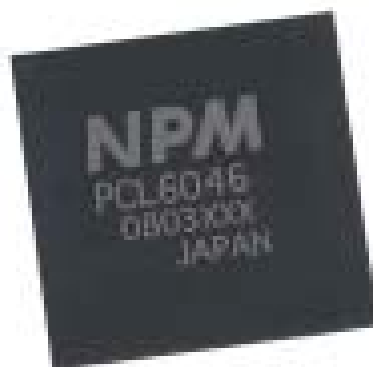




Pulse Control LSI

PCL6046

User's Manual



INDEX

1. Introduction	1
1.1 How to use this manual	1
1.1.1 Symbol description	1
1.1.2 Terminology	3
1.1.3 Notation	4
1.2 Handling the product	5
1.2.1 Storing	5
1.2.2 Unpacking	5
1.2.3 Safety	5
1.3 Product Warranty	9
1.3.1 Warranty period	9
1.3.2 Warranty scope	9
1.4 Notice	9
1.5 Confirmation	9
2. Outline	10
2.1 Features	10
2.2 Configuration	15
3. Specifications	16
4. Hardware description	17
4.1 External dimensions	17
4.2 Pin assignment diagram	18
4.3 Functions of pins	19
4.4 CPU interface	24
4.4.1 68000 interface	25
4.4.2 H8 interface	26
4.4.3 8086 interface	27
4.4.4 Z80 interface	28
5. Software description	29
5.1 CPU communication	29
5.1.1 Access method	29
5.1.2 Address map	31
5.1.3 Command writing	41
5.1.4 Register writing	43
5.1.5 Register reading	45
5.1.6 Main status reading	47
5.1.7 General-purpose output port writing	47
5.1.8 Sub-status & general purpose I/O port reading	48

5.2 Status & General-purpose I/O Port	49
5.2.1 Main status (MSTS).....	49
5.2.2 Sub status (SSTS) and general-purpose input/output port (IOP)	52
5.2.3 Extension status (RSTS)	53
5.2.4 Interpolation status (RIPS)	53
5.3 Commands	54
5.3.1 Operation commands	54
5.3.2 Control commands	56
5.4 Registers	60
5.4.1 Speed control registers	62
5.4.2 Position control register	69
5.4.3 Environment setting register.....	72
5.4.4 Counter register	93
5.4.5 Comparator register	95
5.4.6 Counter latch register	97
5.4.7 Interrupt control register	98
5.4.8 Status display register	104
5.5 Operation mode	108
5.5.1 Command control	109
5.5.2 Positioning control	109
5.5.3 Pulser control.....	111
5.5.4 Switch control	118
5.5.5 Origin return control.....	122
5.5.6 Sensor control	141
5.5.7 Linear interpolation 1 control	144
5.5.8 Linear interpolation 2 control	146
5.5.9 Circular interpolation control.....	148
5.5.10 U-axis synchronous control	151
6. Function description.....	153
6.1 Reset	153
6.1.1 Hardware reset	154
6.1.2 Software reset	154
6.2 Pre-register.....	155
6.2.1 Continuous operation	155
6.2.2 Continuous comparison.....	157
6.3 Speed control	159
6.3.1 Speed patterns	159
6.3.2 Speed setting example.....	160
6.3.3 Manual correction calculation of FH speed	161
6.3.4 Target speed override.....	165
6.3.5 Number of circular interpolation steps.....	166
6.3.6 Constant synthesized speed control	168
6.4 Position control.....	169
6.4.1 Target position override 1 (RMV).....	169

6.4.2 Target position override 2 (PCS)	172
6.4.3 End point draw operation	173
6.5 Output pulse control	174
6.5.1 Output pulse mode	174
6.5.2 Operation complete timing	175
6.5.3 Output pulse width control	176
6.6 Idling control	177
6.7 Mechanical external input control	178
6.7.1 End limit (+EL, -EL)	178
6.7.2 Slow-down (+SD, -SD)	180
6.7.3 Origin (ORG), Encoder Z phase (EZ)	183
6.8 Servo motor interface	185
6.8.1 Positioning complete (INP)	185
6.8.2 Deviation counter clear (ERC)	186
6.8.3 Alarm (ALM)	188
6.9 External start / Simultaneous start	189
6.9.1 Simultaneous start (CSTA)	189
6.9.2 Own axis start (STA)	191
6.9.3 Axis selection start (SELn)	192
6.10 External stop / Simultaneous stop	193
6.10.1 Simultaneous stop (CSTP)	193
6.10.2 Axis selection stop (SELn)	194
6.11 Emergency stop	195
6.12 Counter	196
6.12.1 Counter type and input specifications	196
6.12.2 Counter clear	200
6.12.3 Counter latch	202
6.12.4 Counter count stop and input stop	204
6.13 Comparator	206
6.13.1 Comparator types and functions	206
6.13.2 Software limit	214
6.13.3 Out-of-step detection of stepping motor	216
6.13.4 Index output	220
6.13.5 Ring count	222
6.13.6 Bulk override	224
6.14 Backlash correction	229
6.15 Slip correction	230
6.16 Vibration suppression	231
6.17 Synchronous start	232
6.17.1 Start by stopping the specified axis	232
6.17.2 Start with internal sync signal	240

6.18 Interrupt request (INT).....	244
6.18.1 Error interrupt	245
6.18.2 Event interrupt	246
6.18.3 Operation stop interrupt.....	247
6.19 General-purpose one shot	248
7. Electrical Characteristics	249
7.1 Absolute maximum ratings	249
7.2 Recommend operating conditions	249
7.3 DC characteristics	250
7.4 AC characteristics	250
7.4.1 Reference clock.....	250
7.4.2 CPU IF = 0 (68000)	251
7.4.3 CPU IF = 1 (H8).....	252
7.4.4 CPU IF = 2 (8086)	253
7.4.5 CPU IF = 3 (Z80)	254
7.5 Operation timing.....	255
7.5.1 RST signal	256
7.5.2 SRST command	256
7.5.3 EA, EB signals.....	256
7.5.4 PA, PB signals	257
7.5.5 Start command	257
7.5.6 CSTA signal	257

1. Introduction

Thank you for choosing our pulse control LSI, the "PCL6046."

This manual describes the specifications, functions, connections, and usages of PCL6046.

Be sure to read this manual thoroughly and keep it handy in order to use the product appropriately.

1.1 How to use this manual

1. Reproduction of this manual in whole or in part without permission is prohibited by the Copyright Act.
2. The contents of this manual are subject to change without the prior notice along with the improvement of performance and quality.
3. Although this manual is produced with the utmost care, please contact our sales representative if there are any questions, errors or omissions.

1.1.1 Symbol description

1.1.1.1 Physical damage level

In this manual, the physical damage level is defined as follows.

- Serious injury
Those that might cause aftereffects such as loss of sight, wound, burn, electric shock, fracture, poisoning, or those requiring hospitalization or long-term outpatient treatment.
- Minor injury
Those not requiring hospitalization or long-term outpatient treatment. (Other than "serious injury" above)

1.1.1.2 Hazardous level

The product is designed with the top priority for the safety of operators. However, due to the nature of the product, there are risks that cannot be eliminated. In this manual, the seriousness and level of these risks are divided into three categories: "Danger," "Warning," and "Caution." Be sure to read and understand the symbols descriptions thoroughly before operating or performing maintenance work on the product.

"Danger", "Warning", and "Caution" are indicated in the order of severity of hazard: (danger > warning > caution), and the meanings are described underneath.



D a n g e r

"Danger" indicates that it might cause an imminent risk that could result in the death or serious injury of the operator during operations of this product.




W a r n i n g

"Warning" indicates that it may result in the death or serious injury of the operator during operations of this product.

C a u t i o n

"Caution" indicates that it may result in minor injury of the operator during operations of this product.

C a u t i o n

"Caution" without warning symbol  indicates that the operator is not likely to be injured, but it can cause damage or result in a malfunction to this product, your equipment, or your instruments.

In addition to the hazardous level classifications described above, the following notations are also used.

I m p o r t a n c e

"Importance" indicates the information and contents that must be known particularly in operations and maintenance works of this product.

R e m a r k s

"Remarks" initiates the useful information or contents for operations and maintenance works of this product.

1.1.1.3 Warning symbol

In this manual, the following symbols are added along with the notations "Danger," "Warning," "Caution," and "Importance" to indicate the warning contents in an easy-to-understand manner.



This symbol indicates that a high voltage may be applied.

Failure to confirm safety or mishandling of this product might cause a risk of electric shock, burn, or death.



This symbol indicates that some parts have a high surface temperature, and the mishandling can cause a risk of burns.



This symbol indicates that mishandling may cause a fire.



This symbol indicates "prohibited" actions that must not be performed in the operation and the maintenance work of this product.



This symbol indicates "mandatory" actions that must be performed in the operation and the maintenance work of this product.

1.1.2 Terminology

Terminology used in this manual is described below.

Refer to our web pages for terms that are not described in this section.

- 1st pre-register (= Pre-register)

Pre-register is a register to set the continuous operation data during operation.

It exists for every function such as for positioning control, speed control, and the like.

When the operation mode is completed, each pre-register value will be copied to the current register at the same time so as to start the next operation mode.

- 2nd pre-register

This register is provided in the previous stage of 1st pre-register, which is a register for setting continuous operation data.

Usually, if the 2nd pre-register is available, you write to the current register or 1st pre-register via the 2nd pre-register.

- Common-pulse mode (OUT, DIR)

One of the output forms of pulse signals for driving a motor.

Output pulse signals (OUT) and direction signals (DIR) are output.

- 2-pulse mode (PLS, MNS)

One of the output forms of pulse signals for driving a motor.

Plus direction pulse signals (PLS) and Minus direction pulse signals (MNS) are output.

- 90-degree phase difference mode (PHA and PHB)

One of the output forms of pulse signals for driving a motor.

A-phase pulse signals (PHA) and B-phase pulse signals (PHB) with 90-degree phase differences are output.

The signal frequency will be 1/4 of the operating speed of the motor.

Therefore, an inexpensive interface circuit can be used.

- CW

Clockwise.

In CW circular interpolation, X-axis operates in the plus direction and Y-axis operates in the minus direction in the first quadrant.

- CCW

Counterclockwise.

In CCW circular interpolation, X-axis operates in the minus direction and Y-axis operates in the plus direction in the first quadrant.

1.1.3 Notation

- (1) For the suffixes of pin names, register names, and bit names, "x" indicates the X-axis , "y" indicates the Y-axis , "z" indicates the Z-axis, and "u" indicates the U-axis. "N" indicates any axis.
- (2) Negative logic pins and negative logic signals are not indicated by overbars, or the like.
For logic details, see "4.3 Functions of pins".
- (3) In the explanation of the bits of registers, "0" indicates that the bit is prohibited to use other than "0" when writing. Also, the bit is fixed at "0" when reading.
- (4) The specific bit of a status or a register is shown as "status name.bit name" or "register name.bit name".
(For example, RMD.MSY represents MSY bit in RMD register.)
- (5) If there is a description of time, it shows a value at "Reference clock frequency = 19.6608 MHz" unless otherwise specified.
- (6) Regarding the signal status of "ON" or "OFF", "Rising edge", "H-level" or "1" indicates "ON" in the case of positive logic.
"Falling edge", "L-level" or "0" indicates "ON" in the case of negative logic.
- (7) The numeric suffix "b" represents a binary number, and "h" represents a hexadecimal number.
No suffix is added to a decimal number.
Even if it is a binary or hexadecimal number, a suffix is not added in some graphs or when the binary or hexadecimal number is the same as the decimal number.
- (8) The consecutive bits are indicated by " : ".
(For example, MST5 [7:0] represents the 7th to 0th bits of MST5.)

1.2 Handling the product

1.2.1 Storing

Store the product in an environment where condensation does not occur at a temperature from -65°C to $+150^{\circ}\text{C}$.

1.2.2 Unpacking

Check if the quantity of the product you ordered, and moisture-proof desiccants are included in the package when unpacked.


1.2.3 Safety


This section describes the basic safety precautions for safer operations.

Follow the instructions below when using the product.

Failure to comply with the items may result in injuries or disasters.

1.2.3.1 Precaution in design

 **C a u t i o n**



- Keep the operating voltage, operating temperature, input /output voltage/ current, etc. within the rated ranges. If used outside the rated range, it may increase the failure rate even if the LSI operates normally in the short term. Even within the rated range, the failure rate changes depending on the operating temperature and voltage. Take this into consideration when designing your equipment. Additionally, never exceed the absolute maximum ratings, even for a very short time.
- Take precautions against the influence of heat in the environment and keep the temperature around the LSI as low as possible.
- Note that ignoring the following precautions may result in latching up and may cause overheating and smoke:
 - Keep the voltage levels of input/output pins within the absolute maximum rated ranges.
 - Please consider the timing of power-on.
 - Be careful not to introduce external noise into LSIs.
 - Fix the potential of unused input pins to V_{DD} or GND, or pull them up or down.
 - Pull up or down the potential of unused bidirectional pins.
 - Do not short-circuit the outputs.
 - Protect LSIs from induction, static electricity and the like from high voltage generating circuits.
- Prevent LSIs from being applied overvoltage caused by a noise, surge, or static electricity.

1.2.3.2 Precautions for transportation and storing LSIs



C a u t i o n



- Always handle LSIs and the packages including LSIs with care. Do not throw or drop them. It may damage the LSIs or tear the aluminum-laminated packaging material and impair the airtightness.
- Do not store LSIs in a location exposed to water droplets or direct sunlight.
- Do not store LSIs in a location where corrosive gases are present or in excessively dusty environments.
- Store the LSIs in an anti-static storage container, and do not apply physical load to LSIs.
- Follow the caution signs and instructions on the packaging box when transporting or storing.
- The temperature and humidity of the storage location should be 30 ° C and 70% RH or less as a guide.
- Store in a place where there is minimal temperature fluctuation.
- Sudden changes in temperature during storage cause condensation, oxidation and corrosion of leads, and poor solder wettability.
- Install an antistatic mat on the surface of the storage shelf and ground the surface of the mat. (Resistance between surface and ground 7.5×10^5 to 1×10^9 ohm)
- When you remove LSIs from the packaging and then store them again, use an antistatic storage container.

1.2.3.3 Precautions for handling environment



C a u t i o n



- Humidity should be 40 to 70 % as a guide. Take into consideration the moisture absorption after opening the moisture-proof packaging product.
- Make sure to ground all equipment, tools, and jigs that are present at the work site.
- Place a conductive mat on the floor in the work area to prevent static electricity on the floor surface and ground it.
(Resistance between surface and ground 1×10^9 ohm or less)
- Use a conductive mat on the surface of the work desk and ground it.
(Resistance between surface and ground 7.5×10^5 to 1×10^9 ohm)
- Do not use metal on the surface of the work desk. If metal is used, its low resistance will cause a sudden discharge when the charged LSI touches it.
- When picking up the surfaces of LSIs with a vacuum, prevent electrostatic charges such as by using conductive rubber at the tip of the contact part.
- Do not allow charged objects (work clothes, human body, etc.) to touch the LSI.
- Use a filter for OA equipment to shield the surface of the display (CRT, etc.) in the work area from static electricity. Avoid switching ON and OFF during work as much as possible.
- Use a conductive cover or conductive casters to ground the work chair to the floor.
(Resistance between seat and ground 1×10^{10} ohm or less)
- Operators must wear wrist straps which are grounded through resistance.
(When worn, resistance between surface and ground 7.5×10^5 to 3.5×10^7 ohm)
- Handle the LSIs and the packages including LSIs with care, and avoid mechanical vibration and impact.

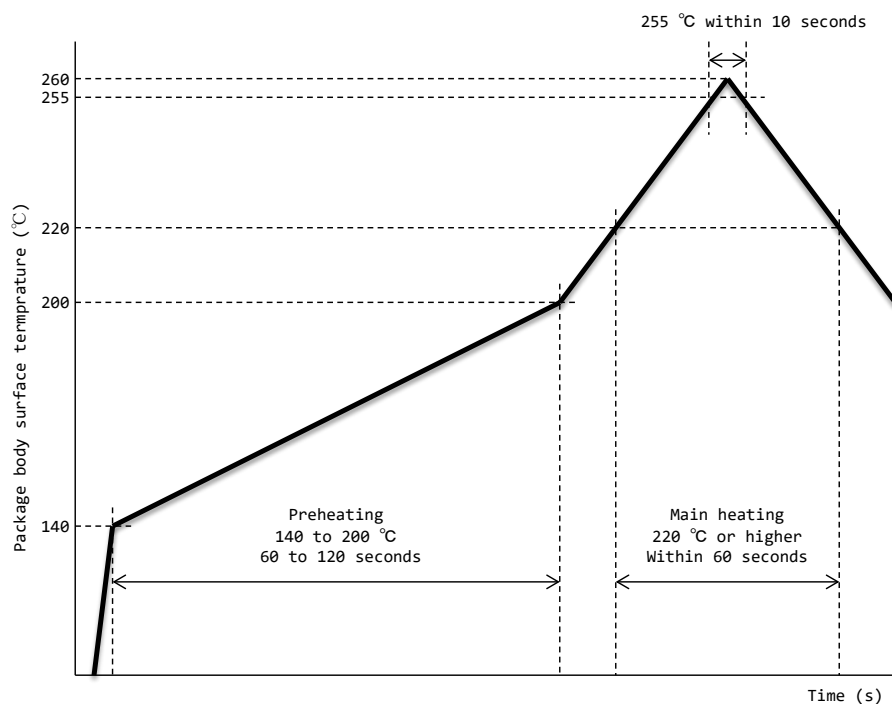
1.2.3.4 Precautions for installation



C a u t i o n

- Plastic packages are easy to absorb moisture, and even if left indoors, the moisture absorption will progress over time. If the LSI is put into a solder reflow oven while absorbing moisture, the resin may crack or the adhesion between the resin and the frame may deteriorate.
- The storage conditions before mounting are 30 °C or less and 80 % RH or less for 1 year before opening the moisture-proof package and 30 °C or less and 70 % RH or less for 1 month after opening the moisture-proof package. (Storage rank: Equivalent to MSL2a)
- If you are concerned about moisture absorption, dry the package before reflowing.
- Drying is at 125 ± 5 °C for 20 hours or more and 36 hours or less. The number of times should be within 2 times.
- Basically, it is necessary to dry when 720 hours has passed after opening the moisture-proof package. When soldering by a method such as infrared reflow that heats the whole, work within the following condition range and the reflow is limited to twice.

The temperature profile of the infrared reflow oven (the temperature is the package surface temperature) is within the range shown in the figure below.



[Profile (lead-free solder mounting compatible)]

- Soldering by the solder dipping method causes a sudden temperature change in the package. Avoid this method as it can damage the LSI.
- Perform manual soldering work using the soldering iron under the following conditions.
 - The max temperature of the tip should be 350 °C, within 5 seconds, and limited to twice.
 - Be careful not to let the soldering iron come into contact with anything other than the leads, such as the package body.

1.3 Product Warranty

This is the warranty of the product purchased from Nippon Pulse Motor. When the product is purchased from a supplier other than NPM, please contact that supplier regarding the product's warranty.

1.3.1 Warranty period

The warranty period is one year from the date of delivery to an assigned place.

1.3.2 Warranty scope

If any defect is found in a product during the warranty period under the normal use following this document, NPM will repair or replace the product without charge.

However, the following cases are not covered by the warranty even during the warranty period.

- 1) Products modified or repaired by anyone other than NPM or a person authorized by NPM.
- 2) Defects that result from dropping after the delivery or mishandling in transit.
- 3) Natural deterioration, wearing, and fatigue of components.
- 4) Defects result from any usage other than the original described in this manual.
- 5) Defects result from natural disaster or force majeure such as fires, earthquakes, lightning strikes, winds, floods, salts or electrical surges.
- 6) Defects or damages result from a cause that is not the fault of NPM.

1.4 Notice

This document aims to describe the details of functions of the product. It does not warrant fitness for a particular purpose of the customer. Also, the examples of applications and circuit diagrams in this manual are included only for your reference. Please confirm the features and the safety of device or equipment before use.

1.5 Confirmation

Please do not use this product in the following conditions. If you need to use in the following conditions, please contact our sales representatives:

1. Any equipment that may require a high reliability or a safety, such as nuclear facilities, electricity or gas supply systems, transportation facilities, vehicles, various safety systems, medical equipment, etc.
2. Any equipment that may directly affect human survival or property.
3. Usages under conditions or circumstances that are not specified in the catalog, manual, etc.

For applications that may cause serious damages to a human life or property due to failure of this product, ensure high reliability and safety by redundant design.

2. Outline

2.1 Features

PCL6046 is a 4-axis pulse control LSI for stepping motors and servo motors.

In the CMOS configuration, the connection with a CPU can be selected from four types of parallel bus interfaces.

In communication with a CPU, you can input commands, input/output data, and output signals such as interrupt requests.

Motor speed can be controlled at constant speed or with acceleration/deceleration. For acceleration/deceleration, you can select either linear or S-curve.

Motor positioning can be controlled by continuous, set amount, or sensor signals.

- CPU interface

The following four types of CPU interface circuits are built-in.

1. 16-bit interface for 68000
2. 16-bit interface for H8
3. 16-bit interface for 8086
4. 8-bit interface for Z80

- Direct access to internal registers

When connecting all A9 to A0 pins, the register control command is not required.

In this case, you can access the internal registers directly in the 1024-byte address space.

A register control command is required when connecting A7 to A3 pins to VDD or GND.

In this case, you can access the internal registers indirectly only in the 32-byte address space.

- Acceleration/deceleration speed control

Linear acceleration/deceleration and S-curve acceleration/deceleration are available.

For S-curve acceleration/deceleration, linear acceleration/deceleration lines can be inserted in the middle by setting S-curve sections.

For S-curve section setting, acceleration and deceleration characteristics can be set independently.

You can perform S-curve deceleration after Linear acceleration, or vice versa.

- Interpolation operation

You can perform linear interpolation with any two to four axes, and circular interpolation with any two axes.

- Target speed override

Target speed can be changed during operation.

However, if you avoid the triangular drive in automatic FH correction function, the target speed override will not work. If you set S-curve acceleration/deceleration and constant synthesized speed control function for interpolation control of two or more axes, the optimum slow-down point cannot be re-calculated. Set the acceleration/deceleration characteristics to linear or set the constant synthesized speed control function to OFF.

- Target position override

You can change the target position during operation.

The following two types of target position override functions are built-in.

1) Target position (feed amount) can be changed while feeding in the positioning control.

If the current position has exceeded the newly entered position, the motor will decelerate-stop (immediate stop in FL or FH constant speed pattern), and then move in the reverse direction.

2) Continuous movement is performed until PCS signal is input, and positioning control is performed when PCS signal is input.

- Triangle drive avoidance (FH correction function)

In the positioning mode, when there are a small number of output pulses, this function automatically lowers the maximum speed and avoids triangle driving.

- Pre-register

The 1st and 2nd data for continuous operation (feed amount, initial speed, feed speed, acceleration rate, deceleration rate, speed magnification rate, slow-down point, operation mode, center coordinate of circular interpolation, S-curve section in acceleration, S-curve section in deceleration, number of circular interpolation steps) can be written during operation. When the current operation mode is completed, the 1st data for continuous operation is automatically executed. In addition, 1st and 2nd data for continuous comparison can be written to the comparator as well.

- Counter

The following four counters are available for each axis.

Counter	Main purpose	Count target
Counter 1	Command position control (32 bit)	Command pulse
Counter 2	Mechanical position control (32 bit) (Can be used as a general-purpose counter)	Encoder Command pulse Manual pulser
Counter 3	Controlling the deviation between the command position and the mechanical position (16 bit)	Deviation between command pulse and encoder Deviation between command pulse and manual pulser Deviation between encoder and manual pulser
Counter 4	IDX signal output (32 bit) (Can be used as a general-purpose counter)	Command pulse Encoder Manual pulser 1/2 cycle of reference clock

All counters can be cleared by writing a command or by inputting a CLR signal.

You can latch them by writing a command, or by inputting an LTC or ORG signal, and can clear them when latching.

The Counter 1, Counter 2, and Counter 4 have a ring count function that repeats counting through a specified counting range.

- Comparator

There are five comparators for each axis. They can compare the target values and counter values.

The count values to be compared can be selected from all four counters.

- Software limit function

You can set software limits using two circuits of the comparator.

When a motor enters the software limit range, the motor will stop immediately or decelerate-stop. After that motor can only move in the opposite direction to the previous movement.

- Backlash correction / Slip correction

Backlash correction corrects the feed amount each time the moving direction is changed. Slip correction corrects the feed amount regardless of the moving direction. However, the backlash correction cannot be applied even if the direction is changed while performing a circular interpolation.

- Index output

PCL6046 can output IDX signals at specified intervals.

- Simultaneous start

You can start any multiple axes simultaneously by writing a command or inputting a CSTA signal.

Any multiple axes can be selected from multiple PCL6046s.

- Simultaneous stop

You can stop any multiple axes simultaneously by writing a command, inputting a CSTOP signal, or by abnormal stopping of any axis. Any axis or any multiple axes can be selected from multiple PCL6046s.

- Vibration suppression

Specify a control constant in advance and add one pulse each for reverse and forward movement just before stopping. By adding the two pulses, vibration can be suppressed when stopping.

- Manual pulser input

By applying manual pulse signals, you can rotate a motor directly in 1 pulse increment.

The input signals can be either 90-degree phase difference signals (1x, 2x, or 4x) or up/down signals.

For input signals, the magnification circuit of 1 to 32 times and the division circuit of $\frac{1 \text{ to } 2048}{2048}$ are built-in.

End limit (+ELn, -ELn) and software limit (+SL, -SL) settings are available.

At each limit position, the command pulse is stopped, but the operation mode is not cancelled.

Therefore, the command pulse output in the opposite direction can be continued.

- External switch input

You can operate a motor directly in the direction of travel by inputting signals by an external switch.

There are two input pins for the signals: forward rotation (+DRn) and reverse rotation (-DRn).

- Out-of-step detection

If the command pulse and encoder pulse are used for a counter (for deviation), out-of-step can be detected by a comparator.

- Idling pulse output

You can set the number of pulses to operate at the initial speed (FL) before accelerating to the operating speed (FH). You can also avoid out-of-step during acceleration by setting the self-starting frequency of a stepping motor to the initial speed.

- Operation mode

Various operation modes are built-in depending on the combination of control method, operation method, and functions.

<Examples of the operation modes>

- (1) Continuous movement in command control.
- (2) Continuous movement and incremental movement in pulser control.
- (3) Continuous movement and incremental movement in switch control.
- (4) Return to origin in sensor control.
- (5) Incremental movement of positioning control.
- (6) Incremental movement start in positioning control by CSTA signal.
- (7) Incremental movement start in positioning control by PCS signal.

- Mechanical signal input

The following five signals can be input for each axis.

- 1) +ELn: When this signal is turned ON, while moving in the positive (+) direction, movement on this axis stops immediately or decelerate-stop). When this signal is ON, no further movement occurs in the positive (+) direction. (The motor can be rotated in the negative (-) direction.)
- 2) -ELn: Functions the same as the +EL signal except that it works in the negative (-) direction.
- 3) +SDn: Decelerates or decelerate-stops according to the software settings. In the deceleration setting, if this signal is turned ON during high-speed operation, the speed will be decelerated to FL speed. Also, if this signal is already ON at the start, FL constant speed operation will be performed. In the deceleration stop setting, if this signal is turned ON during high-speed operation, the motor decelerates to FL speed and stop. Also, if this signal is already ON at the start, the motor will not operate.
- 4) -SDn: Similar to +SD signal, processed in the case of -direction operation.
- 5) ORGn: Input signal for an origin return operation.

For safety, make sure +EL and -EL signals stay ON from the EL position until the end of each stroke.

The input logic for these signals can be changed using ELLn pin.

The input logic of +SD, -SD and ORG signals can be changed using software.

- Servomotor Interface

The following two signals can be input and one signal can be output:

- 1) INP: Inputs INP (positioning complete) signal that is output by a servo motor driver.
- 2) ERC: Outputs ERC (deviation counter clear) signal to a servomotor driver.
- 3) ALM: Inputs ALM (Alarm) signal that is output by a servomotor driver.

Regardless of the direction of operation, when this signal is ON, movement on this axis stops immediately or decelerate-stops. When this signal is ON at start, no movement occurs on this axis.

The input/output logic of INP, ERC, and ALM signals can be changed using software.

ERC signal is a pulse output. The pulse length can be set. (11 μ s to 100 ms. A level output is also available.)

- Origin return sequences

In sensor control, in addition to the origin (ORG_n), the encoder Z-phase (EZ_n), end limit (+EL_n, -EL_n), and slow-down (+SD_n, -SD_n) can be combined to perform various origin return sequences.

- 1) Stops when ORG signal is turned ON
- 2) Stops when the specified number of EZ signals is counted after ORG signal is turned ON.
- 3) Stops when ORG signal turns ON. After reversing, stops when the specified number of EZ signals is counted.
- 4) Stops when EL signal in the operating direction turns ON.
- 5) Stops when EL signal in the operating direction turns ON. After reversing, stops after the specified number of EZ signals is counted.
- 6) Decelerates when SD signal in the operating direction is ON and stops when the ORG signal is ON.
- 7) Decelerates when ORG signal turns ON and stops when the specified number of EZ signals is counted.
- 8) Decelerate-stops when ORG signal is ON. After reversing, stops after the specified number of EZ signals is counted.
- 9) Decelerate-stops after ORG signal memorizes the ON position. After reversing, stops at the memorized position.
- 10) After ORG signal is turned ON, the position where the specified number of EZ signals is counted is memorized. Decelerates and stops. After reversing, stops at the memorized position.
- 11) Decelerates and stops when EL signal in the operating direction is ON. After reversing, the position where the specified number of EZ signals is counted, and decelerate-stops. After reversing, stops at the memorized position.

- Output pulse specifications

Output pulses can be selected among common-pulse mode (OUT/DIR), 2-pulse mode (PLS/MNS) or 90-degree phase difference mode (PHA/PHB). The output logic can also be selected.

In 90-degree phase difference mode, frequency of output signals reduces to 1/4 of operation speed.

Therefore, if the mode is selected, frequency characteristics of interface circuit can be lowered.

- Emergency stop input

When CEMG signal is turned ON, operation stops immediately. When this signal is ON at start, no movement is allowed.

- Interrupt request output

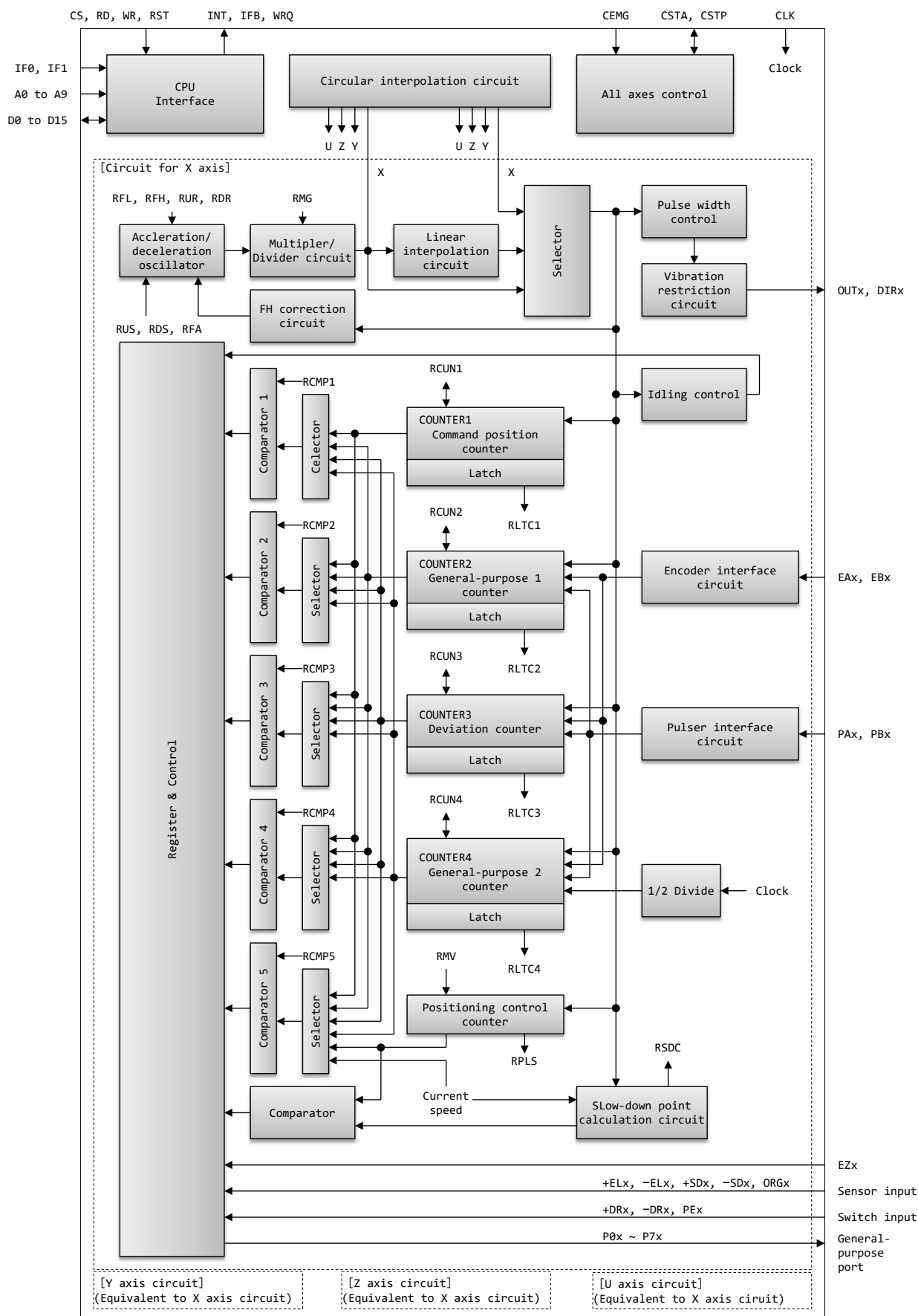
By various factors, L-level can be output from INT pin.

Output from INT pin by OR logic of each factor for each axis.

When more than one PCL6046 are used, wired OR connections are not possible.

2.2 Configuration

Controlling PCL6046 requires a crystal oscillator outputting recommended frequency of 19.6608 MHz and a CPU with a Parallel-bus interface with a 16-bit or 8-bit data bus.



3. Specifications

The following table shows the specifications of PCL6046 such as the functions.

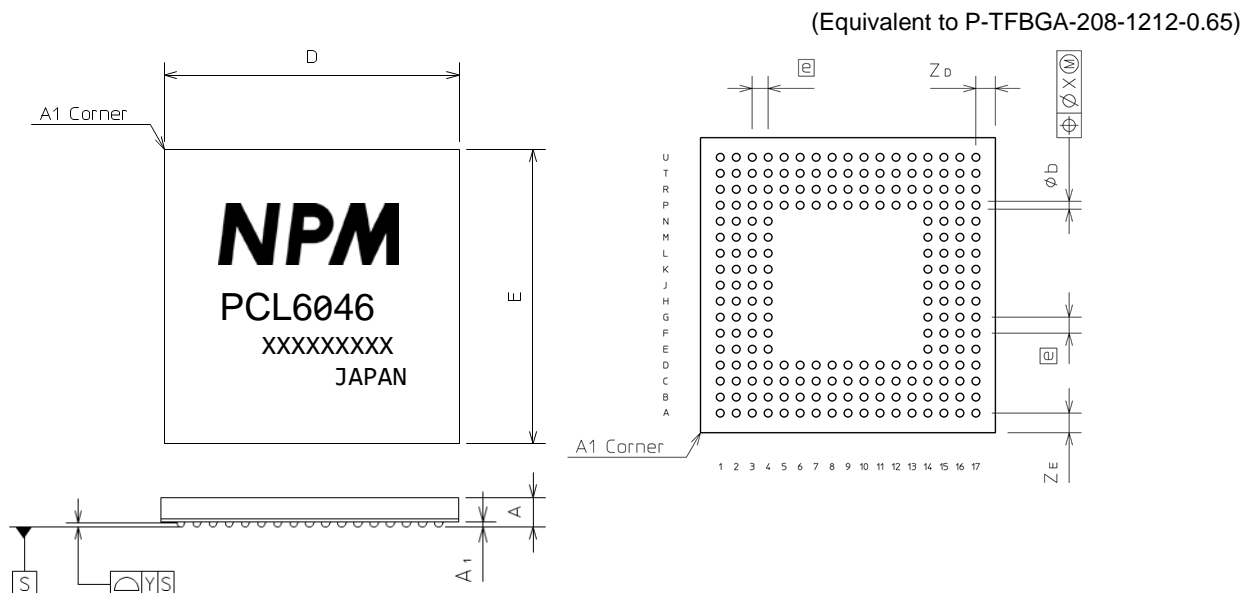
Item	Description
Number of axes [axis]	4 axes
Positioning control range [pulses]	-2,147,483,648 to +2,147,483,647 (32 bits)
Number of registers used for setting speeds [/axis]	3 (FL, FH, and FA)
Speed setting step number [steps]	1 to 65,535 (16 bits)
Speed magnification range [times]	0.1 to 100 <Examples of the reference clock 19.6608 MHz> Multiply by 0.1: 0.1 to 6,553.5 pps Multiply by 1: 1 to 65,535 pps Multiply by 100: 100 to 6,553,500 pps <Example of the reference clock 30.0 MHz> Multiply by 152.5: 152.5 to 9,999,847 pps (pps: pulse per second)
Slow-down point setting range [pulses]	0 to 16,777,215 (24 bits)
Acceleration/deceleration characteristics	Four types: linear acceleration, linear deceleration, S-curve acceleration, and S-curve deceleration can be combined.
Acceleration rate setting range [step ⁻¹]	1 to 65,535 (16 bits)
Deceleration rate setting range [step ⁻¹]	1 to 65,535 (16 bits)
Counter [Circuit/axis]	4: Counter 1: Command position counter (32 bits) Counter 2: General-purpose counter 1 (32 bits) Counter 3: Deviation counter (16 bits) Counter 4: General-purpose counter 2 (32 bits)
Comparators [Circuit/axis]	5 (32 bits)
Interpolation functions	Linear interpolation: One set of any two or more axes, and one set of any one or more axes Circular interpolation: One set of any two axes
Reference clock frequency [MHz] (f _{CLK})	19.6608 (Max: 30 MHz)
CPU interface (Parallel-Bus)	68000 (16 bits), H8 (16 bits), 8086 (16 bits), Z80 (8 bits)
Package type	208 pin TFBGA
Package size [mm]	12 × 12 (Mold part)
Weight [g]	0.28 (typ.)
Power supply [V]	Single power supply of 3.3 V
Storage temperature [°C]	-65 to +150
Operating ambient temperature [°C]	-40 to +85 (T _j = -40 to +125 [°C], θ _{j-a} = 24[°C/W])
Chip configuration	C-MOS

4. Hardware description

The following shows the external dimensions and pin list. The connection between PCL6046 and a CPU is explained.

4.1 External dimensions

A1 pin is located at the upper left of PCL6046 marking.



Symbol	Dimension in Millimeters		
	Minimum	Nominal	Maximum
D	11.90	12.00	12.10
E	11.90	12.00	12.10
A	-	-	1.20
A ₁	0.17	0.22	0.27
$\square e$	-	0.65	-
b	0.27	0.32	0.37
X	-	-	0.08
Y	-	-	0.10
Z _E	0.70	0.80	0.90
Z _D	0.70	0.80	0.90

4.2 Pin assignment diagram

TOP VIEW

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	(NC)	GND	ELLz	VDD	GND	GND	PEz	LTCu	+DRu	EZu	LTCz	+DRz	PCSy	PBz	EBz	VDD	(NC)	A
B	VDD	ELLu	ELLy	CEMG	GND	CLK	PEy	CLRu	GND	EBu	CLRz	GND	-DRy	PAz	EAz	PBy	GND	B
C	IF0	RST	ELLx	CSTP	GND	VDD	PEx	PCSu	PBu	EAu	PCSz	LTCy	+DRy	EZz	EBy	EZy	PAY	C
D	CS	GND	IF1	CSTA	GND	PEu	GND	-DRu	PAu	VDD	-DRz	CLRy	GND	CLRz	LTCx	GND	EAY	D
E	A0	GND	WR	RD										VDD	+DRx	-DRx	PCsx	E
F	A4	A3	A2	A1										EBx	EZx	PAX	PBx	F
G	A7	A6	A5	VDD										ALMu	INPu	GND	EAX	G
H	INT	GND	A9	A8										+SDu	-SDu	+ELu	-ELu	H
J	D0	VDD	IFB	WRQ										ALMz	INPz	GND	ORGu	J
K	GND	D3	D2	D1										+SDz	-SDz	+ELz	-ELz	K
L	D7	D6	D5	D4										ALMy	INPy	VDD	ORGz	L
M	D10	D9	D8	GND										+SDy	-SDy	+ELy	-ELy	M
N	D13	D12	VDD	D11										ALMx	INPx	GND	ORGy	N
P	P0x	GND	D15	D14	P3y	P7y	P2z	P6z	P1u	P5u	OUTx	DIRy	VDD	ERCx	-SDx	+ELx	-ELx	P
R	P3x	P2x	P1x	P0y	P4y	GND	P3z	P7z	P2u	P6u	DIRx	GND	OUTu	ERCy	GND	ORGx	+SDx	R
T	P5x	P4x	P7x	P1y	P5y	P0z	P4z	VDD	P3u	P7u	VDD	OUTz	DIRu	ERCz	BSYx	BSYz	VDD	T
U	(NC)	P6x	VDD	P2y	P6y	P1z	P5z	P0u	P4u	GND	OUTy	DIRz	GND	ERCu	BSYy	BSYu	(NC)	U
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

4.3 Functions of pins

- [I/O] column shows the direction of signals.
I: Input, O: Output, B: Bidirectional.
- [Logic] column shows the logic of signals.
P: Positive logic, N: Negative logic, #: Can be changed by software, %: Set by hardware.
- [Unuse] column shows the connection destination when not in use.
Open: Not connected, Vdd: VDD or pull-up connection, Pup: Pull-up connection.
Pull-up resistors of 5 to 10 kohm are recommended.
As described in "7.3 DC characteristics", some input pins and bidirectional pins have built-in pull-up resistors.
The built-in pull-up resistor is to prevent floating.
It is recommended to pull up outside to support noise resistance.
- All signal input pins can input 0 to +5 V level.
- Output pins of all signals can be pulled up to +5 V, but cannot output more than VDD.
A pull-up resistor value of 5 kohm or higher is recommended.

Signal name	Ball No.	I/O	Logic	Unuse	Description
GND	A2, A5, A6, B5, B9, B12, B17, C5, D2, D5, D7, D13, D16, E2, G16, H2, J16, K1, M4, N16, P2, R6, R12, R15, U10, U13	-	-	-	Power supply pin connected to GND Connect all GND pins to GND power supply.
VDD	A4, A16, B1, C6, D10, E14, G4, J2, L16, N3, P13, T8, T11, T17, U3	-	-	-	Power supply pin connected to +3.3 V Connect all VDD pins to +3.3 V power supply.
RST	C2	I	N	-	Input pin for the hardware reset signal For details, see "6.1.1 Hardware reset".
CLK	B6	I	P	-	Input pin for the reference clock signal Connect a crystal oscillator outputting the recommended frequency of 19.6608 MHz.

Signal name	Ball No.	I/O	Logic	Unuse	Description																																			
IF0, IF1	C1 D3	I	P	-	Input pins for CPU interface selection <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>CPU interface</th> <th>IF1</th> <th>IF0</th> <th>RD</th> <th>WR</th> <th>A0</th> <th>WRQ</th> </tr> </thead> <tbody> <tr> <td>68000(16 bit)</td> <td>L</td> <td>L</td> <td>VDD</td> <td>R/W</td> <td>LDS</td> <td>DTACK</td> </tr> <tr> <td>H8(16 bit)</td> <td>L</td> <td>H</td> <td>RD</td> <td>HWR</td> <td>GND</td> <td>WAIT</td> </tr> <tr> <td>8086(16 bit)</td> <td>H</td> <td>L</td> <td>RD</td> <td>WR</td> <td>GND</td> <td>READY</td> </tr> <tr> <td>Z80(8 bit)</td> <td>H</td> <td>H</td> <td>RD</td> <td>WR</td> <td>A0</td> <td>WAIT</td> </tr> </tbody> </table>	CPU interface	IF1	IF0	RD	WR	A0	WRQ	68000(16 bit)	L	L	VDD	R/W	LDS	DTACK	H8(16 bit)	L	H	RD	HWR	GND	WAIT	8086(16 bit)	H	L	RD	WR	GND	READY	Z80(8 bit)	H	H	RD	WR	A0	WAIT
CPU interface	IF1	IF0	RD	WR	A0	WRQ																																		
68000(16 bit)	L	L	VDD	R/W	LDS	DTACK																																		
H8(16 bit)	L	H	RD	HWR	GND	WAIT																																		
8086(16 bit)	H	L	RD	WR	GND	READY																																		
Z80(8 bit)	H	H	RD	WR	A0	WAIT																																		
CS	D1	I	N	-	Input pin for chip selection signal Enable RD and WR pins at CS = L level.																																			
RD	E4	I	N	-	Input pin for read signal Enable RD pin at CS = L level.																																			
WR	E3	I	N	-	Input pin for the write signal Enable WR pin at CS = L level.																																			
A0, A1, A2, A3, A4, A5, A6, A7, A8, A9	E1, F4, F3, F2, F1, G3, G2, G1, H4, H3	I	P	-	Input pins for address signal For the reduced address method, connect A3 to A7 pins to VDD or GND by selecting CPU interface. If IF1 = L level, connect VDD. If IF1 = H level, connect GND.																																			
INT	H1	O	N	Open	Output pin for interrupt request signal See "6.18 Interrupt request (INT)" for details.																																			
WRQ	J4	O	N	Open	Output pin for wait request signal																																			
IFB	J3	O	N	Open	Output pin for interface running signal																																			
D0, D1, D2, D3, D4, D5, D6, D7	J1, K4, K3, K2, L4, L3, L2, L1	B	P	-	Bit0 to Bit7 of the data bus are connected.																																			
D8, D9, D10, D11, D12, D13, D14, D15	M3, M2, M1, N4, N2, N1, P4, P3	B	P	Pup	Bit8 to Bit15 of the data bus are connected. For Z80-Bus (8bit), pull-up connection is required.																																			
CSTA	D4	B	N	Pup	CSTA signal input/output pin See "6.9.1 Simultaneous start (CSTA)" for details.																																			
CSTP	C4	B	N	Pup	CSTP signal input/output pin See "6.10 External stop / simultaneous stop" for details.																																			
CEMG	B4	I	N	Vdd	CEMG signal input pin See "6.11 Emergency stop " for details.																																			

Signal name	Ball No.	I/O	Logic	Unuse	Description
ELLx, ELLy, ELLz, ELLu	C3, B3, A3, B2	I	N	Vdd	Input pins to set the input logic of +EL and -EL signals See "6.7.1 End limit (+EL, -EL)" for details.
+ELx, +ELy, +ELz, +ELu	P16, M16, K16, H16	I	N%	Vdd	+EL signal input pins See "6.7.1 End limit (+EL, -EL)" for details.
-ELx, -ELy, -ELz, -ELu	P17, M17, K17, H17	I	N%	Vdd	-EL signal input pins See "6.7.1 End limit (+EL, -EL)" for details.
+SDx, +SDy, +SDz, +SDu	R17, M14, K14, H14	I	N#	Vdd	+SD signal input pins See "6.7.2 Slow-down (+SD, -SD)" for details.
-SDx, -SDy, -SDz, -SDu	P15, M15, K15, H15	I	N#	Vdd	-SD signal input pins See "6.7.2 Slow-down (+SD, -SD)" for details.
ORGx, ORGy, ORGz, ORGu	R16, N17, L17, J17	I	N#	Vdd	ORG signal input pins See "6.7.3 Origin (ORG), encoder Z phase (EZ)" for details.
ALMx, ALMy, ALMz, ALMu	N14, L14, J14, G14	I	N#	Vdd	ALM signal input pins Connect to the servo motor driver. See "6.8.3 Alarm (ALM)" for details.
OUTx, OUTy, OUTz, OUTu	P11, U11, T12, R13	O	N#	Open	Command pulse signal output pins See "6.5 Output pulse control" for details.
DIRx, DIRy, DIRz, DIRu	R11, P12, U12, T13	O	N#	Open	The output pins for the command pulse signal. See "6.5 Output pulse control" for details.

Signal name	Ball No.	I/O	Logic	Unuse	Description
EAx, EAy, EAz, EAu	G17, D17, B15, C10	I	-	Vdd	Encoder A phase signal input pins See "6.12.1 Counter type and input specifications" for details.
EBx, EBy, EBz, EBu	F14, C15, A15, B10	I	-	Vdd	Encoder B phase signal input pins See "6.12.1 Counter types and input specifications" for details.
EZx, EZY, EZz, EZu	F15, C16, C14, A10	I	N#	Vdd	Encoder Z-phase signal input pins See "6.7.3 Origin (ORG), encoder Z phase (EZ)" for details.
PAX, PAY, PAZ, PAu	F16, C17, B14, D9	I	-	Vdd	Manual pulser A-phase signal input pins See "5.5.3 Pulser control" for details.
PBx, PBy, PBz, PBu	F17, B16, A14, C9	I	-	Vdd	Manual pulser B-phase signal input pins See "5.5.3 Pulser control" for details.
PEX, PEY, PEZ, PEu	C7, B7, A7, D6	I	N	Vdd	Input pins for manual pulser and external switch valid signals See "5.5.3 Pulser control" and "5.5.4 Switch control" for details.
+DRx, +DRy, +DRz, +DRu	E15, C13, A12, A9	I	N#	Vdd	+DR signal input pins See "5.5.4 Switch control" for details.
-DRx, -DRy, -DRz, -DRu	E16, B13, D11, D8	I	N#	Vdd	-DR signal input pins See "5.5.4 Switch control" for details.
PCSx, PCSy, PCSz, PCSu	E17, A13, C11, C8	I	N#	Vdd	Input pins for PCS and STA signals. See "6.4.2 Target position override 2 (PCS)" and "6.9.2 Own axis start (STA)" for details.

Signal name	Ball No.	I/O	Logic	Unuse	Description
INPx, INPy, INPz, INPu	N15, L15, J15, G15	I	N#	Vdd	INP signal input pins Connect to a servo motor driver. See "6.8.1 Positioning complete (INP)" for details.
CLR _x , CLR _y , CLR _z , CLR _u	D14, D12, B11, B8	I	N#	Vdd	CLR signal input pins See "6.12.2 Counter clear" for details.
LTC _x , LTC _y , LTC _z , LTC _u	D15, C12, A11, A8	I	N#	Vdd	LTC signal input pins See "6.12.3 Counter latch" for details.
ERC _x , ERC _y , ERC _z , ERC _u	P14, R14, T14, U14	O	N#	Open	ERC signal output pins Connect to a servo motor driver. See "6.8.2 Deviation counter clear (ERC)" for details.
BSY _x , BSY _y , BSY _z , BSY _u	T15, U15, T16, U16	O	N	Open	BSY signal output pins See "5.2.1 Main status (MSTS)" for details.
P0 _x /FUP _x , P0 _y /FUP _y , P0 _z /FUP _z , P0 _u /FUP _u	P1, R4, T6, U8	B	P	Pup	Input/output pins for general-purpose input/output signals. Also serves as signal output pins for under acceleration.
P1 _x /FDW _x , P1 _y /FDW _y , P1 _z /FDW _z , P1 _u /FDW _u	R3, T4, U6, P9	B	P	Pup	Input/output pins for general-purpose input/output signals. Also serves as signal output pins for under deceleration.
P2 _x /MVC _x , P2 _y /MVC _y , P2 _z /MVC _z , P2 _u /MVC _u	R2, U4, P7, R9	B	P	Pup	Input/output pins for general-purpose input/output signals. Also serves as signal output pins during constant speed drive.
P3 _x /CP1 _x (+SL _x), P3 _y /CP1 _y (+SL _y), P3 _z /CP1 _z (+SL _z), P3 _u /CP1 _u (+SL _u)	R1, P5, R7, T9	B	P	Pup	Input/output pins for general-purpose input/output signals. Also serves as output pins for the signal when Comparator 1 condition is satisfied and also for +SL signal.

Signal name	Ball No.	I/O	Logic	Unuse	Description
P4x/CP2x(-SLx), P4y/CP2y(-SLy), P4z/CP2z(-SLz), P4u/CP2u(-SLu)	T2, R5, T7, U9	B	P	Pup	Input/output pins for general-purpose input/output signals. Also serves as output pins for the signal when Comparator 2 condition is satisfied and also for -SL signal.
P5x/CP3x, P5y/CP3y, P5z/CP3z, P5u/CP3u	T1, T5, U7, P10	B	P	Pup	Input/output pins for general-purpose input/output signals. Also serves as output pins for the signal when Comparator 3 condition is satisfied.
P6x/CP4x(IDXx), P6y/CP4y(IDXy), P6z/CP4z(IDXz), P6u/CP4u(IDXu)	U2, U5, P8, R10	B	P	Pup	Input/output pins for general-purpose input/output signals. Also serves as output pins for the signal when Comparator 4 condition is satisfied.
P7x/CP5x, P7y/CP5y, P7z/CP5z, P7u/CP5u	T3, P6, R8, T10	B	P	Pup	Input/output pins for general-purpose input/output signals. Also serves as output pins for the signal when Comparator 5 condition is satisfied.

4.4 CPU interface

The connection to a CPU can be selected from four types: 68000, H8, 8086, and Z80 types.

Select the connection with IF1 and IF0 pins as follow:

CPU interface	IF1	IF0	RD	WR	A0	WRQ
68000(16 bit)	L	L	VDD	R/W	LDS	DTACK
H8(16 bit)	L	H	RD	HWR	GND	WAIT
8086(16 bit)	H	L	RD	WR	GND	READY
Z80(8 bit)	H	H	RD	WR	A0	WAIT

There are two types of connection methods of address pins: Full address method and reduced address method.

In Full address method, you connect all address signals to a CPU and uses a 1024-byte address space.

In reduced address method, you use 32-byte address space without connecting the address signals of A7 to A3 pins to a CPU.

You connect A7 to A3 pins of reduced address method to VDD or GND depending on the CPU interface.

There are two types of register access methods: direct access method and indirect access method.

With Full address method, you can use both access methods.

Only the indirect access method can be used with reduced address method.

For the register access method, see "5.1.1 Access method".

4.4.1 68000 interface

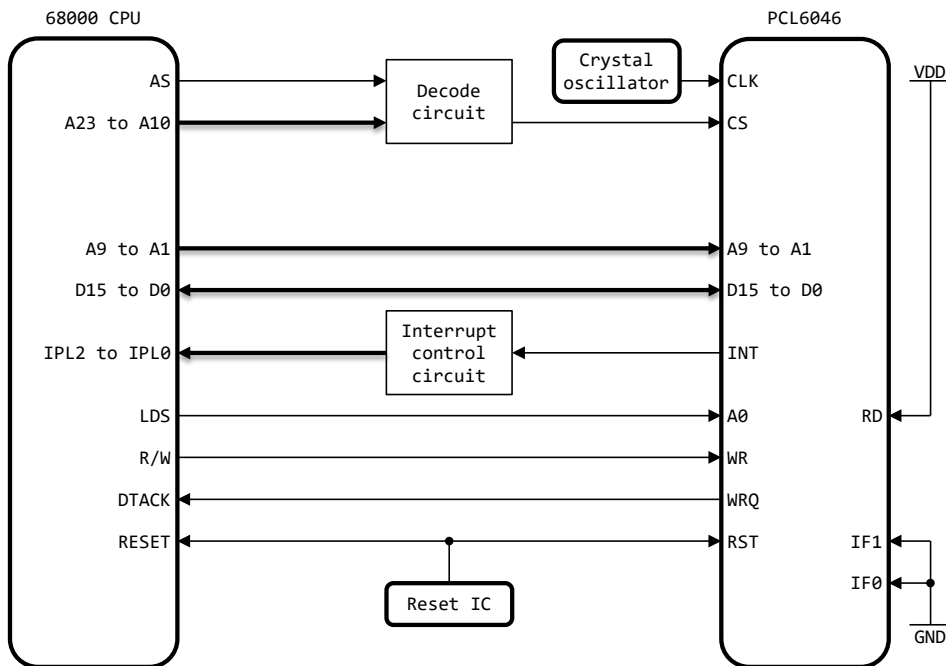
16-bit interface with R/W, LDS, and DTACK signals.

8-bit access is not allowed.

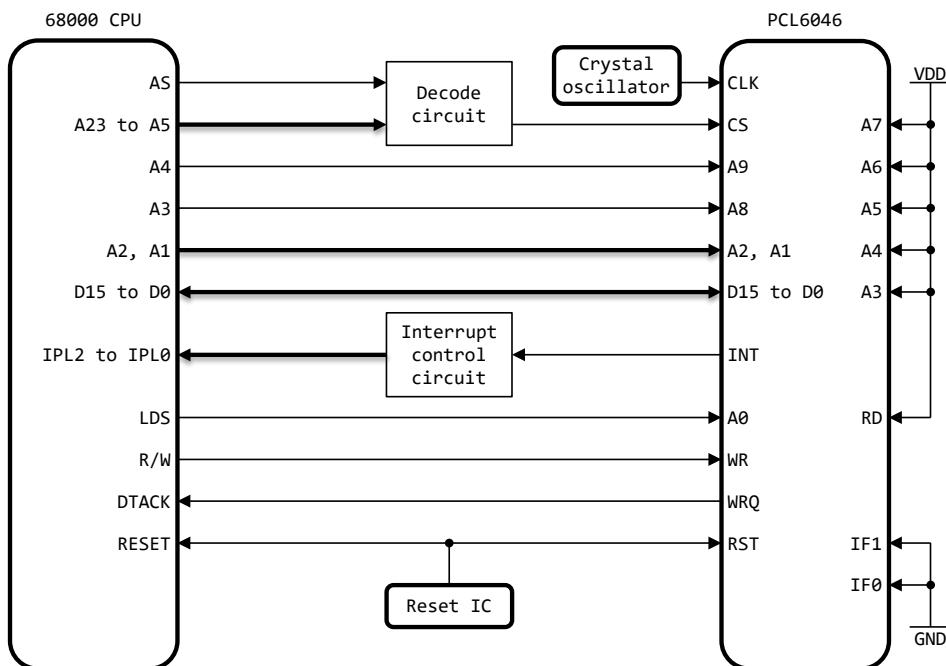
The lower address corresponds to the higher word in I/O buffer.

The Interface is for VME buses or 68000 series CPUs.

4.4.1.1 68000 interface example (Full address)



4.4.1.2 68000 interface example (Reduced address)



4.4.2 H8 interface

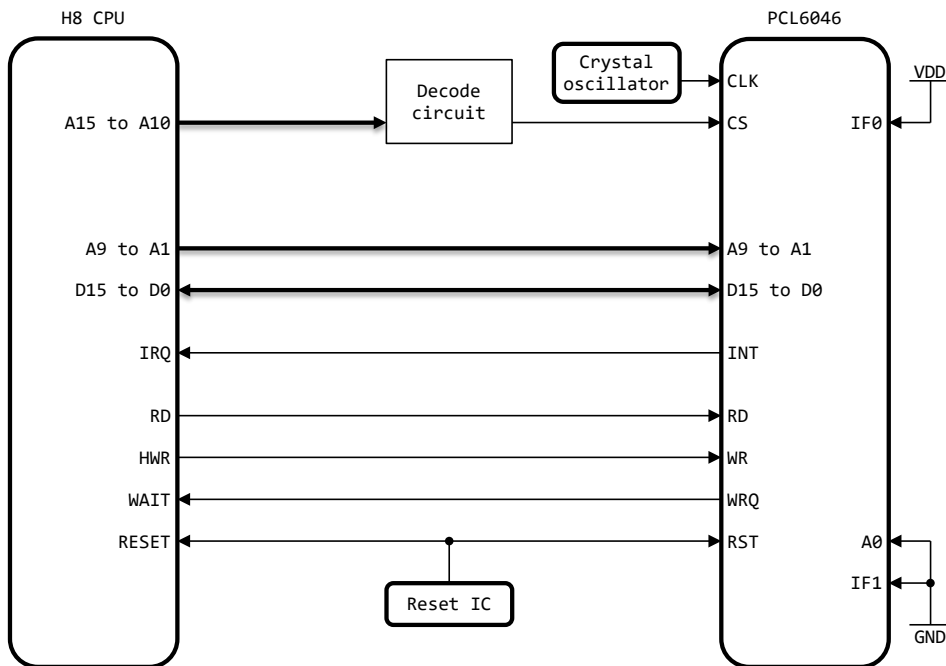
16-bit interface with RD, HWR, and WAIT signals.

8-bit access is not allowed.

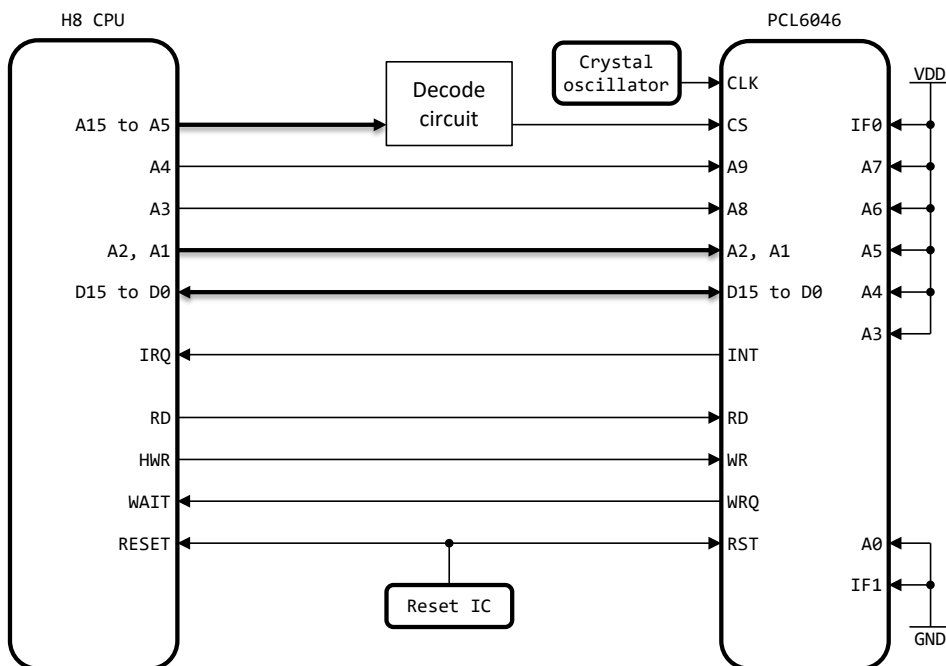
The lower address corresponds to the higher word in I/O buffer.

The interface is for H8 series CPUs.

4.4.2.1 H8 interface address example (Full address)



4.4.2.2 H8 interface example (Reduced address)



4.4.3 8086 interface

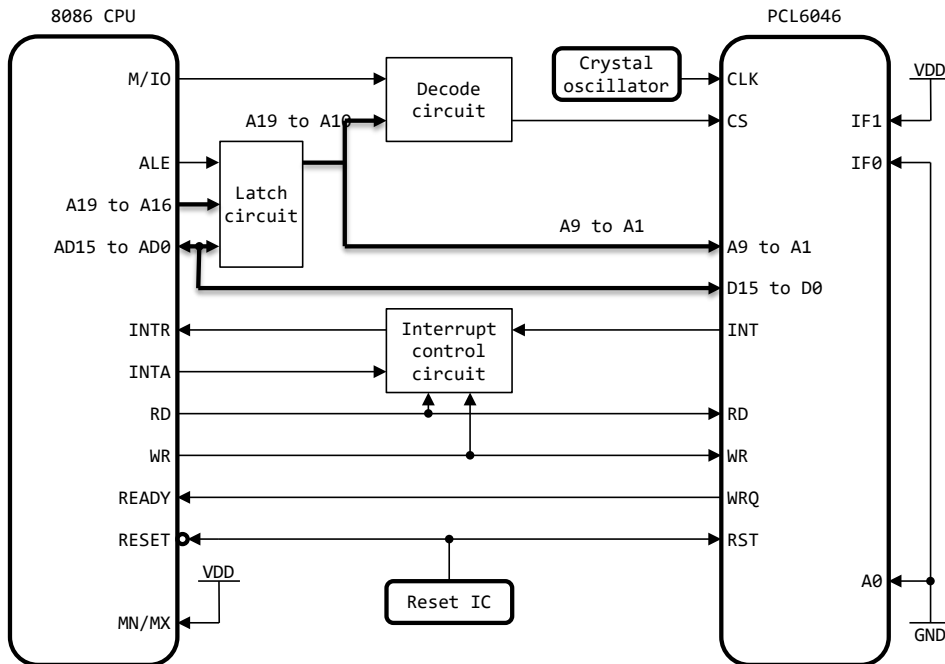
16-bit interface with RD, W/R, and READY signals.

8-bit access is not allowed.

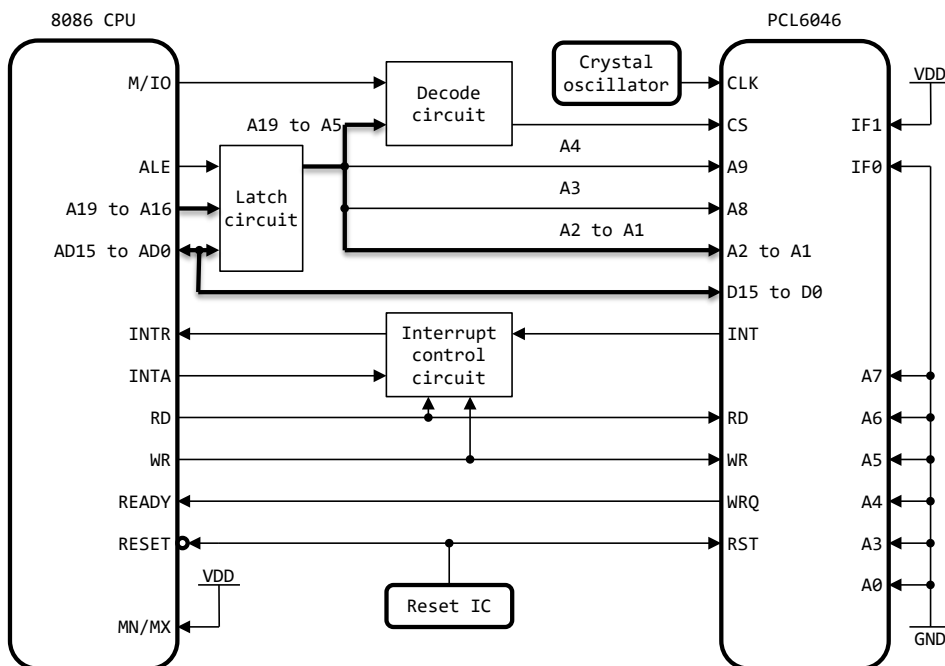
The lower address corresponds to the lower word in I/O buffer.

The interface is for 8086 series CPUs.

4.4.3.1 8086 interface example (Full address)



4.4.3.2 8086 interface example (Reduced address)



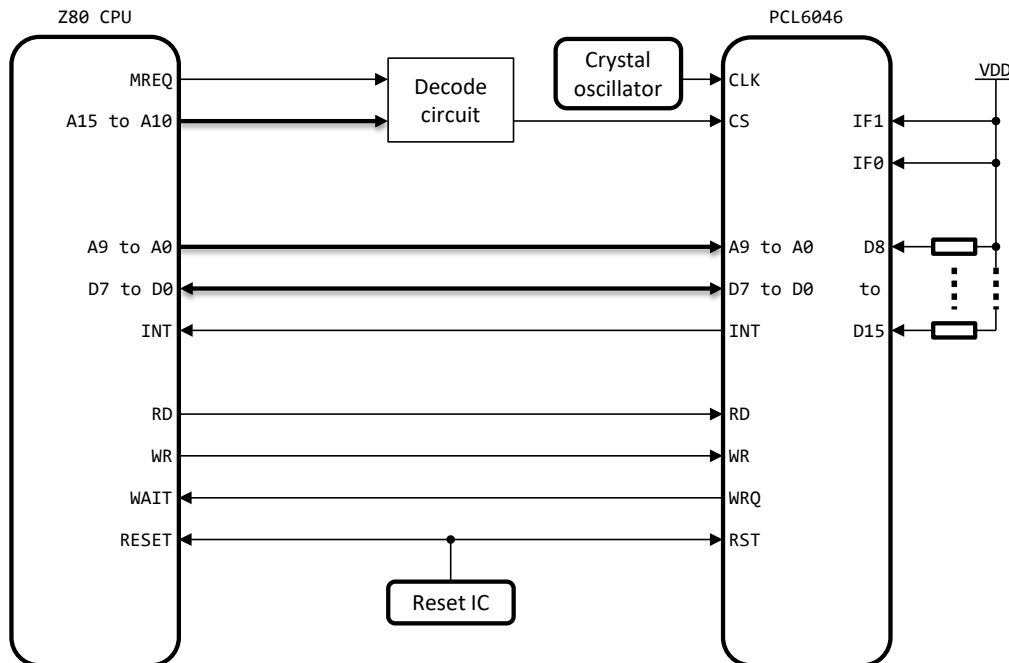
4.4.4 Z80 interface

8-bit interface with RD, W/R, and WAIT signals.

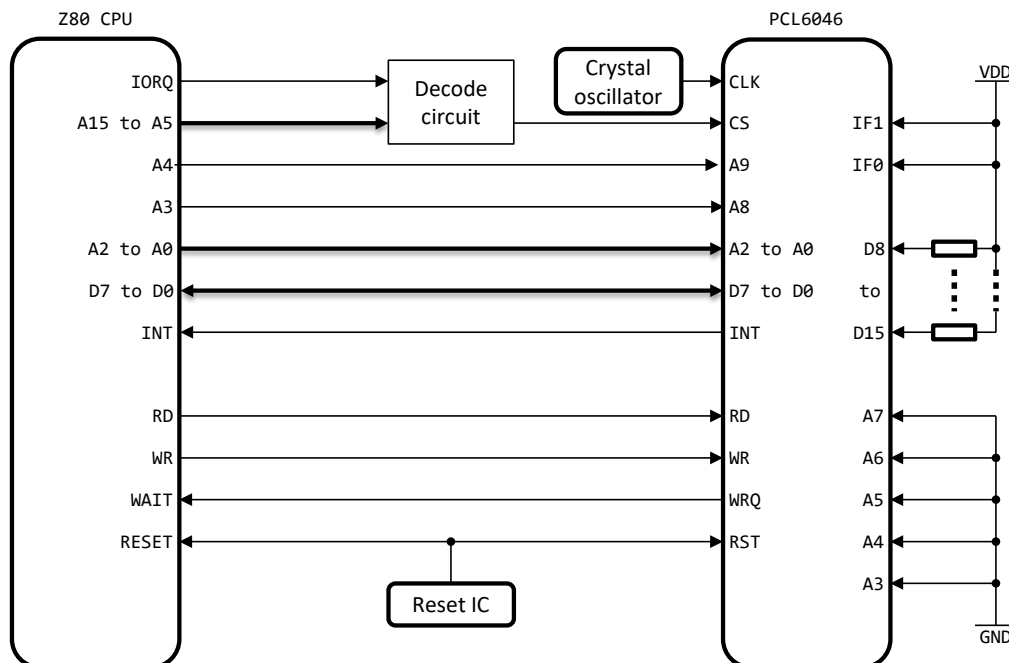
The lower address corresponds to the lower word in I/O buffer.

The interface is for Z80 series CPUs.

4.4.4.1 Z80 interface example (Full address)



4.4.4.2 Z80 interface example (Reduced address)



5. Software description

This chapter describes the communication from a CPU to PCL6046 and shows the commands and registers of PCL6046.

5.1 CPU communication

The communication method from a CPU to PCL6046 is parallel communication.

5.1.1 Access method

If the CPU connection is Full address method, you can use both direct and indirect access methods.

If the CPU connection is Reduced address method, you can only use indirect access method.

5.1.1.1 Direct access method

In direct access method, you can access all address maps of PCL6046.

In this case, you can read and write registers without going through the I/O buffer.

When reading or writing registers, be sure to access from a lower address.

For 68000 and H8 series, access from the upper data. For 8086 and Z80 series, access from the lower data.

For example, for PRMV (12345678h) with a data length of 32 bits in 68000 communication, the lower address is 16 bits of upper data.

Address	Lower address	Upper address
	1111 0100b	1111 0110b
Data	Upper data	Lower data
	1234h	5678h

Conversely, for PRMV (12345678h) with a data length of 32 bits in 8086 communication. the lower address is 16 bits of lower data.

Address	Lower address	Upper address
	0000 1000b	0000 1010b
Data	Lower data	Upper data
	5678h	1234h

Similarly, for PRMV (1245678h) with a data length of 32 bits in Z80 communication, the lowest address is 8 bits of the lowest data.

Address	Lowest address	Lower address	Upper address	Highest address
	0000 1000b	0000 1001b	0000 1010b	0000 1011b
Data	Lowest data	Lower data	Upper data	Highest data
	78h	56h	34h	12h

5.1.1.2 Indirect access method

The indirect access method can only access some address maps of PCL6046.

In this case, you can read and write registers through the I/O buffer.

When reading from a register, write register-read command before reading the I/O buffer.

When writing to a register, write to the I/O buffer and then write the register-write command.

I/O buffer is located on each axis.

Therefore, you can access to the same registers on multiple axes simultaneously by writing a single register control command.

Even in the Full address method, you can use the indirect access method together.

5.1.2 Address map

The address map differs depending on the CPU to be connected and the access method.

5.1.2.1 68000 communication address map (Full address)

Axis	A9, A8	A7 to A0	bit	R/W	Name	Description
X	00b	FEh	16	R	MSTSW	Main status
			16	W	COMW	Axis selection, command
		FCh	16	R	SSTSW	Sub-status, general-purpose I/O ports
			16	W	OTPW	General-purpose output port
		FAh	16	R/W	BUFW0	I/O buffer lower data
		F8h	16	R/W	BUFW1	I/O buffer upper data
		F4h	32	R/W	PRMV	Feed amount (target position) pre-register
		F0h	32	R/W	PRFL	FL speed step number pre-register
		ECh	32	R/W	PRFH	FH speed step number pre-register
		E8h	32	R/W	PRUR	Acceleration rate pre-register
		E4h	32	R/W	PRDR	Deceleration rate pre-register
		E0h	32	R/W	PRMG	Speed magnification pre-register
		DCh	32	R/W	PRDP	Slow-down point pre-register
		D8h	32	R/W	PRMD	Operation mode pre-register
		D4h	32	R/W	PRIP	Circular interpolation center position pre-register
		D0h	32	R/W	PRUS	Acceleration S-curve section pre-register
		CCh	32	R/W	PRDS	Deceleration S-curve section pre-register
		C8h	32	R/W	PRCP5	Comparator 5 comparison value pre-register
		C4h	32	R/W	PRCI	Circular interpolation step number pre-register
		C0h	32	-	-	-
		BCh	32	R/W	RMV	Feed amount (target position) register
		B8h	32	R/W	RFL	FL speed step number register
		B4h	32	R/W	RFH	FH speed step number register
		B0h	32	R/W	RUR	Acceleration rate register
		ACh	32	R/W	RDR	Deceleration rate register
		A8h	32	R/W	RMG	Speed magnification register
		A4h	32	R/W	RDP	Slow-down point register
		A0h	32	R/W	RMD	Operation mode register
		9Ch	32	R/W	RIP	Circular interpolation center position register
		98h	32	R/W	RUS	Acceleration S-curve section register
		94h	32	R/W	RDS	Deceleration S-curve section register
		90h	32	R/W	RFA	FA speed step number register
8Ch	32	R/W	RENV1	Environment setting 1 register		

Axis	A9, A8	A7 to A0	bit	R/W	Name	Description
X	00b	88h	32	R/W	RENV2	Environment setting 2 register
		84h	32	R/W	RENV3	Environment setting 3 register
		80h	32	R/W	RENV4	Environment setting 4 register
		7Ch	32	R/W	RENV5	Environment setting 5 register
		78h	32	R/W	RENV6	Environment setting 6 register
		74h	32	R/W	RENV7	Environment setting 7 register
		70h	32	R/W	RCUN1	Counter 1 (Command position) register
		6Ch	32	R/W	RCUN2	Counter 2 (General-purpose 1) register
		68h	32	R/W	RCUN3	Counter 3 (Deviation) register
		64h	32	R/W	RCUN4	Counter 4 (General-purpose 2) register
		60h	32	R/W	RCMP1	Comparator 1 comparison value register
		5Ch	32	R/W	RCMP2	Comparator 2 comparison value register
		58h	32	R/W	RCMP3	Comparator 3 comparison value register
		54h	32	R/W	RCMP4	Comparator 4 comparison value register
		50h	32	R/W	RCMP5	Comparator 5 comparison value register
		4Ch	32	R/W	RIRQ	Event interrupt request register
		48h	32	R	RLTC1	Counter 1 (command position) latch register
		44h	32	R	RLTC2	Counter 2 (general-purpose 1) latch register
		40h	32	R	RLTC3	Counter 3 (deviation) latch register
		3Ch	32	R	RLTC4	Counter 4 (general-purpose 2) latch register
		38h	32	R	RSTS	Extension status register
		34h	32	R/W	REST	Error interrupt request register
		30h	32	R/W	RIST	Event interrupt factor register
		2Ch	32	R	RPLS	Remaining pulse number register
		28h	32	R	RSPD	Current speed step number register
		24h	32	R	RSDC	Slow-down point auto calculation value register
		10h	160	-	-	-
		0Ch	32	R/W	RCI	Circular interpolation step number register
08h	32	R	RCIC	Circular interpolation step counter register		
04h	32	-	-	-		
00h	32	R	RIPS	Interpolation status register		
Y	01b	(Same as X-axis)				
Z	10b	(Same as X-axis)				
U	11b	(Same as X-axis)				

5.1.2.2 68000 communication address map (Reduced address)

Axis	A9, A8	A2, A1	bit	R/W	Name	Description
X	00b	11b	16	R	MSTSW	Main status
			16	W	COMW	Axis selection, command
		10b	16	R	SSTSW	Sub-status, general-purpose input/output port
			16	W	OTPW	General-purpose output port
		01b	16	R/W	BUFW0	Input/output buffer lower data
		00b	16	R/W	BUFW1	Input/output buffer upper data
Y	01b	(Same as X-axis)				
Z	10b	(Same as X-axis)				
U	11b	(Same as X-axis)				

5.1.2.3 H8 communication address map (Full address)

The address map for H8 communication is the same as for 68000 communication.

See "5.1.2.1 68000 communication address map (Full address)".

5.1.2.4 H8 communication address map (Reduced address)

The address map for H8 communication is the same as for 68000 communication.

See "5.1.2.2 68000 communication address map (Reduced address)".

