Motionnet

RemoteI/O & RemoteMotion

G9004A

(CPU emulation device)

User's Manual

NPM Nippon Pulse Motor Co., Ltd.

[Preface]

Thank you for considering our super high-speed serial communicator LSI, the "G9000" series.
To learn how to use the G9000 series device, read this manual and "G9001A/G9002A" user's 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 the 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. Please use the latest version. We provide the latest version on our web site.

(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.

[Logic indicators]

(1) Terminal names and signal names with a bar above the name use negative logic.
Ex.: $\overline{\text{RST}}$ means that the RST terminal uses negative logic.

Table of contents

1. Outline

This LSI is a CPU emulation device (G9004A). This LSI can be connected to our Motionnet and perform data communications with a center device (G9001A).
Either of the following two operation modes can be selected from a terminal.

1) CPU emulation mode
   In this mode, the G9004A emulates CPU terminal signals using data communicated from the center device (G9001A). Although the communication format from the center device is limited, this LSI outputs signals identical to those from CPU terminals. Therefore it can control various LSIs that are normally connected to a CPU.
2) Message communication mode
   One word (16 bits) is reserved by the system in this mode, which is used for communication (the data commands and formats are specified). The user can use the remaining 127 words to communicate data. The format for the message data is not specified, which means that this LSI can communicate freely with almost any CPU that is connected to the center device and to this LSI.

2. Features

- Compatible with our Motionnet.
- Can control various CPU support LSIs using data communications.
- A maximum of 256 bytes of data can be communicated in one sentence.
- A maximum of 64 devices can be connected using one line.
- It has a safety design using a communication failure detection circuit (contains a watchdog timer).
- Powered from just 3.3 VDC.

3. General specifications

3-1. Communication system specifications

Item	Description
Reference clock	40 MHz or 80 MHz
Communication speed	2.5 M, 5 M, 10 M, or 20 Mbps
Communication sign	NRZ coding
Communication protocol	NPM original method
Communication method	Half-duplex communication
Communication I/F	RS-485 or pulse transfer
Connection method	Multi-drop connection
Number of local devices	64 devices max.
Cyclic communication cycle when 20 Mbps	When using 8 local devices (IN: 128 points, OUT: 128 points) --- 0.12 msec. When using 16 local devices (IN: 256 points, OUT: 256 points) --- 0.24 msec. When using 32 local devices (IN: 512 points, OUT: 512 points) --- 0.49 msec. When using 64 local devices (IN: 1024 points, OUT: 1024 points) --- 0.97 msec. Note: The communication cycle will be extended dependent upon data communication.

3-2. Specifications of the CPU emulation device (G9004A)

Item		Description
CPU emulation mode		
	Communication sentence length	1 to 128 words/frame (1 word = 16 bits)
	Data buffer size	128 words
	Data communication time	When communicating 5 words (write to one register in the PCL): 21.7 µs
	Data transfer method	Status: Cyclic transfer, Data: Transient transfer
	Control address space	64 bytes
	CPU interface	8-bit I/F --- Z80, 6809 etc. 16-bit I/F --- 8086, H8, 68000 etc.
Message communication mode		
	Communication sentence length	1 to 128 words/frame (1 word = 16 bits)
	Data buffer size	128 words (1 word: reserved for the system, 127 words: Message data)
	Data communication time	When communicating 128 words: 169.3 µs
	Data transfer system	Status: Cyclic transfer Data communication: Transient transfer
	CPU interface	8-bit I/F --- Z80, 6809 etc. 16-bit I/F --- 8086, H8, 68000 etc.
Others		
	Package	80 pin QFP (mold size: 12 x 12 x 1.4 mm)
	Power supply	+3.0V to +3.6V
	Storage temperature range	-65 to +150˚C
	Operating temperature range	-40 to +85˚C

4. Hardware description

4-1. A list of terminals (QFP-80)

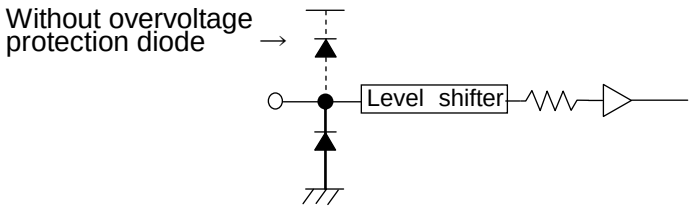
No.	Signal name	I/O	Logic	Description	Option
1	MOD	I	-	Select operation mode (Low: Message communication, High: CPU emulation)	U 5V
2	TUD	I	-	Set operation when outputting watchdog timer	U 5V
3	TMD	I	-	Set watchdog timer	U 5V
4	LCK0	I	-	Division rate of local bus control clock (LCLK): 0	U 5V
5	LCK1	I	-	Division rate of local bus control clock (LCLK): 1	U 5V
6	LWT0	I	Positive	Interval time of local bus: 0	U 5V
7	LWT1	I	Positive	Interval time of local bus: 1	U 5V
8	LIF0	I	-	Local bus I/F mode 0	U 5V
9	LIF1	I	-	Local bus I/F mode 1	U 5V
11	$\overline{\text{LCS}}$	B	Negative	Chip selection for local bus	5V S
12	$\overline{\text{LWR}}$ (R/W)	B	Negative	Write signal for local bus	5V S
13	$\overline{\text{LRD}}$ (E)(LS)	B	Negative	Read signal for local bus	5V S
14	LA0	B	Positive	Address for local bus: 0	5V S
16	LA1	B	Positive	Address for local bus: 1	5V S
17	LA2	B	Positive	Address for local bus: 2	5V S
18	LA3	B	Positive	Address for local bus: 3	5V S
19	LA4	B	Positive	Address for local bus: 4	5V S
20	LA5	B	Positive	Address for local bus: 5	5V S
22	LD0	B	Positive	Data for local bus: 0	5V S
23	LD1	B	Positive	Data for local bus: 1	5V S
24	LD2	B	Positive	Data for local bus: 2	5V S
25	LD3	B	Positive	Data for local bus: 3	5V S
27	LD4	B	Positive	Data for local bus: 4	5V S
28	LD5	B	Positive	Data for local bus: 5	5V S
29	LD6	B	Positive	Data for local bus: 6	5V S
30	LD7	B	Positive	Data for local bus: 7	5V S
32	LD8	B	Positive	Data for local bus: 8	5V S
33	LD9	B	Positive	Data for local bus: 9	5V S
34	LD10	B	Positive	Data for local bus: 10	5V S
35	LD11	B	Positive	Data for local bus: 11	5V S
37	LD12	B	Positive	Data for local bus: 12	5V S
38	LD13	B	Positive	Data for local bus: 13	5V S
39	LD14	B	Positive	Data for local bus: 14	5V S
40	LD15	B	Positive	Data for local bus: 15	5V S
42	$\overline{\text{LWRQ}}$ (ACK)	B	Negative	Wait request for local bus	5V S
43	$\overline{\text{LIRQ}}$	B	Negative	Interrupt request for local bus	5V S
44	$\overline{\text{LIFB}}$	O	Negative	Busy interface for local bus	5V
45	$\overline{\text{LRST}}$	O	Negative	Reset for local bus	5V
47	LCLK	O	-	Local bus control block	5V
48	$\overline{\text{MRER}}$	O	Negative	When a communication error is received, this signal becomes Low level for a rated interval.	5V
49	$\overline{\text{MSEL}}$	O	Negative	When sending data to this chip, this signal becomes Low level for a rated interval.	5V
51	SOEH	O	Positive	Enable serial output	5V
52	$\overline{\text{SOEL}}$	O	Negative	Enable serial output	5V
53	SO	O	Positive	Serial output	5V
54	$\overline{\text{TOUT}}$	O	Negative	Watchdog timer output	5V
55	$\overline{\text{DNSO}}$	O	Negative	Serial output of device number for next chip	5V
57	BRK	I	Positive	Break frame send request	D 5V S

No.	Signal name	I/O	Logic	Description	Option
58	SPD0	I	-	Communication speed: 0	U 5V
59	SPD1	I	-	Communication speed: 1	U 5V
60	CKSL	I	-	Clock rate selection (L: 40 MHz, H: 80 MHz)	U 5V
61	DNSM	I	-	Device number selection mode	U 5V
62	$\overline{DN0}$	I	Negative	Device number bit 0 (common with serial input)	U 5V
63	$\overline{DN1}$	I	Negative	Device number bit 1	U 5V
64	$\overline{DN2}$	I	Negative	Device number bit 2	U 5V
65	$\overline{DN3}$	I	Negative	Device number bit 3	U 5V
66	$\overline{DN4}$	I	Negative	Device number bit 4	U 5V
67	$\overline{DN5}$	I	Negative	Device number bit 5	U 5V
69	SI	I	Positive	Serial input	D 5V S
70	SOEI	I	Positive	Enable serial output	D 5V S
72	CLK	I	-	Reference clock	S
74	$\overline{RST}$	I	Negative	Reset	S
15 26 36 46 56 71 75 76 78 79 80	GND		-	GND	
10 21 31 41 50 68 73 77	VDD		-	+3.3 V power input	

Note 1: "I/O" column, "I" refers to input, "O" refers to output and "B" refers to bi-directional. In the "I/O"

Note 2: Terminals with "5V" described in the above Option column, can be connected with the 5V TTL level IC. In the case of input terminals, they are connected with 5V CMOS, 3.3V CMOS, TTL, LVTTL, etc. In the case of output terminals, they are connected with 3.3V CMOS (5V is not available.), TT, LVTTL, etc. However, more than 3.3V cannot be supplied to output terminals. (Example: 5V CMOS can be connected using 5V pull-up resister.)

Note 3: As for terminals with "5V" described in the above Option column, please note of the followings.
- These terminals can be input 5V level signals. These are the input that overvoltage protection diode is deleted on 3.3V lines. If overvoltage may possibly be charged due to reflection, linking, or inductive noise, we recommend inserting a diode for overvoltage protection.



- Outputs (including bi-directional) from 5V devices can be connected to the center device as far as these are TTL level. (Even when a signal is pulled up to 5V, the output level will be less than 3.3 V.)

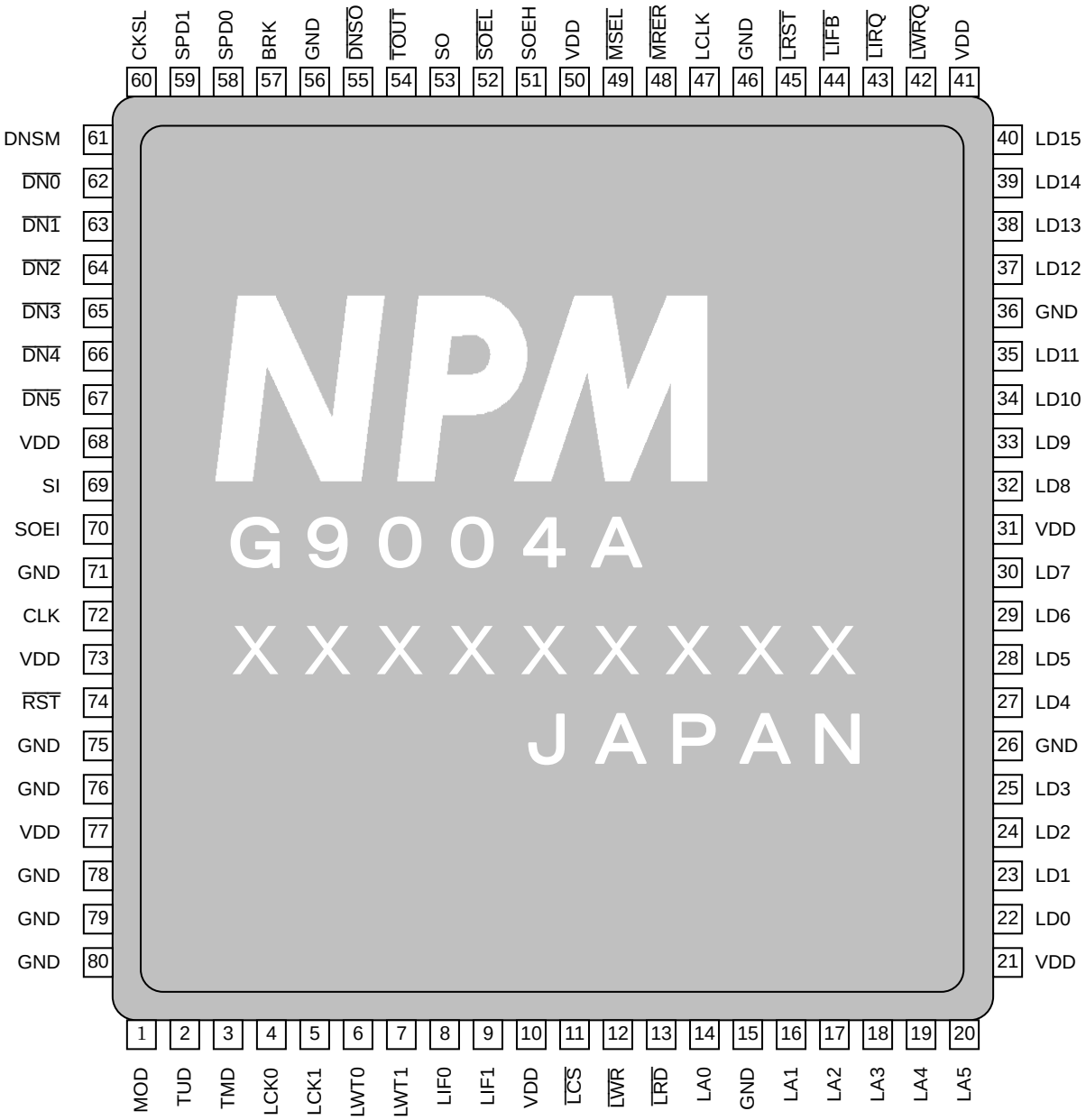
However, CMOS level signals cannot be connected.

- On the CPU bus interface, pull up of a 5 V level is possible for stabilizing bus lines (prevent floating). Use 10 k-ohm or larger capacity pull up resistors.

Note 4: Terminals with "U" described in the above Option column, have a built-in pull-up terminal. Terminals with "D" described in the above Option column, have a built-in pull-up terminal.

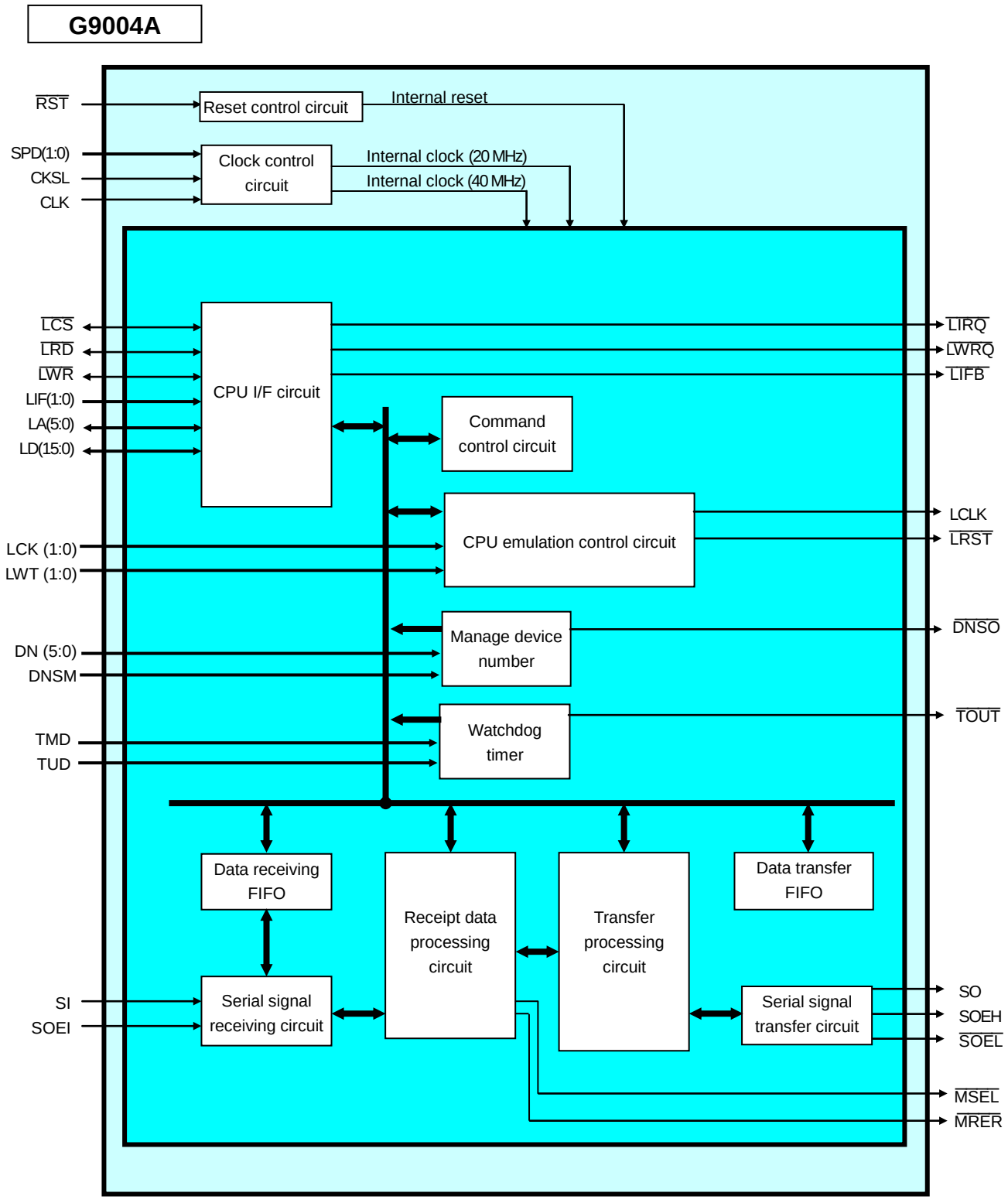
Note 5: Terminals with "S" described in the above Option column are Schmitt trigger.

4-2. Terminal assignment drawings



Note: Locate each pin number from the markings on the chip.
As shown in the figure above, pin number 1 is at the Lower left of the NPM logo mark.

4-3. Entire block diagram



5. Functions of terminals

5-1. CLK

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

CKSL = Low: 40 MHz
CKSL = High: 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, longer communication lines 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 80 MHz frequency, you may prepare a separate 40 MHz oscillator for this LSI.

5-2. $\overline{RST}$

This is an input terminal for a reset signal.
By inputting a Low level signal, the center device is reset.
The $\overline{RST}$ line must be held Low for at least 12 reference clock cycles.
After turning ON the power, a reset signal must be input before starting communication.

5-3. $\overline{DH0}$ to $\overline{DH5}$

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

5-4. DNSM

Select the input method for loading the device address.

1) When the DNSM = High
Input numeric values 0 to 3Fh with negative logic using DIP-SW etc. for $\overline{DN0}$ to $\overline{DN5}$ Terminals.

2) When the DNSM = Low
Input a $\overline{DNSO}$ signal that is output by some other chip on the $\overline{DN0}$ 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 $\overline{DN0}$ to $\overline{DN5}$ to GND.

5-5. $\overline{\text{DNSO}}$

The numeric equivalent of the address on $\overline{\text{DN0}}$ to $\overline{\text{DN5}}$ + 1 will be output after being converted into a serial bit stream.
Connect this output to another local device's $\overline{\text{DN0}}$ 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.
In the case that continuous address by $\overline{\text{DNSO}}$signal is set, it is necessary to have at least about 50 µs until the next step address is confirmed.

5-6. SPD0 to SPD1

Specify communication speed with these terminals.
All of the devices on the communication line shall be set to the same speed.

SPD1	SPD0	Communication speed
Low	Low	2.5 Mbps
Low	High	5 Mbps
High	Low	10 Mbps
High	High	20 Mbps

5-7. TUD

A watchdog timer is included on the chip to assist in administration of the communication status.
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 TUD = Low --- A reset signal is output.

5-8. 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
Low	5 ms	10 ms	20 ms	40 ms
High	20 ms	40 ms	80 ms	160 ms

5-9. TOUT

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

5-10. SO

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

5-11. SOEH, $\overline{\text{SOEL}}$

Output enable signal for communication.
Difference between SOEH and $\overline{\text{SOEL}}$ is that only the logic is different.
When sending signals, SOEH will become High and $\overline{\text{SOEL}}$ will become Low.

5-12. SOEI

When using more than one device (G9004A), connect the SOEH signal of the other device (G9004A) to this terminal.
By being wired OR with the output enable signal from this I/O device, the device outputs an enable signal to SOEH or SOEL.
When not used, connect to the GND.

5-13. SI

Serial input signal for communication. (Positive logic)

5-14. $\overline{\text{MRER}}$

This is a monitor output to check communication quality.
When the center device receives an error frame such as a CRC error, the signal becomes Low only for 128 cycles (3.2 µs) of the CLK.
By measuring the condition using the counter, you can check communication quality.

5-15. $\overline{\text{MSEL}}$

Communication status monitor output.
When this device receives a frame intended for this device and everything is normal (when communication $\overline{\text{MRER}}$ is OFF), this terminal goes Low for exactly 128 CLK cycles (3.2 µs). This can be used to check the cyclic communication time.

5-16. BRK

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

5-17. MOD

Selects the operating mode for this IC.
MOD = Low: Message communication mode
MOD= High: CPU emulation mode

5-18. LCK0 to LCK1

Selects the clock frequency (LCLK) for controlling a local bus.

LCK1	LCK 0	The clock frequency (LCLK)
Low	Low	2 MHz
Low	High	4 MHz
High	Low	10 MHz
High	High	20 MHz

When this IC is in CPU emulation mode, the speed of the emulation depends on the above setting. However, when the message communication mode is selected, only the LCLK output frequency will change, and the operation speed remains constant.

5-19. LCLK

Outputs a clock for controlling a local bus.
If needed, it can be used for an external circuit.

5-20. LIF0 to LIF1

Using these terminals, select the CPU interface specifications for the local bus.

LIF1	LIF 0	CPU-I/F interface
Low	Low	I/F-16 bit (2) (68000, etc)
Low	High	I/F-16 bit (1) (8086, H8, etc)
High	Low	I/F-8 bit (2) (6809, etc)
High	High	I/F-8 bit (1) (Z80, etc)

When IF1=Low and IF0=Low, the $\overline{\text{LWRQ}}$ terminal, the $\overline{\text{LWR}}$ terminal, and the $\overline{\text{LRD}}$ terminal become ACK signal, R/W signal and LS signal respectively.
When IF1 = High and IF0 = Low, the LRD terminals becomes an E signal.

5-21. $\overline{\text{LRST}}$

Output a reset signal for the local bus.
In any of the following cases, this signal goes Low.
1) When a Low is applied to the $\overline{\text{RST}}$ terminal.
2) When TUD is Low, the watchdog timer has timed out (only effective during approximate 32 cycles on LCLK).
The LRST terminal status can be changed by setting bit 1 of port 3 (See section 6 "Setting the status information and G9004A operation information.")
(When bit 1 = 0, $\overline{\text{LRST}}$ = High. When bit 1 = 1, $\overline{\text{LRST}}$ = Low.)

5-22. LIFB

Outputs an interface busy signal for the local bus.
Use this signal in the message communication mode when connecting to a CPU that does not have a wait control input terminal.
This terminal goes Low when a command or data is sent from a CPU, or when the status is being read. When this LSI completes its internal processing, it goes High. Make sure that this terminal is High, and then you can access the LSI.
This terminal cannot be used in the CPU emulation mode.

5-23. LA0 to LA5

Address signals for the local bus. These will be either input or output terminals, depending on the mode selected.
In the CPU emulation mode, the device outputs address signals from terminals LA0 to LA5.
In the message communication mode, input address signals on LA0 and LA1. Pull LA2 to LA5 down to GND (5~10Kohm resistors).

5-24. $\overline{\text{LCS}}$

This is a chip select signal for the local bus. It will be either an input or output terminal, depending on the mode selected.
In the CPU emulation mode, this device outputs a chip select signal for an external circuit.
In the message communication mode, you supply a chip select signal that will be used to access this LSI.

5-25. $\overline{\text{LWR}}$

This terminal is used for a write signal for a local bus. It will be either an input or output terminal, depending on the mode selected.
In the CPU emulation mode, this terminal outputs a write signal for external circuit.
In the message communication mode, you supply a write signal in order to access this LSI.

5-26. $\overline{\text{LRD}}$

This terminal is used for a read signal for a local bus. It will be either an input or output terminal, depending on the mode selected.
In the CPU emulation mode, this terminal outputs a read signal for an external circuit.
In the message communication mode, you supply a read signal in order to access this LSI.

5-27. $\overline{\text{LWRQ}}$

This terminal is used for a wait request signal for a local bus. It will be either an input or output terminal, depending on the mode selected.
In the CPU emulation mode, you supply a wait request signal from an external circuit.
In the message communication mode, this terminal outputs a wait request signal when the device cannot be accessed.

5-28. $\overline{\text{LIRQ}}$

This terminal is used for an interrupt request signal for a local bus. It will be either an input or output terminal, depending on the mode selected.
In the CPU emulation mode, you supply an interrupt request signal from an external circuit.
In the message communication mode, this terminal outputs an interrupt request signal from internal operations.

5-29. LD0 to LD7

These terminals are used for the Low-byte signals of a local data bus. These are bi-directional terminals.

5-30. LD8 to LD15

These terminals are used for the High-byte signals of a local data bus. These are bi-directional terminals.
When using an 8-bit CPU interface, pull up these terminals to VDD (5 to 10 K-ohm resistors).

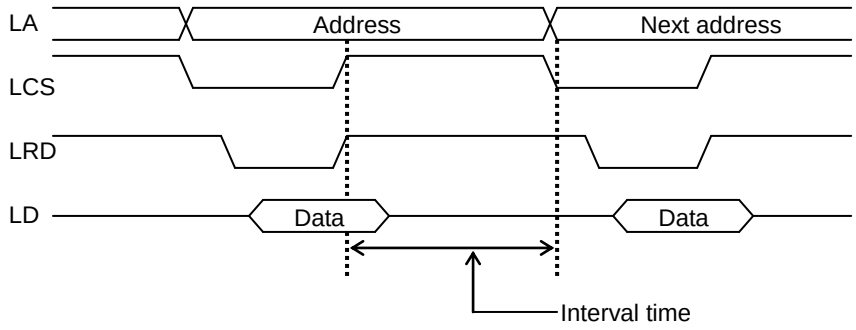
5-31. LWT0 and LWT1

These terminals are used to set the time interval between writing and reading to a local bus.
Use these terminals only in the CPU emulation mode. These are not used in the message communication mode.

