

**16 bits, Fixed-point Digital Signal Processor**

$\mu$ PD77016 is a 16 bits fixed-point DSP (Digital Signal Processor) developed for digital signal processing with its demand for high speed and precision.

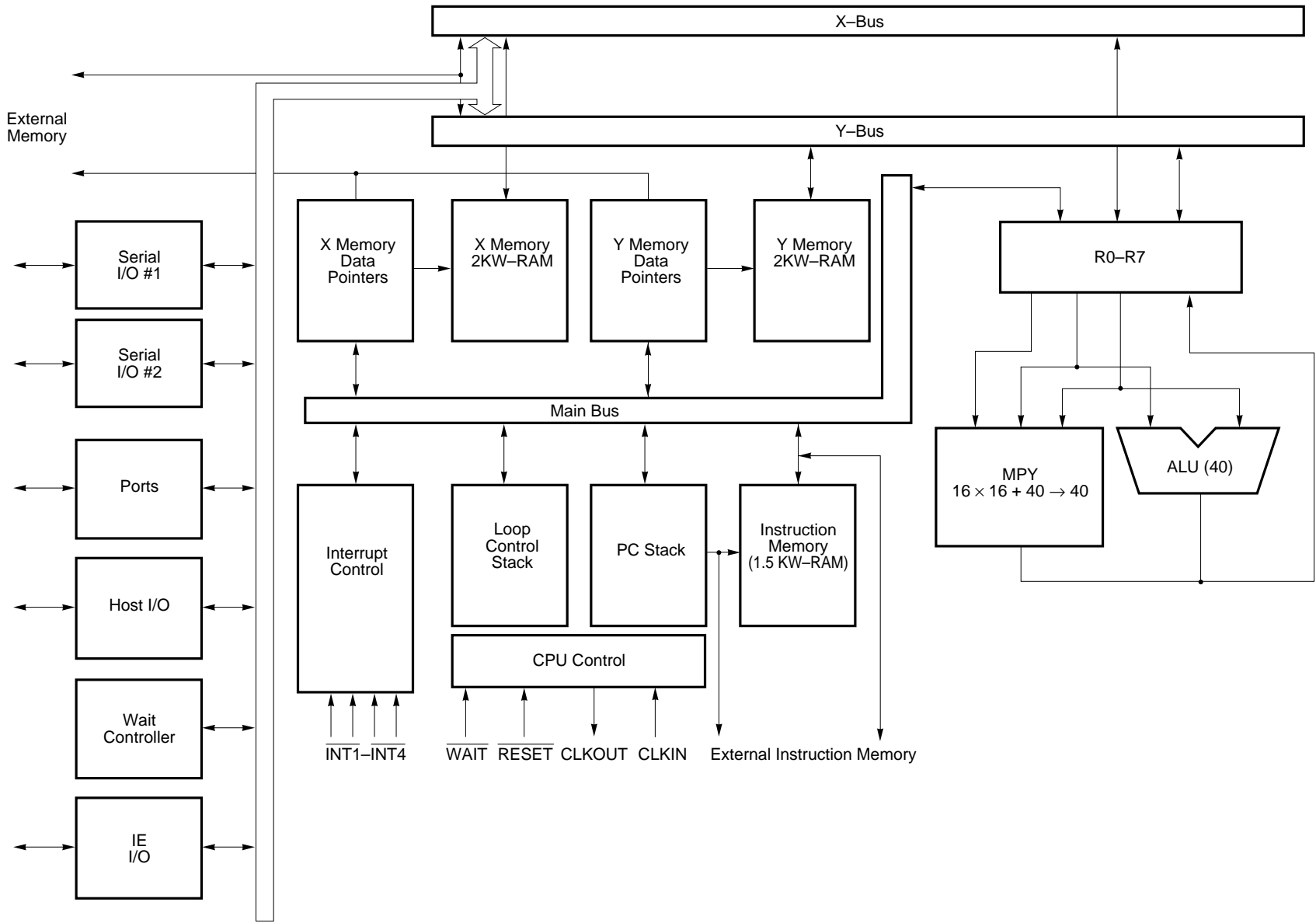
**FEATURES**

- FUNCTIONS
  - Instruction cycle: 30 ns (MIN.) with 33 MHz clock
  - 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
  - Program memory area: 64K words  $\times$  32 bits
  - Two independent data memory areas: 64K words  $\times$  16 bits (X/Y memory)
- ON-CHIP PERIPHERAL
  - I/O port: 4 bits
  - Serial I/O (16 bits): 2 channels
- CMOS
- +5 V single power supply

**ORDERING INFORMATION**

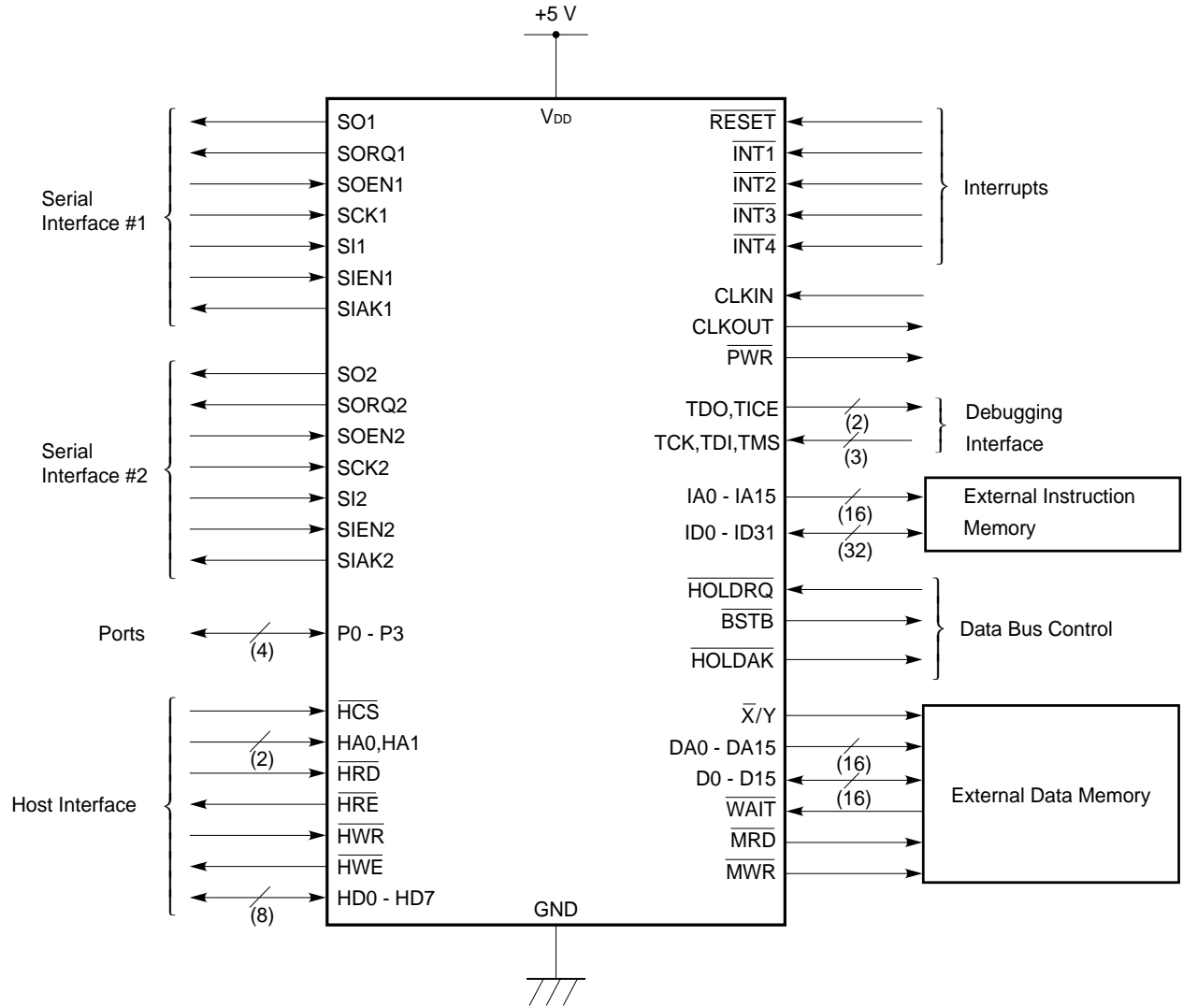
| Part Number         | Package  |
|---------------------|--|
| $\mu$ PD77016GM-KMD | 160-pin plastic QFP (FINE PITCH) (24 $\times$ 24 mm) |

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



BLOCK DIAGRAM

FUNCTIONAL PIN GROUPS



★ Functional Differences among the  $\mu$ PD7701 $\times$  Family

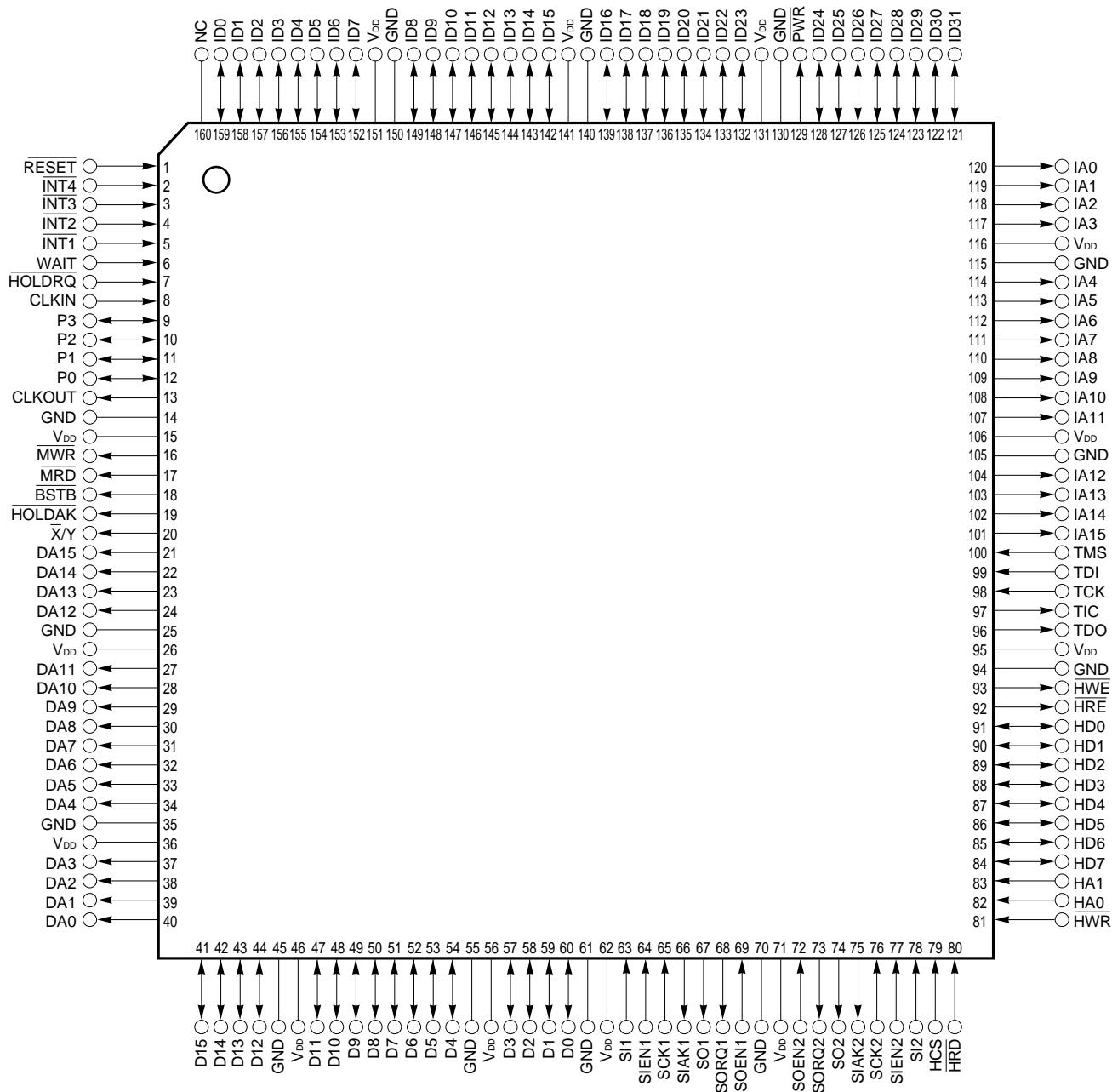
| Item   | $\mu$ PD77016  | $\mu$ PD77015   | $\mu$ PD77017 | $\mu$ PD77018  | $\mu$ PD77018A  | $\mu$ PD77019        | $\mu$ PD77019-013                         |
|--|--|---|---------------|----------------|---|----------------------|---|
| Internal instruction RAM                       | 1.5K words   | 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<br>(Maximum operation speed) | 30 ns (33 MHz)                                       |   |               |                | 16.6 ns (60 MHz)  |                      |   |
| External clock<br>(at maximum operation speed) | 66 MHz   | 33/16.5/8.25/4.125 MHz<br>Variable multiple rate (1, 2, 4, 8 ) by mask option.  |               |                | 60/30/20/15/7.5 MHz<br>Variable multiple rate (1, 2, 3, 4, 8 ) by<br>mask option. |                      | 15 MHz<br>Multiple rate is<br>fixed to 4. |
| Crystal<br>(at maximum operation speed)        | –  | 33 MHz  |               |                | 60 MHz  |                      | –   |
| Instruction                                    | –  | STOP instruction is added.  |               |                |   |                      |   |
| Serial interface (2 Channels)                  | Channel 1 has the<br>same functions<br>as channel 2. | Channel 1 has the same functions as that of the $\mu$ PD77016.<br>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<br>116-pin plastic BGA                                       | 100-pin plastic TQFP |   |

**Remark** The  $\mu$ PD77019-013 internal ROM area is masked already by the void code to use as RAM based DSP without mask code ordering process.