5.1.2.5 8086 communication address map (Full address)

Axis	A9, A8	A7 to A0	bit	R/W	Name	Description
X	00b	00h	16	R	MSTSW	Main status
			16	W	COMW	Axis selection, command
		02h	16	R	SSTSW	Sub-status, general-purpose I/O ports
			16	W	OTPW	General-purpose output port
		04h	16	R/W	BUFW0	I/O buffer lower data
		06h	16	R/W	BUFW1	I/O buffer upper data
		08h	32	R/W	PRMV	Feed amount (target position) pre-register
		0Ch	32	R/W	PRFL	FL speed step number pre-register
		10h	32	R/W	PRFH	FH speed step number pre-register
		14h	32	R/W	PRUR	Acceleration rate pre-register
		18h	32	R/W	PRDR	Deceleration rate pre-register
		1Ch	32	R/W	PRMG	Speed magnification pre-register
		20h	32	R/W	PRDP	Slow-down point pre-register
		24h	32	R/W	PRMD	Operation mode pre-register
		28h	32	R/W	PRIP	Circular interpolation center position pre-register
		2Ch	32	R/W	PRUS	Acceleration S-curve section pre-register
		30h	32	R/W	PRDS	Deceleration S-curve section pre-register
		34h	32	R/W	PRCP5	Comparator 5 comparison value pre-register
		38h	32	R/W	PRCI	Circular interpolation step number pre-register
		3Ch	32	-	-	-
		40h	32	R/W	RMV	Feed amount (target position) register
		44h	32	R/W	RFL	FL speed step number register
		48h	32	R/W	RFH	FH speed step number register
		4Ch	32	R/W	RUR	Acceleration rate register
		50h	32	R/W	RDR	Deceleration rate register
		54h	32	R/W	RMG	Speed magnification register
		58h	32	R/W	RDP	Slow-down point register
		5Ch	32	R/W	RMD	Operation mode register
		60h	32	R/W	RIP	Circular interpolation center position register
		64h	32	R/W	RUS	Acceleration S-curve section register
		68h	32	R/W	RDS	Deceleration S-curve section register
		6Ch	32	R/W	RFA	FA speed step number register
70h	32	R/W	RENV1	Environment setting 1 register		
74h	32	R/W	RENV2	Environment setting 2 register		
78h	32	R/W	RENV3	Environment setting 3 register		
7Ch	32	R/W	RENV4	Environment setting 4 register		

Axis	A9, A8	A7 to A0	bit	R/W	Name	Description
X	00b	80h	32	R/W	RENV5	Environment setting 5 register
		84h	32	R/W	RENV6	Environment setting 6 register
		88h	32	R/W	RENV7	Environment setting 7 register
		8Ch	32	R/W	RCUN1	Counter 1 (Command position) register
		90h	32	R/W	RCUN2	Counter 2 (General-purpose 1) register
		94h	32	R/W	RCUN3	Counter 3 (Deviation) register
		98h	32	R/W	RCUN4	Counter 4 (General-purpose 2) register
		9Ch	32	R/W	RCMP1	Comparator 1 comparison value register
		A0h	32	R/W	RCMP2	Comparator 2 comparison value register
		A4h	32	R/W	RCMP3	Comparator 3 comparison value register
		A8h	32	R/W	RCMP4	Comparator 4 comparison value register
		ACh	32	R/W	RCMP5	Comparator 5 comparison value register
		B0h	32	R/W	RIRQ	Event interrupt request register
		B4h	32	R	RLTC1	Counter 1 (command position) latch register
		B8h	32	R	RLTC2	Counter 2 (general-purpose 1) latch register
		BCh	32	R	RLTC3	Counter 3 (deviation) latch register
		C0h	32	R	RLTC4	Counter 4 (general-purpose 2) latch register
		C4h	32	R	RSTS	Extension status register
		C8h	32	R/W	REST	Error interrupt factor register
		CCh	32	R/W	RIST	Event interrupt factor register
		D0h	32	R	RPLS	Remaining pulse number register
		D4h	32	R	RSPD	Current speed step number register
		D8h	32	R	RSDC	Slow-down point auto calculation value register
		DCh	160	-	-	-
F0h	32	R/W	RCI	Circular interpolation step number register		
F4h	32	R	RCIC	Circular interpolation step counter register		
F8h	32	-	-	-		
FCh	32	R	RIPS	Interpolation status register		
Y	01b	(Same as X-axis)				
Z	10b	(Same as X-axis)				
U	11b	(Same as X-axis)				

5.1.2.6 8086 communication address map (Reduced address)

Axis	A9, A8	A2, A1	bit	R/W	Name	Description
X	00b	00b	16	R	MSTSW	Main status
			16	W	COMW	Axis selection, command
		01b	16	R	SSTSW	Sub-status, general-purpose input/output port
			16	W	OTPW	General-purpose output port
		10b	16	R/W	BUFW0	Input/output buffer lower data
		11b	16	R/W	BUFW1	Input/output buffer upper data
Y	01b	(Same as X-axis)				
Z	10b	(Same as X-axis)				
U	11b	(Same as X-axis)				

5.1.2.7 Z80 communication address map (Full address)

Axis	A9, A8	A7 to A0	bit	R/W	Name	Description
X	00b	00h	8	R	MSTSB0	Main status [7:0]
			8	W	COMB0	Command
		01h	8	R	MSTSB1	Main status [15:8]
			8	W	COMB1	Axis selection
		02h	8	R	IOPB	General-purpose input/output port
			8	W	OTPB	General-purpose output port
		03h	8	R	SSTSB	Sub status
		04h	8	R/W	BUFB0	Input/output buffer [7:0]
		05h	8	R/W	BUFB1	Input/output buffer [15:8]
		06h	8	R/W	BUFB2	Input/output buffer [23:16]
		07h	8	R/W	BUFB3	Input/output buffer [31:24]
		08h	32	R/W	PRMV	Feed amount (target position) pre-register
		0Ch	32	R/W	PRFL	FL speed step number pre-register
		10h	32	R/W	PRFH	FH speed step number pre-register
		14h	32	R/W	PRUR	Acceleration rate pre-register
		18h	32	R/W	PRDR	Deceleration rate pre-register
		1Ch	32	R/W	PRMG	Speed magnification pre-register
		20h	32	R/W	PRDP	Slow-down point pre-register
		24h	32	R/W	PRMD	Operation mode pre-register
		28h	32	R/W	PRIP	Circular interpolation center position pre-register
		2Ch	32	R/W	PRUS	Acceleration S-curve section pre-register
		30h	32	R/W	PRDS	Deceleration S-curve section pre-register
		34h	32	R/W	PRCP5	Comparator 5 comparison value pre-register
		38h	32	R/W	PRCI	Circular interpolation step number pre-register
		3Ch	32	-	-	-
		40h	32	R/W	RMV	Feed amount (target position) register
		44h	32	R/W	RFL	FL speed step number register
		48h	32	R/W	RFH	FH speed step number register
		4Ch	32	R/W	RUR	Acceleration rate register
		50h	32	R/W	RDR	Deceleration rate register
		54h	32	R/W	RMG	Speed magnification register
		58h	32	R/W	RDP	Slow-down point register
5Ch	32	R/W	RMD	Operation mode register		
60h	32	R/W	RIP	Circular interpolation center position register		
64h	32	R/W	RUS	Acceleration S-curve section register		
68h	32	R/W	RDS	Deceleration S-curve section register		

Axis	A9, A8	A7 to A0	bit	R/W	Name	Description
X	00b	6Ch	32	R/W	RFA	FA speed step number register
		70h	32	R/W	RENV1	Environment setting 1 register
		74h	32	R/W	RENV2	Environment setting 2 register
		78h	32	R/W	RENV3	Environment setting 3 register
		7Ch	32	R/W	RENV4	Environment setting 4 register
		80h	32	R/W	RENV5	Environment setting 5 register
		84h	32	R/W	RENV6	Environment setting 6 register
		88h	32	R/W	RENV7	Environment setting 7 register
		8Ch	32	R/W	RCUN1	Counter 1 (Command position) register
		90h	32	R/W	RCUN2	Counter 2 (general-purpose 1) register
		94h	32	R/W	RCUN3	Counter 3 (deviation) register
		98h	32	R/W	RCUN4	Counter 4 (general-purpose 2) register
		9Ch	32	R/W	RCMP1	Comparator 1 comparison value register
		A0h	32	R/W	RCMP2	Comparator 2 comparison value register
		A4h	32	R/W	RCMP3	Comparator 3 comparison value register
		A8h	32	R/W	RCMP4	Comparator 4 comparison value register
		ACh	32	R/W	RCMP5	Comparator 5 comparison value register
		B0h	32	R/W	RIRQ	Event interrupt request register
		B4h	32	R	RLTC1	Counter 1 (command position) latch register
		B8h	32	R	RLTC2	Counter 2 (general-purpose 1) latch register
		BCh	32	R	RLTC3	Counter 3 (deviation) latch register
		C0h	32	R	RLTC4	Counter 4 (general-purpose 2) latch register
		C4h	32	R	RSTS	Extension status register
		C8h	32	R/W	REST	Error interrupt factor register
		CCh	32	R/W	RIST	Event interrupt factor register
		D0h	32	R	RPLS	Remaining pulse number register
		D4h	32	R	RSPD	Current speed step number register
		D8h	32	R	RSDC	Slow-down point auto calculation value register
		DCh	160	-	-	-
		F0h	32	R/W	RCI	Circular interpolation step number register
F4h	32	R	RCIC	Circular interpolation step counter register		
F8h	32	-	-	-		
FCh	32	R	RIPS	Interpolation status register		
Y	01b	(Same as X-axis)				
Z	10b	(Same as X-axis)				
U	11b	(Same as X-axis)				

5.1.2.8 Z80 communication address map (reduced address)

Axis	A9, A8	A2 to A0	bit	R/W	Name	Description
X	00b	000b	8	R	MSTSB0	Main status [7:0]
			8	W	COMB0	Command
		001b	8	R	MSTSB1	Main status [15:8]
			8	W	COMB1	Axis selection
		010b	8	R	IOPB	General-purpose input/output port
			8	W	OTPB	General-purpose output port
		011b	8	R	SSTSB	Sub status
		100b	8	R/W	BUFB0	Input/output buffer [7:0]
		101b	8	R/W	BUFB1	Input/output buffer [15:8]
		110b	8	R/W	BUFB2	Input/output buffer [23:16]
111b	8	R/W	BUFB3	Input/output buffer [31:24]		
Y	01b	(Same as X-axis)				
Z	10b	(Same as X-axis)				
U	11b	(Same as X-axis)				

5.1.3 Command writing

Axis selection (SELn) and command (COM) should be written in COMW (COMB1, COMB0) address.

COMW															
COMB1								COMB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	SELu	SELz	SELy	SELx	COM							

COMW.COMB1: Axis selection writing area

Write a command to the axis from SELx to SELu, on which "1" is written.

If you set 1 to multiple bits, you can write a same command to the selected multiple axes.

If you select the axes here, you can control all axes with COMW of X-axis alone, without using COMWs of Y, Z, U-axis. If you set 0 to all SELn, only the own axis (the axis selected by A9 and A8 pins) is regarded as selected.

COMW.COMB0: Command writing area.

For the command, see "5.3 Command".

In Z80 communication, after writing 8 bits of axis selection to COMB1 address, write 8 bits of the command to COMB0 address.

Therefore, be sure to use 8 bits write command to write the data to COMB1 address first.

In other communication, 16 bits including axis selection and command are written to COMW address.

Also, in the following cases, a waiting time is required before the next access.

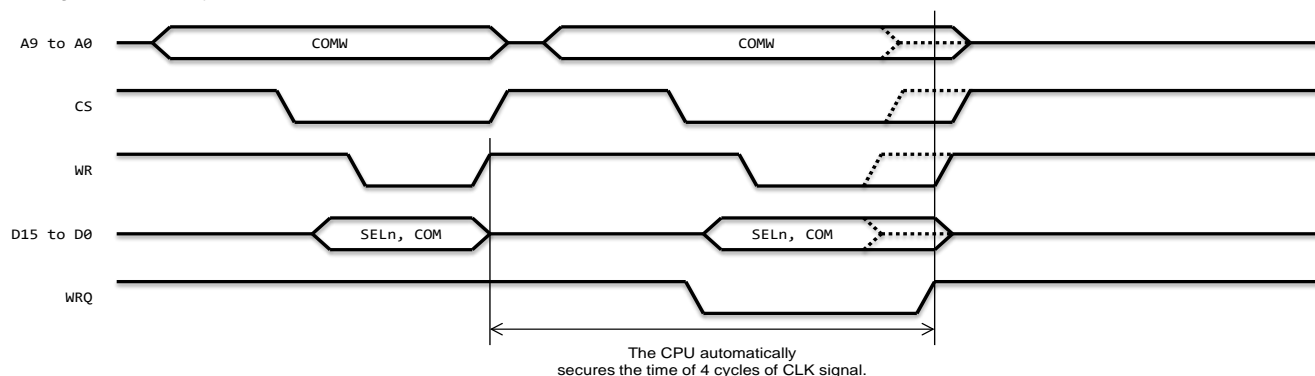
1. From writing a command to writing the next command
2. From writing the register write command to writing the next data to I/O buffer
3. From writing the register read command to reading the data in I/O buffer

If WRQ signal is available in a CPU, the CPU automatically reserves the waiting time.

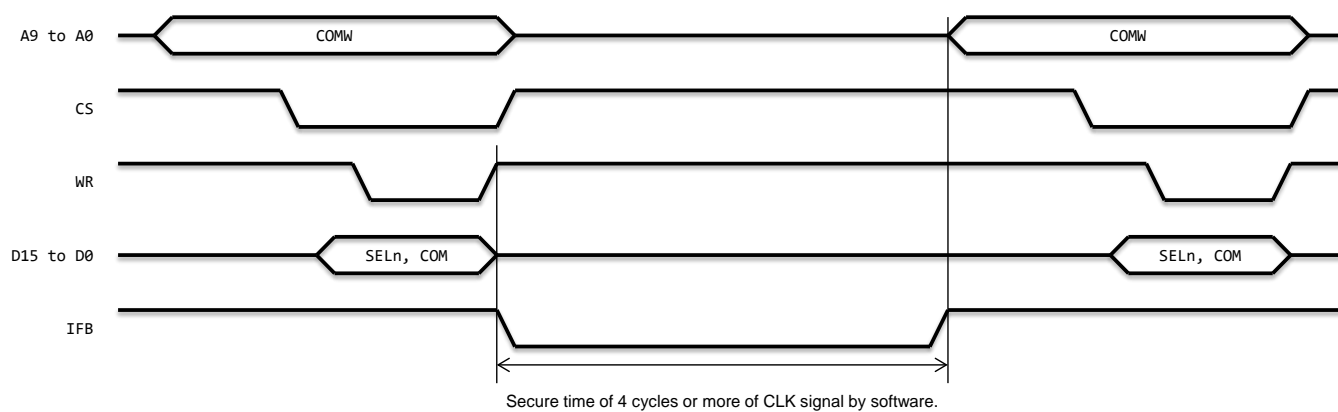
WRQ signal is at L level while both CS and IFB signals are at L level.

IFB signal is at L level during the waiting time that needs to be secured.

If the next access is performed without using WRQ signal, the waveform will be like the dotted line in the figure below, and writing process may fail.



If WRQ signal is not used by the CPU, use software to secure a waiting time of 4 cycles or more for CLK signal.



If WRQ signal is not used by the CPU, check IFB = H level before accessing.

5.1.4 Register writing

In the case of the Full address method, register writing can be selected from either direct access method or indirect access method. For the Reduced address method, use indirect access method.

5.1.4.1 Register writing by direct access method

Access directly the address corresponding to the register.

I/O buffers and register control commands are not used.

Register write data is buffered for 32 bits when writing from the lowest address of the target register.

At the end of the write cycle of the highest address, the buffered register write data is written to the register at once. Therefore, even if the register write data is less than 32 bits, it is necessary to write 32 bits.

With the direct access method, WRQ signal may not be output correctly when writing a lower address.

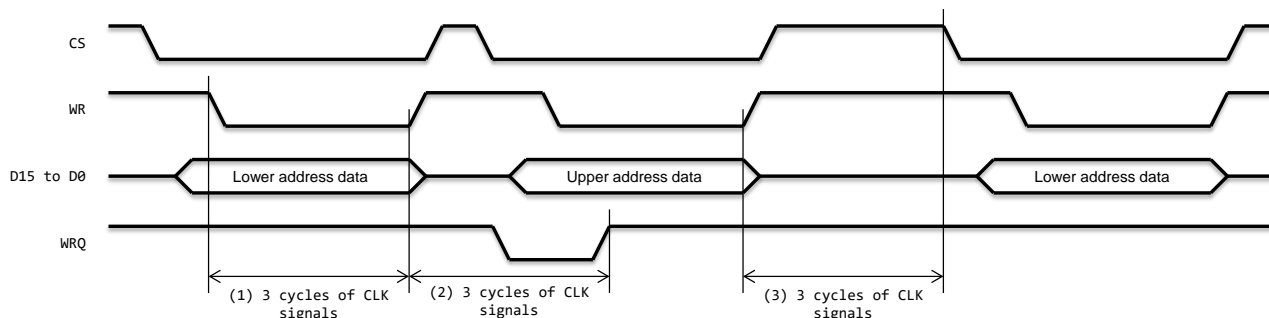
When using a CPU interface other than the 68000 connection, use one of the following methods to avoid the problem.

1. How to adjust the L level output width of WR signal

- (1) Secure an L level output width of at least three cycles of CLK signal for WR signal.
- (2) After writing the data of a lower address, WRQ signal changes to L level.

However, WRQ signal returns to H level before the rising edge of WR signal when writing the data at an upper address.

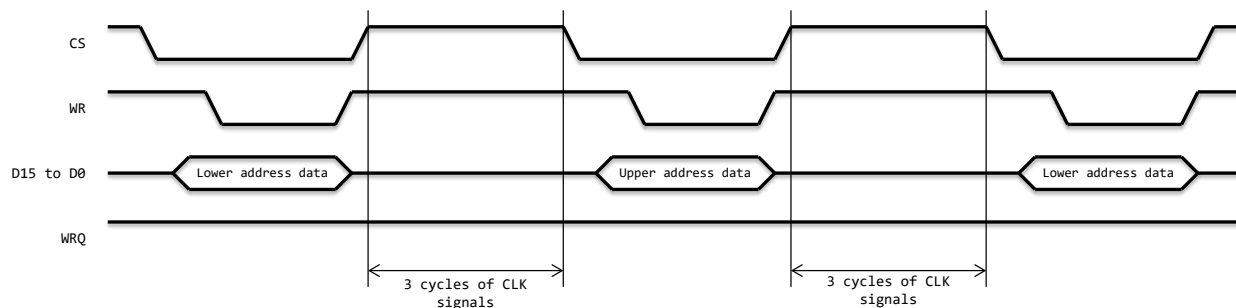
- (3) In addition, after WR = H level of the highest address, secure a waiting time of three cycles or more for CLK signal.



2. How to adjust H level output width of CS signal

Secure an H level output width of three cycles or more of CLK signal for CS signal.

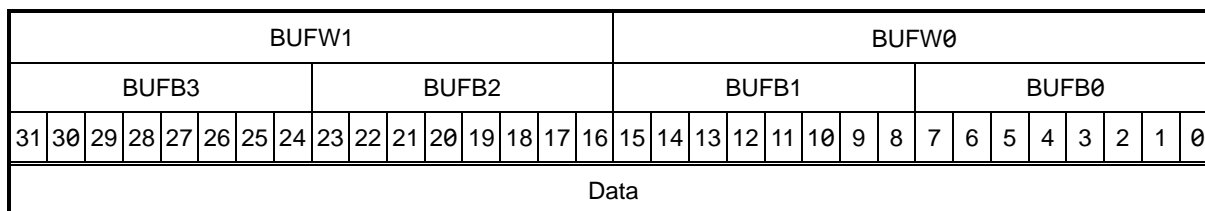
No WRQ signal is output because write process ends while CS = H level.



Please see "Product non-conformity information (Issued No.DB70241-0)" in our website [Japanese page] for any non-conformity phenomenon.

5.1.4.2 Register writing by indirect access method

Register write data (BUF) is written to BUFW1 and BUFW0 addresses.



BUFW1 (BUFB3, BUFB2): Write the upper data.

BUFW0 (BUFB1, BUFB0): Write the lower data

In Z80 communication, register write data is written to BUFB3, BUFB2, BUFB1, BUFB0 addresses.

In other communication, register write data is written to BUFW1 and BUFW0 addresses.

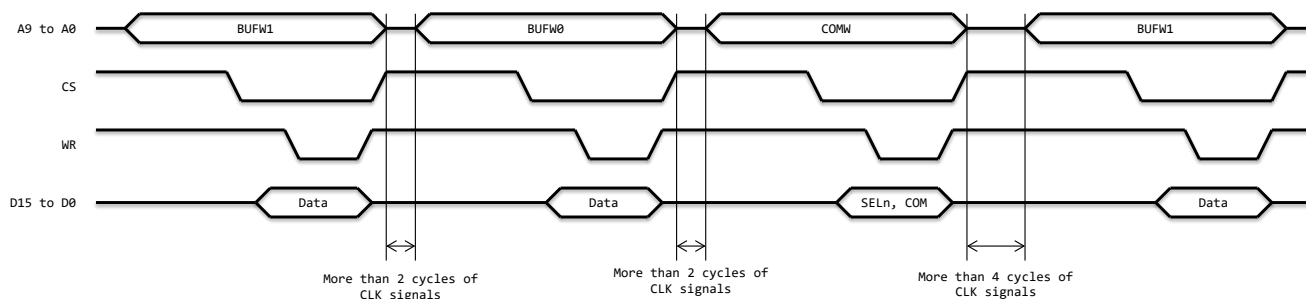
The data to be written to I/O buffer can be any order.

Be sure to write 32 bits of data because the previous write or read data remains in I/O buffer.

Secure a waiting time of two cycles of CLK signal for each data to be written.

After that, when a register write command is written, the register write data is written to the register at once.

After writing the register write command, secure the waiting time for four cycles of CLK signal.



For a register write command, see "5.3.2.10 Register control command".

With the indirect access method, if you specify axes when writing a command, you can write to the same register on multiple axes at the same time.

In this case, set the write data in I/O buffer of each axis.

Software example (H8):

```

var Address = 0x0; // Address is X-axis (Input/output buffer)
var BufferData = 0x000000123; // Input/output buffer is 0000 0123h
OutputBufferData(Address, BufferData); // Write data to PCL6046.
Address = 0x4; // Address is X-axis (Input/output buffer)
BufferData = 0x456789AB; // Input/output buffer is 4567 89ABh
OutputBufferData(Address, BufferData); // Write data to PCL6046.
Address = 0x3; // Address is X-axis (axis selection and command)
var Command = 0x0380; // Axis selection is Y- and X-axes, command is WPRMV (80h)
OutputCommand(Address, Command); // Write command to PCL6046.

```

5.1.5 Register reading

In the case of the full address method, register reading can be selected either from the direct access method and the indirect access method. For the Reduced address method, use the indirect access method.

5.1.5.1 Reading registers by direct access method

Access the address corresponding to the register directly.

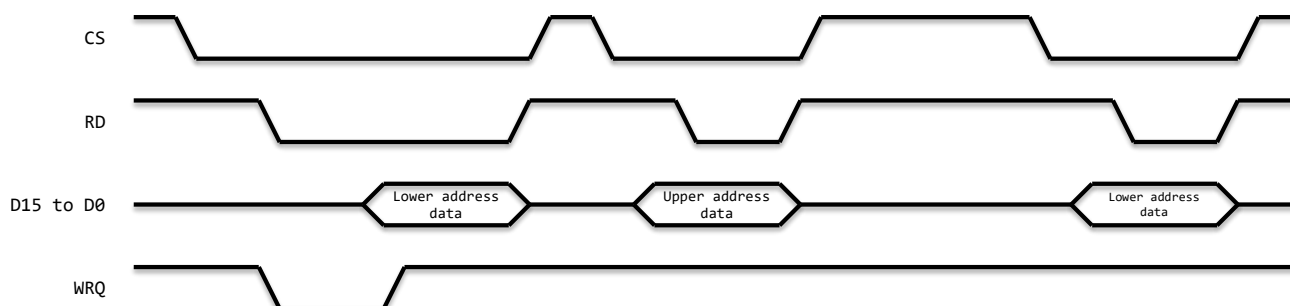
I/O buffers and register control commands are not used.

When reading from the lowest address of the target register, the register read data buffers 32 bits.

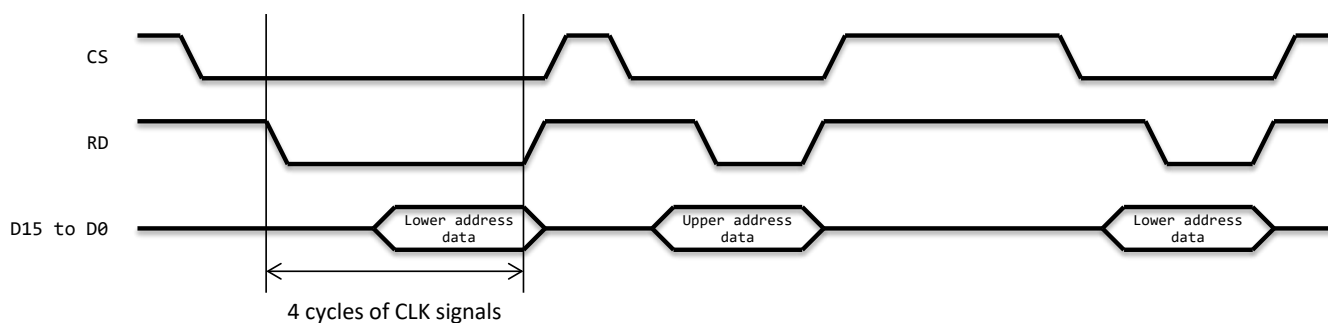
Subsequent read cycles of addresses other than the lowest address read buffered register read data.

Therefore, even if the register read data is less than 32 bits, it is necessary to read 32 bits.

With the direct access method, WRQ signal is output when the lowest address is read.

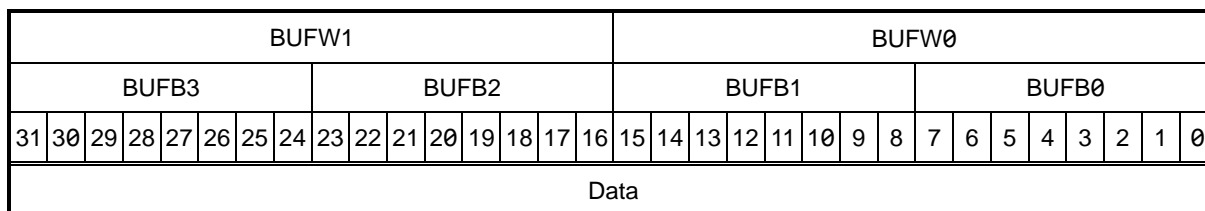


When not using WRQ signal, secure an L level output width of four cycles or more of CLK signal for RD signal.



5.1.5.2 Reading registers by indirect access method

Register read data (BUF) is read from BUFW1 and BUFW0 addresses.



BUFW1 (BUFB3, BUFB2): Read out the upper data.

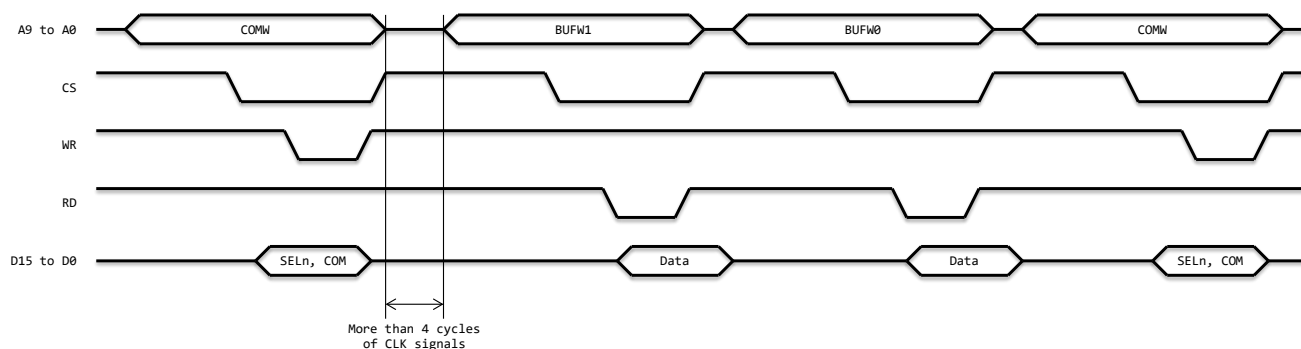
BUFW0 (BUFB1, BUFB0): Read out the lower data.

When register read command is written, the register read data is collectively read to I/O buffer.

In Z80 communication, register read data is read to BUFB3, BUFB2, BUFB1 and BUFB0 addresses.

In other communication, register read data is read to BUFW1 and BUFW0 addresses.

The data read from I/O buffer can be read in any order.



For the register read command, see "5.3.2.10 Register control command".

In the indirect access method, if an axis is specified when writing a command, the command can be read from the same register on multiple axes at the same time.

When reading the command position counter and the like, the same timing value can be read on all axes without latching.

In this case, the read data is set in the I/O buffer for each axis.

Software example (H8):

```

var Address = 0x3; // Address is X-axis (Axis selection & command)
var Command = 0x03C0; // Axis selection is Y- and X-axes, command is RPRMV (C0h)
OutputCommand(Address, Command); // Write command to PCL6046
Address = 0x0; // Address is X-axis (input/output buffer)
var BufferData = InputBufferData(Address); // Read data from PCL6046
Address = 0x4; // Address is Y-axis (input/output buffer)
BufferData = InputBufferData(Address); // Read data from PCL6046

```

5.1.6 Main status reading

The main status (MSTS) is read from the MSTSW address.

MSTSW															
MSTSB1								MSTSB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSTS															

MSTSW (MSTSB1, MSTSB0): Reads the main status.

In Z80 communication, the main status is read from MSTSB1 and MSTSB0 addresses.

For other communications, the main status is read from MSTSW address.

The main status is updated when one or more CLK signal cycles are input while RD = H level and CS = H level.

Therefore, if the read polling cycle is short, the main status may not be updated.

See "5.2.1 Main status (MSTS)" for the main status.

5.1.7 General-purpose output port writing

The output status of general-purpose output port (IOP) is written to OTPW address.

OTPW															
-								OTPB							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	OTP7	OTP6	OTP5	OTP4	OTP3	OTP2	OTP1	OTP0

OTPW (OTPB): Writes the output status of general-purpose output port.

OTP7 to OTP0 bits correspond to P7n to P0n pins.

Writing 1 to the general-purpose output port outputs H level from general-purpose I/O pin.

No signal is output from general-purpose I/O pin set for the general-purpose input port.

If you change the setting of general-purpose output port, written status will be output.

In Z80 communication, the output status of general-purpose output port is written to OTPB address.

For other communications, write the output status of general-purpose output port to OTPW address.

Write 0 to the upper 8 bits of OTPW address.

Also see "5.2.2 Sub status (SSTS) and general-purpose input/output port (IOP)" about general-purpose input/output ports.

5.1.8 Sub-status & general purpose I/O port reading

The input/output status of sub-status (SSTS) and general-purpose input/output port (IOP) is read from SSTS address.

SSTS															
SSTSB								IOPB							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSTS								IOP							

SSTS.SSTSB: Reads the sub-status.

SSTS.IOPB: Reads the I/O status of general-purpose I/O port.

In Z80 communication, the sub-status is read from SSTSB address and the I/O status of general-purpose I/O port is read from IOPB address.

For other communications, the sub-status and the I/O status of general purpose I/O port are read from SSTS address.

The sub-status is updated when one or more CLK signal cycles are input while CS = H level.

See "5.2.2 Sub status (SSTS) and general-purpose input/output port (IOP)" for more information.

5.2 Status & General-purpose I/O Port

There are four statuses in status:

- Main status (MSTS)
- Sub-status (SSTS)
- Extended status (RSTS)
- Interpolation status (RIPS)

We explain the I/O status of general-purpose I/O port together with the sub-status.

5.2.1 Main status (MSTS)

Operating status, interrupt status, comparator establishment status, pre-register confirmation status, and the like can be read.

MSTSW															
MSTSB1								MSTSB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPDF	SPRF	SEOR	SCP5	SCP4	SCP3	SCP2	SPC1	SSC		SINT	SERR	SEND	SENI	SRUN	SSCM

Bit	Name	Description
0	SSCM	The write status of a start command. 0: Operation stopped (or initial status immediately after reset). 1: The start command has been written.
1	SRUN	The operating status of an operation mode. 0: Stopping. H level is output from BSYn pin. 1: Running. L level is output from BSYn pin.
2	SENI	The status in which an operation stop interrupt is generated. 0: No operation stop interrupt occurs. Or RENV2.IEND = 0 is set. 1: Operation stop interrupt is generated. L level can be output from INT pin. If RENV5.MSMR = 0, the bit returns to MST.SENI = 0 within three cycles of CLK signal after reading. If RENV5.MSMR = 1, the bit returns to MST.SENI = 0 after writing a SENIR (2Dh) command.
3	SEND	Operation mode is stopped. 0: Start command has been written (or the initial status immediately after reset). When MST.SRUN bit changes from 1 to 0, the bit changes to MST.SEND = 1. 1: Operation stopped. By writing a start command, the bit returns MST.SEND = 0.
4	SERR	0: No error interrupt occurred. 1: An error interrupt occurred. L level can be output from INT pin. When all bits that are 1 in REST register become 0, the bit returns to MST.SERR = 0.
5	SINT	0: No event interrupt occurred. 1: An event interrupt occurred. L level can be output from INT pin. When all bits that are 1 in RIST register become 0, the bit returns to MST.SINT = 0.

Bit	Name	Description
7, 6	SSC	Sequence number (RMD.MSN) during execution or when stopped. It can be used to control steps of operation blocks when creating the control software. The sequence number does not affect the operation.
8	SCP1	0: The comparison condition of comparator 1 is not satisfied. 1: The comparison condition of comparator 1 is satisfied.
9	SCP2	0: The comparison condition of comparator 2 is not satisfied. 1: The comparison condition of comparator 2 is satisfied.
10	SCP3	0: The comparison condition of comparator 3 is not satisfied. 1: The comparison condition of comparator 3 is satisfied.
11	SCP4	0: The comparison condition of comparator 4 is not satisfied. 1: The comparison condition of comparator 4 is satisfied.
12	SCP5	0: The comparison condition of comparator 5 is not satisfied. 1: The comparison condition of comparator 5 is satisfied.
13	SEOR	The result of an attempt to override a target position. 0: Written to RMV register during operation and the target position override was successful. Or you have not attempted to override the target position. 1: Written to RMV register while stopped and the target position override failed. If RENV5.MSMR = 0, the bit returns to MST.SEOR = 0 within three cycles of CLK signal after reading. If RENV5.MSMR = 1, the bit returns to MST.SEOR = 0 after writing SEORR (2Eh) command.
14	SPRF	0: The 2nd pre-register for continuous operation data is undetermined. 1: The 2nd pre-register for continuous operation data is determined.
15	SPDF	0: The 2nd pre-register for continuous comparison data is undetermined. 1: The 2nd pre-register for continuous comparison data is determined.

The following shows the bit change timing of status.

The dotted line of MST.SEND bit is the initial status immediately after reset.

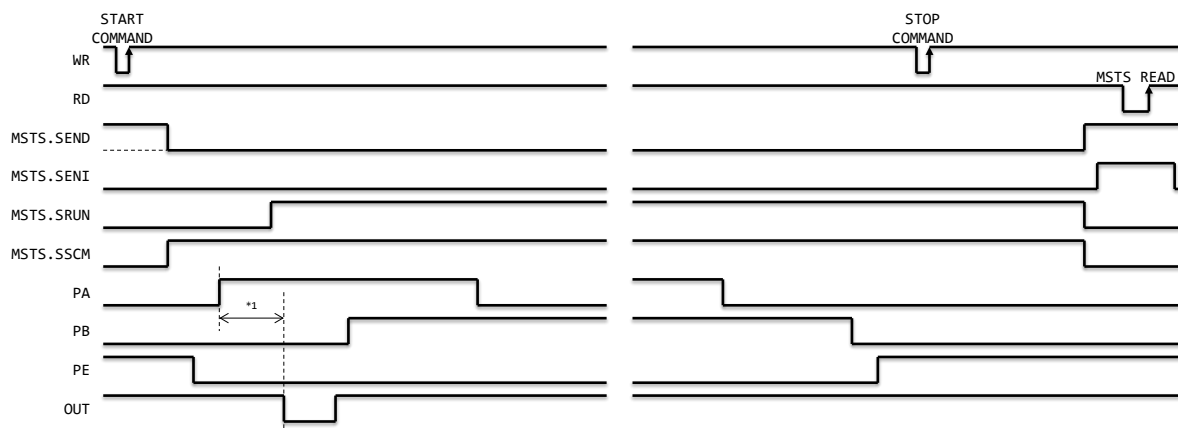
MST.SENI bit changes from MST.SRUN bit with a delay of one CLK signal cycle.

For the change time of other signals, see "7.5 Operation timing".

1. Continuous movement of command control (RMD.MOD = 00h)

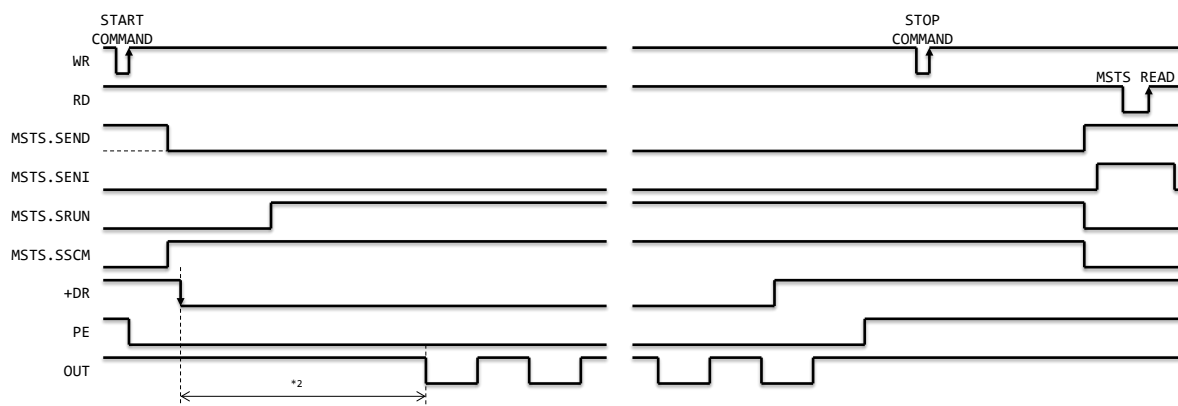


2. Continuous movement of pulser control (RMD.MOD = 01h)



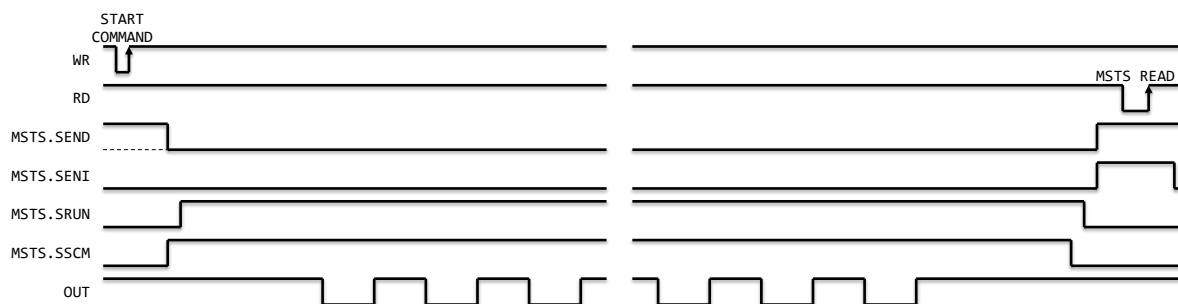
*1 At the longest, there will be a delay of the FH speed cycle.

3. Continuous movement of switch control (RMD.MOD = 02h)



*2 There will be a delay up to the start delay time (TCMDPLS).

4. Incremental movement of positioning control (RMD.MOD = 41h)



5.2.2 Sub status (SSTS) and general-purpose input/output port (IOP)

You can read the input status of an input signal, the speed status during operation, the input/output status of a general-purpose input/output port.

SSTSW															
SSTSB								IOPB							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSD	SORG	SMEL	SPEL	SALM	SFC	SFD	SFU	IOP7	IOP6	IOP5	IOP4	IOP3	IOP2	IOP1	IOP0

Bit	Name	Description
7: 0	IOP7 to IOP0	Input/output status of general-purpose input/output ports, P7n to P0n pins. 0: L level 1: H level
8	SFU	Operation status of acceleration. 0: Other than accelerating 1: Accelerating
9	SFD	Operating status of deceleration. 0: Other than decelerating 1: Decelerating
10	SFC	Constant speed operating status. 0: Other than constant speed 1: Constant speed
11	SALM	Input status of ALM signal. 0: OFF 1: ON The input logic can be changed with RENV1.ALML bit.
12	SPEL	+EL signal input status. 0: OFF 1: ON The input logic can be changed with ELLn pin.
13	SMEL	-EL signal input status. 0: OFF 1: ON The input logic can be changed with ELLn pin.
14	SORG	Input status of ORG signal. 0: OFF 1: ON The input logic can be changed with RENV1.ORGL bit.

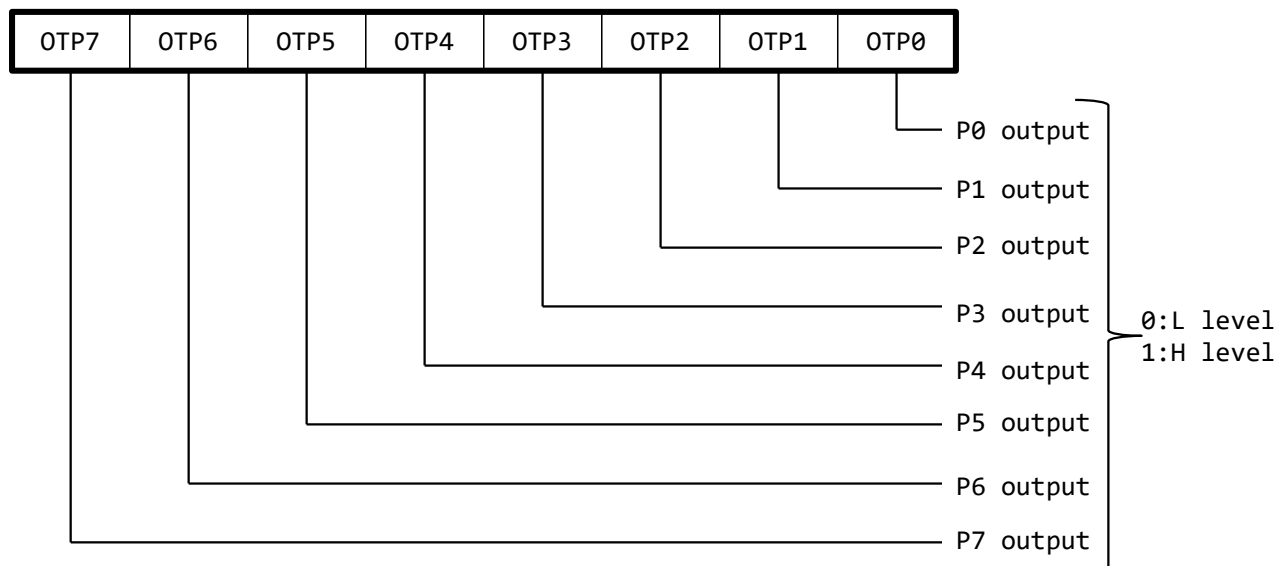
15	SSD	<p>The latch status of SD signal in the operating direction.</p> <p>0: OFF</p> <p>1: ON</p> <p>The input logic can be changed with RENV1.SDL bit.</p> <p>The input status of +SD signal can be read from RSTS.PSDI bit.</p> <p>The input status of -SD signal can be read from RSTS.MSDI bit.</p> <p>The latch status of +SD signal can be read from RSTS.PSDL bit.</p> <p>The latch status of -SD signal can be read from RSTS.MSDL bit.</p>
----	-----	---

During backlash correction and slip correction operations, SSTS.SFU = 0, SSTS.SFD = 0 and SSTS.SFC = 0.

At this time, MSTS.SRUN = 1 can be read.

To check during operation, read the main status.

In addition, the bit layout of general-purpose output port is shown as follows.



5.2.3 Extension status (RSTS)

You can read the signal status, operating status, and operating direction of Input/output pins.

See "5.4.8.1 RSTS: Extension status " for the information of extension status.

5.2.4 Interpolation status (RIPS)

You can read the setting status and operating status of linear interpolation and circular interpolation.

See "5.4.8.2 RIPS: Interpolation status" for the information of interpolation status.

5.3 Commands

This command includes "Operation command" and "Control command".

5.3.1 Operation commands

This command executes start and stop of the operation mode.

5.3.1.1 Start commands

This command starts an operation mode while stopped.

When written during operation, the pre-register of continuous operation data is determined, and the continuous operation start command is set.

COM	Symbol	Description
50h	STAF _L	Start operations with the speed pattern of FL constant speed start.
51h	STAF _H	Start operations with the speed pattern of FH constant speed start.
52h	STAD	Start operations with the speed pattern of high-speed start 1 (from FH speed to deceleration stop).
53h	STAUD	Start operations with the speed pattern of high-speed start 2 (from acceleration to deceleration stop).

See "6.3.1 Speed pattern" for speed patterns.

5.3.1.2 Remaining pulses start commands

These commands are used to output the remaining pulses when the operation mode that stops at RPLS = 0 is stopped halfway (RPLS > 0). If you write while stopped, the operation mode is started without updating the remaining pulse count (RPLS) to the feeding amount (RMV).

Do not write during operation.

COM	Symbol	Description
54h	CNTFL	With the FL constant speed pattern, the operation mode is started without updating RPLS to RMV.
55h	CNTFH	With the FH constant speed pattern, the operation mode is started without updating RPLS to RMV.
56h	CNTD	With the high speed 1 pattern, the operation mode is started without updating RPLS to RMV.
57h	CNTUD	With the high speed 2 pattern, the operation mode is started without updating RPLS to RMV.

See "6.3.1 Speed pattern" for speed patterns.

5.3.1.3 Simultaneous start commands

These commands start the axis waiting for the input of CSTA signal (RSTS.CND = 0010b).

COM	Symbol	Description
06h	CMSTA	CSTA signal is output from CSTA pin.
2Ah	SPSTA	CSTA signal is not output from CSTA pin. Start operation mode while waiting for the input of CSTA signal (RSTS.CND = 0010b).

For CSTA signals, see "6.9.1 Simultaneous start (CSTA)"

5.3.1.4 Speed change commands

These commands change the target speed and speed pattern during operation.

They will be ignored if writing while stopped.

COM	Symbol	Description
40h	FCHGL	Change to FL speed immediately. Speed pattern becomes FL constant speed pattern and changes to FL speed immediately.
41h	FCHGH	Change to FH speed immediately Speed pattern becomes FH constant speed pattern and changes to FH speed immediately.
42h	FSCHL	Decelerate and change to FL speed. Speed pattern becomes the speed pattern of high speed 1, decelerates and changes to FL speed.
43h	FSCHH	Accelerate and change to FH speed. Speed pattern becomes the speed pattern of high speed 2, accelerates and changes to FH speed.

For speed patterns, see "6.3.1 Speed pattern"

5.3.1.5 Stop commands

These commands stop operation during operation.

Also they cancel waiting for CSTA signal input as well as continuous operation by pre-register.

COM	Symbol	Description
49h	STOP	Stop immediately and complete the operation mode.
4Ah	SDSTP	Decelerate-stop to complete the operation mode. If writing during FL constant speed operation, the operation will stop immediately.

For continuous operation, see "6.2.1 Contiguous operation".

5.3.1.6 Simultaneous stop command

This command stops the axis with CSTP signal input enabled (RMD.MSPE = 1) setting.

COM	Symbol	Description
07h	CMSTP	CSTP signal is output from CSTP pin. Multiple axes with CSTP signal input enabled can complete the operation mode. When RENV1.STPM = 1 is set, deceleration stop and the operation mode will be completed.

For CSTP signals, see "6.10 External stop / simultaneous stop".

5.3.1.7 Emergency stop command

This command is written to stop an operation immediately during operation.

COM	Symbol	Description
05h	CMEMG	Stop emergently and cancel the operation mode. Also cancel the continuous operation by the pre-register.

For an emergency stop, see "6.11 Emergency stop"

For continuous operation, see "6.2.1 Contiguous operation".

5.3.2 Control commands

These commands control general-purpose output bits, registers, and counters.

5.3.2.1 NOP command

This command does not affect operations or controls.

COM	Symbol	Description
00h	NOP	This command does not affect the operation. Writing command will be processed.

5.3.2.2 General-purpose output bit control commands

These commands control the general-purpose output port (OPT) bit by bit.

COM	Symbol	Description
10h	P0RST	Write 0 to OPT0 bit to reset P0n pin to L level. When RENV2.P0M = 11b and RENV2.P0L = 0, general-purpose one-shot signal in negative logic is output. * 1
11h	P1RST	Write 0 to OPT1 bit to reset P1n pin to L level. When RENV2.P1M = 11b and RENV2.P1L = 0, general-purpose one-shot signal in negative logic is output. * 1
12h	P2RST	Write 0 to OPT2 bit to reset P2n pin to L level.
13h	P3RST	Write 0 to OPT3 bit to reset P3n pin to L level.
14h	P4RST	Write 0 to OPT4 bit to reset P4n pin to L level.
15h	P5RST	Write 0 to OPT5 bit to reset P5n pin to L level.
16h	P6RST	Write 0 to OPT6 bit to reset P6n pin to L level.
17h	P7RST	Write 0 to OPT7 bit to reset P7n pin to L level.
18h	P0SET	Write 1 to OPT0 bit to set P0n pin to H level. When RENV2.P0L=1 with RENV2.P0M=11b, general-purpose one-shot signal in positive logic is output * 1
19h	P1SET	Write 1 to OPT1 bit to set P1n pin to H level. when RENV2.P1L=1 with RENV2.P1M=11b, general-purpose one-shot signal in positive logic is output * 1
1Ah	P2SET	Write 1 to OPT2 bit to set P2n pin to H level.
1Bh	P3SET	Write 1 to OPT3 bit to set P3n pin to H level.
1Ch	P4SET	Write 1 to OPT4 bit to set P4n pin to H level.
1Dh	P5SET	Write 1 to OPT5 bit to set P5n pin to H level.
1Eh	P6SET	Write 1 to OPT6 bit to set P6n pin to H level.
1Fh	P7SET	Write 1 to OPT7 bit to set P7n pin to H level.

For the batch control of general-purpose output ports, see "5.1.7 General-purpose output port writing".

For the output of general-purpose one-shot signal, see "6.19 General-purpose one shot".

* 1. The output pulse width of a general-purpose one-shot signal is 23 to 25 ms.

5.3.2.3 Reset control command

After a hardware reset, you can use the software reset if you want to reset again.

COM	Symbol	Description
04h	SRST	Reset PCL6046 by software. After writing this command, wait at least 12 cycles of CLK signal before restarting CPU communication.

For resetting, see "6.1 Reset"

5.3.2.4 Counter control commands

This command clears the count value in a counter to 0.

COM	Symbol	Description
20h	CUN1R	Clears the count value of counter 1 (RCUN1) to 0.
21h	CUN2R	Clears the count value of counter 2 (RCUN2) to 0.
22h	CUN3R	Clears the count value of counter 3 (RCUN3) to 0.
23h	CUN4R	Clears the count value of counter 4 (RCUN4) to 0.

See "6.12 Counter" for a counter.

5.3.2.5 ERC signal control commands

These commands control the output of ERC signal, which is one of the control signals for a servo motor.

COM	Symbol	Description
24h	ERCOUT	Outputs ERC signal from ERCn pin.
25h	ERCRST	Resets ERC signal when ERC signal is output with the setting of RENV1.EPW = 111b.

For ERC signal, see "6.8.2 Deviation counter clear (ERC)".

5.3.2.6 Pre-register control commands

These commands control pre-registers.

COM	Symbol	Description
26h	PRECAN	Cancel the determined status in all pre-registers for continuous operation.
27h	PCPCAN	Cancel the determined status in the pre-register for comparator 5 comparison.
2Bh	PRESHF	Shifts the data in all pre-registers for continuous operation.
2Ch	PCPSHF	Shifts the data in the pre-register for comparator 5 comparison.
4Fh	PRSET	Determines the pre-registers for continuous operation as the data for override.

For the pre-register, see "6.2 Pre-register".

For the data for override, see "6.13.6 Bulk override".

5.3.2.7 PCS control command

This command controls the input of PCS signal.

COM	Symbol	Description
28h	STAON	Used in target position override 2. Starts positioning control instead of inputting PCS signal to PCSn pin.

For PCS signals, see "6.4.2 Target position override 2 (PCS)"

5.3.2.8 Counter latch control command

This command controls the counter latch.

COM	Symbol	Description
29h	LTCH	Latches RCUN1 to 4 register values to RLTC1 to 4 registers instead of inputting LTC signal to LTCn pin.

For LTC signal, see "6.12.3 Counter latch".

5.3.2.9 Interrupt control commands

This command clears the interrupt bit in the main status.

COM	Symbol	Description
2Dh	SENIR	Clears to MST.SENI = 0.
2Eh	SEORR	Clears to MST.SEOR = 0.

For interrupt control, see "6.18 Interrupt request (INT)"

5.3.2.10 Register control commands

The indirect access method uses register control commands to read and write registers and pre-registers.

No	Description	Register					2nd pre-register				
		Name	Read command		Write command		Name	Read command		Write command	
			COMB0	Symbol	COMB0	Symbol		COMB0	Symbol	COMB0	Symbol
1	Feed amount (target position)	RMV	D0h	RRMV	90h	WRMV	PRMV	C0h	RPRMV	80h	WPRMV
2	FL speed step number	RFL	D1h	RRFL	91h	WRFL	PRFL	C1h	RPRFL	81h	WPRFL
3	FH speed step number	RFH	D2h	RRFH	92h	WRFH	PRFH	C2h	RPRFH	82h	WPRFH
4	Acceleration rate	RUR	D3h	RRUR	93h	WRUR	PRUR	C3h	RPRUR	83h	WPRUR
5	Deceleration rate	RDR	D4h	RRDR	94h	WRDR	PRDR	C4h	RPRDR	84h	WPRDR
6	Speed magnification rate	RMG	D5h	RRMG	95h	WRMG	PRMG	C5h	RPRMG	85h	WPRMG
7	Slow-down point	RDP	D6h	RRDP	96h	WRDP	PRDP	C6h	RPRDP	86h	WPRDP
8	Operation mode	RMD	D7h	RRMD	97h	WRMD	PRMD	C7h	RPRMD	87h	WPRMD
9	Circular interpolation center	RIP	D8h	RRIP	98h	WRIP	PRIP	C8h	RPRIP	88h	WPRIP
10	S-curve section in acceleration	RUS	D9h	RRUS	99h	WRUS	PRUS	C9h	RPRUS	89h	WPRUS
11	S-curve section in deceleration	RDS	DAh	RRDS	9Ah	WRDS	PRDS	CAh	RPRDS	8Ah	WPRDS
12	FA speed step number	RFA	DBh	RRFA	9Bh	WRFA	-	-	-	-	-
13	Environment setting 1	RENV1	DCh	RRENV1	9Ch	WRENV1	-	-	-	-	-
14	Environment setting 2	RENV2	DDh	RRENV2	9Dh	WRENV2	-	-	-	-	-
15	Environment setting 3	RENV3	DEh	RRENV3	9Eh	WRENV3	-	-	-	-	-
16	Environment setting 4	RENV4	DFh	RRENV4	9Fh	WRENV4	-	-	-	-	-
17	Environment setting 5	RENV5	E0h	RRENV5	A0h	WRENV5	-	-	-	-	-
18	Environment setting 6	RENV6	E1h	RRENV6	A1h	WRENV6	-	-	-	-	-
19	Environment setting 7	RENV7	E2h	RRENV7	A2h	WRENV7	-	-	-	-	-
20	Counter 1 (command position)	RCUN1	E3h	RRCUN1	A3h	WRCUN1	-	-	-	-	-
21	Counter 2 (general-purpose 1)	RCUN2	E4h	RRCUN2	A4h	WRCUN2	-	-	-	-	-
22	Counter 3 (deviation)	RCUN3	E5h	RRCUN3	A5h	WRCUN3	-	-	-	-	-
23	Counter 4 (general-purpose 2)	RCUN4	E6h	RRCUN4	A6h	WRCUN4	-	-	-	-	-
24	Data for comparator 1	RCMP1	E7h	RRCMP1	A7h	WRCMP1	-	-	-	-	-
25	Data for comparator 2	RCMP2	E8h	RRCMP2	A8h	WRCMP2	-	-	-	-	-
26	Data for comparator 3	RCMP3	E9h	RRCMP3	A9h	WRCMP3	-	-	-	-	-
27	Data for comparator 4	RCMP4	EAh	RRCMP4	AAh	WRCMP4	-	-	-	-	-
28	Data for comparator 5	RCMP5	EBh	RRCMP5	ABh	WRCMP5	PRCP5	CBh	RPRCP5	8Bh	WPRCP5
29	Event interrupt request	RIRQ	ECh	RRIRQ	ACh	WRIRQ	-	-	-	-	-
30	Counter 1 (command position) latch	RLTC1	EDh	RRLTC1	-	-	-	-	-	-	-
31	Counter 2 (general-purpose 1) latch	RLTC2	EEh	RRLTC2	-	-	-	-	-	-	-
32	Counter 3 (deviation) latch	RLTC3	EFh	RRLTC3	-	-	-	-	-	-	-
33	Counter 4 (general-purpose 2) latch	RLTC4	F0h	RRLTC4	-	-	-	-	-	-	-
34	Extension status	RSTS	F1h	RRSTS	-	-	-	-	-	-	-
35	Error interrupt factor	REST	F2h	RREST	B2h	WREST	-	-	-	-	-
36	Event interrupt factor	RIST	F3h	RRIST	B3h	WRIST	-	-	-	-	-
37	Remaining pulse number	RPLS	F4h	RRPLS	-	-	-	-	-	-	-
38	Current speed step number	PSPD	F5h	RPSPD	-	-	-	-	-	-	-
39	Slow-down point auto calculation value	RSDC	F6h	RRSDC	-	-	-	-	-	-	-
40	Number of circular interpolation steps	RCI	FCh	RRCI	BCh	WRCI	PRCI	CCh	RPRCI	8Ch	WPRCI
41	Circular interpolation step counter	RCIC	FDh	RRCIC	-	-	-	-	-	-	-
42	Interpolation status	RIPS	FFh	RRIPS	-	-	-	-	-	-	-

Usually, writing to the register or to 1st pre-register is done via 2nd pre-register.

You cannot read 1st pre-register.

For pre-registers, see "6.2 Pre-register".

5.4 Registers

There are eight major types and 42 registers.

For the pre-register, see "6.2 Pre-register".

No.	Details	Name	Range	R/W	Pre-register	Remarks
1	Feed amount (target position)	RMV	-2,147,483,648 to +2,147,483,647	R/W	PRMV	Position control
2	FL speed step number	RFL	1 to 65,535	R/W	PRFL	Speed control
3	FH speed step number	RFH	1 to 65,535	R/W	PRFH	Speed control
4	Acceleration rate	RUR	1 to 65,535	R/W	PRUR	Speed control
5	Deceleration rate	RDR	0 to 65,535	R/W	PRDR	Speed control
6	Speed magnification rate	RMG	2 to 4,095	R/W	PRMG	Speed control
7	Slow-down point	RDP	-8,388,608 to +8,388,607 0 to 16,777,215	R/W	PRDP	Position control
8	Operation mode	RMD	(4 byte)	R/W	PRMD	Environment setting
9	Circular interpolation center	RIP	-2,147,483,648 to +2,147,483,647	R/W	PRIP	Position control
10	S-curve section in acceleration	RUS	0 to 32,767	R/W	PRUS	Speed control
11	S-curve section in deceleration	RDS	0 to 32,767	R/W	PRDS	Speed control
12	FA speed step number	RFA	1 to 65,535	R/W	-	Environment setting
13	Environment setting 1	RENV1	(4 byte)	R/W	-	Environment setting
14	Environment setting 2	RENV2	(4 byte)	R/W	-	Environment setting
15	Environment setting 3	RENV3	(4 byte)	R/W	-	Environment setting
16	Environment setting 4	RENV4	(4 byte)	R/W	-	Environment setting
17	Environment setting 5	RENV5	(4 byte)	R/W	-	Environment setting
18	Environment setting 6	RENV6	(4 byte)	R/W	-	Environment setting
19	Environment setting 7	RENV7	(4 byte)	R/W	-	Environment setting
20	Counter 1 (command position)	RCUN1	-2,147,483,648 to +2,147,483,647	R/W	-	Counter
21	Counter 2 (general-purpose 1)	RCUN2	-2,147,483,648 to +2,147,483,647	R/W	-	Counter
22	Counter 3 (deviation)	RCUN3	-32,768 to +32,767	R/W	-	Counter
23	Counter 4 (general-purpose 2)	RCUN4	-2,147,483,648 to +2,147,483,647	R/W	-	Counter
24	Comparison data for comparator 1	RCMP1	-2,147,483,648 to +2,147,483,647	R/W	-	Comparator
25	Comparison data for comparator 2	RCMP2	-2,147,483,648 to +2,147,483,647	R/W	-	Comparator

No.	Details	Name	Range	R/W	Pre-register	Remarks
26	Comparison data for comparator 3	RCMP3	-2,147,483,648 to +2,147,483,647	R/W	-	Comparator
27	Comparison data for comparator 4	RCMP4	-2,147,483,648 to +2,147,483,647	R/W	-	Comparator
28	Comparison data for comparator 5	RCMP5	-2,147,483,648 to +2,147,483,647	R/W	PRCP5	Comparator
29	Event interrupt request	RIRQ	(4 byte)	R/W	-	Interrupt control
30	Counter 1 (command position) latch	RLTC1	-2,147,483,648 to +2,147,483,647	R	-	Counter latch
31	Counter 2 (general-purpose 1) latch	RLTC2	-2,147,483,648 to +2,147,483,647	R	-	Counter latch
32	Counter 3 (deviation) latch	RLTC3	-32,768 to +32,767 0 to 65,535	R	-	Counter latch
33	Counter 4 (general-purpose 2) latch	RLTC4	-2,147,483,648 to +2,147,483,647	R	-	Counter latch
34	Extension status	RSTS	(4 byte)	R		Status display
35	Error interrupt factor	REST	(4 byte)	R/W	-	Interrupt control
36	Event interrupt factor	RIST	(4 byte)	R/W	-	Interrupt control
37	Remaining pulse number	RPLS	0 to 2,147,483,647	R	-	Position control
38	Current speed step number	PSPD	(4 byte)	R	-	Speed control
39	Slow-down point auto calculation value	RSDC	0 to 16,777,215	R	-	Position control
40	Number of circular interpolation steps	RCI	0 to 4,294,967,295	R/W	PRCI	Position control
41	Circular interpolation step counter	RCIC	0 to 4,294,967,295	R	-	Position control
42	Interpolation status	RIPS	(4 byte)	R	-	Status display

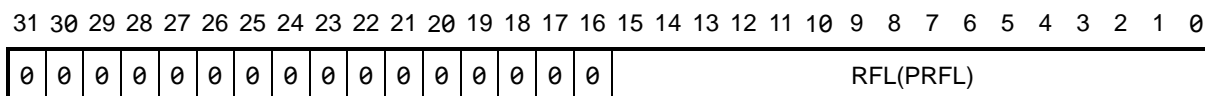
Warning

During operation, do not write data to environment setting registers which do not have pre-registers.
That may cause unintended behavior.

5.4.1 Speed control registers

These registers are for speed controls.

5.4.1.1 RFL(PRFL): FL speed step number



A7 to A0 address of direct access method [68000, H8]: RFL(B8h), PRFL(F0h)

A7 to A0 address of direct access method [8086, Z80]: RFL(44h), PRFL(0Ch)

Register control command of indirect access method: RRFL(D1h), RPRFL(C1h), WRFL(91h), WPRFL(81h)

Register to set FL speed (initial speed, stop speed) by speed step number.

PRFL register is the pre-register of RFL register.

$$FL[pps] = RFL \times \frac{f_{CLK}[Hz]}{(RMG + 1) \times 65,536}$$

$$= RFL \times MG$$

$$RFL = FL[pps] \times \frac{(RMG + 1) \times 65,536}{f_{CLK}[Hz]}$$

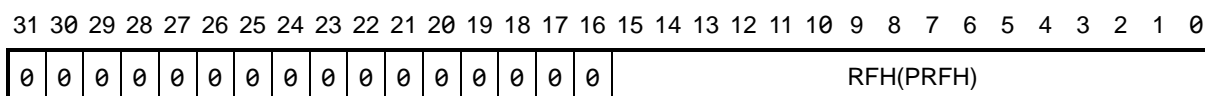
FL: FL speed

MG: Speed magnification

When the speed magnification is set to 1x, the setting value of RFL register becomes the FL speed [pps] as it is.

The setting range is 1 to 65,535. Be sure to set 1 or higher.

5.4.1.2 RFH(PRFH): FH speed step number



A7 to A0 address of direct access method [68000, H8]: RFH(B4h), PRFH(ECh)

A7 to A0 address of direct access method [8086, Z80]: RFH(48h), PRFH(10h)

Register control command of indirect access method: RRFH(D2h), RPRFH(C2h), WRFH(92h), WPRFH(82h)

Register to set FH speed (operating speed, maximum speed) by speed step number.

PRFH register is the pre-register of RFH register.

$$FH[pps] = RFH \times \frac{f_{CLK}[Hz]}{(RMG + 1) \times 65,536}$$

$$= RFH \times MG$$

$$RFH = FH[pps] \times \frac{(RMG + 1) \times 65,536}{f_{CLK}[Hz]}$$

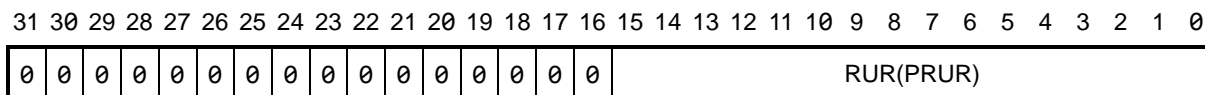
FH: FH speed

MG: Speed magnification

When the speed magnification is set to 1x, the setting value of RFH register becomes the FH speed [pps] as it is.

The setting range is 1 to 65,535. Be sure to set 1 or higher.

5.4.1.3 RUR(PRUR): Acceleration rate



A7 to A0 address of direct access method [68000, H8]: RUR(B0h), PRUR(E8h)

A7 to A0 address of direct access method [8086, Z80]: RUR(4Ch), PRUR(14h)

Register control command of indirect access method: RRUR(D3h), RPRUR(C3h), WRUR(93h), WPRUR(83h)

Register to set the acceleration rate.

PRUR register is the pre-register of RUR register.

The relationship between the acceleration time and RUR register is as follows, depending on the RMD.MSMD bit and the RUS register value.

1. Linear acceleration (RMD.MSMD = 0)

$$TU[s] = \frac{(RFH - RFL) \times (RUR + 1) \times 4}{f_{CLK}[Hz]} \qquad RUR = \frac{f_{CLK}[Hz] \times TU[s]}{(RFH - RFL) \times 4} - 1$$

TU: Acceleration time

2. S-curve acceleration without linear section (RMD.MSMD = 1 and RUS = 0)

$$TU[s] = \frac{(RFH - RFL) \times (RUR + 1) \times 8}{f_{CLK}[Hz]} \qquad RUR = \frac{f_{CLK}[Hz] \times TU[s]}{(RFH - RFL) \times 8} - 1$$

TU: Acceleration time

3. S-curve acceleration with linear section (RMD.MSMD = 1 and RUS > 0)

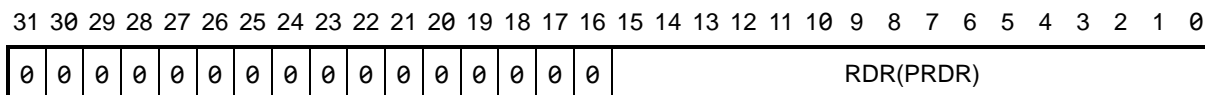
$$TU[s] = \frac{(RFH - RFL + 2 \times RUS) \times (RUR + 1) \times 4}{f_{CLK}[Hz]} \qquad RUR = \frac{f_{CLK}[Hz] \times TU[s]}{(RFH - RFL + 2 \times RUS) \times 4} - 1$$

TU: Acceleration time

The larger the setting value of RUR register, the longer the acceleration time and the slower the acceleration.

The setting range is 1 to 65,535. Be sure to set 1 or higher.

5.4.1.4 RDR(PRDR): Deceleration rate



A7 to A0 address of direct access method [68000, H8]: RDR(ACh), PRDR(E4h)

A7 to A0 address of direct access method [8086, Z80]: RDR(50h), PRDR(18h)

Register control command of indirect access method: RRDR(D4h), RPRDR(C4h), WRDR(94h), WPRDR(84h)

Register to set the deceleration rate.

PRDR register is the pre-register of RDR register.

When using automatic slow-down point setting (RMD.MSDP = 0), the following restrictions will apply:

When using constant synthesized speed control (RMD.MIPF = 1) for linear interpolation 1 and circular interpolation:

Set the deceleration time = acceleration time.

When other than the above:

Satisfy the deceleration time ≤ acceleration time x 2.

If you cannot meet the above restrictions, use the manual slow-down point setting (RMD.MSDP = 1).

The relationship between the deceleration time and RDR register is as follows, depending on the RMD.MSMD bit and the RDS register value.

1. Linear deceleration (RMD.MSMD = 0)

$$TD[s] = \frac{(RFH - RFL) \times (RDR + 1) \times 4}{f_{CLK}[Hz]}$$

$$RDR = \frac{f_{CLK}[Hz] \times TD[s]}{(RFH - RFL) \times 4} - 1$$

TD: Deceleration time

2. S-curve deceleration without a linear section (RMD.MSMD = 1 and RDS = 0)

$$TD[s] = \frac{(RFH - RFL) \times (RDR + 1) \times 8}{f_{CLK}[Hz]}$$

$$RDR = \frac{f_{CLK}[Hz] \times TD[s]}{(RFH - RFL) \times 8} - 1$$

TD: Deceleration time

3. S-curve deceleration with a linear section (RMD.MSMD = 1 and RDS = 0)

$$TD[s] = \frac{(RFH - RFL + 2 \times RDS) \times (RDR + 1) \times 4}{f_{CLK}[Hz]}$$

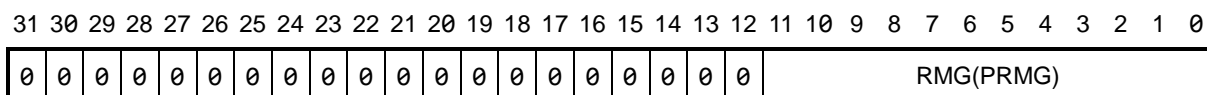
$$RDR = \frac{f_{CLK}[Hz] \times TD[s]}{(RFH - RFL + 2 \times RDS) \times 4} - 1$$

TD: Deceleration time

The larger the setting value of RDR register, the longer the deceleration time and the slower the deceleration.

The setting range is 0 to 65,535. If you set 0, the setting value of RUR register will be used as well.

5.4.1.5 RMG(PRMG): Speed magnification



A7 to A0 address of direct access method [68000, H8]: RMG(A8h), PRMG(E0h)

A7 to A0 address of direct access method [8086, Z80]: RMG(54h), PRMG(1Ch)

Register control command of indirect access method: RRMG(D5h), RPRMG(C5h), WRMG(95h), WPRMG(85h)

Register that sets the relationship between the speed step number and the actual speed.

PRMG register is the pre-register of RMG register.

$$MG = \frac{f_{CLK}[Hz]}{(RMG + 1) \times 65,536}$$

$$RMG = \frac{f_{CLK}[Hz]}{MG \times 65,536} - 1$$

MG: Speed magnification

The following is an example of the speed magnification setting at $f_{CLK} = 19.6608$ MHz.

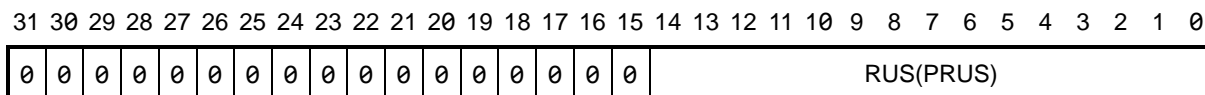
Setting value	Speed magnification (MG) [Times]	Actual speed range [pps]	Setting value	Speed magnification (MG) [Times]	Actual speed range [pps]
2999 (BB7h)	0.1	0.1 to 6,553.5	59 (03Bh)	5	5 to 327,675
1499 (5DBh)	0.2	0.2 to 13,107.0	29 (01Dh)	10	10 to 655,350
599 (257h)	0.5	0.5 to 32,767.5	14 (00Eh)	20	20 to 1,310,700
299 (12Bh)	1	1 to 65,535	5 (005h)	50	50 to 3,276,750
149 (095h)	2	2 to 131,070	2 (002h)	100	100 to 6,553,500

The higher the magnification, the coarser the interval between the set speeds.

Please use the lowest possible magnification according to the actual speed range.

The setting range is 2 to 4,095. Be sure to set 2 or higher.

5.4.1.6 RUS(PRUS): S-curve acceleration section



A7 to A0 address of direct access method [68000, H8]: RUS(98h), PRUS(D0h)

A7 to A0 address of direct access method [8086, Z80]: RUS(64h), PRUS(2Ch)

Register control command of indirect access method: RRUS(D9h), RPRUS(C9h), WRUS(99h), WPRUS(89h)

Register to set the S-curve section of S-curve acceleration.

PRUS register is the pre-register of RUS register.

$$S_{SU}[pps] = RUS \times \frac{f_{CLK}[Hz]}{(RMG + 1) \times 65,536}$$

$$= RUS \times MG$$

$$RUS = S_{SU}[pps] \times \frac{(RMG + 1) \times 65,536}{f_{CLK}[Hz]}$$

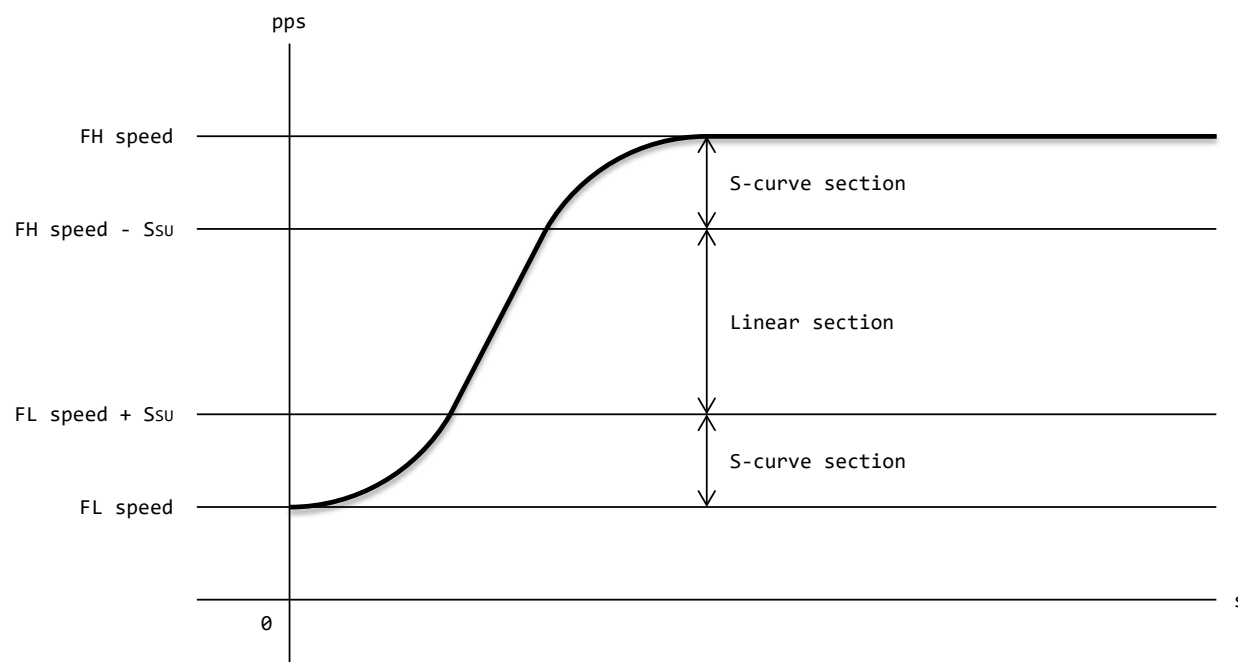
S_{SU} : S-curve acceleration section

MG: Speed magnification

It is enabled by setting S-curve acceleration / deceleration (RMD.MSMD = 1).

From FL speed to FL speed + S_{SU} and from FH speed - S_{SU} to FH speed are the sections that accelerate in an S-curve.

From FL speed + S_{SU} to FH speed - S_{SU} is the section that accelerates linearly.

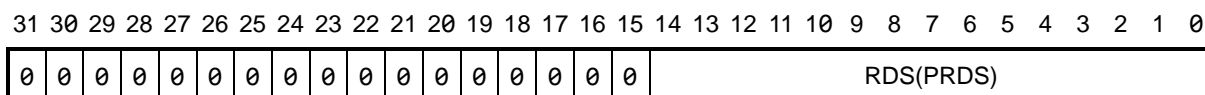


The smaller the setting value of RUS register, the shorter the S-curve acceleration section, and the closer to linear acceleration.

The setting range is 0 to 32,767.

If you set to 0, $\frac{RFH-RFL}{2}$ is substituted for complete S-curve acceleration with no linear acceleration section.

5.4.1.7 RDS(PRDS): S-curve deceleration section



A7 to A0 address of direct access method [68000, H8]: RDS(94h), PRDS(CCh)

A7 to A0 address of direct access method [8086, Z80]: RDS(68h), PRDS(30h)

Register control command of indirect access method: RRDS(DAh), RPRDS(CAh), WRDS(9Ah), WPRDS(8Ah)

Register to set the S-curve section in S-curve deceleration.

PRDS register is the pre-register of RDS register.

$$S_{SD}[pps] = RDS \times \frac{f_{CLK}[Hz]}{(RMG + 1) \times 65,536}$$

$$= RDS \times MG$$

$$RDS = S_{SD}[pps] \times \frac{(RMG + 1) \times 65,536}{f_{CLK}[Hz]}$$

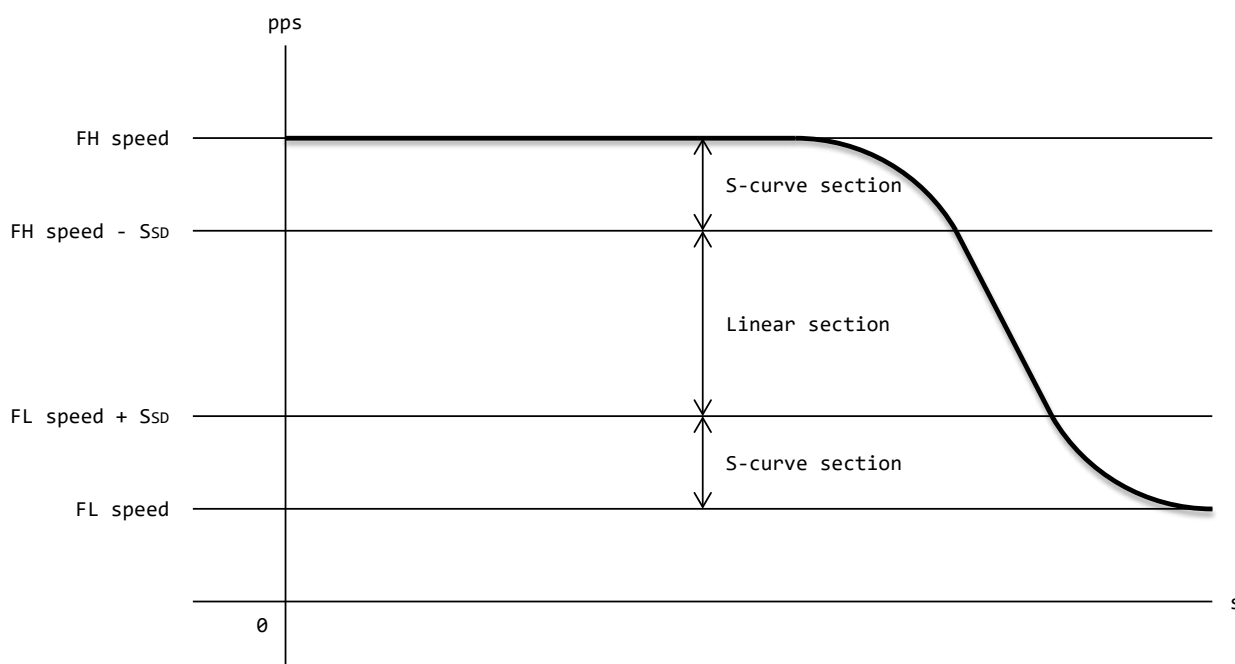
S_{SD} : S-curve deceleration section

MG: Speed magnification

It is enabled by setting S-curve acceleration / deceleration (RMD.MSMD = 1).

From FH speed to FH speed - S_{SD} and from FL speed + S_{SD} to FL speed are the sections that decelerate in an S-curve.

From FH speed - S_{SD} to FL speed + S_{SD} is the section that decelerates linearly.

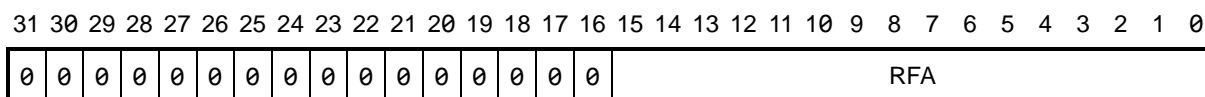


The smaller the setting value of RDS register, the shorter the S-curve deceleration section, and the closer to linear deceleration.

The setting range is 0 to 32,767.

If you set to 0, $\frac{RFH-RFL}{2}$ is substituted for complete S-curve deceleration without a linear deceleration section.

5.4.1.8 RFA: FA speed step number



A7 to A0 address of direct access method [68000, H8]: RFA(90h)

A7 to A0 address of direct access method [8086, Z80]: RFA(6Ch)

Register control command of indirect access method: RRFA(DBh), WRFA(9Bh)

Register to set FA speed (backlash correction speed and slip correction speed) by speed step number.

It is also used for the reverse speed in an origin return operation.

$$FA[pps] = RFA \times \frac{f_{CLK}[Hz]}{(RMG + 1) \times 65,536}$$

$$RFA = FA[pps] \times \frac{(RMG + 1) \times 65,536}{f_{CLK}[Hz]}$$

$$= RFA \times MG$$

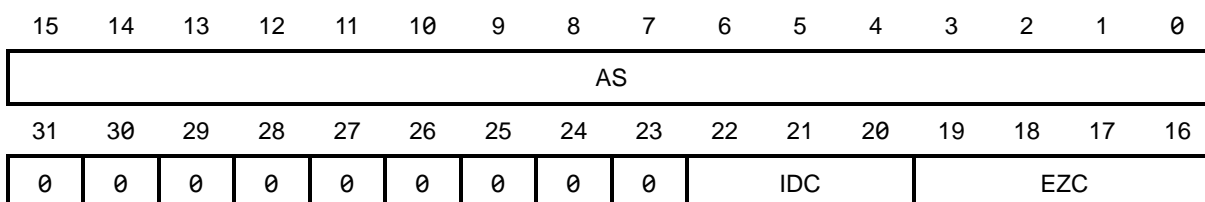
FA: FA speed

MG: Speed magnification

When the speed magnification is set to 1x, the setting value in RFA register becomes FA speed [pps] as it is.

The setting range is 1 to 65,535. Be sure to set 1 or higher.

5.4.1.9 RSPD: Current speed step number



A7 to A0 address of direct access method [68000, H8]: RSPD(28h)

A7 to A0 address of direct access method [8086, Z80]: RSPD(D8h)

Register control command of indirect access method: RRSPD(F5h)

This register acquires the current speed step number, EZ signal count value, and idling count value.

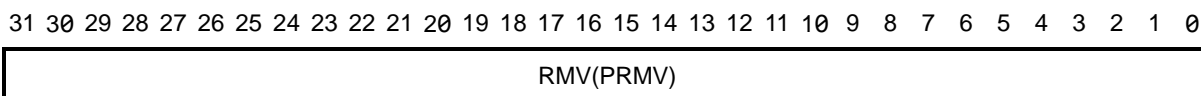
Read-only.

Bit	Name	Description
15:0	AS	Current speed can be read by the step number (the same unit as RFL register and RFH register). It is 0 when stopped. In pulser controls, it is the step number of set speed (RFH register value).
19:16	EZC	The input count value of EZ signal used for origin return control and sensor control can be read. It is a down-counter. Initial value is the value of RENV3.EZD bit.
22:20	IDC	The idling count value used for idling control can be read. It is a down-counter. Initial value is the value of RENV5.IDL bit.
31:23	0	Always acquire 0.

5.4.2 Position control register

This register is for position control.

5.4.2.1 RMV(PRMV): Feed amount (target position)



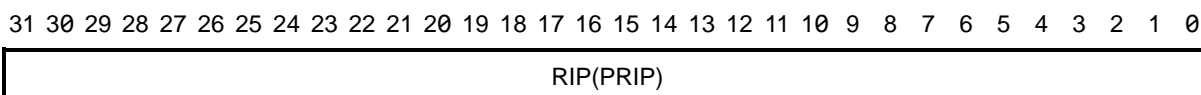
A7 to A0 address of direct access method [68000, H8]: RMV(BCh), PRMV(F4h)
 A7 to A0 address of direct access method [8086, Z80]: RMV(40h), PRMV(08h)
 Register control command of indirect access method: RRMV(D0h), RPRMV(C0h), WRMV(90h), WPRMV(80h)

Register to set the amount of movement (target position).

PRMV register is the pre-register of RMV register.

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.2.2 RIP(PRIP): Circular interpolation center



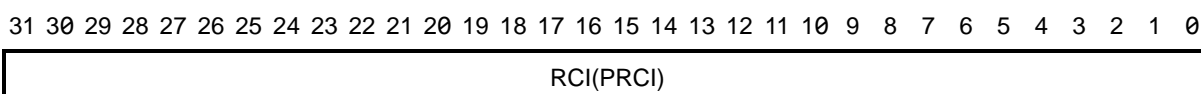
A7 to A0 address of direct access method [68000, H8]: RIP(9Ch), PRIP(D4h)
 A7 to A0 address of direct access method [8086, Z80]: RIP(60h), PRIP(28h)
 Register control command of indirect access method: RRIP(D8h), RPRIP(C8h), WRIP(98h), WPRIP(88h)

Register to set the center position of circular interpolation or the main axis movement amount of linear interpolation 2 (RMV of the axis of maximum movement amount).

PRIP register is the pre-register of RIP register.

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.2.3 RCI(PRCI): Number of circular interpolation steps



A7 to A0 address of direct access method [68000, H8]: RCI(0Ch), PRCI(C4h)
 A7 to A0 address of direct access method [8086, Z80]: RCI(F0h), PRCI(38h)
 Register control command of indirect access method: RRCI(FCh), RPRCI(CCh), WRCI(BCh), WPRCI(8Ch)

Register to set the number of circular interpolation steps of the control axis.

It is not included in U-axis, which does not become the control axis for circular interpolation.

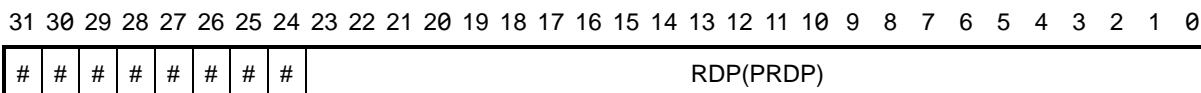
PRCI register is the pre-register of RCI register.

By setting the number of circular interpolation steps, you can use the slow-down point when performing deceleration control in circular interpolation.

