

Ver 2.0



W O b i t 6 1 - 5 4 8 P o z n a n , u l . W i e r z b i e c i c e 6 6 / 7

Contents

1. OUTLINE	1
2. The Descriptions of Functions.....	4
2.1 Pulse Output Command.....	4
2.1.1 Fixed Pulse Driving Output	4
2.1.2 Continuous Pulse Driving Output.....	5
2.2 Speed Curve.....	5
2.2.1 Constant Speed Driving	5
2.2.2 Trapezoidal Driving.....	6
2.2.3 S-curve Acceleration/Deceleration Driving	7
2.2.4 Pulse Width and Speed Accuracy.....	9
2.3 Position Control	11
2.3.1 Logic Position Counter and Real position Counter	11
2.3.2 Compare Register and Software Limit.....	11
2.4 Interpolation	12
2.4.1 Linear Interpolation.....	12
2.4.2 Circular Interpolation.....	14
2.4.3 The Bit Pattern Interpolation	16
2.4.4 Constant Vector Speed	19
2.4.5 Continuous Interpolation	21
2.4.6 The Acceleration / Deceleration Control in Interpolation.....	24
2.4.6 Single-step interpolation (from Command or External Signal)	27
2.5 Interrupt.....	28
2.6 Other Functions.....	29
2.6.1 Driving By External Pulses.....	29
2.6.2 Pulse Output Type Selection	30
2.6.3 Pulse Input Type Selection.....	30
2.6.4 Hardware Limit Signals.....	30
2.6.5 Interface to Servo Motor Drivers	31
2.6.6 Emergency Stop	31
2.6.7 Status Output	31
2.6.8 General Purpose Output.....	31
3. Pin Assignments and Signal Description.....	33
4. Register.....	38
4.1 Register Address by 16-bit Data Bus	38
4.2 Register Address by 8-bit Data Bus.....	40
4.3 Command Register: WR0.....	41
4.4 Mode Register1: WR1	41
4.5 Mode Register2: WR2	42
4.6 Mode Register3: WR3	43
4.7 Output Register: WR4	45
4.8 Interpolation Mode Register: WR5	45
4.9 Data Register: WR6/WR7	46
4.10 Main Status Register: RR0	47
4.11 Status Register 1: RR1	48
4.12 Status Register 2: RR2	49
4.13 Status Register 3: RR3	49
4.14 Input Register: RR4 / RR5.....	50
4.15 Data-Read Register: RR6 / RR7.....	50
5. Command Lists	51
6. Commands for Data Writing.....	53
6.1 Range Setting	53
6.2 S-curve Acceleration Rate Setting	53
6.3 Acceleration Setting.....	54
6.4 Deceleration Setting.....	54
6.5 Initial Speed Setting	54
6.6 Drive Speed Setting.....	55

6.7 Output Pulse Number / Interpolation Finish Point Setting.....	55
6.8 Manual Decelerating Point Setting.....	55
6.9 Circular Center Setting.....	56
6.10 Logical Position Counter Setting.....	56
6.11 Real position Counter Setting.....	56
6.12 COMP+ Register Setting.....	56
6.13 COMP- Register Setting.....	56
6.14 Acceleration Counter Offsetting.....	57
6.15 NOP (Used for Axis Switching).....	57
7. Commands for Reading Data.....	58
7.1 Logical Position Counter Reading.....	58
7.2 Real position Counter Reading.....	58
7.3 Current Drive Speed Reading.....	58
7.4 Current Acceleration / Deceleration Reading.....	58
8. Driving Commands.....	59
8.1 +Direction Fixed Pulse Driving.....	59
8.2 -Direction Fixed Pulse Driving.....	59
8.3 +Direction Continuous Driving.....	59
8.4 -Direction Continuous Driving.....	60
8.5 Drive Status Holding.....	60
8.6 Drive Status Holding Release / Finishing Status Clear.....	60
8.7 Decelerating Stop.....	60
8.8 Sudden Stop.....	60
9. Interpolation Commands.....	61
9.1 2-Axis Linear Interpolation.....	61
9.2 3-Axis Linear Interpolation.....	61
9.3 CW Circular Interpolation.....	61
9.4 CCW Circular Interpolation.....	61
9.5 2-Axis Bit Pattern Interpolation.....	61
9.6 3-Axis Bit Pattern Interpolation Drive.....	62
9.7 BP Register Data Writing Enabling.....	62
9.8 BP Register Data Writing Disabling.....	62
9.9 BP Data Stack.....	62
9.10 BP Data Clear.....	63
9.11 Single Step Interpolation.....	63
9.12 Deceleration Enabling.....	63
9.13 Deceleration Disabling.....	63
9.14 Interpolation Interrupt Clear.....	63
10. Connection Examples.....	64
10.1 Connection Example for 68000 CPU.....	64
10.2 Connection Example for Z80 CPU.....	64
10.3 Connection Example.....	65
10.4 Pulse Output Interface.....	65
10.5 Connection Example for Input Signals.....	66
10.6 Connection Example for Encoder.....	66
11. Example Program.....	67
12. Electrical Characteristics.....	70
12.1 DC Characteristics.....	70
12.2 AC Characteristic.....	70
12.2.1 Clock.....	70
12.2.2 Read / Write Cycle.....	71
12.2.3 BUSYN Signal.....	71
12.2.4 SCLK/Output Signal Timing.....	71
12.2.5 Input Pulses.....	72
12.2.6 General Purpose Input / Output Signals.....	72
13. Timing of Input / Output Signals.....	73
13.1 Power-On Reset.....	73
13.2 Fixed Pulse or Continuous Driving.....	73

13.3 Interpolation.....	74
13.4 Start Driving after Hold Command.....	74
13.5 Sudden Stop	74
13.6 Decelerating Stop	74
14. Pinout.....	76
15. Specifications	77
Appendix A: Speed Curve Profile.....	1

1. OUTLINE

MCX314 is a 4-axis motion control IC which can control 4 axes of either stepper motor or pulse type servo drivers for position, speed, and interpolation controls. All of the MCX314's functions 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 4 Axes**
Each of the four 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. The accuracy of the frequency of the pulse output is less than +/- 0.1% (at CLK=16MHz). The speed of driving pulse output can be freely changed during the driving.
- **S-curve Acceleration and Deceleration**
Each axis can be preset with S-curve or trapezoidal acceleration/deceleration individually. Using S-curve command will drive the output pulse in a parabolic shaped acceleration and deceleration. Besides these, MCX314 has a special method to prevent from the happening of triangular curve when S-curve is commanded.
- **Linear Interpolation**
Any 2 or 3 axes can be selected to perform linear interpolation. The position boundary is between coordinates -8,388,608 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**
Any 2 axes can be selected to perform circular interpolation. The position boundary is between coordinates -8,388,608 and +8,388,607, and the positioning error is within +/- 1.0 LSB. The interpolation speed range is from 1 PPS to 4 MPPS.
- **Bit Pattern Interpolation**
Any 2 or 3 axes can be selected to perform the bit pattern interpolation, and the interpolation data is calculated by CPU; CPU writes the bit data into MCX314. Then, MCX314 outputs pulses continuously at the preset driving speed. So, the user can process any interpolation curve by this mode.
- **Continuous Interpolation**
Different interpolation methods can be used continuously, linear interpolation -> circular interpolation -> linear interpolation .. The maximum driving speed of performing continuous interpolation is 2 MHz.
- **Constant Vector Speed Control**
This function performs a constant vector speed. During the interpolation driving, MCX314 can set a 1.414 times pulse cycle for 2-axis simultaneous pulse output, and a 1.732-time pulse cycle for 3-axis simultaneous pulse output.
- **Position Control**
Each axis has a 32-bit logic position counter and a 32-bit real position counter. The logic position counter counts the output pulse numbers, and the real position counter counts the feedback pulse numbers 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.
- **Driving by External Signal**
It is possible to control each axis by external signals. The +/- direction fixed pulse driving and continuous driving can be also performed through the external signals. This function is used for JOG or teaching modes, and will share the CPU load.
- **Input/ Output Signal**
Each axis has 4 points of input signals to perform deceleration and stop in driving. These input signals are for high-speed near-by home search, home search and z-phase search during the home returning. Each axis is with 8 output points for general output.

■ Servo Motor Feedback Signals

Each axis includes input pins for servo feedback signals such as in-positioning, close loop positioning control and servo alarm.

■ Interrupt Signals

Interrupt signals can be generated when: (1). the start / finish of a constant speed drive during the trapezoidal 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

MCX314 can be connected to either 8-bit or 16-bit CPU

www.DataSheet4U.com

Figure 1.1 is the IC functional block diagram. It consists of same functioned X, Y, Z and U axes control sections and interpolation counting sections. Figure 1.2 is the functional block diagram of each axis control section.

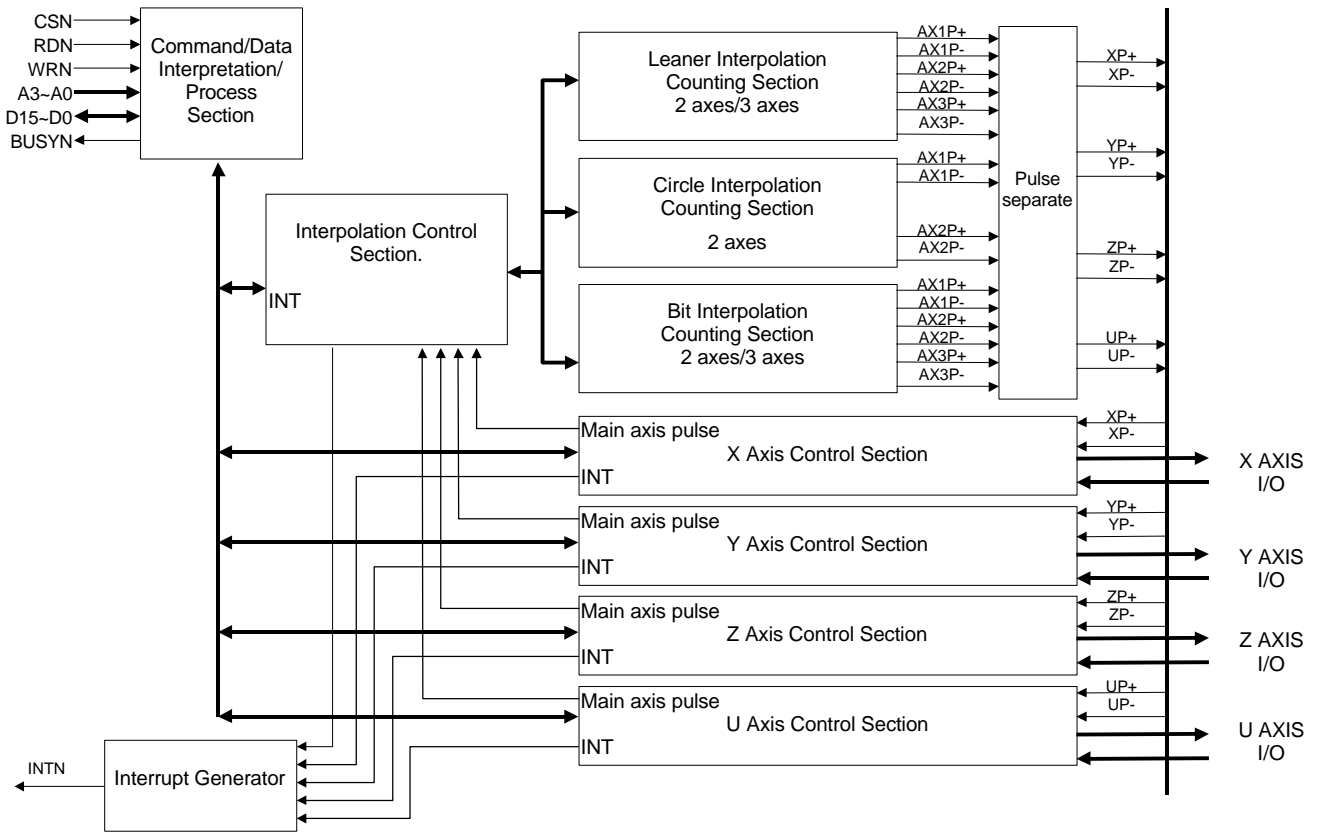
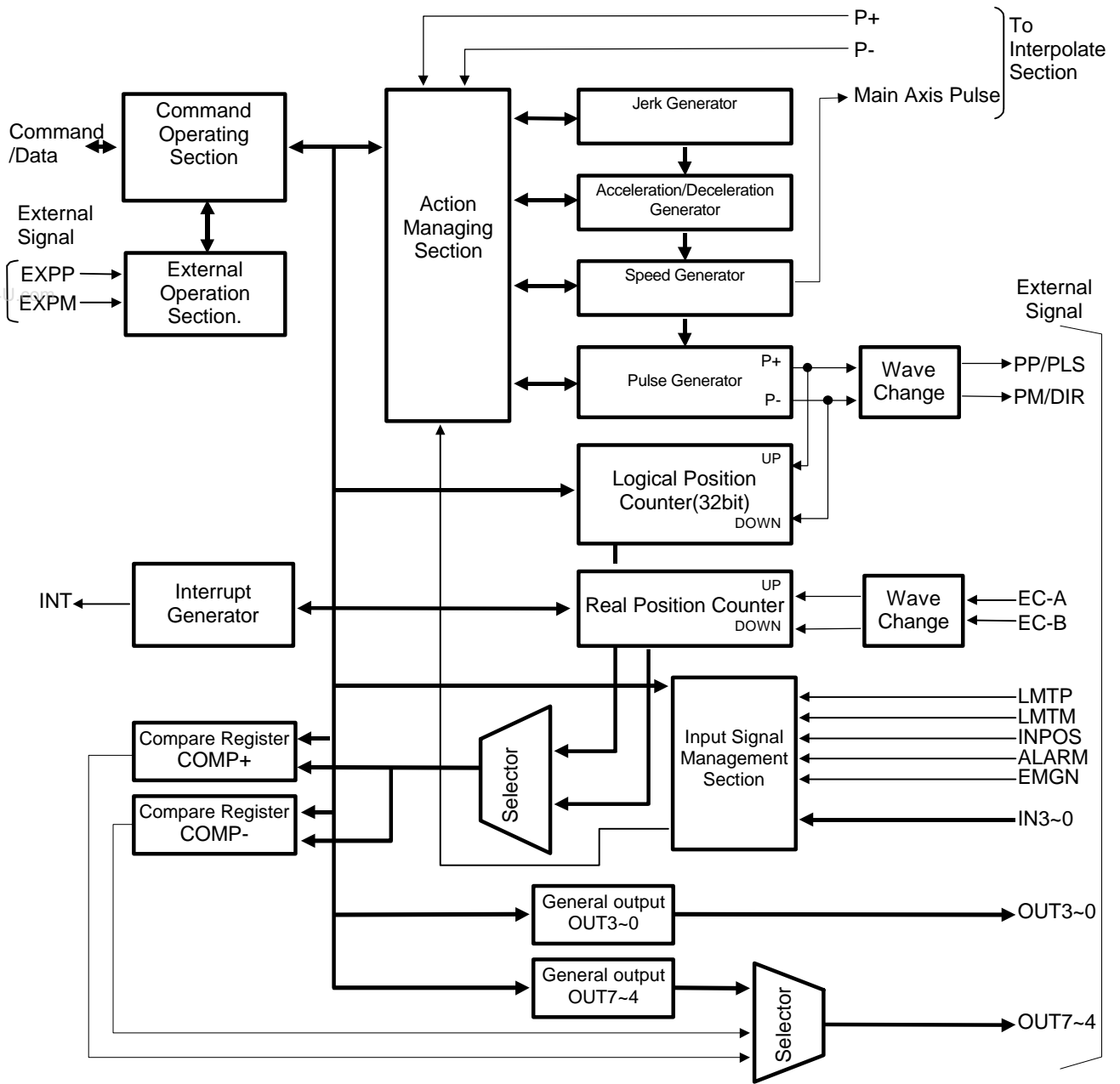


Figure 1.1 MCX314 Functional Block Diagram



Note 1: EMGN is for all axes use.

Fig.1.2 Control Block Diagram

2. The Descriptions of Functions

2.1 Pulse Output Command

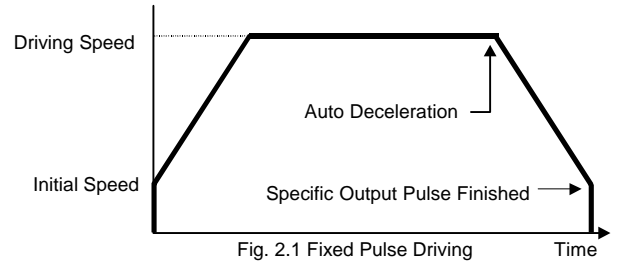
There are two kinds of pulse output commands: fixed pulse driving output and continuous pulse driving output.

2.1.1 Fixed Pulse Driving Output

When host CPU writes a pulse numbers into MCX314 for fixed pulse driving, and configures the performance such as acceleration / deceleration, speed. MCX314 will generate the pulses and output them automatically. When output pulse numbers are equal to the command pulse numbers, MCX314 stops the output. The profile is showing in Fig.2-1.

Concerning the execution of fixed pulse driving in acceleration / deceleration, it is necessary to set the following parameters:

■ Range	R
■ Acceleration/Deceleration	A/D
■ Initial Speed	SV
■ Driving Speed	V
■ Output Pulse Numbers	P



■ Changing Output Pulse Numbers in Driving

The output pulse numbers can be changed in the fixed pulse 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.

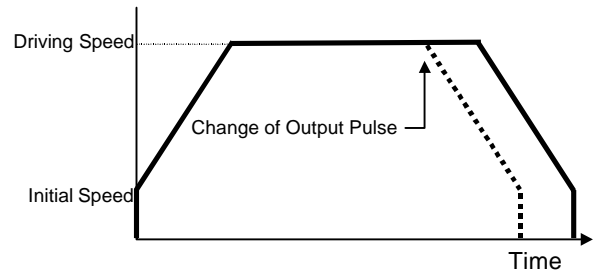


Fig. 2.2 Changing The Output Pulse Numbers in Driving

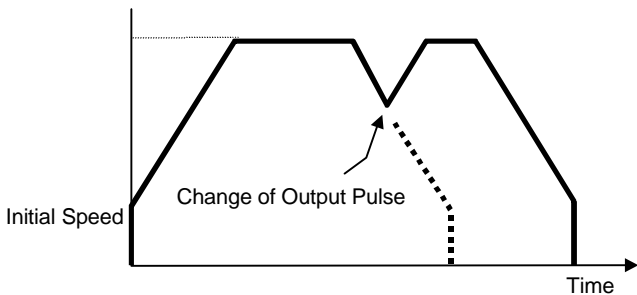


Fig. 2.3 Changing Command During Deceleration

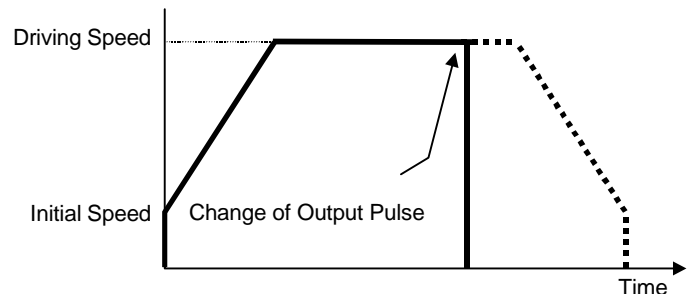


Fig.2.4 Changing The Lesser Pulse Numbers Than Output Pulse Stop

■ Manual Setting Deceleration for fixed pulse Acceleration/Deceleration Driving

As shown in Fig. 2.1, generally the deceleration of fixed pulse acceleration /deceleration driving is controlled automatically by MCX314. 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 pulse 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 pulse driving.

■ Offset Setting for Acceleration/Deceleration Driving

The offset function can be used for compensating the pulses when the decelerating speed doesn't reach the setting initial speed during the S-curve fixed pulse driving. MCX314 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 start the deceleration.

When setting the offset for deceleration, MCX314 will start deceleration early for the offset. The remaining pulses of offset will be driving output at the initial speed (see Fig. 2.5).

The default value for offset is 8 when MCX314 power-on reset. It is not necessary to change the shift pulse value in the case of acceleration/deceleration fixed pulse driving.

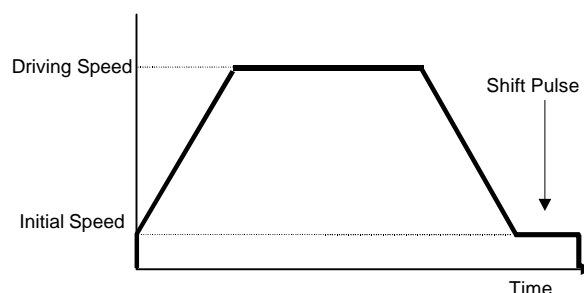


Fig.2.5 Remaining Pulse in Acc. / Dec. Fixed Pulse

2.1.2 Continuous Pulse Driving Output

When the continuous driving is performed, MCX314 will drive pulse output in a specific speed until stop command or external stop signal is happened.

The main application of continuous driving is: home searching, teaching or speed control.

Two stop commands are for stopping the continuous driving. One is "decelerating stop", and the other is "sudden stop".

Four input pins, IN3~IN0, of each axis can be connected for external decelerating and sudden stop signals. Enable / disable, active levels and mode setting are possible.

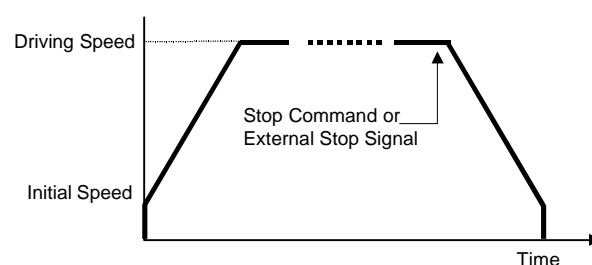


Fig.2.6 Continuous Driving

■ Stop Condition for External Input IN3~IN0 in Continuous Driving

The input pins IN3~IN0 can be used for home searching, near-by home searching and Z-phase searching. Enable / disable and logical levels can be set at WR1 of each axis.

For the application of high-speed searching, the user can set MCX314 in the acceleration/deceleration continuous driving mode and enable IN1 in WR1. And then, MCX314 will perform the decelerating stop when the external signal IN1 is active.

For the application of low-speed searching, the user can set MCX314 in the constant-speed continuous driving and enable IN1. Then, MCX314 will perform the sudden stop when IN1 is active.

2.2 Speed Curve

The speed driving profile in MCX314 can be configured as constant speed driving, trapezoidal driving or S-curve acceleration/deceleration.

2.2.1 Constant Speed Driving

When the driving speed set in MCX314 is lower than the initial speed, 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.

- Range R
- Initial Speed SV
- Drive Speed V
- Output Pulse Numbers P (Only applicable for the fixed pulse driving)

The items should be preset in accordance with the requirement.

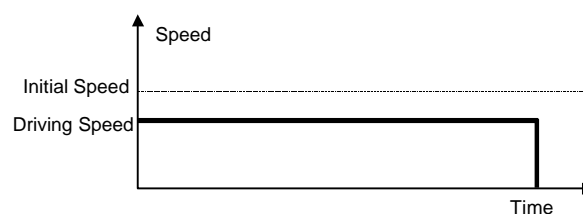


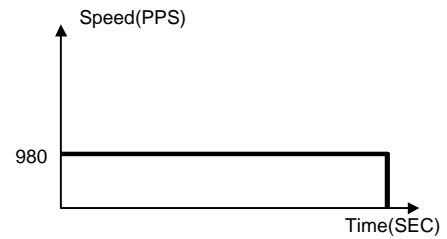
Fig.2.7 Constant Speed Driving

■ 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(M) = 1
- Initial Speed SV=980: Initial Speed ≥ Driving Speed
- Driving Speed V=980: Should be less than initial speed

Please refer each parameter in Chapter 6.



2.2.2 Trapezoidal Driving

Trapezoidal driving is starting from the initial speed to the designated drive speed. The accelerating pulses will be counted, and the deceleration (automatic deceleration) starts from the drive speed to initial speed once the remaining pulse numbers are less than the accelerating pulse numbers.

When the decelerating stop command is performed during the acceleration, or when the pulse numbers of the fixed pulse drive do not reach the designated drive speed, the driving will be decelerating during acceleration, as show in Fig. 2.8. (triangle driving profile prevention, see appendix A3)

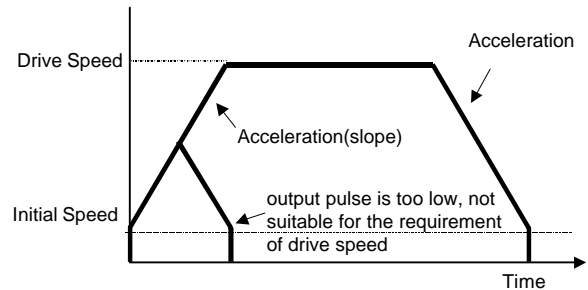


Fig.2.8 Trapezoidal Driving

Usually, the user should set the same acceleration and deceleration rates. For some cases, the acceleration and deceleration can be set individually by setting the D1 of WR3 to 1. When the deceleration is set individually in fixed pulse driving, the automatic deceleration will not be performed, but the manual deceleration is required. The user should set the bit D1 of Register WR3 as 1, then use decelerating command (03h) to set the deceleration.

