User's Manual
For
PCL6114/6144
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 "Handling Precautions" which include details about installing these ICs, can be found at the end of this manual.

[Cautions]

(1) Copying all or any part of this manual without written approval is prohibited by copyright laws.
(2) The specifications of these LSIs 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 these LSIs, regardless of item (3) above.

- Explanation of the description in this manual

1. "X" "y" "z" and "u" at the foot of terminal names and bit names refer to X axis, Y axis, Z axis and U axis, respectively.
2. Terminals with an overline above the name (ex. $\overline{\text{RST}}$) use negative logic. Their logic cannot be changed. Terminals without an overline use positive logic. Their logic can be changed.
3. When describing the bits in registers, "n" refers to a bit position. "0" refers to a bit position and it is prohibited to be written to any other than "0" and this bit always returns "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 PCL6114 and PCL6144 are CMOS LSIs designed to provide the oscillating, high-speed pulses needed to drive stepper motors and servomotors (pulse train input types) by various commands from CPU.
It can offer various types of control over the pulse train and therefore motor performance. These include continuous operation, positioning, origin return at a constant speed, linear acceleration/deceleration, and S-curve acceleration/deceleration.
The number of control axes is as follows: one for the PCL6114 and four for the PCL6144. They offer linear interpolation of multiple axes (using single or multiple LSIs), check of a LSI'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 LSIs can be operated by single voltage power supply from 3.0 ~ 3.6 V.
The output signal level range is 0 ~ 3.3 V. The input signal level range is 0 ~ 3.3 V, or 0 ~ 5 V.

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

♦ CPU bus I/F
These LSIs have built-in parallel bus I/F (8-bit: 1 type, 16-bit: 3 types) and serial bus I/F(Four-wire synchronous) and they can be connected to a wide variety of CPUs.

♦ 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 by setting S-curve range.
The S-curve range can specify each acceleration and deceleration characteristics independently. Therefore, you can create an acceleration/deceleration profile that consists of linear acceleration and S-curve deceleration, or vice versa.

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

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

♦ Target position override 1) and 2)
1) Target position (feeding amount) can be changed while feeding in positioning mode.
If the current position exceeds a newly entered position, a motor will decelerate and stop (stop immediately when already feeding at a constant speed), and then feed in the reverse direction.
2) Operation starts in the same as a continuous mode. When an external signal is received, the LSI outputs specified number of pulses and a motor will stop.

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

♦ Pre-register function
Next set of data (feeding amount, initial speed, feeding speed, acceleration rate, deceleration rate, speed magnification rate, ramping-down point, operation mode, S-curve range on acceleration, S-curve range on deceleration) can be written while executing current data.

When the current operation is complete, the system will immediately execute the next operation.

♦ Counter circuits

The following two counters are available separately for each axis.

Counter	Purpose of use	Count Input
COUNTER 1	32-bit counter for position control	Output pulses, EA/EB signal input
COUNTER 2	32-bit counter for position control	Output pulses, EA/EB signal input

Both of them can also latch a value by writing a command, or by providing an LTC, or ORG signal. They can also be reset soon after writing a command and latching values.

♦ Comparators

There are 4 comparator circuits for each axis. They can be used to compare target values and internal counter values.
Comparator 1 can be compared with COUNTER 1 and Comparator 2 can be compared with COUNTER 2.
Comparator 3 and 4 are for software limit function only.

♦ Simultaneous start function

Multiple axes controlled by this LSI, or controlled by multiple sets of this LSI, are started to move at the same time by a command or by an external signal.

♦ Simultaneous stop function

All multiple axes controlled by this LSI, or controlled by multiple sets of this LSI, are 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.
Input signals can be 90 degree phase difference signals (1x, 2x, or 4x) or up and down signals.
When an EL signal of a feed direction is input, the LSI stops outputting pulses. However, movement in the opposite direction is available without any commands.

♦ Direct input of external operation switch

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

♦ Operation mode

The basic operations of this LSI are: continuous operation, positioning, origin return, and linear interpolation. By setting bits for optional operation modes, you can use a variety of operations.

<Examples of operation modes>

1) Start/stop by a command.
2) Continuous operation and positioning operation using a manual pulsar.
3) Continuous operation and positioning operation using an external operation switch.
4) Origin return operation.
5) Positioning operation using commands.
6) Start of positioning operation using $\overline{\text{CSTA}}$ input.
7) Feed for a specified amount after turning ON the PCS. (Target position override (2))

♦ Origin return sequences

<Examples of origin return sequences>

1) Feeds at constant speed and stops when the ORG signal is turned ON
2) Feeds at constant speed and stops when the LSI finishes counting specified number of EZ pulses (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, stops when the LSI finishes counting specified number of EZ pulses.

♦ 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, the motor stops immediately (or decelerates and stops). When this signal is ON, no further movement occurs 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 operation, a motor will decelerate to the FL speed. If this signal is ON and then a motor is started, the motor will run at the FL constant 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 will decelerate to the FL speed and then stop.
4) ORG: Input signal for an origin 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 deviation counter clear signal to a servomotor driver.
3) ALM: Regardless of the direction of operation, when this signal is ON, the motor stops immediately (decelerates and stops). When this signal is ON, no movement can occur on this axis.
In timer mode, movement cannot be stopped using the ALM input.
Even though the motor is stopped, the LSI will output an $\overline{INT}$ (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 us ~ 104 ms. A level output is also available.)

♦ Output pulse specification

Common pulse mode, 2-pulse mode or 90-degree phase difference mode can be selected. The logic can also be selected.

♦ Emergency stop signal ($\overline{CEMG}$) input

When this signal is turned ON, movement on all axes stops immediately. While this signal is ON, no movement is allowed on any axes.
This input cannot be disabled. Even in timer mode, movement stops.

♦ Interrupt signal output

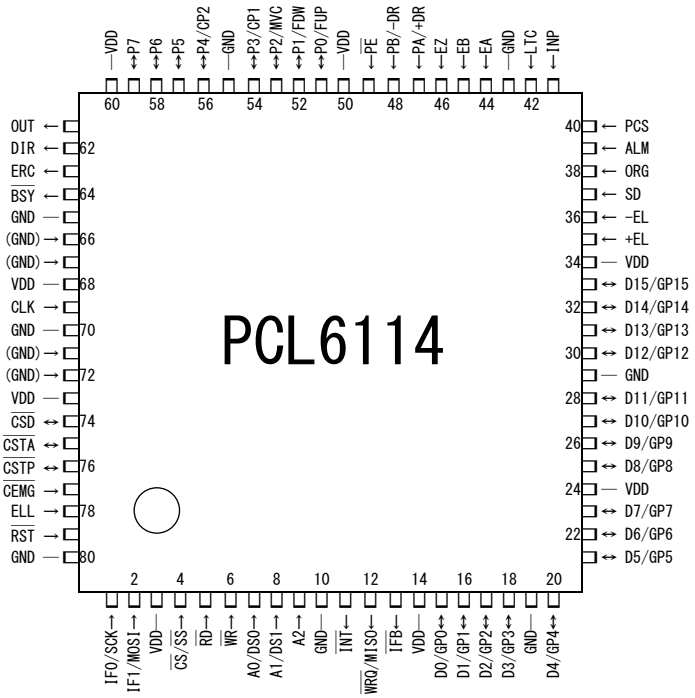
An $\overline{INT}$ signal (interrupt request) can be output for many factors.
The LSI outputs a signal from $\overline{INT}$ terminal with ORed logic of various factors on each axis.
(When more than one LSI is used, wired OR connections are invalid. ($\overline{INT} \neq$ Hi-z))

2. Specification

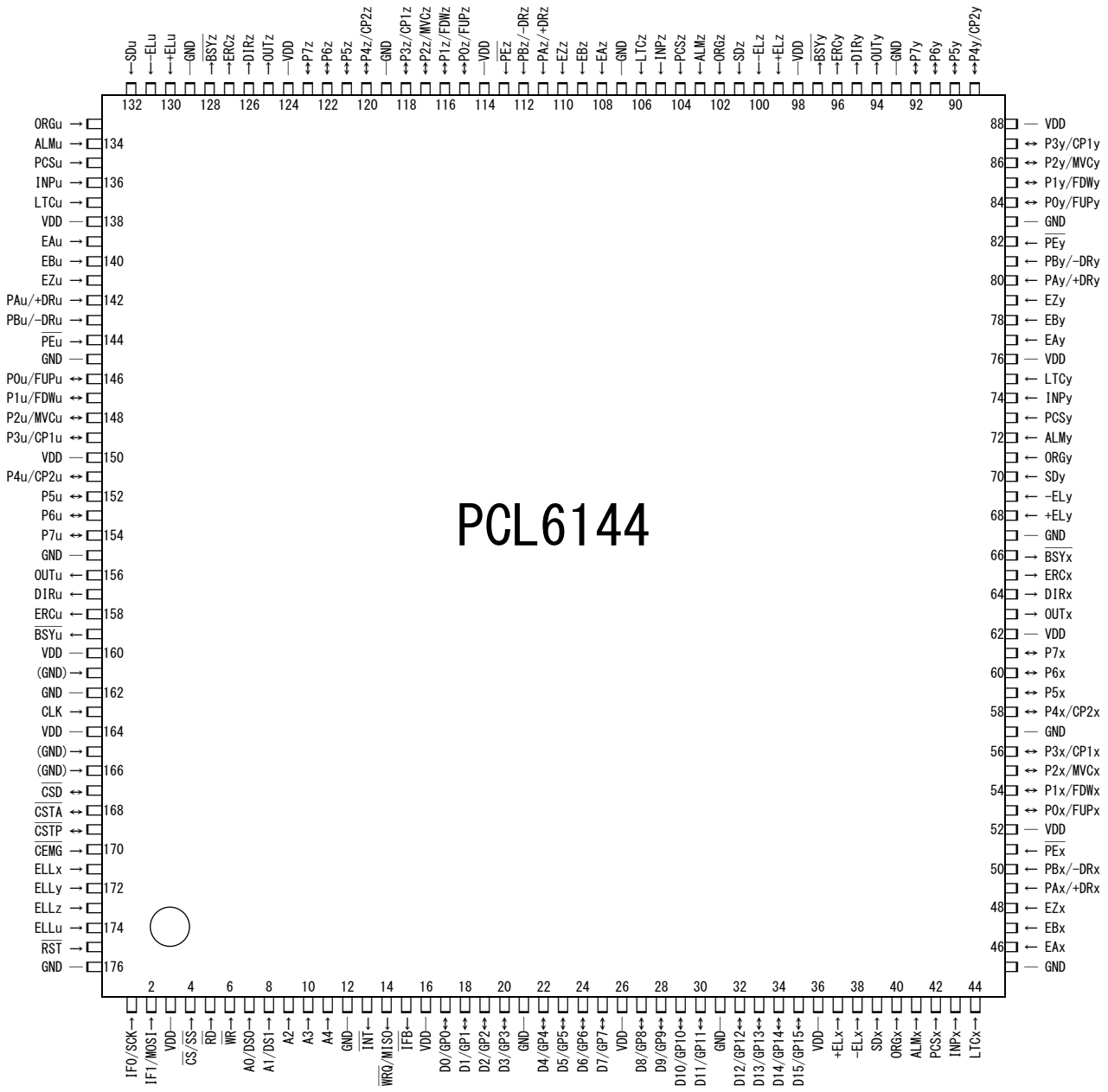
Item	Description
Number of control axes	PCL6114: One PCL6144: Four (X, Y, Z, and U axes)
Reference clock	Standard: 19.6608 MHz (Max. 30 MHz)
Positioning control range	-2,147,483,648 ~ +2,147,483,647 (32-bit)
Ramping-down point setting range	0 ~ 16,777,215 (24-bit)
Number of registers used for setting speeds	Two for each axis (FL and FH)
Speed setting step range	1 ~ 16,383 (14-bit)
Speed magnification range	Change of reference clock varies speed range at the rate. 1. When reference clock = 19.6608 MHz, 0.293x ~ 600x. (The following is an example.) When 0.3x is selected: 0.3 ~ 4,914.9 pps When 1x is selected :1 ~ 16,383 pps When 600x is selected: 600 ~ 9,829,800 pps 2. When reference clock =30 MHz, 0.447x ~ 915.527x. (The following is an example.) When 0.5x is selected: 0.5 ~ 8,191.7 pps When 1x is selected: 1 ~ 16.383.5 pps When 915.527x is selected: 915.527 ~ 14.999.084.5 pps
Acceleration/deceleration characteristics	Linear and S-curve acceleration/deceleration. Acceleration and deceleration characteristics can be specified independently.
Acceleration rate setting range	1 ~ 65,535 (16-bit)
Deceleration rate setting range	1 ~ 65,535 (16-bit)
Ramping-down point automatic setting	The automatic setting is available only when acceleration and deceleration curves are symmetrical.
Feed speed automatic correction function	Automatically lowers the feeding speed for short distance positioning is lowered.
Manual operation input	Manual pulsar (PA/PB) input, external switch (+DR/-DR)input
Counter	COUNTER 1: Position control counter (32-bit) COUNTER 2: Position control counter (32-bit)
Comparators	General purpose comparators : 32-bit x 2 circuits / axis Soft limit only : 32-bit x 2 circuits / axis
CPU bus I/F	8-bit parallel bus I/F 16-bit parallel bus I/F Serial bus I/F (SCK≦CLK/1.5)
Interpolation functions	Linear interpolation: Any 2 ~ 4 axes
Operating temperature range	-40 ~ +85 °C
Power supply	3.0 ~ 3.6 V
Package	PCL6114: 80-pin QFP (External dimensions 14 x 14 mm) PCL6144: 176-pin QFP (External dimensions 26 x 26 mm)

3. Terminal Assignment Diagram

3-1. PCL6114



3-2. PCL6144



Note: On actual products, a mark similar to an indexing mark (mark O) 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. Identify the 1st terminal by the position of the mark O.

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 "Direction" 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 signal logic. "P" means positive logic and “N” means negative logic. “#” means changeable with software. "%" means a hardware setting.
Note 4: The "Handling" column describes how to deal with terminals when they are not used. (Some terminals need to be handled 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.

(1) PCL6114

Signal name	Terminal No.	Direc -tion	Logic	Hand -ling	Description
GND	10, 19, 29, 43, 55, 65, 70, 80	IN	-	-	Connects to GND. Make sure to connect all of these terminals.
VDD	3, 14, 24, 34, 50, 60, 68, 73	IN	-	-	Connects to 3.3V (+3.0 ~ 3.6V). Make sure to connect all of these terminals.
$\overline{RST}$	79	IN	N	-	Reset signal ($\overline{RST}$) input Make sure to set this signal=L level at least once after turning ON the power and before starting operation. Input and holding $\overline{RST}$=L level for at least 8 cycles of the reference clock. For details about status after reset, see section “11-1. Reset".
CLK	69	IN	-	-	As standard, input a 19, 6608 MHz reference clock signal (CLK). The LSI creates output pulses based on the clock input to this terminal.
IF0/SCK IF1/MOSI	1 2	IN	-	-	Set CPU bus I/F mode with parallel bus I/F. Please see “6-2. Parallel bus I/F” for details. With serial bus I/F, these are input terminals of serial clock (SCK) and write data (MOSI).
$\overline{CS}$/ $\overline{SS}$	4	IN	N	-	With parallel bus I/F, and, $\overline{RD}$ and $\overline{WR}$ terminals are enabled when $\overline{CS}$ terminal is L level. With serial bus I/F, this is an input terminal of slave select signal ($\overline{SS}$).
$\overline{RD}$ $\overline{WR}$	5 6	IN	N	-	When $\overline{RD}$ and $\overline{WR}$ terminals are L level at the rising of reset signal, serial bus I/F mode is selected. In other case, parallel bus I/F mode is selected. With serial bus I/F, these are input terminals of read signal ($\overline{RD}$) and write signal ($\overline{WR}$).
A0/DS0 A1/DS1 A2	7 8 9	IN	P	-	With parallel bus I/F, inputs address bus (A0 ~ A2). With serial bus I/F, sets device select number (DS0 and DS1). According to setting of device select number, one slave select signal ($\overline{SS}$) is extended to select up to 4 LSIs.
$\overline{INT}$	11	OUT	N	OP	Outputs interrupt request signal ($\overline{INT}$) to a CPU. There are 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. Error interrupt is determined by reading REST register and event interrupt is determined by reading RIST register.
$\overline{WRQ}$/MISO	12	OUT	N	OP	With parallel bus I/F, outputs negative logic wait request signal ($\overline{WRQ}$) to CPU. With serial bus I/F, outputs positive logic read data (MISO).

Signal name	Terminal No.	Direc -tion	Logic	Hand -ling	Description
$\overline{IFB}$	13	OUT	N	OP	Outputs signal ($\overline{IFB}$) used to indicate that the LSI is processing commands. If you connect with a CPU that does not have a wait control input terminal, make sure that this terminal is H level before you access next. With parallel bus I/F, please make it open.
D0/GP0 D1/GP1 D2/GP2 D3/GP3 D4/GP4 D5/GP5 D6/GP6 D7/GP7	15 16 17 18 20 21 22 23	I/O	P	PU or PD	With parallel bus I/F, connect the lower 8-bit data bus (D0~D7) when connecting a 16-bit data bus. Connects all data bus with connecting 8-bit data bus. With serial bus I/F, these terminals are general I/O ports (GP0~GP7). When these terminals are not used, pull-up or pull down these terminals.
D8/GP8 D9/GP9 D10/GP10 D11/GP11 D12/GP12 D13/GP13 D14/GP14 D15/GP15	25 26 27 28 30 31 32 33	I/O	P	PU or PD	With parallel bus I/F, connect the upper 8-bit data bus (D8~D15) when connecting a 16-bit data bus. With serial bus I/F, these terminals are general I/O ports (GP8~GP15). When 8 bit data bus is used or these terminals are not used, pull-up or pull down these terminals.
$\overline{CSD}$	74	I/O	N	PU	Inputs and outputs simultaneous deceleration signals. When performing multiple axis control using more than one LSI to decelerate movement on these axes at the same time, connect all of the $\overline{CSD}$ terminals to each other. Even when using this signal, it should be pulled up to VDD. The terminal status can be checked on RSTS register.
$\overline{CSTA}$	75	I/O	N	PU	Inputs and outputs simultaneous starts. When performing multiple axis control using more than one LSI to start movement on these axes at the same time, connect all of the $\overline{CSTA}$ terminals to each other. Even when using this signal, it should be pulled up to VDD. The terminal status can be checked on RSTS register.
$\overline{CSTP}$	76	I/O	N	PU	Inputs and outputs terminal for simultaneous stops. When performing multiaxis control using more than one LSI to stop movement on these axes at the same time, connect all of the $\overline{CSTP}$ terminals to each other. Even when using this signal, it should be pulled up to VDD. The terminal status can be checked on RSTS register.
$\overline{CEMG}$	77	IN	N	+V	Inputs emergency stop signals. While $\overline{CEMG}$ terminal is L level, a motor does not start. If $\overline{CEMG}$ terminal turns to L level while in operation, all axes will stop immediately. The terminal status can be checked on RSTS register.
ELL	78	IN	-	-	Sets input logic for +EL signals and –EL signals L level: Input logic of +EL and –EL is positive. H level: Input logic of +EL and –EL is negative.
+EL	35	IN	H	+V	Inputs end limit signal input in the positive (+) direction. When this terminal is ON while feeding in the positive (+) direction, movement stops immediately or decelerates and stops. Input logic is specified using the ELL terminal. The terminal status can be checked using sub status.
-EL	36	IN	N%	+V	Inputs end limit signal input in the negative (-) direction. When this terminal is ON while feeding in negative (-) direction, movement stops immediately, or decelerates and stops. Input logic is specified using the ELL terminal. The terminal status can be checked using sub status.

Signal name	Terminal No.	Direc -tion	Logic	Hand -ling	Description
SD	37	IN	N#	+V	Inputs deceleration (deceleration stop) signal. Select input method between Level and latched inputs. Input logic can be selected using software. The terminal status can be checked using sub status.
ORG	38	IN	N#	+V	Inputs origin position signal input Used for origin return. (Edge detection) Input logic can be selected using software. The terminal status can be checked using sub status.
ALM	39	IN	N#	+V	Inputs alarm signal. When this signal is ON, movement of the axis stops immediately, or decelerates and stops. Input logic can be selected using software. The terminal status can be checked using sub status.
PCS	40	IN	N#	GN	The LSI starts positioning when this signal turns ON. (Target position override 2) Input logic can be changed using software. The terminal status can be checked using the RSTS register.
INP	41	IN	N#	GN	Inputs positioning complete signal input from servo driver (in-position signal) The input logic can be changed using software. The terminal status can be checked using the RSTS register.
LTC	42	IN	N#	GN	Latches counter value of COUNTER 1 and COUNTER 2 The input logic can be changed using software. The terminal status can be checked using the RSTS register.
EA	44	IN	-	GN	Input phase A or (+) direction encoder signals. When inputting 90-degree phase difference signals and the EA signal phase is ahead of the EB signal, the LSI counts up pulses. The counting direction can be changed using software.
EB	45	IN	-	GN	Input phase B or (-) direction encoder signals. When inputting 90-degree phase difference signals and the EB signal phase is ahead of the EA signal, the LSI counts up pulses. The counting direction can be changed using software.
EZ	46	IN	N#	GN	Inputs an EZ signal used in origin return mode. EZ signal is a market that is output once for each turn of the encoder). Use of the EZ signal improves origin return precision. Input logic can be changed by using software. The terminal status can be checked by using the RSTS register.
PA/+DR	47	IN	-	GN	Inputs phase A or (-) direction manual pulsar signals (PA) or (+) direction external switch. Whether this input signal is enabled or disabled is selected by $\overline{PE}$ terminal. Terminal function and operation direction are selected by software.
PB/-DR	48	IN	-	GN	Inputs phase B or (+) direction manual pulsar signals (PB) or (-) direction external switch. Whether this input signal is enabled or disabled is selected by $\overline{PE}$ terminal. Terminal function and operation direction are selected by software.
$\overline{PE}$	49	IN	N	GN	Inputs signal ($\overline{PE}$) to enable or disable PA/+DR terminal and PB/-DR terminal. L level: PA/+DR terminal and PB/-DR terminal are enabled. H level: PA/+DR terminal and PB/-DR terminal are disabled.
P0/FUP	51	I/O	-	PD	Common terminal used for general purpose port (P0) or acceleration monitor output (FUP). Terminal function and output logic can be set using software.
P1/FDW	52	I/O	-	PD	Common terminal used for general purpose port (P1) or deceleration monitor output (FDW). Terminal function and output logic can be set using software.
P2/MVC	53	I/O	-	PD	Common terminal used for general purpose port (P2) or constant speed monitor output (MVC). Terminal function and output logic can be set using software.

Signal name	Terminal No.	Direc -tion	Logic	Hand -ling	Description
P3/CP1	54	I/O	-	PD	Common terminal used for general purpose port (P3) or comparator 1 output (CP1). Terminal function and output logic can be set using software.
P4/CP2	56	I/O	-	PD	Common terminal used for general purpose port (P4) or comparator 2 output (CP2). Terminal function and output logic can be set using software.
P5	57	I/O	-	PD	Terminal for general-purpose I/O port (P5). Terminal function is selected by software.
P6	58	I/O	-	PD	Terminal for general-purpose I/O port (P6). Terminal function is selected by software.
P7	59	I/O	-	PD	Terminal for general-purpose I/O port (P7). Terminal function is selected by software.
OUT	61	OUT	N#	OP	Outputs pulse train for controlling a motor (command pulse). Output specifications are selected by software.
DIR	62	OUT	N#	OP	Outputs pulse train for controlling a motor (command pulse). Output specifications are selected by software.
ERC	63	OUT	N#	OP	Outputs deviation counter clear signal of a servo driver. Pulse width is changed and output logic is set by software. Terminal status can be checked using RSTS register.
$\overline{\text{BSY}}$	64	OUT	N	OP	Terminal for operations status monitor output ($\overline{\text{BSY}}$). It is L level while operating.
(GND)	66, 67, 71, 72	IN	-	GN	Input terminal for delivery inspection. Connect it to GND.

(2) PCL6144

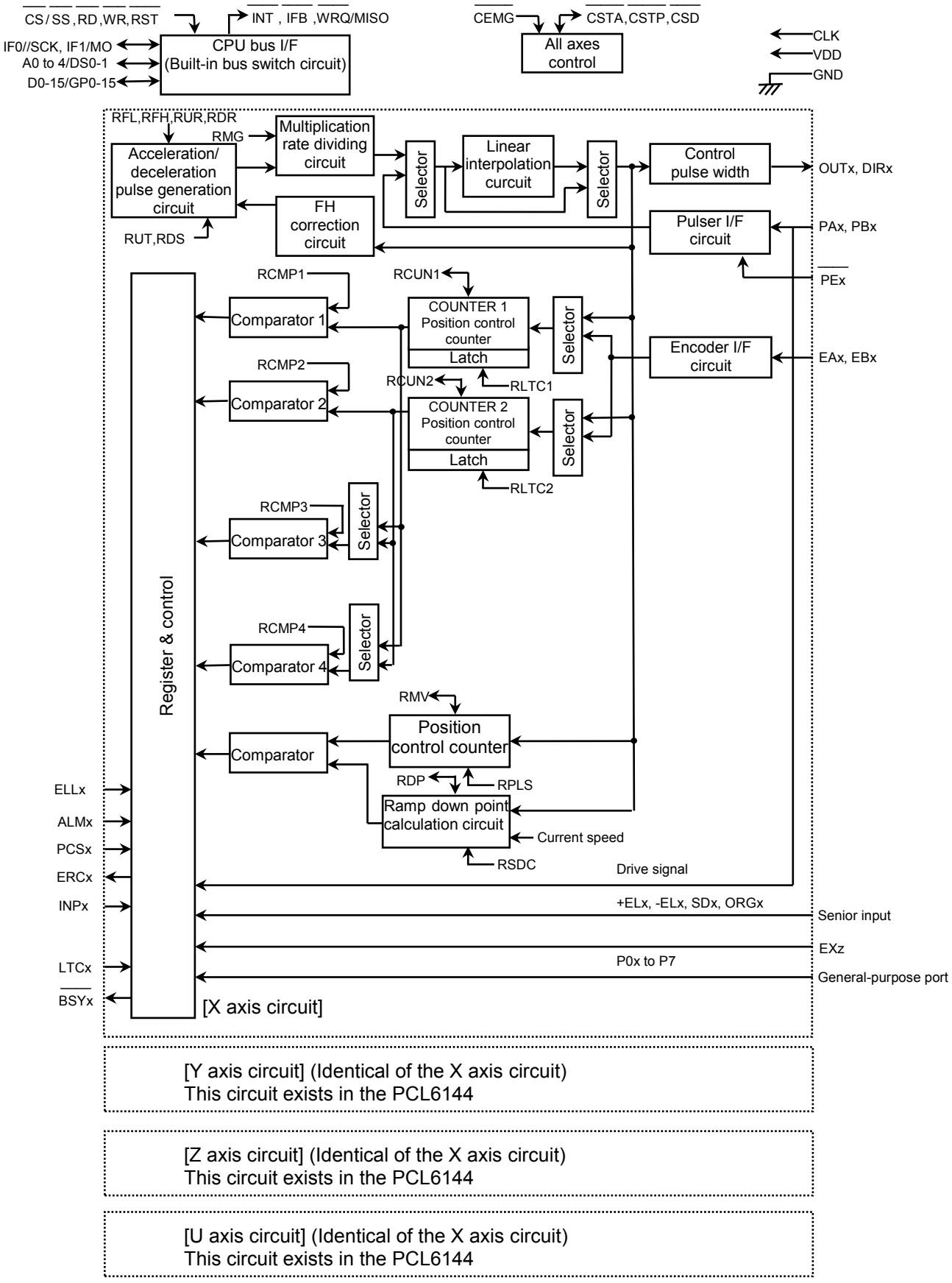
Signal name	Terminal No.	Direc -tion	Logic	Hand -ling	Description
GND	12, 21, 31, 45, 57, 67, 83, 93, 107, 119, 129, 145, 155, 162, 176	IN	-	-	Connects to GND. Make sure to connect all of these terminals.
VDD	3, 16, 26, 36, 52, 62, 76, 88, 98, 114, 124, 138, 150, 160, 164	IN	-	-	Connects to 3.3V (+3.0 ~ 3.6V). Make sure to connect all of these terminals.
$\overline{RST}$	175	IN	N	-	Reset signal ($\overline{RST}$) input Make sure to set this signal=L level at least once after turning ON the power and before starting operation. Input and holding $\overline{RST}$=L level for at least 8 cycles of the reference clock. For details about status after reset, see section “11-1. Reset".
CLK	163	IN	-	-	As standard, input a 19, 6608 MHz reference clock signal (CLK). The LSI creates output pulses based on the clock input to this terminal.
IF0/SCK IF1/MOSI	1 2	IN	-	-	Set CPU bus I/F mode with parallel bus I/F. Please see “6-2. Parallel bus I/F” for details. With serial bus I/F, these are input terminals of serial clock (SCK) and write data (MOSI).
$\overline{CS}$ / $\overline{SS}$	4	IN	N	-	With parallel bus I/F, and, $\overline{RD}$ and $\overline{WR}$ terminals are enabled when $\overline{CS}$ terminal is L level. With serial bus I/F, this is an input terminal of slave select signal ($\overline{SS}$).
$\overline{RD}$ $\overline{WR}$	5 6	IN	N	-	When $\overline{RD}$ and $\overline{WR}$ terminals are L level at the rising of reset signal, serial bus I/F mode is selected. In other case, parallel bus I/F mode is selected. With serial bus I/F, these are input terminals of read signal ($\overline{RD}$) and write signal ($\overline{WR}$).
A0/DS0 A1/DS1 A2 A3 A4	7 8 9 10 11	IN	P	-	With parallel bus I/F, inputs address bus (A0~A2). With serial bus I/F, sets device select number (DS0 and DS1). According to setting of device select number, one slave select signal ($\overline{SS}$) is extended to select up to 4 LSIs.
$\overline{INT}$	13	OUT	N	OP	Outputs interrupt request signal ($\overline{INT}$) to a CPU. There are 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. Error interrupt is determined by reading REST register and event interrupt is determined by reading RIST register.
$\overline{WRQ}$/MISO	14	OUT	N	OP	With parallel bus I/F, outputs negative logic wait request signal ($\overline{WRQ}$) to CPU. With serial bus I/F, outputs positive logic read data (MISO).
$\overline{IFB}$	15	OUT	N	OP	Outputs signal ($\overline{IFB}$) used to indicate that the LSI is processing commands. If you connect with a CPU that does not have a wait control input terminal, make sure that this terminal is H level before you access next. With parallel bus I/F, please make it open.
D0/GP0 D1/GP1 D2/GP2 D3/GP3 D4/GP4 D5/GP5 D6/GP6 D7/GP7	17 18 19 20 22 23 24 25	I/O	P	PU or PD	With parallel bus I/F, connect the lower 8-bit data bus (D0 ~D7) when connecting a 16-bit data bus. Connects all data bus with connecting 8-bit data bus. With serial bus I/F, these terminals are general I/O ports (GP0~GP7). When these terminals are not used, pull-up or pull down these terminals.

Signal name	Terminal No.	Direc -tion	Logic	Hand -ling	Description
D8/GP8 D9/GP9 D10/GP10 D11/GP11 D12/GP12 D13/GP13 D14/GP14 D15/GP15	27 28 29 30 32 33 34 35	I/O	P	PU or PD	With parallel bus I/F, connect the upper 8-bit data bus (D8 ~D15) when connecting a 16-bit data bus. With serial bus I/F, these terminals are general I/O ports (GP8~GP15). When 8 bit data bus is used or these terminals are not used, pull-up or pull down these terminals.
$\overline{CSD}$	167	I/O	N	PU	Inputs and outputs simultaneous deceleration signals. When performing multiple axis control using more than one LSI to decelerate movement on these axes at the same time, connect all of the $\overline{CSD}$ terminals to each other. Even when using this signal, it should be pulled up to VDD. The terminal status can be checked on RSTS register.
$\overline{CSTA}$	168	I/O	N	PU	Inputs and outputs simultaneous starts. When performing multiple axis control using more than one LSI to start movement on these axes at the same time, connect all of the $\overline{CSTA}$ terminals to each other. Even when using this signal, it should be pulled up to VDD. The terminal status can be checked on RSTS register.
$\overline{CSTP}$	169	I/O	N	PU	Inputs and outputs terminal for simultaneous stops. When performing multiaxis control using more than one LSI to stop movement on these axes at the same time, connect all of the $\overline{CSTP}$ terminals to each other. Even when using this signal, it should be pulled up to VDD. The terminal status can be checked on RSTS register.
$\overline{CEMG}$	170	IN	N	+V	Inputs emergency stop signals. While $\overline{CEMG}$ terminal is L level, a motor does not start. If $\overline{CEMG}$ terminal turns to L level while in operation, all axes will stop immediately. The terminal status can be checked on RSTS register.
ELLn	X : 171 Y : 172 Z : 173 U : 174	IN	-	-	Sets input logic for +EL signals and –EL signals L level: Input logic of +EL and –EL is positive. H level: Input logic of +EL and –EL is negative.
+ELn	X : 37 Y : 68 Z : 99 U : 130	IN	N%	+V	Inputs end limit signal input in the positive (+) direction. When this terminal is ON while feeding in the positive (+) direction, movement stops immediately or decelerates and stops. Input logic is specified using the ELL terminal. The terminal status can be checked using sub status.
-ELn	X : 38 Y : 69 Z : 100 U : 131	IN	N%	+V	Inputs end limit signal input in the negative (-) direction. When this terminal is ON while feeding in negative (-) direction, movement stops immediately, or decelerates and stops. Input logic is specified using the ELL terminal. The terminal status can be checked using sub status.
SDn	X : 39 Y : 70 Z : 101 U : 132	IN	N#	+V	Input deceleration (deceleration stop) signal Selects the input method: Level or Latched inputs. The input logic can be selected using software. The terminal status can be checked using sub status.
ORGn	X : 40 Y : 71 Z : 102 U : 133	IN	N#	+V	Origin position signal input Used for origin return. (Edge detection.) The input logic can be selected using software. The terminal status can be checked using sub status.
ALMn	X : 41 Y : 72 Z : 103 U : 134	IN	N#	+V	Alarm signal input 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 sub status.

Signal name	Terminal No.	Direc -tion	Logic	Hand -ling	Description
PCSn	X : 42 Y : 73 Z : 104 U : 135	IN	N#	GN	The LSI will start positioning when this signal turns ON. (Target position override 2) The input logic can be changed using software. The terminal status can be checked using the RSTS register.
INPn	X : 43 Y : 74 Z : 105 U : 136	IN	N#	GN	Inputs positioning complete signal input from servo driver (in-position signal) The input logic can be changed using software. The terminal status can be checked using the RSTS register.
LTCn	X : 44 Y : 75 Z : 106 U : 137	IN	N#	GN	Latches counter value of COUNTER 1 and COUNTER 2. The input logic can be changed using software. The terminal status can be checked using the RSTS register.
EAn	X : 46 Y : 77 Z : 108 U : 139	IN	-	GN	Input phase A or (+) direction encoder signals. When inputting 90-degree phase difference signals and the EA signal phase is ahead of the EB signal, the LSI counts up pulses. The counting direction can be changed using software.
EBn	X : 47 Y : 78 Z : 109 U : 140	IN	-	GN	Input phase B or (-) direction encoder signals. When inputting 90-degree phase difference signals and the EB signal phase is ahead of the EA signal, the LSI counts up pulses. The counting direction can be changed using software.
EZn	X : 48 Y : 79 Z : 110 U : 141	IN	N#	GN	Inputs an EZ signal used in origin return mode. EZ signal is a market that is output once for each turn of the encoder). Use of the EZ signal improves origin return precision. Input logic can be changed by using software. The terminal status can be checked by using the RSTS register.
PAn/+DRn	X : 49 Y : 80 Z : 111 U : 142	IN	-	GN	Inputs phase A or (-) direction manual pulsar signals (PA) or (+) direction external switch. Whether this input signal is enabled or disabled is selected by $\overline{PE}$ terminal. Terminal function and operation direction are selected by software.
PBn/-DRn	X : 50 Y : 81 Z : 112 U : 143	IN	-	GN	Inputs phase B or (+) direction manual pulsar signals (PB) or (-) direction external switch. Whether this input signal is enabled or disabled is selected by $\overline{PE}$ terminal. Terminal function and operation direction are selected by software.
$\overline{PE}$n	X : 51 Y : 82 Z : 113 U : 144	IN	N	GN	Inputs signal ($\overline{PE}$) to enable or disable PA/+DR terminal and PB/-DR terminal. L level: PA/+DR terminal and PB/-DR terminal are enabled. H level: PA/+DR terminal and PB/-DR terminal are disabled.
P0n/FUPn	X : 53 Y : 84 Z : 115 U : 146	I/O	-	PD	Common terminal used for general purpose port (P0) or acceleration monitor output (FUP). Terminal function and output logic can be set using software.
P1n/FDWn	X : 54 Y : 85 Z : 116 U : 147	I/O	-	PD	Common terminal used for general purpose port (P1) or deceleration monitor output (FDW). Terminal function and output logic can be set using software.
P2n/MVCn	X : 55 Y : 86 Z : 117 U : 148	I/O	-	PD	Common terminal used for general purpose port (P2) or constant speed monitor output (MVC). Terminal function and output logic can be set using software.
P3n/CP1n	X : 56 Y : 87 Z : 118 U : 149	I/O	-	PD	Common terminal used for general purpose port (P3) or comparator 1 output (CP1). Terminal function and output logic can be set using software.

Signal name	Terminal No.	IN/ OUT	Logic	Hand -ling	Description
P4n/CP2n	X : 58 Y : 89 Z : 120 U : 151	I/O	-	PD	Common terminal used for general purpose port (P4) or comparator 2 output (CP2). Terminal function and output logic can be set using software.
P5n	X : 59 Y : 90 Z : 121 U : 152	I/O	-	PD	Terminal for general-purpose I/O port (P5). Terminal function is selected by software.
P6n	X : 60 Y : 91 Z : 122 U : 153	I/O	-	PD	Terminal for general-purpose I/O port (P6). Terminal function is selected by software.
P7n	X : 61 Y : 92 Z : 123 U : 154	I/O	-	PD	Terminal for general-purpose I/O port (P7). Terminal function is selected by software.
OUTn	X : 63 Y : 94 Z : 125 U : 156	OUT	N#	OP	Outputs pulse train for controlling a motor (command pulse). Output specifications are selected by software.
DIRn	X : 64 Y : 95 Z : 126 U : 157	OUT	N#	OP	Outputs pulse train for controlling a motor (command pulse). Output specifications are selected by software.
ERCn	X : 65 Y : 96 Z : 127 U : 158	OUT	N#	OP	Outputs deviation counter clear signal of a servo driver. Pulse width is changed and output logic is set by software. Terminal status can be checked using RSTS register.
$\overline{\text{BSY}}$n	X : 66 Y : 97 Z : 128 U : 159	OUT	N	OP	Terminal for operations status monitor output ($\overline{\text{BSY}}$). It is L level while operating.
(GND)	161, 165, 166	IN	-	GN	Input terminal for delivery inspection. Connect it to GND.

5. Block Diagram



6. CPU bus I/F

This LSI contains that 4 types of parallel bus I/F circuits and 1 type of serial bus I/F circuit as CPU bus I/F circuits in order to facilitate connection to various CPUs.

6-1. Hardware design precautions

- All of the input terminals can handle 0 ~ +5 V levels.
- Although all of the output terminals can be pulled up to +5 V, they cannot output 3.3 V or more. Recommended resistance value ranges more than 5 ohm of pull-up resistance.
- Any unused terminals among P0~P7 should be pulled down to GND externally or pulled up to VDD. Recommend resistance value ranges from 5k ~ 10k ohm.
- When connecting a CPU with an 8-bit bus I/F, D8~D15 should be pulled down to GND external yor pulled up to VDD. Recommended resistance value ranges from 5k ~ 10k ohm.
- Input logic of +EL and –EL signal can be changed by ELL terminals.
- When CPU is connected by serial bus I/F, any unused terminals among P0~P15 should be pulled down to GND externally or pulled up to VDD. Recommend resistance value ranges from 5k ~ 10k ohm.

6-2. Parallel bus I/F

6-2-1. Setting of connected CPU

To select parallel bus I/F circuits, use the IF0 and IF1 terminals.
Shown below are some circuit examples. To use some other CPUs, select an appropriate interface after referring to section "12-4. AC characteristics".

[Example of connecting CPU signals with parallel bus I/F]

Setting status		Interface Name	CPU type	CPU signal to connect to the 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 an R/$\overline{W}$ 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.