The setting range is 0 to 4,294,967,295.

For the number of circular interpolation steps, see "6.3.5 Number of circular interpolation steps".

5.4.2.4 RDP(PRDP): Slow-down point



A7 to A0 address of direct access method [68000, H8]: RDP(A4h), PRDP(DCh)

A7 to A0 address of direct access method [8086, Z80]: RDP(58h), PRDP(20h)

Register control command of indirect access method: RRDP(D6h), RPRDP(C6h), WRDP(96h), WPRDP(86h)

Register to set the slow-down point (deceleration start point).

PRDP register is the pre-register of RDP register.

The contents differ depending on the setting of RMD.MSDP bit.

<Automatic slow-down point setting (RMD.MSDP = 0)>

Set the offset of slow-down point, which will be set automatically.

Starts deceleration when the RPLS register value becomes less than or equal to the RSDC register value.

If RDP register has a positive number, deceleration starts earlier. After deceleration ends, a motor operates at FL speed and then stops.

If RDP register has a negative number, deceleration starts delayed. A motor stops before reaching FL speed.

Set to 0 if no offset is required.

The setting range is -8,388,608 (800000h) to 8,388,607 (7FFFFFFh). Set the same value in bit 31 to 24 as in bit 23.

For example, in the case of -8,388,608, 1 will be set in bit 23, so you write FF800000h.

< Manual slow-down point setting (RMD.MSDP = 1)>

Set the slow-down point value which will be set manually.

Starts deceleration when the RPLS register value becomes less than or equal to the RSDC register value.

To find the optimum value of the slow-down point manual setting, FL and FH speed values are required.

The optimum value in RDP register is as follows, depending on RMD.MSMD bit and the RDS register value.

1. Linear deceleration (RMD.MSMD = 0)

$$RDP[pulse] = \frac{(RFH^2 - RFL^2) \times (RDR + 1)}{(RMG + 1) \times 32,768}$$

When FH correction function OFF (RMD.MADJ = 1) is set, the optimum value for a triangular drive are as follows.

(Do not want to change the value to be set in RFH register)

$$RDP[pulse] = \frac{RMV \times (RDR + 1)}{RUR + RDR + 2}$$

When using idling control, replace RMV with RMV- (RENV5.IDL-1) for calculation.

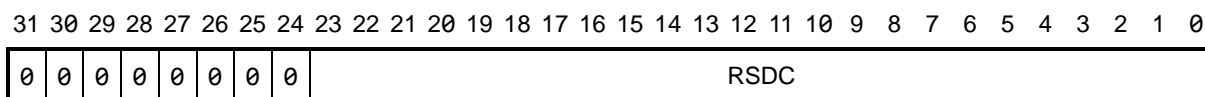
2. S-curve deceleration without linear section (RMD.MSMD = 1 and RDS = 0)

$$RDP[pulse] = \frac{(RFH^2 - RFL^2) \times (RDR + 1) \times 2}{(RMG + 1) \times 32,768}$$

3. S-curve deceleration with linear section (RMD.MSMD = 1 and RDS > 0)

$$RDP[pulse] = \frac{(RFH + RFL) \times (RFH - RFL + 2 \times RDS) \times (RDR + 1)}{(RMG + 1) \times 32,768}$$

5.4.2.5 RSDC: Slow-down point auto calculation value

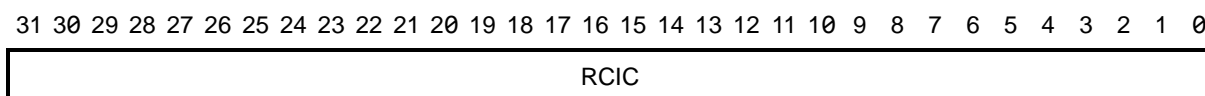


A7 to A0 address of direct access method [68000, H8]: RSDC(24h)
 A7 to A0 address of direct access method [8086, Z80]: RSDC(D8h)
 Register control command of indirect access method: RRSDC(F6h)

This register acquires the automatic calculation of slow-down point when the slow-down point is automatically set (RMD.MSDP = 0). Read-only.

For example, in the case of -8,388,608, bit 23 becomes 1, so 00800000h is read out.

5.4.2.6 RCIC: Circular interpolation step number



A7 to A0 address of direct access method [68000, H8]: RCIC(08h)
 A7 to A0 address of direct access method [8086, Z80]: RCIC(F4h)
 Register control command of indirect access method: RRCI(FDh)

Register to acquire the number of circular interpolation steps in circular interpolation.

Since this register is common to all axes, the value will be the same no matter which axis is read.

Read-only.

Updates the RCIC register value with the RCI register value when starting a circular interpolation.

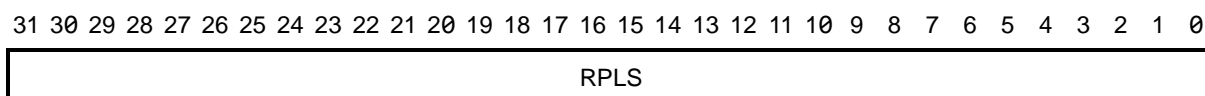
The RCIC register value is down-counted to 0 at each pulse output in circular interpolation.

If the setting number of circular interpolation steps is large, circular interpolation will stop even if RCIC > 0.

If the setting number of circular interpolation steps is small, circular interpolation will continue even after RCIC = 0.

For the number of circular interpolation steps, see "6.3.5 Number of circular interpolation steps".

5.4.2.7 RPLS: Remaining pulse number



A7 to A0 address of direct access method [68000, H8]: RPLS(2Ch)
 A7 to A0 address of direct access method [8086, Z80]: RPLS(D0h)
 Register control command of indirect access method: RRPLS(F4h)

Register to acquire the remaining pulse number in positioning operations.

Read-only.

When the RMV register value is changed, the RPLS register value is recalculated and updated.

When starting positioning control, the RPLS register value is also recalculated and updated.

When starting other than positioning control, the RPLS register value is updated to the RMV register value.

The number of remaining pulses is down-counted for each pulse output.

For positioning controls, RPLS=0 completes the operation mode.

5.4.3 Environment setting register

This register is for setting the environment.

5.4.3.1 RMD(PRMD): Operation mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIPF	MPCS	MSDP	METM	MCCE	MSMD	MINP	MSDE	MENI	MOD						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	MIPM	MPIE	MADJ	MSPO	MSPE	MAX				MSY	MSN		

A7 to A0 address of direct access method [68000, H8]: RMD(BCh), PRMD(F4h)

A7 to A0 address of direct access method [8086, Z80]: RMD(40h), PRMD(08h)

Register control command of indirect access method: RRMD(D0h), RPRMD(C0h), WRMD(90h), WPRMD(80h)

Register to set the operation mode.

PRMD register is the pre-register of RMD register.

Bit	Name	Description
6: 0	MOD	<p>Set the operation mode.</p> <p>000 0000 (00h): Operation mode of continuous movement in +direction by command control. 000 1000 (08h): Operation mode of continuous movement in -direction by command control. 000 0001 (01h): Operation mode of continuous movement in pulser control. 000 0010 (02h): Operation mode of continuous movement in switch control.</p> <p>001 0000 (10h): Operation mode of origin return in +direction in origin return control. 001 1000 (18h): Operation mode of origin return in -direction in origin return control. 001 0010 (12h): Operation mode of origin escape in +direction in origin return control. 001 1010 (1Ah): Operation mode of origin escape in -direction in origin return control. 001 0101 (15h): Operation mode of origin search in +direction in origin return control. 001 1101 (1Dh): Operation mode of origin search in -direction in origin return control.</p> <p>010 0000 (20h): Operation mode of movement to +EL or +SL in sensor control. 010 1000 (28h): Operation mode of movement to -EL or -SL in sensor control. 010 0010 (22h): Operation mode of -EL or -SL escape in sensor control. 010 1010 (2Ah): Operation mode of +EL or +SL escape in sensor control. 010 0100 (24h): Operation mode of movement for EZ count in +direction in sensor control. 010 1100 (2Ch): Operation mode of movement for EZ count in -direction in sensor control.</p> <p>100 0000 (41h): Operation mode of incremental movement in positioning control. 100 1000 (42h): Operation mode in which absolute position is specified by counter 1 in positioning control. 100 0010 (43h): Operation mode in which absolute position is specified by counter 2 in positioning control. 100 1010 (44h): Operation mode for returning to zero point with counter 1 in positioning control. 100 0100 (45h): Operation mode for returning to zero point with counter 2 in positioning control. 100 1100 (46h): Operation mode of one pulse in +direction in positioning control. 100 1110 (4Eh): Operation mode of one pulse in -direction in positioning control. 100 0111 (47h): Timer operation mode in positioning control.</p> <p>101 0001 (51h): Operation mode of incremental movement in pulser control. 101 0010 (52h): Operation mode in which the absolute position is specified with counter 1 in pulser control. 101 0011 (53h): Operation mode in which the absolute position is specified with counter 2 in pulser control. 101 0100 (54h): Operation mode for returning to zero point with counter 1 in pulser control. 101 0101 (55h): Operation mode for returning to zero point with counter 2 in pulser control.</p> <p>101 0110 (56h): Operation mode of incremental movement in switch control.</p> <p>110 0000 (60h): Operation mode of continuous movement in linear interpolation 1 control. 110 0001 (61h): Operation mode of incremental movement in linear interpolation 1 control.</p>

Bit	Name	Description
		<p>110 0010 (62h): Operation mode of continuous movement in linear interpolation 2 control. 110 0011 (63h): Operation mode of incremental movement in linear interpolation 2 control.</p> <p>110 0100 (64h): Operation mode of circular interpolation in the CW direction in circular interpolation control. 110 0101 (65h): Operation mode of circular interpolation in the CCW direction in circular interpolation control.</p> <p>110 0110 (66h): Operation mode of circular interpolation in the CW direction in U-axis synchronization control. 110 0111 (67h): Operation mode of circular interpolation in the CCW direction in U-axis synchronization control.</p> <p>110 1000 (68h): Operation mode of continuous movement with linear interpolation 1 in pulser control. 110 1001 (69h): Operation mode of incremental movement with linear interpolation 1 in pulser control. 110 1010 (6Ah): Operation mode of continuous movement with linear interpolation 2 in pulser control. 110 1011 (6Bh): Operation mode of incremental movement with linear interpolation 2 in pulser control. 110 1100 (6Ch): Operation mode of circular interpolation in the CW direction in pulser control. 110 1101 (6Dh): Operation mode of circular interpolation in the CCW direction in pulser control.</p> <p>Do not set any other values.</p> <p>For the operation mode, see "5.5 Operation mode".</p>
7	MENI	<p>Even if the operation stop interrupt is enabled (RENV2.IEND = 1), the operation stop interrupt (MSTS.SENI) can be disabled when the pre-register is determined (RSTS.PFM = 10b or 11b).</p> <p>0: Also MSTS.SENI = 1 when RSTS.PFM = 10b or 11b 1: It will NOT MSTS.SENI = 1 when RSTS.PFM = 10b or 11b</p> <p>The occurrence of an operation stop interrupt can be suppressed while the next continuous operation remains.</p>
8	MSDE	<p>Sets the input function of +SDn and -SDn pins.</p> <p>0: General-purpose input pin 1: Decelerate or decelerate-stop when SD signal in the operating direction is ON</p> <p>+SD signal input status is acquired from RSTS.PSDI bit, and -SD signal input status is acquired from RSTS.MSDI bit.</p>
9	MINP	<p>Sets the input function of INPn pin.</p> <p>0: General-purpose input pin 1: Operation mode is completed when INP signal is ON</p> <p>The input status of INP signal is acquired in RSTS.SINP bit.</p> <p>See "6.8.1 Positioning complete (INP)" for INP signals.</p>
10	MSMD	<p>Sets the operation of acceleration / deceleration.</p> <p>0: Linear acceleration / deceleration 1: S-curve acceleration / deceleration</p> <p>Please set RMD.MSMD = 1 for linear acceleration, S-curve deceleration, or a combination of S-curve acceleration and linear deceleration.</p> <p>If you set the S-curve section to a small value (RUS = 1, RDS = 1), the characteristics will be almost the same as linear acceleration or linear deceleration.</p>
11	MCCE	<p>Sets the counting function of counter 1.</p> <p>0: Count 1: No count. Pulses can be output while counting of counter 1 is stopped</p>

Bit	Name	Description
12	METM	<p>Sets the completion timing of an operation mode.</p> <p>0: Sets to output pulse cycle complete</p> <p>1: Sets to output pulse ON width complete</p> <p>The operation mode completion timing will be advanced by the OFF width of the final pulse.</p> <p>When using the vibration suppression function, please set the output pulse cycle completed (RMD.METM = 0).</p> <p>Also set RMD.METM = 0 when performing continuous operation by using a pre-register.</p>
13	MSDP	<p>Sets how to set the slow-down point.</p> <p>0: Automatic setting</p> <p>1: Manual setting</p> <p>When using the automatic setting in circular interpolation control, set the number of circular interpolation steps in RCI register.</p>
14	MPCS	<p>Sets the input function of PCSn pin.</p> <p>When RENV1.PCSM = 1, it can also serve as the STA signal input pin to start the own axis only.</p> <p>0: General-purpose input pin</p> <p>1: PCS signal input pin for target position override 2</p> <p>For the target position override 2, see "6.4.2 Target position override 2 (PCS)"</p> <p>The status of an input signal to PCSn pin is acquired in RSTS.SPCS bit.</p>
15	MIPF	<p>Sets the constant synthesized speed control in an interpolation operation.</p> <p>0: Synthesized speed is NOT controlled to be constant</p> <p>1: Synthesized speed is controlled to be constant</p> <p>For constant synthesized speed constant control, see "6.3.6 Constant synthesized speed control"</p>
17,16	MSN	<p>Sets the 2-bit sequence number.</p> <p>The sequence number is obtained in MST5.SSC bit</p> <p>The sequence number does not affect the operation</p> <p>It can be used to control operation blocks when creating the control software.</p> <p>For the control of operation blocks, see "6.2.1 Contiguous operation ".</p>
19,18	MSY	<p>Sets the start timing when writing a start command.</p> <p>00b: Starts immediately.</p> <p>01b: Starts with CSTA signal ON, STA signal ON, or SPSTA (2Ah) command. If RENV1.PCSM = 0, starts with CSTA signal ON or SPSTA (2Ah) command. If RENV1.PCSM = 1, starts with STA signal ON or SPSTA (2Ah) command.</p> <p>10b: Starts with an internal synchronization signal (RENV5.SYI).</p> <p>11b: Starts when the specified axis (RMD.MAX) is stopped.</p>
23: 20	MAX	<p>Sets the axis to confirm to be stopped when RMD.MSY = 11b.</p> <p>Example: 0001b: Starts at X-axis 's stop 0010b: Starts at Y-axis 's stop 0100b: Starts at Z-axis's stop 1000b: Starts at U-axis 's stop 0101b: Starts at both X-axis and Z-axis' stop 1111b: Starts at all axes' stop.</p> <p>If you want to include the own axis to be stopped in the condition, also set RENV2.SMAX = 1.</p>

Bit	Name	Description
24	MSPE	<p>Sets the input function of CSTP pin.</p> <p>0: General-purpose input pin 1: Decelerates or stops immediately by inputting CSTP signal</p> <p>The input status of CSTP signal is acquired in RSTS.SSTP bit.</p>
25	MSPO	<p>Sets the output function of CSTP pin.</p> <p>0: General-purpose output pin. You can output a negative logic one-shot pulse with CMSTP (07h) command.</p> <p>1: Output a negative logic one-shot pulse when own axis stops abnormally.</p>
26	MADJ	<p>Sets how to correct the FH speed.</p> <p>0: Automatic correction (automatically avoid a triangular drive) 1: Manual correction (Not automatically avoid a triangular drive).</p> <p>When combining the following settings, please set the manual correction (RMD.MADJ = 1).</p> <ul style="list-style-type: none"> • Incremental movement by linear interpolation 1 (RMD.MOD = 61h) • S-curve acceleration / deceleration (RMD.MSMD = 1) • Constant synthesized speed control (RMD.MIPF = 1) <p>Also set the manual correction (RMD.MADJ = 1) to obtain the operating time accurately.</p> <p>Obtain the FH speed in manual correction in "6.3.3 Manual correction calculation of FH speed".</p>
27	MPIE	<p>Sets the end-point-draw function.</p> <p>0: Stops on the arc without the point-draw operation in circular interpolation control. 1: Moves to the end point with the end-point-draw operation in circular interpolation control.</p> <p>For the end-point-draw operation, see "6.4.3 End point draw operation".</p>
28	MIPM	<p>Sets the completion condition of circular interpolation.</p> <p>0: Determines the end-point coordinates of a completion condition in 90-degree increments. Operates so as to draw an arc shorter if the start and end points are in the same quadrant.</p> <p>1: Determines the end-point coordinates of a completion condition in 45-degree increments. Operates so as to draw an arc longer if the start and end points are in the same quadrant.</p> <p>For the completion conditions for circular interpolation, see "5.5.9.1 Circular interpolation in CW direction (64h)".</p>
31: 29	0	Always set it to 0.

5.4.3.2 RENV1: Environment setting 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERCL	EPW			EROR	EROE	ALML	ALMM	ORGL	SDL	SDLT	SDM	ELM	PMD		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PDTC	PCSM	INTM	DTMF	DRF	FLTR	DRL	PCSL	LTCL	INPL	CLRM	CLRL	STPM	STAM	ETW	

A7 to A0 address in direct access method [68000, H8]: RENV1(8Ch)

A7 to A0 address in direct access method [8086, Z80]: RENV1(70h)

Register control command in indirect access method: RRENV1(DCh), WRENV1(9Ch)

A register that sets the specifications of input/output pins.

Bit	Name	Description																																																	
2: 0	PMD	<p>Sets the output pulse mode.</p> <table border="1"> <thead> <tr> <th rowspan="2">PMD</th><th colspan="2">+Direction</th><th colspan="2">-Direction</th></tr> <tr> <th>OUT (PLS)</th><th>DIR (MNS)</th><th>OUT (PLS)</th><th>DIR (MNS)</th></tr> </thead> <tbody> <tr> <td>000</td><td></td><td>High</td><td></td><td>Low</td></tr> <tr> <td>001</td><td></td><td>High</td><td></td><td>Low</td></tr> <tr> <td>010</td><td></td><td>Low</td><td></td><td>High</td></tr> <tr> <td>011</td><td></td><td>Low</td><td></td><td>High</td></tr> <tr> <td>100</td><td></td><td>High</td><td>High</td><td></td></tr> <tr> <td>101</td><td> </td><td></td><td> </td><td></td></tr> <tr> <td>110</td><td> </td><td></td><td> </td><td></td></tr> <tr> <td>111</td><td></td><td>Low</td><td>Low</td><td></td></tr> </tbody> </table> <p>* Both edges are enabled for 90-degree phase difference signals (PHA, PHB) of 101b and 110b. 90-degree phase difference signal outputs four pulses in one cycle.</p>	PMD	+Direction		-Direction		OUT (PLS)	DIR (MNS)	OUT (PLS)	DIR (MNS)	000		High		Low	001		High		Low	010		Low		High	011		Low		High	100		High	High		101					110					111		Low	Low	
PMD	+Direction			-Direction																																															
	OUT (PLS)	DIR (MNS)	OUT (PLS)	DIR (MNS)																																															
000		High		Low																																															
001		High		Low																																															
010		Low		High																																															
011		Low		High																																															
100		High	High																																																
101																																																			
110																																																			
111		Low	Low																																																
3	ELM	<p>Sets the input processing of +EL signal and -EL signal.</p> <p>0: To stop immediately when EL signal in the operating direction is ON</p> <p>1: To decelerate-stop when EL signal in the operating direction is ON</p> <p>When the deceleration stop is set, input an EL signal in the operating direction to start the deceleration.</p> <p>Be careful not to collide as the motor passes through the EL position and stops.</p>																																																	

Bit	Name	Description
4	SDM	Sets the input processing of +SD and –SD signals. 0: Decelerates when SD signal in the operating direction is ON 1: Decelerate-stops when SD signal in the operating direction is ON
5	SDLT	Sets the latch function of +SD signal and –SD signal inputs. Used when the signal widths of +SD signal and –SD signals are short. 0: No latch the SD signal in the operating direction. Input status of +SD signal is acquired in RSTS.PSDI bit Input status of –SD signal is acquired in RSTS.MSDI bit 1: Latch the SD signal in the operating direction. Latch status of +SD signal is acquired in RSTS.PSDL bit Latch status of –SD signal is acquired in RSTS.MSDL bit If the SD signal in the operating direction is OFF, the latch status of a SD signal in the operating direction will be OFF when starting. When RENV1.SDLT = 0, the latch status of +SD signal and –SD signal will turn OFF. Latch status of the SD signal in the operating direction can be obtained in SSTS.SSD bit.
6	SDL	Sets the input logic of +SD and –SD signals. 0: Negative logic. 1: Positive logic.
7	ORGL	Sets the input logic of ORG signal. 0: Negative logic 1: Positive logic
8	ALMM	Sets the input processing of ALM signal. 0: ALM signal is ON to stop immediately. 1: ALM signal is ON to decelerate-stop.
9	ALML	Sets the input logic of ALM signal. 0: Negative logic 1: Positive logic
10	EROE	Sets the output function of ERCn pin at the immediate stop due to an abnormal stop factor. ERC signal can be output at the time of immediate stop by turning +EL, –EL, ALM, and CEMG signal ON. ERC signal can be output even when CEMG (05h) command is used to stop immediately. 0: No ERC signal is output at the time of immediate stop due to an abnormal stop factor. 1: ERC signal is output at the time of immediate stop due to an abnormal stop factor. Even when RMD.MOD = 20h and 28h, ERC signal is output at the immediate stop by turning +EL and –EL signal ONs.
11	EROR	Sets the output function of ERCn pin at the stop due to an origin return factor. 0: No ERC signal is output when stopped due to an origin return factor. 1: ERC signal is output when stopped due to an origin return factor. For ERC signals, see "6.8.2 Deviation counter clear (ERC)".

Bit	Name	Description
14:12	EPW	<p>Sets the ON-width of ERC signal.</p> <p>000b: 11 to 13 μs 001b: 91 to 98 μs 010b: 364 to 391 μs 011b: 1.4 to 1.6 ms 100b: 11 to 13 ms 101b: 46 to 50 ms 110b: 93 to 100 ms 111b: Level output</p> <p>Turn the bit OFF with ERCRST (25h) command if RENV1.EPW = 111b is set,</p>
15	ERCL	<p>Sets the output logic of ERC signal.</p> <p>0: Negative logic 1: Positive logic</p>
17,16	ETW	<p>Sets the OFF-width of ERC signal.</p> <p>00b: 0 μs 01b: 11 to 13 μs 10b: 1.4 to 1.6 ms 011b: 93 to 100 ms</p>
18	STAM	<p>Sets the input specifications of CSTA signal.</p> <p>0: Level trigger 1: Edge trigger (falling edge)</p>
19	STPM	<p>Sets the input processing of CSTP signal.</p> <p>0: Stop immediately 1: Decelerate-stop</p>
20	CLRL	<p>Sets the input logic for CLR signal.</p> <p>0: Negative logic 1: Positive logic</p>
21	CLRM	<p>Sets the input specifications of CLR signal.</p> <p>0: Edge trigger (OFF to ON) 1: Level trigger</p>
22	INPL	<p>Sets the input logic of INP signal.</p> <p>0: Negative logic 1: Positive logic</p>
23	LTCL	<p>Sets the input logic of LTC signal.</p> <p>0: Negative logic 1: Positive logic</p> <p>Latches the counter count value when LTC signal changes from OFF to ON in RENV5.LTM = 00b,</p>
24	PCSL	<p>Sets the input logic of PCS and STA signals.</p> <p>0: Negative logic 1: Positive logic</p>
25	DRL	<p>Sets the input logic of +DR and -DR signals.</p> <p>0: Negative logic 1: Positive logic</p>
26	FLTR	<p>Sets the input noise filters of +EL, -EL, +SD, -SD, ORG, ALM, INP and CEMG signals.</p> <p>0: Recognizes the pulse signals of width of 0.05 μs or more. 1: Ignores the pulse signals of width of 3 μs or less completely.</p>

Bit	Name	Description																														
27	DRF	Sets the input noise filters of +DR, -DR, PE signals. 0: Recognizes the pulse signals of width of 0.05 μs or more. 1: Ignores the pulse signals of width of 26 ms or less.																														
28	DTMF	Sets the direction-change timer time. 0: If RENV1.PMD = 000b to 011b, wait 0.2 ms for pulse output after changing direction. 1: If RENV1.PMD = 000b to 011b, wait 0.5 μs for pulse output after changing direction.																														
29	INTM	Sets the output function of INT pin. 0: If an interrupt factor occurs, L level is output from INT pin. 1: Even if an interrupt factor occurs, H level is output from the INT pin.																														
30	PCSM	Sets the input function of CSTA pin and PCSn pin. 0: CSTA pin input is enabled. The setting of RMD.MPCS bit is reflected in PCSn pin. 1: CSTA pin input is disabled. PCSn pin also inputs the STA signal to start only for its own axis.																														
		<table border="1"> <thead> <tr> <th></th> <th>RENV1.PCSM=0 RMD.MPCS=0</th> <th>RENV1.PCSM=0 RMD.MPCS=1</th> <th>RENV1.PCSM=1 RMD.MPCS=0</th> <th>RENV1.PCSM=1 RMD.MPCS=1</th> </tr> </thead> <tbody> <tr> <td>RSTS.SSTA (CSTA=L)</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>RSTS.SSTA (STA: ON)</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>RSTS.SPCS (CSTA=L)</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>RSTS.SPCS (PCS: ON)</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>Override (PCS: ON)</td> <td>Disabled</td> <td>Enabled</td> <td>Disabled</td> <td>Enabled</td> </tr> </tbody> </table>		RENV1.PCSM=0 RMD.MPCS=0	RENV1.PCSM=0 RMD.MPCS=1	RENV1.PCSM=1 RMD.MPCS=0	RENV1.PCSM=1 RMD.MPCS=1	RSTS.SSTA (CSTA=L)	1	1	0	0	RSTS.SSTA (STA: ON)	0	0	1	1	RSTS.SPCS (CSTA=L)	0	0	0	0	RSTS.SPCS (PCS: ON)	1	1	1	1	Override (PCS: ON)	Disabled	Enabled	Disabled	Enabled
			RENV1.PCSM=0 RMD.MPCS=0	RENV1.PCSM=0 RMD.MPCS=1	RENV1.PCSM=1 RMD.MPCS=0	RENV1.PCSM=1 RMD.MPCS=1																										
		RSTS.SSTA (CSTA=L)	1	1	0	0																										
		RSTS.SSTA (STA: ON)	0	0	1	1																										
		RSTS.SPCS (CSTA=L)	0	0	0	0																										
RSTS.SPCS (PCS: ON)	1	1	1	1																												
Override (PCS: ON)	Disabled	Enabled	Disabled	Enabled																												
31	PDTC	Sets the output pulse width control function. 0: When the output speed of a command pulse is 2.4 kpps or less, the output pulse width is fixed at 0.2 ms. 1: The output pulse width can fluctuate at a duty ratio of 50% regardless of the output speed of a command pulse.																														

5.4.3.3 RENV2: Environment setting 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P7M		P6M		P5M		P4M		P3M		P2M		P1M		P0M	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POFF	EOFF	SMAX	PMSK	IEND	PDIR	PIM		EZL	EDIR	EIM		PINF	EINF	P1L	P0L

A7 to A0 address in direct access method[68000, H8]: RENV2(88h)

A7 to A0 address in direct access method[8086, Z80]: RENV2(74h)

Register control command of indirect access method: RRENV2(DDh), WRENV2(9Dh)

A register that sets the specifications of general-purpose input/output pins, EA, EB signals, and PA / PB signals.

Bit	Name	Description
1,0	P0M	Sets the input/output function of P0n pin. 00b: General-purpose input pin 01b: General-purpose output pin 10b: Outputs FUP signal in negative logic in acceleration 11b: Outputs a general-purpose one-shot signal
3,2	P1M	Sets the input/output function of P1n pin. 00b: General-purpose input pin 01b: General-purpose output pin 10b: FDW signal in negative logic during deceleration 11b: Outputs a general-purpose one-shot signal.
5,4	P2M	Sets the input/output function of P2n pin. 00b: General-purpose input pin 01b: General-purpose output pin 10b: Outputs MVC signal in negative logic during constant speed operation. 11b: Outputs MVC signal in positive logic during constant speed operation.
7,6	P3M	Sets the input/output function of P3n pin. 00b: General-purpose input pin 01b: General-purpose output pin 10b: Outputs CP1 signal in negative logic when the comparator 1 condition is satisfied. 11b: Output CP1 signal in positive logic when the comparator 1 condition is satisfied.
9,8	P4M	Sets the input/output function of P4n pin. 00b: General-purpose input pin 01b: General-purpose output pin 10b: Outputs CP2 signal in negative logic when the comparator 2 condition is satisfied. 11b: Outputs CP2 signal in positive logic when the comparator 2 condition is satisfied.

Bit	Name	Description
11,10	P5M	Sets the input/output function of P5n pin. 00b: General-purpose input pin 01b: General-purpose output pin 10b: Outputs CP3 signal in negative logic when the comparator 3 condition is satisfied. 11b: Output CP3 signal in positive logic when the comparator 3 condition is satisfied.
13,12	P6M	Sets the input/output function of P6n pin. 00b: General-purpose input pin 01b: General-purpose output pin 10b: Outputs CP4 signal in negative logic when the comparator 4 condition is satisfied. 11b: Outputs CP4 signal in positive logic when the comparator 4 condition is satisfied.
15,14	P7M	Sets the input/output function of P7n pin. 00b: General-purpose input pin 01b: General-purpose output pin 10b: Outputs CP5 signal in negative logic when the comparator 5 condition is satisfied. 11b: Outputs CP5 signal in positive logic when the comparator 5 condition is satisfied.
16	P0L	Sets the output logic of FUP signal and general-purpose one-shot signal that can be output from P0n pin. 0: Negative logic 1: Positive logic
17	P1L	Sets the output logic of FDW signal and general-purpose one-shot signal that can be output from P1n pin. 0: Negative logic 1: Positive logic
18	EINF	Sets the input noise filter for EA, EB, and EZ signals. 0: Recognize signals with a pulse width of 0.05 μ s or more. 1: Recognize signals with a pulse width of 0.15 μ s or more.
19	PINF	Sets the input noise filter for PA and PB signals. 0: Recognize signals with a pulse width of 0.05 μ s or more. 1: Recognize signals with a pulse width of 0.15 μ s or more.
21,20	EIM	Sets input specifications for EA and EB signals. 00b: 90-degree phase difference mode 1x 01b: 90-degree phase difference mode 2x 10b: 90-degree phase difference mode 4x 11b: 2-pulse mode. See "6.12.1 Counter type and input specifications" for details,
22	EDIR	Sets the counting direction of EA and EB signals. 0: Count up when the phase of EA signal is advanced. 1: Count up when the phase of EB signal is advanced.

Bit	Name	Description
23	EZL	Sets the input logic of EZ signal. 0: Negative logic 1: Positive logic Counts when EZ signal changes from OFF to ON.
25,24	PIM	Sets the input specifications for PA and PB signals. 00b: 90-degree phase difference mode 1x 01b: 90-degree phase difference mode 2x 10b: 90-degree phase difference mode 4x 11b: 2-pulse mode. See "5.5.3 Pulser control" for details.
26	PDIR	Sets the counting direction of PA and PB signals. 0: Count up when the phase of PA signal is advanced. 1: Count up when the phase of PB signal is advanced.
27	IEND	Sets the functional specifications of MSTS.SENI bit. 0: Disabled. Keeps MSTS.SENI = 0 at operation stop. 1: Enabled. Changes to MSTS.SENI = 1 at operation stop.
28	PMSK	Sets the output function of a command pulse. 0: Valid. Outputs a command pulse. 1: Invalid. Not output a command pulse. In either case, the counter will work.
29	SMAX	Sets the functional specifications when RMD.MAX bit includes the own axis with RMD.MSY = 11b. 0: Not start when the own axis stops at the end in the case the own axis is included in RMD.MAX bit. 1: Starts even when the own axis stops at the end in the case the own axis is included in the RMD.MAX bit.
30	EOFF	Sets the input function of EA and EB signals. 0: Enabled 1: Disabled. Also not detect an input error.
31	POFF	Sets the input function of PA and PB signals. 0: Enabled 1: Disabled. Also not detect an input error.

5.4.3.4 RENV3: Environment setting 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BSYC	C14	C13	C12	EZD			ORM							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CU4H	CU3H	CU2H	0	CU4B	CU3B	CU2B	CU1B	CU4R	CU3R	CU2R	CU1R	CU4C	CU3C	CU2C	CU1C

A7 to A0 address of direct access method[68000, H8]: RENV3(84h)

A7 to A0 address of direct access method[8086, Z80]: RENV3(78h)

Register control command of indirect access method: RENV3(DEh), WRENV3(9Eh)

Register to set the specifications of an origin return operation and the function of a counter.

Bit	Name	Description
3:0	ORM	<p>Selects the origin return method.</p> <p>0000b: Origin return 0</p> <ul style="list-style-type: none"> When ORG signal changes from OFF to ON, a motor stops immediately in FL or FH constant speed operation. Decelerate-stops in high-speed 1 or 2 operation. Counter clear timing: When ORG signal changes from OFF to ON. <p>0001b: Origin return 1</p> <ul style="list-style-type: none"> When ORG signal changes from OFF to ON, a motor stops immediately in FL or FH constant speeds operation. Decelerate-stops in high-speed 1 or 2 operation. Operate in the opposite direction at FA constant speed until ORG signal changes from ON to OFF. Operates in the initial direction at FA speed and stop immediately when ORG signal changes from OFF to ON. Counter clear timing: When ORG signal changes from OFF to ON after moving at FA speed. <p>0010b: Origin return 2</p> <ul style="list-style-type: none"> After operating from ORG signal OFF to ON, stops immediately at the time of EZ count-up in FL and FH constant speed operations. In the high speed 1 or 2 operation, decelerates when changing ORG signal OFF to ON and stops at the time of EZ count-up. Counter clear timing: When the specified number of EZ is counted up. <p>0011b: Origin return 3</p> <ul style="list-style-type: none"> After operating from ORG signal OFF to ON, stop immediately in FL and FH constant speed, and decelerate-stops in high-speed 1 and 2 at the time of EZ count-up. Counter clear timing: When the specified number of EZ is counted up. <p>0100b: Origin return 4</p> <ul style="list-style-type: none"> When ORG signal changes from OFF to ON, a motor stops immediately in FL or FH constant speed operations. Decelerate-stops in high-speed 1 or 2 operation. Operate in the opposite direction at FA constant speed until ORG signal changes from ON to OFF. Stops immediately when the specified number of EZ is counted up. Counter clear timing: When the specified number of EZ is counted up. <p>0101b: Origin return 5</p> <ul style="list-style-type: none"> When ORG signal changes from OFF to ON, a motor stops immediately in FL or FH constant speed operation. Decelerate-stops in high-speed 1 or 2 operation. After operating from ORG signal ON to OFF in the opposite direction, stops immediately in FL or FH constant speed operation, and decelerate-stops at high speed 1 or 2 operation. Counter clear timing: When the specified number of EZ is counted up. <p>0110b: Origin return 6</p> <ul style="list-style-type: none"> When EL signal changes from OFF to ON, a motor stops immediately, and decelerate-stop when RENV1.ELM = 1 in high speed 1 or 2 operation. Operate in the opposite direction at FA constant speed until EL signal changes from ON to OFF, then stops immediately. Counter clear timing: When EL signal changes from ON to OFF. <p>0111b: Origin return 7</p> <ul style="list-style-type: none"> When EL signal changes from OFF and ON, a motor stops immediately. Decelerate-stops when RENV1.ELM = 1 in high speed 1 or 2 operation. After operating from EL signal ON to OFF in the opposite direction at FA constant speed, stop immediately when the specified number of EZ is counted up. Counter clear timing: At immediate stop when the specified number of EZ is counted up. <p>1000b: Origin return 8</p> <ul style="list-style-type: none"> When EL signal changes from OFF and ON, a motor stops immediately. Decelerate-stops when RENV1.ELM = 1 in high speed 1 or 2 operation. Counter clear timing: When the specified number of EZ is counted up.

Bit	Name	Description
		<p>1001b: Origin return 9 – Return to zero point (operate until RCUN2 = 0) after the origin return operation 0.</p> <p>1010b: Origin return 10 – Return to zero point (operate until RCUN2 = 0) after the origin return operation 3.</p> <p>1011b: Origin return 11 – Return to zero point (operate until RCUN2 = 0) after the origin return operation 5.</p> <p>1100b: Origin return 12 – Return to zero point (operate until RCUN2 = 0) after the origin return operation 8.</p>
7:4	EZD	<p>Sets the initial input count value of EZ signal used for an origin return control and a sensor control. The setting range is 0000b (1 time) to 1111b (16 times).</p>
9,8	CI2	<p>Sets the count target for counter 2. 00b: EA, EB signal 01b: Command pulse signal 10b: PA, PB signal 11b: Prohibited</p>
11,10	CI3	<p>Sets the count target for counter 3. 00b: Deviation count between command pulse signal and EA, EB signals 01b: Deviation count between command pulse signal and PA, PB signals 10b: Deviation count of EA, EB signals and PA, PB signals 11b: Prohibited</p>
13,12	CI4	<p>Sets the count target of counter 4. 00b: Command pulse signal 01b: EA, EB signal 10b: PA, PB signal 11b: $\frac{f_{CLK}}{2}$ signal</p>
14	BSYC	<p>Sets the count limit for counter 4. 0: No limit 1: Count only when BSY = L level</p>
15	0	Always set to 0.
16	CU1C	<p>Sets whether or not to clear counter 1 when CLR signal is ON. 0: Not clear 1: Clear.</p>
17	CU2C	<p>Sets whether or not to clear counter 2 when the CLR signal is ON. 0: Not clear 1: Clear.</p>
18	CU3C	<p>Sets whether or not to clear counter 3 when the CLR signal is ON. 0: Not clear 1: Clear</p>
19	CU4C	<p>Sets whether or not to clear counter 4 when the CLR signal is ON. 0: Not clear 1: Clear</p>
20	CU1R	<p>Sets whether or not to clear counter 1 when the origin is reached in the origin return control. 0: Not clear 1: Clear</p>

Bit	Name	Description
21	CU2R	Sets whether or not to clear counter 2 when the origin is reached by the origin return control. 0: Not clear 1: Clear
22	CU3R	Sets whether or not to clear counter 3 when the origin is reached by the origin return control. 0: Not clear 1: Clear
23	CU4R	Sets whether or not to clear counter 4 when the origin is reached by the origin return control. 0: Not clear 1: Clear
24	CU1B	Sets whether or not counter 1 counts during backlash correction and slip correction. 0: Not count 1: Count
25	CU2B	Sets whether or not counter 2 counts during backlash correction and slip correction. 0: Not count 1: Count
26	CU3B	Sets whether or not counter 3 counts during backlash correction and slip correction. 0: Not count 1: Count
27	CU4B	Sets whether or not counter 4 counts during backlash correction and slip correction. 0: Not count 1: Count
28	0	Always set to 0. * 1
29	CU2H	Sets whether or not counter 2 count. 0: Count 1: Not count
30	CU3H	Sets whether or not counter 3 count. 0: Count 1: Not count
31	CU4H	Sets whether or not counter 4 count. 0: Count 1: Not count

* 1 The count function of counter 1 is set by RMD.MCCE bit.

5.4.3.5 RENV4: Environment setting 4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2RM	C2D	C2S	C2C	C1RM	C1D	C1S	C1C								
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C4D	C4S	C4C	IDXM	C3D	C3S	C3C									

A7 to A0 address of direct access method[68000, H8]: RENV4(80h)
A7 to A0 address of direct access method[8086, Z80]: RENV4(7Ch)
Register control command of indirect access method: RRENV4(DFh), WRENV4(9Fh)

Register to set the functions of Comparator 1 to Comparator 4.

Bit	Name	Description
1,0	C1C	Sets the comparison target of Comparator 1. 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 If RENV4.C1C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767).
4:2	C1S	Sets the comparison conditions for Comparator 1. 001b: RCMP1 = Comparison target (regardless of counting direction) 010b: RCMP1 = Comparison target (only during count-up) 011b: RCMP1 = Comparison target (only during count-down) 100b: RCMP1 > Comparison target 101b: RCMP1 < Comparison target 110b: Plus side software-limit (RCMP1 < RCUN1) Also set RENV4.C1C = 00b. Others: Comparison conditions are always un-satisfied.
6,5	C1D	Sets the processing when the condition of Comparator 1 is satisfied. 00b: No processing. It can be used to output INT signals or CP1 signals, and to perform an internal synchronization start. 01b: Stops immediately. 10b: Decelerate-stops. 11b: Overrides at a time. If RENV4.C1S = 110b is set, stop immediately when RENV4.C1D = 00b or 11b is set.
7	C1RM	Counter 1 can be ring-counted up to the P1 register value. 0: Not perform ring count. 1: Performs ring count.
9,8	C2C	Sets the comparison target of Comparator 2. 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 If RENV4.C2C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767).
12:10	C2S	Sets the comparison conditions for Comparator 2. 001b: RCMP2 = Comparison target (regardless of counting direction). 010b: RCMP2 = Comparison target (only during count-up). 011b: RCMP2 = Comparison target (only during count-down).

Bit	Name	Description
		<p>100b: RCMP2 > Comparison target.</p> <p>101b: RCMP2 < Comparison target.</p> <p>110b: Minus side software limit (RCMP2 > RCUN1).</p> <p>Also set RENV4.C2C = 00b.</p> <p>Others: Comparison conditions are always un-satisfied.</p>
14,13	C2D	<p>Sets the processing when the condition of Comparator 2 is satisfied.</p> <p>00b: No processing. It can be used to output INT signals and CP2 signals, and to perform an internal synchronization start.</p> <p>01b: Stops immediately.</p> <p>10b: Decelerate-stops.</p> <p>11b: Overrides at a time.</p> <p>If RENV4.C2S = 110b is set, stop immediately when RENV4.C2D = 00b or 11b is set.</p>
15	C2RM	<p>Counter 2 can be ring-counted up to the RCMP2 register value.</p> <p>0: Not perform ring count.</p> <p>1: Perform ring count.</p>
17,16	C3C	<p>Sets the comparison target of Comparator 3.</p> <p>00b:RCUN1 01b: RCUN2 10b:RCUN3 11b: RCUN4</p> <p>If RENV4.C3C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767).</p>
20: 18	C3S	<p>Sets the comparison conditions for Comparator 3.</p> <p>001b: RCMP3 = comparison target (regardless of counting direction).</p> <p>010b: RCMP3 = Comparison target (only during count-up).</p> <p>011b: RCMP3 = Comparison target (only during count-down).</p> <p>100b: RCMP3 > Comparison target.</p> <p>101b: RCMP3 < Comparison target.</p> <p>110b: Prohibited.</p> <p>Others: Comparison conditions are always un-satisfied.</p>
22,21	C3D	<p>Sets the processing when the condition of Comparator 3 is satisfied.</p> <p>00b: No processing. It can be used to output INT signals and CP3 signals, and to perform an internal synchronization start.</p> <p>01b: Stops immediately.</p> <p>10b: Decelerate-stops.</p> <p>11b: Overrides at a time.</p>
23	IDXM	<p>Sets the output conditions of IDX signal.</p> <p>0: Level output with the logic set in RENV2.P6M bit.</p> <p>When RCUN4 = RCMP4 is established, IDX signal is output at the level.</p> <p>1: A pulse is output in the logic set in RENV2.P6M bit.</p> <p>When changing to RCUN4 = 0, IDX signal in a CLK signal 2-cycle width is pulse-output.</p>
25,24	C4C	<p>Sets the comparison target of Comparator 4.</p> <p>00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4</p> <p>If RENV4.C4C = 10b, compare with the absolute value in RCUN3 register (0 to 32,767).</p>

Bit	Name	Description
29: 26	C4S	<p>Sets the comparison conditions for Comparator 4.</p> <p>0001b: RCMP4 = Comparison target (regardless of counting direction)</p> <p>0010b: RCMP4 = Comparison target (only during count-up)</p> <p>0011b: RCMP4 = Comparison target (only during count-down)</p> <p>0100b: RCMP4 > Comparison target</p> <p>0101b: RCMP4 < Comparison target</p> <p>0111b: Comparison condition is not always satisfied.</p> <p>1000b: IDX signal is output under the comparison condition of RENV4.IDXM bits (regardless of the counting direction).</p> <p>1001b: IDX signal is output under the comparison condition of RENV4.IDXM bits (only during count-up).</p> <p>1010b: IDX signal is output under the comparison condition of RENV4.IDXM bits (only during count-down).</p> <p>Others: Comparison conditions are always un-satisfied.</p> <p>When using RENV4.C4S = 1000b, 1001b, 1010b, also set RENV4.C4C = 11b.</p> <p>In this case, when using RENV4.IDXM = 1, set a positive value in RCMP4 register.</p>
31: 30	C4D	<p>Sets the processing when the condition of Comparator 4 is satisfied.</p> <p>00b: No processing. It can be used to output INT signals and CP4 signals, and to perform an internal synchronization start.</p> <p>01b: Stops immediately.</p> <p>10b: Decelerate-stops.</p> <p>11b: Overrides at a time.</p>

5.4.3.6 RENV5: Environment setting 5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LTOF	LTFD	LTM	0	IDL	C5D	C5S	C5C								
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	CU4L	CU3L	CU2L	CU1L	ISMR	MSMR	SYI	SYO				

A7 to A0 address of direct access method[68000, H8]: RENV5(7Ch)

A7 to A0 address of direct access method[8086, Z80]: RENV5(80h)

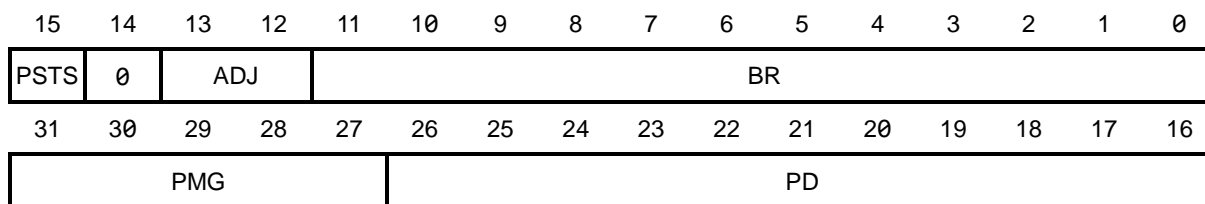
Register control command of indirect access method: RRENV5(E0h), WRENV5(A0h)

Register to set the function of Comparator 5.

Bit	Name	Description
2: 0	C5C	Sets the comparison target of Comparator 5. 000b: RCUN1 001b: RCUN2 010b: RCUN3 011b: RCUN4 100b: RPLS (number of remaining pulses) 101b: RSPD.AS (current speed) When RENV5.C5C = 10b, compare with the absolute value in RCUN3 register (0 to 32,767).
5: 3	C5S	Sets the comparison conditions of Comparator 5. 001b: RCMP5 = Comparison target (regardless of counting direction) 010b: RCMP5 = Comparison target (only during count-up) 011b: RCMP5 = Comparison target (only during count-down) 100b: RCMP5 > Comparison target 101b: RCMP5 < Comparison target Others: Comparison conditions are always un-satisfied.
7,6	C5D	Sets the processing when the condition of Comparator 5 is satisfied. 00b: No processing. It can be used to output INT signals and CP5 signals, and to perform an internal synchronization start. 01b: Stops immediately. 10b: Decelerate-stops. 11b: Overrides at a time.
10:8	IDL	Sets the number of idling pulse outputs. 000b: No idling pulse is output. 001b to 111b: zero to six pulses are output.
11	0	Always set to 0.
13,12	LTM	Sets the timing to latch the values in RCUN1 to 4 registers. 00b: When the LTC signal is changed from OFF to ON. 01b: When the ORG signal is changed from OFF to ON. 10b: When the condition of comparator 4 is satisfied. 11b: When the condition of comparator 5 is satisfied.
14	LTFD	Sets whether to latch the current speed, instead of counter 3. 0: Latches RCUN3 register (count value of counter 3). 1: Latches RSPD.AS bit (current speed step number).

Bit	Name	Description
15	LTOF	Sets to latch only at the write timing of LTCH (29h) command. 0: Latches also at the timing selected by the RENV5.LTM bit. 1: Latches only at the write timing of LTCH (29h) command.
19:16	SYO	Sets the output timing of the internal synchronization signal. 0001b: When Comparator 1 condition is satisfied. 0010b: When Comparator 2 condition is satisfied. 0011b: When Comparator 3 condition is satisfied. 0100b: When Comparator 4 condition is satisfied. 0101b: When Comparator 5 condition is satisfied. 1000b: At the start of acceleration. 1001b: At the end of acceleration. 1010b: At the start of deceleration. 1011b: At the end of deceleration. Other: No output the internal synchronization signal.
21,20	SYI	Sets the input target of the internal synchronization signal. 00b: X-axis internal synchronization signal 01b: Y-axis internal synchronization signal 10b: Z-axis internal synchronization signal 11b: U-axis internal synchronization signal
22	MSMR	Sets how to clear MST.SENI and MST.SEOR bits. 0: Cleared automatically when the main status is read. 1: Not cleared automatically when the main status is read. MST.SENI bit can be cleared manually by writing SENIR (2Dh) command. MST.SEOR bit can be cleared manually by writing SEORR (2Eh) command.
23	ISMR	Sets how to clear the bits in RIST and REST registers. 0: Write the read command for each register to clear each register to 0. Even with the full-address direct access method, each register can be cleared to 0 by writing a read command for each register. 1: Writing the read command for each register does not clear each register to 0. In either case, you can write 1 to the corresponding bit in each register to clear to 0.
24	CU1L	Sets the function to clear counter 1 to 0 immediately after latching counter 1. 0: Not clear counter 1 to 0. 1: Clears counter 1 to 0.
25	CU2L	Sets the function to clear counter 2 to 0 immediately after latching counter 2. 0: Not clear counter 2 to 0. 1: Clears counter 2 to 0.
26	CU3L	Sets the function to clear counter 3 to 0 immediately after latching counter 3. 0: Not clear counter 3 to 0. 1: Clears counter 3 to 0.
27	CU4L	Sets the function to clear counter 4 to 0 immediately after latching counter 4. 0: Not clear counter 4 to 0. 1: Clears counter 4 to 0.
31: 28	0	Always set to 0.

5.4.3.7 RENV6: Environment setting 6



A7 to A0 address of direct access method[68000, H8]: RENV6(78h)

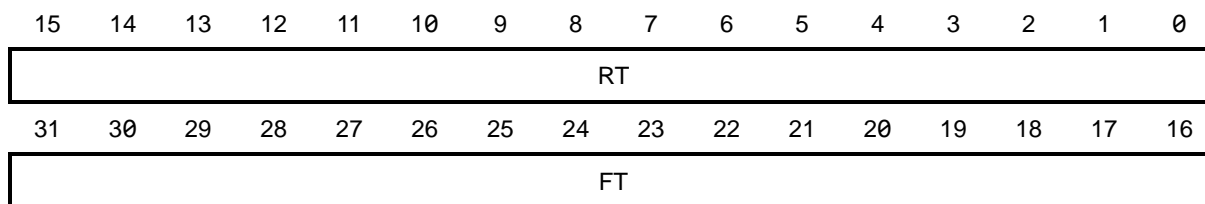
A7 to A0 address of direct access method[8086, Z80]: RENV6(84h)

Register control command of indirect access method: RRENV6(E1h), WRENV6(A1h)

Register to set the correction data of feed amount.

Bit	Name	Description
11:0	BR	Sets the backlash correction amount or slip correction amount. The setting range is 0 to 4,095.
13,12	ADJ	Sets the function to correct the feed amount. 00b: Not correct the feed amount 01b: Backlash correction 10b: Slip correction 11b: Prohibited
14	0	Always set to 0.
15	PSTP	Sets the processing when writing a stop command in pulser control. 0: Stops by ignoring the PA and PB signals that have been input. 1: Outputs the command pulse corresponding to the PA and PB signals that have been input, and then stops. Stops ignoring RENV6.PSTP = 1 in an interpolation control (68h, 69h, 6Ah, 6Bh, 6Ch, 6Dh).
26:16	PD	Sets the numerator for dividing the input of PA and PB signals. 0: Not divide. 1 to 2047: Divides the frequency to the set value / 2048.
31:27	PMG	Sets the value for multiplying the input of PA and PB signals. 0 to 31: Multiplies by the value obtained by adding 1 to the set value.

5.4.3.8 RENV7: Environment setting 7



A7 to A0 address of direct access method[68000, H8]: RENV7(74h)

A7 to A0 address of direct access method[8086, Z80]: RENV7(88h)

Register control command of indirect access method: RRENV7(E2h), WRENV7(A2h)

Register to set the control time of vibration suppression function.

Bit	Name	Description
15: 0	RT	<p>Sets the cycle of a reverse pulse.</p> <p>The cycle of a reverse pulse is the time obtained by multiplying 32 cycles of the CLK signal by the set value.</p> <p>The setting range is 0 to 65,535.</p>
31: 16	FT	<p>Sets the cycle of a forward pulse.</p> <p>The cycle of a forward pulse is the time obtained by multiplying 32 cycles of the CLK signal by the set value.</p> <p>The setting range is 0 to 65,535.</p>

For vibration suppression function, see "6.16 Vibration suppression".

..

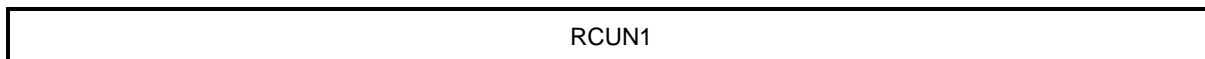
5.4.4 Counter register

They are registers for counters.

See "6.12 Counter" for counters.

5.4.4.1 RCUN1: Counter 1 (command position)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RCUN1(70h)

A7 to A0 address of direct access method[8086, Z80]: RCUN1(8Ch)

Register control command of indirect access method: RRCUN1(E3h), WRCUN1(A3h)

Register to acquire the count value in counter 1 (Command position).

Dedicates for counting command pulse signals.

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.4.2 RCUN2: Counter 2 (general-purpose 1)

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RCUN2(6Ch)

A7 to A0 address of direct access method[8086, Z80]: RCUN2(90h)

Register control command of indirect access method: RRCUN2(E4h), WRCUN2(A4h)

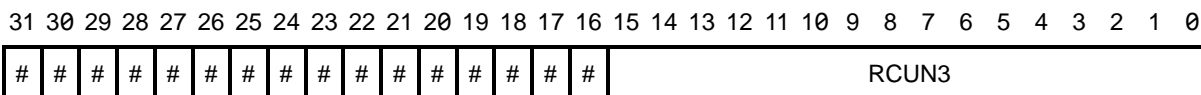
Register to acquire the count value of counter 2 (General-purpose 1).

With RENV3.CI2 bit, you can select the count from the following three types.

RENV3.CI2	Count target
00b	Encoder signal (EA,EB)
01b	Command pulse signal
10b	Manual pulser signal (PA,PB)
11b	Prohibited

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.4.3 RCUN3: Counter 3 (Deviation)



A7 to A0 address of direct access method[68000, H8]: RCUN3(68h)

A7 to A0 address of direct access method[8086, Z80]: RCUN3(94h)

Register control command of indirect access method: RRCUN3(E5h), WRCUN3(A5h)

Register to acquire the count value of counter 3 (deviation).

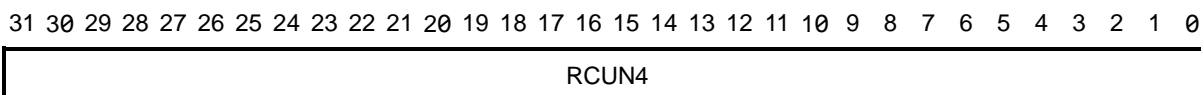
With RENV3.CI3 bit, you can select the count from the following three types.

RENV3.CI3	Count target
00b	Deviation between command pulse signals and encoder signals (EA,EB)
01b	Deviation between command pulse signals and manual pulser signals (PA,PB)
10b	Deviation between encoder signals (EA,EB) and manual pulser signals (PA,PB)
11b	Prohibited

The same value as bit 15 is read for # in bits 31 to 16.

The setting range is -32,768 to +32,767.

5.4.4.4 RCUN4: Counter 4 (General-purpose 2)



A7 to A0 address of direct access method[68000, H8]: RCUN4(64h)

A7 to A0 address of direct access method[8086, Z80]: RCUN4(98h)

Register control command of indirect access method: RRCUN4(E6h), WRCUN4(A6h)

Register to acquire the count value of counter 4 (General-purpose 2).

With RENV3.CI4 bit, you can select the count from the following four types.

RENV3.CI4	Count target
00b	Command pulse signals
01b	Encoder signals (EA,EB)
10b	Manual pulser signals (PA,PB)
11b	$\frac{f_{CLK}}{2}$ signals

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.5 Comparator register

This is the register for comparators.

For the comparator, see "6.13 Comparator".

5.4.5.1 RCMP1: Comparator 1 comparison value

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RCMP1(60h)
 A7 to A0 address of direct access method[8086, Z80]: RCMP1(9Ch)
 Register control command of indirect access method: RRCMP1(E7h), WRCMP1(A7h)

Register to set the comparison value in Comparator 1.

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.5.2 RCMP2: Comparator 2 comparison value

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RCMP2(5Ch)
 A7 to A0 address of direct access method[8086, Z80]: RCMP2(A0h)
 Register control command of indirect access method: RRCMP2(E8h), WRCMP2(A8h)

Register to set the comparison value in Comparator 2.

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.5.3 RCMP3: Comparator 3 comparison value

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RCMP3(58h)
 A7 to A0 address of direct access method[8086, Z80]: RCMP3(A4h)
 Register control command of indirect access method: RRCMP3(E9h), WRCMP3(A9h)

Register to set the comparison value in Comparator 3.

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.5.4 RCMP4: Comparator 4 comparison value

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



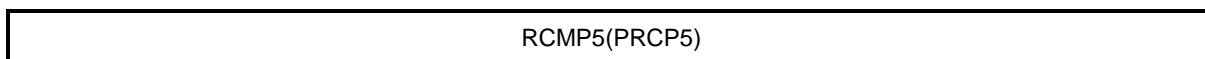
A7 to A0 address of direct access method[68000, H8]: RCMP4(54h)
 A7 to A0 address of direct access method[8086, Z80]: RCMP4(A8h)
 Register control command of indirect access method: RRCMP4(EAh), WRCMP4(AAh)

Register to set the comparison value in Comparator 4.

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.5.5 RCMP5(PRCP5): Comparator 5 comparison value

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RCMP5(50h), PRCP5(C8h)

A7 to A0 address of direct access method[8086, Z80]: RCMP5(ACh), PRCP5(34h)

Register control command of indirect access method: RRCMP5(EBh) , RPRCP5(CBh), WRCMP5(ABh), WPRCP5(8Bh)

Register to set the comparison value in Comparator 5.

PRCP5 register is the pre-register of RCMP5 register.

For the pre-register of RCMP5 register, see "6.2.2 Continuous comparison".

The setting range is -2,147,483,648 to +2,147,483,647.

5.4.6 Counter latch register

This is the register for counter latch.

Input of LTC and ORG signals or writing of LTCH (29h) command can latch the count value of the corresponding counter.

RLTC3 register can also latch the current speed step number.

See "6.12.3 Counter latch" for counter latches.

5.4.6.1 RLTC1: Counter 1 (Command position) latch

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RLTC1(48h)
 A7 to A0 address of direct access method[8086, Z80]: RLTC1(B4h)
 Register control command of indirect access method: RRLTC1(EDh)

Register to acquire the latch data of counter 1 (command position).

The data range is -2,147,483,648 to +2,147,483,647.

5.4.6.2 RLTC2: Counter 2 (General-purpose 1) latch

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RLTC2(44h)
 A7 to A0 address of direct access method[8086, Z80]: RLTC2(B8h)
 Register control command of indirect access method: RRLTC2(Eeh)

Register to acquire the latch data in counter 2 (General-purpose 1).

The data range is -2,147,483,648 to +2,147,483,647.

5.4.6.3 RLTC3: Counter 3 (Deviation) latch

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RLTC3(40h)
 A7 to A0 address of direct access method[8086, Z80]: RLTC3(BCh)
 Register control command of indirect access method: RRLTC3(EFh)

Counter 3 (Deviation), or the register to acquire the latch data of the current speed step number.

The # of bits 31 to 16 is the same value as bit 15 when RENV5.LTFD = 0. When RENV5.LTFD = 1, 0 is read.

The data range is -32,768 to +32,767 when RENV5.LTFD = 0. When RENV5.LTFD = 1, it is 0 to 65,535.

5.4.6.4 RLTC4: Counter 4 (General-purpose 2) latch

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



A7 to A0 address of direct access method[68000, H8]: RLTC4(3Ch)
 A7 to A0 address of direct access method[8086, Z80]: RLTC4(C0h)
 Register control command of indirect access method: RRLTC4(F0h)

Register to acquire the latch data of counter 4 (General-purpose 2).

The data range is -2,147,483,648 to +2,147,483,647.

5.4.7 Interrupt control register

This is a register for interrupt control.

For interrupts, see "6.18 Interrupt request (INT)".

5.4.7.1 RIRQ: Event interrupt request

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IROL	IRLT	IRCL	IRC5	IRC4	IRC3	IRC2	IRC1	IRDE	IRDS	IRUE	IRUS	IRND	IRNM	IRN	IREN
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	IRSA	IRDR	IRSD

A7 to A0 address of direct access method[68000, H8]: RIRQ(4Ch)

A7 to A0 address of direct access method[8086, Z80]: RIRQ(B0h)

Register control command of indirect access method: RRIRQ(ECh), WRIRQ(ACh)

Register to set the event interrupt request.

When an event interrupt factor that you set "1" in RIRQ register occurs, the bit in the corresponding RIST register becomes 1.

Bit	Name	Description
0	IREN	1: An interrupt is generated when the operation mode stops normally.
1	IRN	1: An interrupt is generated when the pre-register is determined (RSTS.PFM > 0) when the operation stops.
2	IRNM	1: An interrupt is generated when 2nd pre-register for continuous operation changes to be writable. (MSTS.SPRF bit changed from 1 to 0)
3	IRND	1: An interrupt is generated when 2nd pre-register for continuous comparison changes to be writable. (MSTS.SPDP bit changed from 1 to 0)
4	IRUS	1: An interrupt is generated when acceleration is started. (SSTS.SFU bit changed from 0 to 1)
5	IRUE	1: An interrupt is generated when acceleration is completed. (SSTS.SFU bit changed from 1 to 0)
6	IRDS	1: An interrupt is generated when deceleration is started. (SSTS.SFD bit changed from 0 to 1)
7	IRDE	1: An interrupt is generated when deceleration is completed. (SSTS.SFD bit changed from 1 to 0)
8	IRC1	1: An interrupt is generated when the comparison condition of comparator 1 is satisfied. (MSTS.SCP1 changed from 0 to 1)
9	IRC2	1: An interrupt is generated when the comparison condition of comparator 2 is satisfied. (MSTS.SCP2 changed from 0 to 1)
10	IRC3	1: An interrupt is generated when the comparison condition of comparator 3 is satisfied. (MSTS.SCP3 changed from 0 to 1)
11	IRC4	1: An interrupt is generated when the comparison condition of comparator 4 is satisfied. (MSTS.SCP4 changed from 0 to 1)
12	IRC5	1: An interrupt is generated when the comparison condition of comparator 5 is satisfied. (MSTS.SCP5 changed from 0 to 1)

Bit	Name	Description
13	IRCL	1: An interrupt occurs when CLR signal is turned ON and the count value is cleared. If there is no counter to clear (RENV3.CU1C to CU4C = 0000b), no interrupt is generated.
14	IRLT	1: An interrupt occurs when LTC signal is turned ON and the count value is latched. If there is no LTC signal at the latching timing (RENV5.LTM ≠ 00b), no interrupt is generated.
15	IROL	1: An interrupt occurs when the ORG signal is turned ON and the count value is latched. If there is no ORG signal at the latching timing (RENV5.LTM ≠ 01b), no interrupt is generated.
16	IRSD	1: An interrupt occurs when SD signal in the operating direction turns ON.
17	IRDR	1: An interrupt occurs when either +DR signal or -DR signal changes. When PEn = H level, no interrupt is generated.
18	IRSA	1: An interrupt is generated when CSTA signal is turned ON (RENV1.PCSM = 0). An interrupt is also generated when STA signal is turned ON (RENV1.PCSM = 1).
31:19	0	Always set to 0.

5.4.7.2 REST: Error interrupt factor

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESAO	ESPO	ESIP	ESDT	0	ESSD	ESEM	ESSP	ESAL	ESML	ESPL	ESC5	ESC4	ESC3	ESC2	ESC1
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ESPE	ESEE

A7 to A0 address of direct access method[68000, H8]: REST(34h)

A7 to A0 address of direct access method[8086, Z80]: REST(C8h)

Register control command of indirect access method: RREST(F2h), WREST(B2h)

Register to acquire the cause of an error interrupt.

The error interrupt factor occurs only when each condition is satisfied.

When an error interrupt factor occurs, the corresponding bit in REST register becomes 1.

When any bit of REST register is 1, L level can be output from INT pin.

In REST register, writing 1 to the corresponding bit clears the bit to 0.

If RENV5.ISMR = 0 is set, all bits will be cleared to 0 even when writing RREST (F2h) command.

If RENV5.ISMR = 1 is set, they will not be cleared when writing RREST (F2h) command.

(The setting of RENV5.ISMR bit also affects RIST register)

Bit	Name	Description
0	ESC1	1: An abnormal stop occurred because the comparison condition of Comparator 1 was satisfied. (Including stop by +SL)
1	ESC2	1: An abnormal stop occurred because the comparison condition of Comparator 2 was satisfied. (Including stop by -SL)
2	ESC3	1: An abnormal stop occurred because the comparison condition of Comparator 3 was satisfied.
3	ESC4	1: An abnormal stop occurred because the comparison condition of Comparator 4 was satisfied.
4	ESC5	1: An abnormal stop occurred because the comparison condition of Comparator 5 was satisfied.
5	ESPL	1: Stopped abnormally because +EL signal turned ON.
6	ESML	1: Stopped abnormally because -EL signal turned ON.
7	ESAL	1: Stopped abnormally because ALM signal turned ON.
8	ESSP	1: Stopped abnormally because CSTP signal turned ON.
9	ESEM	1: Stopped abnormally because CEMG signal turned ON.
10	ESSD	1: Stopped abnormally when RENV1.SDM = 1 and SD signal in the operating direction turned ON.
11	0	Always acquire 0.
12	ESDT	1: Stopped abnormally because there was an error in interpolation setting data. * 1
13	ESIP	1: Stopped abnormally because the interpolation axis other than own axis stopped abnormally during an interpolation operation,
14	ESPO	1: Stops abnormally because the buffer counter (16 bit) for inputting PA and PB signals overflowed.
15	ESAO	1: Stops abnormally because the circular interpolation range (signed 32 bits) was exceeded during an interpolation operation.
16	ESEE	1: EA and EB signal input errors have occurred. * 2 Operation mode will not stop.
17	ESPE	1: PA and PB signal input errors have occurred. * 3 Operation mode will not stop.

Bit	Name	Description
31:18	0	Always acquire 0.

- * 1 ESDT: An error in the interpolation setting data occurs when a start command is written in the following statuses.
- (1) In the operation mode of linear interpolation 1 control (RMD.MOD = 60h, 61h, 68h, 69h), two or more axes are not set.
 - (2) In the operation mode of linear interpolation 2 control (RMD.MOD = 62h, 63h, 6Ah, 6Bh), the feed amount of the main axis is not set (one or higher number is not set in RIP register).
 - (3) The number of axes in the operation mode of circular interpolation control (RMD.MOD = 64h, 65h, 66h, 67h, 6Ch, 6Dh) is not two.
 - (4) The center position is not set in the operation mode of circular interpolation control.
(0 or higher number is not set in the RIP register of both axes)
 - (5) U-axis does not operate in the U-axis interpolation control operation mode (RMD.MOD = 66h, 67h).

The error occurs if U-axis stops first in the operation mode of U-axis interpolation control.

- * 2 ESEE: EA and EB signals changed simultaneously in 90-degree phase difference mode. Or the signals were input simultaneously in 2-pulse mode. The error occurs when the power of an encoder is turned ON or when noise is detected.
If the cause is from power-ON, no action is required.
If the cause is from noise, you need to take an action depending on the frequency of occurrence.
- * 3 ESPE: PA and PB signals changed simultaneously in 90-degree phase difference mode. Or the signals were input simultaneously in 2-pulse mode.
The error occurs when the power of a manual pulser is turned ON or when noise is detected.
If the cause is from power-ON, no action is required.
If the cause is from noise, you need to take an action depending on the frequency of occurrence.

5.4.7.3 RIST: Event interrupt factor

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISOL	ISLT	ISCL	ISC5	ISC4	ISC3	ISC2	ISC1	ISDE	ISDS	ISUE	ISUS	ISND	ISNM	ISN	ISEN
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	ISSA	ISMd	ISPD	ISSD

A7 to A0 address of direct access method[68000, H8]: RIST(30h)

A7 to A0 address of direct access method[8086, Z80]: RIST(CCh)

Register control command of indirect access method: RRIST(F3h), WRIST(B3h)

Register to acquire the event interrupt factor.

The event interrupt factor occurs when the condition of RIRQ register is satisfied.

When an event interrupt factor occurs, the corresponding bit in RIST register becomes 1.

When any bit of RIST register is 1, L level can be output from INT pin.

In RIST register, writing 1 to the corresponding bit clears the bit to 0.

If RENV5.ISMR = 0 is set, all bits will be cleared to 0 even when writing RRIST (F3h) command.

If RENV5.ISMR = 1 is set, it will not be cleared when writing the RRIST (F3h) command.

(The setting of RENV5.ISMR bit also affects REST register)

Bit	Name	Description
0	ISEN	1: The operation mode stopped normally.
1	ISN	1: The pre-register shifted due to the operation stop.
2	ISNM	1: The 2nd pre-register for continuous operation changed to be writable. (MSTS.SPRF bit changed from 1 to 0)
3	ISND	1: The 2nd pre-register for continuous comparison changed to be writable. (MSTS.SPFD bit changed from 1 to 0)
4	ISUS	1: Acceleration started. (SSTS.SFU bit changed from 0 to 1)
5	ISUE	1: Acceleration ended. (SSTS.SFU bit changed from 1 to 0)
6	ISDS	1: Deceleration started. (SSTS.SFD bit changed from 0 to 1)
7	ISDE	1: Deceleration ended. (SSTS.SFD bit changed from 1 to 0)
8	ISC1	1: The comparison condition of comparator 1 was satisfied. (MSTS.SCP1 changed from 0 to 1)
9	ISC2	1: The comparison condition of comparator 2 was satisfied. (MSTS.SCP2 changed from 0 to 1)
10	ISC3	1: The comparison condition of Comparator 3 was satisfied. (MSTS.SCP3 changed from 0 to 1)
11	ISC4	1: The comparison condition of Comparator 4 was satisfied. (MSTS.SCP4 changed from 0 to 1)

Bit	Name	Description
12	ISC5	1: The comparison condition of Comparator 5 was satisfied. (MSTS.SCP5 changed from 0 to 1)
13	ISCL	1: CLR signal was turned ON to clear the count value. If there is no counter to clear (RENV3.CU1C to CU4C = 0000b), no interrupt is generated.
14	ISLT	1: LTC signal was turned ON to latch the count value. If there is no LTC signal at the latching timing (RENV5.LTM ≠ 00b), no interrupt is generated.
15	ISOL	1: ORG signal was turned ON to latch the count value. If there is no ORG signal at the latching timing (RENV5.LTM ≠ 01b), no interrupt is generated.
16	ISSD	1: SD signal in the operating direction was turned ON.
17	ISPD	1: +DR signal was changed. When PEn = H level, no interrupt is generated.
18	ISMD	1: -DR signal was changed. When PEn = H level, no interrupt is generated.
19	ISSA	1: CSTA signal is ON (RENV1.PCSM = 0) or STA signal is ON (RENV1.PCSM = 1).
31: 20	0	Always acquires 0.

5.4.8 Status display register

This register is for indicating the status.

5.4.8.1 RSTS: Extension status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSDI	SLTC	SCLR	SDRM	SDRP	SEZ	SERC	SPCS	SEMG	SSTP	SSTA	SDIR	CND			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	MSDL	PSDL	PFM		PFC		MSDI	SINP

A7 to A0 address of direct access method[68000, H8]: RSTS(38h)
A7 to A0 address of direct access method[8086, Z80]: RSTS(C4h)
Register control command of indirect access method: RRSTS(F1h)

Register to acquire the operation mode and the status of various signals.

Bit	Name	Description																
3: 0	CND	<p>Indicates the operating status.</p> <table border="0"> <tr> <td>0000b: Stopping</td><td>0001b: Waiting for +DR, -DR signal input</td></tr> <tr> <td>0010b: Waiting for CSTA signal input</td><td>0011b: Waiting for input of internal synchronization signal</td></tr> <tr> <td>0100b: Waiting for the other axis to stop</td><td>0101b: Waiting for ERC signal OFF timer to complete</td></tr> <tr> <td>0110b: Waiting for the direction change timer to complete</td><td>0111b: Backlash correction is in progress</td></tr> <tr> <td>1000b: Waiting for input of PA and PB signals</td><td>1001b: FA constant speed operation is in progress</td></tr> <tr> <td>1010b: FL constant speed operation is in progress</td><td>1011b: Accelerating</td></tr> <tr> <td>1100b: FH constant speed operation is in progress</td><td>1101b: Decelerating</td></tr> <tr> <td>1110b: Waiting for input of INP signal</td><td>1111b: Other (during start control)</td></tr> </table> <p>The other status (RSTS.CND = 1111b) shifts to another status by inputting the CLK signal several times.</p>	0000b: Stopping	0001b: Waiting for +DR, -DR signal input	0010b: Waiting for CSTA signal input	0011b: Waiting for input of internal synchronization signal	0100b: Waiting for the other axis to stop	0101b: Waiting for ERC signal OFF timer to complete	0110b: Waiting for the direction change timer to complete	0111b: Backlash correction is in progress	1000b: Waiting for input of PA and PB signals	1001b: FA constant speed operation is in progress	1010b: FL constant speed operation is in progress	1011b: Accelerating	1100b: FH constant speed operation is in progress	1101b: Decelerating	1110b: Waiting for input of INP signal	1111b: Other (during start control)
0000b: Stopping	0001b: Waiting for +DR, -DR signal input																	
0010b: Waiting for CSTA signal input	0011b: Waiting for input of internal synchronization signal																	
0100b: Waiting for the other axis to stop	0101b: Waiting for ERC signal OFF timer to complete																	
0110b: Waiting for the direction change timer to complete	0111b: Backlash correction is in progress																	
1000b: Waiting for input of PA and PB signals	1001b: FA constant speed operation is in progress																	
1010b: FL constant speed operation is in progress	1011b: Accelerating																	
1100b: FH constant speed operation is in progress	1101b: Decelerating																	
1110b: Waiting for input of INP signal	1111b: Other (during start control)																	
4	SDIR	<p>Indicates the direction of an operation.</p> <p>0: +Direction 1: -Direction</p>																
5	SSTA	<p>Indicates the input status of CSTA signal or STA signal.</p> <p>0: OFF 1: ON</p> <p>The input logic of CSTA signal is in negative. The input logic of STA signal is selected by RENV1.PCSL bit.</p>																
6	SSTP	<p>Indicates the input status of CSTP signal.</p> <p>0: OFF 1: ON</p> <p>The input logic of CSTP signal is in negative.</p>																
7	SEMG	<p>Indicates the input status of CEMG signal.</p> <p>0: OFF 1: ON</p> <p>The input logic of CEMG signal is in negative.</p>																

Bit	Name	Description
8	SPCS	Indicates the status of a signal input to PCSn pin. 0: OFF 1: ON The logic of a signal input to PCSn pin is selected by RENV1.PCSL bit.
9	SERC	Indicates the output status of ERC signal. 0: OFF 1: ON The logic of an ERC signal is selected by RENV1.ERCL bit.
10	SEZ	Indicates the input status of EZ signal. 0: OFF 1: ON The input logic of an EZ signal is selected by RENV2.EZL bit.
11	SDRP	Indicates the input status of +DR signal. 0: OFF 1: ON The input logic of a +DR signal and a -DR signal is selected by RENV1.DRL bit. The status also changes when PEn = H level.
12	SDRM	Indicates the input status of -DR signal. 0: OFF 1: ON The input logic of a +DR signal and a -DR signal is selected by RENV1.DRL bit. The status also changes when PEn = H level.
13	SCLR	Indicates the input status of CLR signal. 0: OFF 1: ON The input logic of an CLR signal is selected by RENV1.CLRL bit.
14	SLTC	Indicates the input status of LTC signal. 0: OFF 1: ON The input logic of an LTC signal is selected by RENV1.LTCL bit.
15	PSDI	Indicates the input status of +SD signal. 0: OFF 1: ON The input logic of +SD and -SD signals is selected by RENV1.SDL bit.

Bit	Name	Description																				
16	SINP	Indicates the input status of INP signal. 0: OFF 1: ON The input logic of an INP signal is selected by RENV1.INPL bit.																				
17	MSDI	Indicates the input status of –SD signal. 0: OFF 1: ON The input logic of +SD and –SD signals is selected with RENV1.SDL bit.																				
19,18	PFC	Indicates the determined status of the pre-register for a continuous comparison. <table border="1"> <thead> <tr> <th>PFC</th> <th>2nd pre-register (PRCP5)</th> <th>1st pre-register</th> <th>Current register (RCMP5)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Undetermined</td> <td>Undetermined</td> <td>Undetermined</td> </tr> <tr> <td>1</td> <td>Undetermined</td> <td>Undetermined</td> <td>Determined</td> </tr> <tr> <td>2</td> <td>Undetermined</td> <td>Determined</td> <td>Determined</td> </tr> <tr> <td>3</td> <td>Determined</td> <td>Determined</td> <td>Determined</td> </tr> </tbody> </table>	PFC	2nd pre-register (PRCP5)	1st pre-register	Current register (RCMP5)	0	Undetermined	Undetermined	Undetermined	1	Undetermined	Undetermined	Determined	2	Undetermined	Determined	Determined	3	Determined	Determined	Determined
PFC	2nd pre-register (PRCP5)	1st pre-register	Current register (RCMP5)																			
0	Undetermined	Undetermined	Undetermined																			
1	Undetermined	Undetermined	Determined																			
2	Undetermined	Determined	Determined																			
3	Determined	Determined	Determined																			
21,20	PFM	Indicates the determined status of the pre-register for a continuous operation. <table border="1"> <thead> <tr> <th>PFM</th> <th>2nd pre-register</th> <th>1st pre-register</th> <th>Current register</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Undetermined</td> <td>Undetermined</td> <td>Undetermined</td> </tr> <tr> <td>1</td> <td>Undetermined</td> <td>Undetermined</td> <td>Determined</td> </tr> <tr> <td>2</td> <td>Undetermined</td> <td>Determined</td> <td>Determined</td> </tr> <tr> <td>3</td> <td>Determined</td> <td>Determined</td> <td>Determined</td> </tr> </tbody> </table>	PFM	2nd pre-register	1st pre-register	Current register	0	Undetermined	Undetermined	Undetermined	1	Undetermined	Undetermined	Determined	2	Undetermined	Determined	Determined	3	Determined	Determined	Determined
PFM	2nd pre-register	1st pre-register	Current register																			
0	Undetermined	Undetermined	Undetermined																			
1	Undetermined	Undetermined	Determined																			
2	Undetermined	Determined	Determined																			
3	Determined	Determined	Determined																			
22	PSDL	Indicate the latch status of a +SD signal. 0: OFF 1: ON The input logic for +SD and –SD signals is selected with RENV1.SDL bit.																				
23	MSDL	Indicates the latch status of –SD signal. 0: OFF 1: ON The input logic for +SD and –SD signals is selected with RENV1.SDL bit.																				
31:24	0	Always acquires 0.																				

5.4.8.2 RIPS: Interpolation status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPFu	IPFz	IPFy	IPFx	IPSu	IPSz	IPSy	IPSx	IPEu	IPEz	IPEy	IPEx	IPLu	IPLz	IPLy	IPLx
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	SED	SDM	IPCC	IPCW	IPE	IPL

A7 to A0 address of direct access method[68000, H8]: RIPS(00h)

A7 to A0 address of direct access method[8086, Z80]: RIPS(FCh)

Register control command of indirect access method: RRIPS(FFh)

This register acquires the status of various interpolation controls.

Bit	Name	Description
0	IPLx	1: Linear interpolation 1 operation mode is set in X-axis
1	IPLy	1: Linear interpolation 1 operation mode is set in Y-axis
2	IPLz	1: Linear interpolation 1 operation mode is set in Z-axis
3	IPLu	1: Linear interpolation 1 operation mode is set in U-axis
4	IPEx	1: Linear interpolation 2 operation mode is set in X-axis
5	IPEy	1: Linear interpolation 2 operation mode is set in Y-axis
6	IPEz	1: Linear interpolation 2 operation mode is set in Z-axis
7	IPEu	1: Linear interpolation 2 operation mode is set in U-axis
8	IPSx	1: Circular interpolation operation mode is set in X-axis
9	IPSy	1: Circular interpolation operation mode is set in Y-axis
10	IPSz	1: Circular interpolation operation mode is set in Z-axis
11	IPSu	1: Circular interpolation operation mode is set in U-axis
12	IPFx	1: Constant synthesized speed operation is set in X-axis
13	IPFy	1: Constant synthesized speed operation is set in Y-axis
14	IPFz	1: Constant synthesized speed operation is set in Z-axis
15	IPFu	1: Constant synthesized speed operation is set in U-axis
16	IPL	1: Operation mode of linear interpolation 1 is in progress
17	IPE	1: Operation mode of linear interpolation 2 is in progress
18	IPCW	1: Circular interpolation operation in the CW direction is in progress
19	IPCC	1: Circular interpolation operation in the CWW direction is in progress
21,20	SDM	Indicates the current quadrant in a circular interpolation operation mode (00: 1st quadrant, 01: 2nd quadrant, 10: 3rd quadrant, 11: 4th quadrant)
23,22	SED	Indicates the final quadrant in a circular interpolation operation mode (00: 1st quadrant, 01: 2nd quadrant, 10: 3rd quadrant, 11: 4th quadrant)
31:24	0	(Always acquire 0.)

5.5 Operation mode

There are 44 operation modes to select with the combinations of control methods and movement methods.

Selects the operation mode by RMD.MOD bit.