(T_{LCLK}: LCLK clock cycle)

LWT1	LWT0	Time interval
Low	Low	T_{LCLK}
Low	High	$3xT_{LCLK}$
High	Low	$5xT_{LCLK}$
High	High	$9xT_{LCLK}$

[An example of the reading procedure when using a 16-bit CPU interface (1)]



5-32. CKSL

Selects the clock specifications for the input on the CLK terminal.
When CKSL = Low, supply a 40 MHz clock signal on the CLK terminal. The duty cycle should be approximately 50%.
If the duty cycle is too far away from 50%, the number of communication faults will increase.
When CKSL = High, the device uses the CLK signal input after dividing by 2 internally. Therefore, the duty cycle will not have such a great influence. In this case, supply an 80 MHz clock signal.

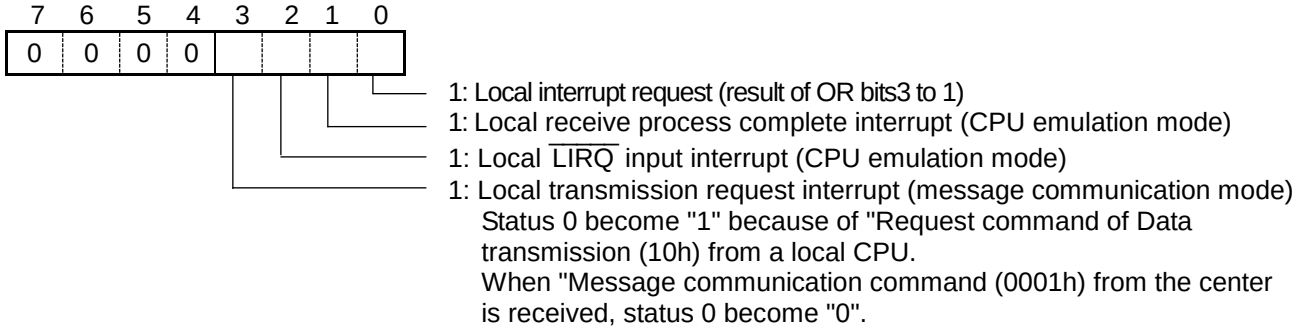
5-33. VDD, GND

Supply DC +3.0V to +3.6V for power on the VDD terminal.
Make sure to use all the terminals.

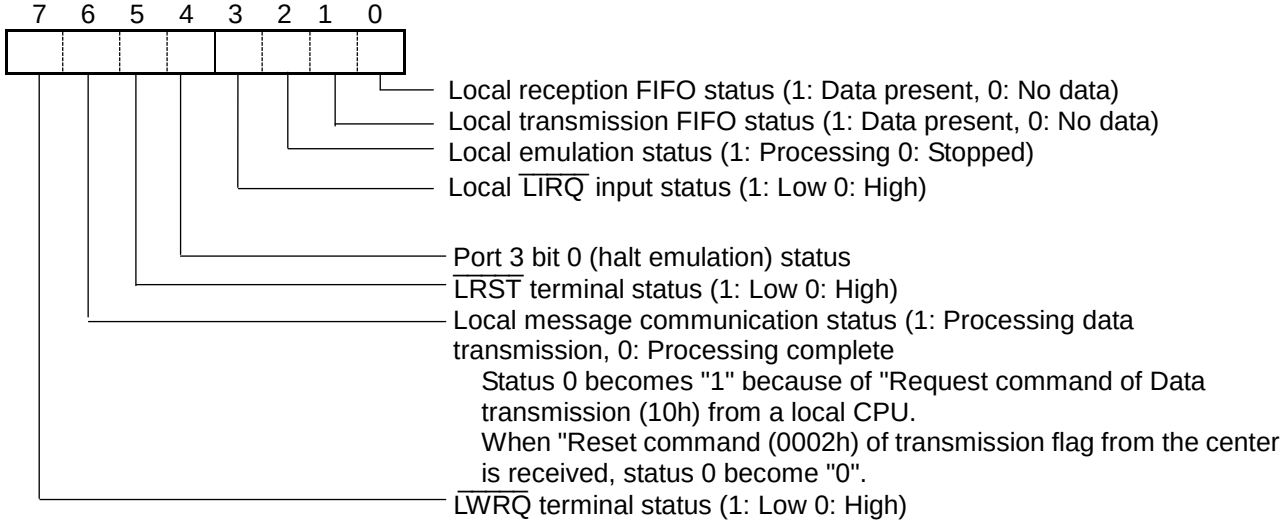
6. Setting the status and operation information for the G9004A

During the cyclic communication and during data communication, the status register information (registers 0 to 2) is written to the port data area (ports 0 to 2) that corresponds to the device address in the center device. When the operating information is written to the port data area (port 3) that corresponds to the center device's device address, it is passed along to the CPU emulation device (G9004A) using the cyclic communication.

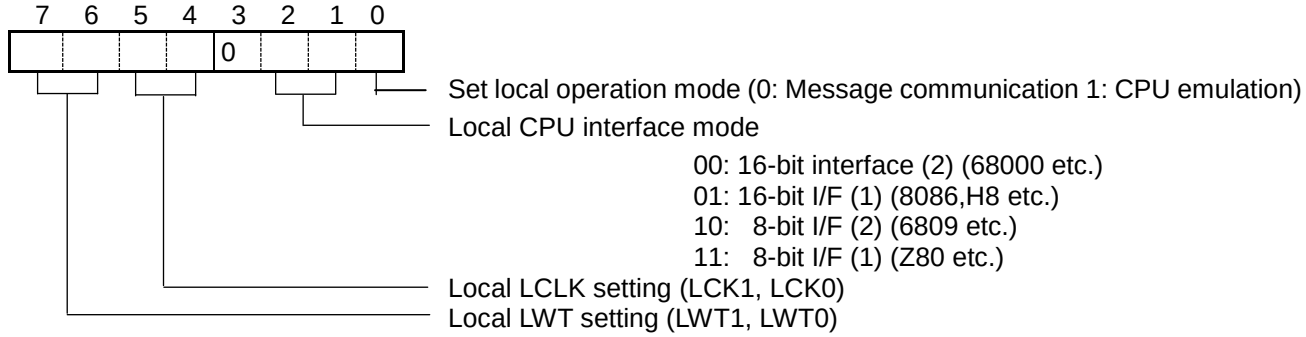
Status 0 (Interrupt information) <Port 0>



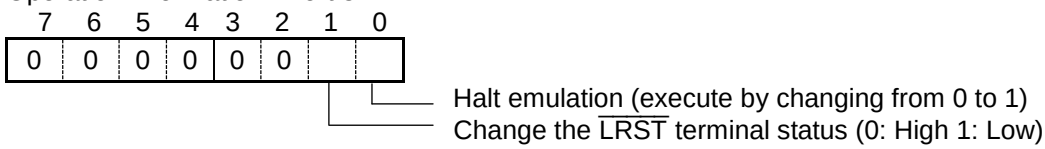
Status 1 (Status information) <Port 1>



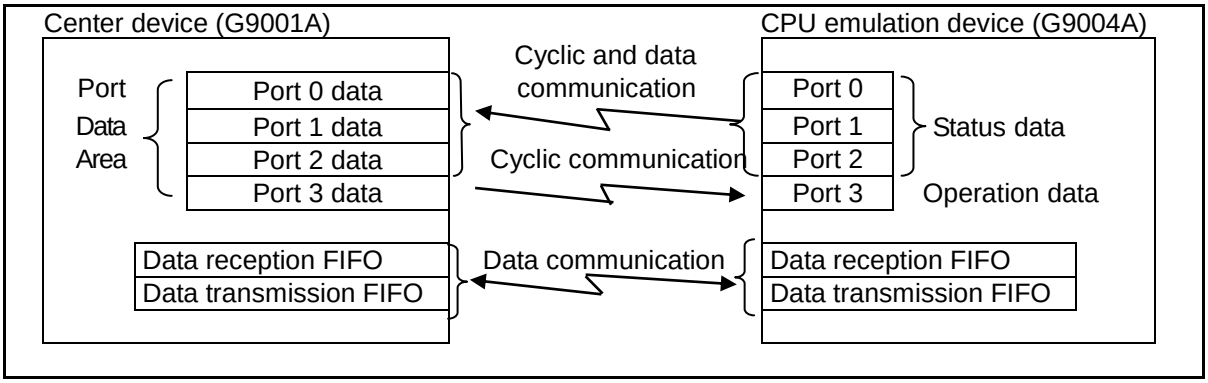
Status 2 (setting information) <Port 2>



Operation information <Port 3>

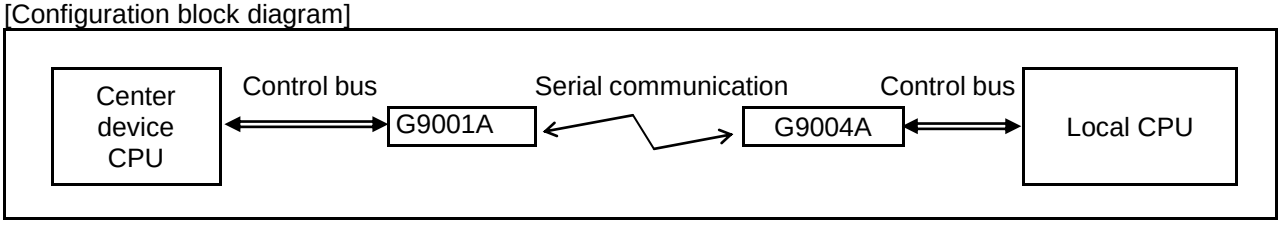


[Communication image]



7. Message communication mode (MOD = Low)

[Configuration block diagram]



7-1. Terminals for use by a local CPU

Terminal name	I/O	Logic	Description
ILF0	I		Local bus interface mode 0
ILF1	I		Local bus interface mode 1
$\overline{\text{LRST}}$	O	Negative	Resets the local bus
LA0 to LA1	I	Positive	Address for local bus lines 0 to 1
LA2 to LA5	I	Positive	Pull down to GND (5 to 10Kohm resistors)
$\overline{\text{LCS}}$	I	Negative	Chip select for the local bus
$\overline{\text{LWR}}$	I	Negative	Write signal for the local bus
$\overline{\text{LRD}}$	I	Negative	Read signal for the local bus
$\overline{\text{LWRQ}}$	O	Negative	Wait request for the local bus
$\overline{\text{LRIFB}}$	O	Negative	Interface busy for the local bus
$\overline{\text{LIRQ}}$	O	Negative	Interrupt request for the local bus
LD0 to LD7	B	Positive	Low-byte signal for the local bus
LD8 to LD15	B	Positive	High-byte signal for the local bus

7-2. I/O map for the local CPU

16-bit interface (1) and (2) (8086, H8, and 68000) [LIF1 = Low]

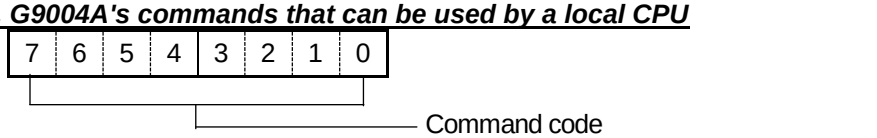
LA1	Writing	Reading
0	Write command	Read status
1	Write transmission data	Read reception data

8-bit I/f (1) and (2) (Z80, 6809 etc.) [LIF1 = High]

LA0 to LA1	Writing	Reading
00	Write command	Read status
01	Invalid	Invalid
10	Write transmission data	Read reception data
11	Invalid	Invalid

7-3. Command and status information that can be used by a local CPU

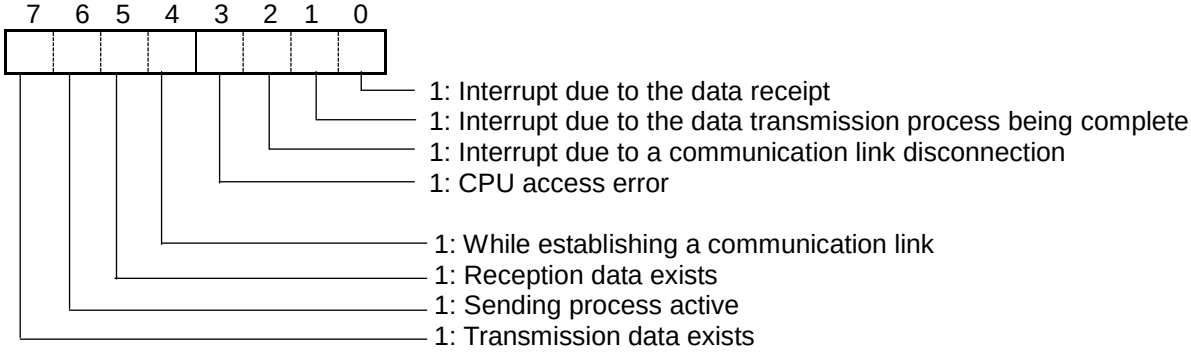
7-3-1. G9004A's commands that can be used by a local CPU



Code	Description
00h	No meaning. (Does not affect the operation)
01h	Software reset. Has the same result as applying a Low to the $\overline{RST}$ terminal. See the Note
02h	Resets only the data transmission FIFO. If written during the data sending process (status register bit 6 = 1) it may cause a CPU access error (status register bit 3 = 1).
03h	Resets only the data reception FIFO. Writing this command while no data is being received (status register bit 5 = 0) may cause a CPU access error (status register bit 3 = 1).
04h	The device enters a break frame waiting status. This has the same results as applying a High pulse to the BRK terminal.
10h	Data send request. Make the status register 0 bit 3 in the CPU emulation device equal 1 (set the sending request interrupt), so that the center device will be notified that transmission data exists.

Note: When turning ON the power, a reset signal must be supplied ($\overline{RST}$ = Low).

7-3-2. G9004A status information that can be seen from a local CPU



Bit	Item	Description
0	Data receipt interrupt	When the G9004A receives message data from the center device, this bit becomes 1 and an interrupt signal is output ($\overline{LIRQ}$ = Low). After this status register is read, this bit returns to 0. When the device only receives an information command, this bit stays 0 and the device does not output an interrupt.
1	Data transmission process complete interrupt	After writing a data transmission command (10h), when the G9004A receives an information command (reset transmission processing flag: 0002h), this bit becomes 1 and the device outputs an interrupt signal ($\overline{LIRQ}$ = Low). After this status register is read, this bit returns to 0.

Bit	Item	Description
2	Communication link disconnected interrupt	When the interval between sending one data sentence and the next from the center device to the PCL exceeds a specified time (time out), this bit becomes 1, and the device outputs an interrupt signal ($\overline{\text{LIRQ}}$ = Low). After reading this status register, the interrupt signal is reset. This is used to monitor the watchdog timer output (1: $\overline{\text{TOUT}}$ = Low)
3	CPU access error	When the PCL device does any of the folLowing on a command from a CPU, this bit becomes 1, and the device outputs an interrupt signal ($\overline{\text{LIRQ}}$ = Low). After this status register is read, this bit returns to 0. - Try to write data to the transmission FIFO during transmission processing (status register bit 6 = 1), or writing a transmission FIFO reset command (02h). - Try to read the reception FIFO even though it has received no data (status register bit 5 = 0), or writing a data reception FIFO reset command (03h).
4	Communication link valid	This bit is used to see if the communication line is connected. When the device detects a change on the communication line, this bit becomes 1. If the communication line does not change within a specified time, this bit becomes 0.
5	Received data exists	When there is message data in the data reception FIFO, this bit becomes 1.
6	Data transmission in progress	After writing the data transmission command (10h), this bit becomes 1. When data communication with the center device is complete (when a receiving information command 0002h issued by the center device), this bit becomes 0.
7	Transmission data exists	If there is message data in the data transmission FIFO, this bit becomes 1.

7-4. Information command for the center device (G9001A)

To send a message, the first word in the transmission FIFO of the center device is used for the information command.
The remaining 127 words can be used freely as message data (any format is alLowed).

Center device transmission FIFO		
Address	Upper	Lower
00h	Information command	
01h to 7Fh	Message data (any format)	

[Information command]

Command code	Description
0001h	Message transmission. Use this command to send a message or data from the center device to a local CPU. You can attach a message or data at the end of this command. (There is also an Information command without any message.) Transmits FIFO data of G9004A for sending to G9001A.
0002h	When the G9004A receives this command, the data transmission process complete interrupt < local status register bit 1 = 1 > and the device resets the transmission processing flag < local status register bit 6 >. You can attach a message or data at the end of this command.
0003h	Resend request command When the G9004A receives this command, it sends the same data as it last sent. Use this command when the center device cannot receive data from the local device due a data communication error or other reason. Note. Use this when sending request is failed because of some causes. In the case that the 0001h command is used in place of resend request command, the operation of 0003h command after that is not guaranteed.

7-5. Message communication procedure

The device model numbers are shown in parenthesis.
Numbers marked with ( ) mean that the operations are carried out by a CPU.
Assume that the local device address (for the G9004A) is "08h."
In addition, the port status information for the Cycle communication or data communication is sent to the port data area in the center device (G9001A). In order to generate an interrupt (position *1) in the center device (G9001A) when an interrupt request (bit 0 = 1 on port 0) occurs in the local device (G9004A), you must enable the input change interrupt that corresponds to port 0 (set it to 1).

(1) When the center device (G9001A) is the first to send a message and the local device (G9004A) responds

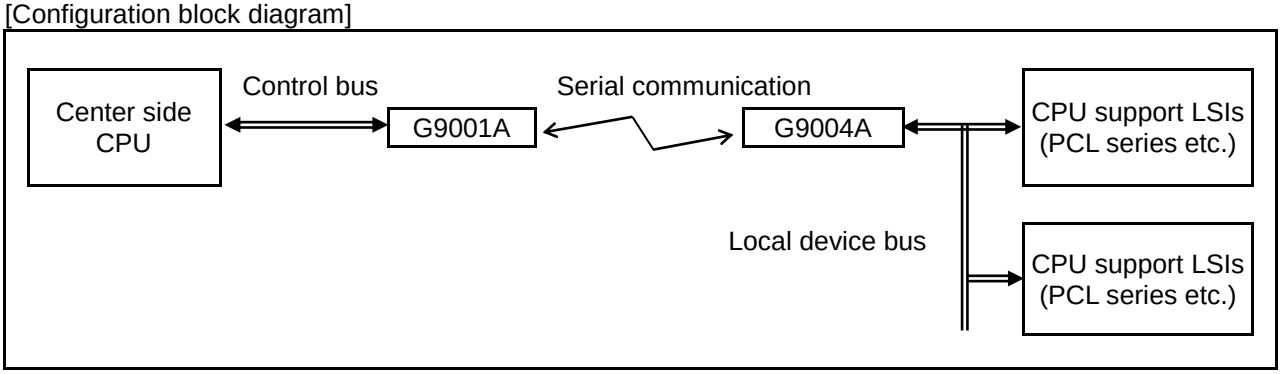
[Center device (G9001A)]	[Local device (G9004A)]
[Start]	
(1) Write an information command (0001h: Message transmission) to the data transmission FIFO. (2) Place a message in the data transmission FIFO. (3) Write a data communication command (4008h).	
****************** Data communication (1st time) ******************	
- Interrupt occurs (4) The center device reads the status. ▽ Data transmission complete <bit 0=1>	- Interrupt occurs (1) Read the status ▽ Data reception interrupt <bit 0 = 1> (2) Read the message in the data reception FIFO. (3) Place the message in the data transmission FIFO. (4) Write a data transmission command (10h). - Set the status for port 0 (interrupt information) ▼ Local device data transmission request interrupt <Bit 3 = 1>
*1 - Interrupt occurs ← - - - - - - - - - - - -	- - ▼ Local device interrupt request <Bit 0 = 1>
(5) The center device reads the status. ▽ Input change interrupt <bit 2 = 1> (6) Input change interrupt (7) Send an information command to the data transmission FIFO. Write (0001h: Message transmission) (8) Write a data communication command (4008h).	
****************** Data communication (2nd time) ******************	When the Local device receives a message from the center device, it sends the data in the FIFO to the center device.
- Interrupt occurs (9) Read the status ▽ Data communication complete <bit 0=1> (10) Read the message in the data reception FIFO. (11) Send an information command (0002h: reset the data transmission processing flag) to the data transmission FIFO. This is how the center device informs the local device that the data was received normally. (12) Write a data communication command (4008h).	
****************** Data communication (3rd time) ******************	
- Interrupt occurs (13) Read the center device status ▽ Data communication complete <bit 0 = 1>	- Reset the data transmission FIFO. - Interrupt occurs (5) Read the local device status ▽ Transmission process complete interrupt <bit 1 = 1>

(2) When the local device (G9004A) is the first to send a message and the center device (G9001A) responds

[Center device (G9001A)]	[Local device (G9004A)]
[Start]	
*1 - Interrupt occurs ←・—・—・—・—・— (1) Read the center device status ▽ Input change interrupt <bit 2=1> (2) Reset the input change interrupt (3) Send an information command (0001h: message transmission) to the data transmission FIFO. (4) Place a message in the data transmission FIFO. (5) Write a data communication command (4008h).	(1) Place a message in the data transmission FIFO. (2) Write a data send command (10h). - Set the status for port 0 (interrupt information) ▼ Local device data send request interrupt <bit 3 = 1> ▼ Local device interrupt request <bit 0 = 1>
********** Data communication (1st time) **********	
	After receiving a message from the center device, the local device sends the data in the FIFO to the center device.
- Interrupt occurs (6) The center device reads the status. ▽ Data communication compete <bit 0 = 1> (7) Read the message in the data reception FIFO. (8) Send an information command (0002h: Reset the data send processing flag) to the data transmission FIFO. This is how the center device informs the local device that the data was received normally. (9) Write a data communication command (4008h).	- Interrupt occurs (3) The local device reads the status. ▽ Data received interrupt <Bit 0=1> (4) Read the message in the data reception FIFO.
********** Data communication (2nd time) **********	
- Interrupt occurs (10) Read the status ▽ Data communication compete <bit 0 = 1>	- Reset data transmission FIFO. - Interrupt occurs (5) Read the status of the local device ▽ Data transmission process complete interrupt <bit 1=1>
[End of message communication]	

8. CPU emulation mode (MOD = High)

[Configuration block diagram]



8-1. Terminals on the G9004A

Terminal name	I/O	Logic	Description
LIF0	I		Local bus interface mode 0
LIF1	I		Local bus interface mode 1
LWT0	I	Positive	Local bus interval time setting 0
LWT1	I	Positive	Local bus interval time setting 1
$\overline{\text{LRST}}$	O	Negative	Local bus reset
LA0 to LA5	O	Positive	Local bus address
$\overline{\text{LCS}}$	O	Negative	Local bus chip select
$\overline{\text{LWR}}$	O	Negative	Local bus write signal
$\overline{\text{LRD}}$	O	Negative	Local bus read signal
$\overline{\text{LWRQ}}$	I	Negative	Local bus wait request
$\overline{\text{LIRQ}}$	I	Negative	Local bus interrupt request
LD0 to LD7	B	Positive	Local bus data 0 to 7
LD8 to LD15	B	Positive	Local bus data 8 to 15

Shown below are representative CPUs and the corresponding terminals on the G9004A.

Typical CPU		G9004A	
CPU name	Terminal name	G9004A terminal name	G9004A mode
Z80	$\overline{\text{RD}}$	$\overline{\text{LRD}}$	8 bit interface (1)
	$\overline{\text{WR}}$	$\overline{\text{LWR}}$	
8086	$\overline{\text{RD}}$	$\overline{\text{LRD}}$	16 bit interface (1)
	$\overline{\text{WR}}$	$\overline{\text{LWR}}$	
	Upper enable	None	
	Lower enable	None	
68000	R/$\overline{\text{W}}$	$\overline{\text{LWR}}$	16 bit interface (2)
	Upper strobe	$\overline{\text{LRD}}$ (No distinction between upper/Lower)	
	Lower strobe	$\overline{\text{LRD}}$ (No distinction between upper/Lower)	
H8	R/$\overline{\text{W}}$	$\overline{\text{LRD}}$	16 bit interface (1)
	Upper WR	$\overline{\text{LWR}}$ (No distinction between upper/Lower)	
	Lower WR	$\overline{\text{LWR}}$ (No distinction between upper/Lower)	
6809	R/$\overline{\text{W}}$	$\overline{\text{LWR}}$	8 bit interface (2)
	E	$\overline{\text{LRD}}$	

8-2. Control method for using a center device (G9001A)

When you want to perform CPU emulation using the G9004A, write the commands and data using the formats specified for the data transmission FIFO in the center device (G9001A). The center device's data transmission FIFO can store up to 128 words, including write commands, write data, and read commands.

When writing data, there is no data to be sent from the G9004A. But, when reading data, the G9004A will send a read command and then read data from the center device

Therefore, more than 128 words of data cannot be handled by the G9004A.

[When writing data]

Center device data transmission FIFO		
Address	Upper	Lower
00h	Write command	
01h	Writing data (1)	
02h	Writing data (2)	
-	-	
-	-	
-	-	

[When reading data]

Center device data transmission FIFO		
Address	Upper	Lower
00h	Read command	
01h	Read command	
02h	Read command	
-	-	
-	-	
-	-	

[Combined case]

Center device data transmission FIFO		
Address	Upper	Lower
00h	Write command	
01h	Writing data (1)	
02h	Writing data (2)	
03h	Read command	
04h	Read command	
05h	Writing data	
-	-	
-	-	

If the G9004A is set up to use an 8-bit CPU interface, the folLowing precautions should be noted.

