

Pulse Control LSI for Controlling a Serial Bus

## User's Manual

---

PCD2112

---

3rd version created October 5, 2009

**NPM** Nippon Pulse Motor Co., Ltd.

### [Preface]

We thank you for considering the use of our serial bus pulse control LSI "PCD2112."  
Before using the PCD2112, we ask you to read this manual thoroughly to become familiar with the product.  
For details about installing this LSI, please refer to the "Handling Precautions" at the end of this manual.

### [Precautions]

- 1) Transmission or copying all or any part of this manual is prohibited without prior written approval.
- 2) The specifications provided in this manual may be changed without prior notice, to improve our product's performance or quality.
- 3) This manual was created with the utmost care. However, if you have any questions, find errors or missing parts in the manual, please let us know.
- 4) Nippon Pulse Motor is not liable for any results occurred in using this manual, regardless any problems or errors have been reported as mentioned in 3) above.

### [Precautions for the indications]

- 1) Terminal names and signal names preceded by number sign (#) mean that the item uses negative logic.  
Ex.: #INT: This means that the INT terminal uses negative logic.  
Negative logic is also shown as  $\overline{\text{INT}}$ .
- 2) Registers with bit names, such as environment setting registers, are expressed as REGISTER NAME.BIT NAME.  
Ex.: RENV1.PMD: Refers to the setting of the PMD bit in the RENV1 register.

# Table of Contents

1. Functions and Features .....	1
1-1. Outline .....	1
1-2. Features .....	1
2. Specifications .....	2
2-1. Basic specifications .....	2
2-2. Pin assignment diagram .....	3
2-3. Circuit block diagram .....	3
2-4. List of terminal names, functions, and pin assignments .....	4
2-5. System mode .....	5
2-5-1. CPU connection mode (MODE = LOW) .....	6
2-5-2. Stand-alone operation (MODE = HIGH) .....	7
3. Terminal Descriptions .....	8
3-1. Outline of the terminal functions .....	8
3-2. Terminal functions in each system mode .....	9
3-3. Terminal functions .....	10
3-3-1. CLK [7] .....	10
3-3-2. #RST [3] .....	10
3-3-3. MODE [5] .....	10
3-3-4. MTYP [33] .....	10
3-3-5. #INT [4] .....	11
3-3-6. #SCK [40], DI [35], DO [36], #CS [38], #CSO [34] .....	11
3-3-7. PA/PDR [9], PB/MDR [10] .....	11
3-3-8. A0/PS0 [44], A1/PS1 [45], PS2 [46], PS3 [47], PS4 [48] .....	12
3-3-9. EA [11], EB [12], EZ [13] .....	12
3-3-10. #BSY/#END [20] .....	12
3-3-11. SD [15] .....	12
3-3-12. PELL [22], MELL [21] .....	13
3-3-13. PEL [17] (+EL), MEL [16] (-EL) .....	13
3-3-14. ORG [14] .....	13
3-3-15. #STA [42] .....	13
3-3-16. STP [43] .....	13
3-3-17. OUT/PH1 [26], DIR/PH2 [27] .....	14
3-3-18. ERC/CDW [19] .....	14
3-3-19. UB [24] .....	14
3-3-20. INP/FH [23] .....	14
3-3-21. P0 [1], P1 [2] .....	15
3-3-22. P2/PH3 [28], P3/PH4 [29] .....	15
3-3-23. VDD [8, 30, 41], GND [6, 18, 25, 31, 32, 37, 39] .....	15

4. Memory Map .....	16
4-1. Detailed memory map .....	17
5. Serial Communication Format .....	18
5-1. Bus control command .....	18
5-2. Basic operation .....	19
5-3. Reading data from PCD2112 .....	19
5-4. Writing data to PCD2112 .....	19
5-5. Reading the bus status .....	20
5-6. Moving to the EEPROM access mode .....	20
5-7. Ending the EEPROM access mode .....	20
5-8. EEPROM control .....	21
5-9. How to identify a PCD when multiple PCDs are connected .....	22
6. Command .....	23
6-1. Operation command .....	23
6-1. Operation command .....	24
6-1-1. Start command .....	24
6-1-2. Remaining pulses start command .....	24
6-1-3. Speed change command .....	24
6-1-4. Stop command .....	24
6-2. Control command .....	25
6-2-1. Software reset command .....	25
6-2-2. Counter reset command .....	25
6-2-3. ERC output control command .....	25
6-2-4. STA output control command .....	25
6-2-5. Other commands .....	25
7. Registers .....	26
7-1. RMD register (Address: 23h to 24h) .....	26
7-2. RMV register (Address: 1Fh to 22h) .....	26
7-3. RMG register (Address: 14h to 15h) .....	27
7-4. RFL register (Address: 19h to 1Ah) .....	27
7-5. RFH1, RFH2 register (FH1 address: 1Bh to 1Ch, FH2 address: 1Dh to 1Eh,) .....	27
7-6. RUR register (Address: 12h to 13h) .....	28
7-7. RDR register (Address: 10h to 11h) .....	28
7-8. RUS register (Address: 0Eh to 0Fh) .....	28
7-9. RDS register (Address: 0Ch to 0Dh) .....	28
7-10. RDP register (Address: 16h to 18h) .....	28
7-11. RENV1 register (Address: 08h to 0Bh) .....	29
7-12. RENV2 register (Address: 05h to 07h) .....	31
7-13. RSTS registers (Address: 26h to 29h) .....	33
7-14. RIST registers (Address: 2Ah to 2Bh) .....	34
7-15. RSPD register (Address: 36h to 37h) .....	35

7-16. RCUN register (Address: 2Eh to 31h) .....	35
7-17. RDWC register (Address: 32h to 35h) .....	35
7-18. RIOP register (Address: 2Ch to 2Dh) .....	35
<b>8. Operation Modes (RMD.MOD) .....</b>	<b>36</b>
8-1. Continuous operation (MOD = "0000") .....	36
8-2. Origin return operation (MOD = "0001") .....	36
8-3. Origin return operation with a feed amount limit (MOD = "0010") .....	36
8-4. Move out of the origin position (MOD = "0011") .....	36
8-5. Move out of EL (MOD = "0100") .....	36
8-6. Positioning operation (MOD="0101") .....	37
8-7. Timer operation (MOD="0110") .....	37
8-8. Continuous operation using the DR input (MOD="0111") .....	37
8-9. Inching operation using the DR input (MOD="1000") .....	37
8-10. Continuous operation using a pulsar (MOD="1001") .....	38
8-11. Positioning operations using a pulsar (MOD="1010") .....	38
<b>9. Speed Patterns .....</b>	<b>39</b>
9-1. Speed patterns .....	39
9-2. Set speed pattern .....	40
9-2-1. RMV: Positioning amount setting register (28 bits) .....	41
9-2-2. RFL: FL speed setting register (13 bits) .....	41
9-2-3. RFH1, RFH2: FH speed setting register (13 bits) .....	41
9-2-4. RUR: Acceleration rate setting register (16 bits) .....	41
9-2-5. RDR: Deceleration rate setting register (16 bits) .....	42
9-2-6. RMG: Multiplication rate setting register (12 bits) .....	42
9-2-7. RDP: Ramp down point setting register (24 bits) .....	43
9-2-8. RUS: Acceleration S-curve setting register (12 bits) .....	44
9-2-9. RDS: Deceleration S-curve setting range register (12 bits) .....	44
9-3. Manual FH correction .....	45
9-4. Examples of acceleration/deceleration pattern setting .....	49
9-5. Changing the speed pattern during operation .....	50
<b>10. Other Functions .....</b>	<b>52</b>
10-1. Output pulse control .....	52
10-1-1. Output pulse modes .....	52
10-1-2. Output pulse length and operation complete timing .....	53
10-2. Operation status check output .....	53
10-2-1. #BSY/#END signal (signals output during or at the end of an operation) .....	53
10-3. Idling control .....	55

10-4. External mechanical inputs	56
10-4-1. PEL, MEL signals	56
10-4-2. SD signal	57
10-4-3. ORG, EZ signals	60
10-4-4. EA, EB signals	61
10-5. Servomotor I/F	62
10-5-1. INP signal	62
10-5-2. ERC signal (clear the deviation counter)	63
10-6. Stepper motor I/F	64
10-6-1. CDW signal (current reduction signal)	64
10-6-2. PH1 to PH4 signals (excitation sequence)	65
10-7. External start/simultaneous start	66
10-7-1. #STA signal in CPU connection mode	66
10-7-2. #STA signal and PS0 to PS4 signals in stand alone mode	68
10-8. External stop/simultaneous stop	69
10-8-1. STP signal	69
10-9. Counter	70
10-9-1. Counter types and input methods	70
10-9-2. Resetting the counter	70
11. Supplemental Description of the Stand Alone Operation	71
11-1. Requirements for selecting an EEPROM	71
11-2. Writing data to the EEPROM	71
11-2-1. Data writing procedures	71
11-2-2. EEPROM address map	72
12. Characteristics	73
12-1. Absolute maximum ratings	73
12-2. Recommended operating conditions	73
12-3. DC characteristics	73
12-4. AC characteristics	74
12-4-1. Reference clock	74
12-4-2. Serial bus I/F	75
12-5. Operation timing	76
13. External Dimensions	79
14. Handling Precautions	80
14-1. Design precautions	80
14-2. Precautions for transporting and storing LSIs	80
14-3. Precautions for installation	80

## 1. Functions and Features

### 1-1. Outline

This LSI is a pulse generator chip that connects to a CPU through a 4-line serial bus interface.

Since the PCD2112 does not need many signal lines to connect it to CPU data buses, address buses, and control buses, it works well with small CPU packages, and can effectively use any general-purpose port to connect to the CPU.

And, the device supports two types of control: 1) CPU connection, which allows the CPU to maintain control; and, 2) stand alone control in which the device operates independently using data in an external EEPROM.

### 1-2. Features

#### ◆ Serial bus I/F

The PCD2112 connects to a CPU using a simple 4-wire serial bus, reducing the number of terminals needed for CPU bus control and thus increasing the number of general I/O terminals available on the CPU. Also, the CPU can be connected to the PCD2112 the same way you would connect it to an EEPROM, using 4 wires.

#### ◆ Small package

Due to its use of a serial bus, the PCD2112 is small package (QFP-48) that still offers multiple functions. The external dimensions of the device are 9 mm (including pins) on each side and the molded package is 7 mm on a side.

#### ◆ Single voltage 3.3 V power supply

The PCD2112 needs only a single 3.3 V ( $\pm 10\%$ ) power supply.

The output signal level from this device is 3.3 V. It can accept input signals of 5 V (TTL) in addition to 3.3 V inputs.

#### ◆ High speed pulse train output

The PCD2112 outputs 2.4 Mpps with the reference clock of 9.8302 MHz (standard) and 5 Mpps with 20 MHz (maximum).

It can produce three types of output: common pulse output (pulse and direction signal), 2-pulse output (positive and negative pulses), and a 90° phase difference output with 4x multiplication.

#### ◆ Outputs excitation sequence signals for 2-phase stepper motors

The PCD2112 can output unipolar (2-2 phase and 1-2 phase) and bipolar (2-2 phase and 1-2 phase) excitation sequence signals.

#### ◆ Acceleration/deceleration control

The PCD2112 can perform linear acceleration/deceleration and S-curve acceleration/deceleration.

In the S-curve acceleration/deceleration mode, the middle of the speed pattern can contain a linear acceleration/deceleration part. (By setting the S-curve range)

The acceleration and deceleration characteristics can be set independently. (However, they cannot be set independently when the auto ramp down function is used.)

#### ◆ Speed override

The feed speed can be changed in mid operation.

#### ◆ Triangle profile suppression function (FH correction function)

If the amount of output pulses are small in the positioning mode, the PCD2112 automatically lower the maximum speed, FH, which suppresses the triangle motion profile.

#### ◆ Simultaneous start function

When using multiple PCD2112s, you can start them all at the same time by connecting a dedicated terminal (STA) on each one.

## ◆ A variety of operation modes

- Continuous operation (command control, manual pulsar control, manual switch control)
- Origin return operation (stop with a signal from a origin position sensor, stop when a feed amount limit is reached)
- Operation to move away from the origin position, operation to move away from the EL position.
- Positioning operation command control, manual pulsar control, manual switch control)
- Timer operation

## ◆ Stand-alone operation

By connecting a 4-wire EEPROM to the PCD2112, you can operate the unit without a CPU.  
You can store up to 32 types of operation data in the EEPROM.2. Specifications

## 2. Specifications

### 2-1. Basic specifications

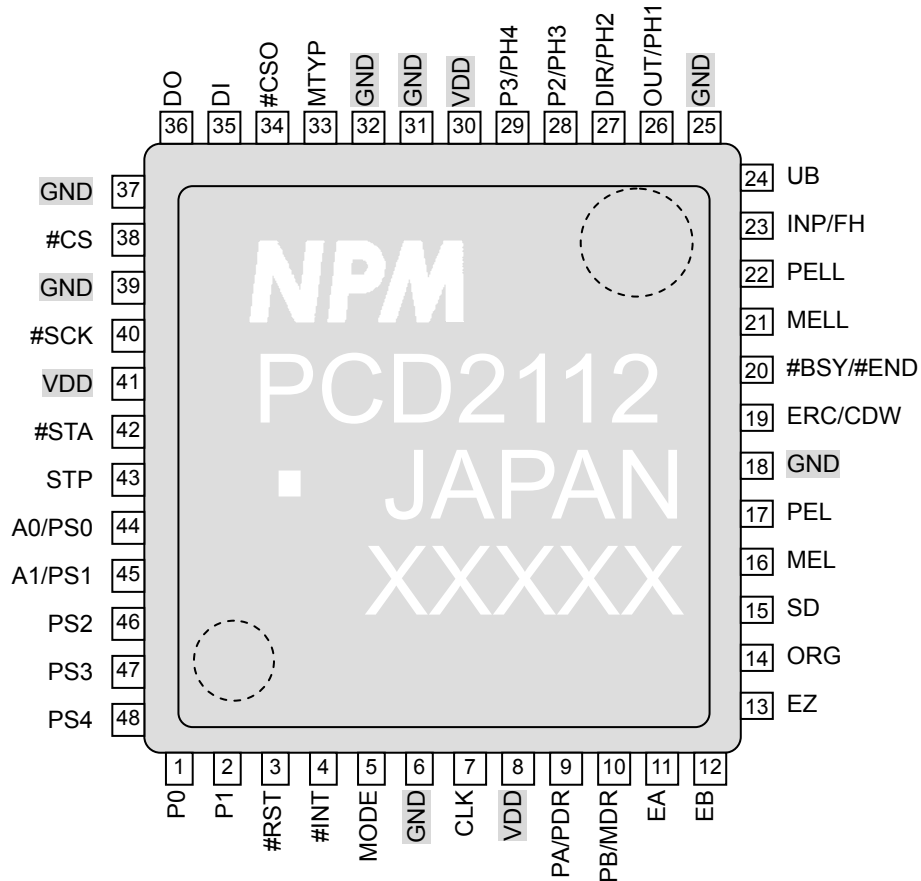
Item	Description
Power supply	+3.3 V $\pm$ 10%
Reference clock	9.8304 MHz standard (20 MHz max.)
CUP interface	4-wire system, serial bus
Range of positioning pulses (user set)	0 to 268,435,455 (28 bits)
Range of speeds (user set)	1 to 8,191 steps (13 bits)
Speed multiplication range (user set)	0.5x to 300x (Using a standard 9.830 MHz clock) When set to 0.5x: will deliver 0.5 to 4,095.5 PPS When set to 1x: will deliver 1 to 8,191 PPS When set to 300x: will deliver 300 to 2,457,300 PPS
Number of registers for setting the speed	Three (FL, FH1, FH2)
Ramp-down point range (user set)	0 to 16,777,215 (24 bits)
Acceleration/deceleration characteristics	Linear acceleration/deceleration, S-curve acceleration/deceleration
Acceleration rate range (user set)	1 to 65,535 (16 bits)
Deceleration rate range (user set)	1 to 65,535 (16 bits)
Up/down counter	-2,147,483,648 to +2,147,483,647 (32 bits)
General-purpose I/O terminal	4 bits (2 bits are fixed to output. 2 bit are selectable as input or output.)
Excitation sequence output	Excitation sequence output for stepper motors: Selectable 1-2 phase or 2-2 phase, Unipolar or Bipolar
Typical operations	<ul style="list-style-type: none"> <li>- Continuous operation at low/high speed</li> <li>- Origin return operation at low/high speed</li> <li>- Origin return operation at low/high speed with a feed amount limit.</li> <li>- Move away from the origin position at low/high speed.</li> <li>- Move away from the EL position at low /high speed.</li> <li>- Positioning operation at low /high speed.</li> <li>- Timer operation</li> <li>- Continuous operation and positioning operation using a <math>\pm</math>DR switch.</li> <li>- Continuous operation and positioning operation using a pulsar input.</li> <li>- Immediate stop/deceleration stop.</li> <li>- Speed changes in mid operation.</li> </ul>
Ambient operating temperature	-40 to +85°C
Storage temperature	-65 to +150°C
Package	48-pin QFP (molded package: 7 x 7 mm)

## Notes:

- 1) The target position cannot be changed in the middle of a positioning operation (position override is not available). However, feed speed can be changed during operation.
- 2) When using the auto ramp down point function, the acceleration and deceleration characteristics to must be identical.

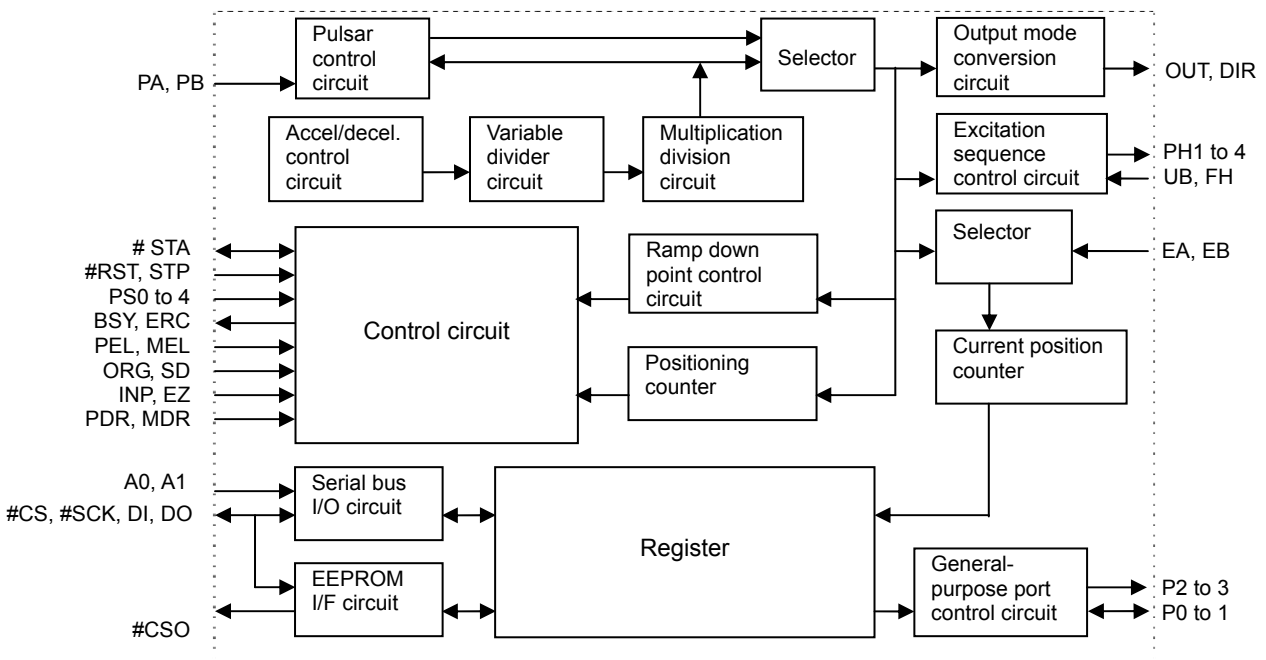


2-2. Pin assignment diagram



Note: Check the position of each pin number by referring to the markings.  
As shown in the figure above, pin 1 is in the lower left corner of the chip.

2-3. Circuit block diagram



## 2-4. List of terminal names, functions, and pin assignments

Pin	Terminal name	I/O	U/D	Processing	TTL	I <sub>OH</sub>	I <sub>OL</sub>	Description
1	P0	I/O	PD	(PD)	○	3mA	6mA	General-purpose I/O terminal 0 (Open when MODE = HIGH)
2	P1	I/O	PD	(PD)	○	3mA	6mA	General-purpose I/O terminal 1 (Open when MODE = HIGH)
3	#RST	I	PD	Must be set	○	-	-	Reset
4	#INT	O	OD	Open	○	-	6mA	Interrupt request
5	MODE	I	PD	Must be set	○	-	-	Operation mode (LOW: Connected to a CPU, HIGH: Stand alone operation)
6	GND	I						GND
7	CLK	I	-	Must be set	○	-	-	Reference clock
8	VDD	I						+3.3V
9	PA/PDR	I	PD	(PD)	○	-	-	Pulsar A/Positive direction switch
10	PB/MDR	I	PD	(PD)	○	-	-	Pulsar B/Negative direction switch
11	EA	I	PD	(PD)	○	-	-	Encoder A phase
12	EB	I	PD	(PD)	○	-	-	Encoder B phase
13	EZ	I	PD	(PD)	○	-	-	Encoder Z phase
14	ORG	I	PD	(PD)	○	-	-	Origin position sensor
15	SD	I	PD	(PD)	○	-	-	Ramp down sensor (used in both positive and negative directions)
16	MEL	I	PD	(PD)	○	-	-	Positive direction end limit
17	PEL	I	PD	(PD)	○	-	-	Negative direction end limit
18	GND	I						GND
19	ERC/CDW	O	-	Open	○	3mA	9mA	Clear deviation counter/current reduction
20	#BSY/#END	O	-	Open	○	3mA	9mA	Busy/Operation complete
21	MELL	I	PD	Must be set	○	-	-	MEL input logic (Held LOW when MEL is not used)
22	PELL	I	PD	Must be set	○	-	-	PEL input logic (Held LOW when MEL is not used)
23	INP/FH	I	PD	(PD)	○	-	-	In position/Full (L), half (H)
24	UB	I	PD	(PD)	○	-	-	Unipolar (L), bipolar
25	GND	I						GND
26	OUT/PH1	O	-	Must be set	○	3mA	9mA	Driving pulse/1-phase excitation signal
27	DIR/PH2	O	-	Must be set	○	3mA	9mA	Driving pulse/2-phase excitation signal
28	P2/PH3	O	PD	(PD)	○	3mA	9mA	General -purpose output terminal 2/3 phase excitation signal
29	P3/PH4	O	PD	(PD)	○	3mA	9mA	General-purpose output terminal 3/4 phase excitation signal
30	VDD	I						+3.3V
31	GND	I	-		x	-	-	Input terminal for shipping inspection (Must be connected to GND)
32	GND	I	-		x	-	-	Input terminal for shipping inspection (Must be connected to GND)
33	MTYP	I	PD	Must be set	○		-	LOW: Pulse train output, HIGH: Excitation sequence output
34	#CSO	O	-	Open	○	3mA	6mA	CS output for EEPROM
35	DI	I/O	PD	Must be set	○	3mA	6mA	Data input for serial communications (output when MODE = HIGH)
36	DO	I/O	PD	Must be set	○	3mA	6mA	Data output for serial communications (input when MODE = HIGH)
37	GND	I						GND
38	#CS	I/O	PD	Must be set	○	3mA	6mA	CS input for serial communications (output when MODE = HIGH)
39	GND	I						GND
40	#SCK	I/O	PD	Must be set	○	3mA	6mA	Synchronous clock for serial communications (output when MODE = HIGH)
41	VDD	I						+3.3V
42	#STA	I/O	PD	Open	○	3mA	6mA	External start
43	STP	I	PD	(PD)	○	-	-	External stop
44	A0/PS0	I	PD	(PD)	○	-	-	Device selection 0/pattern selection 0
45	A1/PS1	I	PD	(PD)	○	-	-	Device selection 1/pattern selection 1

Pin	Terminal name	I/O	U/D	Processing	TTL	I <sub>OH</sub>	I <sub>OL</sub>	Description
46	PS2	I/O	PD	(PD)	○	3mA	6mA	Pattern selection 2
47	PS3	I/O	PD	(PD)	○	3mA	6mA	Pattern selection 3
48	PS4	I/O	PD	(PD)	○	3mA	6mA	Pattern selection 4

U/D column (internal circuit): PD; Integrated pull down resistor (20K to 107K $\Omega$ ), OD; Open drain.

Processing column (when not in use): Open; Not connected, PD; Pull down, PU; Pull up,

(PD); Pulled down is preferable (5K to 10K $\Omega$ ), but can be not connected

TTL column (TTL-I/F): ○: 5V (TTL) I/F compatible (However, the output terminal will not reach 5 V when pulled up externally.)

## 2-5. System mode

Using the PCD2112, you can create any of the following 4 types of systems by setting 2 input terminals (MODE and MTYP).

However, do not change this setting while power is being supplied to the PCD2112.

Some terminals may be changed between input and output by changing the setting. Therefore, you must be careful when designing external circuits.

- 1) Pulse train output system when connected to a CPU (MODE = LOW, MTYP = LOW)  
Connect to a CPU using a 4-wire serial bus, and the PCD2112 outputs pulse trains.
- 2) Excitation sequence output system connect to a CPU (MODE = LOW, MTYP = HIGH)  
Connect it to a CPU using a 4-wire serial bus, and the PCD2112 will output excitation sequence signals. Any one of 4 types of excitation sequences can be selected by setting 2 input terminals (UP, FH).
- 3) PCD2112 stand alone operation pulse train output (MODE = HIGH, MTYP = LOW)  
The PCD2112 reads its operation data from an external EEPROM (with a maximum of 32 types of operation data) by selecting one of 32 data types using the PS0 to PS4 terminals. The pulse train for that data type is output.  
You only need a CPU to write data to the EEPROM. However, a CPU is not needed during operation.
- 4) Stand alone operation excitation sequence output (MODE = HIGH, MTYP = HIGH)  
The PCD2112 reads its operation data from an external EEPROM (with a maximum of 32 types of operation data) by selecting one of 32 data types using the PS0 to PS4 terminals. The excitation sequence signal for that data type is output.  
You may choose any one of four excitation sequences by setting 2 input terminals (UB, FH).

System mode				MODE	MTYP	UB	FH
Connected to a CPU	Pulse train output			L	L	L	(INP)
	Excitation sequence output	Unipolar	Full step (2-2 phase)	L	H	L	L
			Half step (1-2 phase)	L	H	L	H
		Bipolar	Full step (2-2 phase)	L	H	H	L
Half step (1-2 phase)			L	H	H	H	
Stand alone operation	Pulse train output			H	L	L	(INP)
	Excitation sequence output	Unipolar	Full step (2-2 phase)	H	H	L	L
			Half step (1-2 phase)	H	H	L	H
		Bipolar	Full step (2-2 phase)	H	H	H	L
Half step (1-2 phase)			H	H	H	H	

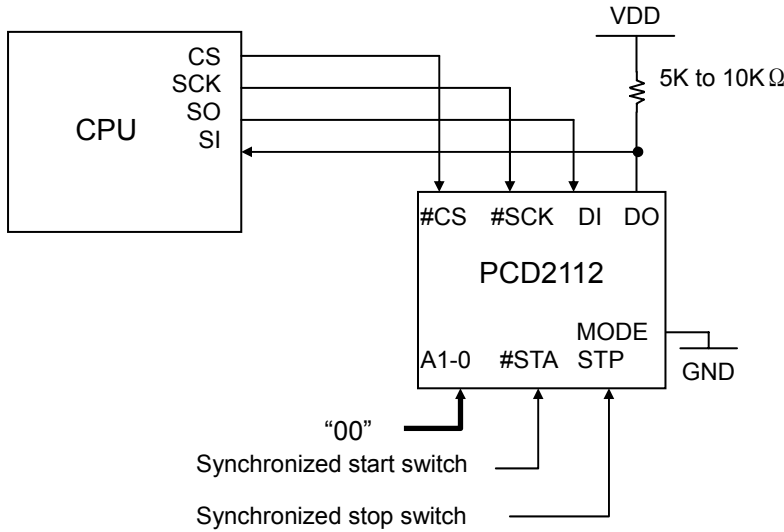
Note 1: In the pulse train output mode, the UB terminal settings will be ignored. However, since the PCD2112 contains internal pull down resistors, setting the terminals LOW can decrease the current consumption.

Note 2: In the pulse train output mode, the INP/FH terminal will function as an INP terminal.