When performing the trapezoidal driving, the following parameters should be preset.

- Range R
- Acceleration A : Acceleration and deceleration
- Deceleration D : Option for individual deceleration.
- Initial Speed SV
- Driving Speed V
- Output Pulse Number P : Only for fixed pulse driving

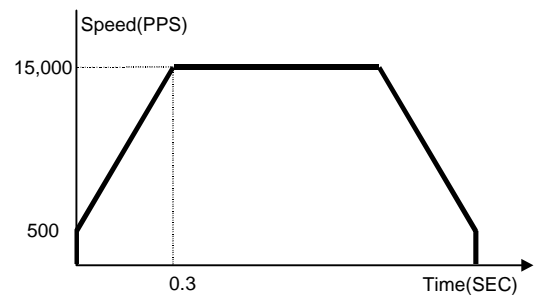
The □ items should be preset in accordance with the requirement.

■ 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(M)= 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.



2.2.3 S-curve Acceleration/Deceleration Driving

In case of S-curve acceleration / deceleration driving, the acceleration profile is not linear. The value of acceleration / deceleration is shaped as the trapezoid; see Fig. 2.9.

In acceleration, there are three regions with different acceleration values. At the beginning, the acceleration increase linearly from 0 to the specific value A with a specific rate of acceleration K, which shows the driving speed increase parabolically in this region.

Then, the driving speed increases in a constant acceleration in region b. And, in section c, the acceleration decelerates linearly to 0 with the rate of deceleration K. So the acceleration of S-curve includes regions a, b and c.

In deceleration, as same as acceleration, the driving speed decelerate parabolically in three regions d, e and f.

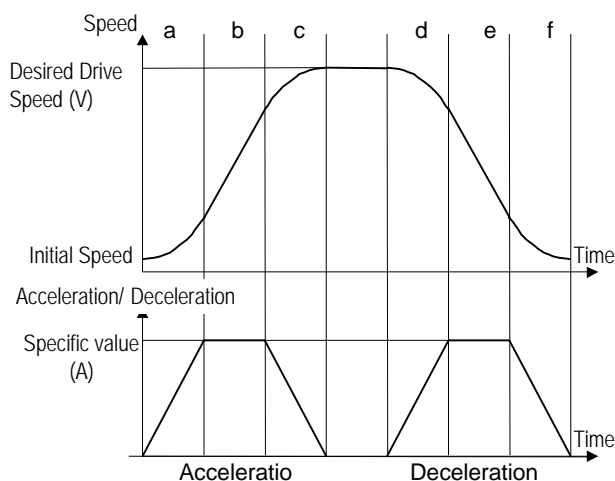


Fig.2.9 S-Curve Acceleration/Deceleration Driving

■Complete S-curve and Partial S-curve

The desired driving speed is V. When V (speed in region a) ≤ speed in region a, the region b will disappear. This condition is called **complete S-curve**. Otherwise, it is called **partial S-curve**. Please check the parameters and examples in Appendix A.

In order to execute S-curve acceleration / deceleration, the user has to set bit D2 of register WR3 to 1, and the following parameters are necessary to be set.

- Range R
- Jerk K
- Acceleration A : The designated value of acceleration and deceleration
- Deceleration D : The designated deceleration value of individual setting
- Initial Speed SV
- Drive Speed V
- Output Pulse Number P : Used for fixed pulse driving

■The Prevention of Triangle Driving Profile

When the fixed pulse trapezoidal driving is performed, and also when the deceleration is performed before the acceleration stops, the triangle driving profile is coming out. The prevention of triangle driving profile in S-curve acceleration / deceleration driving will be discussed as follows.

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.

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

Therefore, the total output pulse number $p(t)$ 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

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

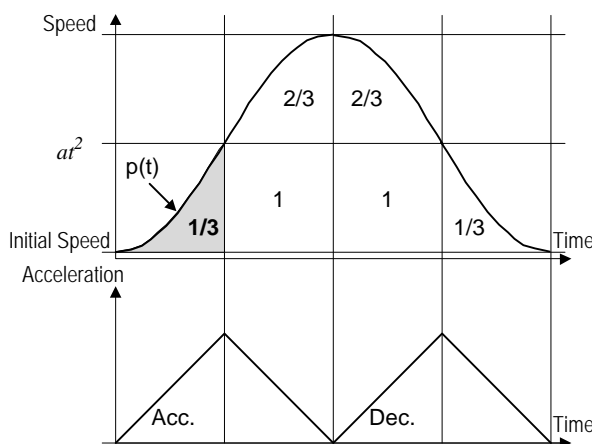


Fig. 2.10 The rule of 1/12 of Parabolic Acceleration/Deceleration

Therefore, when the output pulse in acceleration of S-curve is more than 1/12 of total output pulse, MCX314 will stop increasing acceleration and start to decrease the acceleration value.

■The Decelerating Stop for Preventing the Triangle Driving Profile in S-curve Driving

When the decelerating stop is commanded, or when the external signals IN3~IN0 are active during the S-curve acceleration / deceleration driving, the acceleration rate is decreasing, then the deceleration starts when the acceleration rate reaches 0.

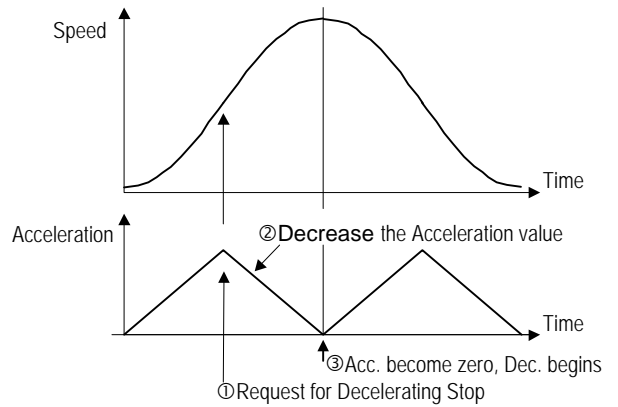


Fig. 2.11 Decelerating Stop During S-curve Acc./ Dec. Driving

■Constraints for S-curve Acceleration / Deceleration Driving

- The drive speed cannot be changed during the fixed pulse S-curve acceleration / deceleration driving.
- When the fixed-pulse S-curve acceleration / deceleration driving is performed, the change of the output pulse numbers during the deceleration will not result a normal S-curve driving profile.
- In case of executing circular interpolation, bit pattern interpolation and continuous interpolation, S-curve acceleration/deceleration cannot be executed normally.
- If the S-curve output pulses run out before the deceleration to the initial speed, the user can use offset function.
- When the S-curve output pulses are decelerating to the initial speed, but still some pulses remains, the user can modify the parameter K and driving speed V to avoid this situation.

■Example of Setting Parameters (Complete S-curve Acceleration /Deceleration)

*Setting a complete S-curve acceleration to output the drive speed from 0 to 40K PPS in 0.4 sec.

As shown in Fig. Ex.3, if complete S-curve acceleration is required, the driving must be 20K PPS at time 0.2sec. And then, it will accelerate to 40K PPS at time 0.4sec.

To calculate the maximum acceleration value acc. at time 0.2sec, we know the speed at time 0.2sec is 20K PPS. So we get

$$20000PPS = 0.2sec \times acc. /2$$

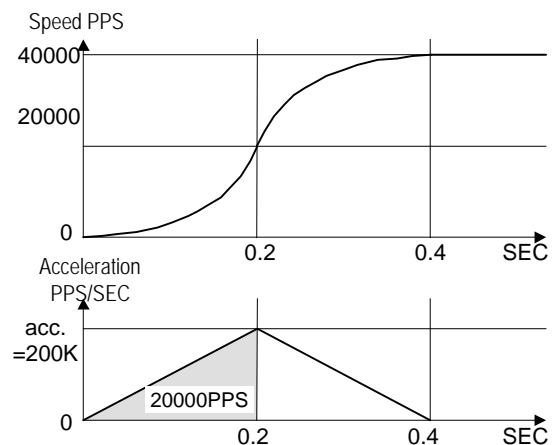
so,

$$acc. = 200K PPS/sec$$

To calculate the jerk, we get

$$200KPPS/0.2sec=1000KPPS/sec^2$$

In a complete S-curve acceleration/deceleration, the speed curve is depended on the jerk. Since the acceleration / deceleration does not exit in the partial S-curve, it should be preset over than 200KPPS/SEC.



- Range R = 800,000 ; multiple(M) = 10
- Jerk K = 625 ; ((62.5 x 10⁶)/625) x 10 = 1,000KPPS/SEC²
- Acceleration A = 160 ; 125 x 160 x 10 = 200KPPS
- Initial Speed SV = 100 ; 100 x 10 = 1000KPPS
- Drive Speed V = 4000 ; 4000x 10 = 40,000PPS

Please refer each parameter in Chapter 6.

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

*Setting a partial S-curve acceleration to output the drive speed from 0 to 40K PPS in 0.6 sec.

At first, a parabolic accelerating is executed to 10KPPS in 0.2 sec., then the linear acceleration goes up to 30KPPS, and then the parabolic acceleration reaches to 40KPPS.

We get

$$10K\text{ PPS} = \text{Acceleration} \times 0.2\text{sec} / 2$$

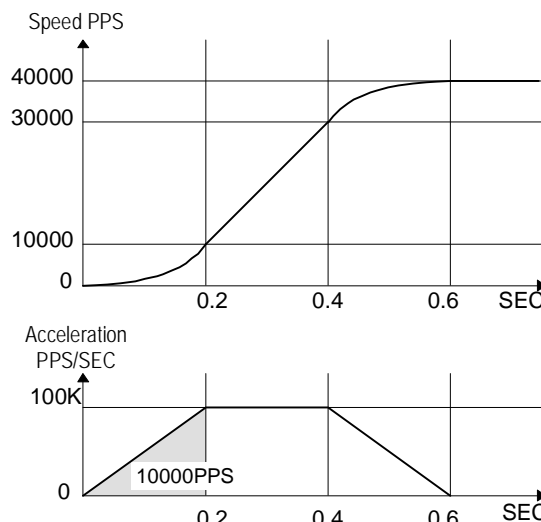
$$\text{Acceleration} = 100K\text{ PPS/sec}$$

And, the jerk is

$$100K\text{PPS/sec}/0.2\text{sec}=500K\text{PPS/sec}^2$$

- Range R = 800,000 : multiple = 10
- Jerk K = 1250 : $((62.5 \times 10^6)/1250) \times 10 = 500 \times 10^3 \text{ PPS/SEC}^2$
- Acceleration A = 80 : $125 \times 80 \times 10 = 100 \times 10^3 \text{ PPS/SEC}$
- Initial Speed SV = 100 : $100 \times 10 = 1000 \text{ PPS}$
- Drive Speed V = 4,000 : $4,000 \times 10 = 40,000 \text{ PPS}$

Please refer each parameter in Chapter 6.



2.2.4 Pulse Width and Speed Accuracy

■Duty Ratio of Driving Pulse

The period time of +/- direction pulse driving of each axis is decided by system clock CLK. The tolerance is within ±1SCLK (For CLK=16MHz, the tolerance is ±125nSEC).

Basically, the duty ratio of each pulse is 50% as show in Fig. 2.12. 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 50uSEC on its Low level and the period is 1mSEC.

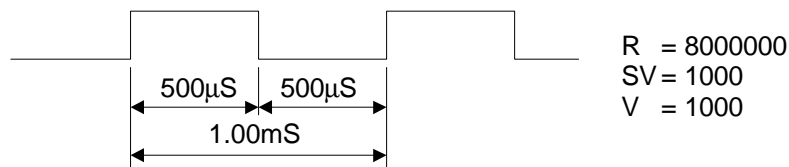


Fig. 2.12 Output of Drive Pulse (1000PPS)

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. 2.13.

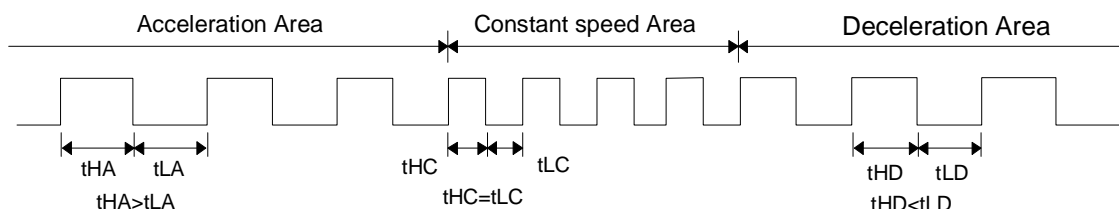


Fig. 2.13 Comparison of Drive Pulse Length in Acceleration / Deceleration

■The Accuracy of Drive Speed

The clock(SCLK) running in MCX14 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 a exact multiple of SCLK period(125nSEC). Otherwise, the driving pulse will not very stable. The following table shows the frequency (speed) of driving pulse of MCX314 can be, there are all exact the multiple of 125nSEC.

Multiple	Drive Speed (PPS)	Multiple	Drive Speed (PPS)	Multiple	Drive Speed (PPS)	Multiple	Drive Speed (PPS)
		11	727 K	95	84,211	995	8040
2	4.000 M	12	667 K	96	83,333	996	8032
3	2.667 M	13	615 K	97	82,474	997	8024
4	2.000 M	14	571 K	98	81,632	998	8016
5	1.600 M	15	533 K	99	80,808	999	8008
6	1.333 M	16	500 K	100	80,000	1000	8000
7	1.143 M	17	471 K	101	79,208	1001	7992
8	1.000 M	18	444 K	102	78,431	1002	7984
9	889 K	19	421 K	103	77,670	1003	7976
10	880K	20	400 K	104	76,923	1004	7968

As shown in the table above, it is not very stable to set any desired driving speed. However, MCX314 can make any drive speed in using the following method.

When the preset range value: R= 80,000 (Multiple=100), drive speed setting value: V=4900, the output pulse is set $4900 \times 100=490KPPS$. From the able above, the output 490KPPS cannot be exactly made because the period of 490KPPS is 16.326 times of SCLK (125nSEC).

Therefore, as shown in Figure 2.14, MCX314 combines 16 times and 17 times of SCLK period in a rate of 674:326 to generate an average 490KPPS.

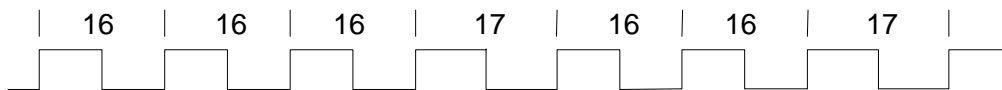


Fig. 2.14 The driving pulse of 490KPPS when SCLK is 8MHz

According to this method, MCX314 can generate a constant speed driving pulse in a very high accuracy. In general, the higher of the driving speed, the lower of the accuracy. But for MCX314, it still can maintain relative accuracy when the driving 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 tolerance about 1SCLK(125nSEC). This is no matter when putting the driving to a motor because the tolerance will be absorbed by the inertia of motor system.

2.3 Position Control

Fig 2.15 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.

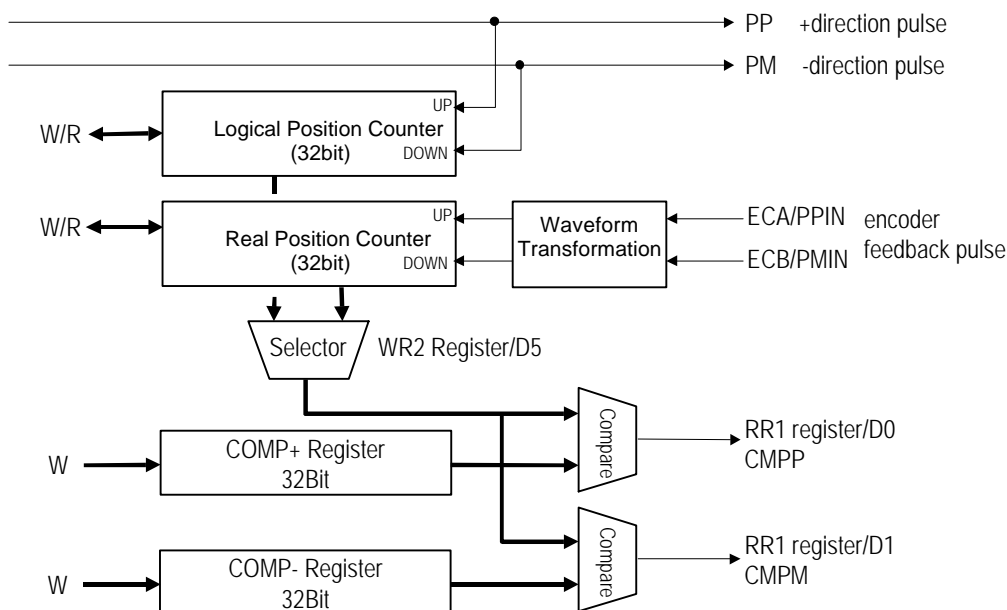


Fig. 2.15 Position Control Block Diagram

2.3.1 Logic Position Counter and Real position Counter

The logic position counter is counting the driving pulses in MCX314. 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, 2.15, 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 register WR2. 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. 2.16 is an example for COMP+ = 10000, COMP- = -10000.

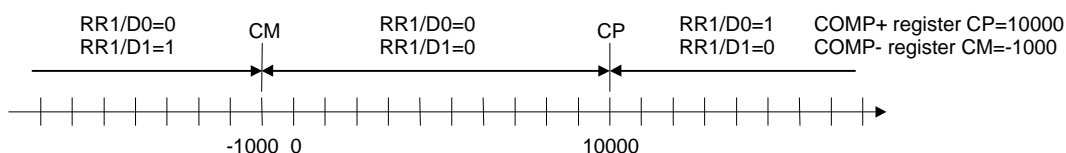


Fig. 2.16 Example of COMP+/- Register Setting

COMP+ and COMP- registers can be used as software +/- limit. When D0 and D1 bits of register WR2 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 register RR2 will change to 1. If the value of logical / actual counter is smaller than that of COMP+, the D0 bit of register RR2 will change to 0 automatically. Host CPU can write the COMP+ and COMP- registers any time. However, when MCX314 is reset, the register values are random.

2.4 Interpolation

This 4-axis motion control IC can perform any 2 / 3 axes linear interpolation, any 2 axes circular interpolation and any 2 / 3 axes bit pattern interpolation. Bits D0, D1 (ax 1), D2, D3 (ax 2) and D4, D5 (ax 3) of register WR5 can be pointed for performing the interpolation. In the process of interpolation driving, all the calculations will follow the main axis (ax1). So, the user has to set the parameters such as initial speed and drive speed of the main axis before performing the interpolation. During the linear interpolation, it is not necessary to set the main axis as “long axis”.

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 n-DRV of the interpolating axis 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.

i 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 of the designated interpolating axis. PR0 will become 1, and RR2 (error register) of this axis will be read out. Note: In case of circular or bit pattern interpolation, the “active” of hardware or software limit, in either + or - direction, will stop the interpolation driving.

j In-position Signal for Servo Motor

During the interpolation driving, when the in-position signal (nINPOS) of each driving 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

Any 2 or 3 axes of the 4 axes can be set for linear interpolation. To execute the linear interpolation, the user can, according to the present point coordinates, set the finish point coordinates and the interpolation command(s) for 2 or 3 axes.

As shows in Fig. 2.17 the proceeding for linear interpolation is performing from the start point to the finish point.

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. 2.17.

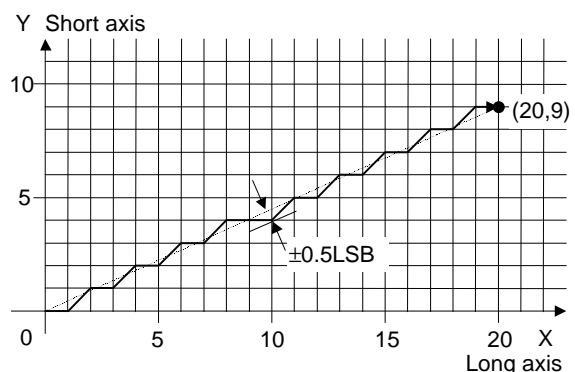


Fig. 2.17 The Position Accuracy for Linear Interpolation

As shown in Fig. 2.18, 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.

The range for each axis is a 24-bit signed counter, from -8,388,607 ~ +8,386,807.

Note: The user cannot set -8,388,608.

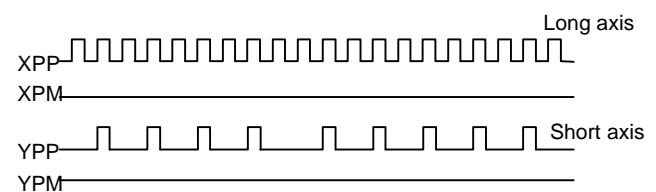


Fig. 2.18 The Example for Pulse Output at Finish Point (X=20, Y=9)

i 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: +300, Y: -200). The interpolation drive speed is constant: 1000PPS.

```

WR5 ← 0004h write ; map ax1 to X axis, ax2 to Y
axis

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

WR6 ← 03E8h write ; initial speed :1,000PPS
WR0 ← 0104h write

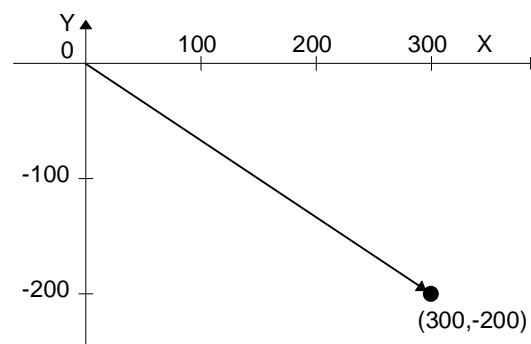
WR6 ← 03E8h write ; drive speed: 1,000PPS
WR0 ← 0105h write

WR6 ← 012Ch write ; finish point of X axis: 300
WR7 ← 0000h write
WR0 ← 0106h write

WR6 ← FF38h write ; finish point of Y axis: -200
WR7 ← 0000h write
WR0 ← 0206h write

WR0 ← 0030h write ; linear interpolation driving for 2 axes enabling

```



i The example of linear interpolation for 3 axes

Executing linear interpolation drive for X, Y and Z axes from the current position to the finish position (X: 15,000, Y: 16,000, Z: 20,000). The initial speed = 500PPS, acceleration / deceleration = 40,000PPS/SEC, drive speed = 5,000PPS.

```

WR5 ← 0024h write ; define: ax1=X axis, ax2=Y axis, ax3= Z axis

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

WR6 ← 0140h write ; accel./decel. speed: 40,000/SEC
WR0 ← 0102h write ; 40,000 / 125 / 1 = 320=140h

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

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

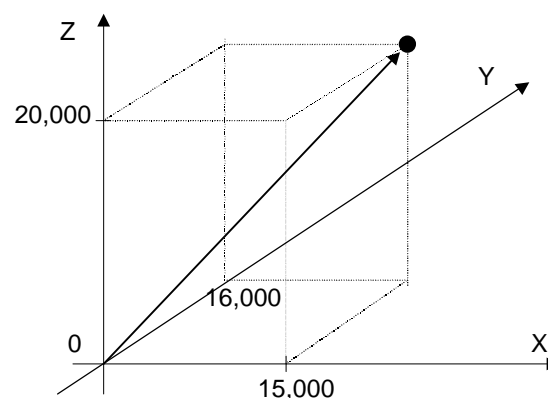
WR6 ← 3A98h write ; finish point of X axis:15,000
WR7 ← 0000h write
WR0 ← 0106h write

WR6 ← 3E80h write ; finish point of Y axis:; -16,000
WR7 ← 0000h write
WR0 ← 0206h write

WR6 ← 4E20h write ; finish point of Z axis; 20,000
WR7 ← 0000h write
WR0 ← 0406h write

WR0 ← 003Bh write ; deceleration enabling
WR0 ← 0031h write ; linear interpolation driving for 3 axes enabling

```



2.4.2 Circular Interpolation

Any 2 axes of the 4 axes can be selected 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. 2.19, 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.

In Fig. 2.20, 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 ax1 is always larger than that of ax2 in quadrants 0, 3, 4 and 7, so we call ax1 is the long axis (ax2 is the short axis) in these quadrants; in quadrants 1, 2, 5 and 6, ax2 is the long axis (ax1 is the short axis).

The short axis will output pulses regularly, and the long axis will output pulses depending on the interpolation calculation.

