

## 16 bits, Fixed-point Digital Signal Processor

The μPD77019-013 is a masked 16 bits fixed-point DSP (Digital Signal Processor) developed for digital signal processing with its demand for high speed and precision.

The μPD77019-013 internal ROM area is masked already by the void code to use as RAM based DSP without mask code ordering process. Also the μPD77019-013 can operate as simplified evaluate chip for as the μPD7701x family.

About mask ROM and mask option, there are following differences between the μPD77019-013 and μPD77019.

	μPD77019-013	μPD77019
PLL clock multiple rate	Fixed to 4	Variable multiple rate (1, 2, 3, 4, 8) by mask option
Crystal resonator connection	Disable (PLL clock multiple rate is fixed to 4.)	Enable (PLL clock multiple rate is set to 1 by mask option.)
Clock Input pin	External clock is connected to the X1 pin. Leave the X2 pin open.	External clock is connected to the X1 pin. Crystal is connected between the X1 pin and X2 pin.
Clock output pin	Output internal system clock.	Low level fixed, or internal system clock output is selectable by mask option.
Internal mask ROM	Not available (already masked by the void code)	Coding user program or data when ordering mask ROM.
Self boot	Enable to boot from external data area (Boot information data is masked.) Refer to <b>2.6 Boot Function</b> .	Enable to boot from internal data ROM or external data area.

### ORDERING INFORMATION

Part Number	Package
μPD77019GC-013-9EU	100-pin plastic TQFP (FINE PITCH) (14 × 14 mm)

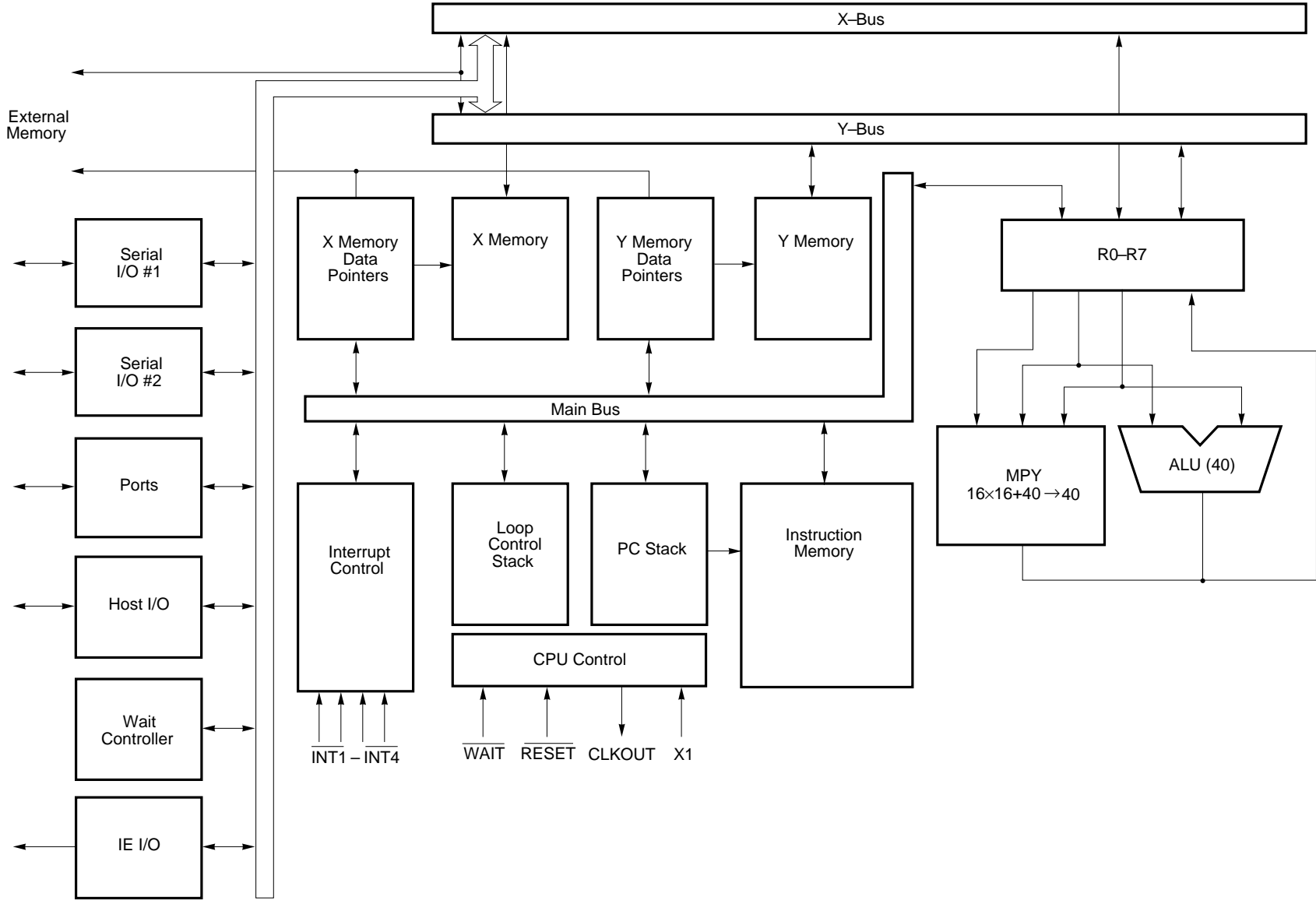
The μPD7701x family consists of the μPD77016, 77015, 77017, 77018, 77018A and μPD77019.

The information in this document is subject to change without notice.

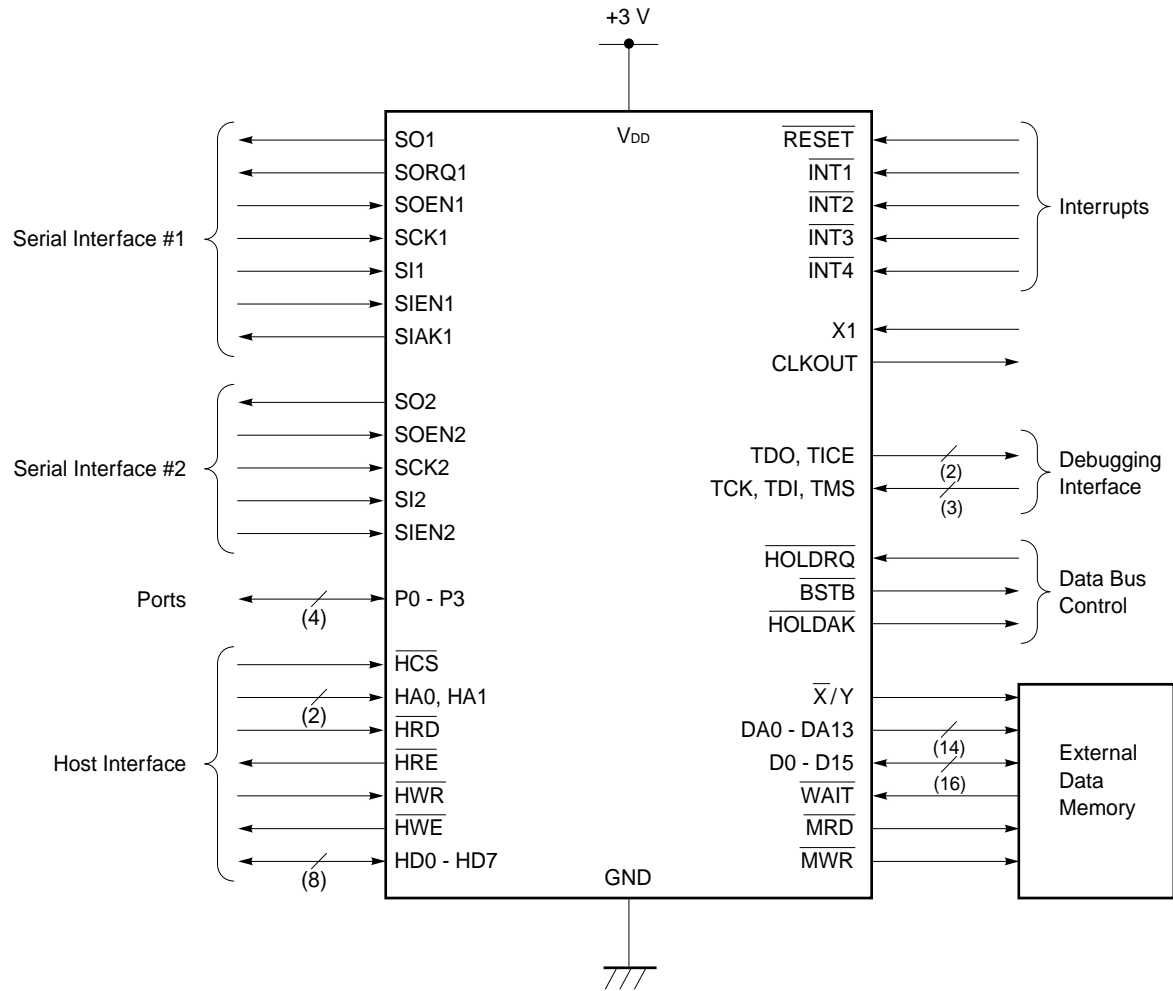
**FEATURES**

- FUNCTIONS
  - Instruction cycle: 16.6 ns (MIN.)
    - Operation clock: 60 MHz
    - External clock: 15 MHz
  - Dual load/store
  - Hardware loop function
  - Conditional execution
  - Executes product-sum operation in one instruction cycle
- PROGRAMMING
  - 16 bits  $\times$  16 bits + 40 bits  $\rightarrow$  40 bits multiply accumulator
  - 8 general registers (40 bits each)
  - 8 ROM/RAM data pointer: each data memory area has 4 registers
  - 10 source interrupts (external: 4, internal: 6)
  - 3 operand instructions (example:  $R0 = R0 + R1L * R2L$ )
  - Nonpipeline on execution stage
- MEMORY AREAS
  - Instruction memory area : 64K words  $\times$  32 bits
  - Data memory areas : 64K words  $\times$  16 bits  $\times$  2 (X memory, Y memory)
- CLOCK GENERATOR
  - On-chip PLL to provide higher operation clock (60 MHz max.) than the external clock. PLL clock multiple rate is fixed to 4.
- ON-CHIP PERIPHERAL
  - I/O port: 4 bits
  - Serial I/O (16 bits): 2 channels
  - Host I/O (8 bits): 1 channel
- CMOS
- +3 V single power supply

BLOCK DIAGRAM



FUNCTIONAL PIN GROUPS



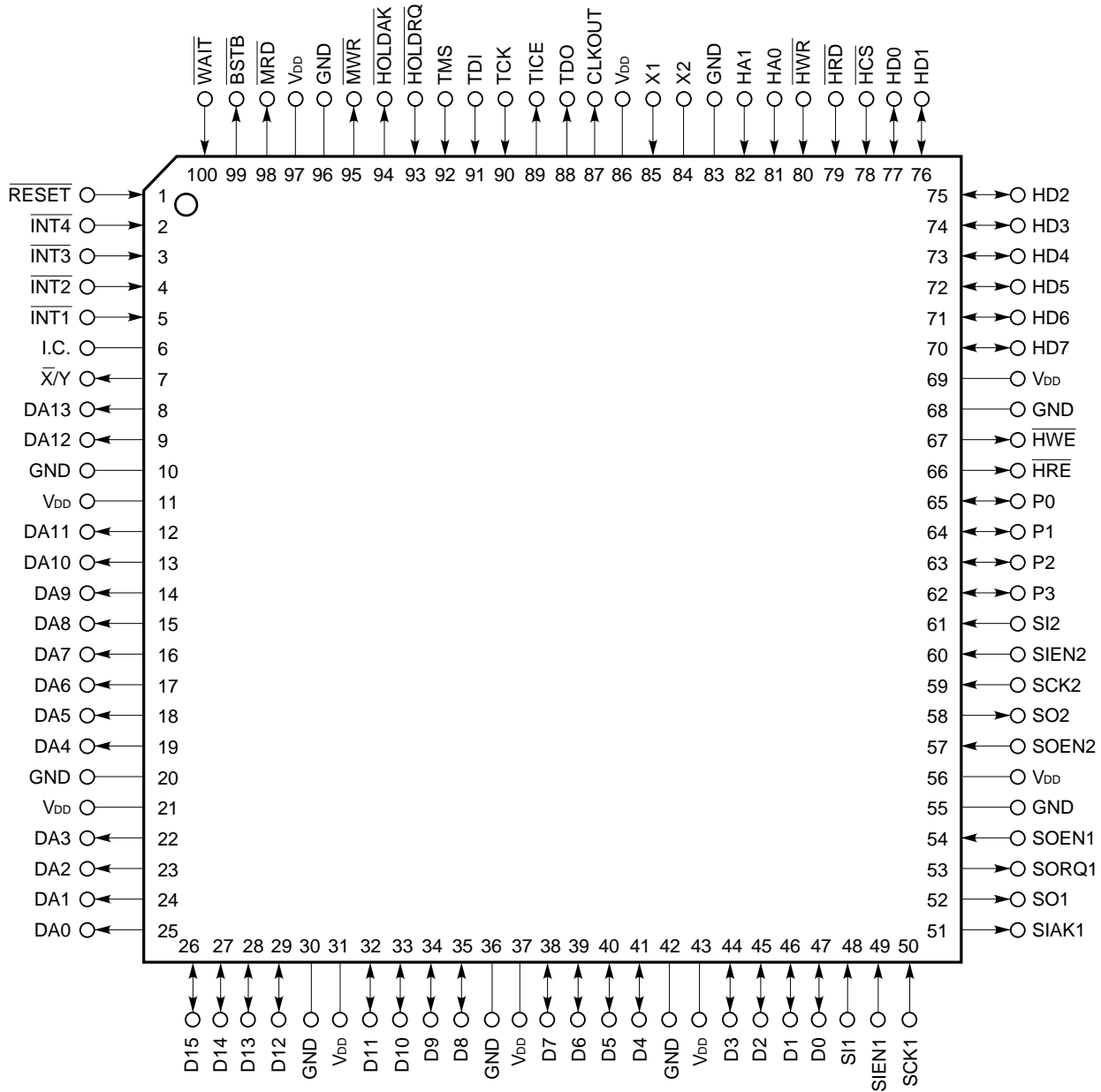
## Functional Differences among the μPD7701× Family

Item	μPD77016	μPD77015	μPD77017	μPD77018	μPD77018A	μPD77019	μPD77019-013
Internal instruction RAM	1.5K words	256 words				4K words	
Internal instruction ROM	None	4K words	12K words	24K words		None	
External instruction memory	48K words	None					
Data RAM (X/Y memory)	2K words each	1K words each	2K words each	3K words each			
Data ROM (X/Y memory)	None	2K words each	4K words each	12K words each		None	
External data memory	48K words each	16K words each					
Instruction cycle (Maximum operation speed)	30 ns (33 MHz)				16.6 ns (60 MHz)		
External clock (at maximum operation speed)	66 MHz	33/16.5/8.25/4.125 MHz Variable multiple rate (1, 2, 4, 8 ) by mask option.			60/30/20/15/7.5 MHz Variable multiple rate (1, 2, 3, 4, 8 ) by mask option.		15 MHz Multiple rate is fixed to 4.
Crystal (at maximum operation speed)	–	33 MHz			60 MHz		–
Instruction	–	STOP instruction is added.					
Serial interface (2 Channels)	Channel 1 has the same functions as channel 2.	Channel 1 has the same functions as that of the μPD77016. Channel 2 has no SORQ2 or SIAK2 pin (Channel 2 is used for CODEC connection).					
Power supply	5V	3 V					
Package	160-pin plastic QFP	100-pin plastic TQFP			100-pin plastic TQFP 116-pin plastic BGA	100-pin plastic TQFP	