2-5-1. CPU connection mode (MODE = LOW)

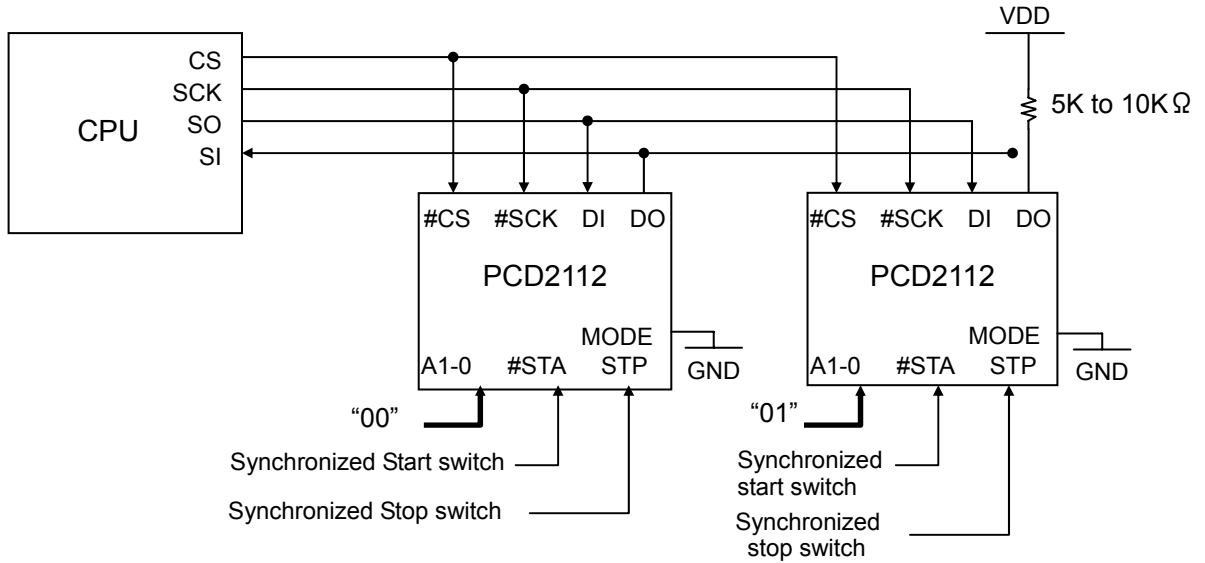
In this configuration, the PCD2112 is connected to a CPU by a 4-wire serial interface. Using a single CS output, the CPU can be connected to multiple PCD2112 chips. (If you want to control chips other than the PCD2112, you will have to set up separate CS output.)

1) Normal connections



You may also start and stop operations from the CPU. The SO output terminal must be pulled up externally as it will be high impedance when it is not outputting signals.

2) Connecting multiple PCD2112 chips



You may also start and stop operations from the CPU.

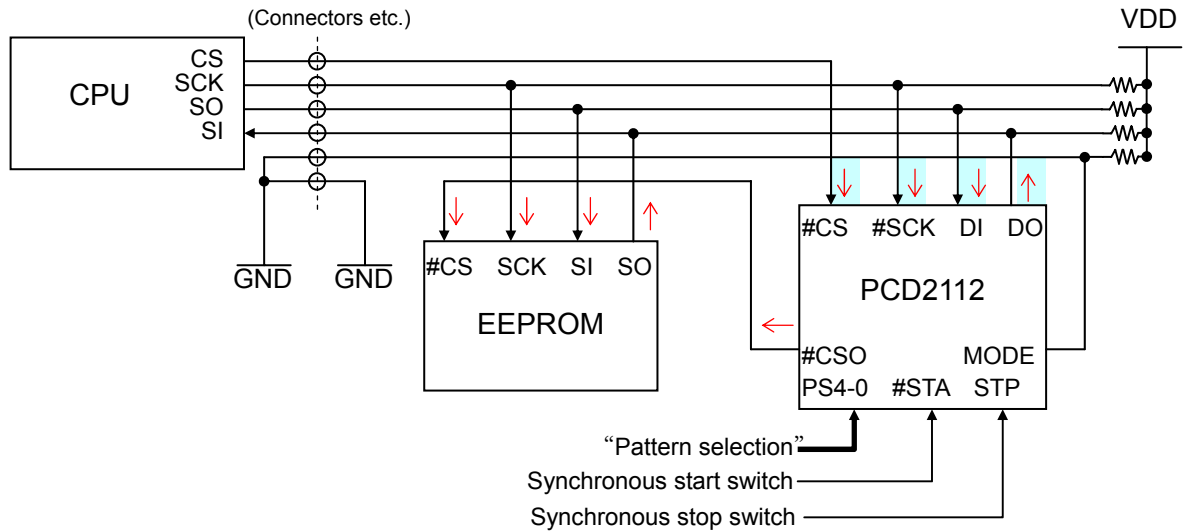
2-5-2. Stand-alone operation (MODE = HIGH)

The PCD2112 runs according to the operation pattern selected by hardware. A maximum of 32 data types are stored in a 4-wire serial EEPROM to choose from. To write data to the EEPROM, change the mode to a 4-wire CPU connection.

**When doing this, do not connect the CS from the CPU to the EEPROM (Instead, connect #CS0 on the PCD2112).**

1) When writing to the EEPROM

Use the CPU to write data into the EEPROM.



The CPU outputs the clock signal (SCK).

The CPU outputs the CS (chip select) signal for the PCD2112. The CPU also outputs CS for the EEPROM through the PCD2112.

The data terminals (SO, SI) are used between CPU and EEPROM as well as CPU and PCD2112 to communicate.

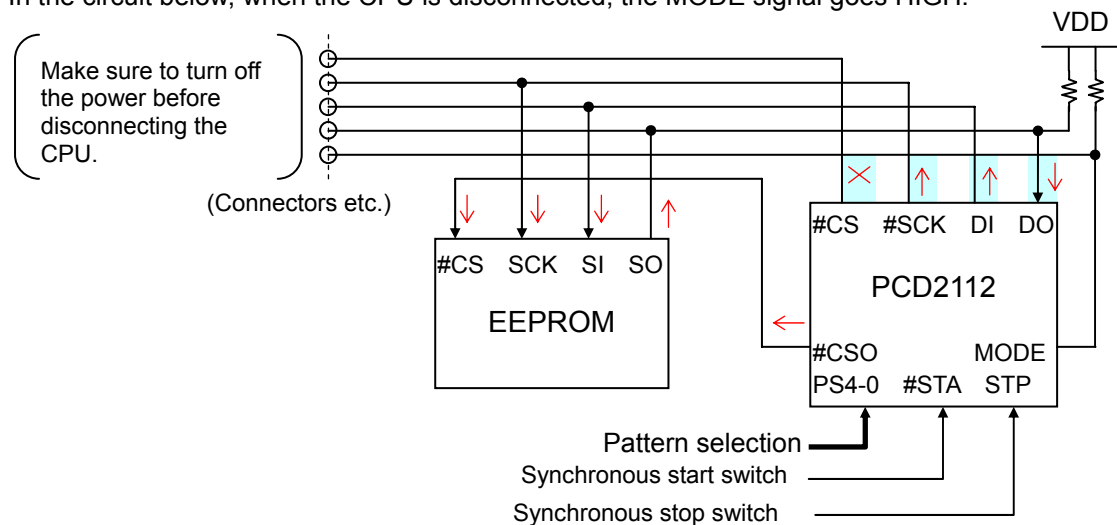
Note: The pattern selection PS0 and PS1 terminals become device selection terminals A0 and A1.

For communication with the CPU, set A0 and A1 identical to bits 4 and 5 for the bus control command.

2) Connections for stand alone operation

After writing the data to the EEPROM, disconnect the CPU before starting operation.

In the circuit below, when the CPU is disconnected, the MODE signal goes HIGH.



The PCD2112 will output the clock (SCK) and chip select (CS) signals.

Data (SO, SI) are communicated between the PCD and the EEPROM.

### 3. Terminal Descriptions

#### 3-1. Outline of the terminal functions

Terminal name	Terminal number(s)	I/O	Logic setting	Description
<b>Power supply</b>				
VDD	8, 30, 41	I		+3.3V
GND	6, 18, 25, 31, 32, 37, 39	I		GND
<b>System settings</b>				
MODE	5	I		Operation mode (LOW: Connect to a CPU; HIGH: Stand alone operation)
MTYP	33	I		LOW: Pulse train output, HIGH: Excitation sequence output
<b>Serial communications (MODE = LOW: When connected to a CPU)</b>				
#CS	38	I		CS input for serial communications
#SCK	40	I		Synchronous clock input for serial communications
DI	35	I		Data input for serial communications
DO	36	O		Data output for serial communications
<b>Serial communications (MODE = HIGH: When in stand alone operation)</b>				
#CS	38	O		CS output for serial communications
#SCK	40	O		Synchronous clock output for serial communications
DI	35	O		Data input for serial communications
DO	36	I		Data output for serial communications
<b>Operation control</b>				
#RST	3	I		Reset
CLK	7	I		Reference clock
A0, A1	44, 45	I		Device selection
#INT	4	O		Interrupt request
#BSY/#END	20	O		Operating/operation complete
ERC/CDW	19	O	RENV1.ERCL	Clear the deflection counter/reduce the current
INP	23	I	RENV1.INPL	In-position/full (LOW), half (HIGH)
#STA	42	I/O		External start
STP	43	I	RENV1.STPL	External stop
P0, P1	1, 2	I/O		General purpose I/O terminals 0 to 1
P2, P3	28, 29	O		General purpose output terminals 2 to 3
<b>Sensor inputs</b>				
PA, PB	9, 10	I	RENV2.PIM	Pulsar signal (1 pulse, 2 pulses, 90 phase difference signal)
PDR, MDR	9, 10	I	RENV1.DRL	Driving switch signal
EA, EB	11, 12	I	RENV2.EIM	Encoder signal (1 pulse, 2 pulses, 90 phase difference signal)
EZ	13	I	RENV2.EZL	Encoder Z phase
PELL, MELL	22, 21	I		EL input logic control
PEL, MEL	17, 16	I	PELL, MELL	End limit signal
ORG	14	I	RENV1.ORGL	Origin position sensor signal
SD	15	I	RENV1.SDL	Ramp down sensor signal
<b>To drive motors</b>				
OUT, DIR	26,27	O	RENV1.PMD	Drive pulse, driving direction
FH	23	I		In position /full (LOW), half (HIGH)
UB	24	I		Unipolar (L), bipolar/ERC
PH1, PH2, PH3, PH4	26, 27, 28, 29	O		Drive pulse/1-phase excitation signal
<b>Stand alone operation</b>				
#CSO	34	O		CS output for the EPROM
PS0, PS1, PS2, PS3, PS4	44, 45, 46, 47, 48	I		Pattern selection, 0 to 4

Note: The table above shows the signals classified according to their functions. Some terminals have multiple functions and therefore the same terminal numbers appear more than once.

### 3-2. Terminal functions in each system mode

The functions of the following terminal depend on the mode selected.

System mode Terminal name		Connected to a CPU				Stand alone operation			
		Pulse train output		Excitation sequence output		Pulse train output		Excitation sequence output	
		Terminal function	I/O	Terminal function	I/O	Terminal function	I/O	Terminal function	I/O
5	MODE	(LOW)	I	(LOW)	I	(HIGH)	I	(HIGH)	I
33	MTYP	(LOW)	I	(HIGH)	I	(LOW)	I	(HIGH)	I
40	SCK		I		I		O		O
35	DI		I		I		O		O
36	DO		O		O		I		I
38	#CS		I		I	(Same as #CSO)	O	(Same as #CSO)	O
44	A0/PS0	A0	I	A0	I	PS0	I	PS0	I
45	A1/PS1	A1	I	A1	I	PS1	I	PS1	I
46	PS2	Note 3	I	Note 3	I	PS2	I	PS2	I
47	PS3	Note 3	I	Note 3	I	PS3	I	PS3	I
48	PS4	Note 3	I	Note 3	I	PS4	I	PS4	I
1	P0	P0	I/O	P0	I/O	Note 3	O	Note 3	O
2	P1	P1	I/O	P1	I/O	Note 3	O	Note 3	O
24	UB	(LOW)	I	UB	I	(LOW)	I	UB	I
23	INP/FH	INP	I	FH	I	INP	I	FH	I
26	OUT/PH1	OUT	O	PH1	O	OUT	O	PH1	O
27	DIR/PH2	DIR	O	PH2	O	DIR	O	PH2	O
28	P2/PH3	P2	O	PH3	O	Note 3	O	PH3	O
29	P3/PH4	P3	O	PH4	O	Note 3	O	PH4	O

Note 1: Empty cells simply mean that the function of these terminals is the same as described in the terminal name chart. Multiple names in the terminal name chart means that the terminal function can be selected by the software.

Note 2: LOW and HIGH refer to level applied to the input terminals.

Note 3. Pull these terminals down (5 k to 10 k-ohm resistor).

3-3. Terminal functions

3-3-1. CLK [7]

This is an input terminal for the reference clock. (The standard clock frequency is 9.8304 MHz)  
 The precision of the reference clock frequency affects the precision of the output pulse speed.  
 And, it also affects the input sensitivity of the start timing, #STA, #ORG, #EG, and STP signals, as well as the read and write timings.

3-3-2. #RST [3]

This is an input terminal for a reset signal.  
 Pull this terminal LOW and supply a CLK signal for 8 or more clock cycles. The PCD will be reset.  
 Make sure to reset the PCD at least once after turning the power on and before starting to use the unit to use.

After the PCD is reset its condition is as shown below.

Item	Reset status (initial status)
Internal register	0
Control command	0
P0 to P3 terminals	Inputs
OUT terminal, DIR terminal	HIGH
BSY/ERC terminals	HIGH
PH1, PH2, PH3, PH4 terminals	STEP sequence status. See Note

Note: See section 11-7-2, PH1 to PH4 signals.

3-3-3. MODE [5]

Input terminal used to select the PCD2112 system mode.  
 For details about the system mode, see section 2-5, System mode.

MODE	System mode
LOW	Connected to a CPU
HIGH	Stand alone operation connected to an EEPROM

3-3-4. MTYP [33]

Input terminal used to select the method for controlling a motor.

Setting Terminal name	Mode	Remarks
LOW	Pulse train output mode	The INP/FH terminals function as INP inputs. The OUT/PH1 terminals output pulse trains. The DIR/PH2 terminals output direction signals. The P2/PH3 terminals function as the P2 general-purpose output terminal. The P3/PH4 terminals function as the P3 general-purpose output terminal.
HIGH	Excitation sequence output mode	The INP/FH terminals function as the FH input. The OUT/PH1 to P3/PH4 terminals output excitation sequence signals.



## 3-3-5. #INT [4]

This is an output terminal for an interrupt request signal sent to the CPU. It goes LOW when an interrupt condition occurs.

It is an open-drain output so it can be wired-OR connection. However, an external pull up resistor (5 K to 10 K ohms) is always needed.

Use bits 22 to 16 of the RENV2 register to set the interrupt conditions. The #INT signal can also be output when an abnormal stop is caused by the  $\pm$ #EL signal, STP signal, etc., or when deceleration stop and immediate stop commands are received, or when a deceleration start or start from an external signal is performed.

## 3-3-6. #SCK [40], DI [35], DO [36], #CS [38], #CSO [34]

These are the serial communication terminals.

Setting Terminal name	MODE = LOW (Connected to a CPU)	MODE = HIGH (Stand alone operation)
#SCK	Input for a synchronous clock serial signal.	Outputs a synchronous clock serial signal.
DI	Serial input signal	Serial output signal
DO	Serial output signal	Serial input signal
#CS	Chip select input	Outputs a signal the same as CS0. See the Note
#CSO	Chip select output to the EEPROM. (When you need to control both the PCD and EEPROM from the CPU through a single CS signal line.) See the Note	

Note: May be left open if not used.

## 3-3-7. PA/PDR [9], PB/MDR [10]

These terminals function as follows, according to the MOD bit settings in the RMD register.

Setting MOD to any value outside the range of 7 to 10 will be invalid. (Even when the setting is invalid, the condition can be checked using RSTS.)

Setting Terminal name	RMD register: MOD = 9, 10	RMD register: MOD = 7, 8
PA/PDR	PA (pulsar input)	PDR (positive external switch)
PB/MDR	PB (pulsar output)	MDR (negative external switch)

PA, PB: Input signals from a pulsar. The motor can be operated synchronously with this pulsar. The input setting for the PA/PB signals can be selected: 2 pulses, 90° phase difference (1x, 2x, or 4x magnification) controlled by the RENV2.PIM setting.

PDR, MDR: Input for an external switch signal. It can be set to keep operating while the switch is pressed or to inch along with each press.

The input logic of this terminal can be selected by setting DRL in the RENV1 register.

A noise filter can be applied to the PA/PDR, PB/MDR inputs by setting RENV2.PINF.

The signal levels of the PA/PDR, PB/MDR terminals can be checked in the status registers(RSTS.SPA, RSTS.SPB).

3-3-8. A0/PS0 [44], A1/PS1 [45], PS2 [46], PS3 [47], PS4 [48]

In stand-alone operation (MODE = HIGH), these terminals function as PS0 to PS4 input terminals for selecting operating patterns.

When connected to a CPU (MODE = LOW), these terminals function as A0 and A1 inputs for selecting device addresses.

In both cases, input logic is positive. (0: LOW and 1: HIGH)

Terminal name \ Setting	MODE = LOW (Connected to a CPU)	MODE = HIGH (stand alone operation)
A0/PS0	A0: Device address bit 0	PS0: Operation pattern setting 0
A1/PS1	A1: Device address bit 1	PS1: Operation pattern setting 1
PS2	Note 1	PS2: Operation pattern setting 2
PS3	Note 1	PS3: Operation pattern setting 3
PS4	Note 1	PS4: Operation pattern setting 4

The devices that match the device selection settings and bits 4 and 5 of the bus control command are selected for connection. Therefore, a maximum of 4 PCD2112s can be connected to one serial bus.

Note 1: Do not change input levels on these terminals. Changing the input level while in operation will stop the operation immediately, and the current position counter (RCUN) will be reset.

If the input status of PS0 to PS4 changes during stand alone operation, the PCD will immediately stop the operation.

3-3-9. EA [11], EB [12], EZ [13]

EA and EB are encoder signal input terminals.

You can select the input specifications for these terminals as 2-pulse or 90 phase difference (1x, 2x, or 4x magnifications)) by RENV2.EIN setting.

EN is the input terminal for encoder Z phase signals. It may be used in origin return operations.

You can check the level on each terminal using the status registers (RSTS.SEA, RSTS.SEB, and RSTS.SEZ).

3-3-10. #BSY/#END [20]

This terminal outputs signals to indicate that the PCD2112 is busy or has completed its operation. Its function is as follows, according to the RENV1.ENDM setting.

Terminal name \ Setting	RENV1.ENDM = 0	RENV1.ENDM = 1
#BSY/#END	BSY (operating)	END (operation complete)

#BSY: Outputs a LOW while in operation and returns HIGH after the operation is complete.

#END: When the operation is ended by #STA going LOW, this terminal goes LOW. When #STA goes HIGH, this terminal returns to HIGH. For details about the differences between the BSY and END signals, see section 10-2-1, #BSY/#END signals.

3-3-11. SD [15]

This is the ramp down switch signal input terminal.

By setting RMD.MSDE, you can enable/disable the SD signal. The input logic of this signal can be set in RENV1.SDL.

By setting RENV1.SDM, you can set the PCD2112 either to decelerate, or to decelerate and stop when this signal turns on.

When decelerate is selected, the motor will start decelerating when the SD signal turns on and will accelerate again when the SD signal is turned off.

When the decelerate and stop is selected, the motor will accelerate again when SD turns off while decelerating. However, if SD turns off after the motor has stopped, the motor will stay stopped.

By setting RENV1.SDLT, SD can be latched so that the motor can be decelerated and stopped by receiving a one-shot pulse signal.

When starting operation, the PCD latches the input status of this terminal. Then, the PCD responds to a change in this signal from off to on.

You can monitor the input status of the SD signal in the RSTS.SSD register. You can also monitor the SD status after it is latched in the RSTS.SDIN register.

## 3-3-12. PELL [22], MELL [21]

These terminals set the input logic for the PEL and MEL signals.

Set the PEL logic using the PELL terminal, and the MEL logic using the MELL terminal.

ELL terminal	Operation
LOW	PEL, MEL use positive logic.
HIGH	PEL, MEL use negative logic.

## 3-3-13. PEL [17] (+EL), MEL [16] (-EL)

Input terminals for the end limit (EL) signals. You can set the input logic for these signals using the PELL and MELL terminals.

When an EL signal corresponding to the moving direction turns on, the motor will stop immediately. Then, the motor will stay stopped, even after the EL signal turns off.

If an EL signal corresponding to the moving direction is already on when the motor starts an operation, the PCD2112 will not start the motor.

The EL signal will be ignored while in timer mode.

You can check the status of these signals in the RSTS.SPEL and SRTS.SMEL registers.

## 3-3-14. ORG [14]

This is an input terminal for the origin position switch signal. You can set the input logic of this terminal in the RENV1.ORGL register.

When this signal changes from off to on while in the origin return mode (MOD = 1, 2), the motor will stop immediately. Then, the motor will remain stopped, even if the ORG signal turns off again.

Please note that if the motor is started while the ORG input is on, the motor cannot be stopped.

If this signal turns off while moving away from the origin position (MOD = 3), the motor will stop immediately and will stay stopped, even if the ORG signal turns on again. If the ORG signal is already off when starting an operation, the motor will stay stopped.

You can check the status of this signal in the RSTD.SORG register.

## 3-3-15. #STA [42]

This is an input terminal for an external start signal. An external pull up resistor is needed to use the external start signal.

By setting RMD.MSY to 1, after a start command has been entered, the motor will start on the falling edge of the #STA signal. Make sure the external circuit is arranged as an open collector (or open drain). The PCD2112 does not respond to signals shorter than 5 cycles of the reference clock frequency.

This terminal can output a one shot pulse signal when an STAO command is written.

You can check the #STA terminal signal level in the RSTS.SSTA register.

## 3-3-16. STP [43]

This is an input terminal for an external stop signal. You can set the input logic of this signal using the RENV1.STPL register.

Enable/disable this signal by setting RMD.MSPE. Set the operation (immediate stop or decelerate and stop) by setting RENV1.STPM. If this signal turns off after the motor has stopped, the PCD2112 will keep the motor stopped.

If the STP signal is already on when an operation is started, the motor will not start.

You can check the status of the STP input in RSTS.SSTP.

3-3-17. OUT/PH1 [26], DIR/PH2 [27]

Set the MTYP terminal to control the function of these terminals

Setting Terminal name	MTYP = LOW (Pulse train output mode)	MTYP = HIGH (Excitation sequence output mode)
OUT/PH1	OUT output	PH1 output
DIR/PH2	DIR output	PH2 output

OUT, DIR: Pulse output terminal for driving the motor.

You can choose one of eight output specifications by setting RENV1.PMD.

While in the timer mode, the pulse output will be masked. (A direction signal is output.)

PH1, PH2: Output phase signals for a stepper motor.

Together with the P2/PH3 and P3/PH4 terminals, these terminals output phase distribution signals for a stepper motor. The output of these terminals depends on the setting for the UB and FH terminals.

The output status of PH1 to PH3 can be checked in RSTS.SPH.

3-3-18. ERC/CDW [19]

The output of these terminals is as shown below by setting RENV1.CDWS.

Setting Terminal name	CDWS = 0	CDWS = 1
ERC/CDW	ERC output	CDW output

ERC output: Outputs a deflection counter clear signal for a servomotor system.

CDW output: Outputs current reduction signal for a stepper motor.

You can set the output logic of these signals by setting RENV1.ERCL. You can check the status of these signals in RSTS.SERC.

3-3-19. UB [24]

This is an input terminal for setting the excitation system. You can check the input level in RSTS.SPUB.

UB terminal	Operation
LOW	Output the unipolar excitation sequence for PH1 to PH4.
HIGH	Output the bipolar excitation sequence for PH1 to PH4

3-3-20. INP/FH [23]

These terminals function as shown below, according to the status of MTYP input terminal

Setting Terminal name	MTYP = LOW (Pulse train output mode)	MTYP = HIGH (Excitation sequence output mode)
FH/INP	INP input	FH input

INP input: Input terminal for the in-position signal (positioning complete signal).

By supplying a signal from the servomotor driver, the PCD2112 can be set to complete its operation by stopping the motor after the pulse output is complete.

You can check input logic of these signals using RENV1.INPL, and the input status in RSTS.SINP.

FH input: Terminal used to set the excitation sequence.

Select between 2-2 phase excitation and 1-2 phase excitation for a 2-phase stepper motor. When this is LOW, 2-2 phase is selected. When it is HIGH, 1-2 phase is selected.

## 3-3-21. P0 [1], P1 [2]

These are general-purpose input/output terminals that can be used when the PCD2112 is connected to a CPU.

You can set each bit either for input or output by RIOP.PMD setting.

## 3-3-22. P2/PH3 [28], P3/PH4 [29]

These terminals function as shown below, according to the setting of MTYP.

Setting Terminal name	MTYP = LOW (Pulse train output mode)	MTYP = HIGH (Excitation sequence output mode)
P2/PH3	General-purpose output terminal P2	Phase signal output PH3
P3/PH4	General-purpose output terminal P3	Phase signal output PH4

General-purpose output terminals (P2, P3): These are general-purpose output terminals.

Phase signal output PHn:

Output terminal for the stepper motor excitation signal.

The excitation sequence signal can be switched by synchronizing this signal with the output pulses.

The F/H terminal settings let you select either 1-2 phase or 2-2 phase excitation sequences. By setting the U/B terminals, you can select a unipolar or bipolar excitation sequence.

The excitation sequence cannot be switched when the timer mode is selected.

## 3-3-23. VDD [8, 30, 41], GND [6, 18, 25, 31, 32, 37, 39]

These are input terminals for the power supply.

Supply +3.3VDC  $\pm$ 10% to VDD. Make sure to connect all of these terminals.

## 4. Memory Map

The control registers are arranged in the memory map so that they can be accessed from a CPU, the same way memory is accessed.

The registers that are used the most often are next to the command register (RCOM), so that the write time can be reduced by using a block writing process. (You can also process all of the registers as a single block.)

Address	Register name	Attribute	Bit length	Remark
04h to 07h	RENV2	R/W	32	Environment setting 2
08h to 0Bh	RENV1	R/W	30	Environment setting 1
0Ch to 0Dh	RDS	R/W	12	S-curve range for deceleration
0Eh to 0Fh	RUS	R/W	12	S-curve range for acceleration
10h to 11h	RDR	R/W	16	Deceleration rate
12h to 13h	RUR	R/W	16	Acceleration rate
14h to 15h	RMG	R/W	12	Speed magnification rate
16h to 18h	RDP	R/W	24	Ramp down point
19h to 1Ah	RFL	R/W	13	FL speed (initial speed) step value
1Bh to 1Ch	RFH1	R/W	13	FH1 speed (operating speed) step value.
1Dh to 1Eh	RFH2	R/W	13	FH2 speed (operating speed) step value
1Fh to 22h	RMV	R/W	28	Number of pulses to output
23h to 24h	RMD	R/W	15	Operation mode
25h	RCOM	R/W	8	Command
26h to 29h	RSTS	R	32	Status
2Ah to 2Bh	RIST	R/W	16	Interrupt factor register
2Ch to 2Dh	RIOP	R/W	16	General-purpose I/O terminal control
2Eh to 31h	RCUN	R/W	32	Current position counter
32h to 35h	RDWC	R	28	Number of positioning pulses remaining
36h to 37h	RSPD	R	13	Current speed monitor

When writing multiple bytes of data using serial communication, the PCD2112 writes the communication data one byte at a time, so that it cannot overwrite all the bytes in one register at one time.

The RFH1 and RFH2 registers can be used to set the FH speed. These are used to change the speed during operation.

To change the speed during operation while using FH1 speed, first put a new speed in the FH2 register. Then, switch to FH2 speed using a speed change command.