PIN CONFIGURATION

μPD77016GM-KMD

160-pin plastic QFP (FINE PITCH) (24 × 24 mm) (Top View)



## PIN IDENTIFICATION

|                                 |                                   |
|---------------------------------|-----------------------------------|
| $\overline{\text{BSTB}}$ :      | Bus Strobe                        |
| CLKIN:                          | Clock Input                       |
| CLKOUT:                         | Clock Output                      |
| D0-D15:                         | 16 Bits Data Bus                  |
| DA0-DA15:                       | 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                 |
| HWR:                            | Host Write                        |
| IA0-IA15:                       | Instruction Memory Address Output |
| ID0-ID31:                       | Instruction Data Input            |
| $\overline{\text{INT1-INT4}}$ : | Interrupt                         |
| $\overline{\text{MRD}}$ :       | Memory Read Output                |
| $\overline{\text{MWR}}$ :       | Memory Write Output               |
| N.C:                            | No Connection                     |
| P0-P3:                          | Port                              |
| $\overline{\text{PWR}}$ :       | Program Memory Write Strobe       |
| $\overline{\text{RESET}}$ :     | Reset                             |
| SCK1,SCK2:                      | Serial Clock Input                |
| SI1,SI2:                        | Serial Data Input                 |
| SIK1,SIK2:                      | Serial Input Acknowledge          |
| SIEN1,SIEN2:                    | Serial Input Enable               |
| SO1,SO2:                        | Serial Data Output                |
| SOEN1,SOEN2:                    | Serial Output Enable              |
| SORQ1,SORQ2:                    | Serial Output Request             |
| TCK:                            | Test Clock Input                  |
| TDI:                            | Test Data Input                   |
| TDO:                            | Test Data Output                  |
| TICE:                           | Test In-Circuit Emulator          |
| TMS:                            | Test Mode Select                  |
| $V_{DD}$ :                      | Power Supply                      |
| $\overline{\text{WAIT}}$ :      | Wait Input                        |
| $\overline{\text{X/Y}}$ :       | X/Y Memory Select                 |

| Pin No. | Symbol          | Pin No. | Symbol          | Pin No. | Symbol          | Pin No. | Symbol          |
|---------|-----------------|---------|-----------------|---------|-----------------|---------|-----------------|
| 1       | RESET           | 41      | D15             | 81      | HWR             | 121     | ID31            |
| 2       | INT4            | 42      | D14             | 82      | HA0             | 122     | ID30            |
| 3       | INT3            | 43      | D13             | 83      | HA1             | 123     | ID29            |
| 4       | INT2            | 44      | D12             | 84      | HD7             | 124     | ID28            |
| 5       | INT1            | 45      | GND             | 85      | HD6             | 125     | ID27            |
| 6       | WAIT            | 46      | V <sub>DD</sub> | 86      | HD5             | 126     | ID26            |
| 7       | HOLDRQ          | 47      | D11             | 87      | HD4             | 127     | ID25            |
| 8       | CLKIN           | 48      | D10             | 88      | HD3             | 128     | ID24            |
| 9       | P3              | 49      | D9              | 89      | HD2             | 129     | PWR             |
| 10      | P2              | 50      | D8              | 90      | HD1             | 130     | GND             |
| 11      | P1              | 51      | D7              | 91      | HD0             | 131     | V <sub>DD</sub> |
| 12      | P0              | 52      | D6              | 92      | HRE             | 132     | ID23            |
| 13      | CLKOUT          | 53      | D5              | 93      | HWE             | 133     | ID22            |
| 14      | GND             | 54      | D4              | 94      | GND             | 134     | ID21            |
| 15      | V <sub>DD</sub> | 55      | GND             | 95      | V <sub>DD</sub> | 135     | ID20            |
| 16      | MWR             | 56      | V <sub>DD</sub> | 96      | TDO             | 136     | ID19            |
| 17      | MRD             | 57      | D3              | 97      | TICE            | 137     | ID18            |
| 18      | BSTB            | 58      | D2              | 98      | TCK             | 138     | ID17            |
| 19      | HOLDAK          | 59      | D1              | 99      | TDI             | 139     | ID16            |
| 20      | X/Y             | 60      | D0              | 100     | TMS             | 140     | GND             |
| 21      | DA15            | 61      | GND             | 101     | IA15            | 141     | V <sub>DD</sub> |
| 22      | DA14            | 62      | V <sub>DD</sub> | 102     | IA14            | 142     | ID15            |
| 23      | DA13            | 63      | SI1             | 103     | IA13            | 143     | ID14            |
| 24      | DA12            | 64      | SIEN1           | 104     | IA12            | 144     | ID13            |
| 25      | GND             | 65      | SCK1            | 105     | GND             | 145     | ID12            |
| 26      | V <sub>DD</sub> | 66      | SIK1            | 106     | V <sub>DD</sub> | 146     | ID11            |
| 27      | DA11            | 67      | SO1             | 107     | IA11            | 147     | ID10            |
| 28      | DA10            | 68      | SORQ1           | 108     | IA10            | 148     | ID9             |
| 29      | DA9             | 69      | SOEN1           | 109     | IA9             | 149     | ID8             |
| 30      | DA8             | 70      | GND             | 110     | IA8             | 150     | GND             |
| 31      | DA7             | 71      | V <sub>DD</sub> | 111     | IA7             | 151     | V <sub>DD</sub> |
| 32      | DA6             | 72      | SOEN2           | 112     | IA6             | 152     | ID7             |
| 33      | DA5             | 73      | SORQ2           | 113     | IA5             | 153     | ID6             |
| 34      | DA4             | 74      | SO2             | 114     | IA4             | 154     | ID5             |
| 35      | GND             | 75      | SIK2            | 115     | GND             | 155     | ID4             |
| 36      | V <sub>DD</sub> | 76      | SCK2            | 116     | V <sub>DD</sub> | 156     | ID3             |
| 37      | DA3             | 77      | SIEN2           | 117     | IA3             | 157     | ID2             |
| 38      | DA2             | 78      | SI2             | 118     | IA2             | 158     | ID1             |
| 39      | DA1             | 79      | HCS             | 119     | IA1             | 159     | ID0             |
| 40      | DA0             | 80      | HRD             | 120     | IA0             | 160     | NC              |

CONTENTS

|  |           |
|--|-----------|
| <b>1. PIN FUNCTIONS</b> .....                    | <b>9</b>  |
| 1.1 Pin Functions .....                          | 9         |
| 1.2 Recommended Connection for Unused Pins ..... | 14        |
| <b>2. FUNCTIONS</b> .....                        | <b>15</b> |
| <b>2.1 Pipeline Processing</b> .....             | <b>15</b> |
| 2.1.1 Outline .....                              | 15        |
| 2.1.2 Instructions with Delay .....              | 15        |
| <b>2.2 Program Control Unit</b> .....            | <b>16</b> |
| <b>2.3 Operation Unit</b> .....                  | <b>16</b> |
| 2.3.1 General register (R0 to R7) .....          | 16        |
| 2.3.2 MAC: Multiply ACcumulator .....            | 17        |
| 2.3.3 ALU: Arithmetic Logic Unit .....           | 17        |
| 2.3.4 BSFT: Barrel ShiFTer .....                 | 17        |
| 2.3.5 SAC: Shifter And Count Circuit .....       | 17        |
| 2.3.6 CJC: Condition Judge Circuit .....         | 17        |
| <b>2.4 Memory</b> .....                          | <b>18</b> |
| 2.4.1 Instruction RAM Outline .....              | 19        |
| 2.4.2 Data Memory Outline .....                  | 19        |
| 2.4.3 Data Memory Addressing .....               | 19        |
| <b>2.5 On-chip Peripheral Circuit</b> .....      | <b>20</b> |
| 2.5.1 Serial Interface Outline .....             | 20        |
| 2.5.2 Host Interface Outline .....               | 20        |
| 2.5.3 General Input/output Ports Outline .....   | 20        |
| 2.5.4 Wait Cycle Register .....                  | 20        |
| <b>3. INSTRUCTIONS</b> .....                     | <b>21</b> |
| 3.1 Outline .....                                | 21        |
| 3.2 Instruction Set and Operation .....          | 22        |
| <b>4. ELECTRICAL SPECIFICATIONS</b> .....        | <b>29</b> |
| <b>5. PACKAGE DRAWING</b> .....                  | <b>50</b> |
| <b>6. RECOMMENDED SOLDERING CONDITIONS</b> ..... | <b>51</b> |



## 1. PIN FUNCTIONS

### 1.1 Pin Functions

#### • Power supply

| Symbol          | Pin No.   | I/O | Function         |
|-----------------|---|-----|------------------|
| V <sub>DD</sub> | 15, 26, 36, 46, 56, 62, 71, 95, 106, 116, 131, 141, 151 | –   | +5V power supply |
| GND             | 14, 25, 35, 45, 55, 61, 70, 94, 105, 115, 130, 140, 150 | –   | Ground           |

#### • System control

| Symbol | Pin No. | I/O | Function                           |
|--------|---------|-----|------------------------------------|
| CLKIN  | 8       | I   | External clock input               |
| CLKOUT | 13      | O   | Internal system clock output       |
| RESET  | 1       | I   | Internal system reset signal input |

#### • Interrupt

| Symbol  | Pin No.    | I/O | Function  |
|---|------------|-----|---|
| $\overline{\text{INT4}} - \overline{\text{INT1}}$ | 2, 3, 4, 5 | I   | Maskable external interrupt input<br>• Falling edge detection |

• External data memory interface

| Symbol              | Pin No.        | I/O         | Function  |
|---------------------|----------------|-------------|---|
| $\overline{X/Y}$    | 20             | O<br>(3S)   | Memory select signal output<br><ul style="list-style-type: none"> <li>• 0: X memory is used.</li> <li>• 1: Y memory is used.</li> </ul>   |
| DA15 - DA0          | <b>Note 1.</b> | O<br>(3S)   | Address bus to external data memory<br><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.<br/>These pins are set to low level; 0x0000, by reset.<br/>They continue outputting low level until the first external memory access.</li> </ul> |
| D15 - D0            | <b>Note 2.</b> | I/O<br>(3S) | 16 bits data bus to external data memory<br><ul style="list-style-type: none"> <li>• External data memory is accessed.</li> </ul>   |
| $\overline{MRD}$    | 17             | O<br>(3S)   | Read output<br><ul style="list-style-type: none"> <li>• Reads external memory</li> </ul>  |
| $\overline{MWR}$    | 16             | O<br>(3S)   | Write output<br><ul style="list-style-type: none"> <li>• Writes external memory</li> </ul>  |
| $\overline{WAIT}$   | 6              | I           | Wait signal input<br><ul style="list-style-type: none"> <li>• Wait cycle is input when external memory is read.<br/>1: No wait<br/>0: Wait</li> </ul>   |
| $\overline{HOLDRQ}$ | 7              | I           | Hold request signal input<br><ul style="list-style-type: none"> <li>• Input low level when external data memory bus is expected to use.</li> </ul>  |
| $\overline{BSTB}$   | 18             | O           | Bus strobe signal output<br><ul style="list-style-type: none"> <li>• Outputs low level while the μPD77016 is occupying external memory bus.</li> </ul>  |
| $\overline{HOLDAK}$ | 19             | O           | Hold acknowledge signal output<br><ul style="list-style-type: none"> <li>• Outputs low level when the μPD77016 permits external device to use external data memory bus.</li> </ul>  |

**Note 1.** DA15 to DA0 pins are located on Pin No. 21 - 24, 27 - 34, 37 - 40.

**Note 2.** D15 to D0 pins are located on Pin No. 41 - 44, 47 - 54, 57 - 60.

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

## • Serial interface

| Symbol | Pin No. | I/O    | Function                   |
|--------|---------|--------|----------------------------|
| SCK1   | 65      | I      | Clock input for serial 1   |
| SORQ1  | 68      | O      | Serial output 1 request    |
| SOEN1  | 69      | I      | Serial output 1 enable     |
| SO1    | 67      | O (3S) | Serial data output 1       |
| SIEN1  | 64      | I      | Serial input 1 enable      |
| SI1    | 63      | I      | Serial data input 1        |
| SCK2   | 76      | I      | Clock input for serial 2   |
| SORQ2  | 73      | O      | Serial output 2 request    |
| SOEN2  | 72      | I      | Serial output 2 enable     |
| SO2    | 74      | O (3S) | Serial data output 2       |
| SIEN2  | 77      | I      | Serial input 2 enable      |
| SI2    | 78      | I      | Serial data input 2        |
| SI AK1 | 66      | O      | Serial input 1 acknowledge |
| SI AK2 | 75      | O      | Serial input 2 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              | 83      | I        | Specifies register which HD7 to HD0 access<br>1: Accesses HST: Host interface status register when HA1 = 0<br>0: Accesses HDT <sub>(out)</sub> : Host transmit data register when $\overline{HRD} = 0$<br>0: Accesses HDT <sub>(in)</sub> : Host receive data register when $\overline{HWR} = 0$ |
| HA0              | 82      | I        | Specifies bits of registers which HD7 to HD0 access<br>• 1: Accesses bits 15-8 of HST, HDT (out), HDT (in)<br>• 0: Accesses bits 7-0 of HST, HDT (out), HDT (in)   |
| $\overline{HCS}$ | 79      | I        | Chip select input  |
| $\overline{HRD}$ | 80      | I        | Host read input  |
| $\overline{HWR}$ | 81      | I        | Host write input   |
| $\overline{HRE}$ | 92      | O        | Host read enable output  |
| $\overline{HWE}$ | 93      | O        | Host write enable output   |
| HD7 - HD0        | 84 - 91 | 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 | 9 - 12  | I/O | I/O port |

• External instructions memory interface

| Symbol                  | Pin No.        | I/O      | Function   |
|-------------------------|----------------|----------|--|
| IA15 - IA0              | <b>Note 1.</b> | O (3S)   | Address bus to external instruction memory<br><ul style="list-style-type: none"> <li>• Even the internal instruction memory is accessed, the address is output to the external instruction memory. In this case, the μPD77016 ignores data of external instruction memory output.</li> </ul> |
| ID31 - ID0              | <b>Note 2.</b> | I/O (3S) | 32 bits instruction input  |
| $\overline{\text{PWR}}$ | 129            | O (3S)   | Program memory write strobe<br><ul style="list-style-type: none"> <li>• Write strobe for external instruction memory. This pin loads program to external instruction memory (not internal memory) while μPD77016 is in boot operation.</li> </ul>  |

**Note 1.** IA15 to IA0 pins are located on these pins: 101 to 104, 107 to 114, 117 to 120

**2.** ID31 to ID0 pins are located on these pins: 121 to 128, 132 to 139, 142 to 149, 152 to 159

**Remark** The state of the pins added 3S becomes high impedance when  $\overline{\text{RESET}}$  is input.

• Debugging interface

| Symbol | Pin No. | I/O | Function      |
|--------|---------|-----|---------------|
| TDO    | 96      | O   | For debugging |
| TICE   | 97      | O   | For debugging |
| TCK    | 98      | I   | For debugging |
| TDI    | 99      | I   | For debugging |
| TMS    | 100     | I   | For debugging |

1.2 Recommended Connection for Unused Pins

| Pin   | I/O | Recommended connection                     |
|---|-----|--|
| $\overline{\text{INT1}} - \overline{\text{INT4}}$ | I   | connect to $V_{DD}$                        |
| $\overline{X}/Y$                                  | O   | open                                       |
| DA0 - DA15  | O   |  |
| D0 - D15 <sup>Note 1</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, SORQ2                                      | O   | open                                       |
| SO1, SO2  | O   |  |
| SIAK1, SIAK2                                      | 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>Note 2</sup>                       | I/O | connect to $V_{DD}$ or GND, via a resistor |
| P0 - P3   | I/O |  |
| ID0 - ID31  | I/O |  |
| IA0 - IA15  | O   | open                                       |
| $\overline{\text{PWR}}$                           | 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 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 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 μPD77016 pipeline processing.

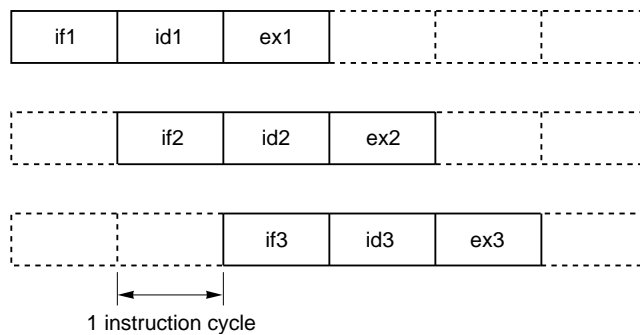
**2.1.1 Outline**

The μPD77016 basic operations are executed in following 3-stage pipeline.

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

When the μPD77016 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 μPD77016 has external 4 interruptions and internal 6 interruptions from peripheral, and specifies interrupt enable or disable independently.

The HALT instruction causes the μPD77016 to place in low power standby mode.