PIN CONFIGURATION (Top View)

μPD77019GC-013-9EU

100-pin plastic TQFP (FINE PITCH) (14 × 14 mm)



## PIN IDENTIFICATION

$\overline{\text{BSTB}}$ :	Bus Strobe
CLKOUT:	Clock Output
D0-D15:	16 Bits Data Bus
DA0-DA13:	External Data Memory Address Bus
GND:	Ground
HA0,HA1:	Host Data Access
$\overline{\text{HCS}}$ :	Host Chip Select
HD0-HD7:	Host Data Bus
$\overline{\text{HOLDAK}}$ :	Hold Acknowledge
$\overline{\text{HOLDRQ}}$ :	Hold Request
$\overline{\text{HRD}}$ :	Host Read
$\overline{\text{HRE}}$ :	Host Read Enable
$\overline{\text{HWE}}$ :	Host Write Enable
$\overline{\text{HWR}}$ :	Host Write
I.C.:	Internally Connected
$\overline{\text{INT1-INT4}}$ :	Interrupt
$\overline{\text{MRD}}$ :	Memory Read Output
$\overline{\text{MWR}}$ :	Memory Write Output
P0-P3:	Port
$\overline{\text{RESET}}$ :	Reset
SCK1,SCK2:	Serial Clock Input
SI1,SI2:	Serial Data Input
SIACK1:	Serial Input Acknowledge
SIEN1,SIEN2:	Serial Input Enable
SO1,SO2:	Serial Data Output
SOEN1,SOEN2:	Serial Output Enable
SORQ1:	Serial Output Request
TCK:	Test Clock Input
TDI:	Test Data Input
TDO:	Test Data Output
TICE:	Test In-Circuit Emulator
TMS:	Test Mode Select
V <sub>DD</sub> :	Power Supply
$\overline{\text{WAIT}}$ :	Wait Input
X1, X2:	Crystal connection
$\overline{\text{X/Y}}$ :	X/Y Memory Select

**PIN NAME**

Pin No.	Symbol	Pin No.	Symbol	Pin No.	Symbol	Pin No.	Symbol
1	$\overline{\text{RESET}}$	26	D15	51	SI AK1	76	HD1
2	$\overline{\text{INT4}}$	27	D14	52	SO1	77	HD0
3	$\overline{\text{INT3}}$	28	D13	53	SORQ1	78	$\overline{\text{HCS}}$
4	$\overline{\text{INT2}}$	29	D12	54	SOEN1	79	$\overline{\text{HRD}}$
5	$\overline{\text{INT1}}$	30	GND	55	GND	80	$\overline{\text{HWR}}$
6	I.C. <b>Note 1</b>	31	V <sub>DD</sub>	56	V <sub>DD</sub>	81	HA0
7	$\overline{\text{X}}/\text{Y}$	32	D11	57	SOEN2	82	HA1
8	DA13	33	D10	58	SO2	83	GND
9	DA12	34	D9	59	SCK2	84	X2 <b>Note 2</b>
10	GND	35	D8	60	SIEN2	85	X1
11	V <sub>DD</sub>	36	GND	61	SI2	86	V <sub>DD</sub>
12	DA11	37	V <sub>DD</sub>	62	P3	87	CLKOUT
13	DA10	38	D7	63	P2	88	TDO
14	DA9	39	D6	64	P1	89	TICE
15	DA8	40	D5	65	P0	90	TCK
16	DA7	41	D4	66	$\overline{\text{HRE}}$	91	TDI
17	DA6	42	GND	67	$\overline{\text{HWE}}$	92	TMS
18	DA5	43	V <sub>DD</sub>	68	GND	93	$\overline{\text{HOLDRQ}}$
19	DA4	44	D3	69	V <sub>DD</sub>	94	$\overline{\text{HOLDAK}}$
20	GND	45	D2	70	HD7	95	$\overline{\text{MWR}}$
21	V <sub>DD</sub>	46	D1	71	HD6	96	GND
22	DA3	47	D0	72	HD5	97	V <sub>DD</sub>
23	DA2	48	SI1	73	HD4	98	$\overline{\text{MRD}}$
24	DA1	49	SIEN1	74	HD3	99	$\overline{\text{BSTB}}$
25	DA0	50	SCK1	75	HD2	100	$\overline{\text{WAIT}}$

- Note 1.** I.C. (Internally Connected): Leave this pin open.  
**2.** Leave this pin open.



CONTENTS

1. PIN FUNCTIONS ..... 10

    1.1 Pin Functions ..... 10

    1.2 Recommended Connection for Unused Pins ..... 15

2. FUNCTIONS ..... 16

    2.1 Pipeline Processing ..... 16

        2.1.1 Outline ..... 16

        2.1.2 Instructions with Delay ..... 16

    2.2 Program Control Unit ..... 17

    2.3 Operation Unit ..... 17

        2.3.1 General register (R0 to R7) ..... 17

        2.3.2 MAC: Multiply ACcumulator ..... 18

        2.3.3 ALU: Arithmetic Logic Unit ..... 18

        2.3.4 BSFT: Barrel ShiFTer ..... 18

        2.3.5 SAC: Shifter And Count Circuit ..... 18

        2.3.6 CJC: Condition Judge Circuit ..... 19

    2.4 Memory ..... 19

        2.4.1 Instruction RAM Outline ..... 20

        2.4.2 Data Memory Outline ..... 21

        2.4.3 Data Memory Addressing ..... 22

    2.5 On-chip Peripheral Circuit ..... 22

        2.5.1 Serial Interface Outline ..... 22

        2.5.2 Host Interface Outline ..... 22

        2.5.3 General Input/output Ports Outline ..... 22

        2.5.4 Wait Cycle Register ..... 22

    2.6 Boot Function ..... 23

3. INSTRUCTIONS ..... 24

    3.1 Outline ..... 24

    3.2 Instruction Set and Operation ..... 25

4. ELECTRICAL SPECIFICATIONS ..... 33

5. PACKAGE DRAWING ..... 53

6. RECOMMENDED SOLDERING CONDITIONS ..... 54

## 1. PIN FUNCTIONS

### 1.1 Pin Functions

#### • Power supply

Symbol	Pin No.	I/O	Function
V <sub>DD</sub>	11, 21, 31, 37, 43, 56, 69, 86, 97	–	+3V power supply
GND	10, 20, 30, 36, 42, 55, 68, 83, 96	–	Ground

#### • System control

Symbol	Pin No.	I/O	Function
X1	85	I	Clock input
X2	84	–	Leave this pin open
CLKOUT	87	O	Internal system clock output
$\overline{\text{RESET}}$	1	I	Internal system reset signal input

#### • Interrupt

Symbol	Pin No.	I/O	Function
$\overline{\text{INT4}} - \overline{\text{INT1}}$	2 - 5	I	Maskable external interrupt input • Falling edge detection

## • External data memory interface

Symbol	Pin No.	I/O	Function
$\overline{X/Y}$	7	O (3S)	Memory select signal output <ul style="list-style-type: none"> <li>• 0: X memory is used.</li> <li>• 1: Y memory is used.</li> </ul>
DA13 - DA0	8, 9, 12 -19, 22 - 25	O (3S)	Address bus to external data memory <ul style="list-style-type: none"> <li>• External data memory is accessed.</li> <li>• During the external memory is not accessed, these pins keep the previous level.</li> </ul> These pins are set to low level; 0x0000, by reset. They continue outputting low level until the first external memory access.
D15 - D0	26 -29, 32 - 35, 38 - 41, 44 - 47	I/O (3S)	16 bits data bus to external data memory <ul style="list-style-type: none"> <li>• External data memory is accessed.</li> </ul>
$\overline{MRD}$	98	O (3S)	Read output <ul style="list-style-type: none"> <li>• Reads external memory</li> </ul>
$\overline{MWR}$	95	O (3S)	Write output <ul style="list-style-type: none"> <li>• Writes external memory</li> </ul>
$\overline{WAIT}$	100	I	Wait signal input <ul style="list-style-type: none"> <li>• Wait cycle is input when external memory is read.</li> </ul> 1: No wait 0: Wait
$\overline{HOLDRQ}$	93	I	Hold request signal input <ul style="list-style-type: none"> <li>• Input low level when external data memory bus is expected to use.</li> </ul>
$\overline{BSTB}$	99	O	Bus strobe signal output <ul style="list-style-type: none"> <li>• Outputs low level while the <math>\mu</math>PD77019-013 is occupying external memory bus.</li> </ul>
$\overline{HOLDAK}$	94	O	Hold acknowledge signal output <ul style="list-style-type: none"> <li>• Outputs low level when the <math>\mu</math>PD77019-013 permits external device to use external data memory bus.</li> </ul>

**Remark** The state of the pins added 3S becomes high impedance when the external memory is not accessed and when bus release signal ( $\overline{HOLDAK} = 0$ ) is output.

- Serial interface

Symbol	Pin No.	I/O	Function
SCK1	50	I	Clock input for serial 1
SORQ1	53	O	Serial output 1 request
SOEN1	54	I	Serial output 1 enable
SO1	52	O (3S)	Serial data output 1
SIEN1	49	I	Serial input 1 enable
SI1	48	I	Serial data input 1
SCK2	59	I	Clock input for serial 2
SOEN2	57	I	Serial output 2 enable
SO2	58	O (3S)	Serial data output 2
SIEN2	60	I	Serial input 2 enable
SI2	61	I	Serial data input 2
SIK1	51	O	Serial input 1 acknowledge

**Remark** The state of the pins added 3S becomes high impedance, when data output have been finished or  $\overline{\text{RESET}}$  is input.

• Host interface

Symbol	Pin No.	I/O	Function
HA1	82	I	Specifies register which HD7 to HD0 access 1: Accesses HST: Host interface status register when HA1 = 0 0: Accesses HDT(in): Host transmit data register when $\overline{\text{HWR}} = 0$ 0: Accesses HDT(out): Host receive data register when $\overline{\text{HRD}} = 0$
HA0	81	I	Specifies bits of registers which HD7 to HD0 access • 1: Accesses bits 15-8 of HST, HDT(in) or HDT(out) • 0: Accesses bits 7-0 of HST, HDT(in) or HDT(out)
$\overline{\text{HCS}}$	78	I	Chip select input
$\overline{\text{HRD}}$	79	I	Host read input
$\overline{\text{HWR}}$	80	I	Host write input
$\overline{\text{HRE}}$	66	O	Host read enable output
$\overline{\text{HWE}}$	67	O	Host write enable output
HD7 - HD0	70 - 77	I/O (3S)	8 bits host data bus

**Remark** The state of the pins added 3S becomes high impedance when the host does not access host interface.

• I/O port

Symbol	Pin No.	I/O	Function
P3 - P0	62 - 65	I/O	I/O port

• Debugging interface

Symbol	Pin No.	I/O	Function
TDO	88	O	For debugging
TICE	89	O	For debugging
TCK	90	I	For debugging
TDI	91	I	For debugging
TMS	92	I	For debugging

• Other

Symbol	Pin No.	I/O	Function
I.C.	6	–	Internal connected pin. Leave this pin open. <b>Caution</b> When any signal is applied to or read out from this pin, normal operation of the μPD77019-013 is not assured.

1.2 Recommended Connection for Unused Pins

Pin	I/O	Recommended connection
$\overline{\text{INT1}} - \overline{\text{INT4}}$	I	connect to $V_{DD}$
$\overline{\text{X}}/\overline{\text{Y}}$	O	open
DA0 - DA13	O	
D0 - D15 <sup>Note1</sup>	I/O	connect to $V_{DD}$ or GND, via a resistor
$\overline{\text{MRD}}$ , $\overline{\text{MWR}}$	O	open
$\overline{\text{WAIT}}$	I	connect to $V_{DD}$
$\overline{\text{HOLDRQ}}$	I	
$\overline{\text{BSTB}}$	O	open
$\overline{\text{HOLDAK}}$	O	
SCK1, SCK2	I	connect to $V_{DD}$ or GND
SI1, SI2	I	
SOEN1, SOEN2	I	connect to GND
SIEN1, SIEN2	I	
SORQ1	O	open
SO1, SO2	O	
SIK1	O	
HA0, HA1	I	connect to $V_{DD}$ or GND
$\overline{\text{HCS}}$	I	connect to $V_{DD}$
$\overline{\text{HRD}}$ , $\overline{\text{HWR}}$	I	
$\overline{\text{HRE}}$ , $\overline{\text{HWE}}$	O	open
HD0 - HD7 <sup>Note2</sup>	I/O	connect to $V_{DD}$ or GND, via a resistor
P0 - P3	I/O	
TCK	I	connect to GND, via a resistor
TDO, TICE	O	open
TMS, TDI	I	open(pull-up internally)
CLKOUT	O	open

**Notes 1.** Can leave open, if no access to external data memory is executed in the whole of program.

But in the HALT mode or STOP mode when the current consumption is reduced, connect a pin as recommended connection.

**2.** Can leave open, if  $\overline{\text{HCS}}$ ,  $\overline{\text{HRD}}$ ,  $\overline{\text{HWR}}$  are fixed to high level. But in the HALT mode or STOP mode when the current consumption is reduced, connect a pin as recommended connection.

**Remark** I: Input pin, O: Output pin, I/O: Input/Output pin

**2. FUNCTIONS**

**2.1 Pipeline Processing**

This section describes the μPD77019-013 pipeline processing.

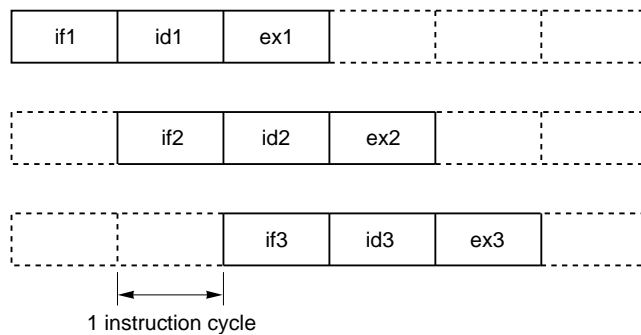
**2.1.1 Outline**

The μPD77019-013 basic operations are executed in following 3-stage pipeline.

- (1) instruction fetch; if
- (2) Instruction decoding; id
- (3) execution; ex

When the μPD77019-013 operates a result of a instruction just executed before, the data is input to ALU in parallel with written back to general registers. Pipeline processing actualizes programming without delay time to execute instructions and write back data. Three successive instructions and their processing timing are shown below.