## 4-1. Detailed memory map

Address	Name (8 bits)	
00	Reserved	Reserved (access prohibited)
01	Reserved	
02	Reserved	
03	Reserved	
04	RENV2 (31 to 24)	Environment setting register 2
05	RENV2 (23 to 16)	
06	RENV2 (15 to 08)	
07	RENV2 (07 to 00)	
08	RENV1 (31 to 24)	Environment setting register 1
09	RENV1 (23 to 16)	
0A	RENV1 (15 to 08)	
0B	RENV1 (07 to 00)	
0C	RDS (15 to 08)	Specify S-curve range for deceleration
0D	RDS (07 to 00)	
0E	RUS (15 to 08)	Specify S-curve range for acceleration
0F	RUS (07 to 00)	
10	RDR (15 to 08)	Deceleration rate
11	RDR (07 to 00)	
12	RUR (15 to 08)	Acceleration rate
13	RUR (07 to 00)	
14	RMG (15 to 08)	Speed multiplication rate
15	RMG (07 to 00)	
16	RDP (23 to 16)	Ramp down point
17	RDP (15 to 08)	
18	RDP (07 to 00)	
19	RFL (15 to 08)	Initial speed
1A	RFL (07 to 00)	
1B	RFH1 (15 to 08)	Operating speed 1
1C	RFH1 (07 to 00)	
1D	RFH2 (15 to 08)	Operating speed 2
1E	RFH2 (07 to 00)	
1F	RMV (31 to 24)	Feed amount
20	RMV (23 to 16)	
21	RMV (15 to 08)	
22	RMV (07 to 00)	
23	RMD (15 to 08)	Operation mode
24	RMD (07 to 00)	
25	RCOM (07 to 00)	Command
26	RSTS (31 to 24)	Status
27	RSTS (23 to 16)	
28	RSTS (15 to 08)	
29	RSTS (07 to 00)	
2A	RIST (15 to 08)	Interrupt cause
2B	RIST (07 to 00)	
2C	RIOP (15 to 08)	General-purpose I/O terminal control
2D	RIOP (07 to 00)	
2E	RCUN (31 to 24)	Current position counter
2F	RCUN (23 to 16)	
30	RCUN (15 to 08)	
31	RCUN (07 to 00)	
32	RDWC (31 to 24)	Used for work pieces in positioning mode
33	RDWC (23 to 16)	
34	RDWC (15 to 08)	
35	RDWC (07 to 00)	
36	RSDP (15 to 08)	Current speed
37	RSDP (07 to 00)	
38		Not used
39		
3A		
3B		
3C		
3D		
3E		
3F		

## 5. Serial Communication Format

The PCD2112 functions exclusively as a slave in 4-wire serial communication. The CPU is the master. Communication will start when the CPU changes the CS terminal from HIGH to LOW. When the CS terminal turns HIGH, the communication is ended. There are 7 communication formats.

- 1) Reading register values from the PCD2112  
(bus control command: 1 byte) + (start address: 1 byte) + (several bytes data read)
- 2) Writing data to a register in the PCD2112  
(bus control command: 1 byte) + (start address: 1 byte) + (several bytes data written)
- 3) Reading the bus status  
(bus control command: 1 byte) + (bus status value: 1 byte)
- 4) Access control to an EEPROM in stand alone operation  
(bus control command: 1 byte)
- 5) Reading data from an EEPROM in stand alone operation  
(EEPROM command: 1 byte) + (start address: 2 bytes) + (several bytes of data read)
- 6) Writing data to an EEPROM in stand alone operation  
(EEPROM command: 1 byte) + (start address: 2 bytes) + (several bytes of data written)
- 7) Reading the EEPROM status in stand alone operation  
(EEPROM command: 1 byte) + (1 byte of status read value)

### 5-1. Bus control command

Since a 4-wire serial bus is used to control the PCD2112, access to the PCD2112 is through half-duplex communication. The PCD2112 has to classify the communication passing over this serial bus as “read,” “write,” or “other.” Bus commands are used to allow the PCD2112 to perform this classification.

Command	Code	Description
READ	00nn 0011	Read data from the PCD2112
WRITE	00nn 0010	Write data to the PCD2112
RDSR	00nn 0101	Read the bus status
WREN	00nn 0110	Shift to the EEPROM access mode
WRDI	00nn 0100	End the EEPROM access mode

The PCD2112 terminals to A0 and A1 are used select the device number. One CS signal can be connected to up to four PCD2112s.

Set the device number of the communication target in bits 4 and 5 (nn), in the code table above. (When A1 = LOW, Bit 5 = 0)

If you want to control multiple PCDs with one CS signal, please read section 5-9, *Identifying a PCD when multiple PCDs are connected*.

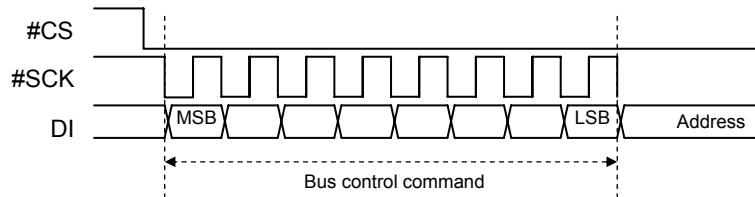


## 5-2. Basic operation

The basic serial bus operation will start on the falling edge of the #CS signal, as shown below.

After the #CS signal drops, the PCD2112 reads the DI input while synchronizing itself with the rising edge of the #SCK signal.

A bus control command is assigned to the first 8 bits.

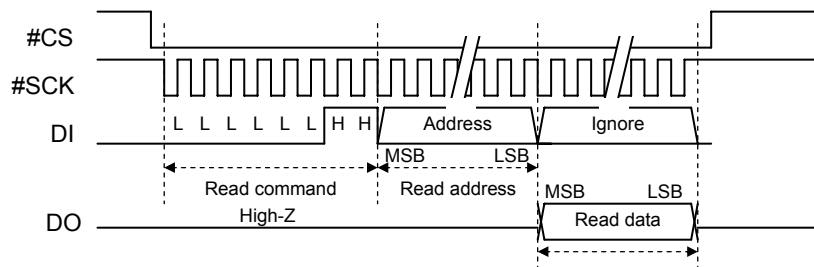


## 5-3. Reading data from PCD2112

When the CPU sends a bus control command (READ: 03h) and a read address (8 bits) to the PCD2112, the PCD2112 responds by outputting the data from the specified start address, synchronized with the next falling edge of the #SCK signal.

Data is output in units of 8 bits.

The #SCK signal is pulsed while #CS is low and the PCD2112 reads data continuously. The PCD2112 controls the read address using a 6-bit read address counter, so that it can output the same data in a 64 byte cycle. When the PCD attempts to read an undefined address (00h to 03h, 38h to 3Fh), the data read will always be 00h.

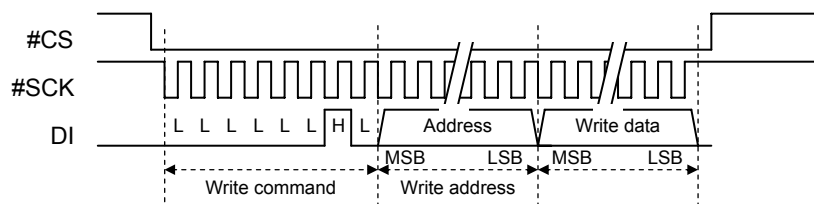


## 5-4. Writing data to PCD2112

When the CPU sends a bus control command (WRITE: 02h), a write address (8 bits), and write data to the PCD2112 in that order, the PCD2112 will capture these data on the rising edge of #SCK.

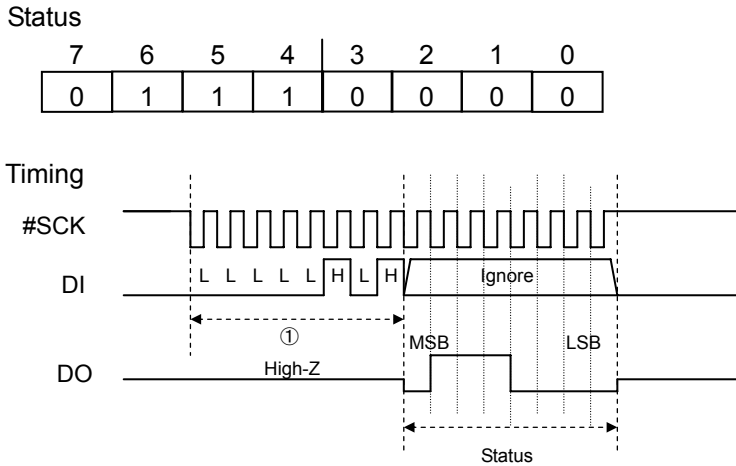
Keep supplying the #SCK signal and hold #CS LOW. The CPU will write data to the PCD2112 continuously. In this case, the CPU continues to write data by incrementing the address by 1 each time.

The PCD2112 controls the address internally using an 8-bit write address counter, so that the write address will be the same for a 256 byte cycle. When the PCD attempts to write to an undefined address (00h to 03h, 38h to 3Fh), the PCD2112 will ignore the data.



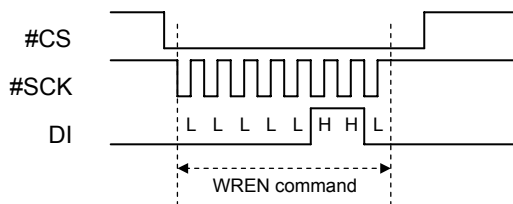
5-5. Reading the bus status

This operation is used to check access to an EEPROM in stand alone operation. After writing a WREN command (06h), the CPU can communicate with the EEPROM. After writing a WRDI command (04h), the CPU returns to communicating with the PCD2112. The CPU reads the bus status in order to find out which device is the current communication target. If bits 4 to 6 of the bus status byte are “111,” the PCD2112 is the communication target. However, some types of EEPROMs have “XXX” in bits 4 to 6 of the EEPROM bus status byte. In this case, the CPU cannot check the bus status. When the PCD2112 is the communication target and the CPU sends a bus control command (RDSR: 05h), the CPU can read the following status by synchronizing with the next SCK signal.



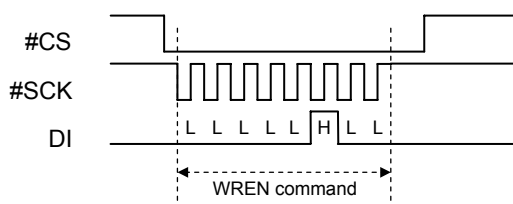
5-6. Moving to the EEPROM access mode

This operation is used to confirm the access to the EEPROM in stand alone operation. In stand alone operation, #CSO output from the PCD2112 is connected to CS on the EEPROM. Soon after power is applied, #CSO output on the PCD2112 is HIGH (CS = off). In other words, the CPU cannot access to the EEPROM. In this condition, if the CPU sends a WREN command (06h) to the PCD2112, the PCD2112 will pass along the #CS it receives to the EEPROM, through the #CS output (EEPROM access mode). While accessing the EEPROM, the serial communication target of the CPU is the EEPROM, and the PCD2112 will not accept any command except a WRDI command.



5-7. Ending the EEPROM access mode

This operation is used to end the access to the EEPROM in stand alone operation. While the PCD2112 is accessing the EEPROM, the PCD2112 will only respond to a WRDI command (04h) from the CPU. Soon after receiving this command, the PCD will end the EEPROM access mode and #CSO output from the PCD2112 will be pulled HIGH. Then, the PCD communication returns to communication with the CPU.



04h is the WRDI command. In the EEPROM specification, this is defined as a write disable command.

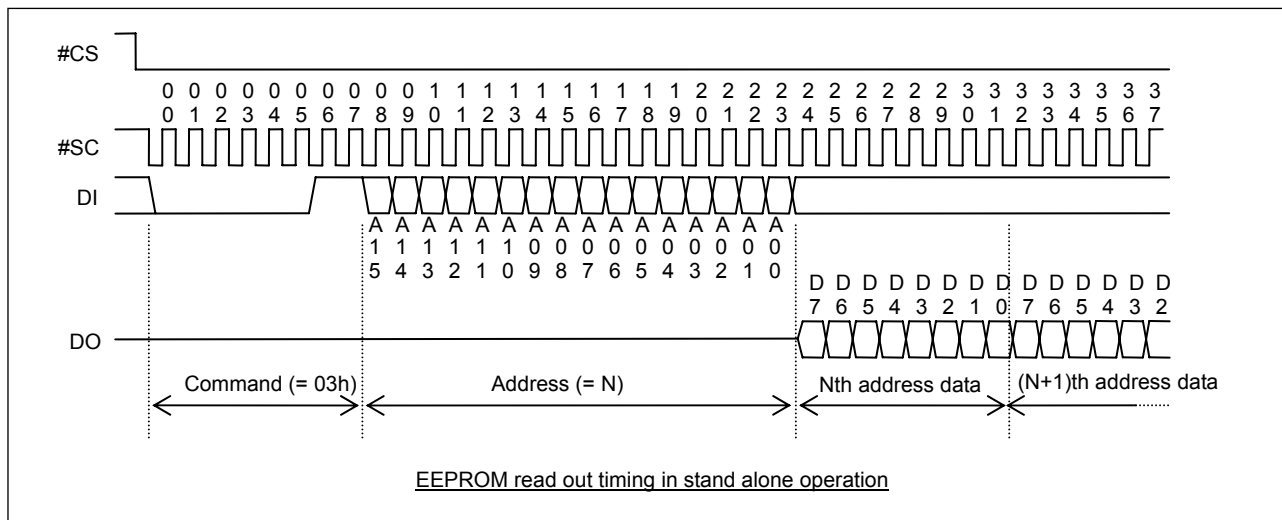
Therefore, when this WRDI command is issued, the PCD2112 ends the EEPROM access mode and writing to the EEPROM will be prohibited.

### 5-8. EEPROM control

The CPU only has an access to the EEPROM in order to change contents of the EEPROM. This is done by connecting the CPU temporarily, while the PCD2112 is in stand alone operation. For details, see section 2-5-2, *Stand alone operation*.

The EEPROMs used in stand alone operation must meet the following conditions.

- 1) 16K bit (2 K-bytes) or more with an 8-bit architecture (Ex.: 2Kx8 bits)
- 2) It's must be accessible using 16-bit addresses
- 3) It must respond to an 03h Read command.
- 4) The access speed must be at least 1/2 of the reference clock (CLK), or faster.



Note: The DI terminal becomes an output terminal and the DO terminal becomes input terminal in stand alone operation mode.

After the PCD2112 changes to the EEPROM access mode, refer to the instruction manual for the EEPROM in order to write data to the EEPROM.

Note: To enable writing to the EEPROM, put the PCD2112 in the EEPROM access mode. Then, issue a *write enable* command. (Send "000001100" two times.)

#### Write data from the CPU to the EEPROM

- 1) Send a WREN (06h) bus control command to the EEPROM to put the EEPROM in access mode.
- 2) Send a WREN (06h) EEPROM control command to the EEPROM to prepare it for writing.
- 3) Send data to the EEPROM in this order: control command WRITE (02h), 16 bit address, and data to write. The number of bytes that can be written at one time depends on the EEPROM used. Therefore you may need to divide the data into several groups in order to write it to the EEPROM.

For details about the EEPROM internal address map, see section 11, *Supplemental Explanation of Stand Alone Operation*.

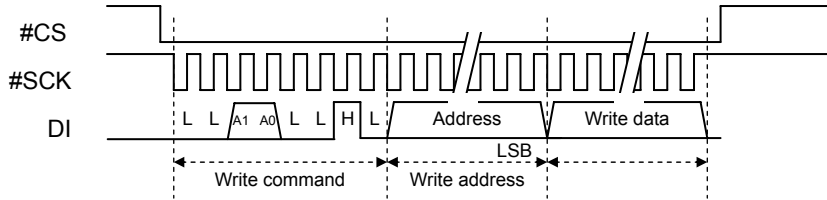
- 4) Send a WRDI (04h) bus control command.

The EEPROM and the PCD2112 will both receive this communication. And, when the EEPROM enter the write prohibited condition, the EEPROM access mode will end.

5-9. How to identify a PCD when multiple PCDs are connected

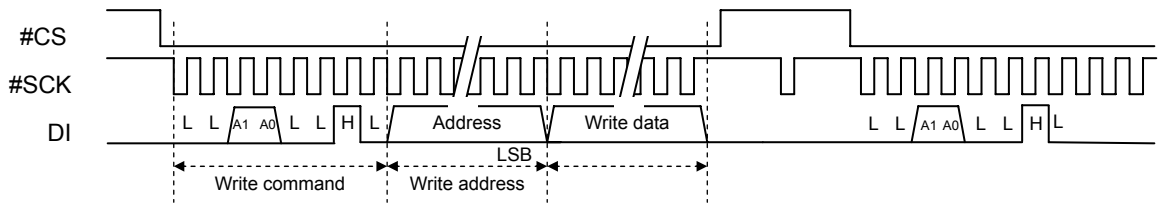
The PCD2112 has 2 device number terminals (A0, A1). These are only effective in the CPU mode. Therefore, up to four PCD2112s can be connected to the same serial bus (using a common CS). Use the 4th and 5th bits of a bus command to specify the device number.

[An example of a bus control command with multiple PCDs]



Note: To switch between PCD2112s, you have to supply #SCK for at least one clock cycle, while #CS = HIGH.

The CPU must change the output logic of the #SCK signal temporarily.



## 6. Command

Values written into the RCOM register (address: 25h) will become commands.

### Command list

Hex	Name	Description
00h	NOP	Invalid
01h	SRST	Software reset
05h	SRDS	Disable SRST command (initial status)
0Ah	SREN	Enable SRST command <span style="float: right;">Note 3</span>
10h	ERCO	ERC signal output
11h	ERCR	Stop ERC signal output
12h	STAO	STA output
13h	CUNR	Reset current position counter
14h	TRST	Reset direction change timer
30h	EMGSP	Emergency stop
31h	STOP	Immediate stop
32h	SDSTP	Deceleration stop
40h	STAF <sub>L</sub>	FL constant speed start
41h	STAF <sub>H1</sub>	FH1 constant speed start
42h	STAF <sub>H2</sub>	FH2 constant speed start
43h	STAUD <sub>1</sub>	Start with acceleration (FL→FH1)
44h	STAUD <sub>2</sub>	Start with acceleration (FL→FH2)
50h	CNTFL	FL constant speed remaining pulses start
51h	CNTFH <sub>1</sub>	FH1 constant speed remaining pulses start
52h	CNTFH <sub>2</sub>	FH2 constant speed start
53h	CNTUD <sub>1</sub>	Remaining pulses start using acceleration/deceleration (FL→FH1)
54h	CNTUD <sub>2</sub>	Remaining pulses start using acceleration/deceleration (FL→FH2)
60h	FCHGL	FL speed immediate speed change
61h	FCGH <sub>1</sub>	FH1 speed immediate speed change
62h	FCGH <sub>2</sub>	FH2 constant speed start
63h	FSCHL	Decelerate to FL speed
64h	FSCHH <sub>1</sub>	Accelerate/decelerate to FH1 speed
65h	FSCHH <sub>2</sub>	Accelerate/decelerate to FH2 speed

Note 1: Both the FH1 and FH2 speeds can be used to set the FH speed. These are both simply referred to as FH. This notation also applies to commands. For example, specifying a STAF<sub>H</sub> command means that you can use either SATFH<sub>1</sub> or STAFH<sub>2</sub>.

Note 2: Do not issue a start command while actually in operation (while outputting pulses). We cannot warranty the operation in this case.

Note 3: You must write an SRENC command before you write an SRST command.

## 6-1. Operation command

## 6-1-1. Start command

RCOM	Name	Description
40h	STAF <sub>L</sub>	Start at FL speed
41h	STAF <sub>H1</sub>	Start at FH1 speed
42h	STAF <sub>H2</sub>	Start at FH2 speed
43h	STAUD <sub>1</sub>	Start while accelerating from FL speed to FH1 speed
44h	STAUD <sub>2</sub>	Start while accelerating from FL speed to FH2 speed

## 6-1-2. Remaining pulses start command

RCOM	Name	Description
50h	CNTFL	Start feeding the remaining pulses at FL speed
51h	CNTFH <sub>1</sub>	Start feeding the remaining pulses at FH1 speed
52h	CNTFH <sub>2</sub>	Start feeding the remaining pulses at FH2 speed
53h	CNTUD <sub>1</sub>	Start feeding the remaining pulses while accelerating from FL speed to FH1 speed
54h	CNTUD <sub>2</sub>	Start feeding the remaining pulses while accelerating from FL speed to FH2 speed

The remaining pulses start command is used to feed the remaining pulses to an axis after a positioning operation has halted.

## 6-1-3. Speed change command

RCOM	Name	Description
60h	FCHGL	Instantly change to FL speed
61h	FCHGH <sub>1</sub>	Instantly change to FH1 speed
62h	FCHGH <sub>2</sub>	Instantly change to FH2 speed
63h	FSCHL	Decelerate to FL speed
64h	FSCHH <sub>1</sub>	Accelerate or decelerate to FH1 speed
65h	FSCHH <sub>2</sub>	Accelerate or decelerate to FH2 speed

## 6-1-4. Stop command

RCOM	Name	Description
30h	EMGSP	Emergency stop (stop even in the middle of the last pulse)
31h	STOP	Immediate stop (stop after output of the current pulse is complete)
32h	SDSTP	Deceleration stop

## 6-2. Control command

## 6-2-1. Software reset command

RCOM	Name	Description
01h	SRST	Reset the program
05h	SRDS	Disable the SRST command (default status)
0Ah	SREN	Enable the SRST command

## 6-2-2. Counter reset command

RCOM	Name	Description
13h	CUNR	Reset the current position counter

## 6-2-3. ERC output control command

Note: The PCD will ignore this command when the MTYP terminal is HIGH (while outputting an excitation sequence).

RCOM	Name	Description
10h	ERCO	Output an ERC signal
11h	ERCR	Reset the ERC output that was set to respond to a level output.

## 6-2-4. STA output control command

RCOM	Name	Description
12h	STAO	Output a one shot (LOW) signal for a simultaneous start through the #STA terminal.

## 6-2-5. Other commands

RCOM	Name	Description
00h	NOP	Does not affect operations.
14h	TRST	Reset the direction change timer.

## 7. Registers

### 7-1. RMD register (Address: 23h to 24h)

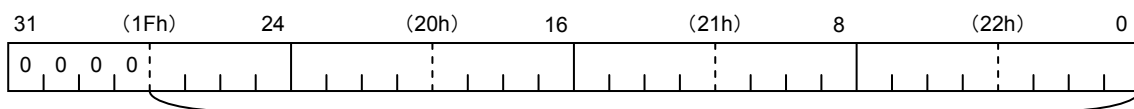
Registers used to set the operation mode.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	MSY	MSPE	MADJ	MINP	MCCE	MSDP	MSDE	MSMD	MDIR	MOD3	MOD2	MOD1	MOD0

Address	Bit	Name	Description	
24h	03 to 00	MOD3 to 0	Operation mode 0000: Continuous operation      1000: DR switch positioning operation 0001: Origin return operation      1001: Continuous pulsar operation 0010: Origin return operation with feed amount limit      1010: Pulsar positioning operation 0011: Move out from origin position      1011: Use prohibited 0100: Move out from EL      1100: Use prohibited 0101: Positioning operation      1101: Use prohibited 0110: Timer operation      1110: Use prohibited 0111: Continuous DR switch operation      1111: Use prohibited	
			04	MDIR Assign a moving direction 0: (+) direction 1: (-) direction <b>The setting for this bit is ignored when using an external switch (PDR, MDR).</b>
			05	MSMD Set acceleration/deceleration characteristics 0: Linear acceleration/deceleration 1: S-curve acceleration/deceleration
			06	MSDE 1: Enable SD input
			07	MSDP Ramp down point method (0: Auto setting, 1: Manual setting)
			23h	08
09	MINP 1: Enable the INP input			
10	MADJ 1: Disable FH correction			
11	MSPE 1: Deceleration stop or immediate stop on an STP input.			
12	MSY 1: After a start command is written, the start is delayed until #STA turns on.			
13	Not defined (Always set to 0)			
14	Not defined (Always set to 0)			
15	Not defined (Always set to 0)			

### 7-2. RMV register (Address: 1Fh to 22h)

This register is used to specify a number of output pulses in a positioning operation. Set the moving direction using RMD, MDIR.

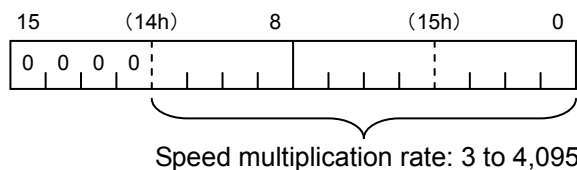


Number of output pulses: 0 to 268,435,455



### 7-3. RMG register (Address: 14h to 15h)

This register is used to set the speed multiplication rate.



$$(\text{RMG value}) = (\text{Reference clock frequency [Hz]} / (\text{Speed multiplication rate} \times 8192) - 1$$

$$(\text{Speed multiplication rate}) = (\text{Reference clock frequency [Hz]} / ((\text{RMG setting value} + 1) \times 8192)$$

When the reference clock is 9.8304 MHz, the relationship between these terms is as shown below.

Setting	Multiplication rate	Output speed range (pps)
2399 (95Fh)	0.5	0.5 to 4,095.5
1199 (4AFh)	1	1 to 8,191
599 (257h)	2	2 to 16,382
239 (0EFh)	5	5 to 40,955
119 (077h)	10	10 to 81,910

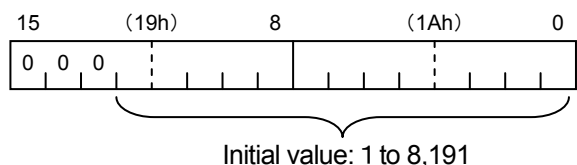
Setting	Multiplication rate	Output speed range (pps)
59 (03Bh)	20	20 to 163,820
23 (017h)	50	50 to 409,550
11 (00Bh)	100	100 to 819,100
5 (005h)	200	200 to 1,638,200
3 (003h)	300	300 to 2,457,300

### 7-4. RFL register (Address: 19h to 1Ah)

This register is used to specify the initial speed (stop speed) at high-speed (with acceleration/deceleration) operation that uses a speed step value.

The relationship between the operating speed and speed step value is as follows.

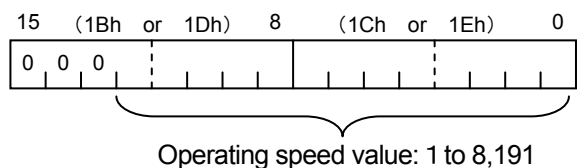
$$(\text{Speed step value}) \times (\text{Speed multiplication rate}) = \text{Speed [pps]}$$



### 7-5. RFH1, RFH2 register (FH1 address: 1Bh to 1Ch, FH2 address: 1Dh to 1Eh,)

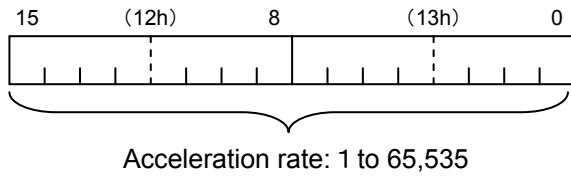
These registers are used to set an operating speed using a speed step value.

They are used to set the FH speed. Two registers RFH1 and RFH2 are available. They can be used to assign two different speeds, to change the FH speed during operation.



7-6. RUR register (Address: 12h to 13h)

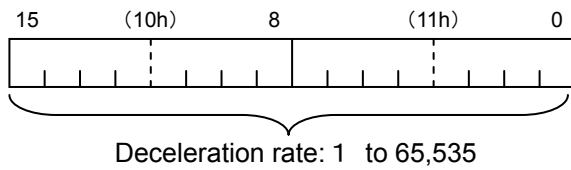
This register is used to set the acceleration rate.



7-7. RDR register (Address: 10h to 11h)

This register is used to set the deceleration rate.

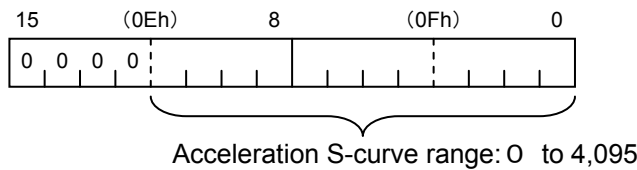
By setting this register to 0, the RUS setting will be applied.



7-8. RUS register (Address: 0Eh to 0Fh)

Sets the S-curve range in S-curve acceleration using the speed step value.

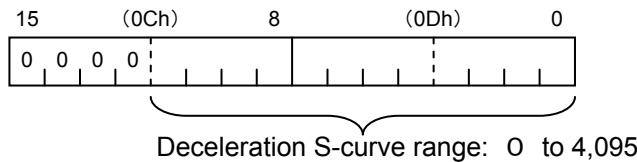
By setting this register to 0, the (RFH-RFL)/2 setting will be applied (RFH: either RFH1 or RFH2).



7-9. RDS register (Address: 0Ch to 0Dh)

Sets the S-curve range in S-curve deceleration using the speed step value.

By setting this register to 0, the (RFH-RFL)/2 setting will be applied (RFH: either RFH1 or RFH2).



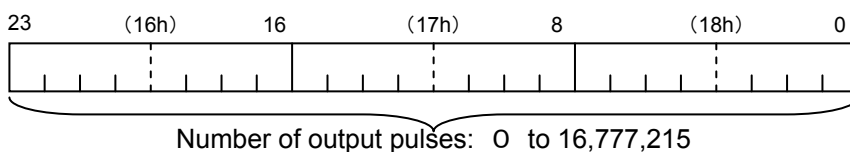
7-10. RDP register (Address: 16h to 18h)

This register is used to set the ramp down point (deceleration starting point) during a high speed positioning operation using acceleration/deceleration.

The ramp down point can be set automatically or manually. Select either method using RMD.MSDP.

