Motionnet

RemoteI/O & RemoteMotion

G9003

(PCL device)

User's Manual

NPM Nippon Pulse Motor Co., Ltd.

[Preface]

Thank you for considering our super high-speed serial communicator LSI, the "G9000."
To learn how to use the G9000, read this manual to become familiar with the product.
The handling precautions for installing this LSI are described at the end of this manual. Make sure to read them before installing the LSI.

[What the Motionnet is]

As a next generation communication system, the Motionnet can construct faster, more volume large scale, wire saving systems than conventional T-NET systems (conventional LSI product to construct serial communication system by NPM). Further, it has data communication function, which the T-NET does not have, so that the Motionnet can control data control devices such as in the PCL series (pulse train generation LSI made by NPM).

The Motionnet system consists of one center device connected to a CPU bus, and maximum 64 local devices, and they are connected by using cables of two or three conductive cores.

[Cautions]

(1) Copying all or any part of this manual without written approval is prohibited.

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

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

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

[Descriptions of indicators]

(1) When describing register bits, "n" refers to the bit position and "0" refers to a bit position that can only be written with a "0." It also means the bit will always be read as "0"

(2) Unless otherwise described, the timing for clocks discussed in this manual is a CLK speed of 40 MHz.

(3) Terminal names and signal names that start with a # use negative logic.
Ex.: #CS means that the CS terminal uses negative logic. This has the same meaning as $\overline{\text{CS}}$.

INDEX

1. Outline

This LSI is an axis control device for the Motionnet system. On receiving a command from the center device (G9001A), it can output high-speed pulses to drive stepper motors and servomotors.
Using a variety of speed patterns, including constant speed, linear acceleration/deceleration, and S-curve acceleration/deceleration, this device affords control of various actions including continuous feeding, positioning, and origin return operations.
If all of the devices connected to a center device are PCL devices (G9003), the system can be constructed to control up to 64 axes while reducing the needed wiring. Using routine communications, the system allows you to check the operation status and interrupt it with various conditions.

2. Features

- Communications
  Maximum transfer speed is 20 Mbps.
  The system can control up to 64 axes.
  If a communication error occurs, it can stop outputting pulses and reset the output ports.

- Power supply
  Single power supply voltage: +3.3 V.
  Interfaces with 5V ICs are possible (except for clock input and communication related terminals).

- Interrupt signal output
  An interrupt request can be output to the center device by various factors.

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

- Speed override
  The feed speed can be changed in the middle of any feed operation.

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

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

- Simultaneous start function
  Multiple axes controlled by the same LSI can be started at the same time using an external signal.

- Simultaneous stop function
  Multiple axes controlled by the same LSI can be stopped at the same time using an external signal.

- Excitation sequence for 2-phase stepper motors

This device can output excitation sequences for 2-phase stepper motors, for both unipolar and bipolar systems.

- A variety of counter circuits

The following four counters are available separately for each axis.

Counter	Use or purpose	Counter Input/Output
COUNTER1	28-bit counter for control of the command position	Outputs pulses
COUNTER2	28-bit counter for mechanical position control (Can be used as general-purpose counter)	Outputs pulses EA/EB input PA/PB input
COUNTER3	16-bit counter for controlling the deviation between the command position and the machine's current position, or 16-bit general-use counter with the synchronous signal output function.	Outputs pulses EA/EB input PA/PB input 1/4096 of reference clock Outputs pulses and EA/EB input Outputs pulses and PA/PB input EA/EB input and PA/PB input

All counters can be reset by writing a command or by providing a CLR signal.

The counter data can also be latched by writing a command, or by providing an LTC or ORG signal.

COUNTER3 counters can be used as a ring counter that repeats counting through a specified counting range by using IDX (synchronous) signal output function.

- Comparator

There are three comparator circuits for each axis. They can be used to compare target values and internal counter values.

The counter to compare can be selected from COUNTER1 (command position counter), COUNTER2 (mechanical position counter), COUNTER3 (deflection counter).

Comparators 1 and 2 can also be used as software limits (+SL, -SL).

- Software limit function

You can set software limits using 2 comparators' circuits.

When the mechanical position approaches the software limit range, the LSI will instruct the motors to stop immediately or to stop by deceleration. After that these axes can only be moved in the direction opposite their previous travel.

- Backlash correction function

The LSI has a backlash correction function.

Each time the feed direction is changed, the LSI applies a backlash correction

- Synchronous signal output function

The LSI can output pulse signals at the specified intervals.

- Vibration restriction function

Specify a control constant in advance and add one pulse each for reverse and forward feed just before stopping.

Using this function, vibration can be decreased while stopping.

- Manual pulsar input function

By applying manual pulse signals (PA/PB), you can rotate a motor directly.

The input signals can be 90 phase difference signals (1x, 2x, or 4x) or up and down signals.

In addition to the magnification rates above, the LSI contains an integral pulse number magnification circuit which multiplies by 1x to 32x and a pulse quantity division circuit which is divided by 2048 (1/2048 to 2048/2048).

EL signal and software limit settings can be used, and the LSI stops the output of pulses. It can also feed in the opposite direction.

- Out-of-step detection function

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

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

- Output pulse specifications

  Output pulses can be set to a Common pulse or Two-pulse mode. The output logic can also be selected.

- Idling pulse output function

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

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

- Operation mode

  The basic operations of this LSI consists of continuous operation, positioning, and origin return.

  By setting the optional operation mode bits, you can use a variety of operations.

  <Examples of the operation modes>

  1) Start/stop by command.
  2) Continuous operation and positioning operation using PA/PB inputs (manual pulsar).
  3) Origin return operation.
  4) Positioning operation using commands.
  5) Hardware start of the positioning operation using #STA input.
  6) Change the target position after turning ON the PCS. (Delay control)

- Variety of origin return sequences

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

- Mechanical input signals

  The following four signals can be input.

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

  For safety, make sure the +EL and -EL signals stay on from the EL position until the end of each stroke. The input logic for these signals can be changed using the ELL terminal.

  The input logic of the SD and ORG signals can be changed using software.

- Digital servomotor I/F

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

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

  The input logic of the INP, ERC, and ALM signals can be changed using software.

  The ERC signal is a pulsed output. The pulse length can be set. (12 μsec to 104 msec. A level output is also available.)

- Emergency stop signal (#EMG) input

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

3. Specifications

3-1. Device specifications

Item	Description
General-purpose input/output	General-purpose input/output ports. (1 port = 8 bits) Individual bits can be set for input or output.
Communication data length	1 to 4 words / frame (1 word = 16 bits)
Data buffer length	4 words
Data communication time	When using 3-word communication (written to one register in the PCL device(G9003)) --- 19.3 µs
Transfer system	I/O ports: Cyclic transfer Data communication: Transient transfer.
Package type	80 pins QFP (Mold section: 12 x 12 x 1.4 mm)
Power supply	3.3 V ±10%
Storage temperature range	-40 to +125 C
Operating temperature range	-40 to +85 C

3-2. Communication system specifications

Item	Description
Reference clock Note 1	40 MHz or 80 MHz
Communication speed Note 2	2.5 M, 5 M, 10 M, or 20 Mbps
Communication sign	NRZ sign
Communication protocol	NPM original method
Communication method	Half-duplex communication
Communication I/F Note 3	RS-485 or pulse transformer
Connection method	Multi-drop connection
Number of local devices	64 devices max.

Note 1: When to transfer data with 20 Mbps speed, and if the clock duty can be maintained to ideal "50:50" condition, the center device can be operated by inputting 40 MHz clock signal.
The above ideal conditions mean that an oscillator and the PCL device (G9003) are connected as 1:1 and close to each other. Actually, even these good conditions cannot establish 50:50. However, a duty proximate to the ideal one will be established.
Even if the ideal duty is broken a little, when signal lines are shorter and/or the number of local devices is smaller, the center device can operate without any trouble. (For the details, see the section for the "CLK" terminal.)
When the signal lines are longer and/or the number of connected local devices is high and if it is difficult to warranty the clock duty, you should take measures such as to prepare an 80 MHz signal or a 40 MHz clock proprietary to the PCL device (G9003).
To select a clock rate, specify using the LSI terminal. In either clock rate, the maximum speed of 20 Mbps is the same.

Note 2: Select the communication speed using the LSI terminal. Regardless of the selection of the communication speed, the input clock remains the same.

Note 3: NPM recommends a system that uses a pulse transformer.

3-3. Specifications for the axis control section

Item	Description
Positioning control range	-134,217,728 to +134,217,727 (28-bit)
Ramping-down point setting range	0 to 16,777,215 (24-bit)
Number of registers used for setting speeds	Three for each axis (FL, FH, and FA (speed correction))
Speed setting step range	1 to 100,000 (17-bits) Note1
Speed magnification range	Multiply by 0.1 to 66.6 Multiply by 0.1 = 0.1 to 10,000.0 pps Multiply by 1 = 1 to 100,000 pps Multiply by 50 = 50 to 5,000,000 pps
Acceleration/deceleration characteristics	Selectable acceleration/deceleration pattern for both increasing and decreasing speed separately, using Linear and S-curve acceleration/deceleration.
Acceleration rate setting range	1 to 65,535 (16-bit) Ex: 1→ 100,000 pps acceleration time: 80 msec (set 1) to 2621 sec (set 65535)
Deceleration rate setting range	1 to 65,535 (16-bit)
Ramping-down point automatic setting	Automatic setting within the range of (deceleration time) < (acceleration time x 2)
FH correction function (Eliminates triangle pattern driving)	If the feed amount is too small, the PCL device (G9003) has to start decelerating before it has completed the acceleration, and this will create a triangular shaped speed pattern. In order to eliminate this triangular speed pattern, this function automatically reduces the operation speed so that the triangle speed pattern will be avoided.
Manual operation input	Manual pulsar input
Counter	COUNTER1: Command position counter (28-bit) COUNTER2: Mechanical position counter (28-bit) COUNTER3: Deflection counter (16-bit)
Comparators	28-bits x 3 circuits

Note 1: Values above 100,000 cannot be entered. Even if a value over 100,000 is entered, the register value will only be 100,000.

4. Hardware description

4-1. A list of terminals

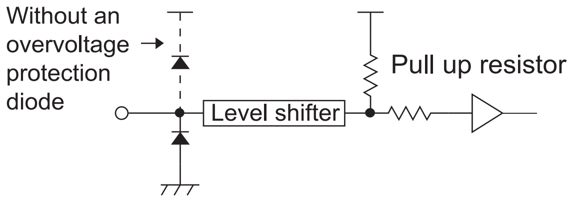
No.	Signal name	I/O	Logic	Description	5V interface
1	VDD			Power supply +3.3 V	
2	#DN0	I_U	Negative	Device number bit 0 (Common with the serial input)	Possible
3	#DN1	I_U	Negative	Device number bit 1	Possible
4	#DN2	I_U	Negative	Device number bit 2	Possible
5	#DN3	I_U	Negative	Device number bit 3	Possible
6	#DN4	I_U	Negative	Device number bit 4	Possible
7	#DN5	I_U	Negative	Device number bit 5	Possible
8	VDD			Power supply input +3.3 V	
9	VDD			Power supply input +3.3 V	
10	VDD			Power supply input +3.3 V	
11	VDD			Power supply input +3.3 V	
12	GND			GND	
13	#DNSO	O	Negative	Serial output of the next chips device number	Possible
14	DNSM	I_U		Device number set mode	Possible
15	SOEI	I_D	Positive	Enable serial output	
16	#SOEL	O	Negative	Enable serial output	
17	SOEH	O	Positive	Enable serial output	
18	SO	O	Positive	Serial output	
19	GND			GND	
20	SI	I	Positive	Serial input	
21	GND			GND	
22	VDD			Power supply input +3.3V	
23	VDD			Power supply input +3.3V	
24	EA	I_U		Encoder A phase signal	Possible
25	EB	I_U		Encoder B phase signal	Possible
26	EZ	I_U	Negative %	Encoder Z phase signal	Possible
27	PA	I_U		Manual pulser A phase signal	Possible
28	PB	I_U		Manual pulser B phase signal	Possible
29	GND			GND	
30	ERC	O	Negative %	Request to clear a deflection counter in a driver	Possible
31	OUT	O	Negative %	Pulse train output	Possible
32	DIR	O		Feed direction	Possible
33	VDD			Power supply input +3.3 V	
34	#CP1	O	Negative	Comparator 1 output	Possible
35	#CP2	O	Negative	Comparator 2 output	Possible
36	#CP3	O	Negative	Comparator 3 output	Possible
37	PCS	I_U	Negative %	Start positioning control	Possible
38	LTC	I_U	Negative %	Counter value latch signal	Possible
39	CLR	I_U	Negative %	Counter clear signal	Possible
40	INP	I_U	Negative %	In-position (Positioning complete)	Possible
41	GND			GND	
42	#EMG	I_U	Negative	Emergency stop	Possible
43	+EL	I_U	Negative#	(+) end limit	Possible
44	-EL	I_U	Negative#	(-) end limit	Possible

No.	Signal name	I/O	Logic	Description	5V interface
45	SD	I_U	Negative %	Deceleration request signal	Possible
46	ORG	I_U	Negative %	Origin position signal	Possible
47	ALM	I_U	Negative %	Alarm signal (Stop request)	Possible
48	VDD			Power supply input +3.3 V	
49	#STA	B_U	Negative	External start	Possible
50	#STP	B_U	Negative	External stop	Possible
51	P0	B_U		General-purpose I/O terminal 0	Possible
52	GND			GND	
53	P1	B_U		General-purpose I/O terminal 1	Possible
54	P2	B_U		General-purpose I/O terminal 2	Possible
55	P3	B_U		General-purpose I/O terminal 3	Possible
56	P4	B_U		General-purpose I/O terminal 4	Possible
57	P5	B_U		General-purpose I/O terminal 5	Possible
58	P6	B_U		General-purpose I/O terminal 6	Possible
59	P7	B_U		General-purpose I/O terminal 7	Possible
60	VDD			Power supply input +3.3 V	
61	#BSY/PH1	O	Negative/ Positive	Operation-in-progress signal / Excitation sequence output 1	Possible
62	#FUP/PH2	O	Negative/ Positive	Acceleration monitor output / Excitation sequence output 2	Possible
63	#FDW/PH3	O	Negative/ Positive	Deceleration monitor output /Excitation sequence output 3	Possible
64	#MVC/PH4	O	Negative/ Positive	Constant speed monitor output /Excitation sequence output 4	Possible
65	ELL	I_U		Set ±EL input logic	Possible
66	GND			GND	
67	#MSEL	O	Negative	Goes LOW for a certain interval while this chip is sending/receiving data.	Possible
68	#MRER	O	Negative	Goes LOW for a certain interval when an abnormal communication has been received.	Possible
69	#TOUT	O	Negative	Watchdog timer output	Possible
70	BRK	I_D	Positive	Break frame send request	Possible
71	TUD	I_U		Select operation method for outputting watchdog timer signal	Possible
72	TMD	I_U		Watchdog timer setting	Possible
73	VDD			Power supply input +3.3 V	
74	CLK	I		Reference clock	
75	GND			GND	
76	CKSL	I_U		Select clock rate	Possible
77	SPD0	I_U		Communication speed setting 0	Possible
78	SPD1	I_U		Communication speed setting 1	Possible
79	GND			GND	
80	#RST	I_U	Negative	Reset	Possible

Note 1: "I" in the I/O column expresses input, "O" as output, and "B" as both directions.

Note 2: All the inputs, including bi-directional signals can be interfaced with 5 V lines. They can be connected to 5 V CMOS, 3.3 V CMOS, TTL, and LVTTL devices. All the outputs, except the outputs related to communications (including bi-directional signals), can be interfaced with 5 V lines. They can be connected to 3.3 V CMOS, TTL, and LVTTL devices. To connect a 5 V CMOS device, connect pull up resistors (5~10K ohms) to +5 V.

Note 3: Inputs that can be interfaced with 5 V lines are not equipped with an overvoltage prevention diode for the 3.3 V lines. If overvoltage is possible due to a reflection, linking, or to inductive noise, we recommend inserting a diode to protect against overvoltage.



Note 4: "I_U" and "B_U" in the table indicate terminals with a pull up resistor to prevent floating. "I_D" indicates terminals with a pull down resistor to prevent floating. The inputs that can be connected to 5 V lines are not connected directly to pull up/pull down resistors (a few 10K-ohms to a few 100 k-ohms). They are connected after a level shifter.

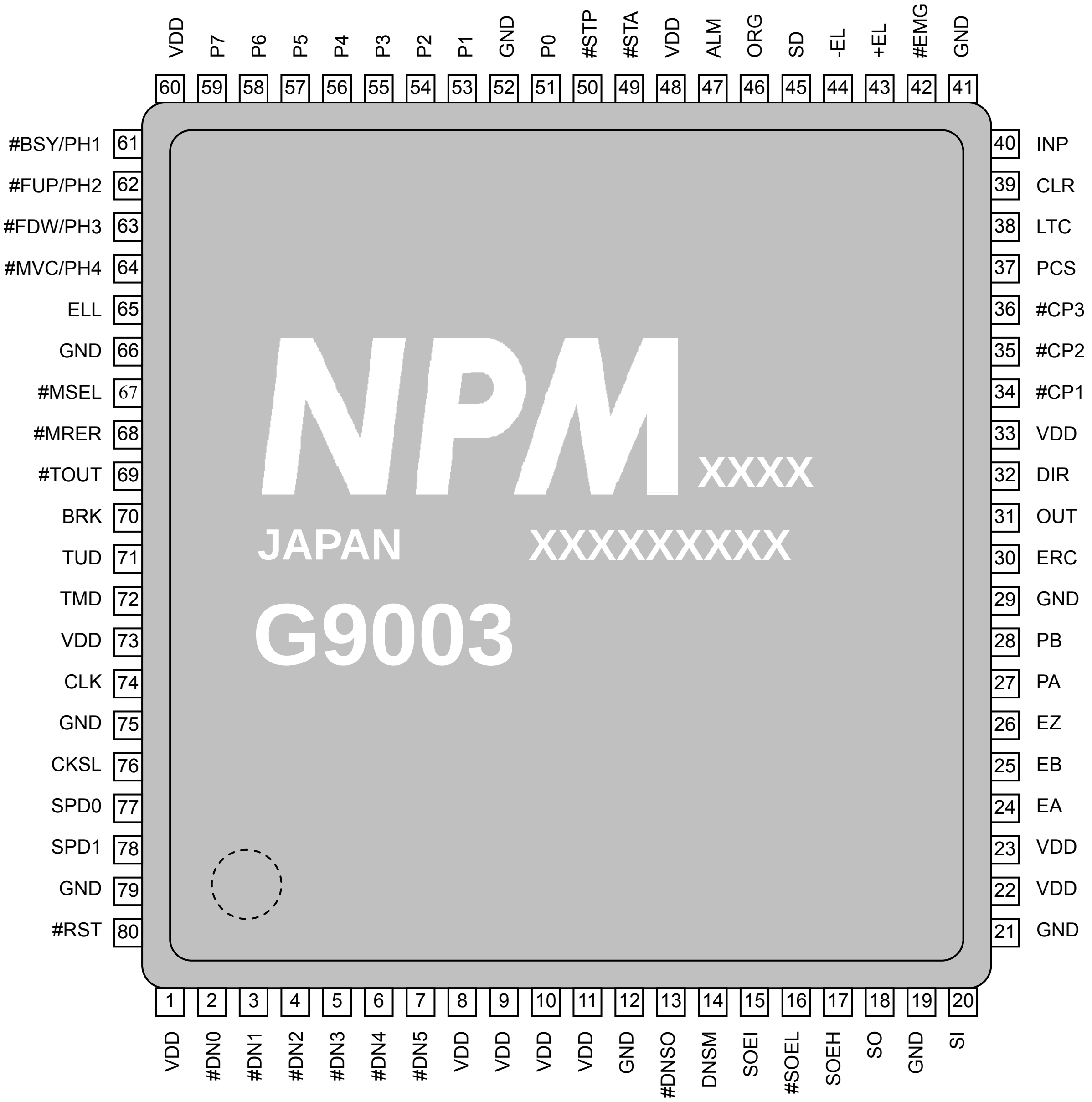
If you want to drive these terminals using an open collector, you must connect pull up resistors (5~10K-ohms) externally.

Input terminals that are not used and which have internal pull up/ pull down resistors can be left open. However, we recommend pulling these unused terminals up to 3.3 V externally, or connect them directly to the 3.3 V or GND terminals.

Note 5: Leave the unused output terminal's open.

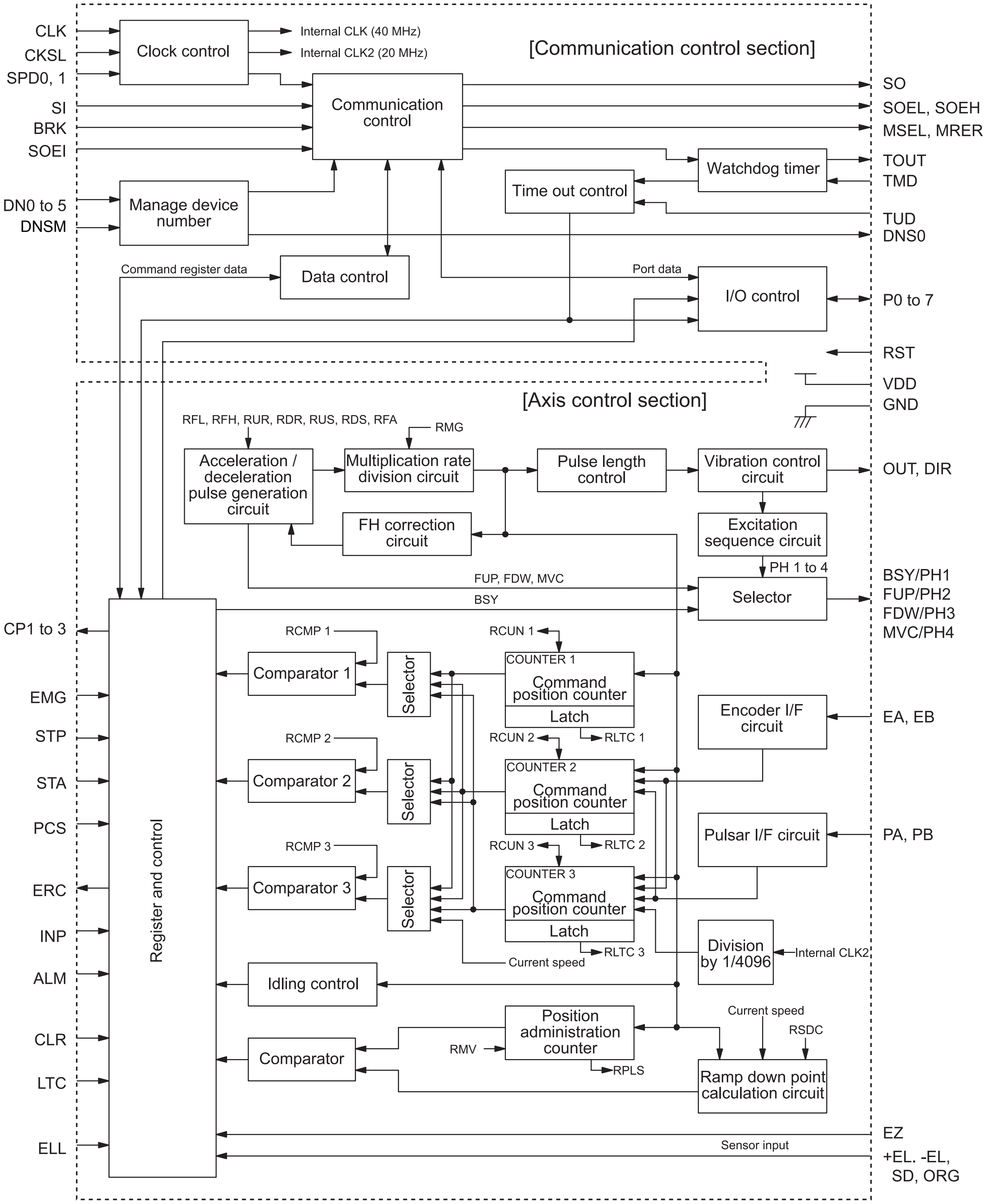
Note 6: "Negative" and "Positive" in the logic column mean negative logic and positive logic. In addition, a "#" means that the terminal's logic can be changed with software. A "%" means that the terminal's logic setting can be changed by another terminal. The logic shown in the table is the default condition. The DIR terminal logic shown is when it is used in Two-pulse mode.

4-2. Terminal allocation diagram



Note: As you can see in the figure above, pin number 1 is to the lower left of the LSI model name marked on the chip.

4-3. Entire block diagram



4-4. Functions of terminals

4-4-1. CLK

This is an input terminal of the reference clock. By setting the CKSL terminal, either of the following clock rate signals can be connected.

CKSL = L: 40 MHz
CKSL = H: 80 MHz

By selecting either of these clock rates, the serial communication transfer rate does not change. This clock rate selection affects communication precision.
For a small-scale serial communication and transfer rate below 10 Mbps, use of the center device with 40 MHz does not give any restriction.
With 20 Mbps transfer speed; however, a longer communication line or a large number of connected local devices may deteriorate communication precision due to collapse of signals on the circuit. This deterioration of communication quality can be corrected inside the LSI if the deterioration level is not much. In order to improve correction precision; however, evenness of the clock duty is required. In other words, if the duty is ideal (50:50), the capacity to correct collapse of the signals in the communication lines can be improved. On the contrary, if the duty is not ideal, the center device cannot cope with collapses of the communication line.
As a result, if the duty is close to ideal, the center device can be used with 40 MHz. When connecting more than one oscillator, the duty will not be ideal. In this case, select 80 MHz. The center device divides the frequency inside and creates 40 MHz frequency.
If you do not want to use 80 MHz frequency, you may prepare a separate 40 MHz oscillator for this LSI.
The CLK terminal cannot be connected to 5 V. Supply only a 3.3 V CMOS level signal to the CLK terminal

4-4-2. #RST

This is an input terminal for a reset signal.
By input L level signal, the center device is reset. As the center device synchronizes with a clock, arrange a circuit so that it does not disconnect the clock while resetting. Longer than 10 clock cycles is required during resetting.

4-4-3. CKSL

Use to select clock rate.
L: Connect 40 MHz clock frequency to the CLK terminal.
H: Connect 80 MHz clock frequency to the CLK terminal.
Select this when the duty of the 40 MHz clock collapses a lot.

4-4-4. #DN0 to #DN5

Input terminals for setting device address.
Since these terminals use negative logic, setting all the terminals to LOW calls up device address "3Fh."
There are two methods for entering a device address. Select the input method using the DNSM terminal.

4-4-5. DNSM

Select the input method for loading the device address.

1) When the DNSM = H
   Specify an address from 00h to 3Fh using the #DN0 to #DN5 terminals.

2) When the DNSM = L
   Input a #DNSO signal that is output by some other chip on the #DN0 terminal on this device. When using this input method, this chip has an address equal to the other chip's address plus one.
   When using this method, connect terminals #DN1 to #DN5 to GND.
   When two sequential sets of serial data match, the data is taken to be a device address.

4-4-6. #DNSO

The numeric equivalent of the address on #DN0 to #DN5 + 1 will be output after being converted into a serial bit stream.
Connect this output to another local device's #DN0 terminal (make all the other DNSM terminals of that local device LOW), so that other devices can get the address and pass it along to the next data-sending device.
Please note that the next address after "3Fh" (#DN(5:0) = " 000000") is "00h."
In the case that continuous address by #DNSO signal is set, it is necessary to have at least approximately 50 μs until the next address is confirmed.

4-4-7. SPD0, SPD1

Set the communication speed
All of the devices on the same communication line must be set to the same speed.

SPD1	SPD0	Communication speed
L	L	2.5 Mbps
L	H	5 Mbps
H	L	10 Mbps
H	H	20 Mbps

4-4-8. TUD

A watchdog timer is included on the chip to assist in administration of the communication status (see the "TMD" terminal section).
When the data transmission interval from a center device to this device exceeds the set time, the watchdog timer times out.
This terminal is used to set output conditions when the watchdog timer times out.
When TUD = HIGH --- The LSI keeps its current status.
When the TUD = LOW --- Reset I/O port output, and immediately stops pulse output (stop operation).

4-4-9. TMD

Specify the time for the watchdog timer.
The watchdog timer is used to administer the communication status.
When the interval between data packets sent from a center device is longer than the specified interval, the watchdog timer times out (the timer restarts its count at the end of each data packet received from a center device). The time out may occur because of a problem on the communication circuit, such as disconnection, or simply because the center device has stopped communicating.
The time used by the watchdog timer varies with communication speed selected.

TMD terminal	Watchdog timer setting			
	20 Mbps	10 Mbps	5 Mbps	2.5 Mbps
L	5 ms	10 ms	20 ms	40 ms
H	20 ms	40 ms	80 ms	160 ms

4-4-10. #TOUT

Once the watchdog timer has timed out, this terminal goes LOW.

4-4-11. SO

Serial output signal for communication. (Positive logic, tri-state output)

4-4-12. SOEH, #SOEL

Output enable signal for communication.
The difference between the SOEH and #SOEL is that the logic is inverted.
When sending, SOEH = HIGH and #SOEL = LOW.

4-4-13. SOEI

When using more than one PCL device, connect the SOEH signal of the other PCL device (G9003) to this terminal.
By being wire OR'ed with the output enable signal from this PCL device (G9003), the device outputs an enable signal to SOEH or #SOEL.

4-4-14. SI

Serial input signal for communication. (Positive logic)

4-4-15. #MRER

Monitor output used to check communication quality.
When the PCL device (G9003) receives an error frame such as a CRC error, this terminal goes LOW for exactly 128 CLK cycles (3.2 μs).
By timing this interval using a counter, you can check the quality of the communication.

4-4-16. #MSEL

Communication status monitor output.
When the PCL device (G9003) receives a frame intended for this device and everything is normal (when communication MFER is OFF), this terminal goes LOW for exactly 128 CLK cycles (3.2 μs).
This can be used to check the cyclic communication time.

4-4-17. BRK

By providing HIGH pulses that are longer than the specified interval, the PCL device (G9003) will be made to wait for a break frame.
When the PCL device receives a break frame send request from a center device, it immediately sends a break frame.
The break frame is 60 bits long.
A pulse at least 3200 μsec long is needed, in order to be seen as the BRK input pulse (positive logic).

4-4-18. P0 to P7

Using software, these terminals can be set to function as general-purpose input or output terminals.
These terminals have built-in pull up resistors to prevent floating. When not used, they can be left open.
However, if you want to improve the noise resistance of the chip, pull them up (5 to 10 K-ohms).

4-4-19. #STA, #STP

If you want to start multiple LSI devices simultaneously, connect the #STA terminals of all the LSI devices together.
If you want to stop multiple LSI devices simultaneously, connect the #STP terminals of all the LSI devices together.
These terminals have built in pull up resistors to prevent floating. When not used, they can be left open. However, if you want to improve the noise resistance of the chip, pull them up (5 to 10 K-ohms).

4-4-20. #EMG

This is the emergency stop input terminal.
While this is set LOW, the PCL device (G9003) prohibits operation. If this signal goes LOW while the motor is operating, the motor will stop immediately.
This terminal has a built in a pull up resistor to prevent floating. When not used, it can be left open. However, if you want to improve the noise resistance of the chip, pull it up (5 to 10 K-ohms), or connect it to VDD.

4-4-21. ELL

This terminal is used to set the input logic of the +EL and -EL signals. When this terminal is LOW, the respective signal is set for positive logic.

4-4-22. +EL, -EL

Provide the stroke end signals to these terminals. Their input logic can be changed using the ELLn terminals.
When this signal (for the feed direction) turns ON, the motor stops immediately, or decelerates and stops, depending on the conditions.
These terminals have built in pull up resistors to prevent floating. When not used, they can be left open. If you want to improve the noise resistance of the chip, pull them up (5 to 10 K-ohms) or connect them to VDD

4-4-23. SD

Input for the deceleration signal (decelerate and stop). Software can be used to change the input logic of this terminal.
This input has a latch function.
This terminal has a built in pull up resistor to prevent floating. When not used, it can be left open. However, if you want to improve the noise resistance of the chip, pull it up (5 to 10 K-ohms), or connect it to VDD.

4-4-24. ORG

Input for an origin return signal. Software can be used to change the input logic of this terminal.
This input has a latch function.
This terminal has a built in pull up resistor to prevent floating. When not used, it can be left open. However, if you want to improve the noise resistance of the chip, pull it up (5 to 10 K-ohms), or connect it to VDD.

4-4-25. ALM

Input for an alarm signal. Software can be used to change the input logic of this terminal.
When this signal turns ON, the motor stops immediately, or decelerates and stops, depending on the conditions.
This terminal has a built in pull up resistor to prevent floating. When not used, it can be left open. However, if you want to improve the noise resistance of the chip, pull it up (5 to 10 K-ohms), or connect it to VDD.

4-4-26. OUT, DIR

While the PCL device (G9003) is in the common pulse mode, it sends feed pulses from the OUT terminal, and supplies a direction signal from the DIR terminal.
While the PCL device (G9003) is in the Two-pulse mode, it outputs positive direction feed pulses from the OUT terminal, and negative direction feed pulses from the DIR terminal.

4-4-27. PA, PB

Used to operate the motor from external pulses, such as a manual pulsar.
90 phase difference signals or Two-pulses (up pulse and down pulse) can be supplied to these terminals. The 90 phase difference signals can be multiplied by 2 or by 4.
These terminals have built in pull up resistors to prevent floating. When not used, they can be left open. If you want to improve the noise resistance of the chip, pull them up (5 to 10 K-ohms) or connect them to VDD

4-4-28. EA, EB, EZ

Use these terminals to control the current position using an encoder.
90 phase difference signals or Two-pulses (up pulse and down pulse) can be input on these terminals. The 90 phase difference signals can be multiplied by 2 or by 4.
The EZ input is used for origin return operations. Software can be used to change the input logic of these terminals. These terminals have built in pull up resistors to prevent floating. When not used, they can be left open. If you want to improve the noise resistance of the chip, pull them up (5 to 10 K-ohms) or connect them to VDD.

4-4-29. PCS

The PCL device (G9003) starts positioning control. (Override 2 of the target position) when a signal is applied to this terminal.
Software can be used to change the input logic of this terminal.
This terminal has a built in pull up resistor to prevent floating. When not used, it can be left open. However, if you want to improve the noise resistance of the chip, pull it up (5 to 10 K-ohms), or connect it to VDD.

4-4-30. INP

Input for a positioning-complete signal from a servo driver.
The output of the INT can be delayed until this signal is input.
Software can be used to change the input logic of this terminal.
This terminal has a built in pull up resistor to prevent floating. When not used, it can be left open. However, if you want to improve the noise resistance of the chip, pull it up (5 to 10 K-ohms), or connect it to VDD.

4-4-31. CLR

Reset the specified counter (COUNTER1 to 3) by inputting a signal (can be used to reset more than one counter).
Software can be used to change the input logic of this terminal.
This terminal has a built in pull up resistor to prevent floating. When not used, it can be left open. However, if you want to improve the noise resistance of the chip, pull it up (5 to 10 K-ohms), or connect it to VDD.

4-4-32. LTC

Latch the specified counter (COUNTER1 to 3) by inputting a signal (can latch more than one counter).
Software can be used to change the input logic of this terminal.
This terminal has a built in pull up resistor to prevent floating. When not used, it can be left open.
However, if you want to improve the noise resistance of the chip, pull it up (5 to 10 K-ohms), or connect it to VDD.

4-4-33. ERC

Outputs a one-shot pulse to clear a deflection counter for a servo driver.
The output logic and pulse length can be set using software. (A level output is also possible.)
If this terminal is not used, leave it open.

4-4-34. #BSY/PH1

When #BSY is selected, the PCL device (G9003) outputs a LOW while the motor is operating.
When PH1 is selected, the PCL device (G9003) outputs an excitation sequence for a 2-phase stepper motor.
Select #BSY/PH1 using the RMD register.
If this terminal is not used, leave it open.

4-4-35. #FUP/PH2

When #FUP is selected, the PCL device (G9003) outputs a LOW while the motor is accelerating.
When PH2 is selected, the PCL device (G9003) outputs an excitation sequence for a 2-phase stepper motor.
Select #FUP/PH2 using the RMD register.
If this terminal is not used, leave it open.

4-4-36. #FDW/PH3

When #FDW is selected, the PCL device (G9003) outputs a LOW while the motor is decelerating.
When PH3 is selected, the PCL device (G9003) outputs an excitation sequence for a 2-phase stepper motor.
Select #FDW/PH3 using the RMD register.
If this terminal is not used, leave it open.

4-4-37. #MVC/PH4

When #MVC is selected, the PCL outputs a LOW while the motor is fed at a constant speed.
When PH4 is selected, the PCL outputs an excitation sequence for a 2-phase stepper motor.
Select #MVC/PH4 using the RMD register.
If this terminal is not used, leave it open.

4-4-38. #CP1

When the conditions for Comparator 1 are met, the PCL device (G9003) outputs a LOW on #CP1.
If this terminal is not used, leave it open.

4-4-39. #CP2

When the conditions for Comparator 2 are met, the PCL device (G9003) outputs a LOW on #CP2.
If this terminal is not used, leave it open.

4-4-40. #CP3

When the conditions for Comparator 3 are met, the PCL device (G9003) outputs a LOW on #CP3.
If this terminal is not used, leave it open.

5. Description of the software

5-1. Outline of control

5-1-1. Communication control

- The center device (G9001A) controls all the communication.
- One communication cycle consists of a communication from the center device to the local devices, and the communication from the local devices back to the center device.
- The response from the local devices may include I/O information and data.
- This PCL device (G9003) is a local device.

5-1-2. Communication type

System communications, cyclic communications, and data communications are the three communication types available.

1) System communications

With the system communications, the center device automatically confirms the connection status, device type, and I/O port settings of each local device.
By starting the system communication, the center device polls all of the local devices (device No. 0 to 63), one by one, and refreshes the "device information" area according to the response from the local devices.

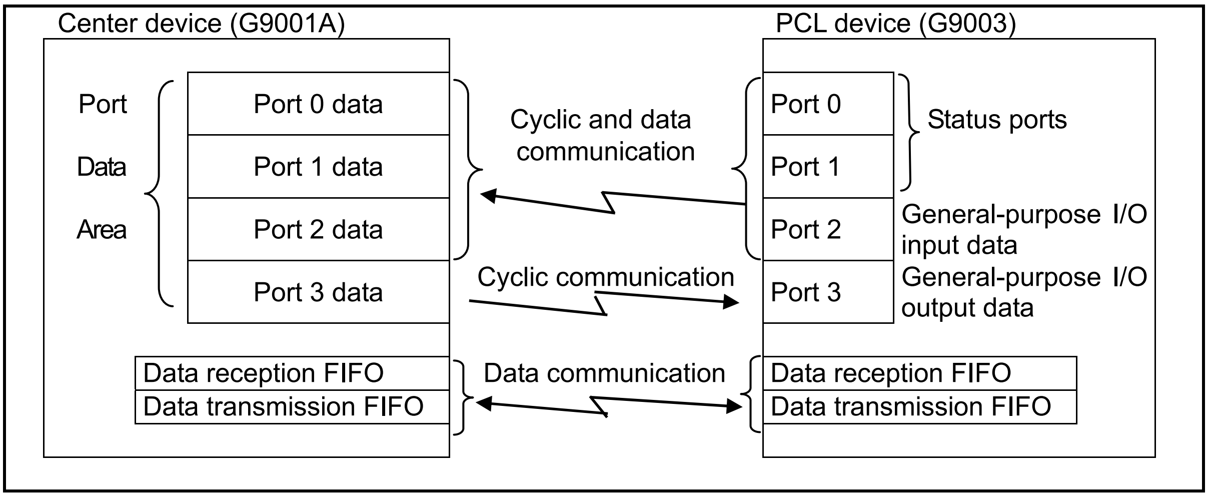
2) Cyclic communication

In cyclic communication, the center device communicates continuously to perform I/O control of the I/O devices. This communication takes place in cycles. (Communication starts with the local device that has the lowest device number and proceeds through all the devices that are present. When the communication with the device that has the highest number is complete, the center device again starts to communicate with the local device that has the lowest device number.) If the communication target is a data device, it exchanges information such as device status.
By writing a cyclic communication start command, the center device communicates only with devices whose "device information" bit is set to 1.
This communication continues until a cyclic communication stop command is written.
This PCL device (G9003) checks the status and input and output conditions of the general-purpose I/O ports using cyclic communications.

3) Data communication

In data communication, the center device communicates with other data devices, such as the PCL device (G9003).
Normally, the center device executes cyclic communications continuously. A data communication command from a CPU allows you to perform data communications by interrupting the cyclic communications.
After writing data to the data transmitting FIFO of the center device, write a send data command. The center device will start the data communication on an interrupt when the current cyclic communication is complete.
After a local device has received data, it will ignore any further data received until it has read out all of the data received, and it will not send any response to the center device while reading the data. The center device will generate a no response error in this case and retry the communication.

[Conceptual communication diagram]



- An example of how to write the data "01234567h" to the RMV (the feed amount register) in a PCL device (G9003). [An example at the center device]

When using a 16-bit CPU

1) First, write an RMV write command (0090h) to the transmitting FIFO (006h).
2) Next, write the lower 16 bits data (4567h) for the RMV register into the transmitting FIFO (006h).
3) Finally, write the upper 16 bits data (0123h) to be sent to the RMV register into the transmitting FIFO (006h).

	Details of the data transmitting FIFO
1st word	0090h
2nd word	4567h
3rd word	0123h

5-2. Functional settings for the PCL device (G9003)

5-2-1. I/O port

There are four I/O ports. The highest port, Port 3, is generally used for output. Ports 0, 1, and 2 are generally used for input. As shown in the figure below, they are arranged from the highest to the lowest port: for use as a general-purpose I/O terminal output data, for setting general -purpose I/O terminal input data, for the main status upper byte, and main status lower byte.

The general-purpose I/O terminals are set for input or output using register RENV2. Therefore, the data settings for the general-purpose I/O output will only be effective when the general-purpose I/O terminals are set up as outputs.

The general-purpose I/O output data terminals can be reset when the watchdog timer times out.

The output status can be checked by reading the general-purpose I/O input data terminals.

Port 3	Port 2	Port 1	Port 0
General-purpose output data [IOPOB]	General-purpose I/O input data [IOPIB]	Main status (upper byte) [MSTSB1]	Main status (lower byte) [MSTSB0]

5-2-1-1. Main status

MSTSB1								MSTSB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SBSY	0	0	0	0	SEVT	SERR	SEND	SINT

Bit	Bit name	Description
0	SINT	Set to 1 when an interrupt occurs. (Goes to 1 when bit 1, 2, or 3 becomes 1.)
1	SEND	When an operation stops, the device generates an interrupt and this bit is set to 1. This bit is returned to 0 by the Interrupt Reset command (0008h).
2	SERR	Set to 1 when an error interrupt occurs. It is set back to 0 by reading the RESET.
3	SEVT	Set to 1 when an event interrupt occurs. It is set back to 0 by reading the RIST.
4 to 7	Not defined	(Always set to 0)
8	SBSY	Set to 1 when the LSI starts to output pulses. It is set back to 0 by stopping operation.
9 to 15	Not defined	(Always set to 0)

5-2-1-2. General I/O terminal input data (IOPIB)

IOPIB							
7	6	5	4	3	2	1	0
IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0

Bit	Bit name	Description
0 to 7	IP 0 to 7	Read the status of terminals P0 to P7. (0: LOW, 1: HIGH)

5-2-1-3. General I/O terminal output data (IOPOB)

IOPOB

7	6	5	4	3	2	1	0
OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0

Bit #.	Bit name	Description
0 to 7	OP 0 to 7	Set the output status of terminals P0 to P7. (0: LOW, 1: HIGH)

5-3. Command (Operation commands & Control commands)

The PCL device (G9003) control axes through data communications from the center device by using the following commands. Two command types are available, "Commands without data" and "Commands with data".
If data is added to a "command without data," the PCL device (G9003) will ignore the added data.
For "commands with data," if the attached data exceeds the number of effective bits, the PCL (G9003) ignores the data beyond the effective number of bits. If a "command with data" is sent from the center device without attaching data, the PCL (G9003) will not refresh the respective data areas.
If more than 5 words of data are transferred, the center device will generate a local side receive process error (ERAE). In this case, the command will not be resent.

5-3-1. Operation commands

5-3-1-1. Start command

1) Start command
To start operating, write one of these commands while stopped

Symbol	1st word	2nd word	3rd word	Description	Response
STAFL	0050h			FL constant speed start	Response frame only
STAFH	0051h			FH constant speed start	Response frame only
STAUD	0053h			High-speed start (Acceleration -> FH constant speed -> Deceleration)	Response frame only

2) Residual amount start command
If a positioning operation is halted without completing, use these commands to drive the motor the residual number of pulses needed for the positioning operation.