**Pipeline Processing Timing**



**2.1.2 Instructions with Delay**

The following instructions have delay time in execution.

- (1) Instructions to control interrupt  
2 instruction cycles have been taken between instruction fetch and execution.
- (2) Inter-register transfer instructions and immediate data set instructions  
When data is set in data pointer, it needs 2 instruction cycles before the data is valid.



**2.2 Program Control Unit**

Program control unit controls not only count up of program counter in normal operation, but loop, repeat, branch, halt and interrupt.

In addition to loop stack of loop 4 level and program stack of 15 level, software stack can be used for multi-loop and multi-interrupt/subroutine call.

The μPD77019-013 has external 4 interruptions and internal 6 interruptions from peripheral, and specifies interrupt enable or disable independently.

The HALT and STOP instructions cause the μPD77019-013 to place in low power standby mode.

When the HALT instruction is executed, power consumption decreases. HALT mode is released by interrupt input or hardware reset input. It takes several system clock to recover.

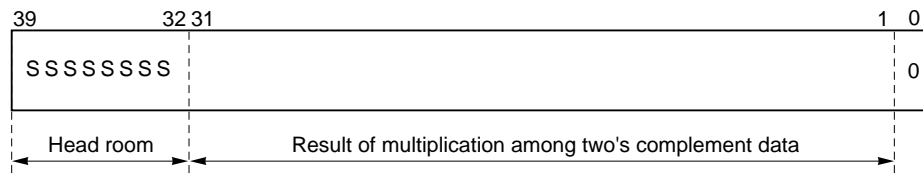
When the STOP instruction is executed, power consumption decreases. STOP mode is released by hardware reset input. It takes a few ms to recover.

**2.3 Operation Unit**

Operation unit consists of the following five parts.

- 40 bits general register × 8 for data load/store and input/output of operation data
- 16 bits × 16 bits + 40 bits → 40 bits multiply accumulator
- 40 bits Data ALU
- 40 bits barrel shifter
- SAC: shifter and count circuit.

Standard word length is 40 bits to make overflow check and adjustment easy, and to accumulate the result of 16 bits × 16 bits multiplication correctly.



**2.3.1 General register (R0 to R7)**

The μPD77019-013 has eight 40 bits registers for operation input/output and load/store with memory. General register consists of the following three parts.

- R0L to R7L (bit 15 to bit 0)
- R0H to R7H (bit 31 to bit 16)
- R0E to R7E (bit 39 to bit 32)

But each of RnL, RnH and RnE are treated as a register in the following conditions.

**(1) General register used as 40 bits register**

General registers are treated as 40 bits register, when they are used for the following aims.

- (a) Operand for triminal operation (except for multiplier input)
- (b) Operand for dyadic operation (except for multiplier and shift value)
- (c) Operand for monadic operation (except for exponent instructions)
- (d) Operand for operation
- (e) Operand for conditional judge
- (f) Destination for load instruction (with sign extension and 0 clear)

**(2) General register used as 32 bits register**

Bit 31 to bit 0 of general register are treated as 32 bits register, when it is used for a operand of exponent instruction.

**(3) General register used as 24 bits register**

Bit 39 to bit 16 of general register are treated as 24 bits register, when it is used for destination with extended sign for a load/store instruction.

**(4) General register used as 16 bits register**

Bit 31 to bit 16 of general register are treated as 16 bits register, when it is used for the following aims.

- (a) Signed operand for multiplier
- (b) Source/destination for load/store instruction

Bit 15 to bit 0 of general register are treated as 16 bits register, when it is used for the following aims.

- (c) Unsigned operand for multiplier
- (d) Shift value for shift instruction
- (e) Source/destination for load/store instruction
- (f) Source/destination for inter-register transfer instruction
- (g) Destination for immediate data set instruction
- (f) Hardware loop times

**(5) General register used as 8 bits register**

Bit 39 to bit 32 of general register are treated as 8 bits register, when it is used for source/destination of load/store instruction.

**2.3.2 MAC: MultiplY ACcumulator**

MAC multiplies a pair of 16 bits data, and adds or subtract the result and 40 bits data. MAC outputs 40 bits data.

MAC operates three types of multiplication: signed data  $\times$  signed data, signed data  $\times$  unsigned data and unsigned data  $\times$  unsigned data.

Result of multiplication and 40 bits data for addition can be added after 1 or 16 bits arithmetic shift right.

**2.3.3 ALU: Arithmetic Logic Unit**

ALU performs arithmetic operation and logic operation. Both input/output data are 40 bits.

**2.3.4 BSFT: Barrel ShiFTer**

BSFT performs shift right/left operation. Both input/output data are 40 bits. There are two types of shift right operations; arithmetic shift right which sign is extended, and logic shift right which is input 0 in MSB first.

**2.3.5 SAC: Shifter And Count Circuit**

SAC calculates and outputs shift value for normalization. SAC is input 32 bits data and outputs the 40 bits data. Then, bit 39 to bit 5 of output data is always 0.

### 2.3.6 CJC: Condition Judge Circuit

CJC judges whether condition is true or false with 40 bits input data. A conditional instruction is executed when the result is true, and not executed when the result is false.

## 2.4 Memory

The  $\mu$ PD77019-013 has one instruction memory area (64K words  $\times$  32 bits) and two data memory areas (64K words  $\times$  16 bits each). It adopts Harvard-type architecture, with instruction memory area and data memory areas separated.

The  $\mu$ PD77019-013 has 2 sets of data addressing units, which are dedicated for addressing data memory area. Each addressing unit consists of four data pointers, four index registers, a modulo register and addressing ALU.

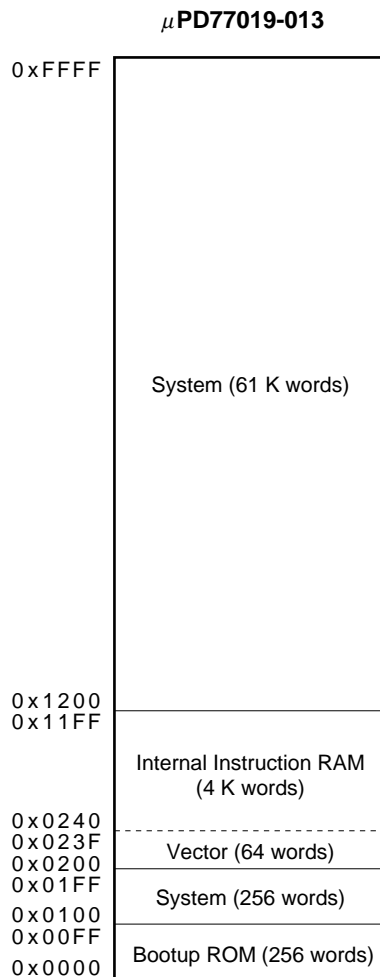
X memory area addresses are specified by DP0 to DP3, and Y memory area addresses are specified by DP4 to DP7. After memory access, DPn (with the same subscript), can be modified by DNn value. Modulo operation is performed with DMX for DP0 to DP3, with DMY for DP4 to DP7.

**2.4.1 Instruction RAM Outline**

The μPD77019-013 has an instruction RAM (4K words × 32 bits).

A system vector area is assigned to 64 words of the instruction RAM. Internal instruction RAM is initialized and rewritten by boot program.

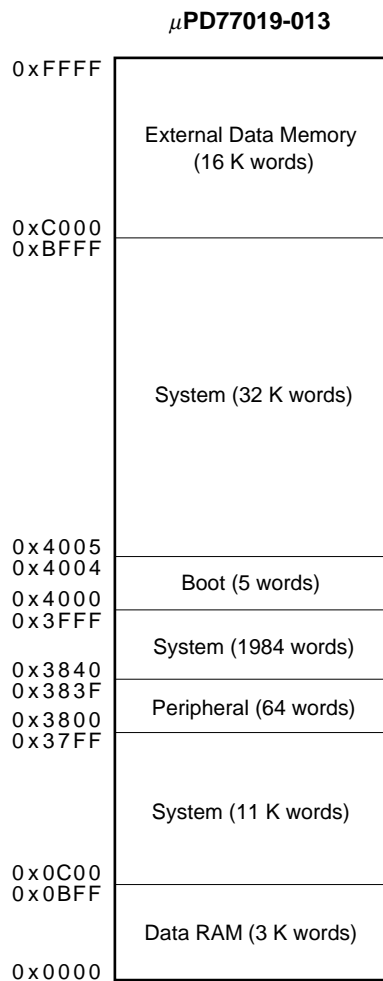
Boot up ROM contains the program loading instruction code to internal instruction RAM.



**Caution** When any data is accessed or stored to system address, normal operation of the device is not assured.

**2.4.2 Data Memory Outline**

The μPD77019-013 has two data memory areas (64K words × 16 bits each) in X and Y memory areas. Every memory areas consists of 3K words × 16 bits data RAM. As the μPD77019-013 has interface with the external data memory, 16K words × 16 bits external data memory space can be added to X/Y memories. Each data memory area includes on-chip peripheral area which consists of 64 words. When the external data memory area is accessed, instruction cycle can be 2 or more by wait function.



**Caution** When any data is accessed or stored to system address, normal operation of the device is not assured.

### 2.4.3 Data Memory Addressing

There are following two types of data memory addressing.

- Direct addressing

The address is specified in the instruction field.

- Indirect addressing

The address is specified by the data pointer (DP). DP can get a bit reverse before addressing. It can update the DP value after accessing data memory.

## 2.5 On-chip Peripheral Circuit

The μPD77019-013 includes serial interface, host interface, general input/output ports and wait cycle registers. They are mapped in both X and Y memory areas, and are accessed as memory mapped I/O by the μPD77019-013 CPU.

### 2.5.1 Serial Interface Outline

The μPD77019-013 has 2 channel serial interfaces. Serial I/O clock must be provided from external. Frame length can be programmed independently to be 8 bits or 16 bits. MSB first or LSB first can also be selected. Data is input/output by hand shaking for an external device, and by interrupts, polling or wait function in internal.

### 2.5.2 Host Interface Outline

The μPD77019-013 has 8 bits parallel ports as host interface to input/output data to and from host CPU and DMA controller. When an external device accesses host interface, HA0 and HA1 pins; which are host address input pins; specifies bit 15 to bit 8 and bit 7 to bit 0. The μPD77019-013 includes 3 registers consisting of 16 bits, which are dedicated for input data, output data and status. The μPD77019-013 has three types of interface method for internal and external data; interrupts, polling and wait function.

### 2.5.3 General Input/output Ports Outline

General input/output ports consist of 4 bits. User can set each port as input or output. The μPD77019-013 includes two registers. One is 4 bits register for input/output data, and the other is 16 bits for control.

### 2.5.4 Wait Cycle Register

The wait cycle registers consist of 16 bits. It is used to set wait cycle number when external memory is accessed. When external data memory area (0xC000 - 0xFFFF) is accessed, 0, 1, 3, or 7 wait cycle can be set.

When external data memory area is accessed, wait cycle can be also set by  $\overline{\text{WAIT}}$  pin.

## 2.6 Boot Function

There are following cautions on using boot function of the  $\mu$ PD77019-013.

The  $\mu$ PD77019-013 has self-boot mode and host boot mode.

In host boot mode, the boot code is transferred from host side; external host CPU via host interface. The boot function in host boot mode is same function as other  $\mu$ PD7701x family products have.

In self-boot mode, the boot codes are transferred from external data memory or internal data ROM to instruction RAM.

The  $\mu$ PD77019-013 executes self-boot function transferred from external data memory only. Because internal data ROM of the  $\mu$ PD77019-013 is masked already and the boot codes cannot be transferred. Other  $\mu$ PD7701x family products store their boot information in 0x4000:Y-0x4004:Y. But the user cannot define these addresses, because they are in the internal ROM of the  $\mu$ PD77019-013.

Then in the  $\mu$ PD77019-013, the following boot information is defined to Y data ROM for self-boot mode.

0x4000:Y	0	(Y memory boot/word boot)
0x4001:Y	0xC0C0	(DWTR value is set to 7 wait access mode)
0x4002:Y	0	(IWTR value is set to 0 wait access mode)
0x4003:Y	0xC000	(start address of boot code stored)
0x4004:Y	0x1000	(4K words as program step)

To execute boot function in self-boot mode, connect external Y data area with more than the 8K words 16 bits data width PROM (7 wait accessible) where the boot code should be stored.

### 3. INSTRUCTIONS

#### 3.1 Outline

All μPD77019-013 instructions are one-word instructions, consisting of 32 bits. And they are executed in 16.6 ns (min.) per instruction. There are following 9 instruction types.

**(1) Trinomial instructions**

: specify the Acc operation. 3 of general registers are specified optionally as the operation object.

**(2) Dyadic operation instructions**

: specify the Acc, ALU or shifter operation. 2 of general registers are specified optionally as the operation object. Some instructions can specify a general register and immediate data.

**(3) Monadic operation instructions**

: specify operations by ALU. 1 general register is specified optionally as the operation object.

**(4) Load/store instructions**

: transfer 16 bits data from memory to general registers, from general registers to memory and between general registers.

**(5) Inter-register transfer instructions**

: transfer data between general register and other registers.

**(6) Immediate data set instructions**

: set immediate data at general registers or each registers of address operation unit.

**(7) Branch instructions**

: specify the direction of the program flow.

**(8) Hardware loop instructions**

: specify times of instruction repeating.

**(9) Control Instructions**

: specify the control program.



### 3.2 Instruction Set and Operation

An operation is written according to the rules for expressing. An expression of instructions having two or more descriptions can have only one selected.

#### (a) Expressions and selectable registers

Expression and selectable registers are shown as follows.

Expression	Selectable registers
ro, ro', ro"	R0 - R7
rl, rl'	R0L - R7L
rh, rh'	R0H - R7H
re	R0E - R7E
reh	R0EH - R7EH
dp	DP0 - DP7
dn	DN0 - DN7
dm	DMX, DMY
dpx	DP0 - DP3
dpy	DP4 - DP7
dpx_mod	DPn, DPn++, DPn--, DPn##, DPn%%, !DPn## (n = 0 - 3)
dpy_mod	DPn, DPn++, DPn--, DPn##, DPn%%, !DPn## (n = 4 - 7)
dp_imm	DPn##imm (n = 0 - 7)
*xxx	content of memory address xxx  <b>Example</b> When the content of DP0 register is 1000, *DP0 shows the content of memory address 1000.

**(b) Modifying data pointers**

Data pointers are modified after memory access. The results are valid immediately after instruction execution. It is impossible to modify without memory access.

Description	Operation
DPn	No operation: DPn value does not change.
DPn++	$DPn \leftarrow DPn+1$
DPn--	$DPn \leftarrow DPn-1$
DPn##	DPn ← DPn + DNn: Adds DN0-DN7 corresponding to DP0-DP7 <b>Example</b> DP0 ← DP0 + DN0
DPn%%	(n = 0 - 3) $DPn = ((DPL + DNn) \bmod (DMX + 1)) + DP_H$
	(n = 4 - 7) $DPn = ((DPL + DNn) \bmod (DMY + 1)) + DP_H$
!DPn##	Access memory after DPn value is bit-reversed After memory access, $DPn \leftarrow DPn + DNn$
DPn##imm	$DPn \leftarrow DPn + imm$

**(c) Concurrent processing instructions**

○ shows concurrent processing instruction.

