# Pulse Control LSI <br> PCL6045BL <br> 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 ..... 19
4.3 Pin definitions ..... 20
4.4 CPU interface ..... 26
4.4.1 68000 interface ..... 26
4.4.2 H8 interface. ..... 27
4.4.3 8086 interface ..... 28
4.4.4 Z80 interface ..... 29
5. Software description ..... 30
5.1 CPU communication ..... 30
5.1.1 Access method ..... 30
5.1.2 Address map ..... 30
5.1.3 Command Write ..... 32
5.1.4 Register Write ..... 35
5.1.5 Register Read ..... 37
5.1.6 Main status read ..... 39
5.1.7 General-purpose output port write. ..... 40
5.1.8 Sub-status \& General Purpose I/O port Read ..... 40
5.2 Status and General-purpose I/O port ..... 41
5.2.1 Main status (MSTS) ..... 41
5.2.2 Sub-status (SSTS) \& General Purpose I/O port (IOP) ..... 45
5.2.3 Extension status (RSTS) ..... 46
5.2.4 Interpolation status (RIPS) ..... 46
5.3 Commands ..... 47
5.3.1 Operation commands ..... 47
5.3.2 Control commands ..... 49
5.4 Registers ..... 54
5.4.1 Speed control register ..... 56
5.4.2 Position control register ..... 63
5.4.3 Environment setting register ..... 67
5.4.4 Counter register ..... 91
5.4.5 Comparator register ..... 93
5.4.6 Counter latch register ..... 95
5.4.7 Interrupt register ..... 97
5.4.8 Status display register. ..... 103
5.5 Operation mode ..... 107
5.5.1 Command control ..... 108
5.5.2 Positioning control ..... 108
5.5.3 Pulser control ..... 110
5.5.4 Switch control ..... 118
5.5.5 Origin return control ..... 121
5.5.6 Sensor control ..... 140
5.5.7 Linear interpolation 1 control ..... 143
5.5.8 Linear interpolation 2 control ..... 146
5.5.9 Circular interpolation control ..... 149
5.5.10 U-axis synchronous control ..... 152
6. Function description ..... 155
6.1 Reset ..... 155
6.1.1 Hardware reset ..... 156
6.1.2 Software reset ..... 156
6.2 Pre-register ..... 157
6.2.1 Continuous operation ..... 158
6.2.2 Continuous comparison. ..... 161
6.3 Speed control ..... 163
6.3.1 Speed patterns ..... 163
6.3.2 Speed setting example ..... 164
6.3.3 Manual correction calculation of the target speed ..... 165
6.3.4 Target speed override. ..... 169
6.3.5 Circular interpolation step number ..... 170
6.3.6 Constant synthesized speed control ..... 172
6.4 Position control ..... 173
6.4.1 Target position override 1 (RMV) ..... 173
6.4.2 Target position override 2 (PCS) ..... 176
6.4.3 End-point-draw operation ..... 178
6.5 Output pulse control ..... 179
6.5.1 Output pulse mode ..... 179
6.5.2 Operation complete timing ..... 181
6.5.3 Output pulse width control ..... 183
6.6 Idling control ..... 184
6.7 Mechanical external input control ..... 185
6.7.1 End limit (+EL, -EL) ..... 185
6.7.2 Slow-down (SD) ..... 187
6.7.3 Origin (ORG), Encoder Z phase (EZ) ..... 190
6.8 Servo motor driver interface ..... 192
6.8.1 Positioning complete (INP) ..... 192
6.8.2 Deviation counter clear (ERC) ..... 193
6.8.3 Alarm (ALM) ..... 195
6.9 External start / Simultaneous start ..... 196
6.9.1 Simultaneous start (CSTA) ..... 196
6.9.2 Own axis start (STA) ..... 199
6.9.3 Axis selection start (SELn) ..... 201
6.10 External stop / Simultaneous stop ..... 202
6.10.1 Simultaneous stop (CSTP) ..... 202
6.10.2 Axis selection stop (SELn) ..... 204
6.11 Emergency stop ..... 205
6.12 Counter ..... 207
6.12.1 Counter types and input specifications ..... 207
6.12.2 Counter clear ..... 211
6.12.3 Counter latch ..... 213
6.12.4 Counter count stop and input stop ..... 215
6.13 Comparator ..... 217
6.13.1 Comparator types and functions ..... 217
6.13.2 Software limit ..... 225
6.13.3 Out-of-step detection of stepping motor ..... 227
6.13.4 Index output ..... 231
6.13.5 Ring count ..... 233
6.13.6 Bulk override ..... 235
6.14 Backlash correction ..... 241
6.15 Slip correction ..... 242
6.16 Vibration suppression ..... 243
6.17 Synchronous start ..... 244
6.17.1 Start when the specified axis stops ..... 244
6.17.2 Start with internal sync signal ..... 252
6.17.3 Continuous interpolation using circular interpolation dummy operation ..... 256
6.18 Interrupt request (INT) ..... 257
6.18.1 Error interrupt ..... 258
6.18.2 Event interrupt ..... 259
6.18.3 Operation stop interrupt ..... 260
6.19 General-purpose one shot ..... 261
7. Electrical Characteristics ..... 262
7.1 Absolute maximum ratings ..... 262
7.2 Recommend operating conditions ..... 262
7.3 DC characteristics ..... 263
7.4 AC characteristics ..... 264
7.4.1 Reference clock ..... 264
7.4.2 CPU IF = 0 (68000) ..... 265
7.4.3 CPU IF = 1 (H8) ..... 267
7.4.4 CPU IF = 2 ( 8086 ) ..... 269
7.4.5 CPU IF = 3 (Z80) ..... 271
7.5 Operation timing ..... 273
7.5.1 RST signal ..... 274
7.5.2 SRST command ..... 274
7.5.3 EA, EB signals ..... 274
7.5.4 PA, PB signals ..... 275
7.5.5 Start command ..... 275
7.5.6 CSTA signal ..... 275

## 1. Introduction

Thank you for choosing our pulse control LSI, the "PCL6045BL."
This manual describes the specifications, functions, connections, and usages of PCL6045BL.
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.


$$
W \quad a \quad r \quad n \quad n \quad g
$$

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

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

## Caution

"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.

```
l m p o r t a n ce
```

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

## $R \quad e \quad m \quad a \quad r \quad k \quad s$

"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 "Important " 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 "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 can 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.

- FL speed

Lower frequency (start speed, stop speed)
Specify the starting speed of a motor [pps].

## - FH speed

Higher frequency (star speed, target speed)
Specify the max speed of a motor [pps].

- FA speed

Frequency for adjustment (backlash, slip, sensor detection)
Specify the max starting speed of a motor [pps].

### 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 all axes.
(2) Negative logic pins and negative logic signals are not indicated by overbars, or the like. For logic details, see "4.3 Pin definitions".
(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 $=19.6608 \mathrm{MHz}$ " 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, MSTS [7:0] represents the 7th to 0th bits of MSTS.)

### 1.2 Handling the product

### 1.2.1 Storing

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

### 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

## 4

C a $u \quad t \quad 0 \quad n$

- Keep the operating voltage, operating temperature, I/O 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 I/O 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 $V_{D D}$ 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, serge, or static electricity.


### 1.2.3.2 Precautions for transportation and storing LSIs

## ! <br> C a $u$ tion

- 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 $30^{\circ} \mathrm{C}$ and $70 \% \mathrm{RH}$ 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 $7.5 \times 10^{5}$ to $1 \times 10^{9} \Omega$ )
- 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

## ! $\quad$ C a $u$ tion

- Humidity should be 30 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 $1 \times 10^{9} \Omega$ or less)
- Use a conductive mat on the surface of the work desk and ground it.
(Resistance between surface and ground $7.5 \times 10^{5}$ to $1 \times 10^{9} \Omega$ )
- 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 $1 \times 10^{10} \Omega$ or less)
- Operators must wear wrist straps which are grounded through resistance. (When worn, resistance between surface and ground $7.5 \times 10^{5}$ to $3.5 \times 10^{7} \Omega$ )
- Handle the LSIs and the packages including LSIs with care and avoid mechanical vibration and impact.


### 1.2.3.4 Precautions for installation

## ! Caution

- 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 $30^{\circ} \mathrm{C}$ or less as well as $85 \% \mathrm{RH}$ or less for one year before opening the moisture-proof package. For one week after opening the moistureproof package at $30^{\circ} \mathrm{C}$ or less and $70 \%$ RH or less after opening the moisture-proof package. (storage rank: equivalent to MSL3)
- When moisture absorption is a concern, dry the package before reflowing.
- Drying is at $125 \pm 5^{\circ} \mathrm{C}$ 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 168 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.


- 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 $350^{\circ} \mathrm{C}$, 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.

Free repairs will only be conducted at NPM locations; no repairs will be made by business trips.
Warranty period of repaired product is the same as the warranty period before repair.
This warranty covers the product itself. The detriments or damages induced by the product failure etc. will not be covered by the warranty.

### 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

PCL6045BL 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 perform command inputs, data inputs/outputs, and interrupt request signal outputs, etc.

All written data can be read out to debug the application programs.
Motor speed can be performed at constant speed or at varied speed with acceleration/deceleration. For acceleration/ deceleration, you can select either linear or S-curve.

Motor positioning can be performed in incremental, absolute, by commands and by sensor signals.

## - CPU interface

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

1. 16 -bit interface for 68000
2. $\mathbf{1 6}$-bit interface for H 8
3. 16 -bit interface for 8086
4. 8 -bit interface for Z 80

## - Acceleration/deceleration speed control

Linear acceleration/deceleration and S-curve acceleration/deceleration are available.
For S-curve acceleration/deceleration, linear sections can be inserted in the middle by setting S-curve sections.
For S-curve sections setting, the acceleration and deceleration characteristics can be set independently.
You can perform S-curve deceleration after Linear acceleration, or vice versa.

- Slow-down point setting

You can set the slow-down point (deceleration start point) that is compared with the remaining number of pulses when decelerating and stopping in positioning control, etc. For example, set 100 pulses for the slow-down point if you want to start deceleration 100 pulses before the target position to stop. In this case, you can start decelerating 100 pulses before the target position regardless of whether the movement amount is 500 pulses or 1000 pulses. You can also set the optimum slow-down point automatically.

- 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.

## - Target position override

You can change the target position (feed amount) during operation.
The following two types of target position override functions are implemented:

1) Target position 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 constant speed or FH constant speed pattern) and will move in the reverse direction.
2) Continuous movement is performed until an external signal is input to start positioning controls. Positioning controls
start when the external signal is input.

- Triangle drive avoidance (FH correction function)

In the positioning mode, this function automatically lowers the target speed ( FH ) to avoid a triangle driving when the feeding amount is too small.

## - Pre-register

The 1st and 2nd data for continuous operation (feed amount, initial speed, moving speed, acceleration rate, deceleration rate, speed magnification rate, slow-down point, operation mode, center position 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 shifted to the current data, and the operation starts automatically. 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 (28 bit) | Command pulse |
| Counter 2 | Mechanical position control (28 bit) <br> (Can be used as a general-purpose <br> counter) | Encoder <br> Command pulse <br> Manual pulser |
| Counter 3 | Controlling the deviation between the <br> command position and the mechanical <br> position (16 bit) | Deviation between command pulse and encoder <br> Deviation between command pulse and manual pulser <br> Deviation between encoder and manual pulser |
| Counter 4 | IDX signal output (28 bit) <br> (Can be used as a general-purpose <br> counter) | Command pulse <br> Encoder <br> Manual pulser <br> $1 / 2$ 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, by inputting an LTC or ORG signal. You can clear them when latching. The Counter
1 , Counter 2, and Counter 4 have the ring count function that repeats counting the specified counting range.

## - Comparator

There are five comparators for each axis. They can compare the target values and the 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 entering the software limit range, the motor stops immediately or decelerate-stop. After that motor can only move in the opposite direction to the previous movement.

## - Backlash correction / Slip correction

Backlash correction function in gears, etc. can correct the feed amount each time the moving direction is changed. Slip correction functions such as with pulleys correct the feed amount every time regardless of the moving direction.

- Index output

PCL6045BL can output IDX signals for the index table 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 among multiple PCL6045BLs.

## - Simultaneous stop

You can stop any multiple axes simultaneously by writing a command, inputting a CSTP signal, or caused by abnormal stops of any axis. Any axis or any multiple axes can be selected from multiple PCL6045BLs.

If the CSTP signal is already ON at the start, the operation does not start.

- Vibration suppression

Set the frequency of vibration suppression so that one pulse each in reverse and forward direction are added just before stopping. By adding two pulses, vibration can be suppressed when stopping.

- Manual pulser input

By applying manual pulse signals, you can rotate a motor directly by one-pulse increment.
The input signals can be either 90 -degree phase difference signals ( $1 \mathrm{x}, 2 \mathrm{x}$, or 4 x ) or up/down signals.
For input signals, the magnification circuit of 1 to 32 times and the division circuit of $\frac{1}{2048}$ to $\frac{2048}{2048}$ 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 does not stop.
Therefore, the command pulse output in the opposite direction can be continued.

## - Drive switch input

You can operate a motor directly in the direction of travel by inputting signals by a drive switch.
There are two input pins for drive switch signals: + direction (+DRn) and - direction (-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 use it to avoid out-of-step in accelerations with a stepping motor.

- 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) Incremental movement and absolute movement in positioning operations
(3) Continuous movement, incremental movement and absolute movement in pulser controls
(4) Continuous movement and incremental movement in switch controls
(5) Origin return movement in origin return operations
(6) Sensor position movement in sensor controls
(7) Continuous movement and Incremental movement in the linear interpolation operation 1 (acceleration/ deceleration is available)
(8) Continuous movement and Incremental movement in the linear interpolation operation 2 (Available between multiple LSIs)
(9) Continuous movement and Incremental movement in the circular interpolation operations
(10) Incremental movement in U-axis synchronous control

## - Mechanical signal input

The following four signals can be input for each axis.
(1) +EL: When this signal is turned ON while moving in + direction, the axis stops immediately or decelerate-stop).

When this signal is already ON at start, no further movement made in + direction. (can move in - direction).
(2) -EL: Functions the same as +EL signal when moving in - direction.
(3) SD: Decelerates or decelerate-stops by software settings

In the deceleration setting, the axis will decelerate to FL speed if this signal is turned ON during high-speed operation. Also, the axis will perform constant speed operation if this signal is already ON at start. In the deceleration stop setting, the axis will decelerate to FL speed and stop if this signal is turned ON during high-speed operation. Also, the axis will not start if this signal is already ON at start.
(4) ORG: Input signal for an origin return operation

For safety, make sure that +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 by ELLn pin.
The input logics of SD and ORG signals can be changed by software.

## - Servomotor driver 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. The axis will not start if this signal is already ON at start.

The I/O logic of INP, ERC, and ALM signals can be changed by software settings.
ERC signal is a pulse output and the pulse length can be selected ( $11 \mu \mathrm{~s}$ to 100 ms . Level outputs are available).

## - Origin return sequences

In sensor control, encoder Z-phase (EZ), end limit (+EL, -EL), and slow-down (SD) signals can be used in addition to origin (ORG) signal to perform various origin return sequences.
<Examples of origin return sequence>

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 is turned ON.
5) Stops when EL signal in the operating direction is turned ON. After reversing, stops after the specified number of EZ signals is counted.
6) Decelerates when SD signal is turned ON and stops when ORG signal is turned ON.
7) Decelerates when ORG signal is turned ON and stops when the specified number of EZ signals is counted.
8) Decelerate-stops when ORG signal is tuned ON. After reversing, stops after the specified number of EZ signals is counted.
9) Decelerate-stops after ORG signal ON position is memorized. In reversing, stops at the memorized position.
10) After ORG signal is turned ON, the position where the specified number of EZ signals counts is memorized and decelerates and stops. After reversing, stops at the memorized position.
11) Decelerates and stops when EL signal in the operating direction is ON. In reversing, the position where the specified number of EZ signals counts is memorized 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, the operation will not start.

- Interrupt request output

L-level signals can be output from INT pin by various factors.
Output from INT pin by OR logic of each factor for each axis.
When multiple PCL6045BLs are used, wired OR connections are not possible.

### 2.2 Configuration

Controlling PCL6045BL requires a crystal oscillator outputting recommended frequency of 19.6608 MHz and a CPU or FPGA, etc. with a Parallel-bus interface with the 16 -bit or 8 -bit data bus.


## 3. Specifications

The following table shows the specifications of PCL6045BL such as the functions.

| Item | Description |
| :---: | :---: |
| Number of axes [axis] | 4 |
| Positioning control range [pulses] | $-134,217,728$ to $+134,217,727$ (28 bits) |
| Number of registers used for setting speeds [type/axis] | 3 (FL, FH, and FA) |
| Speed setting step number [steps] | 1 to 65,535 (16 bits) |
| Speed magnification range [Multiplication] | 0.1 to 100 <br> <Examples of the reference clock 19.6608 MHz <br> Multiply by 0.1 : $\quad 0.1$ to $6,553.5 \mathrm{pps}$ <br> Multiply by 1: $\quad 1$ to $65,535 \mathrm{pps}$ <br> Multiply by 100: 100 to 6,553,500 pps <br> (pps: pulse per second) |
| Slow-down point setting range [pulses] | 0 to 16,777,215 (24 bits) |
| Acceleration/deceleration characteristics | 4 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: <br> Counter 1: Command position counter (28 bits) <br> Counter 2: General-purpose counter 1 ( 28 bits) <br> Counter 3: Deviation counter <br> (16 bits) <br> Counter 4: General-purpose counter 2 ( 28 bits) |
| Comparators [Circuit/axis] | 5 (28 bits) |
| Interpolation functions | Linear interpolation 1: One set of any two or more axes <br> Linear interpolation 2: One set of any one or more axes (Interpolation operations by multiple PCL6045BLs are available). <br> Circular interpolation: One set of any two axes |
| Reference clock frequency [ MHz ] (fcık) | 19.6608 (Max: 20 MHz ) |
| CPU interface (Parallel-Bus) | 68000 (16 bits), H8 (16 bits), 8086 (16 bits), $\mathrm{Z80}$ (8 bits) |
| Package type | 176 pin LQFP |
| Package size [mm] | $24 \times 24$ (Mold part) |
| Weight [g] | 1.96 (typ.) |
| Power supply [V] | 3.3 (Single power supply) |
| Storage temperature [ ${ }^{\circ} \mathrm{C}$ ] ( $T_{\text {stg }}$ ) | -65 to +150 |
| Operating ambient temperature $\left[{ }^{\circ} \mathrm{C}\right]\left(T_{a}\right)$ | -40 to $+85 \quad\left(\mathrm{~T}_{\mathrm{j}}=-40\right.$ to $\left.+125\left[{ }^{\circ} \mathrm{C}\right], \quad \theta_{j-a}=34\left[{ }^{\circ} \mathrm{C} / \mathrm{W}\right]\right)$ |
| Chip configuration | C-MOS |

## 4. Hardware description

The following explains the connection between PCL6045BL and a CPU by showing the external dimensions and pin list.

### 4.1 External dimensions

A1 pin is located at the lower left of PCL6045BL model name marking.
(Equivalent to P-LQFP176-2424-0.50)


Pin detail drawing (5:1)

| Symbol | Dimension in Millimeters |  |  |
| :---: | :---: | :---: | :---: |
|  | Minimum | Nominal | Maximum |
| E | 23.90 | 24.00 | 24.10 |
| D | 23.90 | 24.00 | 24.10 |
| $\mathrm{H}_{\mathrm{E}}$ | 25.60 | 26.00 | 26.40 |
| $\mathrm{H}_{\mathrm{D}}$ | 25.60 | 26.00 | 26.40 |
| e | - | 0.50 | - |
| b | 0.17 | 0.22 | 0.27 |
| X | - | - | 0.08 |
| A | - | - | 1.70 |
| $\mathrm{A}_{1}$ | 0.00 | 0.10 | 0.20 |
| $\mathrm{A}_{2}$ | 1.30 | 1.40 | 1.50 |
| Y | - | - | 0.08 |
| L | 0.30 | 0.50 | 0.75 |
| $\mathrm{L}_{1}$ | 0.80 | 1.00 | 1.20 |
| c | 0.09 | 0.15 | 0.20 |
| $\theta$ | $0^{\circ}$ | $5^{\circ}$ | $10^{\circ}$ |

### 4.2 Pin assignment diagram

Pin 1 is located at the lower left of PCL6045BL model name marking.


### 4.3 Pin definitions

1. [I/O] column shows the direction of signals

I: Input, O: Output, B: Bidirectional
2. [Logic] column shows the logic of signals.

P: Positive logic, N: Negative logic, \#: Can be changed by software, \%: Can be changed by hardware.
3. [Resistance] column shows whether a pull-up resistor is built-in or not.

U: Pull-up resistor is built-in.
4. [Unused] column shows the recommended connection destination when not in use.

Open: Not connected, Vdd: Pull-up or VDD connection, Pup: Pull-up connection.
Pull-up resistors of 5 to $10 \mathrm{k} \Omega$ are recommended.
Some input pins and bidirectional pins have built-in pull-up resistors (described in "Resistance" column" or "7.3 DC characteristics". These built-in pull-up resistors are to prevent floating.

When you do not use the pins, it is recommended to add external pull-up resistors to improve noise resistance.
5. All signal input pins can input 0 to +5 V level.
6. Output pins of all signals can be pulled up to +5 V , but cannot output more than VDD.

A pull-up resistor value of $5 \mathrm{k} \Omega$ or higher is recommended.

| Signal name | Pin No. | 1/0 | Logic | Resistance | Unused | Description |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| GND | $\begin{aligned} & 17,25,39, \\ & 56,77,105, \\ & 127,163,176 \end{aligned}$ | - | - | - | - | Power supply pin connected to GND <br> Connect all GND pins to GND power supply. |
| VDD | $\begin{aligned} & 12,33,61, \\ & 88,100,121, \\ & 144,149, \\ & 161,162, \\ & 165,166,167 \end{aligned}$ | - | - | - | - | Power supply pin connected to +3.3 V <br> Connect all VDD pins to +3.3 V power supply. |
| RST | 175 | 1 | N | U | - | Input pin for the hardware reset signal (RST) For details, see "6.1.1 Hardware reset". |
| CLK | 164 | 1 | P | - | - | Input pin for the reference clock signal (CLK) Connect a crystal oscillator outputting the recommended frequency of 19.6608 MHz . |
| IF0, IF1 | $\begin{aligned} & 1, \\ & 2 \end{aligned}$ | 1 |  | U | - | Input pins for CPU interface selection For details, see "4.4 CPU interface". |


| Signal name | Pin No. | I/O | Logic | Resistance | Unused | Description |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| CS | 3 | 1 | N | - | - | Input pin for chip selection signal (CS) <br> Enable RD and WR pins by CS = L level. |
| RD | 4 | 1 | N | - | - | Input pin for read signal (RD) <br> Enable RD pin by CS = L level. |
| WR | 5 | 1 | N | - | - | Input pin for the write signal (WR) Enable WR pin by CS = L level. |
| A0, A1, <br> A2, A3, <br> A4, | $\begin{aligned} & 6,7 \\ & 8, \\ & 10 \end{aligned}$ | 1 | P | - | - | Input pins for address signal (A0 to A4) |
| INT | 11 | 0 | N | - | Open | Output pin for interrupt request signal See "6.18 Interrupt request (INT)" for details. |
| WRQ | 13 | 0 | N | - | Open | Output pin for wait request signal (WRQ) |
| IFB | 14 | 0 | N | - | Open | Output pin for interface running signal (IFB) |
| D0, D1, <br> D2, D3, <br> D4, D5, <br> D6, D7 | $\begin{array}{lll} 15, & 16, & 18, \\ 19, & 20, & 21, \\ 22, & 23 & \end{array}$ | B | P | - | - | Bit0 to Bit7 of the data bus are connected. |
| D8, D9, <br> D10, <br> D11, <br> D12, <br> D13, <br> D14, D15 | $\begin{array}{lll} 24, & 26, & 27, \\ 28, & 29, & 30, \\ 31, & 32 & \end{array}$ | B | P | - | Pup | Bit8 to Bit15 of the data bus are connected. For $\mathbf{Z 8 0}$-Bus ( 8 bits), pull-up connection is required. Eight data buses can be pulled up with just one pull-up resistor. |
| CSTA | 168 | B | N | U | Pup | Simultaneous start signal (CSTA) I/O pin. See "6.9.1 Simultaneous start (CSTA)" for details. |
| CSTP | 169 | B | N | U | Pup | Simultaneous stop signal (CSTP) I/O pin. See "6.10 External stop / Simultaneous stop". |
| CEMG | 170 | 1 | N | U | Vdd | Emergency stop signal (CEMG) input pin See "6.11 Emergency stop ". |
| ELLx, <br> ELLy, <br> ELLz, <br> ELLu | 171, <br> 172, <br> 173, <br> 174 | 1 | - | U | Vdd | Input pins to set the input logic of End limit signals( $\pm E L$ ). <br> See "6.7.1 End limit (+EL, -EL)". |


| Signal name | Pin No. | 1/0 | Logic | Resistance | Unused | Description |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { +ELx, } \\ & \text { +ELy, } \\ & \text { +ELz, } \\ & +E L u \end{aligned}$ | 34, <br> 66, <br> 97, <br> 130 | 1 | N\% | U | Vdd | +Direction End limit signal (+EL) input pins See "6.7.1 End limit (+EL, -EL)". |
| $\begin{aligned} & -E L x, \\ & -E L y, \\ & -E L z, \\ & -E L u \end{aligned}$ | 35, <br> 67, <br> 98, <br> 131 | 1 | N\% | U | Vdd | -Direction End limit signal (-EL) input pins See "6.7.1 End limit (+EL, -EL)" for details. |
| $\begin{aligned} & \text { SDx, } \\ & \text { SDy, } \\ & \text { SDz, } \\ & \text { SDu } \end{aligned}$ | 36, <br> 68, <br> 99, <br> 132 | 1 | N\# | U | Vdd | Slow-down signal (SD) input pins <br> See "6.7.2 Slow-down (SD)" for details. |
| ORGx, <br> ORGy, <br> ORGz, <br> ORGu | 37, <br> 69, <br> 101, <br> 133 | 1 | N\# | U | Vdd | Origin signal (ORG) input pins <br> See "6.7.3 Origin (ORG), Encoder Z phase (EZ)" <br> for details. |
| ALMx, <br> ALMy, <br> ALMz, <br> ALMu | $\begin{aligned} & 38, \\ & 70, \\ & 102, \\ & 134 \end{aligned}$ | 1 | N\# | U | Vdd | Alarm signal (ALM) input pins Connect to a servo motor driver. <br> See "6.8.3 Alarm (ALM)" for details. |
| OUTx, <br> OUTy, <br> OUTz, <br> OUTu | 57, 78, 122, 145 | 0 | N\# | - | Open | Command pulse signals (OUT/PLS/PHA) output pins <br> See "6.5 Output pulse control" for details. |
| DIRx, <br> DIRy, <br> DIRz, <br> DIRu | 58, 79, 123, 146 | 0 | N\# | - | Open | Command pulse signals (DIR/MNS/PHB) output pins <br> See "6.5 Output pulse control" for details. |
| EAx, <br> EAy, <br> EAz, <br> EAu | 40, <br> 71, <br> 103, <br> 135 | 1 | - | U | Vdd | Encoder A phase signal (EA) input pins See "6.12.1 Counter types and input specifications" for details. |
| EBx, <br> EBy, <br> EBz, <br> EBu | 41, 72, 104, 136 | 1 | - | U | Vdd | Encoder B phase signal (EB)input pins See "6.12.1 Counter types and input specifications" for details. |


| Signal name | Pin No. | 1/0 | Logic | Resistance | Unused | Description |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| EZx, <br> EZy, <br> EZz, <br> EZu | 42, <br> 73, <br> 106, <br> 137 | 1 | N\# | U | Vdd | Encoder Z phase signal (EZ) input pins See "6.7.3 Origin (ORG), Encoder Z phase (EZ)" for details. |
| PAx, <br> PAy, <br> PAz, <br> PAu | 43 , <br> 74, <br> 107, <br> 138 | 1 | - | U | Vdd | Manual pulser A-phase signal (PA) input pins See "5.5.3 Pulser control" for details. |
| $\begin{aligned} & \text { PBx, } \\ & \text { PBy, } \\ & \text { PBz, } \\ & \text { PBu } \end{aligned}$ | 44 , 75, 108, 139 | 1 | - | U | Vdd | Manual pulser B-phase (PB) signal input pins See "5.5.3 Pulser control". |
| PEx, <br> PEy, <br> PEz, <br> PEu | 45 , <br> 76, <br> 109, <br> 140 | 1 | N | U | Vdd | Manual pulser and drive switch effective signals (PE) input pins <br> See "5.5.3 Pulser control" and "5.5.4 Switch control". |
| $\begin{aligned} & \text { +DRx, } \\ & \text { +DRy, } \\ & \text { +DRz, } \\ & \text { +DRu } \end{aligned}$ | 46, <br> 82, <br> 110, <br> 141 | 1 | N\# | U | Vdd | +Direction drive switch (+DR) signal input pins See "5.5.4 Switch control". |
| $\begin{aligned} & \text {-DRx, } \\ & \text {-DRy, } \\ & \text {-DRz, } \\ & \text {-DRu } \end{aligned}$ | 47, <br> 83, <br> 111, <br> 142 | 1 | N\# | U | Vdd | -Direction drive switch (-DR) signal input pins See "5.5.4 Switch control" for details. |
| $\begin{aligned} & \text { PCSx, } \\ & \text { PCSy, } \\ & \text { PCSz, } \\ & \text { PCSu } \end{aligned}$ | 48 , <br> 84, <br> 112, <br> 143 | 1 | N\# | U | Vdd | Pulse count start signal (PCS) and Own axis start signal (STA) input pins. <br> See "6.4.2 Target position override 2 (PCS)" and "6.9.2 Own axis start (STA)". |
| INPx, <br> INPy, <br> INPz, <br> INPu | 49, <br> 85, <br> 113, <br> 150 | 1 | N\# | U | Vdd | In-Position signal (INP) input pins <br> Connect to a servo motor driver. <br> See "6.8.1 Positioning complete (INP)". |
| CLRx, <br> CLRy, <br> CLRz, <br> CLRu | 50, 86, <br> 114, 151 | 1 | N\# | U | Vdd | Counter clear signal input pins See "6.12.2 Counter clear". |


| Signal name | Pin No. | 1/O | Logic | Resistance | Unused | Description |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { LTCx, } \\ & \text { LTCy, } \\ & \text { LTCz, } \\ & \text { LTCu } \end{aligned}$ | 51, <br> 87, <br> 115, $152$ | 1 | N\# | U | Vdd | Counter latch signal (LTC) input pins See "6.12.3 Counter latch". |
| ERCx, <br> ERCy, <br> ERCz, <br> ERCu | 59, 80, 124, 147 | 0 | N\# | - | Open | Deviation counter clear (ERC)signal (ERC) output pins. <br> Connect to a servo motor driver. <br> See "6.8.2 Deviation counter clear (ERC)". |
| BSYx, <br> BSYy, <br> BSYz, <br> BSYu | 60, <br> 81, <br> 125, <br> 148 | 0 | N | - | Open | Busy signal (BSY) output pins See "5.2.1 Main status (MSTS)". |
| P0x, <br> P0y, <br> P0z, <br> POu | 52, <br> 89, <br> 116, $153$ | B | P | U | Pup | General-purpose I/O port 0 signal(P0)pins <br> This pin is also used for outputting acceleration signal (FUP). <br> See "5.4.3.3 RENV2: Environment setting 2". |
| $\begin{aligned} & \text { P1x, } \\ & \text { P1y, } \\ & \text { P1z, } \\ & \text { P1u } \end{aligned}$ | 53, <br> 90, <br> 117, <br> 154 | B | P | U | Pup | I/O pin for general-purpose I/O port 1 (P1). <br> This pin is also used for outputting deceleration signal (FDW). <br> See "5.4.3.3 RENV2: Environment setting 2". |
| $\begin{aligned} & \mathrm{P} 2 \mathrm{x}, \\ & \mathrm{P} 2 \mathrm{y}, \\ & \mathrm{P} 2 \mathrm{z}, \\ & \mathrm{P} 2 \mathrm{u} \end{aligned}$ | 54, 91, 118, 155 | B | P | U | Pup | I/O pin for the general-purpose I/O port 2 (P2). <br> This pin is also used for outputting constant speed in progress (MVC). <br> See "5.4.3.3 RENV2: Environment setting 2". |
| $\begin{aligned} & \text { P3x, } \\ & \text { P3y, } \\ & \text { P3z, } \\ & \text { P3u } \end{aligned}$ | 55, <br> 92, <br> 119, <br> 156 | B | P | U | Pup | I/O pin for general-purpose I/O port 3 (P3). <br> This pin is also used for outputting comparator 1 condition-met signal (CP1) or the plus soft limit signal (+SL). <br> See "5.4.3.3 RENV2: Environment setting 2". |
| P4x, <br> P4y, <br> P4z, <br> P4u | 62, 93, 120, 157 | B | P | U | Pup | I/O pin for general-purpose I/O port 4 (P4). <br> This pin is also used for outputting comparator 2 condition-met signal (CP2) or the minus softlimit signal (-SL). <br> See "5.4.3.3 RENV2: Environment setting 2". |


| Signal name | Pin No. | 1/0 | Logic | Resistance | Unused | Description |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { P5x, } \\ & \text { P5y, } \\ & \text { P5z, } \\ & \text { P5u } \end{aligned}$ | 63, <br> 94, <br> 126, <br> 158 | B | P | U | Pup | I/O pin for general-purpose I/O port 5 (P5). <br> This pin is also used for outputting comparator 3 condition-met signal (CP3). <br> See "5.4.3.3 RENV2: Environment setting 2". |
| $\begin{aligned} & \text { P6x, } \\ & \text { P6y, } \\ & \text { P6z, } \\ & \text { P6u } \end{aligned}$ | 64, <br> 95, <br> 128, <br> 159 | B | P | U | Pup | I/O pins for general-purpose I/O port 6 (P6). <br> This pin is also used for outputting Comparator <br> 4 condition-met signal (CP4) and outputting index signal (IDX). <br> See "5.4.3.3 RENV2: Environment setting 2". |
| P7x, <br> P7y, <br> P7z, <br> P7u | 65 96, 129, 160 | B | P | U | Pup | I/O pins for general-purpose I/O port 7 (P7). <br> This pin is also used for outputting Comparator 5 condition-met signal (CP5). <br> See "5.4.3.3 RENV2: Environment setting 2". |

### 4.4 CPU interface

The connection to a CPU can be selected from four types: 68000, H8, 8086, and Z 80 types.
Select the connection with IF1 and IF0 pins as follow.
The following table shows the correspondence between PCL6045BL pins and CPU pins.

| PCL6045BL pin name | IF1 | IF0 | RD | WR | A0 | WRQ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Type (bit width, CUP Interface \#) | L | L | VDD | R/W | LDS | DTACK |
| $68000(16,0)$ | L | H | RD | HWR | GND | WAIT |
| H8 $(16,1)$ | H | L | RD | WR | GND | READY |
| $8086(16,2)$ | H | H | RD | WR | A0 | WAIT |
| Z80 $(8,3)$ |  |  |  |  |  |  |

### 4.4.1 68000 interface

Interface of 16-bit range of R/W signal, LDS signal, and DTACK signal.
8 bits cannot be accessed.
The lower address corresponds to the upper word of I/O buffer.
Interface for VME busses and 68000-series CPU.


### 4.4.2 H8 interface

Interface of 16-bit width of RD signal, HWR signal, and WAIT signal.
8 bits cannot be accessed.
The lower-address corresponds to the upper-word of I/O buffer.
Interface for H 8 system CPU.


### 4.4.3 8086 interface

Interface of 16-bit width of RD signal, WR signal, and READY signal.
8-bit cannot be accessed.
The lower address corresponds to the lower word of I/O buffer.
Interface for 8086-series CPU.


### 4.4.4 Z80 interface

Interface of 8-bit range of RD signal, WR signal, and WAIT signal.
16-bit cannot be accessed.
The low-order address corresponds to the low-order byte of I/O buffer.
Interface for Z80 system CPU.


## 5. Software description

This chapter describes the communication from a CPU and shows the commands and registers for PCL6045BL.

### 5.1 CPU communication

The communication method from a CPU to PCL6045BL is via parallel communication.

### 5.1.1 Access method

Command writing and status reading directly access the target address.
Register reads and writes are accessed through I/O buffer by means of register control commands.
When reading from register, write register reading command and then read I/O buffer.
When writing to register, write to I/O buffer before writing to register writing command.
I/O buffer is located on the respective axis.

### 5.1.2 Address map

The address map differs depending on CPU to be connected.

### 5.1.2.1 $\mathbf{6 8 0 0 0}$ Communication address map

| Axis | A4, A3 | A2, A1 | Bits | R/W | Address name | Description |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| U | 11b | 11b | 16 | R | MSTSW | Main status |
|  |  |  | 16 | W | COMW | Axis Selection, Commands |
|  |  | 10b | 16 | R | SSTSW | Sub-status, General-purpose I/O port |
|  |  |  | 16 | W | OTPW | General-purpose output port |
|  |  | 01b | 16 | R/W | BUFW0 | I/O buffer Lower-data |
|  |  | 00b | 16 | R/W | BUFW1 | I/O buffer Upper-data |
| Z | 10b | (Same as U axis) |  |  |  |  |
| Y | 01b | (Same as U axis) |  |  |  |  |
| X | 00b | (Same as U axis) |  |  |  |  |

### 5.1.2.2 H8 communication address map

The address map of H 8 communication is the same as that of 68000 communication.
See "5.1.2.1 68000 Communication address map".

### 5.1.2.3 $\mathbf{8 0 8 6}$ Communication address map

| Axis | A4, A3 | A2, A1 | Bits | R/W | Address name | Description |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| X | 00b |  | 16 | R | MSTSW | Main status |
|  |  |  | 16 | W | COMW | Axis Selection, Commands |
|  |  |  | 16 | R | SSTSW | Sub-status, General-purpose I/O port |
|  |  |  | 16 | W | OTPW | General-purpose output port |
|  |  | 10b | 16 | R/W | BUFW0 | I/O buffer Lower-data |
|  |  | 11b | 16 | R/W | BUFW1 | I/O buffer Upper-data |
| Y | 01b | (Same as X axis) |  |  |  |  |
| Z | 10b | (Same as $X$ axis) |  |  |  |  |
| U | 11b | (Same as X axis) |  |  |  |  |

### 5.1.2.4 Z80 communication address map

| Axis | A4, A3 | A2~A0 | Bits | R/W | Address 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 I/O port |
|  |  |  | 8 | W | OTPB | General-purpose output port |
|  |  | 011b | 8 | R | SSTSB | Sub-status |
|  |  | 100b | 8 | R/W | BUFB0 | I/O buffer [7:0] |
|  |  | 101b | 8 | R/W | BUFB1 | I/O buffer [15:8] |
|  |  | 110b | 8 | R/W | BUFB2 | I/O buffer [23:16] |
|  |  | 111b | 8 | R/W | BUFB3 | I/O 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 Write

Axis selection (SELn) and commanding (COM) are written to 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 |  |  |  |  |  |  |  |  |

COMW.COMB1: This is the writing area for axis selection.
Writes a command to the axis among SELx to SELu where 1 has been written.
If more than one bit is set to 1 , the same command can be written to more than one selected axis.
If you select the axis here, you can control each axis only by the COMW of $X$ axis without using COMWs of $Y, Z$, and $U$ axes.

For example, if a 02 h with SELY $=1$ is written to COMB1 of $X$ axis, COMB0 of $X$ axis is performed with $Y$ axis only. If 0 is set to all SELn, only the own axis (axis selected by A4 and A3 pins) is regarded as selected.

COMW.COMB0: Command write area
For details on "Commands", see "5.3 Commands ".

Z80 communication always uses 8-bit transmission and writes to COMB1 first.
Write COMB1 address to axis selection 8 bit first, and then write the command 8 bit to COMB0 address.
For 16-bit transmission of Z80, COMB0 address may be written first.
Avoid the risk that the previous data in COMB1 address will not be updated and the command will be executed with an unintended axis selection.

In other communications, 16 -bit of axis selection and command is written to COMW address.

In the following cases, a wait time is required before the next access.

1. From writing any command to writing the next command
2. From writing register writing command until writing the next data to I/O buffer
3. From writing register reading command to reading data from I/O buffer

When using WRQ signal by CPU, CPU automatically reserves a standby period. WRQ signal will be at L level while CS signal and IFB signal are both at L level. IFB signal will be at L level for the waiting period that needs to be secured.

If the following access is performed without using WRQ signal, the waveform will be as shown by the dotted line in the figure below, and writing may fail.


If WRQ signal is not used in CPU, secure a waiting time of at least four CLK signal cycles by software.


Secure the time of 4 cycles or more of CLK signal by software.

If $W R Q$ is not used, check IFB $=\mathrm{H}$ level before accessing.

Sample software (H8 system):

| $/ /$ Writing STAFL (50h) Commands to X axis |  |
| :--- | :--- |
| var Address $=0 \times 3 ;$ | // Addresses : COMW of X axis |
| var Command $=0 \times 0050 ;$ | // Axis selection : Own axis only (00h) |
|  | // Command : STAFL $\quad$ (50h) |
| Writeln16bit (Address, Command) ; | // Write Commands to PCL6045BL |

## $R \quad e \quad m \quad a \quad k \quad s$

From now on, the programming language used in software sample is based on C\#.
The specifications of the program used are as follows.
var: Identifier of Variant type.
Address : In H8 system, 4-bit from A4 pin to A1 pin.
Writeln16bit(Address,Dt16bit): Prepare a method to write 16 bit to Address.
Writeln32bit(Address,Dt32bit): Prepare a method to write 32 bit to Address.
ReadOut16bit(Address,Dt16bit): Prepare a method to read 16 bit from Address.
ReadOut32bit(Address,Dt32bit): Prepare a method to read 32 bit from Address.

### 5.1.4 Register Write

Register Write data (BUF) writes to BUFW1 and BUFW0 addresses.

| BUFW1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BUFW0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| BUFB3 |  |  |  |  |  |  |  | BUFB2 |  |  |  |  |  |  |  |  | BUFB1 |  |  |  |  |  |  |  |  | BUFB0 |  |  |  |  |  |  |  |
| 3 | 30 | 29 | 28 | 827 | 726 | 2625 | 24 | 23 | 22 | 22 |  | 20 | 19 | 18 | 17 | 16 | 15 | 15 | 41 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BUF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

BUFW1 (BUFB3, BUFB2): Writes the upper data.
BUFW0 (BUFB1, BUFB0): Writes the lower data.

For Z 80 communication, register write data is written to BUFB3, BUFB2, BUFB1, and BUFB0 address.
For other communications, write the register writing data to BUFW1 and BUFW0.
Data written to I/O buffer can be written in any order.
Because the previous write or read data remains in I/O buffer, be sure to write 32-bit amount of data.

Wait for two CLK cycles for each data to be written.
After that, if entering the register writing command, you can write the data to the register in a batch. After writing register writing command, secure a wait time of four CLK cycles.


For details about register writing command, see "5.3.2.10 Register control commands".

If you specify axes when writing commands, you can write to the same register of multiple axes at the same time. In such cases, set the write data to I/O buffer for the respective axis.

Sample software (H8 system):

| $/ /$ The process to write data to PRMV register of X axis and Y axis simultaneously |  |
| :--- | :--- |
| Var Address = 0x0; | // Addresses: I/O buffer of X axis |
| Var BufData = 0x01234567; | // I/O buffer is 0123 4567h (19,088,743) |
| WriteIn32bit (Address, BufData); | // Write BufData to PCL6045BL |
| Address = 0x4; | // Address: Y axis's I/O buffer |
| BufData = 0x089ABCDE; | // I/O buffer is 089A BCDEh (-124,076,834) |
| WriteIn32bit (Address, BufData); | // Write BufData to PCL6045BL |
| Address = 0x3; | // Addresses: COMW of X axis |
| Var Command = 0x0380; | // Axis selection: Y axis and X axis (03h) |
| Writeln16bit (Address, Command); | // Command: WPRMV (80 h) |

### 5.1.5 Register Read

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

| BUFW1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | BUFW0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| BUFB3 |  |  |  |  |  |  |  | BUFB2 |  |  |  |  |  |  |  | BUFB1 |  |  |  |  |  |  |  |  | BUFB0 |  |  |  |  |  |  |  |
| 31 | 30 | 29 | 28 | 82 | 2726 | 25 | 24 | 23 | 22 | 21 | 120 | 19 | 18 | 17 | 16 | 15 |  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BUF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

BUFW1 (BUFB3, BUFB2): Read the upper- data.
BUFW0 (BUFB1, BUFB0): Read the lower-data.

When register reading command is written, register read data is batch-read to I/O buffer. In Z 80 communication, register read data is read to BUFB3, BUFB2, BUFB1, and BUFB0 addresses. In other communications, register data will be read out to BUFW1 and BUFW0. Data read through I/O buffer can be read in any order.


For details about register reading command, see "5.3.2.10 Register control commands".

If you specify the axes to read when the command is written, multiple axis can be read from the same register at the same time.

When reading the command position counter, etc., the data at the same timing can be read in all axes without latching. In these cases, the read data is set to I/O buffer for the respective axis.

Sample software (H8 system):

| // Reading register data from PRMV register of X axis and Y axis at the same time |  |
| :---: | :---: |
| var Address $=0 \times 3$ : | // Addresses: COMW of $X$ axis |
| var Command = 0x03C0; | // Axis selection: Y axis and X axis (03h) |
|  | // Command: RPRMV (C0h) |
| Writeln16bit (Address, Command); | // Write Command to PCL6045BL |
| Address $=0 \times 0$; | // Addresses: I/O buffer of X axis |
| var BufDatX; | // Data of I/O buffer in X axis |
| ReadOut32bit (Address, BufDatX); | // Read BufDatX from PCL6045BL |
| Address = 0x4; | // Address: I/O buffer of Y axis |
| var BufDatY; | // Data of I/O buffer in Y axis |
| ReadOut32bit (Address, BufDatY); | // Read BufDatY from PCL6045BL |

### 5.1.6 Main status read

Main status (MSTS) is read from 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): Read the main status.

In Z80 communication, the main status is read from MSTSB1, MSTSB0 addresses.
In other communications, the main status is read from MSTSW address.
The main status is updated by inputting more than one cycle of CLK while RD (LS) $=\mathrm{H}$ level or CS $=\mathrm{H}$ level.

## I mportance

If the read polling period is shorter than one CLK cycle, MSTS may not be updated.

For details, see "5.2.1 Main status (MSTS)".