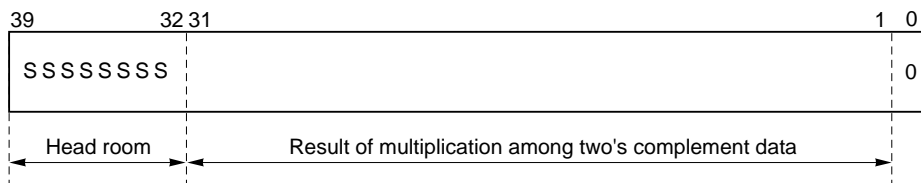
When the HALT instruction is executed, power consumption decreases. HALT mode is released by interrupt input or hardware reset input. It takes several system clock to recover.

**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 μPD77016 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 × signed data, signed data × unsigned data and unsigned data × 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**

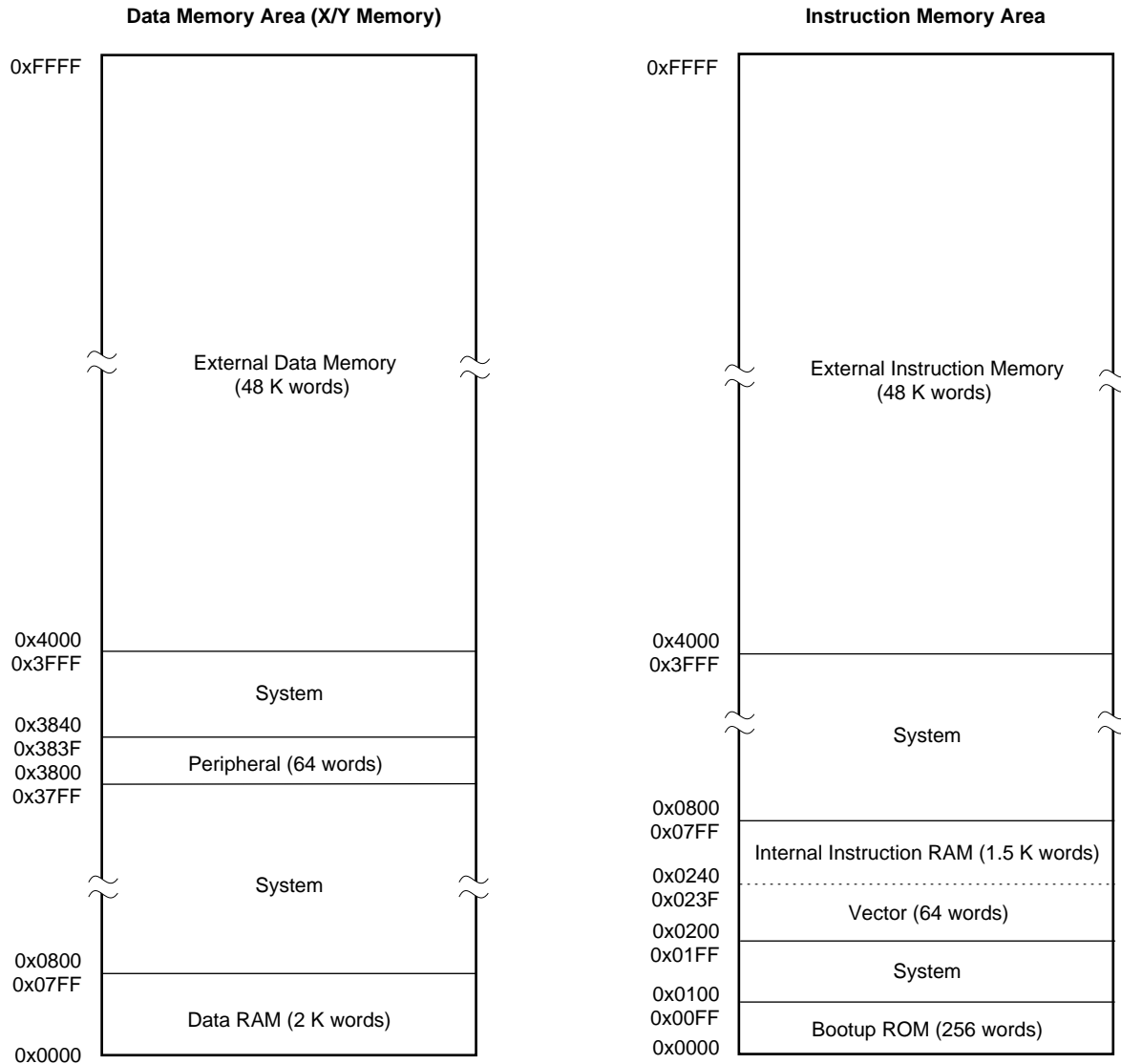
CJC judges whether condition is true or false with 40 bits input data. A conditional instruction is executed when the result is true, and not executed when the result is false.

2.4 Memory

The μPD77016 has one instruction memory area (64K words × 32 bits) and two data memory areas (64K words × 16 bits each). It adopts Harvard-type architecture, with instruction memory area and data memory areas separated.

The μPD77016 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. Memory areas are shown below.

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.



**Caution** When any data is accessed or stored to system address, normal operation of the μPD77016 is not assured.

### 2.4.1 Instruction RAM Outline

The  $\mu$ PD77016 has an instruction RAM (1.5 words  $\times$  32 bits). A system vector area is assigned to 64 words of the instruction RAM. Internal RAM is initialized and rewritten by boot program.

Additionally external memory expansion is available as the  $\mu$ PD77016 has interface with the external instruction memory. When RAM is used as the external memory, it can be initialized and rewritten by boot program.

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

When the external instruction memory area is accessed, instruction cycle can be 2 or more by wait function.

### 2.4.2 Data Memory Outline

The  $\mu$ PD77016 has two data memory areas (64 words  $\times$  16 bits each) in X and Y memory areas.

Each memory areas consists of 2K words  $\times$  16 bits data RAM. Additionally, data memory expansion is available as the  $\mu$ PD77016 has interface with the external data memory.

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.

### 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 μPD77016 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 μPD77016 CPU.

### 2.5.1 Serial Interface Outline

The μPD77016 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 μPD77016 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 μPD77016 includes 3 registers consisting of 16 bits, which are dedicated for input data, output data and status. The μPD77016 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 μPD77016 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. 0, 1, 3, or 7 wait cycle can be set in every data area which is divided into 8, and in every X and Y memory area which is divided into 4.

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

### 3. INSTRUCTIONS

#### 3.1 Outline

All  $\mu$ PD77016 instructions are one-word instructions, consisting of 32 bits. And they are executed in 30 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<br><br><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##       | $DPn \leftarrow DPn + DNn$ : Adds DN0-DN7 corresponding to DP0-DP7<br><b>Example</b> $DP0 \leftarrow DP0 + DN0$ |
| DPn%%       | $(n = 0 - 3) \quad DPn = ((DP_L + DNn) \bmod (DMX + 1)) + DP_H$   |
|             | $(n = 4 - 7) \quad DPn = ((DP_L + DNn) \bmod (DMY + 1)) + DP_H$   |
| !DPn##      | Access memory after DPn value is bit-reversed<br>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 μPD77016 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.

# μPD77016 INSTRUCTION SET

|           | Name                                | Mnemonic  | Operation                                    | Concurrent Writing Processing |       |        |                |        |      |     |       |      |    | Flag |  |   |
|-----------|-------------------------------------|---|--|-------------------------------|-------|--------|----------------|--------|------|-----|-------|------|----|------|--|---|
|           |                                     |   |  | TRI.                          | DYAD. | MONAD. | Load/<br>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<br>Multiply add         | $ro = ro + rh * rl$<br>(rl should be a plus<br>integral number.)          | $ro \leftarrow ro + rh * rl$                 |                               |       |        | ○              |        |      |     |       |      |    |      |  | ↕ |
|           | Unsign unsign<br>Multiply add       | $ro = ro + rl * rl'$<br>(rl and rl' should be a plus<br>integral number.) | $ro \leftarrow ro + rl * rl'$                |                               |       |        | ○              |        |      |     |       |      |    |      |  | ↕ |
|           | 1 bit shift Multiply add            | $ro = (ro \gg 1) + rh * rh'$  | $ro \leftarrow \frac{ro}{2} + rh * rh'$      |                               |       |        | ○              |        |      |     |       |      |    |      |  | ↕ |
|           | 16 bits shift Multiply add          | $ro = (ro \gg 16) + rh * rh'$   | $ro \leftarrow \frac{ro}{2^{16}} + rh * rh'$ |                               |       |        | ○              |        |      |     |       |      |    |      |  | ● |
| 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   | $ro' \leftarrow ro \gg rl$                   |                               |       |        | ○              |        |      |     |       |      |    |      |  | ● |
|           | Immediate arithmetic<br>right shift | $ro' = ro$ SRA imm  | $ro' \leftarrow ro \gg imm$                  |                               |       |        |                |        |      |     |       |      |    |      |  | ● |
|           | Logic right shift                   | $ro' = ro$ SRL rl   | $ro' \leftarrow ro \gg rl$                   |                               |       |        | ○              |        |      |     |       |      |    |      |  | ● |
|           | Immediate Logic right shift         | $ro' = ro$ SRL imm  | $ro' \leftarrow ro \gg imm$                  |                               |       |        |                |        |      |     |       |      |    |      |  | ● |
|           | Logic left shift                    | $ro' = ro$ SLL rl   | $ro' \leftarrow ro \ll rl$                   |                               |       |        | ○              |        |      |     |       |      |    |      |  | ● |
|           | Immediate logic left shift          | $ro' = ro$ SLL imm  | $ro' \leftarrow ro \ll imm$                  |                               |       |        |                |        |      |     |       |      |    |      |  | ● |



|              | Name                   | Mnemonic               | Operation   | Concurrent Writing Processing |       |        |                |        |      |     |       |      | Flag |   |
|--------------|------------------------|------------------------|---|-------------------------------|-------|--------|----------------|--------|------|-----|-------|------|------|---|
|              |                        |                        |   | TRI.                          | DYAD. | MONAD. | Load/<br>store | TRANS. | IMM. | BR. | LOOP. | CTL. | OV   |   |
| Dyadic       | And                    | $ro'' = ro \& ro'$     | $ro'' \leftarrow ro \& ro'$   |                               |       |        | ○              |        |      |     |       |      |      | ● |
|              | Immediate and          | $ro' = ro \& imm$      | $ro' \leftarrow ro \& imm$  |                               |       |        |                |        |      |     |       |      |      | ● |
|              | Or                     | $ro'' = ro   ro'$      | $ro'' \leftarrow ro   ro'$  |                               |       |        | ○              |        |      |     |       |      |      | ● |
|              | Immediate or           | $ro' = ro   imm$       | $ro' \leftarrow ro   imm$   |                               |       |        |                |        |      |     |       |      |      | ● |
|              | Exclusive or           | $ro'' = ro \wedge ro'$ | $ro'' \leftarrow ro \wedge ro'$   |                               |       |        | ○              |        |      |     |       |      |      | ● |
|              | Immediate exclusive or | $ro = ro \wedge imm$   | $ro \leftarrow ro \wedge imm$   |                               |       |        |                |        |      |     |       |      |      | ● |
|              | Less than              | $ro'' = LT(ro, ro')$   | if(ro<ro')<br>{ro'' ← 0x0000000001}<br>else {ro'' ← 0x0000000000}   |                               |       |        | ○              |        |      |     |       |      |      |   |
| Monadic      | Clear                  | CLR(ro)                | $ro \leftarrow 0x0000000000$  |                               |       |        | ○              |        |      |     |       |      | ○    | ● |
|              | Increment              | $ro' = ro + 1$         | $ro' \leftarrow ro + 1$   |                               |       |        | ○              |        |      |     |       |      | ○    | ↓ |
|              | Decrement              | $ro' = ro - 1$         | $ro' \leftarrow ro - 1$   |                               |       |        | ○              |        |      |     |       |      | ○    | ↓ |
|              | Absolute               | $ro' = ABS(ro)$        | if (ro<0)<br>{ro' ← -ro}<br>else {ro' ← ro}   |                               |       |        | ○              |        |      |     |       |      | ○    | ↓ |
|              | One's complement       | $ro' = \sim ro$        | $ro' \leftarrow \sim ro$  |                               |       |        | ○              |        |      |     |       |      | ○    | ● |
|              | Two's complement       | $ro' = -ro$            | $ro' \leftarrow -ro$  |                               |       |        | ○              |        |      |     |       |      | ○    | ↓ |
|              | Clip                   | $ro' = CLIP(ro)$       | if (ro>0x007FFFFFFF)<br>{ro' ← 0x007FFFFFFF}<br>else if, (ro<0xFF80000000)<br>{ro' ← 0xFF80000000}<br>else {ro' ← ro}                             |                               |       |        | ○              |        |      |     |       |      | ○    | ↓ |
|              | Round                  | $ro' = ROUND(ro)$      | if (ro>0x007FFF0000)<br>{ro' ← 0x007FFF0000}<br>else if, (ro>0xFF80000000)<br>{ro' ← 0xFF80000000}<br>else {ro' ←<br>(ro + 0x8000) & 0xFFFFF0000} |                               |       |        | ○              |        |      |     |       |      | ○    | ↓ |
|              | Exponent               | $ro' = EXP(ro)$        | $ro' \leftarrow \log_2 \left( \frac{1}{ro} \right)$   |                               |       |        | ○              |        |      |     |       |      | ○    | ● |
| Substitution | $ro' = ro$             | $ro' \leftarrow ro$    |   |                               |       | ○      |                |        |      |     |       | ○    | ●    |   |

|            | Name  | Mnemonic                         | Operation   | Concurrent Writing Processing |       |        |                |        |      |     |       |      |    | Flag |   |
|------------|---|----------------------------------|---|-------------------------------|-------|--------|----------------|--------|------|-----|-------|------|----|------|---|
|            |   |                                  |   | TRI.                          | DYAD. | MONAD. | Load/<br>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))<br>{ro' ← (ro'-ro)<<1}<br>else<br>{ro' ← (ro'+ro)<<1}<br>if (sign(ro')==0<br>{ro' ← ro'+1} |                               |       |        | ○              |        |      |     |       |      | ○  | ↕    |   |
| Load/store | Parallel load/store<br>Note1, Note2.        | 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<br>Note1, Note2, 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'   |                               |       |        |                |        |      |     |       |      |    |      | ● |

- Note 1.** One or both of a mnemonic pair can be written.
- 2.** After execution of load/store, data is modified by mod.
- 3.** 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/<br>store | TRANS. | IMM. | BR. | LOOP. | CTL. | OV |      |   |   |   |
| Load/store                 | Direct addressing<br>load/store <b>Note 1.</b> | dest = *addr                          | dest ← *addr   |                               |       |        |                |        |      |     |       |      |    |      |   | ● |   |
|                            |  | *addr = source                        | *addr ← source |                               |       |        |                |        |      |     |       |      |    |      |   |   |   |
|                            | Immediate index<br>load/store <b>Note 2.</b>   | dest = *dp_imm                        | dest ← *dp     |                               |       |        |                |        |      |     |       |      |    |      |   |   | ● |
|                            |  | *dp_imm = source                      | *dp ← source   |                               |       |        |                |        |      |     |       |      |    |      |   |   |   |
| Inter-register<br>transfer | Inter-register transfer<br><b>Note 3.</b>      | dest = rl                             | dest ← rl      |                               |       |        |                |        |      |     |       |      |    |      | ○ | ● |   |
|                            |  | rl = source                           | rl ← source    |                               |       |        |                |        |      |     |       |      |    |      |   |   |   |
| Immediate<br>data set      | Immediate data set                             | rl = imm<br>(provided imm = 0-0xFFFF) | rl ← imm       |                               |       |        |                |        |      |     |       |      |    |      |   |   |   |
|                            |  | dp = imm<br>(provided imm = 0-0xFFFF) | dp ← imm       |                               |       |        |                |        |      |     |       |      |    |      |   |   |   |
|                            |  | dn = imm<br>(provided imm = 0-0xFFFF) | dn ← imm       |                               |       |        |                |        |      |     |       |      |    |      |   |   |   |
|                            |  | dm = imm<br>(provided imm = 1-0xFFFF) | dm ← imm       |                               |       |        |                |        |      |     |       |      |    |      |   |   | ● |

- Note 1.** One of following mnemonic should be selected: dest = {ro, reh, re, rh, rl}, source = {re, rh, rl}, add =  $\left\{ \begin{array}{l} 0: X-0xFFFF:X \text{ memory} \\ 0: Y-0xFFFF:Y \text{ memory} \end{array} \right\}$ .
- 2.** One of following mnemonic should be selected: dest = {ro, reh, re, rh, rl}, source = {re, rh, rl}.
- 3.** Any register except general registers should be selected as dest or source.