Instruction names are shown in abbreviation.

- TRI : Trinomial
- DYAD : Dyadic
- MONAD : Monadic
- TRANS : Inter-register transfer
- IMM : Immediate data set
- BR : Branch
- LOOP : Hardware loop
- CTR : Control

**(d) State of Overflow flag (OV)**

The following marks show the μPD77019-013 overflow flag state.

- : Not affected
- ↑: 1 is set when the result of operation is overflow.

**Caution** If overflow does not occur after operation, OV is not reset, and keeps the state before operation.

μPD77019-013 INSTRUCTION SET

	Name	Mnemonic	Operation	Concurrent Writing Processing										Flag	
				TRI.	DYAD.	MONAD.	Load/ store	TRANS.	IMM.	BR.	LOOP.	CTL.	OV		
Trinomial	Multiply add	$ro = ro + rh * rh'$	$ro \leftarrow ro + rh * rh'$				○								↕
	Multiply sub	$ro = ro - rh * rh'$	$ro \leftarrow ro - rh * rh'$				○								↕
	Sign unsign Multiply add	$ro = ro + rh * rl$ (rl should be a plus integral number.)	$ro \leftarrow ro + rh * rl$				○								↕
	Unsign unsign Multiply add	$ro = ro + rl * rl'$ (rl and rl' should be a plus integral number.)	$ro \leftarrow ro + rl * rl'$				○								↕
	1 bit shift Multiply add	$ro = (ro >> 1) + rh * rh'$	$ro \leftarrow \frac{ro}{2} + rh * rh'$				○								↕
	16 bits shift Multiply add	$ro = (ro >> 16) + rh * rh'$	$ro \leftarrow \frac{ro}{2^{16}} + rh * rh'$				○								●
Dyadic	Multiply	$ro = rh * rh'$	$ro \leftarrow rh * rh'$				○								●
	Add	$ro' = ro + ro'$	$ro' \leftarrow ro + ro'$				○								↕
	Immediate add	$ro' = ro + imm$	$ro' \leftarrow ro + imm$ (imm≠1)												↕
	Sub	$ro' = ro - ro'$	$ro' \leftarrow ro - ro'$				○								↕
	Immediate sub	$ro' = ro - imm$	$ro' \leftarrow ro - imm$ (imm≠1)												↕
	Arithmetic right shift	$ro' = ro$ SRA rl	$ro' \leftarrow ro >> rl$				○								●
	Immediate arithmetic right shift	$ro' = ro$ SRA imm	$ro' \leftarrow ro >> imm$												●
	Logic right shift	$ro' = ro$ SRL rl	$ro' \leftarrow ro >> rl$				○								●
	Immediate Logic right shift	$ro' = ro$ SRL imm	$ro' \leftarrow ro >> imm$												●
	Logic left shift	$ro' = ro$ SLL rl	$ro' \leftarrow ro << rl$				○								●
Immediate logic left shift	$ro' = ro$ SLL imm	$ro' \leftarrow ro << imm$												●	

	Name	Mnemonic	Operation	Concurrent Writing Processing										Flag	
				TRI.	DYAD.	MONAD.	Load/ store	TRANS.	IMM.	BR.	LOOP.	CTL.	OV		
Dyadic	And	ro" = ro & ro'	ro" ← ro & ro'				○								●
	Immediate and	ro' = ro & imm	ro' ← ro & imm												●
	Or	ro" = ro   ro'	ro" ← ro   ro'				○								●
	Immediate or	ro' = ro   imm	ro' ← ro   imm												●
	Exclusive or	ro" = ro ^ ro'	ro" ← ro ^ ro'				○								●
	Immediate exclusive or	ro = ro ^ imm	ro ← ro ^ imm												●
	Less than	ro" = LT(ro, ro')	if(ro<ro') {ro" ← 0x0000000001} else {ro" ← 0x0000000000}				○								●
Monadic	Clear	CLR(ro)	ro ← 0x0000000000				○						○		●
	Increment	ro' = ro + 1	ro' ← ro + 1				○						○		↕
	Decrement	ro' = ro - 1	ro' ← ro - 1				○						○		↕
	Absolute	ro' = ABS (ro)	if (ro<0) {ro' ← -ro} else {ro' ← ro}				○						○		↕
	One's complement	ro' = ~ro	ro' ← ~ro				○						○		●
	Two's complement	ro' = -ro	ro' ← -ro				○						○		↕
	Clip	ro' = CLIP (ro)	if (ro>0x007FFFFFFF) {ro' ← 0x007FFFFFFF} else if, (ro<0xFF80000000) {ro' ← 0xFF80000000} else {ro' ← ro}				○						○		↕
	Round	ro' = ROUND (ro)	if (ro>0x007FFF0000) {ro' ← 0x007FFF0000} else if, (ro>0xFF80000000) {ro' ← 0xFF80000000} else {ro' ← (ro + 0x8000) & 0xFFFFF0000}				○						○		↕
	Exponent	ro' = EXP (ro)	ro' ← log <sub>2</sub> ( $\frac{1}{ro}$ )				○						○		●
Substitution	ro' = ro	ro' ← ro				○						○		●	

	Name	Mnemonic	Operation	Concurrent Writing Processing										Flag	
				TRI.	DYAD.	MONAD.	Load/ store	TRANS.	IMM.	BR.	LOOP.	CTL.	OV		
Monadic	Cumulation	ro+ = ro	ro' ← ro'+ro				○						○	↕	
	Degression	ro'- = ro	ro' ← ro'-ro				○						○	↕	
	Division	ro/ = ro	if (sign(ro')==sign(ro)) {ro' ← (ro'-ro)<<1} else {ro' ← (ro'+ro)<<1} if (sign(ro')==0 {ro' ← ro'+1}				○						○	↕	
Load/store	Parallel load/store Note 1, Note 2	ro=*dpx_mod ro'=*dpy_mod	ro ← *dpx, ro' ← *dpy	○	○	○								●	
		ro=*dpx_mod *dpy_mod=rh	ro ← *dpx, *dpy ← rh												
		*dpx_mod=rh ro=*dpy_mod	*dpx ← rh, ro ← *dpy												
		*dpx_mod=rh *dpy_mod=rh'	*dpx ← rh, *dpy ← rh'												
	Section load/store Note 1, Note 2, Note 3	dest=*dpx_mod dest'=*dpy_mod	dest ← *dpx, dest' ← *dpy												●
		dest=*dpx_mod *dpy_mod=source	dest ← *dpx, *dpy ← source												
		*dpx_mod=source dest=*dpy_mod	*dpx ← source, dest ← *dpy												
		*dpx_mod=source *dpy_mod=source'	*dpx ← source, *dpy ← source'												

- Notes**
- One or both of a mnemonic pair can be written.
  - After execution of load/store, data is modified by mod.
  - One of following mnemonic should be selected: dest, dest' = {ro, reh, re, rh, rl}, source, source' = {re, rh, rl}.

	Name	Mnemonic	Operation	Concurrent Writing Processing										Flag			
				TRI.	DYAD.	MONAD.	Load/ store	TRANS.	IMM.	BR.	LOOP.	CTL.	OV				
Load/store	Direct addressing load/store <b>Note 1</b>	dest = *addr	dest ← *addr													●	
		*addr = source	*addr ← source														
	Immediate index load/store <b>Note 2</b>	dest = *dp_imm	dest ← *dp														●
		*dp_imm = source	*dp ← source														
Inter-register transfer	Inter-register transfer <b>Note 3</b>	dest = rl	dest ← rl													●	
		rl = source	rl ← source											○			●
Immediate data set	Immediate data set	rl = imm (provided imm = 0-0xFFFF)	rl ← imm														
		dp = imm (provided imm = 0-0xFFFF)	dp ← imm														
		dn = imm (provided imm = 0-0xFFFF)	dn ← imm														
		dm = imm (provided imm = 1-0xFFFF)	dm ← imm														●

- Notes**
- One of following mnemonic should be selected: dest = {ro, reh, re, rh, rl}, source = {re, rh, rl}, add =  $\left\{ \begin{array}{l} 0: X-0xFFFF:X \text{ memory} \\ 0: Y-0xFFFF:Y \text{ memory} \end{array} \right\}$ .
  - One of following mnemonic should be selected: dest = {ro, reh, re, rh, rl}, source = {re, rh, rl}.
  - Any register except general registers should be selected as dest or source.

	Name	Mnemonic	Operation	Concurrent Writing Processing										Flag	
				TRI.	DYAD.	MONAD.	Load/ store	TRANS.	IMM.	BR.	LOOP.	CTL.	OV		
Branch	Jump	JMP imm	PC ← imm											○	●
	Inter-register indirect jump	JMP dp	PC ← dp											○	●
	Subroutine call	CALL imm	SP ← SP + 1 STK ← PC + 1 PC ← imm											○	●
	Inter-register indirect subroutine call	CALL dp	SP ← SP + 1 STK ← PC + 1 PC ← dp											○	●
	Return	RET	PC ← STK SP ← SP - 1											○	●
	Return from interrupt	RETI	PC ← STK STK ← SP - 1 Restore the interrupt enable flag											○	●
Hardware loop	Repeat	REP count	start RC ← count RF ← 0 repeat PC ← PC RC ← RC - 1 end PC ← PC + 1 RF ← 1												●
	Loop	LOOP count (Mnemonics more than two lines)	start RC ← count RF ← 0 repeat PC ← PC RC ← RC - 1 end PC ← PC + 1 RF ← 1												●
	Loop pop	LPOP	LC ← LSR3 LE ← LSR2 LS ← LSR1 LSP ← LSP-1												●
Control	No operation	NOP	PC ← PC + 1												●
	Halt	HALT	CPU stop <b>Note1</b>												●
	Stop	STOP	CPU, PLL, OSC Stop <b>Note2</b>												●
	If	IF (ro cond)	Conditional judge			○		○		○					●
	Forget interrupt	FINT	Forget interrupt requests												●

- Notes 1.** The HALT instruction causes all function except for clock and PLL to halt. The system is placed in much less power consumption mode.  
The contents of internal registers and memories are maintained.  
HALT is released by interrupt input. It takes several system clock to recover.
- 2.** The STOP instruction causes all function including clock and PLL to stop. The system is placed in a minimum-power consumption mode.  
The contents of internal registers and memories are not maintained.  
After the STOP instruction is executed, pin status is maintained.  
STOP is released by hardware reset. It takes a few ms to recover.



4. ELECTRICAL SPECIFICATIONS

**Absolute Maximum Ratings (T<sub>A</sub> = +25 °C)**

Parameters	Symbol	Conditions	Ratings	Unit
Power supply voltage	V <sub>DD</sub>		-0.5 to +4.6	V
Input voltage	V <sub>I</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 3.6 V	-0.5 to +4.1 V <sub>I</sub> < V <sub>DD</sub> +0.5 V	V
Output voltage	V <sub>O</sub>		-0.5 to +4.6	V
Storage temperature	T <sub>stg</sub>		-65 to +150	°C
Operating ambient temperature	T <sub>A</sub>		-40 to +85	°C

**Caution** Exposure to Absolute Maximum Ratings for extended periods may affect device reliability; exceeding the ratings could cause permanent damage. The parameters apply independently. The device should be operated within the limits specified under DC and AC Characteristics.

**Recommended Operating Conditions**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating voltage	V <sub>DD</sub>	t <sub>cC</sub> ≥ 20.0 ns	2.7	3.0	3.6	V
		t <sub>cC</sub> ≥ 16.6 ns	3.0	3.3	3.6	V
Input voltage	V <sub>I</sub>		0		V <sub>DD</sub>	V

**Capacitance (T<sub>A</sub> = +25 °C, V<sub>DD</sub> = 0 V)**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C <sub>I</sub>	f = 1 MHz Unmeasured pins returned to 0 V.		10		pF
Output capacitance	C <sub>O</sub>			10		pF
Input/output capacitance	C <sub>IO</sub>			10		pF

DC Characteristics (T<sub>A</sub> = -40 to +85 °C, V<sub>DD</sub> = 2.7 to 3.6 V)

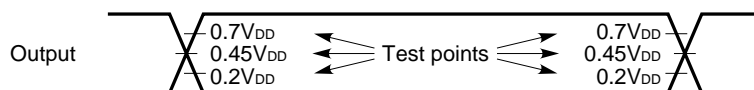
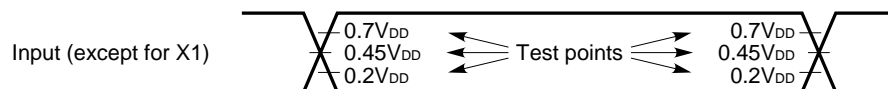
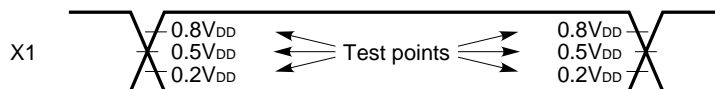
Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
High level input voltage	V <sub>IH</sub>	Except for X1	0.7V <sub>DD</sub>		V <sub>DD</sub>	V
High level X1 input voltage	V <sub>IHC</sub>	X1 input	0.8V <sub>DD</sub>		V <sub>DD</sub>	V
Low level input voltage	V <sub>IL</sub>		0		0.2V <sub>DD</sub>	V
High level output voltage	V <sub>OH</sub>	I <sub>OH</sub> = -2.0 mA	0.7V <sub>DD</sub>			V
		I <sub>OH</sub> = -100 μA	0.8V <sub>DD</sub>			V
Low level output voltage	V <sub>OL</sub>	I <sub>OL</sub> = 2.0 mA			0.2V <sub>DD</sub>	V
High level input leak current	I <sub>LIH</sub>	Except for TDI, TMS, V <sub>I</sub> = V <sub>DD</sub>			10	μA
Low level input leak current	I <sub>LIL</sub>	Except for TDI, TMS, V <sub>I</sub> = 0 V			-10	μA
Pull-up pin current	I <sub>PI</sub>	TDI, TMS, 0 V ≤ V <sub>I</sub> ≤ V <sub>DD</sub>			-250	μA
Power supply current	I <sub>DD</sub>	Active mode, t <sub>CC</sub> = 25 ns, V <sub>IH</sub> = V <sub>DD</sub> , V <sub>IL</sub> = 0 V, no load		<b>TBD</b>	150 <sup>Note 1</sup>	mA
	I <sub>DDH</sub>	HALT mode, t <sub>CC</sub> = 200 ns, V <sub>IH</sub> = V <sub>DD</sub> , V <sub>IL</sub> = 0 V, no load			15 <sup>Note 2</sup>	mA
	I <sub>DDs</sub>	STOP mode, T <sub>A</sub> = +60°C, V <sub>IH</sub> = V <sub>DD</sub> , V <sub>IL</sub> = 0 V, no load			100	μA

**Notes 1.** The MAX. value is measured when a special program that MAX. switching required is executed, and V<sub>DD</sub> = 3.6 V condition.