Sample software (H8 system):

| // Reading main status from $X$ axis |  |
| :--- | :--- |
| var Address = $0 \times 3$; | // Address: MSTSW of $X$ axis |
| var MSTSW_x; | // Main status of $X$ axis |
| ReadOut16bit (Address, MSTSW_x); | // Read MSTSW_X from PCL6045BL |

### 5.1.7 General-purpose output port write

The status of the general-purpose output port (OTP) 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 status of the general-purpose output port.
OTP7 to OTP0 corresponds to P7n to P0n pin.
If writing 1 to the general-purpose output port, H -level is output from the general-purpose I/O
pin.
Nothing is output from the general-purpose I/O pin set to the general-purpose input port.
When the setting is changed from the input port to the output port, the written status is output.
Refer to "5.4.3.3 RENV2: Environment setting 2" for the general-purpose I/O pin.

In Z80 communication, the output status of the general-purpose output port is written to OTPB. In other communications, the output status of the general-purpose output port is written to OTPW. Write 0 to the upper part of 8 bit of OTPW address.

For details on the "General-purpose output port", also see "5.2.2 Sub-status (SSTS) \& General Purpose I/O port (IOP)".

### 5.1.8 Sub-status \& General Purpose I/O port Read

The status of sub-status (SSTS) and general-purpose I/O port (IOP) can be read from SSTSW.

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

STSW.SSTSB: Read sub-status.
SSTSW.IOPB: This command reads the status of the general-purpose I/O port.

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

In other communications, the statuses of sub-status and general-purpose I/O port are read from SSTSW.
Sub-status is updated by inputting more than one cycle of CLK while RD $(L S)=H$ level or CS $=\mathrm{H}$ level.
The general-purpose I/O port status is also updated by inputting one or more CLK cycles while RD (LS) $=\mathrm{H}$ level or CS $=\mathrm{H}$ level.

## l mportance

If the read polling cycle is shorter than one CLK cycle, SSTS and IOP may not be updated.

For details, see "5.2.2 Sub-status (SSTS) \& General Purpose I/O port (IOP)".

### 5.2 Status and General-purpose I/O port

There are four statuses as follows:

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

The status of the General-purpose I/O port (IOP) is described with the Sub-status.

### 5.2.1 Main status (MSTS)

The following items can be read: Operation statuses, with/without interrupts, the establishment status of comparators, and decision status of pre-registers.

| 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 |  |  | SINT | SERR | SEND | SENI | SRUN | SSCM |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 0 | SSCM | Start command write status. <br> 0 : Start command has not been written even once after operation is stopped or reset. <br> 1 : Start command has been written. |
| 1 | SRUN | Operation status of operation mode. <br> 0 : Stopped. H-level is output from BSY pin. <br> 1 : Operation in progress. L-level is output from BSY pin. |
| 2 | SENI | Operation stopped interrupt is generated. <br> 0 : Operation stopped interrupt has not occurred. Or RENV2.IEND $=0$ is set. <br> See "5.4.3.3 RENV2: Environment setting 2" for more information on RENV2.IEND bit. <br> 1 : Operation stopped interrupt occurred. The L-level can be output from INT pin. <br> When RENV5.MSMR $=0$, the count returns to 0 in three CLK cycles after reading. <br> When RENV5.MSMR $=1$, the count returns to 0 after writing $\operatorname{SENIR}(2 \mathrm{Dh})$ command. <br> See "5.4.3.6 RENV5: Environment setting 5" for more information on RENV5.MSMR bit. |
| 3 | SEND | Operation mode is stopped. <br> 0 : Start command has been written or has never been started after resetting. <br> 1 : Operation stopped. MTSTS.SRUN $=1$ is changed to MTSTS.SRUN $=0$. <br> You can confirm that you have started and stopped at least once. |
| 4 | SERR | Error interrupt has occurred. <br> 0 : No error interrupt has occurred. <br> 1 : An error interrupt has occurred. You can output L-level from INT pin. <br> When all the bits set to 1 in REST register become 0, MSTS.SERR bit returns to 0 . |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 5 | SINT | An event interrupt has occurred. <br> 0 : No event interrupt has occurred. <br> 1 : An event interrupt has occurred. You can output L-level from INT pin. <br> When all the bits set to 1 in RIST register become 0 , MSTS.SINT bit returns to 0 . |
| 7,6 | SSC | It shows the sequence number (RMD.MSN) during operation or when stopped. When you create a user program, you can control the operating blocks. <br> The sequence number does not affect operations. <br> See "5.4.3.1 RMD(PRMD): Operation mode" for RMD.MSN bits. |
| 8 | SCP1 | The result to compare with Comparator 1 <br> 0 : Comparator 1 comparison condition is not met. <br> 1 : Comparator 1 comparison condition is met. |
| 9 | SCP2 | The result to compare with Comparator 2 <br> 0 : Comparator 2 comparison condition is not met. <br> 1 : Comparator 2 comparison condition is met. |
| 10 | SCP3 | The result to compare with Comparator 3 <br> 0 : Comparator 3 compare condition is not met. <br> 1 : Comparator 3 comparison condition is met. |
| 11 | SCP4 | The result to compare with Comparator 4 <br> 0 : Comparator 4 compare condition is not met. <br> 1 : Comparator 4 comparison condition is met. |
| 12 | SCP5 | The result to compare with Comparator 5 <br> 0 : Comparator 5 compare condition is not met. <br> 1 : Comparator 5 comparison condition is met. |
| 13 | SEOR | The results of target position override trials. <br> It may change when writing WRMV command. <br> 0 : RMV register is written during operation. <br> Target position override has operated in time. <br> Or, target position override has not been tried. <br> 1 : Writing to RMV register while stopped. <br> Target position override was not in time and stopped. . <br> When RENV5.MSMR $=0$, the count returns to 0 in three CLK cycles after reading. <br> When RENV5.MSMR = 1 , it returns to 0 after SEORR (2Eh) command is written. <br> See "5.4.3.6 RENV5: Environment setting 5" for more information on RENV5.MSMR bit. |
| 14 | SPRF | Determined state of 2 nd pre-register for continuous operation data. <br> $0: 2$ nd pre-register for continuous operation data is in undetermined state. <br> 1: 2nd pre-register for continuous operation data is in determined state. |


| Bit | Name |  |
| :---: | :---: | :---: |
| 15 | SPDF | Determined state of 2nd pre-register for continuation compare data. |
|  |  | $0: 2$ nd pre-register for continuous compare data is in undetermined. |
|  |  |  |

The following figure shows the status bit change timing for common pulse mode.
The dashed line of MSTS.SEND bit in the bit change timing chart is the initial status immediately after resetting. MSTS.SENI bit ON changes from MSTS.SRUN bit OFF by CLK signal 1 cycle delay. For the "time when other signals change", see "7.5 Operation timing ".

1. Continuous movement by command control (RMD.MOD $=00 \mathrm{~h}$ )

MSTS.SRUN bit ON is delayed for BSY signal ON delay time (TCMDBSY) from MSTS.SSCM at the longest.

2. Continuous movement by pulser control (RMD.MOD $=01 \mathrm{~h}$ )

MSTS.SRUN bit ON is delayed for BSY signal ON delay time (Tcmbbsy) from PA signal ON at the longest.


[^0]3. Continuous movement by switch control (RMD.MOD $=02 \mathrm{~h}$ )

MSTS.SRUN bit ON is delayed for BSY signal ON delay time (TcmDBsy) from DR signal ON at the longest.

*2 OUT signal output is delayed for the start delay time (TcmDpls) from DR signal ON at the longest.
4. Incremental movement in Positioning operations (RMD.MOD $=41 \mathrm{~h}$ )

MSTS.SRUN bit ON is delayed for BSY signal ON delay (Tсмdвsy) from MSTS.SSCM bit at the longest.


MSTS.SRUN bit OFF changes from MSTS.SSCM bit OFF by CLK 1 cycle delay.

### 5.2.2 Sub-status (SSTS) \& General Purpose I/O port (IOP)

You can read the signal status of input pins, the speed during operation, the signal status of general-purpose I/O pins.

| 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 | $\begin{aligned} & \text { IOP0 } \\ & \text { to } \\ & \text { IOP7 } \end{aligned}$ | I/O status of general-purpose I/O ports, P7n to P0n pins. <br> 0 : L level <br> 1 : H level |
| 8 | SFU | Operation status of acceleration. <br> 0 : Other than accelerating <br> 1 : Accelerating |
| 9 | SFD | Operating status of deceleration. <br> 0 : Other than decelerating <br> 1 : Decelerating |
| 10 | SFC | Constant speed operation. <br> 0 : Other than constant speed <br> 1 : Constant speed |
| 11 | SALM | Input status of ALM signal. <br> The input state of this signal is through the input noise filter and with the input logic applied. $\begin{aligned} & 0: \text { OFF } \\ & 1: \text { ON } \end{aligned}$ <br> The input logic can be changed by RENV1.ALML bit. |
| 12 | SPEL | +EL signal input status. <br> The input state of this signal is through the input noise filter and with the input logic applied. $0: \text { OFF }$ $1: \mathrm{ON}$ <br> The input logic can be changed with ELLn pin. |
| 13 | SMEL | -EL signal input status. <br> The input state of this signal is through the input noise filter and with the input logic applied. $\begin{aligned} & 0: \text { OFF } \\ & 1: \text { ON } \end{aligned}$ <br> The input logic can be changed with ELLn pin. |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 14 | SORG | Input status of ORG signal. <br> The input state of this signal is through the input noise filter and with the input logic applied. $0: \text { OFF }$ $1: O N$ <br> The input logic can be changed with RENV1.ORGL bit. |
| 15 | SSD | The latch status of SD signal. <br> The input state of this signal is through the input noise filter and with the input logic applied. <br> 0 : OFF <br> 1 ON <br> Input logic can be changed by RENV1.SDL bit. <br> If SD signal is not latched (RENV1.SDLT $=0$ ), it is always OFF. <br> SD signal input status can be read from RSTS.SDIN bit. |

Even during operation, SSTS.SFU $=0$, SSTS.SFD $=0$ and SSTS.SFC $=0$ for backlash correction and slip correction operations. To check "during operation", read the main status (MSTS.SRUN = 1).

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


For example, in RENV2.P0M = 01b, OTP0 bit is output to P0 pin.
At this time, P0 pin output can be read by IOP.IOP0 bit.

### 5.2.3 Extension status (RSTS)

You can read the signal status, operating status, and operating direction of I/O 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 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

There are two types of commands: "Operation command" and "Control command".

### 5.3.1 Operation commands

This command starts and stops 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 become the start command for continuous operations.

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

For speed patterns, see "6.3.1 Speed pattern".

### 5.3.1.2 Remaining pulses start commands

These commands are used to output the remaining pulses when the following operations are stopped halfway:
RMD.MOD $=41 \mathrm{~h}, 42 \mathrm{~h}, 43 \mathrm{~h}, 44 \mathrm{~h}, 45 \mathrm{~h}, 47 \mathrm{~h}, 61 \mathrm{~h}, 63 \mathrm{~h}, 64 \mathrm{~h}, 65 \mathrm{~h}$
If writing while stopped, the remaining number of pulses (RPLS) is not updated to feeding amount (RMV), and operation mode starts.

Do not write during operation.

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

For speed patterns, see "6.3.1 Speed pattern".

### 5.3.1.3 Simultaneous start commands

CSTA, STA signal wait status (RSTS.CND = 0010b) axis can be started.

| COM | Symbol | Description |
| :---: | :---: | :--- |
| 06 h | CMSTA | CSTA signal is output from CSTA pin. <br> Operation mode can be started at all axes in CSTA, STA waiting state (RSTS.CND $=0010 \mathrm{~b})$. |
| $2 A \mathrm{~h}$ | SPSTA | Operation mode can be started at any axis in CSTA, STA waiting state (RSTS.CND $=0010 \mathrm{~b})$. <br> No CSTA is output from CSTA pin. |

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

### 5.3.1.4 Speed change commands

When written during operation, the target speed and speed pattern are changed.
If written while stopped, it will be ignored.

| COM | Symbol | Description |
| :---: | :--- | :--- |
| 40 h | FCHGL | Change to FL speed immediately. <br> Speed pattern becomes FL constant speed pattern and changes to FL speed immediately. |
| 41 h | FCHGH | Change to FH speed immediately. <br> Speed pattern becomes FH constant speed pattern and changes to FH speed immediately. |
| 42 h | FSCHL | Decelerate and change to FL speed. <br> Speed pattern becomes the speed pattern of high speed 1, decelerates and changes to FL speed. |
| 43 h | FSCHH | Accelerate and change to FH speed. <br> 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 and STA signal inputs as well as continuous operations by pre-registers.

| COM | Symbol | Description |
| :---: | :---: | :--- |
| 49 h | STOP | Stop immediately and complete the operation mode. |
| $4 A h$ | SDSTP | Decelerate-stop to complete the operation mode. <br> 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 |
| :---: | :---: | :--- |
| 07 h | CMSTP | CSTP signal is output from CSTP pin. <br> Multiple axes with CSTP signal input enabled can complete the operation mode. <br> When RENV1.STPM = 1 is set, operation decelerate-stops 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 emergency-stop an operation immediately.

| COM | Symbol | Description |
| :---: | :--- | :--- |
| 05 h | CMEMG | Stop emergently and cancel the operation mode. <br> Also cancel the continuous operation with pre-registers. |

For an emergency stop, see "6.11 Emergency stop"
For continuous operation, see "6.2.1 Continuous 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 |  |
| :---: | :---: | :--- |
| 00h | NOP | This command does not affect the operation. <br> Writing command will be processed. |

### 5.3.2.2 General-purpose output bit control commands

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

| COM | Symbol | Description |
| :---: | :---: | :---: |
| 10h | P0RST | Write 0 to OTP0 bit to reset P0 pin to L level. <br> When RENV2.P0M $=11 \mathrm{~b}$ and RENV2.P0L $=0$, general-purpose one-shot signal in negative logic is output. ${ }_{1}$ |
| 11h | P1RST | Write 0 to OTP1 bit to reset P1 pin to L level. <br> When RENV2.P1M $=11 \mathrm{~b}$ and RENV2.P1L $=0$, general-purpose one-shot signal in negative logic is output. ${ }^{*} 1$ |
| 12h | P2RST | Write 0 to OTP2 bit to reset P2 pin to L level. |
| 13h | P3RST | Write 0 to OTP3 bit to reset P3 pin to L level. |
| 14h | P4RST | Write 0 to OTP4 bit to reset P4 pin to L level. |
| 15h | P5RST | Write 0 to OTP5 bit to reset P5 pin to L level. |
| 16h | P6RST | Write 0 to OTP6 bit to reset P6 pin to L level. |
| 17h | P7RST | Write 0 to OTP7 bit to reset P7 pin to L level. |
| 18h | P0SET | Write 1 to OTP0 bit to set P0 pin to H level. <br> When RENV2.P0L=1 with RENV2.P0M=11b, general-purpose one-shot signal in positive logic is output. *1 |
| 19h | P1SET | Write 1 to OTP1 bit to set P1 pin to H level. <br> When RENV2.P1L = 1 with RENV2.P1M = 11b, general-purpose one-shot signal in positive logic is output $*_{1}$ |
| 1Ah | P2SET | Write 1 to OTP2 bit to set P2 pin to H level. |
| 1Bh | P3SET | Write 1 to OTP3 bit to set P3 pin to H level. |
| 1Ch | P4SET | Write 1 to OTP4 bit to set P4 pin to H level. |
| 1Dh | P5SET | Write 1 to OTP5 bit to set P5 pin to H level. |
| 1Eh | P6SET | Write 1 to OTP6 bit to set P6 pin to H level. |
| 1Fh | P7SET | Write 1 to OTP7 bit to set P7 pin to H level. |

For the batch control of general-purpose output ports, see "5.1.7 General-purpose output port write".
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 |
| :---: | :---: | :--- |
| 04 h | SRST | Reset PCL6045BL by software. <br> After writing this command, wait at least 12 cycles of CLK signal before restarting CPU access. |

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

For counters, see "6.12 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 driver.

| COM | Symbol | Description |
| :---: | :---: | :--- |
| 24 h | ERCOUT | Outputs ERC signal from ERC pin. |
| 25 h | ERCRST | Ends waiting for ERC signal ON width and OFF delay completion (RSTS.CND = 0101b). <br> Resets ERC signal output. |

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 |  |
| :---: | :---: | :--- |
| 26 h | PRECAN | Cancels the determined status in all pre-registers for continuous operation. |
| 27 h | PCPCAN | Cancels the determined status in the pre-register for comparator 5 comparison. |
| $2 B h$ | PRESHF | Shifts the data in all pre-registers for continuous operation. |
| 2 Ch | PCPSHF | Shifts the data in the pre-register for comparator 5 comparison. |
| 4 Fh | PRSET | Determines the pre-registers for continuous operation as the data for override. |

For pre-registers, 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 |  |
| :---: | :---: | :--- |
| 28 h | STAON | Used with target position override 2. <br> 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 latching.

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

For LTC signals, 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 MSTS.SENI to 0. |  |
| 2Eh | SEORR | Clears to MSTS.SEOR to 0. |  |

For interrupt controls, see "6.18 Interrupt request (INT)".
-

### 5.3.2.10 Register control commands

This command reads and writes 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 position | 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 | $\begin{aligned} & \text { RRENV } \\ & 1 \end{aligned}$ | 9Ch | WRENV1 | - | - | - | - | - |
| 14 | Environment setting 2 | RENV2 | DDh | $\begin{aligned} & \text { RRENV } \\ & 2 \end{aligned}$ | 9Dh | WRENV2 | - | - | - | - | - |
| 15 | Environment setting 3 | RENV3 | DEh | $\begin{aligned} & \text { RRENV } \\ & 3 \end{aligned}$ | 9Eh | WRENV3 | - | - | - | - | - |
| 16 | Environment setting 4 | RENV4 | DFh | $\begin{aligned} & \text { RRENV } \\ & 4 \end{aligned}$ | 9Fh | WRENV4 | - | - | - | - | - |
| 17 | Environment setting 5 | RENV5 | E0h | $\begin{aligned} & \text { RRENV } \\ & 5 \end{aligned}$ | A0h | WRENV5 | - | - | - | - | - |
| 18 | Environment setting 6 | RENV6 | E1h | $\begin{array}{\|l} \text { RRENV } \\ 6 \end{array}$ | A1h | WRENV6 | - | - | - | - | - |
| 19 | Environment setting 7 | RENV7 | E2h | $\begin{aligned} & \text { RRENV } \\ & 7 \end{aligned}$ | A2h | WRENV7 | - | - | - | - | - |
| 20 | Counter 1 (command position) | RCUN1 | E3h | $\begin{aligned} & \text { RRCUN } \\ & 1 \end{aligned}$ | A3h | WRCUN1 | - | - | - | - | - |
| 21 | Counter 2 (general-purpose 1) | RCUN2 | E4h | $\begin{aligned} & \text { RRCUN } \\ & 2 \end{aligned}$ | A4h | WRCUN2 | - | - | - | - | - |
| 22 | Counter 3 (deviation) | RCUN3 | E5h | $\begin{array}{\|l} \text { RRCUN } \\ 3 \end{array}$ | A5h | WRCUN3 | - | - | - | - | - |
| 23 | Counter 4 (general-purpose 2) | RCUN4 | E6h | $\begin{array}{\|l} \hline \text { RRCUN } \\ 4 \\ \hline \end{array}$ | A6h | WRCUN4 | - | - | - | - | - |
| 24 | Data for comparator 1 | RCMP1 | E7h | $\begin{aligned} & \text { RRCMP } \\ & 1 \end{aligned}$ | A7h | WRCMP1 | - | - | - | - | - |
| 25 | Data for comparator 2 | RCMP2 | E8h | $\begin{aligned} & \text { RRCMP } \\ & 2 \end{aligned}$ | A8h | WRCMP2 | - | - | - | - | - |
| 26 | Data for comparator 3 | RCMP3 | E9h | $\begin{aligned} & \text { RRCMP } \\ & 3 \end{aligned}$ | A9h | WRCMP3 | - | - | - | - | - |
| 27 | Data for comparator 4 | RCMP4 | EAh | $\begin{aligned} & \text { RRCMP } \\ & 4 \end{aligned}$ | AAh | WRCMP4 | - | - | - | - | - |
| 28 | Data for comparator 5 | RCMP5 | EBh | $\begin{aligned} & \text { RRCMP } \\ & 5 \end{aligned}$ | ABh | WRCMP5 | PRCP5 | CBh | RPRCP5 | 8Bh | WPRCP5 |
| 29 | Event interrupt request | RIRQ | ECh | RRIRQ | ACh | WRIRQ | - | - | - | - | - |


| 30 | Counter 1 <br> (command position) latch | RLTC1 | EDh | RRLTC1 | - | - | - | - | - | - | - |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 31 | Counter 2 <br> (general-purpose 1) latch | RLTC2 | EEh | RRLTC2 | - | - | - | - | - | - | - |
| 32 | Counter 3 (deviation) <br> latch | RLTC3 | EFh | RRLTC3 | - | - | - | - | - | - | - |
| 33 | Counter 4 <br> (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 <br> number | PSPD | F5h | RPSPD | - | - | - | - | - | - | - |
| 39 | Slow-down point <br> calculation value | RSDC | F6h | RRSDC | - | - | - | - | - | - | - |
| 40 | Number of circular <br> interpolation steps | RCI | FCh | RRCI | BCh | WRCI | PRCI | CCh | RPRCI | $8 C h$ | WPRCI |
| 41 | Circular interpolation step <br> 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. | Description | Name | Range | R/W | Pre-register | Type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Feed amount (target position) | RMV | $\begin{aligned} & -134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | 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 setting value | RDP | $\begin{gathered} -8,388,608 \text { to } \\ +8,388,607 \\ 0 \text { to } 16,77,215 \end{gathered}$ | R/W | PRDP | Position control |
| 8 | Operation mode | RMD | (4 byte) | R/W | PRMD | Environment setting |
| 9 | Circular interpolation center | RIP | $\begin{aligned} & -134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | 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 | $\begin{aligned} & -134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | R/W | - | Counter |
| 21 | Counter 2 (general-purpose 1) | RCUN2 | $\begin{aligned} & 134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | R/W | - | Counter |
| 22 | Counter 3 (deviation) | RCUN3 | $-32,768$ to +32,767 | R/W | - | Counter |
| 23 | Counter 4 (general-purpose 2) | RCUN4 | $\begin{aligned} & 134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | R/W | - | Counter |
| 24 | Comparison data for comparator 1 | RCMP1 | $\begin{aligned} & 134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | R/W | - | Comparator |
| 25 | Comparison data for comparator 2 | RCMP2 | $\begin{aligned} & 134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | R/W | - | Comparator |


| No. | Description | Name | Range | R/W | Pre-register | Type |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 26 | Comparison data for comparator 3 | RCMP3 | $\begin{aligned} & \text { 134,217,728 to } \\ & +134,217,727 \end{aligned}$ | R/W | - | Comparator |
| 27 | Comparison data for comparator 4 | RCMP4 | $\begin{aligned} & 134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | R/W | - | Comparator |
| 28 | Comparison data for comparator 5 | RCMP5 | $\begin{aligned} & 134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | R/W | PRCP5 | Comparator |
| 29 | Event interrupt request | RIRQ | (4 byte) | R/W | - | Interrupt control |
| 30 | Counter 1 (command position) latch | RLTC1 | $\begin{aligned} & 134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | R | - | Counter latch |
| 31 | Counter 2 (general-purpose 1) latch | RLTC2 | $\begin{aligned} & 134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | R | - | Counter latch |
| 32 | Counter 3 (deviation) latch | RLTC3 | $-32,768$ to $+32,767$ | R | - | Counter latch |
|  |  |  | 0 to 65,535 |  |  |  |
| 33 | Counter 4 (general-purpose 2) latch | RLTC4 | $\begin{aligned} & 134,217,728 \text { to } \\ & +134,217,727 \end{aligned}$ | 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 268,435,455 | R | - | Position control |
| 38 | Current speed step number | RSPD | (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 2,147,483,647 | R/W | PRCI | Position control |
| 41 | Circular interpolation step counter | RCIC | 0 to 2,147,483,647 | R | - | Position control |
| 42 | Interpolation status | RIPS | (4 byte) | R | - | Status display |

## C a ution

Register without the pre-register may perform unintended moves if data is written during operation.

### 5.4.1 Speed control register

These registers are for speed controls.

### 5.4.1.1 RFL(PRFL): FL speed step number


$\square$
Control commands: 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.

$$
\begin{aligned}
F L[p p s] & =R F L \times \frac{f_{C L K}[H z]}{(R M G+1) \times 65,536} \\
& =R F L \times M G
\end{aligned}
$$

$$
R F L=F L[p p s] \times \frac{(R M G+1) \times 65,536}{f_{C L K}[H z]}
$$

FL: FL speed
MG: Speed magnification

When the speed magnification is set to $1 x$, 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



Control commands: 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.

$$
\begin{aligned}
F H[p p s] & =R F H \times \frac{f_{C L K}[H z]}{(R M G+1) \times 65,536} \\
& =R F H \times M G
\end{aligned}
$$

$$
R F H=F H[p p s] \times \frac{(R M G+1) \times 65,536}{f_{C L K}[H z]}
$$

FH: FH speed
MG: Speed magnification

When the speed magnification is set to $1 x$, 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

 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |$\quad$ RUR(PRUR)

Control commands: RRUR(D3h), RPURU(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 RMD.MSMD bit and RUS register.

1. Linear acceleration (RMD.MSMD $=0$ )

$$
T U[s]=\frac{(R F H-R F L) \times(R U R+1) \times 4}{f_{C L K}[H z]} \quad R U R=\frac{f_{C L K}[H z] \times T U[s]}{(R F H-R F L) \times 4}-1
$$

TU: Acceleration time
2. S-curve acceleration without linear section (RMD.MSMD $=1$ and RUS $=0$ )

$$
T U[s]=\frac{(R F H-R F L) \times(R U R+1) \times 8}{f_{C L K}[H z]} \quad R U R=\frac{f_{C L K}[H z] \times T U[s]}{(R F H-R F L) \times 8}-1
$$

TU: Acceleration time
3. S-curve acceleration with linear section (RMD.MSMD $=1$ and RUS >0)

$$
T U[s]=\frac{(R F H-R F L+2 \times R U S) \times(R U R+1) \times 4}{f_{C L K}[H z]} \quad R U R=\frac{f_{C L K}[H z] \times T U[s]}{(R F H-R F L+2 \times R U S) \times 4}-1
$$

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



| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |$\quad$ RDR(PRDR)

Control commands: 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:
"Deceleration time = Acceleration time" must be set.
- Other than the above:
"Deceleration time $\leq$ Acceleration time $\times 2$ " must be satisfied.
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.

1. Linear deceleration (RMD.MSMD $=0$ )

$$
T D[s]=\frac{(R F H-R F L) \times(R D R+1) \times 4}{f_{C L K}[H z]} \quad R D R=\frac{f_{C L K}[H z] \times T D[s]}{(R F H-R F L) \times 4}-1
$$

TD: Deceleration time
2. S-curve deceleration without a linear section (RMD.MSMD $=1$ and RDS $=0$ )

$$
T D[s]=\frac{(R F H-R F L) \times(R D R+1) \times 8}{f_{C L K}[H z]} \quad R D R=\frac{f_{C L K}[H z] \times T D[s]}{(R F H-R F L) \times 8}-1
$$

TD: Deceleration time
3. S-curve deceleration with a linear section (RMD.MSMD $=1$ and RDS $>0$ )

$$
T D[s]=\frac{(R F H-R F L+2 \times R D S) \times(R D R+1) \times 4}{f_{C L K}[H z]} \quad R D R=\frac{f_{C L K}[H z] \times T D[s]}{(R F H-R F L+2 \times R D S) \times 4}-1
$$

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




Control commands: 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.

$$
M G=\frac{f_{C L K}[H z]}{(R M G+1) \times 65,536}
$$

$$
R M G=\frac{f_{C L K}[H z]}{M G \times 65,536}-1
$$

MG: Speed magnification
The following is an example of the speed magnification setting at $f_{C L K}=19.6608 \mathrm{MHz}$.

| Setting value | Speed magnification [Times] | Actual speed range [pps] |  | Setting value | Speed magnification [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

 $\left.\begin{array}{|l|l|l|l|l|l|l|l|l|l|l|l|l|l|l|l|l|}\hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{array}\right] \quad$ RUS(PRUS)

Control commands: RRUS(D9h), RPRUS(C9h), WRUS(99h), WPRUS(89h)
Register to set the S-curve section in S-curve acceleration.
PRUS register is the pre-register of RUS register.
The setting is enabled when s-curve acceleration/deceleration (RMD.MSMD $=1$ ) is set.

$$
\begin{aligned}
S_{S U}[p p s] & =R U S \times \frac{f_{C L K}[\mathrm{~Hz}]}{(R M G+1) \times 65,536} \\
& =R U S \times M G
\end{aligned}
$$

$$
R U S=S_{S U}[p p s] \times \frac{(R M G+1) \times 65,536}{f_{C L K}[H z]}
$$

$\mathrm{S}_{\text {su }}$ : S-curve acceleration section
MG: Speed magnification
The sections from FL speed to FL speed +Ssu and from FH speed -Ssu to FH speed are ones that accelerate in S-curve.
The section from FL speed $+\mathrm{S}_{\text {su }}$ to FH speed $-\mathrm{S}_{\text {su }}$ is one 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 , complete $S$-curve acceleration with no linear acceleration section is substituted for $\frac{R F H-R F L}{2}$.

### 5.4.1.7 RDS(PRDS): S-curve deceleration section

 $\left.\begin{array}{|l|l|l|l|l|l|l|l|l|l|l|l|l|l|l|l|l|}\hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{array}\right] \quad$ RDS(PRDS)

Control commands: 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.
The setting is enabled when s-curve acceleration/deceleration (RMD.MSMD $=1$ ) is set.

$$
S_{S D}[p p s]=R D S \times \frac{f_{C L K}[H z]}{(R M G+1) \times 65,536} \quad R D S=S_{S D}[p p s] \times \frac{(R M G+1) \times 65,536}{f_{C L K}[H z]}
$$

$S_{s d}$ : S-curve deceleration section

$$
=R D S \times M G
$$

MG: Speed magnification

The sections from FH speed to FH speed - SSD and from FL speed +SSD to FL speed are ones that decelerate in S-curve. The sections from FH speed - SSD to FL speed +SSD is one 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 , complete $S$-curve acceleration with no linear acceleration section is substituted for $\frac{R F H-R F L}{2}$.

### 5.4.1.8 RFA: FA speed step number




Control commands: 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.

$$
\begin{array}{rlr}
F A[p p s] & =R F A \times \frac{f_{C L K}[\mathrm{~Hz}]}{(R M G+1) \times 65,536} & R F A=F A[p p s] \times \frac{(R M G+1) \times 65,536}{f_{C L K}[H z]} \\
& =R F A \times M G & \text { FA: FA speed } \\
& \text { MG: Speed magnification }
\end{array}
$$

When the speed magnification is set to $1 x$, 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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | AS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 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 |  | IDC |  |  |  |  |  |

Control commands: RRSPD(F5h)
This register obtains the current speed step number, EZ signal count value, and idling count value.

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

### 5.4.2 Position control register

This register is for position control.

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


$\square$
Control commands: 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 $-134,217,728$ to $+134,217,727$.

The \# in bits 31 to 28 is the same value as bit 27 because they are sign extensions.
If register setting is positive and $+134,217,727,07 F F F F F F h$ can be read.
If register setting is negative and-134,217,728, F8000000h can be read.
Sign extension writes will be ignored.
For example, F8000000h is set whether the value to write is 08000000 h or F 8000000 h .

### 5.4.2.2 RIP(PRIP): Circular interpolation center

$$
\begin{array}{lllllllllllllllllllllllll}
31 & 30 & 29 & 28 & 27 & 26 & 25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6
\end{array} 5
$$



Control commands: 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 $-134,217,728$ to $+134,217,727$.

The \# in bits 31 to 28 is the same value as bit 27 because it is a sign extension.
If register setting is positive and $+134,217,727,07 F F F F F F h$ can be read.
If register setting is negative and $-134,217,728$, F 8000000 h can be read.
Sign extension writes are ignored.
For example, F8000000h is set whether the value to write is 08000000 h or F 8000000 h .

### 5.4.2.3 RCI(PRCI): Number of circular interpolation steps


0 RCI(PRCI)
Control commands: $\operatorname{RRCI}(F C h), \quad \mathrm{RPRCI}(\mathrm{CCh}), \quad \mathrm{WRCI}(\mathrm{BCh}), \quad$ 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 auto settings when performing deceleration control in circular interpolation.

For the number of circular interpolation steps, see "6.3.5 Circular interpolation step number".
The setting range is 0 to $2,147,483,647$.

### 5.4.2.4 RDP(PRDP): Slow-down point




Control commands: 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.
<In case of the 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 becomes less than or equal to the RSDC register.
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 $-8,388,608$ ( 800000 h ) to $8,388,607$ (7FFFFFh).
The \#s in bits 31 to 24 is the same value as bit 23 because it is a sign extension.
If register setting is positive and $+8,388,607,007$ FFFFFh can be read.
If register setting is-8,388,608 in negative, FF800000h can be read.
Sign extension writes are ignored.
For example, FF800000h is set whether the writing is 00800000 h or FF800000h.
<In 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 RDP 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.

1. Linear deceleration (RMD.MSMD $=0$ )

$$
R D P[\text { pulse }]=\frac{\left(R F H^{2}-R F L^{2}\right) \times(R D R+1)}{(R M G+1) \times 32,768}
$$

When FH correction function OFF (RMD.MADJ = 1) is set, the optimum value for a triangular drive are as follows.
(Do not change the value to be set in RFH register)

$$
R D P[p u l s e]=\frac{R M V \times(R D R+1)}{R U R+R D R+2}
$$

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 $=0$ )

$$
R D P[\text { pulse }]=\frac{\left(R F H^{2}-R F L^{2}\right) \times(R D R+1) \times 2}{(R M G+1) \times 32,768}
$$

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

$$
R D P[p u l s e]=\frac{(R F H+R F L) \times(R F H-R F L+2 \times R D S) \times(R D R+1)}{(R M G+1) \times 32,768}
$$

Setting range is from 0 to $16,777,215$ (00FFFFFFh).
Set 0 to \# of bits 31 to 24 .

### 5.4.2.5 RSDC: Slow-down point auto calculation value

 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RSDC |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

Control commands: RRSDC(F6h)
This register obtains the automatic calculation of slow-down point. Decelerates when RPLS register is smaller than RSDC register.

For example, when register is $-8,388,608,00800000 h$ is read out.

### 5.4.2.6 RCIC: Circular interpolation step number



| 0 | RCIC |
| :--- | :--- |

Control commands: RRCI(FDh)
Register to obtain step number for circular interpolation count of circular interpolation.
Reading from any axis is the same because all axes are the same register.
When circular interpolation is started, RCIC register is updated with RCI register.
RCIC register counts down to 0 for each pulse output of circular interpolation.
If the step number for circular interpolation setting is large, circular interpolation stops even if RCIC $>0$.
If the step number for circular interpolation setting is small, circular interpolation operates even after RCIC $=0$. For more information on "step number for circular interpolation", see "6.3.5 Circular interpolation step number ".

### 5.4.2.7 RPLS: Remaining pulse number



| 0 | 0 | 0 | 0 | RPLS |
| :--- | :--- | :--- | :--- | :--- | Control commands :RRPLS(F4h)

Register to obtain the remaining number of pulses up to target position.
When RMV register is changed, RPLS register is recalculated and updated.
RPLS register is recalculated and updated even when positioning control is started.
When starting other than positioning control, RPLS register is updated with RMV register.
If it is incremental moving, RPLS $=$ RMV.
When absolute position value is specified, RPLS = | RMV - RCUN1| or RPLS = | RMV - RCUN2 $\mid$.
The remaining number of pulses is counted down for each pulse output.
For positioning control, the operation mode is completed when RPLS $=0$.

### 5.4.3 Environment setting register

This register is for environment settings.

### 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 | 0 | MPIE | MADJ | MSPO | MSPE | MAX |  |  |  | MSY |  | MSN |  |

Control commands: 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 |  |
| :---: | :---: | :--- | :--- |


| Bit | Name | Description |
| :---: | :---: | :---: |
|  |  | 1000111 (47h): Timer operation mode in positioning control. <br> 1010001 (51h): Operation mode of incremental movement in pulser control. <br> 1010010 (52h): Operation mode in which the absolute position is specified with counter 1 in pulser control. 1010011 (53h): Operation mode in which the absolute position is specified with counter 2 in pulser control. 1010100 (54h): Operation mode for returning to zero point with counter 1 in pulser control. 1010101 (55h): Operation mode for returning to zero point with counter 2 in pulser control. <br> 1010110 (56h): Operation mode of incremental movement in switch control. <br> 1100000 (60h): Operation mode of continuous movement in linear interpolation 1 control. 1100001 (61h): Operation mode of incremental movement in linear interpolation 1 control. <br> 1100010 ( 62 h ): Operation mode of continuous movement in linear interpolation 2 control. 1100011 (63h): Operation mode of incremental movement in linear interpolation 2 control. <br> 1100100 (64h): Operation mode of circular interpolation in the CW direction in circular interpolation control. <br> 1100101 (65h): Operation mode of circular interpolation in the CCW direction in circular interpolation control. <br> 1100110 (66h): Operation mode of circular interpolation in the CW direction in U-axis synchronous control. 1100111 (67h): Operation mode of circular interpolation in the CCW direction in U-axis synchronous control. <br> 1101000 (68h): Operation mode of continuous movement with linear interpolation 1 in pulser control. 1101001 (69h): Operation mode of incremental movement with linear interpolation 1 in pulser control. 1101010 (6Ah): Operation mode of continuous movement with linear interpolation 2 in pulser control. 1101011 (6Bh): Operation mode of incremental movement with linear interpolation 2 in pulser control. 1101100 (6Ch): Operation mode of circular interpolation in the CW direction in pulser control. 1101101 (6Dh): Operation mode of circular interpolation in the CCW direction in pulser control. 110 1111(6Fh): Dummy operation mode with circular interpolation control. <br> Do not set any other values. <br> For details on "operation mode", see "5.5 Operation mode". |
| 7 | MENI | Even if the operation stop interrupt is enabled (RENV2.IEND = 1), the operation stop interrupt bit (MSTS.SENI) can be disabled when the pre-register is determined (RSTS.PFM = 10b or 11 b ). <br> 0 : Also MSTS.SENI $=1$ when RSTS.PFM $=10 \mathrm{~b}$ or 11 b <br> 1: It will NOT MSTS.SENI $=1$ when RSTS.PFM $=10 \mathrm{~b}$ or 11 b <br> The occurrence of an operation stop interrupt can be suppressed while the next continuous operation remains. |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 8 | MSDE | Sets the input function of SDn pins. <br> 0: General-purpose input pin. <br> 1: Decelerate or decelerate-stop when SD signal turns ON. SD signal status is obtained from RSTS.SDIN. |
| 9 | MINP | Sets the input function of INPn pin. <br> 0 : General-purpose input pin <br> 1: Operation mode is completed when INP signal is ON The input status of INP signal is obtained in RSTS.SINP bit. See "6.8.1 Positioning complete (INP)" for INP signals. |
| 10 | MSMD | Sets the characteristics of acceleration / deceleration. <br> 0: Linear acceleration / deceleration <br> 1: S-curve acceleration / deceleration <br> Please set RMD.MSMD $=1$ for linear acceleration, S-curve deceleration, or a combination of S-curve acceleration and linear deceleration. <br> If you set a small value to the $S$-curve section ( $R U S=1, R D S=1$ ), the characteristics will be almost the same as a linear acceleration or a linear deceleration. |
| 11 | MCCE | Sets the counting function of counter 1. <br> 0 : Count <br> 1: Does not count. Pulses can be output while counter 1 is stopped |
| 12 | METM | Sets the completion timing of an operation. <br> 0: Output pulse cycle completed <br> 1: Output-pulse ON width completed. The operation completion timing is advanced by OFF cycle of the last pulse. <br> When using vibration restriction function, set the output-pulse cycle completion (RMD.METM $=0$ ). <br> Set RMD.METM $=0$ to continue operation by pre-register. |
| 13 | MSDP | Sets how to set the slow-down point. <br> 0 : Automatic setting <br> 1: Manual setting <br> When using the automatic setting in circular interpolation control, set the number of circular interpolation steps in RCI register. |
| 14 | MPCS | Sets the input function of PCSn pin. <br> When RENV1.PCSM $=1$, it can serve as the STA signal input pin to start the own axis only. <br> 0 : General-purpose input pin <br> 1: PCS signal input pin for target position override 2 <br> For the target position override 2, see "6.4.2 Target position override 2 (PCS)" <br> The status of an input signal to PCSn pin is obtained in RSTS.SPCS bit. |
| 15 | MIPF | Sets the constant synthesized speed control in an interpolation operation. <br> 0 : Synthesized speed is NOT controlled to be constant <br> 1 : Synthesized speed is controlled to be constant <br> For constant synthesized speed constant control, see "6.3.6 Constant synthesized speed control" |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 17,16 | MSN | Sets the 2-bit sequence number. <br> The sequence number is obtained in MSTS.SSC bit <br> The sequence number does not affect the operation <br> It can be used to control operation blocks when creating the control software. <br> For the control of operation blocks, see "6.2.1 Continuous operation ". |
| 19,18 | MSY | Sets the start timing after writing to start command. <br> 00b: Start immediately. <br> 01b: If RENV1.PCSM $=0$, start with CSTA signal ON or SPSTA(2Ah) command. <br> If RENV1.PCSM $=1$, start with STA signal ON or SPSTA(2Ah) command. <br> 10b: Start with the inner synchronous signal (RENV5.SYI). <br> 11b: Starts when the specified axis (RMD.MAX) is stopped. <br> CSTA can be input via CSTA pin. You can also enter it with CMSTA (06h) command. STA can be input via PCSn pin. |
| 23: 20 | MAX | Sets axis to be used for checking stoppage at RMD.MSY $=11 \mathrm{~b}$. <br> e.g.:0001b: Starts when $X$ axis stops. <br> 0010b: Starts when $Y$ axis stops. <br> 0100b: Starts when $Z$ axis stops. <br> 1000b: Starts when U axis stops. <br> 0101b: Starts when both $X$ axis and $Z$ axis stop. <br> 1111b: Starts when all axes stop. <br> To include the stopping of own axis as the condition, set RENV2.SMAX to 1 as well. |
| 24 | MSPE | Sets the input function of CSTP pin. <br> 0 : General-purpose input pin <br> 1 : Stops by inputting CSTP signal <br> The status of CSTP pin is obtained by RSTS.SSTP bit. |
| 25 | MSPO | Sets the output function of CSTP pin. <br> 0 : General-purpose output pin. <br> You can output a negative logic one-shot pulse with CMSTP (07h) command. <br> 1 : Output a negative logic one-shot pulse when own axis stops abnormally. |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 26 | MADJ | Sets the FH correction function. <br> 0 : Automatic correction (automatically avoid a triangular drive). <br> 1: Manual correction (Not automatically avoid a triangular drive). <br> When combining the following settings, make sure to set the automatic correction (RMD.MADJ = 0). <br> - Incremental movement by linear interpolation 1 (RMD.MOD $=61 \mathrm{~h}$ ) <br> - Selects S-curve acceleration / deceleration (RMD.MSMD = 1) <br> - Enables constant synthesized speed control (RMD.MIPF = 1) <br> - Selects slow down point auto setting (RMD.MSDP=0) <br> Also set the manual correction (RMD.MADJ $=1$ ) to obtain the operating time accurately. <br> Obtain the FH speed in manual correction in "6.3.3 Manual correction calculation of the target speed". |
| 27 | MPIE | Sets the end-point-draw function. <br> 0 : Stops on the arc without the point-draw operation in circular interpolation control. <br> 1: Moves to the end-point with the end-point-draw operation in circular interpolation control. <br> For "End-point-draw operation", see "6.4.3 End-point-draw operation ". |
| 31: 28 | 0 | Always set it to 0 . |

### 5.4.3.2 RENV1: Environment setting 1



Control commands: RENV1(DCh), WRENV1(9Ch)
A register to set the specifications of I/O pins.


| Bit | Name | Description |
| :---: | :---: | :---: |
| 3 | ELM | Sets the input processing of EL signal ON in the operating direction. <br> 0 : Immediate stop <br> 1: Decelerate-stop <br> If decelerate-stop is set, deceleration starts by EL signal ON in the operating direction. <br> Even if EL signal becomes OFF during this deceleration, deceleration will continue to be stopped. <br> Be careful of collision because the motor passes through the EL position before stopping. |
| 4 | SDM | Sets the input processing of SD signals. <br> 0 : Decelerates <br> 1: Decelerate-stop <br> For "SD signals", see "6.7.2 Slow-down (SD) ". |
| 5 | SDLT | Sets the latch function of SD signal inputs. <br> Used when the signal width of SD signals is too narrow. <br> 0: No latch SD signals. <br> Input status of SD signals are obtained in RSTS.SDIN bit. <br> 1: Latch SD signals. <br> Latch status of SD signals are obtained in SSTS.SSD bit <br> If SD signal is OFF at start, SSTS.SSD $=0$ <br> When you set RENV1.SDLT $=0$, SSTS.SSD $=0$. <br> For "SD signals", see "6.7.2 Slow-down (SD)". |
| 6 | SDL | Sets the input logic of SD signals. <br> 0 : Negative logic <br> 1: Positive logic |
| 7 | ORGL | Sets the input logic of ORG signal. <br> 0 : Negative logic <br> 1: Positive logic |
| 8 | ALMM | Sets the input processing of ALM signal. <br> 0 : Immediate stop <br> 1: Decelerate-stop <br> When deceleration stop is set, decelerating start at ALM ON. <br> Even if ALM is turned OFF during this deceleration, deceleration will continue to be stopped. |
| 9 | ALML | Sets the input logic of ALM signal. <br> 0 : Negative logic <br> 1: Positive logic |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 10 | EROE | Sets the output function of ERCn pin at the immediate stop due to an abnormal stop factor. <br> ERC signal can be output at the time of immediate stop by +EL, -EL, ALM, CEMG signal and CEMG command (05h). <br> 0 : No ERC signal is output. <br> 1: ERC signal is output. <br> When both RMD.MOD $=20 \mathrm{~h}$ and 28 h , ERC signal is output at the immediate stop by EL signal in the moving direction ON. <br> For details on "ERC signal", see "6.8.2 Deviation counter clear (ERC) ". |
| 11 | EROR | Sets the output function of ERCn pin at the stop due to an origin return factor. <br> 0 : No ERC signal is output. <br> 1: ERC signal is output. <br> For ERC signals, see "6.8.2 Deviation counter clear (ERC)". |
| 14:12 | EPW | Sets the ON-width of ERC signal. <br> If RENV1.EPW $=111 \mathrm{~b}$ is set, level signals are output instead of pulse signals. <br> The output level signals are turned OFF by ERCRST (25h) command. |
| 15 | ERCL | Sets the output logic of ERC signal. <br> 0: Negative logic <br> 1: Positive logic |
| 17,16 | ETW | Sets the OFF delay-time of ERC signal. <br> 00b: $0 \mu \mathrm{~s} \quad 01 \mathrm{~b}: 11$ to $13 \mu \mathrm{~s} \quad$ 10b: 1.4 to $1.6 \mathrm{~ms} \quad 011 \mathrm{~b}: 93$ to 100 ms |
| 18 | STAM | Sets the input specifications of CSTA signal. <br> 0 : Level trigger <br> 1: Edge trigger (falling edge) |
| 19 | STPM | Sets the input processing of CSTP signal. <br> 0: Immediate stop <br> 1: Decelerate-stop |
| 20 | CLRL | Sets the input logic for CLR signal. <br> 0: Negative logic <br> 1: Positive logic |
| 21 | CLRM | Sets the input specifications of CLR signal. <br> 0: Edge trigger (From OFF to ON) <br> 1: Level trigger |
| 22 | INPL | Sets the input logic of INP signal. <br> 0 : Negative logic <br> 1: Positive logic |



| Bit | Name | Description |
| :---: | :---: | :---: |
| 31 | PDTC | Sets the output pulse width control function. <br> 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. <br> 1: The output pulse width fluctuates 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 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 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 |  |  | EZL | EDIR |  |  | PINF | EINF | P1L | P0L |