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

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 +7.0                  | V    |
| Input voltage                 | V <sub>I</sub>   |            | -0.5 to V <sub>DD</sub> + 0.5 | V    |
| Output voltage                | V <sub>O</sub>   |            | -0.5 to V <sub>DD</sub> + 0.5 | 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.

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

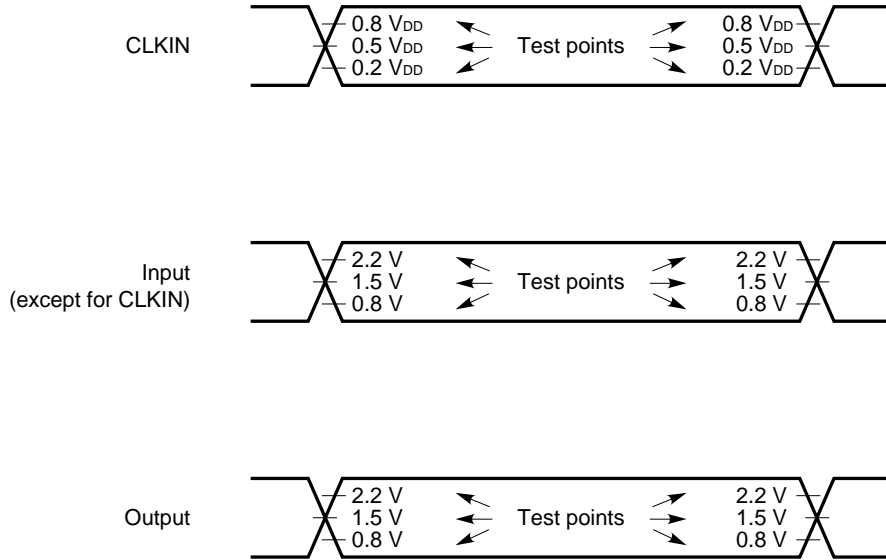
| Parameters         | Symbol         | Conditions   | MIN. | TYP. | MAX. | Unit |
|--------------------|----------------|--|------|------|------|------|
| Input capacitance  | C <sub>I</sub> | f <sub>c</sub> = 1 MHz<br>Unmeasured pins returned to 0 V. |      |      | 15   | pF   |
| Output capacitance | C <sub>O</sub> |  |      |      | 15   | pF   |

DC characteristics (T<sub>A</sub> = -40 to +85 °C, V<sub>DD</sub> = 5 V ±10 %)

| Parameters                    | Symbol                          | Conditions  | MIN.               | TYP. | MAX.                  | Unit |
|-------------------------------|---------------------------------|---|--------------------|------|-----------------------|------|
| High level input voltage      | V <sub>IH</sub>                 | except for $\overline{\text{RESET}}$ , $\overline{\text{CLKIN}}$ ,<br>$\overline{\text{INT1}} - \overline{\text{INT4}}$ , $\overline{\text{WAIT}}$ , $\overline{\text{HCS}}$ ,<br>$\overline{\text{HRD}}$ , $\overline{\text{HWR}}$ , TCK, TDI, TMS | 2.2                |      | V <sub>DD</sub> + 0.5 | V    |
|                               | V <sub>IHC</sub>                | $\overline{\text{RESET}}$ , $\overline{\text{INT1}} - \overline{\text{INT4}}$ , $\overline{\text{WAIT}}$ , $\overline{\text{HCS}}$ ,<br>$\overline{\text{HRD}}$ , $\overline{\text{HWR}}$ , TCK, TDI, TMS   | 0.7V <sub>DD</sub> |      | V <sub>DD</sub> + 0.5 | V    |
| Low level input voltage       | V <sub>IL</sub>                 | except for $\overline{\text{RESET}}$ , $\overline{\text{CLKIN}}$ ,<br>$\overline{\text{INT1}} - \overline{\text{INT4}}$ , $\overline{\text{WAIT}}$ , $\overline{\text{HCS}}$ ,<br>$\overline{\text{HRD}}$ , $\overline{\text{HWR}}$ , TCK, TDI, TMS | -0.5               |      | +0.8                  | V    |
|                               | V <sub>ILC</sub>                | $\overline{\text{RESET}}$ , $\overline{\text{INT1}} - \overline{\text{INT4}}$ , $\overline{\text{WAIT}}$ , $\overline{\text{HCS}}$ ,<br>$\overline{\text{HRD}}$ , $\overline{\text{HWR}}$ , TCK, TDI, TMS   | -0.5               |      | 0.2V <sub>DD</sub>    | V    |
| High level CLKIN voltage      | V <sub>IHX</sub>                |   | 0.8V <sub>DD</sub> |      | V <sub>DD</sub> + 0.5 | V    |
| Low level CLKIN voltage       | V <sub>ILX</sub>                |   | -0.5               |      | 0.2V <sub>DD</sub>    | V    |
| High level output voltage     | V <sub>OH</sub>                 | I <sub>OH</sub> = -2.5 mA   | 0.8V <sub>DD</sub> |      |                       | V    |
| Low level output voltage      | V <sub>OL</sub>                 | I <sub>OL</sub> = 2.5 mA  |                    |      | 0.4                   | V    |
| Low level input current       | I <sub>IL</sub>                 | TDI, TMS, V <sub>I</sub> = 0 V  |                    |      | -400                  | μA   |
| High level input leak current | I <sub>LIH</sub>                | 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   |
| Power supply current          | I <sub>DD</sub> <sup>Note</sup> | Active mode, t <sub>cCl</sub> = 15 ns<br>V <sub>IH</sub> = V <sub>DD</sub> , V <sub>IL</sub> = 0 V, no load   |                    | 140  | 300                   | mA   |
|                               | I <sub>DDH</sub>                | HALT mode, t <sub>cCl</sub> = 15 ns,<br>V <sub>IH</sub> = V <sub>DD</sub> , V <sub>IL</sub> = 0 V, no load  |                    | 80   |                       | mA   |
|                               | I <sub>DDS</sub>                | CLKIN = 0 V<br>V <sub>IH</sub> = V <sub>DD</sub> , V <sub>IL</sub> = 0 V, no load   |                    | 10   |                       | μA   |

**Note** The TYP. value is measured when a general program is executed, and V<sub>DD</sub> = 5 V condition. The MAX. value is measured when a special program that max. switching required is executed, and V<sub>DD</sub> = 5.5 V condition.

Measurement Standards Common to Switching Characteristics



AC Characteristics ( $T_A = -40$  to  $+85$  °C,  $V_{DD} = 5 V \pm 10\%$ ,  $C_L = 30 pF$ )

Clock

Required Timing Condition

| Parameters             | Symbol     | Conditions | MIN. | TYP. | MAX.           | Unit |
|------------------------|------------|------------|------|------|----------------|------|
| CLKIN cycle time       | $t_{cCl}$  |            | 15   |      | 500            | ns   |
| CLKIN high level width | $t_{wClH}$ |            | 6.75 |      | $0.55 t_{cCl}$ | ns   |
| CLKIN low level width  | $t_{wClL}$ |            | 6.75 |      | $0.55 t_{cCl}$ | ns   |
| CLKIN rise/fall time   | $t_{rCl}$  |            |      |      | 6              | ns   |

Switching Characteristics

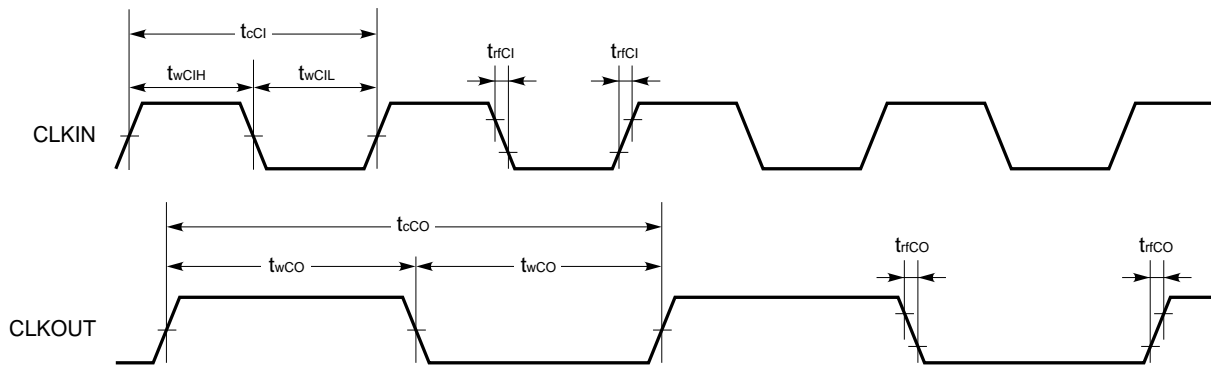
| Parameters            | Symbol    | Conditions | MIN.          | TYP.       | MAX. | Unit |
|-----------------------|-----------|------------|---------------|------------|------|------|
| CLKOUT cycle time     | $t_{cCO}$ |            |               | $2t_{cCl}$ |      | ns   |
| CLKOUT level width    | $t_{wCO}$ |            | $t_{cCl} - 3$ |            |      | ns   |
| CLKOUT rise/fall time | $t_{rCO}$ |            |               |            | 3    | ns   |

Reset, Interrupt

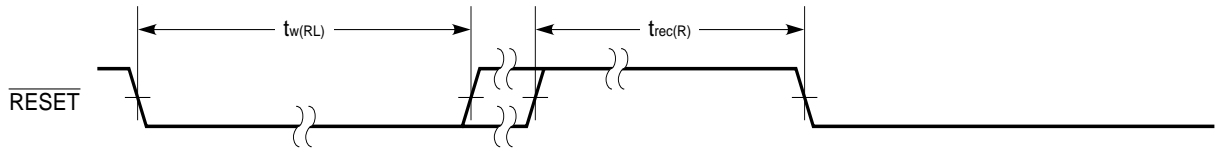
Required Timing Condition

| Parameters                                   | Symbol         | Conditions | MIN.        | TYP. | MAX. | Unit |
|--|----------------|------------|-------------|------|------|------|
| RESE $\bar{T}$ low level width               | $t_{w(RL)}$    |            | 4 $t_{cCO}$ |      |      | ns   |
| RESE $\bar{T}$ recovery time                 | $t_{rec(R)}$   |            | 4 $t_{cCO}$ |      |      | ns   |
| INT $\bar{1}$ -INT $\bar{4}$ low level width | $t_{w(INTL)}$  |            | 3 $t_{cCO}$ |      |      | ns   |
| INT $\bar{1}$ -INT $\bar{4}$ recovery time   | $t_{rec(INT)}$ |            | 3 $t_{cCO}$ |      |      | ns   |

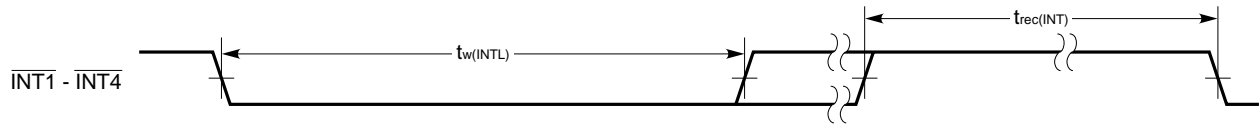
Clock Input/Output Timing



Reset, Interrupt Timing



Interrupt Timing



### External Data Memory Access

#### Required Timing Condition

| Parameters                   | Symbol        | Conditions | MIN. | TYP. | MAX. | Unit |
|------------------------------|---------------|------------|------|------|------|------|
| Read data setup time         | $t_{suDDR D}$ |            | 14   |      |      | ns   |
| Read data hold time          | $t_{hDDR D}$  |            | 0    |      |      | ns   |
| $\overline{WAIT}$ setup time | $t_{suWA}$    |            | 8    |      |      | ns   |
| $\overline{WAIT}$ hold time  | $t_{hWA}$     |            | 0    |      |      | ns   |

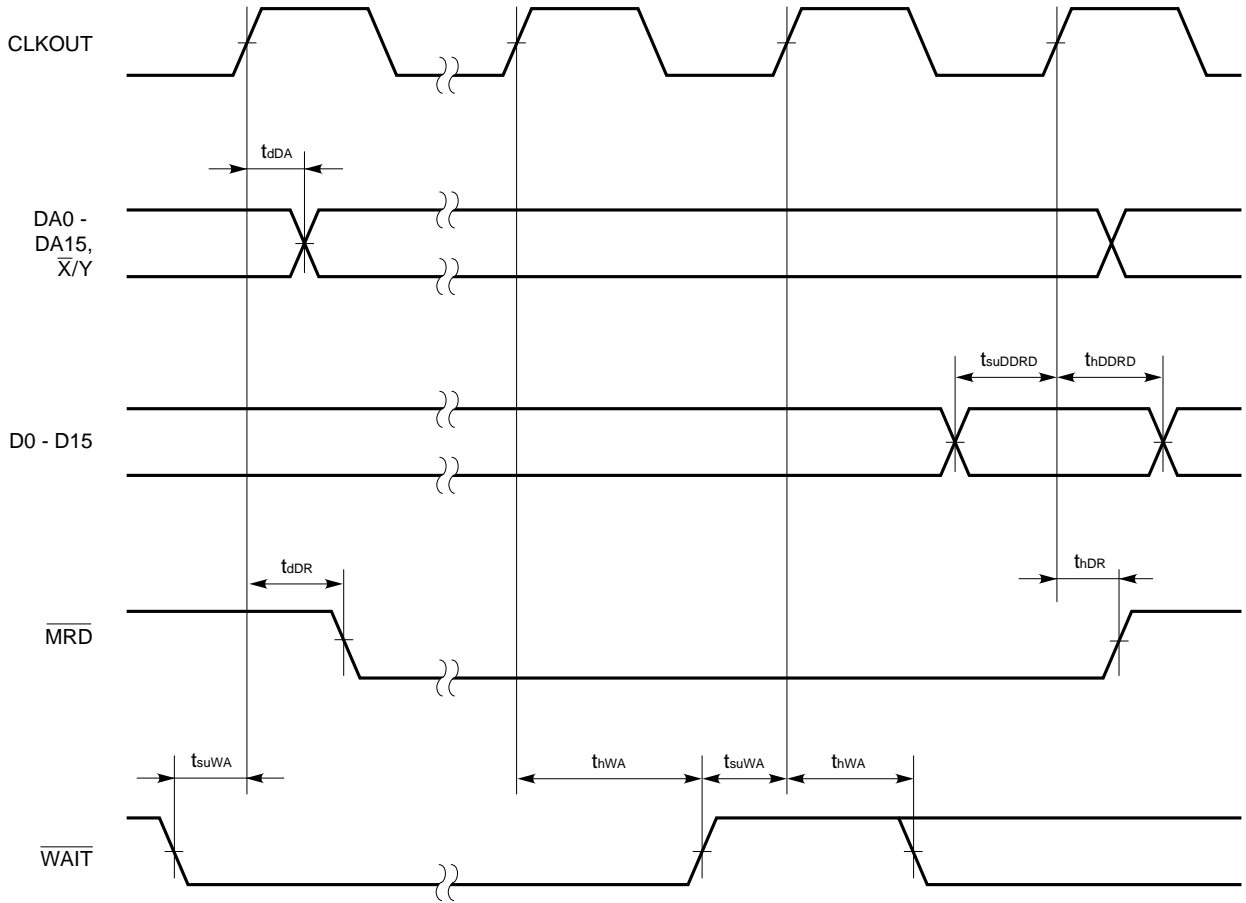
#### Switching Characteristics

| Parameters                         | Symbol      | Conditions | MIN.   | TYP. | MAX. | Unit |
|------------------------------------|-------------|------------|--|------|------|------|
| Address output delay time          | $t_{dDA}$   |            | 0  |      | 6    | ns   |
| $\overline{MRD}$ output delay time | $t_{dDR}$   |            | 0  |      | 8    | ns   |
| $\overline{MRD}$ hold time         | $t_{hDR}$   |            | 0  |      | 8    | ns   |
| Write data setup time              | $t_{sDDWD}$ |            | $t_{cCI} + t_{wCIH} - 15 + t_{cDW}$<br><b>Note</b> |      |      | ns   |
| Write data output hold time        | $t_{hDDWD}$ |            | 0  |      | 15   | ns   |
| $\overline{MWR}$ output delay time | $t_{dDW}$   |            | $t_{wCIH} - 4$                                     |      |      | ns   |
| $\overline{MWR}$ setup time        | $t_{suDW}$  |            | $t_{wCIL} - 4$                                     |      |      | ns   |
| $\overline{MWR}$ low level width   | $t_{wDWL}$  |            | $t_{cCI} - 4 + t_{cDW}$<br><b>Note</b>             |      |      | ns   |
| $\overline{MWR}$ high level width  | $t_{wDWH}$  |            | $t_{cCI} - 4$                                      |      |      | ns   |

