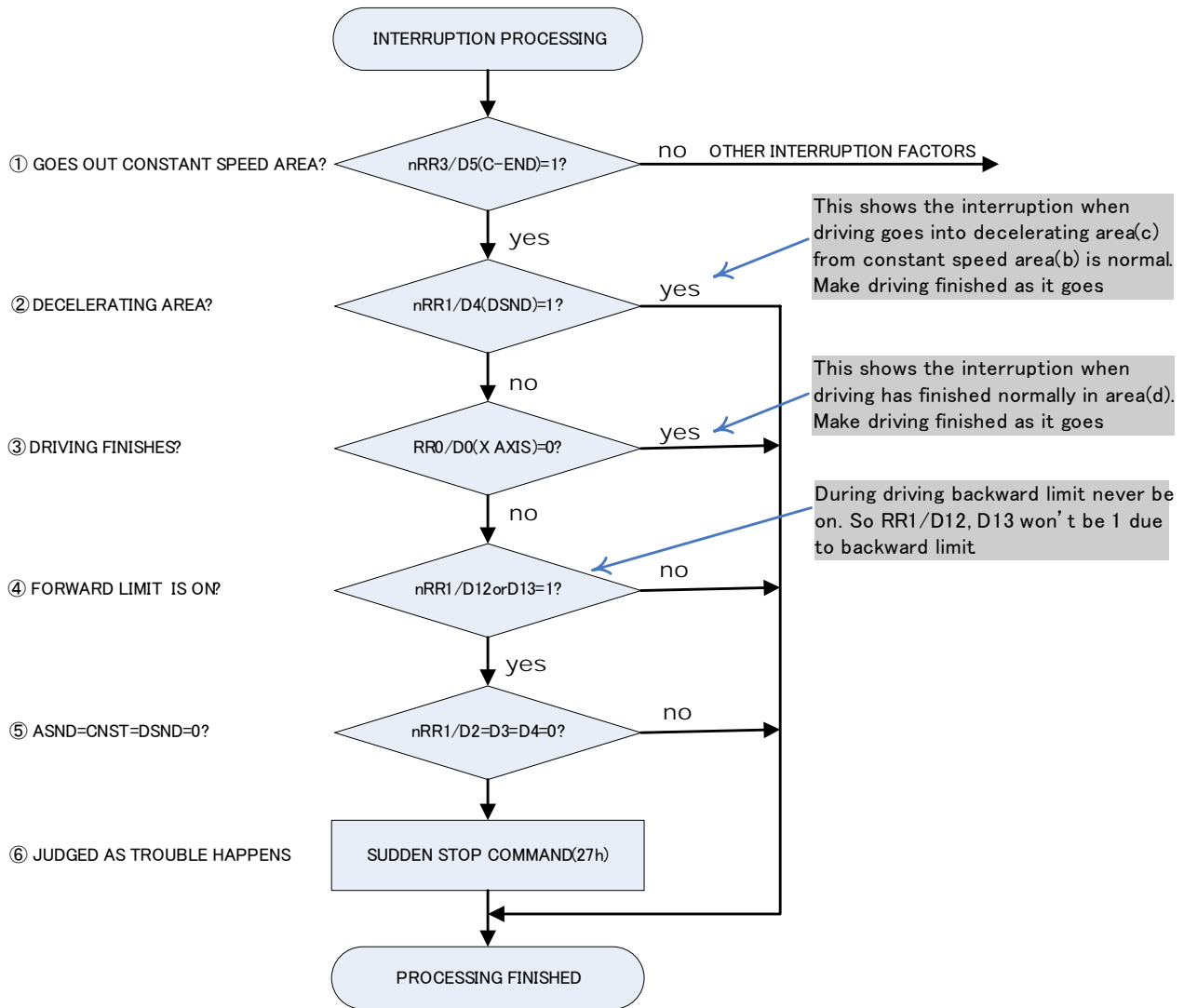
2 Hardware limit(nLMTP/M signal) at deceleration stop mode 【Case ②】

Basically use hardware limit(nLMTP/M signal) at Sudden stop mode when fixed pulse driving in S-curve acceleration/deceleration. Out of necessity, when use hardware limit at deceleration stop mode, prepare the following measures. If multi-axes are controlled in the same time, (1) When interruption from IC can be used is effective.