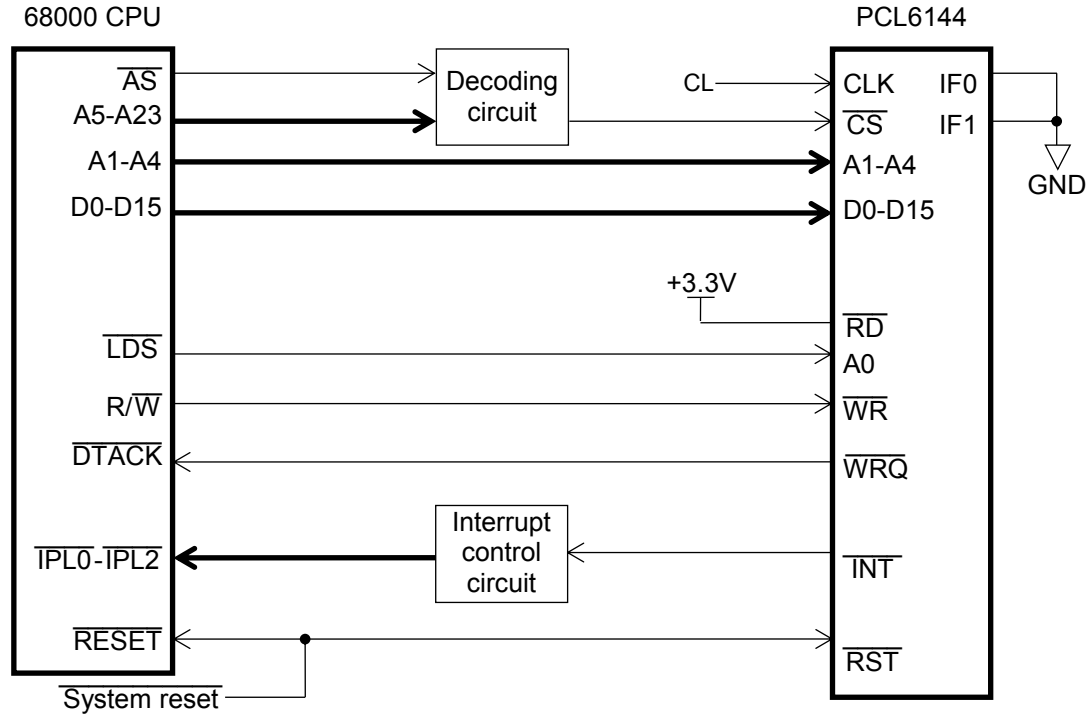
16-bit I/F-2: A 16-bit interface with an RD input and a WR input.
The lower addresses correspond to the upper word in the I/O buffer.
Convenient for H8 series CPUs.

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.

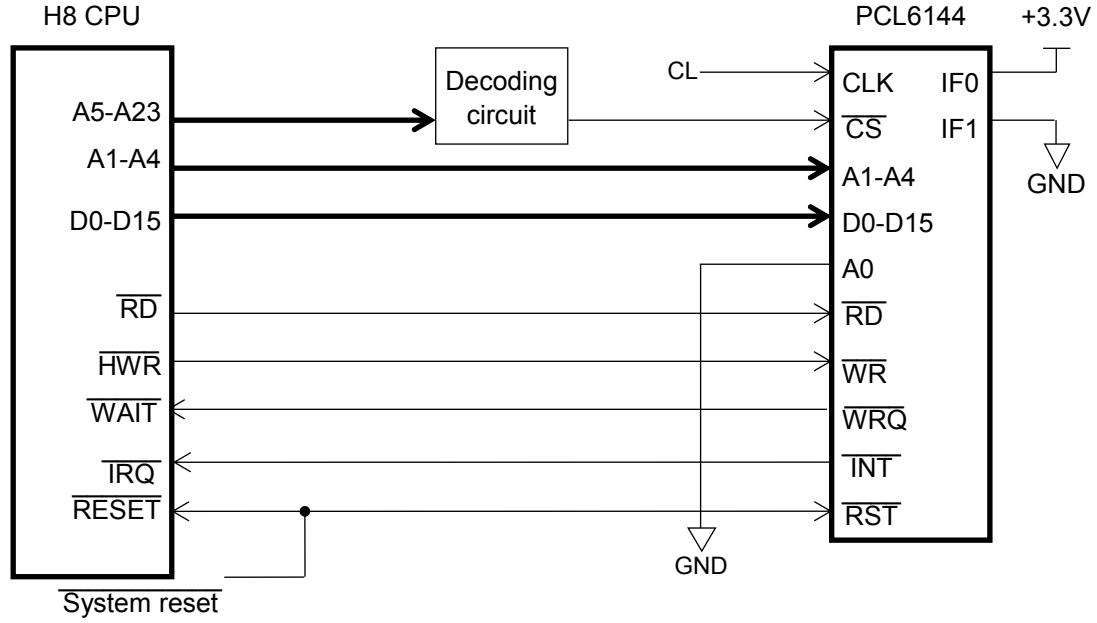
6-2-2. Examples of CPU bus I/F

(1) 16-bit parallel bus I/F-1
Setting to select CPU bus I/F : IF1=L, IF0=L



Note: In PCL6144, A0 and A1~A4 are used. In PCL6114, A0 and A1~A2 are used.
16 bit I/F can be accessed by word (16-bit) only. It is not accessed by byte (8-bit).

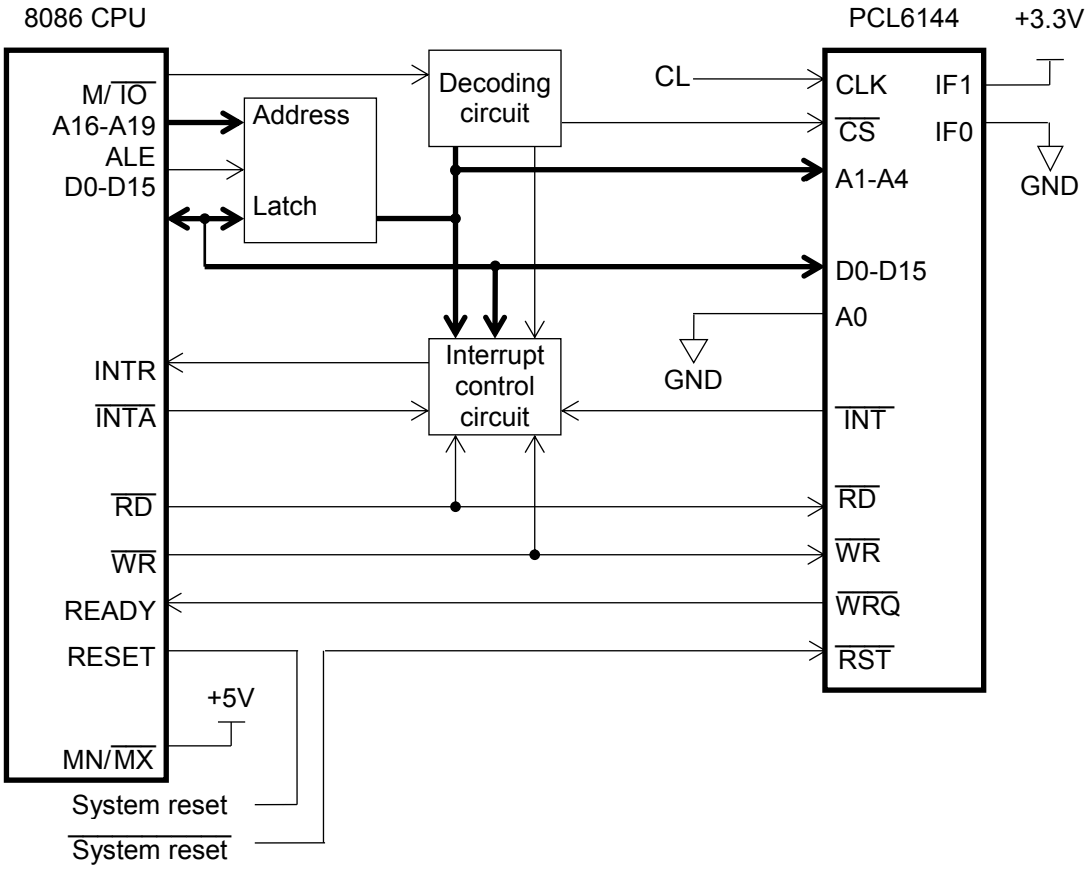
(2) 16-bit parallel bus I/F-2
Setting to select CPU bus I/F : IF1 = L, IF0 = H



Note: In PCL6144, A0 is connected to GND and A1~A4 are used. In PCL6114, A0 is connected to GND and A1~A2 are used. 16 bit I/F can be accessed by word (16-bit) only. It is not accessed by byte (8-bit).

(3) 16-bit parallel bus I/F-3

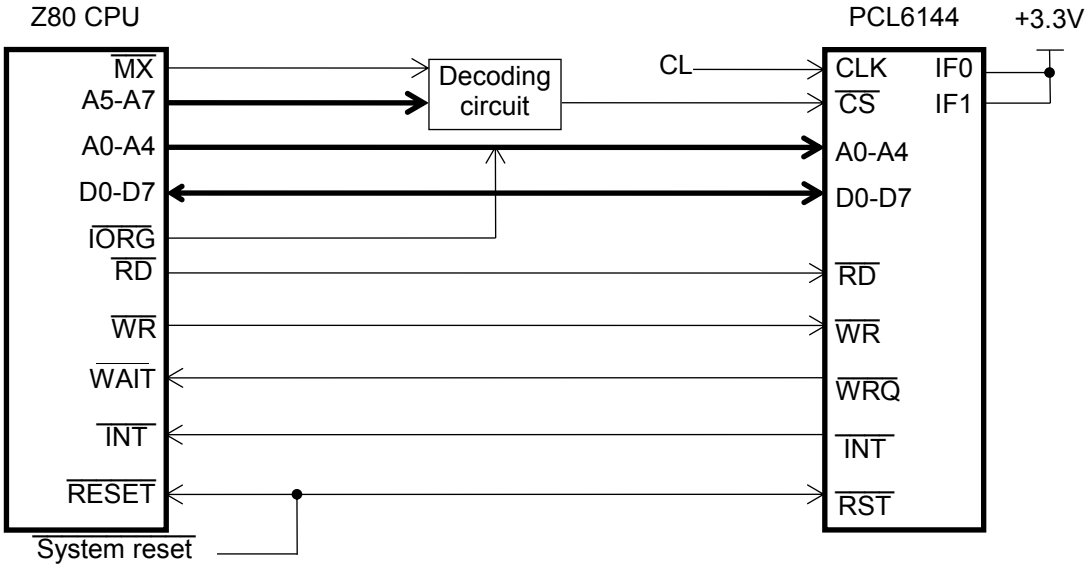
Setting to select CPU bus I/F : IF1 = H, IF0 = L



Note: In PCL6144, A0 is connected to GND and A1~A4 are used. In PCL6114, A0 is connected to GND and A1~A2 are used. 16 bit I/F can be accessed by word (16 bit) only. It is not accessed by byte (8-bit).

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

Setting to select CPU bus I/F : IF1 = H, IF0 = H



Note: In PCL6144, A0~A4 are used. In PCL6114, A0~A2 are used.

6-2-3. Address map

6-2-3-1. Axis arrangement map

In this LSI, the control address range for each axis is independent. It is selected by using address input terminal A4 and A3, 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 PCL6144. The PCL6114 does not have an A4 and A3 address input. Only X axis is available.

6-2-3-2. Internal map of each axis

The internal map of each axis is defined by address line inputs (A0,) A1 and A2.
<When 16-bit I/F-1 or 16-bit I/F-2 mode is selected>
1) Write cycle

A2	A1	Address bus	Processing details
1	1	COMW	Write axis assignment and control command
1	0	OTPW	Change status of general-purpose output ports (bits assigned as outputs are enabled only.)
0	1	BUFW0	Write to input/output buffer (bits15~0)
0	0	BUFW1	Write to input/output buffer (bits 31~16)

2) Readout cycle

A2	A1	Address bus	Processing details
1	1	MSTSW	Read main status (bits 15~0)
1	0	SSTSW	Read sub status and general-purpose I/O port.
0	1	BUFW0	Read from input/output buffer (bits 15~0)
0	0	BUFW1	Read from input/output buffer (bits 31~16)

<When 16-bit I/F-3 mode is selected>
1) Write cycle

A2	A1	Address bus	Processing details
0	0	COMW	Write axis assignment and control command
0	1	OTPW	Change status of the general-purpose output ports (only bits assigned as outputs are effective)
1	0	BUFW0	Write to input/output buffer (bits 15~0)
1	1	BUFW1	Write to input/output buffer (bits 31~16)

2) Readout cycle

A2	A1	Address bus	Processing details
0	0	MSTSW	Read main status (bits 15~0)
0	1	SSTSW	Read sub status or general-purpose input/output port
1	0	BUFW0	Read from input/output buffer (bits 15~0)
1	1	BUFW1	Read from input/output buffer (bits 31~16)

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

1) Write cycle

A2	A1	A0	Address bus	Processing details
0	0	0	COMB0	Write control commands
0	0	1	COMB1	Specify an axis (specify a axis to execute control commands)
0	1	0	OTPB	Change status of general-purpose output port (bits assigned as outputs are enabled only)
0	1	1	-	(Disabled)
1	0	0	BUFB0	Write to input/output buffer (bits 7~0)
1	0	1	BUFB1	Write to input/output buffer (bits 15~8)
1	1	0	BUFB2	Write to input/output buffer (bits 23~16)
1	1	1	BUFB3	Write to input/output buffer (bits 31~24)

2) Read cycle

A2	A1	A0	Address bus	Processing details
0	0	0	MSTSB0	Read main status (bits 7~0)
0	0	1	MSTSB1	Read main status (bits 15~8)
0	1	0	IOPB	Read general-purpose I/O ports
0	1	1	SSTSB	Read sub status
1	0	0	BUFB0	Read from input/output buffer (bits 7~0)
1	0	1	BUFB1	Read from input/output buffer (bits 15~8)
1	1	0	BUFB2	Read from input/output buffer (bits 23~16)
1	1	1	BUFB3	Read from input/output buffer (bits 31~24)

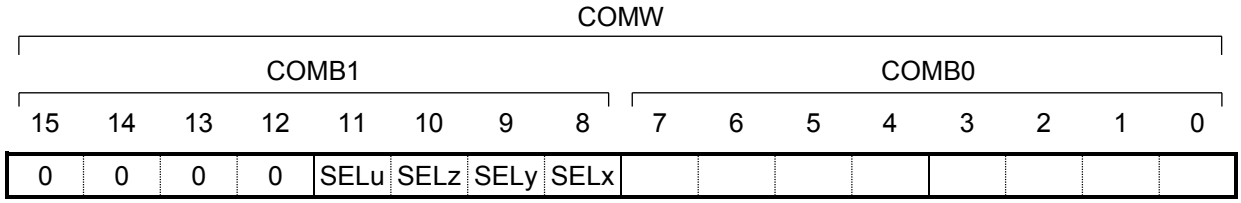
6-2-4. Description of map details

6-2-4-1. Write command code and axis selection

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

COMB0: Sets command code. For details, see “7. Commands.”

SELu~x: Selects an axis for executing commands. If all of the bits are 0, its own axis (selected by setting of A4 and A3) is selected. To write same command to more than one axis, set the bits of the selected axes to “1”. When writing to a register, details of the input/output buffer are written into the register for each axis. When reading from a register, details in the register are written into the input/output buffer for each axis.



Note 1: Axis selection using SELu~x are effective for all commands, not only register write/read commands.

Note 2: The PCL6144 has SELx~u. However, the PCL6114 does not have COMB1.

There are two methods to write to and read from a register, as follows. Mixed use of these methods is allowed because these are methods to develop software. The example below uses the PCL6144.

(1) Writing commands and data input and output are written as one set pae axis and 4 sets are used.
In this case, axis assignment (COMB1), other than starting or stopping an interpolation operation, is performed with 00h.
However, if $\overline{\text{CSTA}}$ and $\overline{\text{CSTP}}$ signals are used to start or stop an interpolation operation, 00h can also be used for this command.
When using multiple sets of PCL6114 or PCL6144 LSIs, a common program can be created easily.

<In the case 16-bit bus I/F=3>

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

(2) Writing commands uses common address and data I/O area is used per axis.
In this case, the axis must be specified for each command that is written. (However, SRST (04h) command (software reset) ignores axis assignment.)
Same registers of all axes are written / read with one command. Therefore the data setting time is reduced.

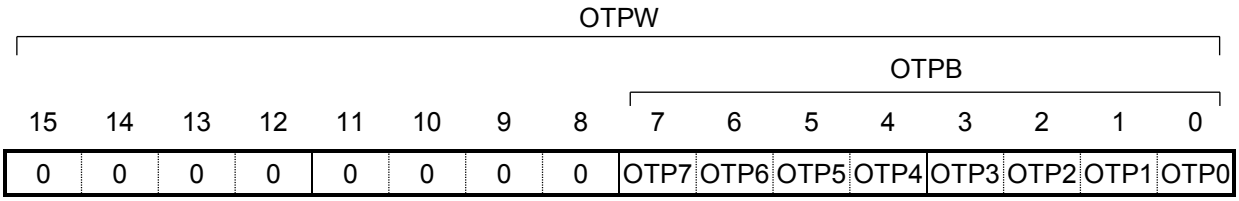
<In the case 16-bit I/F=3>

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

Note: The examples above use COMW on the X axis. Using COMW on any other axis can perform identical operation.

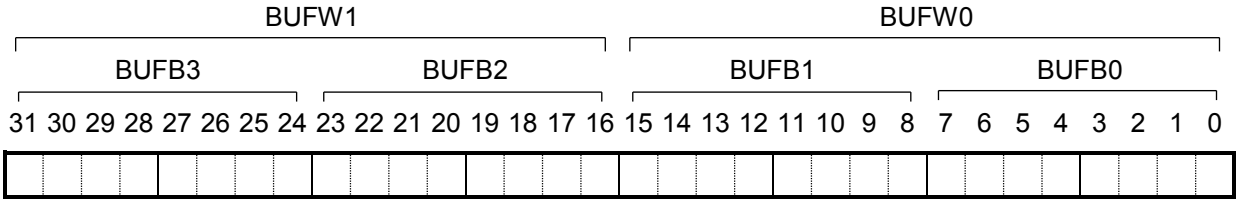
6-2-4-2. Write to output ports

Specifies output terminal status among general purpose I/O terminals P0~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.
OTP7~0: Specifies status of output terminals P7n~P0n (n = x, y, z and u).
A H level signal is output to the terminal when the corresponding bit is set to “1”.



6-2-4-3. Write/read input/output buffer

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



6-2-4-4. Reading main status

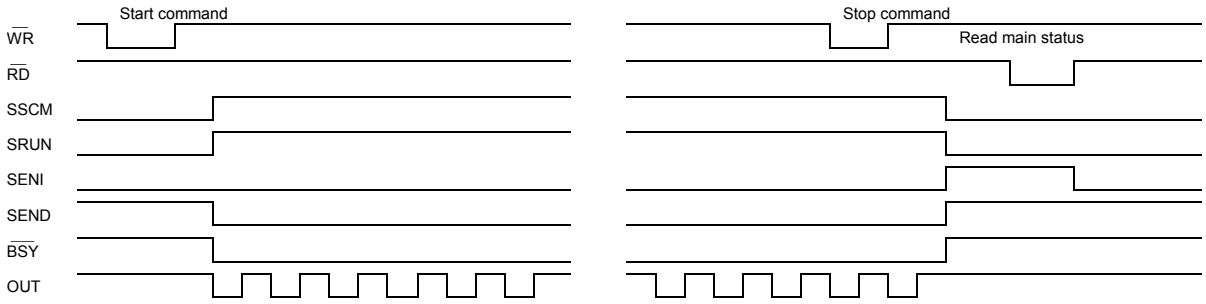
MSTSW

MSTSB1								MSTSB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SPRF	SEOR	0	SCP4	SCP3	SCP2	SCP1	SSC1	SSC0	SINT	SERR	SEND	SENI	SRUN	SSCM

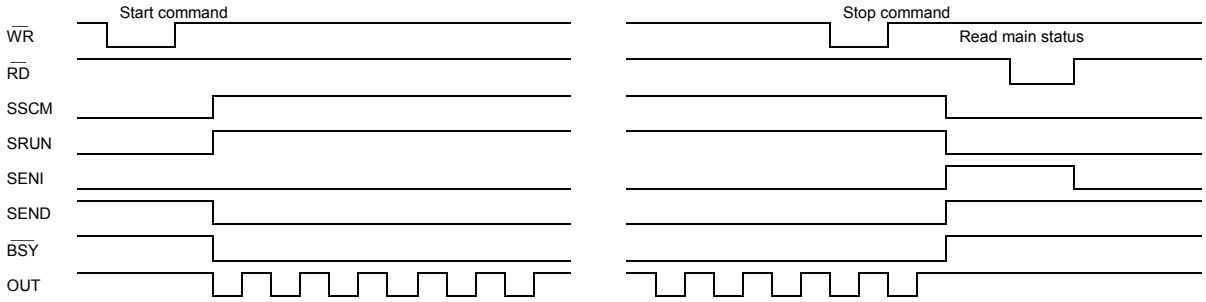
Bit	Bit name	Details
0	SSCM	Becomes “1” by writing a start command. Becomes “0” when the operation is stopped.
1	SRUN	Becomes “1” by start of pulse output. Becomes “0” when the operation is stopped.
2	SENI	When stop interrupt signal output is enabled (RENV2.IEND=“1”) and stop interrupt occurs, this bit becomes “1”. When the function is disabled (RENV2.IEND=0), it is always “0”. When automatic reset stop function is disabled (RENV2.MRST=“0”, it returns to “0” by reading main status. When the function is enabled (RENV2.IEND=“1”), it returns to “0” by executing SENIR (2Dh) command.
3	SEND	Becomes “0” by writing start command. Becomes “1” when the operation is stopped.
4	SERR	Becomes 1 when an error interrupt occurs. When automatic reset stop function is disabled (RENV2.MRST=“0”), it returns to “0” by reading REST register. When the function is enabled (RENV2.IEND=“1”), it returns to “0” by writing “1” to all bits of RIST register that are “1”.
5	SINT	Becomes “1” when an event interrupt occurs. When automatic reset stop function is disabled (RENV2.MRST=“0”, it returns to “0” by reading MSTSW status. When the function is enabled (RENV2.IEND=1), it returns to “0” by writing “1” to all bits of RIST register that are “1”.
7~6	SSC1~0	Sequence number for execution or stopping.
8	SCP1	Becomes “1” when the COMPARATOR 1 comparison conditions are met.
9	SCP2	Becomes “1” when the COMPARATOR 2 comparison conditions are met.
10	SCP3	It is used to monitor comparator comparison result for detecting (+) software limit. If PCL61x4 additional function is enabled (RENV3.M614=“1”), it becomes “1” when counter value for software limit control selected by counter selection for software limit control (RENV3.SLCU) is more than RCMP3, regardless of setting of software limit function selection (RENV3.SLM). When software for PCL61x3 is used, “0” is always gotten.
11	SCP4	It is used to monitor comparator comparison result for detecting (-) software limit. If PCL61x4 additional function is enabled (RENV3.M614=“1”), it becomes “1” when counter value for software limit control selected by counter selection for software limit control (RENV3.SLCU) is less than RCMP4, regardless of setting of software limit function selection (RENV3.SLM). When software for PCL61x3 is used, “0” is always gotten.
12	Not defined	(Always “0” is gotten.)
13	SEOR	When target position override cannot be executed (reading the RMV register while stopped), this bit changes to “1”. When automatic reset stop function is disabled (RENV2.MRST=“0”, it returns to “0” by reading main status. When the function is enabled (RENV2.MRST=“1”), it returns to “0” by executing SEORR (2Eh) command.
14	SPRF	Becomes 1 when the pre-register for the subsequent operation data is full.
15	Not defined	(Always “0” is gotten.)

Status change timing chart

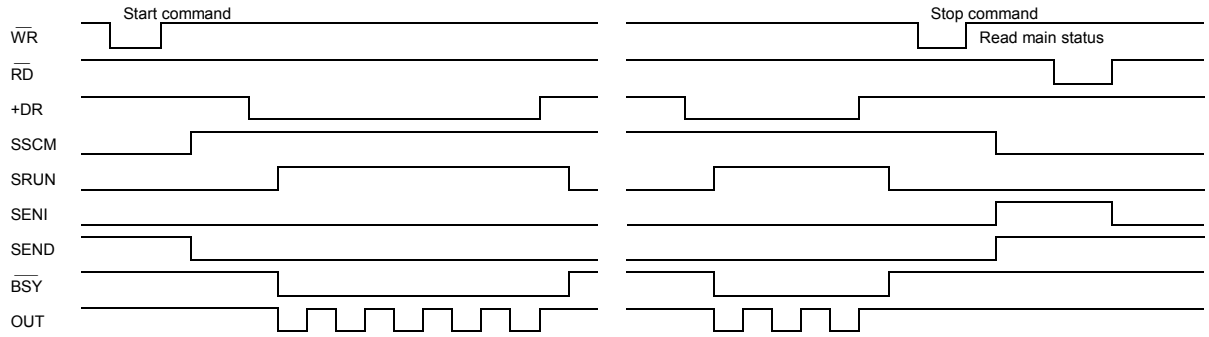
1) In continuous mode (RMD.MOD6~0=00h, 08h)



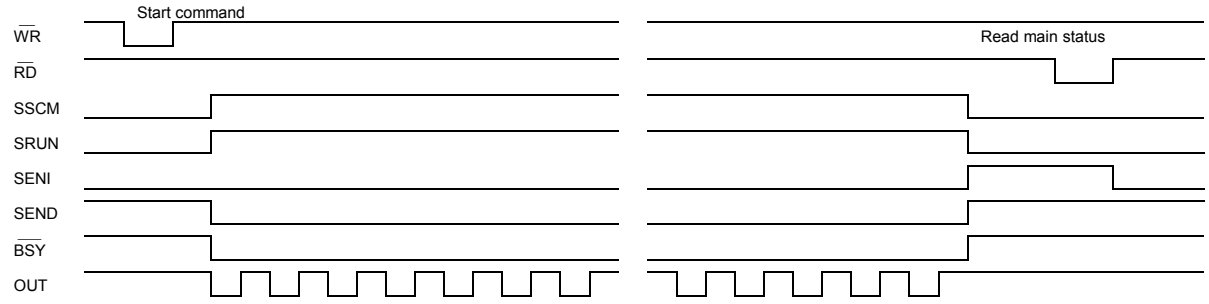
2) In PA/ PB continuous mode (RMD.MOD6~0=01h)



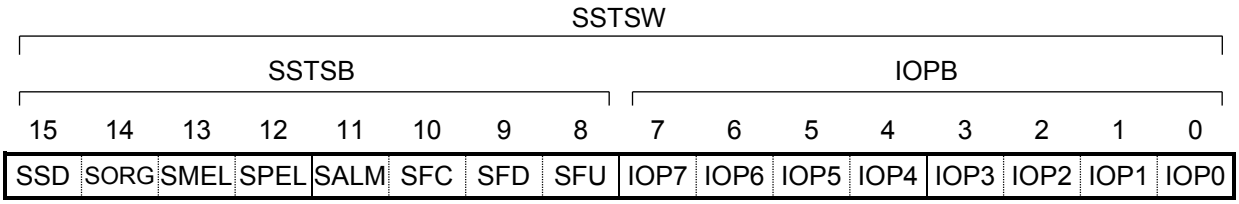
3) In DR continuous mode (RMD.MOD6~0=02h)



4) When the auto stop mode is selected such as positioning operation mode (RMD.MOD6~0=41h).



6-2-4-5. Reading sub status and input/output ports



Bit	Bit name	Description
7~0	IOP7~0	Reads status of P7~0 (0: L level, 1: H level)
8	SFU	Becomes “1” while accelerating.
9	SFD	Becomes “1” while decelerating.
10	SFC	Becomes “1” while feeding at constant speed.
11	SALM	Becomes “1” when ALM input is ON.
12	SPEL	Becomes “1” when +EL input is ON.
13	SMEL	Becomes “1” when -EL input is ON.
14	SORG	Becomes “1” when ORG input is ON.
15	SSD	Becomes “1” when SD input is ON. (Latch signal of SD input.)

6-3. Serial bus I/F

6-3-1. Setting of connected CPU

To select serial bus I/F circuit, use $\overline{RD}$ terminal and $\overline{WR}$ terminal.

[Example of connecting CPU signals with serial bus I/F]

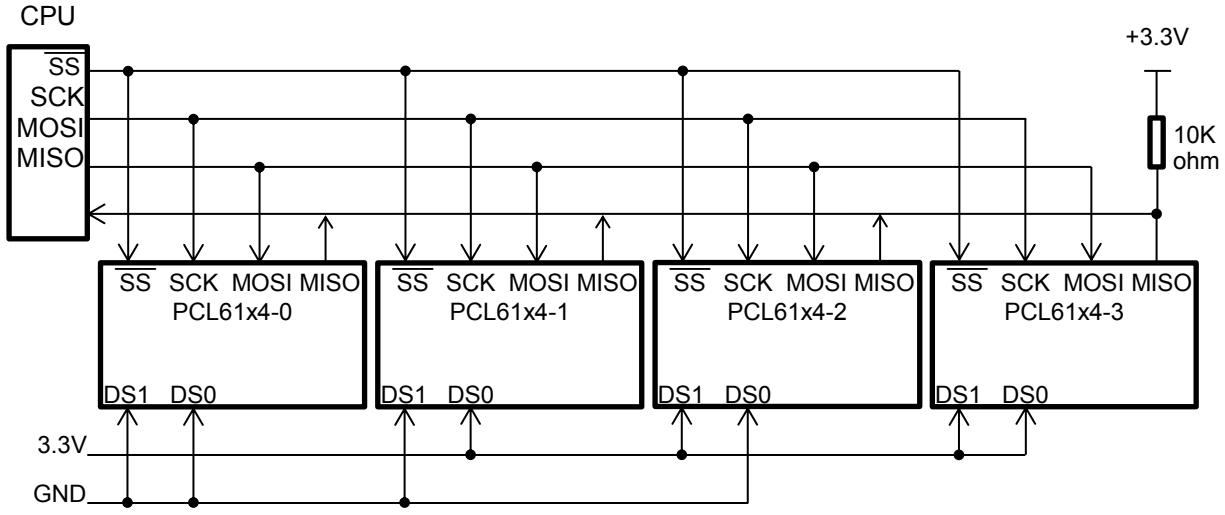
Setting		I/F name	CPU signals connected to terminals			
$\overline{RD}$	$\overline{WR}$		IF0 terminal	IF1 terminal	$\overline{CS}$ terminal	$\overline{WRQ}$ terminal
L	L	Serial I/F	SCK	MOSI	$\overline{SS}$	MISO
Other than the above		Parallel I/F	Described in 6.2.1. Setting of connected CPU			

Serial bus I/F: 4-wire synchronous serial bus I/F is built-in.
Extended connection with up to 4 LSIs is available with one slave select signal ($\overline{SS}$). LSIs connected are identified by setting device selection information to DS0 and DS1 terminals.

SCK (Serial Clock) is serial bus I/F clock terminal.
$\overline{SS}$ (Slave Select) is a terminal to select Slave (this LSI).
MOSI (Master Output Slave Input) is an input terminal from Master (CPU) to Slave (this LSI).
MISO (Master Input Slave Output) is an output terminal from Slave (this LSI) to Master (CPU).

6-3-2. Example of connecting CPU bus I/F

Setting condition to select CPU bus I/F : $\overline{RD}$=L and $\overline{WR}$=L.



Notes. Pull-up resistance is connected in order to prevent damage to CPU and PCL61x4 at floating.

6-3-3. Communication format

There are 4 types of communication formats according to types of axis selection cord.

1. Format to write commands
   1) Operation command

MOSI :	Axis selection code(S7-S0)	Command(C7-C0)

One communication has one axis selection code and one command.
Please see “7-1-1-2. Writing an operation command of serial bus I/F” for details of operation commands.
(General purpose output bit control command and control command is the same as operation command.)

   2) Register write command

MOSI :	Axis selection code (S7-S0)	Command(C7-C0)	Data x	Data y	Data z	Data u

One communication has one axis selection code and one command.
Number of data written equals to selected number of data by axis selection code.
Data size is 32 bits and order is “data[7-0]+data[15-8]+data[23-16]+data[31-24]”.
Please see “7-4-1-2. Writing to register of serial bus I/F” for details of register write command.

   3) Register read command

MOSI :	Axis selection code (S7-S0)	Command(C7-C0)

One communication has one axis selection code and one command.

MISO :	Data x	Data y	Data z	Data u

Number of data read equals to selected number of data by axis selection code.
Data size is 32 bits and order is “data[7-0]+data[15-8]+data[23-16]+data[31-24]”.
Please see “7-4-2-2. Reading from register of serial bus I/F” for details of register read command.

2. Format to read main format

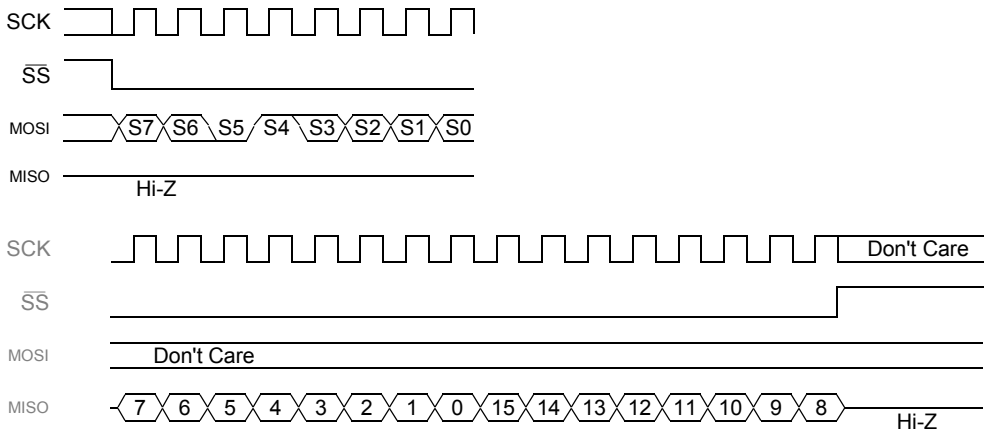
MOSI :	Axis selection code (S7-S0)

One communication has one axis selection code and one command.

MISO :	Data x	Data y	Data z	Data u

Number of data read equals to selected number of data by axis selection code.
Data size is 16 bits and order is “data[7-0]+data[15-8]”.
Please see “6-2-4-4.Reading main status” for details of main status.

Example : Reading main status



S7-S0 : Axis selection code (Format to read main status is S5=“0” and S4=“1”)
15-0 : Main status read data

3. Format to write general purpose port

MOSI :	Axis selection code (S7-S0)	Data x	Data y	Data z	Data u

One communication has one axis selection code.
Number of Data written equals to selected number of data by axis selection code.
Data size is 8 bit and the order is "data[7-0]".
Please see “7-5-1-2. Writing general-purpose output control command of serial bus I/F” for details of general port data.

4. Format to read sub status

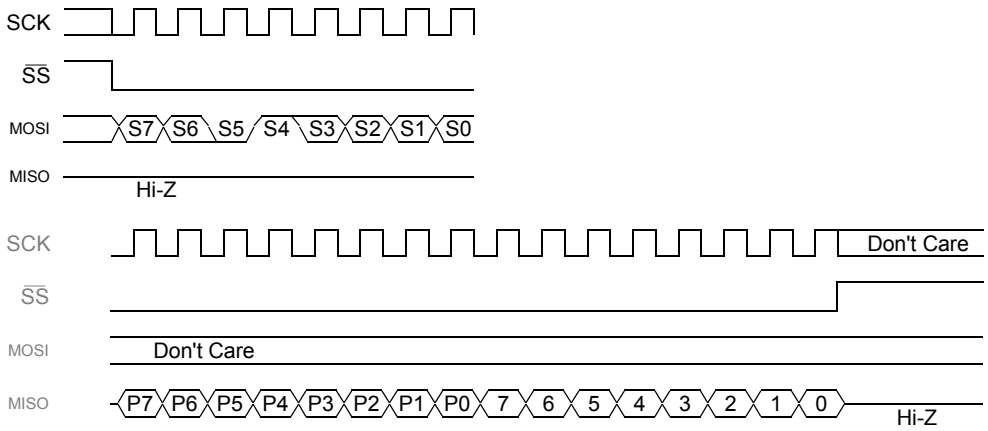
MOSI : | Axis selection code (S7-S0) |

One communication has one axis selection code and one command.

MISO : | Data x | Data y | Data z | Data u |

Number of data read equals to selected number of data by axis selection code.
Data size is 16 bits and the order is "data[7-0]+data[15-8]".
"Data[7-0]" is general-purpose ports input data and "data[15-8]" is sub status.
Please see "6-2-4-5. Reading sub status and input/output ports" for details of sub status data.

Example : Reading sub status



S7-S0 : Axis selection code (Format to read sub status is S5="1" and S4="1".)
P7-P0 : General purpose port read data
7-0 : Sub status read data

Note.

1. If several axes are selected by axis selection code, several axes can be accessed in one serial communication. In that case, data order is X>Y>Z>U among the selected axes.
2. Writing commands (registers) and general purpose port for several axes are written at the rising of $\overline{SS}$ signal.
3. Reading registers for several axes latches condition at writing "C0" bit and the results are read in series.
4. Reading main status and sub status (general purpose ports) latches condition at writing "C0" bit and the results are read in series.

6-3-3-1. Axis selection code

Axis selection code is 1 byte. It consists of the following 8 bits.

Bit	PCL6114	PCL6144
0	X axis selection	X axis selection
1	(Fixed to "0")	Y axis selection
2	(Fixed to "0")	Z axis selection
3	(Fixed to "0")	U axis selection
4	Type selection A	
5	Type selection B	
6	Device selection 0	
7	Device selection 1	

1. Axis selection (Bits 3-0)
   Selects an axis to write to or read from.
   The axis that axis select bit is “1” is selected.
   If all axis selection bits are “0” , only X axis is selected. (The same as “0001”b.)

2. Type selection (Bits 5-4)
   Selects 4 types of communication format.

Type selection		Communication format
B	A	
0	0	Writing command
0	1	Reading main status
1	0	Writing general purpose port
1	1	Reading sub status and general purpose port

3. Device selection (Bits 7-6)
   Normally only one LSI is connected by one slave select signal ($\overline{SS}$). However, this LSI can connect with up to 4 LSIs. Bits of device selection 1 and 0 corresponds to device select number terminals (DS1 and DS0). Several LSIs that has different setting of device select number terminals cannot be accessed simultaneously.

Device Selection		Terminal to select Device number	
1	0	DS1	DS0
0	0	L	L
0	1	L	H
1	0	H	L
1	1	H	H

6-3-3-2. Command

Command code is 1 byte. Please see “7. Commands” for details.

6-3-3-3. Data

Data is integral multiple of 1 byte and is lined from lower byte to upper byte and each byte is lined from Bit7(MSB) to Bit0(LSB) in order. Even though data is less than 8 bits, set data in a byte by substituting “0” for lacking bits.
When data is less than 4 bytes at writing register, set data in 4 byte unit by substituting 00h for lacking bytes as well.
When writing register for several axes in block, set data in 4 byte unit per axis.

7. Commands (Operation and Control Commands)

7-1. Operation commands

When you use 8 bit parallel bus I/F, 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 parallel bus I/F is used). The LSI will start and stop, as well as change the speed of the output pulses.
When you use 16 bit parallel I/F, the LSI will write 16-bit data including axis assignment and commands. In the case of serial bus I/F, it will write command write format.

7-1-1. Procedure for writing an operation command

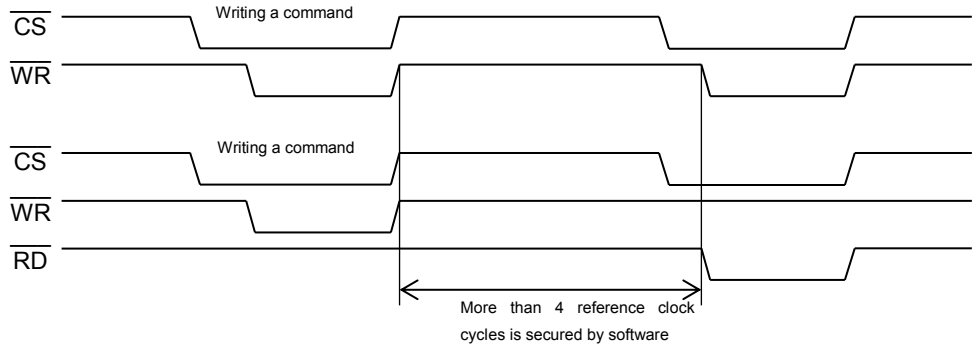
A waiting time of 4 reference clock cycles (approximately 0.2 us) is required when the next command is followed.

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, access is it after confirming that the $\overline{IFB}$ output signal = H level.

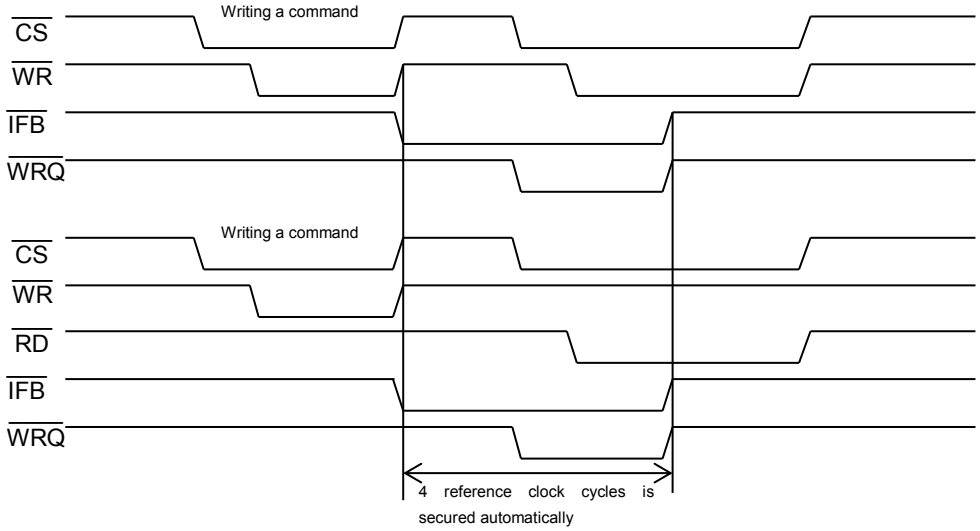
7-1-1-1. Writing an operation command of parallel bus I/F (the axis assignment is omitted)

Writes a command to COMB0.

