Reference

User's Manual For PCL6046 Pulse Control LSI

Preliminary

NPM

Nippon Pulse Motor Co., Ltd.

[Preface]

Thank you for considering our pulse control LSI, the "PCL6046."
To learn how to use the PCL6046, read this manual to become familiar with the product.
The handling precautions for installing this LSI are described at the end of this manual. Make sure to read them before installing the LSI.

[Cautions]

(2) The specifications of this LSI may be changed to improve performance or quality without prior notice.

(3) Although this manual was produced with the utmost care, if you find any points that are unclear, wrong, or have inadequate descriptions, please let us know.

(4) We are not responsible for any results that occur from using this LSI, regardless of item (3) above.

■ Explanation of the descriptions in this manual

1. The "x" "y" "z" and "u" of terminal names and bit names refer to the X axis, Y axis, Z axis and U axis, respectively.

2. Terminals with a # (ex #RST) are negative logic. Their logic cannot be changed. Terminals without a # are positive logic. Their output logic can be changed.

3. When describing the bits in registers, "n" refers to the bit position. A "0" means that the bit is in position 0 and that it is prohibited to write to any bit other than "0". Finally, this bit will always return a "0" when read out.

4. Specified bit of specified register is referred to as (register name).(bit name). (ex. RMD.MSDE)

INDEX

1. Outline and Features

1-1. Outline

The PCL6046 is a CMOS LSI designed to provide the oscillating, high-speed pulses needed to drive stepper motors and servomotors (pulse string input types) using various commands.
It can offer various types of control over the pulse strings and therefore the motor performance. These include continuous feeding, positioning, and origin return, etc. at a constant speed, linear acceleration/deceleration, and S-curve acceleration/deceleration.
The PCL6046 controls four axes. It can control the linear interpolation of two to four axes, circular interpolations between any two axes, confirm PCL operation status, and output an interrupt with various conditions. It also integrates an interface for servo motor drivers.
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

- CPU-I/F
  The PCL6046 contains the following CPU interface circuits.
  1) 8-bit interface for Z80 CPU.
  2) 16-bit interface for 8086 CPU.
  3) 16-bit interface for H8 CPU.
  4) 16-bit interface for 68000 CPU.

- Direct access to internal registers
  If address buses A0 to A9 are connected, it is available to write into/read from register directly without any commands. If only A0 to A2, A8 and A9 are connected, it is available to use 32-bite occupied area.

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

- Interpolation operation
  Feeding with linear interpolation of any two to four axes and circular interpolation of any two axes are both possible.

- Speed override
  The feed speed can be changed in the middle of any feed operation.
  However, the feed speed cannot be changed during operation when the synthesized speed constant control for linear interpolation is ON while using S-curve deceleration.

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

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

- Pre-register function
  The next two sets of data (feed amount, initial speed, feed speed, acceleration rate, deceleration rate, speed magnification rate, ramping-down point, operation mode, center of circular interpolation, S-curve range on an acceleration, S-curve range on a deceleration, number of steps for circular interpolation) can be written while executing the current data. The next set of data and sets of data after next can be written in advance of their execution for checking by the comparator.
  When the current operation is complete, the system will immediately execute the next operation.

- A variety of counter circuits

The following four counters are available separately for each axis.

Counter	Use or purpose	Counter Input
COUNTER 1	32-bit counter for control of the command position	Output pulses
COUNTER 2	32-bit counter for mechanical position control (Can be used as a general-purpose counter)	EA/EB input Output pulses PA/PB input
COUNTER 3	16-bit counter for controlling the deflection between the command position and the machine's current position	Output pulses and EA/EB input Output pulses and PA/PB input EA/EB input and PA/PB input
COUNTER 4	32-bit counter used to output synchronous signals (Can be used as a general-purpose counter)	Output pulses EA/EB input PA/PB input 1/2 of reference clock

All counters can be reset by writing a command or by providing a CLR signal.
They can also be latched by writing a command, or by providing an LTC or ORG signal.
The PCL6046 can also be set to reset automatically soon after latching these signals.
The COUNTER 1, COUNTER 2, and COUNTER 4 counters have a ring count function that repeats counting through a specified counting range.

- Comparator

There are five comparator circuits for each axis. They can be used to compare target values and internal counter values.
The counter to compare can be selected from COUNTER 1 (command position counter), COUNTER 2 (mechanical position counter), COUNTER 3 (deflection counter), and COUNTER 4 (a general-purpose counter).
Comparators 1 and 2 can also be used as software limits (+SL, -SL).

- Software limit function

You can set software limits using two of the comparator's circuits.
When the mechanical position approaches the software limit range, the LSI will instruct the motors to stop immediately or to stop by deceleration. After that these axes can only be moved in the direction opposite to their previous travel.

- Backlash correction function / Slip correction function

Both the backlash and slip corrections are available. Backlash correction corrects the feed amount each time the feed direction is changed. Slip correction corrects the feed amount regardless of the feed direction. However, the backlash correction cannot be applied while performing a circular interpolation.

- Synchronous signal output function

The LSI can output pulse signals for each specified rate interval.

- Simultaneous start function

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

- Simultaneous stop function

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

- Vibration restriction function

Specify a control constant in advance and add one pulse each for reverse and forward feed just before stopping.
Using this function, vibration can be decreased while stopping.

- Manual pulsar input function

By applying manual pulse signals (PA/PB), you can rotate a motor directly.
The input signals can be 90˚ phase difference signals (1x, 2x, or 4x) or up and down signals.
In addition to the magnification rates above, the PCL6046 contains an integral pulse number magnification circuit which multiplies by 1x to 32x and a pulse quantity division circuit of (1 to 2048)/2048. Software limit settings can be used, and the PCL stops outputting pulses. It can also feed in the opposite direction.

- Direct input of operation switch

  Positive and negative direction terminals (±DR) are provided to drive a motor with an external operation switch.

  These switches turn the motor forward (+) and backward (-).

- Out-of-step detection function

  This LSI has a deflection counter which can be used to compare command pulses and encoder signals (EA/EB).

  It can be used to detect out-of-step operation and to confirm a position by using a comparator.

- Idling pulse output function

  This function outputs a preset number of pulses at the initial speed (FL) before a high-speed start acceleration operation.

  Even if a value near to the maximum starting pulse rate is set during acceleration, this function is effective in preventing out-of-step operation for stepper motors.

- Operation mode

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

  <Examples of the operation modes>

  1) Start/stop by command.
  2) Continuous operation and positioning operation using PA/PB inputs (manual pulsar).
  3) Operate for specified distances or in continuous operation using +DR/-DR signals (drive switch).
  4) Origin return operation.
  5) Positioning operation using commands.
  6) Hardware start of the positioning operation using #CSTA input.
  7) Change the target position after turning ON the PCS. (Delay control)

- Variety of origin return sequences (Homing)

  The following patterns can be used.

  1) Feeds at constant speed and stops when the ORG signal is turned ON
  2) Feeds at constant speed and stops when an EZ signal is received (after the ORG signal is turned ON).
  3) Feeds at constant speed, reverses when the ORG signal is turned ON, and stops when an EZ signal is received.
  4) Feeds at constant speed and stops when the EL signal is turned ON. (Normal stop)
  5) Feeds at constant speed, reverses when the EL signal is turned ON, and stops when an EZ signal is received.
  6) Feeds at high speed, decelerates when the SD signal is turned ON, and stops when the ORG signal is turned ON.
  7) Feeds at high speed, decelerates when the ORG signal is turned ON, and stops when an EZ signal is received.
  8) Feeds at high speed, decelerates and stops after the ORG signal is turned ON. Then, reverses to feed and stops when an EZ signal is received.
  9) Feeds at high speed, decelerates and stops by memorizing the position when the ORG signal is turned ON, and stops at the memorized position.
  10) Feeds at high speed, decelerates to the position stored in memory when an EZ signal is received after the ORG signal is turned ON. Then, returns to the memorized position if an overrun occurs.
  11) Feeds at high speed, reverses after a deceleration stop triggered by the EL signal, and stops when an EZ signal is received.

- Mechanical input signals

The following five signals can be input for each axis.

1) +EL: When this signal is turned ON, while feeding in the positive (+) direction, movement on this axis stops immediately (with deceleration). When this signal is ON, no further movement occurs on the axis in the positive (+) direction. (The motor can be rotated in the negative (-) direction.)
2) -EL: Functions the same as the +EL signal except that it works in the negative (-) direction.
3) +SD: This signal can be used as a deceleration signal or a deceleration stop signal, according to the software setting. When this is used as a deceleration signal, and when this signal is turned ON during a high speed feed operation, the motor on this axis will decelerate to the FL speed. If this signal is ON and movement on the axis is started, the motor on this axis will run at the FL 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 on this axis will decelerate to the FL speed and then stop.
4) -SD: Functions the same as the +SD signal except that it works in the negative (-) direction.
5) 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, -SD and ORG signals can be changed using software.

- Servomotor I/F

The following three signals can be used as an interface for each axis

1) INP: Input positioning complete signal that is output by a servomotor driver.
2) ERC: Output deflection counter clear signal to a servomotor driver.
3) ALM: Regardless of the direction of operation, when this signal is ON, movement on this axis stops immediately (deceleration stop). When this signal is ON, no movement can occur on this axis.

The input/output logic of the INP, ERC, and ALM signals can be changed using software.
The ERC signal is a pulsed output. The pulse length can be set. (12 μsec to 104 msec. A level output is also available.)

- Output pulse specifications

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

- Emergency stop signal (#CEMG) input

When this signal is turned ON, movement on both axes stops immediately. While this signal is ON, no movement is allowed on any axes.

- Interrupt signal output

An #INTsignal (interrupt request) can be output for many reasons.
The #INT terminal output signal can use ORed logic for lots of conditions on each axis.
(When more than one 6045BL LSI is used, wired OR connections are not possible.)

2. Specifications

Item	Description
Number of axes	4 axes (X, Y, Z, and U axis)
Reference clock	Standard: 19.6608 MHz (Max. 30 MHz)
Positioning control range	-2,147,483,648 to +2,147,483,647 (32-bit)
Ramping-down point setting range	0 to 16,777,215 (24-bit)
Number of registers used for setting speeds	Three for each axis (FL, FH, and FA (speed correction))
Speed setting step range	1 to 65,535 (16-bits)
Speed magnification range	Multiply by 0.1 to 100 Multiply by 0.1 = 0.1 to 6,553.5 pps Multiply by 1 = 1 to 65,535 pps Multiply by 100 = 100 to 6,553,500 pps (When the reference clock is 19.6608 MHz) Multiply by 152.5 = 152.5 to 9,999,847 pps (When the reference clock is 30.0MHz)
Acceleration/deceleration characteristics	Selectable acceleration/deceleration pattern for both increasing and decreasing speed separately, using Linear and S-curve acceleration/deceleration.
Acceleration rate setting range	1 to 65,535 (16-bit)
Deceleration rate setting range	1 to 65,535 (16-bit)
Ramping-down point automatic setting	Automatic setting within the range of (deceleration time) < (acceleration time x 2)
Feed speed automatic correction function	Automatically lowers the feed speed for short distance positioning moves.
Manual operation input	Manual pulsar input, pushbutton switch input
Counter	COUNTER 1: Command position counter (32-bit) COUNTER 2: Mechanical position counter (32-bit) COUNTER 3: Deflection counter (16-bit) COUNTER 4: General-purpose counter (32-bit)
Comparators	32-bits x 5 circuits / axis
Interpolation functions	Linear interpolation: Any 2 to 4 axes, Circular interpolation: Any 2 axes
Operating temperature range	-40 to +70^{o}C
Power supply	Single power supply of 3.3 V±10%
Package	208-pin BGA

3. Terminal Assignment Diagram

TOP VIEW

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

4. Functions of Terminals

<table>
<tr><th>Signal name</th><th>Terminal No.</th><th>Input/ output</th><th>Logic</th><th>Description</th></tr>
<tr><td>VSS</td><td>A2, A5,
A6, B5,
B9, B12
B17, C5,
D2, D5,
D7, D13,
D16, E2,
G16, H2,
J16, K1,
M4, N16,
P2, R6,
R12, R15,
U10, U13</td><td>Power source</td><td></td><td>Supply a negative power.
Make sure to connect all of these terminals.</td></tr>
<tr><td>VDD</td><td>A4, A16,
B1, C6,
D10, E14,
G4, J2,
L16, N3,
P13, T8,
T11, T17,
U3</td><td>Power source</td><td></td><td>Supply +3.3 VDC power.
The allowable power supply range is +3.3 VDC ±10%.
Make sure to connect all of these terminals.</td></tr>
<tr><td>#RST</td><td>C2</td><td>Input</td><td>Negative</td><td>Input reset signal.
Make sure to set this signal LOW after turning ON the power and before starting operation. Input at least 8 cycles of the reference clock while holding #RST low.
For details about the chip's status after a reset, see section 11-1, "Reset", in this manual.</td></tr>
<tr><td>CLK</td><td>B6</td><td>Input</td><td></td><td>Input a reference clock signal.
The reference clock frequency is 19.6608 MHz. The LSI creates output pulses based on the clock input on this terminal.</td></tr>
<tr><td>IF0
IF1</td><td>C1
D3</td><td>Input</td><td></td><td>Enter the CPU-I/F mode
<table>
<tr><th rowspan="2">IF1</th><th rowspan="2">IF0</th><th rowspan="2">CPU example</th><th colspan="4">CPU signal connected to the terminal</th></tr>
<tr><th>#RD</th><th>#WR</th><th>A0</th><th>#WRQ</th></tr>
<tr><td>L</td><td>L</td><td>68000</td><td>+3.3V</td><td>R/#W</td><td>#LDS</td><td>#DTACK</td></tr>
<tr><td>L</td><td>H</td><td>H8</td><td>#RD</td><td>#HWR</td><td>(GND)</td><td>#WAIT</td></tr>
<tr><td>H</td><td>L</td><td>8086</td><td>#RD</td><td>#WR</td><td>(GND)</td><td>READY</td></tr>
<tr><td>H</td><td>H</td><td>Z80</td><td>#RD</td><td>#WR</td><td>A0</td><td>#WAIT</td></tr>
</table></td></tr>
<tr><td>#CS</td><td>D1</td><td>Input</td><td>Negative</td><td>When the signal level on this terminal is LOW, the #RD and #WR terminals will be valid.</td></tr>
<tr><td>#RD
#WR</td><td>E4
E3</td><td>Input</td><td>Negative</td><td>Connect to the I/F terminal of the CPU. The #RD and #WR terminals are valid when #CS terminal is LOW.</td></tr>
<tr><td>A0, A1, A2,
A3, A4, A5,
A6, A7, A8,
A9</td><td>E1, F4,
F3, F2,
F1, G3,
G2, G1,
H4, H3</td><td>Input</td><td>Positive</td><td>Address control signals
If only A0 to A2, A8 and A9 are connected, address area can be made small.</td></tr>
<tr><td>#INT</td><td>H1</td><td>Output</td><td>Negative</td><td>Outputs an interrupt request signal (IRQ) to an external CPU.
After this terminal is turned ON, the signal will return to OFF when a REST (error interrupt cause) or RIST (event interrupt cause) signal is received. The output status can be checked with an MSTSW (main status) command signal.
The #INT output signal can be masked.
When more than one 6046 LSI is used, a wired OR connection between #INT terminals is not allowed.</td></tr>
</table>

Signal name	Terminal No.	Input/ output	Logic	Description
#WRQ	J4	Output	Negative	Outputs a wait request signal to cause a CPU to wait. Please make sure to connect it with CPU when direct access to internal register can be used. The LSI needs 4 reference clock cycles to process each command. If the #WRQ signal is not used, make sure that an external CPU does not access this LSI during this interval.
#IFB	J3	Output	Negative	Signal used to indicate that the LSI is processing commands. Use this signal to make connections with a CPU that does not have a wait control input terminal. When the LSI receives a write command from a CPU, this signal will go LOW. When the LSI finishes processing, this signal will go HIGH. The LSI makes sure that this terminal is HIGH and then proceeds to the next step.
D0, D1, D2, D3, D4, D5, D6, D7	J1, K4 K3, K2 L4, L3 L2, L1	Input/ Output	Positive	Bi-directional data bus. When connecting a 16-bit data bus, connect the lower 8 signal lines here.
D8, D9, D10, D11, D12, D13, D14, D15	M3, M2 M1, N4 N2, N1 P4, P3	Input/ Output	Positive	Bi-directional data bus. When connecting a 16-bit data bus, connect the upper 8 signal lines here. When a Z80-I/F (IF1 = H, IF0 = H) is used, provide a pull up resistor (5k to 10 K-ohms) on VDD. (One resistor can be used for all 8 lines.)
#CSTA	D4	Input/ Output *	Negative	Input/Output terminal for simultaneous start. When more than one LSI is used and you want to start them simultaneously, connect this terminal on each LSI. The terminal status can be checked using an RSTS command signal (extension status).
#CSTP	C4	Input/ Output *	Negative	Input/Output terminal for a simultaneous stop. When more than one LSI is used and you want to stop them simultaneously, connect this terminal on each LSI. The terminal status can be checked using an RSTS command signal (extension status).
#CEMG	B4	Input U	Negative	Input for an emergency stop. While this signal is LOW, motion cannot start. If this signal changes to LOW while in operation, all the motors will stop operation immediately.
ELLx ELLy ELLz ELLu	C3 B3 A3 B2	Input U		Specify the input logic for the ±EL signal. LOW: The input logic on ±EL is positive. HIGH: The input logic on ±EL is negative.
+ ELx + ELy + ELz + ELu	P16 M16 K16 H16	Input U	Negative %	Input end limit signal in the positive (+) direction. When this signal is ON while feeding in the positive (+) direction, motion of an axis will stop immediately or will decelerate and stop. Specify the input logic using the ELL terminal. The terminal status can be checked using an SSTSW command signal (sub status).
- ELx - ELy - ELz - ELu	P17 M17 K17 H17	Input U	Negative %	Input end limit signal in the negative (-) direction. When this signal is ON while feeding in negative (-) direction, motion of an axis will stop immediately, or will decelerate and stop. Specify the input logic using the ELL terminal. The terminal status can be checked using an SSTSW command signal (sub status).
+ SDx + SDy + SDz + SDu	R17 M14 K14 H14	Input U	Negative#	Input + direction 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 an SSTSW command signal (sub status).

Signal name	Terminal No.	Input/ output	Logic	Description
- SDx - SDy - SDz - SDu	P15 M15 K15 H15	Input U	Negative#	Input - direction 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 an SSTSW command signal (sub status).
ORGx ORGy ORGz ORGu	R16 N17 L17 J17	Input U	Negative #	Input origin position signal. Used for origin position operations. (Edge detection.) The input logic can be selected using software. The terminal status can be checked using an SSTSW command signal (sub status).
ALMx ALMy ALMz ALMu	N14 L14 J14 G14	Input U	Negative #	Input alarm signal. When this signal is ON, motion of an axis stops immediately, or will decelerate and stop. The input logic can be selected using software. The terminal status can be checked using an SSTSW command signal (sub status).
OUTx OUTy OUTz OUTu	P11 U11 T12 R13	Output	Negative #	Output command pulses for controlling a motor. When Common Pulse mode is selected: Output pulses and the feed direction is determined by DIR signals. When Two-pulse output mode is selected: Outputs pulses in the positive (+) direction. When 90˚ phase difference mode is selected: Outputs DIR signals and 90˚phase difference signals. The output logic can be changed using software.
DIRx DIRy DIRz DIRu	R11 P12 U12 T13	Output	Negative #	Output command pulses for controlling a motor, or outputs direction signal. When Common Pulse mode is selected: Outputs a direction signal. When Two-pulse output mode is selected: Output pulses in the negative (-) direction. When 90˚ phase difference mode is selected: Outputs DIR signals and 90˚phase difference signals. The output logic can be changed using software
EAx, EBx EAy, EBy EAz, EBz EAu, EBu	G17, F14, D17, C15, B15, A15, C10, B10	Input U		Input this signal when you want to control the mechanical position using the encoder signal. Input a 90˚ phase difference signal (1x, 2x, 4x) or input positive (+) pulses on EA and negative (-) pulses on EB. When inputting 90˚ phase difference signals, if the EA signal phase is ahead of the EB signal, the LSI will count up (count forward) pulses. The counting direction can be changed using software.
EZx EZy EZz EZu	F15 C16 C14 A10	Input U	Negative #	Input a marker signal (this signal is output once for each turn of the encoder) when using the marker signal in origin return mode. Use of the EZ signal improves origin return precision. The input logic can be changed using software. The terminal status can be checked using an RSTS command signal (extension status).
PAx, PBx PAy, PBy PAz, PBz PAu, PBu	F16, F17, C17, B16, B14, A14, D9, C9	Input U		Input for receiving external drive pulses, such as manual pulsar. You can input 90˚ phase difference signals (1x, 2x, 4x) or positive (+) pulses (on PA) and negative (-) pulses (on PB). When 90˚ phase difference signals are used, if the signal phase of PA is ahead of the PB signal, the LSI will count up (count forward) pulses. The counting direction can be changed using software.
#PEx #PEy #PEz #PEu	C7 B7 A7 D6	Input U	Negative	Setting these terminals LOW enables PA/PB and +DR/-DR input. By inputting an axis change switch signal, one manual pulsar can be used alternately for four axes.
+DRx,-DRx +DRy,-DRy +DRz,-DRz	E15, E16, C13, B13, A12, D11	Input U	Negative #	You can start operation of the PCL with these signals manually using external switches. Specifying the feed length, constant speed continuous feed, and

Signal name	Terminal No.	Input/ output	Logic	Description
+DRu,-DRu	A9, D8			high-speed continuous feed are possible. The input logic can be changed using software. The terminal status can be checked using an RSTS command signal (extension status).
PCSx PCSy PCSz PCSu	E17 A13 C11 C8	Input U	Negative #	The PCL starts its positioning operation according to this input signal. (Override 2 of the target position.) The input logic can be changed using software. The terminal status can be checked using an RSTS command signal (extension status).
INPx INPy INPz INPu	N15 L15 J15 G15	Input U	Negative #	Input the position complete signal from servo driver (in-position signal). Input logic can be changed using software. The terminal status can be checked using an RSTS command signal (extension status).
CLRx CLRy CLRz CLRu	D14 D12 B11 B8	Input U	Negative #	Reset a specified counter (more than one is available) from COUNTER1 to 4. The input logic can be changed using software. The terminal status can be checked using an RSTS command signal (extension status).
LTCx LTCy LTCz LTCu	D15 C12 A11 A8	Input U	Negative #	Latch counter value of specified counters (more than one is available) from COUNTER1 to 4. The input logic can be changed using software. The terminal status can be checked using an RSTS command signal.
ERCx ERCy ERCz ERCu	P14 R14 T14 U14	Output	Negative #	Outputs a deflection counter clear signal to a servo driver as a pulse. The output logic and pulse width can be changed using software. A LEVEL signal output is also available. The terminal status can be checked using an RSTS command signal.
#BSYx #BSYy #BSYz #BSYu	T15 U15 T16 U16	Output	Negative	Outputs a LOW signal while feeding.
P0x/FUPx P0y/FUPy P0z/FUPz P0u/FUPu	P1 R4 T6 U8	Input/ Output *	Positive	Common terminal for general purpose I/O and FUP. (See Note 5.) As an FUP terminal, it outputs a LOW signal while accelerating. As a general purpose I/O terminal, three possibilities can be specified: input terminal, output terminal, and one shot pulse output terminal. The usage, output logic of the FUP and one shot parameters can be changed using software.
P1x/FDWx P1y/FDWy P1z/FDWz P1u/FDWu	R3 T4 U6 P9	Input/ Output *	Positive	Common terminal for general purpose I/O and FDW. (See Note 5.) As an FDW terminal, it outputs a LOW signal while decelerating. As a general purpose I/O terminal, three possibilities can be specified: input terminal, output terminal, and one shot pulse output terminal. The usage, output logic of the FDW and one shot pulse parameters can be changed using software.
P2x/MVCx P2y/MVCy P2z/MVCz P2u/MVCu	R2 U4 P7 R9	Input/ Output *	Positive	Common terminal for general purpose I/O and MVC. (See Note 5.) When used as an MVC terminal, it outputs a signal while performing a constant speed feed. The usage and output logic of the MVC can be changed using software.
P3x/CP1x (+SLx) P3y/CP1y (+SLy) P3z/CP1z (+SLz) P3u/CP1u (+SLu)	R1 P5 R7 T9	Input/ Output *	Positive	Common terminal for general purpose I/O and CP1 (+SL). (See Note 5.) When used as a CP1 (+SL) terminal, it outputs a signal while satisfying the conditions (within +SL) of comparator 1. The output logic of CP1 (+SL) as well as the selection of input or output functions can be changed using software.

Signal name	Terminal No.	Input/ output	Logic	Description
P4x/CP2x (-SLx) P4y/CP2y (-SLy) P4z/CP2z (-SLz) P4u/CP2u (-SLu)	T2 R5 T7 U9	Input/ Output *	Positive	Common terminal for general purpose I/O and CP2 (-SL). When used as a CP2 (-SL) terminal, it outputs a signal while satisfying the conditions (within -SL) of comparator 2. The output logic of CP2 (-SL) as well as the selection of input or output functions can be changed using software. (See Note 5.)
P5x/CP3x P5y/CP3y P5z/CP3z P5u/CP3u	T1 T5 U7 P10	Input/ Output *	Positive	Common terminal for general purpose I/O and CP3. (See Note 5.) When used as a CP3 terminal, it outputs a signal while satisfying the conditions of comparator 3. The output logic of CP3 as well as the selection of input or output functions can be changed using software.
P6x/CP4x P6y/CP4y P6z/CP4z P6u/CP4u	U2 U5 P8 R10	Input/ Output *	Positive	Common terminal for general purpose I/O and CP4. (See Note 5.) When used as a CP4 terminal, it outputs a signal while satisfying the conditions of comparator 4. The output logic of CP4 as well as the selection of input or output functions can be changed using software.
P7x/CP5x P7y/CP5y P7z/CP5z P7u/CP5u	T3 P6 R8 T10	Input/ Output *	Positive	Common terminal for general purpose I/O and CP5. (See Note 5.) When used as a CP5 terminal, it outputs a signal while establishing the conditions of comparator 5. The output logic of CP5 as well as the selection of input or output functions can be changed using software.

Note 1: "Input U" refers to an input with a pull up resistor. The internal pull up resistance (40 K to 240 K-ohms) is only used to keep a terminal from floating. If you want to use the LSI with an open collector system, an external pull up resistor (5k to 10 K-ohms) is required.
As a noise prevention measure, pull up unused terminals to VDD using an external resistor (5 k to 10 K-ohms), or connect them directly to VDD.

Note 2: "Input/Output *" refers to a terminal with a pull up resistor. The internal pull up resistor (40 K to 240 K-ohms) is only used to keep a terminal from floating. If it is connected in a wired OR circuit, an external pull up resistor (5 k to 10 K-ohms) is required.
As a noise prevention measure, pull up unused terminals to VDD using an external resistor (5 k to 10 K-ohms).

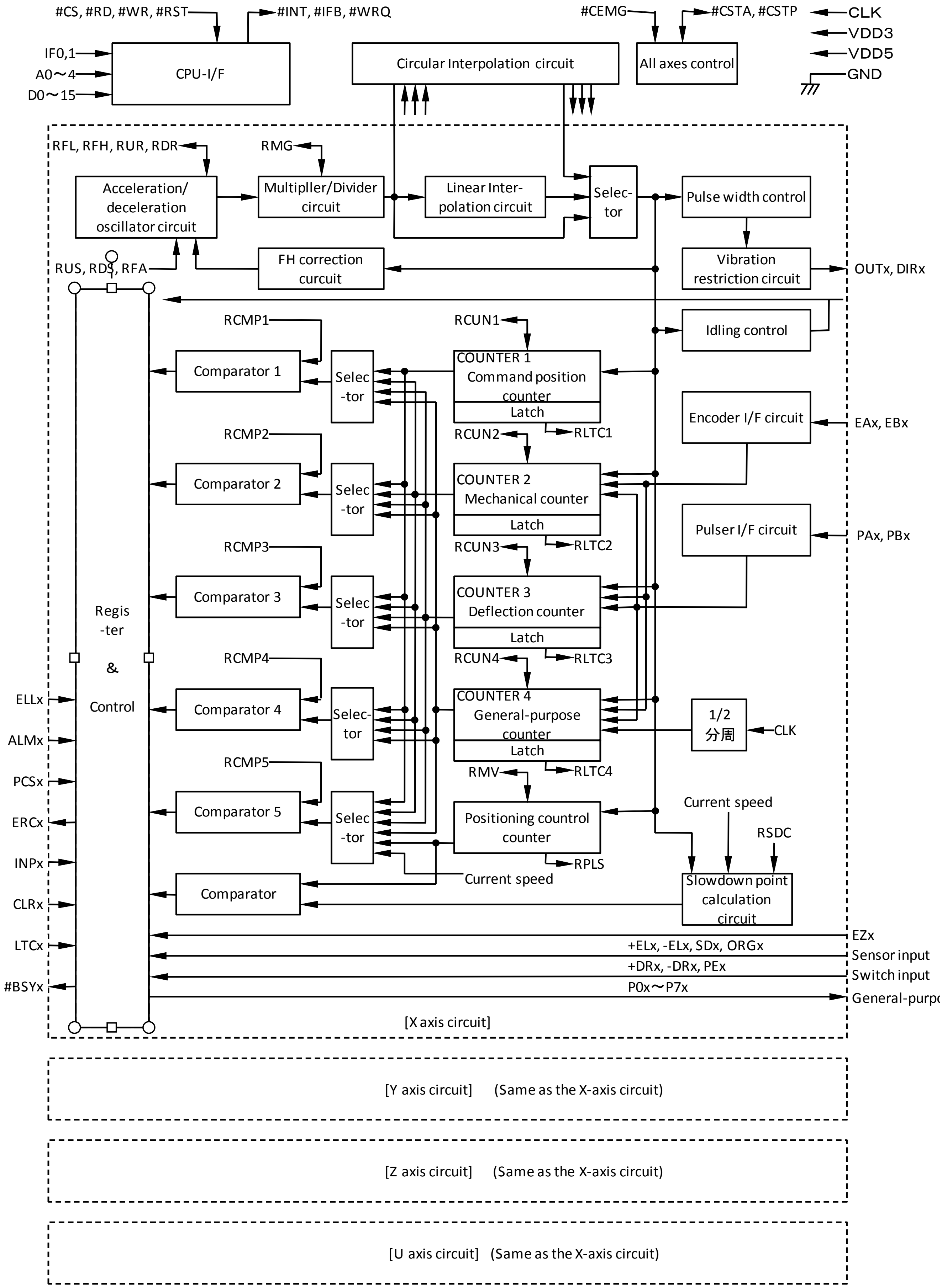
Note 3: If an output terminal is not being used, leave it open.

Note 4: "Positive" refers to positive logic. "Negative" refers to negative logic. "#" means that the logic can be changed using software. "%" means that the logic can be changed by the setting on another terminal. The logic shown refers only to the initial status of the terminal. The DIR terminal is initially in a Two-pulse mode.

Note 5: Use the RENV2 register to select an output signal.
When P0 to P7 are set up as output terminals, they can be controlled simultaneously as 8 bits or one bit at a time using output bit control commands, depending on what is written to the output port (OTPB).
When P0 and P1 are set up as one shot pulse output terminals, they will output a one shot signal (T = Approx. 26 msec) using the output bit control command.

Note 6: ORG input is synchronized with output pulses, sampled and controlled by a change of sampling result. Therefore, keep ORG sensor ON for longer than feed amount for one pulse.

5. Block Diagram



6. CPU Interface

6-1. Setting up connections to a CPU

This LSI can be connected to four types of CPUs by changing the hardware settings.
Use the IF0 and IF1 terminals to change the settings and connect the CPU signal lines as follows.

Setting status		CPU type	CPU signal to connect to the 6045BL terminals			
IF1	IF0		#RD terminal	#WR terminal	A0 terminal	#WRQ terminal
L	L	68000	+3.3V	R/#W	#LDS	#DTACK
L	H	H8	#RD	#HWR	(GND)	#WAIT
H	L	8086	#RD	#WR	(GND)	READY
H	H	Z80	#RD	#WR	A0	#WAIT

There are two access schemes of address signals as follows. (As for A0, please refer to the above.)

1. Full-address scheme
   All address terminals of A0 to A9 are connected to CPU address bus (A1 to A9) and 1024-byte address area is occupied.
   It is available to read from or write to internal register directly without using commands. However, to access to each register, make sure to access to 4-byte from the lower address. Additionally, make sure that CPU wait by #WRQ output signal. It is available to access indirectly using I/O buffer.

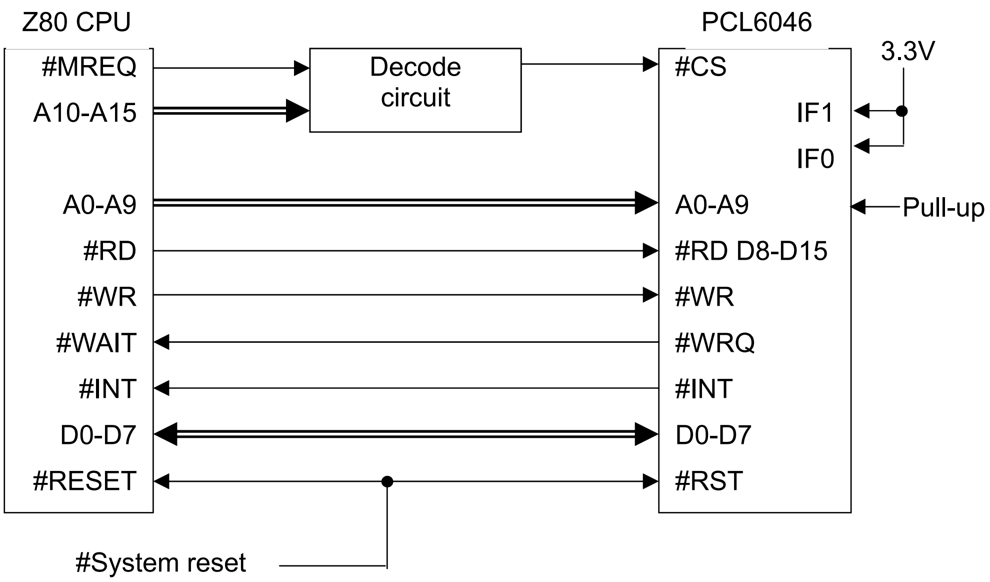
2. Reduced address scheme
   If only A1, A2, A8 and A9 address terminal are connected to CPU address buses (A1 to A4), 32-byte address area is occupied.
   To write into internal register, write "Register write command" after writing data into I/O buffer (4 byte).
   To read from internal register, read from I/O buffer after writing "Register readout commands".

6-2. Precautions for designing hardware

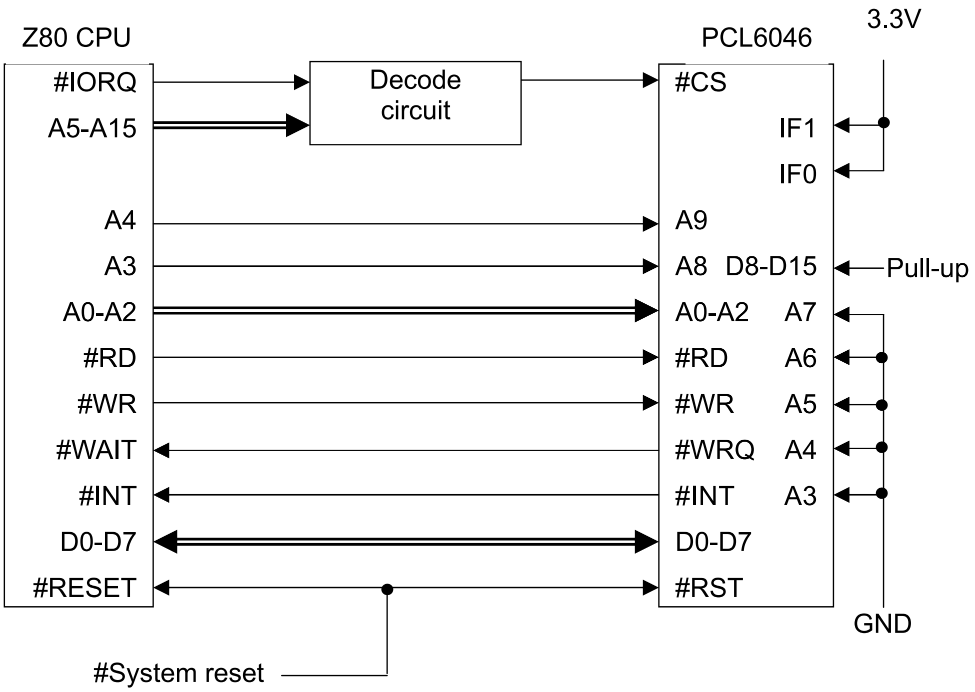
- To reset the LSI, hold the #RST signal LOW, and input the CLK signal for at least 8-clock cycles.
- Connect unused P0 to P7 terminals to VDD through a pull up resistor (5k to 10k ohms).
- When connecting a CPU with an 8-bit bus, pull up terminals D8 to D15 to VDD using an external resistor (5k to 10k ohms). (Shared use of one resister for the 8 lines is available.)
- Use the ELL terminal to change the ±EL signal input logic.
- When access to internal register directly on full-address circuit, make sure that # WRQ output signal make CPU in a wait status.
  In the case to use a CPU that cannot output #WRG signal, make #RD signal width more than 4 cycles of CLK signal, and confirm that #IFB signal is high level before access.

6-3. CPU interface circuit block diagram

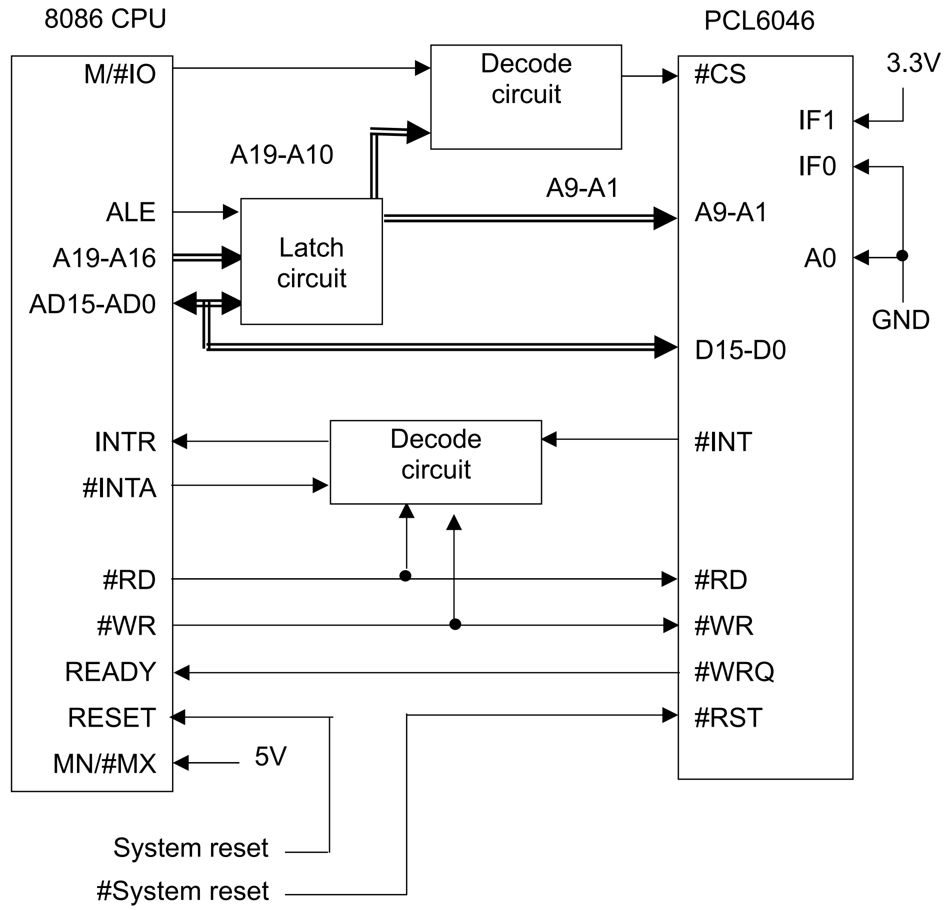
1) Z80 interface (memory map, full-address)



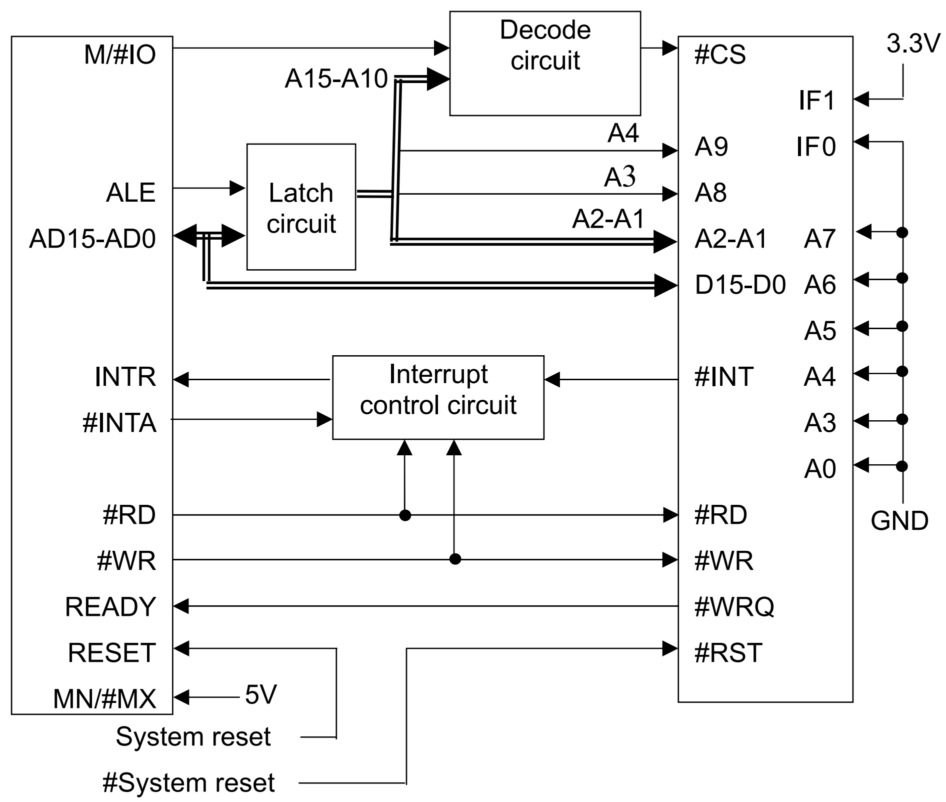
2) Z80 interface (I/O map, reduced address)



3) 8086 interface (Memory map, full address)

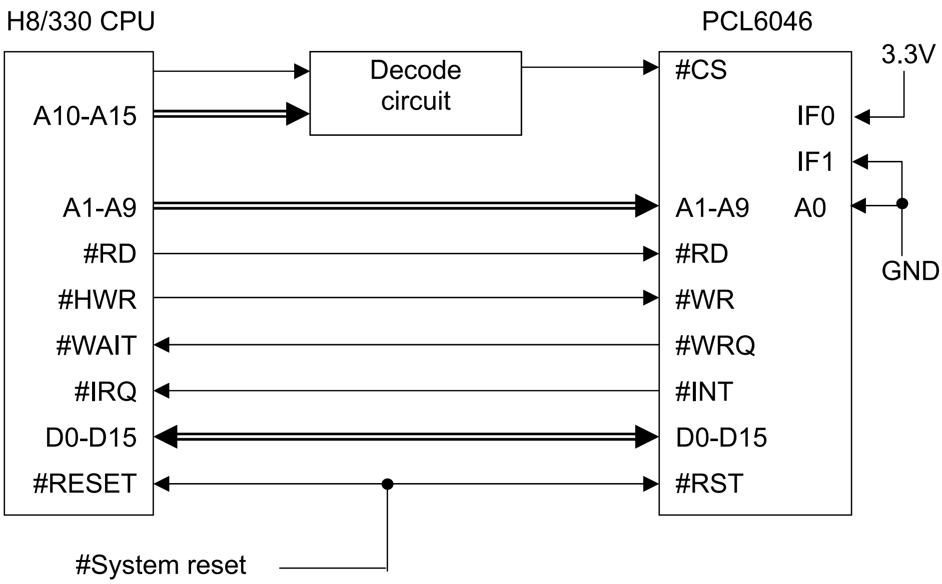


4) 8086 interface (I/O map, reduced address)

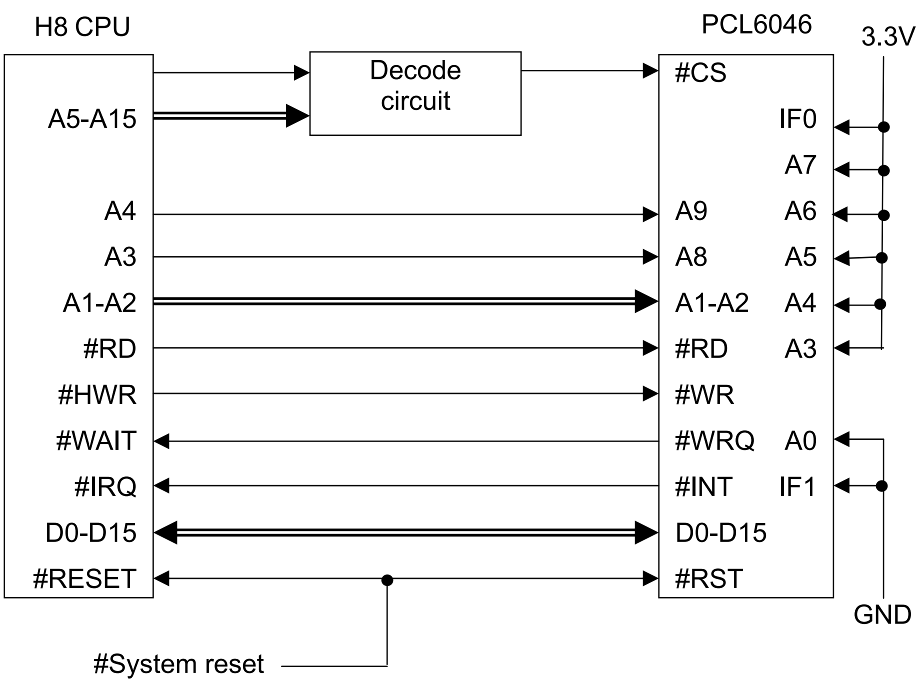


Notes With 8086 interface, only word (16 bit) access is available. Byte (8 bit) access cannot be used.

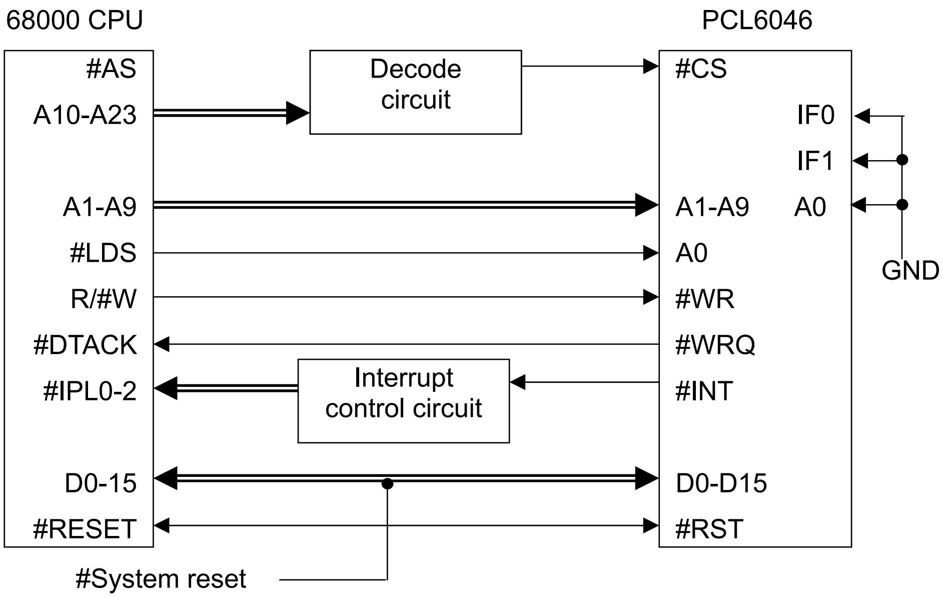
3) H8 interface (full address)



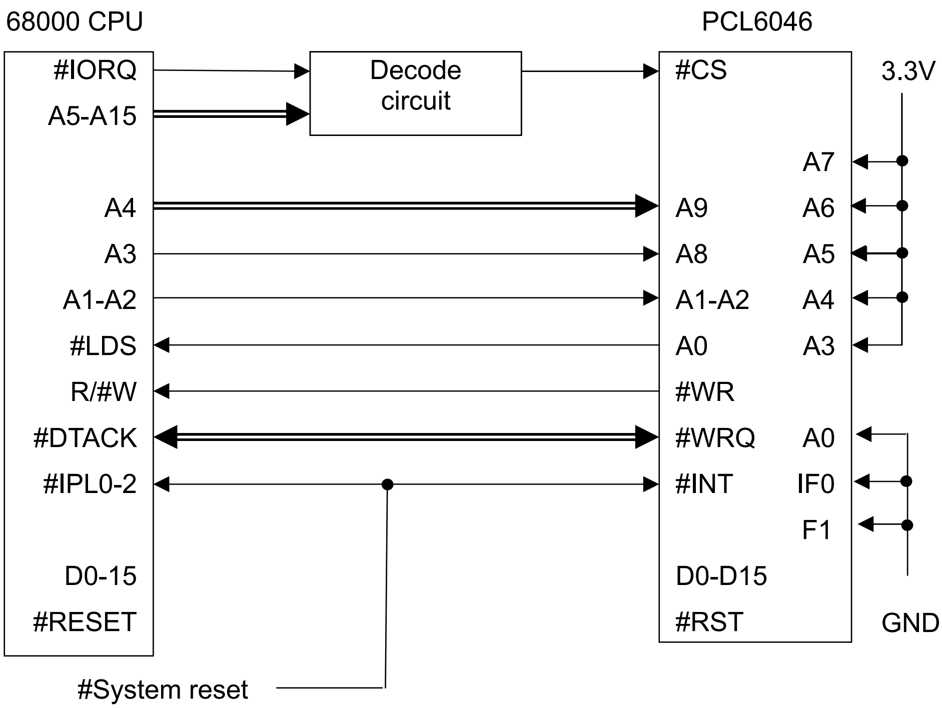
3) H8 interface (reduced address)



7) 68000 interface (full address)



7) 68000 interface (reduced address)



Note: For the 8086, H8, and 68000 interfaces, only word (16-bit) access is available. Byte (8-bit) access is not available.

6-4. Address map

6-4-1. Axis arrangement map

In this LSI, the control address range for each axis is independent. It is selected by using address input terminal A8 and A9, as shown below.

A9	A8	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

6-4-2. Internal map of each axis

The following shows address signal and processing of write / readout cycle.
Please refer to "6-5. Description of the map details" and "8-3. Description of the registers" in detail.
There are two connection schemes of address input terminals: full-address scheme and reduced address scheme. These access schemes to internal register are different.
In a full address scheme, direct access to internal register and access through I/O buffer can be available.
In a reduced address scheme, only indirect access is available.
The internal map of each axis is defined by address input A0 to 7.

> Notes: When you access registers by direct access scheme, make sure to access from lower address to upper address in order by register unit (4 bites).
> Access for Z80 and 8086 should be from lower data to upper data. Access for H8 and 68000 I/F should be from upper data to lower data.