You can convert by each operation frequency f[MHz] on by each power supply V<sub>DD</sub>[V].  
 Max. current value (T<sub>A</sub> = +25°C) = (1.40 × V<sub>DD</sub> - 1.74) × f + 18 [mA]

- 2.** This value is measured when V<sub>DD</sub> = 3.6 V condition.  
 You can convert by each operation frequency f[MHz] on by each power supply V<sub>DD</sub>[V].  
 HALT current value (T<sub>A</sub> = +25°C) = (0.17 × V<sub>DD</sub> - 0.38) × f + 5.9 [mA]

AC Timing Test Points



AC Characteristics (T<sub>A</sub> = -40 to +85 °C, V<sub>DD</sub> = 2.7 to 3.6 V)

**Clock**

**Required Timing Condition (V<sub>DD</sub> = 2.7 V to 3.6 V)**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CLKIN cycle time	t <sub>cCX</sub>	PLL multiple rate: 4 (fixed)	80		160	ns
CLKIN high level width	t <sub>wCXH</sub>		8		t <sub>cCX</sub> - 8 - 2t <sub>riCX</sub> <sup>Note</sup>	ns
CLKIN low level width	t <sub>wCXL</sub>		8		t <sub>cCX</sub> - 8 - 2t <sub>riCX</sub> <sup>Note</sup>	ns
CLKIN rise/fall time	t <sub>riCX</sub>				15	ns

**Note** 0.5t<sub>cCX</sub> - t<sub>riCX</sub> ≥ 8 (MIN.)

**Required Timing Condition (V<sub>DD</sub> = 3.0 V to 3.6 V)**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CLKIN cycle time	t <sub>cCX</sub>	PLL multiple rate: 4 (fixed)	66.6		160	ns
CLKIN high level width	t <sub>wCXH</sub>		6.5		t <sub>cCX</sub> - 6.5 - 2t <sub>riCX</sub> <sup>Note</sup>	ns
CLKIN low level width	t <sub>wCXL</sub>		6.5		t <sub>cCX</sub> - 6.5 - 2t <sub>riCX</sub> <sup>Note</sup>	ns
CLKIN rise/fall time	t <sub>riCX</sub>				15	ns

**Note** 0.5t<sub>cCX</sub> - t<sub>riCX</sub> ≥ 6.5 (MIN.)

**Switching Characteristics**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Internal clock cycle time	t <sub>cC</sub>	Active mode		t <sub>cCX</sub> /N <sup>Note</sup>		ns
		HALT mode		8t <sub>cCX</sub> /N <sup>Note</sup>		ns
CLKOUT cycle time	t <sub>cCO</sub>			t <sub>cC</sub>		ns
CLKOUT level width	t <sub>wCO</sub>		0.5t <sub>cCO</sub> - 5			ns
CLKOUT rise/fall time	t <sub>riCO</sub>				5	ns

**Note** N: PLL multiple rate (N = 4)

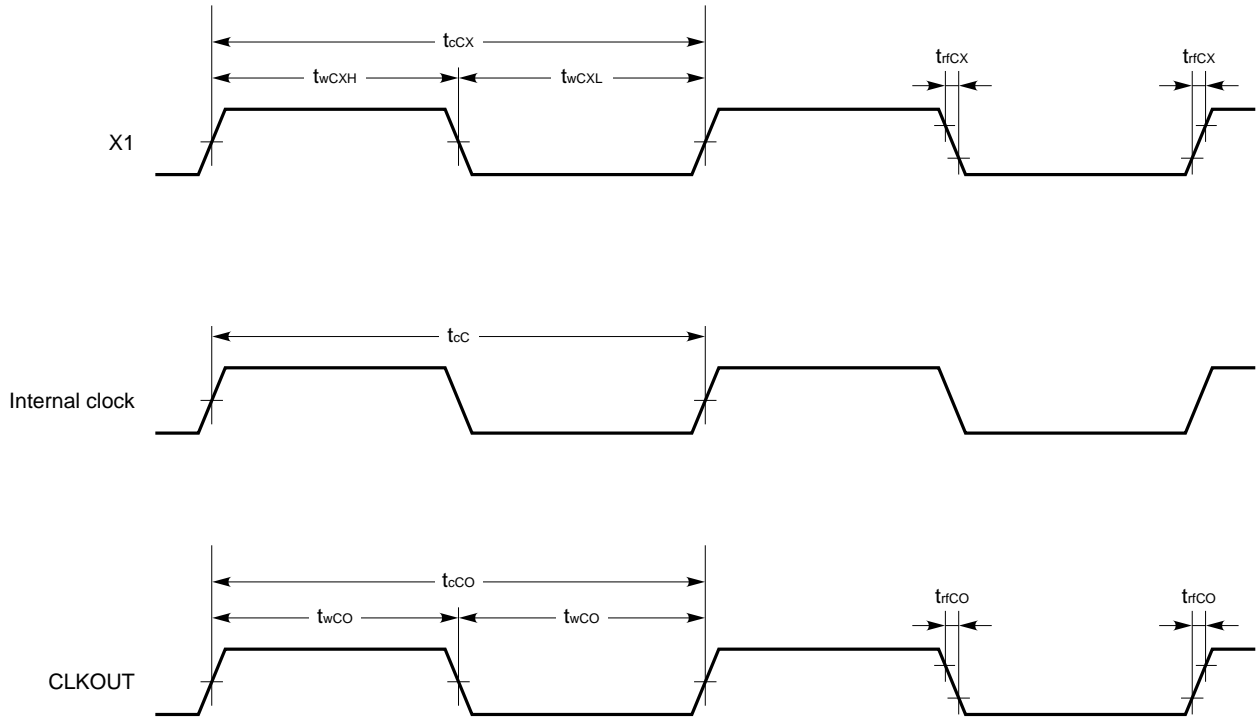
**Reset, Interrupt**

**Required Timing Condition**

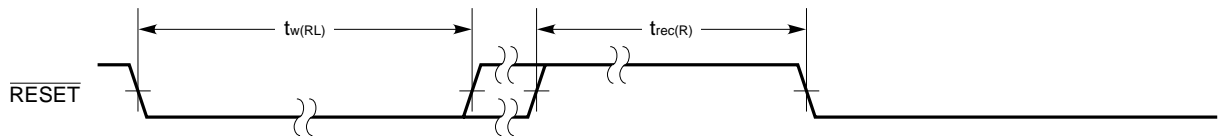
Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
RESE $\bar{T}$ low level width	$t_{w(RL)}$	External clock is input, at power on or STOP mode	100 <sup>Note 1</sup>			μs
		Active mode or HALT mode	4 $t_{cC}$ <sup>Note 2</sup>			ns
RESE $\bar{T}$ recovery time	$t_{rec(R)}$		4 $t_{cC}$			ns
$\overline{INT1}$ - $\overline{INT4}$ low level width	$t_{w(INTL)}$		3 $t_{cC}$ <sup>Note 2</sup>			ns
$\overline{INT1}$ - $\overline{INT4}$ recovery time	$t_{rec(INT)}$		3 $t_{cC}$			ns

- Notes 1.** The  $t_{w(RL)}$  indicates a time between oscillator starts to provide clock and PLL becomes stable. The  $t_{w(RL)}$  depends on the rating of oscillator. At power on, the  $t_{w(RL)}$  is measured after the point that power supply voltage reaches to 2.7V.
- 2.** Note that, during HALT mode,  $t_{cC}$  is extended to 8 times as long as that of Active mode.

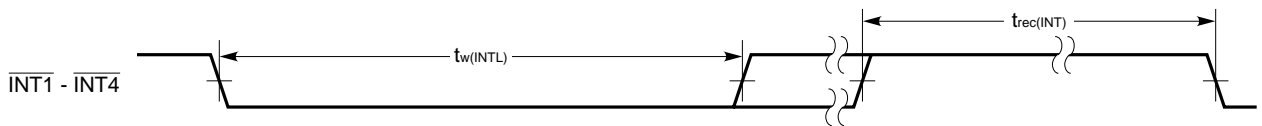
**Clock Input/Output Timing**



**Reset Timing**



**Interrupt Timing**



External Data Memory Access

Required Timing Condition

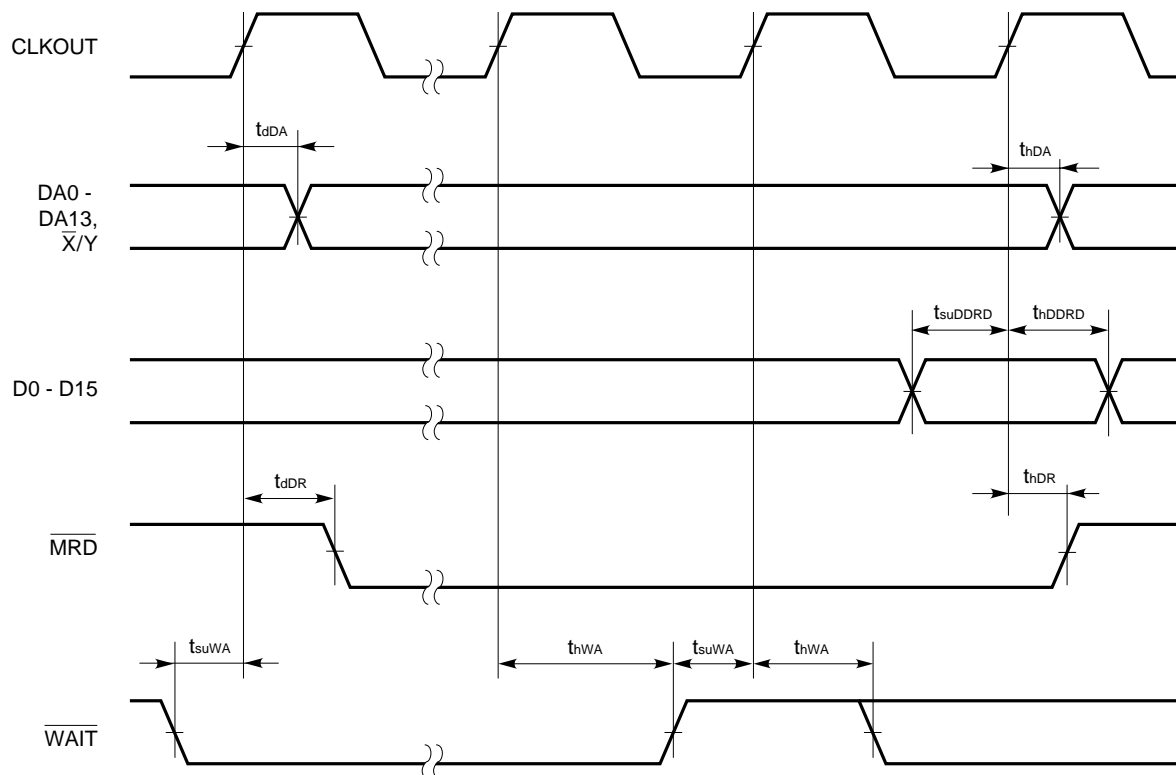
Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Read data setup time	$t_{suDDR D}$		15			ns
Read data hold time	$t_{hDDR D}$		0			ns
$\overline{WAIT}$ setup time	$t_{suWA}$		12			ns
$\overline{WAIT}$ hold time	$t_{hWA}$		0			ns

Switching Characteristics

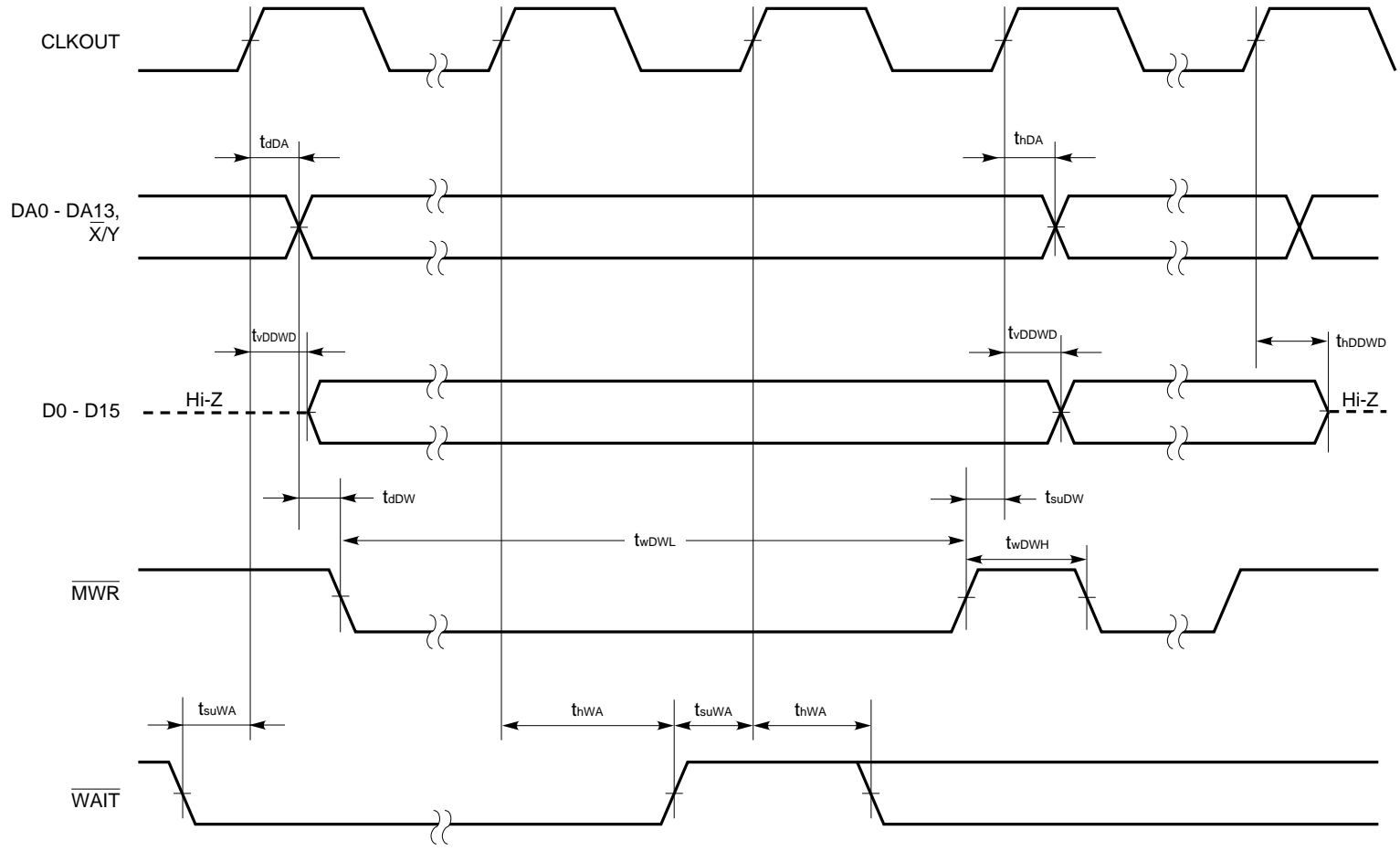
Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Address output delay time	$t_{dDA}$				8	ns
Address output hold time	$t_{hDA}$		0			ns
$\overline{MRD}$ output delay time	$t_{dDR}$				8	ns
$\overline{MRD}$ hold time	$t_{hDR}$		0			ns
Write data output valid time	$t_{vDDWD}$				16	ns
Write data output hold time	$t_{hDDWD}$		0			ns
$\overline{MWR}$ output delay time	$t_{dDW}$		$0.25t_{cC} - 5$			ns
$\overline{MWR}$ setup time	$t_{suDW}$		0			ns
$\overline{MWR}$ low level width	$t_{wDWL}$		$0.5t_{cC} - 3$ + $t_{cDW}$ <b>Note</b>			ns
$\overline{MWR}$ high level width	$t_{wDWH}$		$0.5t_{cC} - 5$			ns