Control commands: RRENV2(DDh), WRENV2(9Dh)

Registers to set the specifications of general-purpose I/O pins: EA, EB signals as well as PA, PB signals.

| Bit | Name | Description |
| :---: | :---: | :---: |
| 1,0 | P0M | Sets the I/O function of P0n pin. <br> 00b: General-purpose input pin <br> 01b: General-purpose output pin <br> 10b: Outputs FUP signal in acceleration. The output logic is set by RENV2.P0L bit <br> 11b: Outputs a general-purpose one-shot signal. The output logic is set by RENV2.P0L bit |
| 3,2 | P1M | Sets the I/O function of P1n pin. <br> 00b: General-purpose input pin <br> 01b: General-purpose output pin <br> 10b: Outputs FDW signal in deceleration. The output logic is set by RENV2.P1L bit <br> 11b: Outputs a general-purpose one-shot signal. The output logic is set by RENV2.P1L bit |
| 5,4 | P2M | Sets the I/O function of P 2 n pin. <br> 00b: General-purpose input pin <br> 01b: General-purpose output pin <br> 10b: Outputs MVC signal in negative logic during constant speed operation <br> 11b: Outputs MVC signal in positive logic during constant speed operation |
| 7,6 | P3M | Sets the I/O function of P3n pin. <br> 00b: General-purpose input pin <br> 01b: General-purpose output pin <br> 10b: Outputs CP1 signal in negative logic while the comparator 1 condition is met. <br> 11b: Output CP1 signal in positive logic while the comparator 1 condition is met. |
| 9,8 | P4M | Sets the I/O function of P 4 n pin. <br> 00b: General-purpose input pin <br> 01b: General-purpose output pin <br> 10b: Outputs CP2 signal in negative logic while the comparator 2 condition is met. <br> 11b: Outputs CP2 signal in positive logic while the comparator 2 condition is met. |
| 11,10 | P5M | Sets the I/O function of P5n pin. <br> 00b: General-purpose input pin <br> 01b: General-purpose output pin <br> 10b: Outputs CP3 signal in negative logic while the comparator 3 condition is met. <br> 11b: Output CP3 signal in positive logic while the comparator 3 condition is met. |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 13,12 | P6M | Sets the I/O function of P6n pin. <br> 00b: General-purpose input pin <br> 01b: General-purpose output pin <br> 10b: Outputs CP4 signal in negative logic while comparator 4 condition is met. <br> 11b: Outputs CP4 signal in positive logic while comparator 4 condition is met. |
| 15,14 | P7M | Sets the I/O function of P7n pin. <br> 00b: General-purpose input pin <br> 01b: General-purpose output pin <br> 10b: Outputs CP5 signal in negative logic while the comparator 5 condition is met. <br> 11b: Outputs CP5 signal in positive logic while the comparator 5 condition is met. |
| 16 | P0L | Sets the output logic of FUP signal and general-purpose one-shot signal that can be output from POn pin. <br> 0 : Negative logic <br> 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. <br> 0 : Negative logic <br> 1: Positive logic |
| 18 | EINF | Sets the input noise filters for $E A, E B$, and $E Z$ signals. <br> 0 : Responds certainly to pulse signals of $0.05 \mu$ s or more width <br> 1: Ignores completely pulse signals of $0.10 \mu \mathrm{~s}$ or less width Responds certainly to pulse signals of $0.15 \mu$ s or more width |
| 19 | PINF | Sets the input noise filters for PA and PB signals. <br> 0 : Responds certainly to pulse signals of $0.05 \mu$ s or more width <br> 1: Ignores completely pulse signals of $0.10 \mu \mathrm{~s}$ or less width Responds certainly to pulse signals of $0.15 \mu$ s or more width |
| 21,20 | EIM | Sets input specifications for $E A$ and $E B$ signals. <br> 00b: 90-degree phase difference mode 1 x <br> 01b: 90-degree phase difference mode $2 x$ <br> 10b: 90-degree phase difference mode $4 x$ <br> 11b: 2-pulse mode. <br> See "6.12.1 Counter type and input specifications" for details, |
| 22 | EDIR | Sets the counting direction of EA and EB signals. <br> 0 : Counts up while the phase of EA signal is advanced $\ln 90$-degree phase difference mode. Counts up at the rising edge of EA signal in 2-pulse mode <br> 1: Counts up while the phase of EB signal is advanced In 90-degree phase difference mode. Counts up at the rising edge of EB signal in 2-pulse mode |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 23 | EZL | Sets the input logic of EZ signal. <br> 0 : Negative logic <br> 1: Positive logic <br> Counts down of RSPD.EZC bit when EZ signal changes from OFF to ON. |
| 25,24 | PIM | Sets the input specifications for PA and PB signals. <br> 00b: 90-degree phase difference mode 1 x <br> 01b: 90-degree phase difference mode $2 x$ <br> 10b: 90-degree phase difference mode $4 x$ <br> 11b: 2-pulse mode. <br> See "5.5.3 Pulser control" for details. |
| 26 | PDIR | Sets the counting direction of PA and PB signals. <br> 0 : Counts up while the phase of PA signal is advanced. <br> 1: Counts up while the phase of PB signal is advanced. |
| 27 | IEND | Sets the functional specifications of operation stop interrupt (MSTS.SENI). <br> 0 : Disabled. Maintain MSTS.SENI $=0$ at operation stop. <br> 1: Enabled. Changeable to MSTS.SENI $=1$ at operation stop. <br> See "6.18.3 Operation stop interrupt" for the details. |
| 28 | PMSK | Sets the output function of the command pulse. <br> 0 : Enabled. Outputs the command pulse <br> 1: Disabled. Does not output the command pulse In either case, the counter works. |
| 29 | SMAX | Sets the functional specifications when RMD.MAX bit includes the own axis with RMD.MSY $=11 \mathrm{~b}$. <br> 0 : Does not start when the own axis stops at the end. <br> 1: Starts even when the own axis stops at the end. <br> See "6.17.1.1 Stop selection of own axis interrupt" for the details. |
| 30 | EOFF | Sets the input function of EA and EB signals. <br> 0: Enabled <br> 1: Disabled. Not detect the input error, either. |
| 31 | POFF | Sets the input functions of PA and PB signals. <br> 0: Enabled <br> 1: Disabled. Not detect an input error, either. |

### 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 | Cl 4 |  | Cl3 |  | Cl 2 |  | 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 |

Control commands: RRENV3(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 | Select the origin return method: <br> 0000b : Origin return 0 (ORG signal, deceleration stop) <br> 0001b : Origin return 1 (ORG signal, reversal, FA speed, immediate stop) <br> 0010b : Origin return 2 (ORG signal, EZ signal, immediate stop) <br> 0011b : Origin return 3 (ORG signal, EZ signal, deceleration stop) <br> 0100b : Origin return 4 (ORG signal, reversal, FA speed, EZ signal, immediate stop) <br> 0101b : Origin return 5 (ORG signal, reversal, EZ signal, deceleration stop) <br> 0110b : Origin return 6 (EL signal, reversal, FA speed, immediate stop) <br> 0111b : Origin return 7 (EL signal, reversal, FA speed, EZ signal, immediate stop) <br> 1000b : Origin return 8 (EL signal, reversal, EZ signal, deceleration stop) <br> 1001b : Origin return 9 (Origin return 0, 0-point return) <br> 1010b: Origin return 10 (Origin return 3, 0-point return) <br> 1011b : Origin return 11 (Origin return 5, 0-point return) <br> 1100b : Origin return 12 (Origin return 8, 0-point return) <br> For "Origin return control pattern", see "5.5.5.1.1 Origin return 0 (0000b)" onward. |
| 7:4 | EZD | Sets the initial input count value of EZ signal used for an origin return control and a sensor control. <br> The setting range is 0000b ( 1 time) to 1111 b ( 16 times). |
| 9,8 | C12 | Sets the count target for counter 2. <br> 00b : EA and EB signals <br> 01b : Command pulse signal <br> 10b: PA and PB signals <br> 11b: Setting prohibited |
| 11,10 | Cl3 | Sets the count target for counter 3. <br> 00b: Deviation count between command pulse signal and EA and EB signals <br> 01b: Deviation count between command pulse signal and PA and PB signals <br> 10b: Deviation count of EA and EB signals and PA and PB signals <br> 11b: Prohibited |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 13,12 | Cl4 | Sets the count target of counter 4. <br> 00b: Command pulse signal <br> 01b: EA and EB signals <br> 10b: PA and PB signals <br> 11b: $\frac{f_{C L K}}{2}$ signal |
| 14 | BSYC | Sets the count limit for counter 4. <br> 0 : No limit <br> 1: Count only when BSY = L level <br> When RENV3.CI4 = 11b, the operating time can be measured with counter 4. |
| 15 | 0 | Always set to 0 . |
| 16 | CU1C | Sets whether or not to clear counter 1 when CLR signal is ON. <br> 0 : Not clear <br> 1: Clear. |
| 17 | CU2C | Sets whether or not to clear counter 2 when the CLR signal is ON. <br> 0 : Not clear <br> 1: Clear. |
| 18 | CU3C | Sets whether or not to clear counter 3 when the CLR signal is ON. <br> 0 : Not clear <br> 1: Clear |
| 19 | CU4C | Sets whether or not to clear counter 4 when the CLR signal is ON. <br> 0 : Not clear <br> 1: Clear |
| 20 | CU1R | Sets whether or not to clear counter 1 when the origin is reached in the origin return control. <br> 0 : Not clear <br> 1: Clear |
| 21 | CU2R | Sets whether or not to clear counter 2 when the origin is reached by the origin return control. <br> 0 : Not clear <br> 1: Clear |
| 22 | CU3R | Sets whether or not to clear counter 3 when the origin is reached by the origin return control. <br> 0 : Not clear <br> 1: Clear |
| 23 | CU4R | Sets whether or not to clear counter 4 when the origin is reached by the origin return control. <br> 0 : Not clear <br> 1: Clear |
| 24 | CU1B | Sets whether or not counter 1 counts the command pulses during backlash correction or slip correction. <br> 0 : Not count <br> 1: Count |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 5 | CU2B | Sets whether or not counter 2 counts the command pulses even during backlash compensation and slip compensation. Enabled when counter 2 is set to count command pulses (RENV3.CI2 $=01$ b). <br> 0 : Not count <br> 1: Count |
| 26 | CU3B | Sets whether or not counter 3 counts the command pulses even during backlash correction and slip correction. Enabled when counter 3 is set to count command pulses (RENV3.CI3 $=00 \mathrm{~b}, 01 \mathrm{~b}$ ) <br> 0 : Not count <br> 1: Count |
| 27 | CU4B | Sets whether or not counter 4 counts the command pulses even during backlash correction and slip correction. Enabled when counter 4 is set to count command pulses (RENV3.CI4 $=00 \mathrm{~b}$ ) <br> 0 : Not count <br> 1: Count |
| 28 | 0 | Always set to 0 . <br> (The count function of counter 1 is set by the RMD.MCCE bit.) |
| 29 | CU2H | Sets whether or not counter 2 counts. <br> 0 : Count <br> 1: Not count |
| 30 | CU3H | Sets whether or not counter 3 counts. <br> 0 : Count <br> 1: Not count |
| 31 | CU4H | Sets whether or not counter 4 counts. <br> 0 : Count <br> 1: Not count |

### 5.4.3.5 RENV4: Environment setting 4

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

Control commands: 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. <br> 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 <br> If RENV4.C1C $=10 \mathrm{~b}$, compare with the absolute value of RCUN3 register ( 0 to 32,767 ). |
| 4:2 | C1S | Sets the comparison conditions for Comparator 1. <br> 001b: RCMP1 = Comparison target (regardless of counting direction) <br> 010b: RCMP1 = Comparison target (only during count-up) <br> 011b: RCMP1 = Comparison target (only during count-down) <br> 100b: RCMP1 > Comparison target <br> 101b: RCMP1 < Comparison target <br> 110b: Plus side software-limit (RCMP1 < RCUN1) Also set RENV4.C1C $=00 \mathrm{~b}$. <br> Others: Comparison conditions are always un-met. |
| 6,5 | C1D | Sets the processing when the condition of Comparator 1 is met. <br> 00b: Do nothing. It can be used to output INT or CP1 signals, and to perform an internal synch start. <br> 01b: Immediate stop <br> 10b: Decelerate-stop <br> 11b: Bulk overrides <br> If RENV4.C1S $=110 \mathrm{~b}$ is set, stop immediately even when RENV4.C1D $=00 \mathrm{~b}$ or 11 b is set. |
| 7 | C1RM | Counter 1 can be ring-counted up to the P1 register. <br> 0 : Normal count <br> 1: Ring count. |
| 9,8 | C2C | Sets the comparison target of Comparator 2. <br> 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 <br> If RENV4.C2C $=10 \mathrm{~b}$, compare with the absolute value of RCUN3 register ( 0 to 32,767 ). |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 12: 10 | C2S | Sets the comparison conditions for Comparator 2. <br> 001b: RCMP2 = Comparison target (regardless of counting direction). <br> 010b: RCMP2 = Comparison target (only during count-up). <br> 011b: RCMP2 = Comparison target (only during count-down). <br> 100b: RCMP2 > Comparison target. <br> 101b: RCMP2 < Comparison target. <br> 110b: Minus side software limit (RCMP2 > RCUN1). Also set RENV4.C2C $=00 \mathrm{~b}$. <br> Others: Comparison conditions are always un-met. |
| 14,13 | C2D | Sets the processing when the condition of Comparator 2 is met. <br> 00b: Do nothing. It can be used to output INT signals and CP2 signals, and to perform an internal synch start. <br> 01b: Immediate stop <br> 10b: Decelerate-stop <br> 11b: Overrides at a time <br> If RENV4.C2S $=110 \mathrm{~b}$ is set, stop immediately when RENV4.C2D $=00 \mathrm{~b}$ or 11 b is set. |
| 15 | C2RM | Counter 2 can be ring-counted up to the RCMP2 register value. <br> 0 : Normal count <br> 1: Ring count. |
| 17,16 | C3C | Sets the comparison target of Comparator 3. <br> 00b:RCUN1 01b: RCUN2 10b:RCUN3 11b: RCUN4 <br> If RENV4.C3C $=10 \mathrm{~b}$, compare with the absolute value of RCUN3 register ( 0 to 32,767 ). |
| 20: 18 | C3S | Sets the comparison conditions for Comparator 3. <br> 001b: RCMP3 = comparison target <br> (Regardless of counting direction). <br> 010b: RCMP3 = Comparison target (only during count-up). <br> 011b: RCMP3 = Comparison target (only during count-down). <br> 100b: RCMP3 > Comparison target. <br> 101b: RCMP3 < Comparison target. <br> 110b: Setting prohibited <br> Others: Comparison conditions are always un-met. |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 22,21 | C3D | Sets the processing when the condition of Comparator 3 is met. <br> 00b: Do nothing. It can be used to output INT signals and CP3 signals, and to perform an internal synchronization start. <br> 01b: Stops immediately. <br> 10b: Decelerate-stop. <br> 11b: Overrides at a time. |
| 23 | IDXM | Sets the output conditions of IDX signal. <br> 0 : When RCUN4 = RCMP4 is established, IDX signal is output at the level. <br> 1: When changing to RCUN4 $=0$, IDX signal in a CLK signal 2-cycle width is pulse-output. IDX signal is the logical output set in RENV2.P6M bit. |
| 25,24 | C4C | Sets the comparison target of Comparator 4. <br> 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 <br> If RENV4.C4C $=10 \mathrm{~b}$, compare with the absolute value in RCUN3 register ( 0 to 32,767 ). |
| 29: 26 | C4S | Sets the comparison conditions for Comparator 4. <br> 0001b: RCMP4 = Comparison target <br> (regardless of counting direction) <br> 0010b: RCMP4 = Comparison target (only during count-up) <br> 0011b: RCMP4 = Comparison target (only during count-down) <br> 0100b: RCMP4 > Comparison target <br> 0101b: RCMP4 < Comparison target <br> 0111b: Comparison condition is not always met. <br> 1000b: IDX signal is output under the comparison condition of RENV4.IDXM bit (regardless of the counting direction). <br> 1001b: IDX signal is output under the comparison condition of RENV4.IDXM bit (only during count-up). <br> 1010b: IDX signal is output under the comparison condition of RENV4.IDXM bit (only during count- down). <br> Others: Comparison conditions are always un-met. <br> When using RENV4.C4S $=1000 \mathrm{~b}, 1001 \mathrm{~b}, 1010 \mathrm{~b}$, also set RENV4.C4C $=11 \mathrm{~b}$. <br> In this case, when using RENV4.IDXM $=1$, set a positive value in RCMP4 register. |
| 31: 30 | C4D | Sets the processing when the condition of Comparator 4 is met. <br> 00b: Do nothing. It can be used to output INT signals and CP4 signals, and to perform an internal synchronization start. <br> 01b: Immediate stop <br> 10b: Decelerate-stop <br> 11b: Overrides at a time. |

### 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 |  | PDSM | 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 |  |  |  |  |  |  |

Control commands : 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. <br> 000b: RCUN1 001b: RCUN2 010b: RCUN3 011b: RCUN4 <br> 100b: RPLS (number of remaining pulses) 101b: RSPD.AS (current speed) <br> 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. <br> 001b: RCMP5 = Comparison target (regardless of counting direction) <br> 010b: RCMP5 = Comparison target (only during count-up) <br> 011b: RCMP5 = Comparison target (only during count-down) <br> 100b: RCMP5 > Comparison target <br> 101b: RCMP5 < Comparison target <br> Others: Comparison conditions are always un-met. |
| 7,6 | C5D | Sets the processing when the condition of Comparator 5 is met. <br> 00b: Do nothing. It can be used to output INT signals and CP5 signals, and to perform an internal synchronization start. <br> 01b: Immediate stop <br> 10b: Decelerate-stop <br> 11b: Overrides at a time. |
| 10:8 | IDL | Sets the number of idling pulse outputs. <br> 000b: No idling pulse is output. <br> 001b to 111b: Zero to six pulses are output. |
| 11 | 0 | RMD.MOD $=01 \mathrm{~h}$ and 02 h are used to set the operation when EL signal is ON and stopped. <br> 0 : Continues operation mode. No error interrupt is generated. <br> To exit EL position value, simply input the reverse-direction signal. <br> 1: Complete operation mode. An error interrupt is generated. <br> To exit EL position, write start command, then input the reverse-direction signals. |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 13,12 | LTM | Sets the timing to latch RCUN1 register to RLTC1 register. <br> This setting also latches RCUN2 and 4 registers into RLTC2 and 4 registers. <br> The target to be latched to RLTC3 register is selected by RENV5.LTFD bit. <br> 00b: From LTC signal OFF to ON <br> 01b: From ORG signal OFF to ON <br> 10b: When the condition of comparator 4 is met. <br> 11b: When the condition of comparator 5 is met. |
| 14 | LTFD | Sets the target to be latched in RLTC3 register. <br> 0: Counter 3 (RCUN3) <br> 1: Present Velocity Steps (RSPD.AS) |
| 15 | LTOF | Sets to latch only at the write timing of LTCH (29h) command. <br> 0 : Latches also at the timing selected by the RENV5.LTM bit <br> 1: Latches only at the write timing of LTCH (29h) command |
| 19:16 | SYO | Sets the generation timing of the internal synchronization signal. <br> 0001b: When Comparator 1 condition is met 0010b: When Comparator 2 condition is met <br> 0011b: When Comparator 3 condition is met 0100b: When Comparator 4 condition is met <br> 0101b: When Comparator 5 condition is met <br> 1000b: At the start of acceleration 1001b: At the end of acceleration <br> 1010b: At the start of deceleration 1011b: At the end of deceleration <br> Other: Internal synchronization signal is not generated. <br> See "6.17.2 Start with internal sync signal" for details. |
| 21,20 | SYI | Sets the input target of the internal synchronization signal. <br> 00b: X-axis internal synchronization signal 01b: Y-axis internal synchronization signal <br> 10b: Z-axis internal synchronization signal 11b: U-axis internal synchronization signal |
| 22 | MSMR | Sets how to clear MSTS.SENI and MSTS.SEOR bits. <br> 0 : Cleared by reading the main status. <br> 1: Not cleared reading the main status. <br> In either case, the following interrupt control commands are enabled: MSTS.SENI bit can be cleared by writing SENIR (2Dh) command. <br> MSTS.SEOR bit can be cleared by writing SEORR (2Eh) command. |
| 23 | ISMR | Sets how to clear the bits in RIST and REST registers. <br> 0: Each register is cleared by writing a read command to each register. <br> 1: Writing a read command to each register does not clear each register. <br> 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. <br> 0 : Not clear counter 1 to 0 . <br> 1: Clears counter 1 to 0. |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 25 | CU2L | Sets the function to clear counter 2 to 0 immediately after latching counter 2. <br> 0 : Not clear counter 2 to 0 . <br> 1: Clears counter 2 to 0. |
| 26 | CU3L | Sets the function to clear counter 3 to 0 immediately after latching counter 3. <br> 0 : Not clear counter 3 to 0 . <br> 1: Clears counter 3 to 0 . |
| 27 | CU4L | Sets the function to clear counter 4 to 0 immediately after latching counter 4. <br> 0 : Not clear counter 4 to 0 . <br> 1: Clears counter 4 to 0 . |
| 31: 28 | 0 | Always set to 0 . |

### 5.4.3.7 RENV6: Environment setting 6



Control commands : 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. <br> 00b: Not correct the feed amount 01b: Backlash correction <br> 10b: Slip correction <br> 11b: Prohibited |
| 14 | 0 | Always set to 0 . |
| 15 | PSTP | Sets the processing when writing a stop command in pulser control. <br> 0: Stops by ignoring the PA and PB signals that have been input. <br> 1: Outputs the command pulse corresponding to the PA and PB signals that have been input, and then stops. <br> Stops ignoring RENV6.PSTP = 1 in interpolation controls ( $68 \mathrm{~h}, 69 \mathrm{~h}, 6 \mathrm{Ah}, 6 \mathrm{Bh}, 6 \mathrm{Ch}$ and 6Dh). |
| 26:16 | PD | Sets the numerator for dividing the input of PA and PB signals. <br> 0 : Do not divide. <br> 1 to 2047: The frequency is divided by the setting value/2048. |
| 31:27 | PMG | Sets the value for multiplying the input of PA and PB signals. 0 to 31: Multiplies the setting value by adding 1. |

For details on "backlash correction", see "6.14 Backlash correction ".
For details on "slip correction", see "6.15 Slip correction ".

### 5.4.3.8 RENV7: Environment setting 7



Register to set the control time of vibration suppression function.

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

See "6.16 Vibration suppression" for vibration suppression function.

### 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)


$\square$ RCUN1

Control commands : RRCUN1(E3h), WRCUN1(A3h)
Register to obtain the count value in counter 1 (Command position).
Dedicates for counting command pulse.
The setting range is $-134,217,728$ to $+134,217,727$.

The \# in bits 31 to 28 is the same value as bit 27 because they are sign extensions.
If register setting is positive and $+134,217,727$, you can read 07FFFFFFh.
If register setting is negative and $-134,217,728$, you can read F8000000h.
Sign extension writes are ignored.
For example, F8000000h is set whether the value to write is 08000000 h or F 8000000 h .

### 5.4.4.2 RCUN2: Counter 2 (General-purpose 1)


$\square$ RCUN2

Control commands: RRCUN2(E4h), WRCUN2(A4h)
Register to obtain the count value of counter 2 (General-purpose 1).
With RENV3.Cl2 bit, you can select the count from the following three types.

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

The setting range is-134,217,728 to $+134,217,727$.

The \#s in bits 31 to 28 are the same value as bit 27 because they are sign extensions.
If register setting is positive and $+134,217,727$, you can read 07FFFFFFh.
If register setting is negative and-134,217,728, you can read F8000000h.
Sign extension writes are ignored.
For example, F8000000h is set whether the value to write is 08000000 h or F 8000000 h .

### 5.4.4.3 RCUN3: Counter 3 (Deviation)

 | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ | $\#$ |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| RCUN3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Control commands: RRCUN3(E5h), WRCUN3(A5h)
Register to obtain 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 |
| :--- | :--- |
| 00 b | Deviation between command pulse signals and encoder signals (EA,EB) |
| 01 b | Deviation between command pulse signals and manual pulser signals (PA,PB) |
| 10 b | Deviation between encoder signals (EA,EB) and manual pulser signals (PA,PB) |

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

The \# in bits 31 to 16 is the same value as bit 15 because they are sign extensions.
If register setting is positive and $+32,767$, you can read 00007 FFFh .
If register setting is-32,768 in negative, you can read FFFF8000h.
Sign extension writes are ignored.
For example, FFFF8000h is set whether the writing is 00008000 h or FFFF8000h.

### 5.4.4.4 RCUN4: Counter 4 (General-purpose 2)

$\begin{array}{llllllllllllllllll}31 & 30 & 29 & 28 & 27 & 26 & 25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 \\ 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0\end{array}$
$\square$

## RCUN4

Control commands: RRCUN4(E6h), WRCUN4(A6h)
Register to obtain the count value of counter 4 (General-purpose 2).
With RENV3.Cl4 bit, you can select the count from the following four types.

| RENV3.CI4 | Count target |
| :---: | :--- |
| 00 b | Command pulse signals |
| 01 b | Encoder signals (EA, EB) |
| 10 b | Manual pulser signals (PA, PB) |
| 11 b | $\frac{f_{C L K}}{2}$ signal |

The setting range is $-134,217,728$ to $+134,217,727$.

The \# in bits 31 to 28 is the same value as bit 27 because they are sign extensions.
If register setting is positive and $+134,217,727$, you can read 07FFFFFFh.
If register setting is negative and-134,217,728, you can read F8000000h.
Sign extension writes are ignored.
For example, F8000000h is set whether the value to write is 08000000 h or F8000000h.

### 5.4.5 Comparator register

This is the register for comparators.
For the comparator, see "6.13 Comparator".

The \# in bits 31 to 28 of each register is the same as bit 27 because they are sign extensions.
If register setting is positive and $+134,217,727$, you can read 07FFFFFFh.
If register setting is negative and-134,217,728, you can read F8000000h.
Sign extension writes are ignored.
For example, F8000000h is set whether the value to write is 08000000 h or F8000000h.

### 5.4.5.1 RCMP1: Comparator 1 comparison value




Control commands: RRCMP1(E7h), WRCMP1(A7h)
This register sets comparator 1 compare value.
The setting range is-134,217,728 to $+134,217,727$.

### 5.4.5.2 RCMP2: Comparator 2 comparison value




Control commands: RRCMP2(E8h), WRCMP2(A8h)
Register to set the comparison value in Comparator 2.
The setting range is $-134,217,728$ to $+134,217,727$.

### 5.4.5.3 RCMP3: Comparator 3 comparison value


$\square$
Control commands: RRCMP3(E9h), WRCMP3(A9h)
Register to set the comparison value in Comparator 3.
The setting range is $-134,217,728$ to $+134,217,727$.

### 5.4.5.4 RCMP4: Comparator 4 comparison value

$\left.\begin{array}{lllllllllllllllllllllllll}31 & 30 & 29 & 28 & 27 & 26 & 25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7\end{array}\right) 6$
$\square$
Control commands : RRCMP4(EAh), WRCMP4(AAh)
Register to set the comparison value in Comparator 4.
The setting range is $-134,217,728$ to $+134,217,727$.

### 5.4.5.5 RCMP5(PRCP5): Comparator 5 comparison value

 | $\#$ | $\#$ | $\#$ | $\#$ |
| :--- | :--- | :--- | :--- |
| RCMP5(PRCP5) |  |  |  |

Control commands : RRCMP5(EBh), RPRCP5(CBh), WRCMP5(ABh), PRCP5(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 $-134,217,728$ to $+134,217,727$.

### 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




Control command: RRLTC1(EDh)
Register to obtain the latch data of counter 1 (command position).
The data range is $-134,217,728$ to $+134,217,727$.

The \# in bits 31 to 28 is the same value as bit 27 because it is a sign extension.
If register setting is positive and $+134,217,727$, you can read 07FFFFFFh.
If register setting is negative and $-134,217,728$, you can read F8000000h.
Sign extension writes are ignored.
For example, F8000000h is set whether the value to write is 08000000h or F8000000h.

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

3130292827262524232221201918171615141312111096 \# |  | $\#$ | $\#$ |  |
| :--- | :--- | :--- | :--- |
| RLTC2 |  |  |  |

Control commands : RRLTC2(EEh)

Register to obtain the latch data in counter 2 (General-purpose 1).
The data range is $-134,217,728$ to $+134,217,727$.

The \# in bits 31 to 28 is the same value as bit 27 because they are sign extensions.
If register setting is positive and $+134,217,727,07 F F F F F F h$ can be read.
If register setting is negative and-134,217,728, F 8000000 h can be read.
Sign extension writes are ignored.
For example, F8000000h is set whether the value to write is 08000000 h or F8000000h.

### 5.4.6.3 RLTC3: Counter 3 (Deviation) latch




Control command: RRLTC3(EFh)
Counter 3 (Deviation), or the register to obtain the latch data of the current speed step number.
When latching counter 3 (RENV5.LTFD = 0), data can be $-32,768$ to $+32,767$.
When latching the present-speed steps (RENV5.LTFD $=1$ ), data is 0 to 65535 .

Bits 31 to 16 \# are the same as bit 15 when RENV5.LTFD $=0$ because they are sign-extended.
If register setting is positive and $+32,767$, you can read 00007 FFFh.
If register setting is-32,768 in negative, you can read FFFF8000h.
Sign extension writes are ignored.
For example, FFFF8000h is set whether the writing is 00008000 h or FFFF8000h.

### 5.4.6.4 RTC4: Counter 4 (General-purpose 2) latch


$\square$ RLTC4

Control commands : RRLTC4(F0h)
Register to obtain the latch data of counter 4 (General-purpose 2).
The data range is $-134,217,728$ to $+134,217,727$.

The \# in bits 31 to 28 is the same value as bit 27 because it is a sign extension.
If register setting is positive and $+134,217,727$, you can read 07FFFFFFh.
If register setting is negative and-134,217,728, you can read F8000000h.
Sign extension writes are ignored.
For example, F8000000h is set whether the value to write is 08000000 h or F8000000h.

### 5.4.7 Interrupt 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 |

Control commands : 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 $2 n d$ 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.SPDF 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 met. (MSTS.SCP1 changed from 0 to 1 ) |
| 9 | IRC2 | 1: An interrupt is generated when the comparison condition of comparator 2 is met. (MSTS.SCP2 changed from 0 to 1 ) |
| 10 | IRC3 | 1: An interrupt is generated when the comparison condition of comparator 3 is met. (MSTS.SCP3 changed from 0 to 1) |
| 11 | IRC4 | 1: An interrupt is generated when the comparison condition of comparator 4 is met. (MSTS.SCP4 changed from 0 to 1 ) |
| 12 | IRC5 | 1: An interrupt is generated when the comparison condition of comparator 5 is met. <br> (MSTS.SCP5 changed from 0 to 1 ) |
| 13 | IRCL | 1: An interrupt occurs when CLR signal is turned ON and the count value is cleared. <br> If there is no counter to clear (RENV3.CU1C to CU4C $=0000 \mathrm{~b}$ ), no interrupt is generated. |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 14 | IRLT | 1: An interrupt occurs when LTC signal is turned $O N$ and the count value is latched. If the latching timing is not a LTC signal. (RENV5.LTM $\neq 00 \mathrm{~b}$ ), no interrupt is generated. |
| 15 | IROL | 1: An interrupt occurs when the ORG signal is turned ON and the count value is latched. If the latching timing is not ORG signal (RENV5.LTM $\neq 01 \mathrm{~b}$ ), no interrupt is generated. |
| 16 | IRSD | 1: An interrupt occurs when SD signal turns ON. |
| 17 | IRDR | 1: An interrupt occurs whether $+D R$ signal or -DR signal changes. When PEn $=\mathrm{H}$ level, no interrupt is generated. |
| 18 | IRSA | 1: An interrupt is generated when CSTA signal is turned ON (RENV1.PCSM $=0$ ). <br> 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 |

Control commands: RREST(F2h), WREST(B2h)
Register to retrieve the error interrupt factor.
For details on the error interrupt, see "6.18.1 Error interrupt."

| Bit | Name | Description |
| :---: | :---: | :--- |
| 0 | ESC1 | 1: An abnormal stop occurred when the comparison condition of Comparator 1 is met. <br> (Including stop by +SL) |
| 1 | ESC2 | 1: An abnormal stop occurred when the comparison condition of Comparator 2 is met. <br> (Including stop by -SL) |
| 2 | ESC3 | 1: An abnormal stop occurred when the comparison condition of Comparator 3 is met. |
| 3 | ESC4 | 1: An abnormal stop occurred when the comparison condition of Comparator 4 is met. |
| 4 | ESC5 | 1: An abnormal stop occurred when the comparison condition of Comparator 5 is met. |
| 5 | ESPL | 1: Stopped abnormally when +EL signal turns ON. |
| 6 | ESML | 1: Stopped abnormally when -EL signal turns ON. |
| 7 | ESAL | 1: Stopped abnormally when ALM signal turns ON. |
| 8 | ESSP | 1: Stopped abnormally when CSTP signal turns ON. |
| 9 | ESEM | 1: Stopped abnormally when CEMG signal turns ON. |
| 10 | ESSD | 1: Stopped abnormally when RENV1.SDM = 1 and SD signal in the operating direction turns ON. |
| 11 | 0 | Always obtain 0. |
| 12 | ESDT | 1: Stopped abnormally when an error occurs in the interpolation setting data. |
| 13 | ESIP | 1: Stopped abnormally when the interpolation axis other than own axis stopped abnormally during an <br> interpolation operation. |
| 14 | ESPO | 1: Stops abnormally when the buffer counter (16 bit) for inputting PA and PB signals overflowed. |

* 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 $=60 \mathrm{~h}, 61 \mathrm{~h}, 68 \mathrm{~h}, 69 \mathrm{~h}$ ) is not greater than two axes.
(2) In the operation mode of linear interpolation 2 control (RMD.MOD $=62 \mathrm{~h}, 63 \mathrm{~h}, 6 \mathrm{Ah}, 6 \mathrm{Bh}$ ), the feed amount of the main axis (RIP) is not set (RIP register of the main axis is set to 0 ).
(3) Circular interpolation control operation mode (RMD.MOD $=64 \mathrm{~h}, 65 \mathrm{~h}, 66 \mathrm{~h}, 67 \mathrm{~h}, 6 \mathrm{Ch}, 6 \mathrm{Dh}$ ) is not two axis.
(4) The center position is not set in the operation mode of circular interpolation control.
(Both axis are set to RIP register 0)
(5) U-axis does not operate in the U-axis interpolation control operation mode (RMD.MOD $=66 \mathrm{~h}, 67 \mathrm{~h}$ ).

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 |

Control commands: RRIST(F3h), WRIST(B3h)
Register to obtain the event interrupt factor. For details on the Event interrupt, see "6.18.2 Event interrupt".

| Bit | Name | Description |
| :---: | :---: | :---: |
| 0 | ISEN | 1: The operation mode stopped normally. |
| 1 | ISN | 1: Pre-register was shifted because pre-register was fixed (RSTS.PFM $>0$ ) when the operation stopped. |
| 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.SPDF bit changed from 1 to 0 ) |
| 4 | ISUS | 1: Acceleration started. <br> (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 is met. <br> (MSTS.SCP1 changed from 0 to 1 ) |
| 9 | ISC2 | 1: The comparison condition of comparator 2 is met. <br> (MSTS.SCP2 changed from 0 to 1 ) |
| 10 | ISC3 | 1: The comparison condition of Comparator 3 is met. <br> (MSTS.SCP3 changed from 0 to 1 ) |
| 11 | ISC4 | 1: The comparison condition of Comparator 4 is met.\} <br> (MSTS.SCP4 changed from 0 to 1) |
| 12 | ISC5 | 1: The comparison condition of Comparator 5 is met. <br> (MSTS.SCP5 changed from 0 to 1) |
| 13 | ISCL | 1: CLR signal is turned ON to clear the count value. <br> If the latching timing is not a LTC signal. (RENV3.CU1C to CU4C = 0000b), no interrupt is generated. |
| 14 | ISLT | 1: LTC signal is turned ON to latch the count value. <br> If the latching timing is not a LTC signal. (RENV5.LTM $\neq 00 \mathrm{~b}$ ), no interrupt is generated. |
| 15 | ISOL | 1: ORG signal is turned ON to latch the count value. <br> If there is no ORG signal at the latching timing (RENV5.LTM $\neq 01 \mathrm{~b}$ ), no interrupt is generated. |
| 16 | ISSD | 1: SD signal is turned ON. |


| Bit | Name | Description |
| :---: | :---: | :--- |
| 17 | ISPD | $1:+$ DR signal is changed. <br> When PEn $=\mathrm{H}$ level, no interrupt is generated. |
| 18 | ISMD | $1:-$ DR signal is changed. <br> When PEn = H level, no interrupt is generated. |
| 19 | ISSA | $1:$ CSTA pin is enabled (RENV1.PCSM $=0$ ) and CSTA signal turns ON. <br> Or CSTA pin is disabled (RENV1.PCSM $=1$ ) and STA signal turns ON. |
| $31: 20$ | 0 | Always obtains 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 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| SDIN | 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 | 0 | 0 | PFM | PFC | 0 | SINP |  |  |

Control commands : RRSTS(F1h)
Register to obtain the operation mode and the status of various signals.

| Bit | Name | Description |
| :---: | :---: | :---: |
| 3:0 | CND | Indicates the operating status. <br> 0000b: Stopping <br> 0001b: Waiting for +DR, -DR signal input <br> 0010b: Waiting for CSTA and STA signal inputs <br> 0011b: Waiting for input of internal synchronization signal <br> 0100b: Waiting for the other axis to stop <br> 0101b: Wait for ERC signal ON duration and completion OFF delay <br> 0110b: Waiting for the direction change timer to complete <br> 0111b: In backlash correction <br> 1000b: Waiting for input of PA and PB signals 1001b: In FA constant speed operation <br> 1010b: In FL constant speed operation 1011b: Accelerating <br> 1100b: In FH constant speed operation 1101b: Decelerating <br> 1110b: Waiting for input of INP signal 1111b: Other (during start control) <br> The other status (RSTS.CND = 1111b) shifts to another status by inputting the CLK signal several times. |
| 4 | SDIR | Indicates the direction of an operation. <br> 0 : +Direction <br> 1: -Direction |
| 5 | SSTA | Indicates the input status of CSTA signal or STA signal. $\begin{aligned} & 0: \text { OFF } \\ & 1: \text { ON } \end{aligned}$ <br> The input logic of CSTA signal is negative. <br> The input logic of STA signal is selected by RENV1.PCSL bit. |
| 6 | SSTP | Indicates the input status of CSTP signal. <br> 0 : OFF <br> 1: ON <br> The input logic of CSTP signal is negative. |


| Bit | Name | Description |
| :---: | :---: | :---: |
| 7 | SEMG | Indicates the input status of CEMG signal. $\begin{aligned} & \text { 0: OFF } \\ & \text { 1: ON } \end{aligned}$ <br> The input logic of CEMG signal is negative. |
| 8 | SPCS | Indicates the status of a signal input to PCSn pin. $\begin{aligned} & \text { 0: OFF } \\ & \text { 1: ON } \end{aligned}$ <br> The logic of a signal input to PCSn pin is selected by RENV1.PCSL bit. |
| 9 | SERC | Indicates the output status of ERC signal. $\begin{aligned} & \text { 0: OFF } \\ & \text { 1: ON } \end{aligned}$ <br> The logic of an ERC signal is selected by RENV1.ERCL bit. |
| 10 | SEZ | Indicates the input status of EZ signal. $\begin{aligned} & \text { 0: OFF } \\ & \text { 1: ON } \end{aligned}$ <br> The input logic of an EZ signal is selected by RENV2.EZL bit. |
| 11 | SDRP | Indicates the input status of $+D R$ signal. <br> 0 : OFF <br> 1: ON <br> The input logic of a +DR signa is selected by RENV1.DRL bit. <br> RENV1.DRL bit is the same as -DR. <br> RSTS. SDRP bit changes when PEn $=\mathrm{H}$. |
| 12 | SDRM | Indicates the input status of -DR signal. <br> 0 : OFF <br> 1: ON <br> The input logic of a -DR signal is selected by RENV1.DRL bit. <br> RSTS. SDRM setting also changes when $\mathrm{PEn}=\mathrm{H}$. |
| 13 | SCLR | Indicates the input status of CLR signal. $\begin{aligned} & \text { 0: OFF } \\ & \text { 1: ON } \end{aligned}$ <br> The input logic of an CLR signal is selected by RENV1.CLRL bit. |
| 14 | SLTC | Indicates the input status of LTC signal. <br> 0 : OFF <br> 1: ON <br> The input logic of an LTC signal is selected by RENV1.LTCL bit. |


| Bit | Name | Description |  |  |
| :---: | :---: | :---: | :---: | :---: |
| 15 | PSDI | Indicates the input status of SD signal. <br> 0 : OFF <br> 1: ON <br> The input logics of SD signals are selected by RENV1.SDL bit. |  |  |
| 16 | SINP | Indicates the input status of INP signal. $\begin{aligned} & \text { 0: OFF } \\ & \text { 1: ON } \end{aligned}$ <br> The input logic of an INP signal is selected by RENV1.INPL bit. |  |  |
| 17 | MSDI | 0 is always obtained. |  |  |
| 19,18 | PFC | Indicates the determined sta | 1st pre-register Undetermined Undetermined Determined Determined | a continuous co <br> Current register <br> (RCMP5) <br> Undetermined <br> Determined <br> Determined <br> Determined |
| 21,20 | PFM | Indicates the determined sta | of the pre-regist <br> 1st pre-register <br> Undetermined <br> Undetermined <br> Determined <br> Determined | a continuous op <br> Current register <br> Undetermined <br> Determined <br> Determined <br> Determined |
| 31:22 | 0 | 0 is always obtained. |  |  |

### 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 | SED |  | SDM |  | IPCC | IPCW | IPE | IPL |

Control commands: RRIPS(FFh)
This register is to obtain the status of various interpolation controls.
Interpolation status is shared by all axis in common.
The same content can be read regardless of axis-selection.

| 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 | IFPx | 1: Constant synthesized speed operation is set in X -axis |
| 13 | IFPy | 1: Constant synthesized speed operation is set in Y -axis |
| 14 | IFPz | 1: Constant synthesized speed operation is set in Z-axis |
| 15 | IFPu | 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 <br> (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, $\quad 01: 2$ nd quadrant, $\quad$ 10: 3rd quadrant, $\quad$ 11: 4th quadrant) |
| 31: 24 | 0 | 0 is always obtained. |

### 5.5 Operation mode

There are 45 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> <br> 0000000 (00h): Operation mode of continuous movement in +direction in command control 0001000 ( 08 h ): Operation mode of continuous movement in -direction in command control 0000001 (01h): Operation mode of continuous movement in pulser control 0000010 (02h): Operation mode of continuous movement in switch control 0010000 (10h): Operation mode of origin return in +direction in origin return control 0011000 (18h): Operation mode of origin return in -direction in origin return control 0010010 (12h): Operation mode to escape from origin in +direction in origin return control 0011010 (1Ah): Operation mode to escape from origin in -direction in origin return control 0010101 (15h): Operation mode of origin search in +direction in origin return control 0011101 (1Dh): Operation mode of origin search in -direction in origin return control 0100000 (20h): Operation mode to move up to +EL or +SL in sensor control 0101000 (28h): Operation mode to move up to -EL or -SL in sensor control 0100010 (22h): Operation mode to escape from -EL or -SL in sensor control 0101010 (2Ah): Operation mode to escape from +EL or +SL in sensor control 0100100 (24h): Operation mode of movement for EZ count in +direction in sensor control 0101100 (2Ch): Operation mode of movement for EZ count in -direction in sensor control 100 0001(41h): Operation mode of incremental movement in positioning control 1000010 (42h): Operation mode in which the absolute position is specified in counter 1 in positioning control 1000011 (43h): Operation mode in which the absolute position is specified in counter 2 in positioning control 1000100 (44h): Operation mode to return to zero point with counter 1 in positioning control 1000101 (45h): Operation mode to return to zero point with counter 2 in positioning control 1000110 (46h): Operation mode of one pulse in +direction in positioning control 1001110 (4Eh): Operation mode of one pulse in -direction in positioning control 1000111 (47h): Timer operation mode in positioning control 1010001 (51h): Operation mode of incremental movement in pulser control $1010010(52 \mathrm{~h})$ : Operation mode in which the absolute position is specified in counter 1 in pulser control 1010011 (53h): Operation mode in which the absolute position is specified in counter 2 in pulser control 1010100 (54h): Operation mode to return to zero point with counter 1 in pulser control 1010101 (55h): Operation mode to return to zero point with counter 2 in pulser control 1010110 (56h): Operation mode of incremental movement in switch control 1100000 (60h): Operation mode of continuous movement in linear interpolation 1 control 1100001 (61h): Operation mode of incremental movement in linear interpolation 1 control 1100010 ( 62 h ): Operation mode of continuous movement in linear interpolation 2 control 1100011 (63h): Operation mode of incremental movement in linear interpolation 2 control 1100100 (64h): Operation mode of circular interpolation in CW direction in circular interpolation control 1100101 (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 synchronous control 1100111 (67h): Operation mode of circular interpolation in CCW direction in U-axis synchronous control 1101000 (68h): Operation mode of continuous movement with linear interpolation 1 in pulser control 1101001 (69h): Operation mode of incremental movement with linear interpolation 1 in pulser control 1101010 (6Ah): Operation mode of continuous movement with linear interpolation 2 in pulser control 1101011 (6Bh): Operation mode of incremental movement with linear interpolation 2 in pulser control 1101100 (6Ch): Operation mode of circular interpolation in CW direction in pulser control 1101101 (6Dh): Operation mode of circular interpolation in CCW direction in pulser control 1101111 (6Fh): Dummy operation mode with circular interpolation control Do not set any other values. | RMD.MOD(6:0) |

### 5.5.1 Command control

This control method is to stop with the 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 stops, 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 stops, 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 RMV register setting value is changed, RPLS register is updated by the absolute value of RPLS register. RPLS register counts down for each command pulse output.