In Fig. 2.21, it is an example to generate a circle with the center point (-11,0) and the finish point (0,0). Its radius is 11. In Fig. 2.22 shows the pulse output.

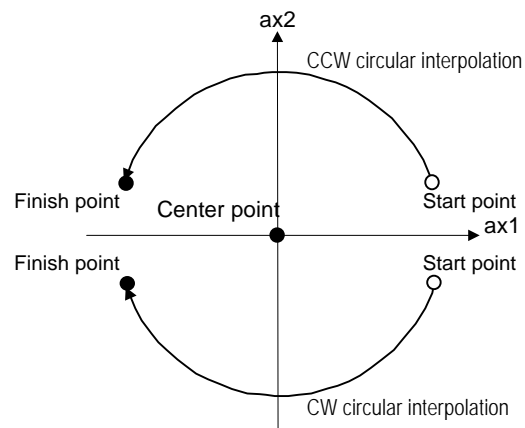


Fig. 2.19 CW/CCW Circular Interpolation

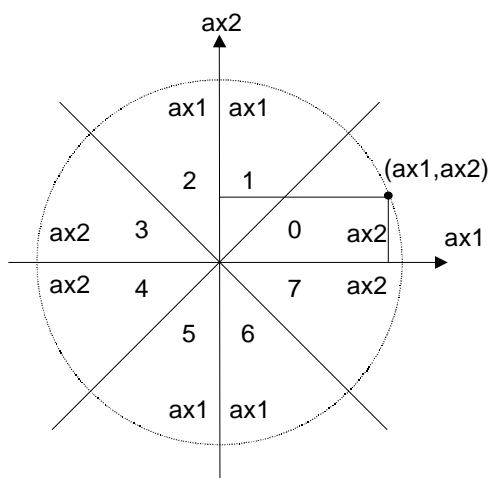


Fig. 2.20 The 0~7 Quadrants And Short Axes in Circular Interpolation Calculation

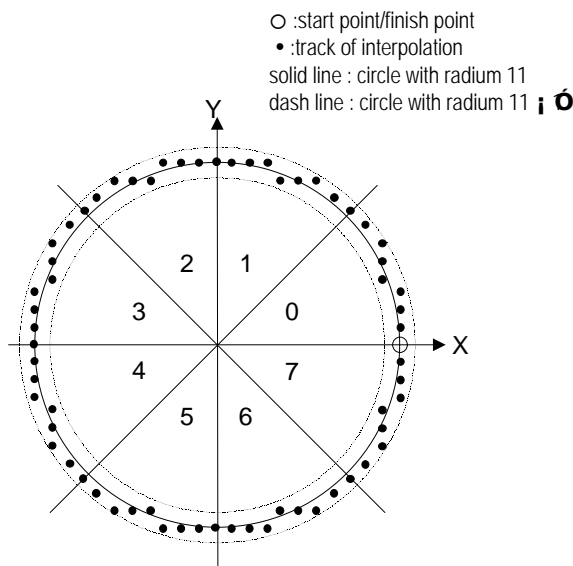


Fig. 2.21 The Example of Circular Interpolation

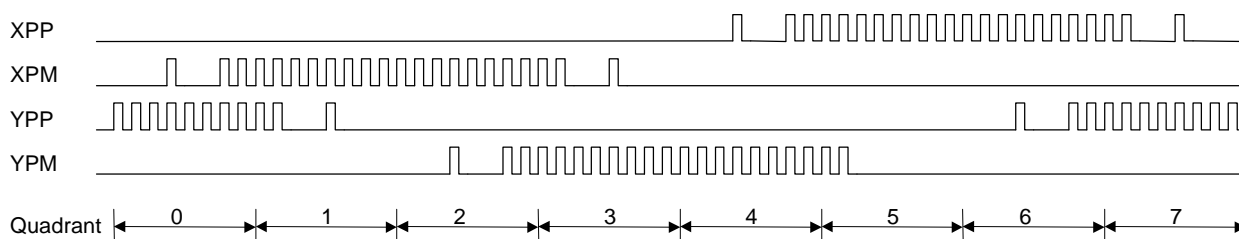


Fig. 2.22 The Example of Pulse Output in Circular Interpolation Driving

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 on the circular track. The IC will make an in-position checking by the short axis. If the value of finish point is as same as that of short axis, this circular interpolation is finished.

Fig. 2.23 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 ax2 is the short axis in quadrant 4. So the interpolation is finished when the ax2 is 299.

The range of interpolation coordinate is from the start point to -8,388,608 ~ +8,388,607, and the interpolation error for all the range is within ± 1 LSB. The speed of interpolation driving is 1PPS ~ 4MPPS.

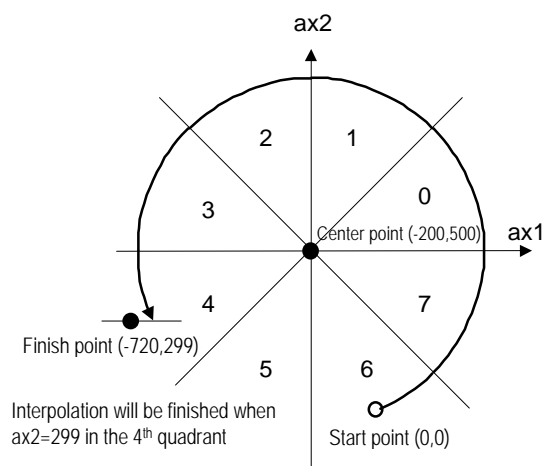


Fig. 2.23 Example of In-position Check in Circular Interpolation

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 ← 0104h write ; define: ax1:X axis, ax2:Y axis, and with 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 x 1.414 = 5,656,000
WR0 ← 0200h write

WR0 ← 01F4h write ; initial speed : 500 x 2 = 1000PPS
WR0 ← 0104h write

WR6 ← 01F4h write ; drive speed : 500 x 2 = 1000PPS
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
    
```

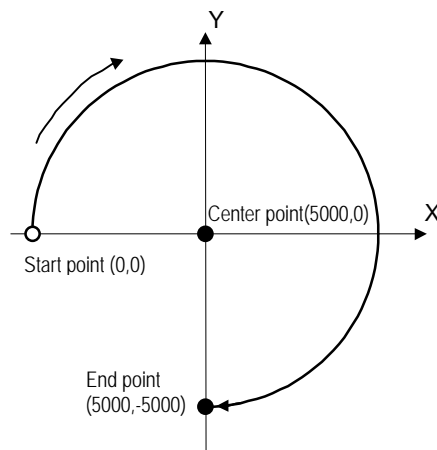


Fig. EX.2 CW Circular Interpolation in Constant Vector Speed

2.4.3 The Bit Pattern Interpolation

MCX314 is able to receive the interpolation data from the host CPU, and output pulses at a specific speed. The host CPU executes the interpolation for 2 or 3 axes, generates a set of pulse data, then writes the commands into MCX314. MCX314 will output the pulses at a specific 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 2 or 3 axes, into MCX314.

If a bit in the bit pattern data from CPU is "1", MCX314 will output a pulse at the time unit; if it is "0", MCX314 will not output any pulse at the time unit.

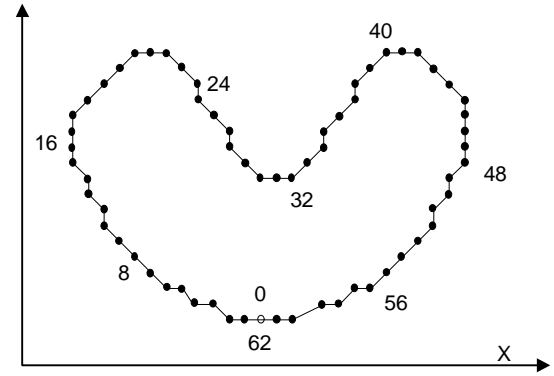


Fig. 2.24 Example for Bit Pattern Interpolation

For example, if the user want to generate the X-Y profile (see Fig. 2.24), 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, MCX314 will check the registers once and decide to output a pulse or not depending on the bit pattern.

←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. 2.25 is the block diagram of bit pattern interpolation for the 1st axis in MCX314.

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.

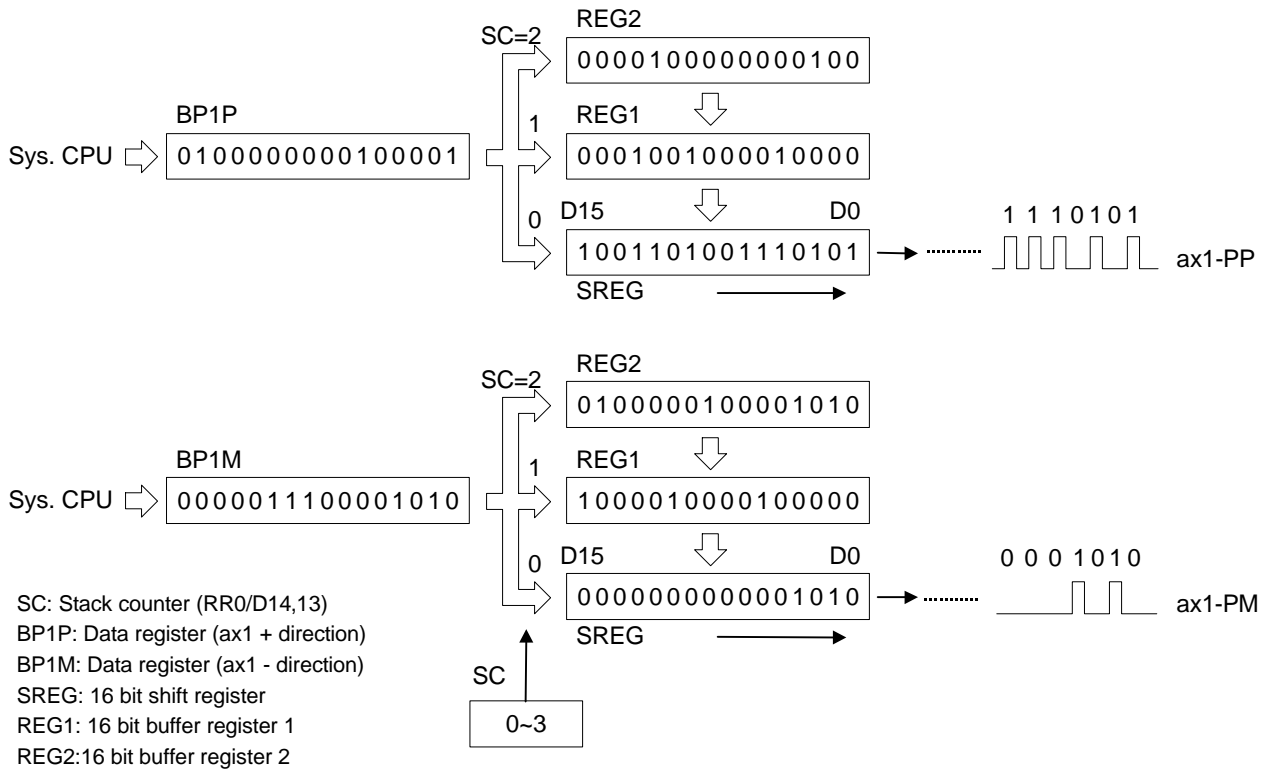


Fig. 2.25 The Structure of Register for The Bit Pattern Interpolation (for 1 axis)

Stacking counter (SC) is a 2-bit counter. Its value is between 0 and 3, which can be read from D14,13 of register RR0. 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 MCX314 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 MCX314 again.

In order to make MCX314 output the bit pattern data continuously, the host CPU should write the data into MCX314 before SC counts down to 0. MCX314 will output a interrupt requirement signal to host CPU when SC counts down from 2 to 1.

j The limitation for the speed of bit pattern interpolation driving

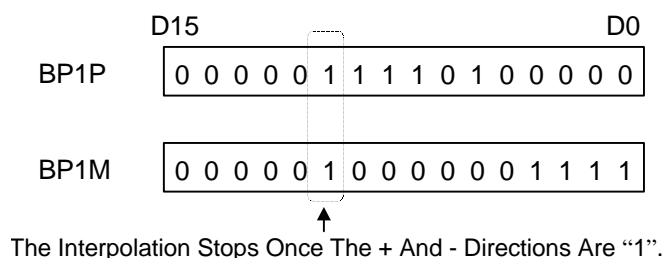
The maximum pulse output speed of MCX314 is 4MHz in bit pattern interpolation mode. However, the maximum speed will depend on the data update rate of host CPU if the bit pattern data are more than 48bits. For example of the X and Y axes bit pattern interpolation, if the host CPU needs 100usec to update new 16-bit data for X and Y axes. The maximum speed is $16/100\mu\text{SEC}=160\text{KPPS}$.

j 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 ax1.

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



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

② The host CPU stops writing any command into MCX314.

When SC=0, and when no other data is updated, MCX314 will stop outputting pulse. Then, the bit pattern interpolation is finished.

j 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 the master axis (ax1) which is executing the bit pattern interpolation. MCX314 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 MCX314 must be cleared in using BP register (3Dh).

j 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 MCX314 must be cleared.

i Writing the bit pattern data into the register in MCX314

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	ax1 +direction data	nWR2
0	1	1	BP1M	ax1 -direction data	nWR3
1	0	0	BP2P	ax2 +direction data	WR4
1	0	1	BP2M	ax2 -direction data	WR5
1	1	0	BP3P	ax3 +direction data	WR6
1	1	1	BP3M	ax3 -direction data	WR7

Note: BP3P and BP3M share the same registers: WR6 and 7.

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

Address				Name of register	Address				Name of register
A3	A2	A1	A0		A3	A2	A1	A0	
0	0	0	0		1	0	0	0	BP2PL
0	0	0	1		1	0	0	1	BP2PH
0	0	1	0		1	0	1	0	BP2ML
0	0	1	1		1	0	1	1	BP2MH
0	1	0	0	BP1PL	1	1	0	0	BP3PL
0	1	0	1	BP1PH	1	1	0	1	BP3PH
0	1	1	0	BP1ML	1	1	1	0	BP3ML
0	1	1	1	BP1MH	1	1	1	1	BP3MH

Note: BPmPL, BPmPH, BPmML, BPmMH represent the following bit groups (m is 1 ~ 3).

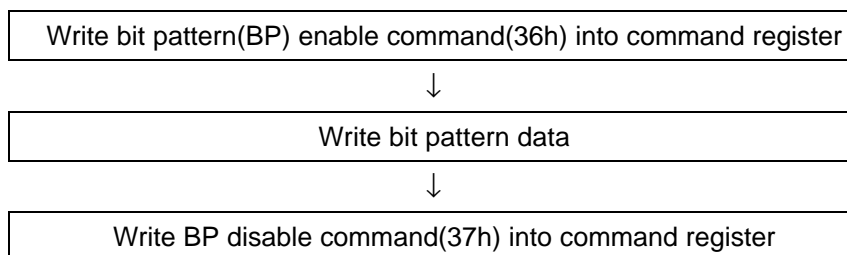
BPmPL : the low byte of BPmP (D7 ~ D0)

BPmPH : the high byte of BPmP (D15 ~ D8)

BPmML : the low byte of BPmM (D7 ~ D0)

BPmMH : the high byte of BPmM (D15 ~ D8)

For some addresses of bit pattern data registers are as same as nWR2 ~ nWR7, the host CPU can not write any data into the bit pattern data register since MCX314 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.

i The example of bit pattern interpolation

The bit interpolation example is shown in Fig. 2.24. We set X axis as ax1, Y axis as ax2 and a constant

speed: 1000PPS in a constant vector speed interpolation driving.

WR5 ← 0104h write ;Define ax1: X , ax2:Y
 WR6 ← 0900h write ;setting the master axis 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
 ; command of bit 0~15th
 BP1P ← 0000h write ;X axis +direction command
 BP1M ← 2BFFh write ;X axis -direction command
 BP2P ← FFD4h write ;Y axis +direction command
 BP2M ← 0000h write ;Y axis -direction command
 WR0 ← 0038h write ;BP data stacking, SC=1
 ; command of bit 16~31th
 BP1P ← F6FEh write ;X axis +direction command
 BP1M ← 0000h write ;X axis -direction command
 BP2P ← 000Fh write ;Y axis +direction command

BP2M ← 3FC0h write ;Y axis -direction command
 WR0 ← 0038h write ;BP data stacking, SC=2
 ; command of bit 32~47th
 BP1P ← 1FDBh write ;X axis +direction command
 BP1M ← 0000h write ;X axis -direction command
 BP2P ← 00FFh write ;Y axis +direction command
 BP2M ← FC00h write ;Y axis -direction command
 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
 ; command of bit 48~63th
 BP1P ← 4000h write ;X axis +direction command
 BP1M ← 7FF5h write ;X axis -direction command
 BP2P ← 0000h write ;Y axis +direction command
 BP2M ← 0AFFh write ;Y axis -direction command
 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

Note1

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

; Executing bit pattern interpolation by interrupt

During the bit pattern interpolation, MCX314 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 MCX314 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 MCX314. 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 MCX314 finishes the bit pattern interpolation.

2.4.4 Constant Vector Speed

MCX314 is with the constant vector speed control function which can control the resultant speed of two axes to keep the speed in constant.

Fig. 2.26 shows the profile of 2 axes interpolation driving. The vector speed reflects 1.414 times of the individual axis driving speed. So, we have to set the speed of 1.414 times to keep the vector speed for 2-axis driving.

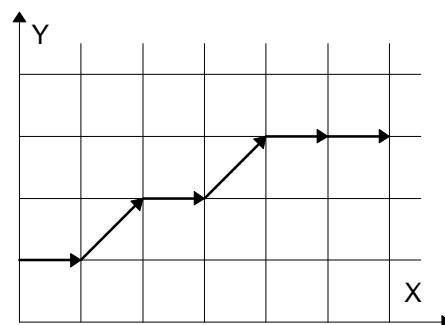


Fig. 2.26 Example of 2 axis interpolation

; 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 slave-axis (ax2) to be 1.414 times of the value of the master-axis (ax1). Therefore, MCX314 will use the range parameter of master-axis if only 1 axis outputs pulses. However, when 2 axes output pulses simultaneously, MCX314 will use the range parameter of slave-axis to implement the pulse period to 1.414 times.

; The Example of Constant Vector Speed Interpolation for 2 Axes

As shown below, the master-axis (ax1) = X axis, the slave-axis (ax2) = Y axis, and the interpolation is at a constant vector speed: 1000PPS. The result of driving pulse output is shown in Fig. 2.27.

WR5 ← 0104h	write	; define ax1: X axis, ax2:Y axis ; constant vector speed	WR0 ← 0105h	write	
WR6 ← 0900h	write	; setting parameter of the master axis	WR6 ← 03E8h	write	; finish point of X
WR7 ← 003Dh	write	; range: 4,000,000 (multiple=2)	WR7 ← 0000h	write	
WR0 ← 0100h	write		WR0 ← 0106h	write	
WR6 ← 4DC0h	write	; range of 2-axis constant vector speed	WR6 ← 0190h	write	; finish point of Y
WR7 ← 0056h	write	; 4,000,000x1.414=5,656,000	WR7 ← 0000h	write	
WR0 ← 0200h	write		WR0 ← 0206h	write	
WR6 ← 01F4h	write	; initial speed: 500x2=1000PPS	WR0 ← 0030h	write	; 2-axis linear interpolation starting
WR0 ← 0104h	write				
WR6 ← 01F4h	write	; drive speed: 500x2=1000PPS			

www.DataSheet4U.com

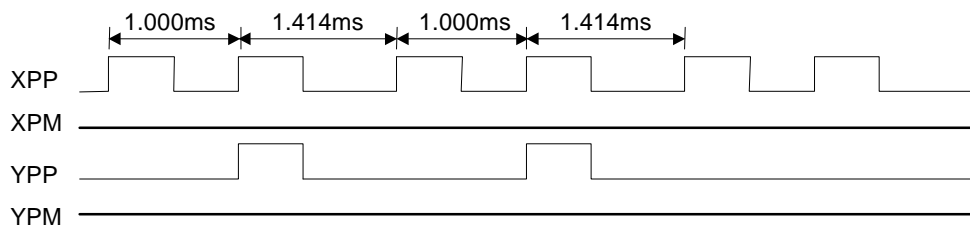


Fig. 2.27 The Example of 2-axis Interpolation at A Constant Vector Speed (speed=1000PPS)

Setting Constant Vector Speed for 3 Axes

As same as the setting process of 2 axes, the user should first set the values of D9 and D8 of register WR5 to 0 and 1. Then, set the range R of ax2 to 1.414 times of the value of the master-axis (ax1); then, set the range R of ax3 to 1.732 times of the value of the master axis.

After setting the range of constant vector speed for 3 axes, MCX314 will use the range parameter of ax1 if only 1 axis outputs pulses. However, when 2 or 3 axes output pulses simultaneously, MCX314 will use the range parameter of ax2 or ax3 to implement the pulse period. See Fig. 2.28.

User may set the values of D9 and D8 of register WR5 to 0 and 1 for 2-axis constant vector speed driving even in the 3-axis interpolation.

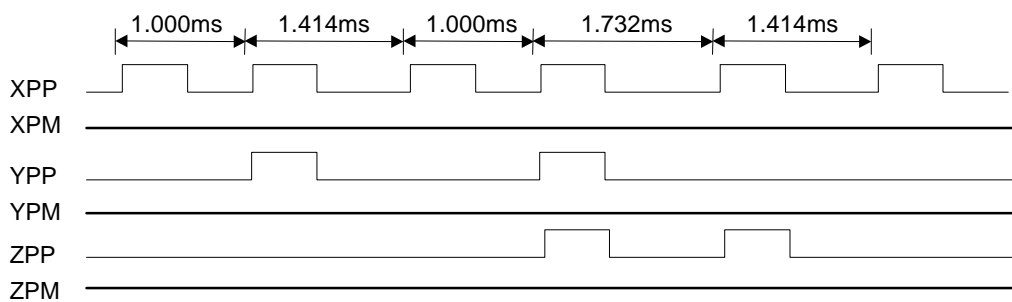


Fig. 2.28 Example for 3-axis Interpolation at A Constant Vector Speed (speed=1000PPS)

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 or 1.732 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 command into MCX314 before the previous interpolation command is finished.

Continuous Interpolation in Using D9 of RR0

If D9 (CNEXT) of register RR0 is 1, MCX314 is ready to accept the next interpolation command. If D9 is 0, the host CPU is not able to write the next interpolation command into MCX314.

The D9 will become 1 when the present command is executed. MCX314 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 MCX314 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 command to MCX314.

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.

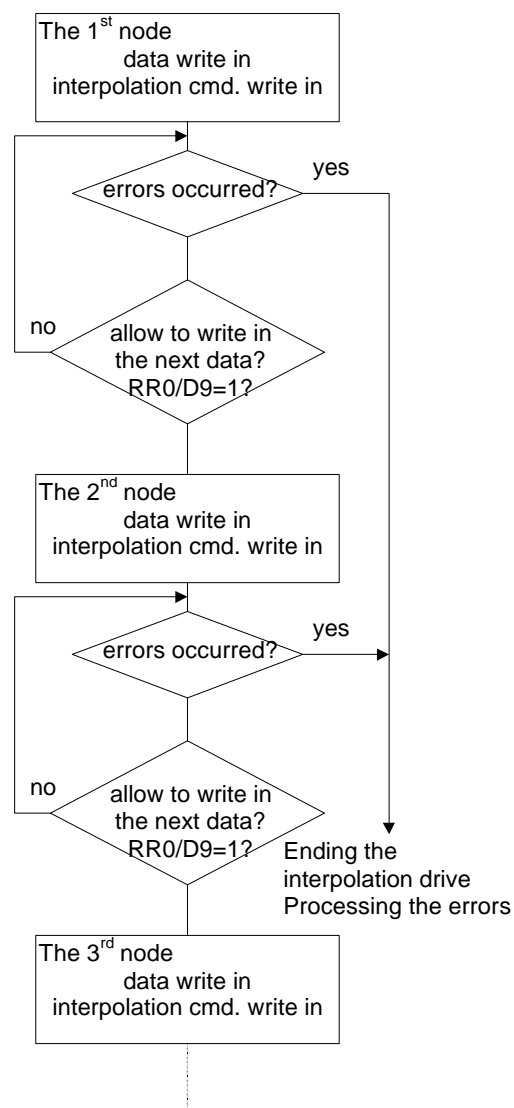
Errors Occurring in the Process of Continuous Interpolation

If an error such as over-traveling occurs in the process of continuous interpolation, the drive will stop at the present interpolation node. The following interpolation command 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 command is loaded.

Attentions for Continuous Interpolation

- Before setting the interpolation command, the user should first set other data such as center point, finish point.. for each node.
- The maximum speed for the continuous interpolation is 2MHz.
- The following interpolation command must be loaded before the previous interpolation command is finished.
- The node driving time should be longer than the time for error checking and the command setting of next node during the interpolation.
- It is impossible to operate 2-axis and 3-axis continuous interpolations at the same time.
- It is not allowed to change the axis assignment during the process of continuous interpolation.



The Example of Continuous Interpolation

Fig. 2.29 shows an example of executing continuous interpolation beginning at point (0,0) from node 1, 2, 3..to the node 8. In node 1, 3, 5, and 7, the linear interpolation will be executed; in node 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 ← 0104h write ; define ax1: X axis , ax2: Y axis, constant vector speed
WR6 ← 0900h write ; setting the parameter of master axis
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 enabling
    
```

```

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

   RR0(D9) read ; waiting for next node' 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 enabling
    
```

