User's Manual
For
PCL6113/6123/6143
Pulse Control LSI

NPM
Nippon Pulse Motor Co., Ltd.

[Preface]

Thank you for considering our pulse control LSI, the "PCL6100 series."
Before using the product, read this manual to become familiar with the product.
Please note that the section "Precautions for handling," which include details about installing this IC, can be found at the end of this manual.

[Cautions]

(1) Copying all or any part of this manual without written approval is prohibited.
(2) The specifications of this LSI may be changed to improve performance or quality without prior notice.
(3) Although this manual was produced with the utmost care, if you find any points that are unclear, wrong, or have inadequate descriptions, please let us know.
(4) We are not responsible for any results that occur from using this LSI, regardless of item (3) above.

- Explanation of the descriptions in this manual

1. The "x" "y" "z" and "u" of terminal names and bit names refer to the X axis, Y axis, Z axis and U axis, respectively.
2. Terminals with a bar over the name (ex. $\overline{\text{RST}}$) are negative logic. Their logic cannot be changed. Terminals without a bar over the name are positive logic. Their output logic can be changed.
3. When describing the bits in registers, "n" refers to the bit position. A "0" means that the bit is in position 0, and that it is prohibited to write to any bit other than the "0" bit. Finally, this bit will always return a "0" when read.
4. Unless otherwise indicated, figures related to timing (intervals) in this manual are based on a reference clock of 19.6608 MHz.

INDEX

1. Outline and Features

1-1. Outline

The PCL6113, PCL6123, PCL6143 are CMOS LSIs designed to provide the oscillating, high-speed pulses needed to drive stepper motors and servomotors (pulse string input types).
It can offer various types of control over the pulse strings and therefore the motor performance. These include continuous operation, positioning, zero return at a constant speed, linear acceleration/deceleration, and S-curve acceleration/deceleration.
The number of control axes is as follows: one for the PCL6113, two for the PCL6123, and 4 for the PCL6143. They offer linear interpolation of multiple axes (using single or multiple PCLs), confirmation of a PCL's operation status, and interrupt output by a variety of conditions. In addition, they are equipped with servomotor driver control features.
These functions can be used with simple commands. The intelligent design philosophy reduces the burden on the CPU units to control motors.

1-2. Features

♦ Single voltage power supply 3.3 V
These PCLs can be operated from a 3.3 V (±10%) single voltage power supply.
The output signal level range is 0 to 3.3 V. The input signal level range is 0 to 3.3 V, or 0 to 5 V.

♦ Super high-speed pulse train output
9.8 Mpps can be output when using a 19.6608 MHz (standard) reference clock, or 15 Mpps when using a 30 MHz (maximum) reference clock.

♦ CPU-I/F
These PCLs all contain integral interface circuits for four different CPU types, and they can be connected to a wide variety of CPUs.
Examples of CPU types: Z80, 8086, H8, or 68000 etc.

♦ Acceleration/deceleration speed control
Linear acceleration/deceleration and S-curve acceleration/deceleration are available.
Linear acceleration/deceleration can be inserted in the middle of an S-curve acceleration/deceleration curve. (Specify the S-curve range.)
The S-curve range can specify each acceleration and deceleration independently. Therefore, you can create an acceleration/deceleration profile that consists of linear acceleration and S-curve deceleration, or vice versa.

♦ Interpolation
These PCLs can perform linear interpolation (offering synchronized operation) of any number of axes.

♦ Speed override
In single axis operation, the speed can be changed during operation in any of the operation modes. However, the speed cannot be changed during linear interpolation.

♦ Overriding target position 1) and 2)
1) The target position (feed amount) can be changed while feeding in the positioning mode.
If the current position exceeds the newly entered position, the motor will decelerate, stop (immediate stop when already feeding at a low speed), and then feed in the reverse direction.
2) Starts operation the same as in the continuous mode and, when it receives an external signal, it will stop after the specified number of pulses.

♦ Triangle drive elimination (FH correction function)
In the positioning mode, when there are a small number of output pulses, this function automatically lowers the maximum speed and eliminates triangle driving.

♦ Look ahead function
The next set of data (feed amount, initial speed, feed speed, acceleration rate, deceleration rate, speed magnification rate, ramping-down point, operation mode, S-curve range on an acceleration, S-curve range

on a deceleration) can be written while executing the current data.
When the current operation is complete, the system will immediately execute the next operation.

♦ A variety of counter circuits
The following four counters are available separately for each axis.

Counter	Use or purpose	Counter Input/Output
COUNTER1	28-bit counter for control of the command position	Outputs pulses, EA/EB input
COUNTER2	28-bit counter for mechanical position control	Outputs pulses, EA/EB input

Both of them can also be latched by writing a command, or by providing an LTC, or ORG signal.
The PCLs can also be set to reset automatically soon after latching these signals.

♦ Comparator
There are 2 comparator circuits for each axis. They can be used to compare target values and internal counter values.
Comparator 1 can be compared with COUNTER1 and Comparator 2 can be compared with COUNTER2.

♦ Simultaneous start function
Multiple axes controlled by the same LSI, or controlled by multiple sets of this LSI, can be started at the same time.

♦ Simultaneous stop function
Multiple axes controlled by the same LSI, or controlled by multiple sets of this LSI, can be stopped at the same time by a command, by an external signal, or by an error stop on any axis.

♦ Manual pulsar input function
By applying manual pulse signals, you can rotate a motor directly.
The input signals can be 90˚ phase difference signals (1x, 2x, or 4x) or up and down signals.
When an EL signal of the feed direction is input, the PCL stops the output of pulses. But, it can feed in the opposite direction without any command.

♦ Direct input of operation switch
An input terminal for operation switch is provided to directly drive a motor with an external operation switch.
These switches turn the motor forward (+) and backward (-).
The results of a switch press can be set to keep feeding pulses while pressed down, or to feed a single, specified number of pulses for each press of the switch.

♦ Operation mode
The basic operations of this LSI are: continuous operation, positioning, zero return, and linear interpolation. By setting the optional operation mode bits, you can use a variety of operations.
<Examples of the operation modes>
1) Start/stop by command.
2) Continuous operation and positioning operation using a manual pulsar.
3) Single-shot or continuous operation using the drive switch.
4) Zero return operation.
5) Positioning operation using commands.
6) Hardware start of the positioning operation using $\overline{\text{CSTA}}$ input.
7) Feed for a specified amount after turning ON the PCS. (Position override (2))

♦ Zero return sequences
1) Feeds at low speed and stops when the ORG signal is turned ON
2) Feeds at low speed and stops when an EZ signal is received (after the ORG signal is turned ON).
3) Feeds at high speed, decelerates when the SD signal is turned ON, and stops when the ORG signal is turned ON.
4) Feeds at high speed, decelerates, and stops when the ORG signal is turned ON.
5) Feeds at high speed, starts deceleration when the ORG signal is turned ON. Then, it stops when an

EZ signal is received.

♦ Mechanical input signals
The following four signals can be input for each axis.
1) +EL: When this signal is turned ON, while feeding in the positive (+) direction, movement on this axis stops immediately (with deceleration). When this signal is ON, no further movement occurs on the axis in the positive (+) direction. (The motor can be rotated in the negative (-) direction.)
2) -EL: Functions the same as the +EL signal except that it works in the negative (-) direction.
3) SD: This signal can be used as a deceleration signal or a deceleration stop signal, according to the software setting. When this is used as a deceleration signal, and when this signal is turned ON during a high speed feed operation, the motor on this axis will decelerate to the FL speed. If this signal is ON and movement on the axis is started, the motor on this axis will run at the FL low speed. When this signal is used as a deceleration stop signal, and when this signal is turned ON during a high speed feed operation, the motor on this axis will decelerate to the FL speed and then stop.
4) ORG: Input signal for a zero return operation.
For safety, make sure the +EL and -EL signals stay on from the EL position until the end of each stroke.
The input logic for these signals can be changed using the ELL terminal.
The input logic of the SD and ORG signals can be changed using software.

♦ Servomotor I/F
The following three signals can be used as an interface for each axis.
1) INP: Input positioning complete signal that is output by a servomotor driver.
2) ERC: Output deflection counter clear signal to a servomotor driver.
3) ALM: Regardless of the direction of operation, when this signal is ON, movement on this axis stops immediately (deceleration stop). When this signal is ON, no movement can occur on this axis. While the PCL is operating in the timer mode, it cannot be stopped using the ALM input. Even though the PCL is stopped, it will output an INT (interrupt request) when an ALM signal is received.
The input logic of the INP, ERC, and ALM signals can be changed using software.
The ERC signal is a pulsed output. The pulse length can be set. (12 µsec to 104 msec. A level output is also available.)

♦ Output pulse specifications
Output pulses can be set to a common pulse, 2-pulse mode or 90°phase difference mode. The output logic can also be selected.

♦ Emergency stop signal ($\overline{\text{CEMG}}$) input
When this signal is turned ON, movement on all axes stops immediately. While this signal is ON, no movement is allowed on either axes.
This input cannot be disabled. The PCL will stop when this signal is present, even it is in the timer mode.

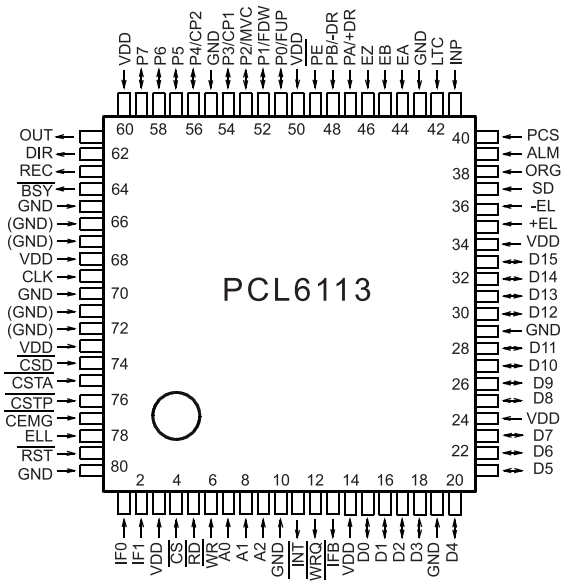
♦ Interrupt signal output
An $\overline{\text{INT}}$ signal (interrupt request) can be output for many reasons.
The $\overline{\text{INT}}$ terminal output signal can use ORed logic for lots of conditions on each axis.
(When more than one LSI is used, wired OR connections are not possible.)

2. Specifications

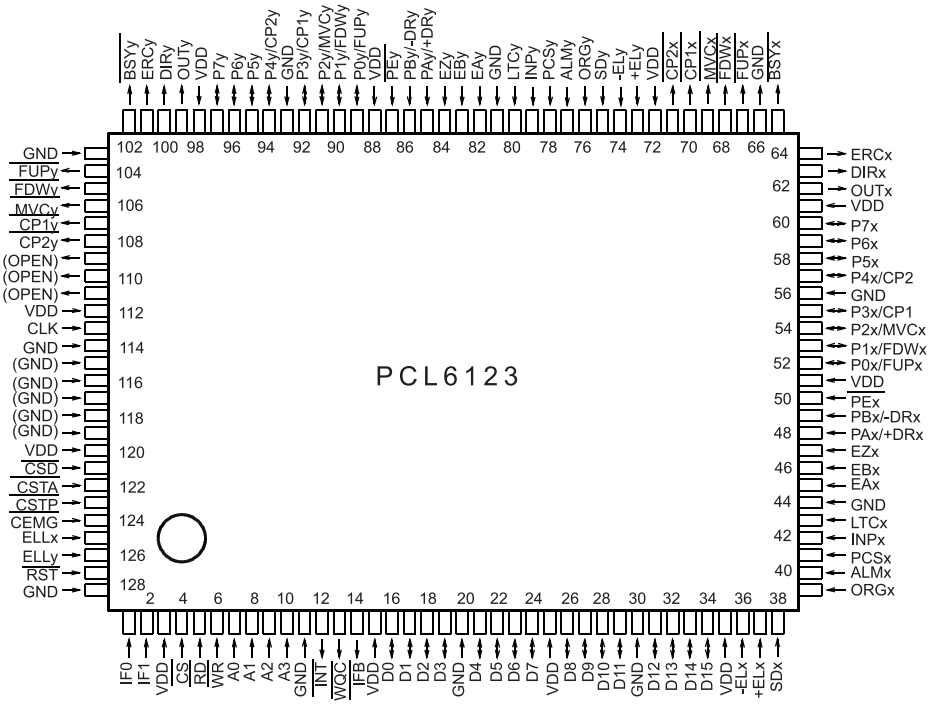
Item	Description
Number of control axes	PCL6113: One PCL6123: Two (X and Y axes) PCL6143: Four (X, Y, Z, and U axes)
Reference clock	Standard: 19.6608 MHz (Max. 30 MHz)
Positioning control range	-134,217,728 to +134,217,727 (28-bit)
Ramping-down point setting range	0 to 16,777,215 (24-bit)
Number of registers used for setting speeds	Two for each axis (FL and FH)
Speed setting step range	1 to 16,383 (14-bits)
Speed multiplication range	0.3x to 600x (Below are examples with a 19.6608 MHz reference clock.) When 0.3x is selected: 0.3 to 4,914.9 pps When 1x is selected: 1 to 16,383 pps When 600x is selected: 600 to 9,829,800. pps The available pulse speed range varies with the reference clock speed. When the reference clock is 30 MHz and if multiplication rate is 600x, the maximum speed will be 15 Mpps.
Acceleration/deceleration characteristics	Selectable acceleration/deceleration pattern for both increasing and decreasing speed separately, using Linear and S-curve acceleration/deceleration.
Acceleration rate setting range	1 to 16,383 (14-bits)
Deceleration rate setting range	1 to 16,383 (14-bits)
Ramping-down point automatic setting	The automatic setting is only available when the acceleration and deceleration curves are symmetrical.
Feed speed automatic correction function	Automatically lowers the feed speed for short distance positioning moves.
Manual operation input	Manual pulsar input, pushbutton switch input
Counter	COUNTER1: Position control counter (28 bits) COUNTER2: Position control counter (28 bits)
Comparators	28-bits x 2 circuits / axis
Interpolation functions	Linear interpolation: Any 2 to 4 axes
Operating temperature range	-40 to +80^{o}C
Power supply	Single voltage power supply: 3.3 V±10%
Package	PCL6113: 80-pin QFP PCL6123: 128-pin QFP PCL6143: 176-pin QFP

3. Terminal Assignment Diagram

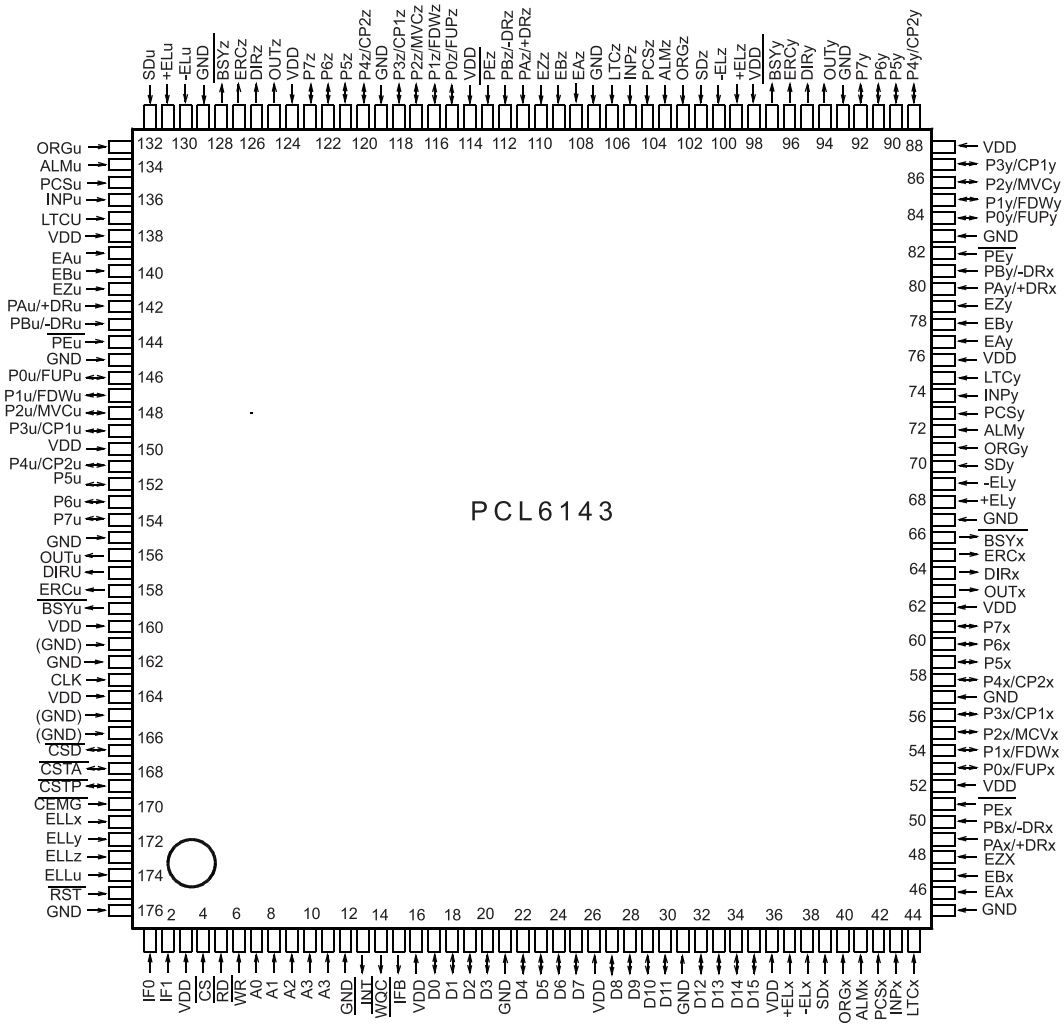
3-1. PCL6113



3-2. PCL6123



3-3. PCL6143



Note: On the actual products, a mark similar to an indexing mark (O mark) may be printed on the LSI for production reasons. The model name and the position of the 1st terminal are as shown in the terminal allocation drawings. You can also identify the 1st terminal by the position of the O mark.

4. Functions of Terminals

Note 1: The letter "n" at the end of each signal name stands for an axis name (x, y, z, or u). (Ex.: ELLn etc.)
Note 2: In the "IN/OUT" column, "IN" indicates an input terminal and "OUT" indicates an output terminal. "I/O" indicates a bi-directional terminal.
Note 3: The logic column indicates the signal logic: Positive or Negative. "P" and "N" are default initial values that can be changed with software. "H" is a hardware setting.
Note 4: The "Handling" column describes how to deal with terminals when they are not used. (Some terminals must be controlled, even when they are being used.)
"OP" means leave open (disconnected). "PU" means pull up. "PD" means pull down. "+V" must be connected to VDD or pulled up. "GN" means a connection to GND. The pull up/down resistance values should be in the range of 5 k to 10 k-ohms.

Signal name	Terminal No.			Input/ output	Logic	Treat -ment	Description
	PCL 6113	PCL 6123	PCL 6143				
GND	10, 19, 29, 43, 55, 65, 70, 80	11, 20, 30, 44, 56, 66, 81, 93, 103, 114, 128	12, 21, 31, 45, 57, 67, 83, 93, 107, 119 129, 145 155, 162 176	Power source			Supply a negative power. Make sure to connect all of these terminals.
VDD	3, 14, 24, 34, 50, 60, 68, 73	3, 15, 25, 35, 51, 61, 72, 88, 98, 112, 120	3, 16, 26, 36, 52, 62, 76, 88, 98, 114, 124, 138, 150, 160, 164	Power source			Supply +3.3 VDC power. The allowable power supply range is +3.3 VDC ±10%. Make sure to connect all of these terminals.
$\overline{\text{RST}}$	79	127	175	Input	Negative		Input reset signal. Make sure to set this signal LOW after turning ON the power and before starting operation. Input and holding $\overline{\text{RST}}$ low for at least 8 cycles of the reference clock. For details about the chip's status after a reset, see section 11-1, "Reset", in this manual.
CLK	69	113	163	Input			As standard, input a 19,6608 MHz reference clock signal. The LSI creates output pulses based on the clock input on this terminal.

<table>
<tr><th rowspan="2">Signal name</th><th colspan="3">Terminal No.</th><th rowspan="2">Input/ output</th><th rowspan="2">Logic</th><th rowspan="2">Treat -ment</th><th rowspan="2">Description</th></tr>
<tr><th>PCL 6113</th><th>PCL 6123</th><th>PCL 6143</th></tr>
<tr><td>IF0
IF1</td><td>1
2</td><td>1
2</td><td>1
2</td><td>Input</td><td></td><td></td><td>Enter the CPU-I/F mode
<table>
<tr><th rowspan="2">IF1</th><th rowspan="2">IF0</th><th rowspan="2">CPU</th><th colspan="4">CPU signals to conect to terminals.</th></tr>
<tr><th>$\overline{RD}$</th><th>$\overline{WR}$</th><th>A0</th><th>$\overline{WRQ}$</th></tr>
<tr><td>L</td><td>L</td><td>68000</td><td>(VDD)</td><td>RW</td><td>LDS</td><td>$\overline{DTACK}$</td></tr>
<tr><td>L</td><td>H</td><td>H8</td><td>$\overline{RD}$</td><td>$\overline{HWR}$</td><td>(GND)</td><td>$\overline{WAIT}$</td></tr>
<tr><td>H</td><td>L</td><td>8086</td><td>$\overline{RD}$</td><td>$\overline{WR}$</td><td>(GND)</td><td>READY</td></tr>
<tr><td>H</td><td>H</td><td>Z80</td><td>$\overline{RD}$</td><td>$\overline{WR}$</td><td>A0</td><td>$\overline{WAIT}$</td></tr>
</table></td></tr>
<tr><td>$\overline{CS}$</td><td>4</td><td>4</td><td>4</td><td>Input</td><td>Negative</td><td></td><td>When the signal level on this terminal is LOW, the $\overline{RD}$ and $\overline{WR}$ terminals will be valid.</td></tr>
<tr><td>$\overline{RD}$
$\overline{WR}$</td><td>5
6</td><td>5
6</td><td>5
6</td><td>Input</td><td>Negative</td><td></td><td>Connect the I/F signals to the CPU. The $\overline{RD}$ and $\overline{WR}$ terminals are valid when $\overline{CS}$ terminal is LOW.</td></tr>
<tr><td>A0
A1
A2
A3
A4</td><td>7
8
9</td><td>7
8
9
10</td><td>7
8
9
10
11</td><td>Input</td><td>Positive</td><td></td><td>Address control signals
For details about terminal A0, see the section describing the IF1 and IF0 terminals.</td></tr>
<tr><td>$\overline{INT}$</td><td>11</td><td>12</td><td>13</td><td>Output</td><td>Negative</td><td>OP</td><td>Outputs an interrupt request signal to a CPU. There is three types of interrupt signals: a stop interrupt, error interrupt, and an event interrupt. The interrupt type can be determined by reading the main status. Each interrupt will be reset by reading the main status, REST (error interrupt cause) register, or RIST (event interrupt cause) register.
The $\overline{INT}$ signal can be masked.</td></tr>
<tr><td>$\overline{WRQ}$</td><td>12</td><td>13</td><td>14</td><td>Output</td><td>Negative</td><td>OP</td><td>Outputs a wait request signal to cause a CPU to wait.
The LSI needs 4 reference clock cycles to process each command.
If you will not be using the $\overline{WRQ}$ signal, check the IFB terminal signal level so that you won't try to access the LSI while it is processing a command.
$\overline{WRQ}$ will only be LOW when $\overline{CS}$ and $\overline{IFB}$ are LOW.</td></tr>
<tr><td>$\overline{IFB}$</td><td>13</td><td>14</td><td>15</td><td>Output</td><td>Negative</td><td>OP</td><td>Signal used to indicate that the LSI is processing commands.
Use this signal to make connections with a CPU that does not have a wait control input terminal.
When the LSI receives a write command from a CPU, this signal will go LOW. When the LSI finishes processing, this signal will go HIGH. The LSI makes sure that this terminal is HIGH and then proceeds to the next step.</td></tr>
</table>

Signal name	Terminal No.			Input/ output	Logic	Treat -ment	Description
	PCL 6113	PCL 6123	PCL 6143				
D0 to D3	15 to 18	16 to 19	17 to 20	Input/ Output	Positive		Bi-directional data bus. When connecting a 16-bit data bus, connect the lower 8 signal lines here.
D4 to D7	20 to 23	21 to 24	22 to 25				
D8 to D11	25 to 28	26 to 29	27 to 30	Input/ Output	Positive	PU or PD	Bi-directional data bus. When connecting a 16-bit data bus, connect the upper 8 signal lines here. When IF0 and IF1 are HIGH, pull these signals down to GND or up to VDD. (A single resistor can be used by combining the lines.)
D12 to D15	30 to 33	31 to 34	32 to 35				
$\overline{\text{CSD}}$	74	121	167	Input/ Output	Negative	PU	This is an input/output terminal for simultaneous deceleration. When performing multiple axis control using more than one PCL, if you want to decelerate the PCLs at the same time, connect all of the $\overline{\text{CSD}}$ terminals to each other. When using this signal, a pull up resistor to VDD is required. The terminal status can be checked on the RSTS terminal (extension status).
$\overline{\text{CSTA}}$	75	122	168	Input/ Output	Negative	PU	This is an input/output terminal for simultaneous starts. When performing multiple axis control using more than one PCL, if you want to start the PCLs at the same time, connect all of the $\overline{\text{CSTA}}$ terminals to each other. When using this signal, a pull up resistor to VDD is required. The terminal status can be checked on the RSTS terminal (extension status).
$\overline{\text{CSTP}}$	76	123	169	Input/ Output	Negative	PU	This is an input/output terminal for simultaneous stops. When performing multiple axis control using more than one PCL, if you want to stop the PCLs at the same time, connect all of the $\overline{\text{CSTP}}$ terminals to each other. When using this signal, a pull up resistor to VDD is required. The terminal status can be checked on the RSTS terminal (extension status).
$\overline{\text{CEMG}}$	77	124	170	Input	Negative	+V	Input for an emergency stop. While this signal is LOW, the PCL cannot start. If this signal changes to LOW while in operation, all the motors will stop operation immediately. The terminal status can be checked on the RSTS terminal (extension status).

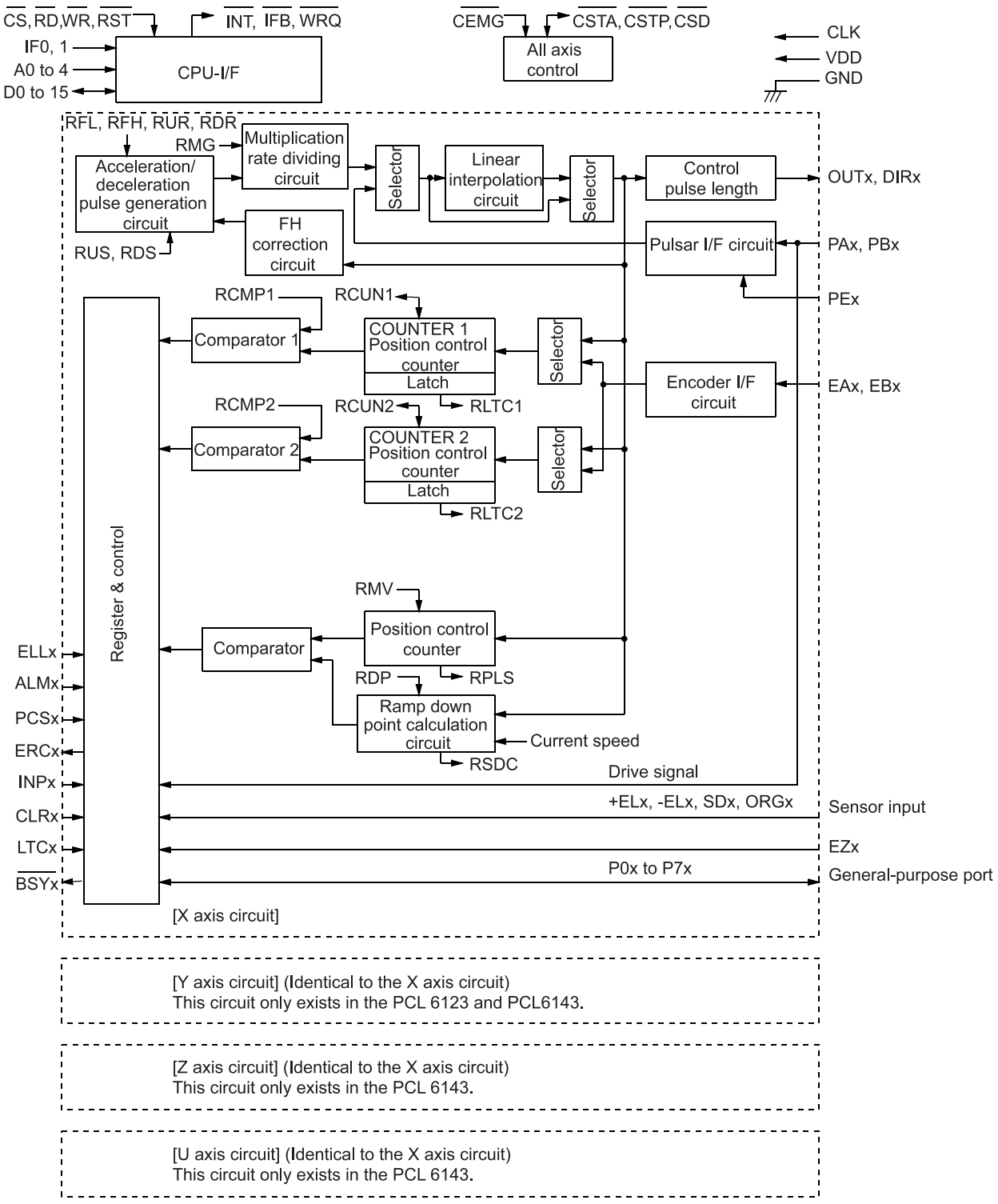
Signal name	Terminal No.			Input/ output	Logic	Treat -ment	Description
	PCL 6113	PCL 6123	PCL 6143				
ELL (ELLn)	78	X: 125 Y: 126	X: 171 Y: 172 Z: 173 U: 174	Input			Specify the input logic for the ±EL signal. LOW: The input logic on ±EL is positive. HIGH: The input logic on ±EL is negative.
+EL (+ELn)	35	X: 36 Y: 73	X: 37 Y :68 Z: 99 U: 130	Input	H	+V	Input end limit signal in the positive (+) direction. When this signal is ON while feeding in the positive (+) direction, the motor on that axis will stop immediately or will decelerate and stop. Specify the input logic using the ELL terminal. The terminal status can be checked using an SSTSW command signal (sub status).
-EL (-ELn)	36	X: 37 Y: 74	X: 38 Y: 69 Z: 100 U: 131	Input	H	+V	Input end limit signal in the negative (-) direction. When this signal is ON while feeding in negative (-) direction, the motor on that axis will stop immediately, or will decelerate and stop. Specify the input logic using the ELL terminal. The terminal status can be checked using an SSTSW command signal (sub status).
SD (SDn)	37	X: 38 Y: 75	X: 39 Y: 70 Z: 101 U: 132	Input	N	+V	Input deceleration signal. Selects the input method: LEVEL or LATCHED inputs. The input logic can be selected using software. The terminal status can be checked using an SSTSW command signal (sub status).
ORG (ORGn)	38	X: 39 Y: 76	X: 40 Y: 71 Z: 102 U: 133	Input	N	+V	Input zero position signal. Used for zero return and other operations. (Edge detection.) The input logic can be selected using software. The terminal status can be checked using an SSTSW command signal (sub status).
ALM (ALMn)	39	X: 40 Y: 77	X: 41 Y: 72 Z: 103 U: 134	Input	N	+V	Input alarm signal. When this signal is ON, the motor on that axis stops immediately, or will decelerate and stop. The input logic can be selected using software. The terminal status can be checked using an SSTSW command signal (sub status).
PCS (PCSn)	40	X: 41 Y: 78	X: 42 Y: 73 Z: 104 U: 135	Input	N	GN	The PCL will start positioning when this signal changes. (Target position override 2) The input logic can be changed using software. The terminal status can be checked using an RSTS command signal.

Signal name	Terminal No.			Input/ output	Logic	Treat -ment	Description
	PCL 6113	PCL 6123	PCL 6143				
INP (INPn)	41	X: 42 Y: 79	X: 43 Y: 74 Z: 105 U: 136	Input	N	GN	Input the position complete signal from servo driver. (in-position signal) The input logic can be changed using software. The terminal status can be checked using an RSTS command signal.
LTC (LTCn)	42	X: 43 Y: 80	X: 44 Y: 75 Z: 106 U: 137	Input	N	GN	Latch counter value of COUNTER 1, COUNTER2. The input logic can be changed using software. The terminal status can be checked using an RSTS command signal.
EA (EAn)	44	X: 45 Y: 82	X: 46 Y: 77 Z: 108 U: 139	Input		GN	Input this signal when you want to control the position using the encoder signal. Input a 90˚ phase difference signal (1x, 2x, 4x) or input positive (+) pulses on EA and negative (-) pulses on EB. When inputting 90˚ phase difference signals, if the EA signal phase is ahead of the EB signal, the LSI will count pulses. The counting direction can be changed using software.
EB (EBn)	45	X: 46 Y: 83	X: 47 Y: 78 Z: 109 U: 140				
EZ (EZn)	46	X: 47 Y: 84	X: 48 Y: 79 Z: 110 U: 141	Input	N	GN	Input a marker signal (this signal is output once for each turn of the encoder) when using the marker signal in zero return mode. Use of the EZ signal improves zero return precision. The input logic can be changed using software. The terminal status can be checked using an RSTS command signal (extension status).
PA (PAn) +DR (+DRn)	47	X: 48 Y: 85	X: 49 Y: 80 Z: 111 U: 142	Input		GN	This is a common input used to trigger either an external pulse (PA, PB), such as a manual pulsar, or an external switch (+DR, -DR). The use of this input will vary with the operation mode setting. When inputting external pulses, you can input 90° phase difference signals (1x, 2x, 4x) or positive (+) pulses (on PA) and negative (-) pulses (on PB). The relation between the input and feed direction can be changed using software.
PB (PBn) -DR (-DRn)	48	X: 49 Y: 86	X: 50 Y: 81 Z: 112 U: 143				
$\overline{\text{PEx}}$	49	X: 50 Y: 87	X: 51 Y: 82 Z: 113 U: 144	Input	Negative	GN	Setting these terminals LOW enables PA/PB. By inputting an axis change switch signal, one manual pulsar can be used alternately for four axes.

Signal name	Terminal No.			Input/ output	Logic	Treat -ment	Description
	PCL 6113	PCL 6123	PCL 6143				
P0/FUP (P0n/FUPn)	51	X: 52 Y: 89	X: 53 Y: 84 Z: 115 U: 146	Input/ Output		PD	Common terminal for general purpose I/O and FUP. When this terminal is used as a general-purpose I/O, you can set it for input or output. When used as an FUP terminal, it will output a signal while accelerating. The FUP output logic can be set using software.
P1/FDW (P1n/FDWn)	52	X: 53 Y: 90	X: 54 Y: 85 Z: 116 U: 147	Input/ Output		PD	Common terminal for general purpose I/O and FDW. When this terminal is used as a general-purpose I/O, you can set it for input or output. When used as an FDW terminal, it will output a signal while decelerating. The FDW output logic can be set using software.
P2/MVC (P2n/MVCn)	53	X: 54 Y: 91	X: 55 Y: 86 Z: 117 U: 148	Input/ Output		PD	Common terminal for general purpose I/O and MVC. When this terminal is used as a general-purpose I/O, you can set it for input or output. When used as an MVC terminal, it will output a signal during operation at a constant speed. The MVC output logic can be set using software.
P3/CP1 (P3n/CP1n)	54	X: 55 Y: 92	X: 56 Y: 87 Z: 118 U: 149	Input/ Output		PD	Common terminal for general purpose I/O and CP1. When this terminal is used as a general-purpose I/O, you can set it for input or output. When used as a CP1 terminal, it will output a signal while establishing the Comparator 1 condition. The CP1 output logic can be set using software.
P4/CP2 (P4n/CP2n)	56	X: 57 Y: 94	X: 58 Y: 89 Z: 120 U: 151	Input/ Output		PD	Common terminal for general purpose I/O and CP2. Note5 When this terminal is used as a general-purpose I/O, you can set it for input or output. When used as CP2 terminal, it will output a signal while establishing the Comparator 2 condition. The CP1 output logic can be set using software.
P5 (P5n)	57	X: 58 Y: 95	X: 59 Y: 90 Z: 121 U: 152	Input/ Output		PD	This is a general-purpose terminal. Set it for use as an input or output terminal using software.
P6 (P6n)	58	X: 59 Y: 96	X: 60 Y: 91 Z: 122 U: 153	Input/ Output		PD	This is a general-purpose terminal. Set it for use as input or output terminal using software.

Signal name	Terminal No.			Input/ output	Logic	Treat -ment	Description
	PCL 6113	PCL 6123	PCL 6143				
P7 (P7n)	59	X: 60 Y: 97	X: 61 Y: 92 Z: 123 U: 154	Input/ Output		PD	This is a general-purpose terminal. Set it for use as input or output terminal using software.
OUT (OUTn)	61	X: 62 Y: 99	X: 63 Y: 94 Z: 125 U: 156	Output	N	OP	Outputs command pulses for controlling a motor. The output specifications are determined by selecting the common pulse mode, 2-pulse mode, or 90° phase difference mode. Set the output mode using software.
DIR (DIRn)	62	X: 63 Y: 100	X: 64 Y: 95 Z: 126 U: 157				
ERC (ERCn)	63	X: 64 Y: 101	X: 65 Y: 96 Z: 127 U: 158	Output	N	OP	Outputs a deflection counter clear signal to a servo driver. The output logic and pulse width can be changed using software. The terminal status can be checked using an RSTS command signal.
$\overline{\text{BSY}}$ ($\overline{\text{BSYn}}$)	64	X: 65 Y: 102	X: 66 Y: 97 Z: 128 U: 159	Output	Negative	OP	Outputs a LOW signal during operation.
$\overline{\text{FUPn}}$		X: 67 Y: 104		Output	Negative	OP	This signal is LOW during acceleration.
$\overline{\text{FDWn}}$		X: 68 Y: 105		Output	Negative	OP	This signal is LOW during deceleration.
$\overline{\text{MVCn}}$		X: 69 Y: 106		Output	Negative	OP	This signal is LOW during constant speed operation.
$\overline{\text{CP1n}}$		X: 70 Y: 107		Output	Negative	OP	This signal is LOW while establishing the Comparator 1 conditions.
$\overline{\text{CP2n}}$		X: 71 Y: 108		Output	Negative	OP	This signal is LOW while establishing the Comparator 2 conditions.
(OPEN)		109,110, 111		Output		OP	Output terminal for checking the PCL when delivered. Do not make any connections to this terminal.
(GND)	66, 67, 71, 72	115, 116, 117, 118, 119	161, 165, 166	Input		GN	This signal is LOW during deceleration.

5. Block Diagram



6. CPU Interface

6-1. Setting the CPU interface type

These PCLs contain the following 4 CPU interface types, in order to facilitate connection to various CPUs. To select a specific type, use the IF0 and IF1 terminals.
Shown below are some circuit examples. To use some other CPU, select the appropriate interface after referring to section "12-5. AC characteristics."

[Example of connections for CPU signals]

Setting status		Interface Name	CPU type	CPU signal to connect to the 6045A terminals			
IF1	IF0			$\overline{RD}$ terminal	$\overline{WR}$ terminal	A0 terminal	$\overline{WRQ}$ terminal
L	L	16-bit I/F-1	68000	+3.3V	R/$\overline{W}$	$\overline{LDS}$	$\overline{DTACK}$
L	H	16-bit I/F-2	H8	$\overline{RD}$	$\overline{HWR}$	(GND)	$\overline{WAIT}$
H	L	16-bit I/F-3	8086	$\overline{RD}$	$\overline{WR}$	(GND)	READY
H	H	8-bit I/F	Z80	$\overline{RD}$	$\overline{WR}$	A0	$\overline{WAIT}$

16-bit I/F-1: A 16-bit interface with a R/W mode input, strobe input, and acknowledge output.
The lower addresses correspond to the upper word in the I/O buffer.
Convenient for use with VME bus and 68000 series CPUs.

16-bit I/F-2: A 16-bit interface with an RD input and a WR input.
The lower addresses correspond to the upper word in the I/O buffer.
Convenient for H8 series CPUs.

16-bit I/F-3: A 16-bit interface with an RD input and a WR input.
The lower addresses correspond to the lower word in the I/O buffer.
Convenient for use with 8086 series CPUs.

8-bit I/F: An 8-bit interface with an RD input and a WR input.
The lower addresses correspond to the lower word in the I/O buffer.
Convenient for use with Z80 series CPUs.

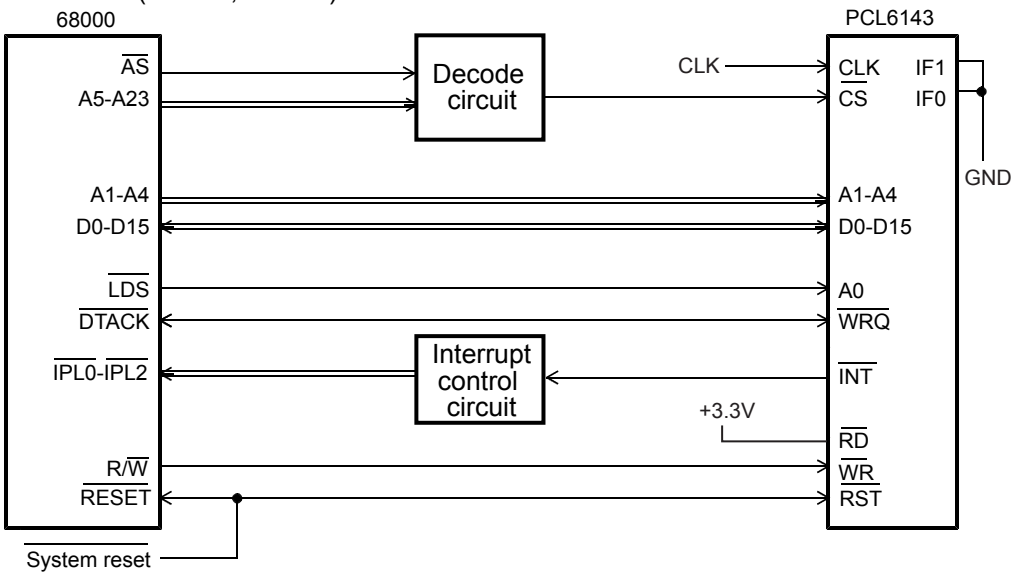
6-2. Hardware design precautions

- All of the input terminals can handle 0 to +5 V signal levels.
- Although all of the output terminals can be pulled up to +5 V (through 5k ohms or more), the output current can not be increased above that available at 3.3 V.
- To reset the LSI, hold the $\overline{RST}$ signal LOW, and input the CLK signal for at least 8-clock cycles.
- Any unused terminals from P0 to P7 should be pulled down to GND externally. (5k to 10k ohms)
- When connecting a CPU with an 8-bit bus, pull down terminals D8 to D15 to GND using an external resistor (5 k to 10 k-ohm). (Shared use of one resister for the 8 lines is available.)
- Use the ELL terminal to change the ±EL signal input logic.

6-3. Examples of CPU interfaces

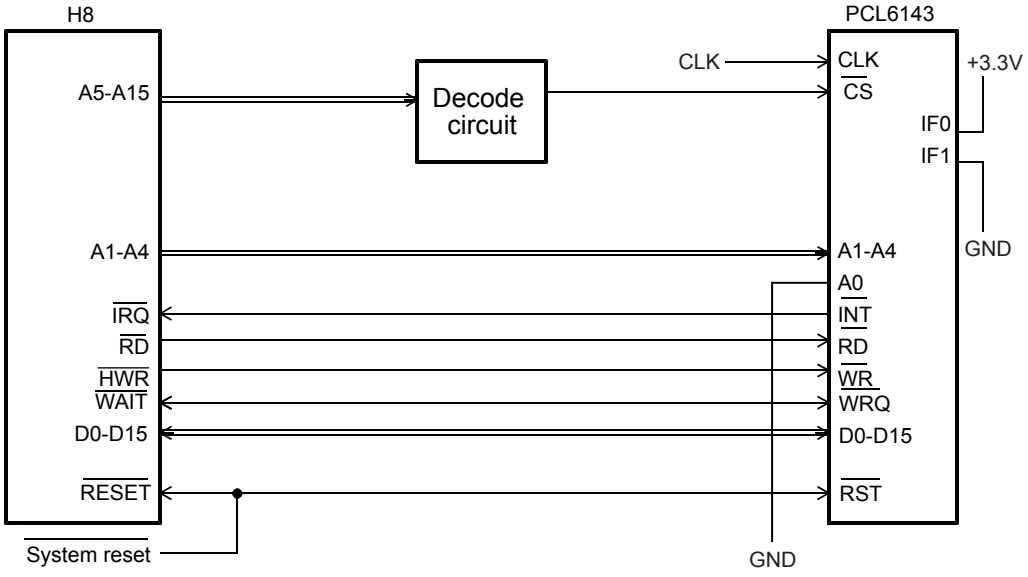
Note: When using the 16-bit I/F, the PCL can only access words (16 bits), not bytes (8 bits).

(1) 16-bit I/F-1 (IF1 = L, IF0 = L)



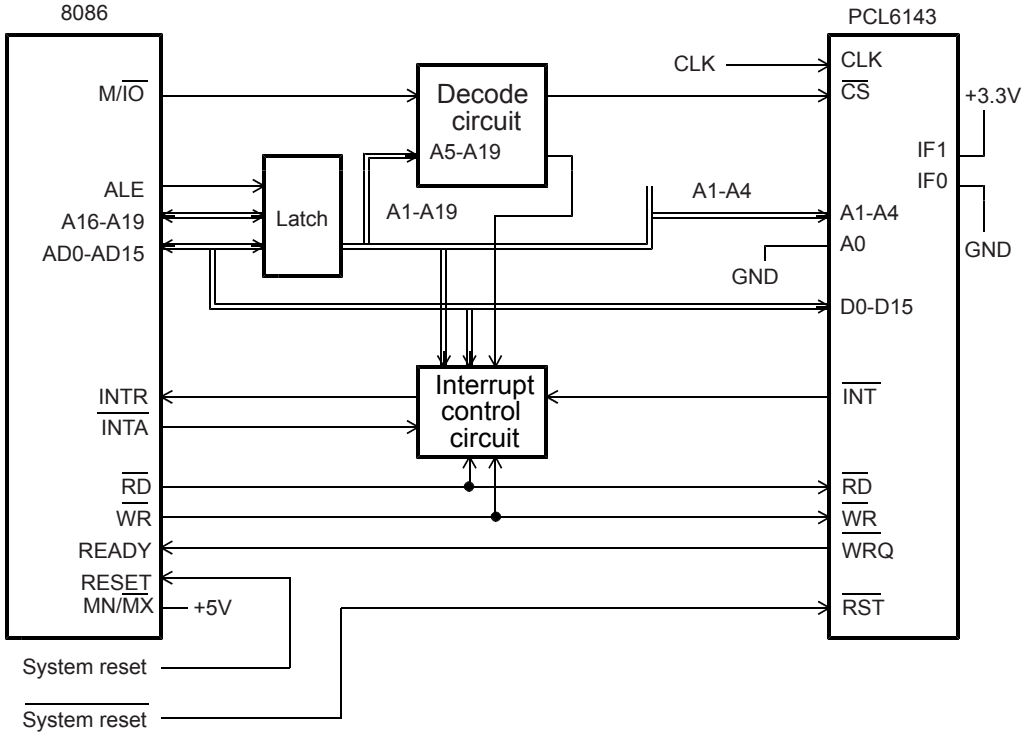
Note: The PCL6113 uses A1 to A2. The PCL6123 uses A1 to A3. The PCL6143 uses A1 to A4.