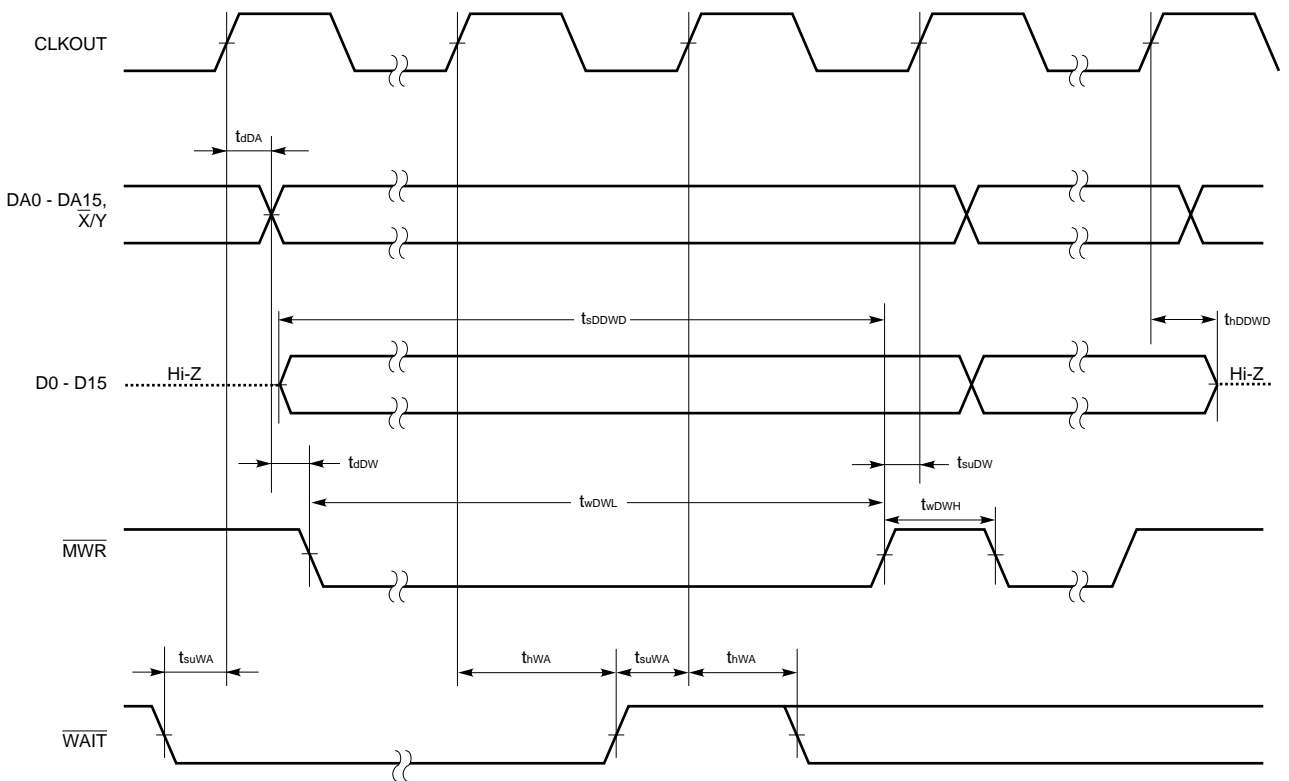
**Note**  $t_{cDW}$ : Data wait cycle



External Data Memory Read Operation



External Data Memory Write Operation



## External Instruction Memory Access

## Required Timing Condition

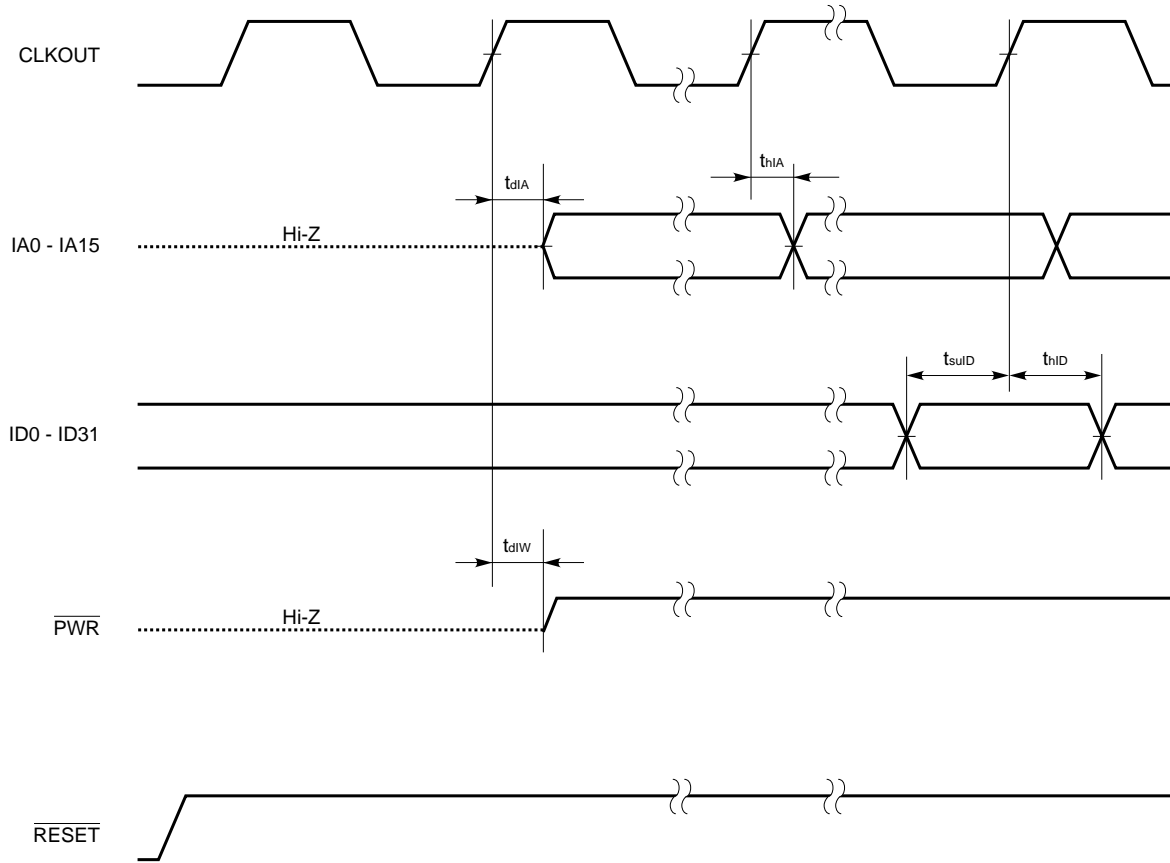
| Parameters                  | Symbol    | Conditions | MIN. | TYP. | MAX. | Unit |
|-----------------------------|-----------|------------|------|------|------|------|
| ID setup time (to CLKOUT ↑) | $t_{sID}$ |            | 14   |      |      | ns   |
| ID hold time (to CLKOUT ↑)  | $t_{hID}$ |            | 0    |      |      | ns   |

## Switching Characteristics

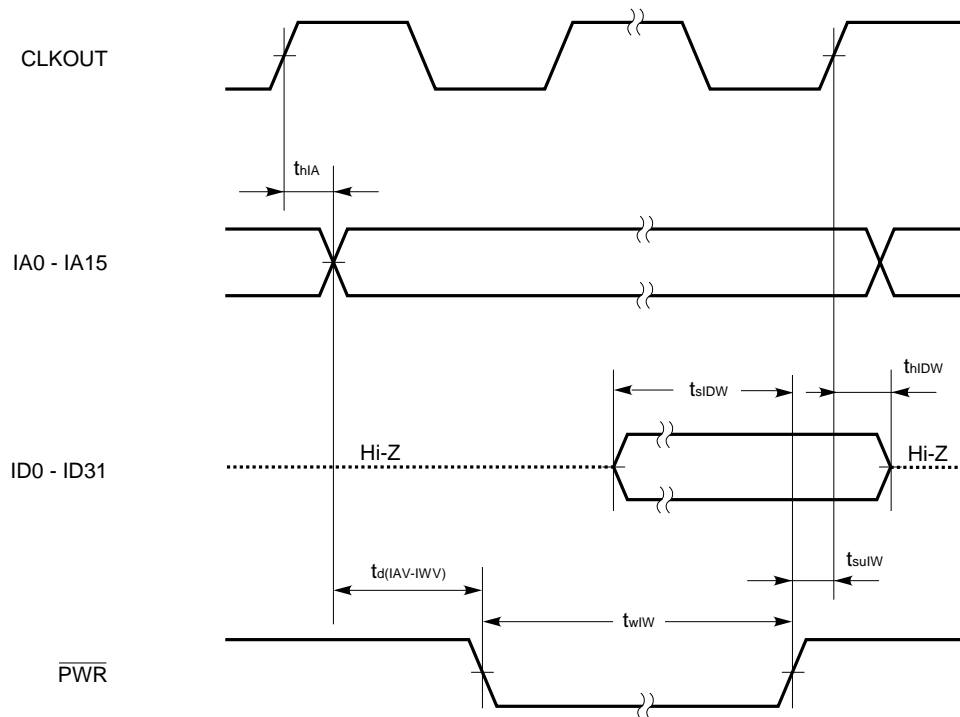
| Parameters               | Symbol           | Conditions | MIN.                         | TYP. | MAX. | Unit |
|--------------------------|------------------|------------|------------------------------|------|------|------|
| IA output delay time     | $t_{dIA}$        |            |                              | 10   |      | ns   |
| IA hold time             | $t_{hIA}$        |            | 0                            |      | 6    | ns   |
| ID write setup time      | $t_{sIDW}$       |            | $t_{cCI} + t_{wCIH}$<br>- 15 |      |      | ns   |
| ID write hold time       | $t_{hIDW}$       |            | 0                            |      |      | ns   |
| PWR output delay time    | $t_{dIW}$        |            |                              | 10   |      | ns   |
| Address → PWR setup time | $t_{d(IAV-IWV)}$ |            | $t_{cCI} + t_{wCIH}$<br>- 4  |      |      | ns   |
| PWR setup time           | $t_{sIW}$        |            | $t_{wCIL} - 4$               |      |      | ns   |
| PWR width                | $t_{wIW}$        |            | $t_{cCO} - 4$<br>+ $t_{cIW}$ |      |      | ns   |

**Remark**  $t_{cIW}$ : Instruction wait cycle

**External Instruction Memory Read Operation**



**External Instruction Memory Write Operation**



**Bus Arbitration**

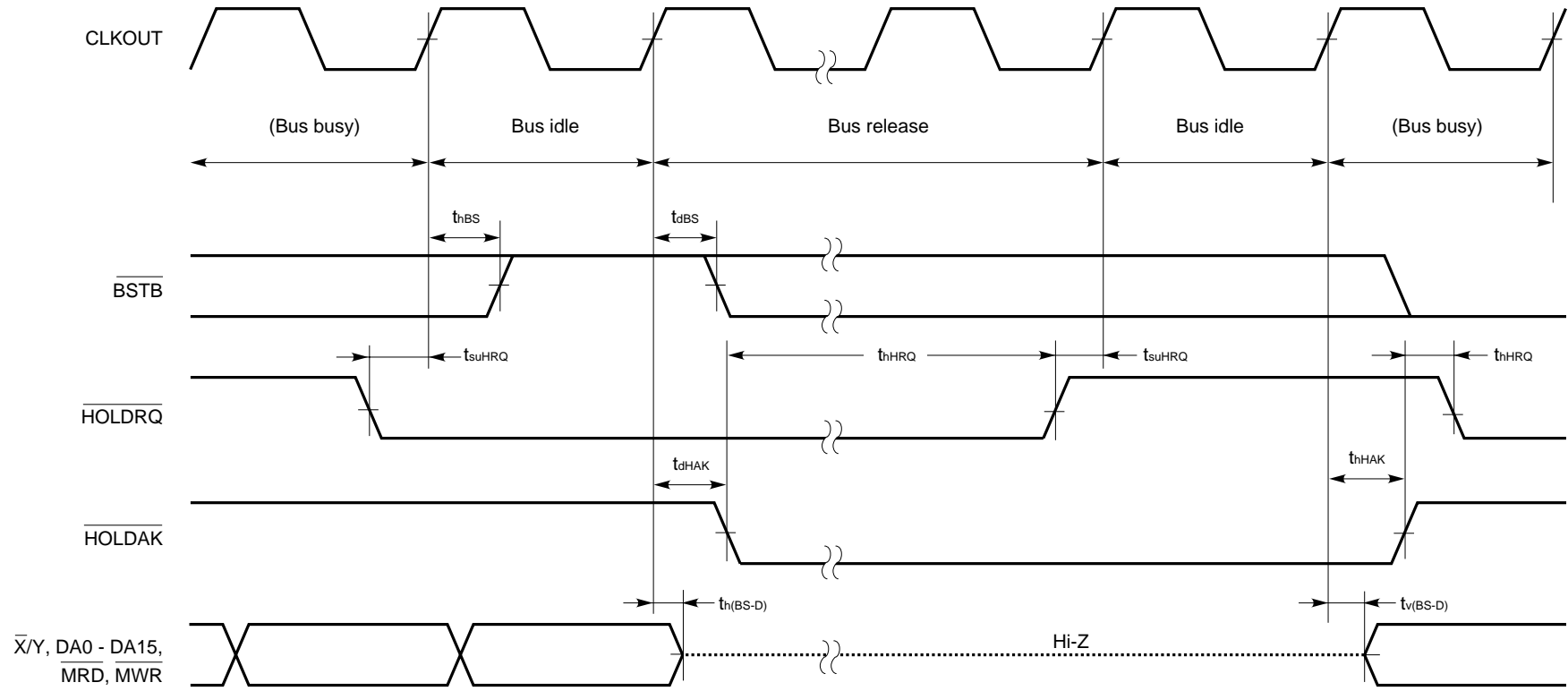
**Required Timing Condition**

| Parameters                            | Symbol             | Conditions | MIN. | TYP. | MAX. | Unit |
|---------------------------------------|--------------------|------------|------|------|------|------|
| $\overline{\text{HOLDRQ}}$ setup time | $t_{\text{suHRQ}}$ |            | 8    |      |      | ns   |
| $\overline{\text{HOLDRQ}}$ hold time  | $t_{\text{hHRQ}}$  |            | 0    |      |      | ns   |