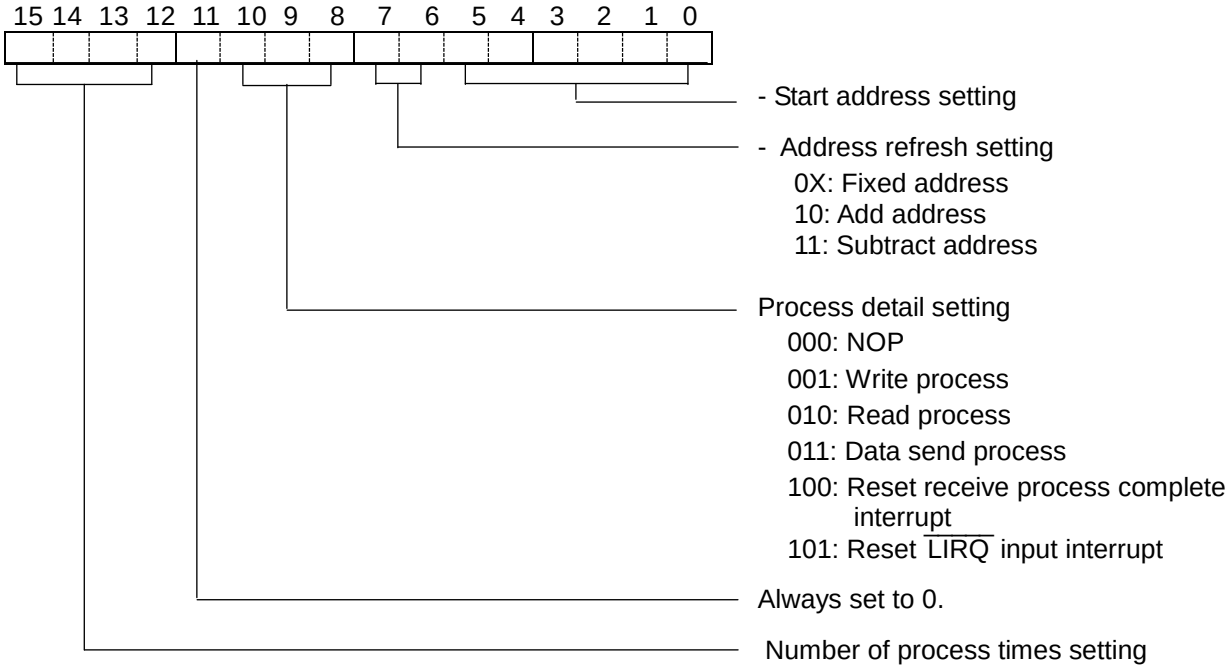
- When the center device (G9001A) uses a 16-bit interface
  In the case that the number of writing for G9004 is odd number, add one byte of dummy data. This dummy data is ignored on the G9004A side (Writing operation is not executed.), the next operation command will be processed.
  Additionally, in the case that writing of odd number bytes is executed several time simultaneously, the data is returned in the packed state so as to cross the word boarder. (On the G9004A side, any dummy data are not added.)

- When the center device (G9001A) uses an 8-bit interface
  The operation command for the G9004A must be written to the same address in the center device (G9001A) data transmission FIFO by arranging the upper and lower bytes in order. If the number of data bytes to write is odd, write one dummy data byte to make the total even. Then use the following operation commands for the G9004A at the same address.
  Shown below is an example where 3 bytes of data are to be written. Write a dummy data byte to the upper 8 bits of address 02h.

Center device (G9001A) data transmission FIFO data		
Address	Upper 8 bytes	Lower 8 bytes
00h	Write command (upper)	Write command (Lower)
01h	Write data 2	Write data 1
02h	Write data 4 (dummy data)	Write data 3
03h	Write command (upper)	Write command (Lower)
04h	Write data 5	Write data 4
05h	-	-
06h	-	-

Write the operation commands to the same addresses.

8-2-1. Command



◆ Start address setting
- Set the top address for the read or write process.
- If the CPU emulation device (G9004A) is using a 16-bit CPU interface, the Lower bit setting is ignored and always treated as 0.
- Commands other than writing and reading processes are invalid.

◆ Address refresh setting
- When the number of processing times does not equal zero, select an address refresh method
- When the number of processing times is 0, commands other than writing and reading processes are invalid.

◆ Process detail setting
- Select the processing detail for the CPU emulation device (G9004A)

Item	Processing detail
NOP	"Do-nothing" command. The device does nothing. Reception process complete interrupt does not occur. When sending FIFO for G9004A has any data, do not use this command. (The operation is not guaranteed.)
Read process	The CPU emulation device (G9004A) reads the number of bytes specified in the processing register, starting from the start address.
Write process	The CPU emulation device (G9004A) writes the number of bytes specified in the processing register, starting from the start address.
Data send process Note	The device sends the data that was read by using the read process to the center device. In addition, if the reception process complete interrupt flag (bit 1 on port 0) is 1, and this command is received, the device will send the same data as it last sent (the data resend process).
Reset the reception process complete interrupt flag Note	Reset the reception process complete interrupt flag (bit 1 on port 0) to 0. In the case that the bit number 1 of the state 1 is "1", the value returned to 0 after receiving this command.
LIRQ input interrupt reset Note	Reset the LIRQ input interrupt (bit 2 on port 0) to 0.

Note. Please use it with the beginning word of sending data without fail. If it is used at except the beginning, it is ignored.

◆ Processing bytes setting

- Set the number of processing bytes when you want to read or write continuously. (Set a burst cycle). Enter (the number of processing bytes - 1) as the setting.

When the device is used with a 16-bit CPU interface, set the number in units of words. Otherwise, when the device is used with an 8-bit CPU interface, set the number in units of bytes.

Commands other than write/read processes are not allowed.

8-2-2. Examples of CPU emulation control procedures

The device model number is shown in parenthesis.
Numbers marked with ( ) mean that the operations are carried out by a center device (G9001A) emulating a CPU.
Assume that the local device address (for the G9004A) is "08h."
In addition, the port status information of the cyclic communication or data communication is sent to the port data area of the center device (G9001A). In order to generate an interrupt (position *1) in the center device (G9001A) when an interrupt request (bit 0 = 1 on port 0) occurs in the local device (G9004A), you must enable the input change interrupt that corresponds to port 0 (set it to 1).

8-2-2-1. Examples of writing single units of data (16-bit CPU interface)

[Write the data 1234h to address 2]

<table>
<tr><th>[Center device (G9001A)]</th><th>[Local device (G9004A)]</th></tr>
<tr><td colspan="2">[Start]</td></tr>
<tr><td>(1) Place a command (0102h) in the data transmission FIFO.
(2) Place the data (1234h) into the data transmission FIFO.
(3) Write a data communication command (4008h).
<table>
<tr><th colspan="3">Data transmission FIFO</th></tr>
<tr><th>Address</th><th>Upper</th><th>Lower</th></tr>
<tr><td>00h</td><td>01h</td><td>02h</td></tr>
<tr><td>01h</td><td>12h</td><td>34h</td></tr>
</table>
</td><td></td></tr>
<tr><td colspan="2">********** Data communication (1st time) **********</td></tr>
<tr><td>- Interrupt occurs
(4) The center device reads the status
▽ Data communication complete <bit 0=1>

*1
- Interrupt occurs ←・ー・ー・ー・ー・ー・
(5) The center device reads the status
▽ Input change interrupt<Bit 2=1>
(6) Reset the input change interrupt
(7) Write (0400h) to the data transmission FIFO. [Reset reception process complete interrupt].
<table>
<tr><th colspan="3">Data transmission FIFO</th></tr>
<tr><th>Address</th><th>Upper</th><th>Lower</th></tr>
<tr><td>00h</td><td>04h</td><td>00h</td></tr>
</table>
(8) Write a data communication command (4008h)</td><td>1. Write 1234h to address 2.
2. Set status port 0 (interrupt information)
▼ Receive process complete interrupt <bit 1=1>
▼ Interrupt request <Bit 0=1></td></tr>
<tr><td colspan="2">********** Data communication (2nd time) **********</td></tr>
<tr><td>- Interrupt occurs
(9) The center device reads the status.
▽ Data communication complete <Bit 0=1></td><td></td></tr>
<tr><td colspan="2">[End emulation communication]</td></tr>
</table>

8-2-2-2. Example of writing continuous data (16-bit CPU interface)

[Write 1234h to address 4, 2345h to address 6, and 3456h to address 8]

<table>
<tr><th>[Center device (G9001A)]</th><th>[Local device (G9004A)]</th></tr>
<tr><td colspan="2">[Start]</td></tr>
<tr><td>(1) Place a command (2184h) in the data transmission
<table>
<tr><th colspan="3">Data transmission FIFO</th></tr>
<tr><th>Address</th><th>Upper</th><th>Lower</th></tr>
<tr><td>00h</td><td>21h</td><td>84h</td></tr>
<tr><td>01h</td><td>12h</td><td>34h</td></tr>
<tr><td>02h</td><td>23h</td><td>45h</td></tr>
<tr><td>03h</td><td>34h</td><td>56h</td></tr>
</table>
(2) Place data (1234h) in the data transmission FIFO.
(3) Place data (2345h) in the data transmission FIFO.
(4) Place data (3456h) in the data transmission FIFO.
(5) Write a data communication command (4008h).</td><td></td></tr>
<tr><td colspan="2">********** Data communication (1st time) **********</td></tr>
<tr><td>- Interrupt occurs
(6) The center device reads the status.
▽ Data communication complete <bit 0=1>
*1
Interrupt occurs
(7) The center device reads the status.
▽ Input change interrupt <bit 2=1>
(8) Reset the input change interrupt
(9) Place a command (0400h) in the data transmission FIFO. [Reset reception process complete interrupt]
<table>
<tr><th colspan="3">Data transmission FIFO</th></tr>
<tr><th>Address</th><th>Upper</th><th>Lower</th></tr>
<tr><td>00h</td><td>04h</td><td>00h</td></tr>
</table>
(10) Write a data communication command (4008h).</td><td>1. Write 1234h to address 4.
2. Write 2345h to address 6.
3. Write 3456h to address 8.
4. Set the status on port 0 (interrupt information).
▼ Receive process complete interrupt <Bit 1 = 1>
▼ Interrupt request <Bit 0=1></td></tr>
<tr><td colspan="2">********** Data communication (2nd time) **********</td></tr>
<tr><td>- Interrupt occurs
(11) Center device reads the status.
▽ Data communication complete <Bit 0=1></td><td></td></tr>
<tr><td colspan="2">[End emulation communication]</td></tr>
</table>

8-2-2-3. Example of reading continuous data (16-bit CPU interface)

[Read data addresses 16, 14, 12, and 10 in this order]

<table>
<tr><th>[Center device (G9001A)]</th><th>[Local device (G9004A)]</th></tr>
<tr><td colspan="2">[Start]</td></tr>
<tr><td>(1) Place a command (32D0h) in the data transmission FIFO.
<table><tr><th colspan="3">Data transmission FIFO</th></tr><tr><th>Address</th><th>Upper</th><th>Lower</th></tr><tr><td>00h</td><td>32h</td><td>D0h</td></tr></table>
(2) Write a data communication command (4008h).</td><td></td></tr>
<tr><td colspan="2">********** Data communication (1st time) **********</td></tr>
<tr><td>- Interrupt occurs
(3) The center device reads the status
▽ Data communication complete <bit 0=1></td><td>1. Place a command (32D0h) in the data transmission FIFO.
<table><tr><th colspan="3">Data transmission FIFO</th></tr><tr><th>Address</th><th>Upper</th><th>Lower</th></tr><tr><td>00h</td><td>3Ah</td><td>D0h</td></tr><tr><td>01h</td><td>FFh</td><td>EEh</td></tr><tr><td>02h</td><td>DDh</td><td>CCh</td></tr><tr><td>03h</td><td>BBh</td><td>AAh</td></tr><tr><td>04h</td><td>99h</td><td>88h</td></tr></table>
2. Read from address 16 (FFEEh) and write to the data transmission FIFO.
3. Read from address 14 (DDCCh) and write to the data transmission FIFO.
4. Read from address 12 (BBAAh) and write to the data transmission FIFO.
5. Read from address 10 (9988h) and write to the data transmission FIFO.
6. Set the status for port 0 (interrupt information)
▼ Receive process complete interrupt <Bit 1=1>
▼ Interrupt request <Bit 0=1></td></tr>
<tr><td>*Note 1
- Interrupt occurs ←
(4) The center device reads the status
▽ Input change interrupt <bit 2=1>
(5) Reset the input change interrupt
(6) Place a command (0300h) in the data transmission FIFO.
<table><tr><th colspan="3">Data transmission FIFO</th></tr><tr><th>Address</th><th>Upper</th><th>Lower</th></tr><tr><td>00h</td><td>03h</td><td>00h</td></tr></table>
(7) Write a data communication command (4008h).</td><td></td></tr>
<tr><td colspan="2">Send the content of the transmission FIFO to the center</td></tr>
<tr><td colspan="2">********** Data communication (2nd time) **********</td></tr>
<tr><td>- Interrupt occurs
(8) Read the center device status
▽ Data communication complete <bit 0=1>
(9) Read the data from reception FIFO.
(10) Write a command (0400h) in the data [Interrupt reset of reception process completion]
<table><tr><th colspan="3">Data transmission FIFO</th></tr><tr><th>Address</th><th>Upper</th><th>Lower</th></tr><tr><td>00h</td><td>04h</td><td>00h</td></tr></table>
(11) Write a data communication command (4008h)</td><td></td></tr>
<tr><td colspan="2">********** Data communication (3rd time) **********</td></tr>
<tr><td>Interrupt occurs
(12) Read the center device status
▽ Data communication complete <bit 0=1></td><td></td></tr>
<tr><td colspan="2">[End emulation communication]</td></tr>
</table>

8-2-2-4. Example of combined processing (16-bit CPU interface)

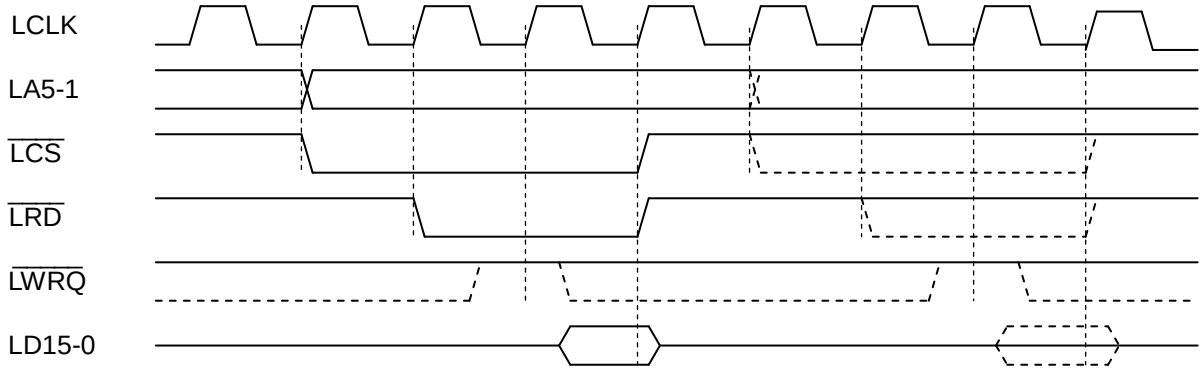
[After writing 1234h to address 0, read addresses 4 and 6]

<table>
<tr><th>[Center device (G9001A)]</th><th>[Local device (G9004A)]</th></tr>
<tr><td colspan="2">[Start]</td></tr>
<tr><td>(1) Place a command (0100h) in the data transmission FIFO.
(2) Place data (1234h) in the data transmission FIFO.
(3) Place a command (1284h) in the data transmission FIFO.
(4) Write a data communication command (4008h).
<table>
<tr><th colspan="3">Data transmission FIFO</th></tr>
<tr><th>Address</th><th>Upper</th><th>Lower</th></tr>
<tr><td>00h</td><td>01h</td><td>00h</td></tr>
<tr><td>01h</td><td>12h</td><td>34h</td></tr>
<tr><td>02h</td><td>12h</td><td>84h</td></tr>
</table>
</td><td></td></tr>
<tr><td colspan="2">********** Data communication (1st time) **********</td></tr>
<tr><td>- Interrupt occurs
(5) The center device reads the status.
▽ Data communication complete <Bit 0=1></td><td>1. Write 1234h to address 0.
2. Place a command (1284h) in the data transmission FIFO.
3. Read from address 4 (FFEEh) and write it to the data transmission FIFO.
4. Read from address 6 (DDCCh) and write it to the data transmission FIFO.
5. Set the status on port 0 (interrupt information).
▼ Receive process complete interrupt <Bit 1 = 1>
▼ Interrupt request <Bit 0=1>
<table>
<tr><th colspan="3">Data transmission FIFO</th></tr>
<tr><th>Address</th><th>Upper</th><th>Lower</th></tr>
<tr><td>00h</td><td>12h</td><td>84h</td></tr>
<tr><td>01h</td><td>FFh</td><td>EEh</td></tr>
<tr><td>02h</td><td>DDh</td><td>CCh</td></tr>
</table>
</td></tr>
<tr><td>*1
- Interrupt occurs ←
(6) The center device reads the status.
▽ Input change interrupt <Bit 2=1>
(7) Reset the input change interrupt. Write (0300h) to the data transmission FIFO.
[Data sending process]
(8) Write a data communication command (4008h).
<table>
<tr><th colspan="3">Data transmission FIFO</th></tr>
<tr><th>Address</th><th>Upper</th><th>Lower</th></tr>
<tr><td>00h</td><td>03h</td><td>00h</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>Send the contents of the data transmission FIFO to the center device.</td></tr>
<tr><td colspan="2">********** Data communication (2nd time) **********</td></tr>
<tr><td>- Interrupt occurs
(10) The center device reads the status.
▽ Complete data communication <bit 0=1>
(11) Read the data from the data receipt FIFO.
(12) Place a command (0400h) in the data transmission FIFO.
[Reset the reception process complete interrupt]
(13) Write a data communication command (4008h).
<table>
<tr><th colspan="3">Data transmission FIFO</th></tr>
<tr><th>Address</th><th>Upper</th><th>Lower</th></tr>
<tr><td>00h</td><td>04h</td><td>00h</td></tr>
</table>
</td><td></td></tr>
<tr><td colspan="2">********** Data communication (3rd time) **********</td></tr>
<tr><td>- Interrupt occurs
(14) The center device reads the status.
▽ Compete data communication <Bit 0=1></td><td></td></tr>
<tr><td colspan="2">[End emulation communication]</td></tr>
</table>

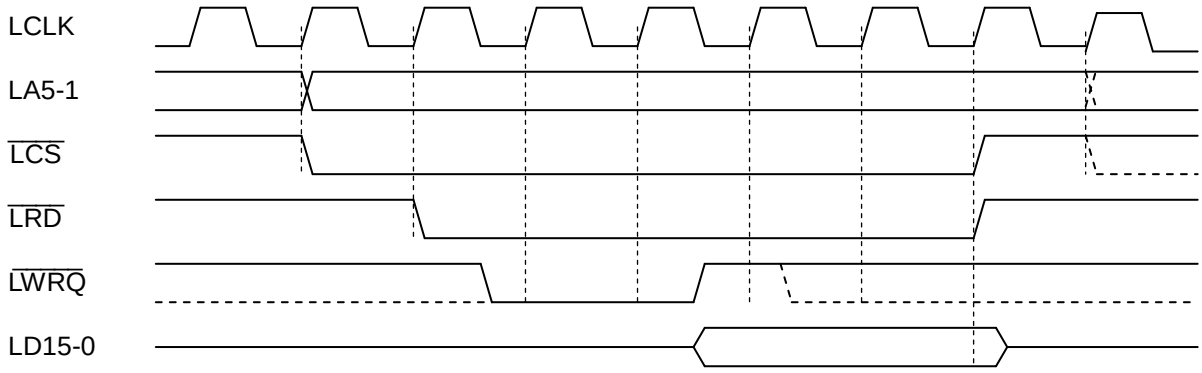
8-3. Emulation timing

8-3-1. 16-bit CPU I/F (1) (8086, H8 etc.)

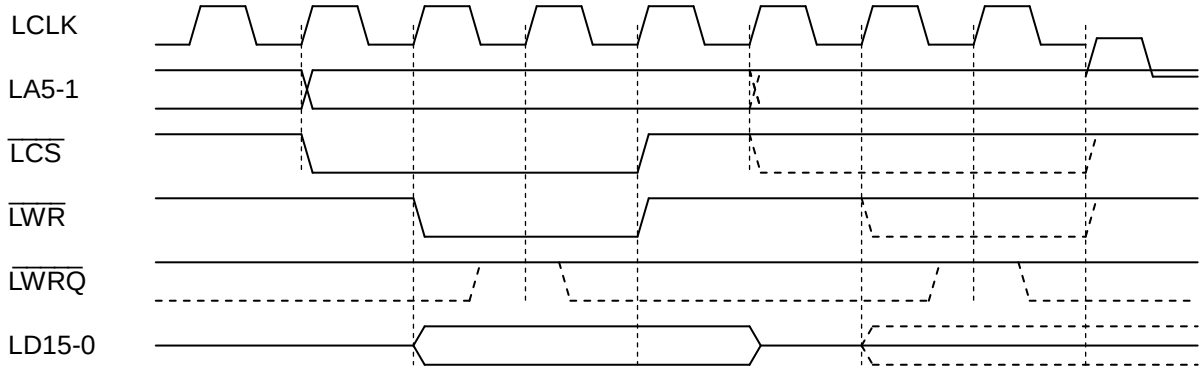
8-3-1-1. Read cycle (1) (without a wait cycle)



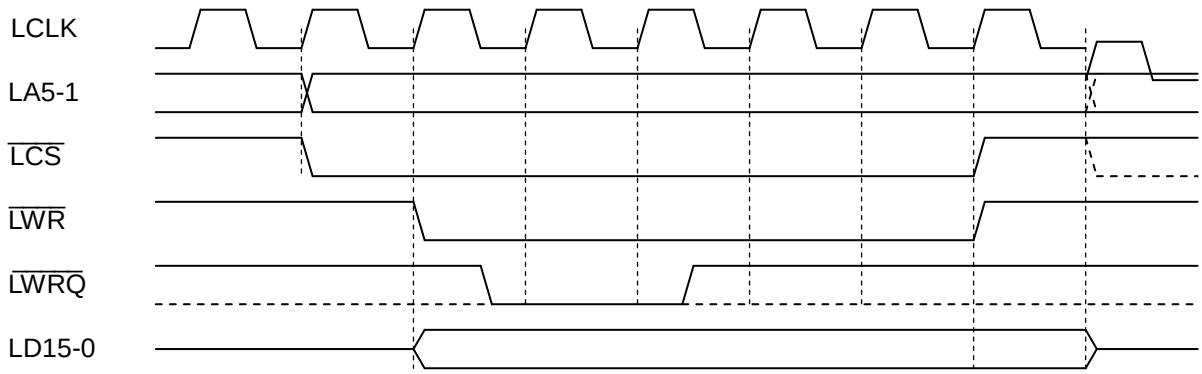
8-3-1-2. Read cycle (2) (wait 2 cycles)



8-3-1-3. Write cycle (1) (without a wait cycle)

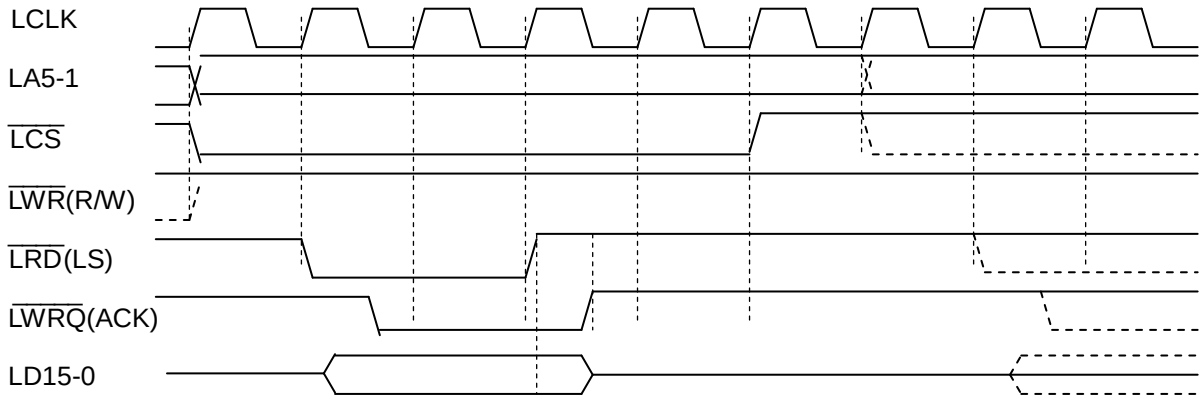


8-3-1-4. Write cycle (2) (wait 2 cycle)

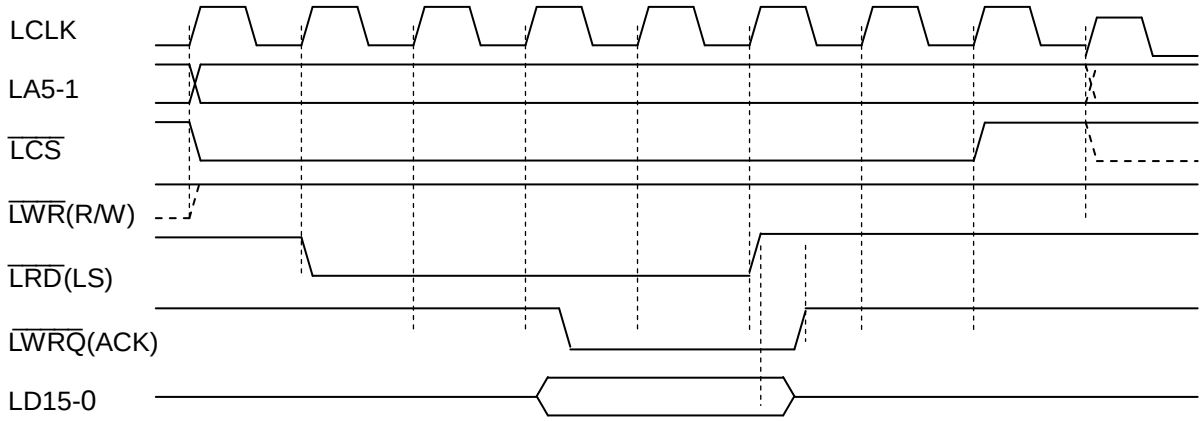


8-3-2. 16-bit CPU I/F (2) (68000 etc.)