(2) 16-bit I/F-2 (IF1 = L, IF0 = H)



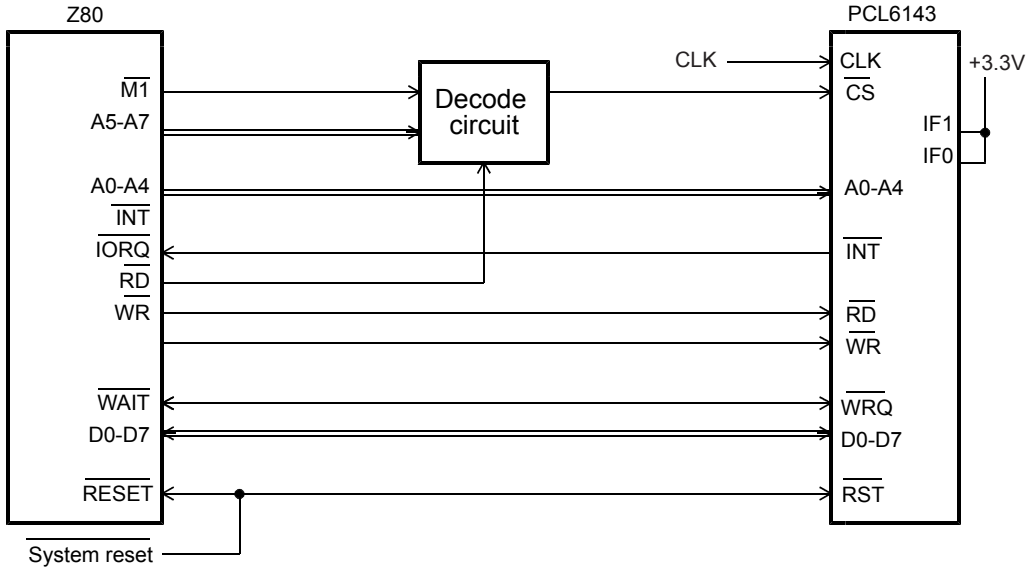
Note: The PCL6113 uses A1 to A2. The PCL6123 uses A1 to A3. The PCL6143 uses A1 to A4.

(3) 16-bit I/F-3 (IF1 = H, IF0 = L)



Note: The PCL6113 uses A1 to A2. The PCL6123 uses A1 to A3. The PCL6143 uses A1 to A4.

(4) 8-bit I/F (IF1 = H, IF0 = H)



Note: The PCL6113 uses A0 to A2. The PCL6123 uses A0 to A3. The PCL6143 uses A0 to A4.

6-4. Address map

6-4-1. Axis arrangement map

In this LSI, the control address range for each axis is independent. It is selected by using address input terminal A3 and A4, as shown below.

A4	A3	Detail
0	0	X axis control address range
0	1	Y axis control address range
1	0	Z axis control address range
1	1	U axis control address range

Note: The table on the left is for the PCL6143.
The PCL6123 does not have an A4 address line. Only the X and Y axes are available.
The PCL6113 does not have A4 or A3 address lines. Only the X axis is available.

6-4-2. Internal map of each axis

The internal map of each axis is defined by A0, A1 and A2 address line inputs.

<When 16-bit I/F-1 or 16-bit I/F-2 mode is selected>

1) Write cycle

A1 to A2	Address signal	Processing detail
11	COMW	Specify an axis, write a control command.
10	OTPW	Change the status of the general-purpose output port (only bits assigned as outputs are effective)
01	BUFW0	Write to the input/output buffer (bits 0 to 15)
00	BUFW1	Write to the input/output buffer (bits 16 to 31)

2) Readout cycle

A1 to A2	Address signal	Processing detail
11	MSTSW	Read the main status (bits 0 to 15)
10	SSTSW	Read the sub status and general-purpose I/O port.
01	BUFW0	Read from the input/output buffer (bits 0 to 15)
00	BUFW1	Read from the input/output buffer (bits 16 to 31)

<When 16-bit I/F-3 mode is selected>

1) Write cycle

A1 to A2	Address signal	Processing detail
00	COMW	Write the axis assignment and control command
01	OTPW	Change the status of the general-purpose output port (only bits assigned as outputs are effective)
10	BUFW0	Write to the input/output buffer (bits 0 to 15)
11	BUFW1	Write to the input/output buffer (bits 16 to 31)

2) Readout cycle

A1 to A2	Address signal	Processing detail
00	MSTSW	Read the main status (bits 0 to 15)
01	SSTSW	Read the sub status or general-purpose input/output port
10	BUFW0	Read from the input/output buffer (bits 0 to 15)
11	BUFW1	Read from the input/output buffer (bits 16 to 31)

<When 8-bit I/F mode is selected>

1) Write cycle

A0 to A2	Address signal	Processing detail
000	COMB0	Write control commands
001	COMB1	Specify an axis (specify control command execution axis)
010	OTPB	Change the status of the general-purpose output port (only bits assigned as outputs are effective)
011		(Invalid)
100	BUFB0	Write to the input/output buffer (bits 0 to 7)
101	BUFB1	Write to the input/output buffer (bits 8 to 15)
110	BUFB2	Write to the input/output buffer (bits 16 to 23)
111	BUFB3	Write to the input/output buffer (bits 24 to 31)

2) Read cycle

A0 to A2	Address signal	Processing detail
000	MSTSB0	Read the main status (bits 0 to 7)
001	MSTSB1	Read the main status (bits 8 to 15)
010	IOPB	Read the general-purpose output port
011	SSTSB	Read the sub status
100	BUFB0	Read from the input/output buffer (bits 0 to 7)
101	BUFB1	Read from the input/output buffer (bits 8 to 15)
110	BUFB2	Read from the input/output buffer (bits 16 to 23)
111	BUFB3	Read from the input/output buffer (bits 24 to 31)

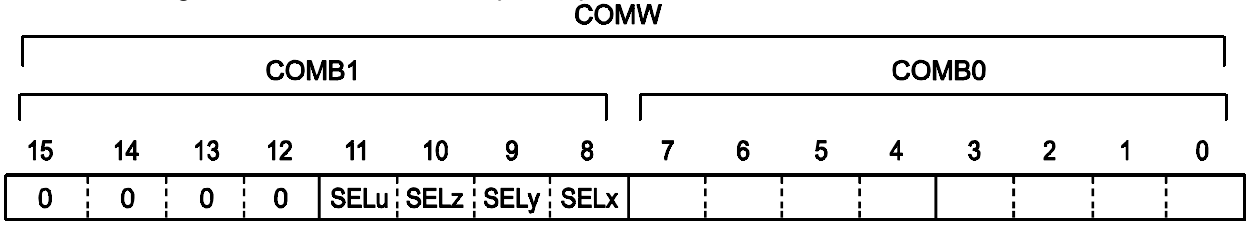
6-5. Description of the map details

6-5-1. Write the command code and axis selection (COMB0, COMB1)

Write the commands for reading and writing to registers and the start and stop control commands for each axis.

COMB0: Set the command code. For details, see 7. "Command (Operation and Control commands)."

SELu to x: Select an axis for executing the command. If all of the bits are 0, only this axis (selected by A4, A3) is selected. To write the same command to more than one axis, set the bits of the selected axes to 1. When you write to a register, the details of the input/output buffer are written into the register for each axis. When you read from a register, the details in the register are written into the input/output buffer for each axis.



Note 1: Specifications using SELu to SELx (used to specify the axis for execution) are effective for all commands, not only register write/read commands.

Note 2: The PCL6143 has SELu to SELx, and the PCL6123 has SELx to SELy. However, the PCL6113 does not have COMB1.

There are two methods to write to a register, as follows: Mixed use of these methods is allowed. The example below uses the PCL6143.

(1) Commands and data I/O are written as one set per axis, and a total of up to 4 sets can be used.
In this case, the axis specification (COMB1), other than starting or stopping an interpolation operation, is performed using 00h.
However, if CSTA and CSTP signals are used to start or stop an interpolation operation, 00h can also be used for this command.
When using multiple sets of PCL6113, 6123, and 6143 LSIs, a common program can be created easily.

A1 to A4	Symbol	Description
0000	COMW_X	X axis command
0010	BUFW0_X	X axis I/O buffer (bit 0 to 15)
0011	BUFW1_X	X axis I/O buffer (bit 16 to 31)
0100	COMW_Y	Y axis command
0110	BUFW0_Y	Y axis I/O buffer (bit 0 to 15)
0111	BUFW1_Y	Y axis I/O buffer (bit 16 to 31)
1000	COMW_Z	Z axis command
1010	BUFW0_Z	Z axis I/O buffer (bit 0 to 15)
1011	BUFW1_Z	Z axis I/O buffer (bit 16 to 31)
1100	COMW_U	U axis command
1110	BUFW0_U	U axis I/O buffer (bit 0 to 15)
1111	BUFW1_U	U axis I/O buffer (bit 16 to 31)

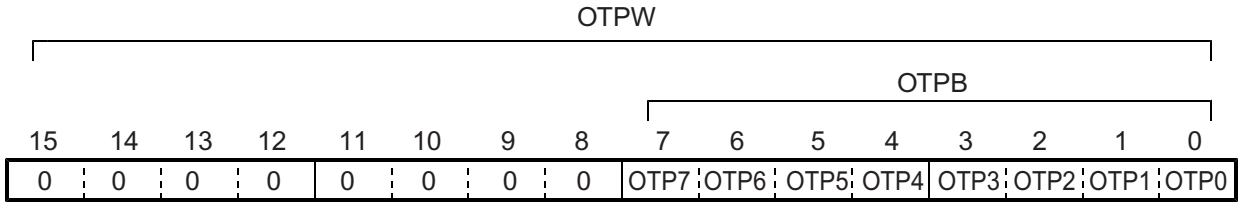
(2) Write the commands to common addresses, and write the data to the I/O area for each axis independently.
In this case, the axis must be specified for each command that is written. (However, the software reset command (SRST) ignores any axis specification.)
One command writes/reads all the axes in the same register, reducing the data setting time.

A1 to A4	Symbol	Description
0000	COMW	Command
0010	BUFW0_X	X axis I/O buffer (bit 0 to 15)
0011	BUFW1_X	X axis I/O buffer (bit 16 to 31)
0110	BUFW0_Y	Y axis I/O buffer (bit 0 to 15)
0111	BUFW1_Y	Y axis I/O buffer (bit 16 to 31)
1010	BUFW0_Z	Z axis I/O buffer (bit 0 to 15)
1011	BUFW1_Z	Z axis I/O buffer (bit 16 to 31)
1110	BUFW0_U	U axis I/O buffer (bit 0 to 15)
1111	BUFW1_U	U axis I/O buffer (bit 16 to 31)

Note: The examples above use COMW on the X axis. However, using COMW on any other axis will perform the identical operation.

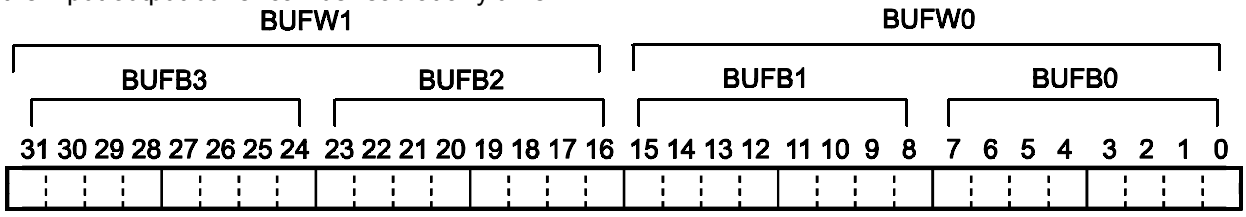
6-5-2. Write to an output port (OTPW, OTPB)

Specify output terminal status from the general purpose I/O terminals P0 to P7.
Bits corresponding to terminals not set as outputs are ignored.
When writing a word, the upper 8 bits are ignored. However, they should be set to 0 for future compatibility.
OTP0 to 7: Specify the status of output terminals P7n to P0n (n = x, y, z and u).
A HIGH is output when the bit is set to 1.



6-5-3. Write/read the input/output buffer (BUFW, BUFB)

When you want to write data into a register, after placing the data in the input/output buffer, write a "register write command" into COMB0. The data in the input/output buffer will be copied into the register.
When you want to write data into the input/output buffer, write a "register read command" into COMB0. The data in the register will be copied to the input/output buffer. Then you can read the data from the input/output buffer.
The order for writing and reading buffers BUFW0 to 1 (BUFB0 to 3) is not specified. The data written in the input/output buffer can be read at any time.



6-5-4. Reading the main status (MSTSW, MSTSB)

MSTSW

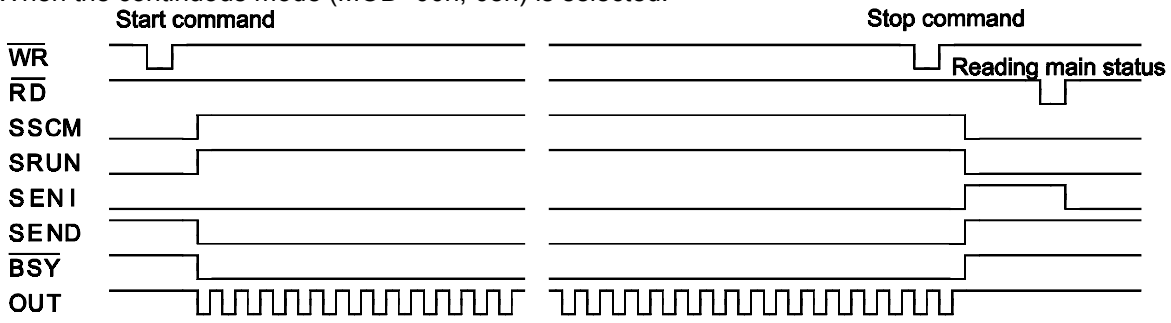
MSTSB1 MSTSB0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SPRF	SEOR	0	0	0	SCP2	SCP1	SSC1	SSC0	SINT	SERR	SEND	SENI	SRUN	SSCM

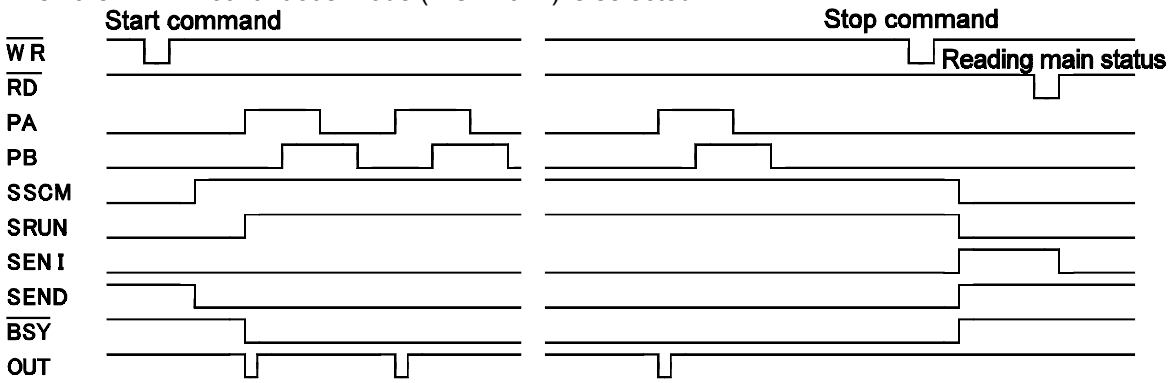
Bit	Bit name	Details
0	SSCM	Set to 1 by writing a start command. Set to 0 when the operation is stopped.
1	SRUN	Set to 1 by the start pulse output. Set to 0 when the operation is stopped.
2	SENI	Stop interrupt flag When IEND in RENV2 is 1, the PCL turns ON the INT output when the status changes from operating to stop, and the SENI bit becomes 1. (After the main status is read, it returns to 0.) When IEND is set to 0, this flag will always be 0.
3	SEND	Set to 0 by writing start command. Set to 1 when the operation is stopped.
4	SERR	Set to 1 when an error interrupt occurs. Set to 0 by reading the RESET.
5	SINT	Set to 1 when an error interrupt occurs. Set to 0 by reading the RIST.
6 to 7	SSC0 to 1	Sequence number for execution or stopping.
8	SCP1	Set to 1 when the COMPARATOR 1 comparison conditions are met.
9	SCP2	Set to 1 when the COMPARATOR 2 comparison conditions are met.
10 to 12		Not defined (always 0)
13	SEOR	When a positioning override cannot be executed (reading the RMV register while stopped), this signal changes to 1. After the main status is read, it changes to 0.
14	SPRF	Set to 1 when the pre-register for the subsequent operation data is full.
15		Not defined (always 0)

Status change timing chart

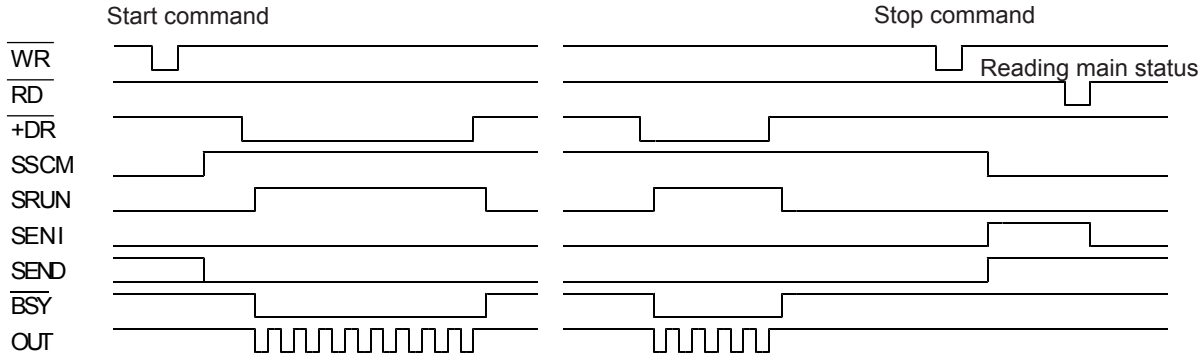
1) When the continuous mode (MOD=00h, 08h) is selected.



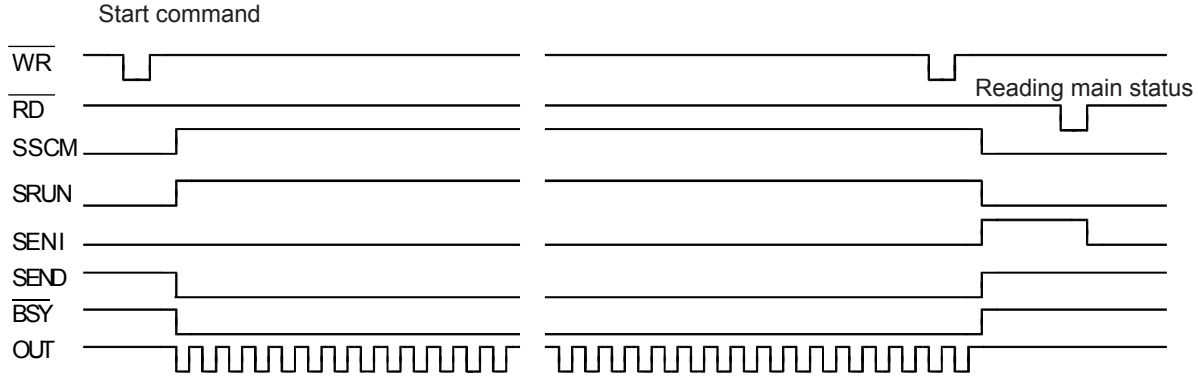
2) When the PA/ PB continuous mode (MOD=01h) is selected.



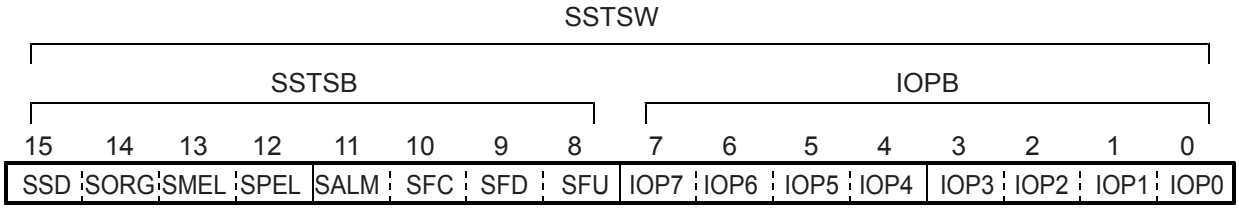
3) When the DR continuous mode (MOD=02h) is selected.



4) When the auto stop mode is selected such as positioning operation mode (MOD=41h).



6-5-5. Reading the sub status and input/output port (SSTSW, SSTSB, IOPB)



Bit	Bit name	Description
0 to 7	IOP0 to 7	Read the status of P0 to 7 (0: L level, 1: H level)
8	SFU	Set to 1 while accelerating.
9	SFD	Set to 1 while decelerating.
10	SFC	Set to 1 while feeding at low speed.
11	SALM	Set to 1 when the ALM input is ON.
12	SPEL	Set to 1 when the +EL input is ON.
13	SMEL	Set to 1 when the -EL input is ON.
14	SORG	Set to 1 when the ORG input is ON.
15	SSD	Set to 1 when the SD input is ON. (Latches the SD signal.)

7. Commands (Operation and Control Commands)

7-1. Operation commands

After writing the axis assignment data to COMB1 (address 1 when an 8-bit-I/F is used), write the command to COMB0 (address 0 when an 8-bit-I/F is used), the LSI will start and stop, as well as change the speed of the output pulses.

When any other interface mode is selected, the PCL will write 16-bit data including axis specifications and commands.

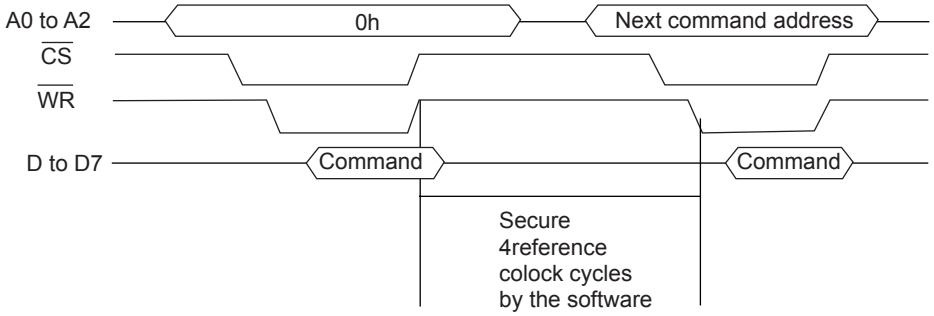
7-1-1. Procedure for writing an operation command (the axis assignment is omitted)

Write a command to COMB0.

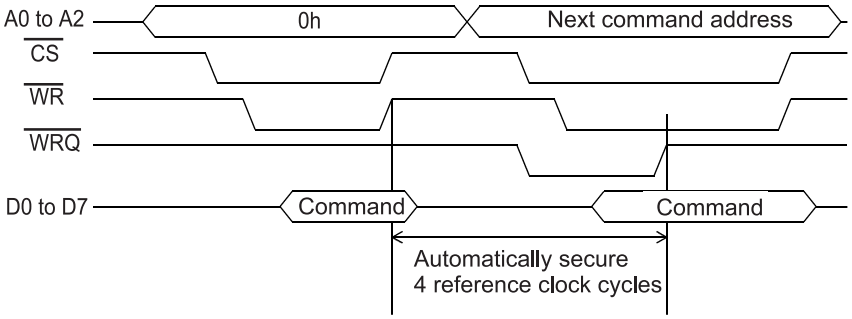
A waiting time of 4 register reference clock cycles (approximately 0.2 μsec when CLK = 19.6608 MHz) is required for the interval between "writing a command" and "writing the next command," "writing a register" and "writing the I/O buffer," and between "reading a register" and "reading the I/O buffer." When the $\overline{WRQ}$ output signal is used by connecting it to the CPU, the CPU automatically ensures this waiting time.

If you want to use a CPU that does not have this waiting function, arrange the program sequence so that access is only allowed after confirming that the $\overline{IFB}$ output signal is HIGH.

1) When not using $\overline{WRQ}$



2) When not using $\overline{WRQ}$



7-1-2. Start command

1) Start command

If this command is written while stopped, the motor will start rotating. If this command is written while the motor is operating, it is taken as the next start command.

COMB0	Symbol	Description
50h	STAFL	FL low speed start
51h	STAFH	FH low speed start
52h	STAD	High speed start 1 (FH low speed -> deceleration stop) Note. 1
53h	STAUD	High speed start 2 (Acceleration -> FH low speed -> Deceleration stop) Note. 1

Note 1: For details, see section 10-1, "Speed patterns."

2) Residual pulses start command

Write this command after the motor is stopped on the way to a positioning, it will continue movement for the number of pulses left in the deflection counter.

COMB0	Symbol	Description
54h	CNTFL	Residual pulses FL low speed start
55h	CNTFH	Residual pulses FH low speed start
56h	CNTD	High speed start 1 residual pulses (FH constant speed -> Deceleration stop)
57h	CNTUD	High speed start 2 residual pulses (Acceleration -> FH constant speed -> Deceleration stop)

3) Simultaneous start command

By setting the RMD register, the LSI will start an axis which is waiting for $\overline{\text{CSTA}}$ signal.

COMB0	Symbol	Description
06h	CMSTA	Output one shot of the start pulse from the $\overline{\text{CSTA}}$ terminal.
2Ah	SPSTA	Only this axis will process the command, the same as when the $\overline{\text{CSTA}}$ signal is input.

7-1-3. Speed change command

Write this command while the motor is operating, the motor on that axis will change its feed speed. If this command is written while stopped it will be ignored.

COMB0	Symbol	Description
40h	FCHGL	Change to the FL speed immediately.
41h	FCHGH	Change to the FH speed immediately.
42h	FSCHL	Decelerate and change to the FL speed.
43h	FSCHH	Accelerate and change to the FH speed.

7-1-4. Stop command

1) Stop command

Write this command to stop feeding while operating.

COMB0	Symbol	Description
49h	STOP	Write this command while in operation to stop immediately.
4Ah	SDSTP	Write this command while feeding at FH low speed or high speed, the motor on that axis will decelerate to the FL low speed and stop. If this command is written while the axis is being fed at FL low speed, the motor on that axis will stop immediately.

2) Simultaneous stop command

Stop the motor on any axis whose $\overline{\text{CSTP}}$ input stop function has been enabled by setting the RMD register.

COMB0	Symbol	Description
07h	CMSTP	Outputs one shot of pulses from the $\overline{\text{CSTP}}$ terminal to stop movement on that axis.

3) Emergency stop command

Stops an axis in an emergency

COMB0	Symbol	Description
05h	CMEMG	Emergency stop (same as a $\overline{\text{CEMG}}$ signal input)

7-1-5. NOP (do nothing) command

COMB0	Symbol	Description
00h	NOP	This command does not affect the operation.

7-2. General-purpose output bit control commands

These commands control the individual bits of output terminals P0 to P7.
When the terminals are designated as outputs, the LSI will output signals from terminals P0 to P7.
Commands that have not been designated as outputs are ignored.
The write procedures are the same as for the Operation commands.
In addition to this command, by writing to a general-purpose output port (OTPB: Address 2 when an 8-bit-I/F is used), you can set 8 bits as a group. See section 7-5, "General-purpose output port control."

COMB0	Symbol	Description	COMB0	Symbol	Description
10h	P0RST	Make P0 LOW.	18h	P0SET	Make P0 HIGH.
11h	P1RST	Make P1 LOW.	19h	P1SET	Make P1 HIGH.
12h	P2RST	Make P2 LOW.	1Ah	P2SET	Make P2 HIGH.
13h	P3RST	Make P3 LOW.	1Bh	P3SET	Make P3 HIGH.
14h	P4RST	Make P4 LOW.	1Ch	P4SET	Make P4 HIGH.
15h	P5RST	Make P5 LOW.	1Dh	P5SET	Make P5 HIGH.
16h	P6RST	Make P6 LOW.	1Eh	P6SET	Make P6 HIGH.
17h	P7RST	Make P7 LOW.	1Fh	P7SET	Make P7 HIGH.

7-3. Control command

Set various controls, such as the reset counter.
The procedures for writing are the same as the operation commands.

7-3-1. Software reset command

Used to reset this LSI.

COMB0	Symbol	Description
04h	SRST	Software reset. (Same function as making the $\overline{RST}$ terminal LOW.)

7-3-2. Counter reset command

Reset counters to zero.

COMB0	Symbol	Description
20h	CUN1R	Reset COUNTER1.
21h	CUN2R	Reset COUNTER2.

7-3-3. ERC output control command

Control the ERC signal using commands.

COMB0	Symbol	Description
24h	ERCOUT	Outputs the ERC signal.
25h	ERCRST	Resets the output when the ERC signal output is specified to a level type output.

7-3-4. Pre-register control command

Cancel the pre-register settings.
See section "8-2. Pre-register" in this manual for details about the pre-register.

COMB0	Symbol	Description
26h	PRECAN	Cancel the operation pre-register.

7-3-5. PCS input command

Entering this command has the same results as inputting a signal on the PCS terminal.

COMB0	Symbol	Description
28h	STAON	Alternative to a PCS terminal input.

7-3-6. LTCH input (counter latch) command

Entering this command has the same result as inputting a signal on the LTC terminal.

COMB0	Symbol	Description
29h	LTCH	Alternative to an LTC (latch counter) terminal input.

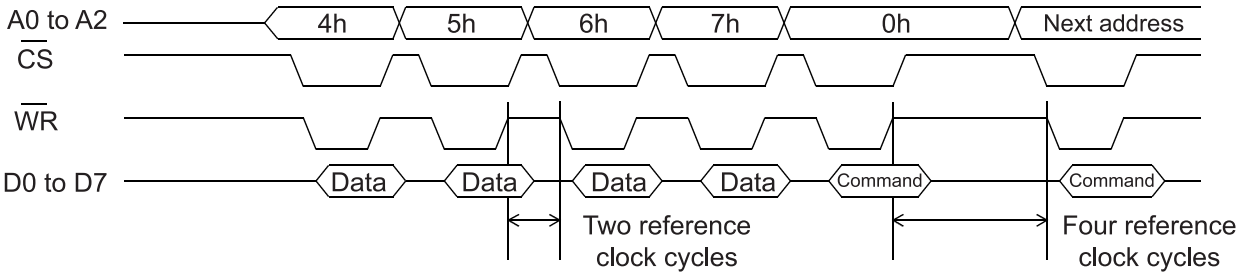
7-4. Register control command

By writing a Register Control command to COMB0 (Address 0 when an 8-bit-I/F is used), the LSI can copy data between a register and the I/O buffer.

Note: When using the I/O buffer while responding to an interrupt, a precaution is required, reading the I/O buffer contents before using it and returning it to its original value after use.

7-4-1. Procedure for writing data to a register (the axis assignment is omitted)

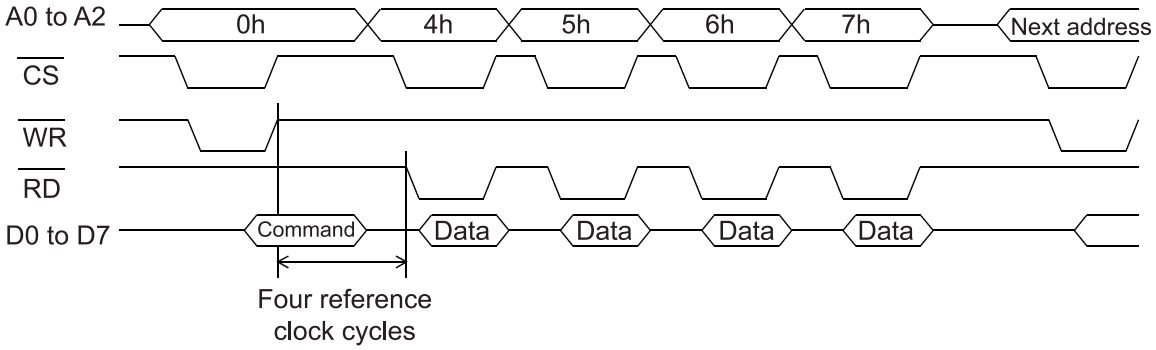
1) Write the data that will be written to a register into the I/O buffer (addresses 4 to 7 when an 8-bit-I/F is used). The order in which the data is written does not matter. However, secure two reference clock cycles between these writings.
2) Then, write a "register write command" to COMB0 (address 0 when an 8-bit-I/F is used).
   After writing one set of data, wait at least two cycles (approx. 0.2 μsec when CLK = 19.6608 MHz) before writing the next set of data.
   In both case 1) and case 2), when the $\overline{WRQ}$ output is connected to the CPU, the CPU wait control function will provide the waiting time between write operations automatically.



7-4-2. Procedure for reading data from a register (the axis assignment is omitted)

1) First, write a "register read out command" to COMB0 (address 0 when an 8-bit-I/F is used).
2) Wait at least four reference clock cycles (approx. 0.2 μsec when CLK = 19.6608 MHz) for the data to be copied to the I/O buffer.
3) Read the data from the I/O buffer (addresses 4 to 7 when an 8-bit-I/F is used). The order for reading data from the I/O buffer does not matter. There is no minimum time between read operations.

When the $\overline{WRQ}$ output is connected to the CPU, the CPU wait control function will provide the waiting time between write operations automatically.



7-4-3. Table of register control commands

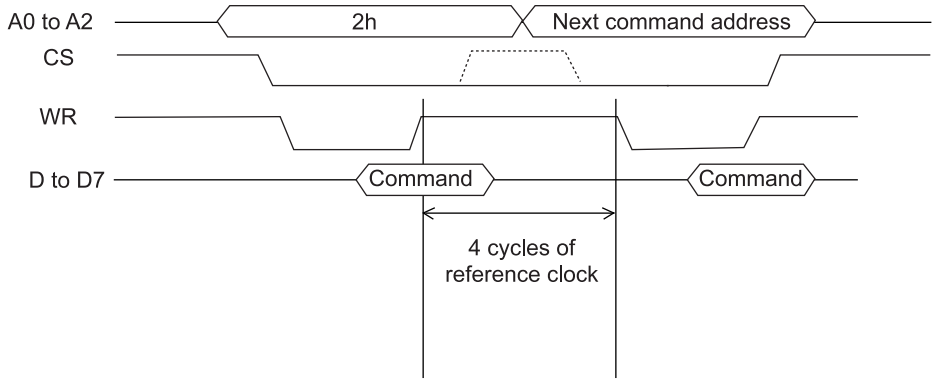
No.	Detail	Bit	Register					Pre-register				
			Name	Read command		Write command		Name	Read command		Write command	
				COMB0	Symbol	COMB0	Symbol		COMB0	Symbol	COMB0	Symbol
1	Feed amount	28	RMV	D0h	RRMV	90h	WRMV	PRMV	C0h	RPRMV	80h	WPRMV
2	Initial speed	14	RFL	D1h	RRFL	91h	WRFL	PRFL	C1h	RPRFL	81h	WPRFL
3	Operation speed	14	RFH	D2h	RRFH	92h	WRFH	PRFH	C2h	RPRFH	82h	WPRFH
4	Acceleration rate	14	RUR	D3h	RRUR	93h	WRUR	PRUR	C3h	RPRUR	83h	WPRUR
5	Deceleration rate	14	RDR	D4h	RRDR	94h	WRDR	PRDR	C4h	RPRDR	84h	WPRDR
6	Speed magnification rate	12	RMG	D5h	RRMG	95h	WRMG	PRMG	C5h	RPRMG	85h	WPRMG
7	Ramping-down point	24	RDP	D6h	RRDP	96h	WRDP	PRDP	C6h	RPRDP	86h	WPRDP
8	Operation mode	30	RMD	D7h	RRMD	97h	WRMD	PRMD	C7h	RPRMD	87h	WPRMD
9	Linear interpolation main axis data	27	RIP	D8h	RRIP	98h	WRIP	PRIP	C8h	RPRIP	88h	WPRIP
10	Acceleration S-curve range	13	RUS	D9h	RRUS	99h	WRUS	PRUS	C9h	RPRUS	89h	WPRUS
11	Deceleration S-curve range	13	RDS	DAh	RRDS	9Ah	WRDS	PRDS	CAh	RPRDS	8Ah	WPRDS
12	Environment setting 1	32	RENV1	DCh	RRENV1	9Ch	WRENV1					
13	Environment setting 2	31	RENV2	DDh	RRENV2	9Dh	WRENV2					
14	Environment setting 3	22	RENV3	DEh	RRENV3	9Eh	WRENV3					
15	COUNTER1 (command)	28	RCUN1	E3h	RRCUN1	A3h	WRCUN1					
16	COUNTER2 (mechanical)	28	RCUN2	E4h	RRCUN2	A4h	WRCUN2					
17	Data for comparator 1	28	RCMP1	E7h	RRCMP1	A7h	WRCMP1					
18	Data for comparator 2	28	RCMP2	E8h	RRCMP2	A8h	WRCMP2					
19	Event INT setting	12	RIRQ	ECh	RRIRQ	ACh	WRIRQ					
20	COUNTER1 latched data	28	RLTC1	EDh	RRLTC1							
21	COUNTER2 latched data	28	RLTC2	EEh	RRLTC2							
22	Extension status	17	RSTS	F1h	RRSTS							
23	Error INT status	9	REST	F2h	RREST							
24	Event INT status	13	RIST	F3h	RRIST							
25	Positioning counter	28	RPLS	F4h	RRPLS							
26	EZ counter, speed monitor	20	RSPD	F5h	RRSPD							
27	Ramping-down point	24	PSDC	F6h	RPSDC							

7-5. General-purpose output port control command

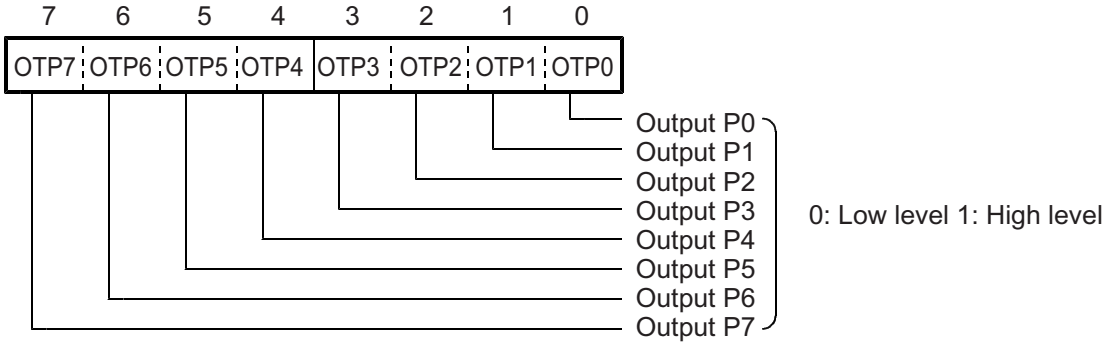
By writing an output control command to the output port (OTPB: Address 2 when using an 8-bit-I/F interface), the PCL will control the output of the P0 to P7 terminals.
When the I/O setting for P0 to P7 is set to output, the PCL will output signals from terminals P0 to P7 to issue the command.
When writing words to the port, the upper 8 bits are discarded. However, they should be set to zero to maintain future compatibility.
The output status of terminals P0 to P7 are latched, even after the I/O setting is changed to input.
The output status for each terminal can be set individually using the bit control command.

7-5-1. Command writing procedures

Write control data to output port (OTPB: Address 2 when an 8-bit-I/F is used).
To continue with the next command, the LSI must wait for four reference clock cycles (approx. 0.2 µsec when CLK = 19.6608 MHz). The $\overline{WRQ}$ terminal outputs a wait request signal.



7-5-2. Command bit allocation



8. Registers

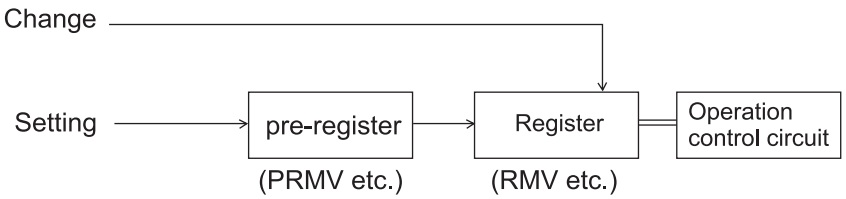
8-1. Table of registers

The following registers are available for each axis.

No.	Register name	Bit length	R/W	Details	2nd pre-register name
1	RMV	28	R/W	Feed amount, target position	PRMV
2	RFL	14	R/W	Initial speed	PRFL
3	RFH	14	R/W	Operation speed	PRFH
4	RUR	14	R/W	Acceleration rate	PRUR
5	RDR	14	R/W	Deceleration rate	PRDR
6	RMG	12	R/W	Speed magnification rate	PRMG
7	RDP	24	R/W	Ramping-down point	PRDP
8	RMD	30	R/W	Operation mode	PRMD
9	RIP	27	R/W	Main axis feed amount during linear interpolation	PRIP
10	RUS	13	R/W	S-curve acceleration range	PRUS
11	RDS	13	R/W	S-curve deceleration range	PRDS
12	RENV1	32	R/W	Environment setting 1 (specify I/O terminal details)	
13	RENV2	31	R/W	Environment setting 2 (specify general-purpose port details)	
14	RENV3	22	R/W	Environment setting 3 (specify zero return and counter details)	
15	RCUN1	28	R/W	COUNTER1 (command position)	
16	RCUN2	28	R/W	COUNTER2 (mechanical position)	
17	RCMP1	28	R/W	Comparison data for comparator 1	
18	RCMP2	28	R/W	Comparison data for comparator 2	
19	RIRQ	12	R/W	Specify event interruption cause	
20	RLTC1	28	R	COUNTER1 (command position) latch data	
21	RLTC2	28	R	COUNTER2 (mechanical position) latch data	
22	RSTS	17	R	Extension status	
23	REST	9	R	Error INT status	
24	RIST	13	R	Event INT status	
25	RPLS	28	R	Positioning counter (number of residual pulses to feed)	
26	RSPD	20	R	EZ counter, current speed monitor	
27	RSDC	24	R	Automatically calculated ramping-down point	

8-2. Pre-registers

The following registers and start commands have pre-registers:
RMV, RFL, RFH, RUR, RDR, RMG, RDP, RMD, RIP, RUS and RDS.
The term pre-register refers to a register which contains the next set of operation data while the current step is executing. This LSI has the following 2-layer structure and executes FIFO operation.



Normally, operation data are written into the pre-register.
To change the current operation status, such as changing the speed, the new data are written into the register.
The data will be shifted (copied) from the pre-register to the register when the next start command is written, or at the end of an operation.
One set of operation data uses multiple pre-registers (PRMV, PRFH,,,,). If the current operation completes before the next set of operation data has been placed in all of the pre-registers, the PCL may start with incomplete data. In order to prevent this problem, the "determined/not determined" status is used.
When a start command is written, the other operation data is considered to be determined, and the PCL will continue its operation immediately after the current operation is complete.
The writing and operating procedures for the pre-registers are shown below.

1) When both the pre-register and register are empty, data that is written to the pre-register will also be written to the register. (Data 1 not determined status).

2) By writing a start command, the contents of the register are declared determined and the PCL will start the operation.

3) During operation, write the next operation data to the pre-register. (A subsequent set of data that is the same as the previous set does not need to be written.)
Since the register is currently in the "determined" status, the next set of operation data is only written to the pre-register. (Data 2)

4) By writing a start command for the next operation, the data in the pre-register is declared to be determined (complete).

5) When the first operation is finished, the data is transferred from the pre-register to the register. The PCL will then start operation according to the next set of operation data (Data 2).

Procedure	Pre-register	Register	SPRF
	0 Not determined	0 Not determined	0
1)	Not determined data 1	Not determined data 1	0
2)	Not determined data 1	Determined data 1	0
3)	Not determined data 2	Determined data 1	0
4)	Determined data 2	Determined data 1	1
5)	Not determined data 2	Determined data 2	0
6)	Not determined data 2	Not determined data 2	0

6) When that operation is complete, the data is again transferred from the pre-register to the register. However, in this case the next set of operation data is "not determined," and so the PCL stops operation.

In step (5) above, the data in the pre-register is "not determined," allowing you to write the next set of operation data.
Data written to the pre-register when the data in the pre-register is already "determined" will be ignored.
When the pre-register is declared to be determined, the SPRF bit in the main status (MSTSW) register will be 1.
Also, the PCL can be set to output an $\overline{\text{INT}}$ signal when the pre-register changes from determined to not determined status by setting the RIRQ (event interrupt cause) register.

Further, in any of the following cases, the pre-register has a "not determined" status, so that you can cancel a continuous start when the current operation is finished.
1) Writing a pre-register cancel command (26h).
2) A stop ordered by using the immediate stop command (49h) or deceleration stop command (4Ah). While in a positioning operation, when the deceleration stop command is written during auto deceleration, the PCL will go to the target position. However, the pre-register is declared "not determined" and the next operation will be cancelled.
3) When the PCL stops because of an error (When any of the bits 0 to 6 in the RESET register changes to a 1.)

Note: To automatically start the next operation using the data already in the pre-register, set the operation complete timing to "end of cycle" (set METM in the RMD to 0). If the "end of pulse" (set METM in the RMD to 1) is selected, the interval between the last pulse and the next operation's start pulse will be narrower: 14 x T_{CLK} (T_{CLK}: Reference clock cycle).
For details, see section 11-3-2. "Output pulse length and operation complete timing."

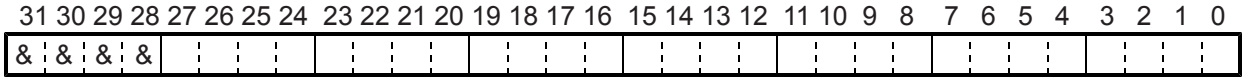
8-3.Description of the registers

The initial value of all the registers and pre-registers is "0."
Please note that with some registers, a value of "0" is outside the allowable setting range.

> Note 1: Bits marked with an "*" asterisk are ignored when written and return a "0" when read.
> Note 2: Bits marked with an "&" are ignored when written. They will be the same as the uppermost bit in the empty column when read. (Extended symbols)

8-3-1. PRMV (RMV) registers

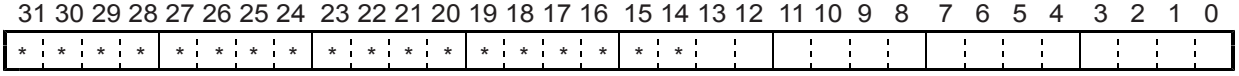
These registers are used to specify the target position for positioning operations. The set details change with each operation mode.
PMV is the register for PRMV.