1. When $\overline{WRQ}$ output is not used



2. When $\overline{WRQ}$ output is used

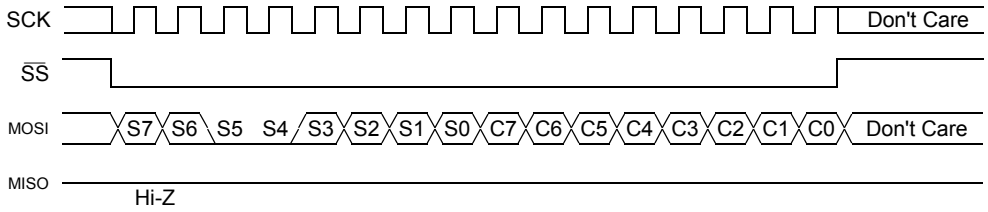


Notes: $\overline{WRQ}$ signal becomes L level while $\overline{CS}$ and $\overline{IFB}$ become L level.

7-1-1-2. Writing an operation command of serial bus I/F

Uses format to write commands.

Writing operation command



S7-0 : Axis selection code (S5="0" and S4="0" because of format to write commands.)
C7-0 : Command

7-1-2. Start commands

1) Start commands

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

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

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

2) Residual pulses start commands

If these commands are written after the motor is stopped on the way to a positioning, it will continue movement for the number of pulses left in the positioning counter.

COMB0	Symbol	Description
54h	CNTFL	FL constant speed start for remaining pulses
55h	CNTFH	FH constant speed start for remaining pulses
56h	CNTD	High speed start 1 for remaining pulses (FH constant speed → Deceleration stop)
57h	CNTUD	High speed start 2 for remaining pulses (Acceleration → FH constant speed → Deceleration stop)

3) Simultaneous start commands

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

COMB0	Symbol	Description
06h	CMSTA	Outputs one shot of the start pulse from the $\overline{\text{CSTA}}$ terminal.
2Ah	SPSTA	Performs the same processing as when a $\overline{\text{CSTA}}$ signal is supplied, for its own axis only.

7-1-3. Speed change commands

If these commands are written 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	Changes to the FL speed immediately (Change to same operation status as start at FL constant speed).
41h	FCHGH	Changes to the FH speed immediately. (Change to same operation status as start at FL constant speed).
42h	FSCHL	Decelerates and changes to the FL speed. (Change to same operation status as start at high-speed).
43h	FSCHH	Accelerates and changes to the FH speed. (Change to same operation status as start at high-speed).

7-1-4. Stop command

1) Stop commands

If these commands are written while the motor is operating, motor will stop.

COMB0	Symbol	
49h	STOP	Stops immediately if this command is written while the motor is operating.
4Ah	SDSTP	Decelerates to the FL constant speed and stops if this command is written while feeding at FH constant speed or high speed Stops immediately if this command is written while feeding at FL constant speed.

2) Simultaneous stop command

Stop movement 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.

3) Emergency stop command

Stops movement on an 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 output of terminals P0~P7 by bit.
When the terminals are designated as outputs, signals are output from terminals P0~P7 in response to commands. Commands that correspondent to terminals set as other than outputs are ignored.
The write procedures are the same as for operation commands.
In addition to this command, with parallel bus I/F, you can set 8 bits as a group by writing to a general-purpose output port (OTPB: Address 2 with an 8-bit I/F). With serial bus I/F, you can set 8 bits as a group by writing format to write general purpose ports. See section “7-5. General-purpose output port control commands.”

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

7-3. Control commands

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

7-3-1. Software reset command

Resets this LSI.

COMB0	Symbol	Description
04h	SRST	Software reset. (Same function as $\overline{RST}$ terminal = L level.)

Note: After writing this command, do not access during 12 cycles of CLK.

7-3-2. Counter reset command

Resets counter value to zero.

COMB0	Symbol	Description
20h	CUN1R	Reset COUNTER 1.
21h	CUN2R	Reset COUNTER 2.

7-3-3. ERC output control command

Controls 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 output.

7-3-4. Pre-register control command

Cancels pre-register settings.
See section "8-2. Pre-register", for details about pre-register.

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

7-3-5. PCS input command

Entering this command has the same results as turning PCL signal ON.

COMB0	Symbol	Description
28h	STAON	Substitute PCS input.

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

Entering this command has the same result as turning LTC signal ON.

COMB0	Symbol	Description
29h	LTCH	Substitute LTC (latch counter) input.

7-3-7. SENI, SEOR reset command

When function to stop automatic reset of main status is enabled (RENV2.MRST="1"), this command reset each bit manually.

COMB0	Symbol	Description
2Dh	SENIR	Reset stop interrupt bit (MSTSW.SENI).
2Eh	SEORR	Reset bit of target position override failure (MSTSW.SEOR).

7-4. Register control command

With parallel bus I/F, the LSI can copy data between a register and the I/O buffer by writing a Register Control command to COMB0 (Address 0 with an 8-bit I/F). With serial bus I/F, it copy data between register and I/O buffer by writing format to write commands.

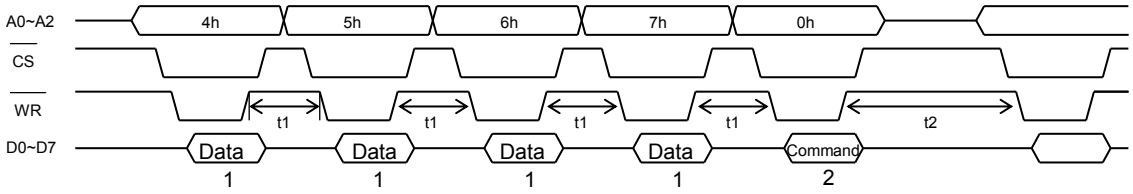
Note: When using the I/O buffer while responding to an interrupt, you need to read I/O buffer data and use PUSH command before performing the interrupt routine. After using it, you need to restore it to an original value and use POP command.

7-4-1. Procedure for writing data to a register

7-4-1-1. Writing to register of parallel bus I/F (the axis assignment is omitted)

1) Writes the data that will be written to a register into the I/O buffer (addresses 4~7 with 8-bit I/F). The order in which the data is written does not matter. However, secure two reference clock cycles (t1) between these writings.
2) Then, write a "register write command" to COMB0 (address 0 with 8-bit I/F).
   After writing one set of data, wait at least four cycles (approx. 0.2 us, t2) before writing the next set of data.

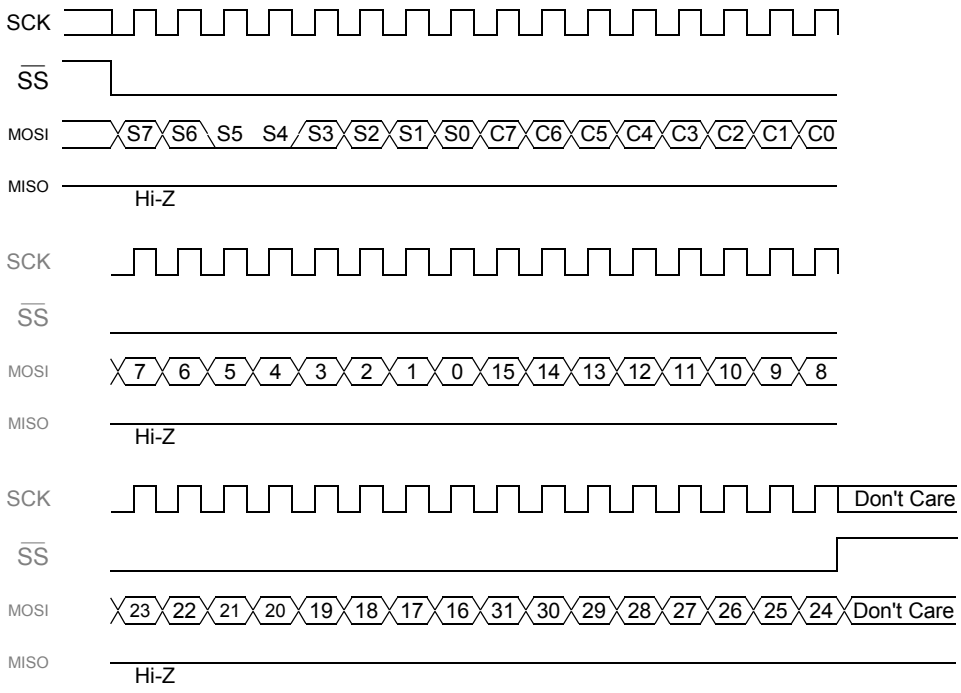
   In both case 1) and case 2), when the $\overline{WRQ}$ output is connected to the CPU, the CPU wait control function will provide the waiting time between write operations automatically.



7-4-1-2. Writing to register of serial bus I/F

Uses format to write commands

Writing to register



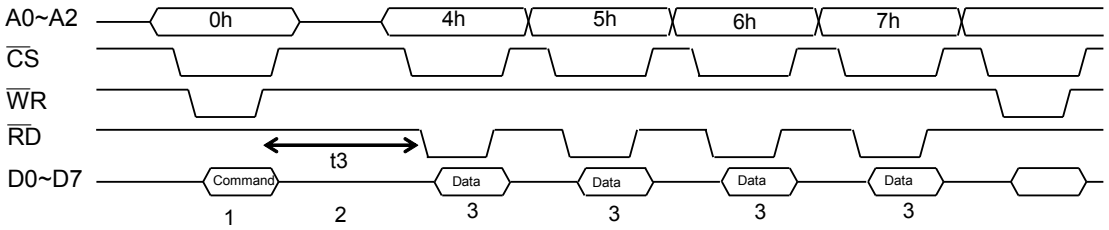
S7-0 : Axis selection code (S5="0" and S4="0" because of format to write commands.)
C7-0 : Command
31-0 : Register write data

7-4-2. Procedure for reading data from a register

7-4-2-1. Reading from register of parallel bus I/F (the axis assignment is omitted)

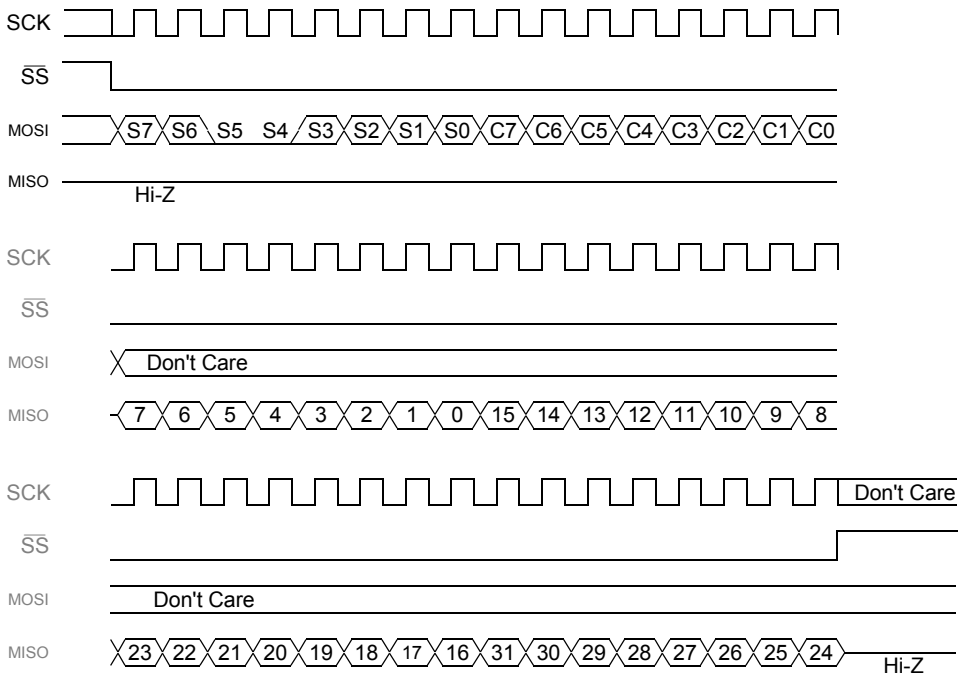
1) First, write a "register read out command" to COMB0 (address 4~7 when an 8-bit I/F is used).
2) Wait at least four reference clock cycles (approx. 0.2 usec, t3) for the data to be copied to the I/O buffer.
3) Read the data from the I/O buffer (address 4 to 7 when an 8-bit-I/F is used). The order for 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.



7-4-2-2. Reading from register of serial bus I/F

Uses format to write commands
Example. Reading from register



S7-0 : Axis selection code (Because of command write format, S5="0" and S4="0".)
C7-0 : Command
31-0 : Register read data

7-4-3. Table of register control commands

The following registers are available for each axis.

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	32	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	16	RUR	D3h	RRUR	93h	WRUR	PRUR	C3h	RPRUR	83h	WPRUR
5	Deceleration rate	16	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 feed data	32	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	32	RENV2	DDh	RRENV2	9Dh	WRENV2	-	-	-	-	-
14	Environment setting 3	26	RENV3	DEh	RRENV3	9Eh	WRENV3	-	-	-	-	-
15	COUNTER 1 (command)	32	RCUN1	E3h	RRCUN1	A3h	WRCUN1	-	-	-	-	-
16	COUNTER 2 (mechanical)	32	RCUN2	E4h	RRCUN2	A4h	WRCUN2	-	-	-	-	-
17	Comparison data data for comparator 1	32	RCMP1	E7h	RRCMP1	A7h	WRCMP1	-	-	-	-	-
18	Comparison data for comparator 2	32	RCMP2	E8h	RRCMP2	A8h	WRCMP2	-	-	-	-	-
19	Comparison data for comparator 3 (+) software limit	32	RCMP3	E9h	RRCMP3	A9h	WRCMP3	-	-	-	-	-
20	Comparison data for comparator 4 (-) software limit	32	RCMP4	EAh	RRCMP4	AAh	WRCMP4	-	-	-	-	-
21	Event interrupt factor setting	13	RIRQ	ECh	RRIRQ	ACh	WRIRQ	-	-	-	-	-
22	COUNTER 1 latched data	32	RLTC1	EDh	RRLTC1	-	-	-	-	-	-	-
23	COUNTER 2 latched data	32	RLTC2	EEh	RRLTC2	-	-	-	-	-	-	-
24	Extension status	17	RSTS	F1h	RRSTS	-	-	-	-	-	-	-

No.	Detail	Bit	Register					Pre-register				
			Name	Read command		Write command		Name	Read command		Write command	
				COMB0	Symbol	COMB0	Symbol		COMB0	Symbol	COMB0	Symbol
25	Get error interrupt factor status	11	REST	F2h	RREST	B2h	WREST	-	-	-	-	-
26	Get event interrupt factor status	16	RIST	F3h	RRIST	B3h	WRIST	-	-	-	-	-
27	Positioning counter	32	RPLS	F4h	RRPLS	-	-	-	-	-	-	-
28	EZ counter, current speed monitor	20	RSPD	F5h	RRSPD	-	-	-	-	-	-	-
29	Ramping-down point setting	24	PSDC	F6h	RPSDC	-	-	-	-	-	-	-

The followings are common registers. (These are used with serial bus I/F)

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	Setting of general purpose (GP0~15) specifications	16	RGPM	FAh	RRGPM	BAh	WRGPM	-	-	-	-	-
2	Setting of general purpose (GP0~15) data	16	RGPD	FBh	RRGPD	BBh	WRGPD	-	-	-	-	-

* Common registers can be accessed from any axes. The priority of setting at writing for several axes in block is X>Y>Z>U.

7-5. General-purpose output port control commands

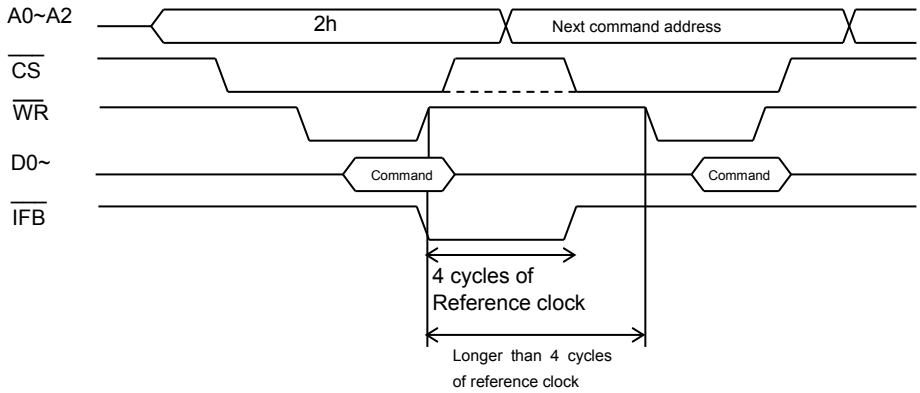
With parallel bus I/F, the LSI will control the output of the P0 to P7 terminals by writing an output control command to the output port (OTPB: Address 2 when using an 8-bit-I/F). With serial bus I/F, it will write format to write general purpose ports.
When the I/O setting for P0~P7 is set to output, the LSI will output signals from terminals P0~P7 in response to commands. With 16-bit parallel bus I/F, the upper 8 bits are discarded. However, they should be set to zero to maintain future compatibility.
The output setting of terminals P0~P7 is latched, even after the I/O setting is set as input.
The output setting for each terminal can be set individually using bit control commands.

7-5-1. Command writing procedures

To continue with a next command, the LSI must wait for four reference clock cycles (approx. 0.2 usec). The $\overline{WRQ}$ terminal outputs a wait request signal.
If the $\overline{WRQ}$ terminal signal is not connected to CPU, please ensure the interval by software or access after the $\overline{IFB}$ output terminal is confirmed to be H level.

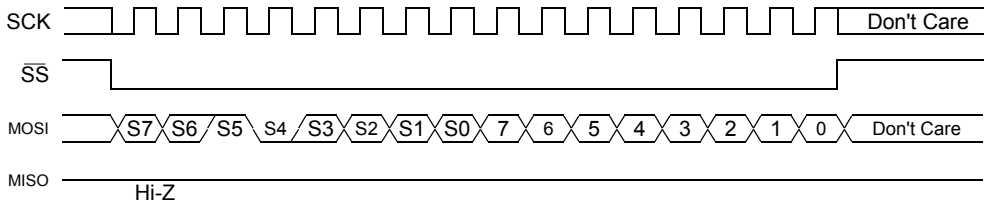
7-5-1-1. Writing general purpose output control command of parallel bus I/F

Writes commands to output port (OTPB: Address 2 with 8-bit I/F).



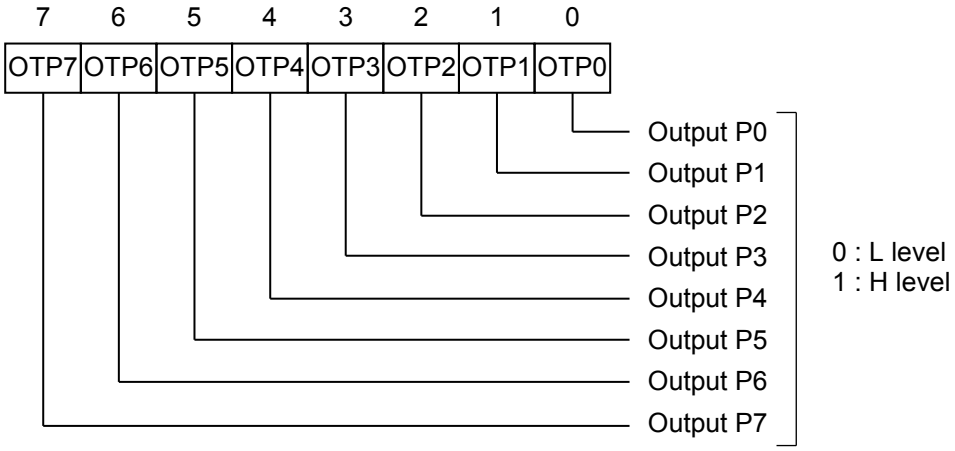
7-5-1-2. Writing general purpose output control command of serial bus I/F

Uses format to write to general purpose ports.
Writing to general purpose ports



S7-0 : Axis selection code (because of format to write general purpose ports, S5="1" and S4="0")
7-0 : Bit allocation data of general purpose port

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	Pre-register name
1	RMV	32	R/W	Feeding amount, target position	PRMV
2	RFL	14	R/W	Initial speed	PRFL
3	RFH	14	R/W	Operation speed	PRFH
4	RUR	16	R/W	Acceleration rate	PRUR
5	RDR	16	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	32	R/W	Main axis feeding amount during linear interpolation	PRIP
10	RUS	13	R/W	S-curve range during acceleration	PRUS
11	RDS	13	R/W	S-curve range during deceleration	PRDS
12	RENV1	32	R/W	Environment setting 1 (specifies I/O terminal details)	-
13	RENV2	32	R/W	Environment setting 2 (specifies general-purpose port details)	-
14	RENV3	26	R/W	Environment setting 3 (specifies counter details)	-
15	RCUN1	32	R/W	COUNTER 1 (command position counter)	-
16	RCUN2	32	R/W	COUNTER 2 (mechanical position counter)	-
17	RCMP1	32	R/W	Comparison data for comparator 1	-
18	RCMP2	32	R/W	Comparison data for comparator 2	-
19	RCMP3	32	R/W	Comparison data for comparator 3 (+) software limit	-
20	RCMP4	32	R/W	Comparison data for comparator 4 (-) software limit	-
21	RIRQ	13	R/W	Specifies event interrupt factor	-
22	RLTC1	32	R	COUNTER 1 (command position) latched data	-
23	RLTC2	32	R	COUNTER 2 (mechanical position) latched data	-
24	RSTS	17	R	Extension status	-
25	REST	11	R	Gets error interrupt factor	-
26	RIST	16	R	Gets event interrupt factor	-
27	RPLS	32	R	Positioning counter (number of remaining pulses to feed)	-
28	RSPD	20	R	EZ counter, current speed monitor	-
29	RSDC	24	R	Ramping-down point setting value	-

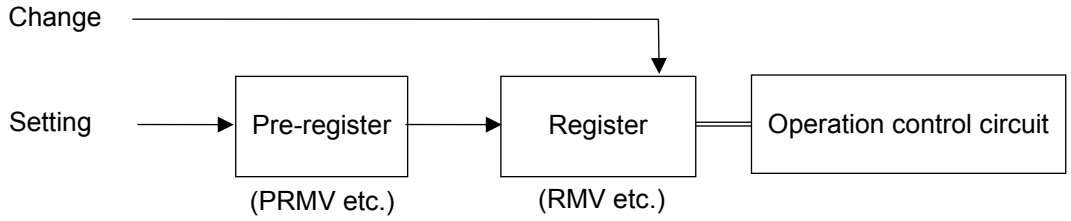
The followings are common registers. (These are used with serial bus I/F.)

No.	Register name	Bit length	R/W	Details	Pre-register name
1	RGPM	16	R/W	Specifies setting of general purpose ports (GP0~15)	-
2	RGPD	16	R/W	Data of general purpose ports (GP0~15)	-

8-2. Pre-register

The registers (RMV, RFL, RFH, RUR, RDR, RMG, RDP, RMD, RIP, RUS and RDS) and start commands have pre-registers.

The term pre-register refers to a register which sets 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 pre-register. Start command's pre-register is for write only.
To change the current operation status, such as changing speed, new data are written into the register.
The data will be shifted (copied) from "pre-register" to "register" at the end of an operation.
One set of operation data uses multiple pre-registers (PRMV, PRFH, etc.). If the current operation completes before the next set of operation data has been placed in all of the pre-registers, the LSI may start with incomplete data. In order to prevent this problem, "determined/not determined" status is used.
When a start command is written, the operation data is considered to be determined, and the LSI 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 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 to be determined and the LSI 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 only. (Data 2)

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

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

Procedure	Pre-register	Register	SPRF
Reset	0 Not determined	0 Not determined	0
1)	Data 1 Not determined	Data 1 Not determined	0
2)	Data 1 Not determined	Data 1 Determined	0
3)	Data 2 Not determined	Data 1 Determined	0
4)	Data 2 Determined	Data 1 Determined	1
5)	Data 2 Not determined	Data 2 Determined	0
6)	Data 2 Not determined	Data 2 Not determined	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 LSI stops operation.

In step (5) and (6) above, the data in the pre-register is "not determined", therefore, the next set of operation data can be written.
Data written to the pre-register when the data in the pre-register is already "determined" will be ignored.
When the pre-register is determined, MSTSW.SPRF=“1”.
Also, the LSI outputs an $\overline{\text{INT}}$ signal when the pre-register changes to "not determined" status by setting RIRQ.IRNM register.

Further, in any of the following cases, the pre-register has a "not determined" status, so that you can cancel a continuous start even if the current operation is finished.

1) Writing PRECAN (26h) command.
2) A stop ordered by using the immediate STOP (49h) command or SDSTP (4Ah) command.
   While in a positioning operation, when SDSTP (4Ah) command is written during auto deceleration, the mechanical position reaches the target position. However, the pre-register is declared to be "not determined" and the next operation will be cancelled.
3) When the motor stops because of an error (When any of bits10~9 and 6~0 in the REST register changes to “1”.)

Note: To automatically start the next operation using pre-register, set the operation complete timing to "end of cycle" (PRMD.METM=“0”). If the "end of pulse" (PRMD.METM=“1”) is selected, the interval between the last pulse and the next operation's start pulse will be narrow, that is, 17 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 registers

The default values of all the registers and pre-registers are “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. (Code Extension)

8-3-1. PRMV (RMV) register

This register is used to specify target position for positioning operations. RMV is the register for PRMV.

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

Setting range: -2,147,483,648 ~ +2,147,483,647.
By changing the RMV register while in operation, the target position can be overridden.

8-3-2. PRFL (RFL) register

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

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

8-3-3. PRFH (RFH) register

This pre-register is used to specify the operation speed.
RFH is the register for PRFH. Write to this register to override operation speed.

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

8-3-4. PRUR (RUR) register

This pre-register is used to specify the acceleration rate.
RUR is the register for PRUR.

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

Setting range is 1 ~ 16,383.

8-3-5. PRDR (RDR) register

This pre-register is used to specify deceleration rate.
RDR is the register for PRDR.

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 ~ 16,383.
When PRDR = “0”, deceleration rate will be a value set by PRUR register
Note: When automatic setting is selected for the ramp down point (PRMD.MSDP = “0”), enter the same value as used for the PRUR register, or “0”.

8-3-6. PRMG (RMG) register

This pre-register is used to set the speed magnification rate.
RMG is the register for PRMG.

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 setting range is 1 ~ 4,095.
Set relationship between the speed register PRFL (RFL), PRFH (RFH) values and actual operation speeds.
The actual operation speed [pps] is a product of speed magnification rate and 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 ~ 4,914.9	59 (003Bh)	20	20 ~ 327,660
2399 (095Fh)	0.5	0.5 ~ 8,191.5	23 (0017h)	50	50 ~ 819,150
1199 (04AFh)	1	1 ~ 16,383	11 (000Bh)	100	100 ~ 1,638,300
599 (0257h)	2	2 ~ 32,766	5 (0005h)	200	200 ~ 3,276,600
239 (00EFh)	5	5 ~ 81,915	2 (0002h)	400	400 ~ 6,553,200
119 (0077h)	10	10 ~ 163,830	1 (0001h)	600	600 ~ 9,829,800

8-3-7. PRDP (RDP) register

This pre-register is used to set a ramping-down point (deceleration start point) for positioning operations.
RDP is the register for PRDP.

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
#	#	#	#	#	#	#	#																								

Bits marked with a "#" symbol of PRDP register are ignored when written and their content changes when read according to the setting of PRMD.MSDP.
Bits marked with a "#" symbol of RDP register are ignored when written and their content changes when read according to the setting of RMD.MSDP.

MSDP	Setting details	bit #	Setting range
0	Offset for automatically set values. When a positive value is entered, the LSI will start deceleration earlier and the FL speed range will be used longer. When a negative value is entered, the LSI will start deceleration later and the speed will not reach the FL speed.	Same as bit 23.	-8,388,608 ~ +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 ~ 16,777,215

8-3-8. PRMD (RMD) register

This pre-register is used to set 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	MOD6~0						

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	MSY	MSY0	MSN1	MSN0

Bits	Bit name	Description
Sets basic operation mode		
6~0	MOD	Set operation mode. “000 0000”b (00h): Continuous positive rotation controlled by command control. “000 1000”b (08h): Continuous negative rotation controlled by command control. “000 0001”b (01h): Continuous operation controlled by pulsar (PA/PB) input. “000 0010”b (02h): Continuous operation controlled by external signal (+DR/-DR) input. “001 0000”b (10h): Positive rotation origin return operation. “001 1000”b (18h): Negative rotation origin return operation. “100 0001”b (41h): Positioning operation (specifies incremental target position) “100 0111”b (47h): Timer operation “101 0001”b (51h): Positioning operation controlled by pulsar (PA/PB) input. “101 0110”b (56h): Positioning operation controlled by external signal (+DR/-DR) input. “110 0010”b (62h): Continuous linear interpolation “110 0011”b (63h): Linear interpolation
7	Not defined	(Always set “0”.)
Optional setting items		
8	MSDE	0: SD input will be ignored. (Checking can be done with sub status and RSTS register) 1: Decelerates (deceleration stop) by turning SD input ON.
9	MINP	0: Delay using an INP input will be invalid. (Checking can be done with RSTS register) 1: Completes operation by turning INP input ON.
10	MSMD	Specify an acceleration/deceleration type for high speed feed. (0: Linear acceleration/deceleration. 1: S-curve acceleration/deceleration.)
11	MCCE	1: Stop counting output pulses on COUNTER 1 and COUNTER 2. This is used to move a mechanical part without changing the position controlled by the LSI When “EA/EB input” is selected for the counter input selection (RENV3.CIS1, RENV3.CIS2), the LSI 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 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, PRUR = PRDR and PRUS = PRDS should be set.
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”.)
17~16	MSN1~0	When you want to control an operation block, specify a 2-bit sequence number. By reading the main status, a sequence number currently being executed (MSTSW.SSC1~0) can be checked. Setting the sequence number does not affect the operation.

Bits	Bit name	Description
19~18	MSY1~0	After writing a start command, the LSI will start an axis synchronization operation based on other timing. "00"b : Start immediately. "01"b : Start on a $\overline{\text{CSTA}}$ input (or CMSTA (06h) command or STSTA(2Ah) command. "10"b : Start with an internal synchronous start signal. "11"b : Start when a specified axis stops moving.
23~20	MAX3~0	Specify an axis to check for an operation stop with PRMD.MSY="11"b. Setting examples "0001"b : Starts when X axis stops. "0010"b : Starts when Y axis stops. "0100"b : Starts when Z axis stops. "1000"b : Starts when U axis stops. "0101"b : Starts when both X and Z axes stop. "1111"b : 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 = L level. Set this bit to "1" to decelerate simultaneously with other axes.
29	MCDO	1: Outputs a L level from $\overline{\text{CSD}}$ terminal in decelerating or at FL constant speed.
31~30	Not defined	(Always set 0.)

8-3-9. PRIP (RIP) register

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

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

- When PRMD.MOD6~0 register is set as shown below, the register is enabled.
"110 0010"b (62h): Continuous linear interpolation (continuous operation with linear interpolation ratio).
"110 0011"b (63h): Linear interpolation.
- Setting range: 0 ~ 2,147,483,648.

8-3-10. PRUS (RUS) register

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

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 ~ 8,191.
When "0" is entered, the value of $\frac{\text{PRFH-PRFL}}{2}$ will be calculated internally and applied.

8-3-11. PRDS (RDS) register

This pre-register is used to specify S-curve range of 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 ~ 8,191.

When “0” is entered, the value of $\frac{PRFH-PRFL}{2}$ 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 (PRMD.MSDP = 0).

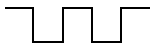
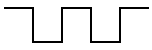
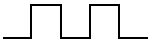
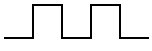
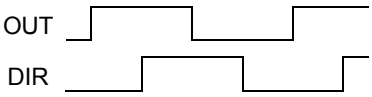
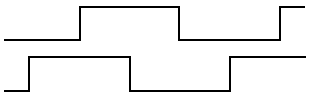
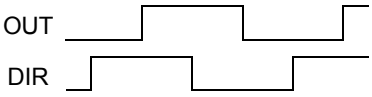
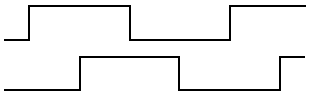
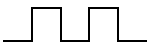
8-3-12. RENV1 register

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
2~0	PMD2~0	Specifies OUT output pulse details

PMD2~0	Operation in (+) direction		Operation in (-) direction	
	OUT output	DIR output	OUT output	DIR output
“000”b		High		Low
“001”b		High		Low
“010”b		Low		High
“011”b		Low		High
“100”b		High	High	
“101”b				
“110”b				
“111”b		Low	Low	

Bits	Bit name	Description
3	ELM	Specifies the process to occur when the EL signal input is turned ON. (0: Immediate stop. 1: Deceleration stop.) Note.1 and 2
4	SDM	Specifies the process to occur when the SD signal input is turned ON. (0: Deceleration only. 1: Deceleration and stop.) Note. 2
5	SDLT	Specifies the latch function of the SD signal input. (0: OFF. 1: ON.) Turns ON when the SD signal width is short. When the SD input is OFF at the start, the latch signal is reset. The latch signal is also reset when RENV.SDLT= “0”.

Bits	Bit name	Description
6	SDL	Specifies the SD signal input logic. (0: Negative logic. 1: Positive logic.)
7	ORGL	Specifies the ORG signal input logic. (0: Negative logic. 1: Positive logic.)
8	ALMM	Specifies the process to occur when the ALM signal 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.)
10	EROE	1: Automatically outputs an ERC signal when the motor 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.
11	EROR	1: Automatically outputs the ERC signal when origin return is completed.
14~12	EPW2~0	Specify the output pulse width of the ERC signal. (when CLK=19.6608MHz) "000"b : 11 ~ 13 us "100"b : 11 ~ 13 ms "001"b : 91 ~ 98 us "101"b : 46 ~ 50 ms "010"b : 360 ~ 390 us "110"b : 93 ~ 100 ms "011"b : 1.4 ~ 1.6 ms "111"b : Level output
15	ERCL	Specifies the ERC signal output logic. (0: Negative logic. 1: Positive logic.)
17~16	ETW1~0	Specify the ERC signal OFF timer time. (when CLK=19.6608MHz) "00"b : 0 us "01"b : 11 ~ 13 us "10"b : 1.4 ~ 1.6 ms "11"b : 93 ~ 100 ms
18	STAM	Specifies the $\overline{\text{CSTA}}$ signal input type. (0: Level trigger. 1: Edge trigger.)
19	STPM	Specifies a stop method using $\overline{\text{CSTP}}$ input. (0: Immediate stop. 1: Deceleration stop.) Note 2.
21~20	FTM1~0	Selects features of +EL, -EL, SD, ORG, ALM, and INP signals input. "00"b : Signal pulses shorter than 3.2 us are ignored. (when CLK=19.6608MHz) "01"b : Signal pulses shorter than 25 us are ignored. (when CLK=19.6608MHz) "10"b: Signal pulses shorter than 200 us are ignored. (when CLK=19.6608MHz) "11"b : Signal pulses shorter than 1.6 ms are ignored. (when CLK=19.6608MHz)
22	INPL	Specifies INP signal input logic. (0: Negative logic. 1: Positive logic.)
23	LTCL	Specifies operation edge for LTC signal. (0: Falling edge. 1: Rising edge) Note.3
24	PCSL	Specifies PCS signal input logic. (0: Negative logic. 1: Positive logic.)
25	DRL	Specifies +DR, -DR signal input logic. (0: Negative logic. 1: Positive logic.)
26	FLTR	1: Apply a noise filter for +EL, -EL, SD, ORG, ALM, or INP signals input. When a noise filter is applied, signal pulses shorter than the pulse width specified by RENV1.FTM1~0 are ignored.
27	DRF	1: Apply a noise filter on the +DR, -DR, or PE signals input. When a noise filter is applied, signals pulses shorter than 32 ms (CLK=19.6608MHz) are ignored.
28	DTMF	1: Turn OFF the direction change timer (0.2 ms) function.
29	INTM	1: Mask an INT signal output. (Interrupt circuit can be changed.)
30	PCSM	1: Make the PCS signal input as $\overline{\text{CSTA}}$ signal for its own axis only.
31	PMSK	1: Masks output pulses

Note1: When a deceleration stop (RENV1.ELM = "1") is specified to occur when the EL signal input turns ON, the motor will start the deceleration when the EL input is turned ON. Therefore, the motor will stop after a machine passing over the EL signal position. In this case, be careful to avoid collisions of mechanical systems.

Note 2: In the operation condition of FL constant start and FH constant start, motor stops immediately though setting to deceleration and stop is set.

Note 3: It may be latched at the change of setting according to status of LTC signal.

8-3-13. RENV2 register

This is a register for the Environment 2 settings. Specify the function of the general-purpose ports, 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	CSPO	P7M	P6M	P5M	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
MRST	IEND	ORM	EZL	EZD3	EZD2	EZD1	EZD0	PDIR	PINF	PIM1	PIM0	EDIR	EINF	EIM1	EIM0

Bits	Bit name	Description
1~0	P0M1~0	Specifies operation of P0/FUP terminals "00"b : General-purpose input "01"b : General-purpose output "10"b : Output FUP (acceleration) signal with negative logic. "11"b : Output FUP (acceleration) signal with positive logic.
3~2	P1M1~0	Specifies operation of P1/FDW terminals "00"b : General-purpose input "01"b : General-purpose output "10"b : Output FDW (decelerating) signal with negative logic. "11"b : Output FDW (decelerating) signal with positive logic.
5~4	P2M1~0	Specifies operation of P2/MVC terminal. "00"b : General-purpose input "01"b : General-purpose output "10"b : Output MVC (constant speed feeding) signal with negative logic. "11"b : Output MVC (constant speed feeding) signal with positive logic.
7~6	P3M1~0	Specifies operation of P3/CP1 terminals. "00"b : General-purpose input "01"b : General-purpose output "10"b : Output CP1 (Comparator 1 conditions satisfied) signal with negative logic. "11"b : Output CP1 (Comparator 1 conditions satisfied) signal with positive logic.
9~8	P4M1~0	Specifies operation of P4/CP2 terminals. "00"b : General-purpose input "01"b : General-purpose output "10"b : Output the CP2 (Comparator 2 conditions satisfied) signal with negative logic. "11"b : Output the CP2 (Comparator 2 conditions satisfied) signal with positive logic.
10	P5M	Specifies operation of P5 terminals. 0: General-purpose input 1: General-purpose output
11	P6M	Specifies operation of P6 terminals. 0: General-purpose input 1: General-purpose output.
12	P7M	Specifies operation of P7 terminals. 0: General-purpose input 1: General-purpose output
13	CSPO	1: When RMD/MSP0 = 1, the LSI will output a $\overline{\text{CSTA}}$ signal when the motor is 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.)
17~16	EIM1~0	Specifies the EA/EB input operation. "00"b : Multiply a 90-degree phase difference by 1 (Count up when the EA input phase is ahead.) "01"b : Multiply a 90-degree phase difference by 2 (Count up when the EA input phase is ahead.) "10"b : Multiply a 90-degree phase difference by 4 (Count up when EA input phase is ahead.) "11"b : Count up when the EA signal rises, count down when the EB signal rises.
18	EINF	1: Apply a noise filter to EA/EB/EZ input.

Bits	Bit name	Description
		Pulse inputs less than 3 CLK signal cycles long are ignores.
19	EDIR	1: Reverse a counting direction of EA/EB inputs.
21~20	PIM1~0	Specifies PA/PB input operation. "00"b : Multiply a 90-degree phase difference by 1 (Count up when the PA input phase is ahead.) "01"b : Multiply a 90-degree phase difference by 2 (Count up when the PA input phase is ahead.) "10"b : Multiply a 90-degree phase difference by 4 (Count up when PA input phase is ahead.) "11"b : Count up when the PA signal rises, count down when the PB signal rises.
22	PINF	1: Applies a noise filter to PA/PB input. Pulse inputs less than 3 CLK signal cycles long are ignored.
23	PDIR	1: Reverse the counting direction of the PA/PB inputs.
27~24	EZD3~0	Specifies an EZ count value used for origin return. "0000"b (1st time) ~ "1111"b (16th time)
28	EZL	Specifies EZ signal input logic. (0: Falling edge. 1: Rising edge.)
29	ORM	Select an origin return method. 0: Origin 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: Origin return operation 1 - When the LSI is feeding at constant speed, after the ORG input turns from OFF to ON it will stop immediately by finishing counting the specified number of the EZ signals. When the LSI is feeding at high speed, it will decelerate by turning the ORG input from OFF to ON and then immediately stop by finishing counting the specified number of the EZ signals. - COUNTER reset timing: When finishing counting the specified number of 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	MRST	Reading MSTSW.SENI bit, MSTSW.SEOR bit, REST register and RIST register Controls stop function to reset automatically. 0: Disabled (resets automatically) 1: Enabled (does not reset automatically) However, with serial bus I/F, this function is always fixed to "Enabled" (RENV2.MRST="1").