If the speed pattern is high-speed 1 or high-speed 2 , decelerating starts when it becomes RPLS < RSDC. When RPLS $=0$, the command pulse stops.

You can stop halfway with the stop command.

### 5.5.2.1 Incremental movement (41h)

When starting, the RPLS register 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 $<0$.
When the command pulse stops, the operation mode is completed.

When starting with $\mathrm{RMV}=0(\mathrm{RPLS}=0)$, the operation mode completes without outputting the command pulse.

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

When starting, the RPLS register is updated with the absolute value of the difference between RCUN1 and RMV register. When started, the command pulses are output in +direction if RMV > RCUN1 and output in -direction if RMV < RCUN1. When the 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 $=42 \mathrm{~h}$, except that RCUN2 register is used instead of RCUN1 register.

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

With RMD.MOD $=42 \mathrm{~h}$, the operation is the same as when $\mathrm{RMV}=0$ is set.
Other than RMV $=0$, the RPLS register is also updated with $\mathrm{RMV}=0$.

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

With RMD.MOD $=43 \mathrm{~h}$, the operation is the same as when $\mathrm{RMV}=0$ is set. Other than RMV $=0$, the RPLS register is also updated with $\mathrm{RMV}=0$.

### 5.5.2.6 One pulse in plus direction (46h)

With RMD.MOD $=41 \mathrm{~h}$, the operation is the same as when $\mathrm{RMV}=1$ is set.
Other than RMV $=1$, the RPLS register is also updated with $R M V=1$.

### 5.5.2.7 One pulse in minus direction (4Eh)

With RMD.MOD $=41 \mathrm{~h}$, the operation is the same as when $\mathrm{RMV}=-1$ is set.
Other than RMV $=-1$, the RPLS register is also updated as $R M V=-1$.

### 5.5.2.8 Timer (47h)

When starting, the RPLS register 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.

Set the RMV register from 1 to $134,217,727$.
When starting with $\mathrm{RMV}=0($ 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. (For example, if you set 120 pulses at 1000 pps, the operation will pause for 120 ms )

The operation does not stop by neither +EL, -EL, SD signal inputs nor software limit. It stops by 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 can be controlled in synchronization with the inputs of PA and PB signals.

It can be used when the input pin of $P A, P B$ signal is enabled ( $P E n=L$ ) and also the input function of $P A, P B$ signal is enabled (RENV2.POFF = 0).

When PEn pin is used, multiple axes can be controlled by switching with one set of manual pulser.
Because PEn pin has a built-in pull-up resistor, inputting PA and PB signals is disabled when the pin is open.
The addition of an external pull-up resistor is recommended for improved noise-immunity.


The input noise filter of PE signal can be set with RENV1.DRF.
The input noise filter of PA, PB signal can be set with RENV2.PINF.

When started, "PA, PB Waiting for inputting" (RSTS.CND $=1000$ b) is established.
Then, command pulse signal is output in synchronization with PA signal and PB signal.
Use FH constant speed for velocity patterning.

When PA, PB is reversed even in pulser control, backlash correction function operates.
Note that this cannot be handled if PA, PB is turned back to "backlash correction" (RSTS.CND = 0111b).

PA signal and PB signal can be selected from 4 types by RENV2.PIM.
-90-degree phase difference mode $1 x$
-90-degree phase difference mode $2 x$

- 90-degree phase difference mode $4 x$
- 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 $4 x, 32$ 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 $1 \times($ RENV2.PIM $=00 b)$

2. 90-degree phase difference mode $2 x$ (RENV2.PIM $=01 \mathrm{~b}$ )

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

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


When RENV6.PMG $=2(3 x)$ is set.


When RENV6.PD = 512 (512/2048 division) is set.


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 / PB signals and the output timing of command pulses will have a tolerance of the internal pulse cycle at the longest.
<Calculating the Maximum Input-Frequency (FP)>
Since a pulser is rotated manually, incoming frequencies of PA and PB signals are not constant.
The maximum input frequency (FP) is limited by FH speed, input specifications, multiplication setting, and division setting.

Set FP higher than required.
Increasing FH speed to take into account FP results in a narrower power pulse width.
FH speed can be set to motor driver input speed upper limit.
However, when the multiplication function is used with a stepping motor, it must be lower than starting frequency.

When the input frequency exceeds the FH speed, it will be buffered by the input buffer counter (signed 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 $\neq 0$

$$
F P<F H \div P I M G \div(\text { RENV6.PMG }+1) \div(\text { RENV6.PD } \div 2048)
$$

2) When RENV6.PD $=0$

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

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

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

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

| RENV2.PIM | RENV6.PMG | RENV6.PD | Calculation r |  |
| :---: | :---: | :---: | :---: | :---: |
| 00b (90-degree phase difference mode 1x) | $0(1 \mathrm{x})$ | 0 | $F P<F H \div 1 \div 1$ | $=F H$ |
|  | 0(1x) | 1024 | $F P<F H \div 1 \div 1 \div \frac{1}{2}$ | $=F H \times 2$ |
|  | 2(3x) | 0 | $F P<F H \div 1 \div 3$ | $=F H \div 3$ |
| 01b (90-degree phase difference mode 2 x ) | 0(1x) | 0 | $F P<F H \div 2 \div 1$ | $=F H \div 2$ |
|  | 0(1x) | 1024 | $F P<F H \div 2 \div 1 \div \frac{1}{2}$ | $=F H$ |
|  | 2(3x) | 0 | $F P<F H \div 2 \div 3$ | $=F H \div 6$ |
| 10b (90-degree phase difference mode 4 x ) | 0(1x) | 0 | $F P<F H \div 4 \div 1$ | $=F H \div 4$ |
|  | 0(1x) | 1024 | $F P<F H \div 4 \div 1 \div \frac{1}{2}$ | $=F H \div 2$ |
|  | 2(3x) | 0 | $F P<F H \div 4 \div 3$ | $=F H \div 12$ |
| 11b (2-pulse mode) | 0(1x) | 0 | $F P<F H \div 1 \div 1$ | $=F H$ |
|  | 0(1x) | 1024 | $F P<F H \div 1 \div 1 \div \frac{1}{2}$ | $=F H \times 2$ |
|  | 2(3x) | 0 | $F P<F H \div 1 \div 3$ | $=F H \div 3$ |

If the input frequencies of PA signal and PB signal are not constant, the shortest cycle will become FP.


When stopping immediately with STOP (49h) command, the total output pulse does not necessarily the integral multiple of a multiplication value. For operation mode (RMD.MOD $=01 \mathrm{~h}$ ) in pulser control, the following can be selected:

- Set RENV6.PSTP $=0$ to stop the operation prior to the total output pulse being an integral multiple of the multiplication value.
- Set RENV6.PSTP = 1 to delay the operation until the total output pulse becomes an integral multiple of the multiplication value.

If pulser control is set to another operation mode, even if RENV6.PSTP is set to 1 , operation stops immediately with STOP(49h) command.

| Name and description | Target |
| :---: | :---: |
| <PE Noise-in filter> <br> 0 : Signals with a pulse width of $0.05 \mu \mathrm{~s}$ or greater are reliably reacted. <br> 1: Pulse width completely ignores any signals below 26 ms . | RENV1.DRF(27) |
| <PA, PB Noise-in filter> <br> 0 : Signals with a pulse width of $0.05 \mu \mathrm{~s}$ or greater are reliably reacted. <br> 1: Pulse width completely ignores signals of $0.10 \mu \mathrm{~s}$ or less. Signals with a pulse width of $0.15 \mu$ s or greater are reliably reacted. | RENV2.PINF(19) |
| <PA Signal and PB Signal Settings> <br> 00b: 90-degree phase difference mode 1 x . <br> 01b: 90-degree phase difference mode multiplied by 2. <br> 10b: 90-degree phase difference mode multiplied by 4. <br> 11b: two-pulse mode 。 | RENV2.PIM(25:24) |
| <Counting Direction of PA Signal and PB Signal> <br> 0 : Count up when PA signal-input phase is advanced. <br> 1: Count up when $P B$ signal-input phase is advanced. | RENV2.PDIR(26) |
| <PA Signal and PB Signal Input Function> <br> 0: Enabled <br> 1: Disabled. No input errors are detected. | RENV2.POFF(31) |
| <Processing when writing a stop-command in pulser control> <br> 0 : The input PA, PB is ignored, and the operation is stopped. <br> 1: Outputs command pulse corresponding to PA, PB input, and then stops. <br> For interpolating control ( $68 \mathrm{~h}, 69 \mathrm{~h}, 6 \mathrm{Ah}, 6 \mathrm{Bh}, 6 \mathrm{Ch}$ and 6 Dh ), the operation will stop ignoring RENV6.PSTP $=1$. | RENV6.PSTP(15) |


| Name and description | Target |
| :--- | :--- |
| <Numerator of PA signal and PB signal Input> | RENV6.PD(26:16) |
| $0:$ Do not divide. |  |
| 1 to 2047: Divide by the setting value/2048. | RENV6.PMG(31:27) |
| <Multiplication of PA and PB Input>   <br> 0 to 31: Multiplies the setting value by adding 1. RSTS.CND(3:0)  <br> <Operation status>   <br> < Error interrupt factor (ESPE)> REST.ESPE(17)  <br> 1: PA and PB signal input error occurred. Operation mode does not stop. REST.ESPO(14)  <br> < Error interrupt factor (ESPO)>   |  |

When counting plus direction, the operation stops by + EL signal ON.
When counting minus direction, the operation stops by -EL signal ON.
When RENV5.PDSM $=0$, EL signal is stopped by $O N$ in the operating direction. The error interrupt is not generated, and operation mode continues. If this happens, inputting a backward PA or PB signal can escape from EL position value of the stopping factor. When RENV5.PDSM $=1$, EL signal is stopped by ON , an erroneous interrupt is generated, and operation mode is completed.

### 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 ( $R S T S . C N D=1000 \mathrm{~b}$ ), a command pulse will start to be output.

The counting direction is determined by PA and PB signals as well as RENV2.PDIR bit.

| RENV2.PIM | RENV2.PDIR | PA and PB signals | Count direction |
| :---: | :---: | :---: | :---: |
| 00b, 01b, 10b: 90-degree phase difference mode | 0 | PA input phase advances. | + direction |
|  |  | PB input phase advances. | - direction |
|  | 1 | PB input phase advances. | + direction |
|  |  | PA input phase advances. | - 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 |

When the stop command is written, the operation is completed.

### 5.5.3.2 Incremental movement (51h)

Positioning control is performed in synchronization with inputting PA and PB signals. Upon start, RPLS register is updated with the absolute value of RMV register.

When PA or PB signal is input while waiting to be input for PA and PB signals (RSTS.CND $=1000 \mathrm{~b}$ ), command pulse starts outputting. The counting direction is determined as plus direction if RMV $>0$ and as minus direction if RMV $<0$.

PA and PB signals and RENV2.PDIR do not affect the counting direction.
When RPLS $=0$, the command pulse is stopped.
When the command pulse is stopped, operation mode is completed.
After operation mode is completed, command pulse signal is not output if PA and PB signals are input.

If you attempt a start by $\mathrm{RMV}=0$, operation mode is completed without outputting command pulse.
Operation mode can also be completed by executing the 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 RMV register absolute value.
If a PA or PB signal is input while waiting for PA or PB signal input ( $R S T S . C N D=1000 \mathrm{~b}$ ), 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.
Operation mode can also be completed by executing the 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 ( 54 h )

In RMD.MOD $=52 \mathrm{~h}$, the operation is the same as when $\mathrm{RMV}=0$ is set.
Even in other than RMV $=0$, RPLS register is updated with $\mathrm{RMV}=0$.

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

With RMD.MOD $=53 \mathrm{~h}$, the operation is the same as when $\mathrm{RMV}=0$ is set.
Even in other than RMV $=0$, RPLS register is updated with $\mathrm{RMV}=0$.

### 5.5.3.7 Continuous movement by linear interpolation 1 ( 68 h )

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 $=1000 \mathrm{~b}$ ), 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 $P A$ and $P B$ signal inputs.
If a PA or PB signal is input while waiting for PA or PB signal input (RSTS.CND $=1000 \mathrm{~b}$ ), the command pulse will start to be output. For the continuous 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 $=1000$ b), 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 $=1000$ b), 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 ( $R S T S . C N D=1000 b$ ), the command pulse will start to be output. For the circular interpolation in CW direction, see "5.5.9.1 Circular interpolation in CW (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 $=1000$ b), 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 the trigger.

This control can be used when PEn = L level.
PEn pin allows you to switch between multiple axes with a set of drive switches.
The connection method is the same as in a pulser control.
For details, see "5.5.3 Pulser control".

Input status of + DR signal can be checked with RSTS.SDRP bit.
Input status of - DR signal can be checked with RSTS.SDRM bit.
Input logics of + DR and - DR signals can be set with RENV1.DRL bit.
Input noise-filter of PE signal, + DR signal, and - DR signal can be set using RENV1.DRF bit.
When + DR signal and - DR signal change, the interrupt of RIST register can be set using RIRQ.IRDR bis.
Interrupt to RIST register is made by RIST.ISPD $=1$ when the + DR signal changes and RIST.ISMD $=1$ when-DR signal changes.

When the unit is started, the unit enters RSTS.CND $=0001 \mathrm{~b}$ (+DR, -DR signal-input wait status).
Then, 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 pins input logic> <br> 0 : Negative logic <br> 1: Positive logic | RENV1.DRL(25) |
| < Input noise-filter for +DR, -DR, PE pins > <br> 0 : A signal with a pulse width of $0.05 \mu$ s or more will react reliably. <br> 1: A signal with a pulse width of 26 ms or less is ignored completely. | RENV1.DRF(27) |
| <Event interrupt factor (ISPD)> <br> 1: +DR signal has changed. | RIST.ISPD(17) |
| <Event interrupt factor (ISMD)> <br> 1: -DR signal has changed. | RIST.ISMD(18) |
| <Operating status> 0001b: Waiting for +DR, -DR signal input | RSTS.CND(3: 0) |
| <Signal status (SDRP)> <br> 0 : +DR signal OFF <br> 1: +DR signal ON | RSTS.SDRP(11) |
| < Signal status (SDRM)> <br> 0 : -DR signal OFF <br> 1: -DR signal ON | RSTS.SDRM(12) |

When counting in positive direction, the operation stops by $+E L$ signal $O N$.
When counting in negative direction, the operation stops by - EL signal ON.
When RENV5.PDSM $=0$, operation mode continues. The error interrupt is not generated when EL signal is stopped by ON in the operating direction. In such cases, you can escape from EL position value of the stopping factor by inputting a $+D R$ or $-D R$ signal in the reverse direction.

When RENV5.PDSM = 1, EL signal is stopped by ON, an error interrupt is generated, and operation mode is completed.

### 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.

### 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 134,217727 .
When +DR signal or -DR signal is ON while stopped, RPLS register is updated with the RMV register.
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 stops, and the operation mode can continue.
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 to complete 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 $+D R$ or $-D R$ signal is enabled only while waiting for the input of $+D R$ signal or $-D R$ signal (RSTS.CND = 0001b). +DR signal operation example:

(1) Because they are edge triggers, not operate even if the operation mode is started with $+D R$ signal $O N$.
(2) Because of an incremental movement, the command pulse is stopped at RPLS $=0$ even if the + DR signal is kept ON .
(3) Because of an incremental movement, even if + DR signal OFF is input, the command pulse is output until RPLS $=0$.
(4) Because RSTS.CND $\neq 00001 \mathrm{~b}$ during operation, inputting +DR signal ON does not affect the operation.
(5) Because RSTS.CND $\neq 00001$ b during operation, inputting -DR signal ON does not affect the operation.

When EL signal in the operating direction is stopped by ON, DR signal ON in the reverse direction is enabled.

+ Example of EL signal operation:

(1) When the operation mode starts, 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, it does not operate if +DR signal is ON.
(4) While +EL signal is ON, it operates in -direction if -DR signa is ON.
(5) When +EL signal is OFF, it operates in +direction if +DR signal is ON.


### 5.5.5 Origin return control

This control method is to stop at the origin.
Also see "6.8.2 Deviation counter clear (ERC)" to control a servo motor.

Input status of ORG signal can be checked with SSTS.SORG bit. Input logic of ORG signal can be set with RENV1.ORGL bit.

ORG signal noises filter can be set with RENV1.FLTR bit.
When ORG signal is used, SSTS.SORG bit is sampled at the timing when command pulse changes from ON to OFF. The mechanical system should be designed so that ORG signal can move more than command pulse output of 1 pulse.

Input status of EZ signal can be checked with RSTS.SEZ bit. Input logic of EZ signal can be set with RENV2.EZL bits. Input-noise filter of EZ can be set with RENV2.EINF. The number of input counts of EZ signal (down-count default value) can be set with RENV3.EZD bit. Down-count value of EZ signal can be checked with RSPD.EZC bit.

+ EL signal input status can be checked with SSTS.SPEL bit.
- EL signal input status can be checked with SSTS.SMEL bit. Input logic of + EL signal and - EL signal can be set with ELLn pin.
+ EL signal and - EL signal input processing can be set with RENV1.ELM bits.
Input-noise filters of + EL signal and - EL signal can be set with RENV1.FLTR bit

| Names and Descriptions | Target |
| :---: | :---: |
| < ORG signal input status > $\begin{aligned} & 0: \text { OFF } \\ & 1: \text { ON } \end{aligned}$ | SSTS.SORG(14) |
| < ORG signal input logic > <br> 0 : Negative logic <br> 1: Positive logic | RENV1.ORGL(7) |
| <+ EL, -EL and ORG signals input-noise filters > <br> 0 : Signals with a pulse width of $0.05 \mu \mathrm{~s}$ or greater are reliably reacted. <br> 1: Pulse width $3 \mu \mathrm{~s}$ or less signals are completely ignored. | RENV1.FLTR(26) |
| <EZ Signal Input status> $\begin{aligned} & \text { 0: OFF } \\ & \text { 1: ON } \end{aligned}$ | RSTS.SEZ(10) |
| <EZ Signal input logic> <br> 0 : Negative logic <br> 1: Positive logic | RENV2.EZL(23) |


| Names and Descriptions | Target |
| :---: | :---: |
| <EA, EB, EZ signal Noise-in filter> <br> 0 : Pulse signal width of $0.05 \mu$ s or greater are reliably reacted. <br> 1: Pulse signal width of $0.10 \mu$ s or less are completely ignored. <br> Pulse signal width of $0.15 \mu \mathrm{~s}$ or greater are reliably reacted. | RENV2.EINF(18) |
| <Number of EZ signals (Default down-count value)> 0000b (once) to 1111b (16 times). | RENV3.EZD(7:4) |
| <EZ signal down-count value > <br> The default value of down-counting is the bit value of RENV3.EZD. | RSPD.EZC(19:16) |
| <+ EL signal input status> <br> 0: OFF <br> 1: ON | SSTS.SPEL(12) |
| <- EL signal input status > <br> 0: OFF <br> 1: ON | SSTS.SMEL(13) |
| <input logic of EL signal in operating direction> <br> L: Positive logic <br> H: Negative logic | ELLn pin |
| <Processing for inputting EL signals in the operation orientation> <br> 0: Immediate stop <br> 1: Deceleration stop | RENV1.ELM(3) |

### 5.5.5.1 Origin return in + direction (10h)

When started, this LSI starts to output command pulses in the +direction.
When the condition for the origin return is met, 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 $+E L$ signal is turned ON.

Set by RENV3.CU1R to CU4R bits whether or not to clear the corresponding counter at the origin position.
Set by RENV1.EROR bit whether or not to output ERC signal when stopped due to the origin return factor.

| Names and Descriptions | Target |
| :---: | :---: |
| <Origin return method> <br> 0000b: Origin return 0 (ORG signal, Deceleration stop) <br> 0001b: Origin return 1 (ORG signal, Reverse, FA Speed, Immediate stop) <br> 0010b: Origin return 2 (ORG signal, EZ signal, Immediate stop) <br> 0011b: Origin return 3 (ORG signal, EZ signal, Deceleration stop) <br> 0100b: Origin return 4 (ORG signal, Reverse, FA Speed, EZ signal, Immediate stop) <br> 0101b: Origin return 5 (ORG signal, Reverse, EZ signal, Deceleration stop) <br> 0110b: Origin return 6 (EL signal, Reverse, FA Speed, Immediate stop) <br> 0111b: Origin return 7 (EL signal, Reverse, FA Speed, EZ signal, Immediate stop) <br> 1000b: Origin return 8 (EL signal, Reverse, FA Speed, EZ signal, Deceleration stop) <br> 1001b: Origin return 9 (Origin return 0, 0-point return) <br> 1010b: Origin return 10 (Origin return 3, 0-point return) <br> 1011b: Origin return 11 (Origin return 5, 0-point return) <br> 1100b: Origin return 12 (Origin return 8, 0-point return) <br> Refer to 5.5.5.1.1 "Origin return 0 ( 0000 b )" and later for the details of "Origin return patterns". | RENV3.ORM(3:0) |
| <Counter 1 is cleared when the Origin is reached by Origin return control> <br> 0 : Does not clear <br> 1: Clear | RENV3.CU1R(20) |
| <Counter 2 is cleared when the Origin is reached by Origin return control> <br> 0 : Does not clear <br> 1: Clear | RENV3.CU2R(21) |
| <Counter 3 is cleared when the Origin is reached by Origin return control> <br> 0 : Does not clear <br> 1: Clear | RENV3.CU3R(22) |
| <Counter 4 is cleared when the Origin is reached by Origin return control> <br> 0 : Does not clear <br> 1: Clear | RENV3.CU4R(23) |
| <Outputting Function of ERCn pin when Stopped by Origin return> <br> 0: ERC signal is not output <br> 1: ERC is output | 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 and FH constant speed patterns, the operation stops at the origin position.
In high speed 1 and 2 speed patterns, the operation stops after passing the origin position.

Example: STAFL (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.


V: 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: STAUD (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.

$\boldsymbol{\nabla}$ : 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 $=45$ (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.


### 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 signal ONs.

Completes the operation mode.


V: Counter clear timing when RENV3.CUnR $=1(n=1,2,3,4)$ is set.
@: ERC signal output timing when RENV1.EROR = 1 is set.

## C a u tion

Even during deceleration, the operation stops immediate when EZ signal ON is counted as many times as specified.

### 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.

@: 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.

$\nabla$ : Counter clear timing when RENV3.CUnR $=1(\mathrm{n}=1,2,3,4)$ 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 (RENV1.ELM = 1, dotted line)
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.
Stops immediately when EL signal is changed from ON to OFF.
Completes the operation mode.


* : Output-timing of ERC when RENV1.EROE = 1 and RENV1.ELM $=0$.

When + EL signal turns ON from OFF, the operation stops immediately without decelerating, and then reverses after outputting ERC .
$\boldsymbol{\nabla}$ : Counter clearing timing when RENV3.CUnR $=1(\mathrm{n}=1,2,3,4)$ is set.
@: Output-timing of ERC 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, RENV1.ELM=1, dotted line)
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.

*: Output-timing of ERC when RENV1.EROE $=1$ and RENV1.ELM $=0$. When + EL signal turns ON from OFF, the operation stops immediately without decelerating, and then reverses after outputting ERC .
$\boldsymbol{\nabla}$ : Counter clearing timing when RENV3.CUnR $=1(\mathrm{n}=1,2,3,4)$ is set.
@: Output-timing of ERC 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 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, RENV1.ELM = 1, dotted line)
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.


* : Output-timing of ERC when RENV1.EROE = 1 and RENV1.ELM $=0$.

When + EL signal turns ON from OFF, the operation stops immediately without decelerating, and then reverses after outputting ERC .

V: Counter clear timing when RENV3.CUnR $=1(n=1,2,3,4)$ is set.
@: Output-timing of ERC 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.


### 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.

$\boldsymbol{\nabla}$ : 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 $+E L$ 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, RENV1.ELM=1, dotted line)
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.


When + EL signal turns ON from OFF, the operation stops immediately without decelerating, and then reverses after outputting ERC .
$\boldsymbol{\nabla}$ : Counter clear timing when RENV3.CUnR $=1(n=1,2,3,4)$ is set.

### 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 operation direction and EL 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 Escape from the origin position in the +direction (12h)" except that the operation is reversed.

### 5.5.5.5 Origin search in the +direction (15h)

Set 1 to 134,217,727 in RMV register.
For origin return method, select origin return method (RENV3.ORM $=0,1,2,3,4,5,9,10,11$ ) that does not use EL signal.

## C a u tion

In this operation mode, EL signal in the operating direction is the reverse function.
Do not select origin return method (RENV3.ORM $=6,7,8,12$ ) that uses EL signal.

When it starts, it operates in one of the following depending on the condition.

1. When ORG signal is OFF, "return to origin in the +direction".

When ORG signal turns ON, the operation mode is completed.
The figure below shows the case of Origin return 0 (RENV3.ORM $=0$ ) and STAFL ( 50 h ).

2. When ORG signal is ON, performs "incremental movement" in the -direction.
"Incremental movement" in the -direction repeats until ORG signal turns OFF.
After "Incremental movement", when ORG signal turns OFF, performs "return to origin in the +direction".
When ORG signal turns ON, the operation mode is completed.
The figure below shows the case of Origin return 0 (RENV3.ORM $=0$ ) and STAFL ( 50 h ).

3. When + EL signal is ON, perform "origin return in the -direction (RENV3.ORM = 0)".

In origin return mode opposite to the specified direction, RENV3.ORM setting must be ignored and origin return 0 must be performed.

Use ORG signal ON for "incremental movement " in the -direction.
"incremental movement " in the -direction is repeated until the ORG signal turns OFF.
After passing through OFF from ORG signal ON, perform "origin return in the +direction".
When ORG signal turns ON, the operation mode is completed.
The figure below shows the case of Origin return 0 (RENV3.ORM = 0) and STAFL (50h).


### 5.5.5.6 Origin search in the -direction (1Dh)

Operates in the same way as "5.5.5.5 Origin search in the +direction (15h)" except that the operation direction and EL 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.

The input status of + EL signal can be checked with SSTS.SPEL bits.
The input status of - EL signal can be checked with SSTS.SMEL bits. Input logic of + EL signal and-EL signal can be set at ELL pin.

+ EL signal and-EL signal input processes can be set with RENV1.ELM bits.
The input noise filter of + EL signal and -EL signal can be set with RENV1.FLTR bits.

The input status of EZ signal can be checked with RSTS.SEZ bit.
Input logic of EZ signal can be set using RENV2.EZL bits.
Input noise filter of EZ signal can be set by RENV2.EINF.
The number of EZ signal input (down count default value) can be set using RENV3.EZD bits.
EZ down-counts can be checked with RSPD.EZC bits.

For +SL and -SL positions, see "6.13.2 Software limit".

| Name and description | Target |
| :---: | :---: |
| <+EL signal input status> <br> 0: OFF. <br> 1: ON. | SSTS.SPEL(12) |
| <-EL signal input status> <br> 0 : OFF. <br> 1: ON. | SSTS.SMEL(13) |
| <Input logic of EL signal in operating direction> <br> L: Positive logic <br> H: Negative logic | ELLn pin |
| <Processing for inputting EL signal in the operation direction> <br> 0: Immediate stop <br> 1: Deceleration stop | RENV1.ELM(3) |
| <+ EL,-EL, ORG signal input-noise filter > <br> 0 : Signals with a pulse width of $0.05 \mu \mathrm{~s}$ or greater are reliably reacted <br> 1: Signals with a pulse width of $3 \mu$ s or less are completely ignored. | RENV1.FLTR(26) |
| <EZ Signal input status> <br> 0 : OFF. <br> 1: ON. | RSTS.SEZ(10) |
| <EZ Signal input logic> <br> 0 : Negative logic <br> 1: Positive logic | RENV2.EZL(23) |


| Name and description | Target |
| :---: | :---: |
| <EA, EB and EZ input noise filters> <br> 0 : Signals with a pulse width of $0.05 \mu \mathrm{~s}$ or greater are reliably reacted <br> 1: Signals with a pulse width $0.10 \mu$ s or less are completely ignored. <br> Signals with a pulse width of $0.15 \mu$ s or greater are reliably reacted. | RENV2.EINF (18) |
| <Number of EZ Signal input (Down count default value) > 0000b (once) to 1111b (16 times) | RENV3.EZD (7:4) |
| <EZ signal down-count value> <br> The initial value of down-counting is the value of RENV3.EZD. | RSPD.EZC (19:16) |

### 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 met, 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 $+E L$ signal is ON or +SL condition is met, 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, signals and conditions are reversed.

### 5.5.6.3 Escape from -EL or -SL (22h)

When the operation starts, it starts outputting command pulse to + direction.
When - EL signal turns OFF and -SL condition are not met, the command pulse stops. When the command pulse stops, the operation is completed.

Performs the abnormal stop when + EL signal turns ON.
Performs the abnormal stop even if $+S L$ condition is met.

If you attempt to start while -EL signal is OFF and -SL condition is un-met, 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 Escape from -EL or -SL (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 operation direction is reversed.

### 5.5.7 Linear interpolation 1 control

This control is to use a PCL6045BL 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 assigned in the order of $X, Y, Z$-axis among the interpolation axes.
The interpolation operation axes are all axes for linear interpolation 1 control.

| No. | Interpolation operation | Interpolation control axis |
| :---: | :--- | :---: |
| 1. | Linear interpolation $1(\mathrm{X}, \mathrm{Y}, \mathrm{Z}, \mathrm{U}$ axes $)$ | X -axis |
| 2. | Linear interpolation $1(\mathrm{Y}, \mathrm{Z}, \mathrm{U}$ axes $)$ | Y -axis |
| 3. | Linear interpolation $1(\mathrm{Z}, \mathrm{U}$ axes $)$ | Z axis |

In the following cases, set the same speed magnification (RMG) and FA speed (RFA) for all Interpolation operation axis.

1. Backlash correction (RENV6.ADJ $=01 \mathrm{~b}$ ) is set.
2. Slip correction (RENV6.ADJ $=10 \mathrm{~b}$ ) is set.

Acceleration/Deceleration, and synthetic speed constant control can be used.
See "6.3.6 Constant synthesized speed control" for "Constant synthetic speed Control".

Interpolation status can be checked on RIPS register shared by all axes.

Start command and Stop commands set Interpolation operation axis to axis selection (SELn) to write to any axis's COMW area. For example, if a 0653 h is written to COMW area of X axis, Y axis and Z axis can start linear interpolation 1 control at high speed 2.

Linear interpolation accuracy:

Linear interpolation draws a straight line from the current to the end-point coordinates. The figure on the right is an example of drawing a straight line to the endpoint coordinates (10, 4).

The position accuracy against the specified straight line in a linear interpolation is $\pm 0.5$ LSB (Least Significant Bit)
 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 coordinate 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 the same operation mode (RMD.MOD $=60 \mathrm{~h}$ ) has been set only to one axis, REST.ESDT = 1 will be set, and an operation will stop without outputting a command pulse. Writing a stop command will stop the command pulse.

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

The following is an example of setting to move the X -axis and Y -axis continuously at a ratio of 2:5.

| Register | X-axis | Y-axis |
| :--- | :---: | :---: |
| RMD.MOD | 60 h | 60 h |
| RMV | 2 | 5 |
| RFL | 5 | 1 |
| RFH | 50000 | 1 |
| RUR | 29 | 1 |
| RDR | 0 | 0 |
| RMG | 0 | 2 |
| 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 speed ratio for the absolute value of RMV register.
All Interpolation operation axis will have the same speed if RMV = 0 is set for all Interpolation operation axis.

### 5.5.7.2 Incremental movement (61h)

When started, outputs the command pulses in the +direction if RMV $>0$ and in the -direction if RMV $<0$.
If the same operation mode (RMD.MOD $=61 \mathrm{~h}$ ) has been set only to one axis, REST.ESDT $=1$ will be set, and an operation will stop without outputting a command pulse.

The main axis is axis whose absolute value of RMV register is the largest among all Interpolation operation axis.
Following axes are the Interpolation operation axes other than main axis.
The command pulse of a following axis is outputted by pulling out command pulse of the main axis.
When the main axis reaches RPLS $=0$, the command pulse will stop.
When the command pulse is stopped, operation mode will be completed.

When all Interpolation operation axes are $\mathrm{RMV}=0$, operation mode will be completed without outputting command pulse.

Set the same slowdown point setting (RMD.MSDP) and FH correction value setting (RMD.MADJ) for all Interpolation operation axis.

In the following cases, all Interpolation operation axis will decelerate when the main axis is RPLS < RSDC.
A) Slowdown point manual setting (RMD.MSDP =1).
B) Slowdown-point auto setting (RMD.MSDP $=0$ ) and FH correction value is manual compensation (RMD.MADJ =1).

The following is a setting example and a command pulse output example that move X -axis, Y -axis, and Z -axis incremental.

| Register | X-axis | Y-axis | Z-axis |
| :--- | :---: | :---: | :---: |
| RMD.MOD | 61 h | 61 h | 61 h |
| RMD.MIPF | 0 | 0 | 0 |
| RMV | 5 | 10 | 2 |
| RFH | 50000 | 1 | 1 |
| RMG | 149 | 2 | 2 |
| Interpolation control axis | 0 | - | - |
| Main axis | - | 0 | - |
| Following axis | 0 | - | 0 |

Setting the speed to an axis other than interpolation control axes will not affect the operation.


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 PCL6045BL and performs linear interpolation on any one or more axes.
With multiple PCL6045BLs, you can start linear interpolations 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.

Set the speed to all Interpolation operation axes.
Interpolation operation axes are all axes for linear interpolation 2 control.
Set the speed of main axis to all Interpolation operation axes.
Set the same speed magnification rate (RMG) for all Interpolation operation axes.

Acceleration, deceleration, and synthetic speed constant control cannot be used.

Interpolation status can be checked on RIPS register shared by all axes.

The start command and the stop command set Interpolation operation axis to axis selection (SELn) to write to any axis's COMW area.

When using more than one PCL6045BL, turn simultaneous start at CSTA pin and turn simultaneous stop at CSTP pin. For details on "simultaneous start", see "6.9.1 Simultaneous start (CSTA) ".

For more information on "simultaneous stop", see "6.10 External stop / Simultaneous stop".
Set RMD.MSPE = 1 and RMD.MSPO = 1 for all Interpolation operation axes in the event of an abnormal stop.

## C a u tion

If more than one PCL6045BL is used, CLK signals are supplied from one crystal oscillator to all PCL6045BLs. If CLK signals are supplied from separate crystal oscillators, the frequency error causes errors in the interpolating locus and the operating times.

Linear interpolation accuracy:
A linear interpolation draws a straight line from the present coordinate to the end-point coordinate. The figure on the right shows an example of drawing a straight line up to the end-point coordinate (10, 4).

Position value accuracy for the specified line at linear interpolation is $\pm 0.5 \mathrm{LSB}$ (Least Significant Bit within the total interpolated area.

LSB is the smallest units of RMV register and is the spacing between the squares in the diagram. It is an equivalent to resolution in the mechanical coordinate system.

### 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 $<0$.
The main axis is the axis whose absolute value of RMV register is the largest in Interpolation operation axis.
Following axis is Interpolation operation axis other than the main axis.
Command pulse of the following axis is outputted by pulling out command pulse of the master axis.
When the stop command is written, command pulse will be stopped.
When command pulse is stopped, operation mode will be completed.

The following is an example to use two PCL6045BLs: PCL6045BL_a and PCL6045BL_b. The X-axis, Y-axis, and Z-axis of PCL6045BL_a are moved continuous to the $X$-axis of PCL6045BL_b at 8:5:2:10 ratio, respectively.

| Register | PCL6045BL_a |  |  | PCL6045BL_b |
| :--- | ---: | ---: | ---: | ---: |
|  | Xa axis | Ya axis | Za axis | Xb axis |
| RMD.MOD | 62 h | 62 h | 62 h | 62 h |
| RMD.MSY | 01 b | 01 b | 01 b | 01 b |
| RMV | 8 | 5 | 2 | 10 |
| RIP | 10 | 10 | 10 | 10 |
| RFH | 50000 | 50000 | 50000 | 50000 |
| RMG | 149 | 149 | 149 | 149 |
| Interpolation control axis | $\circ$ | $\circ$ | $\circ$ | $\circ$ |
| Main axis | - | - | - | $\circ$ |
| Following axis | $\circ$ | $\circ$ | $\circ$ | - |

For RMV register, set the interpolating ratio.
Set the RMV register of the main axis to the RIP register of all Interpolation operation axes.
If RIP $=0$ is set in Interpolation operation axes, REST.ESDT $=1$ will be set and the operation will stop without outputting the 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 $<0$.
If either interpolation operation axis is with RIP $=0$, REST.ESDT $=1$ will be set, and an operation will stop without outputting a command pulse.

The main axis is the axis whose absolute value of RMV register is the largest in Interpolation operation axis.
Following axis is Interpolation operation axis other than the main axis.
Command pulse of the following axis is outputted by pulling out command pulse of the master axis.
When the main axis reaches RPLS $=0$, command pulse will be stopped.
If the main axis is in another PCL6045BL, command pulse is stopped when each Interpolation operation axis becomes RPLS $=$ 0.

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

The following is an example to use two PCL6045BLs: PCL6045BL_a and PCL6045BL_b. The X-axis, Y-axis, and Z-axis of PCL6045BL_a are moved incremental to the X-axis of PCL6045BL_b at 8:5:2:10 ratio, respectively.

| Register | PCL6045BL_a |  |  | PCL6045BL_b |
| :--- | :---: | :---: | :---: | :---: |
|  | Xa axis | Ya axis | Za axis | Xb axis |
| RMD.MOD | 62 h | 62 h | 62 h | 62 h |
| RMD.MSY | 01 b | 01 b | 01 b | 01 b |
| RMV | 8 | 5 | 2 | 10 |
| RIP | 10 | 10 | 10 | 10 |
| RFH | 50000 | 50000 | 50000 | 50000 |
| RMG | 149 | 149 | 149 | 149 |
| Interpolation operation axis | 0 | 0 | 0 | 0 |
| Main axis | - | - | - | 0 |
| Following axis | 0 | 0 | 0 | - |



For the timing of BSY signal and OUT signal, see "7.5 Operation timing".

Set the feeding amount in RMV register.
Set the RMV register of the main axis to the RIP register of all Interpolation operation axes.

### 5.5.9 Circular interpolation control

This control is to use a single PCL6045BL to perform a circular interpolation by any two axes.
The remaining two axes can perform other operations other than a circular interpolation control.

The speed is set to the interpolation control axes.
The interpolation control axes are assigned in the order of $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$-axis in the interpolation axes. Interpolation operation axis is all axis used for the circular interpolation control.

| 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 |
| 4$)$ | Circular interpolation $(X, U$ axes $)$ | $X$ 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/deceleration in a circular interpolation control, see "6.3.5 Circular interpolation step number" For constant synthesized speed control, see "6.3.6 Constant synthesized speed control"

The interpolation status can be checked with RIPS register, which is common to all axes.
Start and Stop commands can be written to any axis's COMW area by setting the Interpolation operation axes to axis selection (SELn).

Circular interpolation accuracy:
Circular interpolation draws a circle from the current coordinates to the end-point 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 $\pm 0.5$ 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 $11 \pm 0.5$

### 5.5.9.1 Circular interpolation in CW (64h)

When started, it starts outputting command pulses 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 ".
When quadrant with the end-point is reached and 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 are examples of the circular interpolations of 360 -degree (perfect circle), 90 -degree, 180 -degree, and 270degree performed by X and Y -axes.

| Register | $360^{\circ}$ (A) |  | $90^{\circ}$ (B) |  | $180^{\circ}$ (C) |  | 270º ${ }^{\text {( }) ~}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | X-axis | Y-axis | X-axis | Y-axis | X-axis | Y-axis | X-axis | Y-axis |
| RMD.MOD | 64h |  |  |  |  |  |  |  |
| RMD.MPIE | 0 |  |  |  |  |  |  |  |
| RMV | 0 | 0 | 100 | 100 | 200 | 0 | 100 | -100 |
| RIP (O) | 100 | 0 | 100 | 0 | 100 | 0 | 100 | 0 |
| RCl | 564 | 0 | 141 | 0 | 282 | 0 | 423 | 0 |
| Interpolation control axis | $\bigcirc$ | - | $\bigcirc$ | - | $\bigcirc$ | - | $\bigcirc$ | - |

Setting the speed and number of circular interpolation steps (RCI) besides the interpolation control axis, won't affect the operation.

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


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. Permanent circular interpolation moves continuously without reaching the end-point. 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 ".

Trajectory when the start point and end point are on the same quadrant)
Center $\mathrm{O}(-40,-60)$, end point $\mathrm{E}(-20,-30)$. Thick broken line is end-point-draw operation when RMD.MPIE $=1$ is set.


Stops when Y axis matches the end point so as to shorten the trajectory.

When the end point is on $X$ axis or $Y$ axis, the end point quadrant is on quadrant short in the forward direction. For example, if the start point is the first quadrant and the end point is on X axis, the end point quadrant is the first quadrant because the direction of travel is CW.

### 5.5.9.2 Circular interpolation in CCW (65h)

When started, it starts outputting command pulses so that two axes draw a circle in CCW direction.
Otherwise, it operates in the same manner as described in "5.5.9.1 Circular interpolation in CW (64h) ".

### 5.5.9.3 Circular interpolation dummy (6Fh)

This operation mode is not used alone.
For details, see "6.17.3 Continuous interpolation using circular interpolation dummy operation".

### 5.5.10 U-axis synchronous control

This control uses one PCL6045BL to synchronize with U-axis and perform circular interpolation by the remaining two axes. Helical interpolation is available for applications such as threading and drilling. With remaining axes, you can perform operations other than a 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 assigned 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) <br> U-axis synchronization (X, Y-axes) | U -axis |
| $(2)$ | Linear interpolation 1 (X, U-axes) <br> U-axis synchronization (Y, Z axes) | X -axis |
| $(3)$ | Linear interpolation 1 (Y, U-axes) <br> U-axis synchronization (X, Z axes) | Y -axis |
| $(4)$ | Linear interpolation 1 (Z, U-axes) <br> U-axis synchronization (X, Y-axes) | Z-axis |

Set the same value of speed magnification (RMG) to all interpolation axes.
The interpolation operation axes are all axes that perform a circular interpolation in U-axis synchronous control.

Combined with linear interpolation 1 control, you can use synthesized speed constant control for the circular interpolation operation axis. For synthesized speed constant control, see "6.3.6 Constant synthesized speed 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 that is two or more larger than the number of circular interpolation steps. For the end-point-draw operation (RMD.MPIE =1), add the number of pulses for the end-pointdraw operation to the number of circular interpolation steps.

The interpolation status can be checked in RIPS register, which is common to all axes.
Start and Stop commands are written to any axis's COMW area after setting the interpolation operation axis to axis selection (SELn).

## Circular interpolation accuracy:

Circular interpolation draws an arc from the present coordinate to the end-point coordinate. The figure on the right shows an example of drawing a perfect circle with a radius of 11 .

When in circular interpolation, position value accuracy for the specified curve is $\pm 0.5$ LSB (Least Significant Bit) within the total interpolating area.

LSB is the smallest units of RMV register and is the spacing between the squares in the diagram. It corresponds to the resolution of the mechanical coordinate system.


- :Interpolation trajectory

Solid line: circle with radius 11
Dashed line: circle with radius $11 \pm 0.5$

### 5.5.10.1 Circular interpolation with U-axis synchronization 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 synchronous control (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 | 66 h | 66 h | 61 h | 61 h |
| RMD.MIPF | 1 | 1 | 0 | 0 |
| RMV | 0 | 0 | 282 | 565 |
| RIP | 100 | 0 | 0 | 0 |
| RCI | 0 | 0 | 0 | 0 |
| RFL | 1 | 1 | 65,500 | 1 |
| RFH | 1 | 1 | 65,535 | 1 |
| RUR | 1 | 1 | 1 | 1 |
| RDR | 0 | 0 | 0 | 0 |
| RMG | 2 | 2 | 2 | 2 |
| RENV2.PMSK | 0 | 0 | 0 | 1 |
| Interpolation control axis | - | - | $O$ | - |
| Main axis | - | - | - | $\bigcirc$ |

Continuous operation using a pre-register enables helical interpolation to connect spirals in the table above.
For example, you can control the acceleration in the first lap, FH constant speed in the second lap and later, and deceleration stop in the last lap.

- In the first lap, you can accelerate to FH constant speed by writing commands: PRMDn.MSDP $=1$, PRDPu $=0$, and STAUD (53h).
- After the second lap, FH constant speed can be continued by writing STAFH (51h) command.
- In the final lap, you can stop after decelerating to FL constant speed by writing commands: PRMDn.MSDP = 0 and PRDPu $=80$, STAD (52h). Automatic slow-down point setting cannot be used because there is no acceleration section in (STAD (52h) command speed pattern


### 5.5.10.2 Circular interpolation with U-axis synchronization in CCW (67h)

When started, command pulses are output so that two axes draw a circle in CCW direction in synchronization with U-axis. Other than the above, it operates in the same way as "5.5.10.1 Circular interpolation with U-axis synchronization in CW (66h) ".

## 6. Function description

This chapter describes the features of PCL6045BL.

### 6.1 Reset

There are two types of reset in PCL6045BL: hardware reset and software reset.
After resetting, PCL6045BL 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 |
| I/O buffer | 0 |
| INT pin | H level |
| WRQ pin | H level |
| IFB pin | Hi-Z |
| D0 to D15 pins | Input pin |
| P0n to P7n pins | H level |
| CSTA pin | H level |
| CSTP pin | H level |
| OUTn pin | H level |
| DIRn pin | H level |
| ERCn pin | BSYn pin |

The signal to the input terminal passes through the noise filter even immediately after reset.
There is a delay corresponding to the noise filter before the state of the input pins is reflected to the status and generalpurpose I/O ports.