Name and description	Target
<Operation mode selection> 000 0000 (00h): Operation mode of continuous movement in +direction in command control 000 1000 (08h): Operation mode of continuous movement in -direction in command control 000 0001 (01h): Operation mode of continuous movement in pulser control 000 0010 (02h): Operation mode of continuous movement in switch control 001 0000 (10h): Operation mode of origin return in +direction in origin return control 001 1000 (18h): Operation mode of origin return in -direction in origin return control 001 0010 (12h): Operation mode to escape from origin in +direction in origin return control 001 1010 (1Ah): Operation mode to escape from origin in -direction in origin return control 001 0101 (15h): Operation mode of origin search in +direction in origin return control 001 1101 (1Dh): Operation mode of origin search in -direction in origin return control 010 0000 (20h): Operation mode to move up to +EL or +SL in sensor control 010 1000 (28h): Operation mode to move up to -EL or -SL in sensor control 010 0010 (22h): Operation mode to escape from -EL or -SL in sensor control 010 1010 (2Ah): Operation mode to escape from +EL or +SL in sensor control 010 0100 (24h): Operation mode of movement for EZ count in +direction in sensor control 010 1100 (2Ch): Operation mode of movement for EZ count in -direction in sensor control 100 0000 (41h): Operation mode of incremental movement in positioning control 100 1000 (42h): Operation mode in which the absolute position is specified in counter 1 in positioning control 100 0010 (43h): Operation mode in which the absolute position is specified in counter 2 in positioning control 100 1010 (44h): Operation mode to return to zero point with counter 1 in positioning control 100 0100 (45h): Operation mode to return to zero point with counter 2 in positioning control 100 1100 (46h): Operation mode of one pulse in +direction in positioning control 100 0110 (4Eh): Operation mode of one pulse in -direction in positioning control 100 0111 (47h): Timer operation mode in positioning control 101 0001 (51h): Operation mode of incremental movement in pulser control 101 0010 (52h): Operation mode in which the absolute position is specified in counter 1 in pulser control 101 0011 (53h): Operation mode in which the absolute position is specified in counter 2 in pulser control 101 0100 (54h): Operation mode to return to zero point with counter 1 in pulser control 101 0101 (55h): Operation mode to return to zero point with counter 2 in pulser control 101 0110 (56h): Operation mode of incremental movement in switch control 110 0000 (60h): Operation mode of continuous movement in linear interpolation 1 control 110 0001 (61h): Operation mode of incremental movement in linear interpolation 1 control 110 0010 (62h): Operation mode of continuous movement in linear interpolation 2 control 110 0011 (63h): Operation mode of incremental movement in linear interpolation 2 control 110 0100 (64h): Operation mode of circular interpolation in CW direction in circular interpolation control 110 0101 (65h): Operation mode of circular interpolation in CCW direction in circular interpolation control 110 0110(66h): Operation mode of circular interpolation in CW direction in U-axis synchronization control 110 0111 (67h): Operation mode of circular interpolation in CCW direction in U-axis synchronization control 110 1000 (68h): Operation mode of continuous movement with linear interpolation 1 in pulser control 110 1001 (69h): Operation mode of incremental movement with linear interpolation 1 in pulser control 110 1010 (6Ah): Operation mode of continuous movement with linear interpolation 2 in pulser control 110 1011 (6Bh): Operation mode of incremental movement with linear interpolation 2 in pulser control 110 1100 (6Ch): Operation mode of circular interpolation in CW direction in pulser control 110 1101 (6Dh): Operation mode of circular interpolation in CCW direction in pulser control	RMD.MOD(6:0)
Do not set any other values.	

5.5.1 Command control

This control method is to stop with a stop command.

5.5.1.1 Continuous movement in plus direction (00h)

Starts to output command pulses in +direction when started.

Stops to output command pulses when writing a stop command.

When the command pulse is stopped, the operation mode is completed.

You can flexibly control the speed during operation by using the target speed override and speed change commands.

5.5.1.2 Continuous movement in minus direction (08h)

Starts to output command pulses in -direction when started.

Stops to output command pulses when writing a stop command.

When the command pulse is stopped, the operation mode is completed.

You can flexibly control the speed during operation by using the target speed override and speed change commands.

5.5.2 Positioning control

This control method is to stop when the number of remaining pulses becomes 0 (RPLS = 0).

When the RMV register value is changed, the RPLS register value is updated to the RMV register absolute value.

You can stop halfway with a stop command.

5.5.2.1 Incremental movement (41h)

When starting, the RPLS register value is updated with the absolute value of RMV register.

When started, command pulses start to be output in +direction if $RMV > 0$ and in -direction if $RMV < 0$.

If the speed pattern is either high speed 1 or high speed 2, starts decelerating when $RPLS < RDP$.

When $RPLS = 0$, command pulses will stop.

When command pulses stop, the operation mode is completed.

When starting with $RMV = 0$ ($RPLS = 0$), the operation mode completes without outputting command pulses.

5.5.2.2 Specify the absolute position by counter 1 (42h)

When starting, the RPLS register value is updated with the absolute value of the difference between RCUN1 register value and RMV register value. When started, command pulses will be output in +direction if $RMV > RCUN1$ and in -direction if $RMV < RCUN1$. If the speed pattern is either high speed 1 or high speed 2, starts decelerating when $RPLS < RDP$.

When $RPLS = 0$, command pulse will stop.

When command pulses stop, the operation mode is completed.

When starting with $RMV = RCUN1$ ($RPLS = 0$), the operation mode completes without outputting command pulses.

5.5.2.3 Specify the absolute position by counter 2 (43h)

Same as RMD.MOD = 42h, except that RCUN2 register is used instead of RCUN1 register.

5.5.2.4 Zero-point return by counter 1 (44h)

With RMD.MOD = 42h, the operation is the same as when RMV = 0 is set.

Other than RMV = 0, the RPLS register value is also updated with RMV = 0.

5.5.2.5 Zero-point return by counter 2 (45h)

With RMD.MOD = 43h, the operation is the same as when RMV = 0 is set.

Other than RMV = 0, the RPLS register value is also updated with RMV = 0.

5.5.2.6 One pulse in plus direction (46h)

With RMD.MOD = 41h, the operation is the same as when RMV = 1 is set.

Other than RMV = 1, the RPLS register value is also updated with RMV = 1.

5.5.2.7 One pulse in minus direction (4Eh)

With RMD.MOD = 41h, the operation is the same as when RMV = -1 is set.

Other than RMV = -1, the RPLS register value is also updated as RMV = -1.

5.5.2.8 Timer (47h)

When starting, the RPLS register value is updated with RMV register.

When started, command pulses are not output until RPLS = 0.

Use FL constant speed or FH constant speed for the speed pattern.

When RPLS = 0, the operation mode is completed.

Please set the RMV register value from 1 to 2,147,483,647.

When starting with RMV = 0 (RPLS = 0), the operation mode will be completed.

You can also complete the operation mode by the stop command.

You can use it to set an arbitrary stop time between operation modes in continuous operation using a pre-register.

The operating time in the timer mode can be used as a pause timer in continuous operations.

(For example, if you set 120 pulses at 1000 pps, the operation will pause for 120 ms)

The operation does not stop by +EL, -EL, +SD, -SD signal inputs and software limit.

It stops when inputting ALM, CSTP, or CEMG signal.

Backlash correction, slip correction, vibration suppression, or direction change timer function will be disabled.

Counter 1 does not operate because this LSI does not output a command pulse.

Even if RMD.MINP = 1, no delay in completing the operation mode due to the INP signal.

Please set RMD.METM = 0 to reduce the error in internal operating time.

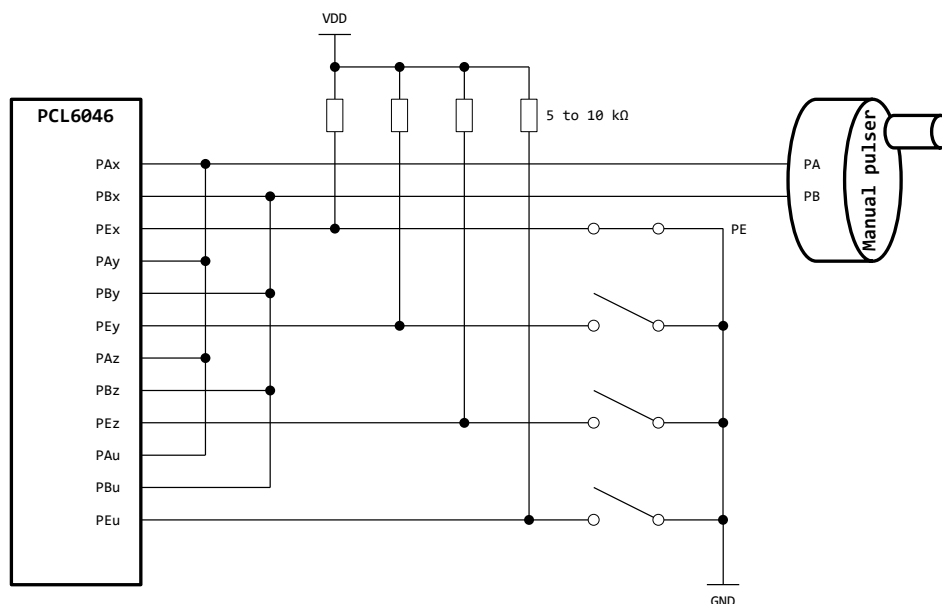
5.5.3 Pulser control

Each operation mode is controlled in synchronization with inputs of PA and PB signals.

It can be used when $PE_n = L$ level and $RENV2.POFF = 0$.

PE_n pin allows a pair of manual pulsers to switch between multiple axes.

Since PE_n pin has a built-in pull-up circuit, the PA and PB signal inputs are disabled when they are open.



The input noise filter can be set for PE signal with $RENV1.DRF$ bit as well as for PA and PB signals with $RENV2.PINF$ bit.

When starting, this LSI will wait for inputs of PA and PB signals ($RSTS.CND = 1000b$).

Consequently, the command pulse is output in synchronization with PA signal and PB signal inputs.

Can use FH constant speed for the speed pattern.

The backlash correction function also works in pulser control.

However, if PA and PB signals inputs are reversed during backlash correction, the function will not work.

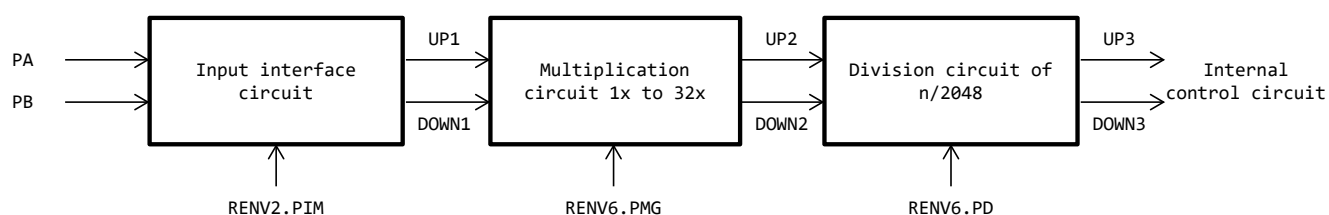
The input specifications of PA and PB signals can be selected from the following four types with $RENV2.PIM$ bit.

- 90-degree phase difference mode 1x
- 90-degree phase difference mode 2x
- 90-degree phase difference mode 4x
- 2 pulse mode

Three 90-degree phase difference modes are via the multiplying circuit (1 to 32) and dividing circuit ($n / 2048$).

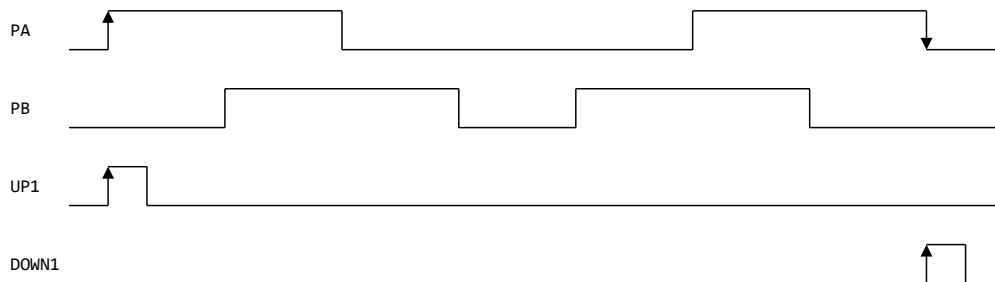
Maximum 128 multiplications (90-degree phase difference mode 4x, 32 multiplications, no division).

The multiplication is set by $RENV6.PMG$ bit, and the division is set with $RENV6.PD$ bit.

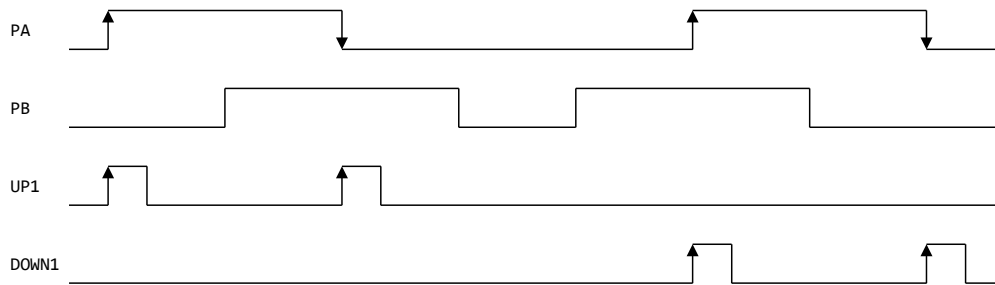


UP1 and DOWN1 signals are as follows by setting the RENV2.PIM bit.

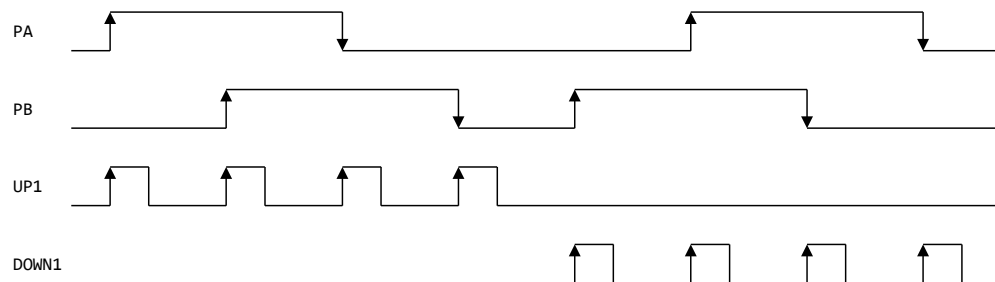
1. 90-degree phase difference mode 1x (RENV2.PIM = 00b)



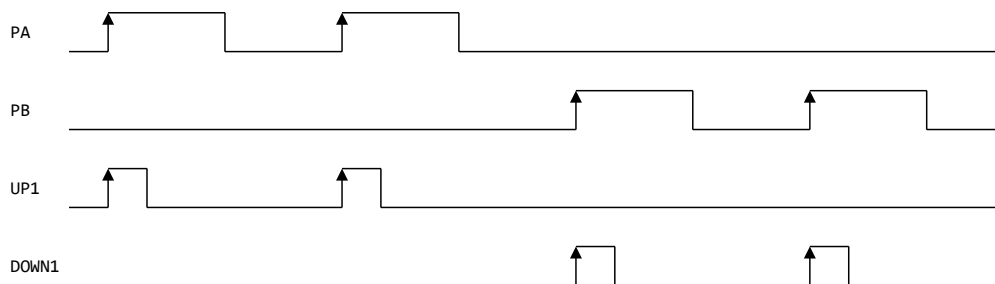
2. 90-degree phase difference mode 2x (RENV2.PIM = 01b)



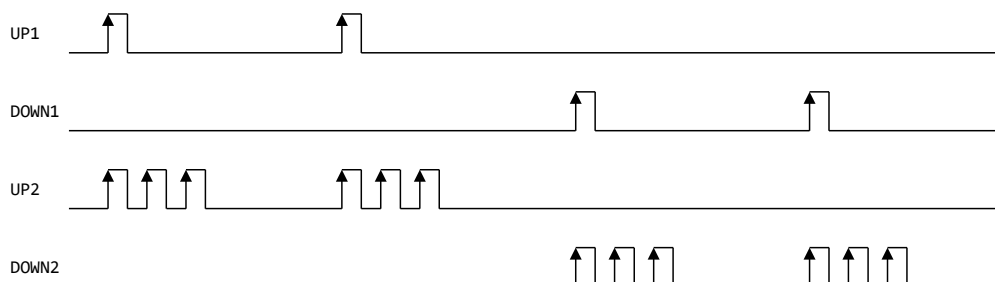
3. 90-degree phase difference mode 4x (RENV2.PIM = 10b)



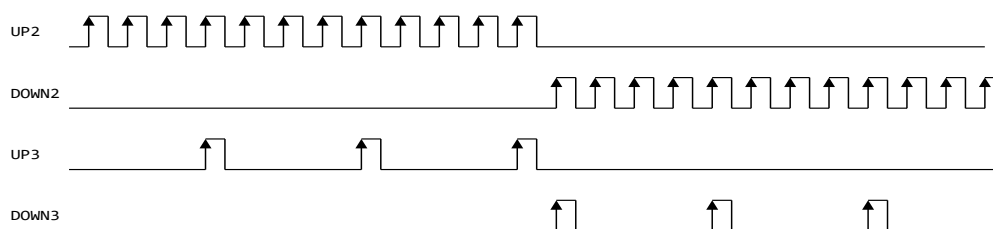
4. 2-pulse mode (RENV2.PIM = 11b)



In RENV6.PMG = 2 (3x) , it becomes as follows.



In $RENV6.PD = 512$ (512/2048 division) is set, it becomes as follows.



In synchronization with UP3 and DOWN3 signals, the internal pulse of FH speed is output with some being omitted.

Therefore, the input timing of PA and PB signals and the output timing of command pulses will have an error of the internal pulse cycle at the longest.

FP (maximum input frequency) of PA and PB signals is limited by FH speed, input specifications, multiplication settings, and frequency division settings. The frequency is not constant because the pulser is rotated manually.

Please set the FP to be faster than necessary.

If FH speed is increased in accordance with FP, the output pulse width becomes narrower. For FH speed, the upper limit of the set speed is the max input speed of the motor driver. However, when using the multiplication function with a stepping motor, the set speed must be lower than the motor starting frequency.

When the input frequency exceeds the FH speed, it is buffered by the input buffer counter (16 bit).

If the input buffer counter overflows, a $REST.ESPO = 1$ error interrupt will occur.

If the PA signal and PB signal inputs change at the same time, an error interrupt of $REST.ESPE = 1$ will occur.

- 1) When $RENV6.PD \neq 0$

$$FP < FH \div PIMG \div (RENV6.PMG + 1) \div (RENV6.PD \div 2048)$$

- 2) When $RENV6.PD = 0$

$$FP < FH \div PIMG \div (RENV6.PMG + 1)$$

$PIMG$ in the formula is as follows, depending on the setting of $RENV2.PIM$ bit.

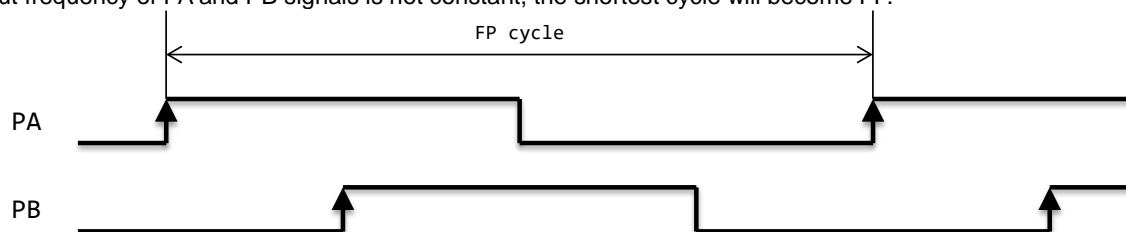
$RENV2.PIM$	$PIMG$
00b (90-degree phase difference mode 1x)	1
01b (90-degree phase difference mode 2x)	2
10b (90-degree phase difference mode 4x)	4
11b (2 pulse mode)	1

The calculation examples including $RENV6.PMG$ bit and $RENV6.PD$ bit are as follows.

$RENV2.PIM$	$RENV6.PMG$	$RENV6.PD$	Calculation result
00b (90-degree phase difference mode 1x)	0(1x)	0	$FP < FH \div 1 \div 1 = FH$
	0(1x)	1024	$FP < FH \div 1 \div 1 \div \frac{1}{2} = FH \times 2$
	2(3x)	0	$FP < FH \div 1 \div 3 = FH \div 3$

01b (90-degree phase difference mode 2x)	0(1x)	0	$FP < FH \div 2 \div 1 = FH \div 2$
	0(1x)	1024	$FP < FH \div 2 \div 1 \div \frac{1}{2} = FH$
	2(3x)	0	$FP < FH \div 2 \div 3 = FH \div 6$
10b (90-degree phase difference mode 4x)	0(1x)	0	$FP < FH \div 4 \div 1 = FH \div 4$
	0(1x)	1024	$FP < FH \div 4 \div 1 \div \frac{1}{2} = FH \div 2$
	2(3x)	0	$FP < FH \div 4 \div 3 = FH \div 12$
11b (2-pulse mode)	0(1x)	0	$FP < FH \div 1 \div 1 = FH$
	0(1x)	1024	$FP < FH \div 1 \div 1 \div \frac{1}{2} = FH \times 2$
	2(3x)	0	$FP < FH \div 1 \div 3 = FH \div 3$

If the input frequency of PA and PB signals is not constant, the shortest cycle will become FP.



When stopping immediately with STOP (0049h) command, the total output pulse is not always the integral multiple of a multiplication value.

Set RENV6.PSTP = 1 if you delay an operation stop until the total output pulse becomes an integral multiple of a multiplication value.

If you stop the operation before the total output pulse becomes an integral multiple of a multiplication value, set RENV6.PSTP = 0.

However, in the case of interpolation control (68h, 69h, 6Ah, 6Bh, 6Ch, 6Dh), RENV6.PSTP = 1 is ignored and the operation stops.

Name and description	Target
<Input specifications of PA and PB signals> 00b: 90-degree phase difference mode 1x 01b: 90-degree phase difference mode 2x 10b: 90-degree phase difference mode 4x 11b: 2 pulse mode	RENV2.PIM(25:24)
<Counting directions of PA and PB signals> 0: Counts up when the phase of PA signal is advanced. 1: Counts up when the phase of PB signal is advanced.	RENV2.PDIR(26)
< Input functions of PA and PB signals > 0: Enabled. 1: Disabled. Input errors are not detected as well.	RENV2.POFF(31)
< Input noise filter of PE signal > 0: A signal with the pulse width of 0.05 μs or more will react reliably. 1: A signal with the pulse width of 26 ms or less is ignored completely.	RENV1.DRF(27)

Name and description	Target
<Processing when writing a stop command under pulser controls> 0: Stops by ignoring input PA and PB signals. 1: Output the command pulse corresponding to input PA and PB signals, and then stop. For interpolation control (68h, 69h, 6Ah, 6Bh, 6Ch, 6Dh), RENV6.PSTP = 1 is ignored and the operation stops.	RENV6.PSTS(15)
<Divided molecule of PA and PB signal inputs> 0: Not divide. 1 to 2047: Divide the frequency by the set value / 2048.	RENV6.PD(26:16)
<Multiplying value of PA and PB signal inputs> 0 to 31: Multiply by the value obtained by adding 1 to the set value.	RENV6.PMG(31:27)
<Operating status> 1000b: Waiting for PA and PB signal inputs	RSTS.CND(3:0)
<Error interrupt factor (ESPE)> 1: PA or PB signal input error has occurred. The operation mode does not stop.	REST.ESPE(17)
<Error interrupt factor (ESPO)> 1: Stopped abnormally because the buffer counter (16 bit) for inputting PA and PB signals overflowed.	REST.ESPO(14)

5.5.3.1 Continuous movement (01h)

Command control is performed in synchronization with PA and PB signal inputs.

If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND = 1000b), a command pulse will start to be output. The counting direction is determined by PA, PB signals and RENV2.PDIR bit.

RENV2.PIM	RENV2.PDIR	PA, PB signals	Count direction
00b, 01b, 10b: 90-degree phase difference mode	0	PA input phase advances PB.	+direction
		PB input phase advances PA.	-direction
	1	PB input phase advances PA.	+direction
		PA input phase advances PB.	-direction
11b: 2 pulse mode	0	PA signal rising edge	+direction
		PB signal rising edge	-direction
	1	PB signal rising edge	+direction
		PA signal rising edge	-direction

Writing a stop command completes the operation mode.

When counting +direction, the motion stops when +EL signal turns ON.

When counting -direction, movement stops when -EL signal turns ON.

No error interrupt occurs by stopping when +EL signal turns ON as well as -EL signal turns ON.

An operation mode continues even if stopping by turning +EL signal ON or -EL signal ON.

By inputting PA and PB signals in the opposite direction, you can escape from the EL position of the stop factor.

5.5.3.2 Incremental movement (51h)

Positioning control is performed in synchronization with PA and PB signal inputs.

When starting, the RPLS register value is updated with the RMV register absolute value.

If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND = 1000b), a command pulse will start to be output.

The counting direction will be determined in +direction if $RMV > 0$, or in -direction if $RMV < 0$.

PA, PB signals and RENV2.PDIR bit have no effect on the counting direction.

When $RPLS = 0$, the command pulse is stopped.

When the command pulse is stopped, the operation mode is completed.

After completing the operation mode, the command pulse is not output when PA and PB signals are input.

Attempting to start with $RMV = 0$ completes the operation mode without outputting a command pulse.

You can also complete the operation mode by executing a stop command.

5.5.3.3 Specify absolute position by counter 1 (52h)

Positioning control is performed in synchronization with PA and PB signal inputs.

When starting, the RPLS register value is updated with the absolute value of difference between the RCUN1 register value and the RMV register value.

If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND = 1000b), the command pulse will start to be output.

The counting direction is determined in +direction if $RMV > RCUN1$ and in -direction if $RMV < RCUN1$.

PA, PB signals and RENV2.PDIR bit have no effect on the counting direction.

When $RPLS = 0$, the command pulse is stopped.

When the command pulse is stopped, the operation mode is completed.

After completing the operation mode, the command pulse is not output when PA and PB signals are input.

Attempting to start with $RMV = RCUN1$ completes the operation mode without outputting a command pulse.

You can also complete the operation mode by executing a stop command.

5.5.3.4 Specify absolute position by counter 2 (53h)

Same as $RMD.MOD = 52h$, except that RCUN2 register is used instead of RCUN1 register.

5.5.3.5 Zero-point return by counter 1 (54h)

With $RMD.MOD = 52h$, the operation is the same as when $RMV = 0$ is set.

Even in other than $RMV = 0$, the RPLS register value is updated with $RMV = 0$.

5.5.3.6 Zero-point return by counter 2 (55h)

With $RMD.MOD = 53h$, the operation is the same as when $RMV = 0$ is set.

Even in other than $RMV = 0$, the RPLS register value is updated with $RMV = 0$.

5.5.3.7 Continuous movement by linear interpolation 1 (68h)

Linear interpolation 1 control is performed in synchronization with PA and PB signal inputs.

If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND = 1000b), the command pulse will start to be output. For the continuous movement of linear interpolation 1 control, see "5.5.7.1 Continuous movement (60h)".

5.5.3.8 Incremental movement by linear interpolation 1 (69h)

Linear interpolation 1 control is performed in synchronization with PA and PB signal inputs.

If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND = 1000b), the command pulse will start to be output. For the incremental movement of linear interpolation 1 control, see "5.5.7.2 Incremental movement (61h)".

5.5.3.9 Continuous movement by linear interpolation 2 (6Ah)

Linear interpolation 2 control is performed in synchronization with PA and PB signal inputs.

If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND = 1000b), the command pulse will start to be output. For continuous movement of linear interpolation 2 control, see "5.5.8.1 Continuous movement (62h)".

5.5.3.10 Incremental movement by linear interpolation 2 (6Bh)

Linear interpolation 2 control is performed in synchronization with PA and PB signal inputs.

If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND = 1000b), the command pulse will start to be output. For the incremental movement of linear interpolation 2 control, see "5.5.8.2 Incremental movement (63h)".

5.5.3.11 Circular interpolation in CW direction (6Ch)

Circular interpolation control is performed in synchronization with PA and PB signal inputs.

If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND = 1000b), the command pulse will start to be output. For the circular interpolation in CW direction, see "5.5.9.1 Circular interpolation in CW direction (64h)".

5.5.3.12 Circular interpolation in CCW direction (6Dh)

Linear interpolation 1 control is performed in synchronization with PA and PB signal inputs.

If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND = 1000b), the command pulse will start to be output. For the circular interpolation in CCW direction, see "5.5.9.2 Circular interpolation in CCW (65h)".

5.5.4 Switch control

Each operation mode is controlled by the input of +DR and -DR signals as a trigger.

This control can be used when PEn = L level.

PEn pin allows you to switch between multiple axes with a set of external switches.

The connection method is the same as in a pulser control.

For details, see "5.5.3 Pulser control".

The input logic of +DR and -DR signals can be set with RENV1.DRL bit.

With RENV1.DRF bit, you can set an input noise filter for PE signal and +DR, -DR signals.

RIRQ.IRDR bit can generate an interrupt in RIST register when +DR or -DR signal changes.

RIST.ISPD = 1 when +DR signal changes, and RIST.ISMD = 1 when -DR signal changes.

You can check the input status of +DR signal with RSTS.SDRP bit and the input status of -DR signal with RSTS.SDRM bit.

When started, this LSI waits for +DR and -DR signals to be input (RSTS.CND = 0001b).

After that, the command pulse is output in +direction when +DR signal is input and in -direction when -DR signal is input.

Name description	Target
<+DR, -DR signal input logic> 0: Negative logic 1: Positive logic	RENV1.DRL(25)
<input noise filter for +DR, -DR, PE signal > 0: A signal with a pulse width of 0.05 μs or more will react reliably. 1: A signal with a pulse width of 26 ms or less is ignored completely.	RENV1.DRF(27)
<Event interrupt factor (ISPD)> 1: +DR signal has changed.	* 1 RIST.ISPD(17)
<Event interrupt factor (ISMD)> 1: -DR signal has changed.	* 1 RIST.ISMD(18)
<Operating status> 0001b: Waiting for +DR, -DR signal input	RSTS.CND(3: 0)
<+DR signal input status> 0: OFF 1: ON	* 2 RSTS.SDRP(11)
<-DR signal input status> 0: OFF 1: ON	* 2 RSTS.SDRM(12)

* 1 When PEn = H level, no interrupt is generated.

* 2 The status also changes when PEn = H level.

5.5.4.1 Continuous movement (02h)

Command control is performed in conjunction with +DR and -DR signal ONs.

When started, a command pulse is output in +direction while +DR signal is ON and in -direction while -DR signal is ON.

The operation will not be completed only by turning off +DR signal or -DR signal.

While waiting for +DR signal and -DR signal input (RSTS.CND = 0001b), you can control as many times as you like by turning +DR signal ON or -DR signal ON. Writing a stop command completes the operation mode.

+DR and -DR signals are edge triggers.

Even if you start with +DR signal ON or -DR signal ON, it will be ignored.

When +DR signal ON is input, move in +direction, and when +DR signal OFF is input, the movement stops.

While +DR signal is ON, -DR signal input is ignored.

When -DR signal ON is input, move in -direction, and when -DR signal OFF is input, the movement stops.

While -DR signal is ON, +DR signal input is ignored.

If the speed pattern is in high speed 1 or high speed 2, decelerate-stops with +DR signal OFF or -DR signal OFF.

If the reverse direction is turned ON while decelerating with +DR signal OFF or -DR signal OFF, moves in the reverse direction after decelerate-stops.

When counting in +direction, the movement stops when +EL signal is ON.

When counting in -direction, the movement stops when -EL signal is ON.

No error interrupt occurs when +EL signal is turned ON or -EL signal is turned ON.

The operation mode continues even if the operation is stopped by turning +EL signal ON or -EL signal ON.

By inputting +DR signal or -DR signal in the opposite direction, you can escape from EL position of the stop factor.

5.5.4.2 Incremental movement (56h)

Positioning control is performed in conjunction with +DR and -DR signal ONs.

Set the RMV register value from 1 to 2,147,483,647.

When +DR signal or -DR signal is ON while stopped, the RPLS register value is updated with the RMV register value.

When started, a command pulse is output in +direction when +DR signal is ON and in -direction when -DR signal is ON.

When RPLS = 0, the command pulse is stopped, and the operation mode is continued.

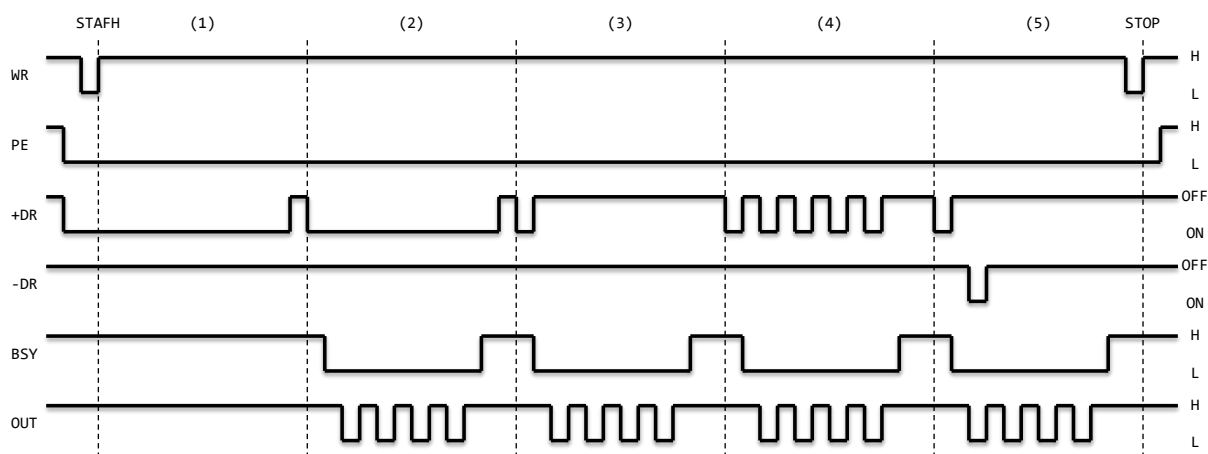
While waiting for +DR signal and -DR signal input (RSTS.CND = 0001b), you can control as many times as you like with +DR signal or -DR signal. Writing a stop command completes the operation mode.

+DR and -DR signals are edge triggers.

Even if you start with +DR signal ON or -DR signal ON, it will be ignored.

Inputs of +DR or -DR signal is enabled only while waiting for the input of +DR signal or -DR signal (RSTS.CND = 0001b).

+DR signal operation example:



- (1) Since they are edge triggers, not operate even if the operation mode is started with +DR signal ON.
- (2) Since in an incremental movement, the command pulse is stopped at RPLS = 0 even if the +DR signal is kept ON.
- (3) Since in an incremental movement, even if +DR signal OFF is input, the command pulse is output until RPLS = 0.
- (4) Since RSTS.CND ≠ 0001b during operation, inputting +DR signal ON does not affect the operation.
- (5) Since RSTS.CND ≠ 0001b during operation, inputting -DR signal ON does not affect the operation.

When counting in +direction, the operation stops when +EL signal is ON.

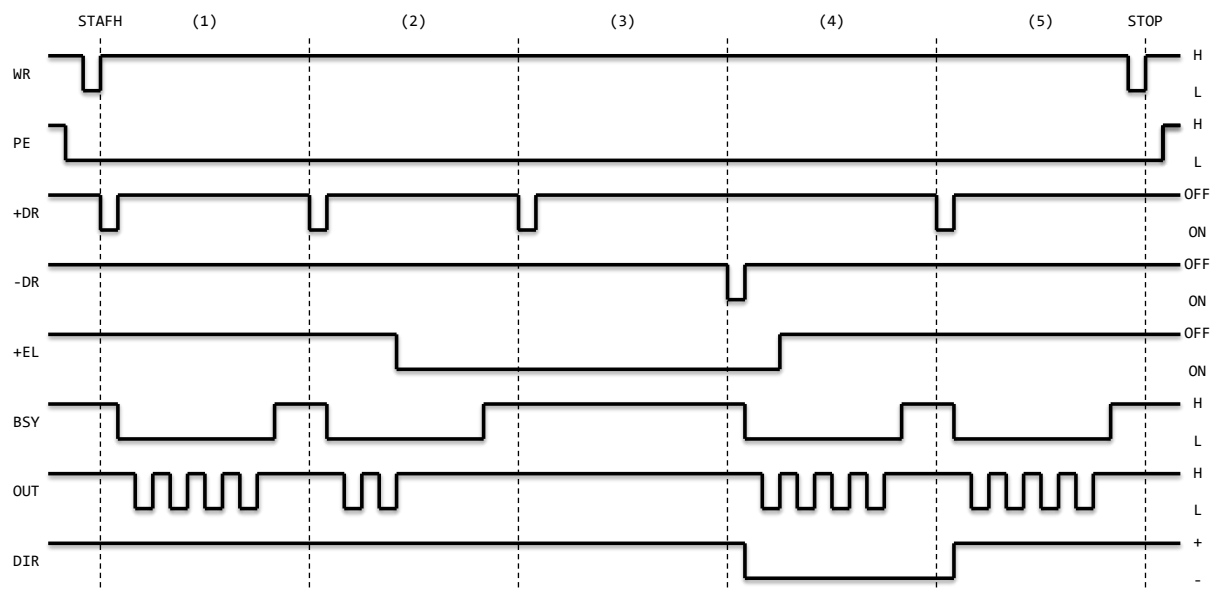
When counting in -direction, the operation stops when -EL signal is ON.

No error interrupt occurs when the operation stops by +EL signal ON or -EL signal ON.

Operation mode is not completed when the operation stops by turning +EL signal ON or -EL signal ON.

While stopped by turning on EL signal in the operating direction, turning on DR signal in the opposite direction is enabled.

+EL signal operation example:



- (1) When the operation mode is started, operate in +direction by +DR signal ON.
- (2) When operating with +DR signal ON, operation stops when +EL signal is ON.
- (3) When +EL signal is ON, not operate if +DR signal is ON.
- (4) While +EL signal is kept ON, operates in -direction when -DR signal is ON.
- (5) When +EL signal is turned OFF, operates in +direction when +DR signal is ON.

5.5.5 Origin return control

This control method is to stop at the origin.

See also “6.8.2 Deviation counter clear (ERC)” to control a servo motor.

You can set the input logic of ORG signal with RENV1.ORGL bit.

RENV1.FLTR bit allows you to set an input noise filter on ORG signal.

You can check the input status of ORG signal with SSTS.SORG bit.

ORG signal is sampled in synchronization with the output of a command pulse.

Therefore, design your system so that the input width of ORG signal exceeds the length that can be moved by the command pulse output of 1 pulse.

The input logic of EZ signal can be set with RENV2.EZL bit.

The number of EZ signal inputs (default down-count) can be set with RENV3.EZD bit.

You can check the down-count value of EZ signal with RSPD.EZC bit.

RENV2.EINF bit allows you to set an input noise filter on EZ signal.

You can check the input status of EZ signal with RSTS.SEZ bit.

In ELLn pin, you can set the input logic for +EL and -EL signals.

RENV1.ELM bit can be used to set the input processing for +EL and -EL signals.

With the RENV1.FLTR bit, you can set an input noise filter for +EL and -EL signals.

You can check the input status of +EL signal with SSTS.SPEL bit.

You can check the input status of -EL signal with SSTS.SMEL bit.

Name and description	Target
<ORG signal input logic> 0: Negative logic 1: Positive logic	RENV1.ORGL(7)
< Input noise filter for +EL, -EL and ORG signals > 0: A signal with the pulse width of 0.05 μs or more will react reliably. 1: A signal with the pulse width of 3 μs or less is ignored completely.	RENV1.FLTR(26)
<ORG signal input status> 0: OFF 1: ON	SSTS.SORG(14)
<EZ signal input logic> 0: Negative logic 1: Positive logic EZ signal changes from OFF to ON to count.	RENV2.EZL(23)
<Initial value of EZ signal input count> 0000b (1 time) to 1111b (16 times)	RENV3.EZD(7:4)

Name and description	Target
<EZ signal input count value> The initial value is the value of RENV3.EZD bit.	RSPD.EZC(19:16)
<Input noise filter for EA, EB and EZ signals> 0: A signal with the pulse width of 0.05 μs or more will react reliably. 1: A signal with the pulse width of 0.15 μs or more will react reliably.	RENV2.EINF(18)
<EZ signal input status> 0: OFF 1: ON	RSTS.SEZ(10)
<+EL and -EL signal input logics> L: Positive logic H: Negative logic	ELLn pin
<Input processing of +EL signal and -EL signal> 0: Stops immediately when the EL signal in the operating direction is ON 1: Decelerate-stops when the EL signal in the operating direction is ON	RENV1.ELM(3)
<+EL signal input status> 0: OFF 1: ON	SSTS.SPEL(12)
<-EL signal input status> 0: OFF 1: ON	SSTS.SMEL(13)

5.5.5.1 Origin return in +direction (10h)

When started, this LIS starts to output command pulses in the +direction.

When the condition for the origin return is satisfied, command pulses are stopped.

When the command pulses are stopped, the operation mode is completed.

Use RENV3.ORM bit to set the origin return method.

Depending on the origin return method, origin return control will be continued without conducting an abnormal stop when +EL signal is turned ON.

Set whether to clear the corresponding counter at the origin by RENV3.CU1R to CU4R bits.

Set whether to output ERC signal when stopped due to the origin return factor with RENV1.EROR bit.

Name and description	Target
<p><Origin return method></p> <p>0000b: Origin return 0</p> <ul style="list-style-type: none"> - When ORG signal is changed from OFF to ON, FL and FH constant speed operations stop immediately. High speed 1 and 2 operations decelerate-stop. - Counter clear timing: When ORG signal is changed from OFF to ON. <p>0001b: Origin return 1</p> <ul style="list-style-type: none"> - When ORG signal is changed from OFF to ON, FL and FH constant speed operations stop immediately. High speed 1 and 2 operations decelerate-stop. Operates in the opposite direction at FA constant speed until ORG signal is changed from ON to OFF. Operates in the initial direction at FA speed and stops immediately when ORG signal is changed from OFF to ON. - Counter clear timing: When ORG signal is changed from OFF to ON after moving at FA speed. <p>0010b: Origin return 2</p> <ul style="list-style-type: none"> - After operating until ORG signal is changed from OFF to ON, FL and FH constant speed operations stop immediately when the specified number of EZ signals is counted up. High speed 1 and 2 operations decelerate. And they stop when the specified number of EZ signals is counted up. - Counter clear timing: When the EZ signals is counted up. <p>0011b: Origin return 3</p> <ul style="list-style-type: none"> - Operates until ORG signal is changed from OFF to ON, FL and FH constant speed operations stop immediately when the specified number of EZ signals is counted up. High speed 1 and 2 operations decelerate-stop when the specified number of EZ signals is counted up. - Counter clear timing: When the specified number of EZ signals is counted up. <p>0100b: Origin return 4</p> <ul style="list-style-type: none"> - When ORG signal is changed from OFF to ON, FL and FH constant speed operations stop immediately. High speed 1 and 2 operations decelerate-stop. Operates in the opposite direction at FA constant speed until ORG signal is changed from ON to OFF, and stops immediately when the specified number of EZ signals is counted up. - Counter clear timing: When the specified number of EZ signals is counted up. <p>0101b: Origin return 5</p> <ul style="list-style-type: none"> - When ORG signal is changed from OFF to ON, FL and FH constant speed operations stop immediately. High speed 1 and 2 operations decelerate-stop. Operates in the opposite direction until ORG signal is changed from ON to OFF, FL and FH constant speed operations stops immediately, and High speed 1 and 2 operations decelerate-stop when the specified number of EZ signals is counted up. - Counter clear timing: When the specified number of EZ signals is counted up. 	<p>RENV3.ORM(3:0)</p>

Name and description	Target
<p>0110b: Origin return 6</p> <ul style="list-style-type: none"> - Stops immediately when EL signal is changed from OFF and ON, and High speed 1 and 2 operations decelerate-stop when RENV1.ELM = 1. After operating in the opposite direction at FA constant speed until EL signal is changed from ON to OFF, then stops immediately. - Counter clear timing: When EL signal is changed from ON to OFF. <p>0111b: Origin return 7</p> <ul style="list-style-type: none"> - Stops immediately when EL signal is changed from OFF and ON, and decelerate-stops when RENV1.ELM = 1 in high speeds 1 and 2 operations. Operates in the opposite direction at FA constant speed until EL signal is changed from ON to OFF, and stops immediately when the specified number of EZ signals is counted up. - Counter clear timing: When stopped immediately after the specified number of EZ signals is counted up. <p>1000b: Origin return 8</p> <ul style="list-style-type: none"> - Stops immediately when EL signal is changed from OFF to ON, and decelerate-stops when RENV1.ELM = 1 in high speed 1 and 2 operations. Operates in the opposite direction at FA constant speed until EL signal is changed from ON to OFF, and stops immediately when the specified number of EZ signals is counted up. Decelerate-stops in high speeds 1 and 2 operations. - Counter clear timing: When the specified number of EZ signals is counted up. <p>1001b: Origin return 9</p> <ul style="list-style-type: none"> - After the origin return 0 operation, returns to zero point (operates until RCUN2 = 0). <p>1010b: Origin return 10</p> <ul style="list-style-type: none"> - After the origin return 3 operation, returns to zero point (operates until RCUN2 = 0). <p>1011b: Origin return 11</p> <ul style="list-style-type: none"> - After the origin return 5 operation, returns to zero point (operates until RCUN2 = 0). <p>1100b: Origin return 12</p> <ul style="list-style-type: none"> - After the origin return 8 operation, returns to zero point (operates until RCUN2 = 0). 	
<p><Clears counter 1 when the origin is reached by origin return control ></p> <p>0: Not clear.</p> <p>1: Clear.</p>	RENV3.CU1R(20)
<p><Clears counter 2 when the origin is reached by origin return control ></p> <p>0: Not clear.</p> <p>1: Clear.</p>	RENV3.CU2R(21)
<p><Clear counter 3 when the origin is reached by origin return control ></p> <p>0: Not clear.</p> <p>1: Clear.</p>	RENV3.CU3R(22)
<p><Clear counter 4 when the origin is reached by origin return control ></p> <p>0: Not clear.</p> <p>1: Clear.</p>	RENV3.CU4R(23)
<p><Output function of ERCn pin when stopped due to origin return factor></p> <p>0: ERC signal is not output when stopped by the origin return factor.</p> <p>1: ERC signal is output when stopped by the origin return factor.</p> <p>See "6.8.2 Deviation counter clear (ERC).</p>	RENV1.EROR(11)

5.5.5.1.1 Origin return 0 (0000b)

Sets the position where ORG signal turns ON from OFF as the origin.

In FL or FH constant speed pattern, stops at the origin position.

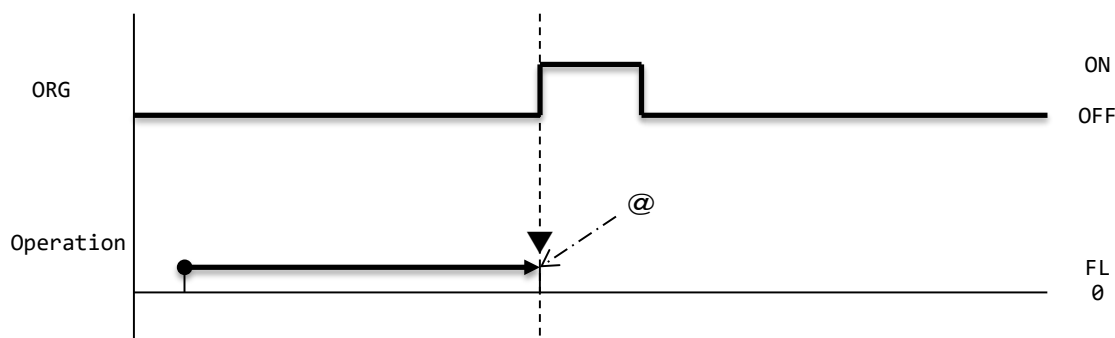
In high speed 1 or 2 speed pattern, stops after passing the origin position.

Example: STAF_L (50h) command

Operates in the +direction with FL constant speed pattern.

Stops immediately when ORG signal is changed from OFF to ON

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

@: ERC signal output timing when RENV1.EROR = 1 is set.

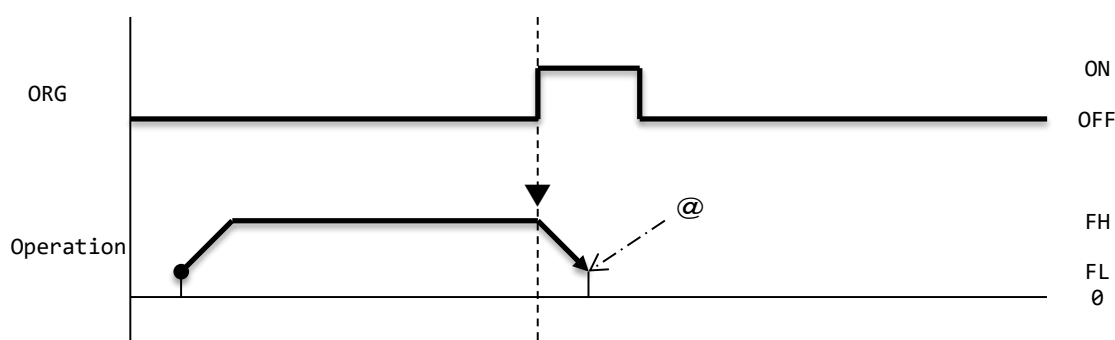
Use them when controlling a servo motor.

Example: STAU_D (53h) command

Operates in the +direction with high speed 2 speed pattern.

Decelerates and stops when ORG signal is changed from OFF to ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

If you clear the mechanical position counter, you can count the feed amount from the origin position.

You can move to the origin position by using RMD.MOD = 45h (return to Zero point with counter 2).

@: ERC signal output timing when RENV1.EROR = 1 is set.

It is not required, but can be used to control a servo motor.

5.5.5.1.2 Origin return 1 (0001b)

Sets the position where ORG signal turns ON from OFF as the origin.

Stops at the origin position.

Example: STAUD (53h) command

Operates in the +direction with high speed 2 speed pattern.

Decelerate-stops when ORG signal is changed from OFF to ON.

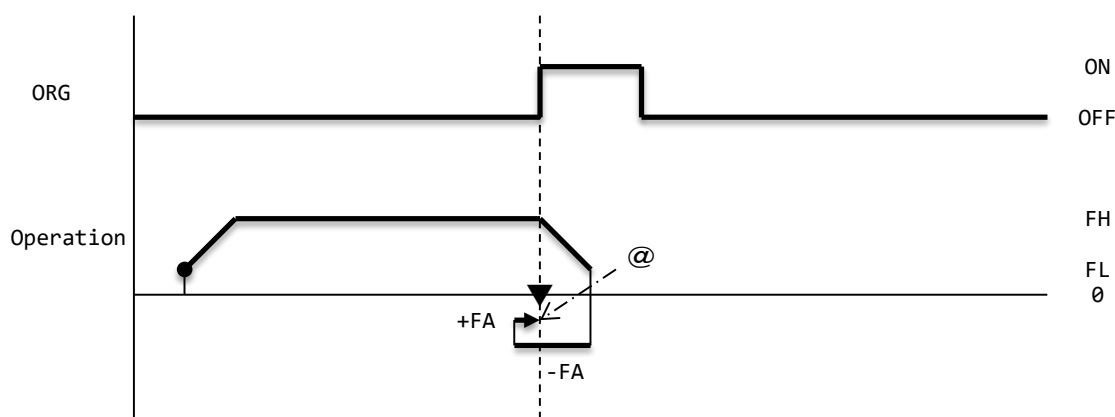
Operates in the -direction with FA constant speed.

Stops immediately when ORG signal is changed from ON to OFF.

Operates in the +direction with FA constant speed.

Stops immediately when ORG signal is changed from OFF to ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

@: ERC signal output timing when RENV1.EROR = 1 is set.

5.5.5.1.3 Origin return 2 (0010b)

Sets the position where EZ signal is turned ON for the specified number of times as the origin after ORG signal turns ON from OFF.

Stops at the origin position.

Example: STAUD (53h) command (RENV3.EZD = 0001b)

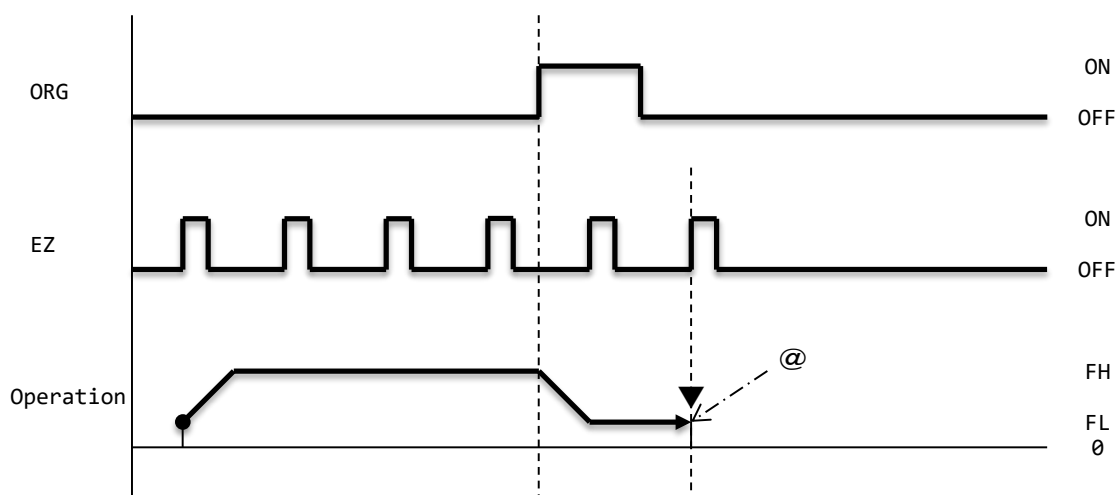
Operates in the +direction with high speed 2 speed pattern.

Decelerates when ORG signal is changed from OFF to ON.

Counts the number of EZ signals changed from OFF to ON, and stops immediately when the specified number of EZ signals is ON.

Completes the operation mode.

Even during decelerating after ORG signal turns ON, the operation stops immediately when the specified number of EZ signals is ON. For the number of times EZ signal ON, set the number to stop after decelerating after ORG signal ON.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

@: ERC signal output timing when RENV1.EROR = 1 is set.

5.5.5.1.4 Origin return 3 (0011b)

Sets the position where the specified number of EZ signal is turned ON as the origin after ORG signal turns ON from OFF.

Passes through the origin position and stops in high speed 1 and 2 speed patterns.

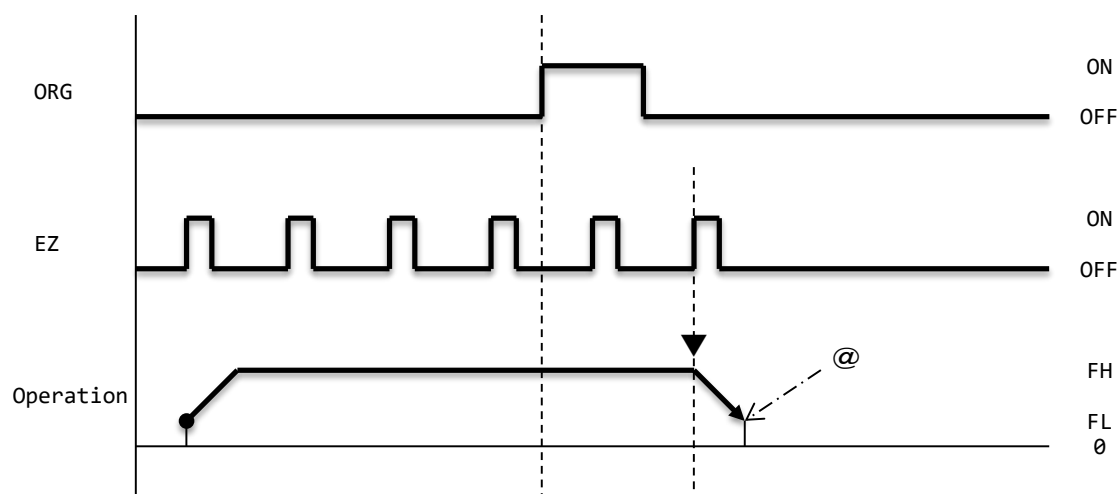
Example: STAUD (53h) command (RENV3.EZD = 0001b)

Operates in the +direction with high speed 2 speed pattern.

Does not decelerate when ORG signal is changed from OFF to ON.

Counts the number the EZ signal changed from OFF to ON and decelerate-stops when the specified number of EZ signals is turned ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

@: ERC signal output timing when RENV1.EROR = 1 is set.

5.5.5.1.5 Origin return 4 (0100b)

After stopping when ORG signal is changed from OFF to ON, reverses, and sets the position where the specified number of EZ signal is turned ON as the origin.

Stops at the origin position.

Example: STAUD (53h) command (RENV3.EZD = 0001b)

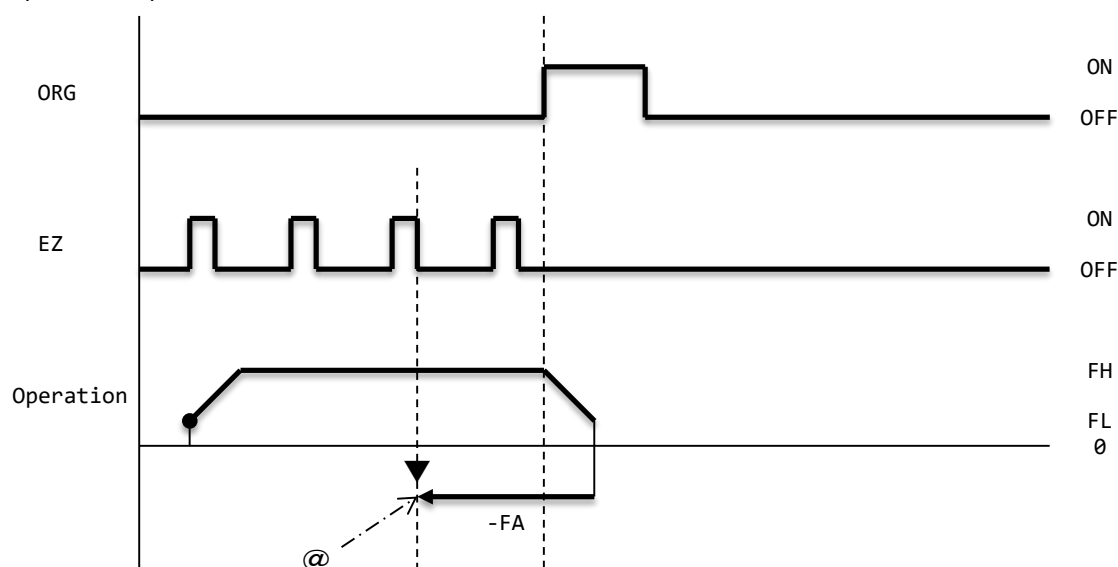
Operates in the +direction with high speed 2 speed pattern.

Decelerate-stops when ORG signal is changed from OFF to ON.

Operates in the -direction with FA constant speed pattern.

Counts the number of EZ signal changed from OFF to ON, and stops immediately when the specified number of EZ signals is turned ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

ⓐ: ERC signal output timing when RENV1.EROR = 1 is set.

5.5.5.1.6 Origin return 5 (0101b)

After stopping when ORG signal is changed from OFF to ON, reverses, and sets the position where the specified number of EZ signal is turned ON as the origin.

Passes through the origin position and stops in high speed 1 and 2 speed patterns.

Example: STAUD (53h) command (RENV3.EZD = 0001b)

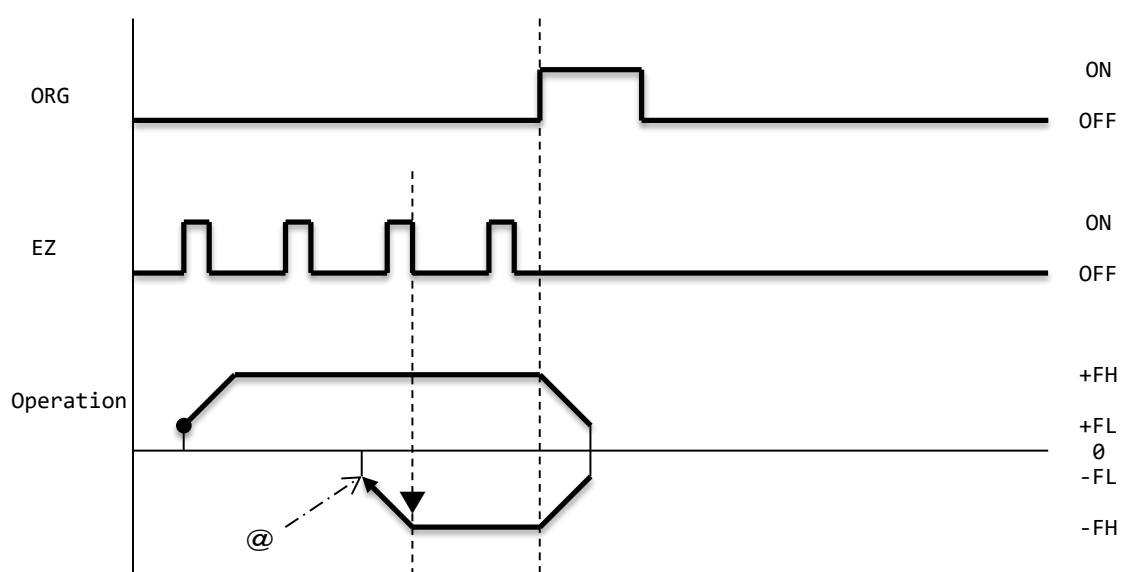
Operates in the +direction with high speed 2 speed pattern.

Decelerate-stops when ORG signal is changed from OFF to ON.

Operates in the -direction with high speed 2 speed pattern.

Counts the number the EZ signal is changed from OFF to ON, and decelerate-stops when the specified number of EZ signals is turned ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

@: ERC signal output timing when RENV1.EROR = 1 is set.

5.5.5.1.7 Origin return 6 (0110b)

Sets the position where EL signal is changed from OFF to ON as the origin.

Stops at the origin position.

Example: STAUD (53h) command

Operates in the +direction with high speed 2 speed pattern.

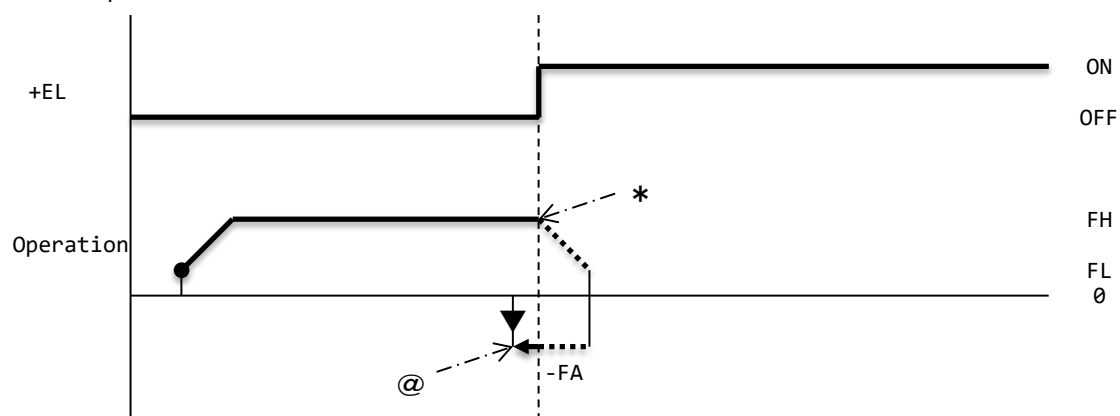
Decelerate-stops when EL signal is changed from OFF to ON.

Even if +EL signal turns ON, the motor does not stop abnormally, nor an error interrupt occur.

Operates in the -direction with FA speed operation.

Stops immediately when EL signal is changed from ON to OFF.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

*: ERC signal output timing when RENV1.EROR = 1 and RENV1.ELM = 0 are set.

Stop immediately without decelerating and return when +EL signal is changed from OFF to ON.

@: ERC signal output timing when RENV1.EROR = 1 is set

5.5.5.1.8 Origin return 7 (0111b)

After stopping when +EL signal is changed from OFF to ON, reverses and operates at FA speed. Then, sets the position where the specified number of EZ signal is changed ON as the origin.

Stops at the origin position.

Example: STAUD (53h) command (RENV3.EZD = 0001b)

Operates in the +direction with high speed 2 speed pattern.

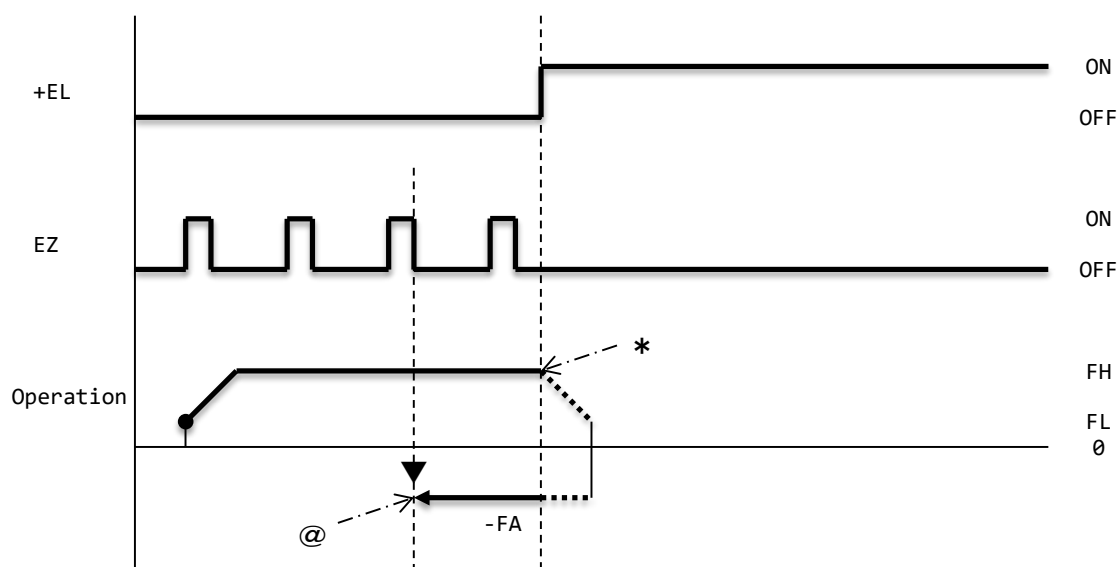
Decelerate-stops when +EL signal is changed from OFF to ON.

Even if +EL signal turns ON, the motor does stop abnormally, nor an error interrupt occurs.

Operates in the -direction with FA constant speed pattern.

Counts the number that EZ signal is changed from OFF to ON and stops immediately when the specified number of EZ signals turn ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

*: ERC signal output timing when RENV1.EROE = 1 and RENV1.ELM = 0 are set.

When +EL signal is changed from OFF to ON, stops immediately instead of decelerating.

@: ERC signal output timing when RENV1.EROR = 1 is set.

5.5.5.1.9 Origin return 8 (1000b)

After stopping when +EL signal is changed from OFF to ON, reverses and operates at the speed pattern selected by a start command, and then sets the position where the specified number of EZ signals turn ON as the origin.

Passes through the origin position and stops in high speed 1 and 2 speed patterns.

Example: STAUD (53h) command (RENV3.EZD = 0001b)

Operates in the +direction with high speed 2 speed pattern.

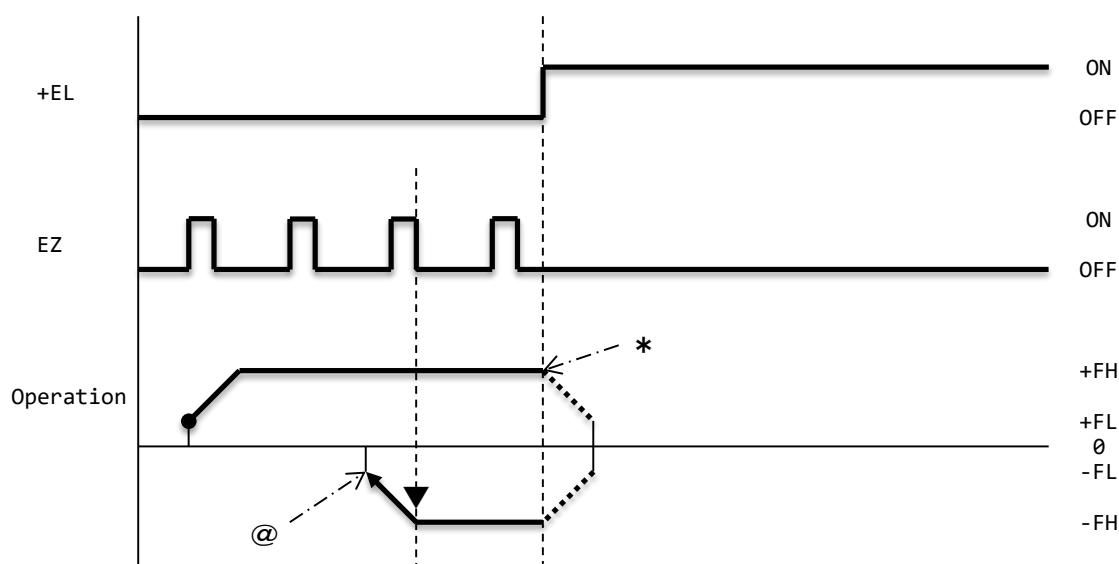
Decelerates and stops when +EL signal is changed from OFF to ON.

Even if +EL signal turns ON, the motor does not stop abnormally, nor an error interrupt occurs.

Operates in the -direction with high speed 2 speed pattern.

Counts the number that EZ signal is changed from OFF to ON, decelerate-stops when the specified number of EZ signals turn ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

*: ERC signal output timing when RENV1.EROE = 1 and RENV1.ELM = 0 are set.

When the +EL signal is changed from OFF to ON, stops immediately instead of decelerating.

@: ERC signal output timing when RENV1.EROR = 1 is set.

5.5.5.1.10 Origin return 9 (1001b)

Sets the position where ORG signal is changed from OFF to ON as the origin.

After the origin return 0 operation, returns to zero point (operate until RCUN2 = 0).

Stops at the origin position.

Set the encoder as the count target of counter 2.

Example: STAUD (53h) command (RENV3.CU2R = 1)

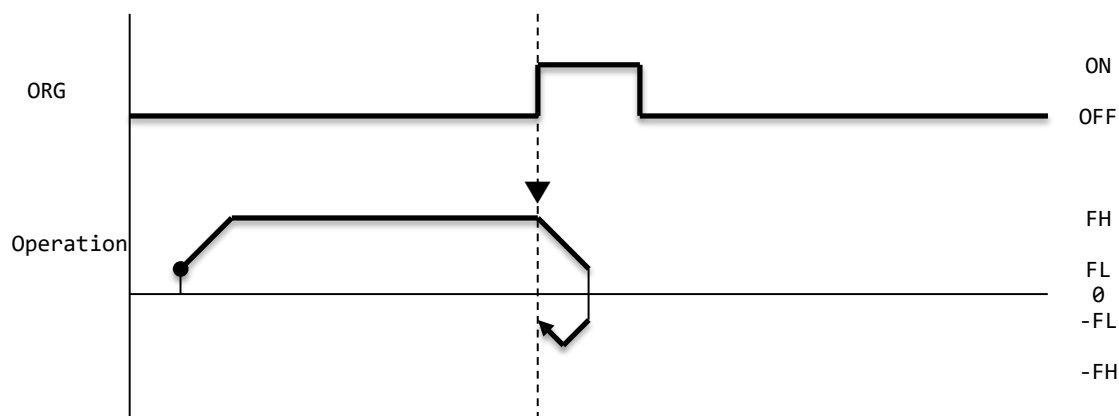
Operates in the +direction with high speed 2 speed pattern.

Decelerates and stops when ORG signal is changed from OFF to ON.

In the operation mode of returning to zero point with counter 2 in a positioning control, operates in the -direction with high speed 2 speed pattern.

Stops at the position where ORG signal is changed from OFF to ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

5.5.5.1.11 Origin return 10 (1010b)

After stopping when ORG signal is changed from OFF to ON, sets the position where the specified number of EZ signal is turns ON as the origin.

After an origin return 3 operation, returns to zero point (operates until RCUN2 = 0).

Stops at the origin position.

Set the encoder for the count target of counter 2.

Example: STAUD (53h) command (RENV3.EZD = 0001b)

Operates in the +direction with high speed 2 speed pattern.

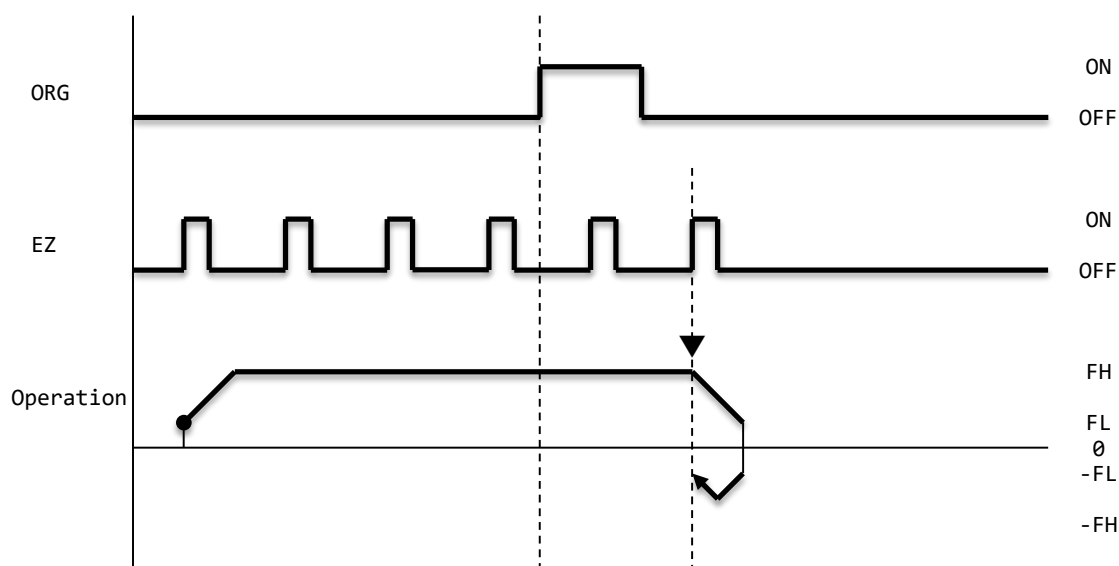
Does not decelerate when ORG signal is changed from OFF to ON.

Counts the number that EZ signal is changed from OFF to ON, and decelerate-stops when the specified number of EZ signal turns ON.

In the operation mode of returning to zero point with counter 2 in a positioning control, operates in the -direction with high speed 2 speed pattern.

Stops at the position where the specified number of EZ signals turn ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

5.5.5.1.12 Origin return 11 (1011b)

After stopping when ORG signal is changed from OFF to ON, reverses, and sets the position where the specified number of EZ signal turns ON as the origin.

After an origin return 5 operation, returns to zero point (operates until RCUN2 = 0).

Stops at the origin position.

Set the encoder for the count target of counter 2.

Example: STAUD (53h) command (RENV3.EZD = 0001b)

Operates in the +direction with high speed 2 speed pattern.

Decelerate-stops when the ORG signal is changed from OFF to ON.

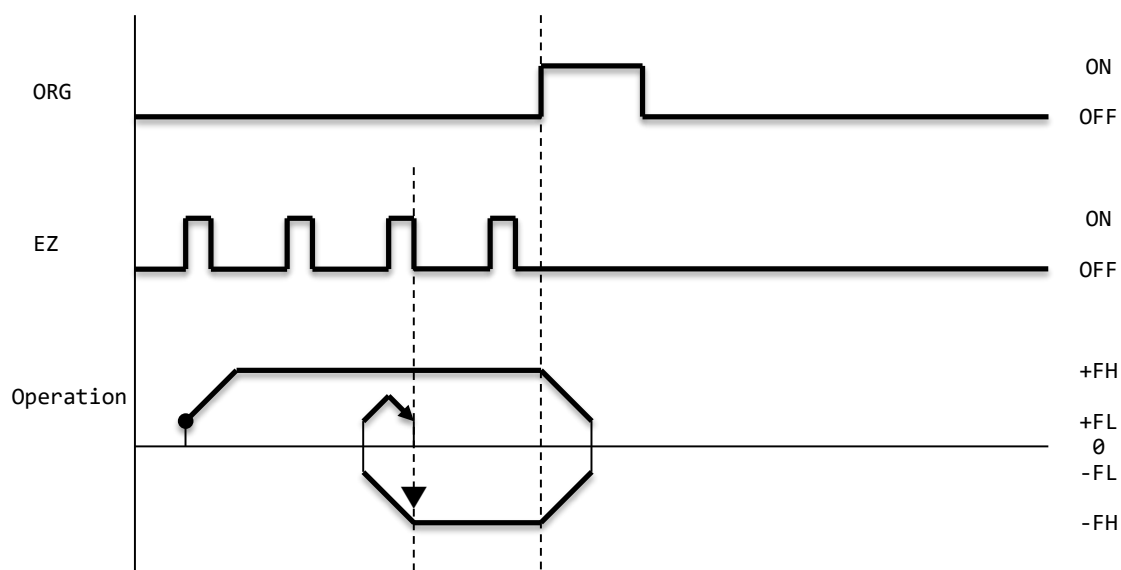
Operates in a high speed 2 speed pattern in -direction.

Counts the number that EZ signal is changed from OFF to ON, decelerate-stops when the specified number of EZ signals turn ON.

In the operation mode of returning to zero point with counter 2 in a positioning control, operates in the +direction with high speed 2 speed pattern.

Stops at the position where the specified number of EZ signals turn ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

5.5.5.1.13 Origin return 12 (1100b)

After stopping when +EL signal is changed from OFF to ON, reverses, and sets the position where the specified number of EZ signal turns ON as the origin.

After the origin return 8 operation, returns to zero point (operates until RCUN2 = 0).

Stops at the origin position.

Sets the encoder for the count target of counter 2.

Example: STAUD (53h) command (RENV3.EZD = 0001b)

Operates in the +direction with high speed 2 speed pattern.

Decelerates and stops when +EL signal is changed from OFF to ON.

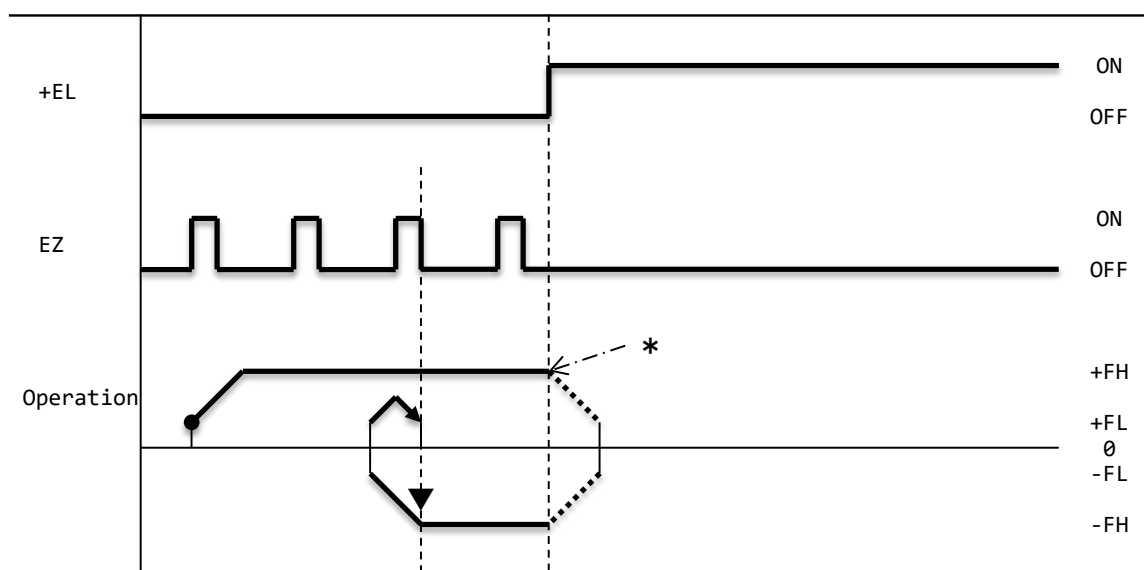
Even if +EL signal turns ON, the motor does not stop abnormally, nor an error interrupt occurs.

Operates in the -direction with high speed 2 speed pattern.

Counts the number that EZ signal is changed from OFF to ON, stops immediately when the specified number of EZ signals turns ON.

Stops at the position where the specified number of EZ signal turns ON.

Completes the operation mode.



▼: Counter clear timing when RENV3.CUnR = 1 (n = 1, 2, 3, 4) is set.

*: ERC signal output timing when RENV1.EROE = 1 and RENV1.ELM = 0 are set.

When the +EL signal is changed from OFF to ON, stops immediately instead of decelerating.

5.5.5.2 Origin return in the –direction (18h)

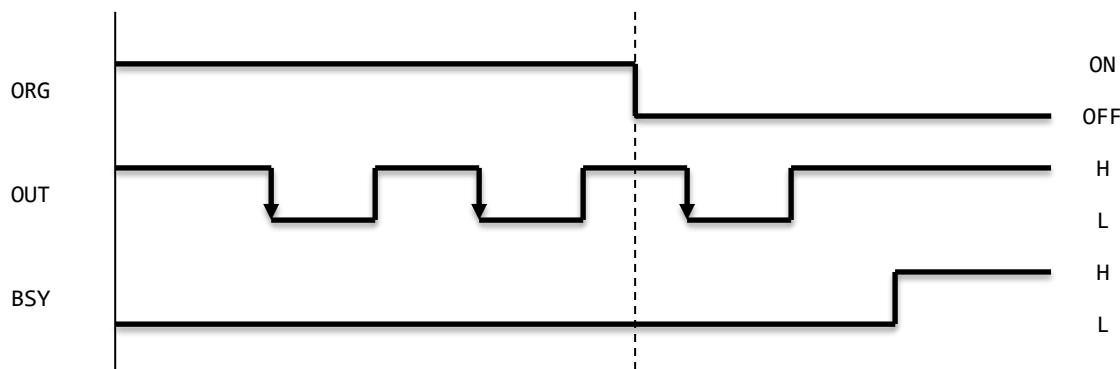
Operates in the same way as "5.5.5.1 Origin return in +direction (10h)" except that the direction and signal are reversed.

5.5.5.3 Escape from the origin position in the +direction (12h)

When started, operates in the +direction until escaping from ORG signal ON.

Use FL constant speed or FH constant speed for the speed pattern.

If starts with ORG signal ON, outputs one pulse after ORG signal turns OFF, and completes the operation mode.



If you start with ORG signal OFF, the operation mode is completed without outputting command pulses.

5.5.5.4 Escape from the origin position in the –direction (1Ah)

Operates in the same way as "5.5.5.3 +direction exit (12h)" except that the direction and signal are reversed.

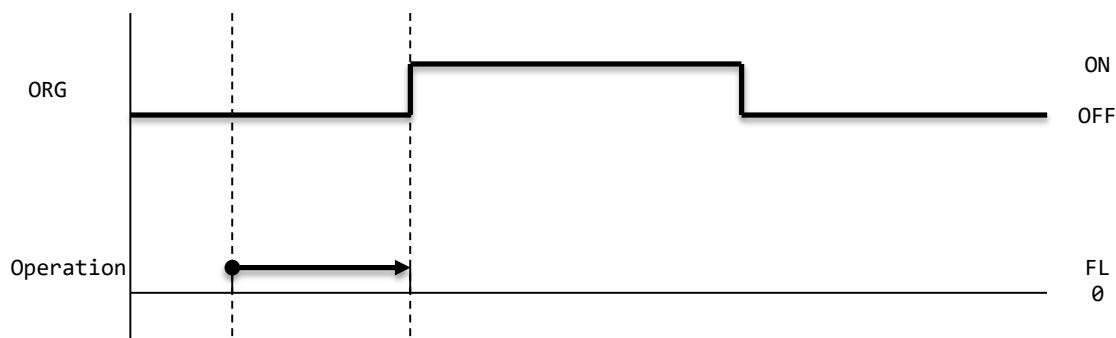
5.5.5.5 Origin search in the +direction (15h)

Set 1 to 2,147,483,647 in RMV register.