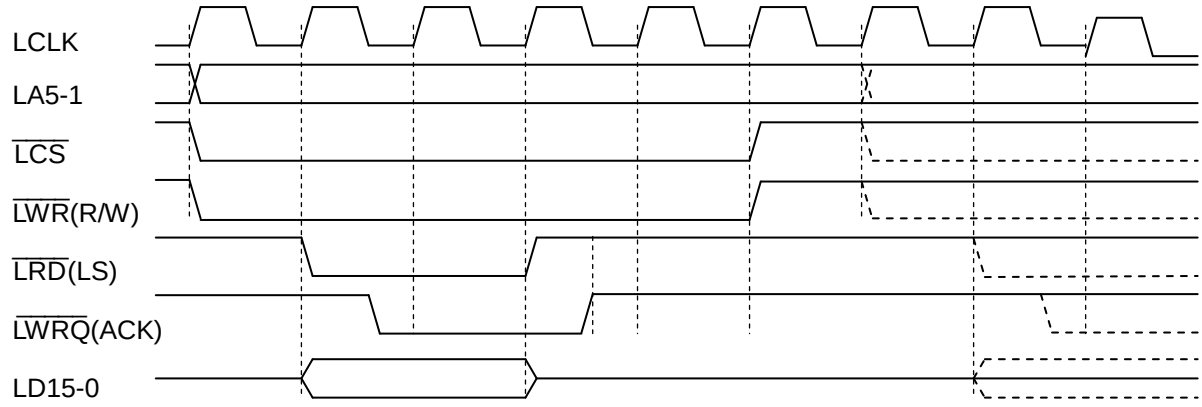
8-3-2-1. Read cycle (1) (without a wait cycle)



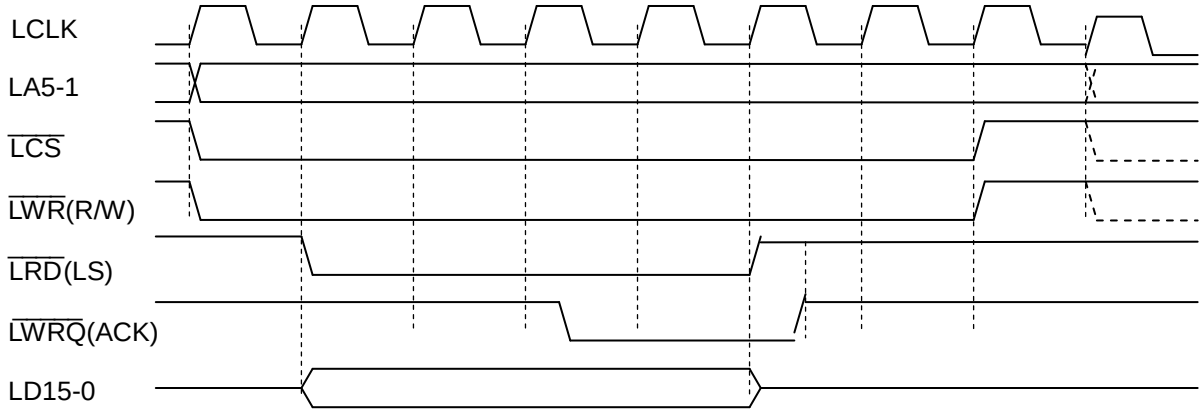
8-3-2-2. Read cycle (2) (wait 2 cycles)



8-3-2-3. Write cycle (1) (without a wait cycle)

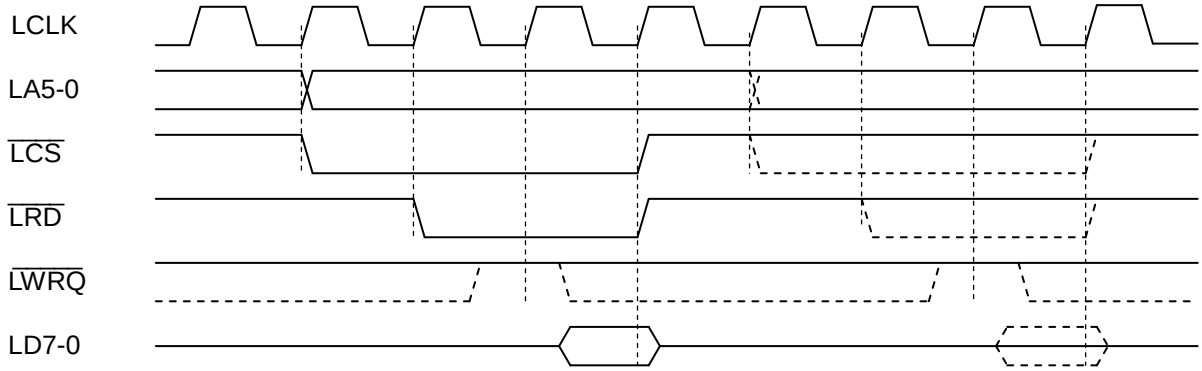


8-3-2-4. Write cycle (2) (wait 2 cycles)

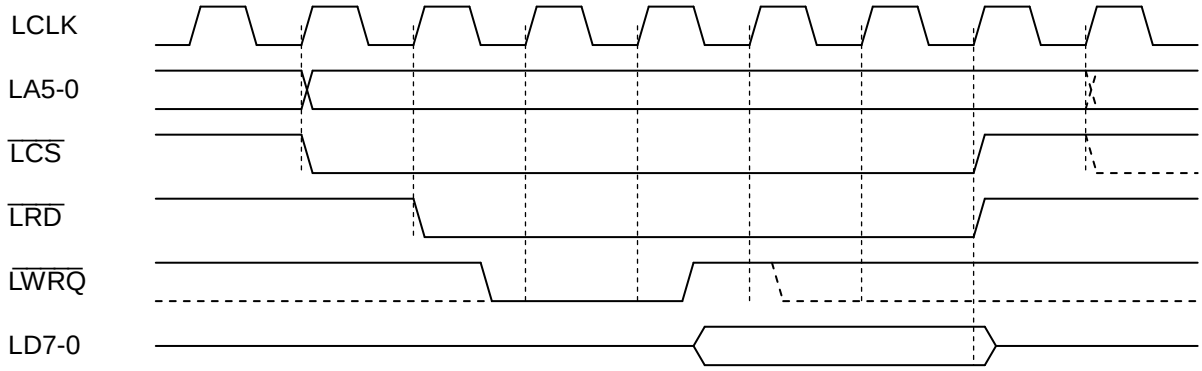


8-3-3. 8-bit CPU I/F (Z80 etc.)

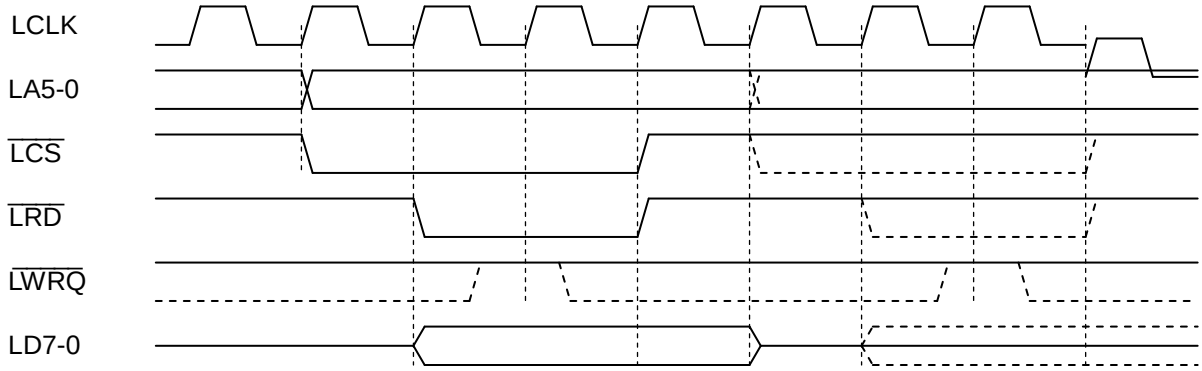
8-3-3-1. Read cycle (1) (without a wait 2 cycles)



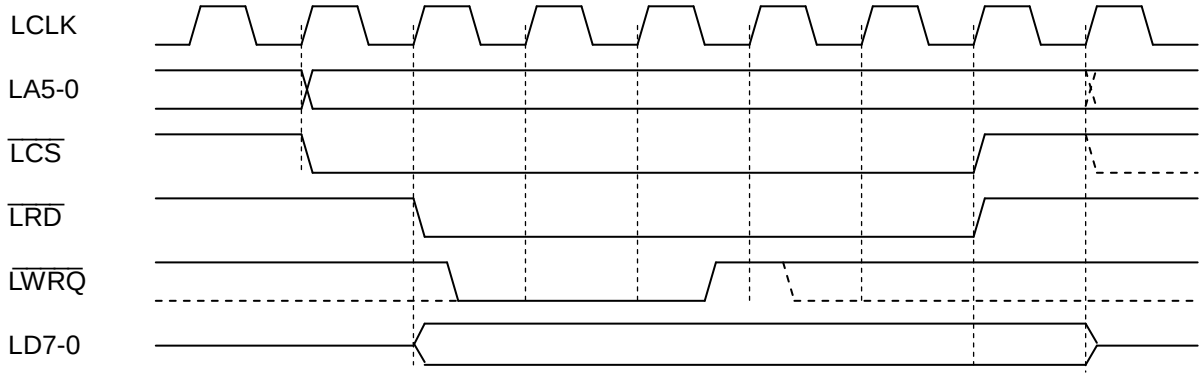
8-3-3-2. Read cycle (2) (wait 2 cycles)



8-3-3-3. Write cycle (1) (without a wait cycle)

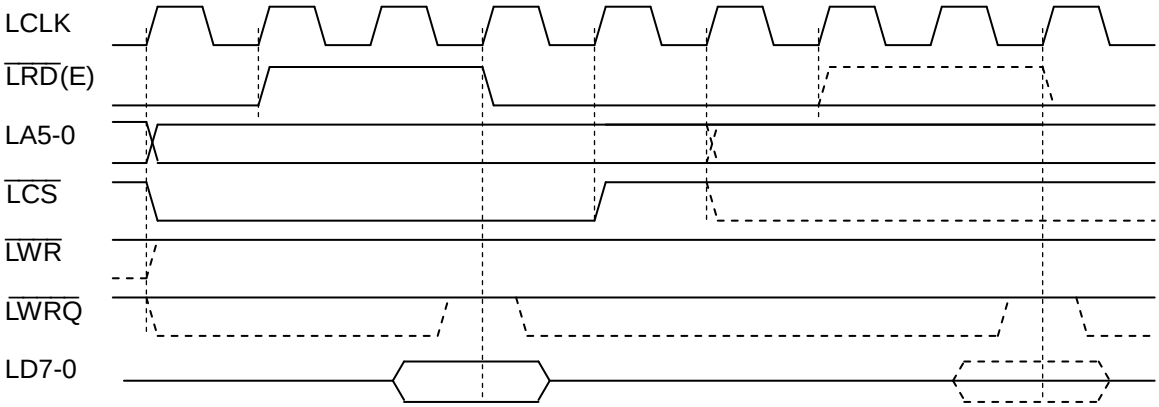


8-3-3-4. Write cycle (2) (wait 2 cycles)

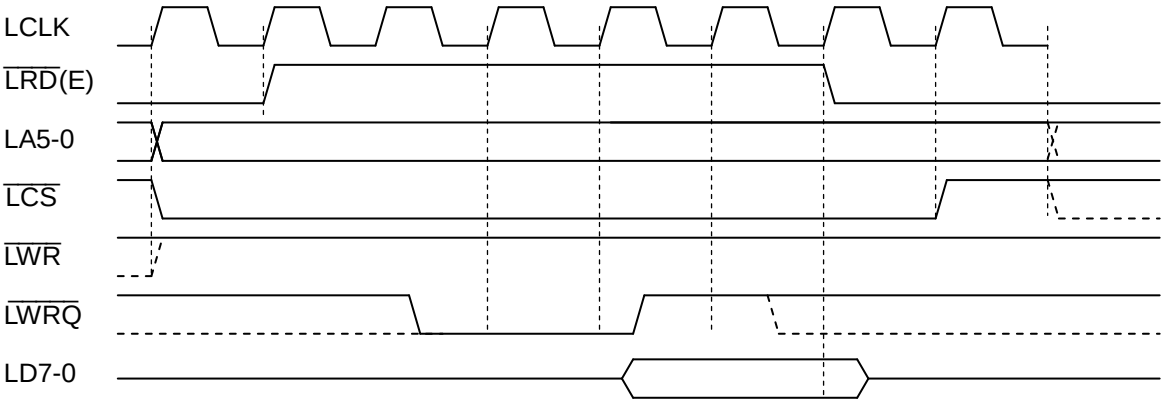


8-3-4. 8-bit CPU I/F (2) (6809 etc.)

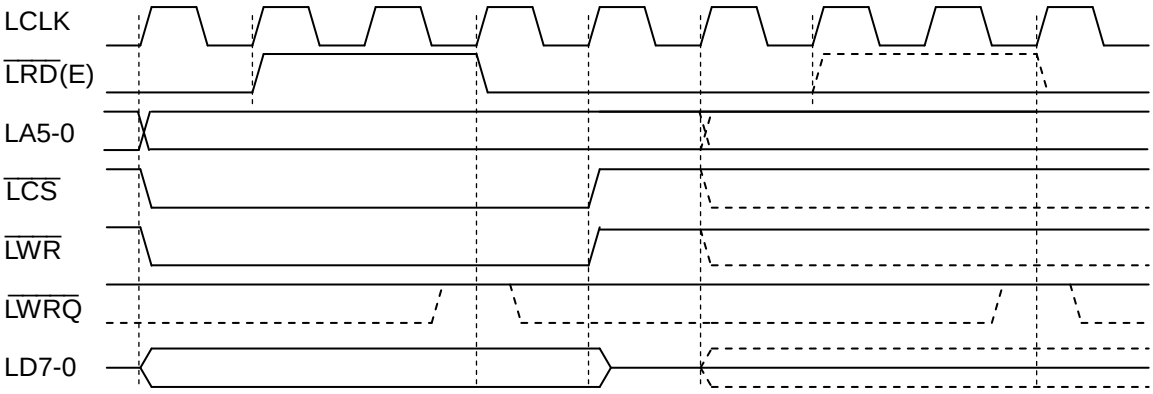
8-3-4-1. Read cycle (2) (without a wait cycle)



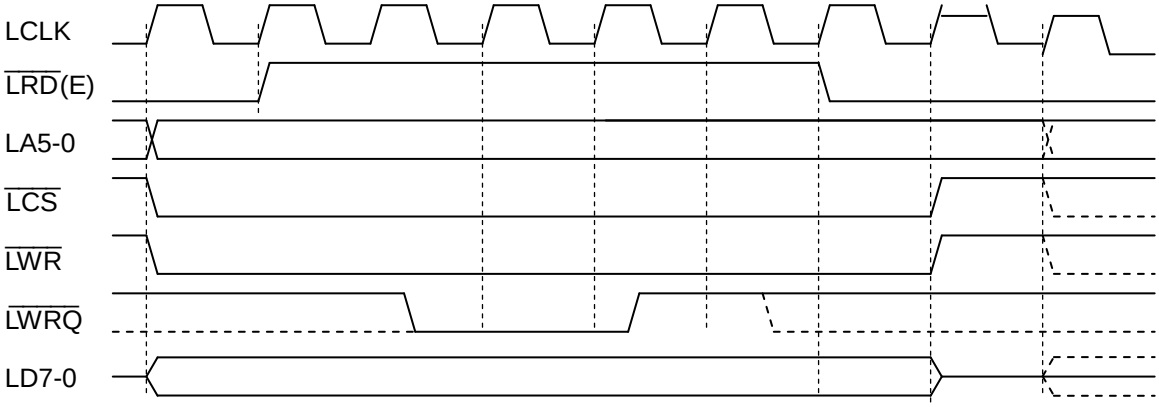
8-3-4-2. Read cycle (2) (wait 2 cycles)



8-3-4-3. Write cycle (1) (without a wait cycle)

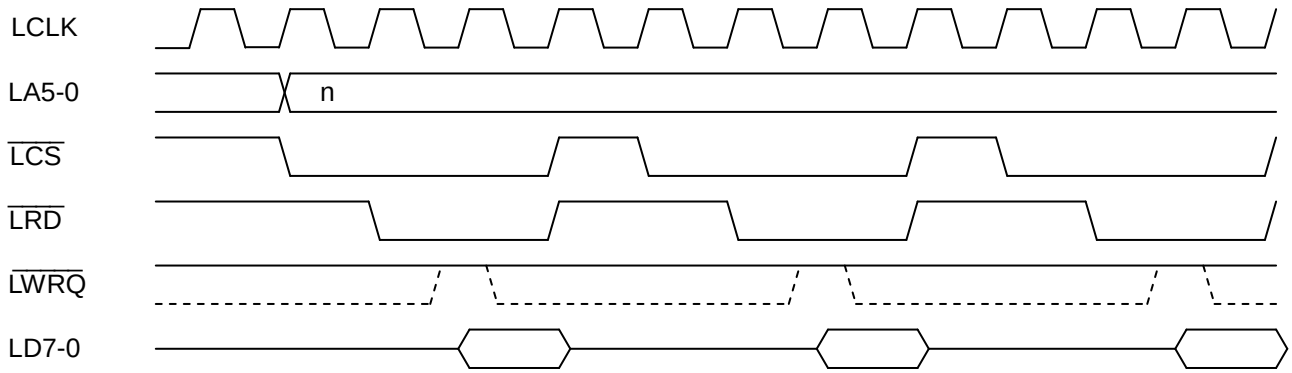


8-3-4-4. Write cycle (1) (wait 2 cycles)

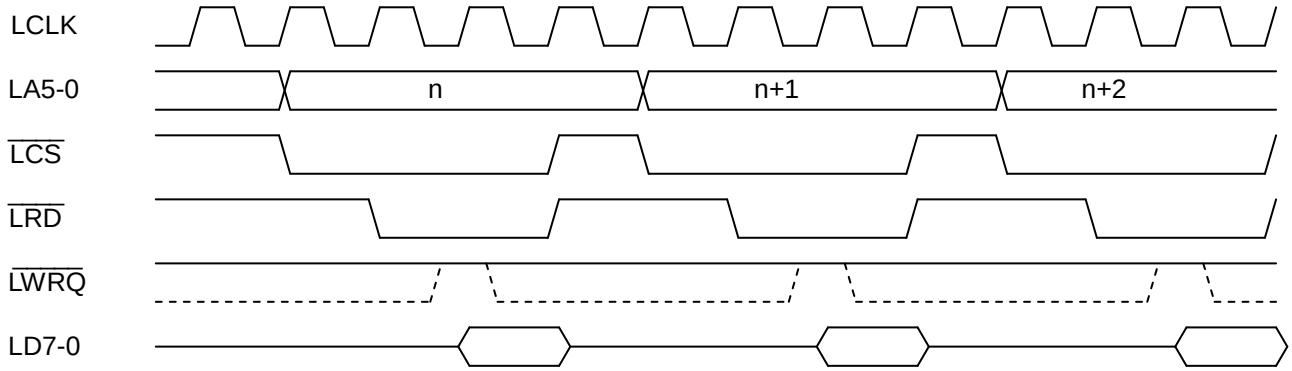


8-3-5. Example of a burst cycle (8-bit CPU-I/F (2) (Z80 etc.)

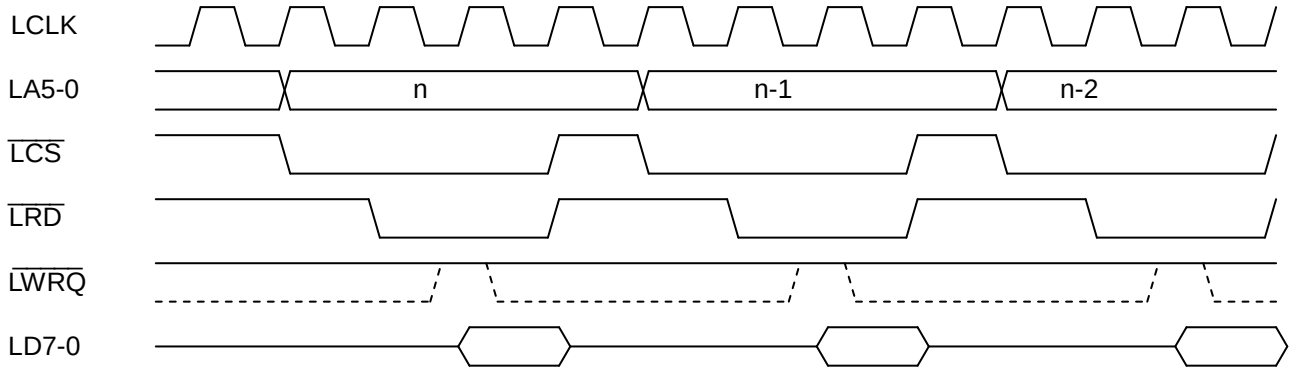
8-3-5-1. Burst read cycle (1) (Fixed address)



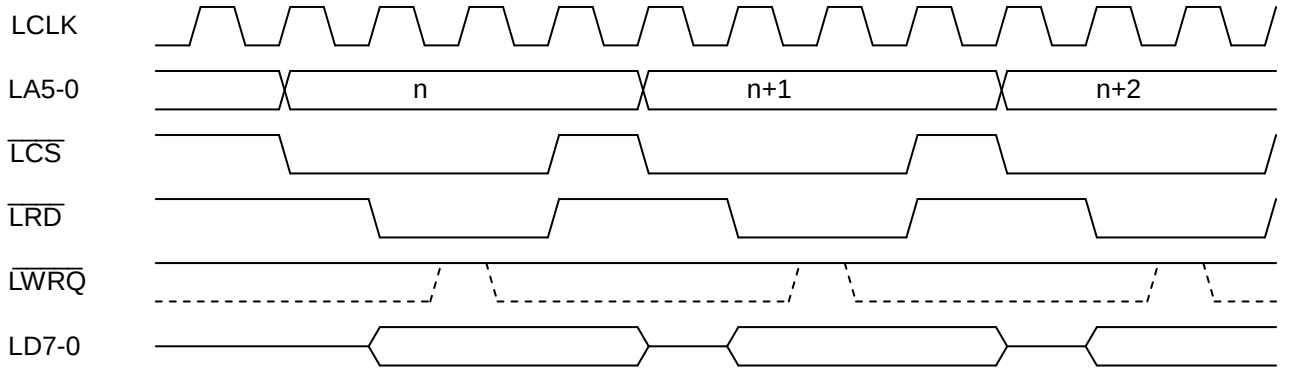
8-3-5-2. Bust read cycle (2) (add an address)



8-3-5-3. Burst read cycle (3) (subtract an address)



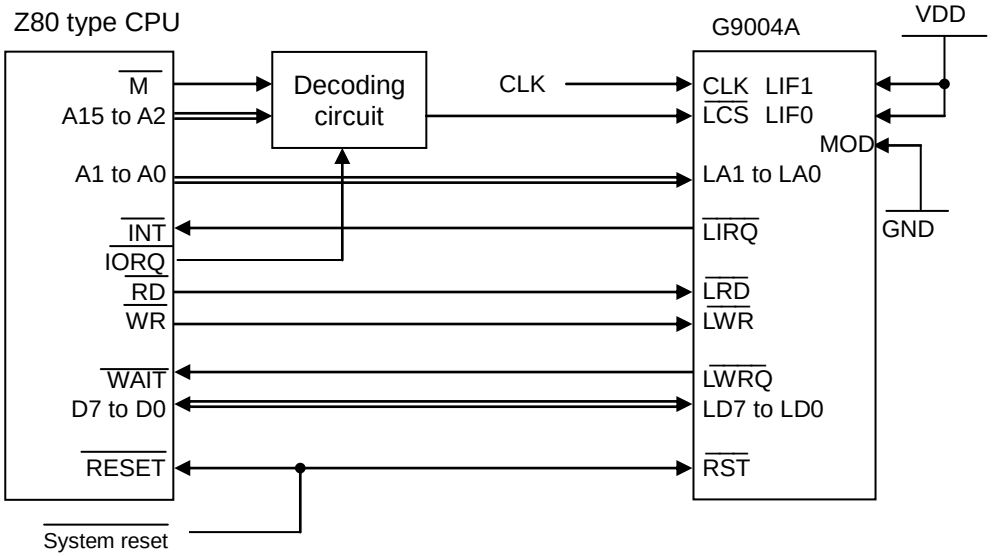
8-3-5-4. Burst write cycle (add an address)



9. Connection examples and recommended environment

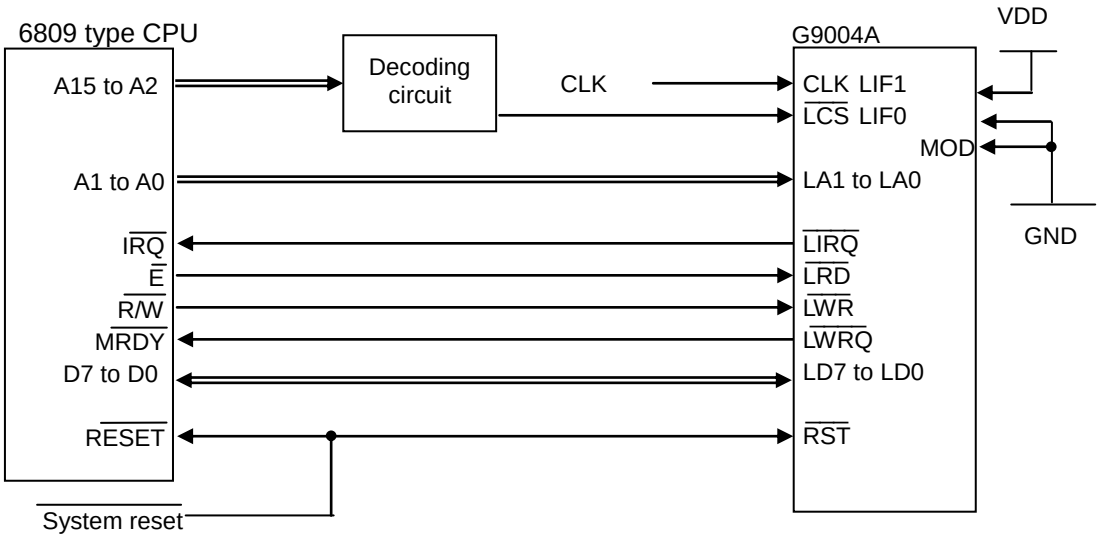
9-1. Example of a connection to a CPU using the CPU message communication mode (MOD = Low).