Procedure A

```

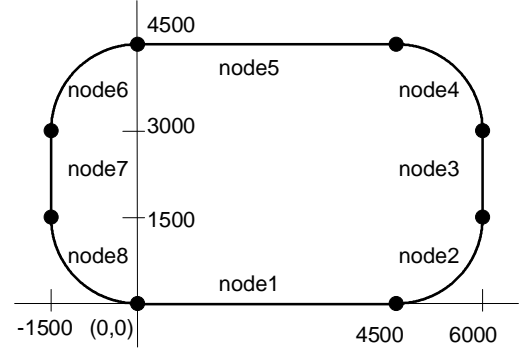
WR6 ← 0000h write ; finish point X: 0
WR7 ← 0000h write
WR0 ← 0106h write

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

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

Procedure A

...
...



www.DataSheet4U.com

(same procedure for nodes 4 ~ 8.)

2.4.6 The Acceleration / Deceleration Control in Interpolation

Different from other IC chips only allowing constant speed for executing the interpolations, MCX314 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.

i The Acceleration / Deceleration for 2-axis / 3-axis Interpolation

It is possible to perform trapezoidal and S-curve acceleration/deceleration driving during the execution of 2-axis / 3-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 master axis decelerating point. For instance, while executing 3-axis linear interpolation of master axis (ax1): X, ax2 : Y and ax3 : Z, the finish point : (X: -20000, Y: 30000, Z: -50000), and the assumed pulse numbers needed for deceleration are 5000. In such situation, the absolute value of Z axis is the largest, so we can set up $50000-5000=45000$ to be the manual deceleration point of the master axis: X. Please refer to the example of 3-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.

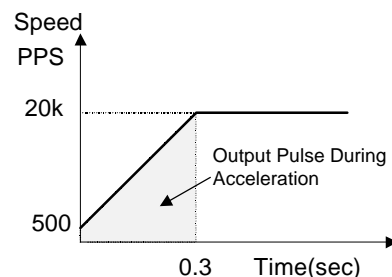
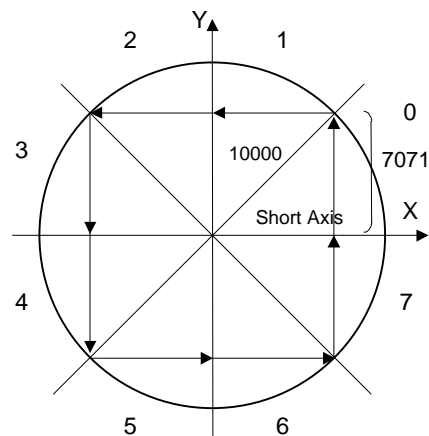
In the figure, the circle tracks through all the 8 quadrants: 0~7. In quadrant 0, Y axis is the short axis and its displacement is about

$$10000 / \pi \approx 7071$$

The total output pulses numbers of the short axis are $7071 \times 8 = 56568$.

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
WR5 ← 0004h write ; define ax1: X , ax2: Y
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: 5000x4=20000PPS
WR0 ← 0105h write

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

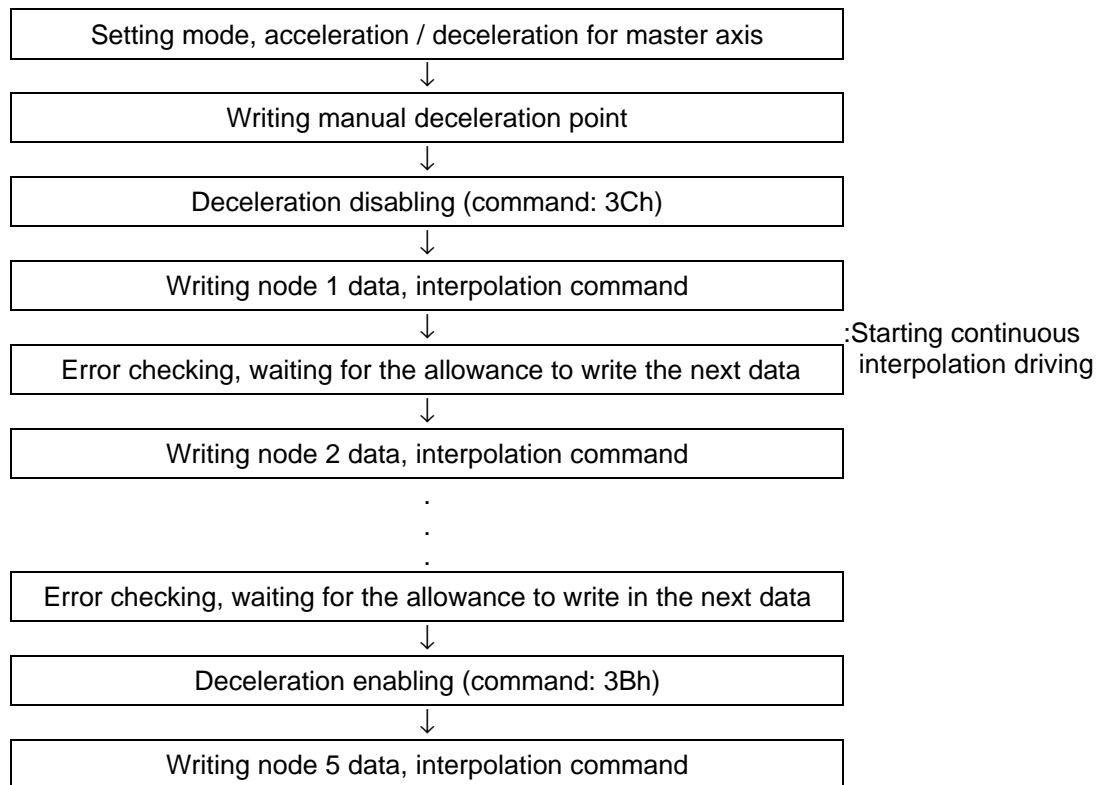
WR0 ← 003Bh write ; deceleration enabling
WR0 ← 0033h write ; CCW circular interpolation starting
    
```

i 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 the master axis executing the deceleration in the last node. The user should disable the deceleration, then start the interpolation driving. Before writing the interpolation command to the final node which will execute the deceleration, the user should enable the deceleration at first. The deceleration will start if the output pulses are larger than the master axis based pulses in the final node.

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



- i The manual deceleration point is related to the master axis' driving pulses which comes from node 5. For instance, assumed that it needs 2000 pulses for decelerating stop, and the total amount of pulse output form node 5 is 5000. So, the manual deceleration point will be $5000 - 2000 = 3000$.
- i The cycle of deceleration should be started and finished within the same node.

2.4.6 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, the master 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 master axis' s drive speed. The Low level width is kept until next command or external signal comes.

Fig. 2.30 is the example showing the execution of single-step interpolation from an external signal. The master axis' s initial speed is 500PPS, the drive speed is at 500PPS constant speed driving. The Hi level width of output pulse is 1msec.

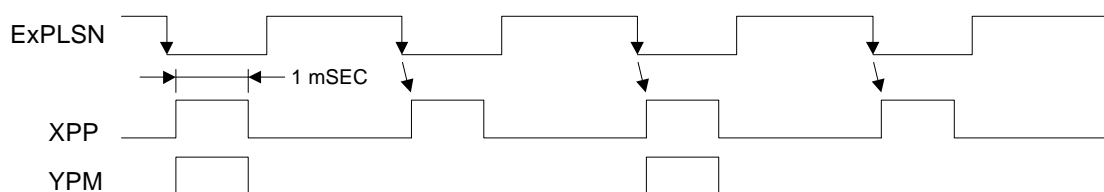


Fig. 2.30 Example of single step interpolation (500PPS) by external signal(EXPLSN)

i Command Controlled Single-step Interpolation

The command: 3Ah is for single-step interpolation. The user can set D12 of register WR5 to 1 to enable the command controlled single-step interpolation. The operating procedure is shown as follow.

- ① Set D12 of register WR5 to 1.
It will enable the command controlled single-step interpolation.
- ② Set the initial and drive speeds of the master axis in the interpolation process with the same value, and the driving becomes constant speed. If the host CPU writes single step command into MCX314 at most 1 mSEC, the user should set the drive speed more than 1000PPS.
- ③ Set interpolation data. (start point, center point.).
- ④ Write interpolation command.
Although the interpolation command is enabled, there is no pulse output because the single-step is command controlled.
- ⑤ 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.

j External Signal Controlled Single-step Interpolation

The EXPLSN pin (29) is used for the single-step interpolation from the external signal. The user can set D11 of register WR5 to 1 to enable the external signal controlled single-step interpolation. Normally, the EXPLSN 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.

- ① Set D11 of register WR5 to 1.
It will enable the external signal controlled single-step interpolation.
- ② Set the initial and driving speeds of the master 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 EXPLSN. This is necessary for this controlled mode. And it will set the MCX314 into a constant speed mode.
- ③ Set interpolation data. (starting point, center point.).
- ④ Write interpolation command.
Although the interpolation command is enabled, there is no pulse output because the single-step is command controlled.
- ⑤ EXPLSN input on Low level
The interpolation pulse will be output from each axis after 2~5 CLK the pulse falling down. The Low level pulse width of EXPLSN has to be longer than 4CLK. Furthermore, the pulse cycle of EXPLSN has to be longer than the setting speed cycle of master axis. The user may repeat the Low level of EXPLSN 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 EXPLSN Low level again to stop the driving (the user may try software reset also). After this, all the following input pulses on EXPLSN Low level will not be active.

Note: When connecting the EXPLSN Low level with the manual connector, please assure the EXPLSN signal is not chattering.

2.5 Interrupt

The interrupt is generated from X, Y, Z, or U 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. 2.31.

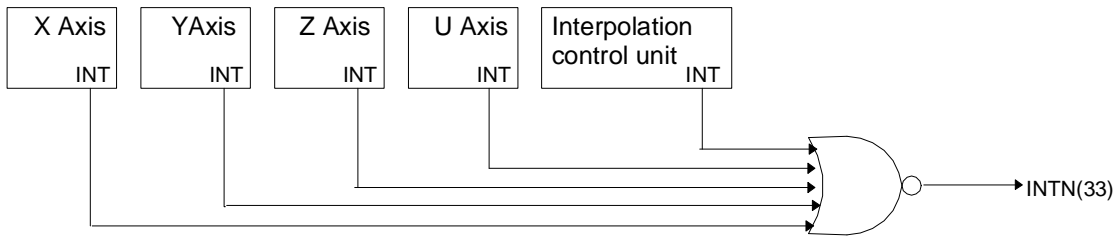


Figure 2.31 Interrupt Signal Path in IC

Every interrupt can be enabled or disabled individually. During the power resetting, all interrupt signals are disabled.

■ Interrupt of X, Y, Z, and U Axes

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

Enable / Disable nWR1 Register	Status nRR3 Register	The Factors of Interrupt Happening
D8(PULSE)	D0(PULSE)	when one pulse outputs... (The interrupt will be generated at the rising edge of pulse output for + direction driving.)
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 MCX314 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.

www.DataSheet4U.com 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.

2.6 Other Functions

2.6.1 Driving By External Pulses

Fixed pulse driving and continuous pulse driving can be controlled by either commands or external signals, which can reduce the load of host CPU.

Each axis has two input signals, nEXPP and nEXPM. 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.

■Fixed Pulse Driving Mode

Set bits D4 and D3 of register WR3 to 1 and 0 respectively, and set all the parameters of fixed pulse driving. Once nEXPP is falling down to the Low level (\downarrow) the + direction fixed pulse driving will start; once nEXPP is raising to the Hi level (\uparrow) the - direction fixed pulse 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.

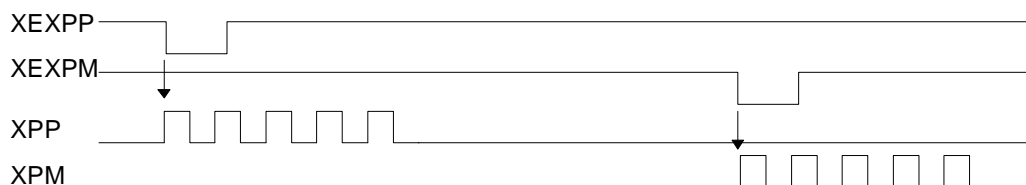


Fig. 2.32 Example of The Fixed Pulse Driving by External Signal

[Note] When connecting the input signal with a mechanical connector, the signal chattering would happen, especially if the output pulse numbers are few. Please add a de-bounce circuit to avoid the chattering.

■Continuous Pulse Driving Mode

Set bits D4 and D3 of WR3 register to be 1 and 0 respectively, and set all the parameters of continuous pulse driving. Once nEXPP is falling down to the Low level (\downarrow) the + direction continuous pulse driving will start; once nEXPP is raising to the Hi level (\uparrow) the - direction fixed pulse driving will start. When nEXPP 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.

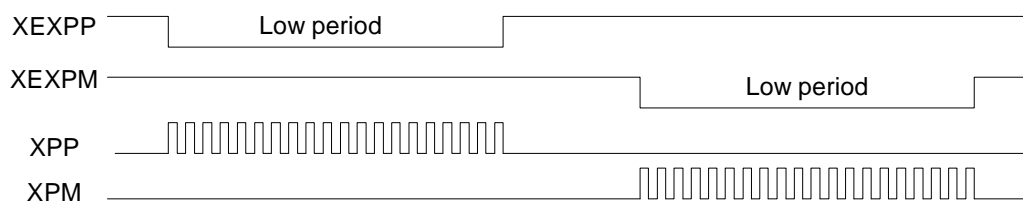


Figure 2.33 Example of Continuous Driving by External Signal

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 / direction is set on the positive logical level)

Pulse Output Type	Drive Direction	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

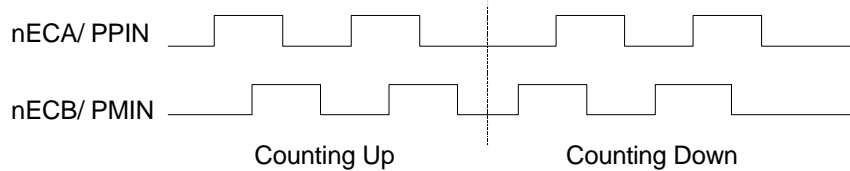
Bit D6 (PLSMD) of register WR2 is used for the selection of pulse output type.

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

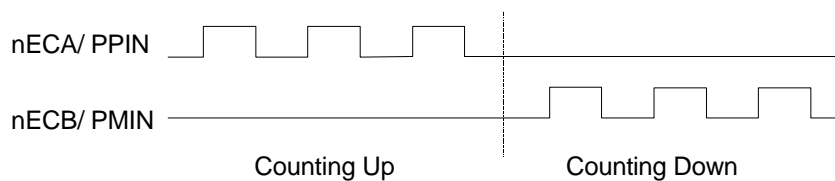
[Note] Please refer to Chapter 13.2 and 13.3 for the pulse signal (nPLS) and direction signal (nDIR) in 1-pulse 1-direction pulse outputting.

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. When A/B quadrature pulse type is selected, the position counter will count up if phase A leads phase B; the position counter will count down if phase B leads phase A. The pulse cycle can be set 1/2 or 1/4.



When Up / Down pulse type is selected, nECA/PPIN is used for count-up input, and ECB/PPIN for count-down input. The counting is during the + direction pulses are up (j) ̢



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 PRO 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

The user can use general output signals nOUT7~4 or nOUT3~0 to clear and/or reset the out put signals such as deviation counter and alarm reset of servo motor drivers.

www.DataSheet4U.com

2.6.6 Emergency Stop

Signal EMGN is able to perform the emergency stop function for all of the 4 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.

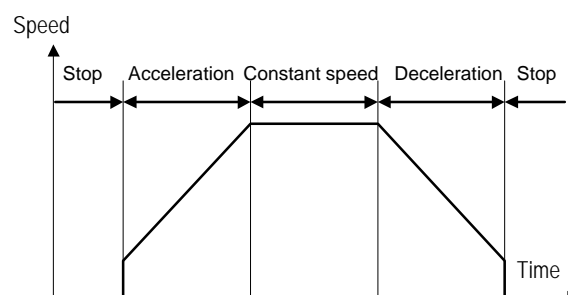
Please check the following methods to perform the emergency stop function from the host CPU.

- ① Execute the sudden stop commend for all of the 4 axes at the same time...
Appoint all of the 4 axes, then write the sudden stop command (27h) to register WR0.
- ② Reset software limit...
Write 800h to register WR0 to reset software limit.

2.6.7 Status Output

nDRIVE output signals and bits D3~0 (n-DRV) of register RR0 can be used for drive / stop status output of each axis.

The driving status of acceleration / constant speed / deceleration will be output to bits D2 (ASND), D3 (CNST) and D4 (DSDN), and also the signals nOUT6 / ASND and nOUT7 / DSND will show the levels. However, these output signals and general purpose output signals share the same terminal, D7 (OUTSL) bit of register WR3 should be set 1 for drive status output.



Drive Status	Status Register				Output Signal		
	RR0/n-DRV	nRR1/ASN D	nRR1/CNS T	nRR1/DSN D	nDRIVE	nOUT6/AS ND	nOUT7/DS ND
Stop	0	0	0	0	Low	Low	Low
Acceleration	1	1	0	0	Hi	Hi	Low
Constant Speed	1	0	1	0	Hi	Low	Low
Deceleration	1	0	0	1	Hi	Low	Hi

Moreover, in S-curve accelerating/decelerating driving, the state of acceleration /constant speed/ deceleration will be also shown to bits D5 (AASND), D6 (ACNST), and D7 (ADSND) of register RR1.

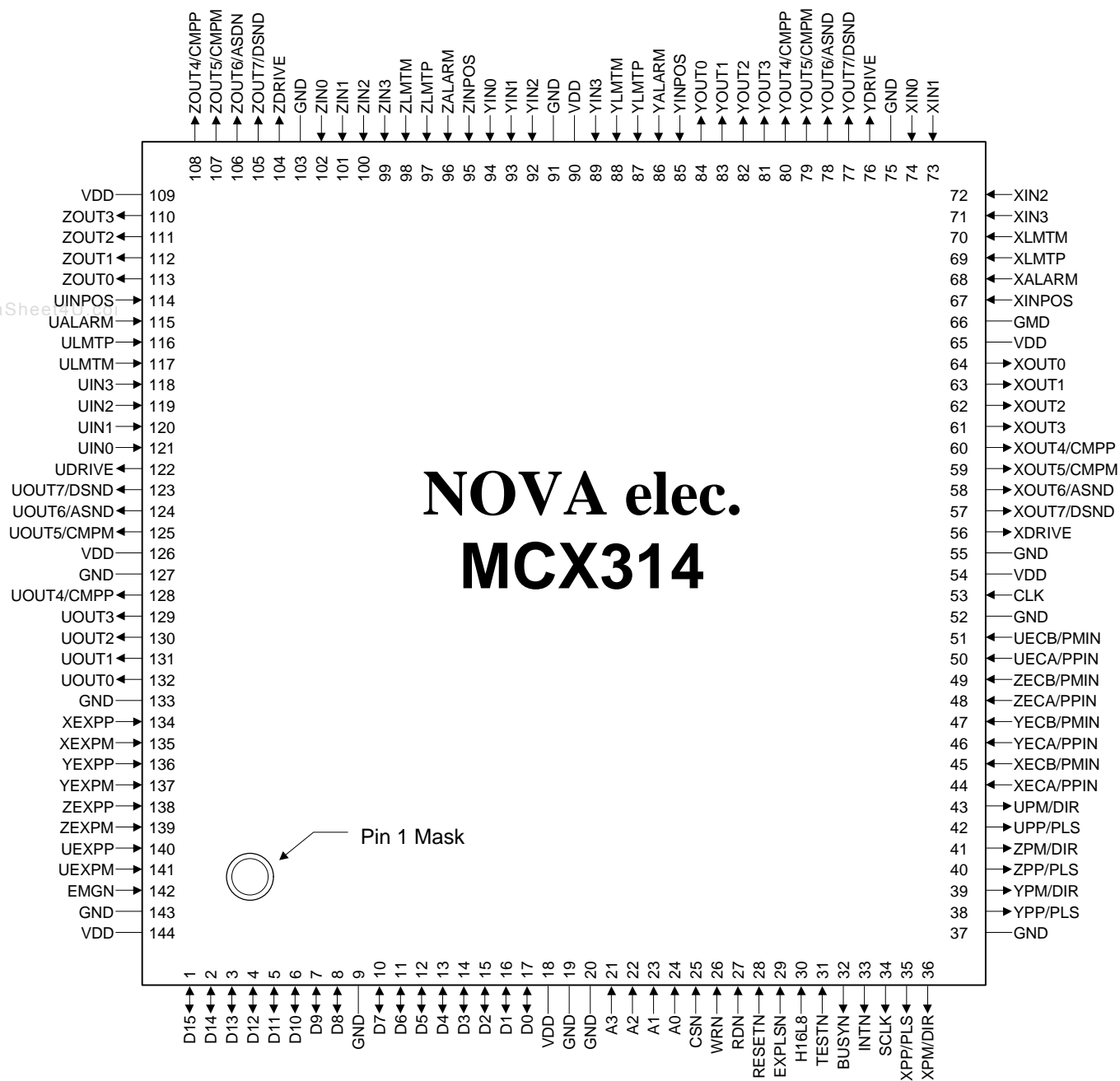
2.6.8 General Purpose Output

In MCX314, there are 8 general purpose output pins, nOUT3~0 & nOUT7~4, for each axis. However, during the outputting, nOUT7~4 cannot be used cause they share the same terminals with the position comparison output and drive status output. NOUT3~0 can be output when the output levels of register WR4 have been set. If the user wants to use nOUT7~4 signals, D7(OUTSL) of register WR3 should be set in the “general purpose output mode”, then the output levels of D11~8(OUT7~4) of register WR3 can be set for outputting. It is possible to use the general purpose output signal for motor driver current-OFF, deviation counting clear and alarm reset...

When resetting, each bit of WR4 and nWR3 registers will be cleared, then, their output levels will be kept

MCX314
on the Low level.

3. Pin Assignments and Signal Description



144pinQFP, Dimension: 30.9x30.9 mm, Pitch: 0.65mm

■ Signal Description

Signals X000, Y000, Z000, and U000 are input / output signals for X, Y, Z, and U axes, where n stands for X, Y, Z, and U. If the signals are named O00N, they are negative-active or low-active.

Signal Name	Pin No.	Input /Output	Signal Description
CLK	53	Input A	Clock: clock signal for internal synchronous loop of MCX314 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	1~8, 10~17	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	21~24	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 MCX314 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 MCX314. 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 MCX314. 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	28	Input A	Reset: reset (return to the initial setting) signal for MCX314. Setting RESETN to Low for more than 4 CLK cycles will reset MCX314. The RESETN setting is necessary when the power is on. (Note) If there is no clock input to MCX314, setting the RESETN to Low still cannot reset this IC.
EXPLSN	29	Input A	External Pulse: pulse input signal for external pulse interpolation The normal setting is Hi level. When the external pulse interpolation occurs, EXPLSN is down (↓) the interpolation calculation starts, and one pulse for each axis interpolation is output. The width of EXPLSN on the Low level must be more than 4 CLK.
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 MCX314, the process will take 2 CLK to 4 CLK (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 MCX314 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. There is no SCLK output when RESETN is on the Low level.
XPP/PLS YPP/PLS ZPP/PLS UPP/PLS	35 38 40 42	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.