When started, operates in one of the following ways, depending on the conditions:

1. When ORG signal is OFF, "return to origin in the +direction (RENV3.ORM)".

When ORG signal turns ON, the operation mode is completed.

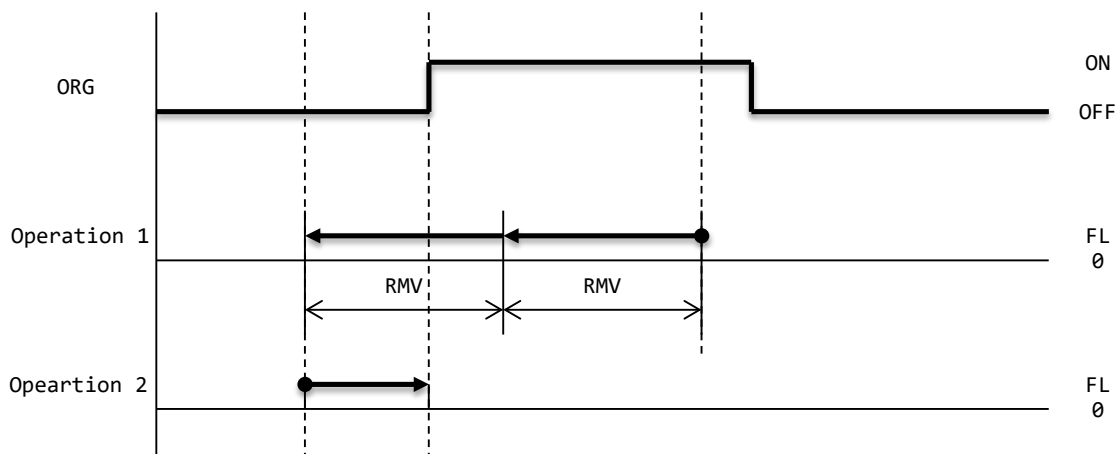


2. When ORG signal is ON, performs "incremental movement" in the –direction.

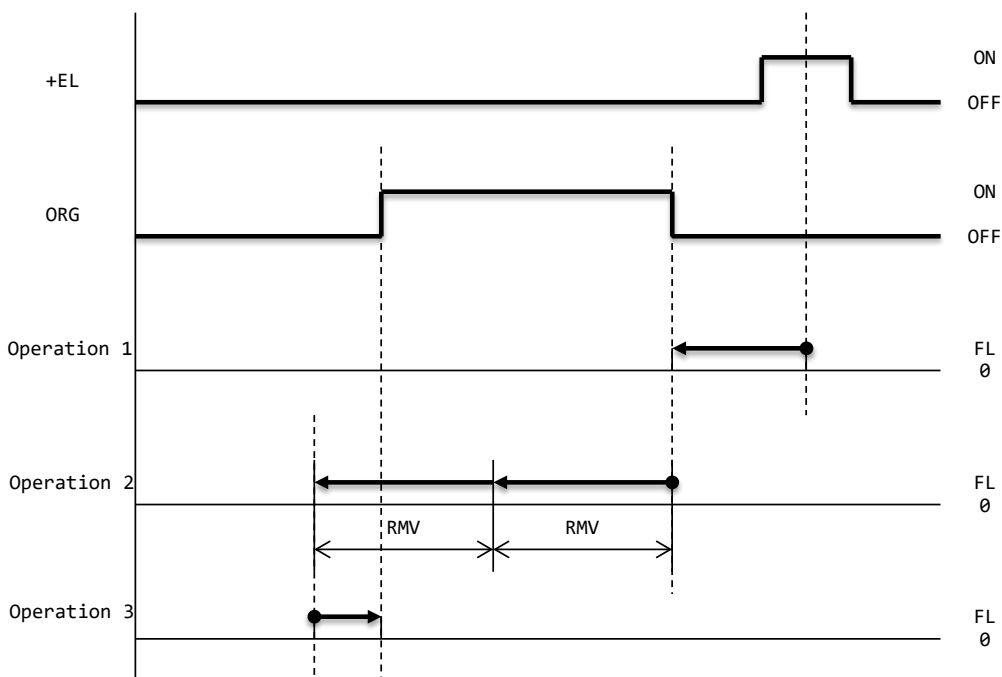
"Incremental movement" in the –direction repeats until ORG signal turns OFF.

After passing from ON to OFF of ORG signal, performs "return to origin in the +direction".

When ORG signal turns, the operation mode is completed.



3. When +EL signal is ON, performs "return to origin (origin return 0) in the -direction".
 When ORG signal is ON, performs "incremental movement" in the -direction.
 "Incremental movement" in the -direction repeats until ORG signal turns OFF.
 After passing from ON to OFF of ORG signal, performs "return to origin (RENV3.ORM) in the +direction".
 When ORG signal turns ON, the operation mode is completed.



The above explanation is for origin return 0 (RENV3.ORM = 0000b) and STAF (51h) command.

In other cases, operates according to the setting and the speed pattern.

5.5.5.6 Origin search in the -direction (1Dh)

Operates in the same way as "5.5.5.5 +direction origin search (15h)" except that the direction and signal are reversed.

5.5.6 Sensor control

This control is to stop by +EL signal, -EL signal, +SL position, -SL position, or EZ signal.

You can:

set the input logic of +EL and -EL signals by ELLn pin.

set the processes by +EL and -EL signal inputs by RENV1.ELM bit.

set an input noise filter in +EL and -EL signals by RENV1.FLTR bit.

check the status of +EL signal input by SSTS.SPEL bit.

check the status of -EL signal input by SSTS.SMEL bit.

set the logic of EZ signal input by RENV2.EZL bit.

set the number of times that EZ signal turns ON (initial down-count value) in RENV3.EZD bit.

check the down-count value of EZ signals by RSPD.EZC bit.

set an input noise filter in EZ signal by RENV2.EINF bit.

check the status of EZ signal input by RSTS.SEZ bit.

For +SL and -SL positions, see "6.13.2 Software limit "

Name and description	Target
<+EL, -EL signal input logic> L: Positive logic H: Negative logic	ELLn pin
<Processing of +EL signal and -EL signal Inputs > 0: Stops immediately when the EL signal in the operating direction turns ON. 1: Decelerate-stops when the EL signal in the operating direction turns ON.	RENV1.ELM(3)
<Input noise filter for +EL, -EL or ORG signal> 0: Signals with the pulse width of 0.05 μs or more will react reliably. 1: Signals with the pulse width of 3 μs or less will be ignored completely.	RENV1.FLTR(26)
<+EL signal input status> 0: OFF 1: ON	SSTS.SPEL(12)
<- EL signal input status> 0: OFF 1: ON	SSTS.SMEL(13)
<EZ signal input logic> 0: Negative logic 1: Positive logic EZ signals are counted when changed from OFF to ON.	RENV2.EZL(23)
<Initial count value of EZ signal input> 0000b (1 time) to 1111b (16 times)	RENV3.EZD(7:4)

Name and description	Target
<Count value of EZ signal input> The initial value is the value of RENV3.EZD bit.	RSPD.EZC(19:16)
<Noise filter for EA, EB, EZ signal inputs> 0: Signals with the pulse width of 0.05 μs or more will react reliably. 1: Signals with the pulse width of 0.15 μs or more will react reliably.	RENV2.EINF(18)
<EZ signal input status> 0: OFF 1: ON	RSTS.SEZ(10)

5.5.6.1 Move to +EL or +SL (20h)

When started, outputs the command pulses in the +direction.

When +EL signal ON or +SL condition is satisfied, the command pulse output stops.

When the command pulse output stops, the operation mode is completed.

No error interrupt occurs when the operation stops by turning +EL signal and -EL signal ON.

No error interrupt occurs even if the operation stops by turning +SL position ON and -SL position ON.

Because they are not abnormal stops, the continuous operation by the pre-register will not be canceled.

If you attempt to start while +EL signal is ON or +SL condition is satisfied, the operation mode is completed without outputting the command pulses.

5.5.6.2 Move to -EL or -SL (28h)

Operates in the same way as "5.5.6.1 Move to +EL or +SL (20h)" except that the direction, signal and conditions are reversed.

5.5.6.3 Escape from -EL or -SL (22h)

When started, operates in the +direction until -EL signal turns ON and -SL condition becomes un-satisfied.

When -EL signal turns ON and -SL condition becomes un-satisfied, command pulses are stopped.

When command pulses are stopped, the operation mode is completed.

Stops abnormally when EL signal in the operating direction is ON.

Stops abnormally even if SL position in the operating direction is ON.

If you attempt to start while -EL signal is OFF and -SL condition is un-satisfied, the operation mode is completed without outputting the command pulses.

5.5.6.4 Escape from +EL or +SL (2Ah)

Operates in the same way as "5.5.6.3 -EL or -SL exit (22h)" except that the direction, signal and conditions are reversed.

5.5.6.5 Move in the +direction for a specified number of EZ counts (24h)

When started, operates in the +direction until EZ signal turns ON for the specified number of times.

Use FL constant speed or FH constant speed for the speed pattern.

When EZ signal turns ON for the specified number of times, the command pulse is stopped.

When the command pulse is stopped, the operation mode is completed.

5.5.6.6 Move in the -direction for a specified number of EZ counts (2Ch)

Operates in the same way as "5.5.6.5 Move in the +direction for a specified number of EZ counts (24h) " except that the direction is reversed.

5.5.7 Linear interpolation 1 control

This control is to use a PCL6046 LSI to perform a linear interpolation operation by any two to four axes.

The remaining axes can perform other operations than the linear interpolation 1 control.

The speed is set on the interpolation control axis.

The interpolation control axes are determined in the order of X, Y, Z-axis among the interpolation axes.

No.	Interpolation operation	Interpolation control axis
1.	Linear interpolation 1 (X, Y, Z, U axes)	X-axis
2.	Linear interpolation 1 (Y, Z, U axes)	Y-axis
3.	Linear interpolation 1(Z, U axes)	Z axis

Set the same speed magnification value (RMG) to all interpolation axes.

Acceleration, deceleration, and constant synthesized speed control can be used.

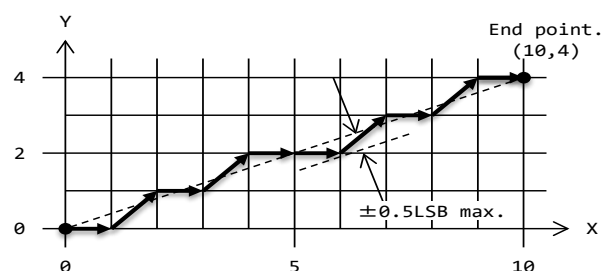
For synthesized speed constant control, see "6.3.6 Synthesized speed constant control"

The interpolation status can be checked with RIPS register, which is common to all axes. You can write start and stop commands to the all interpolation axes after setting from SELx bit to SELu bit of the selected axes.

Linear interpolation accuracy:

Linear interpolation draws a straight line from the current to the end coordinates. The figure on the right is an example of drawing a straight line to the end point coordinates (10, 4).

The position accuracy against the specified straight line in a linear interpolation is ± 0.5 LSB within the entire interpolation. LSB is the smallest unit of RMV register and is the interval between the squares. It corresponds to the resolution of a mechanical system.



5.5.7.1 Continuous movement (60h)

When started, outputs the command pulses in the +direction if RMV > 0 and in the -direction if RMV < 0.

If RMD.MOD = 60h is set only to one axis, REST.ESDT = 1 is set, and an operation stops without outputting a command pulse.

Writing a stop command stops the command pulse.

When the command pulse is stopped, the operation mode is completed.

The following is an example of setting to move the X-axis and Y-axis continuously at a ratio of 5:2.

Register	X-axis	Y-axis
RMD.MOD	60h	60h
RMV	5	2
RFL	5	0
RFH	50000	0

Register	X-axis	Y-axis
RUR	29	0
RDR	0	0
RMG	149	149
Interpolation control axis	○	—

The speed settings to other axes than the interpolation control axes do not affect the operation even if they are set.

Set the interpolation ratio for the absolute value of RMV register.

Even if the number of pulses set in RMV register is output, the operation mode is not completed.

If RMV = 0 is set for all interpolation axes, the speeds of all interpolation axes will be equal.

5.5.7.2 Incremental movement (61h)

When started, the command pulse starts to be output in the +direction if RMV > 0 and in the -direction if RMV < 0.

If RMD.MOD = 61h is set only to one axis, REST.ESDT = 1 is set and operation stops without outputting a command pulse.

Linear interpolation 1 control can accelerate or decelerate when the speed pattern is set in high speed 1 or high speed 2.

Set the same value for all interpolation axes in RMD.MSDP bit and RMD.MADJ bit.

If RMD.MSDP = 1 and RDP > 0, deceleration starts when the main axis becomes RPLS < RDP.

If RMD.MSDP = 0 and RMD.MADJ = 1, deceleration starts when the main axis becomes RPLS < RSDC.

The main axis has the maximum RMV register absolute value among the interpolation axes.

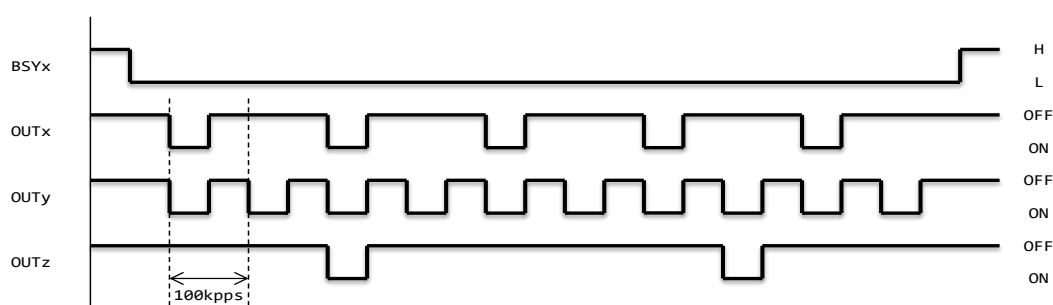
When the main axis reaches RPLS = 0, the command pulse is stopped.

When the command pulse is stopped, the operation mode is completed.

If you attempt to start the main axis with RMV = 0, the operation mode is completed without outputting the command pulse.

The following is a setting example and a command pulse output example that move X-axis, Y-axis, and Z-axis incremental to each other.

Register	X-axis	Y-axis	Z-axis
RMD.MOD	61h	61h	61h
RMD.MIPF	0	0	0
RMV	5	10	2
RFH	50000	0	0
RMG	149	149	149
Interpolation control axis	○	—	—
Main axis	—	○	—



For the timing of BSY signal and OUT signal, see "7.5 Operation timing".

5.5.8 Linear interpolation 2 control

This control is to use one or more PCL6046 and performs linear interpolation on any one or more axes.

With multiple PCL6046s, you can perform linear interpolation on any five or more axes for synchronous operation with equal operating time. The remaining axes can perform operations other than linear interpolation 2 control.

All interpolation axes are interpolation control axes.

Set the speed of the main axis for all interpolation axes.

For speed magnification (RMG), set the same value for all interpolation axes.

Acceleration, deceleration, and constant synthesized speed control cannot be used.

The interpolation status can be checked by RIPS register, which is common to all axes.

You can write start and stop commands to all interpolation axes after setting from SELx bit to SELu bit of the axis selection.

When using multiple PCL6046s, perform simultaneous start with CSTA pin and simultaneous stop with CSTP pin.

For simultaneous start, see "6.9.1 Simultaneous start (CSTA)"

For simultaneous stop, see "6.10 External stop / simultaneous stop".

Set RMD.MSPE = 1 and RMD.MSPO = 1 to all interpolation axes in the case that an abnormal stop occurs.

5.5.8.1 Continuous movement (62h)

When started, the command pulse starts to be output in the +direction if RMV > 0 and in the -direction if RMV < 0.

Among the interpolation axes, the axis with the maximum RMV register absolute value is the main axis.

Writing a stop command stops the command pulse.

When the command pulse is stopped, the operation mode is completed.

The following is an example to use two PCL6046: PCL6046_a and PCL6046_b. The X-axis, Y-axis, and Z-axis of PCL6046_a are continuously moved at 8:5:2:10 ratio respectively with the x-axis of PCL6046_b.

Register	PCL6046_a			PCL6046_b
	Xa axis	Ya axis	Za axis	Xb axis
RMD.MOD	63h	62h	62h	62h
RMD.MSY	01b	01b	01b	01b
RMV	8	5	2	10
RIP	10	10	10	10
RFH	50000	50000	50000	50000
RMG	149	149	149	149
Interpolation control axis	○	○	○	○
Main axis	—	—	—	○

Set the interpolation ratio to the RMV register value.

Set the RMV register value of the main axis to the RIP registers of all interpolation axes.

If RIP = 0 is set to the interpolation axis, REST.ESDT = 1 is set and the operation stops without outputting command pulses.

5.5.8.2 Incremental movement (63h)

When started, the command pulse starts to be output in the +direction if RMV > 0 and in the -direction if RMV < 0.

Among the interpolation axes, the axis with the maximum RMV register absolute value is the main axis.

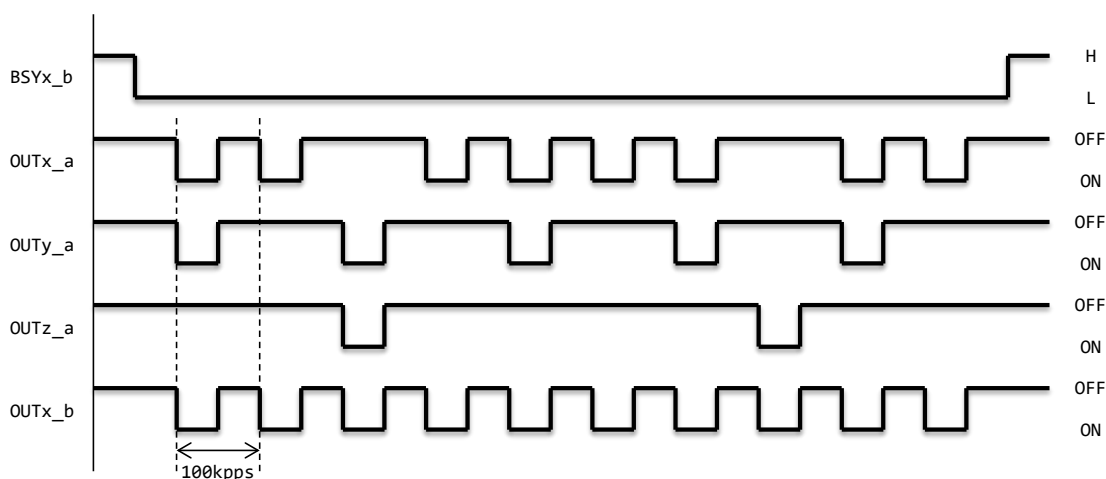
When the main axis reaches RPLS = 0, the command pulse is stopped.

When the command pulse is stopped, the operation mode is completed.

The following is a setting example to use two PCL6046s are used and the X-axis, Y-axis, and Z-axis of PCL6046_a are moved incremental to the X-axis of PCL6046_b at 8:5:2:10, respectively, when PCL6046_a and PCL6046_b are used.

The following is an example to use two PCL6046s: PCL6046_a and PCL6046_b. The X-axis, Y-axis, and Z-axis of PCL6046_a are moved incremental to the X-axis of PCL6046_b at 8:5:2:10 ratio, respectively.

Register	PCL6046_a			PCL6046_b
	Xa axis	Ya axis	Za axis	Xb axis
RMD.MOD	63h	63h	63h	63h
RMD.MSY	01b	01b	01b	01b
RMV	8	5	2	10
RIP	10	10	10	10
RFH	50000	50000	50000	50000
RMG	149	149	149	149
Interpolation control axis	○	○	○	○
Main axis	—	—	—	○



For the timing of BSY signal and OUT signal, see "7.5 Operation timing".

Set the feed amount in RMV register value.

Set the RMV register value of the main axis in RIP registers of all interpolation axes.

If RIP = 0 is set for the interpolation axis, REST.ESDT = 1 is set and the command pulse is stopped without being output.

5.5.9 Circular interpolation control

This control is to use a PCL6046 to perform circular interpolation by any two axes.

The remaining two axes can perform other operations instead of circular interpolation control.

The speed is set to the interpolation control axes.

The interpolation control axes are determined in the order of X, Y, Z-axis in the interpolation axes.

No.	Interpolation operation	Interpolation control axis
1)	Circular interpolation (X, Y-axes)	X-axis
2)	Circular interpolation (Y, Z axes)	Y-axis
3)	Circular interpolation (Z, U-axes)	Z-axis

Set the same value for the speed magnification (RMG) to all interpolation axes.

Acceleration, deceleration, and constant synthesized speed control can be used.

For acceleration and deceleration in circular interpolation control, see "6.3.5 Number of circular interpolation steps"

For constant synthesized speed control, see "6.3.6 Synthesized speed constant control"

The interpolation status can be checked with RIPS register, which is common to all axes.

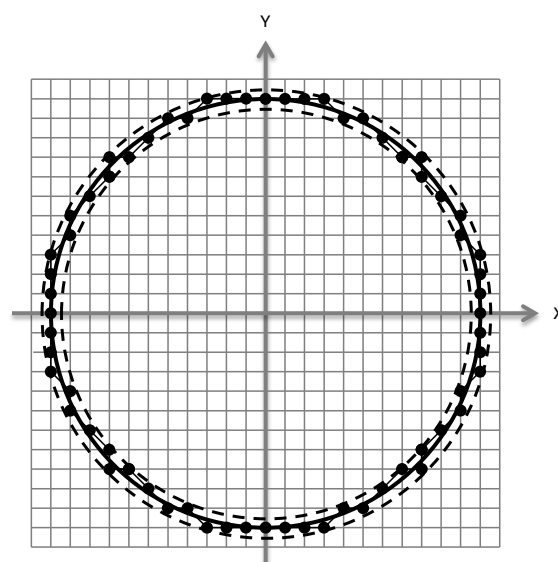
You can set the start and stop commands to all interpolation axes after setting from SELx bit to SELu bit of the axis selection.

Circular interpolation accuracy:

Circular interpolation draws a circle from the current coordinates to the end coordinates. The figure on the right is an example of drawing a perfect circle with a radius of 11.

The position accuracy for the specified curve during circular interpolation is ± 0.5 LBS within the entire interpolation section.

LSB is the smallest unit of RMV register and is the interval between the squares in the figure on the right. It corresponds to the resolution of a mechanical system.



●: Interpolation track
 Solid line: A circle of radius 11
 Dotted line: A circle of radius 11 ± 0.5

5.5.9.1 Circular interpolation in CW direction (64h)

When started, starts to output a command pulse so that two axes draw a circle in CW direction.

If REST.ESDT = 1 is set, the operation mode is canceled without outputting the command pulse.

For the REST.ESDT bit, see "5.4.7.2 REST: Error interrupt factor ".

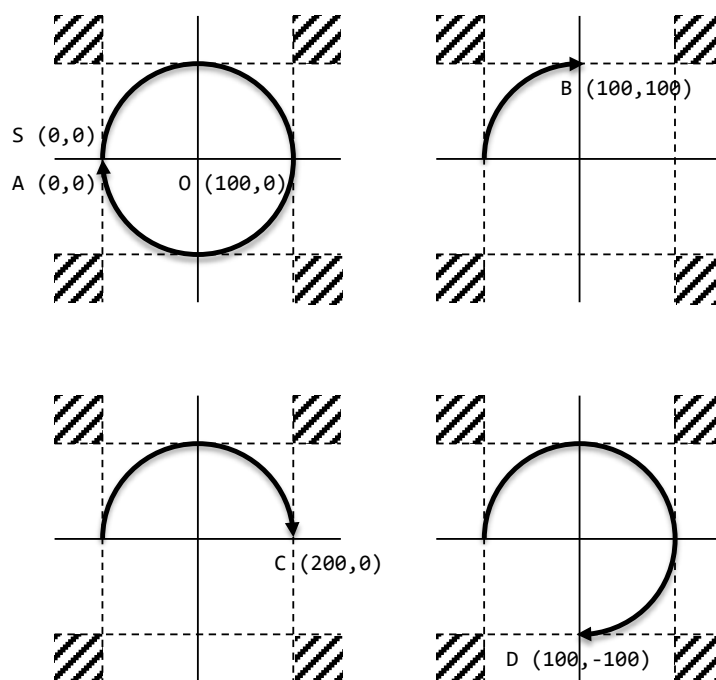
If one of the two axes reaches the end point coordinates on the circle, the command pulse is stopped.

When the command pulse is stopped, the operation mode is completed.

The following is the examples to set circular interpolations of 360-degree (perfect circle), 90-degree, 180-degree, and 270-degree by X and Y-axes.

Register	360°(A)		90°(B)		180°(C)		270°(D)	
	X-axis	Y-axis	X-axis	Y-axis	X-axis	Y-axis	X-axis	Y-axis
RMD.MOD	64h							
RMD.MPIE	0							
RMD.MIPM	0							
RMV	0	0	100	100	200	0	100	-100
RIP (O)	100	0	100	0	100	0	100	0
RCI	564	0	141	0	282	0	423	0
Interpolation control axis	○	—	○	—	○	—	○	—

Setting the speed and number of circular interpolation steps (RCI) other than the interpolation control axis does not affect the operation. For the number of circular interpolation steps, see "6.3.5 Number of circular interpolation steps".



If the end point coordinates of circular interpolation are set in the shaded area in the above figure, the permanent circular interpolation will be performed. Write the stop command to stop the command pulse.

Circular interpolation completes the interpolation operation where one axis reaches the end point in the end point quadrant. Therefore, even if the circular interpolation operation is completed, the specified end point coordinates may not be reached. If you want to automatically move to the end point coordinates outside the circular after the circular interpolation operation is completed, set the end point draw operation. For the end point draw, see "6.4.3 End point draw operation".

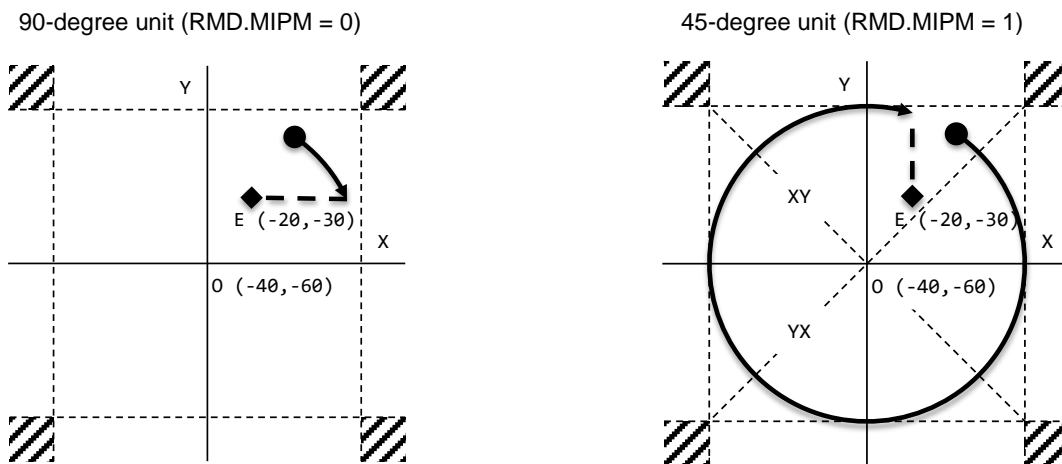
In PCL6046, you can select the end point check from 90-degree and 45-degree units when the start point and end point are in the same quadrant.

In 90-degree unit (RMD.MIPM = 0), the end point check can start from the start quadrant if the start point S is in the same quadrant as the end point E.

In 45-degree unit (RMD.MIPM = 1), it is judged by the positional relationship between the start point S and the end point E if the start point S is in the same quadrant as the end point E.

Examples of 90-degree and 45-degree trajectories:

Center O: (-40, -60) End point E: (-20, -30) (Thick broken line is the end point draw operation when RMD.MPIE = 1 is set)



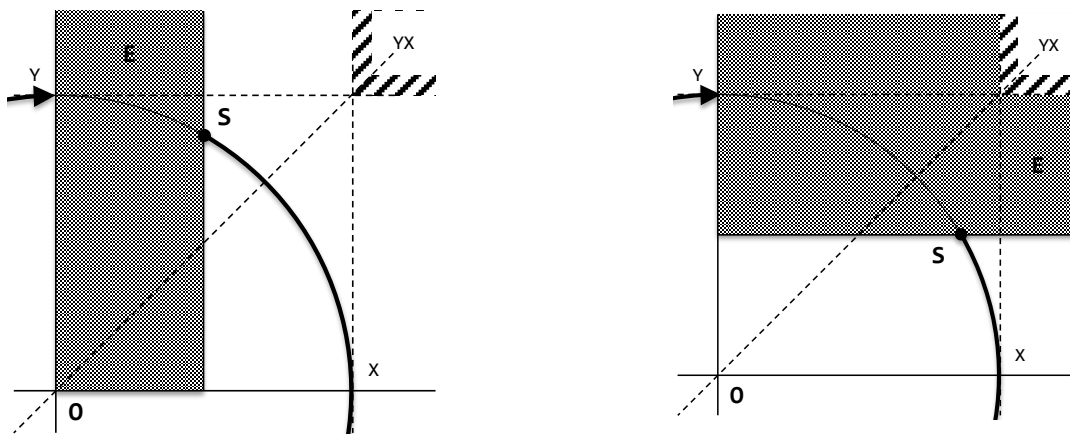
(Check the end point from the start quadrant even in the same quadrant).
Y-axis stops at coincidence with end point coordinates so that the trajectory becomes shorter

(Although it is in the same quadrant, the end point is not Checked.)
X-axis stops at coincidence with end point coordinates so that the trajectory becomes longer.

Not check the end point in the positional relationship between the start point S and the end point E (RMD.MIPM = 1)

The starting point S is in Y-YX-axis area in the first quadrant

The starting point S is in YX-X-axis area in the first quadrant



If the end point E is in the shaded area of each figure, the end point check will not be performed until the current position will return to this quadrant. In first quadrant, if $|RIPy| - |RIPx|$ is a negative number, the starting point S is determined in the area of YX-X-axis (right figure). If the starting point S is directly above YX-axis, $|RIPy| - |RIPx| = 0$ is determined not to be a negative number, and to be in the area of Y-YX-axis. The area pattern in and after the second quadrant is 90-degree rotations of the above figures.

5.5.9.2 Circular interpolation in CCW (65h)

When started, command pulses are output so that the two axes draw a circle in CCW direction.

If REST.ESDT = 1 is set, the operation mode is canceled without outputting the command pulse.

For REST.ESDT bit, see "5.4.7.2 REST: Error interrupt factor".

Other than this, operates in the same way as "5.5.9.1 Circular interpolation in CW direction (64h)".

5.5.10 U-axis synchronous control

This control is to use one PCL6046 to synchronize with U-axis and perform circular interpolation by the remaining two axes.

The remaining axes can perform operations other than circular interpolation.

The speed is set on the interpolation control axis.

The interpolation control axis is U-axis.

If linear interpolation 1 control is used together, the interpolation control axis is determined in the order of X, Y, and Z axes among the axes that move incremental to each other.

No.	Interpolation operation	Interpolation control axis
(1)	Positioning (U-axis) U-axis synchronization (X, Y-axes)	U-axis
(2)	Linear interpolation 1 (X, U-axes) U-axis synchronization (Y, Z axes)	X-axis
(3)	Linear interpolation 1 (Y, U-axes) U-axis synchronization (X, Z axes)	Y-axis
(4)	Linear interpolation 1 (Z, U-axes) U-axis synchronization (X, Y-axes)	Z-axis

Set the same value for the speed magnification (RMG) for all interpolation axes.

Combined with linear interpolation 1 control, you can use synthesized speed constant control for the circular interpolation axis.

For synthesized speed constant control, see "6.3.6 Synthesized speed constant control"

In RMV register of U-axis, which is also the interpolation control axis, set a value that exceeds the number of circular interpolation steps. If the circular interpolation operation is not completed when the final pulse of U-axis is output, abnormal stop (REST.ESDT = 1) occurs. Therefore, in the RMV register of U-axis, set a value greater than the number of circular interpolation steps.

For the end point draw operation (RMD.MPIE = 1), add the number of pulses for the end point draw operation to the number of circular interpolation steps.

The interpolation status can be checked in RIPS register, which is common to all axes.

The start command and stop command set from SELx to SELu bit of the axis selection and write to all the interpolation axes.

You can set the start and stop commands to all interpolation axes after setting from SELx to SELu bits of the axis selection.

5.5.10.1 Circular interpolation in CW (66h)

When started, command pulses are output so that two axes draw a circle in CW direction in synchronization with U-axis.

If REST.ESDT = 1 is set, the operation mode is canceled without outputting the command pulse.

For REST.ESDT bit, see "5.4.7.2 REST: Error interrupt factor".

When started as well as U-axis becomes RPLS = 0 during the circular interpolation operation, REST.ESDT = 1 will be set.

When either of the two axes reaches the end point coordinates on the arc, the command pulse stops.

When the command pulse stops, the operation mode is completed.

The following is an example in which X-axis and Y-axis are controlled by U-axis synchronization (perfect circle), and Z-axis and U-axis are controlled by linear interpolation 1 (1/2 of the number of circular interpolation steps).

Register	X-axis	Y-axis	Z-axis	U-axis
RMD.MOD	66h	66h	61h	61h
RMD.MIPF	1	1	0	0
RMV	0	0	282	565
RIP	100	0	0	0
RCI	0	0	0	0
RFL	0	0	65,500	0
RFH	0	0	65,535	0
RUR	0	0	1	0
RDR	0	0	0	0
RMG	2	2	2	2
RENV2.PMSK	0	0	0	1
Interpolation control axis	—	—	○	—
Main axis	—	—	—	○

You can also connect the spirals in the table above by using a continuous operation using the pre-register.

On the first lap, you can accelerate to FH constant speed by writing PRMDn.MSDP = 1, PRDPu = 0, STAUD (53h) commands.

On the second lap, you write STAFH (51h) command to continue the FH constant speed operation.

On the final lap, you can write PRMDn.MSDP = 0, PRDPu = 80, and STAD (52h) commands to decelerate to FL constant speed and stop. (Since there is no acceleration section in the speed pattern of high speed 1, the slow-down point cannot be set automatically.)

5.5.10.2 Circular interpolation in CCW (67h)

When started, command pulses are output so that the axes draw a circular in CCW direction in synchronization with U-axis.

If REST.ESDT = 1 is set, the operation mode is canceled without outputting the command pulse.

For the REST.ESDT bit, see "5.4.7.2 REST: Error interrupt factor".

Other than the above, it operates in the same way as "5.5.10.1 Circular interpolation in CW (66h)".

6. Function description

This chapter describes the features in PCL6046 LSI.

6.1 Reset

There are two types of reset in PCL6046: hardware reset and software reset.

After resetting, PCL6046 will be in the default status shown in the table below.

Item	Default
Register	0
Pre-register	0
Axis selection, Command	0
General-purpose output port	0
Input/output buffer	0
INT pin	H level
WRQ pin	H level
IFB pin	H level
D0 to D15 pins	Hi-Z
P0n to P7n pins	Input
CSTA pin	H level
CSTP pin	H level
OUTn pin	H level
DIRn pin	H level
ERCn pin	H level
BSYn pin	H level

C a u t i o n



After turning ON the PCL6046, be sure to perform a "hardware reset" before you start using it.

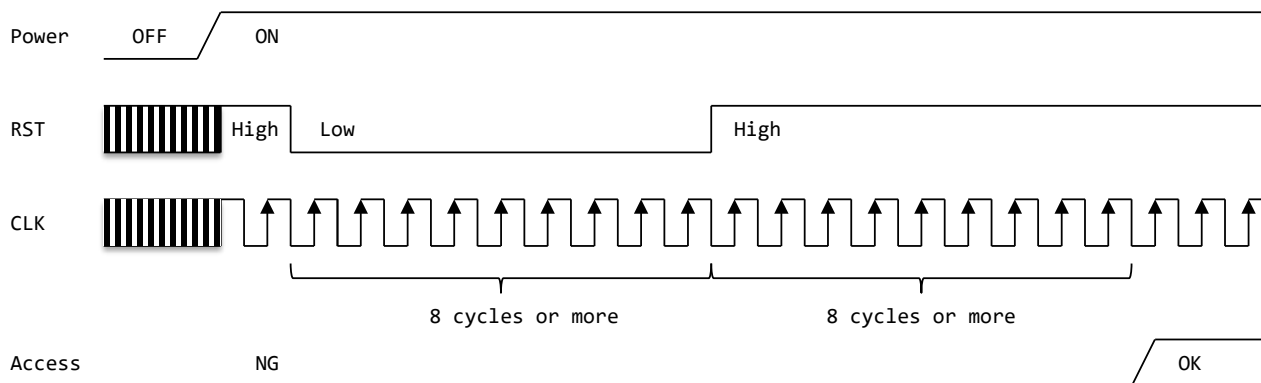
Before the reset is completed, a bi-directional pin may be an output pin.

Be careful of short circuits and heat generation.

6.1.1 Hardware reset

In PCL6046, you need to input RST signal to RST pin between when the power is turned ON and when CPU communication is started.

For RST signal, input an L level signal with 8 cycles or more of the CLK signal and an H level signal with 8 cycles or more of the CLK signal.

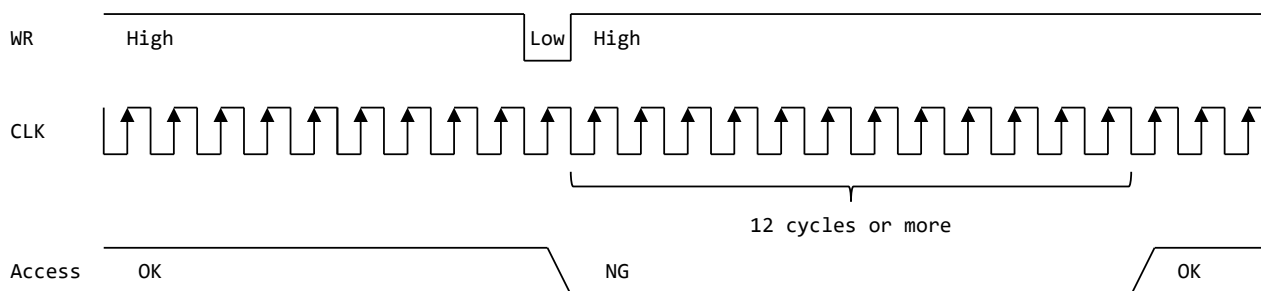


6.1.2 Software reset

After a hardware reset, you can use a software reset if you want to perform a reset again.

In a software reset, you write SRST (04h) command to one of the axes.

After writing SRST (04h) command, re-start CPU communication after 12 cycles or more of the CLK signal have elapsed.



6.2 Pre-register

An operation mode starts after you write the settings such as speed control and position control to registers.

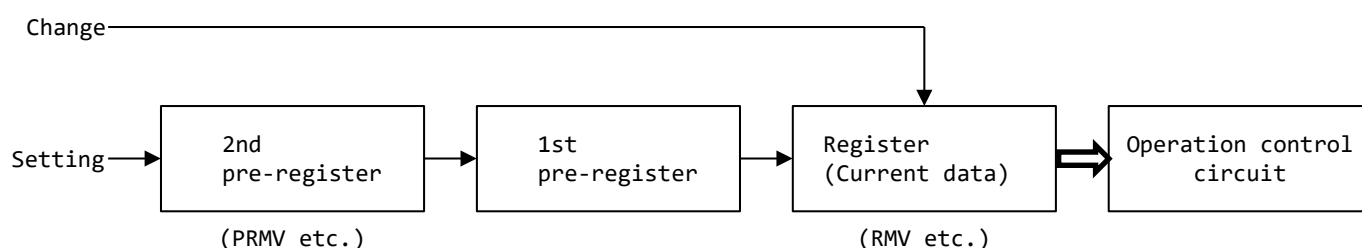
If you write the next settings to the register for subsequent operation after completing the operation mode, it will result in downtime for the write time.

The pre-registers allow you to write subsequent actions during operation, thus removing write downtime.

RMV, RFL, RFH, RUR, RDR, RMG, RDP, RMD, RIP, RUS, RDS and RCI registers and start command have the pre-registers for continuous operations.

The pre-registers for continuous operations are to set the continuous operation data and the continuous operation start command during operation.

There is a two-stage configuration as shown in the figure below, and operates in FIFO (queue).



RCMP5 register also has the pre-register for continuous comparison.

The pre-register for continuous comparison is to set the continuous comparison data.

There is a two-stage configuration as shown in the figure below, and operates in FIFO (queue).

Writing to a register with the pre-register is written to the 2nd pre-register.

If the data is the same as the previous one, no need to write to the 2nd pre-register.

To change the data in the current register in the determined status, write the data directly to the current register.

6.2.1 Continuous operation

The data written to the 2nd pre-register during stop shifts to the current register and also becomes the current data.

During operation, it shifts to the 1st pre-register and becomes the data for the 1st continuous operation.

If the 1st pre-register is determined during operation, it will not shift to the 1st pre-register and will be the data for the 2nd continuous operation.

The current register data and pre-register data for operations are determined by writing a start command.

The determined data in the 1st pre-register shifts when the current data completes the operation mode and starts automatically.

You can check the fixed status of the pre-register for continuous operation with RSTS.PFM bit and MST.SPRF bit.

Writing to the 2nd pre-register is invalid when MST.SPRF = 1 (RSTS.PFM = 11b).

When writing data for continuous operation, wait until the current data completes the operation mode and becomes MST.SPRF = 0.

To change the data in the 2nd pre-register in determined status, change it to RSTS.PFM = 01b with PRECAN (26h) command.

Then, if you set RSTS.PFM = 10b with the start command for the 1st pre-register, you can change only the 2nd pre-register.

The relationship between the write status of the pre-register and RSTS.PFM bit / MST.SPRF bit is as follows.

No.	Method	2 nd pre-register	1 st pre-register	Current register	PFM	SPRF
1	This is the default status when operation is stopped.	0 (Undetermined)	0 (Undetermined)	0 (Undetermined)	0	0
2	While stopped, write data 1 to the 2nd pre-register. Data 1 is copied to the 1st pre-register. Data 1 is also copied to the current register.	Data 1 (Undetermined)	Data 1 (Undetermined)	Data 1 (Undetermined)	0	0
3	Write start command 1. Data 1 in the current register is determined. Starts an operation with data 1 and start command 1.	Data 1 (Undetermined)	Data 1 (Undetermined)	Data 1 (Determined)	1	0
4	While operating with data 1, write data 2 to the 2nd pre-register for a continuous operation. Writing the same data as the last time can be omitted. Since the current register is determined, the data 2 is copied only to the 1st pre-register.	Data 2 (Undetermined)	Data 2 (Undetermined)	Data 1 (Determined)	1	0
5	While operating with data 1, write start command 2 for a continuous operation. Data 2 in the 1st pre-register is determined.	Data 2 (Undetermined)	Data 2 (Determined)	Data 1 (Determined)	2	0
6	While operating with data 1, write data 3 for a continuous operation to the 2nd pre-register. Writing the same data as the last time can be omitted. Since the 1st pre-register is determined, the data 3 will not be copied.	Data 3 (Undetermined)	Data 2 (Determined)	Data 1 (Determined)	2	0
7	Write start command 3 for a continuous operation while operating with data 1. Data 3 of the 2nd pre-register is determined.	Data 3 (Determined)	Data 2 (Determined)	Data 1 (Determined)	3	1
8	Data 1 completes the operation mode. Data 2 is copied to the current register. Data 3 is copied to the 1st pre-register. Start the operation with data 2 and start command 2. The 2nd pre-register becomes undetermined. You can write if you have data 4 or start command 4.	Data 3 (Undetermined)	Data 3 (Determined)	Data 2 (Determined)	2	0
9	Data 2 completes the operation mode. Data 3 is copied to the current register. Start the operation with data 3 and start command 3. The 1st pre-register becomes undetermined.	Data 3 (Undetermined)	Data 3 (Undetermined)	Data 3 (Determined)	1	0
10	Data 3 completes the operation mode. The current register becomes undetermined, and a continuous operation is completed.	Data 3 (Undetermined)	Data 3 (Undetermined)	Data 3 (Undetermined)	0	0

If the 2nd pre-register for a continuous operation is set to be writable (RIRQ.IRNM = 1) in the interrupt request, the 2nd pre-register writable (RIST.ISNM = 1) for continuous operation of interrupt factor can be generated when the 2nd pre-register for a continuous operation changes to the undetermined status (MSTS.SPRF bit is 1 to 0).

To start automatically using continuous operation, set the operation mode completion timing to final pulse cycle completed (PRMD.METM = 0).

If "the final pulse ON width completed" is set (PRMD.METM = 1), the interval between the final pulse and the first pulse of continuous operation becomes narrower, and the motor driver may malfunction.

For the completion timing of an operation mode, see "6.5.2 Operation complete timing".

The continuous operation is canceled by writing PRECAN (26h) command, writing a stop command (49h, 4Ah), or stopping due to the cause of an error interrupt. The data for continuous operation can be shifted with the PRESHF (2Bh) command even if the previous operation mode has not been completed.

When controlling with a user program of machine tool such as NC (Numerical Control), you may want to control the operation blocks to be executed.

In this case, it is difficult to control the operation block only by the control software, so use the RMD.MSN bit.

RMD.MSN bit can be repeatedly set from 0 to 3 for each operation block.

MSTS.SSC bit can be read during operation to correspond to the operation block executed by the user program.

6.2.2 Continuous comparison

When RSTS.PFC = 00b, the 2nd pre-register (PRCP5) write data shifts to the current register (RCMP5).

At this time, RCMP5 register is determined, and the write data becomes the current data.

When RSTS.PFC = 01b, it shifts to the 1st pre-register and becomes the 1st continuous comparison data.

When RSTS.PFC = 10b, it will not be shifted and will be the 2nd continuous comparison data.

The data for continuous comparison written to PRCP5 register is determined in the order of writing.

The determined data in the 1st pre-register shifts to the current register when the current comparison result changes from true to false.

You can check the determined status of the pre-register for continuous comparison with RSTS.PFC bit and MSTS.SPFD bit.

Writing to PRCP5 register is invalid when MSTS.SPFD = 1 (RSTS.PFC = 11b).

When writing data for a continuous comparison, wait until the current comparison shifts to MSTS.SPFD = 0.

To change the data in PRCP5 register in determined status, change it to RSTS.PFC = 01b with PCPCAN (27h) command.

Then, a continuous comparison data for the 1st pre-register and the continuous comparison data for the 2nd pre-register are written.

Writing to RCMP5 register when RSTS.PFC > 00b overrides the current data.

When RSTS.PFC = 00b, do not write to RCMP5 register, but write to PRCP5 register.

The relationship between the pre-register write status and RSTS.PFC bit / MSTS.SPFD bit is as follows.

No.	Procedures	2 nd pre-register	1 st pre-register	Current register	PFC	SPDF
1	Default status.	0 (Undetermined)	0 (Undetermined)	0 (Undetermined)	0	0
2	Write data 1 to PRCP5 register. Data 1 is copied to the 1st pre-register. Data 1 is copied to RCMP5 register. RCMP5 register is determined.	Data 1 (Undetermined)	Data 1 (Undetermined)	Data 1 (Determined)	1	0
3	Write data 2 to PRCP5 register. Data 2 is copied to the 1st pre-register. The 1st pre-register is determined.	Data 2 (Undetermined)	Data 2 (Determined)	Data 1 (Determined)	2	0
4	Write data 3 to PRCP5 register. PRCP5 register is determined.	Data 3 (Determined)	Data 2 (Determined)	Data 1 (Determined)	3	1
5	The comparison result of data 1 changes from true to false. Data 2 is copied to RCMP5 register. Data 3 is copied to the 1st pre-register.	Data 3 (Undetermined)	Data 3 (Determined)	Data 2 (Determined)	2	0
6	The comparison result of data 2 changes from true to false. Data 3 is copied to RCMP5 register.	Data 3 (Undetermined)	Data 3 (Undetermined)	Data 3 (Determined)	1	0
7	The comparison result of data 3 changes from true to false. RCMP5 register becomes undetermined and the continuous comparison is completed.	Data 3 (Undetermined)	Data 3 (Undetermined)	Data 3 (Undetermined)	0	0

If the 2nd pre-register for a continuous operation is set to be writable (RIRQ.IRND = 1) in the interrupt request, the 2nd pre-register writable (RIST.ISND = 1) for continuous operation of interrupt factor can be generated when the 2nd pre-register for a continuous operation changes to the undetermined status (MSTS.SPFD bit is 1 to 0).

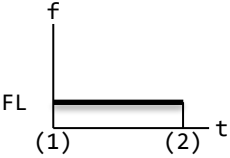
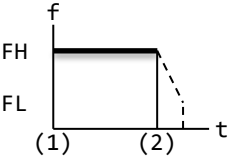
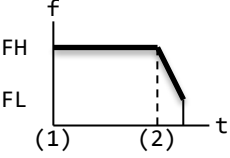
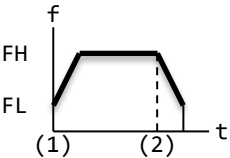
The data for continuous comparison can be shifted with PCPSHF (2Ch) command even if the condition of RENV5.C5C bit is not satisfied.

6.3 Speed control

This section describes the speed control functions such as speed patterns selected by operation commands and speed setting examples.

6.3.1 Speed patterns

Describes the speed patterns selected by operation commands.

Speed pattern	Continuous movement operation mode	Incremental movement operation mode
FL constant speed 	(1) Executes STAF _L (50h) command ⇒ FL constant speed start (2) Executes STOP (49h) command or SDSTP (4Ah) command ⇒ Immediate stop	(1) Executes STAF _L (50h) command ⇒ FL constant speed start (2) RPLS = 0 or Execute STOP (49h) command or SDSTP (4Ah) command ⇒ Immediate stop
FH constant speed 	(1) Executes STAF _H (51h) command ⇒ FH constant speed start (2) Executes STOP (49h) command ⇒ Immediate stop * Executes SDSTP (4Ah) command ⇒ Starts deceleration and stops at FL speed.	(1) Execute the STAF _H (51h) command ⇒ FH constant speed start. (2) RPLS = 0 or Executes STOP (49h) command ⇒ Immediate stop * Executes SDSTP (4Ah) command ⇒ Starts deceleration and stops at FL speed
High-speed 1 	(1) Execute STAD (52h) command ⇒ FH constant speed start * RENV5.IDL > 0 ⇒ FL constant speed start ⇒ Outputs idling pulses. ⇒ Accelerates to FH speed. (2) Executes SDSTP (4Ah) command ⇒ Starts deceleration and stops at FL speed. * Executes STOP (49h) command ⇒ Immediate stop	(1) Executes STAD (52h) command ⇒ FH constant speed start * RENV5.IDL > 0 ⇒ FL constant speed start ⇒ Outputs idling pulses. ⇒ Accelerates to FH speed. (2) RPLS < RSDC ⇒ Starts deceleration. * RPLS = 0 or Executes STOP (49h) command. ⇒ Immediate stop * Executes SDSTP (4Ah) command ⇒ Starts deceleration and stops at FL speed.
High-speed 2 	(1) Executes STAUD (53h) command ⇒ FL constant speed start. ⇒ Outputs idling pulses. ⇒ Accelerates to FH speed. (2) Executes SDSTP (4Ah) command. ⇒ Starts deceleration and stops at FL speed. * Executes STOP (49h) command ⇒ Immediate stop	(1) Executes STAUD (53h) command ⇒ FL constant speed start. ⇒ Outputs idling pulse. ⇒ Accelerates to FH speed. (2) RPLS < RSDC ⇒ Starts deceleration. * RPLS = 0 or Executes STOP (49h) command ⇒ Immediate stop * Executes SDSTP (4Ah) command ⇒ Starts deceleration and stops at FL speed.

For idling pulses, see "6.6 Idling control".

6.3.2 Speed setting example

The following example shows the case where:

$f_{CLK} = 19.6608 \text{ MHz}$ 、FL speed = 10 pps、FH speed = 100 kpps、Acceleration time = 300 ms、Deceleration time = 300 ms.

1. With FH speed <math> < 65,535\text{pps} \times 2</math>, obtain the RMG register value so that doubles the speed multiplier from f_{CLK} .
RMG=149(095h)

$$RMG = \frac{19,660,800[\text{Hz}]}{2 \times 65,536} - 1 = 149$$

For speed magnification, see "5.4.1.5 RMG(PRMG): Speed magnification".

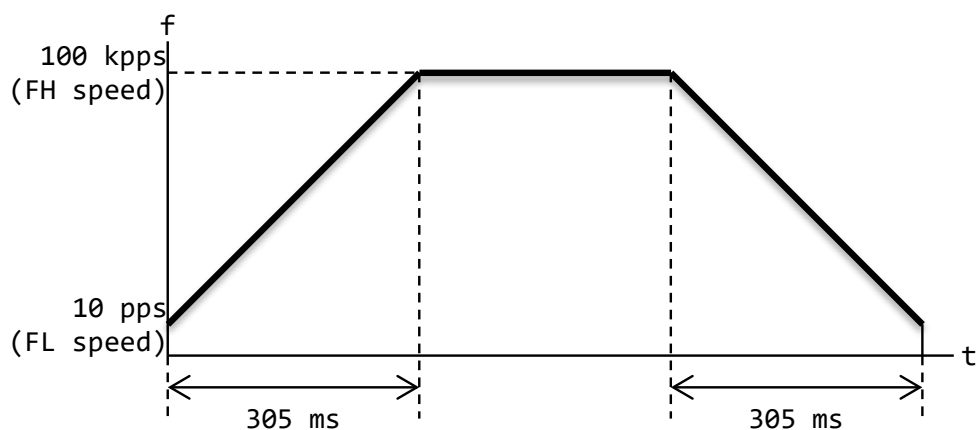
2. Obtain the RFH register value so that FH speed=100 kpps from the speed magnification.
RFH=50,000(C350h)
3. Obtain the RFL register value so that FL speed=10 kpps from the speed magnification.
RFL=5(0005h)
4. Obtain the RUR register value so that Acceleration time=300 ms from calculation formula.
RUR=28.494(001Ch or 001Dh)

$$RUR = \frac{19,660,800[\text{Hz}] \times 0.3[\text{s}]}{(50,000 - 5) \times 4} - 1 = 28.494$$

5. Because of Deceleration time = Acceleration time, the RDR register value can also be set to 0.
RDR=28.494(0000h)

<Example of speed setting of RUR = 29>

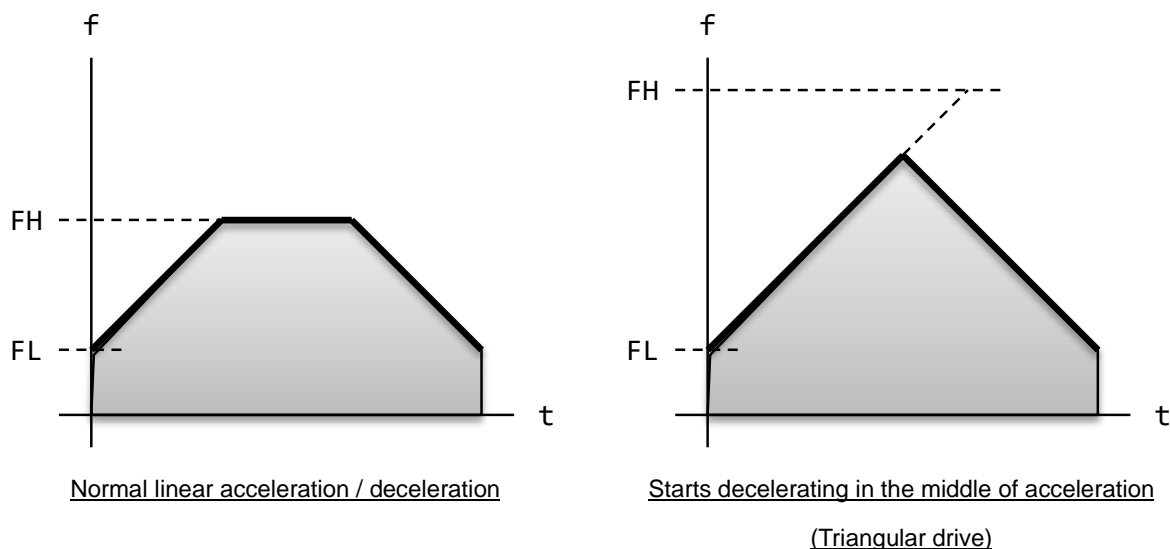
Writing register	Setting value	Actual value
PRFL	0005h	10 pps
PRFH	C350h	100 kpps
PRMG	095h	2 times
PRUR	001Dh	305 ms
PRDR	0000h	305 ms



6.3.3 Manual correction calculation of FH speed

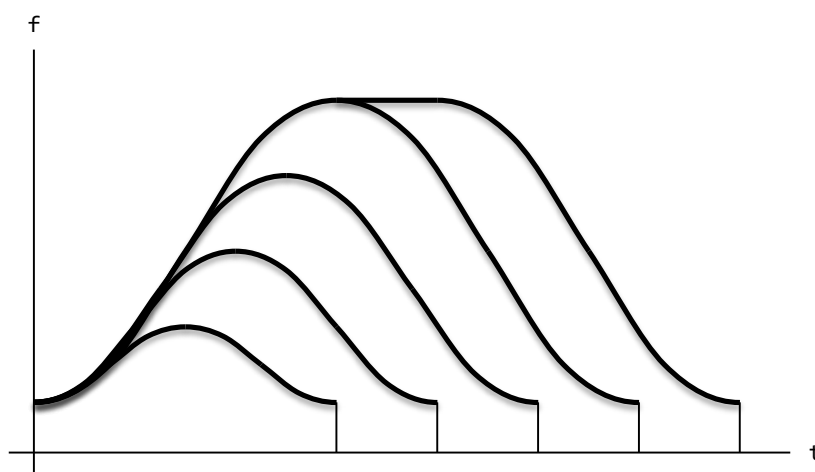
When accelerating or decelerating in an operation mode that allows you to set a target position, the speed pattern may become triangular drive. The target is RMD.MOD = 41h, 42h, 43h, 44h, 45h, 51h, 52h, 53h, 54h, 55h, 56h, 61h, 64h, 65h, 66h, 67h, 69h, 6Ch and 6Dh.

If FH speed is too high for the feed amount, or if a feed amount is too small for the FH speed, it will be the triangular drive.



To avoid a triangular drive (RMD.MADJ = 0), the FH speed is automatically lowered.

In this case, if the acceleration and deceleration curves are asymmetric, an error will occur.



Automatic correction of operating speed proportional to the amount of movement

If the slow-down point is set automatically (RMD.MSDP = 0), the slow-down point will also be corrected.

In this case, if you set deceleration time > acceleration time x 2, the start of deceleration will be delayed and the operation will stop before reaching the FL speed.

This can be avoided by setting an offset in RDP register or setting manually the slow-down point (RMD.MSDP = 1).

The following describes how to calculate the FH speed, which does not result in triangular drive when the acceleration and deceleration curves are asymmetric.

6.3.3.1 Linear acceleration/deceleration

The FH speed in linear acceleration / deceleration ($RMD.MSMD = 0$) is calculated by the following formula.

If

$$RMV \cong \frac{(RFH^2 - RFL^2) \times (RUR + RDR + 2)}{(RMG + 1) \times 32768} ,$$

$$RFH \cong \sqrt{\frac{(RMG + 1) \times 32768 \times RMV}{RUR + RDR + 2} + RFL^2}$$

6.3.3.2 Complete S-curve acceleration/deceleration

The FH speed in a complete S-curve acceleration / deceleration ($RMD.MSMD = 1$, $RUS = 0$, $RDS = 0$) without a linear acceleration / deceleration section is calculated by the following formula.

If

$$RMV \cong \frac{(RFH^2 - RFL^2) \times (RUR + RDR + 2) \times 2}{(RMG + 1) \times 32768} ,$$

$$RFH \cong \sqrt{\frac{(RMG + 1) \times 32768 \times RMV}{(RUR + RDR + 2) \times 2} + RFL^2}$$

6.3.3.3 Partial S-curve acceleration/deceleration

The FH speed in S-curve acceleration / deceleration ($RMD.MSMD = 1$, $RUS > 0$ or $RDS > 0$) with a linear acceleration / deceleration section based on the relationship between RUS register and RDS register can be obtained as follows.

6.3.3.3.1.1 RUS = RDS

If

$$RMV \cong \frac{(RFH + RFL) \times (RFH - RFL + 2 \times RUS) \times (RUR + RDR + 2)}{(RMG + 1) \times 32768} \text{ and}$$

$$RMV > \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 32768} ,$$

$$RFH \cong -RUS + \sqrt{\frac{(RMG + 1) \times 32768 \times RMV}{(RUR + RDR + 2)} + (RUS - RFL)^2} .$$

If

$$RMV \cong \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 32768} ,$$

Change to complete S-curve acceleration / deceleration ($RUS = 0$, $RDS = 0$) without a linear acceleration / deceleration section.

$$RFH \cong \sqrt{\frac{(RMG + 1) \times 32768 \times RMV}{(RUR + RDR + 2) \times 2} + RFL^2}$$

6.3.3.3.1.2 RUS < RDS

If

$$RMV \leq \frac{(RFH + RFL) \times ((RFH - RFL) \times (RUR + RDR + 2) + 2 \times RUS \times (RUR + 1) + 2 \times RDS \times (RDR + 1))}{(RMG + 1) \times 32768} \quad \text{and}$$

$$RMV > \frac{(RDS + RFL) \times (RDS \times (RUR + 2 \times RDR + 3) + RUS \times (RUR + 1)) \times 4}{(RMG + 1) \times 32768} ,$$

$$RFH \leq \frac{-A + \sqrt{A^2 + B}}{RUR + RDR + 2}$$

However, $A = RUS \times (RUR + 1) + RDS \times (RDR + 1)$

$$B = ((RMG + 1) \times 32768 \times RMV - 2 \times A \times RFL + (RUR + RDR + 2) \times RFL^2) \times (RUR + RDR + 2)$$

If

$$RMV \leq \frac{(RDS + RFL) \times (RDS \times (RUR + 2 \times RDR + 3) + RUS \times (RUR + 1)) \times 4}{(RMG + 1) \times 32768} \quad \text{and}$$

$$RMV > \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 32768} ,$$

change to S-curve deceleration ($RUS > 0$, $RDS = 0$) without a linear deceleration section.

$$RFH \leq \frac{-A + \sqrt{A^2 + B}}{RUR + 2 \times RDR + 3}$$

However, if $A = RUS \times (RUR + 1)$

$$B = ((RMG + 1) \times 32768 \times RMV - 2 \times A \times RFL + (RUR + 2 \times RDR + 3) \times RFL^2) \times (RUR + 2 \times RDR + 3)$$

$$RMV \leq \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 32768} ,$$

change to S-curve acceleration / deceleration ($RUS = 0$, $RDS = 0$) without a linear acceleration / deceleration part.

$$RFH \leq \sqrt{\frac{(RMG + 1) \times 32768 \times RMV}{(RUR + RDR + 2) \times 2} + RFL^2}$$

6.3.3.3.1.3 RUS>RDS

If

$$RMV \leq \frac{(RFH + RFL) \times ((RFH - RFL) \times (RUR + RDR + 2) + 2 \times RUS \times (RUR + 1) + 2 \times RDS \times (RDR + 1))}{(RMG + 1) \times 32768} \quad \text{and}$$

$$RMV > \frac{(RUS + RFL) \times (RUS \times (2 \times RUR + RDR + 3) + RDS \times (RDR + 1)) \times 4}{(RMG + 1) \times 32768} ,$$

$$RFH \leq \frac{-A + \sqrt{A^2 + B}}{RUR + RDR + 2}$$

However, $A = RUS \times (RUR + 1) + RDS \times (RDR + 1)$

$$B = ((RMG + 1) \times 32768 \times RMV - 2 \times A \times RFL + (RUR + RDR + 2) \times RFL^2) \times (RUR + RDR + 2)$$

If

$$RMV \leq \frac{(RUS + RFL) \times (RUS \times (2 \times RUR + RDR + 3) + RDS \times (RDR + 1)) \times 4}{(RMG + 1) \times 32768} \quad \text{and}$$

$$RMV > \frac{(RDS + RFL) \times RDS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 32768} ,$$

change to S-curve acceleration (RUS = 0, RDS > 0) without a linear acceleration section.

$$RFH \leq \frac{-A + \sqrt{A^2 + B}}{2 \times RUR + RDR + 3}$$

However, if $A = RDS \times (RDR + 1)$

$$B = ((RMG + 1) \times 32768 \times RMV - 2 \times A \times RFL + (2 \times RUR + RDR + 3) \times RFL^2) \times (2 \times RUR + RDR + 3)$$

$$RMV \leq \frac{(RDS + RFL) \times RDS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 32768} ,$$

change to S-curve acceleration / deceleration (RUS = 0, RDS = 0) without a linear acceleration / deceleration part.

$$RFH \leq \sqrt{\frac{(RMG + 1) \times 32768 \times RMV}{(RUR + RDR + 2) \times 2} + RFL^2}$$

6.3.4 Target speed override

The target speed can be overridden (speed change) by re-writing RFH, RUR, RDR, RUS, and RDS registers during operation.

While operating the FL or FH constant speed pattern, the speed changes to a new speed without accelerating or decelerating.

While operating in high speed 1 or 2 speed pattern, accelerates or decelerates to reach the new speed.

The target speed override is reflected from the output pulse on write.

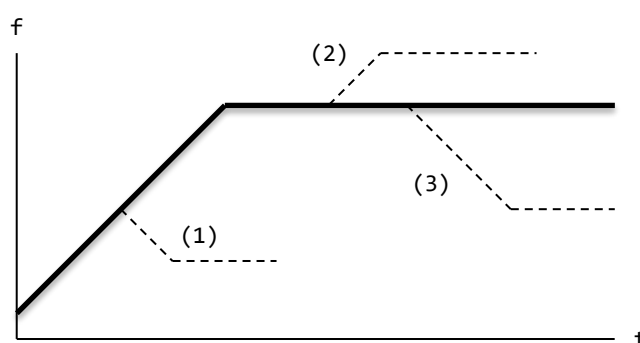
To rewrite multiple registers at once, see "6.13.6 Bulk override".

When setting the slow-down point automatically (RMD.MSDP = 0), do not rewrite the values other than the RFH register.

If you re-write RFL, RUR, RDR, RUS, and RDS registers, the RSDC register values will not be calculated correctly.

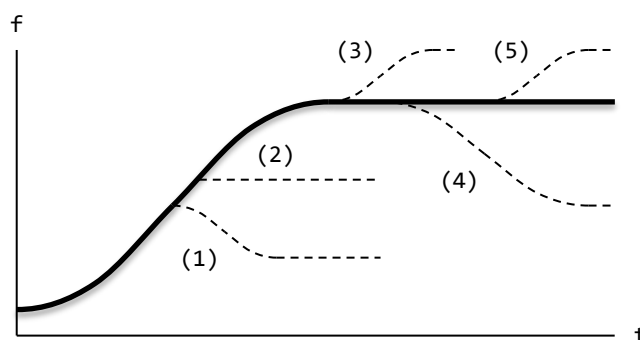
The target is RMD.MOD = 41h, 42h, 43h, 44h, 45h, 51h, 52h, 53h, 54h, 55h, 56h, 61h, 64h, 65h, 66h, 67h, 69h, 6Ch and 6Dh.

Example of speed pattern change due to speed change during linear acceleration / deceleration operations



- (1) Change RFH register value during acceleration: If the changed speed is lower than the current speed, it will decelerate linearly to that speed.
- (2) (3) Change RFH register value after acceleration: Linear acceleration or linear deceleration to that speed.

Example of speed pattern change due to speed change during S-curve acceleration / deceleration operation



- (1) Change RFH register value during acceleration: If the changed speed is lower than the current speed, it will perform S-curve deceleration to that speed.
- (2) Change RFH register value during acceleration: If the changed speed is higher than or equal to the current speed and less than or equal to the original target speed, the S-curve characteristics will not be changed and the speed will be accelerated to the changed speed.
- (3) Change RFH register value during acceleration: If the changed speed exceeds the original target speed, it will accelerate to the original target speed without changing the S-curve characteristics and re-accelerate to that changed speed.
- (4) (5) Change RFH register value after acceleration: S-curve accelerates or decelerates to the change speed.

6.3.5 Number of circular interpolation steps

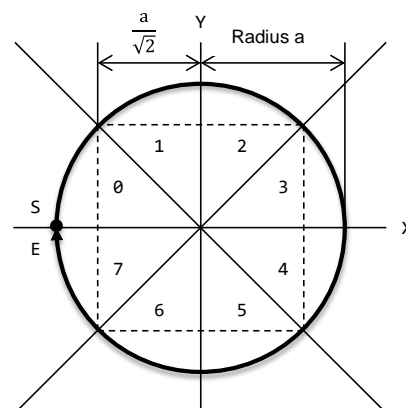
If the number of circular interpolation steps is set in RCI register, deceleration starts when the interpolation control axis becomes RCIC <RSDC. The number of circular interpolation steps is the number of pulses (number of steps) output by either axis.

The interpolation control axes are determined in the order of X, Y, Z axis in the interpolation axes. For slow-down point setting (RMD.MSDP) and FH correction setting (RMD.MADJ), set the same value for all interpolation axes.

In circular interpolation control, the speed of each interpolation axis fluctuates like a trigonometric function. The acceleration / deceleration characteristics of each interpolation axis are synthesized with this speed fluctuation. So, the synthesized speed differs depending on the interpolation angle, and the acceleration / deceleration characteristics do not match the settings.

RCI register (number of circular interpolation steps) is used to generate the deceleration start timing in circular interpolation controls. When decelerating with a circular interpolation control, set the number of circular interpolation steps of 1 or more in RCI register. When calculating the number of circular interpolation steps, divide the XY plane shown in the figure below into eight areas at the center of the circle. The status of the command pulse output by each axis in each area is as follows.

Area	Interpolated X-axis output pulse	Interpolated Y-axis output pulse
0	Outputs according to the interpolation calculation result	Always outputs
1	Always outputs	Outputs according to the interpolation calculation result
2	Always outputs	Outputs according to the interpolation calculation result
3	Outputs according to the interpolation calculation result	Always outputs
4	Outputs according to the interpolation calculation result	Always outputs
5	Always outputs	Outputs according to the interpolation calculation result
6	Always outputs	Outputs according to the interpolation calculation result
7	Outputs according to the interpolation calculation result	Always outputs



In this way, either axis always outputs pulses in any area.

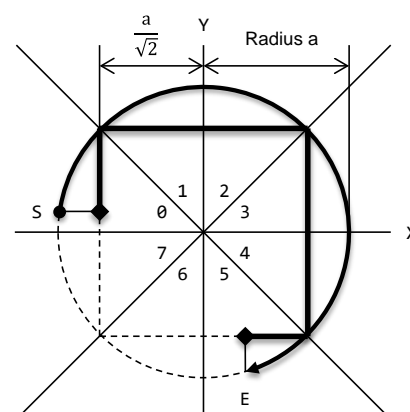
Therefore, the number of circular interpolation steps is equal to the number of pulses traveling along the trajectory of the inscribed square.

For example, if you draw a 90-degree circular with a radius of a , the number of circular interpolation steps will be $2 \times \frac{a}{\sqrt{2}}$

Set this value in RCI register.

To find the number of circular interpolation steps at any start and end points as shown on the right, follow the procedures below:

- (1) Specify which area the start point S belongs to out of the areas 0 to 7 from the center coordinates and find the intersection of the perpendiculars drawn from the start point to the inscribed square.
- (2) Specify which area the end point E belongs to out of the areas 0 to 7 from the end point coordinates and the center coordinates and find the intersection of the perpendiculars drawn from the end point to the inscribed square.
- (3) Find the length from the intersection of the start point perpendicular to the intersection of the end perpendicular on the inscribed square and set it in RCI register.



If the end point position is not on the circle, add the number of pulses required for the end point draw operation and set it in the RCI register.

If the RCI register value is set smaller than the calculation result, deceleration starts earlier, and FL constant speed time will occur. If a value larger than the calculation result is set, deceleration starts delayed and will stop before reaching FL speed. In either case, the interpolation trajectory will be the same as the constant speed circular interpolation.

When using the manual slow-down point setting (RMD.MSDP = 1), the formula of slow-down point (RDP) in the positioning control can be applied with the number of circular interpolation steps (RCI) as the feed amount (RMV).

However, when using constant synthesized speed control (RMD.MIPF = 1), obtain it from the change in the RCIC register value in the experiment.

R e m a r k s

If you can perform a test run, the optimum RCI value can be determined by the following experiment.

1. Set the command pulse output invalid (RENV2.PMSK = 1).
2. Set RCI = FFFFFFFFh (maximum value: 4,294,967,295).
3. Set the register values required for other circular interpolation operations.
4. Start the circular interpolation operation with STAFH (51h) command.
5. When the circular interpolation operation stops, calculate the difference between the RCI register and RCIC register values.

The calculation result will be the optimum RCI value.

I m p o r t a n t

The start point of circular interpolation control can always be specified on a circle.

However, it is often not possible to specify the end point on a circle.

To reach the end point outside the circle, perform the end point draw operation after the circular interpolation.

The number of pulses for this end point draw operation must also be added to the number of circular interpolation steps.

For the end point draw operation, see "6.4.3 End point draw operation".

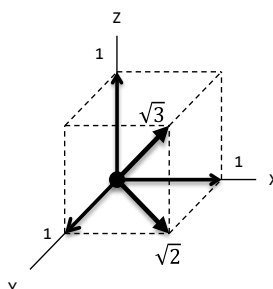
6.3.6 Constant synthesized speed control

Constant synthesized speed control is a function to keep the synthesized speed of the axes that perform linear interpolation 1 control and circular interpolation control constant.

Operation mode	MRD.MIPF=1	Description
Linear interpolation 1 control (60h, 61h)	○	Enabled
Linear interpolation 2 control (62h, 63h)	×	Disabled
Circular interpolation control (64h, 65h)	○	Enabled
U-axis synthesized control (66h, 67h)	△	Enabled for circular interpolation axis in combination with linear interpolation 1 control

Constant synthesized speed control can be performed when the setting value of RMD.MOD bit is the same and the axis with RMD.MIPF = 1 setting. It cannot be used with linear interpolation 2 control. If you use the axis that does not perform circular interpolation and U-axis together, you can perform the constant synthesized speed control on circular interpolation axis even with U-axis synchronous control.

When two orthogonal axes output pulses simultaneously, the moving distance is $\sqrt{2}$ times that of one axis, so the moving speed will be $\sqrt{2}$ times. Constant synthesized speed control controls the moving speed to be constant by ensuring $\sqrt{2}$ times the time until the next pulse is output. When three orthogonal axes output pulses simultaneously, the travel distance is $\sqrt{3}$ times that of one axis, so $\sqrt{3}$ times the time is secured.



When 4-axis output pulses simultaneously, $\sqrt{3}$ times time is secured although constant synthesized speed control is not always performed. For example, when performing an incremental movement of linear interpolation 1 by three axes, X, Y, and Z, only $\sqrt{2}$ times time is secured even with simultaneous output of three axes when RMD.MIPF = 0 only on Z axis.

Since the interpolation control moves 1 LSB in mechanical resolution, the actual movement time is different from that of the ideal trajectory. For example, if the operation speed of the trajectory shown in "Linear interpolation accuracy" of "5.5.7 Linear interpolation 1 control" is 1 pps, the movement time is 10 seconds. In constant synthesized speed control, $\sqrt{2}$ times time is secured for the trajectory moving in the direction of 45 degrees, so $6 + 4\sqrt{2} \approx 11.66$ seconds is the moving time. The ideal trajectory movement time is $\sqrt{10^2 + 4^2} \approx 10.77$ seconds, so use constant synthesized speed control noting this difference.

When RMD.MIPF = 1 is set, it is not recommended to use acceleration or deceleration simultaneously. If the control axis is RUR \neq RDR, the RSDC register value will not be calculated properly when RMD.MIPF = 1 is set. When the speed is changed with S-curve acceleration / deceleration characteristics, the RSDC register value will not be calculated properly if RMD.MIPF = 1 is set.

For circular interpolation control, setting RMD.MIPF = 1 and RMD.MSDP = 0 will calculate the RSDC register value properly. However, both the start and the end points must be on the arc center axis (in 90-degree increments).

6.4 Position control

This section describes position control functions such as re-writing RMV registers and waiting for PCS signal input.

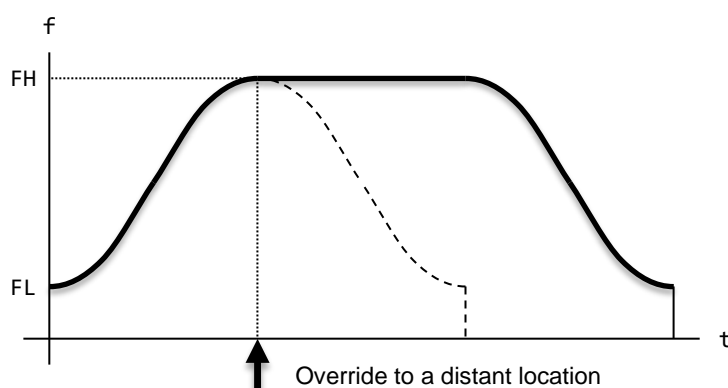
6.4.1 Target position override 1 (RMV)

Target position override 1 is available in RMD.MOD = 41h, 42h and 43h operation modes.

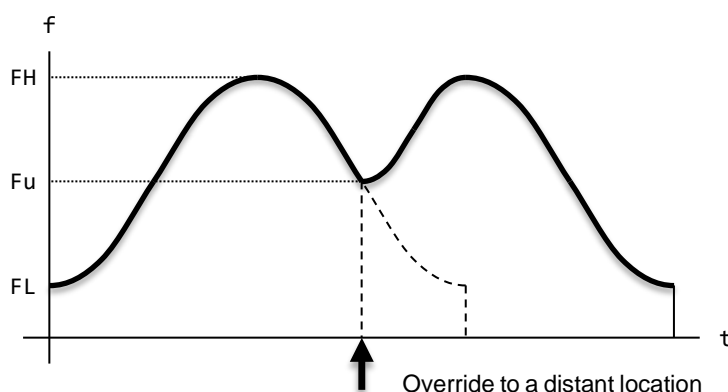
In other modes of operation, do not re-write RMV register during operation.

You can change the target position by writing a new target position to RMV register.

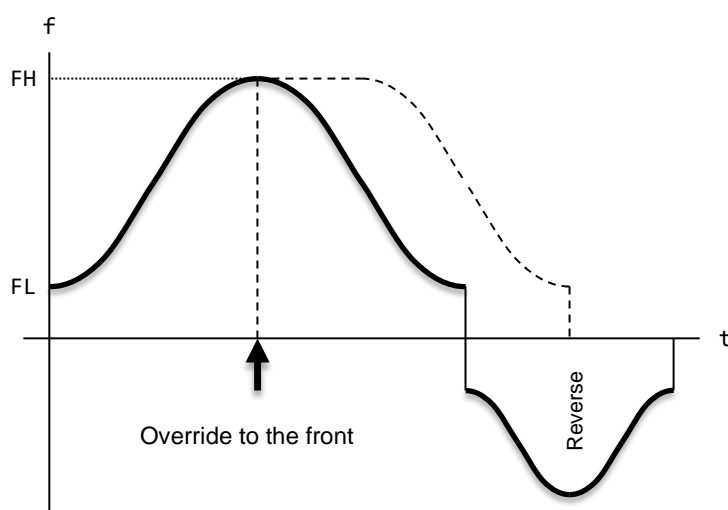
1. If you override the new target position farther than the initial target position during acceleration or constant speed, the operation with the same speed pattern will be maintained, and the operation mode is completed at the new target position.



2. If you override the new target position farther than the initial target position during deceleration, the operation mode will be completed at the new target position after re-accelerating from that position to FH speed. If the current speed at the time of change is F_u speed, the re-acceleration curve will be the same as the normal acceleration curve of $FL = F_u$.



3. If you override a new target position short of the initial target position while passing through the new target position or during deacceleration, the operation will be reversed after decelerate-stops and completes the operation mode at the new target position.



You can change the target position by writing a new target position to RMV register.

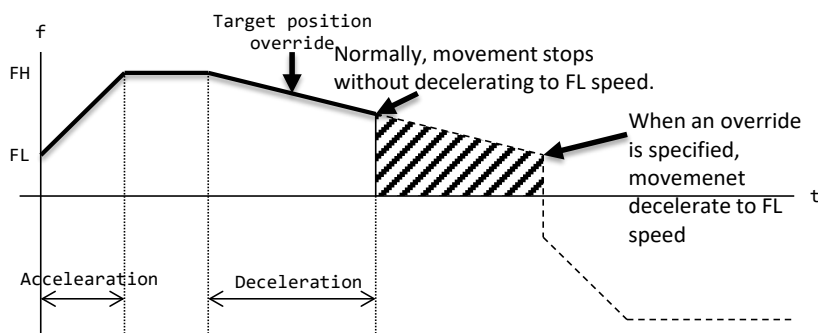
If the operation mode is an incremental position of positioning control (RMD.MOD = 41h), the new target position will be the incremental position from the start.

For example, if the target position is overridden RMV = 200 when RPLS = 50 during running at RMV = 100, it will be recalculated to RPLS = 150, neither overwritten (RPLS = 200) nor added (RPLS = 250).

You can re-write the target position (RMV) as many times as you like until the operation mode is completed.

When accelerating or decelerating, setting the slow-down point automatically (RMD.MSDP = 0) causes a cumulative error in the RSDC register value.

If deceleration time > acceleration time x 2, deceleration to FL speed may not be possible as shown in the figure below.



During deceleration due to RMD.MSDP = 0, the new target position can be overridden before the initial target position.

In this case, continues decelerating to FL speed as shown by the broken line in the above figure.

When the operation reaches FL speed, it reverses the direction and positions at the new target position.

Therefore, overrun of deceleration stop (shaded part in the above figure) occurs against to the initial target position.

To avoid this overrun, use it within the range where the deceleration time does not exceed twice the acceleration time. If it exceeds twice the acceleration time, use the manual slow-down point setting (RMD.MSDP = 1).

During deceleration with RPLS < RSDC, the new target position can be overridden before the initial target position.

In this case, after decelerating and stopping, the operation reverses and positions at a new target position.

During deceleration for this deceleration stop, deceleration continues to FL speed even if the new target position is overridden far away again. When the operation reaches the FL speed, it accelerates again and positions it at the new target position.

During deceleration with RPLS < RSDC, the new target position can be overridden farther than the initial target position.

In this case, immediately re-accelerate and position at the new target position.

The target position override is effective only during operation (FL constant speed, FH constant speed, acceleration, deceleration, backlash correction). If you override just before stopping, the override may not be accepted.

If the target position override is ignored, it will be set to stop outside the target position (MSTS.SEOR = 1).

This happens when writing to RMV register in the stopped status after completing the operation mode.

In the operation mode where the target position can be overridden, it occurs even before the operation mode start.

MSTS.SEOR bit can be reset to MSTS.SEOR = 0 with the SEORR (2Eh) command. If RENV5.MSMR = 0 is set, reading the main status will also reset to MSTS.SEOR = 0.

R e m a r k s

When shifting the override data by writing PRESHF (2Bh) command or when the condition of Comparator 5 is satisfied, the following registers are overridden.

- Speed control registers (RFL, RFH, RUR, RDR, RMG, RUS, RDS)
- Position control registers (RMV, RDP, RIP, RCI)
- Environment setting register (RMD)

Overriding the position control register also changes RPLS register.