When auto setting is selected (RMD.MSDP = 0), the RDP setting will be added to the auto setting value.

When manual setting is selected (RMD.MSDP = 1), the PCD will start decelerating when the *RDP setting* ≥ *Number of remaining pulses*.



## 7-11. RENV1 register (Address: 08h to 0Bh)

The following is the environment setting 1 registers.

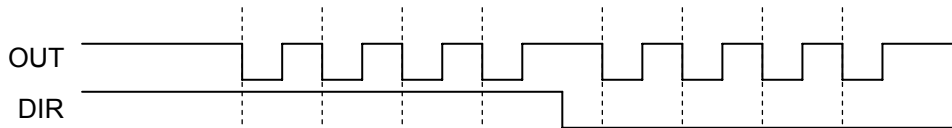
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKP	IDL2	IDL1	IDL0	PHMA	PHMK	EROR	EROE	CDWS	ETW2	ETW1	ETW0	ERCL	EPW2	EPW1	EPW0
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DTMF	ENDM	STAM	FLTR	INPL	DRL	STPL	STPM	ORGL	SDL	SDLT	SDM	ELM	PDM2	PDM1	PDM0

Address	Bit	Name	Description																																																	
0Bh	02 to 00	PMD2 to 0	The pulse output specification (When set to 101 or 110, the PCD will only apply 4x multiplication) <u>Note. 1</u>																																																	
			<table border="1"> <thead> <tr> <th rowspan="2">PMD 2 to 0</th><th colspan="2">Operation in the (+) direction</th><th colspan="2">Operation in the (-) direction</th></tr> <tr> <th>OUT output</th><th>DIR output</th><th>OUT output</th><th>DIR output</th></tr> </thead> <tbody> <tr> <td>000</td><td></td><td>High</td><td></td><td>Low</td></tr> <tr> <td>001</td><td></td><td>High</td><td></td><td>Low</td></tr> <tr> <td>010</td><td></td><td>Low</td><td></td><td>High</td></tr> <tr> <td>011</td><td></td><td>Low</td><td></td><td>High</td></tr> <tr> <td>100</td><td></td><td>High</td><td>High</td><td></td></tr> <tr> <td>101</td><td>OUT  DIR </td><td></td><td>OUT  DIR </td><td></td></tr> <tr> <td>110</td><td>OUT  DIR </td><td></td><td>OUT  DIR </td><td></td></tr> <tr> <td>111</td><td></td><td>Low</td><td>Low</td><td></td></tr> </tbody> </table>	PMD 2 to 0	Operation in the (+) direction		Operation in the (-) direction		OUT output	DIR output	OUT output	DIR output	000		High		Low	001		High		Low	010		Low		High	011		Low		High	100		High	High		101	OUT DIR		OUT DIR		110	OUT DIR		OUT DIR		111		Low	Low	
			PMD 2 to 0		Operation in the (+) direction		Operation in the (-) direction																																													
				OUT output	DIR output	OUT output	DIR output																																													
			000		High		Low																																													
			001		High		Low																																													
			010		Low		High																																													
			011		Low		High																																													
			100		High	High																																														
			101	OUT DIR		OUT DIR																																														
110	OUT DIR		OUT DIR																																																	
111		Low	Low																																																	
03	ELM	Sets the procedure when PEL or MEL is input (0: Immediate stop, 1: Deceleration stop) <u>Note 2</u>																																																		
04	SDM	Sets the procedure when the SD input turns on. (0: Deceleration only, 1: Deceleration stop)																																																		
05	SDLT	Set the SD input latch function (0: OFF, 1: ON) Turn this bit on when the SD signal is very short. If the SD input is off when starting operation, the latch signal will be reset. The latch signal is also reset when SDLT = 0.																																																		
06	SDL	Set the SD signal input logic. (0: Negative logic, 1: Positive logic)																																																		
07	ORGL	Set the ORG signal input logic (0: Negative logic, 1: Positive logic)																																																		
0Ah	08	TPSM	Set the operation when the STP input is on. (0: Immediate stop, 1: Deceleration stop)																																																	
	09	STPL	Set the STP signal input logic (0: Negative logic, 1: Positive logic)																																																	
	10	DRL	Set the PDR, MDR input logic (0: Negative logic, 1: Positive logic)																																																	
	11	INPL	Set the INP signal input logic (0: Negative logic, 1: Positive logic)																																																	
	12	FLTR	1: Apply a filter to the PEL, MEL, SD, ORG, and INP inputs. By applying a filter, signals shorter than 4 μs will be ignored.																																																	
	13	STAM	Specify the #STA input specification. (0: Level trigger, 1: Edge trigger)																																																	
	14	ENDM	0: Output a #BSY signal from #BSY/#END terminals. 1: Output an #END signal from #BSY/#END terminals.																																																	
15	DTMF	0: Delay the start timing of an operation by 0.2 ms after the DIR output changes. Enable this function by setting RENV1.PMD = 0 to 3. (In order to prevent incorrect operation of a motor driver.) 1: Disable the direction change timer 0.2 ms function.																																																		

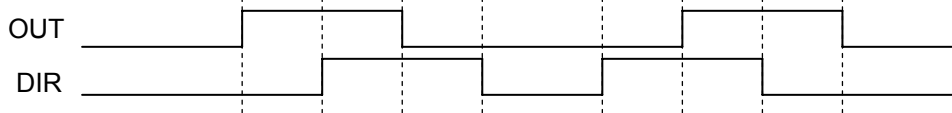
Address	Bit	Name	Description
09h	18 to 16	EPW2 to 0	Set the pulse length of the ERC output signal (preliminary excitation on time). <b>Note 3</b> 000: 12 $\mu$ s, 011:1.6ms, 110:104 ms, 001: 102 $\mu$ s, 100: 13 ms, 111: output, 010: 409 $\mu$ s, 101: 52 ms
	19	ERCL	Set the output logic of the ERC/CDW signal (0: Negative logic, 1: Positive logic)
	22 to 20	ETW2 to 0	Set the ERC signal OFF timer time (Excitation off delay). <b>Note 3</b> 000: 0 $\mu$ s, 011: 52 ms, 111: 416 ms, 001: 12 $\mu$ s, 100: 104 ms, 111: 832 ms, 010: 1.6 ms, 101: 208 ms
	23	CDWS	0: Output an ERC (deflection clear) signal from ERC/CDW. 1: Output a CDW (current reduction) signal from ERC/CDW terminal
08h	24	EROE	1: Automatically output the ERC signal when the motor stops immediately by a PEL or MEL input or by an emergency stop command.
	25	EROR	1: Automatically output the ERC signal when a origin return is complete.
	26	PHMK	1: Disable the excitation sequence output.
	27	PHMA	1: Disable the excitation sequence output when CDW is output.
	30 to 28	IDL2 to 0	Set the number of idling pulses: 0 to 7 When starting with acceleration, the PCD only starts accelerating after outputting the assigned number of pulses at FL speed.
	31	MSKP	1: Masks the pulse output and changes the excitation sequence output.

Note 1. Shown below is a comparison of output pulses when PMD (0 to 2) = "000"and "101."

PMD (0 to 2) = "000"



PMD (0 to 2) = "101"



Note 2. If the deceleration stop (ELM = 1) is set, the motor will start decelerating when the EL input turns on and it will stop when it passes the EL position. Therefore, pay special attention for this setting to avoid collisions of mechanical systems.

Note 3. This is used to reduce the current when CDSW =1.  
The time described above is correct when the reference clock is 9.8304 MHz.

## 7-12. RENV2 register (Address: 05h to 07h)

These are the environment setting 2 registers.

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	EPSE	EECE	ESDW	ESTA	ECMD	EEND	MSKI
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
STAF	PINF	PIM1	PIM0	EOFF	EINF	EIM1	EIM0	EZD3	EZD2	EZD1	EZD0	EZL	CUNI	ORCR	ORM

Address	Bit	Name	Description
07h	00	ORM	Select a origin return method. 0: At constant speed, the motor will stop immediately when the ORG input turns on. At high speed, the motor will decelerate and stop when the ORG input turns on. (By passing the ORG point) 1: At constant speed, the motor will stop immediately when the EZ signal counts up after the ORG signal turns on. At high speed, the motor will decelerate when the ORG input turns on, and it will stop when the EZ signal counts up.
	01	ORCR	1: In origin return operation, the PCD resets the current position counter at the origin position.
	02	CUNI	Select an input for the current position counter. (0: Output pulse, 1: EA, EB)
	03	EZL	Set the input logic of the EZ signal (0: Negative logic, 1: Positive logic)
	07 to 04	EZD3 to 0	Set the EZ count up value. Setting 0 (count to 1) to setting 15 (count to 16)
06h	09 to 08	EIM1 to 0	Set the EA, EB input specifications 00: 90 phase difference, 1x (Count up when the EA input phase advances) 01: 90 phase difference, 2x (Count up when the EA input phase advances) 10: 90 phase difference, 4x (Count up when the EA input phase advances) 11: Count up on the rising edge of EA, count down on the falling edge of EB.
	10	EINF	1: Apply a noise filter to EA, EB, and EZ. Ignore pulse inputs that last less than 3 CLK signal cycles.
	11	EOFF	1: Disable the EA, EB input
	13 to 12	PIM1 to 0	Set the PA/PB input specification 00: 90 phase difference, 1 x (Count up when the PA input phase advances) 01: 90 phase difference, 2 x (Count up when the PA input phase advances) 10: 90 phase difference, 4 x (Count up when the PA input phase advances) 11: Count up on the rising edge of PA, count down on the falling edge of PB.
	14	PINF	1: Apply a noise filter to the PA/PDR, PB/MDR input signals The filter characteristic depends on the operation mode (MOD setting) When PDR or MDR are input, Ignore pulse inputs that last less than 40 ms. When PA or PB are input, Ignore pulse inputs that last less than 30 ms.
15	STAF	1: Apply a noise filter to the #STA, STP input signals. When STAF=1, ignore pulse input that last less than 40 ms. Note: When you are using the STAO command, set STAF = 0.	

Address	Bit	Name	Description
05h	16	MSKI	1: Mask the INT output. (the internal interrupt circuit will change accordingly.)
	17	EEND	1: Send an interrupt during an auto stop.
	18	ECMD	1: Send an interrupt when stopped with a command.
	19	ESTA	1: Send an interrupt when performing a simultaneous start with the STA input.
	20	ESDW	1: Send an interrupt when starting deceleration.
	21	EECE	1: Send an interrupt when an EA/EB error occurs.
	22	EPSE	1: Send an interrupt when a PA/PB error occurs.
	23		Not defined (Always set to 0)
04h	31 to 24		Not defined (Always set to 0)

## 7-13. RSTS registers (Address: 26h to 29h)

The following is the status information registers.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SINT	SSTP	SSTA	SSDP	SCM3	SCM2	SCM1	SCM0	SERC	SINP	SORG	SDIN	SSD	SPEL	SMEL	SMVZ
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SPA	SPB	SEA	SEB	SZC3	SZC2	SZC1	SZC0	SEZ	SPFH	SPUB	SPHZ	SPH4	SPH3	SPH2	SPH1

Address	Bit	Name	Description
29h	03 to 00	SPH4 to 1	Monitor phase signals (PH4 to 1) (1 = HIGH)
	04	SPHZ	1: Excitation sequence is the default status
	05	SPUB	1: UB input is HIGH
	06	SPFH	1: FH input is HIGH
	07	SEZ	1: EZ input is ON
28h	11 to 08	SZC3 to 0	EZ counter value
	12	SEB	1: EB input terminal is HIGH
	13	SEA	1: EA input terminal is HIGH
	14	SPB	1: PB/MDR input terminal is HIGH
	15	SPA	1: PA/PDR input terminal is HIGH
27h	16	SMVZ	1: Countdown counter for positioning is at 0
	17	SMEL	1: MEL input terminal is on
	18	SPEL	1: PEL input terminal is on
	19	SSD	1: SD input terminal is on (SD latched)
	20	SDIN	1: SD input terminal is on (SD is set as an input terminal)
	21	SORG	1: ORG input terminal is on
	22	SINP	1: INP input terminal is on
	23	SERC	1: ERC output terminal is on
26h	27 to 24	SCM3 to 0	Operation status monitor 0000: Stopping                      0111: In FL constant speed operation 0001: Waiting for a DR input      1000: Accelerating 0010: Waiting for an STA input    1001: FH1 constant speed feeding 0011: Waiting for a PA/PB input    1010: FH2 constant speed feed 0100: Direction change timer counts    1011: Decelerating 0101: ERC timer counts            1100: Waiting for an INP input 0110: Current timer counts        1111: Processing a start/stop
	28	SSDP	1: Passing by the ramp down point
	29	SSTA	1: #STA terminal is on (LOW)
	30	SSTP	1: STP terminal is on
	31	SINT	1: #INT terminal is on (LOW)

7-14. RIST registers (Address: 2Ah to 2Bh)

The following registers are used to check the cause of an interrupt and reset it.  
 To reset a specific bit, write "1" in the bit you want.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	ISDW	ISTA	ICMD	IEND	IPSE	IECE	IPOV	ISSD	ISTP	IPEL	IMEL

Address:	Bit	Name	Description
2Bh	00	IMEL	1: Stopped on an MEL (-EL) input
	01	IPEL	1: Stopped on a PEL (+EL) input
	02	ISTP	1: Stopped on an STP input
	03	ISSD	1: Stopped when the SD input turned on
	04	IPOV	1: Pulsar circuit overflow
	05	IECE	1: EA/EB input error occurred (motor does not stop) <a href="#">Note 1</a>
	06	IPSE	1: PA/PB input error occurred (motor does not stop) <a href="#">Note 1</a>
2Ah	07	IEND	1: Automatic stop (positioning operation, etc.) <a href="#">Note 1</a>
	08	ICMD	1: Stopped by an immediate stop command or deceleration stop command <a href="#">Note 1</a>
	09	ISTA	1: Started with an STA input <a href="#">Note 1</a>
	10	ISDW	1: Started to ramp down <a href="#">Note 1</a>
	11		Always set to 0
	12		Always set to 0
	13		Always set to 0
	14		Always set to 0
15		Always set to 0	

When an interrupt occurs, the #INT terminal that output the interrupt goes LOW.  
 You can mask the #INT signal (RENV2.MSKI = 1). However, even when masked, the RIST register will change when an interrupt occurs.

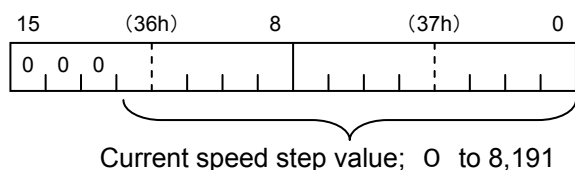
Note 1: To make one of these interrupts occur, you must enable the interrupt in the RENV2 register.

Interrupt flag	Enable
IECE	"EECE" bit in RENV2 (21)
IPSE	"EPSE" bit in RENV2 (22)
IEND	"EEND" bit in RENV2 (17)
ICMD	"ECMD" bit in RENV2 (18)
ISTA	"ESTA" bit in RENV2 (19)
ISDW	"ESDW" bit in RENV2 (20)



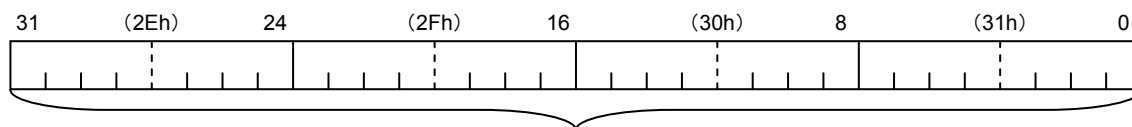
7-15. RSPD register (Address: 36h to 37h)

The current speed can be read from this register using the speed step value. (When stopped, this is 0.)



7-16. RCUN register (Address: 2Eh to 31h)

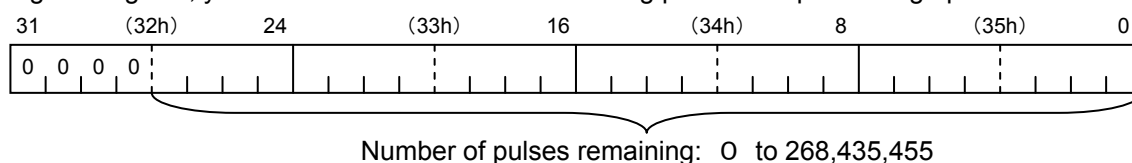
The following is the current position counter. Negative numbers are expressed as two's complements.



Current position : -2,147,483,648 to +2,147,483,647

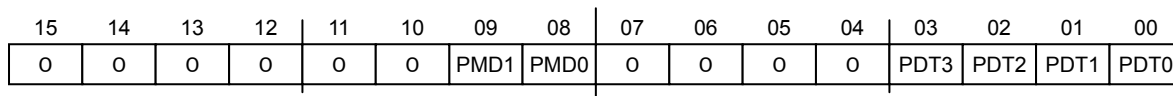
7-17. RDWC register (Address: 32h to 35h)

Using this register, you can read the number of remaining pulses in a positioning operation.



7-18. RIOP register (Address: 2Ch to 2Dh)

When the CPU system is selected (MODE terminal = LOW), you can set the general-purpose I/O terminals (P0 to P1) for input or output, and you can change general-purpose output terminals and check terminal status of the outputs.



Address	Bit	Name	Description
2Dh	00	PDT0	When writing: P0 output data, when reading: P0 terminal status (1 = HIGH)
	01	PDT1	When writing: P1 output data, when reading: P1 terminal status (1 = HIGH)
	02	PDT2	When writing: P2 output data, when reading: P2 terminal status (1 = HIGH)
	03	PDT3	When writing: P3 output data, when reading: P3 terminal status (1 = HIGH)
	04 to 07		Be sure to set to 0
2Ch	08	PMD0	Sets the P0 terminal (0: General-purpose input, 1: General-purpose output)
	09	PMD1	Sets the P1 terminal (0: General-purpose input, 1: General-purpose output)
	10 to 15		Be sure to set 0

The use of the general-purpose I/O or output terminals depends on the system mode (MODE, MTYP). In the stand-alone mode, none of the general-purpose ports can be used.

## 8. Operation Modes (RMD.MOD)

### 8-1. Continuous operation (MOD = "0000")

This mode is used to keep a motor turning after it is started by a start command, until it is stopped by a stop command.

Specify the direction of rotation in RMD.MDIR.

The motor will stop when the EL signal turns on.

When the positive direction is selected, the PEL terminal can be used. When the negative direction is selected, the MEL terminal should be used.

To start a motor in opposite direction after it is stopped by the EL signal, you have to write a new start command.

### 8-2. Origin return operation (MOD = "0001")

After the motor is started, the motor will stop when the ORG input turns on.

The stop method can be selected using the RENV2.ORM.

If you want to stop the motor by counting up pulses on the EZ input (by setting the RENV2.DRM = 1), specify the number to count up to in RENV2.EZD0 to 3. The PCD2112 counts up to that number +1 (EZD +1). If you make RENV2.EZD0 to 3 equal to 0, the PCD2112 will count one EZ input signal.

All of the start commands can be used: FL constant speed, FH constant speed, or acceleration/deceleration.

You can set the ERC signal to be output after a origin return operation is complete (See the RENV1 register).

Specify the feed direction using the RMD.MIDR.

### 8-3. Origin return operation with a feed amount limit (MOD = "0010")

After the motor is started, it will stop when the ORG input turns on, or after the PCD2112 outputs the number of pulses specified in RMV.

You can check the reason that the motor was stopped (at the origin position or by feeding the specified amount) by reading RSTS.SMVZ.

When RSTS.SMVZ = 0, the motor was stopped by the ORG input. When RSTS.SMVZ = 1, the motor was stopped by counting the specified number of pulses.

Use any start command for this operation: FL constant speed, FH constant speed, or acceleration/deceleration.

You can stop the motor by counting the pulses on the EZ input (See the origin return operation).

The PCD2112 can be set to output the ERC signal when the origin return operation completes (See RENV1 register.)

Set the feed direction using RMD.MDIR.

### 8-4. Move out of the origin position (MOD = "0011")

After a start command is written, the axis will move away from the origin position.

Make sure to use either the FL or FH constant speeds to move out of the origin position.

If you write a start command while the ORG input is OFF, the PCD will stop the movement on the axis as a normal stop, without outputting any pulses.

Since the PCD2112 samples the ORG input when outputting pulses, if the motor is started at constant speed while the ORG input is on, the PCD2112 will outputting one pulse and stop the motor normally after the ORG input turns off.

### 8-5. Move out of EL (MOD = "0100")

In this operation mode, the motor continues rotating until the EL input goes off.

When a start command is written while the EL signal is turned OFF, the PCD will stop the motor normally without outputting any pulses.

#### 8-6. Positioning operation (MOD="0101")

In this operation mode, the PCD2112 outputs the number of pulses set in the RMV register, and then will stop the motor automatically.

Set the moving direction in MRD.MDIR.

After starting, the value in the RMV register is copied to the RDWC register, and then the PCD2112 decrements this value.

#### 8-7. Timer operation (MOD="0110")

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 PCD2112 does not output any pulses in this operation.

Therefore, after a constant speed start command, the internal operation time will be a product of the output pulse frequency and the PMV register value.

In this operation mode the PCD2112 ignores any input signals on EPL, MED, SE, etc.

The STP input is enabled.

#### 8-8. Continuous operation using the DR input (MOD="0111")

In this operation mode, the PA/PDR and PB/MDR terminals are used as PDR and MDR inputs.

After a start command is written, turn on the PDR input and the motor will start to rotate in the positive direction. When the PDR input is turned off, the motor will stop operation. When the start with acceleration command is used, the motor will decelerate and stop when this input is turned off.

Similarly, with the MDR input, the motor will rotate in the negative direction.

Feed direction assignments using RMD.MDIR are ignored.

The motor will stop when the relevant EL signal (PEL, MEL) is turned on. However, the motor can be rotated in the reverse direction by turning on the DR input for the opposite direction, without having to write a new start command. Also, the PCD2112 does not output an interrupt signal when the motor is stopped by the EL signal.

#### 8-9. Inching operation using the DR input (MOD="1000")

In this operation mode, the PA/PDR and PB/MDR terminals are used as PDR and MDR inputs.

After a start command is written, turn on the PDR input and the motor will start to rotate in the positive direction for the number of pulses specified in the RMV register. When the MDR input is used, the motor will rotate in the negative direction.

Feed direction assignments using RMD.MDIR are ignored.

The motor will stop when the relevant EL signal (PEL, MEL) is turned on. However, the motor can be rotated in the reverse direction by turning on the DR input for the opposite direction, without having to write a new start command. Also, the PCD2112 does not output an interrupt signal when the motor is stopped by the EL signal.

Please note that this operation cannot be used as substitute positioning operation (MON = 0101) because an error might occur in accumulated feed amount after stopping with the EL signal. Use this operation for manual feeding. (There will not be an error between the mechanical position and RCUN.)

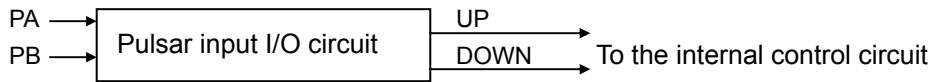
8-10. Continuous operation using a pulsar (MOD="1001")

By selecting this operation mode, both PA/PDR and PB/MDR terminals can function as pulsar input terminals (PA and PB).

After a start command is written, when the PCD2112 receives pulsar trains, the PCD2112 will output pulses according to the rotation direction specified for that pulsar.

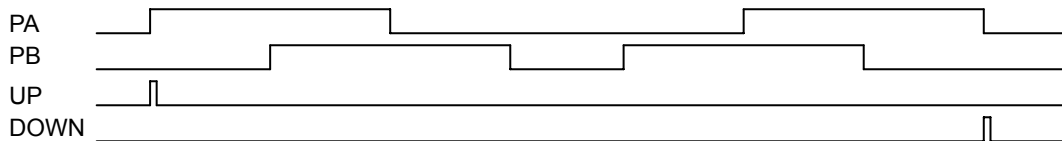
Make sure to use either the FL or FH constant speed start command.

The motor will stop when the EL (PEL, MEL) input turns on. By using the reverse pulsar terminal, the motor will also turn the other way without needing to write a new start command. The PCD2112 does not output an interrupt signal when the motor is stopped from the EL terminal.

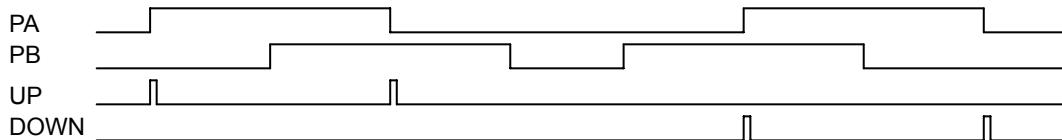


You may change pulse input specifications using RENV2.PIM.

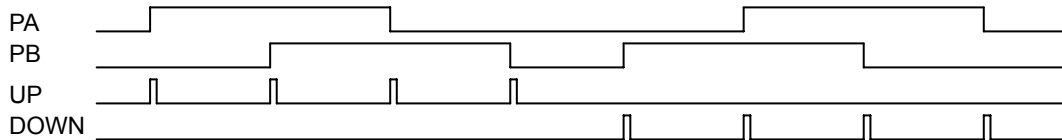
1) Example of a 90 phase difference signal, 1x input (RENV2.PIM = 00)



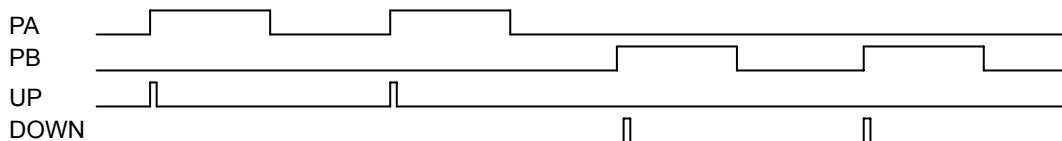
2) Example of a 90 phase difference signal, 2x input (RENV2.PIM = 01)



3) Example of a 90 phase difference signal, 4x input (RENV2.PIM = 10)



4) Example of a 2-pulse input (RENV2.PIM = 11)



8-11. Positioning operations using a pulsar (MOD="1010")

By selecting this operation mode, the PA/PDR, PB/MDR terminals function as pulsar input terminals (PA, PB).

If pulsar train signals are received by the PCD2112 after a start command is written, the PCD2112 will output the number of pulses specified in the RMV register in the direction specified in RMD.MDIR, synchronized with the pulsar's signal input.

Make sure to use either the FL or FH constant speed start command.

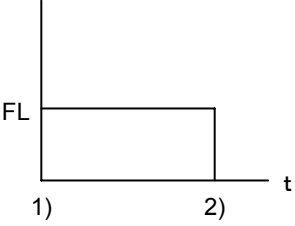
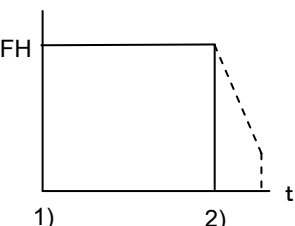
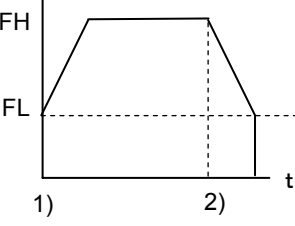
The motor will stop when the EL (PE, MEL) input turns on. Then, the PCD2112 will output an interrupt signal.

To restart this mode after it is stopped by an EL signal, you have to write a new start command.

This operation mode is only intended to be used for synchronous positioning, not for manual operation.

## 9. Speed Patterns

### 9-1. Speed patterns

Speed pattern	Continuous mode	Positioning operation mode
<p>FL constant speed operation</p>  <p>The graph shows a constant speed level (FL) on the vertical axis and time (t) on the horizontal axis. A horizontal line starts at point 1) and ends at point 2), representing a constant speed operation.</p>	<ol style="list-style-type: none"> <li>1) Write an FL constant speed start command (40h).</li> <li>2) Stop feeding by writing an immediate stop (31h) or a deceleration stop (32h) command.</li> </ol>	<ol style="list-style-type: none"> <li>1) Write an FL constant speed start command (40h).</li> <li>2) Stop when the positioning counter equals zero, or by writing an immediate stop (31h) / deceleration stop (32h) command.</li> </ol>
<p>FH constant speed operation</p>  <p>The graph shows a constant speed level (FH) on the vertical axis and time (t) on the horizontal axis. A horizontal line starts at point 1) and ends at point 2), after which a dashed line shows a deceleration ramp down to zero.</p>	<ol style="list-style-type: none"> <li>1) Write an FH constant speed start command (41h).</li> <li>2) Stop feeding by writing an immediate stop command (31h).</li> </ol>	<ol style="list-style-type: none"> <li>1) Write an FH constant speed start command (41h).</li> <li>2) Stop when the positioning counter equals zero, or by writing an immediate stop (31h) / deceleration stop (32h) command.</li> </ol>
<p>* When the deceleration stop command (32h) is written to the register, the PCD2112 will start decelerating.</p>		
<p>Operation with accel/decel.</p>  <p>The graph shows speed on the vertical axis (with levels FL and FH) and time (t) on the horizontal axis. The profile starts at FL, ramps up to FH (point 1), stays constant at FH, and then ramps down to FL (point 2) before stopping.</p>	<ol style="list-style-type: none"> <li>1) Write a start command with acceleration / deceleration (43h).</li> <li>2) Start deceleration by writing a deceleration stop command (32h).</li> </ol>	<ol style="list-style-type: none"> <li>1) Write a start command with acceleration (43h)</li> <li>2) The motor will start decelerating when the ramp down point is reached, or when an acceleration stop command (32h) is written.</li> </ol>
<p>* When the deceleration stop command (49h) is written to the register, the PCD2112 will stop the motor immediately.</p>		<p>* When the ramp down point is set manually (RMD.MSDP=1), and if the ramp down point setting value (RDP) is "0," the motor will stop immediately.</p>

### 9-2. Set speed pattern

Enter speed patterns in the registers listed below.