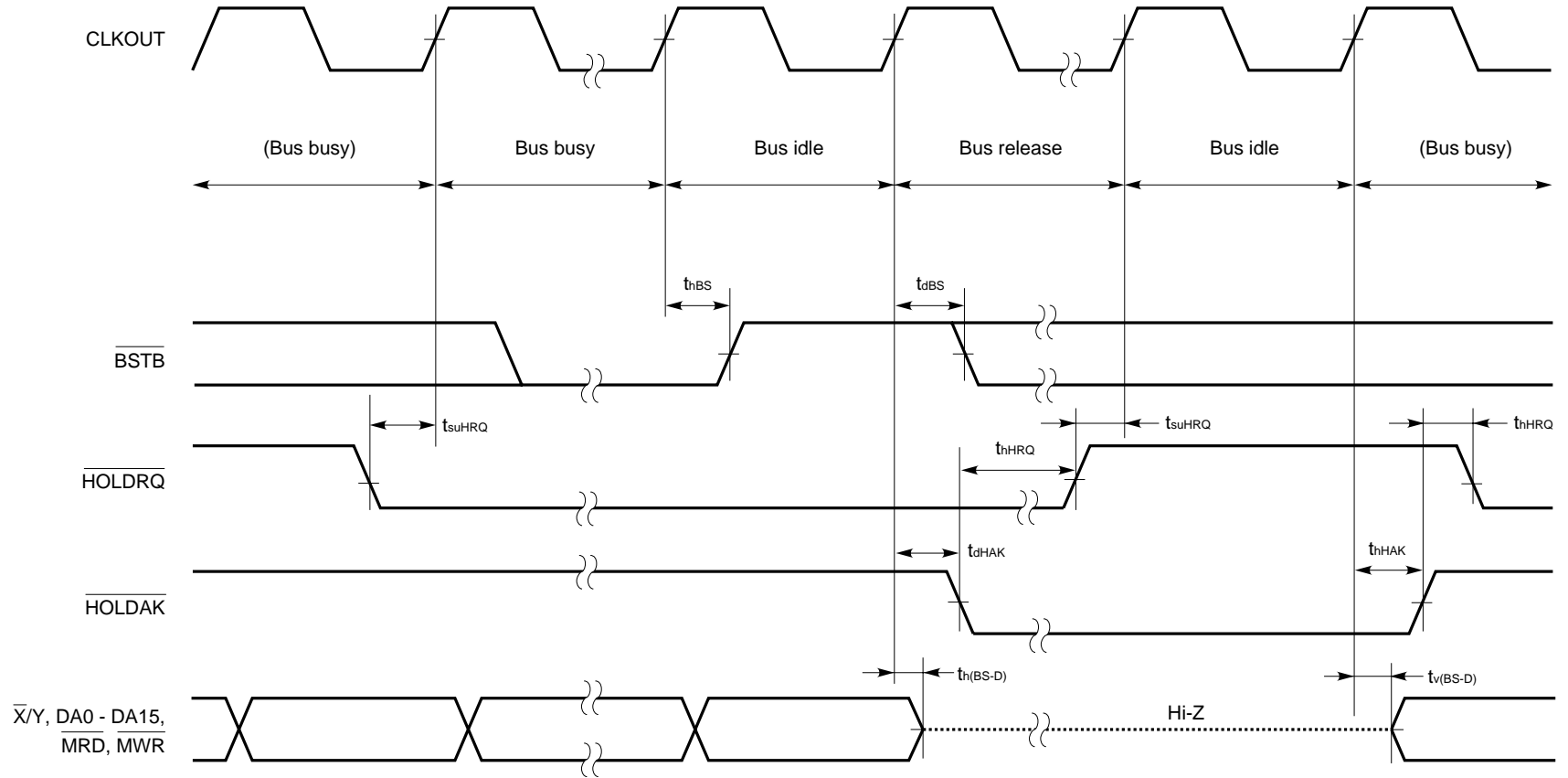
**Switching Characteristics**

| Parameters                                   | Symbol               | Conditions | MIN. | TYP. | MAX. | Unit |
|--|----------------------|------------|------|------|------|------|
| $\overline{\text{BSTB}}$ hold time           | $t_{\text{hBS}}$     |            | 0    |      | 6    | ns   |
| $\overline{\text{BSTB}}$ output delay time   | $t_{\text{dBS}}$     |            | 0    |      | 6    | ns   |
| $\overline{\text{HOLDAK}}$ output delay time | $t_{\text{dHAK}}$    |            | 0    |      | 6    | ns   |
| $\overline{\text{HOLDAK}}$ hold time         | $t_{\text{hHAK}}$    |            | 0    |      | 6    | ns   |
| Data hold time when bus arbitration          | $t_{\text{h(BS-D)}}$ |            |      |      | 15   | ns   |
| Data valid time after bus arbitration        | $t_{\text{v(BS-D)}}$ |            |      |      | 15   | ns   |

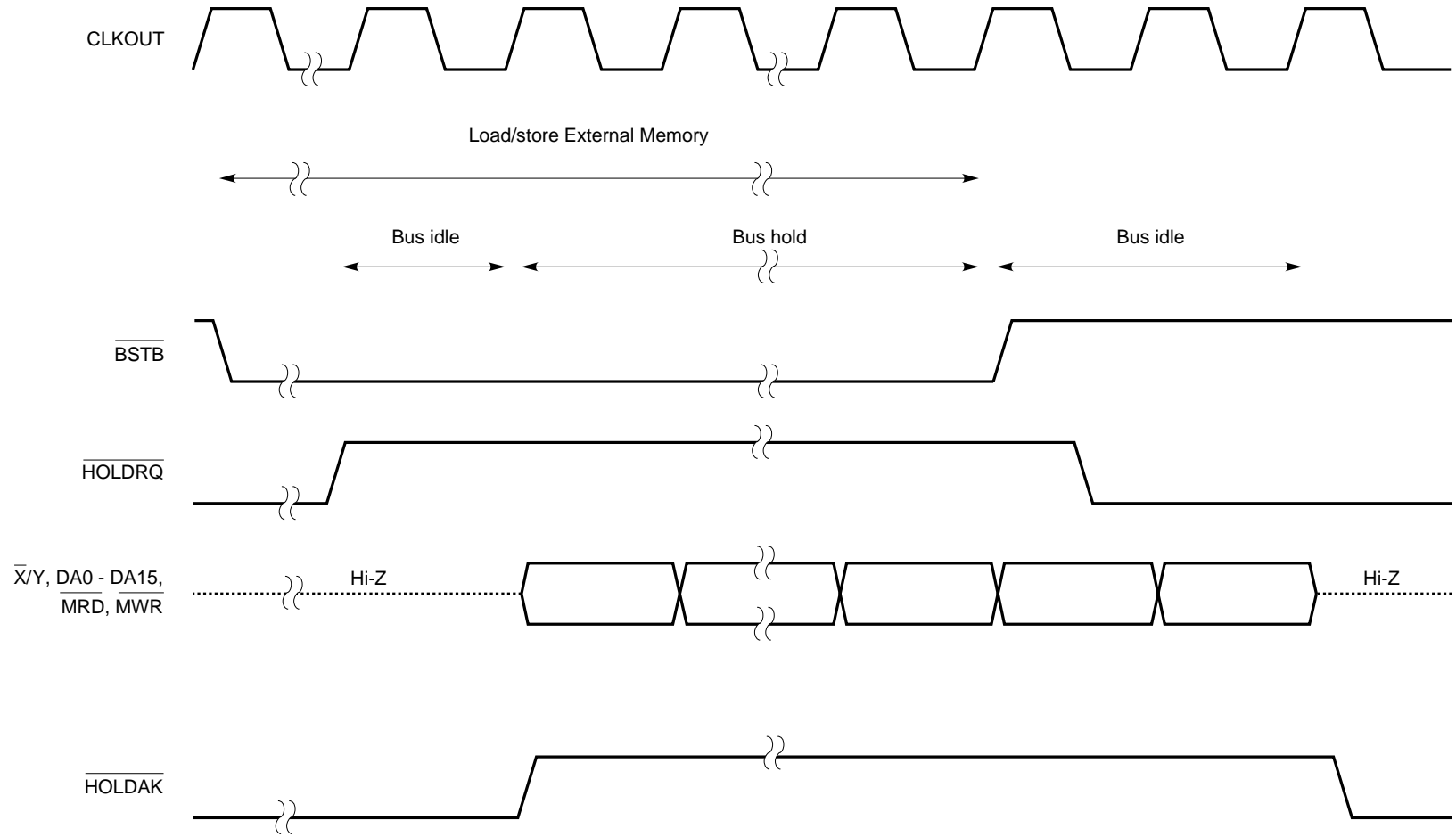
Bus Arbitration Timing (Bus idle)



## Bus Arbitration Timing (Bus busy)



Bus Arbitration Timing (Bus slave)



**Serial Interface**

**Required Timing Condition**

| Parameters                     | Symbol       | Conditions | MIN.       | TYP. | MAX. | Unit |
|--------------------------------|--------------|------------|------------|------|------|------|
| SCK input cycle time           | $t_{cSC}$    |            | $2t_{cCO}$ |      |      | ns   |
| SCK input high/low level width | $t_{wSC}$    |            | 25         |      |      | ns   |
| SCK input rise/fall time       | $t_{rSC}$    |            |            | 3    | 20   | ns   |
| SOEN recovery time             | $t_{recSOE}$ |            | 10         |      |      | ns   |
| SOEN hold time                 | $t_{hSOE}$   |            | 5          |      |      | ns   |
| SIEN recovery time             | $t_{recSIE}$ |            | 10         |      |      | ns   |
| SIEN hold time                 | $t_{hSIE}$   |            | 5          |      |      | ns   |
| SI setup time                  | $t_{suSI}$   |            | 10         |      |      | ns   |
| SI hold time                   | $t_{hSI}$    |            | 0          |      |      | ns   |

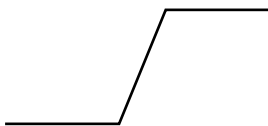
**Switching Characteristics**

| Parameters             | Symbol     | Conditions | MIN. | TYP. | MAX. | Unit |
|------------------------|------------|------------|------|------|------|------|
| SORQ output delay time | $t_{dSOR}$ |            | 0    |      | 30   | ns   |
| SORQ hold time         | $t_{hSOR}$ |            | 0    |      | 30   | ns   |
| SO valid time          | $t_{vSO}$  |            | 0    |      | 30   | ns   |
| SO hold time           | $t_{hSO}$  |            |      |      | 60   | ns   |
| SIAM output delay time | $t_{dSIA}$ |            | 0    |      | 30   | ns   |
| SIAM hold time         | $t_{hSIA}$ |            | 0    |      | 30   | ns   |

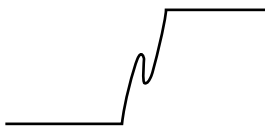
**Notes for Serial Clock**

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.

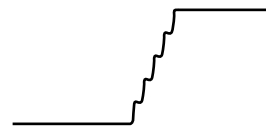
It considers for the serial clock to make a waveform stable especially about the rising and falling.



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



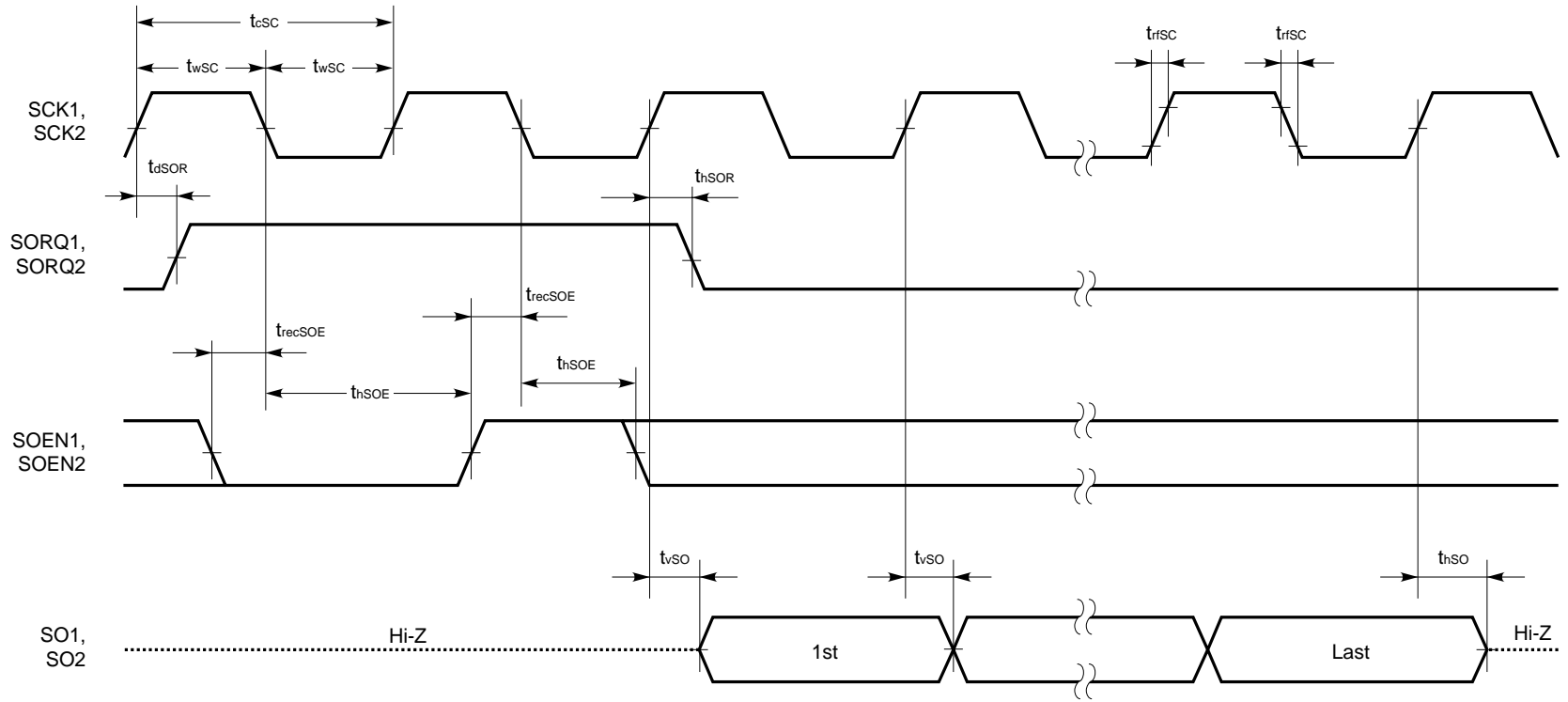
**Example 2. no good example**  
It doesn't bound. It doesn't make noise one above another.