Setting range: -134,217,728 to +134,217,727.
By changing the RMV register while in operation, the feed length can be overridden.

8-3-2. PRFL (RFL) registers

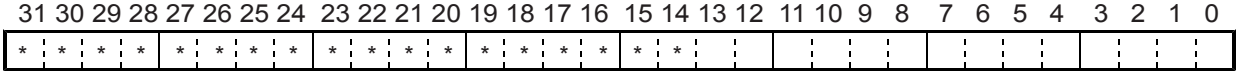
These pre-registers are used to set the initial speed (stop seed) for high speed (with acceleration /deceleration) operations.
RFL is the register for PRFL.



The setting range is 1 to 16,383. However, the actual speed [pps] may vary with the speed magnification rate setting in the PRMG register.

8-3-3. PRFH (RFH) registers

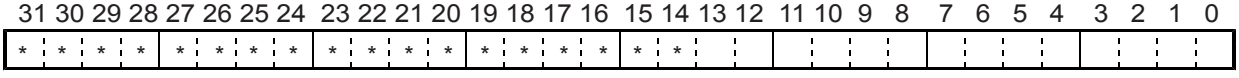
These pre-registers are used to specify the operation speed.
RFH is the working register for PRFH. Write to this register to override the current speed.



The setting range is 1 to 16,383. However, the actual speed [pps] may vary with the speed magnification rate set in the PRMG register.

8-3-4. PRUR (RUR) registers

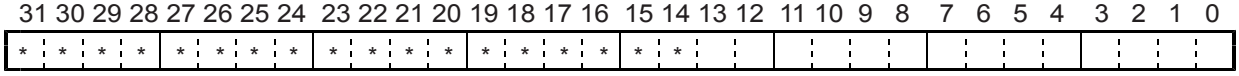
These pre-registers are used to specify the acceleration rate.
RUR is the register for PRUR.



Setting range is 1 to 16,383.

8-3-5. PRDR (RDR) registers

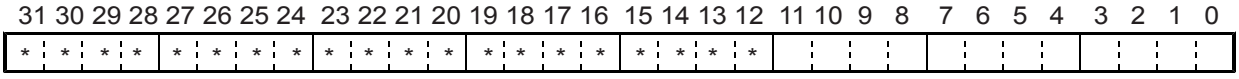
These pre-registers are used to specify the deceleration rate.
RDR is the register for PRDR.



The normal setting range is 1 to 16,383.
When PRDR = 0, the deceleration rate will be the value set by PRUR.
Note: When automatic setting is selected for the ramp down point (MSDP = 0), enter the same value as used for the PRUR, or 0, in this register.

8-3-6. PRMG (RMG) registers

These pre-registers are used to set the speed magnification rate.
RMG is the register for PRMG.



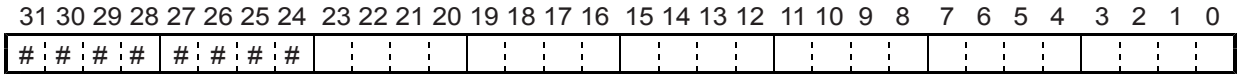
The setting range is 1 to 4,095.
Sets the relationship between the speed register PRFL (RFL), PRFH (RFH) values and the operation speeds.
The actual operation speed [pps] is a product of the speed magnification rate and the speed register setting.

[Setting example when the reference clock is 19.6608 MHz]

Setting	Speed magnification rate	Operation speed setting range [pps]	Setting	Speed magnification rate	Operation speed setting range [pps]
3999 (0F9Fh)	0.3	0.3 to 4,914.9	59 (003Bh)	20	20 to 327,660
2399 (095Fh)	0.5	0.5 to 8,191.5	23 (0017h)	50	50 to 819,150
1199 (04AFh)	1	1 to 16,383	11 (000Bh)	100	100 to 1,638,300
599 (0257h)	2	2 to 32,766	5 (0005h)	200	200 to 3,276,600
239 (00EFh)	5	5 to 81,915	2 (0002h)	400	400 to 6,553,200
119 (0077h)	10	10 to 163,830	1 (0001h)	600	600 to 9,829,800

8-3-7. PRDP (RDP) registers

These pre-registers are used to set a ramping-down point (deceleration start point) for positioning operations.
RDP is the 2nd register for PRDP.



Bits marked with a "#" symbol are ignored when written and change their setting when read according to the setting of MSDP (bit 13) in the PRMD register.

MSDP	Setting details	bit #	
0	Offset for automatically set values. When a positive value is entered, the PCL will start deceleration earlier and the FL speed range will be used longer. When a negative value is entered, the PCL will start deceleration later and will not reach the FL speed.	Same as bit 23.	-8,388,608 to +8,388,607
1	When number of pulses left drops to less than a set value, the motor on that axis starts to decelerate.	0	0 to +8,388,607

8-3-8. PRMD (RMD) registers

These pre-registers are used to set the operation mode.
RMD is the register for PRMD.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	MPCS	MSDP	METM	MCCE	MSMD	MINP	MSDE	0				MOD			

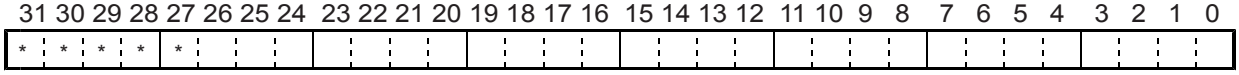
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	MCD0	MCDE	0	MADJ	MSPO	MSPE	MAX3	MAX2	MAX1	MAX0	MSY1	MSY0	MSN1	MSN0

Bits	Bit name	Description
Setting basic operation mode		
0 to 6	MOD	Set operation mode. 000 0000 (00h): Continuous positive rotation controlled by command control. 000 1000 (08h): Continuous negative rotation controlled by command control. 000 0001 (01h): Continuous operation controlled by pulsar (PA/PB) input. 000 0010 (02h): Continuous operation controlled by external signal (+DR/-DR) input. 001 0000 (10h): Positive rotation zero return operation. 001 1000 (18h): Negative rotation zero return operation. 100 0001 (41h): Positioning operation (specify the incremental target position) 100 0111 (47h): Timer operation 101 0001 (51h): Positioning operation controlled by pulsar (PA/PB) input. 101 0110 (56h): Positioning operation controlled by external signal (+DR/-DR) input. 110 0010 (62h): Continuous linear interpolation 110 0011 (63h): Linear interpolation
7	Not defined	(Always set 0)
Optical setting items		
8	MSDE	0: SD input will be ignored. (Checking can be done with RSTS in sub status) 1: Decelerates (deceleration stop) by turning ON the input.
9	MINP	0: Delay using an INP input will be possible. (Checking can be done with RSTS in sub status) 1: Completes operation by turning ON the INP input.
10	MSMD	Specify an acceleration/deceleration type for high speed feed. (0: Linear accel/decel. 1: S-curve accel/decel.)
11	MCCE	1: Stop counting output pulses on COUNTER1 and 2. This is used to move a mechanical part without changing the PLC control position When the counter input selection (RENV3: CIS1, CIS2) is set to EA/EB, the PCL will not stop counting when this bit is set.
12	METM	Specify the operation complete timing. (0: End of cycle. 1: End of pulse.) When selecting continuous operation using the pre-register, select "end of cycle."
13	MSDP	Specify the ramping-down point for high speed feed. (0: Automatic setting. 1: Manual setting.) Effective for positioning operations and linear interpolation feeding. When automatic setting is selected, set PRUR = PRDR and PRUS = PRDS.
14	MPCS	1: While in automatic operation, control the number of pulses after the PCS input is turned ON. (Override 2 for the target position.)
15	Not defined	(Always set 0)
16 to 17	MSN0 to 1	When you want to control an operation block, specify a sequence number using 2 bits. By reading the main status (MSTSW), a sequence number currently being executed (SSC0 to 1) can be checked. Setting the sequence number does not affect the operation.

Bits	Bit name	Description
18 to 19	MSY0 to 1	After writing a start command, the LSI will start an axis synchronization operation based on other timing. 00: Start immediately. 01: The PCL starts on a $\overline{\text{CSTA}}$ input (or command 06h, 2Ah). 10: Start with an internal synchronous start signal. 11: Start when a specified axis stops moving.
20 to 23	MAX0 to 3	Specify an axis to check for an operation stop when the value of MSY 0 to 1 is 11. Setting examples 0001: Starts when the X axis stops. 0010: Starts when the Y axis stops. 0100: Starts when the Z axis stops. 1000: Starts when the U axis stops. 0101: Starts when both the X and Z axes stop. 1111: Starts when all axes stop.
24	MSPE	1:Deceleration stop or immediate stop by $\overline{\text{CSTP}}$ input. This is used for a simultaneous stop with another axis when this other axis stops with an error.
25	MSPO	1: Outputs a $\overline{\text{CSTP}}$ (simultaneous stop) signal when stopping due to an error.
26	MADJ	Specify an FH correction function. (0: ON. 1: OFF.)
27	Not defined	(Always set 0)
28	MCDE	1: Decelerates when $\overline{\text{CSD}}$ input goes LOW. Set this bit to 1 to decelerate simultaneously with other axes.
29	MCDO	1: Outputs a LOW on the $\overline{\text{CSD}}$ terminal when decelerating or at FL constant speed.
30 to 31	Not defined	(Always set 0.)

8-3-9. PRIP (RIP) registers

This is a pre-register used to specify the number of pulses for the main axis feed in linear interpolation (the absolute value of the longest feed axis is set as the PRMV value)
RIP is the register for PRIP.



- When MOD (bits 0 to 6) of the PRMD register are set as shown below, the register is enabled.
  110 0010 (62h): Continuous linear interpolation (continuous operation with the linear interpolation ratio).
  110 0011 (63h): Linear interpolation.
- Setting range: 0 to +134,217,727

8-3-10. PRUS (RUS) registers

These pre-registers are used to specify the S-curve range of the S-curve acceleration.
RUS is the register for PRUS.

The normal setting range is 1 to 8,191.
When 0 is entered, the value of (PRFH - PRFL)/2 will be calculated internally and applied.

8-3-11. PRDS (RDS) registers

These pre-registers are used to specify the S-curve range of the S-curve deceleration.
RDS is the register for PRDS.

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
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*													

The normal setting range is 1 to 8,191.
When 0 is entered, the value of (PRFH - PRFL)/2 will be calculated internally and applied.
Note: Specify the same value for the PRUS register when automatic setting of the ramp down point is selected (MSDP = 0).

8-3-12. RENV1 register

This register is used for Environment setting 1. This is mainly used to set the specifications for input/output terminals.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERCL	EPW2	EPW1	EPW0	EROR	EROE	ALML	ALMM	ORGL	SDL	SDLT	SDM	ELM	PMD2	PMD1	PMD0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMSK	PCSM	INTM	DTMF	DRF	FLTR	DRL	PCSL	LTCL	INPL	FTM1	FTM0	STPM	STAM	ETW1	ETW0

Bits	Bit name	Description
0 to 2	PMD0 to 2	Specify OUT output pulse details PMD 2～0 / When feeding in a positive direction: OUT output, DIR output / When feeding in a negative direction: OUT output, DIR output 0 0 0: OUT High, DIR Low 0 0 1: OUT High, DIR Low 0 1 0: OUT Low, DIR High 0 1 1: OUT Low, DIR High 1 0 0: OUT High, DIR High 1 0 1: OUT, DIR 1 1 0: OUT, DIR 1 1 1: OUT Low, DIR Low
3	ELM	Specify the process to occur when the EL input is turned ON. (0: Immediate stop. 1: Deceleration stop.) Note 1, 2
4	SDM	Specify the process to occur when the SD input is turned ON. (0: Deceleration only. 1: Deceleration and stop.)
5	SDLT	Specify the latch function of the SD input. (0: OFF. 1: ON.) Turns ON when the SD signal width is short. When the SD input is OFF while starting, the latch signal is reset. The latch signal is also reset when SDLT is 0.
6	SDL	Specify the SD signal input logic. (0: Negative logic. 1: Positive logic.)
7	ORGL	Specify the ORG signal input logic. (0: Negative logic. 1: Positive logic.)
8	ALMM	Specify the process to occur when the ALM input is turned ON. (0: Immediate stop. 1: Deceleration stop.) Note 2
9	ALML	Specify the ALM signal input logic. (0: Negative logic. 1: Positive logic.)

Bits	Bit name	Description
10	EROE	1: Automatically outputs an ERC signal when the axis is stopped immediately by a +EL, -EL, ALM, or $\overline{\text{CEMG}}$ input signal. However, the ERC signal is not output when a deceleration stop occurs on the axis.
11	EROR	1: Automatically output the ERC signal when the axis completes a zero return.
12 to 14	EPW0 to 2	Specify the pulse width of the ERC output signal. (CLK=19.6608MHz) 000: 12 μsec 100: 13 msec 001: 102 μsec 101: 52 msec 010: 409 μsec 110: 104 msec 011: 1.6 msec 111: Level output
15	ERCL	Specify the ERC signal output logic. (0: Negative logic. 1: Positive logic.)
16 to 17	ETW0 to 1	Specify the ERC signal OFF timer time. (CLK=19.6608MHz) 00: 0 μsec 01: 12 μsec 10: 1.6 msec 11: 104 msec
18	STAM	Specify the $\overline{\text{CSTA}}$ signal input type. (0: Level trigger. 1: Edge trigger.)
19	STPM	Specify a stop method using $\overline{\text{CSTP}}$ input. (0: Immediate stop. 1: Deceleration stop.) Note 2
20 to 21	FTM 0 to 1	Select features of +EL, -EL, SD, ORG, ALM, and INP filters. 00: Pulse length shorter than 3.2 μsec are ignored. (When CLK=19.6608MHz) 01: Pulse length shorter than 25 μsec are ignored. (When CLK=19.6608MHz) 10: Pulse length shorter than 200 msec are ignored. (When CLK=19.6608MHz) 11: Pulse length shorter than 1.6 msec are ignored. (When CLK=19.6608MHz)
22	INPL	Specify the INP signal input logic. (0: Negative logic. 1: Positive logic.)
23	LTCL	Specify the operation edge for the LTC signal. (0: Falling. 1: Rising)
24	PCSL	Specify the PCS signal input logic. (0: Negative logic. 1: Positive logic.)
25	DRL	Specify the +DR, -DR signal input logic. (0: Negative logic. 1: Positive logic.)
26	FLTR	1: Apply a filter to the +EL, -EL, SD, ORG, ALM, or INP inputs. When a filter is applied, signal pulses shorter than the pulse length specified by FTM0 to 1 are ignored.
27	DRF	1: Apply a filter on the +DR, -DR, or PE inputs. When a filter is applied, signals pulses shorter than 32 msec (CLK=19.6608MHz) are ignored.
28	DTMF	1: Turn OFF the direction change timer (0.2 msec) function.
29	INTM	1: Mask an INT output. (Changes the interrupt circuit.)
30	PCSM	1: Only allow the PCS input on the local axis $\overline{\text{CSTA}}$ signal.
31	PMSK	1: Masks output pulses

Note1: When a deceleration stop (ELM = 1) has been specified to occur when the EL input turns ON, the axis will start the deceleration when the EL input is turned ON. Therefore, the axis will stop by passing over the EL position. In this case, be careful to avoid collisions of mechanical systems.

Note 2: When deceleration stop is selected, this bit remains ON until the PCL decelerates and stops. The PCL determines whether it has stopped normally or not according to the stop timing. Therefore, if an error stop signal is input while decelerating with high speed positioning, the PCL may determine whether the stop was normal. In this case, the PCL will continue to the next operation without canceling the data stored in the pre-registers. If a constant error stop signal is input, the PCL will not continue to the next operation and it will stop with an error.

8-3-13. RENV2 register

This is a register for the Environment 2 settings. Specify the function of the general-purpose port, EA/EB input, and PA/PB input.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POFF	EOFF	CSP0	P7M0	P6M0	P5M0	P4M1	P4M0	P3M1	P3M0	P2M1	P2M0	P1M1	P1M0	P0M1	P0M0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	IEND	ORM	EZL	EZD3	EZD2	EZD1	EZD0	PDIR	PINF	PIM1	PIM0	EDIR	EINF	EIM1	EIM0

Bits	Bit name	Description
0 to 1	P0M0 to 1	Specify the operation of the P0/FUP terminals 00: General-purpose input 01: General-purpose output 10: Output the FUP (acceleration) signal. 11: Output the FUP (acceleration) signal.
2 to 3	P1M0 to 1	Specify the operation of the P1/FDW terminals 00: General-purpose input 01: General-purpose output 10: Output the FDW (deceleration) signal with negative logic. 11: Output the FDW (deceleration) signal with positive logic.
4 to 5	P2M0 to 1	Specify the operation of the P2/MVC terminal. 00: General-purpose input 01: General-purpose output 10: Output the MVC (low speed feeding) signal with negative logic. 11: Output the MVC (low speed feeding) signal with positive logic.
6 to 7	P3M0 to 1	Specify the operation of the P3/CP1 terminals. 00: General-purpose input 01: General-purpose output 10: Output the CP1 (satisfied the Comparator 1 conditions) signal with negative logic. 11: Output the CP1 (satisfied the Comparator 1 conditions) signal with positive logic.
8 to 9	P4M0 to 1	Specify the operation of the P4/CP2 terminals. 00: General-purpose input 01: General-purpose output 10: Output the CP2 (satisfied the Comparator 2 conditions) signal with negative logic. 11: Output the CP2 (satisfied the Comparator 2 conditions) signal with positive logic.
10	P5M	Specify the operation of the P5 terminals. 0: General-purpose input 1: General-purpose output
11	P6M	Specify the operation of the P6 terminals. 0: General-purpose input 1: General-purpose output.
12	P7M	Specify the operation of the P7 terminals. 0: General-purpose input 1: General-purpose output
13	CSPO	1: When the RMD/MSP0 = 1, the PCL will output a $\overline{\text{CSTP}}$ when stopped with a command.
14	EOFF	1: Disables EA/EB input. (Also disables input error detection.)
15	POFF	1: Disables PA/PB input. (Also disables input error detection.)
16 to 17	EIM0 to 1	Specify the EA/EB input operation. 00: Multiply a 90° phase difference by 1 (Count up when the EA input phase is ahead.) 01: Multiply a 90° phase difference by 2 (Count up when the EA input phase is ahead.) 10: Multiply a 90° phase difference by 4 (Count up when EA input phase is ahead.) 11: Count up when the EA signal rises, count down when the EB signal falls.

Bits	Bit name	Description
18	EINF	1: Apply a noise filter to EA/EB/EZ input. Ignores pulse inputs less than 3 CLK signal cycles long.
19	EDIR	1: Reverse the counting direction of the EA/EB inputs.
20 to 21	PIM0 to 1	Specify the PA/PB input operation. 00: Multiply a 90° phase difference by 1 (Count up when the PA input phase is ahead.) 01: Multiply a 90° phase difference by 2 (Count up when the PA input phase is ahead.) 10: Multiply a 90° phase difference by 4 (Count up when PA input phase is ahead.) 11: Count up when the EA signal rises, count down when the PB signal falls.
22	PINF	1: Apply a noise filter to PA/PB input. Note 3. Ignore pulse inputs less than 3 CLK signal cycles long.
23	PDIR	1: Reverse the counting direction of the PA/PB inputs.
24 to 27	EZD0 to 3	Specify an EZ count value to be used for zero return. 0000 (1st time) to 1111 (16th time)
28	EZL	Specify EZ signal input logic. (0: Falling edge. 1: Rising edge.)
29	ORM	Select a zero return method. 0: Zero return operation 0 - Immediately stops by turning the ORG input from OFF to ON. (Decelerates and stops when at high speed.) - COUNTER reset timing: When the ORG input changes from OFF to ON. 1: Zero return operation 1 - When the PCL is feeding at constant speed, after the ORG input turns from OFF to ON it will stop immediately by counting up the EZ signals. When the PCL is feeding at high speed, it will decelerate by turning the ORG input from OFF to ON and then immediately stop by counting up the EZ signals. - COUNTER reset timing: When counting up the EZ signals.
30	IEND	1: Outputs an INT signal when stopping, regardless of whether the stop was normal or due to an error.
31	Not defined	(Always specify 0.)

8-3-14. RENV3 register

This register holds environment setting 3. Specify the counter function, latch function, and simultaneous start function.

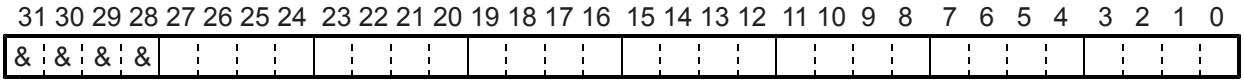
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2S1	C2S0	C1S1	C1S0	C2RM	CU2R	L0F2	CU2L	C1RM	CU1R	L0F1	CU1L	CU2H	CU1H	CIS2	CIS1

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	SYI1	SYI0	SYO3	SYO2	SYO1	SYO0

Bit	Bit name	Description
0	CIS1	Enable input counting on COUNTER1 0: Output pulse 1: EA/EB input
1	CIS2	Enable input counting on COUNTER2 0: EA/EB input 1: Output pulse
2	CU1H	1: Stops counting by COUNTER1.
3	CU2H	1: Stops counting by COUNTER2.
4	CU1L	1: Resets COUNTER1 while latching the contents of COUNTER1.
5	LOF1	1: Stop latching the contents of COUNTER1 with the LTC input. (Only effective for software.)
6	CU1R	1: Latches (and resets) COUNTER1 when a zero return operation is complete.
7	C1RM	1: Set COUNTER1 to ring counter operation using Comparator 1.
8	CU2L	1: Resets COUNTER2 while latching the contents of COUNTER2.
9	LOF2	1: Stop latching the contents of COUNTER2 with the LTC input. (Only effective for software.)
10	CU2R	1: Latches (and resets) COUNTER2 when a zero return operation is complete.
11	C2RM	1: Set COUNTER2 to ring counter operation using Comparator 2.
12 to 13	C1S0 to 1	Select a comparison method for Comparator 1 00: Turn the comparator function off. 01: RCMP1 data = Comparison counter 10: RCMP1 data > Comparison counter 11: RCMP1 data < Comparison counter
14 to 15	C2S0 to 1	Select a comparison method for Comparator 2 00: Turn the comparator function off. 01: RCMP2 data = Comparison counter 10: RCMP2 data > Comparison counter 11: RCMP2 data < Comparison counter
16 to 19	SYO0 to 3	Select the output timing for the internal synchronizing signal. 0001: When the Comparator 1 conditions are met. 0010: When the Comparator 2 conditions are met. 1000: When starting acceleration. 1001: When ending acceleration. 1010: When starting deceleration. 1011: When ending deceleration. Others: Do not output the internal synchronizing signal.
20 to 21	SYI0 to 1	Specify which axis will provide the PCL with the internal synchronization signal. 00: Internal synchronizing signal output by the X axis. 01: Internal synchronizing signal output by the Y axis. 10: Internal synchronizing signal output by the Z axis. 11: Internal synchronizing signal output by the U axis
22 to 31	Not defined	(Always set to 0.)

8-3-15. RCUN1 register

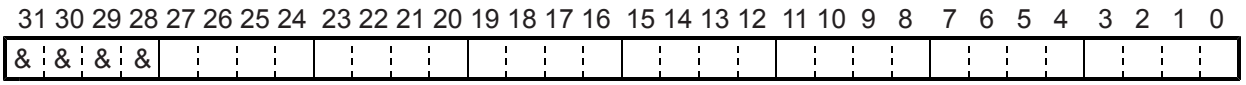
This register is used to set and read COUNTER1.
Setting range: -134,217,728 to +134,217,727



For details about the counters, see section "11-10. Counters."

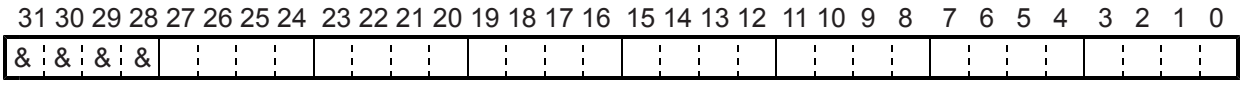
8-3-16. RCUN2 register

This register is used to set and read COUNTER2.
Setting range: -134,217,728 to +134,217,727



8-3-17. RCMP1 register

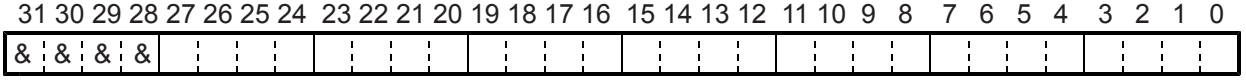
Specify the comparison data for Comparator 1.
Setting range: -134,217,728 to +134,217,727



For details about the counters, see section "11-11. Counters."

8-3-18. RCMP2 register

Specify the comparison data for Comparator 2.
Setting range: -134,217,728 to +134,217,727



8-3-19. RIRQ register

Enables event interruption cause.

Bits set to 1 that will enable an event interrupt for that event.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	IRSA	IRDR	IRSD	IROL	IRLT	IRC2	IRC1	IRDE	IRDS	IRUE	IRUS	IRNM	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	0	0	0

Bit	Bit name	Description
0	IREN	Stopping normally.
1	IRNM	When enabled to write to the pre-register.
2	IRUS	Starting acceleration.
3	IRUE	When ending acceleration.
4	IRDS	When starting deceleration.
5	IRDE	When ending deceleration.
6	IRC1	When Comparator 1 conditions are met.
7	IRC2	When Comparator 2 conditions are met.
8	IRLT	When latching the count value with an LTC signal input. (When LOF1 = LOF2 = 1 in RENV3, an interrupt will not occur.)
9	IROL	When latching the count value with an ORG signal input.
10	IRSD	When the SD input is ON. (Even when the SD input is disabled by setting MSDE = 0 in the PRMD register, an interrupt will occur.)
11	IRDR	When the ±DR (PA,PB) input is changed. (When $\overline{PE}$ = H, the interrupt will not occur.)
12	IRSA	When the $\overline{CSTA}$ input is ON.
13 to 31	Not defined	(Always set to 0.)

8-3-20. RLTC1 register

Latched data for COUNTER1. (Read only.)

The contents of COUNTER1 are copied when triggered by the LTC, an ORG input, or an LTCH command.

Data range: -134,217,728 to +134,217,727

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
&	&	&	&																												

For details about the counters, see section "11-10. Counters."

8-3-21. RLTC2 register

Latched data for COUNTER2 (Read only.)

The contents of COUNTER2 are copied when triggered by the LTC, an ORG input, or an LTCH command.

Data range: -134,217,728 to +134,217,727

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
&	&	&	&																												

8-3-22. RSTS register

The extension status can be checked. (Read only.)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SINP	SDIN	SLTC	SDRM	SDRP	SEZ	SERC	SPCS	SEMG	SSTP	SSTA	SCD	CND3	CND2	CND1	CND0

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	0	SDIR

Bit	Bit name	Description
0 to 3	CND0 to 3	Reports the operation status. 0000: Under stopped condition 0001: Waiting for DR input 0010: Waiting for $\overline{\text{CSTA}}$ input 0011: Waiting for an internal synchronous signal 0100: Waiting for another axis to stop. 0101: Waiting for a completion of ERC timer 0110: Waiting for a completion of direction change timer 1000: Waiting for PA/PB input. 1010: Feeding at FL low speed. 1011: Accelerating 1100: Feeding at FH low speed. 1101: Decelerating 1110: Waiting for INP input. Others: (controlling start/stop)
4	SCD	When the $\overline{\text{CSD}}$ input signal is ON, this bit becomes 1.
5	SSTA	Becomes 1 when the $\overline{\text{CSTA}}$ input signal is turned ON.
6	SSTP	Becomes 1 when the $\overline{\text{CSTP}}$ input signal is turned ON.
7	SEMG	Becomes 1 when the $\overline{\text{CEMG}}$ input signal is turned ON.
8	SPCS	Becomes 1 when the PCS input signal is turned ON.
9	SERC	Becomes 1 when the ERC input signal is turned ON.
10	SEZ	Becomes 1 when the EZ input signal is turned ON.
11	SDRP	Becomes 1 when the +DR (PA) input signal is turned ON.
12	SDRM	Becomes 1 when the -DR (PB) input signal is turned ON.
13	SLTC	Becomes 1 when the LTC input signal is turned ON.
14	SDIN	Becomes 1 when the SD input signal is turned ON. (Status of SD input terminal.)
15	SINP	Becomes 1 when the INP input signal is turned ON.
16	SDIR	Operation direction (0: Positive direction, 1: Negative direction)
17 to 31	Not defined	(Always set to 0.)

8-3-23. REST register

Used to check the error interrupt cause. (Read only.)
The corresponding bit will be "1" when that item has caused an error interrupt.
This register is reset when read.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ESPE	ESEE	ESP0	ESSD	ESEM	ESSP	ESAL	ESML	ESPL

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	0	0

Bit	Bit name	Description
0	ESPL	Stopped by the +EL input being turned ON.
1	ESML	Stopped by the -EL input being turned ON.
2	ESAL	Stopped by turning the ALM input ON, or when an ALM input occurs while stopping.
3	ESSP	Stopped by the $\overline{\text{CSTP}}$ input being turned ON.
4	ESEM	Stopped by the $\overline{\text{CEMG}}$ input being turned ON, or when an ALM input occurs while stopping.
5	ESSD	Decelerated and stopped by the SD input being turned ON.
6	ESPO	An overflow occurred in the PA/PB input buffer counter.
7	ESEE	An EA/EB input error occurred. (Does not stop)
8	ESPE	A PA/PB input error occurred. (Does not stop)
9 to 31	Not defined	(Always set to 0.)

8-3-24. RIST register

This register is used to check the cause of event interruption. (Read only.)
When an event interrupt occurs, the bit corresponding to the cause will be set to 1.
This register is reset when read.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	ISSA	ISMD	ISPD	ISSD	ISOL	ISLT	ISC2	ISC1	ISDE	ISDS	ISUE	ISUS	ISNM	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	0	0	0	0

Bit	Bit name	Description
0	ISEN	Stopped automatically.
1	ISNM	Available to write operation to the pre-register.
2	ISUS	Starting acceleration.
3	ISUE	Ending acceleration.
4	ISDS	Starting deceleration.
5	ISDE	Ending deceleration.
6	ISC1	The comparator 1 conditions were met.
7	ISC2	The comparator 2 conditions were met.
8	ISLT	The count value was latched by an LTC input.
9	ISOL	The count value was latched by an ORG input.
10	ISSD	The SD input turned ON.
11	ISPD	The +DR (PA) input changed.
12	ISMD	The -DR (PB) input changed.
13	ISSA	The $\overline{\text{CSTA}}$ input turned ON.
14 to 31	Not defined	(Always set to 0.)

8-3-25. RPLS register

This register is used to check the value of the positioning counter (number of pulses left for feeding). (Read only.)
At the start of positioning operation, this value will be the absolute value in the RMV register. Each pulse that is output will decrease this value by one.
Data range: 0 to 134,217,728

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
0	0	0	0																												

8-3-26. RSPD register

This register is used to check the EZ count value and the current speed. (Read only.)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	AS13	AS12	AS11	AS10	AS9	AS8	AS7	AS6	AS5	AS4	AS3	AS2	AS1	AS0

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	ECZ3	ECZ2	ECZ1	ECZ0

Bit	Bit name	Description
0 to 13	AS0 to 13	Read the current speed as a step value (same units as for RFL and RFH). When stopped the value is 0.
14 to 15	Not defined	(Always set to 0.)
16 to 19	ECZ0 to 3	Read the count value of EZ input that is used for a zero return.
20 to 31	Not defined	(Always set to 0.)

8-3-27. RSDC register

This register is used to check the automatically calculated ramping-down point value for the positioning operation. (Read only.)

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
0	0	0	0	0	0	0	0																								

The setting range will vary with the method used to set the ramp down point.
When automatic setting is selected, the available range is (the automatic set value + RDP set value). It is expressed using 24 bits which are equal to -8,388,608 to +8,388,607. The value changes with the acceleration/deceleration settings.
When manual setting is selected (MSDP = 1), the range is 0 to 16,777,215 plus a fixed value that is equal to the RDP set value.

9. Operation Mode

Specify the basic operation mode using the MOD area (bits 0 to 6) in the RMD (operation mode) register.

9-1. Continuous operation mode using command control

This is a mode of continuous operation. A start command is written and operation continues until a stop command is written.

MOD	Operation method	Direction of movement
00h	Continuous operation from a command	Positive direction
08h	Continuous operation from a command	Negative direction

Stop by turning ON the EL signal corresponding to the direction of operation.
When operation direction is positive, +EL can be used. When operation direction is negative, -EL is used.
In order to start operation in the reverse direction after stopping the motion by turning ON the EL signal, a new start command must be written.

9-2. Positioning operation mode

The following 2 operation types are available for positioning operations.

MOD	Operation method	Direction of movement
41h	Positioning operation	Positive direction when PRMV ≧ 0 Negative direction when PRMV < 0
47h	Timer operation (PRMV≧0)	Positive direction (DIR = H). However, the pulse output is masked.

9-2-1. Positioning operation (MOD: 41h)

This is a positioning mode used by placing a value in the PRMV (target position) register.
The feed direction is determined by the sign set in the PRMV register.
When starting, the RMV register absolute setting value is loaded into the positioning counter (RPLS). The PCL counts down pulses with operations, and when the value of the positioning counter drops to 0, movement on the axes stops. When you set the PRMV register value to zero to start a positioning operation, the LSI will stop outputting pulses immediately.

9-2-2. Timer operation (MOD: 47h)

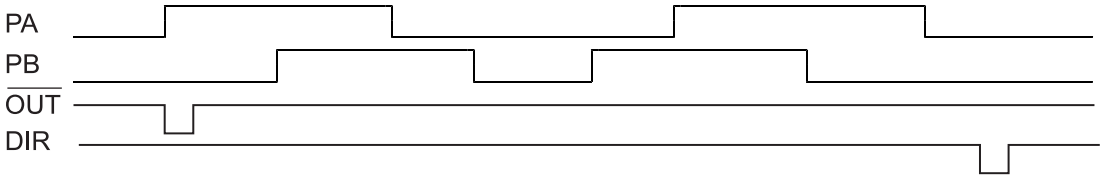
This mode allows the internal operation time to be used as a timer.
The internal effect of this operation is identical to the positioning operation. However, the LSI does not output any pulses (they are masked).
Therefore, the internal operation time using the low speed start command will be a product of the frequency of the output pulses and the RMV register setting. (Ex.: When the frequency is 1000 pps and the RMS register is set to 120 pulses, the internal operation time will be 120 msec.)
Write a positive number (1 to 134,217,727) into the RMV register. Negative numbers are treated as unsigned positive numbers.
The ±EL input signal, SD input signal, and ALM input are ignored. (These are always treated as OFF.)
The ALM input signal $\overline{\text{CSTP}}$ input signal, and $\overline{\text{CEMG}}$ input signals are effective.
The direction change timer function is disabled.
Regardless of the MINP setting (bit 9) in the RMD (operation mode) register, an operation complete delay controlled by the INP signal will not occur.
In order to eliminate deviations in the internal operation time, set the METM (bit 12) in the PRMD register to zero and use the cycle completion timing of the output pulse as the operation complete timing.

9-3. Pulsar (PA/PB) input mode

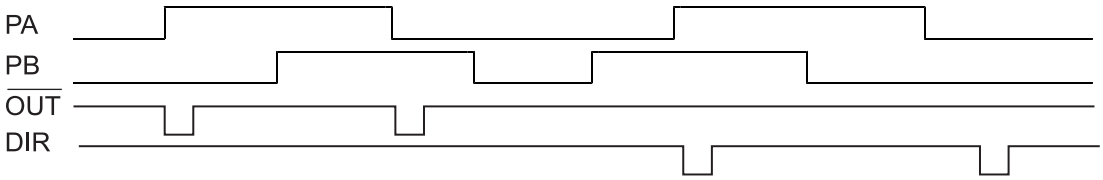
This mode is used to allow operations from a pulsar input.
In order to enable pulsar input, bring the $\overline{PE}$ terminal LOW. Set POFF in the RENV2 register to zero.
It is also possible to apply a filter on the $\overline{PE}$ input.
After writing a start command, when a pulsar signal is input, the LSI will output pulses to the OUT terminal.
Use an FL low speed start (STAFL: 50h) or an FH low speed start (STAFH: 51h).
Input pulsar signals on the PA and PB terminals. The input specification can be selected from the four possibilities below by setting the PIM0 to 1 bits in the RENV2 (environment setting 2).

- Supply a 90˚ phase difference signal (1x, 2x, or 4x).
- Supply either positive or negative pulses.

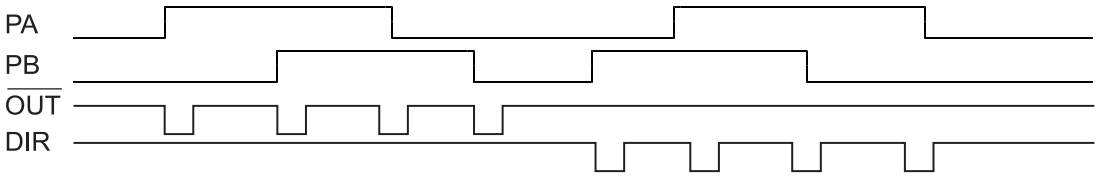
Shown below are diagrams of the operation timing. (RENV1: PMD = 100 --- When outputting 2 pulses)
1) When using 90° phase difference signals and 1x input (PIM = 00)



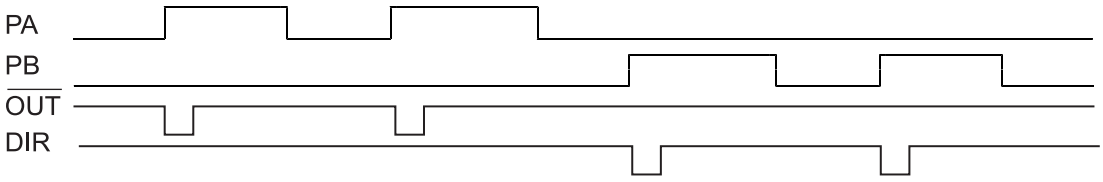
2) When using 90° phase difference signals and 2x input (PIM = 01)



3) When using 90° phase difference signals and 4x input (PIM = 10)



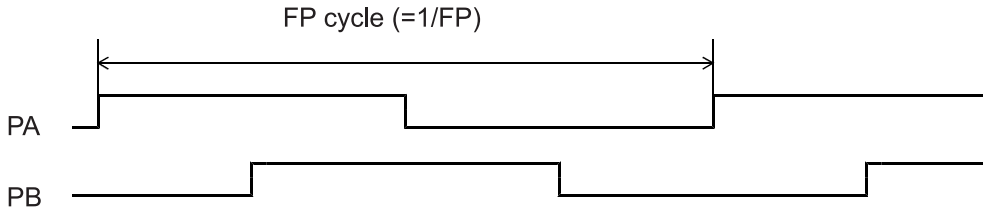
4) When using two pulse input.



The pulsar input mode is triggered by an FL constant speed start command (50h) or by an FH constant speed start command (51h).
Pulsar input causes the PCL to output pulses with some pulses from the FL speed or FH speed pulse outputs being omitted. Therefore, there may be a difference in the timing between the pulsar input and output pulses, up to the maximum internal pulse frequency.

The maximum input frequency for pulsar signals (FP) is restricted by the FL speed when an FL low speed start is used, and by the FH speed when an FH low speed start is used. The LSI outputs $\overline{INT}$ signals as errors when both the PA and PB inputs change simultaneously, or when the input frequency is exceeded, or if the input/output buffer counter (4 bits) overflows. This can be monitored by the REST (error interrupt factor) register.

FP < (speed) / (input I/F multiply value)
Example: When the pulse input setting speed is 1000 pps with a 90˚ phase difference and a 2x input multiplication rate, the input frequency on the PA terminal is less than 500 Hz.



Note: When the PA/ PB input frequency fluctuates, take the shortest frequency, not average frequency, as "Frequency of FP" above.

<Setting relationship of PA/PB input>

Specify the PA/PB input <Set to PIM0 to 1 (bit 20 to 21) in RENV2> 00: 90˚ phase difference, 1x 10: 90˚ phase difference, 4x 01: 90˚ phase difference, 2x 11: 2 sets of up or down input pulses	[RENV2] (WRITE) 23 16 - - n n - - - -
Specify the PA/PB input count direction <Set to PDIR (bit 23) in RENV2> 0: Count up when the PA phase is leading. Or, count up on the rising edge of PA. 1: Count up when the PB phase is leading. Or, count up on the rising edge of PB.	[RENV2] (WRITE) 23 16 n - - - - - - -
Enable/disable PA/PB input <Set POFF (bit 15) in RENV2> 0: Enable PA/PB input 1: Disable PA/PB input.	[RENV2] (WRITE) 15 8 n - - - - - - -
Set the ± DR, $\overline{PE}$ input filter <Set DRF (bit 27) in RENV1> 1: Insert a filter on ± DR input and $\overline{PE}$ input By setting the filter, the PCL ignores signals shorter than 32 msec.	[RENV1] (WRITE) 31 24 - - - - n - - -
Reading operation status <CND (bit 0 to 3) in RSTS> 1000 : wait for PA/ PB input.	[RSTS] (READ) 7 0 - - - - n n n n
Reading PA/PB input error <ESPE (bit 8) in REST> ESPE (bit 17) = 1: Occurs a PA/PB input error	[REST] (READ) 15 8 0 0 0 0 0 0 0 n
Reading PA/PB input buffer counter status <ESP0 (bit 6) in REST> ESPO (bit 6) = 1: Occurs an overflow.	[REST] (READ) 7 0 - n - - - - - -

* In the descriptions in the right hand column, "n" refers to the bit position. "0" refers to bit positions where it is prohibited to write any value except zero and the bit will always be zero when read.

The pulsar input mode has the following 2 operation types.
The direction of movement for continuous operation can be changed by setting the RENV2 register, without changing the wiring connections for the PA/PB inputs.

MOD	Operation mode	Direction of movement
01h	Continuous operation using pulsar input	Determined by the PA/PB input.
51h	Positioning operation using pulsar input (absolute position)	Feeds in a positive direction when PRMV ≧ 0. Feeds in a negative direction when PRMV < 0.

9-3-1. Continuous operation using a pulsar input (MOD: 01h)

This mode allows continuous operation using a pulsar input.
When PA/PB signals are input after writing a start command, the LSI will output pulses to the OUT terminal.
The feed direction depends on PA/PB signal input method and the value set in PDIR.

PA/PB input method	PDIR	Feed direction	PA/PB input
90˚ phase difference signal (1x, 2x, and 4x)	0	Positive direction	When the PA phase leads the PB phase.
		Negative direction	When the PB phase leads the PA phase.
	1	Positive direction	When the PB phase leads the PA phase.
		Negative direction	When the PA phase leads the PB phase.
2 pulse input of positive and negative pulses	0	Positive direction	PA input rising edge.
		Negative direction	PB input rising edge.
	1	Positive direction	PB input rising edge.
		Negative direction	PA input rising edge.

The PCL stops operation when the EL signal in the current feed direction is turned ON. But the PCL can be operated in the opposite direction without writing a restart command.
When stopped by the EL input, no error interrupt ($\overline{\text{INT}}$output) will occur.
To release the operation mode, write an immediate stop command (49h).

9-3-2. Positioning operations using a pulsar input (MOD: 51h)

The PCL positioning is synchronized with the pulsar input by using the PRMV setting as incremental position data.
When starting an axis, the PCL loads the RMV register value into the positioning counter.
The feed direction is determined by the sign in the PRMV register.
When PA/PB signals are input, the LSI outputs pulses and the positioning counter counts down. When the value in the positioning counter reaches zero, movement on the axis will stop and another PA/ PB input will be ignored. Set the PRMV register value to zero and start the positioning operation. The LSI will stop movement on the axis immediately, without outputting any command pulses.

9-4. External switch operation mode

This mode allows operations with inputs from an external switch.
The external switch input terminals (+DR, -DR) are common with the pulsar signal input terminal. Apply a positive direction switch signal to the PA/+DR terminal, and a negative direction switch signal to the PB/-DR terminal.
To enable inputs from an external switch, bring the $\overline{PE}$ terminal LOW.
After writing a start command, when a +DR/-DR signal is input, the LSI will output pulses to the OUT terminal.
Set the RENVI (environment 1) register to specify the output logic of the ±DR input signal. The $\overline{INT}$ signal can be set to send an output when ±DR input is changed. If $\overline{PE}$ = L, the PCL will output pulses regardless of the operation mode selected.
The RSTS (extension status) register can be used to check the operating status and monitor the ±DR input.
It is also possible to apply a filter to the ±DR or $\overline{PE}$ inputs.

Set the input logic of the +DR/-DR signals <Set DRL (bit 25) in RENV1 > 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 31 24 - - - - - - n -
Applying a ±DR or $\overline{PE}$ input filter <Set DRF (bit 27) in RENV1> 1: Apply a filter to PA, PB, $\overline{PE}$ inputs When a filter is applied, pulses shorter than 32 msec will be ignored.	[RENV1] (WRITE) 31 24 - - - - n - - -
Setting an event interrupt cause <Set IRDR (bit 11) in RIRQ> 1: Output the $\overline{INT}$ signal when ±DR signal changed input.	[RIRQ] (WRITE) 15 8 0 0 0 0 n - - -
Reading the event interrupt cause <ISPD (bit 11) and ISMD (bit 12) in RIST> ISPD (bit 11) = 1: When the +DR signal input changes. ISMD (bit 12) = 1: When the -DR signal input changes.	[RIST] (READ) 15 8 0 0 - n n - - -
Read operation status < CND (bits 0 to 3) in RSTS> 0001: Waiting for a DR input	[RSTS] (READ) 7 0 - - - - n n n n
Reading the ±DR signal <SDRP (bit 11) and SDRM (bit 12) in RSTS> SDRP = 0: +DR signal is OFF SDRP = 1: +DR signal is ON SDRM = 0: -DR signal is OFF SDRM = 1: -DR signal is ON	[RSTS] (READ) 15 8 - - - n n - - -

The external switch operation mode has the following two forms

MOD	Operation mode	Direction of movement
02h	Continuous operation using an external switch.	Determined by +DB, - DR input.
56h	Positioning operation using an external switch.	Determined by +DB, - DR input.

9-4-1. Continuous operation using an external switch (MOD: 02h)

This mode is used to operate an axis only when the DR switch is ON.
After writing a start command, turn the +DR signal ON to feed the axis in the positive direction, turn the -DR signal ON to feed the axis in the negative direction, using a specified speed pattern.
By turning ON an EL signal for the feed direction, movement on the axis will stop. However, the axis can feed in the reverse direction.
An error interrupt ($\overline{INT}$ output) will not occur.
To end this operation mode, write an immediate stop command (49h).
If the axis is being fed with high speed commands (52h, 53h), movement on the axis will decelerate and stop when the DR input turns OFF. If the DR input for reverse direction turns ON while decelerating, movement on the axis will decelerate and stop. Then it will resume in the opposite direction.

[Setting example]

1) Bring the $\overline{PE}$ input LOW.
2) Specify RFL, RFH, RUR, RDR, and RMG (speed setting).
3) Enter "0000010" for MOD (bits 0 to 6) in the RMD (operation mode) register
4) Write a start command (50h to 53h).

CND (bits 0 to 3) of the RSTS (extension status) register will wait for "0001: DR input."
In this condition, turn ON the +DR or -DR input terminal. The axis will move in the specified direction using the specified speed pattern as long as the terminal is kept ON.