6.4.2 Target position override 2 (PCS)

Target position override 2 (RMD.MPCS = 1) can be used in the operation modes of RMD.MOD = 41h, 42h and 43h.

Do not set the target position override 2 in other modes of operation.

The minimum pulse width of PCS signal requires 2 cycles (0.1 μ s) of the CLK signal.

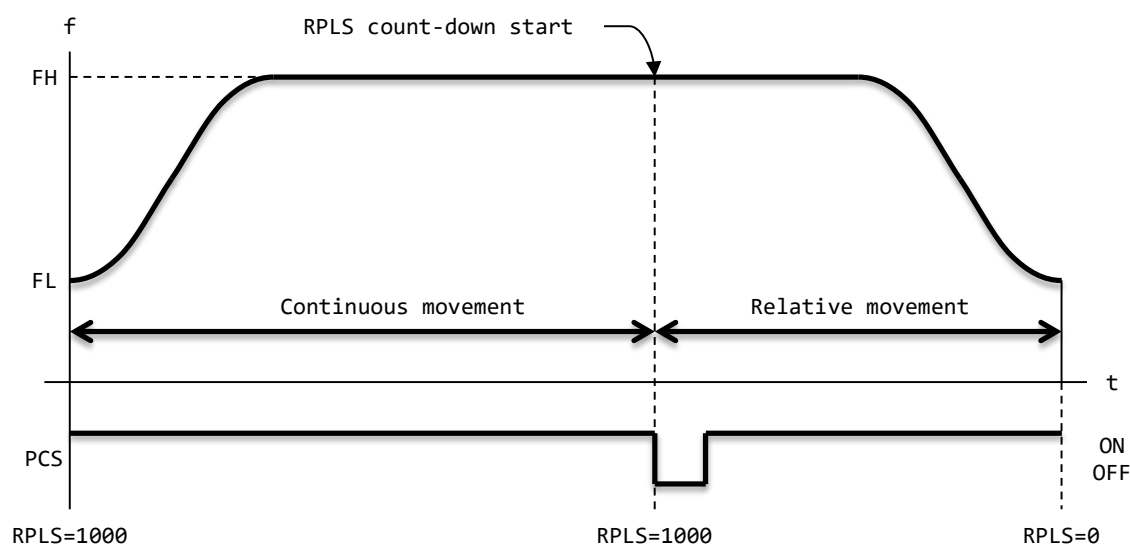
Target position override 2 operation works when you set RENV1.PCSM = 0 and RMD.MPCS = 1 to start the operation mode.

When the operation mode starts, it moves like the continuous operation mode, but RPLS register does not count down.

RPLS register counts down from the position where PCS signal is changed to ON, and each operation mode starts.

PCS signal is sampled in synchronization with the CLK signal the operation mode starts.

If PCS signal is already ON before the operation mode starts, RPLS register counts down immediately after the start.



The input logic of PCS signal can be changed with RENV1.PCSL bit.

The input status of PCS signal can be read by RSTS.SPCS bit.

Name and description	Target
<PCS _n pin input function> 0: General-purpose input pin 1: Input pin for PCS signal for target position override 2	RMD.MPCS(14)
<PCS signal input logic> 0: Negative logic. 1: Positive logic.	RENV1.PCSL(24)
<PCS signal input status> 0: OFF 1: ON	RSTS.SPCS(8)
<PCS signal input substitution> Instead of inputting PCS signal to PCS _n pin, a positioning control can be started.	STAON(28h)

6.4.3 End point draw operation

If you set the end point outside the arc in circular interpolation operation, you can move to the end point in a straight line when the circular interpolation ends. This operation is called "End point draw operation."

The coordinates of the end point are not on the arc, except for the arc angle, which is an integral multiple of 90-degree.

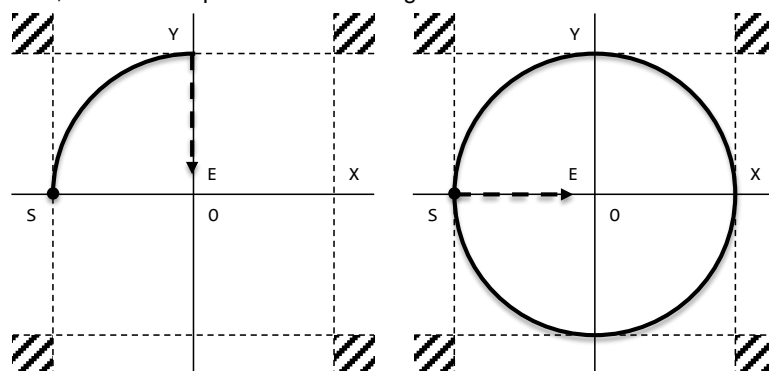
Therefore, when the end point draw operation is disabled (RMD.MPIE = 0), the specified end point is not reached.

In order to reduce the cumulative error of misalignment, use it with the end point draw operation enabled (RMD.MPIE = 1).

If the end point draw operation is disabled (RMD.MPIE = 0) and the end point is not reached, the interpolation axis will be RPLS > 0. Check the operating direction of insufficient number of pulses in RCUN1 register or RCUN2 register.

In the end point draw operation, the interpolation operation is completed when one axis reaches the end point in the end point quadrant, and the other axis also moves to the end point. The speed of the end point draw operation is the same as the speed of the circular interpolation operation. When enabling the end point pull operation (RMD.MPIE = 1), add the number of end point draw pulses to the RCI register value.

If the end point is located directly above the coordinate axis, the next quadrant of the coordinate axis where the end point exists is determined as the end point quadrant, and the end point draw operation starts. Therefore, a circle is drawn right above the axis where the end point exists, and the end point is drawn along the axis.



(The thick dashed line is the end point draw operation when RMD.MPIE = 1)

In other cases, an arc is drawn by setting of RMD.MIPM bit, and the end point draw operation is performed from the arc to the end point coordinates.

For the trajectory by setting the RMD.MIPM bit, see "5.5.9.1 Circular interpolation in CW direction (64h)".

Name and description	Target
<End point draw function> 0: No end point draw function is performed in the circular interpolation and stops on the arc. 1: End point draw function is performed in the circular interpolation and moves up to the end point.	RMD.MPIE(27)
<Circular interpolation completion condition> 0: Determines the end point coordinates of the completion condition in 90-degree increment. If the start and end points are in the same quadrant, operates so that the arc to be shorter. 1: Determines the end point coordinates of the completion condition in 45-degree increment. If the start and end points are in the same quadrant, operates so that the arc to be shorter.	RMD.MIPM(28)

6.5 Output pulse control

You can select the output pulse mode, output pulse width control, and the operation mode completion timing.

6.5.1 Output pulse mode

The output pulse mode can be selected with RENV1.PMD bit according to the input format of a motor driver.

There are 4 types of common pulse modes, Two types of 2-pulse modes, and two types of 90-degree phase difference modes.

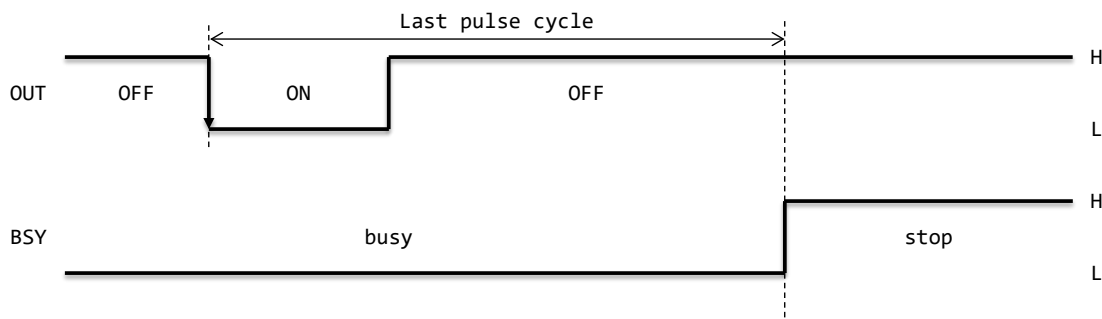
- Common pulse mode (OUT, DIR): Output pulse signals (OUT) and direction signals (DIR) are output. (RENV1.PMD=000b to 011b)
- 2-pulse mode (PLS, MNS): Plus direction pulse signals (PLS) and minus direction pulse signal (MNS) are output. (RENV1.PMD=100b, 111b)
- 90-degree phase difference mode (PHA, PHB): A-phase pulse signal (PHA) and B-phase pulse signal (PHB) with 90-degree phase difference are output. (RENV1.PMD=101b, 110b)

Name and description					Target
<Output pulse mode>					RENV1.PMD(2:0)
PMD	+direction		-direction		
	OUT (PLS)	DIR (MNS)	OUT (PLS)	DIR (MNS)	
000		High		Low	
001		High		Low	
010		Low		High	
011		Low		High	
100		High	High		
101					
110					
111		Low	Low		
<p>※ Both edges of 90-degree phase difference signals (PHA, PHB) in 101b and 110b are valid. The 90-degree phase difference signal outputs 4 pulses in one cycle.</p>					RENV1.DTMF(28)
<p>Set the direction change timer time.</p> <p>0: If RENV1.PMD = 000b to 011b, wait 0.2 ms for pulse output after changing directions.</p> <p>1: If RENV1.PMD = 000b to 011b, wait 0.5 μs for pulse output after changing directions.</p>					

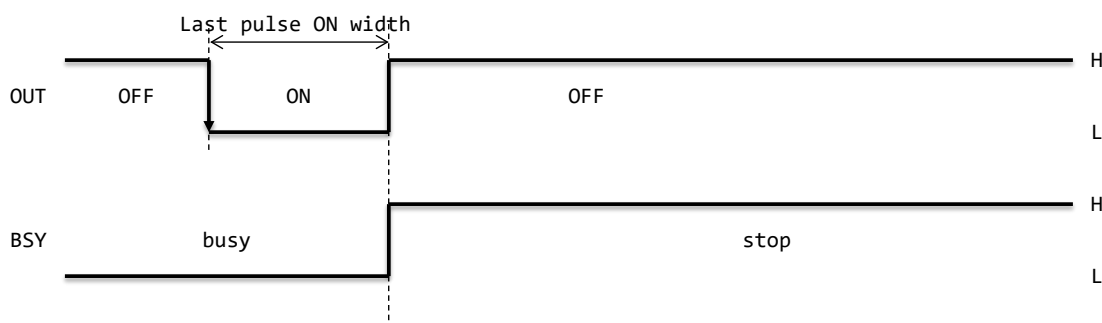
6.5.2 Operation complete timing

By setting the final pulse ON width completed (RMD.METM = 1), the operation mode can be completed without waiting for the completion of the final pulse cycle.

1. Final pulse cycle completed (RMD.METM = 0)



2. Final pulse ON width completed (RMD.METM = 1)



If RMD.METM = 0, the first pulse of the next block is output $10 \times T_{CLK}$ after the completion of an operation mode.

If RMD.METM = 1, the first pulse of the next block is output after a minimum of $15 \times T_{CLK}$ from the completion of an operation mode. T_{CLK} is one reference clock cycle.

Name and description	Target
<Operation completion timing> 0: Output pulse cycle completed. 1: Output pulse ON width completed The operation mode completion timing is advanced by the OFF width of the final pulse. When using the vibration suppression function, set the output pulse cycle completion (RMD.METM = 0). Also set RMD.METM = 0 when using pre-registered continuous operation.	RMD.METM(12)
<Operating signal> 0: Stopping: BSYn pin outputs H level. 1: Operating: BSYn pin outputs L level.	MSTS.SRUN(1)

6.5.3 Output pulse width control

When the output speed of a command pulse is $\frac{f_{CLK}}{8192}$ (2.4 kpps) or less, the output pulse width is fixed narrowly to $f_{CLK} \times 4096$ (0.2 ms). If it is more than this, the output pulse width will fluctuate with a duty ratio of 50%.

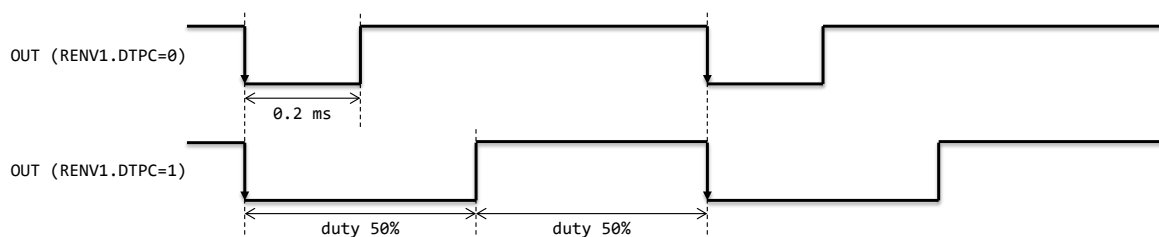
If you set the operation completion timing to the final pulse ON width completed (RMD.METM = 1), the operation mode can be completed even faster.

For example, when accelerating from 100 pps to 5 Kpps, the output pulse width is constant at 0.2 ms from 100 pps to 2.4 Kpps. During acceleration, only the pulse period changes narrowly.

Above 2.4 Kpps, the duty is 50% and the output pulse width exceeds 0.2 ms.

After that, the duty remains at 50% and accelerates to 5 Kpps.

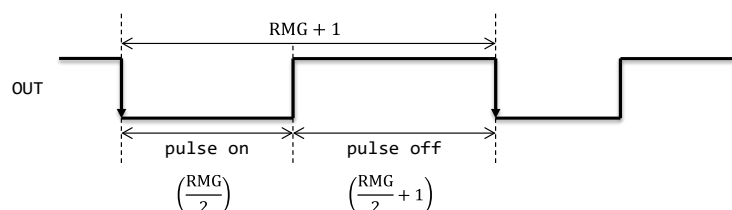
When the output pulse width control is disabled (RENV1.PDTC = 1), the output pulse width fluctuates with a duty of 50% even at low speeds.



If the RMG register value is even, the following error will occur even if the output pulse width is set to a duty ratio of 50%.

$$ON\ time: OFF\ time = \frac{RMG}{2} : \frac{RMG}{2} + 1$$

For example, when RMG = 14 (Eh), ON time: OFF time = 7: 8, and OFF time becomes longer.



I m p o r t a n t

When outputting a maximum speed of 6.5 Mpps at 100x magnification (RMG = 2), the pulse-ON-time is approximately 50 ns and the pulse-OFF-time is approximately 100 ns.

Therefore, the frequency response of the external interface circuit should be at least 10 MHz (pulse period: 2 x 50 ns).

Name and description	Target
<Output pulse width control> 0: When the output speed of a command pulse is 2.4 Kpps or less, the output pulse width is fixed at 0.2 ms. 1: The output pulse width fluctuates with a duty ratio of 50% regardless of the output speed of a command pulse.	RENV1.PDTC(31)
<Speed magnification> For details, see "5.4.1.5 RMG(PRMG): Speed magnification".	RMG

6.6 Idling control

At the start of acceleration of a stepping motor, the first few pulses can be output at FL speed and then acceleration can start.

The pulse outputting is called idling pulses, and the occurrence of step-out can be reduced.

Set the number of idling pulses to RENV5.IDL bit.

When the RENV5.IDL bit value n is set to 0, acceleration starts at the same time as a command pulse is output.

If acceleration starts from the 0th pulse, the 1st pulse is faster than FL speed, so it is shorter than the pulse cycle of FL speed.

If the RENV5.IDL bit value n is set to 0, it appeared to be $n = 1$.

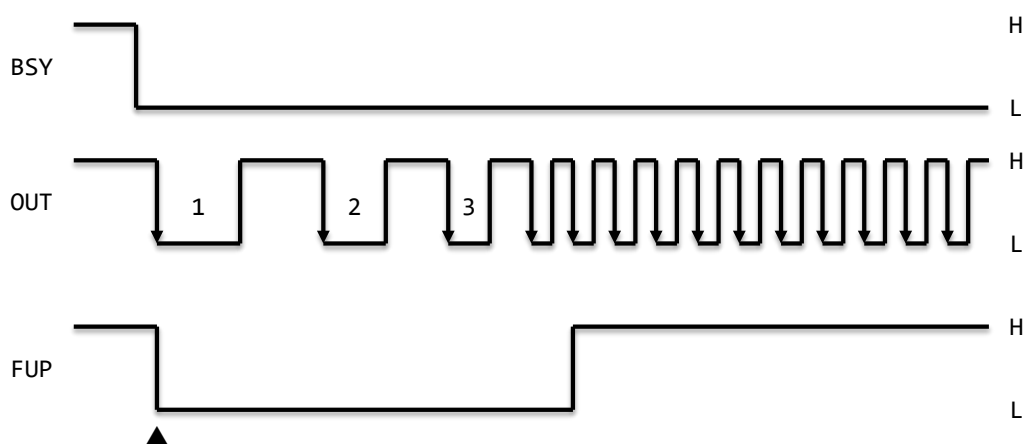
However, if $n > 0$, acceleration will start from FL speed even if you start with STAD (52h) command.

If the setting value n of RENV5.IDL bit is set to 1 or more, acceleration starts at the timing when n th pulse is output.

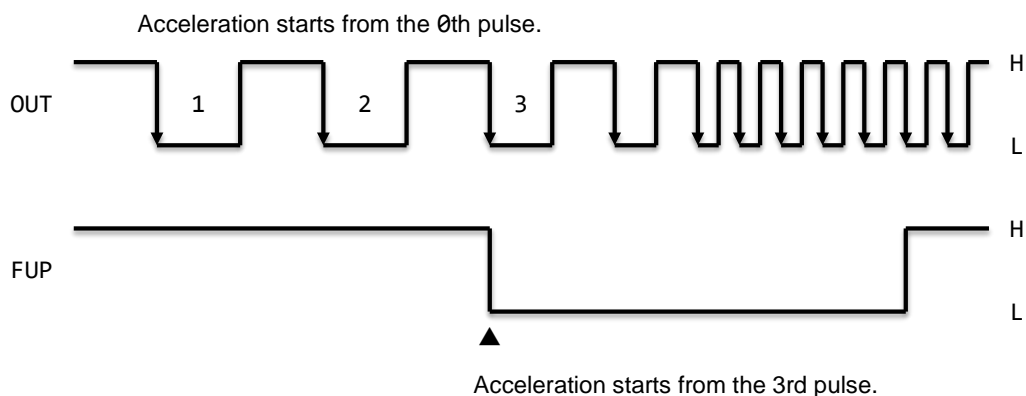
Since acceleration starts from the n th pulse, if $n > 1$, the first $n-1$ pulse becomes FL speed.

[Idling pulse setting value and acceleration start timing]

When $n=0,1$



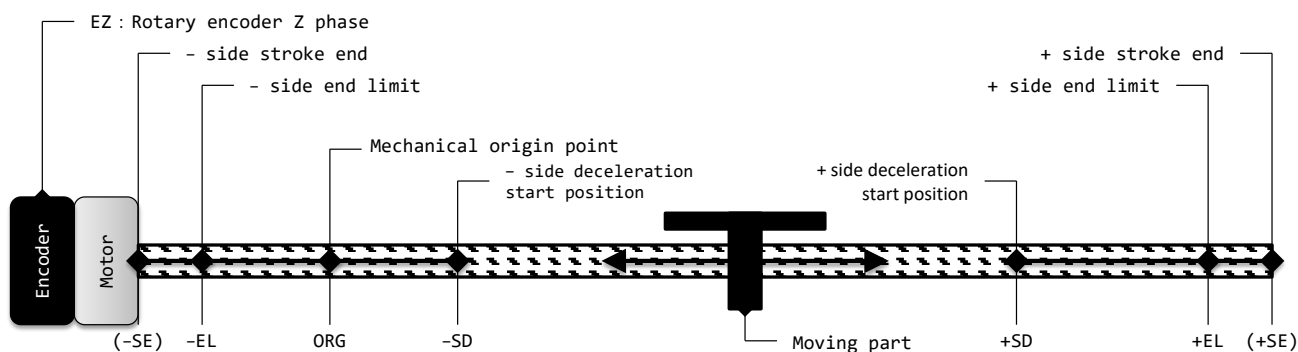
When $n=3$



Name and description	Target
<Number of idling pulse outputs> 000b: No idling pulse is output. 001b to 111b: 0 to 6 pulses are output.	RENV5.IDL(10:8)
<Idling count value> Down counter. The initial value is the value of RENV5.IDL bit.	RSPD.IDC(22:20)

6.7 Mechanical external input control

In addition to the termination switch (+ELn, -ELn), origin switch (ORGn), and deceleration switch (+SDn, -SDn) that are assembled in an actuator like a slider in the figure below, the Z-phase (EZn) output of a rotary encoder can be used as an external input trigger to perform a various controls.



6.7.1 End limit (+EL, -EL)

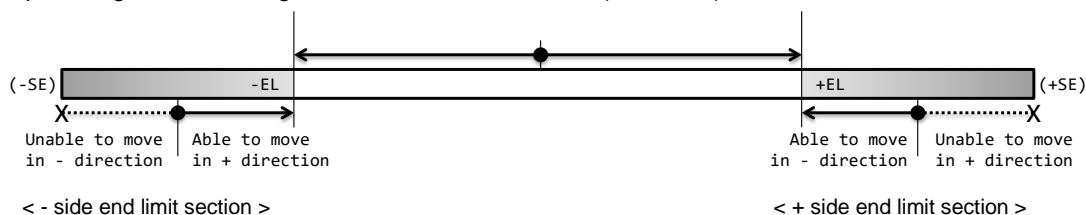
+EL signal is ON when operating in the +direction, and the -EL signal is ON when operating in the -direction, resulting in an abnormal stop. In the operation mode of RMD.MOD = 47h (timer), it does not stop abnormally.

The stop method (RENV1.ELM) can be selected from immediate stop or deceleration stop.

If you select deceleration stop, the operation will stop after passing through +EL position or -EL position.

If +EL signal is ON, it will not start in the +direction, and if the -EL signal is ON, it will not start in the -direction.

For safety, keep +EL signal and -EL signal ON until each stroke end (+SE, -SE).

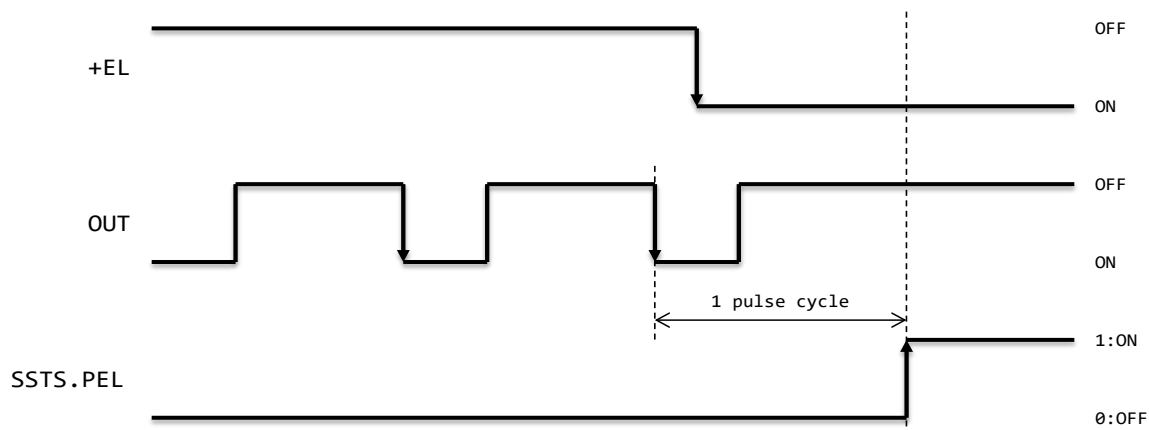


The Input logics (ELLn pin) for +EL signal and -EL signal can be selected.

The input noise filter (RENV1.FLTR) can also be selected for +EL signal and -EL signal.

The abnormal stop due to +EL and -EL signals can be read by the error interrupt factor (REST.ESPL, ESML).

The input status of +EL and -EL signals can be read by sub-status (SSTS.SPEL, SMEL).



Some operation modes in origin return control (RMD.MOD = 10h, 18h, 15h, 1Dh) may not stop abnormally.

Even some operation modes in sensor controls (RMD.MOD = 20h, 28h, 22h, 2Ah) may not stop abnormally.

See the description of each operation mode.

Name and description	Target
<Input logic of +EL signal and -EL signal> L: Positive logic. H: Negative logic.	ELLn pin
<Input processing of +EL signal and -EL signal> 0: EL signal in the operating direction turns ON to stop immediately. 1: EL signal in the operating direction turns ON to decelerate-stop.	REMV1.ELM(3)
<Input noise filter for +EL, -EL, +SD, -SD, ORG, ALM, INP, CEMG signals > 0: Signal with the pulse width of 0.05 μ s or more reacts reliably. 1: Ignore signals with the width of 3 μ s or less completely.	RENV1.FLTR(26)
<Error interrupt factor (+EL)> 1: +EL signal turns ON to stop abnormally.	REST.ESPL(5)
<Error interrupt factor (-EL)> 1: - EL signal turns ON to stop abnormally.	REST.ESML(6)
<+EL signal input status> 0: OFF. 1: ON.	SSTS.SPEL(12)
<-EL signal input status> 0: OFF. 1: ON.	SSTS.SMEL(13)

6.7.2 Slow-down (+SD, -SD)

The input function (RMD.MSDE) of +SD and -SD pins can be selected.

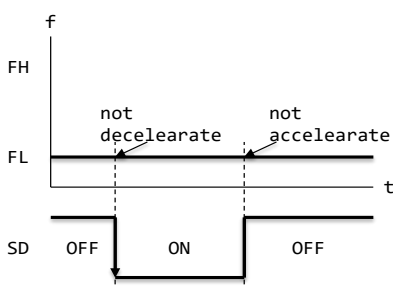
If you set RMD.MSDE = 1, +SD and -SD signals will be enabled during operation.

You can select (1) deceleration, (2) latch & deceleration, (3) deceleration stop, and (4) latch & deceleration stop, with RENV1.SDM bit and RENV1.SDLT bit.

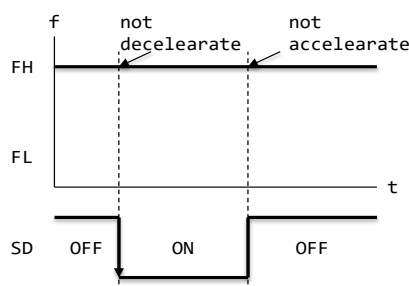
(1) Deceleration <RENV1.SDM = 0, RENV1.SDLT = 0>

- In FL and FH constant speed patterns, +SD and -SD signals are ignored.
- In high-speed 1 and 2 speed patterns, if SD signal in the operating direction turns ON, the operation decelerates to FL speed. In this case, after or during deceleration, if SD signal in the operating direction turns OFF, it accelerates to FH speed.
- If SD signal in the operating direction turns ON if writing STAD (52h) or STAUD (53h) command, the machine operates at FL speed. In this case, when SD signal in the operating direction turns OFF, it accelerates to FH speed.

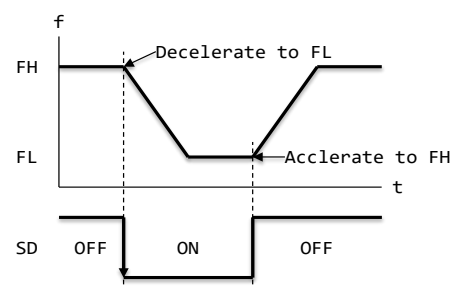
STAFH(51h) command:



STAFH(51h) command:



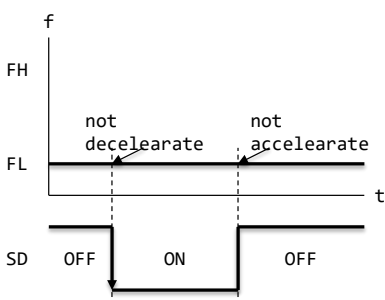
STAD(52h),STAUD(53h) command:



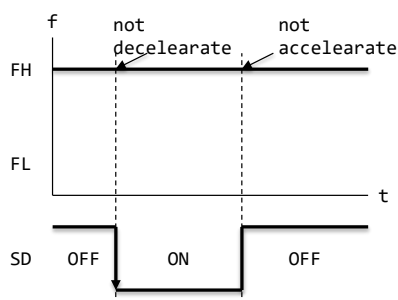
(2) Latch & deceleration <RENV1.SDM = 0, RENV1.SDLT = 1>

- In FL and FH constant speed patterns, +SD and -SD signals are ignored.
- In high-speed 1 and 2 speed patterns, if SD signal in the operating direction is ON, the operation decelerates to FL speed. In this case, even if SD signal in the operating direction turns OFF after or during deceleration, it will not accelerate.
- If SD signal in the operating direction turns ON when writing STAD (52h) or STAUD (53h) command, the machine operates at FL speed. In this case, even if SD signal in the operating direction turns OFF, it will not accelerate to FH speed.

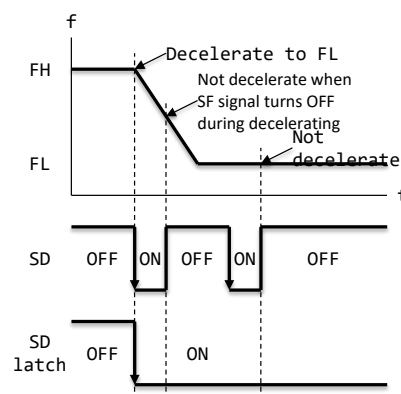
STAFH(51h) command:



STAFH(51h) command:



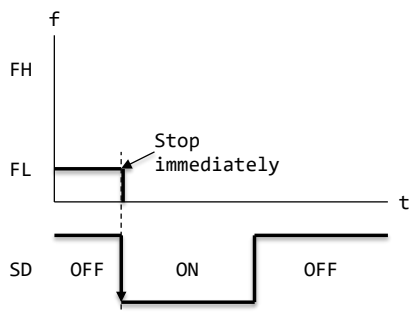
STAD(52h),STAUD(53h) command:



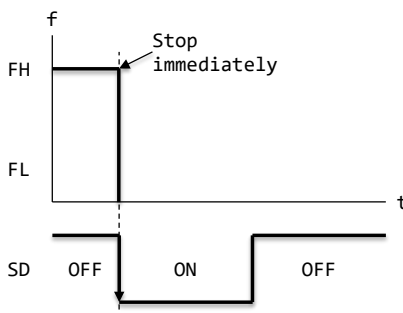
(3) Deceleration stop <RENV1.SDM = 1, RENV1.SDLT = 0>

- In FL and FH constant speed patterns, if SD signal in the operating direction turns ON, the operation stops immediately.
- In high-speed 1 and 2 speed patterns, if SD signal in the operating direction turns ON, the operation decelerates to FL speed and stops.
- In this case, if SD signal in the operating direction turns OFF during deceleration, it accelerates to FH speed.
- If SD signal in the operation direction turns ON when writing a start command, the operation mode is completed without starting.
- If the operation stops by turning ON the SD signal in the operating direction, an error interrupt (REST.ESSD) is generated at the time of stop.

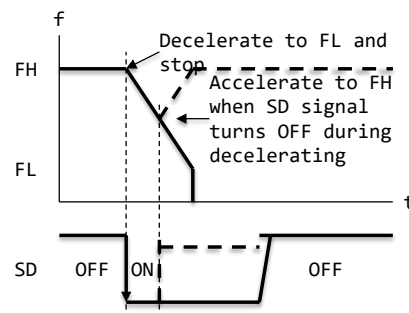
STAFH(50h) command:



STAFH(51h) command:



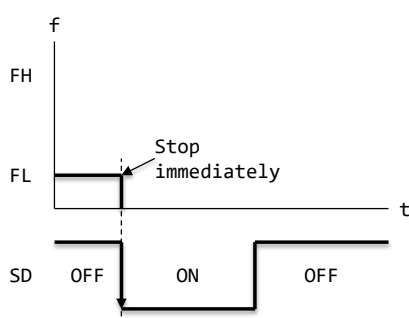
STAD(52h),STAUD(53h) command:



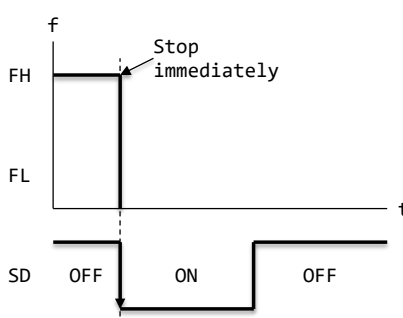
(4) Latch & deceleration stop <RENV1.SDM = 1, RENV1.SDLT = 1>

- In FL and FH constant speed patterns, if SD signal in the operating direction turns ON, the operation stops immediately.
- In high-speed 1 and 2 speed patterns, if SD signal in the operating direction turns ON, the operation decelerates to FL speed and stops. In this case, even if slow-down signal turns OFF during deceleration, the operation will not accelerate.
- If SD signal in the operation direction turns ON when writing the start command, the operation mode is completed without starting.
- If the operation stops by turning ON the SD signal in the operating direction, an error interrupt (REST.ESSD) is generated at the time of stop.

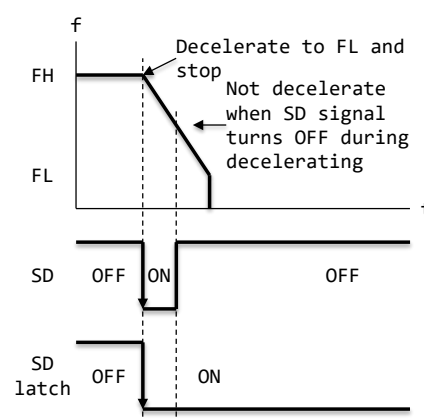
STAFH(50h) command:



STAFH(51h) command:



STAD(52h),STAUD(53h) command:



The Input logic (RENV1.SDL) of +SD and -SD signals can be selected.

The latch status of SD signal in the operating direction can be read by the sub status (SSTS.SSD).

The input latch function (RENV1.SDLT) of +SD and -SD signals can be selected.

If RENV1.SDLT = 1 is set, SSTS.SSD = 1 will be set when SD signal in the operating direction turns ON.

If RENV1.SDLT = 0 is set, or if SD signal in the operating direction is OFF when starting, it returns to SSTS.SSD = 0.

The Input noise filter (RENV1.FLTR) of +SD and -SD signals can be selected.

Abnormal stop due to +SD and -SD signals can be read by the error interrupt factor (REST.ESSD).

The input status of +SD and -SD signals can be read by the extended status (RSTS.PSDI, RSTS.MSDI).

The latch status of +SD and -SD signals can be read by the extended status (RSTS.PSDL, RSTS.MSDL).

Name and description	Target
<Input function of +SDn pin and -SDn pin > 0: General-purpose input pin 1: SD signal in the operating direction turns ON for deceleration or deceleration stop.	RMD.MSDE(8)
<+SD and -SD signals input processing> 0: SD signal in the operating direction turns ON, the operation will decelerate. 1: SD signal in the operating direction turns ON, the operation will decelerate-stop.	RENV1.SDM(4)
<+SD and -SD signals input latch function> 0: Not latch SD signal in the operating direction. 1: Latch SD signal in the operating direction.	RENV1.SDLT(5)
<+SD and -SD signals input logic> 0: Negative logic. 1: Positive logic.	RENV1.SDL(6)
< Input noise filters for +EL, -EL, +SD, -SD, ORG, ALM, INP and CEMG signals > 0: Signals with the pulse width of 0.05 μ s or more will react reliably. 1: Ignore signals with the pulse width of 3 μ s or less completely.	RENV1.FLTR(26)
<Error interrupt factor (+SD, -SD)> 1: +SD signal or the -SD signal turns ON to stop abnormally.	REST.ESSD(10)
<Latch status of SD signal in the operating direction> 0: OFF 1: ON	SSTS.SSD(15)
<+SD signal input status> 0: OFF 1: ON	RSTS.PSDI(15)
<-SD signal input status> 0: OFF 1: ON	RSTS.MSDI(17)
<+SD signal latch status> 0: OFF 1: ON	RSTS.PSDL(22)
<-SD signal latch status> 0: OFF 1: ON	RSTS.MSDL(23)

6.7.3 Origin (ORG), Encoder Z phase (EZ)

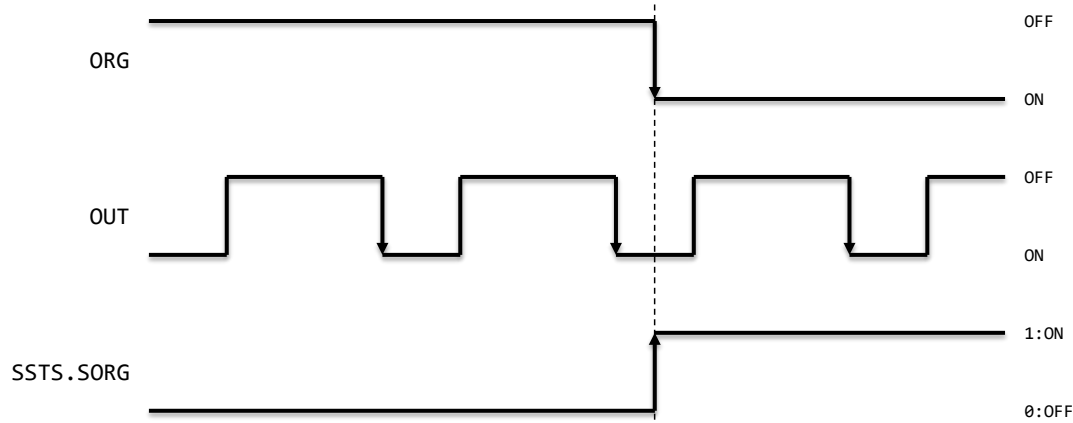
ORG signal and EZ signal are used in the origin return control mode.

EZ signal is also used in some operation modes in sensor controls (RMD.MOD = 24h, 2Ch).

The input logic (RENV1.ORGL) of ORG signal can be selected.

The input noise filter (RENV1.FLTR) of ORG signal can also be selected.

The input status of ORG signal can be read by sub status (SSTS.SORG).



SSTS.SORG bit changes after passing through the input noise filter.

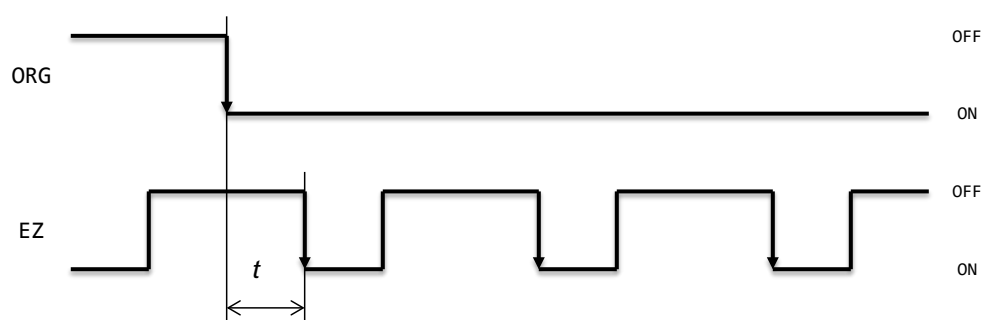
The input logic (RENV2.EZL) of EZ signal can be selected.

The input noise filter (RENV2.EINF) of EZ signal can also be selected.

The default value (RENV3.EZD) of the down-count of EZ signal can be set, and the count value (RSPD.EZC) can be read.

The input status of EZ signal can be read by the extended status (RSTS.SEZ bit).

Whether to count the first EZ signal ON after ORG signal ON will depend on the t time in the figure below.



(RMD.MOD=10h, RENV1.ORGL=0, RENV1.FLTR=0, RENV2.EZL=0, RENV2.EINF=0, RENV3.ORM=0010b)

- ① When $2 \times T_{CLK} \leq t$, counts.
- ② When $T_{CLK} < t < 2 \times T_{CLK}$, counting is determined.
- ③ When $t \leq T_{CLK}$, does not count.

T_{CLK} : Reference clock cycle

Name and description	Target
<p><Operation mode using ORG signal or EZ signal></p> <p>10h: Operation mode for origin return in the + direction in origin return control.</p> <p>18h: Operation mode for origin return in the – direction in origin return control.</p> <p>12h: Operation mode to escape from the origin position in the + direction by origin return control.</p> <p>1Ah: Operation mode to escape from the origin position in the –direction by origin return control.</p> <p>15h: Operation mode for origin search in the + direction by origin return control.</p> <p>1Dh: Operation mode for origin search in the – direction by origin return control.</p> <p>24h: Operation mode of moves by EZ count in the + direction by origin return control.</p> <p>2Ch: Operation mode of moves by EZ count in the – direction by origin return control.</p>	RMD.MOD(6:0)
<p><ORG signal input logic></p> <p>0: Negative logic.</p> <p>1: Positive logic.</p>	RENV1.ORGL(7)
<p>< Input noise filters of + EL, –EL, +SD, –SD, ORG, ALM, INP and CEMG signals ></p> <p>0: Signals with the pulse width of 0.05 μs or more will react reliably.</p> <p>1: Ignore signals with the pulse width of 3 μs or less completely.</p>	RENV1.FLTR(26)
<p><Input noise filter of EA, EB and EZ signals></p> <p>0: Signals with the pulse width of 0.05 μs or more will react reliably.</p> <p>1: Signals with the pulse width of 0.15 μs or more will react reliably.</p>	RENV2.EINF(18)
<p><EZ signal input logic></p> <p>0: Negative logic</p> <p>1: Positive logic</p> <p>EZ signal counts from OFF to ON.</p>	RENV2.EZL(23)
<p><Origin return method></p> <p>See "5.4.3.4 RENV3: Environment setting 3".</p>	RENV3.ORM(3:0)
<p><Initial value of EZ signal input count></p> <p>0000b (1 time) to 1111b (16 times).</p>	RENV3.EZD(7: 4)
<p><EZ signal input count value></p> <p>The initial value is in RENV3.EZD bit.</p>	RSPD.EZC(19: 16)
<p><ORG signal input status></p> <p>0: OFF</p> <p>1: ON</p>	SSTS.SORG(14)
<p><EZ signal input status></p> <p>0: OFF</p> <p>1: ON</p>	RSTS.SEZ(10)

6.8 Servo motor interface

You can connect dedicated signals to a servo motor driver.

The dedicated signals are: positioning complete output (INP), deviation counter clear input (ERC) and alarm output (ALM).

You can use the signals to perform various controls.

6.8.1 Positioning complete (INP)

Pulse-train input type servo motor drivers have the command pulse input and a feedback pulse input. In the servo motor drivers, there is a deviation counter that counts the difference of the inputs. The drivers keep a motor running until the deviation counter reaches 0, even if the command pulse is stopped. When the absolute value of the deviation counter becomes less than the set value, the servo motor driver outputs an INP signal.

When waiting for INP signal input (RMD.MINP = 1) is set, the operation mode completion timing is delayed until INP signal is input. In this case, the following changes will also be delayed when INP signal is input.

- Main status stop conditions (MSTS.SSCM, SRUN, SENI, SEND, SERR and SINT)
- Extended status operating status (RSTS.CND)

However, abnormal and emergency stops (CEMG, CEMEG) due to ALM signals complete the operation mode without being affected by this delay.

The input logic (RENV1.INPL) of INP signal can be selected.

The input noise filter (RENV1.FLTR) of INP signal can also be selected.

If INP signal is already ON when the pulse output is completed, the operation mode is completed without delay.

The input status of INP signal can be read by the extended status (RSTS.SINP).

Name and description	Target
<Input function of INPn pin > 0: General-purpose input 1: INP signal is ON and the operation mode is completed.	RMD.MINP(9)
<INP signal input logic> 0: Negative logic 1: Positive logic	RENV1.INPL(22)
< Input noise filters for +EL, -EL, +SD, -SD, ORG, ALM, INP, CEMG signal > 0: Signals with the pulse width of 0.05 μs or more will react reliably. 1: Ignore signals with the pulse width of 3 μs or less completely.	RENV1.FLTR(26)
<INP signal input status> 0: OFF 1: ON	RSTS.SINP(16)

6.8.2 Deviation counter clear (ERC)

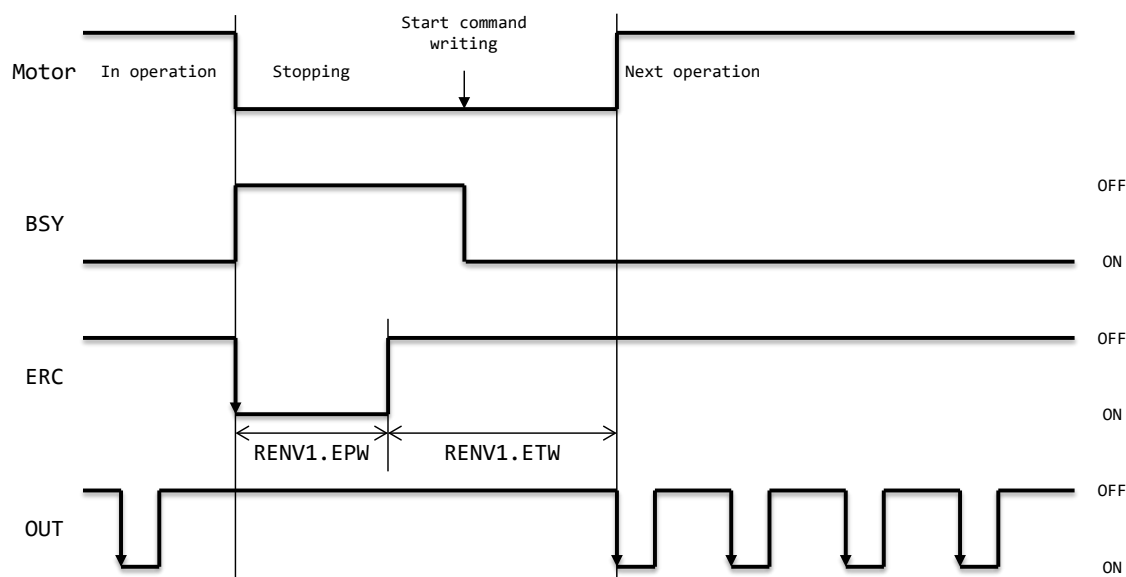
Servo motor drivers do not stop servo controls until the deviation counter reaches 0. Therefore, even if the command pulse outputs stop, servo motors do not stop immediately. To stop the servo motors immediately when an origin return operation is completed, it is necessary to clear this deviation counter to 0. For this, an ERC signal can be output.

The ON width (RENV1.EPW) of an ERC signal can be selected.

If level signal (RENV1.EPW = 111b) is set for the ON width of an ERC signal, turn it OFF with ERCCRST (25h) command.

When the ERC signal is ON, the servo motor driver control will be OFF, so the servo motor may rotate slightly. Therefore, be sure to change the level of the ERC signal to OFF by using ERCCRST (25h) command.

Some servo motor drivers take a long time to receive the next command pulse after an ERC signal is changed to OFF. In this case, you can select the OFF width (RENV1.ETW) of the ERC signal.



If RENV1.EROR = 1 is set, the ERC signal can be automatically output when the origin return is completed.

For the timing to output the ERC signal, see "5.5.5.1 Origin return in +direction (10h)".

If RENV1.EROE = 1 is set, the ERC signal can be automatically output when an abnormal stop occurs. When decelerate-stopped, the ERC signal is not output.

The target abnormal stop is by the input of +EL, -EL, ALM, and CEMG signals and CMEMG (05h) command writing. Even if RMD.MOD = 20h and 28h, the ERC signal is output when +EL signal and -EL signal are changed to ON and the operation stops immediately.

The ERC signal can be output arbitrarily by writing ERCOUT (24h) command. If the ERC signal is output during an abnormal stop or by using ERCOUT (24h) command, an error will occur between the command position and the mechanical position. Therefore, check the ERC signal OFF before performing an origin return operation.

The output logic (RENV1.ERCL) of the ERC signal can be selected. The output status of the ERC signal can be read by the extended status (RSTS.SERC).

Name and description	Target
<p><ERCn pin output function at the time of immediate stop due to an abnormal stop factor></p> <p>0: Does not output the ERC signal. 1: Outputs the ERC signal.</p>	RENV1.EROE(10)
<p><Output function of ERCn pin when stopped due to origin return factor></p> <p>0: Does not output the ERC signal. 1: Output the ERC signal.</p> <p>See "6.8.2 Deviation counter clear (ERC)" for the ERC signals.</p>	RENV1.EROR(11)
<p><ON width of ERC signal></p> <p>000b: 11 to 13 μs 001b: 91 to 98 μs 010b: 360 to 390 μs 011b: 1.4 to 1.6 ms 100b: 11 to 13 ms 101b: 46 to 50 ms 110b: 93 to 100 ms 111b: Level output</p>	RENV1.EPW(14: 12)
<p><ERC signal input logic></p> <p>0: Negative logic. 1: Positive logic.</p>	RENV1.ERCL(15)
<p><ERC signal OFF width></p> <p>00b: 0 μs 01b: 11 to 13 μs 10b: 1.4 to 1.6 ms 011b: 93 to 100 ms</p>	RENV1.ETW(17,16)
<p><ERC signal output status></p> <p>0: OFF 1: ON</p>	RSTS.SERC(9)
<p><Emergency stop command></p> <p>Emergency stop to complete the operation mode. It also cancels the continuous operation by the pre-register.</p>	CMEMG(05h)
<p><ERC signal output></p> <p>ERC signal is output from ERCn pin.</p>	ERCOUT(24h)
<p><ERC signal reset></p> <p>When ERC signal is output with the setting of RENV1.EPW = 111b, the output of ERC signal will be reset.</p>	ERCRST(25h)

6.8.3 Alarm (ALM)

A servo motor driver outputs the ALM signal when an error occurs.

If the ALM signal from a servo motor driver is input to the ALMn pin during operation, the operation will stop abnormally.

If deceleration stop (RENV1.ALMM = 1) is set in the ALM signal input processing, deceleration stop is performed in high-speed 1 and 2 speed patterns. In this case, do not turn OFF the ALM signal until the operation stops.

When the ALM signal is ON, the operation will not start.

When waiting for INP signal input (RMD.MINP = 1) is set, it will not affect the operation of abnormal stop due to ALM signal.

When an alarm occurs, the operation mode is completed even if a servo motor driver cannot output an INP signal.

See "6.8.1 Positioning complete (INP)" for INP signals.

The input logic (RENV1.ALML) of an ALM signal can be selected.

The input noise filter (RENV1.FLTR) of an ALM signal can also be selected.

Abnormal stop due to an ALM signal can be read by the error interrupt cause (REST.ESAL).

The input status of an ALM signal can be read by the sub status (SSTS.SALM).

Name and description	Target
< ALM signal input processing > 0: Immediate stop 1: Deceleration stop	RENV1.ALMM(8)
< ALM signal input logic > 0: Negative logic 1: Positive logic	RENV1.ALML(9)
< Input noise filter for +EL, -EL, +SD, -SD, ORG, ALM, INP and CEMG signals > 0: A signal with the pulse width of 0.05 μs or more will react reliably. 1: Ignore signals with the pulse width of 3 μs or less completely.	RENV1.FLTR(26)
< Error interrupt factor (ALM) > 1: Operation stopped abnormally when ALM signal turns ON.	REST.ESAL(7)
< ALM signal input status > 0: OFF 1: ON	SSTS.SALM(11)

6.9 External start / Simultaneous start

You can start with an external signal using CSTA and PCSn pins. Also, you can use CSTA pin to start multiple axes at the same time.

6.9.1 Simultaneous start (CSTA)

You can start externally by inputting a one-shot pulse CSTA signal or level signal to CSTA pin.

By connecting the CSTA pins of multiple PCL6046s, the axes of multiple PCL6046s can be started at the same time. If you set CSTA signal input wait (RMD.MSY = 01b), you can wait for CSTA signal input.

The input logic of a CSTA signal cannot be selected and is fixed to negative. The input specifications of a CSTA signal can be selected with RENV1.STAM bit.

When you write a start command, it will be in the CSTA signal input waiting status (RSTS.CND = 0010b).

If RENV1.STAM = 0 is set when CSTA = L level is set from the stopped status, the operation starts when the start command is written. If RENV1.STAM = 1 is set, the operation will start when the falling edge of the CSTA signal is input to CSTA pin.

The input interrupt for a CSTA signal can be set in the event interrupt request (RIRQ.IRSA). This can be read by the event interrupt factor (RIST.ISSA). The input status of the CSTA signal can be read by the extended status (RSTS.SSTA).

You can output a one-shot pulse CSTA signal by writing CMSTA (06h) command. You can start externally by pulling up CSTA pin.

When SPSTA (2Ah) command is written, one-shot pulse CSTA signal will not be output.

Even if the CSTA pin is pulled up, it will not start externally. In this case, a command is written and the axis waiting for CSTA signal input (RSTS.CND = 0010b) starts.

The operation mode can be canceled by writing a stop command in the CSTA signal input waiting status (RSTS.CND = 0010b).

<Simultaneous start procedure>

Pull up the CSTA pins of PCL6046s that start at the same time together.

Set RMD.MSY = 01b for the axes that start at the same time.

Write a start command to put it in the CSTA signal input wait status (RSTS.CND = 0010b).

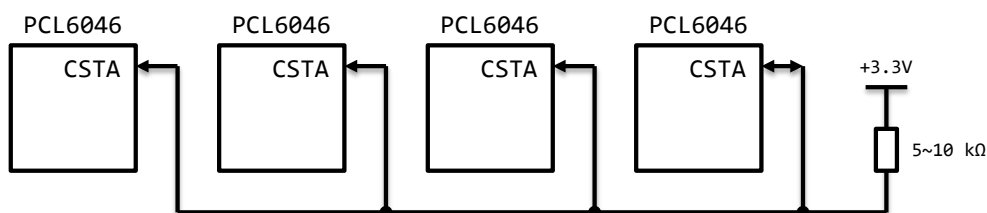
After that, you can start simultaneously by the following two methods.

1. Write the CMSTA (06h) command

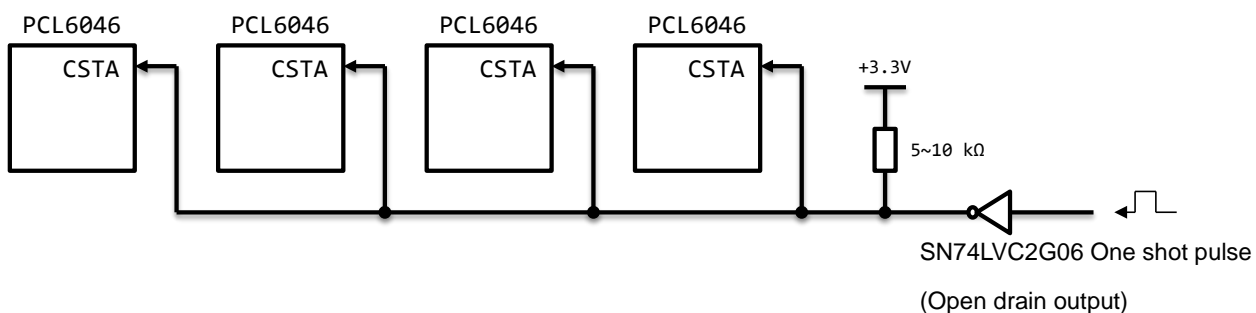
One-shot pulse with the pulse width of 8 cycles (0.4 μ s) of the CLK signal is output from CSTA pin.

All PCL6046s connected by the CSTA pins input a one-shot pulse to start the effective axis.

The PC6046 that outputs the one-shot pulse also re-inputs the one-shot pulse to start the effective axis.



2. Input a one-shot pulse with the pulse width of 4 cycles (0.2 μ s) or more of the CLK signal to CSTA pin. The one-shot pulse is input to all PCL6046s connected by the CSTA pins to start effective axes.



Name and description	Target
<Start timing after writing a start command> 01b: Starts by inputting CSTA signal or the own axis start signal. If RENV1.PCSM = 0, starts with CSTA = L level or SPSTA (2Ah) command. If RENV1.PCSM = 1, starts when STA signal turns ON or starts by the SPSTA (2Ah) command.	RMD.MSY(19,18)
<CSTA signal input specifications> 0: Level trigger 1: Edge trigger (falling edge)	RENV1.STAM(18)
<Interrupt request (IRSA)> 1: An interrupt is generated when CSTA signal turns ON (RENV1.PCSM = 0). An interrupt is also generated when STA signal is changed turns ON (RENV1.PCSM = 1).	RIRQ.IRSA(18)
<Interrupt (ISSA)> 1: CSTA signal turns ON (RENV1.PCSM = 0) or STA signal turns ON (RENV1.PCSM = 1).	RIST.ISSA(19)
<Operating status> 0010b: Waiting for CSTA signal input.	RSTS.CND(3:0)
<CSTA signal input status> 0: OFF 1: ON	RSTS.SSTA(5)
<CSTA signal output> CSTA signal is output from CSTA pin.	CMSTA(06h)
<Own axis start> Does not output CSTA signal from CSTA pin. Operation mode can start while waiting for the input of CSTA signal (RSTS.CND = 0010b).	SPSTA(2Ah)

6.9.2 Own axis start (STA)

You can start externally by inputting a one-shot pulse STA signal to PCSn pin.

By setting CSTA signal input wait (RMD.MSY = 01b) and STA signal (RENV1.PCSM = 1), you can wait for STA signal input.

The input logic of STA signal can be selected with RENV1.PCSL bit.

When you write a start command, the operation will be in the CSTA signal input waiting status (RSTS.CND = 0010b).

The operation starts when STA signal ON is input to PCSn pin.

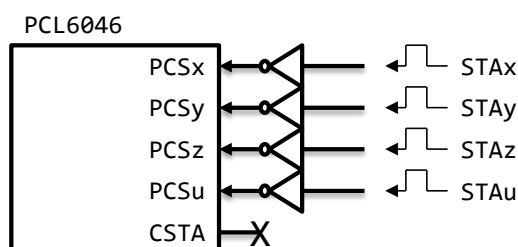
For STA signal, input a one-shot pulse with the pulse width of 4 cycles (0.2 μ s) or more of the CLK signal from to PCSn pin.

At this time, even if a CSTA signal is input to CSTA pin, the operation will not start.

When SPSTA (2Ah) command is written, a one-shot pulse CSTA signal is not output.

Even if the CSTA pin is pulled up, the operation will not start externally.

In this case, the command is written and the axis waiting for CSTA signal input (RSTS.CND = 0010b) will start.



The input interrupt for STA signal can be set with the event interrupt request (RIRQ.IRSA). This can be read by the event interrupt factor (RIST.ISSA). The input status of the STA signal can be read by the extended status (RSTS.SPSCS).

Name and description	Target
<p><Start timing after writing a start command></p> <p>01b: Starts by inputting CSTA signal or own axis start signal.</p> <p>If RENV1.PCSM = 0, starts with CSTA = L level or SPSTA (2Ah) command.</p> <p>If RENV1.PCSM = 1, starts when STA signal turns ON or by SPSTA (2Ah) command.</p>	RMD.MSY(19,18)
<p><STA signal input logic></p> <p>0: Negative logic</p> <p>1: Positive logic</p>	RENV1.PCSL(24)
<p><Input function of CSTA pin and PCSn pin ></p> <p>0: Input of CSTA pin is valid.</p> <p>Setting in RMD.MPCS bit is reflected in PCSn pin.</p> <p>1: Input of CSTA pin is invalid.</p> <p>STA signal to start only own axis is input in PCSn pin.</p>	RENV1.PCSM(30)
<p><Interrupt request (IRSA)></p> <p>1: An interrupt is generated when CSTA signal turns ON (RENV1.PCSM = 0).</p> <p>An interrupt is also generated when STA signal turns ON (RENV1.PCSM = 1).</p>	RIRQ.IRSA(18)

Name and description	Target
<Interrupt factor (ISSA)> 1: CSTA signal turns ON (RENV1.PCSM = 0) or STA signal turns ON (RENV1.PCSM = 1).	RIST.ISSA(19)
<Operation status> 0010b: Waiting for CSTA signal input	RSTS.CND(3:0)
<STA signal input status> 0: OFF 1: ON	RSTS.SPCS(8)
<Original axis start> Does not output CSTA signal from CSTA pin. Operation mode can be started while waiting for the input of CSTA signal (RSTS.CND = 0010b).	SPSTA(2Ah)

6.9.3 Axis selection start (SELn)

PCL6046 can write the same command to multiple axes at the same time using axis selection (SELn).

At this time, if one PCL6046 is used, multiple axes can be started at the same time by writing a start command.

Software example (H8):

```

var Address = 0x00;           // Specify a common (X-axis) command area for the address
var Command = 0x0350;        // Specify X-axis and Y-axis (03h) for axis selection
                              // Specify STAF (50h) command as the command
OutputPCL (Address, Command); // Write the axis selection and a command to PCL6046

```

For details on writing commands, see "5.1.3 Write a command".

6.10 External stop / Simultaneous stop

You can use CSTOP pin to stop immediately or decelerate-stop with external signals.

You can also stop simultaneously using CSTOP pin.

6.10.1 Simultaneous stop (CSTOP)

You can input a one-shot signal to CSTOP pin to stop externally.

By connecting CSTOP pins of multiple PCL6046s, each axis of multiple PCL6046s can be stopped simultaneously.

If you set deceleration stop or immediate stop ($RMD.MSPE = 1$) by inputting CSTOP signal, you can wait for the input of CSTOP signal. The input logic of CSTOP signal cannot be selected and is fixed to negative logic. The input processing of CSTOP signal can be selected with $RENV1.STPM$ bit.

The input interrupt of CSTOP signal can be read by the error interrupt cause ($REST.ESSP$).

The input status of CSTOP signal can be read by the extended status ($RSTS.SSTOP$).

You can output one-shot pulse CSTOP signal by writing CMSTOP (07h) command.

The operation can be stopped externally by pulling up CSTOP pin.

<Procedures for simultaneous stop>

Pull up CSTOP pins of PCL6046 that stop simultaneously together.

Set $RMD.MSPE = 1$ to start the axes that will stop simultaneously.

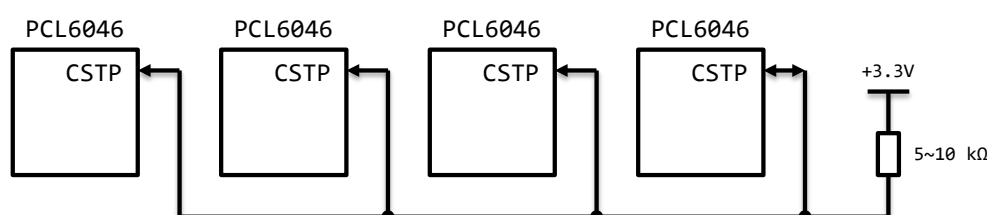
After that, you can stop simultaneously by the following three methods:

1. Write CMSTOP (07h) command

One-shot pulse with the pulse width of 8 cycles ($0.4 \mu s$) of the CLK signal is output from CSTOP pin.

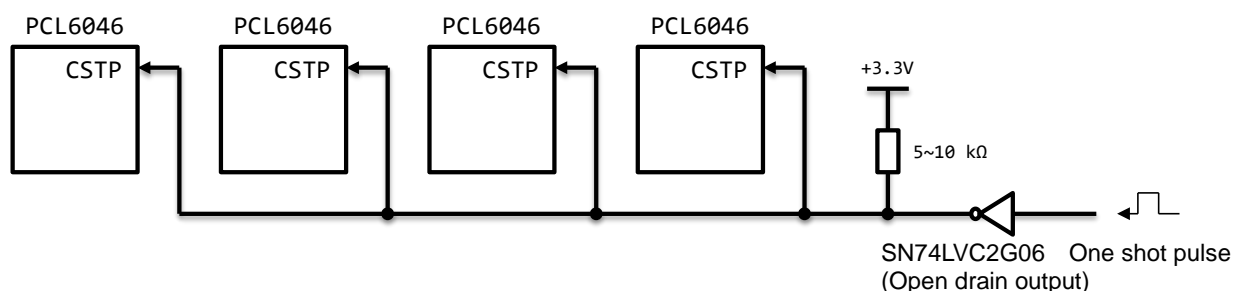
One-shot pulse will be input to all PCL6046s connected to the CSTOP pins to stop the effective axis.

The one-shot pulse will be re-input to the PCL6046 that has output the one-shot pulse and will stop the effective axis.



2. Input a one-shot pulse with the pulse width of 4 cycles ($0.2 \mu s$) or more of the CLK signal from the outside to CSTOP pin.

All PCL6046s connected by the CSTOP pins input a one-shot pulse to stop the effective axis.



3. The axis for which RMD.MSPO = 1 is set stops abnormally.

A one-shot pulse with the pulse width of 8 cycles (0.4 μ s) of the CLK signal is output from CSTP pin.

All PCL6046s connected by the CSTP pins input a one-shot pulse to stop the effective axis.

The one-shot pulse will be re-input to the PC6046 that has output the one-shot pulse and will stop the effective axis.

Name and description	Target
<Input function of CSTP pin > 0: General-purpose input pin 1: Deceleration stop or immediate stop by inputting CSTP signal Input status of CSTP signal is acquired by RSTS.SSTP bit.	RMD.MSPE(24)
<CSTP pin output function> 0: General-purpose output You can output a negative logic one-shot pulse with CMSTP (07h) command. 1: Outputs a negative logic one-shot pulse when the own axis stops abnormally.	RMD.MSPO(25)
<CSTP signal input processing> 0: Immediate stop 1: Deceleration stop	RENV1.STPM(19)
<Error interrupt factor (ESSP)> 1: Stops abnormally because CSTP signal turns ON.	REST.ESSP(8)
<CSTP signal input status> 0: OFF 1: ON	RSTS.SSTP(6)
<Simultaneous stop> CSTP signal is output from CSTP pin. Multiple axes in CSTP signal input enabled status can complete the operation mode.	CMSTP(07h)

6.10.2 Axis selection stop (SELn)

PCL6046 can write the same command to multiple axes using axis selection (SELn).

At this time, if there is only one PCL6046, multiple axes can be stopped simultaneously by writing a stop command.

Software example (H8):

```

var Address = 0x00;           // Specify a common (X-axis) command area for the address
var Command = 0x0349;        // Specify X-axis and Y-axis (03h) for axis selection
                               // Specify the STOP (49h) command as the command
OutputPCL(Address, Command); // Write the axis selection and commands to PCL6046

```

For details on writing commands, see "5.1.3 Write a command".

6.11 Emergency stop

CEMG pin can be used to perform an emergency stop on all axes with an external signal.

An emergency stop can be performed by inputting a one-shot signal to CEMG pin.

By connecting multiple CEMG pins of multiple PCL6046s, all axes can be stopped in an emergency.

You cannot start while CEMG = L level.

The input logic of CEMG signal cannot be selected and is fixed to negative logic.

The input noise filter (RENV1.FLTR) of CEMG pin can also be selected.

The input status of CEMG signal can be read by the extended status (RSTS.SEMG).

The input interrupt of CEMG signal can be read by the error interrupt factor (REST.ESEM).

Check REST.ESEM bits of each axis because the input of CEMG signal can perform an emergency stop for all operating axes.

When writing CMEMG (05h) command to an axis, the written axis can be stopped in an emergency.

When you set to wait for input of INP signal (RMD.MINP = 1), an emergency stop by CEMG signal is not affected by this.

When CEMG signal is input, the operation mode is canceled even if a servo motor driver cannot output an INP signal.

See "6.8.1 Positioning complete (INP)" for INP signals.

Name and description	Target
<Input noise filters of +EL, -EL, +SD, -SD, ORG, ALM, INP, CEMG signals> 0: Signal with the pulse width of 0.05 μ s or more will react reliably. 1: Ignore signals with the pulse width of 3 μ s or less completely.	RENV1.FLTR(26)
<CEMG signal input status> 0: OFF 1: ON	RSTS.SEMG(7)
<Error interrupt factor (ESEM)> 1: Operation stopped abnormally because CEMG signal turns ON.	REST.ESEM(9)
<Emergency stop command> Emergency stop and cancel the operation mode.	CMEMG(05h)

C a u t i o n

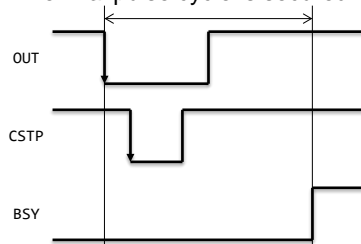
In an emergency stop operation, the final pulse width cannot be secured, and a spike-like pulse may occur.

When a spike-like pulse occurs, the command position and the mechanical position may deviate.

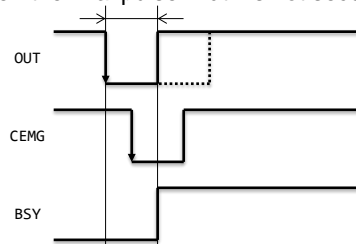
(The motor driver cannot accept the pulse, only the command position counter counts)

Therefore, after an emergency stop, return to the origin and match the command position with the mechanical position.

<Stops other than emergency stops>
The final pulse cycle is secured.



<Emergency stop>
Even the final pulse width is not secured.



6.12 Counter

The counters includes the counter for the number of remaining pulses (RPLS) and counters 1 through 4.

For the counter for the number of remaining pulses, see "5.4.2.7 RPLS: Remaining pulse number" and "5.5.2 Positioning control". Counters 1 to 4 are described in this section.

6.12.1 Counter type and input specifications

You can use the four counters to perform the following function:

- Control of the command position (command pulse) with counter 1
- Control of the mechanical position (encoder) with counter 2
- Stepping motor step-out detection with counter 3 and comparator 3
- IDX signal output with counter 4 and comparator 4

Counter 1 is for command pulse input only.

You can select the input to counters 2-4 with RENV3.CI2, CI3, and CI4-bit.

	Counter 1	Counter 2	Counter 3	Counter 4
Name	Command position	General-purpose 1	Deviation	General-purpose 2
Type	Up/Down	Up/Down	Deviation	Up/Down
Bit length	32	32	16	32
Default count target	Command pulse (Command position)	Encoder (Mechanical position)	Deviation between command pulse and encoder	Command pulse (General-purpose)
Command pulse (OUT, DIR)	Can be input	Can be input	Can be input	Can be input
Encoder (EA, EB)	-	Can be input	Can be input	Can be input
Manual pulser (PA, PB)	-	Can be input	Can be input	Can be input
$\frac{f_{CLK}}{2}$	-	-	-	Can be input

f_{CLK} : Reference clock frequency

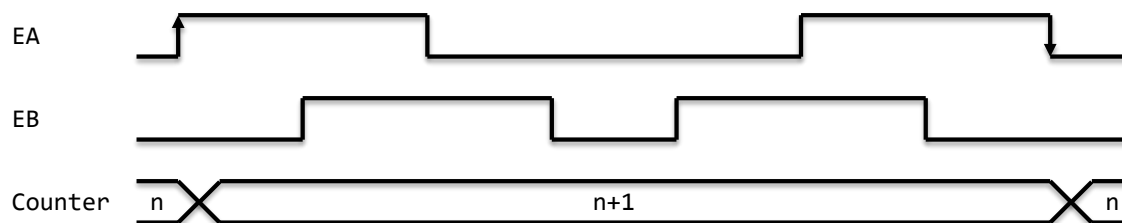
Name and description	Target
<Counting target of counter 2> 00b: EA, EB signal 01b: Command pulse signal 10b: PA, PB signal 11b: Setting prohibited	RENV3.CI2(9,8)
<Counting target of counter 3> 00b: Deviation count between the command pulse signal and EA/ EB signals. 01b: Deviation count between the command pulse signal and PA/ PB signals. 10b: Deviation count between EA/ EB signals and PA/ PB signals. 11b: Setting prohibited.	RENV3.CI3(11,10)
<Counting target of counter 4> 00b: Command pulse signal 01b: EA/ EB signals 10b: PA/ PB signals 11b: $\frac{f_{CLK}}{2}$ signals	RENV3.CI4(13,12)

6.12.1.1 Encoder (EA, EB) signal count

For the encoder (EA, EB) signal, the input noise filter (RENV2.EINF) can be selected.

The input specifications can be selected with RENV2.EIM bit, and the counting direction can be selected with RENV2.EDIR bit.

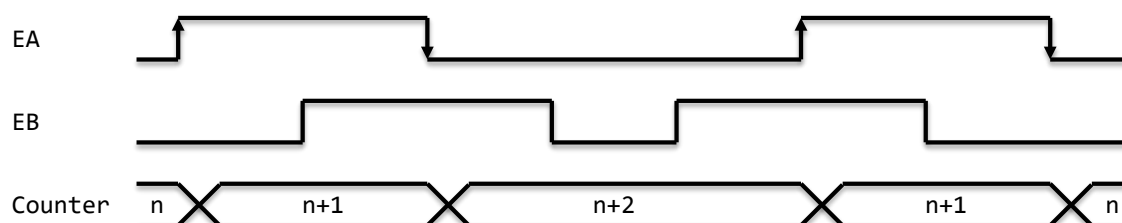
(1) RENV2.EIM=00b: 90-degree phase difference mode (1x)



Count-up: EA signal rises when EB signal is at L level.

Count-down: EA signal falls when EB signal is at L level.

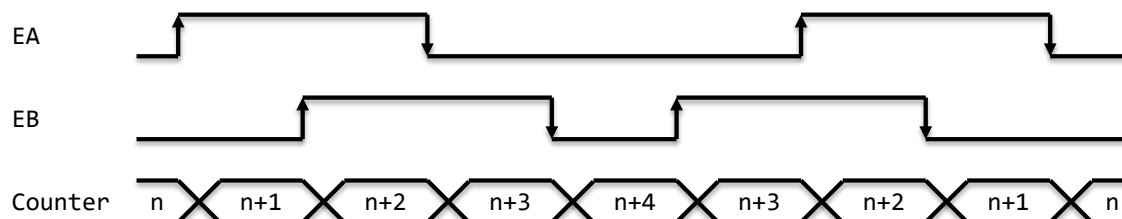
(2) RENV2.EIM=01b: 90-degree phase difference mode (2x)



Count-up: EA signal rises when EB signal is at L level, and EA signal falls when EB signal is at H level.

Count-down: EA signal rises when EB signal is at H level, and EA signal falls when EB signal is at L level.

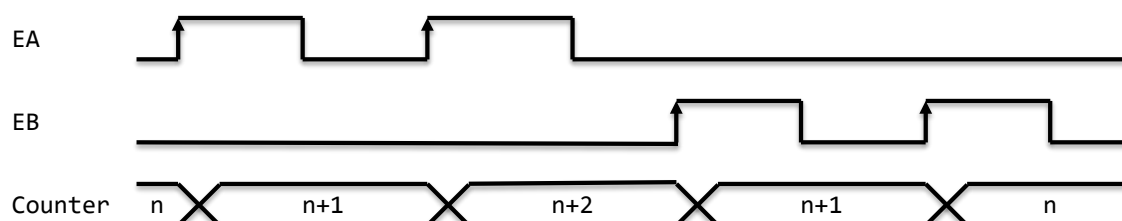
(3) RENV2.EIM=10b: 90-degree phase difference mode (4x)



Count-up: EA signal rises when EB signal is at L level, EA signal falls when EB signal is at H level,
EB signal rises when EA signal is at H level, EB signal falls when EA signal is at L level.

Count-down: EA signal rises when EB signal is at H level, EA signal falls when EB signal is at L level,
EB signal rises when EA signal is at L level, EA signal falls when EA signal is at H level.

(4) RENV2.EIM=11b: 2-pulse mode



Count-up: EA signal rises.

Count-down: EB signal rises.

If RENV2.EDIR = 1 is set, the counting direction will be reversed.

Setting RENV2.EOFF = 1 disables the inputs of EA and EB signals.

EA and EB signal input errors can be read by the error interrupt factor (REST.ESEE).

It occurs when EA and EB signal inputs change simultaneously in 90-degree phase difference mode.

It also occurs when EA and EB signal are input simultaneously in 2-pulse mode.

I m p o r t a n c e

If you turn ON the power to the output source of EA and EB signals (encoder or motor driver) after resetting PCL6046, an EA / EB signal input error may occur.

It usually occurs when noise is detected in EA, EB signals.

Most noise affects EA and EB signals simultaneously.

If it occurs frequently, take noise reduction measures to prevent counting mistakes.

Name and description	Target
<Input noise filters of EA, EB and EZ signals> 0: Signal with the pulse width of 0.05 μs or more will react reliably. 1: Signal with the pulse width of 0.15 μs or more will react reliably.	RENV2.EINF(18)
< Input specifications of EA and EB signals> 00b: 90-degree phase difference mode x1 multiplication 01b: 90-degree phase difference mode x2 multiplication 10b: 90-degree phase difference mode x4 multiplication 11b: 2-pulse mode	RENV2.EIM(21,20)
<Counting direction of EA and EB signals> 0: Count up when the phase of EA signal is advanced. 1: Count up when the phase of EB signal is advanced.	RENV2.EDIR(22)
< Input function of EA and EB signals> 0: Valid. 1: Invalid. Also not detect input errors.	RENV2.EOFF(30)
<Error interrupt factor (ESEE)> 1: EA/ EB signal input error occurred. The operation mode does not stop.	REST.ESEE(16)

6.12.1.2 Manual pulser signal (PA, PB) count

The input noise filter (RENV2.PINF) for manual pulser pins(PAn, PBn) can be selected.

The input specifications can be selected with RENV2.PIM bit, and the count direction can be selected with RENV2.PDIR bit. For RENV2.PIM bit, see "5.5.3 Pulser control".

If RENV2.EDIR = 1 is set, the counting direction is reversed.

Setting RENV2.POFF = 1 disables the inputs of PA and PB signals.

The input error of PA and PB signals can be read by the error interrupt factor (REST.ESPE).

It occurs when the signals change simultaneously in 90-degree phase difference mode and when input simultaneously in 2-pulse mode.

Name and description	Target
<PA and PB signal input noise filter> 0: Signal with the pulse width of 0.05 μs or more will react reliably. 1: Signal with the pulse width of 0.15 μs or more will react reliably.	RENV2.PINF(19)
<PA and PB signal input specifications> 00b: 90-degree phase difference mode x1 multiplication. 01b: 90-degree phase difference mode x2 multiplication. 10b: 90-degree phase difference mode x4 multiplication. 11b: 2-pulse mode. For details, see "5.5.3 Pulser control".	RENV2.PIM(25,24)
<PA and PB signal counting directions> 0: Count up when the phase of PA signal is advanced. 1: Count up when the phase of PB signal is advanced.	RENV2.PDIR(26)
<PA and PB signal input functions> 0: Valid. 1: Invalid. Also not detect input errors.	RENV2.POFF(31)
<Error interrupt factor (ESPO)> 1: Abnormal stops because the buffer counter (16 bit) for inputting PA and PB signals overflowed.	REST.ESPO(14)
<Error interrupt factor (ESPE)> 1: PA or PB signal input error occurred. The operation mode does not stop.	REST.ESPE(17)

C a u t i o n

The count target of a manual pulser (PA, PB) is the signals obtained by multiplying RENV6.PMG bit and dividing the RENV6.PD bit.

6.12.2 Counter clear

Counters 1 to 4 can be cleared by the following 5 methods respectively.

- CLR signal ON (RENV3.CU1C, CU2C, CU3C and CU4C)
- Arriving the origin position in origin return control (RENV3.CU1R, CU2R, CU3R and CU4R)
- Immediately after latching the counter (RENV3.CU1L, CU2L, CU3L and CU4L)
- Writing a counter control command (CUN1R, CUN2R, CUN3R and CUN4R)
- Writing 0 to counter 1 to 4 registers (RCUN1, RCUN2, RCUN3 and RCUN4)

The input logic (RENV1.CLRL) of CLR signal can be selected.

The input specification (RENV1.CLRM) of CLR signal can also be selected.

The input interrupt for CLR signal can be set in the event interrupt request (RIRQ.IRCL).

This can be read by the event interrupt factor (RIST.ISCL).

The input status of CLR signal can be read by the extended status (RSTS.SCLR).

Name and description	Target
<CLR signal input logic> 0: Negative logic 1: Positive logic	RENV1.CLRL(20)
<CLR signal input specifications> 0: Edge trigger (OFF to ON) 1: Level trigger	RENV1.CLRM(21)
<Clear counter 1 by CLR signal turning ON> 0: Not clear. 1: Clear.	RENV3.CU1C(16)
<Clear counter 2 by CLR signal turning ON > 0: Not clear. 1: Clear.	RENV3.CU2C(17)
<Clear counter 3 by CLR signal turning ON > 0: Not clear. 1: Clear.	RENV3.CU3C(18)
<Clear counter 4 by CLR signal turning ON > 0: Not clear. 1: Clear.	RENV3.CU4C(19)
<Clear counter 1 when the origin is reached in origin return control > 0: Not clear. 1: Clear.	RENV3.CU1R(20)
<Clear counter 2 when the origin is reached in origin return control > 0: Not clear. 1: Clear.	RENV3.CU2R(21)

Name and description	Target
<Clear counter 3 when the origin is reached in origin return control > 0: Not clear. 1: Clear.	RENV3.CU3R(22)
<Clear counter 4 when the origin is reached in origin return control > 0: Not clear. 1: Clear.	RENV3.CU4R(23)
<Clear counter 1 to 0 immediately after latching counter 1> 0: Not clear. 1: Clear.	RENV5.CU1L(24)
<Clear counter 2 to 0 immediately after latching counter 2> 0: Not clear. 1: Clear.	RENV5.CU2L(25)
<Clear counter 3 to 0 immediately after latching counter 3> 0: Not clear. 1: Clear.	RENV5.CU3L(26)
<Clear counter 4 to 0 immediately after latching counter 4> 0: Not clear. 1: Clear.	RENV5.CU4L(27)
<Event interrupt request (IRCL)> 1: An interrupt occurs when CLR signal turns ON and the count value is cleared.	RIRQ.IRCL(13)
<Event interrupt factor (ISCL)> 1: CLR signal turns ON and the count value was cleared.	RIST.ISCL(13)
<CLR signal input status> 0: OFF 1: ON	RSTS.SCLR(13)
<Counter 1 control> Clear the count value of counter 1 (RCUN1) to 0.	CUN1R(20h)
<Counter 2 control> Clear the count value of counter 2 (RCUN2) to 0.	CUN2R(21h)
<Counter 3 control> Clear the count value of counter 3 (RCUN3) to 0.	CUN3R(22h)
<Counter 4 control> Clear the count value of counter 4 (RCUN4) to 0.	CUN4R(23h)

C a u t i o n

When clearing the counter immediately after latching, it will be +1 or -1 instead of 0 if you input the count signals during the clearing process.

6.12.3 Counter latch

All counter values can be latched at once at one of the following five timings:

- When LTC signal is changed from OFF to ON.
- When ORG signal is change from OFF to ON.
- When Comparator 4 condition is satisfied.
- When Comparator 5 condition is satisfied.
- When the command is written

The latched value can be read in RLTC1 to RLTC4 registers.

The input specifications of LTC signal can be selected with RENV1.LTCL bit.

The minimum pulse width of LTC signal requires two CLK signal cycles (0.1 μ s).

The timing for latching a counter can be selected with RENV5.LTM bit.

If setting RENV5.LTOF = 1, you can ignore the selection of RENV5.LTM bit and select only LTCH (29h) command.

If setting RENV5.LTFD = 1, RLTC3 register latches the current speed step numbers instead of counter 3.

The input interrupt for LTC signal can be set with the event interrupt request (RIRQ.IRLT).

This can be read by the event interrupt factor (RIST.ISLT).

The input status of LTC signal can be read in extended status (RSTS.SLTC).

The input interrupt for ORG signal can be set with the event interrupt request (RIRQ.IROL).

This can be read by the event interrupt factor (RIST.ISOL).

The input status of ORG signal can be read in extended status (SSTS.SORG).