<When used with the Z80 I/F (Direct access)>

<table>
<tr><th>A7～A0
(Hex)</th><th>Read / Write</th><th>Address signal</th><th>Processing detail</th></tr>
<tr><td rowspan="2">00</td><td>Read</td><td>MSTSB0</td><td>Read out main status (bit 0 to 7)</td></tr>
<tr><td>Write</td><td>COMB0</td><td>Write control command</td></tr>
<tr><td rowspan="2">01</td><td>Read</td><td>MSTSB1</td><td>Read out main status (bit 8 to 15)</td></tr>
<tr><td>Write</td><td>COMB1</td><td>Specify axes (Specify axes to execute control command)</td></tr>
<tr><td rowspan="2">02</td><td>Read</td><td>IOPB</td><td>Read out general-purpose I/O ports</td></tr>
<tr><td>Write</td><td>OTPB</td><td>Change status of a general-purpose output port (only bits specified to output are available.</td></tr>
<tr><td>03</td><td>Read</td><td>SSTSB</td><td>Read out sub status</td></tr>
<tr><td>04 to 07</td><td>Read / Write</td><td>BUFB3～</td><td>Read from /Write into I/O buffer (bit 0 to 31)</td></tr>
<tr><td>08 to 0B</td><td>Read / Write</td><td>PRMV</td><td>Read from /Write into PRMV register (bit 0 to 31)<table><tr><td>Address</td><td>0B</td><td>0A</td><td>09</td><td>08</td></tr><tr><td>Bit number</td><td>24 to 31</td><td>16 to 23</td><td>8 to 15</td><td>0 to 7</td></tr></table></td></tr>
<tr><td>0C to 0F</td><td>Read / Write</td><td>PRFL</td><td>Read from /write into PRFL register (bit 0 to 31)</td></tr>
<tr><td>10 to 13</td><td>Read / Write</td><td>PRFH</td><td>Read from /write into PRFH register (bit 0 to 31)</td></tr>
<tr><td>14 to 17</td><td>Read / Write</td><td>PRUR</td><td>Read from /write into PRUR register (bit 0 to 31)</td></tr>
<tr><td>18 to 1B</td><td>Read / Write</td><td>PRDR</td><td>Read from /write into PRDR register (bit 0 to 31)</td></tr>
<tr><td>1C to 1F</td><td>Read / Write</td><td>PRMG</td><td>Read from /write into PRMG register (bit 0 to 31)</td></tr>
<tr><td>20 to 23</td><td>Read / Write</td><td>PRDP</td><td>Read from /write into PRDP register (bit 0 to 31)</td></tr>
<tr><td>24 to 27</td><td>Read / Write</td><td>PRMD</td><td>Read from /write into PRMD register (bit 0 to 31)</td></tr>
<tr><td>28 to 2B</td><td>Read / Write</td><td>PRIP</td><td>Read from /write into PRIP register (bit 0 to 31)</td></tr>
<tr><td>2C to 2F</td><td>Read / Write</td><td>PRUS</td><td>Read from /write into PRUS register (bit 0 to 31)</td></tr>
<tr><td>30 to 33</td><td>Read / Write</td><td>PRDS</td><td>Read from /write into PRDS register (bit 0 to 31)</td></tr>
</table>

A7～A0 (Hex)	Read / Write	Address signal	Processing detail
34 to 37	Read / Write	PRCP5	Read from /write into PRCP5 register (bit 0 to 31)
38 to 3B	Read / Write	PRCI	Read from /write into PRCI register (bit 0 to 31)
3C to 3F	Read / Write		Invalid (00H is output when reading)
40 to 43	Read / Write	RMV	Read from /write into RMV register (bit 0 to 31)
44 to 47	Read / Write	RFL	Read from /write into RFL register (bit 0 to 31)
48 to 4B	Read / Write	RFH	Read from /write into RFH register (bit 0 to 31)
4C to 4F	Read / Write	RUR	Read from /write into RUR register (bit 0 to 31)
50 to 53	Read / Write	RDR	Read from /write into RDR register (bit 0 to 31)
54 to 57	Read / Write	RMG	Read from /write into RMG register (bit 0 to 31)
58 to 5B	Read / Write	RDP	Read from /write into RDP register (bit 0 to 31)
5C to 5F	Read / Write	RMD	Read from /write into RMD register (bit 0 to 31)
60 to 63	Read / Write	RIP	Read from /write into RIP register (bit 0 to 31)
64 to 67	Read / Write	RUS	Read from /write into RUS register (bit 0 to 31)
68 to 6B	Read / Write	RDS	Read from /write into RDS register (bit 0 to 31)
6C to 6F	Read / Write	RFA	Read from /write into RFA register (bit 0 to 31)
70 to 73	Read / Write	RENV1	Read from /write into RENV1 register (bit 0 to 31)
74 to 77	Read / Write	RENV2	Read from /write into RENV2 register (bit 0 to 31)
78 to 7B	Read / Write	RENV3	Read from /write into RENV3 register (bit 0 to 31)
7C to 7F	Read / Write	RENV4	Read from /write into RENV4 register (bit 0 to 31)
80 to 83	Read / Write	RENV5	Read from /write into RENV5 register (bit 0 to 31)
84 to 87	Read / Write	RENV6	Read from /write into RENV6 register (bit 0 to 31)
88 to8B	Read / Write	RENV7	Read from /write into RENV7 register (bit 0 to 31)
8C to 8F	Read / Write	RCUN1	Read from /write into RCUN1 register (bit 0 to 31)
90 to 93	Read / Write	RCUN2	Read from /write into RCUN2 register (bit 0 to 31)

<When used with the 8086 I/F (Indirect access)>

A2～A1	Read / Write	Address	Processing detail
0 0	Read	MSTSW	Read out main status (bit 0 to 15)
	Write	COMW	Write axis assignment and control command
0 1	Read	SSTSW	Read out sub status and a general-purpose I/O port
	Write	OTPW	Change status of a general-purpose output port (only bits specified to output are available.
1 0	Read / Write	BUFW0	Read from /Write into I/O buffer (bit 0 to 15)
1 1	Read / Write	BUFW1	Read from /Write into I/O buffer (bit 16 to 31)

<When used with the H8 and 8086 I/F (Direct access)>

<table>
<tr><th>A7～A0
(Hex)</th><th>Read / Write</th><th>Address signal</th><th>Processing detail</th></tr>
<tr><td rowspan="2">FE</td><td>Read</td><td>MSTSW</td><td>Read out main status (bit 0 to 15)</td></tr>
<tr><td>Write</td><td>COMW</td><td>Write axis command and control command</td></tr>
<tr><td rowspan="2">FC</td><td>Read</td><td>SSTSW</td><td>Read out sub status and general-purpose I/O ports</td></tr>
<tr><td>Write</td><td>OTPW</td><td>Change status of a general-purpose output port (only bits specified to output are available.</td></tr>
<tr><td>F8,FA</td><td>Read / Write</td><td>BUFW0 to BUFW1</td><td>Read from /Write into I/O buffer (bit 0 to 31)</td></tr>
<tr><td>F4,F6</td><td>Read / Write</td><td>PRMV</td><td>Read from /Write into PRMV register (bit 0 to 31)<table><tr><td>Address</td><td>F4</td><td>F6</td></tr><tr><td>Bit number</td><td>16 to 31</td><td>0 to 15</td></tr></table></td></tr>
<tr><td>F0,F2</td><td>Read / Write</td><td>PRFL</td><td>Read from /write into PRFL register (bit 0 to 31)</td></tr>
<tr><td>EC,EE</td><td>Read / Write</td><td>PRFH</td><td>Read from /write into PRFH register (bit 0 to 31)</td></tr>
<tr><td>E8,EA</td><td>Read / Write</td><td>PRUR</td><td>Read from /write into PRUR register (bit 0 to 31)</td></tr>
<tr><td>E4,E6</td><td>Read / Write</td><td>PRDR</td><td>Read from /write into PRDR register (bit 0 to 31)</td></tr>
<tr><td>E0,E2</td><td>Read / Write</td><td>PRMG</td><td>Read from /write into PRMG register (bit 0 to 31)</td></tr>
<tr><td>DC,DE</td><td>Read / Write</td><td>PRDP</td><td>Read from /write into PRDP register (bit 0 to 31)</td></tr>
<tr><td>D8,DA</td><td>Read / Write</td><td>PRMD</td><td>Read from /write into PRMD register (bit 0 to 31)</td></tr>
<tr><td>D4,D6</td><td>Read / Write</td><td>PRIP</td><td>Read from /write into PRIP register (bit 0 to 31)</td></tr>
<tr><td>D0,D2</td><td>Read / Write</td><td>PRUS</td><td>Read from /write into PRUS register (bit 0 to 31)</td></tr>
<tr><td>CC,CE</td><td>Read / Write</td><td>PRDS</td><td>Read from /write into PRDS register (bit 0 to 31)</td></tr>
<tr><td>C8,CA</td><td>Read / Write</td><td>PRCP5</td><td>Read from /write into PRCP5 register (bit 0 to 31)</td></tr>
<tr><td>C4,C6</td><td>Read / Write</td><td>PRCI</td><td>Read from /write into PRCI register (bit 0 to 31)</td></tr>
<tr><td>C0,C2</td><td>Read / Write</td><td></td><td>Invalid (00H is output when reading)</td></tr>
<tr><td>BC,BE</td><td>Read / Write</td><td>RMV</td><td>Read from /write into RMV register (bit 0 to 31)</td></tr>
<tr><td>B8,BA</td><td>Read / Write</td><td>RFL</td><td>Read from /write into RFL register (bit 0 to 31)</td></tr>
<tr><td>B4,B6</td><td>Read / Write</td><td>RFH</td><td>Read from /write into RFH register (bit 0 to 31)</td></tr>
<tr><td>B0,B2</td><td>Read / Write</td><td>RUR</td><td>Read from /write into RUR register (bit 0 to 31)</td></tr>
<tr><td>AC,AE</td><td>Read / Write</td><td>RDR</td><td>Read from /write into RDR register (bit 0 to 31)</td></tr>
<tr><td>A8,AA</td><td>Read / Write</td><td>RMG</td><td>Read from /write into RMG register (bit 0 to 31)</td></tr>
<tr><td>A4,A6</td><td>Read / Write</td><td>RDP</td><td>Read from /write into RDP register (bit 0 to 31)</td></tr>
<tr><td>A0,A2</td><td>Read / Write</td><td>RMD</td><td>Read from /write into RMD register (bit 0 to 31)</td></tr>
<tr><td>9C,9E</td><td>Read / Write</td><td>RIP</td><td>Read from /write into RIP register (bit 0 to 31)</td></tr>
<tr><td>98,9A</td><td>Read / Write</td><td>RUS</td><td>Read from /write into RUS register (bit 0 to 31)</td></tr>
<tr><td>94,96</td><td>Read / Write</td><td>RDS</td><td>Read from /write into RDS register (bit 0 to 31)</td></tr>
<tr><td>90,92</td><td>Read / Write</td><td>RFA</td><td>Read from /write into RFA register (bit 0 to 31)</td></tr>
<tr><td>8C,8E</td><td>Read / Write</td><td>RENV1</td><td>Read from /write into RENV1 register (bit 0 to 31)</td></tr>
<tr><td>88,8A</td><td>Read / Write</td><td>RENV2</td><td>Read from /write into RENV2 register (bit 0 to 31)</td></tr>
<tr><td>84,86</td><td>Read / Write</td><td>RENV3</td><td>Read from /write into RENV3 register (bit 0 to 31)</td></tr>
<tr><td>80,82</td><td>Read / Write</td><td>RENV4</td><td>Read from /write into RENV4 register (bit 0 to 31)</td></tr>
<tr><td>7C,7E</td><td>Read / Write</td><td>RENV5</td><td>Read from /write into RENV5 register (bit 0 to 31)</td></tr>
<tr><td>78,7A</td><td>Read / Write</td><td>RENV6</td><td>Read from /write into RENV6 register (bit 0 to 31)</td></tr>
<tr><td>74,76</td><td>Read / Write</td><td>RENV7</td><td>Read from /write into RENV7 register (bit 0 to 31)</td></tr>
<tr><td>70,72</td><td>Read / Write</td><td>RCUN1</td><td>Read from /write into RCUN1 register (bit 0 to 31)</td></tr>
<tr><td>6C,6E</td><td>Read / Write</td><td>RCUN2</td><td>Read from /write into RCUN2 register (bit 0 to 31)</td></tr>
<tr><td>68,6A</td><td>Read / Write</td><td>RCUN3</td><td>Read from /write into RCUN3 register (bit 0 to 31)</td></tr>
<tr><td>64,66</td><td>Read / Write</td><td>RCUN4</td><td>Read from /write into RCUN4 register (bit 0 to 31)</td></tr>
</table>

A7～A0 (Hex)	Read / Write	Address signal	Processing detail
60,62	Read / Write	RCMP1	Read from /write into RCMP1 register (bit 0 to 31)
5C,5E	Read / Write	RCMP2	Read from /write into RCMP2 register (bit 0 to 31)
58,5A	Read / Write	RCMP3	Read from /write into RCMP3 register (bit 0 to 31)
54,56	Read / Write	RCMP4	Read from /write into RCMP4 register (bit 0 to 31)
50,52	Read / Write	RCMP5	Read from /write into RCMP5 register (bit 0 to 31)
4C,4E	Read / Write	RIRQ	Read from /write into RIRQ register (bit 0 to 31)
48,4A	Read	RLTC1	Read from /write into RLTC1 register (bit 0 to 31)
44,46	Read	RLTC2	Read from /write into RLTC2 register (bit 0 to 31)
40,42	Read	RLTC3	Read from /write into RLTC3 register (bit 0 to 31)
3C,3E	Read	RLTC4	Read from /write into RLTC4 register (bit 0 to 31)
38,3A	Read	RSTS	Read from /write into RSTS register (bit 0 to 31)
34,36	Read / Write	REST	Read from /write into REST register (bit 0 to 31)
30,32	Read / Write	RIST	Read from /write into RIST register (bit 0 to 31)
2C,2E	Read	RPLS	Read from /write into RPLS register (bit 0 to 31)
28,2A	Read	RSPD	Read from /write into RSPD register (bit 0 to 31)
24,26	Read	RSDC	Read from /write into RSDC register (bit 0 to 31)
10 to22	Read / Write		Invalid (00H is output when reading)
0C,0E	Read / Write	RCI	Read from /write into RCI register (bit 0 to 31)
08,0A	Read	RCIC	Read from /write into RCIC register (bit 0 to 31)
04,06	Read / Write		Invalid (00H is output when reading)
00,02	Read	RIPS	Read from /write into RCIPS register (bit 0 to 31)

<When used with the H8 and 8086 I/F (Indirect access)>

A2～A1	Read / Write	Address	Processing detail
11	Read	MSTSW	Read out main status (bit 0 to 15)
	Write	COMW	Write axis assignment and control command
10	Read	SSTSW	Read out sub status and a general-purpose I/O port
	Write	OTPW	Change status of a general-purpose output port (only bits specified to output are available.
01	Read / Write	BUFW0	Read from /Write into I/O buffer (bit 0 to 15)
00	Read / Write	BUFW1	Read from /Write into I/O buffer (bit 16 to 31)

6-5. Description of the map details

6-5-1. Write a command code and axis selection (COMW, COMB)

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

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

SELx to u: Select an axis for executing the command. If all of the bits are 0, only the own axis (selected by A8, A9) is selected. To write the same command to more than one axis, set the bits of the selected axes to 1. When you write to a register, the details of the input/output buffer are written into the register for each axis. When you read from a register, the details in the register are written into the input/output buffer for each axis.

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

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

Specify output terminal status from the general purpose I/O terminals P0 to P7.
Bits corresponding to terminals not set as outputs are ignored.
When writing a word, the upper 8 bits are ignored. However, they should be set to 0 for future compatibility.

OTP0 to 7: Specify the status of output terminals P0n to P7n (n = x, y, z, u).
A HIGH is output when the bit is set to 1.

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

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

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

BUFW1																BUFW0															
BUFB3								BUFB2								BUFB1								BUFB0							
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

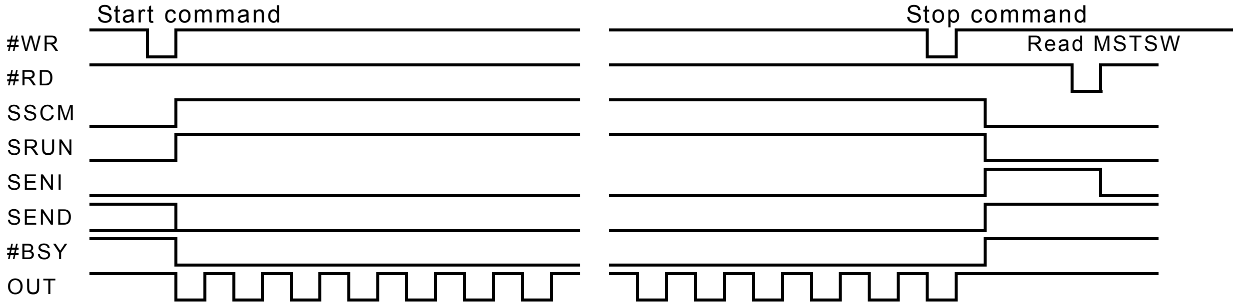
6-5-4. Reading the main status (MSTSW, MSTSB)

MSTSW															
MSTSB1								MSTSB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPDF	SPRF	SEOR	SCP5	SCP4	SCP3	SCP2	SCP1	SSC1	SSC0	SINT	SERR	SEND	SENI	SRUN	SSCM

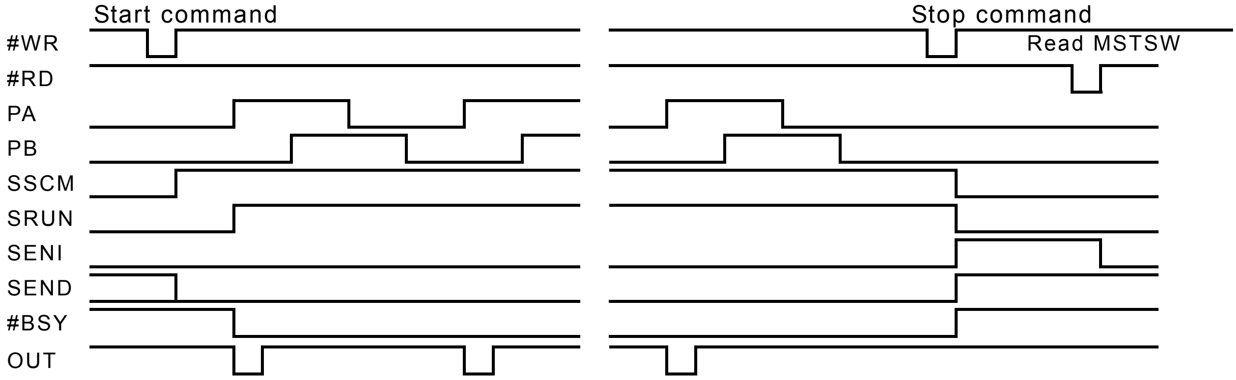
Bit	Bit name	Details
0	SSCM	Set to 1 by writing a start command. Set to 0 when the operation is stopped.
1	SRUN	Set to 1 by the start pulse output. Set to 0 when the operation is stopped.
2	SENI	Stop interrupt flag When RENV2.IEND is 1, the PCL turns ON the INT output when the status changes from operating to stop, and the SENI bit becomes 1. After the main status is read, it returns to 0. However, when RENV5.MSMR (bit 23) is 1, it does not return to 0, but remains 1. When RENV2.IEND is 0, this flag will always be 0.
3	SEND	Set to 0 by writing start command. Set to 1 when the operation is stopped.
4	SERR	Set to 1 when an error interrupt occurs. Set to 0 by reading the REST (in the case of that all REST are 0).
5	SINT	Set to 1 when an event or interrupt occurs. Set to 0 by reading the RIST(in the case of that all RIST are 0).
6 to 7	SSC0 to 1	Sequence number for execution or stopping.
8	SCP1	Set to 1 when the COMPARATOR 1 comparison conditions are met.
9	SCP2	Set to 1 when the COMPARATOR 2 comparison conditions are met.
10	SCP3	Set to 1 when the COMPARATOR 3 comparison conditions are met.
11	SCP4	Set to 1 when the COMPARATOR 4 comparison conditions are met.
12	SCP5	Set to 1 when the COMPARATOR 5 comparison conditions are met.
13	SEOR	When a positioning override cannot be executed (writing the RMV register while stopped), this signal changes to 1. After the main status is read, it changes to 0. However, when RENV5.MSMR (bit 23) is 1, it does not return to 0, but remains 1.
14	SPRF	Set to 1 when the pre-register for the subsequent operation data is full.
15	SPDF	Set to 1 when the pre-register for comparator 5 is full.

Status change timing chart

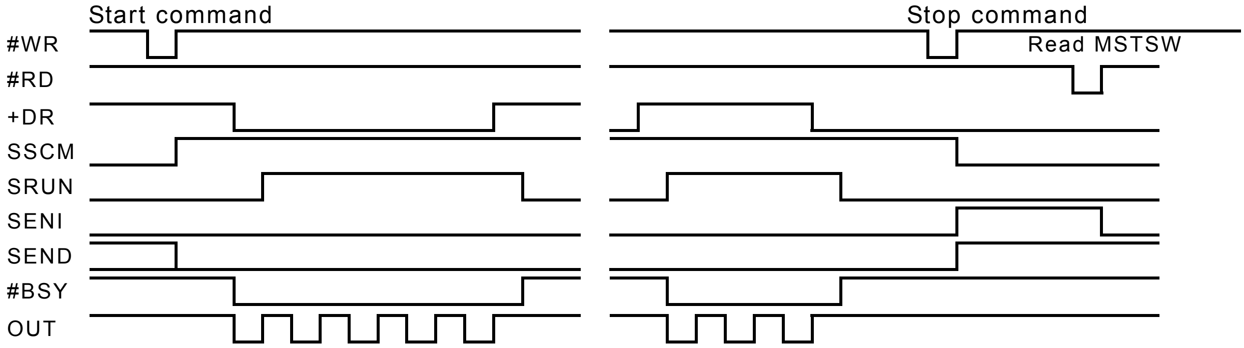
1) When the continuous mode (MOD=00h, 08h) is selected.



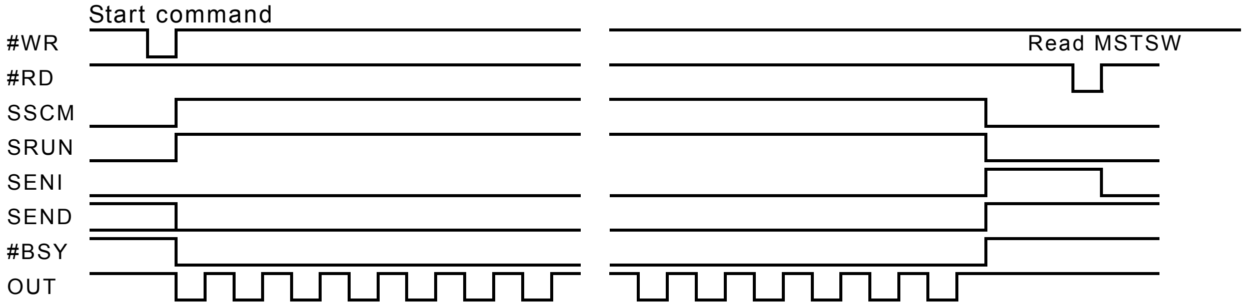
2) When the PA/ PB continuous mode (MOD=01h) is selected.



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



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



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

SSTSW

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

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

Note: When the backlash or slip correction function is used, SFU, SFD and SFC will all be 0. The main Status SRUN will be 1, even if this correction is used.

7. Commands (Operation and Control Commands)

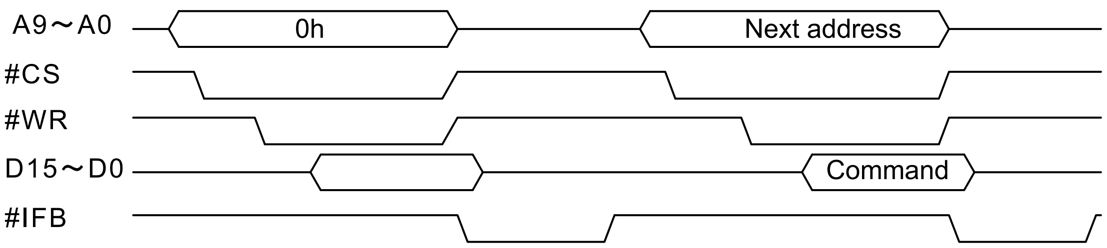
7-1. Operation commands

By writing the command to COMB0 (address 0 when a Z80 I/F is used) after writing the axis assignment data to COMB1 (address 1 when a Z80 I/F is used), the LSI will start and stop, as well as change the speed of the output pulses.
When an 8086, H8, or 68000 I/F is used, write 16-bit data, which combines the axis assignment and operation command data.

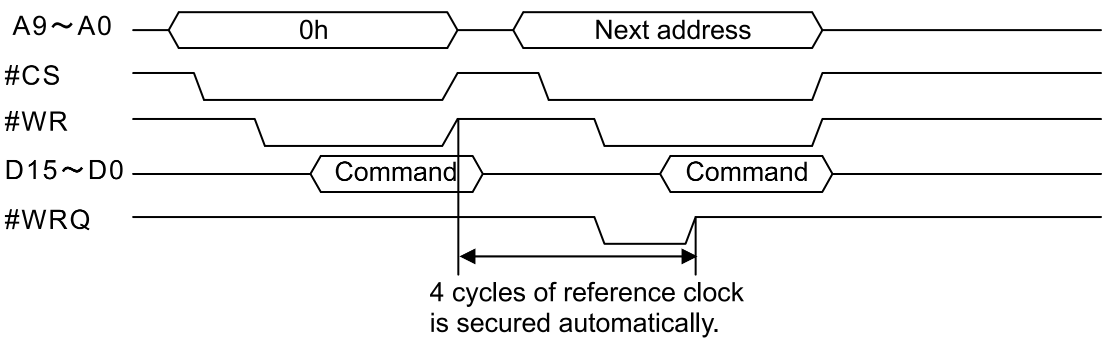
7-1-1. Procedure for writing an operation command (the axis assignment is omitted)

Write a command to COMB0 (address 0 when a Z80 I/F is used). A waiting time of 4 register reference clock cycles (approximately 0.2 μsec when CLK = 19.6608 MHz) is required for the interval between "writing a command" and "writing the next command", "writing a register" and "rewriting the I/O buffer," and between "reading a register" and "reading the I/O buffer." When the #WRQ output signal is used by connecting it to the CPU, the CPU automatically ensures this waiting time.
If you want to use a CPU that does not have this waiting function, arrange the program sequence so that access is only allowed after confirming that the #IFB output signal is HIGH or the time of 4 reference clock cycles is ensured by software.

1) When not using #IFB



2) When using #WRQ



7-1-2. Start command

1) Start command

If this command is 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 the 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. 1
53h	STAUD	High speed start 2 (Acceleration → FH constant speed → Deceleration stop) Note. 1

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

2) Residual pulses start command

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

COMB0	Symbol	Description
54h	CNTFL	Residual pulses FL constant speed start
55h	CNTFH	Residual pulses FH constant speed start
56h	CNTD	Residual pulses high speed start 1 (FH constant speed start without acceleration, with deceleration)
57h	CNTUD	Residual pulses high speed start 2 (With acceleration and deceleration.)

3) Simultaneous start command

By setting the RMD register, the LSI will start an axis which is waiting for #CSTA signal.

COMB0	Symbol	Description
06h	CMSTA	Output one shot of the start pulse from the #CSTA terminal.
2Ah	SPSTA	Only own axis will process the command, the same as when the #CSTA signal is input.

7-1-3. Speed change command

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

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

7-1-4. Stop command

1) Stop command

Write this command to stop feeding while operating.

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

2) Simultaneous stop command

Stop the motor on any axis whose #CSTP input stop function has been enabled by setting the RMD register.

COMB0	Symbol	Description
07h	CMSTP	Outputs one shot of pulses from the #CSTP terminal to stop movement on that axes.

3) Emergency stop command

Stops an axis in an emergency

COMB0	Symbol	Description
05h	CMEMG	Emergency stop (same as a #CEMG signal input)

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

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

7-2. General-purpose output bit control commands

These commands control the individual bits of output terminals P0 to P7.
When the terminals are designated as outputs, the LSI will output signals from terminals P0 to P7. Commands that have not been designated as outputs are ignored.
The write procedures are the same as for the Operation commands.
In addition to this command, by writing to a general-purpose output port (OTPB: Address 2 when a Z80 I/F is used), you can set 8 bits as a group. See section 7-5, "General-purpose output port control command."

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

The P0 and P1 terminals can be set for one shot output (T = approx. 26 msec.) using the RENV2 (Environment setting 2) register, and the output logic can be selected.
To use them as one shot outputs, set the P0 terminal to P0M (bits 0 and 1) = 11, or, set the P1 terminal to P1M (bits 2 and 3) = 11. To change the output logic, set P0L (bit 16) on the P0 terminal and P1L (bit 17) on the P1 terminal.
In order to perform a one-shot output from the P0 and P1 terminals, a bit control command should be written. However, the command you need to write will vary, depending on the output logic selected. See the table below for the details.

Terminal	Logic setting	Bit control command	Terminal	Logic setting	Bit control command
P0	Negative logic (P0L = 0)	P0RST (10h)	P1	Negative logic (P1L = 0)	P1RST (11h)
	Positive logic (P0L = 1)	P0SET (18h)		Negative logic (P1L = 1)	P1SET (19h)

When writing control commands to output ports (OTPB: address 2 for the Z80 interface), the P0 and P1 terminals will not change.

7-3. Control command

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

7-3-1. Software reset command

Used to reset this LSI.

COMB0	Symbol	Description
04h	SRST	Software reset. (Same function as making the #RST terminal LOW.)

7-3-2. Counter reset command

Reset counters to zero.

COMB0	Symbol	Description
20h	CUN1R	Reset COUNTER1 (command position).
21h	CUN2R	Reset COUNTER2 (mechanical position).
22h	CUN3R	Reset COUNTER3 (deflection counter).
23h	CUN4R	Reset COUNTER4 (general-purpose counter).

7-3-3. ERC output control command

Control the ERC signal using commands.

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

7-3-4. Pre-register control command

Make pre-register settings undetermined and transfer pre-register data to a register.
See section "8-2. Pre-registers" in this manual for details about the pre-register.

COMB0	Symbol	Description
26h	PRECAN	Make the operation pre-register undetermined.
27h	PCPCAN	Make the RCMP5 operation pre-register (PRCP5) undetermined.
2Bh	PRESHF	Shift the operation pre-register data.
2Ch	PCPSHF	Shift the RCMP5 operation pre-register data.
4Fh	PRSET	Make data in a pre-register determined as speed pattern change data by a comparator.

7-3-5. PCS input command

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

COMB0	Symbol	Description
28h	STAON	Substitute a PCS terminal input.

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

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

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

7-3-7. Command to reset status

Resets specified bit of main status.

COMB0	Symbol	Description
2Dh	SENIR	Reset MSTSW.SENI.
2Eh	SEORR	Reset MSTSW.SEOR.

7-4. Register control command

There are two access methods: Direct access method and indirect access method.
However, in the case that CPU is connected by reduced address scheme, direct access method cannot be used.

[Direct access method]
It accesses the address corresponding to register directly. In order to sample or change all bits of register simultaneously, 32 bit latch for direct access is integrated.

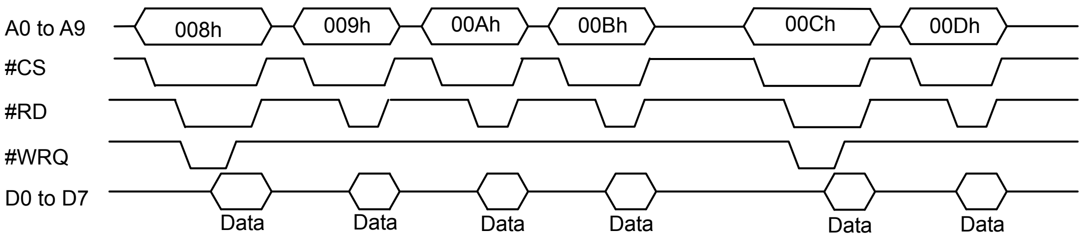
In read cycle of the lowest address of each register for reading out, all bit of register will be copied to a latch, and in read cycle of other address than the lowest address, latched data is read out.

For this process to latch, CPU is made in a wait status by outputting #WRQ signal during the read cycle of the lowest address, it is needed to connect #WRQ output with CPU.
When #WRQ is not used, make #RD signal width more than 4 cycles of CLK.

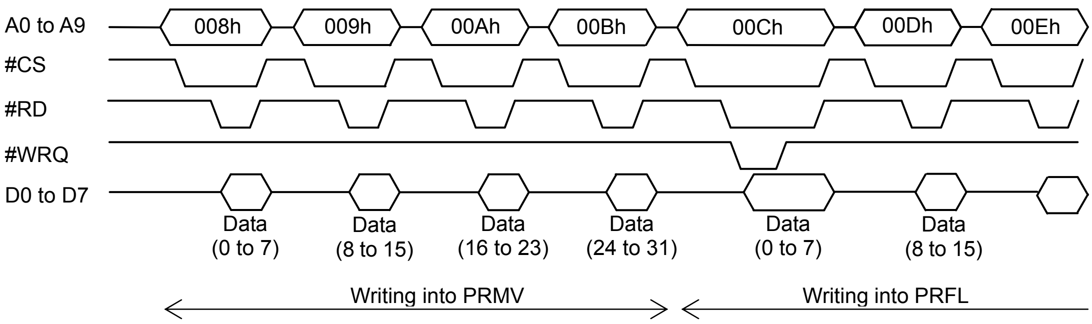
When writing data, written data is stored in 32-bit latch. Just after write cycle of the upper address of each register completes, it copies 32 bits of registers at once. Therefore, even though writing into registers whose bit width are short, it is necessary to write as 4-byte data.

For the process, please access from lower address to upper address in order in 4 byte data basis for both reading out process and writing process when using direct access.

When reading out registers (Z80 I/F)



When writing registers (Z80 I/F)



After writing into address 00Bh, copy process into register starts. If writing into address 00Ch is started while copy process, #WRQ is output.

[Indirect access method]
By writing a Register Control command to COMB0 (Address 0 when a Z80 I/F is used), the LSI can copy data between a register and the I/O buffer.

When reading out, contents of specified register are copied into I/O buffer by writing "Register reading out command." Then, data is output from I/O buffer.

When writing, "Register writing command is written after writing data into I/O buffer, contents of I/O buffer is copied into I/O buffer.

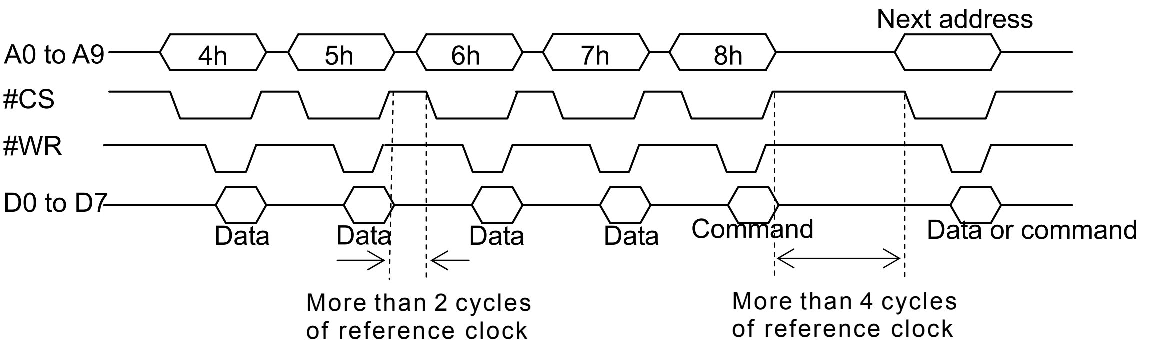
When the I/O buffer is used in the program for responding to an interrupt, note to read the I/O buffer contents before using it, perform PUSH operation it and return it to its original value after use.

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

1) Write the data that will be written to a register into the I/O buffer (addresses 4 to 7 when a Z80 I/F is used). The order in which the data is written does not matter. However, secure two reference clock cycles between these writings.

2) Then, write a "register writing command" to COMB0 (address 0 when a Z80 I/F is used). After writing one set of data, wait at least 4 cycles (approx. 0.2 μsec when CLK = 19.6608 MHz) before writing the next set of data.

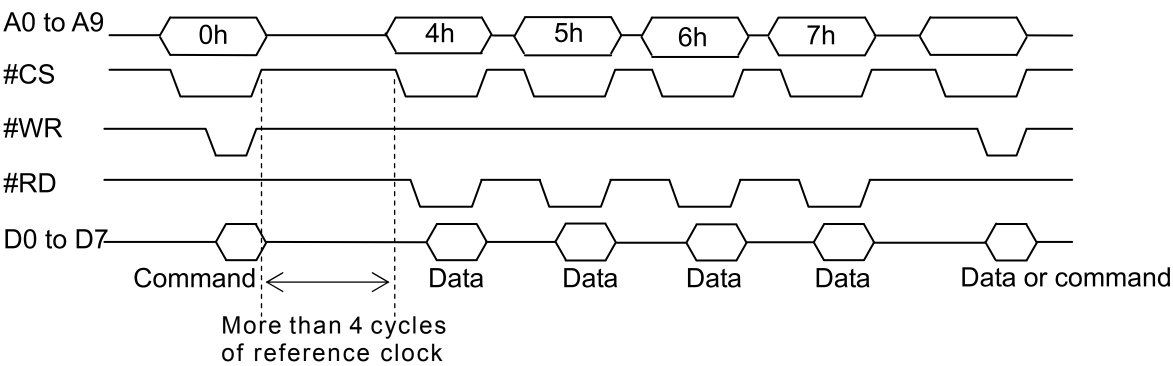
In both case1) and case 2), when the WRQ output is connected to the CPU, the CPU wait control function will provide the waiting time between write operations automatically.



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

1) First, write a "register reading out command" to COMB0 (address 0 when a Z80 I/F is used).
2) Wait at least four reference clock cycles (approx. 0.2 μsec when CLK = 19.6608 MHz) for the data to be copied to the I/O buffer.
3) Read the data from the I/O buffer (addresses 4 to 7 when a Z80 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 #WRQ output is connected to the CPU, the CPU wait control function will provide the waiting time between write operations automatically.



7-4-3. Table of register control commands

No	Detail	Register					2nd pre-register				
		Name	Read command		Write command		Name	Read command		Write command	
			COMB0	Symbol	COMB0	Symbol		COMB0	Symbol	COMB0	Symbol
1	Feed amount, target position	RMV	D0h	RRMV	90h	WRMV	PRMV	C0h	RPRMV	80h	WPRMV
2	Initial speed	RFL	D1h	RRFL	91h	WRFL	PRFL	C1h	RPRFL	81h	WPRFL
3	Operation speed	RFH	D2h	RRFH	92h	WRFH	PRFH	C2h	RPRFH	82h	WPRFH
4	Acceleration rate	RUR	D3h	RRUR	93h	WRUR	PRUR	C3h	RPRUR	83h	WPRUR
5	Deceleration rate	RDR	D4h	RRDR	94h	WRDR	PRDR	C4h	RPRDR	84h	WPRDR
6	Speed magnification rate	RMG	D5h	RRMG	95h	WRMG	PRMG	C5h	RPRMG	85h	WPRMG
7	Ramping-down point	RDP	D6h	RRDP	96h	WRDP	PRDP	C6h	RPRDP	86h	WPRDP
8	Operation mode	RMD	D7h	RRMD	97h	WRMD	PRMD	C7h	RPRMD	87h	WPRMD
9	Circular interpolation center	RIP	D8h	RRIP	98h	WRIP	PRIP	C8h	RPRIP	88h	WPRIP
10	Acceleration S-curve range	RUS	D9h	RRUS	99h	WRUS	PRUS	C9h	RPRUS	89h	WPRUS
11	Deceleration S-curve range	RDS	DAh	RRDS	9Ah	WRDS	PRDS	CAh	RPRDS	8Ah	WPRDS
12	Feed amount correction speed	RFA	DBh	RRFA	9Bh	WRFA					
13	Environment setting 1	RENV1	DCh	RRENV1	9Ch	WRENV1					
14	Environment setting 2	RENV2	DDh	RRENV2	9Dh	WRENV2					
15	Environment setting 3	RENV3	DEh	RRENV3	9Eh	WRENV3					
16	Environment setting 4	RENV4	DFh	RRENV4	9Fh	WRENV4					
17	Environment setting 5	RENV5	E0h	RRENV5	A0h	WRENV5					
18	Environment setting 6	RENV6	E1h	RRENV6	A1h	WRENV6					
19	Environment setting 7	RENV7	E2h	RRENV7	A2h	WRENV7					
20	COUNTER1 (command position)	RCUN1	E3h	RRCUN1	A3h	WRCUN1					
21	COUNTER2 (mechanical position)	RCUN2	E4h	RRCUN2	A4h	WRCUN2					
22	COUNTER3 (deflection counter)	RCUN3	E5h	RRCUN3	A5h	WRCUN3					
23	COUNTER4 (general purpose)	RCUN4	E6h	RRCUN4	A6h	WRCUN4					
24	Data for comparator 1	RCMP1	E7h	RRCMP1	A7h	WRCMP 1					
25	Data for comparator 2	RCMP2	E8h	RRCMP2	A8h	WRCMP 2					
26	Data for comparator 3	RCMP3	E9h	RRCMP3	A9h	WRCMP 3					
27	Data for comparator 4	RCMP4	EAh	RRCMP4	AAh	WRCMP 4					
28	Data for comparator 5	RCMP5	EBh	RRCMP5	ABh	WRCMP 5	PRCP5	CBh	RPRCP5	8Bh	WPRCP5
29	Event INT setting	RIRQ	ECh	RRIRQ	ACh	WRIRQ					

No	Detail	Register					2nd pre-register				
		Name	Read command		Write command		Name	Read command		Write command	
			COMB0	Symbol	COMB0	Symbol		COMB0	Symbol	COMB0	Symbol
30	COUNTER1 latched data	RLTC1	EDh	RRLTC1							
31	COUNTER2 latched data	RLTC2	EEh	RRLTC2							
32	COUNTER3 latched data	RLTC3	EFh	RRLTC3							
33	COUNTER4 latched data	RLTC4	F0h	RRLTC4							
34	Extension status	RSTS	F1h	RRSTS							
35	Error INT status	REST	F2h	RREST	B2h	WREST					
36	Event INT status	RIST	F3h	RRIST	B3h	WRIST					
37	Positioning counter	RPLS	F4h	RRPLS							
38	EZ counter, speed monitor	RSPD	F5h	RRSPD							
39	Ramping-down point	PSDC	F6h	RPSDC							
40	Circular interpolation stepping number	RCI	FCh	RRCI	BCh	WRCI	PRCI	CCh	RPRCI	8Ch	WPRCI
41	Circular interpolation stepping counter	RCIC	FDh	RRCIC							
42	Interpolation status	RIPS	FFh	RRIPS							

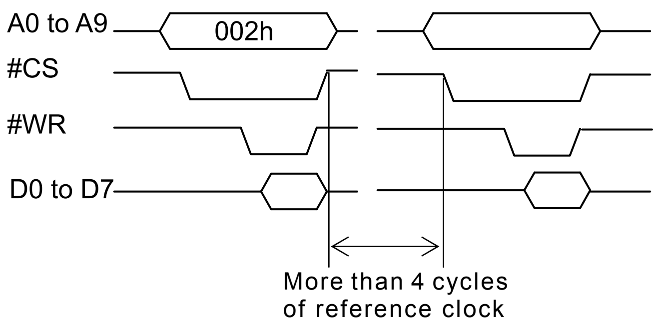
7-5. General-purpose output port control command

By writing an output control command to the output port (OTPB: Address 2 when using a Z80 interface), the PCL will control the output of the P0 to P7 terminals.
When the I/O setting for P0 to P7 is set to output, the PCL will output signals from terminals P0 to P7 to issue the command.
When writing words to the port, the upper 8 bits are discarded. However, they should be set to zero to maintain future compatibility.
The output status of terminals P0 to P7 is latched, even after the I/O setting is changed to input.
The output status for each terminal can be set individually using the bit control command.

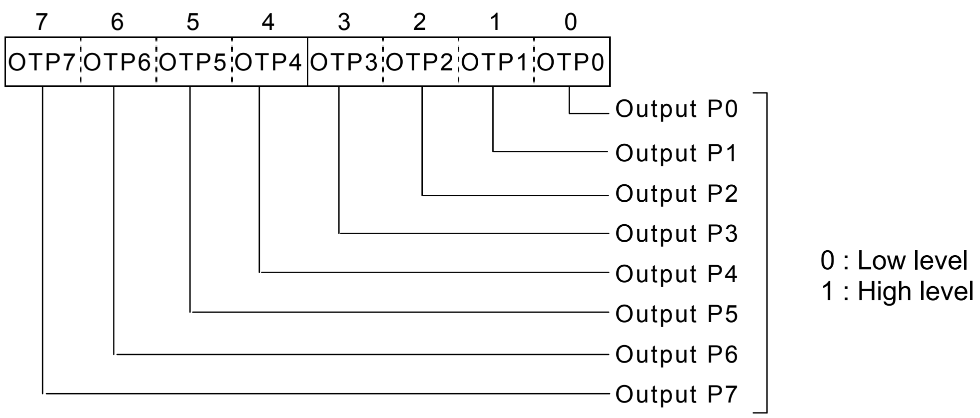
7-5-1. Command writing procedures

Write control data to output port (OTPB: Address 2 when a Z80 I/F is used).

To continue with the next command, the LSI must wait for four reference clock cycles (approx. 0.2 μsec when CLK = 19.6608 MHz). The #WRQ terminal outputs a wait request signal.



7-5-2 Command bit allocation



8. Registers

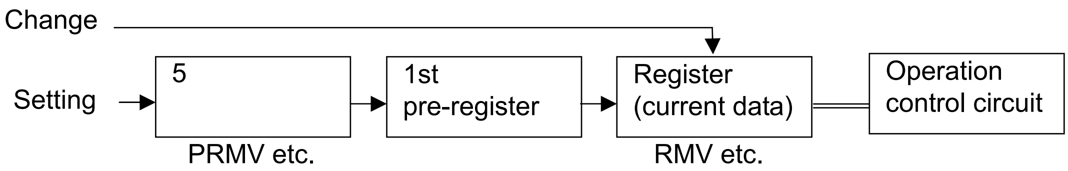
8-1. Table of registers

The following registers are available for each axis.

No.	Register name	Bit length	R/W	Details	2nd pre-register name
1	RMV	32	R/W	Feed amount, target position	PRMV
2	RFL	16	R/W	Initial speed	PRFL
3	RFH	16	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	Circular interpolation center position, master axis feed amount with linear interpolation and with multiple chips	PRIP
10	RUS	15	R/W	S-curve acceleration range	PRUS
11	RDS	15	R/W	S-curve deceleration range	PRDS
12	RFA	16	R/W	Speed at amount correction	
13	RENV1	32	R/W	Environment setting 1 (specify I/O terminal details)	
14	RENV2	32	R/W	Environment setting 2 (specify general-purpose port details)	
15	RENV3	32	R/W	Environment setting 3 (specify origin return and counter details)	
16	RENV4	32	R/W	Environment setting 4 (specify details for comparators 1 to 4)	
17	RENV5	28	R/W	Environment setting 5 (specify details for comparator 5)	
18	RENV6	32	R/W	Environment setting 6 (specify details for feed amount correction)	
19	RENV7	32	R/W	Environment setting 7 (specify vibration reduction control details)	
20	RCUN1	32	R/W	COUNTER1 (command position)	
21	RCUN2	32	R/W	COUNTER2 (mechanical position)	
22	RCUN3	16	R/W	COUNTER3 (deflection counter)	
23	RCUN4	32	R/W	COUNTER4 (general-purpose counter)	
24	RCMP1	32	R/W	Comparison data for comparator 1	
25	RCMP2	32	R/W	Comparison data for comparator 2	
26	RCMP3	32	R/W	Comparison data for comparator 3	
27	RCMP4	32	R/W	Comparison data for comparator 4	
28	RCMP5	32	R/W	Comparison data for comparator 5	PRCP5
29	RIRQ	19	R/W	Specify event interruption cause	
30	RLTC1	32	R	COUNTER1 (command position) latch data	
31	RLTC2	32	R	COUNTER2 (mechanical position) latch data	
32	RLTC3	16	R	COUNTER3 (deflection counter) latch data	
33	RLTC4	32	R	COUNTER4 (general-purpose) latch data	
34	RSTS	24	R	Extension status	
35	REST	18	R/W	Error INT status	
36	RIST	20	R/W	Event INT status	
37	RPLS	32	R	Positioning counter (number of residual pulses to feed)	
38	RSPD	23	R	EZ counter, current speed monitor	
39	RSDC	24	R	Automatically calculated ramping-down point	
40	RCI	32	R/W	Number of steps for circular interpolation	PRCI
41	RCIC	32	R	Circular interpolation step counter	
42	RIPS	24	R	Interpolation status	

8-2. Pre-registers

The following registers and start commands have pre-registers:
RMV, RFL, RFH, RUR, RDR, RMG, RDP, RMD, RIP, RUS, RDS, RCI, and RCMP5.
The term pre-register refers to a register which contains the next set of operation data while the current step is executing. This LSI has the following 2-layer structure and executes FIFO operation.
The pre-registers consist of two groups: the operation pre-registers (PRMV, PRFL, PRFH, PRUR, PRDR, PRMG, PRDP, PRMD, PRIP, PRUS, PRDS, PRCI) and the comparator pre-register (PRCP5).



8-2-1. Writing to the operation pre-registers

The pre-registers have a two-layer structure and each register can contain up to two pieces of operation data. Write the data to a pre-register (P register name). Registers that don't need to be changed do not need to be rewritten.
When the PCL stops its current operation, the data you wrote to the pre-registers is shifted into the working registers and used as the current data. When the PCL is operating, the data remains stored as pre-register data. The data will be transferred into the pre-registers when a start command is issued.
When the current operation completes, the data will be shifted into the working registers and the PCL starts the new operation automatically. The status of the pre-registers can be checked by reading PFM in the RSTS register. When the PFM is value is "11," SPRF in the main status register (MSTSW) changes to "1". Writing data while the pre-register is full is not allowed.
To change the current operating status before the operation is complete, such as when you want to change the speed, write the new data directly to the working register.

The relationship between the write status of the pre-registers and the possible PFC values are as follows.

Procedure	2nd pre-register	1st pre-register	Working register	PFM	SPRF
Initial status	0 Undetermined	0 Undetermined	0 Undetermined	00	0
Write Operation Data 1	Data 1 is undetermined	Data 1 is undetermined	Data 1 is undetermined	00	0
Write a Start command	Data 1 is undetermined	Data 1 is undetermined	Data 1 is determined	01	0
Write Operation Data 2 and a Start command while in operation	Data 2 is undetermined	Data 2 is determined	Data 1 is determined	10	0
Write Operation Data 3 and Start command while in operation	Data 3 is determined	Data 3 is determined	Data 1 is determined	11	1
The operation using Operation Data 1 is complete	Data 3 is undetermined	Data 3 is determined	Data 2 is determined	10	0

Also, by setting an event interrupt cause in the RIRQ register (IRNM), the PCL can be set to output an #INT signal as the 2nd pre-register changes from "determined" to "undetermined" status when the operation is complete.

Note: When you want the next operation to start automatically using the pre-registers, set the operation completion timing to "cycle completion (METM = 0 on PRMD)." When pulse completion (METM = 1 on PRMD)" is set, the time between the last pulse and next operation start pulse will be as little as 15x T_{CLK} (T_{CKL}: Reference clock cycle).
For details, see 11-3-2. "Control the output pulse width and operation completion timing."

8-2-2. Cancel the operation pre-register

Use a pre-register Cancel command (26h) and a Stop command (49h, 4Ah) to cancel all the data in the pre-registers, and their status then becomes undetermined. The pre-register data are also cancelled if the PCL stops with an error.

8-2-3. Writing to the comparator pre-registers

Comparator 5 has pre-registers. To overwrite the current data, write directly to RCMP5. To write to the pre-register, write to PRCP5.
The comparator data will be determined only by writing to PRCP5. The status of the comparator pre-register can be checked by reading PFC in the RSTS register. When the PFC value is 11, SPDF in the main status register (MSTSW) will be 1. Writing data to the pre-register when it is full is not allowed.
After the conditions have been established, the comparator data in the pre-register will be shifted when the condition changes from false to true.
Comparator data can be written regardless of axis' motion (stopped/operating).
-
The relationship between the pre-register writing status and the PFC values are as follows.

Procedure	2nd pre-register	1st pre-register	Working register	PFC	SPDF
Initial status	0 Undetermined	0 Undetermined	0 Undetermined	00	0
Write Data 1 to PRCP5	Data 1 is undetermined	Data 1 is undetermined	Data 1 is determined	01	0
Write Data 2 to PRCP5	Data 2 is undetermined	Data 2 is determined	Data 1 is determined	10	0
Write Data 3 to PRCP5	Data 3 is determined	Data 2 is determined	Data 1 is determined	11	1
Comparison result for Data 1 changes from true to false	Data 3 is undetermined	Data 3 is determined	Data 2 is determined	10	0

Also, by setting an event interrupt cause in the RIRQ register (IRND), the PCL can be set to output an #INT signal as the 2nd pre-register changes from "determined" to "undetermined" status when the operation is complete.

8-2-4. Cancel the comparator pre-register data

The pre-register cancel command (27h) will cancel the pre-register data and its status becomes undetermined. However, please note that the register will not change to the undetermined status.

8-3.Description of the registers

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

8-3-1. PRMV (RMV) register

This register is used to specify the target position for positioning operations. The set details may vary with each operation mode.
PMV 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 to +2,147,483,647.
By changing the RMV register while in operation, the feed length 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 to 65,535. 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 working register for PRFH. Write to this register to override the current speed.

The setting range is 1 to 65,535. However, the actual speed [pps] may vary with the speed magnification rate set in the PRMG register.

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

8-3-4. 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 to 65,535.

Note 1: Bits marked with an "*" (asterisk) will be ignored when written and are 0 when read.
Note 2: Bits marked with an "&" symbol will be ignored when written and will be the same value as the upper most bit among the non-marked bits. (Sign extension)

8-3-5. PRDR (RDR) register

This pre-register is used to specify the 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 to 65,535.
When PRDR = 0, the deceleration rate will be the value set by PRUR.

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

[Setting example when the reference clock is 19.6608 MHz]

Setting	Speed magnification rate	Operation speed setting range [pps]	Setting	Speed magnification rate	Operation speed setting range [pps]
2999	0.1x	0.1 to 6,553.5	59	5x	5 to 327,675
1499	0.2x	0.2 to 13,107.0	29	10x	10 to 655,350
599	0.5x	0.5 to 32,767.5	14	20x	20 to 1,310,700
299	1x	1 to 65,535	5	50x	50 to 3,276,750
149	2x	2 to 131,070	2	100x	100 to 6,553,500

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 2nd 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 are ignored when written and change their setting when read according to the setting of MSDP (bit 13) in the PRMD register.

MSDP	Setting details	bit #	bit #
0	Offset for automatically set values. When a positive value is entered, an axis will start deceleration earlier and the FL speed range will be used longer. When a negative value is entered, an axis will start deceleration later and will not reach the FL speed.	Same as bit 23	-8,388,608 to +8,388,607
1	When number of pulses left drops to less than a set value, an axis starts to decelerate.	0	0 to +16,777,215

Note 1: Bits marked with an "*" (asterisk) will be ignored when written and are 0 when read.
Note 2: Bits marked with an "&" symbol will be ignored when written and will be the same value as the upper most bit among the non-marked bits. (Sign extension.)

8-3-8. PRMD (RMD) register

This pre-register is used to set the operation mode.
RMD is the register for PRMD.

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

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

Bits	Bit name	Description
Setting basic operation mode		
0 to 6	MOD	Set operation mode. 000 0000 (00h): Continuous positive rotation controlled by command control. 000 1000 (08h): Continuous negative rotation controlled by command control. 000 0001 (01h): Continuous operation controlled by pulsar (PA/PB) input. 000 0010 (02h): Continuous operation controlled by external signal (+DR/-DR) input. 001 0000 (10h): Positive rotation origin return operation. 001 1000 (18h): Negative rotation origin return operation. 001 0010 (12h): Positive feed leaving from the origin position. 001 1010 (1Ah): Negative feed leaving from the origin position. 001 0101 (15h): Origin search in the positive direction 001 1101 (1Dh): Origin search in the negative direction 010 0000 (20h): Feed to +EL or +SL position. 010 1000 (28h): Feed to -EL or -SL position. 010 0010 (22h): Move away from the -EL or -SL position. 010 1011 (2Ah): Move away from the +EL or +SL position. 010 0100 (24h): Feed in the positive direction for a specified number of EZ counts. 010 1100 (2Ch): Feed in the negative direction for a specified number of EZ counts. 100 0001 (41h): Positioning operation (specify the incremental target position) 100 0010 (42h): Positioning operation (specify the absolute position in COUNTER1) 100 0010 (43h): Positioning operation (specify the absolute position in COUNTER2) 100 0100 (44h): Zero return of command position (COUNTER1). 100 0101 (45h): Zero return of mechanical position (COUNTER2). 100 0110 (46h): Single pulse operation in the positive direction. 100 1110 (4Eh): Single pulse operation in the negative direction. 100 0111 (47h): Timer operation 101 0001 (51h): Positioning operation controlled by pulsar (PA/PB) input. 101 0010 (52h): Positioning operation is synchronized with PA/PB (specify the absolute position of COUNTER1) 101 0011 (53h): Positioning operation is synchronized with PA/PB (specify the absolute position of COUNTER2) 101 0100 (54h): Zero return to the command position controlled by pulsar (PA/PB) input. 101 0101 (55h): Zero return to a mechanical position controlled by pulsar (PA/PB) input. 101 0110 (56h): Positioning operation controlled by external signal (+DR/-DR) input. 110 0000 (60h): Continuous linear interpolation 1 (continuous operation with linear interpolation 1) 110 0001 (61h): Linear interpolation 1 110 0010 (62h): Continuous linear interpolation 2 (continuous operation with linear interpolation 2) 110 0011 (63h): Linear interpolation 2 110 0100 (64h): CW circular interpolation operation 110 0101 (65h): CCW circular interpolation operation.

Bits	Bit name	Description
		110 0110 (66h): Clockwise circular interpolation, synchronized with the U axis (circular linear interpolation) 110 0111 (67h): Counter-clockwise circular interpolation, synchronized with the U axis (circular linear interpolation) 110 1000 (68h): Continuous linear interpolation 1, synchronized with PA/PB 110 1001 (69h): Linear interpolation 1, synchronized with PA/PB 110 1010 (6Ah): Continuous linear interpolation 2, synchronized with PA/PB. 110 1011 (6Bh): Linear interpolation 2, synchronized with PA/PB. 110 1100 (6Ch): Clockwise circular interpolation, synchronized with PA/PB 110 1101 (6Dh): Counter-clockwise circular interpolation, synchronized with PA/PB 110 1111 (6Fh): Dummy circular interpolation
7	MENI	1: When the pre-register is set, the PCL will not output an INT signal, even if IEND becomes 1.
8	MSDE	0: SD input will be invalid. (Checking can be done with sub status (SSTSW) or extended status (RSTS)) 1: Decelerates (deceleration stop) by turning ON the input.
9	MINP	0: Delay using an INP input will be disabled. (Checking can be done with extended status (RSTS)) 1: Completes operation by turning ON the INP input.
10	MSMD	Specify an acceleration/deceleration type for high speed feed. (0: Linear accel/decel. 1: S-curve accel/decel.)
11	MCCE	1: Stop COUNTER1 (command position) This is used to move a mechanical part without changing the PCL control position
12	METM	Specify the operation completion timing. (0: End of cycle. 1: End of pulse.) When using the vibration reduction function, select "End of pulse."
13	MSDP	Specify the ramping-down point for high speed feed. (0: Automatic setting. 1: Manual setting.) Effective for positioning operations and linear interpolation feeding.
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	MIPF	1: Make synthetic speed constant while performing interpolation feeding.
16 to 17	MSN0 to 1	When you want to control an operation block, specify a sequence number using 2 bits. By reading the main status (MSTSW), a sequence number currently being executed (SSC0 to 1) can be checked. Setting the sequence number does not affect the operation.
18 to 19	MSY0 to 1	After writing a start command, the LSI will start an axis synchronization operation based on other timing. 00: Starts immediately. 01: Starts on a #CSTA input (or command 06h, 2Ah). 10: Starts with an internal synchronous start signal. 11: Starts when a specified axis stops moving.
20 to 23	MAX 0 to 3	Specify an axis to check for an operation stop when the value of MSY 0 to 1 is 11. Setting examples 0001: Starts when the X axis stops. 0010: Starts when the Y axis stops. 0100: Starts when the Z axis stops. 1000: Starts when the U axis stops. 0101: Starts when both the X and Z axes stop. 1111: Starts when all axes stop.
24	MSPE	1: Deceleration stop or immediate stop by #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 #CSTP (simultaneous stop) signal when stopping due to an error.
26	MADJ	Specify an FH correction function. (0: ON. 1: OFF.) When S-shaped deceleration is selected (MSMD = 1) and the operation is set to use linear interpolation 1 (MOD = 61h) with a synthesized speed constant control (MIPF = 1), make sure to turn this bit ON.
27	MPIE	1: After the circular interpolation operation is complete, the PCL will draw to the end point automatically.

Bits	Bit name	Description
28	MIPM	0: Make conditions for circular interpolation completion same as PCL6045B's. 1: Define a new condition for circular interpolation completion.
29	MSDC	0: Uses count method only when interpolation operation is performed with constant synthesized speed control like PCL6045B. Otherwise, calculation method is used. 1: Fix the method to set ramp-down point automatically, to "count method".
30 to 31	Not defined	(Always set to 0.)

8-3-9. PRIP (RIP) register

This pre-register is used to set the center position for circular interpolation or a master axis feed amount for linear interpolation 2.
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 MOD (bits 0 to 6) of the PRMD register is set as shown below, the register is enabled.
  - 110 0010 (62h): Continuous linear interpolation 2 (continuous operation with the linear interpolation 2).
  - 110 0011 (63h): Linear interpolation 2.
  - 110 0100 (64h): Circular interpolation in a CW direction.
  - 110 0101 (65h): Circular interpolation in a CCW direction.
- With Continuous linear interpolation 2 and Linear interpolation 2, specify the feed amount on the master axis using an incremental value.
- With circular interpolation, enter a circular center position using an incremental value.
- Setting range: -2,147,483,648 to +2,147,483,647

8-3-10. PRUS (RUS) register

This pre-register is used to specify the S-curve range of the 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 to 32,767.
When 0 is entered, the value of (PRFH - PRFL)/2 will be calculated internally and applied.

8-3-11. PRDS (RDS) register

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

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

The normal setting range is 1 to 32,767.
When 0 is entered, the value of (PRFH - PRFL)/2 will be calculated internally and applied.

8-3-12. RFA register

This register is used to specify the constant speed for backlash correction or slip correction.
This is also used as a reverse constant speed for an origin return operation.