9-4-2. Positioning operation using an external switch (MOD: 56h)

This mode is used for positioning based on the DR input rising timing.
When started, the data in the RMV register is loaded into the positioning counter. When the DR input is ON, the LSI will output pulses and the positioning counter will start counting down pulses. When the positioning counter value reaches zero, the PCL stops operation.
Even if the DR input is turned OFF or ON again during the operation, it will have no effect on the operation. If you make the REMV register value 0 and start a positioning operation, the PCL will stop operation immediately without outputting any command pulses.
Turn ON the +DR signal to feed in the positive direction. Turn ON the -DR signal to feed in the negative direction.
By turning ON the EL signal for the feed direction, movement on the axis will stop. However, the axis can be feed in the reverse direction.
An error interrupt ($\overline{INT}$ output) will not occur.

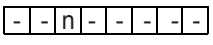
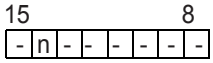
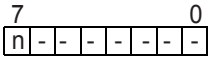
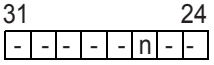
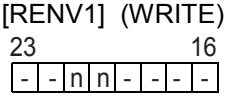
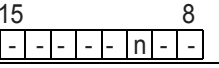
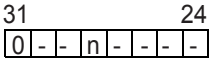
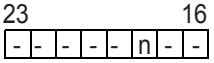
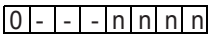
9-5. Zero return operation mode

Zero return operation varies with the MOD setting of the PRMD register, the ORM settings of the RENV2 register and the type of start command, as follows:

MOD	ORM	Command	Operation description
10h	0	50h	Feeds in a positive direction at a constant FL speed and stops immediately when the ORG input changes from OFF to ON.
		51h	Feeds in a positive direction at a constant FH speed and stops immediately when the ORG input changes from OFF to ON.
		53h	Starts and accelerates from the FL to the FH speed in a positive direction; starts deceleration when the ORG input changes from OFF to ON. When the PCL has decelerated to the FL speed, it stops feeding pulses. Also, if the PCL completes its deceleration to FL speed by a signal from the SD input before the ORG input changes, the PCL will stop immediately when the ORG input changes from OFF to ON.
	1	50h	Feeds in a positive direction at a constant FL speed, after the ORG input changes from OFF to ON. The PCL stops immediately after counting the specified number of EZ input signals.
		51h	Feeds in a positive direction at a constant FH speed after the ORG input changes from OFF to ON. The PCL stops immediately after counting the specified number of EZ input signals.
		53h	Starts and accelerates from FL to FH speed in a positive direction. Starts to decelerate when the ORG input changes from OFF to ON. After counting the specified number of EZ input signals, the PCL stops. Also, if the PCL completes its deceleration to FL speed by a signal from the SD input before the ORG input changes, the PCL will stop soon after the ORG input changes from OFF to ON, once it has counted the specified number of EZ input signals.
18h	0	50h	Feeds in a negative direction at a constant FL speed and stops immediately when the ORG input changes from OFF to ON.
		51h	Feeds in a negative direction at a constant FH speed and stops immediately when the ORG input changes from OFF to ON.
		53h	Starts and accelerates from the FL to the FH speed in a negative direction; starts deceleration when the ORG input changes from OFF to ON. When the PCL has decelerated to the FL speed, it stops feeding pulses. Also, if the PCL completes its deceleration to FL speed by a signal from the SD input before the ORG input changes, the PCL will stop immediately when the ORG input changes from OFF to ON.
	1	50h	Feeds in a negative direction at a constant FL speed, after the ORG input changes from OFF to ON. The PCL stops immediately after counting the specified number of EZ input signals.
		51h	Feeds in a negative direction at a constant FH speed after the ORG input changes from OFF to ON. The PCL stops immediately after counting the specified number of EZ input signals.
		53h	Starts and accelerates from the FL to the FH speed in a negative direction; starts deceleration when the ORG input changes from OFF to ON. After counting the specified number of EZ inputs, the PCL stops. Also, if the PCL completes its deceleration to FL speed by a signal from the SD input before the ORG input changes, the PCL will stop soon after the ORG input changes from OFF to ON, once it has counted the specified number of EZ input signals.

Depending on the operation method, the zero position operation uses the ORG or EZ inputs.
Specify the input logic (ORGL) of the ORG input signal in the RENV1 (environment 1) register. This register's terminal status can be monitored with an SSTSW (sub status) command (SORG).
Specify the input logic and the number for EZ to count up of the EZ input signal in the RENV2 (environment 2) register. Status of this terminal can be monitored at the RSTS (extension status) register (SEZ).

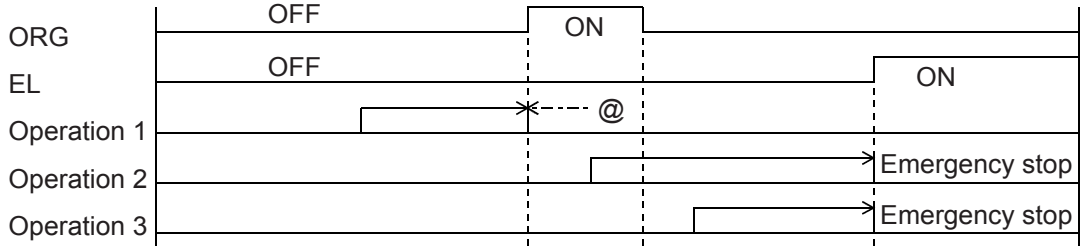
You can apply an input filter to the ORG input signal by setting the FLTR bit in the RENV2 register. To enable the EZ input signal, set the EINF bit in the RENV1 register.

Selection of the zero return operation mode <ORM (bit 29) in RENV2> 0: Use only the ORG signal. 1: Use the ORG signal and EZ signals.	[RENV1] (WRITE) 
Reading the ORG signal <SORG (bit 14) in SSTSW> 0: Turn OFF the ORG signal. 1: Turn ON the ORG signal.	[SSTSW] (READ) 
Select input logic of the ORG signal <ORGL (bit 7) in RENV1> 0: Negative logic. 1: Positive logic.	[RENV1] (WRITE) 
Set the ORG, SD input filter <FLTR (bit 26) in RENV1> 1: Apply a noise filter to the ±EL, SD, ORG, ALM, and INP inputs. When the filter is applied, signals which are shorter than the FTM pulse length will be ignored.	[RENV1] (WRITE) 
Specify a time constant for the input filter <FLM (bit 20, 21) in RENV1> 00: 3.2 µs 10: 200 µs 01: 25 µs 11: 1.6 ms	[RENV1] (WRITE) 
Reading the EZ signal <SEZ (bit 10) in RSTS> 0: Turn OFF the EZ signal. 1: Turn ON the EZ signal.	[RSTS] (READ) 
Set the input logic for the EZ signal <EZL (bit 28) in RENV2> 0: Rising edge. 1: Falling edge.	[RENV2] (WRITE) 
Apply an input filter to EA, EB, and EZ <EINF (bit 18) in RENV2> 1: Apply a noise filter to these inputs. Signals that are shorter than a CLK 3 cycle will be ignored.	[RENV2] (WRITE) 
Specify an EZ count amount <EZD0 to 3 (bits 24 to 27) in RENV2> Specify the number of EZ pulses needed to qualify for a zero return completion. Specify the value (Number of pulses-1) in bits EZD0 to 3. Enter a number from 0 to 15.	[RENV2] (WRITE) 

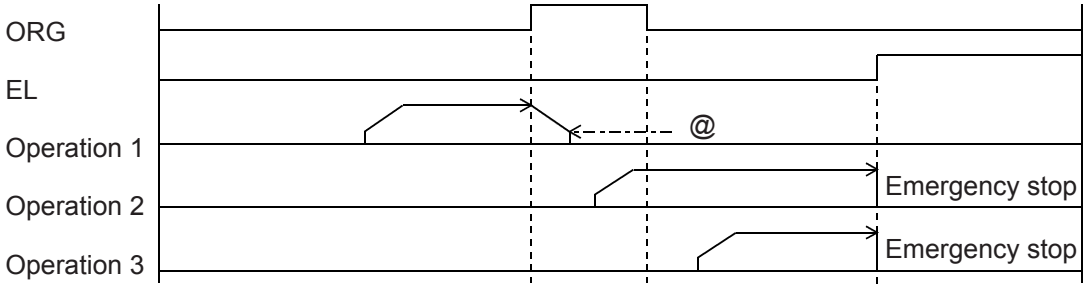
When a zero return is complete, the LSI can latch (and reset) the counter and output an ERC (deflection counter clear) signal.
The RENV3 register is used to set the basic zero return method. That is, whether or not to reset the counter when the zero return is complete. Specify whether or not to output the ERC signal in the RENV1 register.
For details about the ERC signal, see 11-5-2, "ERC signal."

9-5-1. Zero return operation 0 (ORM = 0)

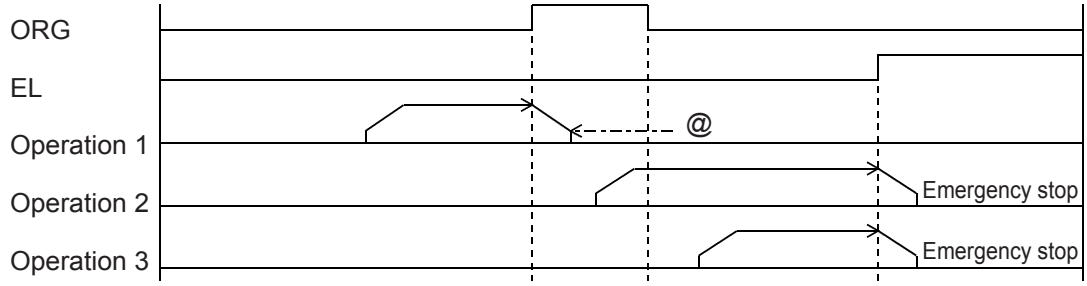
□ Low speed operation <Sensor: EL (ELM = 0), ORG>



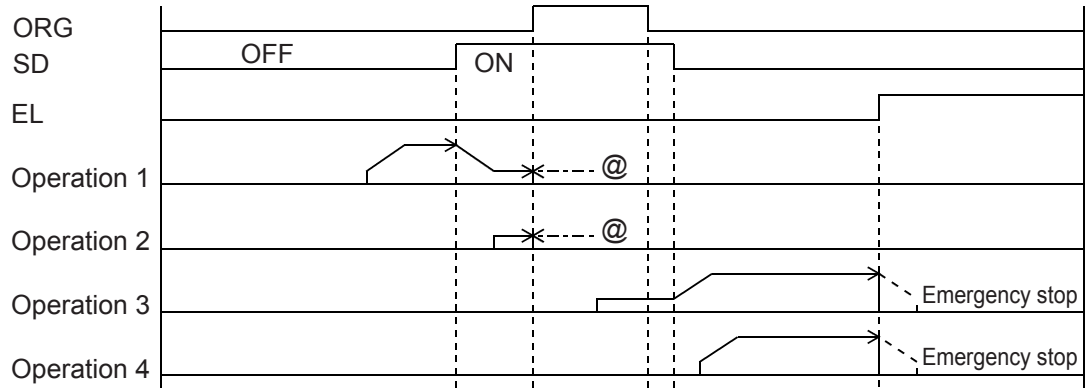
■ High speed operation <Sensor: EL (ELM = 0), ORG>
Even if the axis stops normally, it may not be at the zero position. However, COUNTER2 (mechanical position) provides a reliable value.



■ High speed operation <Sensor: EL (ELM = 1), ORG>
Even if the axis stops normally, it may not be at the zero position. However, COUNTER2 (mechanical position) provides a reliable value.



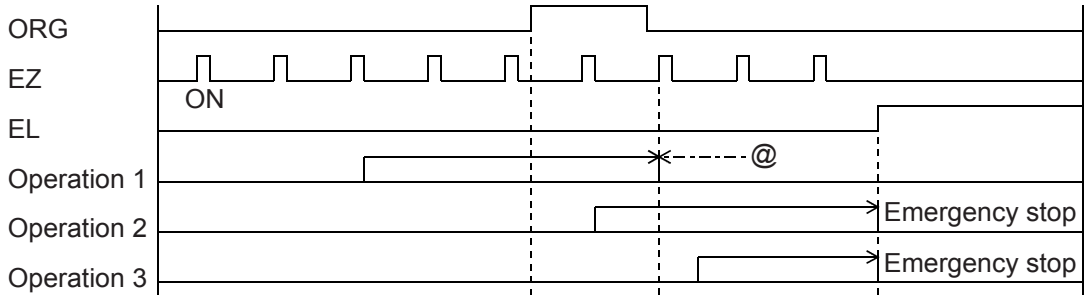
■ High speed operation <Sensor: EL (ELM = 1), SD (SDM = 0, SDLT = 0), ORG>



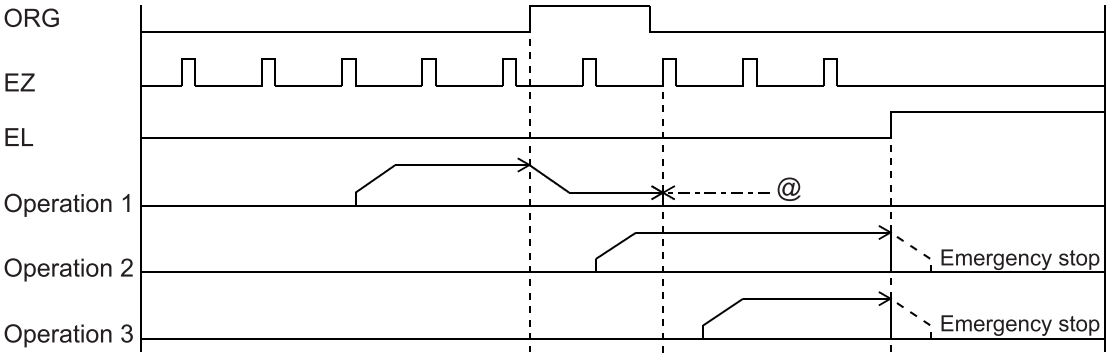
Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for the zero stopping position.

9-5-2. Zero return operation 1 (ORM=1)

□ Low speed operation <Sensor: EL (ELM = 0), ORG, EZ (EZD = 0001)>



■ High speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>



Note: Positions marked with "@" reflect ERC signal output timing when "Automatically output an ERC signal" is selected for the zero stopping position.

9-6. Linear interpolation operation

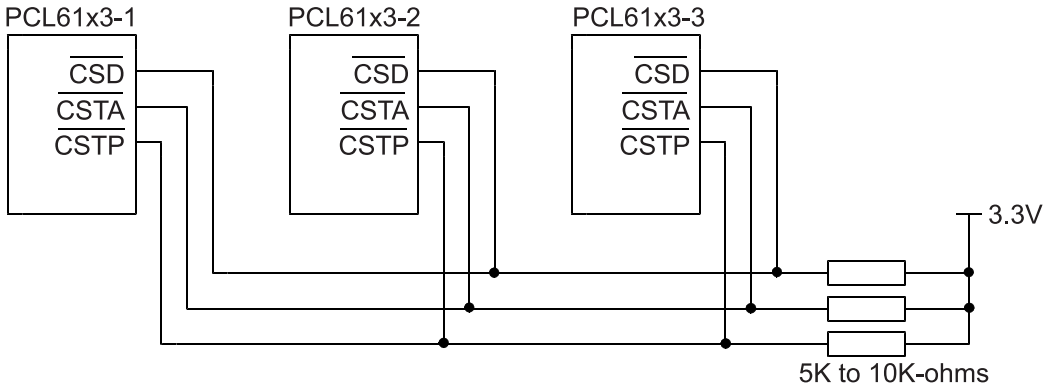
9-6-1.Outline of interpolation operation

Using one or more PCLs, you can operate linear interpolation feed.

MOD	Operation mode
62h	Continuous linear interpolation
63h	Linear interpolation

Just like in the linear interpolation mode, in continuous linear interpolation the PCL feeds multiple axes at a specified rate. However, PCL operations can still be started and stopped with commands, the same as in linear interpolation.
With the linear interpolation, the PCL automatically stops after the specified feed amount.

The linear interpolation circuit in this PCL interpolates between a dummy axis associated with each axis and the actual axis.
By entering maximum feed amount data for each and every dummy axis, the PCLs will execute an indirect linear interpolation between the axes.
As each interpolated axis operates independently, the start timing, deceleration timing, and error stop timing must be matched between the axes.
When you want to use multiple PCLs and have them interpolate for each other, connect CSD, CSTA, and CSTP terminals on each PCL to each other and provide a pull up resistor (5 k to 10 k-ohms) on VDD (3.3v) for each signal line.
Even when performing interpolation within a single PCL, a pull up resistor is required.



9-6-2. Interpolation procedures

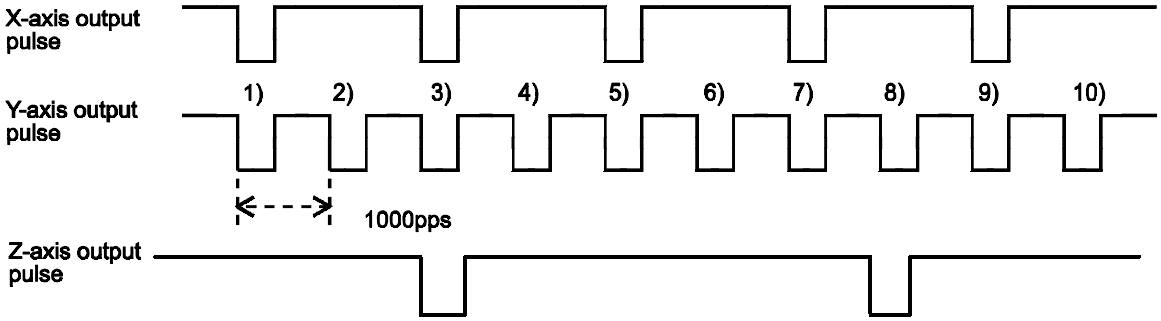
1) Enter a feed amount with a sign in the PRMV register for each axis. The sign specifies the feed direction.
2) Enter the absolute value of the PRMV (from the axis with the largest feed amount) in the PRIP registers of all the axes that will perform an interpolation.
3) Specify the speed pattern (PRFL, PRFH, PRUR, PRMG, PRDP, PRDR, PRUS, PRDS) that will be used for the axis with the maximum feed amount for all the axes that will perform an interpolation.
When you want to specify a synthesized speed, obtain the speed factor for the axis with the maximum feed amount by calculation from the CPU. Then, enter this speed for all the axes that will perform an interpolation.
4) If any of the axes performing an interpolation stops due to an error, and if you want to stop all the other axes performing an interpolation, set the MSPE and MSP0 bits in the PRMD register on those axes to 1.
5) When you want to interpolate using acceleration/deceleration, set the MCDE and MSD0 bits in the PRMD register to 1 for all the axes that will perform an interpolation.
6) When you want to perform an interpolation using only one PCL, specify the axis to interpolate in the upper byte (COMB1) when writing the start command.

When you want to perform an interpolation using multiple PCLs, set the MSY0 and 1 bits in the PRMD register to 01, on all the axes that will perform an interpolation. Then write a postponed start command (waiting for a CSTA input).

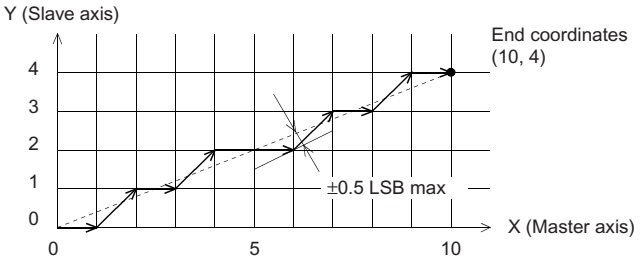
After setting all the axes that will perform an interpolation for a postponed start, write the CSTA output command 06h (simultaneous start) to any of these axes and all of the axes that will perform the interpolation will start at the same time.
Other axes that are not interpolating can be operated independently.

[Setting example] Use the settings below and write a start command (0751h). The PCL will output pulses with the timing shown in the figure below. Entering values in the blank items will not affect operation.

Setting	X axis	Y axis	Z axis
MOD	63h	63h	63h
PRMV value	5	10	2
Operation speed	1000 pps	1000 pps	1000 pps



[Precision of linear interpolation]
As shown in the figure on the right, linear interpolation executes an interpolation from the current coordinates to the end coordinates. The positional precision of a specified line during linear interpolation will be ±0.5 LSB throughout the interpolation range.
"LSB" refers to the minimum feed unit for the PRMV register setting. It corresponds to the resolution of the mechanical system. (distance between tick marks in the figure on the right.)



9-6-3. Operation during interpolation

- Acceleration/deceleration operations
  In addition to constant speed operation, these axes can accelerate/decelerate (linear acceleration or S-curve), and a ramp down point with an automatic setting is also available.
  However, the following restrictions apply:
  1) The settings for MSDP and MADJ in the PRMD register must be identical for all the axes that will perform an interpolation.
  2) If you want to use the manual setting (MSDP = 1) for the ramp down point, enter the value for the longest feed axis in the PRDP registers of all the axes that will perform an interpolation.

- Error stop
  If any of the axes performing the interpolation stops on an error, the other axes performing an interpolation will also stop by the CSTP function (simultaneous stop function). Axes that did not encounter an error will show ESSP = 1 when the REST register is read (error stop cause). This allows you to identify the axis that had an error.

- SD input
  When SD input is enabled (MSDE (bit 8) in the PRMD register is set to 1) by processing the CSD terminal, and if the SD input turns ON either of the axes, both axes will decelerate or decelerate and stop.

◆ Continuous interpolation

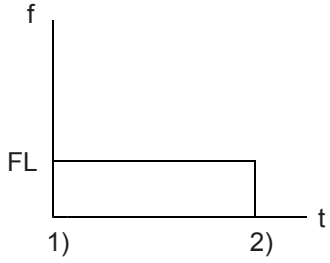
The PCL can use the pre-register to make a continuous linear interpolation.

Continuous interpolation refers to linear interpolation operations performed successively.

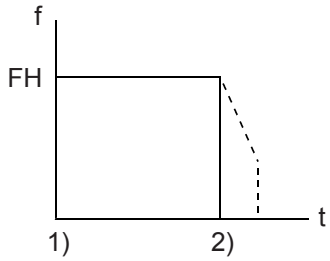
An example of the settings for continuous interpolation using the pre-register is shown in section 11-11-1, "Start triggered by a stop on another axis."

10. Speed patterns

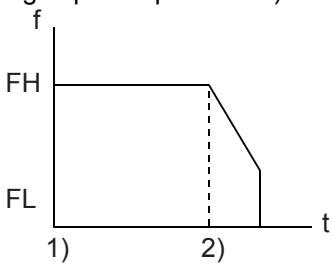
10-1. Speed patterns

<table>
<tr><th>Speed pattern</th><th>Continuous mode</th><th>Positioning operation mode</th></tr>
<tr><td>FL low speed operation
 
</td><td>1) Write an FL speed start command (50h).

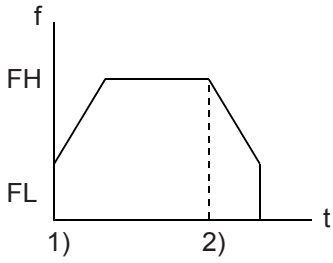
2) Stop feeding by writing an immediate stop (49h) or deceleration stop (4Ah) command.</td><td>1) Write an FL speed start command (50h).

2) Stop feeding when the positioning counter reaches zero, or by writing an immediate stop (49h) or deceleration stop (4Ah) command.</td></tr>
<tr><td rowspan="2">FH low speed operation
 
</td><td>1) Write an FH speed start command (51h).

2) Stop feeding by writing an immediate stop command (49h).</td><td>1) Write an FH speed start command (51h).

2) Stop feeding when the positioning counter reaches zero, or by writing an immediate stop (49h) command.</td></tr>
<tr><td colspan="2">* When the deceleration stop command (4Ah) is written to the register, the PCL starts deceleration.</td></tr>
<tr><td rowspan="2">High speed operation 1)
 
</td><td>1) Write high speed start command 1 (52h).

2) Start deceleration by writing a deceleration stop command (4Ah).</td><td>1) Write high speed start command 1 (52h).

2) Start deceleration when a ramping-down point is reached or by writing a deceleration stop command (4Ah).</td></tr>
<tr><td>* When the deceleration stop command (49h) is written to the register, the PCL immediately stops operation.</td><td>* When positioning with a high speed start command 1 (52h), <u>the ramping-down point is fixed to the manual setting,</u> regardless of the setting for MSDP (bit 13) in the PRMD. If the ramping-down point setting (PRDP) is zero, the axis will stop immediately.</td></tr>
<tr><td rowspan="2">High speed operation 2)
 
</td><td>1) Write high speed command 2 (53h).

2) Start deceleration by writing a deceleration stop command (4Ah).</td><td>1) Write high speed start command 2 (53h).

2) Start deceleration when a ramping-down point is reached or by writing a deceleration stop command (4Ah).</td></tr>
<tr><td>* When the deceleration stop command (49h) is written to the register, the PCL immediately stops operation.</td><td>* If the ramping-down point is set to manual (MSDP = 1 in the PRMD), and the ramping-down value (PRDP) is zero, the axis will stop immediately.</td></tr>
</table>

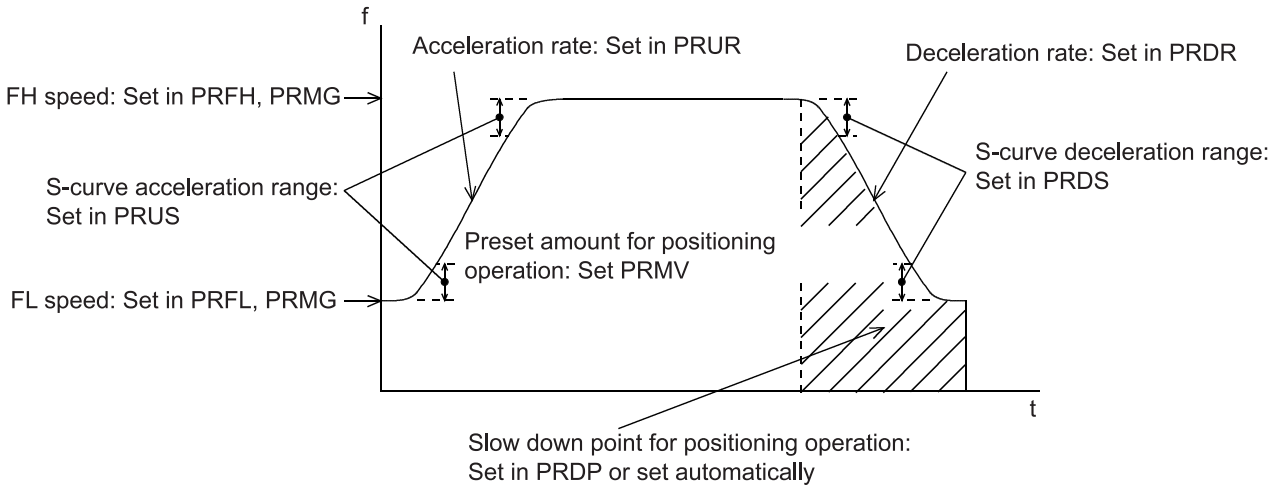
10-2. Speed pattern settings

Specify the speed pattern using the registers (pre-registers) shown in the table below.
If the next register setting is the same as the current value, there is no need to write to the register again.

Pre-register	Description	Bit length setting range	Setting range	Register
PRMV	Positioning amount	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	RMV
PRFL	Initial speed	14	1 to 16,383 (03FFFh)	RFL
PRFH	Operation speed	14	1 to 16,383 (03FFFh)	RFH
PRUR	Acceleration rate	14	1 to 16,383 (03FFFh)	RUR
PRDR	Deceleration rate Note 1	14	0 to 16,383 (03FFFh)	RDR
PRMG	Speed magnification rate	12	1 to 4,095 (0FFFh)	RMG
PRDP	Ramping-down point	24	0 to 16,777,215 (0FFFFFFh)	RDP
PRUS	S-curve acceleration range	13	0 to 8,191 (1FFFh)	RUS
PRDS	S-curve deceleration range	13	0 to 8,191 (1FFFh)	RDS

Note 1: If PRDR is set to zero, the deceleration rate will be the value set in the PRUR.

[Relative position of each register setting for acceleration and deceleration factors]



♦ PRFL: FL speed setting register (14-bit)

Specify the speed for FL low speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 16,383 (3FFFh).
The speed will be calculated from the value in PRMG.

$$\text{FL speed [pps]} = \text{PRFL} \times \frac{\text{Reference clock frequency [Hz]}}{(\text{PRMG} + 1) \times 16384}$$

♦ PRFH: FH speed setting register (14-bit)

Specify the speed for FH low speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 16,383 (3FFFh).
When used for high speed operations (acceleration/deceleration operations), specify a value larger than PRFL.
The speed will be calculated from the value placed in PRMG.

$$\text{FH speed [pps]} = \text{PRFH} \times \frac{\text{Reference clock frequency [Hz]}}{(\text{PRMG} + 1) \times 16384}$$

♦ PRUR: Acceleration rate setting register (14-bit)

Specify the acceleration characteristic for high speed operations (acceleration/deceleration operations), in the range of 1 to 16,383 (3FFFh)

Relationship between the value entered and the acceleration time will be as follows:

1) Linear acceleration (MSMD = 0 in the PRMD register)

$$\text{Acceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRUR} + 1) \times 2}{\text{Reference clock frequency [Hz]}}$$

2) S-curve without a linear range (MSMD=1 in the PRMD register and PRUS register =0)

$$\text{Acceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRUR} + 1) \times 4}{\text{Reference clock frequency [Hz]}}$$

3) S-curve with a linear range (MSMD=1 in the PRMD register and PRUS register >0)

$$\text{Acceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL} + 2 \times \text{PRUS}) \times (\text{PRUR} + 1) \times 2}{\text{Reference clock frequency [Hz]}}$$

♦ PRDR: Deceleration rate setting register (14-bit)

Normally, specify the deceleration characteristics for high speed operations (acceleration/deceleration operations) in the range of 1 to 16,383 (3FFFh).

To select the ramp down point auto setting (MSDP = 0 in the PRMD register), set the PRDR register the same as PRUR register setting, or enter 0.

When PRDR = 0, the deceleration rate will be the value placed in the PRUR.

The relationship between the value entered and the deceleration time is as follows.

1) Linear deceleration (MSMD = 0 in the PRMD register)

$$\text{Deceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRDR} + 1) \times 2}{\text{Reference clock frequency [Hz]}}$$

2) S-curve deceleration without a linear range (MSMD=1 in the PRMD register and PRDS register = 0)

$$\text{Deceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRDR} + 1) \times 4}{\text{Reference clock frequency [Hz]}}$$

3) S-curve deceleration with a linear range (MSMD=1 in the PRMD register and PRDS register >0)

$$\text{Deceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL} + 2 \times \text{PRDS}) \times (\text{PRDR} + 1) \times 2}{\text{Reference clock frequency [Hz]}}$$

♦ PRMG: Magnification rate register (12-bit)

Specify the relationship between the PRFL and PRFH settings and the speed, in the range of 1 to 4,095 (0FFFh). As the magnification rate is increased, the speed setting units will tend to be approximations. Normally set the magnification rate as low as possible.

The relationship between the value entered and the magnification rate is as follows.

$$\text{Magnification rate} = \frac{\text{Reference clock frequency [Hz]}}{(\text{PRMG} + 1) \times 16384}$$

[Magnification rate setting example, when the reference clock =19.6608 MHz] (Output speed unit: pps)

Setting	Magnification rate	Output speed range	Setting	Magnification rate	Output speed range
3999 (0F9Fh)	0.3	0.3 to 4,914.9	59 (003Bh)	20	20 to 327,660
2399 (095Fh)	0.5	0.5 to 8,191.5	23 (0017h)	50	50 to 819,150
1199 (04AFh)	1	1 to 16.383	11 (000Bh)	100	100 to 1,638,300
599 (0257h)	2	2 to 32,766	5 (0005h)	200	200 to 3,276,600
239 (00EFh)	5	5 to 81,915	2 (0002h)	400	400 to 6,553,200
119 (0077h)	10	10 to 163,830	1 (0001h)	600	600 to 9,829,800

The maximum output speed of this IC can be attained when the reference clock is 30 MHz, PRMG = 1, and PRFH = 16383.
In these conditions, the multiplication rate is 915.527x and the IC will output 14.999 Mpps.

♦ PRDP: Ramping-down point register (24-bits)

Specify the value used to determine the deceleration start point for positioning operations that include acceleration and deceleration.
The meaning of the value specified in the RDP changes with the "ramping-down point setting method", (MSDP) in the PRMD register.

<When set to manual (MSDP=1 in the PRMD register)>

Set the number of pulses at which to start deceleration, in the range of 0 to16,777,215 (0FFFFFFh).
When the (PRDP set value) ≧ (Number of residual pulses), the PCL will start decelerating.

Note: In order to obtain the correct manual setting value, you have to know the actual maximum speed. When there is only a small feed amount and the motor would have to decelerate while still accelerating, or if the maximum speed is automatically modified by the FH correction function, the PCL cannot calculate the manual setting value.
Therefore, in this case turn OFF the FH correction function before trying the operation.
Alternatively, you can calculate the manual FH correction and then obtain the corrected maximum speed using the following equations.

The optimum value of the ramping down position can be as follows.

1) Linear deceleration (MSMD=0 of the PRMD register)

$$\text{Optimum value [Number of pulses]} = \frac{(PRFH^2 - PRFL^2) \times (PRDR + 1)}{(PRMG + 1) \times 16384}$$

2) S-curve deceleration without a linear range (MSMD=1 in the PRMD register and the PRDS register =0)

$$\text{Optimum value [Number of pulses]} = \frac{(PRFH^2 - PRFL^2) \times (PRDR + 1) \times 2}{(PRMG + 1) \times 16384}$$

3) S-curve deceleration with a linear range (MSMD=1 in the PRMD register and the PRDS register >0)

$$\text{Optimum value [Number of pulses]} = \frac{(PRFH + PRFL) \times (PRFH - PRFL + 2 \times PRDS) \times (PRDR + 1)}{(PRMG + 1) \times 16384}$$

Start deceleration at the point when the (positioning counter value) ≦ (RDP set value).
When the value for the ramping-down point is smaller than the optimum value, the speed when stopping will be faster than the FL speed. On the other hand, if it is larger than the optimum value, the axis will feed at FL low speed after decelerating.

<When set to automatic (MSDP = 0 in the PRMD register)>

This is an offset value for the automatically set ramping-down point. Set in the range of -8,388,608 (800000h) to 8,388,607 (7FFFFFh).

When the offset value is a positive number, the axis will start deceleration at an earlier stage and will feed at the FL speed after decelerating. When a negative number is entered, the deceleration start timing will be delayed.

If the offset is not required, set to zero.

♦ PRUS: S-curve acceleration range register (13-bit)

Specify the S-curve acceleration range for S-curve acceleration/deceleration operations in the range of 1 to 8,191 (1FFFh).

The S-curve acceleration range S_{SU} will be calculated from the value placed in PRMG.

$$S_{SU}\,[pps] = PRUS \times \frac{\text{Reference clock frequency [Hz]}}{(PRMG + 1) \times 16384}$$

In other words, speeds between the FL speed and (FL speed + S_{SU}), and between (FH speed - S_{SU}) and the FH speed, will be S-curve acceleration operations. Intermediate speeds will use linear acceleration.

However, if zero is specified, "(PRFH - PRFL)/2" will be used for internal calculations, and the operation will be an S-curve acceleration without a linear component.

♦ PRDS: S-curve deceleration range setting register (13-bit)

Same as the PRUS, specify an S-curve deceleration range for the S-curve acceleration/deceleration operation between 1 and 8,191 (1FFFh).

The S-curve acceleration range S_{SD} will be calculated from the value placed in PRMG.

$$S_{SD}\,[pps] = PRDS \times \frac{\text{Reference clock frequency [Hz]}}{(PRMG + 1) \times 16384}$$

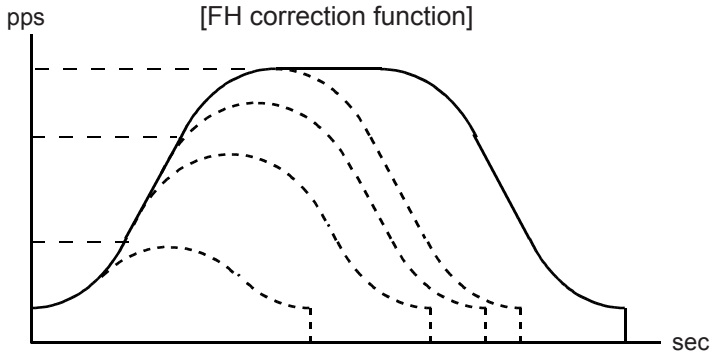
In other words, speeds between the FL speed and (FL speed + S_{SD}), and between (FH speed - S_{SD}) and the FH speed, will be S-curve deceleration operations. Intermediate speeds will use linear deceleration.

However, if zero is specified, "(PRFH - PRFL)/2" will be used for internal calculations, and the operation will be an S-curve deceleration without a linear component.

10-3. Manual FH correction

When the FH correction function is turned ON (MADJ = 0 in the PRMD register), and when the feed amount is too small for a normal acceleration and deceleration operation, the LSI will automatically lower the FH speed to eliminate triangle driving.
In addition, the ramp down point auto setting will also change according to the FH correction result.

However, the ramp down point auto setting function can only be used when the acceleration curve and deceleration curve are symmetrical. In other words, if you want to make the acceleration and deceleration curves asymmetrical, the slow down point needs to be changed to a manual setting. In order to obtain the correct manual setting value, you have to know the maximum speed.
Therefore, you have to turn OFF the FH correction function and manually correct the FH value.



Automatic correction of the maximum speed for changing the feed amount.

< To execute FH correction manually>

1) Linear acceleration/deceleration speed (MSMD=0 in the PRMD register)

$$PRMV \leqq \frac{(PRFH^2 - PRFL^2) \times (PRUR + PRDR + 2)}{(PRMG + 1) \times 16384}$$

$$PRFH \leqq \sqrt{\frac{(PRMG + 1) \times 16384 \times PRMV}{PRUR + PRDR + 2} + PRFL^2}$$

2) S-curve acceleration without linear acceleration (MSMD=1 in the PRMD and PRDS registers = 0)

When $PRMV \leqq \frac{(PRFH^2 - PRFL^2) \times (PRUR + PRDR + 2) \times 2}{(PRMG + 1) \times 16384}$

$$PRFH \leqq \sqrt{\frac{(PRMG + 1) \times 16384 \times PRMV}{(PRUR + PRDR + 2) \times 2} + PRFL^2}$$

3) S-curve acceleration/deceleration with linear acceleration/deceleration (MSMD = 1 in the PRMD register and the PRUS register > 0, PRDS register > 0)

(3)-1. When PRUS = PRDS

(i) Set up a small linear acceleration range

$$PRMV \leqq \frac{(PRFH + PRFL) \times (PRFH - PRFL + 2 \times PRUS) \times (PRUR + PRDR + 2)}{(PRMG + 1) \times 16384}$$ and

$$PRMV > \frac{(PRUS + PRFL) \times PRUS \times (PRUR + PRDR + 2) \times 4}{(PRMG + 1) \times 16384}$$

$$PRFH \leqq -PRSU + \sqrt{(PRUS - PRFL)^2 + \frac{(PRMG + 1) \times 16384 \times PRMV}{(PRUR + PRDR + 2)}}$$

(ii) Eliminate the linear acceleration/deceleration range

$$PRMV \leqq \frac{(PRUS + PRFL) \times PRUS \times (PRUR + PRDR + 2) \times 4}{(PRMG + 1) \times 16384}$$

Change to S-curve acceleration/deceleration without a linear acceleration/deceleration range (PRUS = 0, PRDS = 0),

$$PRFH \leqq \sqrt{\frac{(PRMG + 1) \times 16384 \times PRMV}{(PRUR + PRDR + 2) \times 2} + PRFL^2}$$

Reference		
PRMV: Positioning amount	PRFL: Initial speed	PRFH: Operation speed
PRUR: Acceleration rate	PRDR: Deceleration rate	PRMG: Speed magnification rate
PRUS: S-curve acceleration range	PRDS: S-curve deceleration range	

(3)-2. When PRUS < PRDS

(i) Set up a small linear acceleration/deceleration range
When

$$PRMV \leqq \frac{(PRFH+PRFL) \times \{(PRFH-PRFL) \times (PRUR + PRDR + 2) + 2 \times PRUS \times (PRUR+1) + 2 \times PRDS \times (PRDR + 1)\}}{(PRMG + 1) \times 16384}$$

and

$$PRMV > \frac{(PRDS+PRFL) \times \{PRDS \times (PRUR + 2 \times PRDR + 3) + PRUS \times (PRUR + 1)\} \times 4}{(PRMG + 1) \times 16384},$$

$$PRFH \leqq \frac{-A + \sqrt{A^2 + B}}{PRUR + PRDR + 2}$$

However, $A = PRUS \times (PRUR + 1) + PRDS \times (PRDR + 1)$
$B = \{(PRMG + 1) \times 16384 \times PRMV - 2 \times A \times PRFL + (PRUR + PRDR + 2) \times PRFL^2\} \times (PRUR + PRDR + 2)$

(ii) Eliminate the linear acceleration/deceleration range and set up a small linear acceleration section.
When

$$PRMV \leqq \frac{(PRDS + PRFL) \times \{PRDS \times (PRUR + 2 \times PRDR + 3)\} + PRUS \times (PRUR +1)\} \times 4}{(PRMG + 1) \times 16384}$$ and

$$PRMV > \frac{(PRUS + PRFL) \times PRUS \times (PRUR + PRDR + 2) \times 8}{(PRMG + 1) \times 16384}$$

Change to S-curve acceleration/deceleration without any linear acceleration/deceleration (PRUS>0, PRDS=0)

$$PRFH \leqq \frac{-A + \sqrt{A^2 + B}}{PRUR + 2 \times PRDR + 3}$$

However, $A = PRUS \times (PRUR + 1)$,
$B = \{(PRMG + 1) \times 16384 \times PRMV - 2 \times A \times PRFL + (PRUR + 2 \times PRDR + 3) \times PRFL^2\} \times (PRUR + 2 \times PRDR + 3)$

(iii) Eliminate the linear acceleration/deceleration range

$$\text{When } PRMV \leqq \frac{(PRUS + PRFL) \times PRUS \times (PRUR + PRDR + 2) \times 8}{(PRMG + 1) \times 16384}$$

Change to S-curve acceleration/deceleration without any linear acceleration/deceleration (PRUS=0, PRDS=0),

$$PRFH \leqq \sqrt{\frac{(PRMG + 1) \times 16384 \times PRMV}{(PRUR + PRDR + 2) \times 2} + PRFL^2}$$

Reference		
PRMV: Positioning amount	PRFL: Initial speed	PRFH: Operation speed
PRUR: Acceleration rate	PRDR: Deceleration rate	PRMG: Speed magnification rate
PRUS: S-curve acceleration range	PRDS: S-curve deceleration range	

(3)-3. When PRUS>PRDS

(i) Set up a small linear acceleration/deceleration range

When

$$PRMV \leqq \frac{(PRFH + RFL) \times \{(PRFH - PRFL) \times (PRUR + PRDR + 2) + 2 \times PRUS \times (PRUR + 1) + 2 \times PRDS \times (PRDR + 1)\}}{(PRMG + 1) \times 16384}$$

and

$$PRMV > \frac{(PRUS + PRFL) \times \{PRUS \times (2 \times PRUR + PRDR + 3) + PRDS \times (PRDR + 1) \times 4}{(PRMG + 1) \times 16384},$$

$$PRFH \leqq \frac{-A + \sqrt{A^2 + B}}{PRUR + PRDR + 2}$$

However, $A = PRUS \times (PRUR + 1) + PRDS \times (PRDR + 1)$,
$B = \{(PRMG + 1) \times 16384 \times PRMV - 2 \times A \times PRFL + (PRUR + PRDR + 2) \times PRFL^2\} \times (PRUR + PRDR + 2)$

(ii) Eliminate the linear acceleration section and set up a small linear deceleration range.

When

$$PRMV \leqq \frac{(PRUS + PRFL) \times \{PRUS \times (2 \times PRUR + PRDR + 3) + PRDS \times (PRDR + 1)\} \times 4}{(PRMG + 1) \times 16384} \text{ and}$$

$$PRMV > \frac{(PRDS + PRFL) \times PRDS \times (PRUR + PRDR + 2) \times 8}{(PRMG + 1) \times 16384},$$

Change to S-curve acceleration/deceleration without any linear acceleration (PRUS = 0, PRDS > 0)

$$PRFH \leqq \frac{-A + \sqrt{A^2 + B}}{2 \times PRUR + PRDR + 3}$$

However, $A = PRDS \times (PRDR + 1)$,
$B = \{(PRMG + 1) \times 16384 \times PRMV - 2 \times A \times PRFL + (2 \times PRUR + PRDR + 3) \times PRFL^2\} \times (2 \times PRUR + PRDR + 3)$

(iii) Eliminate the linear acceleration/deceleration range

$$\text{When } PRMV \leqq \frac{(PRDS + PRFL) \times PRDS \times (PRUR + PRDR + 2) \times 8}{(PRMG + 1) \times 16384}$$

Change to S-curve acceleration/deceleration without any linear acceleration/deceleration (PRUS = 0, PRDS = 0),

$$PRFH \leqq \sqrt{\frac{(PRMG + 1) \times 16384 \times PRMV}{(PRUR + PRDR + 2) \times 2} + PRFL^2}$$

Reference		
PRMV: Positioning amount	PRFL: Initial speed	PRFH: Operation speed
PRUR: Acceleration rate	PRDR: Deceleration rate	PRMG: Speed magnification rate
PRUS: S-curve acceleration range	PRDS: S-curve deceleration range	

10-4. Example of setting up an acceleration/deceleration speed pattern

Ex. Reference clock = 19.6608 MHz
When the start speed =10 pps, the operation speed =100 kpps, the accel/decl time = 300 msec, and linear acceleration/deceleration is selected.

1) Select the 10x mode for multiplier rate in order to get 100 kpps output
   PRMG = 119 (0077h)

2) Since the 10x mode is selected to get an operation speed 100 kpps,
   PRFH = 10000 (2710h)

3) In order to set a start speed of 10 pps, the rate magnification is set to the 10x mode.
   PRFL = 10 (000Ah)

4) In order to make the acceleration/deceleration time 300 msec, calculate from the equation for the acceleration time and the RUR value.

$$\text{Acceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRUR} + 1) \times 2}{\text{Reference clock frequency [Hz]}}$$

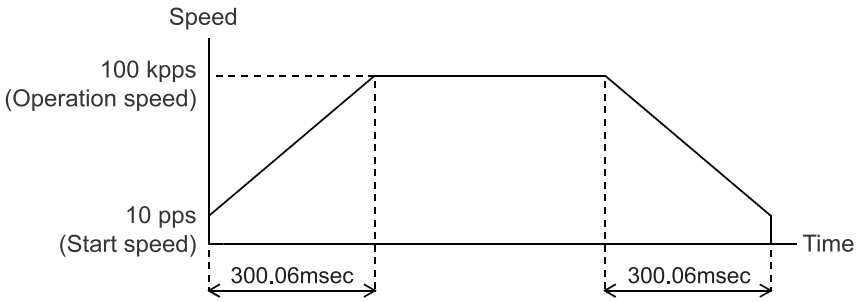
$$0.3 = \frac{(10000 - 1) \times (\text{PRUR} + 1) \times 2}{19.6608 \times 10^6}$$

PRUR = 293.94

However, since only integers can be entered for PRUR, use 293 or 294. The actual acceleration/deceleration time will be 299.04 msec if PRUR = 293, or 300.06 msec if PRUR = 29.