9-1-1. 8-bit I/F (1) (IF1 = High, IF0 = High)



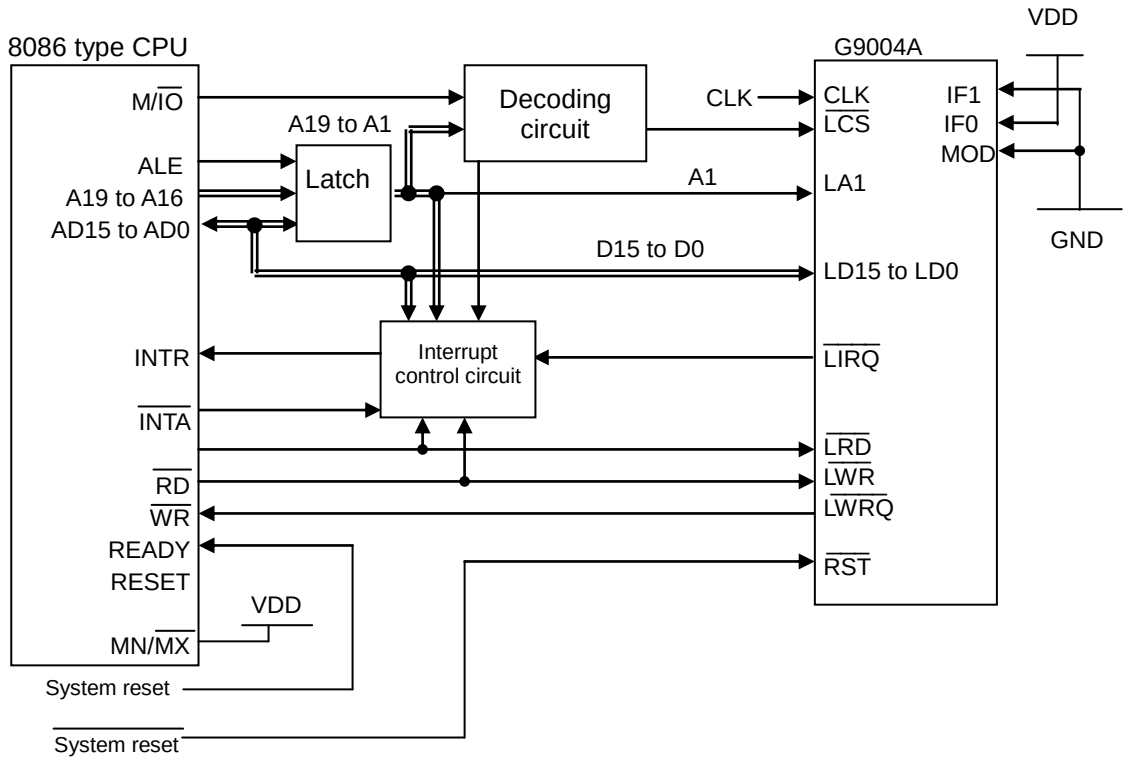
Note 1: When you use an interrupt controller, the CPU will output IORQ as an interrupt acknowledge signal that is used to determine the interrupt vector. At this time, when this LSI's $\overline{LCS}$ terminal goes Low, the LSI may output a $\overline{LWRQ}$ signal and still not be able to capture the vector properly. Therefore, arrange the decoding circuit so that it only functions when the M1 signal is High.
Note 2: Pull up terminals LD15 to LD8 to the power supply externally (5 to 10 k-ohms).
Note 3: Pull the LA5 to LA2 input terminals down to GND using external resistors (5 to 10 K-ohms).

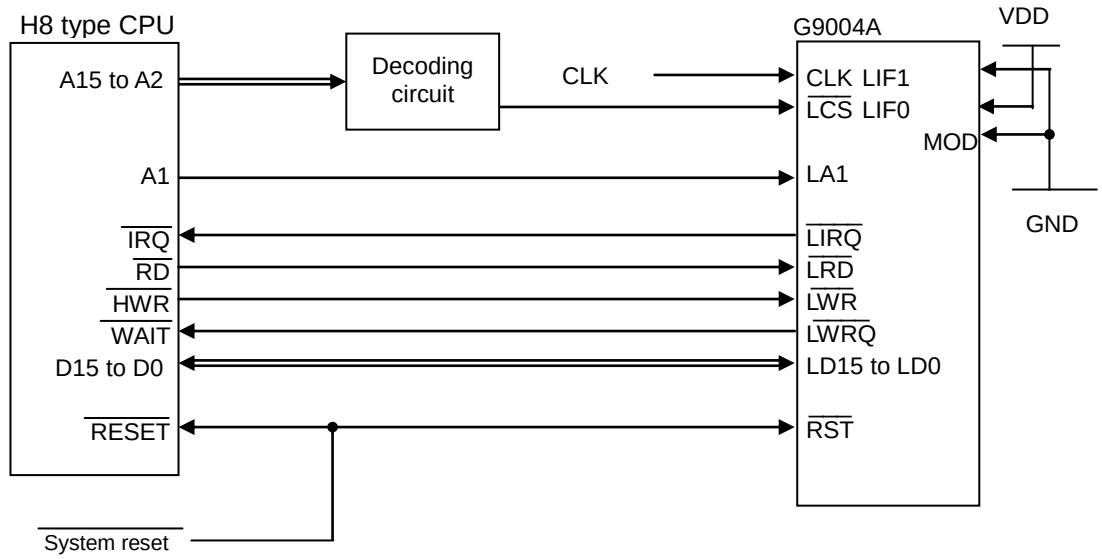
9-1-2. 8-bit I/F (2) (IF1 = High, IF0 = Low)



Note 1: Pull the LA5 to LA2 terminals down to the GND (5 to 10 K-ohms).
Note 2: Pull the LD15 to LD8 terminals up to the power supply using external resistors (5 to 10 K-ohms).

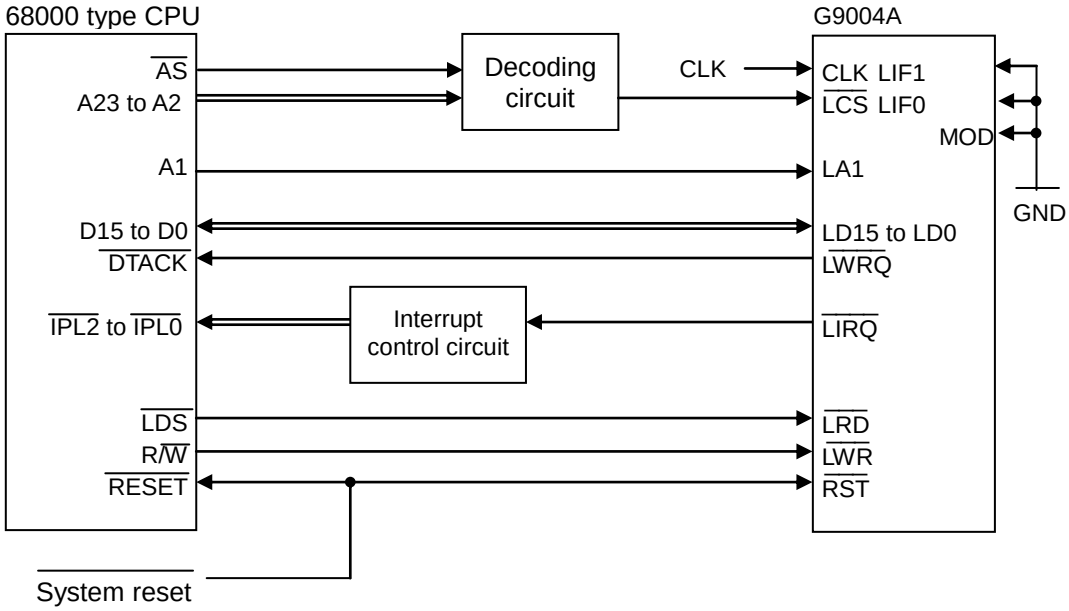
9-1-3. 16-bit I/F (1) (IF1=Low, IF0=High)





Note 1: Pull the LA5 to LA2, and LA0 terminals down to GND (5 to 10 K-ohms).

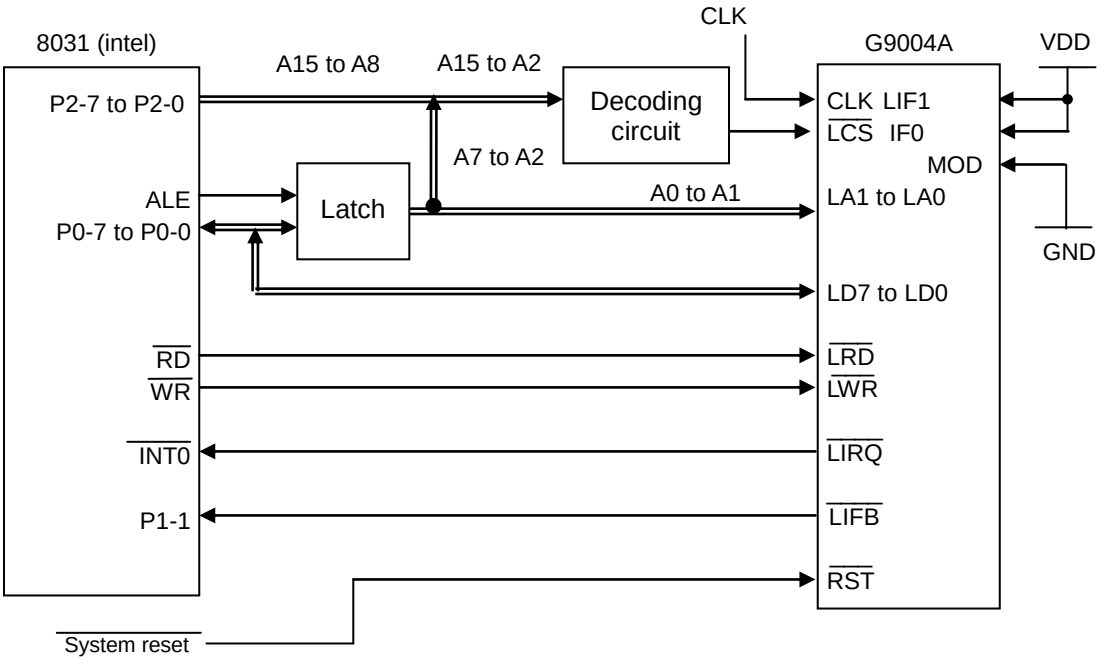
9-1-4. 16-bit I/F (2) (IF1 = Low, IF0 = Low)



Note 1: Pull the LA5 to LA2, and LA0 input terminals down to GND (5 to 10 K-ohms).

9-1-5. Connecting to a CPU without a wait function

The center device can be connected to a CPU that does not have a wait function.
Let's look at an example with the CPU interface using 8-bit I/F (1) while it is connected to an Intel 8031 8-bit CPU.
Since this CPU does not have a terminal for executing a wait function, care is needed when programming.



[Points]

1) Set IF1 = H and IF0 = H (8-bit I/F (1)).
2) Since the 8031 does not have a wait terminal, the WRQ terminal cannot be used.
   However, the G9004A needs a certain internal processing time to access (write/read) a CPU.
   And a wait function is therefore essential for continuous access operations.
   In the example above, the $\overline{\text{LIFB}}$ output terminal on the G9004A is connected to a port on the 8031.
   The $\overline{\text{LIFB}}$ bit is monitored using a routine in the 8031, so that the 8031 does not try to access the G9004A while it is processing a command.

Note 1: Pull the LA5 to LA2 terminals down to GND (5~10Kohms).

9-2. Access timing when the CPU message communication mode is selected (MOD = Low)

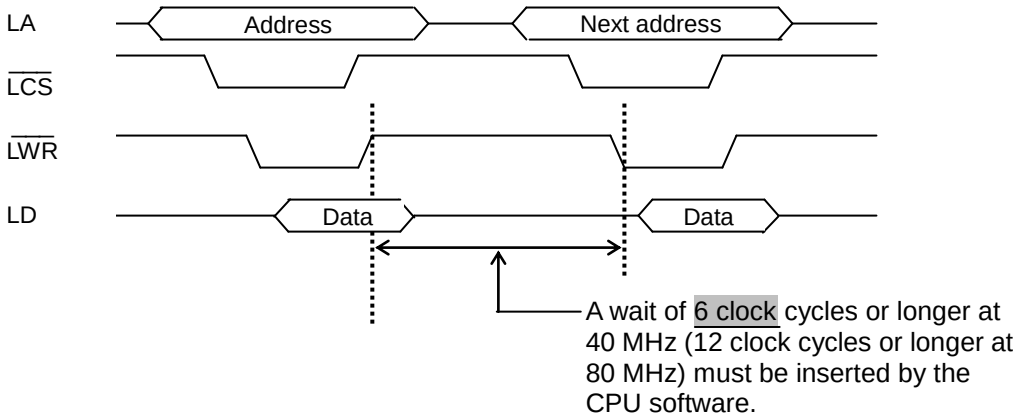
9-2-1. Normal access

CPUs that have a wait function can be connected to the $\overline{\text{LWRQ}}$ terminal on the G9004A so that they can be used without special concern for signal timing.
However, CPUs without a wait function must monitor the $\overline{\text{LIFB}}$ output or use one of the following timing schemes (this is essential).

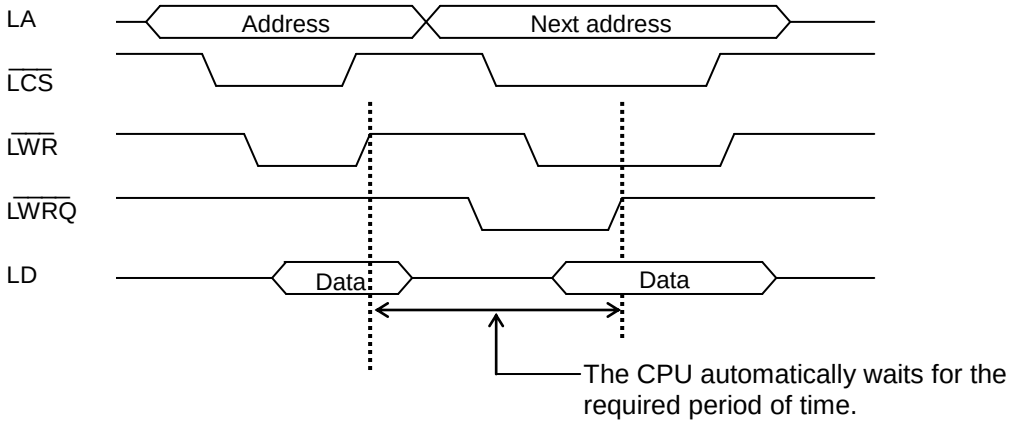
9-2-2. Write to command or data transfer FIFO

The timing for writing to command (address 0 in the 8-bit I/O mode (1)) or the data transfer FIFO (address 2 in the 8-bit I/O mode (1)) is shown below.
A wait time is necessary to perform continuous writing. The wait must be 6 clock cycles or longer at 40 MHz (12 clock cycles or longer at 80MHz).

1) Does not use the $\overline{\text{LWRQ}}$ output (CPU does not have a wait function)



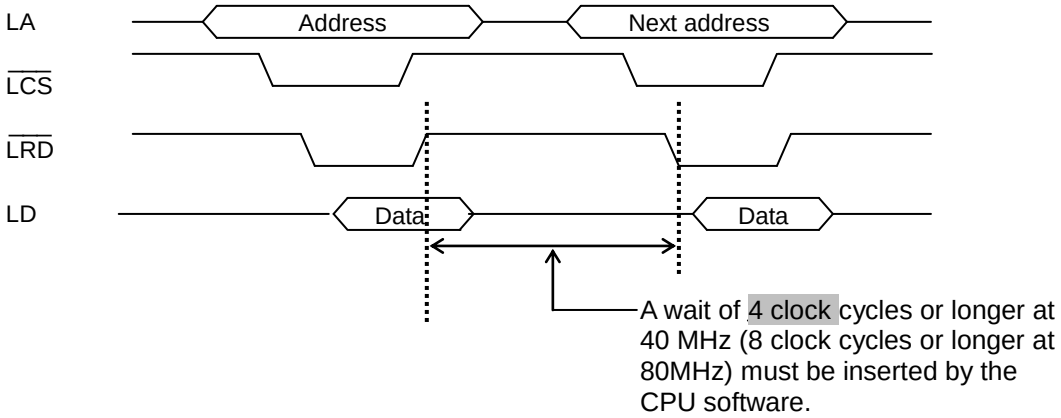
2) Uses the $\overline{\text{LWRQ}}$ output (CPU has a wait function)



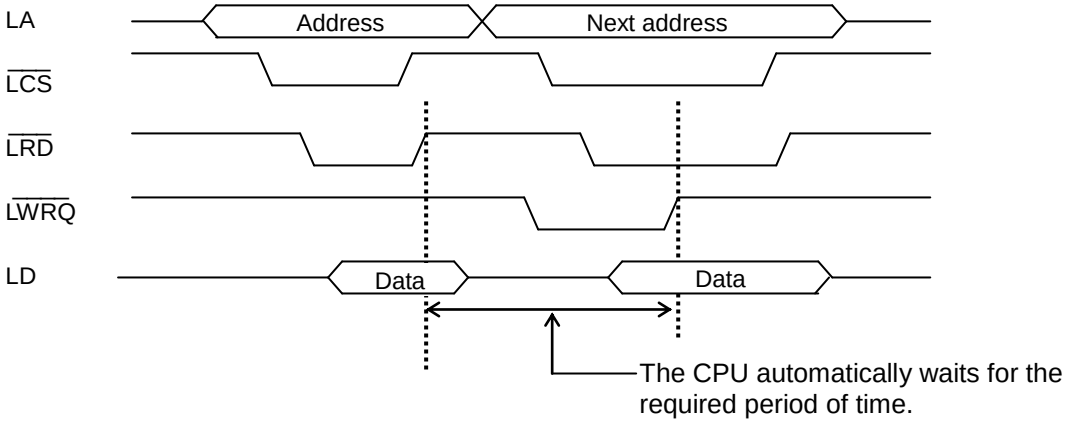
9-2-3. Read status

Shown below is the timing when reading from address 0 in the 8-bit I/O mode (1).
A wait time is necessary to perform continuous writing. The wait must be 4 clock cycles or longer at 40 MHz (8 clock cycles or longer at 80 MHz).

1) Does not use the $\overline{\text{LWRQ}}$ output (CPU does not have a wait function)



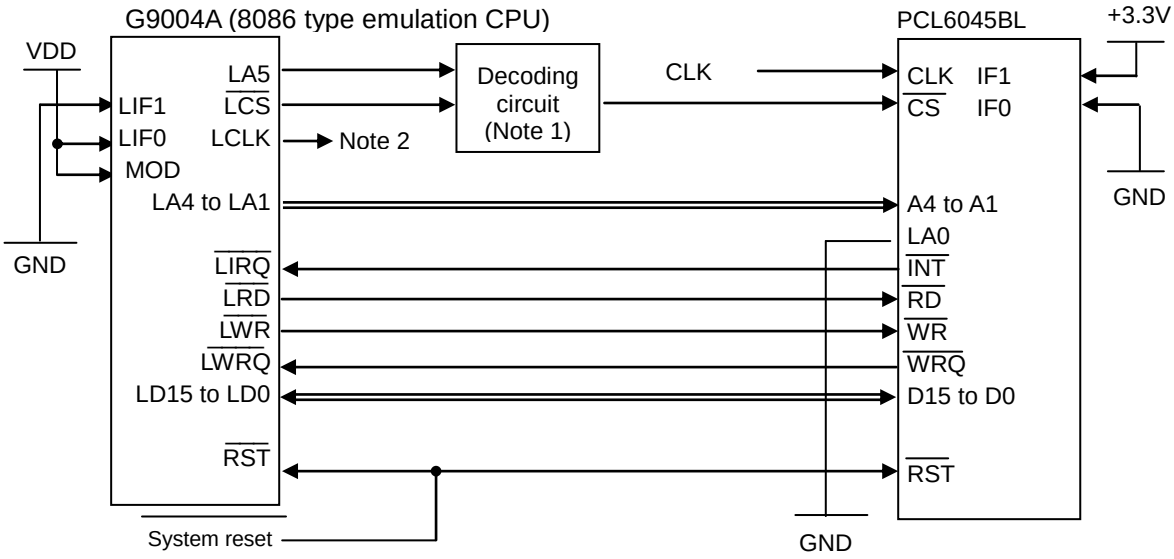
2) Uses the $\overline{\text{LWRQ}}$ output (CPU has a wait function)



9-3. Connection to peripheral LSIs when the CPU emulation mode is selected

9-3-1. Connections to a PCL6045BL (8086 type CPU emulation)

The PCL6045BL is a pulse control LSI for NPM's motion control. One PCL6045BL can generate pulse trains for four axes.

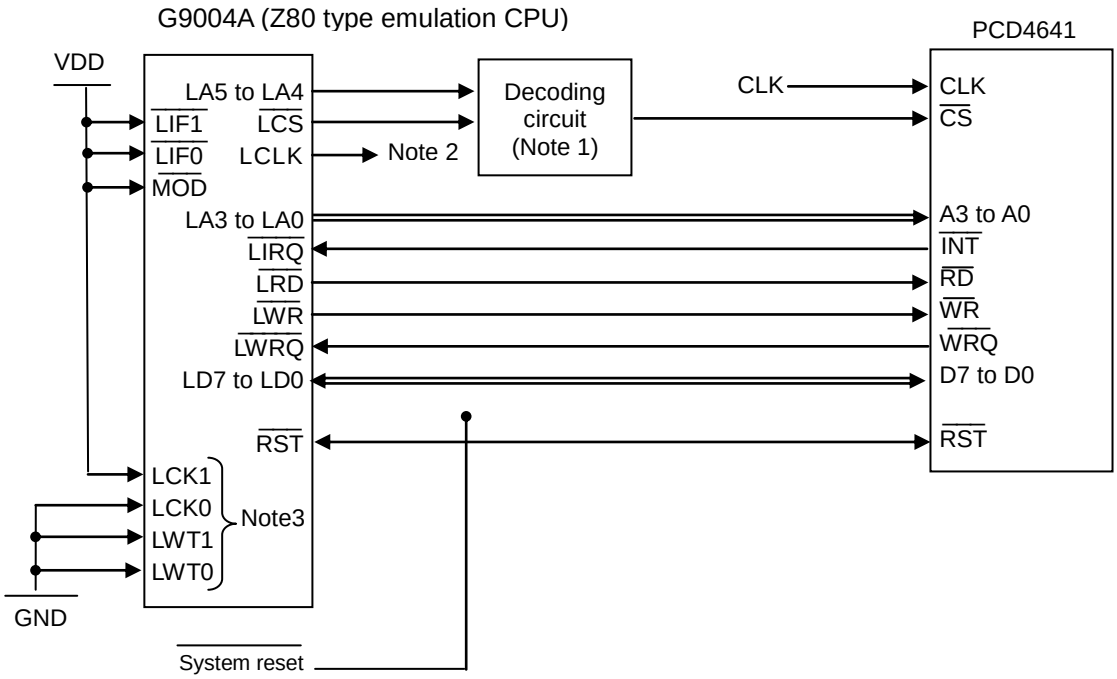


Note 1: The G9004A can be connected to two PCL6045BLs.
In this case, the "LA5" signal is used to tell the two chips apart. This line is the equivalent of a decoder circuit.
When only one device is connected to the G9004A, there is no need to provide a decoder circuit. The $\overline{\text{LCS}}$ output terminal on the G9004A can be connected to the terminal on the PCL6045BL.

Note 2: There are only 4 types of the "LCLK" clock output on CPU emulation device (G9004A): 2MHz, 4MHz, 10MHz and 20MHz. The PCL6045BL basically needs a 19.6608 MHz clock, therefore, if it is connected to some other clock speed, such as a 20MHz clock, calculation such as multiplication setting becomes complex. For details, see the user's manual for the PCL6045BL.

9-3-2. Connections to the PCD4641 (Z80 type CPU emulation)

The PCD4641 is an NPM pulse control LSI for use with stepper motors.



Note 1: When only one device is connected to the G9004A, there is no need for a decoder circuit. The $\overline{LCS}$ output terminal on the G9004A can be connected to the $\overline{CS}$ terminal on the PCD4641.

Note 2: There are only 4 types of the "LCLK" clock output from the CPU emulation device (G9004A): 2MHz, 4MHz, 10MHz and 20MHz. The PCD4641 basically needs a 4.9152 MHz clock, therefore, if ”LCLK” is input, calculation such as multiplication setting become complex. For details, see the user's manual for the PCD4641.

Note 3: LCK1 = High, LCK0 = Low
LWT1 = Low, LWT0 = Low
When these settings are used, the device will have the status shown below:

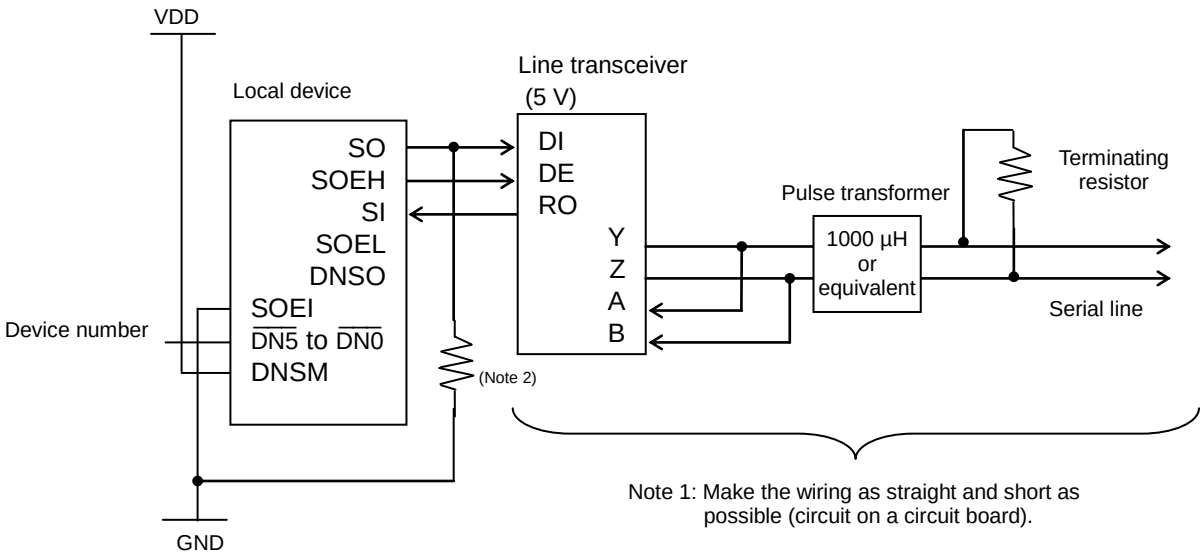
- LCLK = 10 MHz
- Local bus access interval = T_{LCLK} (100 ns)

The “local bus access interval" is waiting time for next access when the CPU emulation device is reading from or writing to the PCD4641. In the case that an LSI that does not have a terminal for WAIT CPU is controlled, you can set optimal access timing by adjusting interval time.