Symbol	1st word	2nd word	3rd word	Description	Response
CNTFL	0054h			Residual amount FL constant speed start	Response frame only
CNTFH	0055h			Residual amount FH constant speed start	Response frame only
CNTUD	0057h			Residual amount high speed start	Response frame only

3) Start command with a feed amount
Write a feed amount and a start command while stopped. The feed amount is useful for positioning operations.

Symbol	1st word	2nd word	3rd word	Description	Response
RMSTFL	0058h	Lower byte data	Upper byte data	WRITE RMV REGISTER + FL constant speed start	Response frame only
RMSTFH	0059h	Lower byte data	Upper byte data	WRITE RMV REGISTER + FH constant speed start	Response frame only
RMSTUD	005Bh	Lower byte data	Upper byte data	WRITE RMV REGISTER + High-speed start	Response frame only

Note 1: If the command is sent without any data, the RMV register will be set to 0 and the motor will be driven "0" feed amount.

4) Simultaneous start command

When several devices are waiting for an #STA signal to arrive by setting the RMD registers, write these commands to start multiple axes simultaneously.

Symbol	1st word	2nd word	3rd word	Description	Response
CMSTA	0006h			#STA output (simultaneous start)	Response frame only
SPSTA	002Ah			Substitute #STA input	Response frame only

5-3-1-2. Speed change command

If any of these commands are written while operating, the operation speed will be changed. If they are written while stopped, the devices will ignore the command.

Symbol	1st word	2nd word	3rd word	Description	Response
FCHGL	0040h			Change to FL constant speed immediately	Response frame only
FCHGH	0041h			Change to FH constant speed immediately	Response frame only
FSCHL	0042h			Decelerate to FL speed	Response frame only
FSCHH	0043h			Accelerate to FH speed	Response frame only

5-3-1-3. Stop command

1) Stop command

Write one of these commands to stop the operation.

Symbol	1st word	2nd word	3rd word	Description	Response
STOP	0049h			Immediate stop	Response frame only
SDSTP	004Ah			Decelerate and stop	Response frame only

2) Simultaneous stop command

When this command is input, the device stops any axis whose #STP input stop function is enabled by setting the RMD register.

Symbol	1st word	2nd word	3rd word	Description	Response
CMSTP	0007h			#STP output (simultaneous stop)	Response frame only

3) Emergency stop command

Stop the motor in an emergency

Symbol	1st word	2nd word	3rd word	Description	Response
CMEMG	0005h			Emergency stop	Response frame only

5-3-2. Control commands

Commands to control various items such as resetting the counters.

5-3-2-1. NOP (do nothing) command

Symbol	1st word	2nd word	3rd word	Description	Response
NOP	0000h			Invalid command	Response frame only

5-3-2-2. SEND interrupt reset command

Symbol	1st word	2nd word	3rd word	Description	Response
INTRS	0008h			Interrupt output (main status_bit 0)	Response frame only

5-3-2-3. Software reset command

Reset the registers and commands stored in this device (except for communication related items)

Symbol	1st word	2nd word	3rd word	Description	Response
SRST	0004h			Reset the software	Response frame only

5-3-2-4. Counter reset command

Set the specified counter to 0.

Symbol	1st word	2nd word	3rd word	Description	Response
CUN1R	0020h			Reset COUNTER1 (command position)	Response frame only
CUN2R	0021h			Reset COUNTER2 (machine position)	Response frame only
CUN3R	0022h			Reset COUNTER3 (general-purpose, deflection)	Response frame only

5-3-2-5. ERC output control command

Control the ERC signal using commands.

Symbol	1st word	2nd word	3rd word	Description	Response
ERCOUT	0024h			Output an ERC signal	Response frame only
ERCRST	0025h			Reset the ERC signal	Response frame only

5-3-2-6. PCS input command

Has the same results as turning ON the PCS input.

Symbol	1st word	2nd word	3rd word	Description	Response
Staon	0028h			Substitute pcs input	Response frame only

5-3-2-7. LTC input (counter latch) command

Has the same results as turning on the LTC input

Symbol	1st word	2nd word	3rd word	Description	Response
LTCH	0029h			Substitute LTC input	Response frame only

5-3-3. Register control commands

5-3-3-1. Register write controls

Symbol	1st word	2nd word	3rd word	Description	Response
WRMVOR	0080h	Lower byte data	Upper byte data	Overwrite the RMV register	Response frame only
WRMV	0090h	Lower byte data	Upper byte data	Write to the RMV register	Response frame only
WRFL	0091h	Lower byte data	Upper byte data	Write to the RFL register	Response frame only
WRFH	0092h	Lower byte data	Upper byte data	Write to the RFH register	Response frame only
WRUR	0093h	Data		Write to the RUR register	Response frame only
WRDR	0094h	Data		Write to the RDR register	Response frame only
WRMG	0095h	Data		Write to the RMG register	Response frame only
WRDP	0096h	Lower byte data	Upper byte data	Write to the RDP register	Response frame only
WRMD	0097h	Lower byte data	Upper byte data	Write to the RMD register	Response frame only
WRUS	0099h	Data		Write to the RUS register	Response frame only
WRDS	009Ah	Data		Write to the RDS register	Response frame only
WRFA	009Bh	Lower byte data	Upper byte data	Write to the RFA register	Response frame only
WRENV1	009Ch	Lower byte data	Upper byte data	Write to the RENV1 register	Response frame only
WRENV2	009Dh	Lower byte data	Upper byte data	Write to the RENV2 register	Response frame only
WRENV3	009Eh	Lower byte data	Upper byte data	Write to the RENV3 register	Response frame only
WRENV4	009Fh	Lower byte data	Upper byte data	Write to the RENV4 register	Response frame only
WRENV5	00A0h	Lower byte data	Upper byte data	Write to the RENV5 register	Response frame only
WRENV6	00A1h	Lower byte data	Upper byte data	Write to the RENV6 register	Response frame only
WRCUN1	00A3h	Lower byte data	Upper byte data	Write to the RCUN1 register	Response frame only
WRCUN2	00A4h	Lower byte data	Upper byte data	Write to the RCUN2 register	Response frame only
WRCUN3	00A5h	Data		Write to the RCUN3 register	Response frame only
WRCMP1	00A7h	Lower byte data	Upper byte data	Write to the RCMP1 register	Response frame only
WRCMP2	00A8h	Lower byte data	Upper byte data	Write to the RCMP2 register	Response frame only
WRCMP3	00A9h	Lower byte data	Upper byte data	Write to the RCMP3 register	Response frame only
WRIRQ	00ACh	Lower byte data	Upper byte data	Write to the RIRQ register	Response frame only

5-3-3-2. Register read controls

Symbol	1st word	2nd word	3rd word	Description	Response
RRMV	00D0h			Read the RMV register	ARMV data
RRFL	00D1h			Read the RFL register	ARFL data
RRFH	00D2h			Read the RFH register	ARFH data
RRUR	00D3h			Read the RUR register	ARUR data
RRDR	00D4h			Read the RDR register	ARDR data
RRMG	00D5h			Read the RMG register	ARMG data
RRDP	00D6h			Read the RDP register	ARDP data
RRMD	00D7h			Read the RMD register	ARMD data
RRUS	00D9h			Read the RUS register	ARUS data
RRDS	00DAh			Read the RDS register	ARDS data
RRFA	00DBh			Read the RFA register	ARFA data
RRENV1	00DCh			Read the RENV1 register	ARENV1 data
RRENV2	00DDh			Read the RENV2 register	ARENV2 data
RRENV3	00DEh			Read the RENV3 register	ARENV3 data
RRENV4	00DFh			Read the RENV4 register	ARENV4 data
RRENV5	00E0h			Read the RENV5 register	ARENV5 data
RRENV6	00E1h			Read the RENV6 register	ARENV6 data
RRCUN1	00E3h			Read the RCUN1 register	ARCUN1 data
RRCUN2	00E4h			Read the RCUN2 register	ARCUN2 data
RRCUN3	00E5h			Read the RCUN3 register	ARCUN3 data
RRCMP1	00E7h			Read the RCMP1 register	ARCMP1 data
RRCMP2	00E8h			Read the RCMP2 register	ARCMP2 data
RRCMP3	00E9h			Read the RCMP3 register	ARCMP3 data
RRIRQ	00ECh			Read the RIRQ register	ARIRQ data
RRLTC1	00EDh			Read the RLTC1 register	ARLTC1 data
RRLTC2	00EEh			Read the RLTC2 register	ARLTC2 data
RRLTC3	00EFh			Read the RDTC3 register	ARLTC3 data
RRSTS	00F1h			Read the RSTS register	ARSTS data
RREST	00F2h			Read the REST register	AREST data
RRIST	00F3h			Read the RIST register	ARIST data
RRPLS	00F4h			Read the RPLS register	ARPLS data
RRSPD	00F5h			Read the RSPD register	ARSPD data
RRSDC	00F6h			Read the RSDC register	ARSDC data

Note 1: If a register read command has 4 words of data attached, the PCL device (G9003) will not respond and the center device will generate a data communication error (EDTE).

5-3-3-3. Response data from read commands

Symbol	1st word	2nd word	3rd word	Description	Effective number of bits
ARMV	00D0h	Lower byte data	Upper byte data	Read the RMV register	28
ARFL	00D1h	Lower byte data	Upper byte data	Read the RFL register	17
ARFH	00D2h	Lower byte data	Upper byte data	Read the RFH register	17
ARUR	00D3h	Data	0	Read the RUR register	16
ARDR	00D4h	Data	0	Read the RDR register	16
ARMG	00D5h	Data	0	Read the RMG register	11
ARDP	00D6h	Lower byte data	Upper byte data	Read the RDP register	24
ARMD	00D7h	Lower byte data	Upper byte data	Read the RMD register	25
ARUS	00D9h	Data	0	Read the RUS register	16
ARDS	00DAh	Data	0	Read the RDS register	16
ARFA	00DBh	Lower byte data	Upper byte data	Read the RFA register	17
ARENV1	00DCh	Lower byte data	Upper byte data	Read the RENV1 register	30
ARENV2	00DDh	Lower byte data	Upper byte data	Read the RENV2 register	23
ARENV3	00DEh	Lower byte data	Upper byte data	Read the RENV3 register	31
ARENV4	00DFh	Lower byte data	Upper byte data	Read the RENV4 register	28
ARENV5	00E0h	Lower byte data	Upper byte data	Read the RENV5 register	32
ARENV6	00E1h	Lower byte data	Upper byte data	Read the RENV6 register	32
ARCUN1	00E3h	Lower byte data	Upper byte data	Read the RCUN1 register	28
ARCUN2	00E4h	Lower byte data	Upper byte data	Read the RCUN2 register	28
ARCUN3	00E5h	Data	Sign extension	Read the RCUN3 register	16
ARCMP1	00E7h	Lower byte data	Upper byte data	Read the RCMP1 register	28
ARCMP2	00E8h	Lower byte data	Upper byte data	Read the RCMP2 register	28
ARCMP3	00E9h	Lower byte data	Upper byte data	Read the RCMP3 register	28
ARIRQ	00ECh	Data	0	Read the RIRQ register	15
ARLTC1	00EDh	Lower byte data	Upper byte data	Read the RLTC1 register	28
ARLTC2	00EEh	Lower byte data	Upper byte data	Read the RLTC2 register	28
ARLTC3	00EFh	Lower byte data	Upper byte data	Read the RLTC3 register	17
ARSTS	00F1h	Lower byte data	Upper byte data	Read the RSTS register	23
AREST	00F2h	Data	0	Read the REST register	15
ARIST	00F3h	Data	0	Read the RIST register	15
ARPLS	00F4h	Lower byte data	Upper byte data	Read the RPLS register	28
ARSPD	00F5h	Lower byte data	Upper byte data	Read the RSPD register	27
ARSDC	00F6h	Lower byte data	Upper byte data	Read the RSDC register	24

5-4. Register

The initial value of all the registers is "0."
If the new value you want to set is the same as the current value, you do not need to overwrite it.

5-4-1. Speed setting registers

These registers are used to set the operating speeds.
Please note that with some registers, if a "0" is placed in the register, it will be outside the allowed setting range.
For details about speed setting, see "7-2. Setting speed patterns."

5-4-1-1. RMV: Positioning amount register (28 bits)

These registers are used to specify the target position for positioning operations.

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

The details for setting the register may vary with the operation mode.
Setting range: -134,217,728 to +134,217,727.
By changing the RMV register while in operation, the feed length can be overridden.

5-4-1-2. RFL: FL speed registers (17 bits)

These registers are used to set the initial speed (stop seed) for high speed (with acceleration /deceleration) operations.

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

Set the speed for FL constant speed operation and the start speed for high-speed operation (acceleration/deceleration operation): must be in the range of 1 to 100,000 (186A0h). Values from 100,000 to 131,071 (186A0h to 1FFFFh) will be treated as 100,000.
The actual operation speed will be the value calculated using the RMG value.

5-4-1-3. RFH: FH speed registers (17 bits)

These registers are used to specify the operation speed.

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

By changing the RFH register during operation, you can override the current speed.
Set the speed for FH constant speed and high-speed operation (acceleration/deceleration operation): must be in the range of 1 to 100,000 (186A0h). Values from 100,000 to 131,071 (186A0h to 1FFFFh) will be treated as 100,000.
When you choose high-speed operation (acceleration/deceleration operation), use a value larger than the RFL value that is specified.
The actual operation speed will be the value calculated using the RMG set value.

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

5-4-1-4. RUR: Acceleration rate register (16 bits)

These registers are used to specify the acceleration rate.

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

Sets the acceleration characteristics for high-speed operation (acceleration/deceleration operation), range: 1 to 65,535 (0FFFFh).

5-4-1-5. RDR: Deceleration rate registers (16 bits)

These registers are used to specify the deceleration rate.

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

Normally, the deceleration characteristic of the high-speed operation (acceleration/deceleration) is set within 1 to 65,535 (0FFFFh).
Even when an automatic ramp down point setting is selected (MSDP = 0 in the RMD register), the value set in the RDR register will be used as the deceleration rate. When the RDR = 0, the deceleration rate will be the value placed in the RUR.

5-4-1-6. RMG: Multiplication rate register (11 bits)

These registers are used to set the speed magnification rate.

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

Specify the relationship between the RFL, RFH, and RFA values and the operating speed, within the range of 2 to 2,047 (07FFh). The higher the multiplication rate, the coarser the speed steps that can be selected. Normally, use as small a multiplication rate as possible. The operation speed [PPS] will be the product of multiplying the speed rate by the speed register setting.

5-4-1-7. RDP: Ramp down point register (24 bits)

These registers are used to set a ramping-down point (deceleration start point) for positioning operations.

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

Sets the value used to determine the deceleration start point in an acceleration/deceleration or positioning operation.
Bits with a "#" symbol are ignored when written and change their setting when read according to the setting of MSDP (bit 12) in the RMD register.

MSDP	Setting details	bit #
0	Offset for automatically set values. When a positive value is entered, the PCL device (G9003) will start deceleration earlier and the FL speed range will be used longer. When a negative value is entered, the PCL device (G9003) will start deceleration later and will not reach the FL speed.	Same as bit 23.
1	When number of pulses left drops to less than a set value, the motor on that axis starts to decelerate.	0

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

5-4-1-8. RUS: Acceleration S-curve range register (16 bits)

These registers are used to specify the S-curve range of the S-curve acceleration.

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

Specify the S-curve acceleration value for an S-curve acceleration/deceleration operation, within the range of 1 to 50,000 (0C350h).
Values from 50,000 to 65,535 (0C350h to 0FFFFh) will all be treated as 50,000.
The S-curve acceleration range S_{SU} will be calculated from the RMG value.
If "0" is entered, the PCL device (G9003) will substitute the value calculated by (RFH - RFL) / 2, and will operate using an S-curve that does not have any linear sections.
If a value larger than (RFH - RFL) / 2 is entered, the PCL device (G9003) will not reach the maximum acceleration speed and the acceleration time will not match the speed calculated using the formula.
Therefore enter a value smaller than "(RFH - RFL) / 2."

5-4-1-9. RDS: Deceleration S-curve range register (16 bits)

These registers are used to specify the S-curve range of the S-curve deceleration.

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

Specify the S-curve deceleration value for an S-curve acceleration/deceleration operation, within the range of 1 to 50,000 (0C350h).
Values from 50,000 to 65,535 (0C350h to 0FFFFh) will all be treated as 50,000.
The S-curve deceleration range S_{SD} will be calculated from the RMG value.
If "0" is entered, the PCL device (G9003) will substitute the value calculated by (RFH - RFL) / 2, and will operate using an S-curve that does not have any linear sections.
If a value larger than (RFH - RFL) / 2 is entered, the PCL device (G9003) will not reach the maximum deceleration speed and the deceleration time will not match the speed calculated using the formula.
Therefore enter a value smaller than "(RFH - RFL) / 2."

5-4-1-10. RFA: FA speed register (17 bits)

This register is used to set the constant speed for backlash correction.

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

Set the speed to feed a specific amount (FA speed) during backlash correction, within the range of 1 to 100,000 (186A0h).
Values from 100,000 to 131,071 (186Ah to 1FFFFh) will all be treated as 100,000.
The actual operation speed will be calculated from the RMG value.
This value can also be used for the reverse constant speed during an origin return operation.

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

5-4-2. Environment setting registers

The environment setting registers consist of registers used to set and monitor the operation mode, counters, comparators, environment, and interrupt controls.

Register	Description	Bit length	Set range	R/W
RMD	Set the operation mode	25		R/W
RENV1	Environment Register 1	31		R/W
RENV2	Environment Register 2	23		R/W
RENV3	Environment Register 3	31		R/W
RENV4	Environment Register 4	28		R/W
RENV5	Environment Register 5	32		R/W
RENV6	Environment Register 6	32		R/W
RCUN1	COUNTER1 (Instruction position)	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	R/W
RCUN2	COUNTER2 (Machine position)	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	R/W
RCUN3	COUNTER3 (General-purpose, Deflection)	16	-32,768 to 32,767 (8000h) (7FFFh)	R/W
RCMP1	Comparison data for Comparator 1	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	R/W
RCMP2	Comparison data for Comparator 2	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	R/W
RCMP3	Comparison data for Comparator 3	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	R/W
RIRQ	Set various event interrupt factors	15		R/W
RCUN1	COUNTER1 latch data	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	R/W
RCUN2	COUNTER2 latch data	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	R/W
RCUN3	COUNTER3 latch data	17	-32,768 to 32,767 (8000h) (7FFFh)	R/W
RSTS	Extension status	23		R
REST	Error interrupt status	15		R
RIST	Event interrupt status	15		R
RPLS	Positioning Counter (residual number of pulses to feed)	28	0 to 134,217,728 (8000000h)	R
RSPD	EZ counter, current speed monitor	27	1 to 100,000 (186A0h), and others	R
RSDC	Automatic ramp down point calculated value	24	0 to 16,777,215 (0FFFFFFh)	R

5-4-2-1. RMD registers

These registers are used to set the operation mode.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	MIOR	MFH	MUB	MMPH	MPH	MINT	MMSK	MADJ	MSPO

Bits	Bit name	Description
Setting basic operation mode		
0 to 6	MOD	Set operation mode. 000 0000 (00h): Continuous positive rotation controlled by command control. 000 1000 (08h): Continuous negative rotation controlled by command control. 000 0001 (01h): Continuous operation controlled by pulsar (PA/PB) input. 001 0000 (10h): Positive rotation origin return operation. 001 1000 (18h): Negative rotation origin return operation. 001 0010 (12h): Positive feed leaving from the origin position. 001 1010 (1Ah): Negative feed leaving from the origin position. 001 0101 (15h): Origin search in the positive direction 001 1101 (1Dh): Origin search in the negative direction 010 0000 (20h): Feed to +EL or +SL position. 010 1000 (28h): Feed to -EL or -SL position. 010 0010 (22h): Move away from the -EL or -SL position. 010 1010 (2Ah): Move away from the +EL or +SL position. 010 0100 (24h): Feed in the positive direction for a specified number of EZ counts. 010 1100 (2Ch): Feed in the negative direction for a specified number of EZ counts. 100 0001 (41h): Positioning operation (specify the incremental target position) 100 0010 (42h): Positioning operation (specify the absolute position in COUNTER1) 100 0011 (43h): Positioning operation (specify the absolute position in COUNTER2) 100 0100 (44h): Zero return of command position (COUNTER1). 100 0101 (45h): Zero return of mechanical position (COUNTER2). 100 0110 (46h): Single pulse operation in the positive direction. 100 1110 (4Eh): Single pulse operation in the negative direction. 100 0111 (47h): Timer operation 101 0001 (51h): Positioning operation controlled by pulsar (PA/PB) input. 101 0010 (52h): Positioning operation is synchronized with PA/PB (specify the absolute position of COUNTER1) 101 0011 (53h): Positioning operation is synchronized with PA/PB (specify the absolute position of COUNTER2 101 0100 (54h): Zero return to the specified position controlled by pulsar (PA/PB) input. 101 0101 (55h): Zero return to a mechanical position controlled by pulsar (PA/PB) input.
7	Not defined	(Always set to 0.)
Optical setting items		
8	MSDE	0: SD input will be ignored. (Checking can be done with RSTS in sub status) 1: Decelerates (deceleration stop) by turning ON the input.
9	MINP	0: Delay using an INP input will be possible. (Checking can be done with RSTS in sub status) 1: Completes operation by turning ON the INP input.

Bits	Bit name	Description
10	MSMD	Specify an acceleration/deceleration type for high speed feed. (0: Linear accel/decel. 1: S-curve accel/decel.)
11	METM	Specify the operation complete timing. (0: End of cycle. 1: End of pulse.) When using the vibration reduction function, select "End of pulse."
12	MSDP	Specify the ramping-down point for high speed feed. (Effective during positioning operations.) (0: Automatic setting. 1: Manual setting.)
13	MPCS	Enable or disable PCS input. (1: When positioning, the PCL (G9003) device controls the number of pulses after the PCS input turns ON.) [Override 2 for the target position.]
14	MSY	After writing a start command, the PCL device (G9003) can delay the start until some other event occurs. 0: Starts immediately. 1: Starts when the #STA input is supplied, or with the simultaneous start command 0006h, 002Ah.
15	MSPE	1: Decelerate and stop or stop the motor immediately when the #STP input is supplied. When the #STP input is active because of an error stop on another axis, or because the external #STP signal is present, the motor will stop at the same time.
16	MSPO	1: Outputs an #STP (simultaneous stop) signal when stopping due to an error.
17	MADJ	Specify an FH correction function. (0: ON. 1: OFF.)
18	MMSK	1: Masks the pulse output.
19	MINT	1: Masks the interrupt output (SINT). (The interrupt status will continue to change.)
20	MPH	Select the signal output for the following terminals: #BSY/PH1, #FUP/PH2, #FDW/PH3, and #MVC/PH4. 0: Output #BSY, #FUP, #FDW, and #MVC signals. 1: Output PH1, PH2, PH3, and PH4 signals.
21	MMPH	Mask PH1, PH2, PH3, and PH4 signals. 0: Outputs a LOW from PH1, PH2, PH3, and PH4. 1: Outputs the PH1, PH2, PH3, and PH4 real-time signals.
22	MUB	Select the excitation method fro the PH1, PH2, PH3, and PH4 signals. 0: Output excitation sequence for a 2-phase unipolar motor. 1: Output excitation sequence for a 2-phase bipolar motor.
23	MFH	Select the excitation sequence for the PH1, PH2, PH3, and PH4 signals. 0: Output a full-step excitation sequence. 1: Output a half-step excitation sequence.
24	MIOR	Select a monitoring method for the general-purpose I/O port output setting bit. This is used when you do not want the input change interrupt in the center device to function when an output port changes. 0: Read the output bit status from port 2. 1: Regardless of status of the output bit, the respective bit on port 2 becomes "0."
25 to 31	Not defined	(Always set to 0.)

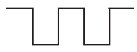
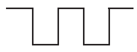
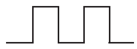
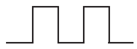
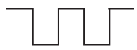
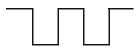
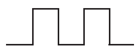
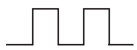
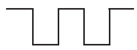
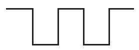
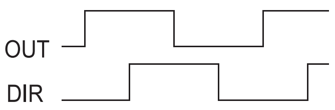
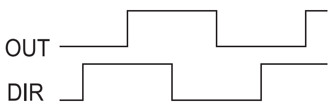
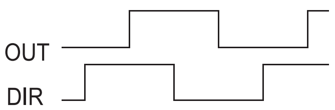
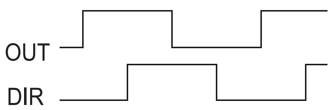
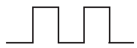
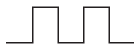
5-4-2-2. RENV1 register

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	PDTC	SEDR	SEDM	DTMF	FLTR	PCSL	LTCL	INPL	CLR1	CLR0	STPM	STAM	ETW1	ETW0

Bits	Bit name	Description
0 to 2	PMD0 to 2	Specify OUT output pulse details

PMD 0 to 2	When feeding in a positive direction		When feeding in a negative direction	
	OUT output	DIR output	OUT output	DIR output
000		High		Low
001		High		Low
010		Low		High
011		Low		High
100		High	High	
101	 OUT DIR		 OUT DIR	
110	 OUT DIR		 OUT DIR	
111		Low	Low	

Bits	Bit name	Description
3	ELM	Specify the process to occur when the EL input is turned ON. (0: Immediate stop. 1: Deceleration stop.) Note 1, 2
4	SDM	Specify the process to occur when the SD input is turned ON. (0: Deceleration only. 1: Deceleration and stop.)
5	SDLT	Specify the latch function of the SD input. (0: OFF. 1: ON.) Turns ON when the SD signal width is short. When the SD input is OFF while starting, the latch signal is reset. The latch signal is also reset when SDLT is 0.
6	SDL	Specify the SD signal input logic. (0: Negative logic. 1: Positive logic.)
7	ORGL	Specify the ORG signal input logic. (0: Negative logic. 1: Positive logic.)
8	ALMM	Specify the process to occur when the ALM input is turned ON. (0: Immediate stop. 1: Deceleration stop.)
9	ALML	Specify the ALM signal input logic. (0: Negative logic. 1: Positive logic.)
10	EROE	1: Automatically outputs an ERC signal when the axis is stopped immediately by a +EL, -EL, ALM, or #EMG input signal. However, the ERC signal is not output when a deceleration stop occurs on the axis. When the EL signal is specified for a normal stop, by setting MOD = "010X000" (feed to the EL position) in the RMD register, the ERC signal is output if an immediate stop occurs.

Bits	Bit name	Description
11	EROR	1: Automatically output the ERC signal when the axis completes an origin return.
12 to 14	EPW0 to 2	Specify the pulse width of the ERC output signal. 000: 12 μsec 001: 102 μsec 010: 409 μsec 011:1.6 msec 100: 13 msec 101: 52 msec 110: 104 msec 111: Level output
15	ERCL	Specify the ERC signal output logic. (0: Negative logic. 1: Positive logic.)
16 to 17	ETW0 to 1	Specify the ERC signal OFF timer time. 00: 0 μsec 10: 1.6 msec 01: 12 μsec 11: 104 msec
18	STAM	Specify the #STA signal input type. (0: Level trigger. 1: Edge trigger.)
19	STPM	Specify a stop method using #STP input. (0: Immediate stop. 1: Deceleration stop.)
20 to 21	CLR0 to 1	Specify a CLR input. 00: Clear on the falling edge 10: Clear on a LOW. 01: Clear on the rising edge 11: Clear on a HIGH.
22	INPL	Specify the INP signal input logic. (0: Negative logic. 1: Positive logic.)
23	LTCL	Specify the operation edge for the LTC signal. (0: Falling. 1: Rising)
24	PCSL	Specify the PCS signal input logic. (0: Negative logic. 1: Positive logic.)
25	FLTR	0: Apply a filter to the +EL, -EL, SD, ORG, ALM, or INP inputs. When a filter is applied, signal pulses shorter than 4 μsec are ignored.
26	DTMF	1: Turn OFF the direction change timer (0.2 msec) function. Note 3
27	SEDM	1: Mask SEND output. (Interrupt status will change.)
28	SEDR	1: Reset SEND when starting.
29	PDTC	1: Always make the pulse width a 50% duty cycle.
30	Not defined	(Always set to 0.)
31	Not defined	(Always set to 0.)

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

Note 2: When deceleration stop is selected, this bit remains ON until the PCL device (G9003) decelerates and stops. The PCL device (G9003) determines whether it has stopped normally or not according to the stop timing. Therefore, if an error stop signal is input while decelerating with high speed positioning, the PCL device (G9003) may determine whether the stop was normal. If a constant error stop signal is input, the PCL device (G9003) will not continue to the next operation and it will stop with an error.

Note 3: This value is used when the internal reference clock is 40 MHz.

5-4-2-3. RENV2 register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P1M1	P1M0	PINF	EZL	EDIR	EIM1	EIM0	EINF	P7M	P6M	P5M	P4M	P3M	P2M	P1M	P0M

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	IDL2	IDL1	IDL0	0	POFF	EOFF	PDIR

Bits	Bit name	Description
0	P0M	Specify the operation of the P0 terminals 00: General-purpose input 01: General-purpose output
1	P1M	Specify the operation of the P1 terminals 00: General-purpose input 01: General-purpose output
2	P2M	Specify the operation of the P2 terminal. 00: General-purpose input 01: General-purpose output
3	P3M	Specify the operation of the P3 terminals. 00: General-purpose input 01: General-purpose output
4	P4M	Specify the operation of the P4 terminals. 00: General-purpose input 01: General-purpose output.
5	P5M	Specify the operation of the P5 terminals. 00: General-purpose input 01: General-purpose output
6	P6M	Specify the operation of the P6 terminals. 00: General-purpose input 01: General-purpose output
7	P7M	Specify the operation of the P7 terminals. 00: General-purpose input 01: General-purpose output
8	EINF	0: Apply a noise filter to EA/EB/EZ input. Note 1.
9 to 10	EIM0 to 1	Specify the EA/EB input operation. 00: Multiply a 90 phase difference by 1 (Count up when the EA input phase is ahead.) 01: Multiply a 90 phase difference by 2 (Count up when the EA input phase is ahead.) 10: Multiply a 90 phase difference by 4 (Count up when EA input phase is ahead.) 11: Count up when the PA signal rises, count down when the EB signal falls.
11	EDIR	1: Reverse the counting direction of the EA/EB inputs.
12	EZL	Specify EZ signal input logic. (0: Falling edge. 1: Rising edge.)
13	PINF	0: Apply a noise filter to the PA/PB input. Note 1
14 to 15	PIM0 to 1	Specify the PA/PB input operation. 00: Multiply a 90 phase difference by 1 (Count up when the PA input phase is ahead.) 01: Multiply a 90 phase difference by 2 (Count up when the PA input phase is ahead.) 10: Multiply a 90 phase difference by 4 (Count up when PA input phase is ahead.) 11: Count up when the PA signal rises, count down when the PB signal falls.
16	PDIR	1: Reverse the counting direction of the PA/PB inputs.
17	EOFF	1: Disable EA/EB input.
18	POFF	1: Disable PA/PB input.

Bits	Bit name	Description
19	Not defined	(Always set to 0.)
20 to 22	IDL 0 to 2	Set the number of idling pulses (0 to 7 pulses).
23 to 31	Not defined	(Always set to 0.)

Note 1: Signals shorter than 6 reference clock cycles are ignored.

5-4-2-4. RENV3 register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	BSYC	CI32	CI31	CI30	CI21	CI20	EZD3	EZD2	EZD1	EZD0	ORM3	ORM2	ORM1	ORM0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	CU3H	CU2H	CU1H	0	CU3B	CU2B	CU1B	0	CU3R	CU2R	CU1R	0	CU3C	CU2C	CU1C

Bit	Bit name	Description
0 to 3	ORM0 to 3	Specify an origin return method. 0000: Origin return operation 0 - Stops immediately (deceleration stop when feeding at high speed) by changing the ORG input from OFF to ON. - COUNTER reset timing: When the ORG input is turned ON. 0001: Origin return operation 1 - Stops immediately (deceleration stop when feeding at high speed) by changing the ORG input from OFF to ON, and feeds in the opposite direction at RFA constant speed until ORG input is turned OFF. Then, feeds in the original direction at RFA speed. While doing so, it will stop immediately when the ORG input is turned ON again. - COUNTER reset timing: When the ORG input is turned ON. 0010: Origin return operation 2 - When feeding at constant speed, movement on the axis stops immediately by counting the EZ signal after the ORG input is turned ON. When feeding at high speed, movement on the axis decelerates when the ORG input is turned ON and stops immediately by counting the EZ counts. - COUNTER reset timing: When counting the EZ signal. 0011: Origin return operation 3 - When feeding at constant speed, movement on the axis stops immediately by counting the EZ signal after the ORG input is turned ON. When feeding at high speed, the axis will decelerate and stop by counting the EZ signal after the ORG input is turned ON. - COUNTER reset timing: When counting the EZ signal. 0100: Origin return operation 4 - Stops immediately (deceleration stop when feeding at high speed) by turning the ORG input ON, and feeds in the reverse direction at RFA constant speed. Stops immediately by counting the EZ signal. - COUNTER reset timing: When counting the EZ signal. 0101: Origin return operation 5 - Stop immediately (deceleration stop when feeding at high speed) and reverse direction when the ORG input is turned ON. Then, stop immediately when counting the EZ signal. - COUNTER reset timing: When counting the EZ signal. 0110: Origin return operation 6 - Stop immediately (deceleration stop when ELM = 1) by turning ON the EL input, and reverse at RFA constant speed. Then stop immediately by turning OFF the EL input. - COUNTER reset timing: When EL input is OFF. 0111: Origin return operation 7 - Stop immediately (deceleration stop when ELM = 1) by turning ON the EL input, and reverse direction at RFA constant speed. Then stop immediately by counting the EL signal. - COUNTER reset timing: When stopped by counting the EL input.

Bit	Bit name	Description
0 to 3	ORM0 to 3	1000: Origin return operation 8 - Stop immediately (deceleration stop when ELM = 1) and reverse direction by turning ON the EL signal. Then stop immediately (deceleration stop when feeding at high speed) when counting the EZ signal. - COUNTER reset timing: When counting the EZ signal. 1001: Origin return operation 9 - After executing origin return operation 0, move back to the zero position (operate until COUNTER2 = 0). 1010: Origin return operation 10 - After executing origin return operation 3, move back to the zero position (operate until COUNTER2 = 0). 1011: Origin return operation 11 - After executing origin return operation 5, move back to the zero position (operate until COUNTER2 = 0). 1100: Origin return operation 12 - After executing origin return operation 8, move back to the zero position (operate until COUNTER2 = 0).
4 to 7	EZD0 to 3	Specify the EZ count up value that is used for origin return operations. 0000 (1st count) to 1111 (16th count)
8 to 9	CI20 to 21	Select the input count source for COUNTER2 (mechanical position). 00: EA/EB input 01: Output pulse 10: PA/PB input
10 to 12	CI30 to 32	Select the input count source for COUNTER3 (deflection counter) 000: Output pulse 001: EA/EB input 010: PA/PB input 011: 1/4096 clock of internal reference clock. 100: Output pulse and EA/EB input (deflection counter) 101: Output pulse and PA/PB input (deflection counter) 110: EA/EB input and PA/PB input (deflection counter)
13	BSYC	1: Operate COUNTER3 only while LSI is operating (#BSY is LOW).
14 to 15	Not defined	(Always set to 0.)
16	CU1C	1: Reset COUNTER1 (command position) when the CLR input turns ON.
17	CU2C	1: Reset COUNTER2 (mechanical position) when the CLR input turns ON.
18	CU3C	1: Reset COUNTER3 (general-purpose, deflection counter) when the CLR input turns ON.
19	Not defined	(Always set to 0.)
20	CU1R	1: Reset COUNTER1 (command position) when the origin return is complete.
21	CU2R	1: Reset COUNTER2 (mechanical position) when the origin return is complete.
22	CU3R	1: Reset COUNTER3 (deflection counter) when the origin return is complete.
23	Not defined	(Always set to 0.)
24	CU1B	1: Operate COUNTER1 (command position) while in backlash/slip correction mode.
25	CU2B	1: Operate COUNTER2 (mechanical position) while in backlash/slip correction mode.
26	CU3B	1: Operate COUNTER3 (deflection counter) while in backlash/slip correction mode.
27	Not defined	(Always set to 0.)
28	CU1H	1: Stop the counting operation on COUNTER1 (command position)
29	CU2H	1: Stop the counting operation on COUNTER2 (mechanical position).
30	CU3H	1: Stop the counting operation on COUNTER3 (general-purpose, deflection counter).
31	Not defined	(Always set to 0.)

5-4-2-5. RENV4 register

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	LTOF	LTFD	LTM1	LTM0	C3D1	C3D0	C3S3	C3S2	C3S1	C3S0	C3C1	C3C0

Bit	Bit name	Description
0 to 1	C1C0 to 1	Select a comparison counter for comparator 1. Note 1 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (general-purpose, deflection counter) 11: The comparison conditions were never met.
2 to 4	C1S0 to 2	Select a comparison method for comparator 1. Note 2 001: RCMP1 data = Comparison counter (regardless of counting direction) 010: RCMP1 data = Comparison counter (while counting up) 011: RCMP1 data = Comparison counter (while counting down) 100: RCMP1 data > Comparison counter data 101: RCMP1 data < Comparison counter data 110: Use as positive end software limit (RCMP1< COUNTER1) Others: Treats that the comparison conditions are not satisfied.
5 to 6	C1D0 to 1	Select a process to execute when the Comparator 1 conditions are met. 00: None (use as an INT, terminal output.) 01: Immediate stop. 10: Deceleration stop.
7	Not defined	(Always set to 0.)
8 to 9	C2C0 to 1	Select a comparison counter for Comparator 2. Note 1. 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (general-purpose, deflection counter) 11: The comparison conditions were never met.
10 to 12	C2S0 to 2	Select a comparison method for Comparator 2. Note 2. 001: RCMP2 data = Comparison counter (regardless of counting direction) 010: RCMP2 data = Comparison counter (while counting up) 011: RCMP2 data = Comparison counter (while counting down) 100: RCMP2 data > Comparison counter data 101: RCMP2 data < Comparison counter data 110: Use as negative end software limit (RCMP2>COUNTER1) Others: Treats that the comparison conditions do not meet.
13 to 14	C2D0 to 1	Select a process to execute when the Comparator 2 conditions are met. 00: None (use as an INT, terminal output) 01: Immediate stop. 10: Deceleration stop.
15	Not defined	(Always set to 0.)
16 to 17	C3C0 to 1	Select a comparison counter for Comparator 3. Note 1 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (deflection counter) 11: The comparison conditions were never met.

18 to 21	C3S0 to 3	Select a comparison method for comparator 3. Note 3 0001: RCMP3 data = Comparison counter (regardless of counting direction) 0010: RCMP3 data = Comparison counter (while counting up) 0011: RCMP3 data = Comparison counter (while counting down) 0100: RCMP3 data > Comparison counter data 0101: RCMP3 data < Comparison counter data 0111: Prohibited setting 1000: Use as an output for the IDX (synchronous) signal (regardless of the count direction) 1001: Use as an output for the IDX (synchronous) signal (while counting up) 1010: Use as an output for the IDX (synchronous) signal (while counting down) Others: Always handle as though the comparison conditions are not met.
22 to 23	C3D0 to 1	Select a process to execute when the Comparator 3 conditions are met. 00: None (use as an INT, terminal output.) 01: Immediate stop. 10: Deceleration stop.
24 to 25	LTM0 to 1	Select latch timing of counter (COUNTER 1 to 3) 00: When LTC input goes ON from OFF 01: ORG input 10: When comparator 2 conditions are met. 11: When comparator 3 conditions are met.
26	LTFD	1: Latch current speed data despite COUNTER 3.
27	LTOF	1: Stop latching with a hardware timing. (Effective only for software)
28 to 31	Not defined	(Always set to 0.)

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

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

When the software limit is set, the motor will stop, regardless of the settings on C1D0 to D1 and C2D0 to D1.

(When deceleration stop is selected, the motor will decelerate to a stop when it is started by the high-speed start command.)

Note 3: When C3S0 to 3 is set to 1000 to 1010 (synchronous signal output), select COUNTER3 (general-purpose) for the comparison counter. The other counters cannot be selected.

To set the comparator, select a positive value.

5-4-2-6. RENV5 register

This is a register for the Environment 5 settings. It is primarily used to set feed amount correction data.

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

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

Bit	Bit name	Description
0 to 11	BR0 to 11	Enter a backlash correction amount or a slip correction amount. (0 to 4095)
12	ADJ	Select a feed amount correction method. 00: Turn OFF the correction function. 01: Backlash correction
13 to 14	Not defined	(Always set to 0.)
15	PSTP	1: Even if a stop command is written, the PCL device (G9003) operate for the number of pulses that are already input on PA/PB.
16 to 26	PD0 to 10	Specifies the division ratio for pulses on the PA/PB input. The number of pulses are divided using the set value/2048. When 0 is entered, the division circuit will be OFF. (= 2048/2048) [Setting range : 0 to 2,047]
27 to 31	PMG0 to 4	Specifies the magnification rate for pulses on the PA/PB input. The number of pulses are multiplied by the set value + 1. [Setting range : 0 to 31]

5-4-2-7. RENV6 register

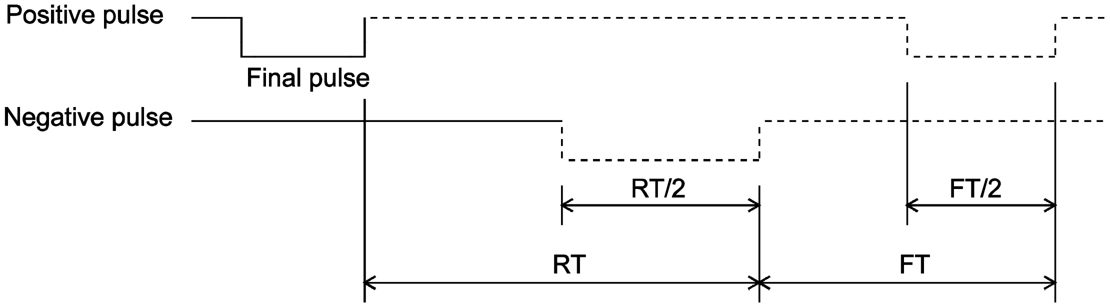
This is a register for the Environment 6 settings. It is primarily used to enter the time for the vibration reduction function. If both RT and FT data are other than zero, the vibration reduction function is turned ON.

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

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

Bit	Bit name	Description
0 to 15	RT0 to 15	Enter the FT time shown in the figure below. [Setting range: 0 to 65,535] The units are 64 ticks of the reference clock (approx. 1.6 μs).
16 to 31	FT0 to 15	Enter the FT time shown in the figure below. [Setting range: 0 to 65,535] The units are 64 ticks of the reference clock (approx. 1.6 μs).

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



Set the time [RT, FT] = Enter a value x 1.6 (μs) [when the internal reference clock is 40 MHz]

5-4-2-8. RCUN1 register

This is a register used for COUNTER1 (command position counter).

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

This is a counter used exclusively for command pulses.
Setting rage: -134,217,728 to +134,217,727.

5-4-2-9. RCUN2 register

This is a register used for COUNTER2 (mechanical position counter).

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

It can count three types of pulses: Command pulses, encoder signals (EA/EB input), pulsar inputs (PA/PB input).
Setting range: -134,217,728 to +134,217,727.

5-4-2-10. RCUN3 register

This is a register used for COUNTER3 (deflection counter).

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

It can count three types of deflections: between command pulses and encoder signals, between command pulses and pulsar signals, and between encoder signals and pulsar signals.
Setting range: -32,768 to +32,767.
The PCL (G9003) device will not count values exceeding the setting and it shows the maximum value.

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

5-4-2-11. RCMP1 register

Specify the comparison data for Comparator 1.

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

Setting range: -134,217,728 to +134,217,727.

5-4-2-12. RCMP2 register

Specify the comparison data for Comparator 2.

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

Setting range: -134,217,728 to +134,217,727.

5-4-2-13. RCMP3 register

Specify the comparison data for Comparator 3.

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

Setting range: -134,217,728 to +134,217,727.

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

5-4-2-14. RIRQ register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	IRNP	0	IRSA	IRSD	IROL	IRLT	IRCL	IRC3	IRC2	IRC1	IRDE	IRDS	IRUE	IRUS	IREN

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

Bit	Bit name	Description
0	IREN	Stopping normally.
1	IRUS	Starting acceleration.
2	IRUE	When ending acceleration.
3	IRDS	When starting deceleration.
4	IRDE	When ending deceleration.
5	IRC1	When Comparator 1 conditions are met.
6	IRC2	When Comparator 2 conditions are met.
7	IRC3	When Comparator 3 conditions are met.
8	IRCL	When resetting the count value with a CLR signal input.
9	IRLT	When latching the count value with an LTC signal input.
10	IROL	When latching the count value with an ORG signal input.
11	IRSD	When the SD input is ON.
12	IRSA	When the #STA input is ON.
13 to 31	Not defined	(Always set to 0.)