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

Although the setting range is 1 to 65,535, the actual speed [pps] varies with the speed magnification rate setting in the RMG register.

Note 1: Bits marked with an "*" (asterisk) will be ignored when written and are 0 when read.
Note 2: Bits marked with an "&" symbol will be ignored when written and will be the same value as the upper most bit among the non-marked bits. (Sign extension)

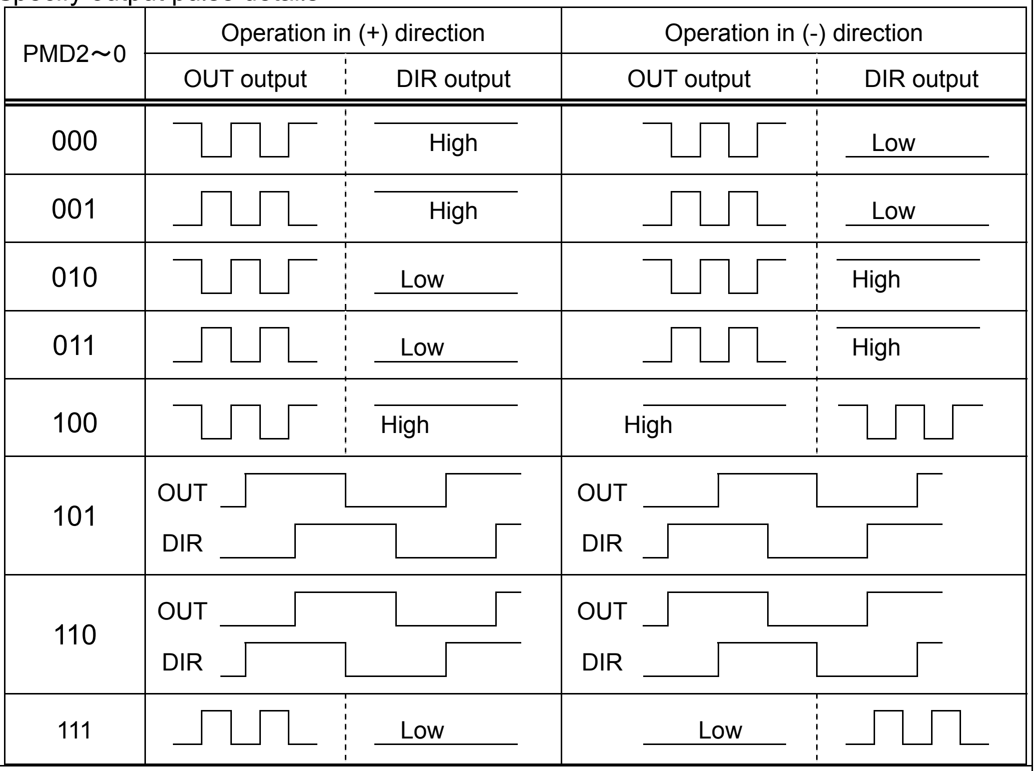
8-3-13. RENV1 register

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
PDTC	PCSM	INTM	DTMF	DRF	FLTR	DRL	PCSL	LTCL	INPL	CLR1	CLR0	STPM	STAM	ETW1	ETW0

Bits	Bit name	Description
0 to 2	PMD0 to 2	Specify output pulse details (see table below)



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

Bits	Bit name	Description
3	ELM	Specify the process to occur when the EL input is turned ON. (0: Immediate stop. 1: Deceleration stop) Note 1
4	SDM	Specify the process to occur when the SD input is turned ON. (0: Deceleration only. 1: Deceleration stop.)
5	SDLT	Specify the latch function of the SD input. (0: OFF. 1: ON.) Turns ON when the SD signal width is short. When the SD input is OFF while starting, the latch signal is reset. The latch signal is also reset when SDLT is 0.
6	SDL	Specify the SD signal input logic. (0: Negative logic. 1: Positive logic.)
7	ORGL	Specify the ORG signal input logic. (0: Negative logic. 1: Positive logic.)
8	ALMM	Specify the process to occur when the ALM input is turned ON. (0: Immediate stop. 1: Deceleration stop.)
9	ALML	Specify the ALM signal input logic. (0: Negative logic. 1: Positive logic.)
10	EROE	1: Automatically outputs an ERC signal when the axis is stopped immediately by a +EL, -EL, ALM, or #CEMG input signal. However, the ERC signal is not output when a deceleration stop occurs on the axis. Even if the EL signal is specified for a normal stop, by setting MOD = "010X000" (feed to the EL position) in the RMD register, the ERC signal is output if an immediate stop occurs.
11	EROR	1: Automatically output the ERC signal when the axis completes an origin return.
12 to 14	EPW0 to 2	Specify the pulse width of the ERC output signal. 000: 12 μsec 001: 102 μsec 010: 409 μsec 011: 1.6 msec 100: 13 msec 101: 52 msec 110: 104 msec 111: Level output
15	ERCL	Specify the ERC signal output logic. (0: Negative logic. 1: Positive logic.)

Bits	Bit name	Description
16 to 17	ETW0 to 1	Specify the ERC signal OFF timer time. 00: 0 μsec 10:1.6 msec 01:12 μsec 11:104 msec
18	STAM	Specify the #CSTA signal input type. (0: Level trigger. 1: Edge trigger.)
19	STPM	Specify a stop method using #CSTP input. (0: Immediate stop. 1: Deceleration stop.) Note 2
20 to 21	CLR0 to 1	Specify a CLR input. 00: Clear on the falling edge 10: Clear on a LOW. 01: Clear on the rising edge 11: Clear on a HIGH.
22	INPL	Specify the INP signal input logic. (0: Negative logic. 1: Positive logic.)
23	LTCL	Specify the operation edge for the LTC signal. (0: Falling. 1: Rising)
24	PCSL	Specify the PCS signal input logic. (0: Negative logic. 1: Positive logic.)
25	DRL	Specify the +DR, -DR signal input logic. (0: Negative logic. 1: Positive logic.)
26	FLTR	1: Apply a filter to the +EL, -EL, SD, ORG, ALM, or INP inputs. When a filter is applied, signal pulses shorter than 4 μsec are ignored.
27	DRF	1: Apply a filter to the +DR, -DR, or PE inputs. When a filter is applied, signals pulses shorter than 32 msec are ignored.
28	DTMF	1: Turn OFF the direction change timer (0.2 msec) function.
29	INTM	1: Mask an INT output. (Changes the interrupt circuit.)
30	PCSM	1: Make PCS input as a #CSTA signal for only the own axis.
31	PDTC	1: Keep the pulse width at a 50% duty cycle.

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

8-3-14. RENV2 register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P7M1	P7M0	P6M1	P6M0	P5M1	P5M0	P4M1	P4M0	P3M1	P3M0	P2M1	P2M0	P1M1	P1M0	P0M1	P0M0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POFF	EOFF	SMAX	PMSK	IEND	PDIR	PIM1	PIM0	EZL	EDIR	EIM1	EIM0	PINF	EINF	P1L	P0L

Bits	Bit name	Description
0 to 1	P0M0 to 1	Specify the operation of the P0/FUP terminals 00: General-purpose input 01: General-purpose output 10: Output the FUP (acceleration) signal. 11: General-purpose one shot signal output (T = 26 msec) Note: 1
2 to 3	P1M0 to 1	Specify the operation of the P1/FDW terminals 00: General-purpose input 01: General-purpose output 10: Output the FDW (deceleration) signal. 11: General-purpose one shot signal output (T = 26 msec) Note: 1
4 to 5	P2M0 to 1	Specify the operation of the P2/MVC terminal. 00: General-purpose input 01: General-purpose output 10: Output the MVC (constant speed feeding) signal with negative logic. 11: Output the MVC (constant speed feeding) signal with positive logic.
6 to 7	P3M0 to 1	Specify the operation of the P3/CP1 (+SL) terminals. 00: General-purpose input 01: General-purpose output 10: Output the CP1 (satisfied the Comparator 1 conditions) signal with negative logic. 11: Output the CP1 (satisfied the Comparator 1 conditions) signal with positive logic.
8 to 9	P4M0 to 1	Specify the operation of the P4/CP2 (-SL) terminals. 00: General-purpose input 01: General-purpose output 10: Output the CP2 (satisfied the Comparator 2 conditions) signal with negative logic. 11: Output the CP2 (satisfied the Comparator 2 conditions) signal with positive logic.
10 to 11	P5M0 to 1	Specify the operation of the P5/CP3 terminals. 00: General-purpose input 01: General-purpose output 10: Output the CP3 (satisfied the Comparator 3 conditions) signal with negative logic. 11: Output the CP3 (satisfied the Comparator 3 conditions) signal with positive logic.
12 to 13	P6M0 to 1	Specify the operation of the P6/CP4 terminals. 00: General-purpose input 01: General-purpose output 10: Output the CP4 (satisfied the Comparator 4 conditions) signal with negative logic. 11: Output the CP4 (satisfied the Comparator 4 conditions) signal with positive logic.
14 to 15	P7M0 to 1	Specify the operation of the P7/CP5 terminals. 00: General-purpose input 01: General-purpose output 10: Output the CP5 (satisfied the Comparator 5 conditions) signal with negative logic. 11: Output the CP5 (satisfied the Comparator 5 conditions) signal with positive logic.
16	P0L	Specify the output logic when the P0 terminal is used for FUP or as a one shot. (0: Negative logic. 1: Positive logic.)
17	P1L	Specify the output logic when the P1 terminal is used for FDW or as a one shot. (0: Negative logic. 1: Positive logic.)
18	EINF	1: Apply a noise filter to EA/EB/EZ input.

Bits	Bit name	Description
		Ignores pulse inputs less than 3 CLK signal cycles long.
19	PINF	1: Apply a noise filter to PA/PB input. Ignore pulse inputs less than 3 CLK signal cycles long.
20 to 21	EIM0 to 1	Specify the EA/EB input operation. 00: Multiply a 90˚ phase difference by 1 (Count up (count forward) when the EA input phase is ahead.) 01: Multiply a 90˚ phase difference by 2 (Count up (count forward) when the EA input phase is ahead.) 10: Multiply a 90˚ phase difference by 4 (Count up (count forward) when EA input phase is ahead.) 11: Count up (count forward) when the EA signal rises, count down when the EB signal rises.
22	EDIR	1: Reverse the counting direction of the EA/EB inputs.
23	EZL	Specify EZ signal input logic. (0: Falling edge. 1: Rising edge.)
24 to 25	PIM0 to 1	Specify the PA/PB input operation. 00: Multiply a 90˚ phase difference by 1 (Count up (count forward) when the PA input phase is ahead.) 01: Multiply a 90˚ phase difference by 2 (Count up (count forward) when the PA input phase is ahead.) 10: Multiply a 90˚ phase difference by 4 (Count up (count forward) when PA input phase is ahead.) 11: Count up (count forward) when the PA signal rises, count down when the PB signal rises.
26	PDIR	1: Reverse the counting direction of the PA/PB inputs.
27	IEND	1: Outputs an INT signal when stopping, regardless of whether the stop is normal or due to an error.
28	PMSK	1: Masks output pulses.
29	SMAX	1: Enable a start operation that is triggered by stop on the own axis.
30	EOFF	1: Disable EA/EB input.
31	POFF	1: Disable PA/PB input.

Note 1: For details about outputting a general-purpose one shot signal, see 7-2 "General-purpose output bit control commands."

8-3-15. RENV3 register

This is a register for the Environment 3 settings. Origin return methods and counter operation specifications are the main function of this register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	BSYC	CI41	CI40	CI31	CI30	CI21	CI20	EZD3	EZD2	EZD1	EZD0	ORM3	ORM2	ORM1	ORM0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CU4H	CU3H	CU2H	0	CU4B	CU3B	CU2B	CU1B	CU4R	CU3R	CU2R	CU1R	CU4C	CU3C	CU2C	CU1C

Bit	Bit name	Description
0 to 3	ORM0 to 3	Specify an origin method. 00000: Origin return operation 0 - The axis will stop immediately (or make a deceleration stop when feeding at high speed) when the ORG input turns ON. - COUNTER reset timing: When the ORG input turns ON. 0001: Origin return operation 1 - The axis will stop immediately (or make a deceleration stop when feeding at high speed) when the ORG input turns ON. Then, it will feed in the opposite direction at RFA constant speed until the ORG input turns OFF. Then, the axis will move back in the original direction at RFA speed and stop instantly when ORG input turns ON again. - COUNTER reset timing: When the ORG input signal turns ON. 0010: Origin return operation 2 - After the ORG input turns ON when feeding at constant speed, the LSI will start counting EZ pulses. The axis will stop immediately when the LSI finishes counting EZ pulses. After the ORG input turns ON when feeding at high speed, the axis will start decelerating. At the same time, the LSI will start counting EZ pulses. When the LSI finishes counting EZ pulses, the axis will stop instantly. - COUNTER reset timing: When finishing counting EZ pulses. 0011: Origin return operation 3 - After the ORG signal turns ON when feeding at constant speed, the LSI will start counting EZ pulses. The axis will stop instantly when the LSI finishes counting EZ pulses. After the ORG signal turns ON when feeding at high speed, the LSI will start counting EZ pulses. When the LSI finishes counting EZ pulses, the axis will decelerate and stop. When feeding at constant speed, movement on the axis stops immediately by counting the EZ signal after the ORG input is turned ON. When feeding at high speed, the axis will decelerate and stop by counting the EZ signal after the ORG input is turned ON. 0100: Origin return operation 4 - After the ORG input turns ON when feeding at constant speed , the axis will stop immediately (or make a deceleration stop when feeding at high speed). Then, the axis will start feeding in the opposite direction at RFA constant speed. After the ORG input turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting EZ pulses, the axis will stop instantly. - COUNTER reset timing: When finishing counting the EZ pulses. 0101: Origin return operation 5 - After the ORG input turns ON when feeding at constant speed, the axis will stop immediately (or make a deceleration stop when feeding at high speed). Then, the axis will start feeding in the opposite direction. After the ORG input turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting EZ pulses, the axis will stop instantly (or make a deceleration stop when feeding at high speed). - COUNTER reset timing: When finishing counting the EZ pulses.

Bit	Bit name	Description
		0110: Origin return operation 6 - After the EL input turns ON when feeding at constant speed, the axis will stop immediately (or make a deceleration when ELM is 1). Then, the axis will start feeding in the opposite direction at RFA constant speed. When the EL signal turns OFF, the axis will stop instantly when the LSI finishes counting the EZ pulses. - COUNTER reset timing: When the EL input is OFF. 0111: Origin return operation 7 - After the EL signal turns ON when feeding at constant speed, the axis will stop immediately (or make a deceleration when ELM is 1). Then, the axis will start feeding in the opposite direction at RFA constant speed. After the EL signal turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting EZ pulses, the axis will stop instantly. - COUNTER reset timing: When stopped by finishing counting the EL pulses. 1000:Origin return operation 8 - After the EL signal turns ON when feeding at constant speed, the axis will stop immediately (or make a deceleration when ELM is 1). Then, the axis will start feeding in the opposite direction at RFL constant speed. After the EL signal turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting EZ pulses, the axis will stop instantly. - CONTER reset timing: When finishing counting the EZ signal. 1001: Origin return operation 9 - After the process in origin return operation 0 has executed, it returns to zero (operates until COUNTER2 = 0). 1010: Origin return operation 10 - After the process in origin return operation 3 has executed, it returns to zero (operates until COUNTER2 = 0). 1011: Origin return operation 11 - After the process in origin return operation 5 has executed, it returns to zero (operates until COUNTER2 = 0). 1100: Origin return operation 12 - After the process in origin return operation 8 has executed, it returns to zero (operates until COUNTER2 = 0).
4 to 7	EZD0 to 3	Specify the EZ count value that is used for origin return operations. 0000 (1st count) to 1111 (16th count)
8 to 9	CI20 to 21	Select the input count source for COUNTER2 (mechanical position). 00: EA/EB input 01: Output pulse 10: PA/PB input
10 to 11	CI30 to 31	Select the input count source for COUNTER3 (deflection counter) 00: Output pulse and EA/EB input (deflection counter) 01: Output pulse and PA/PB input (deflection counter) 10: EA/EB input and PA/PB input (deflection counter)
12 to 13	CI40 to 41	Select the input count source for COUNTER4 (general-purpose) 00: Output pulse 01: EA/EB input 10: PA/PB input 11: Divide the CLK count by 2
14	BSYC	1: Operate COUNTER4 only while LSI is operating (#BST is low).
15	Not defined	(Always set to 0.)
16	CU1C	1: Reset COUNTER1 (command position) when the CLR input turns ON.
17	CU2C	1: Reset COUNTER2 (mechanical position) when the CLR input turns ON.
18	CU3C	1: Reset COUNTER3 (deflection counter) when the CLR input turns ON.
19	CU4C	1: Reset COUNTER4 (general-purpose) when the CLR input turns ON.
20	CU1R	1: Reset COUNTER1 (command position) when the origin return is complete.
21	CU2R	1: Reset COUNTER2 (mechanical position) when the origin return is complete.
22	CU3R	1: Reset COUNTER3 (deflection counter) when the origin return is complete.
23	CU4R	1: Reset COUNTER4 (general-purpose) when the origin return is complete.
24	CU1B	1: Operate COUNTER1 (command position) while in backlash/slip correction mode.

Bit	Bit name	Description
25	CU2B	1: Operate COUNTER2 (mechanical position) while in backlash/slip correction mode.
26	CU3B	1: Operate COUNTER3 (deflection counter) while in backlash/slip correction mode.
27	CU4B	1: Operate COUNTER4 (general-purpose) while in backlash/slip correction mode.
28	Not defined	(Always set to 0.)
29	CU2H	1: Stop the counting operation on COUNTER2 (mechanical position). Note 1.
30	CU3H	1: Stop the counting operation on COUNTER3 (deflection counter).
31	CU4H	1: Stop the counting operation on COUNTER4 (general-purpose).

Note 1: To stop the counting on COUNTER1 (command position), change MCCE (bit 11) in the RMD register.

8-3-16. RENV4 register

This register is used for Environment 4 settings. Set up comparators 1 to 4.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2RM	C2D1	C2D0	C2S2	C2S1	C2S0	C2C1	C2C0	C1RM	C1D1	C1D0	C1S2	C1S1	C1S0	C1C1	C1C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C4D1	C4D0	C4S3	C4S2	C4S1	C4S0	C4C1	C4C0	IDXM	C3D1	C3D0	C3S2	C3S1	C3S0	C3C1	C3C0

Bit	Bit name	Description
0 to 1	C1C0 to 1	Select a comparison counter for comparator 1. Note 1 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (deflection counter) 11: COUNTER4 (general-purpose)
2 to 4	C1S0 to 2	Select a comparison method for comparator 1. Note 2 001: RCMP1 data = Comparison counter (regardless of counting direction) 010: RCMP1 data = Comparison counter (while counting up (count forward)) 011: RCMP1 data = Comparison counter (while counting down) 100: RCMP1 data > Comparison counter data 101: RCMP1 data < Comparison counter data 110: Use as positive end software limit (RCMP1< COUNTER1) Others: Treats that the comparison conditions are not satisfied. Note 4
5 to 6	C1D0 to 1	Select a process to execute when the Comparator 1 conditions are met. 00: None (use as an #INT, terminal output, or internal synchronous start) 01: Immediate stop 10: Deceleration stop 11: Rewrite operation data with pre-register data (change speed)
7	C1RM	1: Use COUNTER1 for ring counter operation by using Comparator 1. See "11-11-5. Ring count function."
8 to 9	C2C0 to 1	Select a comparison counter for Comparator 2. Note 1. 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (deflection counter) 11: COUNTER4 (general purpose)
10 to 12	C2S0 to 2	Select a comparison method for Comparator 2. Note 2. 001: RCMP2 data = Comparison counter (regardless of counting direction) 010: RCMP2 data = Comparison counter (while counting up (count forward)) 011: RCMP2 data = Comparison counter (while counting down) 100: RCMP2 data > Comparison counter data 101: RCMP2 data < Comparison counter data 110: Use as negative end software limit (RCMP2>COUNTER1) Others: Treats that the comparison conditions do not meet. Note 4.
13 to 14	C2D0 to 1	Select a process to execute when the Comparator 2 conditions are met. 00: None (use as an #INT, terminal output, or internal synchronous start) 01: Immediate stop. 10: Deceleration stop. 11: Rewrite operation data with pre-register data (change speed)
15	C2RM	1: Use COUNTER2 for ring counter operation by using Comparator 2. See "11-11-5. Ring count function."
16 to 17	C3C0 to 1	Select a comparison counter for Comparator 3. Note 1 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (deflection counter) 11: COUNTER4 (general-purpose)

Bit	Bit name	Description
18 to 20	C3S0 to 2	Select a comparison method for comparator 3. Note 2 001: RCMP3 data = Comparison counter (regardless of counting direction) 010: RCMP3 data = Comparison counter (while counting up (count forward) 011: RCMP3 data = Comparison counter (while counting down) 100: RCMP3 data > Comparison counter data 101: RCMP3 data < Comparison counter data 110: Prohibited setting Others: Treats that the comparison conditions do not meet.
21 to 22	C3D0 to 1	Select a process to execute when the Comparator 3 conditions are met. 00: None (use as an #INT, terminal output, or internal synchronous start) 01: Immediate stop. 10: Deceleration stop. 11: Rewrite operation data with pre-register data (change speed)
23	IDXM	0: Outputs an IDX signal while COUNTER4 = RCMP2. 1: When COUNTER4 reaches 0 by counting, the PCL outputs an IDX signal of two CLK cycles width. (This is only possible when the values in C4S0 to C4S3 are 1000 to 1010.)
24 to 25	C4C0 to 1	Select a comparison counter for Comparator 4. Note 1. 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (deflection counter) 11: COUNTER4 (general purpose)
26 to 29	C4S0 to 3	Select a comparison method for Comparator 4. Note 3. 0001: RCMP4 data = Comparison counter (regardless of counting direction) 0010: RCMP4 data = Comparison counter (while counting up (count forward)) 0011: RCMP4 data = Comparison counter (while counting down) 0100: RCMP4 data > Comparison counter data 0101: RCMP4 data < Comparison counter data 0111: Treats that the comparison conditions are not satisfied. 1000: Use as IDX (synchronous) signal output (regardless of counting direction) 1001: Use as IDX (synchronous) signal output (while counting up (count forward)) 1010: Use as IDX (synchronous) signal output (while counting down) Others: Treats that the comparison conditions are not satisfied.
30 to 31	C4D0 to 1	Select a process to execute when the Comparator 4 conditions are satisfied. 00: None (use as an #INT, terminal output, or internal synchronous start) 01: Immediate stop. 10: Deceleration stop. 11: Rewrite operation data with pre-register data (change speed)

Note 1: When COUNTER3 (deflection counter) is selected as the comparison counter, the LSI compares the counted absolute value and the comparator data. (Absolute value range: 0 to 32,767.)

Note 2: When you specify C1S0 to 2 = 110 (positive software limit) or C2S0 to 2 = 110 (negative software limit), select COUNTER1 (command position) as the comparison counter.

Note3: When C4S0 to 3 is set to 1000 to 1010 (synchronous signal output), select COUNTER4 (genera-purpose) for the comparison counter. The other counters cannot be selected.
To set the comparator, select a positive value.

Note 4: When this bit is used as software limit, the PCL stops operation regardless of the settings for selecting a process when the conditions are satisfied. However, when the PCL is operating and "10: Deceleration stop" is selected, it only uses a deceleration stop when operating at high speed. In all other cases, it stops immediately.

8-3-17. RENV5 register

This is a register for the Environment 5 settings. Settings for Comparator 5 are its main use.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LTOF	LTFD	LTM1	LTM0	0	IDL2	IDL1	IDL0	C5D1	C5D0	C5S2	C5S1	C5S0	C5C2	C5C1	C5C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	CU4L	CU3L	CU2L	CU1L	ISMR	MSMR	SYI1	SYI0	SYO3	SYO2	SYO1	SYO0

Bit	Bit name	Description
0 to 2	C5C0 to 2	Select a comparison counter for comparator 5. 000: COUNTER1 (command position) 011: COUNTER4 (general-purpose) 001: COUNTER2 (mechanical position) 100: Positioning counter 010: COUNTER3 (deflection counter) 101: Current speed data
3 to 5	C5S0 to 2	Select a comparison method for comparator 5. 001: RCMP5 data = Comparison counter (regardless of counting direction) 010: RCMP5 data = Comparison counter (while counting up (count forward)) 011: RCMP5 data = Comparison counter (while counting down) 100: RCMP5 data > Comparison counter 101: RCMP5 data < Comparison counter Others: Treats that the comparison conditions are not met.
6 to 7	C5D0 to 1	Select a process to execute when the Comparator 5 conditions are satisfied. 00: None (use as an INT, terminal output, or internal synchronous start) 01: Immediate stop. 10: Deceleration stop. 11: Rewrite operation data with pre-register data (change speed)
8 to 10	IDL0 to 2	Enter the number of idling pulses. (0 to 7 pulses)
11	Not defined	(Always set to 0.)
12 to 13	LTM0 to 1	Specify the latch timing for a counter (COUNTER1 to 4). 00: When the LTC input turns ON. 01: On an ORG input 10: When the Comparator 4 conditions are satisfied. 11: When the Comparator 5 conditions are satisfied.
14	LTFD	1: Latch the current speed in place of COUNTER3.
15	LTOF	1: Stop the latch by timing of a hardware operation. (Only used by software.)
16 to 19	SYO0 to 3	Select the output timing of the internal synchronous signal. 0001: When the Comparator 1 conditions are satisfied. 0010: When the Comparator 2 conditions are satisfied. 0011: When the Comparator 3 conditions are satisfied. 0100: When the Comparator 4 conditions are satisfied. 0101: When the Comparator 5 conditions are satisfied. 1000: When starting acceleration. 1001: When ending acceleration. 1010: When starting deceleration. 1011: When ending deceleration. Others: Internal synchronous signal output is OFF.
20 to 21	SYI0 to 1	Select an input source when starting with an internal synchronous signal. 00: Internal synchronous signal output from the X axis. 01: Internal synchronous signal output from the Y axis. 10: Internal synchronous signal output from the Z axis. 11: Internal synchronous signal output from the U axis.
22	MSMR	1: Stop auto function to reset SENI and SEDR when main status is read out. To reset, use command SENIR and SEORR.

Bit	Bit name	Description
23	ISMR	1: Stop auto function to be reset when RIST register and REST register are read out. (To reset this bit, write to RIST and REST registers.)
24	CU1L	1: Resets COUNTER1 at the same time COUNTER1 is latched.
25	CU2L	1: Resets COUNTER2 at the same time COUNTER2 is latched.
26	CU3L	1: Resets COUNTER3 at the same time COUNTER3 is latched.
27	CU4L	1: Resets COUNTER4 at the same time COUNTER4 is latched.
28 to 31	Not defined	(Always set to 0.)

8-3-18. RENV6 register

This is a register for the Environment 6 settings. It is primarily used to set feed amount correction data.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSTP	0	ADJ1	ADJ0	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMG4	PMG3	PMG2	PMG1	PMG0	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

Bit	Bit name	Description
0 to 11	BR0 to 11	Enter a backlash correction amount or a slip correction amount. (0 to 4095)
12 to 13	ADJ0 to 1	Select a feed amount correction method. 00: Turn OFF the correction function. 01: Backlash correction 10: Slip correction
14	Not defined	(Always set to 0.)
15	PSTP	1: Even if a stop command is written, the PCL will operate for the number of pulses that are already input on PA/PB. Note 1.
16 to 26	PD0 to 10	Specifies the division ratio for pulses on the PA/PB input. Number of pulses = set value/2048. When 0 is entered, the division circuit will be OFF. (= 2048/2048)
27 to 31	PMG0 to 4	Specifies the magnification rate for pulses on the PA/PB input. Number of pulses=number of pulses input from PA/PB x (set value+1).

Note 1: When PSTP is 1, the Stop command will be ignored when #BSYn = H (OFF), regardless of the operation mode. Before writing a Stop command, check the main status register. When SRUN = 0, change PSTP to 0 and then write a Stop command.

8-3-19. RENV7 register

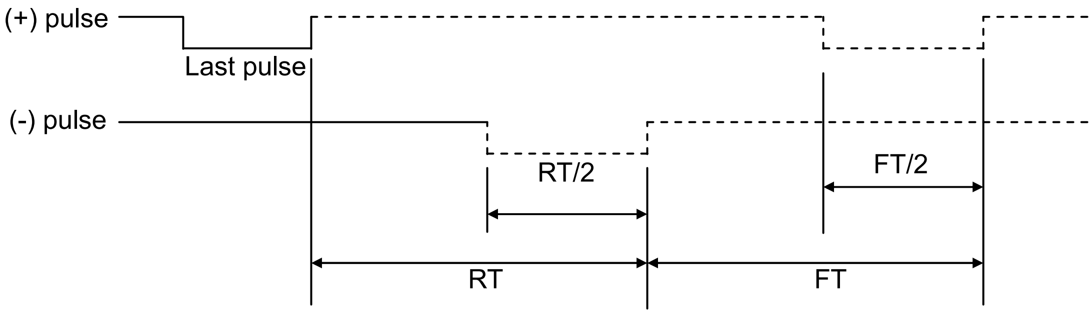
This is a register for the Environment 7 settings. It is primarily used to enter the time for the vibration reduction function. If both RT and FT data are other than zero, the vibration reduction function is turned ON.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0

Bit	Bit name	Description
0 to 15	RT0 to 15	Enter the RT time shown in the figure below. The units are 32 ticks of the reference clock (approx. 1.6 μsec). (0 to 65,535)
16 to 31	FT0 to 15	Enter the FT time shown in the figure below. The units are 32 ticks of the reference clock (approx. 1.6 μsec). (0 to 65,535)

The dotted lines in the figure below are pulses added by the vibration reduction function.



8-3-20. RCUN1 register

This is a register used for COUNTER1 (command position counter).
This is a counter used exclusively for command pulses.
Setting rage: -2,147,483,648 to +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

8-3-21. RCUN2 register

This is a register used for COUNTER2 (mechanical position counter).
It can count three types of pulses: Command pulses, encoder signals (EA/EB input), pulsar signals (PA/PB input).
Setting range: -2,147,483,648 to +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

8-3-22. RCUN3 register

This is a register used for COUNTER3 (deflection counter).
It can count three types of deflections: Between command pulses and encoder signals, between command pulses and pulsar signals, and between encoder signals and pulsar signals.
Setting range: -32,768 to +32,767.

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
&	&	&	&	&	&	&	&	&	&	&	&	&	&	&	&																

8-3-23. RCUN4 register

This is a register used for COUNTER4 (general-purpose counter).
It can count four types of signals: Command pulses, encoder signals (EA/EB input), pulsar signals (PA/PB input), and 1/2 ticks of the reference clock.
Setting range: -2,147,483,648 to +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 the counters, see section 11-10, "Counter."

8-3-24. RCMP1 register

Specify the comparison data for Comparator 1.
Setting range: -2,147,483,648 to +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

8-3-25. RCMP2 register

Specify the comparison data for Comparator 2.
Setting range: -2,147,483,648 to +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

Note 1: Bits marked with an "*" (asterisk) will be ignored when written and are 0 when read.
Note 2: Bits marked with an "&" symbol will be ignored when written and will be the same value as the upper most bit among bits having no marks when read. (Sign extension)

8-3-26. RCMP3 register

Specify the comparison data for Comparator 3.
Setting range: -2,147,483,648 to +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

8-3-27. RCMP4 register

Specify the comparison data for Comparator 4.
Setting range: -2,147,483,648 to +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

8-3-28. RCMP5 (PRCP5) register

Specify the comparison data for Comparator 5.
PRCP5 is the 2nd pre-register for RCMP5.
Normally, use RCMP5. To use the comparator pre-register function, use PRCP5.
Setting range: -2,147,483,648 to +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 the comparators, see section 11-11, "Comparator."

Note 1: Bits marked with an "*" (asterisk) will be ignored when written and are 0 when read.
Note 2: Bits marked with an "&" symbol will be ignored when written and will be the same value as the upper most bit among bits having no marks when read. (Sign extension)

8-3-29. RIRQ register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IROL	IRLT	IRCL	IRC5	IRC4	IRC3	IRC2	IRC1	IRDE	IRDS	IRUE	IRUS	IRND	IRNM	IRN	IREN

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	IRSA	IRDR	IRSD

Bit	Bit name	Description
0	IREN	When stopping normally.
1	IRN	When starting the next operation continuously.
2	IRNM	When writing to the 2nd pre-register.
3	IRND	When writing to the 2nd pre-register for Comparator 5.
4	IRUS	When starting acceleration.
5	IRUE	When ending acceleration.
6	IRDS	When starting deceleration.
7	IRDE	When ending deceleration.
8	IRC1	When Comparator 1 conditions are satisfied.
9	IRC2	When Comparator 2 conditions are satisfied.
10	IRC3	When Comparator 3 conditions are satisfied.
11	IRC4	When Comparator 4 conditions are satisfied.
12	IRC5	When Comparator 5 conditions are satisfied.
13	IRCL	When resetting the count value with a CLR signal input.
14	IRLT	When latching the count value with an LTC signal input.
15	IROL	When latching the count value with an ORG signal input.
16	IRSD	When the SD input is ON.
17	IRDR	When the ±DR input changes.
18	IRSA	When the #CSTA input is ON.
19 to 31	Not defined	(Always set to 0.)

8-3-30. RLTC1 register

Latched data for COUNTER1 (command position). (Read only.)
The contents of COUNTER1 are copied when triggered by the LTC, an ORG input, or an LTCH command.
Data range: -2,147,483,648 to +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

8-3-31. RLTC2 register

Latched data for COUNTER2 (mechanical position). (Read only.)
The contents of COUNTER2 are copied when triggered by the LTC, an ORG input, or an LTCH command.
Data range: -2,147,483,648 to +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

8-3-32. RLTC3 register

Latched data for COUNTER3 (deflection counter) or current speed. (Read only.)
The contents of COUNTER3 or the current speed are copied when triggered by the LTC, an ORG input, or an LTCH command. When the RENV5.LTFD is 0, the register latches the COUNTER3 data. When the LTFD is 1, the register latches the current speed. When the LTFD is 1 and movement on the axis is stopped, the latched data will be 0.
Data range when LTFD is 0: -32,768 to +32,767.
Data range when LTFD is 1: 0 to 65,535.

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 "$" will be the same as bit 15 when LTFD (bit 14) in the RENV5 register is 0 (sign extension), and they will be 0 when the LTFD is 1.

8-3-33. RLTC4 register

Latched data for COUNTER4 (general-purpose). (Read only.)
The contents of COUNTER4 are copied when triggered by the LTC, an ORG input, or an LTCH command.
Data range: -2,147,483,648 to +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 the counter data latch, see section 11-10, "Counter."

Note 1: Bits marked with an "*" (asterisk) will be ignored when written and are 0 when read.
Note 2: Bits marked with an "&" symbol will be ignored when written and will be the same value as the upper most bit among bits having no marks when read. (Sign extension)

8-3-34. 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
PSDI	SLTC	SCLR	SDRM	SDRP	SEZ	SERC	SPCS	SEMG	SSTP	SSTA	SDIR	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	MSDL	PSDL	PFM1	PFM0	PFC1	PFC0	MSDI	SINP

Bit	Bit name	Description	
0 to 3	CND0 to 3	Reports the operation status. 0000: Under stopped condition 0001: Waiting for DR input 0010: Waiting for #CSTA input 0011: Waiting for an internal synchronous signal 0100: Waiting for another axis to stop. 0101: Waiting for a completion of ERC timer 0110: Waiting for a completion of direction change timer 0111: Correcting backlash	1000: Waiting for PA/PB input 1001: Feeding at FA constant speed. 1010: Feeding at FL constant speed. 1011: Accelerating 1100: Feeding at FH constant speed. 1101: Decelerating 1110: Waiting for INP input. 1111: Others (controlling start)
4	SDIR	Operation direction (0: Positive direction. 1: Negative direction.)	
5	SSTA	Becomes 1 when the #CSTA input signal is turned ON.	
6	SSTP	Becomes 1 when the #CSTP input signal is turned ON.	
7	SEMG	Becomes 1 when the #CEMG input signal is turned ON.	
8	SPCS	Becomes 1 when the PCS input signal is turned ON.	
9	SERC	Becomes 1 when the ERC input signal is turned ON.	
10	SEZ	Becomes 1 when the EZ input signal is turned ON.	
11	SDRP	Becomes 1 when the +DR input signal is turned ON.	
12	SDRM	Becomes 1 when the -DR input signal is turned ON.	
13	SCLR	Becomes 1 when the CLR input signal is turned ON.	
14	SLTC	Becomes 1 when the LTC input signal is turned ON.	
15	PSDI	Becomes 1 when the +SD input signal is turned ON. (Status of +SD input terminal.)	
16	SINP	Becomes 1 when the INP input signal is turned ON.	
17	MSDI	Becomes 1 when the -SD input signal is turned ON. (Status of -SD input terminal.)	
18 to 19	PFC0 to 1	Used to monitor the condition of the RCMP5 pre-register.	
20 to 21	PFM0 to 1	Used to monitor the condition of the operation pre-registers (for other than RCMP5).	
22	PSDL	Becomes 1 when the +SD latch signal is turned ON.	
23	MSDL	Becomes 1 when the -SD latch signal is turned ON.	
24 to 31	Not defined	(Always set to 0.)	

8-3-35. REST register

Used to check the error interrupt cause. (Read only.)
The corresponding bit will be "1" when an error interrupt occurs.
This register is reset by the following procedure.
However, When RENV5.ISMR (bit 24) =1, this register is not reset. It is reset by writing data to REST.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESAO	ESPO	ESIP	ESDT	0	ESSD	ESEM	ESSP	ESAL	ESML	ESPL	ESC5	ESC4	ESC3	ESC2	ESC1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ESPE	ESEE

Bit	Bit name	Description
0	ESC1	Stopped when Comparator 1 conditions are satisfied. (+SL)
1	ESC2	Stopped when Comparator 2 conditions are satisfied. (-SL)
2	ESC3	Stopped when Comparator 3 conditions are satisfied.
3	ESC4	Stopped when Comparator 4 conditions are satisfied.
4	ESC5	Stopped when Comparator 5 conditions are satisfied.
5	ESPL	Stopped by the +EL input being turned ON.
6	ESML	Stopped by the -EL input being turned ON.
7	ESAL	Stopped by the ALM input being turned ON.
8	ESSP	Stopped by the #CSTP input being turned ON.
9	ESEM	Stopped by the #CEMG input being turned ON.
10	ESSD	Decelerated and stopped by the SD input being turned ON.
11	Not defined	(Always set to 0.)
12	ESDT	Stopped by an interpolation operation data error. (Note 1)
13	ESIP	Simultaneously stopped with another axis due to an error stop on the other axis during interpolation.
14	ESPO	Stopped when an overflow occurs in the PA/PB input buffer counter.
15	ESAO	Stopped when the positioning counter counts beyond the range during interpolation.
16	ESEE	An EA/EB input error occurs. (The driving is not stopped.)
17	ESPE	A PA/PB input error occurs. (The driving is not stopped.)
18 to 31	Not defined	(Always set to 0.)

Note 1: In any of the following cases, ESDT will be 1.

1) Write a Start command using linear interpolation 1 mode (MOD = 60h, 61h, 68h, and 69h) on only one axis.
2) Write a Start command using circular interpolation mode (MOD = 64h, 65h, 66h, 67h, 6Ch, and 6Dh) on only one axis.
3) Write a Start command after setting PRIP (circular center coordinates) to (0, 0) using the circular interpolation mode.
4) Write a Start command using circular interpolation mode on 3 or 4 axes.
5) Write a Start command using linear interpolation 2 mode (MOD = 62h, 63h, 6Ah, and 6Bh) while RIP is 0.
6) Tried to write a Start command using circular interpolation mode (MOD = 66h, 67h) while synchronized with the U axis. But the U axis does not respond. Or, the U axis completes operation while in circular interpolation mode.

8-3-36. RIST register

This register is used to check event interrupt cause. (Read only.)
When an event interrupt occurs, the bit corresponding to the cause will be set to 1.
This register is reset by the following procedure.
However, When RENV5.ISMR (bit 24) =1, this register is not reset. It is reset by writing data to REST.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISOL	ISLT	ISCL	ISC5	ISC4	ISC3	ISC2	ISC1	ISDE	ISDS	ISUE	ISUS	ISND	ISNM	ISN	ISEN

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	ISSA	ISMD	ISPD	ISSD

Bit	Bit name	Description
0	ISEN	When stopping automatically.
1	ISN	When the next operation starts continuously.
2	ISNM	When it is available to write operation to the 2nd pre-register.
3	ISND	When it is available to write operation to the 2nd pre-register for Comparator 5.
4	ISUS	When starting acceleration.
5	ISUE	When ending acceleration.
6	ISDS	When starting deceleration.
7	ISDE	When ending deceleration.
8	ISC1	When the comparator 1 conditions are satisfied.
9	ISC2	When the comparator 2 conditions are satisfied.
10	ISC3	When the comparator 3 conditions are satisfied.
11	ISC4	When the comparator 4 conditions are satisfied.
12	ISC5	When the comparator 5 conditions are satisfied.
13	ISCL	When the count value is reset by a CLR signal input.
14	ISLT	When the count value is latched by an LTC input.
15	ISOL	When the count value is latched by an ORG input.
16	ISSD	When the SD input turns ON.
17	ISPD	When the +DR input changes.
18	ISMD	When the -DR input changes.
19	ISSA	When the #CSTA input turns ON.
20 to 31	Not defined	(Always set to 0.)

8-3-37. RPLS register

This register is used to check the value of the positioning counter (number of pulses left for feeding). (Read only.)
At the start, this value will be the absolute value in the RMV register. A value in a register decreases for each pulse output. Read data range: 0 to 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

8-3-38. RSPD register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AS15	AS14	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	IDC2	IDC1	IDC0	ECZ3	ECZ2	ECZ1	ECZ0

Bit	Bit name	Description
0 to 15	AS0 to 15	Read current speed as a step value (same units as for RFL and RFH). When stopped the value is 0. (0 to 65,535)
16 to 19	ECZ0 to 3	Read a count value of EZ input that is used for an origin return. (0 to 15)
20 to 22	IDC0 to 2	Read an idling count value. (0 to 7)
23 to 31	Not defined	(Always set to 0.)

8-3-39. RSDC register

This register is used to check the automatically calculated ramping-down point value for the positioning operation. (Read only.)
Read data range: 0 to 16,777,215.

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																								

8-3-40. PRCI (RCI) register

This is a pre-register used to set circular interpolation stepping number.
RCI is the register for the PRCI.
These registers only exist for the X, Y, and Z axes. They do not exist for the U axis because the U axis is not available for circular interpolation control.

To decelerate during a circular interpolation, enter the number of steps (number of pulses calculated by the formula) required for the circular interpolation. Entering a number other than 0 can decelerate the speed by using an automatic ramping-down point.
Setting range: 0 to 4,294,967,295

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

8-3-41. RCIC register

This register is used to read the count of the number of circular interpolation steps that have been completed. (Read only.)
The RCI register value is loaded when a circular interpolation is started. This value is decreased by one for each circular interpolation step. However, if the counter value is 0, the PCL will not decrease it further.
The counter value at the completion of a circular interpolation is held in the PCL memory until the start of the next circular interpolation operation. The range for this value is 0 to 4,294,967,295.
This register is shared by all axes, and the value is same when read from any axis.

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

8-3-42. RIPS register

This register is used to check the interpolation setting status and the operation status. (Read only.)
This register is shared by all axes, and the value is same when read from any axis.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPFu	IPFz	IPFy	IPFx	IPSu	IPUz	IPSy	IPSx	IPEu	IPEz	IPEy	IPEx	IPLz	IPLy	IPLy	IPLx

●

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	SED1	SED0	SDM1	SDM0	IPCC	IPCW	IPE	IPL

Bit	Bit name	Description
0	IPLx	1: X axis is in linear interpolation 1 mode.
1	IPLy	1: Y axis is in linear interpolation 1 mode.
2	IPLz	1: Z axis is in linear interpolation 1 mode.
3	IPLu	1: U axis is in linear interpolation 1 mode.
4	IPEx	1: X axis is in linear interpolation 2 mode.
5	IPEy	1: Y axis is in linear interpolation 2 mode.
6	IPEz	1: Z axis is in linear interpolation 2 mode.
7	IPEu	1: U axis is in linear interpolation 2 mode.
8	IPSx	1: X axis is in circular interpolation mode.
9	IPSy	1: Y axis is in circular interpolation mode.
10	IPSz	1: Z axis is in circular interpolation mode.
11	IPSu	1: U axis is in circular interpolation mode.
12	IPFx	1: X axis is specified for constant synthesized speed.
13	IPFy	1: Y axis is specified for constant synthesized speed.
14	IPFz	1: Z axis is specified for constant synthesized speed.
15	IPFu	1: U axis is specified for constant synthesized speed.
16	IPL	1: Executing linear interpolation 1.
17	IPE	1: Executing linear interpolation 2.
18	IPCW	1: Executing a CW directional circular interpolation.
19	IPCC	1: Executing a CCW directional circular interpolation.
20 to 21	SDM0 to 1	Current quadrant of a circular interpolation (00: 1st quadrant, 01: 2nd quadrant, 10: 3rd quadrant, 11: 4th quadrant)
22 to 23	SED0 to 1	Final phase in a circular interpolation (00: 1st quadrant, 01: 2nd quadrant, 10: 3rd quadrant, 11: 4th quadrant)
24 to 31	Not defined	(Always set to 0.)

9. Operation Mode

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

9-1. Continuous operation mode using command control

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

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

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

9-2. Positioning operation mode

The following seven operation types are available for positioning operations.

MOD	Operation method	Direction of movement
41h	Positioning operation (specify target increment position)	Positive direction when PRMV ≥ 0 Negative direction when PRMV < 0
42h	Positioning operation (specify the absolute position in COUNTER1)	Positive direction when PRMV ≥ COUNTER1 Negative direction when PRMV < COUNTER1
43h	Positioning operation (specify the absolute position in COUNTER2)	Positive direction when PRMV ≥ COUNTER2 Negative direction when PRMV < COUNTER2
44h	Return to command position 0 (COUNTER1)	Positive direction when COUNTER1 ≤ 0 Negative direction when COUNTER1 > 0
45h	Return to machine position 0 (COUNTER2)	Positive direction when COUNTER2 ≤ 0 Negative direction when COUNTER2 > 0
46h	One pulse operation	Positive direction
4Eh	One pulse operation	Negative direction
47h	Timer operation	

9-2-1. Positioning operation (specify a target position using an incremental value) (PRMD.MOD: 41h)

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

9-2-2. Positioning operation (specify the absolute position in COUNTER1) (PRMD.MOD: 42h)

This mode only uses the difference between the PRMV (target position) register value and COUNTER1.
Since the COUNTER1 value is stored when starting to move, the target position cannot be overridden by changing the COUNTER1 value although it can be overridden by changing the RMV value.
The direction of movement can be set automatically by evaluating the relative relationship between the PRMV register setting and the value in COUNTER1.
At the start, the difference between the RMV setting and the value stored in COUNTER1 is loaded into the positioning counter (RPLS). The PCL outputs pulses so that the difference becomes 0. When the positioning counter value reaches zero, the PCL stops outputting pulses.
If the PRMV register value is made equal to the COUNTER1 value and the positioning operation is started, the PCL will immediately stop operation without outputting any command pulses.

9-2-3. Positioning operation (specify the absolute position in COUNTER2) (PRMD.MOD: 43h)

This mode only uses the difference between the PRMV (target position) register setting and the value in COUNTER2.
Since the COUNTER2 value is stored when starting a positioning operation, the target position cannot be overridden by changing the value in COUNTER2, although it can override the target position by changing the value in RMV.
The direction of movement can be set automatically by evaluating the relationship between the PRMV register setting and the value in COUNTER2.
At the start, the difference between the RMV setting and the value stored in COUNTER2 is loaded into the positioning counter (RPLS). The PCL outputs pulses so that the difference becomes 0. When the positioning counter value reaches zero, the PCL stops outputting pulses.
If the PRMV register value is made equal to the COUNTER2 value and the positioning operation is started, the PCL will immediately stop operation without outputting any command pulses.

9-2-4. Command position 0 return operation (PRMD.MOD: 44h)

This mode is used to continue operation until the COUNTER1 (command position) value becomes zero.
The direction of movement is set automatically by the sign for the value in COUNTER1 when starting.
This operation is the same as when positioning (specify the absolute position in COUNTER1) by entering zero in the PRMV register; however, there is no need to specify the PRMV register.

9-2-5. Mechanical position 0 return operation (PRMD.MOD: 45h)

This mode is used to continue operations until the value in COUNTER2 (mechanical position) becomes zero.
The number of output pulses and feed direction are set automatically by internal calculations based on the COUNTER2 value when starting.
This operation is the same as when positioning (specify the absolute position in COUNTER2) by entering zero in the PRMV register. However, there is no need to specify the PRMV register.

9-2-6. One pulse operation (PRMD.MOD: 46h, 4Eh)

In this mode, a single pulse is output.
This operation is identical to a positioning operation (incremental target positioning) that writes a "1" (or "-1") to the PRMV register. However, with this operation, you do need not to write a "1" or "-1" to the PRMV register.

9-2-7. Timer operation (PRMD.MOD: 47h)

This mode allows the internal operation time to be used as a timer.
The internal effect of this operation is identical to the positioning operation. However, the LSI does not output any pulses.
Therefore, the internal operation time using the constant speed start command will be a product of the frequency of the output pulses and the RMV register setting. (Ex.: When the frequency is 1000 pps and the RMV register is set to 120 pulses, the internal operation time will be 120 msec.)
Write a positive number (1 to 2,147,483,647) into the PRMV register.
The ±EL input signal, ±SD input signal, and software limits are ignored. (These are always treated as OFF.)
The ALM input signal, #CSTP input signal, and #CEMG input signals are effective.
The backlash/slip correction, vibration restriction function, and when changing direction, this timer function is disabled.
The LSI stops counting from COUNTER1 (command position).
Regardless of the PRMD.MINP (bit 9) 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 (bit 12) to zero and use the cycle completion timing of the output pulse as the operation complete timing.

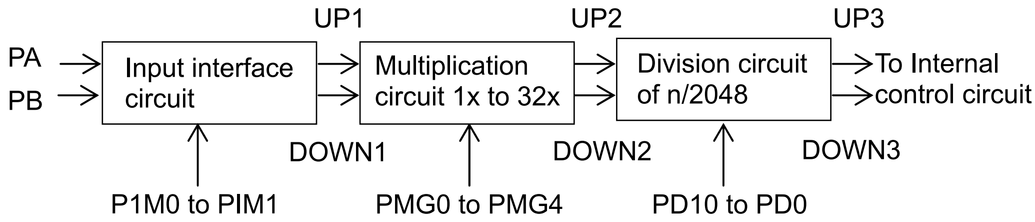
9-3. Pulsar (PA/PB) input mode

This mode is used to allow operations from a pulsar input.
In order to enable pulsar input, bring the #PE terminal LOW. Set POFF in the RENV2 register to zero.
It is also possible to apply a filter on the #PE input.
After writing a start command, when a pulsar signal is input, the LSI will output pulses to the OUT terminal.
Use an FL constant speed start (STAFL: 50h) or an FH constant speed start (STAFH: 51h).
Four methods are available for inputting pulsar signals through the PA/PB input terminal by setting the RENV2 (environmental setting 2) register.

- Supply a 90˚ phase difference signal (1x, 2x, or 4x).
- Supply either count-up (count-forward) or count-down pulses (Two-pulse input).

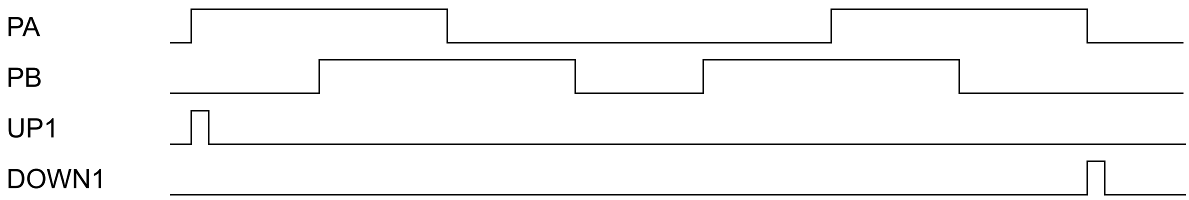
Note: The backlash correction function is available with the pulsar input mode. However, reversing pulsar input while in the backlash correction is unavailable.

Besides the above 1x to 4x multiplication, the PCL has a multiplication circuit of 1x to 32x and division circuit of (1 to 2048)/2048. For setting the multiplication from 1x to 32x, specify RENV6.PMG0 to 4 and for setting the division of n/2048, specify RENV6.PD0 to 10.

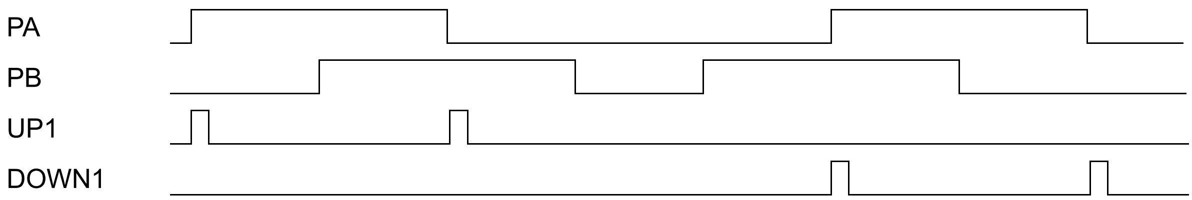


The timing of the UP1 and DOWN1 signals will be as follows by setting of the PIM0 to PIM1 in the RENV2.

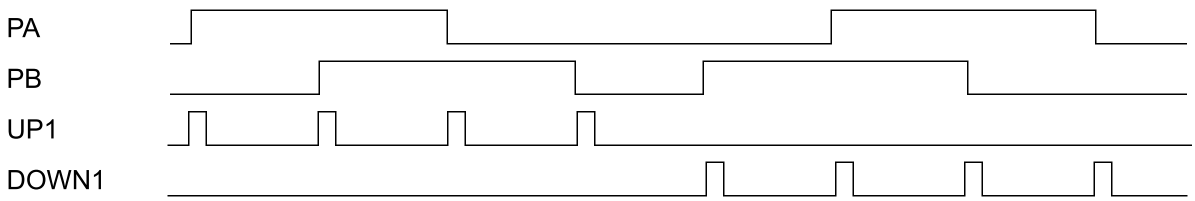
1) When using 90˚ phase difference signals and 1x input (RENV2.PIM = 00)



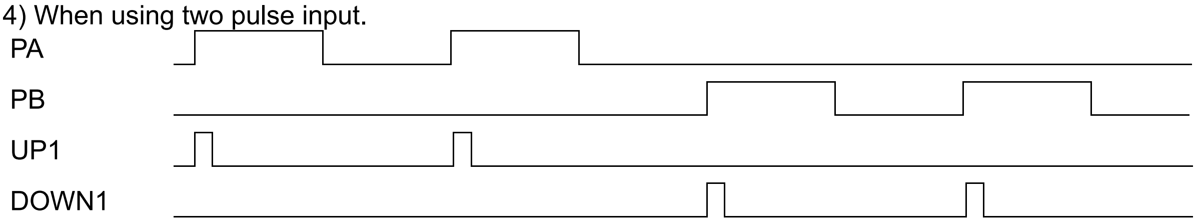
2) When using 90˚ phase difference signals and 2x input (RENV2.PIM = 01)



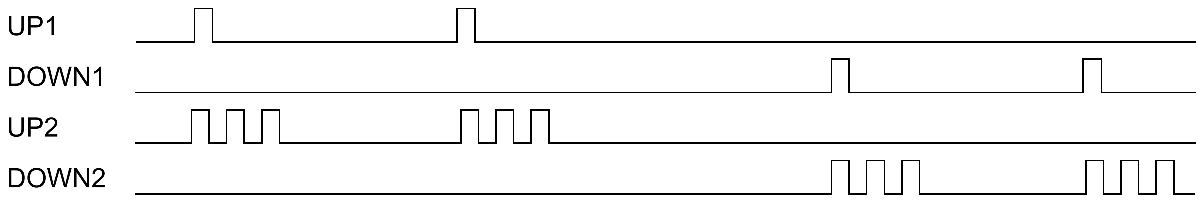
3) When using 90˚ phase difference signals and 4x input (RENV2.PIM = 10)



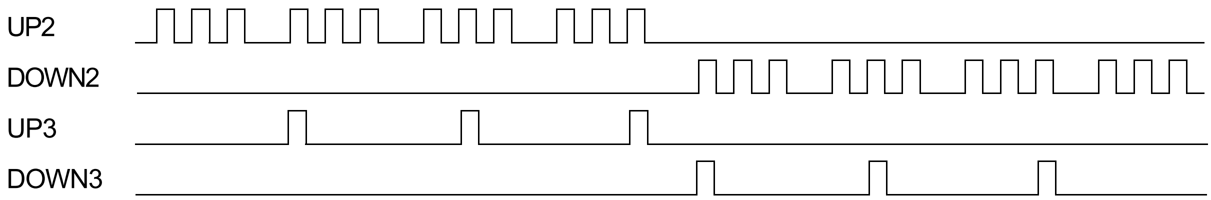
4) When using two pulse input.



When the 1x to 32x multiplication circuit is set to 3x (RENV6.PMG = 2), operation timing will be as follows.