**Note**  $t_{cDW}$ : Data wait cycle

External Data Memory Access Timing (Read)



External Data Memory Access Timing (Write)



**Bus Arbitration****Required Timing Condition**

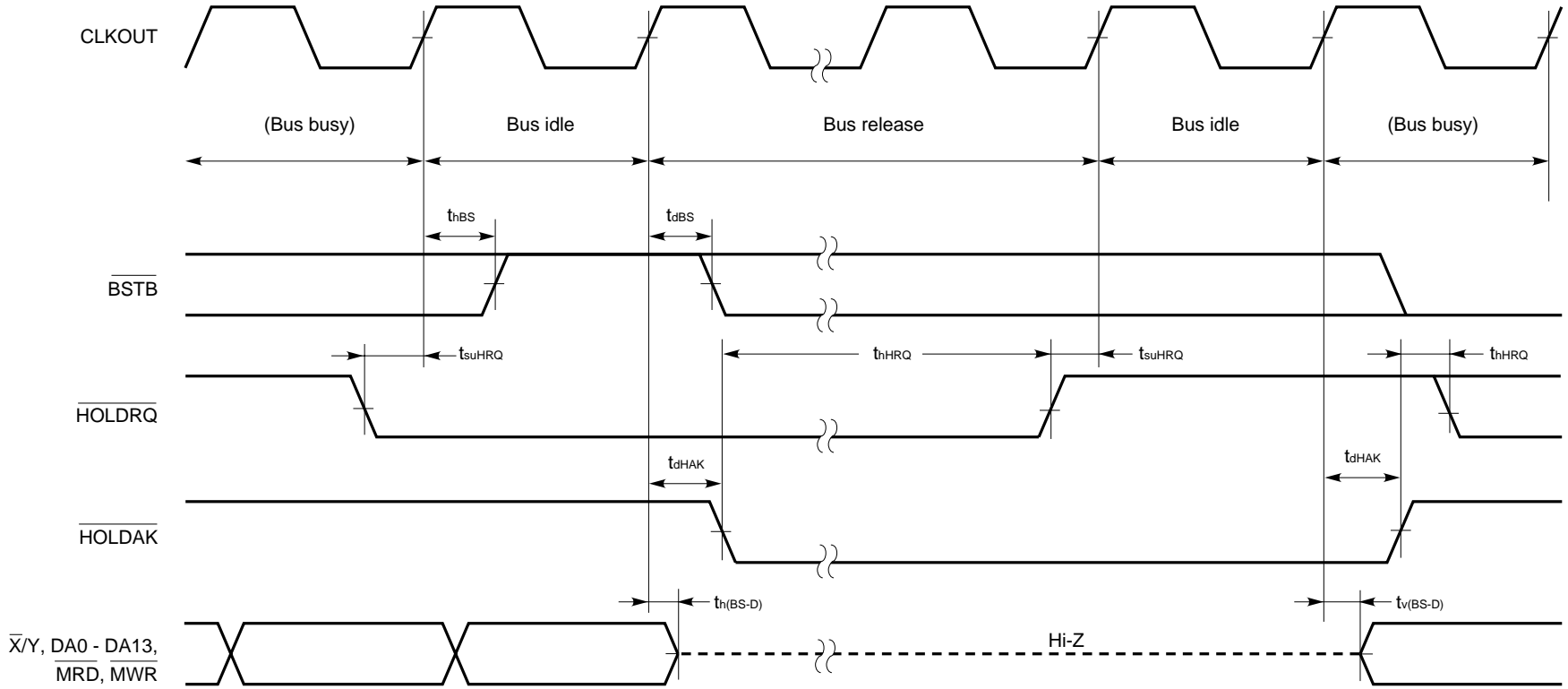
Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{HOLDRQ}}$ setup time	$t_{\text{suHRQ}}$		12			ns
$\overline{\text{HOLDRQ}}$ hold time	$t_{\text{hHRQ}}$		0			ns

**Switching Characteristics**

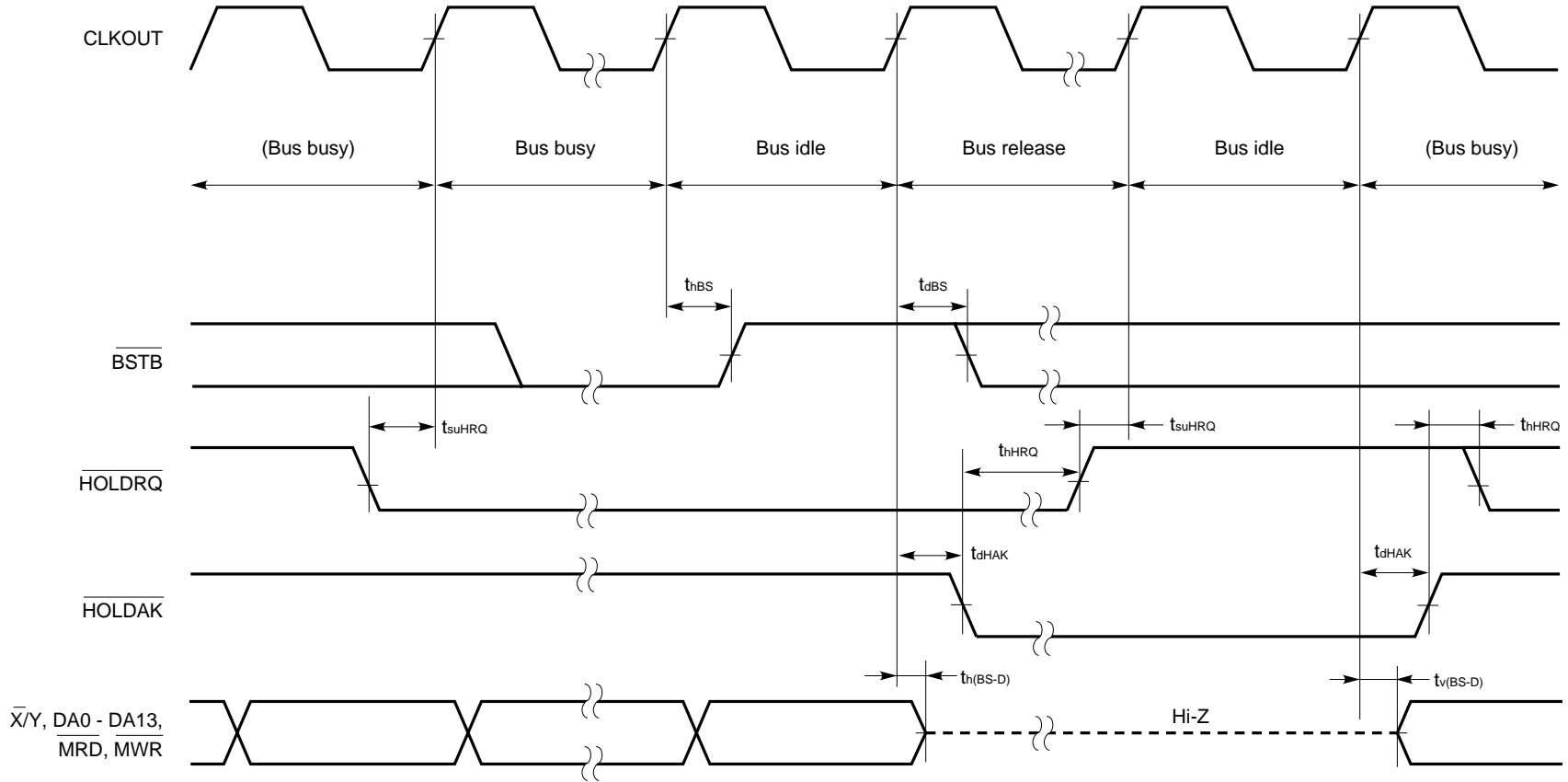
Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{BSTB}}$ hold time	$t_{\text{hBS}}$		0			ns
$\overline{\text{BSTB}}$ output delay time	$t_{\text{dBS}}$				12	ns
$\overline{\text{HOLDAK}}$ output delay time	$t_{\text{dHAK}}$				12	ns
Data hold time when bus arbitration	$t_{\text{h(BS-D)}}$				30	ns
Data valid time after bus arbitration	$t_{\text{v(BS-D)}}$				15	ns



Bus Arbitration Timing (Bus idle)



Bus Arbitration Timing (Bus busy)



**Serial Interface**

**Required Timing Condition**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCK input cycle time	$t_{cSC}$		$2t_{cC}$			ns
SCK input high/low level width	$t_{wSC}$		25			ns
SCK input rise/fall time	$t_{rSC}$				20	ns
SOEN recovery time	$t_{recSOE}$		20			ns
SOEN hold time	$t_{hSOE}$		0			ns
SIEN recovery time	$t_{recSIE}$		20			ns
SIEN hold time	$t_{hSIE}$		0			ns
SI setup time	$t_{suSI}$		20			ns
SI hold time	$t_{hSI}$		0			ns

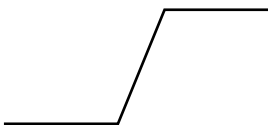
**Switching Characteristics**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SORQ output delay time	$t_{dSOR}$				30	ns
SORQ hold time	$t_{hSOR}$		0			ns
SO valid time	$t_{vSO}$				30	ns
SO hold time	$t_{hSO}$		0			ns
SI AK output delay time	$t_{dSIA}$				30	ns
SI AK hold time	$t_{hSIA}$		0			ns

**Notes for Serial Clock**

Serial clock inputs SCK1 and SCK2 are sensitive to any kind of interfering signals (noise on power supply, induced voltage, etc.). Spurious signals can cause malfunction of the device. Special care for the serial clock design should be taken. Careful grounding, decoupling and short wiring of SCK1 and SCK2 are recommended. Intersection of SCK1 and SCK2 with other serial interface lines or close wiring to lines carrying high frequency signals or large changing currents should be avoided.

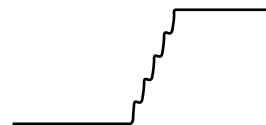
It considers for the serial clock to make a waveform stable especially about the rising and falling.