5-4-2-15. RLTC1 register

Latched data for COUNTER1 (command position). (Read only.)

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

The contents of COUNTER1 are copied when triggered by the LTC, an ORG input, or an LTCH command.
Data range: -134,217,728 to +134,217,727.

5-4-2-16. RLTC2 register

Latched data for COUNTER2 (mechanical position). (Read only.)

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

The contents of COUNTER2 are copied when triggered by the LTC, an ORG input, or an LTCH command.
Data range: -134,217,728 to +134,217,727.

5-4-2-17. RLTC3 register

Latched data for COUNTER3 (deflection counter) or current speed. (Read only.)

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

The contents of COUNTER3 or the current speed are copied when triggered by the LTC, an ORG input, or an LTCH command.
When the LTFD in the RENV4 register is 0, the register latches the COUNTER3 data. When the LTFD is 1, the register latches the current speed.
When the LTFD is 1 and movement on the axis is stopped, the latched data will be 0.
Data range when LTFD is 0: -32,768 to +32,767.
Data range when LTDF is 1: 0 to 100,000.

When the PCL device (G9003) latches COUNTER3 data (LTFD (bit 26) in the RENV4 is 0), bits shown as "$" and "%" will have the same sign extension as bit 15.
When the PCL device (G9003) latches the current speed data (LTFD (bit 26) in the RENV4 is 1), bits shown as "$" will become "0," and the lower 17 bits with "%" will contain the current speed data.

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

5-4-2-18. RSTS register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERC	SPCS	SEMG	SSTP	SSTA	SDIN	SSD	SORG	SMEL	SPEL	SALM	SDIR	CND3	CND2	CND1	CND0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	SPH4	SPH3	SPH2	SPH1	SPLS	SCP3	SCP2	SCP1	SINP	SLTC	SCLR	SEZ

Bit	Bit name	Description	
0 to 3	CND0 to 3	Reports the operation status. 0000: Under stopped condition 0001: Waiting for #STA input 0010: Waiting for a completion of ERC timer 0011: Waiting for a completion of direction change timer 0100: Correcting backlash 0101: Waiting for PA/PB input 0110: Feeding at FA constant speed 0111: Feeding at FL constant speed	1000: Accelerating 1001: Feeding at FH constant speed. 1010: Decelerating 1011: Waiting for INP input. 1111: Others (controlling start) 1100: Not defined 1101: Not defined 1110: Not defined
4	SDIR	Operation direction (0: Positive direction. 1: Negative direction.)	
5	SALM	Set to 1 when the ALM input signal is ON.	
6	SPEL	Set to 1 when the +EL input signal is ON.	
7	SMEL	Set to 1 when the -EL input signal is ON.	
8	SORG	Set to 1 when the ORG input signal is ON.	
9	SSD	Set to 1 when the SD input signal is ON. (SD latch status.)	
10	SDIN	Set to 1 when the SD input signal is ON. (SD terminal input status.)	
11	SSTA	Becomes 1 when the #STA input signal is turned ON.	
12	SSTP	Becomes 1 when the #STP input signal is turned ON.	
13	SEMG	Becomes 1 when the #EMG input signal is turned ON.	
14	SPCS	Becomes 1 when the PCS input signal is turned ON.	
15	SERC	Becomes 1 when the ERC input signal is turned ON.	
16	SEZ	Becomes 1 when the EZ input signal is turned ON.	
17	SCLR	Becomes 1 when the CLR input signal is turned ON.	
18	SLTC	Becomes 1 when the LTC input signal is turned ON.	
19	SINP	Becomes 1 when the INP input signal is turned ON.	
20	SCP1	Set to 1 when the CMP1 comparison conditions are met.	
21	SCP2	Set to 1 when the CMP2 comparison conditions are met.	
22	SCP3	Set to 1 when the CMP3 comparison conditions are met.	
23	SPLS	Set to 1 when the pulse output (±) is ON. Note 1	
24	SPH1	Set to 1 when the PH1 excitation signal output is ON.	
25	SPH2	Set to 1 when the PH2 excitation signal output is ON.	
26	SPH3	Set to 1 when the PH3 excitation signal output is ON.	
27	SPH4	Set to 1 when the PH4 excitation signal output is ON.	
28 to 31	Not defined	(Always set to 0.)	

Note 1: Logical sum output of the OUT/DIR signals. When the 90 phase difference signal output is selected, the PCL device (G9003) monitors the original pulse output.

5-4-2-19. REST register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	ESPE	ESEE	ESOR	0	ESNT	ESP0	ESSD	ESEM	ESSP	ESAL	ESML	ESPL	ESC3	ESC2	ESC1

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

Bit	Bit name	Description
0	ESC1	Stopped when Comparator 1 conditions were met. (+SL)
1	ESC2	Stopped when Comparator 2 conditions were met. (-SL)
2	ESC3	Stopped when Comparator 3 conditions were met.
3	ESPL	Stopped by the +EL input being turned ON.
4	ESML	Stopped by the -EL input being turned ON.
5	ESAL	Stopped by the ALM input being turned ON.
6	ESSP	Stopped by the #STP input being turned ON.
7	ESEM	Stopped by the #EMG input being turned ON.
8	ESSD	Decelerated and stopped by the SD input being turned ON.
9	ESPO	An overflow occurred in the PA/PB input buffer counter.
10	ESNT	Stopped by a communication error.
11	Not defined	(Always set to 0.)
12	ESOR	Position override could not be executed.
13	ESEE	An EA/EB input error occurred. (Does not stop)
14	ESPE	A PA/PB input error occurred. (Does not stop)
15 to 31	Not defined	(Always set to 0.)

5-4-2-20. RIST register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	ISNP	0	ISSA	ISSD	ISOL	ISLT	ISCL	ISC3	ISC2	ISC1	ISDE	ISDS	ISUE	ISUS	ISEN

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

Bit	Bit name	Description
0	ISEN	Stopped automatically.
1	ISUS	Starting acceleration.
2	ISUE	Ending acceleration.
3	ISDS	Starting deceleration.
4	ISDE	Ending deceleration.
5	ISC1	The comparator 1 conditions were met.
6	ISC2	The comparator 2 conditions were met.
7	ISC3	The comparator 3 conditions were met.
8	ISCL	The count value was reset by a CLR signal input.
9	ISLT	The count value was latched by an LTC input.
10	ISOL	The count value was latched by an ORG input.
11	ISSD	The SD input turned ON.
12	ISSA	The #STA input turned ON.
13 to 31	Not defined	(Always set to 0.)

5-4-2-21. RPLS register

This register is used to check the value of the positioning counter (number of pulses left for feeding). (Read only.)

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

At the start, this value will be the absolute value in the RMV register. Each pulse that is output will decrease this value by one.

5-4-2-22. RSPD register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AS15	AS14	AS13	AS12	AS11	AS10	AS9	AS8	AS7	AS6	AS5	AS4	AS3	AS2	AS1	AS0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	IDC2	IDC1	IDC0	ECZ3	ECZ2	ECZ1	ECZ0	0	0	0	AS16

Bit	Bit name	Description
0 to 16	AS0 to 16	Read the current speed as a step value (same units as for RFL and RFH). When stopped the value is 0.
17 to 19	Not defined	(Always set to 0.)
20 to 23	ECZ0 to 3	Read the count value of EZ input that is used for an origin return.
24 to 26	IDC0 to 2	Read the idling count value.
27 to 31	Not defined	(Always set to 0.)

5-4-2-23. RSDC register

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

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

6. Operation Mode

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

6-1. Continuous operation mode using command control

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

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

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

6-2. Positioning operation mode

The following seven operation types are available for positioning operations.

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

6-2-1. Positioning operation (specify a target position using an incremental value) (MOD: 41h)

This is a positioning mode used by placing a value in the RMV (target position) register.
The feed direction is determined by the sign set in the RMV register.
When starting, the RMV register setting is loaded into the positioning counter (RPLS). The positioning counter counts down with each pulse output, and the PCL device (G9003) stops feeding when the counter reaches 0. When you set the RMV register value to zero to start a positioning operation, the LSI will stop outputting pulses immediately.

6-2-2. Positioning operation (specify the absolute position in COUNTER1) (MOD: 42h)

This mode only uses the difference between the RMV (target position) register value and COUNTER1. Since the COUNTER1 value is stored when starting to move, the PCL device (G9003) cannot be overridden by changing the COUNTER1 value. But, the target position can be overridden by changing the RMV value.
The direction of movement can be set automatically by evaluating the relative relationship between the RMV register setting and the value in COUNTER1.
At start up, the difference between the RMV setting and the value stored in COUNTER1 is loaded into the positioning counter (RPLS). The positioning counter counts down with each pulse output, and when the positioning counter value reaches zero, it stops operation.
If the RMV register value is made equal to the COUNTER1 value and the positioning operation is started, the PCL device (G9003) will immediately stop operation without outputting any command pulses.

6-2-3. Positioning operation (specify the absolute position in COUNTER2) (MOD: 43h)

This mode only uses the difference between the RMV (target position) register setting and the value in COUNTER2.
Since the COUNTER2 value is stored when starting a positioning operation, the PCL device (G9003) cannot be overridden by changing the value in COUNTER2; however, it can override the target position by changing the value in RMV.
The direction of movement can be set automatically by evaluating the relationship between the RMV register setting and the value in COUNTER2.
At start up, the difference between the RMV setting and the value stored in COUNTER2 is loaded into the positioning counter (RPLS). The positioning counter counts down with each pulse output, and when the positioning counter value reaches zero, it stops operation.
If the RMV register value is made equal to the COUNTER2 value and the positioning operation is started, the PCL device (G9003) will immediately stop operation without outputting any command pulses.
Also, this operation does not use feedback control. So, if encoder signals are input to COUNTER2, the value of COUNTER2 at the completion of the feed may be different from the target position.

6-2-4. Command position 0 return operation (MOD: 44h)

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

6-2-5. Machine position 0 return operation (MOD: 45h)

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

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

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

6-2-7. Timer operation (MOD: 47h)

This mode allows the internal operation time to be used as a timer.
The internal effect of this operation is identical to the positioning operation. However, the LSI does not output any pulses (they are masked).
Therefore, the internal operation time using the constant speed start command will be a product of the frequency of the output pulses and the RMV register setting. (Ex.: When the frequency is 1000 pps and the RMV register is set to 120 pulses, the internal operation time will be 120 msec.)
Write a positive number (1 to 134,217,727) into the RMV register.
The ±EL input signal, SD input signal, and software limits are ignored. (These are always treated as OFF.)
The ALM input signal, #STP input signal, and #EMG input signals are effective.
The backlash/slip correction, vibration restriction function, and when changing direction, this timer function is disabled.
The LSI stops counting from COUNTER1 (command position).
Regardless of the MINP setting (bit 9) in the RMD (operation mode) register, an operation complete delay controlled by the INP signal will not occur.
In order to eliminate deviations in the internal operation time, set the METM (bit 11) in the PRMD register to zero and use the cycle completion timing of the output pulse as the operation complete timing.

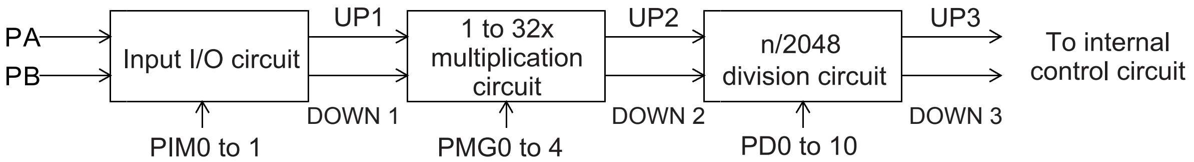
6-3. Pulsar (PA/PB) input mode

This mode is used to allow operations from a pulsar input.
In order to enable pulsar input, set POFF (bit 18) in the RENV2 register to zero.
It is also possible to apply a filter on the PA/PB input.
After writing a start command, when a pulsar signal is input, the LSI will output pulses to the OUT terminal.
Use an FL low speed start (STAFL: 0050h) or an FH low speed start (STAFH: 0051h).
Four methods are available for inputting pulsar signals through the PA/PB input terminal by setting the RENV2 (environmental setting 2) register.

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

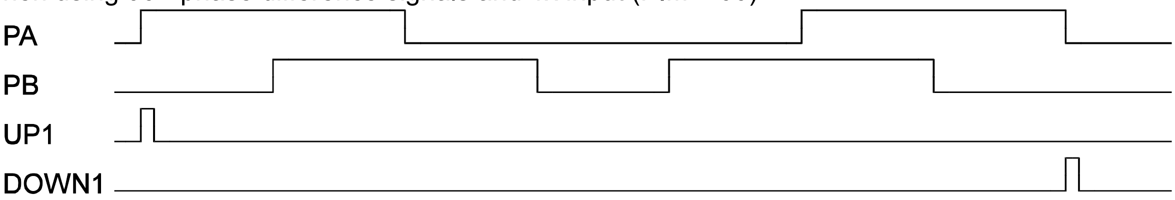
Note: The backlash correction function is available with the pulsar input mode. However, reversing pulsar input while in the backlash correction is unavailable.

Besides the above 1x to 4x multiplication, the PCL has a multiplication circuit of 1x to 32x and division circuit of (1 to 2048)/2048. For setting the multiplication from 1x to 32x, specify the PMG0 to 4 in the RENV5 and for setting the division of n/2048, specify the PD0 to 10 in the RENV5.

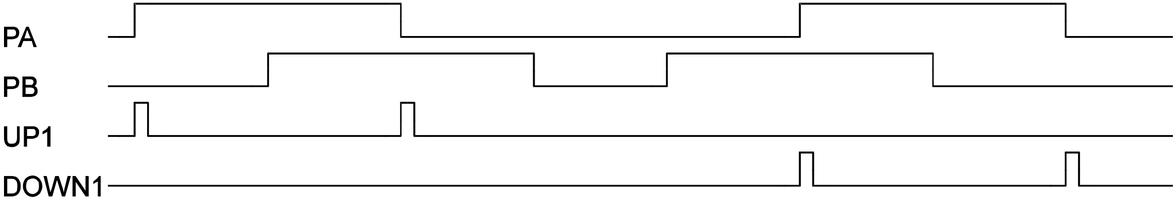


The timing of the UP1 and DOWN1 signals will be as follows by setting of the PIM0 to PIM1 in the RENV2.

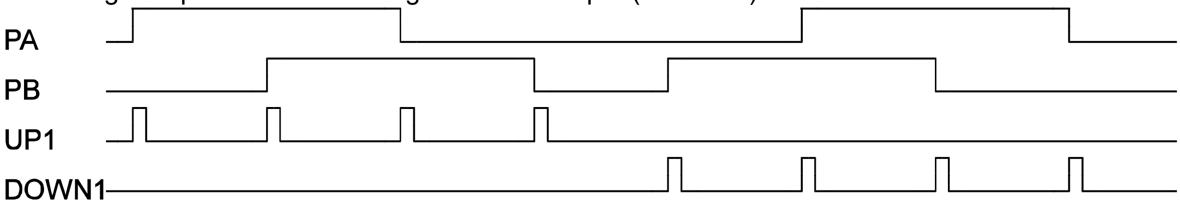
1) When using 90˚ phase difference signals and 1x input (PIM = 00)



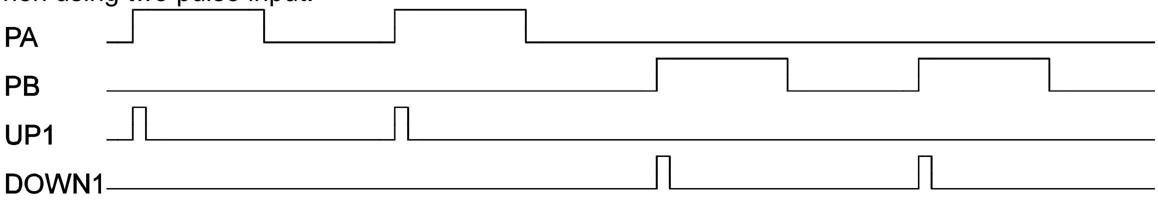
2) When using 90˚ phase difference signals and 2x input (PIM = 01)



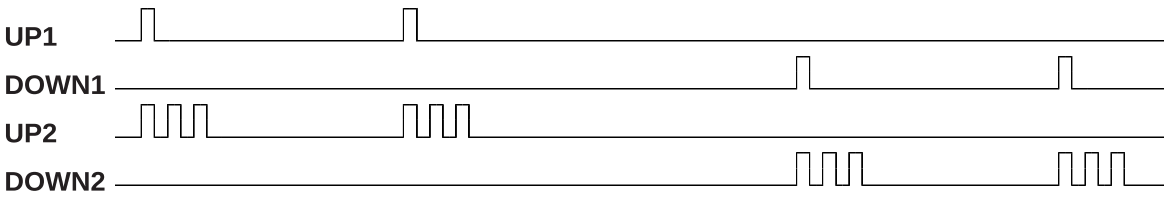
3) When using 90˚ phase difference signals and 4x input (PIM = 10)



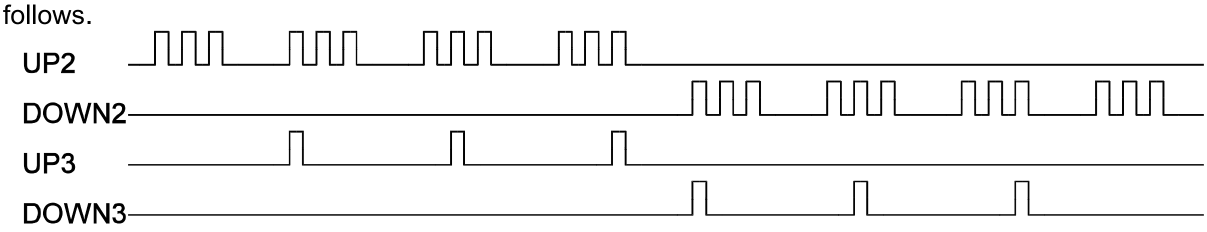
4) When using two pulse input.



When the 1x to 32x multiplication circuit is set to 3x (PMG = 2 on the RENV5), operation timing will be as follows.



When the n/2048 division circuit is set to 512/2048 (PD =512 on the RENV5), operation timing will be as follows.



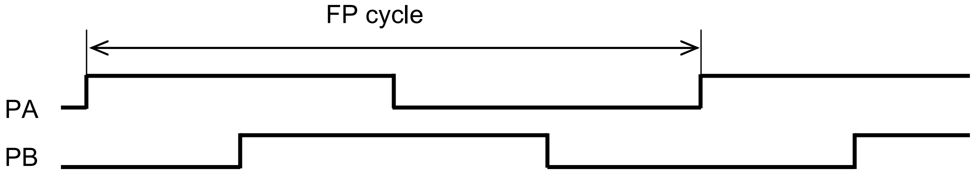
The pulsar input mode is triggered by an FL constant speed start command (0050h) or by an FH constant speed start command (0051h).
Pulsar input causes the PCL device (G9003) to output pulses with some pulses from the FL speed or FH speed pulse outputs being omitted. Therefore, there may be a difference in the timing between the pulsar input and output pulses, up to the maximum internal pulse frequency.

The maximum input frequency for pulsar signals is restricted by the FL speed when an FL constant speed start is used, and by the FH speed when an FH constant speed start is used. The LSI generates an interrupt as errors when both the PA and PB inputs change simultaneously, or when the input frequency is exceeded, or if the input/output buffer counter (deflection adjustment 16-bit counter for pulsar input and output pulse) overflows. This can be monitored by the REST (error interrupt factor) register.

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

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

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



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

<Setting relationship of PA/PB input>

Specify the PA/PB input <Set to PIM0 to 1 (bit 14 to 15) in RENV2> 00: 90 phase difference, 1x 10: 90 phase difference, 4x 01: 90 phase difference, 2x 11: 2 sets of up or down input pulses	[RENV2] (WRITE) 15 8 n n - - - - - -
Specify the PA/PB input count direction <Set to PDIR (bit 16) in RENV2> 0: Count up when the PA phase is leading. Or, count up on the rising edge of PA. 1: Count up when the PB phase is leading. Or, count up on the rising edge of PB.	[RENV2] (WRITE) 23 16 - - - - - - - n
Enable/disable PA/PB input <Set POFF (bit 18) in RENV2> 0: Enable PA/PB input 1: Disable PA/PB input.	[RENV2] (WRITE) 23 16 - - - - - n - -
Set PA/PB input filter <Set PINF (bit 13) in RENV2> 1: Insert a filter on PA/PB input (6 cycles of reference clock) By setting the filter, the PCL device (G9003) ignores signals shorter than 150ns.	[RENV2] (WRITE) 15 8 - - n - - - - -
Reading operation status <CND (bit 0 to 3) in RSTS> 0101 : wait for PA/ PB input.	[RSTS] (READ) 7 0 - - - - n n n n
Reading PA/PB input error <ESPE (bit 14) in REST> ESPE (bit 14) = 1: Occurs a PA/PB input error	[REST] (READ) 15 8 - n - - - - - -
Reading PA/PB input buffer counter status <ESP0 (bit 9) in REST> ESPO (bit 9) = 1: Occurs an overflow.	[REST] (READ) 15 8 - - - - - - n -

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

The pulsar input mode has the following 6 operation types.
The direction of movement for continuous operation can be changed by setting the RENV2 register, without changing the wiring connections for the PA/PB inputs.

MOD	Operation mode	Direction of movement
01h	Continuous operation using pulsar input	Determined by the PA/PB input.
51h	Positioning operation using pulsar input (absolute position)	Determined by the sign of the RMV value.
52h	Positioning operation using pulsar input (COUNTER1 absolute position)	Determined by the relationship of the RMV and COUNTER1 values.
53h	Positioning operation using pulsar input (COUNTER2 absolute position)	Determined by the relationship of the RMV and COUNTER2 values.
54h	Specified position (COUNTER1) zero point return operation using pulsar input	Determined by the sign of the value in COUNTER1.
55h	Specified position (COUNTER2) zero point return operation using pulsar input	Determined by the sign of the value in COUNTER2.

6-3-1. Continuous operation using a pulsar input (MOD: 01h)

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

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

The PCL device (G9003) stops operation when the EL signal in the current feed direction is turned ON.
But the PCL device (G9003) can be operated in the opposite direction without writing a restart command.
When stopped by the EL input, no error interrupt (INT output) will occur.
To release the operation mode, write an immediate stop command (0049h).

Note: When the "immediate stop command (0049h)" is written while the PCL device (G9003) is performing a multiplication operation (caused by setting PIM 0 to 1 and PMG 0 to 4), the PCL device (G9003) will stop operation immediately and the total number of pulses that are output will not be an even multiple of the magnification. When PSTP in RENV5 is set to 1, the PCL device (G9003) delays the stop timing until an even multiple of pulses has been output.
However, if PSTP = 1, the PCL device (G9003) maintains the stop command latched status, regardless of the operation mode selected. When SBSY = 0, the stop command will be disabled.
When using PA/PB input operation and PSTP = 1, check the main status before writing a stop command. When SBSY = 0, return PSTP to 0 and write the stop command.
However, after a stop command is sent by setting PSTP to 1, check the MSTS. If SBSY is 0, set PSTP to 0. (When SBSY is 0 while PSTP is 1, the PCL device (G9003) will latch the stop command.)

6-3-2. Positioning operations using a pulsar input (MOD: 51h)

The PCL device (G9003) positioning is synchronized with the pulsar input by using the RMV setting as incremental position data.
The feed direction is determined by the sign in the RMV (target position) register.
When the RMV register value is loaded to the position counter at start and PA/PB signals are input, the LSI outputs pulses and the positioning counter counts down. When the value in the positioning counter reaches zero, movement on the axis will stop and another PA/ PB input will be ignored.
Set the RMV register value to zero and start the positioning operation. The LSI will stop movement on the axis immediately, without outputting any command pulses.

6-3-3. Positioning operation using pulsar input (specify absolute position to COUNTER1) (MOD: 52h)

The PCL device (G9003) positioning is synchronized with the pulsar input by using the RMV setting as the absolute
value for COUNTER1.
The direction of movement is determined by the relationship between the value in RMV and the value in COUNTER1.
When starting, the difference between the values in RMV and COUNTER1 is loaded into the positioning counter. When a PA/PB signal is input, the PCL device (G9003) outputs pulses and decrements the positioning counter.
When the value in the positioning counter reaches "0," the PCL device (G9003) ignores any further PA/PB input. If you try to start with RMV = COUNTER1, the PCL device (G9003) will not output any pulses and it will stop immediately.

6-3-4. Positioning operation using pulsar input (specify the absolute position in COUNTER2) (MOD: 53h)

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

6-3-5. Command position zero return operation using a pulsar input (MOD: 54h)

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

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

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

6-4. Origin position operation mode

The following six zero position operation modes are available.

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

Depending on the operation method, the zero position operation uses the ORG, EZ, or ±EL inputs.
Specify the input logic of the ORG input signal in the RENV1 (environment 1) register. This register's terminal status can be monitored with an RSTS (extension status) command.
Specify the input logic of the EZ input signal in the RENV2 (environment 2) register. Specify the number for EZ to count up to for an origin return complete condition in the RENV3 (environment 3) register. This register's terminal status can be monitored by reading the RSTS register.
Specify the logic for the ±EL input signal using the ELL input terminals. Specify the operation to execute when the signal turns ON (immediate stop/deceleration stop) in the RENV1 register. This register's terminal status can be monitored with an RSTS (extension status) command.
An input filter can be applied to the ORG input signal and ±EL input signal by setting the RENV1 register.

Set the ORG signal input logic <Set ORGL (bit 7) in RENV1 > 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 n - - - - - - -
Read the ORG signal <SORG (bit 8) in RSTS> 0: Turn OFF the ORG signal 1: Turn ON the ORG signal	[RSTS] (READ) 15 8 - - - - - - - n
Set the EZ signal input logic <Set EZL (bit 12) in RENV2> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 15 8 - - - n - - - -
Set the EZ count <Set EZD0 to 3 (bits 4 to 7) in RENV3> Specify the number for EZ to count up to that will indicate an origin return completion. Enter the value (the count minus 1) in EZD0 to 3. Setting range: 0 to 15.	[RENV3] (WRITE) 7 0 n n n n - - - -
Read the EZ signal < SEZ (bit 16) in RSTS> 0: Turn OFF the EZ signal 1: Turn ON the EZ signal	[RSTS] (READ) 23 16 - - - - - - - n
Set the ±EL signal input logic <ELL input terminal> L: Positive logic input H: Negative logic input	
Specify a method for stopping when the ±EL signal turns ON <Set ELM (bit 3) in RENV1 > 0: Immediate stop when the ±EL signal turns ON. 1: Deceleration stop when the ±EL signal turns ON.	[RENV1] (WRITE) 7 0 - - - - n - - -

Read the ±EL signal <SPEL (bit 6), SMEL (bit 7) in RSTS>
SPEL = 0: Turn OFF + EL signal SPEL = 1: Turn ON + EL signal
SMEL = 0: Turn OFF - EL signal SMEL = 1: Turn ON - EL signal
	[RSTS] (READ)
	7 0
	n n - - - - - -

Applying an input filter to the ±EL and ORG inputs <Set FLTR (bit 25) in RENV1> 1: Apply a filter to the ±EL and ORG inputs. By applying a filter, pulses shorter than 4 µsec will be ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

6-4-1. Origin return operation

After writing a start command, the axis will continue feeding until the conditions for an origin return complete are satisfied.
MOD: 10h Positive direction origin return operation
18h Negative direction origin return operation
When an origin return is complete, the LSI will reset the counter and output an ERC (deflection counter clear) signal.
The RENV3 register is used to set the basic origin return method. That is, whether or not to reset the counter when the origin return is complete. Specify whether or not to output the ERC signal in the RENV1 register.
For details about the ERC signal, see 8-6-2, "ERC signal."

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

[RENV3] (WRITE)

7							0
-	-	-	-	n	n	n	n

0000: Origin return operation 0
- Stop immediately (deceleration stop when feeding at high speed) when the ORG signal turns ON
- COUNTER reset timing: When the ORG input signal turns ON.

0001: Origin return operation 1
- Stop immediately (deceleration stop when feeding at high speed) when the ORG signal turns ON. Next, feed in the reverse direction at RFA low speed until the ORG signal turns OFF. Then, the axis moves back in the original direction at RFA speed and stops immediately when ORG turns ON again.
- COUNTER reset timing: When the ORG input signal turns ON.

0010: Origin return operation 2
- When feeding at low speed, after the ORG signal turns ON, movement on the axis stops immediately when the EZ counter finishes counting up. When feeding at high speed, after the ORG signal turns ON, the axis decelerates and stops immediately when the EZ counter finishes counting up.
- COUNTER reset timing: When the EZ counter finishes counting up.

0011: Origin return operation 3
- When feeding at low speed, after the ORG signal turns ON, movement on the axis stops immediately when the EZ counter finishes counting up. When feeding at high speed, after the ORG signal turns ON, the axis decelerates and stops immediately when the EZ counter finishes counting up.
- COUNTER reset timing: When the EZ counter finishes counting up.

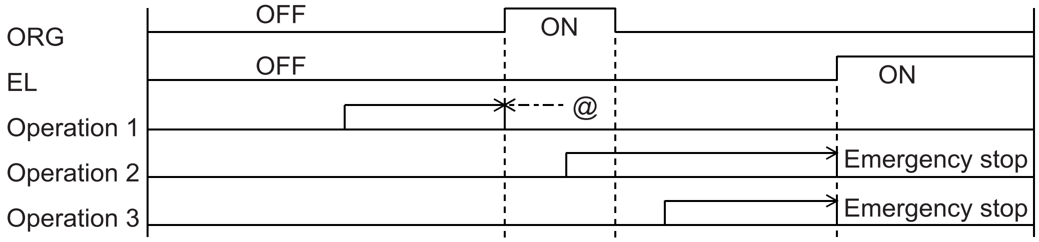
0100: Origin return operation 4
- Movement on the axis stops immediately (decelerate and stop when feeding at high speed) when the ORG input is turned ON. Next, the direction of movement is reversed at RFA low speed. Then, it stops immediately when the EZ counter finishes counting up.
- COUNTER reset timing: When the EZ counter finishes counting up.

0101: Origin return operation 5 - Movement on the axis stops immediately and is reversed (decelerates and stops when feeding at high speed) when the ORG input is turned ON. Then, all movement stops immediately (decelerates and stops when feeding at high speed) when the EZ counter finishes counting up. - COUNTER reset timing: When the EZ counter finishes counting up. 0110: Origin return operation 6 - Movement on the axis stops immediately (decelerates and stops when ELM is 1) when the EL signal turns ON, and it reverses at RFA low speed. Then, all movement stops immediately when the EL signal is turned OFF. - COUNTER reset timing: When the EL signal is turned OFF. 0111: Origin return operation 7 - Movement on the axis stops immediately (decelerates and stops when ELM is 1) when the EL signal turns ON, and reverses at RFA low speed. Then, all movement stops immediately when the EZ counter finishes counting up. - COUNTER reset timing: When the EZ counter finishes counting up. 1000: Origin return operation 8 Movement on the axis stops immediately (decelerates and stops when ELM is 1) when the EL signal turns ON, and reverses. Then it stops immediately (decelerates and stops when feed at high speed) when the EZ counter finishes counting up. - COUNTER reset timing: When the EZ counter finishes counting up. 1001: Origin return operation 9 - After the process in origin return operation 0 has executed, it returns to zero (operates until COUNTER2 = 0). 1010: Origin return operation 10 - After the process in origin return operation 3 has executed, it returns to zero (operates until COUNTER2 = 0). 1011: Origin return operation 11 - After the process in origin return operation 5 has executed, it returns to zero (operates until COUNTER2 = 0). 1100: Origin return operation 12 - After the process in origin return operation 8 has executed, it returns to zero (operates until COUNTER2 = 0).	[RENV3] (WRITE) 7 0 - - - - n n n n
Settings after a origin return complete <Set CU1R to 3R (bits 20 to 22) in RENV3> CU1R (bit 20) =1: Reset COUNTER1 (command position) CU2R (bit 21) =1: Reset COUNTER2 (mechanical position) CU3R (bit 22) =1: Reset COUNTER3 (general-purpose, deflection counter)	[RENV3] (WRITE) 23 16 - n n n - - - -
Setting the ERC signal for automatic output <Set EROR (bit 11) in RENV1> 0: Does not output an ERC signal when an origin return is complete. 1: Automatically outputs an ERC signal when an origin return is complete.	[RENV1] (WRITE) 15 8 - - - - n - - -

6-4-1-1. Origin return operation 0 (ORM = 0000)

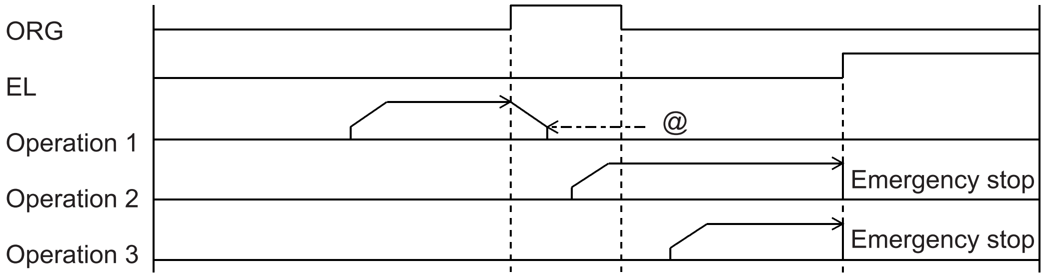
□ Constant speed operation <Sensor: EL (ELM = 0), ORG>

[Starting from here, □ indicates constant speed operation, and ■ indicates high speed operation.]



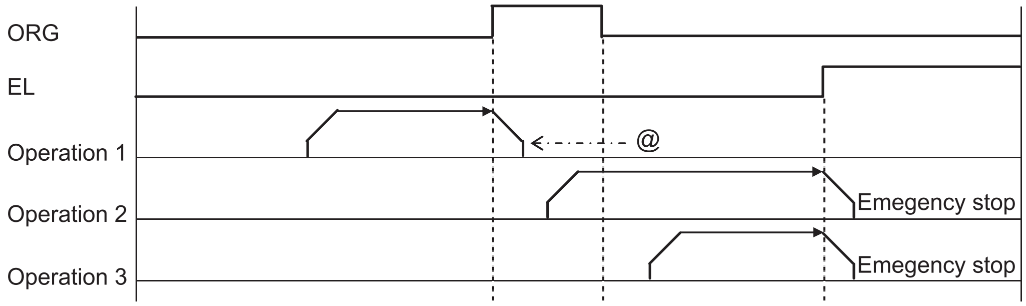
■ High speed operation <Sensor: EL (ELM = 0), ORG>

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

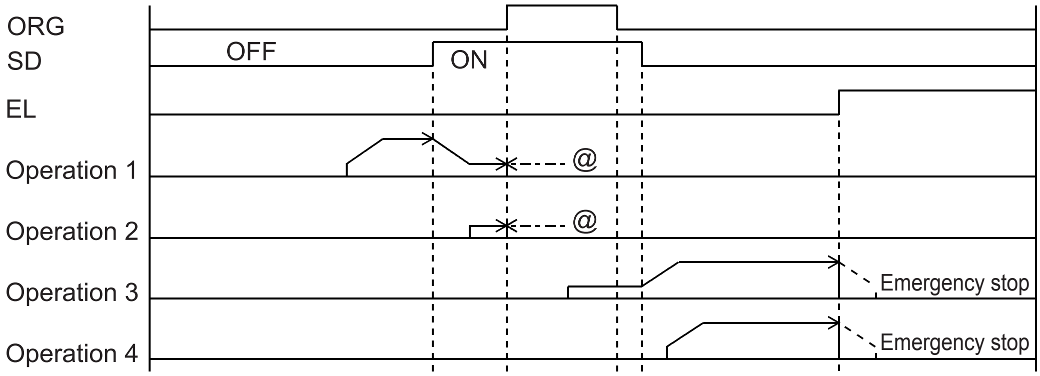


■ High speed operation <Sensor: EL (ELM = 1), ORG>

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



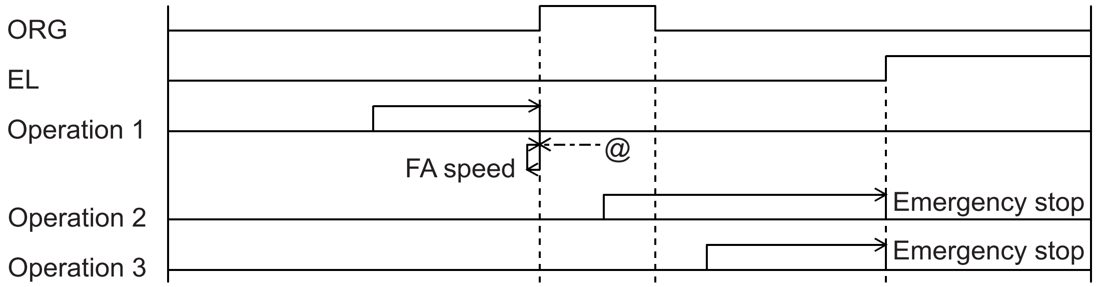
■ High speed operation <Sensor: EL (ELM = 1), SD (SDM = 0, SDLT = 0), ORG>



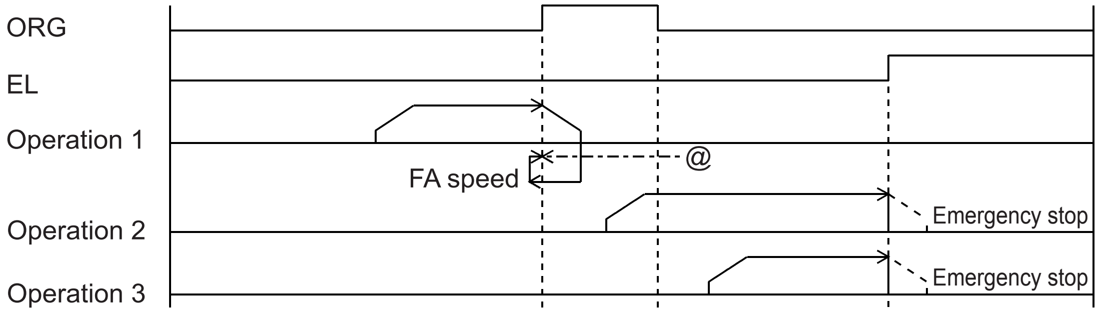
Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for the origin stopping position.

6-4-1-2. Origin return operation 1 (ORM=0001)

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

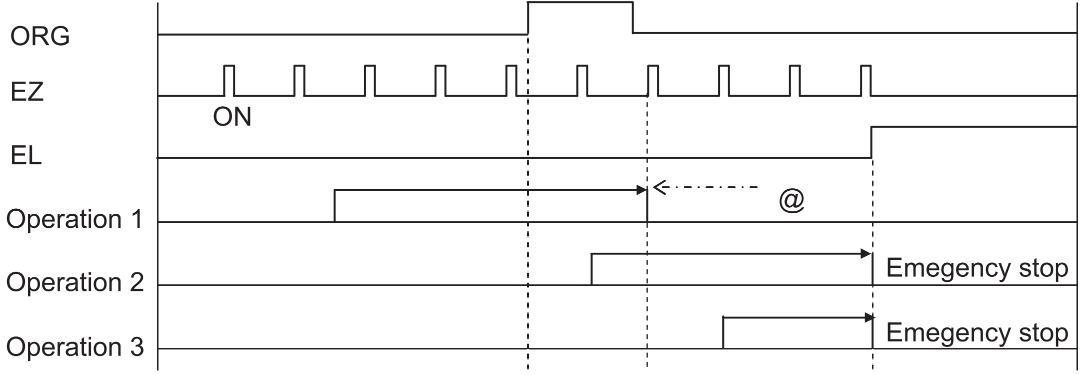


■ High speed operation <Sensor: EL, ORG>

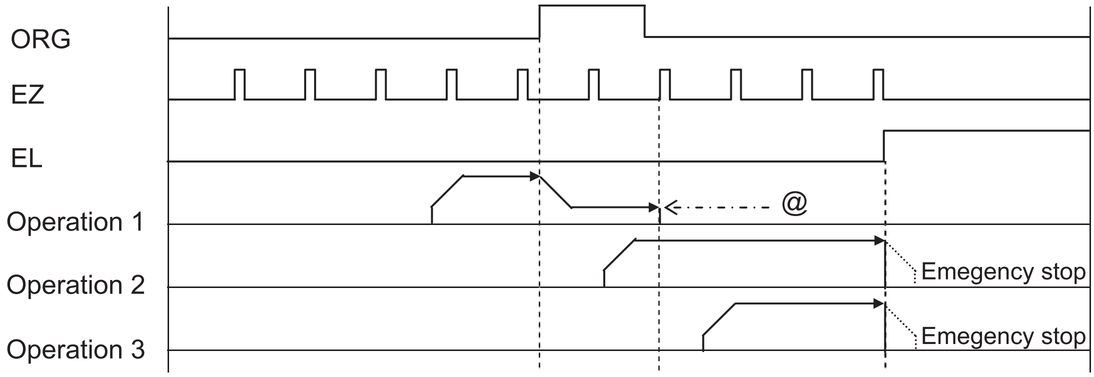


6-4-1-3. Origin return operation 2 (ORM = 0010)

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



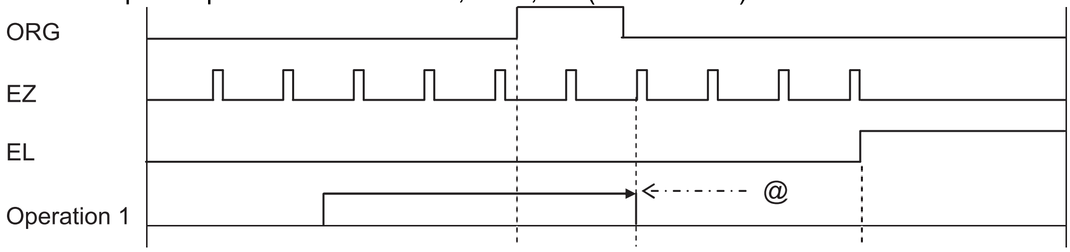
■ High speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>



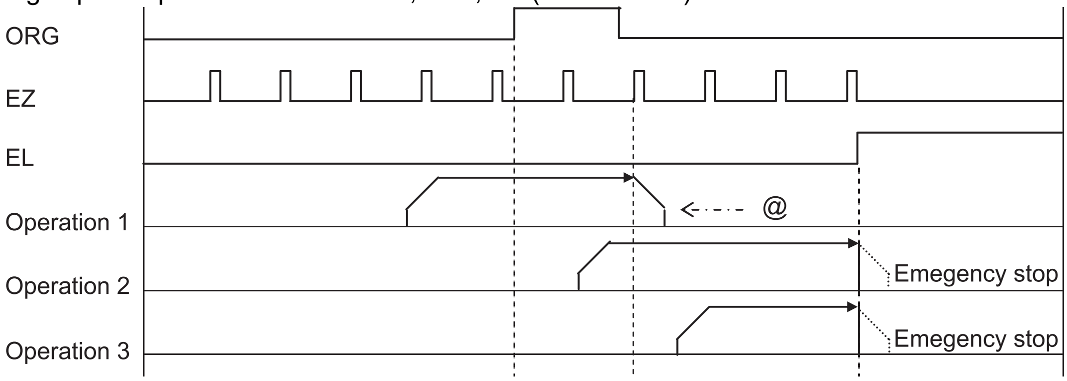
Note: Positions marked with "@" reflect ERC signal output timing when "Automatically output an ERC signal" is selected for the origin stopping position.