9-4. Connections to a serial communication line

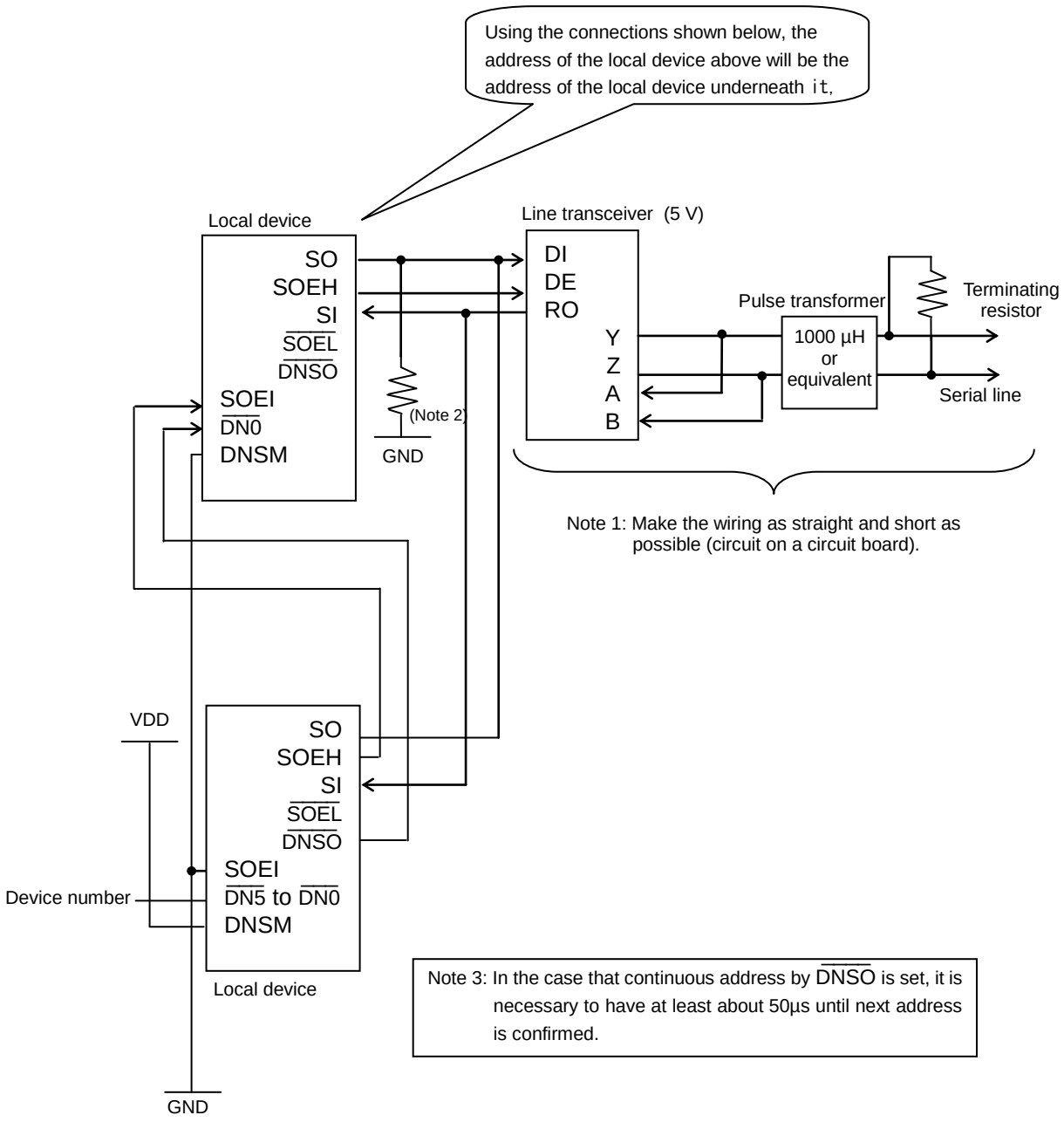
Use RS-485 line transceivers and pulse transformers (1000 µH or equivalent) to make serial communication connections.
Connect the line transceivers as shown below.
Connect terminating resistors (which match the cable impedance) at both ends of the transmission line.
The terminating resistors can be either before or after the pulse transformer. The same effect will be obtained at either position.
For a line transceiver, we recommend that a 5V line transceiver is used. When using a 5 V line transceiver, 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

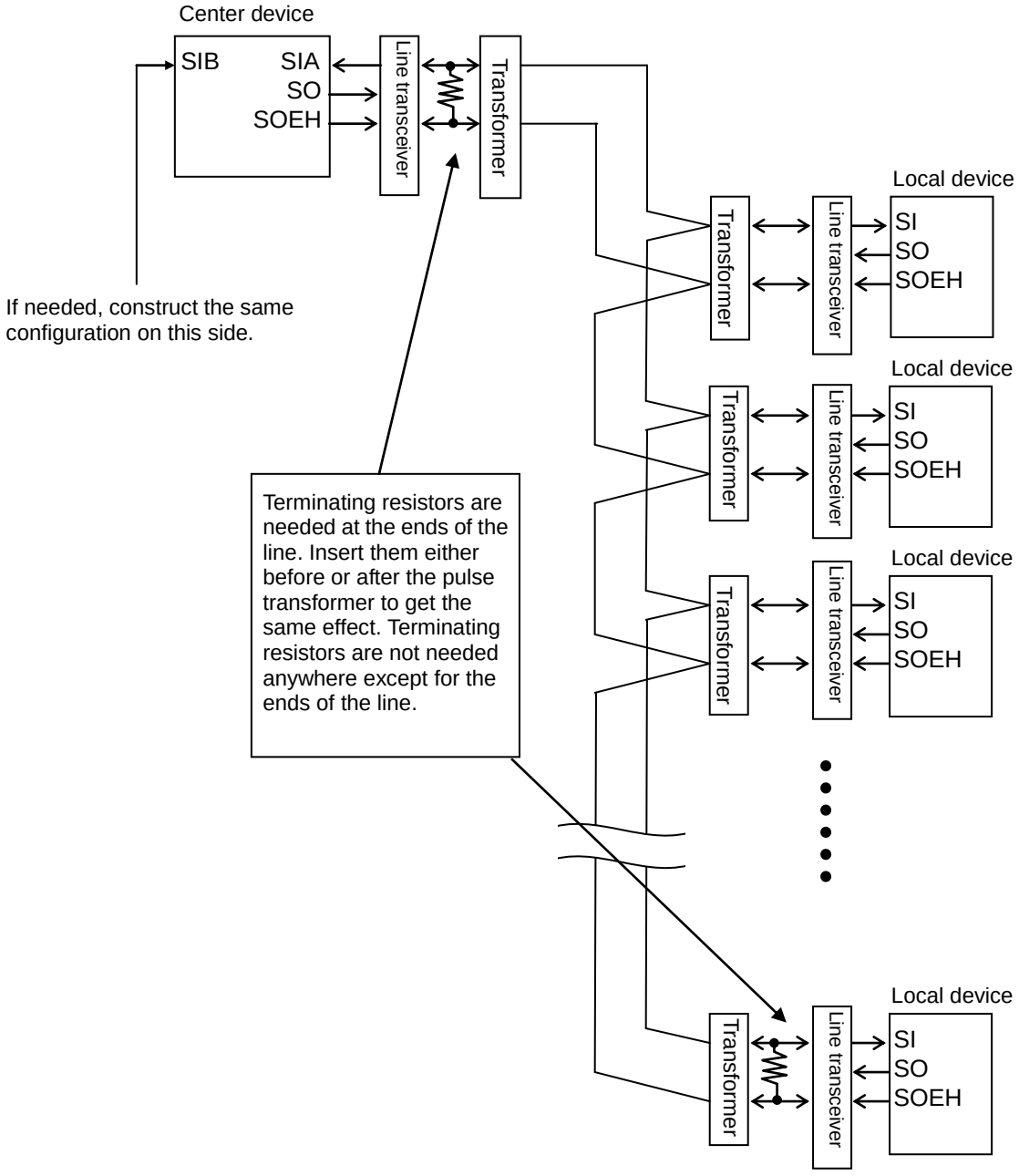


Note 1: When connecting the serial lines to line transceivers, make the path as short and straight as possible.
Running these lines on a PC board could deteriorate the communication performance.

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

9-5. Complete configuration

We recommend a configuration with the center device at one end of the line and the local devices at 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.



9-6. 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.

9-6-1. 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 pair 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.

9-6-2. 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.

9-6-3. 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.

9-6-4. I/F chip

We selected I/C chips with specifications better than the RS485 standard.
For RS485 line transceiver, we strongly recommend that 5V line transceivers are used. In the experiment, we used 5 V line transceivers. When 5 V line transceivers are used, inserting an IC such as level shifters to a signal line does not be needed.

9-6-5. 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	JPC Co., Ltd	NPT102F
Line transceiver	TEXAS INSTRUMENTS	SN65HVD1176

9-6-6. 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.

- Estimating cable length in the system design phase
  In the first estimate, use shorter line lengths. In the actual system configuration, lines may be lengthened.
  Estimates made using the maximum length may lead to impossible communication distances.

- Minimum cable length
  Each cable must be at least 60 cm long. Although this may seem contradictory to the excess cable precaution, this minimum length is necessary.

- Using different cables in one system
  Do not mix cables from different manufacturers, even when they are in the same category. (Different cable models from the same manufacturer should not be used either.)
  Using different cables together may deteriorate the communication quality.

10. Center device (G9001A)

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

1) Write command to the center device (16 bits)

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

2) Write command to the center device (8 bits)

Outp (Address, Data)	
Address	Value corresponding to the address map in the center device (16 bits).
Data	Data to write (8 bits)
Return value	None

3) Read command from the center device (16 bits)

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

4) Read command from the center device (8 bits)

Inp (Address)	
Address	Value corresponding to the address map in the center device (16 bits).
Return value	Read data (8 bits)

Also, see the individual items in the "Message communication procedure" and "CPU emulation procedure" sections.

10-1. Program example of the CPU emulation mode

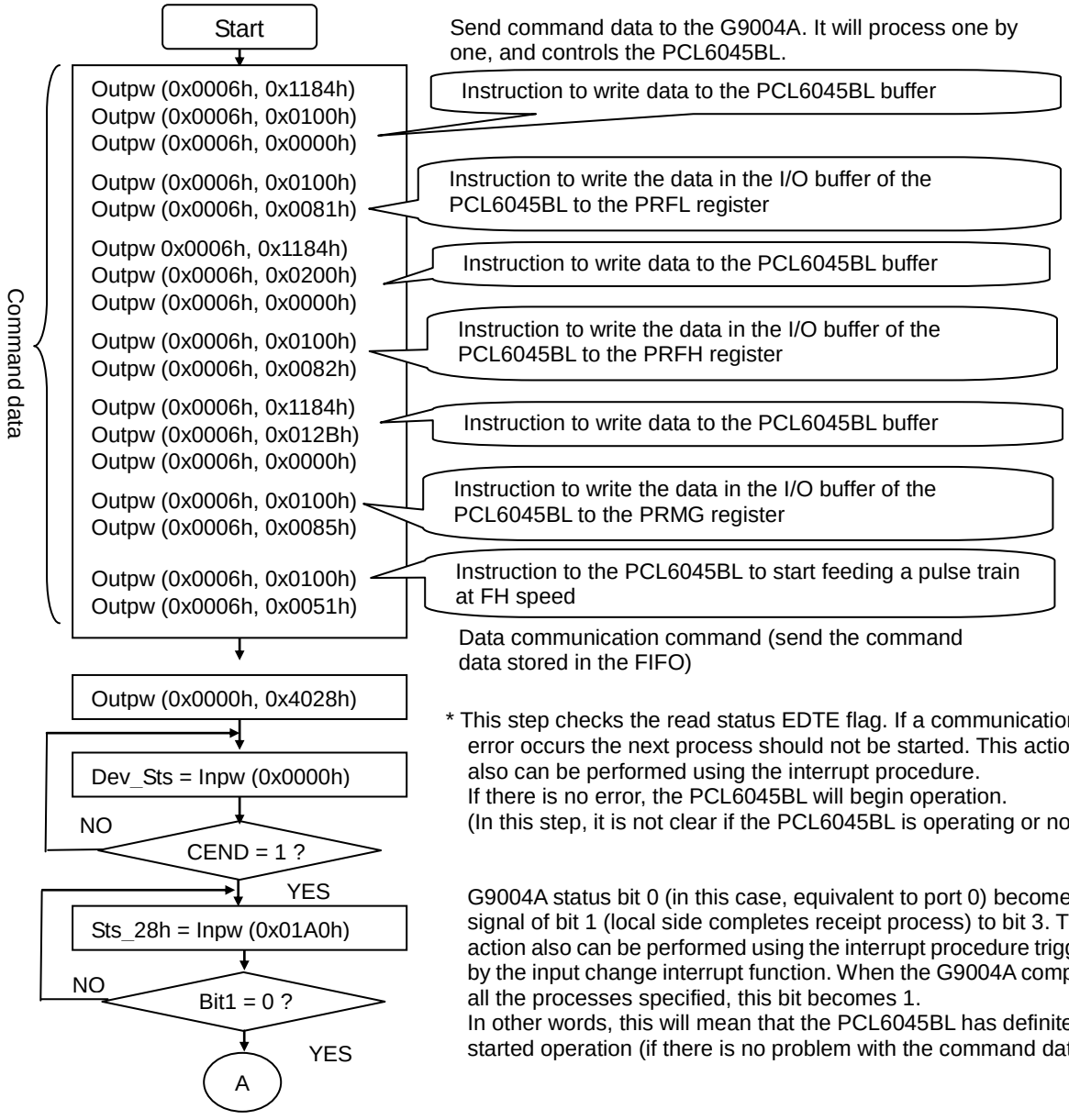
Using the CPU interface mode of the G9004A, the PCL6045BL uses a 16-bit I/F (1), and the PCD4641 uses an 8-bit I/F (1).

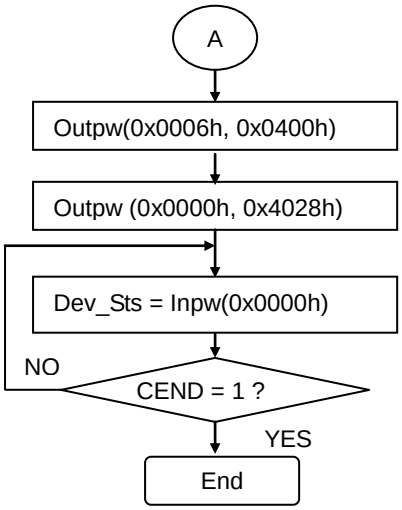
10-1-1. Control example of the PCL6045BL

Shown below is a program example that lets the center device (G9001A) control a PCL6045BL through a G9004A.
The PCL6045BL is a pulse control LSI for NPM's motion control. Set the CPU-I/F to 8086 mode (IF0 = L, IF1 = H).
One PCL6045BL can generate pulse trains for four axes.
In the example below, use the device address of "28h" for the G9004A.

Registers to set in the PCL6045BL

Register name	Set value	Remark
PRFL	00000100h	
PRFH	00000200h	
PRMG	012Bh	Multiplication rate = 1





Put a reset instruction command for the " local receive processing complete " flag in the FIFO.

Write a data communication command (send the command data written in the FIFO)

When CEND = H, the local device flag has been reset.

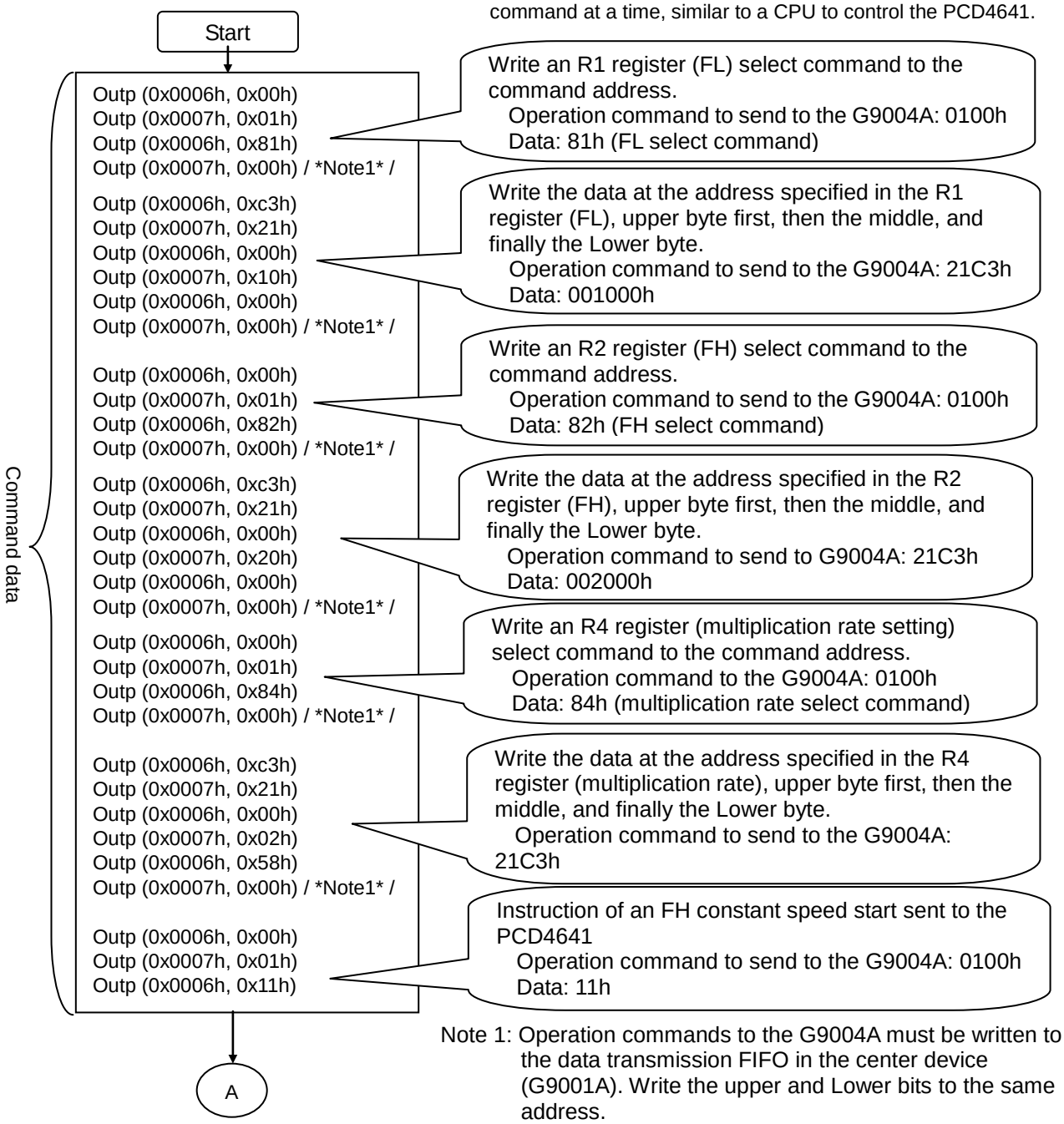
10-1-2. Control example of PCD4641

Below shows a program example that lets a center device (G9001A) control a PCD4641 through a G9004A. In this example, the center device (G9001A) communicates in CPU-I/F model 4 (Z80 type 8-bit CPU). The PCD4641 is an NPM's pulse control LSI for stepper motors. In the example below, use a device address of "28h" for the G9004A.

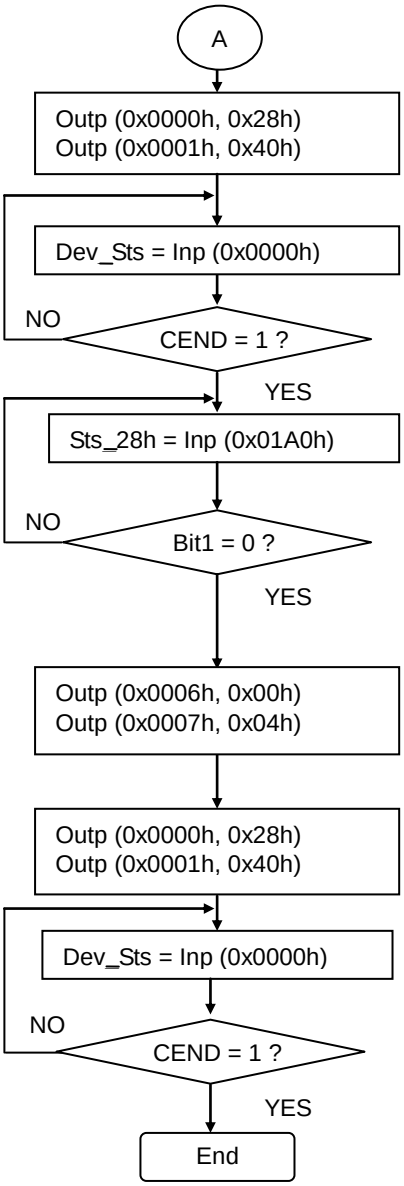
Registers to set in the PCD4641

Register name	Set value	Remark
PRFL	001000h	
PRFH	002000h	
PRMG	000258h	Multiplication rate = 1

Send command data to the G9004A and it will process it one command at a time, similar to a CPU to control the PCD4641.



Note 1: Operation commands to the G9004A must be written to the data transmission FIFO in the center device (G9001A). Write the upper and Lower bits to the same address.
If only 8 bits are needed, you will still have to write dummy data to upper 8 bits, and sent an operation command to the G9004A using the same address.



Data communication command (send the command data stored in the FIFO)
Write in order of Lower to upper bit.

* This step checks the read status EDTE flag. If a communication error occurs the next process should not be started. This action also can be performed using the interrupt procedure.
If there is no error, the PCD4641 will begin operation.
(In this step, it is not clear if the PCD4641 is operating or not.)

- G9004A status bit 1 (in this case, equivalent to port 0) indicates whether reception by the local devices is complete. This action also can be performed using the interrupt procedure triggered by the input change interrupt function. When the G9004A completes all the processes specified, this bit becomes 1.
In other words, this will mean that the PCD4641 has definitely started operation (if there is no problem with the command data).

Put a reset instruction command for the " local receive processing complete " flag in the FIFO.

Write a data communication command (send the command data stored in the FIFO)

When CEND = H, the local device flag has been reset.

11. Electrical Characteristics

11-1. Absolute maximum ratings

Item	Symbol	Rating	Unit
Power supply voltage	V_{DD}	V_{SS} -0.3 to +4.0	V
Input voltage	V_{IN}	V_{SS} -0.3 to V_{DD} +0.5	V
Input voltage (5V-I/F)	V_{IN}	V_{SS} -0.3 to +7.0	V
Output voltage (5V-I/F)	V_{OUT}	V_{SS} -0.3 to +7.0	V
Output current / Terminal	I_{OUT}	±30	mA
1Storage temperature	T_{STO}	-65 to +150	^{o}C

11-2. Recommended operating conditions

Item	Symbol	Rating	Unit
Power supply voltage	V_{DD}	+3.0 to +3.6	V
Input voltage	V_{IN}	-0.3 to V_{DD}+0.3	V
Input voltage (5V-I/F)	V_{IN}	-0.3 to +5.8 Note 1	V
Storage temperature	T_a	-40 to +85	^{o}C

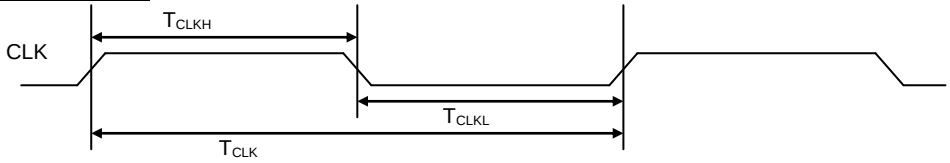
Note 1. Do not add more than output voltage from outside when outputting "H" level.