If the value you want is the same as the value previously written, you do not need to write it again.

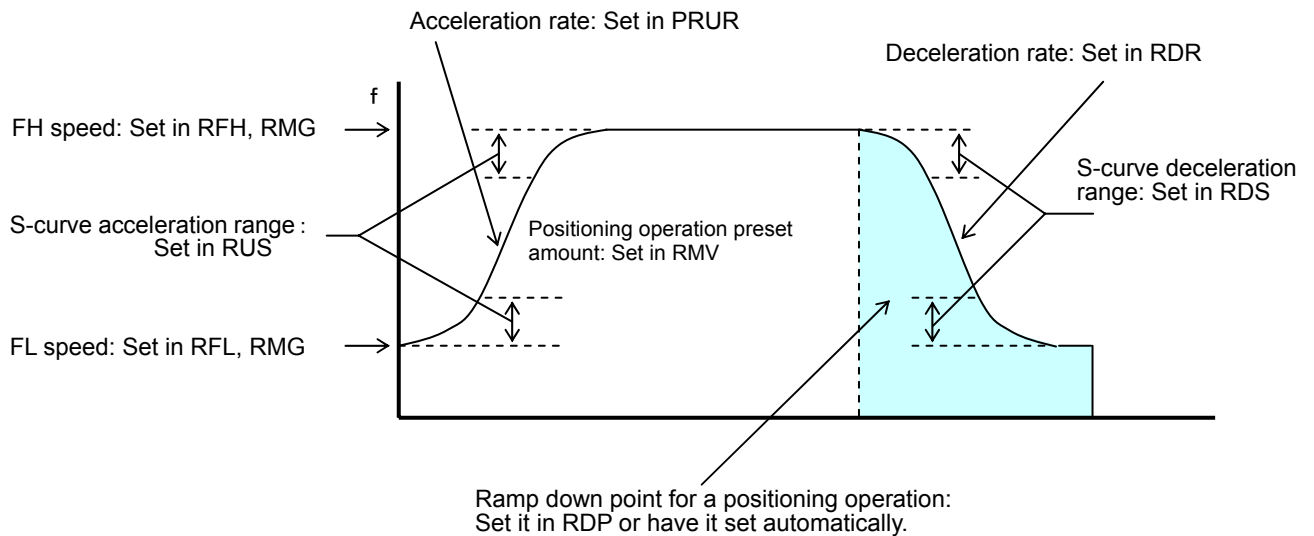
Be careful because certain registers recognizes 0 as outside the allowed range.

Register	Description	Bit length	Allowed range of settings	R/W
RMV	Positioning amount	28	0 to 268,435,455 (FFFFFFFh)	R/W
RFL	Initial speed	13	1 to 8,191 (1FFFh)	R/W
RFH	Operating speed <span style="float:right">Note 1</span>	13	1 to 8,191 (1FFFh)	R/W
RUR	Acceleration rate	16	1 to 65,535 (FFFFh)	R/W
RDR	Deceleration rate <span style="float:right">Note 2</span>	16	1 to 65,535 (FFFFh)	R/W
RMG	Speed multiplication rate	12	3 to 4,095 (FFFh)	R/W
RDP	Ramp down point	24	0 to 16,777,215 (FFFFFFFh)	R/W
RUS	S-curve range in acceleration	12	0 to 4,095 (FFFh)	R/W
RDS	S-curve range in deceleration	12	0 to 4,095 (FFFh)	R/W

Note 1: Use either RFH1 or RFH2 as the RFH.

Note 2: If the RDR is set to zero, the deceleration rate will be the value in RUR.

[How each register is used, in each area of an acceleration and deceleration operation, is shown below.]



Note: The Clock frequency term used in the equations below is the frequency of the CLK signal expressed in Hertz.

#### 9-2-1. RMV: Positioning amount setting register (28 bits)

This register is used to set a target position for a positioning operation.  
The allowable range of the settings is 0 to 268,435,455.

#### 9-2-2. RFL: FL speed setting register (13 bits)

This register is used to set the initial speed (and stop speed) for high-speed operation (with accel/decel). Specify an FL constant speed and a start speed for high-speed operations (acceleration/deceleration operations) in the range of 1 to 8,191.

The speed will be calculated from the value in PMG.

$$\text{FL speed [pps]} = \frac{\text{Clock frequency [Hz]}}{(\text{RMG} + 1) \times 8,192}$$

#### 9-2-3. RFH1, RFH2: FH speed setting register (13 bits)

This register is used to set the operating speed.

If you restart using another RFH while in operation (issue an FSCHH2 start command (65h) specifying FH2 while the PCD2112 is operating at FH1, or issue an FSCHH1 start command (64h) specifying FH1 while the PCD2112 is operating at FH2), you can change the FH speed on the fly.

Using serial communication, the PCD2112 can change speed with each byte. Therefore, if you change the FH1 or FH2 speed register while the motor is rotating at a specified speed, the motor may operate unexpectedly due to time difference in the register changes. (See 9-5. *Changing the speed pattern during operation.*)

Specify the FH constant speed and operating speed at high speed (acceleration/deceleration operations) in the range of 1 to 8,191.

Specify a value larger than RFL for high-speed operations (acceleration/deceleration operations).

The speed will be calculated from the value placed in PMG.

$$\text{Acceleration time [s]} = \text{RFH} \times \frac{\text{Clock frequency [Hz]}}{(\text{RMG} + 1) \times 8,192}$$

#### 9-2-4. RUR: Acceleration rate setting register (16 bits)

This register is used to set the acceleration rate.

Specify the acceleration rate for high-speed operations (acceleration/deceleration operations), in the range of 1 to 65,535 (0FFFFh).

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

1) Linear acceleration (RMD.MSMD = 0)

$$\text{Acceleration time [s]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1) \times 4}{\text{Clock frequency [Hz]}}$$

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

$$\text{Acceleration time [s]} = \frac{(\text{PFH} - \text{RFL}) \times (\text{RUR} + 1) \times 4}{\text{Clock frequency [Hz]}}$$

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

$$\text{Acceleration time [s]} = \frac{(\text{PFH} - \text{PFL} + 2 \times \text{RUS}) \times (\text{RUR} + 1) \times 2}{\text{Clock frequency [Hz]}}$$

## 9-2-5. RDR: Deceleration rate setting register (16 bits)

This register is used to set the deceleration rate.

Normally, the user specifies the deceleration rate for high-speed operations (acceleration/deceleration operations), in the range of 1 to 65,535 (0FFFFh).

Even if the ramp-down point mode is set to automatic (MSDP = 0 in the PRMD register), the values placed in the RDR register will be used as the deceleration rate, or set to 0.

However, when RDR = 0, the deceleration rate will be the value placed in the PUR.

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

1) Linear deceleration (RMD.MSMD = 0)

$$\text{Deceleration time [s]} = \frac{(\text{PFH} - \text{RFL}) \times (\text{RDR} + 1) \times 2}{\text{Clock frequency [Hz]}}$$

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

$$\text{Deceleration time [s]} = \frac{(\text{PFH} - \text{PFL}) \times (\text{RDR} + 1) \times 4}{\text{Clock frequency [Hz]}}$$

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

$$\text{Deceleration time [s]} = \frac{(\text{PFH} - \text{PFL} + 2 \times \text{RDS}) \times (\text{RDR} + 1) \times 2}{\text{Clock frequency [Hz]}}$$

## 9-2-6. RMG: Multiplication rate setting register (12 bits)

This register is used to set the speed multiplication rate.

Specify the relationship between the RFL and RFH settings and the speed, in the range of 2 to 4,095 (0FFFh). As the multiplication rate increases, the speed setting units will tend to be approximations.

Normally the multiplication rate should be set as low as possible.

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

$$\text{Multiplication rate} = \frac{\text{Clock frequency [Hz]}}{(\text{RMG} + 1) \times 8,192}$$

[Magnification rate setting example, when the reference clock = 9.8304 MHz]

Setting	Multiplication rate	Output speed range (pps)	Setting	Multiplication rate	Output speed range (pps)
2399 (95Fh)	0.5	0.5 to 4,095.5	59 (03Bh)	20	20 to 163,820
1199 (4AFh)	1	1 to 8,191	23 (017h)	50	50 to 409,550
599 (257h)	2	2 to 16,382	11 (00Bh)	100	100 to 819,100
239 (0EFh)	5	5 to 40,955	5 (005h)	200	200 to 1,638,200
119 (077h)	10	10 to 81,910	3 (003h)	300	300 to 2,457,300



## 9-2-7. RDP: Ramp down point setting register (24 bits)

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

<When set manually > (RMD.MSDP = 1)

Enter the number of pulses at which to start deceleration, in the range of 0 to 16,777,215 (0FFFFFFh). The PCD2112 will start decelerating when (RDP value)  $\geq$  (Remaining number of pulses)

Note: To obtaining an effective manual setting value, you have to know actual maximum speed.

When the feed amount is small, the motor will have to decelerate even while it is accelerating, or when the maximum speed must be modified by the FH correction function, the PCD2112 cannot calculate the manual value to use.

Therefore, when using manual ramp down setting, turn off the FH correction function.

In addition, if you want execute a manual FH correction, obtain the maximum speed after the correction. Then calculate the following.

Calculate the maximum value for the ramp down point as follows.

1) Linear deceleration (RMD.MSMD = 0)

$$\text{Optimum value [pulse]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 8,192}$$

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

$$\text{Optimum value [pulse]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1) \times 2}{(\text{RMG} + 1) \times 8,192}$$

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

$$\text{Optimum value [pulse]} = \frac{(\text{RFH} + \text{RFL}) \times (\text{RFH} - \text{RFL} + 2 \times \text{RDS}) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 8,192}$$

The motor will start decelerating when (Positioning counter value)  $\leq$  (RDP setting value).

If the value of set for the ramp down point is smaller than the optimum value, the speed while stopping will be faster than the FL speed. On the other hand, if it is larger than the optimum value, the motor will start an FL constant operation after decelerating.

<When automatic setting is selected> (RMD.MSDP = 0)

This is an offset value used to set the ramp-down point automatically. Set it within 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 FL speed after decelerating. When a negative number is entered, the deceleration start timing will be delayed. If an offset is not required, set it to zero.

When the value for the ramp-down point is smaller than the optimum value, the speed while 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.

## 9-2-8. RUS: Acceleration S-curve setting register (12 bits)

This register is used to set the S-curve range during acceleration.

Specify the S-curve acceleration range for S-curve acceleration/deceleration operations within the range of 1 to 4,095.

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

$$S_{SU} [\text{pps}] = \text{RUS} \times \frac{\text{Clock frequency [Hz]}}{(\text{PRMG} + 1) \times 8,192}$$

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

However, if zero is specified,  $(RFH - RFL) / 2$  will be used for internal calculations, and the operation will be an S-curve acceleration without a linear portion.

If the minimum value 1 is entered, the deceleration will be almost equal to a linear acceleration.

If you set a larger value than  $(RFH - RFL) / 2$ , the motor will not reach maximum acceleration speed. In this case, the acceleration time will be different from the results from the equation above. Therefore, enter a value smaller than  $(RFH - RFL) / 2$ .

## 9-2-9. RDS: Deceleration S-curve setting range register (12 bits)

This register is used to set the S-curve range during deceleration.

Specify the S-curve deceleration range for S-curve acceleration/deceleration operations in the range of 1 to 4,095.

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

$$S_{SD} [\text{pps}] = \text{RDS} \times \frac{\text{Clock frequency [Hz]}}{(\text{RMG} + 1) \times 8,192}$$

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

However, if zero is specified,  $(RFH - RFL) / 2$  will be used for internal calculations, and the operation will be an S-curve deceleration without a linear portion.

If the minimum value 1 is entered, the deceleration will be almost equal to a linear deceleration.

If you set a larger value than  $(RFH - RFL) / 2$ , the motor will not reach maximum acceleration speed. In this case, the deceleration time will be different from the results from the equation above. Therefore, enter a value smaller than  $(RFH - RFL) / 2$ .

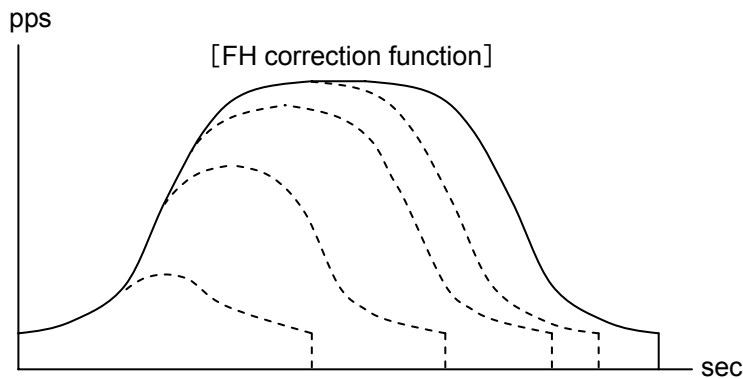
### 9-3. Manual FH correction

When the FH correction function is turned on (MAGD in the RMD register is set to 0), and if the feed amount in an acceleration/deceleration positioning operation is too small, this function will automatically lower the FH speed to suppress triangle motion profile.

The automatic ramp down point setting also follows the results of the FH correction.

However, the automatic ramp down point setting function can only be used when the acceleration curve and deceleration curve are symmetrical. That is, in order to use asymmetrical acceleration and deceleration curves, the ramp down point must be set manually. To obtain the correct manual setting value, you have to know the maximum speed.

Therefore, in this case, you have to turn off the FH correction function and set the FH value as a manual correction.



Automatic correction of the maximum speed according to the feed amount

< Example of a manual FH correction >

1) Linear acceleration/deceleration (RMD.MSMD = 0)

$$\text{When RMS} \leq \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RUR} + \text{RDR} + 2)}{(\text{RMG} + 1) \times 8,192}$$

$$\text{RFH} \leq \sqrt{\frac{(\text{RMG} + 1) \times 8,192 \times \text{RMV}}{\text{RUR} + \text{RDR} + 2} + \text{RFL}^2}$$

2) S-curve acceleration/deceleration without linear acceleration/deceleration portions (RMD.MSMD =1 and RUS = 0, RDS = 0)

$$\text{When RMV} \leq \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RUR} + \text{RDR} + 2)}{(\text{RMG} + 1) \times 8,192}$$

$$\text{RFH} \leq \sqrt{\frac{(\text{RMG} + 1) \times 8,192 \times \text{RMV}}{(\text{RUR} + \text{RDR} + 2) \times 2} + \text{RFL}^2}$$

3) S-curve acceleration/deceleration with linear acceleration/deceleration portions (RMD.MSMD = 1 and RUS > 0, RDS > 0)

3)-1. When RUS = RDS

(i) Use a small linear acceleration/deceleration range

$$\text{When } RMV \leq \frac{(RFH + RFL) \times (RFH - RFL + 2 \times RUS) \times (RUR + RDR + 2)}{(RMG + 1) \times 8,192} \quad \text{And}$$

$$RMV > \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 4}{(RMG + 1) \times 8,192} \quad \text{And}$$

$$RFH \leq -RUS + \sqrt{(RUS + RFL)^2 + \frac{(RMG + 1) \times 8,192 \times RMV}{(RUR + RDR + 2)}}$$

(ii) Eliminate the linear acceleration/deceleration range

$$\text{When } RMV \leq \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 4}{(RMG + 1) \times 8,192}$$

Change to S-curve acceleration/deceleration without linear acceleration/deceleration portions (set RUS = 0, RDS = 0),

$$RFH \leq \sqrt{\frac{(RMG + 1) \times 8,192 \times RMV}{(RUR + RDR + 2) \times 2} + RFL^2}$$

Reference:

RMV: Positioning feed amount, RFL: Initial speed, RFH: Operating speed, RUR: Acceleration rate, RDR: Deceleration rate, RMG: Speed magnification rate, RUS: S-curve range of acceleration, RDS: S-curve range of deceleration

## 3)-2. When RUS &lt; RDS

(i) Use a small linear acceleration/deceleration range

$$\text{When RMV} \leq \frac{(\text{RFH} + \text{RFL}) \times \{(\text{RFH} - \text{RFL}) \times (\text{RUR} + \text{RDR} + 2) + 2 \times \text{RUS} \times (\text{RUR} + 1) + 2 \times \text{RDS} \times (\text{RDR} + 1)\}}{(\text{RMG} + 1) \times 8,192}$$

$$\text{And RMV} > \frac{(\text{RDS} + \text{RFL}) \times \{\text{RDS} \times (\text{RUR} + 2 \times \text{RDR} + 3) + \text{RUS} \times (\text{RUR} + 1)\} \times 4}{(\text{RMG} + 1) \times 8,192}$$

$$\text{RFH} \leq \frac{-A + \sqrt{A^2 + B}}{\text{RUR} + \text{RDR} + 2}$$

However,

$$A = \text{RUS} \times (\text{RUR} + 1) + \text{RDS} \times (\text{RDR} + 1)$$

$$B = \{(\text{RMG} + 1) \times 8,192 \times \text{RMV} - 2 \times A \times \text{RFL} + (\text{RUR} + \text{RDR} + 2) \times \text{RFL}^2\} \times (\text{RUR} + \text{RDR} + 2)$$

(ii) Eliminate the linear deceleration range or use a small linear acceleration range

$$\text{When RMV} \leq \frac{(\text{RDS} + \text{RFL}) \times \{\text{RDS} \times (\text{RUR} + 2 \times \text{RDR} + 3) \times \text{RUS} \times (\text{RUR} + 1)\} \times 4}{(\text{RMG} + 1) \times 8,192}$$

$$\text{And RMV} > \frac{(\text{RUS} + \text{RFL}) \times \text{RUS} \times (\text{RUR} + \text{RDR} + 2) \times 8}{(\text{RMG} + 1) \times 8,192}$$

Change to S-curve deceleration without a linear acceleration/deceleration portion (Set RUS &gt; 0, RDS = 0),

$$\text{RFH} \leq \frac{-A + \sqrt{A^2 + B}}{\text{RUR} + 2 \times \text{RDR} + 3}$$

However,

$$A = \text{RUS} \times (\text{RUR} + 1)$$

$$B = \{(\text{RMG} + 1) \times 8,192 \times \text{RMV} - 2 \times A \times \text{RFL} + (\text{RUR} + 2 \times \text{RDR} + 3) \times \text{RFL}^2\} \times (\text{RUR} + 2 \times \text{RDR} + 3)$$

(iii) Eliminate the linear acceleration/deceleration

$$\text{When RMV} \leq \frac{(\text{RUS} + \text{RFL}) \times \text{RUS} \times (\text{RUR} + \text{RDR} + 2) \times 8}{(\text{RMG} + 1) \times 8,192} \quad \text{And}$$

Change to S-curve acceleration/deceleration with a linear acceleration/deceleration portion,  
(Set RUS=0, RDS=0),

$$\text{RFH} \leq \sqrt{\frac{(\text{RMG} + 1) \times 8,192 \times \text{RMV}}{(\text{RUR} + \text{RDR} + 2) \times 2} + \text{RFL}^2}$$

## Reference:

RMV: Positioning feed amount, RFL: Initial speed, RFH: Operating speed, RUR: Acceleration rate,  
RDR: Deceleration rate, RMG: Speed magnification rate, RUS: S-curve range of acceleration,  
RDS: S-curve range of deceleration

## 3)-3. When RUS &gt; RDS

(i) Use a small linear acceleration/deceleration range

$$\text{When RMV} \leq \frac{(\text{RFH} + \text{RFL}) \times \{(\text{RFH} - \text{RFL}) \times (\text{RUR} + \text{RDR} + 2) + 2 \times \text{RUS} \times (\text{RUR} + 1) + 2 \times \text{RDS} \times (\text{RDR} + 1)\}}{(\text{RMG} + 1) \times 8,192}$$

$$\text{And RMV} > \frac{(\text{RDS} + \text{RFL}) \times \{\text{RDS} \times (\text{RUR} + 2 \times \text{RDR} + 3) + \text{RUS} \times (\text{RUR} + 1)\} \times 4}{(\text{RMG} + 1) \times 8,192}$$

$$\text{RFH} \leq \frac{-A + \sqrt{A^2 + B}}{\text{RUR} + \text{RDR} + 2}$$

However,

$$A = \text{RUS} \times (\text{RUR} + 1) + \text{RDS} \times (\text{RDR} + 1)$$

$$B = \{(\text{RMG} + 1) \times 8,192 \times \text{RMV} - 2 \times A \times \text{RFL} + (\text{RUR} + \text{RDR} + 2) \times \text{RFL}^2\} \times (\text{RUR} + \text{RDR} + 2)$$

(ii) Eliminate the linear acceleration range, or use a small linear acceleration range,

$$\text{When RMV} \leq \frac{(\text{RDS} + \text{RFL}) \times \{\text{RUS} \times (2 \times \text{RUR} + \text{RDR} + 3) \times \text{RUS} \times (\text{RUR} + 1)\} \times 4}{(\text{RMG} + 1) \times 8,192}$$

$$\text{And RMV} > \frac{(\text{RUS} + \text{RFL}) \times \text{RUS} \times (\text{RUR} + \text{RDR} + 2) \times 8}{(\text{RMG} + 1) \times 8,192}$$

Change to S-curve acceleration without a linear acceleration/deceleration portion (set RUS = 0, RDS &gt; 0),

$$\text{RFH} \leq \frac{-A + \sqrt{A^2 + B}}{2 \times \text{RUR} + \text{RDR} + 3}$$

However,

$$A = \text{RDS} \times (\text{RDR} + 1)$$

$$B = \{(\text{RMG} + 1) \times 8,192 \times \text{RMV} - 2 \times A \times \text{RFL} + (2 \times \text{RUR} + \text{RDR} + 3) \times \text{RFL}^2\} \times (2 \times \text{RUR} + \text{RDR} + 3)$$

(iii) Eliminate the linear acceleration/deceleration range

$$\text{When RMV} \leq \frac{(\text{RUS} + \text{RFL}) \times \text{RUS} \times (\text{RUR} + \text{RDR} + 2) \times 8}{(\text{RMG} + 1) \times 8,192}$$

Change to S-curve acceleration/deceleration without linear acceleration/deceleration (set RUS = 0, RDS = 0),

$$\text{RFH} \leq \sqrt{\frac{(\text{RMG} + 1) \times 8,192 \times \text{RMV}}{(\text{RUR} + \text{RDR} + 2) \times 2} + \text{RFL}^2}$$

## Reference:

RMV: Positioning feed amount, RFL: Initial speed, RFH: Operating speed, RUR: Acceleration rate,  
 RDR: Deceleration rate, RMG: Speed magnification rate, RUS: S-curve range of acceleration,  
 RDS: S-curve range of deceleration

## 9-4. Examples of acceleration/deceleration pattern setting

Ex.: When the start speed = 20 pps, the operating speed = 110 kpps, and the acceleration/deceleration time = 300ms,

- 1) Set the reference clock frequency to 9830400 Hz (9.8304 MHz).
- 2) In order to output 110 kpps, set the multiplication rate to 20x.  
RMG = 59 (3Bh)
- 3) In order to set operating speed to 110 kpps, set the multiplication rate to 20x,  
RFH = 5500 (157Ch)
- 4) In order to set start speed to 20 pps, set the multiplication rate to 20x,  
RFL = 1 (1h)
- 5) In order to set acceleration/deceleration time to 300 ms, set RUR = 268 from the relationship between the acceleration/deceleration time and the RUR value.

$$\text{Acceleration time [e]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1) \times 2}{9,830,400}$$

$$0.3 = \frac{(5500 - 1) \times (\text{RUR} + 1) \times 2}{9,830,400}$$

From the above,

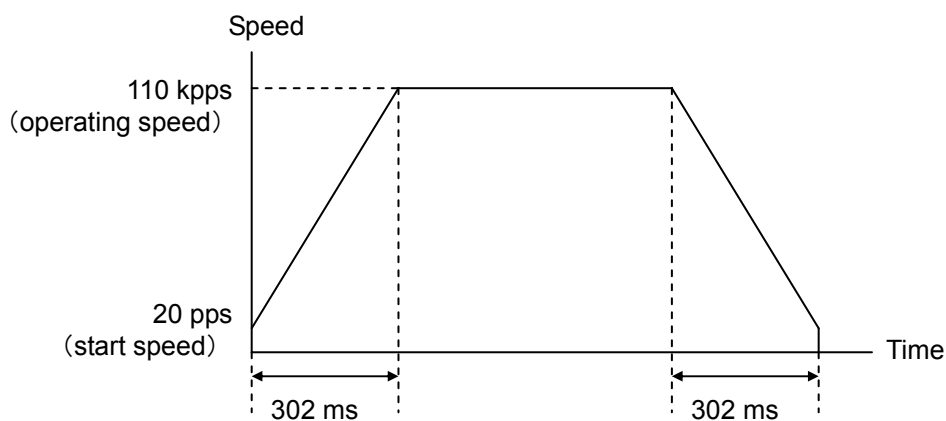
$$\text{RUR} = \frac{9,830,400 \times 0.3}{(5500 - 1) \times 2} - 1$$

$$\underline{\text{RUR} = 268.1505728}$$

However, only whole numbers can be placed in RUR, so put 268 or 269 into RUR.

That is, the actual acceleration/deceleration time will be 300.95 ms (when RUR = 268) or 302.07 ms (when RUR = 269)

A figure showing the speed pattern when RUR = 269



9-5. Changing the speed pattern during operation

You can change the speed registers (RFL, RFH1, and RFH2) in mid operation. The change is performed in units of bytes using serial communication. Due to the time difference between bytes, the motor may operate in an unintended pattern.

For example, a 10FFh or 0800h condition may occur while in the middle of changing RFH1 from 08FFh to 1000h.

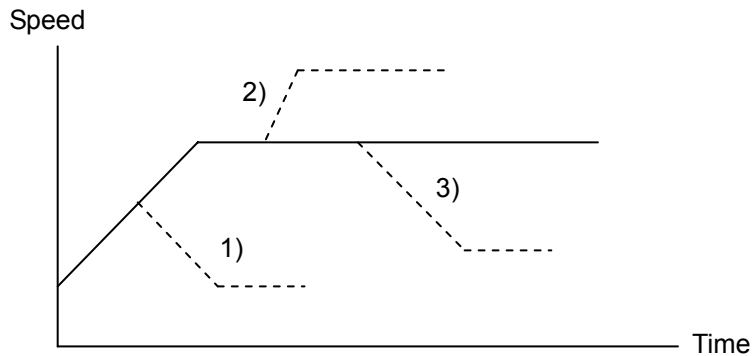
Therefore, if you want to change the speed during an operation, use the RFH1 and RFH2 registers alternately, as shown below.

- 1) Using a STAUD1 start command, start operation at FL speed and accelerate to FH1 speed.
- 2) Put new speed data into the RFH2 register.
- 3) Write an FSCHH2 command to change the speed control from FH1 to FH2 register.
- 4) Next enter new speed data in the RFH1 register.
- 3) Write an FSCHH1 command to change the speed control from FH2 to FH1 register.

If RFL, RUR, RDR, RUS, RDS, or RMG is changed while the ramp down point is being set automatically, the PCD2112 cannot handle the auto setting value and it may operate at FL speed after deceleration or it may not decelerate to FL speed.

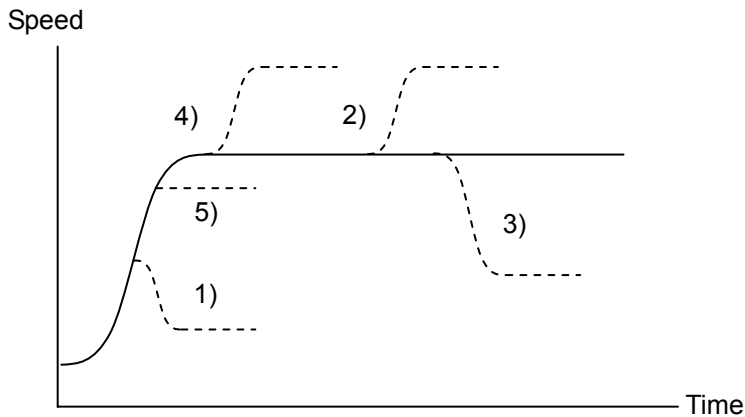
When changing speed using FH1 and FH 2 alternately, the speed pattern change is different between a linear acceleration/deceleration and an S-curve acceleration/deceleration.