6-4-1-4. Origin return operation 3 (ORM = 0011)

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

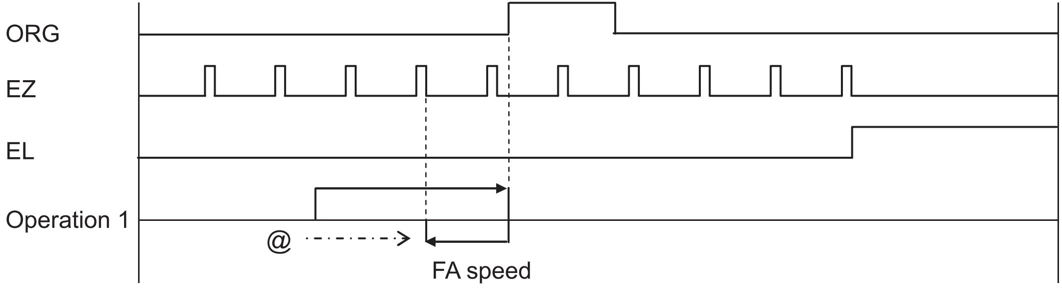


■ High speed operation <Sensor: EL,ORG, EZ (EZD = 0001)>

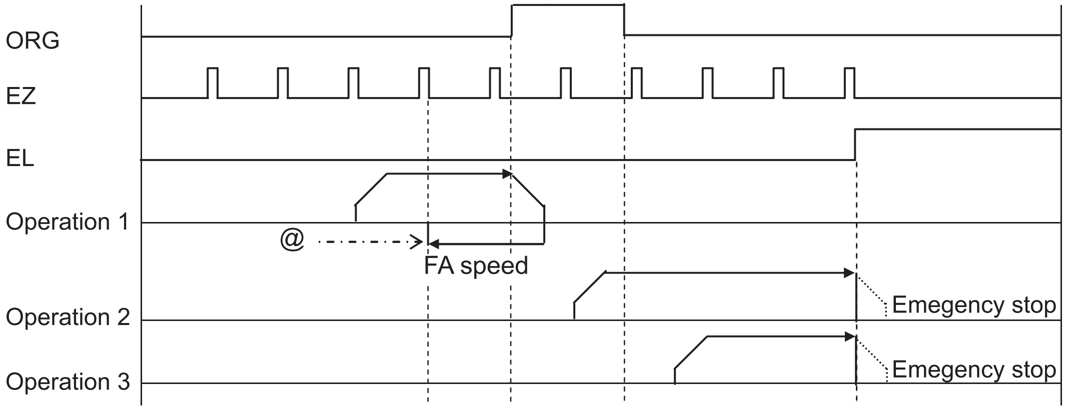


6-4-1-5. Origin return operation 4 (ORM = 0100)

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



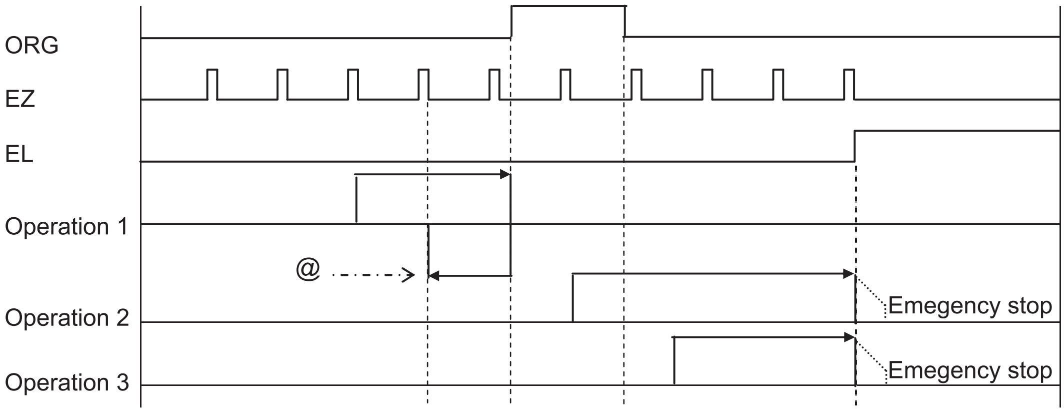
■ High speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>



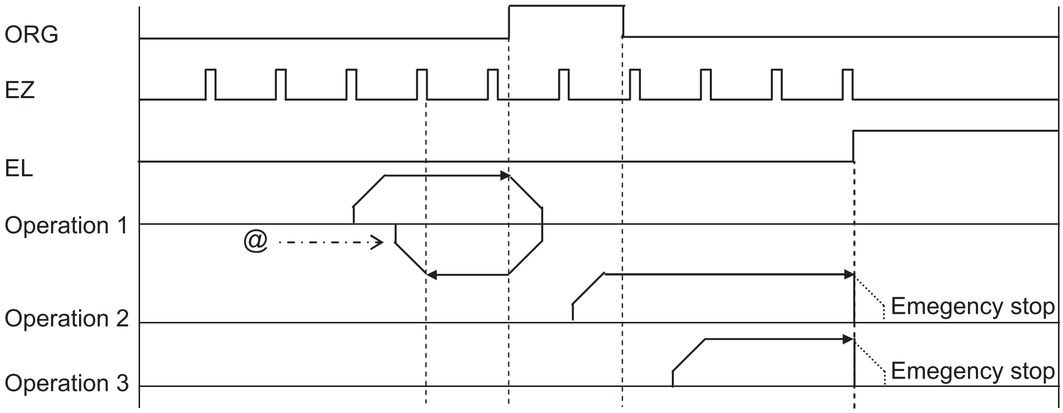
Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for the origin stopping position.

6-4-1-6. Origin return operation 5 (ORM = 0101)

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

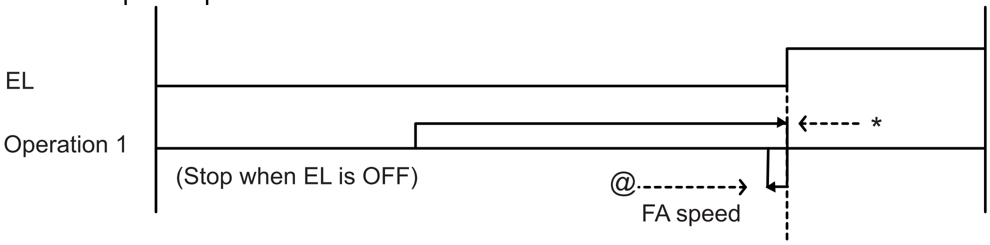


■ High speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>

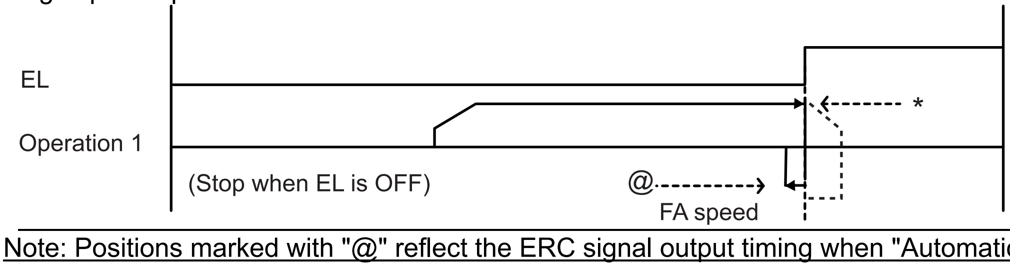


6-4-1-7. Origin return operation 6 (ORM = 0110)

□ Constant speed operation <Sensor: EL>



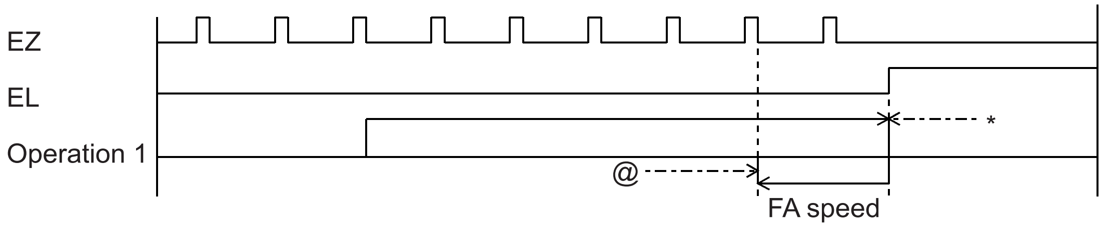
■ High speed operation <Sensor: EL>



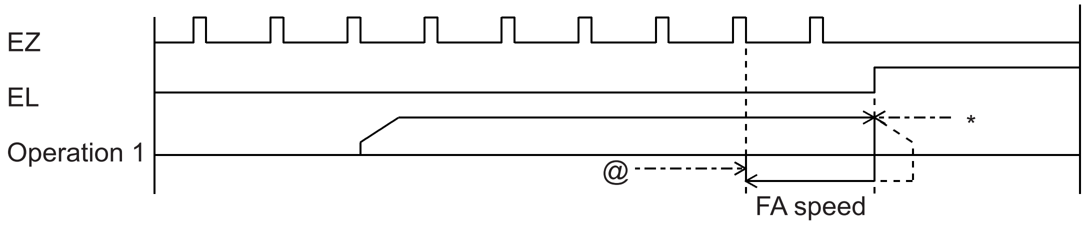
Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for the origin stopping position.
Also, when EROE (bit 10) is 1 in the RENV1 register and ELM (bit 3) is 0, the LSI will output an ERC signal at positions marked with an asterisk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

6-4-1-8. Origin return operation 7 (ORM = 0111)

□ Constant speed operation <Sensor: EL, EZ (EZD = 0001)>

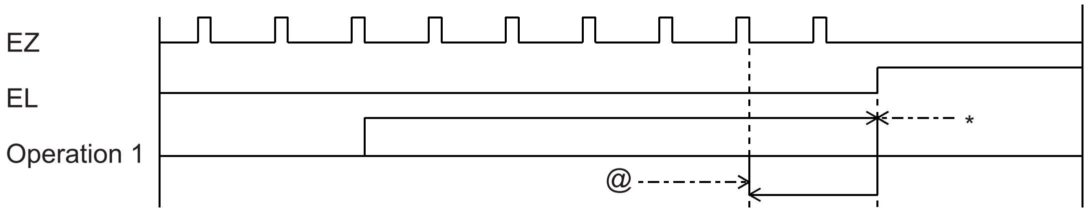


■ High speed operation <Sensor: EL, EZ (EZD = 0001)>

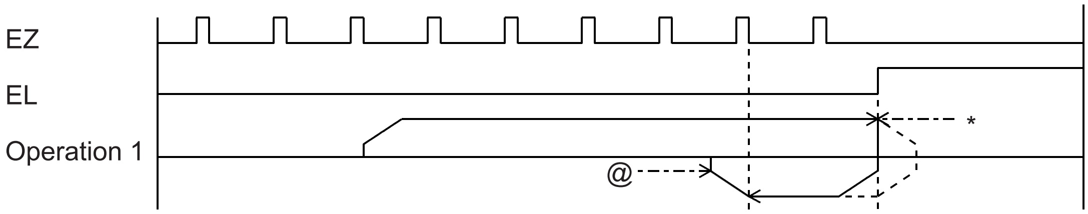


6-4-1-9. Origin return operation 8 (ORM=1000)

□ Constant speed operation <Sensor: EL, EZ (EZD = 0001)>

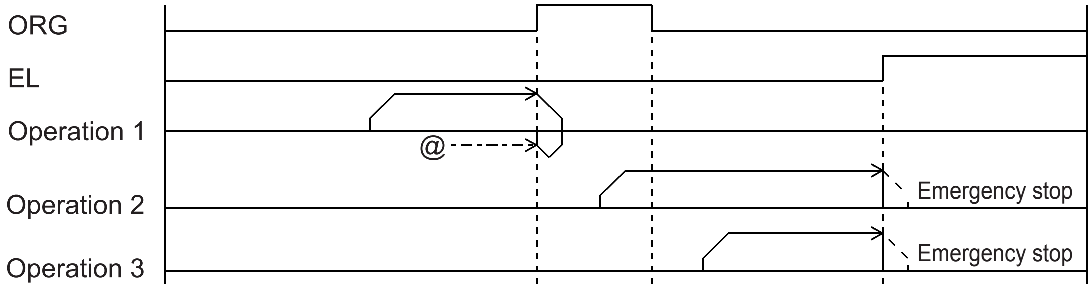


■ High speed operation <Sensor: EL, EZ (EZD = 0001)>



6-4-1-10. Origin return operation 9 (ORM = 1001)

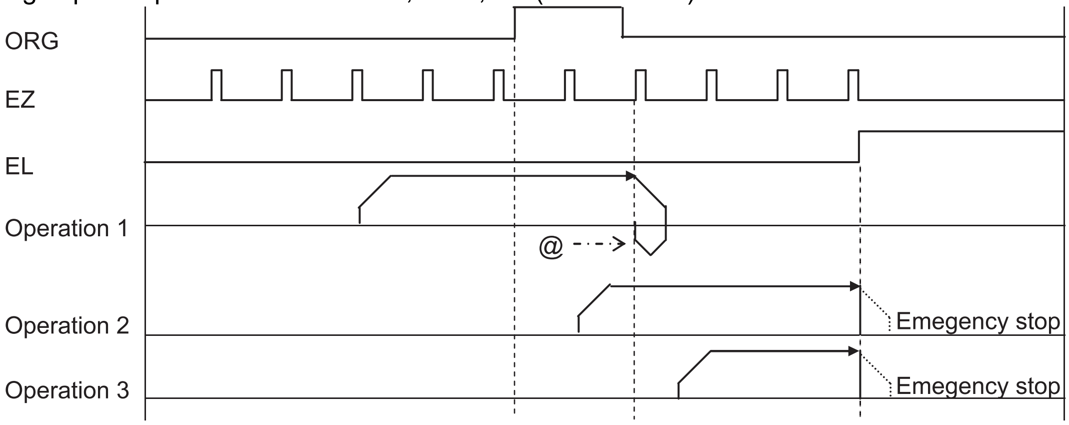
■ High speed operation <Sensor: EL, ORG)>



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

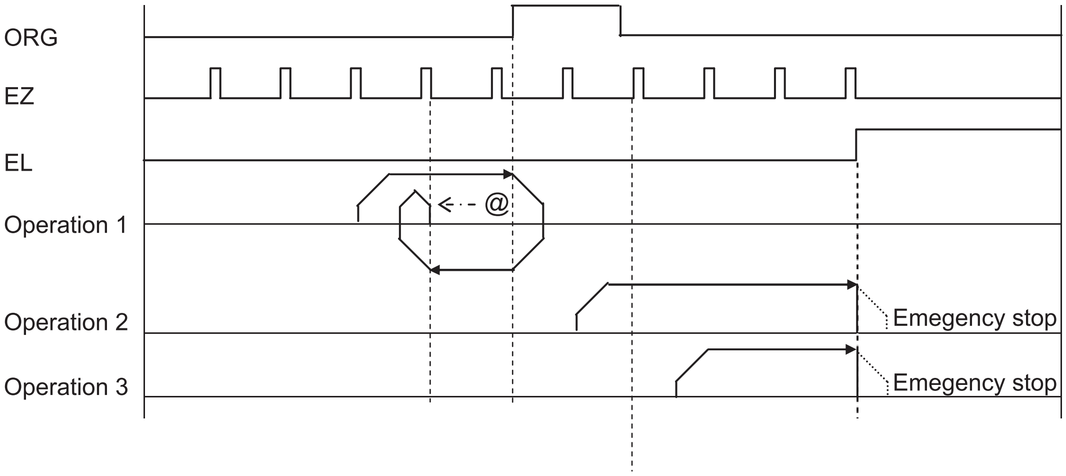
6-4-1-11. Origin return operation 10 (ORM = 1010)

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



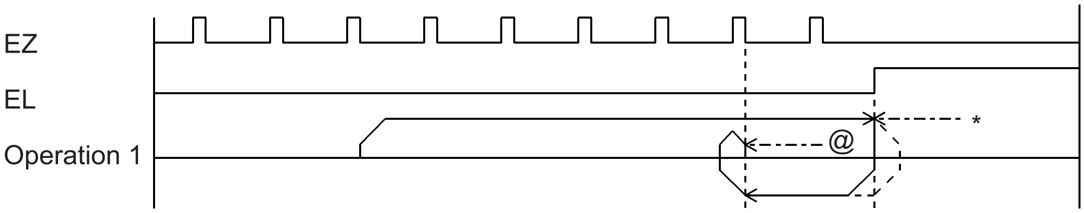
6-4-1-12. Origin return operation 11 (ORM = 1011)

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



6-4-1-13. Origin return operation 12 (ORM = 1100)

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



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

6-4-2. Leaving the origin position operations

After writing a start command, the axis will leave the origin position (when the ORG input turns ON). Make sure to use the "Constant speed start command (0050h, 0051h)" when leaving the origin position. When you write a start command while the ORG input is OFF, the LSI will stop the movement on the axis as a normal stop, without outputting pulses.
Since the ORG input status is sampled when outputting pulses, if the PCL device (G9003) starts at constant speed while the ORG signal is ON, it will stop operation after outputting one pulse, since the ORG input is turned OFF. (Normal stop)

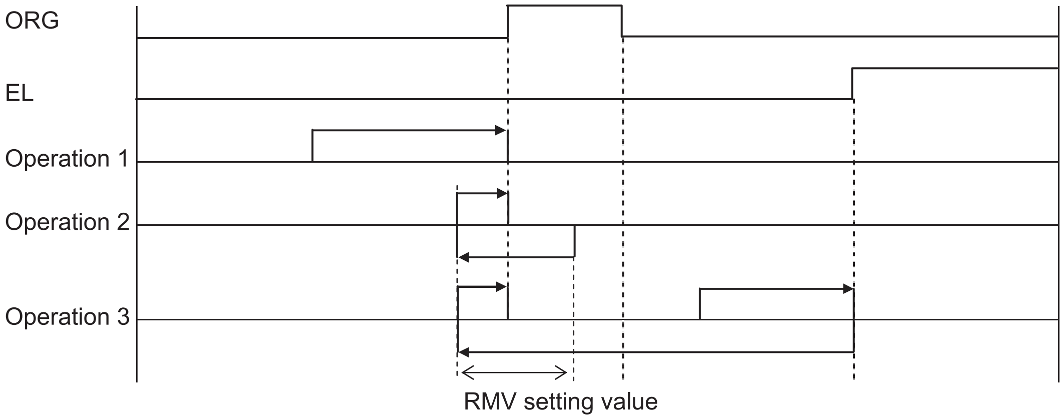
MOD: 12h Leave the origin position in the positive direction
1Ah Leave the origin position in the negative direction

6-4-3. Origin search operation

This mode is used to add functions to a origin return operation. It consists of the following possibilities.
1) A "Origin return operation" is made in the opposite direction to the one specified.
2) A "Leaving the origin position using positioning operations" is executed in the opposite direction to the one specified.
3) A "Origin return operation" is executed in the specified direction.
Operation 1: If the ORG input is turned ON after starting, movement on the axis will stop normally.
Operation 2: If the ORG input is already turned ON when starting, the axis will leave the origin position using positioning operations, and then begin a "origin return operation."
Operation 3: If movement on the axis is stopped by an EL signal while operating in the specified direction, the axis will execute a "origin return operation (ORM = 0000)" and a "leaving the origin position by positioning" in the opposite direction. Then it will execute an "origin return operation" in the specified direction.
When "leaving the origin position by positioning," the axis will repeat the positioning operation for the number of pulses specified in the RMV (target position) register, until the origin position has been left. Enter a positive number (1 to 134,217,727) in the RMV register.
MOD: 15h Origin search operation in the positive direction
1Dh Origin search operation in the negative direction

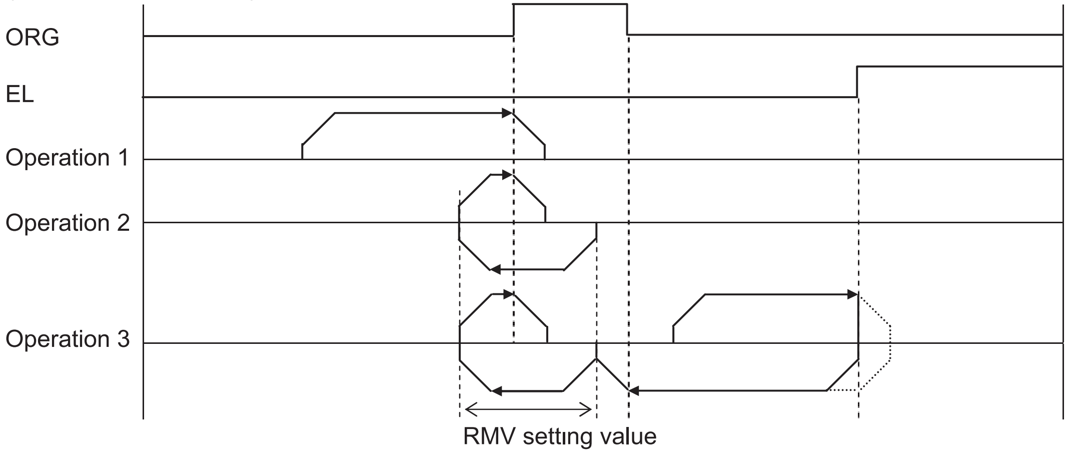
6-4-3-1. Origin return operation 0 (ORM=0000)

□ Constant speed operation <Sensor: EL, ORG>



■ High speed operation <Sensor: EL, ORG>

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



6-5. EL or SL operation mode

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

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

To specify the ±EL input signal, set the input logic using the ELL input terminal. Select the operation type (immediate stop / deceleration stop) when the input from that terminal is ON in the RENV1 (Environment setting 1) register. The status of the terminal can be monitored using the RSTS (extension status) register. For details about setting the SL (software limit), see section 8-11-2, "Software limit function."

Select the ±EL signal input logic <ELL input terminal> L: Positive logic input H: Negative logic input	
Select the stop method to use when the ±EL signal is turned ON <Set ELM (bit 3) in RENV1> 0: Stop immediately when the ±EL signal turns ON. 1: Decelerates and stops when the ±EL signal turns ON.	[RENV1] (WRITE) 7 0 - - - - n - - -

Reading the ±EL signal <SPEL (bit 6), SMEL (bit 7) in RSTS>
SPEL=0: Turn OFF +EL signal SPEL=1: Turn ON +EL signal
SMEL=0: Turn OFF -EL signal SMEL=1: Turn ON -EL signal
	[RSTS] (READ)
	7 0
	n n - - - - - -

Setting the ±EL input filter <Set the FLTR (bit 25) in RENV1 > 0: Apply a filter to the ±EL, ORG input. After applying a filter, signals shorter than 4 µsec will be ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

6-5-1. Feed until reaching an EL or SL position

This mode is used to continue feeding until the EL or SL (soft limit) signal is turned ON and then the operation stops normally.
When a start command is written on the position where the EL or SL signal is turned ON, the LSI will not output pulses and it will stop the axis normally. When a start command is written to the axis while the EL and SL signals are OFF, the axis will stop when the EL or SL signal is turned ON. (Normal stop.)
MOD: 20h Feed until reaching the +EL or +SL position.
28h Feed until reaching the -EL or -SL position.

6-5-2. Leaving an EL or SL position

This mode is used to continue feeding until the EL or SL (software limit) signal is turned OFF.
When a start command is written on the position where the EL and SL signals are turned OFF, the LSI will not output pulses and it will stop the axis normally.
When starting an operation while the EL input or SL signal is ON, the PCL will stop operation normally when both the EL input and SL signal are OFF.
MOD: 22h Leave from a -EL or -SL position
2Ah Leave from a + EL or +SL position

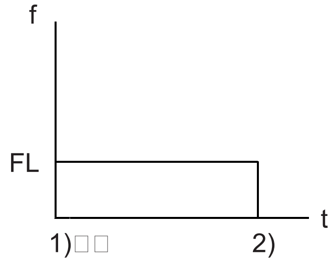
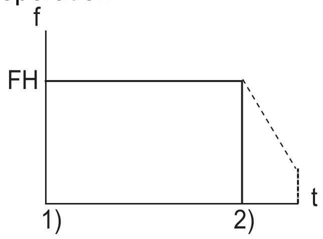
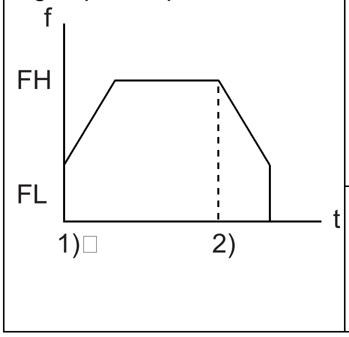
6-6. EZ count operation mode

This mode is used to count EZ signal of the number (EZD set value +1) written into the RENV3 register.
MOD: 24h Feed until the EZ count is complete in positive direction.
2Ch Feed until the EZ count is complete in negative direction.
After a start command is written, the axis stops immediately (or decelerates and stops when feeding at high speed) after the EZ count equals the number stored in the register.
The EZ count can be set from 1 to 16.
Use the constant speed start command (0050h, 0051h) for this operation. When the high speed start command is used, the axis will start decelerating and stop when the EZ signal turns ON, so that the motion on the axis overruns the EZ position.
Specify logical input for the EZ signal in the RENV2 (environment setting 2) register, and the EZ number to count to in the RENV3 (environment setting 3) register. The terminal status can be monitored by reading the RSTS (extension status) register.

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

7. Speed patterns

7-1. Speed patterns

Speed pattern	Continuous mode	Positioning operation mode
FL constant speed operation  	1) Write an FL constant speed start command (0050h). 2) Stop feeding by writing an immediate stop (0049h) or deceleration stop (004Ah) command.	1) Write an FL constant speed start command (0050h). 2) Stop feeding when the positioning counter reaches zero, or by writing an immediate stop (0049h) or deceleration stop (004Ah) command.
FH constant speed operation  	1) Write an FH constant speed start command (0051h). 2) Stop feeding by writing an immediate stop command (0049h).	1) Write an FH constant speed start command (0051h). 2) Stop feeding when the positioning counter reaches zero, or by writing an immediate stop (0049h) command.
	* When the deceleration stop command (004Ah) is written to the register, the PCL device (G9003) starts deceleration.	
High speed operation  	1) Write high speed command 2 (0053h). 2) Start deceleration by writing a deceleration stop command (004Ah).	1) Write high speed start command (0053h). 2) Start deceleration when a ramping-down point is reached or by writing a deceleration stop command (004Ah). * When the automatic ramp down point setting is set to manual (MSDP = 1 in the RMD), and the ramp down point value (RDP) is set to "0," the PCL device (G9003) immediately stops the motor.
	* When the deceleration stop command (0049h) is written to the register.	

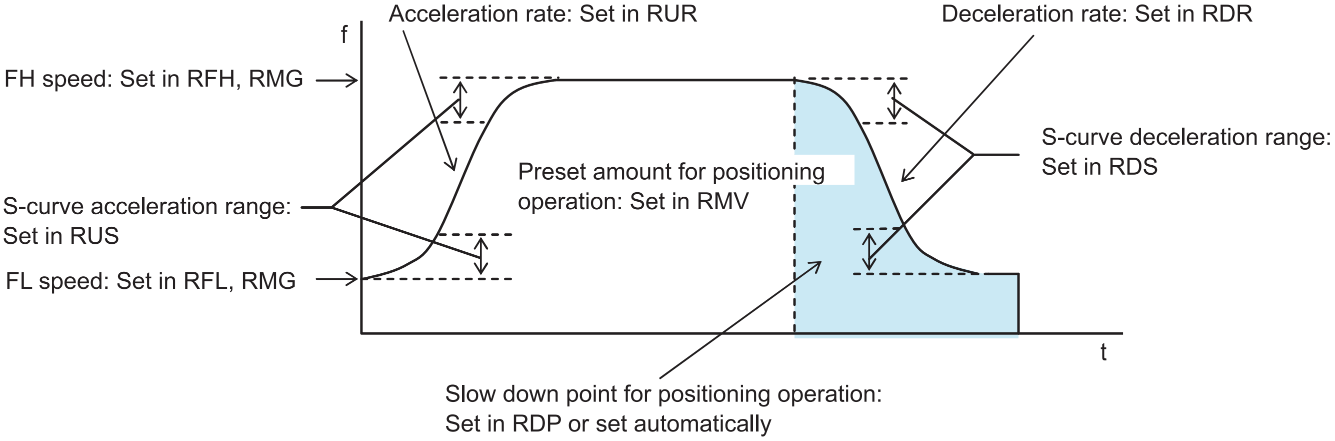
7-2. Speed pattern settings

Specify the speed pattern using the registers shown in the table below.
If the next register setting is the same as the current value, there is no need to write to the register again.
Please note that with some registers, a setting of "0" may be outside the allowable range.

Pre-register	Description	Bit length setting range	Setting range	R/W
RMV	Positioning amount	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	R/W
RFL	Initial speed	17	1 to 100,000 (186A0h) Note2	R/W
RFH	Operation speed	17	1 to 100,000 (186A0h) Note2	R/W
RUR	Acceleration rate	16	1 to 65,535 (0FFFFh)	R/W
RDR	Deceleration rate Note 1	16	0 to 65,535 (0FFFFh)	R/W
RMG	Speed magnification rate	11	2 to 2,047 (7FFh)	R/W
RDP	Ramping-down point	24	0 to 16,777,215 (0FFFFFFh)	R/W
RUS	S-curve acceleration range	16	0 to 50,000 (0C350h) Note 3	R/W
RDS	S-curve deceleration range	16	0 to 50,000 (0C350h) Note 3	R/W
RFA	Feed amount correction speed	17	0 to 100,00 (186A0h) Note 2	R/W

Note 1: If RDR is set to zero, the deceleration rate will be the value set in the RUR.
Note 2: All values from 186A0h to 1FFFFh will be treated as 186A0h.
Note 3: All values from 0C350h to 0FFFFh will be treated as 0C350h.

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


Acceleration rate: Set in RUR
Deceleration rate: Set in RDR
f
FH speed: Set in RFH, RMG
S-curve deceleration range:
Set in RDS
Preset amount for positioning
operation: Set in RMV
S-curve acceleration range:
Set in RUS
FL speed: Set in RFL, RMG
t
Slow down point for positioning operation:
Set in RDP or set automatically

[The number "40,000,000", used in the formulas here, is the frequency in Hz of the internal reference clock.]

- RMV: register (28 bits)

  This register is used to set the target position for positioning operations.
  The details for setting it may vary with the operation mode selected.
  Setting rage: -134,217,728 to +134,217,727
  By changing the RMV register during operation, you can override the feed amount.

- RFL: FL speed setting register (17 bits)

  This register is used to set the initial speed (and stopping speed) in a high-speed operation (with acceleration).
  Specify the speed for FL constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 100,000 (186A0h). All values from 100,000 to 131,071 (186A0h to 1FFFFh) will be treated as 100,000.
  The actual operation speed will be obtained from a calculation using the RMG value.

$$\text{FL speed [pps]} = \text{RFL} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

- RFH: FH speed setting register (17 bits)

  This register is used to set the operation speed.
  The speed can be changed in the middle of an operation by changing the RFH register setting.
  Specify the speed for FH constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 100,000 (186A0h). All values from 100,000 to 131,071 (186A0h to 1FFFFh) will be treated as 100,000.
  When used for high speed operations (acceleration/deceleration operations), specify a value larger than RFL.
  The actual operation speed will be obtained from a calculation using the RMG value.

$$\text{FH speed [pps]} = \text{RFH} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

- RUR: Acceleration rate setting register (16 bits)

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

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

$$\text{Acceleration time [s]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1) \times 8}{40{,}000{,}000}$$

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

$$\text{Acceleration time [s]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1) \times 16}{40{,}000{,}000}$$

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

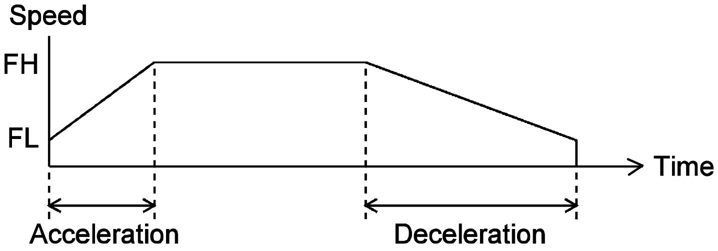
$$\text{Acceleration time [s]} = \frac{(\text{RFH} - \text{RFL} + 2 \times \text{RUS}) \times (\text{RUR} + 1) \times 8}{40{,}000{,}000}$$

◆ RDR: Deceleration rate setting register (16 bits)

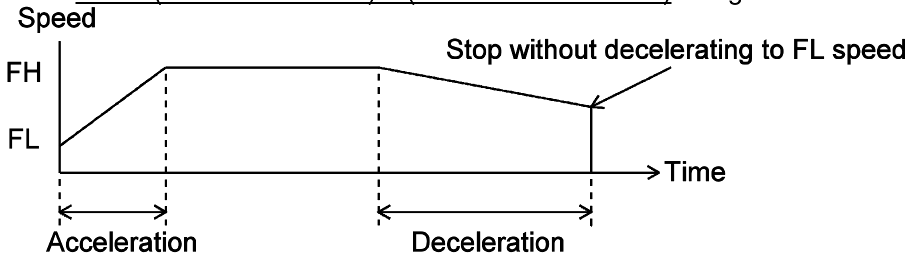
This register is used to set the deceleration rate.
Normally, specify the deceleration characteristics for high speed operations (acceleration/deceleration operations) in the range of 1 to 65,535 (0FFFFh).
Even if the ramping-down point is set to automatic (MSDP = 0 in the RMD register), the value placed in the RDR register will be used as the deceleration rate.
However, when RDR = 0, the deceleration rate will be the value placed in the RUR.

When you want to set the auto ramp-down point, adjust it so that (deceleration time) ≦ (acceleration time x 2).
If the (deceleration time) > (acceleration time x 2), the motor may not be able to decelerate to the FL speed when stopping. In this case, select the manual ramp-down point setting method (MSDP = 1 in the RMD register).

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



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



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

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

$$\text{Deceleration time [s]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RDR} + 1) \times 8}{40{,}000{,}000}$$

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

$$\text{Deceleration time [s]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RDR} + 1) \times 16}{40{,}000{,}000}$$

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

$$\text{Deceleration time [s]} = \frac{(\text{RFH} - \text{RFL} + 2 \times \text{RDS}) \times (\text{RDR} + 1) \times 8}{40{,}000{,}000}$$

- RMG: Magnification rate register (11 bits)

This register is used to set the speed multiplication rate.
Specify the relationship between the RFL, RFH and RFA settings and the speed, in the range of 2 to 2,047 (07FFh). As the magnification rate is increased, the speed setting units will tend to be approximations. Normally set the magnification rate as low as possible.
The relationship between the value entered and the magnification rate is as follows.

$$\text{Magnification rate} = \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

[Magnification rate setting example, when the reference clock =40 MHz] (Output speed unit: pps)

Setting	Magnification rate	Output speed range	Setting	Magnification rate	Output speed range
1999 (7CFh)	0.1	0.1 to 10,000.0	39 (27h)	5	5 to 500,000
999 (3E7h)	0.2	0.2 to 20,000.0	19 (13h)	10	10 to 1,000,000
399 (18Fh)	0.5	0.5 to 50,000.0	9 (09h)	20	20 to 2,000,000
199 (0C7h)	1	1 to 100,000	3 (3h)	50	50 to 5,000,000
99 (63h)	2	2 to 200,000	2 (2h)	66.6	66.6.. to 6,666,666.6..

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

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

<When set to manual (MSDP=1 in the RMD register)>
Set the number of pulses at which to start deceleration, in the range of 0 to16,777,215 (0FFFFFFh).
The optimum value for the ramping-down point can be calculated as shown in the equation below.

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

$$\text{Optimum value [Number of pulses]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 50{,}000}$$

However, the optimum value for a triangle start, without changing the value in the RFH register while turning OFF the FH correction function (MADJ = 1 in the RMD register) will be calculated as shown the equation below.
(When using idling control, modify the value for RMV in the equation below by deducting the number of idling pulses from the value placed in the RMV register. The number of idling pulses will be "1 to 6" when IDL = 0 to 7 in RENV2.)

$$\text{Optimum value [Number of pulses]} = \frac{\text{RMV} \times (\text{RDR} + 1)}{\text{RUR} + \text{RDR} + 2}$$

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

$$\text{Optimum value [Number of pulses]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1) \times 2}{(\text{RMG} + 1) \times 50{,}000}$$

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

$$\text{Optimum value [Number of pulses]} = \frac{(\text{RFH} + \text{RFL}) \times (\text{RFH} - \text{RFL} + 2 \times \text{RDS}) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 50{,}000}$$

Start deceleration at the point when the (positioning counter value) ≦ (RDP set value).

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

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

When the offset value is a positive number, the axis will start deceleration at an earlier stage and will feed at the FL speed after decelerating. When a negative number is entered, the deceleration start timing will be delayed. If the offset is not required, set to zero.

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

♦ RUS: S-curve acceleration range register (16 bits)

This register is used to specify the S-curve range in an S-curve acceleration.
Specify the S-curve acceleration range for S-curve acceleration/deceleration operations in the range of 1 to 50,000 (0C350h).
Settings from 50,000 to 65,535 (0C350h to 0FFFFh) will all be treated as 50,000.
The S-curve acceleration range S_{SU} will be calculated from the value placed in RMG.

$$S_{SU}\ [\text{pps}] = \text{RU} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

In other words, speeds between the FL speed and (FL speed + S_{SU}), and between (FH speed - S_{SU}) and the FH speed, will be S-curve acceleration operations. Intermediate speeds will use linear acceleration.
However, if zero is specified, "(RFH - RFL)/2" will be used for internal calculations, and the operation will be an S-curve acceleration without a linear component.
If the minimum value "1" is specified, the PCL device (G9003) will operate with nearly linear acceleration.
If a larger value than "(RFH - RFL) / 2" is specified, the motor will not reach the maximum acceleration speed and the acceleration time will be different from the calculated value. Therefore, enter a value smaller than "(RFH - RFL) / 2."

♦ RDS: S-curve deceleration range setting register (16 bits)

This register is used to specify the S-curve range in an S-curve deceleration
Specify the S-curve deceleration range for S-curve acceleration/deceleration operations in the range of 1 to 50,000 (0C350h).
Settings from 50,000 to 65,535 (0C350h to 0FFFFh) will all be treated as 50,000.
The S-curve acceleration range S_{SD} will be calculated from the value placed in RMG.

$$S_{SD}\ [\text{pps}] = \text{RDS} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

In other words, speeds between the FL speed and (FL speed + S_{SD}), and between (FH speed - S_{SD}) and the FH speed, will be S-curve deceleration operations. Intermediate speeds will use linear deceleration.
However, if zero is specified, "(RFH - RFL)/2" will be used for internal calculations, and the operation will be an S-curve deceleration without a linear component.
If the minimum value "1" is specified, the PCL device (G9003) operates with nearly linear acceleration.
If a larger value than "(RFH - RFL) / 2" is specified, the motor will not reach the maximum deceleration speed and the deceleration time will be different from the calculated value. Therefore, enter a value smaller than "(RFH - RFL) / 2."

- RFA: FL speed setting register (17bits)

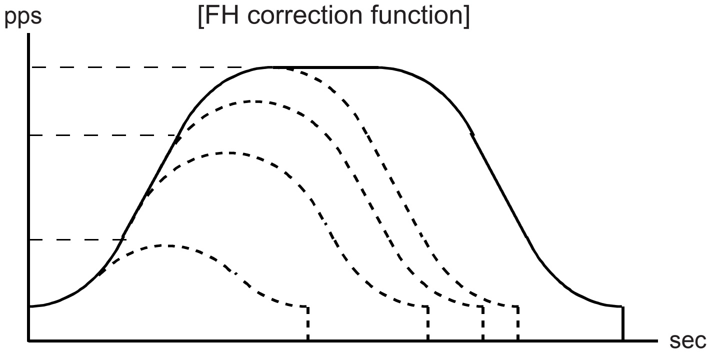
This register is used to set the constant speed during backlash correction.
Set the correction speed feed amount for use during backlash within the range of 1 to 100,000 (186A0h).
Numbers from 100,000 to 131,071 (186A0h to 1FFFFh) will all be treated as 100,000.
The actual operating speed will be the value calculated using the RMG setting.
This register value is also used for the reverse constant speed during zero position return.

$$\text{FA speed [pps]} = \text{RFA} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

7-3. Manual FH correction

When the FH correction function is turned ON (MADJ = 0 in the RMD register), and when the feed amount is too small for a normal acceleration and deceleration operation, the LSI will automatically lower the FH speed to eliminate triangle driving.
However, if values in the RUR and RDR registers are set so that the (deceleration time) > (acceleration time x 2), do not use the FH correction function.

In order to eliminate triangle driving without using the FH correction function (MADJ = 1 in the RMD register), lower the FH speed before starting the acceleration/deceleration operation.
When using idling control, enter a value for RMV in the equation below after deducting the number of idling pulses. The number of idling pulses will be 1 to 6 when IDL = 2 to 7 in RENV2.



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

< To execute FH correction manually>

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

$$\mathrm{RMV} \leqq \frac{(\mathrm{RFH}^2 - \mathrm{RFL}^2) \times (\mathrm{RUR} + \mathrm{RDR} + 2)}{(\mathrm{RMG} + 1) \times 50000}$$

$$\mathrm{RFH} \leqq \sqrt{\frac{(\mathrm{RMG} + 1) \times 50000 \times \mathrm{RMV}}{\mathrm{RUR} + \mathrm{RDR} + 2} + \mathrm{RFL}^2}$$

2) S-curve acceleration without linear acceleration (MSMD=1 in the RMD register and the RVS register = 0, RDS registers = 0)
When

$$\mathrm{RMV} \leqq \frac{(\mathrm{RFH}^2 - \mathrm{RFL}^2) \times (\mathrm{RUR} + \mathrm{RDR} + 2) \times 2}{(\mathrm{RMG} + 1) \times 50000}$$

$$\mathrm{RFH} \leqq \sqrt{\frac{(\mathrm{RMG} + 1) \times 50000 \times \mathrm{RMV}}{(\mathrm{RUR} + \mathrm{RDR} + 2) \times 2} + \mathrm{RFL}^2}$$

3) S-curve acceleration/deceleration with linear acceleration/deceleration (MSMD = 1 in the RMD register and the RUS register > 0, RDS register > 0)
(3)-1. When RUS = RDS
(i) Set up a small linear acceleration range
When

$$\mathrm{RMV} \leqq \frac{(\mathrm{RFH} + \mathrm{RFL}) \times (\mathrm{RFH} - \mathrm{RFL} + 2 \times \mathrm{RUS}) \times (\mathrm{RUR} + \mathrm{RDR} + 2)}{(\mathrm{RMG} + 1) \times 50000} \quad \text{and}$$

$$\mathrm{RMV} > \frac{(\mathrm{RUS} + \mathrm{RFL}) \times \mathrm{RUS} \times (\mathrm{RUR} + \mathrm{RDR} + 2) \times 8}{(\mathrm{RMG} + 1) \times 50000}$$

$$\mathrm{RFH} \leqq -\ \mathrm{RSU} + \sqrt{(\mathrm{RUS} - \mathrm{RFL})^2 + \frac{(\mathrm{RMG} + 1) \times 50000 \times \mathrm{RMV}}{(\mathrm{RUR} + \mathrm{RDR} + 2)}}$$

(ii) Eliminate the linear acceleration/deceleration range

When $\mathrm{RMV} \leqq \dfrac{(\mathrm{RUS} + \mathrm{RFL}) \times \mathrm{RUS} \times (\mathrm{RUR} + \mathrm{RDR} + 2) \times 8}{(\mathrm{RMG} + 1) \times 50000}$

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

$$\mathrm{RFH} \leqq \sqrt{\frac{(\mathrm{RMG} + 1) \times 50000 \times \mathrm{RMV}}{(\mathrm{RUR} + \mathrm{RDR} + 2) \times 2} + \mathrm{RFL}^2}$$

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

(3)-2. When RUS < RDS

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

$$\mathrm{RMV} \leqq \frac{(\mathrm{RFH}+\mathrm{RFL}) \times \{(\mathrm{RFH}-\mathrm{RFL}) \times (\mathrm{RUR} + \mathrm{RDR} + 2) + 2 \times \mathrm{RUS} \times (\mathrm{RUR}+1) + 2 \times \mathrm{RDS} \times (\mathrm{RDR} + 1)\}}{(\mathrm{RMG} + 1) \times 50000}$$

and

$$\mathrm{RMV} > \frac{(\mathrm{RDS}+\mathrm{PRFL}) \times \{\mathrm{RDS} \times (\mathrm{RUR} + 2 \times \mathrm{RDR} + 3) + \mathrm{RUS} \times (\mathrm{RUR} + 1)\} \times 4}{(\mathrm{RMG} + 1) \times 50000},$$

$$\mathrm{RFH} \leqq \frac{-A + \sqrt{A^2 + B}}{\mathrm{RUR} + \mathrm{RDR} + 2}$$

However, $A = \mathrm{RUS} \times (\mathrm{RUR} + 1) + \mathrm{RDS} \times (\mathrm{RDR} + 1)$
$B= \{(\mathrm{RMG} + 1) \times 50000 \times \mathrm{RMV} - 2 \times A \times \mathrm{RFL} + (\mathrm{RUR} + \mathrm{RDR} + 2) \times \mathrm{RFL}^2\} \times (\mathrm{RUR} + \mathrm{RDR} + 2)$

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

$$\mathrm{RMV} \leqq \frac{(\mathrm{RDS} + \mathrm{RFL}) \times \{\mathrm{RDS} \times (\mathrm{RUR} + 2 \times \mathrm{RDR} + 3)\} + \mathrm{RUS} \times (\mathrm{RUR} +1)\} \times 4}{(\mathrm{RMG} + 1) \times 50000} \quad \text{and}$$

$$\mathrm{RMV} > \frac{(\mathrm{RUS} + \mathrm{RFL}) \times \mathrm{RUS} \times (\mathrm{RUR} + \mathrm{RDR} + 2) \times 8}{(\mathrm{RMG} + 1) \times 50000},$$

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

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

However, $A = \mathrm{RUS} \times (\mathrm{RUR} + 1)$,
$B= \{(\mathrm{RMG} + 1) \times 50000 \times \mathrm{RMV} - 2 \times A \times \mathrm{RFL} + (\mathrm{RUR} + 2 \times \mathrm{RDR} + 3) \times \mathrm{RFL}^2\} \times (\mathrm{RUR} + 2 \times \mathrm{RDR} + 3)$

(iii) Eliminate the linear acceleration/deceleration range

$$\text{When } \mathrm{RMV} \leqq \frac{(\mathrm{RUS} + \mathrm{RFL}) \times \mathrm{RUS} \times (\mathrm{RUR} + \mathrm{RDR} + 2) \times 8}{(\mathrm{RMG} + 1) \times 50000}$$

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

$$\mathrm{RFH} \leqq \sqrt{\frac{(\mathrm{RMG} + 1) \times 50000 \times \mathrm{RMV}}{(\mathrm{RUR} + \mathrm{RDR} + 2) \times 2} + \mathrm{RFL}^2}$$

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

(3)-3. When RUS>RDS

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

When

$$\mathrm{RMV} \leqq \frac{(\mathrm{RFH} + \mathrm{RFL}) \times \{(\mathrm{RFH} - \mathrm{RFL}) \times (\mathrm{RUR} + \mathrm{RDR} + 2) + 2 \times \mathrm{RUS} \times (\mathrm{RUR} + 1) + 2 \times \mathrm{RDS} \times (\mathrm{RDR} + 1)\}}{(\mathrm{RMG} + 1) \times 50000}$$

and

$$\mathrm{RMV} > \frac{(\mathrm{RUS} + \mathrm{RFL}) \times \{\mathrm{RUS} \times (2 \times \mathrm{RUR} + \mathrm{RDR} + 3) + \mathrm{RDS} \times (\mathrm{RDR} + 1) \times 4}{(\mathrm{RMG} + 1) \times 50000},$$

Then,

$$\mathrm{RFH} \leqq \frac{-A + \sqrt{A^2 + B}}{\mathrm{RUR} + \mathrm{RDR} + 2}$$

However, A = RUS x (RUR + 1) + RDS x (RDR + 1),
B= {(RMG + 1) x 50000 x RMV - 2 x A x RFL + (RUR + RDR + 2) x RFL^2} x (RUR + RDR + 2)

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

When

$$\mathrm{RMV} \leqq \frac{(\mathrm{RUS} + \mathrm{RFL}) \times \{\mathrm{RUS} \times (2 \times \mathrm{RUR} + \mathrm{RDR} + 3) + \mathrm{RDS} \times (\mathrm{RDR} + 1)\} \times 4}{(\mathrm{RMG} + 1) \times 50000} \quad \text{and}$$

$$\mathrm{RMV} > \frac{(\mathrm{RDS} + \mathrm{RFL}) \times \mathrm{RDS} \times (\mathrm{RUR} + \mathrm{RDR} + 2) \times 8}{(\mathrm{RMG} + 1) \times 50000},$$

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

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

However, A = RDS x (RDR + 1),
B= {(RMG + 1) x 50000x RMV - 2 x A x RFL + (2 x RUR + RDR + 3) x RFL^2} x (2 x RUR + RDR + 3)

(iii) Eliminate the linear acceleration/deceleration range

$$\text{When } \mathrm{RMV} \leqq \frac{(\mathrm{RUS} + \mathrm{RFL}) \times \mathrm{RUS} \times (\mathrm{RUR} + \mathrm{RDR} + 2) \times 8}{(\mathrm{RMG} + 1) \times 50000}$$

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

$$\mathrm{RFH} \leqq \sqrt{\frac{(\mathrm{RMG} + 1) \times 50000 \times \mathrm{RMV}}{(\mathrm{RUR} + \mathrm{RDR} + 2) \times 2} + \mathrm{RFL}^2}$$

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

7-4. Example of setting up an acceleration/deceleration speed pattern

Ex. When the start speed =10 pps, the operation speed =110 kpps, and the accel/decel time = 300 msec,

1) Select the 2x mode for multiplier rate in order to get 110 kpps output
RMG = 99 (63h)

2) Since the 2x mode is selected to get an operation speed 110 kpps,
RFH = 55000 (D6D8h)

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

4) In order to make the acceleration/deceleration time 300 ms, set RUR = 26.275, from the equation for the acceleration time and the RUR value.

$$\text{Acceleration time [s]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1) \times 8}{40{,}000{,}000}$$

$$0.3 = \frac{(55000 - 5) \times (\text{RUR} + 1) \times 8}{40{,}000{,}000}$$

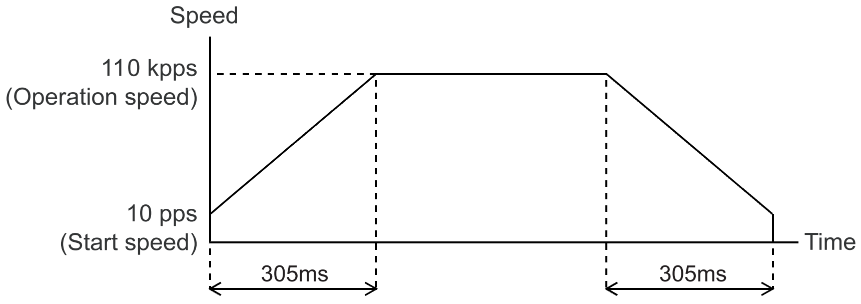
Then,

$$\text{RUR} = \frac{40{,}000{,}000 \times 0.3}{(55000 - 5) \times 8} - 1$$

RUR = 26.275

However, since only integers can be entered for RUR, use "26" or "27." The actual acceleration/deceleration time will be 297 msec if RUR = "26", or 308 msec if RUR = "27."