11-3. DC characteristics

Item	Symbol	Condition	Min.	Max.	Unit
Current consumption	I_{dd}	CLK = 80 MHz, after reset		34	mA
Output leakage current	I_{OZ}		-1	1	µA
Input capacitance				10	pF
Low input current	I_{IL}	$\overline{DN0}$ to $\overline{DN5}$, DNSM, SPD0 to 1, TUD, TMD, LCK0 to 1, LIFO0 to 1, LWT0 to 1, CKSL, MOD	-165		µA
		Input terminals and input/output terminals other than the above.	-1		µA
High input voltage	I_{IH}	SOEI, SI, BRK		190	µA
		Input terminals and input/output terminals other than the above.		1	µA
Low input voltage	V_{IL}	V_{DD}=MAX, CLK, RST	-0.3	0.8	V
		V_{DD}=MAX, input and input /output terminals other than the above		0.6	V
High input current	V_{IH}	V_{DD}=MIN, CLK,RST	2.0	V_{DD}+0.3	V
		V_{DD}=MIN, input and input/output terminals other than the above	2.4		V
Hysteresis voltage	V_H	V_{DD}=MIN	0.1		V
Low output voltage	V_{OL}	I_{OL} = 6 mA		Vss+0.4	V
High output voltage	V_{OH}	I_{OH} = -6 mA	V_{DD}-0.4		V
Low output current	I_{OL}	V_{OL} = 0.4 V		6	mA
High output current	I_{OH}	V_{OH} = V_{DD} -0.4 V	-6		mA
Internal pull up, pull down resistance	R_{Pud}		20	120	K-ohm

11-4. AC characteristics

11-4-1. System clock



1) When setting CKSL = L and data transfer rate = 20 Mbps

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

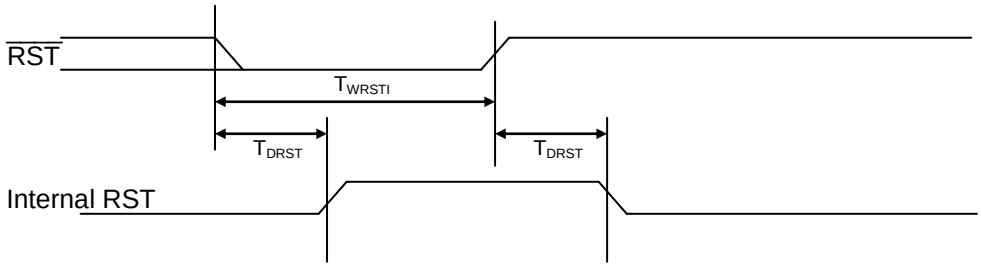
2) When setting CKSL = L and data transfer rate = 10 Mbps

Item	Symbol	Min.	Max.	Unit
Frequency	f_{CLK}	-	40	MHz
Cycle	T_{CLK}	25		ns
High duration	T_{CLKH}	-	-	ns
Low duration	T_{CLKL}	-	-	ns

3) 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

11-4-2. Reset timing



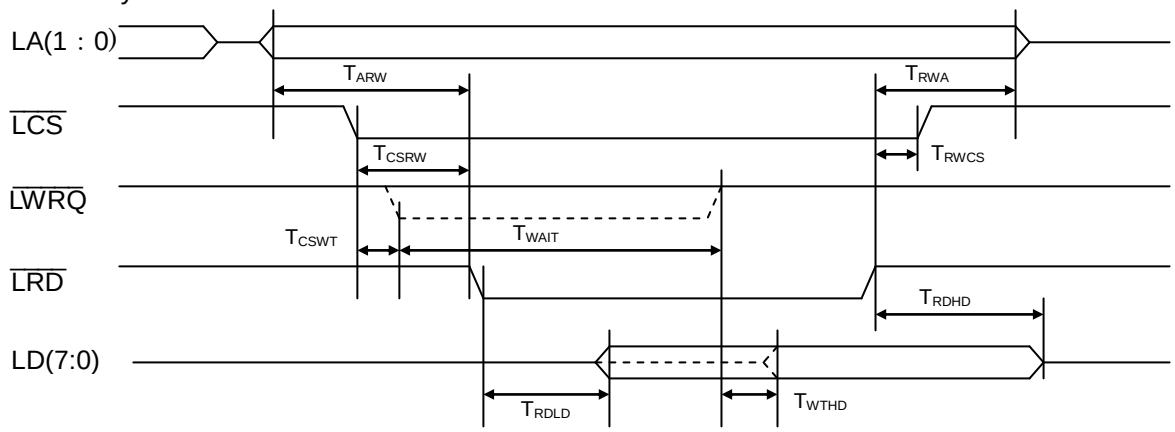
Item	Symbol	Min.	Max.	Unit
Reset length	T_{WRSTI}	10	-	Clock cycles
Delay time	T_{DRST}	-	10	Clock cycles

Note 1: The 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.

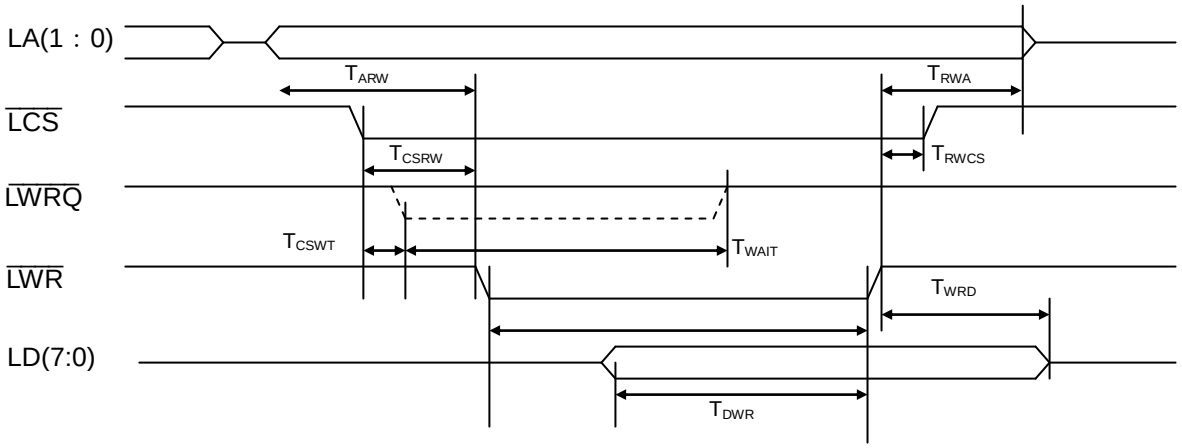
11-5. Timing of CPU message communication mode

11-5-1. 8-bit I/F (1) (IF1 = High, IF0 = High)

- Read cycle



- Write cycle

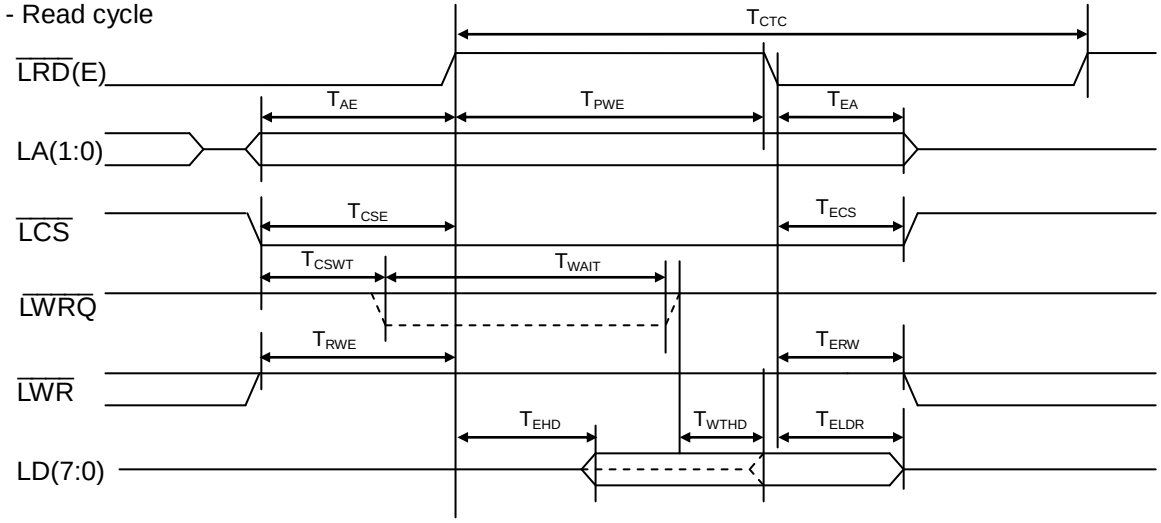


Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{ARW}		17		ns
Address hold time for $\overline{LRD}$, $\overline{LWR}$ ↑	T_{RWA}		0		ns
$\overline{LCS}$ setup time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{CSRW}		5		ns
$\overline{LCS}$ hold time for $\overline{LRD}$, $\overline{LWR}$ ↑	T_{RWCS}		0		ns
$\overline{LWRQ}$=ON delay time for $\overline{LCS}$ ↓	T_{CSWT}	C_L = 40pF		12	ns
$\overline{LWRQ}$ signal Low time	T_{WAIT}	Note 1		$6T_{CLK}$ +11	ns
Data output delay time for $\overline{LRD}$ ↓	T_{RDLD}	C_L = 40pF		29	ns
Data output delay time for $\overline{LWRQ}$ ↑	T_{WTHD}	C_L = 40pF		16	ns
Data float delay time for $\overline{LRD}$ ↑	T_{RDHD}	C_L = 40pF		30	ns
$\overline{LWR}$ signal width	T_{WR}	Note 2	12		ns
Data setup time for $\overline{LWR}$ ↑	T_{DWR}		22		ns
Data hold time for $\overline{LRD}$ ↑	T_{WRD}		0		ns

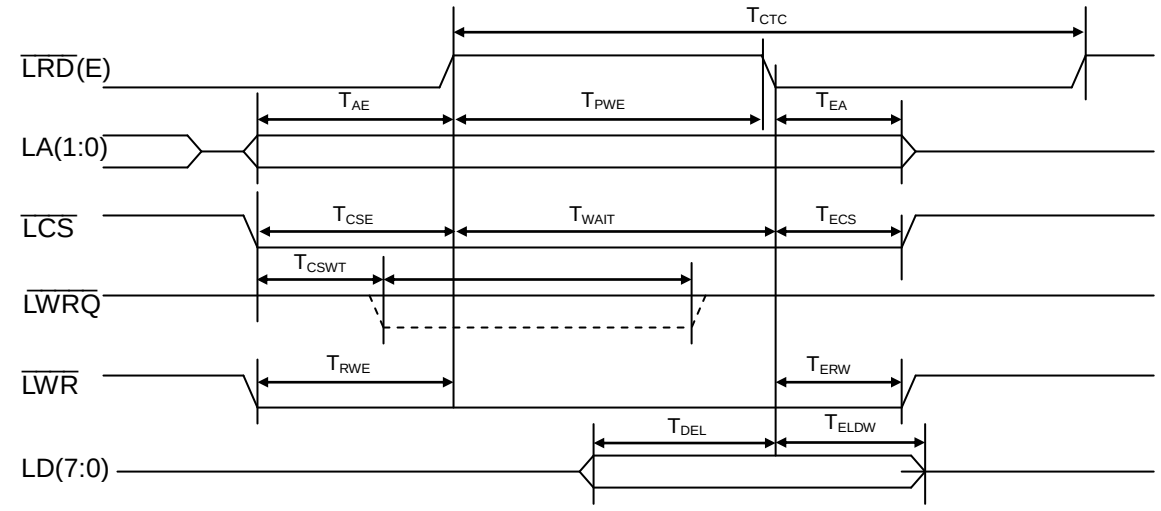
Note1: When CKSL = Low or CKSL = High, the data output delay time will be 12 T_{CLK} + 11.
Note 2: The time that the WRQ signal is output will be the interval after WRQ goes High until WR goes High.

11-5-2. 8-bit I/F (2) (IF1 = High, IF0 = Low)

- Read cycle



- Write cycle

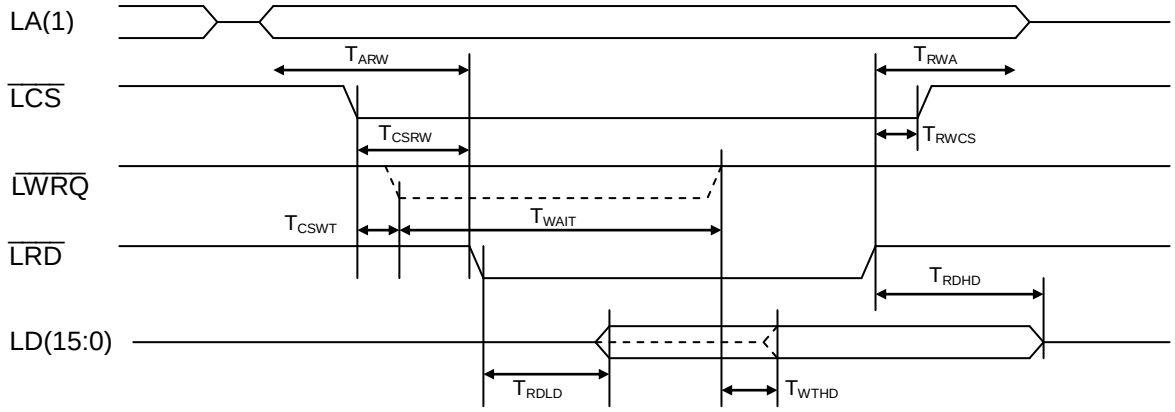


Item	Symbol	Condition	Min.	Max.	Unit
Enable cycle time	T_{CYC}		100		ns
Enable pulse width	T_{PWE}		40		ns
Address setup time for $\overline{LRD}$ (E) ↑	T_{AE}		17		ns
Address hold time for $\overline{LRD}$ (E) ↓	T_{EA}		0		ns
R/W setup time for $\overline{LRD}$ (E) ↑	T_{RWE}		5		ns
R/W hold time for $\overline{LRD}$ (E) ↓	T_{ERW}		5		ns
$\overline{LCS}$ setup time for $\overline{LRD}$ (E) ↑	T_{CSE}		5		ns
$\overline{LCS}$ hold time for $\overline{LRD}$ (E) ↓	T_{ECS}		0		ns
$\overline{LWRQ}$=ON delay time for $\overline{LCS}$ ↓	T_{CSWT}	C_L = 40pF		12	ns
$\overline{LWRQ}$ signal Low time	T_{WAIT}	Note 1		$6T_{CLK}$+11	ns
Data output delay time for $\overline{LRD}$ (E) ↑	T_{EHD}	C_L = 40pF		19	ns
Data output delay time for $\overline{LWRQ}$ ↑	T_{WTHD}	C_L = 40pF		6	ns
Data float delay time for $\overline{LRD}$ (E) ↓	T_{ELDR}	C_L = 40pF		19	ns
Data setup time for $\overline{LRD}$ (E) ↓	T_{DEL}		22		ns
Data hold time for $\overline{LRD}$ (E) ↓	T_{ELDW}		0		ns

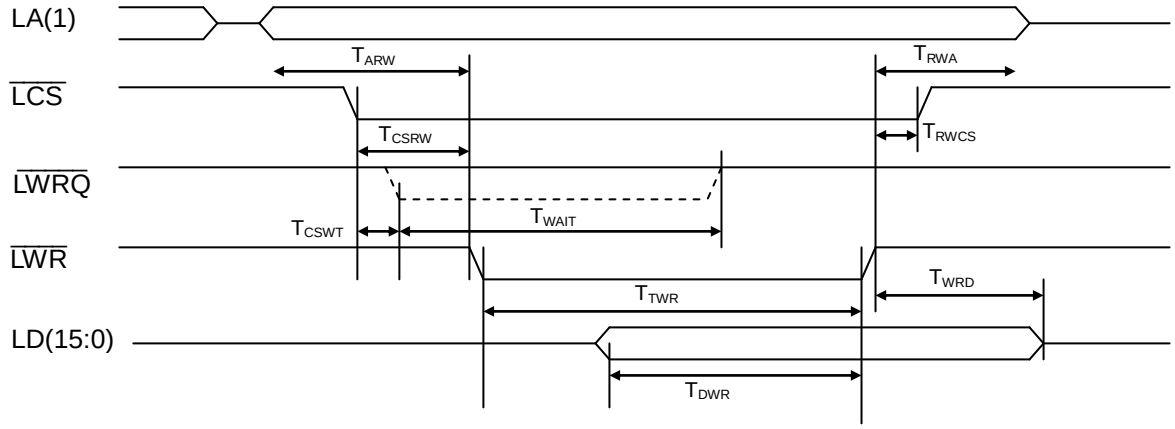
Note 1: When CKSL = Low or CKSL = High, the data output delay time will be 12 T_{CLK} + 11.

11-5-3. 16-bit I/F (1) (IF1 = Low, IF0 = High)

- Read cycle



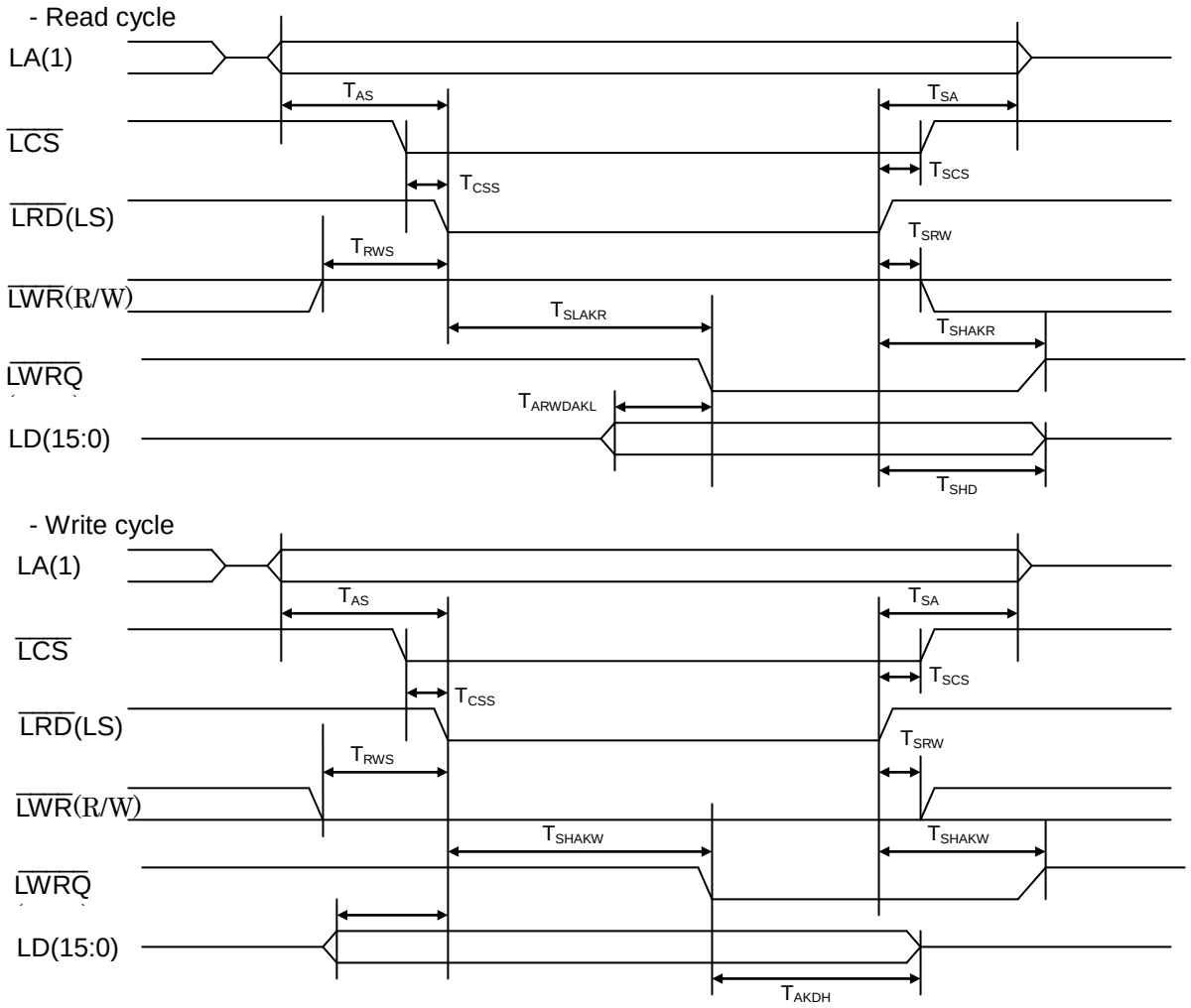
- Write cycle



Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{ARW}		17		ns
Address hold time for $\overline{LRD}$, $\overline{LWR}$ ↑	T_{RWA}		0		ns
$\overline{LCS}$ setup time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{CSRW}		5		ns
$\overline{LCS}$ hold time for $\overline{LRD}$, $\overline{LWR}$ ↑	T_{RWCS}		0		ns
$\overline{LWRQ}$=ON delay time for $\overline{LCS}$ ↓	T_{CSWT}	C_L = 40pF		12	ns
$\overline{LWRQ}$ signal Low time	T_{WAIT}	Note 1		$6T_{CLK}$ +11	ns
Data output delay time for $\overline{LRD}$ ↓	T_{RDLD}	C_L = 40pF		29	ns
Data output delay time for $\overline{LWRQ}$ ↑	T_{WTHD}	C_L = 40pF		16	ns
Data float delay time for $\overline{LRD}$ ↑	T_{RDHD}	C_L = 40pF		30	ns
$\overline{LWR}$ signal width	T_{WR}	Note 2	12		ns
Data setup time for $\overline{LWR}$ ↑	T_{DWR}		22		ns
Data hold time for $\overline{LWR}$ ↑	T_{WRD}		0		ns

Note1: When CKSL = Low or CKSL = High, the data output delay time will be 12 T_{CLK} + 11.
Note 2: The time that the WRQ signal is output will be the interval after WRQ goes High until WR goes High.

11-5-4. 16-bit I/F (2) (IF1 = Low, IF0 = Low)



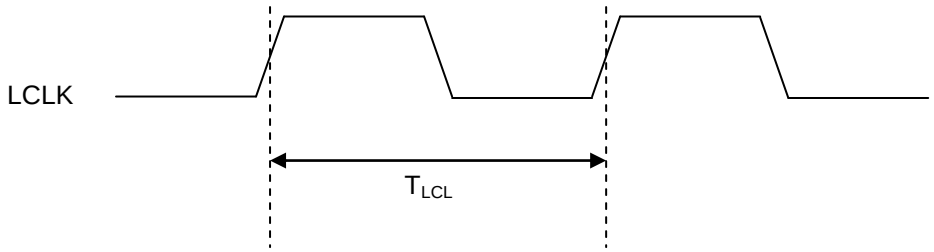
Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{LRD}$ (LS) ↓	T_{As}		17		ns
Address hold time for $\overline{LRD}$ (LS) ↑	T_{sa}		0		ns
$\overline{LCS}$ setup time for $\overline{LRD}$ (LS) ↓	T_{css}		8		ns
$\overline{LCS}$ hold time for $\overline{LRD}$ (LS) ↑	T_{scs}		0		ns
R/W setup time for $\overline{LRD}$ (LS) ↓	T_{RWS}		3		ns
R/W hold time for $\overline{LRD}$ (LS) ↑	T_{SRW}		4		ns
ACK=ON delay time for $\overline{LRD}$ (LS) ↓	T_{SLAKR}	C_L = 40pF Note 1	$2T_{CLK}$	$6T_{CLK}$+9	ns
	T_{SLAKW}	C_L = 40pF Note 1	$2T_{CLK}$	$6T_{CLK}$+9	ns
ACK=ON delay time for $\overline{LRD}$ (LS) ↑	T_{SHAKR}	C_L = 40pF		16	ns
	T_{SHAKW}	C_L = 40pF		16	ns
Data float delay time for $\overline{LWRQ}$ (ACK) ↓	T_{DAKLR}	C_L = 40pF Note 2	$2T_{CLK}$		ns
Data float delay time for $\overline{LRD}$ (LS) ↑	T_{SHD}	C_L = 40pF		30	ns
Data setup time for $\overline{LRD}$ (LS) ↑	T_{DSL}		22		ns
Data hold time for $\overline{LWRQ}$ (ACK) ↓	T_{AKDH}		0		ns

Note 1: When CKSL = Low or CKSL = High, MIN = $4T_{CLK}$ and MAX = $12T_{CLK}$+9.
Note 2: When CKSL = Low or CKSL = High, MIN = $4T_{CLK}$.

11-6. Timing when CPU emulation is selected

11-6-1. LCLK timing

The LCLK uses the following timing, even when the device is not in the emulation mode.