Signal Name	Pin No.	Input /Output	Signal Description
XPM/DIR YPM/DIR ZPM/DIR UPM/DIR	36 39 41 43	Output A	Pulse -/Pulse: - direction drive 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 ZECA/PPIN UECA/PPIN	44 46 48 50	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 ZECB/PMIN UECB/PMIN	45 47 49 51	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.
XDRIVE YDRIVE ZDRIVE UDRIVE	56 76 104 122	Output A	Drive: output signal of driving It will become Hi level when the driving command of +/- direction pulse output is executed. When the nINPOS signal of servo motor is enabled, nINPOS will be active, and the Drive will become Hi.
XOUT7/DSND YOUT7/DSND ZOUT7/DSND UOUT7/DSND	57 77 105 123	Output A	General Output 7 / Descend: general purpose output signals After the axis is appointed by WR0 register, nOUT7~4 can output the 1/0 data of D11~8 in WR3 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/ASND YOUT6/ASND ZOUT6/ASND UOUT6/ASND	58 78 106 124	Output A	General Output 6 / 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.
XOUT5/CMPM YOUT5/CMPM ZOUT5/CMPM UOUT5/CMPM	59 79 107 125	Output A	General Output 5 / 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-.
XOUT4/CMPP YOUT4/CMPP ZOUT4/CMPP UOUT4/CMPP	60 80 108 128	Output A	General Output 4 / 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+.
XOUT3-0 YOUT3-0 ZOUT3-0 UOUT3-0	61~64 81~84 110~113 129~132	Output A	General Output 3~0: 4 general output signals for each axis nOUT3~0 can output the 1/0 data of D15~0 in WR4 register to Hi / Low. They become Low when the IC is reset. Compared with the setting of nOUT7~4, it is easier cause there is no need to have the appointed axis.

Signal Name	Pin No.	Input /Output	Signal Description
XINPOS YINPOS ZINPOS UINPOS	67 85 95 114	Input A	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 ZALARM UALARM	68 86 96 115	Input A	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.
XLMTTP YLMTTP ZLMTTP ULMTTP	69 87 97 116	Input A	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 HKMT+ of RR2 register becomes 1.
XLMTM YLMTM ZLMTM ULMTM	70 88 98 117	Input A	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 HKMT- of RR2 register becomes 1.
XIN3-0 Y IN3-0 Z IN3-0 U IN3-0	71~74 89,92~94 99~102 118~121	Input A	Input 3~0: 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 IN3~IN0. The signal status can be read from register RR4 / RR5.
XEXPP YEXPP ZEXPP UEXPP	134 136 138 140	Input A	External Operation +: + direction drive starting signal from external source When the fixed pulse driving is commanded from an external source, + direction driving will start if this signal is down (\downarrow) Otherwise, when the continuous pulse driving is commanded from an external source, + driving will start if this signal is on the Low level.
XEXPM YEXPM ZEXPM UEXPM	135 137 139 141	Input A	External Operation -: - direction drive starting signal from external source When the fixed pulse driving is commanded from an external source, - direction driving will start if this signal is down (\downarrow) Otherwise, when the continuous pulse driving is commanded from an external source, - direction driving will start if this signal is on the Low level.
EMGN	142	Input A	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 axes 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	9,19,20, 37,52,55, 66,75,91, 103,127, 133,143		Ground (0V) Terminal All of the 13 pins must be connected to 0V.
VDD	18,54,65, 90,109, 126,144		+ 5V Power Terminal. All of the 7 pins must be connected to +5V.

■ Input/ Output Loop

Input A	More than 10 k Ω ~ hundreds of kilo impedance is for internal impedance, which can pull up the VDD to the TTL level input of Smith trigger. CMOS and TTL can be connected. The user should open, or pull up with + 5V if the input is not used.
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 collector 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 k Ω) 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).

www.DataSheet4U.com

■ Notes for the Design of Circuitry

(1) De-coupling Capacitor

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

(2) Noise Generated by Terminal Induction

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

(3) Reflection on Transfer Path

The load capacity for outputting types A, B, and bi-direction type A is 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 MCX314, and what are the mapping addresses of these registers. Please refer to Chapter 2.4.3 for the registers (BP1P/M, BP2P/M, BP3P/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			Symbol	Register Name	Contents
A2	A1	A0			
0	0	0	WR0	Command Register	for setting axis assignment and command
0	0	1	XWR1 YWR1 ZWR1 UWR1	X axis mode register 1 Y axis mode register 1 Z axis mode register 1 U axis mode register 1	for setting the logical levels of external decelerating stop, enable / disable, and the valid / invalid of interrupt for each axis for each axis
0	1	0	XWR2 YWR2 ZWR2 UWR2	X axis mode register 2 Y axis mode register 2 Z axis mode register 2 U axis mode register 2	for setting the limit signal mode, driving pulse mode, encoder input signal mode, and the logical levels and enable / disable of servo motor signal for each axis
			BP1P	BP1P register	for setting the + direction bit data of the first axis in bit pattern interpolation
0	1	1	XWR3 YWR3 ZWR3 UWR3	X axis mode register 3 Y axis mode register 3 Z axis mode register 3 U axis mode register 3	for setting the manual deceleration, individually decelerating, and S-curve acceleration/ deceleration mode for each axis, external operation mode, and general purpose output OUT7~4
			BP1M	BP1M register	for setting the - direction bit data of the first axis in bit pattern interpolation
1	0	0	WR4	Output register	for setting the general output OUT3 ~ 0.
			BP2P	BP2P register	for setting the + direction bit data of the second axis in bit pattern interpolation
1	0	1	WR5	Interpolation mode register	axis assignment for settings the constant speed driving mode, step output mode and interrupt
			BP2M	BP2M register	for setting the - direction bit data of the second axis in bit pattern interpolation
1	1	0	WR6	Data writing register 1	for setting the low word 16-bit (D15-D0) for data writing.
			BP3P	BP3P register	for setting the + direction bit data of the third axis in bit pattern interpolation
1	1	1	WR7	Data writing register 2	for setting the high word 16-bit (D31-D16) for data writing.
			BP3M	BP3M register	for setting the - direction bit data of the third axis in bit pattern interpolation

- Each axis is with WR1, WR2 and WR3 mode registers. Each register is for 4-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 are BP1P~3P and BP1M~3M. 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.
- Please be noted that registers WR6 and BP3P / WR7 and BP3M share the same register hardware.
- 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 YRR1 ZRR1 URR1	X axis status register 1 Y axis status register 1 Z axis status register 1 U axis status register 1	comparison result, acceleration status and finishing status
0 1 0	XRR2 YRR2 ZRR2 URR2	X axis status register 2 Y axis status register 2 Z axis status register 2 U axis status register 2	error message
0 1 1	XRR3 YRR3 ZRR3 URR3	X axis status register 3 Y axis status register 3 Z axis status register 3 U axis status register 3	interrupt message
1 0 0	RR4	Input register 1	I/O input for X and Y axes
1 0 1	RR5	Input register 2	I/O input for Z and U axes
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 WR1, WR2 and WR3 mode registers. Each register is for 4-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 (WR0L), then to the low word byte (WR0H).

■ Write Register in 8-bit Data Bus

Address				Write Register
A3	A2	A1	A0	
0	0	0	0	WR0L
0	0	0	1	WR0H
0	0	1	0	XWR1L, YWR1L, ZWR1L, UWR1L
0	0	1	1	XWR1H, YWR1H, ZWR1H, UWR1H
0	1	0	0	XWR2L, YWR2L, ZWR2L, UWR2L
0	1	0	1	XWR2H, YWR2H, ZWR2H, UWR2H
0	1	1	0	XWR3L, YWR3L, ZWR3L, UWR3L
0	1	1	1	XWR3H, YWR3H, ZWR3H, UWR3H
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				Read Register
A3	A2	A1	A0	
0	0	0	0	RR0L
0	0	0	1	RROH
0	0	1	0	XRR1L, YRR1L, ZRR1L, URR1L
0	0	1	1	XRR1H, YRR1H, ZRR1H, URR1H
0	1	0	0	XRR2L, YRR2L, ZRR2L, URR2L
0	1	0	1	XRR2H, YRR2H, ZRR2H, URR2H
0	1	1	0	XRR3L, YRR3L, ZRR3L, URR3L
0	1	1	1	XRR3H, YRR3H, ZRR3H, URR3H
1	0	0	0	RR4L, BP2PL
1	0	0	1	RR4H, BP2PH
1	0	1	0	RR5L, BP2ML
1	0	1	1	RR5H, BP2MH
1	1	0	0	RR6L, BP3PL
1	1	0	1	RR6H, BP3PH
1	1	1	0	RR7L, BP3ML
1	1	1	1	RR7H, BP3MH

4.3 Command Register: WR0

Command register is used for the axis assignment and command registration for each axis in MCX314. 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.

H								L							
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RESE T	0	0	0	U	Z	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.
- D11 ~ 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 decide which axis' s register will be written.

The register consists of the bit for setting enable / disable and enable logical levels of input signal IN3~IN0 (decelerating stop / sudden stop during the driving) and bit for occurring the interrupt enable / disable.

Once IN3~IN1 are active, when the fixed pulse / continuous pulse driving starts, and also when IN 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 .

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≥C+	P<C+	P<C-	P≥C-	PULS E	IN3-E	IN3-L	IN2-E	IN2-L	IN1-E	IN1-L	IN0-E	IN0-L
Interrupt Enable / Disable								Driving Stop Input Signal Enable / Disable							

- D7,5,3,1 INm-E The bit for setting enable / disable of driving stop input signal INm 0: disable, 1: enable
D6,4,2,0 INm-L The bit for setting enable logical levels for input signal INm 0: stop on the Low level, 1: stop on the Hi level

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

- D8 PULSE Interrupt occurs when the pulse is up (i) (drive pulse is set on the positive logical level)
D9 P ≥ C - Interrupt occurs when the value of logical / real position counter is larger than or equal

- D10 P < C- to that of COMP- register
Interrupt occurs when the value of logical / real position counter is smaller than that of COMP- register
 - D11 P < C + to that of COMP+ register
Interrupt occurs when the value of logical / real position counter is smaller than that of COMP+ register
 - D12 P ≥ C + to that of COMP+ register
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 start of the constant speed drive during an acceleration / deceleration driving
 - D14 C - STA Interrupt occurs at the end 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

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	PINM D	DIR-L	PLS-L	PLSM D	CMPS L	HLMT- +	HLMT +	LMTM D	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 commends will not be executed.
- 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 commends 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 (nLMTM) 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 COMPSL 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 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	+ 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
Real position counter will count up or down when encoder input signal is triggered.
When quadrature pulse input type is engaged, the “count up” will happen if the positive

logical level pulses are input to phase A; the “count down” will happen if the positive logical level pulses are input to phase B. So, it will count up and down when these 2 signals go up and 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 PIND 1,0 The division setting for quadrature encoder input.

D11	D10	Division	Up / down pulse input is not available.
0	0	1/1	
0	1	1/2	
1	0	1/4	
1	1	Invalid	

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, MCX314 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 doesn't 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, and the setting of general purpose output OUT7~4.

WR3	H				L											
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	0	OUT7	OUT6	OUT5	OUT4	OUTS L	0	0	EXOP 1	EXOP 0	SACC	DSND E	MANL D

D0 MANLD Setting manual / automatic deceleration for the fixed pulse 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: in accordance with the rate of the acceleration

1: individual decelerating rate setting

When 0 is set, the deceleration will follow the acceleration setting. So, 0 must be set for automatic deceleration. When 1 is set, the rates of acceleration and deceleration should be different, So, 1 must be set for manual deceleration.

D2 SACC Setting trapezoidal driving / S-curve acceleration / deceleration driving

0: trapezoidal driving, 1: S-curve acceleration / deceleration driving

Before S-curve acceleration / deceleration driving is engaged, jerk (K) should be set.

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

D4	D3	
0	0	external signals disabled
0	1	continuous driving mode
1	0	fixed pulse driving mode
1	1	external signals disabled

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 pulse driving mode is engaged, the + direction fixed pulse driving starts once the nEXPP signal is falling to the Low level from the Hi level; the - direction fixed pulse

- D7 OUTSL Driving status outputting or used as general purpose output signals (nOUT7~4)
 0: nOUT7~4: general purpose output
 The levels of D11~8 will be output through nOUT7~4.
 1: nOUT4~7: driving status output (see the table below)

Signal Name	Output Description
OUT4/CMPP	Hi: if logical / real position counter \geq COMP+ register Low : if logical / real position counter $<$ COMP+ register
OUT5/CMPM	Hi: if logical / real position counter $<$ COMP- register Low: if logical / real position counter \geq COMP- register
OUT6/ASND	When the driving command is engaged, the level becomes Hi once the driving status is in acceleration.
OUT7/DSND	When the driving command is engaged, the level becomes Hi once the driving status is in deceleration.

- D11~8 OUTm Level setting for output signals OUT7~4 as general purpose output signals
 0: Low level output, 1: Hi level output

D15~D0 will be set to 0 while resetting. D15~12, D5 and D6 should be always set 0.

4.7 Output Register: WR4

This register is used for setting the general purpose output signals nOUT3~0. This 16-bit register locates 4 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.

	H								L							
WR4	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	UOUT	UOUT	UOUT	UOUT	ZOUT	ZOUT	ZOUT	ZOUT	YOUT	YOUT	YOUT	YOUT	XOUT	XOUT	XOUT	XOUT
	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0

D15~D0 will be set to 0 while resetting, and nOUT3~0 signals become Low level.

4.8 Interpolation Mode Register: WR5

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

	H								L							
WR5	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	BPINT	CIINT	0	CMPL	EXPL	0	LSPD	LSPD	0	0	AX31	AX30	AX21	AX20	AX11	AX10
				S	S		1	0								
	Interrupt		Step output		Constant Vector Speed					ax3		ax2		Ax1		

D1, 0 AX11, 10 ax1 (master axis) assignment for interpolation
Axis codes are shown as follows.

Axis	Code (Binary)	
X	0	0
Y	0	1
Z	1	0
U	1	1

1st axis: X, 2nd axis: Y, 3rd axis: Z

D5	D4	D3	D2	D1	D0
1	0	0	1	0	0

For ax1 (master axis) will have the basic pulses of starting interpolation calculation, the speed parameter which is for constant / acceleration / deceleration driving should be set before the driving.

D3, 2 AX21, 20 ax2 assignment according to the codes shown in the table above

D5,4 AX31, 30 ax3 assignment for 3-axis interpolation, according to the codes shown in the table above

Setting any value if it is only 2-axis interpolation.

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	3-axis constant vector speed

When 2-axis constant vector speed mode is engaged, the user should set the range (R) of ax2 to be 1.414 times of the range (R) of master axis (ax1).

When 3-axis constant vector speed mode is engaged, the user should set the range (R) of ax2 to be 1.414 times and the range (R) of ax3 to be 1.732 times of the range (R) of master axis (ax1).

D11 EXPLS When it is 1, the external (EXPLSN) controlled single step interpolation mode is engaged.

D12 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 RR6 setting, and the high-word data-writing 16-bit (WD31~WD16) is for register RR7 setting.

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

	H								L							
WR7	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	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, to set the finish point of circular interpolation is using 4 bytes. Even the calculation range (-8388608 ~ +833607) is 24-bit long, the user should fill the total 32 bytes.

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.

	H							L								
RR0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	-	BPSCI	BPSC 0	ZONE 2	ZONE 1	ZONE 0	CNEXT T	I-DRV	U- ERR	Z-ERR	Y-ERR	X-ERR	U- DRV	Z-DRV	Y-DRV	X-DRV
									Error Status of Each Axis				Driving Status of Each Axis			

- D3~0 n-DRV Displaying driving status of each axis
When the bit is 1, the axis is outputting drive pulses; when the bit is 0, the driving of the axis is finished.
Once the in-position input signal nINPOS for servo motor is active, nINPOS will return to 0 after the drive pulse output is finished.
- D7~4 n-ERR Displaying error status of each axis
If any of the error bits (D5~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
When the bit is 1, it is ready for inputting parameters for next node and also ready for writing interpolation command data.
- D12~10 ZONE0
ZONE1
ZONE2 Displaying the quadrant of the current position in circular interpolation

D12	D11	D10	Quadrant
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

D14, 13 BPSC1, 0 In bit pattern interpolation driving, it displays the value of the stack counter (SC).

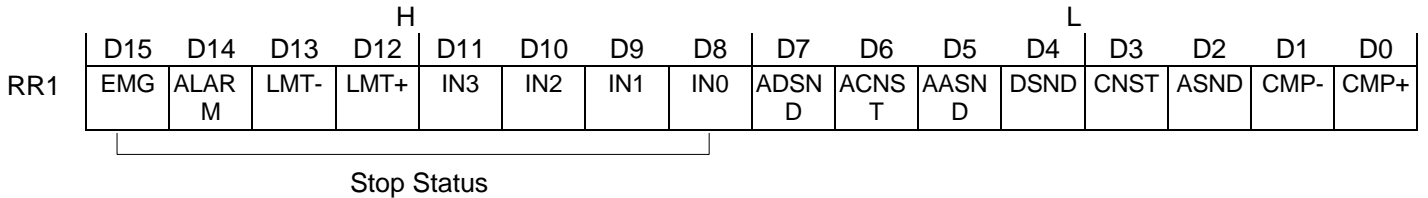
D14	D13	Stack Counter (SC) Value
0	0	0
0	1	1
1	0	2
1	1	3

In bit pattern interpolation driving, when SC = 3, it shows the stack is full. When SC = 2, there is one word (16-bit) space for each axis. When SC = 1, there is a 2-word (16-bit) space for each axis. When SC = 0, it shows all the stacks are empty, and the bit-pattern interpolation is finished.

4.11 Status Register 1: RR1

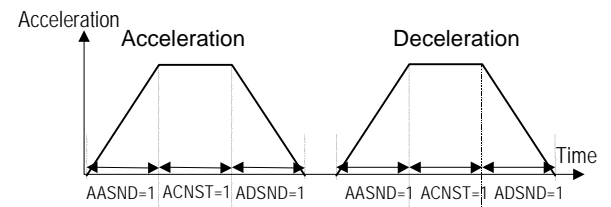
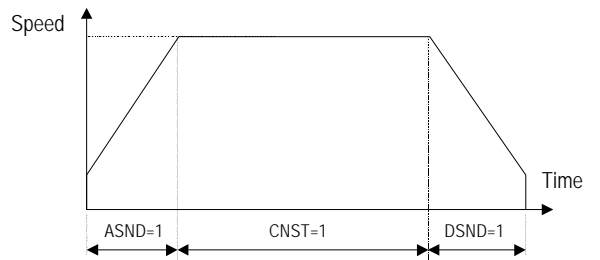
Each axis is with status register 1. The axis specified by NOP command or the condition before decide which axis' s register will be read.

The register can display the comparison result between logical / real position counter and COMP +/- , the acceleration status of acceleration / deceleration driving, jerk of S-curve acceleration / deceleration and the status of driving finishing.



www.DataSheet4U.com

- D0 CMP + Displaying the comparison result between logical / real position counter and COMP+ register
1: logical / real position counter ≥ COMP+ register
0: logical / real position counter < COMP+ register
- D1 CMP - Displaying the comparison result between logical / real position counter and COMP- register
1: logical / real position counter ≤ COMP- register
0: logical / real position counter > COMP- register
- D2 ASND It becomes 1 when in acceleration.
- D3 CNST It becomes 1 when in constant speed driving.
- D4 DSND It becomes 1 when in deceleration.
- D5 AASND In S-curve, it becomes 1 when acceleration / deceleration increases.
- D6 ACNST In S-curve, it becomes 1 when acceleration / deceleration keeps constant.
- D7 ADSND In S-curve, it becomes 1 when acceleration / deceleration decreases.
- D11~8 IN3~0 If the driving is stopped by one of external decelerating stop signals (nIN3 ~ 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 (nLMTP), 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 pulse driving and continuous driving are shown as follows:

- ① when all the drive pulses are output in fixed-pulse driving,
- ② when deceleration stop or sudden stop command is written,
- ③ when software limit is enabled, and is active,
- ④ when external deceleration signal is enabled, and active,
- ⑤ when external limit switch signals (nLMTP, nLMTM) become active,
- ⑥ when nALARM signal is enabled, and active, and
- ⑦ when EMGN signal is on the Low level.

Above factors ① and ② can be controlled by the host CPU, and factor ③ can be confirmed by register RR2 even the driving is finished. As for factors ④ ~ ⑦, 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. Once the bit becomes 1, it reflects an error occurs. When one or more of D5~D0 bits of RR2 register are 1, n-ERR bits of main status register RR0 become 1.

RR2	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	-	-	-	-	-	-	-	-	-	-	EMG	ALAR M	HLMT-	HLMT +	SLMT-	SLMT +

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

D1 SLMT - During the - direction driving, when logical / real position counter $<$ COMP- (COMP- enabled, 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.

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 which is with the interrupt factor becomes 1. The user should set the interrupt factor through register WR1 to perform the interrupt.

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-	PULS E

D0 PULSE When the drive pulse is up (;) (drive pulse is set on the positive logical level)

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. When 8-bit data bus is used, the reading data of RR3L register is cleared.

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.

	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RR4	Y-ALM	Y-INP	Y-EX-	Y-EX+	Y-IN3	Y-IN2	Y-IN1	Y-IN0	X-ALM	X-INP	X-EX-	X-EX+	X-IN3	X-IN2	X-IN1	X-IN0

	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RR5	U-ALM	U-INP	U-EX-	U-EX+	U-IN3	U-IN2	U-IN1	U-IN0	Z-ALM	Z-INP	Z-EX-	Z-EX+	Z-IN3	Z-IN2	Z-IN1	Z-IN0

Bit Name	Input Signal
n-IN0	nIN0
n-IN1	nIN1
n-IN2	nIN2
n-IN3	nIN3

Bit Name	Input Signal
n-EX+	nEXPP
n-EX-	nEXPM
n-INP	nINPOS
n-ALM	nALARM

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.

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

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

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

5. Command Lists

■ Write Commands

Code	Command	Symbol	Data Range	Data Length
00	Range setting	R	8,000,000(multiple=1)~16,000(multiple=500)	4 bytes
01	Jerk setting	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 of interpolation setting	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	-2,147,483,648 ~ +2,147,483,647	4
0C	COMP - register setting	CM	-2,147,483,648 ~ +2,147,483,647	4
0D	Acceleration counter offsetting	AO	0 ~ 65,535	2
0F	NOP (used for axis switching)			

[Note] Data range is shown above. Some parameter data ranges are shorter than data length. When those parameters are written, the total data length should be completely filled.

■ Formula Calculation for Parameters

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

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

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

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

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

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

■ Data Reading Commands

Code	Command	Symbol	Data Range	Data Length
10	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

■ Driving Commands

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

■ Interpolation Commands

Code	Command
30h	2-axis leaner interpolation
31h	3-axis leaner interpolation
32h	CW circular interpolation
33h	CCW circular interpolation
34h	2-axis bit pattern interpolation
35h	3-axis bit pattern interpolation
36h	BP register writing enabled*
37h	BP register writing disabled
38h	BP data stack
39h	BP data clear
3Ah	1-step interpolation
3Bh	Deceleration valid
3Ch	Deceleration invalid
3Dh	Interpolation interrupt clear

*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, the high word data can be written into register WR7 and the low word into register WR6. 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]

- 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.
- Except acceleration offset (OA), the other parameters are unknown while resetting. So, please per-set proper values for those driving related parameters before the driving starts.

www.DataSheet4U.com

6.1 Range Setting

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

"R" is the parameter determining the multiple of drive speed, acceleration / deceleration and jerk. The multiple calculation is shown in the following formula:

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

For the parameter setting range of drive speed, acceleration / deceleration and jerk 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) can not be changed during the driving. The speed will be changed discontinuously.

6.2 S-curve Acceleration Rate Setting

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

"K" is the parameter determining the increasing / decreasing rate of acceleration / deceleration, in a time unit, of S-curve acceleration / deceleration. 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 S-curve acceleration is 1 ~ 65,535, the jerk range is shown as follows:

	K=65535	K=1
When Multiple = 1,	954 PPS/SEC ²	~ 62.5 x 10 ⁶ PPS/SEC ²
When Multiple = 500,	477 x 10 ³ PPS/SEC ²	~ 31.25 10 ⁹ PPS/SEC ²

*In this manual, jerk is defined the increasing / decreasing rate of acceleration / deceleration in a time unit. However, jerk should cover the decreasing rate of acceleration and increasing rate of acceleration.

6.3 Acceleration Setting

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

“A” is the parameter determining the acceleration or deceleration of the trapezoidal driving. For S-curve acceleration / deceleration (see fig. 2.9, page 7), it shows the linear acceleration until a specific value (A) driving. 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:

	A=1	A=8000
When Multiple = 1	125 PPS/SEC	~ 1 x 10 ⁶ PPS/SEC.
When Multiple = 500	62.5 x 10 ³ PPS/SEC	~ 500 x 10 ⁶ PPS/SEC

6.4 Deceleration Setting

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

When acceleration / deceleration is set individually (D1 of register WR3 = 1), “D” is the parameter determining the deceleration of the trapezoidal driving. For S-curve acceleration / deceleration, the designated deceleration can be set (see fig 2.9, page 7) until a specific value (D) is driving. The deceleration calculation is shown in the following formula:

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

When acceleration / deceleration is set individually (D1 of register WR3 = 1), the automatic deceleration cannot be performed. The user should use manual deceleration.

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 constant speed period in trapezoidal driving. The initial speed calculation is shown in the following formula:

$$\text{Initial Speed (PPS)} = SV \times 125 \times \frac{8,000,000}{\text{Multiple}}$$

For stepper motors, the user should set the initial speed smaller than the self-starting frequency of stepper motors.