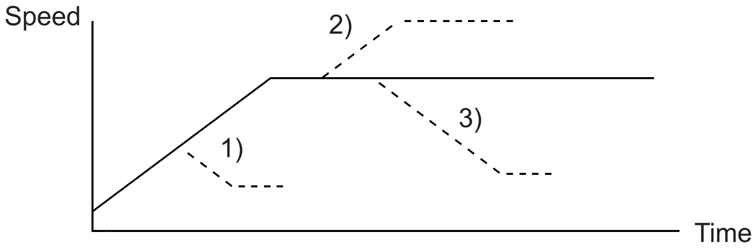
An example of the speed pattern when RUR = 27



7-5. Changing speed patterns while in operation

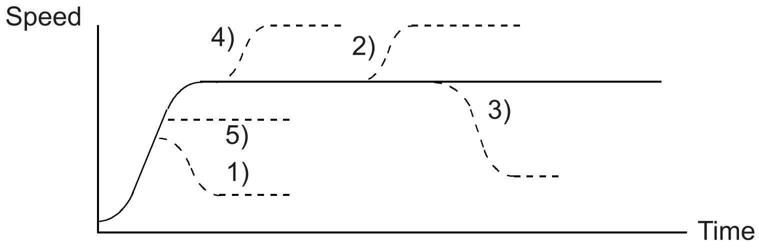
By changing the RFH, RUR, RDR, RUS, or RDS registers during operation, the speed and acceleration can be changed on the fly. However, if the ramping-down point was set to automatic (MSDP = 0 in the RMD register) for the positioning mode, do not change the values for RFL, RUR, RDR, RUS, or RDS. The automatic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

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



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

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



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

8. Description of the functions

8-1. Reset

After turning ON the power, make sure to reset the LSI before beginning to use it.
To reset the LSI, hold the #RST terminal LOW while supplying at least 10 cycles of a reference clock signal.
After a reset, the various portions of the LSI will be configured as follows.

Item	Reset status (initial status)
Internal registers	0
Control command	0
P0 to P7 terminals	Input terminal
#STA, #STP terminals	HIGH
OUT, DIR terminals	HIGH
ERC terminal	HIGH
#BSY/PH1, #FUP/PH2, #FDW/PH3, #MVC/PH4 terminals	HIGH

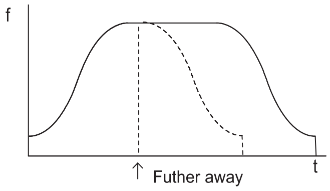
8-2. Position override

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

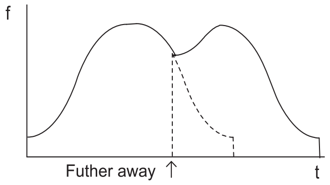
8-2-1. Target position override 1

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

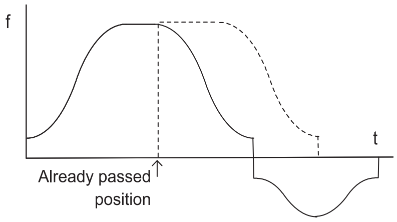
1) If the new target position is further away from the original target position during acceleration or constant speed operation, the axis will maintain the operation using the same speed pattern and it will complete the positioning operation at the position specified in the new data (new RMV value).



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

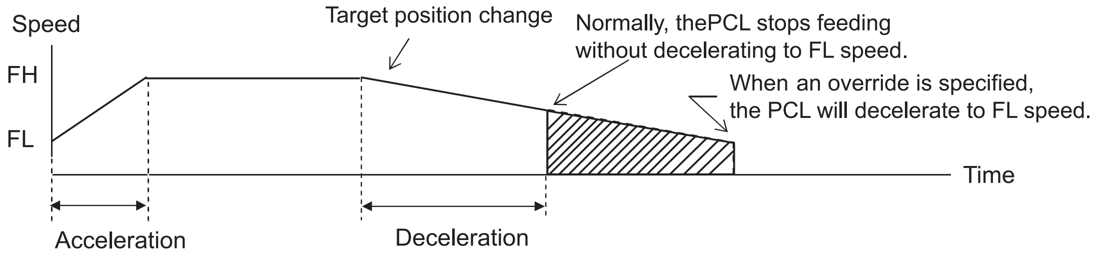


3) If the axis has already passed over the new target position, or the target position is changed to a position that is closer than the original position during deceleration, movement on the axis will decelerate and stop. Then, the movement will reverse and complete the positioning operation at the position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



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

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



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

Note 2: The position override is only effective during operation (FL/FH constant speed operation, during accelerating/decelerating, or during backlash correction).
If the speed override is triggered just before the motor stops, the speed override may not be accepted. If you need to order a speed override just before a stop, you must determine if the PCL device (G9003) can accept the override or not by the rotation position of the motor when you trigger the override.
By using write override (0080h) to the RMV register, you can generate an interrupt when the PCL device (G9003) fails to override. In this case too, you must determine whether the override is accepted or not by the stop position. The cause of the interrupt can be read in the REST (error interrupt cause) register.

The PCL device (G9003) generates an interrupt when an override is written (0080h) to the RMV register while the PCL device (G9003) is stopped. That is, it declares that an error has occurred. If you try to write an override (0080h) to the RMV register before the PCL device (G9003) starts, an interrupt error will also occur.

8-2-2. Target position override 2 (PCS signal)

By making MPCS in the RMD (operation mode) register "1," the PCL (G9003) will perform positioning operations for the amount specified in the RMV register, based on the timing of this command after the operation start (after it starts outputting instruction pulses) or on the "ON" timing of the PCS input signal.
A PCS input signal can change the input logic. The PCS terminal status can be monitored using the RSTS register (extension status).

Setting pulse control using the PCS input <Set MPCS (bit 13) in RMD> 1: Positioning for the number of pulses stored in the RMV, starting from the time at which the PCS input signal is turned ON.	[RMD] (WRITE) 15 8 - - n - - - - -
Setting the PCS input logic <Set PCSL (bit 24) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 31 24 - - - - - - - n
Reading the PCS signal < SPCS (bit 14) in RSTS> 0: Turn OFF PCS 1: Turn ON PCS	[RSTS] (READ) 15 8 n - - - - - - -
PCS substitution input <STAON: Control command> Perform processes that are identical to those performed by supplying a PCS signal.	[Control command] 0028h

8-3. Output pulse control

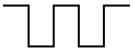
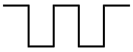
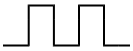
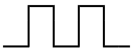
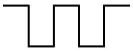
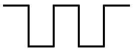
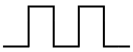
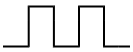
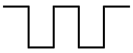
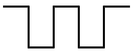
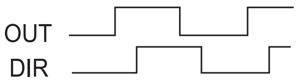
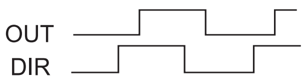
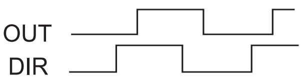
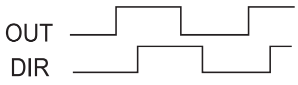
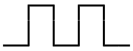
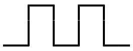
8-3-1. Output pulse mode

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

Common pulse mode: Outputs operation pulses from the OUT terminal and outputs the direction signal from the DIR terminal.
2-pulse mode: Outputs positive direction operation pulses from the OUT terminal, and outputs negative direction operation pulses from the DIR terminal.
90 phase difference pulse mode: This mode is used to output 90 phase difference pulses through the OUT and DIR terminals. One 90 phase difference is equivalent to one pulse in the ordinary and bi-directional pulse modes.

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

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

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

[RENV1] (WRITE)

7							0
-	-	-	-	-	n	n	n

Setting the direction change timer (0.2 msec) function <Set DTMF (bit 26) in RENV1>
0: ON
1: OFF

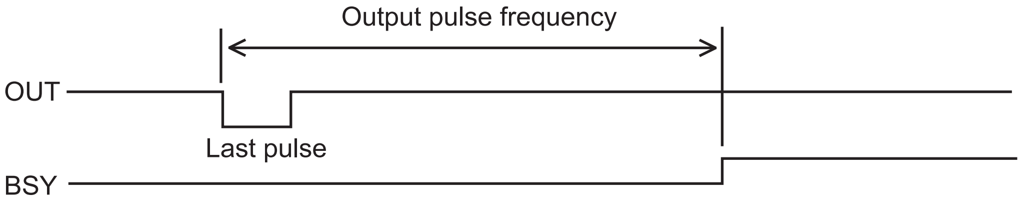
[RENV1] (WRITE)

31							24
-	-	-	-	-	n	-	-

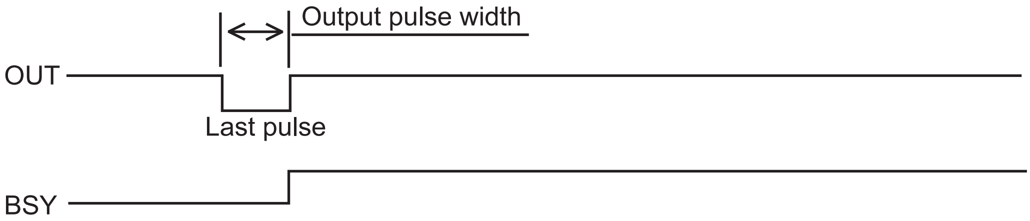
8-3-2. Control the output pulse width and operation complete timing

In order to increase the stopping speed, this LSI controls the output pulse width.
When the output pulse speed is slower than 1/16384 of reference clock (approx. 2.4 Kpps when 40 MHz), the pulse width is constant and is 8192 cycles of the reference clock (approx. 200 µsec when 40 MHz). For faster pulse speeds than this, the duty cycle is kept constant (approx. 50%). By setting PDTC (bit 29) in the RENV1 register (environment setting 1), the output pulse width can be set to make a constant duty cycle (50%).
Also, when setting METM (operation completion timing setting) in the RMD register (operation mode), the operation complete timing can be changed.

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



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

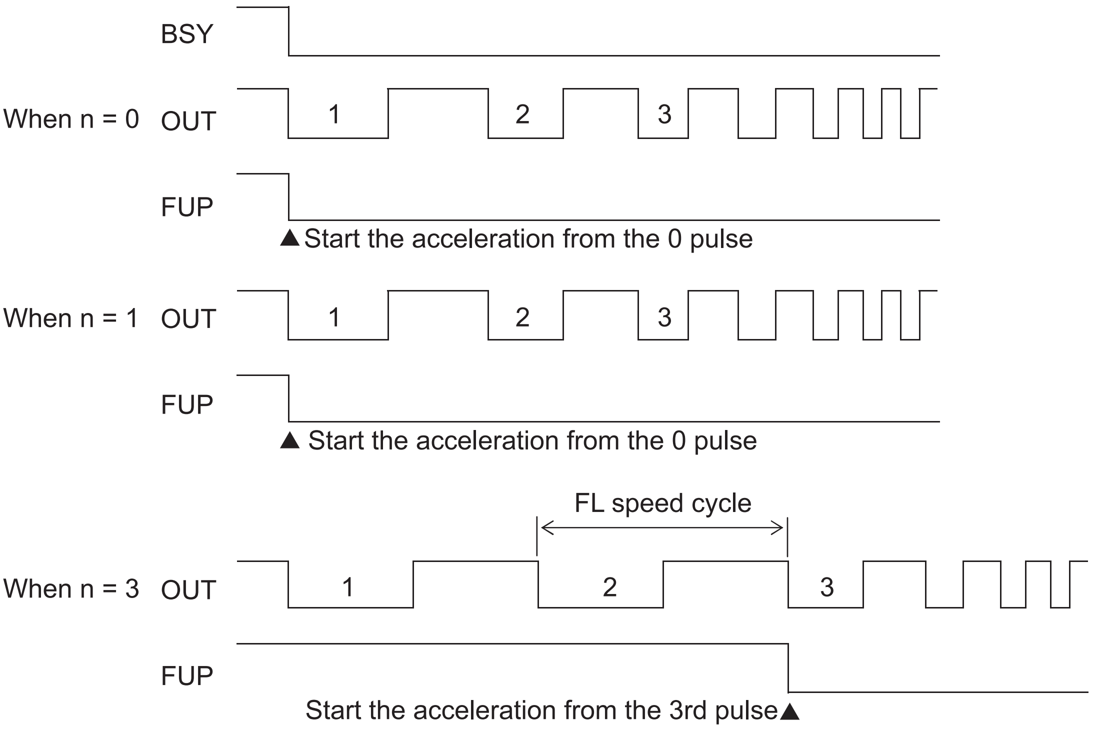


Setting the operation complete timing <Set METM (bit 11) in RMD> 0: At the end of a cycle of a particular output frequency 1: Complete when the output pulse turns OFF.	[RMD] (WRITE) 15 8 - - - - n - - -
Setting the output pulse width <Set PDTC (bit 29) in RENV1> 0: Automatically change between a constant output pulse and a constant duty cycle (approx. 50%) in accord with variations in speed. 1: Keep the output pulse width at a constant duty cycle (approx. 50%).	[RENV1] (WRITE) 31 24 - - n - - - - -

8-4. Idling control

When starting an acceleration or a deceleration operation, it can be started after the output of a few pulses at FL speed (idling output). Set the number of pulses for idling in IDL of the RENV5 register (environment setting 5).
If you will not be using this function, enter a value "n" of 0 or 1. The LSI will start the acceleration at the same time it begins outputting pulses. Therefore, the start speed obtained from an initial 2-pulse frequency will be faster than the FL speed.
To use this function, enter a value "n" of 2 to 7. The LSI will start the acceleration by beginning its output on the "n" th pulse. Therefore, the start speed will be the FL speed and the FL speed can be set to start automatically at upper speed limit.
If this function is used with the positioning mode, the total feed amount will not change.

[Setting idling pulses and the acceleration start timing]


BSY
When n = 0
OUT
1
2
3
FUP
▲Start the acceleration from the 0 pulse
When n = 1
OUT
1
2
3
FUP
▲ Start the acceleration from the 0 pulse
FL speed cycle
When n = 3
OUT
1
2
3
FUP
Start the acceleration from the 3rd pulse▲

Set the number of idling pulses <Set IDL0 to 2 (bits 20 to 22) in RENV2> Specify the number of idling pulses, from 0 to 7. Start accelerating at FL speed after outputting the specified number of pulses.	[RENV2] (WRITE) 23 16 - n n n - - - -
Read the idling control counter value < IDC0 to 2 (bits 24 to 26) in RSPD> Read the idling control counter.	[RSPD] (READ) 31 24 - - - - - n n n

8-5. Mechanical external input control

8-5-1. +EL, -EL signal

When an end limit signal (a +EL signal when feeding in the + direction) in the feed direction turns ON while operating, the axis will stop immediately or decelerate and stop. After stopping, even if the EL signal is turned OFF, the axis will remain stopped. For safety, keep the EL signal ON until the axis reaches the end of the stroke.
If the EL signal is ON when writing a start command, the axis cannot start moving in the direction of the particular EL signal that is ON.
By setting ELM in the RENV1 (environment setting 1) register, the stopping pattern for use when the EL signal is turned ON can be set to immediate stop or deceleration stop (high speed start only). However, when the deceleration stop is selected, keep the EL input "ON" until the motor stops. If the EL signal goes ON during a deceleration stop operation, and this signal is not held ON until the motor stops, the PCL device (G9003) will treat it as a normal stop.
The minimum pulse width of the EL signal is 160 reference clock cycles (4 µsec) when the input filter is ON. When the input filter is turned OFF, the minimum pulse width is 4 reference clock cycles (0.1 µsec).
The EL signal can be monitored by reading RSTS (extension status).
By reading the REST register, you can check for an error interrupt caused by the EL signal turning ON.
When in the timer mode, this signal is ignored. Even in this case, the EL signal can be monitored by reading RSTS (extension status).
The input logic of the EL signal can be set for each axis using the ELL input terminal.

Set the input logic of the ±EL signal <ELL input terminal> L: Positive logic input H: Negative logic input	
Stop method to when the ±EL signal turns ON <Set ELM (bit 3) in RENV1> 0: Immediate stop by turning ON the ±EL signal 1: Deceleration stop by turning ON the ±EL signal	[RENV1] (WRITE) 7 0 - - - - n - - -

Reading the ±EL signal <SPEL (bit 6), SMEL (bit 7) in RSTS>
SPEL = 0:Turn OFF the +EL signal SPEL = 1: Turn ON the +EL signal
SMEL = 0:Turn OFF the -EL signal SMEL = 1: Turn ON the -EL signal
	[RSTS] (READ)
	7 0
	n n - - - - - -

Setting the ±EL input filter <Set FLTR (bit 25) in RENV1> 0: Apply a filter to the ±EL and ORG input Apply a filter and any signals shorter than 4 µsec pulse width are ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

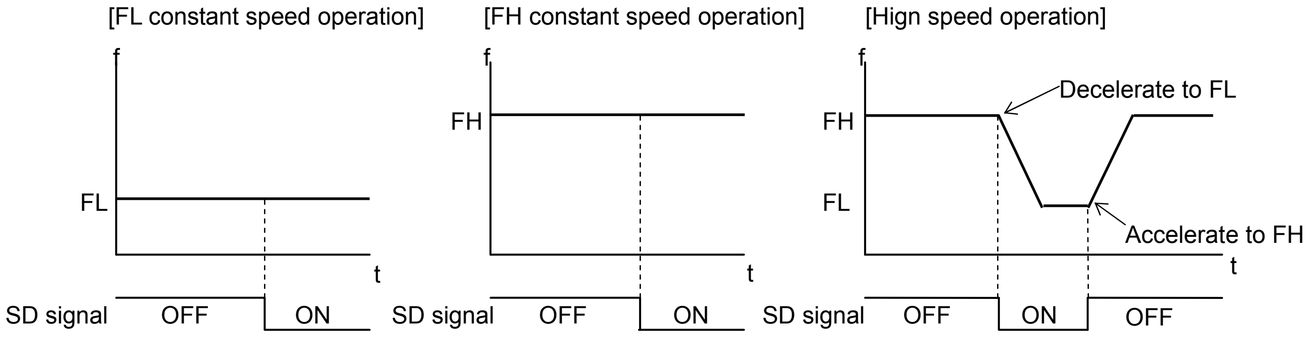
Note 1: Operation after turning ON the EL signal may be different for the origin return operation (6-5-1), the zero search operation (6-5-3), and the EL or SL operation mode (6-6). See the description of each operation mode.

8-5-2. SD signal

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

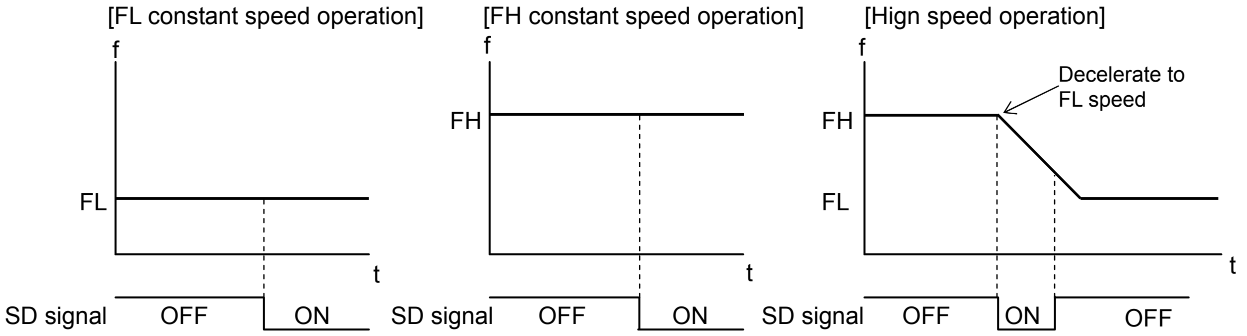
1) Deceleration < SDM (bit 4) = 0, SDLT (bit 5) = 0 in RENV1 register>

- While feeding at constant speed, the SD signal is ignored. While in high speed operation the axis decelerates to the FL speed when the SD signal is turned ON. After decelerating, or while decelerating, if the SD signal turns OFF, the axis will accelerate to the FH speed.
- If the SD signal is turned ON when the high speed command is written, the axis will operate at FL speed. When the SD signal is turned OFF, the axis will accelerate to FH speed.


[FL constant speed operation] [FH constant speed operation] [Hign speed operation]

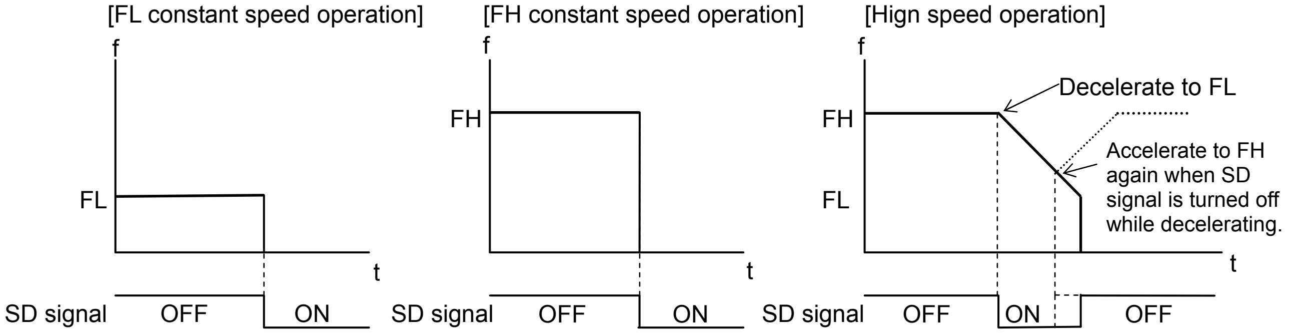
2) Latch and decelerate <SDM (bit 4) = 0, SDLT (bit 5) = 1 in RENV1 register>

- While feeding at constant speed, the SD signal is ignored. While in high speed operation, decelerate to FL speed by turning the SD signal ON. Even if the SD signal is turned OFF after decelerating or while decelerating, the axis will continue moving at FL speed and will not accelerate to FH speed.
- If the SD signal is turned ON while writing a high speed command, the axis will feed at FL speed. Even if the SD signal is turned OFF, the axis will not accelerate to FH speed.


[FL constant speed operation] [FH constant speed operation] [Hign speed operation]

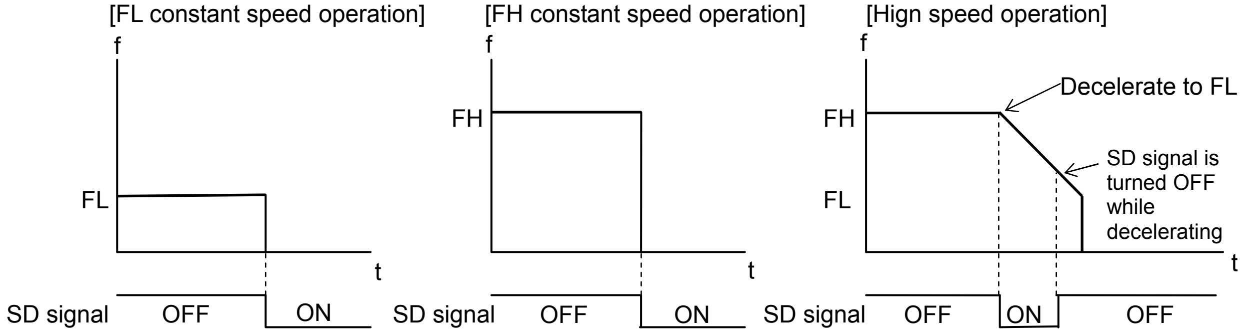
3) Deceleration stop <SDM (bit 4) = 1, SDLT (bit 5) = 0 in RENV1 register>

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


[FL constant speed operation] [FH constant speed operation] [Hign speed operation]

4) Latched, deceleration stop <SDM (bit 4) = 1, SDLT (bit 5)=1 in RENV1>

- If the SD signal is turned ON while in constant speed operation, the axis will stop. If the SD signal is turned ON while in high speed operation, the axis will decelerate to FL speed and then stop. Even if the SD signal is turned OFF during deceleration, the axis will not accelerate.
- If the SD signal is turned ON while writing a start command, the axis will not start moving and the operation will not be completed.
- Generate an interrupt when stopped.


[FL constant speed operation]
f
FL
t
SD signal
OFF
ON
[FH constant speed operation]
f
FH
t
SD signal
OFF
ON
[Hign speed operation]
f
FH
FL
Decelerate to FL
SD signal is turned OFF while decelerating
t
SD signal
OFF
ON
OFF

The input logic of the SD signal can be changed. If the latched input is set to accept input from the SD signal, and if the SD signal is OFF at the next start, the latch will be reset. The latch is also reset when the latch input (SDLT of the RENV1) is set to zero.
The minimum pulse width of the SD signal is 160 reference clock cycles (4.0 μsec) when the input filter is ON. When the input filter is turned OFF, the minimum pulse width is 4 reference clock cycles (0.1 μsec). (When CLK = 40 MHz.)
The latch signal of the SD signal can be monitored by reading RSTS (extension status). The SD signal terminal status can be monitored by reading RSTS (extension status). By reading the REST register, you can check for an error interrupt caused by the SD signal turning ON.

Enable/disable SD signal input <Set MSDE (bit 8) in RMD> 0: Disable SD signal input 1: Enable SD signal input	[RMD] (WRITE) 15 8 - - - - - - - n
Input logic of the SD signal <Set SDL(bit 6) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 - n - - - - - -
Set the operation pattern when the SD signal is turned ON <Set SDM (bit 4) in RENV1> 0: Decelerates on receiving the SD signal and feeds at FL constant speed 1: Decelerates and stops on receiving the SD signal	[RENV1] (WRITE) 7 0 - - - n - - - -
Select the SD signal input type <Set SDLT (bit 5) in RENV1> 0: Level input 1: Latch input To release the latch, turn OFF the SD input when next start command is written or select level input.	[RENV1] (WRITE) 7 0 - - n - - - - -
Reading the latch status of the SD signal <SSD (bit 9) in RSTS> 0: The SD latch signal is OFF 1: The SD latch signal is ON	[RSTS] (READ) 15 8 - - - - - - n -
Reading the SD signal < SDIN (bit 10) in the RSTS register> 0: The SD signal is OFF 1: The SD signal is ON	[RSTS] (READ) 15 8 - - - - - n - -
Reading the cause of an interrupt when stopped by the SD signal <ESSD (bit 8) in RESET> 1: Deceleration stop caused by the SD signal turning ON	[REST] (READ) 15 8 - - - - - - - n
Apply an input filter to SD <Set FLTR (bit 25) in RENV1> 0: Apply a filter to the SD input By applying a filter, signals with a pulse width of 4 μsec or less will be ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

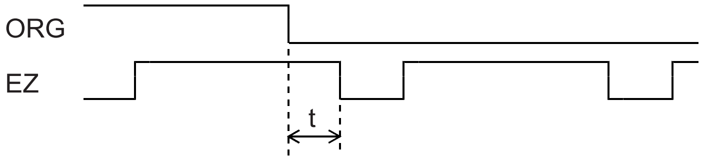
8-5-3. ORG, EZ signals

These signals are enabled in the origin return modes (origin return, leave origin position, and origin position search) and in the EZ count operation modes. Specify the operation mode and the operation direction using the RMD register (operation mode).
The PCL device (G9003) latches the ORG signal on the rising edge of the output pulse (negative logic).
The minimum pulse length of the ORG signal is one cycle period of the output pulses.
Since the ORG signal input is latched internally, there is no need to keep the external signal ON.
The ORG latch signal is reset when stopped.

The input logic of the ORG and EZ signals can be changed using the RENV1 (environment 1) register and RENV2 (environment 2) registers, respectively.

By reading the RSTS (extension status) register, you can monitor the status of the ORG and EZ terminals.
For details about the origin return operation modes, see 6-5, "Origin position operation mode."

ORG signal and EZ signal timing



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

Enabling the ORG and EZ signals <Set MOD (bits 0 to 6) in RMD> 001 0000: Origin return in the positive direction 001 0010: Leave origin position in the positive direction 001 0101: Origin position search in the positive direction 010 0100: EZ counting in the positive direction 001 1000: Origin return in the negative direction 001 1010: Leave origin position in the negative direction 001 1101: Origin position search in the negative direction 010 1100: EZ count operation in the negative direction	[RMD] (WRITE) 7 0 0 n n n n n n n
Set the origin return method <Set ORM0 to 3 (bits 0 to 3) in RENV3> See the RENV3 register description	[RENV3] (WRITE) 7 0 - - - - n n n n
Set the input logic for the ORG signal <Set ORGL (bit 7) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 n - - - - - - -
Read the ORG signal <SORG (bit 8) in RSTS> 0: The ORG signal is OFF 1: The ORG signal is ON	[RSTS] (READ) 15 8 - - - - - - - n
Set the EZ count number <Set EZD0 to 3 (bits 4 to 7) in RENV3> Set the origin return completion condition and the EZ count number for counting. Specify the value (the number to count to -1) in EZD0 to 3. The setting range is 0 to 15.	[RENV3] (WRITE) 7 0 n n n n - - - -
Specify the input logic of the EZ signal <Set EZL (bit 12 in RENV2> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 15 8 - - - n - - - -
Read the EZ signal <SEZ (bit 16) in RSTS> 0: The EZ signal is OFF 1: The EZ signal is ON	[RSTS] (READ) 23 16 - - - - - - - n
Apply an input filter to EZ <Set EINF (bit 8) in RENV2> 0: Apply a filter to the EZ input By applying a filter, input signal shorter than 6 cycles of the CLK input will be ignored.	[RENV2] (WRITE) 15 8 - - - - - - - n

8-6. Servomotor I/F (Case in digital servo)

8-6-1. INP signal

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

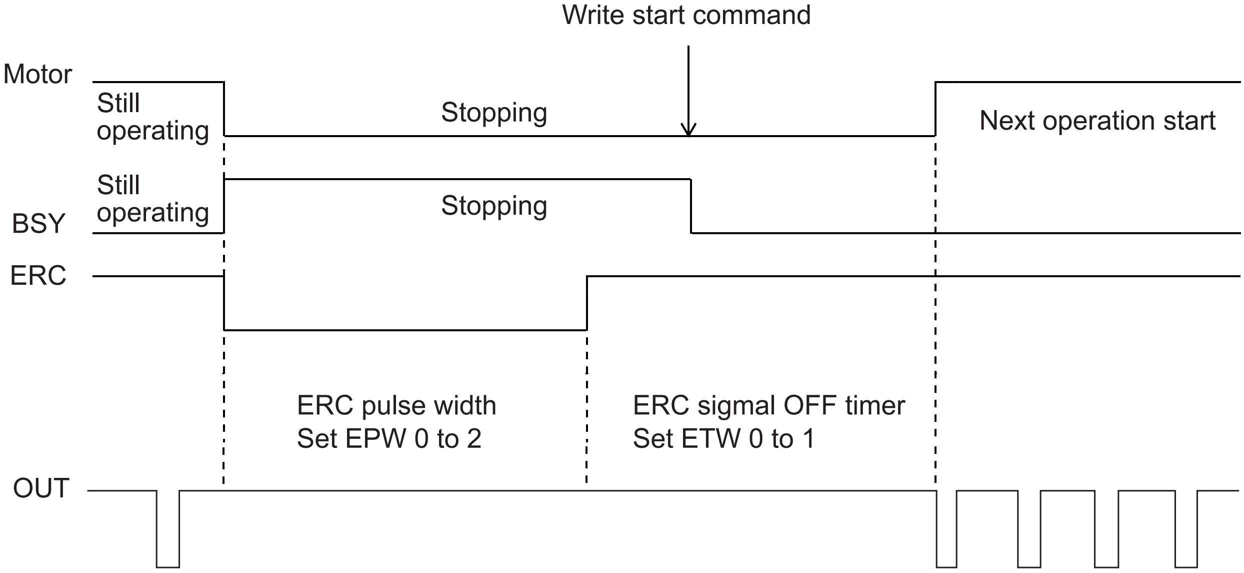
This LSI can receive a positioning complete signal (INP signal) from a servo driver in place of the pulse output complete timing to determine when an operation is complete.
When the INP signal input is used to indicate the completion status of an operation, the #BSY signal when an operation is complete, the main status (bit 0 to 3 and 8 of the MSTS, stop condition), and the extension status (CND0 to 3 of RSTS, operation status) will also change when the INP signal is input.

The input logic of the INP signal can be changed.
The minimum pulse width of the INP signal is 160 reference clock cycles (4 μsec) when the input filter is ON. If the input filter is OFF, the minimum pulse width will be 4 reference clock cycles (0.1 μsec). (When CLK = 40 MHz)
If the INP signal is already ON when the PCL device (G9003) is finished outputting pulses, it treats the operation as complete, without any delay.
The INP signal can be monitored by reading the RSTS register (extension status).

Set the operation complete delay using the INP signal <Set MINP (bit 9) in RMD> 0: No operation complete delay waiting for the INP signal. 1: Operation complete (status, #BSY) delay until the INP signal turns ON.	[RMD] (WRITE) 15 8 - - - - - - n -
Input logic of the INP signal <Set INPL (bit 22) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 23 16 - n - - - - - -
Reading the INP signal <SINP (bit 19) in RSTS> 0: The INP signal is OFF 1: The INP signal is ON	[RSTS] (READ) 23 16 - - - - n - - -
Set the INP input filter <FLTR (bit 25) in RENV1> 0: Apply a filter to the INP input. By applying a filter, pulses less than 4 μsec in width are ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

8-6-2. ERC signal

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



In order to output an ERC signal at the completion of an origin return operation, set EROR (bit 11) = 1 in the RENV1 register (environment setting 1) to make the ERC signal an automatic output. For details about ERC signal output timing, see the timing waveform in section 6-4-1, "Origin return operation."
In order to output an ERC signal for an immediate stop based on the EL signal, ALM signal, or #EMG signal input, or on the emergency stop command (0005h), set EROE (bit 10) = 1 in the RENV1 register, and set automatic output for the ERC signal. (In the case of a deceleration stop, the ERC signal cannot be output, even when set for automatic output.)
The ERC signal can be output by writing an ERC output command (0024h).
The output logic of the ERC signal can be changed by setting the RENV1 register. Read the RSTS (extension status) register to monitor the ERC signal.

Set automatic output for the ERC signal <Set EROE (bit 10) in RENV1> 1: Does not output an ERC signal when stopped by EL, ALM, or #EMG input. 1: Automatically outputs an ERC signal when stopped by EL, ALM, or #EMG input.	[RENV1] (WRITE) 15 8 - - - - - n - -
Set automatic output for the ERC signal <Set EROR (bit 11) in RENV1> 0: Does not output an ERC signal at the completion of an origin return operation. 1: Automatically outputs an ERC signal at the completion of an origin return operation.	[RENV1] (WRITE) 15 8 - - - - n - - -
Set the ERC signal output width <Set EPW0 to 2 (bits 12 to 14) in RENV1> 000: 12 μsec 100: 13 msec 001: 102 μsec 101: 52 msec 010: 409 μsec 110: 104 msec 011: 1.6 msec 111: Logic level output	[RENV1] (WRITE) 15 8 - n n n - - - -
Select output logic for the ERC signal <Set ERCL (bit 15) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 15 8 n - - - - - - -
Specify the ERC signal OFF timer time <Set ETW0 to 1 (bits 16 to 17) in RENV1> 00: 0 μsec 10: 1.6 msec 01: 12 μsec 11: 104 msec	[RENV1] (WRITE) 23 16 n - - - - - - -
Read the ERC signal <SERC (bit 15) in RSTS> 0: The ERC signal is OFF 1: The ERC signal is ON	[RSTS] (READ) 15 8 n - - - - - - -
Emergency stop command <MEMG: Operation command> Output an ERC signal	[Operation command] 0005h
ERC signal output command <ERCOUT: Control command> Turn ON the ERC signal	[Control command] 0024h
ERC signal output reset command <ERCRST: Control command> Turn OFF the ERC signal	[Control command] 0025h

8-6-3. ALM signals

Input alarm (ALM) signal.
When the ALM signal turns ON while in operation, the axis will stop immediately or decelerate and stop.
To stop using deceleration, keep the ALM input ON until the axis stops operation.
However, the axis only decelerates and stops on an ALM signal if it was started with a high speed start.
If the ALM signal is ON when a start command is written, the LSI will not output any pulses.
The minimum pulse width of the ALM signal is 160 reference clock cycles (4 μsec) if the input filter is ON.
If the input filter is OFF, the minimum pulse width is 4 reference clock cycles (0.1 μsec). (When CLK = 40 MHz.)
The input logic of the ALM signal can be changed. The signal status of the ALM signal can be monitored by reading RSTS (extension status).

Stop method when the ALM signal is ON <Set ALMM (bit 8) in RENV1> 0: Stop immediately when the ALM signal is turned ON 1: Deceleration stop (high speed start only) when the ALM signal is turned ON	[RENV1] (WRITE) 15 8 - - - - - - - n
Input logic setting of the ALM signal <Set ALML (bit 9) in RENV1> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 15 8 - - - - - - n -
Read the ALM signal <SALM (bit 5) in RSTS> 0: The ALM signal is OFF 1: The ALM signal is ON	[RSTS] (READ) 7 0 - - n - - - - -
Reading the cause of a stop when the ALM signal is turned ON <ESAL (bit 5) in REST> 1: Stop due to the ALM signal being turned ON	[REST] (READ) 7 0 - - n - - - - -
Set the ALM input filter <Set FLTR (bit 25) in RENV1> 0: Apply a filter to the ALM input When a filter is applied, pulses less than 4 μsec pulse in width will be ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

8-7. External start, simultaneous start

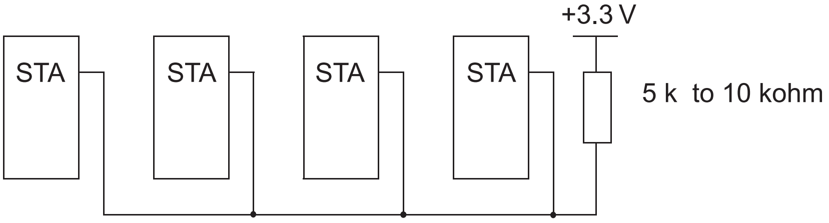
This LSI can start when triggered by an external signal on the #STA terminals. Set MSY (bits 14) = "1" in the RDM register (operation mode) and the LSI will start feeding when the #STA goes LOW.
When you want to control multiple axes using more than one LSI, connect the #STA terminal on each LSI and set the axes to "waiting for #STA input", to start them all at the same time. In this example a start signal can be output through the #STA terminal.
The input logic on the #STA terminals cannot be changed.
By setting the RIRQ register (event interrupt cause), an interrupt occurs together with a simultaneous start (when the #STA input is ON).
By reading the RIST register, the cause of an event interrupt can be checked. The operation status (waiting for #STA input), and status of the #STA terminal can be monitored by reading the RIST register.

<How to make a simultaneous start>
Set MSY (bits 14) = "1"in the RMD register for the axes you want to start. Write a start command and put the LSI in the "waiting for #STA input" status. Then, start the axes simultaneously by either of the methods described below.

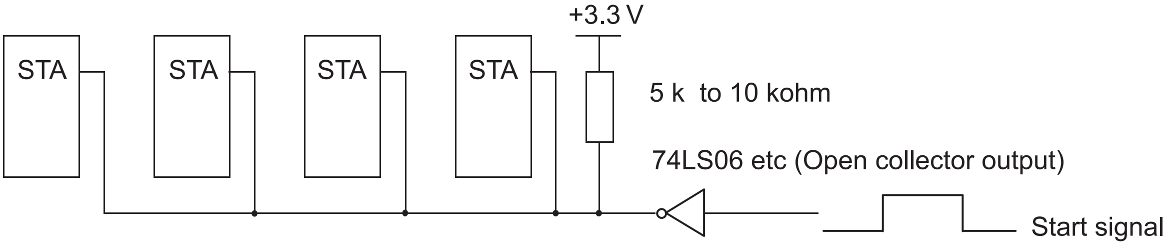
1) By writing a simultaneous start command, the LSI will output a one shot signal of 16 reference clock cycles (approx. 0.4 μsec when CLK = 40 MHz) from the #STA terminal.
2) Input hardware signal from outside.
Supply a hardware signal by driving the terminal with open collector output (74LS06 or equivalent).