8-3-14. RENV3 register

This register is for environment setting 3. Specifies 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	LOF2	CU2L	C1RM	CU1R	COF1	CU1L	CU2H	CU1H	CIS2	CIS1

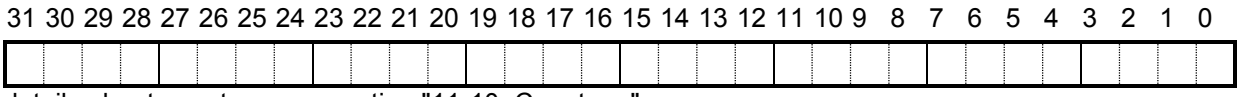
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	M614	SLCU	SLM1	SLM0	SYI1	SYI0	SYO3	SYO2	SYO1	SYO0

Bit	Bit name	Description
0	CIS1	Selects input counted by COUNTER 1 0: Output pulse 1: EA/EB input
1	CIS2	Selects input counted by COUNTER 2 0: EA/EB input 1: Output pulse
2	CU1H	1: Stops counting of COUNTER 1.
3	CU2H	1: Stops counting of COUNTER 2.
4	CU1L	1: Resets COUNTER 1 when latching the contents of COUNTER 1.
5	LOF1	1: Stop latching the contents of COUNTER 1 with the LTC signal input. (Effective only for software.)
6	CU1R	1: Latches (and resets) COUNTER 1 when an origin return operation is complete.
7	C1RM	1: Sets COUNTER 1 to ring counter operation using Comparator 1.
8	CU2L	1: Resets COUNTER 2 when latching the contents of COUNTER 2.
9	LOF2	1: Stop latching the contents of COUNTER 2 with the LTC signal input. (Effective only for software.)
10	CU2R	1: Latches (and resets) COUNTER 2 when an origin return operation is complete.
11	C2RM	1: Sets COUNTER 2 to ring counter operation using Comparator 2.
13~12	C1S1~0	Selects a comparison method for Comparator 1 “00”b : Turn the comparator function off. “01”b : RCMP1 data = Comparison counter “10”b : RCMP1 data > Comparison counter “11”b : RCMP1 data < Comparison counter
15~14	C2S1~0	Selects a comparison method for Comparator 2 “00”b : Turn the comparator function off. “01”b : RCMP2 data = Comparison counter “10”b : RCMP2 data > Comparison counter “11”b : RCMP2 data < Comparison counter
19~16	SYO3~0	Selects the output timing for the internal synchronous signal. “0001”b : When the Comparator 1 conditions are met. “0010”b : When the Comparator 2 conditions are met. “1000”b : When starting acceleration. “1001”b : When ending acceleration. “1010”b : When starting deceleration. “1011”b : When ending deceleration. Others: The internal synchronous signal is not output.
21~20	SYI1~0	Selects input used when start is triggered by internal synchronous signals. “00”b : Internal synchronous signal output by the X axis. “01”b : Internal synchronous signal output by the Y axis. “10”b : Internal synchronous signal output by the Z axis. “11”b : Internal synchronous signal output by the U axis
23~22	SLM1~ 0	Controls software limit function. "00"b : Stops software limit function. "01"b : Event interrupt occurs at position of software limit. (Motor does not stop.) "10"b : Motor stops immediately at position of software limit and error interrupt occurs. "11"b : Motor decelerates and stops at position of software limit and error interrupt occurs.

Bit	Bit name	Description
24	SLCU	Selects a counter to control software limit. 0 : COUNTER 1 1 : COUNTER 2
25	M614	Enables additional functions to PCL61x4 (monitoring of software limit and operation status of slave axis for linear interpolation) that are disabled in order to keep compatibility with PCL61x3. 0 : Disable 1 : Enabled
31~26	Not defined	(Always set to “0”.)

8-3-15. RCUN1 register

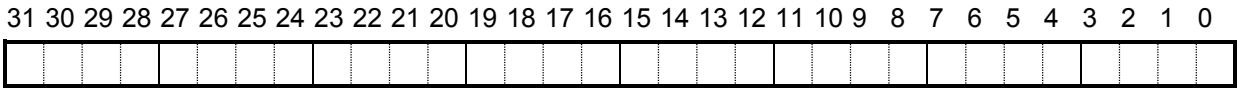
This register is used to set and read COUNTER 1.
Setting range: -2,147,483,648 ~ +2,147,483,647



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

8-3-16. RCUN2 register

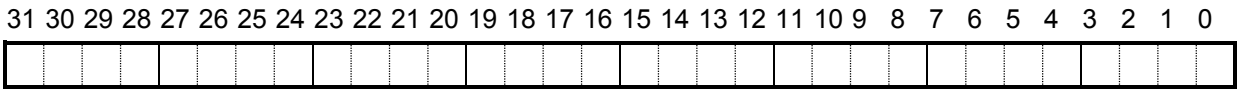
This register is used to set and read COUNTER 2.
Setting range: -2,147,483,648 ~ +2,147,483,647



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

8-3-17. RCMP1 register

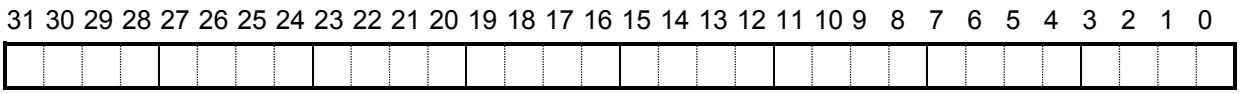
Specifies comparison data for Comparator 1.
Setting range: -2,147,483,648 ~ +2,147,483,647



For details about comparators, see section "11-11. Comparators."

8-3-18. RCMP2 register

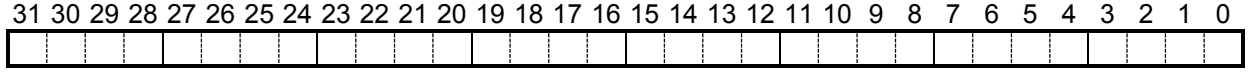
Specifies comparison data for Comparator 2.
Setting range: -2,147,483,648 ~ +2,147,483,647



For details about comparators, see section "11-11. Comparators."

8-3-19. RCMP3 register

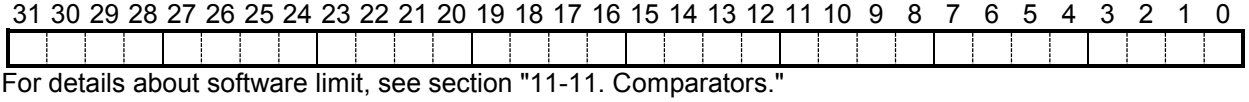
Specifies comparison data for (+) software limit
Setting range : -2,147,483,648 ~+2,147,483,647



For details about software limit, see section "11-11. Comparators."

8-3-20. RCMP4 register

Specifies comparison data for (-) software limit
Setting range : -2,147,483,648 ~ +2,147,483,647



For details about software limit, see section "11-11. Comparators."

8-3-21. RIRQ register

Specifies event interrupt factor.
Set bits correspond to contents that you want to enable event interrupts to “1”..

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 writing pre-register is enabled
2	IRUS	When 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 me.
8	IRLT	When latching the count value with an LTC signal input. (When RENV3.LOF1 = “1” and RENV3.LOF2 = “1”, an interrupt will not occur.)
9	IROL	When the ORG input is ON. (When RENV3.CU1R = “0” and RENV3.CU2R = “0”, an interrupt will not occur.)
10	IRSD	When the SD input is ON. (Even when the SD input is disabled by setting PRMD.MSDE = “0”, an interrupt will occur.)
11	IRDR	When the +DR(PA) and -DR(PB) input are changed. (When $\overline{PE}$ = H level, the interrupt will not occur.)
12	IRSA	When the $\overline{CSTA}$ input is ON.
31~13	Not defined	(Always set to 0.)

8-3-22. RLTC1 register

Latched data for COUNTER 1. (Read only.)
The contents of COUNTER 1 are copied when triggered by the LTC, an ORG input, or an LTCH (29h) command.
Data range: -2,147,483,648 ~ +2,147,483,647

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 counter data latch, see section "11-10. Counters."

8-3-23. RLTC2 register

Latched data for COUNTER 2 (Read only.)
The contents of COUNTER 2 are copied when triggered by the LTC, an ORG input, or an LTCH (29h)command.
Data range: 2,147,483,648 ~ +2,147,483,647

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 counter data latch, see section "11-10. Counters."

8-3-24. 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
3~0	CND3~0	Reports operation status. "0000"b : While stopping "0001"b : Waiting for DR input "0010"b : Waiting for $\overline{\text{CSTA}}$ input "0011"b : Waiting for an internal synchronous signal "0100"b : Waiting for another axis to stop. "0101"b : Waiting for a completion of ERC timer "0110"b : Waiting for a completion of direction change timer "1000"b : Waiting for PA/PB input. "1010"b : Feeding at FL constant speed. "1011"b : Accelerating "1100"b : Feeding at FH constant speed. "1101"b : Decelerating "1110"b : Waiting for INP input. Others : (while controlling start/stop) Note. "Other" status is changed to other status after some clock of CLK. Check the status by reading again.
4	SCD	Becomes "1" when the $\overline{\text{CSD}}$ signal input is ON.
5	SSTA	Becomes "1" when the $\overline{\text{CSTA}}$ signal input is turned ON.
6	SSTP	Becomes "1" when the $\overline{\text{CSTP}}$ signal input is turned ON.
7	SEMG	Becomes "1" when the $\overline{\text{CEMG}}$ signal input is turned ON.
8	SPCS	Becomes "1" when the PCS signal input is turned ON.
9	SERC	Becomes "1" when the ERC signal input is turned ON.
10	SEZ	Becomes "1" when the EZ signal input is turned ON.
11	SDRP	Becomes "1" when the +DR (PA) signal input is turned ON.
12	SDRM	Becomes "1" when the -DR (PB) signal input is turned ON.
13	SLTC	Becomes "1" when the LTC signal input is turned ON.
14	SDIN	Becomes "1" when the SD signal input is turned ON. (Status of SD input terminal.)
15	SINP	Becomes "1" when the INP signal input is turned ON.
16	SDIR	Operation direction (0: Positive direction, 1: Negative direction)
31~17	Not defined	(Always set to "0".)

8-3-25. REST register

Checks error interrupt factor.
The corresponding bit will be "1" when an error interrupt occurs.
When RENV2.MRST="0" is set, all bits are reset by reading REST register.
When RENV2.MRST="1" is set, only corresponding bit is reset by writing "1" to a bit that you want to reset.
In parallel I/F, you can select between RENV2.MRST="0" and RENV2.MRST="1".
In serial I/F, setting is fixed to RENV2.MRST="1". You cannot set to RENV2.MRST="0".

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	ESMS	ESPS	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	Stopped by an overflow occurred in the PA/PB input buffer counter.
7	ESEE	Stopped by an EA/EB input error occurred. (The motor does not stop)
8	ESPE	Stopped by a PA/PB input error occurred. (The motor does not stop)
9	ESPS	Stopped at detecting (+) software limit (Enable only when "RENV3.SLM1=1")
10	ESMS	Stopped at detecting (-) software limit (Enable only when "RENV3.SLM1=1")
31~11	Not defined	(Always set to "0".)

8-3-26. RIST register

Checks event interrupt factor.
When an event interrupt occurs, the bits corresponding to the factor will be “1”.
When RENV2.MRST=“0” is set, all bits are reset by reading REST register.
When RENV2.MRST=“1” is set, only corresponding bit is reset by writing “1” to a bit that you want to reset.
In parallel bus I/F, you can select between RENV2.MRST= “0” and “RENV2.MRST= “1”.
In serial bus I/F, setting is fixed to RENV2.MRST= “1”. You cannot set to “RENV2.MRST=“0”

This register is reset when read.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISMS	ISPS	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	When stopped normally.
1	ISNM	When writing to pre-register is enabled.
2	ISUS	When starting acceleration.
3	ISUE	When ending acceleration.
4	ISDS	When starting deceleration.
5	ISDE	When ending deceleration.
6	ISC1	When the comparator 1 conditions are met.
7	ISC2	When the comparator 2 condition are met.
8	ISLT	When the count value is latched by an LTC input.
9	ISOL	When the count value is latched by an ORG input.
10	ISSD	When the SD input turned ON.
11	ISPD	When the +DR (PA) input changed.
12	ISMD	When the -DR (PB) input changed.
13	ISSA	When the $\overline{\text{CSTP}}$ input turned ON.
14	ISPS	Detecting (+) software limit (Enable only when “RENV3.SLM1=0” and RENV3.SLM0=”1”.)
15	ISMS	Detecting (-) software limit (Enable only when “RENV3.SLM1=0” and RENV3.SLM0=”1”.)
31~16	Not defined	(Always set to “0”.)

8-3-27. RPLS register

Checks value of the positioning counter (number of pulses left for feeding). (Read only.)
At the start of positioning operation, this value will be an absolute value in the RMV register. Each pulse that is output will decrease this value by one.
Data range: 0 ~ 2,147,483,648

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

8-3-28. RSPD register

This register is used to check EZ count value and 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
13~0	AS13~0	Current speed can be read as a step value (same units as for RFL or RFH register). When stopped the value is “0”.
15~14	Not defined	(Always set to “0”.)
19~16	ECZ3~0	Count value of EZ signal input that is used for an origin return can be read.
31~20	Not defined	(Always set to “0”.)

8-3-29. RSDC register

Checks an 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 (PRMD.MSDP=”0”) is selected, “automatic setting value + RDP setting value” is expressed using 24 bits. The range is from -8,388,608 ~ +8,388,607.
When manual setting is selected (PRMD.MSDP = “1”), the range is 0 to ~16,777,215 and a fixed value that is equal to RDP register set value.

8-3-30. RGPM register

This register specifies specifications of general purpose input/output ports (GP0~15) used with serial bus I/F. In the case of parallel bus I/F, it is invalid.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GM15	GM14	GM13	GM12	GM11	GM10	GM9	GM8	GM7	GM6	GM5	GM4	GM3	GM2	GM1	GM0

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
15~0	GM15~0	Selects from input and output as specifications of terminals GP15~0. (0: Input port, 1: Output port)
31~16	Not defined	(Always set to "0".)

8-3-31. RGPD register

This register monitors status of terminals (GP0 to 15) of general input/output ports used with serial bus I/F and sets general purpose output ports.

When read, it monitors status of terminals (GP15 to 0) regardless of setting as input port or output port. When written, it changes the status of terminals set as output ports. (With parallel bus I/F, it is invalid.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GD15	GD14	GD13	GD12	GD11	GD10	GD9	GD8	GD7	GD6	GD5	GD4	GD3	GD2	GD1	GD0

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
15~0	GD15~0	When read, monitors status of terminals (GP15~0). (0: L level, 1: H level) When written, sets output ports of GP15~0. (0: L level, 1: H level)
31~16	Not defined	(Always set to "0".)

9. Operation Mode

Specifies the basic operation mode using PRMD.MOD6~0.

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	(+) direction
08h	Continuous operation from a command	(-) direction

Operation is stopped 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 should 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	(+) direction when PRMV ≥ “0” (-)direction when PRMV < “0”
47h	Timer operation (PRMV≥”0”)	(+) direction (DIR = H). However, pulse output is masked.

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

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

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

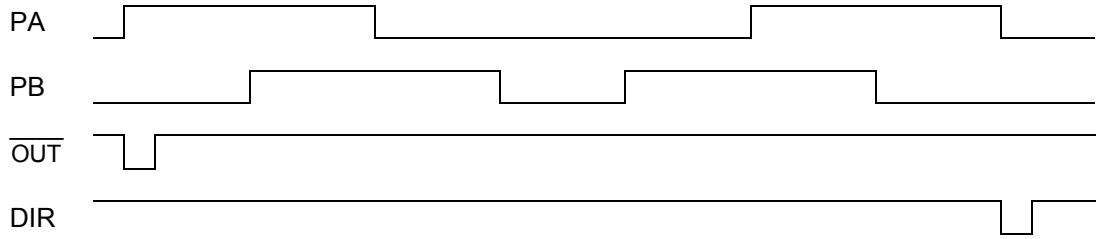
This mode allows the internal operation time to be used as a timer.
The internal effect of this operation is identical to the positioning operation. However, any pulses are not output and they are masked. (The counter does not count.)
Internal operation time using the constant speed start command will be a product of the frequency of the output pulses and the RMV register. (Ex.: When the frequency is 1000 pps and the RMV register set to 120 pulses, the internal operation time will be 120 msec.)
Write a positive number (1 ~ 2,147,483,647) into the PRMV register. Negative numbers are treated as unsigned positive numbers.
The +EL signal, -EL signal, SD input signal, and ALM signal are ignored. (These are always treated as OFF.)
The $\overline{\text{CSTP}}$ signal, and $\overline{\text{CEMG}}$ signals are enabled.
The direction change timer function is stopped.
Regardless of the RMD.MINP setting, an operation complete delay controlled by the INP signal will not occur.
In order to eliminate deviations in the internal operation time, set the PRMD.METM= “0” and select "end of cycle".

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

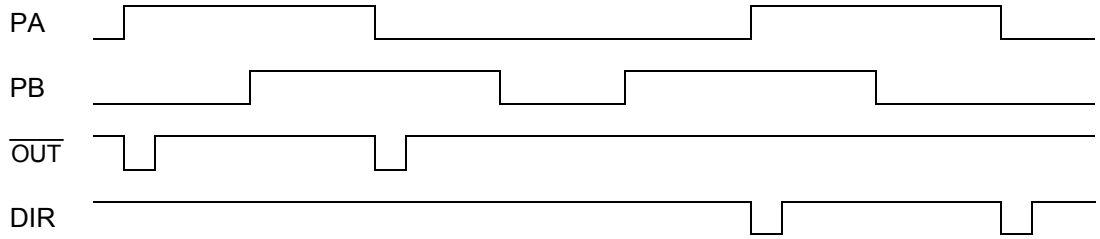
This mode is used to allow operations from a pulsar input.
In order to enable pulsar input, please set that $\overline{PE}$ terminal is L level and REMV2.POFF =0.
It is also possible to apply a noise filter to $\overline{PE}$ signal.
After writing a start command, when a pulsar signal is input, the LSI will output pulses to the OUT terminal.
Use STAFL (50h) command or STAFH (51h) command as a start command.
Input pulsar signals to the PA and PB terminals. The input specification can be selected from the four possibilities below by setting to RENV2.PIM1~0.

- Supply a 90-degree phase difference signal (1x, 2x, or 4x).
- Supply (+) pulse and (-) pulse (Two-pulse input).

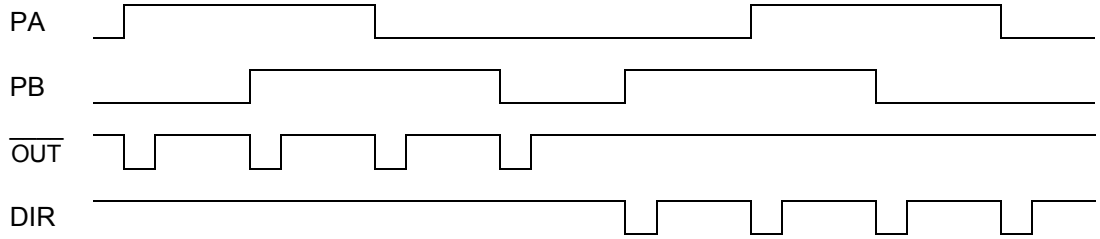
Shown below are diagrams of the operation timing. (RENV1.PMD = “100”b --- When outputting 2 pulses)
1) When using 90-degree phase difference signals and 1x input (RENV2.PIM = “00”b)



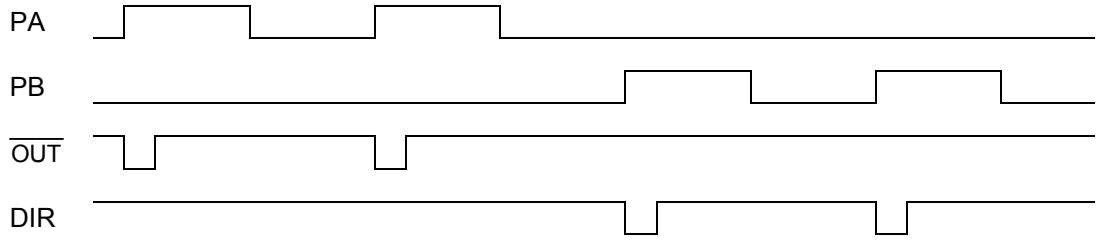
2) When using 90-degree phase difference signals and 2x input (RENV2.PIM = “01”b)



3) When using 90-degree phase difference signals and 4x input (RENV2.PIM = “10”b)



4) When using two pulse input. (RENV2.PIM = “11”b)

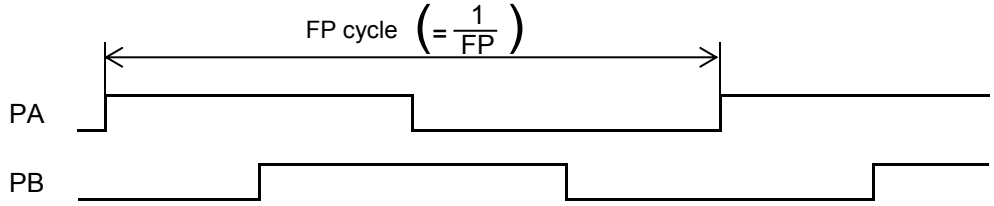


The pulsar input mode is triggered by STAFL (50h) command or STAFH (51h) command.
Pulsar input causes the LSI to output pulses with some pulses from the FL speed or FH speed pulse outputs being omitted. Therefore, there may be a difference in the timing between the pulsar input and output pulses, up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

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

$$FP < \frac{\text{Setting speed}}{\text{input I/F multiply value}}$$

Example: When the pulse input setting speed is 1000 pps with a 90-degree phase difference and a 2x input multiply value, 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 " FP cycle" above.

<Setting relationship of PA/PB input>

Specify the PA/PB input <RENV2.PIM1~0 (bits 21~20)> "00"b : 90-degree phase difference, 1x "10"b : 90-degree phase difference, 4x "01"b : 90-degree phase difference, 2x "11"b : 2 pulse input	[RENV2] (WRITE) 23 16 - - n n - - - -
Specify the PA/PB input count direction <RENV2.PDIR (bit 23)> 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 <RENV2.POFF (bit 15)> 0: Enable PA/PB input 1: Disable PA/PB input.	[RENV2] (WRITE) 15 8 n - - - - - - -
Apply an input noise filter to +DR, -DR, $\overline{PE}$ <RENV1.DRF (bit 27)> 1: Apply a noise filter to +DR, -DR and $\overline{PE}$ input By setting a noise filter, the LSI ignores signals shorter than 32 msec.	[RENV1] (WRITE) 31 24 - - - - n - - -
Reading operation status <RSTS.CND3~0(bits 3~0)> "1000"b : Wait for PA/ PB input.	[RSTS] (READ) 7 0 - - - - n n n n
Reading PA/PB input error <REST.ESPE (bit 8)> 1: A PA/PB input error occurs.	[REST] (READ) 15 8 0 0 0 0 0 0 0 n
Reading PA/PB input buffer counter status <REST.ESP0 (bit 6)> 1: An overflow occurs.	[REST] (READ) 7 0 - n - - - - - -

* In the descriptions in the right hand column, "n" refers to a bit position. "0" refers to a bit position 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 (incremental position)	Feeds in a (+) direction when PRMV ≥ 0. Feeds in a (-) 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 RENV2.PDIR.

PA/PB input method	PDIR	Feed direction	PA/PB input
90-degree phase difference signal (1x, 2x, and 4x)	0	(+) direction	When the PA phase leads the PB phase.
		(-) direction	When the PB phase leads the PA phase.
	1	(+) direction	When the PB phase leads the PA phase.
		(-) direction	When the PA phase leads the PB phase.
(+) pulse and (-) pulse	0	(+) direction	PA input rising edge.
		(-) direction	PB input rising edge.
	1	(+) direction	PB input rising edge.
		(-) direction	PA input rising edge.

The LSI stops operation when the EL signal in the current feed direction is turned ON. But the LSI 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 STOP (49h) command.

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

The positioning operation is synchronized with the pulsar input by using the PRMV register as incremental position data.
The feed direction is determined by the sign in the PRMV register.
At the start, the LSI loads the RMV register into the positioning counter.
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 will stop and another PA/ PB input will be ignored.
When you set PRMV register to “0” and start the positioning operation, the LSI will stop movement 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 L level.
After writing a start command, when a +DR and -DR signal is input, the LSI will output pulses to the OUT terminal.
Set the RENVI register to specify the output logic of the +DR and -DR signal. The $\overline{INT}$ signal can be set to send an output when +DR and –DR signals input are changed. If $\overline{PE}$ = L level, the LSI will output pulses regardless of the operation mode selected.
The RSTS register can be used to check the operating status and monitor the +DR and –DR signals.
It is also possible to apply a noise filter to the +DR, -DR and $\overline{PE}$ inputs.

Set the input logic of the +DR and -DR signals <RENV1.DRL (bit 25)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 31 24 - - - - - - n -
Applying a noise filter to +DR, -DR and $\overline{PE}$ inputs <RENV1.DRF (bit 27)> 1: Apply a noise filter to +DR, -DR and $\overline{PE}$ inputs When a noise filter is applied, pulses shorter than 32 ms will be ignored.	[RENV1] (WRITE) 31 24 - - - - n - - -
Setting an event interrupt factor <RIRQ.IRDR (bit 11)> 1: Output the $\overline{INT}$ signal when +DR and –DR signal input changed.	[RIRQ] (WRITE) 15 8 0 0 0 0 n - - -
Reading the event interrupt cause <RIST.ISPD (bit 11) and RIST.ISMD (bit 12)> ISPD = “1” : When the +DR signal input changes. ISMD = “1” : When the -DR signal input changes.	[RIST] (READ) 15 8 0 0 - n n - - -
Read operation status <RSTS.CND3~0 (bits 3~0)> “0001”b : Waiting for a DR input	[RSTS] (READ) 7 0 - - - - n n n n
Reading the +DR and -DR signals <RSTS.SDRP (bit 11) and RSTS.SDRM (bit 12)> 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 operations

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

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

This mode is used to operate a motor only when the DR signal is ON.
After writing a start command, turn the +DR signal ON to feed the motor in (+) direction, turn the -DR signal ON to feed the motor in (-) direction, using a specified speed pattern.
By turning ON an EL signal for the feed direction, the motor will stop. However, the motor can feed in the reverse direction.
An error interrupt ($\overline{INT}$output) will not occur when the motor is stopped by the EL signal.
To end this operation mode, write STOP (49h) command.
When STAD (52h) command and STAUD (53h) command are written, the motor will decelerate and stop when the DR signal turns OFF. If the DR signal for reverse direction turns ON while decelerating, the motor will decelerate and stop. Then it will resume in the opposite direction.

[Setting example]
1) Bring the $\overline{PE}$ signal input “L”.
2) Specify PRFL, PRFH, PRUR, PRDR, and PRMG (speed setting).
3) Set PRMD.MOD 6~0 to “0000010”b
4) Write STAFL (50h), STAFH (51h), STAD (52h) or STAUD(53h) commands.
RSTS.CND3~0 will wait for “0001”b (Wait for DR input).
In this condition, when turning +DR or -DR input terminal ON, The motor will rotate 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 signal turning ON.
When a start command is executed, the LSI waits for DR signal input.
When a stop command is executed, waiting for DR signals stops.
When DR signal is ON after the start, the value in RMV register is leaded into the PRLS register, and the LSI starts outputting pulses.
Value in PRLS register downs count per pulse output and when the value becomes “0”, the LSI waits for DR signal input.
Even though DR signal turns OFF in operation or turns ON again, operation is not affected.
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, the motor will stop. However, the motor can be feed in the reverse direction.
An error interrupt ($\overline{\text{INT}}$output) will not occur when the motor is stopped by the EL signal.

9-5. Origin return operation mode

Origin return operation varies with PRMD.MOD6~0, RENV2.ORM and the type of start command, as follows:

MOD	ORM	COMB	Operation description
10h	0	50h	Feeds in a positive direction at a FL constant speed and stops immediately when the ORG input changes from OFF to ON.
		51h	Feeds in a positive direction at a FH constant 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 and starts deceleration when the ORG input changes from OFF to ON. When the LSI has decelerated to the FL speed, it stops feeding pulses. Also, if the LSI completes its deceleration to FL speed by a signal from the SD input before the ORG input changes, the LSI will stop immediately when the ORG input changes from OFF to ON.
	1	50h	Feeds in a positive direction at a FL constant speed. After the ORG input changes from OFF to ON, the LSI stops operation immediately after counting the specified number of EZ input signals.
		51h	Feeds in a positive direction at a FH constant speed. After the ORG input changes from OFF to ON, the LSI 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 movement stops. Also, if the LSI completes its deceleration to FL speed by a signal from the SD input before the ORG input changes, the movement 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 FL constant speed and stops immediately when the ORG input changes from OFF to ON.
		51h	Feeds in a negative direction at a FH constant 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 and starts deceleration when the ORG input changes from OFF to ON. When the LSI has decelerated to the FL speed, it stops feeding pulses. Also, if the LSI completes its deceleration to FL speed by a signal from the SD input before the ORG input changes, the LSI will stop immediately when the ORG input changes from OFF to ON.
	1	50h	Feeds in a negative direction at a FL constant speed. After the ORG input changes from OFF to ON, the LSI stops immediately after counting the specified number of EZ input signals.
		51h	Feeds in a negative direction at a FH constant speed. After the ORG input changes from OFF to ON, the movement 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 and starts deceleration when the ORG input changes from OFF to ON. After counting the specified number of EZ inputs, the movement LSI stops. Also, if the LSI completes its deceleration to FL speed by a signal from the SD input before the ORG input changes, the LSI 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 origin position operation uses the ORG or EZ signals.
Input logic of the ORG signal is specified in RENV1.ORGL. This register's terminal status can be monitored with an SSTSW.SORG.
A noise filter can be applied to ORG signals by setting RENV. FLTR.
Input logic and the number for EZ to count up of the EZ input signal are specified in RENV2.EZL and RENV2.EZD. Status of this terminal can be monitored in the RSTS.SEZ.
An input noise filter can be applied to EZ signal by setting the RENV2.EINF.

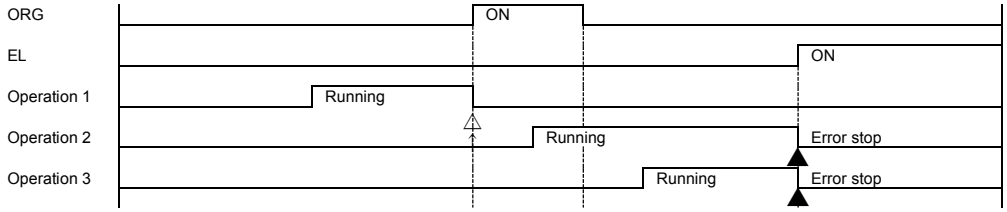
Selection of the origin return operation mode <RENV2.ORM (bit 29)> 0: Use only the ORG signal. 1: Use the ORG signal and EZ signals.	[RENV2] (WRITE) 31 24 - - n - - - - -
Reading the ORG signal <SSTSW.SORG (bit 14)> 0: Turn OFF the ORG signal. 1: Turn ON the ORG signal.	[SSTSW] (READ) 15 8 - n - - - - - -
Select input logic of the ORG signal <RENV1.ORGL (bit 7)> 0: Negative logic. 1: Positive logic.	[RENV1] (WRITE) 7 0 n - - - - - - -
Apply an input noise filter to ORG and SD <RENV1.FLTR (bit 26)> 1: Apply a noise filter to the +EL, -EL, SD, ORG, ALM, and INP inputs. When the filter is applied, signals which are shorter than pulse width set in RENV1.FTM1~0 will be ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -
Specify a time constant for input filter <RENV1.FTM1~0 (bits 21~20)> “00”b : 3.2us “10”b : 200 us “01”b : 25 us “11”b : 1.6 ms	[RENV1] (WRITE) 23 16 - - n n - - - -
Reading the EZ signal <RSTS .SEZ (bit 10)> 0: Turn OFF the EZ signal. 1: Turn ON the EZ signal.	[RSTS] (READ) 15 8 - - - - - n - -
Set the input logic for the EZ signal <RENV2.EZL (bit 28)> 0: Falling edge. 1: Rising edge	[RENV2] (WRITE) 31 24 0 - - n - - - -
Apply an noise filter to EA, EB, and EZ signals. <RENV2.EINF (bit 18)> 1: Apply a noise filter to these inputs. Signals that are shorter than a CLK 3 cycle will be ignored.	[RENV2] (WRITE) 23 16 - - - - - n - -
Specify an EZ count amount <RENV2.EZD3~0 (bits 27~24)> Specify the number of EZ pulses needed to qualify for an origin return completion. Specify the value (Number of pulses-1) in EZD3~0. Set values from 0 ~ 15.	[RENV2] (WRITE) 31 24 0 - - - n n n n

When an origin return is complete, the LSI can latch (and reset) the counter and output an ERC (deviation counter clear) signal.
The RENV3 register is used to set the basic origin return method and whether or not to reset the counter when the origin 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.”

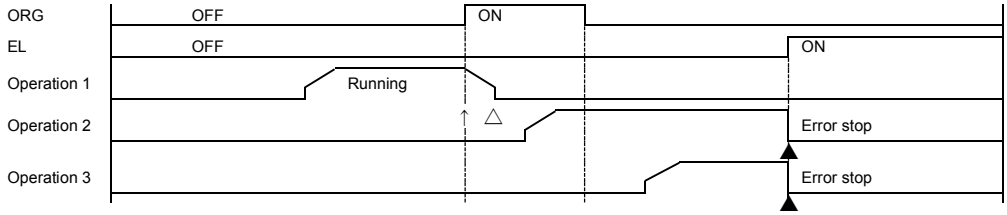
9-5-1. Origin return operation 0 (ORM = “0”)

△ : Timing when ERC signals are output with RENV1.EROR=“1”.
▲ : Timing when ERC signals are output with RENV1.EROR=“1”
↑ : Timing when Counter 1 (Counter 2) is latched (or reset) with RENV3.CU1R=“1” or RENV3.CU1R=“1”.

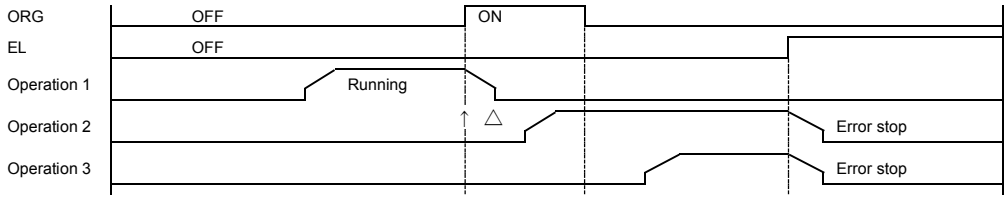
□Constant seed operation <Sensor: RENV1.ELM=“0” or “1”, ORG>
To output ERC signal when operation stops at the origin, set RENV1.EROR = “0”.
To reset a counter at the origin position, set RENV3.CU1R= “1” or RENV3.CU2R= “1”.



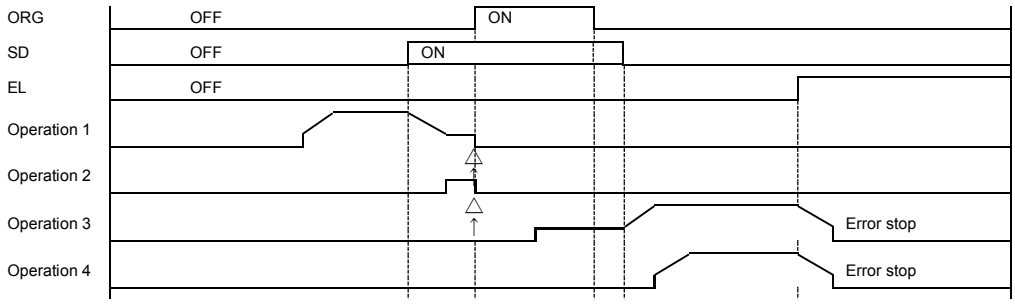
■High speed operation <Sensor: RENV1.ELM = “0”), ORG>
Even if the motor stops normally, it may not be at the origin position. However, COUNTER 2 (mechanical position) provides a reliable value.
If ERC signal is output when operation stops at the origin, an error occurs on a counter value. Set RENV1.EROR= “0”.
To reset a counter at the origin position, set RENV3.CU1R=“1” or RENV3.CU2R = “1”.



■ High speed operation <Sensor: EL (RENV1.ELM = “1”), ORG>
Even if the motor stops normally, it may not be at the origin position. However, COUNTER 2 (mechanical position) provides a reliable value.
If ERC signal is output when operation stops at the origin, an error occurs on a counter value. Set RENV1.EROR=“0”.
To reset a counter at the origin position, set RENV3.CU1R=“1” or RENV3.CU2R=“1”.
When operation decelerates and stops at EL signal, ERC signal is not output.

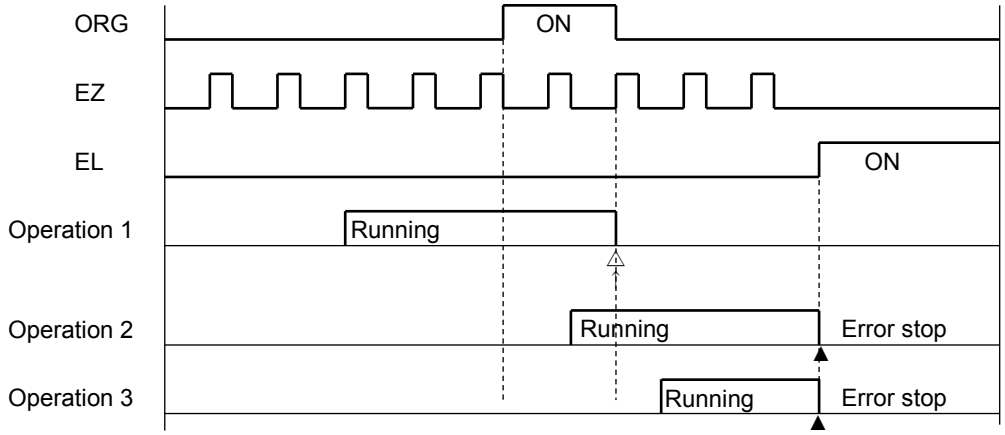


■ High speed operation <Sensor: EL (RENV1.ELM = “1”), SD (RENV1.SDM = “0”, RENV1.SDLT=“0”), ORG>
To output ERC signal when operation stops at the origin, set RENV1.EROR=“1”.
To reset a counter at the origin position, set RENV3.CU1R=“1” or RENV3.CU2R=”1”.

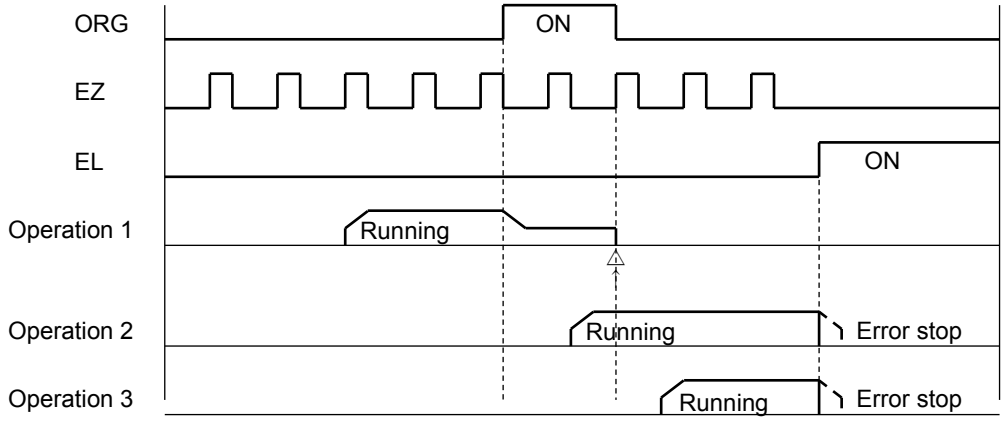


9-5-2. Origin return operation 1 (ORM=1)

□ Constant speed operation <Sensor: EL (ELM = 0), ORG, EZ (RENV2.EZD = “0001”b)>



■ High speed operation <Sensor: EL, ORG, EZ (RENV2.EZD = “0001”b)>



9-6. Linear interpolation operation

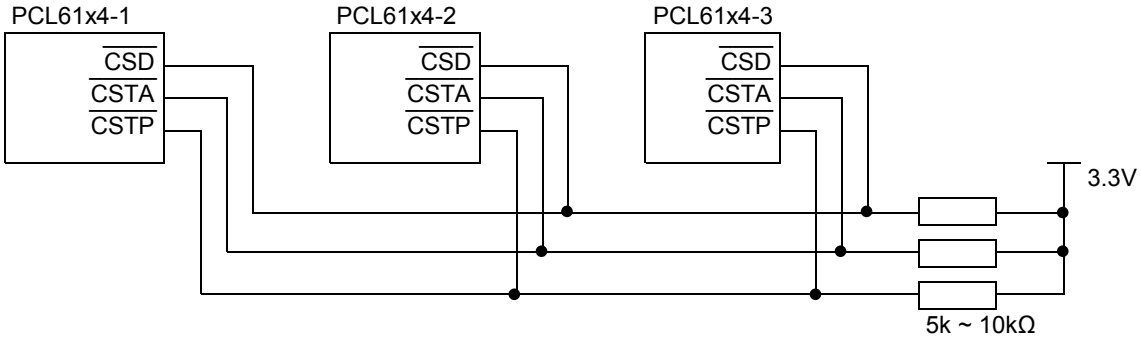
9-6-1. Outline of interpolation operation

Using one or more LSIs, you can operate linear interpolation operation.