When the n/2048 division circuit is set to 512/2048 (RENV6.PD =512), operation timing will be as follows.



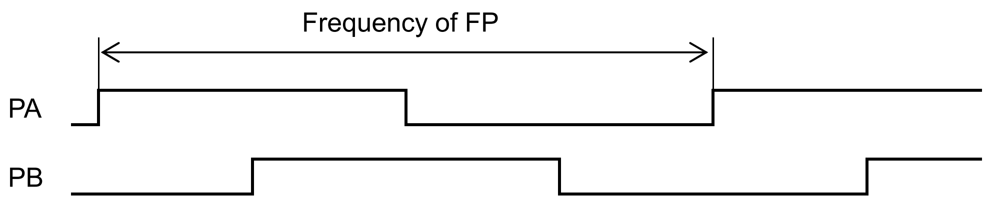
The pulsar input mode is triggered by an FL constant speed start command (50h) or by an FH constant speed start command (51h).
Pulsar input causes the PCL to output pulses with some pulses from the FL speed or FH speed pulse outputs being omitted. Therefore, there may be a difference in the timing between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

The maximum input frequency for pulsar signals 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 outputs #INT signals as errors when both the PA and PB inputs change simultaneously, or when the input frequency is exceeded and the input/output buffer counter (deflection adjustment 16-bit counter for pulsar input and output pulse) overflows. This can be monitored by the REST (error interrupt factor) register.

FP < (speed) / (input I/F phase value) / (PMG setting value + 1) / (PD setting value /2048), PD setting value ≠ 0
FP < (speed) / (input I/F phase value) / (PMG setting value + 1), PD setting value = 0

<Examples of the relationship between the FH (FL) speed [pps] and the pulsar input frequency FP [pps]>

PA/PB input I/F settingmethod	PMG setting value	PD setting value	Usable range
Two-pulse input	0 (1x)	0	FP < FH (FL)
	0 (1x)	1024	FP < FH (FL) x 2
	2 (3x)	0	FP < FH (FL) / 3
90˚ phase difference 1x	0 (1x)	0	FP < FH (FL)
	0 (1x)	1024	FP < FH (FL) x 2
	2 (3x)	0	FP < FH (FL) / 3
90˚ phase difference 2x	0 (1x)	0	FP < FH (FL) / 2
	0 (1x)	1024	FP < FH (FL)
	2 (3x)	0	FP < FH (FL) / 6
90˚ phase difference 4x	0 (1x)	0	FP < FH (FL) / 4
	0 (1x)	1024	FP < FH (FL) / 2
	2 (3x)	0	FP < FH (FL) / 6



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

<Setting relationship of PA/PB input>

Specify the PA/PB input <Set to RENV2.PIM0 to 1 (bit 24 to 25)> 00: 90˚ phase difference, 1x 10: 90˚ phase difference, 4x 01: 90˚ phase difference, 2x 11: 2 sets of up or down input pulses	[RENV2] (WRITE) 31 24 - - - - - - n n
Specify the PA/PB input count direction <Set to RENV2.PDIR (bit 26)> 0: Count up (count forward) when the PA phase is leading. Or, count up (count forward) on the rising edge of PA. 1: Count up (count forward) when the PB phase is leading. Or, count up (count forward) on the rising edge of PB.	[RENV2] (WRITE) 31 24 - - - - - n - -
Enable/disable PA/PB input <Set RENV2.POFF (bit 31)> 0: Enable PA/PB input 1: Disable PA/PB input.	[RENV2] (WRITE) 31 24 n - - - - - - -
Set the +/- DR, #PE input filter <Set RENV1.DRF (bit 27)> 1: Insert a filter on +/- DR input and #PE input By setting the filter, the PCL ignores signals shorter than 32 msec.	[RENV1] (WRITE) 31 24 - - - - n - - -
Reading operation status <RSTS.CND (bit 0 to 3)> 1000: wait for PA/ PB input.	[RSTS] (READ) 7 0 - - - - n n n n
Reading PA/PB input error <REST.ESPE (bit 17)> ESPE (bit 17) = 1: A PA/PB input error occurs.	[REST] (READ) 23 16 0 0 0 0 0 0 n -
Reading PA/PB input buffer counter status <REST.ESP0 (bit 14)> ESPO (bit 14) = 1: An overflow occurs.	[REST] (READ) 15 8 - n - - - - - -

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

The pulsar input mode has the following 12 operation types.
The direction of movement for continuous operation can be changed by setting the RENV2 register, without changing the wiring connections for the PA/PB inputs.

MOD	Operation mode	Direction of movement
01h	Continuous operation using pulsar input	Determined by the PA/PB input.
51h	Positioning operation using pulsar input (absolute position)	Determined by the sign of the PRMV value.
52h	Positioning operation using pulsar input (COUNTER1 absolute position)	Determined by the relationship of the RMV and COUNTER1 values.
53h	Positioning operation using pulsar input (COUNTER2 absolute position)	Determined by the relationship of the RMV and COUNTER2 values.
54h	Specified position (COUNTER1) zero point return operation using pulsar input	Determined by the sign of the value in COUNTER1.
55h	Specified position (COUNTER2) zero point return operation using pulsar input	Determined by the sign of the value in COUNTER2.
68h	Continuous linear interpolation 1 using pulsar input	Determined by the sign of the value in PRMV.
69h	Linear interpolation 1 using pulsar input	Determined by the sign of the value in PRMV.
6Ah	Continuous linear interpolation 2 using pulsar input	Determined by the sign of the value in PRMV.
6Bh	Linear interpolation 2 using pulsar input	Determined by the sign of the value in PRMV.
6Ch	CW circular interpolation using pulsar input	Determined by the circular interpolation operation
6Dh	CCW circular interpolation using pulsar input	Determined by the circular interpolation operation

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

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

PA/PB input method	PDIR	Feed direction	PA/PB input
90° phase difference signal (1x, 2x, and 4x)	0	Positive direction	When the PA phase leads the PB phase.
		Negative direction	When the PB phase leads the PA phase.
	1	Positive direction	When the PB phase leads the PA phase.
		Negative direction	When the PA phase leads the PB phase.
Two-pulse input of count-up (count forward) or count -down pulses	0	Positive direction	PA input rising edge.
		Negative direction	PB input rising edge.
	1	Positive direction	PB input rising edge.
		Negative direction	PA input rising edge.

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

Note: When the "immediate stop command (49h)" is written while the PCL is performing a multiplication operation (caused by setting RENV2.PIM 0 to 1 and RENV6.PMG 0 to 4), the PCL will stop operation immediately and the total number of pulses that are output will not always be an integral multiple of the magnification. When RENV6.PSTP is set to 1, the PCL delays the stop timing until an integral multiple of pulses has been output. However, after a stop command is sent by setting PSTP to 1, check the MSTSW. If SRUN is 0, set PSTP to 0. (When SRUN is 0 while RENV6.PSTP is 1, the PCL will latch the stop command.)

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

The PCL positioning is synchronized with the pulsar input by using the PRMV setting as incremental position data.
This mode allows positioning using a pulsar input.
The feed direction is determined by the sign in the PPMV register.
At the start, the content in the PRMV register is loaded to the positioning counter.
When PA/PB signals are input, the PCL outputs pulses and decrements the positioning counter. When the value in the positioning counter reaches zero, movement on the axis will stop and the PCL any further ignores PA/PB input. If you set the PRMV register value to zero and start the positioning operation, the PCL will stop movement on the axis immediately without outputting any command pulses.

9-3-3. Positioning operation using pulsar input (specify absolute position to COUNTER1) (PRMD.MOD: 52h)

The PCL positioning is synchronized with the pulsar input by using the PRMV setting as the absolute value for COUNTER1.
The direction of movement is determined by the magnitude relationship between the value in PRMV and the value in COUNTER1.
At the start, the difference between the values in RMV and COUNTER1 is loaded into the positioning counter.
When PA/PB signals are input, the PCL outputs pulses and decrements the positioning counter.
When the value in the positioning counter reaches "0," movement on the axis will stop and PCL any further ignores PA/PB input. If you try to start with PRMV = COUNTER1, the PCL will not output any pulses and it will stop immediately.

9-3-4. Positioning operation using pulsar input (specify the absolute position in COUNTER2) (PRMD.MOD: 53h)

The operation procedures are the same as MOD= 52h, except that this function uses COUNTER2 instead of COUNTER1.

9-3-5. Command position zero return operation using a pulsar input (PRMD.MOD: 54h)

This mode is used to feed the axis using a pulsar input until the value in COUNTER1 (command position) becomes zero. The number of pulses output and the feed direction are set automatically by internal calculation, using the COUNTER1 value at the start.
When setting the COUNTER1 value to zero and start the positioning operation, the LSI will stop movement on the axis immediately, without outputting any command pulses.

9-3-6. Mechanical position zero return operation using pulsar input (PRMD.MOD: 55h)

Except for using COUNTER2 instead of COUNTER1, the operation details are the same as for PRMD.MOD = 54h.

9-3-7. Continuous linear interpolation 1 using pulsar input (PRMD.MOD: 68h)

Performs continuous linear interpolation 1, synchronized with the pulsar input.
For continuous linear interpolation 1 operation details, see section "9-8. Interpolation operations".

9-3-8. Linear interpolation 1 using pulsar input (PRMD,MOD: 69h)

Performs linear interpolation 1, synchronized with the pulsar input.
Any pulsar inputs after operation is complete will be ignored.
For linear interpolation 1 operation details, see section "9-8. Interpolation operations."

9-3-9. Continuous linear interpolation 2 using pulsar input (PRMD.MOD: 6Ah)

Performs continuous linear interpolation 2, synchronized with the pulsar input.
For continuous linear interpolation 2 operation details, see section "9-8. Interpolation operations".

9-3-10. Linear interpolation 2 using pulsar input (PRMD.MOD: 6Bh)

Performs linear interpolation 2, synchronized with the pulsar input.
Any pulsar inputs after operation is complete will be ignored.
For linear interpolation 2 operation details, see section "9-8. Interpolation operations."

9-3-11. CW circular interpolation using pulsar input (PRMD.MOD: 6Ch)

Performs CW circular interpolation, synchronized with the pulsar input.
Any pulsar inputs after operation is complete will be ignored.
For CW circular interpolation operation details, see section "9-8. Interpolation operations."

9-3-12. CCW circular interpolation using pulsar input (PRMD.MOD: 6Dh)

Performs CCW circular interpolation, synchronized with the pulsar input.
Any pulsar inputs after operation is complete will be ignored.
For CCW circular interpolation operation details, see section "9-8. Interpolation operations."

9-4. External switch (±DR) operation mode

This mode allows operations with inputs from an external switch.
To enable inputs from an external switch, bring the #PE terminal LOW.
After writing a start command, when a +DR/-DR signal is input, the LSI will output pulses to the OUT terminal.
Set the RENV1.DRL to specify the output logic of the ±DR input signal. The #INT signal can be set to send an output when ±DR input is changed.
The RSTS (extension status) register can be used to check the operating status and monitor the ±DR input.
It is also possible to apply a filter to the ±DR or #PE inputs.

Set the input logic of the +DR/-DR signals <Set RENV1.DRL (bit 25)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 31 24 - - - - - - n -
Applying a ±DR or #PE input filter <Set RENV1.DRF (bit 27)> 1: Apply a filter to ±DR input or #PE inputs When a filter is applied, pulses shorter than 32 msec will be ignored.	[RENV1] (WRITE) 31 24 - - - - n - - -
Setting an event interrupt cause <Set RIRQ.IRDR (bit 17)> 1: Output the #INT signal when ±DR signal input changes.	[RIRQ] (WRITE) 23 16 0 0 0 0 0 - n -
Reading the event interrupt cause <RIST.ISPD (bit 17) and RIST.ISMD (bit 18)> ISPD(bit 17) = 1: When the +DR signal input changes. ISMD(bit 18) = 1: When the -DR signal input changes.	[RIST] (READ) 23 16 0 0 0 0 - n n -
Read operation status < RSTS.CND (bits 0 to 3)> 0001: Waiting for a DR input	[RSTS] (READ) 7 0 - - - - n n n n
Reading the ±DR signal <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 forms

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 (PRMD.MOD: 02h)

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

[Setting example]
1) Bring the #PE input LOW.
2) Specify PRFL, PRFH, PRUR, PRDR, and PRMG (speed setting).
3) Enter "0000010" for MOD (bits 0 to 6) in the PRMD (operation mode) register
4) Write a start command (50h to 53h).
CND (bits 0 to 3) of the RSTS (extension status) register will wait for "0001: DR input."
In this condition, turn ON the +DR or -DR input terminal. The axis will move in the specified direction using the specified speed pattern as long as the terminal is kept ON.

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

This mode is used for positioning based on the timing when the DR input turns ON.
At the start, the data in the RMV register is loaded into the positioning counter. When the DR input is ON, the LSI will output pulses and the positioning counter will start counting down pulses. When the positioning counter value reaches zero, the PCL stops operation.
Even if the DR input is turned OFF or ON again during the operation, it will have no effect on the operation. If you make the RMV register value 0 and start a positioning operation, the PCL will stop operation immediately without outputting any command pulses.
Turn ON the +DR signal to feed in the positive direction. Turn ON the -DR signal to feed in the negative direction.
By turning ON the EL signal corresponding to the feed direction, the axis will stop operation and issue an error interrupt (#INT output).

9-5. Origin position operation mode

The following six origin position operation modes are available.

MOD	Operation mode	Direction of movement
10h	Origin return operation	Positive direction
18h	Origin return operation	Negative direction
12h	Leaving the origin position operation	Positive direction
1Ah	Leaving the origin position operation	Negative direction
15h	Origin position search operation	Positive direction
1Dh	Origin position search operation	Negative direction

Depending on the operation method, the origin position operation uses the ORG, EZ, or ±EL inputs.
Specify the input logic of the ORG input signal in RENV1.ORGL. This register's terminal status can be monitored with an SSTSW (sub status) command.
Specify the input logic of the EZ input signal in RENV2.EZL. Specify the number for EZ to count for an origin return complete condition in the RENV3.EZD0 to 3. This register's terminal status can be monitored by reading RSTS.SEZ.
Specify the logic for the ±EL input signal using the ELL input terminals. Specify the operation to execute when the signal turns ON (immediate stop/deceleration stop) in RENV1.ELM. This register's terminal status can be monitored with an SSTSW.SPEL and SSTS.SMEL.
An input filter can be applied to the ORG input signal and ±EL input signal by setting the RENV1 register.
ORG input is sampled in synchronization with output pulses. Keep ORG input ON more than 1 pulse interval.

Set the ORG signal input logic <Set RENV1.ORGL (bit 7)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 n - - - - - - -
Read the ORG signal <SSTSW.SORG (bit14)> 0: Turn OFF the ORG signal 1: Turn ON the ORG signal	[SSTSW] (READ) 15 8 - n - - - - - -
Set the EZ signal input logic <Set RENV2.EZL (bit 23)> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 23 16 n - - - - - - -
Set the EZ count <Set RENV3.EZD0 to 3 (bits 4 to 7)> Specify the number for EZ to count that will indicate a zero return completion. Enter the value (the count minus 1) in EZD0 to 3. Setting range: 0 to 15.	[RENV3] (WRITE) 7 0 n n n n - - - -
Read 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 ±EL signal input logic <ELL input terminal> L: Positive logic input H: Negative logic input	
Specify a method for stopping when the ±EL signal turns ON <Set RENV1.ELM (bit 3)> 0: Immediate stop when the ±EL signal turns ON. 1: Deceleration stop when the ±EL signal turns ON.	[RENV1] (WRITE) 7 0 - - - - n - - -
Read the ±EL signal <SSTSW.SPEL (bit 12), SSTS.SMEL (bit 13)> SPEL = 0: Turn OFF + EL signal SPEL = 1: Turn ON + EL signal SMEL = 0: Turn OFF - EL signal SMEL = 1: Turn ON - EL signal	[SSTSW] (READ) 15 8 - - n n - - - -
Applying an input filter to the ±EL and ORG inputs <Set RENV1.FLTR (bit 26)> 1: Apply a filter to the ±EL and ORG inputs. By applying a filter, pulses shorter than 4 μsec will be ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -

9-5-1. Origin return operation

After writing a start command, the axis will continue feeding until the conditions for an origin return complete are satisfied.

PRMD.MOD: 10h Positive direction origin return operation
18h Negative direction origin return operation

When a zero return is complete, the LSI will reset the counter and output an ERC (deflection counter clear) signal.
The RENV3 register is used to set the basic origin return method. That is, whether or not to reset the counter when the origin return is complete. Specify whether or not to output the ERC signal in RENV1.EROR.
For details about the ERC signal, see 11-6-2, "ERC signal."

Set the origin return method <Set RENV3.ORM0 to 3 (bits 0 to 3)>

[RENV3] (WRITE)

7							0
-	-	-	-	n	n	n	n

0000: Origin return operation 0
- The axis will stop immediately (or make a deceleration stop when feeding at high speed) when the ORG input turns ON.
- COUNTER reset timing: When the ORG input turns ON.

0001: Origin return operation 1
- The axis will stop immediately (or make a deceleration stop when feeding at high speed) when the ORG input turns ON. Then, it will feed in the opposite direction at RFA constant speed until the ORG input turns OFF. Then, the axis will move back in the original direction at RFA speed and stop instantly when ORG input turns ON again.
- COUNTER reset timing: When the ORG input signal turns ON.

0010: Origin return operation 2
- After the ORG input turns ON when feeding at constant speed, the LSI will start counting EZ pulses. The axis will stop immediately when the LSI finishes counting EZ pulses.
  After the ORG input turns ON when feeding at high speed, the axis will start decelerating. At the same time, the LSI will start counting EZ pulses. When the LSI finishes counting EZ pulses, the axis will stop instantly.
- COUNTER reset timing: When finishing counting EZ pulses.

0011: Origin return operation 3
- After the ORG signal turns ON when feeding at constant speed, the LSI will start counting EZ pulses. The axis will stop instantly when the LSI finishes counting EZ pulses. After the ORG signal turns ON when feeding at high speed, the LSI will start counting EZ pulses. When the LSI finishes counting EZ pulses, the axis will decelerate and stop.
  When feeding at constant speed, movement on the axis stops immediately by counting the EZ signal after the ORG input is turned ON. When feeding at high speed, the axis will decelerate and stop by counting the EZ signal after the ORG input is turned ON.
- COUNTER reset timing: When finishing counting the EZ pulses.

0100: Origin return operation 4
- After the ORG input turns ON when feeding at constant speed , the axis will stop immediately (or make a deceleration stop when feeding at high speed). Then, the axis will start feeding in the opposite direction at RFA constant speed. After the ORG input turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting EZ pulses, the axis will stop instantly.
- COUNTER reset timing: When finishing counting the EZ pulses.

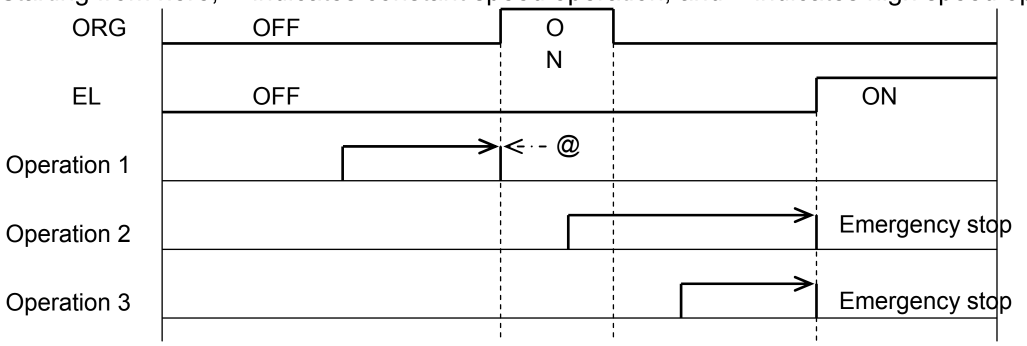
0101: Origin return operation 5
- After the ORG input turns ON when feeding at constant speed, the axis will stop immediately (or make a deceleration stop when feeding at high speed). Then, the axis will start feeding in the opposite direction. After the ORG input turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting EZ pulses, the axis will stop instantly (or make a deceleration stop when feeding at high speed).
- COUNTER reset timing: When finishing counting the EZ pulses.

0110: Origin return operation 6

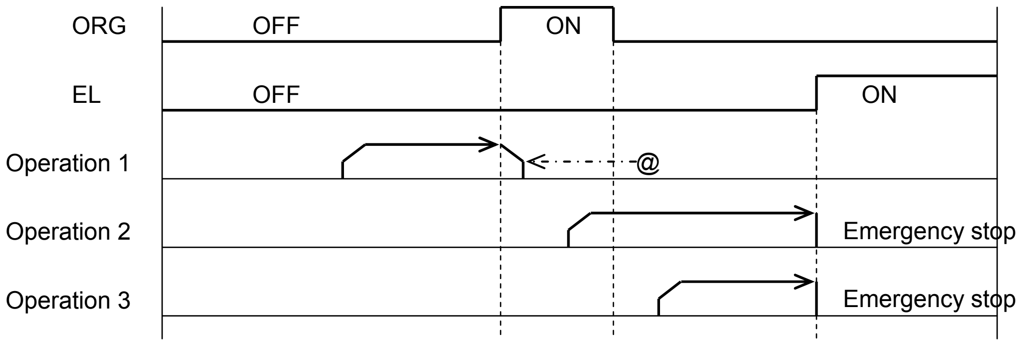
- After the EL input turns ON when feeding at constant speed, the axis will stop immediately (or make a deceleration when ELM is 1). Then, the axis will start feeding in the opposite direction at RFA constant speed. When the EL signal turns OFF, the axis will stop instantly when the LSI finishes counting the EZ pulses. - COUNTER reset timing: When the EL input is OFF. 0111: Origin return operation 7 - After the EL signal turns ON when feeding at constant speed, the axis will stop immediately (or make a deceleration when ELM is 1). Then, the axis will start feeding in the opposite direction at RFA constant speed. After the EL signal turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting EZ pulses, the axis will stop instantly. - COUNTER reset timing: When stopped by finishing counting the EL pulses. 1000:Origin return operation 8 - After the EL signal turns ON when feeding at constant speed, the axis will stop immediately (or make a deceleration when ELM is 1). Then, the axis will start feeding in the opposite direction at RFL constant speed. After the EL signal turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting EZ pulses, the axis will stop instantly. - CONTER reset timing: When finishing counting the EZ signal. 1001: Origin return operation 9 - After the process in origin return operation 0 has executed, it returns to zero (operates until COUNTER2 = 0). 1010: Origin return operation 10 - After the process in origin return operation 3 has executed, it returns to zero (operates until COUNTER2 = 0). 1011: Origin return operation 11 - After the process in origin return operation 5 has executed, it returns to zero (operates until COUNTER2 = 0). 1100: Origin return operation 12 - After the process in origin return operation 8 has executed, it returns to zero (operates until COUNTER2 = 0).	[RENV3] (WRITE) 7 0 - - - - n n n n
Settings after an origin return complete <Set RENV3.CU1R to 4R (bits 20 to 23)> CU1R (bit 20) =1: Reset COUNTER1 (command position) CU2R (bit 21) =1: Reset COUNTER2 (mechanical position) CU3R (bit 22) =1: Reset COUNTER3 (deflection counter) CU4R (bit 23) =1: Reset COUNTER4 (general-purpose)	[RENV3] (WRITE) 23 16 n n n n - - - -
Setting the ERC signal for automatic output <Set RENV1.EROR (bit 11)> 0: Does not output an ERC signal when an origin return is complete. 1: Automatically outputs an ERC signal when an origin return is complete.	[RENV1] (WRITE) 15 8 - - - - n - - -

9-5-1-1. Origin return operation 0 (ORM = 0000)

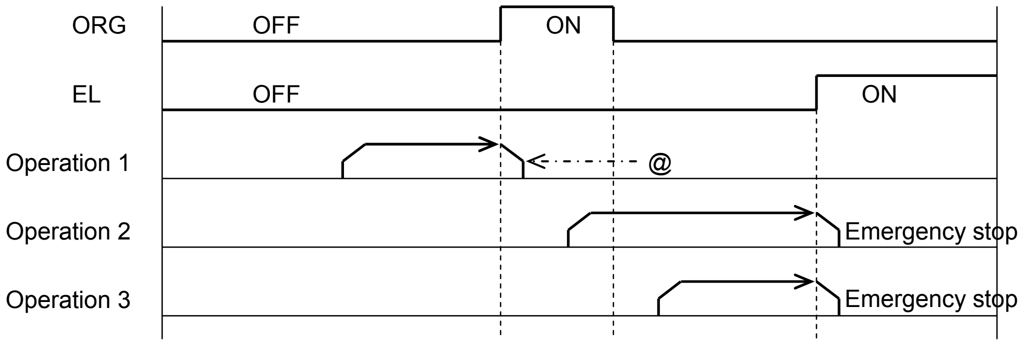
□ Constant speed operation <Sensor: EL (ELM = 0), ORG>
[Starting from here, □ indicates constant speed operation, and ■ indicates high speed operation.]



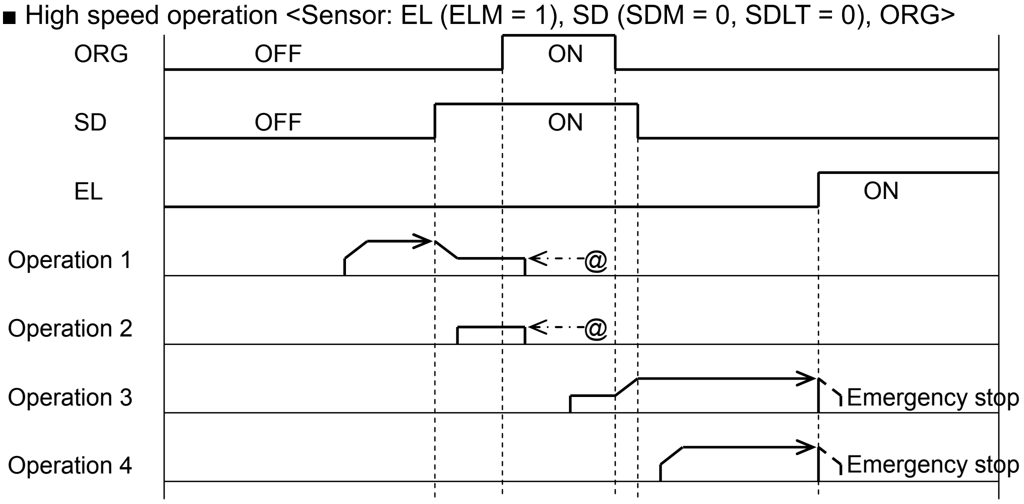
■ High speed operation <Sensor: EL (ELM = 0), ORG>
Even if the axis stops normally, it may not be at the origin position. However, COUNTER 2 (mechanical position) provides a reliable value



■ High speed operation <Sensor: EL (ELM = 1), ORG>
Even if the axis stops normally, it may not be at the origin position. However, COUNTER 2 (mechanical position) provides a reliable value.



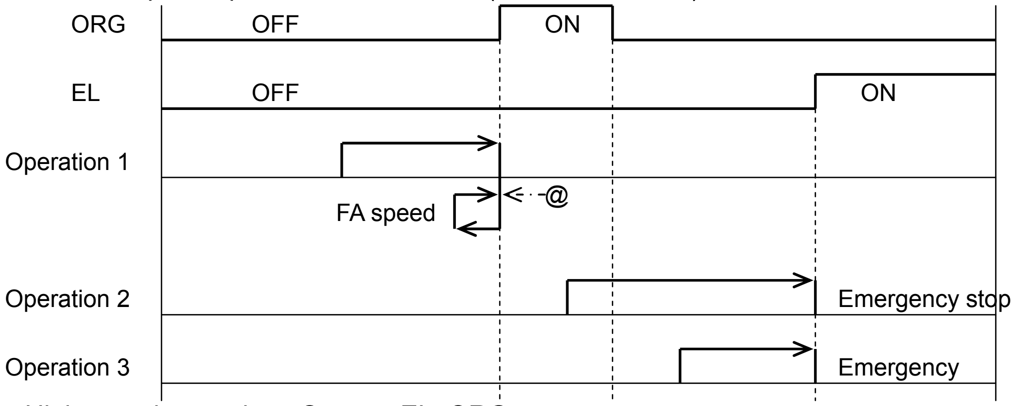
■ High speed operation <Sensor: EL (ELM = 1), SD (SDM = 0, SDLT = 0), ORG>



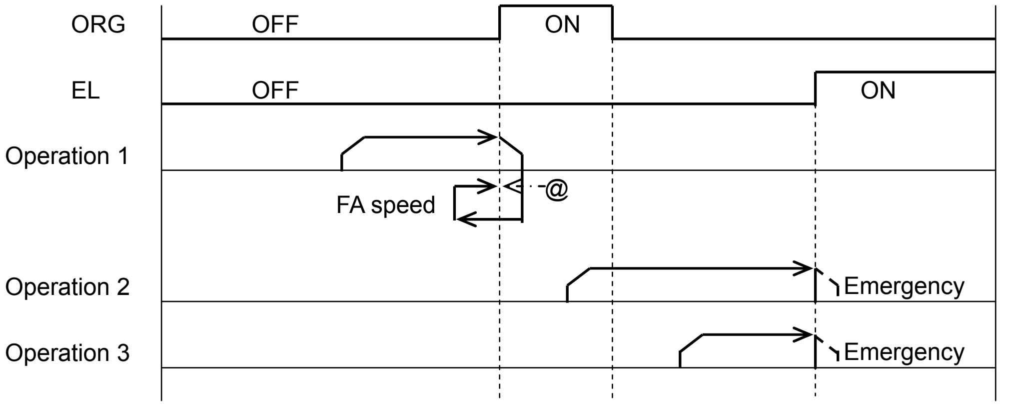
Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for stopping at the origin return .

9-5-1-2. Origin return operation 1 (RENV3.ORM=0001)

□ Constant speed operation <Sensor: EL (RENV1.ELM = 0), ORG>

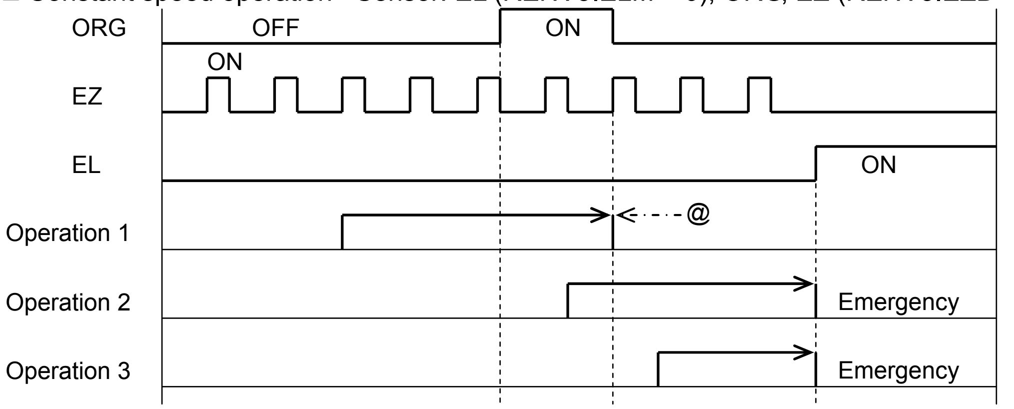


■ High speed operation <Sensor: EL, ORG>

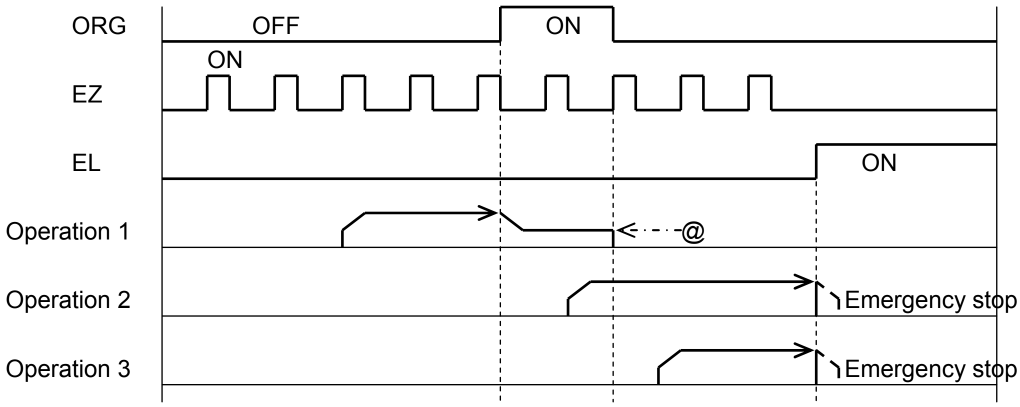


9-5-1-3. Origin return operation 2 (ORM = 0010)

□ Constant speed operation <Sensor: EL (RENV3.ELM = 0), ORG, EZ (RENV3.EZD = 0001)>



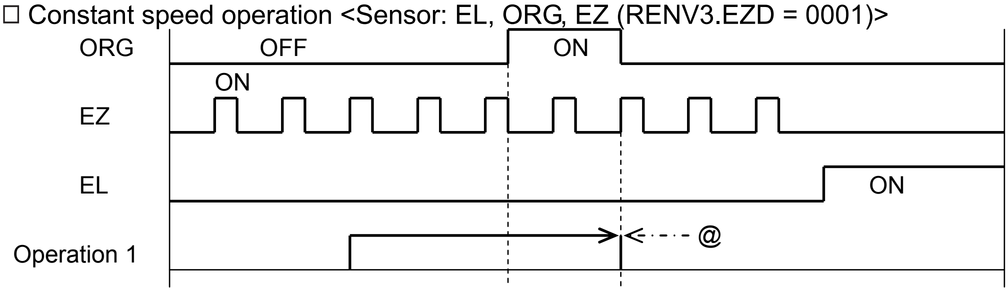
■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>



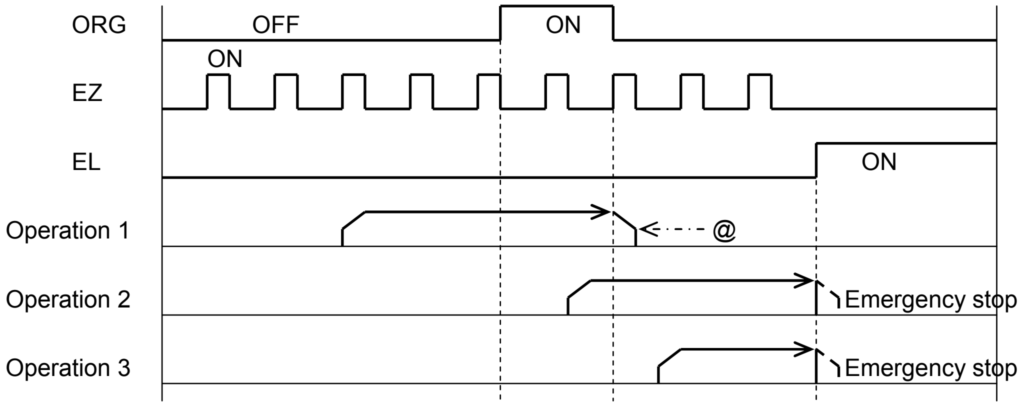
Note: Positions marked with "@" reflect ERC signal output timing when "Automatically output an ERC signal" is selected for stopping at the origin return.

9-5-1-4. Origin return operation 3 (RENV3.ORM = 0011)

☐ Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>

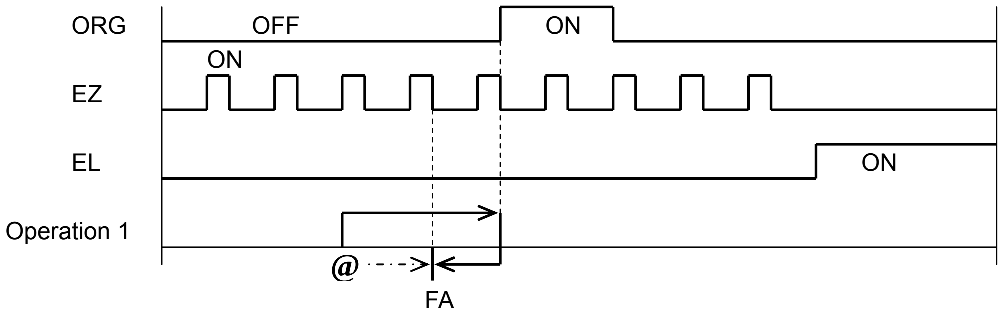


■ High speed operation <Sensor: EL,ORG, EZ (EZD = 0001)>

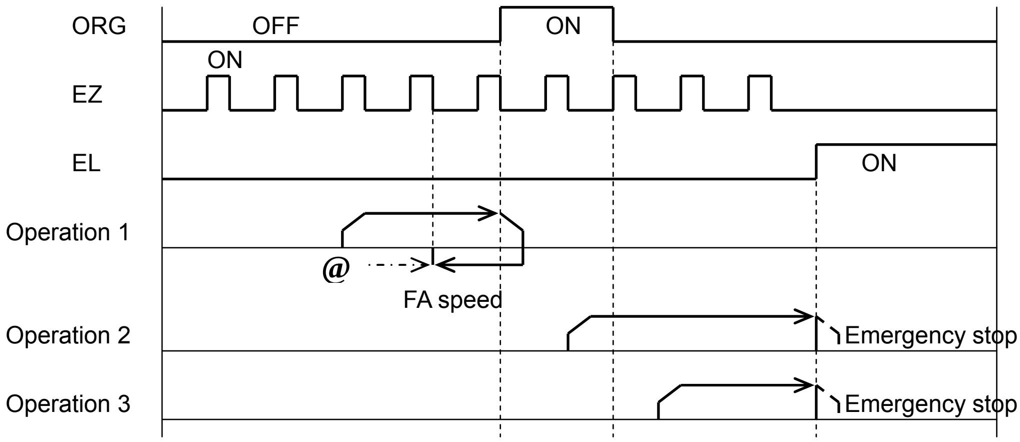


9-5-1-5. Origin return operation 4 (ORM = 0100)

☐ Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>



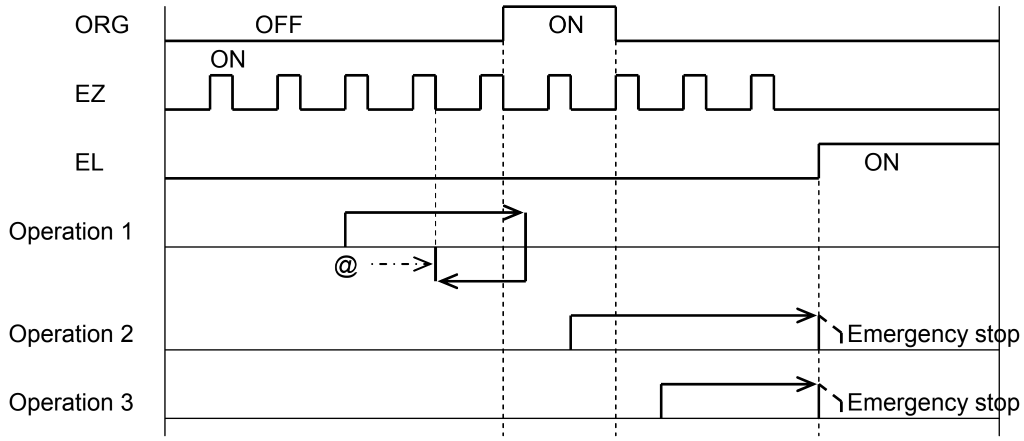
■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>



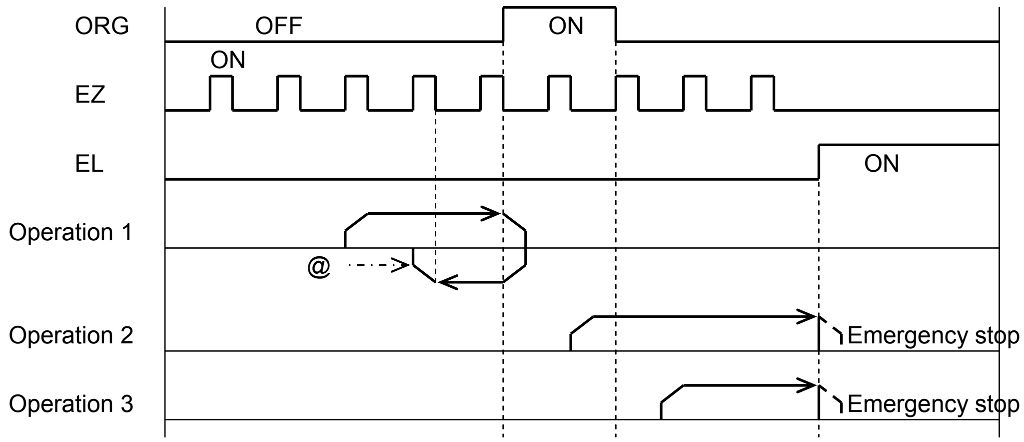
Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for stopping at the origin return.

9-5-1-6. Origin return operation 5 (ORM = 0101)

☐ Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>

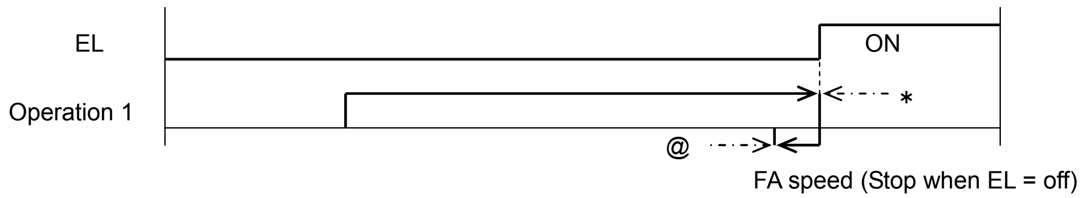


■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>

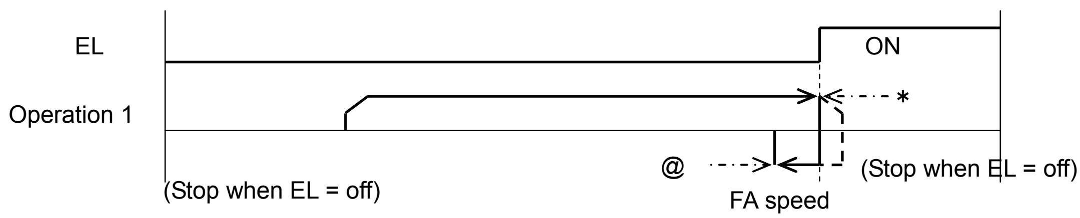


9-5-1-7. Origin return operation 6 (RENV3.ORM = 0110)

☐ Constant speed operation <Sensor: EL>



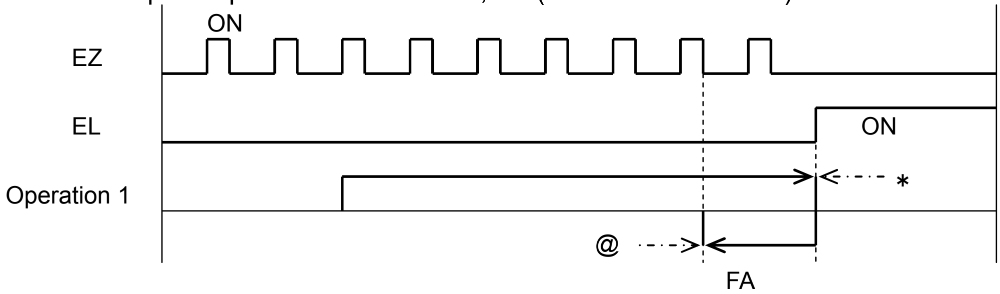
■ High speed operation <Sensor: EL>



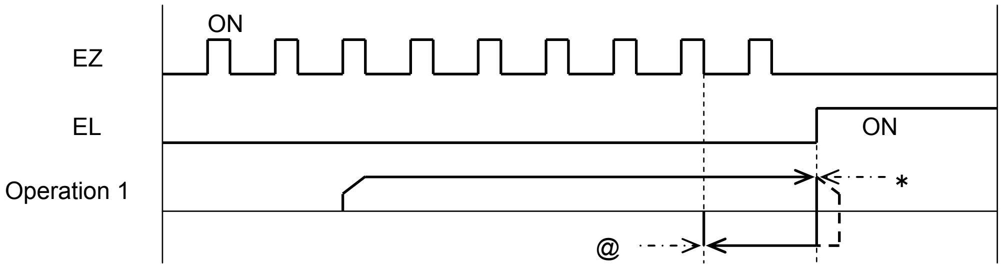
Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for stopping at the origin return.
Also, when EROE (bit 10) is 1 in the RENV1 register and ELM (bit 3) is 0, the LSI will output an ERC signal at positions marked with an asterisk (※).

9-5-1-8. Origin return operation 7 (RENV3.ORM = 0111)

☐ Constant speed operation <Sensor: EL, EZ (RENV3.EZD = 0001)>

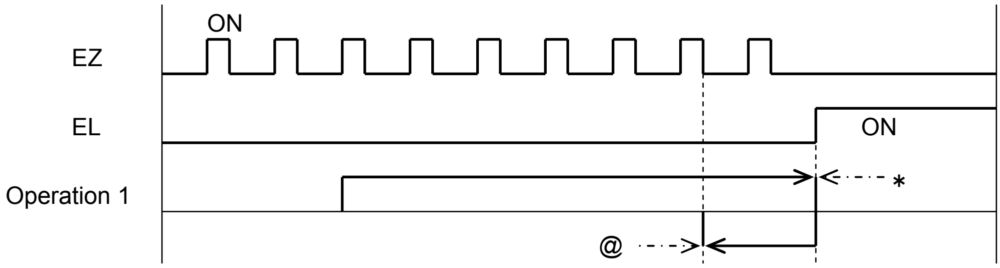


■ High speed operation <Sensor: EL, EZ (RENV3.EZD = 0001)>

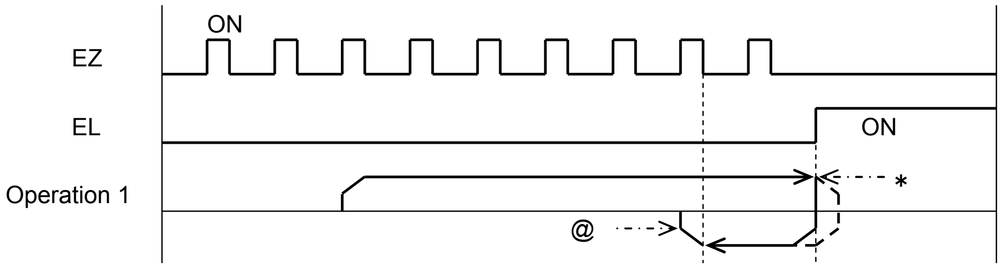


9-5-1-9. Origin return operation 8 (RENV3.ORM=1000)

☐ Constant speed operation <Sensor: EL, EZ (RENV3.EZD = 0001)>

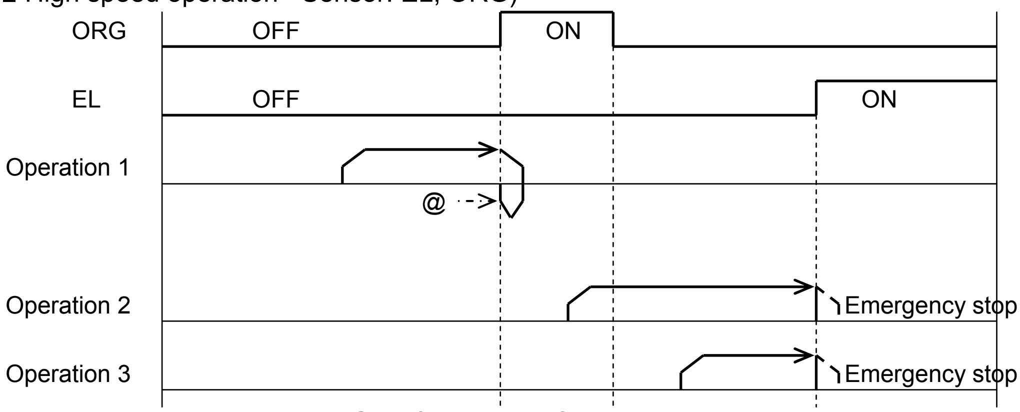


■ High speed operation <Sensor: EL, EZ (RENV3.EZD = 0001)>



9-5-1-10. Origin return operation 9 (RENV3.ORM = 1001)

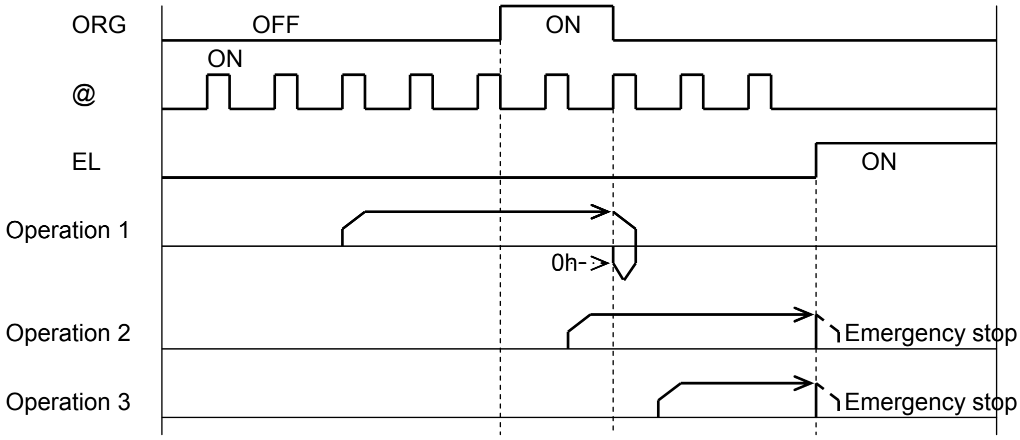
■ High speed operation <Sensor: EL, ORG)>



Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for stopping at the origin return.
Also, when REMV1.EROE (bit 10)="1" and RENV1.ELM (bit 3)="0" is 0, the LSI will output an ERC signal at positions marked with an asterisk (*).

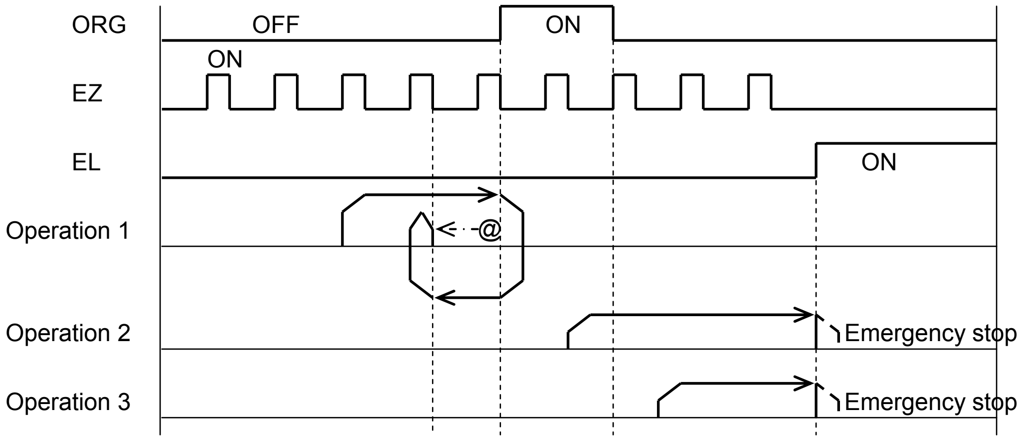
9-5-1-11. Origin return operation 10 (RENV3.ORM = 1010)

■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>



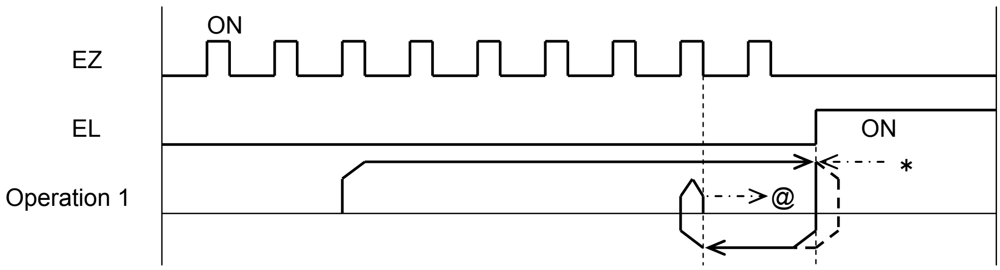
9-5-1-12. Origin return operation 11 (RENV3.ORM = 1011)

■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001)>



9-5-1-13. Origin return operation 12 (RENV3.ORM = 1100)

■ High speed operation <Sensor: EL, EZ (RENV3.EZD = 0001)>



Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for the zero stopping position. Also, when RENV1.EROE (bit 10)=1 and RENV1.ELM (bit 3)=0, the LSI will output an ERC signal at positions marked with an asterisk (*).

9-5-2. Leaving the origin position operations

After writing a start command, the axis will leave the origin position (when the ORG input turns ON).
Make sure to use the "Constant speed start command (50h, 51h)" when leaving the origin position.
When you write a start command while the ORG input is OFF, the LSI will stop the movement on the axis as a normal stop, without outputting pulses.
Since the ORG input status is sampled when outputting pulses, if the PCL starts at constant speed while the ORG signal is ON, it will stop operation after outputting one pulse, since the ORG input is turned OFF. (Normal stop)

PRMD.MOD: 12h Leave the origin position in the positive direction
1Ah Leave the origin position in the negative direction

9-5-3. Origin search operation

This mode is used to add functions to an origin return operation. It consists of the following possibilities.

1) An "Origin return operation" is made in the opposite direction to the one specified.
2) A "Leaving the origin position using positioning operations" is executed in the opposite direction to the one specified.
3) An "Origin return operation" is executed in the specified direction.

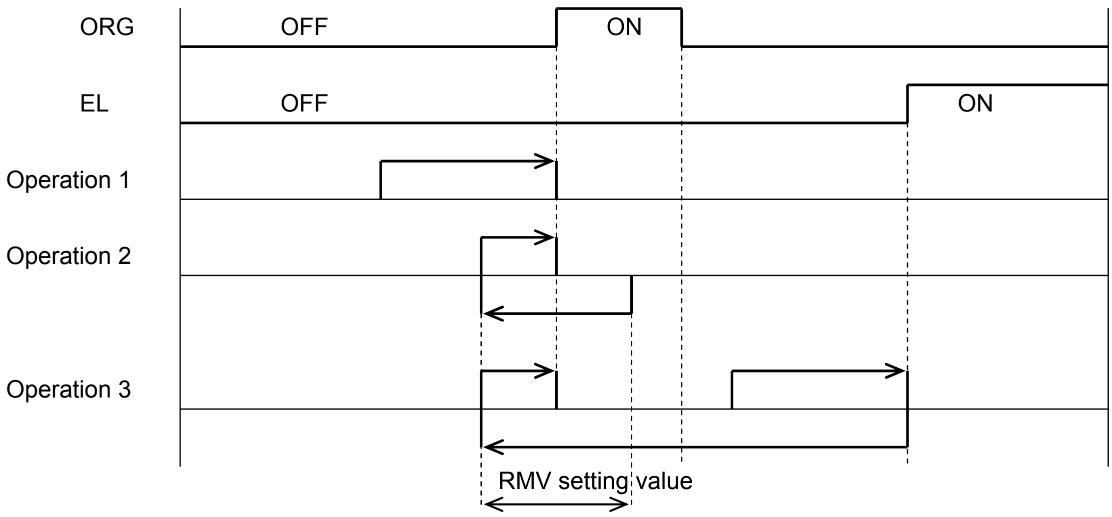
Operation 1: If the ORG input is turned ON after starting, movement on the axis will stop normally.
Operation 2: If the ORG input is already turned ON when starting, the axis will leave the origin position using positioning operations, and then begin an "origin return operation."
Operation 3: If movement on the axis is stopped by an EL signal while operating in the specified direction, the axis will execute an "origin return operation (ORM = 0000)" and a "leaving the origin position by positioning" in the opposite direction. Then it will execute an "origin return operation" in the specified direction.

When "leaving the origin position by positioning," the axis will repeat the positioning operation for the number of pulses specified in the RMV (target position) register, until the origin position has been left. Enter a positive number (1 to 134,217,727) in the RMV register.

PRMD.MOD: 15h Origin search operation in the positive direction
1Dh Origin search operation in the negative direction

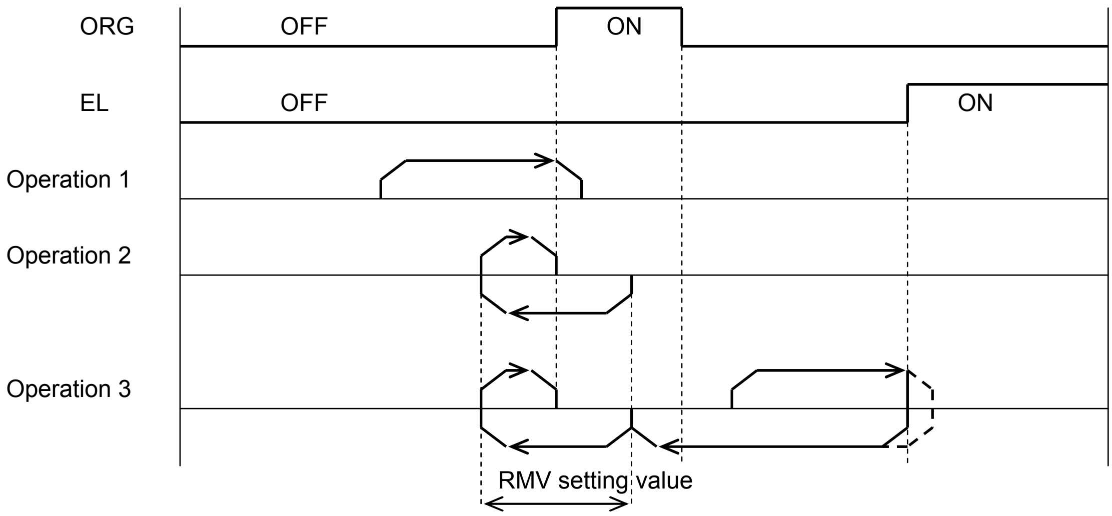
9-5-3-1. Origin return operation 0 (RENV3.ORM=0000)

☐ Constant speed operation <Sensor: EL, ORG>



■High speed operation <Sensor: EL, ORG>

Even if the axis stops normally, it may not be at the origin position. However, COUNTER2 (mechanical position) provides a reliable value.