5) Since the acceleration and deceleration times are equal, place a 0 in the PRDR register and the deceleration rate will be the same as the value in PRUR.

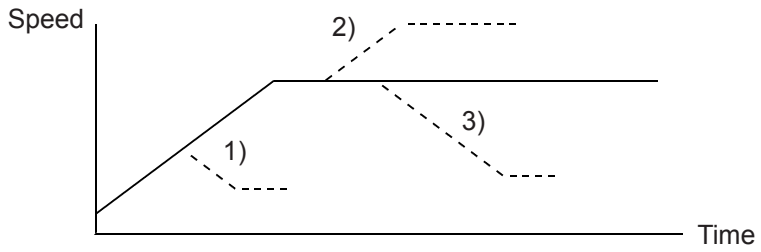
An example of the speed pattern when PRUR = 294



10-5. Changing speed patterns while in operation

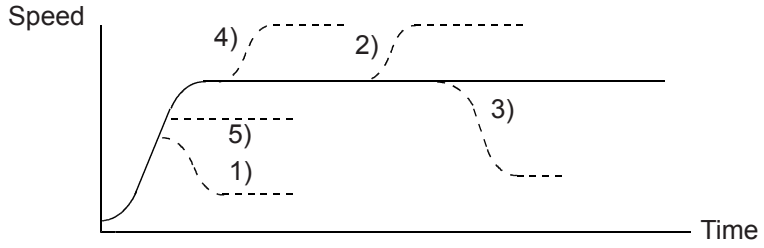
By changing the RFH, RUR, RDR, RUS, or RDS registers during operation, the speed and acceleration can be changed on the fly. However, if the ramping-down point was set to automatic (MSDP = 0 in the RDM register) for the positioning mode, do not change the values for RFL, RUR, RDR, RUS, or RDS. The automatic ramping-down point function will not work correctly.
When using S-curve acceleration/deceleration, and the ramp down point auto setting is selected, if you want to change the speed during operation, you must set PRUS = PRDS = 0. If the PCL reached the ramp down point while still accelerating and started to ramp down, it would stop feeding without decelerating to the FL speed. Therefore, in this case, you must be careful about changing the speed timing. When using linear acceleration/deceleration, you do need not to be concerned about this timing.

An example of changing the speed pattern by changing the speed, during a linear acceleration/deceleration operation



1) Use a small RFH while accelerating or decelerating the axis until it reaches the correct speed.
2), 3) Change RFH after the acceleration/deceleration is complete. The axis will continue accelerating or decelerating until it reaches the new speed.

An example of changing the speed pattern by changing the speed during S-curve acceleration/deceleration operation



1) Use a small RFH and if ((change speed) < (speed before change)) and the axis will accelerate/decelerate using an S-curve until it reaches the correct speed.
5) Use a small RFH and if ((change speed) ≧ (speed before change)) and the axis will accelerate/decelerate without changing the S-curve's characteristic until it reaches the correct speed.
4) Use a large RFH while accelerating and the axis will accelerate to the original speed entered without changing the S-curve's characteristic. Then it will accelerate again until it reaches the newly set speed.
2), 3) If RFH is changed after the acceleration/deceleration is complete, the axis will accelerate/decelerate using an S-curve until it reaches the correct speed.

11. Description of the Functions

11-1. Reset

After turning ON the power, make sure to reset the LSI before beginning to use it.
To reset the LSI, hold the $\overline{RST}$ terminal LOW while supplying at least 8 cycles of a reference clock signal.
After a reset, the various portions of the LSI will be configured as follows.

Item	Reset status (initial status) n = x, y, z, u
Internal registers, pre-register	0
Control command buffer	0
Axis assignment buffer	0
Input/output buffer	0
$\overline{INT}$ terminal	HIGH
$\overline{WRQ}$ terminal	HIGH
$\overline{IFB}$ terminal	HIGH
D0 to D7 terminals	High Z (impedance)
D8 to D15 terminals	High Z (impedance)
P0n to P7n terminals	Input terminal
$\overline{CSD}$ terminal	HIGH
$\overline{CSTA}$ terminal	HIGH
$\overline{CSTP}$ terminal	HIGH
OUTn terminal	HIGH
DIRn terminal	HIGH
ERCn terminal	HIGH
$\overline{BSYn}$ terminal	HIGH

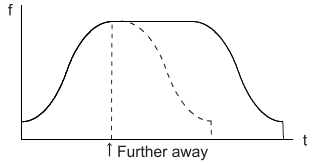
11-2. Position override

This LSI can override (change) the target position freely during operation.
However, the PCL cannot execute a position override during linear interpolation.
There are two methods for overriding the target position.

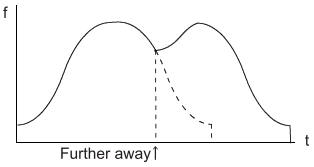
11-2-1. Target position override 1

By rewriting the target position data (RMV register value), the target position can be changed.
The starting position is used as a reference to change target position.

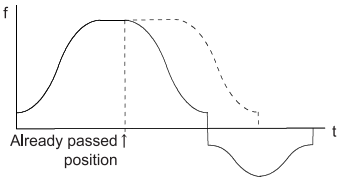
1) If the new target position is further away from the original target position during acceleration or low speed operation, the axis will maintain the operation using the same speed pattern and it will complete the positioning operation at the position specified in the new data (new RMV value).



2) If the new target position is further away from the original target position during deceleration, the axis will accelerate from the current position to FH speed and complete the positioning operation at the position specified in the new data (new RMV value).
Assume that the current speed is Fu, and when RFL = Fu, a curve of the next acceleration will be equal to a normal acceleration curv

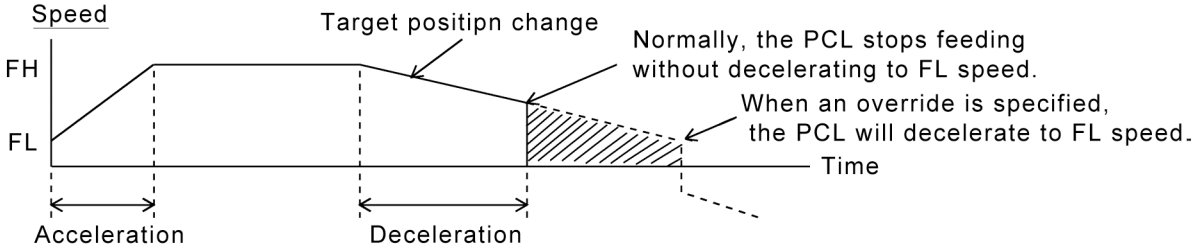


3) If the axis has already passed over the new target position, or the target position is changed to a position that is closer than the original position during deceleration, movement on the axis will decelerate and stop. Then, the movement will reverse and complete the positioning operation at the position specified in the new data (new RMV value).



The axis accelerates/decelerates only when starting in high speed. The target position data (RMV register value) can be rewritten any number of times until the positioning operation is complete.

Note1: When positioning while using acceleration/deceleration, even if the PCL cannot decelerate to the FL speed, it will stop at the specified position (placing a priority on the stop position). If the position override is applied and the PCL has to reverse feed, it will decelerate to the FL speed and then stop (placing a priority on speed).
Therefore, it may possible that when a motor reverse is caused by the position override, the motor may feed pulses that cross over the target position and then reverse back to it.



Note 2: The position override is only valid while feeding.
When the PCL receives an override command just a little before stopping a feed, it may not respond to the override command. For this reason, check SEOR in the main status after stopped.
If the override is ignored, the SEOR will become "1."

The PCL will set SEOR to "1" when it receives a command in the RMV register (90h) while feeding is stopped to allow the override command to be evaluated. Therefore, if the command is written to the RMV register while stopped, before feeding starts, the SEOR will also become "1."

When the override command is ignored, the PCL will set SEOR to "1" after stopped.
After reading the MSTS, the PCL will set SEOR to "0" within three CLK cycles.

11-2-2. Target position override 2 (PCS signal)

By making MPCS in the PRMD (operation mode) register "1," the PCL will perform positioning operations for the amount specified in the PRMV register, based on the timing of this command after the operation start (after it starts outputting instruction pulses) or on the "ON" timing of the PCS input signal. A PCS input signal can change the input logic. The PCS terminal status can be monitored using the RSTS register (extension status).

Setting pulse control using the PCS input <Set MPCS (bit 14) in PRMD> 1: Positioning for the number of pulses stored in the PRMV, starting from the time at which the PCS input signal is turned ON.	[PRMD] (WRITE) 15 8 - n - - - - - -
Setting the PCS input logic <Set PCSL (bit 24) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 31 24 - - - - - - - n
Reading the PCS signal < SPCS (bit 8) in RSTS> 0: Turn OFF PCS 1: Turn ON PCS	[RSTS] (READ) 15 8 - - - - - - - n
PCS substitution input <STAON: Operation command> Perform processes that are identical to those performed by supplying a PCS signal.	[Operation command] 28h

11-3. Output pulse control

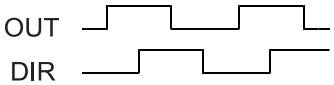
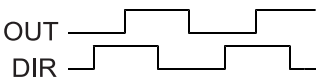
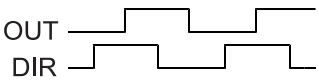
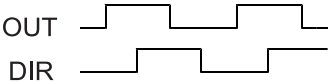
11-3-1. Output pulse mode

There are four types of common command pulse output modes and two types of 2-pulse modes, and two types of 90° phase difference mode.

Common pulse mode:	Outputs operation pulses from the OUT terminal and outputs the direction signal from the DIR terminal. (MOD = 000 to 011)
2-pulse mode:	Outputs positive direction operation pulses from the OUT terminal, and outputs negative direction operation pulses from the DIR terminal. (MOD = 100, 111)
90° phase difference mode:	This mode outputs signals from the OUT terminal and DIR terminal with a 90° phase difference. (MOD = 101, 110)

The output mode for command pulses is set in PMD (bits 0 to 2) in RENV1 (environment setting 1).
If motor drivers using the common pulse mode need a lag time (since the direction signal changes, until receiving a command pulse), use a direction change timer.
When DTMP (bit 28) in the RENV1 (environment setting 1) is set to 0, the operation can be delayed for one direction change timer unit (0.2 msec), after changing the direction identification signal.
When DTMF is 1, the PCL will output pulses 10 CLK cycles (0.5 μs) after DIR changes.

Setting the pulse output mode <Set PMD0 to 2 (bits 0 to 2) in RENV1>

PMD0 to 2	When feeding in the positive direction		When feeding in the negative direction	
	OUT output	DIR output	OUT output	DIR output
000		High		Low
001		High		Low
010		Low		High
011		Low		High
100		High	High	
101				
110				
111		Low	Low	

[RENV1] (WRITE)

7							0
-	-	-	-	-	n	n	n

Setting the direction change timer (0.2 msec) function <Set DTMF (bit 28) in RENV1>
0: ON
1: OFF

[RENV1] (WRITE)

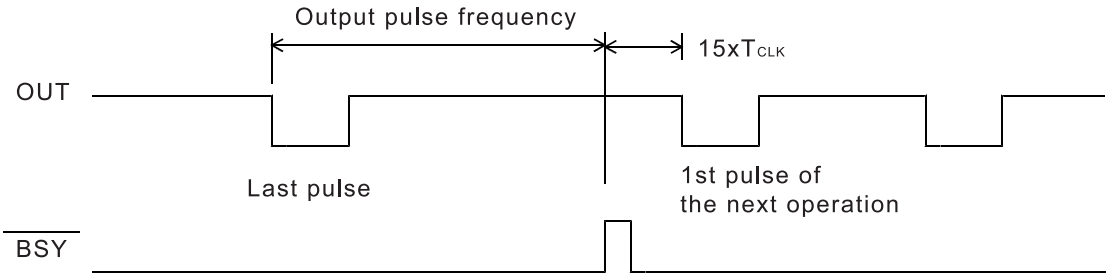
31							24
-	-	-	n	-	-	-	-

11-3-2. Control the output pulse length and operation complete timing

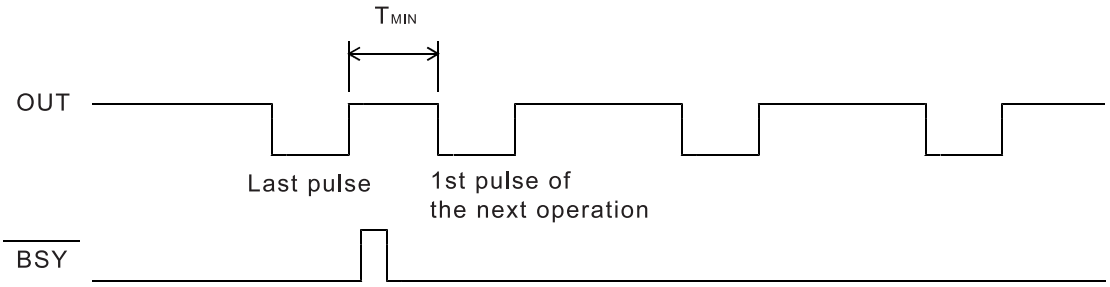
Output pulse length is a 50% duty cycle.
When the PRMG setting is an even number, the duty cycle may deviate slightly and the ON time may be shorter than the OFF time.
(Pulse ON time) / (Pulse cycle) = (PRMG set value / 2) / (PRMG set value +1)

Also, when setting METM (operation completion timing setting) in the PRMD register (operation mode), the operation complete timing can be changed.

1) When METM = 0 (the point at which the output frequency cycle is complete) in the PRMD register



2) When METM = 1 (when the output pulse is OFF) in the PRMD register



When set to "complete when the output pulse is OFF," the time interval "Min" from the last pulse until the next starting pulse output will be T_{MIN} = 17 x T_{CLK}. (T_{CLK}: Reference clock frequency)

Setting the operation complete timing <Set METM (bit 12) in PRMD> 0: At the end of a cycle of a particular output frequency 1: Complete when the output pulse turns OFF.	[RMD] (WRITE) 15 8 - - - n - - - -

11-4. Mechanical external input control

11-4-1. +EL, -EL signal

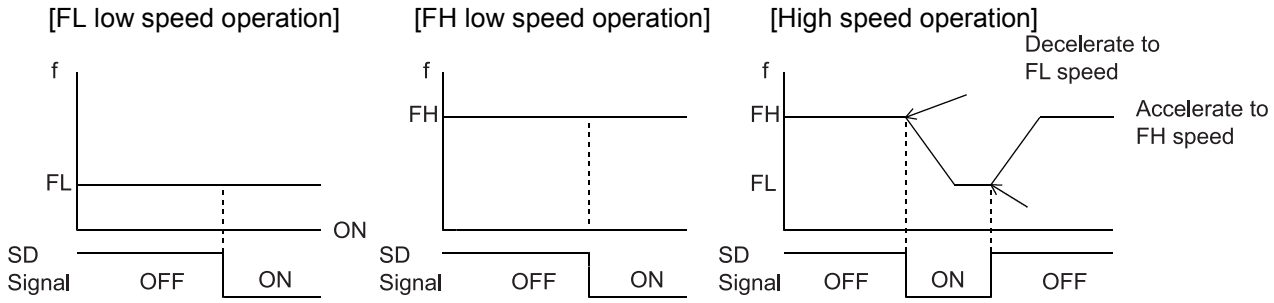
When an end limit signal (a +EL signal when feeding in the + direction) in the feed direction turns ON while operating, the axis will stop immediately or decelerate and stop. After stopping, even if the EL signal is turned OFF, the axis will remain stopped. For safety, keep the EL signal ON until the axis reaches the end of the stroke.
If the EL signal is ON when writing a start command, the axis cannot start moving in the direction of the particular EL signal that is ON.
By setting ELM in the RENV1 (environment setting 1) register, the stopping pattern for use when the EL signal is turned ON can be set to immediate stop or deceleration stop (high speed start only).
The minimum pulse width of the EL signal is 2 cycles of reference clock cycles (0.4 μs) when the input filter is OFF.
When the input filter is OFF the minimum pulse time for the EL signal is two reference clock cycles (0.1 μs). When the input filter is ON, the PCL will not respond to pulse signals shorter than the specified time.
By reading the SSTSW (sub status), you can monitor the EL signal.
By reading the REST register, you can check for an error interrupt caused by the EL signal turning ON.
When in the timer mode, this signal is ignored. Even in this case, the EL signal can be monitored by reading SSTSW (sub status).
The input logic of the EL signal can be set for each axis using the ELL input terminal.

Set the input logic of the ±EL signal <ELL input terminal> L: Positive logic input H: Negative logic input	
Stop method to when the ±EL signal turns ON <Set ELM (bit 3) in RENV1> 0: Immediate stop by turning ON the ±EL signal 1: Deceleration stop by turning ON the ±EL signal	[RENV1] (WRITE) 7 0 - - - - n - - -
Setting the ±EL input filter <Set FLTR (bit 26) in RENV1> 1: Inset filters to ±EL, SD, ORG, ALM, and INP inputs. When the filter is inserted, pulses shorter than the FTM set value are ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -
Select the input filter characteristics <Set FTM (bits 20, 21) in RENV1> 00: 3.2 μs 10: 200 μs 01: 25 μs 11: 1.6 ms	[RENV1] (WRITE) 23 16 - - n n - - - -
Reading the ±EL signal <SPEL (bit 12), SMEL (bit 13) in SSTSW> SPEL = 0:Turn OFF the +EL signal SPEL = 1: Turn ON the +EL signal SMEL = 0:Turn OFF the -EL signal SMEL = 1: Turn ON the -EL signal	[SSTSW] (READ) 15 8 - - n n - - - -
Reading the stop cause when the ±EL signal turns on <ESPL (bit 0), ESML (bit 1) in REST> ESPL = 1: Stop by turning ON the +EL signal ESML = 1: Stop by turning ON the -EL signal	[REST] (READ) 7 0 - - - - - - n n

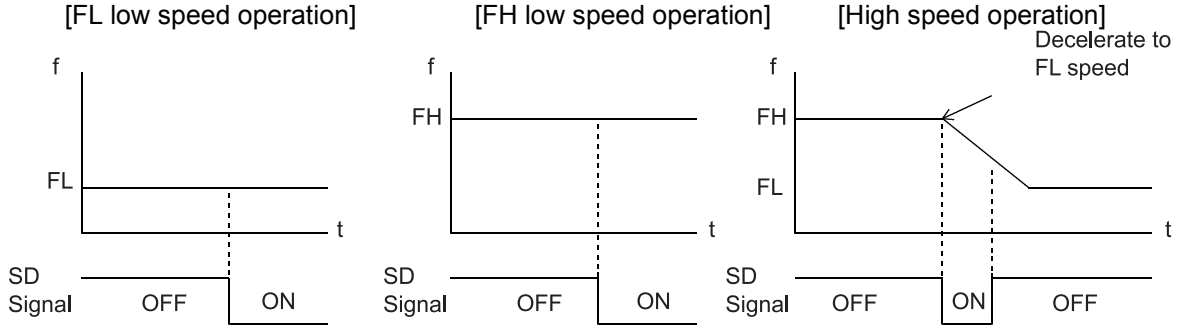
11-4-2. SD signal

If the SD signal input is disabled by setting MSDE in the PRMD register (operation mode), the SD signal will be ignored.
If the SD signal is enabled and the SD signal is turned ON while in operation, the axis will: 1) decelerate, 2) latch and decelerate, 3) decelerate and stop, or 4) latch and perform a deceleration stop, according to the setting of SDM and SDLT in the RENV1 register (environment setting 1).

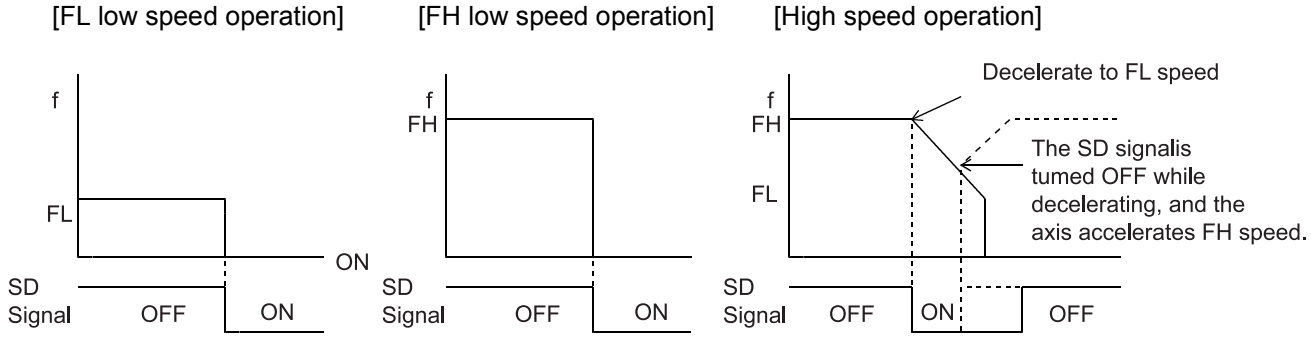
1) Deceleration < SDM (bit 4) = 0, SDLT (bit 5) = 0 in RENV1 register>
   -While feeding at low speed, the SD signal is ignored. While in high speed operation the axis decelerates to the FL speed when the SD signal is turned ON. After decelerating, or while decelerating, if the SD signal turns OFF, the axis will accelerate to the FH speed.
   - If the SD signal is turned ON when the high speed command is written, the axis will operate at FL speed. When the SD signal is turned OFF, the axis will accelerate to FH speed.



2) Latch and decelerate <SDM (bit 4) = 0, SDLT (bit 5) = 1 in RENV1 register>
- While feeding at low speed, the SD signal is ignored. While in high speed operation, decelerate to FL speed by turning the SD signal ON. Even if the SD signal is turned OFF after decelerating or while decelerating, the axis will continue moving at FL speed and will not accelerate to FH speed.
- If the SD signal is turned ON while writing a high speed command, the axis will feed at FL speed. Even if the SD signal is turned OFF, the axis will not accelerate to FH speed.

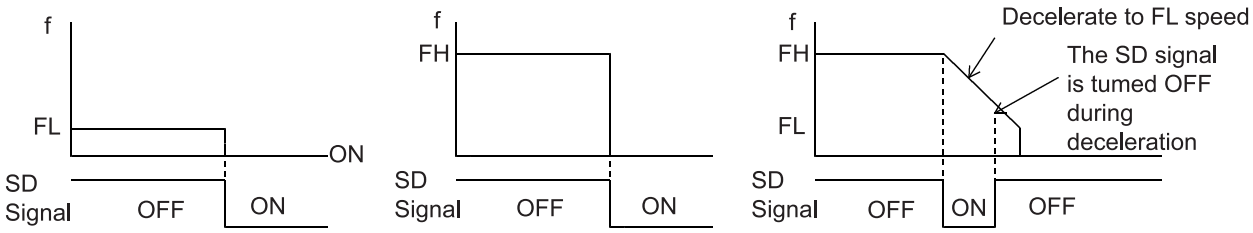


3) Deceleration stop <SDM (bit 4) = 1, SDLT (bit 5) = 0 in RENV1 register>
- If the SD signal is turned ON while in low speed operation, the axis will stop. While in high speed operation, the axis will decelerate to FL speed when the SD signal is turned ON, and then stop. If the SD signal is turned OFF during deceleration, the axis will accelerate to FH speed.
- If the SD signal is turned ON after writing a start command, the axis will complete its operation without another start.
- When stopped, the axis will output an $\overline{\text{INT}}$ signal.



4) Latched, deceleration stop <SDM (bit 4) = 1, SDLT (bit 5)=1 in RENV1>
- If the SD signal is turned ON while in low speed operation, the axis will stop. If the SD signal is turned ON while in high speed operation, the axis will decelerate to FL speed and then stop. Even if the SD signal is turned OFF during deceleration, the axis will not accelerate.
- If the SD signal is turned ON while writing a start command, the axis will not start moving and the operation will not be completed.
- While stopped, the LSI outputs an $\overline{\text{INT}}$ signal.

[FL low speed operation] [FH low speed operation] [High speed operation]



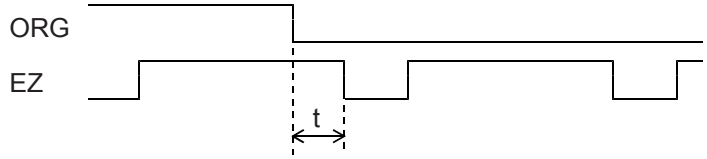
The input logic of the SD signal can be changed. If the latched input is set to accept input from the SD signal, and if the SD signal is OFF at the next start, the latch will be reset. The latch is also reset when the latch input is set to zero.
When the input filter is OFF the minimum pulse time for the SD signal is two reference clock cycles (0.1 μs). When the input filter is ON, the PCL will not respond to pulse signals shorter than the specified time.
The latch signal of the SD signal can be monitored by reading SSTSW (sub status). The SD signal terminal status can be monitored by reading RSTS (extension status). By reading the REST register, you can check for an error interrupt caused by the SD signal turning ON.

Enable/disable SD signal input <Set MSDE (bit 8) in PRMD> 0: Enable SD signal input 1: Disable SD signal input	[RMD] (WRITE) 15 8 - - - - - - - n
Input logic of the SD signal <Set SDL(bit 6) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 - n - - - - - -
Set the operation pattern when the SD signal is turned ON <Set SDM (bit 4) in RENV1> 0: Decelerates on receiving the SD signal and feeds at FL low speed 1: Decelerates and stops on receiving the SD signal	[RENV1] (WRITE) 7 0 - - - n - - - -
Select the SD signal input type <Set SDLT (bit 5) in RENV1> 0: The SD signal is level input 1: The SD signal is latch input (To release the latch, turn OFF the SD input when next start command is written or select Level input.)	[RENV1] (WRITE) 7 0 - - n - - - - -
Reading the latch status of the SD signal <SSD (bit 15) in SSTSW> 0: The SD latch signal is OFF 1: The SD latch signal is ON	[SSTSW] (READ) 15 8 n - - - - - - -
Reading the SD signal < SDIN (bit 14) in the RSTS register> 0: The SD signal is OFF 1: The SD signal is ON	[RSTS] (READ) 15 8 - n - - - - - -
Reading the cause of an $\overline{INT}$ when stopped by the SD signal <ESSD (bit 5) in RESET> 1: Deceleration stop caused by the SD signal turning ON	[REST] (READ) 7 0 - - n - - - - -
Apply an input filter to SD <Set FLTR (bit 26) in RENV1> 1: Apply a filter to the ±EL, SD, ORG, ALM and INP input By applying a filter, signals with a pulse length shorter than the FTM value will be ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -
Select the input filter characteristics <Set FTM (bits 20, 21) in RENV1> 00: 3.2 μs 10: 200 μs 01: 25 μs 11: 1.6 ms	[RENV1] (WRITE) 23 16 - - n n - - - -

11-4-3. ORG, EZ signals

These signals are enabled in the zero return modes.
When the input filter is OFF the minimum pulse time for the ORG signal is 2 reference clock cycles (0.1 μs). When the input filter is ON, the PCL will not respond to pulse signals shorter than the specified time. In addition, the ORG signal is sampled during the period that the output pulse is ON, so the ORG input must be latched ON for more than one pulse.
The input logic of the ORG signal and EZ signal can be changed using the RENV1 register and RENV 2 register.
The ORG terminal status can be monitored by reading SSTSW (sub status). The EZ terminal status can be monitored by reading the RSTS register (extension status).
For details about the zero return operation modes, see 9-5, "Zero position operation mode."

ORG signal and EZ signal timing (When the input filter is OFF) T_{CLK}: Reference clock cycle



(i) When $t \geqq 2 \times T_{CLK}$, counts.
(ii) When $T_{CLK} < t < 2 \times T_{CLK}$, counting is undetermined.
(iii) When $t \leqq T_{CLK}$, do not count.

Enabling the ORG and EZ signals <Set MOD (bits 0 to 6) in PRMD> 001 0000: Zero return in the positive direction 010 1000: Zero return in the negative direction	[PRMD] (WRITE) 7 0 0 n n n n n n n
Setting the zero return method <Set ORM (bit 29) in RENV2> 0: Use only the ORG input. 1: Use both the ORG input and EZ input.	[RENV2] (WRITE) 31 24 - - n - - - - -
Set the input logic for the ORG signal <Set ORGL (bit 7) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 n - - - - - - -
Set the ORG input filter <FLTR (bit 26) in RENV1> 1: Apply a filter to the ±EL, SD, ORG ALM, and INP input. By applying a filter, pulses shorter than the FTM set value are ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -
Setting the time constant for the input filter <Set FTM (bits 20, 21) in RENV1> 00: 3.2 μs 10: 200 μs 01: 25 μs 11: 1.6 ms	[RENV1] (WRITE) 23 16 - - n n - - - -
Read the ORG signal <SORG (bit 14) in SSTSW> 0: The ORG signal is OFF 1: The ORG signal is ON	[SSTSW] (READ) 15 8 - n - - - - - -
Set the EZ count number <Set EZD0 to 3 (bits 24 to 27) in RENV2> Set the zero return completion condition and the EZ count number for counting. Specify the value (the number to count to ñ 1) in EZD0 to 3. The setting range is 0 to 15.	[RENV2] (WRITE) 31 24 - - - - n n n n
Specify the input logic of the EZ signal <Set EZL (bit 28) in RENV2> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 31 24 - - - n - - - -
Read the EZ signal <SEZ (bit 10) in RSTS> 0: The EZ signal is OFF 1: The EZ signal is ON	[RSTS] (READ) 15 8 - - - - - n - -
Set the EZ input filter <EINF (bit 18) in RENV1> 1: Apply a filter to the EA, EB, EZ input. By applying a filter, input signal pulses shorter than 3 cycles of CLK are ignored.	[RENV1] (WRITE) 23 16 - - - - - n - -

11-5. Servomotor I/F

11-5-1. INP signal

The pulse strings input accepting servo driver systems have a deflection counter to count the difference between command pulse inputs and feedback pulse inputs. The driver controls to adjust the difference to zero. In other words, the effective function of servomotors is to delete command pulses and, even after the command pulses stop, the servomotor systems keep feeding until the count in the deflection counter reaches zero.

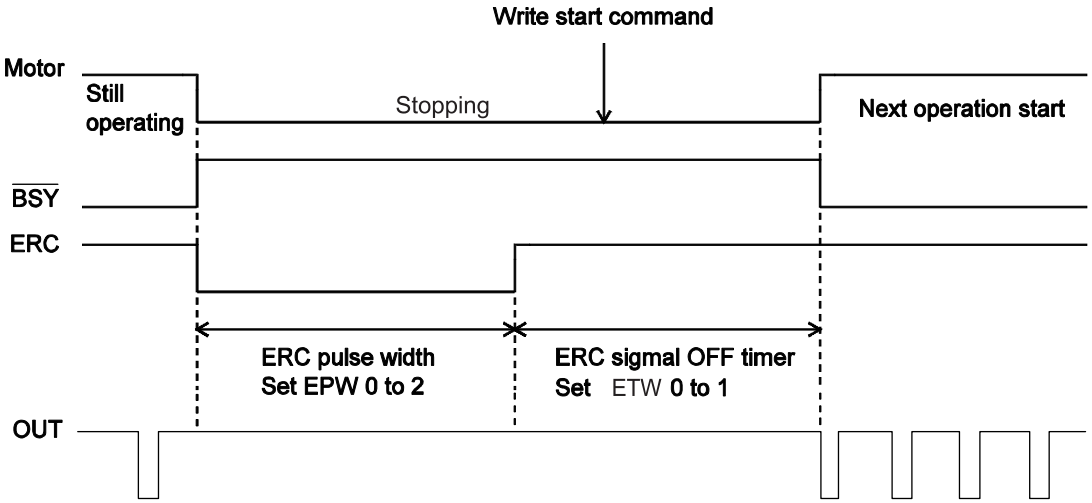
This LSI can receive a positioning complete signal (INP signal) from a servo driver in place of the pulse output complete timing, to determine when an operation is complete.
When the INP signal input is used to indicate the completion status of an operation, the $\overline{BSY}$ signal when an operation is complete, the main status (bits 0 to 5 of the MSTSW, stop condition), and the extension status (CND0 to 3, operation status) will also change when the INP signal is input.

The input logic of the INP signal can be changed.
The minimum pulse width of the INP signal is 2 cycles of the reference clock (0.1 μsec) when the input filter is OFF. If the input filter is ON, the PCL does not receive pulses shorter than the set length.
If the INP signal is already ON when the PCL is finished outputting pulses, it treats the operation as complete, without any delay.
The INP signal can be monitored by reading the RSTS register (extension status).

Set the operation complete delay using the INP signal <Set MINP (bit 9) in PRMD> 0: No operation complete delay waiting for the INP signal. 1: Operation complete (status, $\overline{BSY}$) delay until the INP signal turns ON.	[PRMD] (WRITE) 15 8 - - - - - - n -
Input logic of the INP signal <Set INPL (bit 22) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 23 16 - n - - - - - -
Reading the INP signal <SINP (bit 16) in RSTS> 0: The INP signal is OFF 1: The INP signal is ON	[RSTS] (READ) 23 16 0 0 0 0 0 0 0 n
Set the INP input filter <FLTR (bit 26) in RENV1> 1: Apply a filter to the ±EL, SD, ORG ALM and INP input. By applying a filter, pulses shorter than the FTM set value	[RENV1] (WRITE) 31 24 - - - - - n - -
Select the input filter characteristics <Set FTM (bits 20, 21) in RENV1> 00: 3.2 μsec 10: 200 μsec 01: 25 μsec 11: 1.6 msec	[RENV1] (WRITE) 23 16 - - n n - - - -

11-5-2. ERC signal

A servomotor delays the stop until the deflection counter in the driver reaches zero, even after command pulses have stopped being delivered. In order to stop the servomotor immediately, the deflection counter in the servo driver must be cleared.
This LSI can output a signal to clear the deflection counter in the servo driver. This signal is referred to as an "ERC signal." The ERC signal is output as one shot signal or a logic level signal. The output type can be selected by setting EPW in the RENV1 register (environment setting 1). If an interval is required for the servo driver to recover after turning OFF the ERC signal (HIGH) before it can receive new command pulses, the ETW signal OFF timer can be selected by setting ETW in the RENV1 register.



In order to output an ERC signal at the completion of a zero return operation, set EROR (bit 11) = 1 in the RENV1 register (environment setting 1) to make the ERC signal an automatic output. For details about ERC signal output timing, see the timing waveform in section 9-5-1, "Zero return operation."
In order to output an ERC signal for an immediate stop based on the EL signal, ALM signal, or $\overline{\text{CEMG}}$ signal input, or on the emergency stop command (05h), set EROE (bit 10) = 1 in the RENV1 register, and set automatic output for the ERC signal. (In the case of a deceleration stop, the ERC signal cannot be output, even when set for automatic output.)
The ERC signal can be output by writing an ERC output command (24h).
The output logic of the ERC signal can be changed by setting the RENV1 register. Read the RSTS (extension status) register to monitor the ERC signal.

Set automatic output for the ERC signal <Set EROE (bit 10) in RENV1> 1: Does not output an ERC signal when stopped by EL, ALM, or $\overline{\text{CEMG}}$ input. 1: Automatically outputs an ERC signal when stopped by EL, ALM, or $\overline{\text{CEMG}}$ input.	[RENV1] (WRITE) 15 8 - - - - - n - -
Set automatic output for the ERC signal <Set EROR (bit 11) in RENV1> 0: Does not output an ERC signal at the completion of a zero return operation. 1: Automatically outputs an ERC signal at the completion of a zero return operation.	[RENV1] (WRITE) 15 8 - - - - n - - -
Set the ERC signal output width <Set EPW0 to 2 (bits 12 to 14) in RENV1> 000: 12 μsec 100: 13 msec 001: 102 μsec 101: 52 msec 010: 408 μsec 110: 104 msec 011: 1.6 msec 111: Logic level output	[RENV1] (WRITE) 15 8 - n n n - - - -
Select output logic for the ERC signal <Set ERCL (bit 15) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 15 8 n - - - - - - -

Specify the ERC signal OFF timer time <Set ETW0 to 1 (bits 16 to 17) in RENV1> 00: 0 μsec 10: 1.6 msec 01: 12 μsec 11: 104 msec	[RENV1] (WRITE) 23 16 - - - - - - n n
Read the ERC signal <SERC (bit 9) in RSTS> 0: The ERC signal is OFF 1: The ERC signal is ON	[RSTS] (READ) 15 8 0 - - - - - n -
Emergency stop command <CMEMG: Bit control command> Output an ERC signal	[Bit control command] 05h
ERC signal output command <ERCOUT: Bit control command > Turn ON the ERC signal	[Bit control command] 24h
ERC signal output reset command <ERCRST: Bit control command > Turn OFF the ERC signal	[Bit control command] 25h

11-5-3. ALM signals

Input alarm (ALM) signal.
When the ALM signal turns ON while in operation, the axis will stop immediately or decelerate and stop.
To stop using deceleration, keep the ALM input ON until the axis stops operation.
However, the axis only decelerates and stops on an ALM signal if it was started with a high speed start.
If the ALM signal is ON when a start command is written, the LSI will not output any pulses.
The minimum pulse width of the ALM signal is 2 cycles of the reference clock (0.1 μs) if the input filter is OFF.
If the input filter is ON, the PCL does not receive pulses shorter than the specified length.
The input logic of the ALM signal can be changed. The signal status of the ALM signal can be monitored by reading SSTSW (sub status).

Stop method when the ALM signal is ON <Set ALMM (bit 8) in RENV1> 0: Stop immediately when the ALM signal is turned ON 1: Deceleration stop (high speed start only) when the ALM signal is turned ON	[RENV1] (WRITE) 15 8 - - - - - - - n
Input logic setting of the ALM signal <Set ALML (bit 9) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 15 8 - - - - - - n -
Read the ALM signal <SALM (bit 11) in SSTSW> 0: The ALM signal is OFF 1: The ALM signal is ON	[SSTSW] (READ) 15 8 - - - - n - - -
Reading the cause of a stop when the ALM signal is turned ON <ESAL (bit 7) in REST> 1: Stop due to the ALM signal being turned ON	[REST] (READ) 7 0 n - - - - - - -
Set the ALM input filter <Set FLTR (bit 26) in RENV1> 1: Apply a filter to the ±EL, SD, ORG ALM and INP input When a filter is applied, pulses shorter than the FTM set value will be ignored.	[RENV1] (WRITE) 31 24 - - - - n - -
Select the input filter characteristics <Set FTM (bits 20, 21) in RENV1> 00: 3.2 μsec 10: 200 μsec 01: 25 μsec 11: 1.6 msec	[RENV1] (WRITE) 23 16 - - n n - - - -

11-6. External start, simultaneous start

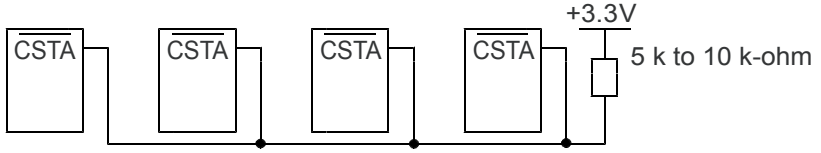
11-6-1. $\overline{\text{CSTA}}$ signal

This LSI can start when triggered by an external signal on the $\overline{\text{CSTA}}$ terminals. Set MSY (bits 18 and 19) = 01 in the PRDM register (operation mode) and the LSI will start feeding when the $\overline{\text{CSTA}}$ goes LOW. When you want to control multiple axes using more than one LSI, connect the $\overline{\text{CSTA}}$ terminal on each LSI and set the axes to "waiting for $\overline{\text{CSTA}}$ input", to start them all at the same time. In this example a start signal can be output through the $\overline{\text{CSTA}}$ terminal.
The input logic on the $\overline{\text{CSTA}}$ terminals cannot be changed.
By setting the RIRQ register (event interrupt cause), the $\overline{\text{INT}}$ signal can be output together with a simultaneous start (when the $\overline{\text{CSTA}}$ input is ON). By reading the RIST register, the cause of an event interrupt can be checked.
The operation status (waiting for $\overline{\text{CSTA}}$ input), and status of the $\overline{\text{CSTA}}$ terminal (OR of the $\overline{\text{CSTA}}$ signals) can be monitored by reading the RIST register, or RSTS register (extension status), respectively.

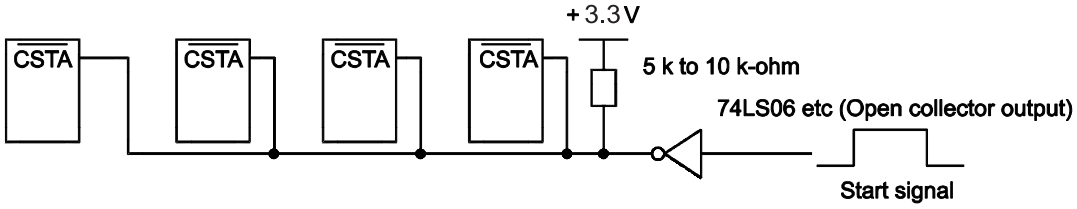
<How to make a simultaneous start>
Set MSY0 to 1 (bits 18 and 19) in the RMD register for the axes you want to start. Write a start command and put the LSI in the "waiting for $\overline{\text{CSTA}}$ input" status. Then, start the axes simultaneously by either of the methods described below.
1) By writing a simultaneous start command, the LSI will output a one shot signal of 8 reference clock cycles (approx. 0.4 µsec when CLK = 19.6608 MHz) from the $\overline{\text{CSTA}}$ terminal.
2) Input hardware signal from outside.
Supply a hardware signal by driving the terminal with open collector output (74LS06 or equivalent).

$\overline{\text{CSTA}}$ signals can be supplied as level trigger or edge trigger inputs. However, when level trigger input is selected, if $\overline{\text{CSTA}}$ = L or a start command is written, the axis will start immediately.
After connecting the $\overline{\text{CSTA}}$ terminals on each LSI, each axis can still be started independently using start commands.
To release the "waiting for $\overline{\text{CSTA}}$ input" condition, write an immediate stop command (49h).

1) To start axes controlled by different LSIs simultaneously, connect the LSIs as follows.



2) To start simultaneously from an external circuit, or use a single axis as an external start, connect the LSIs as follows.



For start signal, supply a one shot input signal with a pulse width of at least 4 reference clock cycles (approx. 0.2 µsec when CLK = 19.6608 MHz).

$\overline{\text{CSTA}}$ input <Set MSY0 to 1 (bits 18 and 19) in PRMD> 01: Start by inputting a $\overline{\text{CSTA}}$ signal	[PRMD] (WRITE) 23 16 - - - - n n - -
Specify the input specification for the $\overline{\text{CSTA}}$ signal <Set STAM (bit 18) in RENV1> 0: Level trigger input for the $\overline{\text{CSTA}}$ signal 1: Edge trigger input for the $\overline{\text{CSTA}}$ signal	[RENV1] (WRITE) 23 16 - - - - - n - -
Read the $\overline{\text{CSTA}}$ signal <SSTA (bit 5) in RSTS> 0: The $\overline{\text{CSTA}}$ signal is OFF 1: The $\overline{\text{CSTA}}$ signal is ON	[RSTS] (READ) 7 0 - - n - - - - -
Read the operation status <CND (bits 0 to 3) in RSTS> 0010: Waiting for $\overline{\text{CSTA}}$ input	[RSTS] (READ) 7 0 - - - - n n n n
Set an event interrupt cause <Set IRSA (bit 12) in RIRQ> 1: Output an $\overline{\text{INT}}$ signal when the $\overline{\text{CSTA}}$ input is ON.	[RIRQ] (WRITE) 15 8 0 0 0 n - - - -
Reading the event interrupt cause <ISSA (bit 13) in RIST> 1: When the $\overline{\text{CSTA}}$ signal is ON.	[RIST] (READ) 15 8 0 0 n - - - - -
Simultaneous start command <CMSTA: Operation command> Output a one shot pulse 8 reference clock cycles long from the $\overline{\text{CSTA}}$ terminal. (The $\overline{\text{CSTA}}$ terminal is bi-directional. It can receive signals output from other PCLs.)	[Operation command] 06h
Local axis only, simultaneous start command <SPSTA: Operation command> Used the same way as when a $\overline{\text{CSTA}}$ signal is supplied, for a local axis only.	[Operation command]] 2Ah

11-6-2. PCS signal

The PCS input is a terminal originally used for the target position override 2 function. By setting the PCSM (bit 30) to "1" in the RENV1 (environment 1) register, and the MSY (bits 18 and 19) to "01" in the PRMD (operation mode) register, the PCS input signal can enable the $\overline{\text{CSTA}}$ signal for only its own axis.

The input logic of the PCS input signal can be changed. The terminal status can be monitored by reading the RSTS register (extension status).

Specify the function of the PCS signal <Set PCSM (bit 30) in RENV1> 1: Make PCS input effective $\overline{\text{CSTA}}$ on only the local axis.	[RENV1] (WRITE) 31 24 - n - - - - - -
Set the Waiting for $\overline{\text{CSTA}}$ input <Set MSY0 to 1 (bits 18 and 19) in RMD> 01: Start on a $\overline{\text{CSTA}}$ input.	[RMD] (WRITE) 23 16 - - - - n n - -
Set the input logic of the PCS signal <Set PCSL (bit 24) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 31 24 - - - - - - - n
Read the PCS signal <SPCS (bit 8) in RSTS> 0: The PCS signal is OFF 1: The PCS signal is ON	[RSTS] (READ) 15 8 - - - - - - - n

11-7. External stop / simultaneous stop

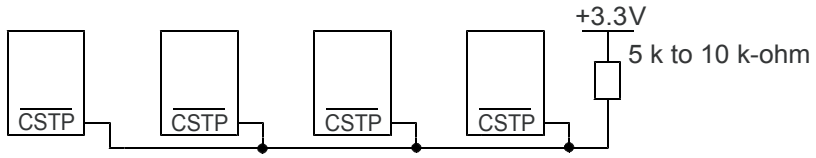
This LSI can execute an immediate stop or a deceleration stop triggered by an external signal using the $\overline{\text{CSTP}}$ terminal. Set MSPE (bit 24) = 1 in the PRMD register (operation mode) to enable a stop from a $\overline{\text{CSTP}}$ input. The axis will stop immediately or decelerate and stop when the $\overline{\text{CSTP}}$ terminal is LOW. However, a deceleration stop is only used for a high speed start. When the axis is started at low speed, the signal on the $\overline{\text{CSTP}}$ terminal will cause an immediate stop.
The input logic of the $\overline{\text{CSTP}}$ terminal cannot be changed.
When multiple LSIs are used to control multiple axes, connect all of the $\overline{\text{CSTP}}$ terminals from each LSI and input the same signal so that the axes which are set to stop on a $\overline{\text{CSTP}}$ input can be stopped simultaneously. In this case, a stop signal can also be output from the $\overline{\text{CSTP}}$ terminal.
When an axis stops because the $\overline{\text{CSTP}}$ signal is turned ON, an $\overline{\text{INT}}$ signal can be output. By reading the REST register, you can determine the cause of an error interrupt. You can monitor $\overline{\text{CSTP}}$ terminal status by reading the RSTS register (extension status).