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

Continuous linear interpolation operates on multi axes at a specified rate just like the linear interpolation, and be started and stopped with commands like continuous mode.
In contrast, the operation automatically stops after feeding specified amount in linear interpolation.

The linear interpolation circuit in the LSI interpolates between a dummy axis associated with each axis and its own axis.
By entering maximum feed amount data for each to every dummy axis, the LSIs 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 LSIs and have them interpolate for each other, connect $\overline{\text{CSD}}$, $\overline{\text{CSTA}}$, and $\overline{\text{CSTP}}$ terminals on each LSI to other same terminals on other LSI and provide a pull up resistor (5 k ~ 10 k-ohms) on VDD (3.3 V) for each signal line.
Even when performing interpolation within a single LSI, a pull up resistor is required for each terminal.



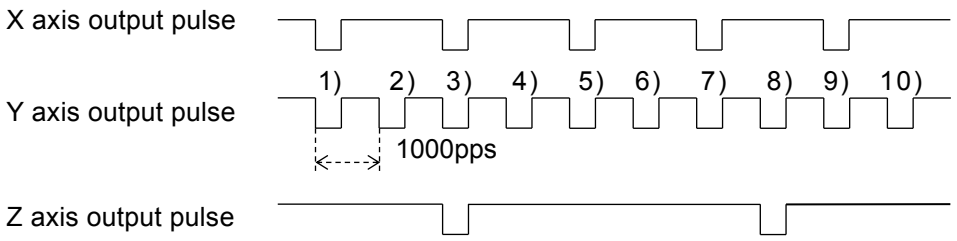
9-6-2. Interpolation procedures

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 PRMV register (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 PRMD.MSPE="1" and PRMD.MSPO on those axes to "1".
5) When you want to interpolate using acceleration/deceleration, set the PRMD.MCDE="1" and PRMD.MCDO ="1" for all the axes that will perform an interpolation.
6) When you want to perform an interpolation using only one LSI, specify the axis to interpolate in the upper byte (COMB1) when writing a start command.
   When you want to perform an interpolation using multiple LSIs, set PRMD.MSY1~0 ="01"b, on all the axes that will perform an interpolation. Then write a waiting for start command (waiting for a $\overline{\text{CSTA}}$input).

After setting all the axes that will perform an interpolation to wait start, write CMSTA (06h) command to any of these axes. 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.

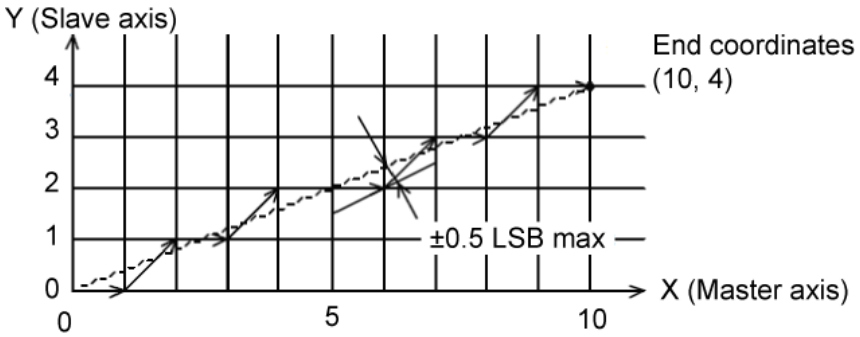
[Setting example] Use the settings below and write STAFH (0751h) command. The LSI will output pulses with the timing shown in the figure below.

Setting	X axis	Y axis	Z axis
PRMD.MOD6~0	63h	63h	63h
PRMV	5h	Ah	2h
Operation speed	1000 pps	1000 pps	1000 pps



[Precision of linear interpolation]

As shown in the figure on the right, linear interpolation is executed 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. It corresponds to the resolution of the mechanical system. (distance between squares in the figure on the right.)



9-6-3. Operation during interpolation

- Acceleration/deceleration operations
  In addition to constant speed operation, these LSIs can control accelerate/decelerate (linear acceleration or S-curve), and a ramp down point with an automatic setting is also available.
  However, the following restrictions are applied:
  1) The settings for PRMD.MSDP and PRMD.MADJ must be identical for all the axes that will perform an interpolation.
  2) If you want to use the manual setting (PRMD.MSDP =“1”) for the ramp down point, enter manual setting value for the longest feeding 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 on error by the $\overline{\text{CSTP}}$ terminal. Axes that did not encounter an error will show RSTS.ESSP = 1 when read. This allows you to identify the axis that had an error.

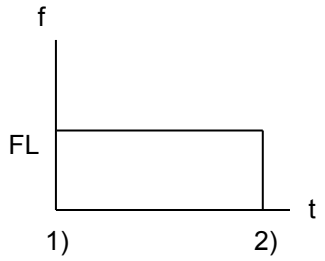
- SD input
  When SD input is enabled (PRMD.MSDE=“1”) by processing the $\overline{\text{CSD}}$ terminal, and if the SD input turns ON on any of the axes, all axes will decelerate or decelerate and stop.

- Continuous interpolation
  The LSI 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 pre-register is shown in section “11-12-1. Start triggered by another axis stopping.”

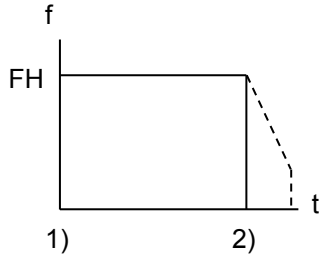
- Operation status monitor
  If you make additional features to PCL61x4 enabled (RENV3.M614=“1”), you can monitor operation status of slave axis (RSTS.CND3~0) as well as master axis.

10. Speed patterns

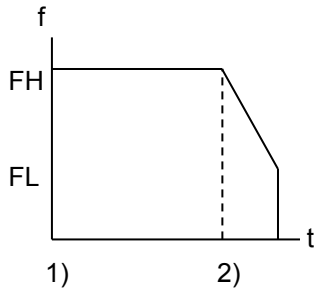
10-1. Speed patterns

<table>
<tr><th>Speed pattern</th><th>Continuous mode</th><th>Positioning operation mode</th></tr>
<tr><td rowspan="2">FL constant speed operation
 
</td><td>1) Write STAFL (50h) command.

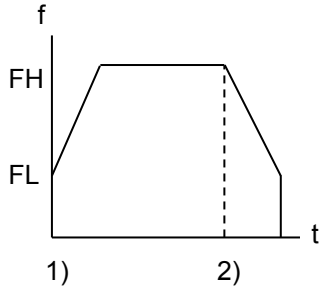
2) Stop feeding by writing STOP (49h) or SDSTP (4Ah) command.</td><td>1) Write STAFL (50h) command.

2) Stop feeding when PRLS=“0”, or by writing STOP (49h) command or SDSTP (4Ah) command.</td></tr>
<tr><td colspan="2">*FL constant speed operation does not perform acceleration / deceleration operation by +EL, -EL, SD, ALM, CSTP, ORG, +DR, -DR signals input.</td></tr>
<tr><td rowspan="2">FH constant speed operation
 
</td><td>1) Write STAFH (51h) command.

2) Stop feeding by writing STOP (49h) command.</td><td>1) Write STAFH (51h) command.

2) Stop feeding when PRLS=“0”, or by writing STOP (49h) command.</td></tr>
<tr><td colspan="2">* When SDSTP (4Ah) command is written to the register, movement decelerates and stops.
* FH constant speed operation does not perform acceleration / deceleration operation by +EL, -EL, SD, ALM, CSTP, ORG, +DR, -DR signals input.</td></tr>
<tr><td rowspan="3">High speed operation 1)
 
</td><td>1) STAD (52h) command.

2) Start deceleration by writing SDSTP (4Ah) command.</td><td>1) Write STAD (52D) command.

2) Start deceleration when a ramping-down point is reached or by writing SDSTP (4Ah) command.</td></tr>
<tr><td>* When the immediate stop command (49h) is written to the register, the LSI immediately stops operation.</td><td>* Motor stops immediately when RPLS=“0” or “RMD.MSDP=“0) (auto setting) an “RDP=“0”.
(The motor operates as manual setting.)</td></tr>
<tr><td colspan="2">* High speed operation 1) performs acceleration / deceleration operation by external signal input.</td></tr>
<tr><td rowspan="3">High speed operation 2)
 
</td><td>1) Write STAUD (53h) command.

2) Start deceleration by writing SDSTP (4Ah) command.</td><td>1) Write STAD (52h) command.

2) Start deceleration when a ramping-down point is reached or by writing SDSTP (4Ah) command.</td></tr>
<tr><td>* When STOP (49h) command is written, the LSI immediately stops operation.</td><td>* Motor stops immediately when RPLS=“0” or “RMD.MSDP=“0) (auto setting) an “RDP=“0”.</td></tr>
<tr><td colspan="2">* High speed operation 2) performs acceleration / deceleration operation by +EL, -EL, SD, ALM, CSTP, ORG, +DR and -DR signal input.</td></tr>
</table>

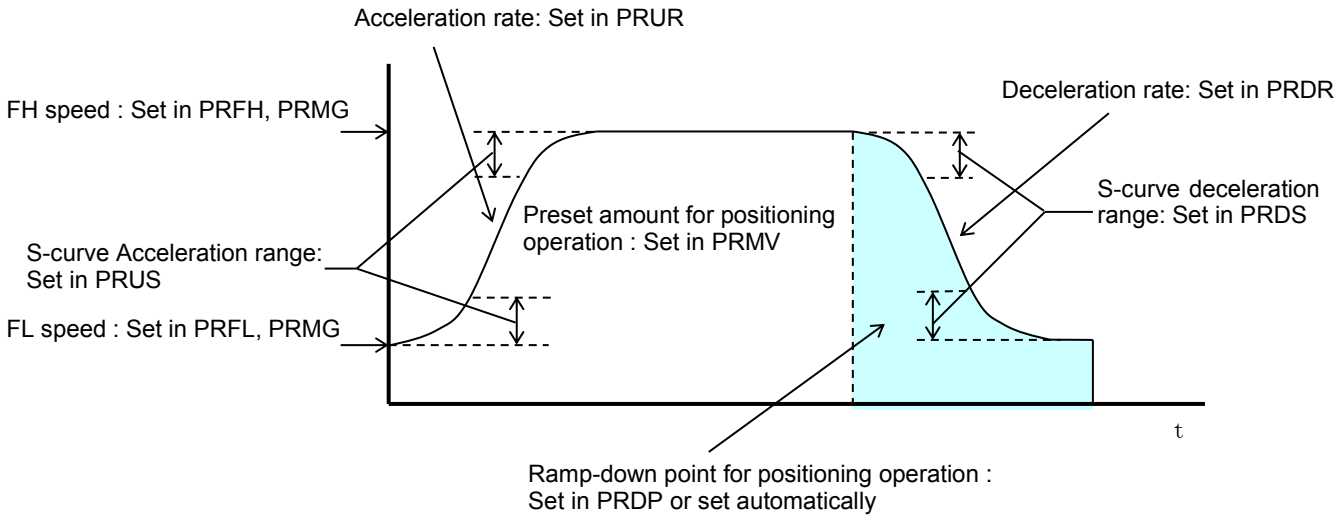
10-2. Speed pattern settings

Speed pattern is specified by using 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 again.

Pre-register	Description	Bit length setting range	Setting range	Register
PRMV	Positioning amount	32	-2,147,483,648 ~ +2,147,483,647 (80000000h) (7FFFFFFFh)	RMV
PRFL	Start speed	14	1 ~ 16,383 (3FFFh)	RFL
PRFH	Operation speed	14	1 ~ 16,383 (3FFFh)	RFH
PRUR	Acceleration rate	16	1 ~ 65,535 (FFFFh)	RUR
PRDR	Deceleration rate Note 1	16	0 ~ 65,535 (FFFFh)	RDR
PRMG	Speed magnification rate	12	1 ~ 4,095 (0FFFh)	RMG
PRDP	Ramping-down point	24	0 ~ 16,777,215 (FFFFFFh)	RDP
PRUS	S-curve acceleration range	13	0 ~ 8,191 (1FFFh)	RUS
PRDS	S-curve deceleration range	13	0 ~ 8,191 (1FFFh)	RDS

Note 1: When PRDR =“0”, the deceleration rate will be the value set in the PRUR.

[Register setting data used for acceleration and deceleration]



Notes. The times such as acceleration and deceleration calculated by the following formula are approximate values of normal stop in continuous operation mode. The effort of approximate value is less than 1%. Error may be bigger than 1% in positioning mode.

- PRFL: FL speed setting register (14-bit)
  Specify speed for FL constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 ~ 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 speed for FH constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 ~ 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 (16-bit)
  Specify acceleration characteristic for high speed operations (acceleration/deceleration operations), in the range of 1 ~ 65,535(FFFFh)
  Relationship between the value entered and the acceleration time will be as follows:

  1) Linear acceleration (PRMD.MSMD = “0”)

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

  2) S-curve without a linear range (PRMD.MSMD=“1” and PRUS =“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 (PRMD.MSMD=“1” and PRUS >“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 (16-bit)
  Normally, specify the deceleration characteristics for high speed operations (acceleration/deceleration operations) in the range of 1 ~ 65,535(FFFFh).
  To select the ramp down point auto setting (PRMD.MSDP = 0), set the PRDR register the same as PRUR register setting, or PRDR=“0”.
  When PRDR = 0, the deceleration rate will be the value placed in the PRUR register.

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

  1) Linear deceleration (PRMD.MSMD =“0”)

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

  2) S-curve deceleration without a linear range (PRMD.MSMD=“1” and PRDS =“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 (PRMD.MSMD=“1” and PRDS >“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 register settings and the speed, in the range of 1 ~ 4,095 (0FFFh). As the magnification rate is increased, the speed setting units will tend to be approximations.
  Normally set the magnification rate to an appropriate small rate to fit for output speed range.
  The relationship between the value entered and the magnification rate is as follows.

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

[Examples of magnification rate setting 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 ~ 4,914.9	59 (003Bh)	20	20 ~ 327,660
2399 (095Fh)	0.5	0.5 ~ 8,191.5	23 (0017h)	50	50 ~ 819,150
1199 (04AFh)	1	1 ~ 16.383	11 (000Bh)	100	100 ~ 1,638,300
599 (0257h)	2	2 ~ 32,766	5 (0005h)	200	200 ~ 3,276,600
239 (00EFh)	5	5 ~ 81,915	2 (0002h)	400	400 ~ 6,553,200
119 (0077h)	10	10 ~ 163,830	1 (0001h)	600	600 ~ 9,829,800

The maximum output speed of this LSI 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 LSI will output 14.999 Mpps.

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

Specify the value used to determine a ramping-down point for positioning operations that include acceleration and deceleration.
The meaning of the value specified in the PRDP register varies according to setting of "ramping-down point setting method" (PRMD.MSDP).

<When set to manual (PRMD.MSDP="1") for ramping-down point>
Set the number of pulses at ramping-down point in the range of 0 ~ 16,777,215 (FFFFFFh).
When the (PRDP set value) ≥ (Number of residual pulses), the LSI will start decelerating.

Note: In order to obtain correct manual setting value of ramping-down point, 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 LSI cannot calculate the manual setting value of ramping-down point.
Therefore, when ramping-down point is set manual set (PRMD.MSDP=1), turn OFF the FH correction function (PRMD.MADJ=1).
Alternatively after manual FH correction, calculate the following equation using corrected maximum speed.

The optimum value of the ramping down point can be as follows.
1) Linear deceleration (PRMD.MSMD="0")

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

2) S-curve deceleration without a linear range (PRMD.MSMD="1" and the PRDS ="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 (PRMD.MSMD="1" and the PRDS >"0")

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

Deceleration is started at the point that the (positioning counter value) ≤ (PRDP 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 motor will feed at FL constant speed after decelerating.

<In the case of setting that ramping-down point is set automatically (PRMD.MSDP =“0”)>
This is an offset value for the automatically set ramping-down point. Set in the range of -8,388,608 (800000h) ~ 8,388,607 (7FFFFFh).
When the offset value is a positive number, the motor 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 “0”.

- 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 ~ 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 “0” 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 ~ 8,191 (1FFFh).
The S-curve acceleration range S_{SD} will be calculated from the value placed in PRMG.

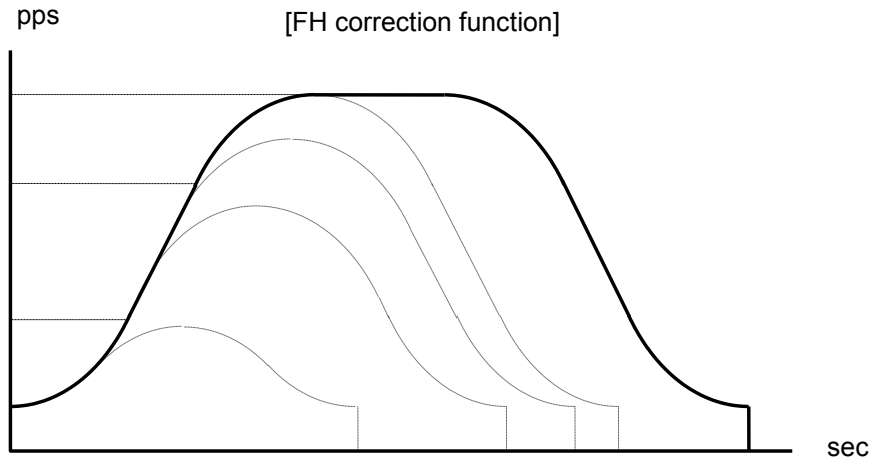
$$S_{SD}\ [pps] = PRDS \times \frac{\text{Reference clock frequency [Hz]}}{(PRMG + 1) \times 16384}$$

In other words, speeds between the FH speed and (FH speed - S_{SD}), and between (FL speed + S_{SD}) and the FL speed, will be S-curve deceleration operations. Intermediate speeds will use linear deceleration.
However, if “0” 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 function

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

However, the ramping 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 ramping down point needs to be changed to a manual setting (PRMD.MSDP=“1”). In order to obtain the correct manual setting value of ramping-down point, you have to know the maximum speed.
Therefore, you have to turn OFF the FH correction function (PRMD.MADJ=“1”) and manually correct the FH value (PRFH register).



Automatic correction of the maximum speed for by feeding amount

< To execute FH correction manually>

1) Linear acceleration/deceleration speed (PRMD.MSMD="0"
When

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

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

2) S-curve acceleration without linear acceleration (PRMD.MSMD="1", PRUS="0" and PRDS="0")
When

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

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

3) S-curve acceleration/deceleration with linear acceleration/deceleration (PRMD.MSMD="1" and PRUS> "0", or PRDS > "0")
(3)-1. When PRUS = PRDS

(i) Make a linear acceleration range smaller
When

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

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

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

(ii) Eliminate the linear acceleration/deceleration range
When

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

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

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

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) Make a linear acceleration/deceleration range smaller

When

$$PRMV \leq \frac{(PRFH+PRFL)x\{(PRFH-PRFL)x(PRUR+PRDR+2)+2xPRUSx(PRUR+1)+2xPRDSx(PRDR+1)\}}{(PRMG + 1) \times 16384}$$

and

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

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

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 make a linear acceleration section smaller.

When

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

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

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

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

However, $A = PRUS \times (PRUR + 1)$,

$B = \{(PRMG+1)x16384xPRMV-2xAxPRFL+(PRUR+2xPRDR+3)xPRFL^2\}x(PRUR+2xPRDR+3)$

(iii) Eliminate the linear acceleration/deceleration range

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

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

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

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) Make a linear acceleration/deceleration range smaller

When

$$PRMV \leq \frac{(PRFH+PRFL)x\{(PRFH-PRFL)x(PRUR+PRDR+2)+2xPRUSx(PRUR+1)+2xPRDSx(PRDR+1)\}}{(PRMG+1)\ x\ 16384}$$

and

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

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

However, A = PRUS x (PRUR + 1) + PRDS x (PRDR + 1)

B= {(PRMG + 1) x 16384 x PRMV - 2 x A x PRFL + (PRUR + PRDR + 2) x $PRFL^{2)}$ x (PRUR + PRDR + 2)

(ii) Eliminate the linear acceleration section and make a linear deceleration range smaller.

When

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

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

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

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

However, A = PRDS x (PRDR + 1),

B= {(PRMG+1)x16384xPRMV-2xAxPRFL+(2xPRUR+PRDR+3)x$PRFL^2$}x(2xPRUR+PRDR+3)

(iii) Eliminate the linear acceleration/deceleration range

When

$$PRMV \leq \frac{(PRDS + PRFL)\ x\ PRDS\ x\ (PRUR + PRDR + 2)\ x\ 8}{(PRMG + 1)\ x\ 16384}$$

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

$$PRFH \leq \sqrt{\frac{(PRMG + 1)\ x\ 16384\ x\ PRMV}{(PRUR + PRDR + 2)\ x2} + PRFL^2}$$

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/decel time = 300 ms, and linear acceleration/deceleration is selected.

1) Select the 10x mode for magnification 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 magnification rate is set to the 10x mode.
   PRFL = 1 (0001h)

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

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

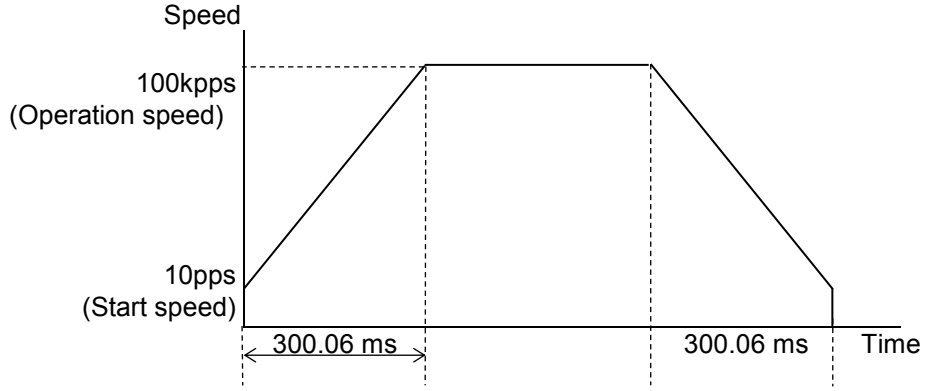
$$0.3\ [\text{s}] = \frac{(10000 - 1) \times (\text{PRUR} + 1) \times 2}{19.6608 \times 10^6}$$

$$\therefore \text{PRUR} = 293.94$$

However, since only integers can be entered for the PRUR register, use “293” or “294”. The actual acceleration/deceleration time will be 299.04 ms (if PRUR = “293”), or 300.06 ms (if PRUR = “294”).

5) Since the acceleration and deceleration times are equal, set PRDR=“0” 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.

However, during positioning operation mode of ramping-down automatic setting (PRMD.MSDP="0"), there are the following restrictions about operation speed override. If you set and execute without following the restriction, function to set ramping-down point automatically cannot follow with operation and stop speed may be higher than FL speed or moving at FL speed occurs after completing deceleration.

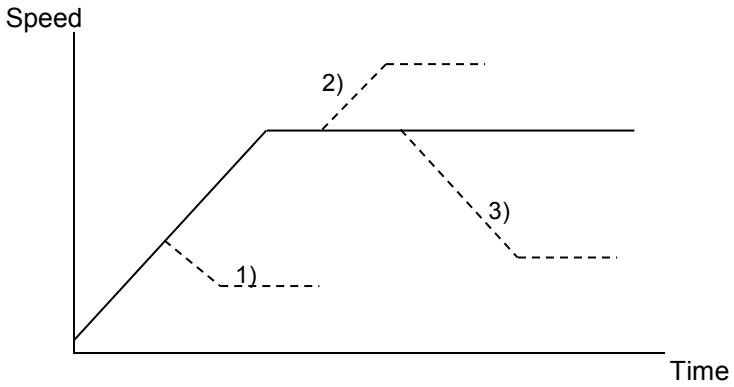
[During linear acceleration / deceleration]

1. Make acceleration and deceleration curves symmetric with RUR=RDR.
2. Only RFH can be changed during operation.

[During S-curve acceleration / deceleration]

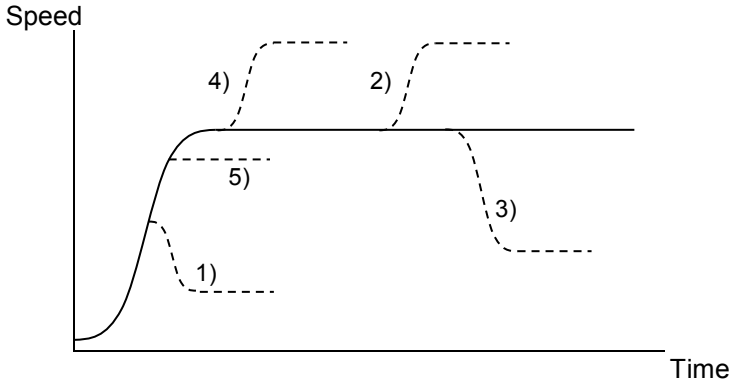
1. Set RUR=RDR="0" so as that acceleration / deceleration characteristic become symmetric.
2. In order to change register in operation, please send data to only RFH register.
3. During acceleration or deceleration, do not change registers.
4. Set to OFF (RMD.MADJ="1" for FH correction control.
5. If ramping-down point is reached to start ramping-down during acceleration by speed change, movement stops before the speed reach to FL speed. Therefore, please be careful to timing of speed change.

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



1) Make RFH smaller while accelerating, the motor accelerates/decelerates until the speed reaches the correct speed.
2), 3) Change RFH after the acceleration/deceleration is complete. The motor will continue accelerating or decelerating until the speed reaches the new speed.

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



1) If make RFH smaller and if ((changed speed) < (speed before change)), the motor will decelerate using an S-curve until the speed reaches the correct speed.
5) If make RFH smaller and if ((changed speed) ≥ (speed before change)), the motor will accelerate without changing the S-curve's characteristic until the speed reaches the correct speed.
4) If make RFH larger while accelerating, the motor will accelerate to the original speed entered without changing the S-curve's characteristic. Then it will accelerate again until the speed reaches the newly set speed.
2), 3) If RFH is changed after the acceleration/deceleration is complete, the motor will accelerate/decelerate using an S-curve until the speed 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, input at least 8 cycles of a reference clock signal while $\overline{RST}$ =L level.
Status after a reset is configured as follows.

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

11-2. Target position override

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

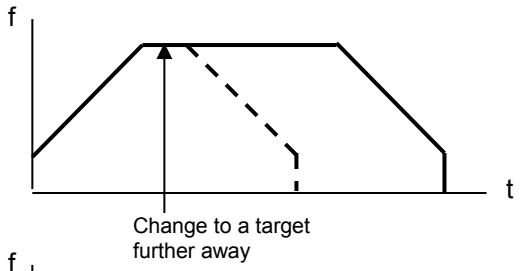
11-2-1. Target position override 1

If acceleration / deceleration characteristics are symmetric, target position can be overridden by change a value of target position (RMV register value).
However, during S-curve acceleration / deceleration with ramping-down point automatic setting (PRMD.MSDP="0"), there are the following restriction. If you set and execute without following the restriction, function to set ramping-down point automatically cannot follow with operation and stop speed may be higher than FL speed or moving at FL speed occurs after completing deceleration.

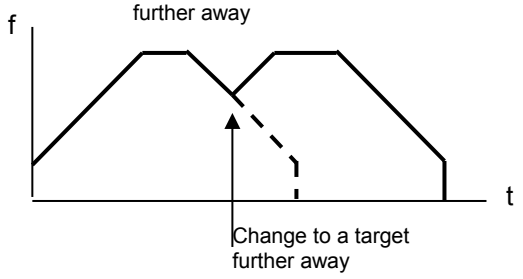
- Do not change target position during acceleration / deceleration.
- Set FH correction control OFF(RMD.MADJ ="1").
- You can override target position only at the timing the speed does not reach to ramping-down point during acceleration / deceleration caused by target position override.

Change a target position by using a start position as a base.

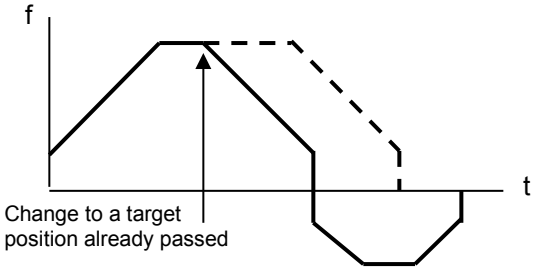
1) If the new target position is further away from the original target position during acceleration or constant speed operation, the motor will maintain the operation using the same speed pattern and it will complete the positioning operation at the position specified in the new target position (new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



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

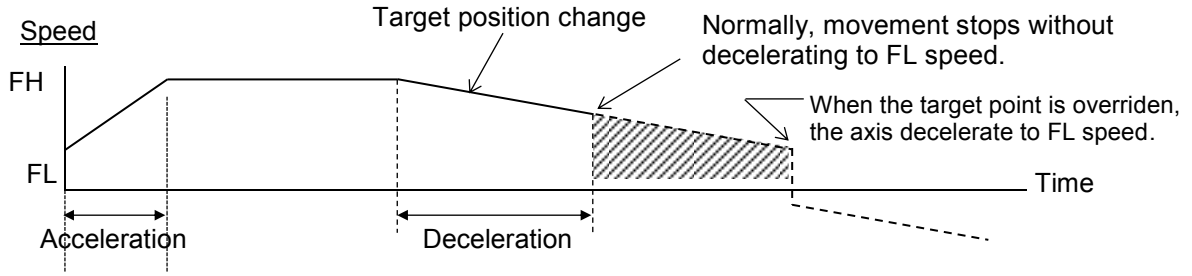


3) If the machine position has already passed over the new target position, or the target position is changed to a position that is closer than the original position during deceleration, the motor will decelerate and stop. Then, the movement will reverse and complete the positioning operation at the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



Acceleration / deceleration is performed only high-speed start. Target position can be changed any number of times until positioning is complete.

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



Note 2: If the LSI starts decelerating by changing the target to a closer position, and if you perform a “position override” to a position further away during this deceleration, the LSI will not re-accelerate. It will feed to the new target after decelerating to complete.

Note 3: The position override is only valid while feeding.
When the LSI receives an override command just a little before stopping a feed, it may not respond to the override command. For this reason, check MSTSW.SEOR after the motor is stopped.
If the override is ignored, MSTSW.SEOR = "1".

When RMV register (90h) is written while stopping, MSTSW.SEOR = “1” for identification.
Therefore, if data is written to the RMV register while stopped before starting to feed, MSTSW.SEOR =“1”. The register is returned to MSTSW.SEOR=“0” within 3 cycles of CLK signals after reading main status with RENV2.MRST=“0” and after writing SEORR (2Eh) commands with RENV.MRST=“1”.

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

By setting PRMD.MPCS="1”, the LSI 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 command pulses) or on the "ON" timing of the PCS input signal.
A PCS signal can change the input logic. The PCS terminal status can be monitored using the RSTS register.

Setting pulse control using the PCS input <PRMD.MPCS (bit 14)> 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 <RENV1.PCSL (bit 24)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 31 24 - - - - - - - n
Reading the PCS signal < RSTS.SPCS (bit 8)> 0: Turn OFF PCS signal 1: Turn ON PCS signal	[RSTS] (READ) 15 8 - - - - - - - n
PCS substitution input < Control command: STAON> Perform processes that are identical to those performed by supplying a PCS signal.	[Control 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-degree phase difference mode.

Common pulse mode: Outputs operation pulses from the OUT terminal and outputs the direction identification signal from the DIR terminal. (RENV.PMD = “000”b ~ “011”b)

2-pulse mode: Outputs positive direction operation pulses from the OUT terminal, and outputs negative direction operation pulses from the DIR terminal. (RENV.PMD = “100”b to “111”b)

90-degree phase difference mode: This mode outputs signals from the OUT terminal and DIR terminal with a 90-degree phase difference. (RENV.PMD = “101”b, “110”b.)

To set output mode for command pulses, RENV1.PMD is used.
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 RENV1.DTMP =“0”, the operation can be delayed for one direction change timer unit (0.2 ms), after changing the direction identification signal.
When RENV1.DTMF = “1”, the LSI will start to output pulses after 10 CLK cycles (0.5 us) after DIR changes.

Setting the pulse output mode <RENV1.PMD2~0 (bits 2~0)>

PMD2~0	When feeding in the positive direction		When feeding in the negative direction	
	OUT output	DIR output	OUT output	DIR output
“000”b		High		Low
“001”b		High		Low
“010”b		Low		High
“011”b		Low		High
“100”b		High	High	
“101”b	OUT / DIR		OUT / DIR	
“110”b	OUT / DIR		OUT / DIR	
“111”b		Low	Low	

[RENV1] (WRITE)

7							0
-	-	-	-	-	n	n	n

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

[RENV1] (WRITE)

31							24
-	-	-	n	-	-	-	-

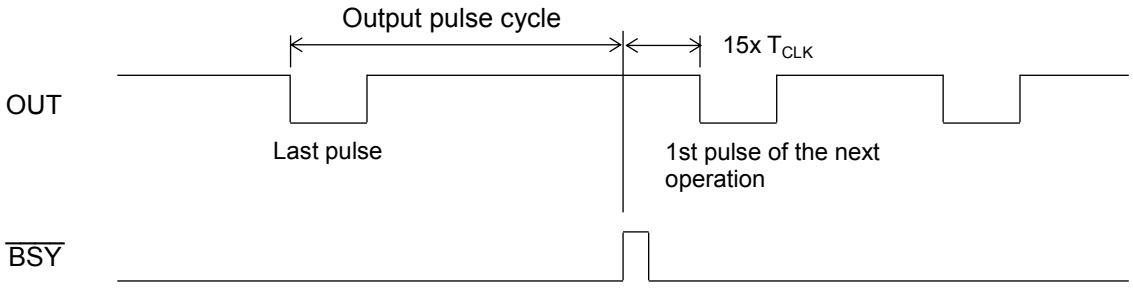
11-3-2. Output pulse length and operation complete timing

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.

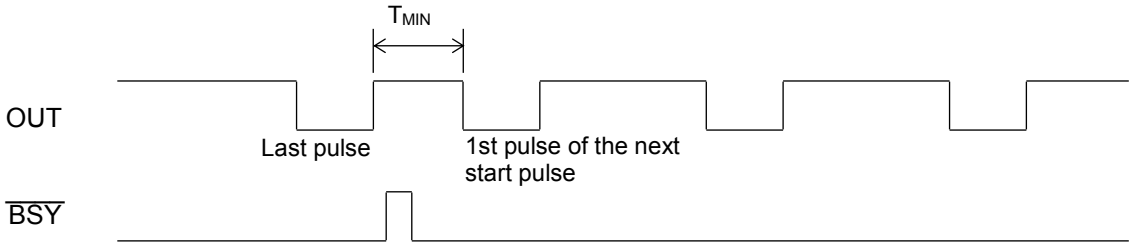
$$\frac{\text{Pulse ON time}}{\text{Pulse cycle}} = \frac{(\frac{(\text{PRMG set value})}{2})}{(\text{PRMG set value} + 1)}$$

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

1) When PRMD.METM = “0” (the point at which the output frequency cycle is complete)



2) When PRMD.METM = “1” (when the output pulse is OFF)



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

Setting the operation complete timing <PRMD.METM (bit 12)> 0: At the end of a cycle of an output frequency 1: When the output pulse is OFF.	[PRMD] (WRITE) 15 … 8 - - - n - - - -

11-4. Mechanical external input control

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

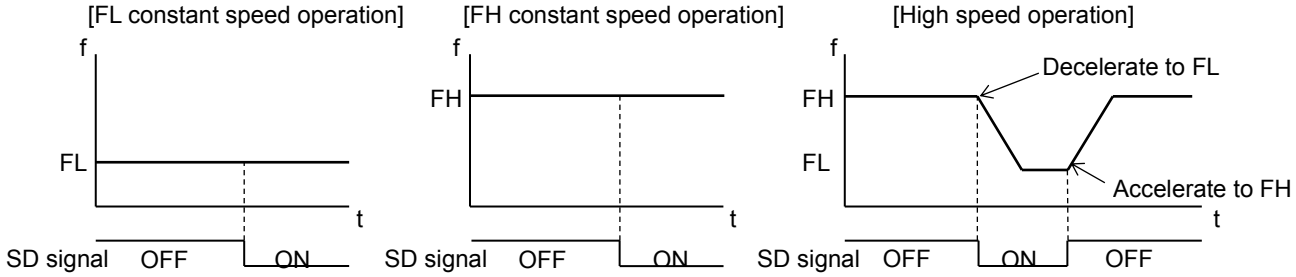
When an end limit signal (a +EL signal when feeding in the + direction) in the feed direction turns ON while operating, motion of a machine will stop immediately or decelerate and stop. After it stops, even if the EL signal is turned OFF, a machine will remain stopped. For safety, please design a structure of the machine so that the EL signal keeps ON until a machine reaches the end of the stroke.
If the EL signal is ON when writing a start command, the movement cannot start in the direction of the particular EL signal that is ON.
By setting RENV1.ELM, the stopping pattern for use when the EL signal is turned ON can be set to immediate stop or deceleration stop (in operation of high speed start only). When the deceleration stop is selected, please note to have room mechanically because the motor stops after passing through the EL position.
When the input noise filter is OFF, the minimum pulse time for the EL signal is one reference clock cycle (0.05 us). When the input noise filter is ON, the LSI will not respond to pulse signals shorter than a specified time.
By reading SSTSW.SPEL and SSTSW.SMEL, you can monitor EL signal.
By reading REST.ESPL and REST.ESML, you can check for an error interrupt factor 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.SPEL and SSTSW.SMEL.
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 and –EL signals <ELL input terminal> L: Positive logic input H: Negative logic input	
Stop method when the +EL and –EL signals turn ON <RENV.ELM (bit 3)> 0: Stop immediately by turning ON the +EL or –EL signal 1: Decelerate and stop by turning ON the +EL or –EL signal (Stop immediately in operation of FL constant start and FH constant start.)	[RENV1] (WRITE) 7 0 - - - - n - - -
Setting the +EL and –EL input noise filter <RENV1.FLTR (bit 26)> 1: Apply a noise filters to +EL, -EL, SD, ORG, ALM, and INP inputs. When a noise filter is inserted, pulses shorter than the value set in RENV1.FTM1~0 are ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -
Select the input noise filter characteristics <RENV1.FTM1~0 (bits 21~20)> “00”b : 3.2 us “10”b : 200us “01”b : 25 us “11”b : 1.6 ms	[RENV1] (WRITE) 23 16 - - n n - - - -
Reading the +EL or –EL signal <SSTSW.SPEL (bit 12), SSTSW.SMEL (bit 13)> 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 factor when the ±EL signal turns on <REST.ESPL (bit 0), REST.ESML (bit 1)> ESPL = “1”: Stop by turning ON the +EL signal ESML = “1”: Stop by turning ON the -EL signal	[REST] (READ) 7 0 - - - - - - n n

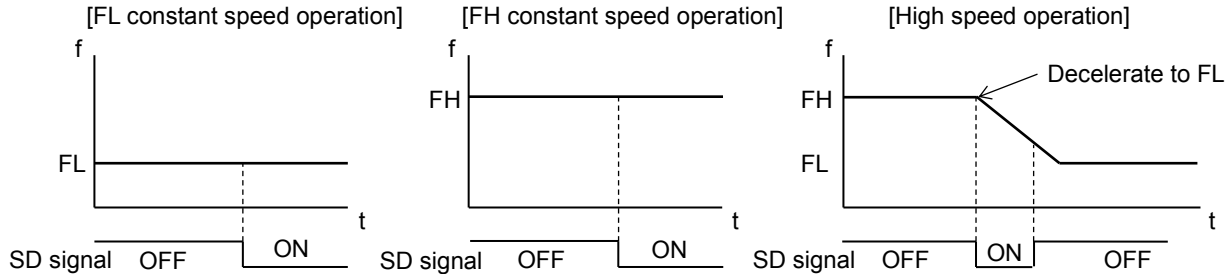
11-4-2. SD signal

When PRMD.MSDE="0", the SD signal will be ignored.
If the SD signal is enabled and the SD signal is turned ON while in operation, the motor will: 1) decelerate, 2) latch and decelerate, 3) decelerate and stop, or 4) latch and perform a deceleration stop, according to the setting of RENV1.SDM and RENV.SDLT.