Name and description	Target
<LTC signal input specifications> 0: Turns on at the falling edge. 1: Turns on at the rising edge.	RENV1.LTCL(23)
<Timing to latch counters 1 to 4> 00b: When LTC signal is changed from OFF to ON. 01b: When ORG signal is changed from OFF to ON. 10b: When the condition of comparator 4 is satisfied. 11b: When the condition of comparator 5 is satisfied.	RENV5.LTM(13,12)
<Latch the current speed instead of counter 3> 0: Latches RCUN3 register (counter 3). 1: Latches RSPD.AS bit (current speed).	RENV5.LTFD(14)
<Latch only at the write timing of LTCH (29h) command> 0: Latches even at the timing selected by RENV5.LTM bit. 1: Latches only at the write timing of LTCH (29h) command.	RENV5.LTOF(15)

Name and description	Target
<Event interrupt request (IRLT)> 1: When RENV5.LTM = 00b is set, an interrupt occurs when LTC signal turns ON.	RIRQ.IRLT(14)
<Event interrupt request (IROL)> 1: When RENV5.LTM = 01b is set, an interrupt occurs when ORG signal turns ON.	RIRQ.IROL(15)
<Event interrupt factor (ISLT)> 1: When RENV5.LTM = 00b is set, LTC signal turns ON.	RIST.ISLT(14)
<Event interrupt factor (ISOL)> 1: When RENV5.LTM = 01b is set, ORG signal turns ON.	RIST.ISOL(15)
<LTC signal input status> 0: OFF 1: ON	RSTS.SLTC(14)
<ORG signal input status> 0: OFF 1: ON	SSTS.SORG(14)
<Counter latch control> Latch the RCUN1 to 4 register values to RLTC1 to 4 registers.	LTCH(29h)

6.12.4 Counter count stop and input stop

Counter 1 is stopped in three ways as follows:

- If RMD.MOD = 47h is set, it will not count.
- If RMD.MCCE = 1 is set, it will not count.
- If RENV3.CU1B = 0 is set, it does not count during backlash correction and slip correction operations.

Counter 2 is stopped in two ways as follows:

- If RENV3.CU2B = 0 is set, it will not count during backlash correction and slip correction operations.
- If RENV3.CU2H = 1 is set, it will not count.

Counter 3 is stopped in two ways as follows:

- If RENV3.CU3B = 0 is set, it will not count during backlash correction and slip correction operations.
- If RENV3.CU3H = 1 is set, it will not count.

Counter 4 is stopped in three ways as follows:

- If RENV3.BSYC = 1 is set, it counts only during BSYn = L level.

If using in combination with RENV3.CI4 = 11b $\frac{f_{CLK}}{2}$, the operating time can be controlled by $RCUN4 \times 2 \times T_{CLK}$

f_{CLK} : Reference clock frequency T_{CLK} : Reference clock cycle

- If RENV3.CU4B = 0 is set, it will not count during backlash correction and slip correction operations.
- If RENV3.CU4H = 1 is set, it will not count.

Name and description	Target
<Operation mode> 100 0111 (47h): Timer operation mode by positioning control.	RMD.MOD(6:0)
<Count function of counter 1> 0: Counts. 1: Does not count. Pulses can be output while counter 1 counting is stopped.	RMD.MCCE(11)
<Count limit of counter 4> 0: No limit. 1: Counts only when BSYn = L level.	RENV3.BSYC(14)
<Counter 1 counts during backlash correction and slip correction> 0: Does not count. 1: Counts.	RENV3.CU1B(24)
<Counter 2 counting during backlash correction and slip correction> 0: Does not count. 1: Counts.	RENV3.CU2B(25)
<Counter 3 counting during backlash correction and slip correction> 0: Does not count. 1: Counts.	RENV3.CU3B(26)
<Counter 4 counting during backlash correction and slip correction> 0: Does not count. 1: Counts.	RENV3.CU4B(27)

Name and description	Target
<Counter 2 counting> 0: Counts. 1: Does not count.	RENV3.CU2H(29)
<Counter 3 counting> 0: Counts. 1: Does not count.	RENV3.CU3H(30)
<Counter 4 count> 0: Counts. 1: Does not count.	RENV3.CU4H(31)

6.13 Comparator

PCL6046 has a built-in 32-bit comparator with 5 circuits / axis.

6.13.1 Comparator types and functions

Using RENV4 register and RENV5 register, you can select the comparison target, the comparison condition, and the processing method with a comparator when the comparison conditions are satisfied.

6.13.1.1 Comparator comparison target

Comparison target	RENV4.C1C (Comparator 1)	RENV4.C2C (Comparator 2)	RENV4.C3C (Comparator 3)	RENV4.C4C (Comparator 4)	RENV5.C5C (Comparator 5)
Counter 1 (RCUN1)	00	00	00	00	000
Counter 2 (RCUN2)	01	01	01	01	001
Counter 3 (RCUN3)	10	10	10	10	010
Counter 4 (RCUN4)	11	11	11	11	011
Number of remaining pulses (RPLS)	-	-	-	-	100
Current speed (RSPD.AS)	-	-	-	-	101
Pre-register	No	No	No	No	Yes
Purpose	+SL signal output, Counter 1 ring count	-SL signal output, Counter 2 ring count	Genera-purpose	IDX signal output	Genera-purpose

For +SL and -SL signal outputs(software limit), see "6.13.2 Software limit".

For IDX signal output, see "6.13.4 Index output".

For Counter 1 Ring counts and Counter 2 Ring counts, see "6.13.5 Ring count".

Name and description	Target
<Comparison target of Comparator 1> 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 If RENV4.C1C = 10b, compare with the absolute value in RCUN3 register (0 to 32,767).	RENV4.C1C(1,0)
<Comparison target of Comparator 2> 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 If RENV4.C2C = 10b, compare with the absolute value in RCUN3 register (0 to 32,767).	RENV4.C2C(9,8)
<Comparison target of Comparator 3> 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 If RENV4.C3C = 10b, compare with the absolute value in RCUN3 register (0 to 32,767).	RENV4.C3C(17,16)
<Comparison target of Comparator 4> 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 If RENV4.C4C = 10b, compare with the absolute value in RCUN3 register (0 to 32,767).	RENV4.C4C(25,24)
<Comparison target of Comparator 5> 000b: RCUN1 001b: RCUN2 010b: RCUN3 011b: RCUN4 100b: RPLS (number of remaining pulses) 101b: RSPD.AS (current speed) If RENV5.C5C = 010b, compare with the absolute value in RCUN3 register (0 to 32,767).	RENV5.C5C(2:0)

6.13.1.2 Comparison conditions for comparison target

Comparison conditions	RENV4.C1S (Comparator 1)	RENV4.C2S (Comparator 2)	RENV4.C3S (Comparator 3)	RENV4.C4S (Comparator 4)	RENV5.C5S (Comparator 5)
Always un-satisfied	000	000	000	0000	000
Comparison target = * 1	001	001	001	0001	001
Comparison target = * 2	010	010	010	0010	010
Comparison target = * 3	011	011	011	0011	011
Comparison target <	100	100	100	0100	100
Comparison target >	101	101	101	0101	101
Software limit	110	110	-	-	-
IDX signal output * 1	-	-	-	1000	-
IDX signal output * 2	-	-	-	1001	-
IDX signal output * 3	-	-	-	1010	-
Counter 1 Ring count	001 (RENV4.C1RM=1)	-	-	-	-
Counter 2 Ring count	-	001 (RENV4.C2RM=1)	-	-	-

* 1 The condition is satisfied regardless of the counting direction.

* 2 The condition is satisfied only when counting up.

* 3 The condition is satisfied only when counting down.

For software limits, see "6.13.2 Software limit".

For IDX signal output, see "6.13.4 Index output".

For Counter 1 Ring counts and Counter 2 Ring counts, see "6.13.5 Ring count".

Name and description	Target
<p><Comparator 1 comparison conditions></p> <p>001b: RCMP1 = Comparison target. (Regardless of counting direction)</p> <p>010b: RCMP1 = Comparison target. (Only when counting up)</p> <p>011b: RCMP1 = Comparison target. (Only when counting down)</p> <p>100b: RCMP1 > Comparison target.</p> <p>101b: RCMP1 < Comparison target.</p> <p>110b: +side software limit (RCMP1 <RCUN1). Also set RENV4.C1C = 00b.</p> <p>Others: Comparison conditions are always un-satisfied.</p>	RENV4.C1S(4:2)
<p><Comparator 2 comparison conditions></p> <p>001b: RCMP2 = Comparison target. (Regardless of counting direction)</p> <p>010b: RCMP2 = Comparison target. (Only when counting up)</p> <p>011b: RCMP2 = Comparison target. (Only when counting down)</p> <p>100b: RCMP2 > Comparison target.</p> <p>101b: RCMP2 < Comparison target.</p> <p>110b: -Side software limit (RCMP2 > RCUN1). Also set RENV4.C2C = 00b.</p> <p>Others: Comparison conditions are always un-satisfied.</p>	RENV4.C2S(12:10)
<p><Comparator 3 comparison conditions></p> <p>001b: RCMP3 = Comparison target. (Regardless of counting direction)</p> <p>010b: RCMP3 = Comparison target. (Only when counting up)</p> <p>011b: RCMP3 = Comparison target. (Only when counting down)</p> <p>100b: RCMP3 > Comparison target.</p> <p>101b: RCMP3 < Comparison target.</p> <p>110b: Setting is prohibited.</p> <p>Others: Comparison conditions are always un-satisfied.</p>	RENV4.C3S(20:18)

Name and description	Target
<p><Comparator 4 comparison conditions></p> <p>0001b: RCMP4 = Comparison target. (Regardless of counting direction)</p> <p>0010b: RCMP4 = Comparison target. (Only when counting up)</p> <p>0011b: RCMP4 = Comparison target. (Only when counting down)</p> <p>0100b: RCMP4 > Comparison target.</p> <p>0101b: RCMP4 < Comparison target.</p> <p>1000b: IDX signal is output under the comparison condition of RENV4.IDXM bits. (Regardless of counting direction)</p> <p>1001b: IDX signal is output under the comparison condition of RENV4.IDXM bits. (Only when counting up)</p> <p>1010b: IDX signal is output the under the comparison condition of RENV4.IDXM bits. (Only when counting down)</p> <p>Others: Comparison conditions are always un-satisfied.</p> <p>When using RENV4.C4S = 1000b, 1001b and 1010b, also set RENV4.C4C = 11b.</p> <p>In this case, when using RENV4.IDXM = 1, set a positive value to RCMP4.</p>	RENV4.C4S(29: 26)
<p><Comparator 5 comparison conditions></p> <p>001b: RCMP5 = Comparison target. (Regardless of counting direction)</p> <p>010b: RCMP5 = Comparison target. (Only when counting up)</p> <p>011b: RCMP5 = Comparison target. (Only when counting down)</p> <p>100b: RCMP5 > Comparison target.</p> <p>101b: RCMP5 < Comparison target.</p> <p>Others: Comparison conditions are always un-satisfied.</p>	RENV5.C5S(5: 3)

6.13.1.3 Processing method when comparison conditions are satisfied

Processing method	RENV4.C1D (Comparator 1)	RENV4.C2D (Comparator 2)	RENV4.C3D (Comparator 3)	RENV4.C4D (Comparator 4)	RENV5.C5D (Comparator 5)
No action	00	00	00	00	00
Immediate stop	01	01	01	01	01
Deceleration stop	10	10	10	10	10
Bulk override	11	11	11	11	11

"No action" can be used for INT signal output, CP1 to CP5 signal outputs, and internal synchronization signal output timing.

For "Bulk override", see "6.13.6 Bulk override".

Name and description	Target
<p><Processing when the condition of Comparator 1 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP1 signals, and to start internal synchronization.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p> <p>If RENV4.C1S = 110b is set, the operation will stop immediately even if RENV4.C1D = 00b, 11b is set.</p>	RENV4.C1D(6,5)
<p><Processing when the condition of Comparator 2 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP2 signals, and to start internal synchronization.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p> <p>If RENV4.C2S = 110b is set, the operation will stop immediately even if RENV4.C2D = 00b, 11b is set.</p>	RENV4.C2D(14,13)
<p><Processing when the condition of Comparator 3 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP3 signals, and to start internal synchronization.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p>	RENV4.C3D(22,21)
<p><Processing when the condition of Comparator 4 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP4 signals, and to start internal synchronization.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p>	RENV4.C4D(31,30)

Name and description	Target
<p><Processing when the condition of Comparator 5 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP5 signals, and to start internal synchronization.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p>	RENV5.C5D(7,6)
<p><Event interrupt request (IRC1)></p> <p>1: An interrupt is generated when the comparison condition of comparator 1 is satisfied. (MSTS.SCP1 changed from 0 to 1)</p>	RIRQ.IRC1(8)
<p><Event interrupt request (IRC2)></p> <p>1: An interrupt is generated when the comparison condition of comparator 2 is satisfied. (MSTS.SCP2 changed from 0 to 1)</p>	RIRQ.IRC2(9)
<p><Event interrupt request (IRC3)></p> <p>1: An interrupt is generated when the comparison condition of comparator 3 is satisfied. (MSTS.SCP3 changed from 0 to 1)</p>	RIRQ.IRC3(10)
<p><Event interrupt request (IRC4)></p> <p>1: An interrupt is generated when the comparison condition of comparator 4 is satisfied. (MSTS.SCP4 changed from 0 to 1)</p>	RIRQ.IRC4(11)
<p><Event interrupt request (IRC5)></p> <p>1: An interrupt is generated when the comparison condition of comparator 5 is satisfied. (MSTS.SCP5 changed from 0 to 1)</p>	RIRQ.IRC5(12)
<p><Event interrupt factor (ISC1)></p> <p>1: The comparison condition of comparator 1 was satisfied. (MSTS.SCP1 changed from 0 to 1)</p>	RIST.ISC1(8)
<p><Event interrupt factor (ISC2)></p> <p>1: The comparison condition of comparator 2 was satisfied. (MSTS.SCP2 changed from 0 to 1)</p>	RIST.ISC2(9)
<p><Event interrupt factor (ISC3)></p> <p>1: The comparison condition of Comparator 3 was satisfied. (MSTS.SCP3 changed from 0 to 1)</p>	RIST.ISC3(10)
<p><Event interrupt factor (ISC4)></p> <p>1: The comparison condition of Comparator 4 was satisfied. (MSTS.SCP4 changed from 0 to 1)</p>	RIST.ISC4(11)
<p><Event interrupt factor (ISC5)></p> <p>1: The comparison condition of Comparator 5 was satisfied. (MSTS.SCP5 changed from 0 to 1)</p>	RIST.ISC5(12)
<p><Main status (SCP1)></p> <p>0: The comparison condition of comparator 1 is not satisfied.</p> <p>1: The comparison condition of comparator 1 is satisfied.</p>	MSTS.SCP1(8)

Name and description	Target
<Main status (SCP2)> 0: The comparison condition of comparator 2 is not satisfied. 1: The comparison condition of comparator 2 is satisfied.	MSTS.SCP2(9)
<Main status (SCP3)> 0: The comparison condition of comparator 3 is not satisfied. 1: The comparison condition of comparator 3 is satisfied.	MSTS.SCP3(10)
<Main status (SCP4)> 0: The comparison condition of comparator 4 is not satisfied. 1: The comparison condition of comparator 4 is satisfied.	MSTS.SCP4(11)
<Main status (SCP5)> 0: The comparison condition of comparator 5 is not satisfied. 1: The comparison condition of comparator 5 is satisfied.	MSTS.SCP5(12)
<P3n pin input/output function> 10b: CP1 signal is output at negative logic when the comparator 1 condition is satisfied. 11b: CP1 signal is output at positive logic when the comparator 1 condition is satisfied.	RENV2.P3M(7,6)
<P4n pin input/output function> 10b: CP2 signal is output at negative logic when the comparator 2 condition is satisfied. 11b: CP2 signal is output at positive logic when the comparator 2 condition is satisfied.	RENV2.P4M(9,8)
<P5n pin input/output function> 10b: CP3 signal is output at negative logic when the comparator 3 condition is satisfied. 11b: CP3 signal is output at positive logic when the comparator 3 condition is satisfied.	RENV2.P5M(11,10)
<P6n pin input/output function> 10b: CP4 signal is output at negative logic when the comparator 4 condition is satisfied. 11b: CP4 signal is output at positive logic when the comparator 4 condition is satisfied.	RENV2.P6M(13,12)
<P7n pin input/output function> 10b: CP5 signal is output at negative logic when the comparator 5 condition is satisfied. 11b: CP5 signal is output at positive logic when the comparator 5 condition is satisfied.	RENV2.P7M(15,14)
<Output timing of internal synchronization signal> 0001b: When comparator 1 condition is satisfied 0011b: When comparator 3 condition is satisfied 0101b: When comparator 5 condition is satisfied 0010b: When comparator 2 condition is satisfied 0100b: When comparator 4 condition is satisfied	RENV5.SYO(19: 16)

6.13.2 Software limit

You can use software limits with Comparator 1 and Comparator 2.

If the condition of +SL (Comparator 1) is satisfied while operating in the +direction, the operation will stop abnormally.

If the condition of -SL (Comparator 2) is satisfied while operating in the -direction, the operation will stop abnormally.

Set a counter other than counter 3 for the comparison target of comparator 1.

Set the same counter as the comparison target of Comparator 1 for the comparison target of Comparator 2.

Set the command pulse count to this counter.

Select the stop method (RENV4.C1D, C2D) from immediate stop or deceleration stop.

If you select deceleration stop, the operation will stop after passing through +SL position or -SL position.

When the condition of +SL is satisfied, the operation does not start in the +direction. When that of -SL is satisfied, it does not start in the -direction.

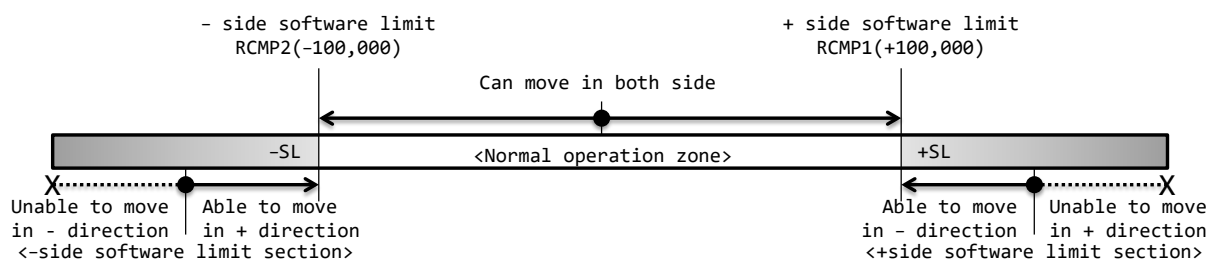
Setting Example:

RENV4 = 00003838h: Comparator 1 is set to stop immediately with +SL.

Comparator 2 is set to stop immediately with -SL.

RCMP1 = +100,000: Set the +side software limit value for the comparator 1 comparison value.

RCMP2 = -100,000: Set the - side software limit value for the comparator 2 comparison value.



Abnormal stop due to +SL and -SL can be read by the error interrupt cause (REST.ESC1, ESC2).

Condition satisfied status of +SL and -SL can be read by the main status (MSTS.SCP1, SCP2).

Name and description	Target
<Comparator 1 comparison target> 00b: RCUN1 01b: RCUN2 11b: RCUN4	RENV4.C1C(1,0)
< Comparator 1 comparison conditions> 110b: +side software limit (RCMP1 < comparison target)	RENV4.C1S(4: 2)
<Processing when the condition of Comparator 1 is satisfied> 01b: Immediate stop. 10b: Deceleration stop. If RENV4.C1S = 110b is set, the operation will stop immediately even if RENV4.C1D = 00b, 11b is set.	RENV4.C1D(6,5)
<Comparator 1 comparison target> 00b: RCUN1 01b: RCUN2 11b: RCUN4	RENV4.C2C(9,8)

Name and description	Target
<Comparator 2 comparison conditions> 110b: –Side software limit (RCMP2 > comparison target)	RENV4.C2S(12:10)
<Processing when the condition of Comparator 2 is satisfied> 01b: Immediate stop. 10b: Deceleration stop. If RENV4.C2S = 110b is set, the operation will stop immediately even if RENV4.C2D = 00b, 11b is set.	RENV4.C2D(14,13)
<Error interrupt factor (CP1 / +SL)> 1: An abnormal stop occurred because the comparison condition of Comparator 1 was satisfied. (Including stop by +SL)	REST.ESC1(0)
<Error interrupt factor (CP2 / –SL)> 1: An abnormal stop occurred because the comparison condition of Comparator 2 was satisfied. (Including stop by –SL)	REST.ESC2(1)
<Satisfied status of CP1> 0: The comparison condition of comparator 1 is not satisfied. 1: The comparison condition of comparator 1 is satisfied.	MSTS.SCP1(8)
<Satisfied status of CP2> 0: The comparison condition of comparator 2 is not satisfied. 1: The comparison condition of comparator 2 is satisfied.	MSTS.SCP2(9)

6.13.3 Out-of-step detection of stepping motor

By setting the counter 3 count value (RCUN3) as the comparison target, out-of-step with a stepping motor can be detected.

Counter 3 counts the deviation between command pulse signals and EA / EB signals.

Set the maximum deviation tolerance (absolute value) in the comparison value in a comparator that is compared with counter 3 to detect out-of-step. When out-of-step is detected, you can select the processing methods from (RENV4.C1D, C2D, C3D, C4D and RENV5.C5D) when the comparison condition is satisfied.

Match the resolution of the encoder to the resolution of a stepping motor.

For a 200 spr stepper motor, use a 200 ppr encoder. (spr: steps per revolution. ppr: pulses per revolution.)

You can select the input specifications (RENV2.EIM) of the feedback to input to EAn and EBn pins.

If an EA or EB signal input error occurs, it can be read by the error interrupt factor (REST.ESEE).

An input error occurs when EA and EB signals change at the same time in 90-degree phase difference mode.

It also occurs when EA and EB signals are input at the same time in 2-pulse mode.

You can clear Counter 3 to 0 by writing CUNR3 (22h) command.

Setting Example:

RENV4 = 00360000h: Set the comparison target of Comparator 3 to counter 3.

(Selecting counter 3 to compare with the absolute value of RCUN3 register)

Comparator 3 Comparison value < Comparison target is set to stop immediately.

RCMP3 = 32: Set the maximum deviation tolerance to 32.

If the absolute value of the deviation between command pulse signals and EA / EB signals exceeds 32, it is considered to be out-of-step. At this time, the operation stops immediately, and an error interrupt is generated.

Name and description	Target
<EA/ EB signal input specifications> 00b: 90-degree phase difference mode x1 multiplication. 01b: 90-degree phase difference mode x2 multiplication. 10b: 90-degree phase difference mode x4 multiplication. 11b: 2-pulse mode. For details, see "6.12.1 Counter type and input specifications".	RENV2.EIM(21,20)
<Counting direction of EA and EB signals> 0: Count up when the phase of EA signal is advanced. 1: Count up when the phase of EB signal is advanced.	RENV2.EDIR(22)

Name and description	Target
<p><Comparator 1 comparison target></p> <p>10b: RCUN3</p> <p>If RENV4.C1C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767).</p>	RENV4.C1C(1,0)
<p><Comparator 2 comparison target></p> <p>10b: RCUN3</p> <p>If RENV4.C2C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767).</p>	RENV4.C2C(9,8)
<p><Comparator 3 comparison target></p> <p>10b: RCUN3</p> <p>If RENV4.C3C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767).</p>	RENV4.C3C(17,16)
<p><Comparator 4 comparison target></p> <p>10b: RCUN3</p> <p>If RENV4.C4C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767).</p>	RENV4.C4C(25,24)
<p><Comparator 5 comparison target></p> <p>010b: RCUN3</p> <p>If RENV5.C5C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767).</p>	RENV5.C5C(2:0)
<p><Comparator 1 comparison conditions></p> <p>101b: RCMP1 < Comparison target</p>	RENV4.C1S(4:2)
<p><Comparator 2 comparison conditions></p> <p>101b: RCMP2 < Comparison target</p>	RENV4.C2S(12:10)
<p><Comparator 3 comparison conditions></p> <p>101b: RCMP3 < Comparison target</p>	RENV4.C3S(20:18)
<p><Comparator 4 comparison conditions></p> <p>0101b: RCMP4 < Comparison target</p>	RENV4.C4S(29:16)
<p><Comparator 5 comparison conditions></p> <p>101b: RCMP5 < Comparison target</p>	RENV5.C5S(5:3)
<p><Processing when the condition of Comparator 1 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP1 signals, and to start internal synchronization.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p>	RENV4.C1D(6,5)
<p><Processing when the condition of Comparator 2 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP2 signals, and internal synchronization start.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p>	RENV4.C2D(14,13)

Name and description	Target
<p><Processing when the condition of Comparator 3 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP3 signals, and internal synchronization start.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p>	RENV4.C3D(22,21)
<p><Processing when the condition of Comparator 4 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP4 signals, and internal synchronization start.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p>	RENV4.C4D(31,30)
<p><Processing when the condition of Comparator 5 is satisfied></p> <p>00b: No processing. It can be used to output INT signals and CP5 signals, and internal synchronization start.</p> <p>01b: Immediate stop.</p> <p>10b: Deceleration stop.</p> <p>11b: Bulk override.</p>	RENV5.C5D(7,6)
<p><Counter 3></p> <p>Register to get counter 3 (deviation).</p> <p>Signed deviation values are obtained.</p>	RCUN3(15: 0)
<p><Comparator 1 comparison value></p> <p>Register in Comparator 1 to set the comparison value.</p>	RCMP1(31: 0)
<p><Comparator 2 comparison value ></p> <p>Register in Comparator 2 to set the comparison value.</p>	RCMP2(31: 0)
<p><Comparator 3 comparison value></p> <p>Register in Comparator 3 to set the comparison value.</p>	RCMP3(31: 0)
<p><Comparator 4 comparison value></p> <p>Register in Comparator 4 to set the comparison value.</p>	RCMP4(31: 0)
<p><Comparator 5 comparison value></p> <p>Register in Comparator 5 to set the comparison value.</p>	RCMP5(31: 0)
<p><Error interrupt factor (ESC1) ></p> <p>1: An abnormal stop occurred because the comparison condition of Comparator 1 was satisfied.</p>	REST.ESC1(0)
<p><Error interrupt factor (ESC2)></p> <p>1: An abnormal stop occurred because the comparison condition of Comparator 2 was satisfied.</p>	REST.ESC2(1)
<p><Error interrupt factor (ESC3)></p> <p>1: An abnormal stop occurred because the comparison condition of Comparator 3 was satisfied.</p>	REST.ESC3(2)
<p><Error interrupt factor (ESC4)></p> <p>1: An abnormal stop occurred because the comparison condition of Comparator 4 was satisfied.</p>	REST.ESC4(3)
<p><Error interrupt factor (ESC5)></p> <p>1: An abnormal stop occurred because the comparison condition of Comparator 5 was satisfied.</p>	REST.ESC5(4)

Name and description	Target
<Error interrupt factor (ESEE)> 1: EA/ EB signal input error has occurred. The operation mode does not stop.	REST.ESEE(16)
<Main status (SCP1)> 0: The comparison condition of comparator 1 is not satisfied. 1: The comparison condition of comparator 1 is satisfied.	MSTS.SCP1(8)
<Main status (SCP2)> 0: The comparison condition of comparator 2 is not satisfied. 1: The comparison condition of comparator 2 is satisfied.	MSTS.SCP2(9)
<Main status (SCP3)> 0: The comparison condition of comparator 3 is not satisfied. 1: The comparison condition of comparator 3 is satisfied.	MSTS.SCP3(10)
<Main status (SCP4)> 0: The comparison condition of comparator 4 is not satisfied. 1: The comparison condition of comparator 4 is satisfied.	MSTS.SCP4(11)
<Main status (SCP5)> 0: The comparison condition of comparator 5 is not satisfied. 1: The comparison condition of comparator 5 is satisfied.	MSTS.SCP5(12)
<Counter 3 control command> Clears counter 3 (deviation) to 0.	CUN3R(22h)

6.13.4 Index output

Comparator 4 can be used to periodically output the index (IDX) signal from CP4n pin.

Set counter 4 as the comparison target of comparator 4.

Then, set Comparator 4 to the outputs of IDX signal (RENV4.C4S = 1000b, 1001b, 1010b).

The count range of RCUN4 register will be from 0 to the RCMP4 register value.

Counting down from 0 results in the RCMP4 register value.

It becomes 0 when performing counting up from the RCMP4 register value.

The setting of RENV4.IDXM bit is valid only when the outputs of IDX signal (RENV4.C4S = 1000b, 1001b, 1010b) is selected.

When RENV4.IDXM = 0 is set, IDX signal becomes level outputs of the logic set in RENV2.P6M bits.

When RCUN4 = RCMP4 is satisfied, IDX signal is output at the level.

If RENV4.C4S = 1001b, 1010b is set for this level output, set RCMP4 to 2 or more.

If RENV4.IDXM = 1 is set, IDX signal becomes the pulse output of the logic set in RENV2.P6M bits.

When changing to RCUN4 = 0, IDX signal with 2-cycle width of the CLK signal is output as a pulse.

With this pulse output setting, even if RCUN4 register is cleared to 0, no IDX signal is output.

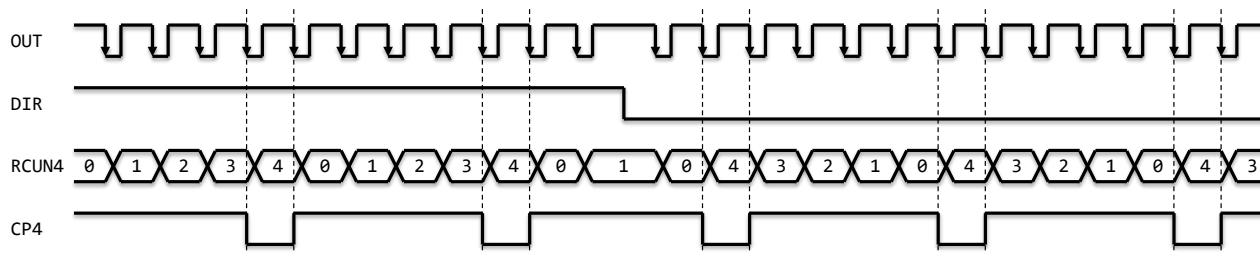
Level outputs setting example:

CP4n pin outputs IDX signal at negative logic.

Counter 4 counts the command pulses from 0 to 4.

When RCUN4 = 4, the L level IDX signal is output.

Setting value: RENV1=00000000h, RENV2=00002000h, RENV3=00000000h, RENV4=23000000h, RCMP4=4.



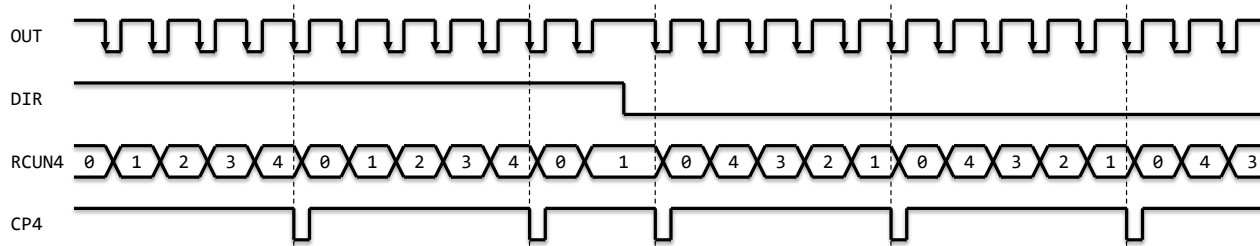
Pulse output setting example:

CP4n pin outputs IDX signal at negative logic.

Counter 4 counts the command pulses from 0 to 4.

When changing to RCUN4 = 0, IDX signal with 2-cycle width of the CLK signal is output.

Setting value: RENV1=00000000h, RENV2=00002000h, RENV3=00000000h, RENV4=23800000h, RCMP4=4.



Name and description	Target
<p><P6n pin input/output function></p> <p>10b: CP4 signal is output at negative logic when the condition of Comparator 4 is satisfied.</p> <p>11b: CP4 signal is output at positive logic when the condition of Comparator 4 is satisfied.</p>	RENV2.P6M(13,12)
<p>< Counter 4 count target ></p> <p>00b: Command pulse signals 01b: EA, EB signals 10b: PA, PB signals 11b: $\frac{f_{CLK}}{2}$ signal</p>	RENV3.CI4(13,12)
<p><IDX signal output conditions></p> <p>0: Level output with the logic set in RENV2.P6M bit.</p> <p> When RCUN4 = RCMP4 is satisfied, IDX signal is output at the level.</p> <p>1: Pulse output with the logic set in RENV2.P6M bit.</p> <p> When changing to RCUN4 = 0, IDX signal of 2-cycle width of the CLK signal is output.</p>	RENV4.IDXM(23)
<p><Comparator 4 comparison target ></p> <p>11b: RCUN4</p>	RENV4.C4C(25,24)
<p><Comparator 4 comparison conditions></p> <p>1000b: IDX signal is output under the comparison condition of RENV4.IDXM bits. (Regardless of counting direction)</p> <p>1001b: IDX signal is output under the comparison condition of RENV4.IDXM bits. (Only during count-up)</p> <p>1010b: IDX signal is output under the comparison condition of RENV4.IDXM bits. (Only during count-down)</p> <p>When using RENV4.C4S = 1000b, 1001b, 1010b, also set RENV4.C4C = 11b.</p> <p>In this case, set a positive value in RCMP4 when using RENV4.IDXM = 1.</p>	RENV4.C4S(29: 26)

6.13.5 Ring count

You can use counter 1 and counter 2 for ring-counting for turntable controls. By using it as a ring counter, you can control the current position of turntables.

Comparator 1 is used for the ring count of counter 1.

Comparator 2 is used for the ring count of counter 2.

The ring count goes from the maximum value to 0 when counting up, and from 0 to the maximum value when counting down.

If you set RENV4.C1RM = 1, RENV4.C1S = 000b, RENV4.C1C = 00b, counter 1 will ring count.

The maximum ring count for counter 1 is set in RCMP1 register.

If you set RENV4.C2RM = 1, RENV4.C2S = 000b, RENV4.C2C = 01b, counter 2 will ring count.

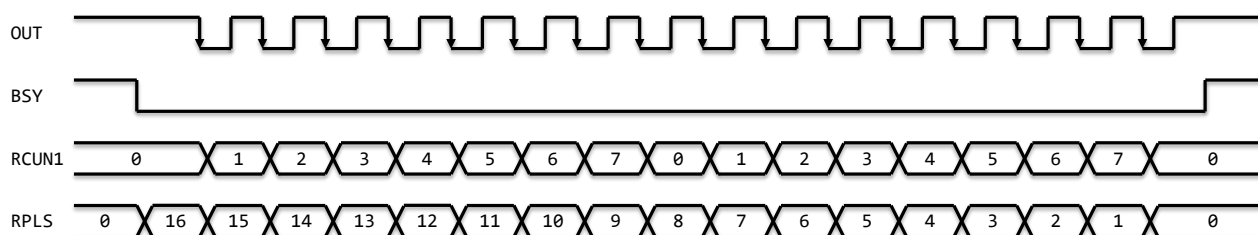
The maximum ring count for counter 2 is set in RCMP2 register.

Ring count setting example:

The incremental movement of positioning control rotates a table of 8 pulses per rotation twice.

The RCUN1 register value after stopping is the same as before starting (0 in the example below).

Setting values: RMV=00000010h, RMD=00000041h, RENV4=00000080h, RCUN1=0, RCMP1=7.



Name and description	Target
<Comparator 1 comparison target> 00b: RCUN1	RENV4.C1C(1,0)
<Comparator 1 comparison conditions> 001b: RCMP1 = Comparison target (regardless of counting direction)	RENV4.C1S(4: 2)
<When the comparator 1 condition is satisfied> 00b: No processing. It can be used to output INT signal and CP1 signal, and internal synchronization start.	RENV4.C1D(6,5)
<Counter 1 ring count> 1: Performs ring count.	RENV4.C1RM(7)
<Comparator 2 comparison target> 01b: RCUN2	RENV4.C2C(9,8)
<Comparator 2 comparison condition> 001b: RCMP2 = Comparison target (regardless of counting direction)	RENV4.C2S(12: 10)
<When the condition of comparator 2 is satisfied> 00b: No processing. It can be used to output INT signals and CP2 signals, and for internal synchronization start	RENV4.C2D(14,13)

Name and description	Target
<Counter 2 ring count> 1: Performs ring count.	RENV4.C2RM(15)

C a u t i o n s

Set the initial value of a ring count counter in the range from 0 to the maximum value (comparator comparison value).
If you start from out the range, it will not work properly.

6.13.6 Bulk override

The data in the pre-register for continuous operation, which is determined by PRSET (4Fh) command is called the overriding data. The overriding data should be written from the 2nd pre-register at least when the current register is determined.

The written undetermined data will be determined as overriding data when writing PRSET (4Fh) command.

It is shared with the pre-register for continuous operation, so you cannot write it when the pre-register is full (MSTS.SPRF = 1).

You cannot identify whether the data in the pre-register for continuous operation is the data for continuous operation or data for overriding.

You can shift (bulk override) the overriding data by writing PRESHF (2Bh) command.

The data can also be overridden at once when the comparison condition of a comparator is satisfied.

If you select encoder signals for the counter to be compared, you can override it all at once at the specified position.

If you select the current speed step number in Comparator 5, you can override all at the specified speed.

If you select the $\frac{f_{CLK}}{2}$ signal for counter 4, you can override it all at once for a specified time.

Bulk override overrides all continuous operation pre-register values into the respective registers.

Even if you have overridden the target position etc. individually right before, it will be overridden to the previous value with bulk override. When using individual overrides together, use PRESHF (2Bh) command to override them all at once.

Name and description	Target
<Main status (SPRF)> 0: 2nd pre-register for continuous operation data is undetermined. 1: 2nd pre-register for continuous operation data is determined.	MSTS.SPRF(14)
<Processing when the condition of Comparator 1 is satisfied> 11b: Bulk override.	RENV4.C1D(6,5)
<Processing when the condition of Comparator 2 is satisfied> 11b: Bulk override.	RENV4.C2D(14,13)
<Processing when the condition of Comparator 3 is satisfied> 11b: Bulk override.	RENV4.C3D(22,21)
<Processing when the condition of Comparator 4 is satisfied> 11b: Bulk override.	RENV4.C4D(31,30)
<Processing when the condition of Comparator 5 is satisfied> 11b: Bulk override.	RENV5.C5D(7,6)
<Pre-register control command (PRESHF)> Shifts all the data in pre-registers for continuous operation.	PRESHF(2Bh)
<Pre-register control command (PRSET)> Sets the pre-register for a continuous operation in determined status as overriding data.	PRSET(4Fh)

6.13.6.1 Bulk override example 1

This is an example of using the overriding data in determined status with PRES HF (2Bh) command, and then starting the next continuous operation. Determine the data 1 for initial operation, the data 1' for overriding, and the data 2 for continuous operation, then start.

No.	Procedures	2 nd pre-register	1 st pre-register	Current register	RSTS. PFM	MSTS. SPRF
1	Default status in stopping.	0 (Undetermined)	0 (Undetermined)	0 (Undetermined)	00	0
2	Set the waiting for input of CSTA signal (PRMD.MSY = 1). Write Data 1 to the 2nd pre-register. Data 1 is copied to the 1st pre-register. Data 1 is also copied to the current register.	Data 1 (Undetermined)	Data 1 (Undetermined)	Data 1 (Undetermined)	00	0
3	Write the start command 1 for the first operation. Determine Data 1 in the current register. Waiting for CSTA signal input (RSTS.CND = 0010b).	Data 1 (Undetermined)	Data 1 (Undetermined)	Data 1 (Determined)	01	0
4	Write Data 1' to the 2nd pre-register. Data 1' is copied to the 1st pre-register.	Data 1' (Undetermined)	Data 1' (Undetermined)	Data 1 (Determined)	01	0
5	Write the PRSET (4Fh) command. Determine the overriding data 1' in the 1st pre-register.	Data 1' (Undetermined)	Data 1' (Determined)	Data 1 (Determined)	10	0
6	Cancel waiting for CSTA signal input (PRMD.MSY = 0). Write Data 2 to the 2nd pre-register. Data 2 is not copied.	Data 2 (Undetermined)	Data 1' (Determined)	Data 1 (Determined)	10	0
7	Write the start command 2 for the continuous operation. Determine the Data 2 for the continuous operation of the 2nd pre-register. Input CSTA signal. Start the operation with Data 1 and Start command 1.	Data 2 (Determined)	Data 1' (Determined)	Data 1 (Determined)	11	1
8	Write PRES HF (2Bh) command. Data 1' is copied to the current register. Data 2 is copied to the 1st pre-register. Operation continues with Data 1'. The speed pattern remains at Start command 1. The 2nd pre-register becomes undetermined.	Data 2 (Undetermined)	Data 2 (Determined)	Data 1' (Determined)	10	0
9	The operation mode of Data 1' is completed. Data 2 is copied to the current register. Start the operation with Data 2 and Start command 2. The 1st pre-register becomes undetermined.	Data 2 (Undetermined)	Data 2 (Undetermined)	Data 2 (Determined)	01	0
10	The operation mode of data 2 is completed. The current register becomes undetermined. Since there is no determined register, the continuous operation is completed.	Data 2 (Undetermined)	Data 2 (Undetermined)	Data 2 (Undetermined)	00	0

6.13.6.2 Bulk override example 2

This is an example in which the next continuous operation starts without using determined overriding data.

Determines the Data 1 for the initial operation, the Data 1' for overriding, and the Data 2 for the continuous operation, and start.

If the operation mode of the current register is completed before the comparator condition is satisfied, the pre-register is also shifted. If there is data for the continuous operation remained in the pre-register, it will operate after the shift.

No.	Procedures	2 nd pre-register	1 st pre-register	Current register	RSTS. PFM	MSTS. SPRF
1	Set PRMD.MSY = 1 and PRMD.MOD = 41h. Set PRMV = 1000. Set the Data 1 for the initial operation in the remaining registers. Write the STAUD (53h) command. Set the Data 1' for overriding. Write PRSET (4Fh) command. Set PRMD.MSY = 0 and PRMD.MOD = 41h. Set the Data 2 for continuous operation in the remaining registers. Write STAUD (53h) command. Set RENV5.C5S = 001b and RENV5.C5D = 11b. (Override all at once with RCMP5 = RCUN1) Set PRCP5 = 1500. RCMP5 register is determined. Set RCUN1 = 0. Write SPSTA (2Ah) command to start.	Data 2 (Determined)	Data 1' (Determined)	Data 1 (Determined)	11	1
2	Stops at RCUN1 = 1000 (RPLS = 0). Data 1' shifts to the register. Data 2 shifts to the 1st pre-register.	Data 2 (Undetermined)	Data 2 (Determined)	Data 1' (Determined)	10	0
3	Since there is no start command, the override ends. Data 2 shifts to the current register. The continuous operation starts	Data 2 (Undetermined)	Data 2 (Undetermined)	Data 2 (Determined)	01	0
4	Stops at RCUN1 = 2000 (RPLS = 0). The current register becomes undetermined. Since there is no determined register, the continuous operation is completed.	Data 2 (Undetermined)	Data 2 (Undetermined)	Data 2 (Undetermined)	00	0

6.13.6.3 Bulk override example 3

This is an example in which the operation mode will be completed without using determined overriding data.

Determines the Data 1 for the initial operation, the Data 1' for the override 1, and the Data 1" for the override 2, and start.

If the operation mode of the current register is completed before the comparator condition is satisfied, the pre-register is also shifted. If there is no continuous operation data left in the pre-register, only shifting is performed.

No.	Procedures	2 nd pre-register	1 st pre-register	Current register	RSTS. PFM	MSTS. SPRF
1	Set PRMD.MSY = 1 and PRMD.MOD = 41h. Set PRMV = 1000. Set the Data 1 for the initial operation in the remaining registers. Write STAUD (53h) command. Determines the Data 1 in the current register. Set the Data 1' for override 1. Write PRSET (4Fh) command. Determines the Data 1' in the 1st register. Set the Data 1" for override 2. Write PRSET (4Fh) command. Determines the Data 1" in the 2nd register. Set RENV5.C5S = 001b and RENV5.C5D = 11b. Set PRCP5 = 1500. RCMP5 register is determined. Set RCUN1 = 0. Write SPSTA (2Ah) command to start.	Data 1" (Determined)	Data 1' (Determined)	Data 1 (Determined)	11	1
2	Operation will stop at RCUN1 = 1000 (RPLS = 0). Data 1' is shifted to the current register. Data 1" is shifted to the 1st pre-register.	Data 1" (Undetermined)	Data 1" (Determined)	Data 1' (Determined)	10	0
3	Since there is no start command, Data 1' is complete. Since there is no Data for continuous operation, Data 1" will not be shifted. Since there is no determined register, continuous operation is completed.	Data 1" (Undetermined)	Data 1" (Undetermined)	Data 1' (Undetermined)	00	0

6.13.6.4 Pre-register for continuous comparison use example

This is an example in which you use the determined overriding data with the comparator, and then start the next continuous operation.

Determine the Data 1 for the initial operation, the Data 1' for the override 1, and the Data 1" for the override 2, and start the operation.

Comparator 5 has the pre-register for continuous comparison (PRCP5).

The use of continuous comparison pre-register allows multiple bulk overrides with multiple comparison conditions.

For the target position of 1000 pulse, the FH speed is overridden at the incremental positions at 251 pulse and 501 pulse.

No.	Procedures	2 nd pre-register	1 st pre-register	Current register	RSTS. PFM	RSTS. PFC
1	Set PRMD.MSY = 1 and PRMD.MOD = 41h. Set PRMV = 1000. Set PRFH = 200. Set the Data 1 for the initial operation in the remaining registers. Write STAUD (53h) command. Set PRFH = 400 as override Data 1'. Write the PRSET (4Fh) command. Set PRFH = 800 as override Data 1". Write PRSET (4Fh) command. Set RENV5.C5S = 001b and RENV5.C5D = 11b. Set PRCP5 = 250 (Comparison A). The RCMP5 register is determined. Set PRCP5 = 500 (Comparison B). The 1st pre-register of the RCMP5 register is determined. Set RCUN1 = 0. Write SPSTA (2Ah) command to start.	PRFH etc. Data 1" (Determined)	Data 1' (Determined)	RFH etc. Data 1 (Determined)	11	10
		PRCP5 Comparison B (Undetermined)	Comparison B (Determined)	RCMP5 Comparison A (Determined)		
2	RCUN1 (251) > RCMP5 (250). Data 1' is shifted to the current register. Data 1" is shifted to the 1st pre-register. The target speed is overridden to RFH = 400.	Data 1" (Undetermined)	Data 1" (Determined)	Data 1' (Determined)	10	01
		Comparison B (Undetermined)	Comparison B (Undetermined)	Comparison B (Determined)		
3	RCUN1 (501) > RCMP5 (500). Data 1" is shifted to the current register. The target speed is overridden to RFH = 800.	Data 1" (Undetermined)	Data 1" (Undetermined)	Data 1" (Determined)	01	00
		Comparison B (Undetermined)	Comparison B (Undetermined)	Comparison B (Undetermined)		
4	Operation will stop at RCUN1 = 1000 (RPLS = 0).	Data 1" (Undetermined)	Data 1" (Undetermined)	Data 1" (Undetermined)	00	00
		Comparison B (Undetermined)	Comparison B (Undetermined)	Comparison B (Undetermined)		

6.14 Backlash correction

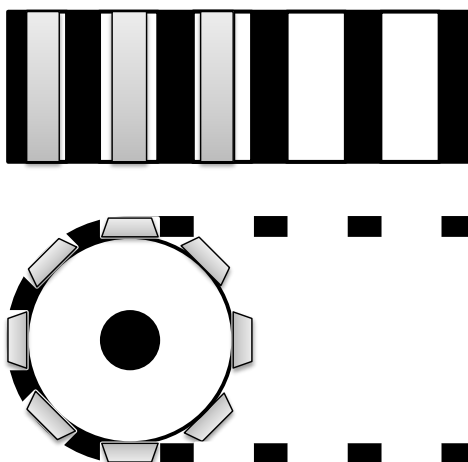
For actuators that use gears or chains, there is a function to compensate backlashes in reversal motions.

If RENV6.ADJ = 01b is set, the number of pulses set in RENV6.BR bits is output at FA speed, and then the operation starts.

Backlash correction is performed at each start when the direction of movement changes.

However, in circular interpolation operation, backlash correction cannot be performed even if the operation direction changes.

With RENV3.CU1B, CU2B, CU3B, CU4B bits, you can set whether to count the correction pulses in addition to the command pulses.



Adjust the RENV6.BR bit and RFA register settings in an experiment with the actual machine.

Name and description	Target
<Counter 1 during backlash correction > 0: Does not count. 1: Count.	RENV3.CU1B(24)
<Counter 2 during backlash correction > 0: Does not count. 1: Count.	RENV3.CU2B(25)
<Counter 3 during backlash correction > 0: Does not count. 1: Count.	RENV3.CU3B(26)
<Counter 4 during backlash correction > 0: Does not count. 1: Count.	RENV3.CU4B(27)
<Backlash correction amount> The setting range is 0 to 4,095.	RENV6.BR(11:0)
<Function to correct the feed amount> 00b: Does not correct feed amount. 01b: Performs backlash correction.	RENV6.ADJ(13,12)

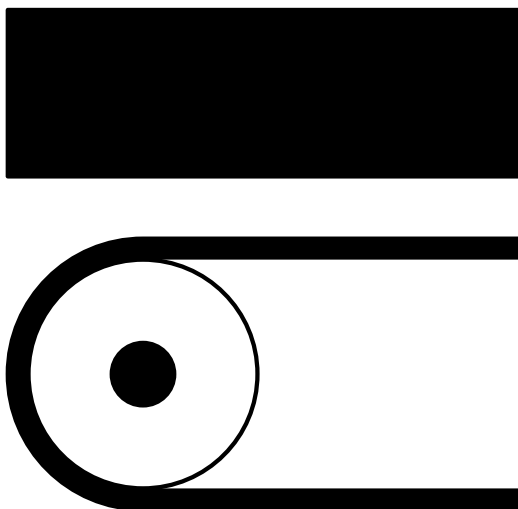
6.15 Slip correction

For actuators that use pulleys or belts, there is a function to correct slip at the start.

If RENV6.ADJ = 10b is set, the number of pulses set in RENV6.BR bits is output at FA speed and then the operation starts.

Slip correction is performed at every time start, regardless of changes in the direction of movement.

With RENV3.CU1B, CU2B, CU3B, CU4B bits, you can set whether to count the correction pulses in addition to the command pulses.



Adjust the RENV6.BR bit and RFA register settings in an experiment with the actual machine.

Name and description	Target
<Counter 1 during slip correction> 0: Does not count. 1: Count.	RENV3.CU1B(24)
<Counter 2 during slip correction> 0: Does not count. 1: Count.	RENV3.CU2B(25)
<Counter 3 during slip correction> 0: Does not count. 1: Count.	RENV3.CU3B(26)
<Counter 4 during slip correction> 0: Does not count. 1: Count.	RENV3.CU4B(27)
<Slip correction amount> The setting range is 0 to 4,095.	RENV6.BR(11: 0)
<Function to correct the feed amount> 00b: Does not correct the feed amount. 10b: Slip correction.	RENV6.ADJ(13,12)

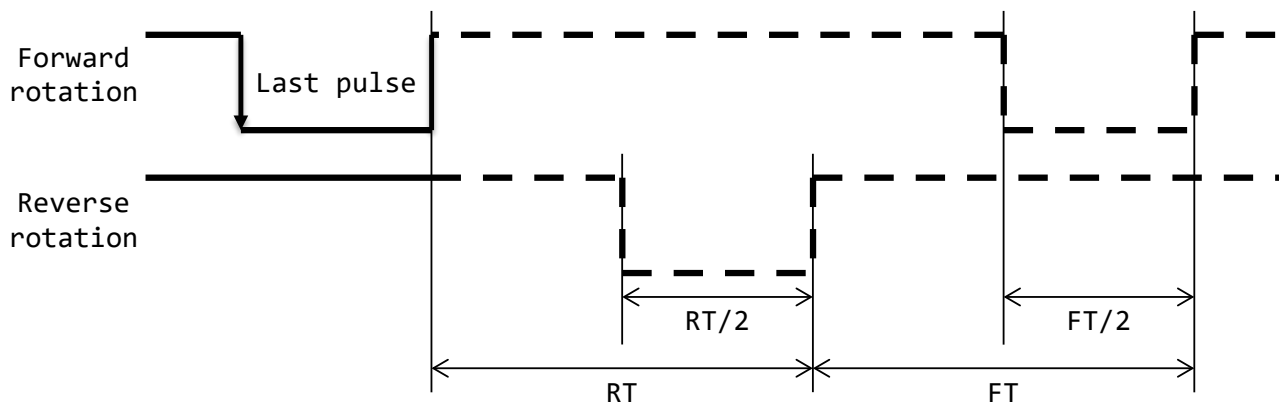
6.16 Vibration suppression

Immediately after the operation mode is completed, PCL6046 outputs one pulse reverse rotation and one pulse forward rotation to suppress vibration.

The vibration suppression pulse can be output at the timing to suppress the vibration generated by the final pulse, which can reduce the settling time. However, as the load conditions change, so does the optimal timing.

The vibration suppression pulse is output when 1 or more is set in both RENV7.RT bit and RENV7.FT bit.

In the +direction operation, the broken line in the figure below will be the pulses added by this function.



Adjust the RENV7.RT bit and RENV7.FT bit settings in an experiment with the actual machine.

Name and description	Target
<Cycle of reverse rotation pulse> The cycle of a reverse rotation pulse is set at x 32 cycles of the CLK signal. The setting range is 0 to 65,535.	RENV7.RT(15:0)
<Cycle of forward rotation pulse> The cycle of a forward rotation pulse is set at x 32 cycles of the CLK signal. The setting range is 0 to 65,535.	RENV7.FT(31:16)

6.17 Synchronous start

At the start timing after writing the start command, you can start an operation in synchronization with the stop of specified axis.

You can also start with the output of an internal sync signal.

6.17.1 Start by stopping the specified axis

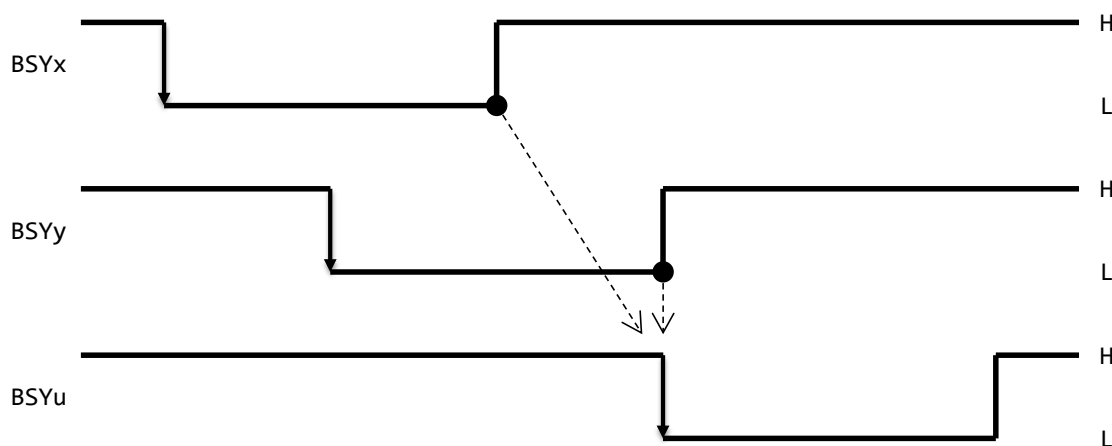
If $RMD.MSY = 11b$ is set, $RSTS.CND = 0100b$ will be set after writing the start command.

An operation starts when the axis which has been set in $RMD.MAX$ bit stops.

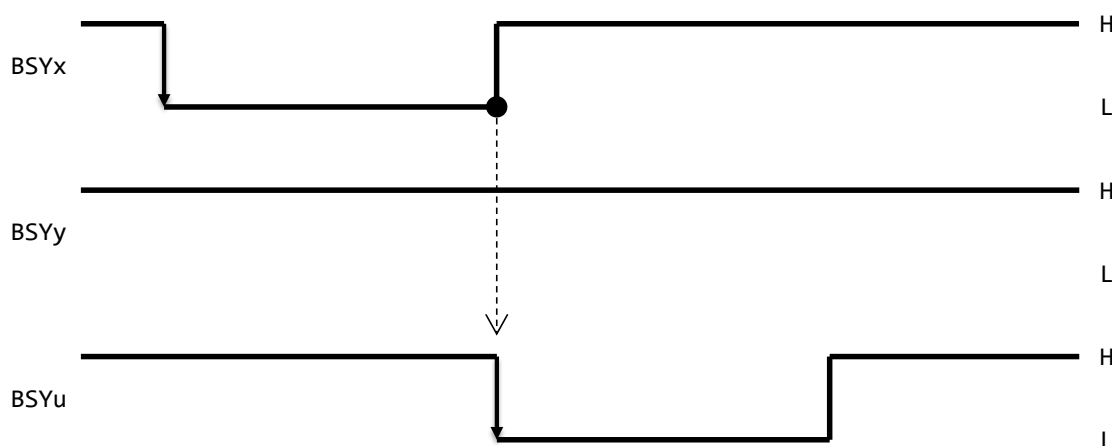
Setting Example:

When you start X-axis and Y-axis after setting 1. to 3. below, U-axis will start when both axes stop.

1. Set the start by stopping the specified axis ($PRMD.MSY = 11b$) to U-axis.
2. Set the X-axis and Y-axis ($PRMD.MAX = 0011b$) as the specified axes to U-axis.
3. Write the start command on U-axis.



If one of the axes set by $RMD.MAX$ bit starts and stops, the condition is satisfied even if the remaining axes do not start.



Name and description	Target
<Start timing after writing the start command> 11b: Starts when the specified axis (RMD.MAX) stops.	RMD.MSY(19,18)
<Axis to confirm stop when RMD.MSY = 11b> Example: 0001b: Starts when X-axis stops. 0010b: Starts when Y-axis stops. 0100b: Starts when Z-axis stops. 1000b: Starts when U-axis stops. 0101b: Starts when both X-axis and Z-axis stop. 1111b: Starts when all axes stop.	RMD.MAX(23:20)
<Operating status> 0100b: Waiting for other axis to stop.	RSTS.CND(3:0)

6.17.1.1 Stop selection of own axis

For start by stopping the specified axis, you can set whether to include own axis stop for the condition with RENV2.SMAX bit.

Name and description	Target
<Functional specifications when RMD.MAX bit includes its own axis with RMD.MSY = 11b> 0: When own axis is included in RMD.MAX bit, the operation will not start if the own axis stops at the end. 1: If own axis is included in RMD.MAX bit, the operation will start even if the own axis stops at the end.	RENV2.SMAX(29)

6.17.1.1.1 Do not include the stop of the own axis in the condition

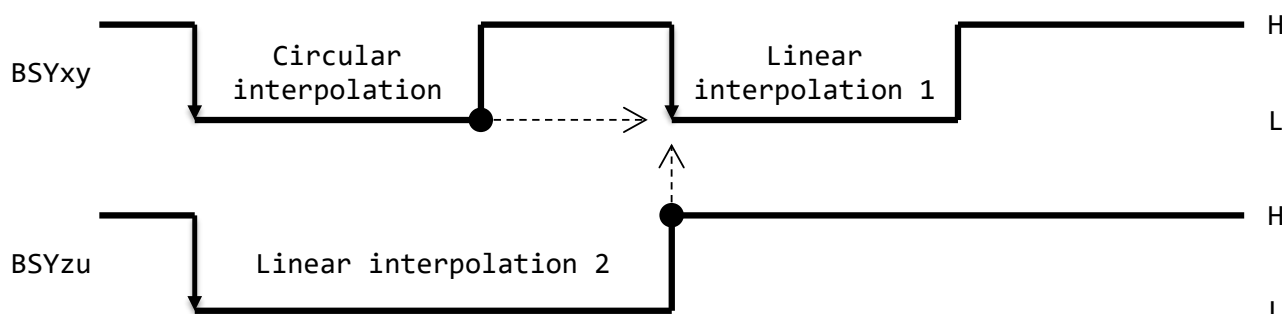
If you do not include the stop of own axis in the condition, you can set RENV2.SMAX = 0 (similar to PCL6045, a conventional PCL product).

Operation example 1-1:

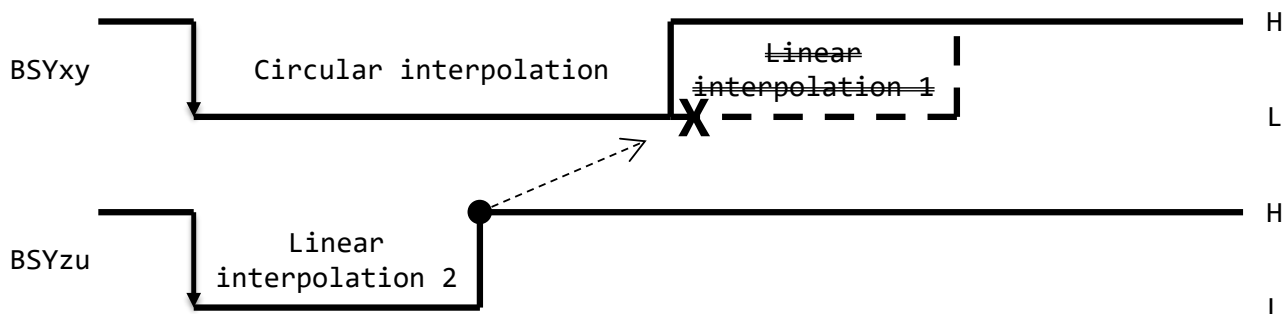
After starting circular interpolation by X-axis and Y-axis, and then starting linear interpolation 2 by Z-axis and U-axis, set the following 1 and 2.

1. Set PRMD = 00FC0061h (PRMD.MAX = 1111b, MSY = 11b, MOD = 61h) to X-axis and Y-axis .
2. Write a start command to X and Y axes.

When linear interpolation 2 stops after the circular interpolation stops, linear interpolation 1 (RMD.MOD = 61h) will start.



If linear interpolation 2 stops before the circular interpolation stops, linear interpolation 1 (RMD.MOD = 61h) will not start.

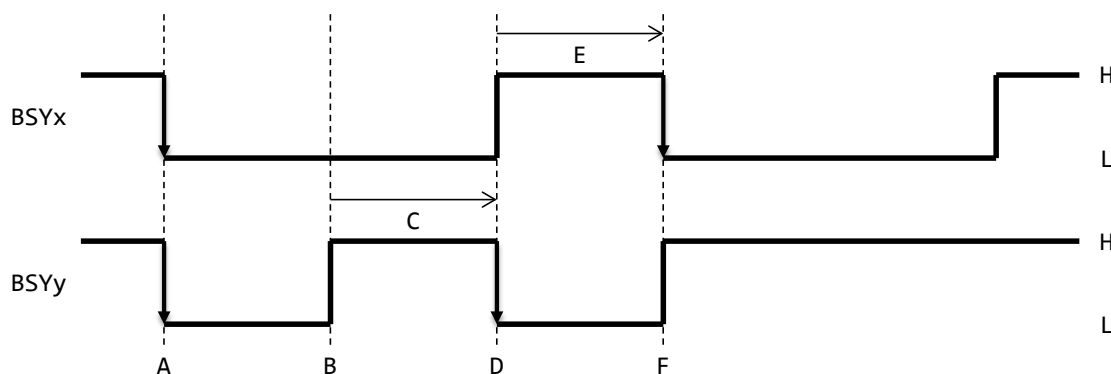


Operation example 1-2:

Set the X-axis operating time > Y-axis operating time, and set 1 to 4 below.

1. Set PRMD = 00040041h (PRMD.MAX = 0000b, MSY = 01b, MOD = 41h) to X-axis and Y-axis.
2. Write a start command to X-axis and Y-axis.
3. Set PRMD = 003C0041h (PRMD.MAX = 0011b, MSY = 11b, MOD = 41h) to X-axis and Y-axis.
4. Write a start command to X-axis and Y-axis.

- A) If writing SPSTA (032Ah) command to X-axis and Y-axis, X-axis and Y-axis will start at the same time.
- B) Y-axis stops before X-axis.
- C) Y-axis waits for X-axis to stop.
- D) Y-axis starts when X-axis stops.
- E) X-axis waits for Y-axis to stop.
- F) X-axis starts when Y-axis stops.



6.17.1.1.2 Include the stop of the own axis in the condition

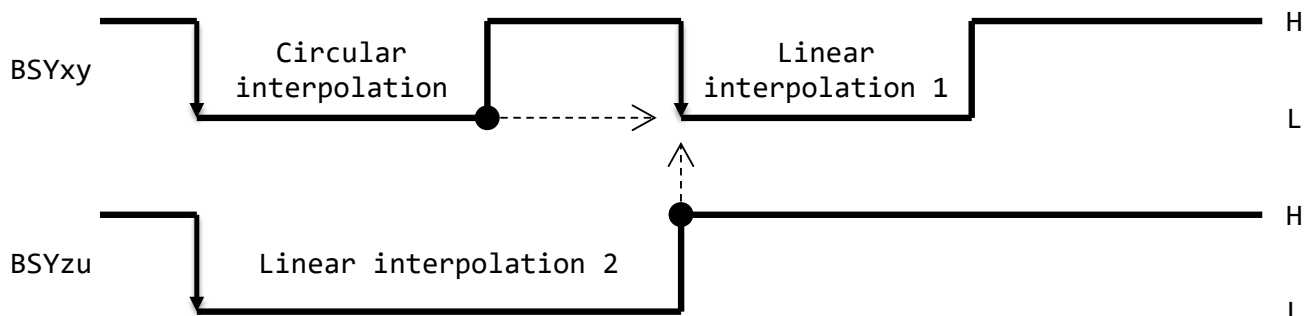
If you want to include the stop of the own axis in the condition, set RENV2.SMAX = 1 (installed from PCL6045B, a conventional PCL product).

Operation example 2-1:

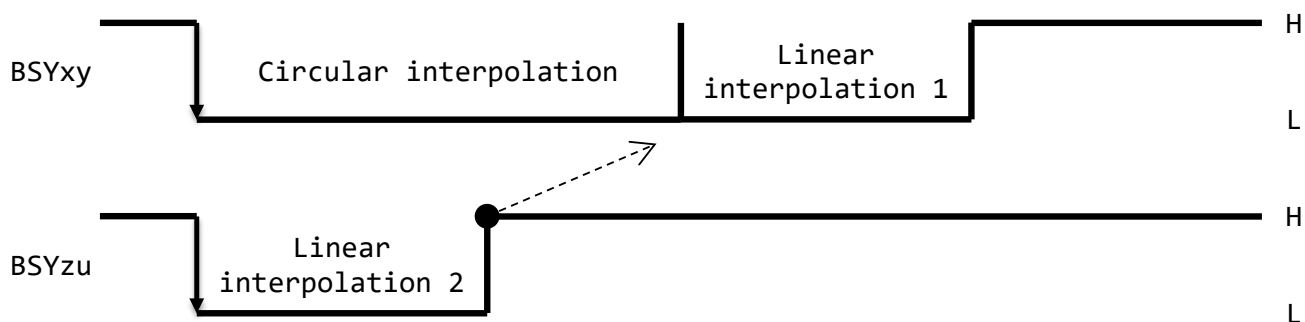
After starting circular interpolation by X-axis and Y-axis, and then starting linear interpolation 2 by Z-axis and U-axis, set the following 1 and 2.

1. Set PRMD = 00FC0061h (PRMD.MAX = 1111b, MSY = 11b, MOD = 61h) to X-axis and Y-axis.
2. Write a start command to X-axis and Y-axis.

When linear interpolation 2 stops after the circular interpolation stops, linear interpolation 1 (RMD.MOD = 61h) will start.



Even if linear interpolation 2 stops before circular interpolation stops, linear interpolation 1 (RMD.MOD = 61h) will start.

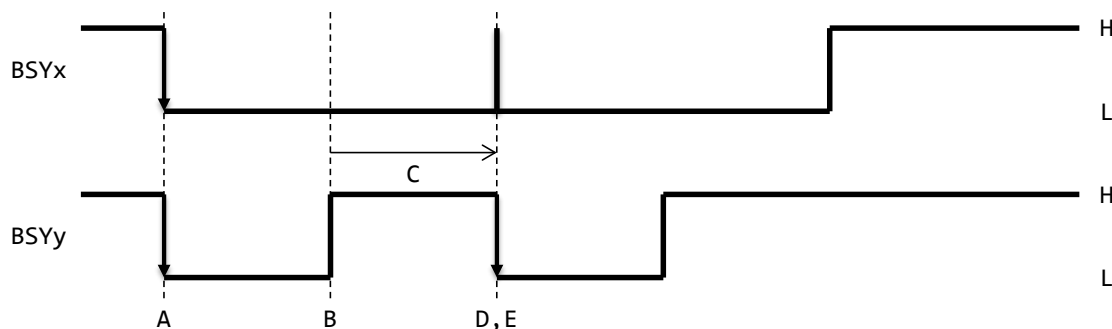


Operation example 2-2:

Set the X-axis operating time > Y-axis operating time, and set 1 to 4 below.

1. Set PRMD = 00040041h (PRMD.MAX = 0000b, MSY = 01b, MOD = 41h) to X-axis and Y-axis.
2. Write a start command to X and Y axes.
3. Set PRMD = 003C0041h (PRMD.MAX = 0011b, MSY = 11b, MOD = 41h) to X-axis and Y-axis.
4. Write the start command to X-axis and Y-axis.

- A) If writing SPSTA (032Ah) command to X-axis and Y-axis, X-axis and Y-axis will start at the same time.
- B) Y-axis stops before X-axis.
- C) Y-axis waits for X-axis to stop.
- D) Y-axis starts when X-axis stops.
- E) X-axis also starts when own axis stops.



6.17.1.2 Continuous interpolation without changing the interpolation axes

In the continuous interpolation that does not change the combination of interpolation axes, it is not necessary to set the start by stopping the specified axis. Since all interpolation axes stop at the same time, continuous interpolation can be performed simply by setting the continuous operation in the pre-register.

Setting example:

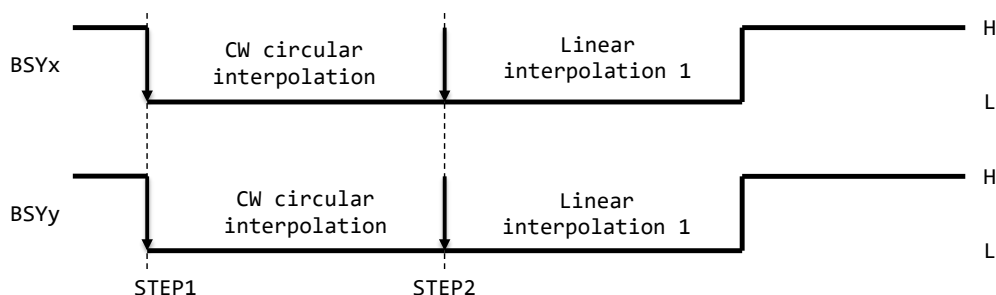
Set the continuous interpolation (from the circular interpolation by X-axis and Y-axis to the linear interpolation 1) that does not change the interpolation axis combination. (Setting in speed control register is omitted)

STEP	Writing target	X-axis value	Y-axis value	Description
1	PRMV	10000	10000	X-axis and Y-axis have target coordinates of 10000,10000 (90-degrees).
	PRIP	10000	0	Center coordinates 10000,0.
	PRMD	00000064h	00000064h	CW circular interpolation by X-axis and Y-axis.
	COMW	0351h	-	Set STAFH (51h) command to X-axis and Y-axis.
2	PRMV	10000	5000	Incremental movement 10000,5000.
	PRMD	00000061h	00000061h	Linear interpolation 1 by X-axis and Y-axis.
	COMW	0351h	-	Set STAFH (51h) command to X-axis and Y-axis.

When STEP1 is set, CW circular interpolation (radius 10000, 90-degrees) by X-axis and Y-axis will start.

When STEP2 is set while STEP1 is in operation, it waits for STEP1 to complete.

When STEP1 is completed, linear interpolation 1 (10000, 5000) by X-axis and Y-axis will start.



6.17.1.3 Continuous interpolation 1 to change the interpolation axes

In the continuous interpolation that changes the combination of interpolation axes, it is necessary to set the dummy operation data and the start by stopping specified axis. The dummy operation is an incremental movement of positioning control that sets $RMV = 0$.

When $RENV2.SMAX = 0$ (same as the conventional PCL6045), set the dummy operation data to the axis to change the combination. The axis that changes the combination waits for the other interpolation axes to stop.

If the dummy operation of all axes is not entered, the continuous operation may stop or the interpolation operation may not stop.

Setting example:

Set the continuous interpolation (from the circular interpolation by X-axis and Y-axis to the linear interpolation 1 and the linear interpolation by X-axis and Z-axis) that changes the combination of interpolation axes.

(Setting in speed control register is omitted)

STEP	Writing target	X-axis value	Y-axis value	Z axis value	Description
1	PRMV	10000	10000	0	X-axis and Y-axis have the target coordinates:10000,10000 (90-degrees). Z axis: 0 feed amount
	PRIP	10000	0	-	Center coordinates: 10000, 0.
	PRMD	00000064h	00000064h	003C0041h	CW circular interpolation by X-axis and Y-axis. Z-axis: dummy operation.
	COMW	0751h	-	-	STAFH (51h) command to all axes. X-axis and Y-axis start. Z-axis waits for X-axis and Y-axis (STEP1) to stop.
2	PRMV	10000	5000	0	Incremental movement,10000, 5000 of X-axis and Y-axes. Z axis: 0 movement.
	PRMD	004C0061h	004C0061h	003C0041h	Linear interpolation 1 by X-axis and Y-axis. Z-axis: dummy operation.
	COMW	0751h	-	-	STAFH (51h) command to all axes. X-axis and Y-axis wait for Z-axis (STEP1) to stop. Z-axis waits for X-axis and Y-axis (STEP2) to stop.
3	PRMV	10000	-	-5000	Incremental movement 10000, -5000 of X-axis and Z-axis.
	PRMD	004C0061h	-	00000061h	Linear interpolation 1 by X-axis and Z-axis.
	COMW	0551h	-	-	STAFH (51h) command to X-axis and Z-axis. X-axis waits for Z-axis (STEP2) to stop. Y-axis will not start. Z axis waits only for the shift of the pre-register.

When STEP1 is set, CW circular interpolation (10000 radius, 90-degrees) by X-axis and Y-axis will start.

When CW circular interpolation of STEP1 is completed, Z-axis dummy operation starts.

If STEP2 is set while STEP1 is operating, completion of the dummy operation of STEP1 will be waited.

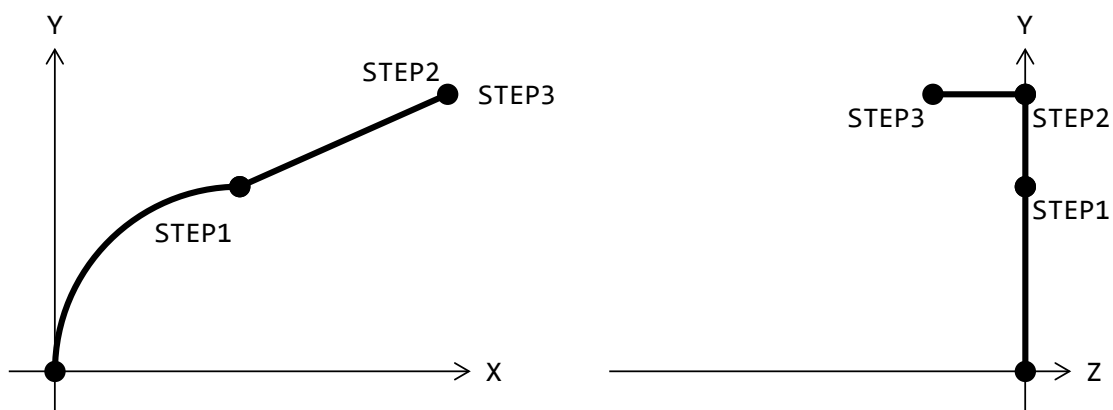
The dummy operation of STEP1 is stopped immediately, and the linear interpolation 1 (10000, 5000) of STEP2 starts.

When the linear interpolation 1 of STEP2 is completed, Z-axis dummy operation starts.

If STEP3 is set while STEP1 or STEP2 is operating, completion of the dummy operation of STEP2 will be waited.

The dummy operation of STEP2 stops immediately, and the linear interpolation 1 (10000, -5000) of STEP3 starts.

When the linear interpolation 1 of STEP3 is completed, the continuous interpolation is completed.



6.17.1.4 Continuous interpolation 2 to change the interpolation axes

In the continuous interpolation that changes the combination of interpolation axes, it is necessary to set the dummy operation data and the start by stopping specified axis. The dummy operation is an incremental movement of the positioning control that sets $RMV = 0$.

When $RENV2.SMAX = 1$ (installed from the conventional PCL6045B), set dummy operation data to all axes of the combination. Enter the dummy operation data to all axes immediately before the continuous interpolation that changes the combination of interpolation axes. If the dummy operation is not entered, the continuous operation may stop or the interpolation operation may not stop.

Setting example:

Set the continuous interpolation (from the circular interpolation by X-axis and Y-axis to the linear interpolation 1 and the linear interpolation by X-axis and Z-axis) that changes the combination of interpolation axes.

(Setting in speed control register is omitted)

STEP	Writing target	X-axis value	Y-axis value	Z axis value	Description
1	PRMV	10000	10000	0	X-axis and Y-axis have the target coordinates: 10000,10000 (90-degrees). Z axis: 0 movement.
	PRIP	10000	0	-	Center coordinates 10000,0.
	PRMD	00000064h	00000064h	00000041h	CW circular interpolation by X-axis and Y-axis. Z-axis: dummy operation.
	COMW	0751h	-	-	STAFH (51h) command to all axes. All axes start.
2	PRMV	10000	5000	0	Incremental movement 10000,5000 of X-axis and Y-axis. Z axis: 0 feed amount.
	PRMD	007C0061h	007C0061h	007C0041h	Linear interpolation 1 by X-axis and Y-axis. Z-axis: dummy operation.
	COMW	0751h	-	-	STAFH (51h) command to all axes. All axes wait for all axes (STEP1) to stop.
3-1	PRMV	0	0	0	Since the combination of interpolation axes is changed, the feed amount 0.
	PRMD	007C0041h	007C0041h	007C0041h	Dummy operations of all axes.
	COMW	0751h	-	-	STAFH (51h) command to all axes. All axes wait for all axes (STEP2) to stop.
3-2	PRMV	10000	0	-5000	Incremental movement 10000, -5000 of X-axis and Z-axis. The feed amount 0 on Y-axis.
	PRMD	007C0061h	007C0041h	007C0061h	Linear interpolation 1 by X-axis and Z-axis. Y-axis: dummy operation.
	COMW	0751h	-	-	STAFH (51h) command to all axes. All axes wait for all axes (STEP3-1) to stop.

When STEP1 is set, CW circular interpolation (radius 10000, 90-degrees) by X-axis and Y-axis and the dummy operation in Z-axis will start.

The dummy operation in STEP1 will stop immediately.

If STEP2 is set while STEP1 is operating, it waits all axes of STEP1 to be completed.

When all axes of STEP1 are completed, the linear interpolation 1 (10000, 5000) of STEP2 and the dummy operation of Z axis will be started.

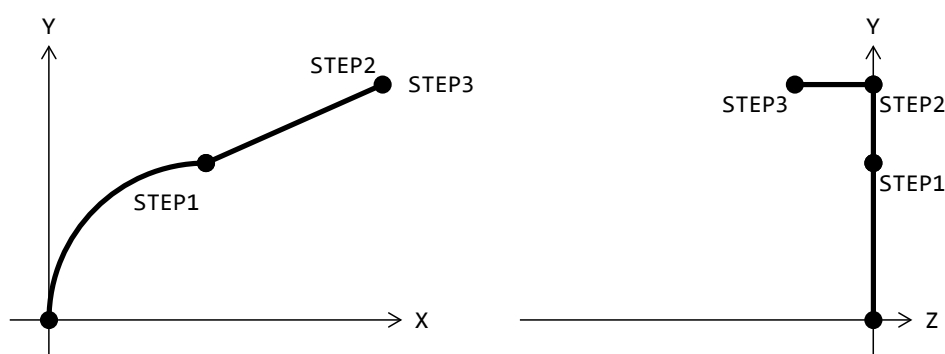
The dummy operation of STEP2 will stop immediately.

If STEP3 is set while STEP1 or STEP2 is operating, it waits all axes of STEP2 to be completed.

When the all axes of STEP2 are completed, the linear interpolation 1 (10000, -5000) of STEP3 and the dummy operation of Y-axis are started.

The dummy operation of STEP3 will stop immediately.

When all the axes of STEP3 are completed, the continuous interpolation will be completed.



The trajectory is the same as "6.17.1.3 Continuous interpolation 1 that changes the interpolation axis".

6.17.2 Start with internal sync signal

When $RMD.MSY = 10b$ is set, $RSTS.CND = 0011b$ will be set after writing the start command.

The operation starts when the axes that are set in $RENV5.SYI$ bit outputs an internal synchronization signal.

The output timing of an internal synchronization signal can be selected from nine types of signals with $RENV5.SYO$ bit.

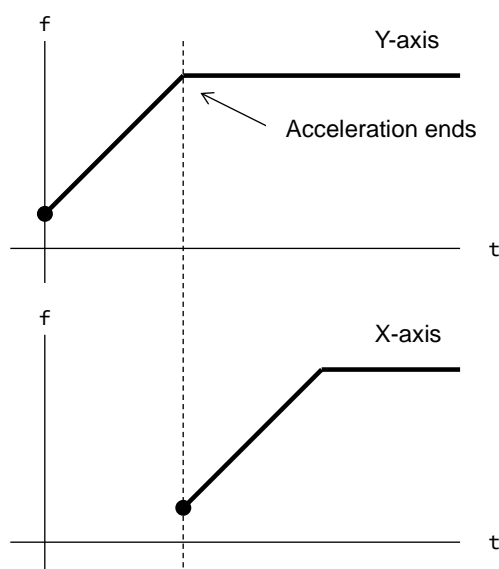
Nine types of signals can be checked through the general-purpose input/output pin, and an event interrupt request (RIRQ) can be set. The generated interrupt can be read by event interrupt factor (RIST).

Setting example 1:

When the acceleration of Y-axis ends, X-axis will start.

After you set the following 1 to 3 below, start X-axis and Y-axis.

1. Set the start with the internal synchronization signal ($PRMDx.MSY = 10b$).
2. Set the use of internal synchronization signal of Y-axis ($RENV5x.SYI = 01b$).
3. Set the internal synchronization signal output ($RENV5y.SYO = 1001b$) when an acceleration ends.

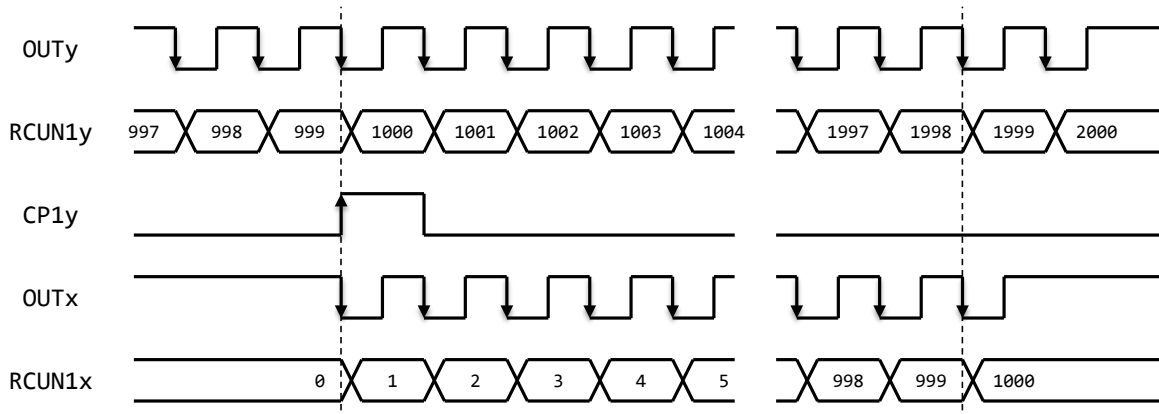


Setting example 2:

When the comparator 1 condition of Y-axis is satisfied, X-axis will start.