<How to make a simultaneous stop>
Set MSPE (bit 24) = 1 in the PRMD register for each of the axes that you want to stop simultaneously. Then start these axes.
Stop these axes using either of the following two methods.

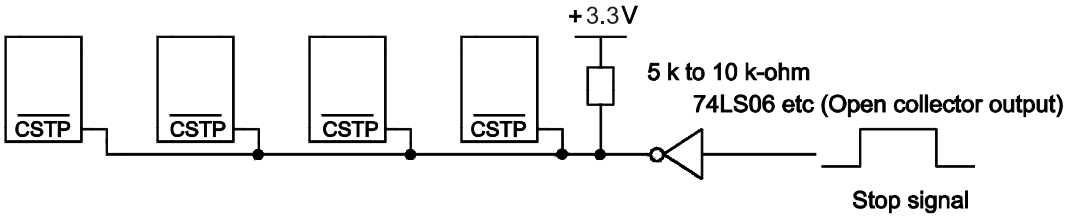
1) By writing a simultaneous stop command, the $\overline{\text{CSTP}}$ terminal will output a one shot signal 8 reference clock cycles in length (approx. 0.4 µsec when CLK = 19.6608 MHz).
2) Supply an external hardware signal
   Supply a hardware signal using an open collector output (74LS06 or equivalent).
3) The CSTP terminal will output a one shot signal for 8 reference clock cycles (approximately 0.4 µsec when CLK = 19.6608 MHz) when a stop caused by an error occurs on an axis that has MSPO = 1 in the PRMD register.

Even when the $\overline{\text{CSTP}}$ terminals on LSIs are connected together, each axis can still be stopped independently by using the stop command.

1) Connect the terminals as follows for a simultaneous stop among different LSIs.



2) To stop simultaneously using an external circuit, connect as follows.



As a stop signal, supply a one shot signal 4 reference clock cycles or more in length (approx. 0.2 µsec when CLK = 19.6608 MHz).

Setting to enable $\overline{\text{CSTP}}$ input <Set MSPE (bit 24) in PRMD> 1. Enable a stop from the $\overline{\text{CSTP}}$ input. (Immediate stop, deceleration stop)	[PRMD] (WRITE) 31 24 0 0 0 0 - - - n
Auto output setting for the $\overline{\text{CSTP}}$ signal <Set to MSPO (bit 25) in the PRMD> 1: When an axis stops because of an error, the PCL will output the $\overline{\text{CSTP}}$ signal. (Output signal width: 8 reference clock cycles)	[PRMD] (WRITE) 31 24 0 0 0 0 - - n -
Set the $\overline{\text{CSTP}}$ to output a signal when an axis is stopped by a command <Set CSP0 (bits 13) in RENV2> 1: When MSP0 = 1 in the PRMD register, the PCL will output the $\overline{\text{CSTP}}$ signal even if an axis is stopped by a command. 0: The PCL will not output a $\overline{\text{CSTP}}$ signal when an axis is stopped by a command.	[RENV2] (WRITE) 15 8 - - n - - - - -
Specify the stop method to use when the $\overline{\text{CSTP}}$ signal is turned ON. <Set STPM (bit 19) in RENV1> 0: Immediate stop when the $\overline{\text{CSTP}}$ signal is turned ON. 1: Deceleration stop when the $\overline{\text{CSTP}}$ signal is turned ON.	[RENV1] (WRITE) 23 16 - - - - n - - -
Read the $\overline{\text{CSTP}}$ signal <SSTP (bit 6) in RSTS> 0: The $\overline{\text{CSTP}}$ signal is OFF 1: The $\overline{\text{CSTP}}$ signal is ON	[RSTS] (READ) 7 0 - n - - - - - -
Read the cause of an error input < ESSP (bit 3) in REST> 1. When stopped because the $\overline{\text{CSTP}}$ signal turned ON.	[REST] (READ) 7 0 - - - - n - - -
Simultaneous stop command <CMSTP: Operation command> Outputs a one shot pulse of 8 reference clock cycles in length from the $\overline{\text{CSTP}}$ terminal. (The $\overline{\text{CSTP}}$ terminal is bi-directional. It can receive signals output from other PCLs.)	[Operation command] 07h

11-8. Emergency stop

This LSI has a $\overline{\text{CEMG}}$ input terminal for use as an emergency stop signal.
While in operation, if the $\overline{\text{CEMG}}$ input goes LOW or if you write an emergency stop command, all the axes will stop immediately. While the $\overline{\text{CEMG}}$ input remains LOW, no axis can be operated.
The logical input of the $\overline{\text{CEMG}}$ terminal cannot be changed.

When the axes are stopped because the $\overline{\text{CEMG}}$ input was turned ON, the LSI will output an $\overline{\text{INT}}$ signal. By reading the REST register, the cause of the error interruption can be determined.
The status of the $\overline{\text{CEMG}}$ terminal can be monitored by reading the REST register (extension status).

Read the $\overline{\text{CEMG}}$ signal <SEMG (bit 7) in RSTS> 0: The $\overline{\text{CEMG}}$ signal is OFF 1: The $\overline{\text{CEMG}}$ signal is ON	[RSTS] (READ) 7 0 n - - - - - - -
Read the cause of an error interrupt <ESEM (bit 4) in REST> 1. Stopped when the $\overline{\text{CEMG}}$ signal was turned ON.	[REST] (READ) 7 0 - - - n - - - -
Emergency stop command <CMEMG: Operation command> The operation is the same as when a $\overline{\text{CEMG}}$ signal is input.	[Operation command] 05h

Note: In a normal stop operation, the final pulse width is normal. However, in an emergency stop operation, the final pulse width may not be normal. It can be triangular. Motor drivers do not recognize triangle shaped pulses, and therefore only the PCL counter may count this pulse. (Deviation from the instructed position control). Therefore, after an emergency stop, you must perform a zero return to match the instructed position with the mechanical position.

11-9. Counter

11-9-1. Counter type and input method

In addition to the positioning counter, this LSI contains two other counters/axis.

The positioning counter is loaded with an absolute value for the RMV register (target position) with each start command, regardless of the operation mode selected. It decreases the value with each pulse that is output. However, if MPCS (bit 14) of the RMD register (operation mode) is set to 1 and a position override 2 is executed, the counter will not decrease until the PCS input turned ON.
Input to COUNTER1 and COUNTER2 can be selected as follows by setting the RENV3 register (environment setting 3).
* "0": Possible to count
Blank: Impossible to count

	COUNTER1	COUNTER2
Counter type	Up/down counter	Up/down counter
Number of bits	28	28
Output pulse	O	O
Encoder (EA/EB) input	O	O

Set COUNTER1 input <CIS1 (bit 0) in RENV3> 0: Output pulses 1: EA/EB input	[RENV3] (WRITE) 7 0 - - - - - - - n
Set COUNTER2 input <CIS2 (bit 1) in RENV3> 0: EA/EB input 1: Output pulses	[RENV3] (WRITE) 7 0 - - - - - - n -

The EA/EB input terminals, that are used as inputs for the counter, can be selected from the following two:
1) Signal input method: Input 90° phase difference signals (1x, 2x, 4x)
Counter direction: Count up when the EA input phase is leading. Count down when the EB input phase is leading.
2) Signal input method: Input 2 sets of positive and negative pulses.
Counter direction: Count up on the rising edge of the EA input. Count down on the falling edge of the EB input.

The counter direction or EA/EB input signals can be reversed.

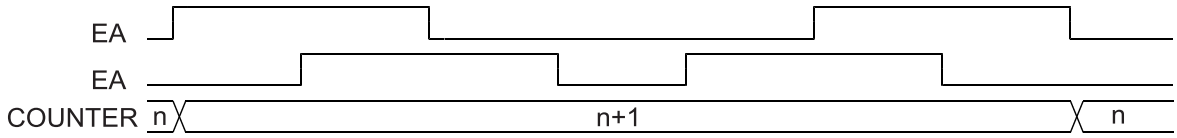
The LSI can be set to sense an error when both the EA and EB input, change simultaneously, and this error can be detected using the REST (error interrupt cause) register.

Set the input signal filter for EA/EB/EZ <Set EINF (bit 18) in RENV2> 0: Turn OFF the filter function 1: Turn ON the filter function (Input signals shorter than 3 reference clock cycles are ignored.)	[RENV2] (WRITE) 23 16 - - - - - n - -
Setting the EA/EB input <Set EIM0 to 1 (bit 16 and 17) in RENV2> 00: 90° phase difference, 1x 10: 90° phase difference, 4x 01: 90° phase difference, 2x 11: 2 sets of up or down input pulses	[RENV2] (WRITE) 23 16 - - - - - - n n
Specify the EA/EB input count direction <Set to EDIR (bit 19) in RENV2> 0: Count up when the EA phase is leading. Or, count up on the rising edge of EA. 1: Count up when the EB phase is leading. Or, count up on the rising edge of EB.	[RENV2] (WRITE) 23 16 - - - - n - - -
Enable/disable EA/EB input <Set EOFF (bit 14) in RENV2> 0: Enable EA/EB input 1: Disable EA/EB input. (EZ input is valid.)	[RENV2] (WRITE) 15 8 - n - - - - - -

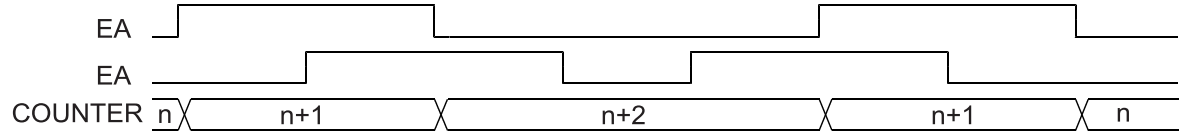
Reading EA/EB input error <ESEE (bit 7 in REST> 1: An EA/EB input error occurred.	[REST] (READ) 7 0 n - - - - - - -

When EDIR is "0," EA/EB input and count timing will be as follows.
For details about the PA/PB input, see section "9-3. Pulsar input mode."

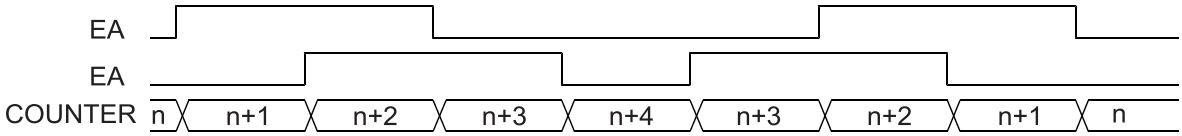
1) When using 90° phase difference signals and 1x input



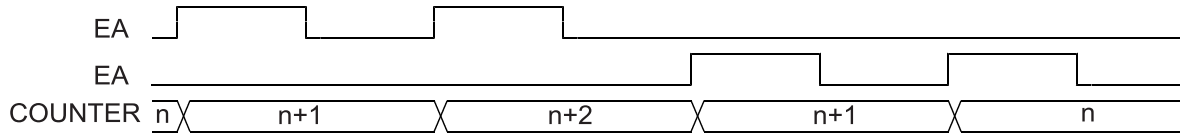
2) When using 90° phase difference signals and 2x input



3) When using 90° phase difference signals and 4x input



4) When two pulses are input (counted on the rising edge)



11-9-2. Counter reset

The following three methods allow all the counters to latch their count value using the RENV3 (environment setting 3) register. The latched values can read from the RLTC1/2 registers.

1) When the LTC signal turns ON.
2) When the ORG signal turns ON.
3) When a command is written.

The input timing of the LTC can be set in the RENV1 (environment setting 1) register. An $\overline{INT}$ signal can be output as an event interrupt factor when the PCL latches the count value by turning ON the LTC and ORG signals.

Write a command to reset the counters. There is no external input terminal to reset the counters. However, the PCL has a function that will clear a counter soon after the count value has been latched. An external latch signal can be input so that you can use the LTC input to reset a counter from the outside. The function used to reset a counter soon after the counter value is latched is referred to as the "latch & clear function."

The latch timing can be set in RENV3 (environment setting 3) register. The $\overline{INT}$ signal can be output to interrupt an event when it is latched by the LTC and ORG inputs.

Specify the LTC signal mode <Set LTCL (bit 23) in RENV1> 0: Latch on the falling edge. 1: Latch on the rising edge.	[RENV1] (WRITE) 23 16 n - - - - - - -
Read the LTC signal <SLTC (bit 13) in RSTS> 0: The LTC signal is OFF 1: The LTC signal is ON	[RSTS] (READ) 15 8 - - n - - - - -
Set the COUNTER1 latch & clear function <Set CU1L (bit 4) in RENV3> 0: COUNTER1 is not cleared after it is latched. 1: COUNTER1 is cleared soon after it is latched.	[RENV3] (WRITE) 7 0 - - - n - - - -
Set the COUNTER2 latch & clear function <Set CU2L (bit 8) in RENV3> 0: COUNTER2 is not cleared after it is latched. 1: COUNTER2 is cleared soon after it is latched.	[RENV3] (WRITE) 15 8 - - - - - - - n
Set COUNTER1 to latch on an external input <Set LOF1 (bit 5) in RENV3> 0: Latch COUNTER1 on an LTC input signal. 1: Do not latch COUNTER1.	[RENV3] (WRITE) 7 0 - - n - - - - -
Set COUNTER2 to latch on an external input <Set LOF2 (bit 9) in RENV3> 0: Latch COUNTER2 on an LTC input signal. 1: Do not latch COUNTER2.	[RENV3] (WRITE) 15 8 - - - - - - n -
Set COUNTER1 to latch on a zero return <Set CU1R (bit 6) in RENV3> 0: Do not latch COUNTER1 at the zero position. 1: Latch COUNTER1 at the zero position.	[RENV3] (WRITE) 7 0 - n - - - - - -
Set COUNTER2 to latch on a zero return <Set CU2R (bit 10) in RENV3> 0: Do not latch COUNTER2 at the zero position. 1: Latch COUNTER2 at the zero position.	[RENV3] (WRITE) 15 8 - - - - - n - -
Set an event interrupt cause <Set IRLT (bit 8) and IROL (bit 9) in RIRQ> IRLT = 1: Output an $\overline{INT}$ signal when the counter value is latched by the LTC signal being turned ON. IROL = 1: Output an $\overline{INT}$ signal when the counter value is latched by the ORG signal being turned ON.	[RIRQ] (WRITE) 15 8 - - - - - - n n
Read the event interrupt cause <ISLT (bit 8), ISOL (bit 9) in RIST> ISLT = 1: Latch the counter value when the LTC signal turns ON. ISOL = 1: Latch the counter value when the ORG signal turns ON.	[RIST] (READ) 15 8 - - - - - - n n
Counter latch command <LTCH: Bit control command> Latch the contents of the counters (COUNTER1 to 2).	[Bit control command] 29h

Counter reset command <CUN1R to CUN2R: Bit control command> 20h: Reset COUNTER1. 21h: Reset COUNTER2.	[Bit control command] 20h 21h

Note: When the latch & clear function is used, and if the clear (or latch) timing matches the count timing, the counter will not become 0. It will be +1 or -1.
When detecting "0" using the comparate function, be careful of these cases.

11-9-3. Stop the counter

There are two methods for stopping counters: stop the count operation or set a mask on the counter input.
The counter operation can be stopped for independently COUNTER1 and COUNTER2. Selection of the counter input is not related to stopping.
When the count input is masked, the input to the selected counter will be stopped.

A counter which is counting output pulses will stop counting if the timer mode is selected, regardless of the counter stop method selected or the setting status.
If a counter is counting output pulses and PMSK = 1 in the RENV1 register, the PCL will not output pulses. However, the counter will continue counting unless it is told to stop.

Stopping COUNTER1 <Set CU1H (bits 2) in RENV3> 1. Stop COUNTER1 counting operation.	[RENV3] (WRITE) 7 0 - - - - - n - -
Stopping COUNTER2 <Set CU2H (bits 3) in RENV3> 1. Stop COUNTER2 counting operation.	[RENV3] (WRITE) 7 0 - - - - n - - -
Set the count input mask for output pulses <Set MCCE (bit 11) in RMD> 1: The counters set to count "output pulses" will stop.	[RMD] (WRITE) 15 8 - - - - n - - -
Set the EA/EB signal input mask <Set E0FF (bit 14) in RENV2> 1: Disable the EA/EB input.	[RENV2] (WRITE) 15 8 - n - - - - - -

11-10. Comparator

11-10-1. Comparator types and functions

This LSI has 2 circuit 28-bit comparators per axis. These are referred to as "Comparator1" and " Comparator2."
Comparator 1 compares the setting in the RCMP1 register with COUNTER1.
Comparator 2 compares the setting in the RCMP2 register with COUNTER2.
One of three comparison methods can be selected (=, <, and >), and the comparison results can be output to a terminal. Also, the PCL can output an $\overline{INT}$ signal such as an event interrupt when comparison condition is met.
A special use of the comparator is to control a ring count function and internal synchronized start function. For descriptions of these functions, see "11-10-2. Ring count function" and "11-11-2. Start on an internal synchronized signal."

Use the RENV2 and RENV3 registers to set the comparators.

Set the comparison conditions for Comparator 1 <Set C1S1, 0 (bits 12, 13) in RENV3> 00: Turn OFF the comparator function 01: (RCMP1) = (COUNTER1) 10: (RCMP1) > (COUNTER1) 11: (RCMP1) < (COUNTER1)	[RENV3] (WRITE) 15 8 - - n n - - - -
Set the comparison conditions for Comparator 2 <Set C2S1, 0 (bits 14, 15) in RENV3> 00: Turn OFF the comparator function 01: (RCMP1) = (COUNTER1) 10: (RCMP1) > (COUNTER1) 11: (RCMP1) < (COUNTER1)	[RENV3] (WRITE) 15 8 n n - - - - - -
Set an event interrupt factor <Set IRC1, 2 (bits 6, 7) in RIRQ> IRC1 (bit 6) = 1: Outputs an $\overline{INT}$ signal when Comparator 1 conditions are met. IRC2 (bit 7) = 1: Outputs an $\overline{INT}$ signal when Comparator 2 conditions are met.	[RIRQ] (WRITE) 7 0 n n - - - - - -
Read the event interrupt factor <Set ISC1, 2 (bits 6, 7) in RIST> IRC1 (bit 6) = 1: When the Comparator 1 conditions are met. IRC2 (bit 7) = 1: When the Comparator 2 conditions are met.	[RIST] (READ) 7 0 n n - - - - - -
Read the comparator condition status <Set SCP1, 2 (bits 8, 9) in MSTSW> SCP1 (bit 8) = 1: When the Comparator 1 conditions are met. SCP2 (bit 9) = 1: When the Comparator 2 conditions are met.	[MSTSW] (READ) 15 8 - - - - - - n n
Set the specifications for the P3/CP1 terminal <Set P3M0 to 1 (bits 6 to 7) in RENV2> 00: General-purpose input 01: General-purpose output 10: Output a CP1 signal (when the Comparator1 conditions are met) using negative logic. 11: Output a CP1 signal (when the Comparator1 conditions are met) using positive logic.	[RENV2] (WRITE) 7 0 n n - - - - - -
Set the specifications for the P4/CP2 terminal <Set P4M0 to 1 (bits 8 to 9) in RENV2> 00: General-purpose input 01: General-purpose output 10: Output a CP2 signal (when the Comparator2 conditions are met) using negative logic. 11: Output a CP2 signal (when the Comparator2 conditions are met) using positive logic.	[RENV2] (WRITE) 15 8 - - - - - - n n

11-10-2. Ring count function

COUNTER1 and COUNTER2 have a ring count function for use in controlling a rotating table.

Set C1RM = 1 in RENV3 and COUNTER1 will be in the ring count mode. Then the PCL can perform the following operations.
- Count value = Count up from the value in RCMP1 until reaching 0.
- Count value = Count down from 0 until the count equals the value in PCMP1.

Set C2RM = 1 in RENV3 and COUNTER2 will be in the ring count mode. Then the PCL can perform the following operations.
- Count value = Count up from the value in RCMP2 until reaching 0.
- Count value = Count down from 0 until the count equals the value in RCMP2.

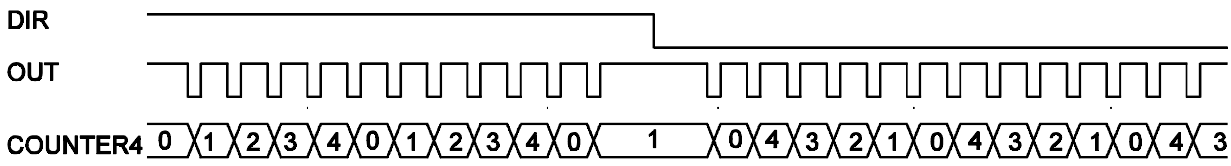
Set COUNTER1 to ring counter operation <set C1RM (bit-7) in RENV3> 1: Operate COUNTER1 as a ring counter.	[RENV2] (WRITE) 7 0 n - - - - - - -
Set COUNTER2 to ring count operation <set C2RM (bit-11) in RENV3> 1: Operate COUNTER2 as a ring counter.	[RENV2] (WRITE) 15 8 - - - - n - - -

Even if the value for PRMV outside the range of 0 to the value in RCMPn, the PCL will continue to perform positioning operations.
When driving a rotating table with 3600 pulses per revolution, and when RCMP1 = 3599, MOD = 41h, and RMV = 7200, the table will rotate twice and the value in COUNTER1, when stopped, will be the same as the value before starting.

Note: To use the ring counter function, set the count value between 0 and the value in RCMPn. If the value is outside the range above, the PCL will not operate normally.
Set the comparator conditions (C1S0 to 1, C2S0 to 1) when using a counter as a ring counter to "00."

Setting example

RENV3 = XXXXXX80h --- COUNTER1 is in ring counter mode (C1RM = 1)
RCMP1 = 4 --- Count range: 0 to 4



11-11. Synchronous starting

This LSI can perform the following operation by setting the PRMD (operation mode) register in advance.

- ♦ Start triggered by another axis stopping.
- ♦ Start triggered by an internal synchronous signal from another axis.

The internal synchronous signal output is available with 6 types of timing. They can be selected by setting the RENV3 (environment setting 3) register. By setting the RIRQ (event interrupt cause) register, an $\overline{INT}$ signal can be output at the same time the internal synchronous signal is output. You can determine the cause of event interrupt by reading the RIST register. The operation status can be checked by reading the RSTS (extension status) register.

11-11-1. Start triggered by another axis stopping

If the start condition is specified as a "Stop on two or more axes," when any of the specified axes stops after operating, and the other axes never start (remain stopped), the axis which is supposed to start when the conditions are met will start operation.
Example 1 below shows how to specify a "stop on two or more axes." In the example, while the X axis (or Y axis) is working, the Y (or X) axis remains stopped. Then, the U axis starts operation when triggered by the X (or Y) axis stopping.

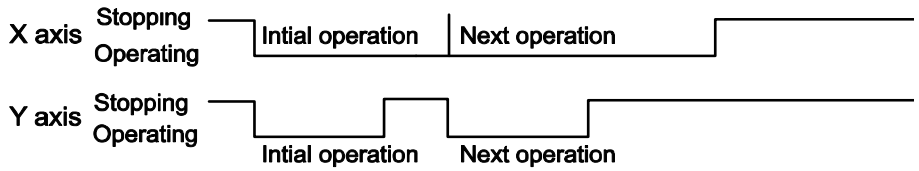
Specify the synchronous starting method <Set MSY0 to 1 (bits 18 & 19) in PRMD> 11: Start triggered by specified axis stopping.	[PRMD] (WRITE) 23 16 - - - - n n - -
Select an axis for confirming a stop (setting example) <Specify the axis using MAX0 to Max3 (bits 20 to 23) in PRMD> 0001: Start when the X axis stops 0010: Start when the Y axis stops 0100: Start when the Z axis stops 1000: Start when the U axis stops 0011: Start when both the X and Y axes have stopped 0101: Start when both the X and Z axes have stopped 1011: Start when the X, Y, and U axes have all stopped 1111: Start when all of the axes have stopped	[PRMD] (WRITE) 23 16 n n n n - - - -
Read the operation status <CND (bits 0 to 3) in RSTS> 0100: Wait for another axis to stop.	[RSTS] (READ) 7 0 - - - - n n n n

[Example 1]
After setting steps 1) to 3), start both the X and Y axes. When both the axes stop, the U axis will start.

1) Set MSY0 to 1 (bits 18 to 19) in PRMD for the U axis to "11." (Start triggered by another axis stopping)
2) Set MAX0 to 3 (bits 20 to 23) in PRMD for the U axis to "0011." (When the Y axis and then the X axis stops)
3) Write a start command for the U axis.

Operation examples
Settings

Operation mode for the X axis in initial operation: MSY0 to 1 = 00, MAX0 to 3 = 0000
Operation mode calling for the X axis in the next operation: MSY0 to 1 = 11, MAX0 to 3 = 0011
Operation mode for the Y axis in initial operation: MSY0 to 1 = 00, MAX0 to 3 = 0000
Operation mode calling for the Y axis in the next operation: MSY0 to 1 = 11, MAX0 to 3 = 0011
(X axis positioning operation time) > (Y axis positioning operation time)



When using continuous interpolation without changing the interpolation axes, you may set the next operation in the pre-register (you don't need to specify any stop conditions) rather than using the "start when another axis stops" function.
When operating the continuous interpolation with changing the interpolation axes, by using the pre-register function, you have to be careful. In this case, put a 0 in the PRMV of the axes that will not move (not be interpolated) and operate them as dummy interpolated axes.

How to perform continuous interpolation while changing the interpolated axis during the interpolation operation (Linear interpolation between the X and Y axes => Linear interpolation between the X and Z axes).

Step	Register	X axis	Y axis	Z axis	Description
1	PRMV	10000	5000	0	Linear interpolation of X: 10000, Y: 5000. The Z axis performs a dummy interpolation operation with zero feed amounts.
	PRIP	10000	10000	10000	
	PRMD	0000_0063h	0000_0063h	0000_0063h	
	Start command: Write 0751h (FH low speed start)				X and Y axes start command
2	PRMV	10000	0	-20000	Linear interpolation of X: 10000, Z: -20000 The Y axis performs a dummy interpolation operation with zero feed amounts. When the X, Y, and Z axes stop feeding, restart the X, Y, and Z axis.
	PRIP	20000	20000	20000	
	PRMD	007C_0063h	007C_0063h	007C_0063h	
	Start command: Write 0751h (FH speed start)				X, Y, and Z axes start command (SPRF will be 1).

11-11-2. Start on an internal synchronous signal

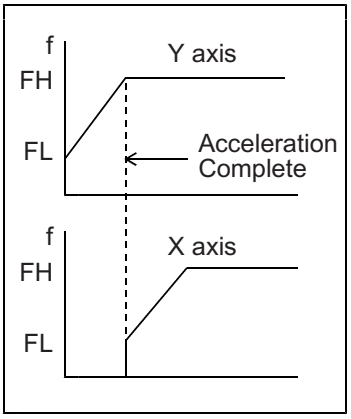
This is a function that allows a start by the same axis that is being controlled when another axis achieves a specified status.
Each axis selects the internal synchronous signal (status signal) from its own axis and outputs it to the other axes.
Select an axis whose internal synchronizing signal will be used to trigger itself to start.
The internal synchronization signal output has 6 possible timings. Select the timing with the RENV3 register.

Setting the synchronous start method <Set MSY0 to 1 (bits 18 to 19) in PRMD> 10: Start by the internal synchronize signal.	[PRMD] (WRITE) 23 16 - - - - n n - -
Setting the internal synchronous signal output timing <Set SY01 to 3 (bits 16 to 19) in RENV3> 0001: When the Comparator1 conditions are met. 0010: When the Comparator2 conditions are met. 1000: When you want to start acceleration. 1001: When you want to finish the acceleration phase. 1010: When you want to start deceleration. 1011: When you want to finish the deceleration phase. Others: Turn OFF the internally synchronized output.	[RENV3] (WRITE) 23 16 - - - - n n n n
Select the internally synchronized signal input <SYI0 to 1 (bit 20 to 21) in RENV3> 00: Use the internal synchronous signal output by the X axis. 01: Use the internal synchronous signal output by the Y axis. 10: Use the internal synchronous signal output by the Z axis. 11: Use the internal synchronous signal output by the U axis.	[RENV3] (WRITE) 23 16 - - n n - - - -
Reading the operation status <CND (bits 0 to 3) in RSTS> 0011: Waiting for an internal synchronous signal	[RSTS] (WRITE) 7 0 - - - - n n n n

Example 1 below shows a case of using the internal synchronous signal.

[Setting example 1]
After completing steps 1) to 3) below, write a start command to the X and Y axes, the X axis will start when the Y axis completes its acceleration.

1) Set MSY0 to 1 (bits 18 &19) in the X axis PRMD to 10. (Start with an internal synchronous signal)
2) Set SYI0 to 1 (bits 20 & 21) in the X axis RENV3 to 01. (Use an internal synchronous signal from the Y axis.)
3) Set SYO0 to 3 (bits 16 to 19) in the Y axis RENV3 to 1001. (Output an internal synchronous signal when the acceleration is complete)



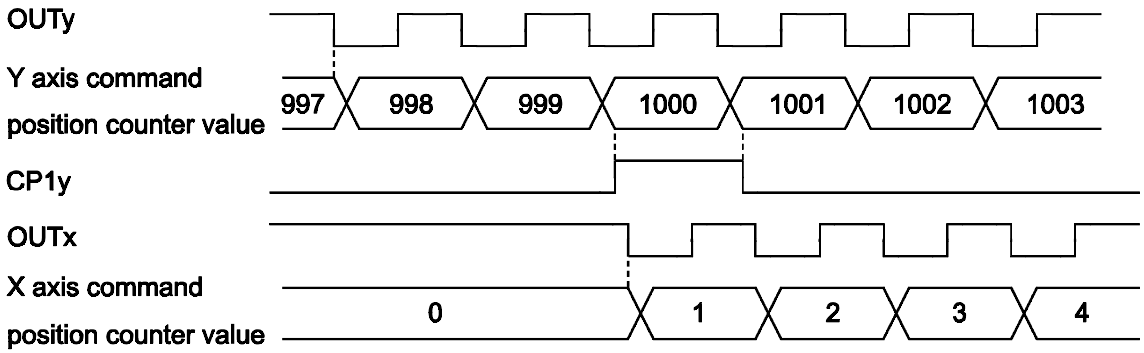
Example 2 shows how to start another axis using the satisfaction of the comparator conditions to generate an internal synchronous signal.
Be careful, since comparator conditions satisfied by timing and the timing of the start of another axis may be different according to the comparison method used by the comparators.

[Example 2]
Use COUNTER1 and Comparator 1 to start the X axis when the Y axis = 1000.

1) Set MSY0 to 1 (bits 18 & 19) in the Y axis PRMD to 10. (Start from an internal synchronous signal)
2) Set SYI0 to 1 (bits 20 & 21) in the X axis RENV3 to 01. (Use an internal synchronous signal from the Y axis)
3) Set SYO0 to 3 (bits 16 to 19) in the Y axis RENV3 to 0001. (Output an internal synchronous signal when the Comparator 1 conditions are satisfied)
4) Set C1S0 to 2 (bits 2 to 4) in the Y axis RENV3 to 001. (Comparison method: Comparator 1 = Comparison counter)

5) Set C1D0 to 1 (bits 5 & 6) in the Y axis RENV3 to 00. (Do nothing when the Comparator 1 condition are satisfied)
6) Set the RCMP1 value of the Y axis to 1000. (Comparison counter value of Comparator 1 is 1000.)
7) Write start commands for the X and Y axes.

The timing chart below shows the period after the Comparator 1 conditions are established and the X axis starts.



Note: In the example above, even if the Y feed amount is set to 2000 and the X feed amount is set to 1000, the X axis will be 1 when the Y axis position equals 1000. Therefore, the operation complete position will be one pulse off for both the X and Y axes. In order to make the operation complete timing the same, set the RCMP1 value to 1001 or set the comparison conditions to "Comparator 1 < comparison counter."

11-12. Output an interrupt signal

This LSI can output an interrupt signal ($\overline{INT}$ signal) : There are 9 types of errors, 14 types of events, and change from operating to stop that can cause an $\overline{INT}$ signal to be output . All of the error causes will always output an $\overline{INT}$ signal. Each of the event causes can be set in the RIRQ register to output an $\overline{INT}$ signal or not.

A stop interrupt is a simple interrupt function which produces an interrupt separate from a normal stop or error stop.
For a normal stop interrupt to be issued, the confirmation process reads the RIST register as described in the Cause of an Event section. If your system needs to provide a stop interrupt whenever a stop occurs, it is easy to use the stop interrupt function.

The $\overline{INT}$signal is output continuously until all the causes on all the axes that produced interrupts have been cleared. An interrupt caused by an error is cleared by writing a "REST (error cause) register read command." An interrupt caused by an event is cleared by writing a "RIST (event cause) register read command." A Stop interrupt is cleared by writing to the main status.

To determine which type of interrupt occurred, on which axis and the cause of the interrupt, follow the procedures below.
1) Read the main status of the X axis and check whether bits 2, 4, or 5 is "1."
2) If bit 2 (SENI) is "1," a Stop interrupt occurs.
3) If bit 4 (SERR) is "1," read the RESET register to identify the cause of the interrupt.
4) If bit 5 (SINT) is "1," read the RIST register to identify the cause of the interrupt.
5) Repeat steps 1) to 4) above for the Y, Z, and U axes.
The steps above will allow you to evaluate the cause of the interrupt and turn the $\overline{INT}$ output OFF.

Note 1: When reading a register from the interrupt routine, the details of the input/output buffer will change. If the $\overline{INT}$ signal is output while the main routine is reading or writing registers, and the interrupt routine starts, the main routine may produce an error. Therefore, the interrupt routine should execute a PUSH/POP on input/output buffer.

Note 2: While processing all axes in steps 1) to 4) above, it is possible that another interrupt may occur on an axis whose process has completed. In this case, if the CPU interrupts reception mode, and is set for edge triggering, the PCL will latch the $\overline{INT}$ output ON and it will not allow a new interrupt to interfere. Therefore, make sure that after you have reset the interrupt reception status the CPU reads main status of all the axes again. Also, make sure there is no $\overline{INT}$ signal output from the PCL. Then, end the interrupt routine.

Note 3: When not using the $\overline{INT}$ terminal, leave it open.
When using more than one PCL, the $\overline{INT}$ terminals cannot be wired ORed.

The $\overline{INT}$ signal output can be masked by setting the RENV1 (environment setting 1) register.
If the $\overline{INT}$ output is masked (INTM = 1), and when the interrupt conditions are satisfied, the status will change. However, the $\overline{INT}$ signal will not go LOW, but will remain HIGH.
While the interrupt conditions are satisfied and if the output mask is turned OFF (INTM = 0 in RENV1), the $\overline{INT}$ signal will go LOW.

Read the interrupt status <SENI(bit2), SERR (bit 4), SINT (bit 5) in MSTSW> SENI = 1: When IEND = 1 and a stop interrupt occurs, make this bit 1. After reading MSTS, it will become 0. SERR = 1: Becomes 1 when an error interrupt occurs. Becomes 0 by reading REST. SINT = 1: Becomes 1 when an event interrupt occurs. Becomes 0 by reading RIST.	[MSTSW] (READ) 7 0 - - n n - n - -
Set the interrupt mask <INTM (bit 29) in RENV1> 1: Mask $\overline{INT}$ output.	[RENV1] (WRITE) 31 24 - - n - - - - -
Setting a stop interrupt <IEDN (bit 30) in RENV2> 1: Enable a stop interrupt.	[RENV2] (WRITE) 31 24 0 n - - - - - -
Read the cause of the error interrupt <RREST: Read out command> Copy the data in the REST register (error interrupt cause) to BUF.	[Read command] F2h
Read the event interrupt cause <RRIST: Read out command> Copy the data in the RIST register (event interrupt cause) to BUF.	[Read command] F3h
Set the event interrupt cause <WRIRQ: Write command> Write the BUF data to the RIRQ register (event interrupt cause).	[Write command] ACh

[Error interrupt causes] <Detail of REST: The cause of an interrupt makes the corresponding bit "1">

Error interrupt cause	Cause (REST)	
	Bit	Bit name
Stopped by turning ON the +EL input	0	ESPL
Stopped by turning ON the -EL input	1	ESML
Stopped by turning ON the ALM input	2	ESAL
Stopped by turning ON the $\overline{CSTP}$ input	3	ESSP
Stopped by turning ON the $\overline{CEMG}$ input	4	ESEM
Deceleration stopped by turning ON the SD input	5	ESSD
Stopped by an overflow of PA/PB input buffer counter occurrence	6	ESPO
An EA/EB input error occurred (does not stop).	7	ESEE
A PA/PB input error occurred (does not stop).	8	ESPE

[Event interrupt causes] < The corresponding interrupt bit is set to 1 and then an interrupt occurred>

Event interrupt cause	Set cause (RIRQ)		Cause (RIST)	
	Bit	Bit name	Bit	Bit name
Automatic stop	0	IREN	0	ISEN
When enabled to write to the pre-register.	1	IRNM	1	ISNM
When acceleration starts	2	IRUS	2	ISUS
When acceleration ends	3	IRUE	3	ISUE
When deceleration starts	4	IRDS	4	ISDS
When deceleration ends	5	IRDE	5	ISDE
When the Comparator 1 conditions are satisfied	6	IRC1	6	ISC1
When the Comparator 2 conditions are satisfied	7	IRC2	7	ISC2
When the counter value is latched by an LTC input	8	IRLT	8	ISLT
When the counter value is latched by an ORG input	9	IROL	9	ISOL
When the SD input is turned ON	10	IRSD	10	ISSD
When the +DR input changes	11	IRDR	11	ISPD
When the -DR input changes			12	ISMD
When the $\overline{CSTA}$ input is turned ON	12	IRSA	13	ISSA

12. Electrical Characteristics

12-1. Absolute maximum ratings

Item	Symbol	Rating	Unit	Remark
Power supply voltage	V_{DD}	-0.3 to + 4.0	V	
Input voltage	V_{IN}	-0.3 to + 7.0	V	
Output current	I_{OUT}	±30	mA	
Storage temperature	Tstg	-65 to +150	°C	

12-2. Recommended operating conditions

Item	Symbol	Rating	Unit	Remark
Power supply voltage	V_{DD}	3.0 to 3.6	V	
Ambient temperature	T J	-40 to +85	°C	No dewing

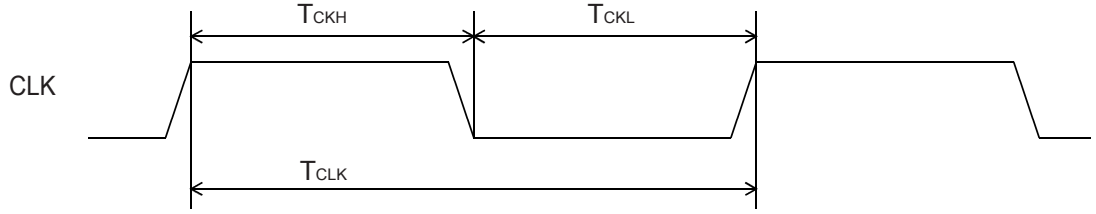
12-3. DC characteristics

Item	Symbol	Condition	Min.	Max.	Unit
Consumption current (PCL6113)	I_{dd1}	CLK = 30 MHz, 1 axis at 15 Mpps, no load		36	mA
Consumption current (PCL6123)	I_{dd2}	CLK = 30 MHz, 2 axes at 15 Mpps, no load		77	mA
Consumption current (PCL6143)	I_{dd4}	CLK = 30 MHz, 4 axes at 15 Mpps, no load		180	mA
Input current leakage	I_{LI}			10	µA
LOW input current (V_{IL} = GND)	I_{IL}	$\overline{CS}$, $\overline{RD}$, $\overline{WR}$, A0 to A4, D0 to D15, CLK	-1		µA
		Input terminals other than the above (Note 1)	-125		
HIGH input current	I_{IH}	$V_{IH} = V_{DD}$		1	µA
		V_{IH} = 5.5 V		30	
LOW input voltage	V_{IL}		-0.3	0.8	V
HIGH input voltage	V_{IH}		2.0	7.0	V
LOW output voltage	V_{OL}	I_{OL} = 6 mA		0.4	V
HIGH output voltage	V_{OH}	I_{OH} = -6 mA	V_{DD} -0.4		V
LOW output current	I_{OL}	V_{OL} = 0.4 V		6	mA
HIGH output current	I_{OH}	$V_{OH} = V_{DD}$ -0.4 V	-6		mA
Internal pull up resistance	R_{PD}	Other than $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, A0 to A4, D0 to D15, CLK	40	240	K-ohm

Note 1: Internal pull down resistors are integrated for safety when open.
Note 2: The signs next to the current values shown (in amperes) refer to current flowing in (a positive value) or out (a negative value).

12-4. AC characteristics 1) (reference clock)

Item	Symbol	Condition	Min.	Max.	Unit
Reference clock frequency	f_{CLK}			30	MHz
Reference clock cycle	T_{CLK}		33		ns
Reference clock HIGH width	T_{CKH}		16		ns
Reference clock LOW width	T_{CKL}		16		ns

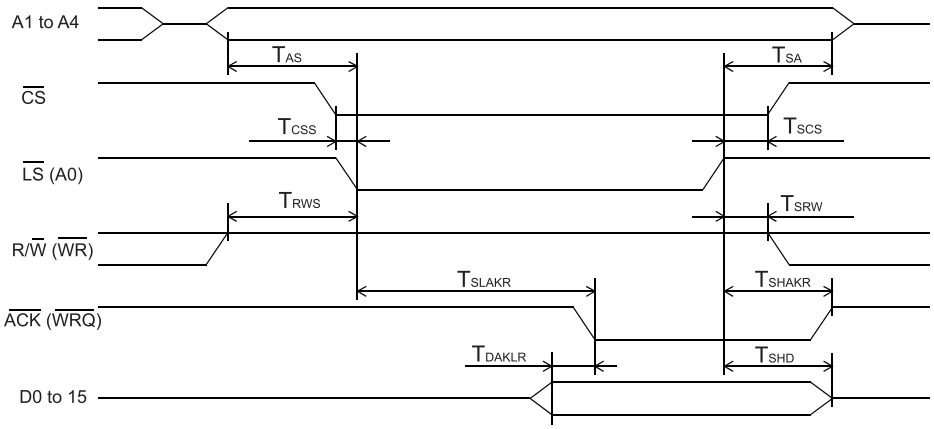


12-5. AC characteristics 2) (CPU I/F)

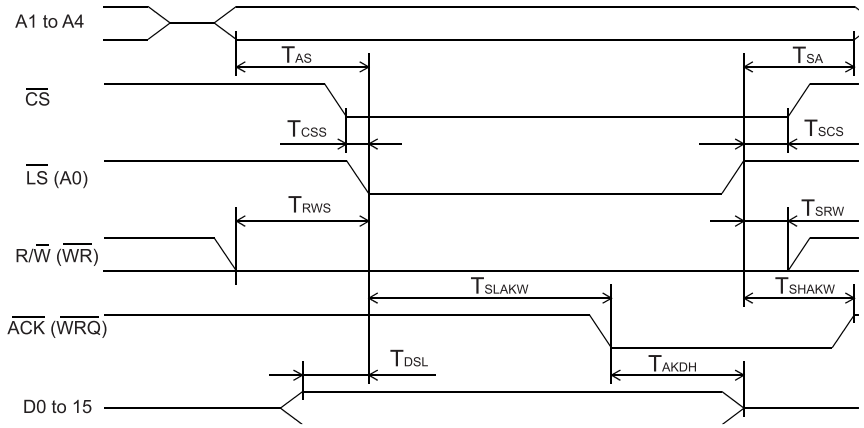
12-5-1. 16-bits I/F-1) (IF1 = L, IF0 = L) 68000

Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{LS}$ ↓	T_{AS}		13		ns
Address hold time for $\overline{LS}$ ↑	T_{SA}		0		ns
CS setup time for $\overline{LS}$↓	T_{CSS}		2		ns
CS hold time for $\overline{LS}$ ↑	T_{SCS}		0		ns
R/$\overline{W}$ setup time for $\overline{LS}$↓	T_{RWS}		1		ns
R/$\overline{W}$ hold time for $\overline{LS}$ ↑	T_{SRW}		1		ns
$\overline{ACK}$ ON delay time for $\overline{LS}$ ↓	T_{SLAKR}	C_L = 40pF	T_{CLK}	$4T_{CLK}$ +15	ns
	T_{SLAKW}	C_L = 40pF	T_{CLK}	$4T_{CLK}$ +15	ns
$\overline{ACK}$ OFF delay time for $\overline{LS}$ ↑	T_{SHAKR}	C_L = 40pF		17	ns
	T_{SHAKW}	C_L = 40pF		17	ns
Data output delay time for $\overline{ACK}$ ↓	T_{DAKLR}	C_L = 40pF	T_{CLK}		ns
Data float delay time for $\overline{LS}$ ↑	T_{SHD}	C_L = 40pF		18	ns
Data setup time for $\overline{LS}$ ↑	T_{DSL}		17		ns
Data hold time for $\overline{ACK}$ ↓	T_{AKDH}		0		ns

<Read cycle>



<Write cycle>

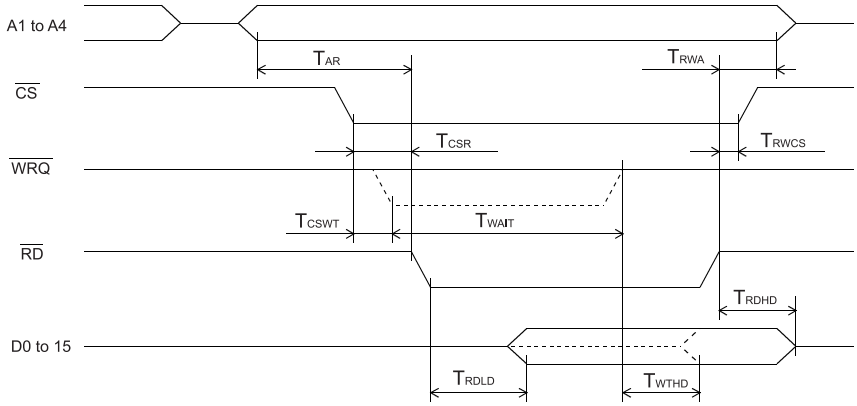


12-5-2. 16-bits I/F-2 (IF1=L, IF0=H) H8

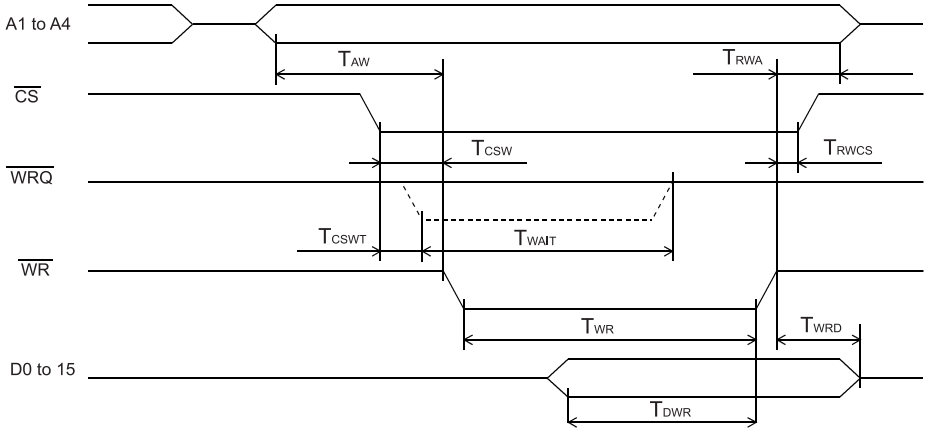
Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{RD}$ ↓	T_{AR}		10		ns
Address setup time for $\overline{WR}$ ↓	T_{AW}		10		ns
Address hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWA}		0		ns
$\overline{CS}$ setup time for $\overline{RD}$↓	T_{CSR}		4		ns
$\overline{CS}$ setup time for $\overline{WR}$ ↓	T_{CSW}		4		ns
$\overline{CS}$ hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWCS}		0		ns
$\overline{WRQ}$ ON delay time for $\overline{CS}$ ↓	T_{CSWT}	C_L = 40pF		13	ns
$\overline{WRQ}$ signal LOW time	T_{WAIT}			$4T_{CLK}$	ns
Data output delay time for $\overline{RD}$ ↓	T_{RDLD}	C_L = 40pF		21	ns
Data output delay time for $\overline{WRQ}$ ↑	T_{WTHD}	C_L = 40pF		10	ns
Data float delay time for $\overline{RD}$ ↑	T_{RDHD}	C_L = 40pF		18	ns
$\overline{WR}$ signal width	T_{WR}	Note 1	11		ns
Data setup time for $\overline{WR}$ ↑	T_{DWR}		12		ns
Data hold time for $\overline{WR}$ ↑	T_{WRD}		0		ns

Note 1: When a $\overline{WRQ}$ signal is output, the duration will be the interval between $\overline{WRQ}$ = H and $\overline{WR}$ = H.