9-6. EL or SL operation mode

The following four modes of EL or SL (software limit) operation are available.

PRMD.MOD	Operation mode	Direction of movement
20(h)	Operate until reaching the +EL or +SL position.	Positive direction
28(h)	Operate until reaching the -EL or -SL position.	Negative direction
22(h)	Leave from the -EL or -SL positions.	Positive direction
2A(h)	Leave from the +EL or +SL positions.	Negative direction

To specify the ±EL input signal, set the input logic using the ELL input terminal. Select the operation type (immediate stop / deceleration stop) to be executed when the input from that terminal is ON using RENV1.ELM. The status of the terminal can be monitored using SSTSW (sub status).
For details about setting the SL (software limit), see section 11-11-2, "Software limit function."

Select the ±EL signal input logic <ELL input terminal> L: Positive logic input H: Negative logic input	
Select the stop method to use when the ±EL signal is turned ON <RENV1.ELM (bit 3)> 0: Stop immediately when the ±EL signal turns ON. 1: Decelerates and stops when the ±EL signal turns ON.	[RENV1] (WRITE) 7 0 - - - - n - - -
Reading the ±EL signal <SSTSW.SPEL (bit 12), SSTSW.SMEL (bit 13)> SPEL=0: Turn OFF +EL signal SPEL=1: Turn ON +EL signal SMEL=0: Turn OFF -EL signal SMEL=1: Turn ON -EL signal	[SSTSW] (READ) 15 8 - - n n - - - -
Setting the ±EL input filter <RENV1.FLTR (bit 26)> 1: Apply a filter to the ±EL input. After applying a filter, signals shorter than 4 μsec will be ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -

9-6-1. Feed until reaching an EL or SL position

This mode is used to continue feeding until the EL or SL (software limit) signal is turned ON and then the operation stops normally.
When a start command is written on the position where the EL or SL signal is turned ON, the LSI will not output pulses and it will stop the axis normally. When a start command is written while the EL and SL signals are OFF, the axis will stop when the EL or SL signal is turned ON. (Normal stop)

PRMD.MOD: 20(h) Feed until reaching the +EL or +SL position.
28(h) Feed until reaching the -EL or -SL position.

9-6-2. Leaving an EL or SL position

This mode is used to continue feeding until the EL or SL (software limit) signal is turned OFF.
When a start command is written on the position where the EL and SL signals are turned OFF, the LSI will not output pulses and it will stop the axis normally.
When starting an operation while the EL input or SL signal is ON, the G9103 will stop operation normally when both the EL input and SL signal are OFF.

PRMD.MOD: 22(h) Leave from a -EL or -SL position
2A(h) Leave from a + EL or +SL position

9-7. EZ count operation mode

This mode is to operate until EZ signal counts reaches the number (EZD setting value +1) written into the RENV3 register.

PRMD.MOD: 24(h) Feed until the EZ count is completed in positive direction.
2C(h) Feed until the EZ count is completed in negative direction.

After a start command is written, the axis stops immediately (or decelerates and stops when feeding at high speed) after the EZ count equals the number stored in the register.
The EZ count can be set from 1 to 16.
Use the constant speed start command (50(h), 51(h)) for this operation. When the high speed start command is used, the axis will start decelerating and stop when the EZ signal turns ON, so that the motion of the axis overruns the EZ position.
Specify logical input for the EZ signal in RENV2.EZL, and the EZ number to count to in RENV3.EZD.. The terminal status can be monitored by reading the RSTS (extension status) register.

Setting the input logic of the EZ signal <Set RENV2.EZL (bit 12)> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 23 16 n - - - - - - -
Setting the EZ count number <Set RENV3.EZD0 to 3 (bits 4 to 7)> Specify the EZ count number after an origin return complete condition. Enter a value (the number to count to minus 1) in EZD 0 to 3. Setting range: 0 to 15.	[RENV3] (WRITE) 7 0 n n n n - - - -
Reading the EZ signal < RSTS.SEZ (bit 16)> 0: Turn OFF the EZ signal 1: Turn ON the EZ signal	[RSTS] (READ) 15 8 - - - - - n - -

9-8. Interpolation operations

9-8-1.Interpolation operations

In addition to each independent operation, this LSI can execute the following interpolation operations.

PRMD.MOD	Operation mode
60h	Continuous linear interpolation 1 for 2 to 4 axes
61h	Linear interpolation 1 for 2 to 4 axes
62h	Continuous linear interpolation 2 for 1 to 4 axes
63h	Linear interpolation 2 for 1 to 4 axes
64h	Circular interpolation (CW)
65h	Circular interpolation (CCW)
66h	CW circular interpolation synchronized with the U axis

PRMD.MOD	Operation mode
67h	CCW circular interpolation synchronized with the U axis.
68h	Continuous linear interpolation 1 synchronized with PA/PB input
69h	Linear interpolation 1 synchronized with PA/PB input
6Ah	Continuous linear interpolation 2 synchronized with PA/PB input.
6Bh	Linear interpolation 2 synchronized with PA/PB input
6Ch	CW circular interpolation synchronized with PA/PB input
6Dh	CCW circular interpolation synchronized with PA/PB input

Continuous linear interpolation is the same as the linear interpolation used to feed multiple axes at specified rates, and to start and stop feeding using commands such as the continuous mode commands.
Interpolation 1 executes an interpolation operation between any two to four axes in the LSI.
Interpolation 2 is used to control five axes or more using more than one LSI, and to control feeding using linear interpolation.
Independent operation of the un-interpolated axes is also possible.
The interpolation settings and operation status can be monitored by reading the RIPS (interpolation status) register.
The RIPS register is shared by all axes. Reading from any axis will return the identical information.
Write start and stop commands to all axes to execute interpolation by setting SELx, SELy, SELz and SELu in COMB1.

[Interpolation operations that can be combined with this LSI]

1) Linear interpolation 1 of two axes.
2) Linear interpolation 1 of three axes.
3) Linear interpolation 1 of four axes.
4) Circular interpolation of two axes
5) Linear interpolation 1 of two axes and circular interpolation of two axes

Axes that are not involved in one of the interpolation operations 1) to 5) above, can be operated independently or can be used to execute a linear interpolation 2.

9-8-2. Interpolation control axis

In Circular interpolation and Linear interpolation 1, specify the speed for one axis only. This axis is referred to as the interpolation control axis. Interpolation control axes can only be in the order X, Y, Z, and U for the axes that are interpolated.
When you want to execute both a circular interpolation and a linear interpolation 1 simultaneously, there will be two interpolation control axes.
When linear interpolation 2 is selected, each axis will be used to control the interpolation.

[Relationship between an interpolation operation and the axes used for interpolation control]

No	Interpolation operation	Interpolation control axis
1)	Linear interpolation 1 of the X, Y, Z, and U axes.	X axis
2)	Linear interpolation 1 of the X, Y, and Z axes.	X axis
3)	Linear interpolation 1 of the Y, Z, and U axes.	Y axis
4)	Linear interpolation 1 of the Y and U axis	Y axis
5)	Circular interpolation of the X and U axis	X axis
6)	Circular interpolation of the X and Z axes, and linear interpolation 1 of the Y and U axes	Circular interpolation: X axis Linear interpolation 1: Y axis

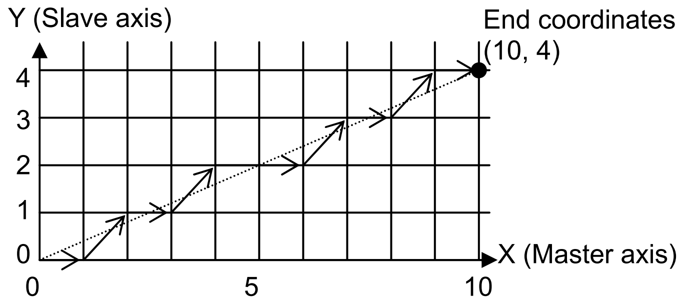
9-8-3. Synthesized speed constant control

This function is used to create a constant synthesized speed for linear interpolation 1 and circular interpolation operations. When linear interpolation 2 is selected, this function cannot be used.
To enable this function, set the PRMD.MIPF (bit 15) to "1" for the axes that you want to have a constant synthesized speed. When the same interpolation mode is selected, the axes whose PRMD.MIPF is set to "1" will have a longer pulse output interval: multiplied by the square root of two ($\sqrt{2}$) for two axis simultaneous output, and by the square root of three ($\sqrt{3}$) for three axis simultaneous output.
For example, when applying linear interpolation 1 to the X, Y, and Z axes, and PRMD.MIPF =1 for only the Y and Z axes, the interval before a pulse output on another axis after simultaneous pulse output on the Y and Z axes will be multiplied by $\sqrt{2}$. When X and Y, or X and Z output pulses at the same time, the interval until the next pulse output will not change.
The synthesized speed constant control can only be used for 2 or 3 axes. When applying linear interpolation 1 to four axes, if PRND.MIPE = 1 for all four axes, and if all four axes output pulses at the same time, the interval will also be multiplied by $\sqrt{3}$.
When the synthesized speed constant control bit is turned ON (MIPF = 1), the synthesized speed (while performing interpolation) will be the operation speed (PRFH) or the initial speed (PRFL) of the interpolated axes.
SRUN, SEND, and SERR in MSTSW (main status byte) for the interpolated axis will change using the same pattern.
The RSPD (speed monitor) feature is only available for the interpolation control axes. However, when linear interpolation 2 is used, the value read out will be the main axis speed.

<Precautions for using the synthesized speed constant control bit (MIPF = 1)>

1) Positioning is possible only at the unit's resolution position for machine operation.
Therefore, even if an interpolation operation is selected, the machine will use the following points to approximate to an ideal or arc, and the actual feed pattern will be point to point (zigzag feeding). With this feed pattern, the actual feed amount will be longer than the ideal linear line or an ideal arc. The function of the synthesized constant speed control in this LSI is to make constant synthesized speeds for multiple axes in simultaneous operation, which does not mean that the speed through the ideal locus (trajectory) is constant.

For example, with linear interpolation in the figure on the right (using the constant synthesized speed feature), the PCL will make a constant synthesized speed in order to feed at a 45° angle by decreasing each axis's speed to $1/\sqrt{2}$.
Therefore, the feeding interval when the feed speed is 1 pps will be $6 + 4\sqrt{2}$ =11.66 seconds.



The length of the ideal line (dotted line) is $\sqrt{(10^2+4^2)}$ = 10.77. If the machine can be fed by just following the ideal line, the feed interval will be 10.77 seconds.
Please take note of the above when using synthesized speed constant control.

2) Acceleration/deceleration operations when the synthesized speed constant control bit is ON (PRMD.MIPF = 1)
Basically, please use a constant speed when MIPF = 1. (The synthesized speed will vary with the acceleration/deceleration.)
When MIPF = 1 and you select linear interpolation 1 or circular interpolation with acceleration /deceleration, the following limitations apply.
- Make the acceleration rate (PRUP) and deceleration rate (PRDR) for the control axes equal.
- Do not change the speed during S-curve acceleration/deceleration.
Failure to follow these guidelines may cause the PCL to decelerate abnormally.

9-8-4. Continuous linear interpolation 1 (PRMD.MOD: 60h)

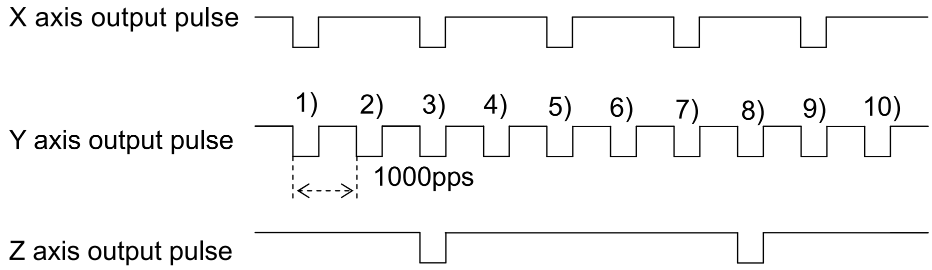
This is the same as linear interpolation 1, and each axis operates at a speed corresponding to the PRMV setting. However, the PCL will continue to output pulses until a stop command is received.
This mode only uses the rate from the PRMV setting for all of the interpolated axes. Therefore, if the PRMV setting for the all of the interpolated axes is zero, the PCL will output pulses to all the interpolated axes at the same speed.

9-8-5. Linear interpolation 1 (PRMD.MOD: 61h)

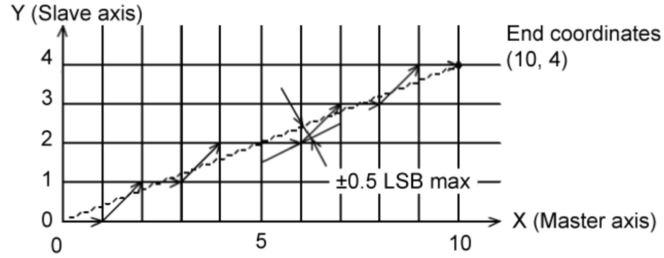
Linear interpolation 1 is used to allow a single LSI to provide interpolation operations between any 2 to 4 axes.
If only one axis is specified and operation is started, an error (ESDT: Stop due to operation data error) will occur.
After setting the operation speed for the interpolation control axes, specify whether to use or not the synthesized speed constant control in the PRMD registers, or specify an end point position in the PRMV register for all of the interpolated axes.
The direction of operation is determined by the sign of the value in the PRMV register.
Automatically, the axis with the maximum feed amount (maximum absolute value in the PRMV register) will be considered as the master axis. The other axes will be the slave axes.
When a start command is written, the LSI will output pulses to the master axis and the slave axes will be supplied a smaller number of pulses than the master axis. Write a start command by setting either the SELx to SELu bits corresponding to the interpolation axes in COMB1 to 1. Either axis can be used to write a start command.

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

Setting	X axis	Y axis	Z axis
MOD	61h	61h	61h
MIPF	0 (OFF)	0 (OFF)	0 (OFF)
PRMV value	5	10	2
Operation speed	1000 pps		
Interpolation control axis	O		
Master axis / slave axis	Slave axis	Master axis	Slave axis



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



9-8-6. Continuous linear interpolation 2 (PRMD.MOD: 62h)

Same as Linear Interpolation 2: the PCL controls each axis using speeds that correspond to the ratios of the values set in PRIP and PRMV. However, in continuous mode the PCL will continue to output pulses until it receives a stop command.

9-8-7. Linear interpolation 2 (PRMD.MOD: 63h)

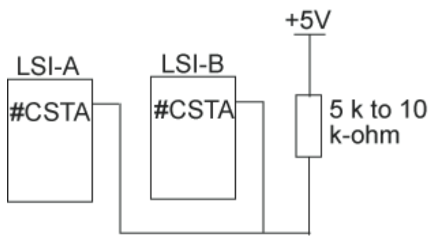
Linear interpolation 2 is used for linear interpolations between 5 or more axes and uses more than one LSI for control.
In this mode, the PCL cannot synchronize the acceleration/deceleration timing between interpolated axes, so this mode cannot be used with acceleration/deceleration.
In order to execute a linear interpolation using multiple LSIs, you must use a simultaneous start signal (#CSTA signal).
For details about the #CSTA signal, see section 11-7, "External start, simultaneous start."
The axis with the maximum amount to be fed is referred to as the master axis during the interpolation and the other axes are slave axes.
Enter the PRMV register setting for the master axis in the PRIP registers of each axis (including the master axis).
In the PRMV registers of the slave axes, enter end point of each axis.
Specify the speed data (PRFL, PRFH, PRUR, PRDR, PRMG, PRDP, PRUS, and PRDS) for the slave axes to be the same as for the master axis.
The feed direction is determined by the sign of the value in the PRMV register.
After writing "01" into PRMD.MSY (bits 18 and 19) of the interpolated axes, write a start command and set the axes to wait for the #CSTA signal input. By entering a #CSTA signal, all of the axes that set to "waiting for #CSTA input" on all of the LSIs will start at the same time.
The master axis provides pulses constantly. The slave axes provide some of the pulses fed to the master axis, but some are omitted.

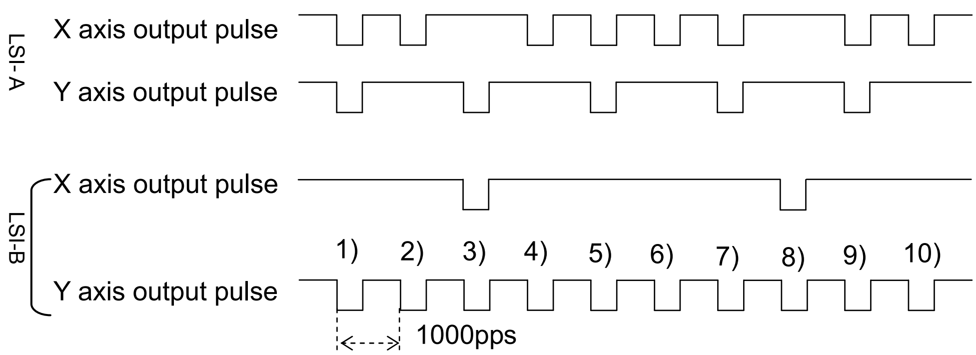
[Setting example]

1) Connect the #CSTA signals between LSI-A and LSI-B.
2) Set up the LSIs as shown below. (Set the PRMD to start with inputting a #CSTA signal.)
3) Write start commands (LSI-A: 0951h, LSI-B: 0651h).
4) Write a #CSTA signal input command (06h) to the X axis on LSI-A.

After completing steps 1) to 4) above, the LSIs will output pulses using the timing shown in the figure below.

Setting	LSI-A		LSI-B	
	X axis	U axis	Y axis	Z axis
PRMD	0004 0063h	0004 0063h	0004 0063h	0004 0063h
PRMV value	8	5	2	10
PRIP value	10	10	10	10
Operation speed	1000 pps	1000 pps	1000 pps	1000 pps
Master axis / slave axis	Slave axis	Slave axis	Slave axis	Master axis





Note: If you start linear interpolation 2 while PRIP = 0, an operation data error (REST.ESDT=1) will occur.

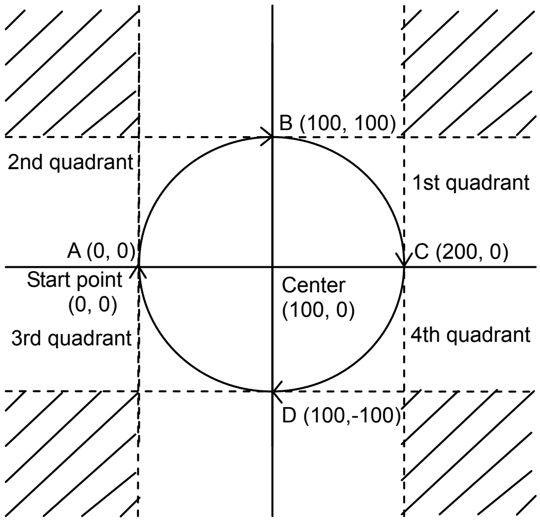
9-8-8. Circular interpolation

This function provides CW circular interpolation (PRMD.MOD: 64h) and CCW circular interpolation (PRMD.MOD: 65h) between any two axes.
If only one axis or 3 to 4 axis is specified for circular interpolation and a start command is written, a data setting error will occur.
Circular interpolation takes the current position as the starting point (coordinate 0, 0) regardless of the values in the counters (COUNTER1 to 4).
After specifying the speed for each interpolated axis, specify whether or not to apply synthesized speed constant control (PRMD.MIPF) for each axis, the end points (the PRMV register value), and the center point (the PRIP register value). If the end point is 0 (the starting point), both axes will draw a simple circle.
The synthesized speed used in the circular interpolation will be the speed (FH/FL) set for the interpolated axes if the synthesized speed constant control is ON (PRMD.MIPF = 1) for both axes.
Write a start command after setting SELx to SELu in COMB1 to 1. Either axis can be used to write a start command.

[Setting example]

As shown in the table below, specify the MOD, MIPF, PRMV, PRIP and operation speed for each axis being interpolated and write a start command (ex. 0351h) that will be used by both axes. The axes will move as shown on the right.

StepNo / Set value	A X axis	A Y axis	B X axis	B Y axis	C X axis	C Y axis	D X axis	D Y axis
MOD	64h (CW circular interpolation)							
MIPF	1 (turn ON synthesized constant speed control)							
PRMV value	0	0	100	100	200	0	100	-100
PRIP value	100	0	100	0	100	0	100	0
Operation result	Simple circle		90^o arc		180^o arc		270^o arc	



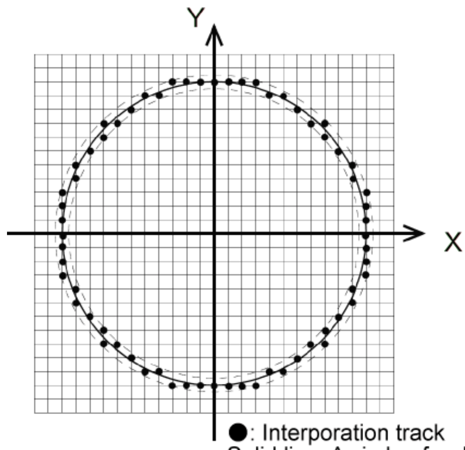
This LSI terminates a circular interpolation operation when either of the axes reaches the end point in the last quadrant, and the end point can be specified as the whole number coordinates nearest to the end position. For this reason, even though the circular interpolation operation is complete, the PCL will not be at the end coordinate specified. To move to the coordinates of the specified end point when the circular interpolation operation is complete, set the PRMD.MPIE "1" and turn ON the end point draw function. After circular interpolation operation, the axis move at the same speed as circular interpolation until it reaches specified end point.
Please note that the axes will not stop moving if the end point of the circular interpolation is set within the shaded areas (perpetual circular motion).

When PRMD.MIPM=1, end point in the last quadrant is controlled on 45^o basis (changed from 90^o basis). Therefore, determination to complete makes an arc longer.

[Circular interpolation precision]

The circular interpolation function draws a circular from the current position to the end coordinate moving CW or CCW.
The positional deviation from the specified curve is ±0.5 LSB.
The figure on the right is an example of how to draw a simple circle with a radius of 11 units.

The LSB refers the minimum feeding unit of the PRMV register setting value. It corresponds to the resolution of mechanical system (size of the cells in the figure right.)



●: Interporation track
Solid line: A circle of radius 11
Dotted line: A circle of radius 11±0.5

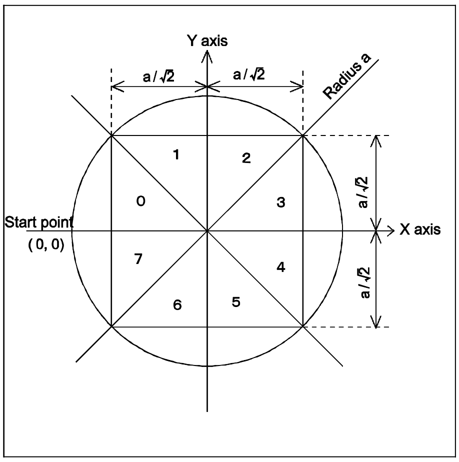
[Circular interpolation with acceleration/deceleration]

To use circular interpolation with acceleration/deceleration, you have to enter the number of pulses required for circular interpolation (circular interpolation step numbers) in the PRCI register for the control axis.

To calculate the number of pulses required for circular interpolation, break the area covered by the X and Y axes into 8 (0 to 7) sections, using the center coordinate of the circular interpolation as the center point. See the figure below.

The output pulse status of each axis in each area is as follows

Area	X axis output pulse	Y axis output pulse
0	Output according to the interpolation calculation result	Always output
1	Always output	Output according to the interpolation calculation result
2	Always output	Output according to the interpolation calculation result
3	Output according to the interpolation calculation result	Always output
4	Output according to the interpolation calculation result	Always output
5	Always output	Output according to the interpolation calculation result
6	Always output	Output according to the interpolation calculation result
7	Output according to the interpolation calculation result	Always output



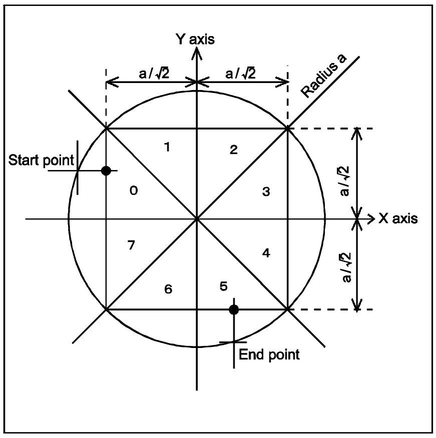
The table above shows the PCL output pulses for either of the axes in each area.

Therefore, the number of pulses required for circular interpolation (the number of circular interpolation steps) is equal to the number of pulses to move around the trajectory of a square that is surrounded by the circle used for the circular interpolation.

For example, to draw a 90^{o} arc with radius "a," the number of pulses required for circular interpolation will be $(a/\sqrt{2}) \times 2$. Enter this value in the PRCI register.

To obtain the number of steps for any start and end points, follow the procedure below.

1) First, determine the area that the start point belongs to (area 0 to 7). Then, draw a horizontal (vertical) line to find the contact point with the square inside the circle.
2) Next, determine the area that the end point belongs to (area 0 to 7). Then, draw a vertical (horizontal) line to find the contact point with the square inside the circle.
3) Find the distance between the two contact points on the square (from 1) and 2) above) and enter this value in the PRIC register.



To continue the end point draw function while setting PRMD.MPIE to "1", enter the value in the PRCI register after adding number of pulses required for the end point draw function.

Note 1: The PRCI register value is used to trigger the start of the deceleration timing. When a smaller value is entered, the PCL will start deceleration sooner and will apply the FL constant time. When a larger value is entered, the PCL will delay the beginning of deceleration and then will have to stop suddenly from faster than the FL speed.
However, the interpolation trajectory is the same as the constant speed circular interpolation.

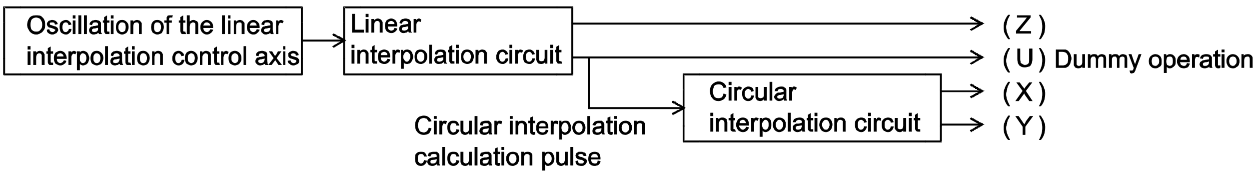
Note 2: To specify a ramp down point manually, think of the PRCI setting as a number of output pulses, so that the PRDP calculation formula for the positioning operation can be used. However, this formula cannot be used when the synthesized constant speed operation is ON. In this case, there is no other way to obtain a ramp down point except by conducting a test to get a value from the change of the RICI value.

9-8-9. Circular interpolation synchronized with the U axis

By synchronizing with the U axis, any two axes can be used for CW circular interpolation (PRMD.MOD: 66h) or CCW circular interpolation (PRMD.MOD: 67h).
If you specify circular interpolation for one axis or for 3 to 4 axes, and try to start the operation, the PCL will declare a data setting error.
When the U axis positioning counter (RPLS) reaches 0 while starting or during a circular interpolation, the PCL will also declare a data setting error.

By simultaneously using with linear interpolation, the PCL can synchronize one axis while performing a circular interpolation on two other axes. This function can be used for things like a circular interpolation between the X and Y axes and to adjust the angle of a jig toward an arc tangent point with the Z axis. Also, in this operation the U axis operation will be a dummy motion and it cannot be used for any other purpose.

<Conceptual figure>



Using the operation above, set the operation mode (RMD) for the X and Y axes to 66H (67h), and set the Z and U axes to 61h.
Enter the number of circular interpolation steps in the PRMV register for the U axis.

For details about how to obtain the number of circular interpolation steps, see the discussion of "circular interpolation with acceleration/deceleration" in the previous section.
To write a start or stop command, make all the bits in SELx to SELu of the COMB1 register equal to "1." Any axis can be used to write "1."

9-8-10. Interpolation operation synchronized with PA/PB

This function uses the PA/PB input signal (after magnification or division) instead of the internal clock.
Any PA/PB input after the interpolation operation is complete will be ignored.

9-8-11. Operation during interpolation

- Acceleration/deceleration operations
  Acceleration and deceleration (linear and S-curve) can be used with Linear interpolation 1 and circular interpolation operations .Automatic setting of ramp down point is available. However, set the MSDP and MADJ in the PRMD register the same for all of the interpolated axes.

  To control the ramp down point while using linear interpolation1, the PCL executes a comparison of RPLS and RSDC for the longest axis. The RSDC setting for any shorter axes will be invalid. However, if more than one axis has the same length and they are the longest axes, to specify a ramp down point manually you must enter the same value for all of the interpolated axes.

  To control the ramp down point while using circular interpolation, the PCL executes a comparison of RCIC and RSDC on the control axis. Therefore, to specify a ramp down point manually, write to RSD on the control axis.

- Error stop
  If any of the axes being interpolated stops with an error, all of the axes being interpolated will stop (SSTSW.SERR = 1). By reading the REST (error stop cause) register, you can determine which axis actually stopped with an error.

- SD input
  When SD input is enabled (PRMD.MSDE is set to 1), and if the SD input turns ON on any axis interpolated, all axes will decelerate or decelerate and stop.

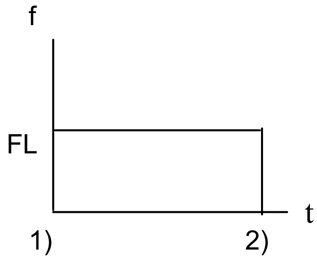
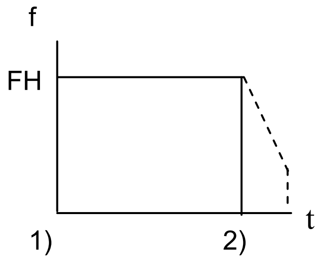
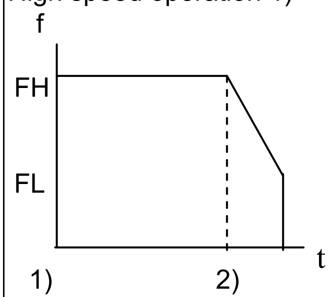
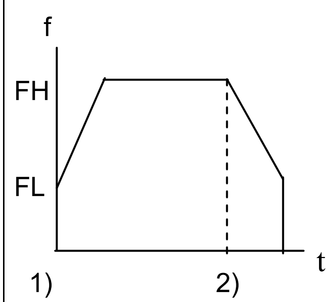
- Idling control
  If any axis is in idling range, none of the interpolated axes will accelerate.

- Correction function
  When a direction is changed by switching of quadrants during circular interpolation, backlash correction and slip correction control cannot be used.

- Continuous interpolation
  The PCL can use the pre-register to make a continuous linear interpolation or circular interpolation. However, when the interpolated axes change during a continuous interpolation, special care is required.
  An example of the settings for continuous interpolation using the pre-register is shown in section 11-14-1, "Start triggered by another axis stopping."

10. Speed patterns

10-1. Speed patterns

Speed pattern	Continuous mode	Positioning operation mode
FL constant speed operation 	1) Write an FL constant speed start command (50h). 2) Stop feeding by writing an immediate stop (49h) or deceleration stop (4Ah) command.	1) Write an FL constant speed start command (50h). 2) Stop feeding when the positioning counter reaches zero, or by writing an immediate stop (49h) or deceleration stop (4Ah) command.
FH constant speed operation 	1) Write an FH constant speed start command (51h). 2) Stop feeding by writing an immediate stop command (49h).	1) Write an FH constant speed start command (51h). 2) Stop feeding when the positioning counter reaches zero, or by writing an immediate stop (49h) command.
	* When the deceleration stop command (4Ah) is written to the register, motion of an axis starts deceleration.	
High speed operation 1) 	1) Write high speed start command 1 (52h). 2) Start deceleration by writing a deceleration stop command (4Ah). * When the deceleration stop command (49h) is written to the register, an axis immediately stops * When idling pulses are added by setting IDL in RENV5 to a non-zero value, after outputting idling pulses at FL speed, motion of an axis will accelerate to FH speed.	1) Write high speed start command 1 (52h). 2) Start deceleration when a ramping-down point is reached or by writing a deceleration stop command (4Ah). * When positioning with a high speed start command 1 (52h), the ramping-down point is fixed to the manual setting, regardless of the setting for MSDP (bit 13) in the PRMD. If the ramping-down point setting (PRDP) is zero, the axis will stop immediately.
High speed operation 2) 	1) Write high speed command 2 (53h). 2) Start deceleration by writing a deceleration stop command (4Ah). * When the deceleration stop command (49h) is written to the register, motion of an axis starts deceleration.	1) Write high speed start command 2 (53h). 2) Start deceleration when a ramping-down point is reached or by writing a deceleration stop command (4Ah). * If the ramping-down point is set to manual (MSDP = 1 in the PRMD), and the ramping-down value (PRDP) is zero, the axis will stop immediately.

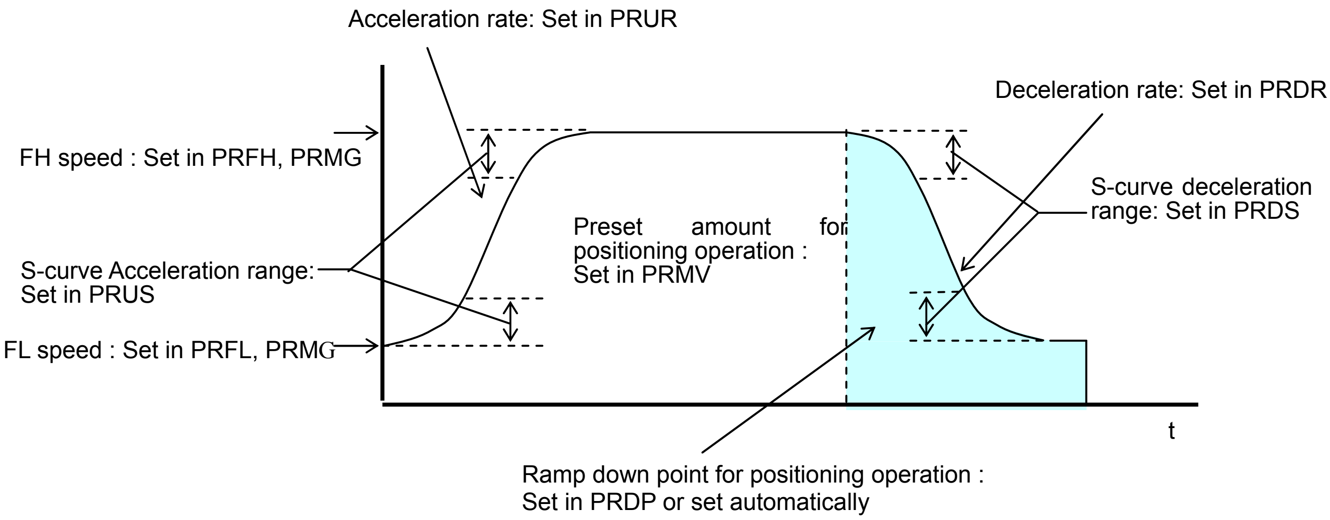
10-2. Speed pattern settings

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

Pre-register	Description	Bit length setting range	Setting range	register
PRMV	Positioning amount	32	-2,147,483,648 to +2,147,483,647 (80000000h) (7FFFFFFFh)	RMV
PRFL	Initial speed	16	1 to 65,535 (0FFFFh)	RFL
PRFH	Operation speed	16	1 to 65,535 (0FFFFh)	RFH
PRUR	Acceleration rate	16	1 to 65,535 (0FFFFh)	RUR
PRDR	Deceleration rate Note 1	16	0 to 65,535 (0FFFFh)	RDR
PRMG	Speed magnification rate	12	2 to 4,095 (0FFFh)	RMG
PRDP	Ramping-down point	24	0 to 16,777,215 (0FFFFFFh)	RDP
PRUS	S-curve acceleration range	15	0 to 32,767 (7FFFh)	RUS
PRDS	S-curve deceleration range	15	0 to 32,767 (7FFFh)	RDS

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

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



- PRFL: FL speed setting register (16-bit)
  Specify the speed for FL constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 65,535 (0FFFFh).
  The speed will be calculated from the value in PRMG.

$$\text{FL speed [pps]} = \text{PRFL x} \frac{\text{Reference clock frequency [Hz]}}{(\text{PRMG}+1)\times 65536}$$

- PRFH: FH speed setting register (16-bit)
  Specify the speed for FH constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 65,535 (0FFFFh).
  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.

$$\text{FH speed [pps]} = \text{PRFH x} \frac{\text{Reference clock frequency [Hz]}}{(\text{PRMG}+1)\times 65536}$$

◆ PRUR: Acceleration rate setting register (16-bit)

Specify the acceleration characteristic for high speed operations (acceleration/deceleration operations), in the range of 1 to 65,535 (0FFFFh)
Relationship between the value entered and the acceleration time will be as follows:

1) Linear acceleration (PRMD.MSMD = 0)

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

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

$$\text{Acceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRUR} + 1) \times 8}{\text{Reference clock frequency [Hz]}}$$

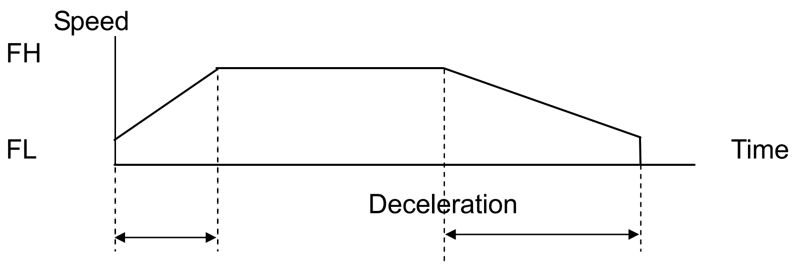
3) S-curve acceleration with a linear range (PRMD.MSMD= and PRUS register >0)

$$\text{Acceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL} + 2 \times \text{PRUS}) \times (\text{PRUR} + 1) \times 4}{\text{Reference clock frequency [Hz]}}$$

◆ PRDR: Deceleration rate setting register (16-bit)

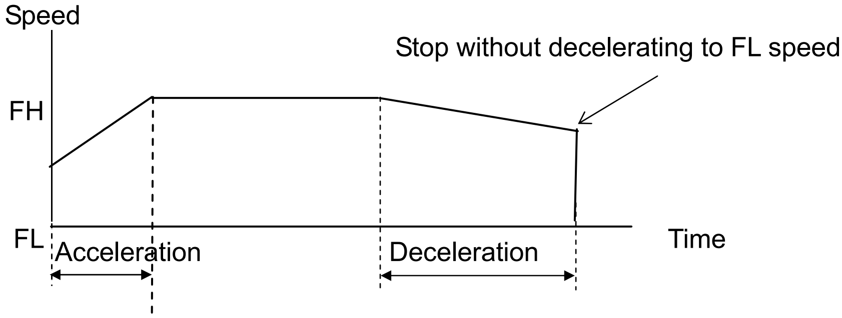
Normally, specify the deceleration characteristics for high speed operations (acceleration/deceleration operations) in the range of 1 to 65,535 (0FFFFh).
Even if the ramping-down point is set to automatic (PRMD.MSDP = 0), the value placed in the PRDR register will be used as the deceleration rate.
However, when PRDR = 0, the deceleration rate will be the value placed in the PRUR.
When the ramping-down point is set to automatic, there are the following restrictions.
While in linear interpolation 1 or circular interpolation operation, and when constant synthesized speed operation (PRMD.MIPF = 1 is selected, make deceleration time same as acceleration time.
For other operations, arrange time so that (deceleration time) ≤ acceleration time x 2.
If setting otherwise, the axis may not decrease the speed to the specified FL speed when stopping. In this case, use a manual ramping-down point (PRMD.MSDP = 1).

< When (deceleration time) ≤ (acceleration time x 2) using an automatic ramping-down point >



<When (deceleration time) > (acceleration time x 2) using an automatic ramping-down point>

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



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

1) Linear deceleration (PRMD.MSMD = 0)

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

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

$$\text{Deceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRDR} + 1) \times 8}{\text{Reference clock frequency [Hz]}}$$

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

$$\text{Deceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL} + 2 \times \text{PRDS}) \times (\text{PRDR} + 1) \times 4}{\text{Reference clock frequency [Hz]}}$$

◆ PRMG: Magnification rate register (12-bit)

Specify the relationship between the PRFL and PRFH settings and the speed, in the range of 2 to 4,095 (0FFFh). As the magnification rate is increased, the speed setting units will tend to be approximations. Normally set the magnification rate as low as possible.

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

$$\text{Magnification rate} = \frac{\text{Referenceclockfrequency[Hz]}}{(\text{PRMG} + 1) \times 65536}$$

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

Setting	Magnification rate	Output speed range	Setting	Magnification rate	Output speed range
2999 (0BB7h)	0.1	0.1 to 6,553.5	59 (3Bh)	5	5 to 327,675
1499 (5DBh)	0.2	0.2 to 13,107.0	29 (1Dh)	10	10 to 655,350
599 (257h)	0.5	0.5 to 32,767.5	14 (0Eh)	20	20 to 1,310,700
299 (12Bh)	1	1 to 65,535	5 (5h)	50	50 to 3,276,750
149 (95h)	2	2 to 131,070	2 (2h)	100	100 to 6,553,500

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

Specify the value used to determine the deceleration start point for positioning operations that include acceleration and deceleration.

The meaning of the value specified in the PRDP varies according to the "ramping-down point setting method", (MSDP) in the PRMD register.

<When set to manual (PRMD.MSDP=1)>

The number of pulses at which to start deceleration, set in the range of 0 to16,777,215 (0FFFFFFh).

The optimum value for the ramping-down point can be calculated as shown in the equation below.

1) Linear deceleration (PRMD.MSMD=0)

$$\text{Optimum value [Number of pulses]} = \frac{(\text{PRFH}^2 - \text{PRFL}^2) \times (\text{PRDR} + 1)}{(\text{PRMG} + 1) \times 32768}$$

However, the optimum value for a triangle start, without changing the value in the PRFH register while turning OFF the FH correction function (MADJ = 1 in the PRMD register) will be calculated as shown the equation below.

(When using idling control, assign the value (subtracts the number of idling pulses from the value place in the PRMV register) to PRMV in the equation below. The number of idling pulses will be "1 to 6" when IDL = 2 to 7 in RENV5.)

$$\text{Optimum value [Number of pulses]} = \frac{\text{PRMV} \times (\text{PRDR} + 1)}{\text{PRUR} + \text{PRDR} + 2}$$

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

$$\text{Optimum value [Number of pulses]} = \frac{(PRFH^2 - PRFL^2) \times (PRDR + 1) \times 2}{(PRMG + 1) \times 32768}$$

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

$$\text{Optimum value [Number of pulses]} = \frac{(PRFH + PRFL) \times (PRFH - PRFL + 2 \times PRDS) \times (PRDR + 1)}{(PRMG + 1) \times +32768}$$

Start deceleration at the point when the (positioning counter value) ≤ (PRDP set value).

<When set to automatic (PRMD.MSDP = 0)>
This is an offset value for the automatically set ramping-down point. Set in the range of -8,388,608 (800000h) to 8,388,607 (7FFFFFFh).
When the offset value is a positive number, the axis will start deceleration at an earlier stage and will feed at the FL speed after decelerating. When a negative number is entered, the deceleration start timing will be delayed. If the offset is not required, set to zero.

When the value for the ramping-down point is smaller than the optimum value, the speed when stopping will be faster than the FL speed. On the other hand, if it is larger than the optimum value, the axis will feed at FL constant speed after decelerating is complete.

◆ PRUS: S-curve acceleration range register (15-bit)
Specify the S-curve acceleration range for S-curve acceleration/deceleration operations in the range of 1 to 32,767 (7FFFh).
The S-curve acceleration range S_{SU} will be calculated from the value placed in PRMG.

$$S_{SU}\text{ [pps]} = PRUS \times \frac{\text{Reference clock frequency[Hz]}}{(PRMG + 1) \times 65536}$$

In other words, speeds between the FL speed and (FL speed + S_{SU}), and between (FH speed - S_{SU}) and the FH speed, will be S-curve acceleration operations. Intermediate speeds will use linear acceleration.
However, if zero is specified, "(PRFH - PRFL)/2" will be used for internal calculations, and the operation will be an S-curve acceleration without a linear component.

◆ PRDS: S-curve deceleration range setting register (15-bit)
Specify the S-curve deceleration range for S-curve acceleration/deceleration operations in the range of 1 to 32,767 (7FFFh).
The S-curve acceleration range S_{SD} will be calculated from the value placed in PRMG.

$$S_{SD}\text{ [pps]} = PRDS \times \frac{\text{Reference clock frequency[Hz]}}{(PRMG + 1) \times 65536}$$

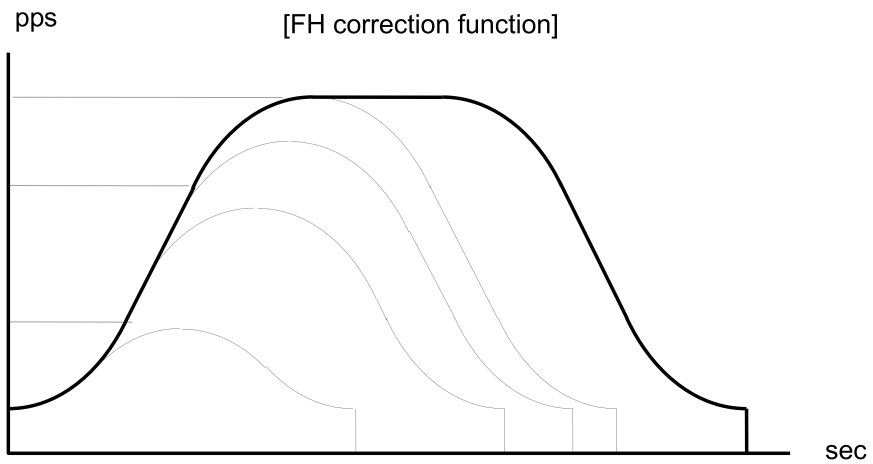
In other words, speeds between the FH speed and (FH speed - S_{SD}), and between (FL speed + S_{SD}) and the FL speed, will be S-curve deceleration operations. Intermediate speeds will use linear deceleration.
However, if zero is specified, "(PRFH - PRFL)/2" will be used for internal calculations, and the operation will be an S-curve deceleration without a linear component.

10-3. Manual FH correction

When the FH correction function is turned ON (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 to eliminate triangle driving.
However, if values in the PRUR and PRDR registers are set so that the (deceleration time) > (acceleration time x 2), do not use the FH correction function.

In order to eliminate triangle driving without using the FH correction function (PRMD.MADJ = 1), lower the FH speed before starting the acceleration/deceleration operation.
When using idling control, enter a value for PRMV in the equation below after deducting the number of idling pulses. The number of idling pulses will be 1 to 6 whenRENV5. IDL = 2 to 7.

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



Automatic correction of the maximum speed for changing the feed amount

< To execute FH correction manually>

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

$$\text{PRMV} \leq \frac{(\text{PRFH}^2 - \text{PRFL}^2) \times (\text{PRUR} + \text{PRDR} + 2)}{(\text{PRMG} + 1) \times 32768}$$

$$\text{PRFH} \leq \sqrt{\frac{(\text{PRMG} + 1) \times 32768 \times \text{PRMV}}{\text{PRUR} + \text{PRDR} + 2} + \text{PRFL}^2}$$

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

$$\text{PRMV} \leq \frac{(\text{PRFH}^2 - \text{PRFL}^2) \times (\text{PRUR} + \text{PRDR} + 2) \times 2}{(\text{PRMG} + 1) \times 32768}$$

$$\text{PRFH} \leq \sqrt{\frac{(\text{PRMG} + 1) \times 32768 \times \text{PRMV}}{(\text{PRUR} + \text{PRDR} + 2) \times 2} + \text{PRFL}^2}$$

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

(i) Make a linear acceleration range smaller
When

$$\text{PRMV} \leq \frac{(\text{PRFH} + \text{PRFL}) \times (\text{PRFH} - \text{PRFL} + 2 \times \text{PRUS}) \times (\text{PRUR} + \text{PRDR} + 2)}{(\text{PRMG} + 1) \times 32768} \text{ and}$$

$$\text{PRMV} > \frac{(\text{PRUS} + \text{PRFL}) \times \text{PRUS} \times (\text{PRUR} + \text{PRDR} + 2) \times 8}{(\text{PRMG} + 1) \times 32768}$$

$$\text{PRFH} \leq -\text{PRUS} + \sqrt{(\text{PRUS} - \text{PRFL})^2 + \frac{(\text{PRMG} + 1) \times 32768 \times \text{PRMV}}{(\text{PRUR} + \text{PRDR} + 2)}}$$

(ii) Eliminate the linear acceleration/deceleration range
When

$$\text{PRMV} \leq \frac{(\text{PRUS} + \text{PRFL}) \times \text{PRUS} \times (\text{PRUR} + \text{PRDR} + 2) \times 8}{(\text{PRMG} + 1) \times 32768}$$

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

$$\text{PRFH} \leq \sqrt{\frac{(\text{PRMG} + 1) \times 32768 \times \text{PRMV}}{(\text{PRUR} + \text{PRDR} + 2) \times 2} + \text{PRFL}^2}$$

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

$$PRMV \le \frac{(PRFH+PRFL)\times\{(PRFH-PRFL)\times(PRUR+PRDR+2)+2\times PRUS\times(PRUR+1)+2\times(PRDR+1)\}}{(PRMG+1)\times 32768}$$

and

$$PRMV > \frac{(PRDS+PRFL)\times\{PRDS\times(PRUR+2\times PRDR+3)+PRUS\times(PRUR+1)\}\times 4}{(PRMG+1)\times 32768}$$

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

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

B= {(PRMG + 1) x 32768 x PRMV - 2 x A x PRFL + (PRUR + PRDR + 2) x $PRFL^2$} x (PRUR + PRDR + 2)

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

When

$$PRMV \le \frac{(PRDS+PRFL)\times\{PRDS\times(PRUR+2\times PRDR+3)+PRUS\times(PRUR+1)\}\times 4}{(PRMG+1)\times 32768}$$

$$PRMV > \frac{(PRUS+PRFL)\times PRUS\times(PRUR+PRDR+2)\times 8}{(PRMG+1)\times 32768}$$

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

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

However, A = PRUS x (PRUR + 1) and

B={(PRMG+1)x32768xPRMV-2xAxPRFL+(PRUR+2xPRDR+3)x$PRFL^2$}x(PRUR+2xPRDR+3)

(iii) Eliminate the linear acceleration/deceleration range

When

$$PRMV \le \frac{(PRUS+PRFL)\times PRUS\times(PRUR+PRDR+2)\times 8}{(PRMG+1)\times 32768}$$

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

$$PRFH \le \sqrt{\frac{(PRMG+1)\times 32768\times PRMV}{(PRUR+PRDR+2)\times 2}+PRFL^2}$$

PRMV: Positioning amount	PRFL: Initial speed	PRFH: Operation speed
PRUR: Speed 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≤

$$\frac{(PRFH + PRFL) \times \{(PRFH - PRFL) \times (PRUR + PRDR + 2) + 2 \times PRUS \times (PRUR + 1) + 2 \times PRDS \times (PRDR + 1\}}{(PRMG + 1) \times 32768}$$

and

$$PRMV > \frac{(PRUS + PRFL) \times \{PRUS \times (2 \times PRUR + PRDR + 3) + PRDS \times (PRDR + 1) \times 4\}}{(PRMG + 1) \times 32768}$$

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

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

B= {(PRMG + 1) x 32768 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

$$PRMV \le \frac{(PRUS + PRFL) \times \{PRUS \times (2 \times PRUR + PRDR + 3) + PRDS \times (PRDR + 1) \times 4\}}{(PRMG + 1) \times 32768}$$ and

$$PRMV > \frac{(PRDS + PRFL) \times PRDS \times (PRUR + PRDR + 2) \times 8}{(PRMG + 1) \times 32768}$$

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

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

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

B= {(PRMG + 1) x 32768 x PRMV - 2 x A x PRFL + (2 x PRUR + PRDR + 3) x $PRFL^2$} x (2 x PRUR + PRDR + 3)

(iii) Eliminate the linear acceleration/deceleration range

When

$$PRMV \le \frac{(PRDS + PRFL) \times PRDS \times (PRUR + PRDR + 2) \times 8}{(PRMG + 1) \times 32768}$$

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

$$PRFH \le \sqrt{\frac{(PRMG + 1) \times 32768 \times PRMV}{(PRUR + PRDR + 2) \times 2} + PRFL^2}$$

PRMV: Positioning amount
PRFL: Initial speed
PRFH: Operation speed
PRUR: Operation speed 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, and the accel/decel time = 300 msec,

1) Select the 2x mode for multiplier rate in order to get 100 kpps output
   PRMG = 149 (95h)

2) Since the 2x mode is selected to get an operation speed 100 kpps,
   PRFH = 50000 (C350h)

3) In order to set a start speed of 10 pps, the rate magnification is set to the 2x mode.
   PRFL = 5 (0005h)

4) In order to make the acceleration/deceleration time 300 msec, set PRUR = 28,494, from the equation for the acceleration time and the PRUR value.

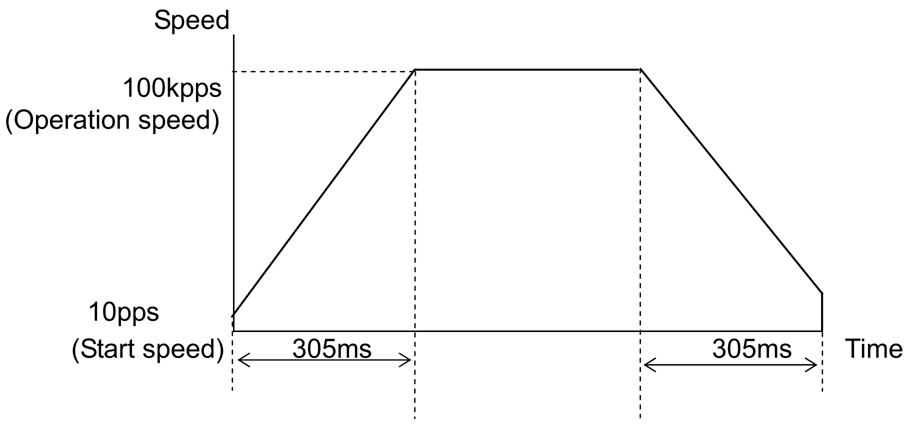
$$\text{Acceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRUR} + 1) \times 4}{\text{Reference clock frequency [Hz]}}$$

$$0.3 = \frac{(50000 - 5) \times (\text{PRUR} + 1) \times 4}{19.6608 \times 10^6}$$

PRUR = 28.494

However, since only integers can be entered for PRUR, use 28 or 29. The actual acceleration/deceleration time will be 295 msec if PRUR = 28, or 305 msec if PRUR = 29.

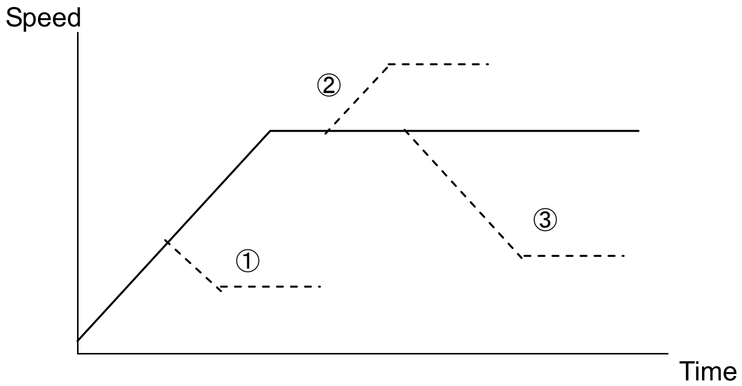
An example of the speed pattern when PRUR = 29



10-5. Changing speed patterns while in operation

By changing the RFH, RUR, RDR, RUS, or RDS registers during operation, the speed and acceleration can be changed on the fly. However, if the ramping-down point is set to automatic (MSDP = 0 in the PRDM register) for the positioning mode, do not change the values for RFL, RUR, RDR, RUS, or RDS. Otherwise, the automatic ramping-down point function will not work correctly.