An example of changing the speed pattern during linear acceleration/deceleration



- 1) When you change to FH2 speed while accelerating to FH1 speed, the motor will continue to accelerate/decelerate.
- 2) & 3) If the speed is changed to FH2 after the acceleration/deceleration is complete, the motor will continue its acceleration/deceleration to the new speed.

An example of a speed change during S-curve acceleration/deceleration





- 1) If you change to FH2 speed while accelerating to FH1 speed, and if  $FH2\ speed < current\ speed$ , the motor will make an S-curve deceleration to FH2 speed.
- 5) If you change to FH2 speed while accelerating to FH1 speed, and if  $FH2\ speed \geq current\ speed$ , the motor will accelerate to FH2 speed using the S-curve characteristics, and then maintain a constant speed at FH2 speed.
- 4) If you change to an FH2 speed that is larger FH1 while accelerating to FH1 speed, the motor will continue to accelerate to the original target speed using the original S-curve characteristics. Then it will accelerate again to the new FH2 speed.
- 2) & 3) If you change the FH speed after acceleration/deceleration has ended, the motor will accelerate/decelerate with the S-curve pattern to the new FH speed.

## 10. Other Functions

### 10-1. Output pulse control

#### 10-1-1. Output pulse modes

As the command pulse output modes, there are four common pulse modes, two 2-pulse modes and two 90 phase difference pulse modes available.

- Common pulse mode: Outputs feed pulses from the OUTP terminal and outputs the direction signal from the DIR terminal.
- 2-pulse mode: Outputs positive direction feed pulses from the OUT terminal and outputs negative direction feed pulses from the DIR terminal.
- 90 phase difference pulse mode: Outputs two pulses that have a 90 phase difference from the OUT terminal and the DIR terminal. A pair of pulses from these two terminals corresponds to one pulse in the common or 2-pulse mode.

Set the mode to use for outputting command pulses in RENV1.PMD (bit 0 to 2).

In the common pulse mode, if your motor driver needs some time after the direction signal changes before it is ready to receive fresh command pulses, use the PCD2112 direction change timer.

By setting RENV1.DTMF =0, the command pulse output can be delayed after a change of in direction signal for the amount of time stored in the direction change timer (0.2 ms).

Pulse output specifications (101 and 110 correspond to multiplication rate 4 only)					RENV1 (02 to 00)
PMD 2 to 0	In the positive direction		In the negative direction		PMD
	OUT output	DIR output	OUT output	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		
Setting the direction change timer (0.2 ms) function					RENV1 (15) DTMF

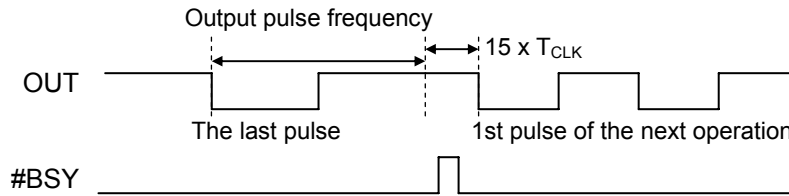
### 10-1-2. Output pulse length and operation complete timing

The output pulse length is 50% of the duty cycle.

If the RMG value is an odd number, a duty cycle error will occur and the ON time will be shorter than the OFF time, as shown in the formula below.

$$(\text{Pulse ON time}) / (\text{Pulse frequency}) = (\text{RMG value} / 2) / (\text{RMG value} + 1)$$

The BSY change timing when an operation ends is shown below.



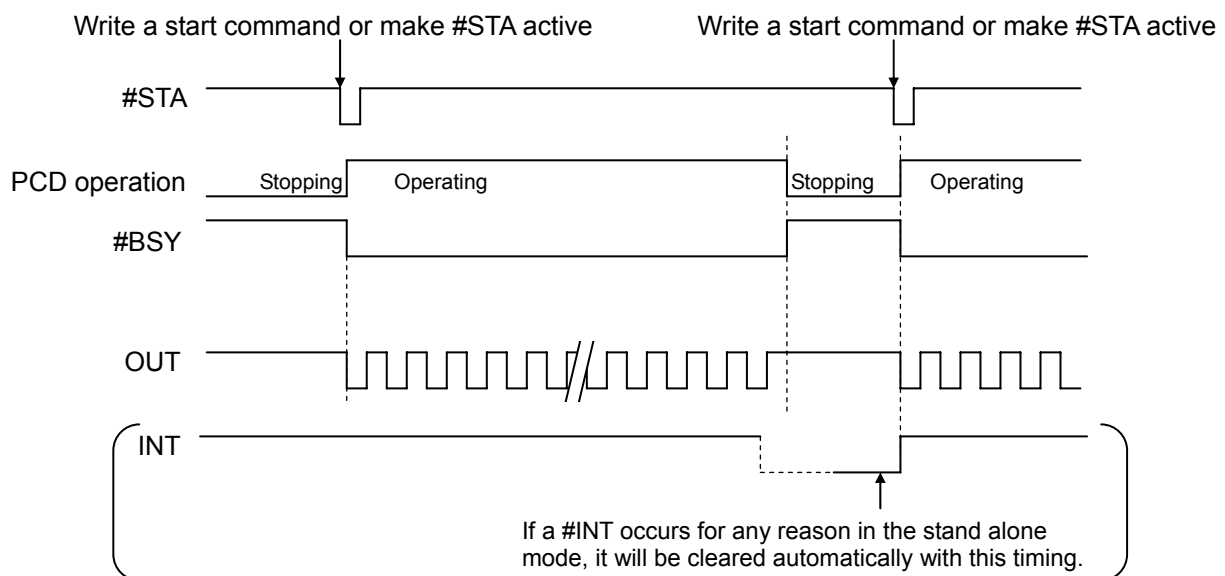
### 10-2. Operation status check output

#### 10-2-1. #BSY/#END signal (signals output during or at the end of an operation)

You can output a #BSY signal or a #END by setting RENV1.ENDM.

##### 1) Output a #BSY signal (during operation)

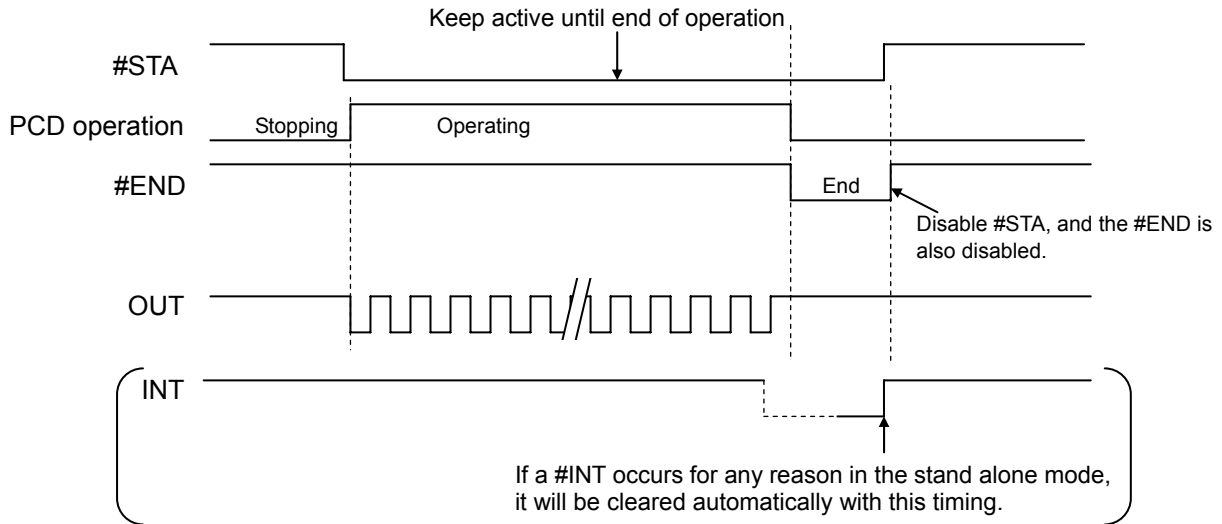
By setting RENV1.ENDM = 0, the PCD2112 will output a signal to indicate that it is in operation. If you are using a servomotor, disable the INP (in position) input (RMD, MINP=1), and the end of an operation can be delayed until the INP signal is input.



2) Output the #END signal (end operation)

In stand alone operation the operation will start when the #STA terminal becomes active. When the operation time is very short, the #BSY signal will be short too, and it may not be possible to check the operation status. In this case, set RENV1.ENDN = 1 and the PCD2112 can use a handshake method from the #STA signal.

If you are controlling a servomotor, enable the INP (in position) input (RMD, MINP = 1) and the end of the operation can be delayed until the INP is input.



MODE 0: CPU connected mode 1: Stand alone mode	MODE External terminal
Set whether or not to use the INP input 0: Disable INP input 1: Enable INP input	RMD (09) MINP
INP input logic 0: Negative logic 1: Positive logic	RENV1 (11) IMPL
Set the #BSY/#END terminal function 0: #BSY output 1: #END output	RENV1 (14) ENDM

### 10-3. Idling control

When the motor is started in an acceleration/deceleration operation, the acceleration can be started after a specified number of pulses at FL speed by using this function.

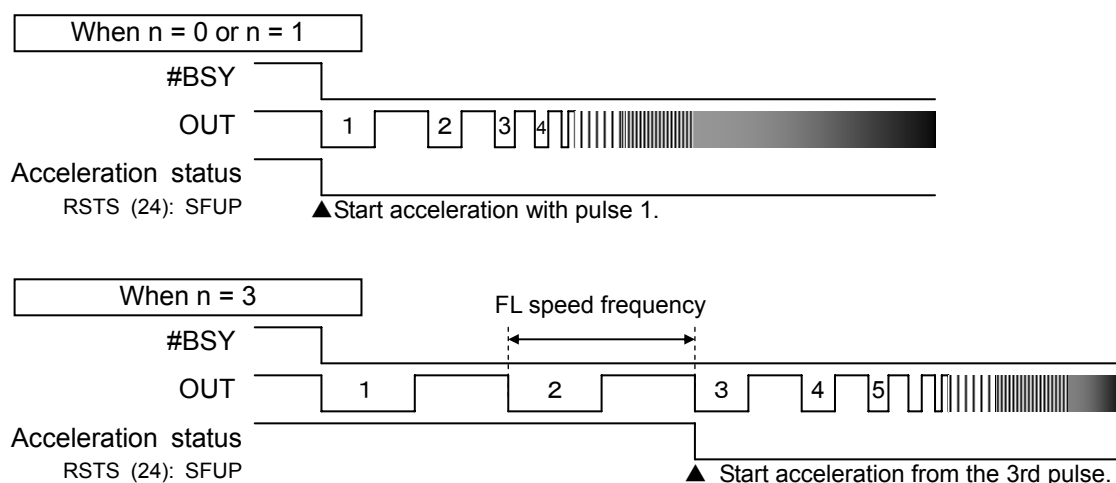
Set the number of pulses to delay the start in RENV1.IDL (0 to 7).

If you do not need this function, set this register to 0 or 1 and the motor will start accelerating at the same time as pulses are output. Therefore, the start speed obtained from the 2-pulse frequency will be faster than the FL speed.

To use this function, put 2 to 7 in the register. The motor will start accelerating when the “n” th pulse is output. Therefore, the start speed will be the FL speed, and the FL speed can be set to nearly the upper limit of the self-start speed.

The total feed amount will not change, even if this function is used for a positioning operation.

[Setting the number of idling pulses and the acceleration start timing]



<p>Output the assigned number of pulses at FL speed and then start accelerating. (IDL (2 to 0) -1) contains the specified number of pulses. If this value is negative value, the PCD2112 recognizes it as 0.</p>	<p>RENV1 (30 to 28) IDL (2 to 0)</p>
--	--

## 10-4. External mechanical inputs

## 10-4-1. PEL, MEL signals

When the PCD2112 receives an end limit signal of the motor moving direction (a PEL signal when rotating in the positive direction), the motor will stop immediately or decelerate and stop. The motor will remain stopped, even if the EL signal turns back off after stopping. However, for safety, you must also arrange a circuit to latch the on condition of the EL signal, until the motor reaches the stroke end.

If the EL signal is ON when a start command is written, the motor cannot be started in the same direction as the direction of the active EL signal.

By setting RENV1.ELM, you can select the operation to be used when the EL signal turns on: immediate stop, or decelerate and stop (only when started with acceleration).

**However, when Decelerate and Stop is selected, the motor will stop passing the EL position. The amount that the motor goes past the EP position depends on the speed. Make sure of no mechanical problem occurs if the motor goes past the EL position.**

In order to prevent further feeding in the direction of the limit after a deceleration and stop, arrange your circuit to latch the EL input on from time the motor passes the EL position until it reaches the deceleration stop complete position.

The minimum EL signal pulse length is approximately 4  $\mu$ s (depending on the reference clock), if the input filter is enabled. By disabling the input filter, the signal length can be as short as 0.1  $\mu$ s (or less depending on the reference clock).

You can monitor the EL signal by reading the RSTS.SMEL, and RSTS.SPEL.

By reading the RIST.IMEL or RIST.IPEL, you may monitor the cause of an interrupt that stopped the motor by turning on the EL signal.

When the timer mode is selected, the PCD ignores the EL signal. However, even in this case you can monitor the EL signal by reading the RSTS register.

You can set the input logic for the PEL signal on the PELL terminal and the input logic for the MEL signal on the MELL terminal.

Setting the input logic of PEL (+EL) signal LOW: Positive logic HIGH: Negative logic	PELL external terminal
Setting the input logic of MEL (-EL) LOW: Positive logic HIGH: Negative logic	MELL external terminal
Stop pattern when stopped by a PEL or MEL signal 0: Immediate stop when PEL or MEL turns on 1: Decelerate and stop when PEL or MEL turns on	RENV1 (03) ELM
Reading PEL, MEL signals SPEL = 0: PEL signal is off, SPEL = 1: PEL signal is on SMEL = 0: MEL signal is off, SMEL = 1: MEL signal is on	RSTS (18) SPEL RSTS (17) SMEL
Applying an input filter to PEL, MEL 1: Apply a filter to the $\pm$ EL inputs. By applying a filter, the PCD2112 will ignore signals that are shorter than 4 $\mu$ s.	RENV1 (12) FLTR

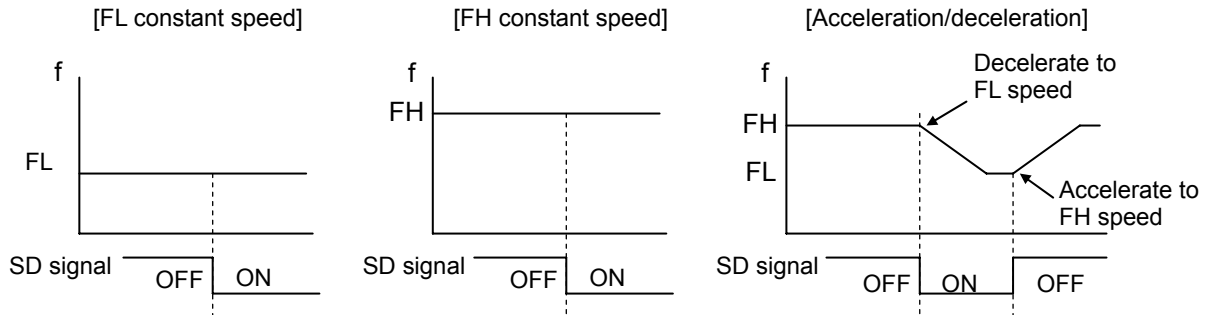
10-4-2. SD signal

When RMD.MSDE = 0, the PCD2112 ignores the SD signal.

When the SD signal is enabled, and if the SD signal goes on during an operation, the motor will: 1) decelerate, 2) latch and decelerate, 3) decelerate and stop, or 4) latch, decelerate and stop, depending on the settings in RENV1.SDM and RENV1.SDLT.

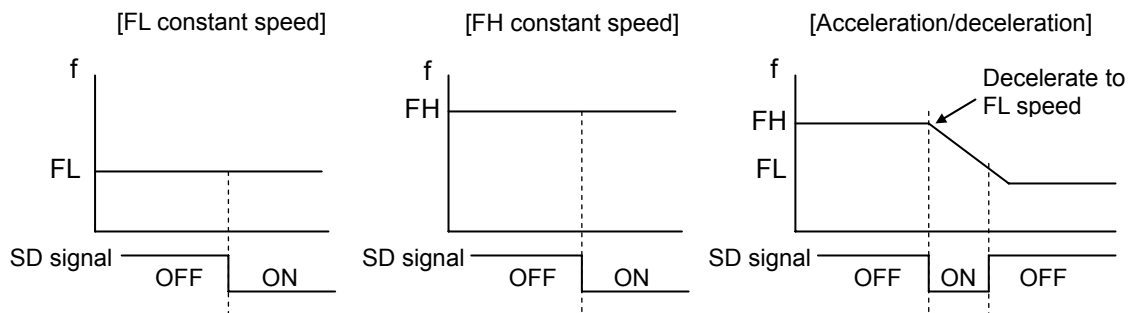
1) Decelerate <RENV1.SDM = 0, RENV1.SDLT = 0>

- While feeding at constant speed, the PCD2112 ignores the SD signal.
- While feeding at high speed, the PCD will decelerate the motor to FL speed when the SD signal turns on. And, if the SD signal goes off during deceleration, the motor will accelerate again to FH speed.
- If the SD signal is already turned on when a high-speed start command is written (with acceleration/deceleration), the motor will rotate at FL speed. When the SD signal turns off, the motor will accelerate to FH speed.



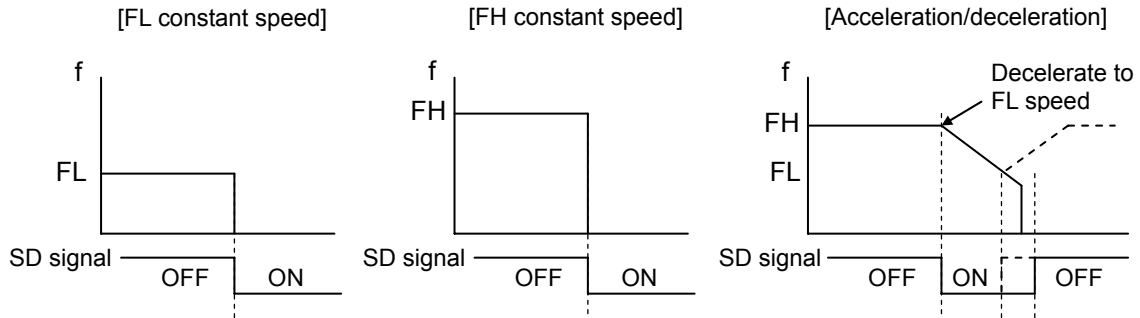
2) Latch speed <RENV1.SDM = 0, RENV1.SDLT = 1>

- While feeding at constant speed, the PCD2112 ignores the SD signal.
- While feeding at high speed, the PCD2112 will decelerate the motor to FL speed when the SD signal turns on. And, even if the SD signal goes off during deceleration, the motor will maintain FL speed and will not accelerate to FH speed.
- If the SD signal is already on when a high-speed start command is written (with acceleration/deceleration), the motor will rotate at FL speed. When the SD signal turns off, the motor will not accelerate to FH speed.
- The PCD2112 watches the SD input status when the next start command is written.



3) Deceleration stop <RENV1.SDM=1 and RENV1.SDLT = 0>

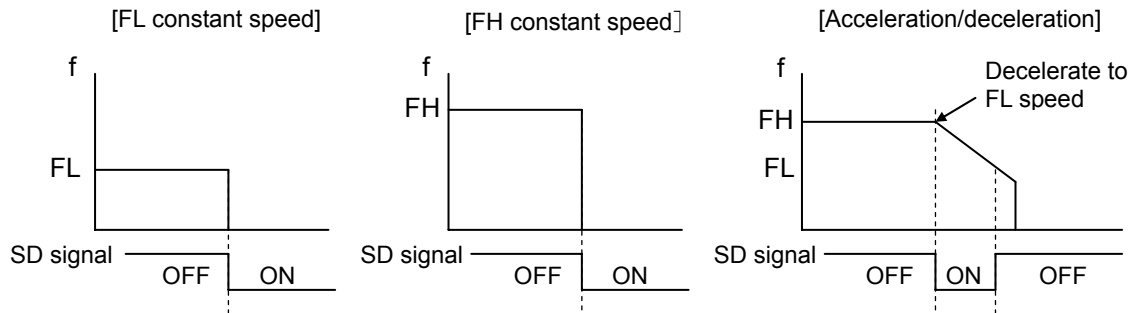
- If the SD signal turns on while feeding at constant speed, the motor will stop immediately.
- While feeding at high speed, the PCD2112 will decelerate the motor to FL speed and stop when the SD signal turns on. If the SD signal turns off during deceleration, the motor will accelerate back to FH speed.
- Once stopped, the motor will remain stopped, even if the SD signal turns off after a deceleration stop.
- If the SD signal is already on when a start command is written, the motor will not start and the operation will end.
- The PCD2112 will generate an interrupt while the motor is stopping.



If the SD signal turns off during deceleration, the motor will accelerate to FH speed (it will not stop).

4) Latched, deceleration stop <RENV1.SDM=1 and RENV1.SDLT = 1>

- If the SD signal turns on while feeding at constant speed, the motor will stop immediately.
- While feeding at high speed, the PCD2112 will decelerate the motor to FL speed when the SD signal turns on. Even if the SD signal goes off during the deceleration, the motor will not accelerate again.
- If the SD signal is already on when a start command is written, the motor will not start and the operation will end.
- The PCD2112 generates an interrupt while the motor is stopping.



The latched SD signal will be reset, even if the latch input setting bit (SDLT in RENV1) is set to 0. The minimum SD signal pulse length is 40 reference clock cycles (4.0 μs) when the input filter is applied. By turning off the input filter, the minimum signal length will be one reference clock cycle (0.1 μs). (CLK = 9.8304MHz)

You can monitor the SD signal latch condition by reading the RSTS.SSD, and the SD terminal condition by reading the RSTS.SDIN. You can monitor the cause of the interrupt by reading the RIST.ISSD.



Enable/disable SD signal input 0: Disable SD signal input 1: Enable SD signal input	RMD (06) MSDE
SD signal input logic 0: Negative logic 1: Positive logic	RENV1 (06) SDL
Operation when SD signal turns on 0: Decelerates and runs at FL constant speed 1: Decelerates and stops	RENV1 (04) SDM
SD signal setting 0: SD signal is a level input 1: SD signal is a latched input Release the latch during the next start or change the setting to level input.	RENV1 (05) SDLT
Reading the latch status for the SD signal 0: The SD latch signal is off. 1: The SD latch signal is on.	RSTS (19) SSD
Reading the SD signal status 0: The SD signal is off. 1: The SD signal is on.	RSTS (20) SDIN
Reading the cause of the interrupt when stopped by the SD signal 1: Decelerated and stopped because the SD signal turned on	RIST (03) ISSD
Applying the SD input filter 1: Apply the filter to the SD input By applying the filter, the PCD2112 will ignore pulses shorter than 4 $\mu$ s.	RENV1 (12) FLTR

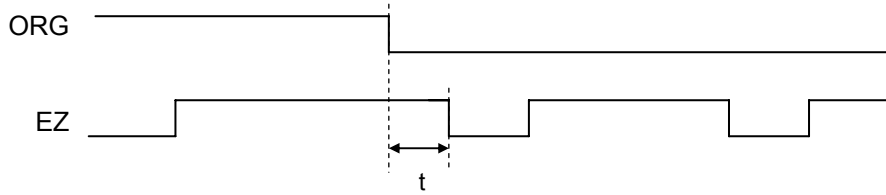
10-4-3. ORG, EZ signals

These signals are enabled in the origin return operation modes (origin return, origin return with limitation, move away from the origin position). Select the operation mode and feed direction using the RMD (operation mode) register.

The ORG signal is sampled on the rising edge of the output pulse (using negative logic). Therefore, the pulse length of the ORG signal must be the length of an output pulse, or longer. Since the ORG signal input is latched inside the PCD2112, you do not need to latch the condition externally. The latched ORG signal will be reset when the motor stops. You can change the input logic of the ORG signal using the RENV1.ORGL. You can change the input logic of the EZ signal using the RENV2.EZL.

You can monitor the condition of the ORG terminal and the EZ terminal in the RSTS.SORG and RSTS.SEZ. For details about the origin return operation mode, see section 8-2, *Origin return operations*.

ORG signal and EZ signal timing



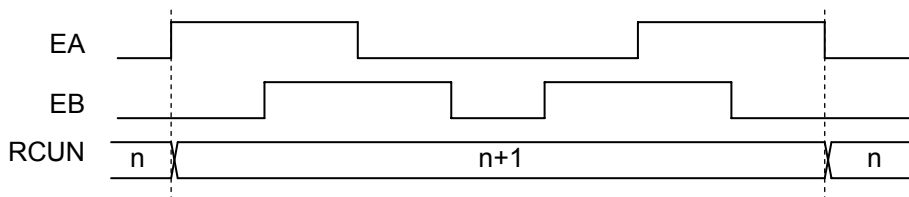
- (i) Counts when  $t \geq 4 \times T_{CLK}$
  - (ii) Not determined in this state  $T_{CLK} < t < 4 \times T_{CLK}$
  - (iii) Does not count when  $t \leq T_{CLK}$
- ( $T_{CLK}$ : Reference clock cycle)

Setting modes that enable the ORG and EZ signals 0001: Origin return operation, 0010: Origin return operation with a feed amount limit 0011: Move out of origin position	RMD (3 to 0) MOD (3 to 0)
ORG signal input logic 0: Negative logic      1: Positive logic	RENV1 (07) ORGL
The ORG signal read timing 0: When the ORG signal turns off      1: When the ORG signal turns on	RSTS (21) SORG
EZ count number Put a number in the EZ counter in an EZ count operation when the origin return completes Enter a value (number -1) in one of the EZ counters EZD3 to 0. The range is 0 to 15	RENV2 (7 to 4) EZD(3 to 0)
EZ signal input logic 0: Falling edge      1: Rising edge	RENV2 (03) EZL
The EZ signal read timing 0: When the EZ signal turns off      1: When the EZ signal turns on	RSTS (07) SEZ
EZ input filter 1: Apply the filter to the EZ input The PCD2112 will ignore input signals shorter than 3 clock cycles.	RENV2 (10) EINF

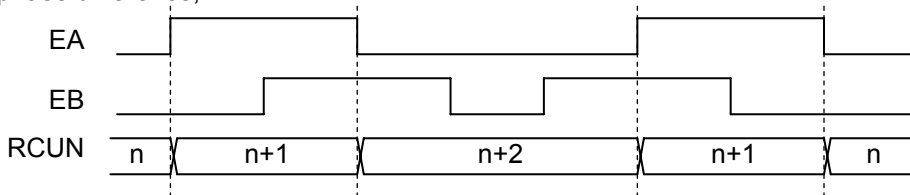
10-4-4. EA, EB signals

These are inputs from an encoder indicating mechanical positions.  
 Set the EA/EB signal input method in RENV2.EIM.  
 The input timing for the EA and EB signals is shown below.

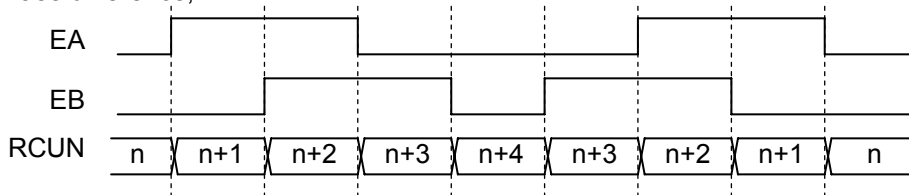
1) 90 phase difference, 1x



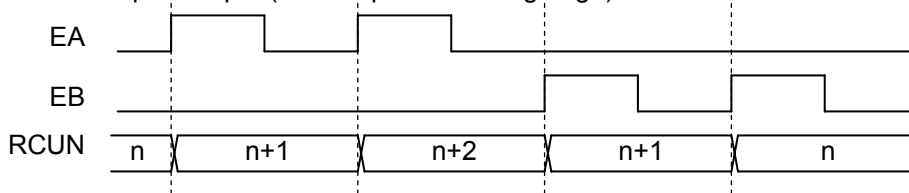
2) 90 phase difference, 2x



3) 90 phase difference, 4x



4) Up and down 2-pulse input (Count up on the rising edge)



The EA/EB signals input modes are set using RENV2.EIM as follows.