<Read cycle>



<Write cycle>

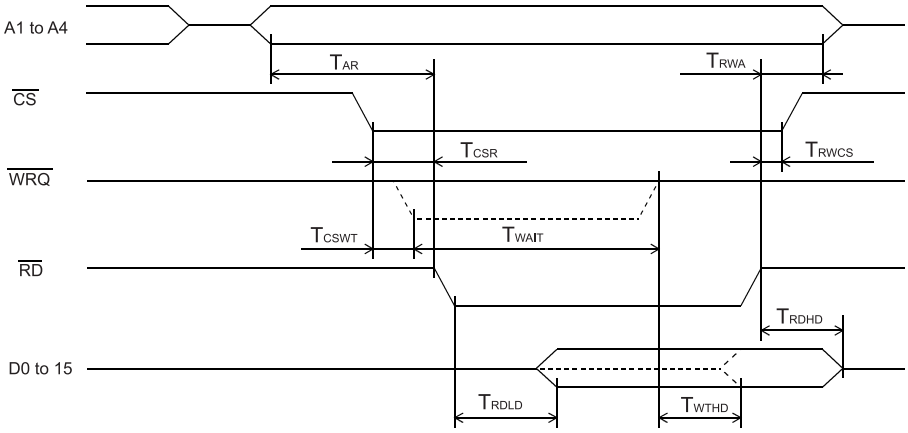


12-5-3. 16-bits I/F-3 (IF1=H, IF0=L) 8086

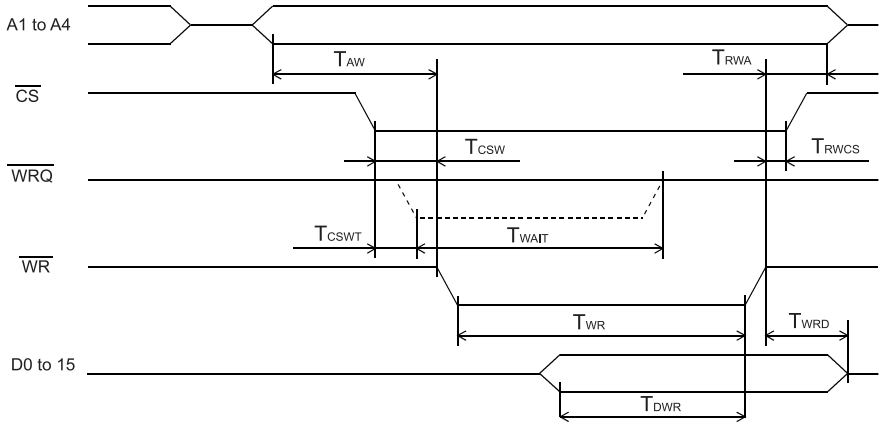
Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{RD}$ ↓	T_{AR}		10		ns
Address setup time for $\overline{WR}$ ↓	T_{AW}		10		ns
Address hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWA}		0		ns
$\overline{CS}$ setup time for $\overline{RD}$↓	T_{CSR}		4		ns
$\overline{CS}$ setup time for $\overline{WR}$ ↓	T_{CSW}		4		ns
$\overline{CS}$ hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWCS}		0		ns
$\overline{WRQ}$ ON delay time for $\overline{CS}$ ↓	T_{CSWT}	C_L = 40pF		13	ns
$\overline{WRQ}$ signal LOW time	T_{WAIT}			$4T_{CLK}$	ns
Data output delay time for $\overline{RD}$ ↓	T_{RDLD}	C_L = 40pF		21	ns
Data output delay time for $\overline{WRQ}$ ↑	T_{WTHD}	C_L = 40pF		10	ns
Data float delay time for $\overline{RD}$ ↑	T_{RDHD}	C_L = 40pF		18	ns
$\overline{WR}$ signal width	T_{WR}	Note 1	11		ns
Data setup time for $\overline{WR}$ ↓	T_{DWR}		12		ns
Data hold time for $\overline{WR}$ ↑	T_{WRD}		0		ns

Note 1: When a $\overline{WRQ}$ signal is output, the duration will be the interval between $\overline{WRQ}$ = H and $\overline{WR}$ = H.

<Read cycle>



<Write cycle>

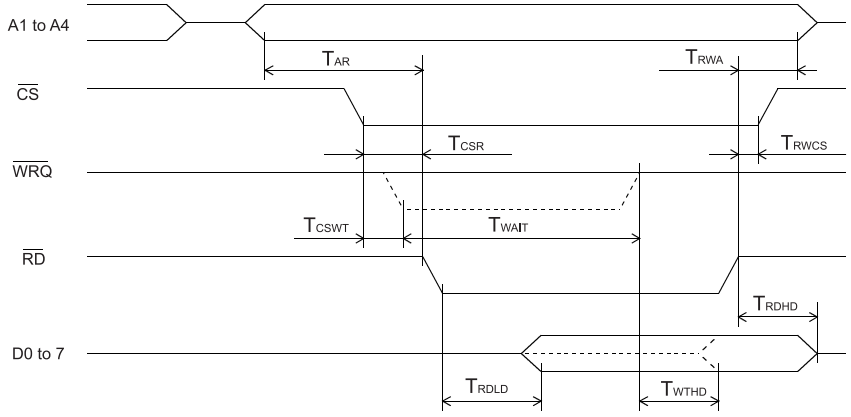


12-5-4. 8-bits I/F-2 (IF1=H, IF0=H) Z80

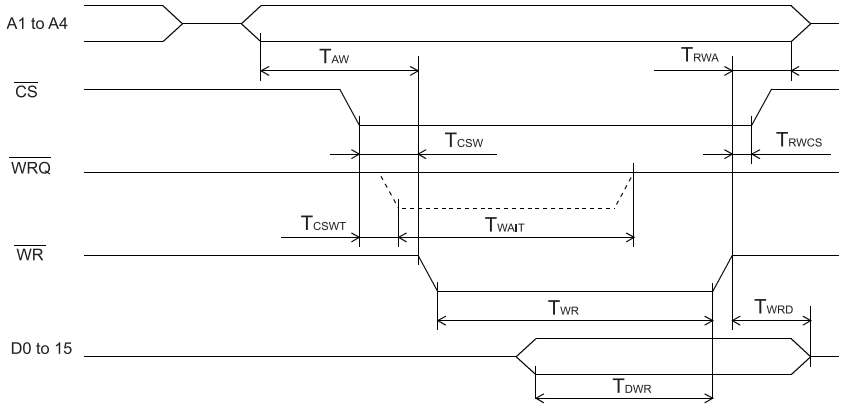
Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{RD}$ ↓	T_{AR}		10		ns
Address setup time for $\overline{WR}$ ↓	T_{AW}		10		ns
Address hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWA}		0		ns
$\overline{CS}$ setup time for $\overline{RD}$↓	T_{CSR}		4		ns
$\overline{CS}$ setup time for $\overline{WR}$ ↓	T_{CSW}		4		ns
$\overline{CS}$ hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWCS}		0		ns
$\overline{WRQ}$ ON delay time for $\overline{CS}$ ↓	T_{CSWT}	C_L = 40pF		13	ns
$\overline{WRQ}$ signal LOW time	T_{WAIT}			$4T_{CLK}$	ns
Data output delay time for $\overline{RD}$ ↓	T_{RDLD}	C_L = 40pF		21	ns
Data output delay time for $\overline{WRQ}$ ↑	T_{WTHD}	C_L = 40pF		10	ns
Data float delay time for $\overline{RD}$ ↑	T_{RDHD}	C_L = 40pF		18	ns
$\overline{WR}$ signal width	T_{WR}	Note 1	11		ns
Data setup time for $\overline{WR}$ ↑	T_{DWR}		12		ns
Data hold time for $\overline{WR}$ ↑	T_{WRD}		0		ns

Note 1: When a $\overline{WRQ}$ signal is output, the duration will be the interval between $\overline{WRQ}$ = H and $\overline{WR}$ = H.

<Read cycle>



<Write cycle>

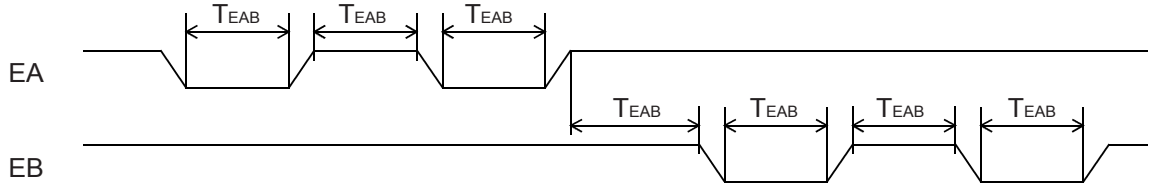


12-6. Operation timing (common for all axes)

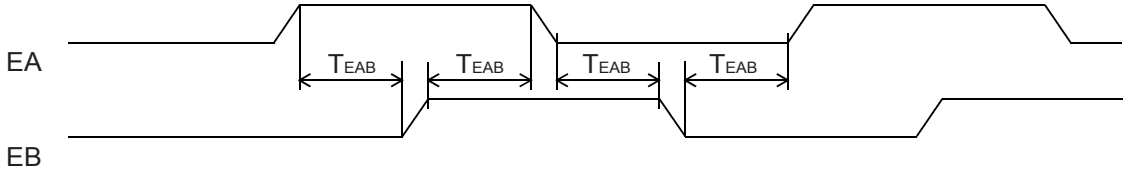
Item	Symbol	Condition	Min.	Max.	Unit
$\overline{RST}$ input signal length		Note 1	10 T_{CLK}		ns
EA, EB, EZ input signal length	T_{EAB}	RENV2: EINF="0"	T_{CLK}		ns
		RENV2: EINF="1"	3 T_{CLK}		
PA, PB input signal length	T_{PAB}	RENV2: PINF="0"	T_{CLK}		ns
		RENV2: PINF="1"	3 T_{CLK}		
ERC output signal length		RENV1 : EPW = "000"	225 T_{CLK}	240 T_{CLK}	ns
		RENV1 : EPW = "001"	1793 T_{CLK}	1920T_{CLK}	
		RENV1 : EPW = "010"	7169 T_{CLK}	7680 T_{CLK}	
		RENV1 : EPW = "011"	28673 T_{CLK}	30720 T_{CLK}	
		RENV1 : EPW = "100"	229377 T_{CLK}	245760 T_{CLK}	
		RENV1 : EPW = "101"	917505 T_{CLK}	983040 T_{CLK}	
		RENV1 : EPW = "110"	1835009 T_{CLK}	1966080 T_{CLK}	
		RENV1 : EPW = "111"	(Level output)		
ERC signal OFF timer time		RENV1 : ETW = "01"	225 T_{CLK}	240 T_{CLK}	ns
		RENV1 : ETW = "10"	28673 T_{CLK}	30720 T_{CLK}	
		RENV1 : ETW = "11"	1835009 T_{CLK}	1966080 T_{CLK}	
+EL, -EL, SD, ORG, ALM, INP input signal length		RENV1 : FLTR = "0"	T_{CLK}		ns
		FLTR="1" & FTM="00"	64 T_{CLK}		
		FLTR="1" & FTM="01"	512 T_{CLK}		
		FLTR="1" & FTM="10"	4096 T_{CLK}		
		FLTR="1" & FTM="11"	32768 T_{CLK}		
+DR(PA), -DR(PB), $\overline{PE}$ input signal length		RENV1 : DRF = "0"	T_{CLK}		ns
		RENV1 : DRF = "1"	1048576 T_{CLK}		
Direction change timer time		RENV1 : DTMF = "0" (ON)	3585 T_{CLK}	3840 T_{CLK}	ns
PCS input signal width			T_{CLK}		ns
LTC input signal width			T_{CLK}		ns
$\overline{CSTA}$ Output signal length			8 T_{CLK}		ns
$\overline{CSTA}$ Input signal length			4 T_{CLK}		ns
$\overline{CSTP}$ Output signal length			8 T_{CLK}		ns
$\overline{CSTP}$ Input signal length			4 T_{CLK}		ns
$\overline{BSY}$ signal ON delay time	T_{CMDBSY}		4 T_{CLK}	5 T_{CLK}	ns
	T_{STABSY}		4 T_{CLK}	5 T_{CLK}	ns
Start delay time	T_{CMDPLS}		15 T_{CLK}	16 T_{CLK}	ns
	T_{STAPLS}		15 T_{CLK}	16 T_{CLK}	ns
Deceleration delay time	T_{CMDFDW}		5 T_{CLK}	6 T_{CLK}	ns
	T_{SDFDW}		5 T_{CLK}	34 T_{CLK}	ns

Note 1: The actual CLK input signal is 10 cycles longer while the $\overline{RST}$ terminal is LOW.

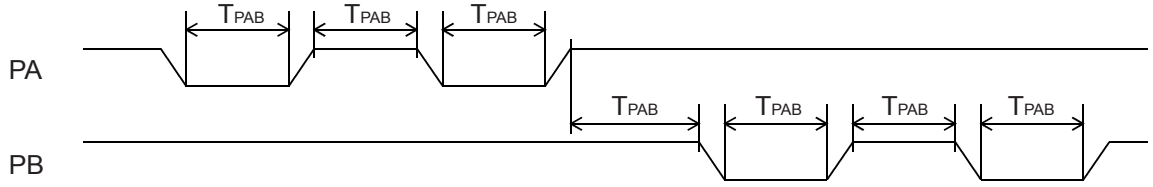
1) When the EA, EB inputs are in the 2-pulse mode



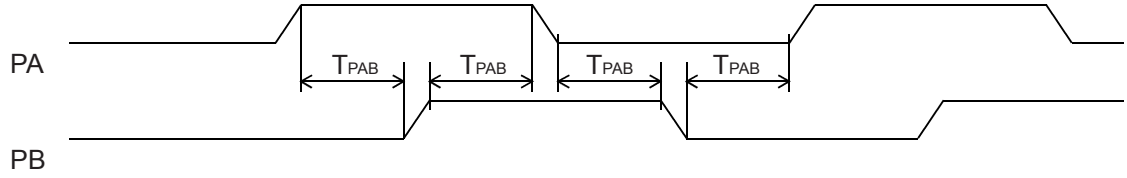
2) When the EA, EB inputs are in the 90 ^{o} phase-difference mode



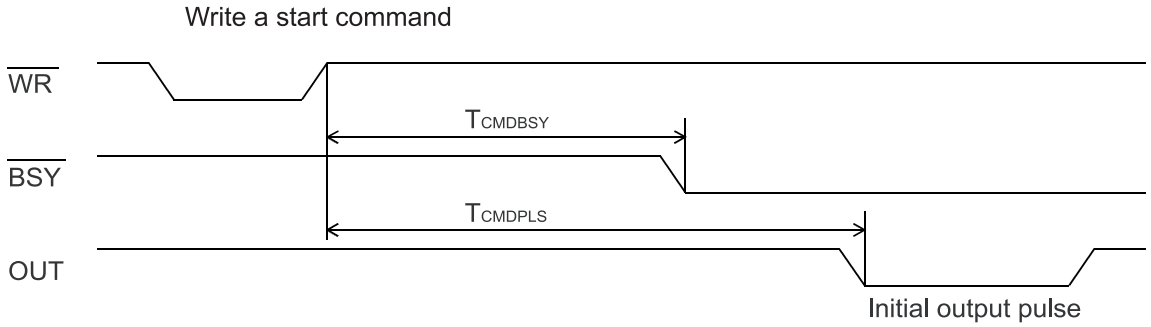
3) When the PA, PB inputs are in the 2-pulse mode



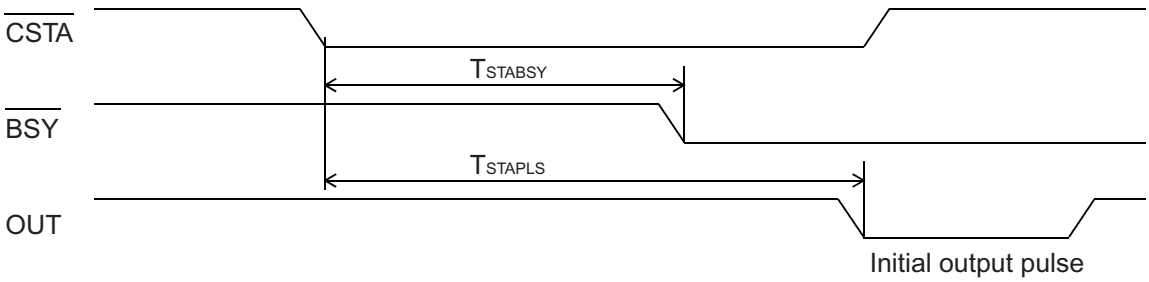
4) When the PA, PB inputs are in the 90^{o} phase-difference mode



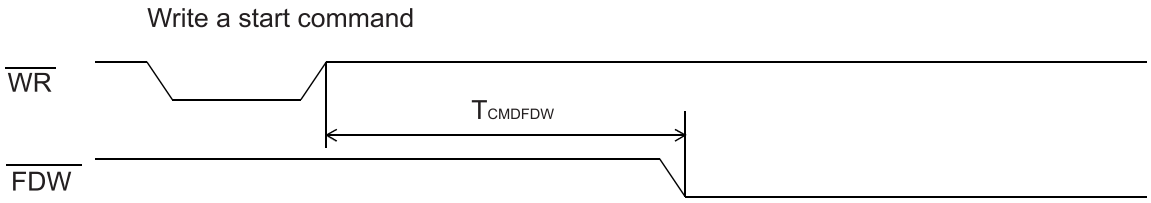
5) Timing for the command mode (when I/M = H, and B/$\overline{W}$ = H)



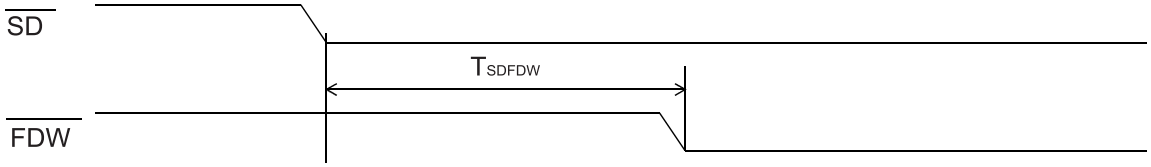
6) Simultaneous start timing



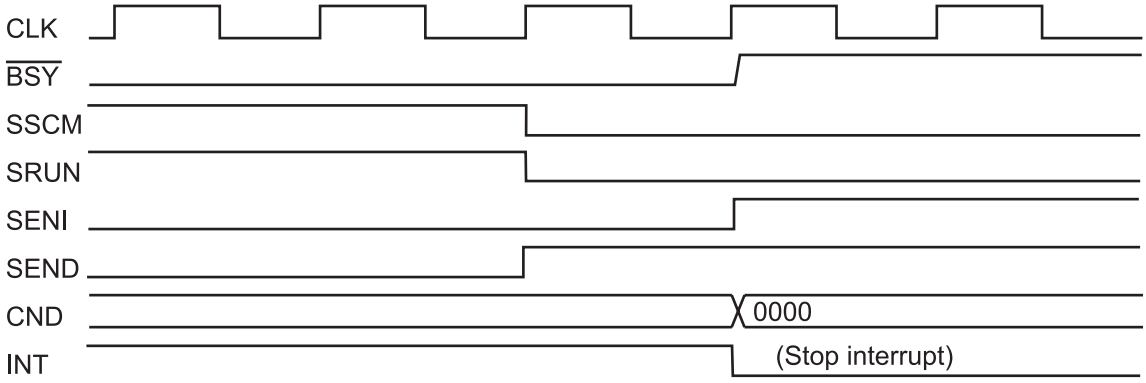
7) Deceleration start timing triggered by a command



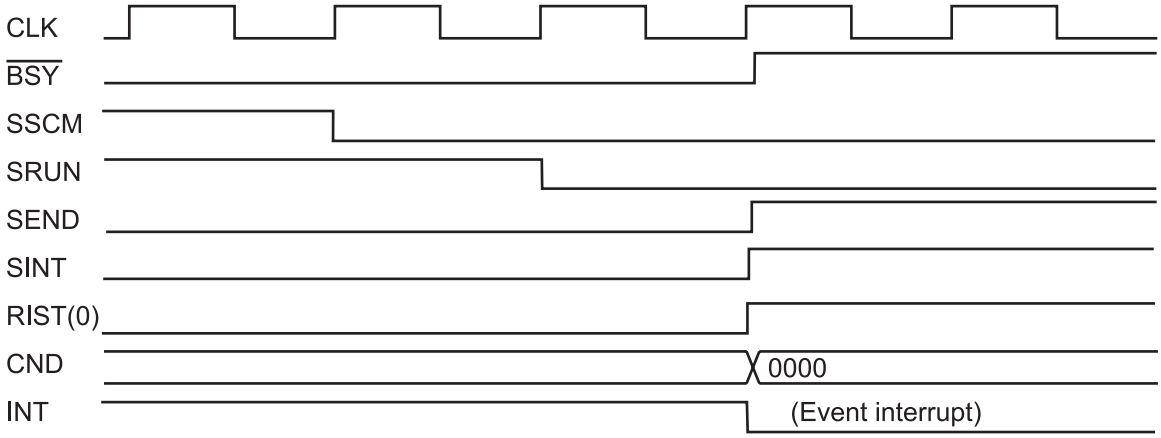
8) Deceleration start timing triggered by the SD input



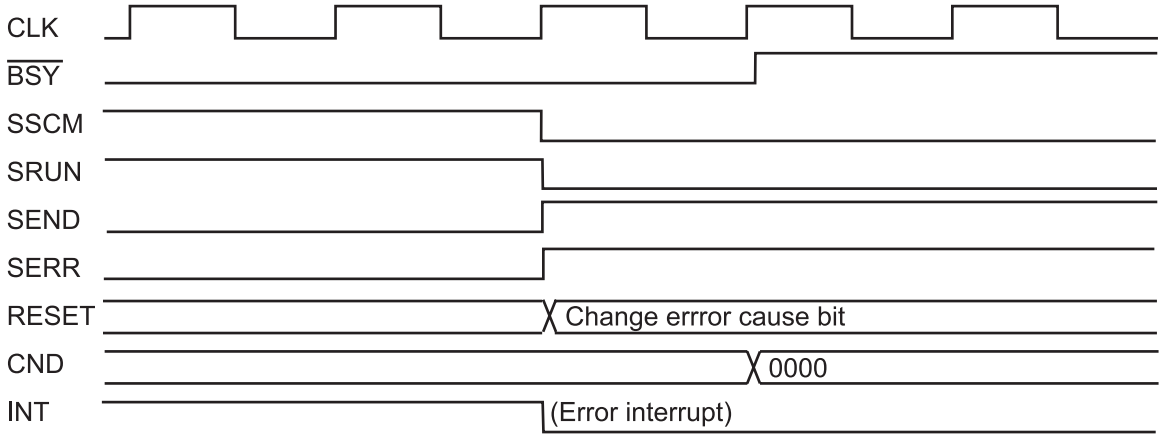
9) Stop timing by a command



10) Stop timing by normal automatic stop



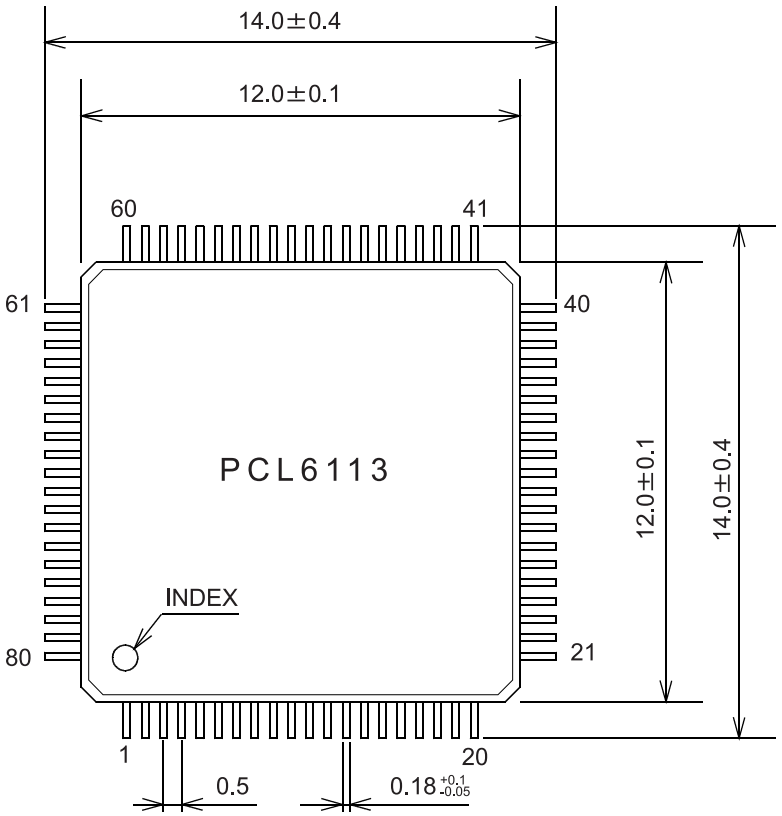
11) Stop timing by error

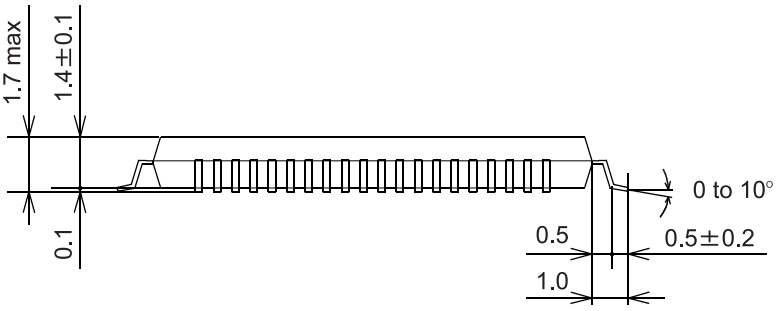


13. External Dimensions

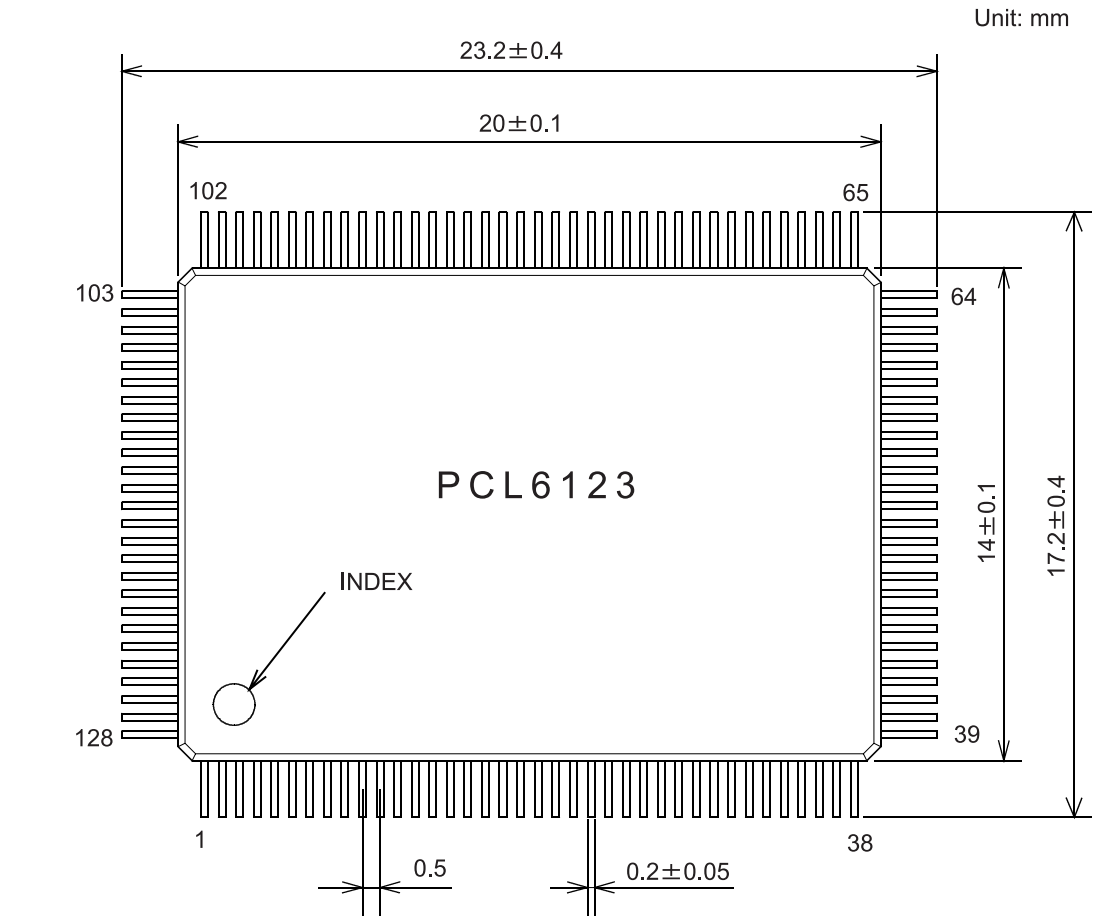
13-1. PCL6113

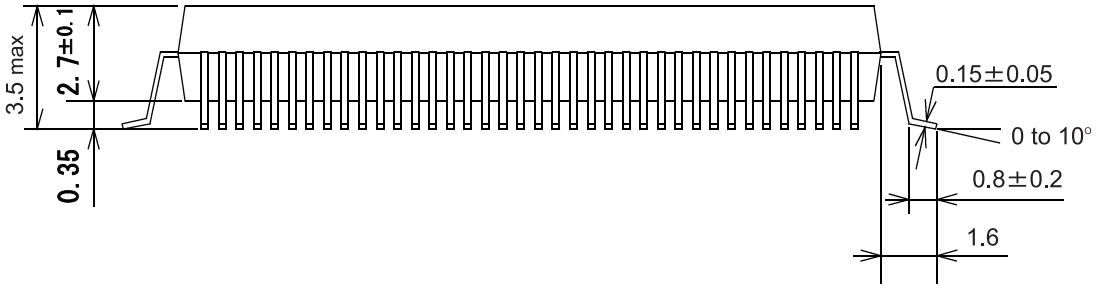
Unit: mm



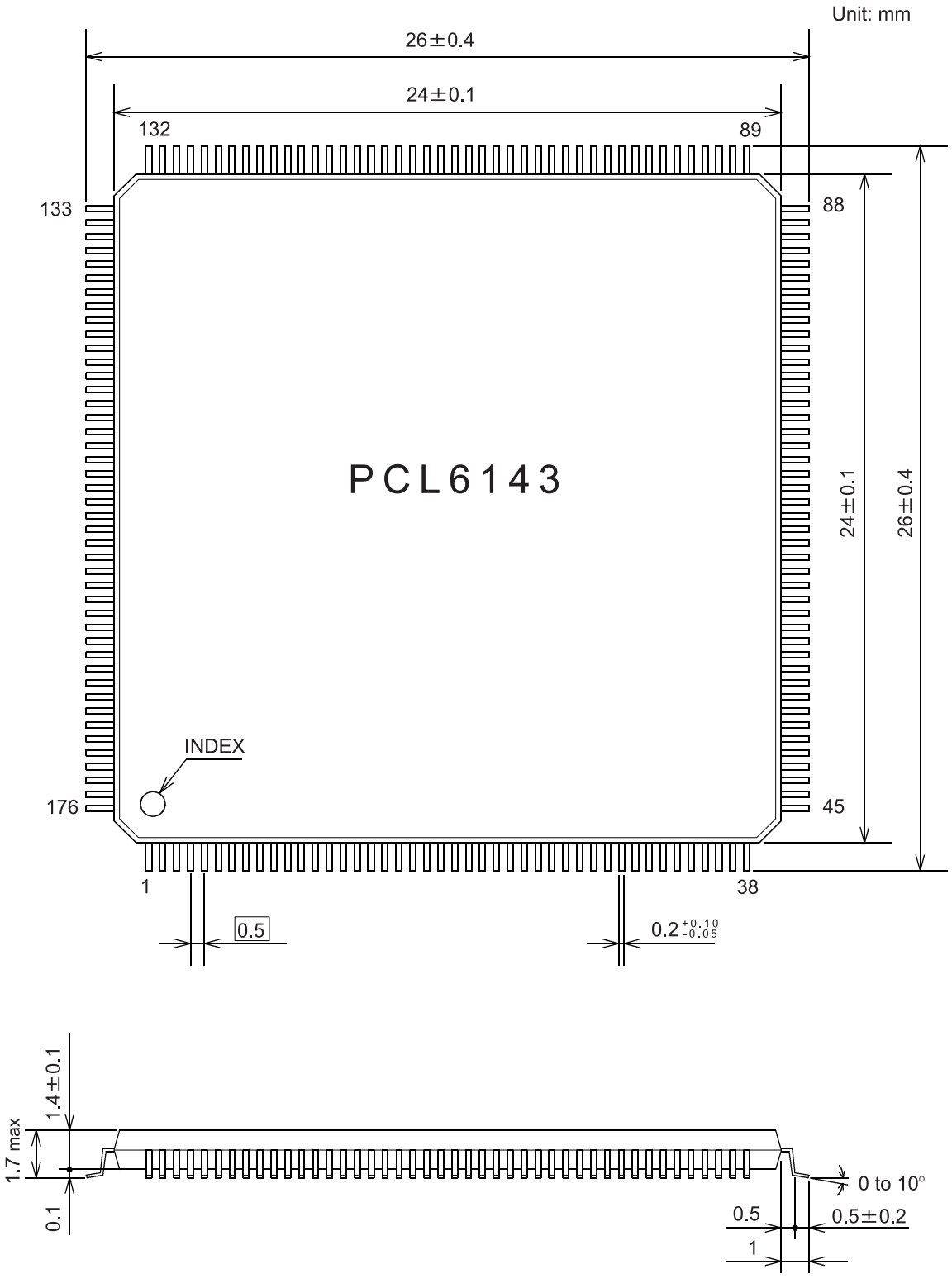


13-2. PCL6123


Unit: mm
23.2±0.4
20±0.1
102
65
103
64
PCL6123
INDEX
14±0.1
17.2±0.4
128
39
1
38
0.5
0.2±0.05


3.5 max
2.7±0.1
0.35
0.15±0.05
0 to 10°
0.8±0.2
1.6

13-3. PCL6143


Unit: mm
26±0.4
24±0.1
132
89
133
88
PCL6143
24±0.1
26±0.4
INDEX
176
45
1
38
0.5
0.2 +0.10 -0.05
1.4±0.1
1.7 max
0.1
0 to 10°
0.5
0.5±0.2
1

Appendix: List of various items

Appendix 1: List of commands

<Operation commands>

COMB0	Symbol	Description	COMB0	Symbol	Description
05h	CMEMG	Emergency stop	50h	STAFL	FL low speed start
06h	CMSTA	$\overline{\text{CSTA}}$ output (simultaneous start)	51h	STAFH	FH low speed start
07h	CMSTP	$\overline{\text{CSTP}}$ output (simultaneous stop)	52h	STAD	High speed start 1 (FH low speed -> Deceleration stop)
40h	FCHGL	Immediate change to FL low speed	53h	STAUD	High speed start 2 (acceleration -> FH low speed -> deceleration)
41h	FCHGH	Immediate change to FH low speed	54h	CNTFL	FL low speed start for remaining number of pulses
42h	FSCHL	Decelerate to FL speed	55h	CNTFH	FH low speed start for remaining number of pulses
43h	FSCHH	Accelerate to FH speed	56h	CNTD	High speed start 1 for remaining number of pulses
49h	STOP	Immediate stop	57h	CNTUD	High speed start 2 for remaining number of pulses
4Ah	SDSTP	Deceleration stop			

< General-purpose port control commands>

COMB0	Symbol	Description	COMB0	Symbol	Description
10h	P0RST	Set the P0 terminal LOW	18h	P0SET	Set the P0 terminal HIGH
11h	P1RST	Set the P1 terminal LOW	19h	P1SET	Set the P1 terminal HIGH
12h	P2RST	Set the P2 terminal LOW	1Ah	P2SET	Set the P2 terminal HIGH
13h	P3RST	Set the P3 terminal LOW	1Bh	P3SET	Set the P3 terminal HIGH
14h	P4RST	Set the P4 terminal LOW	1Ch	P4SET	Set the P4 terminal HIGH
15h	P5RST	Set the P5 terminal LOW	1Dh	P5SET	Set the P5 terminal HIGH
16h	P6RST	Set the P6 terminal LOW	1Eh	P6SET	Set the P6 terminal HIGH
17h	P7RST	Set the P7 terminal LOW	1Fh	P7SET	Set the P7 terminal HIGH

<Control commands>

COMB0	Symbol	Description	COMB0	Symbol	Description
00h	NOP	(Invalid command)	25h	ERCRST	Reset the ERC signal
04h	SRST	Software reset	26h	PRECAN	Cancel the pre-register
20h	CUN1R	Reset COUNTER1	28h	STAON	Substitute PCS input
21h	CUN2R	Reset COUNTER2	29h	LTCH	Substitute LTC input
24h	ERCOUT	Output an ERC signal	2Ah	SPSTA	Uses the same process as the $\overline{\text{CSTA}}$ input, but for this axis

<Register control commands>

No.	Description	Bit	Register					Pre-register				
			Name	Read command		Write command		Name	Read command		Write command	
				COMB0	Symbol	COMB0	Symbol		COMB0	Symbol	COMB0	Symbol
1	Feed amount	28	RMV	D0h	RRMV	90h	WRMV	PRMV	C0h	RPRMV	80h	WPRMV
2	Initial speed	14	RFL	D1h	RRFL	91h	WRFL	PRFL	C1h	RPRFL	81h	WPRFL
3	Operation speed	14	RFH	D2h	RRFH	92h	WRFH	PRFH	C2h	RPRFH	82h	WPRFH
4	Acceleration rate	14	RUR	D3h	RRUR	93h	WRUR	PRUR	C3h	RPRUR	83h	WPRUR
5	Deceleration rate	14	RDR	D4h	RRDR	94h	WRDR	PRDR	C4h	RPRDR	84h	WPRDR
6	Speed magnification rate	12	RMG	D5h	RRMG	95h	WRMG	PRMG	C5h	RPRMG	85h	WPRMG
7	Ramping-down point	24	RDP	D6h	RRDP	96h	WRDP	PRDP	C6h	RPRDP	86h	WPRDP
8	Operation mode	30	RMD	D7h	RRMD	97h	WRMD	PRMD	C7h	RPRMD	87h	WPRMD
9	Linear interpolation main axis data	27	RIP	D8h	RRIP	98h	WRIP	PRIP	C8h	RPRIP	88h	WPRIP
10	S-curve range while accelerating	13	RUS	D9h	RRUS	99h	WRUS	PRUS	C9h	RPRUS	89h	WPRUS
11	S-curve range while decelerating	13	RDS	DAh	RRDS	9Ah	WRDS	PRDS	CAh	RPRDS	8Ah	WPRDS
12	Environment setting 1	32	RENV1	DCh	RRENV1	9Ch	WRENV1					
13	Environment setting 2	31	RENV2	DDh	RRENV2	9Dh	WRENV2					
14	Environment setting 3	22	RENV3	DEh	RRENV3	9Eh	WRENV3					
15	COUNTER1 (command position)	28	RCUN1	E3h	RRCUN1	A3h	WRCUN1					
16	COUNTER2 (mechanical position)	28	RCUN2	E4h	RRCUN2	A4h	WRCUN2					
17	Comparator 1 data	28	RCMP1	E7h	RRCMP1	A7h	WRCMP1					
18	Comparator 2 data	28	RCMP2	E8h	RRCMP2	A8h	WRCMP2					
19	Enable various event interrupts (INTs)	12	RIRQ	ECh	RRIRQ	ACh	WRIRQ					
21	COUNTER1 latch data	28	RLTC1	EDh	RRLTC1							
22	COUNTER2 latch data	28	RLTC2	EEh	RRLTC2							
23	Extension status	17	RSTS	F1h	RRSTS							
24	Error INT status	9	REST	F2h	RREST							
25	Event INT status	13	RIST	F3h	RRIST							
26	Positioning counter	28	RPLS	F4h	RRPLS							
28	EZ counter, speed monitor	20	RSPD	F5h	RRSPD							
27	Ramping-down point	24	RSDC	F6h	RRSDC							

Appendix 2: Label list

Label	Type	Position	Description	Reference
A0	Terminal name	7, 7, 7	Address bus 0 (LSB) (PCL6113, 6123, 6143)	P8, 17
A1	Terminal name	8, 8, 8	Address bus 1 (PCL6113, 6123, 6143)	P8, 17
A2	Terminal name	9, 9, 9	Address bus 2 (PCL6113, 6123, 6143)	P8, 17
A3	Terminal name	10, 10	Address bus 3 (PCL 6123, 6143)	P8, 17
A4	Terminal name	11	Address bus 4 (MSB) (PCL6143)	P8, 17
ALM	Terminal name	39	Driver alarm signal (PCL 6113)	P10, 83
ALML	Register bit	RENV1 9	Set the input logic for the ALM signal (0: Negative, 1: Positive)	P38, 83
ALMM	Register bit	RENV1 8	Select the process to use when the ALM input is ON (0: Immediate stop, 1: Deceleration stop)	P38, 83
ALMu	Terminal name	134	U axis driver alarm signal (PCL6143)	P10, 83
ALMx	Terminal name	40, 41	X axis driver alarm signal (PCL 6123, 6143)	P10, 83
ALMy	Terminal name	77, 72	Y axis driver alarm signal (PCL 6123, 6143)	P10, 83
ALMz	Terminal name	102	Z axis driver alarm signal (PCL6143)	P10, 83
AS0 to 13	Register bit	RSPD 0-15	Monitor current speed	P47
$\overline{BSY}$	Terminal name	64	Operation monitor output (PCL6113)	P13
$\overline{BSYu}$	Terminal name	159	Operation monitor output for the U axis (PCL6143)	P13
$\overline{BSYx}$	Terminal name	65, 66	Operation monitor output for the X axis (PCL 6123, 6143)	P13
$\overline{BSYy}$	Terminal name	102, 97	Operation monitor output for the Y axis (PCL 6123, 6143)	P13
$\overline{BSYz}$	Terminal name	128	Operation monitor output for the Z axis (PCL6143)	P13
BUFB0	Byte map name	4 for Z80	Write/read the input/output buffer (bits 0 to 7).	P18, 20
BUFB1	Byte map name	5 for Z80	Write/read the input/output buffer (bits 8 to 15)	P18, 20
BUFB2	Byte map name	6 for Z80	Write/read the input/output buffer (bits 16 to 23)	P18, 20
BUFB3	Byte map name	7 for Z80	Write/read the input/output buffer (bits 24 to 31)	P18, 20
BUFW0	Word map name	4 for 8086	Write/read the input/output buffer (bits 0 to 15)	P18, 21
BUFW1	Word map name	6 for 8086	Write/read the input/output buffer (bits 16 to 31)	P18, 21
C1S0 to 1	Register bit	RENV3 12-13	Select a comparison counter for comparator1	P42, 93
C1RM	Register bit	RENV3 7	Set COUNTER1 for ring count operation using Comparator 1.	P42, 93
C2S0 to 1	Register bit	RENV3 14-15	Select a comparison counter for Comparator 2	P42, 93
C2RM	Register bit	RENV3 11	Set COUNTER2 for ring count operation using Comparator 2	P42, 93
$\overline{CEMG}$	Terminal name	77, 124, 170	Emergency stop signal (PCL6113, 6123, 6143)	P8, 87
CIS1	Register bit	RENV3 0	Set COUNTER1 to count input pulses	P42, 92
CIS2	Register bit	RENV3 1	Set COUNTER2 to count input pulses	P42, 92
CLK	Terminal name	69, 113, 163	Reference clock (19.6608 MHz as standard) (PCL6113, 6123, 6143)	P7
CMEMG	Command	05h	Emergency stop	P25, 87
CMSTA	Command	06h	Output $\overline{CSTA}$ (simultaneous start) signal	P25, 85
CMSTP	Command	07h	Output $\overline{CSTP}$ (simultaneous stop) signal	P25, 87
CND0 to 3	Register bit	RSTS 0-3	Operation status monitor	P45
CNTD	Command	56h	Remaining high speed start pulses (FH low speed -> Deceleration stop)	P25
CNTFH	Command	55h	Remaining pulses FH low speed start pulses	P25
CNTFL	Command	54h	Remaining pulses FL low speed start pulses	P25
CNTUD	Command	57h	Remaining high speed start pulses (accelerate -> FH low speed -> deceleration stop)	P25
COMB0	Byte map name		Write control command	P19
COMB1	Byte map name		Axis selection	P19
COMW	Word map name		Assign an axis, or write a control command	P19
COUNTER1	Circuit name		28-bit counter1 for command position control	P43, 88
COUNTER2	Circuit name		28-bit counter2 for mechanical position control	P43, 88
$\overline{CP1x}$	Terminal name	70	Comparator 1 output signal for the X axis (PCL6123)	P13
$\overline{CP1y}$	Terminal name	107	Comparator 1 output signal for the Y axis (PCL6123)	P13
$\overline{CP2x}$	Terminal name	71	Comparator 2 output signal for the X axis (PCL6123)	P13
$\overline{CP2y}$	Terminal name	108	Comparator 2 output signal for the Y axis (PCL6123)	P13
$\overline{CS}$	Terminal name	4, 4, 4	Chip select signal (PCL6113, 6123, 6143)	P8
$\overline{CSD}$	Terminal name	74, 121, 167	Simultaneous deceleration signal (PCL6113, 6123, 6143)	P9
CSPO	Register bit	RENV2 13	Output a $\overline{CSTP}$ signal even though an axis is stopped by a command	P40
$\overline{CSTA}$	Terminal name	75, 122, 168	Simultaneous start signal (PCL6113, 6123, 6143)	P9, 85
$\overline{CSTP}$	Terminal name	76, 123, 169	Simultaneous stop signal (PCL6113, 6123, 6143)	P9, 87
CU1H	Register bit	RENV3 2	Stop counting on COUNTER1	P42, 91
CU1L	Register bit	RENV3 4	Reset COUNTER1 right after latching the count value.	P42, 90
CU1R	Register bit	RENV3 6	Reset COUNTER1 when the zero return is complete.	P42, 90
CU2H	Register bit	RENV3 3	Stop the count on COUNTER2	P42, 90
CU2L	Register bit	RENV3 8	Reset COUNTER2 right after latching the count value.	P42, 90
CU2R	Register bit	RENV3 10	Reset COUNTER2 when the zero return is complete	P42, 90
CUN1R	Command	20h	Reset COUNTER1	P27, 91
CUN2R	Command	21h	Reset COUNTER2	P27, 91
D0	Terminal name	15, 16, 17	Data bus 0 (LSB) (PCL6113, 6123, 6143)	P8
D1	Terminal name	16, 17, 18	Data bus 1 (PCL6113, 6123, 6143)	P8
D10	Terminal name	27, 28, 29	Data bus 10 (PCL6113, 6123, 6143)	P8
D11	Terminal name	28, 29, 30	Data bus 11 (PCL6113, 6123, 6143)	P8
D12	Terminal name	30, 31, 32	Data bus 12 (PCL6113, 6123, 6143)	P8
D13	Terminal name	31, 32, 33	Data bus 13 (PCL6113, 6123, 6143)	P8
D14	Terminal name	32, 33, 34	Data bus 14 (PCL6113, 6123, 6143)	P8