(1) When interruption from IC can be used

Constant speed area (“b” area) on Figure 3 is set as occurrence factor of interruption. But this interruption is also happened when there is “d” area just before driving stops on Figure 3 such as when driving speed reaches initial speed or when acceleration becomes 0(zero). There should occur “d” area definitely if the trouble is happened just before driving stops, driving speed hasn't reached initial speed yet, acceleration becomes 0 and decelerating stop requirement is executed. The measure is to judge if the trouble happens or not in the timing of interruption when “d” area ends.

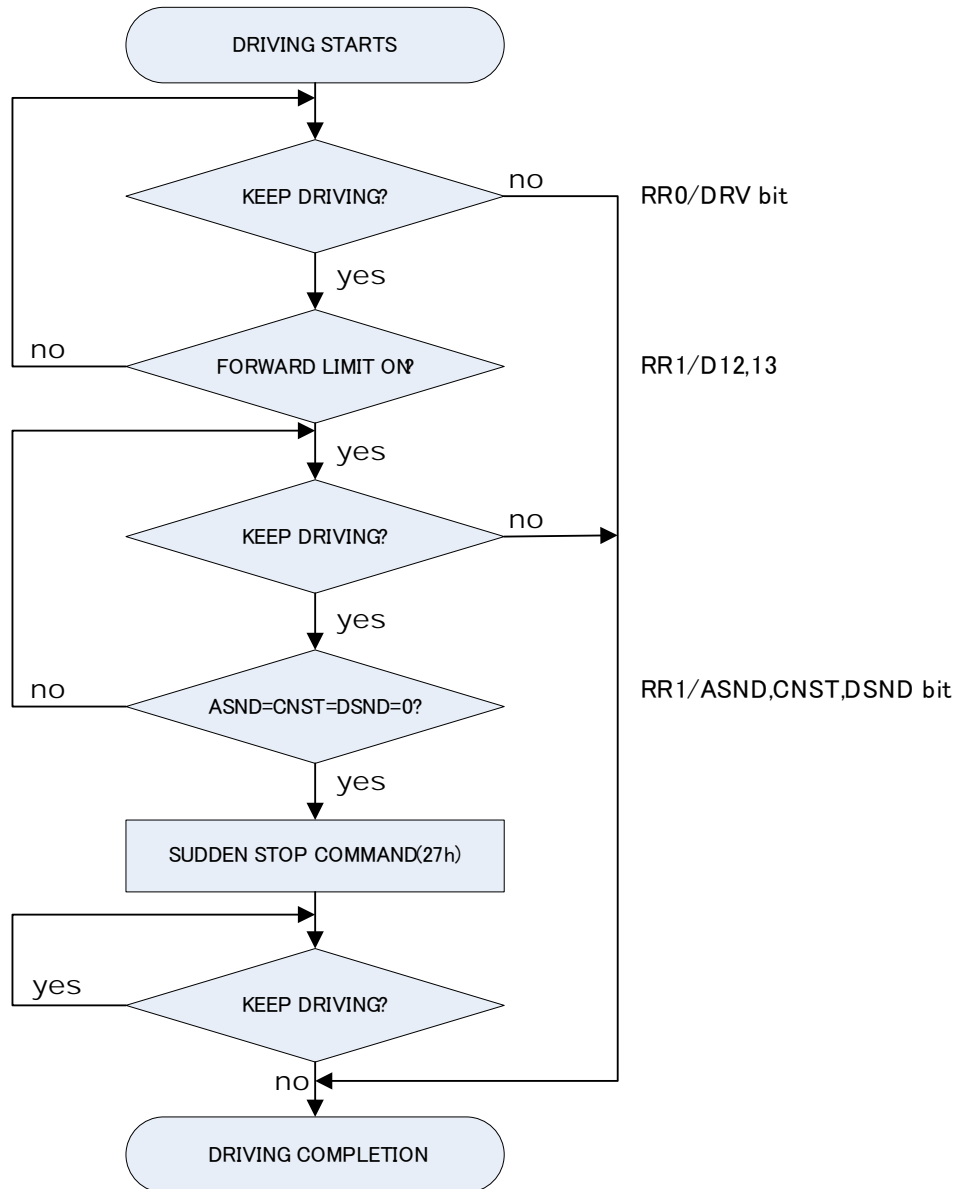
Make constant speed area finishing interruption of IC active(WR1/D13(C-END)=1). Start fixed pulse driving in S-curve acceleration/deceleration and when interruption is happened, interruption processing is executed as follows;