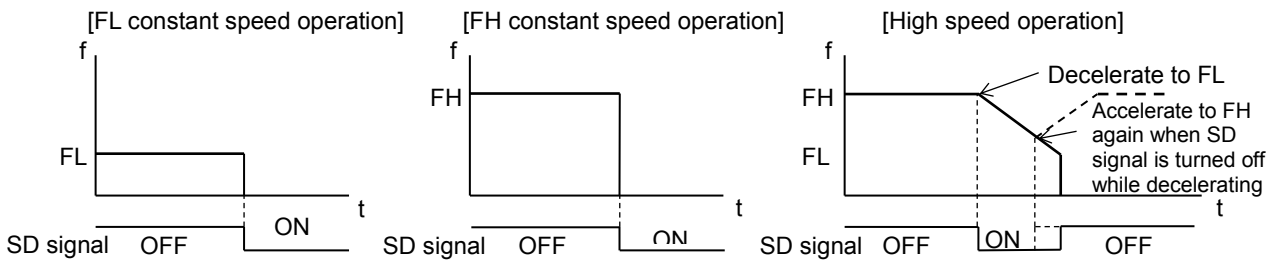
1) Deceleration <RENV1.SDM="0", RENV1.SDLT = "0">
- While feeding at constant speed, the SD signal is ignored.
- While in high speed operation the motor decelerates to the FL speed when the SD signal is turned ON. After decelerating, or while decelerating, if the SD signal turns OFF, the motor will accelerate to the FH speed.
- If the SD signal is turned ON when STAD (52h) command or STAUD (53h) command is written, the motor will operate at FL speed. When the SD signal is turned OFF, the motor will accelerate to FH speed.



2) Latch and deceleration <RENV1.SDM = "0", RENV1.SDLT = "1" >
- While feeding at constant 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 motor will continue moving at FL speed and will not accelerate.
- If the SD signal is turned ON while writing STAD (52h) command or STAUD (53h) command, the motor will feed at FL speed. Even if the SD signal is turned OFF, the motor will not accelerate to FH speed.

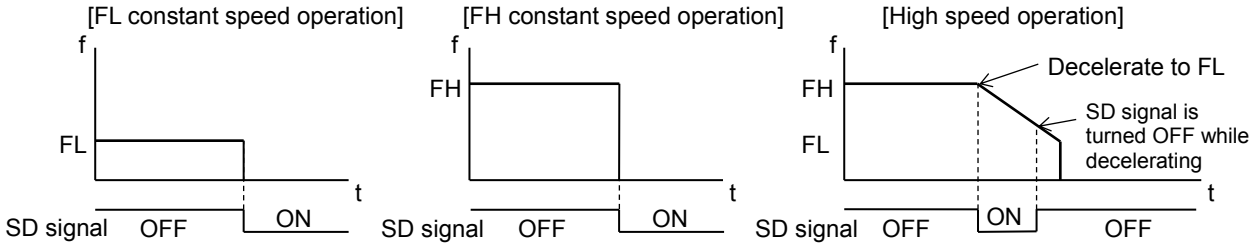


3) Deceleration stop <RENV1.SDM = "1", RENV1.SDLT =" 0" >
- If the SD signal is turned ON while in constant speed operation, the motor will stop immediately.
- While in high speed operation, the motor will decelerate to FL speed when the SD signal is turned ON, and then stop. If the SD signal is turned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



4) Latch, deceleration stop <<RENV1.SDM = “1”, RENV1.SDLT =“ 1” >

- If the SD signal is turned ON while in constant speed operation, the motor will stop immediately.
- If the SD signal is turned ON while in high speed operation, the motor will decelerate to FL speed and then stop. Even if the SD signal is turned OFF during deceleration, the motor will not accelerate.
- If the SD signal is turned ON while writing a start command, the motor will not start moving and the operation will be completed.
- While stopped, the LSI outputs an $\overline{INT}$ signal.

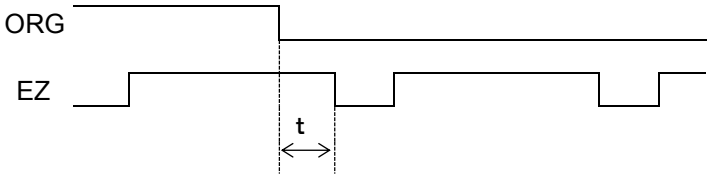


The input logic of the SD signal can be changed. If the latched input is set to accept input from the SD signal (RENV1.SDLT=“1”), and if the SD signal is OFF at the next start, the latch will be reset. The latch is also reset when you select level input as input type (RENV1.SDLT=”0”).
When the input noise filter is OFF the minimum pulse time for the SD signal is two reference clock cycles (0.1 us). When the input noise filter is ON, the LSI will not respond to pulse signals shorter than the specified time. The latch signal of the SD signal can be monitored by reading SSTSW. The SD signal terminal status can be monitored by reading RSTS. By reading the REST register, you can check for an error interrupt factor caused by the SD signal turning ON.

Enable/disable SD signal input <PRMD.MSDE (bit 8)> 0: Disable SD signal input 1: Enable SD signal input	[PRMD] (WRITE) 15 … 8 - - - - - - - n
Input logic of the SD signal <RENV1.SDL(bit 6)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 … 0 - n - - - - - -
Set the operation pattern when the SD signal is turned ON <RENV1.SDM (bit 4)> 0: Decelerates on receiving the SD signal and feeds at FL constant speed 1: Decelerates and stops on receiving the SD signal (Stop immediately in operation of FL constant start and FH constant start.)	[RENV1] (WRITE) 7 … 0 - - - n - - - -
Select the SD signal input type <RENV1.SDLT (bit 5)> 0: The SD signal is to cancel latch input 1: The SD signal is to set latch input (To cancel the latch, turn OFF the SD input at a next start or write latch input cancelation	[RENV1] (WRITE) 7 … 0 - - n - - - - -
Reading the latch status of the SD signal <SSTSW.SSD (bit 15)> 0: The SD latch signal is OFF 1: The SD latch signal is ON	[SSTSW] (READ) 15 … 8 n - - - - - - -
Reading the SD signal <RSTS.SDIN (bit 14)> 0: The SD signal is OFF 1: The SD signal is ON	[RSTS] (READ) 15 … 8 - n - - - - - -
Reading $\overline{INT}$ factor when stopped by the SD signal <REST.ESSD (bit 5)> 1: Deceleration stop caused by the SD signal turning ON	[RSTS] (READ) 7 … 0 - - n - - - - -
Apply an noise input filter to SD input <RENV1.FLTR (bit 26)> 1: Apply a noise filter to the +EL, -EL, SD, ORG, ALM and INP input to ignores signals shorter than a value set in RENV1.FTM are ignored.	[RENV1] (WRITE) 31 … 24 - - - - - n - -
Select the input filter characteristics <RENV1.FTM1~0 (bits 21~20)> “00”b: 3.2 us “10”b: 200 us “01”b: 25 us “11”b: 1.6 m	[RENV1] (WRITE) 23 … 16 - - n n - - - -

11-4-3. ORG, EZ signals

These signals are enabled in the origin return operation modes.
When the input noise filter is OFF, the minimum pulse time for the ORG signal is 1 reference clock cycle (0.05 us). When the input noise filter is ON, the LSI will not respond to pulse signals shorter than a specified time. In addition, the ORG signal is sampled at the timing output pulse is ON, the ORG input must be kept ON for more than one pulse.
The input logic of the ORG signal and EZ signal can be changed using the RENV1 register and RENV2 registers.
The ORG terminal status can be monitored by reading SSTSW. The EZ terminal status can be monitored by reading the RSTS register.
For details about the origin return operation modes, see “9-5, Origin position operation mode.”
ORG signal and EZ signal timing (When the input noise filter is OFF)



T_{CLK}: Reference clock cycle
(i) When $t \geq 2 \times T_{CLK}$, the pulse is counted
(ii) When $T_{CLK} < t < 2 \times T_{CLK}$, counting is undetermined.
(iii) When $t \leq T_{CLK}$, the pulse is not counted.

Enabling the ORG and EZ signals <PRMD.MOD6~0 (bits 6~0)> “001 0000”b: Origin return in the positive direction “010 1000”b: Origin return in the negative direction	[PRMD] (WRITE) 7 0 0 n n n n n n n
Setting the origin return method <RENV2.ORM (bit 29)> 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 <RENV1.ORGL (bit 7)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 n - - - - - - -
Set the ORG input noise filter <RENV1.FLTR(bit 26)> 1: Apply a noise filter to the +EL, -EL, SD, ORG ALM, and INP input. By applying a noise filter, pulses shorter than the value set in RENV1.FTM1~0 are ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -
Setting the time constant for the input noise filter <RENV1.FTM1~0 (bits 21~20)> “00”b: 3.2 us “10”b: 200 us “01”b: 25 us “11”b: 1.6 ms	[RENV1] (WRITE) 23 16 - - n n - - - -
Read the ORG signal <SSTSW.SORG (bit 14)> 0: The ORG signal is OFF 1: The ORG signal is ON	[SSTSW (READ) 15 8 - n - - - - - -
Set the EZ count number <RENV2.EZD3~0 (bits 27~24)> Set the EZ count number for counting. as a condition for origin return operation completion. Specify the value (the number to count to – 1) in EZD3~0. The setting range is 0~15.	[RENV2] (WRITE) 31 24 - - - - n n n n
Specify the input logic of the EZ signal <RENV2.EZL (bit 28)> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 31 24 - - - n - - - -
Read the EZ signal <RSTS.SEZ (bit 10)> 0: The EZ signal is OFF 1: The EZ signal is ON	[RSTS] (READ) 15 8 - - - - - n - -
Set the EZ input noise filter <RENV1.EINF (bit 18)> 1: Apply a noise filter to the EA, EB, EZ input. By applying a noise 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 train input accepting servo driver systems have a deviation counter to count the difference between command pulse inputs and feedback pulse inputs. The driver controls so that the difference becomes zero. In other words, servomotor operates behind command pulses and, even after the command pulses stop, the servomotor systems keep feeding until the count in the deviation counter reaches “0”.

With this LSI, you can select to make this LSI to determine the timing to input a positioning complete signal (INP signal) as when an operation is complete from a servo driver in place of the pulse output complete timing.

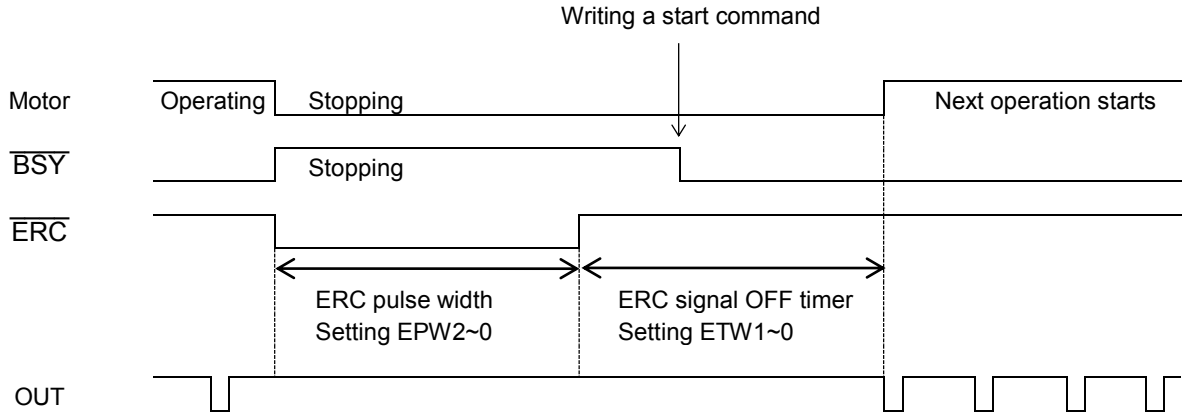
When the INP signal input is used to indicate the completion status of an operation, the $\overline{\text{BSY}}$ signal when an operation is complete, stop condition bits of the main status (MSTSW.SSCM, MSTSW.SRUM, MSTSW.SENI, MSTSW.SEND, MSTSW.SERR, MSTSW.SINT), and operation status of the extension status (RSTS.CND3~0) 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 1 cycle of the reference clock (0.05 usec) when the input noise filter is OFF. If the input noise filter is ON, the LSI does not receive pulses shorter than the set width.
If the INP signal is already ON when the LSI is finished outputting pulses, it treats the operation as complete, without any delay.
The INP signal can be monitored by reading the RSTS register.

Set the operation complete delay using the INP signal <PRMD.MINP (bit 9)> 0: No operation complete delay waiting for the INP signal. 1: Delay operation complete (status, $\overline{\text{BSY}}$) until the INP signal turns ON.	[PRMD] (WRITE) 15 8 - - - - - - n -
Input logic of the INP signal <RENV1.INPL (bit 22)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 23 16 - n - - - - - -
Reading the INP signal <RSTS.SINP (bit 16)> 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 noise filter <RENV1.FLTR (bit 26)> 1: Apply a noise filter to the +EL, -EL, SD, ORG, ALM and INP input. By applying a noise filter, pulses shorter than a value in set RENV1.FTM1~0 are ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -
Select the input noise filter characteristics <RENV1.FTM1~0(bits 21~20)> “00”b: 3.2 us “10”b: 200 us “01”b: 25 us “11”b: 1.6 ms	[RENV1] (WRITE) 23 16 - - n n - - - -

11-5-2. ERC signal

A servomotor delays the stop until the deviation counter in the servo driver reaches “0”, even after command pulses have stopped delivered. In order to stop the servomotor immediately, the deviation counter need to be cleared.
This LSI can output a signal to clear the deviation 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 in RENV1.EPW. If an interval is required for the servo driver to recover after turning H level ERC signal (OFF) before it can receive new command pulses, the ERC signal OFF timer can be selected by setting RENV1.ETW.



In order to output an ERC signal at the completion of an origin return operation, set in RENV1.EROR (bit 11) = 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, Origin return operation (ORM = “0”)."

In order to output an ERC signal by an immediate stop based on the EL signal, ALM signal, or $\overline{\text{CEMG}}$ signal input, or on CMEMG (05h) command, set in RENV1.EROE, 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 ERCOUT (24h) command.
The output logic of the ERC signal can be changed by setting the RENV1 register. By reading the RSTS register to the ERC signal can be monitored.

Set automatic output for the ERC signal <RENV1.EROE (bit 10)> 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 <RENV1.EROR (bit 11)> 0: Does not output an ERC signal at the completion of an origin return operation. 1: Automatically outputs an ERC signal at the completion of an origin return operation.	[RENV1] (WRITE) 15 8 - - - - n - - -
Set the ERC signal output width <RENV1.EPW2 ~0 (bits 14~12)> “000”b : 12 us “100”b : 13 ms “001”b : 102 us “101”b : 52 ms “010”b : 408 us “110”b : 104 ms “011”b : 1.6 ms “111”b : Level output	[RENV1] (WRITE) 15 8 - n n n - - - -
Select output logic for the ERC signal <RENV1.ERCL (bit 15)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 15 8 n - - - - - - -
Specify the ERC signal OFF timer time <RENV1.ETW1~0 (bits 17~16)> “00”b: 0 us “10”b: 1.6 ms “01”b: 12 us “11”b: 104 ms	[RENV1] (WRITE) 23 16 - - - - - - n n

Read the ERC signal <RSTS.SERC (bit 9)> 0: The ERC signal is OFF 1: The ERC signal is ON	[RSTS] (READ) 15 8 0 - - - - - n -
Emergency stop command <CMEMG: Operation command> Output an ERC signal	[Operation command] 05h
ERC signal output command <ERCOUT: Control command > Turn ON the ERC signal	[Control command] 24h
ERC signal output reset command <ERCRST: Control command > Turn OFF the ERC signal	[Control command] 25h

11-5-3. ALM signals

Input alarm (ALM) signal.
When the ALM signal turns ON while in operation, the motor will stop immediately or decelerate and stop.
At the constant speed start, the motor will stop immediately. To stop at high speed start, you can select between to stop immediately or to decelerate and stop.
To stop using deceleration, keep the ALM input ON until the motor stops operation.
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 us) if the input noise filter is OFF.
If the input noise filter is ON, the LSI does not receive pulses shorter than a specified width.
The input logic of the ALM signal can be changed. The signal status of the ALM signal can be monitored by reading sub status.

Stop method when the ALM signal is ON <RENV1.ALMM (bit 8)> 0: Stop immediately when the ALM signal is turned ON 1: Decelerate and stop when the ALM signal is turned ON (Stop immediately in operation of FL constant start and FH constant start.)	[RENV1] (WRITE) 15 8 - - - - - - - n
Input logic setting of the ALM signal <RENV1.ALML (bit 9)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 15 8 - - - - - - n -
Read the ALM signal <SSTSW .SALM (bit 11)> 0: The ALM signal is OFF 1: The ALM signal is ON	[SSTSW] (READ) 15 8 - - - - n - - -
Reading the stop factor when the ALM signal is turned ON <REST.ESAL (bit 7) > 1: Stop due to the ALM signal being turned ON	[REST] (READ) 7 0 n - - - - - - -
Set the ALM input noise filter <RENV1.FLTR (bit 26)> 1: Apply a noise filter to the +EL, -EL, SD, ORG ALM and INP input When a filter is applied, pulses shorter than a value set in RENV1.FTM1~0 are ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -
Select the input noise filter characteristics <RENV1.FTM1~0 (bits 21~20)> "00"b : 3.2 us "10"b : 200 us "01"b : 25 us "11"b : 1.6 ms	[RENV1] (WRITE) 23 16 - - n n - - - -

11-6. External start, simultaneous start

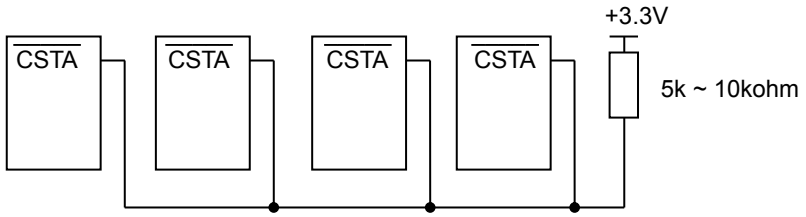
11-6-1. $\overline{\text{CSTA}}$ signal

This LSI can start triggered by an external signal on the $\overline{\text{CSTA}}$ terminals. Set PRMD.MSY (bits 19 and 18) = “01”b and the LSI will start feeding when the $\overline{\text{CSTA}}$ goes “L” level.
In controlling multiple axes using more than one LSI, when you connect the $\overline{\text{CSTA}}$ terminals on each LSI and input the same signal, each axis on the each LSI starts to move. In this example a start signal can be output through the $\overline{\text{CSTA}}$ terminal.
The logic on the $\overline{\text{CSTA}}$ terminals cannot be changed.
By setting the RIRQ register, theLSI outputs $\overline{\text{INT}}$ signal at a simultaneous start (when the $\overline{\text{CSTA}}$ input is ON).
By reading the RIST register, the factor of an event interrupt can be checked.
The operation status (waiting for $\overline{\text{CSTA}}$ input), and status of the $\overline{\text{CSTA}}$ terminal can be monitored by reading the RSTS register.

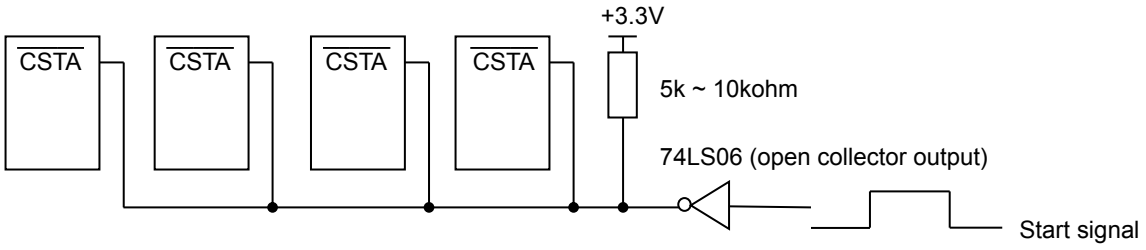
<How to make a simultaneous start>
Set PRMD.MSY1 to 0 (bits 19 and 18) = “01”b for the axes you want to start. Write a start command and make the axis "waiting for $\overline{\text{CSTA}}$ input". Then, start movement on the axes simultaneously by either of the methods described below.
1)Write a simultaneous start command. The LSI will output a one shot signal of 8 reference clock cycles (approx. 0.4 us) from the $\overline{\text{CSTA}}$ terminal.
2) Input a hardware signal from outside.
Supply a hardware signal after driving the terminal with open collector output (74LS06 or equivalent).

$\overline{\text{CSTA}}$ signal can be supplied as level trigger or edge trigger inputs. However, when level trigger input is selected, if $\overline{\text{CSTA}}$ = “L” level and a start command is written, movement of the axis will start immediately.
After connecting the $\overline{\text{CSTA}}$ terminals on each LSI, each axis can start independently using start commands.
To release the "waiting for $\overline{\text{CSTA}}$ input" status, write STOP (49h) command.

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



2) To start simultaneously from an external circuit 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 us).

Waiting for $\overline{\text{CSTA}}$ input <PRMD.MSY1~0 (bits 19~18)> "01"b: Start by inputting a $\overline{\text{CSTA}}$ signal	[PRMD] (WRITE) 23 16 - - - - n n - -
Specify the input specification for the$\overline{\text{CSTA}}$ signal <RENV1.STAM (bit 18)> 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 <RSTS.SSTA (bit 5)> 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 <RSTS.CND 3~0(bits3~0)> "0010"b: Waiting for $\overline{\text{CSTA}}$ input	[RSTS] (READ) 7 0 - - - - n n n n
Set an event interrupt factor <RIRQ.IRSA (bit 12)> 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 factor <RIST.ISSA (bit 13)> 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 of 8 reference clock cycles wide from the $\overline{\text{CSTA}}$ terminal. (The $\overline{\text{CSTA}}$ terminal is bi-directional. It input output)	[Operation command] 06h
Simultaneous start command for only own axis <SPSTA: Operation command> Used the same way as when a $\overline{\text{CSTA}}$ signal is supplied, for an own axis only.	[Operation command]] 2Ah

11-6-2. PCS signal

The PCS input which is used for the target position override 2 function can be enabled the $\overline{\text{CSTA}}$ signal for an own axis only by setting RENV1.PCSM (bit 30) = "1" and PRMD.MSY = "01"b.

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

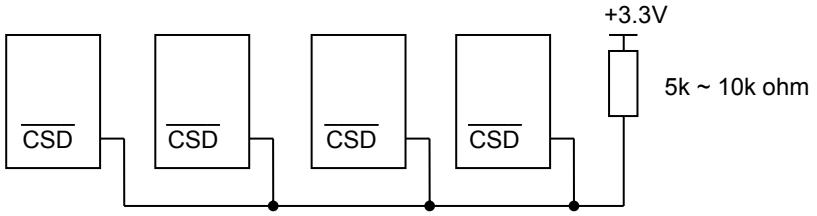
Specify the function of the PCS signal <RENV1.PCSM (bit 30)> 1: Make PCS input effective as $\overline{\text{CSTA}}$ for an own axis only.	[RENV1] (WRITE) 31 24 - n - - - - - -
Set the Waiting for $\overline{\text{CSTA}}$ input <PRMD.MSY1~0 (bits 19~18)> "01"b: Start on a $\overline{\text{CSTA}}$ input.	[PRMD] (WRITE) 23 16 - - - - n n - -
Set the input logic of the PCS signal <RENV1.PCSL (bit 24)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 31 24 - - - - - - - n
Read the PCS signal <RSTS.SPCS (bit 8)> 0: The PCS signal is OFF 1: The PCS signal is ON	[RSTS] (READ) 15 8 - - - - - - - n

11-7. External deceleration / simultaneous deceleration

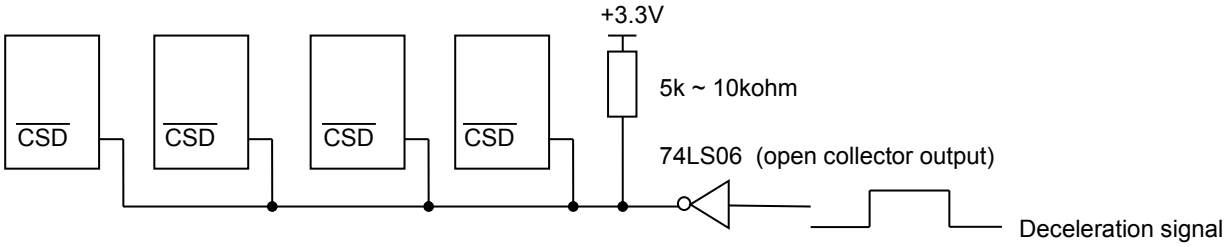
This LSI can execute deceleration and deceleration stop triggered by an external signal using the $\overline{CSD}$ terminal. When PRMD.MSDE="1" to enable a deceleration stop by $\overline{CSD}$ input. The motor decelerates or decelerate and stop on the timing $\overline{CSD}$ = "L". The method how to set deceleration and the setting of deceleration operation are the same as SD signals.
The input logic of the $\overline{CSD}$ terminal cannot be changed.
When multiple LSIs are used to control multiple axes, connect all of the $\overline{CSD}$ terminals from each LSI and input the same signal so that the axes which are set to enable deceleration by $\overline{CSD}$ input. In this case, the LSI outputs a deceleration signal from the $\overline{CSD}$ terminal.
You can monitor $\overline{CSD}$ terminal status by reading the RSTS register.

<How to make a simultaneous deceleration>
Set PRMD.MCDE = "1" for each of the axes that you want to decelerate simultaneously. Then decelerate simultaneously by the following method.
1. Input hardware signals to SD terminals externally.
   $\overline{CSD}$ terminal will output a one shot signal of 8 reference clock cycles in width (approx. 0.4 us).
2. Input hardware signals to $\overline{CSD}$ terminal externally.
   Supply a hardware signal after driving the terminal with an open collector output (74LS06 or equivalent).

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 of 4 reference clock cycles or more in width (approx. 0.2 us).

Setting to enable $\overline{CSD}$ input <PRMD.MCDE (bit 28)> 1. Enable a stop by $\overline{CSD}$ input. (Deceleration or decelerate and stop)	[PRMD] (WRITE) 31 24 0 0 - n 0 - - -
Auto output setting for the $\overline{CSD}$ signal <PRMD.MCDO (bit 29)> 1: Outputs $\overline{CSD}$ signal automatically when $\overline{SD}$ signal is input. (Output signal width: 8 reference clock cycles)	[PRMD] (WRITE) 31 24 0 0 n - 0 - - -
Read the $\overline{CSD}$ signal < RSTS.SCD (bit 4)> 0: The $\overline{CSD}$ signal is OFF 1: The $\overline{CSD}$ signal is ON	[RSTS] (READ) 7 0 - - - n - - - -
Read t error interrupt factor < REST.ESSP (bit 3)> 1. When stopped because the $\overline{CSD}$ signal turned ON.	[RSTS] (READ) 7 0 - - - - n - - -

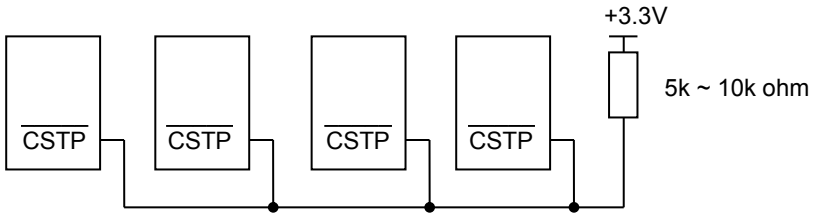
11-8. External stop / simultaneous stop

This LSI can execute an immediate stop or a deceleration stop triggered by an external signal using the $\overline{\text{CSTP}}$ terminal. Set PRMD.MSPE = “1” to enable a stop by a $\overline{\text{CSTP}}$ input. The motor will stop immediately or decelerate and stop when the $\overline{\text{CSTP}}$ terminal is “L” level\. Stop method in the case that the motor starts in constant speed is only immediate start. In the case that the motor starts in high speed, you can select from immediate stop and deceleration 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, the LSI outputs a stop signal from the $\overline{\text{CSTP}}$ terminal.
When a motor stops because the $\overline{\text{CSTP}}$ signal is turned ON, an $\overline{\text{INT}}$ signal is output. By reading the REST register, you can determine an error interrupt factor . You can monitor $\overline{\text{CSTP}}$ terminal status by reading the RSTS register.

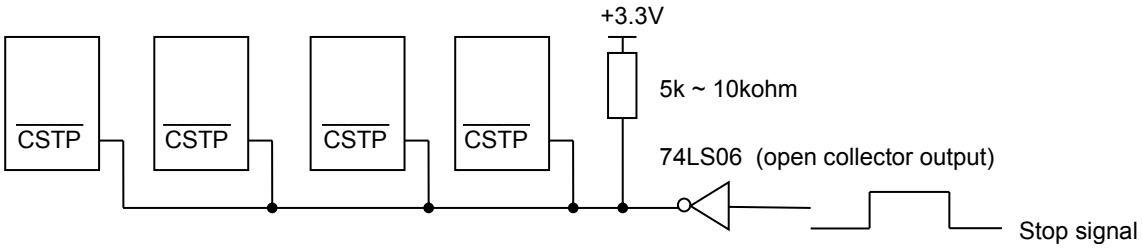
<In the case of simultaneous stop>
Set PRMD.MSPE = “1” for each of the axes that you want to stop simultaneously. Then start these axes.
Stops simultaneously these axes using any of the following three cases.
1) Writing CMSTP (07h) command
   $\overline{\text{CSTP}}$ terminal will output a one shot signal of 8 reference clock cycles in width (approx. 0.4 us).
2) Supply an external hardware signal
   Supply a hardware signal after driving the terminal with an open collector output (74LS06 or equivalent).
3) When axis set to enable $\overline{\text{CSTP}}$ signal output (PRMD.MSPO=”1”) stopped by error
   The $\overline{\text{CSTP}}$ terminal will output a one shot signal of 8 reference clock cycles (approximately 0.4 us) when a stop caused by an error occurs on an axis that has PRMD.MSPO = “1”.

Even when the $\overline{\text{CSTP}}$ terminals on LSIs are connected together, each axis can still be stopped independently by using a 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 of 4 reference clock cycles or more in width (approx. 0.2 us).

Setting to enable $\overline{\text{CSTP}}$ input <PRMD.MSPE (bit 24)> 1. Enable a stop by 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 <PRMD.MSPO (bit 25)> 1: When an axis stops because of an error, the LSI 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 be output a signal when a motor is stopped by a command <RENV2.CSP0 (bit 13)> 0: The LSI will not output a $\overline{\text{CSTP}}$ signal when an axis is stopped by a command. 1: When RMD.MSP0 = "1", the LSI will output the $\overline{\text{CSTP}}$ signal even if a motor 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. <RENV1.STPM (bit 19)> 0: Stop immediately when the $\overline{\text{CSTP}}$ signal is turned ON. 1: Decelerate and stop when the $\overline{\text{CSTP}}$ signal is turned ON. (Stop immediately in operation of FL constant start and FH constant start.)	[RENV1] (WRITE) 23 16 - - - - n - - -
Read the $\overline{\text{CSTP}}$ signal < RSTS.SSTP (bit 6)> 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 interrupt < REST.ESSP (bit 3)> 1. When stopped because the $\overline{\text{CSTP}}$ signal turned ON.	[RSTS] (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.)	[Operation command] 07h

11-9. Emergency stop

This LSI has a $\overline{\text{CEMG}}$ signal input terminal for use as an emergency stop signal.
While in operation, if the $\overline{\text{CEMG}}$ = L level or if you write CMEM (05h) command, all the axes will stop immediately. While the $\overline{\text{CEMG}}$ = L level, no axis can be operated.
The logic of the $\overline{\text{CEMG}}$ signal input terminal cannot be changed.

When the axes are stopped because the $\overline{\text{CEMG}}$ = L level, the LSI will output an $\overline{\text{INT}}$ signal. By reading the REST register, error interruption factor can be determined.
The status of the $\overline{\text{CEMG}}$ signal input terminal can be monitored by reading the RSTS register.

Read the $\overline{\text{CEMG}}$ signal <RSTS.SEMG (bit 7)> 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 <REST.ESEM (bit 4)> 1. Stopped when the $\overline{\text{CEMG}}$ signal was turned ON.	[RSTS] (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 LSI counter may count this pulse. (Deviation from the command position control). Therefore, after an emergency stop, you must perform an origin return to match the command position with the mechanical position.

11-10. Counters

11-10-1. 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 at the start, regardless of the operation mode selected. It decreases the value with each pulse that is output. However, if RMD.MPCS = “1” and during position override 2, the counter value will not decrease until the PCS input turned ON.
Input to COUNTER 1 and COUNTER 2 can be selected as follows by setting the RENV3 register).
* "0": Possible to count

	COUNTER 1	COUNTER 2
Counter type	Up/down counter	Up/down counter
Number of bits	28	28
Output pulse	O	O
Encoder (EA/EB) input	O	O

Set COUNTER 1 input 0: Output pulses 1: EA/EB input	<RENV3.CIS1 (bit 0)>	[RENV3] (WRITE) 7 0 - - - - - - - n
Set COUNTER 2 input 0: EA/EB input 1: Output pulses	<RENV3.CIS2 (bit 1)>	[RENV3] (WRITE) 7 0 - - - - - - n -

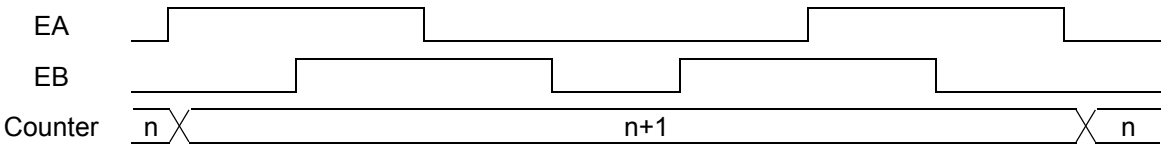
The EA/EB input terminals, that are used as inputs for the counter, can be selected from the following two:
1) Signal input method: Input 90-degree phase difference signals (1x, 2x, 4x)
Counter direction: Count up when the EA input phase is leading.
Count down when the EB input phase is leading.
2) Signal input method: Supply count-up or countdown pulses (Two-pulse input)
Counter direction: Count up on the rising edge of the EA input.
Count down on the rising 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 register.

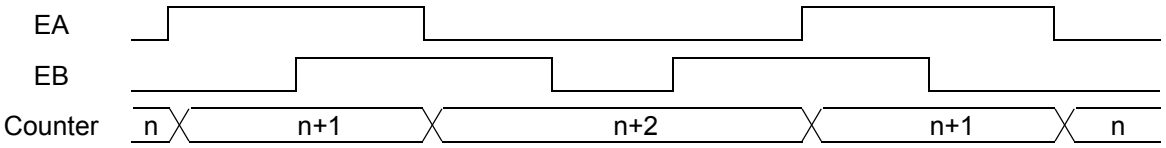
Set the input noise filter for EA/EB/EZ <RENV2.EINF (bit 18)> 0: Turn OFF the filter function 1: Turn ON the filter function (Input signals shorter than 3 reference clock cycles are ignored.)	[RENV2] (WRITE) 23 16 - - - - - n - -
Setting the EA/EB input <RENV2.EIM1~0 (bits 17~16)> "00"b : 90-degree phase difference, 1x "10"b : 90-degree phase difference, 4x "01"b : 90-degree phase difference, 2x "11"b : 2 pulse of up or down input pulses	[RENV2] (WRITE) 23 16 - - - - - - n n
Specify the EA/EB input count direction <RENV2.EDIR (bit 19)> 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 - - -
Mask count input for EA/EB input <RENV2.EOFF (bit 14)> 0: Enable EA/EB input 1: Disable EA/EB input. (EZ input is valid.)	[RENV2] (WRITE) 15 8 - n - - - - - -
Reading EA/EB input error < REST.ESEE (bit 7)> 1: An EA/EB input error occurred.	[REST] (READ) 7 0 n - - - - - - -

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

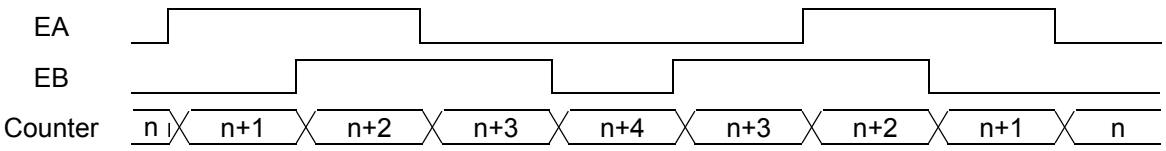
1) When using 90-degree phase difference signals and 1x input



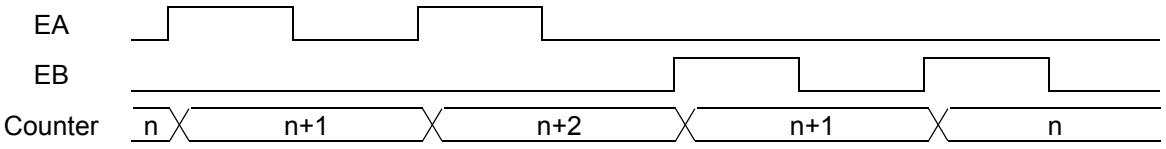
2) When using 90-degree phase difference signals and 2x input



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



4) When using Two-pulse input (counted on the rising edge)



11-10-2. Counter latch and reset

The following three methods allow all the counters to latch their count value using the RENV3 register. The latched values can read from the RLTC1 and RLTC2 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 signal can be set in the RENV1 register. An $\overline{INT}$ signal can be output as an event interrupt factor when the LSI 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 of the counter reset signal However, the LSI 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 signal 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 register. The $\overline{INT}$ signal can be output as an event interrupt factor when the counter value is latched by the LTC and ORG input signals.

Specify the LTC signal <RENV1.LTCL (bit 23)> 0: Latch on the falling edge. 1: Latch on the rising edge.	[RENV1] (WRITE) 23 16 n - - - - - - -
Read the LTC signal <RSTS.SLTC (bit 13)> 0: The LTC signal is OFF 1: The LTC signal is ON	[RSTS] (WRITE) 15 8 - - n - - - - -
Set the COUNTER 1 latch & clear function <RENV3.CU1L (bit 4)> 0: COUNTER 1 is not cleared after it is latched. 1: COUNTER 1 is cleared soon after it is latched.	[RENV3] (WRITE) 7 0 - - - n - - - -
Set the COUNTER 2 latch & clear function <RENV3.CU2L (bit 8)> 0: COUNTER 2 is not cleared after it is latched. 1: COUNTER 2 is cleared soon after it is latched.	[RENV3] (WRITE) 15 8 - - - - - - - n
Set COUNTER 1 to latch on an external input <RENV3.LOF1 (bit 5)> 0: Latch COUNTER 1 on an LTC signal input. 1: Do not latch COUNTER 1 on an LTC signal input.	[RENV3] (WRITE) 7 0 - - n - - - - -
Set COUNTER 2 to latch on an external input <RENV3.LOF2 (bit 9)> 0: Latch COUNTER 2 on an LTC signal input. 1: Do not latch COUNTER 2 on an LTC signal input.	[RENV3] (WRITE) 15 8 - - - - - - n -
Set COUNTER 1 to latch on an origin return <RENV3.CU1R (bit 6)> 0: COUNTER 1 is not latched when returning at the origin position. 1: COUNTER 1 is latched when returning at the origin position.	[RENV3] (WRITE) 7 0 - n - - - - - -
Set COUNTER 2 to latch on an origin return <RENV3.CU2R (bit 10)> 0: COUNTER 2 is not latched when returning at the origin position. 1: COUNTER 2 is latched when returning at the origin position.	[RENV3] (WRITE) 15 8 - - - - - n - -
Set an event interrupt cause <RIRQ.IRLT (bit 8) and RIRQ.IROL (bit 9)> 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 < RIST.ISLT (bit 8), RIST.ISOL (bit 9)> 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: Control command> Latch the contents of the counters (COUNTER 1~2).	[Control command] 29h

Counter reset command <CUN1R~CUN2R: Control command> 20h: Reset COUNTER 1. 21h: Reset COUNTER 2.	[Control command] 20h 21h

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

11-10-3. Stop the counter

There are two methods for stopping counters: stop the count operation (RENV3.CU1H, RENV3.CH2H) or set a mask on the counter input (PRMD.MCCE, RENV2.EOFF).
The counter operation can be stopped for independently COUNTER 1 and COUNTER 2. Selection of the counter input is not related to stopping.
When the count input is set to be masked, the counter that select the input will be stopped.

In timer mode, a counter set "output pulses" as counts input will stop counting, regardless of the counter stop setting selected.
The counter that its count input is set to EA/EB input will performs counting by EA/EB input .

If inputs is regarded as output pulse and a counter set to "counting output pulses"(RENV1.RMSK) is enabled, the LSI will not output pulses. However, the counter will continue counting unless it is set to stop.