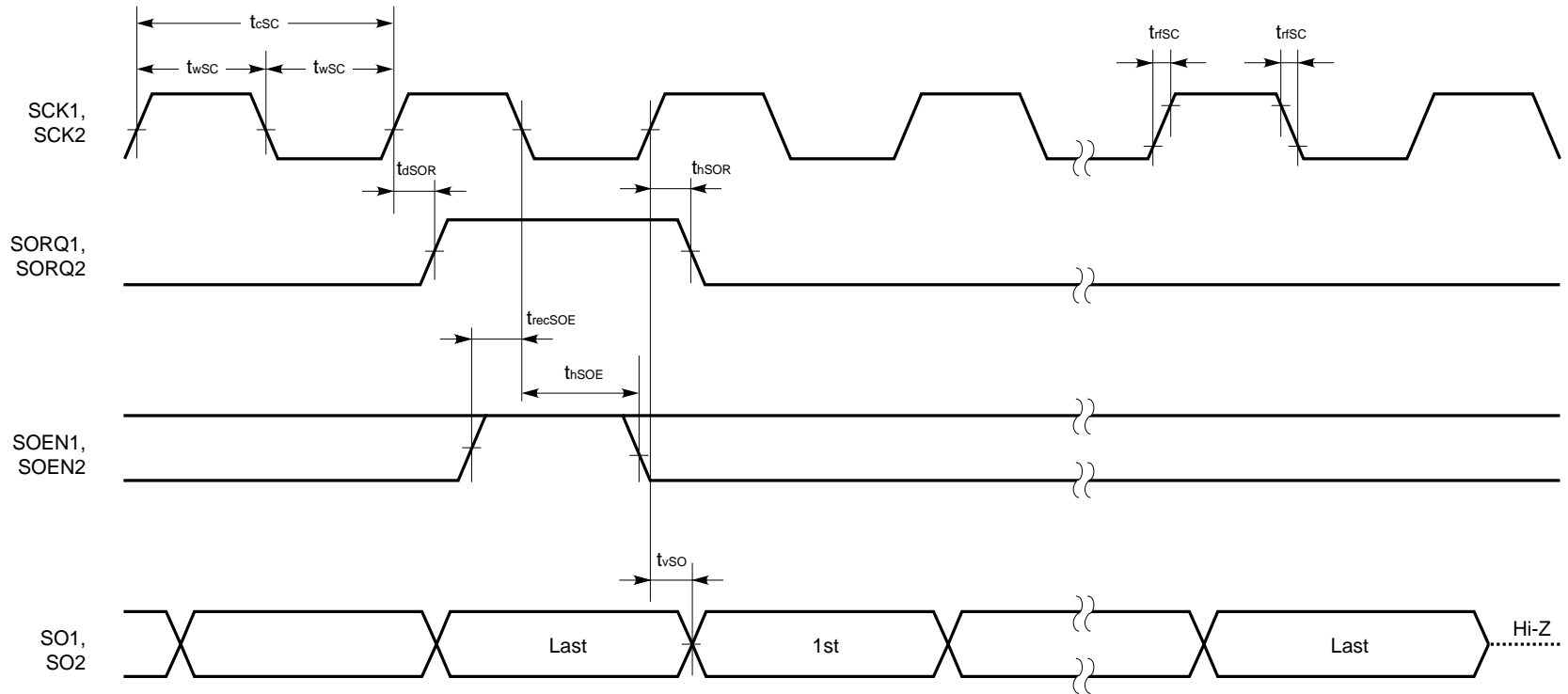
**Example 3. no good example**  
It doesn't make a stair stepping.



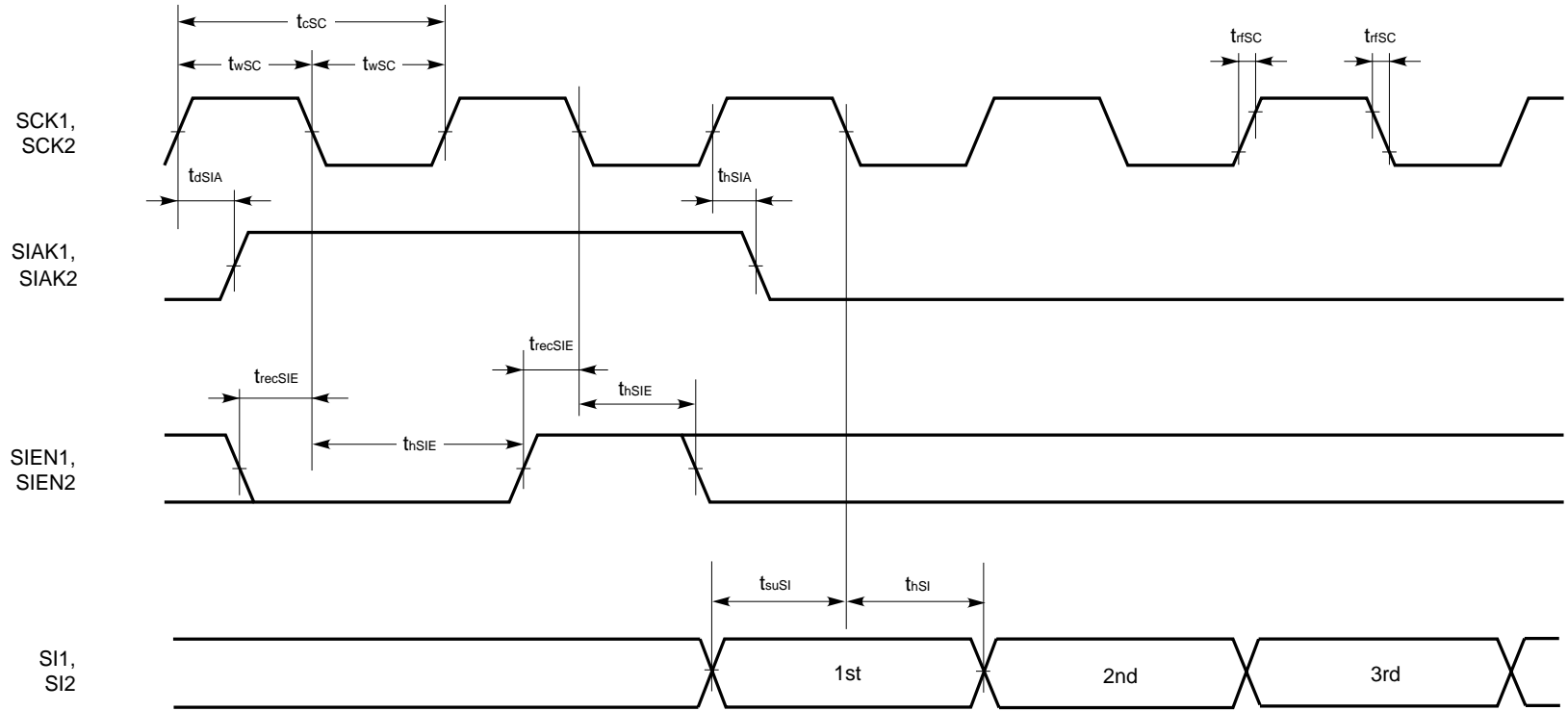
Serial Output Timing 1



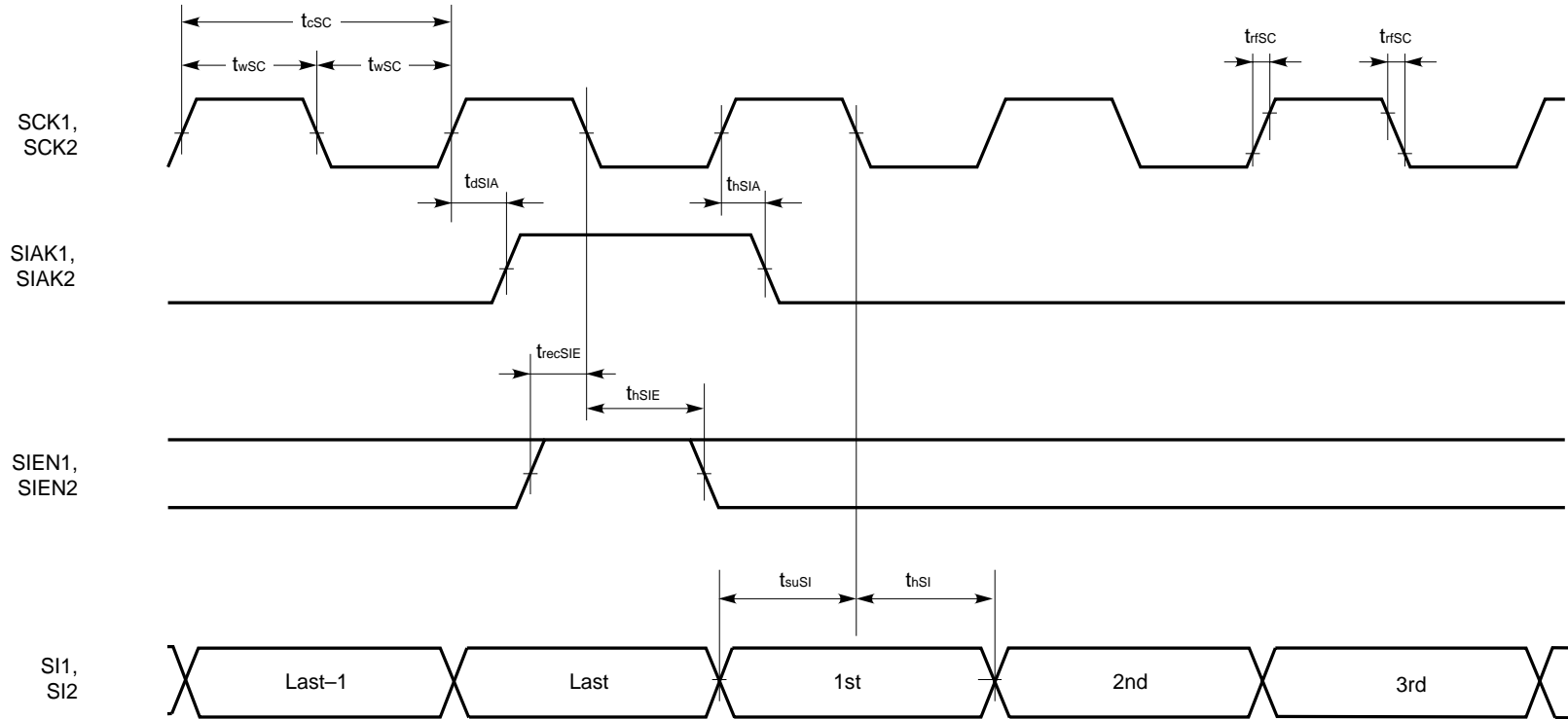
## Serial Output Timing 2 (Continual output)



Serial Input Timing 1



## Serial Input Timing 2 (Continual input)



**Host Interface**

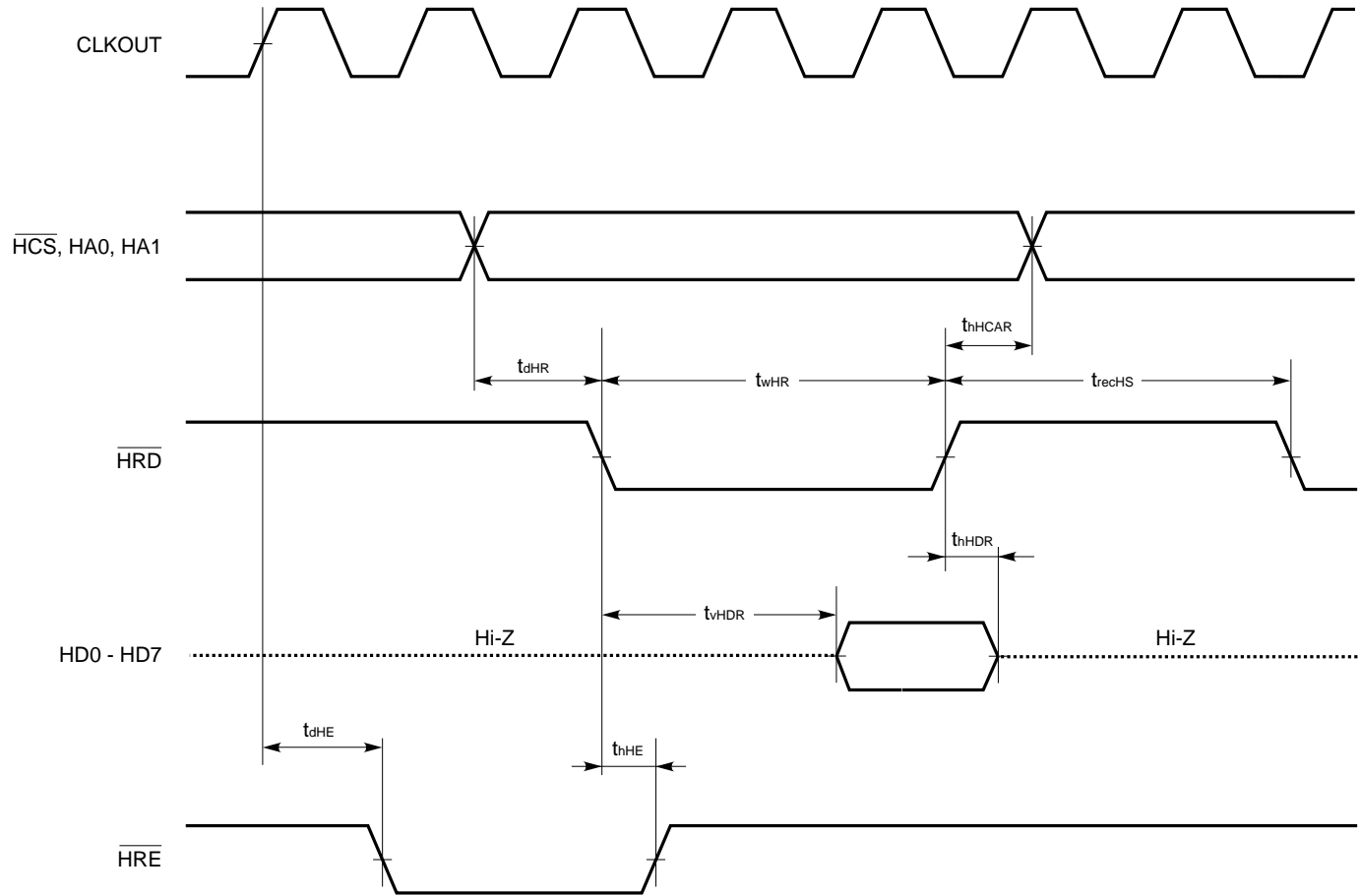
**Required Timing Condition**

| Parameters  | Symbol      | Conditions | MIN.       | TYP. | MAX. | Unit |
|---|-------------|------------|------------|------|------|------|
| $\overline{\text{HRD}}$ delay time                              | $t_{dHR}$   |            | 0          |      |      | ns   |
| $\overline{\text{HRD}}$ width                                   | $t_{wHR}$   |            | $2t_{cCO}$ |      |      | ns   |
| $\overline{\text{HCS}}$ , HA0, HA1 read hold time               | $t_{hCAR}$  |            | 5          |      |      | ns   |
| $\overline{\text{HCS}}$ , HA0, HA1 write hold time              | $t_{hCAW}$  |            | 5          |      |      | ns   |
| $\overline{\text{HRD}}$ , $\overline{\text{HWR}}$ recovery time | $t_{recHS}$ |            | $2t_{cCO}$ |      |      | ns   |
| $\overline{\text{HWR}}$ delay time                              | $t_{dHW}$   |            | 0          |      |      | ns   |
| $\overline{\text{HWR}}$ width                                   | $t_{wHW}$   |            | $2t_{cCO}$ |      |      | ns   |
| $\overline{\text{HWR}}$ hold time                               | $t_{hHDW}$  |            | 5          |      |      | ns   |
| $\overline{\text{HWR}}$ setup time                              | $t_{suHDW}$ |            | 20         |      |      | ns   |