**Example 1. good example**  
Straight rising form and falling form

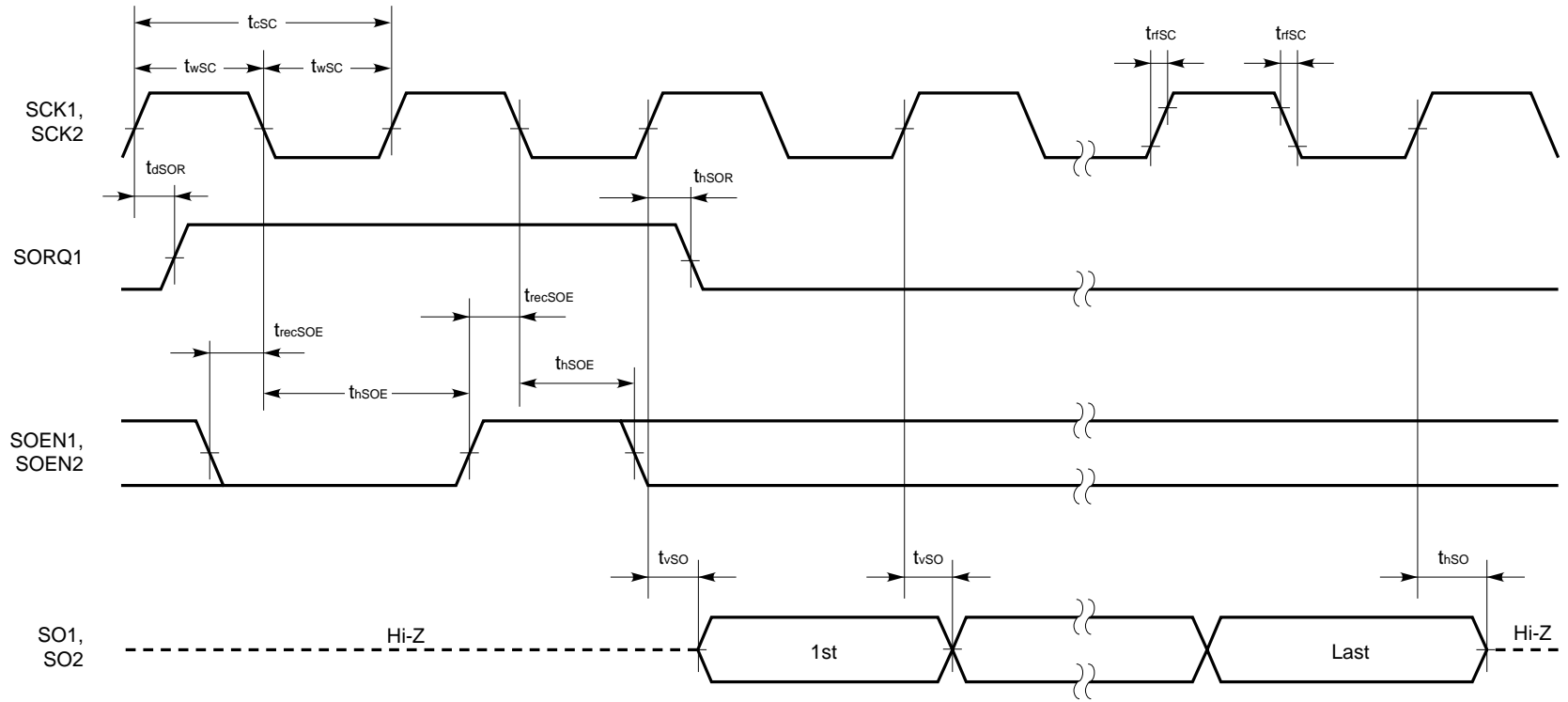


**Example 2. no good example**  
It doesn't bound. It doesn't make noise one above another.

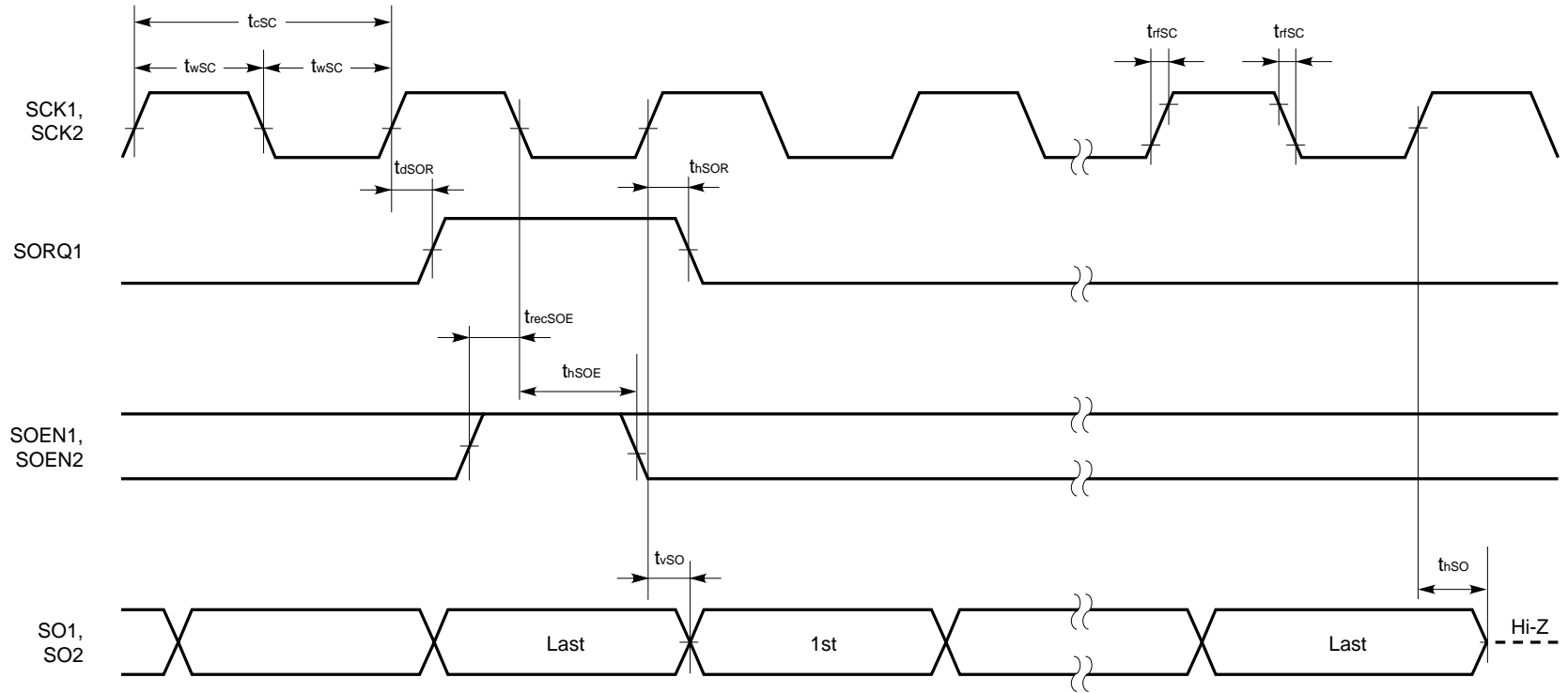


**Example 3. no good example**  
It doesn't make a stair stepping.

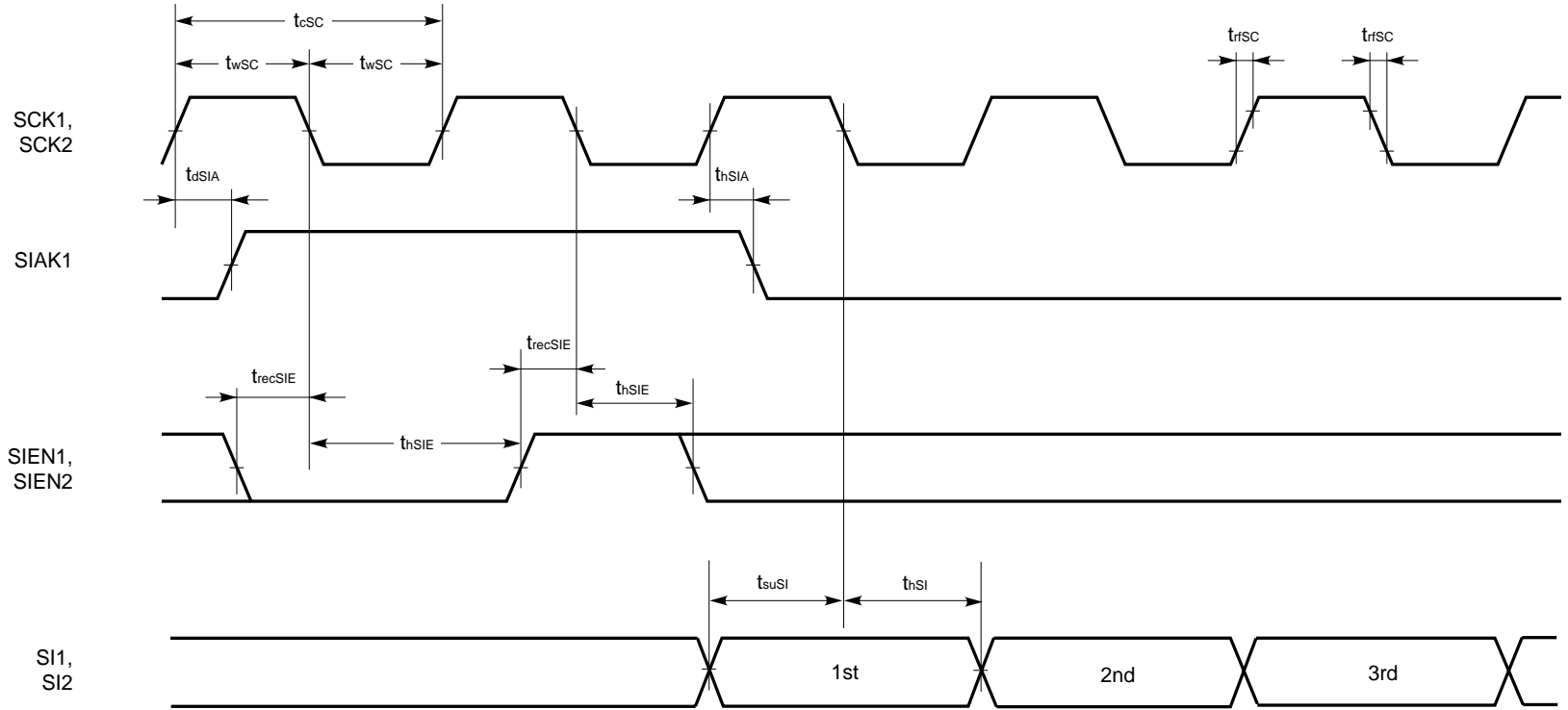
Serial Output Timing 1



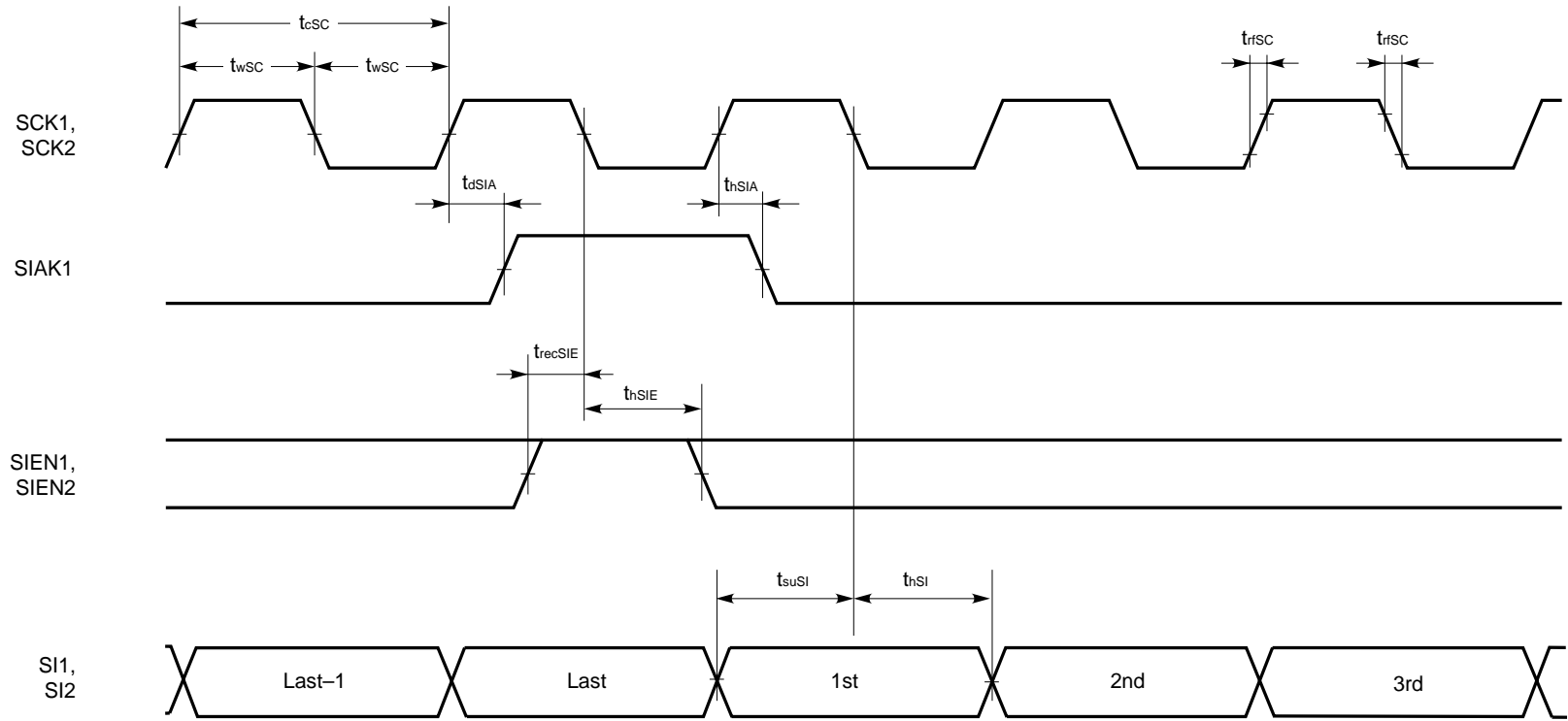
Serial Output Timing 2 (Continual output)



Serial Input Timing 1



Serial Input Timing 2 (Continual input)



**Host Interface**

**Required Timing Condition**

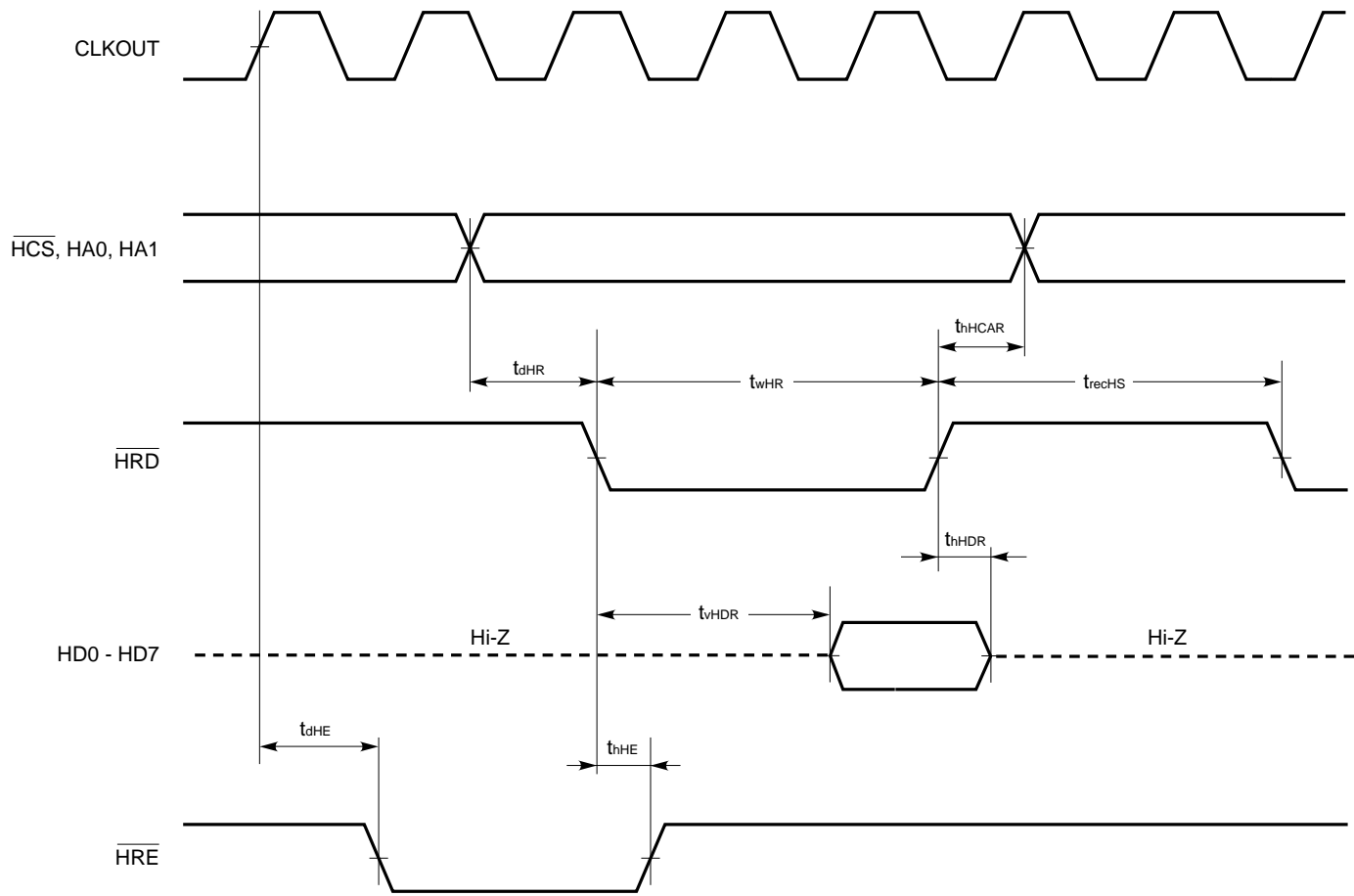
Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{HRD}}$ delay time	$t_{\text{dHR}}$		0			ns
$\overline{\text{HRD}}$ width	$t_{\text{wHR}}$		$2t_{\text{cC}}$			ns
$\overline{\text{HCS}}$ , HA0, HA1 read hold time	$t_{\text{hCAR}}$		0			ns
$\overline{\text{HCS}}$ , HA0, HA1 write hold time	$t_{\text{hCAW}}$		0			ns
$\overline{\text{HRD}}$ , $\overline{\text{HWR}}$ recovery time	$t_{\text{recHS}}$		$2t_{\text{cC}}$			ns
$\overline{\text{HWR}}$ delay time	$t_{\text{dHW}}$		0			ns
$\overline{\text{HWR}}$ width	$t_{\text{wHW}}$		$2t_{\text{cC}}$			ns
$\overline{\text{HWR}}$ hold time	$t_{\text{hHDW}}$		0			ns
$\overline{\text{HWR}}$ setup time	$t_{\text{suHDW}}$		20			ns