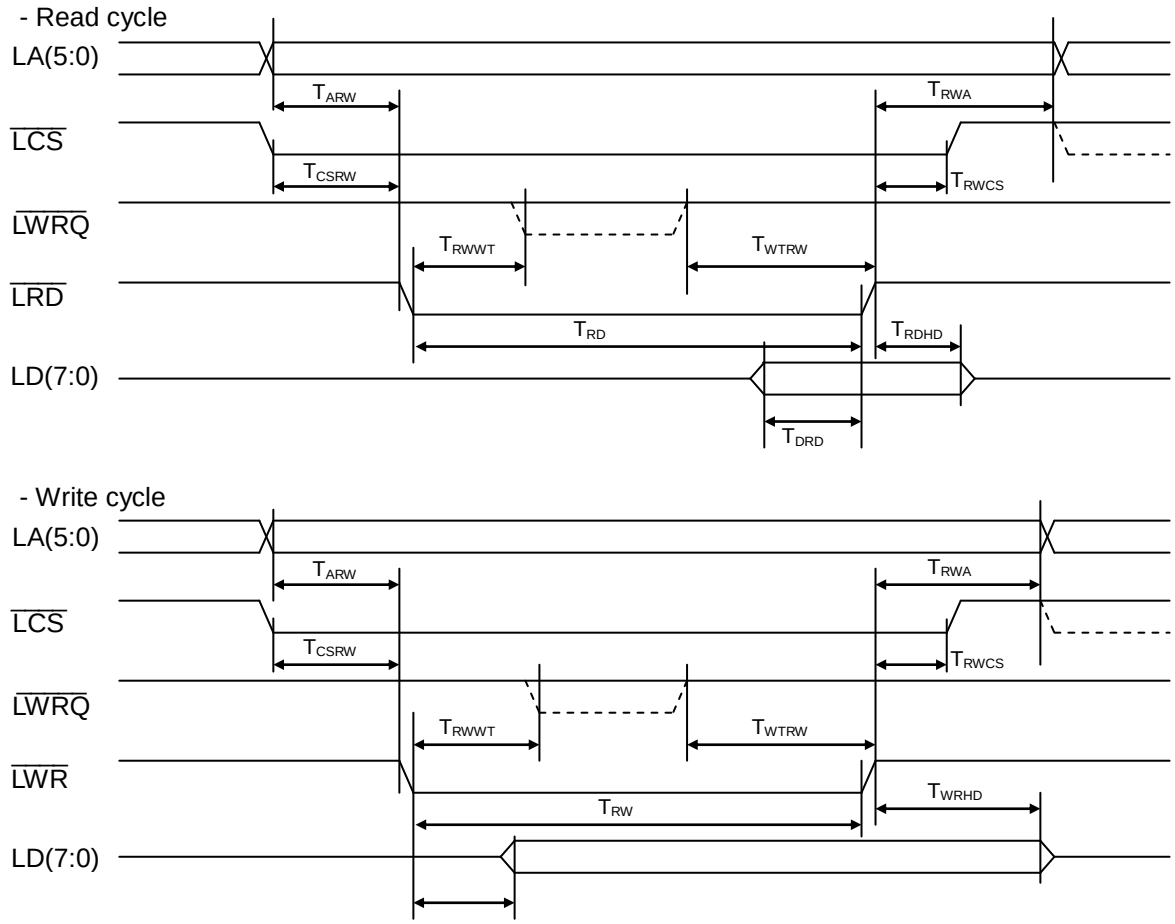
- When CKSL = Low (40 MHz, T_{CLK} = 25 ns)

Item	Symbol	Condition	Cycle	Unit
Clock cycle	T_{LCLK}	LCK1 = L, LCK0 = L 2 MHz	$20T_{CLK}$	ns
		LCK1 = L, LCK0 = H 4 MHz	$10T_{CLK}$	
		LCK1 = H, LCK0 = L 10MHz	$4T_{CLK}$	
		LCK1 = H, LCK0 = H 20 MHz	$2T_{CLK}$	

- When CKSL = High (80 MHz, T_{CLK} = 25 ns)

Item	Symbol	Condition	Cycle	Unit
Clock cycle	T_{LCLK}	LCK1 = L, LCK0 = L 2 MHz	$40T_{CLK}$	ns
		LCK1 = L, LCK0 = H 4 MHz	$20T_{CLK}$	
		LCK1 = H, LCK0 = L 10MHz	$8T_{CLK}$	
		LCK1 = H, LCK0 = H 20 MHz	$4T_{CLK}$	

11-6-2. 8-bit I/F (1) (IF1 = H, IF0 = H)

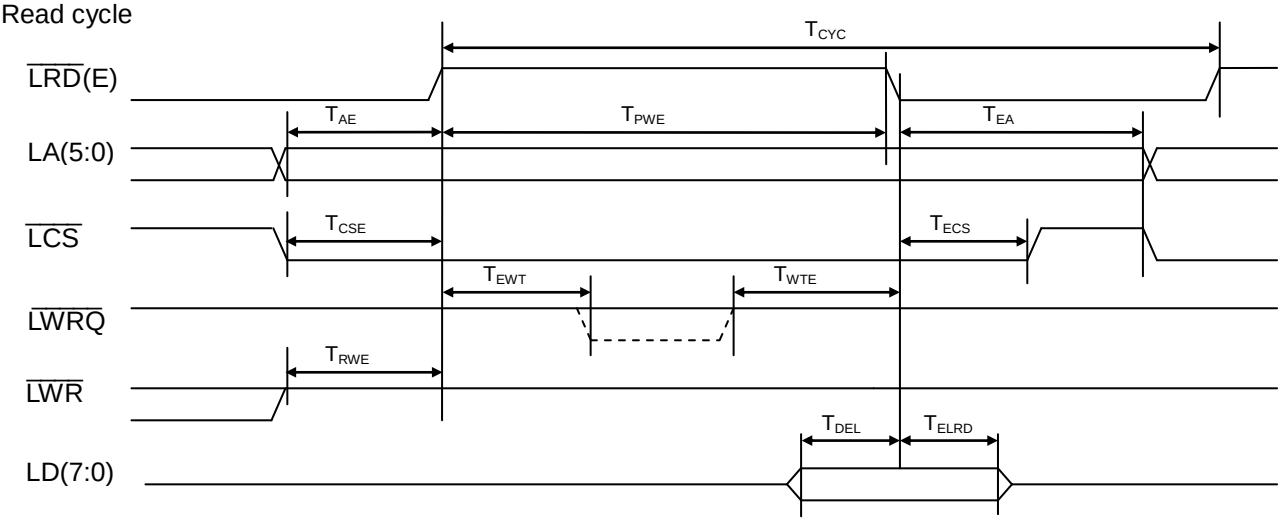


Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{ARW}		T_{LCLK} -1	T_{LCLK} +1	ns
Address hold time for $\overline{LRD}$, $\overline{LWR}$ ↑	T_{RWA}	LWT1 = L, LWT0 = L Note	T_{LCLK} -1	T_{LCLK} +1	ns
CS setup time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{CSRW}		T_{LCLK} -1	T_{LCLK} +1	ns
CS hold time for $\overline{LRD}$, $\overline{LWR}$ ↑	T_{RWCS}		0	1	ns
$\overline{LWRQ}$=ON set up time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{RWWT}			T_{LCLK} -12	ns
$\overline{LRD}$, $\overline{LWR}$ = OFF signal Low time for $\overline{LWRQ}$ ↑	T_{WTRW}	CL = 40pF	$2T_{LCLK}$ +2	$3T_{LCLK}$	ns
Data setup time for $\overline{LRD}$ ↑	T_{DRD}		23		ns
Data hold time for $\overline{LRD}$ ↑	T_{RDHD}		0		ns
$\overline{LRD}$ signal width	T_{RD}		$2T_{LCLK}$		ns
$\overline{LWR}$ signal width	T_{WR}		$2T_{LCLK}$		ns
Data output delay time for $\overline{LWR}$ ↓	T_{WRLD}	CL = 40pF	5	15	ns
Data hold time for $\overline{LWR}$ ↑	T_{WRHD}		T_{LCLK} -2	T_{LCLK} +1	ns

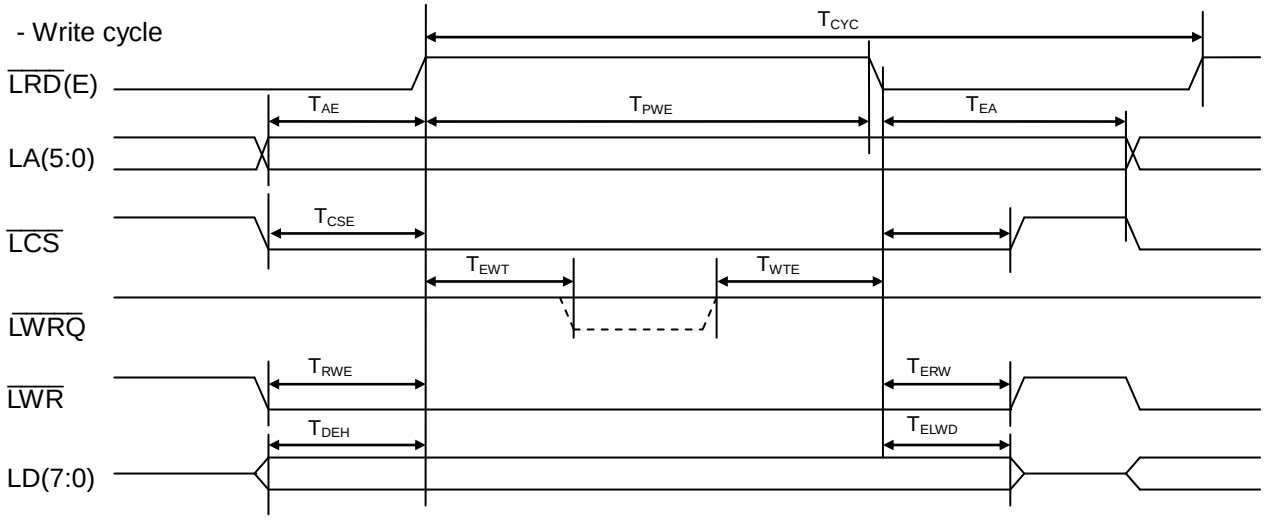
Note: The addresses do not change until the next cycle, so the hold time varies with value used for LWT.

11-6-3. 8-bit I/F (2) (IF1 = H, IF0 = L)

- Read cycle



- Write cycle

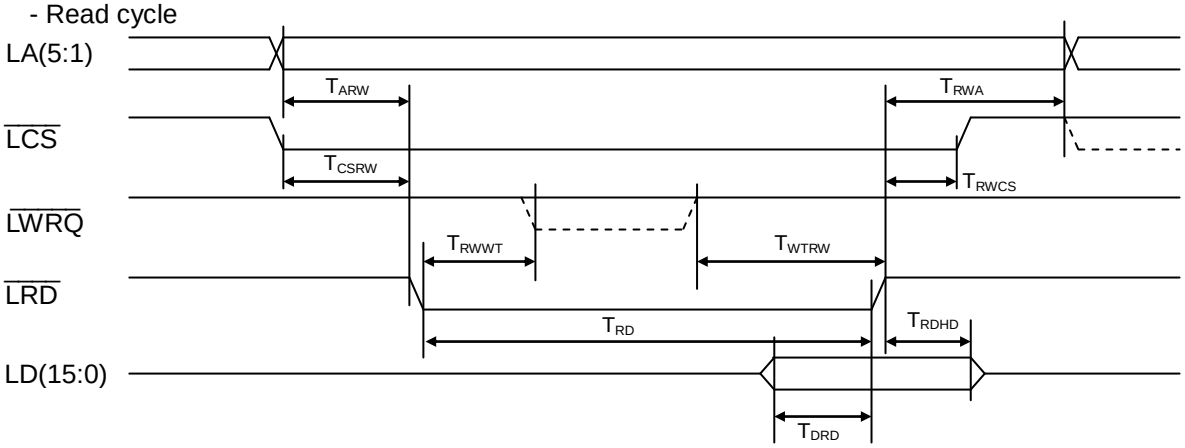


Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{LRD}$ (E) ↑	T_{AE}		T_{LCLK} -1	T_{LCLK} +1	ns
Address hold time for $\overline{LRD}$ (E) ↓	T_{EA}	LWT1 = L, LWT0 = L Note	$2T_{LCLK}$ -1	$2T_{LCLK}$ +1	ns
R/W setup time for $\overline{LRD}$ (E) ↑	T_{RWE}		T_{LCLK} -1	T_{LCLK} +1	ns
R/W hold time for $\overline{LRD}$ (E) ↓	T_{ERW}		T_{LCLK} -1	T_{LCLK} +1	ns
$\overline{LCS}$ setup time for $\overline{LRD}$ (E) ↑	T_{CSE}		T_{LCLK} -1	T_{LCLK} +1	ns
$\overline{LCS}$ hold time for $\overline{LRD}$ (E) ↓	T_{ECS}		T_{LCLK} -1	T_{LCLK} +1	ns
$\overline{LWRQ}$=ON set time for $\overline{LRD}$ (E) ↑	T_{EWT}	CL = 40pF		$2T_{LCLK}$ -12	ns
$\overline{LWRQ}$ signal Low time for $\overline{LWRQ}$ ↑	T_{WTE}	CL = 40pF	T_{LCLK} +2	$2T_{LCLK}$	ns
Data setup time for $\overline{LRD}$ (E) ↓	T_{DEL}		23		ns
Data hold time for $\overline{LRD}$ (E) ↓	T_{ELRD}		0		ns
$\overline{LRD}$ (E) signal width	T_{PWE}		T_{LCLK}		ns
$\overline{LRD}$ (E) cycle time	T_{CYC}		$5T_{LCLK}$		ns
Data setup time for $\overline{LRD}$ (E) ↓	T_{DEL}		T_{LCLK} -3	T_{LCLK} +1	ns
Data hold time for $\overline{LRD}$ (E) ↓	T_{ELWD}		T_{LCLK} -1	T_{LCLK} +1	ns

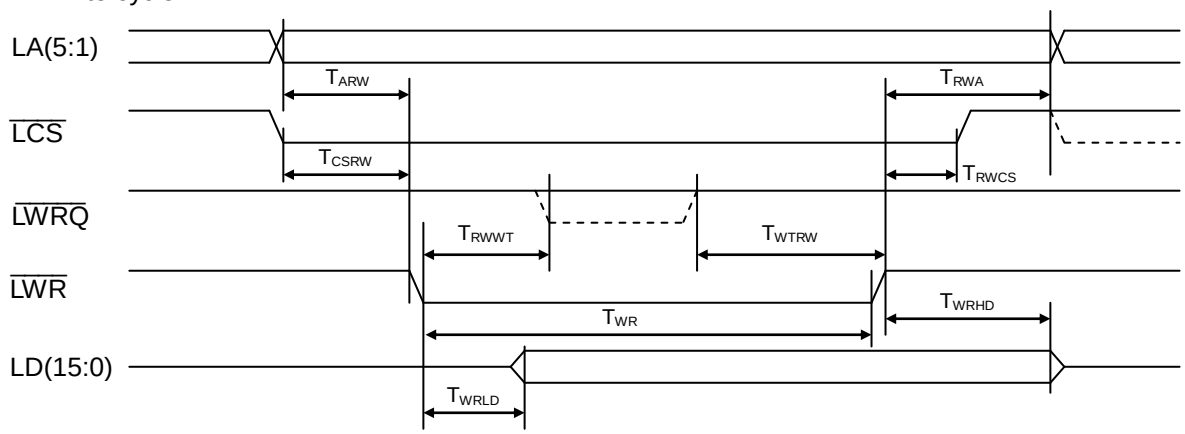
Note 1: When CKSL = Low or CKSL = High, the data output delay time will be 12 T_{CLK} + 11.

11-6-4. 16-bit I/F (1) (IF1 = L, IF0 = H)

- Read cycle



- Write cycle

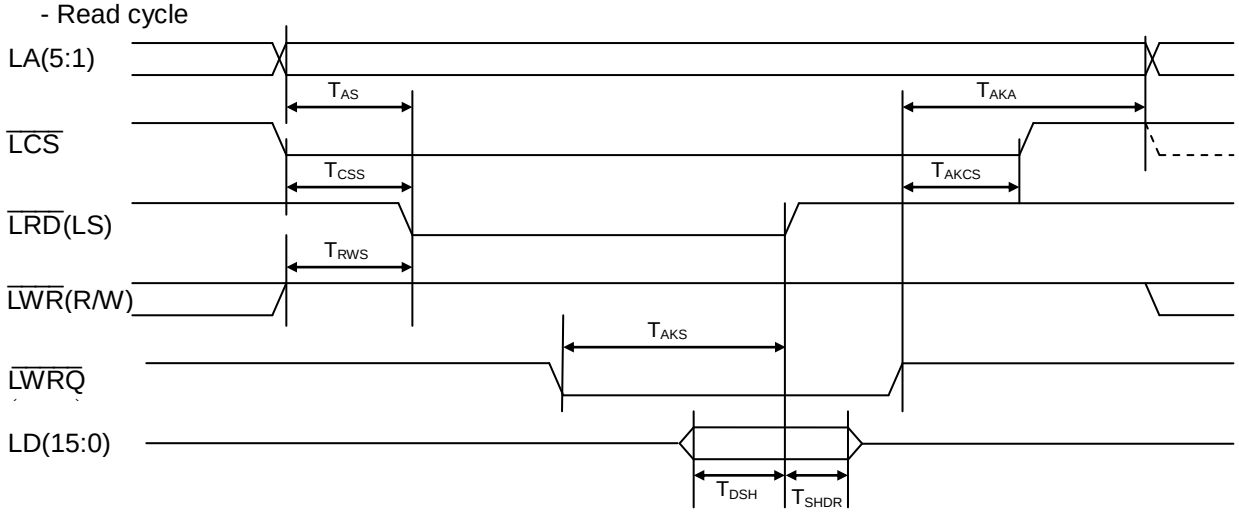


Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{ARW}		T_{LCLK} -1	T_{LCLK} +1	ns
Address hold time for $\overline{LRD}$, $\overline{LWR}$ ↑	T_{RWA}	LWT1 = L, LWT0 = L Note	T_{LCLK} -1	T_{LCLK} +1	ns
CS setup time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{CSRW}		T_{LCLK} -1	T_{LCLK} +1	ns
CS hold time for $\overline{LRD}$, $\overline{LWR}$ ↑	T_{RWCS}		0	1	ns
$\overline{LWRQ}$=ON set time for $\overline{LRD}$, $\overline{LWR}$ ↓	T_{RWWT}			T_{LCLK} -12	ns
$\overline{LRD}$, $\overline{LWRQ}$ = OFF delay time for $\overline{LWRQ}$ ↑	T_{WTRW}	CL = 40pF	$2T_{LCLK}$ +2	$3T_{LCLK}$	ns
Data setup time for $\overline{LRD}$ ↑	T_{DRD}		23		ns
Data hold time for $\overline{LRD}$ ↑	T_{RDHD}		0		ns
$\overline{LRD}$ signal width	T_{RD}		$2T_{LCLK}$		ns
$\overline{LWR}$ cycle time	T_{WR}		$2T_{LCLK}$		ns
Data setup time for $\overline{LWR}$ ↓	T_{WRLD}	CL = 40pF	5	15	ns
Data hold time for $\overline{LWR}$ ↑	T_{WRHD}		T_{LCLK} -2	T_{LCLK} +1	ns

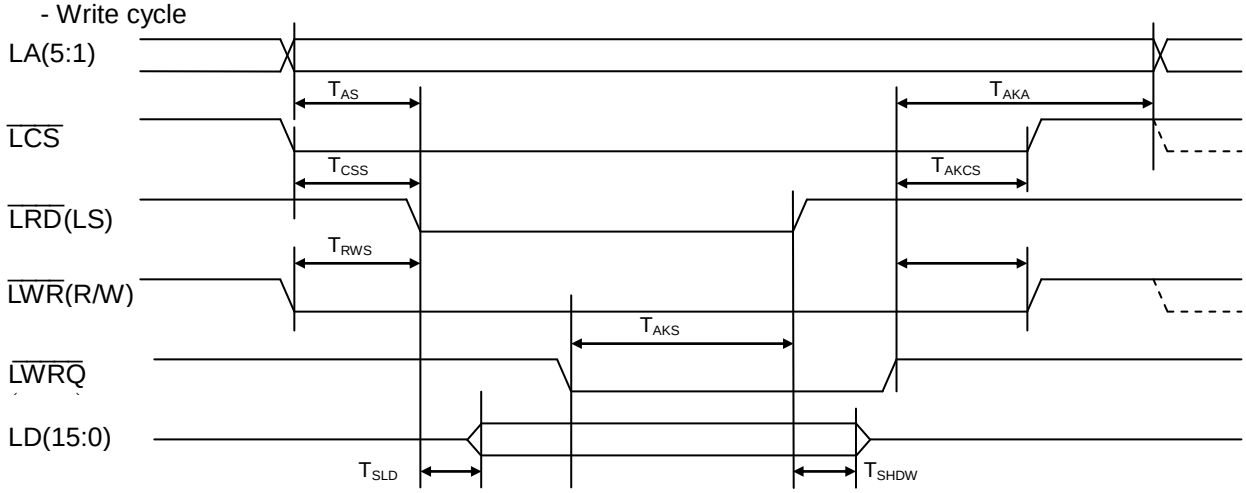
Note: The hold time varies with the LWT set value as the address does not change until next cycle.

11-6-5. 16-bit I/F (2) (IF1 = L, IF0 = L)

- Read cycle



- Write cycle

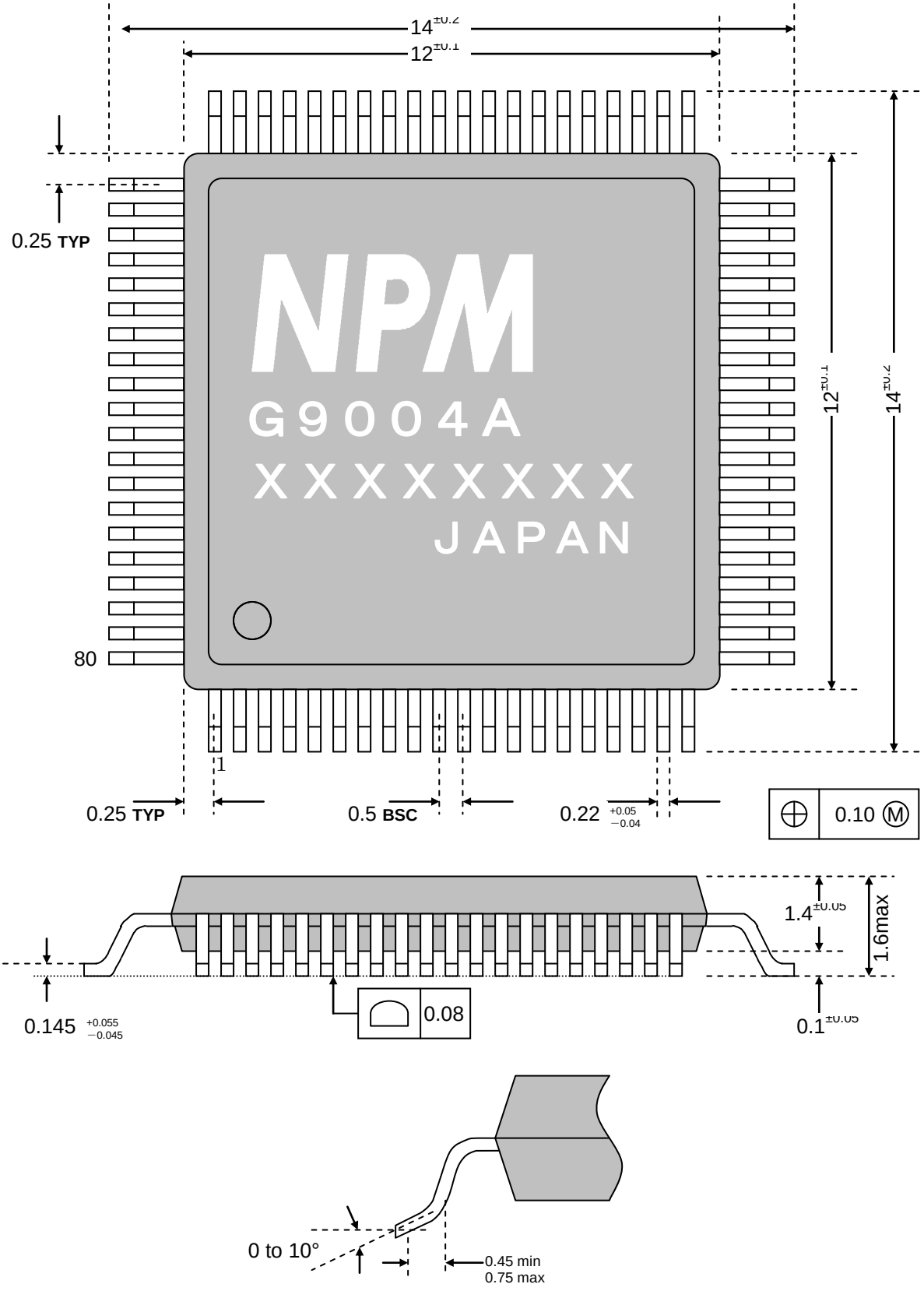


Item	Symbol	Condition	Min.	Max.	Unit
Address setup time for LS ↓	T_{AS}		T_{LCLK} -1	T_{LCLK} +1	ns
Address hold time for ACK ↑	T_{AKA}	LWT1 = L, LWT0 = L Note	$2T_{LCLK}$ -1	$3T_{LCLK}$ +1	ns
$\overline{LCS}$ setup time for LS ↓	T_{CSS}		T_{LCLK} -1	T_{LCLK} +1	ns
$\overline{LCS}$ hold time for ACK ↑	T_{AKCS}		T_{LCLK} -1	$2T_{LCLK}$ +1	ns
R/W setup time for LS ↓	T_{RWS}		T_{LCLK} -1	T_{LCLK} +1	ns
R/W hold time for ACK ↑	T_{AKRW}		T_{LCLK} -1	$2T_{LCLK}$ +1	ns
LS ($\overline{LRD}$) OFF delay time for ACK ↓	T_{AKS}	CL= 40pF	T_{LCLK}	$2T_{LCLK}$ +1	ns
Data setup time for LS ↑	T_{DSH}		23		ns
Data hold time for LS ↑	T_{SHDR}		0		ns
Data output delay time for LS ↓	T_{SLD}	CL= 40pF	2	17	ns
Data hold time for LS ↑	T_{SHDW}		0	1	ns

Note: The hold time varies with the LWT set value as the address does not change until next cycle.

12. External dimensions

80pin LQFP Unit: mm



13. Handling precautions

13-1. Design precautions

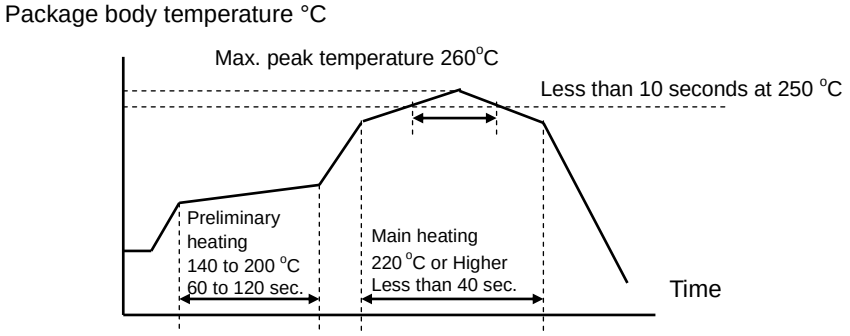
1) Never exceed the absolute maximum ratings, even for a very short time.
2) Take precautions against the influence of heat in the environment, and keep the temperature around the LSI as cool as possible.
3) Please note that ignoring the following may result in latching up and may cause overheating and smoke.
   - Do not apply a voltage greater than V_{DD}+3.3V (greater than 5.8V for 5V connectable terminals) to the input/output terminals and do not pull them below GND.
   - Please consider the voltage drop timing when turning the power ON/OFF. Consider power voltage drop timing when turning ON/OFF the power.
   - 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 overvoltage caused by noise, voltage surges, or static electricity are not fed to the LSI.

13-2. 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.

13-3. Precautions for mounting

1) Plastic packages absorb moisture easily. Even if they are stored indoors, they will absorb moisture as time passes. Putting the packages in to a solder reflow furnace while they contain moisture may cause cracks in plastic case or deteriorate the bonding between the plastic case and the frame. The storage warranty period is one year as long as the moisture barrier bags are not opened.
2) If you are worried about moisture absorption, dry the chip packages thoroughly before reflowing the solder. Dry the packages for 20 to 36 hours at 125+/-5°C. The packages should not be dried more than two times.
3) To heat the entire package for soldering, such as infrared or superheated air reflow, make sure to observe the following conditions and do not reflow more than two times.
   - Temperature profile
     The temperature profile of an infrared reflow furnace must be within the range shown in the figure below. (The temperatures shown are the temperature at the surface of the plastic package.)
   - Maximum temperature
     The maximum allowable temperature at the surface of the plastic package is 260°C peak [A profile]. The temperature must not exceed 250°C [A profile] for more than 10 seconds. In order to decrease the heat stress load on the packages, keep the temperature as Low as possible and as short as possible, while maintaining the proper conditions for soldering.



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

4) Solder dipping causes rapid temperature changes in the packages and may damage the devices. Therefore, do not use this method.
5) In the case of using a soldering iron, the temperature of the soldering bit should be less than 350 degree and the time should be less than 5 seconds. The number of time should be twice or less.

13-4. 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

April 18, 2014

No. DA70120-1/3E