For servo motors, the recommended setting initial speed is higher than the value of $\sqrt{\text{acceleration}}$. For example, if acceleration / deceleration = 125000 PPS/SEC, the speed setting is better larger than $\sqrt{125000}$ = 354 PPS.

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:

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

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]

- In fixed pulse 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.
- In fixed pulse trapezoidal driving, the frequent changes of drive speed may occur residual pulses in the ending 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	2 bytes

Output pulse number setting:

The parameter “P” is setting total output pulse numbers in fixed pulse driving. The value is absolute, unsigned number. The output pulse numbers can be changed during the driving.

Interpolation finish point setting:

This parameter is also setting the finish point of each axis in linear and circular interpolations. The finish points of these axes should be set by relative numbers in 24-bit data length.

Output pulse number setting and interpolation finish point setting should be set for 32-bit data length.

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 pulse 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:

Manual Decelerating Point = Output Pulse Numbers – Pulse Number for Deceleration

6.9 Circular Center Setting

Code	Command	Symbol	Data Range	Data Length
08h	Circular Center 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 nOUT4/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 nOUT5/CMPPM 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	0 ~ 65,535	4 bytes

“AO” is the parameter executing acceleration counter offset.

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

6.15 NOP (Used for Axis Switching)

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

This command doesn't execute anything.

However, it can be used to assign the accessing axis(axis) according to WR1~3 registers of each axis.

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]

- It requires 250 nSEC (maximum) to access the command code when CLK=16MHz. Please read registers RR6 and 7 within this period of time after the present command is written.
- The axis assignment is for one axis. If more than one axes are assigned, the data reading priority is X > Y > Z > U.

7.1 Logical Position Counter Reading

Code	Command	Symbol	Data Range	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).

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 within this period of time.

8.1 +Direction Fixed Pulse Driving

Code	Command
20h	+Direction Fixed Pulse Driving

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

In driving, real 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).

	Multiple (R)	Jerk (K)	Acceleration (A)	Initial Speed (SV)	Drive Speed (V)	Output Pulse (P)
Constant speed	○	X	X	○	○	○
Trapezoidal driving	○	X	○	○	○	○
S-curve Acc./Dec.	○	○	○	○	○	○

8.2 -Direction Fixed Pulse Driving

Code	Command
21h	-Direction Fixed pulse Driving

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

In driving, real 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, real 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, real 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 main 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 / 3 axes linear interpolation, CW / CCW circular interpolation, 2 / 3 axes bit pattern interpolation and other related commands. There is no need to make the axis assignment in setting bits D11~8 of command register WR0. Please set 0 in those bits.

Two procedures should be followed before the interpolation command is executed:

- ① interpolation accessing axes assignment (set in bits D5~D0 of register WR5)
- ② speed parameter setting for master axis

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 2-axis interpolation from present point to finish point. Before driving, the finish point of the 2 corresponding axes should be set by incremental value.

9.2 3-Axis Linear Interpolation

Code	Command
31h	3-axis linear interpolation

This command performs 3-axis interpolation from present point to finish point. Before driving, the finish point of the 3 corresponding axes should be set by incremental value.

9.3 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 2 corresponding axes should be set by incremental value.

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

9.4 CCW Circular Interpolation

Code	Command
33h	CCW circular 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 the 2 corresponding axes should be set by incremental value.

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

9.5 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 the two interpolating axes should be set, and the setting bit data of each axis (each direction) is at most 16; $\Sigma = 48$ -bit. Once the data is over than 48-bit, those remaining data can be filled during the driving.

9.6 3-Axis Bit Pattern Interpolation Drive

Code	Command
35h	3-axis bit pattern interpolation

This command performs 3-axis bit pattern interpolation.

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

9.7 BP Register Data Writing Enabling

Code	Command
36h	BP register data writing enabling

This command enables the bit pattern data writing registers BP1P/M, BP2P/M and PB3P/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.

9.8 BP Register Data Writing Disabling

Code	Command
37h	BP register data writing disabling

This command disables the bit pattern data writing registers BP1P/M, BP2P/M and PB3P/M.

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

9.9 BP Data Stack

Code	Command
38h	BP data stacking

This command stacks the data of bit pattern data writing registers BP1P/M, BP2P/M, and BP3P/M.

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

9.10 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.

9.11 Single Step Interpolation

Code	Command
3Ah	Single step interpolation

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

When D12 bit of register WR5 is set 1, the single step interpolation can be performed. After this command is issued, single step interpolation starts.

9.12 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.

9.13 Deceleration Disabling

Code	Command
3Ch	Deceleration disabling

This command disables the automatic or manual deceleration in interpolation.

9.14 Interpolation Interrupt Clear

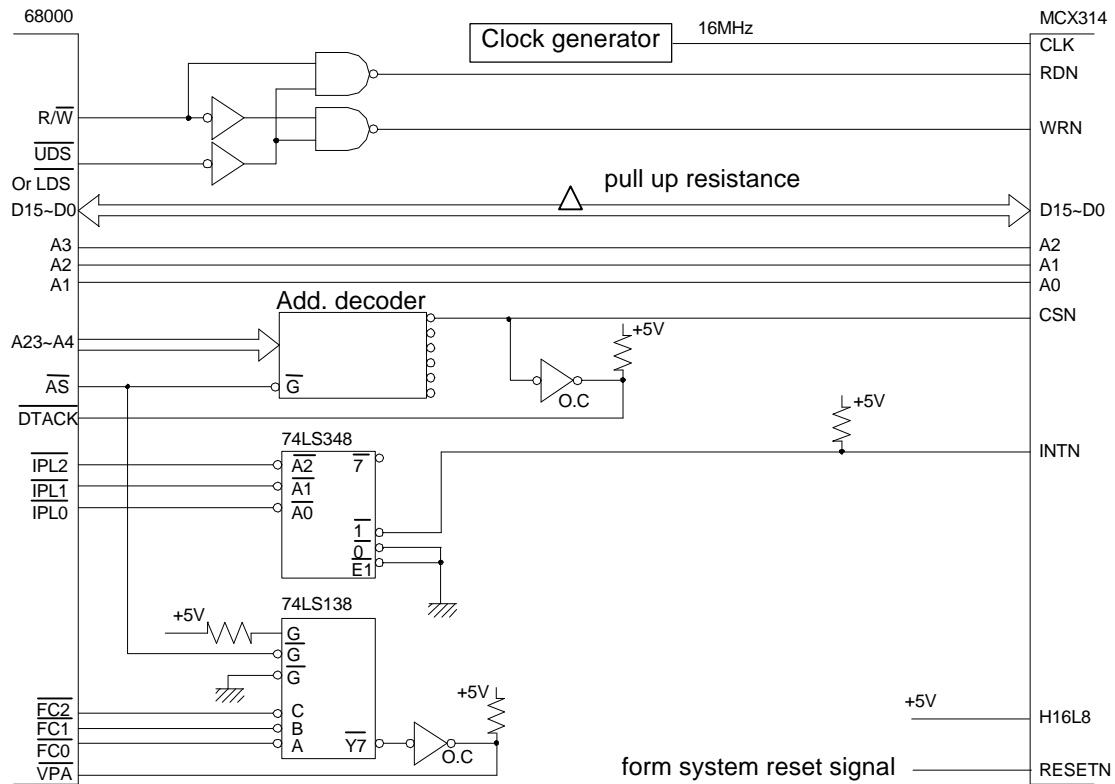
Code	Command
3Dh	Interpolation interrupt clear

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

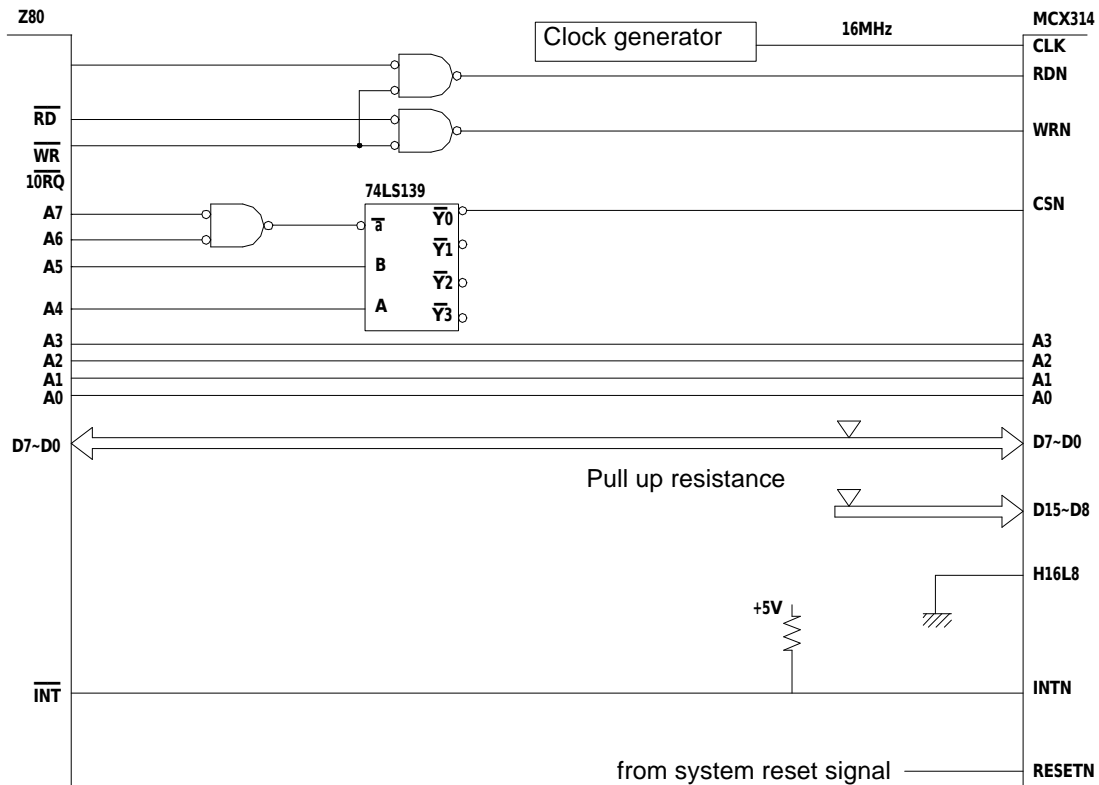
After the bit D15 of WR5 is set to 1 in bit pattern interpolation, the stack counter (SC) is changed from 2 to 1, and the interrupt will be generated. In continuous interpolation, when the bit D14 of WR5 is set to 1, the interrupt will be generated when it is ready to write the interpolation data for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

10. Connection Examples

10.1 Connection Example for 68000 CPU

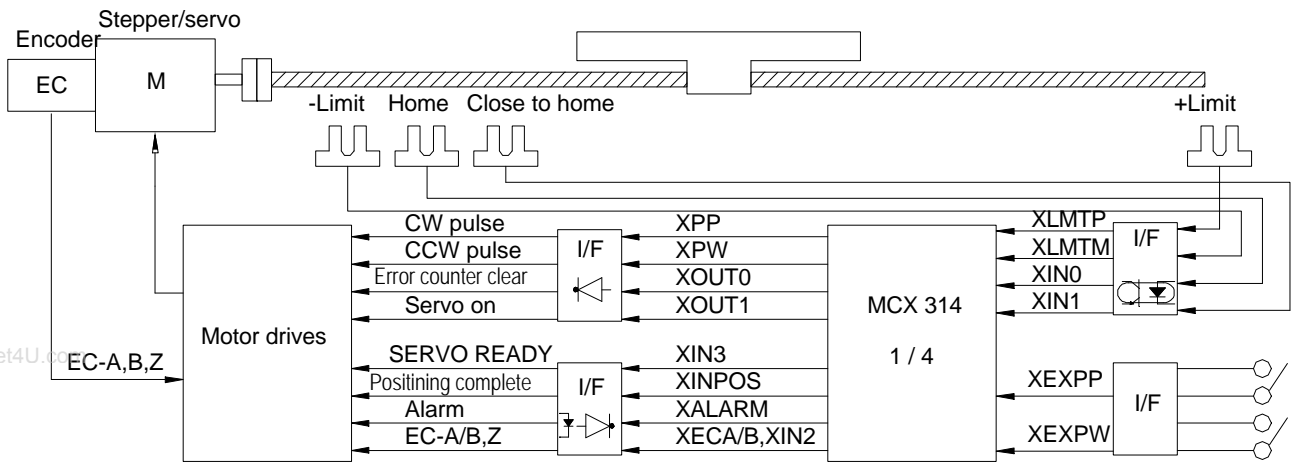


10.2 Connection Example for Z80 CPU



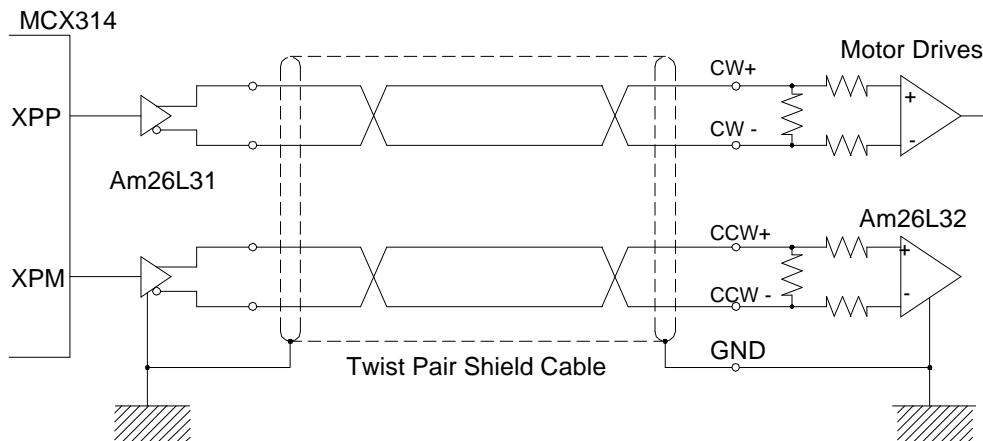
10.3 Connection Example

The figure shown below illustrates the example of 1-axis driving system. 4 axes can be assigned in the same way.

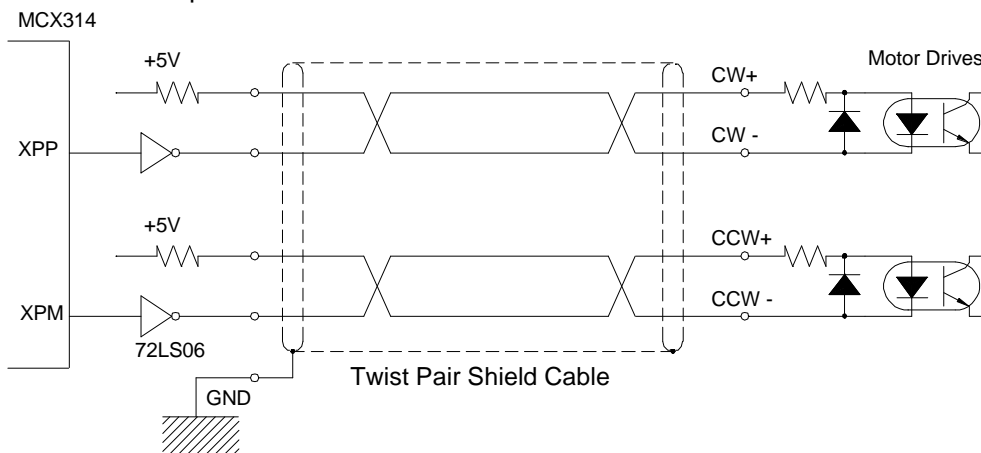


10.4 Pulse Output Interface

■ Output to Motor Drivers in Differential Circuit



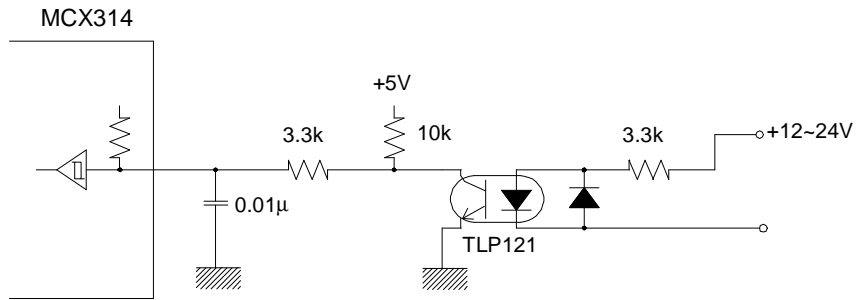
■ Open Collector TTL Output



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

10.5 Connection Example for Input Signals

In MCX314, except (D15 ~ D0), all the input signals are internal pull high and buffered by Smith trigger.

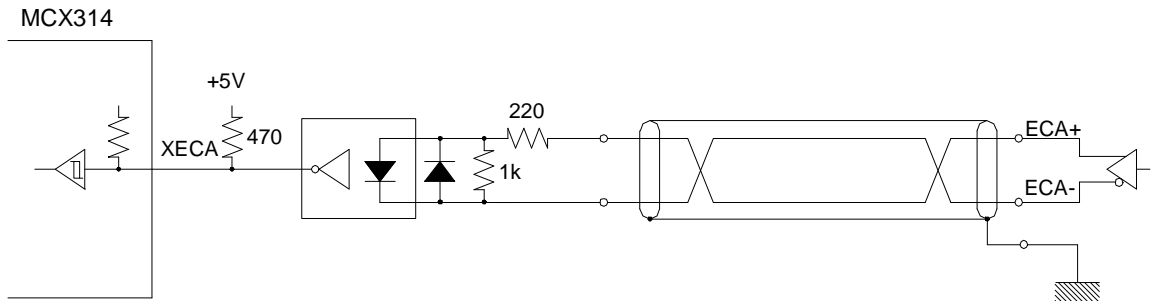


The response time of this loop is about 0.2 ~ 0.4 mSEC.

www.DataSheet4U.com

10.6 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 MCX314.



11. Example Program

The example C program for MCX314 can be compiled by Turbo C++ 4.0 and the operation system DOS/V.

```

#include <stdio.h>
#include <conio.h>
// ----mcx314 register address definition ----
#define adr 0x280 //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 first axis
control
#define bp1m 0x6 //BP - direction data register for the first axis
control
#define bp2p 0x8 //BP + direction data register for the second axis
control
#define bp2m 0xa //BP - direction data register for the second axis
control
#define bp3p 0xc //BP + direction data register for the third axis
control
#define bp3m 0xe //BP - direction data register for the third axis
control

// wreg 1 (axis assignment, data) ----Write register 1 setting
void wreg1(int axis, int wdata)
{
    int a;
    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)
{
    int a;
    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)
{
    int a;
    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)
{
    int a;
    outpw(adr+wr0, (axis<<8)+cmd);
}
// range(axis assignment, data) ----For range (R) setting
void range(int axis, long wdata)
{
    int a;
    outpw(adr+wr7,(wdata>>16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis<<8) + 0x00);
}
// acac(axis assignment, data) ----For S-curve acceleration (K) setting
void acac(int axis, long wdata)
{
    int a;
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis<<8) + 0x01);
}
// acc(axis assignment, data) ----For acceleration/deceleration (A)
setting
void acc(int axis, long wdata)
{
    int a;
    outpw(adr+wr6,wdata);
    outpw(adr+wr0, (axis<<8) + 0x02);
}
// dec( axis assignment, data) ----For deceleration (D) setting
void dec(int axis, long wdata)
{
    int a;
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis<<8) + 0x03);
}
// startv(axis assignment, data) ----For initial speed (SV) setting
void startv(int axis, long wdata)
{
    int a;
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis<<8) + 0x04);
}
// speed(axis assignment, data) ----For drive speed (V) setting
void speed(int axis, long wdata)
{
    int a;
    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)
{
    int a;
    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)
{
    int a;
    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)
{
    int a;
    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)
{
    int a;
    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)
{
    int a;
    outpw(adr+wr7, (wdata>>16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis<<8) + 0x0a);
}
// comp(axis assignment, data) ----For COMP+ (CP) setting

```


MCX314

```

void compm(int axis, long wdata)
i a
outpw(adr+wr7, (wdata>>16) & 0xffff);
outpw(adr+wr6, wdata & 0xffff);
outpw(adr+wr0, (axis<<8) + 0x0b);
i b
// compm(axis assignment, data) -----For COMP- (CM) setting
void compm(int axis, long wdata)
i a
outpw(adr+wr7, (wdata>>16) & 0xffff);
outpw(adr+wr6, wdata & 0xffff);
outpw(adr+wr0, (axis<<8) + 0x0c);
i b
// accofst(axis assignment, data) ----For acceleration counter shift (AO)
setting
void accofst(int axis, long wdata)
i a
outpw(adr+wr7, (wdata>>16) & 0xffff);
outpw(adr+wr6, wdata & 0xffff);
outpw(adr+wr0, (axis<<8) + 0x0d);
i b
// readlp(axis assignment) -----For logical position counter (LP) reading
void readlp(int axis)
i a
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);
i b
// readep(axis assignment) -----For real position counter (EP) reading
void readep(int axis)
i a
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);
i b
// wait(axis assignment) -----For waiting for drive stop
void wait(int axis)
i a
while(inpw(adr+rr0) & axis);
i b
// next_wait() -----Next data setting of waiting for continuous interpolation
void next_wait(void)
i a
while((inpw(adr+rr0) & 0x0200)==0x0);
i b
// bp_wait() ----- Next data setting of waiting for BP interpolation
void bp_wait(void)
i a
while((inpw(adr+rr0) & 0x06000)==0x6000);
i b
// homesrch()-----for homeal point searching on whole axis
void homesrch(void)
i a // [Action] The same for (1) ~ (3) axes
wreg1(0xf, 0x0008); // If homeal signal (in 1) = OFF
speed(0xf, 2000); // -direction motion will be continuously driven at a
speed of 20000 PPS
if((inpw(adr+rr4) & 0x2)==0x2)
// If IN1 signal = ON, deceleration will be stopped.
i a
command(0x1, 0x23);
i b
if((inpw(adr+rr4) & 0x200)==0x200)
i a
command(0x2, 0x23);

```

```

i b
if((inpw(adr+rr5) & 0x2)==0x2)
i a
command(0x4, 0x23);
i b
if((inpw(adr+rr5) & 0x200)==0x200)
i a
command(0x8, 0x23);
i b
wait(0xf);
wreg1(0xf, 0x000c); //(2) + direction motion will be continuously driven at
a speed of 500 PPS
speed(0xf, 50); //When IN1 = OFF, it will be suddenly
stopped
command(0xf, 0x21);
wait(0xf);

wreg1(0xf, 0x0000); //(3) - direction motion will be driven at a speed of
40000 PPS
speed(0xf, 40000); //100-pulse shift
pulse(0xf, 100);
command(0xf, 0x21);
wait(0xf);

lp(0xf, 0); // (4) LP = 0 for X, Y, Z, and U axes
wreg2(0xf, 0x0003); //Software limit on X, Y, Z axes: ON
compp(0x1, 100000); //X: -1000 ~ +100000
compm(0x1, -1000);
compp(0x2, 50000); //Y: -500 ~ +50000
compm(0x2, -500);
compp(0x4, 10000); //Z: -100 ~ +10000
compm(0x4, -100);

//*****<<Main Program>>*****
void main(void)
i a
int count;
outpw(adr+wr0, 0x8000); //Software reset
for(count = 0; count<2; ++count);

command(0xf, 0xf); //-----Whole axis 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, 0x0024); //Interpolation mode register
00000000 00100100

//-----Initial value settings for whole axis driving
accofst(0xf,0); //AO = 0
range(0xf, 800000) //R = 800000 (Multiple = 10)
acac(0xf, 1010); //K = 1010 (Jerk = 619 KPPS/SEC2)
acc(0xf, 100); //A = 100 (Acceleration/deceleration=125
KPPS/SEC)
dec(0xf, 100); //D (Deceleration = 125 KPPS/SEC)
startv(0xf, 100); //SV = 100 (Initial Speed = 1000 PPS)
speed(0xf, 4000); //V = 40000 (Drive Speed = 40000 PPS)
pulse(0xf, 100000); //P = 100000 (output pulse number = 100000)
lp(0xf, 0); //LP = 0 (Logical position counter = 0)

homesrch( ); //-----Home searching on whole axis-----

//-----Trapezoidal driving at X and Y axes
acc(0x3, 200); (Acceleration/deceleration = 250KPPS/SEC)
speed(0x3, 4000); //V = 4000(Drive Speed = 4000 PPS)