Label	Type	Position	Description	Reference
D15	Terminal name	33, 34, 35	Data bus 15 (MSB) (PCL6113, 6123, 6143)	P8
D2	Terminal name	17, 18, 19	Data bus 2 (PCL6113, 6123, 6143)	P8
D3	Terminal name	18, 19, 20	Data bus 3 (PCL6113, 6123, 6143)	P8
D4	Terminal name	20, 21, 22	Data bus 4 (PCL6113, 6123, 6143)	P8
D5	Terminal name	21, 22, 23	Data bus 5 (PCL6113, 6123, 6143)	P8
D6	Terminal name	22, 23, 24,	Data bus 6 (PCL6113, 6123, 6143)	P8
D7	Terminal name	23, 24, 25	Data bus 7 (PCL6113, 6123, 6143)	P8
D8	Terminal name	25, 26, 27	Data bus 8 (PCL6113, 6123, 6143)	P8
D9	Terminal name	26, 27, 28	Data bus 9 (PCL6113, 6123, 6143)	P8
DIR	Terminal name	62	Direction signal for driving a motor (PCL6113)	P13
DIRu	Terminal name	157	Motor drive direction signal for the U axis (PCL6143)	P13
DIRx	Terminal name	63, 64	Motor drive direction signal for the X axis (PCL6123, 6143)	P13
DIRy	Terminal name	100, 95	Motor drive direction signal for the Y axis (PCL6123, 6143)	P13
DIRz	Terminal name	126	Motor drive direction signal for the Z axis (PCL6143)	P13
DRF	Register bit	RENV1 27	Apply a filter to +DR, -DR signal input	P39, 52
DRL	Register bit	RENV1 25	Select +DR, -DR signal input logic (0: Negative logic, 1: Positive logic)	P39, 52
DTMF	Register bit	RENV1 28	Turn OFF the direction change timer (0.2 msec)	P39
EA	Terminal name	44	Encoder A phase signal (PCL6113)	P11
EAu	Terminal name	137	Encoder A phase signal for the U axis (PCL6143)	P11
EAx	Terminal name	45, 46	Encoder A phase signal for the X axis (PCL6123, 6143)	P11
EAy	Terminal name	82, 77	Encoder A phase signal for the Y axis (PCL6123, 6143)	P11
EAz	Terminal name	106	Encoder A phase signal for the Z axis (PCL6143)	P11
EB	Terminal name	45	Encoder B phase signal (PCL6113)	P11
EBu	Terminal name	140	Encoder B phase signal for the U axis (PCL6143)	P11
EBx	Terminal name	46, 47	Encoder B phase signal for the X axis. (PCL6123, 6143)	P11
EBy	Terminal name	83, 78	Encoder B phase signal for the Y axis (PCL6123, 6143)	P11
EBz	Terminal name	109	Encoder B phase signal for the Z axis (PCL6143)	P11
ECZ0 to 3	Register bit	RSPD 16-19	Read the count value of the EZ input to monitor the zero return	P47
EDIR	Register bit	RENV2 19	Reverse the EA, EB input count direction	P40
EIM0 to 1	Register bit	RENV2 16-17	Specify the EA, EB input parameters	P40
EINF	Register bit	RENV2 18	Apply a noise filter to the EA/EB input	P40
+EL	Terminal name	35	(+) end limit signal (PCL6113)	P10
-EL	Terminal name	36	(-) end limit signal (PCL6113)	P10
ELL	Terminal name	78	Select the input logic for the end limit signal (PCL6113)	P10
ELLu	Terminal name	174	Set the input logic of the end limit signal for the U axis (PCL6143)	P9
ELLx	Terminal name	125, 171	Set the input logic of the end limit signal for the X axis (PCL6123, 6143)	P9
ELLy	Terminal name	126, 172	Select the input logic of the end limit signal for the Y axis (PCL6123, 6143)	P9
ELLz	Terminal name	173	Set the input logic of the end limit signal for the Z axis (PCL6143)	P9
ELM	Register bit	RENV1 3	Select the process to execute when the EL input is ON (0: Immediate stop, 1: Deceleration stop)	P38
+ELu	Terminal name	130	(+) end limit signal for the U axis (PCL6143)	P9
-ELu	Terminal name	131	(-) end limit signal for the U axis (PCL6143)	P10
+ELx	Terminal name	36, 37	(+) end limit signal for the X axis (PCL6123, 6143)	P9
-ELx	Terminal name	37, 38	(-) end limit signal for the X axis (PCL6123, 6143)	P10
+ELy	Terminal name	73, 68	(+) end limit signal for the Y axis (PCL6123, 6143)	P9
-ELy	Terminal name	74, 69	(-) end limit signal for the Y axis (PCL6123, 6143)	P10
+ELz	Terminal name	99	(+) end limit signal for the Z axis (PCL6143)	P9
-ELz	Terminal name	100	(-) end limit signal for the Z axis (PCL6143)	P10
EOFF	Register bit	RENV2 14	Invalid EA, EB input	P40
EPW0 to 2	Register bit	RENV1 12-14	Specify the ERC output signal pulse width	P39, 82
ERCL	Register bit	RENV1 15	Set the output logic of the ERC signal (0: Negative logic, 1: Positive logic)	P39, 82
ERCOUT	Command	24h	Output an ERC signal	P27, 83
ERCRST	Command	25h	Reset the output when the ERC signal is set to level output	P27, 83
ERC	Terminal name	63	Clear driver deviation counter output (PCL6113)	P13
ERCu	Terminal name	158	Driver deflection clear output for the U axis (PCL6143)	P13
ERCx	Terminal name	64, 65	Driver deflection clear output for the X axis (PCL6123, 6143)	P13
ERCy	Terminal name	101, 96	Driver deflection clear output for the Y axis (PCL6123, 6143)	P13
ERCz	Terminal name	127	Driver deflection clear output for the Z axis (PCL6143)	P13
EROE	Register bit	RENV1 10	Automatic output of the ERC signal	P39, 82
EROR	Register bit	RENV1 11	Auto output an ERC signal when the zero return is complete	P39, 82
ESAL	Register bit	REST 2	Stops by inputting ALM ON input	P46
ESEE	Register bit	REST 7	An EA/EB input error occurred	P46
ESEM	Register bit	REST 4	Stops by inputting $\overline{\text{EMG}}$ ON input	P46
ESML	Register bit	REST 1	Stopped because the ñEL input turned ON	P46
ESPE	Register bit	REST 8	A PA/PB input error occurred	P46
ESPL	Register bit	REST 0	Stopped because the + EL input turned ON	P46
ESPO	Register bit	REST 6	The PA/PB input buffer counter overflowed	P46
ESSD	Register bit	REST 5	Deceleration stop caused by the SD input turning ON	P46
ESSP	Register bit	REST 3	Stops by inputting $\overline{\text{STP}}$ ON input	P46
ETW0 to 1	Register bits	RENV1 16-17	Specify the ERC signal OFF timer	P39, 83
EZ	Terminal name	46	Encoder Z phase signal (PCL6113)	P11
EZD0 to 3	Register bits	RENV2 24-27	Enter an EZ count value for a zero return	P41, 54

Label	Type	Position	Description	Reference
EZL	Register bit	RENV2 28	Set the input logic for the EZ signal (0: Falling, 1: Rising)	P41, 54
EZu	Terminal name	141	U axis encoder Z phase signal (PCL6143)	P11
EZx	Terminal name	47, 48	X axis encoder Z phase signal (PCL6123, 6143)	P11
EZy	Terminal name	84, 79	Y axis encoder Z phase signal (PCL6123, 6143)	P11
EZz	Terminal name	110	Z axis encoder Z phase signal (PCL6143)	P11
FCHGH	Command	41h	Change immediately to FH speed	P25
FCHGL	Command	40h	Change immediately to FL speed	P25
$\overline{\text{FDWx}}$	Terminal name	68	Deceleration monitor output for the x axis (PCL6123)	P13
$\overline{\text{FDWy}}$	Terminal name	105	Deceleration monitor output for the y axis (PCL6123)	P13
FLTR	Register bit	RENV1 26	Apply input filter	P39
FSCHH	Command	43h	Accelerate to FH speed	P25
FSCHL	Command	42h	Accelerate to FL speed	P25
FTM0 to 1	Register bit	RENV1 20-21	Set a filter time constant for +EL, -EL, SD, ORG, ALM, and INP	P39
$\overline{\text{FUPx}}$	Terminal name	67	Acceleration monitor output for the x axis. (PCL6123)	P13
$\overline{\text{FUPy}}$	Terminal name	104	Acceleration monitor output for the y axis (PCL6123)	P13
IEND	Register bit	RENV2 30	Specify that the stop interrupt will be output.	P41
IF0	Terminal name	1, 1, 1	CPU-I/F mode selection 0 (PCL6113, 6123, 6143)	P7
IF1	Terminal name	2, 2, 2	CPU-I/F mode selection 1 (PCL6113, 6123, 6143)	P7
IFB	Terminal name	13, 14, 15	Busy CPU-I/F (PCL6113, 6123, 6143)	P8
INP	Terminal name	41	In-position input (PCL6113)	P10
INPL	Register bit	RENV1 22	Select input logic of INP signal (0: Negative, 1: Positive)	P39
INPu	Terminal name	150	In position input for the U axis (PCL6143)	P10
INPx	Terminal name	42, 43	In position input for the X axis (PCL6123, 6143)	P10
INPy	Terminal name	79, 74	In position input for the Y axis (PCL6123, 6143)	P10
INPz	Terminal name	105	In position input for the Z axis (PCL6143)	P10
$\overline{\text{INT}}$	Terminal name	11, 12, 13	Interrupt request signal	P8
INTM	Register bit	RENV1 29	Mask the INT output terminal	P39
IOP0 to 7	Sub-status bits	SSTSW 0-7	Read the P0 to P7 terminal status.	P25
IOPB	Byte map name		Read the general I/O port	P18
IRC1	Register bit	RIRQ 6	Enable an INT when the Comparator 1 conditions are met	P44
IRC2	Register bit	RIRQ 7	Enable an INT when the Comparator 2 conditions are met	P44
IRDE	Register bit	RIRQ 5	Enable an INT when the deceleration is finished	P44
IRDR	Register bit	RIRQ 11	Enable an INT when the ±DR (PA, PB) input changes	P44
IRDS	Register bit	RIRQ 4	Enable an INT when the deceleration starts	P44
IREN	Register bit	RIRQ 0	Enable an INT when there is a normal stop	P44
IRLT	Register bit	RIRQ 8	Enable an INT when the count value is latched by an LTC input	P44
IRNM	Register bit	RIRQ 1	Enable an INT when writing to the pre-register for operation is enabled	P44
IROL	Register bit	RIRQ 9	Enable an INT when the count value is latched by an ORG input	P44
IRSA	Register bit	RIRQ 12	Enable an INT by turning ON the $\overline{\text{CSTA}}$ input	P44
IRSD	Register bit	RIRQ 10	Enable an INT by turning ON the SD input	P44
IRUE	Register bit	RIRQ 3	Enable an INT when the acceleration is finished	P44
IRUS	Register bit	RIRQ 2	Enable an INT when acceleration starts	P44
ISC1	Register bit	RIST 6	Comparator 1 conditioned status	P46
ISC2	Register bit	RIST 7	Comparator 2 conditioned status	P46
ISDE	Register bit	RIST 5	Equals 1 when deceleration is finished	P46
ISDS	Register bit	RIST 4	Equals 1 when deceleration starts	P46
ISEN	Register bit	RIST 0	Equals 1 when stopped automatically	P46
ISLT	Register bit	RIST 8	Equals 1 when the count value is latched by an LTC input	P46
ISMD	Register bit	RIST 12	Equals 1 when a -DR input signal is ON.	P47
ISNM	Register bit	RIST 1	Enable writing to the pre-register	P46
ISOL	Register bit	RIST 9	Latched count value from the ORG input	P46
ISPD	Register bit	RIST 11	Equals 1 when the +DR (PA) input is ON	P47
ISSA	Register bit	RIST 13	Equals 1 when the $\overline{\text{CSTA}}$ input is ON	P47
ISSD	Register bit	RIST 10	Equals 1 when the SD input is ON	P47
ISUE	Register bit	RIST 3	Equals 1 when the acceleration is finished	P46
ISUS	Register bit	RIST 2	Equals 1 when to start acceleration	P46
LOF1	Register bit	RENV3 5	Release the latch on COUNTER1 that was triggered by an LTC input.	P42, 90
LOF2	Register bit	RENV3 9	Release the latch on COUNTER2 that was triggered by an LTC input.	P42, 90
LTC	Terminal name	42	Latch input (PCL6113)	P10
LTCH	Command	29h	Substitute the LTC input (for counting or latching)	P26, 90
LTCL	Register bit	RENV1 23	Select the trigger edge for the LTC signal (0: Falling edge, 1: Rising edge)	P38, 90
LTCu	Terminal name	137	Latch the input for the U axis (PCL6143)	P10
LTCx	Terminal name	43, 44	Latch the input for the X axis (PCL6123, 6143)	P10
LTCy	Terminal name	80, 75	Latch the input for the Y axis (PCL6123, 6143)	P10
LTCz	Terminal name	106	Latch the input for the Z axis (PCL6143)	P10
MADJ	Register bit	RMD 26	Disable the FH correction function	P37
MAX0 to 3	Register bits	RMD 20-23	Specify the axis used to control stopping for a simultaneous start	P37, 94
MCCE	Register bit	RMD 11	Stop counting output pulses on COUNTER1, 2	P36
MCDE	Register bit	RMD28	Validate the $\overline{\text{CSD}}$ input	P37
MCDO	Register bit	RMD29	Output $\overline{\text{CSD}}$ while selecting the FL speed	P37

Label	Type	Position	Description	Reference
METM	Register bit	RMD 12	Select the operation completion timing (0: Stop at the end of a cycle, 1: Stop on a pulse)	P36
MINP	Register bit	RMD 9	The operation is complete when the INP input turns ON	P36
MOD	Register bits	RMD 0-6	Operation mode selection	P36
MPCS	Register bit	RMD 14	Start control positioning using a PCI input	P36
MSDE	Register bit	RMD 8	Decelerate (decelerate and stop) when the SD input turns ON	P36
MSDP	Register bit	RMD 13	Specify the ramping-down point manually	P36
MSMD	Register bit	RMD 10	S-curve acceleration/deceleration (linear accel/decel when 0)	P36
MSN0 to 1	Register bits	RMD 16-17	Sequence number used to control the operation block	P36
MSPE	Register bit	RMD 24	Enable $\overline{\text{CSTP}}$ input	P37
MSPO	Register bit	RMD 25	Output a $\overline{\text{CSTP}}$ (simultaneous stop) signal when stopped by an error	P37
MSTSB0	Byte map name		Read the main status (bits 0 to 7)	P18
MSTSB1	Byte map name		Read the main status (bits 8 to 15)	P18
MSTSW	Word map name		Read the main status bits (bits 0 to 15)	P19
MSY0 to 1	Register bit	RMD 18-19	Synchronization start timing	P37, 94
$\overline{\text{MVCx}}$	Terminal name	69	Constant speed monitor output for the x axis (PCL6123)	P12
$\overline{\text{MVCy}}$	Terminal name	106	Constant speed monitor output for the y axis (PCL6123)	P12
NOP	Command	00h	(Invalid command)	P25
ORG	Terminal name	38	Zero position signal (PCL6113)	P10
ORGL	Register bit	RENV1 7		P38
ORGu	Terminal name	133	Zero point signal for U axis (PCL6143)	P10
ORGx	Terminal name	39, 40	Zero point signal for X axis (PCL6123, 6143)	P10
ORGy	Terminal name	76, 71	Zero point signal for Y axis (PCL6123, 6143)	P10
ORGz	Terminal name	101	Zero point signal for Z axis (PCL6143)	P10
ORM	Register bit	RENV2 29	Select a zero return method	P41
OTP0 to 7	General-purpose port name	OTPW 0-7	General-purpose port	P30
OTPB	Byte map name		Change status of general output port (valid only for the output specified bits)	P18
OTPW	Word map name		Change status of general output port (valid only for the output specified bits)	P19
OUT	Terminal name	61	Motor driving pulse signal (PCL6113)	P12
OUTu	Terminal name	156	Motor driving pulse signals for U axis (PCL 6143)	P12
OUTx	Terminal name	62, 63	Motor driving pulse signals for X axis (PCL6123, 6143)	P12
OUTy	Terminal name	99, 94	Motor driving pulse signals for Y axis (PCL6123, 6143)	P12
OUTz	Terminal name	125	Motor driving pulse signals for Z axis (PCL 6143)	P12
P0/FUP	Terminal name	51	General-purpose port 0 / Acceleration monitor output (PCL6113)	P11
P0M0 to 1	Register bits	RENV2 0-1	Specify the P0/FUP terminal details	P40
P0RST	Command	10h	Set the general-purpose output port terminal P0 LOW	P26
P0SET	Command	18h	Set the general-purpose output port terminal P0 HIGH	P26
P0u/FUPu	Terminal name	146	General-purpose port 0 for the U axis / Monitor output during acceleration (PCL 6143)	P11
P0x/FUPx	Terminal name	52, 53	General-purpose port 0 for the X axis / Monitor output during acceleration (PCL6123, 6143)	P11
P0y/FUPy	Terminal name	89, 84	General-purpose port 0 for the Y axis / Monitor output during acceleration (PCL6123, 6143)	P11
P0z/FUPz	Terminal name	115	General-purpose port 0 for the Z axis / Monitor output during acceleration (PCL6143)	P11
P1/FDW	Terminal name	52	General-purpose port 1 / Deceleration monitor output (PCL6113)	P11
P1M0 to 1	Register bits	RENV2 2-3	Specify the P1/FDW terminal details	P40
P1RST	Command	11h	Set the general-purpose output port terminal P1 LOW	P26
P1SET	Command	19h	Set the general-purpose output port terminal P1 HIGH	P26
P1u/FDWu	Terminal name	147	General-purpose port 1 for the U axis / Monitor output during acceleration (PCL6143)	P11
P1x/FDWx	Terminal name	53, 54	General-purpose port 1 for the X axis / Monitor output during acceleration (PCL6123,6143)	P11
P1y/FDWy	Terminal name	90, 85	General-purpose port 1 for the Y axis / Monitor output during acceleration (PCL6123, 6143)	P11
P1z/FDWz	Terminal name	116	General-purpose port 1 for the Z axis / Monitor output during acceleration (PCL6143)	P11
P2/MVC	Terminal name	53	General-purpose port 2 / Feeding at a constant speed (PCL6113)	P11
P2M0 to 1	Register bits	RENV2 4-5	Specify the P2/MVC terminal details	P40
P2RST	Command	12h	Set the general-purpose output port terminal P2 LOW	P26
P2SET	Command	1Ah	Set the general-purpose output port terminal P2 HIGH	P26
P2u/MVCu	Terminal name	148	General-purpose port 2 for the U axis / Feeding at low speed (PCL6143)	P12
P2x/MVCx	Terminal name	54, 55	General-purpose port 2 for the X axis / Feeding at low speed (PCL6123,6143)	P12
P2y/MVCy	Terminal name	91, 86	General-purpose port 2 for the Y axis / Feeding at low speed (PCL6123,6143)	P12
P2z/MVCz	Terminal name	117	General-purpose port 2 for the Z axis / Feeding at rated speed (PCL6143)	P12
P3/CP1	Terminal name	54	General-purpose port 3 / Comparator 1 output (PCL6113)	P12
P3M0 to 1	Register bits	RENV2 6-7	Specify the P3/CP1 terminal details	P40
P3RST	Command	13h	Set the general-purpose output port terminal P3 LOW	P26
P3SET	Command	1Bh	Set the general-purpose output port terminal P3 HIGH	P26
P3u/CP1u	Terminal name	149	General-purpose port 3 for the U axis / Comparator 1 output (PCL6143)	P12
P3x/CP1x	Terminal name	55, 56	General-purpose port 3 for the X axis / Comparator 1 output (PCL6123, 6143)	P12
P3y/CP1y	Terminal name	92, 87	General-purpose port 3 for the Y axis / Comparator 1 output (PCL6123, 6143)	P12
P3z/CP1z	Terminal name	118	General-purpose port 3 for the Z axis / Comparator 1 output (PCL6143)	P12
P4/CP	Terminal name	56	General-purpose port 4 / Comparator 2 output (PCL6113)	P12
P4M0 to 1	Register bits	RENV2 8-9	Specify the P4/CP2 terminal details	P40

Label	Type	Position	Description	Reference
P4RST	Command	14h	Set the general-purpose output port terminal P4 LOW	P26
P4SET	Command	1Ch	Set the general-purpose output port terminal P4 HIGH	P26
P4u/CP2u	Terminal name	151	General-purpose port 4 for the U axis / Comparator 2 output (PCL6143)	P12
P4x/CP2x	Terminal name	57, 58	General-purpose port 4 for the X axis / Comparator 2 output (PCL6123, 6143)	P12
P4y/CP2y	Terminal name	94, 89	General-purpose port 4 for the Y axis / Comparator 2 output (PCL6123, 6143)	P12
P4z/CP2z	Terminal name	120	General-purpose port 4 for the Z axis / Comparator 2 output (PCL6143)	P12
P5	Terminal name	57	General-purpose port 5 (PCL6113)	P12
P5M	Register bit	RENV2 10	Specify the P5 terminal function	P40
P5RST	Command	15h	Set the general-purpose output port terminal P5 LOW	P26
P5SET	Command	1Dh	Set the general-purpose output port terminal P5 HIGH	P26
P5u	Terminal name	152	General-purpose port 5 for the U axis. (PCL6143)	P12
P5x	Terminal name	58, 59	General-purpose port 5 for the X axis. (PCL6123, 6143)	P12
P5y	Terminal name	95, 90	General-purpose port 5 for Y axis. (PCL6123, 6143)	P12
P5z	Terminal name	121	General-purpose port 5 for the Z axis. (PCL6143)	P12
P6	Terminal name	58	General-purpose port 6 (PCL6113)	P12
P6M	Register bit	RENV2 11	Specify the P6 terminal function	P40
P6RST	Command	16h	Set the general-purpose output port terminal P6 LOW	P26
P6SET	Command	1Eh	Set the general-purpose output port terminal P6 HIGH	P26
P6u	Terminal name	153	General-purpose port 6 for the U axis (PCL6143)	P12
P6x	Terminal name	59, 60	General-purpose port 6 for the X axis (PCL6123, 6143)	P12
P6y	Terminal name	96, 91	General-purpose port 6 for the Y axis (PCL6123, 6143)	P12
P6z	Terminal name	122	General-purpose port 6 for the Z axis (PCL6143)	P12
P7	Terminal name	59	General-purpose port 7 (PCL6113)	P12
P7M	Register bit	RENV2 12	Specify the P7 terminal function	P40
P7RST	Command	17h	Set the general-purpose output port terminal P7 LOW	P26
P7SET	Command	1Fh	Set the general-purpose output port terminal P7 HIGH	P26
P7u	Terminal name	154	General-purpose port 7 for the U axis (PCL6143)	P12
P7x	Terminal name	60, 61	General-purpose port 7 for the X axis (PCL6123, 6143)	P12
P7y	Terminal name	97, 92	General-purpose port 7 for the Y axis (PCL6123, 6143)	P12
P7z	Terminal name	123	General-purpose port 7 for the Z axis (PCL6143)	P12
PA / +DR	Terminal name	47	Manual pulsar A phase / +DR input (PCL6113)	P11
PAu/ +DRu	Terminal name	141	Manual pulsar phase A input for the U axis (PCL6143)	P11
PAx/ +DRx	Terminal name	48, 49	Manual pulsar phase A input for the X axis (PCL6123, 6143)	P11
PAy / +DRy	Terminal name	85, 80	Manual pulsar phase A input for the Y axis (PCL6123, 6143)	P11
PAz / +DRz	Terminal name	111	Manual pulsar phase A input for the Z axis (PCL6143)	P11
PB/-DR	Terminal name	48	Manual pulsar B phase / -DR input (PCL6113)	P11
PBu/ -DRu	Terminal name	143	Manual pulsar phase B input for the U axis (PCL6143)	P11
PBx/ -DRx	Terminal name	49, 50	Manual pulsar phase B input for the X axis (PCL6123, 6143)	P11
PBy / -DRy	Terminal name	86, 81	Manual pulsar phase B input for the Y axis (PCL6123, 6143)	P11
PBz / -DRz	Terminal name	112	Manual pulsar phase B input for the Z axis (PCL6143)	P11
PCS	Terminal name	40	Start positioning control (PCL6113)	P10
PCSL	Register bit	RENV1 24	Set the input logic for the PCSn signal (0: Negative logic, 1: Positive logic)	P39
PCSM	Register bit	RENV1 30	Change PCS input to self-referenced $\overline{\text{CSTA}}$ signal	P39
PCSu	Terminal name	135	Start positioning control for the U axis (PCL6143)	P10
PCSx	Terminal name	41, 42	Start positioning control for the X axis (PCL6123, 6143)	P10
PCSy	Terminal name	78, 73	Start positioning control for the Y axis (PCL6123, 6143)	P10
PCSz	Terminal name	104	Start positioning control for the Z axis (PCL6143)	P10
PDIR	Register bit	RENV2 23	Reverse the counting direction of the PA and PB inputs	P41
PE	Terminal name	49	Validate PA, PB, +DR, -DR (PCL6113)	P11
PEu	Terminal name	144	Enable the PA, PB, +DR, -DR inputs for U axis (PCL6143)	P11
PEx	Terminal name	50, 51	Enable the PA, PB, +DR, -DR inputs for X axis (PCL6123, 6143)	P11
PEy	Terminal name	87, 82	Enable the PA, PB, +DR, -DR inputs for Y axis (PCL6123, 6143)	P11
PEz	Terminal name	113	Enable the PA, PB, +DR, -DR inputs for Z axis (PCL6143)	P11
PIM0 to 1	Register bits	RENV2 20-21	Specify the PA and PB input details	P41
PINF	Register bit	RENV2 22	Apply a noise filter to the PA/PB inputs	P41
PMG0 to 2	Register bits	RENV1 0-2	Specify the multiplication rate for the PA/PB inputs.	P38
PMSK	Register bit	RENV1 31	Specify the output pulse mask.	P39
POFF	Register bit	RENV2 15	Disable PA, PB inputs.	P40
PRDP	Pre-register name		Pre-register for RDP	P31, 35
PRDR	Pre-register name		Pre-register for RDR	P31, 35
PRDS	Pre-register name		Pre-register for RDS	P31, 38
PRECAN	Command	26h	Cancel the pre-register.	P27, 33
PRFH	Pre-register name		Pre-register for RFH	P31, 34
PRFL	Pre-register name		Pre-register for RFL	P31, 34
PRIP	Pre-register name		Pre-register for RIP	P31, 37
PRMD	Pre-register name		Pre-register for RMD	P31, 36
PRMG	Pre-register name		Pre-register for RMG	P31, 35
PRMV	Pre-register name		Pre-register for RMV	P31, 34
PRUR	Pre-register name		Pre-register for RUR	P31, 34
PRUS	Pre-register name		Pre-register for RUS	P31, 37
RCMP1	Register name		Comparison data for Comparator 1	P31, 43
RCMP2	Register name		Comparison data for Comparator 2	P31, 43
RCUN1	Register name		COUNTER1	P31, 43

Label	Type	Position	Description	Reference
RCUN2	Register name		COUNTER2	P31, 43
$\overline{RD}$	Terminal name	5, 5, 5	Read signal (PCL6113, 6126, 6143)	P7
RDP	Register name		Ramping-down point	P31, 35
RDR	Register name		Deceleration rate	P31, 35
RDS	Register name		S-curve range of deceleration	P31, 38
RENV1	Register name		Environment setting register 1 (Specify the details for the input/output terminals)	P31, 38
RENV2	Register name		Environment setting register 2 (Specify the details for the general-purpose port)	P31, 40
RENV3	Register name		Environment setting register 3 (Specify the details for the counters)	P31, 42
REST	Register name		Error INT status	P31, 46
RFH	Register name		Operation speed	P31, 34
RFL	Register name		Initial speed	P31, 34
RIP	Register name		Master axis feed amount when executing a linear interpolation	P31, 37
RIRQ	Register name		Enable various event interrupts	P31, 44
RIST	Register name		Event INT status	P31, 46
RLTC1	Register name		COUNTER1 latch data	P31, 44
RLTC2	Register name		COUNTER2 latch data	P31, 44
RMD	Register name		Operation mode	P31, 36
RMG	Register name		Speed magnification rate	P31, 35
RMV	Register name		Feed amount or target position	P31, 34
RPLS	Register name		Number of pulses remaining to be fed	P31, 37
RPRDP	Command	C6h	Copy PRDP data to BUF	P29
RPRDR	Command	C4h	Copy PRDR data to BUF	P29
RPRDS	Command	CAh	Copy PRDS data to BUF	P29
RPRFH	Command	C2h	Copy PRFH data to BUF	P29
RPRFL	Command	C1h	Copy PRFL data to BUF	P29
RPRIP	Command	C8h	Copy PRIP data to BUF	P29
RPRMD	Command	C7h	Copy PRMD data to BUF	P29
RPRMG	Command	C5h	Copy PRMG data to BUF	P29
RPRMV	Command	C0h	Copy PRMV data to BUF	P29
RPRUR	Command	C3h	Copy PRUR data to BUF	P29
RPRUS	Command	C9h	Copy PRUS data to BUF	P29
RRCMP1	Command	E7h	Copy RCMP1 data to BUF	P29
RRCMP2	Command	E8h	Copy RCMP2 data to BUF	P29
RRCUN1	Command	E3h	Copy RCUN1 data to BUF	P29
RRCUN2	Command	E4h	Copy RCUN2 data to BUF	P29
RRDP	Command	D6h	Copy RDP data to BUF	P29
RRDR	Command	D4h	Copy RDR data to BUF	P29
RRDS	Command	DAh	Copy RDS data to BUF	P29
RRENV1	Command	DCh	Copy RENV1 data to BUF	P29
RRENV2	Command	DDh	Copy RENV2 data to BUF	P29
RRENV3	Command	DEh	Copy RENV3 data to BUF	P29
RREST	Command	F2h	Copy REST data to BUF	P29
RRFH	Command	D2h	Copy RFH data to BUF	P29
RRFL	Command	D1h	Copy RFL data to BUF	P29
RRIP	Command	D8h	Copy RIP data to BUF	P29
RRIRQ	Command	ECh	Copy RIRQ data to BUF	P29
RRIST	Command	F3h	Copy RIST data to BUF	P29
RRLTC1	Command	EDh	Copy RLTC1 data to BUF	P29
RRLTC2	Command	EEh	Copy RLTC2 data to BUF	P29
RRMD	Command	D7h	Cop RMD data to BUF	P29
RRMG	Command	D5h	Copy RMG data to BUF	P29
RRMV	Command	D0h	Copy RMV data to BUF	P29
RRPLS	Command	F4h	Copy RPLS data to BUF	P29
RRSDC	Command	F6h	Copy RSDC data to BUF	P29
RRSPD	Command	F5h	Copy RSPD data to BUF	P29
RRSTS	Command	F1h	Copy RSTS data to BUF	P29
RRUR	Command	D3h	Copy RUR data to BUF	P29
RRUS	Command	D9h	Copy RUS data to BUF	P29
RSDC	Register name		Automatically calculated value for the ramping-down point	P31, 47
RSPD	Register name		EZ count / Monitor current speed	P31, 47
$\overline{RST}$	Terminal name	79, 127, 175	Reset signal (PCL6113, 6123, 6143)	P7
RSTS	Register name		Extension status	P31, 45
RUR	Register name		Acceleration rate	P31, 34
RUS	Register name		S-curve range during acceleration	P31, 37
SALM	Sub-status bit	SSTSW 11	Equals 1 when the ALM input is ON	P24
SCD	Register bit	RSTS 4	$\overline{CSD}$ input signal is ON	P45
SCP1	Main status bit	MSTSW 8	Equals 1 when the CMP1 comparison conditions are met	P22
SCP2	Main status bit	MSTSW 9	Equals 1 when the CMP2 comparison conditions are met	P22
SD	Terminal name	37	Ramp down signal (PCL6113)	P10
SDIN	Register bit	RSTS 14	The SD input signal is ON (terminal status)	P45
SDIR	Register bit	RSTS 16	Set the operation direction (0: Plus direction, 1: Minus direction)	P45
SDL	Register bit	RENV1 6	Set the input logic of the SD signal (0: Negative logic, 1: Positive logic)	P38

Label	Type	Position	Description	Reference
SDLT	Register bit	RENV1 5	Specify the latch function for the SD input (0: ON, 1: OFF)	P38
SDM	Register bit	RENV1 4	Select the process to execute when the SD input is ON (0: Deceleration only, 1: Decelerate and stop)	P38
SDRM	Register bit	RSTS 12	Equals 1 when the -DR (PB) input signal is ON	P45
SDRP	Register bit	RSTS 11	Equals 1 when the +DR (PA) input signal is ON	P45
SDSTP	Command	4Ah	Deceleration stop	P25
SDu	Terminal name	132	Ramping-down signal for the U axis (PCL6143)	P10
SDx	Terminal name	38, 39	Ramping-down signal for the X axis (PCL6123, 6143)	P10
SDy	Terminal name	75, 70	Ramping-down signal for the Y axis (PCL6123, 6143)	P10
SDz	Terminal name	101	Ramping-down signal for the Z axis (PCL6143)	P10
SELu	Command bit name	COMW 11	Select the U axis	P20
SELx	Command bit name	COMW 8	Select the X axis	P20
SELy	Command bit name	COMW 9	Select the Y axis	P20
SELz	Command bit name	COMW 10	Select the Z axis	P20
SEMG	Register bit	RSTS 7	$\overline{\text{CEMG}}$ Input signal is ON	P45
SEND	Main status bit	MSTSW 3	Equals 0 when started automatically, becomes 1 when stopped	P22
SENI	Main status bit	MSTSW 2	Equals 1 when an interrupt is caused by stopping.	P22
SEOR	Main status bit	MSTSW 13	Equals 1 when unable to execute a position override.	P22
SERC	Register bit	RSTS 9	Equals 1 when the ERC output signal is ON	P45
SERR	Main status bit	MSTSW 4	Equals 1 when an error interrupt occurs	P22
SEZ	Register bit	RSTS 10	Equals 1 when the EZ input signal is ON	P45
SFC	Sub-status bit	SSTSW 10	Equals 1 when feeding at low speed	P23
SFD	Sub-status bit	SSTSW 9	Equals 1 when decelerating	P23
SFU	Sub-status bit	SSTSW 8	Equals 1 when accelerating	P23
SINP	Register bit	RSTS 15	Equals 1 when the INP input signal is ON	P45
SINT	Main status bit	MSTSW 5	Equals 1 when an event interrupt occurs	P22
SLTC	Register bit	RSTS 13	Equals 1 when the LTC input signal is ON	P45
SMEL	Sub-status bit	SSTSW 13	Equals 1 when the -EL input is ON	P23
SORG	Sub-status bit	SSTSW 14	Equals 1 when the ORG input is ON	P23
SPCS	Register bit	RSTS 8	Equals 1 when the PCS input signal is ON	P45
SPEL	Sub-status bit	SSTSW 12	Equals 1 when the +EL input is ON	P23
SPRF	Main status bit	MSTSW 14	Equals 1 when the next-operation pre-register is full	P22
SPSTA	Command	2Ah	The same process as the $\overline{\text{CSTA}}$ input	P25
SRST	Command	04h	Software reset	P27
SRUN	Main status bit	MSTSW 1	Equals 1 while starting	P22
SSC0 to 1	Main status bits	MSTSW 6-7	Sequence code	P22
SSCM	Main status bit	MSTSW 0	Equals 1 when a start command has already been written	P22
SSD	Sub-status bit	SSTSW 15	Equals 1 when the SD input is ON (latched signal)	P23
SSTA	Register bit	RSTS 5	Equals 1 when the $\overline{\text{CSTA}}$ input signal is ON	P45
SSTP	Register bit	RSTS 6	Equals 1 when the $\overline{\text{CSTP}}$ input signal is ON	P45
SSTSB	Byte map name	3 when using a Z80	Used to read the sub status	P23
SSTSW	Word map name	2 when using an 8086	Used to read the sub status, general input/output port	P18
STAD	Command	52h	High speed start 1 (FH low speed -> deceleration stop)	P24
STAFH	Command	51h	Start using FH low speed	P24
STAFL	Command	50h	Start using FL low speed	P24
STAM	Register bit	RENV1 18	Select $\overline{\text{CSTA}}$ signal input specification (0: Level trigger, 1: Edge trigger)	P39
STAON	Command	28h	Substitute for a PCs input	P27
STAUD	Command	53h	High speed start 2 (acceleration -> FH low speed -> deceleration stop)	P24
STOP	Command	49h	Immediate stop	P24
STPM	Register bit	RENV1 19	Select $\overline{\text{CSTP}}$ stop method (0: Immediate stop, 1: Deceleration stop)	P39
SYI0 to 1	Register bits	RENV3 20-21	Select the axis used to input an internal synchronous signal	P42, 96
SYO0 to 3	Register bits	RENV3 16-19	Set the output timing of the internal synchronous signal	P42, 96
WPRDP	Command	86h	Write BUF data into PRDP	P29
WPRDR	Command	84h	Write BUF data into PRDR	P29
WPRDS	Command	8Ah	Write BUF data into PRDS	P29
WPRFH	Command	82h	Write BUF data into PRFH	P29
WPRFL	Command	81h	Write BUF data into PRFL	P29
WPRIP	Command	88h	Write BUF data into PRIP	P29
WPRMD	Command	87h	Write BUF data into PRMD	P29
WPRMG	Command	85h	Write BUF data into PRMG	P29
WPRMV	Command	80h	Write BUF data into PRMV	P29
WPRUR	Command	83h	Write BUF data into PRUR	P29
WPRUS	Command	89h	Write BUF data into PRUS	P29
$\overline{\text{WR}}$	Terminal name	6, 6, 6	Write signal (PCL6113, 6123, 6143)	P8
WRCMP1	Command	A7h	Write BUF data into the RCMP1 register	P29
WRCMP2	Command	A8h	Write BUF data into the RCMP2 register	P29
WRCUN1	Command	A3h	Write BUF data into the RCUN1 register	P29
WRCUN2	Command	A4h	Write BUF data into the RCUN2 register	P29
WRDP	Command	96h	Write BUF data into the RDP register	P29

Label	Type	Position	Description	Reference
WRDR	Command	94h	Write BUF data into the RDR register	P29
WRDS	Command	9Ah	Write BUF data into the RDS register	P29
WRENV1	Command	9Ch	Write BUF data into the RENV1 register	P29
WRENV2	Command	9Dh	Write BUF data into the RENV2 register	P29
WRENV3	Command	9Eh	Write BUF data into the RENV3 register	P29
WRFH	Command	92h	Write BUF data into the RFH register	P29
WRFL	Command	91h	Write BUF data into the RFL register	P29
WRIP	Command	98h	Write BUF data into the RIP register	P29
WRIRQ	Command	ACh	Write BUF data into the RIRQ register	P29
WRMD	Command	97h	Write BUF data into the RMD register	P29
WRMG	Command	95h	Write BUF data into the RMG register	P29
WRMV	Command	90h	Write BUF data into the RMV register	P29
$\overline{\text{WRQ}}$	Terminal name	12, 13, 14	Wait request signal (PCL6113, 6123, 6143)	P8
WRUR	Command	93h	Write BUF data into the RUR register	P29
WRUS	Command	99h	Write BUF data into the RUS register	P29

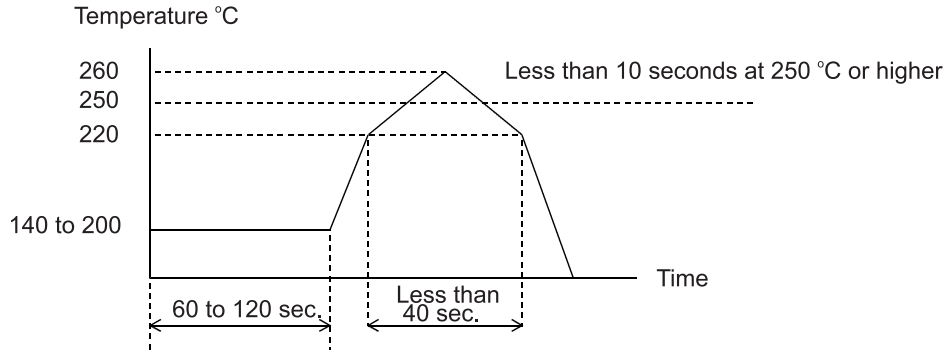
[Handling Precautions]

1. Design precautions
   1) Never exceed the absolute maximum ratings, even for a very short time.
   2) Take precautions against the influence of heat in the environment, and keep the temperature around the LSI as cool as possible.
   3) Please note that ignoring the following may result in latching up and may cause overheating and smoke.
      - Make sure that the voltage on the input/output terminals does not exceed the maximum ratings.
      - Consider power voltage drop timing when turning ON/OFF the power.
      - Be careful not to introduce external noise into the LSI.
      - Hold the unused input terminals to +3.3V or GND level.
      - Do not short-circuit the outputs.
      - Protect the LSI from inductive pulses caused by electrical sources that generate large voltage surges, and take appropriate precautions against static electricity.
   4) Provide external circuit protection components so that overvoltages caused by noise, voltage surges, or static electricity are not fed to the LSI.

2. Precautions for transporting and storing LSIs
   1) Always handle LSIs carefully and keep them in their packages. Throwing or dropping LSIs may damage them.
   2) Do not store LSIs in a location exposed to water droplets or direct sunlight.
   3) Do not store the LSI in a location where corrosive gases are present, or in excessively dusty environments.
   4) Store the LSIs in an anti-static storage container, and make sure that no physical load is placed on the LSIs.

3. Precautions for installation
   1) In order to prevent damage caused by static electricity, pay attention to the following.
      - Make sure to ground all equipment, tools, and jigs that are present at the work site.
      - Ground the work desk surface using a conductive mat or similar apparatus (with an appropriate resistance factor). However, do not allow work on a metal surface, which can cause a rapid change in the electrical charge on the LSI (if the charged LSI touches the surface directly) due to extremely low resistance.
      - When picking up an LSI using a vacuum device, provide anti-static protection using a conductive rubber pick up tip. Anything which contacts the leads should have as high a resistance as possible.
      - When using a pincer that may make contact with the LSI terminals, use an anti-static model. Do not use a metal pincer, if possible.
      - Store unused LSIs in a PC board storage box that is protected against static electricity, and make sure there is adequate clearance between the LSIs. Never directly stack them on each other, as it may cause friction that can develop an electrical charge.
   2) Operators must wear wrist straps which are grounded through approximately 1M-ohm of resistance.
   3) Use low voltage soldering devices and make sure the tips are grounded.
   4) Do not store or use LSIs, or a container filled with LSIs, near high-voltage electrical fields, such those produced by a CRT.
   5) Plastic packages absorb moisture easily. Even if they are stored indoors, they will absorb moisture as time passes.
      If you will be using a soldering method that heats the whole package and you are worried about moisture absorption, dry the packages thoroughly before reflowing the solder.
      Dry the packages for 20 to 36 hours at $125 \pm 5^{\circ}C$. The packages should not be dried more than two times.

6) To heat the entire package for soldering, such as infrared or superheated air reflow, make sure to observe the following conditions and do not reflow more than two times.
   - Temperature profile
     The temperature profile of an infrared reflow furnace must be within the range shown in the figure below. (The temperatures shown are the temperature at the surface of the plastic package.)
   - Maximum temperature
     The maximum allowable temperature at the surface of the plastic package is 260°C peak [A profile]. The temperature must not exceed 250°C [A profile] for more than 10 seconds. In order to decrease the heat stress load on the packages, keep the temperature as low as possible and as short as possible, while maintaining the proper conditions for soldering.



[A profile (applied to lead-free soldering)]

7) Solder dipping causes rapid temperature changes in the packages and may damage the devices. Therefore, do not use this method.

4. Other precautions
   1) When the LSI will be used in poor environments (high humidity, corrosive gases, or excessive amounts of dust), we recommend applying a moisture prevention coating.
   2) The package resin is made of fire-retardant material; however, it can burn. When baked or burned, it may generate gases or fire. Do not use it near ignition sources or flammable objects.
   3) This LSI is designed for use in commercial apparatus (office machines, communication equipment, measuring equipment, and household appliances). If you use it in any device that may require high quality and reliability, or where faults or malfunctions may directly affect human survival or injure humans, such as in nuclear power control devices, aviation devices or spacecraft, traffic signals, fire control, or various types of safety devices, we will not be liable for any problem that occurs, even if it was directly caused by the LSI. Customers must provide their own safety measures to ensure appropriate performance in all circumstances.

NPM NIPPON PULSE MOTOR CO., LTD.

Tokyo Office: No. 16-13, 2-chome, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
Phone: 81-3-3813-8841 Fax: 81-3-3813-7049
E-mail: device@npm.co.jp http://www.pulsemotor.com
U.S. Office: 1073 East Main Street, Radford, VA 24141, U.S.A.
Phone: 1-540-633-1677 Fax: 1-540-633-1674
E-mail: info@nipponpulse.com http://www.nipponpulse.com

MNAL. No. PCL-61XX-1 1B-5205-0.5 (5205) ims