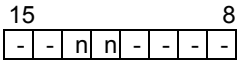
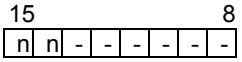
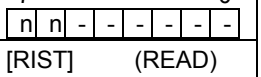
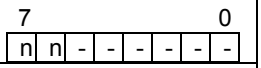
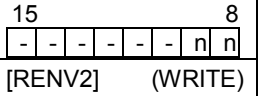
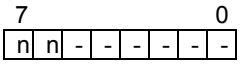
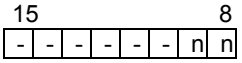
Stopping count operation of COUNTER 1 <RENV3.CU1H (bit 2)> 1. Stop COUNTER 1 counting operation.	[RENV3] (WRITE) 7 0 - - - - - n - -
Stopping count operation of COUNTER 2 <RENV3.CU2H (bit 3)> 1. Stop COUNTER 2 counting operation.	[RENV3] (WRITE) 7 0 - - - - n - - -
Set the count input mask for output pulses <PRMD.MCCE (bit 11)> 1: The counter that is set to count "output pulses" will stop.	[PRMD] (WRITE) 15 8 - - - - n - - -
Set the count input mask for EA/EB pulses <RENV2.EOFF (bit 14)> 1: The counter that is set to count "EA/EB input" will stop.	[RENV2] (WRITE) 15 8 - n - - - - - -
Set to mask output pulse. <RENV1.PMSK (bit 13)> 1: Mask the output pulse.	[RENV1] (WRITE) 31 24 n - - - - - - -

11-11. Comparator

11-11-1. Comparator types and functions

This LSI has four 32-bit comparators circuits per axis. These are referred to as "Comparator 1, 2, 3 and 4".
Comparator 1 compares the setting in the RCMP1 register with COUNTER 1.
Comparator 2 compares the setting in the RCMP2 register with COUNTER 2.
Comparator 3 and 4 is for software limit only.
One of three comparison methods (=, <, and >) between comparator 1 and 2 can be selected, and the comparison results can be output to a terminal. Also, the LSI can output an $\overline{\text{INT}}$ 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 synchronous start function. For descriptions of these functions, see "11-11-2. Ring count function" and "11-12-2. Start on an internal synchronous signal."
For details of Comparator 3 and 4, see “11-11-3. Software limit function”.

Use the RENV2 and RENV3 registers to set the above-described comparators 1 and 2.

Set the comparison conditions for Comparator 1 <RENV3.C1S1~0 (bits 13~12)> “00”b: Turn OFF the comparator function “01”b: (RCMP1) = (COUNTER 1) “10”b: (RCMP1) > (COUNTER 1) “11”b: (RCMP1) < (COUNTER 1)	[RENV3] (WRITE) 
Set the comparison conditions for Comparator 2 <RENV3.C2S1~0 (bits 15~14)> “00”b: Turn OFF the comparator function “01”b: (RCMP2) = (COUNTER 2) “10”b: (RCMP2) > (COUNTER 2) “11”b: (RCMP2) < (COUNTER 2)	[RENV3] (WRITE) 
Set an event interrupt cause <RIRQ.IRC2 (bit 7), RIRQ.IRC1 (bit 6)> IRC1 (bit 6) = “1”: Outputs an $\overline{\text{INT}}$ signal when Comparator 1 conditions are met. IRC2 (bit 7) = “1”: Outputs an $\overline{\text{INT}}$ signal when Comparator 2 conditions are met.	[RIRQ] (WRITE) 
Read the event interrupt cause <RIST.ISC2 (bit 7), RIST.ISC1 (bit 6) > IRC1 (bit 6) = “1”: When the Comparator 1 conditions are met. IRC2 (bit 7) = “1”: When the Comparator 2 conditions are met.	[RIST] (READ) 
Read the comparator condition status<MSTSW.SCP2 (bit 9), MSTSW.SCP1 (bit 8)> SCP1 (bit 8) = “1”: When the Comparator 1 conditions are met. SCP2 (bit 9) = “1”: When the Comparator 2 conditions are met.	[MSTSW] (READ) 
Set the specifications for the P3/CP1 terminal <RENV2.P3M1~0 (bits 7~6) > “00”b: General-purpose input “01”b: General-purpose output “10”b: Output a CP1 signal (when the Comparator 1 conditions are met) using negative logic. “11”b: Output a CP1 signal (when the Comparator 1 conditions are met) using positive logic.	[RENV2] (WRITE) 
Set the specifications for the P4/CP2 terminal <RENV2.P4M1~0 (bits 9~8) > “00”b: General-purpose input “01”b: General-purpose output “10”b: Output a CP2 signal (when the Comparator 2 conditions are met) using negative logic. “11”b: Output a CP2 signal (when the Comparator 2 conditions are met) using positive logic.	[RENV2] (WRITE) 

11-11-2. Ring count function

COUNTER 1 and COUNTER 2 have a ring count function for use in controlling a rotating table.

Set RENV3.C1RM = “1” and COUNTER 1 will be in the ring count mode. Then the LSI can perform the following operations.
- Count value will be “0” when the counter counts up from the value in RCMP1.
- Count value will be the count equals to the value in RCMP1 when the counter counts down from “0”.

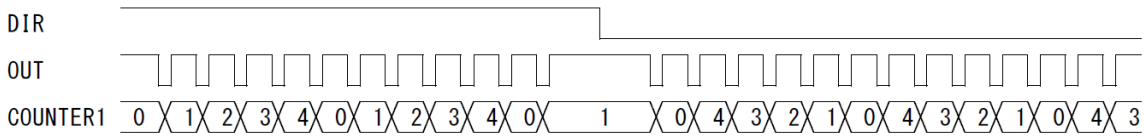
Set RENV3.C2RM = “1” and COUNTER 2 will be in the ring count mode. Then the LSI can perform the following operations.
- Count value will be “0” when the counter counts up from the value in RCMP2.
- Count value will be the count equals to the value in RCMP2 when the counter counts down from “0”.

Set COUNTER 1 to ring counter operation <RENV3.C1RM (bit 7)> 1: Operate COUNTER 1 as a ring counter.	[RENV3] (WRITE) 7 0 n - - - - - - -
Set COUNTER 2 to ring count operation <RENV3.C2RM (bit 11)> 1: Operate COUNTER 2 as a ring counter.	[RENV3] (WRITE) 15 8 - - - - - n - -

Even if the value for PRMV outside the range from 0 to the set value in RCMPn in positioning, the LSI will perform positioning operations.
When driving a rotating table with 3600 pulses per revolution, and when RCMP1 = 3599, RMD.MOD6~0 = 41h, and RMV = 7200, the table will rotate twice and the value in COUNTER 1, when stopped, will be the same as the value before starting.

Note: To use the ring counter function, change the count to a value between 0 and the value in RCMPn. If the value is outside the range above, the LSI will not operate normally.
Set the comparison conditions (RENV3.C1S1~0, RENV3.C2S1~0) to "00"b when using a counter as a ring counter.

Setting example
RENV3 = XXXXXX80h --- COUNTER 1 is in ring counter mode (RENV3.C1RM = “1”)
RCMP1 = 4 --- Count range: 0 ~ 4



11-11-3. Software limit function

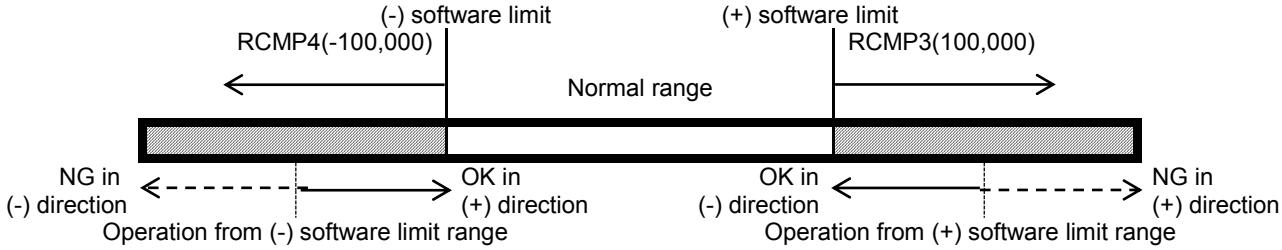
In addition to limit switch control by +EL signal or –EL signal, you can use software limit control using counters for software limit control. RCMP3 register is (+) software limit and RCMP4 register is (-) software limit.

When RENV3.SLCU="0" is set, counter 1 is selected and when RENV3.SLCU="1" is set, counter 2 is selected as a counter for software limit control. The counters are compared with (+) software limit (RCMP3 register) and (-) software limit (RCMP4 register). With limit function control (RENV3.SLM), event interrupt and error interrupt occurs at the timing that counter value for software limit function deviate from the range of software limit.

In order to occur event interrupt, set software limit function control to RENV3.SLM="01"b. Operation does not stop. When RIST.ISPS = "1" is set, event interrupt occurs at the timing the counter value exceed (+) software limit. When RIST.ISMS = "1", event interrupt occurs at the timing the counter value falls below (-) software limit.

In order to occur error interrupt, set software limit function control to RENV3.SLM = "10"b or RENV3.SLM = "11"b. When RENV3.SLM = "10"b is set, operation stops immediately. When RENV3.SLM="11"b, operation decelerates and stops. Even though RENV3.SLM="11"b is set, operation stops immediately in FL constant speed operation and FH constant speed operation. When REST.ESP="1" is set, error interrupt occurs at the timing the counter value exceed (+) software limit. When REST.ESMS="1", event interrupt occurs at the timing the counter value falls below (-) software limit.

PCL61x4 additional function control is set to RENV3.M614 = "1", (+) software limit detection is monitored by MSTSW.SCP3 and (-) software limit detection is monitored MSTSW.SCP4. You can monitor even if software limit function is set RENV3.SLM="00"b, out of relation to setting of software limit function control



[Setting example]
RENV3 = "0280000"h : Software limit monitor (M614) is enabled, counter 1 is selected for for software limit counter(SLCU) is selected
RCMP3 = 100,000 : (+) software limit value
RMCP4 = 100,000 : (-) software limit value

Software limit function control <RENV3.SLM1~0 (bits 23~22)> "00"b : Stops software limit function. "01"b : Event interrupt occurs at the location of software limit. (Operation does not stop.) "10"b : Operation stops immediately at the software limit and error interrupt occurs. "11"b : Operation decelerates and stops at the software limit and error interrupt occurs. (Operation stops immediately in FL constant speed operation and FH constant speed operation.)	[RENV3] (WRITE) 23 16 n n - - - - - -
Selection of counters for software limit control <RENV3.SLCU (bit 24)> 0 : COUNTER 1 is selected. 1 : COUNTER 2 is selected.	[RENV3] (WRITE) 31 24 - - - - - - - n
PCL61x4 additional function control <RENV3.M614 (bit 25)> 0 : Software limit status monitor (MSTSW.SCP3, MSTSW.SCP4) is disabled. 1 : Software limit status monitor (MSTSW.SCP3, MSTSW.SCP4) is enabled.	[RENV3] (WRITE) 31 24 - - - - - - n -

Event interrupt factor <RIST.ISPS (bit 15), RIST.ISMS (bit 14)> RIST.ISPS : Detection of (+) software limit RIST.ISMS : Detection of (-) software limit	[RIST] (READ) 15 8 n n - - - - - -
Error interrupt factor <REST.ESPS (bit 11), REST.ESMS (bit 10)> REST.ESPS : When stops by (+) software limit detection. REST.ESMS : When stops by (-) software limit detection.	[REST] (READ) 15 8 - - - - n n - -
Software limit status monitor <MSTSW.SCP3 (bit 11), MSTSW.SCP4 (bit 10)> MSTSW.SCP3 : Monitor comparative results of comparator for (+) software limit detection. MSTSW.SCP4 : Monitor comparative results of comparator for (-) software limit detection. (-)	[MSTSW] (READ) 15 8 - - - - n n - -

11-12. Synchronous starting

This LSI can perform the following operation by setting the PRMD 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 register. By setting the RIRQ register, an $\overline{INT}$ signal can be output at the same time the internal synchronous signal is output. You can determine the event interrupt factor by reading the RIST register. The operation status can be checked by reading the RSTS register.

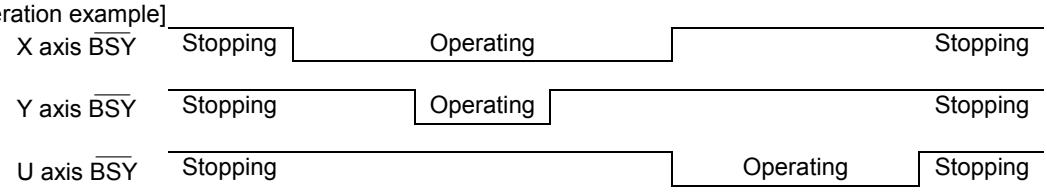
11-12-1. Start triggered by another axis stopping

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 remain stopped, the axis which is supposed to start when the conditions are met will start operation. A setting example for the above operation is shown here.
In the setting example, while the X axis (or Y axis) is working and the Y (or X) axis remains stopped, the U axis starts operation.

Specify the synchronous starting method <PRMD.MSY1~0 (bits 19~8)> "11"b: Start triggered by specified axis stopping.	[PRMD] (WRITE) 23 16 - - - - n n - -
Select an axis for confirming a stop (setting example) <PRMD.MAX3~0 (bits 23~20)> "0001"b: Start when the X axis stops "0010"b: Start when the Y axis stops "0100"b: Start when the Z axis stops "1000"b: Start when the U axis stops "0011"b: Start when both the X and Y axes have stopped "0101"b: Start when both the X and Z axes have stopped "1011"b: Start when the X, Y, and U axes have all stopped "1111"b: Start when all of the axes have stopped	[PRMD] (WRITE) 23 16 n n n n - - - -
Read the operation status <RSTS.CND3~0 (bits 3~0)> "0100"b: Wait for another axis to stop.	[RSTS] (READ) 7 0 - - - - n n n n

[Setting example]
After setting steps 1) ~ 3), start both the X and Y axes. When both the axes stop, the U axis will start.
1) Set PRMD.MSY1~0 (bits 19~18) for the U axis to "11"b. (Start triggered by another axis stopping)
2) Set PRMD.MAX3~0 (bits 23~20) for the U axis to "0011"b. (When both X axis and Y axis stops)
3) Write a start command for the U axis.

[Operation example]



When using continuous interpolation without changing the interpolated axes, all you need to do is to 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 interpolated axes, by using the pre-register function, you have to be careful. In this case, put "0" in the PRMV register 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 to [X: 10000, Y: 5000]. The Z axis performs a dummy interpolation operation with 0 feed amounts.
	PRIP	10000	10000	10000	
	PRMD	0000_0063h	0000_0063h	0000_0063h	
	Start command: Write STAFH (0751h) command				X ,Y and Z axes start command
2	PRMV	10000	0	-20000	Linear interpolation to[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 STAFH(0751h) command				X, Y, and Z axes start command (MSTSW.SPRF="1").

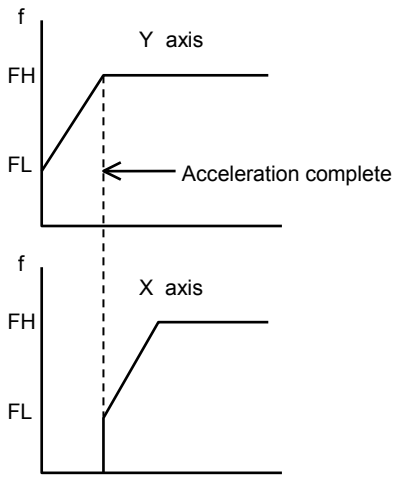
11-12-2. Start on an internal synchronous signal

This is a function to start an own axis 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 output internal synchronous signal to trigger its own axis to start.
The internal synchronization signal output has 6 types of timings. Select the timing with the RENV3 register.

Setting the synchronous start method <PRMD.MSY1~0 (bits 19~18)> "10"b: Start by the internal synchronize signal.	[PRMD] (WRITE) 23 16 - - - - n n - -
Setting the internal synchronous signal output timing <RENV3.SY03~1 (bits 19~16)> "0001"b: When the Comparator 1 conditions are met. "0010"b: When the Comparator 2 conditions are met. "1000"b: When acceleration starts. "1001"b: When acceleration ends. "1010"b: When deceleration starts. "1011"b: When deceleration ends. Others: Turn OFF the internal synchronous output.	[RENV3] (WRITE) 23 16 - - - - n n n n
Select the internal synchronous signal input <RENV3.SYI1~ 0 (bits 21~ 20)> "00"b: Use the internal synchronous signal output by the X axis. "01"b: Use the internal synchronous signal output by the Y axis. "10"b: Use the internal synchronous signal output by the Z axis. "11"b: Use the internal synchronous signal output by the U axis.	[RENV3] (WRITE) 23 16 - - n n - - - -
Reading the operation status <RSTS.CND3~0 (bits 3~0)> "0011"b: Waiting for an internal synchronous signal	[RSTS] (WRITE) 7 0 - - - - n n n n

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

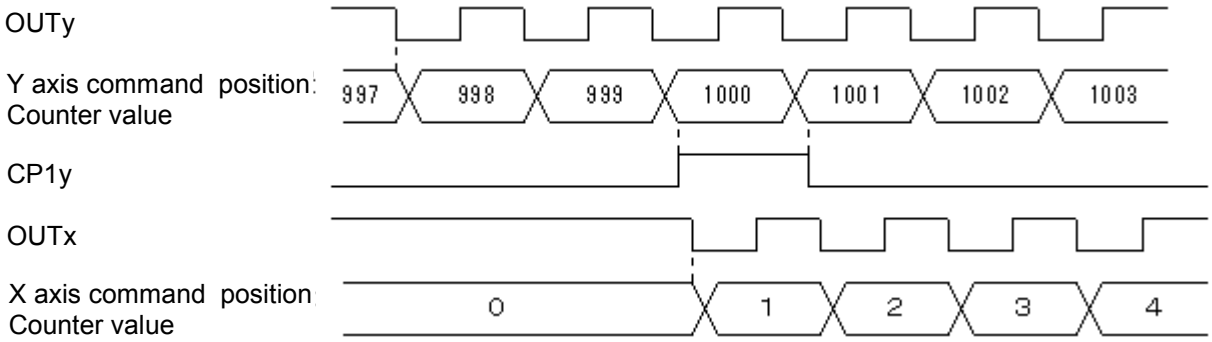
[Setting example 1]
After completing steps 1)~3) below, write a start command to the X and Y axes, the X axis will start when the Y axis completes its acceleration.
1) Set PRMD.MSY1~0 (bits 19~18) of X axis to "10"b. (Start with an internal synchronous signal)
2) Set RENV3.SYI1~0 (bits 21~20) of X axis to "01"b. (Use an internal synchronous signal from the Y axis.)
3) Set RENV3.SYO3~0 (bits 19~16) of Y axis to "1001"b. (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 as an internal synchronous signal.
Be careful that 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.

[Setting Example 2]
Use COUNTER 1 and RCMP1 to start the X axis when the Y axis = 1000.
1) Set PRMD.MSY1~0 (bits 19~18) of X axis to "10"b. (Start by an internal synchronous signal)
2) Set RENV3.SYI1~0 (bits 21~20) of X axis to "01"b. (Use an internal synchronous signal from the Y axis)
3) Set RENV3.SYO3~0 (bits 19~16) of Y axis to "0001"b. (Output an internal synchronous signal when the Comparator 1 conditions are satisfied)
4) Set RENV3.C1S1~0 (bits 13~12) of Y axis Ro "01"b. (Comparison method: RCMP1 data 1 = Comparison counter)
5) Set the RCMP1 value of the Y axis to "1000b". (Comparison counter value of Comparator 1 is "1000".)
6) Write start commands to the X and Y axes.

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



Note: In the example above, even if the Y axis 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."

11-13. Output an interrupt signal

This LSI can output an interrupt signal ($\overline{INT}$ signal). An $\overline{INT}$ signal is output triggered by 11 types of errors, 14 types of events, and change from operating to stop. The $\overline{INT}$ signal is output unconditionally triggered by error interrupt causes. Triggered by event interrupt factors, the $\overline{INT}$ signal is output under the condition set in the RIRQ register. Triggered by stop interrupt factors, the $\overline{INT}$ signal is output under the condition set in RENV2.IEND.

A stop interrupt is a simple interrupt function which produces an interrupt regardless of normal stop or error stop.
For a normal stop interrupt to be issued, the confirmation process is needed to read the RIST register as described in Event factor section. If your system needs only to detect 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 factors on all the axes that produced interrupts have been cleared. When function to stop automatic reset of reading is disabled (RENV2.MRST="0"), error interrupt factor (REST register) is cleared at writing of PREST(F2h) command. Event interrupt factor (RIST register) is cleared at writing "PRIST(F3h) command". Stop interrupt factor (MSTSW.SENI bit) is cleared at reading MSTSW status.
When function to stop automatic reset of reading is enabled (RENV2.MRST="1"), error interrupt factor (REST register) is cleared at writing of WREST(B2h) command. Event interrupt factor (RIST register) is cleared at writing WRIST (B3h) command. Stop interrupt factor (MSTSW.SENI bit) is cleared at writing SENIR (2Dh) command.

To determine which type of interrupt occurred, on which axis and the interrupt factor, 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 (MSTSW.SENI) is "1," a Stop interrupt has occurred.
   When RENV2.RSTS=1, SENIR(2Dh) command is performed and the corresponding bit is cleared.
3) If bit 4 (MSTSW.SERR) is "1," read the REST register to identify the factor of the error interrupt.
   When RENV2.MRST=1, WREST(B2h) command is performed and the corresponding bit is written "1" and cleared.
4) If bit 5 (MSTSW.SINT) is "1," read the RIST register to identify the cause of the event interrupt.
   When RENV2.MRST=1, WRIST(B3h) command is performed and the corresponding bit is written "1" and cleared.
5) Repeat steps 1) ~ 4) above for all axes other than X axis.

The steps above will allow you to evaluate the interrupt factor 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, in order to perform the interrupt routine, please perform PUSH/POP of I/F buffer.

Note 2: While processing all axes in steps 1) ~ 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, is set to edge triggering, the LSI will latch the $\overline{INT}$ output ON and it will not receive a new interrupt. Therefore, make sure that the main status of all axes should be read after you have reset CPU to be ready to receive the interrupt. After you confirm that the $\overline{INT}$ signal is not output from this LSI, the interrupt routine should be completed.

Note 3: When not using the $\overline{INT}$ terminal, leave it open.
When using more than one LSI, the $\overline{INT}$ terminals cannot be wired ORed. ($\overline{INT} \neq$ Hi-Z)

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

Read the interrupt status <MSTSW.SENI(bit 2), MSTSW.SERR (bit 4), MSTSW.SINT (bit 5)> SENI : With RENV2.IEND = “1”, becomes “1” when a stop interrupt occurs. SERR : Becomes “1” when an error interrupt occurs. SINT : Becomes “1” when an event interrupt occurs.	[MSTSW] (READ) 7 0 - - n n - n - -
Set the interrupt mask <RENV1.INTM (bit 29)> 1: Masks $\overline{\text{INT}}$ output.	[RENV1] (WRITE) 31 24 - - n - - - - -
Setting a stop interrupt <RENV2.IEDN (bit 30)> 1: Enable a stop interrupt.	[RENV2] (WRITE) 31 24 - n - - - - - -
Setting function to stop automatic reset of reading <RENV2.MRST(bit 31)> 1: Disabled. (Reset automatically.) 2: Enabled. (Does not reset automatically.) With serial bus I/F, function to stop automatic reset of reading is fixed to “Enabled” (RENV2.) reset of reading	[RENV2] (WRITE) 31 24 n - - - - - - -
Read cause of the error interrupt <RREST: Read out command> Copy the data in the REST register (error interrupt factor) to BUF.	[Read command] F2h
Read event interrupt cause <RRIST: Read out command> Copy the data in the RIST register (event interrupt factor) to BUF.	[Read command] F3h
Set the event interrupt cause <WRIRQ: Write command> Write the BUF data to the RIRQ register (event interrupt factor).	[Write command] ACh
Reset error interrupt factor <WREST: Write command> Write the BUF data to the REST register (reading of error interrupt factor). (When RENV2.MRST = “1” the corresponding bit is reset.)	[Write command] B2h
Reset event interrupt factor <WRIST: Write command> Write the BUF data to the RIST register (reading of event interrupt factor). (When RENV2.MRST = “1”, the corresponding bit is reset.)	[Write command] B3h
Reset stop interrupt factor <SENIR: Control command> Reset stop interrupt (MSTS.SENI bit). (When RENV2.MRST = “1”, MSTSW.SENI is reset.)	[Control command] 2Dh

[Error interrupt causes] <Detail of REST: Make the corresponding bit “1” to occur an interrupt.>

Error interrupt cause	Factor (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 $\overline{\text{CSTP}}$ input	3	ESSP
Stopped by turning ON the $\overline{\text{CEMG}}$ input	4	ESEM
Deceleration stopped by turning ON the SD input	5	ESSD
Stopped by an overflow occurrence of PA/PB input buffer counter	6	ESPO
An EA/EB input error occurred (the motor does not stop).	7	ESEE
A PA/PB input error occurred (the motor does not stop).	8	ESPE
Stopped by detecting (+) software limit (valid only when RENV3.SLM1=“1”.)	9	ESPS
Stopped by detecting (-) software limit (valid only when RENV3.SLM1=“1”.)	10	ESMS

[Event interrupt causes] < Make the corresponding bit “1” to set an interrupt cause and to occur interrupt.>

Event interrupt cause	Set cause (RIRQ)		Cause (RIST)	
	Bit	Bit name	Bit	Bit name
Normal 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 ORG input is turned ON	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 $\overline{\text{CSTA}}$ input is turned ON	12	IRSA	13	ISSA
Stopped by detecting (+) software limit (valid only when RENV3.SLM=1.)	-	-	14	ISPS
Stopped by detecting (-) software limit (valid only when RENV3.SLM = “01”.)	-	-	15	ISMS

12. Electrical Characteristics

12-1. Absolute maximum ratings

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

12-2. Recommended operating conditions

Item	Symbol	Rating	Unit	Remark
Power supply voltage	V_{DD}	3.0 ~ 3.6	V	-
Ambient temperature	T_J	-40 ~ +85	°C	No condensation

12-3. DC characteristics

(1) PCL6114

Item	Symbol	Condition	Min.	Max.	Unit
Consumption current	I_{dd1}	CLK = 30 MHz, 1 axis at 15 Mpps, no load	-	28	mA
Input capacity	-	-	-	10	pF
L level input current (V_{IL} = GND)	I_{IL}	$\overline{CS}$, $\overline{RD}$, $\overline{WR}$, A0~A4, D0~D15, CLK	-1	-	uA
		Input terminals other than the above (Note 1)	-125	-	
H level input current	I_{IH}	$V_{IH} = V_{DD}$	-	1	uA
		V_{IH} = 5.5 V	-	30	
L level input voltage	V_{IL}	-	-0.3	0.8	V
H level input voltage	V_{IH}	-	2.0	5.8	V
L level output voltage	V_{OL}	I_{OL} = 6 mA	-	0.4	V
H level output voltage	V_{OH}	I_{OH} = -6 mA	V_{DD} -0.4	-	V
L level output current	I_{OL}	V_{OL} = 0.4 V	-	6	mA
H level output current	I_{OH}	$V_{OH} = V_{DD}$ -0.4 V	-6	-	mA
Internal pull up resistance	R_{PU}	Other than $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, A0~A4, D0~D15, CLK	40	240	k-ohm

Note 1: Internal pull up 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).

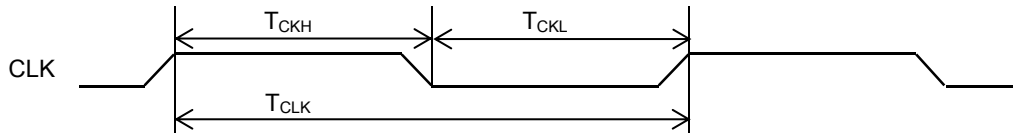
(2) PCL6144

Item	Symbol	Condition	Min.	Max.	Unit
Consumption current	I_{dd4}	CLK = 30 MHz, 4 axes at 15 Mpps, no load	-	129	mA
Input capacity	-	-	-	10	pF
L level input current (V_{IL} = GND)	I_{IL}	$\overline{CS}$, $\overline{RD}$, $\overline{WR}$, A0~A4, D0~D15, CLK	-1	-	uA
		Input terminals other than the above (Note 1)	-125	-	
H level input current	I_{IH}	$V_{IH} = V_{DD}$	-	1	uA
		V_{IH} = 5.5 V	-	30	
L level input voltage	V_{IL}	-	-0.3	0.8	V
H level input voltage	V_{IH}	-	2.0	5.8	V
L level output voltage	V_{OL}	I_{OL} = 6 mA	-	0.4	V
H level output voltage	V_{OH}	I_{OH} = -6 mA	V_{DD} -0.4	-	V
L level output current	I_{OL}	V_{OL} = 0.4 V	-	6	mA
H level output current	I_{OH}	$V_{OH} = V_{DD}$ -0.4 V	-6	-	mA
Internal pull up resistance	R_{PU}	Other than $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, A0~A4, D0~D15, CLK	40	240	k-ohm

Note 1: Internal pull up 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

12-4-1. Reference Clock



(1) PCL6114

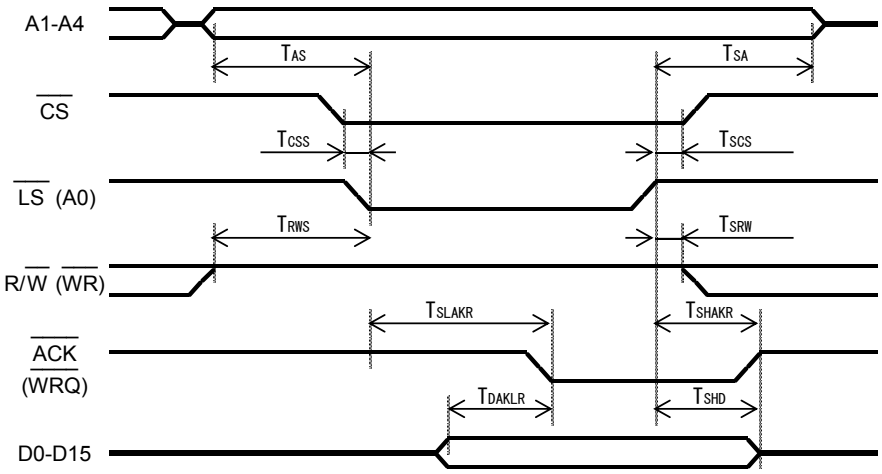
Item	Symbol	Condition	Min.	Max.	Unit
Reference clock frequency	f_{CLK}	-	-	30	MHz
Reference clock cycle	T_{CLK}	-	33	-	ns
Reference clock H level width	T_{CKH}	-	13	-	ns
Reference clock L level width	T_{CKL}	-	13	-	ns

(2) PCL6144

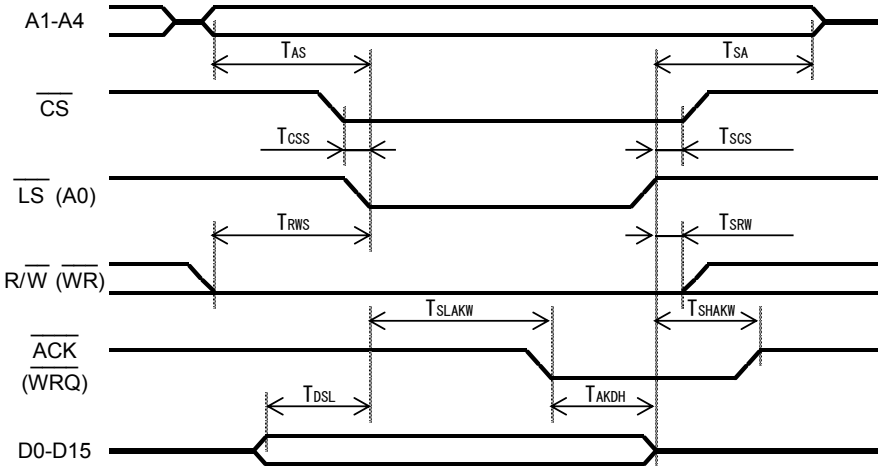
Item	Symbol	Condition	Min.	Max.	Unit
Reference clock frequency	f_{CLK}	-	-	30	MHz
Reference clock cycle	T_{CLK}	-	33	-	ns
Reference clock H level width	T_{CKH}	-	13	-	ns
Reference clock L level width	T_{CKL}	-	13	-	ns

12-4-2. 16-bit I/F-1) (IF1 = L, IF0 = L) 68000, etc

<Read cycle>



<Write cycle>



(1) PCL6114

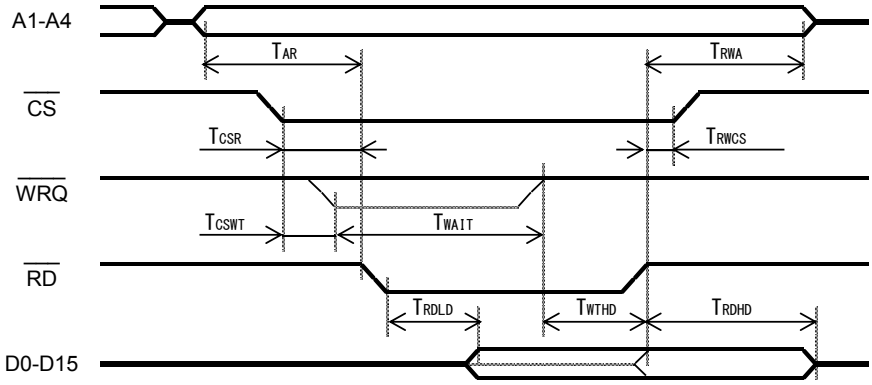
Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{LS}$ ↓	T_{AS}	-	7	-	ns
Address hold time for $\overline{LS}$ ↑	T_{SA}	-	0	-	ns
CS setup time for $\overline{LS}$ ↓	T_{CSS}	-	0	-	ns
CS hold time for $\overline{LS}$ ↑	T_{SCS}	-	1	-	ns
R/$\overline{W}$ setup time for $\overline{LS}$ ↓	T_{RWS}	-	0	-	ns
R/$\overline{W}$ hold time for $\overline{LS}$ ↑	T_{SRW}	-	0	-	ns
$\overline{ACK}$ ON delay time for $\overline{LS}$ ↓	T_{SLAKR}	C_L = 40pF	T_{CLK}	$4 \cdot T_{CLK}$ +16	ns
	T_{SLAKW}	C_L = 40pF	T_{CLK}	$4 \cdot T_{CLK}$ +16	ns
$\overline{ACK}$ OFF delay time for $\overline{LS}$ ↑	T_{SHAKR}	C_L = 40pF	-	15	ns
	T_{SHAKW}	C_L = 40pF	-	15	ns
Data output prior 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}	-	19	-	ns
Data hold time for $\overline{ACK}$ ↓	T_{AKDH}	-	0	-	ns

(2) PCL6144

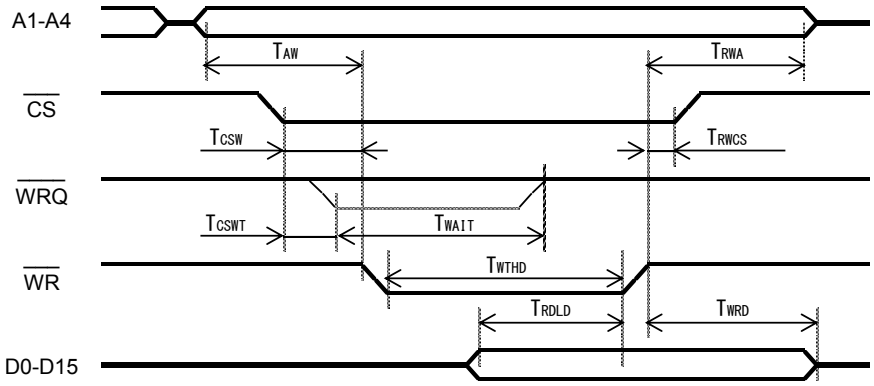
Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{LS}$ ↓	T_{AS}	-	10	-	ns
Address hold time for $\overline{LS}$ ↑	T_{SA}	-	0	-	ns
CS setup time for $\overline{LS}$ ↓	T_{CSS}	-	1	-	ns
CS hold time for $\overline{LS}$ ↑	T_{SCS}	-	0	-	ns
R/$\overline{W}$ setup time for $\overline{LS}$ ↓	T_{RWS}	-	0	-	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}	$4 \cdot T_{CLK}$ +18	ns
	T_{SLAKW}	C_L = 40pF	T_{CLK}	$4 \cdot T_{CLK}$ +18	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 prior time for $\overline{ACK}$ ↓	T_{DAKLR}	C_L = 40pF	T_{CLK}	-	ns
Data float delay time for $\overline{LS}$ ↑	T_{SHD}	C_L = 40pF	-	20	ns
Data setup time for $\overline{LS}$ ↑	T_{DSL}	-	20	-	ns
Data hold time for $\overline{ACK}$ ↓	T_{AKDH}	-	0	-	ns

12-4-3. 16-bit I/F-2 (IF1=L, IF0=H) H8

<Read cycle>



<Write cycle>



(1) PCL6114

Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{RD}$ ↓	T_{AR}	-	7	-	ns
Address setup time for $\overline{WR}$ ↓	T_{AW}	-	7	-	ns
Address hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWA}	-	0	-	ns
$\overline{CS}$ setup time for $\overline{RD}$↓	T_{CSR}	-	0	-	ns
$\overline{CS}$ setup time for $\overline{WR}$ ↓	T_{CSW}	-	0	-	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	-	12	ns
$\overline{WRQ}$ signal L level time	T_{WAIT}	-	-	$4 \cdot T_{CLK}$	ns
Data output delay time for $\overline{RD}$ ↓	T_{RDLD}	C_L = 40pF	-	22	ns
Data output delay time for $\overline{WRQ}$ ↑	T_{WTHD}	C_L = 40pF	-	8	ns
Data float delay time for $\overline{RD}$ ↑	T_{RDHD}	C_L = 40pF	-	17	ns
$\overline{WR}$ signal width	T_{WR}	Note 1	11	-	ns
Data setup time for $\overline{WR}$ ↑	T_{DWR}	-	14	-	ns
Data hold time for $\overline{WR}$ ↑	T_{WRD}	-	0	-	ns

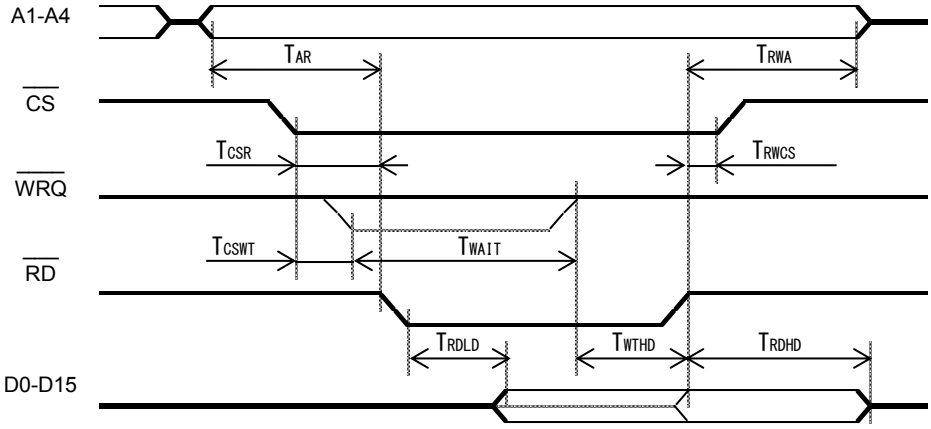
(2) PCL6144

Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{RD}$ ↓	T_{AR}	-	11	-	ns
Address setup time for $\overline{WR}$ ↓	T_{AW}	-	11	-	ns
Address hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWA}	-	0	-	ns
$\overline{CS}$ setup time for $\overline{RD}$↓	T_{CSR}	-	1	-	ns
$\overline{CS}$ setup time for $\overline{WR}$ ↓	T_{CSW}	-	1	-	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	-	14	ns
$\overline{WRQ}$ signal L level time	T_{WAIT}	-	-	$4 \cdot T_{CLK}$	ns
Data output delay time for $\overline{RD}$ ↓	T_{RDLD}	C_L = 40pF	-	23	ns
Data output delay time for $\overline{WRQ}$ ↑	T_{WTHD}	C_L = 40pF	-	8	ns
Data float delay time for $\overline{RD}$ ↑	T_{RDHD}	C_L = 40pF	-	19	ns
$\overline{WR}$ signal width	T_{WR}	Note 1	12	-	ns
Data setup time for $\overline{WR}$ ↑	T_{DWR}	-	14	-	ns
Data hold time for $\overline{WR}$ ↑	T_{WRD}	-	0	-	ns

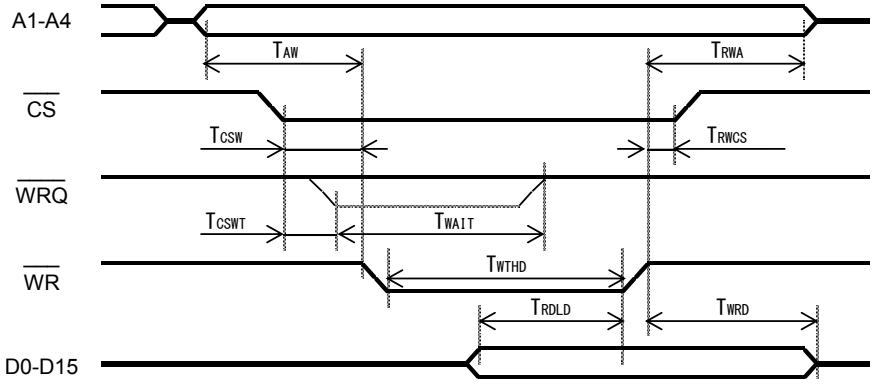
Note 1: When a $\overline{WRQ}$ signal is output, the time will be the interval between $\overline{WRQ}$ = H level and $\overline{WR}$ = H level.

12-4-4. 16-bit I/F-3 (IF1=H, IF0=L) 8086 etc.