- ① Check if driving goes out constant speed area or not. If RR3/D5(C-END) bit of driving axis is 0, there are other interruption factors. Execute those interruption processing.
- ② Check if driving goes into decelerating area or not. If it's RR1/D4(DSND)=1, make it finished as it goes because driving goes into "c" area from "b" area on Figure 3. If it's RR1/D4(DSND)=0, move it to ③ processing because driving has gone out "d" area.
- ③ Check if driving has finished or not. If it's finished, make it finished as it goes because driving has finished normally. But if it's not finished yet, surely trouble have happened because status is shown as it's still driving even though driving has gone out "d" area.
- ④ Check if hard limit is ON or OFF. Since RR1/D12 bit becomes 1 if + limit is ON and RR1/D13 bit becomes 1 if - limit is ON, if it's D12=1 or D13=1, judge forward limit is ON.
- ⑤ If trouble has happened, it becomes ASND=CNST=DSND=0. Check them all.
- ⑥ Execute sudden stop command(27h).

(2) When interruption can't be used

In “d” area on Figure 3, if forward limit is active, the trouble will happen, more exactly saying, it will seldom happen. There isn't any method to avoid it before it happens. So immediately after the trouble happens, take means to stop driving. If the trouble happens (“e” area on Figure 3), status keeps driving(RR0/nDRV=1), status of acceleration/deceleration is 0 in ASND, CNST, DSND. This is completely abnormal state. So the example of measure is shown as follows;