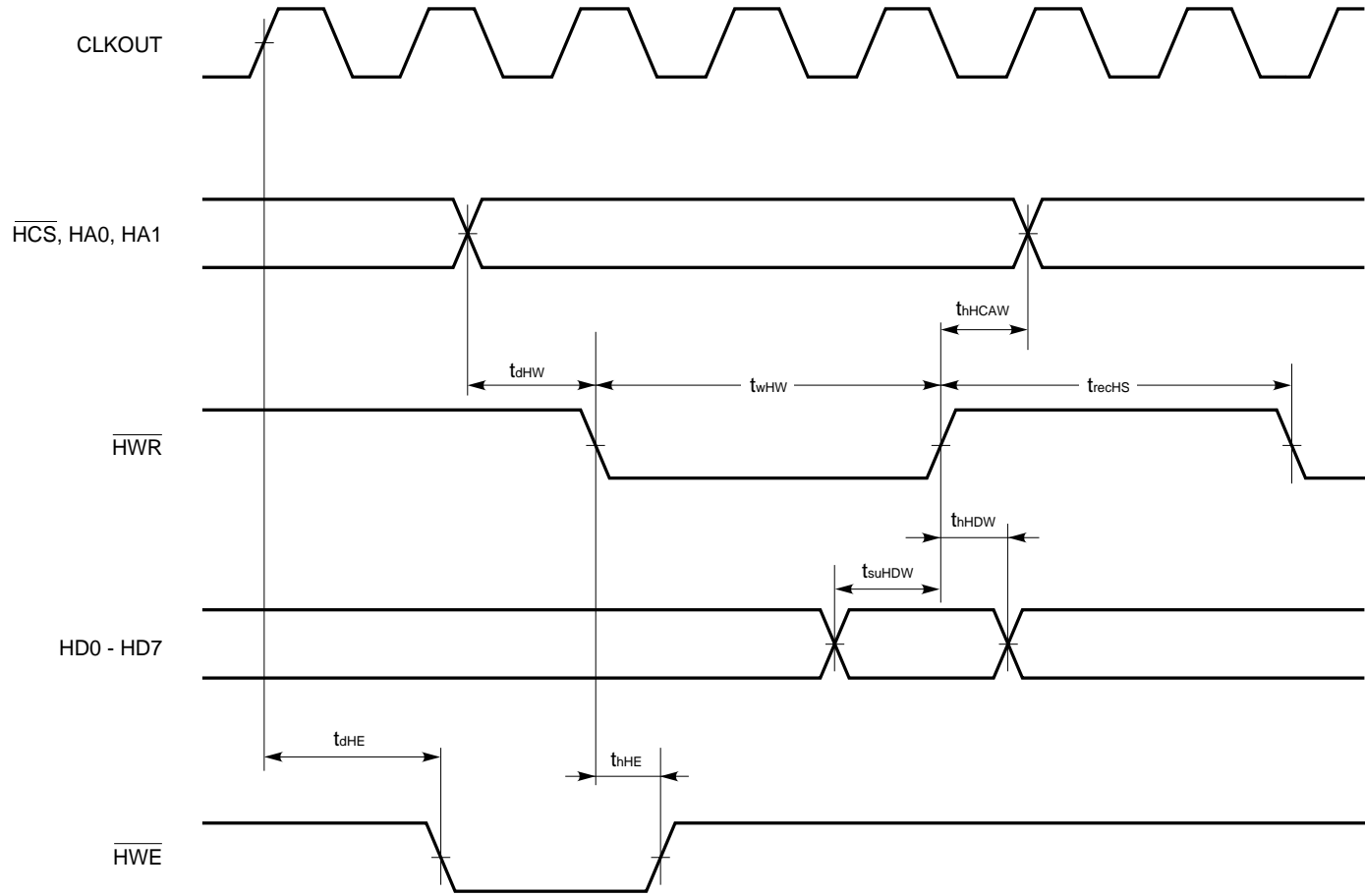
**Switching Characteristics**

| Parameters  | Symbol     | Conditions | MIN. | TYP. | MAX. | Unit |
|---|------------|------------|------|------|------|------|
| $\overline{\text{HRE}}$ , $\overline{\text{HWE}}$ output delay time | $t_{dHE}$  |            |      |      | 30   | ns   |
| $\overline{\text{HRE}}$ , $\overline{\text{HWE}}$ hold time         | $t_{hHE}$  |            |      |      | 20   | ns   |
| $\overline{\text{HRD}}$ valid time                                  | $t_{vHDR}$ |            |      |      | 30   | ns   |
| $\overline{\text{HRD}}$ hold time                                   | $t_{hHDR}$ |            | 0    |      |      | ns   |

## Host Read Interface Timing



Host Write Interface Timing



General Input/Output Ports

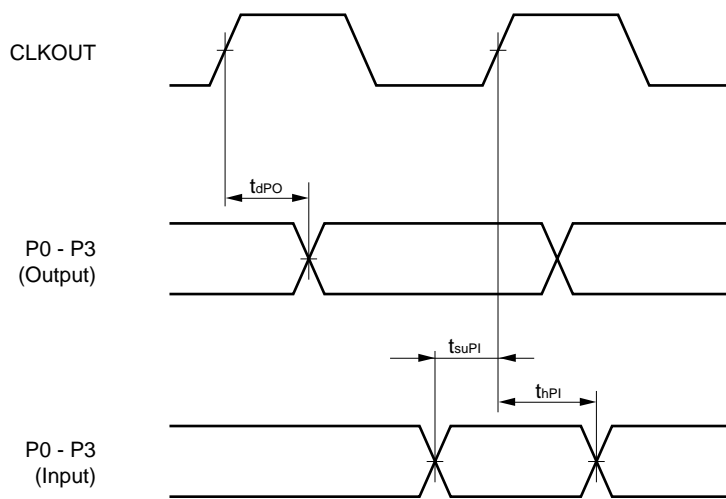
Required Timing Condition

| Parameters            | Symbol     | Conditions | MIN. | TYP. | MAX. | Unit |
|-----------------------|------------|------------|------|------|------|------|
| Port input setup time | $t_{suPI}$ |            | 10   |      |      | 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)

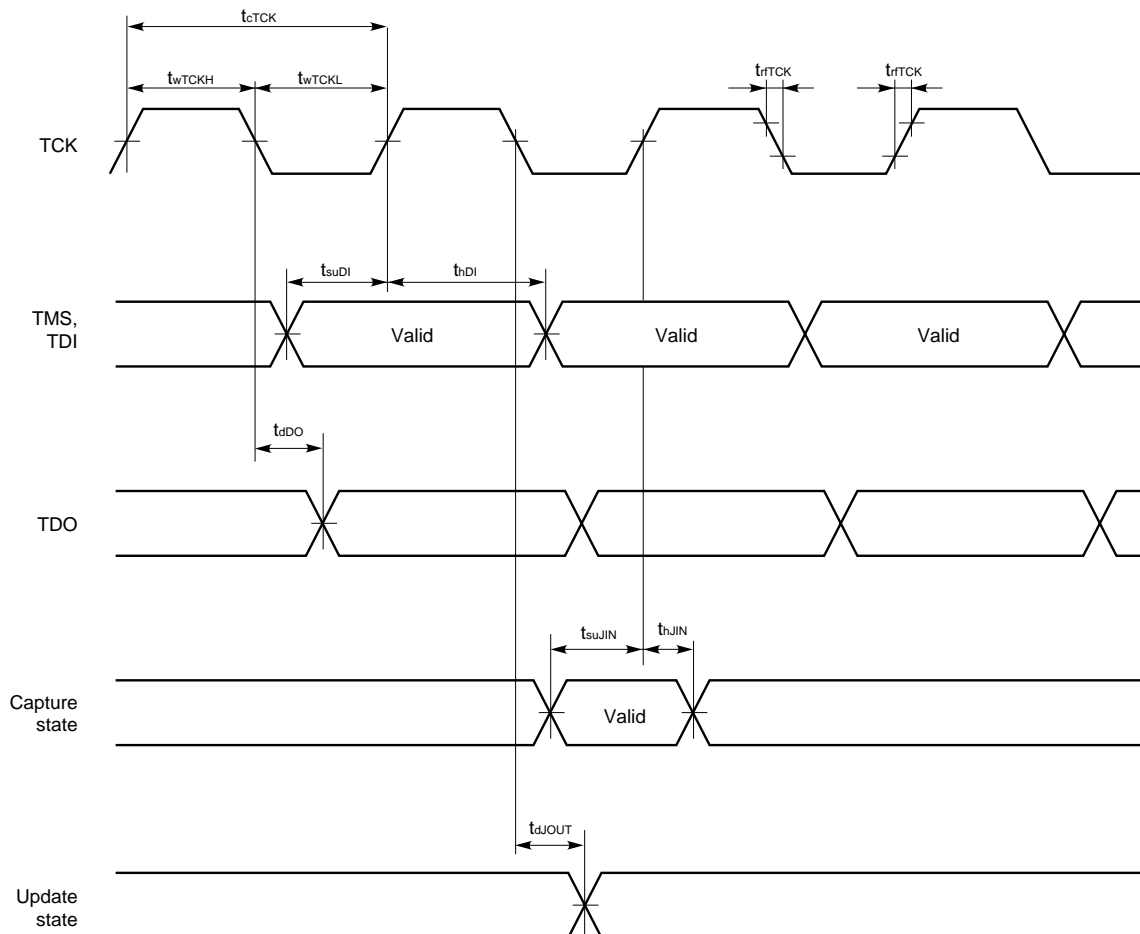
#### Required Timing Condition

| Parameters           | Symbol      | Conditions | MIN.       | TYP. | MAX. | Unit |
|----------------------|-------------|------------|------------|------|------|------|
| TCK cycle time       | $t_{cTCK}$  |            | $4t_{cco}$ |      |      | ns   |
| TCK high level width | $t_{wTCKH}$ |            | 50         |      |      | ns   |
| TCK low level width  | $t_{wTCKL}$ |            | 50         |      |      | ns   |
| TCK rise/fall time   | $t_{rTCK}$  |            |            | 3    | 20   | ns   |
| TMS, TDI setup time  | $t_{suDI}$  |            | 10         |      |      | ns   |
| TMS, TDI hold time   | $t_{hDI}$   |            | 15         |      |      | 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        | $t_{dDO}$   |            | 0    |      | 30   | ns   |
| Output pin output delay time | $t_{dJOUT}$ |            |      |      | 30   | ns   |

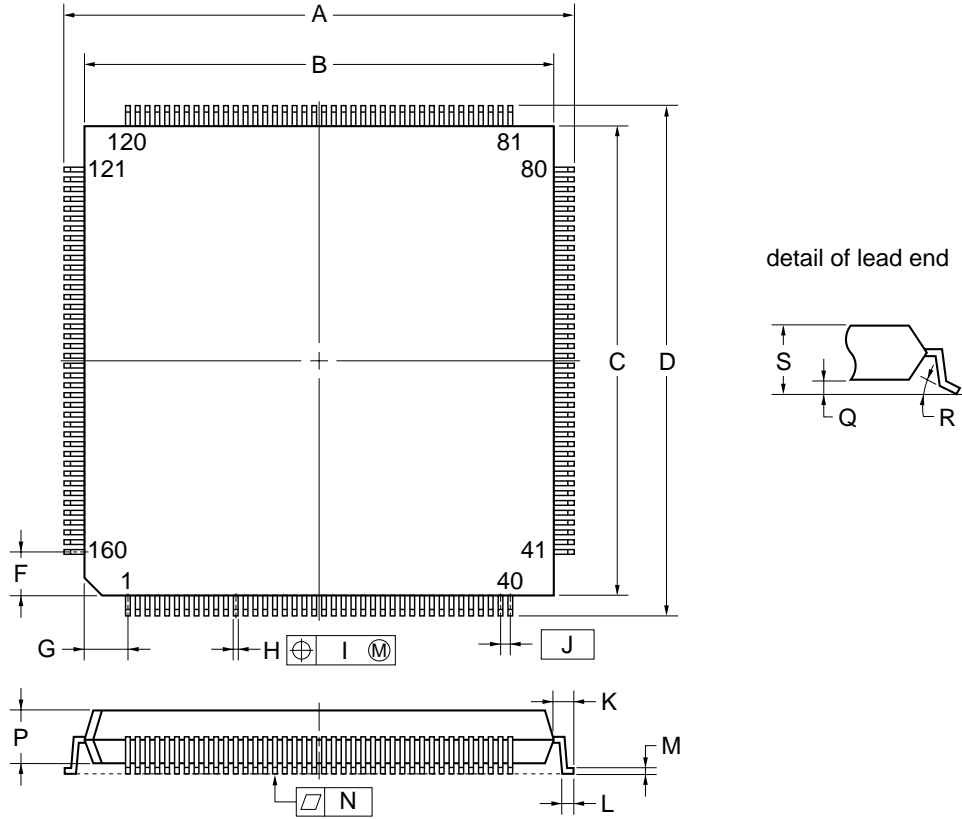
#### Debugging Interface Timing



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

5. PACKAGE DRAWING

160 PIN PLASTIC QFP (FINE PITCH) (□ 24)



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    | 26.0±0.2                               | 1.024 <sup>+0.008</sup> <sub>-0.009</sub> |
| B    | 24.0±0.2                               | 0.945±0.008                               |
| C    | 24.0±0.2                               | 0.945±0.008                               |
| D    | 26.0±0.2                               | 1.024 <sup>+0.008</sup> <sub>-0.009</sub> |
| F    | 2.25                                   | 0.089                                     |
| G    | 2.25                                   | 0.089                                     |
| 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    | 0.17 <sup>+0.03</sup> <sub>-0.07</sub> | 0.007 <sup>+0.001</sup> <sub>-0.003</sub> |
| N    | 0.10                                   | 0.004                                     |
| P    | 2.7                                    | 0.106                                     |
| Q    | 0.4±0.1                                | 0.016 <sup>+0.004</sup> <sub>-0.005</sub> |
| R    | 3° <sup>+7°</sup> <sub>-3°</sub>       | 3° <sup>+7°</sup> <sub>-3°</sub>          |
| S    | 3.3 MAX.                               | 0.130 MAX.                                |

S160GM-50-JMD,KMD

## 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

#### μPD77016GM-KMD: 160-pin plastic QFP (FINE PITCH) (24 × 24 mm)

| Process                | Conditions   | Symbol     |
|------------------------|--|------------|
| Infrared ray reflow    | Peak temperature: 235 °C or below (Package surface temperature),<br>Reflow time: 30 seconds or less (at 210 °C or higher),<br>Maximum number of reflow processes: 1 time,<br>Exposure limit <sup>Note</sup> : 7 days (20 hours pre-baking is required at 125 °C afterwards). | IR35-207-1 |
| VPS                    | Peak temperature: 215 °C or below (Package surface temperature),<br>Reflow time: 40 seconds or less (at 200 °C or higher),<br>Maximum number of reflow processes: 1 time,<br>Exposure limit <sup>Note</sup> : 7 days (20 hours pre-baking is required at 125 °C afterwards). | VP15-207-1 |
| Partial heating method | Pin temperature: 300 °C or below,<br>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.

[MEMO]

[MEMO]

[MEMO]

## 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 device 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.