#STA signals can be supplied as level trigger or edge trigger inputs. However, when level trigger input is selected, if #STA = L or a start command is written, the axis will start immediately.
After connecting the #STA terminals on each LSI, each axis can still be started independently using start commands.
To release the "waiting for #STA input" condition, write an immediate stop command (0049h).

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



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



For start signal, supply a one shot input signal with a pulse width of at least 8 reference clock cycles (approx. 0.2 μsec when CLK = 40 MHz).

#STA input <MSY (bits 14) in RMD> 01: Start by inputting a #STA signal	[RMD] (WRITE) 15 8 - n - - - - - -
Specify the input specification for the #STA signal <Set STAM (bit 18) in RENV1> 0: Level trigger input for the #STA signal 1: Edge trigger input for the #STA signal	[RENV1] (WRITE) 23 16 - - - - - n - -
Read the #STA signal <SSTA (bit 11) in RSTS> 0: The #STA signal is OFF 1: The #STA signal is ON	[RSTS] (READ) 15 8 - - - - n - - -
Read the operation status <CND (bits 0 to 3) in RSTS> 0001: Waiting for #STA input	[RSTS] (READ) 7 0 - - - - n n n n
Set an event interrupt cause <Set IRSA (bit 12) in RIRQ> 1: Generates an interrupt when the #STA input is ON.	[RIRQ] (WRITE) 15 8 - - - n - - - -
Reading the event interrupt cause <ISSA (bit 12) in RIST> 1: When the #STA signal is ON.	[RIST] (READ) 15 8 - - - n - - - -
Simultaneous start command <CMSTA: Operation command> Output a one shot pulse 16 reference clock cycles long from the #STA terminal. (The #STA terminal is bi-directional. It can receive signals output from other PCL devices(G9003).)	[Operation command] 0006h
Local axis only, simultaneous start command <SPSTA: Operation command> Used the same way as when a #STA signal is supplied, for a local axis only.	[Operation command]] 002Ah

8-8. External stop / simultaneous stop

This LSI can execute an immediate stop or a deceleration stop triggered by an external signal using the #STP terminal. Set MSPE (bit 15) = "1" in the RMD register (operation mode) to enable a stop from a #STP input. The axis will stop immediately or decelerate and stop when the #STP terminal is LOW. However, a deceleration stop is only used for a high speed start. When the axis is started at constant speed, the signal on the #STP terminal will cause an immediate stop.

The input logic of the #STP terminal cannot be changed.

When multiple LSIs are used to control multiple axes, connect all of the #STP terminals from each LSI and input the same signal so that the axes which are set to stop on a #STP input can be stopped simultaneously. In this case, a stop signal can also be output from the #STP terminal.

When an axis stops because the #STP signal is turned ON, an interrupt occurs. By reading the REST register, you can determine the cause of an error interrupt. You can monitor #STP terminal status by reading the RSTS register (extension status).

<How to make a simultaneous stop>

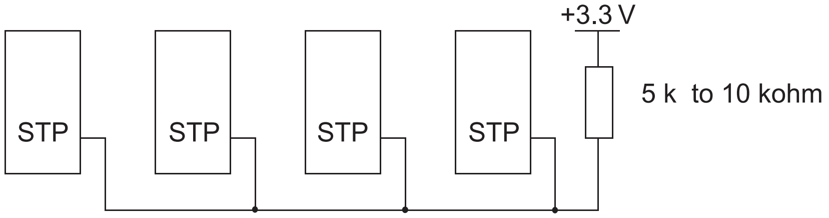
Set MSPE (bit 15) = "1" in the RMD register for each of the axes that you want to stop simultaneously. Then start these axes.

Stop these axes using either of the following 3 methods.

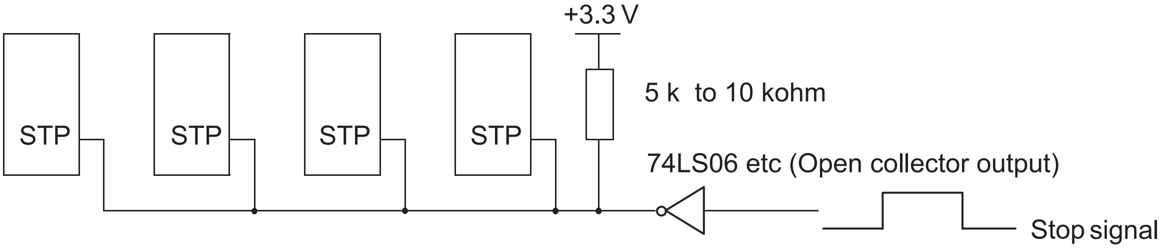
1) By writing a simultaneous stop command, the #STP terminal will output a one shot signal 16 reference clock cycles in length (approx. 0.4 µsec when CLK = 40 MHz).
2) Supply an external hardware signal
   Supply a hardware signal using an open collector output (74LS06 or equivalent).
3) The #STP terminal will output a one shot signal for 16 reference clock cycles (approximately 0.4 µsec when CLK = 40 MHz) when a stop caused by an error occurs on an axis that has MSPO = 1 in the RMD register.

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

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



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



As a stop signal, supply a one shot signal 8 reference clock cycles or more in length (approx. 0.2 μsec when CLK = 40 MHz).

Setting to enable #STP input <Set MSPE (bit 15) in RMD> 1. Enable a stop from the #STP input. (Immediate stop, deceleration stop)	[RMD] (WRITE) 15 8 n - - - - - - -
Auto output setting for the #STP signal <Set to MSPO (bit 16) in the RMD> 1: When an axis stops because of an error, the PCL device (G9003) will output the #STP signal. (Output signal width: 16 reference clock cycles)	[RMD] (WRITE) 23 16 - - - n - - - -
Specify the stop method to use when the #STP signal is turned ON. <Set STPM (bit 19) in RENV1> 0: Immediate stop when the #STP signal is turned ON. 1: Deceleration stop when the #STP signal is turned ON.	[RENV1] (WRITE) 23 16 - - - - n - - -
Read the #STP signal <SSTP (bit 12) in RSTS> 0: The #STP signal is OFF 1: The #STP signal is ON	[RSTS] (READ) 15 8 - - - n - - - -
Read the cause of an error input < ESSP (bit 6) in REST> 1. When stopped because the #STP signal turned ON.	[REST] (READ) 7 0 - n - - - - - -
Simultaneous stop command <CMSTP: Operation command> Outputs a one shot pulse of 16 reference clock cycles in length from the #STP terminal. (The #STP terminal is bi-directional. It can receive signals output from other PCL devices(G9003).)	[Operation command] 0007h

8-9. Emergency stop

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

When the axes are stopped because the #EMG input was turned ON, the LSI will generate an interrupt.
By reading the REST register, the cause of the error interruption can be determined.
The status of the #EMG terminal can be monitored by reading the RSTS register (extension status).

Read the #EMG signal <SEMG (bit 13) in RSTS> 0: The #EMG signal is OFF 1: The #EMG signal is ON	[RSTS] (READ) 15 7 - - n - - - - -
Read the cause of an error interrupt <ESEM (bit 7) in REST> 1. Stopped when the #EMG signal was turned ON.	[REST] (READ) 7 0 n - - - - - - -
Set the input signal filter for #EMG 0: Turn ON the filter function (Input signals shorter than 4 reference clock cycles are ignored.)	[RENV1] (WRITE) 31 24 - - - - - - n -
Emergency stop command <CMEMG: Operation command> The operation is the same as when an #EMG signal is input.	[Operation command] 0005h

Note: In a normal stop operation, the final pulse width is normal. However, in an emergency stop operation, the final pulse width may not be normal. It can be triangular. Motor drivers do not recognize triangle shaped pulses, and therefore only the PCL device (G9003) counter may count this pulse. (Deviation from the instructed position control). Therefore, after an emergency stop, you must perform an origin return to match the instructed position with the mechanical position.

8-10. Counter

8-10-1. Counter type and input method

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

- ♦ Control command position and mechanical position
- ♦ Detect a stepper motor that is "out of step" using COUNTER3 (general-purpose, deflection counter) and a comparator.

The positioning counter is loaded with an absolute value for the RMV register (target position) with each start command, regardless of the operation mode selected. It decreases the value with each pulse that is output. However, if MPCS (bit 13) of the RMD register (operation mode) is set to 1 and a position override 2 is executed, the counter does not decrease until the PCS input is turned ON.
Input to COUNTER1 is exclusively for output pulses. However COUNTERS2 to 3 can be selected as follows by setting the RENV3 register (environment setting 3).

	COUNTER1	COUNTER2	COUNTER3
Counter name	Command position	Mechanical position	General-purpose, deflection
Counter type	Up/down counter	Up/down counter	Deflection counter
Number of bits	28	28	16
Output pulse	Available	Available	Available
Encoder (EA/EB) input		Available	Available
Pulsar (PA/PB) input		Available	Available
1/4096 of reference clock			Available

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

Specify COUNTER2 (mechanical position) input <CI20 to 21 (bit 8 to 9) in RENV3> 00: EA/EB input 01: Output pulses 10: PA/PB input	[RENV3] (WRITE) 15 8 - - - - - - n n
Set COUNTER3 (deflection) input <CI30 to 32 (bit 10 to 12) in RENV3> 000:Output pulses 001: EA/EB input 010: PA/PB input 011: 1/4096 division of the internal reference clock (CLK = 40MHz). 100: Measure the deflection between output pulses and EA/EB input 101: Measure the deflection between output pulses and PA/PB input 110: Measure the deflection between EA/EB input and PA/PB input	[RENV3] (WRITE) 15 8 - - - n n n - -

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

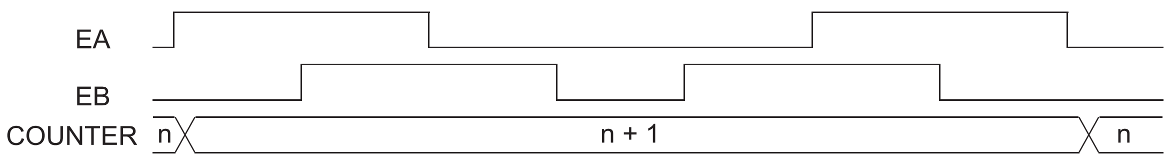
1) Signal input method: Input 90 phase difference signals (1x, 2x, 4x)
   Counter direction: Count up when the EA input phase is leading. Count down when the EB input phase is leading.
2) Signal input method: Input 2 sets of positive and negative pulses.
   Counter direction: Count up on the rising edge of the EA input. Count down on the falling edge of the EB input.

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

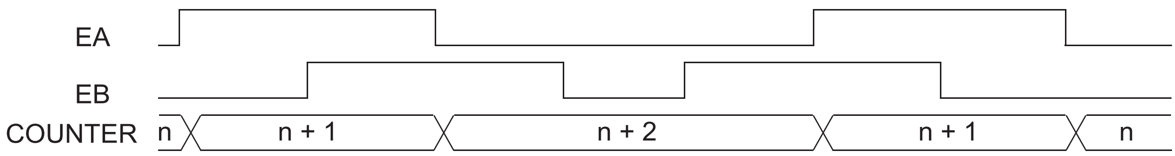
Set the input signal filter for EA/EB/EZ <Set EINF (bit 8) in RENV2> 0: Turn OFF the filter function 1: Turn ON the filter function (Input signals shorter than 6 reference clock cycles are ignored.)	[RENV2] (WRITE) 15 8 - - - - - - - n
Setting the EA/EB input <Set EIM0 to 1 (bit 9 to 10) in RENV2> 00: 90 phase difference, 1x 10: 90 phase difference, 4x 01: 90 phase difference, 2x 11: 2 sets of up or down input pulses	[RENV2] (WRITE) 15 8 - - - - - n n -
Specify the EA/EB input count direction <Set to EDIR (bit 11) in RENV2> 0: Count up when the EA phase is leading. Or, count up on the rising edge of EA. 1: Count up when the EB phase is leading. Or, count up on the rising edge of EB.	[RENV2] (WRITE) 15 8 - - - - n - - -
Enable/disable EA/EB input <Set EOFF (bit 17) in RENV2> 0: Enable EA/EB input 1: Disable EA/EB input. (EZ input is valid.)	[RENV2] (WRITE) 23 16 - - - - - - n -
Set the input signal filter for PA/PB <Set PINF (bit 13) in RENV2> 0: Turn OFF the filter function. 1: Turn ON the filter function (Input signals shorter than 6 reference clock cycles are ignored.)	[RENV2] (WRITE) 15 8 - - n - - - - -
Specify the PA/PB input <Set to PIM0 to 1 (bit 14 to 15) in RENV2> 00: 90 phase difference, 1x 10: 90 phase difference, 4x 01: 90 phase difference, 2x 11: 2 sets of up or down input pulses	[RENV2] (WRITE) 15 8 n n - - - - - -
Specify the PA/PB input count direction <Set to PDIR (bit 16) in RENV2> 0: Count up when the PA phase is leading. Or, count up on the rising edge of PA. 1: Count up when the PB phase is leading. Or, count up on the rising edge of PB.	[RENV2] (WRITE) 23 16 - - - - - - - n
Enable/disable PA/PB input <Set POFF (bit 18) in RENV2> 0: Enable PA/PB input 1: Disable PA/PB input.	[RENV2] (WRITE) 23 16 - - - - - n - -
Reading EA/EB, PA/PB input error <ESEE (bit 13), ESPE (bit 14) in the REST> ESEE (bit 13) = 1: An EA/EB input error occurred. ESPE (bit 14) = 1: A PA/PB input error occurred.	[REST] (READ) 15 8 - n n - - - - -

When EDIR is "0," the EA/EB input and count timing will be as follows.
For details about the PA/PB input, see section "6-3. Pulsar input mode."

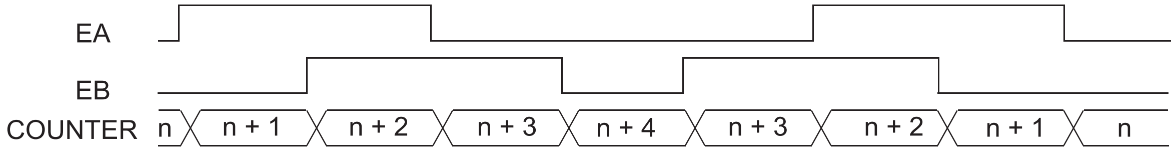
1) When using 90 phase difference signals and 1x input



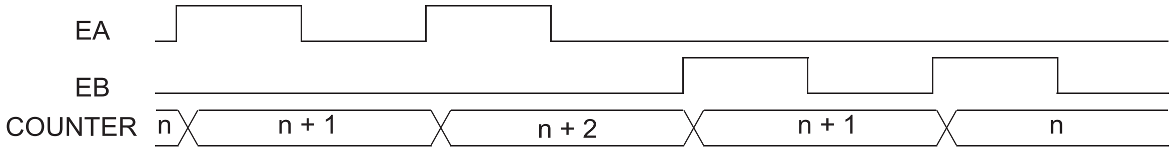
2) When using 90 phase difference signals and 2x input



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



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



8-10-2. Counter reset

All the counters can be reset using any of the following three methods.
1) When the CLR input signal turns ON (set in RENV3).
2) When an origin return is executed (set in RENV3).
3) When a command is written.

The CLR input timing can be set in RENV1 (environment setting 1). As an event interrupt cause, an interrupt can be generated when inputting the CLR.

Action when the CLR signal turns ON <Set CU1C to 3C (bit 16 to 18) in the RENV3> CU1C (bit 16) =1: Reset COUNTER1 (command position). CU2C (bit 17) =1: Reset COUNTER2 (mechanical position). CU3C (bit 18) =1: Reset COUNTER3 (general-purpose, deflection).	[RENV3] (WRITE) 23 16 - - - - 0 n n n
Action when an origin return is complete <Set CU1R to 3R (bit 20 to 22) in RENV3> CU1R (bit 20) =1: Reset COUNTER1 (command position). CU2R (bit 21) =1: Reset COUNTER2 (mechanical position) CU3R (bit 22) =1: Reset COUNTER3 (general-use, deflection)	[RENV3] (WRITE) 23 16 0 n n n - - - -
Action for the CLR signal <Set CLR0 and 1 (bit 20 to 21) in RENV1> 00: Clear on the falling edge 10: Clear on a LOW level 01: Clear on the rising edge 11: Clear on a HIGH level	[RENV1] (WRITE) 23 16 - - - - - - - n
Reading the CLR signal <SCLR (bit 17) in RSTS> 0: The CLR signal is OFF 1: The CLR signal is ON	[RSTS] (READ) 23 16 - - - - - - - n
Set event interrupt cause <Set IRCL (bit 8) in RIRQ> 1: Generate an interrupt signal when resetting the counter value by turning the CLR signal ON.	[RIRQ] (WRITE) 15 8 - - - - - - - n
Read the event interrupt cause <ISCL (bit 8) in RIST> 1: When you want to reset the counter value by turning ON the CLR signal.	[RIST] (READ) 15 8 - - - - - - - n
Counter reset command <CUN1R to CUN4R: Control command> 20h: Reset COUNTER1 (command position). 21h: Reset COUNTER2 (mechanical position). 22h: Reset COUNTER3 (general-purpose, deflection).	[Control command] 0020h 0021h 0022h

Note: When the count up (down) timing and reset timing match, the counter will be set to 0.

8-10-3. Latch the counter and count condition

All the counters can latch their counts using any of the following methods. The setting is made in RENV4 (environment setting 4) register. The latched values can be output from the RLTC1 to 3 registers.

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

The current speed can also be latched instead of COUNTER3 (general-purpose, deflection). Items 1) to 4) above can also be latched by hardware timing.
The LTC input timing can be set by in RENV1 (environment setting 1). An interrupt can be generated when a counter value is latched by turning ON the LTC signal or the ORG signal. This allows you to identify the cause of an event interrupt.

Specify the latch method for a counter (1 to 4) <Set LTM0 to 1 (bit 24 to 25) in RENV4> 00: Turn ON the LTC signal. 01: Turn ON the ORG signal. 10: When the conditions for Comparator 4 are satisfied. 11: When the conditions for Comparator 5 are satisfied	[RENV4] (WRITE) 31 24 - - - - - - n n
Specify the latch method for the current speed <Set LTFD (bit 26) in RENV4> 1: Latch the current speed instead of COUNTER 3 (general-purpose, deflection).	[RENV4] (WRITE) 31 24 - - - - - n - -
Specify latching using hardware <Set LTOF (bit 27) in RENV4> 1: Do not latch 1) to 4) above with hardware timing.	[RENV] (WRITE) 31 24 - - - - n - - -
Specify the LTC signal mode <Set LTCL (bit 23) in RENV1> 0: Latch on the falling edge. 1: Latch on the rising edge.	[RENV1] (WRITE) 23 16 n - - - - - - -
Set an event interrupt cause <Set IRLT (bit 9) and IROL (bit 10) in RIRQ> IRLT = 1: Generates an interrupt when the counter value is latched by the LTC signal being turned ON. IROL = 1: Generates an interrupt when the counter value is latched by the ORG signal being turned ON.	[RIRQ] (WRITE) 15 8 - - - - - n n -
Read the event interrupt cause <ISLT (bit 9), ISOL (bit 10) in RIST> ISLT = 1: Latch the counter value when the LTC signal turns ON. ISOL = 1: Latch the counter value when the ORG signal turns ON.	[RIST] (READ) 15 8 - - - - - n n -
Read the LTC signal <SLTC (bit 18) in RSTS> 0: The LTC signal is OFF 1: The LTC signal is ON	[RSTS] (READ) 23 16 - - - - - n - -
Counter latch command <LTCH: Control command> Latch the contents of the counters (COUNTER1 to 3).	[Control command] 0029h

8-10-4. Stop the counter

COUNTER1 (command position), COUNTER2 (mechanical position), and COUNTER3 (general-purpose, deflection) stop when the RENV3 (environment setting 3) register can be set to stop. COUNTER1 (command position) stops while in timer mode operation.
By setting the RENV3 register, you can stop counting pulses while performing a backlash correction.
COUNTER3 (general-purpose) can be set to count only during operation (#BSY = low) using the RENV3 register. By specifying 1/4096 of the CLK (reference clock) signal, the time after the start can be controlled.

Specify the counting operation for COUNTERS 1 to 3 <Set CU1H to 3H (bits 28 to 30) in RENV3> CU1H (bit 28) = 1: Stop COUNTER1 (mechanical position) CU2H (bit 29) = 1: Stop COUNTER2 (deflection) CU3H (bit 30) = 1: Stop COUNTER3 (general-purpose, deflection)	[RENV3] (WRITE) 31 24 - n n n - - - -
Setting the counters for backlash correction <Set CU1B to 3B (bits 24 to 26) in RENV3> CU1B (bit 24) = 1: Enable COUNTER1 (command position) CU2B (bit 25) = 1: Enable COUNTER2 (mechanical position) CU4B (bit 26) = 1: Enable COUNTER3 (general-purpose, deflection)	[RENV3] (WRITE) 31 24 - - - - - n n n
Specify the counting conditions for COUNTER3 <Set BSYC (bit 13) in RENV3> 1. Enable COUNTER3 (general-purpose, deflection) only while operating (#BSY = L).	[RENV3] (WRITE) 15 8 - - n - - - - -

8-11. Comparator

8-11-1. Comparator types and functions

This LSI has 3 circuits of 28-bit comparators. It compares the values set in the RCMP1 to 3 registers with the counter values.
Comparators 1 to 3 can be used as comparison counters and can be assigned as COUNTERS 1 to 3.
There are many comparison methods and 3 processing methods that can be used when the conditions are met.
Specify the comparator conditions in the RENV4 (environment 4) registers. By using these comparators, you can perform the following.

- Generate an interrupt and output the comparison result externally.
- Immediate stop and deceleration stop operations.
- Software limit function using Comparators 1 and 2.
- Detect out of step stepper motors using COUNTER3 (deflection) and a comparator.
- Output a synchronous signal (IDX) using COUNTER3 (general-purpose) and a Comparator 3.

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

Comparison data	Comparator 1		Comparator 2		Comparator 3	
		C1C0 to 1		C2C0 to 1		C3C0 to 1
COUNTER1 (command position)	O	"00"	O	"00"	O	"00"
COUNTER2 (mechanical position)	O	"01"	O	"01"	O	"01"
COUNTER3 (deflection)	O	"10"	O	"10"	O	"10"
Major application	+SL		-SL		Out-of-step detection, IDX output	

- O: Comparison possible.
- +SL, -SL are used for software limits.
- If COUNTER3 (deflection), that was specified as deflection counter, is selected as comparison counter, the LSI will compare the absolute value of the counter with the comparator data. (Absolute value range: 0 to 32,767)
- Choose the comparison data from C1C0 to 1 (bits 0 to 1), C2C0 to 1 (bits 8 to 9), and C3C0 to 1 (bits 16 to 17).

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

Comparison method	Comparator 1		Comparator 2		Comparator 3	
		C1S0 to 2		C2S0 to 2		C3S0 to 3
Comparator = Comparison counter (regardless of count direction)	O	"001"	O	"001"	O	"0001"
Comparator = Comparison counter (Count up only)	O	"010"	O	"010"	O	"0010"
Comparator = Comparison counter (count down only)	O	"011"	O	"011"	O	"0011"
Comparator > Comparison counter	O	"100"	O	"100"	O	"0100"
Comparator < Comparison counter	O	"101"	O	"101"	O	"0101"
Use for software limits	O	"110"	O	"110"		
IDX (synchronous signal) output (regardless of counting direction)					O	"1000"
IDX (synchronous signal) output (count up only)					O	"1001"
IDX (synchronous signal) output (count down only)					O	"1010"

- O: Comparison possible. Blank: Comparison not possible.
- Comparator 3 must not have C3S0 to 3 set to a value of 0111. Setting any of the values may result in failing to satisfy the comparison conditions.
- When C3S0 to 3 = 1000 to 1010 for Comparator 3 <IDX (synchronous signal) output>, select COUNTER3 (general-purpose, deflection) for use as the comparison counter. Other counters cannot be used for this function. Enter a positive value for the comparator setting.
- When using the comparator function as a software limit, Comparator 1 will be the positive limit value. Then, the PCL device (G9003) looks for the "Comparator < Comparison counter." Comparator 2 will be the negative limit value. Then, the PCL device (G9003) compares "Comparator > Comparison counter." Select COUNTER1 (command position) as the comparison counter.
- Choose the comparison data from C1C0 to 2 (bits 2 to 4), C2C0 to 2 (bits 10 to 12), and C3C0 to 3 (bits 18 to 21) in the RENV4.

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

Processing method when the conditions are met	Comparator 1	Comparator 2	Comparator 3
	C1D0 to 1	C2D0 to 1	C3D0 to 1
Do nothing	"00"	"00"	"00"
Immediate stop operation	"01"	"01"	"01"
Deceleration stop operation	"10"	"10"	"10"
Change operation data to pre-register data	"11"	"11"	"11"

- The bit assignments to select a processing method are as follows.
  C1D0 to 1 (RENV4 bits 5 to 6), C2D0 to 1 (RENV4 bits 13 to 14), and C3D0 to 1 (RENV4 bits 22 to 23).

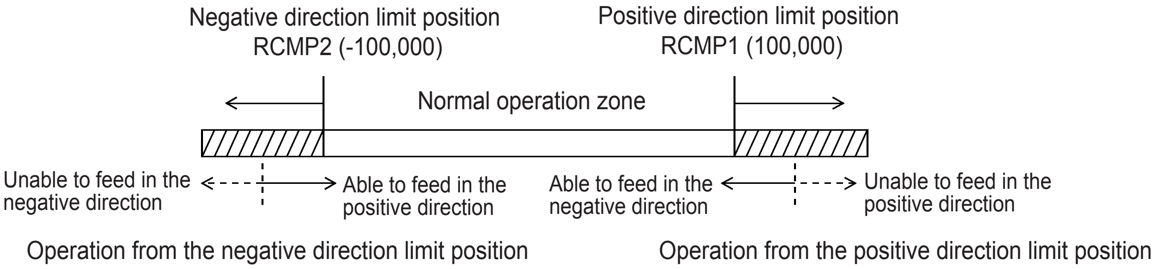
Set an event interrupt cause <Set IRC1 to 3 (bit 5 to 7) in RIRQ> IRC1 (bit 5) = 1: Generate an interrupt when the Comparator 1 conditions are satisfied. IRC2 (bit 6) = 1: Generate an interrupt when the Comparator 2 conditions are satisfied. IRC3 (bit 7) = 1: Generate an interrupt when the Comparator 3 conditions are satisfied.	[RIRQ] (WRITE) 7 0 n n n - - - - -
Read the event interrupt cause <ISC1 to 3 (bit 5 to 7) in RIST> IRC1 (bit 5) = 1: When the Comparator 1 conditions are satisfied. IRC2 (bit 6) = 1: When the Comparator 2 conditions are satisfied. IRC3 (bit 7) = 1: When the Comparator 3 conditions are satisfied.	[RIST] (READ) 7 0 n n n - - - - -
Read the comparator condition status <SCP1 to 3 (bits 20 to 22) in RSTS> SCP1 (bit 20) = 1: When the Comparator 1 conditions are satisfied. SCP2 (bit 21) = 1: When the Comparator 2 conditions are satisfied. SCP3 (bit 22) = 1: When the Comparator 3 conditions are satisfied.	[RSTS] (READ) 23 16 - n n n - - - -
Read the error interrupt cause <ESC1 to 3 (bits 0 to 2) in REST> ESC1 (bit 0) = 1: When stopped by a match of the comparator 1 conditions. (+SL) ESC2 (bit 1) = 1: When stopped by a match of the comparator 2 conditions. (-SL) ESC3 (bit 2) = 1: When stopped by a match of the comparator 3 conditions.	[REST] (READ) 7 0 - - - - - n n n

8-11-2. Software limit function

A software limit function can be set up using comparators 1 and 2.
Select COUNTER1 (command position) as a comparison counter for comparators 1 and 2.
Use Comparator 1 for a positive direction limit and Comparator 2 for a negative direction limit to stop the axis based on the results of the comparator and the operation direction.
When the software limit function is used the following process can be executed.
1) Stop pulse output immediately
2) Decelerate and then stop pulse output
While using the software limit function, if a deceleration stop is selected as the process to use when the comparator conditions are met (C1D, C2D), when an axis reaches the software limit while in a high speed start (command 0052h), that axis will stop using deceleration. When some other process is specified for use when the conditions are met, or while in a constant speed start, that axis will stop immediately.
If a software limit is ON while writing a start command, the axis will not start to move in the direction in which the software limit is enabled. However, it can start in the opposite direction.

[Setting example]

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


Negative direction limit position
RCMP2 (-100,000)
Positive direction limit position
RCMP1 (100,000)
Normal operation zone
Unable to feed in the negative direction
Able to feed in the positive direction
Able to feed in the negative direction
Unable to feed in the positive direction
Operation from the negative direction limit position
Operation from the positive direction limit position

Setting the comparison method for Comparator 1 <Set C1S0 to C1S2 (bits 2 to 4) in RENV4> 001: RCMP1 data = Comparison data (Regardless of count direction) 010: RCMP1 data = Comparison data (While counting up) 011: RCMP1 data = Comparison data (While counting down) 100: RCMP1 data > Comparison counter 101: RCMP1 data < Comparison counter 110: Use as a positive direction software limit (RCMP 1< COUNTER 1) Others: Always assumes that the comparison conditions are not met.	[RENV4] (WRITE) 7 0 - - - n n n - -
Specify the process to use when the Comparator 1 conditions are met <Set C1D0 to C1D1 (bits 5 to 6) in RENV4> 01: Immediate stop 10: Deceleration stop	[RENV4] (WRITE) 7 0 - n n - - - - -
Specify the comparison method for Comparator 2 <Set C2S0 to C2S2 (bits 10 to 12) in RENV4> 001: RCMP2 data = Comparison data (Regardless of count direction) 010: RCMP2 data = Comparison data (While counting up) 011: RCMP2 data = Comparison data (While counting down) 100: RCMP2 data > Comparison counter 101: RCMP2 data < Comparison counter 110: Use as a negative direction software limit. (RCMP 2 > COUNTER 1) Others: Always assumes that the comparison conditions are not met.	[RENV4] (WRITE) 15 8 - - - n n n - -

Specify the process to use when the Comparator 2 conditions are met <Set C2D0 to C2D1 (bits 13 to 14) in RENV4> 01: Immediate stop 10: Deceleration stop	[RENV4] (WRITE) 15 8 - n n - - - - -

Note: The parts in bold face mean that the settings in the example above are allowed.

8-11-3. Out of step stepper motor detection function for stepper motors

If the deflection counter value controlled by the motor command pulses and the feed back pulses from an encoder on a stepper motor exceed the maximum deflection value, the LSI will declare that the stepper motor is out of step. The LSI monitors stepper motor operation using COUNTER3 (general-purpose, deflection counter) and a comparator.
The process which takes place after an out of step condition is detected can be selected from the table. [Processing method to use when the comparator conditions are satisfied].
For this function, use an encoder with the same resolution as the stepper motor.
COUNTER3 (general-purpose, deflection) can be cleared by writing a set command to the deflection counter.
There are two methods for inputting a feedback signal: Input 90 phase difference signals (1x, 2x, 4x) on the EA/EB terminals, input two sets of positive and negative pulses.
When both the EA and EB signals change at the same time, the device generates an interrupt.

[Setting example]
RENV3 = 00001000h: Set COUNTER3 as an EA/EB deflection counter.
RENV4 = 00560000h: Satisfy the conditions of Comparator 3 < COUNTER3 (deflection)
Stop immediately when the conditions are satisfied.
RCMP3 = 32: The maximum deflection value is "32" pulses.
RIRQ = 00000080h: Generate an interrupt when the comparator 3 conditions are met.

Set COUNTER3 (deflection) input <Set CI30 to 32 (bits 10 to 12) in RENV3> 000: Output pulse 001: EA/EB input 010: PA/PB input 011: 1/4096 division of the internal reference clock (CLK = 40 MHz) 100: Count deflection using output pulses and the EA/EB input 101: Count deflection using output pulses and the PA/PB input 110: Count deflection using the EA/EB, PA/PB inputs.	[RENV3] (WRITE) 15 8 - - - n n n - -
Specify the EA/EB input <Set EIM0 to 1 (bits 9 to 10) in RENV2> 00: 90 phase difference, 1x 01: 90 phase difference, 2x 10: 90 phase difference, 4x 11: 2-pulse mode	[RENV2] (WRITE) 15 8 - - - - - n n -
Specify the EA/EB input count direction <Set EDIR (bit 11) in RENV2> 0: When the EA phase is leading, or count up on the EA rising edge. 1: When the EB phase is leading, or count up on the EB rising edge	[RENV2] (WRITE) 15 8 - - - - n - - -
Read the EA/EB input error <ESEE (bit 13) in REST> 1: An EA/EB input error has occurred.	[REST] (READ) 15 8 - - n - - - - -
Set the EA/EB/EZ input filter <Set EINF (bit 8) in RENV2> 0: Turn OFF filter function 1: Turn ON the filter function (Ignore input signals shorter than 6 CLK cycles)	[RENV2] (WRITE) 15 8 - - - - - - - n
Enable/disable EA/EB input <Set EOFF (bit 17) in RENV2> 0: Enable EA/EB input 1: Disable EA/EB input (EZ input is left enabled.)	[RENV2] (WRITE) 23 16 - - - - - - n -
Counter reset command <CUN3R: Control command> Clear COUNTER3 (general-purpose, deflection) to zero.	[Control command] 0022h

Note: The parts in bold face mean that the settings in the example above are allowed.

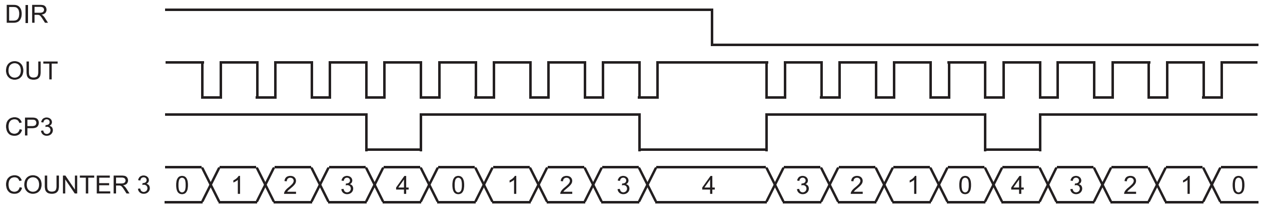
8-11-4. IDX (synchronous) signal output function

Using Comparator 3 and COUNTER3 (general-purpose, deflection counter) that was specified to "general-purpose counter," the device can output signals to the #CP3 terminal at specified intervals. Setting C3C0 and C3C1 to "10" (COUNTER 3) and setting C3S0 and C3S3 to "1000 to "1010" (the IDX output), the PCL device (G9003) can be used for IDX (index) operation.
The counter range of COUNTER3 will be 0 to the value set in RCMP3 [Max. 32,767]. If counting down from 0 the lower limit will be the value set in RCMP3, and if counting up from the value set in RCMP3 the limit will be 0.
The input for COUNTER3 can be set to CI30 to CI32 in RENV3.

Set COUNTER3 (deflection) <Set CI30 to 32 (bits 10 to 12) in RENV3> 000: Output pulse 001: EA/EB input 010: PA/PB input 011: 1/4096 division of the internal reference clock (CLK = 40 MHz) 100: Count deflection using output pulses and the EA/EB input 101: Count deflection using output pulses and the PA/PB input 110: Count deflection using the EA/EB, PA/PB inputs.	[RENV3] (WRITE) 15 8 - - - n n n - -
Select Comparator 3 comparison counter <Set C3C0 to 1 (bits 16 to 17) in RENV4> 00: COUNTER1 (command position) 01: COUNTER2 (machine position) 10: COUNTER3 (general-purpose, deflection) 11: Always treated as no comparison conditions specified	[RENV4] (WRITE) 23 16 - - - - - - n n
Set the comparison method for comparator 3 <Set C3S0 to 3 (bits 18 to 21) in RENV4> 0001: RCMP3 data = Comparison data (regardless of count direction) 0010: RCMP3 data = Comparison data (while counting up) 0011: RCMP3 data = Comparison data (while counting down) 0100: RCMP3 data > Comparison counter 0101: RCMP3 data < Comparison counter 0111: Prohibited 1000: Use as an IDX (synchronous) signal output (regardless of count direction) 1001: Use as an IDX (synchronous) signal output (while counting up) 1010: Use as an IDX (synchronous) signal output (while counting down) Others: Always treated as no comparison conditions specified	[RENV4] (WRITE) 23 16 - - n n n n - -

Note: The parts in bold face mean that the settings in the example above are allowed.

Output example:
Regardless of the feed direction, the PCL device (G9003) will output the IDX signal using negative logic for the output pulses. (Counting range: 0 to 4.)
Settings: RENV3 = 00000000h, RENV4 = 00220000h, RCMP4 = 4



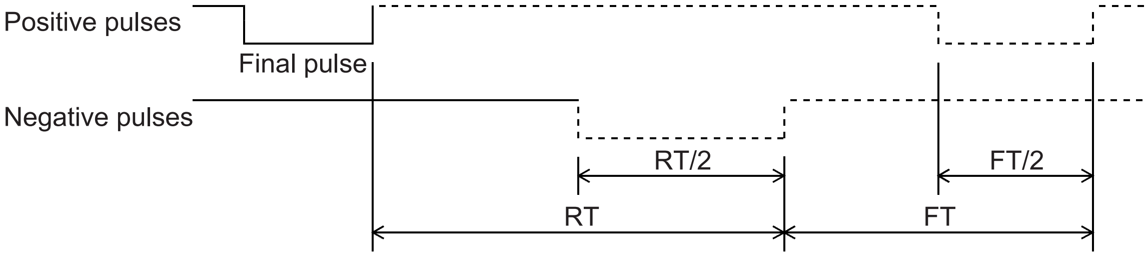
8-12. Backlash correction

This LSI has backlash correction functions. This function outputs the number of command pulses specified for the correction value in the speed setting in the RFA (correction speed) register.
The backlash correction is performed each time the direction of operation changes. The correction amount and method is specified in the RENV5 (environment setting 5) register.
The operation of the counter (COUNTER 1 to 3) can be set using the RENV3 (environment setting 3) register.

Enter the correction value <BR0 to 11 (bits 0 to 11) in RENV5> Backlash correction amount value (0 to 4,095)	[RENV5] (WRITE) 15 8 - - - - n n n n 7 0 - n n n n n n n
Set the correction method <ADJ (bits 12) in RENV5> 0: Turn the correction function OFF 1: Backlash correction	[RENV5] (WRITE) 15 8 - - - n - - - -
Action for backlash/slip correction <CU1B to 3B (bit 24 to 26) in RENV3> CU1B (bit 24) = 1: Enable COUNTER1 (command position) CU2B (bit 25) = 1: Enable COUNTER2 (mechanical position) CU4B (bit 26) = 1: Enable COUNTER3 (general-purpose, deflection)	[RENV3] (WRITE) 31 24 - - - - - n n n

8-13. Vibration restriction function

This LSI has a function to restrict vibration when stopping by adding one pulse of reverse operation and one pulse of forward operation shortly after completing a command pulse operation.
Specify the output timing for additional pulses in the RENV6 (environment setting 6) register.
When both the reverse timing (RT) and the forward timing (FT) are non zero, the vibration restriction function is enabled.
The dotted lines below are pulses added by the vibration restriction function. (An example in the positive direction)



Specify the reverse operation timing <Set RT0 to 15 (bits 0 to 15) in RENV6> RT range: 0 to 65,535 The units are 64x the reference clock frequency (approx. 1.6 μsec when CLK = 40 MHz) Settable range: 0 to approx. 0.1 sec.	[RENV6] (WRITE) 15 8 n n n n n n n n 7 0 n n n n n n n n
Specify the forward operation timing <Set FT0 to 15 (bits 16 to 31) in RENV6> FT range: 0 to 65,535 The units are 64x the reference clock frequency (approx. 1.6 μsec when CLK = 40 MHz) Settable range: 0 to approx. 0.1 sec.	[RENV6] (WRITE) 31 24 n n n n n n n n 23 16 n n n n n n n n

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

8-14. Excitation sequence for stepper motors

This LSI can generate 2-2 phase and 1-2 phase excitation sequences for 2-phase stepper motors to provide unipolar and bipolar driving.
The LSI uses the #BSY, #FUP, #FDW, and #MVC signal terminals (normally used for monitor operation status), and the #BSY/PH1, #FUP/PH2, #FDW/PH3, #MVC/PH4 common terminals to output these signal sequences. To change between monitor and output, set MPH in the RMD (operation mode) register.

When the PH1, PH2, PH3, and PH4 are specified for use as excitation sequence output terminals, the output can be masked (set them all LOW) using MMPH in the RMD register.
To change between unipolar and bipolar signals, set MUB in the RMD register. To change between 2-2 and 1-2 phase excitation, set MFH in the RMD register.
While the LSI is producing an excitation signal for a single phase in 1-2 phase excitation (steps 1, 3, 5, and 7 in the table), if you change to 2-2 phase excitation, the LSI will change to 2 phase excitation status starting with the next output pulse.
By reading the RSTS (extension status) register, you can monitor the excitation sequence status.

[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
(-) Operation direction (+)					

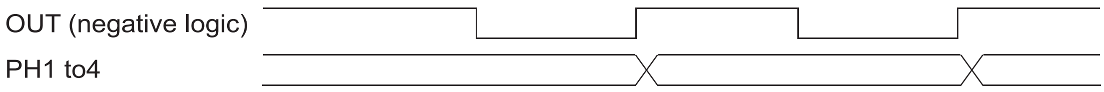
1-2 phase excitation									
STEP	0	1	2	3	4	5	6	7	0
PH1	H	H	H	L	L	L	L	L	H
PH2	L	L	H	H	H	L	L	L	L
PH3	L	L	L	L	H	H	H	L	L
PH4	H	L	L	L	L	L	H	H	H
(-) Operation direction (+)									

[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
(-) Operation 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
(-) Operation direction (+)									

[Change the timing of the excitation sequence]

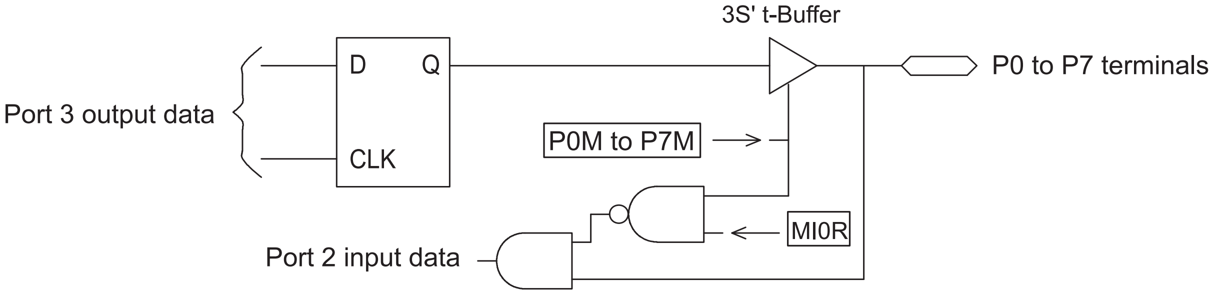


Selection of the #BSY/PH1, #FUP/PH2, #FDW/PH3, #MVC/PH4 terminal output <Set MPH (bit 20) in the RMD> 0: Output actual #BSY, #FUP, #FDW, and #MVC signals. 1: Output actual PH1, PH2, PH3, and PH4 signals.	[RMD] (WRITE) 23 16 - - - n - - - -
Mask PH1, PH2, PH3, and PH4 signals <Set MMPH (bit 21) in the RMD> 0: Output "LOW" level from PH1, PH2, PH3, and PH4 1: Output actual PH1, PH2, PH3, and PH4 signals.	[RMD] (WRITE) 23 16 - - n - - - - -
Select the excitation method using PH1, PH2, PH3, and PH4 signals <Set MMPH (bit 22) in the RMD> 0: Output excitation sequence for 2 phase unipolar 1: Output excitation sequence for 2 phase bipolar	[RMD] (WRITE) 23 16 - n - - - - - -
Select the excitation method using PH1, PH2, PH3, and PH4 signals <Set MMPH (bit 22) in the RMD> 0: Output excitation sequence for full step 1: Output excitation sequence for half step	[RMD] (WRITE) 23 16 n - - - - - - -

Read the excitation sequence signal <Set SPH1 to 4 (bits 24 to 27) in the RSTS>
SPH1 = 1: PH1 is ON (HIGH) status
SPH2 = 1: PH2 is ON (HIGH) status
SPH3 = 1: PH3 is ON (HIGH) status
SPH4 = 1: PH4 is ON (HIGH) status
	[RSTS] (READ)
	31 24
	- - - - n n n n

8-15. General-purpose I/O terminals (P0 to P7)

Although these ports are set as input ports by default, by setting P0M to P7M (bits 0 to 7) in RENV2, they can be set individually for input or output.
The internal arrangement of these terminals is roughly as shown below. Although they are used primarily as input terminals, they can be set to act as a latched output circuit. If they are changed to function as output terminals, the LSI will output a latched status. (The initial status of the latch output in the figure below is Q = LOW.)