After you set 1 to 7 below, start X-axis and Y-axis.

1. Set the start with the internal synchronization signal ($PRMDx.MSY = 10b$).
2. Set the use of internal synchronization signal of Y-axis ($RENV5x.SYI = 01b$).
3. Set the output of the internal synchronization signal to Y-axis ($RENV5y.SYO = 0001b$) when the Comparator 1 condition is satisfied.
4. Set the $RCUN1$ ($RENV4y.C1C = 00b$) register for comparison with Comparator 1 to Y-axis.
5. Set the comparison condition of Comparator 1 to be equal to the comparison target regardless of the counting direction ($RENV4y.C1S = 001b$) to Y-axis.
6. Set the processing of Comparator 1 to be "No action" ($RENV4y.C1D = 00b$) to Y-axis.
7. Set the comparator 1 comparison value ($RCMP1y = 1000$) to Y-axis.



In this example, setting $PRMV_y = 2000$ and $PRMV_x = 1000$ results in $RCUN1x = 1$ at $RCUN1y = 1000$.

Therefore, when $RCUN1Y = 1999$, it will be $RCUN1x = 1000$, and X-axis stops 1 pulse short of the Y-axis.

When setting $RCUN1x = 1000$ with $RCUN1y = 2000$, set “the comparison target size is larger ($RENV4y.C1S = 11b$)”.

Alternatively, set $RCMP1y = 1001$.

Name and description	Target
<Start timing after writing the start command> 10b: Start with internal synchronization signal ($RENV5.SYI$).	$RMD.MSY(19,18)$
<P0n pin input/output function> 10b: FUP signal is output at negative logic during acceleration.	$RENV2.P0M(1,0)$
<P1n pin input/output function> 10b: FDW signal is output at negative logic during deceleration.	$RENV2.P1M(3,2)$
<P3n pin input/output function> 10b: CP1 signal is output with negative logic when the comparator 1 condition is satisfied. 11b: CP1 signal is output with positive logic when the comparator 1 condition is satisfied.	$RENV2.P3M(7,6)$
<P4n pin input/output function> 10b: CP2 signal is output at negative logic when the comparator 2 condition is satisfied. 11b: CP2 signal is output at positive logic when the comparator 2 condition is satisfied.	$RENV2.P4M(9,8)$
<P5n pin input/output function> 10b: CP3 signal is output at negative logic when the comparator 3 condition is satisfied. 11b: CP3 signal is output at positive logic when the comparator 3 condition is satisfied.	$RENV2.P5M(11,10)$
<P6n pin input/output function> 10b: CP4 signal is output at negative logic when the comparator 4 condition is satisfied. 11b: CP4 signal is output at positive logic when the comparator 4 condition is satisfied.	$RENV2.P6M(13,12)$
<P7n pin input/output function> 10b: CP5 signal is output at negative logic when the comparator 5 condition is satisfied. 11b: CP5 signal is output at positive logic when the comparator 5 condition is satisfied.	$RENV2.P7M(15,14)$
<Input target of internal sync signal> 00b: X-axis internal sync signal 10b: Z axis internal sync signal 01b: Y-axis internal sync signal 11b: U-axis internal sync signal	$RENV5.SYI(21,20)$

Name and description	Target
<Output timing of internal synchronous signal> 0001b: When comparator 1 condition is satisfied 0010b: When comparator 2 condition is satisfied 0011b: When comparator 3 condition is satisfied 0100b: When comparator 4 condition is satisfied 0101b: When comparator 5 condition is satisfied 1000b: When the acceleration starts 1001b: When the acceleration ends 1010b: When the deceleration starts 1011b: When the deceleration ends Other: No output of internal synchronization signal	RENV5.SYO(19:16)
<Event interrupt request (IRUS)> 1: An interrupt is generated when acceleration is started. (SSTS.SFU bit changed from 0 to 1)	RIRQ.IRUS(4)
<Event interrupt request (IRUE)> 1: An interrupt is generated when acceleration is completed. (SSTS.SFU bit changed from 1 to 0)	RIRQ.IRUE(5)
<Event interrupt request (IRDS)> 1: An interrupt is generated when deceleration is started. (SSTS.SFD bit changed from 0 to 1)	RIRQ.IRDS(6)
<Event interrupt request (IRDE)> 1: An interrupt is generated when deceleration is completed. (SSTS.SFD bit changed from 1 to 0)	RIRQ.IRDE(7)
<Event interrupt request (IRC1)> 1: An interrupt is generated when the comparison condition of comparator 1 is satisfied. (MSTS.SCP1 changed from 0 to 1)	RIRQ.IRC1(8)
<Event interrupt request (IRC2)> 1: An interrupt is generated when the comparison condition of comparator 2 is satisfied. (MSTS.SCP2 changed from 0 to 1)	RIRQ.IRC2(9)
<Event interrupt request (IRC3)> 1: An interrupt is generated when the comparison condition of comparator 3 is satisfied. (MSTS.SCP3 changed from 0 to 1)	RIRQ.IRC3(10)
<Event interrupt request (IRC4)> 1: An interrupt is generated when the comparison condition of comparator 4 is satisfied. (MSTS.SCP4 changed from 0 to 1)	RIRQ.IRC4(11)
<Event interrupt request (IRC5)> 1: An interrupt is generated when the comparison condition of comparator 5 is satisfied. (MSTS.SCP5 changed from 0 to 1)	RIRQ.IRC5(12)
<Event interrupt factor (ISUS)> 1: Acceleration has started. (SSTS.SFU bit changed from 0 to 1)	RIST.ISUS(4)

Name and description	Target
<Event interrupt factor (ISUE)> 1: The acceleration has ended. (SSTS.SFU bit changed from 1 to 0)	RIST.ISUE(5)
<Event interrupt factor (ISDS)> 1: Deceleration has started. (SSTS.SFD bit changed from 0 to 1)	RIST.ISDS(6)
<Event interrupt factor (ISDE)> 1: The deceleration has been completed. (SSTS.SFD bit changed from 1 to 0)	RIST.ISDE(7)
<Event interrupt factor (ISC1)> 1: The comparison condition of comparator 1 was satisfied. (MSTS.SCP1 changed from 0 to 1)	RIST.ISC1(8)
<Event interrupt factor (ISC2)> 1: The comparison condition of comparator 2 was satisfied. (MSTS.SCP2 changed from 0 to 1)	RIST.ISC2(9)
<Event interrupt factor (ISC3)> 1: The comparison condition of Comparator 3 was satisfied. (MSTS.SCP3 changed from 0 to 1)	RIST.ISC3(10)
<Event interrupt factor (ISC4)> 1: The comparison condition of Comparator 4 was satisfied. (MSTS.SCP4 changed from 0 to 1)	RIST.ISC4(11)
<Event interrupt factor (ISC5)> 1: The comparison condition of Comparator 5 was satisfied. (MSTS.SCP5 changed from 0 to 1)	RIST.ISC5(12)
<Motion status> 0011b : Waiting for internal synchronization signal input	RSTS.CND(3:0)

6.18 Interrupt request (INT)

From INT pin, INT signals that perform interrupt requests can be output.

INT signal continues to be output until all the causes in all the axes that are interrupting are cleared.

There are 17 types of errors, 20 types of events, and 1 type of operation stop as the interrupt factors for each axis.

You can identify the interrupt generation axis and interrupt cause by the following procedures.

- (1) Check if any bit in MSTS.SENI, SERR, SINT is 1 in the main status of X-axis.
- (2) If MSTS.SENI = 1, it means that an operation stop interrupt has occurred.
- (3) MSTS. If SERR = 1, the interrupt cause can be identified in REST register.
- (4) MSTS. If SINT = 1, the interrupt cause can be identified in RIST register.
- (5) Repeat the above steps (1) to (4) for the remaining Y, Z, and U axes.

When a register is read by an interrupt routine, you can re-write the I/O buffer in the indirect access method.

Therefore, when accessing the I/O buffer by the main routine, the processing of the main routine is affected.

When accessing a register in an interrupt routine, implement a FIRO (stack) or the like for the countermeasure.

In full address method, the impact can be ignored by using the direct access method with either or both.

While processing the interrupt generation axis in steps (1) to (4) above, a new interrupt may occur on the processed axes.

In this case, if the CPU interrupt acceptance setting is edge trigger, the occurrence of this new interrupt will not be accepted.

Edge triggers can be supported using the RENV1.INTM bit.

1. Set RENV1.INTM = 1 to all axes.
2. The values in the main status and in the interrupt cause register do not change, and H level signal is output from INT pin.
3. Set RENV1.INTM = 0 to all axes.
4. If there is a new interrupt occurred, L level signal is output from INT pin and an edge trigger can be generated.

Alternatively, read the main status of all axes again and check MSTS.SINT = 1 before the end of the interrupt routine.

If you do not use INT pin, set it open.

Even when using multiple PCL6046s, INT pins cannot be wired or connected to each other. (INT ≠ Hi-Z)

Name and description	Target
<INT pin output function> 0: When an interrupt factor occurs, L level is output from INT pin. 1: Even if an interrupt factor occurs, H level is output from INT pin.	RENV1.INIM(29)

6.18.1 Error interrupt

The error interrupt factor occurs when only one condition is satisfied.

When an error interrupt factor occurs, the corresponding bit in REST register becomes 1.

When any bit of REST register is 1, the L level can be output from INT pin.

In REST register, writing 1 to the corresponding bit can clear the bit to 0.

If RENV5.ISMR = 0 is set, REST register is cleared to 0 even by writing RREST (F2h) command.

If RENV5.ISMR = 1 is set, it will not be cleared by writing RREST (F2h) command.

(The setting of RENV5.ISMR bit also affects RIST register)

Name and description	Target
<p><Error interrupt factor></p> <p>For REST register, see "5.4.7.2 REST: Error interrupt factor".</p>	REST register
<p><How to clear the bits of the RIST register and REST register></p> <p>0: Write the read command to each register to clear each register to 0. Even with the full-address direct access method, each register can be cleared to 0 by writing a read command.</p> <p>1: Writing read command to each register does not clear the register to 0. In either case, you can write 1 to the corresponding bit in each register to clear it to 0.</p>	RENV5.ISMR(23)
<p><Main status (SERR)></p> <p>0: No error interrupt occurred. 1: An error interrupt occurred. L level can be output from INT pin. When all of the bits that are 1 in REST register become 0, it returns to MST.SERR = 0.</p>	MSTS.SERR(4)
<p><Register write (REST)></p> <p>Writes the contents of I/O buffer to REST register. Writing 1 to the corresponding bit will clear the bit to 0.</p>	WREST(B2h)
<p><Register Read (REST)></p> <p>Reads out the contents of REST register to I/O buffer. Setting RENV5.ISMR = 0 clears REST register to 0. In the direct access method, the bits before reading the register by direct access are also cleared. Also check I/O buffers if necessary.</p>	RREST(F2h)

6.18.2 Event interrupt

The event interrupt factor occurs when the condition of RIRQ register is satisfied.

When an event interrupt factor occurs, the corresponding bit in RIST register becomes 1.

When any bit of RIST register is 1, L level can be output from INT pin.

In RIST register, writing 1 to the corresponding bit clears the bit to 0.

Setting RENV5.ISMR = 0 also clears RIST register to 0 by writing RRIST (F3h) command.

Setting RENV5.ISMR = 1 will not clear RIST register by writing RRIST (F3h) command.

(The setting of RENV5.ISMR bit also affects REST register)

Name and description	Target
<p><Event interrupt factor></p> <p>For details on RIST register, see "5.4.7.3 RIST: Event interrupt factor".</p>	RIST register
<p><How to clear the bits of RIST register and REST register></p> <p>0: Write the read command to each register to clear the register to 0.</p> <p>Even in full-address direct access method, each register can be cleared to 0 by writing a read command.</p> <p>1: Writing the read command to each register does not clear the register to 0.</p> <p>In either case, you can write 1 to the corresponding bit in each register to clear it to 0.</p>	RENV5.ISMR(23)
<p><Main status (SINT)></p> <p>0: No event interrupt occurred.</p> <p>1: An event interrupt occurred. L level can be output from INT pin.</p> <p>When all of the bits that are 1 in the RIST register become 0, it returns to MST.SINT = 0.</p>	MSTS.SINT(5)
<p><Register write (RIST)></p> <p>Writes the contents of I/O buffer to RIST register.</p> <p>Writing 1 to the corresponding bit clears the bit to 0.</p>	WRIST(B3h)
<p><Register read (RIST)></p> <p>Reads out the contents of RIST register to I/O buffer.</p> <p>Setting RENV5.ISMR = 0 clears RIST register to 0.</p> <p>In direct access method, the bits before reading the register by direct access will also be cleared.</p> <p>Also check the I/O buffer if necessary.</p>	RRIST(F3h)

6.18.3 Operation stop interrupt

The operation stop interrupt factor occurs when an operation is stopped by RENV2.IEND = 1 setting.

When an interrupt factor occurs, MSTS.SENI bit becomes 1.

When MSTS.SENI bit is 1, L level can be output from INT pin.

MSTS.SENI bit will be cleared to 0 if you write SENIR (2Dh) command.

If you set RENV5.MSMR = 0, MSTS.SENI bit is cleared to 0 even in reading the main status.

If you set RENV5.MSMR = 1, MSTS.SENI bit is not cleared to 0 in reading the main status.

(The setting of RENV5.MSMR bit also affects MSTS.SEOR bit)

There is no difference to distinct between normal stops and abnormal stops as the cause of an operation stop interrupt.

An interrupt by normal stops is one of the event interrupt factors, but it needs to be determined by reading RIST register.

If you do not need to distinguish between a normal stop and an abnormal stop, you can know the completion of an operation mode only by the operation stop interrupt.

If RMD.MENI = 1 is set, MSTS.SENI = 1 will not be set during continuous operation.

An operation stop interrupt can be generated when a series of operation modes such as continuous interpolation are completed.

Name and description	Target
<Operation stop interrupt during continuous operation> 0: When RSTS.PFM = 10b or 11b, MSTS.SENI = 1 will be set. 1: When RSTS.PFM = 10b or 11b, MSTS.SENI = 1 will not be set. Even if the stop interrupt is enabled (RENV2.IEND = 1), the stop interrupt (MSTS.SENI) bit can be disabled when the pre-register is determined (RSTS.PFM = 10b or 11b).	RMD.MENI(7)
<Functional specifications of MSTS.SENI bit> 0: Invalid. Keep MSTS.SENI = 0 by stopping the operation. 1: Valid. Change to MSTS.SENI = 1 by stopping the operation.	RENV2.IEND(27)
<How to clear MSTS.SENI bit and MSTS.SEOR bit> 0: Cleared automatically if the main status is read. 1: Not cleared automatically even if the main status is read. MSTS.SENI bit can be cleared manually by writing SENIR (2Dh) command. MSTS.SEOR bit can be cleared manually by writing SEORR (2Eh) command.	RENV5.MSMR(25)
<Main status (SENI)> 0: No operation stop interrupt occurs. Or RENV2.IEND = 0 is set. 1: An operation stop interrupt occurs. L level can be output from INT pin. If RENV5.MSMR = 0, it returns to 0 within 3 cycles of the CLK signal after reading. If RENV5.MSMR = 1, it returns to 0 by writing SENIR (2Dh) command.	MSTS.SENI(2)
<Interrupt control command (SENI)> Clear to MSTS.SENI = 0.	SENIR(2Dh)

6.19 General-purpose one shot

P0n pin can output a general-purpose one-shot signal by setting RENV2.P0M = 11b.

If RENV2.P0L = 0 is set, a general-purpose one-shot signal with negative logic is output with P0RST (10h) command.

If RENV2.P0L = 1 is set, a general-purpose one-shot signal with positive logic is output with P0SET (18h) command.

P1n pin can also output a general-purpose one-shot signal by setting RENV2.P1M = 11b.

If RENV2.P1L = 0 is set, a general-purpose one-shot signal with negative logic is output with P1RST (11h) command.

If RENV2.P1L = 1 is set, a general-purpose one-shot signal with positive logic is output with P1SET (19h) command.

The output pulse width of a general-purpose one-shot signal is 23 to 25 ms.

Name and description	Target
<P0n pin input/output function> 11b: Outputs a general-purpose one-shot signal.	RENV2.P0M(1,0)
<P1n pin input/output function> 11b: Outputs a general-purpose one-shot signal.	RENV2.P1M(3,2)
<The logic of a general-purpose one-shot signal that can be output from P0n pin > 0: Negative logic. 1: Positive logic.	RENV2.P0L(16)
<The logic of a general-purpose one-shot signal that can be output from P1n pin > 0: Negative logic. 1: Positive logic.	RENV2.P1L(17)
<General-purpose output bit control command (P0RST)> Write 0 to OPT0 bit to reset P0n pin to L level. When RENV2.P0M = 11b and RENV2.P0L = 0, a general-purpose one-shot signal with negative logic is output.	P0RST(10h)
<General-purpose output bit control command (P1RST)> Write 0 to OPT1 bit to reset P1n pin to L level. When RENV2.P1M = 11b and RENV2.P1L = 0, a general-purpose one-shot signal with negative logic is output.	P1RST(11h)
<General-purpose output bit control command (P0SET)> Write 1 to OPT0 bit to set P0n pin to H level. When RENV2.P0M = 11b and RENV2.P0L = 1, a general-purpose one-shot signal with positive logic is output.	P0SET(18h)
<General-purpose output bit control command (P1SET)> Write 1 to OPT1 bit to set P1n pin to H level. When RENV2.P1M = 11b and RENV2.P1L = 1, a general-purpose one-shot signal with positive logic is output.	P1SET(19h)

7. Electrical Characteristics

7.1 Absolute maximum ratings

Item	Symbol	Rating	Unit	Remarks
Power supply voltage	V_{DD}	-0.3 to +4.0	V	-
Input voltage	V_I	-0.3 to +7.0	V	-
Output voltage	V_O	-0.3 to +7.0	V	-
Output current	I_{OUT}	± 30	mA	-
Storage temperature	T_{stg}	-65 to +150	°C	-

7.2 Recommend operating conditions

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks
Power supply voltage	V_{DD}	3.0	3.3	3.6	V	-
Input voltage	V_I	-0.3	-	+5.8	V	-
Operating Ambient temperature	T_{stg}	-40	-	+85	°C	$T_j = -40$ to $+125$ °C, $\theta_{j-a} = 24$ °C/W
Input rising time	T_r	-	-	50	ns	10% to 90% change time of power supply voltage
Input falling time	T_f	-	-	50	ns	10% to 90% change time of power supply voltage

7.3 DC characteristics

Item	Symbol	Conditions	Min.	Max.	Unit
Current consumption	I_{DD}	CLK = 30 MHz, before reset *1	-	230	mA
Output leakage current	I_{OZ}	-	-1	1	μ A
Pin capacitance	C	-	-	10	pF
Low level input current	I_{IL}	*2	-90	-	μ A
		*3	-1	-	
High level input current	I_{IH}	-	-	1	μ A
Low level input voltage	V_{IL}	$V_{DD} = \text{Min.}$	-0.3	0.8	V
High level input voltage	V_{IH}	$V_{DD} = \text{Max.}$	2.0	5.8	V
Low level output voltage	V_{OL}	$I_{OL} = 6\text{mA}$	-	0.4	V
High level output voltage	V_{OH}	$I_{OH} = -6\text{mA}$	*4	$V_{DD}-0.4$	V
Low level output current	I_{OL}	$V_{OL} = 0.4\text{V}$	-	6	mA
High level output current	I_{OH}	$V_{OH} = V_{DD}-0.4\text{V}$	-6	-	mA
internal pull-up resistance	R_{PU}	$V_I = \text{GND}$	*2	40	240 kohm

As for the sign of current, the positive number indicates the inflow current value, and the negative number indicates the outflow current value.

*1 Before resetting, the current consumption reduction circuit does not operate, so the current consumption is maximized.

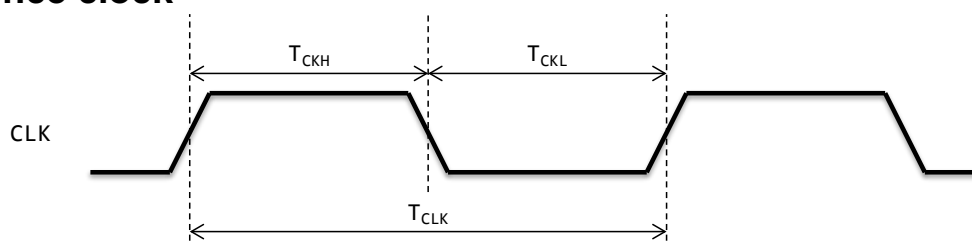
*2 IF0, IF1, CSTA, CSTP, CEMG, ELLn, +ELn, -ELn, +SDn, -SDn, ORGn, ALMn, EAn, EBn, EZn, PAn, PBn, PEn, +DRn, -DRn, PCSn, INPn, CLRN, LTCn, P0n, P1n, P2n, P3n, P4n, P5n, P6n, P7n pins.

*3 Input pins and bidirectional pins other than the above.

*4 Output pin and bidirectional pin.

7.4 AC characteristics

7.4.1 Reference clock

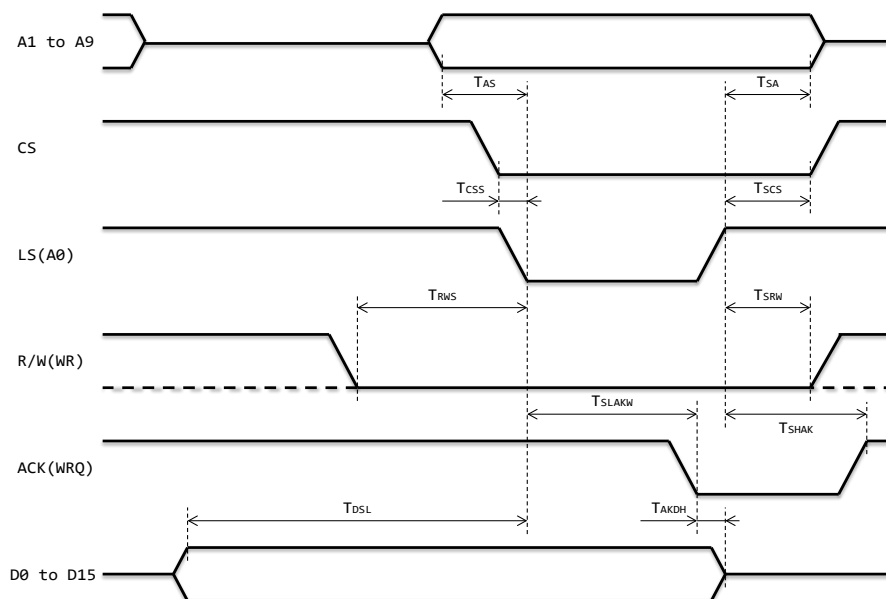


Item	Symbol	Conditions	Min.	Max.	Unit
Frequency	f_{CLK}	-	-	31.25	MHz
Cycle	T_{CLK}	-	32	-	ns
H level width	T_{CKH}	-	13	-	ns
L level width	T_{CKL}	-	13	-	ns

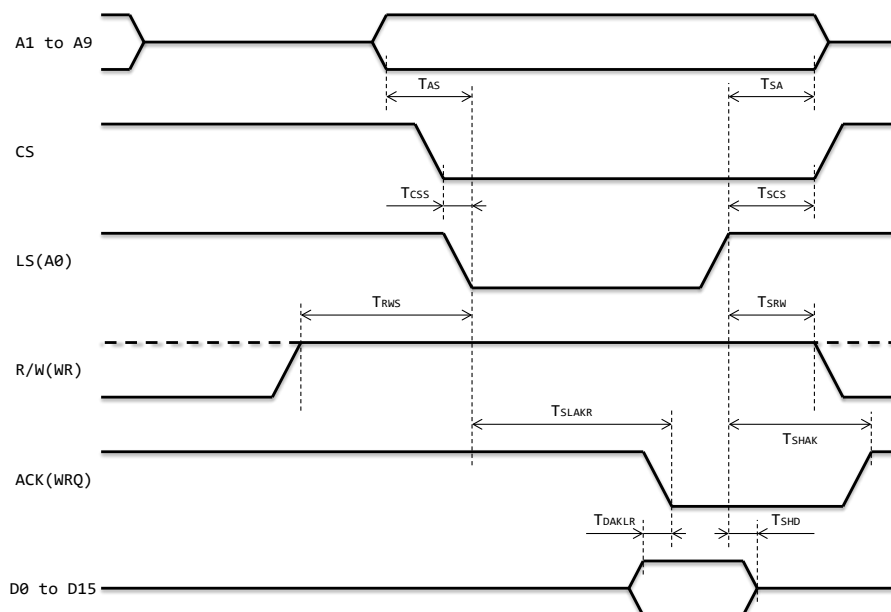
7.4.2 CPU IF = 0 (68000)

If IF1 = L level and IF0 = L level are set, the interface will be for 68000 series CPUs.

<Write cycle>



<Read cycle>

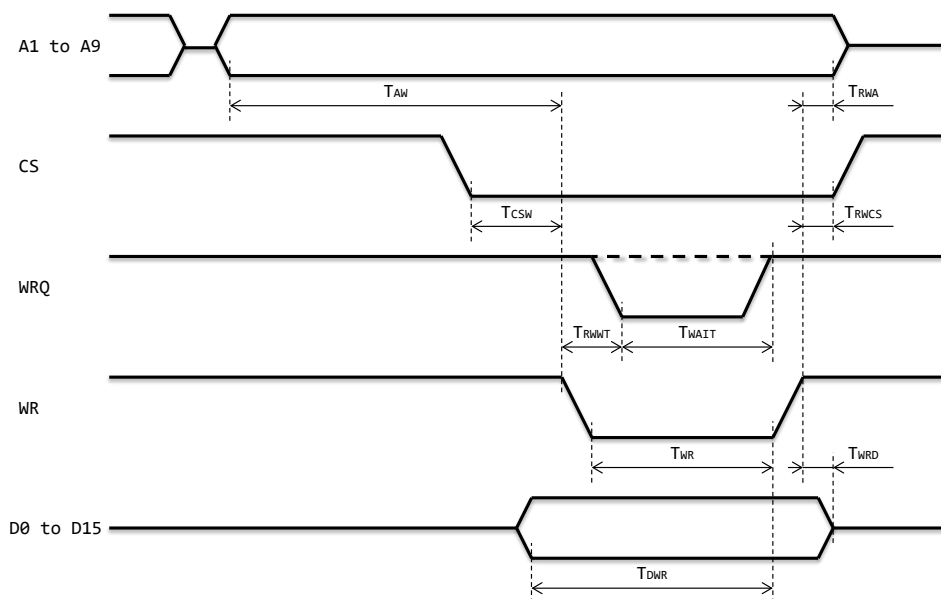


Item	Symbol	Conditions	Min.	Max.	Unit
Address setup time	LS↓ T _{AS}	-	3	-	ns
Address hold time	LS↑ T _{SA}	-	3	-	ns
CS setup time	LS↓ T _{CSS}	-	0	-	ns
CS hold time	LS↑ T _{SCS}	-	3	-	ns
R/W setup time	LS↓ T _{RWS}	-	6	-	ns
R/W hold time	LS↑ T _{SRW}	-	3	-	ns
ACK ON delay time	LS↓ T _{SLAKR}	C _L =40pF	1 · T _{CLK}	7 · T _{CLK}	ns
	LS↓ T _{SLAKW}	C _L =40pF	1 · T _{CLK}	6 · T _{CLK}	ns
ACK OFF delay time	LS↑ T _{SHAK}	C _L =40pF	5	24	ns
Data output lead time	ACK↓ T _{DAKLR}	C _L =40pF	T _{CLK} -3	-	ns
Data float delay time	LS↑ T _{SHD}	C _L =40pF	-	31	ns
Data setup time	LS↑ T _{DSL}	-	12	-	ns
Data hold time	ACK↓ T _{AKDH}	-	0	-	ns

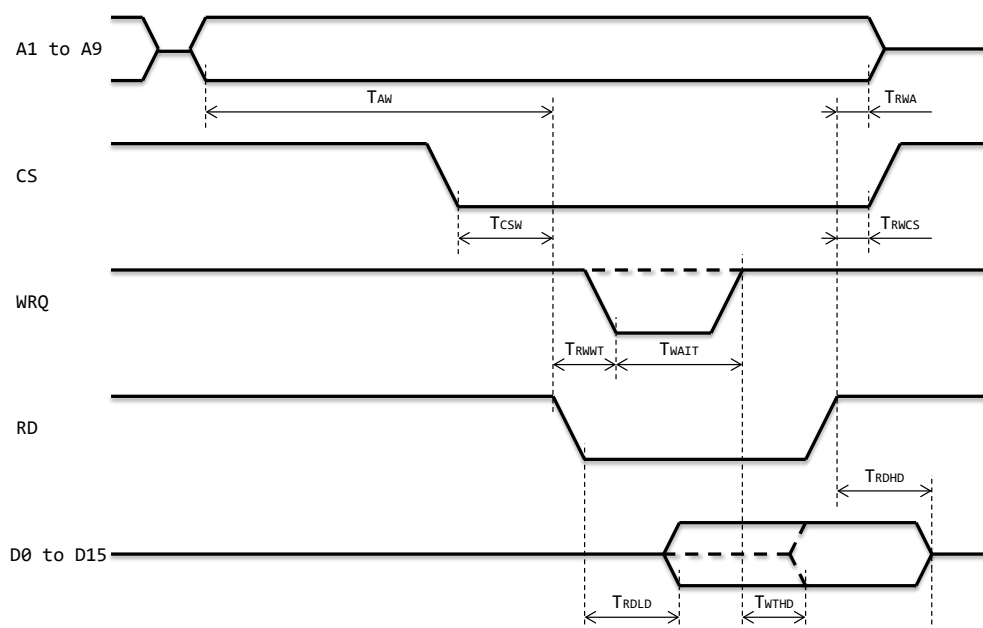
7.4.3 CPU IF = 1 (H8)

If IF1 = L level and IF0 = H level are set, the interface will be for H8 CPUs.

<Write cycle>



<Read cycle>



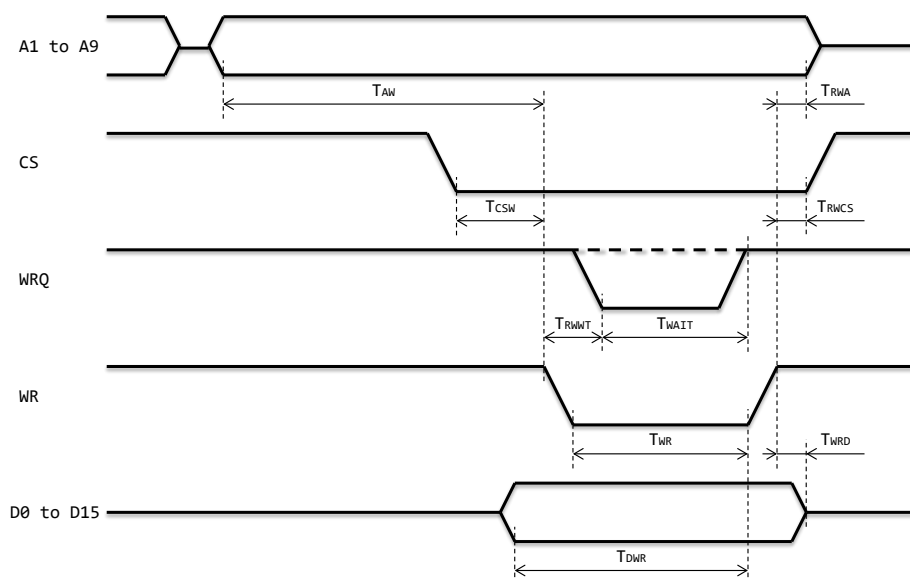
Item	Symbol	Conditions	Min.	Max.	Unit
Address setup time	RD↓	T _{AR}	11	-	ns
Address setup time	WR↓	T _{AW}	10	-	ns
Address hold time	RD↑, WR↑	T _{RWA}	0	-	ns
CS setup time	RD↓	T _{CSR}	3	-	ns
CS setup time	WR↓	T _{CSW}	3	-	ns
CS hold time	RD↑, WR↑	T _{RWCS}	0	-	ns
WRQ ON time	RD↓, WR↓	T _{RWWT}	-	15	ns
WRQ L level time		T _{WAIT}	-	4 · T _{CLK}	ns
Data output delay time	RD↓	T _{RDLD}	-	27	ns
Data output delay time	WRQ↑	T _{WTHD}	-	13	ns
Data float delay time	RD↑	T _{RDHD}	-	23	ns
WR signal width		T _{WR}	*1	-	ns
Data setup time	WR↑	T _{DWR}	8	-	ns
Data hold time	WR↑	T _{WRD}	0	-	ns

*1 When WRQ signal is output, the time from WRQ = H level to WR = H level is reached.

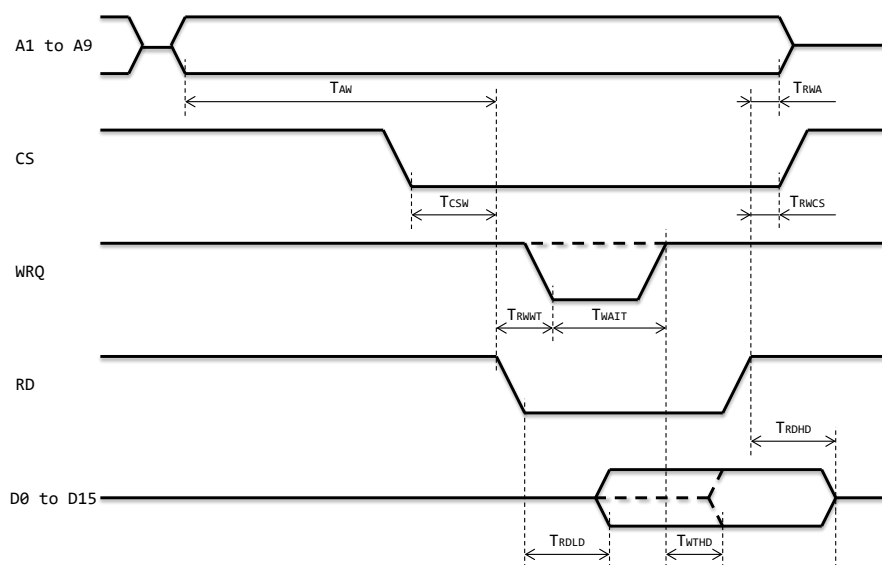
7.4.4 CPU IF = 2 (8086)

If IF1 = H level and IF0 = L level are set, the interface will be for 8086 CPUs.

<Write cycle>



<Read cycle>



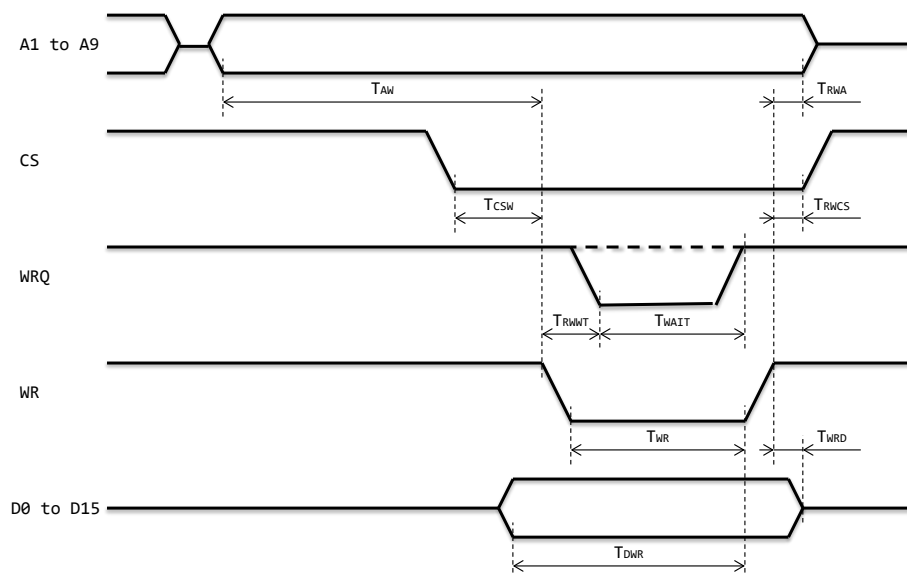
Item	Symbol	Conditions	Min.	Max.	Unit
Address setup time	RD↓ TAR	-	11	-	ns
Address setup time	WR↓ TAW	-	10	-	ns
Address hold time	RD↑, WR↑ TRWA	-	0	-	ns
CS setup time	RD↓ TCSR	-	3	-	ns
CS setup time	WR↓ TCSW	-	3	-	ns
CS hold time	RD↑, WR↑ TRWCS	-	0	-	ns
WRQ ON time	RD↓, WR↓ TRRWT	CL=40pF	-	15	ns
WRQ L level time	TWAIT	-	-	4 · TCLK	ns
Data output delay time	RD↓ TRDLD	CL=40pF	-	27	ns
Data output delay time	WRQ↑ TWTHD	CL=40pF	-	13	ns
Data float delay time	RD↑ TRDHD	CL=40pF	-	23	ns
WR signal width	TWR	* 1	6	-	ns
Data setup time	WR↑ TDWR	-	8	-	ns
Data hold time	WR↑ TWRD	-	0	-	ns

* 1 When the WRQ signal is output, the time from WRQ = H level to WR = H level is reached.

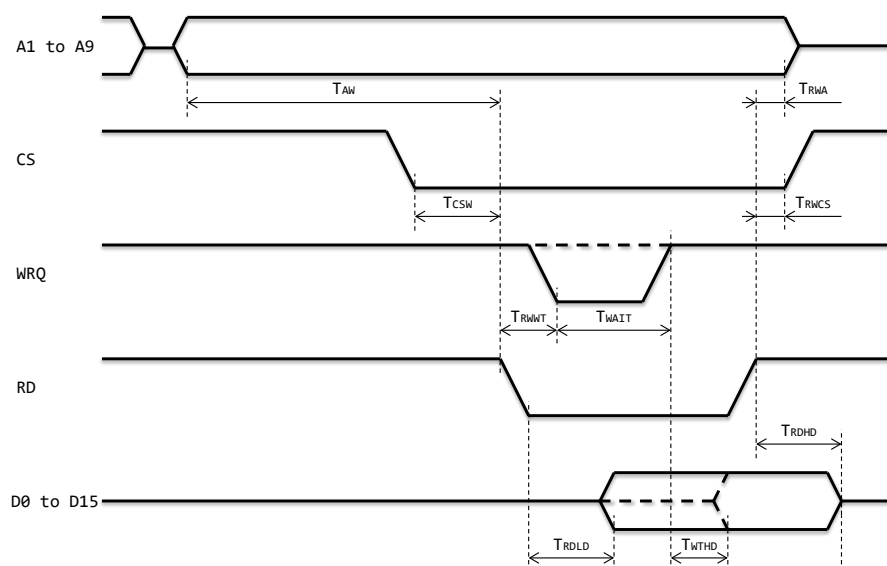
7.4.5 CPU IF = 3 (Z80)

If IF1 = H level and IF0 = H level are set, the interface will be for Z80 CPUs.

<Write cycle>



<Read cycle>



Item	Symbol	Conditions	Min.	Max.	Unit
Address setup time	RD↓	T _{AR}	11	-	ns
Address setup time	WR↓	T _{AW}	10	-	ns
Address hold time	RD↑, WR↑	T _{RWA}	0	-	ns
CS setup time	RD↓	T _{CSW}	3	-	ns
CS setup time	WR↓	T _{CSW}	3	-	ns
CS hold time	RD↑, WR↑	T _{RWCS}	0	-	ns
WRQ ON time	RD↓, WR↓	T _{RWWT}	-	15	ns
WRQ L level time		T _{WAIT}	-	4 · T _{CLK}	ns
Data output delay time	RD↓	T _{RDLD}	-	27	ns
Data output delay time	WRQ↑	T _{WTHD}	-	13	ns
Data float delay time	RD↑	T _{RDHD}	-	23	ns
WR signal width		T _{WR}	*1	6	ns
Data setup time	WR↑	T _{DWR}	-	8	ns
Data hold time	WR↑	T _{DWRD}	-	0	ns

*1 When WRQ signal is output, the time from WRQ = H level to WR = H level is reached.

7.5 Operation timing

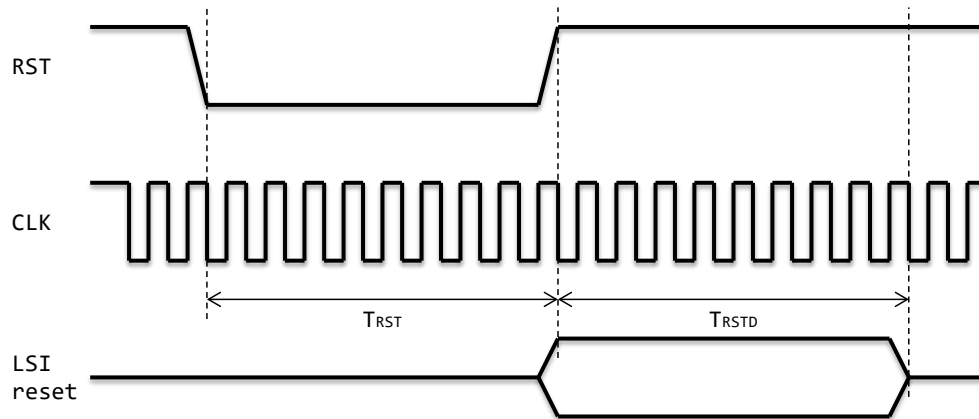
Input signals completely ignore the below the Minimum time and reacts reliably above the Standard time.

Output signals reliably output more than the Minimum time and stops completely within the Standard time.

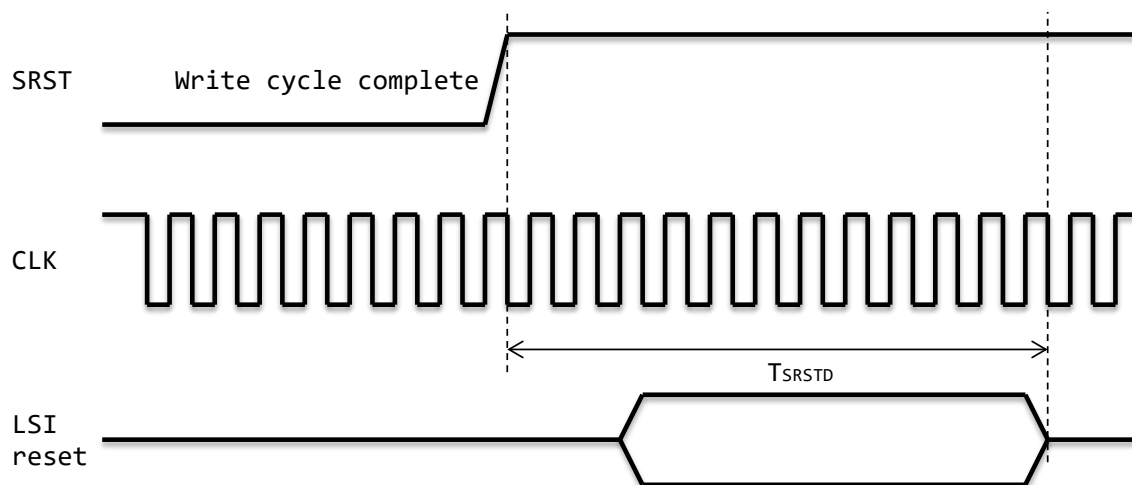
The delay time is not completed if it is less than the Minimum time, and it is completed if more than the Standard time.

Item	Symbol	Condition	Minimum	Standard	Unit
RST input signal	T_{RST}	-	7	8	$\times T_{CLK}$ ns
RST delay time	T_{RSTD}	-	7	8	$\times T_{CLK}$ ns
SRST delay time	T_{SRSTD}	-	11	12	$\times T_{CLK}$ ns
CLR input signal	-	-	-	1	$\times T_{CLK}$ ns
EA, EB, EZ input signals	T_{EAB}	RENV2.EINF=0	-	1	$\times T_{CLK}$ ns
		RENV2.EINF=1	2	3	$\times T_{CLK}$ ns
PA, PB input signals	T_{PAB}	RENV2.PINF=0	-	1	$\times T_{CLK}$ ns
		RENV2.PINF=1	2	3	$\times T_{CLK}$ ns
+EL, -EL, +SD, -SD, ORG, ALM, INP, CEMG input signals	-	RENV1.FLTR=0	-	1	$\times T_{CLK}$ ns
		RENV1.FLTR=1	64	80	$\times T_{CLK}$ ns
ERC output signal ON width	-	RENV1.EPW=000b	224	240	$\times T_{CLK}$ ns
		RENV1.EPW=001b	1792	1920	$\times T_{CLK}$ ns
		RENV1.EPW=010b	7168	7680	$\times T_{CLK}$ ns
		RENV1.EPW=011b	28672	30720	$\times T_{CLK}$ ns
		RENV1.EPW=100b	229376	245760	$\times T_{CLK}$ ns
		RENV1.EPW=101b	917504	983040	$\times T_{CLK}$ ns
		RENV1.EPW=110b	1835008	1966080	$\times T_{CLK}$ ns
ERC output signal OFF width	-	RENV1.ETW=01b	224	240	$\times T_{CLK}$ ns
		RENV1.ETW=10b	28672	30720	$\times T_{CLK}$ ns
		RENV1.ETW=11b	1835008	1966080	$\times T_{CLK}$ ns
General-purpose one-shot output signal	-	-	458738	491505	$\times T_{CLK}$ ns
+DR, -DR, PE input signals	-	RENV1.DRF=0	-	1	$\times T_{CLK}$ ns
		RENV1.DRF=1	524288	655360	$\times T_{CLK}$ ns
PCS input signal	-	-	-	1	$\times T_{CLK}$ ns
LTC input signal	-	-	-	1	$\times T_{CLK}$ ns
CSTA, STA input signals	-	-	4	5	$\times T_{CLK}$ ns
CSTA output signal	-	-	8		$\times T_{CLK}$ ns
CSTP input signal	-	-	4	5	$\times T_{CLK}$ ns
CSTP output signal	-	-	8		$\times T_{CLK}$ ns
BSY signal ON delay time	T_{CMDBSY}	-	5	6	$\times T_{CLK}$ ns
	T_{STABSY}	-	8	9	$\times T_{CLK}$ ns
Start delay time	T_{CMDPLS}	-	16	17	$\times T_{CLK}$ ns
	T_{STAPLS}	-	19	20	$\times T_{CLK}$ ns

7.5.1 RST signal

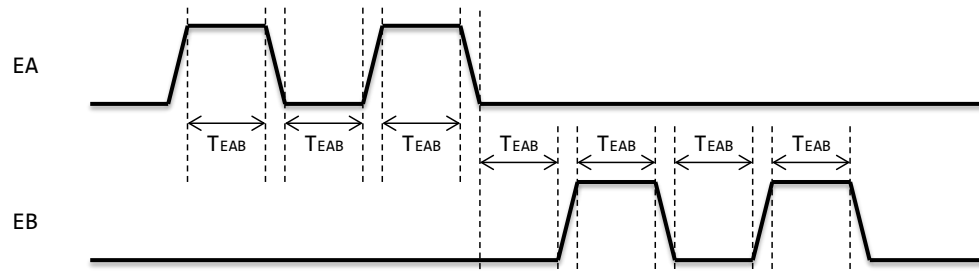


7.5.2 SRST command

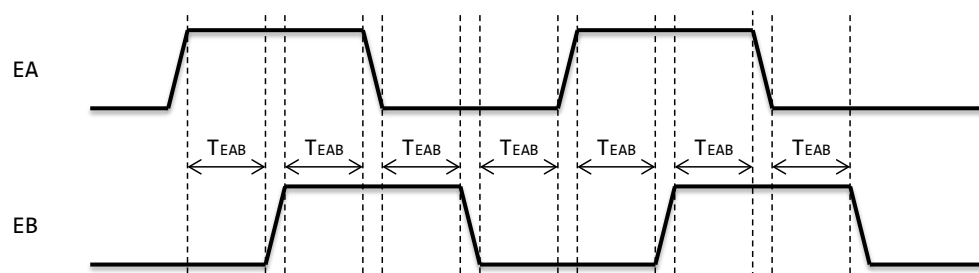


7.5.3 EA, EB signals

7.5.3.1 2-pulse mode (encoder)

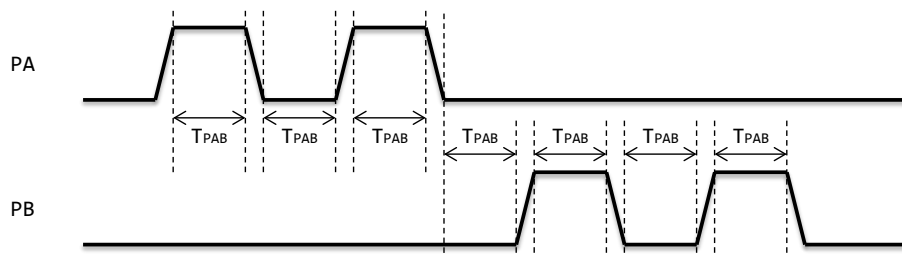


7.5.3.2 90-degree phase difference mode (encoder)

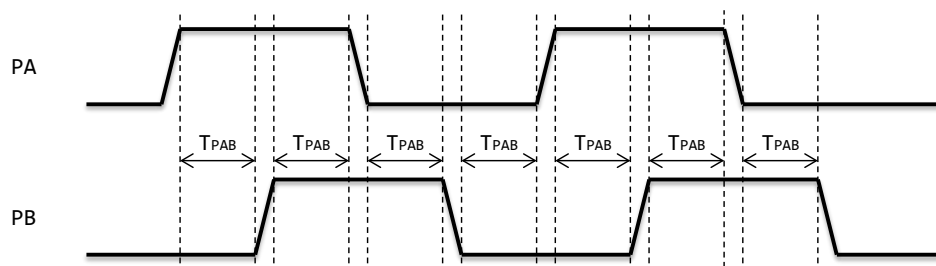


7.5.4 PA, PB signals

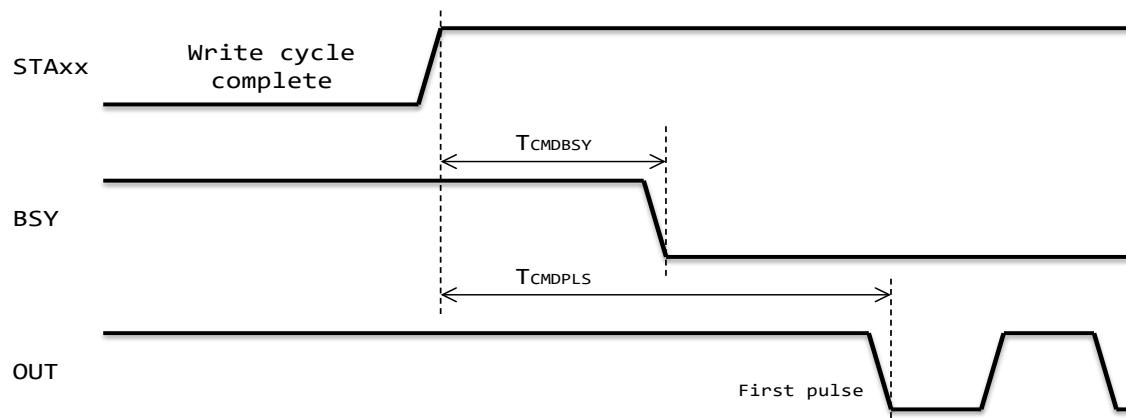
7.5.4.1 2-pulse mode (manual pulser)



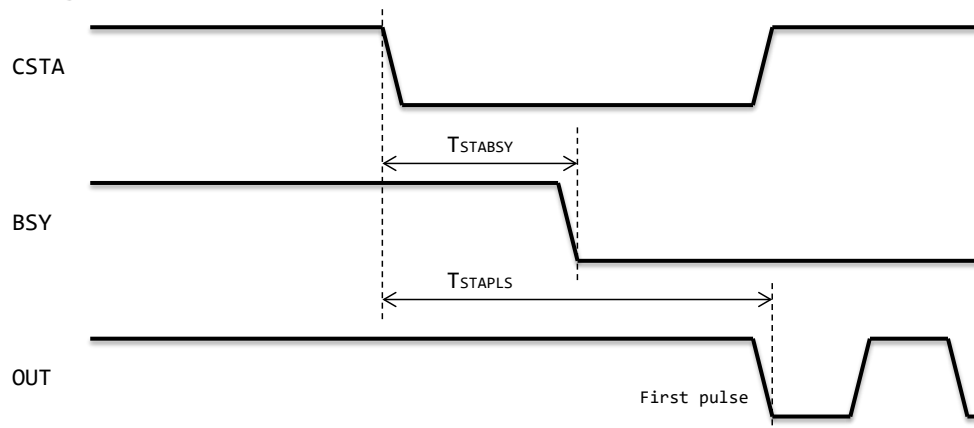
7.5.4.2 90-degree phase difference mode (manual pulser)



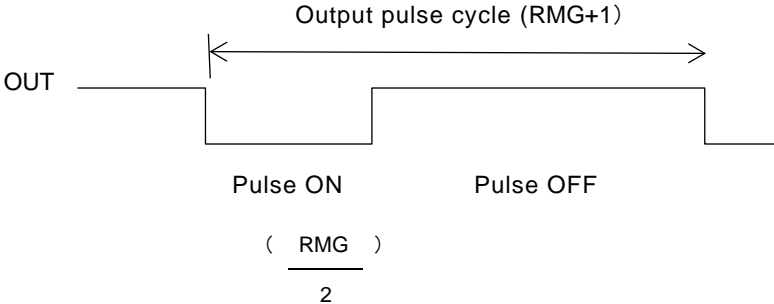
7.5.5 Start command



7.5.6 CSTA signal




Revision history

Revision	Date	Content
1st	Aug 3, 2009	- New document
2nd	Apr 16, 2018	<p>- P38: The following comments are added in "8-2-3.Writing to the comparator pre-registers": "However, when the comparison status between RCMP5 and the comparison target is "true", you must be careful when writing data to PRCP5. When the comparison status becomes "false" by writing data to PRCP5, the shift condition is satisfied and the written data may be deleted due to the shift".</p> <p>- P46: Description in FLTR is revised as follows: Incorrect: Apply filters to the +EL, -EL, SD, ORG, ALM, and INP inputs. Correct: Apply filters to the +EL, -EL, SD, ORG, ALM, INP and CEMG inputs.</p> <p>-P46: "Note 2" is added in "RENV1.PDTC".</p> <p>-P46: The following sentences are added in "Note 2"; "When the value of magnification setting register (RMG) is an even number, an error occurs in the duty ratio. For details, see "11-3-2. Control the output pulse width and operation complete timing".</p> <p>- P111: The following sentence and chart are added in "11-3-2. Control the output pulse width and operation complete timing": [Addition] "However, if RMG register has an even number, an error occurs in the duty ratio. Consequently pulse ON time becomes shorter than pulse OFF time. For example, when RMG register has 2, the output pulse cycle is 3 and the pulse ON time becomes 1. Consequently the duty ratio becomes 1: 2.</p> 
2nd	Apr 16, 2018	<p>- P117: Revision in "Apply an input filter to EZ". Incorrect: <Set FLTR (bit 26) in RENV2> Correct: <Set FLTR (bit 18) in RENV2></p>

Revision	Date	Content																
		<p>- P117: Revision in "Apply an input filter to EZ".</p> <p>Incorrect: By applying a filter, signals with a pulse width of 4 μsec or less will be ignored.</p> <p>Correct: By applying a filter, input pulse (width) less than 3 cycles of CLK signal will be disabled.</p> <p>- P117: Revision in "Apply an input filter to EZ".</p> <p style="text-align: center;">In correct</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">31</div> <div style="text-align: center;"> <p>[RENV2] (WRITE)</p> <p>24</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> <td style="width: 15px;">n</td> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> </tr> </table> </div> </div> <hr style="width: 30%; margin: 10px auto;"/> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">23</div> <div style="text-align: center;"> <p>Correct</p> <p>[RENV2] (WRITE)</p> <p>16</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> <td style="width: 15px;">n</td> <td style="width: 15px;">-</td> <td style="width: 15px;">-</td> </tr> </table> </div> </div> <p>- P134: 4 new charts are added in "[Speed change using the comparator]". One whole page increases by this addition.</p>	-	-	-	-	-	n	-	-	-	-	-	-	-	n	-	-
-	-	-	-	-	n	-	-											
-	-	-	-	-	n	-	-											
2nd	Apr 16, 2018	<p>-P30: The following phrase is added in "Direct access method"; "In read cycle of the lowest address of each register (lower side data in the case of 8086 mode and Z80 mode, the uppermost side data in the case of 68000 mode and H8 mode).</p> <p>-P30: The following sentences are added in "Direct access method"; "Furthermore, after writing access, wait 3 cycles of reference clock of PCL6046 with soft timer etc. (There is a problem in the circuit that controls # WRQ, if you may not be able to write normally unless you use soft timer. For the details of the problem, refer to the product non-conformity information "DB70241-0.")</p>																
2nd	Apr 16, 2018	<p>-P95: Following sentences are added in "- Acceleration/deceleration operations":</p> <p>"If an axis decelerates and stops during linear interpolation or circular interpolation with acceleration/deceleration the following phenomenon can happen; some axes immediately stop without deceleration, or all interpolation axes immediately stop without indicating axes are being stopped.</p> <p>-The factors of deceleration stop are the followings:</p> <ol style="list-style-type: none"> 1) ALM signal input 2) Software limit 3) Comparator 1 to 5 <p>For this reason, set the stop method by 1) to 3) to "immediately stop". Even if the stop method</p>																

Revision	Date	Content
		is "deceleration stop", there is no problem when you use constant speed start.
2nd	Apr 16, 2018	-Revise word "Pulsar" to "Pulser"
2nd	Apr 17, 2018	-Changed "emergency stop" to "error stop". -Changed "µsec" to "µs". -Removed "Label list" in Appendix 3.
3rd	July 2, 2018	-P14: <u>6-3. CPU I/F circuit block diagram</u> 1) Z80 I/F (memory map, full-address) Changed the location of "Pull-up" in the diagram. -P15: 3) 8086 I/F (Memory map, full address) Changed "Decode circuit" to "Interrupt control circuit". -P15: 4) 8086 I/F (I/O map, reduced address) Added CPU brand (8086CPU) and PCL model (PCL6046) in the diagram. -P16: 5) H8 I/F (full address) Removed the arrow(→) beside the "Decode circuit". -P16: 6) H8 I/F (reduced address) Removed the arrow(→) beside the "Decode circuit". -P16: Added "Note" under the "6) H8 I/F (reduced address)" diagram. -P17: Corrected the connecting location of "#System reset" in 7) 68000 I/F (full address) diagram. -P17: 8) 68000 I/F (reduced address) . "Diagram" and "Note" are revised. <u>- 6-4-2. Internal map of each axis</u> <I/F for Z80 (Direct access)> chart is revised: -P19: "Address signal" of "07 to 04" is revised from "BUFB3~" to "BUFB3 to BUFB0". -P20: Added the address below "93 to 90". -P20: Added I/F for Z80 (Indirect access) chart. -P23: The title of the chart is changed from <When used with the H8 and 8068 I/F(Direct access)> to <I/F for H8 or 68000 (Direct access)>.
3rd	July 3, 2018	-P28: 7-1-1. Write an operation command (Axis assignment omitted) Added "Command" to D15 to D0. -P39: 8-2. Pre-registers Changed from "5" to "2nd Pre-register" in "PRMV etc." box. -P39: 8-2-1. Writing to the operation pre-registers. Revised the chart by changing "Set" to "determined".

Revision	Date	Content
		<p>-P40:8-2-3. Writing to the comparator pre-registers</p> <p>Revised the chart by changing "Set" to "determined".</p>
3rd	July 3, 2018	<p>P43: 8-3-8. PRMD (RMD) register</p> <p>Revised bit 28 from "0" to "MIPM".</p> <p>P80: 9-5-1-1. Origin return operation 0 (ORM = 0000) Revised "High speed operation <Sensor: EL (ELM = 1), SD (SDM = 0, SDLT = 0), ORG>" chart.</p> <p>P82: 9-5-1-5. Origin return operation 4 (ORM = 0100) Revised "Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>" chart.</p> <p>P83: 9-5-1-6. Origin return operation 5 (ORM = 0101) Revised "Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>" chart.</p> <p>P85: 9-5-1-11. Origin return operation 10 (RENV3.ORM = 1010) Revised "High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>" chart.</p> <p>P85: 9-5-1-12. Origin return operation 11 (RENV3.ORM = 1011) Revised "High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>" chart.</p> <p>P104: 10-3. Manual FH correction (i) Make a linear acceleration/deceleration range smaller: Revised the equation of "$PRMV \leq$".</p> <p>P105: Manual FH correction (3)-3 When $PRUS > PRDS$ (i) Make a linear acceleration/deceleration range smaller. Revised the equation of "$PRMV \leq$", "$PRMV >$". (ii) Eliminate the linear acceleration section and make a linear deceleration range smaller. Revised the equation of "$PRMV \leq$".</p> <p>P119: 11-6-1. INP signal Replace "deflection" by "deviation".</p> <p>P126: 11-10-1. Counter type and input method Revised the entry of "COUNTER 4 and 1/2 of reference clock from "Not possible" to "Possible".</p> <p>P130: 11-10-3. Latch the counter and count condition:</p>

Revision	Date	Content																								
		Revised "Set an event interrupt cause <Set RIRQ.IRLT (bit 14) and RIRQ.IROL (bit 15)>																								
3rd	July 3, 2018	<p>P131: 11-10-4. Stop the counter:</p> <p><u>Change the position of "n".</u></p> <table border="1" data-bbox="467 421 940 584"> <tr> <td colspan="2" style="text-align: left;">RMD]</td> <td colspan="6" style="text-align: right;">(WRITE)</td> </tr> <tr> <td style="text-align: center;">15</td> <td colspan="5"></td> <td style="text-align: center;">8</td> <td></td> </tr> <tr> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> <td style="text-align: center;">n</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> </tr> </table> <p>P133: <u>11-11-1. Comparator types and functions</u> "[Comparison method] "chart" Revised "C1RM" to "C2RM" under "Comparator 2"</p> <p>P134: "Specify the output timing for an internal synchronous signal": Revised from "Set RENV5.SYO1 to 3 (bits 16 to 19)" to "Set RENV5.SYO0 to 3 (bits 16 to 19)".</p> <p>P143: <u>11-14. Synchronous starting</u> "Specify the internal synchronous signal output timing" Revised from "<Set RENV5.SYO1 to 3 (bits 16 to 19)>" to "<Set RENV5.SYO0 to 3 (bits 16 to 19)>"</p> <p>P156: <u>12-5-2. CPU-I/F 2) (IF1 = H, IF0 = L) 8086</u> Revised the chart by changing the entry between "Address setup time for #WR" and "Min." from "11" to "10".</p> <p>P157: Revised <u>chapter</u> name: <u>CPU-I/F 3) (IF1 = L, IF0 = L) H8</u>  <u>CPU-I/F 3) (IF1 = L, IF0 = H) H8</u></p> <p>P157: <Write cycle> Added \longleftrightarrow under "TDWR".</p> <p>P160: Revised the title to "5) Timing for the command start (IF1=H and IF0=H)" from "5) Timing for the command start (IF1=H and IF0=H)".</p>	RMD]		(WRITE)						15						8		-	-	-	-	n	-	-	-
RMD]		(WRITE)																								
15						8																				
-	-	-	-	n	-	-	-																			
4th	June 21, 2022	TA600091-EN0/0 Corrected errors and reviewed completely in DA70122-1/2E																								

	<p>1.2 Features</p> <p>-Servomotor I/F</p> <p>The ERC signal is a pulsed output. The pulse length can be set.</p> <p>Incorrect: (12 μs to 104 ms, A level-output is also available.)</p> <p>Correct: (11 μs to 100 ms, level-output is also available.)</p> <p>∴ Described in "2.1 Features" in this manual.</p> <p>4. Functions of Terminals</p> <p>"Input/output" column of "IF0,IF1".</p> <p>Incorrect: Input</p> <p>Correct: Input U</p> <p>∴ Described in "7.3 DC characteristics" in this manual.</p> <p>6-3. CPU I/F circuit block diagram</p> <p>7) 68000 I/F (full address).</p> <p>Incorrect: From #LDS and GND to A0. From N.C. to #RD.</p> <p>Correct: From only #LDS to A0. From VDD to #RD.</p> <p>∴ Described in "4.4.1.1 68000 interface example (Full address)" in this manual.</p> <p>8) 68000 I/F (reduced address).</p> <p>Incorrect: From A15-A5 to decode circuit. From A0 to #LDS. From GND to A0. From N.C to #RD.</p> <p>Correct: From A23-A5 to decode circuit. From #LDS to A0. From VDD to #RD.</p> <p>∴ Described in "4.4.1.2 68000 interface example (Reduced address)" in this manual.</p> <p>7-1-4. Stop command</p> <p>3) Emergency stop command</p> <p>Incorrect: Emergency stop (same as a #CEMG signal input)</p> <p>Correct: Stop emergently and cancel the operation mode.</p> <p>∴ Described in "5.3.1.7 Emergency stop command" in this manual.</p> <p>※ CEMG signal stop for all axes in operation emergently.</p> <p>CMEMG(05h) command can stop the written axis emergently.</p> <p>7-2. General-purpose output bit control commands</p> <p>Incorrect: The P0 and P1 terminals can be set for one shot output (T = approx. 26 ms.) using the RENV2 (Environment setting 2) register, and the output logic can be selected.</p> <p>Correct: The P0 and P1 terminals can be set for one shot output (T = 23 to 25 ms) using the RENV2 (Environment setting 2) register, and the output logic can be selected.</p> <p>∴ Described in "5.3.2.2 General-purpose output bit control command" in this manual.</p> <p>8-3-8. PRMD(RMD) register</p>
--	--

Revision	Date	Content																																																	
		<p>Incorrect:</p> <table border="1"> <tr> <td>31</td> <td>30</td> <td>29</td> <td>28</td> </tr> <tr> <td>0</td> <td>0</td> <td>MSDC</td> <td>MIPM</td> </tr> </table> <table border="1"> <tr> <td>29</td> <td>MSDC</td> <td>0: Uses count method only when interpolation operation is performed with constant synthesized speed control like PCL6045B. Otherwise, calculation method is used. 1: Fix the method to set ramp-down point automatically, to "count method".</td> </tr> <tr> <td>31 to 30</td> <td>Not defined</td> <td>(Always set 0.)</td> </tr> </table> <p>Correct:</p> <table border="1"> <tr> <td>31</td> <td>30</td> <td>29</td> <td>28</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>MIPM</td> </tr> </table> <table border="1"> <tr> <td>31:29</td> <td>Not defined</td> <td>(Always set it to 0.)</td> </tr> </table> <p>∴ Described in "5.4.3.1 RMD(PRMD): Operation mode" in this manual. ※ RMD. MSDC = 1 does not function properly.</p> <p>8-3-13. RENV1 register</p> <p>"Description" column in "10 EROE".</p> <p>Correct: ERC signal can be output even when CEMG (05h) command is used to stop immediately.</p> <p>∴ Described in "5.4.3.2 RENV1: Environment setting 1" in this manual.</p> <p>"Description" column of "14:12 EPW".</p> <p>Incorrect:</p> <table border="1"> <tr> <td>000b: 12 μs</td> <td>001b: 102 μs</td> <td>010b: 409 μs</td> <td>011b: 1.6 ms</td> </tr> <tr> <td>100b: 13 ms</td> <td>101b: 52 ms</td> <td>110b: 104 ms</td> <td>111b: Level output</td> </tr> </table> <p>Correct:</p> <table border="1"> <tr> <td>000b: 11 to 13 μs</td> <td>001b: 91 to 98 μs</td> <td>010b: 364 to 391 μs</td> <td>011b: 1.4 to 1.6 ms</td> </tr> <tr> <td>100b: 11 to 13 ms</td> <td>101b: 46 to 50 ms</td> <td>110b: 93 to 100 ms</td> <td>111b: Level output</td> </tr> </table> <p>∴ Described in "5.4.3.2 RENV1: Environment setting 1" in this manual.</p> <p>The description of the EPW bit is the same as that of the rest.</p> <p>"Description" column of "17,16 ETW".</p> <p>Incorrect:</p> <table border="1"> <tr> <td>00b: 0 μs</td> <td>01b: 12 μs</td> <td>10b: 1.6 ms</td> <td>011b: 104 ms</td> </tr> </table> <p>Correct:</p> <table border="1"> <tr> <td>00b: 0 μs</td> <td>01b: 11 to 13 μs</td> <td>10b: 1.4 to 1.6 ms</td> <td>011b: 93 to 100 ms</td> </tr> </table> <p>∴ Described in "5.4.3.2 RENV1: Environment setting 1" in this manual.</p> <p>The description of the ETW bit is the same as that of the rest.</p>	31	30	29	28	0	0	MSDC	MIPM	29	MSDC	0: Uses count method only when interpolation operation is performed with constant synthesized speed control like PCL6045B. Otherwise, calculation method is used. 1: Fix the method to set ramp-down point automatically, to "count method".	31 to 30	Not defined	(Always set 0.)	31	30	29	28	0	0	0	MIPM	31:29	Not defined	(Always set it to 0.)	000b: 12 μs	001b: 102 μs	010b: 409 μs	011b: 1.6 ms	100b: 13 ms	101b: 52 ms	110b: 104 ms	111b: Level output	000b: 11 to 13 μs	001b: 91 to 98 μs	010b: 364 to 391 μs	011b: 1.4 to 1.6 ms	100b: 11 to 13 ms	101b: 46 to 50 ms	110b: 93 to 100 ms	111b: Level output	00b: 0 μs	01b: 12 μs	10b: 1.6 ms	011b: 104 ms	00b: 0 μs	01b: 11 to 13 μs	10b: 1.4 to 1.6 ms	011b: 93 to 100 ms
31	30	29	28																																																
0	0	MSDC	MIPM																																																
29	MSDC	0: Uses count method only when interpolation operation is performed with constant synthesized speed control like PCL6045B. Otherwise, calculation method is used. 1: Fix the method to set ramp-down point automatically, to "count method".																																																	
31 to 30	Not defined	(Always set 0.)																																																	
31	30	29	28																																																
0	0	0	MIPM																																																
31:29	Not defined	(Always set it to 0.)																																																	
000b: 12 μs	001b: 102 μs	010b: 409 μs	011b: 1.6 ms																																																
100b: 13 ms	101b: 52 ms	110b: 104 ms	111b: Level output																																																
000b: 11 to 13 μs	001b: 91 to 98 μs	010b: 364 to 391 μs	011b: 1.4 to 1.6 ms																																																
100b: 11 to 13 ms	101b: 46 to 50 ms	110b: 93 to 100 ms	111b: Level output																																																
00b: 0 μs	01b: 12 μs	10b: 1.6 ms	011b: 104 ms																																																
00b: 0 μs	01b: 11 to 13 μs	10b: 1.4 to 1.6 ms	011b: 93 to 100 ms																																																

Revision	Date	Content
		<p>"Description" column in "26 FLTR".</p> <p>Incorrect: 1: Apply a filter to the +EL, -EL, SD, ORG, ALM, INP and CEMG inputs. When a filter is applied, signal pulses shorter than 4 μsec are ignored.</p> <p>Correct: 0: Recognizes the pulse signals of width of 0.05 μs or more.</p> <p>1: Ignores the pulse signals of width of 3 μs or less completely.</p> <p>∴ Described in "5.4.3.2 RENV1: Environment setting 1" in this manual.</p> <p>The descriptions of FLTR bits are all the same.</p> <p>"Description" column of "27 DRF".</p> <p>Incorrect: 1: Apply a filter to the +DR, -DR, or PE inputs. When a filter is applied, signals pulses shorter than 32 ms are ignored.</p> <p>Correct: 0: Recognizes the pulse signals of width of 0.05 μs or more.</p> <p>1: Ignores the pulse signals of width of 26 ms or less.</p> <p>∴ Described in "5.4.3.2 RENV1: Environment setting 1" in this manual.</p> <p>The description of the DRF bit is the same as that of the rest.</p> <p>"Description" column in "30 PCSM".</p> <p>Correct: 1: Make PCS input as a #CSTA signal for only the own axis. #CSTA pin input is disabled.</p> <p>∴ Described in "5.4.3.2 RENV1: Environment setting 1" in this manual.</p> <p>The descriptions of PCSM bits are all the same.</p> <p>8-3-16. RENV4 Register</p> <p>"Description" column of "23 IDXN".</p> <p>Incorrect: 0: while COUNTER4 = RCMP2.</p> <p>Correct: 0: while COUNTER4 = RCMP4</p> <p>∴ Described in "5.4.3.5 RENV4: Environment setting 4" in this manual.</p> <p>8-3-18. RENV6 register</p> <p>"Description" column of "15 PSTP".</p> <p>Incorrect: 1: Even if the stop command is written, the PCL will operate for the number of pulses that are already input on PA/PB. Note 1</p> <p>Note 1. When PSTP is 1, the Stop command will be ignored when #BSYn = H (OFF), regardless of the operation mode. Before writing a Stop command, check the main status register. When SRUN = 0, change PSTP to 0 and then write a Stop command.</p> <p>Correct: Sets processing when writing stop command in pulsar control.</p> <p>0: Ignores the input PA, PB and stops.</p> <p>1: The command pulse corresponding to the input PA, PB signal is output before stopping. For interpolating control (68h, 69h, 6Ah, 6Bh, 6Ch, 6Dh), RENV6.PSTP=1 is ignored and stopped.</p>

Revision	Date	Content
		<p>∴ Described in "5.4.3.7 RENV6: Environment setting 6" in this manual.</p> <p>9-3. Pulser (PA/PB) input mode</p> <p><Examples of the relationship between the FH (FL) speed [pps] and the pulser input frequency FP [pps]></p> <p>"Usage range" column of "90° phase difference 2x 2 (3x) 0 ".</p> <p>Incorrect: $FP < FH (FL) / 6$</p> <p>Correct: $FP < FH (FL) / 12$</p> <p>∴ Described in "5.5.3 Pulser control" in this manual.</p> <p>9-3-1. Continuous operation using a pulser input. (PRMD.MOD: 01h)</p> <p>In the text "Note".</p> <p>Incorrect: When the "immediate stop command (49h)" is written while the PCL is performing a multiplication operation (caused by setting RENV2.PIM 0 to 1 and RENV6.PMG 0 to 4), the PCL will stop operation immediately and the total number of pulses that are output will not always be an integral multiple of the magnification. When RENV6.PSTP is set to 1, the PCL delays the stop timing until an integral multiple of pulses has been output. However, after a stop command is sent by setting PSTP to 1, check the MSTSW. If SRUN is 0, set PSTP to 0. (When SRUN is 0 while RENV6.PSTP is 1, the PCL will latch the stop command.)</p> <p>Correct: When stopping immediately with STOP (0049h) command, the total output pulse is not always the integral multiple of a multiplication value. Set RENV6.PSTP = 1 if you delay an operation stop until the total output pulse becomes an integral multiple of a multiplication value. If you stop the operation before the total output pulse becomes an integral multiple of a multiplication value, set RENV6.PSTP = 0. However, in the case of interpolation control (68h, 69h, 6Ah, 6Bh, 6Ch, 6Dh), RENV6.PSTP = 1 is ignored and the operation stops.</p> <p>∴ Described in "5.5.3 Pulser control" in this manual.</p> <p>9-4-2. Positioning operation using an external switch. (PRMD.MOD: 56h)</p> <p>Incorrect: By turning ON the EL signal corresponding to the feed direction, the axis will stop operation and issue an error interrupt (#INT output).</p> <p>Correct: When the count direction is + direction, operation stops at +EL signal ON.</p> <p>When the count direction is - direction, operation stops at - EL signal ON.</p> <p>When stopped by +EL signal ON and -EL signal ON, no error interrupt occurs.</p> <p>When stopping by +EL signal ON and -EL signal ON, the operation mode does not complete.</p> <p>∴ Described in "5.5.4 Switch control" in this manual.</p>

Revision	Date	Content
		<p>9-8-1. Interpolation operations</p> <p>Incorrect: 62h Continuous linear interpolation 2 for 1 to 4 axes 63h Linear interpolation 2 for 1 to 4 axes</p> <p>Correct: 62h Continuous movement of linear interpolation 2 control (1 axis or more) 63h Incremental movement of linear interpolation 2 control (1 axis or more)</p> <p>∴ Described in "5.5.8 Linear interpolation 2 control" in this manual.</p> <p>9-8-3. Synthesized speed constant control</p> <p>Correct: If you use the axis that does not perform circular interpolation and U-axis together, you can perform the constant synthesized speed control on circular interpolation axis even with U-axis synchronous control.</p> <p>∴ Described in "5.5.10 U-axis synchronous control" and "6.3.6 Constant synthesized speed control" in this manual.</p> <p>10-1. Speed patterns</p> <p>In the text of "High-speed operation 1) Positioning operation mode".</p> <p>* Incorrect: * When positioning with a high speed start command 1 (52h), the ramping-down point is fixed to the manual setting, regardless of the setting for MSDP (bit 13) in the PRMD.</p> <p>Correct: (2) RPLS < RSDC ⇒ Starts deceleration.</p> <p>∴ Described in "6.3.1 Speed patterns" in this manual.</p> <p>11-1. Reset</p> <p>Incorrect: To reset the LSI, hold the #RST terminal LOW while supplying at least 8 cycles of a reference clock signal.</p> <p>Correct: For RST signal, input an L level signal with 8 cycles or more of the CLK signal and an H level signal with 8 cycles or more of the CLK signal.</p> <p>∴ Described in "6.1.1 Hardware reset" in this manual.</p> <p>11-5-3. ORG, EZ signals</p> <p>Incorrect: The input logic of the ORG signal and EZ signal can be changed using the RENV1 register (environment setting 1).</p> <p>Correct: The input logic (RENV1.ORGL) of ORG signal can be selected. The input logic (RENV2.EZL) of EZ signal can be selected.</p> <p>∴ Described in "6.7.3 Origin (ORG), Encoder Z phase (EZ)" in this manual.</p>

Revision	Date	Content																																																						
		<p>11-8. External stop / simultaneous stop</p> <p>Incorrect: 2) To stop simultaneously using an external circuit, connect as follows.</p> <p>Correct: 2) To perform simultaneous stop from an external circuit, connect as shown in the figure below.</p> <p>∴ Described in "6.10 External stop / Simultaneous stop" in this manual.</p> <p>11-11-1. Comparator types and functions</p> <p>Incorrect: [Comparison method] C4S0 to 3 / Use COUNTER1 as a ring counter ○ "1010" C4S0 to 3 / Use COUNTER2 as a ring counter ○ "1010"</p> <p>Correct: [Comparison data] C4S0 to 3 / Use COUNTER1 as a ring counter C4S0 to 3 / Use COUNTER2 as a ring counter </p> <p>∴ Described in "6.13.1 Comparator types and functions" in this manual.</p> <p>11-14-1. Start triggered by another axis shopping</p> <p>Incorrect:</p> <table border="1"> <tr> <td>2</td> <td>PRMV</td> <td>10000</td> <td>5000</td> <td>X and Y axes perform a linear interpolation 1 with an end point (1000, 5000)</td> </tr> </table> <p>Correct:</p> <table border="1"> <tr> <td>2</td> <td>PRMV</td> <td>10000</td> <td>5000</td> <td>Incremental movement 10000,5000.</td> </tr> </table> <p>∴ Described in "6.17.1.2 Continuous interpolation without changing the interpolation axes" in this manual.</p> <p>12-3. DC Characteristics</p> <p>Incorrect:</p> <table border="1"> <tr> <td rowspan="2">LOW input current</td> <td rowspan="2">I_{IL}</td> <td>Terminals have internal pull-up resistors.</td> <td></td> <td>-82.2</td> <td>μA</td> </tr> <tr> <td>Other than those above.</td> <td></td> <td>-1</td> <td>μA</td> </tr> <tr> <td colspan="6" style="text-align: center;">Omitted</td> </tr> <tr> <td>HIGH output current</td> <td>I_{OH}</td> <td>V_{OH}= V_{DD}-0.4V</td> <td></td> <td>-6</td> <td>mA</td> </tr> </table> <p>Correct:</p> <table border="1"> <tr> <td rowspan="2">Low level input current</td> <td rowspan="2">I_{IL}</td> <td>Pins have internal pull-up resistors.</td> <td>-90</td> <td>-</td> <td>μA</td> </tr> <tr> <td>Other than those above.</td> <td>-1</td> <td>-</td> <td>μA</td> </tr> <tr> <td colspan="6" style="text-align: center;">Omitted</td> </tr> <tr> <td>High level output current</td> <td>I_{OH}</td> <td>V_{OH}= V_{DD}-0.4V</td> <td>-</td> <td>-6</td> <td>mA</td> </tr> </table> <p>∴ Described in "7.3 DC characteristics" in this manual.</p> <p>12-6. Operation timing (Common to all axes)</p> <p>Incorrect: #RST input signal width Note 1 10T_{CLK} ns</p> <p>Note 1. Longer than 10 cycles of CLK signal is necessary to be input while the #RST terminal is LOW.</p>	2	PRMV	10000	5000	X and Y axes perform a linear interpolation 1 with an end point (1000, 5000)	2	PRMV	10000	5000	Incremental movement 10000,5000.	LOW input current	I _{IL}	Terminals have internal pull-up resistors.		-82.2	μA	Other than those above.		-1	μA	Omitted						HIGH output current	I _{OH}	V _{OH} = V _{DD} -0.4V		-6	mA	Low level input current	I _{IL}	Pins have internal pull-up resistors.	-90	-	μA	Other than those above.	-1	-	μA	Omitted						High level output current	I _{OH}	V _{OH} = V _{DD} -0.4V	-	-6	mA
2	PRMV	10000	5000	X and Y axes perform a linear interpolation 1 with an end point (1000, 5000)																																																				
2	PRMV	10000	5000	Incremental movement 10000,5000.																																																				
LOW input current	I _{IL}	Terminals have internal pull-up resistors.		-82.2	μA																																																			
		Other than those above.		-1	μA																																																			
Omitted																																																								
HIGH output current	I _{OH}	V _{OH} = V _{DD} -0.4V		-6	mA																																																			
Low level input current	I _{IL}	Pins have internal pull-up resistors.	-90	-	μA																																																			
		Other than those above.	-1	-	μA																																																			
Omitted																																																								
High level output current	I _{OH}	V _{OH} = V _{DD} -0.4V	-	-6	mA																																																			

Revision	Date	Content
		<p>Correct: #RST input signal width Note 1 $8T_{CLK}$ ns</p> <p>Note 1. Input an L level signal with 8 or more reference clock cycles to #RST pin. Start CPU communication after inputting an H level signal of 8 cycles or more. ∴ Described in "7.5 Operation timing" in this manual.</p>
		



www.pulsemotor.com/global

Information

www.pulsemotor.com/global/contact

Fourth edition issued in June, 2022
Copyright 2009 Nippon Pulse Motor Co., Ltd.