```

```

pulse(0x1, 8000); //xP = 8000
pulse(0x2, 40000); //yP = 40000
command(0x3, 0x20); //- direction constant drive
wait(0x3); //Wait for drive stop

wreg3(0x3, 0x0004); //S-curve acceleration/deceleration driving at X and
Y axes
acac(0x3, 1010); //K = 1010 (S-curve Acceleration = 619 KPPS/SEC2)
acc(0x3, 200); //A = 200 (Acceleration/Deceleration = 250 KPPS/SEC)
speed(0x3, 4000); //V = 4000 (Drive Speed = 4000 PPS)
pulse(0x1, 50000); //xP = 50000
pulse(0x2, 25000); //yP = 25000
command(0x0, 0x21); //- direction constant drive
wait(0x3);
wreg3(0x3, 0x0000); //S-curve acceleration/deceleration mode clearing
//----Linear interpolation drive at X and Y axis
outpw(adr+wr5, 0x0124); //ax1 = x, ax2 = y, ax3 = z, linear speed keeps
constant
range(0x1, 800000); //ax1/R = 800000 (Multiple = 10)
range(0x2, 1131371); //ax1/R = 800000 x 1.414
speed(0x1, 100); //Drive Speed = 1000 PPS 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)
//----Circular interpolation drive at X and Y axes ----
outpw(adr+wr5, 0x0124); //ax1 = x, ax2 = y, ax3 = z, linear speed keeps
constant
range(0x1, 800000); //ax1/R = 800000 (Multiple = 10)
range(0x2, 1131371); //ax2/R = 800000 x 1.414
speed(0x1, 100); //Drive Speed = 1000 PPS constant speed
center(0x1, -5000); //xC = 5000 (Center of X = -5000)
center(0x2, 0); //yC = 0 (Center of Y = 0)
pulse(0x1, 0); //xP = 0 (Finish point of X = 0) Circle
pulse(0x2, 0); //yP = 0 (Finish point of Y = 0)
command(0x0, 0x33); //CCW circular interpolation
wait(0x3);
//---- Bit pattern interpolation at X and Y axes----//(Example of figure
2.24)
speed(0x1,1); //Drive Speed = 10 PPS constant
speed
command(0, 0x36); //Bit pattern data writing permission

outpw(adr+bp1p, 0x0000); //0 ~ 15 bits data writing
outpw(adr+bp1m, 0x2bff);
outpw(adr+bp2p, 0xffd4);
outpw(adr+bp2m, 0x0000);
command(0, 0x38); //Stack

outpw(adr+bp1p, 0xf6fe); //16 ~ 31 bits data writing
outpw(adr+bp1m, 0x0000);
outpw(adr+bp2p, 0x000f);
outpw(adr+bp2m, 0x3fc0);
command(0, 0x38);

outpw(adr+bp1p, 0x1fdb); //32 ~ 47 bits data writing
outpw(adr+bp1m, 0x0000);
outpw(adr+bp2p, 0x00ff);
outpw(adr+bp2m, 0xfc00);
command(0, 0x38);

command(0, 0x38); //2-axis BP interpolation drive starting

bp_wait( ); //Wait for data writing

outpw(adr+bp1p, 0x1fdb); //48 ~ 63 bit3 data writing
outpw(adr+bp1m, 0x0000);
outpw(adr+bp2p, 0x00ff);
outpw(adr+bp2m, 0xfc00);
command(0, 0x38);

command(0,0x37); // Bits pattern data writing inhibition

wait(0x3); //Wait for drive stop

//---- Continuous interpolation at X and Y axes (Example of figure
2.29)
speed(0x1, 100); //Drive Speed = 10 PPS constant speed

pulse(0x1, 4500); // node 1
pulse(0x2, 0);
command(0, 0x30);

next_wait(); //Wait for next data setting
center(0x1,0); // node 2
center(0x2, 1500);
pulse(0x1, 1500);
pulse(0x2, 1500);
command(0, 0x33);

next_wait();
pulse(0x1, 0); // node 3
pulse(0x2, 1500);
command(0, 0x30);
next_wait();
center(0x1, -1500); // node 4
center(0x2, 0);
pulse(0x1, -1500);
pulse(0x2, 1500);
command(0, 0x33);
next_wait(); // node 5
pulse(0x1, -4500);
pulse(0x2, 0);
command(0, 0x30);

next_wait(); // node 6
center(0x1, 0);
center(0x2, -1500);
pulse(0x1, -1500);
pulse(0x2, -1500);
command(0, 0x33);

next_wait(); //node 7
pulse(0x1, 0);
pulse(0x2, -1500);
command(0, 0x30);

next_wait(); //node 8
center(0x1, 1500);
center(0x2, 0);
pulse(0x1, 1500);
pulse(0x2, -1500);
command(0, 0x33);

wait(0x3);
i b

```

12. Electrical Characteristics

12.1 DC Characteristics

■ Absolute Maximum Rated

Item	Symbol	Value	Unit
Power Voltage	VDD	-0.3 ~ +7.0	V
Input voltage	VIN	-0.3 ~ V _{DD} +0.3	V
Input Current	IIN	± 10	mA
Reservation Temperature	TSTG	-40 ~ +125	°C

■ Recommend Operation Environment

Item	Symbol	Value	Unit
Power Voltage	VDD	4.75 ~ 5.25	V
Ambient Temperature	Ta	0 ~ +85	°C

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

■ DC Characteristics

(Ta = 0 ~ +85°C, V_{DD} = 5V ; 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} =0V	-10		10	μA	D15~D0 Input signal
		V _{IN} =0V	-200		-10	μA	Input signal besides D15~D0
High level output voltage	V _{OH}	I _{OH} =-1μA	V _{DD} -0.05			V	Note 1
		I _{OH} =-4 mA	2.4			V	Output Signal besides D15~D0
		I _{OH} =-8 mA	2.4			V	D15~D0 Output signal
Low level output voltage	V _{OL}	I _{OL} =1μA			0.05	V	
		I _{OL} =4 mA			0.4	V	Output signal besides D15~D0
		I _{OL} =8 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 mA, CLK=16 MHz		52	90	mA	

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

■ Pin Capacity

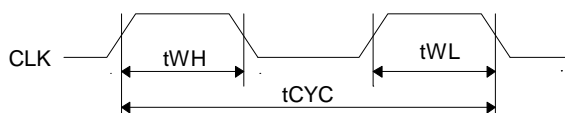
Item	Mark	Condition	Min.	Typ.	Max.	Unit	Remark
Input/ Output capacity	C _{IO}	Ta=25°C			10	pF	D15 ~ D0
Input capacity	C _I	f=1 MHz			10	pF	Other input pins

12.2 AC Characteristics

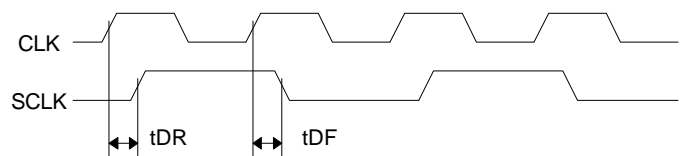
(Ta = 0 ~ +85°C, V_{DD} = 5V ± 5%, Output load condition: 85 pF + 1 TTL)

12.2.1 Clock

■ CLK Input Pulse



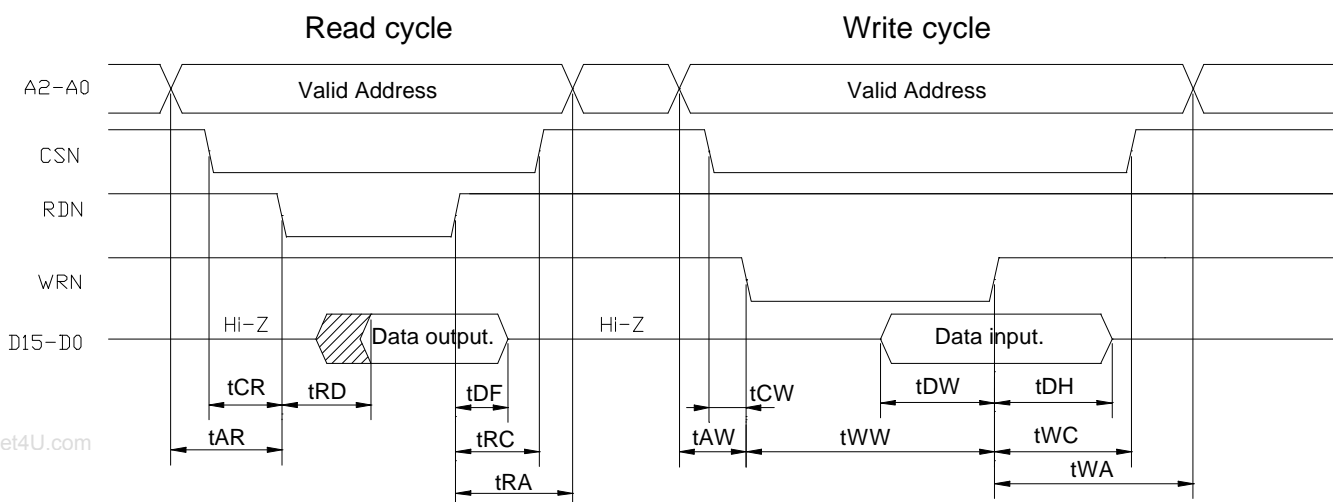
■ SCLK Output Signal



SCLK will not be output during reset.

Symbol	Item	Min.	Max.	Unit
t _{CYC}	CLK Cycle	62.5		nS
t _{WH}	CLK Hi Level Wavelength	20		nS
t _{WL}	CLK Low Level Wavelength	20		nS
t _{DR}	CLK↑ → SCLK↑ Delay Time		21	nS
t _{DF}	CLK↑ → SCLK↓ Delay Time		23	nS

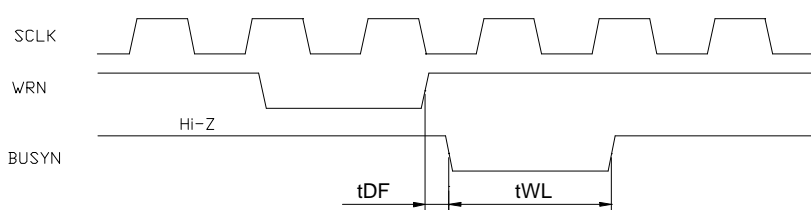
12.2.2 Read / Write Cycle



The figure shown above is used for 16-bit data bus accessing (H16L8 = Hi). For 8-bit data bus (H16L8 = Low), the address signals shown in the figure become A3~A0, and data signals become D7~D0.

Symbol	Item	Min.	Max.	Unit
tAR	Address SETUP Time (to RDN ↓)	0		nS
tCR	CSN SETUP Time (to RDN ↓)	0		nS
tRD	Output Data Delay Time (from RDN ↑)		29	nS
tDF	Output Data Reservation Time (from RDN ↑)	0	30	nS
tRC	CSN Reservation Time (from RDN ↑)	0		nS
tRA	Address Reservation Time (from RDN ↑)	0		nS
tAW	Address SETUP Time (to WRN ↓)	0		nS
tCW	Established Time for CSN (to WRN ↓)	0		nS
tWW	WRN Low Level Wavelength	50		nS
tDW	Established Time for Input Data (to WRN ↑)	30		nS
tDH	Reservation Time for Input Data (from WRN ↑)	10		nS
tWC	CSN Reservation Time (from WRN ↑)	5		nS
tWA	Address Reservation Time (from WRN ↑)	5		nS

12.2.3 BUSYN Signal



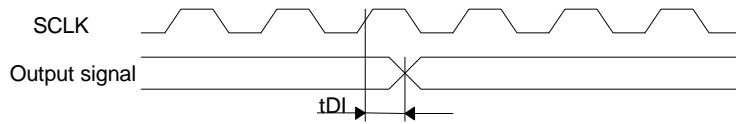
It is low when BUSYN is active. And BUSYN is low after 2 SCLK cycles when WRN ↑ active.

Mark	Item	Min.	Max.	Unit
tDF	WRN ↑ → BUSYN ↓ Delay Time		32	nS
tWL	BUSYN Low Level Wavelength		tCYC x 4+30	nS

tCYC is a cycle of CLK.

12.2.4 SCLK/Output Signal Timing

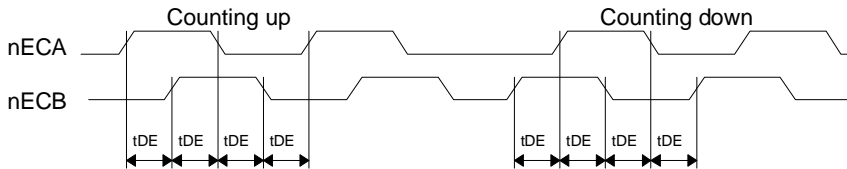
The following output signal is synchronized with SCLK output signal. The level at ACLK ↑ will be changed. Output signals : nPP/PLS, nPM/DIR, nDRIVE, nASND, nDSND, nCMPP, and nCMPPM.



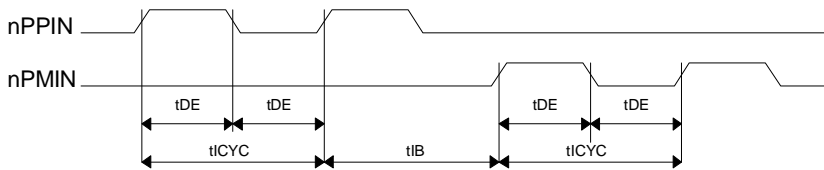
Mark	Item	Min.	Max.	Unit
tDD	SCLK ↑ → Output Signal ↑ ↓ Delay Time	0	20	nS

12.2.5 Input Pulses

■ Quadrature Pulses Input Mode (A/B phases)



■ Up/Down Pules Input Mode



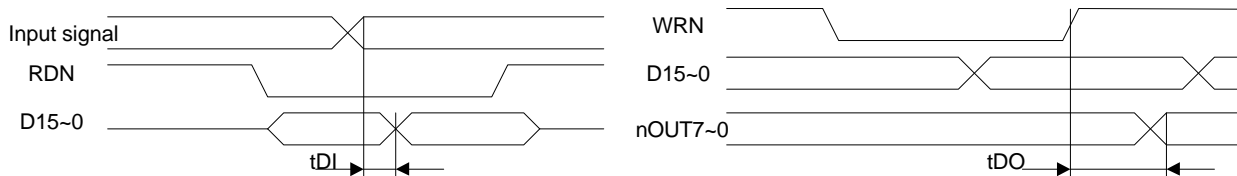
- In A/B quadrature pulse input mode, when nECA and nECB input pulses are changed, the value of real position counter will be changed to the value of those input pulses changed after the period of longest SCLK4 is passed.
- In UP/DOWN pulse input mode, the real position counter will become the value of those input pulses changed, after the period between the beginning of nPPIN, nPMIN ↑ and the time of SCLK 4 cycle is passed.

Mark	Item	Min.	Max.	Unit
tDE	nECA and nECB Phase Difference Time	$t_{CYC} \times 2 + 20$		nS
tIH	nPPIN and nPMIN Hi Level Wavelength	30		nS
tIL	nPPIN and nPMIN Low Level Wavelength	30		nS
tiCYC	nPPIN and nPMIN Cycle	$t_{CYC} \times 2 + 20$		nS
tIB	nPPIN ↑ ↔ nPMIN ↑ Time	$t_{CYC} \times 2 + 20$		nS

12.2.6 General Purpose Input / Output Signals

The figure shown at the lower left hand side illustrates the delay time when input signals nIN3 ~ 0, nEXPP, nEXPM, nINPOS, and nALARM are read through RR4 and RR5 registers.

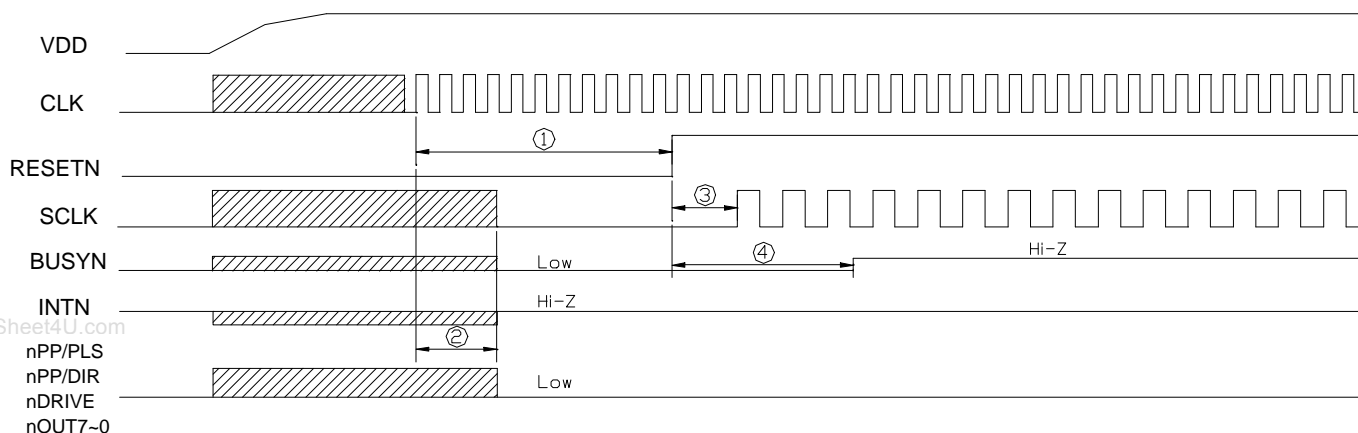
The figure shown at the lower right hand side illustrates the delay time when writing general output signal data into nWR3 and nWR4.



Mark	Item	Min.	Max.	Unit
tDI	Input Signal → Data Delay Time		32	nS
tDO	WRN ↑ → nOUT7~0 Established Time		32	nS

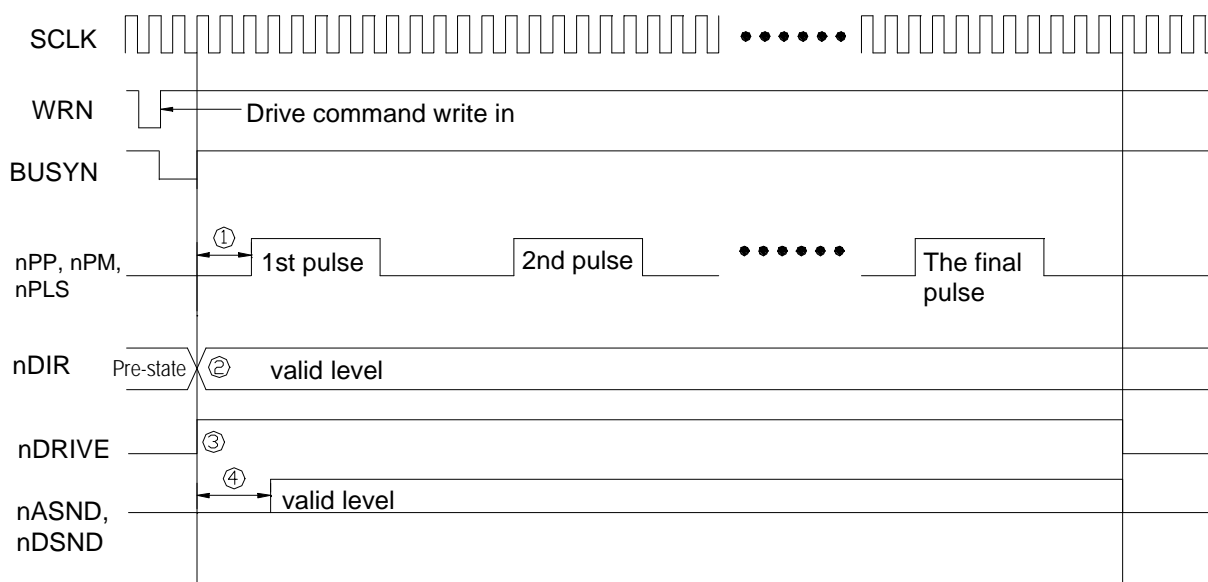
13. Timing of Input / Output Signals

13.1 Power-On Reset



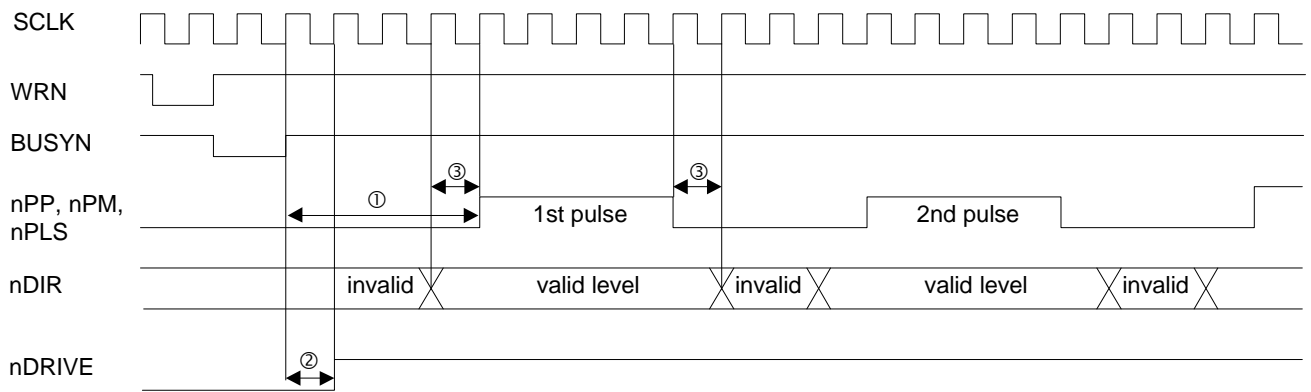
- ① The reset signal input to pin RESETN will keep on the Low level for at least 4 CLK cycles.
- ② When RESETN is on the Low level for 4 CLK cycles maximum, the output signals of MCX314 are decided.
- ③ SCLK will be output after 2 CLK cycles when RESTN return to the Hi level.
- ④ BUSYN keeps on the Low level for 8 CLK cycles when RESTN is on the Hi level.

13.2 Fixed Pulse or Continuous Driving



- ① This first driving pulses (nPP, nPM, and nPLS) will be output after 3 SCLK cycles when BUSYN is ↑.
- ② The nDIR (direction) signal is valid after 1 SCLK cycle when BUSYN is ↑.
- ③ The dDRIVE becomes Hi level when BUSYN is ↑.
- ④ The nASND and nDSND are on invalid level after 3 SCLK cycles when BUSYN is ↑.

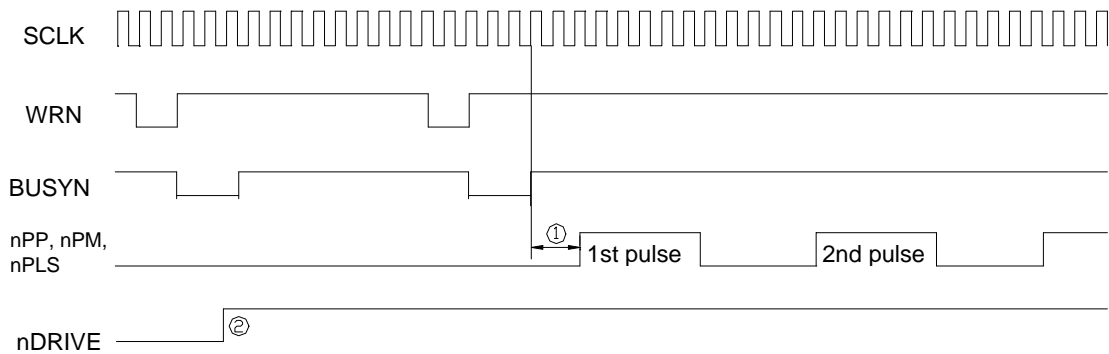
13.3 Interpolation



www.DataSheet4U.com

- ① The first pulses (nPP, nPM, and nPLS) of interpolation driving will be output after 4 SCLK cycles when BUSYN is \uparrow .
- ② nDRIVE will become Hi level after 1 SCLK cycle when BUSYN is \uparrow .
- ③ DIR signal keeps the active level in 1 SCLK cycle before and after the Hi level pulse outputting.

13.4 Start Driving after Hold Command



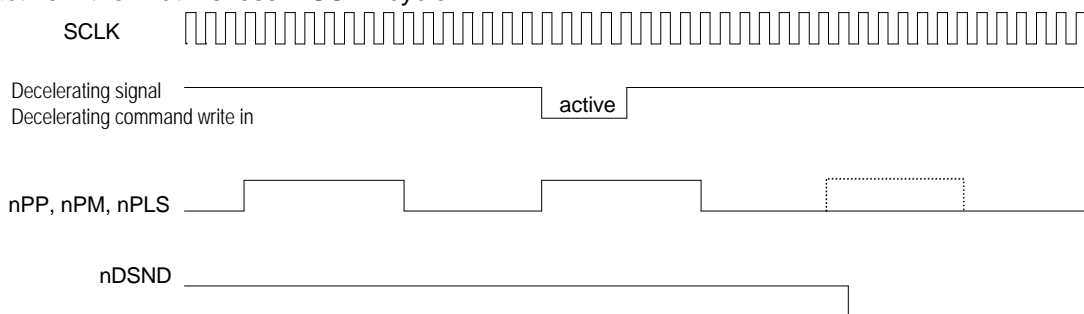
- ① The pulses (nPP, nPM, and nPLS) of each axis will start outputting after 3 SCLK cycles when BUSYN is \uparrow .
- ② nDRIVE will become Hi level when BUSYN is \uparrow for each axis.