When they are set for use as output ports, among writing data to Port 3 of the I/O port, their corresponding bits are set to "1" will go HIGH.
The terminal status can be checked by reading Port 2.
The status data can be masked when the output port is selected using MIOR (bit 24) in the RMD.
To enable the input-change interrupt on the center device (G9001A) using Port 2, set MI0R = 0. Then, even if the output port status is changed, the corresponding port 2 bit will also change so that an interrupt occurs. When MI0R is 1, if the output port status changes, the corresponding port 2 bit will be left at "0."

Setting the general I/O terminals <P0M to P7M (bits 0 to 7) in RENV2> 0: Make the terminal that corresponds to the bit an input terminal. 1: Make the terminal that corresponds to the bit an output terminal.	[RENV2] (WRITE) 7 0 n n n n n n n n
Select the monitoring method for the output setting bits in a general-purpose I/O port. <MIOR (bits 24) in RMD> 0: Read the setting of the output bits on Port 2. 1: Regardless of the setting of the output bits, the bits corresponding to Port 2 will be "0."	[RMD] (WRITE) 31 24 - - - - - - - n
Set the general-purpose I/O terminal data to be output <Set IOPOB (bits 0 to 7) on I/O Port 3> 0: LOW level (when specified as output port) 1: HIGH level (when specified as output port)	[PORT 3] (WRITE) 7 0 n n n n n n n n
Read the general-purpose I/O terminal input data <Set IOPIB (bits 0 to 7) on I/O Port 2> 0: LOW level 1: HIGH level	[PORT 2] (READ) 7 0 n n n n n n n n

8-16. Interrupt output

This LSI can output an interrupt: There are 14 types of errors, 14 types of events, and change from operating to stop. All of the error causes will always output an interrupt. Each of the event causes can be set in the RIRQ register.
If any of the interrupts above occurs, the Main Status bit 0 (SINT) changes from 0 to 1.
When this happens, the center device can generate an interrupt.

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

The interrupt is output continuously until all of the causes of the interrupt have been cleared.
An interrupt caused by an error is cleared by writing a "REST (error cause) register read command." An interrupt caused by an event is cleared by writing a "RIST (event cause) register read command." A Stop interrupt is cleared by writing to the main status. The stop interrupt is cleared by writing a reset command (0008h) for the SEND interrupt.

By setting RENV1 (Environment Setting Register 1), you can choose not to reflect the occurrence of SEND interrupts in the Main Status bit 0 (SINT) or to reset the SEND interrupt when starting an operation.

The causes of an interrupt can be evaluated as follows below.

1) Read the main status of the X axis and check whether bits 1, 2, or 3 is "1."
2) If bit 1 (SEND) is "1," a Stop interrupt occurs. Reset the device using reset command (0008h).
3) If bit 2 (SERR) is "1," read the RESET register to identify the cause of the interrupt.
4) If bit 3 (SEVT) is "1," read the RIST register to identify the cause of the interrupt.

With these procedures, you can identify an interrupt cause and turn OFF the occurrence of the interrupt.

Note 1: Using the interrupt routine, if the center device tries to read the register, the contents of the FIFO buffer in the centerl device may change. If an interrupt occurs while the main routine is reading or writing registers, and the interrupt routine starts, the main routine may produce an error. Therefore, the interrupt routine should execute a PUSH/POP on FIFO buffer.

Note 2: While processing in steps 1) to 4) above, it is possible that another interrupt may occur on an axis whose process has completed. After the CPU has recovered from an interrupt, read the Main Status and check if an interrupt has occurred with the PCL device (G9003). Then end the interrupt routine.

Note 3: The I/O port with the Main Status bit that is used for the interrupt is refreshed by the cyclic transfer. It is also refreshed by the transient transfer of data communications.

The Main Status bit 0 (SINT) can be masked by setting the RMD (operation mode) register.
When masked (MINT = 1 in the RMD), even though the status changes, bit 0 (SINT) in the Main Status will remain "0" and will not change to "1."
When an interrupt occurs, if the output mask is turned OFF (MINT = 0 in the RMD), bit 0 (SINT) in the Main Status will change to "1."

Read the interrupt status <SINT (bit0), SEND (1), SERR (bit 2), SEVT (3) in MSTB0> SINT (bit 0) = 1: Set to 1 when SEND = 1 or SERR = 1 or SEVT = 1. SEND = 1: Set to 1 when stopped. Set to 0 by an interrupt reset command (0008h). SERR = 1: Becomes 1 when an error interrupt occurs. Becomes 0 by reading REST. SIVT = 1: Becomes 1 when an event interrupt occurs. Becomes 0 by reading RIST.	[MSTB0] (READ) 7 0 - - - - n n n n
Set the interrupt mask <MINT (bit 19) in RMD> 1: Mask SINT (bit 0) in the Main Status.	[RMD] (WRITE) 23 16 - - - - n - - -
Set the mask on the stop interrupt <SEDM (bit 27) in RENV1> 0: Enable stop interrupts (reflected to SINT) 1: Mask the stop interrupt (change only the SEND status)	[RENV1] (WRITE) 31 24 - - - - n - - -
Set to reset when a stop interrupt occurs. <SEDR (bit 28) in RENV1> 1: Reset the stop interrupt bit (SEND) when an interrupt occurs.	[RENV1] (WRITE) 31 24 - - - n - - - -
Reset the stop interrupt <INTRS: Control command> Reset the SEND bit (stop interrupt)	[Control command] 0008h
Read the cause of the error interrupt <RREST: Read out command> Copy the data in the REST register (error interrupt cause) to BUF.	[Read command] 00F2h
Read the event interrupt cause <RRIST: Read out command> Read the data in the RIST register (event interrupt cause) to BUF.	[Read command] 00F3h
Set the event interrupt cause <WRIRQ: Control command> Write the FIFO data to the RIRQ register (event interrupt cause).	[Write command] 00ACh

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

Error interrupt cause	Cause (REST)	
	Bit	Bit name
Stopped by Comparator 1 conditions being satisfied (+SL)	0	ESC1
Stopped by Comparator 2 conditions being satisfied (-SL)	1	ESC2
Stopped by Comparator 3 conditions being satisfied	2	ESC3
Stopped by turning ON the +EL input	3	ESPL
Stopped by turning ON the -EL input	4	ESML
Stopped by turning ON the ALM input	5	ESAL
Stopped by turning ON the #STP input	6	ESSP
Stopped by turning ON the #EMG input	7	ESEM
Deceleration stopped by turning ON the SD input	8	ESSD
Stopped by an overflow of PA/PB input buffer counter occurrence	9	ESPO
When stopped by a communication error.	10	ESNT
(Always 0)	11	Not defined
When a position override cannot be executed.	12	ESOR
An EA/EB input error occurred (does not stop).	13	ESEE
An PA/PB input error occurred (does not stop).	14	ESPE

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

Event interrupt cause	Set cause (RIRQ)		Cause (RIST)	
	Bit	Bit name	Bit	Bit name
Automatic stop	0	IREN	0	ISEN
When acceleration starts	1	IRUS	1	ISUS
When acceleration ends	2	IRUE	2	ISUE
When deceleration starts	3	IRDS	3	ISDS
When deceleration ends	4	IRDE	4	ISDE
When the Comparator 1 conditions are satisfied	5	IRC1	5	ISC1
When the Comparator 2 conditions are satisfied	6	IRC2	6	ISC2
When the Comparator 3 conditions are satisfied	7	IRC3	7	ISC3
When the counter value is reset by a CLR signal input	8	IRCL	8	ISCL
When the counter value is latched by an LTC input	9	IRLT	9	ISLT
When the counter value is latched by an ORG input	10	IROL	10	ISOL
When the SD input is turned ON	11	IRSD	11	ISSD
When the #STA input is turned ON	12	IRSA	12	ISSA

9. How to calculate the communication cycle time

The calculations of the communication cycle time can be classified as follows:
K: Communication speed figure

Communication speed (Mbps)	K
20	1
10	2
5	4
2.5	8

N: Number of local devices connected
B: Number of bytes of data to send (when sending 2 bytes of data: B = 2)

9-1. Time required for one cycle

Basic item	Required time (μs)
Communication time required per local device (CT)	7.7 x K

Cycle time = (CT + 7.4) x N (μs)

Ex.: Calculating the cycle time with a communication speed of 20 Mbps and 30 local devices.
(7.7 x 1 + 7.4) x 30 = 453 μs

9-2. Time required for one complete data communication

There are two types of data communications as follows:
1) When there is data in the response from a local device (the data length is variable).
2) When there is no data in the response from a local device.

Basic item	Required time (μs)
Data sending time (ST)	(B x 0.6 + 3.25) x K
Response time with data (JT)	(B x 0.6 + 5.65) x K
Response time without data (JT)	5.05 x K

One complete data communication cycle = ST + JT + 7.4 (μs)

9-3. Total cycle time (including data communication)

The total time can be obtained by adding the data communication times to the ordinary communication cycle time.

Ex.1: Communication speed = 20 Mbps, 34 local devices are connected, and on 4 occasions the data communication consisted of 2 bytes for sending and 6 bytes for receiving.
Cycle time = Cyclic time + (Data communication time) x Number of times of data communication
= (7.7 x 1 + 7.4) x 34 + {(2 x 0.6 + 3.25) x 1 + (6 x 0.6 + 5.65) x 1 + 7.4} x 4
= 513.4 + 21.1 x 4
= 597.8 μs

Note: The formula above contains some margin for error. In actual operation, a shorter total time can be obtained.

10. Electrical Characteristics

10-1. Absolute maximum ratings

(V_{SS}=0V)

Item	Symbol	Rating	Unit
Power supply voltage	V_{DD}	-0.3 to +5.0	V
Input voltage	V_{IN}	-0.3 to V_{DD} +0.3	V
Input voltage (5V I/F)	V_{IN}	-0.3 to +7.0	V
Input current	I_{IN}	±10	mA
Storage temperature	T_{STO}	-40 to +125	°C

10-2. Recommended operating conditions

(V_{SS}=0V)

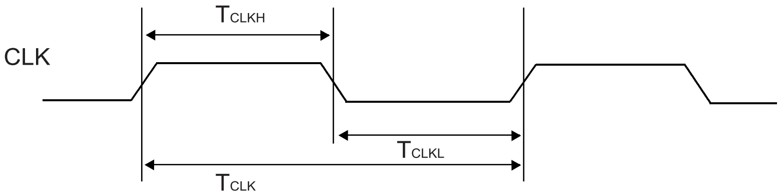
Item	Symbol	Rating	Unit
Power supply voltage	V_{DD}	+3.0 to +3.6	V
Input voltage	V_{IN}	V_{DD}	V
Input voltage (5V I/F)	V_{IN}	to 5.5	V
Storage temperature	T_a	-40 to +85	V

10-3. DC characteristics

Item	Symbol	Condition	Min.	Max.	Unit
Current consumption	I_{dd}	CLK = 80 MHz, Output 6MHz		60	mA
Output leakage current	I_{OZ}		-10	10	μA
Input capacitance				6.8	pF
LOW input current	I_{IL}		-10	10	μA
		With a pull up resistor	-200	-10	μA
HIGH input current	I_{HL}		-10	10	μA
		With a pull down resistor	10	200	V
LOW input current	V_{IL}	Terminals other than CLK		0.8	V
		CLK terminal		VDDx0.2	V
HIGH input current	V_{IH}	Terminals other than CLK	2.0		V
		CLK terminal	VDDx0.8		V
LOW output voltage	V_{OL}	I_{OL} = 1 μA		0.05	V
		I_{OL} = 4 mA		0.4	V
HIGH output voltage	V_{OH}	I_{OH} = -1 μA	VDD-0.05		V
		I_{OL} = -4 mA	2.4		V
LOW output current	I_{OL}	V_{OL} = 0.4 V		4	mA
		Bi-directional I/F, V_{OL} = 0.4 V		8	mA
HIGH output current	I_{OH}	V_{OH} = 2.4 V	-4		mA
		Bi-directional I/F, V_{OH} = 2.4 V	-8		mA
Internal pull up resistance	R_{UP}		25	500	K-ohm

10-4. AC characteristics

10-4-1. System clock



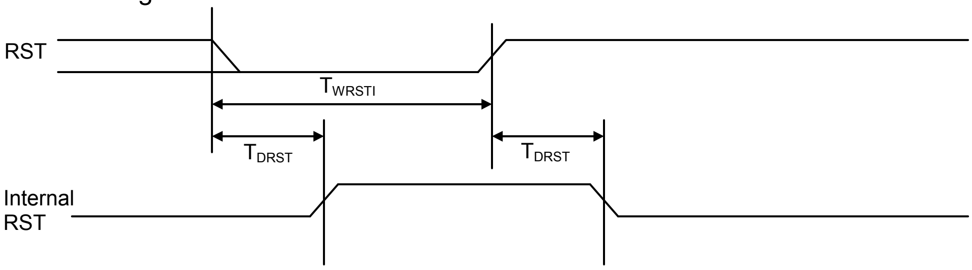
1) When setting CKSL = L

Item	Symbol	Min.	Max.	Unit
Frequency	f_{CLK}	-	40	MHz
Cycle	T_{CLK}	25	-	ns
HIGH duration	T_{CLKH}	10	15	ns
LOW duration	T_{CLKL}	10	15	ns

2) When setting CKSL = H

Item	Symbol	Min.	Max.	Unit
Frequency	f_{CLK}	-	80	MHz
Cycle	T_{CLK}	12.5	-	ns
HIGH duration	T_{CLKH}	-	-	ns
LOW duration	T_{CLKL}	-	-	ns

10-4-2. Reset timing



Item	Symbol	Min.	Max.	Unit
Reset length	T_{WRSTI}	10	-	Clock cycles
Delay time	T_{DRST}	-	10	Clock cycles

Note 1: The PCL device (G9003) is ready to use after the internal #RST goes LOW.

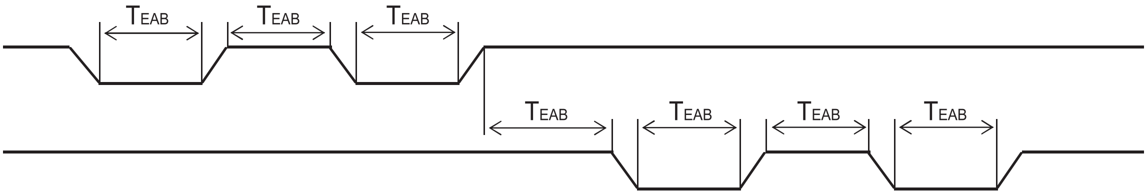
Note 2: The reset signal must last at least 10 cycles of the system clock.
While resetting, Make sure the clock signal is continuously available to the device.
If the clock is stopped while resetting, the device cannot be reset normally.

10-5. Operation timing

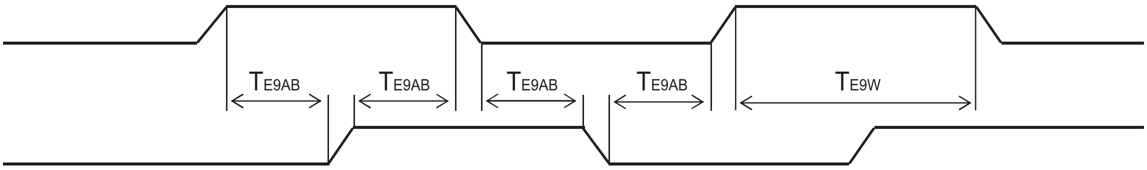
Item		Symbol	Condition	Min.	Max.	Unit
#RST input signal width			Note 1	$10T_{CLK}$		ns
CLR input signal width			Note 2	$4T_{ICK}$		ns
EA, EB input signal width (2 pulses)		T_{EAB}	Note 3	$2T_{ICK}$ ($6T_{ICK}$)		ns
EA, EB input signal width (90^o)		T_{E9AB}		$2T_{ICK}$		ns
EA, EB input signal width (90^o)		T_{E9W}	Note 3	$4T_{ICK}$ ($6T_{ICK}$)		ns
EZ input signal width			Note 3	$2T_{ICK}$ ($6T_{ICK}$)		ns
PA, PB input signal width (2 pulses)		T_{PAB}	Note 3	$2T_{ICK}$ ($6T_{ICK}$)		ns
PA, PB input signal width ($90\ ^o$)		T_{P9AB}		$2T_{ICK}$		ns
PA, PB input signal width ($90\ ^o$)		T_{P9W}	Note 3	$4T_{ICK}$ ($6T_{ICK}$)		ns
ALM input signal width			Note 5	$4T_{ICK}$		ns
INP input signal width			Note 5	$4T_{ICK}$		ns
ERC output signal width			RENV1 bit 12 to 14 = 000	254 x $2T_{ICK}$	254 x $2T_{ICK}$	ns
			RENV1 bit 12 to 14 = 001	254 x $16T_{ICK}$	254 x $16T_{ICK}$	
			RENV1 bit 12 to 14 = 010	254 x $64T_{ICK}$	254 x $64T_{ICK}$	
			RENV1 bit 12 to 14 = 011	254 x $256T_{ICK}$	254 x $256T_{ICK}$	
			RENV1 bit 12 to 14 = 100	254 x $2048T_{ICK}$	254 x $2048T_{ICK}$	
			RENV1 bit 12 to 14 = 101	254 x $8192T_{ICK}$	254 x $8192T_{ICK}$	
			RENV1 bit 12 to 14 = 110	254 x $16384T_{ICK}$	254 x $16384T_{ICK}$	
			RENV1 bit 12 to 14 = 111	LEVEL output		
+EL, -EL input signal width			Note 5	$4\ T_{ICK}$		ns
SD input signal width			Note 5	$4\ T_{ICK}$		ns
ORG input signal width			Note 5	$4\ T_{ICK}$		ns
PCS input signal width				$4\ T_{ICK}$		ns
LTC input signal width				$4\ T_{ICK}$		ns
#STA	Output signal width			$16\ T_{ICK}$		ns
	Input signal width			$10\ T_{ICK}$		ns
#STP	Output signal width			$16\ T_{ICK}$		ns
	Input signal width			$10\ T_{ICK}$		ns
#BSY signal ON delay time		T_{SOEBSY}			$-12\ T_{ICK}$	ns
		T_{STABSY}			$14\ T_{ICK}$	ns
Start delay time		T_{SOEPLS}			$8\ T_{ICK}$	ns
		T_{STAPLS}			$34\ T_{ICK}$	ns
Output port delay time		T_{SOEPRT}			$-26T_{ICK}$	ns

Note 1: "T_{ICK}" in the table above means one cycle (25 nS) of the internal clock 40 MHz.
Note 2: The actual CLK input signal is 10 cycles longer while the RST terminal is LOW.
Note 3: If the input filter is ON < EINF (bit 18) = 1 in RENV2 >, the minimum time will be $6T_{CLK}$.
Note 4: If the input filter is ON < PINF (bit 19) = 1 in RENV2 >, the minimum time will be $6T_{CLK}$.
Note 5: If the input filter is ON < FLTR (bit 26) = 1 in RENV1 >, the minimum time will be $160T_{CLK}$.

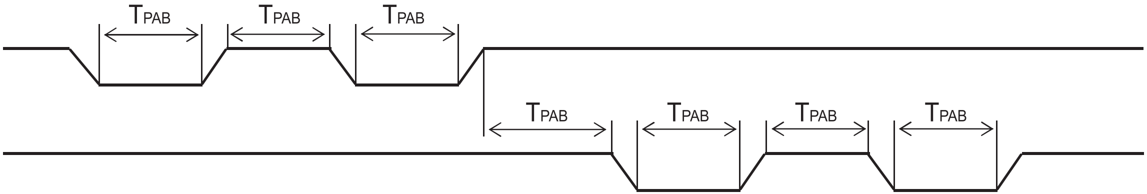
1) When the EA, EB inputs are in the 2-pulse mode



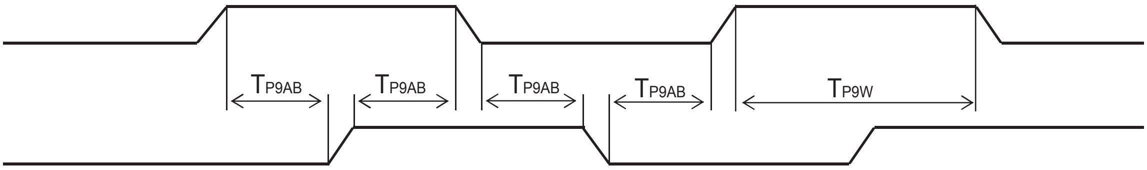
2) When the EA, EB inputs are in the 90 ° phase-difference mode



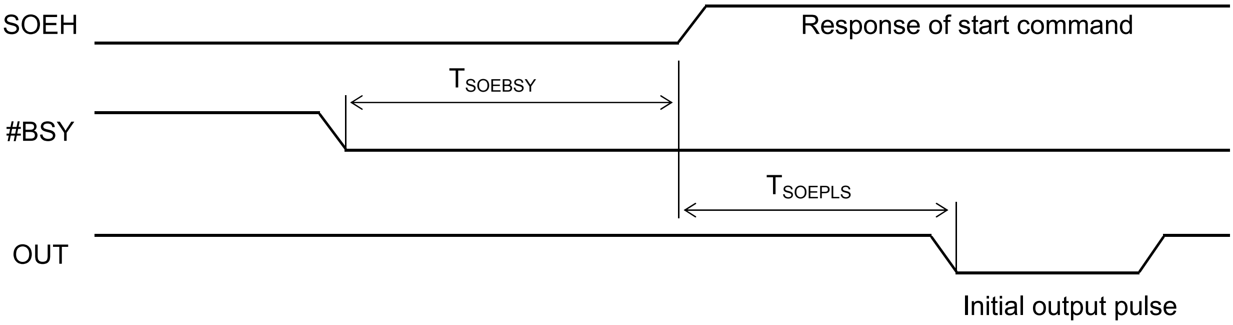
3) When the PA, PB inputs are in the 2-pulse mode



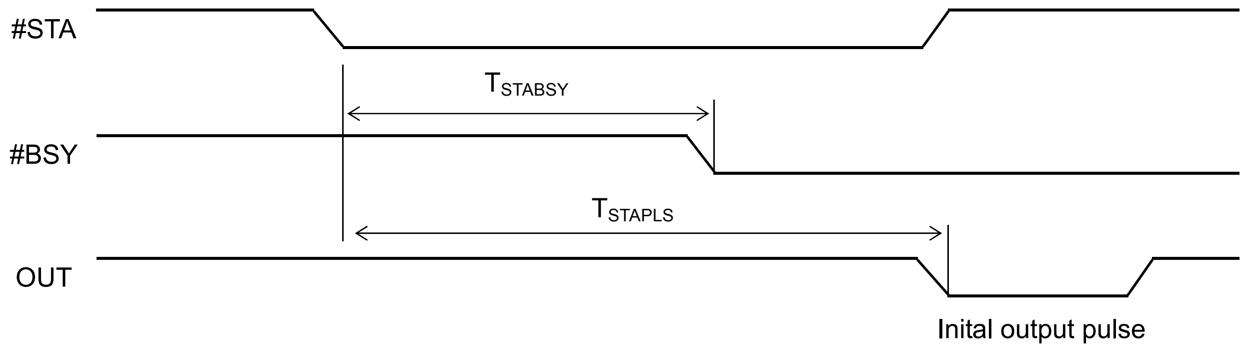
4) When the PA, PB inputs are in the 90° phase-difference mode



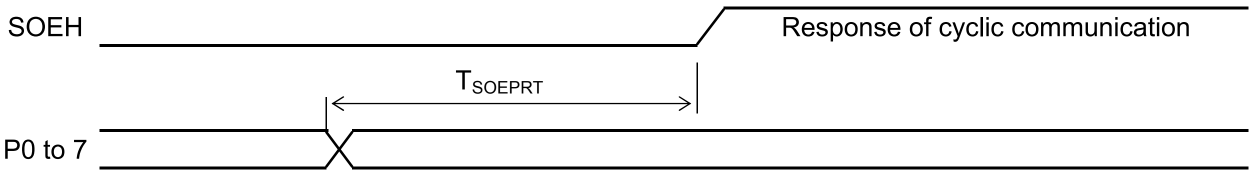
5) Start timing by commands



6) Simultaneous start timing



7) Output port change timing

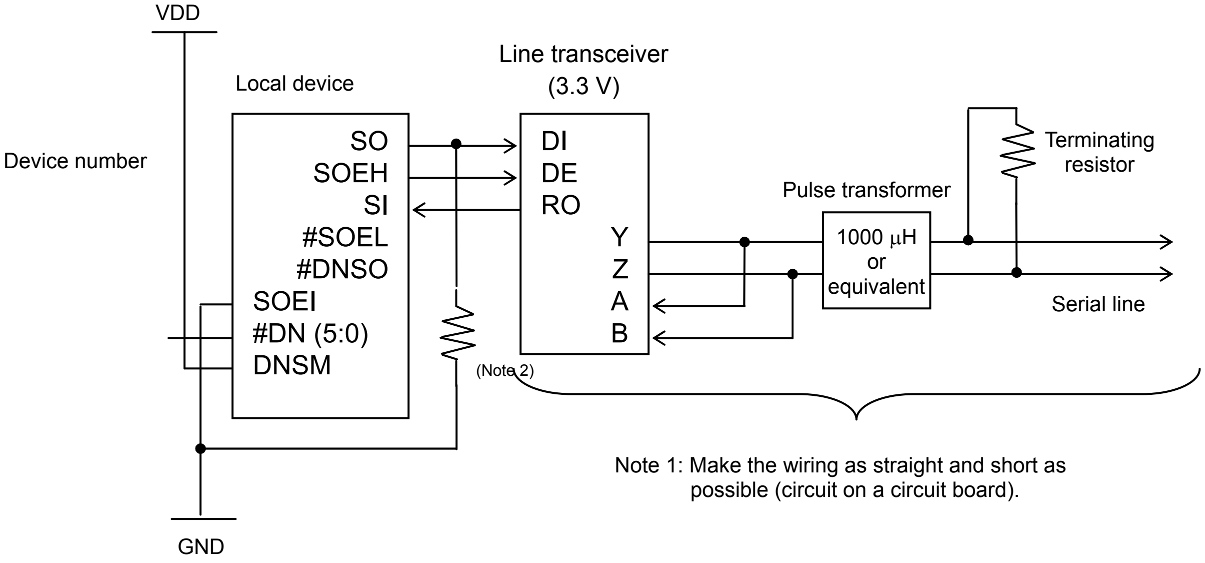


11. Communication example

11-1. PCL device (G9003), line transceiver, and pulse transformer

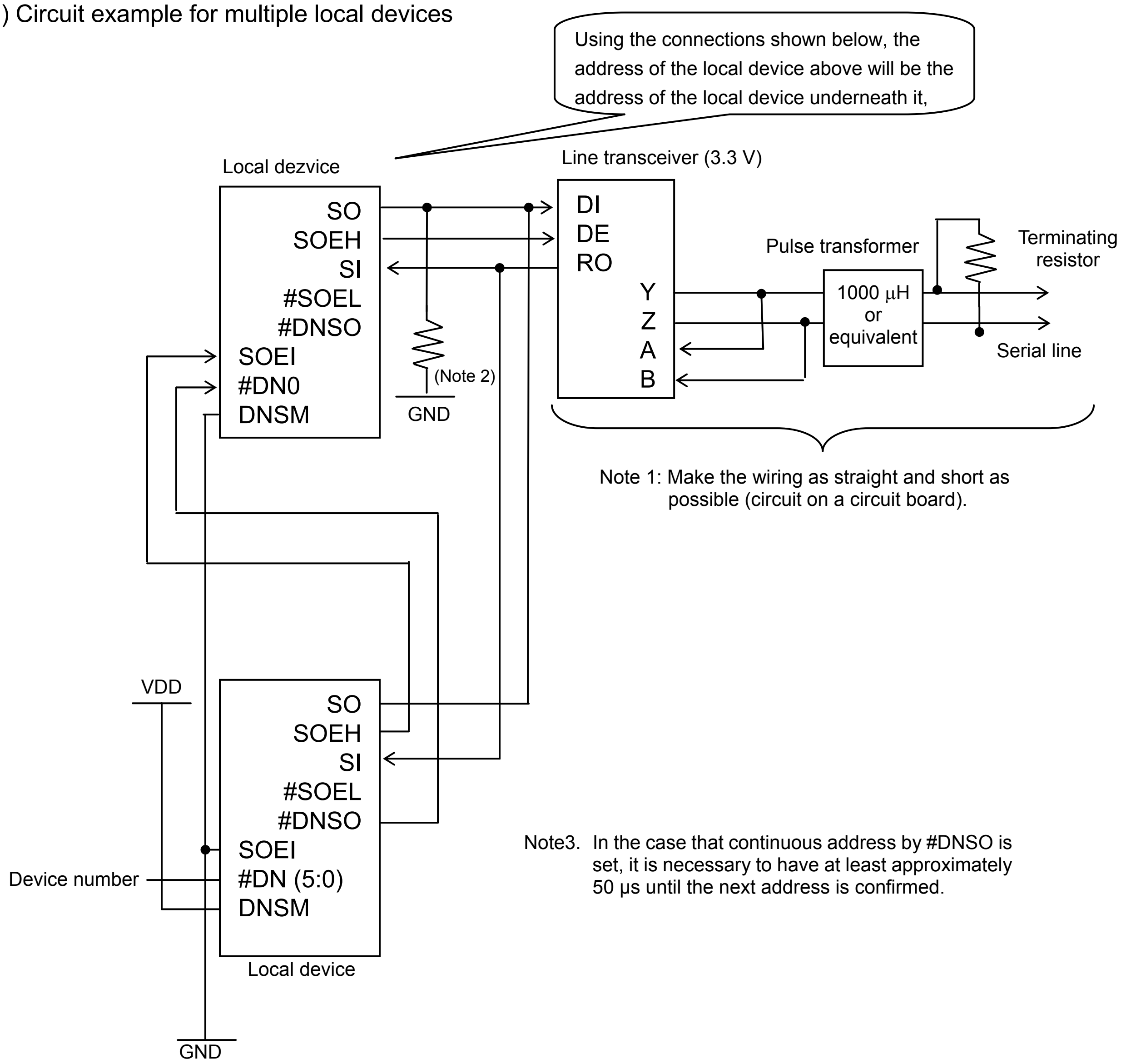
Use RS-485 line transceivers and pulse transformers (1000 μH or equivalent) to make serial communication connections.
Connect the line transceivers as shown below.
Connect terminating resistors (which match the cable impedance) at both ends of the transmission line.
The terminating resistors can be either before or after the pulse transformer. The same effect will be obtained at either position.
When using a 5 V line driver/receiver, ICs such as a level shifter are needed to assert signals on lines such as "SO," "SOEH," and "SI."

(1) Circuit example for a single local device



Note 1: Make the wiring as straight and short as possible (circuit on a circuit board).

(2) Circuit example for multiple local devices

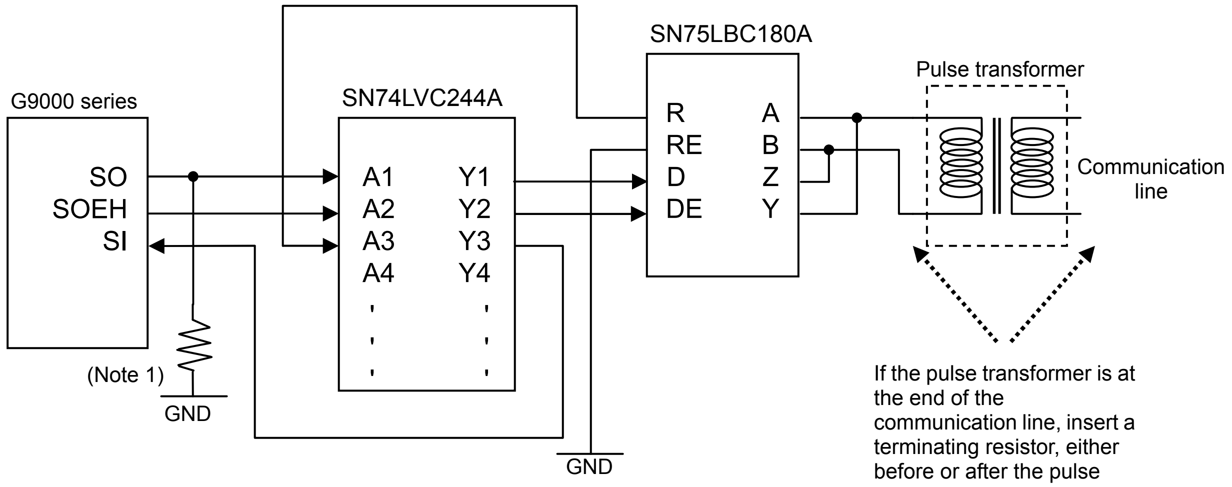

Using the connections shown below, the address of the local device above will be the address of the local device underneath it,
Local dezvice
SO
SOEH
SI
#SOEL
#DNSO
SOEI
#DN0
DNSM
(Note 2)
GND
Line transceiver (3.3 V)
DI
DE
RO
Y
Z
A
B
Pulse transformer
1000 μH or equivalent
Terminating resistor
Serial line
Note 1: Make the wiring as straight and short as possible (circuit on a circuit board).
VDD
SO
SOEH
SI
#SOEL
#DNSO
SOEI
#DN (5:0)
DNSM
Device number
Local device
GND
Note3. In the case that continuous address by #DNSO is set, it is necessary to have at least approximately 50 μs until the next address is confirmed.

Note 1: When connecting the serial lines to line transceivers, make the path as short and straight as possible.
Running these lines on a PC board could deteriorate the communication performance.

Note 2: Pull down resistors to GND should be 5 to 10 k-ohms.

11-2. A connection example of a level shifter

When using a 5 V line transceiver, a level shifter is needed.
Shown below is an example of the connections for a level shifter (TI: SN74LVC244A) and a line transceiver (TI: SN75LBC180A).



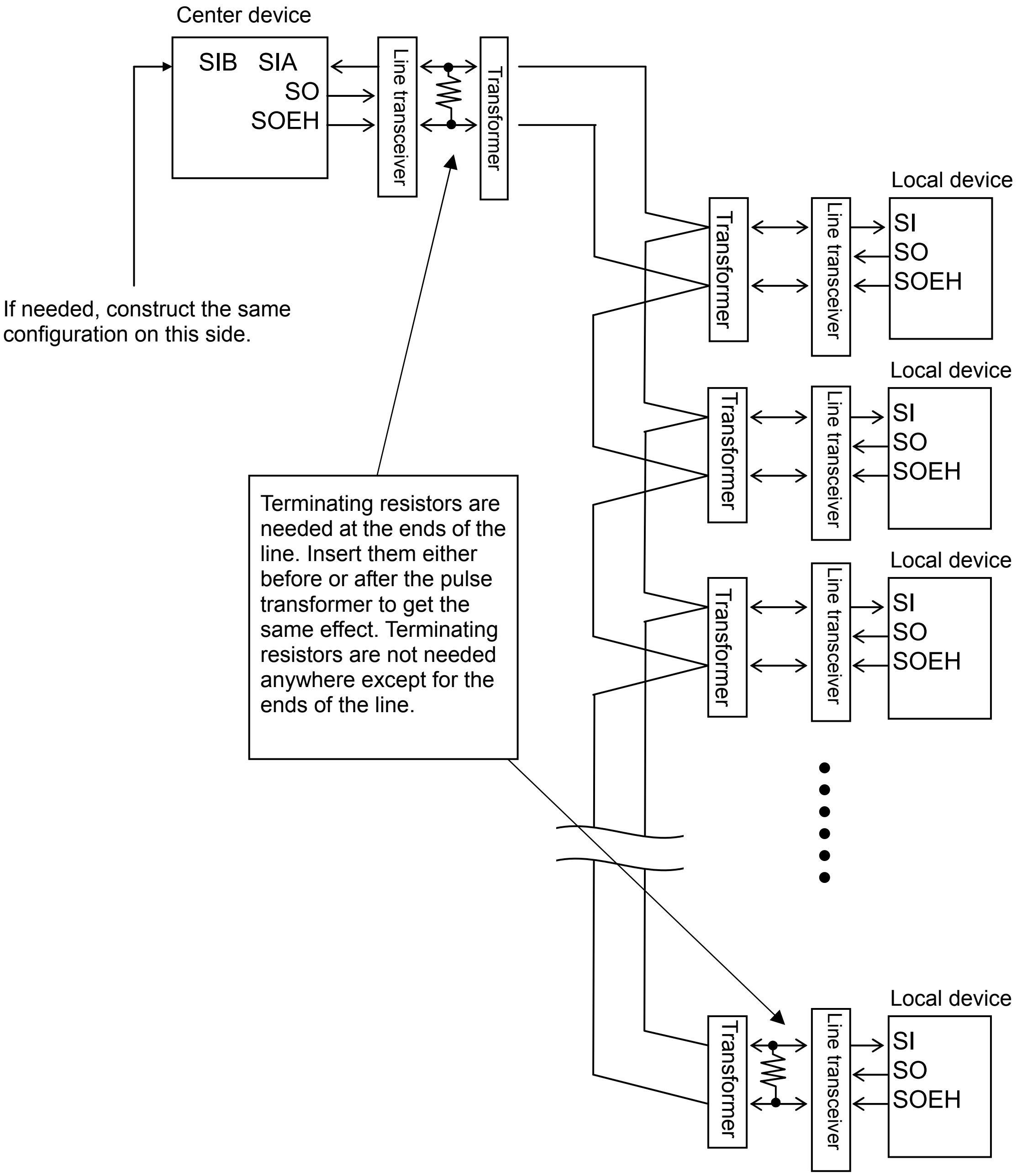
If the pulse transformer is at the end of the communication line, insert a terminating resistor, either before or after the pulse transformer.

Note 1: The pull down resistor to GND should be 5 to 10 k-ohms.

11-3. Complete configuration

We recommend a configuration with the center device at one end of the line and the local devices at the other end, as shown below.

If you want to place the center device in the middle of the line, use two communication lines so that the center device is effectively at the end of each line.



If needed, construct the same configuration on this side.

Terminating resistors are needed at the ends of the line. Insert them either before or after the pulse transformer to get the same effect. Terminating resistors are not needed anywhere except for the ends of the line.

12. Recommended environment

Shown below are the results of our experimental communication results and the environment used for the experiment.
These results can be used to design your own system. However, other system configurations are possible.
The example below is only for your reference.

Conditions						Results
Transmission rate	Number of local devices	Cable used	Terminating resistor	Pulse transformer	I/F chip	Max. length
20 Mbps	32	CAT5	100 ohm	1000 µH	RS485	100 m
20 Mbps	64	CAT5	100 ohm	1000 µH	RS485	50 m
10 Mbps	64	CAT6	100 ohm	1000 µH	RS485	100 m

Note: In the figures above, the maximum length figures are results from ideal conditions in a laboratory.
In actual use, the results may not be the same.

12-1. Cable

Commercially available LAN cables were used.
CAT5: Category 5
CAT6: Category 6
We used these LAN cables because they are high quality, inexpensive, and easy to obtain. Lower quality cables (such as cheap instrument cables) may significantly reduce the effective total length of the line.
LAN cables normally consist of several pairs of wires. Make sure to use wires from the same pair for one set of communication lines.
Even when using cables with the same category and rating, the performance of each cable manufacturer may be different. Always use the highest quality cables in the same category.

12-2. Terminating resistor

Select resistors that match the impedance of the cable used.
Normally, a 100 ohm resistor is recommended. Therefore, we used terminating resistors with this value.
Adjusting this resistor value may improve the transmission line quality.

12-3. Pulse transformer

We recommend using pulse transformers, in order to isolate the GND of each local device.
By isolating the GNDs, the system will have greater resistance to electrical noise. If pulse transformers are not used, the transmission distance may be less.
We used 1000 µH transformers in our experiments.

12-4. I/F chip

We selected I/C chips with specifications better than the RS485 standard.
In the experiment, we used 5 V line transceivers. When 5 V line transceivers are used, level shifters are needed to make the connections.

12-5. Parts used in our experiments

Show below is a list of the parts used in the interface circuits of our experiments.
Use of other parts may change the system's response. This list is only for your reference.

Parts	Manufacturer	Model name
CAT5	Oki Wire Co., Ltd.	F-DTI-C5 (SLA)
CAT6	Oki Wire Co., Ltd.	DTI-C6X
Pulse transformer	Nippon Pulse Motor, Co., Ltd.	NPT102F
Line transceiver	TEXAS INSTRUMENTS	SN75LBC180AP
Level shifter	TEXAS INSTRUMENTS	SN74LVC244ADB

12-6. Other precautions

- Cables

When you are planning long distance transmission, cable quality will be the single most important factor. Specialized cables designed for use as field busses, such as those by CC-Link and LONWORKS, have guaranteed quality and may be easier to use.

- Pulse transformers

Needless to say, the pulse transformers should handle 20 Mbps (10 MHz) without becoming saturated. The transformer's inductance is also important.
Since up to 64 pulse transformers may be connected, the actual working specifications of these devices must be very similar.
We used 1000 μH pulse transformers. However, in order to obtain better response characteristics, you may want to try pulse transformers with a larger reactance.

- Line transceivers

We used TEXAS Instruments chips for the experiments.
Other possibilities are available from MAXIM and LINEAR TECHNOLOGY, who offer very high performance transceivers.

- Connectors

If possible, the connectors should match the cable characteristics.
Although we did not use them, modular type connecters will be better for LAN cables.

- Cable connections

Do not connect one cable to another cable (using connectors etc.).
In a multi-drop system, the number of cables increases as the number of local devices increase.
However, connecting a cable just to extend the line should be avoided.

- Processing of excess cable

Excess cable, left over after making all the runs, should be eliminated.
Unneeded cable length may restrict the line overall usable length, and may introduce electrical noise.

- Circuit board substrate

Create circuits on a substrate with 4 or more layers, to prevent the introduction of noise.

13. Software example

This Chapter outlines software for the center device (G9001A) using flow charts.
In the flow charts, required variables are used for convenience.

13-1. Environment and precautions used for the descriptions

The descriptions below assume that I/F mode 3 is selected. Therefore, a 16-bit data bus is used. Also, these descriptions are based on the assumption that the wiring connections around the center device have been properly prepared and that the connected local devices are turned on. And, of course, we presume that connections to the serial line and the termination resistances are all correct.

13-2. Commands used

We will use the following two commands to access the address map in the center device.

1) Write command to the center device

Outpw (Address, Data)	
Address	Value corresponding to the address map in the center device The lowest bit is fixed to 0.
Data	Data to write (16 bits)
Return value	None

2) Read command from the center device

Inpw (Address)	
Address	Value corresponding to the address map in the center device The lowest bit is fixed to 0.
Return value	Read data (16 bits)

13-3. Center device address map

For details, see the user's manual for the center device.
Interface mode I/F mode 3

A1 to A8		Writing	Reading
0 0000 000	000h	Command bits 0 to 15	Status bits 0 to 15
0 0000 001	002h	Invalid	Interrupt status bits 0 to 15
0 0000 010	004h	Input buffer bits 0 to 15	Output buffer bits 0 to 15
0 0000 011	006h	Data transfer FIFO bits 0 to 15	Data receiving FIFO bits 8 to 15
0 0000 100 | 0 0111 011	008h | 076h	Not defined (56 words) (Any data written here will be ignored.)	Not defined (56 words) (Always read as 00h.)
0 0111 100	078h	Device information (Device No. 0, 1)	Device information (Device No.0, 1)
|	|	|	|
0 1011 011	0B6h	Device information (Device No. 62, 63)	Device information (Device No.62, 63)
0 1011 100	0B8h	Cyclic communication error flags (Device No. 0 to 15)	Cyclic communication error flags (Device No. 0 to 15)
|	|	|	|
0 1011 111	0BEh	Cyclic communication error flags (Device No. 48 to 63)	Cyclic communication error flags (Device No. 48 to 63)
0 1100 000	0C0h	Input change interrupt settings (Device No. 0 to 3)	Input change interrupt settings (Device No. 0 to 3)
|	|	|	|
0 1101 111	0DEh	Input change interrupt settings (Device No. 60 to 63)	Input change interrupt settings (Device No. 60 to 63)
0 1110 000	0E0h	Input change interrupt flags (Device No. 0 to 3)	Input change interrupt flags (Device No. 0 to 3)
|	|	|	|
0 1111 111	0FEh	Input change interrupt flags (Device No. 60 to 63)	Input change interrupt flags (Device No. 60 to 63)
1 0000 000	100h	Port data No. 0 (Device No.0 - Port 0, 1)	Port data No. 0 (Device No.0 - Port 0, 1)
1 0000 001	102h	Port data No. 2 (Device No.0 - Port 2, 3)	Port data No. 2 (Device No.0 - Port 2, 3)
|	|	|	|
1 1111 110	1FCh	Port data No. 252 (Device No.63 - Port 0, 1)	Port data No. 252 (Device No.63 - Port 0, 1)
1 1111 111	1FEh	Port No.255 (Device No.63 - Port 2, 3)	Port data No. 255 (Device No.63 - Port 2, 3)

Note: The hexadecimal notation for the addresses above are written with the assumption that A0 = 0.