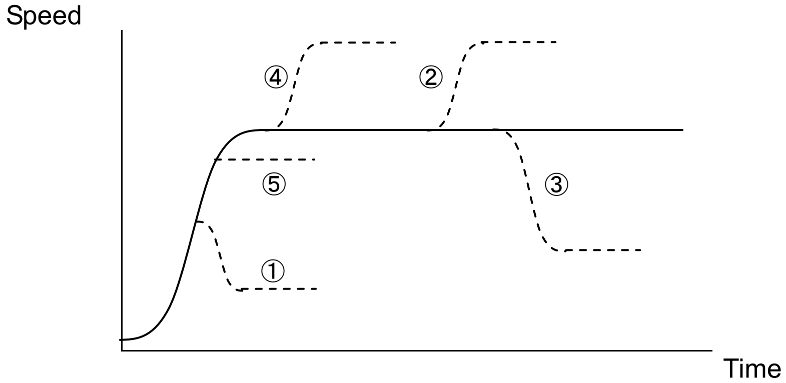
An example of changing the speed pattern by changing the speed, during a linear acceleration/deceleration operation



1) Make RFH smaller while accelerating the axis accelerate or decelerate until it reaches the correct speed.

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

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



1) Make RFH smaller and if ((change speed) < (speed before change)) and the axis will decelerate using an S-curve until it reaches the correct speed.

5) Make RFH smaller and if ((change speed) ≥ (speed before change)) and the axis will accelerate without changing the S-curve's characteristic until it reaches the correct speed.

4) Make RFH larger while accelerating and the axis will accelerate to the original speed entered without changing the S-curve's characteristic. Then it will accelerate again until it reaches the newly set speed.

2), 3) If RFH is changed after the acceleration/deceleration is complete, the axis will accelerate/ decelerate using an S-curve until it reaches the correct speed.

11. Description of the Functions

11-1. Reset

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

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

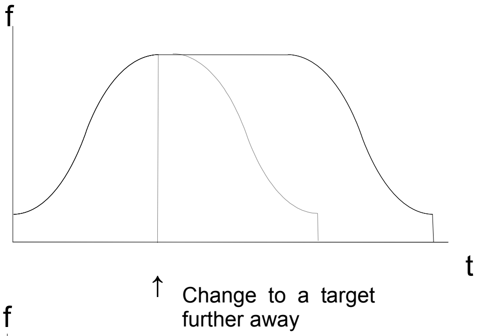
11-2. Position override

This LSI can override (change) the target position freely during operation.
There are two methods for overriding the target position.

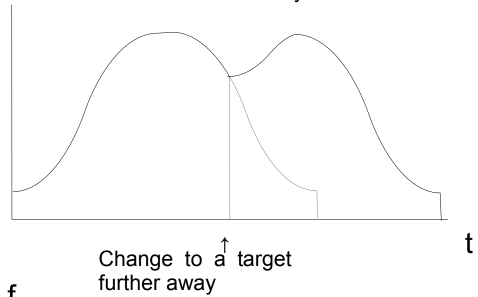
11-2-1. Target position override 1

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

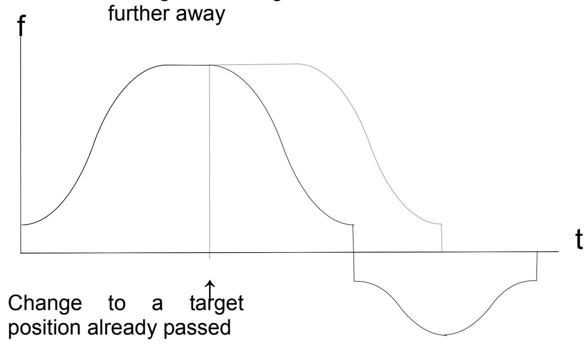
1) If the new target position is further away from the original target position during acceleration or constant speed operation, the axis will maintain the operation using the same speed pattern and it will complete the positioning operation at the position specified in the new data (new RMV value).



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

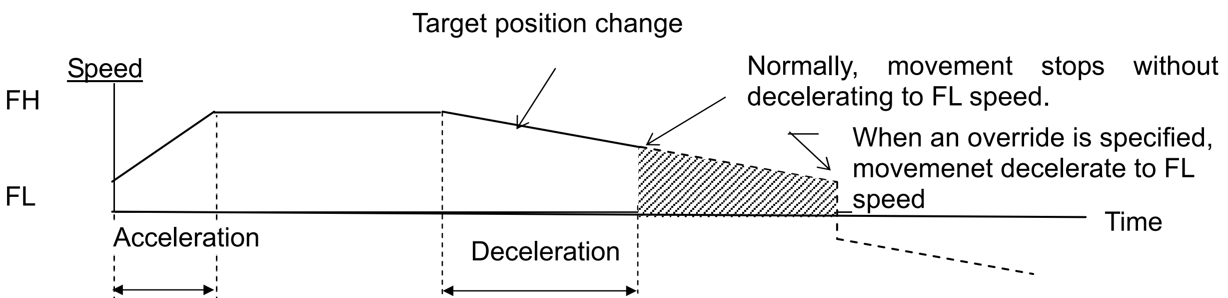


3) If the axis has already passed over the new target position, or the target position is changed to a position that is closer than the original position during deceleration, movement on the axis will decelerate and stop. Then, the movement will reverse and complete the positioning operation at the position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



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

Note1: If the ramping-down point is set to automatic and the (deceleration time) > (acceleration time x 2), it may be the case that the axis cannot reduce the speed to the FL level, as shown below. In this case, if the target position is set closer than original position and the axis is decelerating, the axis will decelerate along the deceleration curve from the new override position, and then slow to the FL speed and finally stop. Then it will start moving to the new position.
Therefore, the axis will overrun the original target position during deceleration (shaded area).



To avoid creating an overrun condition, make sure that the deceleration time is less than two times of the acceleration time, or if the deceleration time is more than double the acceleration time, make the ramping-down point a manual setting.

Note 2: If the LSI starts decelerating by changing the target to a close position, the LSI will not re-accelerate even if you perform a "position override" to a position further away again during this deceleration. It will feed to the more distant target after decelerating to FL speed. Also, if you override the target position to lower than the initial RMV setting value during decelerating using the automatic ramp down point setting, the LSI will not accelerate using the target position override again.
If you change the target position to a further position with the "position override" function while decelerating with the auto ramp down function, the LSI will accelerate again.

Note 3: The position override is only valid while feeding.
If you perform a “position override” operation just before stopping, the PCL may not accept the position override command. To see if the position override command is accepted, check the SEOR bit in the main status after issuing the override command. If the PCL has ignored the override command, the SEOR will be 1.
Please note if an override command is written into the RMV register (90h) while the axis is stopping, the PCL changes SEOR to 1. Therefore, if you write an override command before the axis has started moving, the SEOR will also be changed to1.
If the PCL ignores the override, the SEOR will become 1 when the axis stops. And, after MSTSW is read, SEOR will go back to 0 within 3 reference clock cycles.

Note 4: A Position Override 1 cannot be executed while performing an interpolation operation.

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

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

Setting pulse control using the PCS input <Set 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 <Set 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

Note: A Position Override 2 cannot be executed while performing an interpolation operation.

11-3. Output pulse control

11-3-1. Output pulse mode

There are four types of common pulse output modes, two types of Two-pulse modes and two types of 90˚ phase difference modes as the modes to output command pulses.

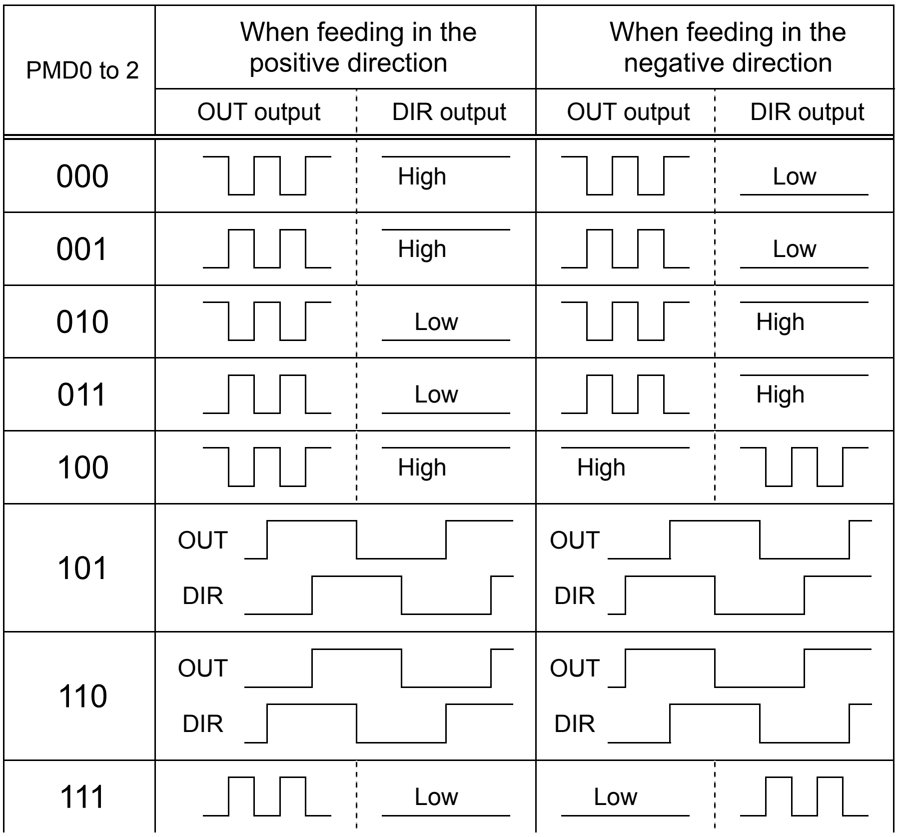
Common pulse mode:	Outputs operation pulses from the OUT terminal and outputs the direction signal from the DIR terminal.
Two-pulse mode:	Outputs positive direction operation pulses from the OUT terminal, and outputs negative direction operation pulses from the DIR terminal.
90˚ phase difference modes:	Outputs 90˚ phase difference pulses through the OUT and DIR terminals.

The output mode for command pulses is set in RENV1.PMD0 to 2 (bits 0 to 2).
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 (bit 28) is set to 0, the operation can be delayed for one direction change timer unit (0.2 msec), after changing the direction identification signal.

Setting the pulse output mode <Set RENV1.PMD0 to 2 (bit0 to 2)>

[RENV1] (WRITE)

7							0
-	-	-	-	-	n	n	n



Setting the direction change timer (0.2 msec) functi
<Set RENV1.DTMF(bit 28)>

0: ON
1: OFF

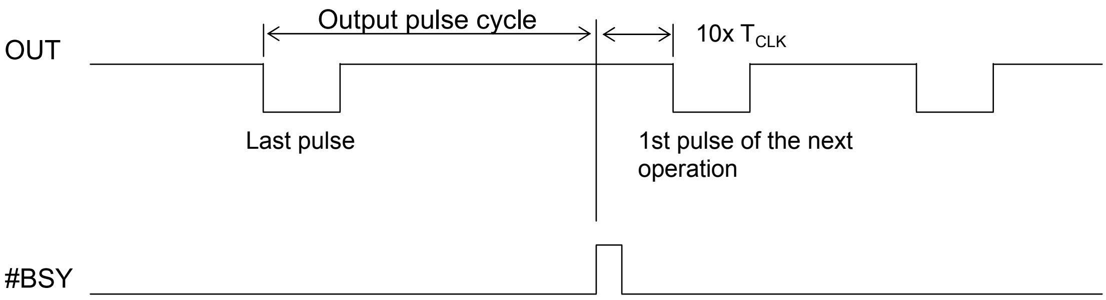
[RENV1] (WRITE)

31							24
-	-	-	n	-	-	-	-

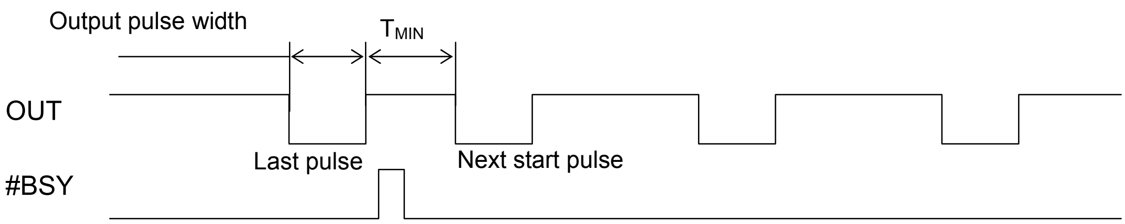
11-3-2. Control the output pulse width and operation complete timing

In order to put forward the timing of stopping, this LSI controls the output pulse width.
When the output pulse speed is slower than 1/8192 of reference clock (approx. 2.4 Kpps when CLK = 19.6608 MHz), the pulse width is constant and is 4096 cycles of the reference clock (approx. 200 μsec when CLK = 19.6608 MHz). For faster pulse speeds than this, the duty cycle is kept constant (approx. 50%). By setting RENV1.PDTC (bit 31), the output pulse width can be fixed to make a constant duty cycle (50%).
Also, when setting PRMD.METM (operation completion timing setting), 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)



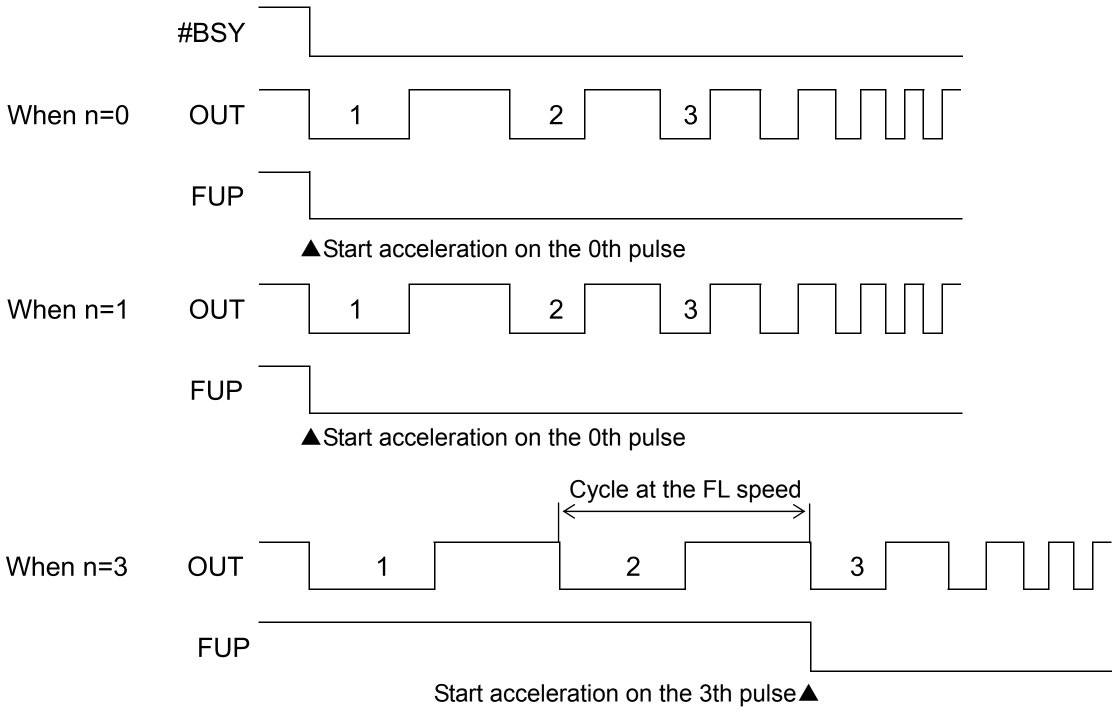
When set to "when the output pulse is OFF," the time interval "Min" from the last pulse until the next starting pulse output will be $T_{MIN} = 15 \times T_{CLK}$. ($T_{CLK}$: Reference clock cycle)

Setting the operation complete timing <Set PRMD.METM (bit 12)> 0: At the end of a cycle of a particular output frequency 1: When the output pulse turns OFF.	[PRMD] (WRITE) 15 8 - - - n - - - -
Setting the output pulse width <Set RENV1.PDTC (bit 31)> 0: Automatically change between a constant output pulse and a constant duty cycle (approx. 50%) in accord with variations in speed. 1: Keep the output pulse width at a constant duty cycle (approx. 50%).	[RENV1] (WRITE) 31 24 n - - - - - - -

11-4. Idling control

When starting acceleration or deceleration operation, it can be started after the output of a few pulses at FL speed (idling output). Set the number of pulses for idling in RENV5.IDL.
If you will not be using this function, enter a value "n" of 0 or 1. The LSI will start acceleration at the same time it begins outputting pulses. Therefore, the start speed obtained from an initial 2-pulse cycle will be faster than the FL speed.
To use this function, enter a value "n" of 2 to 7. The LSI will start the acceleration by beginning its output on the "n" th pulse. Therefore, the start speed will be the FL speed and the FL speed can be set to near the maximum starting pulse rate.
If this function is used with the positioning mode, the total feed amount will not change.

[Setting idling pulses and acceleration start timing]



Set the number of idling pulses <Set RENV5.IDL0 to 2 (bits 8 to 10)> Specify the number of idling pulses, from 0 to 7. Start accelerating at FL speed after outputting the specified number of pulses.	[RENV5] (WRITE) 15 8 - - - - - n n n
Read the idling control counter value <RSPD. IDC0 to 2 (bits 20 to 22)> Read the idling control counter.	[RSPD] (READ) 23 16 0 n n n - - - -

Note: While setting the number of idling pulses, when you write a High-Speed Start 1 command (52h or 56h), motion of an axis will accelerate to FH speed after outputting the specified number of idling pulses at FL speed. Then the operation will be the same as the High-Speed Start 2 command.

11-5. Mechanical external input control

11-5-1. +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 even if the machine moves.
If the EL signal is ON when writing a start command, the axis cannot start moving in the direction of the particular EL signal that is ON.
By setting RENV1.ELM, the stopping pattern for use when the EL signal is turned ON can be set to immediate stop or deceleration stop (high speed start only). If deceleration stop is selected, hold the EL input ON until stopping.
The minimum pulse width of the EL signal is 80 reference clock cycles (4 μsec) when the input filter is ON. When the input filter is turned OFF, the minimum pulse width is two reference clock cycles (0.1 μsec).
The EL signal can be monitored by reading SSTSW (sub status).
By reading the REST register, you can check for an error interrupt caused by the EL signal turning ON.
When in the timer mode, this signal is ignored. Even in this case, the EL signal can be monitored by reading SSTSW (sub status).
The input logic of the EL signal can be set for each axis using the ELL input terminal.

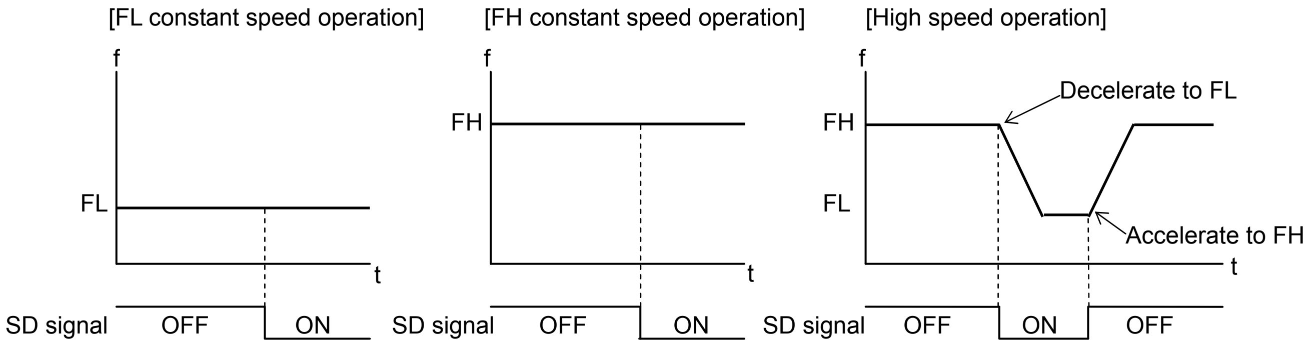
Set the input logic of the ±EL signal <ELL input terminal> L: Positive logic input H: Negative logic input	
Stop method used when the ±EL signal turns ON <Set REMV1.ELM (bit 3)> 0: Immediate stop by turning ON the ±EL signal 1: Deceleration stop by turning ON the ±EL signal	[RENV1] (WRITE) 7 0 - - - - n - - -
Reading the ±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 cause when the ±EL signal turns on <REST.ESPL (bit 5), ESML (bit 6)> ESPL = 1: Stop by turning ON the +EL signal ESML = 1: Stop by turning ON the -EL signal	[REST] (READ) 7 0 - n n - - - - -
Setting the ±EL input filter <Set RENV1.FLTR (bit 26)> 1:Apply a filter to the ±EL input Apply a filter and any signals shorter than 4 μsec pulse width are ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -

Note 1: Operation after turning ON the EL signal may be different from the above for the origin return operation (9-5-1), the origin search operation (9-5-3), and the EL or SL operation mode (9-6). See the description of each operation mode.

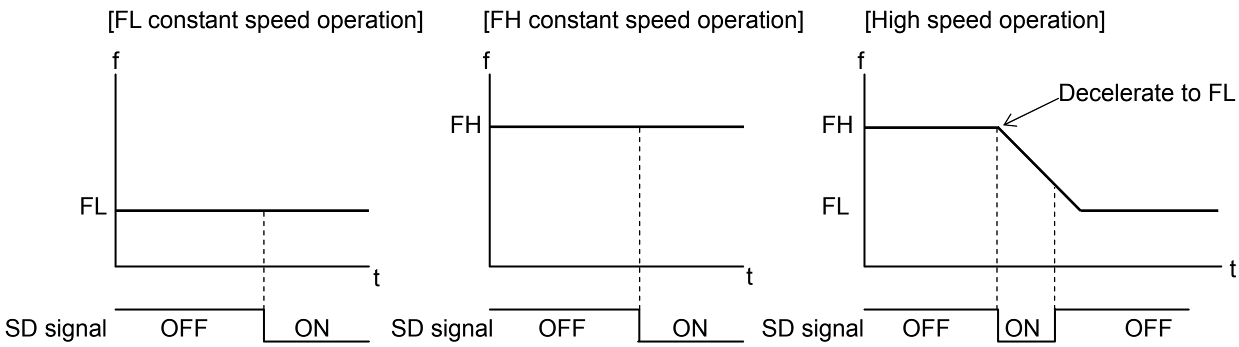
11-5-2. +SD signal, -SD signal

If the SD signal input is disabled by setting MSDE in the PRMD register (operation mode) to 0, the SD signal will be ignored.
If the SD signal is enabled and the SD signal is turned ON while in operation, the axis will: 1) decelerate, 2) latch and decelerate, 3) decelerate and stop, or 4) latch and perform a deceleration stop, according to the setting of SDM and SDLT in the RENV1 register (environment setting 1).

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

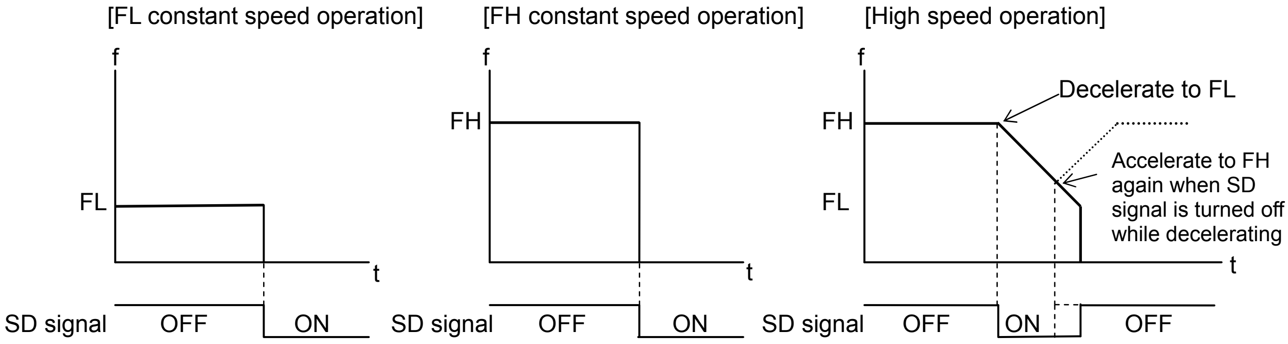


2) Latch and decelerate <RENV1.SDM (bit 4) = 0, RENV1.SDLT (bit 5) = 1>
- While feeding at constant speed, the SD signal is ignored. While in high speed operation, decelerate to FL speed by turning the SD signal ON. Even if the SD signal is turned OFF after decelerating or while decelerating, the axis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



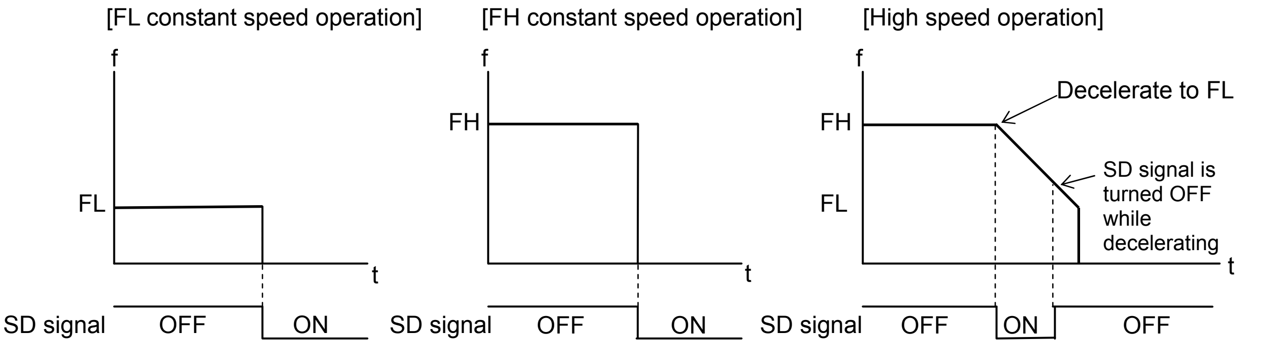
3) Deceleration stop <RENV1.SDM (bit 4) = 1, RENV1.SDLT (bit 5) = 0>

- If the SD signal is turned ON while in constant speed operation, the axis will stop. While in high speed operation, the axis will decelerate to FL speed when the SD signal is turned ON, and then stop. If the SD signal is turned OFF during deceleration, the axis will accelerate to FH speed.
- If the SD signal is turned ON after writing a start command, the axis will complete its operation without another start.
- When stopped, the axis will output an #INT signal.



4) Latch and deceleration stop <RENV1.SDM (bit 4) = 1, REMV1.SDLT (bit 5) = 1>

- If the SD signal is turned ON while in constant speed operation, the axis will stop. If the SD signal is turned ON while in high speed operation, the axis will decelerate to FL speed and then stop. Even if the SD signal is turned OFF during deceleration, the axis will not accelerate.
- If the SD signal is turned ON while writing a start command, the axis will not start moving and the operation will be completed.
- While stopped, the LSI outputs an #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, and if the SD signal is OFF at the next start, the latch will be reset. The latch is also reset when the latch input (RENV1.SDLT) is set to zero.

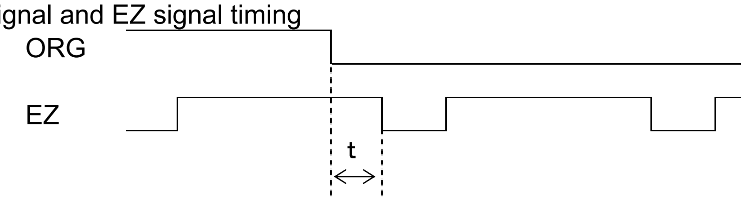
The minimum pulse width of the SD signal is 80 reference clock cycles (4.0 μsec) when the input filter is ON. When the input filter is turned OFF, the minimum pulse width is two reference clock cycles (0.1 μsec). (When CLK = 19.6608 MHz.)
The latch signal of the SD signal can be monitored by reading SSTSW (sub status). The SD signal terminal status can be monitored by reading RSTS (extension status). By reading the REST register, you can check for an error interrupt caused by the SD signal turning ON.

Enable/disable ±SD signal input <Set 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 <Set 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 <Set REMV1.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	[RENV1] (WRITE) 7 0 - - - n - - - -
Select the ±SD signal input type <Set REMV1.SDLT (bit 5)> 0: Level input 1: Latch input To release the latch, turn OFF the SD input when next start command is written or select Level input.	[RENV1] (WRITE) 7 0 - - n - - - - -
Reading the latch status of the ±SD signal <SSTSW.SSD (bit 15)> 0: The SD latch signal in operation direction is OFF 1: The SD latch signal in operation direction is ON	[SSTSW] (READ) 15 8 n - - - - - - -
Reading the ±SD signal < RSTS.PSDI (bit 15),RSTS.MSDI (bit 17), RSTS.PSDL(bit22),RSTS.MSDL(bit 23)> PSDI=0: +SD signal is OFF MSDI=0: -SD signal is OFF PSDI=1: +SD signal is ON MSDI=1: -SD signal is ON PSDL=0: +SD latch signal is OFF MSDL=0: -SD latch signal is OFF PSDL=1: +SD latch signal is ON MSDL=1: -SD latch signal is ON	[RSTS] (READ) 15 8 n - - - - - - - 23 16 n n - - - - n -
Reading the cause of an #INT when stopped by the ±SD signal <REST.ESSD (bit 10)> 1: Deceleration stop caused by the ±SD signal turning ON	[REST] (READ) 15 8 - - - - 0 n - -
Apply an input filter to ±SD <Set RENV1.FLTR (bit 26)> 1: Apply a filter to the ±SD input By applying a filter, signals with a pulse width of 4 μsec or less will be ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -

11-5-3. ORG, EZ signals

These signals are enabled in the origin return modes (origin return, leave origin position, and origin position search) and in the EZ count operation modes. Specify the operation mode and the operation direction using the PRMD register (operation mode).
Since the ORG signal input is latched internally, there is no need to keep the external signal ON.
The ORG latch signal is reset when stopped.
The minimum pulse width of the ORG signal is 80 reference clock cycles (4 μsec) when the input filter is ON. When the input filter is turned OFF, the minimum pulse width is two reference clock cycle (0.1 μsec). (When CLK = 19.6608 MHz.)
The input logic of the ORG signal and EZ signal can be changed using the RENV1 register (environment setting 1).
The ORG terminal status can be monitored by reading SSTSW (sub status). The EZ terminal status can be monitored by reading the RSTS register (extension status).
For details about the origin return operation modes, see 9-5, "Origin position operation mode."

ORG signal and EZ signal timing



(i) When $t \geq 2 \times T_{CLK}$, counts.
(ii) When $T_{CLK} < t < 2 \times T_{CLK}$, counting is undetermined.
(iii) When $t \leq T_{CLK}$, does not count.
T_{CLK}: Reference clock frequency

Enabling the ORG and EZ signals <Set RMD.MOD (bits 0 to 6)> 001 0000: Origin return in the positive direction 001 0010: Leave origin position in the positive direction 001 0101: Origin position search in the positive direction 010 0100: EZ counting in the positive direction 001 1000: Origin return in the negative direction 001 1010: Leave origin position in the negative direction 001 1101: Origin position search in the negative direction 010 1100: EZ count operation in the negative direction	[PRMD] (WRITE) 7 0 0 n n n n n n n
Set the origin return method <Set RENV3.ORM0 to 3 (bits 0 to 3)> See the RENV3 register description	[RENV3] (WRITE) 7 0 - - - - n n n n
Set the input logic for the ORG signal <Set ORGL (bit 7) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 n - - - - - - -
Read the ORG signal <SORG (bit 14) in SSTSW> 0: The ORG signal is OFF 1: The ORG signal is ON	[SSTSW] (READ) 15 8 - n - - - - - -
Set the EZ count number <Set RENV3.EZD0 to 3 (bits 4 to 7)> Set the origin return completion condition and the EZ count number for counting. Specify the value (the number to count – 1) in EZD0 to 3. The setting range is 0 to 15.	[RENV3] (WRITE) 7 0 n n n n - - - -
Specify the input logic of the EZ signal <Set RENV2.EZL> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 23 16 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 - -
Apply an input filter to EZ <Set RENV1.FLTR (bit 26)> 1: Apply a filter to the EZ input By applying a filter, signals with a pulse width of 4 μsec or less will be ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -

11-6. Servomotor I/F

11-6-1. INP signal

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

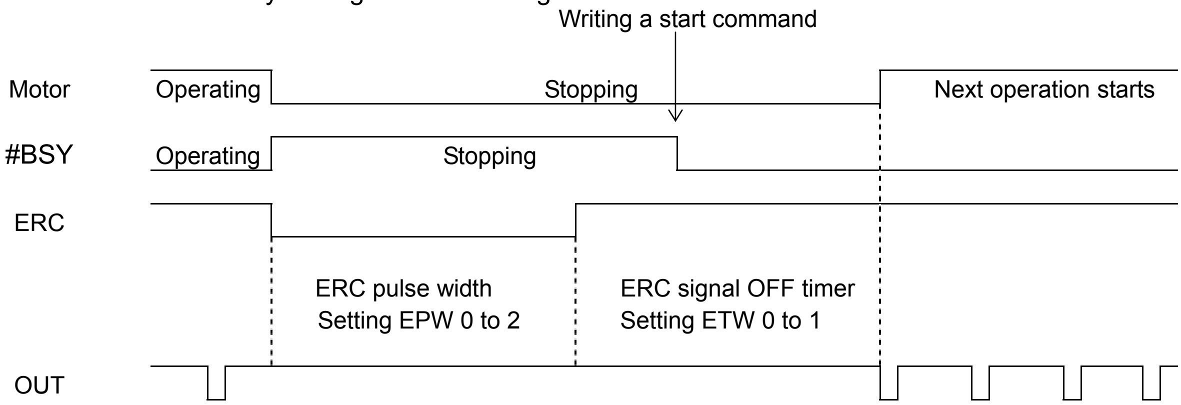
This LSI can receive a positioning complete signal (INP signal) from a servo driver in place of the pulse output complete timing, to determine when an operation is complete.
When the INP signal input is used to indicate the completion status of an operation, the #BSY signal when an operation is complete,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

The input logic of the INP signal can be changed.
The minimum pulse width of the INP signal is 80 reference clock cycles (4 μsec) when the input filter is ON. If the input filter is OFF, the minimum pulse width will be 2 reference clock cycles (0.1 μsec). (When CLK = 19.6608 MHz)
If the INP signal is already ON when the PCL is finished outputting pulses, it treats the operation as complete, without any delay.
The INP signal can be monitored by reading the RSTS register (extension status).

Set the operation complete delay using the INP signal <Set PRMD.MINP (bit 9)> 0: No operation complete delay waiting for the INP signal. 1: Operation complete (status, #BSY) delay until the INP signal turns ON.	[PRMD] (WRITE) 15 8 - - - - - - n -
Input logic of the INP signal <Set RENV1.INPL (bit 22)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 23 8 - 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 filter <RENV1.FLTR (bit 26)> 1: Apply a filter to the INP input. By applying a filter, pulses less than 4 μsec in width are ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -

11-6-2. ERC signal

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



In order to output an ERC signal at the completion of an origin return operation, set 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."
In order to output an ERC signal for an immediate stop based on the EL signal, ALM signal, or #CEMG signal input, or on the emergency stop command (05h), set RENV1.EROE (bit 10) = 1 and set automatic output for the ERC signal. (In the case of a deceleration stop, the ERC signal cannot be output, even when set for automatic output.)
The ERC signal can be output by writing an ERC output command (24h).
The output logic of the ERC signal can be changed by setting the RENV1 register. Read the RSTS (extension status) register to monitor the ERC signal.

Set automatic output for the ERC signal <Set RENV1.EROE (bit 10)> 0: Does not output an ERC signal when stopped by EL, ALM, or #CEMG input. 1: Automatically outputs an ERC signal when stopped by EL, ALM, or #CEMG input.	[RENV1] (WRITE) 15 8 - - - - - n - -
Set automatic output for the ERC signal <Set 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 <Set RENV1.EPW0 to 2 (bits 12 to 14)> 000: 12 μsec 100: 13 msec 001: 102 μsec 101: 52 msec 010: 408 μsec 110: 104 msec 011: 1.6 msec 111: Logic level output	[RENV1] (WRITE) 15 8 - n n n - - - -
Select output logic for the ERC signal <Set RENV1.ERCL (bit 15)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 15 8 n - - - - - - -
Specify the ERC signal OFF timer time <Set RENV1.ETW0 to 1 (bits 16 to 17)> 00: 0 μsec 10: 1.6 msec 01: 12 μsec 11: 104 msec	[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 an ERC signal	[Control command] 24h
ERC signal output reset command <ERCRST: Control command> Turn OFF an ERC signal	[Control command] 25h

11-6-3. ALM signals

Input alarm (ALM) signal.
When the ALM signal turns ON while in operation, the axis will stop immediately or decelerate and stop. When the axis is started at constant speed, the signal on the ALM terminal will cause an immediate stop.
However, the axis only decelerates and stops on an ALM signal if it was started with a high speed start.
To stop using deceleration, keep the ALM input ON until the axis 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 80 reference clock cycles (4 μsec) if the input filter is ON.
If the input filter is OFF, the minimum pulse width is 2 reference clock cycles (0.1 μsec). (When CLK = 19.6608 HMz.)
The input logic of the ALM signal can be changed. The signal status of the ALM signal can be monitored by reading SSTSW (sub status).

Stop method when the ALM signal is ON <SetnRENV1. ALMM (bit 8)> 0: Stop immediately when the ALM signal is turned ON 1: Deceleration stop (high speed start only) when the ALM signal is turned ON	[RENV1] (WRITE) 15 8 - - - - - - - n
Input logic setting of the ALM signal <Set 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 cause of a stop 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 filter <Set RENV1.FLTR (bit 26)> 1: Apply a filter to the ALM input When a filter is applied, pulses less than 4 μsec pulse in width will be ignored.	[RENV1] (WRITE) 31 24 - - - - - n - -

11-7. External start, simultaneous start

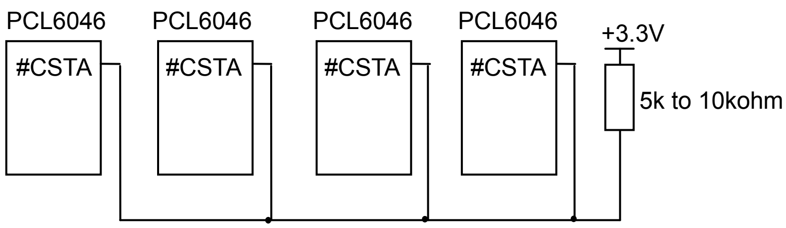
11-7-1. #CSTA signal

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

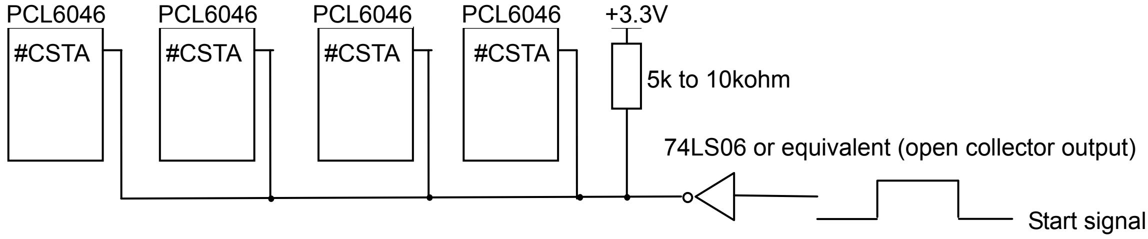
<How to make a simultaneous start>
Set PRMD.MSY0 to 1 (bits 18 to 19) for the axes you want to start. Write a start command and put the LSI in the "waiting for #CSTA input" status. Then, start the axes simultaneously by either of the methods described below.
1) By writing a simultaneous start command,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

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

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



2) To start simultaneously from an external circuit, connect the LSIs as follows.



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

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

11-7-2. PCS signal

The PCS input is a terminal originally used for the target position override 2. However, by setting the RENV1.PCSM (bit 30) to "1" and PRMD.MSY (bits 18 to 19) to "1", the PCS input signal can enable the #CSTA signal for only its own axis.
The input logic of the PCS input signal can be changed. The terminal status can be monitored by reading the RSTS register (extension status).

Specify the function of the PCS signal <Set RENV1.PCSM (bit 30)> 1: Make the PCS input a #CSTA signal that is available only for its own axis.	[RENV1] (WRITE) 31 24 - n - - - - - -
Set the Waiting for #CSTA input <Set RMD.MSY0 to 1 (bits 18 and 19)> 01: Start on a #CSTA input.	[RMD] (WRITE) 23 16 - - - - n n - -
Set the input logic of the PCS signal <Set 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

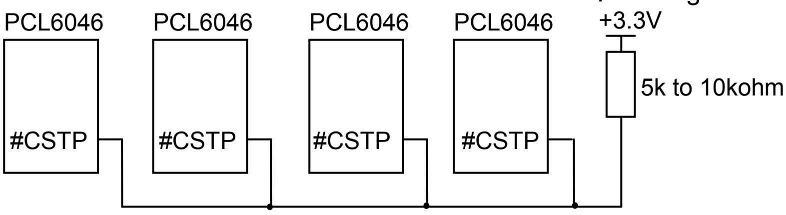
11-8. External stop / simultaneous stop

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

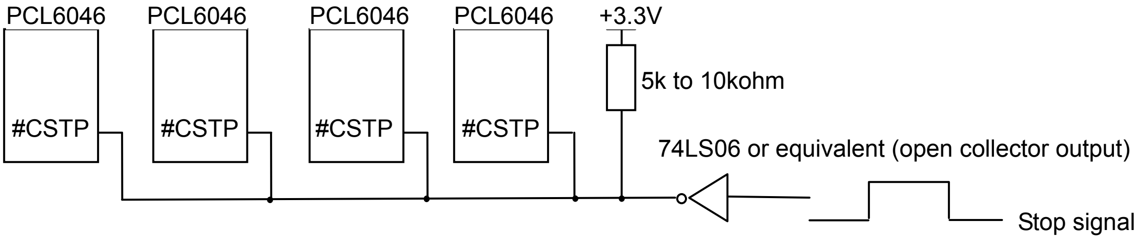
<How to make a simultaneous stop>
Set PRMD.MSPE (bit 24) = 1 for each of the axes that you want to stop simultaneously. Then start these axes. Stop these axes using any of the following three methods.
1) By writing a simultaneous stop command, the #CSTP terminal will output a one shot signal of 8 reference clock cycles in length (approx. 0.4 μsec when CLK = 19.6608 MHz).
2) Supply an external hardware signal
Supply a hardware signal using an open collector output (74LS06 or equivalent).
3) The #CSTP terminal will output a one shot signal of 8 reference clock cycles (approximately 0.4 μsec when CLK = 19.6608 MHz) when a stop caused by an error occurs on an axis that has PRMD.MSPO = 1.

Even when the #CSTP terminals on LSIs are connected together, each axis can still be stopped independently by using the stop command.

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



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



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

Setting to enable #CSTP input <Set PRMD.MSPE (bit 24)> 1. Enable a stop from the #CSTP input. (Immediate stop, deceleration stop)	[PRMD] (WRITE) 31 24 0 0 0 0 - - - n
Auto output setting for the #CSTP signal <Set to PRMD.MSPO (bit 25)> 1: When an axis stops because of an error, the PCL will output the #CSTP signal automatically. (Output signal width: 8 reference clock cycles)	[PRMD] (WRITE) 31 24 0 0 0 0 - - n -
Specify the stop method to use when the #CSTP signal is turned ON. <Set RENV1.STPM (bit 19)> 0: Immediate stop when the #CSTP signal is turned ON. 1: Deceleration stop when the #CSTP signal is turned ON.	[RENV1] (WRITE) 23 16 - - - - n - - -
Read the #CSTP signal <RSTS.SSTP (bit 6)> 0: The #CSTP signal is OFF 1: The #CSTP signal is ON	[RSTS] (READ) 7 0 - n - - - - - -
Read the cause of an error input < REST.ESSP (bit 8)> 1. When stopped because the #CSTP signal turned ON.	[REST] (READ) 15 8 - - - - - - - n
Simultaneous stop command <CMSTP: Operation command> Outputs a one shot pulse of 8 reference clock cycles in length from the #CSTP terminal. (The #CSTP terminal is bi-directional. It can input the output signal again.)	[Operation command] 07h

11-9. Emergency stop

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

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

Read the #CEMG signal <RSTS.SEMG (bit 7)> 0: The #CEMG signal is OFF 1: The #CEMG signal is ON	[RSTS] (READ) 7 0 n - - - - - - -
Read the cause of an error interrupt <REST.ESEM (bit 9)> 1. Stopped when the #CEMG signal is turned ON.	[REST] (READ) 15 8 - - - - - - n -
Emergency stop command <CMEMG: Operation command> The operation is the same as when a #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 glitch. Motor drivers do not recognize glitch pulses, and therefore only the PCL internal 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. Counter

11-10-1. Counter type and input method

In addition to the positioning counter, this LSI contains four other counters. These counters offer the following functions.

- Control command position and mechanical position
- Detect a stepper motor that is "out of step" using COUNTER 3 (deflection counter) and a comparator.
- Output a synchronous signal using COUNTER 4 (general-purpose) and a comparator.

The positioning counter is loaded with an absolute value for the RMV register (target position) at the start, regardless of the operation mode selected. It decreases the value with each pulse that is output. However, if PRMD.MPCS (bit 14) is set to 1 and while a position override 2 is executed, the counter does will not decrease until the PCS input is turned ON.
Input to COUNTER 1 is exclusively for output pulses. However COUNTERS 2 to 4 can be selected as follows by setting the RENV3 register (environment setting 3).

	COUNTER1	COUNTER2	COUNTER3	COUNTER4
Counter name	Command position	Mechanical position	Deflection	General-purpose
Counter type	Up/down counter	Up/down counter	Deflection counter	Up/down counter
Number of bits	32	32	16	32
Output pulse	Possible	Possible	Possible	Possible
Encoder (EA/EB) input	Not possible	Possible	Possible	Possible
Pulsar (PA/PB) input	Not possible	Possible	Possible	Possible
1/2 of reference clock	Not possible	Not possible	Not possible	Not possible

Note: When using pulsar input, use the internal signal result after multiplying or dividing.

Specify COUNTER 2 (mechanical position) input <RENV3.CI20 to 21 (bit 8 to 9)> 00: EA/EB input 01: Output pulses 10: PA/PB input	[RENV3] (WRITE) 15 8 - - - - - - n n
Set COUNTER 3 (deflection) input <RENV3.CI30 to 31 (bit 10 to 11)> 00: Measure the deflection between output pulses and EA/EB input 01: Measure the deflection between output pulses and PA/PB input 10: Measure the deflection between EA/EB input and PA/PB input	[RENV3] (WRITE) 15 8 - - - - n n - -
Set COUNTER 4 (general-purpose) input <RENV3.CI40 to 41 (bit 12 to 13)> 00: Output pulses 01: EA/EB input 10: PA/PB input 11: 1/2 of reference clock (CLK)	[RENV3] (WRITE) 15 8 - - n n - - - -

The EA/EB and PA/PB input terminal, that are used as inputs for the counter, can be set for one of two signal input types by setting the RENV2 (environment setting 2) register.

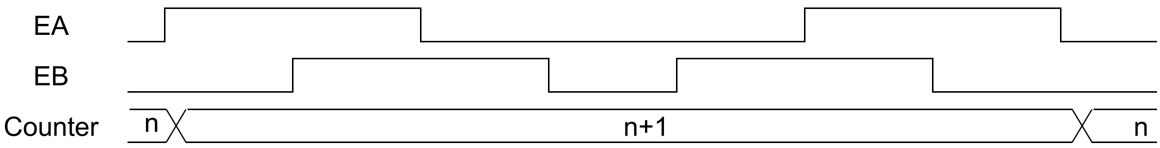
1) Signal input method: Input 90˚ phase difference signals (1x, 2x, 4x)
   Counter direction: Count up (count forward) when the EA input phase is leading. Count down when the EB input phase is leading.
2) Signal input method: Input count-up (count-forward) pulses or count-down pulses (Two-pulse input).
   Counter direction: Count up (count forward) on the rising edge of the EA input. Count down on the falling edge of the EB input.

The counter direction or EA/EB and PA/PB input signals can be reversed.
The LSI can be set to sense an error when both the EA and EB input, or both the PA and PB inputs change simultaneously, and this error can be detected using the REST (error interrupt cause) register.

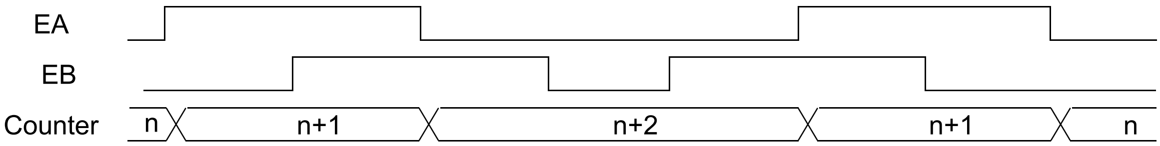
Set the input signal filter for EA/EB/EZ <Set RENV2.EINF (bit 18)> 0: Turn OFF the filter function 1: Turn ON the filter function (Input signals shorter than 3 reference clock cycles are ignored.)	[RENV2] (WRITE) 23 16 - - - - - n - -
Setting the EA/EB input <Set RENV2.EIM0 to 1 (bit 20 to 21)> 00: 90˚ phase difference, 1x 10: 90˚ phase difference, 4x 01: 90˚ phase difference, 2x 11: Input count-up (count forward) pulses or count-down pulses (Two-pulse input)	[RENV2] (WRITE) 23 16 - - n n - - - -
Specify the EA/EB input count direction <Set RENV2.EDIR (bit 22)> 0: Count up (count forward) when the EA phase is leading. Or, count up (count forward) on the rising edge of EA. 1: Count up (count forward) when the EB phase is leading. Or, count up (count forward) on the rising edge of EB.	[RENV2] (WRITE) 23 16 - n - - - - - -
Enable/disable EA/EB input <Set RENV2.EOFF (bit 30)> 0: Enable EA/EB input 1: Disable EA/EB input. (EZ input is valid.)	[RENV2] (WRITE) 31 24 - n - - - - - -
Set the input signal filter for PA/PB <Set RENV2.PINF (bit 19)> 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 - - -
Specify the PA/PB input <Set RENV2.PIM0 to 1 (bit 24 to 25)> 00: 90˚ phase difference, 1x 10: 90˚ phase difference, 4x 01: 90˚ phase difference, 2x 11: Input count-forward pulses or count-down pulses (Two-pulse input)	[RENV2] (WRITE) 31 24 - - - - - - n n
Specify the PA/PB input count direction <Set RENV2.PDIR (bit 26)> 0: Count up (count forward) when the PA phase is leading. Or, count up (count forward) on the rising edge of PA. 1: Count up (count forward) when the PB phase is leading. Or, count up (count forward) on the rising edge of PB.	[RENV2] (WRITE) 31 24 - - - - - n - -
Enable/disable PA/PB input <Set RENV2.POFF (bit 31)> 0: Enable PA/PB input 1: Disable PA/PB input.	[RENV2] (WRITE) 31 24 n - - - - - - -
Reading EA/EB, PA/PB input error <REST.ESEE (bit 16), REST.ESPE (bit 17)> ESEE (bit 16) = 1: An EA/EB input error occurred. ESPE (bit 17) = 1: A PA/PB input error occurred.	[REST] (READ) 23 16 0 0 0 0 0 0 n n

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

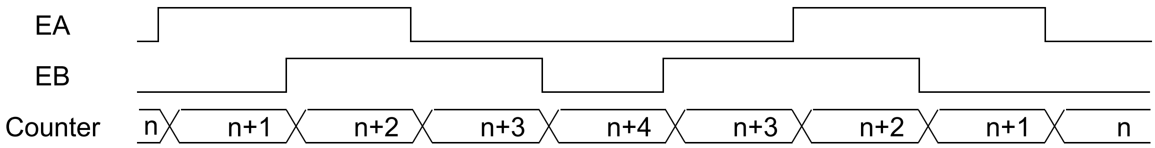
1) When using 90˚ phase difference signals and 1x input



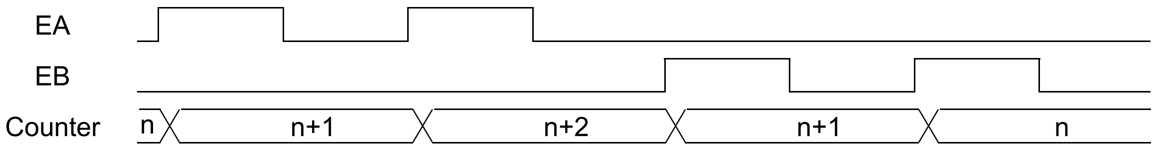
2) When using 90˚ phase difference signals and 2x input



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



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



11-10-2. Counter reset

All the counters can be reset using any of the following three methods.
1) When the CLR input signal turns ON (set in RENV3).
2) When an origin return is executed (set in RENV3).
3) When a command is written.
The PCL can also be specified to reset automatically, soon after latching the counter value.

The CLR input timing can be set in RENV1 (environment setting 1). An #INT signal can be output as an event interrupt cause when a CLR signal is input.

Action when the CLR signal turns ON <Set RENV3.CU1C to 4C (bit 16 to 19)> CU1C (bit 16) =1: Reset COUNTER1 (command position). CU2C (bit 17) =1: Reset COUNTER2 (mechanical position). CU3C (bit 18) =1: Reset COUNTER3 (deflection). CU4C (bit 19) =1: Reset COUNTER4 (general-purpose).	[RENV3] (WRITE) 23 16 - - - - n n n n
Action when an origin return is complete <Set RENV3.CU1R to 4R (bit 20 to 23)> CU1R (bit 20) =1: Reset COUNTER1 (command position). CU2R (bit 21) =1: Reset COUNTER2 (mechanical position) CU3R (bit 22) =1: Reset COUNTER3 (deflection) CU4R (bit 23) =1: Reset COUNTER4 (general-purpose)	[RENV3] (WRITE) 23 16 n n n n - - - -
Setting when latched <Set RENV5.CU1L to 4L (bits 24 to 27)> CU1L (bit 24) = 1: Reset COUNTER1 (command position). CU2L (bit 25) = 1: Reset COUNTER2 (machine position). CU3L (bit 26) = 1: Reset COUNTER3 (deflection). CU4L (bit 27) = 1: Reset COUNTER4 (general-purpose).	[RENV5] (WRITE) 31 24 0 0 0 0 n n n n
Action for the CLR signal <Set RENV1.CLR 0 to 1 (bit 20 to 21)> 00: Clear on the falling edge 10: Clear on a LOW level 01: Clear on the rising edge 11: Clears on a HIGH level	[RENV1] (WRITE) 23 16 - - n n - - - -
Reading the CLR signal <RSTS.SCLR (bit 13)> 0: The CLR signal is OFF 1: The CLR signal is ON	[RSTS] (READ) 15 8 - - n - - - - -
Set event interrupt cause <Set RIRQ.IRCL (bit 13)> 1: Output an #INT signal when resetting the counter value by turning the CLR signal ON.	[RIRQ] (WRITE) 15 8 - - n - - - - -
Read the event interrupt cause <RIST.ISCL (bit 13)> 1: When you want to reset the counter value by turning ON the CLR signal.	[RIST] (READ) 15 8 - - n - - - - -
Counter reset command <CUN1R to CUN4R: Control command> 20h: Set COUNTER1 (command position) to zero 21h: Set COUNTER2 (mechanical position) to zero. 22h: Set COUNTER3 (deflection) to zero. 23h: Set COUNTER4 (general-purpose) to zero	[Control command] 20h 21h 22h 23h

Note: In order to prevent incorrect counts, when the count timing and reset timing match, the counter will be +1 or -1, never 0. Please note this operation detail when detecting 0 with the comparator function.

11-10-3. Latch the counter and count condition

All the counters can latch their counts using any of the following methods. The setting is made in RENV5 (environment setting 5) register. The latched values can be output from the RLTC1 to 4 registers.

1) Turn ON the LTC signal.
2) Turn ON the ORG signal.
3) When the conditions for Comparator 4 are satisfied.
4) When the conditions for Comparator 5 are satisfied.
5) When a command is written.

The current speed can also be latched instead of COUNTER3 (deflection). Latch at the timing to use hardware (above items 1) to 4)) can also stopped.
The LTC input timing can be set by in RENV1 (environment setting 1). An #INT signal can be output when a counter value is latched by turning ON the LTC signal or the ORG signal. This allows you to identify the cause of an event interrupt.