EIM setting	Operation
00	Input with 90 phase difference, 1x.
01	Input with 90 phase difference, 2x.
10	Input with 90 phase difference, 4x.
11	Count up on rising edge of EA, count down on the falling edge of EB.

## 10-5. Servomotor I/F

## 10-5-1. INP signal

A servo driver pulse train input has a deviation counter to count the difference between the number of command pulses input and the number of feed back pulses input. The servo driver function is to make this difference 0. In other words, the servomotor can operate by delaying the command pulses and even if the servo driver stops outputting command pulses, the servomotor continues to rotate until the deflection counter equals 0.

The PCD2112 can end an operation by receiving a positioning complete signal (INP signal) from the servo driver.

When the PCD2112 ends an operation because an INP signal is received, the PCD2112 turns on the BSY signal and changes SBST in the RSTS register, as soon as the INP signal is received. This means that both the BSY signal and SBST register are changed at exactly the same time the operation is complete.

You can change the input logic of the INP signal.

The minimum pulse length for an INP signal is approximately 4  $\mu$ s (depending on the reference clock) when the input filter is applied. But, by disabling the filter, the minimum pulse length will be 0.1  $\mu$ s (depending on the system clock frequency). (CLK=9.8304MHz)

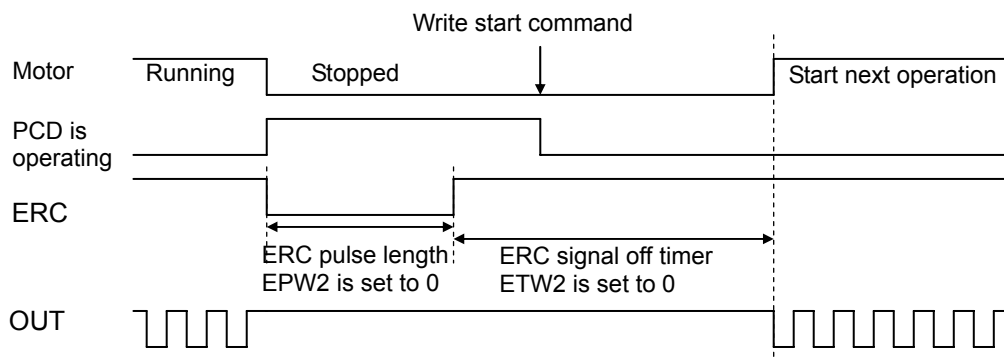
If the INP signal is already on when the output of pulses is complete, the PCD2112 will end the operation immediately. You can monitor the INP signal by reading the RSTS.SINP.

Delay completion of the operation using the INP signal 0: Do not delay completion of the operation 1: Delay completion of the operation (BSY) until INP goes ON	RMD (09) MINP
INP signal Input logic 0: Negative logic 1: Positive logic	RENV1 (11) INPL
Reading the INP signal condition 0: INP is OFF 1: INP is ON	RSTS (22) SINP
Applying a filter to the INP input 1: Apply the filter to INP input By applying the filter, the PCD will ignore all pulses shorter than 4 $\mu$ s.	RENV1 (12) FLTR

## 10-5-2. ERC signal (clear the deviation counter)

The servomotor can delay stopping, even if output of command pulses has stopped, until the deviation counter in the servo driver equals 0. In order to stop rotation of the servomotor immediately, such as in a origin return completion, you have to clear the deviation counter inside the servo driver.

By setting RENV1.CDWS=0, you can output a signal (ERC signal) that will clear the deviation counter in the servo driver. This signal may be output as a one shot or as a level signal. You can select either output by setting the RENV1.EPW0 to 2. If you need a time lag after the ERC signal returns HIGH (off) until the servo driver receives command pulses, you can set the ERC signal off timer in the RENV1.ETW.



To output an ERC signal when a origin return is completed, set RENV1.EROR = 1 and select ERC auto output.

To output the ERC signal in an immediate stop caused by the EL signal, STP signal, or an emergency stop command, set RENV1.EROE = 1 and select ERC signal auto output. (In case of a deceleration stop, the ERC will not be output, even if it is set to auto output.)

The ERC can be output by writing an ERC output command (ERCO).

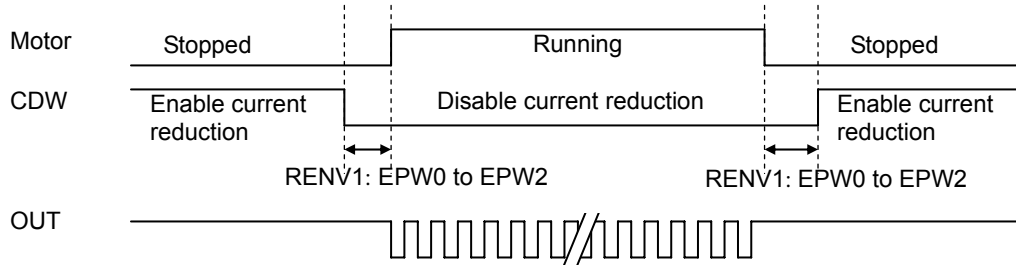
You can set the ERC signal output logic in the RENV1.ERCL. You can monitor the ERC signal by reading the RSTS.SERC.

ERC signal output condition 0: ERC signal output from ERC/CDW terminal 1: CDW signal output from ERC/CDW terminal	RENV1 (23) CDWS
ERC signal Auto output 0: Do not output an ERC signal when stopped by an input on PEL, MEL, or STP. 1: Automatically output an ERC signal when stopped by a PEL, MEL, or STP input.	RENV1 (24) EROE
ERC Auto output 0: Do not output an ERC signal when a origin return is completed. 1: Automatically output an ERC signal when a origin return is completed.	RENV1 (25) EROR
Set the ERC signal output length 000: 12 $\mu$ s      010: 409 $\mu$ s      100: 13 ms      110: 104 ms 001: 102 $\mu$ s      011: 1.6 ms      101: 52 ms      111: Level output	RENV1 (18 to 16) EPW (2 to 0)
ERC output logic 0: Negative logic      1: Positive logic	RENV1 (19) ERCL
Set the ERC signal off timer 000: 0 $\mu$ s      010: 1.6 ms      100: 104 ms      110: 416 ms 001: 12 $\mu$ s      011: 52 ms      101: 208 ms      111: 832 ms	RENV1 (22 to 20) ETW (2 to 0)
Read the ERC signal 0: ERC is OFF, 1: ERC is ON	RSTS (23) SERC
ERC output command Turn the ERC signal on.	Command 10h
ERC reset command Turn the ERC signal off	Command 11h

10-6. Stepper motor I/F

10-6-1. CDW signal (current reduction signal)

By setting RENV1.CDWS=1, the PCD2112 can output a current reduction signal from the ERC/CDW terminal. In this case, RENV1.EPW0 to 2 are used to set a delay time after the current reduction before starting operation. ETW0 to ETW2 are used to set the delay time after an operation completes until the current reduction is enabled.



ERC signal output 0: Output an ERC signal from the ERC/CDW terminal 1: Output a CDW signal from the ERC/CDW terminal	RENV1 (23) CDWS
Time after the current reduction is disabled before the operation starts 000: 12 $\mu$ s      010: 409 $\mu$ s      100: 13 ms      110: 104 ms 001: 102 $\mu$ s      011: 1.6 m      101: 52 ms      111: Level output	RENV1 (18 to 16) EPW (2 to 0)
Time after the operation is complete before a current reduction is enabled. 000: 0 $\mu$ s      010: 1.6 m      100: 104 ms      110: 416 ms 001: 12 $\mu$ s      011: 52 m      101: 208 ms      111: 832 ms	RENV1 (22 to 20) ETW (2 to 0)
CDW signal output logic 0: Negative logic 1: Positive logic	RENV1 (19) ERCL

10-6-2. PH1 to PH4 signals (excitation sequence)

The PCD2112 can output excitation sequences for a 2-phase stepper motor. Using the external FH terminal you can select a 2-2 phase or 1-2 phase excitation sequence. Using the UB external terminal you can select unipolar drive or bipolar drive.

[Unipolar excitation sequence]

2-2 phase excitation					
STEP→	0	1	2	3	0
PH1	H	H	L	L	H
PH2	L	H	H	L	L
PH3	L	L	H	H	L
PH4	H	L	L	H	H
(-)← Rotation direction →(+)					

1-2 phase excitation									
STEP→	0	1	2	3	4	5	6	7	0
PH1	H	H	H	L	L	L	L	L	H
PH2	L	L	H	H	H	L	L	L	L
PH3	L	L	L	L	H	H	H	L	L
PH4	H	L	L	L	L	L	H	H	H
(-)← Rotation direction →(+)									

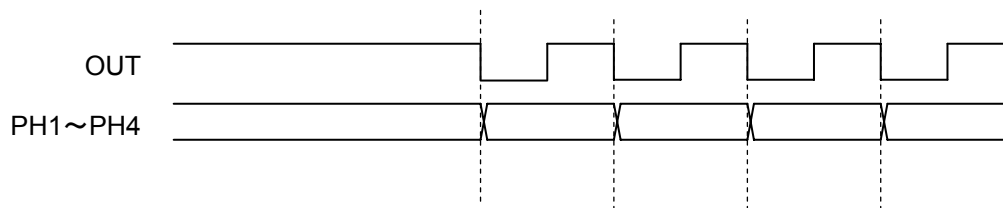
[Bipolar excitation sequence]

2-2 phase excitation					
STEP→	0	1	2	3	0
PH1	H	H	L	L	H
PH2	L	H	H	L	L
PH3	L	L	L	L	L
PH4	L	L	L	L	L
(-)← Rotation direction →(+)					

1-2 phase excitation									
STEP→	0	1	2	3	4	5	6	7	0
PH1	H	H	H	H	L	L	L	L	H
PH2	L	L	H	H	H	H	L	L	L
PH3	L	L	L	H	L	L	L	H	L
PH4	L	H	L	L	L	H	L	L	L
(-)← Rotation direction →(+)									

You can check the status of RSTS.SPH1 to 4.

The sequence changes according to the timing shown below.



10-7. External start/simultaneous start

10-7-1. #STA signal in CPU connection mode

The PCD2112 operation can be started by an external signal on the STA terminal. By setting MSY=1 in RMD, the PCD2112 will start when STA goes LOW.

When controlling multiple axes using multiple PCD2112s, connect the STA terminals of each PCD2112. Then, by setting all the axes to wait for an STA input and supplying a signal, they can all be started simultaneously. In this case, the start signal can be output from the STA terminal.

You cannot change logic of the STA terminal.

By setting the RENV2.ESTA = 1, an interrupt can be generated during a simultaneous start (when STA is on).

You can check the cause of the interrupt by reading the RIST.ISTA. You can also monitor the STA terminal status by reading the RSTS.SSTA.

<How to start axes simultaneously>

After setting RMD.MSY = 1, write a start command, and put the PCD in the waiting for an STA input condition. Then, start the operation using either of the following two methods.

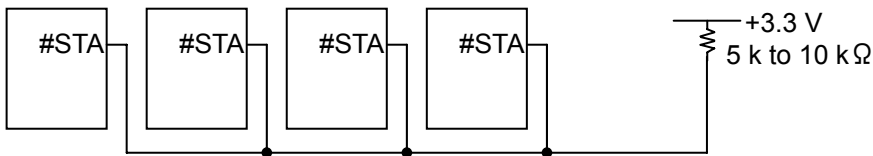
- 1) By writing a simultaneous start command, the PCD will output a one shot signal whose pulse length is 8 reference clock cycles (Approximately 0.9 μs when CLK = 9.8304 MHz)
- 2) Supply an external hardware signal.  
Supply the hardware signal through an open collector output (74LS06 or the like).

The input method for detecting the STA signal can be a level trigger or an edge trigger. However, when level trigger input is selected, the motor will start immediately when a start command is written to an STA terminal that is LOW.

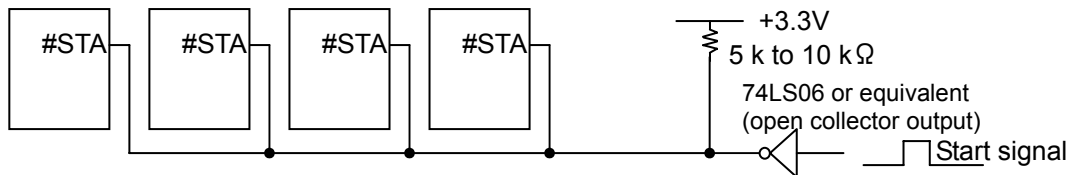
Even when the STA terminals of each PCD are connected, you can still start each axis independently by writing separate start commands.

To release the wait for an STA input condition, write an immediate stop command (32h).

- 1) To start multiple PCDs simultaneously, connect them as shown below.



- 2) To start multiple PCDs simultaneously using an external circuit, connect them as shown below.



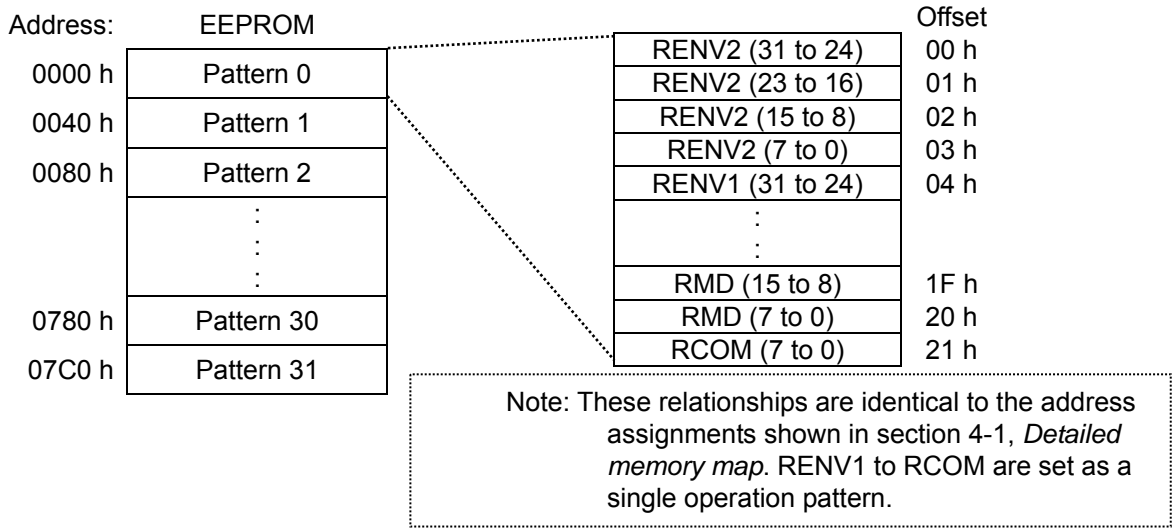
Provide a start signal pulse at least 5 CLK cycles long when RENV2.STAF = 0 and provide a one shot signal for CLK= 393,218 cycles (40 m sec.) or longer when RENV2.STAF = 1. **When ENDM in RENV1 is 1, the PDC2112 will see a STA signal as a level, so you must hold the STA signal LOW until the operation completes.**



Set the input conditions for wait for an STA signal 1: Start on an STA input	RMD (12) MSY
Set the STA signal input specifications 0: STA signal is a level trigger input 1: STA signal is an edge trigger input	RENV1 (13) STAM
STA input signal length 0: When the STA signal is shorter than 0.5 $\mu$ s., the PCD2112 will ignore the signal. 1: When the STA signal is shorter than 40 ms., the PCD2112 will ignore the signal	RENV2 (15) STAF
Read the STA signal 0: STA is OFF 1: STA is ON	RSTS (29) SSTA
Read the operation status 1100: Wait for an STA input	RSTS (27 to 24) SCM
Enable an STA caused interrupt 1: Interrupt caused by the STA input turning on	RENV2 (19) ESTA
Read whether the interrupt was caused by STA input turning on 1: An interrupt occurred because the STA input turned on.	RIST (09) ISTA
STA output command	Command 12h

10-7-2. #STA signal and PS0 to PS4 signals in stand alone mode

When the PCD2112 is used in stand alone mode, various operation patterns can be performed after storing them in an external serial EEPROM. The EEPROM must have at least 2 k-bytes capacity. The EEPROM can store the operation patterns (settings for each register) for more than one PCD.2112.



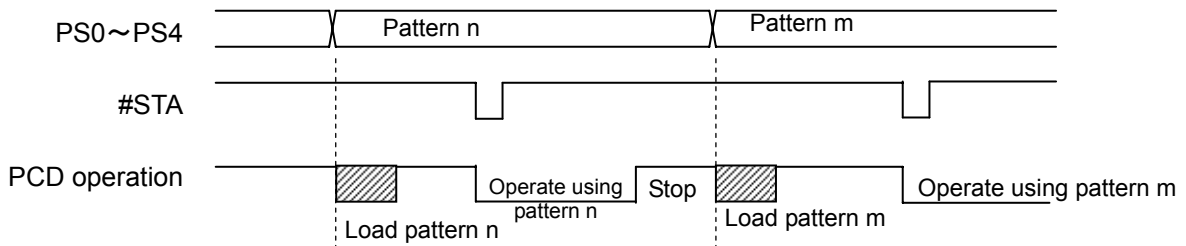
Use external terminals PS0 to PS4 to choose which pattern to use from the EEPROM.

PS (4 to 0)	Pattern to be selected
00000	Pattern 0
00001	Pattern 1
⋮	⋮
11110	Pattern 30
11111	Pattern 31

When the status of terminals PS0 to PS4 changes, the PCD2112 will read the selected operation pattern from the EEPROM. At this moment, since a start command (RCOM) is also read in, the PCD2112 will not wait for a #STA input.

Therefore, you have to set the RMD data in the EEPROM so that MSY = 1.

However, in case of an operation using the DR input or a continuous operation using the pulsar input, the PCD2112 will wait for a PA/PDR or PB/MDR input, which means that you may leave MSY = 0.



When RENV1.ENDM = 1, the PCD2112 responds to the STA signal as a level. Therefore, you must hold the STA signal LOW until the operation completes (until the END output goes LOW).

If the PS0 to PS4 status changes during an operation, the PCD will perform an immediate stop and read in the new pattern.

## 10-8. External stop/simultaneous stop

### 10-8-1. STP signal

The PCD2112 can perform an immediate stop or deceleration stop using an external signal on the STP terminal.

By setting RMD.MSTP = 1 to enable a stop from the STP input, the motor will stop immediately or decelerate and then stop when STP goes LOW. However, a deceleration stop will only be applied when the operation had a high speed start (with acceleration/deceleration). If a constant speed start was used, the motor will stop immediately.

When controlling multiple axes using multiple PCDs, connect the STA terminals of all the PCDs and supply a signal so that all the axes which are enabled to stop on an STP input can be stopped simultaneously. When stopping because the STP terminal received a signal, an interrupt occurs. By reading the RIST.ISTP, you can check the cause of the interrupt. You can monitor the STP terminal status by reading the RSTS.SSTP.

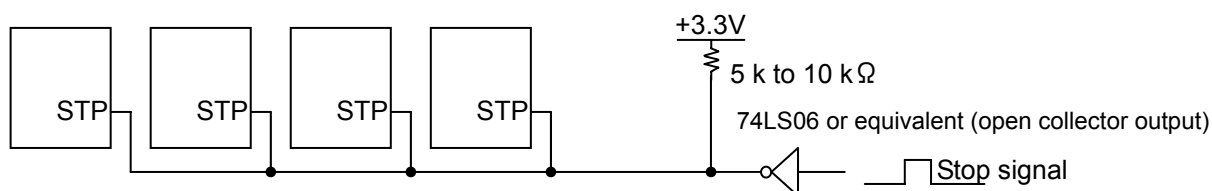
<How to stop all the axes simultaneously>

Set the axes to allow a simultaneous stop with RMD.MSPT (bit 11) = 1 and then start the operation.

When an STP signal is supplied, these axes will all stop simultaneously.

Supply the STP signal through an open collector output (74LS06 or equivalent).

Even though the STP terminals of multiple PCD2112s are connected, you can stop each axis independently by writing separate stop commands.



Provide a start signal pulse at least 5 CLK cycles long when RENV2.STAF = 0 and provide a one shot signal for CLK= 393,218 cycles (40 m sec.) or longer when RENV2.STAF = 1.

Enable waiting for an STP input 1: Enable a stop from an STP input (immediate stop/deceleration stop)	RMD (11) MSPE
Select the stop method by turning on the STP signal 0: Immediate stop when the STP signal turns on 1: Deceleration stop when the STP signal turns on	RENV1 (08) STPM
Set the STP input signal length 0: Ignore STP signals shorter than 0.5 $\mu$ s. 1: Ignore STP signals shorter than 40 ms.	RENV2 (15) STAF
Read the STP signal 0: STP signal is OFF 1: STP signal is ON	RSTS (30) SSTP
Read the cause of the interrupt 1: When stopped by an STP signal.	RIST (02) ISTP

## 10-9. Counter

## 10-9-1. Counter types and input methods

The PCD2112 has one counter circuit (RCUN) in addition to the position control counter (RDWC). Regardless of the operation mode selected, the positioning counter is loaded with the value in RMV (the target position), and the PCD2112 decrements the value each time a pulse is output.

The RCUN has the following specifications.

Item	RCUN specification
Counter function	Up and down counter
Bit length	32 bits including a sign (-2,147,483,648 to 2,147,483,647)
Count	Either command pulses or feedback pulses (EA/EB) from the encoder.

There are two methods for using the EA/EB input terminals by setting the RENV2.EIM0 to 1.

## 1) 90 phase difference signal (1x, 2x, or 4x)

Counts up if the EA input phase is earlier than the EB input. Counts down if the EA input phase comes later than the EB input.

## 2) Signal input method: Input (+) and (-) pulses

Counts up on a rising EA input and counts down on a rising EB input.

Setting the RCUN input 0: Output pulse 1: Encoder (EA/EB) input	RENV2 (02) CUNI
Setting the EA/EB input method 00: 90 phase difference, 1x    10: 90 phase difference, 4x 01: 90 phase difference, 2x    11: Separate input pulses for up and down	RENV2 (09 to 08) EIM (0 to 1)
Apply a noise filter to EA/EB/EZ 1: Apply the filter to each input By applying the filter, signals shorter than 3 reference clock cycles are ignored.	RENV2 (10) EINF
Enable an interrupt caused by an EA/EB error 1: Generate an interrupt when an EA/EB occurs.	RENV2 (21) EECE
Read an EA/EB error 1: An EA/EB error occurred.	RIST (05) IECE
Enable/disable EA/EB error input 0: Enable EA/EB error 1: Disable EA/EB error	RENV2 (11) EOFF
Disable RCUN operation 1: Stop the RCUN count operation (regardless of the counter input)	RMD (08) MCCE

## 10-9-2. Resetting the counter

The RCUN (current position counter) can be reset either by writing a command (20h), or automatically in a origin return operation.

Clear the RCUN counter in a origin return operation. 1: Clear the RCUN counter at the origin position in a origin return operation.	RENV2 (01) ORCR
RCUN counter clear command	Command 13h

## 11. Supplemental Description of the Stand Alone Operation

### 11-1. Requirements for selecting an EEPROM

When the PCD2112 is operated in the stand alone mode, it automatically sets its register values by reading the data from a serial EEPROM connected to it. The features required of this EEPROM are shown below.

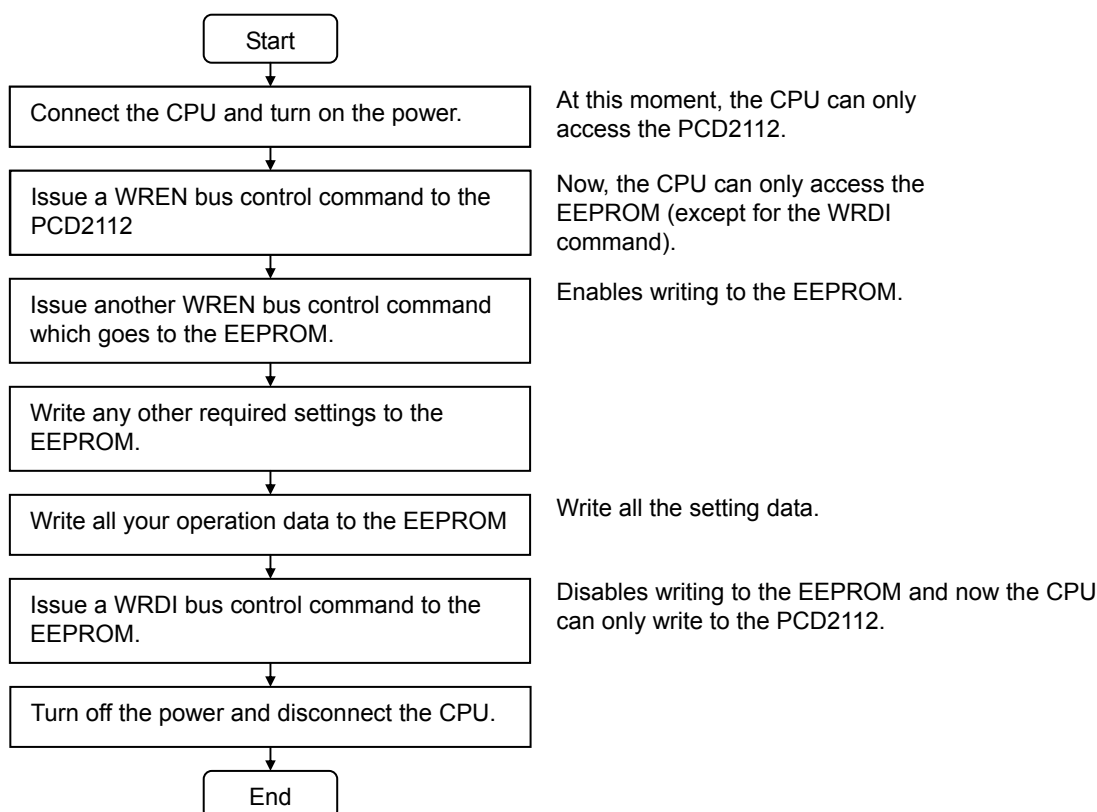
Item	Description						
Access method	Any method that supports SPI mode 0.						
Capacity	16 K-bits or larger (2 K bytes or more)						
Maximum serial clock frequency	1/2 or more of the reference clock (CLK) frequency						
Addressing	Write addresses with 16 bits						
Data size	In units of 8 bits						
Write prohibited mode	Has a write-prohibited mode and the write-prohibited mode can be selected by writing the following bus control command.						
Note	<table border="1"> <thead> <tr> <th>Symbol</th> <th>Command style</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>WRDI</td> <td>0000 0100</td> <td>Enable the write prohibited mode</td> </tr> </tbody> </table>	Symbol	Command style	Description	WRDI	0000 0100	Enable the write prohibited mode
Symbol	Command style	Description					
WRDI	0000 0100	Enable the write prohibited mode					

Note: The PCD2112 can handle either the 3.3VDC or 5VDC power used by EEPROMs. However, to interface it with the CPU, the PCD2112 may fail to communicate if different voltages are used to power the CPU and EEPROM.

### 11-2. Writing data to the EEPROM

#### 11-2-1. Data writing procedures

See the connection diagram in section 5-2, *Stand alone system (MODE=H): Common CS method*. By using the connections shown in item 1) of section 5-2, the PCD2112 will let the CPU write data to the EEPROM. For details about the control EEPROM procedures, see section 8-2-1, *Change to write allowed mode*. A rough idea of the operation is shown below.



11-2-2. EEPROM address map

You can store up to 32 sets of register settings in the EEPROM. (Patterns 0 to 31)  
 Store each pattern in its own a 64 byte area.

Address:	EEPROM	Address	Offset
0000 h	Pattern 0	RENV2 (31 to 24)	00 h 00 h
0040 h	Pattern 1	RENV2 (23 to 16)	01 h 01 h
0080 h	Pattern 2	RENV2 (15 to 08)	02 h 02 h
	⋮	RENV2 (07 to 00)	03 h 03 h
	⋮	RENV1 (31 to 24)	04 h 04 h
	⋮	RENV1 (23 to 16)	05 h 05 h
	⋮	RENV1 (15 to 08)	06 h 06 h
	⋮	RENV1 (07 to 00)	07 h 07 h
	⋮	xxRDS (15 to 08)	08 h 08 h
	⋮	xxRDS (07 to 00)	09 h 09 h
	⋮	xxRUS (15 to 08)	0A h 0A h
	⋮	⋮	⋮
	⋮	⋮	⋮
0780 h	Pattern 30	RMD (15 to 08)	1F h 1F h
07C0 h	Pattern 31	RMD (07 to 00)	20 h 20 h
0800 h	0000 0000	RCOM (07 to 00)	21 h 21 h
	⋮	0000 0000	22 h 22 h
	⋮	⋮	⋮
	⋮	⋮	⋮
	⋮	0000 0000	3F h 3F h
	⋮	RENV2 (31 to 24)	40 h 00 h
	⋮	RENV2 (23 to 16)	41 h 01 h
	⋮	⋮	⋮
	⋮	⋮	⋮
	⋮	0000 0000	7F h 3F h
	⋮	RENV2 (31 to 24)	80 h 00 h
	⋮	⋮	⋮
	⋮	⋮	⋮

Note: These are same addresses as shown in section 4-1, *Detailed memory map*. However, the addresses must increase by 4 bytes for each pattern. Set RENV1 to RCOM as a single operation pattern. Store 0s in unused areas.