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.

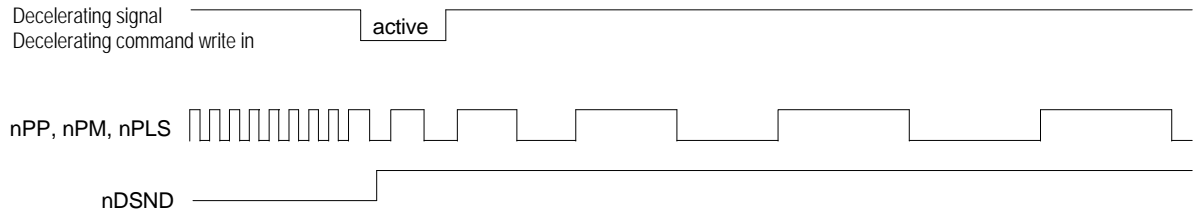
The width of external signals input for sudden stop must be more than 1 SCLK cycle. The stop function will not be active if the width is less 1 SCLK cycle.



13.6 Decelerating Stop

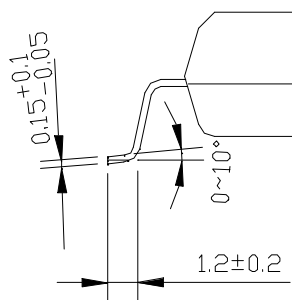
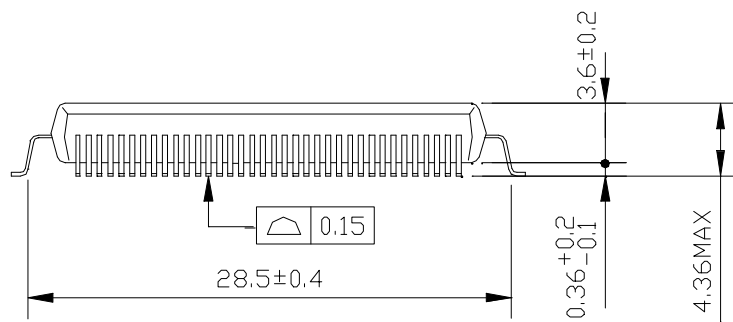
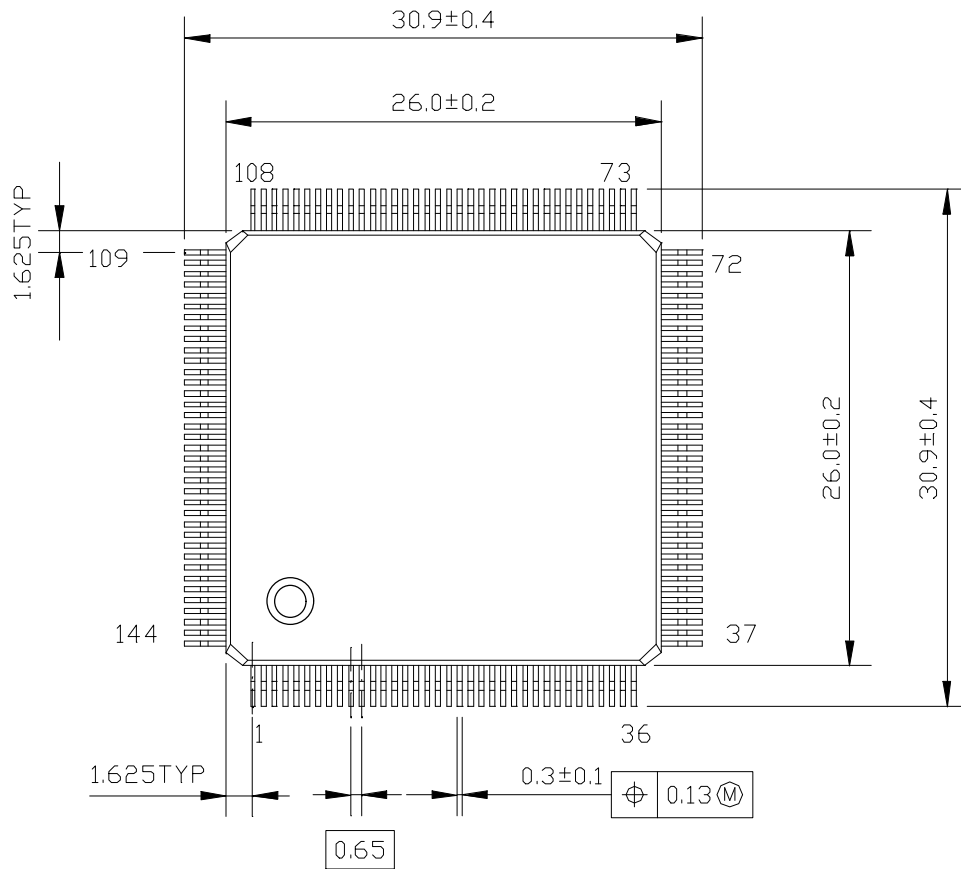
The following figure illustrates the timing of decelerating stop input signal and decelerating commands. The decelerating stop signal are nIN3 ~ 0 and nLMTP/M (When the decelerating mode is engaged)

When speed decelerating signals become active, or the decelerating stop command is written, the decelerating stop function will be performed.



14. Pinout

Unit: mm



15. Specifications

- Control Axis 4 axes
- Data Bus 16/8 bits selectable

Interpolation Functions

■ 2-axes / 3-axes Linear Interpolation

Interpolation Range	Each axis -8,388,608 ~ +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 -8,388,608 ~ +8,388,607
Interpolation Speed	1 ~ 4 MPPS
Interpolation Accuracy	± 1 LSB (Within the range of whole interpolation)

■ 2 axes / 3 axes Bit Pattern Interpolation

Interpolation Speed	1 ~ 4 MPPS (Dependent on CPU data writing time)
---------------------	-------------------------------------------------

■ Related Functions of Interpolation

- Can select any axis ● Constant vector speed ● Continuous interpolation
- Single step interpolation (Command/external signals)

Common Specifications of Each Axis

■ Drive Pulses Output (When CLK = 16 MHz)

Pulse Output Speed Range	1PPS ~ 4MPPS
Pulse Output Accuracy	within ± 0.1% (according to the setting speed)
S-curve Jerk	954 ~ 31.25; 10^9 PPS/S ²
Accelerating / Decelerating Speed	125 ~ 500; 10^6 PPS/S
Drive Speed	1 ~ 4; 10^6 PPS
Output-pulse Number	0 ~ 268435455 / unlimited
Speed Curve	quadrature / trapezoidal / parabolic S-curve
Index Drive Deceleration Mode	auto / 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 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 t) range	-2,147,483,648 ~ +2,147,483,647
Real Position Counter (for feedback pulse) range	-2,147,483,648 ~ +2,147,483,647
Data read and write possible	

- Comparison Register
 - COMP + Register Position comparison range -2,147,483,648 ~ +2,147,483,647
 - COMP – Register Position comparison range -2,147,483,648 ~ +2,147,483,647
 - 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; ~~the~~ volume of COMP-
 - ..the volume of position counter; ~~the~~ volume of COMP-
 - ..the volume of position counter; ~~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

- External Deceleration / Sudden Stop Signal
 - IN0 ~ 3 4 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 Output Signal
 - OUT0 ~ 7 8 points for each axis (wherein 4 points use with drive status output signal pin)

- Driving Status Signal Output
 - ASND (speed accelerating), DSND (speed decelerating),
 - CMPP (position; ~~C~~OMP+), CMPM (position; ~~C~~OMP-)
 - Drive status and status registers readable

- Limit Signals Input
 - 2 points, for each + and - side
 - Logical levels and decelerating / sudden stop selectable

- Emergency Stop Signal Input
 - EMG, 1 point for 4 axes

- Electrical Characters
 - Temperature Range for Driving 0 ~ +45°C (32°F ~ 185°F)
 - Power Voltage for Driving ± 5V ± 5 % (max. power consumption: 90mA)
 - Input / Output Signal Level CMOS, TTL connectable
 - Input Clock Pulse 16,000 MHz (Standard)

- Package
 - 144-pin plastic QFP, pitch = 0.65mm
 - Dimension : 30.9 x 30.9 x 4.36 mm

Appendix A: Speed Curve Profile

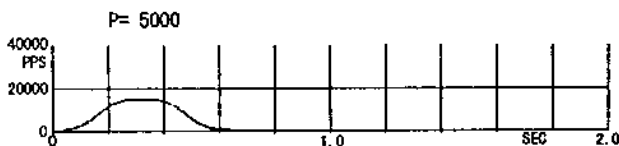
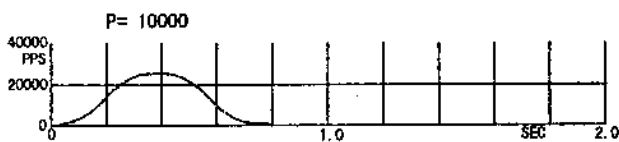
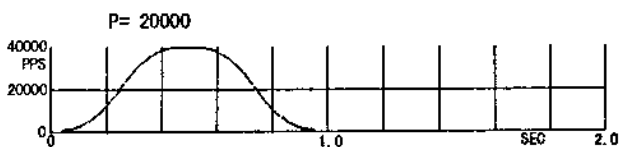
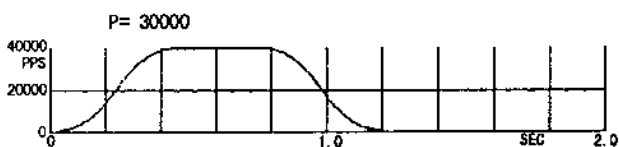
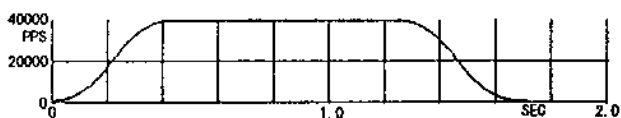
The following curves are based on the test records from MCX314 output drive pulses and speed curve traces.

The complete 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 Full S-curve Acceleration / Deceleration
R=800000 (Rate:10), K=700, A=D=200, SV=100, V=4000, A0=50
 Auto Decelerating Mode
 Jerk=893KPPS/ SEC²
 Accel. / Decel. =250KPPS/SEC
 Initial Speed=1000PPS
 Drive Speed=40KPPS

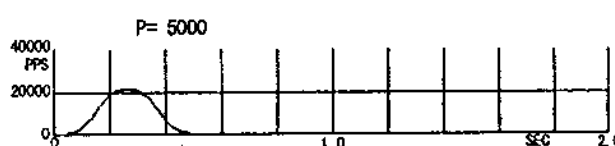
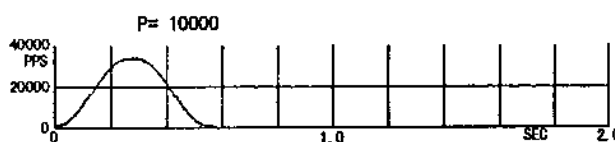
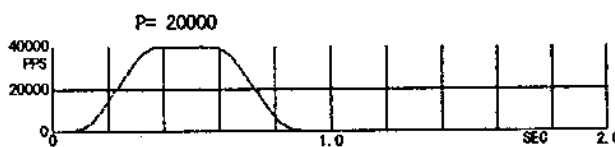
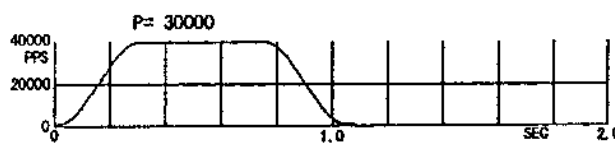
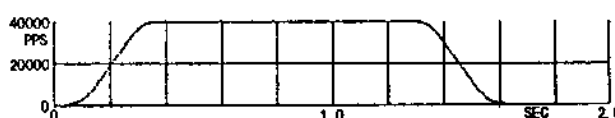
www.DataSheet4U.com

Output Pulse P=50000



■ 40KPPS Partial S-curve Acceleration / Deceleration
R=800000 (Rate:10), K=300, A=D=150, SV=100, V=4000, A0=20
 Auto Decelerating Mode
 Jerk=2083KPPS/ SEC²
 Accel. / Decel. =188KPPS/SEC
 Initial Speed=1000PPS
 Drive Speed=40KPPS

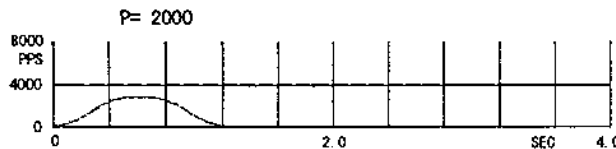
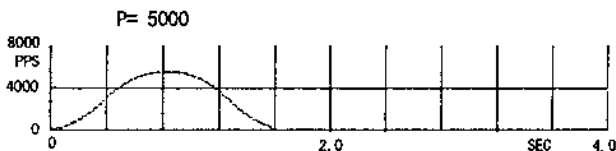
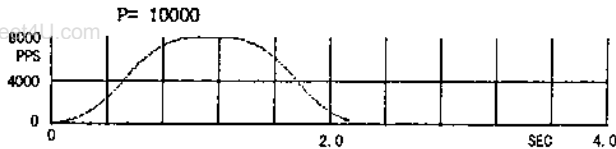
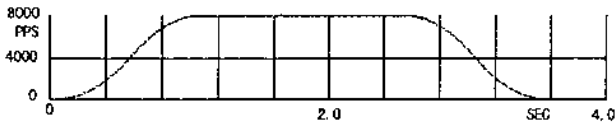
Output Pulse P=50000



■ 8000PPS complete S-curve Acceleration / Deceleration
 R=8000000 (Rate:1), K=2000, A=D=500, SV=100, V=8000, A0=0
 Auto Decelerating Mode

Jerk=31KPPS/ SEC²
 Accel. / Decel. =62.5KPPS/SEC
 Initial Speed=100PPS
 Drive Speed=8000PPS

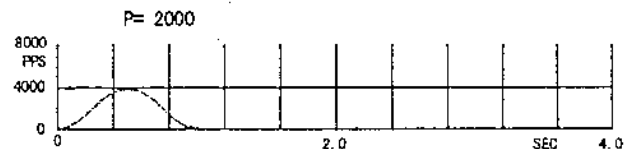
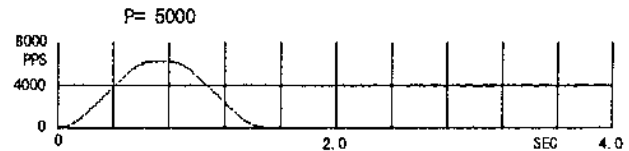
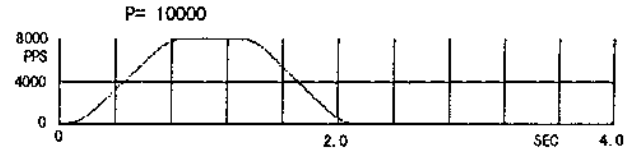
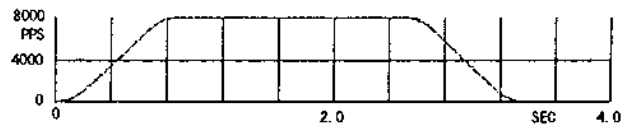
Output Pulse P=20000



■ 8000PPS Partial S-curve Acceleration / Deceleration
 R=800000 (Rate: 1), K=1000, A=D=100, SV=100, V=8000, A0=0
 Auto Decelerating Mode

Jerk=62.5KPPS/ SEC²
 Accel. / Decel. =12.5KPPS/SEC
 Initial Speed=100PPS
 Drive Speed=8000PPS

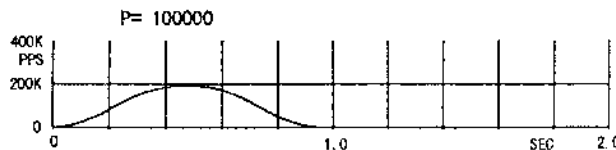
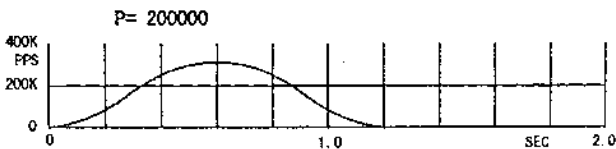
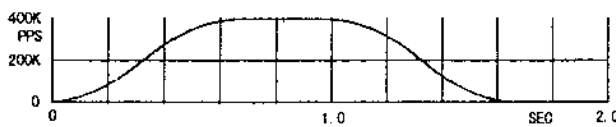
Output Pulse P=20000



■ 400KPPS Full S-curve Acceleration / Deceleration
 R=80000(Rate:100),K=2000,A=D=100,SV=10,V=4000, A0=1000
 Auto Decelerating Mode

Jerk=3.13MPPS/SEC²
 Accel. / Decel. =1.25MPPS/SEC
 Initial Speed=1000PPS
 Drive Speed=400KPPS

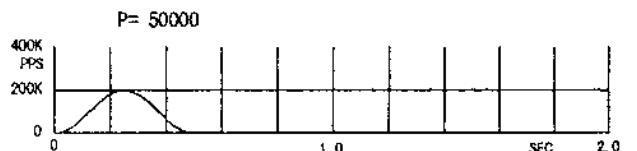
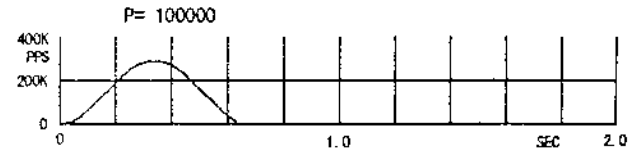
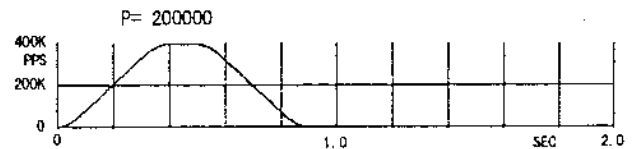
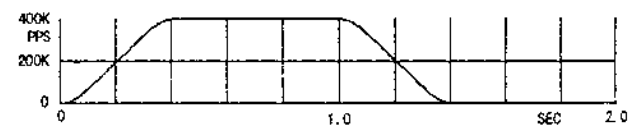
Output Pulse P=400000



■ 400KPPS Partial S-curve Acceleration / Deceleration
 R=80000 (Rate: 100), K=500, A=D=100, SV=10, V=4000, A0=0
 Auto Decelerating Mode

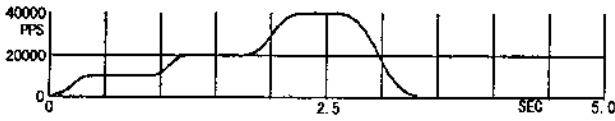
Jerk=12.5MPPS/ SEC²
 Accel. / Decel. =1.25MPPS/SEC
 Initial Speed=1000PPS
 Drive Speed=400KPPS

Output Pulse P=400000



■ Speed Alterations in a Continuous S-curve Acceleration / Deceleration Drive
 R=800000 (Rate: 10), K=2000, A=D=100, SV=100, V=1000; 2000; 4000; decelerating stop

Jerk=312.5KPPS/ SEC²
 Jerk(decel.)=125KPPS/SEC
 Initial Speed=1000PPS
 Drive Speed=10K; 20K; 40KPPS



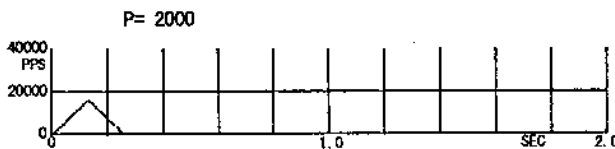
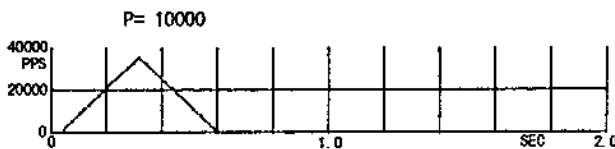
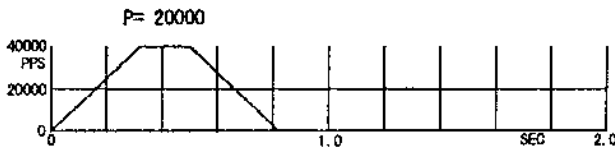
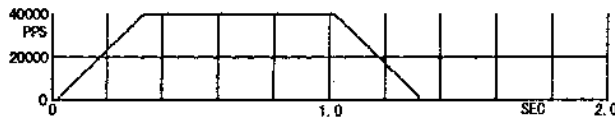
■ 40KPPS Linear Acceleration / Deceleration

www.DataSheet4U.com

R=800000 (Rate: 10), A=D=100, SV=100, V=4000
 WR3/D2, D1, D0=0, 0, 0
 Trapezoidal Mode, Auto Decelerating Mode

Accel./ Decel. =125KPPS/ SEC
 Initial Speed=100PPS
 Drive Speed=40000PPS

Output Pulse P=40000



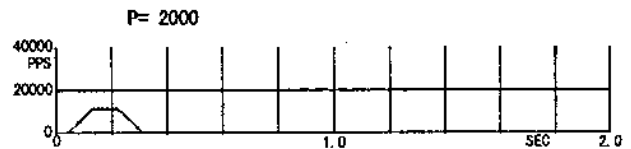
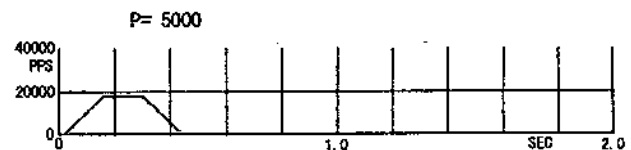
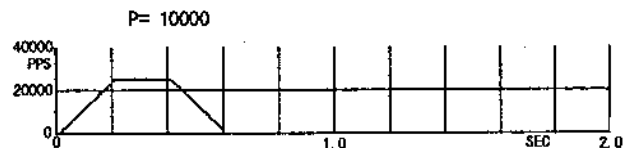
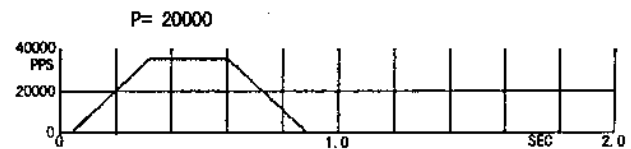
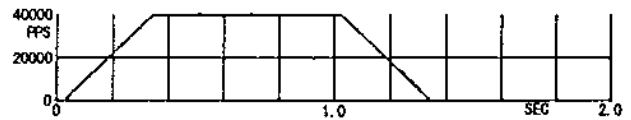
■ Triangle wave Avoiding during the Linear Acceleration / Deceleration

When K=1 (Jerk: Maxium) in the S-curve Accel./ Decel. Mode, "1/4 output pulses" principle can be engaged. Even the output pulses are few, the drive is a trapezoidal drive.

R=800000 (Rate: 10), K=1, A=D=100, SV=100, V=4000
 WR3/D2, D1, D0=1, 0, 0 S-curve Mode, Auto Decelerating Mode

Jerk =625MPPS/ SEC²
 Accel./ Decel. =125KPPS/SEC
 Initial Speed=1000PPS
 Drive Speed=40000PPS

Output Pulse P=40000

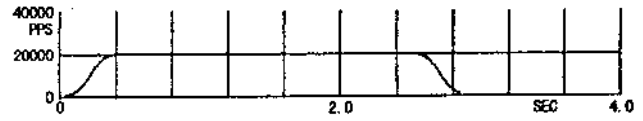
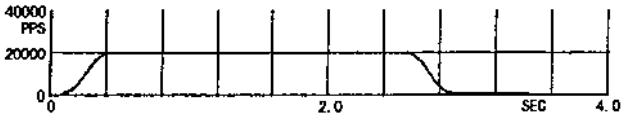


MCX314

- Initial Speed Trailing during the S-curve Accel./ Decel. fixed pulse drive
Try to adjust the volumes of parameters to have a more stable drive.

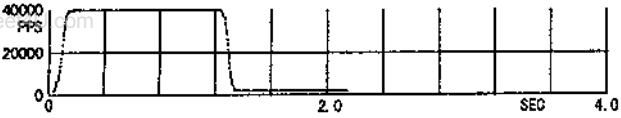
R=800000 (Rate: 10), K=1000, A=D=8000, SV=100, V=2000, P=50000
Auto Decelerating Mode

⇒ K=1010



R=800000 (Rate: 10), K+50, A=D=8000, SV=200, V=4000, A0=0, P=50000
Auto Decelerating Mode

⇒ K=53



www.DataSheet4U.com