Specify the latch method for a counter (1 to 4) <Set RENV5.LTM0 to 1 (bit 12 to 13)> 00: Turn ON the LTC signal. 01: Turn ON the ORG signal. 10: When the conditions for Comparator 4 are satisfied. 11: When the conditions for Comparator 5 are satisfied	[RENV5] (WRITE) 15 8 - - n n - - - -
Specify the latch method for the current speed <Set REMV5.LTFD (bit 14)> 1: Latch the current speed instead of COUNTER 3 (deflection).	[RENV5] (WRITE) 15 8 - n - - - - - -
Specify latching using hardware <Set RENV5.LTOF (bit 15)> 1: Stop latching at the timing to use hardware (above 1) to 4)).	[RENV5] (WRITE) 15 8 n - - - - - - -
Specify the LTC signal mode <Set RENV1.LTCL (bit 23)> 0: Latch on the falling edge. 1: Latch on the rising edge.	[RENV1] (WRITE) 23 16 n - - - - - - -

Set an event interrupt cause <Set RIRQ. IRLT (bit 14) and RIRQ.IROT (bit 15)>
IRLT = 1: Output an #INT signal when the counter value is latched by the LTC signal being turned ON.
IROL = 1: Output an #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 14), RIST.ISOL (bit 15)> 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 - - - - - -
Read the LTC signal <RSTS.SLTC (bit 14)> 0: The LTC signal is OFF 1: The LTC signal is ON	[RSTS] (READ) 15 8 - n - - - - - -
Counter latch command <LTCH: Control command> Latch the contents of the counters (COUNTER1 to 4).	[Control command] 29h

11-10-4. Stop the counter

COUNTER1 (command position) stops when the PRMD.MCCE is set to stop the counter and while in timer mode operation.
COUNTER2 (mechanical position), COUNTER3 (deflection), and COUNTER4 (general-purpose) stop when the RENV3.CU2H to 4H is set to stop.
By setting the RENV3 register, you can stop counting pulses while performing a backlash or slip correction.
COUNTER4 (general-purpose) can be set to count only during operation (BSY = low) using the RENV3 register.
By specifying 1/2 of the CLK (reference clock) signal, the time after the start can be controlled.

Stopping COUNTER1 (command) <Set RMD.MCCE (bit 11)> 1. Stop COUNTER1 (command position).	[RMD] (WRITE) 15 8 n - - - - - - -
Specify the counting operation for COUNTERS 2 to 4 <Set RENV3.CU2H to 4H (bits 29 to 31)> CU2H (bit 29) = 1: Stop COUNTER2 counting (mechanical position) CU3H (bit 30) = 1: Stop COUNTER3 counting (deflection) CU4H (bit 31) = 1: Stop COUNTER4 counting (general-purpose)	[RENV3] (WRITE) 31 24 n n n 0 - - - -
Setting the counters for backlash or slip correction <Set RENV3.CU1B to 4B (bits 24 to 27)> CU1B (bit 24) = 1: Enable COUNTER1 (command position) CU2B (bit 25) = 1: Enable COUNTER2 (mechanical position) CU3B (bit 26) = 1: Enable COUNTER3 (deflection) CU4B (bit 27) = 1: Enable COUNTER4 (general-purpose)	[RENV3] (WRITE) 31 24 - - - 0 n n n n
Specify the counting conditions for COUNTER4 <Set RENV3.BSYC (bit 14)> 1. Enable COUNTER4 (general-purpose) only while operating (#BSY = L).	[RENV3] (WRITE) 15 8 - n - - - - - -

11-11. Comparator

11-11-1. Comparator types and functions

This LSI has 5 circuits of 28-bit comparators per axis. It compares the values set in the RCMP1 to 5 registers with the counter values.
Comparators 1 to 4 can be used as comparison counters and can be assigned as COUNTERS 1 to 4. Comparator 5 can be assigned as COUNTER 1 to 4, a positioning counter, or to track the current speed. There are many comparison methods and four processing methods that can be used when the conditions are met. Specify the comparator conditions in the RENV4 (environment 4) and RENV5 (environment 5) registers. By using these comparators, you can perform the following.

- Use comparators for INT outputs, external output of comparison data, and for internal synchronous starts
- Immediate stop and deceleration stop operations.
- Rewrite operation data with pre-register data (used to change speed while operating).
- Software limit function using Comparators 1 and 2.
- Ring count function using COUNTER1 (command position) and Comparator 1.
- Ring count function using COUNTER2 (mechanical position) and Comparator 2.
- Detect out of step stepper motors using COUNTER3 (deflection) and a comparator.
- Output a synchronous signal (IDX) using COUNTER4 (general-purpose) and a Comparator 4.

Comparator 5 is equipped with a pre-register. It can also output an #INT signal as event interrupt cause when the comparator's conditions are satisfied.

[Comparison data]
Each comparator can select the data for comparison from the items in the following table.

Comparison data	Comparator 1		Comparator 2		Comparator 3		Comparator 4		Comparator 5	
		C1C0 to 1		C2C0 to 1		C3C0 to 1		C4C0 to 1		C5C0 to 2
COUNTER1 (command position)	O	"00"	O	"00"	O	"00"	O	"00"	O	"000"
COUNTER2 (mechanical position)	O	"01"	O	"01"	O	"01"	O	"01"	O	"001"
COUNTER3 (deflection)	O	"10"	O	"10"	O	"10"	O	"10"	O	"010"
COUNTER4 (general-purpose)	O	"11"	O	"11"	O	"11"	O	"11"	O	"011"
Positioning counter									O	"100"
Current speed									O	"101"
Pre-register	None		None		None		None		Yes	
Major application	+SL Use COUNTER1 as a ring counter		-SL Use COUNTER2 as a ring counter				IDX output			

- O: Comparison possible. Blank: Comparison not possible.
- +SL and -SL are used for software limits.
- If COUNTER3 (deflection) is selected as the comparison counter, the LSI will compare the absolute value of the counter with the comparator data. (Absolute value range: 0 to 32,767)
- The bit assignments of the comparison data settings are as follows:

C1C0 to 1 (RENV4 bits 0 to 1), C2C0 to 1 (RENV4 bits 8 to 9), C3C0 to 1 (RENV4 bits 16 to 17), C4C0 to 1 (RENV4 bits 24 to 25), C5C0 to 2 (RENV5 bits 0 to 2)

[Comparison method] Each comparator can be assigned a comparison method from the table below.

Comparison method	Comparator 1			Comparator 2			Compa -rator3		Compa -rator4		Compa -rator5	
		C1S0 to 2	C1RM		C2S0 to 2	C1RM		C3S0 to 2		C4S0 to 3		C5S0 to 2
Comparator = Comparison counter (regardless of count direction)	O	"001"	'0'	O	"001"	'0'	O	"001"	O	"0001"	O	"001"
Comparator = Comparison counter (count up (count forward) only)	O	"010"	'0'	O	"010"	'0'	O	"010"	O	"0010"	O	"010"
Comparator = Comparison counter (count down only)	O	"011"	'0'	O	"011"	'0'	O	"011"	O	"0011"	O	"011"
Comparator > Comparison counter	O	"100"	'0'	O	"100"	'0'	O	"100"	O	"0100"	O	"100"
Comparator < Comparison counter	O	"101"	'0'	O	"101"	'0'	O	"101"	O	"0101"	O	"101"
Use as software limits	O	"110"	'0'	O	"110"	'0'						
IDX (synchronous signal) output (regardless of counting direction)									O	"1000"		
IDX (synchronous signal) output (count up (count forward) only)									O	"1001"		
IDX (synchronous signal) output (count down only)									O	"1010"		
Use COUNTER1 as a ring counter	O	"001"	'1'						O	"1010"		
Use COUNTER2 as a ring counter				O	"001"	'1'			O	"1010"		

- O: Comparison possible. Blank: Comparison impossible.
- When used as software limits, value of Comparator 1 is a positive direction limit value and the comparison method is "comparator < comparison counter". Value of Comparator 2 a negative limit value and the comparison method is "comparator > comparison counter". Select COUNTER1 (command position) for the comparison counter.
- Comparator 3 must not have C3S0 to 2 set to a value of 110. Setting any of the values always result in failing to satisfy the comparison conditions.
- When C4S0 to 3 = 1000 to 1010 for Comparator 4 <IDX (synchronous signal) output>, select COUNTER4 (general-purpose) for use as the comparison counter. Other counters cannot be used for this function. Enter a positive value for the comparator setting.
- The bit assignments for various comparison methods are as follows:
  C1S0 to 2 (RENV4 bits 2 to 4), C2S0 to 2 (RENV4 bits 10 to 12), C3S0 to 2 (RENV4 bits 18 to 20), C4S0 to 3 (RENV4 bits 26 to 29), C5S0 to 2(RENV5 bits 3 to 5)

[Processing method when comparator conditions are satisfied] The processing method that is used when the conditions are satisfied can be selected from the table below.

Processing method when the conditions are satisfied	Comparator 1	Comparator 2	Comparator 3	Comparator 4	Comparator 5
	C1D0 to 1	C2D0 to 1	C3D0 to 1	C4D0 to 1	C5D0 to 1
Does nothing	"00"	"00"	"00"	"00"	"00"
Immediate stop operation	"01"	"01"	"01"	"01"	"01"
Deceleration stop operation	"10"	"10"	"10"	"10"	"10"
Rewrite operation data with pre-register data	"11"	"11"	"11"	"11"	"11"

- "Does nothing " is mainly used for INT output, external output of comparison result, or internal synchronous starts.
- To change the speed pattern while in operation, use "rewrite operation data with pre-register data". The PRMV setting will also be transferred to the RMV. However, this does not affect operation.
- The bit assignments to select a processing method are as follows.
  C1D0 to 1 (RENV4 bits 5 to 6), C2D0 to 1 (RENV4 bits 13 to 14), C3D0 to 1 (RENV4 bits 21 to 22), C4D0 to 1 (RENV4 bits 30 to 31), C5D0 to 1 (RENV5 bits 6 to 7)

[How to set the INT output, external output of comparison results, and internal synchronous starting]

Set an event interrupt cause <Set RIRQ.IRC1 to 5 (bit 8 to 12)> IRC1 (bit 8) = 1 : Output #INT signal when the Comparator 1 conditions are satisfied. IRC2 (bit 9) = 1 : Output #INT signal when the Comparator 2 conditions are satisfied. IRC3 (bit 10) = 1 : Output #INT signal when the Comparator 3 conditions are satisfied. IRC4 (bit 11)= 1: Output #INT signal when the Comparator 4 conditions are satisfied. IRC5 (bit 12)= 1 : Output #INT signal when the Comparator 5 conditions are satisfied.	[RIRQ] (WRITE) 15 8 - - - n n n n n
Read the event interrupt cause <RIST.ISC1 to 5 (bit 8 to 12)> IRC1 (bit 8) = 1: When the Comparator 1 conditions are satisfied. IRC2 (bit 9) = 1: When the Comparator 2 conditions are satisfied. IRC3 (bit 10) = 1: When the Comparator 3 conditions are satisfied. IRC4 (bit 11) = 1: When the Comparator 4 conditions are satisfied. IRC5 (bit 12) = 1: When the Comparator 5 conditions are satisfied.	[RIST] (READ) 15 8 - - - n n n n n
Read the comparator condition status <MSTSW.SCP1 to 5 (bits 8 to 12)> SCP1 (bit 8) = 1: When the Comparator 1 conditions are satisfied. SCP2 (bit 9) = 1: When the Comparator 2 conditions are satisfied. SCP3 (bit 10) = 1: When the Comparator 3 conditions are satisfied. SCP4 (bit 11) = 1: When the Comparator 4 conditions are satisfied. SCP5 (bit 12) = 1: When the Comparator 5 conditions are satisfied.	[MSTSW] (READ) 15 8 - - - n n n n n
Specify the P3/CP1 (+SL) terminal specifications<RENV1.P3M0 to 1 (bits 6 to 7)> 00: General-purpose input 01: General-purpose output 10: Output a CP1 (Comparator 1 conditions satisfied) signal using negative logic. 11: Output a CP1 (Comparator 1 conditions satisfied) signal using positive logic.	[RENV2] (WRITE) 7 0 n n - - - - - -
Specify the P4/CP2 (-SL) terminal specifications<RENV2.P4M0 to 1 (bits 8 to 9)> 00: General-purpose input 01: General-purpose output 10: Output CP2 (Comparator 2 conditions satisfied) signal using negative logic. 11: Output CP2 (Comparator 2 conditions satisfied) signal using positive logic.	[RENV2] (WRITE) 15 8 - - - - - - n n
Specify the P5/CP3 terminal specifications<Set REMV2.P5M0 to 1 (bits 10 to 11)> 00: General-purpose input 01: General-purpose output 10: Output CP3 (Comparator 3 conditions satisfied) signal using negative logic. 11: Output CP3 (Comparator 3 conditions satisfied) signal using positive logic.	[RENV2] (WRITE) 15 8 - - - - n n - -
Specify the P6/CP4 terminal specifications<Set RENV2.P6M0 to 1 (bits 12 to 13)> 00: General-purpose input 01: General-purpose output 10: Output CP4 (Comparator 4 conditions satisfied) signal using negative logic. 11: Output CP4 (Comparator 4 conditions satisfied) signal using positive logic.	[RENV2] (WRITE) 15 8 - - n n - - - -
Specify the P7/CP5 terminal specifications<Set RENV2.P7M0 to 1 (bits 14 to 15)> 00: General-purpose input 01: General-purpose output 10: Output CP5 (Comparator 5 conditions satisfied) signal using negative logic. 11: Output CP5 (Comparator 5 conditions satisfied) signal using positive logic.	[RENV2] (WRITE) 15 8 n n - - - - - -

Specify the output timing for an internal synchronous signal <Set RENV5.SYO1 to 3 (bits 16 to 19)>	[RENV5] (WRITE)
0001: When the Comparator 1 conditions are satisfied. 0010: When the Comparator 2 conditions are satisfied. 0011: When the Comparator 3 conditions are satisfied. 0100: When the Comparator 4 conditions are satisfied. 0101: When the Comparator 5 conditions are satisfied. 1000: When the acceleration starts. 1001: When the acceleration is complete. 1010: When the deceleration starts 1011: When the deceleration is complete. Others: Turn OFF internal synchronous output signal	23 16 - - - - n n n n

[Speed change using the comparator]

When the comparator conditions are satisfied, you can use the function "Rewrite operation data with pre-register data. This function is used to change the speed at a specified position.
Also, Comparator 5 has a pre-register function, and can be specified for use in changing the speed several time.
In this case, use the "command to determine pre-register (4Fh)," to specify several sets of speed data.

If the speed change data (data used with commands to determine) are left in Pre-registers 1 and 2 when the current operation completes (Example 1), or if the speed change data is left in Pre-register 1 and some next operation data exists in Pre-register 2 (Example 2), the PCL will ignore the speed change data and shift the data from the pre-registers.
Then, in Example 2, the PCL will start the next operation after shifting the data from the pre-registers.

Example 1

	(PFM=11)			(PFM=00)
Pre-register 2	Speed change data 2 (determined)		Pre-register 2	Speed change data 2 (undetermined)
Pre-register 1	Speed change data 1 (determined)	Complete current operation →	Pre-register 1	Speed change data 2 (undetermined)
Register	Current operation data (determined)		Register	Speed change data 1 (undetermined)

Example 2

	(PFM=11)			(PFM=01)
Pre-register 2	Next operation data (determined)		Pre-register 2	Next operation data (undetermined)
Pre-register 1	Speed change data (determined)	Complete current operation →	Pre-register 1	Next operation data (undetermined)
Register	Current operation data (determined)		Register	Next operation data (determined)

Determine a pre-register <PRESET: Control command>	[Control command]
Determine the pre-register details as speed change data.	4Fh

11-11-2. Software limit function

A software limit function can be set up using Comparators 1 and 2.
Select COUNTER1 (command position) as a comparison counter for Comparators 1 and 2.
Use Comparator 1 for a positive direction limit and Comparator 2 for a negative direction limit to stop an axis based on the results of the comparator and the operation direction.
When the software limit function is used the following process can be executed.

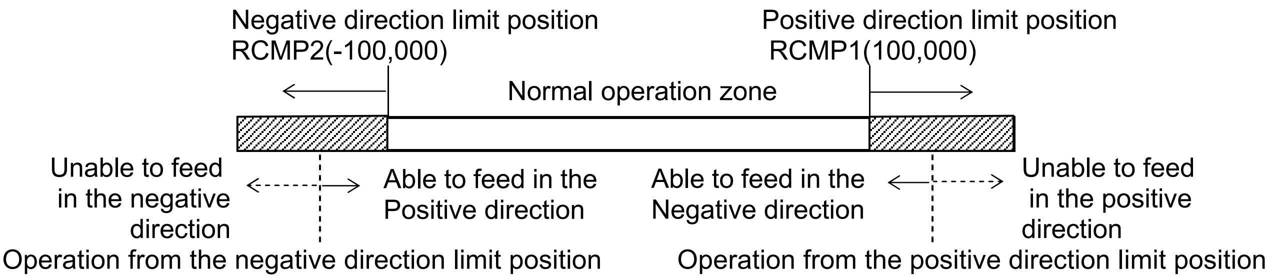
1) Stop pulse output immediately
2) Decelerate and then stop pulse output

While using the software limit function, if a deceleration stop is selected as the process to use when the comparator conditions are satisfied (RENV4.C1D, RENV4.C2D), when a machine reaches the software limit while in a high speed start (52h, 53h), that axis will stop using deceleration. When some other process is specified for use when the conditions are satisfied, or while in a constant speed start, that axis will stop immediately.

If a software limit is ON while writing a start command, the axis will not start to move in the direction in which the software limit is enabled. However, it can start in the opposite direction.

[Setting example]

RENV4=00003838h: Use Comparator 1 as positive direction software limit.
Use Comparator 2 as negative direction software limit.
Set to stop immediately when the software limit is reached.
RCMP1= 100,000: Positive direction limit value
RCMP2= -100,000: Negative direction limit value



Specify the comparison method for Comparator 1 <Set RENV4.C1S0 to C1S2 (bits 2 to 4)> 110: Use as a positive direction software limit	[RENV4] (WRITE) 7 0 - - - n n n - -
Specify the process to use when the Comparator 1 conditions are satisfied <Set RENV4.C1D0 to C1D1 (bits 5 to 6)> 01: Immediate stop 10: Deceleration stop	[RENV4] (WRITE) 7 0 - n n - - - - -
Specify the comparison method for Comparator 2 <Set RENV4.C2S0 to C2S2 (bits 10 to 12)> 110: Use as a negative direction software limit.	[RENV4] (WRITE) 15 8 - - - n n n - -
Specify the process to use when the Comparator 2 conditions are satisfied <Set RENV4.C2D0 to C2D1 (bits 13 to 14)> 01: Immediate stop 10: Deceleration stop	[RENV4] (WRITE) 15 8 - n n - - - - -

11-11-3. Out of step stepper motor detection function

If the deflection counter value controlled by the motor command pulses and the feedback pulses from an encoder on a stepper motor exceed the maximum deflection value, the LSI will declare that the stepper motor is out of step. The LSI monitors stepper motor operation using COUNTER3 (the deflection counter) and a comparator.
The process which takes place after an out of step condition is detected can be selected from the table [Processing method to use when the comparator conditions are satisfied].
For this function, use an encoder with the same resolution as the stepper motor.
COUNTER3 (deflection) can be cleared by writing a set command to the deflection counter.
There are two methods for inputting a feedback signal: Input 90˚ phase difference signals (1x, 2x, 4x) on the EA/EB terminals and input count-up (count-forward) and count-down pulses (Two-pulse mode).
If both EA and EB signals change at the same time, the LSI will treat this as an error and output an #INT signal.
[Setting example]

RENV4 = 00360000h: Satisfy the conditions of Comparator 3 < COUNTER3 (deflection)
Stop immediately when the conditions are satisfied.
RCMP3 = 32: The maximum deflection value is "32" pulses.
RIRQ = 00000400h: Output an #INT signal when the conditions for Comparator 3 are satisfied.

Specify the EA/EB input <Set RENV2.EIM0 to 1 (bits 20 to 21)> 00: 90˚ phase difference, 1x 01: 90˚ phase difference, 2x 10: 90˚ phase difference, 4x 11: Two-pulse mode (count-up (count-forward) pulses and count-down pulses)	[RENV2] (WRITE) 23 16 - - n n 0 0 - -
Specify the EA/EB input count direction <Set RENV2.EDIR (bit 22)> 0: Count up (count forward) when the EA phase is leading or on the EA rising edge. 1: Count up (count forward) when the EB phase is leading or on the EB rising edge	[RENV2] (WRITE) 23 16 - n - - 0 0 - -
Read the EA/EB input error <REST.ESEE (bit 16)> 1: An EA/EB input error has occurred.	[REST] (READ) 23 16 0 0 0 0 0 0 - n
Counter reset command <CUN3R: Control command> Clear COUNTER3 (deflection) to zero.	[Control command] 22h

11-11-4. IDX (synchronous) signal output function

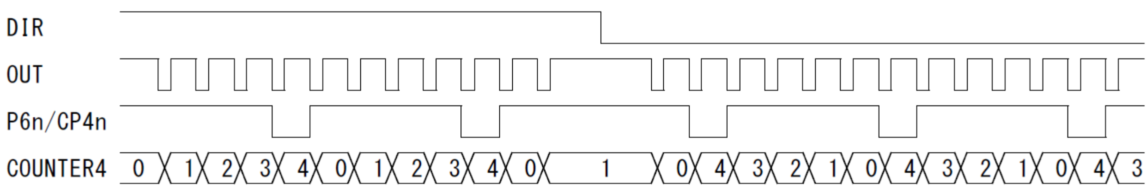
Using Comparator 4 and COUNTER4, the PCL can output signals to the P6n/CP4n terminals at specified intervals. Setting RENV4.C4C0 and C4C1 to "11" (in the general-purpose counter) and setting RENV4.C4S0 thru C4S3 to "1000", "1001 or "1010" (the IDX output), the PCL can be used for IDX (index) operation.
The counter range of COUNTER4 will be 0 to (the value set in RCMP4). If counting down from 0, the next counter value will be the value set in RCMP4, and if counting up (counting forward) from the value set in RCMP3, the next counter value will be 0. (RCMP4 setting range: 1 to 2,147,483,647). The input for COUNTER4 can be set with RENV3.CI40 or CI41.
By setting RENV4.IDXM, you can select either level output or count output.

Select the specification for the P6/CP4 terminals <Set RENV2.P6M0 to 1(bits 12 to 13)> 10: Output an IDX signal using negative logic 11: Output an IDX signal using positive logic	[RENV2] (WRITE) 15 8 - - n n - - - -
Select the count input for COUNTER4 (general-purpose) <Set to RENV3.CI40 to CI41 (bits 12 to 13)> 00: Output pulses 10: PA/PB input 01: EA/EB input 11: 1/2 division of clock of the CLK.	[RENV3] (WRITE) 15 8 - - n n - - - -
Select the comparison counter for Comparator 4 <Set RENV4.C4C0 to 1 (bits 24 to 25)> 11: COUNTER4 (general-purpose).	[RENV4] (WRITE) 31 24 - - - - - - n n
Select the comparison method for COUNTER4 <Set RENV4. C4S0 to 3 (bits 26 to 29)> 1000: IDX output (regardless of count direction) 1001: IDX output (only while counting up (counting forward)) 1010: IDX output (only while counting down)	[RENV4] (WRITE) 31 24 - - n n n n - -
Select the IDX output mode <Set RENV4.IDXM (bit 23)> 0: Outputs an IDX signal while COUNTER4 = RCMP4. 1: Outputs an IDX signal for two CLK cycles when COUNTER4 reaches 0 by counting.	[RENV4] (WRITE) 23 16 n - - - - - - -

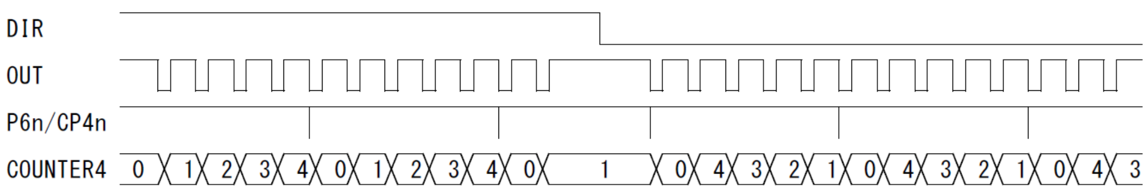
Note: While RENV4.IDXM = 1, writing a "0" to COUNTER4 or resetting COUNTER4 will not output an IDX signal. The setting in IDXM is effective only when RENV4.C4S0 to C4S3 are set to 1000, 1001, or 1010 (synchronous signal output).

Output example 1: (IDXM = 0: Level output)
Note : When IDXM (synchronous signal output) is set to 0 and C4S0 to C4S3 are set to 1001 or 1010, use a count range of RCMP4 ≥ 2.

Regardless of the feed direction, the PCL will output the IDX signal using negative logic for the output pulses. (Counting range: 0 to 4.)
Settings: RENV2 = 00002000h, RENV3 = 00000000h, RENV4 = 23000000h, RCMP4 = 4



Output example 2 (IDXM = 1: Count output)
Regardless of the feed direction, the PCL will output the IDX signal using negative logic for the output pulses. Counting range 0 to 4.
Settings: RENV2 = 00002000h, RENV3 = 00000000h, RENV4 = 23800000h, RCMP4 = 4



11-11-5. Ring count function

COUNTER1 and 2 have a ring count function for use in controlling a rotating table.

Set RENV4.C1PM = 1, RENV4.C1S0 to 2 = 000, and RENV4.C1C0 to 1 = 00 and COUNTER1 will be in the ring count mode. Then the PCL can perform the following operations.

- Count value = If counting up (counting forward) from the value set in RCMP1, the next counter value will be 0.
- Count value = If counting down from 0, the next counter value will be the value set in RCMP1.

Set RENV4.C2PM = 1, RENV4.C2S0 to 2 = 000, and RENV4.C2C0 to 1 = 01 and COUNTER2 will be in the ring count mode. Then the PCL can perform the following operations.

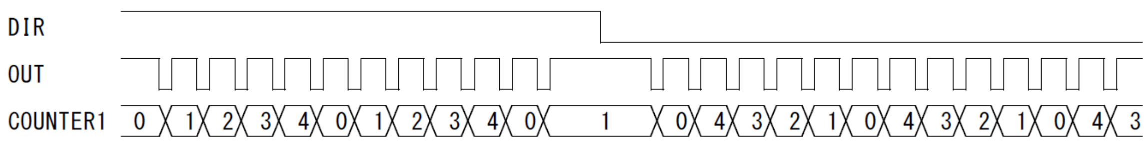
- Count value = If counting up (counting forward) from the value set in RCMP2, the next counter value will be 0.
- Count value = If counting down from 0, the next counter value will be the value set in RCMP2.

Set COUNTER1 to ring counter operation <Set RENV4.C1RM, C1D0 to 1, C1S0 to 2, and C1C0 to 1> 10000000: Operate COUNTER1 as a ring counter.	[RENV2] (WRITE) 7 0 n n n n n n n n
Set COUNTER2 to ring count operation <Set RENV4.C2RM, C2D0 to 1, C2C0 to 1> 10000001: Operate COUNTER2 as a ring counter.	[RENV2] (WRITE) 15 8 n n n n n n n n

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

Note: To use the ring counter function, set the count value between 0 and the value in RCMPn. If the value is outside the range above, the PCL will not operate normally. Set the comparator conditions (C1S0 to 2, C2S0 to 2) when using a counter as a ring counter to "000."

Setting example
RENV4 = XXXXXX80h --- COUNTER1 is in ring counter mode (C1RM = 1, C1S0 to 2 = 000, C1C0 to 1 = 00)
RCMP1 = 4 --- Count range: 0 to 4



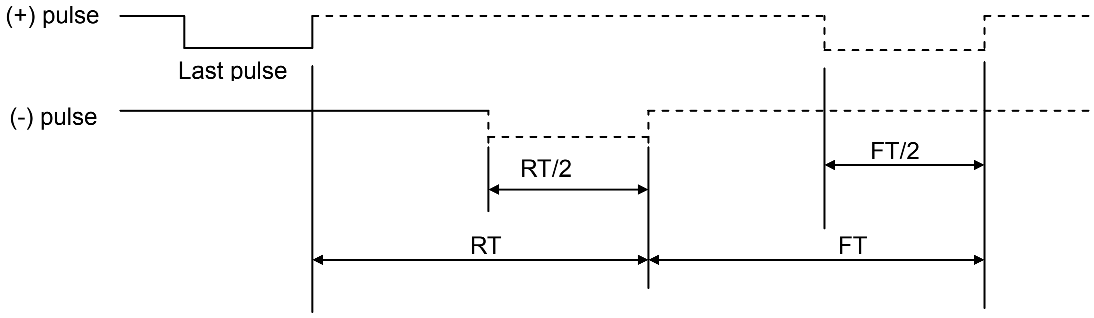
11-12. Backlash correction and slip correction

This LSI has backlash and slip correction functions. These functions output the number of command pulses specified for the correction value in the speed setting in the RFA (correction speed) register before command operation.
The backlash correction is performed each time the direction of operation changes. The slip correction function is performed before a command, regardless of the feed direction. The correction amount and method is specified in the RENV6 (environment setting 6) register.
The operation of the counter (COUNTER 1 to 4) can be set using the RENV3.CU1B to CU4B.

Enter the correction value <RENV6.BR0 to 11 (bits 0 to 11)> Backlash or slip correction amount value (0 to 4095)	[RENV6] (WRITE) 15 8 - - - - n n n n 7 0 n n n n n n n n
Set the correction method <RENV6.ADJ0 to 1 (bits 12 to13)> 00: Turn the correction function OFF 01: Backlash correction 10: Slip correction	[RENV6] (WRITE) 15 8 - - n n - - - -
Action for backlash/slip correction <RENV3.CU1B to 4B (bit 24 to 27)> CU1B (bit 24) = 1: Enable COUNTER1 (command position) CU2B (bit 25) = 1: Enable COUNTER2 (mechanical position) CU3B (bit 26) = 1: Enable COUNTER3 (deflection) CU4B (bit 27) = 1: Enable COUNTER4 (general-purpose)	[RENV3] (WRITE) 31 24 - - - 0 n n n n

11-13. Vibration restriction function

This LSI has a function to restrict vibration when stopping by adding one pulse of reverse operation and one pulse of forward operation shortly after completing a command pulse operation.
Specify the output timing for additional pulses in the RENV7 (environment setting 7) register.
When both the reverse timing (RT) and the forward timing (FT) are non zero, the vibration restriction function is enabled.
The dotted lines below are pulses added by the vibration restriction function. (An example in the positive direction)



Specify the reverse operation timing <Set RENV7.RT0 to 15 (bits 0 to 15)> RT range: 0 to 65,535 The units are 32x of the reference clock cycle (approx. 1.6 μsec when CLK = 19,6608 MHz) Settable range: 0 to approx. 0.1 sec.	[RENV7] (WRITE) 15 8 n n n n n n n n 7 0 n n n n n n n n
Specify the forward operation timing <Set RENV7.FT0 to 15 (bits 16 to 31)> FT range: 0 to 65,535 The units are 32x of the reference clock cycle (approx. 1.6 μsec when CLK = 19,6608 MHz) Settable range: 0 to approx. 0.1 sec.	[RENV7] (WRITE) 31 24 n n n n n n n n 23 16 n n n n n n n n

Note: The optimum values for RT and FT will vary with each piece of machinery and load. Therefore, it is best to obtain these values by experiment.

11-14. Synchronous starting

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

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

The internal synchronous signal output is available with 9 types of timing. They can be selected by setting the RENV5 (environment setting 5) register. By setting the RIRQ (event interrupt cause) register, an #INT signal can be output at the same time the internal synchronous signal is output. You can determine the cause of event interrupt by reading the RIST register. The operation status can be checked by reading the RSTS (extension status) register.

Specify the synchronous starting method <Set PRMD.MSY0 to 1 (bits 18 to 19)> 10: Start with an internal synchronous signal. 11: Start triggered by specified axis stopping.	[PRMD] (WRITE) 23 16 - - - - n n - -
Select an axis for confirming a stop (setting example) <Set PRMD.MAX0 to 3 (bits 20 to 23)> 0001: Start when the X axis stops 0010: Start when the Y axis stops 0100: Start when the Z axis stops 1000: Start when the U axis stops 0011: Start when both the X and Y axes stop. 0101: Start when both the X and Z axes stop 1011: Start when the X, Y, and U axes all stop. 1111: Start when all of the axes stop.	[PRMD] (WRITE) 23 16 n n n n - - - -
Select the synchronous starting mode <Set RENV2.SMAX (bit 29)> 0: Automatic assignment for "Start operation by stopping a specified axis" is invalid. 1: Automatic assignment for 'Start operation by stopping a specified axis" is valid.	[RENV2] (WRITE) 31 24 - - n - - - - -
Specify the internal synchronous signal output timing <Set RENV5.SYO1 to 3 (bits 16 to 19)> 0001: When the Comparator 1 conditions are satisfied. 0010: When the Comparator 2 conditions are satisfied. 0011: When the Comparator 3 conditions are satisfied. 0100: When the Comparator 4 conditions are satisfied. 0101: When the Comparator 5 conditions are satisfied. 1000: When the acceleration is started. 1001: When the acceleration is complete. 1010: When the deceleration is started. 1011: When the deceleration is complete Others: Internal synchronous output signal is OFF.	[RENV5] (WRITE) 23 16 - - - - n n n n
Specify the input for the internal synchronous signal <Set RENV5.SYI0 to 1 (bits 20 to 21)> 00: Use an internal synchronous signal output by the X axis. 01: Use an internal synchronous signal output by the Y axis. 10: Use an internal synchronous signal output by the Z axis. 11: Use an internal synchronous signal output by the U axis.	[RENV5] (WRITE) 23 16 - - n n - - - -
Read the operation status <RSTS.CND0 to 3 (bits 0 to 3)> 0011: Wait for an internal synchronous signal. 0100: Wait for another axis to stop.	[RSTS] (READ) 7 0 - - - - n n n n
Select the event interrupt (#INT output) cause <Set bit 4 to 12 of RIRQ> IRUS (bit 4) = 1: When the acceleration is started. IRUE (bit 5) = 1: When the acceleration is complete. IRDS (bit 6) = 1: When the acceleration is started. IRDE (bit 7) = 1: When the deceleration is complete. IRC1 (bit 8) = 1: When the Comparator 1 conditions are satisfied. IRC2 (bit 9) = 1: When the Comparator 2 conditions are satisfied. IRC3 (bit 10) = 1: When the Comparator 3 conditions are satisfied. IRC4 (bit 11) = 1: When the Comparator 4 conditions are satisfied. IRC5 (bit 12) = 1: When the Comparator 5 conditions are satisfied.	[RIRQ] (WRITE) 7 0 n n n n - - - - 15 8 - - - n n n n n

Read the event interrupt (#INT output) cause <Bit 4 to 12 of RIST>	[RIST] (READ)
ISUS (bit 4) = 1: When the acceleration is started. ISUE (bit 5) = 1: When the acceleration is complete. ISDS (bit 6) = 1: When the deceleration is started. ISDE (bit 7) = 1: When the deceleration is complete. ISC1 (bit 8) = 1: When the Comparator 1 conditions are satisfied. ISC2 (bit 9) = 1: When the Comparator 2 conditions are satisfied. ISC3 (bit 10) = 1: When the Comparator 3 conditions are satisfied. ISC4 (bit 11) = 1: When the Comparator 4 conditions are satisfied. ISC5 (bit 12) = 1: When the Comparator 5 conditions are satisfied.	7 … 0 n n n n - - - - 15 … 8 - - - n n n n n

11-14-1. Start triggered by another axis stopping

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

[Example 1]
After setting steps 1) to 3), start the X axis and Y axis. When both of these axes stop, the U axis starts.
1) Set PRMD.MSY0 to 1 (bits 18 to 19) for the U axis to "11." (Start triggered by another axis stopping.)
2) Set PRMD.MAX0 to 3 (bits 20 to 23) for the U axis to "0011." (When both X axis and Y axis stop.)
3) Write a start command for the U axis.

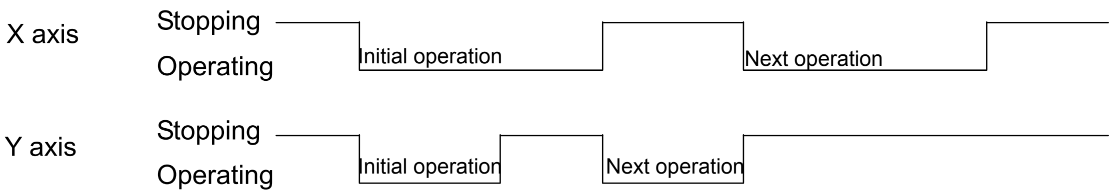
The "start when another axis stops" function has two operation modes: one is PCL6045 compatible and the other is the PCL6045B mode. Select the operation mode using RENV2.SMAX. (When SMAX = 0, the PCL6045 compatible mode is selected.)

[PCL6045 compatible mode]
In order to use "Other axis stops" as a start condition, the status of another axis has to change from operating to stopping after the axis specifying this condition is ready to start its process and then it can wait for the other axis' stop.
For example, if the X and Y axes are performing circular interpolation and "All axes stop" is set as a start condition for the next operation in the pre-register of the X and Y axes and other axes (Z and U axes) are already stopped after circular interpolation, the X and Y axes will never start the linear interpolation because the X and Y axes already stops before the X and Y axes start the process and wait for other axis's stopping, that means the change from operating to stopping does not occurs. The MAX setting cannot include the own axis itself.

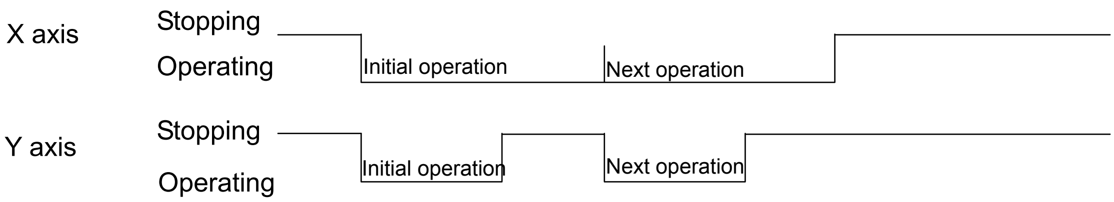
[PCL6045B mode]
When "start when another axis stops" is specified as the start condition for the next operation in a specific pre-register, the working axis itself can be included in the MAX setting.

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

1) When the PCL6045 compatible mode (SMAX = 0) is selected



2) When the PCL6045B mode (RENV1.SMAX = 1) is selected



When using continuous interpolation without changing the interpolation axes, you may set the next operation in the pre-register (you don't need to specify any stop conditions) rather using the "start when another axis stops" function. The settings are shown in Example 2 below.

The example below describes only the items related to the operations. The settings for speed and acceleration are omitted.

[Example 2]
How to set up a continuous interpolation (X-Y axis circular interpolation followed by an X-Y axis linear interpolation) without changing the interpolation axes.

Step	Register	X axis	Y axis	Description
1	PRMV	10000	10000	X and Y axes perform an circular interpolation operation of a 90^o curve with a radius of 10000
	PRIP	10000	0	
	PRMD	0000_0064h	0000_0064h	
	Start command: Write 0351h (FH constant speed start)			X and Y axes start command
2	PRMV	10000	5000	X and Y axes perform a linear interpolation 1 with an end point (1000, 5000)
	PRMD	0000_0061h	0000_0061h	
	Start command: Write 0351h (FH constant speed start)			X and Y axes start command

After the settings above are complete, the LSI will execute a continuous operation in the order shown below.

1. The X and Y axes perform a CW circular interpolation operation of a 90^o curve with a radius of 10000.
2. The X and Y axes perform a linear interpolation (10000, 5000)

Precautions are needed for continuous interpolation operations that change a plane containing interpolated axes using the pre-register function.
Basically, to change a plane containing interpolated axes, enter dummy operation data for all the axes (positioning operations with the feed amount set to 0), and then write the interpolation data for a new plane.

Note:
When changing the interpolated axis, failure to enter dummy operation data for all the axes may cause a continuous operation to stop or the interpolation operation may not stop when desired.

[Example 3 (PCL6045 compatible mode)]
How to perform continuous interpolation while changing the interpolated axes (moving from circular interpolation on the X and Y axes) to (Linear interpolation on the X and Y axes) to (Linear interpolation on the X and Z axes)

STEP	Register	X axis	Y axis	Z axis	Details
1	PRMV	10000	10000	0	The X and Y axes make a 90° circular interpolation with a radius of 10000. The Z axis is given a positioning operation with feed amount of 0. The X and Y axes start immediately. The Z axis has nothing to do and waits for the X and Y axes to stop.
	PRIP	10000	0	0	
	PRMD	0000 _0064h	0000 _0064h	003C _0041h	
	Start command: Write 0751h (FH constant speed start)				The X, Y, and Z axes Start command
2	PRMV	10000	5000	0	The X and Y axes perform linear interpolation 1, and the Z axis is given a positioning operation with a feed amount of 0. The X and Y axes wait for the Z axis to stop, and the Z axis waits for the X and Y axes to stop.
	PRMD	004C _0061h	004C _0061h	003C _0041h	
	Start command: Write 0751h (FH constant start)				The X, Y, and Z axes Start command
3	PRMV	10000	(Previous value)	-5000	X and Z axes perform linear interpolation 1. The X and Y axes wait for the Z axis to stop and the Z axis starts again, just like in continuous operation.
	PRMD	004C _0061h	(Previous value)	0000 _0061h	
	Start command: Write 0551h (FH constant start)				The X and Z axes Start command (X, Z axes SPRF = 1).

Using the settings above, the PCL will perform steps 1 to 5 continuously.

1. Start a CW circular interpolation of 90° with a radius 10000 on the X and Y axes.
2. After the X and Y axes stop, the Z axis positioning operation is complete (because the feed amount is 0).
3. Linear interpolation is performed on the X and Y axes (10000, 5000)
4. After the X and Y axes stop, the Z axis positioning operation is complete (because the feed amount is 0).
5. Linear interpolation is performed on the X and Z axes (10000, -5000).

Note: In STEP3 above, the value for the Y axis is left the same as in the previous step (STEP2), in order not to start the Y axis.

[Example 4 (PCL6045B mode)]
How to perform continuous interpolation while changing the interpolated axes (moving from circular interpolation on the X and Y axes) to (Linear interpolation on the X and Y axes) to (Linear interpolation on the X and Z axes)

STEP	Register	X axis	Y axis	Z axis	Details
1	PRMV	10000	10000	0	The X and Y axes perform a 90° circular interpolation with a radius of 10000. The Z axis is given a positioning operation with a feed amount of 0. The X, Y, and Z axes start immediately.
	PRIP	10000	0	0	
	PRMD	0000 _0064h	0000 _0064h	0000 _0041h	
	Start command: Write 0751h (FH constant speed start)				The X, Y, and Z axes Start command
2	PRMV	10000	5000	0	The X and Y axes perform linear interpolation. The Z axis is given a positioning operation with a feed amount of 0. The X, Y, and Z axes wait for the X, Y, and Z axes to stop.
	PRMD	007C _0061h	007C _0061h	007C _0041h	
	Start command: Write 0751h (FH constant start)				The X, Y, and Z axes Start command
3	PRMV	0	0	0	Since a plane containing interpolated axes is changed, all of the axes are given a dummy operation. The X, Y, and Z axes wait for the X, Y, and Z axes to stop
	PRMD	007C _0041h	007C _0041h	007C _0041h	
	Start command: Write 0751h (FH constant start)				The X, Y, and Z axes Start command
4	PRMV	10000	0	-5000	The X and Z axes perform linear interpolation. The Y axis is given a positioning operation with a feed amount of 0. The X, Y, and Z axes wait for the X, Y, and Z axes to stop
	PRMD	007C _0061h	007C _0041h	007C _0061h	
	Start command: Write 0751h (FH constant start)				X, Y, and Z axis start command.

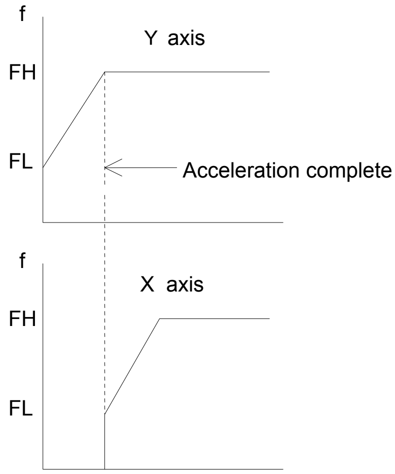
Using the settings above, the PCL will perform steps 1 to 3 continuously. (Specify STEP4 after STEP1 is complete)

1. Start a CW circular interpolation of 90° with a radius of 10000 on the X and Y axes. The Z axis performs a positioning operation with a feed amount of 0.
2. The X and Y axes perform a linear interpolation operation (10000, 5000). The Z axis performs a positioning operation with a feed amount of 0.
3. The X and Z axes perform a linear interpolation operation (10000, -5000). The Y axis performs a positioning operation with a feed amount of 0.

11-14-2. Starting from an internal synchronous signal

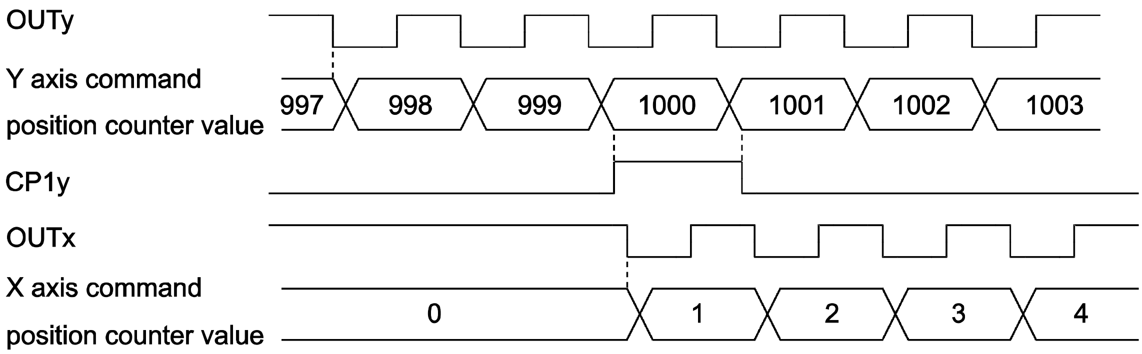
There are 9 types of internal synchronous signal output timing. They can be selected by setting the RENV5 register.
The monitor signal for the internal synchronous signal can be output externally.
Example 1 below shows how to use the end of acceleration for the internal synchronous signal.

[Example 1]
After completing steps 1) to 3) below, write a start command to the X and Y axes, the X axis will start when the Y axis completes its acceleration.
1) Set PRMD.MSY0 to 1 (bits 18 to19) of the X axis to 10. (Start with an internal synchronous signal)
2) Set RENV5.SYI0 to 1 (bits 20 to 21) of the X axis to 01. (Use an internal synchronous signal from the Y axis.)
3) Set RENV5.SYO0 to 3 (bits 16 to 19) of the Y axis to 1001. (Output an internal synchronous signal when the acceleration is complete)



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

[Example 2]
Use COUNTER1 (command position) and Comparator 1 to start the X axis when the Y axis = 1000.
1) Set PRMD.MSY0 to 1 (bits 18 to 19) of the Y axis to 10. (Start from an internal synchronous signal)
2) Set RENV5.SYI0 to 1 (bits 20 to 21) of the X axis to 01. (Use an internal synchronous signal from the Y axis)
3) Set RENV5.SYO0 to 3 (bits 16 to 19) of the Y axis to 0001. (Output an internal synchronous signal when the Comparator 1 conditions are satisfied)
4) Set RENV4.C1C0 to 1 (bits 0 to 1) of the Y axis to 00. (Comparator 1 comparison counter is COUNTER1)
5) Set RENV4.C1S0 to 2 (bits 2 to 4) of the Y axis to 001. (Comparison method: Comparator 1 = Comparison counter)
6) Set RENV4.C1D0 to 1 (bits 5 to 6) of the Y axis to 00. (Do nothing when the Comparator 1 condition are satisfied)
7) Set the RCMP1 value of the Y axis to 1000. (Comparison counter value of Comparator 1 is 1000.)
8) Write start commands for the X and Y axes.
The timing chart below shows the period after the Comparator 1 conditions are satisfied and the X axis starts.



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

Specify the use of the P0/FUP terminal <Set RENV2.P0M0 to 1 (bits 0 to 1)> 10: Output an FUP (accelerating) signal	[RENV2] (WRITE) 7 0 - - - - - - n n
Specify the use of the P1/FDW terminal <Set RENV2.P1M0 to 1 (bits 2 to 3)> 10: Output an FDW (decelerating) signal	[RENV2] (WRITE) 7 0 - - - - n n - -
Select the output logic for P0 (one shot) / FUP <Set RENV2.P0L (bit 16)> 0: Negative logic 1: Positive logic	[RENV2] (WRITE) 23 16 - - - - 0 0 - n
Select the output logic for P1 (one shot) / FDW <Set RENV2.P1L (bit 17)> 0: Negative logic 1: Positive logic	[RENV2] (WRITE) 23 16 - - - - 0 0 n -
Specify the use of the P3/CP1 (+SL) terminal <Set RENV2.P3M0 to 1 (bits 6 to 7)> 10: Output CP1 (Comparator 1 conditions are satisfied) using negative logic. 11: Output CP1 (Comparator 1 conditions are satisfied) using positive logic.	[RENV2] (WRITE) 7 0 n n - - - - - -
Specify the use of the P4/CP2 (-SL) terminal <Set RENV2.P4M0 to 1 (bits 8 to 9)> 10: Output CP2 (Comparator 2 conditions are satisfied) using negative logic. 11: Output CP2 (Comparator 2 conditions are satisfied) using positive logic.	[RENV2] (WRITE) 15 8 - - - - - - n n
Specify the use of the P5/CP3 terminal <Set RENV2.P5M0 to 1 (bits 10 to 11)> 10: Output CP3 (Comparator 3 conditions are satisfied) using negative logic. 11: Output CP3 (Comparator 3 conditions are satisfied) using positive logic.	[RENV2] (WRITE) 15 8 - - - - n n - -
Specify the use of the P6/CP4 terminal <Set RENV2.P6M0 to 1 (bits 12 to 13)> 10: Output CP4 (Comparator 4 conditions are satisfied) using negative logic. 11: Output CP4 (Comparator 4 conditions are satisfied) using positive logic.	[RENV2] (WRITE) 15 8 - - n n - - - -
Specify the use of the P7/CP5 terminal <Set RENV2.P7M0 to 1 (bits 14 to 15)> 10: Output CP5 (Comparator 5 conditions are satisfied) using negative logic. 11: Output CP5 (Comparator 5 conditions are satisfied) using positive logic.	[RENV2] (WRITE) 15 8 n n - - - - - -

11-15. Output an interrupt signal

This LSI can output an interrupt signal (#INT signal): There are 17 types of errors, 19 types of events, and change from operating to stopping that can cause an #INT signal to be output. All of the error interrupt causes will always output an #INT signal. Each of the event causes can be set in the RIRQ register to output an #INT signal or not.

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

To approximate a free curve interpolation using multiple linear interpolation operations, event interrupts will be generated at the end of each linear interpolation. When using the stop interrupt, set PRMD.MENI = 1. You can set it not to output a #INT signal if there is data for the next operation in pre-register.

The #INT signal is output continuously until all the causes on all the axes that produced interrupts have been cleared. In default, error interrupt causes are cleared when writing REST (error cause) register read out command and event interrupt causes are cleared when writing RIST register read out command, and stop interrupt causes are cleared when main status is read out.

However, when RENV5.MSMR (bit 22) or RENV5.ISMR (bit23) = 1, # INT output may not turns OFF because each register or main status are not cleared by. Please refer to "6-5-4. Reading the mains status", "8-3-35. REST register" and "8-3-36. RIST register".

To determine which type of interrupt occurred, on which axis and the cause of the interrupt, follow the procedures below.

1) Read the main status of the X axis and check whether any one of bits 2, 4, or 5 is "1."
2) If bit 2 (SENI) is "1," a Stop interrupt occurs.
3) If bit 4 (SERR) is "1," read the REST register to identify the interrupt cause.
4) If bit 5 (SINT) is "1," read the RIST register to identify the interrupt cause.
5) Repeat steps 1) to 4) above for the Y, Z, and U axes.

The steps above will allow you to determine the interrupt cause and turn the #INT output OFF.

Note 1: When reading a register from the interrupt routine, the details of the input/output buffer will change. If the #INT signal is output while the main routine is reading or writing registers, and the interrupt routine starts, the main routine may produce an error. Therefore, the interrupt routine should execute a PUSH/POP on input/output buffer. In the case of using full-address method, an error does not occur if you separate direct access (for main routine) and indirect access (for interrupt routine).

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

Note 3: When not using the #INT terminal, leave it open.
When using more than one PCL, the #INT terminals cannot be wired ORed.

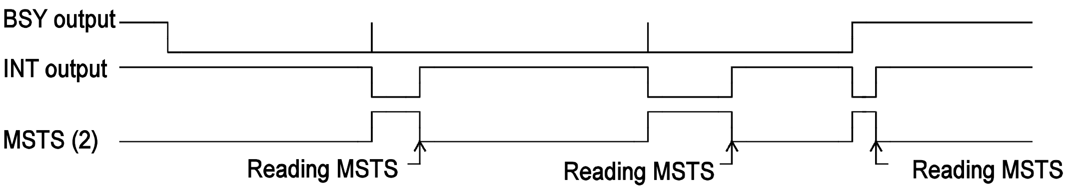
The #INT signal output can be masked by setting the RENV1 (environment setting 1) register.
If the #INT output is masked (RENV1.INTM = 1), and when the interrupt conditions are satisfied, the status will change. However, the #INT signal will not go LOW, but will remain HIGH.
While the interrupt conditions are satisfied and if the output mask is turned OFF (renv1.INTM = 0), the #INT output terminal will go LOW.

Read the interrupt status <MSTSW.SENI(bit2), SERR (bit 4), SINT (bit 5)> SENI = 1: Becomes 1 when IEND = 1 and a stop interrupt occurs. Becomes 0 by reading MSTSW. SERR = 1: Becomes 1 when an error interrupt occurs. Becomes 0 by reading REST. SINT = 1: Becomes 1 when an event interrupt occurs. Becomes 0 by reading RIST.	[MSTSW] (READ) 7 0 - | - | n | n | - | n | - | -
Set the interrupt mask <RENV1.INTM (bit 29)> 1: Mask #INT output.	[RENV1] (WRITE) 31 24 - | - | n | - | - | - | - | -
Setting a stop interrupt <RENV2.IEND (bit 27)> 1: Enable a stop interrupt.	[RENV2] (WRITE) 31 24 - | - | - | - | n | - | - | -
Select the stop interrupt mode <PRMD.MENI (bit 7)> 1: When there is data for the next operation in the pre-register, the PCL will not output a stop interrupt.	[PRMD] (WRITE) 7 0 n | - | - | - | - | - | - | -
Read the cause of the error interrupt <PREST: Read command> Copy the data in the RESET register (error interrupt cause) to BUF.	[Read command] F2h
Read the event interrupt cause <PRIST: Read command> Copy the data in the RIST register (event interrupt cause) to BUF.	[Read command] F3h
Set the event interrupt cause <WRIRQ: Write command> Write the BUF data to the RIRQ register (event interrupt cause).	[Write command] ACh

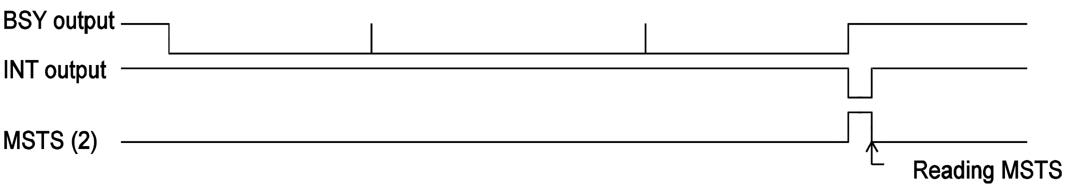
[Operation example with setting PRMD.MENI]

This is operation is used in the case of writing setting for next operation and the operation after that at the start.

1) When RENV2.IEND = 1 and PRMD.MENI = 0



2) When RENV2.IEND = 1 and PRMD.MENI = 1



Note: Even if IEND = 1 and MENI = 1, if no pre-register has been specified (a Start command has been written), interrupt signal is output.