<Read cycle>



<Write cycle>



(1) PCL6114

Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{RD}$ ↓	T_{AR}	-	7	-	ns
Address setup time for $\overline{WR}$ ↓	T_{AW}	-	7	-	ns
Address hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWA}	-	0	-	ns
$\overline{CS}$ setup time for $\overline{RD}$↓	T_{CSR}	-	0	-	ns
$\overline{CS}$ setup time for $\overline{WR}$ ↓	T_{CSW}	-	0	-	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	-	12	ns
$\overline{WRQ}$ signal L level time	T_{WAIT}	-	-	$4 \cdot T_{CLK}$	ns
Data output delay time for $\overline{RD}$ ↓	T_{RDLD}	C_L = 40pF	-	22	ns
Data output delay time for $\overline{WRQ}$ ↑	T_{WTHD}	C_L = 40pF	-	8	ns
Data float delay time for $\overline{RD}$ ↑	T_{RDHD}	C_L = 40pF	-	17	ns
$\overline{WR}$ signal width	T_{WR}	Note 1	11	-	ns
Data setup time for $\overline{WR}$ ↓	T_{DWR}	-	14	-	ns
Data hold time for $\overline{WR}$ ↑	T_{WRD}	-	0	-	ns

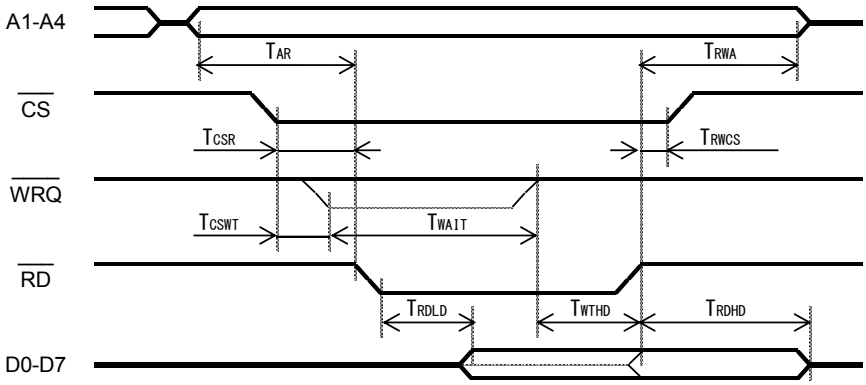
(2) PCL6144

Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{RD}$ ↓	T_{AR}	-	11	-	ns
Address setup time for $\overline{WR}$ ↓	T_{AW}	-	11	-	ns
Address hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWA}	-	0	-	ns
$\overline{CS}$ setup time for $\overline{RD}$↓	T_{CSR}	-	1	-	ns
$\overline{CS}$ setup time for $\overline{WR}$ ↓	T_{CSW}	-	1	-	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	-	14	ns
$\overline{WRQ}$ signal L level time	T_{WAIT}	-	-	$4 \cdot T_{CLK}$	ns
Data output delay time for $\overline{RD}$ ↓	T_{RDLD}	C_L = 40pF	-	23	ns
Data output delay time for $\overline{WRQ}$ ↑	T_{WTHD}	C_L = 40pF	-	8	ns
Data float delay time for $\overline{RD}$ ↑	T_{RDHD}	C_L = 40pF	-	19	ns
$\overline{WR}$ signal width	T_{WR}	Note 1	12	-	ns
Data setup time for $\overline{WR}$ ↓	T_{DWR}	-	14	-	ns
Data hold time for $\overline{WR}$ ↑	T_{WRD}	-	0	-	ns

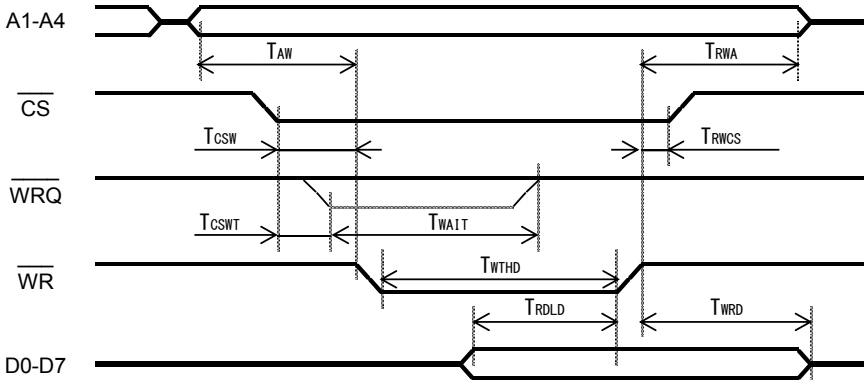
Note 1: When a $\overline{WRQ}$ signal is output, the time will be the interval between $\overline{WRQ}$ = H level and $\overline{WR}$ = H level.

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

<Read cycle>



<Write cycle>



(1) PCL6114

Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{RD}$ ↓	T_{AR}	-	7	-	ns
Address setup time for $\overline{WR}$ ↓	T_{AW}	-	7	-	ns
Address hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWA}	-	0	-	ns
$\overline{CS}$ setup time for $\overline{RD}$↓	T_{CSR}	-	0	-	ns
$\overline{CS}$ setup time for $\overline{WR}$ ↓	T_{CSW}	-	0	-	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	-	12	ns
$\overline{WRQ}$ signal L level time	T_{WAIT}	-	-	$4 \cdot T_{CLK}$	ns
Data output delay time for $\overline{RD}$ ↓	T_{RDLD}	C_L = 40pF	-	16	ns
Data output delay time for $\overline{WRQ}$ ↑	T_{WTHD}	C_L = 40pF	-	4	ns
Data float delay time for $\overline{RD}$ ↑	T_{RDHD}	C_L = 40pF	-	12	ns
$\overline{WR}$ signal width	T_{WR}	Note 1	11	-	ns
Data setup time for $\overline{WR}$ ↑	T_{DWR}	-	14	-	ns
Data hold time for $\overline{WR}$ ↑	T_{WRD}	-	0	-	ns

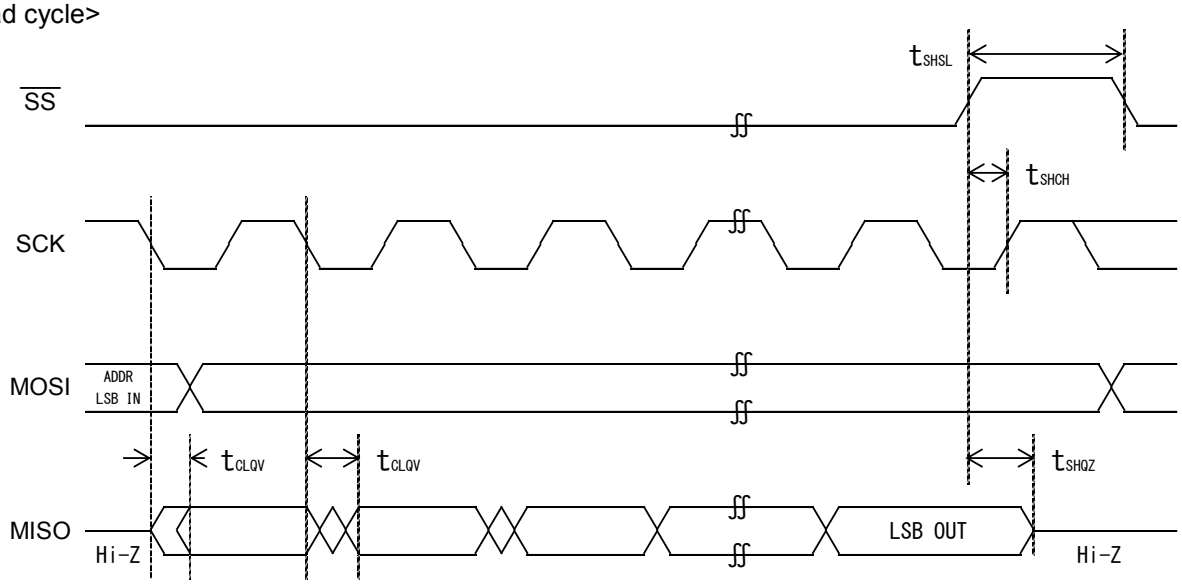
(2) PCL6144

Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{RD}$ ↓	T_{AR}	-	11	-	ns
Address setup time for $\overline{WR}$ ↓	T_{AW}	-	11	-	ns
Address hold time for $\overline{RD}$, $\overline{WR}$ ↑	T_{RWA}	-	0	-	ns
$\overline{CS}$ setup time for $\overline{RD}$↓	T_{CSR}	-	1	-	ns
$\overline{CS}$ setup time for $\overline{WR}$ ↓	T_{CSW}	-	1	-	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	-	14	ns
$\overline{WRQ}$ signal L level time	T_{WAIT}	-	-	$4 \cdot T_{CLK}$	ns
Data output delay time for $\overline{RD}$ ↓	T_{RDLD}	C_L = 40pF	-	18	ns
Data output delay time for $\overline{WRQ}$ ↑	T_{WTHD}	C_L = 40pF	-	3	ns
Data float delay time for $\overline{RD}$ ↑	T_{RDHD}	C_L = 40pF	-	14	ns
$\overline{WR}$ signal width	T_{WR}	Note 1	12	-	ns
Data setup time for $\overline{WR}$ ↑	T_{DWR}	-	14	-	ns
Data hold time for $\overline{WR}$ ↑	T_{WRD}	-	0	-	Ns

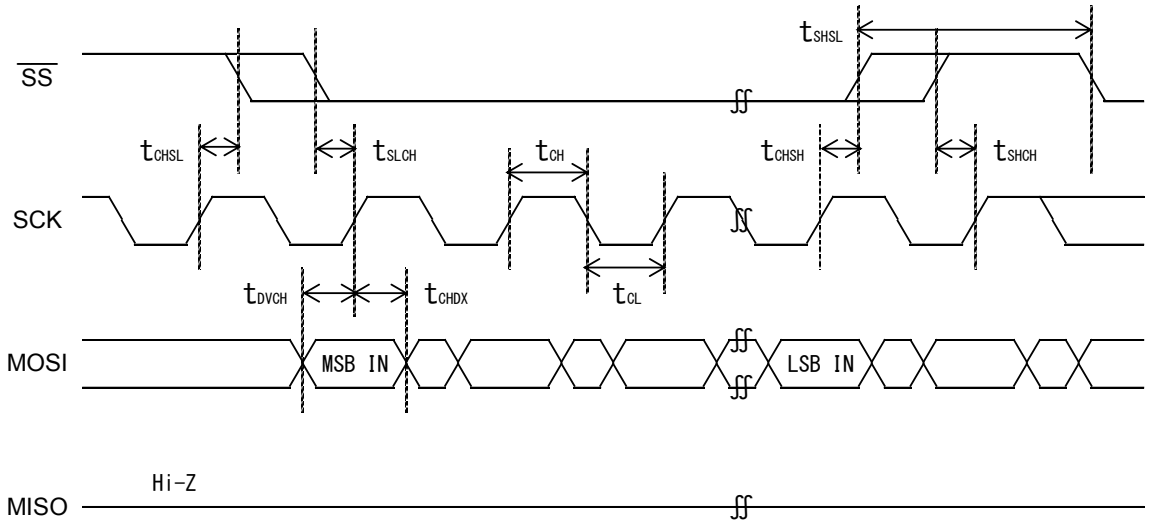
Note 1: When a $\overline{WRQ}$ signal is output, the time will be the interval between $\overline{WRQ}$ = H level and $\overline{WR}$ = H level.

12-4-6.Serial I/F (RD=L, WR=L)

<Read cycle>



<Write cycle>



(1) PCL6114

Item	Symbol	Condition	Min.	Max.	Unit
Serial clock	f_C	-	-	F_{CLK} / 1.5	MHz
Serial clock H time	t_{CH}	-	20	-	ns
Serial clock L time	t_{CL}	-	20	-	ns
$\overline{SS}$ active set up	t_{SLCH}	-	2	-	ns
$\overline{SS}$ non-active set up	t_{SHCH}	-	1	-	ns
$\overline{SS}$ deselect time	t_{SHSL}	-	$3 \cdot T_{CLK}$	-	ns
$\overline{SS}$ active hold time	t_{CHSH}	-	3	-	ns
$\overline{SS}$ non-active hold time	t_{CHSL}	-	3	-	ns
Data set up time	t_{DVCH}	-	2	-	ns
Data hold time	t_{CHDX}	-	4	-	ns
Output disable time	t_{SHQZ}	C_L=40pF	-	16	ns
Output delay time	t_{CLQV}	C_L=40pF	-	15	ns

(2) PCL6144

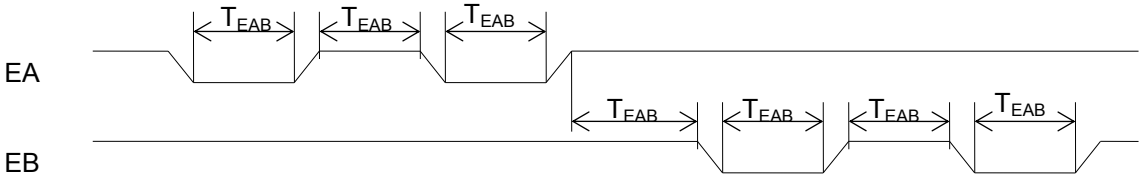
Item	Symbol	Condition	Min.	Max.	Unit
Serial clock	f_C	-	-	F_{CLK} / 1.5	MHz
Serial clock H time	t_{CH}	-	20	-	ns
Serial clock L time	t_{CL}	-	20	-	ns
$\overline{SS}$ active set up	t_{SLCH}	-	4	-	ns
$\overline{SS}$ non-active set up	t_{SHCH}	-	1	-	ns
$\overline{SS}$ deselect time	t_{SHSL}	-	$3 \cdot T_{CLK}$	-	ns
$\overline{SS}$ active hold time	t_{CHSH}	-	4	-	ns
$\overline{SS}$ non-active hold time	t_{CHSL}	-	5	-	ns
Data set up time	t_{DVCH}	-	3	-	ns
Data hold time	t_{CHDX}	-	6	-	ns
Output disable time	t_{SHQZ}	C_L= pF	-	16	ns
Output delay time	t_{CLQV}	C_L= pF	-	21	ns

12-5. Operation timing (common for all axes)

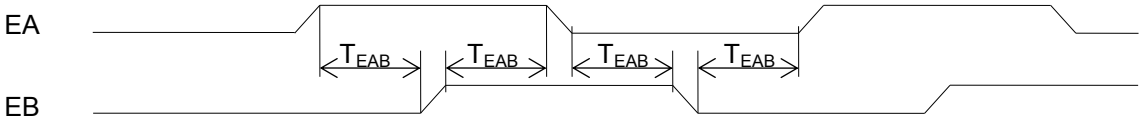
Item	Symbol	Condition	Min.	Max.	Unit
$\overline{RST}$ input signal width	-	Note 1	$8 \cdot T_{CLK}$	-	ns
EA, EB, EZ input signal width	T_{EAB}	RENV2: EINF="0"	T_{CLK}	-	ns
		RENV2: EINF="1"	$3 \cdot T_{CLK}$	-	
PA, PB input signal width	T_{PAB}	RENV2: PINF="0"	T_{CLK}	-	ns
		RENV2: PINF="1"	$3 \cdot T_{CLK}$	-	
ERC output signal width	-	RENV1 : EPW = "000"b	$225 \cdot T_{CLK}$	$240 \cdot T_{CLK}$	ns
		RENV1 : EPW = "001"b	$1793 \cdot T_{CLK}$	$1920 \cdot T_{CLK}$	
		RENV1 : EPW = "010"b	$7169 \cdot T_{CLK}$	$7680 \cdot T_{CLK}$	
		RENV1 : EPW = "011"b	$28673 \cdot T_{CLK}$	$30720 \cdot T_{CLK}$	
		RENV1 : EPW = "100"b	$229377 \cdot T_{CLK}$	$245760 \cdot T_{CLK}$	
		RENV1 : EPW = "101"b	$917505 \cdot T_{CLK}$	$983040 \cdot T_{CLK}$	
		RENV1 : EPW = "110"b	$1835009 \cdot T_{CLK}$	$1966080 \cdot T_{CLK}$	
		RENV1 : EPW = "111"b	(Level output)		
Time of ERC signal OFF timer	-	RENV1 : ETW = "01"b	$225 \cdot T_{CLK}$	$240 \cdot T_{CLK}$	ns
		RENV1 : ETW = "10"b	$28673 \cdot T_{CLK}$	$30720 \cdot T_{CLK}$	
		RENV1 : ETW = "11"b	$1835009 \cdot T_{CLK}$	$1966080 \cdot T_{CLK}$	
+EL, -EL, SD, ORG, ALM, INP, CEMG input signal width	-	RENV1 : FLTR = "0"	T_{CLK}	-	ns
		FLTR="1" & FTM="00"b	$64 \cdot T_{CLK}$	-	
		FLTR="1" & FTM="01"b	$512 \cdot T_{CLK}$	-	
		FLTR="1" & FTM="10"b	$4096 \cdot T_{CLK}$	-	
		FLTR="1" & FTM="11"b	$32768 \cdot T_{CLK}$	-	
+DR(PA), -DR(PB), $\overline{PE}$ input signal width	-	RENV1 : DRF = "0"	T_{CLK}	-	ns
		RENV1 : DRF = "1"	$1048576 \cdot T_{CLK}$	-	
Time of direction change timer	-	RENV1 : DTMF = "0" (ON)	$3585 \cdot T_{CLK}$	$3840 \cdot T_{CLK}$	ns
PCS input signal width	-	-	T_{CLK}	-	ns
LTC input signal width	-	-	T_{CLK}	-	ns
$\overline{CSTA}$ Output signal width	-	-	$8 \cdot T_{CLK}$		ns
$\overline{CSTA}$ Input signal width	-	-	$4 \cdot T_{CLK}$	-	ns
$\overline{CSTP}$ Output signal width	-	-	$8 \cdot T_{CLK}$		ns
$\overline{CSTP}$ Input signal width	-	-	$4 \cdot T_{CLK}$	-	ns
$\overline{BSY}$ signal ON delay time	T_{CMDBSY}	-	$4 \cdot T_{CLK}$	$5 \cdot T_{CLK}$	ns
	T_{STABSY}	-	$4 \cdot T_{CLK}$	$5 \cdot T_{CLK}$	ns
Start delay time	T_{CMDPLS}	-	$15 \cdot T_{CLK}$	$16 \cdot T_{CLK}$	ns
	T_{STAPLS}	-	$15 \cdot T_{CLK}$	$16 \cdot T_{CLK}$	ns
Deceleration delay time	T_{CMDFDW}	-	$5 \cdot T_{CLK}$	$6 \cdot T_{CLK}$	ns
	T_{SDFDW}	-	$2 \cdot T_{CLK}$	$3 \cdot T_{CLK}$	ns

Note 1: Longer than 8 cycles CLK signal is needed to be input while $\overline{RST}$ = L level.

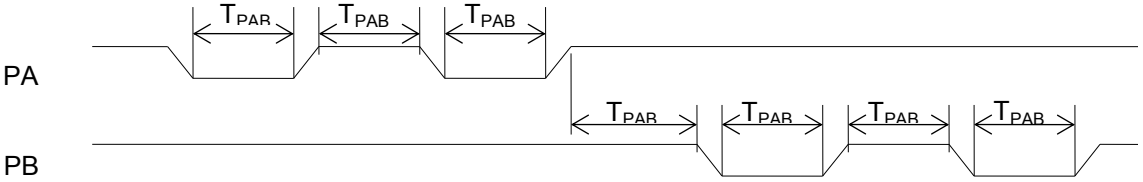
1) When the EA, EB inputs are in the Two-pulse mode



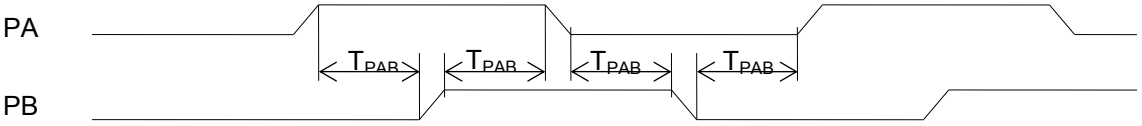
2) When the EA, EB inputs are in the 90-degree phase difference mode



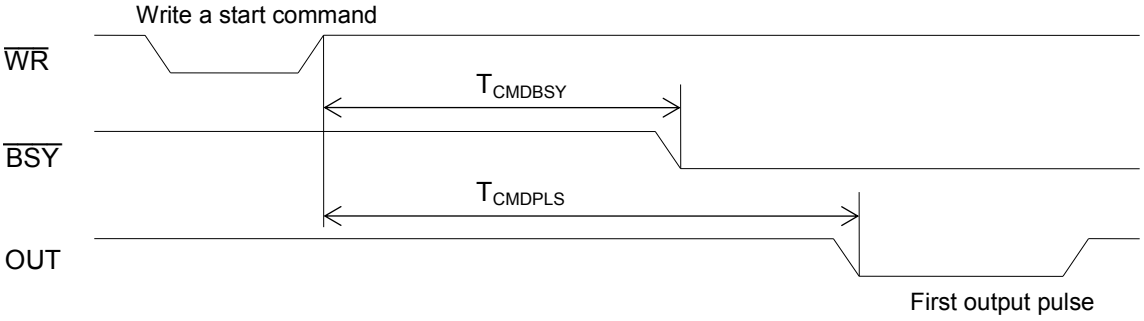
3) When the PA, PB inputs are in the Two-pulse mode



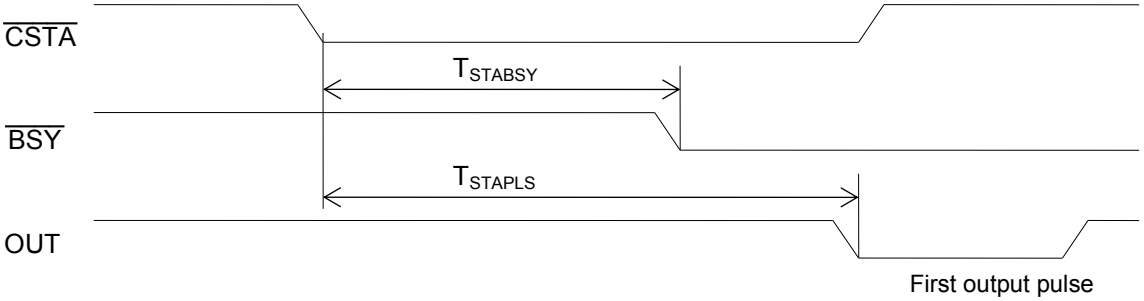
4) When the PA, PB inputs are in the 90-degree phase difference mode



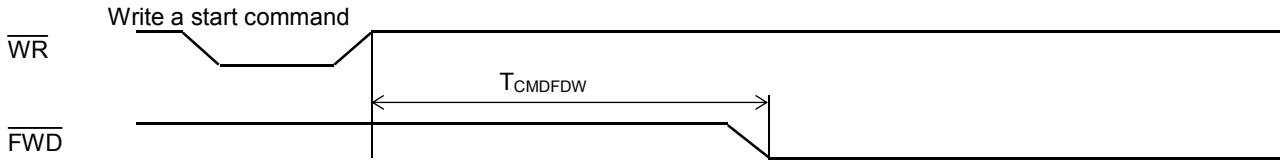
5) Timing for the command start (when I/M = H, and B/$\overline{W}$ = H)



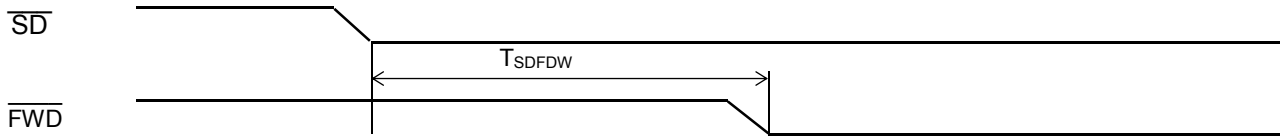
6) Simultaneous start timing



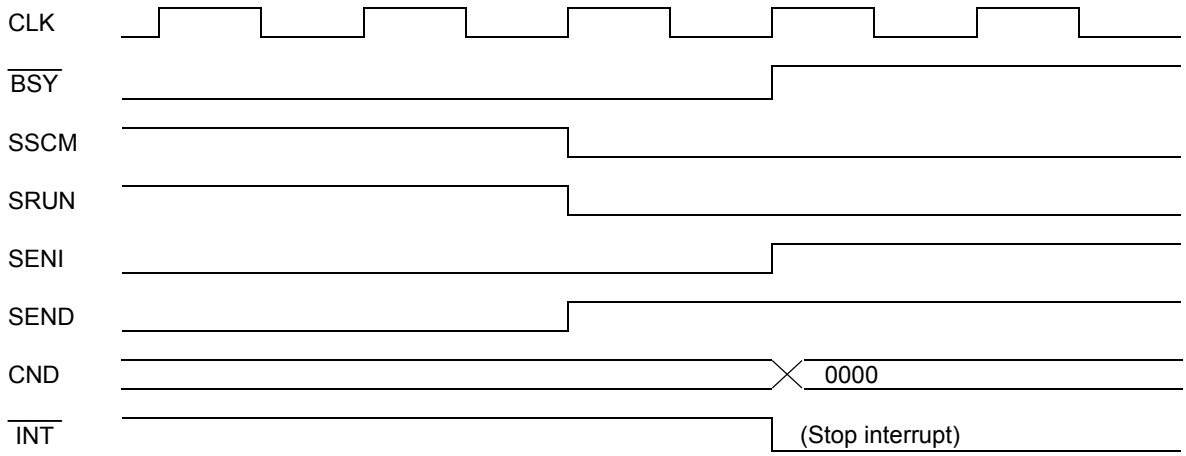
7) Deceleration start timing triggered by a command



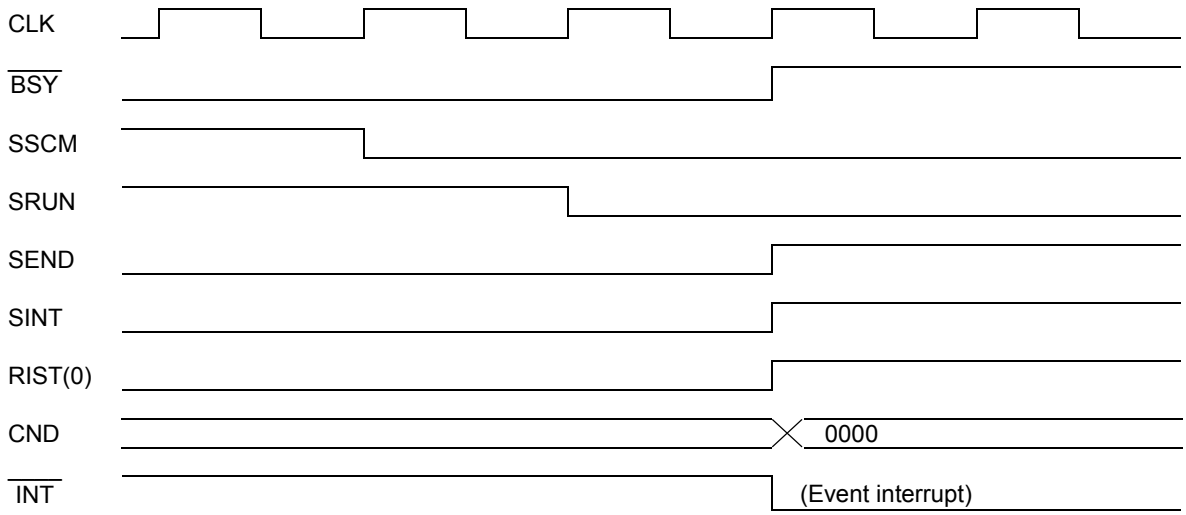
8) Deceleration start timing triggered by the SD input



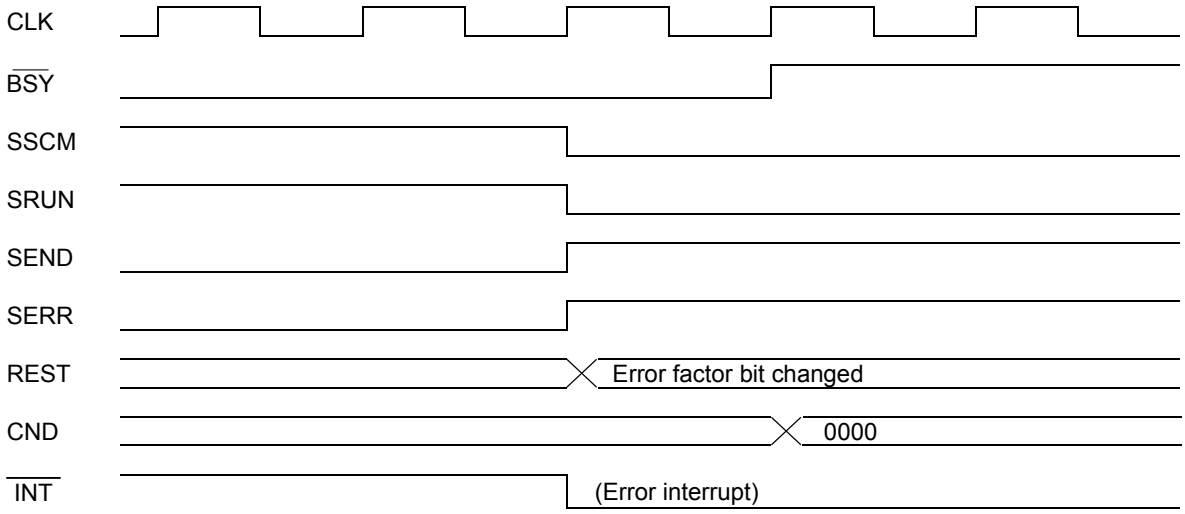
9) Stop timing by a command



10) Stop timing by normal stop



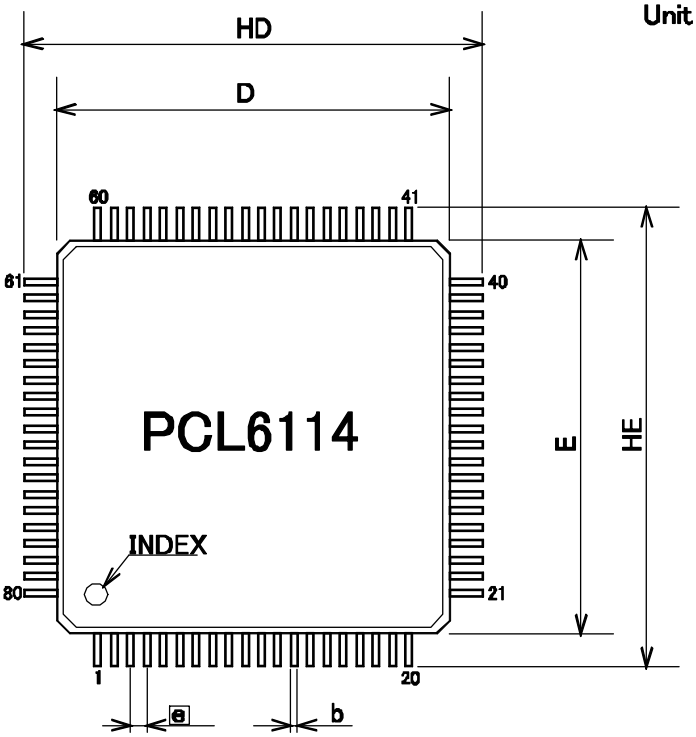
11) Stop timing by error

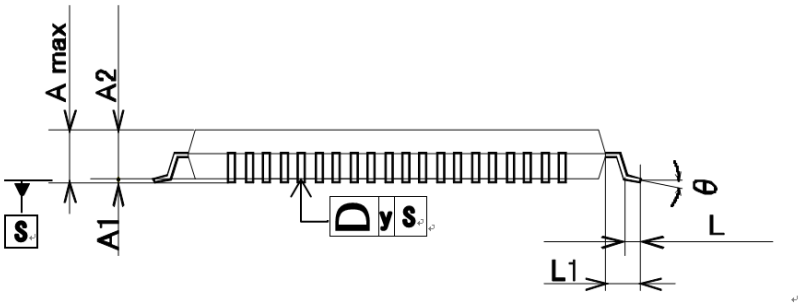


13. External Dimensions

(1) PCL6114

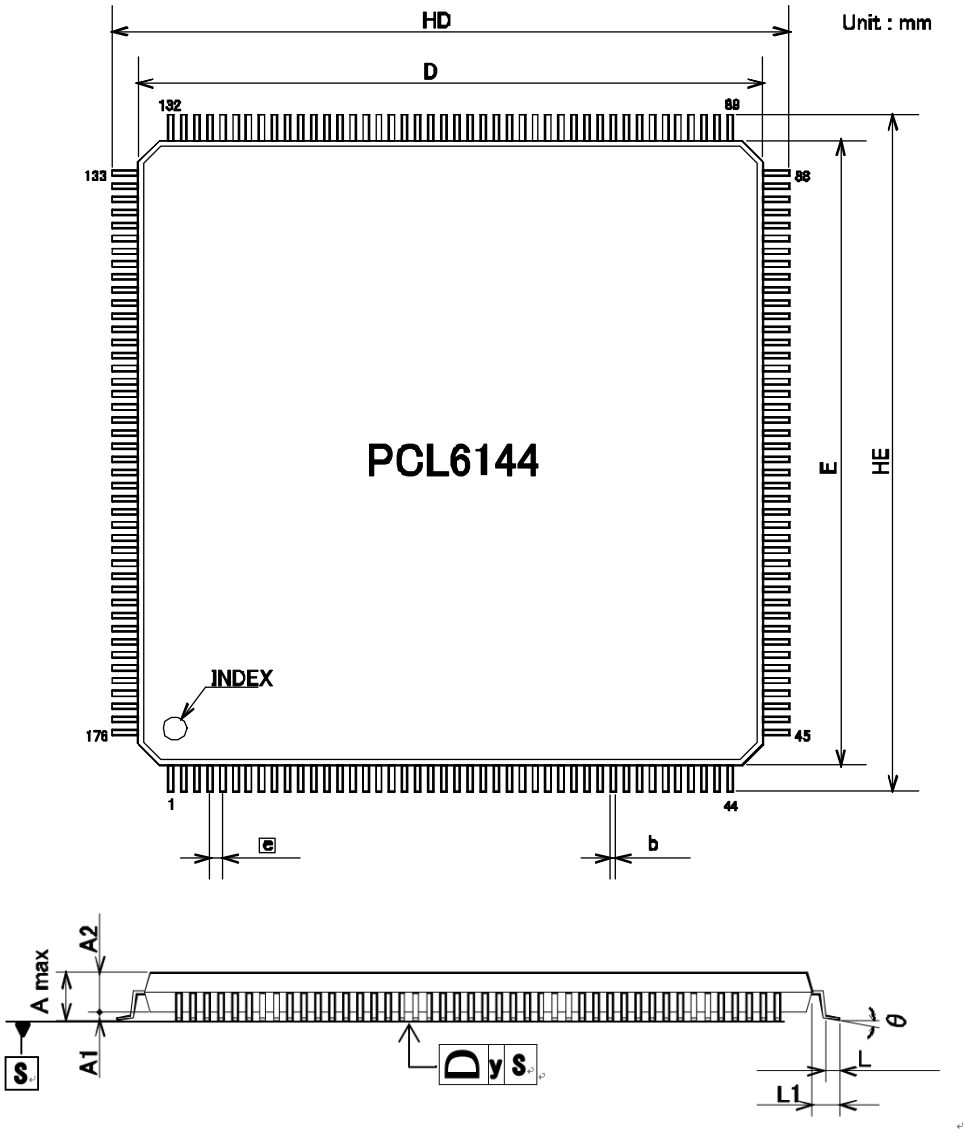
Unit : mm





Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
E	-	12	-
D	-	12	-
HE	-	14	-
HD	-	14	-
e		0.5	
b	0.13	-	0.27
A max	-	-	1.7
A1	-	0.1	-
A2	-	1.4	-
L	0.3	-	0.75
L1	-	1	-
θ	0°	-	10°
y	-	-	0.08

(2) PCL6144



Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
E	-	24	-
D	-	24	-
HE	-	26	-
HD	-	26	-
e		0.5	
b	0.17	-	0.27
A max	-	-	1.7
A1	-	0.1	-
A2	-	1.4	-
L	0.3	-	0.75
L1	-	1	-
θ	0°	-	10°
y	-	-	0.08

14. Command lists

14-1. Operation commands

COMB0	Symbol	Description	COMB0	Symbol	Description
05h	CMEMG	Emergency stop	50h	STAFL	FL constant speed start
06h	CMSTA	$\overline{\text{CSTA}}$ output (simultaneous start)	51h	STAFH	FH constant speed start
07h	CMSTP	$\overline{\text{CSTP}}$ output (simultaneous stop)	52h	STAD	High speed start 1 (FH constant speed → Deceleration stop)
40h	FCHGL	Immediate change to FL constant speed	53h	STAUD	High speed start 2 (acceleration → FH constant speed → Deceleration stop)
41h	FCHGH	Immediate change to FH constant speed	54h	CNTFL	FL constant speed start for remaining number of pulses
42h	FSCHL	Decelerate to FL speed	55h	CNTFH	FH constant 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	-	-	-

14-2. General-purpose port control commands

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

14-3. Control commands

COMB0	Symbol	Description	COMB0	Symbol	Description
00h	NOP	(Invalid command)	26h	PRECAN	Cancel the pre-register
04h	SRST	Software reset	28h	STAON	Substitute PCS input
20h	CUN1R	Reset COUNTER 1	29h	LTCH	Substitute LTC input
21h	CUN2R	Reset COUNTER 2	2Ah	SPSTA	Uses the same process as the $\overline{\text{CSTA}}$ input only for its own axis
24h	ERCOUT	Output an ERC signal	2Dh	SENIR	Reset MSTSW.SENI bit
25h	ERCRST	Reset an ERC signal	2Eh	SEORR	Reset MSTSW.SEOR bit

14-4. 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	32	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	16	RUR	D3h	RRUR	93h	WRUR	PRUR	C3h	RPRUR	83h	WPRUR
5	Deceleration rate	16	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 feed data	32	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	32	RENV2	DDh	RRENV2	9Dh	WRENV2	-	-	-	-	-
14	Environment setting 3	26	RENV3	DEh	RRENV3	9Eh	WRENV3	-	-	-	-	-
15	COUNTER 1 (command)	32	RCUN1	E3h	RRCUN1	A3h	WRCUN1	-	-	-	-	-
16	COUNTER 2 (mechanical)	32	RCUN2	E4h	RRCUN2	A4h	WRCUN2	-	-	-	-	-
17	Comparison data for comparator 1	32	RCMP1	E7h	RRCMP1	A7h	WRCMP1	-	-	-	-	-
18	Comparison data for comparator 2	32	RCMP2	E8h	RRCMP2	A8h	WRCMP2	-	-	-	-	-
19	Comparison data for comparator 3 (+) software limit	32	RCMP3	E9h	RRCMP3	A9h	WRCMP3	-	-	-	-	-
20	Comparison data for comparator 3 (-) software limit	32	RCMP4	EAh	RRCMP4	AAh	WRCMP4	-	-	-	-	-
21	Event interrupt factor setting	13	RIRQ	ECh	RRIRQ	ACh	WRIRQ	-	-	-	-	-
22	COUNTER 1 latched data	32	RLTC1	EDh	RRLTC1	-	-	-	-	-	-	-
23	COUNTER 2 latched data	32	RLTC2	EEh	RRLTC2	-	-	-	-	-	-	-
24	Extension status	17	RSTS	F1h	RRSTS	-	-	-	-	-	-	-
25	Get error interrupt factor	11	REST	F2h	RREST	B2h	WREST	-	-	-	-	-
26	Get event interrupt factor status	16	RIST	F3h	RRIST	B3h	WRIST	-	-	-	-	-
27	Positioning counter	32	RPLS	F4h	RRPLS	-	-	-	-	-	-	-
28	EZ counter, current speed monitor	20	RSPD	F5h	RRSPD	-	-	-	-	-	-	-
29	Ramping-down point setting value	24	PSDC	F6h	RPSDC	-	-	-	-	-	-	-

The followings are common registers. (These are used with serial bus I/F.)

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	Setting of general purpose (GP0~15) specifications	16	RGPM	FAh	RRGPM	BAh	WRGPM	-	-	-	-	-
2	Setting of general purpose (GP0~15) data	16	RGPD	FBh	RRGPD	BBh	WRGPD	-	-	-	-	-

* Common registers can be accessed from any axes. The priority of setting at writing for several axes in block is X>Y>Z>U.

15. Handling Precautions

15-1. 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 +3.3 V 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.

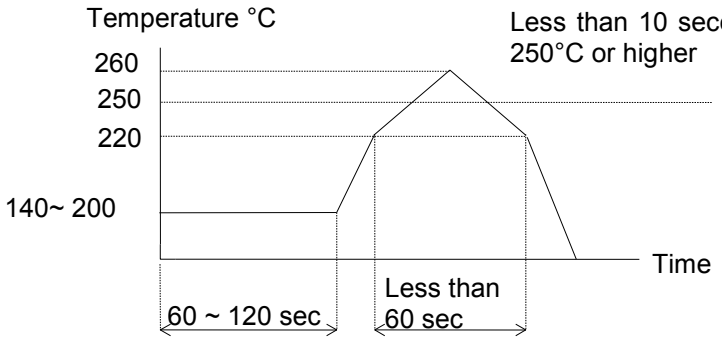
15-2. Precautions for transporting and storing LSIs

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.

15-3. 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 ± 5 °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.



[Allowable temperature 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.

15-4. 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.

May 15, 2015

No. DA70136-1/2E1