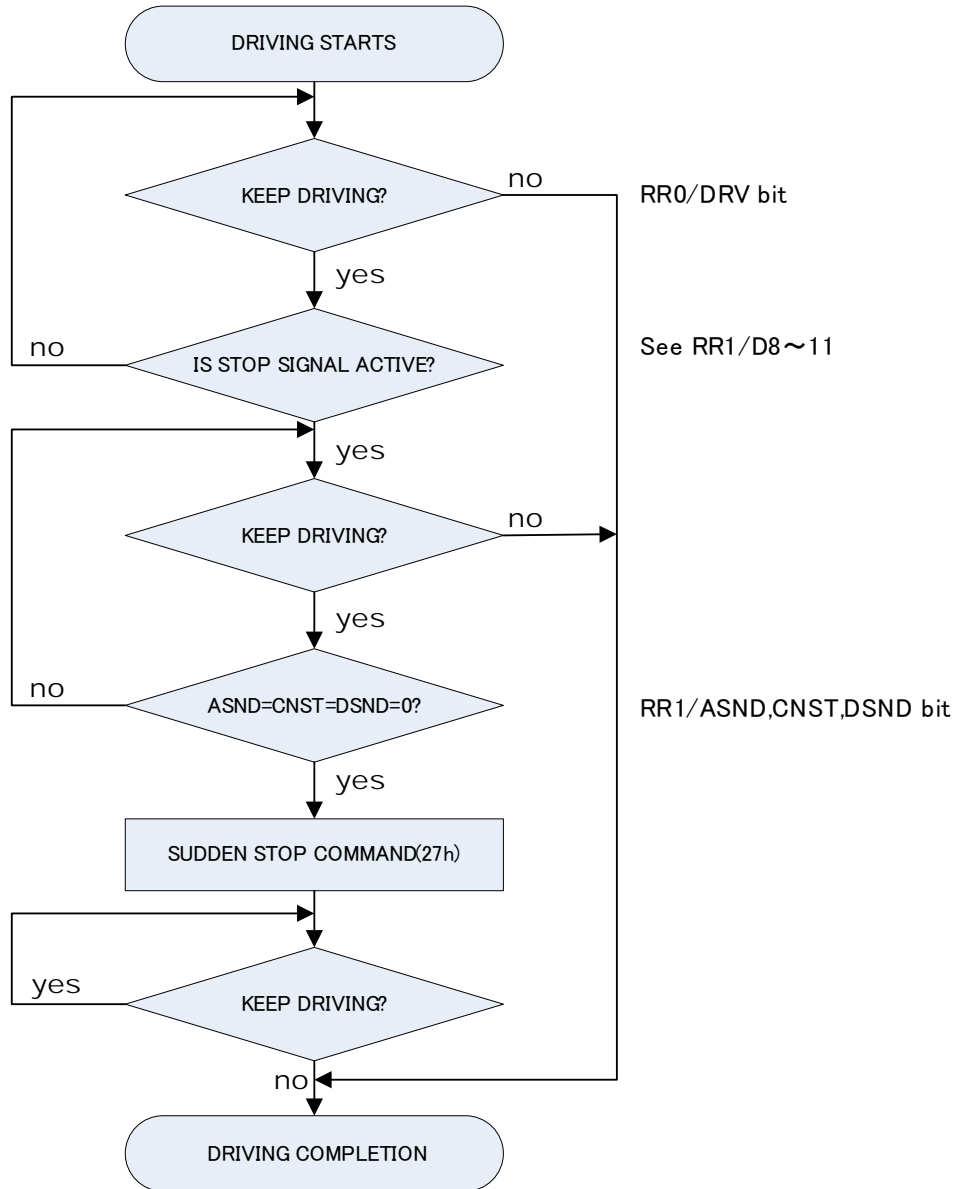
When start fixed pulse driving in S-curve acceleration/deceleration, always read out status of progress directional hardware limit(RR1/D12,D13) with timer interruption. If limit signal becomes active, read out bit status of ASND(D2), CNST(D3) and DSND(D4) in RR1 register and execute Sudden stop command(27h) one time only if all those 3 bits are 0.

3 Software limit 【Case ③】

In fixed pulse driving, the target position(goal) can be calculated from present position(logical position counter value) and number of output pulse. If target position is over value of software limit, don't drive to avoid the trouble.

4 Deceleration stop with STOP(2~0) Signal【Case ④】

Normally deceleration stop with STOP(2~0)signal is performed during continuous pulse driving. But out of necessity, when deceleration stop with STOP(IN) signal is performed during fixed pulse driving in S-curve deceleration, there isn't any method to avoid it before the trouble happens like chapter 2.2. So the example of measure is shown as follows;



IV Notice for continuous interpolation

[Symptom]

In executing continuous interpolation, when writing the interpolation command for the next segment during outputting the final pulse of the present segment, the irregular driving will occur, which keeps outputting pulses.

[Workaround]

- Finish writing interpolation data and command executed by CPU before the final pulse for each segment is output. It needs to set output pulse number and drive speed for each segment to keep enough time to write interpolation data and command.
- Set more than "2" at minimum as the number of output pulse for each segment. If the setting is "1", irregular driving will be executed.

If you need more assistance, please e-mail us at novaelec_info@novaelec.co.jp.