## 12. Characteristics

### 12-1. Absolute maximum ratings

Item	Abbreviation	Rating	Unit	Remark
Supply voltage	$V_{DD}$	-0.5 to +4.6	V	
Input voltage	$V_{IN}$	-0.5 to +6.6	V	$V_{IN} < V_{DD} + 3.0V$
Output current	$I_{OL}$	30	mA	For the #BSY/#END, ERC/CDW, OUT/PH1, DIR/PH2, P2/PH3, and P3/PH4 terminals
		20	mA	For terminals other than those above.
Storage temperature	$T_{stg}$	-65 to +150	C	

### 12-2. Recommended operating conditions

Item	Abbreviation	Rating	Unit	Remarks
Supply voltage	$V_{DD}$	+3.0 to +3.6	V	
Ambient temperature	$T_A$	-40 to +85	C	$T_J = -40$ to + 125

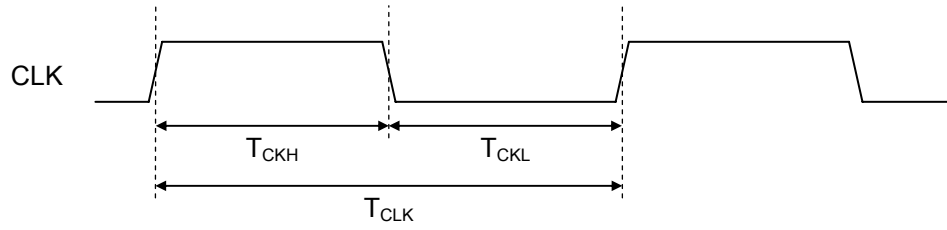
### 12-3. DC characteristics

Item	Abbreviation	Condition	Min	Max	Unit
Current consumption	$I_{dd}$	CLK = 9.8304MHz, no load		16	mA
		CLK = 20.000MHz, no load		31	mA
Input capacitance				9	pF
LOW input current	$I_{IL}$	$V_I = GND$		$\pm 1.0$	$\mu A$
HIGH input current	$I_{IH}$	$V_I = V_{DD}$	28	190	$\mu A$
LOW input voltage	$V_{IL}$		0	0.8	V
HIGH input voltage	$V_{IH}$	CLK, SCK, #CS, DI, DO terminals	2.4	5.5	V
		Other terminals	2.0	5.5	V
LOW output voltage	$V_{OL}$	$I_{OL} = 0mA$		0.1	V
HIGH output voltage	$V_{OH}$	$I_{OH} = 0mA$	$V_{DD} - 0.2$		V
LOW output voltage	$I_{OL}$	#BSY/#END, ERC/CDW, OUT/PH1, DIR/PH2, P2/PH3, P3/PH4 terminals ( $V_{OL} = 0.4 V$ )		9.00	mA
		Other terminal than the above ( $V_{OL} = 0.4 V$ )		6.00	mA
HIGH output voltage	$I_{OH}$	$V_{OH} = 2.4 V$	-3.00		mA
Input fall time	$T_r$		0	200	ns
Input rise time	$T_f$		0	200	ns
Internal pull down resistance	$R_{PD}$	$V_I = V_{DD}$	18.9	107.1	k $\Omega$

12-4. AC characteristics

12-4-1. Reference clock

Item	Abbreviation	Conditions	Min	Max	Unit
Reference clock frequency	$F_{CLK}$			20.5	MHz
Reference clock cycle	$T_{CLK}$		48.8		ns
Reference clock HIGH period	$T_{CKH}$		20		ns
Reference clock LOW period	$T_{CKL}$		20		ns

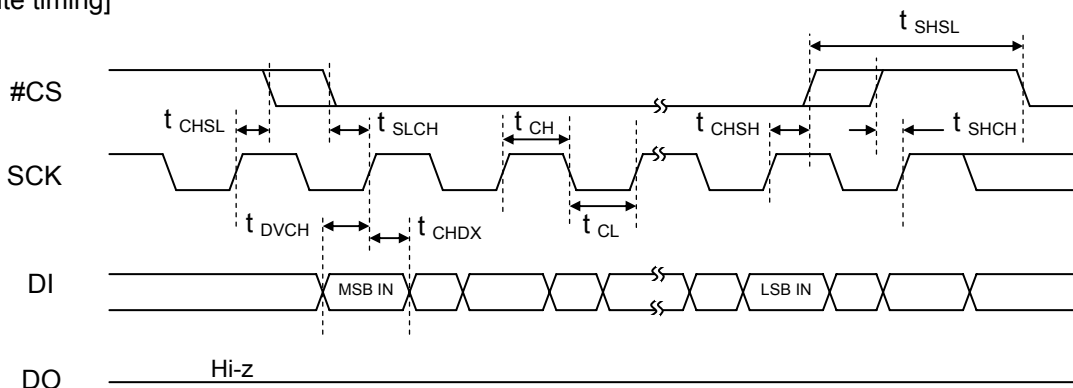




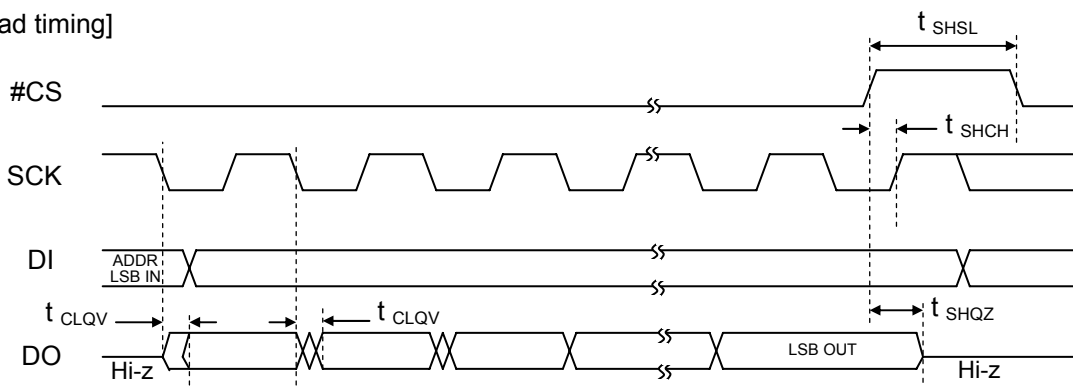
12-4-2. Serial bus I/F

Item	Abbreviation	Condition	Min	Max	Unit
Serial clock	$f_C$			$0.5 \cdot F_{clk}$	MHz
Serial clock (SCK) frequency	$t_C$		$2 \cdot T_{clk}$		MHz
Serial clock HIGH time	$t_{CH}$		$T_{clk}$		ns
Serial clock time	$t_{CL}$		$T_{clk}$		ns
#CS active set up	$t_{SLCH}$		10		ns
#CS inactive set up	$t_{SHCH}$		10		ns
#CS deselect time	$t_{SHSL}$		$t_C$		ns
#CS active hold time	$t_{CHSH}$		10		ns
#CS inactive hold	$t_{CHSL}$		10		ns
Data set up time	$t_{DVCH}$		10		ns
Data hold time	$t_{CHDX}$		10		ns
Output disable time	$t_{SHQZ}$	$C_L = 40pF$		7	ns
Output delay time	$t_{CLQV}$	$C_L = 40pF$		14	ns
#CSO fall delay time	$t_{CSOF}$	$C_L = 40pF$		11	ns
#CSO rise delay time	$t_{CSOR}$	$C_L = 40pF$		11	ns

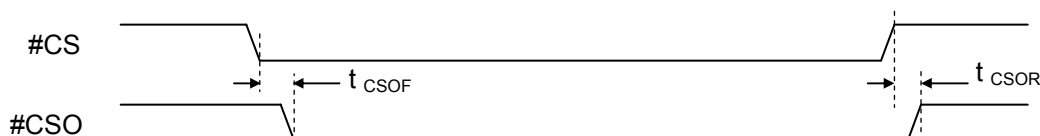
[Write timing]



[Read timing]



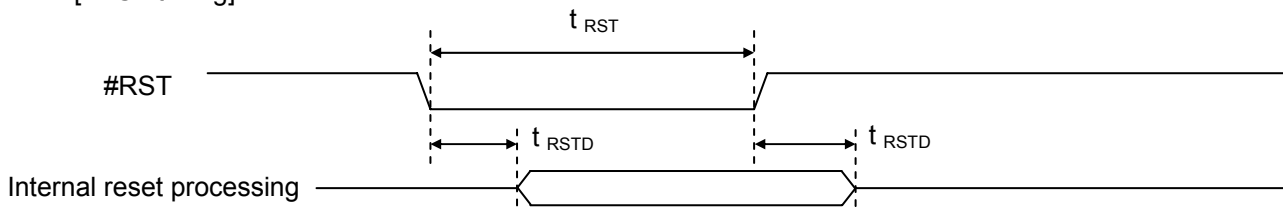
[CSO EEPROM timing] (in the EEPROM access mode)



## 12-5. Operation timing

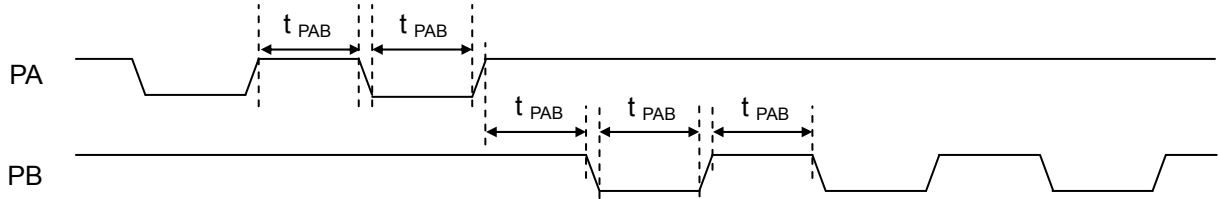
Item	Abbreviation	Condition	Min	Max	Unit
#RST input signal length	$t_{RST}$		$8 \cdot Tclk$		ns
Reset process delay time	$t_{RSTD}$		$8 \cdot Tclk$	$9 \cdot Tclk$	ns
PA, PB input signal length	$t_{PAB}$	RENV2.PINF=0	$1 \cdot Tclk$		ns
		RENV2.PINF=1	$3 \cdot Tclk$		
PDR, MDR input signal length	$t_{DR}$	RENV2.PINF=0	$131072 \cdot Tclk$		ns
		RENV2.PINF=1	$393216 \cdot Tclk$		
EA, EB, EZ input signal length	$t_{EAB}$	RENV2.EINF=0	$1 \cdot Tclk$		ns
		RENV2.EINF=1	$3 \cdot Tclk$		
ORG input signal length	$t_{ORG}$	RENV1.FLTR=0	$1 \cdot Tclk$		ns
		RENV1.FLTR=1	$40 \cdot Tclk$		
SD input signal length	$t_{SD}$	RENV1.FLTR=0	$1 \cdot Tclk$		ns
		RENV1.FLTR=1	$40 \cdot Tclk$		
PEL, MEL input signal length	$t_{EL}$	RENV1.FLTR=0	$1 \cdot Tclk$		ns
		RENV1.FLTR=1	$40 \cdot Tclk$		
#STP input signal length	$t_{STP}$	RENV2.STAF=0	$5 \cdot Tclk$		ns
		RENV2.STAF=1	$393218 \cdot Tclk$		
#STA input signal length	$t_{STA}$	RENV2.STAF=0	$5 \cdot Tclk$		ns
		RENV2.STAF=1	$393218 \cdot Tclk$		
INP input signal length	$t_{INP}$	RENV1.FLTR=0	$1 \cdot Tclk$		ns
		RENV1.FLTR=1	$40 \cdot Tclk$		
ERC output signal length		RENV1.EPW= 000	$125 \cdot Tclk$	$126 \cdot Tclk$	ns
		RENV1.EPW= 001	$992 \cdot Tclk$	$1008 \cdot Tclk$	
		RENV1.EPW= 010	$3968 \cdot Tclk$	$4032 \cdot Tclk$	
		RENV1.EPW= 011	$15872 \cdot Tclk$	$16128 \cdot Tclk$	
		RENV1.EPW= 100	$126976 \cdot Tclk$	$129024 \cdot Tclk$	
		RENV1.EPW= 101	$507904 \cdot Tclk$	$516096 \cdot Tclk$	
		RENV1.EPW= 110	$1015808 \cdot Tclk$	$1032192 \cdot Tclk$	
ERC signal OFF timer time		RENV1.ETW= 001	$125 \cdot Tclk$	$126 \cdot Tclk$	ns
		RENV1.ETW= 010	$15872 \cdot Tclk$	$16128 \cdot Tclk$	
		RENV1.ETW= 011	$507904 \cdot Tclk$	$516096 \cdot Tclk$	
		RENV1.ETW= 100	$1015808 \cdot Tclk$	$1032192 \cdot Tclk$	
		RENV1.ETW= 101	$2031616 \cdot Tclk$	$2064384 \cdot Tclk$	
		RENV1.ETW= 110	$4063232 \cdot Tclk$	$4128768 \cdot Tclk$	
		RENV1.ETW= 111	$8126464 \cdot Tclk$	$8257536 \cdot Tclk$	
#BSY output ON delay time	$t_{BSYF1}$	Command start	$5 \cdot Tclk$	$6 \cdot Tclk$	ns
	$t_{BSYF2}$	#STA start	$7 \cdot Tclk$	$8 \cdot Tclk$	
#BSY output OFF delay time	$t_{BSYR}$		$2 \cdot Tclk$		ns
Start delay time			$12 \cdot Tclk$		ns
SCK cycle in stand alone operation	$t_{MC}$		$2 \cdot Tclk$		ns
CSO delay time in stand alone operation	$t_{MCSO}$		$6 \cdot Tclk$	$7 \cdot Tclk$	ns
Download time in stand alone operation	$t_{MDLD}$		$594 \cdot Tclk$		ns

[#RST timing]

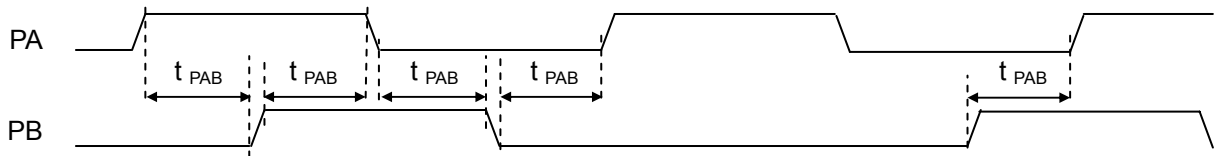


Note: Supply 8 clock cycles within the period that #RST is LOW

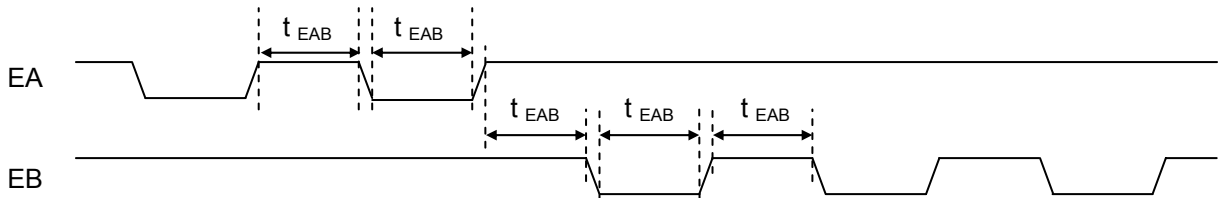
[PA, PB input timing (2-pulse mode)]



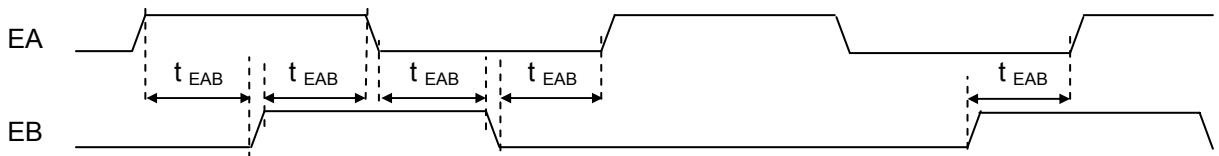
[PA, PB input timing (90 phase difference mode)]



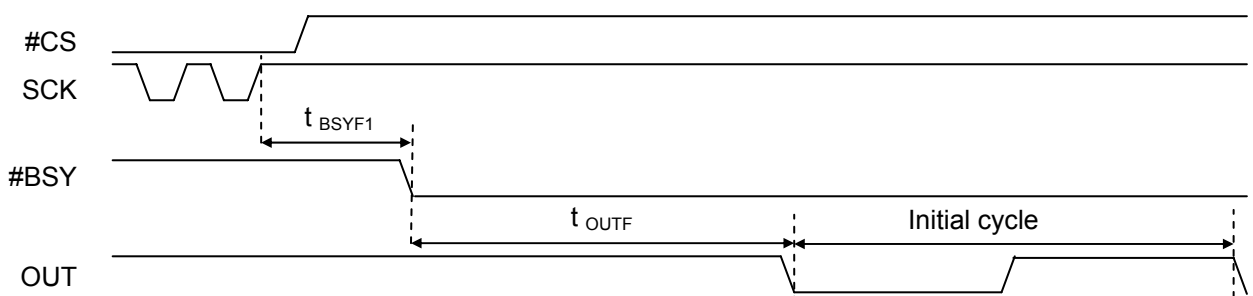
[EA, EB input timing (2-pulse mode)]



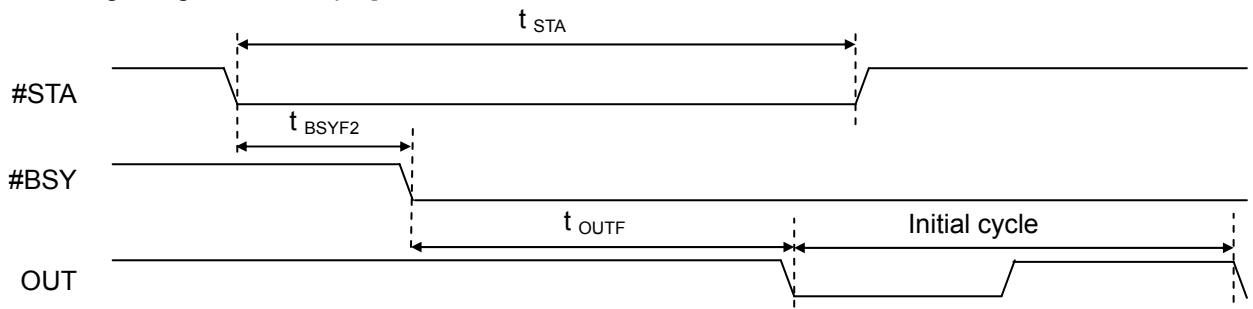
[EA, EB input timing (90 phase difference mode)]



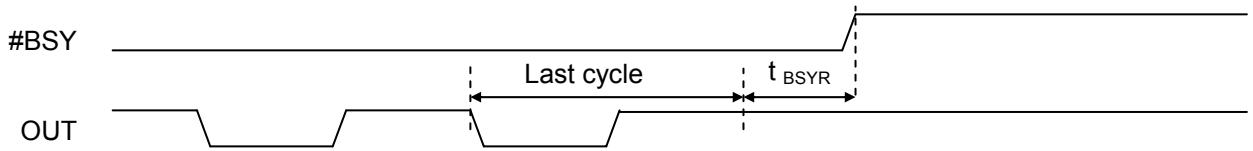
[Start timing using a start command]



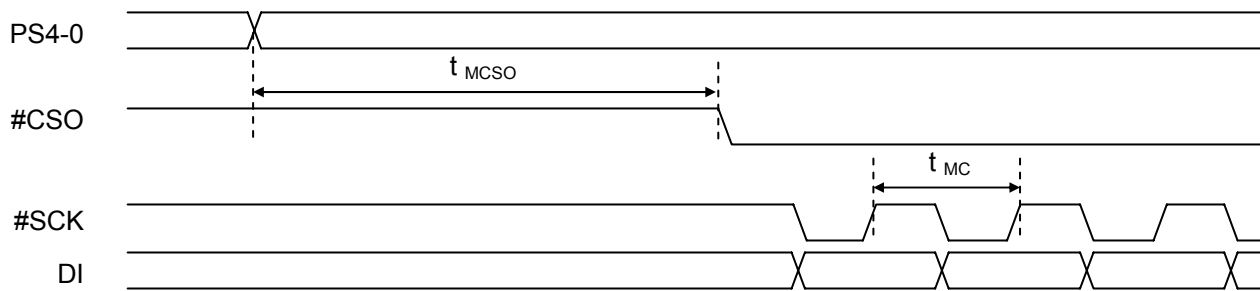
[Start timing using the #STA input]



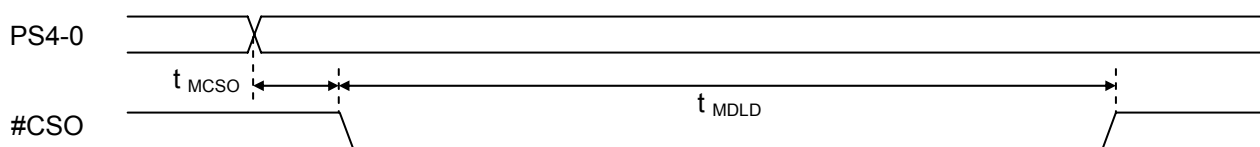
[Operation complete timing]



[Download start timing in stand alone operation]

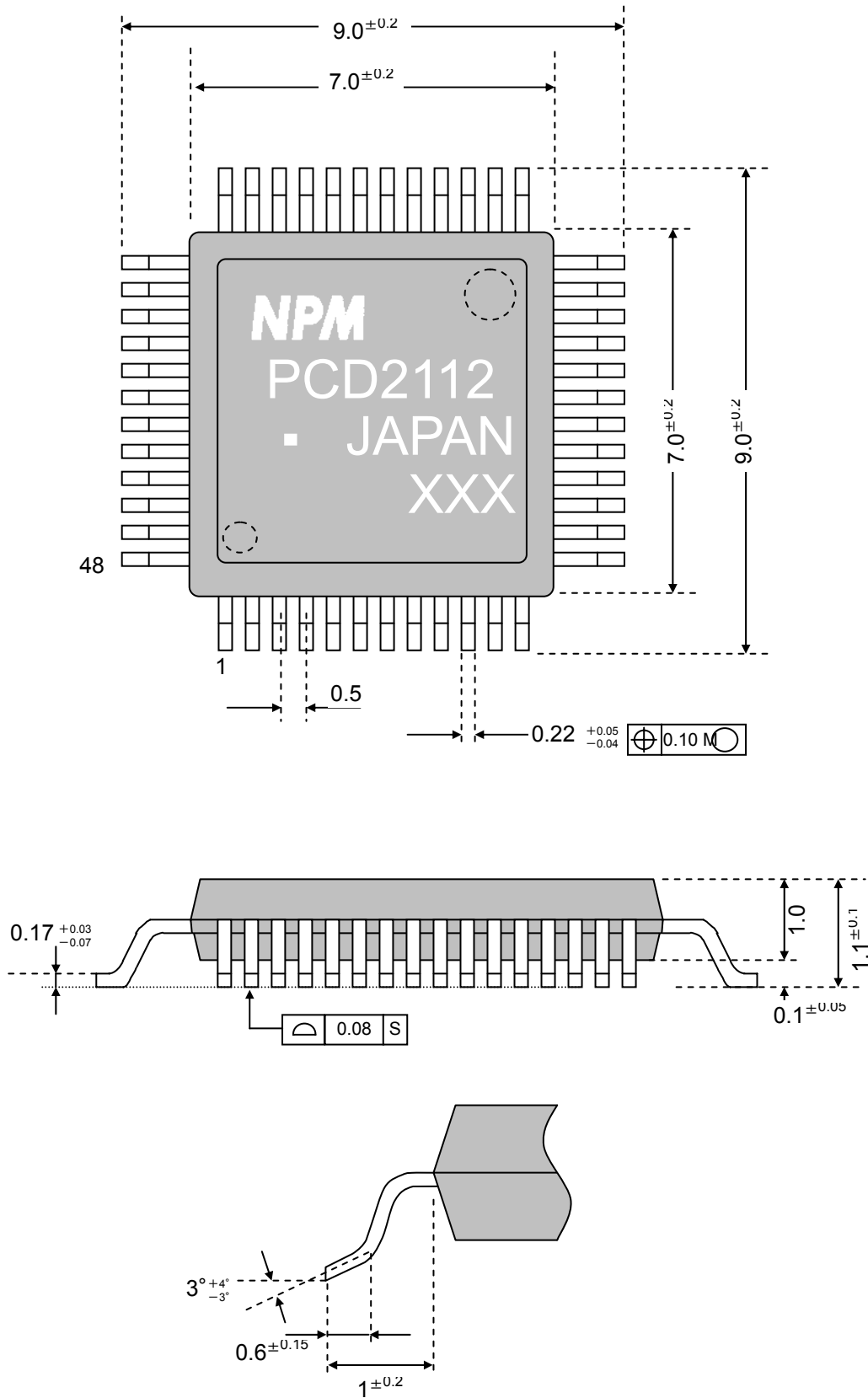


[Download time in stand alone operation]



13. External Dimensions

(Plastic TQFP 48pin) Unit: mm



## 14. Handling Precautions

### 14-1. Design precautions

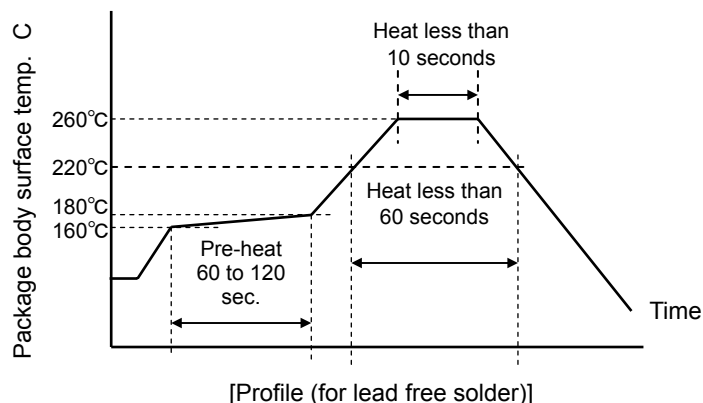
- 1) Never exceed the absolute maximum ratings, even for a very short time.
- 2) Take precautions against the influence of heat in the environment, and keep the temperature around the LSI as cool as possible.
- 3) Please note that ignoring the following may result in latching up and may cause overheating and smoke.
  - Do not apply a voltage greater than +3.3 V to the input/output terminals (greater than +5 V in the case of 5 V interface terminals) and do not pull them below GND.
  - Be careful about the power supply timing.
  - Be careful not to bring any external noise into the LSI.
  - Hold the unused input terminals to +3.3 V or to GND.
  - 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 over voltages caused by noise, voltage surges, or static electricity are not fed to the LSI.

### 14-2. Precautions for transporting and storing LSIs

- 1) Always handle the 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 or direct sunlight.
- 3) Do not store the LSI in a location where harmful gases (corrosive gases, etc.) are present, or in excessively dusty environments.
- 4) Store unused LSIs in an anti-static storage container, and make sure that no physical load is placed on the LSIs.

### 14-3. Precautions for installation

- 1) Plastic packages absorb moisture easily, and they will become damp over time, even when stored in a room.  
When damp LSIs are put in a reflow furnace, cracks may occur in the resin and the adhesive between the resin and frame may deteriorate.  
The storage period before unpacking the chips from the moisture prevention bag is one year.
- 2) If moisture is probable, dry the packages thoroughly before any reflow operation.  
Dry the packages 10 to 72 hours at 125 C.  
Only dry them one time.  
Basically, if an LSI is left for 3 days after being unpacked from its moisture prevention bag, it must be dried.
- 3) If the LSI will be soldered by heating the whole LSI package, such as with an infrared reflow system, operate within the range specified below, and limit the number of reflow events to 2.
  - Temperature profile: The temperature profile of the IR reflow furnace must be within the range shown in the figure below.
  - We recommend that the chlorine content (mass rate %) of the rosin family flux should be 0.2% or less.



- 4) Soldering using the solder dipping method may cause rapid temperature changes in the package and may cause damage to the device. Do not use this method with these LSIs.

----- MEMO -----

First version issued June 18, 2007  
Second version issued January 31, 2008  
Third version issued October 5, 2009