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

Error interrupt cause	Cause (REST)	
	Bit	Bit name
Stopped by Comparator 1 conditions being satisfied (+SL)	0	ESC1
Stopped by Comparator 2 conditions being satisfied (-SL)	1	ESC2
Stopped by Comparator 3 conditions being satisfied	2	ESC3
Stopped by Comparator 4 conditions being satisfied	3	ESC4
Stopped by Comparator 5 conditions being satisfied	4	ESC5
Stopped by turning ON the +EL input	5	ESPL
Stopped by turning ON the -EL input	6	ESML
Stopped by turning ON the ALM input	7	ESAL
Stopped by turning ON the #CSTP input	8	ESSP
Stopped by turning ON the #CEMG input	9	ESEM
Deceleration stopped by turning ON the SD input	10	ESSD
(Always 0)	11	Not defined
Stopped by an operation data error.	12	ESDT
Simultaneously stopped with another axis due to an error stop on the other axis during an interpolation operation	13	ESIP
Stopped by an overflow of PA/PB input buffer counter occurrence	14	ESPO
Stopped by an over range count occurrence while positioning in an interpolation operation	15	ESAO
An EA/EB input error occurs (does not stop).	16	ESEE
A PA/PB input error occurs (does not stop).	17	ESPE

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

Event interrupt cause	Set cause (RIRQ)		Cause (RIST)	
	Bit	Bit name	Bit	Bit name
Automatic stop	0	IREN	0	ISEN
The next operation starts continuously	1	IRN	1	ISN
When it is possible to write an operation to the 2nd pre-register	2	IRNM	2	ISNM
When it is possible to write to the 2nd pre-register for Comparator 5	3	IRND	3	ISND
When acceleration starts	4	IRUS	4	ISUS
When acceleration ends	5	IRUE	5	ISUE
When deceleration starts	6	IRDS	6	ISDS
When deceleration ends	7	IRDE	7	ISDE
When the Comparator 1 conditions are satisfied	8	IRC1	8	ISC1
When the Comparator 2 conditions are satisfied	9	IRC2	9	ISC2
When the Comparator 3 conditions are satisfied	10	IRC3	10	ISC3
When the Comparator 4 conditions are satisfied	11	IRC4	11	ISC4
When the Comparator 5 conditions are satisfied	12	IRC5	12	ISC5
When the counter value is reset by a CLR signal input	13	IRCL	13	ISCL
When the counter value is latched by an LTC input	14	IRLT	14	ISLT
When the counter value is latched by an ORG input	15	IROL	15	ISOL
When the ±SD input is turned ON	16	IRSD	16	ISSD
When the +DR input changes	17	IRDR	17	ISPD
When the -DR input changes			18	ISMD
When the #CSTA input is turned ON	18	IRSA	19	ISSA

12. Electrical Characteristics

12-1. Absolute maximum ratings

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

12-2. Recommended operating conditions

Item	Symbol	Rating	Unit
Power supply voltage	V_{DD}	3.0 to 3.6	V
Ambient temperature	T_J	-40 to +85	˚C
Input rising time	T_r	≤50	ns
Input falling time	T_f	≤50	ns

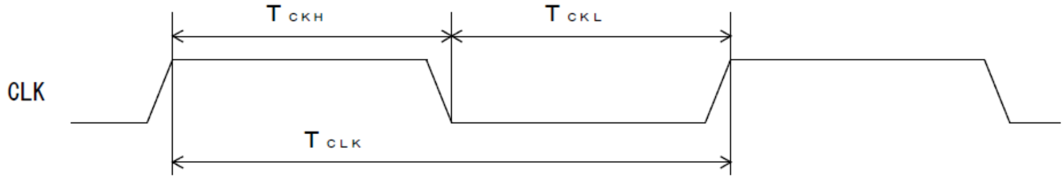
12-3. DC characteristics

Item	Symbol	Condition	Min.	Max.	Unit
Current consumption	I_{DD}	CLK=19.6608 MHz, before reset		150	mA
		CLK=19.6608 MHz, after reset (stop)		104	
		CLK=19.6608 MHz, highest 4-axes operation		119	
		CLK=30.0000 MHz, before reset		230	
		CLK=30.0000MHz, after reset (stop)		162	
		CLK=30.0000MHz, highest 4-axes operation		184	
Output leakage current	I_{OZ}		-1	1	μA
Input capacitance				10	pF
LOW input current	I_{IL}	Terminals have internal pull-up resisters.		-82.5	μA
		Others than those above.		-1	μA
HIGH input current	I_{IH}			1	μA
LOW input voltage	V_{IL}		-0.3	0.8	V
HIGH input voltage	V_{IH}		2.0	5.8	V
LOW output voltage	V_{OL}	I_{OL} = 6 mA		0.4	V
HIGH output voltage	V_{OH}	I_{OH} = -6 mA	V_{DD}-0.4		V
LOW output current	I_{OL}	V_{OL} = 0.4 V		6	mA
HIGH output current	I_{OH}	V_{OH} = V_{DD} -0.4 V		-6	mA
Internal pull up resistance	R_{UP}		40	240	K-ohm

Note: No load for all conditions.
Even though this LSI has a circuit to reduce consumption current, the consumption current increases because the function does not operate before reset.
The sign of current value is + to flow into and - to flow out.

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

Item	Symbol	Condition	Min.	Max.	Unit
Reference clock frequency	f_{CLK}			31.25	MHz
Reference clock cycle	T_{CLK}		32		ns
Reference clock HIGH width	T_{CKH}		13		ns
Reference clock LOW width	T_{CKL}		13		ns



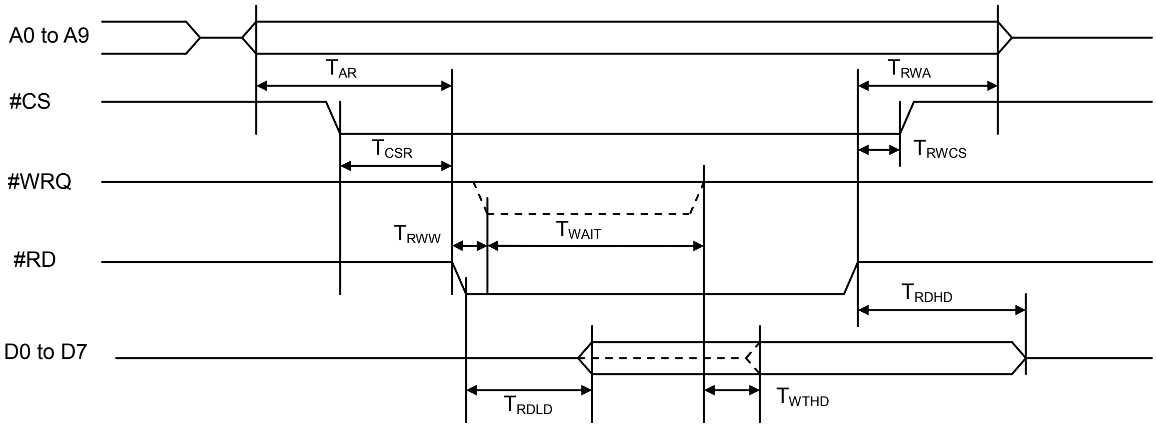
12-5. AC characteristics 2) (CPU- I/F)

12-5-1. CPU-I/F 1) (IF1 = H, IF0 = H) Z80

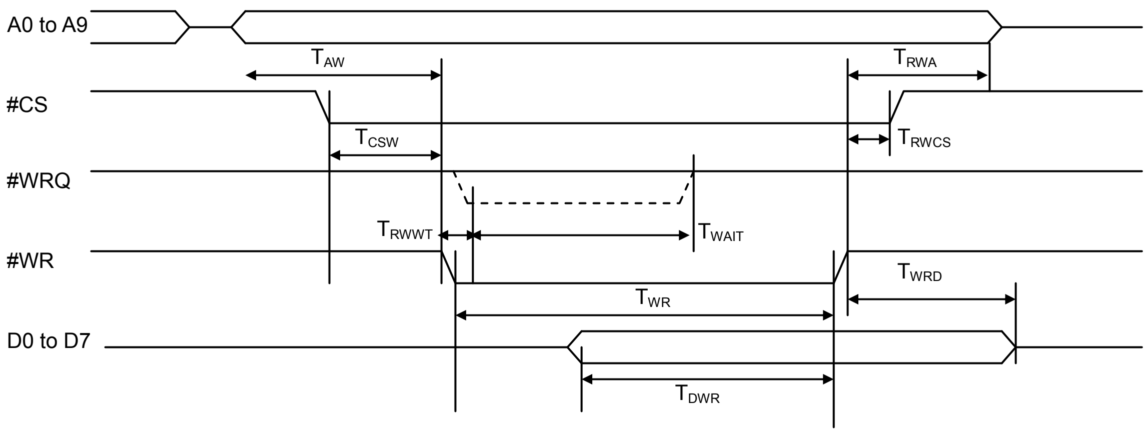
Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for #RD ↓	T_{AR}		11		ns
Address setup time for #WR ↓	T_{AW}		10		ns
Address hold time for #RD, #WR ↑	T_{RWA}		0		ns
#CS setup time for #RD ↓	T_{CSR}		3		ns
#CS setup time for #WR ↓	T_{CSW}		3		ns
#CS hold time for #RD, #WR ↑	T_{RWCS}		0		ns
#WRQ ON delay time for #RD, #WR ↓	T_{RWWT}	C_L = 40pF		15	ns
#WRQ signal LOW time	T_{WAIT}			$4T_{CLK}$	ns
Data output delay time for #RD ↓	T_{RDLD}	C_L = 40pF		27	ns
Data output delay time for #WRQ ↑	T_{WTHD}	C_L = 40pF		13	ns
Data float delay time for #RD ↓	T_{RDHD}	C_L = 40pF		23	ns
#WR signal width	T_{WR}	Note 1	6		ns
Data setup time for #WR ↑	T_{DWR}		8		ns
Data hold time for #WR ↑	T_{WRD}		0		ns

Note 1: When a #WRQ signal is output, the duration will be the interval between #WRQ = H and #WR = H.

<Read cycle>



<Write cycle>

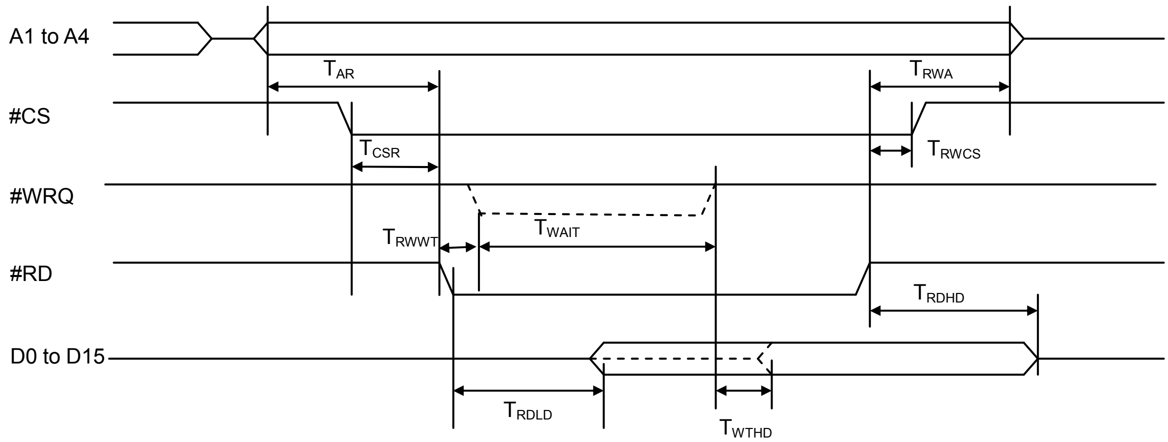


12-5-2. CPU-I/F 2) (IF1 = H, IF0 = L) 8086

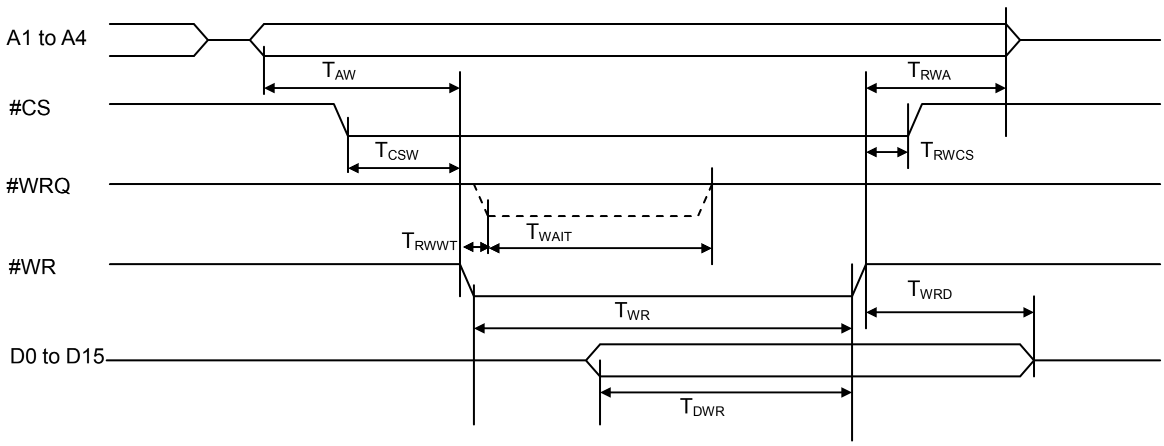
Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for #RD ↓	T_{AR}		11		ns
Address setup time for #WR ↓	T_{AW}		11		ns
Address hold time for #RD, #WR ↑	T_{RWA}		0		ns
#CS setup time for #RD ↓	T_{CSR}		3		ns
#CS setup time for #WR ↓	T_{CSW}		3		ns
#CS hold time for #RD, #WR ↑	T_{RWCS}		0		ns
#WRQ ON delay time for #RD, #WR ↓	T_{RWWT}	C_L = 40pF		15	ns
#WRQ signal LOW time	T_{WAIT}			$4T_{CLK}$	ns
Data output delay time for #RD ↓	T_{RDLD}	C_L = 40pF		27	ns
Data output delay time for #WRQ ↑	T_{WTHD}	C_L = 40pF		13	ns
Data float delay time for #RD ↑	T_{RDHD}	C_L = 40pF		23	ns
#WR signal width	T_{WR}	Note 1	6		ns
Data setup time for #WR ↑	T_{DWR}		8		ns
Data hold time for #WR ↑	T_{WRD}		0		ns

Note 1: When a #WRQ signal is output, the duration will be the interval between #WRQ = H and #WR = H.

<Read cycle>



<Write cycle>

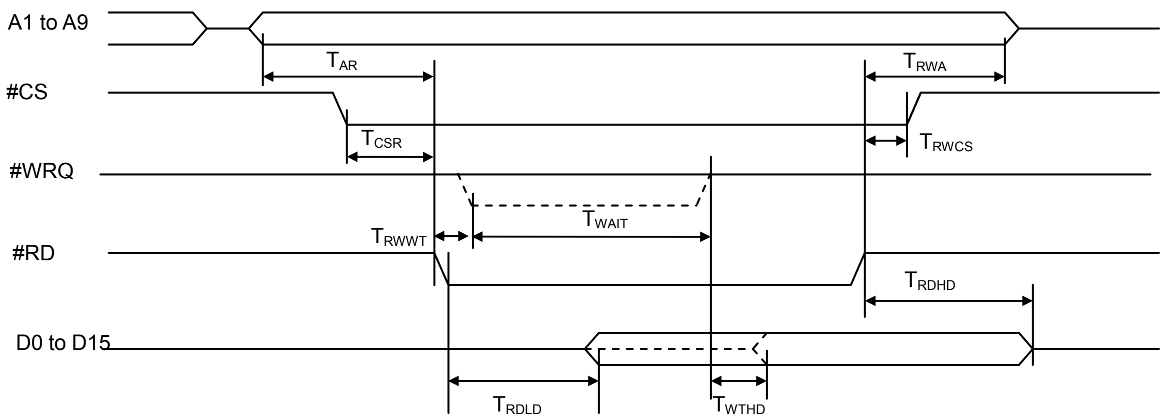


12-5-3. CPU-I/F 3) (IF1 = L, IF0 = L) H8

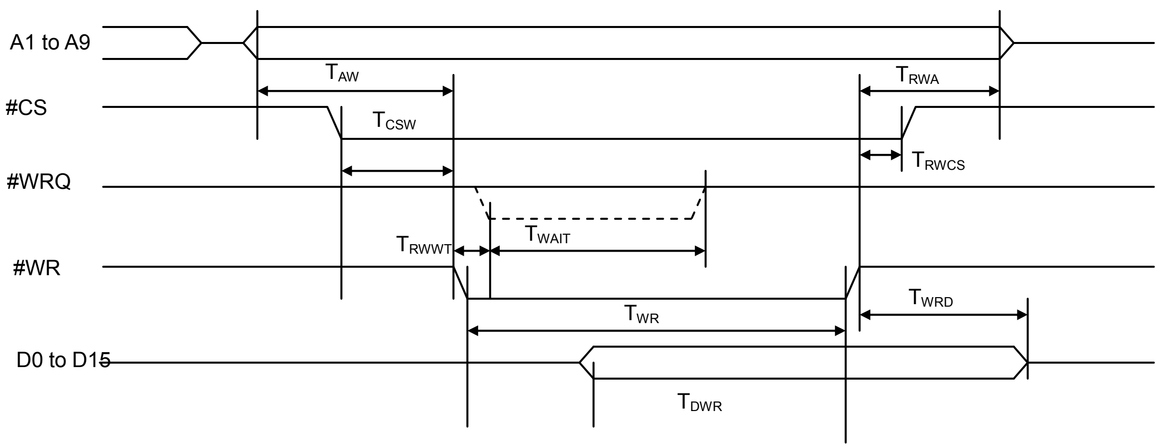
Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for #RD ↓	T_{AR}		11		ns
Address setup time for #WR ↓	T_{AW}		10		ns
Address hold time for #RD, #WR ↑	T_{RWA}		0		ns
#CS setup time for #RD↓	T_{CSR}		3		ns
#CS setup time for #WR ↓	T_{CSW}		3		ns
#CS hold time for #RD, #WR ↑	T_{RWCS}		0		ns
#WRQ ON delay time for #RD, #WR ↓	T_{RWWT}	C_L = 40pF		15	ns
#WRQ signal LOW time	T_{WAIT}			$4T_{CLK}$	ns
Data output delay time for #RD ↓	T_{RDLD}	C_L = 40pF		27	ns
Data output delay time for #WRQ ↑	T_{WTHD}	C_L = 40pF		13	ns
Data float delay time for #RD ↑	T_{RDHD}	C_L = 40pF		23	ns
#WR signal width	T_{WR}	Note 1	6		ns
Data setup time for #WR ↑	T_{DWR}		8		ns
Data hold time for #WR ↑	T_{WRD}		0		ns

Note 1: When a #WRQ signal is output, the duration will be the interval between #WRQ = H and #WR = H.

<Read cycle>



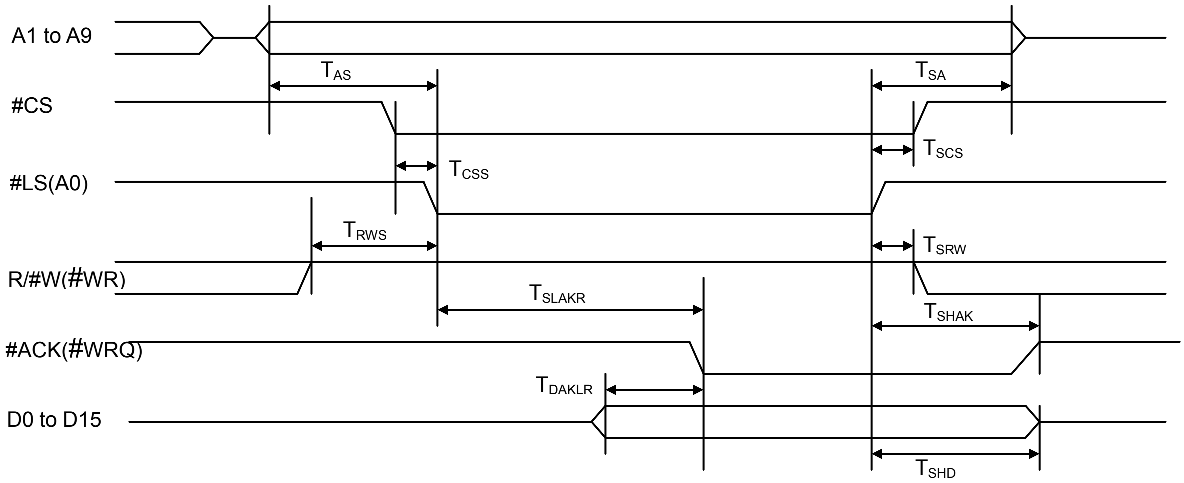
<Write cycle>



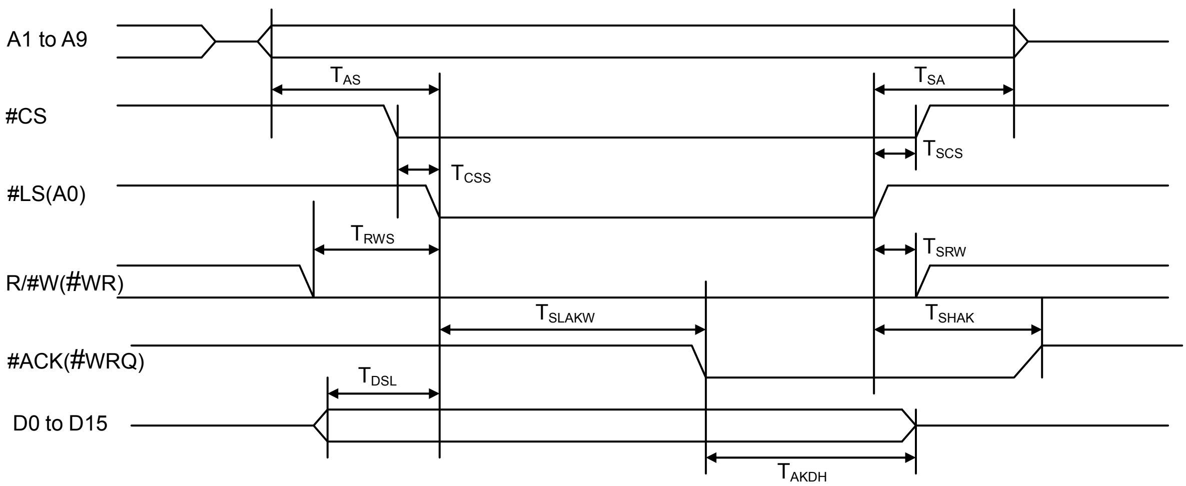
12-5-4. CPU-I/F 4) (IF1 = L, IF0 = L) 68000

Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for #LS ↓	T_{AS}		3		ns
Address hold time for #LS ↑	T_{SA}		3		ns
#CS setup time for #LS ↓	T_{CSS}		0		ns
#CS hold time for #LS ↑	T_{SCS}		3		ns
R/#W setup time for #LS ↓	T_{RWS}		6		ns
R/#W hold time for #LS ↑	T_{SRW}		3		ns
#ACK ON delay time for #LS ↓	T_{SLAKR}	C_L = 40pF	1T_{CLK}	7T_{CLK}	ns
	T_{SLAKW}	C_L = 40pF	1T_{CLK}	6T_{CLK}	ns
#ACK OFF delay time for #LS ↑	T_{SHAK}	C_L = 40pF	5	24	ns
Data output advance time for #ACK ↓	T_{DAKLR}	C_L = 40pF	T_{CLK} -3		ns
Data float delay time for #LS ↑	T_{SHD}	C_L = 40pF		31	ns
Data setup time for #LS ↑	T_{DSL}		12		ns
Data hold time for #ACK ↓	T_{AKDH}		0		ns

<Read cycle>



<Write cycle>

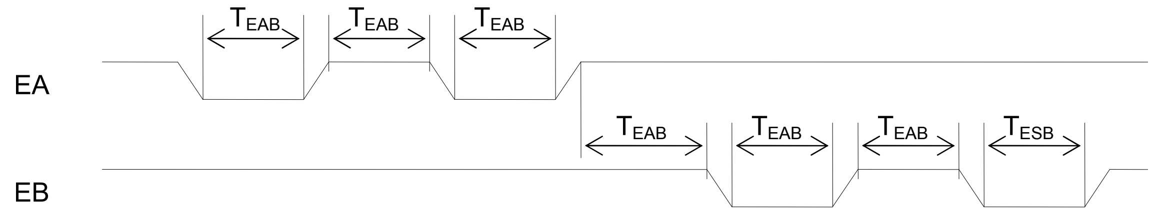


12-6. Operation timing (Common to all axes)

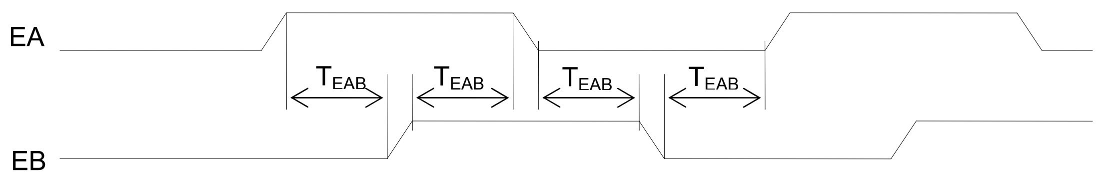
Item	Symbol	Condition	Min.	Max.	Unit
#RST input signal width		Note 1	10T_{CLK}		ns
CLR input signal width			2T_{CLK}		ns
EA, EB input signal width	T_{EAB}	Note 2	1T_{CLK} (3T_{CLK})		ns
EZ input signal width		Note 2	1T_{CLK} (3T_{CLK})		ns
PA, PB input signal width	T_{PAB}	Note 3	1T_{CLK} (3T_{CLK})		ns
ALM input signal width		Note 4	2T_{CLK}		ns
INP input signal width		Note 4	2T_{CLK}		ns
ERC output signal width		RENV1 bit 12 to 14 = 000	254T_{CLK}	255T_{CLK}	ns
		RENV1 bit 12 to 14 = 001	254 x 8T_{CLK}	255 x 8T_{CLK}	
		RENV1 bit 12 to 14 = 010	254 x 32T_{CLK}	255 x 32T_{CLK}	
		RENV1 bit 12 to 14 = 011	254 x 128T_{CLK}	255 x 128T_{CLK}	
		RENV1 bit 12 to 14 = 100	254 x 1024T_{CLK}	255 x 1024T_{CLK}	
		RENV1 bit 12 to 14 = 101	254 x 4096T_{CLK}	255 x 4096T_{CLK}	
		RENV1 bit 12 to 14 = 110	254 x 8192T_{CLK}	255 x 8192T_{CLK}	
		RENV1 bit 12 to 14 = 111	LEVEL output		
+EL, -EL input signal width		Note 4	2T_{CLK}		ns
+SD, -SD input signal width		Note 4	2T_{CLK}		ns
ORG input signal width		Note 4	2T_{CLK}		ns
+DR, -DR input signal width		Note 5	2T_{CLK}		ns
#PE input signal width		Note 5	2T_{CLK}		ns
PCS input signal width			2T_{CLK}		ns
LTC input signal width			2T_{CLK}		ns
#CSTA Output signal width			8T_{CLK}		ns
#CSTA Input signal width			5T_{CLK}		ns
#CSTP Output signal width			8T_{CLK}		ns
#CSTP Input signal width			5T_{CLK}		ns
#BSY signal ON delay time	T_{CMDBSY}			5T_{CLK}	ns
	T_{STABSY}			7T_{CLK}	ns
Start delay time	T_{CMDPLS}			15T_{CLK}	ns
	T_{STAPLS}			17T_{CLK}	ns

Note 1: Longer than 10 cycles of CLK signal is necessary to be input while the #RST terminal is LOW.
Note 2: If the input filter is ON < EINF (bit 18) = 1 in RENV2 >, the minimum time will be 3T_{CLK}.
Note 3: If the input filter is ON < PINF (bit 19) = 1 in RENV2 >, the minimum time will be 3T_{CLK}.
Note 4: If the input filter is ON < FLTR (bit 26) = 1 in RENV1 >, the minimum time will be 80T_{CLK}.
Note 5: If the input filter is ON < DRF (bit 27) = 1 in RENV1 >, the minimum time will be 655,360T_{CLK}.

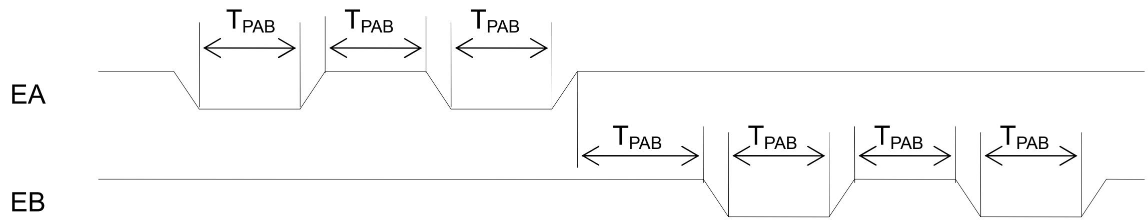
1) When the EA, EB inputs are in the Two-pulse mode



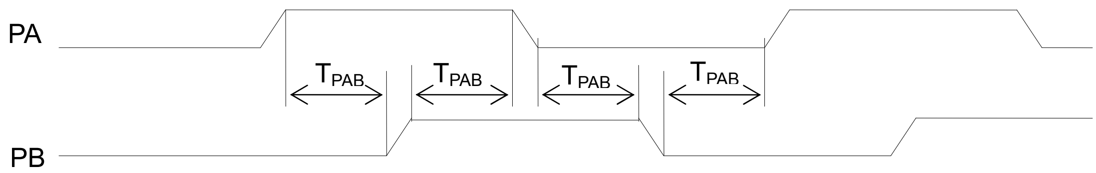
2) When the EA, EB inputs are in the 90 ° phase-difference mode



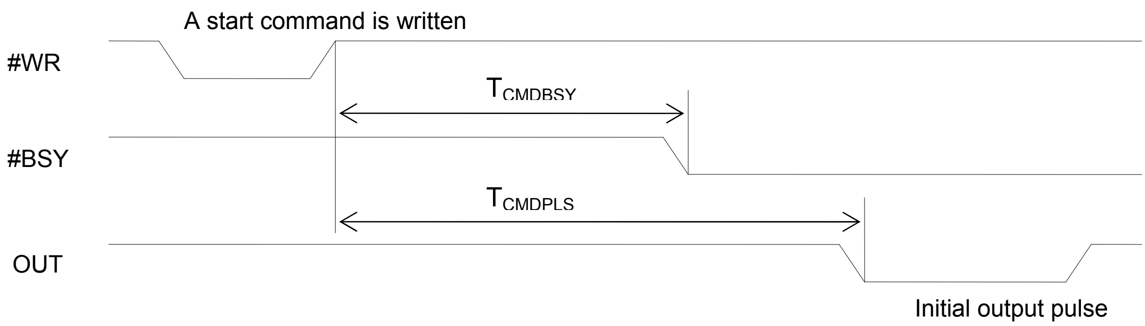
3) When the PA, PB inputs are in the Two-pulse mode



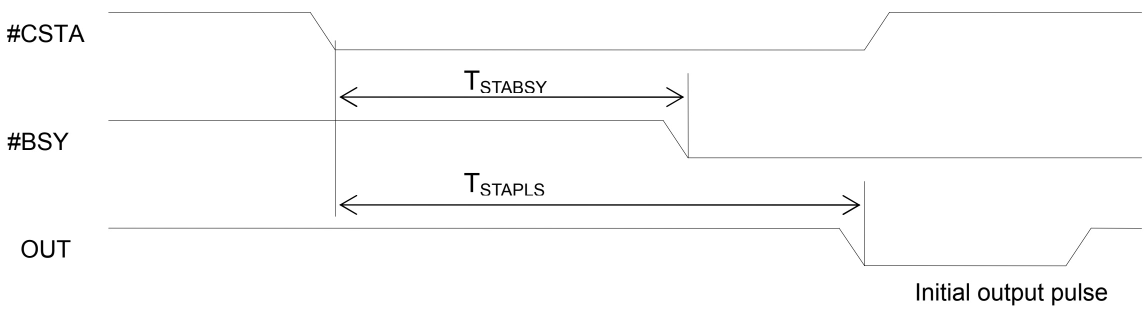
4) When the PA, PB inputs are in the 90° phase-difference mode



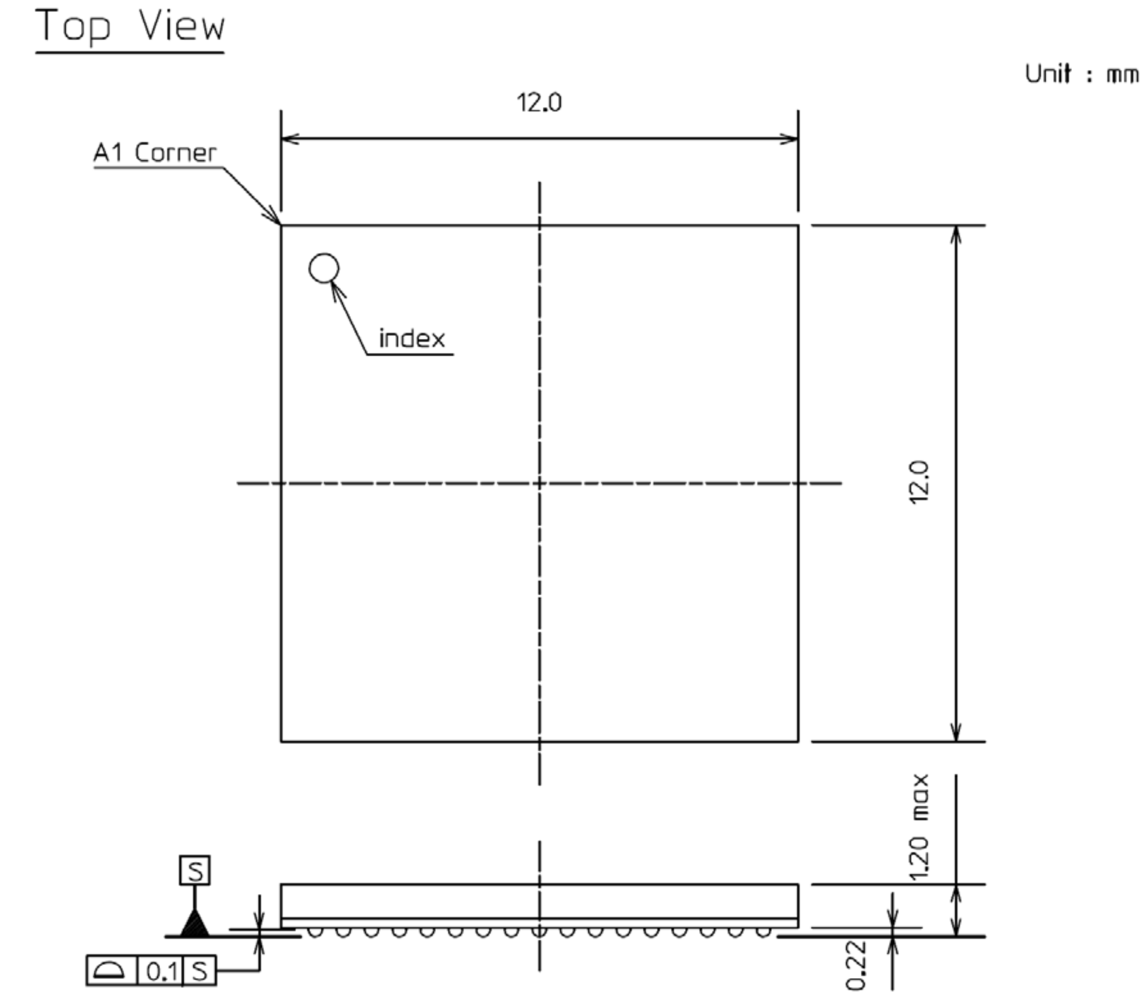
5) Timing for the command start (when I/M = H, and B/#W = H)

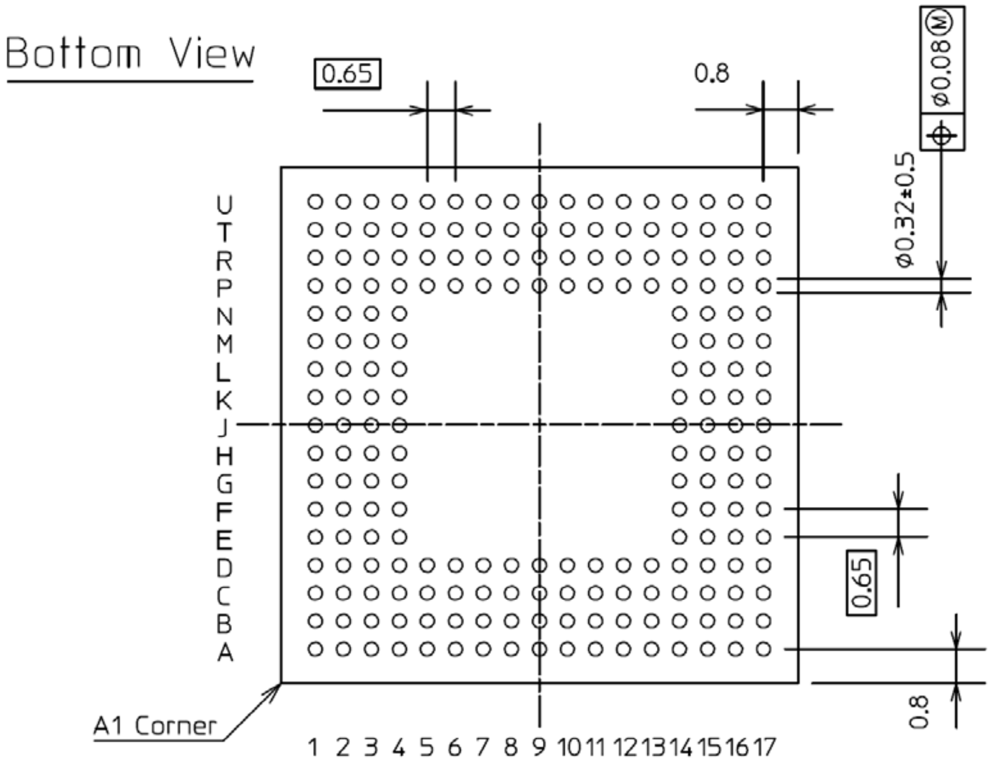


6) Simultaneous start timing



13. External Dimensions





Appendix 1: List of commands

<Operation commands>

COMB0	Symbol	Description	COMB0	Symbol	Description
05h	CMEMG	Emergency stop	50h	STAFL	FL constant speed start
06h	CMSTA	#CSTA output (simultaneous start)	51h	STAFH	FH constant speed start
07h	CMSTP	#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			

< General-purpose port control commands>

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

<Control commands>

COMB0	Symbol	Description	COMB0	Symbol	Description
00h	NOP	(Invalid command)	26h	PRECAN	Make the operation pre-register undetermined
04h	SRST	Software reset	27h	PCPCAN	Make the RCMP5 operation pre-register (PRCP5) undetermined
20h	CUN1R	Reset COUNTER1 (command position)	28h	STAON	Substitute a PCS terminal input
21h	CUN2R	Reset COUNTER2 (mechanical position)	29h	LTCH	Substitute a LTC terminal input
22h	CUN3R	Reset COUNTER3 (deflection counter)	2Ah	SPSTA	Uses the same process as the #CSTA input, but for own axis
23h	CUN4R	Reset COUNTER4 (general-purpose)	2Bh	PRESHF	Shift the operation pre-register data
24h	ERCOUT	Output an ERC signal	2Ch	PCPSHF	Shift the RCMP5 pre-register data
25h	ERCRST	Reset the ERC signal	4Fh	PRSET	Make data in a pre-register determined as speed pattern change data

<Register control commands>

No	Detail	Register					2nd pre-register				
		Name	Read command		Write command		Name	Read command		Write command	
			COMB0	Symbol	COMB0	Symbol		COMB0	Symbol	COMB0	Symbol
1	Feed amount, target position	RMV	D0h	RRMV	90h	WRMV	PRMV	C0h	RPRMV	80h	WPRMV
2	Initial speed	RFL	D1h	RRFL	91h	WRFL	PRFL	C1h	RPRFL	81h	WPRFL
3	Operation speed	RFH	D2h	RRFH	92h	WRFH	PRFH	C2h	RPRFH	82h	WPRFH
4	Acceleration rate	RUR	D3h	RRUR	93h	WRUR	PRUR	C3h	RPRUR	83h	WPRUR
5	Deceleration rate	RDR	D4h	RRDR	94h	WRDR	PRDR	C4h	RPRDR	84h	WPRDR
6	Speed magnification rate	RMG	D5h	RRMG	95h	WRMG	PRMG	C5h	RPRMG	85h	WPRMG
7	Ramping-down point	RDP	D6h	RRDP	96h	WRDP	PRDP	C6h	RPRDP	86h	WPRDP
8	Operation mode	RMD	D7h	RRMD	97h	WRMD	PRMD	C7h	RPRMD	87h	WPRMD
9	Circular interpolation center	RIP	D8h	RRIP	98h	WRIP	PRIP	C8h	RPRIP	88h	WPRIP
10	Acceleration S-curve range	RUS	D9h	RRUS	99h	WRUS	PRUS	C9h	RPRUS	89h	WPRUS
11	Deceleration S-curve range	RDS	DAh	RRDS	9Ah	WRDS	PRDS	CAh	RPRDS	8Ah	WPRDS
12	Feed amount correction speed	RFA	DBh	RRFA	9Bh	WRFA					
13	Environment setting 1	RENV1	DCh	RRENV1	9Ch	WRENV1					
14	Environment setting 2	RENV2	DDh	RRENV2	9Dh	WRENV2					
15	Environment setting 3	RENV3	DEh	RRENV3	9Eh	WRENV3					
16	Environment setting 4	RENV4	DFh	RRENV4	9Fh	WRENV4					
17	Environment setting 5	RENV5	E0h	RRENV5	A0h	WRENV5					
18	Environment setting 6	RENV6	E1h	RRENV6	A1h	WRENV6					
19	Environment setting 7	RENV7	E2h	RRENV7	A2h	WRENV7					
20	COUNTER1 (command position)	RCUN1	E3h	RRCUN1	A3h	WRCUN1					
21	COUNTER2 (mechanical position)	RCUN2	E4h	RRCUN2	A4h	WRCUN2					
22	COUNTER3 (deflection counter)	RCUN3	E5h	RRCUN3	A5h	WRCUN3					
23	COUNTER4 (general purpose)	RCUN4	E6h	RRCUN4	A6h	WRCUN4					
24	Data for comparator 1	RCMP1	E7h	RRCMP1	A7h	WRCMP 1					
25	Data for comparator 2	RCMP2	E8h	RRCMP2	A8h	WRCMP 2					
26	Data for comparator 3	RCMP3	E9h	RRCMP3	A9h	WRCMP 3					
27	Data for comparator 4	RCMP4	EAh	RRCMP4	AAh	WRCMP 4					
28	Data for comparator 5	RCMP5	EBh	RRCMP5	ABh	WRCMP 5	PRCP5	CBh	RPRCP5	8Bh	WPRCP5
29	Event INT setting	RIRQ	ECh	RRIRQ	ACh	WRIRQ					

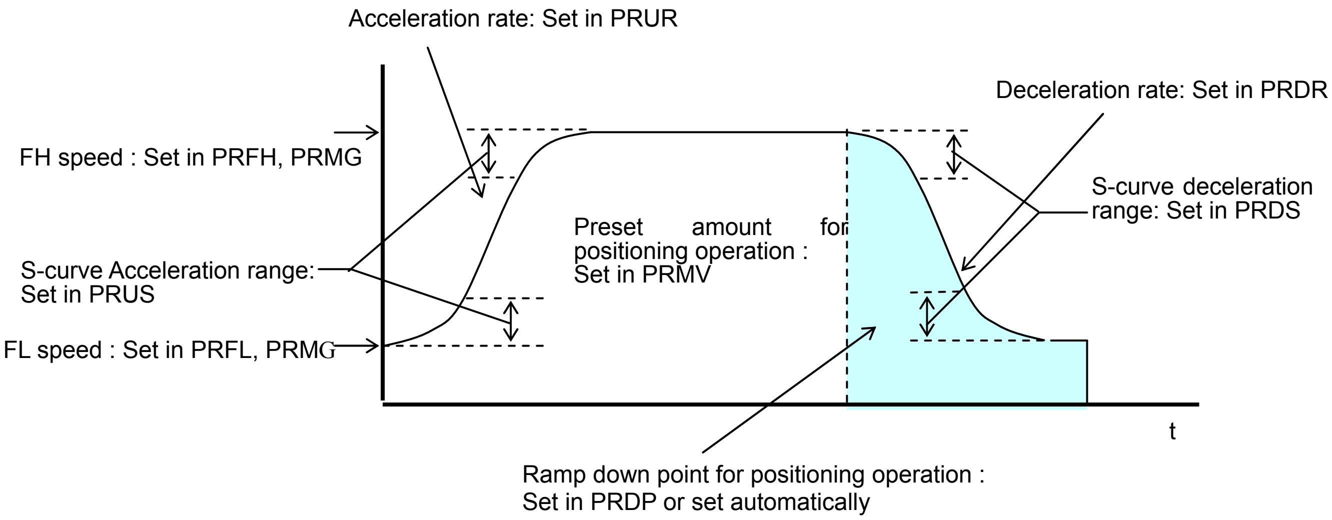
No	Detail	Register					2nd pre-register				
		Name	Read command		Write command		Name	Read command		Write command	
			COMB0	Symbol	COMB0	Symbol		COMB0	Symbol	COMB0	Symbol
30	COUNTER1 latched data	RLTC1	EDh	RRLTC1							
31	COUNTER2 latched data	RLTC2	EEh	RRLTC2							
32	COUNTER3 latched data	RLTC3	EFh	RRLTC3							
33	COUNTER4 latched data	RLTC4	F0h	RRLTC4							
34	Extension status	RSTS	F1h	RRSTS							
35	Error INT status	REST	F2h	RREST	B2h	WREST					
36	Event INT status	RIST	F3h	RRIST	B3h	WRIST					
37	Positioning counter	RPLS	F4h	RRPLS							
38	EZ counter, speed monitor	RSPD	F5h	RRSPD							
39	Ramping-down point	PSDC	F6h	RPSDC							
40	Circular interpolation stepping number	RCI	FCh	RRCI	BCh	WRCI	PRCI	CCh	RPRCI	8Ch	WPRCI
41	Circular interpolation stepping counter	RCIC	FDh	RRCIC							
42	Interpolation status	RIPS	FFh	RRIPS							

Appendix 2: Setting speed pattern

Pre-register	Description	Bit length setting range	Setting range	register
PRMV	Positioning amount	32	-2,147,483,648 to +2,147,483,647 (80000000h) (7FFFFFFFh)	RMV
PRFL	Initial speed	16	1 to 65,535 (0FFFFh)	RFL
PRFH	Operation speed	16	1 to 65,535 (0FFFFh)	RFH
PRUR	Acceleration rate	16	1 to 65,535 (0FFFFh)	RUR
PRDR	Deceleration rate Note 1	16	0 to 65,535 (0FFFFh)	RDR
PRMG	Speed magnification rate	12	2 to 4,095 (0FFFh)	RMG
PRDP	Ramping-down point	24	0 to 16,777,215 (0FFFFFFh)	RDP
PRUS	S-curve acceleration range	15	0 to 32,767 (7FFFh)	RUS
PRDS	S-curve deceleration range	15	0 to 32,767 (7FFFh)	RDS

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

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



◆ PRFL: FL speed setting register (16-bit)
Specify the speed for FL constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 65,535 (0FFFFh).
The speed will be calculated from the value in PRMG.

$$\text{FL speed [pps]} = \text{PRFL x} \frac{\text{Reference clock frequency [Hz]}}{(\text{PRMG}+1)\times 65536}$$

◆ PRFH: FH speed setting register (16-bit)
Specify the speed for FH constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 65,535 (0FFFFh).
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.

$$\text{FH speed [pps]} = \text{PRFH x} \frac{\text{Reference clock frequency [Hz]}}{(\text{PRMG}+1)\times 65536}$$

- PRUR: Acceleration rate setting register (16-bit)

Specify the acceleration characteristic for high speed operations (acceleration/deceleration operations), in the range of 1 to 65,535 (0FFFFh)

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

1) Linear acceleration (PRMD.MSMD = 0)

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

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

$$\text{Acceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRUR} + 1) \times 8}{\text{Reference clock frequency [Hz]}}$$

3) S-curve acceleration with a linear range (PRMD.MSMD= and PRUS register >0)

$$\text{Acceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL} + 2 \times \text{PRUS}) \times (\text{PRUR} + 1) \times 4}{\text{Reference clock frequency [Hz]}}$$

- PRDR: Deceleration rate setting register (16-bit)

Normally, specify the deceleration characteristics for high speed operations (acceleration/deceleration operations) in the range of 1 to 65,535 (0FFFFh).

Even if the ramping-down point is set to automatic (PRMD.MSDP = 0), the value placed in the PRDR register will be used as the deceleration rate.

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

When the ramping-down point is set to automatic, there are the following restrictions.

While in linear interpolation 1 or circular interpolation operation, and when constant synthesized speed operation (PRMD.MIPF = 1 is selected, make deceleration time same as acceleration time.

For other operations, arrange time so that (deceleration time) ≤ acceleration time x 2.

If setting otherwise, the axis may not decrease the speed to the specified FL speed when stopping. In this case, use a manual ramping-down point (PRMD.MSDP = 1).

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

1) Linear deceleration (PRMD.MSMD = 0)

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

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

$$\text{Deceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL}) \times (\text{PRDR} + 1) \times 8}{\text{Reference clock frequency [Hz]}}$$

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

$$\text{Deceleration time [s]} = \frac{(\text{PRFH} - \text{PRFL} + 2 \times \text{PRDS}) \times (\text{PRDR} + 1) \times 4}{\text{Reference clock frequency [Hz]}}$$

◆ PRMG: Magnification rate register (12-bit)

Specify the relationship between the PRFL and PRFH settings and the speed, in the range of 2 to 4,095 (0FFFh). As the magnification rate is increased, the speed setting units will tend to be approximations. Normally set the magnification rate as low as possible.

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

$$\text{Magnification rate} = \frac{\text{Reference clock frequency[Hz]}}{(\text{PRMG} + 1) \times 65536}$$

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

Setting	Magnification rate	Output speed range	Setting	Magnification rate	Output speed range
2999 (0BB7h)	0.1	0.1 to 6,553.5	59 (3Bh)	5	5 to 327,675
1499 (5DBh)	0.2	0.2 to 13,107.0	29 (1Dh)	10	10 to 655,350
599 (257h)	0.5	0.5 to 32,767.5	14 (0Eh)	20	20 to 1,310,700
299 (12Bh)	1	1 to 65,535	5 (5h)	50	50 to 3,276,750
149 (95h)	2	2 to 131,070	2 (2h)	100	100 to 6,553,500

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

Specify the value used to determine the deceleration start point for positioning operations that include acceleration and deceleration.

The meaning of the value specified in the PRDP varies according to the "ramping-down point setting method", (MSDP) in the PRMD register.

<When set to manual (PRMD.MSDP=1)>

The number of pulses at which to start deceleration, set in the range of 0 to16,777,215 (0FFFFFFh).

The optimum value for the ramping-down point can be calculated as shown in the equation below.

1) Linear deceleration (PRMD.MSMD=0)

$$\text{Optimum value [Number of pulses]} = \frac{(\text{PRFH}^2 - \text{PRFL}^2) \times (\text{PRDR} + 1)}{(\text{PRMG} + 1) \times 32768}$$

However, the optimum value for a triangle start, without changing the value in the PRFH register while turning OFF the FH correction function (MADJ = 1 in the PRMD register) will be calculated as shown the equation below.

(When using idling control, assign the value (subtracts the number of idling pulses from the value place in the PRMV register) to PRMV in the equation below. The number of idling pulses will be "1 to 6" when IDL = 2 to 7 in RENV5.)

$$\text{Optimum value [Number of pulses]} = \frac{\text{PRMV} \times (\text{PRDR} + 1)}{\text{PRUR} + \text{PRDR} + 2}$$

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

$$\text{Optimum value [Number of pulses]} = \frac{(PRFH^2 - PRFL^2) \times (PRDR + 1) \times 2}{(PRMG + 1) \times 32768}$$

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

$$\text{Optimum value [Number of pulses]} = \frac{(PRFH + PRFL) \times (PRFH - PRFL + 2 \times PRDS) \times (PRDR + 1)}{(PRMG + 1) \times +32768}$$

Start deceleration at the point when the (positioning counter value) ≤ (PRDP set value).

<When set to automatic (PRMD.MSDP = 0)>
This is an offset value for the automatically set ramping-down point. Set in the range of -8,388,608 (800000h) to 8,388,607 (7FFFFFFh).
When the offset value is a positive number, the axis will start deceleration at an earlier stage and will feed at the FL speed after decelerating. When a negative number is entered, the deceleration start timing will be delayed. If the offset is not required, set to zero.

When the value for the ramping-down point is smaller than the optimum value, the speed when stopping will be faster than the FL speed. On the other hand, if it is larger than the optimum value, the axis will feed at FL constant speed after decelerating is complete.

◆ PRUS: S-curve acceleration range register (15-bit)
Specify the S-curve acceleration range for S-curve acceleration/deceleration operations in the range of 1 to 32,767 (7FFFh).
The S-curve acceleration range S_{SU} will be calculated from the value placed in PRMG.

$$S_{SU} \text{ [pps]} = PRUS \times \frac{\text{Reference clock frequency[Hz]}}{(PRMG + 1) \times 65536}$$

In other words, speeds between the FL speed and (FL speed + S_{SU}), and between (FH speed - S_{SU}) and the FH speed, will be S-curve acceleration operations. Intermediate speeds will use linear acceleration.
However, if zero is specified, "(PRFH - PRFL)/2" will be used for internal calculations, and the operation will be an S-curve acceleration without a linear component.

◆ PRDS: S-curve deceleration range setting register (15-bit)
Specify the S-curve deceleration range for S-curve acceleration/deceleration operations in the range of 1 to 32,767 (7FFFh).
The S-curve acceleration range S_{SD} will be calculated from the value placed in PRMG.

$$S_{SD} \text{ [pps]} = PRDS \times \frac{\text{Reference clock frequency[Hz]}}{(PRMG + 1) \times 65536}$$

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

[Handling Precautions]

1. Design precautions

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

2. Precautions for transporting and storing LSIs

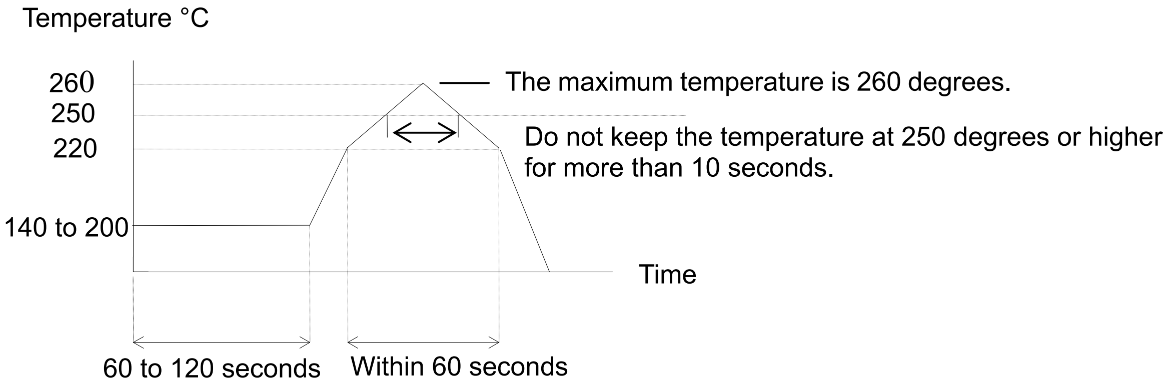
1) Always handle LSIs carefully and keep them in their packages. Throwing or dropping LSIs may damage them.
2) Do not store LSIs in a location exposed to water droplets or direct sunlight.
3) Do not store the LSI in a location where corrosive gases are present, or in excessively dusty environments.
4)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

3. Precautions for installation

1) In order to prevent damage caused by static electricity, pay attention to the following.
- Make sure to ground all equipment, tools, and jigs that are present at the work site.
- Ground the work desk surface using a conductive mat or similar apparatus (with an appropriate resistance factor). However, do not allow work on a metal surface, which can cause a rapid change in the electrical charge on the LSI (if the charged LSI touches the surface directly) due to extremely low resistance.
- When picking up an LSI using a vacuum device, provide anti-static protection using a conductive rubber pick up tip. Anything which contacts the leads should have as high a resistance as possible.
- When using a pincer that may make contact with the LSI terminals, use an anti-static model. Do not use a metal pincer, if possible.
- Store unused LSIs in a PC board storage box that is protected against static electricity, and make sure there is adequate clearance between the LSIs. Never directly stack them on each other, as it may cause friction that can develop an electrical charge.

2) Operators must wear wrist straps which are grounded through approximately 1M-ohm of resistance.
3) Use low voltage soldering devices and make sure the tips are grounded.
4) Do not store or use LSIs, or a container filled with LSIs, near high-voltage electrical fields, such those produced by a CRT.
5) This LSI should be mounted by reflow method of infrared, hot air or combination of infrared and hot air.
6) If LSIs are purchased in less than one unit of packaging box, please dry a package before flow. IT should be dried at 125±5°C for 20 to 36 hours. The LSI must not be exposed to completely dry environment more than 2 times.

7) The maximum temperature of plastic surface is 260 degrees. A peak temperature of the surface of a package body should not exceed 260 degrees and do not keep the temperature at 250 degrees or higher for more than 10 seconds.



8) In reflows, change of temperature when cooling should be less than 3 degrees per second.
9) Up to 2 reflows is allowed.

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.

August 3, 2009

No. DA70122-1E

* The specifications may be changed without notice for improvement.

NPM Nippon Pulse Motor Co., Ltd.

Tokyo Office: **Nippon Pulse Motor Co., Ltd.**
Tachihi Bldg. No. 3, 1, Sakae-cho 6 -Chome, Tachikawa City, Tokyo 190-0003, Japan
Phone: +81-42-534-7701 Fax: +81-42-534-0026
Web: http://www.npm.co.jp E-mail: int-l@npm.co.jp

London Office: **Nippon Pulse Motor Co., Ltd.**
Vista Business Centre 50 Salisbury Road, Hounslow, Middlesex TW4 6JQ UK
Phone: +44-20-8538-0315 Fax: +44-20-8538-0316
Web: http://www.npm.co.jp E-mail: int-l@npm.co.jp

USA: **Nippon Pulse America, Inc.**
4 Corporate Drive, Radford, VA 24141, U.S.A.
Phone: +1-540-633-1677 Fax: +1-540-633-1674
Web: http://www.nipponpulse.com E-mail: info@nipponpulse.com

China: **Nippon Pulse Shanghai Co., Ltd.**
Room 1072, No.555, Pudongdadao Road, Shanghai, 200120, China
Phone: +86-21-6859-2622/2623 Fax: +86-21-6859-2628
Web: http://www.npmchina.com E-mail: yj@npmshanghai.sina.net

MNAL. No. PCL-6045BL-1 1B-5205-0.5 (5205) ims