**Switching Characteristics**

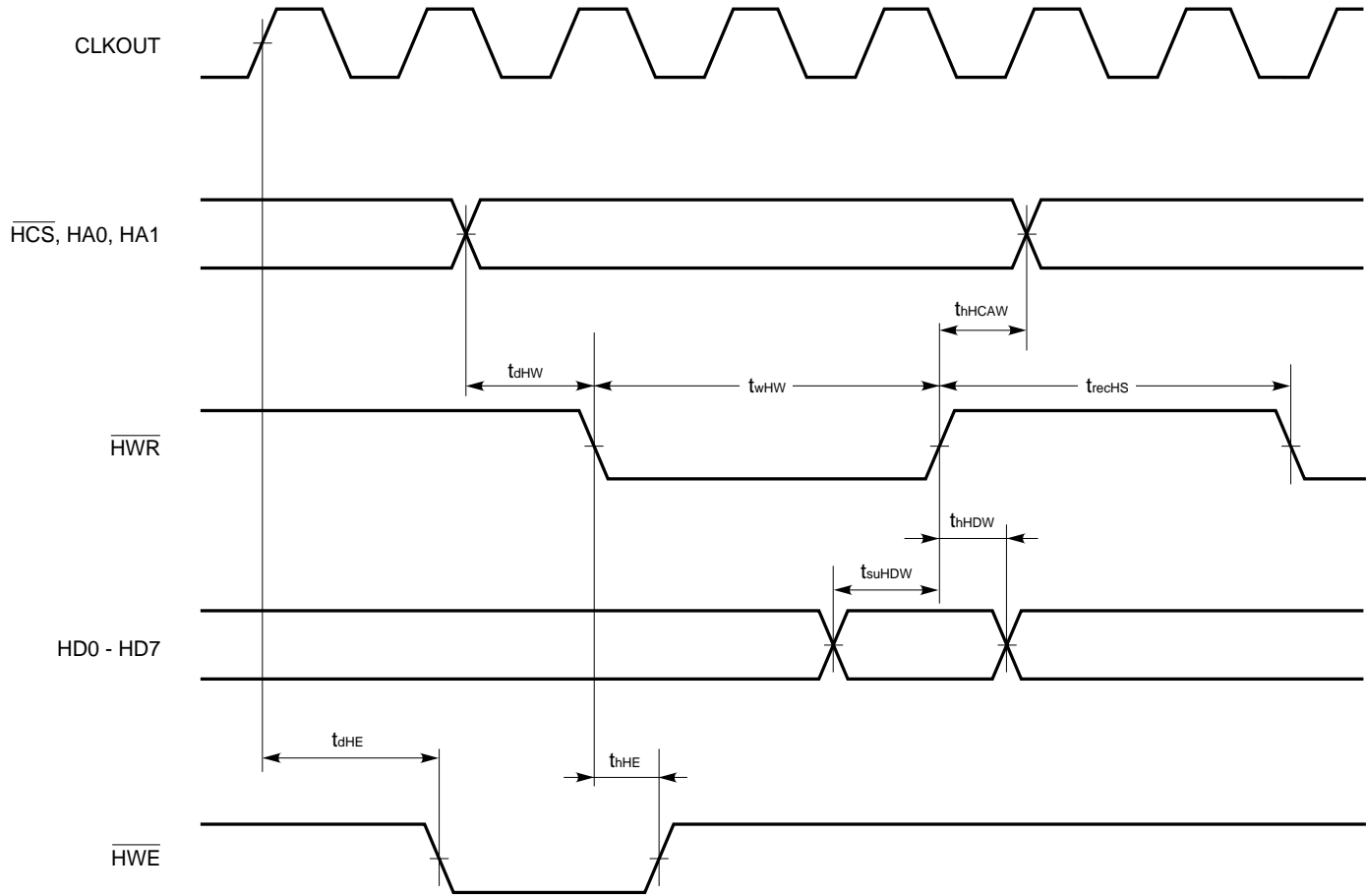
Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{HRE}}$ , $\overline{\text{HWE}}$ output delay time	$t_{\text{dHE}}$				30	ns
$\overline{\text{HRE}}$ , $\overline{\text{HWE}}$ hold time	$t_{\text{hHE}}$				30	ns
$\overline{\text{HRD}}$ valid time	$t_{\text{vHDR}}$				30	ns
$\overline{\text{HRD}}$ hold time	$t_{\text{hHDR}}$		0			ns



Host Interface Timing (Read)



Host Interface Timing (Write)



**General Input/Output Ports**

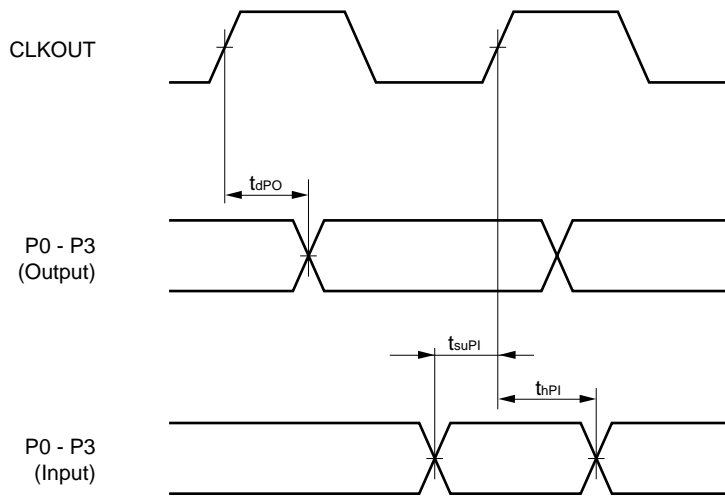
**Required Timing Condition**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Port input setup time	$t_{suPI}$		20			ns
Port input hold time	$t_{hPI}$		10			ns

**Switching Characteristics**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Port output delay time	$t_{dPO}$		0		30	ns

**General Input/Output Ports Timing**



**Debugging Interface (JTAG)**

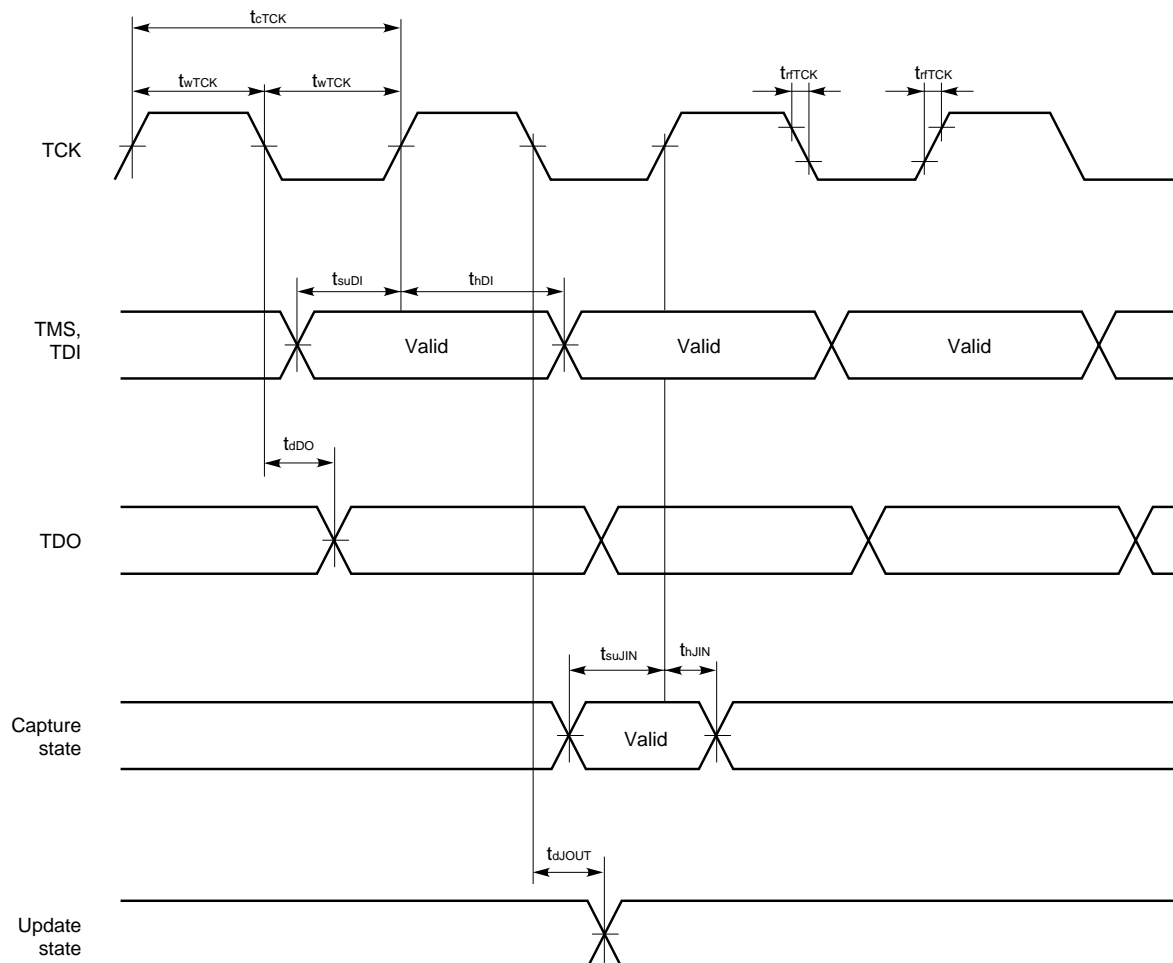
**Required Timing Condition**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
TCK cycle time	$t_{cTCK}$		$4t_{cC}$			ns
TCK high/low level width	$t_{wTCK}$		50			ns
TCK rise/fall time	$t_{rTCK}$				20	ns
TMS, TDI setup time	$t_{suDI}$		10			ns
TMS, TDI hold time	$t_{hDI}$		0			ns
Input pin setup time	$t_{suJIN}$		10			ns
Input pin hold time	$t_{hJIN}$		0			ns

**Switching Characteristics**

Parameters	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
TDO output delay time	$t_{dDO}$				30	ns
Output pin output delay time	$t_{dJOUT}$				30	ns

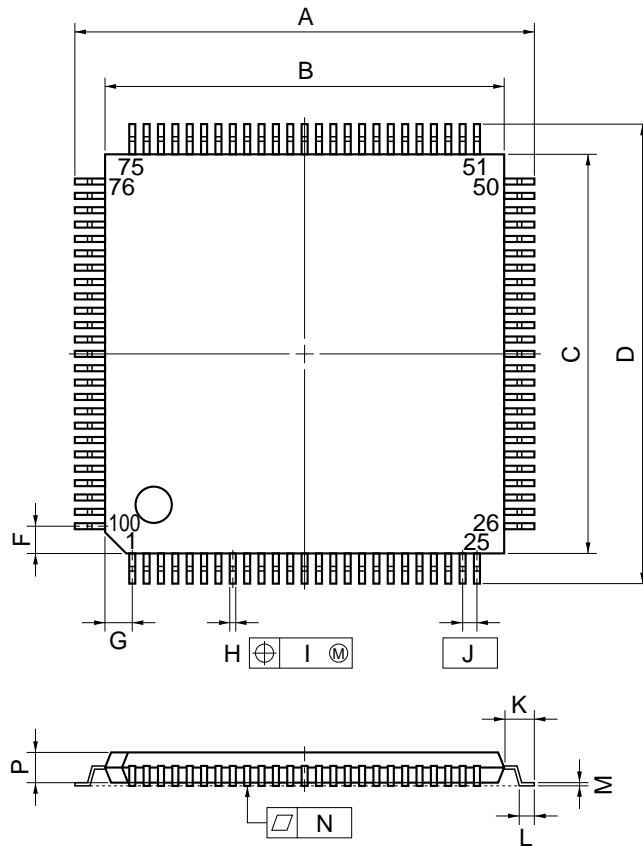
**Debugging Interface Timing**



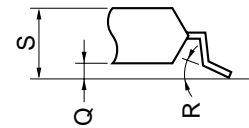
**Remark** For the details of JTAG, refer to "IEEE1149.1."

5. PACKAGE DRAWING

100 PIN PLASTIC TQFP (FINE PITCH) (□14)



detail of lead end



NOTE

Each lead centerline is located within 0.10 mm (0.004 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	16.0±0.2	0.630±0.008
B	14.0±0.2	0.551 <sup>+0.009</sup> <sub>-0.008</sub>
C	14.0±0.2	0.551 <sup>+0.009</sup> <sub>-0.008</sub>
D	16.0±0.2	0.630±0.008
F	1.0	0.039
G	1.0	0.039
H	0.22 <sup>+0.05</sup> <sub>-0.04</sub>	0.009±0.002
I	0.10	0.004
J	0.5 (T.P.)	0.020 (T.P.)
K	1.0±0.2	0.039 <sup>+0.009</sup> <sub>-0.008</sub>
L	0.5±0.2	0.020 <sup>+0.008</sup> <sub>-0.009</sub>
M	0.145 <sup>+0.055</sup> <sub>-0.045</sub>	0.006±0.002
N	0.10	0.004
P	1.0±0.1	0.039 <sup>+0.005</sup> <sub>-0.004</sub>
Q	0.1±0.05	0.004±0.002
R	3° <sup>+7°</sup> <sub>-3°</sub>	3° <sup>+7°</sup> <sub>-3°</sub>
S	1.27 MAX.	0.050 MAX.

S100GC-50-9EU-1

**6. RECOMMENDED SOLDERING CONDITIONS**

When soldering this product, it is highly recommended to observe the conditions as shown below. If other soldering processes are used, or if the soldering is performed under different conditions, please make sure to consult with our sales offices.

For more details, refer to our document “**SEMICONDUCTOR DEVICE MOUNTING TECHNOLOGY MANUAL**” (C10535E).

**Surface Mount Device**

**μPD77019GC-013-9EU: 100-pin plastic TQFP (FINE PITCH) (14 × 14mm)**

Process	Conditions	Symbol
Infrared ray reflow	Peak temperature: 235 °C or below (Package surface temperature), Reflow time: 30 seconds or less (at 210 °C or higher), Maximum number of reflow processes : 2 times, Exposure limit <b>Note</b> : 3 days (10 hours pre-baking is required at 125 °C afterwards).	IR35-103-2
Vapor Phase Soldering	Peak temperature: 215 °C or below (Package surface temperature), Reflow time: 40 seconds or less (at 200 °C or higher), Maximum number of reflow processes : 2 times, Exposure limit <b>Note</b> : 3 days (10 hours pre-baking is required at 125 °C afterwards).	VP15-103-2
Partial heating method	Pin temperature : 300 °C or below, Heat time : 3 seconds or less (Per each side of the device)	—

**Note** Maximum allowable time from taking the soldering package out of dry pack to soldering.  
Storage conditions: 25 °C and relative humidity of 65 % or less.

**Caution** Apply only one kind of soldering condition to a device, except for “partial heating method”, or the device will be damaged by heat stress.

## NOTES FOR CMOS DEVICES

### ① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

**Note:** Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

### ② HANDLING OF UNUSED INPUT PINS FOR CMOS

**Note:** No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to  $V_{DD}$  or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

### ③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

**Note:** Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

[MEMO]

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.

While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.

NEC devices are classified into the following three quality grades:

"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots

Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)

Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

Anti-radioactive design is not implemented in this product.