## C a $u$ t $\mathrm{i} \quad \mathrm{n}$

After turning ON the PCL6045BL, be sure to perform a "hardware reset" before you start using it.
Before resetting is completed, bi-directional pins may be outputs. Be careful of short circuits and heat generation.

### 6.1.1 Hardware reset

In PCL6045BL, you need to input RST signal to RST pin before CPU communication is started after the power is turned ON.
For RST signal, input an L level signal of 8 cycles or more of CLK signal and an H level signal of 8 cycles or more of CLK signal.
Power $\square$



Access $\qquad$ NG

OK

## l mpor tance

RST pin is pulled up internally.
When driving in open-drain mode, use an external 5 to $10 \mathrm{k} \Omega$ resistor to pull up.

### 6.1.2 Software reset

You can use a software reset if you want to perform a reset again after a hardware reset.
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

In an operation mode, the start command is written after the operation settings such as speed control and position control are written to the register. If the next operation setting is written to a register after the operation mode is completed, the operation stops during this writing time. At this time, by using the pre-register, the subsequent operation settings can be written before the operation mode is completed. The tact time can be reduced by eliminating the operation stop time equivalent to this writing time.

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).


When the pre-register is used, the next operation can be started automatically after the current data operation is completed with the minimum stop time. Refer to " 6.5 .2 Operation complete timing " for the "timing at which the next first pulse is output".

RCMP5 register also has the pre-register for continuous comparison.
The pre-register for continuous comparison is used to set the continuous comparison data.
There is a two-stage configuration as shown in the figure below, and operates in FIFO.

Writing to a register with the pre-register is written to 2 nd pre-register.
If the data is the same as the previous one, no need to write to 2 nd 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 2nd pre-register during stop, can shift to the current register and also becomes the current data. During operation, it shifts to 1st pre-register and becomes the data for 1st continuous operation. If 1st pre-register is determined during operation, it will not shift to 1st pre-register and will be the data for the 2 nd 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 preregister 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 MSTS.SPRF bit. Writing to the 2 nd pre-register is disabled when MSTS.SPRF $=1$ (RSTS.PFM $=3$ ). When writing data for continuous operation, wait until MSTS.SPRF $=0$. To change the data in the 2nd pre-register in determined status, change it to RSTS.PFM = 1 with PRECAN (26h) command. After that, by setting RSTS.PFM $=2$ with the start command for the 1st pre-register, you can change the 2nd preregister only.

The relationship between the write status of the pre-register and RSTS.PFM bit / MSTS.SPRF bit is as follows.

| No. | Method | 2nd preregister | 1st preregister | Current register | RSTS. <br> PFM | MSTS. SPRF |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | This is the default status when operation is stopped. |  | 0 (Un- determined) | 0 (Un- determined) | 0 | 0 |
| 2 | While stopped, write data 1 to the 2 nd pre-register. <br> Data 1 is copied to the 1st pre-register. <br> Data 1 is also copied to the current register. | Data 1 <br> (Undetermined) | Data 1 <br> (Undetermined) | Data 1 <br> (Undetermined) | 0 | 0 |
| 3 | Write start command 1. <br> Data 1 in the current register is determined. <br> Starts an operation with data 1 and start command 1. | Data 1 <br> (Undetermined) | Data 1 <br> (Undetermined) | Data 1 (Determined) | 1 | 0 |
| 4 | While operating with data 1 , write data 2 to the 2 nd preregister for a continuous operation. <br> 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. | $\begin{gathered} \text { Data } 2 \\ \text { (Un- } \\ \text { determined) } \end{gathered}$ | Data 2 (Undetermined) | Data 1 (Determined) | 1 | 0 |
| 5 | While operating with data 1 , write start command 2 for a continuous operation. <br> Data 2 in the 1st pre-register is determined. | Data 2 (Un- determined) | Data 2 <br> (Determined) | Data 1 (Determined) | 2 | 0 |
| 6 | While operating with data 1 , write data 3 for a continuous operation to the 2 nd pre-register. <br> 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 <br> (Determined) | Data 1 (Determined) | 2 | 0 |
| 7 | Write start command 3 for a continuous operation while operating with data 1. <br> Data 3 of the 2 nd pre-register is determined. | Data 3 (Determined) | Data 2 <br> (Determined) | Data 1 (Determined) | 3 | 1 |


| No. | Method | 2nd pre- <br> register | 1st pre- <br> register | Current <br> register | RSTS. <br> PFM | MSTS. <br> SPRF |
| :---: | :--- | :--- | :--- | :--- | :--- | :--- |
| 8 | Data 1 completes the operation mode. <br> Data 2 is copied to the current register. <br> Data 3 is copied to the 1st pre-register. <br> Start the operation with data 2 and start command 2. <br> The 2nd pre-register becomes undetermined. <br> You can write if you have data 4 or start command 4. | Data 3 <br> (Un- <br> determined) | Data 3 <br> (Determined) | Data 2 <br> (Determined) | 2 | 0 |
| 9 | Data 2 completes the operation mode. <br> Data 3 is copied to the current register. <br> Start the operation with data 3 and start command 3. <br> The 1st pre-register becomes undetermined. | Data 3 <br> (Un- <br> determined) | Data 3 <br> (Un- <br> determined) | Data 3 <br> (Determined) | 1 | 0 |
| 10 | Data 3 completes the operation mode. <br> The current register becomes undetermined, and <br> continuou operation is completed. | a ata 3 <br> (Un- <br> determined) | Data 3 <br> (Un- <br> determined) | Data 3 <br> (Un- <br> determined) | 0 | 0 |

If the 2 nd pre-register for a continuous operation is set to be writable (RIRQ.IRNM = 1) in the interrupt request, the 2nd preregister writable (RIST.ISNM = 1) for continuous operation of interrupt factor can be generated when the 2 nd 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 (RMD.METM = 0).

If "the final pulse ON width completed" is set (RMD.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 also can cancel the continuous operation.

The overriding data confirmed by PRSET(4Fh) command for continuous operation can be shifted with PRESHF (2Bh) command. For details, see "6.13.6 Bulk override".

When controlling with a user program of machine tool such as NC (Numerical Control), you may want to control the operation blocks in motion.

In this case, use the RMD.MSN bit because it is difficult to control the operation block only by the control software.
RMD.MSN bit can be repeatedly set from 0 to 3 for each operation block.
The user program and the corresponding block can be managed by reading MSTS.SSC during operation.

| Name and description |  |  |  | Target |
| :---: | :---: | :---: | :---: | :---: |
| <Main status (SPRF)> <br> 0: 2nd pre-register for continuous operation is undetermined. <br> 1: 2nd pre-register for continuous operation is determined. |  |  |  | MSTS.SPRF(14) |
| <Main status (SSC)> <br> Sequence number (RMD.MSN) during operation or when stopped. <br> When you create a user program, you can manage the blocks in operation. <br> The sequence number does not affect operation. |  |  |  | MSTS.SSC(7,6) |
| <Operation mode completion timing> <br> 0 : When output pulse cycle is completed <br> 1: When output-pulse ON duration is completed <br> The timing for completion of the operation is advanced by OFF duration of the last pulse. <br> When using the vibration suppressing function, set the output-pulse cycle completion <br> (RMD.METM = 0). <br> Set RMD.METM $=0$ to continue operation using the pre-register. |  |  |  | RMD.METM(12) |
| <Sequence number> <br> Sets 2-bit sequence number. <br> The sequence number currently in operation is obtained from MSTS.SSC. <br> The sequence number does not affect operation. <br> It can be used to manage operation blocks when creating control software. |  |  |  | RMD.MSN $(17,16)$ |
| <Determined status of pre-register for continuous operation> |  |  |  | RSTS.PFM $(21,20)$ |
| 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 |  |
| <Pre-register control command (PRECAN)> <br> Cancels the determined state of pre-registers for all continuous operations. |  |  |  | PRECAN(26h) |
| <Pre-register control command (PRESHF)> <br> The data of the pre-register for all continuous operations is shifted. |  |  |  | PRESHF(2Bh) |

### 6.2.2 Continuous comparison

When RSTS.PFC $=0$, the data written to 2nd pre-register (PRCP5) is shifted to the current register (RCMP5).
At this time, RCMP5 register is determined, and the write data becomes the current data.
When RSTS.PFC $=1$, data is shifted to 1 st pre-register for 1 st continuous comparison.
When RSTS.PFC = 2, it does not shift. It is used for 2 nd continuous comparison. 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.SPDF bit. Writing to PRCP5 register is disabled when MSTS.SPDF $=1$ (RSTS.PFC=3).

In such cases, wait until MSTS.SPDF = 0 before writing data for continuous comparison data.
To change the data of PRCP5 register in the determined state, change the data to RSTS.PFC $=2$ with PCPCAN(27h) command. Then, write continuous comparison data for 1st pre-register and continuous comparison data for 2nd pre-register.

Writing to RCMP5 register when RSTS.PFC > 0 can override the current data.
When RSTS.PFC $=0$, write to PRCP5 register, not writing to RCMP5 register.

The relationship between the register write status and RSTS.PFC bit / MSTS.SPDF bit is as follows.

| No. | Procedures | 2nd pre-register | 1st pre-register | Current register | $\begin{gathered} \hline \text { RSTS. } \\ \text { PFC } \end{gathered}$ | MSTS. SPDF |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Default status. | 0 <br> (Undetermined) | 0 <br> (Undetermined) | 0 (Un- determined) | 0 | 0 |
| 2 | Write data 1 to PRCP5 register. <br> Data 1 is copied to the 1st pre-register. Data 1 is copied to RCMP5 register. RCMP5 register is determined. | Data 1 <br> (Undetermined) | Data 1 <br> (Un- <br> determined) | Data 1 (Determined) | 1 | 0 |
| 3 | Write data 2 to PRCP5 register. <br> Data 2 is copied to the 1st pre-register. <br> The 1st pre-register is determined. | Data 2 <br> (Undetermined) | Data 2 (Determined) | Data 1 (Determined) | 2 | 0 |
| 4 | Write data 3 to PRCP5 register. PRCP5 register is determined. | Data 3 <br> (Determined) | Data 2 <br> (Determined) | Data 1 (Determined) | 3 | 1 |
| 5 | The comparison result of data 1 changes from true to false. <br> Data 2 is copied to RCMP5 register. <br> Data 3 is copied to the 1st pre-register. | Data 3 <br> (Undetermined) | Data 3 (Determined) | Data 2 (Determined) | 2 | 0 |
| 6 | The comparison result of data 2 changes from true to false. <br> Data 3 is copied to RCMP5 register. | Data 3 <br> (Undetermined) | Data 3 <br> (Undetermined) | Data 3 (Determined) | 1 | 0 |
| 7 | The comparison result of data 3 changes from true to false. <br> RCMP5 register becomes undetermined and the continuous comparison is completed. | Data 3 <br> (Un-determined) | Data 3 <br> (Un- <br> determined) | Data 3 <br> (Un- <br> determined) | 0 | 0 |

An interrupt request can be set to enable writing (RIRQ.IRND = 1) to 2nd pre-register for continuous comparison (PRCP5).
This setting can generate an interrupt factor that can be written to PRCP5 register (RIST.ISND $=1$ ).
This interrupt factor occurs when PRCP5 register changes to undetermined (MSTS.SPDF = 1 to 0 ).

Continuous comparison data can be shifted by PCPSHF(2Ch) command. The condition of RENV5.C5S bits does not change from true to false.

For example, writing a RSTS.PFC = "2" to PCPSHF(2Ch) command can set RSTS.PFC to "1".
In this situation, 1st pre-register is copied to RCMP5 register for continuous comparison.

| Name and description |  |  |  | Target |
| :---: | :---: | :---: | :---: | :---: |
| <Main status (SPDF)> <br> 0 : 2nd pre-register for continuous comparison is undetermined. <br> 1: 2nd pre-register for continuous comparison is determined. |  |  |  | MSTS.SPDF(15) |
| <Determination state of the pre-register for continuation comparison> |  |  |  | RSTS.PFC( 21,20 ) |
| PFC | 2nd pre-register | 1st pre-register | Current register |  |
| 0 | Undetermined | Undetermined | Undetermined |  |
| 1 | Undetermined | Undetermined | Determined |  |
| 2 | Undetermined | Determined | Determined |  |
| 3 | Determined | Determined | Determined |  |
| <Comparator 5 comparison conditions> <br> 001b: RCMP5 = Comparison target (independent of counting <br> 010b: RCMP5 = Comparison target (only during count-up) <br> 011b: RCMP5 = Comparison target (only during count-down) <br> 100b: RCMP5 > Comparison target. <br> 101b: RCMP5 < Comparison target. <br> Others: Comparison condition is always not met. |  |  |  | RENV5.C5S(5:3) |
| <Pre-register control command (PCPCAN)> <br> Cancel the determined status of comparator 5 of comparison pre-register. |  |  |  | PCPCAN(27h) |
| <Pre-register control command (PCPSHF)> <br> The data of the comparator 5 of comparison pre-register is shifted. |  |  |  | PCPSHF(2Ch) |

### 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 STAFL (50h) command <br> $\Rightarrow$ FL constant speed start <br> (2) Executes STOP (49h) command or <br> SDSTP (4Ah) command <br> $\Rightarrow$ Immediate stop | (1) Executes STAFL (50h) command <br> $\Rightarrow$ FL constant speed start <br> (2) RPLS $=0$ or <br> Execute STOP (49h) command or SDSTP (4Ah) command $\Rightarrow$ Immediate stop |
| FH constant speed | (1) Executes STAFH (51h) command <br> $\Rightarrow$ FH constant speed start <br> (2) Executes STOP (49h) command <br> $\Rightarrow$ Immediate stop <br> * Executes SDSTP (4Ah) command <br> $\Rightarrow$ Starts deceleration and stops at FL speed. | (1) Execute the STAFH (51h) command <br> $\Rightarrow$ FH constant speed start. <br> (2) RPLS $=0$ or <br> Executes STOP (49h) command <br> $\Rightarrow$ Immediate stop <br> * Executes SDSTP (4Ah) command <br> $\Rightarrow$ Starts deceleration and stops at FL speed |
| High-speed 1 | (1) Execute STAD (52h) command <br> $\Rightarrow$ FH constant speed start <br> * RENV5.IDL > 0 <br> $\Rightarrow$ FL constant speed start <br> $\Rightarrow$ Outputs idling pulses. <br> $\Rightarrow$ Accelerates to FH speed ( $\mathrm{RUR}>0$ ). <br> (2) Executes SDSTP (4Ah) command <br> $\Rightarrow$ Starts deceleration and stops at FL speed. <br> * Executes STOP (49h) command <br> $\Rightarrow$ Immediate stop | (1) Executes STAD (52h) command <br> $\Rightarrow$ FH constant speed start <br> * RENV5.IDL > 0 <br> $\Rightarrow$ FL constant speed start <br> $\Rightarrow$ Outputs idling pulses. <br> $\Rightarrow$ Accelerates to FH speed. <br> (2) RPLS < RSDC <br> $\Rightarrow$ Starts deceleration. <br> * RPLS $=0$ or <br> Executes STOP (49h) command. <br> $\Rightarrow$ Immediate stop <br> * Executes SDSTP (4Ah) command <br> $\Rightarrow$ Starts deceleration and stops at FL speed. |
| High-speed 2 | (1) Executes STAUD (53h) command <br> $\Rightarrow$ FL constant speed start. <br> $\Rightarrow$ Outputs idling pulses. <br> $\Rightarrow$ Accelerates to FH speed. <br> (2) Executes SDSTP (4Ah) command. <br> $\Rightarrow$ Starts deceleration and stops at FL speed. <br> * Executes STOP (49h) command <br> $\Rightarrow$ Immediate stop | (1) Executes STAUD (53h) command <br> $\Rightarrow$ FL constant speed start. <br> $\Rightarrow$ Outputs idling pulse. <br> $\Rightarrow$ Accelerates to FH speed. <br> (2) RPLS < RSDC <br> $\Rightarrow$ Starts deceleration. <br> * RPLS $=0$ or <br> Executes STOP (49h) command <br> $\Rightarrow$ Immediate stop <br> * Executes SDSTP (4Ah) command <br> $\Rightarrow$ Starts deceleration and stops at FL speed. |

[^1]
### 6.3.2 Speed setting example

The procedure of speed setting is shown below. See "5.4.1 Speed control register" for each calculation formula.
The following applies to $f_{C L K}=19.6608 \mathrm{MHz}, ~$ FL speed $=10 \mathrm{pps}, ~ \mathrm{FH}$ speed $=100 \mathrm{kpps}$, Accleration time $=300 \mathrm{~ms}$, Deceleration time $=300 \mathrm{~ms}$.

1. FH speed $<65,535 \mathrm{pps} \times 2$ Therefore, find the setting of RMG register from $f_{C L K}$ that multiplies the speed by 2 .

$$
\mathrm{RMG}=149(095 \mathrm{~h})=\frac{19,660,800[\mathrm{~Hz}]}{2 \times 65,536}-1
$$

2. Calculate the setting of RFH register from the velocity multiplier to be FH speed $=100 \mathrm{kpps}$.

RFH $=50,000(C 350 h)$
3. Calculate the setting of RFL register from the velocity multiplier to be FL speed $=10 \mathrm{pps}$.
$R F L=5(0005 h)$
4. Calculate the setting value of RUR register to be Acceleration time $=300 \mathrm{~ms}$

$$
\operatorname{RUR}=28.494(001 \mathrm{Ch} \text { or } 001 \mathrm{Dh})=\frac{19,660,800[\mathrm{~Hz}] \times 0.3[s]}{(50,000-5) \times 4}-1
$$

5. $\quad$ Deceleration time $=$ Acceleration time. Therefore, 0 can be set in RDR register.

$$
\mathrm{RDR}=28.494(0000 \mathrm{~h})
$$

<Example of speed setting of RUR = 29>

| Writing register | Setting value | Actual value |
| :---: | :---: | :---: |
| PRFL | 0005 h | 10 pps |
| PRFH | C350h | 100 kpps |
| PRMG | 095 h | 2 times |
| PRUR | 001 Dh | 305 ms |
| PRDR | 0000 h | 305 ms |



### 6.3.3 Manual correction calculation of the target 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 $=41 h, 42 h, 43 h, 44 h, 45 h, 51 h, 52 h, 53 h, 54 h, 55 h, 56 h, 61 h, 64 h, 65 h, 66 h, 67 h, 69 h$, $6 \mathrm{Ch}, 6 \mathrm{Dh}$ and 6Fh.

If the target FH speed is too high for the feeding amount, or if the feeding amount is too small for FH speed, the operation will be a triangular drive.


Normal linear acceleration / deceleration


Starts decelerating in the middle of acceleration (Triangular drive)

To avoid a triangular drive (RMD.MADJ $=0$ ), FH speed is automatically lowered.
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 $\times 2$, the start of deceleration will be delayed, and the operation will stop before reaching FL speed. This can be avoided by setting an offset in RDP register or by the manual slow-down point setting (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. If RENV5.IDL > 0, replace RMV in the equation with RMV-RENV5.IDL.

### 6.3.3.1 Linear acceleration/deceleration

The FH speed in linear acceleration / deceleration (RMD.MSMD $=0$ ) is calculated by the following formula.
If

$$
\begin{aligned}
& R M V \leqq \frac{\left(R F H^{2}-R F L^{2}\right) \times(R U R+R D R+2)}{(R M G+1) \times 32768} \\
& R F H \leqq \sqrt{\frac{(R M G+1) \times 32768 \times R M V}{R U R+R D R+2}+R F L^{2}}
\end{aligned}
$$

### 6.3.3.2 Complete S-curve acceleration/deceleration

The FH speed in complete S-curve acceleration / deceleration ( $R M D . M S M D=1, R U S=0, R D S=0$ ) without a linear acceleration / deceleration section is calculated by the following formula.

If

$$
\begin{aligned}
& R M V \leqq \frac{\left(R F H^{2}-R F L^{2}\right) \times(R U R+R D R+2) \times 2}{(R M G+1) \times 32768} \\
& R F H \leqq \sqrt{\frac{(R M G+1) \times 32768 \times R M V}{(R U R+R D R+2) \times 2}+R F L^{2}}
\end{aligned}
$$

### 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 RUS = RDS

If

$$
\begin{aligned}
& R M V \leqq \frac{(R F H+R F L) \times(R F H-R F L+2 \times R U S) \times(R U R+R D R+2)}{(R M G+1) \times 32768} \quad \text { and } \\
& R M V>\frac{(R U S+R F L) \times R U S \times(R U R+R D R+2) \times 8}{(R M G+1) \times 32768} \\
& R F H \leqq-R U S+\sqrt{\frac{(R M G+1) \times 32768 \times R M V}{(R U R+R D R+2)}+(R U S-R F L)^{2}}
\end{aligned}
$$

If

$$
R M V \leqq \frac{(R U S+R F L) \times R U S \times(R U R+R D R+2) \times 8}{(R M G+1) \times 32768}
$$

Change to complete S-curve acceleration / deceleration (RUS $=0, R D S=0$ ) without a linear acceleration / deceleration section.
$R F H \leqq \sqrt{\frac{(R M G+1) \times 32768 \times R M V}{(R U R+R D R+2) \times 2}+R F L^{2}}$

### 6.3.3.3.2 RUS < RDS

If

$$
R M V \leqq \frac{(R F H+R F L) \times((R F H-R F L) \times(R U R+R D R+2)+2 \times R U S \times(R U R+1)+2 \times R D S \times(R D R+1))}{(R M G+1) \times 32768} \quad \text { and }
$$

$$
R M V>\frac{(R D S+R F L) \times(R D S \times(R U R+2 \times R D R+3)+R U S \times(R U R+1)) \times 4}{(R M G+1) \times 32768}
$$

$$
R F H \leqq \frac{-A+\sqrt{A^{2}+B}}{R U R+R D R+2}
$$

However, $\quad A=R U S \times(R U R+1)+R D S \times(R D R+1)$

$$
B=\left((R M G+1) \times 32768 \times R M V-2 \times A \times R F L+(R U R+R D R+2) \times R F L^{2}\right) \times(R U R+R D R+2)
$$

If

$$
\begin{aligned}
& R M V \leqq \frac{(R D S+R F L) \times(R D S \times(R U R+2 \times R D R+3)+R U S \times(R U R+1)) \times 4}{(R M G+1) \times 32768} \quad \text { and } \\
& R M V>\frac{(R U S+R F L) \times R U S \times(R U R+R D R+2) \times 8}{(R M G+1) \times 32768}
\end{aligned}
$$

Change to S-curve deceleration (RUS >0, RDS = 0) without a linear deceleration section.

$$
\begin{aligned}
& R F H \leqq \frac{-A+\sqrt{A^{2}+B}}{R U R+2 \times R D R+3} \\
& \text { However, if } \quad A=R U S \times(R U R+1) \\
& \qquad \begin{array}{l}
B=\left((R M G+1) \times 32768 \times R M V-2 \times A \times R F L+(R U R+2 \times R D R+3) \times R F L^{2}\right) \\
\\
\quad \times(R U R+2 \times R D R+3)
\end{array} \\
& R M V \leqq \frac{(R U S+R F L) \times R U S \times(R U R+R D R+2) \times 8}{(R M G+1) \times 32768}
\end{aligned}
$$

Change to S-curve acceleration / deceleration ( $R U S=0, R D S=0$ ) without a linear acceleration / deceleration part.
$R F H \leqq \sqrt{\frac{(R M G+1) \times 32768 \times R M V}{(R U R+R D R+2) \times 2}+R F L^{2}}$

### 6.3.3.3.3 RUS > RDS

If

$$
R M V \leqq \frac{(R F H+R F L) \times((R F H-R F L) \times(R U R+R D R+2)+2 \times R U S \times(R U R+1)+2 \times R D S \times(R D R+1))}{(R M G+1) \times 32768} \text { and }
$$

$$
R M V>\frac{(R U S+R F L) \times(R U S \times(2 \times R U R+R D R+3)+R D S \times(R D R+1)) \times 4}{(R M G+1) \times 32768}
$$

$$
R F H \leqq \frac{-A+\sqrt{A^{2}+B}}{R U R+R D R+2}
$$

However,

If

$$
\begin{aligned}
& R M V \leqq \frac{(R U S+R F L) \times(R U S \times(2 \times R U R+R D R+3)+R D S \times(R D R+1)) \times 4}{(R M G+1) \times 32768} \quad \text { and } \\
& R M V>\frac{(R D S+R F L) \times R D S \times(R U R+R D R+2) \times 8}{(R M G+1) \times 32768}
\end{aligned}
$$

Change to S-curve acceleration (RUS $=0$, RDS $>0$ ) without a linear acceleration section.

$$
R F H \leqq \frac{-A+\sqrt{A^{2}+B}}{2 \times R U R+R D R+3}
$$

However,
if

$$
A=R D S \times(R D R+1)
$$

$$
\begin{aligned}
B=((R M G+1) & \left.\times 32768 \times R M V-2 \times A \times R F L+(2 \times R U R+R D R+3) \times R F L^{2}\right) \\
& \times(2 \times R U R+R D R+3)
\end{aligned}
$$

$R M V \leqq \frac{(R D S+R F L) \times R D S \times(R U R+R D R+2) \times 8}{(R M G+1) \times 32768}$
Change to S-curve acceleration / deceleration ( $R U S=0, R D S=0$ ) without a linear acceleration / deceleration part.
$R F H \leqq \sqrt{\frac{(R M G+1) \times 32768 \times R M V}{(R U R+R D R+2) \times 2}+R F L^{2}}$

$$
\begin{aligned}
& A=R U S \times(R U R+1)+R D S \times(R D R+1) \\
& B=\left((R M G+1) \times 32768 \times R M V-2 \times A \times R F L+(R U R+R D R+2) \times R F L^{2}\right) \\
& \times(R U R+R D R+2)
\end{aligned}
$$

### 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 in the FL and FH constant-speed patterns, the speed changes to the new speed without accelerating or decelerating.

While operating in high-speed 1 or 2 speed patterns, a motor accelerates or decelerates and reaches the new target 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".

In the auto slow-down point setting (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 will not be calculated correctly.

The target is RMD.MOD $=41 \mathrm{~h}, 42 \mathrm{~h}, 43 \mathrm{~h}, 44 \mathrm{~h}, 45 \mathrm{~h}, 51 \mathrm{~h}, 52 \mathrm{~h}, 53 \mathrm{~h}, 54 \mathrm{~h}, 55 \mathrm{~h}, 56 \mathrm{~h}, 61 \mathrm{~h}, 64 \mathrm{~h}, 65 \mathrm{~h}, 66 \mathrm{~h}, 67 \mathrm{~h}, 69 \mathrm{~h}, 6 \mathrm{Ch}$ and 6Dh.

Example of speed pattern change due to speed change during linear acceleration / deceleration operations

(1) Change RFH register during acceleration:
(2) (3) Change RFH register after acceleration:

If the changed speed is lower than the current speed, it will decelerate linearly to the new target speed.
Linear acceleration or linear deceleration to the new target speed.

## Example of speed pattern change due to speed change during S-curve acceleration / deceleration operation


(1) Change RFH register during acceleration:
(2) Change RFH register during acceleration:
(3) Change RFH register during acceleration:
(4) (5) Change RFH register after acceleration:

If the changed speed is lower than the current speed, it will perform S-curve deceleration to the new target speed.
If the changed speed is higher than or equal to the current Speed, or less than or equal to the original target speed, the S-curve characteristics will not change and the speed will be accelerated to the new target speed.
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 the new target speed.
S-curve accelerates or decelerates to the new target speed.

### 6.3.5 Circular interpolation step number

When performing acceleration/deceleration operations in the circular interpolation, you need to set the circular interpolation step number ( RCI ). When the speed pattern is in high-speed 1 or high-speed 2, the interpolation control axis starts decelerating at RCIC < RSDC. The circular interpolation step number is the number of pulses (step number) that either axis is outputting. The interpolation control axes are determined in the order of $X, Y$, and $Z$ axes. Set the same value to the slow-down point setting (RMD.MSDP) and FH correction setting (RMD.MADJ) for all interpolation operation axes.

In circular interpolation control, the speed of each interpolation axis varies like a trigonometric function. The acceleration/ deceleration motion of each interpolation motion axis is combined with this speed fluctuation, so the resultant speed will differ depending on the interpolation angle, therefore, the acceleration/deceleration operation will not match the settings.

RCI register (Circular interpolation step number) is used to generate the deceleration start timing in circular interpolations. When decelerating with a circular interpolation control, set the number of circular interpolation steps of 1 or more in RCl register. When calculating the circular interpolation step number, divide the XY plane shown in the figure below into eight areas at the center of the circle. The command output pulses by each axis in each area will be as follows.

| Area | Interpolated X-axis output pulse | Interpolated Y-axis output pulse |
| :---: | :---: | :---: |
| 0 | Outputs per the interpolation <br> calculation result | Always outputs |
| 1 | Always outputs | Outputs per the interpolation <br> calculation result |
| 2 | Always outputs | Outputs per the interpolation <br> calculation result |
| 3 | Outputs per the interpolation <br> calculation result | Always outputs |
| 4 | Outputs per the interpolation <br> calculation result | Always outputs |
| 5 | Always outputs | Outputs per the interpolation <br> calculation result |
| 6 | Always outputs | Outputs per the interpolation <br> calculation result |
| 7 | Outputs per the interpolation <br> calculation result | Always outputs |



In this way, any axis always outputs pulses in any area.
Therefore, the circular interpolation step number 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 endpoints 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 point 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 and the center point 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 is not on the circle, add the number of pulses required for the end-point-draw operation and set it in the RCl register.

If the RCI register 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 in the positioning control can be applied with the number of circular interpolation steps (RCI) as the feed amount (RMV).

However, when using the Constant synthesized speed control (RMD.MIPF = 1), obtain it from the change in the RCIC register in the experiment.

## $R$ emarks

If you can perform a test run, the optimum RCl value can be determined by the following experiment.

1. Set the command pulse output disabled (RENV2.PMSK = 1).
2. Set RCI = FFFFFFFFF (maximum value: 4,294,967,295).
3. Set the register 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. The calculation result will be the optimum RCl values.

## I mportance

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.
It is also possible to decelerate by the End-point-draw operation after the circular interpolation.
In this case, 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 or circular interpolation control constant.

| Operation mode | MRD.MIPF = 1 | Description |
| :--- | :---: | :--- |
| Linear interpolation 1 control (60h, 61h) | $\bigcirc$ | Enabled |
| Linear interpolation 2 control (62h, 63h) | $\times$ | Disabled |
| Circular interpolation control (64h, 65h) | $\bigcirc$ | Enabled |
| U-axis synchronous control (66h, 67h) | $\triangle$ | Enabled for circular interpolation axis in <br> combination with linear interpolation 1 control |

Constant synthesized speed control (RMD.MIPF =1) can be used for axes with the same operation mode (RMD.MOD). Cannot be used with linear interpolation 2 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 outputting 4 -axis pulses at the same time, it is assumed to be 3 -dimensional composite speed constant control and $\sqrt{3}$ times the time is secured. Also, when relative movement is performed using linear interpolation 1 control on the 3 axes of $X$, $Y$, and $Z$, if only the $X$ and $Y$ axes are subject to constant composite speed control, simultaneous output of the 3 axes. Even with force, only $\sqrt{2}$ times the time is secured.

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, and $6+4 \sqrt{2} \fallingdotseq 11.66$ seconds is the moving time. The ideal trajectory movement time is $\sqrt{10^{2}+4^{2}} \fallingdotseq 10.77$ seconds, so use constant synthesized speed control considering this difference.

In circular interpolation control, the RSDC register will be calculated correctly if slow-down point automatic setting (RMD.MSDP $=0$ ) is used. However, the condition is that both the start and end points are on the center axis of the arc (in 90-degree increments).

We do not recommend using the constant synthesized speed control with acceleration or deceleration. If the control axis is RUR $\neq$ RDR, the RSDC register will not be calculated correctly in the constant composite speed control. If the speed is changed using S-shaped acceleration/deceleration characteristics, RSDC register will not be calculated correctly in the Constant synthesized speed control.

### 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 $=41 \mathrm{~h}, 42 \mathrm{~h}, 43 \mathrm{~h}$ and 47 h 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.
The following is an example of target position override 1 while operating in high speed 2 speed pattern.

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 Fu speed, the re-acceleration curve will be the same as the normal

 acceleration curve of $\mathrm{FL}=\mathrm{Fu}$.
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.


If the operation mode is an incremental position of positioning control (RMD.MOD $=41 \mathrm{~h}$ ), 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 $R M V=100$, it will be recalculated to RPLS $=150$, neither overwritten ( $R P L S=200$ ) nor added ( $R P L S=250$ ).

The target position (RMV) can be rewritten and re-overridden as many times as required to complete the operating mode. In example 3 above, if you re-override to a position farther away than the initial target position, the speed pattern will change at that timing. During deceleration before reversal, the motor will decelerate to FL speed and then move to the target position for re-override in FL constant speed pattern. If the moving part is in reverse after decelerating and stopping, it will move to the target position for re-override in the high speed 2 speed pattern after decelerating and stopping in reverse.

When accelerating or decelerating, the auto slow-down point setting (RMD.MSDP = 0) causes a cumulative error in the RSDC register value. If the cumulative error cannot be ignored, use manual slow-down point setting (MRD.MSDP=1).

If deceleration time $>$ acceleration time $\times 2$, deceleration to FL speed may not be possible as shown in the figure below.


During deceleration with the auto slow-down point setting, 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 the auto slow-down point setting, the new target position can be overridden before the initial target position. During this deceleration, even if you override the new target position further away, the deceleration will continue to the FL speed. When the FL speed is reached, it accelerates again and positions itself to the new target position.

During deceleration due to auto slow-down point setting, the new target position can be overridden to be further away than the initial target position. In this case, immediately accelerate again and position to the new target position.

The target position override is enabled 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 (MSTS.SEOR = 1 ).
This happens when writing to RMV register in the stopped status.
In the operation mode where the target position can be overridden, it occurs even before starting.
MSTS.SEOR bit can be reset to 0 with the SEORR (2Eh) command.
If the RENV5.MSMR bit is set to 0 , the MSTS.SEOR bit will be reset to 0 even when the main status is read.

## $R$ e marks

When shifting the override data by writing PRESHF (2Bh) command or when the condition of Comparator 5 is met, 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 $=41 \mathrm{~h}$.
Do not use the target position override 2 in other modes of operation.
The minimum pulse width of PCS signal requires 2 cycles ( $0.1 \mu \mathrm{~s}$ ) of the CLK signal.
The PCS signal is sampled in synchronization with the CLK signal after the operation mode starts.

Target position override 2 starts by setting the following two at the same time.

1. PCSn terminal input is PCS signal (RENV1.PCSM $=0$ )
2. Target position override 2 function enabled (RMD.MPCS $=1$ )

When started when the PCS signal is OFF, the RPLS register does not count down and operates like continuous operation mode. Positioning control starts by count-down the RPLS register from the position where the PCS signal turns ON.

If you start when the PCS signal is ON, positioning control will start 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 |
| :--- | :---: |
| <PCSn pin input function> | RMD.MPCS(14) |
| $0:$ General-purpose input pin |  |
| 1: Input pin for PCS signal for target position override 2 | RENV1.PCSL(24) |
| <PCS signal input logic> |  |
| $0:$ Negative logic. |  |
| 1: Positive logic. |  |

<Input function of CSTA pin and PCSn pin>
RENV1.PCSM(30)
0 : CSTA pin input is enabled.
The PCSn pin reflects the setting of the RMD.MPCS bit.
1: CSTA pin input is disabled.
The PCSn pin also inputs the STA signal for starting only the own axis.

|  | $\begin{aligned} \text { RENV1.PCSM } & =0 \\ \text { RMD } \cdot \text { MPCS } & =0 \end{aligned}$ | $\begin{aligned} \text { RENV1.PCSM } & =0 \\ \text { RMD.MPCS } & =1 \end{aligned}$ | $\begin{aligned} \text { RENV1.PCSM } & =1 \\ \text { RMD.MPCS } & =0 \end{aligned}$ | $\begin{aligned} \text { RENV1.PCSM } & =1 \\ \text { RMD.MPCS } & =1 \end{aligned}$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { RSTS.SSTA } \\ & (\text { CSTA = L) } \end{aligned}$ | 1 | 1 | 0 | 0 |  |
| RSTS.SSTA <br> (PCS:ON) | 0 | 0 | 1 | 1 |  |
| RSTS.SPCS $(\text { CSTA }=\mathrm{L})$ | 0 | 0 | 0 | 0 |  |
| RSTS.SPCS <br> (PCS:ON) | 1 | 1 | 1 | 1 |  |
| Override (PCS:ON) | Disabled | Enabled | Disabled | Enabled |  |
| <PCS signal input status > <br> 0 : OFF <br> 1: ON |  |  |  |  | RSTS.SPCS(8) |
| <PCS signal input substitution > <br> Positioning control can be started instead of inputting the PCS signal to the PCSn pin. |  |  |  |  | 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 end-point is 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 avoid 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 operation axis will be RPLS $>0$. Check the operating direction of insufficient number of pulses in RCUN1 register or RCUN2 register.

Circular interpolation completes the interpolation movement when one axis reaches the end-point in the end-point quadrant. The end-point-draw operation moves the other axis to the end-point after the interpolation operation is completed. The speed of the end-point-draw operation draw operation is the same as the speed of the circular interpolation operation. When enabling the end-point-draw operation (RMD.MPIE $=1$ ), add the number of end-point-draw pulses to the RCI register

If the end-point is located directly on 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 on 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.MIPE =1)

In other cases, an arc is drawn and the end point is drawn from the top of the arc to the end point. For the trajectory by setting the RMD.MIPM bit, see "5.5.9.1 Circular interpolation in CW (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. | RMD.MPIE(27) |
| 1: End-point-draw function is performed in the circular interpolation and moves up to the end-point. |  |

### 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): The operation pulse (OUT) and direction identification signal (DIR) are output. (RENV1.PMD $=000 \mathrm{~b}$ 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)


| Set the direction change timer time. | RENV1.DTMF(28) |
| :--- | :--- |
| 0 : If RENV1.PMD $=000 \mathrm{~b}$ to 011 b , wait 0.2 ms for pulse output after changing directions. |  |
| 1: If RENV1.PMD $=000 \mathrm{~b}$ to 011 b , wait $0.5 \mu$ s for pulse output after changing directions. |  |

### 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.

Since the motor driver operates on the edge of the output pulse, the OFF width of the final pulse is not required. For example, if the FL speed stops at 1 pps , the final pulse OFF width of 500 ms is not required if the duty ratio is set to $50 \%$. By combining this with output pulse width control, the operation completion timing can be further shortened. For "Output pulse width control", see "6.5.3 Output pulse width control ".

The following is the operation completion timing is the common pulse mode (RENV1.PMD $=000 \mathrm{~b}$ ).

1. Final pulse cycle completed (RMD.METM = 0)

2. Final pulse ON width completed (RMD.METM $=1$ )


When using continuous operation using pre-register, the timing at which the first pulse of the next block is output changes. When the final pulse cycle is completed (RMD.METM $=0$ ), it is output $12 \times$ TCLK after the completion of the operation mode. If the final pulse ON width is completed (RMD.METM $=1$ ), it will be output at least $17 \times T$ CLK after the final pulse is completed. $T_{C L K}$ is one cycle of the reference clock.

| Name and description | Target |
| :--- | :--- |
| <Operation completion timing> | RMD.METM(12) |
| $0:$ 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 the output pulse cycle completion when using pre-registered continuous operation. |  |


| <Operating signal> | MSTS.SRUN(1) |
| :--- | :--- |
| 0 : Stopping: BSYn pin outputs H-level. |  |
| 1: Operating: BSYn pin outputs L-level. |  |

### 6.5.3 Output pulse width control

When the output speed of a command pulse is $\frac{f_{C L K}}{8192}$ ( 2.4 kpps ) or less, the output pulse width is fixed narrowly to $f_{C L K} \times 4096$ ( 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 earlier.

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 is in even, the following error will occur even if the output pulse width is set to a duty ratio of 50 \%.

$$
\text { ON time: } \text { OFF time }=\frac{R M G}{2}: \frac{R M G}{2}+1
$$

For example, when RMG = 14 (Eh), ON time: OFF time $=7: 8$, and OFF time becomes longer.


## I mportance

When outputting a maximum speed of 6.5 Mpps at 100 x magnification ( $\mathrm{RMG}=2$ ), 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: $2 \times 50 \mathrm{~ns}$ ).

| Name and description | Target |
| :--- | :---: |
| <Output pulse width control> <br> 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. <br> 1: The output pulse width fluctuates with a duty ratio of $50 \%$ regardless of the output speed of a command <br> pulse. | RENV1.PDTC(31) |
| <Speed magnification> | Ror details, see "5.4.1.5 RMG(PRMG): Speed magnification". |

### 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 initial speed will be FL speed.
[Idling pulse setting value and acceleration start timing]
When $n=0,1$


Acceleration starts from the 0th pulse.
When $n=3$


Acceleration starts from the 3rd pulse.

| Name and description | Target |
| :--- | :---: |
| <Number of idling pulse outputs> | RENV5.IDL(10:8) |
| $000 \mathrm{~b}:$ No idling pulse is output. |  |
| 001 b to $111 \mathrm{~b}: 0$ to 6 pulses are output. | RSPD.IDC(22:20) |
| <Idling count value> <br> Down counter. The initial value is the value of RENV5.IDL bit. |  |

### 6.7 Mechanical external input control

In addition to the termination switch (+ELn, -ELn), origin switch (ORGn), and deceleration switch (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)

$+E L$ 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 Timer (RMD.MOD $=47 \mathrm{~h}$ ), 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.
When +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 $+E L$ and $-E L$ signals can be read by sub-status (SSTS.SPEL, SMEL).

In some operation modes in origin return control (RMD.MOD $=10 \mathrm{~h}, 18 \mathrm{~h}, 15 \mathrm{~h}, 1 \mathrm{Dh}$ ), a motor may not stop abnormally. In even some operation modes in sensor controls (RMD.MOD $=20 h, 28 h, 22 h, 2 A h$ ), a motor may not stop abnormally. See the description of each operation mode.

| Name and description | Target |
| :---: | :---: |
| <Input logic of EL signal in the operation direction> <br> L: Positive logic <br> H: Negative logic | ELLn pin |
| <Input processing of EL signal in the operation direction> <br> 0: Immediate stop <br> 1: Decelerate-stop | RENV1.ELM(3) |
| <Input noise filter for +EL, -EL, +SD, -SD, ORG, ALM, INP, CEMG signals > <br> 0 : Signal with the pulse width of $0.05 \mu$ s or more reacts reliably <br> 1: Signals with the width of $3 \mu \mathrm{~s}$ or less is ignored completely | RENV1.FLTR(26) |
| <Error interrupt factor (+EL)> <br> 1: +EL signal turns ON for abnormal stop | REST.ESPL(5) |
| <Error interrupt factor (-EL)> <br> 1: - EL signal turns ON for abnormal stop | REST.ESML(6) |
| <+EL signal input status> <br> 0: OFF <br> 1: ON | SSTS.SPEL(12) |
| <-EL signal input status> | SSTS.SMEL(13) |

### 6.7.2 Slow-down (SD)

The input function (RMD.MSDE) of SD pins can be selected.
If SD signal is ON and deceleration or deceleration stop (RMD.MSDE $=1$ ) is set, SD signal 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 constant and FH constant speed patterns, 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 signa turns OFF, it accelerates to FH speed.
- When SD signal turns ON if writing STAD (52h) or STAUD (53h) command, the machine operates at FL speed. In this case, when SD signal turns OFF, it accelerates to FH speed.

STAFL(50h) command:


STAFH(51h) command:


STAD(52h),STAUD(53h) command:

(2) Latch \& deceleration <RENV1.SDM $=0$, RENV1.SDLT $=1>$

- In FL constant and FH constant speed patterns, SD signals are ignored.
- In high-speed 1 and 2 speed patterns, if SD signal turns ON, the operation decelerates to FL speed. In this case, even if SD signal turns OFF after or during deceleration, it will not accelerate.
- If SD signal turns ON when writing STAD (52h) or STAUD (53h) command, the motor operates at FL speed. In this case, even if SD signal turns OFF, it will not accelerate to FH speed.

STAFL(50h) command:


STAFH(51h) command:


STAD(52h),STAUD(53h) command:

(3) Deceleration stop <RENV1.SDM $=1$, RENV1.SDLT $=0>$

- In FL constant and FH constant speed patterns, if SD signal turns ON, the operation stops immediately.
- In high-speed 1 and 2 speed patterns, if SD signal turns ON, the operation decelerates to FL speed and stops. In this case, if SD signal turns OFF during deceleration, it accelerates to FH speed.
- If SD signal in the operation direction turns $O N$ when writing a start command, the operation mode is completed without starting.
- If the operation stops by turning ON the SD signal, an error interrupt (REST.ESSD) is generated at the time of stop.

STAFL(50h) command:
FH

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 turns ON, the operation stops immediately.
- In high-speed 1 and 2 speed patterns, if SD signal turns ON, the operation decelerates to FL speed and stops. In this case, even if SD signal turns OFF during deceleration, the operation will not accelerate.
- If SD signal turns ON when writing the start command, the operation mode is completed without starting.
- If the operation stops by turning ON the SD signal, an error interrupt (REST.ESSD) is generated at the time of stop.

STAFL(50h) command:


STAFH(51h) command:


STAD(52h),STAUD(53h) command:


The Input logic (RENV1.SDL) of SD signals can be selected.
The latch status of SD signal can be read by the sub status (SSTS.SSD).

The input latch function (RENV1.SDLT) of SD signals can be selected.
If latch the SD signal (RENV1.SDLT = 1) is set, SSTS.SSD = 1 will be set when SD signal turns ON.
If do not latch the SD signal (RENV1.SDLT $=0$ ) is set when SD signal tuns OFF, it returns to SSTS.SSD $=0$.
When SD signal is OFF at the start, it will also return to SSTS.SSD $=0$.