13-4. Center device status

For details, see the user's manual for the center device.

STSW															
STSB1								STSB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DBSY	RBSY	SBSY	0	RDBB	TDBB	REF	0	CAER	ERAE	EDTE	EIOE	IOPC	BRKF	CEND

Bit	Symbol	Description
0	CEND	Becomes 1 when ready for data to be written to the transmitting FIFO buffer. When the system communication or data communication is complete and the next chunk of data can be sent to the transmitting FIFO buffer, this bit becomes 1 and the center device outputs an interrupt signal (INT). Once the status of this bit is read it returns to 0.
1	BRKF	When the center device receives a break frame this bit becomes 1 and an interrupt signal (INT) is output. Once the status of this bit is read it returns to 0.
2	IOPC	Becomes 1 when any input port which had enabled the "input change interrupt setting" and that status changed. The center device then outputs an interrupt signal (INT). This signal is an OR of all 256 "input port change interrupt flag" bits. When all the bits return to 0, this bit returns to 0.
3	EIOE	Becomes 1 when an Cyclic communication error occurs. The center device then outputs an interrupt signal (INT). This signal is an OR of all 64 "Cyclic communication error flag" bits. When all the bits return to 0, this bit returns to 0.
4	EDTE	Becomes 1 when a data or system communication error occurs. The center device then outputs an interrupt signal (INT). Once the status of this bit is read it returns to 0. Note 1
5	ERAE	Becomes 1 when a "local device reception processing error" occurs. The center device then outputs an interrupt signal (INT). Once the status of this bit is read it returns to 0. Then, the device number and details where the reception processing error occurred can be checked by reading the interrupt status.
6	CAER	A CPU access error occurred. When there is a problem accessing a CPU, such as a data send command being written when there is no data to send, this bit becomes 1. The center device then outputs an interrupt signal (INT). Once the status of this bit is read it returns to 0. The details of the error can be checked by reading the interrupt status.
7	(Not defined)	Always 0.
8	REF	When there is not-yet-sent output port data, this bit becomes 1. Write a 1 to the output port area. When cyclic communication to all the ports has completed, this bit returns to 0.
9	TDBB	When there is data to send in the transmitting FIFO, this bit becomes 1. After data is written to the transmitting FIFO, this bit becomes 1. Once a data send command or a transmitting FIFO reset command is written, this bit returns to 0.
10	RDBB	When data has been received in the receiving FIFO, this bit becomes 1. When receiving data from a data device, this bit becomes 1. After a CPU has read all of the data received, this bit returns to 0.
11	(Not defined)	Always 0.
12	SBSY	Becomes 1 when cyclic communication starts.
13	RBSY	Is 1 during a reset.
14	DBSY	Is 1 during system communication or data communication.
15	(Not defined)	Always 0.

Note 1: The details of an error that occurred due to an attempt to communicate a type of information to an I/O device that is different from that called for in the PMD0 to 2 settings, can be checked by reading the interrupt status. (When errors occur on more than one device, only the device number where the last error occurred would be shown.)

13-5. Interrupt status

For details, see the user's manual for the center device.

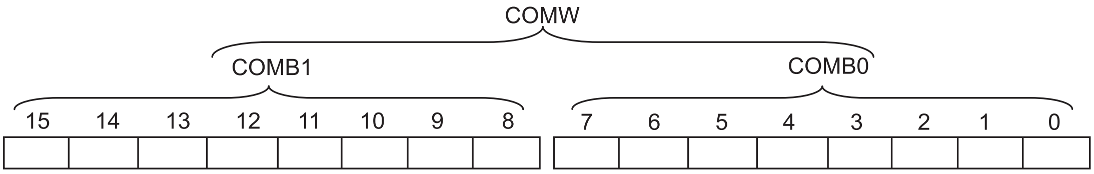
ISTW															
ISTB1								ISTB0							
CAE3	CAE2	CAE1	CAE0	ERA3	ERA2	ERA1	ERA0	LNRV	0	EDN5	EDN4	EDN3	EDN2	EDN1	EDN0

Bit	Symbol	Description
0 to 5	EDN0 to 5	Contains the device number of the device with an EDTE = 1 or ERAE = 1 (error from receiving I/O data that is different from the setting of PMD). These details are stored until the next time an error occurs.
6	(Not defined)	Always 0.
7	LNRV	When a local device is not receiving data, this bit is 1. When the data communication or system communication terminates with an error (EDTE = 1) (only when receiving attribute information), and if a local device cannot receive data from the center device, this bit becomes 1. When the local device has received the data, this bit returns to 0. This condition is stored until the next time an error occurs.
8 to 11	ERA0 to 3	These are identification codes for received data processing errors on a local device. The code is stored until the next time an error occurs. 0001: Received I/O data is different from the PMD settings. 0010: An I/O device received a data communication frame. (*) 0011: A data device received frames larger than the receiving buffer capacity. (*)
12 to 15	CAE0 to 3	These are access error codes from a CPU. The code is stored until the next time an error occurs. 0001: The device number was zero and a cyclic communication start command was written. 0010: Tried to write data with a start sending command without any data to send. 0011: While the DBSY = 1, a device tried to do one of the following: (1) Reading or writing to the transmitting or receiving FIFO. (2) Wrote a system start command or a data communication start command. 0100: Tried to send data to a device that is not is in use.

* When the ERA code is "0010" or "0011," the device number is not available in the EDN.

13-6. Center device command

For details, see the user's manual for the center device.



Note: Write to the 8-bit CPU I/F (IF0=H, IF1=1) in the following order: COMB0 then COMB1.

Command	Description
0000 0000 0000 0000 (0000h)	NOP Invalid command.
0000 0001 0000 0000 (0100h)	Resets the software. Resets the center device. This is the same function as the #RST input.
0000 0010 0000 0000 (0200h)	Resets the transmitting FIFO. Resets only the data transmitting FIFO.
0000 0011 0000 0000 (0300h)	Resets the receiving FIFO. Resets only the data receiving FIFO.
0001 0000 0000 0000 (1000h)	System communication to all devices. Polls all of the devices (device Nos. 0 to 63) one by one, and refreshes the "device information" areas that correspond to each device number. The "device information" contains the following: - Device in use: 0 when no response, and 1 when it responds. - Device type: Reset to 1 when it is a data device. - I/O setting information
0001 0001 0000 0000 (1100h)	System communication to all devices except those devices excluded from cyclic communication. After checking the "device information" area, the center device polls all the devices whose device-in-use bit is set to 0, one by one, and refreshes the "device information" areas that correspond to each device number. The details are refreshed the same as by writing a command 1000h.
0001 0010 00## #### (1200h to 123Fh)	System communication to specified devices. The center device polls only the specified devices and refreshes the "device information" areas that correspond to each specified device number. The details are refreshed the same as by writing a command 1000h.

Note: For all bits marked with a "#," the upper bits of the device address should be set in order, starting from the left end of the # bits.

For bits with marked with an "&," when the port is 0 or 1, set the bit to 0. When the port is 2 or 3, set the bit to 1.

For bits marked with an "x," either 0 or 1 may be used.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>0001 0011 00## ####
(1300h to 133Fh)</td><td>Obtain attribute information for the specified devices.
The polling response frame consists of device attribute information.
This command polls the specified devices and copies the attribute information into the data receiving FIFO.
The "device information" area does not change.
The details of the data receiving FIFO are as follows.
Bits 0 to 4: (Number of bytes for the longest piece of data) / 8 -1
Bits 5 to 7: Not used (not defined)
Bits 8 to 15: Device type code (I/O device: 01h, Data device: 81h)
Bits 16 to 18: Set the I/O port (PMD terminal information when an I/O device is selected)
Bit 19: Always 0
Bits 20 to 31: Data device type (G9003: 000h, G9004A: 001h)</td></tr>
<tr><td>0011 0000 0000 0000
(3000h)</td><td>Start Cyclic communication
Start Cyclic communication with devices that have a 1 in the "device-in-use" bit in the "device information".</td></tr>
<tr><td>0011 0001 0000 0000
(3100h)</td><td>Stop Cyclic communication.
Stop the current Cyclic communication.</td></tr>
<tr><td>0100 0000 00## ####
(4000h to 403Fh)</td><td>Data communication.
Sends data in the transmitting FIFO to the specified devices. The data received in response will be stored in the receiving FIFO.</td></tr>
<tr><td>0100 0001 0000 0000
(4100h)</td><td>Cancel data communication
Halt the data communication and reset the transmitting FIFO.
This command will be ignored after the data has been sent.</td></tr>
<tr><td>0101 0000 0### ##xx
(5000h to 507Fh)</td><td>Write to the "Device information" area.
The contents of the I/O buffer are written into a word in the device information area.
As an example, the relationship between the I/O buffer details and the device information area are listed below.
<table>
<tr><th>Command</th><th>I/O buffer</th><th>Address</th><th>Device No.</th></tr>
<tr><td>5000h</td><td>Bit 0 to 7</td><td>078h</td><td>0</td></tr>
<tr><td></td><td>Bit 8 to 15</td><td>079h</td><td>1</td></tr>
<tr><td>5004h</td><td>Bit 0 to 7</td><td>07Ah</td><td>2</td></tr>
<tr><td></td><td>Bit 8 to 15</td><td>07Bh</td><td>3</td></tr>
<tr><td>5008h</td><td>Bit 0 to 7</td><td>07Ch</td><td>4</td></tr>
<tr><td></td><td>Bit 8 to 15</td><td>07Dh</td><td>5</td></tr>
<tr><td>500Ch</td><td>Bit 0 to 7</td><td>07Eh</td><td>6</td></tr>
<tr><td></td><td>Bit 8 to 15</td><td>07Fh</td><td>7</td></tr>
</table>
Use this function when you want to reduce the number of addresses used in the center device.</td></tr>
</table>

Note: For all bits marked with a "#," the upper bits of the device address should be set in order, starting from the left end of the # bits.

For bits with marked with an "&," when the port is 0 or 1, set the bit to 0. When the port is 2 or 3, set the bit to 1.

For bits marked with an "x," either 0 or 1 may be used.

G9002: I/O device

G9003: PCL device

Command	Description
0101 0001 0##x xxxx (5100h to 517Fh)	Write to the "Cyclic communication error flag" area. The contents of the I/O buffer are written into a word in this area. Use this function when you want to reduce the number of addresses used in this device.
0101 0010 0### #xxx (5200h to 527Fh)	Write to the "input change interrupt setting" area. The contents of the I/O buffer are written into a word in this area. Use this function when you want to reduce the number of addresses used in this device.
0101 0011 0### #xxx (5300h to 537Fh)	Write to the "input change interrupt flag" area. The contents of the I/O buffer are written into a word in this area. Use this function when you want to reduce the number of addresses used in this device.
0101 0100 0### ###& (5400h to 547Fh)	Write to the "port data" area. The contents of the I/O buffer are written into a word in this area. Use this function when you want to reduce the number of addresses used in this device.
0110 0000 0### ##xx (6000h to 607Fh)	Read the "device information" area. The contents of the word in this area are copied to the I/O buffer. Use this function when you want to reduce the number of addresses used in this device.
0110 0001 0##x xxxx (6100h to 617Fh)	Read the "Cyclic communication error flag" area. The contents of the word in this area are copied to the I/O buffer. Use this function when you want to reduce the number of addresses used in this device.
0110 0010 0### #xxx (6200h to 627Fh)	Read the "input change interrupt setting" area. The contents of the word in this area are copied to the I/O buffer. Use this function when you want to reduce the number of addresses used in this device.
0110 0 11 0### #xxx (6300h to 637Fh)	Read the "input change interrupt flag" area. The contents of the word in this area are copied to the I/O buffer. Use this function when you want to reduce the number of addresses used in this device.
0110 0100 0### ###& (6400h to 647Fh)	Read the "port data" area. The contents of the word in this area are copied to the I/O buffer. Use this function when you want to reduce the number of addresses used in this device.

Note: For all bits marked with a "#," the upper bits of the device address should be set in order, starting from the left end of the # bits.

For bits with marked with an "&," when the port is 0 or 1, set the bit to 0. When the port is 2 or 3, set the bit to 1.

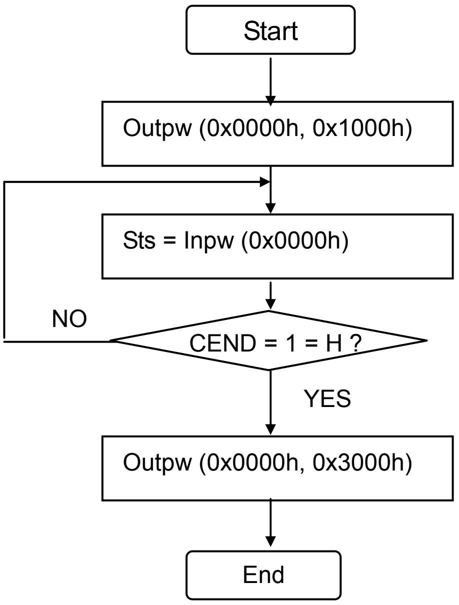
For bits marked with an "x," either 0 or 1 may be used.

If all of the address map byte (512 bytes) requested by the center device are allocated so that a CPU can see them, the commands from "5000h" and after (as shown above) are not needed.

If the resources controlled by a CPU are limited and only 8 bytes are available for addresses, the commands from "5000h" and up can be used to access to all of the addresses owned by the center device.

13-7. Start of the simplest cyclic communication

The simplest example is to issue a system communication command, let the center device automatically collect data from the local devices, and then start cyclic communication.



Send a system communication command to all local devices.

Read status

Waits for completion of the system communication
Without a completion signal the center device cannot start the next operation.

Start cyclic communication

13-8. Communication with port data (port data and data device status)

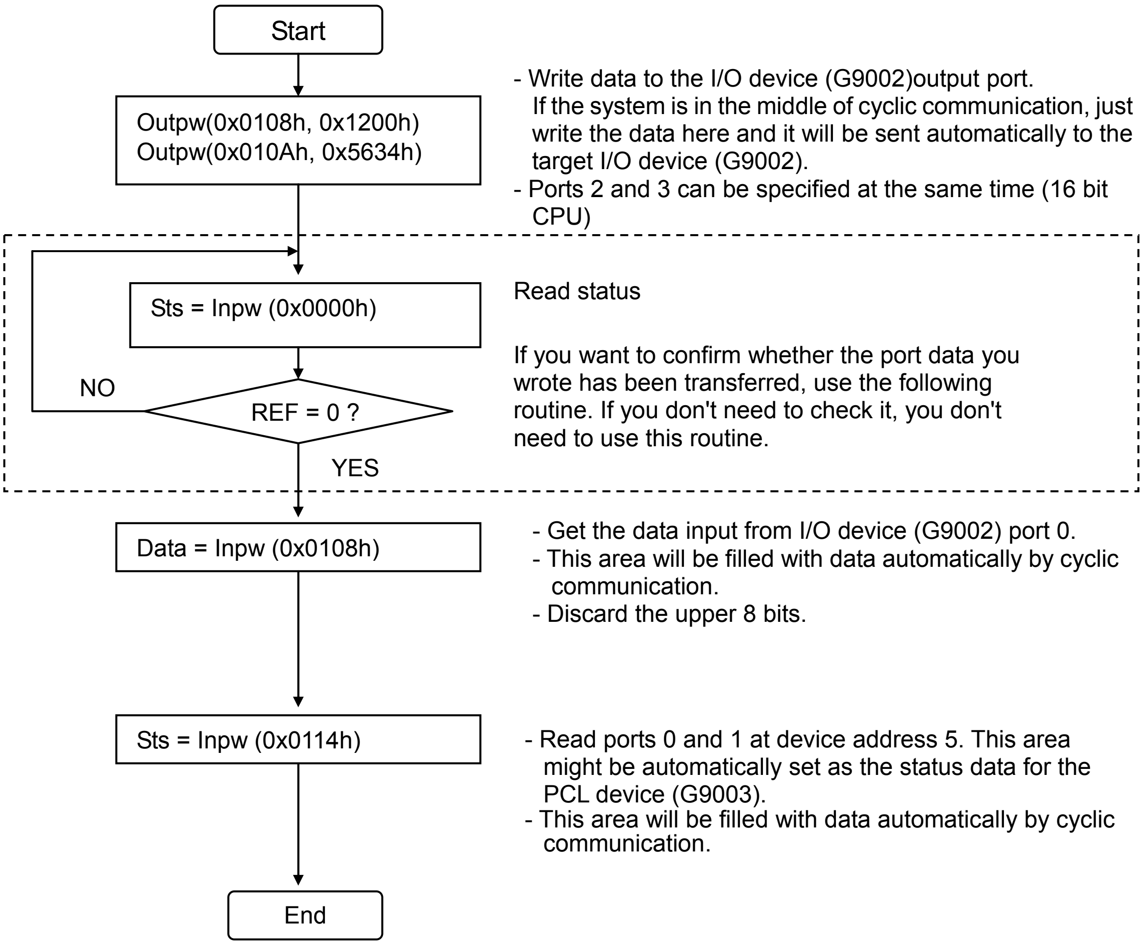
This section describes data exchange using the I/O port on an I/O device (G9002), and how to obtain the status of a data device.
Assume that the local devices to be used are as follows:
Only an example of how to read the status is given for the PCL device (G9003).

Device type	Item to configure	Configuration data	Output data
I/O device	Device address	2	-
	Port 0	Input	-
	Port 1	Output	12h
	Port 2	Output	34h
	Port 3	Output	56h
PCL device	Device address	5	

Note: The port area configuration of the PCL device (G9003) is always as follows (fixed).

Port No.	Mode	Description
Port 0	Input	Main status (MSTSB0) lower 16 bits
Port 1	Input	Main status (MSTSB1) upper 16 bits
Port 2	Input	Input value from the general-purpose I/O port (IOPIB)
Port 3	Output	Output value to the general-purpose I/O port (IOPIB)

1) When the whole address map can be used



- Write data to the I/O device (G9002)output port.
  If the system is in the middle of cyclic communication, just write the data here and it will be sent automatically to the target I/O device (G9002).
- Ports 2 and 3 can be specified at the same time (16 bit CPU)

Read status

If you want to confirm whether the port data you wrote has been transferred, use the following routine. If you don't need to check it, you don't need to use this routine.

- Get the data input from I/O device (G9002) port 0.
- This area will be filled with data automatically by cyclic communication.
- Discard the upper 8 bits.

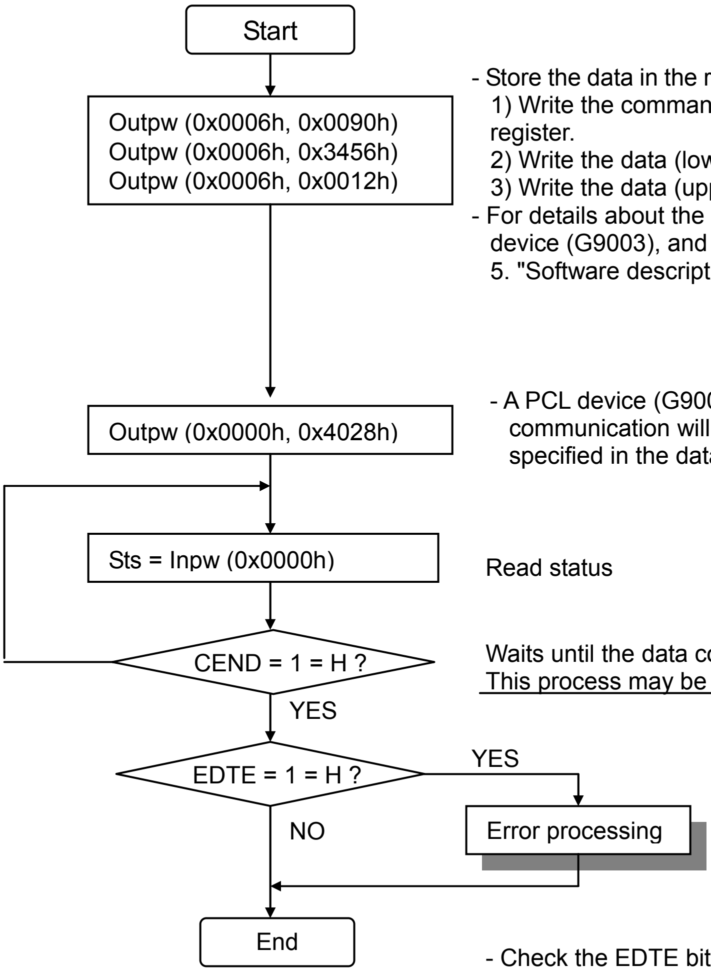
- Read ports 0 and 1 at device address 5. This area might be automatically set as the status data for the PCL device (G9003).
- This area will be filled with data automatically by cyclic communication.

13-9. Data communication 1: Put the value in the register of the PCL device (G9003)

The data communication example below shows data being placed in a register that is integrated in the PCL device (G9003).
Assume that the local devices to be used are as follows.
Assume that "00123456h" will be placed in the "RMV" register of the PCL device (G9003).

Device type	Configuration item	Device number
PCL device	Device address	40 (28h)

.



- Store the data in the receiving FIFO in the following order.
  1) Write the command to the PCL device (G9003) register.
  2) Write the data (lower 16 bits)
  3) Write the data (upper 16 bits)
- For details about the write order for the FIFO in the PCL device (G9003), and the access command, see Chapter 5. "Software description."

- A PCL device (G9003) that receives this communication will write the data to the register specified in the data details.

Read status

Waits until the data communication is complete.
This process may be waiting for an interrupt.

- Check the EDTE bit. If the data communication failed, take the defined steps.
- Note that the EDTE bit will be cleared by reading the status.
- The EDTE bit changes with the same timing as the CEND bit.

A data communication command is constructed as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	0	0	0	#	#	#	#	#	#

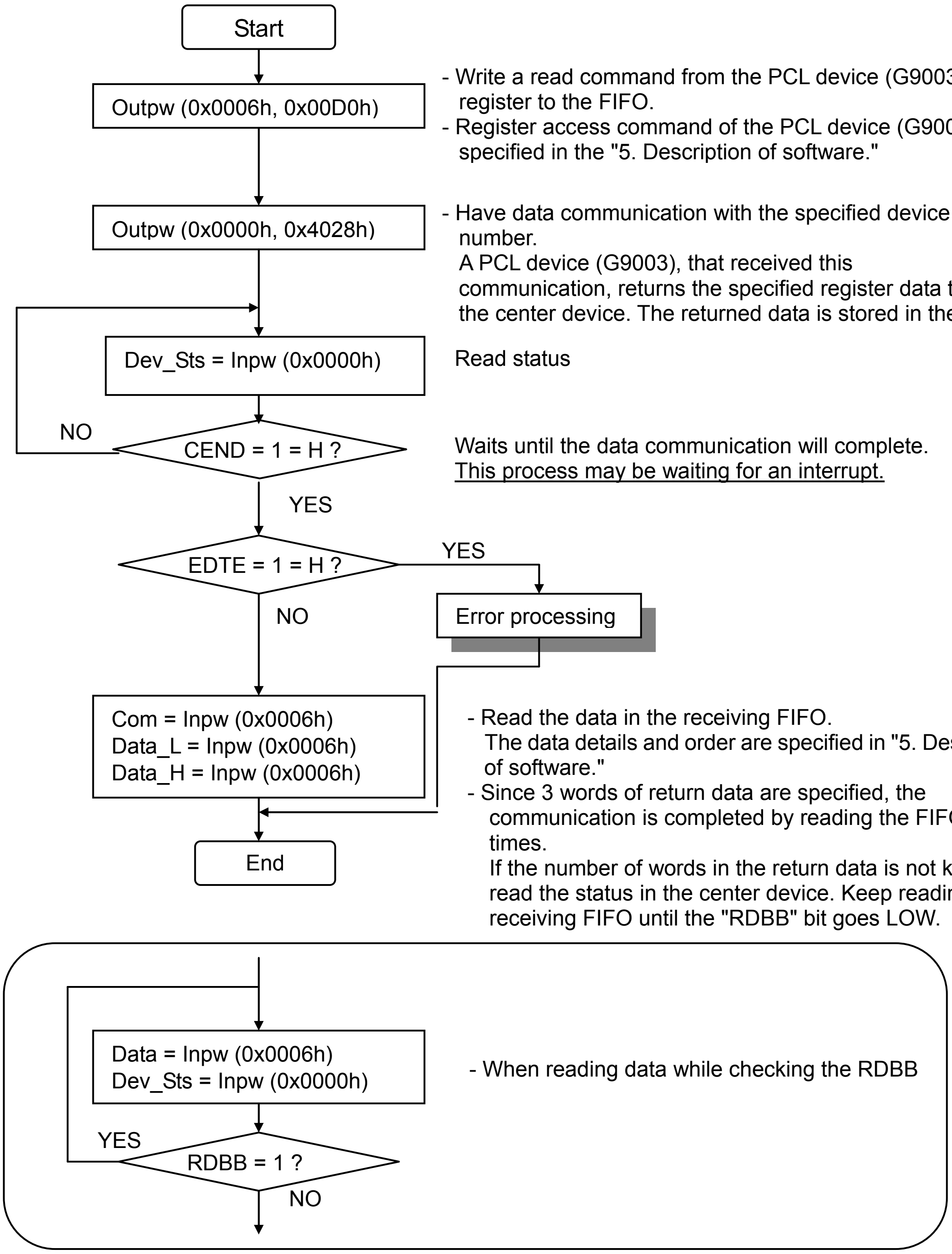
Specify the address in these

13-10. Data communication 2: Read a register in a PCL device (G9003)

The example of data communication below shows how to read a register that is integrated in the PCL device (G9003).
Assume that the local devices to be used are as follows.
Assume you want to read the register value in the PCL device (G9003).

Device type	Configuration item	Configuration data
PCL device	Device address	40 (28h)

.



- Write a read command from the PCL device (G9003) register to the FIFO.
- Register access command of the PCL device (G9003) is specified in the "5. Description of software."

- Have data communication with the specified device number.
A PCL device (G9003), that received this communication, returns the specified register data to the center device. The returned data is stored in the

Read status

Waits until the data communication will complete.
This process may be waiting for an interrupt.

- Read the data in the receiving FIFO.
The data details and order are specified in "5. Description of software."
- Since 3 words of return data are specified, the communication is completed by reading the FIFO three times.
If the number of words in the return data is not known, read the status in the center device. Keep reading the receiving FIFO until the "RDBB" bit goes LOW.

- When reading data while checking the RDBB

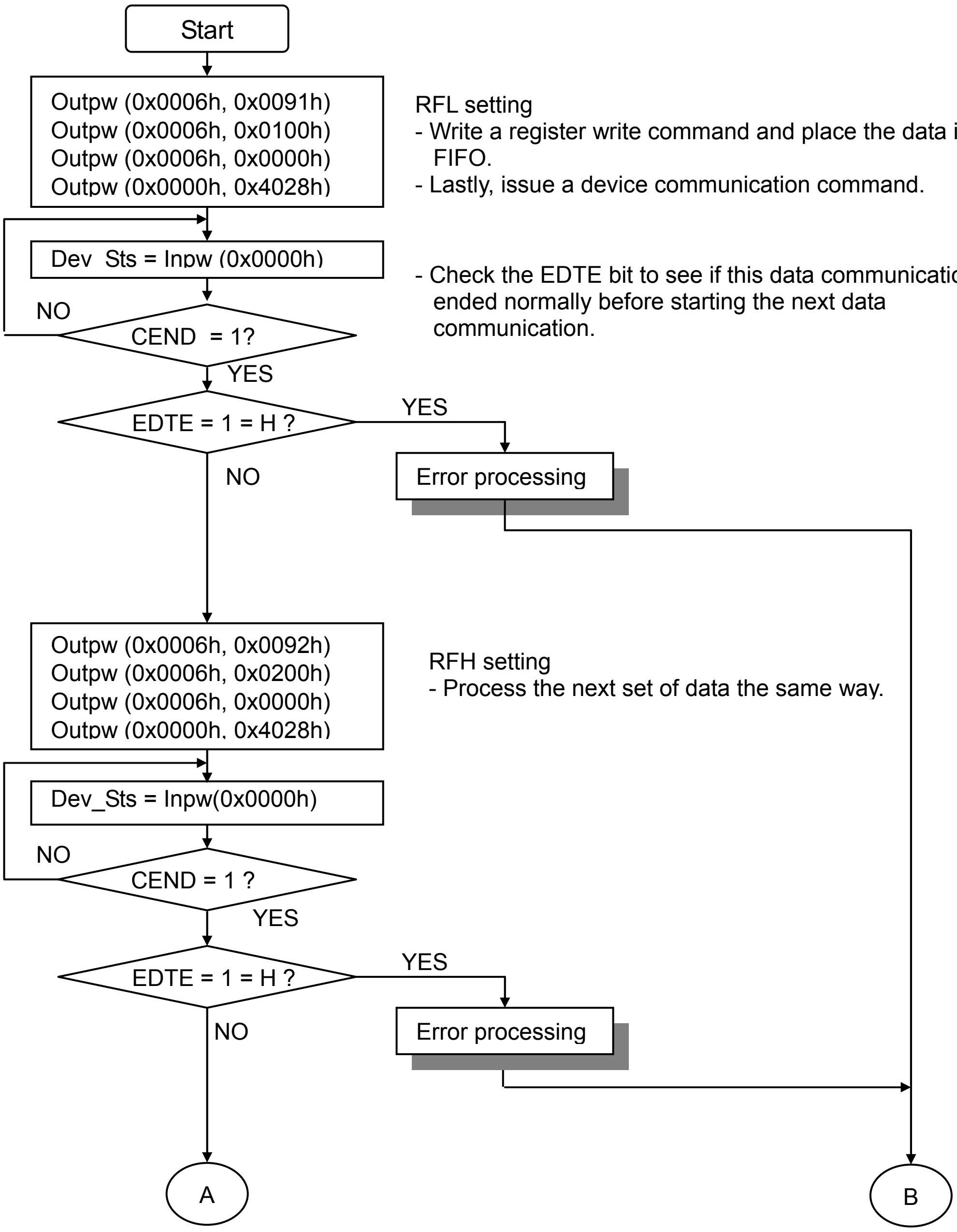
13-11. Data communication 3: Start the PCL device (G9003)

The data communication example below shows how to start pulse output by setting the registers in the PCL device (G9003).

The local devices are the same as in the previous section.

Assume that the data to place in the PCL device (G9003) are as follows (only the data needed to trigger the pulse output).

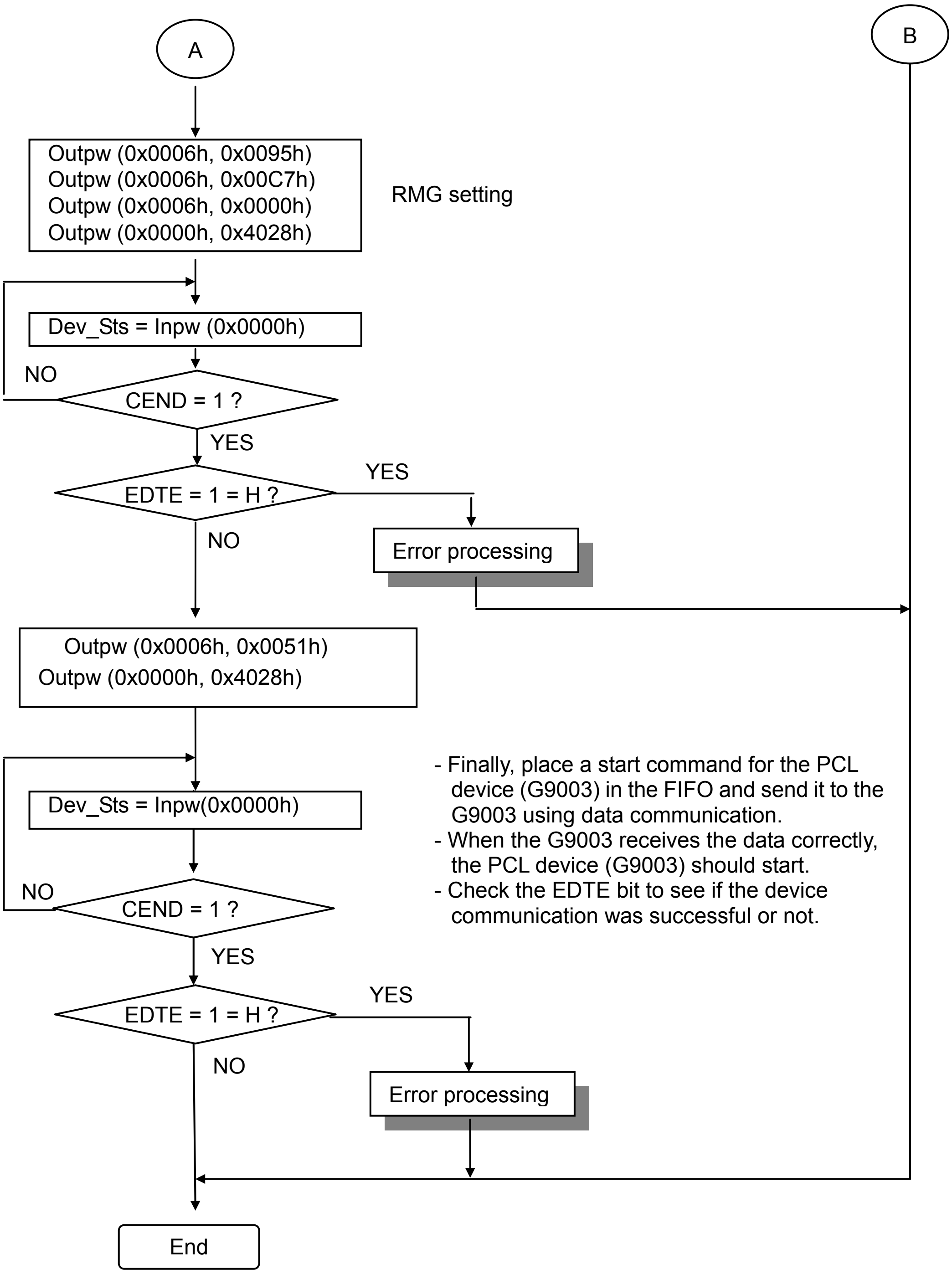
Register name	Set value	Remarks
RFL	00000100h	
RFH	00000200h	
RMG	00C7h	Multiplication rate = 1



RFL setting
- Write a register write command and place the data in the FIFO.
- Lastly, issue a device communication command.

- Check the EDTE bit to see if this data communication ended normally before starting the next data communication.

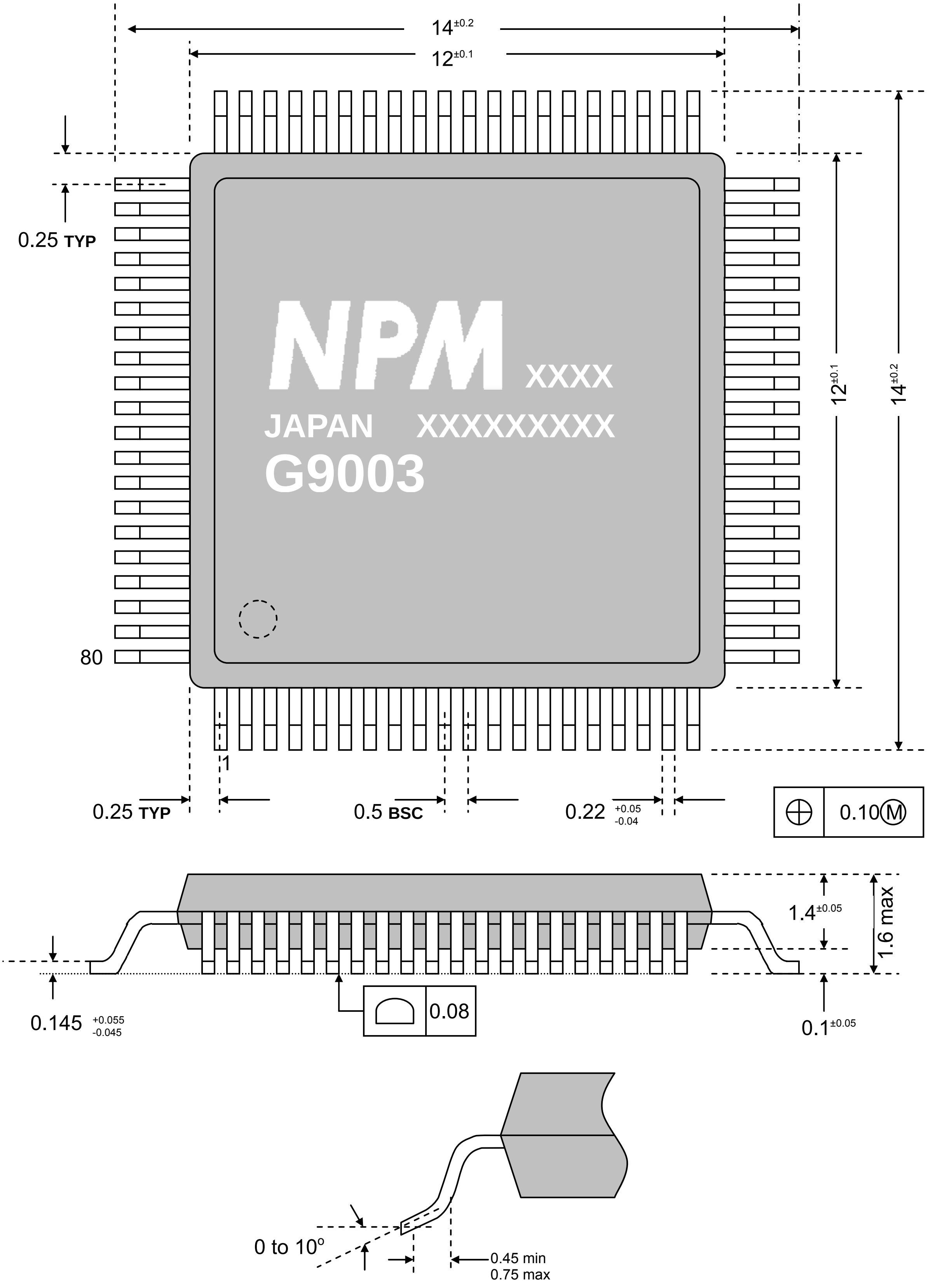
RFH setting
- Process the next set of data the same way.



- Finally, place a start command for the PCL device (G9003) in the FIFO and send it to the G9003 using data communication.
- When the G9003 receives the data correctly, the PCL device (G9003) should start.
- Check the EDTE bit to see if the device communication was successful or not.

14. External dimensions

80-pin, LQFP, Unit: mm



[Handling Precautions]

1. Design precautions

1) Never exceed the absolute maximum ratings, even for a very short time.

2) Take precautions against the influence of heat in the environment, and keep the temperature around the LSI as cool as possible.

3) Please note that ignoring the following may result in latching up and may cause overheating and smoke.

- Do not apply a voltage greater than +3.3V (greater than 5V for 5V connectable terminals) to the input/output terminals and do not pull them below GND.
- Make sure you consider the input timing when power is applied.
- Be careful not to introduce external noise into the LSI.
- Hold the unused input terminals to +3.3 V or GND level.
- Do not short-circuit the outputs.
- Protect the LSI from inductive pulses caused by electrical sources that generate large voltage surges, and take appropriate precautions against static electricity.

4) Provide external circuit protection components so that overvoltages caused by noise, voltage surges, or static electricity are not fed to the LSI.

2. Precautions for transporting and storing LSIs

1) Always handle LSIs carefully and keep them in their packages. Throwing or dropping LSIs may damage them.

2) Do not store LSIs in a location exposed to water droplets or direct sunlight.

3) Do not store the LSI in a location where corrosive gases are present, or in excessively dusty environments.

4) Store the LSIs in an anti-static storage container, and make sure that no physical load is placed on the LSIs.

3. Precautions for mounting

1) In order to prevent damage caused by static electricity, pay attention to the following.

- Make sure to ground all equipment, tools, and jigs that are present at the work site.
- Ground the work desk surface using a conductive mat or similar apparatus (with an appropriate resistance factor). However, do not allow work on a metal surface, which can cause a rapid change in the electrical charge on the LSI (if the charged LSI touches the surface directly) due to extremely low resistance.
- When picking up an LSI using a vacuum device, provide anti-static protection using a conductive rubber pick up tip. Anything which contacts the leads should have as high a resistance as possible.
- When using a pincer that may make contact with the LSI terminals, use an anti-static model. Do not use a metal pincer, if possible.
- Store unused LSIs in a PC board storage box that is protected against static electricity, and make sure there is adequate clearance between the LSIs. Never directly stack them on each other, as it may cause friction that can develop an electrical charge.

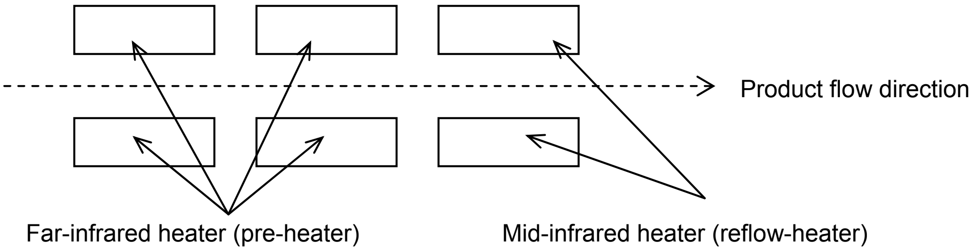
2) Operators must wear wrist straps which are grounded through approximately 1M-ohm of resistance.

3) Use low voltage soldering devices and make sure the tips are grounded.

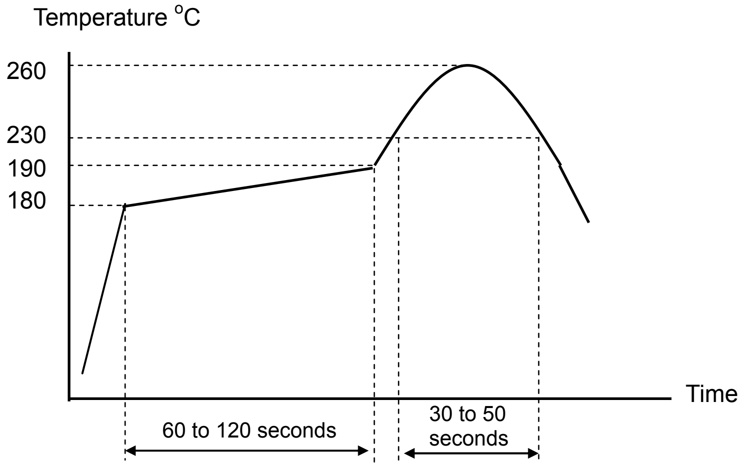
4) Do not store or use LSIs, or a container filled with LSIs, near high-voltage electrical fields, such those produced by a CRT.

5) To preheat LSIs for soldering, we recommend keeping them at a high temperature in a completely dry environment, i.e. 125°C for 24 hours. The LSI must not be exposed to heat more than 2 times.

6) When using an infrared reflow system to apply solder, we recommend the use of a far-infrared pre-heater and mid-infrared reflow devices, in order to ease the thermal stress on the LSIs.


Product flow direction
Far-infrared heater (pre-heater)
Mid-infrared heater (reflow-heater)

Package and substrate surface temperatures must never exceed 260°C and 230°C for 30 to 50 seconds.



[Recommended temperature profile of a far-infrared heater, hot air reflow]

7) When using hot air for solder reflow, the restrictions are the same as for infrared reflow equipment.

8) If you will use a soldering iron, the temperature at the leads must not be 260°C or less for more than 10 seconds, and must not be 350°C or less for more than 3 seconds.

4. Other precautions

1) When the LSI will be used in poor environments (high humidity, corrosive gases, or excessive amounts of dust), we recommend applying a moisture prevention coating.

2) The package resin is made of fire-retardant material; however, it can burn. When baked or burned, it may generate gases or fire. Do not use it near ignition sources or flammable objects.

3) This LSI is designed for use in commercial apparatus (office machines, communication equipment, measuring equipment, and household appliances). If you use it in any device that may require high quality and reliability, or where faults or malfunctions may directly affect human survival or injure humans, such as in nuclear power control devices, aviation devices or spacecraft, traffic signals, fire control, or various types of safety devices, we will not be liable for any problem that occurs, even if it was directly caused by the LSI. Customers must provide their own safety measures to ensure appropriate performance in all circumstances.

Notes