The Input noise filter (RENV1.FLTR) of SD signals can be selected.
Abnormal stop due to SD signals can be read by the error interrupt factor (REST.ESSD).
The input status of SD signals can be read by the extended status (RSTS.SDIN).
The latch status of SD signals can be read by the extended status (RSTS.SSD).
When RENV1.SDLT $=0$, SSTS.SSD bit has the same value as the RSTS.SDIN bit.

| Name and description | Target |
| :---: | :---: |
| <Input function of SDn pin > <br> 0 : General-purpose input pin <br> 1: Decelerate or decelerate-stop. | RMD.MSDE(8) |
| <Input processing of SD signal> <br> 0: Decelerate. <br> 1: Decelerate and stop. | RENV1.SDM(4) |
| <SD signals input latch function> <br> 0 : Not latch SD signal <br> 1: Latch SD signal | RENV1.SDLT(5) |
| <Input logic SD signal> <br> 0 : Negative logic <br> 1: Positive logic | RENV1.SDL(6) |
| < Input noise filters for +EL, -EL, SD, ORG, ALM, INP and CEMG signals > <br> 0 : Signals with the pulse width of $0.05 \mu \mathrm{~s}$ or more will react reliably. <br> 1: Signals with the pulse width of $3 \mu$ s or less are ignored completely. | RENV1.FLTR(26) |
| <Error interrupt factor (SD)> <br> 1: SD signal turns ON to stop abnormally. | REST.ESSD(10) |
| <Latch status of SD signal> $\begin{aligned} & \text { 0: OFF } \\ & 1: \text { ON } \end{aligned}$ | SSTS.SSD(15) |
| <Input status of SD signal > $\begin{aligned} & \text { 0: OFF } \\ & \text { 1: ON } \end{aligned}$ | RSTS.SDIN(15) |

### 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 $=24 \mathrm{~h}, 2 \mathrm{Ch}$ ).

The input logic (RENV1.ORGL) of ORG signal can be selected.
The input noise filter (RENV1.FLTR) of ORG pin can also be selected.
The input status of ORG signal can be read by sub status (SSTS.SORG).
SSTS.SORG bit changes after passing 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 $=10 \mathrm{~h}$, RENV1.ORGL $=0$, RENV1.FLTR $=0$, RENV2.EZL $=0, \quad$ RENV2.EINF $=0$, RENV3. $O R M=0010 \mathrm{~b}$ )
(1) When
(2) When
(3) When
$2 \times T_{C L K} \leqq \quad \mathrm{t}$
$T_{C L K}<\mathrm{t}<2 x T_{C L K} \quad$, counting is undetermined.
$\mathrm{t} \leqq T_{C L K} \quad$, does not count.
$T_{C L K}$ : Reference clock cycle

| Name and description | Target |
| :---: | :---: |
| <Operation mode using ORG signal and EZ signal> <br> 10h: Operation mode for origin return in the + direction in origin return control <br> 18h: Operation mode for origin return in the - direction in origin return control <br> 12h: Operation mode to escape from the origin position in the + direction by origin return control <br> 1Ah: Operation mode to escape from the origin position in the -direction by origin return control <br> 15h: Operation mode for origin search in the + direction by origin return control <br> 1Dh: Operation mode for origin search in the - direction by origin return control <br> 24h: Operation mode of moves by EZ count in the + direction by origin return control <br> 2Ch: Operation mode of moves by EZ count in the - direction by origin return control | RMD.MOD(6:0) |
| <ORG signal input logic> <br> 0 : Negative logic <br> 1: Positive logic | RENV1.ORGL(7) |
| < Input noise filters of + EL, -EL, +SD, -SD, ORG, ALM, INP and CEMG signals > <br> 0 : Signals with the pulse width of $0.05 \mu$ s or more will react reliably <br> 1: Signals with the pulse width of $3 \mu$ s or less are ignored completely | RENV1.FLTR(26) |
| <Input noise filter of EA, EB and EZ signals> <br> 0 : Signals with the pulse width of $0.05 \mu$ s or more will react reliably. <br> 1: Signals with the pulse width of $0.10 \mu$ s or less will be ignored completely. Signals with the pulse width of $0.15 \mu$ s or more will react reliably. | RENV2.EINF(18) |
| <EZ signal input logic> <br> 0 : Negative logic <br> 1: Positive logic <br> EZ signal counts from OFF to ON. | RENV2.EZL(23) |
| <Origin return method> <br> See "5.4.3.4 RENV3: Environment setting 3". | RENV3.ORM(3:0) |
| <Initial value of EZ signal input count> 0000b (1 time) to 1111b (16 times). | RENV3.EZD(7: 4) |
| <EZ signal input count value> <br> The initial value is in RENV3.EZD bit. | RSPD.EZC(19: 16) |
| <ORG signal input status> <br> 0 : OFF <br> 1: ON | SSTS.SORG(14) |
| $\begin{gathered} \text { <EZ signal input status> } \\ 0: \text { OFF } \\ 1: \text { ON } \end{gathered}$ | RSTS.SEZ(10) |

### 6.8 Servo motor driver 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)

Servo motor drivers can output INP (In-position) signals.

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 this absolute value of the deviation counter becomes less than the set value, the servo motor driver can output 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. If the INP signal is input to INPn pin while waiting for INP signal input (RSTS.CND $=1110$ b), the operation mode can be completed.

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)

However, error and emergency stops (CEMG, CMEMG) due to ALM signals complete the operation mode without 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 pin | RMD.MINP(9) |
| 1: Operation mode complete |  |
| <INP signal input logic> | RENV1.INPL(22) |
| 0: Negative logic |  |
| 1: Positive logic | Rnput noise filters for +EL, -EL, SD, ORG, ALM, INP and CEMG signals > |
| $0:$ Signals with the pulse width of $0.05 \mu$ s or longer will react reliably. |  |
| 1: Signals with the pulse width of $3 \mu$ s or less will be lgnored completely. |  |
| <INP signal input status> | RSTS.SINP(16) |
| 0: OFF |  |
| 1: ON |  |

### 6.8.2 Deviation counter clear (ERC)

An ERC (Error/Deflection counter clear) signal can be input to a servo motor driver.

Servo motor drivers do not stop servo controls until the deviation counter reaches 0 . Therefore, even if command pulses stop, servo motors do not stop immediately. To stop the servo motor immediately upon completion of origin return operation, the deviation counter must be cleared to 0 . To clear this deviation counter to 0 , the servo motor driver can input the ERC signal.

The ON width of the pulse signal (RENV1.EPW) can be selected for the ERC signal to the servo motor driver. If a level signal (RENV1.EPW = 111b) is set for the ON width of the ERC signal, turn it OFF by ERCRST (25h) command. If ERC signal is ON, the servo control of the servo motor driver is turned OFF. Therefore, if you set a level signal, be sure to change the level of ERC signal to OFF by ERCRST (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 ERC signal OFF-width delay time (RENV1.ETW).


By setting the ERC signal output during origin return (RENV1.EROR = 1), an ERC signal can be automatically output when the origin return is completed. For the timing to output an ERC signal, see "5.5.5.1 Origin return in + direction (10h) ".

If ERC signal output at abnormal stop (RENV1.EROE=1) is set, an ERC signal can be automatically output when an abnormal or stop occurs. When decelerate-stopped, the ERC signal is not output.
The target abnormal stop is by the input of $+E L$, -EL, ALM, and CEMG signals and CMEMG (05h) command writing. Even if RMD.MOD bit is 20 h and 28 h , an ERC signal is output when $+E L$ signal and $-E L$ 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 and perform 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 |
| :---: | :---: |
| <ERCn pin output function at the time of immediate stop due to an abnormal stop factor> <br> 0 : Does not output the ERC signal at the time of immediate stop due to an abnormal stop factor. <br> 1: Outputs the ERC signal at the time of immediate stop due to an abnormal stop factor. | RENV1.EROE (10) |
| <Output function of ERCn pin when stopped due to origin return factor> <br> 0 : Does not output the ERC signal when stopped due to origin return factor. <br> 1: Output the ERC signal when stopped due to origin return factor. | RENV1.EROR(11) |
| <ON width of ERC signal> | RENV1.EPW(14: 12) |
| <ERC signal output logic> <br> 0 : Negative logic <br> 1: Positive logic | RENV1.ERCL(15) |
|  | RENV1.ETW $(17,16)$ |
| <ERC signal output status> <br> 0: OFF <br> 1: ON | RSTS.SERC(9) |
| <Emergency stop command> <br> Emergency stop to complete the operation mode. <br> It also cancels the continuous operation by the pre-register. | CMEMG(05h) |
| <ERC signal output> <br> ERC signal is output from ERCn pin. | ERCOUT(24h) |
| <ERC signal reset> <br> Completes waiting for ERC signal ON width and OFF delay to complete (RSTS.CND $=0101 \mathrm{~b}$ ). Resets an ERC signal output. | ERCRST(25h) |

### 6.8.3 Alarm (ALM)

A servo motor driver can output an ALM signal.

Servo motor drivers may experience abnormalities such as overload or overcurrent. To notify that the abnormality has occurred, the servo motor driver can output an ALM signal.

If ALM signal is input to ALMn pin while it is in operation (RSTS.CND $\neq 0000 \mathrm{~b}$ ), it will stop abnormally.
If you set deceleration stop (RENV1.ALMM = 1) in ALM signal input processing, the motor will decelerate and stop in high speed 1 and high speed 2 speed patterns. When ALM signal is ON , the operation mode 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 > <br> 0: Immediate stop <br> 1: Deceleration stop | RENV1.ALMM(8) |
| < ALM signal input logic > <br> 0: Negative logic <br> 1: Positive logic | RENV1.ALML(9) |
| < Input noise filter for +EL, -EL, +SD, -SD, ORG, ALM, INP and CEMG signals > <br> 0 : Signals with the pulse width of $0.05 \mu \mathrm{~s}$ or more will react reliably. <br> 1: Signals with the pulse width of $3 \mu$ s or less will be ignored completely. | RENV1.FLTR(26) |
| < Error interrupt factor (ALM) > <br> 1: Abnormal stop due to ALM signal ON | REST.ESAL(7) |
| < ALM signal input status > <br> 0 : OFF <br> 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 PCL6045BLs, the axes of multiple PCL6045BLs can be started at the same time.
As a preparation, set the CSTA and STA signal input wait (RMD.MSY $=01 \mathrm{~b}$ ) and CSTA signal input (RENV1.PCSM $=0$ ). If you write a start command with this setting, you can make the CSTA and STA signal input waiting state (RSTS.CND $=0010$ b). 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.

The axis where RSTS.CND $=0010 \mathrm{~b}$ and RENV1.PCSM $=0$ is the CSTA signal effective axis. If RENV1.STAM $=0$ is set, the CSTA signal effective axis will start when CSTA=L level is input. This setting will start when the start command is written if CSTA $=\mathrm{L}$ level before the start.

If you set RENV1.STAM $=1$, the operation starts 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).

By writing the CMSTA (06h) command, the CSTA signal (one-shot pulse or level signal) can be output. If you pull up the CSTA pin, you can start the CSTA signal effective axis externally.

The SPSTA (2Ah) command starts only the CSTA signal-enabled axes for which the command has been written. Even if the SPSTA (2Ah) command is written, CSTA signal will not be output. Even if CSTA pin is pulled up, other CSTA signal enabled axes will not be externally started.

Axes with RSTS.CND = 0010b can stop the operation mode by writing a stop command.
<Simultaneous start procedure>
Pull up the CSTA pins of PCL6045BLs that start at the same time together.
Set RMD.MSY = 01b and RENV1.PCSM = 0 on the axes that start at the same time to make them the CSTA signal effective axes. Write the start command and wait for the input of CSTA and STA signals (RSTS.CND $=0010 \mathrm{~b}$ ).

After that, you can start simultaneously by the following two methods:

1. Write the CMSTA (06h) command

Output CSTA signals with the pulse width of 8 CLK signal cycles ( $0.4 \mu \mathrm{~s}$ ) from the CSTA pin.
All PCL6045BLs whose CSTA terminals are connected to each other will start if CSTA signal is input and CSTA signal is enabled. The PCL6045BL that outputs CSTA signal will also start if CSTA signal is input again and CSTA signal is disabled for the axis.

2. Input CSTA signal with the pulse width of 4 CLK signal cycles ( $0.2 \mu \mathrm{~s}$ ) or more to CSTA pin from the outside.

The CSTA signal effective axis starts on all PCL6045BLs whose CSTA terminals are connected together

(Open-drain output)

| Name and description | Target |
| :---: | :---: |
| <Start timing after writing a start command> <br> 01b: If RENV1.PCSM $=0$, the operation starts with CSTA signal ON or SPSTA (2Ah) command. <br> If RENV1.PCSM $=1$, the operation starts with STA signal ON or SPSTA (2Ah) command. | RMD.MSY(19,18) |
| <CSTA signal input specifications> <br> 0: Level trigger <br> 1: Edge trigger (Falling edge) | RENV1.STAM(18) |
| <Input function of CSTA pin and PCSn pin> <br> 0 : CSTA pin input is enabled. <br> PCSn pin reflects the setting of RMD.MPCS bit. <br> 1: CSTA pin input is disabled. <br> PCSn pin also inputs STA signal for starting only the own axis. | RENV1.PCSM(30) |
| <Interrupt request (IRSA)> <br> 1: An interrupt is generated when CSTA signal turns ON (RENV1.PCSM = 0). <br> An interrupt is also generated when STA signal turns ON (RENV1.PCSM = 1). | RIRQ.IRSA(18) |
| <Interrupt factor (ISSA)> <br> 1: CSTA pin is enabled (RENV1.PCSM=0) and CSTA signal is turned ON. <br> Or CSTA pin is disabled (RENV1.PCSM = 1) and STA signal is turned ON. | RIST.ISSA(19) |


| <Operating status> <br> 0010b: Waiting for CSTA signal input. | RSTS.CND(3:0) |
| :--- | :--- |
| <CSTA signal input status> <br> $0:$ OFF <br> $1: ~ O N$ | RSTS.SSTA(5) |
| <CSTA signal output> | CSTA signal is output from CSTA pin. |
| Operation mode can be started on all axes that are waiting for CSTA signal input status |  |
| (RSTS.CND = 0010b). |  | | <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). |

### 6.9.2 Own axis start (STA)

External start can be performed by inputting STA signal (one-shot pulse) to PCSn pin.
By using any PCSn pin, you can start any axis on its own.
As a preparation, set CSTA and STA signal input wait (RMD.MSY $=01 \mathrm{~b}$ ) and STA signal input (RENV1.PCSM $=1$ ).
If you write a start command with this setting, you can enter the CSTA and STA signal input waiting state (RSTS.CND = 0010b). The input logic of STA signal can be selected with RENV1.PCSL bit.

The axis with RSTS.CND $=0010 \mathrm{~b}$ and RENV1.PCSM $=1$ is the STA signal effective axis.
The STA signal effective axis starts when STA signal is turned ON to PCSn pin.
In this case, even if you input CSTA signal to CSTA pin, it will not start.

STA signal input interrupt can be set using Event interrupt request (RIRQ.IRSA).
This can be read using the event interrupt factor (RIST.ISSA).
The input status of STA signal can be read using the Extended status (RSTS.SPCS).

SPSTA (2Ah) command starts only STA signal effective axis to which the command has been written.
Even if SPSTA (2Ah) command is written, CSTA signal and STA signal will not be output.
Even if CSTA pin or PCSn pin is pulled up, other STA signal effective axes will not start externally.

Axes with RSTS.CND = 0010b can stop the operation mode by writing a stop command.

For STA signal, input a one-shot pulse with a pulse width of 4 CLK signal cycles ( $0.2 \mu \mathrm{~s}$ ) or more from the outside to PCSn pin. If the PCSn pin that inputs the one-shot pulse is the STA signal effective axis, the operation will start.

> PCL6045BL


| Name and description | Target |
| :--- | :---: |
| <Start timing after writing a start command> | RMD.MSY(19,18) |
| 01b: If RENV1.PCSM = 0, the operation starts with CSTA signal ON or SPSTA (2Ah) command. |  |
| If RENV1.PCSM = 1, the operation starts with STA signal ON or SPSTA (2Ah) command. |  |
| <STA signal input logic> | RENV1.PCSL(24) |
| 0: Negative logic |  |
| 1: Positive logic |  |


| Name and description | Target |
| :---: | :---: |
| <Input function of CSTA pin and PCSn pin > <br> 0 : Input of CSTA pin is enabled. <br> Setting in RMD.MPCS bit is reflected in PCSn pin. <br> 1: Input of CSTA pin is disabled. <br> STA signal to start only own axis is input in PCSn pin. | RENV1.PCSM(30) |
| <Interrupt request (IRSA)> <br> 1: An interrupt is generated when you turn ON CSTA signal (RENV1.PCSM $=0$ ). <br> An interrupt is also generated when you turn ON STA signal (RENV1.PCSM = 1). | RIRQ.IRSA(18) |
| <Interrupt factor (ISSA)> <br> 1: CSTA pin is enabled (RENV1.PCSM=0) and CSTA signal is turned ON. <br> Or CSTA pin is disabled (RENV1.PCSM = 1) and STA signal is turned ON. | RIST.ISSA(19) |
| <Operation status> <br> 0010b: Waiting for CSTA signal input | RSTS.CND(3:0) |
| <STA signal input status> <br> 0: OFF <br> 1: ON | RSTS.SPCS(8) |
| <Own axis start> <br> Operation mode can be started on any axis that is waiting for CSTA signal input (RSTS.CND $=0010 \mathrm{~b}$ ). CSTA signal is not output from CSTA pin. | SPSTA(2Ah) |

### 6.9.3 Axis selection start (SELn)

By using axis selection (SELn), PCL6045BL can write the same command to multiple axes at the same time. At this time, if one PCL6045BL is used, multiple axes can be started at the same time by writing a start command.

Software example (H8):

| var Address = 0x3: | // Address: COMW of X-axis |
| :--- | :--- |
| var Command = 0x0350; | // Axis selection: Y-axis and X-axis (03h) |
|  | // Command: STAFL (50h) |
| WriteIn16bit (Address, Command); | // Write a command to PCL6045BL |

For details on writing commands, see "5.1.3 Command Write".

### 6.10 External stop / Simultaneous stop

You can use CSTP pin to stop immediately or decelerate-stop with external signals.
You can also stop simultaneously using CSTP pin.

### 6.10.1 Simultaneous stop (CSTP)

You can externally stop by inputting the CSTP signal (one-shot pulse signal) to the CSTP pin.
By using the CSTP pins of multiple PCL6045BLs, each axis of multiple PCL6045BLs can be stopped simultaneously.
As a preparation, set the stop (RMD.MSPE =1) by inputting CSTP signal.
The input logic of CSTP signal cannot be selected and is the negative logic.
The input processing of CSTP signal can be selected with RENV1.STPM bit.

The axis with RMD.MSPE $=1$ is the CSTP signal effective axis.
The CSTP signal effective axis will stop when the falling edge of CSTP signal is input to CSTP pin. If CSTP is at L level before starting, the operation will stop when writing the start command.

The input interrupt of CSTP signal can be read by the error interrupt cause (REST.ESSP).
The input status of CSTP signal can be read by the extended status (RSTS.SSTP).

By writing a CMSTP (07h) command, CSTP signal (one-shot pulse signal) can be output.
By pulling up CSTP pin, CSTP signal effective axis can be stopped externally.
<Procedures for simultaneous stop>
Pull up CSTP pins of PCL6045BL that stop simultaneously together.
Set RMD.MSPE $=1$ to start the axes that will stop simultaneously.
After starting, you can stop simultaneously by the following three methods:

1. Write CMSTP (07h) command

One-shot pulse with the pulse width of 8 cycles $(0.4 \mu \mathrm{~s})$ of the CLK signal is output from CSTP pin.
All PCL6045BLs whose CSTP pins are connected to each other input a one-shot pulse to stop the CSTP signal effective axis.
PCL6045BL that outputs the one-shot pulse also inputs the one-shot pulse again to stop the CSTP signal effective axis.

2. Input a one-shot pulse with the pulse width of 4 cycles ( $0.2 \mu \mathrm{~s}$ ) or more of the CLK signal from the outside to CSTP pin. All PCL6045BLs connected by the CSTP 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 $(0.4 \mu \mathrm{~s})$ of the CLK signal is output from CSTP pin. All PCL6045BLs connected by the CSTP pins input a one-shot pulse to stop the effective axis.

| Name and description | Target |
| :--- | :--- |
| <Input function of CSTP pin > | RMD.MSPE(24) |
| 0: General-purpose input pin |  |
| 1: Stop by inputting CSTP signal |  |
| Input status of CSTP signal is obtained by RSTS.SSTP bit. |  |
| <CSTP pin output function> | RMD.MSPO(25) |
| 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. |  |
| <CSTP signal input processing> | RENV1.STPM(19) |
| 0: Immediate stop | REST.ESSP(8) |
| 1: Deceleration stop | RSTS.SSTP(6) |
| <Error interrupt factor (ESSP)> |  |
| 1: Stops abnormally due to CSTP signal ON. |  |
| <CSTP signal input status> | CMSTP(07h) |
| 1: ON | <Simultaneous stop> |
| CSTP signal is output from CSTP pin. |  |
| Multiple axes in CSTP signal input enabled status can complete the operation mode. |  |

### 6.10.2 Axis selection stop (SELn)

PCL6045BL can write the same command to multiple axes using axis selection (SELn).
At this time, if there is only one PCL6045BL, multiple axes can be stopped simultaneously by writing a stop command.

Software example (H8):

| var Address = 0x3: | // Address: COMW of X-axis |
| :--- | :--- |
| var Command = 0x0349; | // Axis selection: Y-axis and X-axis (03h) |
|  | // Command: STOF (49h) |
| WriteIn16bit (Address, Command); | // Write a command to PCL6045BL |

For details on writing commands, see "5.1.3 Command Write".

### 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 CEMG signal (one-shot pulse signal) to CEMG pin.
By connecting multiple CEMG pins of multiple PCL6045BLs, all axes can be stopped in an emergency.

You cannot start while CEMG $=\mathrm{L}$ level.
The input logic of CEMG signal cannot be selected and is faxed 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.

Internal registers and terminal states cannot be reset in emergency stop.

| Name and description | Target |
| :---: | :---: |
| <Input noise filters of $+E L,-E L, S D, O R G$, ALM, INP and CEMG signals> <br> 0 : Signals with the pulse width of $0.05 \mu$ s or more will react reliably. <br> 1: Signals with the pulse width of $3 \mu$ s or less will be Ignored completely. | RENV1.FLTR(26) |
| <CEMG signal input status> $\begin{aligned} & \text { 0: OFF } \\ & \text { 1: ON } \end{aligned}$ | RSTS.SEMG(7) |
| <Error interrupt factor (ESEM)> <br> 1: Operation stopped abnormally due to CEMG signal ON. | REST.ESEM(9) |
| <Emergency stop> <br> Emergency stop and cancel the operation mode. | CMEMG(05h) |

## C a u tion

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> <br> 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 "Remaining pulse number", see "5.4.2.7 RPLS: Remaining pulse number" and "5.5.2 Positioning control." We explain on Counters 1 to 4 in this section.

### 6.12.1 Counter types and input specifications

You can use the four counters to perform the following functions:

- 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 dedicated for command pulse input.
You can select the input to counters 2 to 4 with RENV3. $\mathrm{Cl} 2, \mathrm{Cl} 3$, and Cl 4 -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 | 28 | 28 | 16 | 28 |
| Default count target | Command pulse <br> (Command position) | Encoder <br> (Mechanical position) | Deviation between <br> command pulse and <br> encode signal | Command pulse <br> (General-purpose) |
| Command pulse <br> (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 |
| $\frac{f_{C L K}}{2}$ | - | - |  |  |

$f_{C L K}$ : Reference clock frequency

| Name and description | Target |
| :---: | :---: |
| <Counting target of counter 2> <br> 00b: EA, EB signal 01b: Command pulse signal 10b: PA, PB signal 11b: Setting prohibited | RENV3.CI2 $(9,8)$ |
| <Counting target of counter 3> <br> 00b: Deviation count between the command pulse signal and EA/ EB signals. <br> 01b: Deviation count between the command pulse signal and PA/ PB signals. <br> 10b: Deviation count between EA/ EB signals and PA/ PB signals. <br> 11b: Setting prohibited. | RENV3.CI3 $(11,10)$ |
| <Counting target of counter 4> <br> 00b: Command pulse signal 01b: EA/EB signals 10b: PA/PB signals $11 \mathrm{~b}: \frac{f_{C L K}}{2}$ 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) 90 -degree phase difference mode (1x) (RENV2.EIM $=00 \mathrm{~b})$


Count-down: EA signal falls when EB signal is at L level.
(2) 90-degree phase difference mode (2x) (RENV2.EIM $=01$ b)


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


Count-up: EA signal rises when EB signal is at L level, EA signal falls when EB signal is at H level, $E B$ signal rises when $E A$ signal is at H level, $E B$ signal falls when $E A$ 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, $E B$ signal rises when EA signal is at L level, EA signal falls when EA signal is at H level.
(4) 2-pulse mode (RENV2.EIM = 11b)


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 $p$ or $r$ 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 PCL6045BL, "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 $\mathrm{EA}, \mathrm{EB}$ and EZ signals> <br> 0 : Signals with the pulse width of $0.05 \mu$ s or more will react reliably. <br> 1: Signals with the pulse width of $0.10 \mu$ s or more will be ignored completely. Signals with the pulse width of $0.05 \mu$ s or more will react reliably. | RENV2.EINF(18) |
| < Input specifications of EA and EB signals> <br> 00b: 90-degree phase difference mode x1 multiplication <br> 01b: 90-degree phase difference mode $\times 2$ multiplication <br> 10b: 90-degree phase difference mode $\times 4$ multiplication <br> 11b: 2-pulse mode | RENV2.EIM $(21,20)$ |
| <Counting direction of EA and EB signals> <br> 0: In 90-degree phase difference mode, counts up when the phase of the EA signal is advanced. In 2-pulse mode, counts up at the rise edge of EA signal. <br> 1: In 90-degree phase difference mode, counts up when the phase of the EB signal is advanced. In 2-pulse mode, counts up at the rise edge of EB signal. | RENV2.EDIR(22) |
| < Input function of $E A$ and $E B$ signals> <br> 0 : Enabled. <br> 1: Disabled. Also not detect input errors | RENV2.EOFF(30) |
| <Error interrupt factor (ESEE)> <br> 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 signals (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 2pulse mode.

| Name and description | Target |
| :---: | :---: |
| <PA and PB signal input noise filter> <br> 0 : Signals with the pulse width of $0.05 \mu \mathrm{~s}$ or more will react reliably. <br> 1: Signals with the pulse width of $0.10 \mu \mathrm{~s}$ or more will be ignored completely. Signals with the pulse width of $0.15 \mu$ s or more will react reliably. | RENV2.PINF(19) |
| <PA and PB signal input specifications> <br> 00b: 90-degree phase difference mode $x 1$ multiplication. <br> 01b: 90-degree phase difference mode $\times 2$ multiplication. <br> 10b: 90-degree phase difference mode $x 4$ multiplication. <br> 11b: 2-pulse mode. <br> For details, see "5.5.3 Pulser control". | RENV2.PIM $(25,24)$ |
| <PA and PB signal counting directions> <br> 0 : Count up when the phase of PA signal is advanced. <br> 1: Count up when the phase of PB signal is advanced. | RENV2.PDIR(26) |
| <PA and PB signal input functions> <br> 0 : Enabled. <br> 1: Disabled. Also not detect input errors. | RENV2.POFF(31) |
| <Error interrupt factor (ESPO)> <br> 1: Abnormal stops due to the buffer counter ( 16 bit) for inputting PA and PB signals overflow. | REST.ESPO(14) |
| <Error interrupt factor (ESPE)> <br> 1: PA or PB signal input error occurred. The operation mode does not stop. | REST.ESPE(17) |

## C a ution

The count target of the 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.

- 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 the counter control command (CUN1R, CUN2R, CUN3R and CUN4R)
- Writing 0 to the registers of counter 1 to 4 (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> <br> 0 : Negative logic <br> 1: Positive logic | RENV1.CLRL(20) |
| <CLR signal input specifications> <br> 0: Edge trigger (OFF to ON) <br> 1: Level trigger | RENV1.CLRM(21) |
| <Clear counter 1 by CLR signal turning ON> <br> 0 : Not clear. <br> 1: Clear. | RENV3.CU1C(16) |
| <Clear counter 2 by CLR signal turning ON > <br> 0 : Not clear. <br> 1: Clear. | RENV3.CU2C(17) |
| <Clear counter 3 by CLR signal turning ON > <br> 0 : Not clear. <br> 1: Clear. | RENV3.CU3C(18) |
| <Clear counter 4 by CLR signal turning ON > <br> 0 : Not clear. <br> 1: Clear. | RENV3.CU4C(19) |
| <Clear counter 1 when the origin is reached in origin return control > <br> 0 : Not clear. <br> 1: Clear. | RENV3.CU1R(20) |
| <Clear counter 2 when the origin is reached in origin return control > <br> 0 : Not clear. <br> 1: Clear. | RENV3.CU2R(21) |


| Name and description | Target |
| :---: | :---: |
| <Clear counter 3 when the origin is reached in origin return control > <br> 0 : Not clear <br> 1: Clear | RENV3.CU3R(22) |
| <Clear counter 4 when the origin is reached in origin return control > <br> 0 : Not clear <br> 1: Clear | RENV3.CU4R(23) |
| <Clear counter 1 to 0 immediately after latching counter 1> <br> 0 : Not clear <br> 1: Clear | RENV5.CU1L(24) |
| <Clear counter 2 to 0 immediately after latching counter 2> <br> 0 : Not clear <br> 1: Clear | RENV5.CU2L(25) |
| <Clear counter 3 to 0 immediately after latching counter 3> <br> 0 : Not clear <br> 1: Clear | RENV5.CU3L(26) |
| <Clear counter 4 to 0 immediately after latching counter 4> <br> 0 : Not clear <br> 1: Clear | RENV5.CU4L(27) |
| <Event interrupt request (IRCL)> <br> 1: An interrupt occurs when CLR signal tuns ON and the count value is cleared. | RIRQ.IRCL (13) |
| <Event interrupt factor (ISCL)> <br> 1: CLR signal turns ON and the count value is cleared. | RIST.ISCL(13) |
| <CLR signal input status> <br> 0 : OFF <br> 1: ON | RSTS.SCLR(13) |
| <Counter 1 control> <br> Clear the count value of counter 1 (RCUN1) to 0 . | CUN1R(20h) |
| <Counter 2 control> <br> Clear the count value of counter 2 (RCUN2) to 0 . | CUN2R(21h) |
| <Counter 3 control> <br> Clear the count value of counter 3 (RCUN3) to 0 . | CUN3R(22h) |
| <Counter 4 control> <br> Clear the count value of counter 4 (RCUN4) to 0. | CUN4R(23h) |

## l mpor tance

When clearing the counter immediately after latching, the counter may show +1 or -1 instead of 0 .
This is not a problem because count signals will be counted after clearing if a count signal is input during 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 met.
- When Comparator 5 condition is met.
- 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 ( $0.1 \mu \mathrm{~s}$ ).

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 ( 29 h ) 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 logic> | RENV1.LTCL(23) |
| 1: Pegative logic |  |
| <Timing to latch counters 1 to 4> | RENV5.LTM(13,12) |
| 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 met. | RENV5.LTFD(14) |
| 11b: When the condition of comparator 5 is met. |  |
| <Latch the current speed instead of counter 3> |  |
| 0: Latches RCUN3 register (counter 3). | RENV5.LTOF(15) |
| 1: Latches RSPD.AS bit (current speed). |  |
| <Latch only at the write timing of LTCH (29h) command> |  |
| 1: Latches even at the timing selected by RENV5.LTM bit. |  |


| Name and description | Target |
| :---: | :---: |
| <Event interrupt request (IRLT)> <br> 1: When RENV5.LTM $=00 \mathrm{~b}$ is set, an interrupt occurs when LTC signal turns ON. | RIRQ.IRLT(14) |
| <Event interrupt request (IROL)> <br> 1: When RENV5.LTM $=01 \mathrm{~b}$ is set, an interrupt occurs when ORG signal turns ON. | RIRQ.IROL(15) |
| <Event interrupt factor (ISLT)> <br> 1: When RENV5.LTM $=00 \mathrm{~b}$ is set, LTC signal turns ON. | RIST.ISLT(14) |
| <Event interrupt factor (ISOL)> <br> 1: When RENV5.LTM $=01 \mathrm{~b}$ is set, ORG signal turns ON. | RIST.ISOL(15) |
| <LTC signal input status> <br> 0 : OFF <br> 1: ON | RSTS.SLTC(14) |
| <ORG signal input status> <br> 0 : OFF <br> 1: ON | SSTS.SORG(14) |
| <Counter latch control> <br> 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:

- It will not count if RMD.MOD $=47 \mathrm{~h}$ is set.
- It will not count if RMD.MCCE $=1$ is set.
- It does not count during backlash correction and slip correction operations if RENV3.CU1B $=0$ is set.

Counter 2 is stopped in two ways as follows:

- It will not count during backlash correction and slip correction operations if RENV3.CU2B $=0$ is set.
- It will not count if RENV3.CU2H = 1 is set.

Counter 3 is stopped in two ways as follows:

- It will not count during backlash correction and slip correction operations if RENV3.CU3B $=0$ is set.
- It will not count if RENV3.CU3H = 1 is set.

Counter 4 is stopped in three ways as follows:

- It counts only uring BSYn = L level if RENV3.BSYC $=1$ is set.

If using in combination with RENV3.CI4 $=11 \mathrm{~b} \frac{f_{C L K}}{2}$ ), the operating time can be controlled by $\operatorname{RCUN} 4 \times 2 \times T_{C L K}$ $f_{C L K}$ : Reference clock frequency $\quad T_{C L K}$ : Reference clock cycle

- It will not count during backlash correction and slip correction operations if RENV3.CU4B $=0$ is set.
- It will not count if RENV3.CU4H = 1 is set.

| Name and description | Target |
| :---: | :---: |
| <Operation mode> <br> 1000111 (47h): Timer operation mode by positioning control | RMD.MOD(6:0) |
| <Count function of counter 1> <br> 0: Counts <br> 1: Does not count. Pulses can be output while counter 1 counting is stopped | RMD.MCCE(11) |
| <Count limit of counter 4> <br> 0: No limit <br> 1: Counts only when BSYn = L level | RENV3.BSYC(14) |
| <Counter 1 counts during backlash correction and slip correction> <br> 0 : Does not count <br> 1: Counts. | RENV3.CU1B(24) |
| <Counter 2 counting during backlash correction and slip correction> <br> 0 : Does not count <br> 1: Counts | RENV3.CU2B(25) |
| <Counter 3 counting during backlash correction and slip correction> <br> 0 : Does not count <br> 1: Counts | RENV3.CU3B(26) |
| <Counter 4 counting during backlash correction and slip correction> <br> 0 : Does not count <br> 1: Counts | RENV3.CU4B(27) |


| Name and description | Target |
| :--- | :--- |
| <Counter 2 counting function> |  |
| 0: Counts. | RENV3.CU2H(29) |
| 1: Does not count. Pulses can be output while counter 2 stops counting. |  |
| <Counter 3 counting function> | RENV3.CU3H(30) |
| 0: Counts. |  |
| 1: Does not count. Pulses can be output while counter 3 stops counting. | RENV3.CU4H(31) |
| <Counter 4 count> |  |
| 1: Does not count. Pulses can be output while counter 4 stops counting. |  |

### 6.13 Comparator

PCL6045BL has a built-in 28-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, comparison condition, and processing method with a comparator when the comparison conditions are met.

### 6.13.1.1 Comparator comparison target

| Comparison target | Comparator 1 <br> (RENV4.C1C) | Comparator 2(RENV4.C2C) | Comparator 3 <br> (RENV4.C3C) | Comparator 4 <br> (RENV4.C4C) | Comparator 5 <br> (RENV5.C5C) |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Counter 1 <br> (RCUN1) | 00 | 00 | 00 | 00 | 000 |
| Counter 2 <br> (RCUN2) | 01 | 01 | 01 | 01 | 001 |
| Counter 3 <br> (RCUN3) | 10 | 10 | 10 | 10 | 010 |
| Counter 4 <br> (RCUN4) | 11 | 11 | 11 | 11 | 011 |
| Remaining pulses number (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 count" and "Counter 2 Ring count", see "6.13.5 Ring count".

| Name and description | Target |
| :---: | :---: |
| <Comparison target of Comparator 1> <br> 00b: RCUN1 <br> 01b: RCUN2 <br> 10b: RCUN3 <br> 11b: RCUN4 <br> If RENV4.C1C $=10 \mathrm{~b}$, compare with the absolute value in RCUN3 register ( 0 to 32,767 ). | RENV4.C1C(1,0) |
| <Comparison target of Comparator 2> <br> 00b: RCUN1 <br> 01b: RCUN2 <br> 10b: RCUN3 <br> 11b: RCUN4 <br> If RENV4.C2C $=10 \mathrm{~b}$, compare with the absolute value in RCUN3 register ( 0 to 32,767 ). | RENV4.C2C(9,8) |


| Name and description | Target |
| :---: | :---: |
| <Comparison target of Comparator 3> <br> 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 <br> If RENV4.C3C $=10 \mathrm{~b}$, compare with the absolute value in RCUN3 register ( 0 to 32,767 ). | RENV4.C3C(17,16) |
| <Comparison target of Comparator 4> <br> 00b: RCUN1 01b: RCUN2 10b: RCUN3 11b: RCUN4 <br> If RENV4.C4C $=10 \mathrm{~b}$, compare with the absolute value in RCUN3 register ( 0 to 32,767 ). | RENV4.C4C $(25,24)$ |
| <Comparison target of Comparator 5> <br> 000b: RCUN1 <br> 001b: RCUN2 <br> 010b: RCUN3 <br> 011b: RCUN4 <br> 100b: RPLS (number of remaining pulses) <br> 101b: RSPD.AS (current speed) <br> If RENV5.C5C $=010 \mathrm{~b}$, 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 | Comparator 1 <br> (RENV4.C1S) | Comparator 2 <br> (RENV4.C2S) | Comparator 3 <br> (RENV4.C3S) | Comparator 4 <br> (RENV4.C4S) | Comparator 5 <br> (RENV5.C5S) |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Always un-met | 000 | 000 | 000 | 0000 | 000 |
| Comparison target $=$ ${ }^{*} 1$ | 001 | 001 | 001 | 0001 | 001 |
| Comparison target $=$ <br> *2 | 010 | 010 | 010 | 0010 | 010 |
| Comparison target $=$ <br> *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 | - | - | - | 1000 | - |
| IDX signal output <br> *2 | - | - | - | 1001 | - |
| IDX signal output | - | - | - | 1010 | - |
| Counter 1 <br> Ring count | $\begin{gathered} 001 \\ (\text { RENV4.C1RM = 1) } \end{gathered}$ | - | - | - | - |
| Counter 2 <br> Ring count | - | $\begin{gathered} 001 \\ (\text { RENV4.C2RM = 1) } \end{gathered}$ | - | - | - |

* 1 The condition is met regardless of the counting direction.
* 2 The condition is met only when counting up.
* 3 The condition is met only when counting down.

For "+SL, -SL signal output (software limit)", see "6.13.2 Software limit".
For "IDX signal output", see "6.13.4 Index output".
For "Counter 1 Ring Count" and "Counter 2 Ring Count", see "6.13.5 Ring count".

| Name and description | Target |
| :---: | :---: |
| <Comparator 1 comparison conditions> <br> 001b: RCMP1 = Comparison target <br> (Regardless of counting direction) <br> 010b: RCMP1 = Comparison target <br> (Only when counting up) <br> 011b: RCMP1 = Comparison target <br> (Only when counting down) <br> 100b: RCMP1 > Comparison target <br> 101b: RCMP1 < Comparison target <br> 110b: +side software limit (RCMP1 <RCUN1) <br> Also set RENV4.C1C $=00 \mathrm{~b}$ <br> Others: Comparison conditions are always un-met | RENV4.C1S(4:2) |
| <Comparator 2 comparison conditions> <br> 001b: RCMP2 = Comparison target. <br> (Regardless of counting direction) <br> 010b: RCMP2 = Comparison target. <br> (Only when counting up) <br> 011b: RCMP2 = Comparison target. <br> (Only when counting down) <br> 100b: RCMP2 > Comparison target. <br> 101b: RCMP2 < Comparison target. <br> 110b: -Side software limit (RCMP2> RCUN1). <br> Also set RENV4.C2C $=00 \mathrm{~b}$. <br> Others: Comparison conditions are always un-met. | RENV4.C2S(12:10) |


| Name and description | Target |
| :---: | :---: |
| <Comparator 3 comparison conditions> <br> 001b: RCMP3 = Comparison target. <br> (Regardless of counting direction) <br> 010b: RCMP3 = Comparison target. <br> (Only when counting up) <br> 011b: RCMP3 = Comparison target. <br> (Only when counting down) <br> 100b: RCMP3 > Comparison target. <br> 101b: RCMP3 < Comparison target. <br> 110b: Setting is prohibited. <br> Others: Comparison conditions are always un-met. | RENV4.C3S(20:18) |
| <Comparator 4 comparison conditions> <br> 0001b: RCMP4 = Comparison target. <br> (Regardless of counting direction) <br> 0010b: RCMP4 = Comparison target. <br> (Only when counting up) <br> 0011b: RCMP4 = Comparison target. <br> (Only when counting down) <br> 0100b: RCMP4 > Comparison target. <br> 0101b: RCMP4 < Comparison target. <br> 0111b: Comparison condition is always not met. <br> 1000b: IDX signal is output under the comparison condition of RENV4.IDXM bits. <br> (Regardless of counting direction) <br> 1001b: IDX signal is output under the comparison condition of RENV4.IDXM bits. <br> (Only when counting up) <br> 1010b: IDX signal is output the under the comparison condition of RENV4.IDXM bits. <br> (Only when counting down) <br> Others: Comparison conditions are always un-met. <br> When using RENV4.C4S $=1000 \mathrm{~b}, 1001 \mathrm{~b}$ and 1010 b , also set RENV4.C4C $=11 \mathrm{~b}$. <br> In this case, when using RENV4.IDXM = 1, set a positive value to RCMP4. | RENV4.C4S(29: 26 ) |


| Name and description | Target |
| :--- | :--- |
| <Comparator 5 comparison conditions> | RENV5.C5S(5: 3) |
| 001b: RCMP5 = Comparison target. |  |
| (Regardless of counting direction) |  |
| 010b: RCMP5 = Comparison target. |  |
| (Only when counting up) |  |
| 011b: RCMP5 = Comparison target. |  |
| (Only when counting down) |  |
| 100b: RCMP5 > Comparison target. |  |
| 101b: RCMP5 < Comparison target. |  |
| Others: Comparison conditions are always un-met. |  |

### 6.13.1.3 Processing method when comparison conditions are met

| Processing method | Comparator 1 <br> (RENV4.C1D) | Comparator 2 <br> (RENV4.C2D) | Comparator 3 <br> (RENV4.C3D) | Comparator 4 <br> (RENV4.C4D) | Comparator 5 <br> (RENV5.C5D) |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 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 |
| :---: | :---: |
| <Processing when the condition of Comparator 1 is met> <br> 00b: No action. It can be used to output INT signals and CP1 signals, and to start internal synchronization. <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. <br> If RENV4.C1S $=110 \mathrm{~b}$ is set, the operation will stop immediately even if RENV4.C1D $=00 \mathrm{~b}, 11 \mathrm{~b}$ is set. | RENV4.C1D 6,5 ) |
| <Processing when the condition of Comparator 2 is met> <br> 00b: No action. It can be used to output INT signals and CP2 signals, and to start internal synchronization. <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. <br> If RENV4.C2S $=110 \mathrm{~b}$ is set, the operation will stop immediately even if RENV4.C2D $=00 \mathrm{~b}, 11 \mathrm{~b}$ is set. | RENV4.C2D $(14,13)$ |


| Name and description | Target |
| :---: | :---: |
| <Processing when the condition of Comparator 3 is met> <br> 00b: No action. It can be used to output INT signals and CP3 signals, and to start internal synchronization. <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. | RENV4.C3D $(22,21)$ |
| <Processing when the condition of Comparator 4 is met> <br> 00b: No action. It can be used to output INT signals and CP4 signals, and to start internal synchronization. <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. | RENV4.C4D $(31,30)$ |
| <Processing when the condition of Comparator 5 is met> <br> 00b: No action. It can be used to output INT signals and CP5 signals, and to start internal synchronization. <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. | RENV5.C5D $(7,6)$ |
| <Event interrupt request (IRC1)> <br> 1: An interrupt is generated when the comparison condition of comparator 1 is met. <br> (MSTS.SCP1 changed from 0 to 1 ) | RIRQ.IRC1(8) |
| <Event interrupt request (IRC2)> <br> 1: An interrupt is generated when the comparison condition of comparator 2 is met. <br> (MSTS.SCP2 changed from 0 to 1 ) | RIRQ.IRC2(9) |
| <Event interrupt request (IRC3)> <br> 1: An interrupt is generated when the comparison condition of comparator 3 is met. (MSTS.SCP3 changed from 0 to 1) | RIRQ.IRC3(10) |
| <Event interrupt request (IRC4)> <br> 1: An interrupt is generated when the comparison condition of comparator 4 is met. <br> (MSTS.SCP4 changed from 0 to 1 ) | RIRQ.IRC4(11) |
| <Event interrupt request (IRC5)> <br> 1: An interrupt is generated when the comparison condition of comparator 5 is met. (MSTS.SCP5 changed from 0 to 1 ) | RIRQ.IRC5(12) |
| <Event interrupt factor (ISC1)> <br> 1: The comparison condition of comparator 1 is met. <br> (MSTS.SCP1 changed from 0 to 1 ) | RIST.ISC1(8) |
| <Event interrupt factor (ISC2)> <br> 1: The comparison condition of comparator 2 is met. <br> (MSTS.SCP2 changed from 0 to 1 ) | RIST.ISC2(9) |


| Name and description | Target |
| :---: | :---: |
| <Event interrupt factor (ISC3)> <br> 1: The comparison condition of Comparator 3 is met. <br> (MSTS.SCP3 changed from 0 to 1 ) | RIST.ISC3(10) |
| <Event interrupt factor (ISC4)> <br> 1: The comparison condition of Comparator 4 is met. <br> (MSTS.SCP4 changed from 0 to 1 ) | RIST.ISC4(11) |
| <Event interrupt factor (ISC5)> <br> 1: The comparison condition of Comparator 5 is met. <br> (MSTS.SCP5 changed from 0 to 1 ) | RIST.ISC5(12) |
| <Main status (SCP1)> <br> 0 : The comparison condition of comparator 1 is not met. <br> 1: The comparison condition of comparator 1 is met. | MSTS.SCP1(8) |
| <Main status (SCP2)> <br> 0 : The comparison condition of comparator 2 is not met. <br> 1: The comparison condition of comparator 2 is met. | MSTS.SCP2(9) |
| <Main status (SCP3)> <br> 0 : The comparison condition of comparator 3 is not met. <br> 1: The comparison condition of comparator 3 is met. | MSTS.SCP3(10) |
| <Main status (SCP4)> <br> 0 : The comparison condition of comparator 4 is not met. <br> 1: The comparison condition of comparator 4 is met. | MSTS.SCP4(11) |
| <Main status (SCP5)> <br> 0 : The comparison condition of comparator 5 is not met. <br> 1: The comparison condition of comparator 5 is met. | MSTS.SCP5(12) |
| <P3n pin I/O function> <br> 10b: Outputs the CP1 signal as a negative logic level signal while the comparator 1 condition is met. <br> 11b: Outputs the CP1 signal as a positive logic level signal while the comparator 1 condition is met. | RENV2.P3M $(7,6)$ |
| <P4n pin I/O function> <br> 10b: Outputs the CP2 signal as a negative logic level signal while the comparator 2 condition is met. <br> 11b: Outputs the CP2 signal as a positive logic level signal while the comparator 2 condition is met. | RENV2.P4M $(9,8)$ |
| <P5n pin I/O function> <br> 10b: Outputs the CP3 signal as a negative logic level signal while the comparator 3 condition is met. <br> 11b: Outputs the CP3 signal as a positive logic level signal while the comparator 3 condition is met. | RENV2.P5M $(11,10)$ |
| <P6n pin I/O function> <br> 10b: Outputs the CP4 signal as a negative logic level signal while the comparator 4 condition is met. <br> 11b: Outputs the CP4 signal as a positive logic level signal while the comparator 4 condition is met. | RENV2.P6M $(13,12)$ |


| Name and description | Target |
| :--- | :--- |
| <P7n pin I/O function> | RENV2.P7M(15,14) |
| 10b: Outputs the CP5 signal as a negative logic level signal while the comparator 5 condition is met. |  |
| 11b: Outputs the CP5 signal as a positive logic level signal while the comparator 5 condition is met. |  |
| <Output timing of internal synchronization signal> | RENV5.SYO(19: 16) |
| 0001b: When comparator 1 condition is met 0010b: When comparator 2 condition is met |  |
| 0011b: When comparator 3 condition is met $\quad$ 0100b When comparator 4 condition is met |  |
| 0101b: When comparator 5 condition is met |  |

### 6.13.2 Software limit

You can use software limits with Comparator 1 (RCMP1) and Comparator 2 (RCMP2).
If the condition of + SL (Comparator 1) is met while operating in the +direction, the operation will stop abnormally.
If the condition of -SL (Comparator 2 ) is met 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.

Select the stop method (RENV4.C1D, C2D) of immediate stop or deceleration stop.
If you select deceleration stop, the operation will stop after passing through +SL position or -SL position.
If the condition of $+S L$ is met, the operation does not start in the +direction. If that of $-S L$ is met, it does not start in the -direction.

## Setting Example:

RENV4 $=00003838 \mathrm{~h}$ : 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 to comparator 1 (immediate stop when comparison target is $+100,001$.)
RCMP2 $=-100,000$ : Set the - side software limit value to comparator 2 (immediate stop when comparison target is $-100,001$.)


Abnormal stop due to + SL and $-S L$ can be read by the error interrupt cause (REST.ESC1, ESC2).
Condition met status of + SL and $-S L$ can be read by the main status (MSTS.SCP1, SCP2).

| Name and description | Target |
| :---: | :---: |
| <Comparator 1 comparison target> <br> 00b: RCUN1 <br> 01b: RCUN2 <br> 11b: RCUN4 | RENV4.C1C (1,0) |
| < Comparator 1 comparison conditions> <br> 110b: +side software limit (RCMP1 < comparison target) | RENV4.C1S(4: 2) |
| <Processing when the condition of Comparator 1 is met> <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> If RENV4.C1S $=110 \mathrm{~b}$ is set, the operation will stop immediately even if RENV4.C1D $=00 \mathrm{~b}, 11 \mathrm{~b}$ is set. | RENV4.C1D 6,5 ) |
| <Comparator 1 comparison target> <br> 00b: RCUN1 01b: RCUN2 <br> 11b: RCUN4 | RENV4.C2C(9,8) |


| Name and description | Target |
| :---: | :---: |
| <Comparator 2 comparison conditions> <br> 110b: -Side software limit (RCMP2 > comparison target) | RENV4.C2S(12:10) |
| <Processing when the condition of Comparator 2 is met> <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> If RENV4.C2S $=110 \mathrm{~b}$ is set, the operation will stop immediately even if RENV4.C2D $=00 \mathrm{~b}, 11 \mathrm{~b}$ is set. | RENV4.C2D $(14,13)$ |
| <Error interrupt factor (CP1 / +SL)> <br> 1: An abnormal stop occurred because the comparison condition of Comparator 1 is met. (Including stop by +SL ) | REST.ESC1(0) |
| <Error interrupt factor (CP2 / -SL)> <br> 1: An abnormal stop occurred because the comparison condition of Comparator 2 is met. (Including stop by -SL) | REST.ESC2(1) |
| <Met status of CP1> <br> 0 : The comparison condition of comparator 1 is not met. <br> 1: The comparison condition of comparator 1 is met. | MSTS.SCP1(8) |
| <Met status of CP2> <br> 0 : The comparison condition of comparator 2 is not met. <br> 1 : The comparison condition of comparator 2 is met. | 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 met.

Match the resolution of the encoder to the resolution of a stepping motor. For a 200 spr stepping motor, use a 200 ppr encoder. (spr: steps per rotation. cpr: counts 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 $=00360000 \mathrm{~h}$ :Set the comparison target of Comparator 3 to counter 3.
Comparator 3 Comparison value < Comparison target is set to stop immediately.
RCMP3 = 32: $\quad$ Set the maximum deviation tolerance to 32.

If the 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> | RENV2.EIM(21,20) |
| 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. | Ror details, see "6.12.1 Counter type and input specifications". |
| <Counting direction of EA and EB signals> | RENV2.EDIR(22) |
| 1: Count up when the phase of EA signal is advanced. |  |
| <Comparator 1 comparison target> | RENV4.C1C(1,0) |
| 10b: RCUN3 (RCUN3) |  |
| If RENV4.C1C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767). |  |


| Name and description | Target |
| :---: | :---: |
| <Comparator 2 comparison target> <br> 10b: RCUN3 (RCUN3) <br> If RENV4.C2C $=10 \mathrm{~b}$, compare with the absolute value of RCUN3 register ( 0 to 32,767 ). | RENV4.C2C(9,8) |
| <Comparator 3 comparison target> <br> 10b: RCUN3 (RCUN3) <br> If RENV4.C3C $=10 \mathrm{~b}$, compare with the absolute value of RCUN3 register ( 0 to 32,767 ). | RENV4.C3C $(17,16)$ |
| <Comparator 4 comparison target> <br> 10b: RCUN3 (RCUN3) <br> If RENV4.C4C $=10 \mathrm{~b}$, compare with the absolute value of RCUN3 register ( 0 to 32,767 ). | RENV4.C4C $(25,24)$ |
| <Comparator 5 comparison target> <br> 010b: RCUN3 (RCUN3) <br> If RENV5.C5C = 10b, compare with the absolute value of RCUN3 register (0 to 32,767). | RENV5.C5C(2:0) |
| <Comparator 1 comparison conditions> <br> 101b: RCMP1 < Comparison target | RENV4.C1S(4:2) |
| <Comparator 2 comparison conditions> <br> 101b: RCMP2 < Comparison target | RENV4.C2S(12:10) |
| <Comparator 3 comparison conditions> <br> 101b: RCMP3 < Comparison target | RENV4.C3S(20:18) |
| <Comparator 4 comparison conditions> <br> 0101b: RCMP4 < Comparison target | RENV4.C4S(29:16) |
| <Comparator 5 comparison conditions> <br> 101b: RCMP5 < Comparison target | RENV5.C5S(5:3) |
| <Processing when the condition of Comparator 1 is met> <br> 00b: No action. It can be used to output INT signals and CP1 signals, and to start internal synchronization. <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. | RENV4.C1D 6,5 ) |
| <Processing when the condition of Comparator 2 is met> <br> 00b: No action. It can be used to output INT signals and CP2 signals, and internal synchronization start. <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. | RENV4.C2D $(14,13)$ |
| <Processing when the condition of Comparator 3 is met> <br> 00b: No action. It can be used to output INT signals and CP3 signals, and internal synchronization start <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. | RENV4.C3D $(22,21)$ |


| Name and description | Target |
| :---: | :---: |
| <Processing when the condition of Comparator 4 is met> <br> 00b: No action. It can be used to output INT signals and CP4 signals, and internal synchronization start. <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. | RENV4.C4D $(31,30)$ |
| <Processing when the condition of Comparator 5 is met> <br> 00b: No action. It can be used to output INT signals and CP5 signals, and internal synchronization start. <br> 01b: Immediate stop. <br> 10b: Deceleration stop. <br> 11b: Bulk override. | RENV5.C5D $(7,6)$ |
| <Counter 3> <br> Register to get counter 3 (deviation). <br> Signed deviation values are obtained. | RCUN3(15: 0) |
| <Comparator 1 comparison value> <br> Register in Comparator 1 to set the comparison value. | RCMP1(27: 0) |
| <Comparator 2 comparison value > <br> Register in Comparator 2 to set the comparison value. | RCMP2(27:0) |
| <Comparator 3 comparison value> <br> Register in Comparator 3 to set the comparison value. | RCMP3(27: 0) |
| <Comparator 4 comparison value> <br> Register in Comparator 4 to set the comparison value. | RCMP4(27: 0) |
| <Comparator 5 comparison value> <br> Register in Comparator 5 to set the comparison value. | RCMP5(27: 0) |
| <Error interrupt factor (ESC1) > <br> 1: An abnormal stop occurred because the comparison condition of Comparator 1 is met. | REST.ESC1(0) |
| <Error interrupt factor (ESC2)> <br> 1: An abnormal stop occurred because the comparison condition of Comparator 2 is met. | REST.ESC2(1) |
| <Error interrupt factor (ESC3)> <br> 1: An abnormal stop occurred because the comparison condition of Comparator 3 is met. | REST.ESC3(2) |
| <Error interrupt factor (ESC4)> <br> 1: An abnormal stop occurred because the comparison condition of Comparator 4 is met. | REST.ESC4(3) |
| <Error interrupt factor (ESC5)> <br> 1: An abnormal stop occurred because the comparison condition of Comparator 5 is met. | REST.ESC5(4) |
| <Error interrupt factor (ESEE)> <br> 1: EA/ EB signal input error has occurred. The operation mode does not stop. | REST.ESEE(16) |
| <Main status (SCP1)> <br> 0 : The comparison condition of comparator 1 is not met. <br> 1: The comparison condition of comparator 1 is met. | MSTS.SCP1(8) |


| Name and description | Target |
| :---: | :---: |
| <Main status (SCP2)> <br> 0 : The comparison condition of comparator 2 is not met. <br> 1: The comparison condition of comparator 2 is met. | MSTS.SCP 2(9) |
| <Main status (SCP3)> <br> 0 : The comparison condition of comparator 3 is not met. <br> 1: The comparison condition of comparator 3 is met. | MSTS.SCP3(10) |
| <Main status (SCP4)> <br> 0 : The comparison condition of comparator 4 is not met. <br> 1: The comparison condition of comparator 4 is met. | MSTS.SCP4(11) |
| <Main status (SCP5)> <br> 0 : The comparison condition of comparator 5 is not met. <br> 1: The comparison condition of comparator 5 is met. | MSTS.SCP5(12) |
| <Counter 3 control command> Clears counter 3 (RCUN3) to 0 . | CUN3R(22h) |

### 6.13.4 Index output

Comparator 4 (RCMP4) can be used to periodically output the index (IDX) signal from CP4n pin.

Set the comparison target of comparator 4 (RENV4.C4C) to 11 b of counter 4 (RCUN4).
Then, select the comparison condition for comparator 4 (RENV4.C4S) from IDX signal output (IDX) 1000b, 1001b, and 1010b.

The count range of RCUN4 register will be from 0 to the RCMP4 register.
Counting down from 0 results in the RCMP4 register.
It becomes $\theta$ when performing counting up from the RCMP4 register.

The setting of RENV4.IDXM bit is enabled 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 met, IDX signal is output at the level.
If RENV4.C4S $=1001 b, 1010 b$ is set for this level output, set RCMP4 register to 2 or more.
If RCMP4 register is less than 2, the direction of counting cannot be determined.

If RENV4.IDXM = 1 is set, IDX signal becomes the pulse output of the logic set in RENV2.P6M bits.
When counting RCUN4 $=0$, IDX signal with 2-cycle width of CLK signal is output as a pulse output.
Even if you write 0 to the RCUN4 register or execute the CUN4R (23h) command, the IDX signal will not be 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: RENV2 $=00002000 \mathrm{~h}$, RENV3 $=00000000 \mathrm{~h}$, RENV4 $=23000000 \mathrm{~h}$ and 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: RENV2 $=00002000 \mathrm{~h}$, RENV3 $=00000000 \mathrm{~h}$, RENV4 $=23800000 \mathrm{~h}$, RCMP4 $=4$.


| Name and description | Target |
| :---: | :---: |
| <P6n pin I/O function> <br> 10b: CP4/IDX signal is output at negative logic when the condition of Comparator 4 is met. <br> 11b: CP4/IDX signal is output at positive logic when the condition of Comparator 4 is met. | RENV2.P6M $(13,12)$ |
| < Counter 4 count target > <br> 00b: Command pulse signals 01b: EA, EB signals 10b: PA, PB signals $11 \mathrm{~b}: \frac{f_{C L K}}{2}$ signal | RENV3.CI4 $(13,12)$ |
| <IDX signal output conditions> <br> 0: Level output with the logic set in RENV2.P6M bit. <br> When RCUN4 = RCMP4 is met, IDX signal is output at the level. <br> 1: Pulse output with the logic set in RENV2.P6M bit. <br> When changing to RCUN4 $=0$, IDX signal of 2-cycle width of the CLK signal is output. | RENV4.IDXM(23) |
| <Comparator 4 comparison target > 11b: RCUN4 | RENV4.C4C $(25,24)$ |
| <Comparator 4 comparison conditions> <br> 1000b: IDX signal is output under the comparison condition of RENV4.IDXM bits. (Regardless of counting direction) <br> 1001b: IDX signal is output under the comparison condition of RENV4.IDXM bits. (Only during count-up) <br> 1010b: IDX signal is output under the comparison condition of RENV4.IDXM bits. (Only during count-down) <br> When using RENV4.C4S $=1000 \mathrm{~b}, 1001 \mathrm{~b}, 1010 \mathrm{~b}$, also set RENV4.C4C $=11 \mathrm{~b}$. <br> In this case, set a positive value in RCMP4 register when using RENV4.IDXM $=1$. | RENV4.C4S(29: 26 ) |

### 6.13.5 Ring count

The ring count function can be used with the ring counters used in a rotary encoder.
The ring counter is an up/down counter that operates in a ring operation from 0 to the set maximum value.
When counting up from the maximum value, the ring counter jumps to 0 .
Conversely, when counting down from 0 , the ring counter will jump to its maximum value.
A ring counter can be used to control the angular position of the rotary table, etc.

Counter 1 (RCUN1) and counter 2 (RCUN2) can perform the ring counting.
If you set RENV4.C1RM=1, RENV4.C1S $=000 \mathrm{~b}$, RENV4.C1C=00b, counter 1 will perform the ring count.
Counter 1 ring counter uses comparator 1 (RCMP1) as the maximum value.
If you set RENV4.C2RM $=1$, RENV4.C2S $=000 b, R E N V 4 . C 2 C=01 b$, counter 2 will perform the ring count.
Counter 2 ring counter uses comparator 2 (RCMP2) as the maximum value.

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: $\mathrm{RMV}=00000010 \mathrm{~h}, \mathrm{RMD}=00000041 \mathrm{~h}, \mathrm{RENV} 4=00000080 \mathrm{~h}, \mathrm{RCUN1}=0$, RCMP1 $=7$.


| Name and description | Target |
| :--- | :--- |
| <Comparator 1 comparison target> <br> 00b: RCUN1 | RENV4.C1C(1,0) |
| <Comparator 1 comparison conditions> <br> 000: The comparison condition is always not met. | RENV4.C1S(4: 2) |
| <When the comparator 1 condition is met> <br> 00b: No action. It can be used to output INT signal and CP1 signal, and internal synchronization | RENV4.C1D(6,5) |
| <Counter 1 ring count>  <br> 1: Performs ring count. RENV4.C1RM(7) <br> <Comparator 2 comparison target> RCUN2 |  |


| Name and description | Target |
| :--- | :--- |
| <Comparator 2 comparison condition> <br> 000: The comparison condition is always not met. | RENV4.C2S(12: 10) |
| <When the condition of comparator 2 is met> <br> 00b: No action. It can be used to output INT signals and CP2 signals, and for internal <br> synchronization start | RENV4.C2D(14,13) |
| <Counter 2 ring count> |  |

## C a u tion

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 met.
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 $\frac{f_{C L K}}{2}$ signal for counter 4 , you can override it all at once for a specified time.

Bulk override overrides all continuous operation pre-register 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)> <br> 0: 2nd pre-register for continuous operation data is undetermined. <br> 1: 2nd pre-register for continuous operation data is determined. | MSTS.SPRF(14) |
| <Processing when the condition of Comparator 1 is met> 11b: Bulk override. | RENV4.C1D $(6,5)$ |
| <Processing when the condition of Comparator 2 is met> 11b: Bulk override. | RENV4.C2D $(14,13)$ |
| <Processing when the condition of Comparator 3 is met> 11b: Bulk override. | RENV4.C3D $(22,21)$ |
| <Processing when the condition of Comparator 4 is met> 11b: Bulk override. | RENV4.C4D $(31,30)$ |
| <Processing when the condition of Comparator 5 is met> 11b: Bulk override. | RENV5.C5D $(7,6)$ |
| <Pre-register control command (PRESHF)> <br> Shifts all the data in pre-registers for continuous operation. | PRESHF(2Bh) |
| <Pre-register control command (PRSET)> <br> 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 PRESHF (2Bh) command, and then starting the next continuous operation. Determine the data 1 for initial operation, the data 1 ' for speed change, and the data 2 for continuous operation, then start.

| No. | Procedures | 2nd pre-register | 1st pre-register | Current register | RSTS. PFM | MSTS. SPRF |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Default status in stopping. | 0 (Undetermined) | 0 (Undetermined) | (Undetermined) | 0 | 0 |
| 2 | Set the waiting for input of CSTA signal (PRMD.MSY = 1). <br> Write Data 1 to the 2nd pre-register. <br> Data 1 is copied to the 1st pre-register. <br> Data 1 is also copied to the current register. | Data 1 <br> (Undetermined) | Data 1 <br> (Undetermined) | Data 1 <br> (Undetermined) | 0 | 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 $=0010 \mathrm{~b}$ ) | Data 1 <br> (Undetermined) | Data 1 <br> (Undetermined) | Data 1 <br> (Determined) | 1 | 0 |
| 4 | Write Data 1' to the 2nd pre-register. Data $1^{\prime}$ is copied to the 1st pre-register. | Data 1' <br> (Undetermined) | Data 1' <br> (Undetermined) | Data 1 <br> (Determined) | 1 | 0 |
| 5 | Write the PRSET (4Fh) command. <br> Determine the overriding data 1 ' in the 1st preregister. | Data 1' <br> (Undetermined) | Data 1' <br> (Determined) | Data 1 <br> (Determined) | 2 | 0 |
| 6 | Cancels waiting for CSTA signal input (PRMD.MSY = $0)$. <br> Write Data 2 to the 2nd pre-register. <br> Data 2 is not copied. | Data 2 <br> (Undetermined) | Data 1' <br> (Determined) | Data 1 <br> (Determined) | 2 | 0 |
| 7 | Write the start command 2 for the continuous operation. <br> Determine the Data 2 for the continuous operation of the 2nd pre-register. <br> Input CSTA signal. <br> Start the operation with Data 1 and Start command 1. | Data 2 <br> (Determined) | Data 1' <br> (Determined) | Data 1 <br> (Determined) | 3 | 1 |
| 8 | Write PRESHF (2Bh) command. <br> Data 1' is copied to the current register. Data 2 is copied to the 1st pre-register. Operation continues with Data 1'. <br> The speed pattern remains at Start command 1. The 2nd pre-register remains determined. | Data 2 <br> (Determined) | Data 2 <br> (Determined) | Data 1' <br> (Determined) | 3 | 1 |
| 9 | The operation mode of Data 1' is completed. Data 2 is copied to the current register. <br> The operation starts with Data 2 and Start command 2. <br> The $2^{\text {nd }}$ and $1^{\text {st }}$ pre-registers become undetermined. | Data 2 <br> (Undetermined) | Data 2 <br> (Undetermined) | Data 2 <br> (Determined) | 1 | 0 |
| 10 | The operation mode of data 2 is completed. <br> The current register becomes undetermined. <br> Since there is no determined register, the continuous operation is completed. | Data 2 <br> (Undetermined) | Data 2 <br> (Undetermined) | Data 2 <br> (Undetermined) | 0 | 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 met, 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 | $2^{\text {nd }}$ pre-register | $1^{\text {st }}$ pre-register | Current register | $\begin{aligned} & \text { RSTS. } \\ & \text { PFM } \end{aligned}$ | MSTS. SPRF |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Set to wait for CSTA signal input (PRMD.MSY =01b). <br> Set incremental operation for positioning (PRMD.MOD=41h). <br> Set PRMV=1000. <br> Set the Data 1 for the initial operation in the remaining registers. <br> Write the STAUD (53h) command. <br> Set the Data 1' for overriding. <br> Write PRSET (4Fh) command. <br> Set PRMD.MSY $=0$ and $P R M D . M O D=41 \mathrm{~h}$. <br> Cancels waiting for CSTA signal input (PRMD.MSY = 00b). <br> Set increment movement for positioning control (PRMD.MOD $=41 \mathrm{~h})$. <br> Set data 2 for continued operation in the remaining registers. <br> Write the STAUD (53h) command. <br> Set RENV5.C5S $=001 \mathrm{~b}$ and RENV5.C5D $=11 \mathrm{~b}$. (Override all at once after RCMP5 = RCUN1) Set PRCP5 = 1500. <br> RCMP5 register is determined. <br> Set RCUN1 $=0$. <br> Write SPSTA (2Ah) command to start. | Data 2 <br> (Determined) | Data 1' <br> (Determined) | Data 1 <br> (Determined) | 3 | 1 |
| 2 | It will stop when RCUN1=1000 (RPLS=0). 1st pre-register is shifted to the current register. The 2nd pre-register shifts to 1st preregister. <br> Since there is no start command, the override ends. <br> Data 2 shifts to the current register. <br> The continuous operation starts | Data 2 <br> (Undetermined) | Data 2 <br> (Undetermined) | Data 2 <br> (Determined) | 1 | 0 |
| 3 | Stops at RCUN1 $=2000($ RPLS $=0)$. <br> The current register becomes undetermined. <br> Since there is no determined register, the continuous operation is completed. | Data 2 <br> (Undetermined) | Data 2 <br> (Undetermined) | Data 2 <br> (Undetermined) | 0 | 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 met, the pre-register is also
shifted. If there is no continuous operation data left in the pre-register, only shifting is performed.

| No. | Procedures | 2nd pre-register | 1st pre-register | Current register | RSTS. <br> PFM | MSTS. SPRF |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Set to wait for CSTA signal input (PRMD.MSY =01b). <br> Set incremental operation for positioning (PRMD.MOD $=41 \mathrm{~h}$ ). <br> Set $P R M V=1000$. <br> Set the Data 1 for the initial operation in the remaining registers. <br> Write the STAUD (53h) command. <br> Set the Data 1' for override 1. <br> Write PRSET (4Fh) command. <br> Set the Data 1" for override 2. Write PRSET (4Fh) command. <br> Set RENV5.C5S $=001 \mathrm{~b}$ and RENV5.C5D $=$ 11b. <br> (Bulk override after RCMP5=RCUN1) <br> Set PRCP5 $=1500$. <br> RCMP5 register is determined. <br> Set RCUN1 $=0$. <br> Write SPSTA (2Ah) command to start. | Data 1" <br> (Determined) | Data 1' <br> (Determined) | Data 1 <br> (Determined) | 3 | 1 |
| 2 | Operation will stop at RCUN1 $=1000$ (RPLS $=$ $0)$. <br> Data 1' is shifted to current register. <br> Data 1 " is shifted to 1 st pre-register. <br> Since there is no start command, Data $1^{\prime}$ is complete. <br> 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" <br> (Undetermined) | Data 1" <br> (Undetermined) | Data 1' <br> (Undetermined) | 0 | 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 $1^{\prime \prime}$ 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 | 2nd pre-register | 1st pre-register | Current register | RSTS. PFM | RSTS. PFC |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 4 | Operation will stop at RCUN1 $=1000$ (RPLS = 0). | Data 1" <br> (Undetermined) | Data 1" <br> (Undetermined) | Data 1" <br> (Undetermined) | 0 | 0 |
|  |  | Comparison B <br> (Undetermined) | Comparison B <br> (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 $=01 \mathrm{~b}$ is set, the number of pulses set in RENV6.BR bit 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 the experiment with an actual machine.

| Name and description | Target |
| :---: | :---: |
| <Counter 1 during backlash correction > <br> 0 : Does not count. <br> 1: Count. | RENV3.CU1B(24) |
| <Counter 2 during backlash correction > <br> 0 : Does not count. <br> 1: Count. | RENV3.CU2B(25) |
| <Counter 3 during backlash correction > <br> 0 : Does not count. <br> 1: Count. | RENV3.CU3B(26) |
| <Counter 4 during backlash correction > <br> 0 : Does not count. <br> 1: Count. | RENV3.CU4B(27) |
| <Backlash correction amount> <br> The setting range is 0 to 4,095 . | RENV6.BR(11:0) |
| <Function to correct the feed amount> <br> 00b: Does not correct feed amount. <br> 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 $=10 \mathrm{~b}$ 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.
However, in a circular interpolation operation, slip compensation cannot be performed.
With RENV3.CU1B, CU2B, CU3B and 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 the experiment with an actual machine.

| Name and description | Target |
| :---: | :---: |
| <Counter 1 during slip correction> <br> 0 : Does not count. <br> 1: Count. | RENV3.CU1B(24) |
| <Counter 2 during slip correction> <br> 0 : Does not count. <br> 1: Count. | RENV3.CU2B(25) |
| <Counter 3 during slip correction> <br> 0 : Does not count. <br> 1: Count. | RENV3.CU3B(26) |
| <Counter 4 during slip correction> <br> 0 : Does not count. <br> 1: Count. | RENV3.CU4B(27) |
| <Slip correction amount> <br> The setting range is 0 to 4,095 . | RENV6.BR(11: 0) |
| <Function to correct the feed amount> <br> 00b: Does not correct the feed amount. 10b: Slip correction. | RENV6.ADJ $(13,12)$ |

### 6.16 Vibration suppression

You can use this function in all positioning operation modes other than the Timer.
Immediately after the operation mode is completed, PCL6045BL 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.

When the operation mode ends due to a stop command or an error factor, vibration suppression pulses are not output.

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 the experiment with an actual machine.

| Name and description | Target |
| :--- | :---: |
| <Cycle of reverse rotation pulse> <br> The cycle of a reverse rotation pulse is set at $\times 32$ cycles of the CLK signal. <br> 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 $\times 32$ cycles of the CLK signal. | RENV7.FT(31:16) |
| The setting range is 0 to $65,535$. |  |

### 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 when the specified axis stops

The start timing can be set by (RMD.MSY = 11b) when the specified axis (RMD.MAX) stops.
With this setting, when a start command is written, the operation enters the state waiting for other axes to stop (RSTS.CND = 0100b).

When the specified axis stops, the operation will start.

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 $=0011$ b) as the specified axes to $U$-axis.
3. Write the start command on U-axis.


When one of the specified axes with RMD.MAX bit starts and stops, the condition is met although the remaining axes do not start.


| Name and description | Target |
| :--- | :---: |
| <Start timing after writing the start command> <br> 11b: Starts when the specified axis (RMD.MAX) stops. | RMD.MSY(19,18) |
| <Axis to confirm stop when RMD.MSY = 11b> <br> Example: 0001b: Starts when X-axis stops. <br> 0010b: Starts when Y-axis stops. <br> 0100b: Starts when Z-axis stops. <br> 1000b: Starts when U-axis stops. <br> 0101b: Starts when both X-axis and Z-axis stop. <br> 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 when the specified axis stops, you can select whether to include own axis stop for the condition with RENV2.SMAX bit.

| Name and description | Target |
| :--- | :---: |
| <Functional specifications when RMD.MAX bit specifies its own axis with RMD.MSY = 11b> | RENV2.SMAX(29) |
| 0: Operation will not start if the own axis stops at the end. |  |
| 1: Operation will start although the own axis stops at the end. |  |

### 6.17.1.1.1 Do not include the stop of own axis in the condition

If you do not include the stop of own axis in the condition, you can select a function that does not start when the axis stops at the end (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 $=00 F C 0061 \mathrm{~h}($ PRMD. $M A X=1111 \mathrm{~b}, \mathrm{MSY}=11 \mathrm{~b}, \mathrm{MOD}=61 \mathrm{~h}$ ) 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 ( $R$ MD.MOD $=61 \mathrm{~h}$ ) will start.


If linear interpolation 2 stops before the circular interpolation stops, linear interpolation 1 (RMD.MOD $=61 \mathrm{~h}$ ) cannot start.


Operation example 1-2(RENV2.SMAX = 0) :
Set the X -axis operating time $>\mathrm{Y}$-axis operating time, and set 1 to 4 below.

1. Set $P R M D=00040041 \mathrm{~h}(P R M D . M A X=0000 b, M S Y=01 b, M O D=41 \mathrm{~h})$ to $X$-axis and $Y$-axis.
2. Write a start command to X -axis and Y -axis.
3. Set $P R M D=003 C 0041 \mathrm{~h}(P R M D . M A X=0011 \mathrm{~b}, \mathrm{MSY}=11 \mathrm{~b}, \mathrm{MOD}=41 \mathrm{~h})$ 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 own axis in the condition

If you include the stop of the own axis in the condition, you can select a function that starts although the own axis stops at the end (RENV2.SMAX = 1) (included from PCL6045B, a conventional PCL product).

Operation example 2-1 (RENV2.SMAX = 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 $P R M D=00 F C 0061 \mathrm{~h}$ (PRMD.MAX $=1111 \mathrm{~b}, \mathrm{MSY}=11 \mathrm{~b}, \mathrm{MOD}=61 \mathrm{~h}$ ) 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 $=61 \mathrm{~h}$ ) will start. This is the same as the function that does not start if the own axis stops at the end (RENV2.SMAX = 0).


Although linear interpolation 2 stops before circular interpolation stops, linear interpolation 1 (RMD.MOD $=61 \mathrm{~h}$ ) will start. This is the function that starts although the own axis stops at the end (RENV2.SMAX $=1$ ).


Operation example 2-2(RENV2.SMAX =1) :
Set the X -axis operating time $>\mathrm{Y}$-axis operating time, and set 1 to 4 below.

1. Set $P R M D=00040041 \mathrm{~h}(P R M D . M A X=0000 b, M S Y=01 b, M O D=41 \mathrm{~h})$ to $X$-axis and $Y$-axis.
2. Write a start command to $X$ and $Y$ axes.
3. Set PRMD $=003 \mathrm{C0041}$ ( $\mathrm{PRMD} . M A X=0011 \mathrm{~b}, \mathrm{MSY}=11 \mathrm{~b}, \mathrm{MOD}=41 \mathrm{~h}$ ) 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 operation axes

In the continuous interpolation that does not change the combination of interpolation operation axes, it is not necessary to set the start by stopping the specified axis. Since all axes stop at the same time, continuous interpolation can be performed simply by setting the continuous operation in the pre-register.

Setting example: (from a circular interpolation by $X$ and $Y$ axes to a linear interpolation 1 by $X$ and $Y$ axes):
Sets a continuous interpolation that does not change the combination of interpolation operation axes.
(Setting in the 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-degree). |
|  | 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-degree) 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 operation axes

In the continuous interpolation that changes the combination of interpolation operation 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 $\mathrm{RMV}=0$.

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 axes to stop.

If the dummy operation is not entered, the continuous operation may stop or the interpolation operation may not stop.
As this is a compatible function with older products, we recommend "6.17.3 Continuous interpolation using circular interpolation dummy operation" for PCL6045BL)

Setting example: (from a circular interpolation by $X$ and $Y$ axes to a linear interpolation by $X$ and $Y$ axes 1 and linear interpolation on the X and Z axes)

Set a continuous interpolation to change the combination of interpolation operation axes.
(RENV2.SMAX = 0 and speed control register settings are 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-degree). <br> 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. |
| 2 | PRMV | 10000 | 5000 | 0 | Incremental movement, 10000, 5000 of $X$-axis and Y-axes. <br> Z axis: 0 feed amount . |
|  | PRMD | 004C0061h | 004C0061h | 003C0041h | Linear interpolation 1 by X -axis and Y -axis. <br> Z-axis: dummy operation. |
|  | COMW | 0751h | - | - | STAFH (51h) command to all axes. |
| 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. |

When STEP1 is set, CW circular interpolation (10000 radius, 90-degree) 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 $\mathbf{2}$ to change the interpolation operation axes

In the continuous interpolation that changes the combination of interpolation operation 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. (As this is a compatible function with older products, we recommend "6.17.3 Continuous interpolation using circular interpolation dummy operation" for PCL6045BL)

Setting example: (from a circular interpolation by $X$ and $Y$ axes to a linear interpolation by $X$ and $Y$ axes 1 and linear interpolation on the $X$ and $Z$ axes). Set a continuous interpolation to change the combination of interpolation operation axes. (RENV2.SMAX $=0$ and speed control register settings are 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-degree). <br> 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. |
| 2 | PRMV | 10000 | 5000 | 0 | Incremental movement 10000,5000 of $X$-axis and Y -axis. <br> $Z$ 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. |
| 3-1 | PRMV | 0 | 0 | 0 | Since the combination of interpolation operation axes is changed, feed amount is 0 . |
|  | PRMD | 007C0041h | 007C0041h | 007C0041h | Dummy operations of all axes. |
|  | COMW | 0751h | - | - | STAFH (51h) command to all axes. |
| 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. |

STEP1 is set to start the CW circular interpolation (radius 10000, 90-degree) by X -axis and Y -axis and the Z -axis dummy operation.

STEP1 dummy operation will stop immediately.
STEP2 is set while STEP1 is operating to wait the completion of all axes of STEP1.

STEP1's all axes are completed to start the linear interpolation $1(10000,5000)$ of STEP2 and the dummy operation of $Z$ axis.
STEP2 dummy operation will stop immediately.
STEP3 is set while STEP1 or STEP2 is operating to wait the completion of all axes of STEP2.

STEP2's all axes are completed to start the linear interpolation $1(10000,-5000)$ of STEP3 and the dummy operation of $Y$ axis.

STEP3 dummy operation will stop immediately.
STEP2's all axes are completed to end the continuous interpolation.

The trajectory is the same as "6.17.1.3 Continuous interpolation 1 to change the interpolation operation axes ".

### 6.17.2 Start with internal sync signal

When RMD.MSY $=10 \mathrm{~b}$ is set, RSTS.CND $=0011 \mathrm{~b}$ 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 I/O 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 $=01 \mathrm{~b}$ ).
3. Set the internal synchronization signal output (RENV5y.SYO $=1001 \mathrm{~b}$ ) when an acceleration ends.


Setting example 2:
When the comparator 1 condition of Y -axis is met, X -axis will start.
After you set 1 to 7 below, start $X$-axis and $Y$-axis.

1. Set the start timing of $X$-axis to start using the internal synchronization signal (PRMDx.MSY $=10 \mathrm{~b}$ ).
2. Set the Y -axis internal sync signal (RENV5x.SYI $=01 \mathrm{~b}$ ) to the X -axis internal sync signal.
3. Set the output timing of the $Y$-axis internal synchronization signal to when the comparator 1 condition is satisfied (RENV5y.SYO = 0001b).
4. Set the RCUN1 register (RENV4y.C1C $=00 \mathrm{~b}$ ) as the comparison target for Y -axis comparator 1.
5. Set the comparison condition of Y -axis comparator 1 to be equal to the comparison target (RENV4y.C1S $=001 \mathrm{~b}$ ).
6. Set the processing of $Y$-axis comparator 1 to do nothing (RENV4y.C1D = 00b).
7. Set the $Y$-axis comparator 1 comparison value to 1000 ( $R C M P 1 y=1000$ ).


In this example, setting PRMVy $=2000$ and PRMVx $=1000$ 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> <br> 10b: Start with internal synchronization signal (RENV5.SYI). | RMD.MSY(19,18) |
| <POn pin I/O function> <br> 10b: FUP signal is output at negative logic during acceleration. | RENV2.P0M(1,0) |
| <P1n pin I/O function> <br> 10b: FDW signal is output at negative logic during deceleration. | RENV2.P1M 3 (3) |
| <P3n pin I/O function> <br> 10b: CP1 signal is output with negative logic when the comparator 1 condition is met. <br> 11b: CP1 signal is output with positive logic when the comparator 1 condition is met. | RENV2.P3M 7 (7,6) |
| <P4n pin I/O function> <br> 10b: CP2 signal is output at negative logic when the comparator 2 condition is met. <br> 11b: CP2 signal is output at positive logic when the comparator 2 condition is met. | RENV2.P4M $(9,8)$ |
| <P5n pin I/O function> <br> 10b: CP3 signal is output at negative logic when the comparator 3 condition is met. <br> 11b: CP3 signal is output at positive logic when the comparator 3 condition is met. | RENV2.P5M $(11,10)$ |
| <P6n pin I/O function> <br> 10b: CP4 signal is output at negative logic when the comparator 4 condition is met. <br> 11 b : CP4 signal is output at positive logic when the comparator 4 condition is met. | RENV2.P6M $(13,12)$ |
| <P7n pin I/O function> <br> 10 b : CP5 signal is output at negative logic when the comparator 5 condition is met. <br> 11 b : CP5 signal is output at positive logic when the comparator 5 condition is met. | RENV2.P7M $(15,14)$ |
| $<$ Input target of internal sync signal> <br> 00b: X-axis internal sync signal <br> 01b: Y-axis internal sync signal <br> 10b: $Z$ axis internal sync signal <br> 11b: U-axis internal sync signal | RENV5.SYI 21,20 ) |


| Name and description | Target |
| :---: | :---: |
| Output timing of internal synchronous signal <br> 0001b: When comparator 1 condition is met <br> 0011b: When comparator 3 condition is met <br> 0101b: When comparator 5 condition is met <br> 1000b: When the acceleration starts <br> 1010b: When the deceleration starts <br> Other: No output of internal synchronous signal <br> 0010b: When comparator 2 condition is met <br> 0100b: When comparator 4 condition is met <br> 1001b: When the acceleration ends <br> 1011b: When the deceleration ends | RENV5.SYO(19:16) |
| <Event interrupt request (IRUS)> <br> 1: An interrupt is generated when acceleration is started. <br> (SSTS.SFU bit changed from 0 to 1 ) | RIRQ.IRUS(4) |
| <Event interrupt request (IRUE)> <br> 1: An interrupt is generated when acceleration is completed. <br> (SSTS.SFU bit changed from 1 to 0 ) | RIRQ.IRUE(5) |
| <Event interrupt request (IRDS)> <br> 1: An interrupt is generated when deceleration is started. <br> (SSTS.SFD bit changed from 0 to 1 ) | RIRQ.IRDS(6) |
| <Event interrupt request (IRDE)> <br> 1: An interrupt is generated when deceleration is completed. <br> (SSTS.SFD bit changed from 1 to 0 ) | RIRQ.IRDE(7) |
| <Event interrupt request (IRC1)> <br> 1: An interrupt is generated when the comparison condition of comparator 1 is met. <br> (MSTS.SCP1 changed from 0 to 1 ) | RIRQ.IRC1(8) |
| <Event interrupt request (IRC2)> <br> 1: An interrupt is generated when the comparison condition of comparator 2 is met. (MSTS.SCP2 changed from 0 to 1 ) | RIRQ.IRC2(9) |
| <Event interrupt request (IRC3)> <br> 1: An interrupt is generated when the comparison condition of comparator 3 is met. (MSTS.SCP3 changed from 0 to 1 ) | RIRQ.IRC3(10) |
| <Event interrupt request (IRC4)> <br> 1: An interrupt is generated when the comparison condition of comparator 4 is met. <br> (MSTS.SCP4 changed from 0 to 1 ) | RIRQ.IRC4(11) |
| <Event interrupt request (IRC5)> <br> 1: An interrupt is generated when the comparison condition of comparator 5 is met. <br> (MSTS.SCP5 changed from 0 to 1) | RIRQ.IRC5(12) |
| <Event interrupt factor (ISUS)> <br> 1: Acceleration has started. <br> (SSTS.SFU bit changed from 0 to 1 ) | RIST.ISUS(4) |


| Name and description | Target |
| :---: | :---: |
| <Event interrupt factor (ISUE)> <br> 1: The acceleration has ended. <br> (SSTS.SFU bit changed from 1 to 0 ) | RIST.ISUE(5) |
| <Event interrupt factor (ISDS)> <br> 1: Deceleration has started. <br> (SSTS.SFD bit changed from 0 to 1 ) | RIST.ISDS(6) |
| <Event interrupt factor (ISDE)> <br> 1: The deceleration has ended. <br> (SSTS.SFD bit changed from 1 to 0 ) | RIST.ISDE(7) |
| <Event interrupt factor (ISC1)> <br> 1: The comparison condition of comparator 1 is met. <br> (MSTS.SCP1 changed from 0 to 1 ) | RIST.ISC1(8) |
| <Event interrupt factor (ISC2)> <br> 1: The comparison condition of comparator 2 is met. <br> (MSTS.SCP2 changed from 0 to 1 ) | RIST.ISC2(9) |
| <Event interrupt factor (ISC3)> <br> 1: The comparison condition of Comparator 3 is met. <br> (MSTS.SCP3 changed from 0 to 1 ) | RIST.ISC3(10) |
| <Event interrupt factor (ISC4)> <br> 1: The comparison condition of Comparator 4 is met. <br> (MSTS.SCP4 changed from 0 to 1 ) | RIST.ISC4(11) |
| <Event interrupt factor (ISC5)> <br> 1: The comparison condition of Comparator 5 is met. <br> (MSTS.SCP5 changed from 0 to 1 ) | RIST.ISC5(12) |
| <Motion status> <br> 0011b : Waiting for internal synchronization signal input | RSTS.CND(3:0) |

### 6.17.3 Continuous interpolation using circular interpolation dummy operation

In case of continuous interpolation that changes the combination of interpolation operation axes, the operation mode of circular interpolation dummy (RMD.MOD $=6 \mathrm{Fh}$ ) can be used. The circular interpolation dummy is an operation mode for continuous interpolation with circular interpolation control without setting the RENV2 register or RMD.MAX bit.

By starting at the same time, it will stop at the same time as other circular interpolation controls.
The dummy operation of linear interpolation is positioning control to set the same value as the RMV register of the main axis.

Setting example (from a circular interpolation by $X$ and $Y$ axes to a linear interpolation 1 by $X$ and $Y$ axes as well as a linear interpolation 1 on the X and Z axes):

Set a continuous interpolation to change the combination of interpolation operation axes.
(Speed control register settings are omitted)

| STEP | Write target | X-axis value | $Y$-axis value | Z-axis value | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | PRMV | 10000 | 10000 | 0 | $X$ and $Y$ axes are target coordinates 10000, 10000 (90 degree) <br> $Z$ axis is the target position 0 . |
|  | PPRIP | 10000 | 0 | 0 | Center coordinates 10000, 0 |
|  | PRMD | 00000064h | 00000064h | 0000006Fh | $X$ and $Y$ axes are for a CW circular interpolation. $Z$-axis is for a circular interpolation dummy operation. |
|  | COMW | 0751h | - | - | STAFH (51h) command for all axes |
| 2 | PRMV | 10000 | 5000 | 0 | X-, Y-, and Z-axes are for feed amount of 10000, 5000, 0 . |
|  | PRMD | 00000061h | 00000061h | 00000061h | All axes are for an incremental movement with linear interpolation 1. |
|  | COMW | 0751h | - | - | STAFH (51h) command for all axes. |
| 3 | PRMV | 10000 | 0 | -5000 | X-, Y-, and Z-axes are for feed amount of 10000, 0,5000. |
|  | PRMD | 00000061h | 00000061h | 00000061h | All axes are for an incremental movement with linear interpolation 1. |
|  | COMW | 0751h | - | - | STAFH (51h) command for all axes |

Setting STEP1 starts CW circular interpolation (radius 10000, 90-degree) by $X$ and $Y$ axes and a dummy movement on $Z$ axis.
Z-axis in STEP1 does not output pulses, but it stops immediately at the same time as CW circular interpolation.
If STEP2 is set while STEP1 is in operation, all axes of STEP1 will wait for completion.

When all axes in STEP1 are completed, a linear interpolation $1(10000,5000,0)$ in STEP2 starts.
Z-axis in STEP2 does not output pulses, but it stops immediately at the same time as linear interpolation 1.
If STEP3 is set while STEP1 or STEP2 is in operation, all axes of STEP2 will wait for completion.

When all axes in STEP2 are completed, a linear interpolation 1 (5000, 0, -5000) in STEP3 starts.
Y-axis in STEP3 does not output pulses, but it stops immediately at the same time as linear interpolation 1.
When all axes in STEP3 are completed, continuous interpolation is completed.

The trajectory is the same as "6.17.1.3 Continuous interpolation 1 to change the interpolation operation axes".

### 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 PCL6045BLs, INT pins cannot be wired or connected to each other. (INT $\neq \mathrm{Hi}-\mathrm{Z}$ )

| Name and description | Target |
| :--- | :---: |
| <INT pin output function> | RENV1.INTM(29) |
| $0:$ 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. |  |

### 6.18.1 Error interrupt

The error interrupt factor occurs when only one condition is met.
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 |
| :---: | :---: |
| <Error interrupt factor> <br> For REST register, see "5.4.7.2 REST: Error interrupt factor". | REST register |
| <How to clear the bits of the RIST register and REST register> <br> 0 : Write the read command to each register to clear each register to 0 . <br> Even with the full-address direct access method, each register can be cleared to 0 by writing a read command. <br> 1: Writing read command to each register does not clear the register to 0 . <br> In either case, you can write 1 to the corresponding bit in each register to clear it to 0 . | RENV5.ISMR(23) |
| <Main status (SERR)> <br> 0 : No error interrupt occurred. <br> 1: An error interrupt occurred. L level can be output from INT pin. <br> When all of the bits that are 1 in REST register become 0 , it returns to MSTS.SERR $=0$. | MSTS.SERR(4) |
| <Register Read (REST)> <br> Reads out the contents of REST register to I/O buffer. <br> When RENV5.ISMR is set to 0 , bits that can be read as 1 will be cleared to 0 . | RREST(F2h) |

### 6.18.2 Event interrupt

The event interrupt factor occurs when the condition of RIRQ register is met.
When an event interrupt factor occurs, the corresponding bit in RIST register becomes 1.
When any bit of RIST register is 1 , it becomes MSTS.SINT $=1$, so $L$ level can be output from INT pin.
In RIST register, writing 1 to the corresponding bit clears the bit to 0 .
When setting RENV5.ISMR $=0$, the bits that can be read as 1 by writing the RRIST (F3h) command are cleared to 0 .
When setting RENV5.ISMR = 1, RIST register will not be cleared by writing RRIST (F3h) command.
(The setting of RENV5.ISMR bit also affects REST register)

| Name and description | Target |
| :---: | :---: |
| <Event interrupt factor> <br> For details on RIST register, see "5.4.7.3 RIST: Event interrupt factor". | RIST register |
| <How to clear the bits of RIST register and REST register> <br> 0 : Write the read command to each register to clear the register to 0 . <br> 1: Writing the 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 . | RENV5.ISMR(23) |
| <Main status (SINT)> <br> 0 : No event interrupt occurred. <br> 1: An event interrupt occurred. L level can be output from INT pin. <br> When all of the bits that are 1 in the RIST register become 0 , it returns to MSTS.SINT $=0$. | MSTS.SINT(5) |
| <Register read (RIST)> <br> Reads out the contents of RIST register to I/O buffer. <br> When RENV5.ISMR is set to 0 , bits that can be read as 1 will be cleared to 0 . | 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 when you write SENIR (2Dh) command.
When you set RENV5.MSMR $=0$, MSTS.SENI bit is cleared to 0 even in reading the main status.
When 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 included in the event interrupt factors, but you need check by reading RIST register. If no 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.

When 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> <br> 0 : When RSTS.PFM $=10 \mathrm{~b}$ or $11 \mathrm{~b}, \mathrm{MSTS}$.SENI $=1$ will be set. <br> 1: When RSTS.PFM $=10 \mathrm{~b}$ or $11 \mathrm{~b}, \mathrm{MSTS}$. SENI $=1$ will not be set. <br> 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 $=10 \mathrm{~b}$ or 11b). | RMD.MENI(7) |
| <Functional specifications of MSTS.SENI bit> <br> 0 : Disabled. Keep MSTS.SENI $=0$ by stopping the operation. <br> 1: Enabled. Change to MSTS.SENI = 1 by stopping the operation. | RENV2.IEND(27) |
| <How to clear MSTS.SENI bit and MSTS.SEOR bit> <br> 0 : Cleared automatically if the main status is read. <br> 1: Not cleared automatically even if the main status is read. <br> MSTS.SENI bit can be cleared manually by writing SENIR (2Dh) command. <br> MSTS.SEOR bit can be cleared manually by writing SEORR (2Eh) command. | RENV5.MSMR(25) |
| <Main status (SENI)> <br> 0 : No operation stop interrupt occurs. Or RENV2.IEND $=0$ is set. <br> 1: An operation stop interrupt occurs. L level can be output from INT pin. <br> If RENV5.MSMR $=0$, it returns to 0 within 3 cycles of the CLK signal after reading. <br> 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

General-purpose one-shot signals can be output from the P0 and P1 pins using a general-purpose output bit control command. The output pulse width of the general-purpose one-shot signal is 23 to 25 ms .

P 0 n pin can output a general-purpose one-shot signal by setting RENV2.P0M $=11 \mathrm{~b}$.
When RENV2.P0L = 0 is set, a general-purpose one-shot signal with negative logic is output with PORST (10h) command. General-purpose one-shot signal output of negative logic can be stopped on the way with POSET (18h) command.

When RENV2.P0L = 1 is set, a general-purpose one-shot signal with positive logic is output with P0SET (18h) command. General-purpose one-shot signal output of positive logic can be stopped on the way with P0RST (10h) command.

P1n pin can output a general-purpose one-shot signal by setting RENV2.P1M $=11 \mathrm{~b}$.
When RENV2.P1L = 0 is set, a general-purpose one-shot signal with negative logic is output with P1RST (11h) command. General-purpose one-shot signal output of negative logic can be stopped on the way with P1SET (19h) command. When RENV2.P1L = 1 is set, a general-purpose one-shot signal with positive logic is output with P1SET (19h) command. General-purpose one-shot signal output of positive logic can be stopped on the way with P1RST (11h) command.

| Name and description | Target |
| :---: | :---: |
| <POn pin I/O function> <br> 11b: Outputs a general-purpose one-shot signal. | RENV2.P0M(1,0) |
| <P1n pin I/O function> <br> 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 > <br> 0 : Negative logic. <br> 1: Positive logic. | RENV2.P0L(16) |
| <The logic of a general-purpose one-shot signal that can be output from P1n pin > <br> 0 : Negative logic. <br> 1: Positive logic. | RENV2.P1L(17) |
| <General-purpose output bit control command (P0RST)> <br> Write 0 to OTP0 bit to reset P0n pin to L level. <br> When RENV2.P0M $=11 \mathrm{~b}$ and RENV2.P0L $=0$, a general-purpose one-shot signal with negative logic is output. | P0RST(10h) |
| <General-purpose output bit control command (P1RST)> <br> Write 0 to OTP1 bit to reset P1n pin to L level. <br> When RENV2.P1M $=11 \mathrm{~b}$ and RENV2.P1L $=0$, a general-purpose one-shot signal with negative logic is output. | P1RST(11h) |
| <General-purpose output bit control command (POSET)> <br> Write 1 to OTP0 bit to set P0n pin to H level. <br> When RENV2.P0M = 11b and RENV2.P0L = 1, a general-purpose one-shot signal with positive logic is output. | POSET(18h) |
| <General-purpose output bit control command (P1SET)> <br> Write 1 to OTP1 bit to set P1n pin to H level. <br> 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 | $\mathrm{V}_{\mathrm{DD}}$ | -0.3 to +4.0 | V | - |
| Input voltage | $\mathrm{V}_{\mathrm{I}}$ | -0.3 to +7.0 | V | - |
| Output voltage | $\mathrm{V}_{\mathrm{o}}$ | -0.3 to +7.0 | V | - |
| Output current | Iout | $\pm 30$ | mA | - |
| Storage temperature | $\mathrm{T}_{\mathrm{stg}}$ | -65 to +150 | ${ }^{\circ} \mathrm{C}$ | - |

## C a u t i o n

Regarding the output voltage rating, if a voltage higher than the power supply voltage is applied, a large current may flow.

### 7.2 Recommend operating conditions

| Item | Symbol | Min. | Typ. | Max. | Unit | Remarks |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: |
| Power supply voltage | $\mathrm{V}_{\mathrm{DD}}$ | 3.0 | 3.3 | 3.6 | V | - |
| Input voltage | $\mathrm{V}_{\mathrm{I}}$ | -0.3 | - | +5.8 | V | - |
| Operating Ambient temperature | $\mathrm{T}_{\mathrm{stg}}$ | -40 | - | +85 | ${ }^{\circ} \mathrm{C}$ | $\mathrm{T}_{\mathrm{j}}=-40$ to $+125^{\circ} \mathrm{C}, \theta_{\mathrm{j}-\mathrm{a}}=24^{\circ} \mathrm{C} / \mathrm{W}$ |
| Input rising time | $\mathrm{T}_{\mathrm{r}}$ | - | - | 50 | ns | $10 \%$ to $90 \%$ change time of <br> power supply voltage |
| Input falling time | $\mathrm{T}_{\mathrm{f}}$ | - | - | 50 | ns | $10 \%$ to $90 \%$ change time of <br> power supply voltage |

### 7.3 DC characteristics

| Item | Symbol | Conditions |  | Min. | Max. | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Current consumption | $\mathrm{I}_{\mathrm{DD}}$ | CLK = $20 \mathrm{MHz}, 4$ axes 6.667 Mpps , No load |  | - | 155 | mA |
| Output leakage current | $\mathrm{I}_{\text {LI }}$ | $\mathrm{V}_{\mathrm{IH}}=3.6 \mathrm{~V}, \mathrm{~V}_{\mathrm{IL}}=0 \mathrm{~V}$ | ${ }^{*} 1$ | -1 | 1 | $\mu \mathrm{A}$ |
| Output leakage current | $\mathrm{I}_{0 z}$ | $\mathrm{V}_{\mathrm{OH}}=3.6 \mathrm{~V}, \mathrm{~V}_{\mathrm{OL}}=0 \mathrm{~V}$ | *3 | -1 | 1 | $\mu \mathrm{A}$ |
| High level input voltage | $\mathrm{V}_{\text {IH }}$ | $V_{D D}=3.6 \mathrm{~V}$ |  | 2.0 | 5.8 | V |
| Low level input voltage | $\mathrm{V}_{\text {IL }}$ | $\mathrm{V}_{\mathrm{DD}}=3.0 \mathrm{~V}$ |  | -0.3 | 0.8 | V |
| High level input current | $\mathrm{I}_{\text {IH }}$ | $\mathrm{V}_{\mathrm{IH}}=5.5 \mathrm{~V}, \mathrm{~V}_{\mathrm{DD}}=3.0 \mathrm{~V}$ | ${ }^{*}$ | - | 30 | $\mu \mathrm{A}$ |
| Low level input current | $\mathrm{I}_{\text {IL }}$ | $\mathrm{V}_{\mathrm{IL}}=0 \mathrm{~V}, \mathrm{~V}_{\mathrm{DD}}=3.6 \mathrm{~V}, \mathrm{R}_{\mathrm{PU}}=40 \mathrm{k} \Omega$ | ${ }^{2}$ | -90 | - | $\mu \mathrm{A}$ |
| High level output voltage | $\mathrm{V}_{\mathrm{OH}}$ | $\mathrm{I}_{\mathrm{OH}}=-6 \mathrm{~mA}$ | ${ }^{3}$ | $V_{D D}-0.4$ | - | V |
| Low level output voltage | $\mathrm{V}_{\text {OL }}$ | $\mathrm{I}_{\mathrm{OL}}=6 \mathrm{~mA}$ | ${ }^{*}$ | - | 0.4 | V |
| High level output current | $\mathrm{I}_{\text {OH }}$ | $\mathrm{V}_{\mathrm{OH}}=\mathrm{V}_{\mathrm{DD}}-0.4 \mathrm{~V}$ | *3 | -6 | - | mA |
| Low level output current | $\mathrm{I}_{0}$ | $\mathrm{V}_{\mathrm{OL}}=0.4 \mathrm{~V}$ | *3 | - | 6 | mA |
| Internal pull-up resistance | Rpu | $\mathrm{V}_{\mathrm{I}}=\mathrm{GND}$ | ${ }^{*}$ | 40 | 240 | k $\Omega$ |
| Pin capacity | C | - |  | - | 10 | pF |

As for the sign of current, the positive number indicates the inflow current value, and the negative number indicates the outflow current value.
${ }^{*} 1$ CS, RD, WR, A0, A1, A2, A3, A4, D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, CLK pins.
*2 Input pins and bidirectional pins other than the above.
*3 Output pin and bidirectional pin.

### 7.4 AC characteristics

### 7.4.1 Reference clock



| Item | Symbol | Conditions | Min. | Max. | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Frequency | Fcık | (Recommended frequency 19.6608 MHz ) | - | 20 | MHz |
| Cycle | Tсıк | - | 50 | - | ns |
| H level width | Тскн | - | 20 | - | ns |
| L level width | Tckı | - | 20 | - | 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 $\downarrow$ | TAS | － | 10 | － | ns |
| Address hold time | LS $\uparrow$ | T ${ }_{\text {S }}$ | － | 0 | － | ns |
| CS setup time | LS $\downarrow$ | Tcss | － | 2 | － | ns |
| CS hold time | LS $\uparrow$ | Tscs | － | 2 | － | ns |
| R／W setup time | LS $\downarrow$ | TRWS | － | 4 | － | ns |
| R／W hold time | LS $\uparrow$ | TsRW | － | 2 | － | ns |
| ACK ON delay time | LS $\downarrow$ | Tslakr | $C_{L}=40 \mathrm{pF}$ | $1 \cdot \mathrm{~T}_{\text {CLK }}$ | 5•T ${ }_{\text {CLK }}$ | ns |
|  |  | Tslakw | $C_{L}=40 \mathrm{pF}$ | $1 \cdot \mathrm{~T}_{\text {CLK }}$ | $5 \cdot \mathrm{~T}_{\text {CLK }}$ | ns |
| ACK OFF delay time | LS个 | Tshakr | $C_{L}=40 \mathrm{pF}$ | － | 15 | ns |
|  |  | Tshakw | $C_{L}=40 \mathrm{pF}$ | － | 15 | ns |
| Data output lead time | ACK $\downarrow$ | Tdaklr | $C_{L}=40 \mathrm{pF}$ | $1 \cdot \mathrm{~T}_{\text {CLK }}$ | － | ns |
| Data float delay time | LS个 | Tshd | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ | － | 22 | ns |
| Data setup time | LS个 | TdsL |  | 12 | － | ns |
| Data hold time | ACK $\downarrow$ | $\mathrm{T}_{\text {AKDH }}$ | － | 0 | － | ns |

## 

MSTS，SSTS，and IOP are updated when one or more cycles of CLK signals are input between the CS＝H or LS＝H levels．

### 7.4.3 CPU IF = 1 (H8)

If IF1 $=\mathrm{L}$ level and IF0 $=\mathrm{H}$ level are set, the interface will be for H 8 CPUs.
$<$ Write cycle >

<Read cycle >


| Item |  | Symbol |  | Conditions | Min． | Max． | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Address setup time | RD $\downarrow$ | $\mathrm{T}_{\text {AR }}$ |  | － | 11 | － | ns |
| Address setup time | WR $\downarrow$ | TAW |  | － | 11 | － | ns |
| Address hold time | RD个，WR个 | TRWA |  | － | 0 | － | ns |
| CS setup time | RD $\downarrow$ | TCSR |  | － | 3 | － | ns |
| CS setup time | WR $\downarrow$ | Tcsw |  | － | 3 | － | ns |
| CS hold time | RD个，WR个 | Trwcs |  | － | 0 | － | ns |
| WRQ ON time | CS $\downarrow$ | Tcswt | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 12 | ns |
| WRQ L level time |  | Twalt |  | － | － | $4 \cdot \mathrm{~T}_{\text {cLK }}$ | ns |
| Data output delay time | RD $\downarrow$ | $\mathrm{T}_{\text {RDLD }}$ | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 24 | ns |
| Data output delay time | WRQ个 | Twthd | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 13 | ns |
| Data float delay time | RD个 | Trdhd | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 21 | ns |
| WR signal width |  | Twr |  |  | 7 | － | ns |
| Data setup time | WR个 | TDWR |  | － | 11 | － | ns |
| Data hold time | WR个 | Twrd |  | － | 0 | － | ns |

${ }^{*}$ It is he time from $W R Q=H$ level to $W R=H$ level is reached when $W R Q$ signal is output．

## l m p or r $\quad$ a $\quad \mathrm{n}$ t

MSTS，SSTS，and IOP are updated when one or more cycles of CLK signals are input between the $\mathrm{CS}=\mathrm{H}$ or $\mathrm{LS}=\mathrm{H}$ levels．

### 7.4.4 CPU IF = 2 (8086)

If IF1 $=\mathrm{H}$ level and IF0 $=\mathrm{L}$ level are set, the interface will be for 8086 CPUs.
$<$ Write cycle $>$

$<$ Read cycle $>$


| Item |  | Symbol |  | Conditions | Min． | Max． | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Address setup time | RD $\downarrow$ | $\mathrm{T}_{\text {AR }}$ |  | － | 11 | － | ns |
| Address setup time | WR $\downarrow$ | TAw |  | － | 11 | － | ns |
| Address hold time | RD个，WR个 | TRWA |  | － | 0 | － | ns |
| CS setup time | RD $\downarrow$ | TCSR |  | － | 3 | － | ns |
| CS setup time | WR $\downarrow$ | Tcsw |  | － | 3 | － | ns |
| CS hold time | RD个，WR个 | Trwcs |  | － | 0 | － | ns |
| WRQ ON time | CS $\downarrow$ | Tcswt | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 12 | ns |
| WRQ L level time |  | Twalt |  | － | － | $4 \cdot \mathrm{~T}_{\text {cLK }}$ | ns |
| Data output delay time | RD $\downarrow$ | $\mathrm{T}_{\text {RDLD }}$ | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 24 | ns |
| Data output delay time | WRQ个 | Twthd | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 13 | ns |
| Data float delay time | RD个 | Trdhd | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 21 | ns |
| WR signal width |  | Twr |  |  | 7 | － | ns |
| Data setup time | WR个 | TDwr |  | － | 11 | － | ns |
| Data hold time | WR个 | Twrd |  | － | 0 | － | ns |

${ }^{*}$ It is he time from $W R Q=H$ level to $W R=H$ level is reached when $W R Q$ signal is output．

## I m p o r t a n t

MSTS，SSTS，and IOP are updated when one or more cycles of CLK signals are input between the $\mathrm{CS}=\mathrm{H}$ or $\mathrm{LS}=\mathrm{H}$ levels．

### 7.4.5 CPU IF = $\mathbf{3}$ (Z80)

If IF1 $=\mathrm{H}$ level and IF0 $=\mathrm{H}$ level are set, the interface will be for $\mathrm{Z80}$ CPUs.
$<$ Write cycle $>$
A4 ~ $A \theta$

<Read cycle >


| Item |  | Symbol |  | Conditions | Min． | Max． | Unit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Address setup time | RD $\downarrow$ | TAR |  | － | 11 | － | ns |
| Address setup time | WR $\downarrow$ | TAW |  | － | 11 | － | ns |
| Address hold time | RD个，WR个 | TRWA |  | － | 0 | － | ns |
| CS setup time | RD $\downarrow$ | TCSR |  | － | 3 | － | ns |
| CS setup time | WR $\downarrow$ | Tcsw |  | － | 3 | － | ns |
| CS hold time | RD个，WR个 | Trwcs |  | － | 0 | － | ns |
| WRQ ON time | CS $\downarrow$ | TCswt | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 12 | ns |
| WRQ L level time |  | Twait |  | － | － | 4． $\mathrm{T}_{\text {CLK }}$ | ns |
| Data output delay time | RD $\downarrow$ | TrdL | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 24 | ns |
| Data output delay time | WRQ个 | Twthd | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 13 | ns |
| Data float delay time | RD个 | Trdhd | $\mathrm{C}_{\mathrm{L}}=40 \mathrm{pF}$ |  | － | 21 | ns |
| WR signal width |  | Twr |  |  | 7 | － | ns |
| Data setup time | WR个 | Tdwr |  | － | 11 | － | ns |
| Data hold time | WR $\uparrow$ | TwRD |  | － | 0 | － | ns |

$*_{1}$ It is he time from $W R Q=H$ level to $W R=H$ level is reached when $W R Q$ signal is output．

## I m p or r tacht

MSTS，SSTS，and IOP are updated when one or more cycles of CLK signals are input between the CS＝H or LS＝H levels．

### 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 | TRST | - | 7 | 8 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
| RST delay time | $\mathrm{T}_{\text {RSTD }}$ | - | 7 | 8 | $\times \mathrm{T}_{\text {CLK }} \mathrm{ns}$ |
| SRST delay time | TsRsto | - | 11 | 12 | $\times \mathrm{T}_{\text {cLK }} \mathrm{ns}$ |
| CLR input signal | - | - | - | 1 | $\times \mathrm{T}_{\text {cLK }} \mathrm{ns}$ |
| EA, EB, EZ input signals | $\mathrm{T}_{\text {EAB }}$ | RENV2.EINF=0 | - | 1 | $\times \mathrm{T}_{\text {cLK }} \mathrm{ns}$ |
|  |  | RENV2.EINF=1 | 2 | 3 | $\times \mathrm{T}_{\text {clk }} \mathrm{ns}$ |
| PA, PB input signals | $\mathrm{T}_{\text {PAB }}$ | RENV2. PINF=0 | - | 1 | $\times \mathrm{T}_{\text {cle }} \mathrm{ns}$ |
|  |  | RENV2.PINF=1 | 2 | 3 | $\times \mathrm{T}_{\text {cLK }} \mathrm{ns}$ |
| +EL, -EL, +SD, -SD, ORG, | - | RENV1. FLTR=0 | - | 1 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
| ALM, INP, CEMG input signals |  | RENV1. FLTR=1 | 64 | 80 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
| ERC output signal ON width | - | RENV1. EPW=000b | 224 | 240 | $\times \mathrm{T}_{\text {clk }} \mathrm{ns}$ |
|  |  | RENV1. EPW=001b | 1792 | 1920 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
|  |  | RENV1. EPW=010b | 7168 | 7680 | $\times \mathrm{T}_{\text {cLK }} \mathrm{ns}$ |
|  |  | RENV1. EPW=011b | 28672 | 30720 | $\times \mathrm{T}_{\text {clk }} \mathrm{ns}$ |
|  |  | RENV1. EPW=100b | 229376 | 245760 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
|  |  | RENV1. EPW=101b | 917504 | 983040 | $\times \mathrm{T}_{\text {cLK }} \mathrm{ns}$ |
|  |  | RENV1.EPW=110b | 1835008 | 1966080 | $\times \mathrm{T}_{\text {cLK }} \mathrm{ns}$ |
| ERC output signal OFF width | - | RENV1.ETW=01b | 224 | 240 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
|  |  | RENV1.ETW=10b | 28672 | 30720 | $\times T_{\text {clk }} \mathrm{ns}$ |
|  |  | RENV1.ETW=11b | 1835008 | 1966080 | $\times \mathrm{T}_{\text {cLK }} \mathrm{ns}$ |
| +DR, -DR, PE input signals | - | RENV1. DRF=0 | - | 1 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
|  |  | RENV1. DRF=1 | 524288 | 655360 | $\times \mathrm{T}_{\text {clk }} \mathrm{ns}$ |
| PCS input signal | - | - | - | 1 | $\times \mathrm{T}_{\text {cLK }} \mathrm{ns}$ |
| LTC input signal | - | - | - | 1 | $\times \mathrm{T}_{\text {clk }} \mathrm{ns}$ |
| CSTA input signals | - | - | 4 | 5 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
| CSTA output signal | - | - | 8 |  | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
| CSTP input signal | - | - | 4 | 5 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
| CSTP output signal | - | - | 8 |  | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
| BSY signal ON delay time | Tcmbisy | - | 4 | 5 | $\times \mathrm{T}_{\text {cLk }} \mathrm{ns}$ |
|  | Tstabsy | - | 7 | 8 | $\times \mathrm{T}_{\text {CLK }} \mathrm{ns}$ |
| Start delay time | Tcmbpls | - | 16 | 17 | $\times T_{\text {cLk }} \mathrm{ns}$ |
|  | Tstapls | - | 19 | 20 | $\times \mathrm{T}_{\text {clk }} \mathrm{ns}$ |

### 7.5.1 RST signal



### 7.5.2 SRST command



### 7.5.3 EA, EB signals

### 7.5.3.1 2-pulse mode (encoder)

EA

7.5.3.2 90-degree phase difference mode (encoder)

EA


### 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 | Sep 16, 2009 | - New document |
| 2nd | Apr 16, 2018 | - 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 met and the written data may be deleted due to the shift". <br> - P46: Description in FLTR is revised as follows: <br> Incorrect: Apply filters to the +EL, -EL, SD, <br> ORG, ALM, and INP inputs. <br> Correct: Apply filters to the +EL, -EL, SD, ORG, ALM, INP and CEMG inputs. <br> -P46: "Note 2" is added in "RENV1.PDTC". <br> -P46: The following sentences are added <br> in "Note 2"; <br> "When the value of magnification setting register (RMG) is an even number, an error occurs in the duty ratio. <br> For details, see "11-3-2. Control the output pulse width and operation complete timing". <br> - P111: The following sentence and chart are added in "11-3-2. Control the output pulse width and operation complete timing": <br> [Addition] <br> "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. <br> 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 . |
| 2nd | Apr 16, 2018 | - P117: Revision in "Apply an input filter to EZ". <br> Incorrect: <Set FLTR (bit 26) in RENV2> Correct: <Set FLTR (bit 18) in RENV2> <br> - P117: Revision in "Apply an input filter to EZ". <br> Incorrect: By applying a filter, signals with a pulse width of $4 \mu \mathrm{sec}$ or less will be ignored. <br> Correct: By applying a filter, input pulse (width) less than 3 cycles of CLK signal will be disabled. |


| Revision | Date | Content |
| :---: | :---: | :---: |
|  |  | - P117: Revision in "Apply an input filter to EZ". <br> In correct <br> Correct <br> - P <br> 134: 4 new charts are added in "[Speed change using the comparator]". One whole page increases by this addition. |
| 2nd | Apr 16, 2018 | -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 $Z 80$ mode, the uppermost side data in the case of 68000 mode and H 8 mode). <br> -P30: The following sentences are added in"Direct access method"; "Furthermore, after writing access, wait 3 cycles of reference clock of PCL6045BL 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.") |
| 2nd | Apr 16, 2018 | -P95: Following sentences are added in "- Acceleration/deceleration operations": <br> "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. <br> -The factors of deceleration stop are the followings: <br> 1) ALM signal input <br> 2) Software limit <br> 3) Comparator 1 to 5 <br> For this reason, set the stop method by 1) to 3) to "immediately stop". Even if the stop method 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". <br> -Changed " $\mu \mathrm{sec}$ " to " $\mu \mathrm{s}$ ". <br> -Removed "Label list" in Appendix 3. |
| 3 d | July 2, 2018 | -P14: 6-3. CPU I/F circuit block diagram <br> 1) $Z 80$ I/F (memory map, full-address) <br> Changed the location of "Pull-up" in the diagram. |


| Revision | Date | Content |
| :---: | :---: | :---: |
|  |  | -P15: 3) 8086 I/F (Memory map, full address) <br> Changed "Decode circuit" to "Interrupt control circuit". <br> -P15: 4) 8086 I/F (I/O map, reduced address) <br> Added CPU brand (8086CPU) and PCL model (PCL6045BL) in the diagram. <br> -P16: 5) H8 I/F (full address) <br> Removed the arrow $(\rightarrow)$ beside the "Decode circuit". <br> -P16: 6) H8 I/F (reduced address) <br> Removed the arrow $(\rightarrow)$ beside the "Decode circuit". <br> -P16: Added "Note" under the "6) H8 I/F (reduced address)" diagram. <br> -P17: Corrected the connecting location of "\#System reset" in 7) 68000 I/F (full address) diagram. <br> -P17: 8) 68000 I/F (reduced address) . <br> "Diagram" and "Note" are revised. <br> -6-4-2. Internal map of each axis <br> <।/F for Z80 (Direct access)> chart is revised: <br> -P19: "Address signal" of " 07 to 04 " is revised from "BUFB3~" to "BUFB3 to BUFB0". <br> -P20: Added the address below "93 to 90". <br> -P20: Added I/F for Z80 (Indirect access) chart. <br> -P23: The title of the chart is changed from <When used with the H8 and $8068 \mathrm{I} / \mathrm{F}$ (Direct access)> to <l/F for H 8 or 68000 (Direct access)>. |
| 3rd | July 3, 2018 | -P28: 7-1-1. Write an operation command (Axis assignment omitted) <br> Added "Command" to D15 to D0. <br> -P39: 8-2. Pre-registers <br> Changed from " 5 " to "2nd Pre-register" in "PRMV etc." box. <br> -P39: 8-2-1. Writing to the operation pre-registers. <br> Revised the chart by changing "Set" to "determined". <br> -P40:8-2-3. Writing to the comparator pre-registers <br> Revised the chart by changing "Set" to "determined". |
| 3rd | July 3, 2018 | P43: 8-3-8. PRMD (RMD) register <br> Revised bit 28 from " 0 " to "MIPM". <br> P80: 9-5-1-1. Origin return operation 0 $(O R M=0000)$ <br> Revised "High speed operation <Sensor: EL (ELM = 1), SD (SDM = 0, SDLT = 0), ORG>" chart. <br> P82: 9-5-1-5. Origin return operation 4 (ORM $=0100$ ) <br> Revised "Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>" chart. <br> P83: 9-5-1-6. Origin return operation 5 (ORM = 0101) <br> Revised "Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001) >" chart. |


| Revision | Date | Content |
| :---: | :---: | :---: |
|  |  | P85: 9-5-1-11. Origin return operation 10 (RENV3.ORM $=1010$ ) <br> Revised "High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>" chart. <br> P85: 9-5-1-12. Origin return operation 11 (RENV3.ORM = 1011) <br> Revised "High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>" chart. <br> P104: 10-3. Manual FH correction <br> (i) Make a linear acceleration/deceleration range smaller: <br> Revised the equation of "PRMV $\leqq$ ". <br> P105: Manual FH correction <br> (3)-3 When PRUS>PRDS <br> (i) Make a linear acceleration/deceleration range smaller. <br> Revised the equation of "PRMV $\leqq ", \quad " P R M V>"$. <br> (ii) Eliminate the linear acceleration section and make a linear deceleration range smaller. <br> Revised the equation of "PRMV $\leqq$ ". <br> P119: 11-6-1. INP signal <br> Replace "deflection" by "deviation". <br> P126: 11-10-1. Counter type and input method <br> Revised the entry of "COUNTER 4 and1/2 of reference clock from "Not possible" to "Possible". <br> P130: 11-10-3. Latch the counter and count condition: <br> Revised "Set an event interrupt cause <br> <Set RIRQ. IRLT (bit 14) and RIRQ.IROL (bit 15)> |
| 3 rd | July 3, 2018 | P131: 11-10-4. Stop the counter: <br> Change the position of " $n$ ". <br> P133: 11-11-1. Comparator types and functions <br> " [Comparison method] "chart" <br> Revised "C1RM" to "C2RM" under "Comparator 2" <br> P134: "Specify the output timing for an internal synchronous signal": <br> Revised from "Set RENV5.SYO1 to 3 (bits 16 to 19)" to "Set RENV5.SYO0 to 3 (bits 16 to 19)". <br> P143: 11-14. Synchronous starting <br> "Specify the internal synchronous signal output timing" <br> Revised from "<Set RENV5.SYO1 to 3 (bits 16 to 19)>" to "<Set RENV5.SYO0 to 3 (bits 16 to 19)>" <br> P156: 12-5-2. CPU-I/F 2) (IF1 = H, IF0 = L) 8086 <br> Revised the chart by changing the entry between "Address setup time for \#WR" and "Min." |



| Revision | Date | Content |
| :---: | :---: | :---: |
|  |  | 6-5-4. Reading the main status <br> Changed "set to" to "becomes" <br> 6-5-5. Reading the sub status and input/output port Changed "set to" to "becomes" |
| 4th | July 31, 2018 | 8-3-40. PRCI (RCI) register <br> Changed "the U axis is not available for circular interpolation control" to "the U-axis cannot be a master axis for circular interpolation control" <br> Changed "(number of pulses calculated by the formula)" to "(See [Circular interpolation with acceleration/deceleration] in 9-8-8. Circular interpolation)". <br> 8-3-42. Bit 2 and Bit3 <br> Changed "IPLy" to "IPLz" <br> Changed "IPLz" to "IPLu" <br> 9-2-7. Times operation <br> Added "(be masked)". <br> 9-3. Pulser (PA/PB) input mode <br> Added "as a start command". <br> Changed "<ESP0 bit 14) in REST>" to "<ESPO bit 14) in REST>". <br> 9-4. Changed "PE" to "\#PE". <br> 9-5. Set the EZ count <br> Changed "zero return to" to "origin return". <br> 9-5-1. Origin return operation <br> Changed "zero return to" to "origin return". <br> 9-5-1-1 to 9-5-1-13 <br> Changed the figures. <br> 9-5-3-1. Changed the figures. <br> $9-6$. Setting the $\pm E L$ input filter <br> Deleted "ORG". <br> 9-8-3. 1) Changed "ideal or arc" to "ideal line or arc" <br> 9-8-3. 2) Corrected "PRUP" to "PRUR" <br> 9-8-8. Circular interpolation <br> Added "After circular interpolation operation, the machine moves to the specified end point at the same speed as circular interpolation speed. Please note that the operation will not stop (perpetual circular motion)". <br> Added "within all interpolated area" <br> 10-3. (3)-2. Corrected the first formula <br> 11-3-2. Setting the operation complete timing <br> Corrected "RMD" to "PRMD". |


| Revision | Date | Content |
| :---: | :---: | :---: |
| 4th | July 31, 2018 | 11-5-2. Enable/disable SD signal input <br> Corrected "RMD" to "PRMD". <br> 11-5-3. Apply an input filter to EZ <br> Corrected "<Set FLTR (bit 18) in RENV1>" to"<FINF (bit 18) in RENV2>". <br> 11-6. <br> Changed to "(CND0 to 3, operation status)" to "(CND3 to 0 in RSTS, operation status)". <br> 11-11-1. Major application of Comparator 2 <br> in table. <br> Corrected "COUNTER1" to "COUNTER2". <br> PFC00 Register <br> Corrected "Set" to "Undetermined". <br> 11-11-5. Corrected "C1PM = 1" to "C1RM = 1" <br> Corrected "C2PM =1" to "C2RM = 1" <br> 11-14-3. Note 2. <br> Corrected "the $X$ and $Y$ axes" to "the $Y$ and $Z$ axes" in STEP2 Details <br> Corrected " Y -axis" to " Z -axis" and " Z -axis" to " Y -axis" in STEP3 Details <br> 2. <br> Corrected "the X and Y axes" to "the Y and Z axes" <br> Corrected " Y -axis" to "Z-axis" and "Z-axis" to " Y -axis" <br> 12-1. Corrected " $\mathrm{V}_{\mathrm{dd}}$ " to " $\mathrm{VDD}_{\mathrm{DD}}$ ". <br> 12-2. Corrected " $\mathrm{V}_{\mathrm{dd}}$ " to " $\mathrm{V}_{\mathrm{DD}}$ ". <br> 12-3. Corrected " $V_{d d}$ " to "VDD". <br> L level output voltage <br> Corrected "-6 uA" to "-6 mA". <br> Appendix 3. <br> CU3B Position <br> Corrected " 25 " to " 26 ". <br> CU1R, CU2R, CU3R, CU4R Description <br> Changed "zero return" to "origin return". <br> ECZ, EZD Description <br> Changed "zero return" to "origin return". <br> ESAL Description <br> Changed "zero return" to "origin return". |
| 5th | Dec 26, 2023 | TA600137-ENO/O <br> Corrected errors in "DA70023-1/3E" and thoroughly revised. <br> 1-2. Features <br> - Servomotor driver I/F |


| Revision | Date | Content |
| :---: | :---: | :---: |
|  |  | The ERC signal is a pulsed output. The pulse length can be set. <br> Incorrect: ( $12 \mu \mathrm{~s}$ to 104 ms . A level output is also available. <br> Correct: ( $11 \mu \mathrm{~s}$ to 100 ms . Level outputs are available) <br> $\therefore$ In this manual, see " 2.1 Features". <br> 4. Functions of Terminals <br> I/O column of \#RST. <br> Incorrect: Input <br> Correct: input U <br> $\therefore$ In this document, see "7.3 DC characteristics ". <br> I/O column in IFO,IF1. <br> Incorrect: Input <br> Correct: input U <br> $\therefore$ In this document, see "7.3 DC characteristics". <br> 8-3.8. PRMD(RMD) register <br> Incorrect: <br> Correct : <br> 31 to 28 Not defined (Always set to 0.) <br> $\therefore$ In this manual, see "5.4.3.1 RMD(PRMD): Operation mode." <br> ※ "RMD.MSDC = 1" does not function correctly. <br> 8-3-13. RENV1 register <br> "Description" column of "10\|EROE". <br> Correct : ERC signal can be output at the time of immediate stop by CEMG command (05h). <br> $\therefore$ In this manual, see "5.4.3.1 RMD(PRMD): Operation mode". |


| Revision | Date | Content |
| :---: | :---: | :---: |
|  |  | "Description" column of "14 to $12 \mid E P W 2$ to 0 ". <br> Incorrect: <br> Correct : <br> 000b: 11 to $13 \mu \mathrm{~s} \quad 001 \mathrm{~b}$ : 91 to $98 \mu \mathrm{~s} \quad 010 \mathrm{~b}$ : 360 to $390 \mu \mathrm{~s} 011 \mathrm{~b}: 1.4$ to 1.6 ms 100b: 11 to $13 \mathrm{~ms} \quad$ 101b: 46 to $50 \mathrm{~ms} \quad$ 110b: 93 to 100 ms 111b: Level output <br> $\therefore$ In this manual, see "5.4.3.2 REV1: Environment setting 1". <br> Hereafter, the explanation of EPW bits is the same. <br> "Description" column of "17 to 16\|ETW1 to 0". <br> Incorrect: <br> $00: 0 \mu \mathrm{~s} \quad 01: 12 \mu \mathrm{~s} \quad$ 10b: $1.6 \mathrm{~ms} \quad 011: 104 \mathrm{~ms}$ <br> Correct : <br> 00b: $0 \mu \mathrm{~s} \quad 01 \mathrm{~b}$ : 11 to $13 \mu \mathrm{~s} \quad$ 10b: 1.4 to $1.6 \mathrm{~ms} \quad 011 \mathrm{~b}$ : 93 to 100 ms <br> $\therefore$ In this manual, see "5.4.3.2 RENV1: Environment setting 1". <br> Hereafter, the explanation of ETW bits is the same. <br> "Description" column of "26\|FLTR". <br> Incorrect: 1: When a filter is applied, signal pulses shorter than $4 \mu$ s or less. <br> Correct: 0 : The pulse signals of $0.05 \mu \mathrm{~s}$ or more width are recognized. <br> 1: The pulse signals of $3 \mu \mathrm{~s}$ or less width are completely ignored. <br> $\therefore$ In this manual, see "5.4.3.2 RENV1: Environment setting 1". <br> Hereafter, the explanation of FLTR bits is the same. <br> "Description" column of "27\|DRF". <br> 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. <br> Correct : 0 : The pulse signals of $0.05 \mu \mathrm{~s}$ or more width are recognized. <br> 1: The pulse signals of $3 \mu$ s or less width are completely ignored. <br> $\therefore$ In this manual, see "5.4.3.2 RENV1: Environment setting 1". <br> Hereafter, the explanation of DRF bits is the same. <br> 8-3-16. RENV4 register <br> "Description" column of "23\|IDXM". <br> Incorrect: 0: Outputs an IDX signal while COUNTER4 = RCMP2. <br> Correct: 0 : When RCUN4 = RCMP4 is established, IDX signal is output at the level. <br> $\therefore$ In this manual, see "5.4.3.5 RENV4: Environment setting 4". <br> 8-3-18. RENV6 register <br> "Description" column of "15\|PSTP". <br> Incorrect: 1: Even if a stop command is written, the PCL will operate for the number of pulses that are already input on PA/PB. Note 1. <br> 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. |



| Revision | Date | Content |
| :---: | :---: | :---: |
|  |  | Set RENV6.PSTP=1 to delay the operation until the total output pulse becomes an integral multiple of the multiplication value. <br> However, for interpolating control ( $68 \mathrm{~h}, 69 \mathrm{~h}, 6 \mathrm{Ah}, 6 \mathrm{Bh}, 6 \mathrm{Ch}, 6 \mathrm{Dh}$ ), it will be stopped ignoring RENV6.PSTP=1. <br> $\therefore$ In this manual, see "5.5.3 Pulser control". <br> 9-4-1. Continuous operation using an external switch (MOD:02h) <br> Incorrect: By turning ON an EL signal for the feed direction, movement on the axis will stop. However, the axis can be fed in the reverse direction. An error interrupt (\#INT output) will not occur. <br> Correct: When counting in positive direction, the operation stops by $+E L$ signal $O N$. <br> When counting in negative direction, the operation stops by - EL signal ON. <br> When RENV5.PDSM $=0$, operation mode continues. The error interrupt is not generated when EL signal is stopped by ON in the operating direction. In such cases, you can escape from EL position value of the stopping factor by inputting a +DR or DR signal in the reverse direction. <br> When RENV5.PDSM = 1, EL signal is stopped by ON, an error interrupt is generated, and operation mode is completed. <br> $\therefore$ In this manual, see "5.5.4.1 Continuous movement (02h)". <br> 9-4-2. Positioning operation using an external switch (MOD:56h) <br> Incorrect: By turning ON the EL signal corresponding to the feed direction, the axis will stop operation and an error interrupt (\#INT output) occurs. <br> Correct : When counting in positive direction, the operation stops by + EL signal ON. <br> When counting in negative direction, the operation stops by - EL signal ON. <br> When RENV5.PDSM $=0$, operation mode continues. The error interrupt is not generated when EL signal is stopped by ON in the operating direction. In such cases, you can escape from EL position value of the stopping factor by inputting a +DR or DR signal in the reverse direction. <br> When RENV5.PDSM = 1, EL signal is stopped by ON, an error interrupt is generated, and operation mode is completed. <br> $\therefore$ In this manual, see "5.5.4.2 Incremental movement (56h) ". <br> 9-8-1. Interpolation operations <br> Incorrect: $62 \mathrm{~h} \mid$ Continuous linear interpolation 2 for 1 to 4 axes <br> 63h \| Linear interpolation 2 for 1 to 4 axes <br> Correct: $62 \mathrm{~h} \mid$ Continuously movement in linear interpolation 2 control (1 axis or more) <br> 63h\| Incremental movement in linear interpolation 2 control (1 axis or more) |


| Revision | Date | Content |
| :---: | :---: | :---: |
|  |  | $\therefore$ In this manual, see "5.5.8 Linear interpolation 2 control". <br> 10-1. Speed patterns <br> In the note of "High speed operation 1)\| Positioning mode". <br> 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. <br> Correct : RPLS<RSDC <br> $\Rightarrow$ Start decelerating and stops when RPLS=0. <br> $\therefore$ In this manual , see "6.3.1 Speed patterns ". <br> 11-1. Reset <br> Incorrect: To reset the LSI, hold the \#RST terminal L level while supplying at least 8 cycles of a reference clock signal. <br> Correct : For RST signal, input an L level signal of 8 cycles or more of CLK signal and an H level signal of 8 cycles or more of CLK signal. <br> $\therefore$ In this manual, see "6.1.1 Hardware reset ". <br> 11-5-3. ORG, EZ signals <br> Incorrect: The input logic of the ORG signal and EZ signal can be changed using the RENV1 register (environment setting 1). <br> Correct: The The input logic (RENV1.ORGL) of ORG signal can be selected. <br> The input logic (RENV2.EZL) of EZ signal can be selected. <br> $\therefore$ In this manual, see "6.7.3 Origin (ORG), Encoder Z phase (EZ)". <br> 11-8. External stop/simultaneous stop <br> Incorrect: 2) To stop simultaneously using an external circuit, connect as follows. <br> Correct: 2) To stop simultaneously also from an external circuit, connect as follows. <br> $\therefore$ In this manual, see "6.10 External stop / Simultaneous stop". <br> 11-11-1. Comparator types and functions <br> Incorrect: [Comparison method] C4S3 to 0/ Use COUNTER1 as a ring counter \| "1010" C4S3 to 0/ Use COUNTER2 as a ring counter| "1010" <br> Correct: [Comparison method] C4S3 to 0/ Use COUNTER1 as a ring counter \|-| C4S3 to 0/ Use COUNTER2 as a ring counter |-| - <br> $\therefore$ In this manual, see "6.13.1 Comparator types and functions". |



| Revision | Date | Content |
| :---: | :---: | :---: |
|  |  | CLR input signal width <br> Incorrect: CLR input-signal-width \|||2Tcık | ns <br> Correct: CLR input signal width\|||1Tcıк | |ns <br> $\therefore$ In this manual, see "7.5 Operation timing". <br> ALM, INP, +EL, -EL, SD, ORG input signal width <br> Incorrect: ALM, INP, +EL, -EL, SD, ORG input signal width \|||2Tcık||ns <br> Correct: ALM, INP, +EL, -EL, SD, ORG input signal width\|||1Tcıк| |ns <br> $\therefore$ In this manual, see "7.5 Operation timing". <br> ERC Output signal width <br> Incorrect: ERC Output signal width \|||254×n Tcıк| $255 \times n$ Tcık\| ns <br> Correct : ERC Output signal width \|||224×n Tсак $\mid 240 \times n$ Tсıк $\mid$ ns <br> $\mathrm{n}=1,8,32,128,1024,4096,8192$ <br> $\therefore$ In this manual, see "7.5 Operation timing". <br> +DR, -DR, PE input signal width <br> Incorrect: + DR, -DR, PE input signal width \| || $2 T_{\text {cle }} \mid$ ns <br> Correct: + DR, -DR, PE input signal width \|||1Tcık| |ns <br> $\therefore$ In this manual, see "7.5 Operation timing". <br> PCS, LTC input-signal-width <br> Incorrect: PCS, LTC input signal width \|||2TcıK | ns <br> Correct: PCS,LTC input signal width\|||1Tськ||ns <br> $\therefore$ In this manual, see "7.5 Operation timing". <br> \#BSY signal ON delay time <br> Incorrect: \#BSY signal ON delay time $\left\|\mathrm{T}_{\text {stabsy }}\right\|\left\|7 \mathrm{~T}_{\text {clik }}\right\| \mid n s$ <br> Correct: \#BSY signal ON delay time $\left\|\mathrm{T}_{\text {stabsy }}\right\|\left\|8 \mathrm{~T}_{\text {cıк }}\right\|$ \|ns <br> $\therefore$ In this manual, see "7.5 Operation timing". <br> Start delay time <br>  <br> \|Tstapls $\|\quad\| 17 T_{\text {clik }} \mid$ ns <br> Correct: Start delay \| Tcmdpls $\left\|\left\|17 T_{\text {clk }}\right\|\right.$ \|ns <br> \|Tstapls $\mid$ \|20Tcli $\mid$ ns <br> $\therefore$ In this manual, see "7.5 Operation timing". <br> [Handling Precautions] <br> 3. Precautions for installation <br> Incorrect: Do not keep the temperature at 250 degrees or higher for more than 10 seconds. $250^{\circ} \mathrm{C}$ or higher within 10 seconds <br> $220^{\circ} \mathrm{C}$ or higher within 35 seconds <br> Correct: Do not keep the temperature at 255 degrees (A profile) or higher for more than |


| Revision | Date |  |
| :---: | :---: | :---: |
|  |  | 10 seconds. |
|  |  | within 10 seconds of time exceeding $255^{\circ} \mathrm{C}$ [A profile] |
|  |  | $255^{\circ} \mathrm{C}$ or more within 10 seconds |
|  |  | $220^{\circ} \mathrm{C}$ or higher within 60 seconds |
|  |  | $\therefore$ In this manual, see "1.2.3.4 Precautions for installation". |

# www.pulsemotor.com/global 

Information
www.pulsemotor.com/global/contact

Fifth edition issued in December, 2023
Copyright 2009 Nippon Pulse Motor Co., Ltd.


[^0]:    *1 OUT signal output is delayed for the period of FH speed from PA signal ON at the longest.

[^1]:    For "idling pulses", see "6.6 Idling control".

