# NEC

## MOS Integrated Circuit
# $\mu$**PD70732**

## V810™
## 32-BIT MICROPROCESSOR

The μPD70732 (a.k.a. V810) microprocessor is NEC's first microprocessor of the V810 family™ for embedded control applications.

The V810 employs a RISC architecture for embedded control applications.  This product has high-speed real time response, high-speed integer operation instruction, bit string instruction, floating-point operation instruction, and significantly high cost performance is realized for applications such as facsimile, digital PPC, word processor, image processor, real time control device, etc.

**The functions are described in detail in the following User's Manuals, which should be read before starting design work.**
- **V805™, V810 User's Manual Hardware     : U10661E**
- **V810 Family User's Manual Architecture : U10082E**

**Features**

○ High-performance 32-bit architecture for embedded control application
- 32-bit separate address/data bus
- 1-Kbyte cache memory
- Pipeline structure of 1 clock pitch
- 16-bit fixed instructions (with some exceptions)
- 32-bit general-purpose registers: 32
- 4-Gbyte linear address space
- Register/flag hazard interlocked by hardware

○ Dynamic bus sizing function (16 bits)

○ 16-bit bus fixing function

   16-bit bus system can be configured.

○ Instructions ideal for various application fields
- Floating-point operation instructions (based upon IEEE754 data format)
- Bit string instructions

○ 16 levels of high-speed interrupt responses

○ Clock can be stopped by internal static operation

○ Maximum operating frequency: 16/20/25 MHz

○ Low voltage:  $V_{DD}$ = 2.7 to 3.6 V (Max. 16 MHz)

   $V_{DD}$ = 2.2 to 3.6 V (Max. 10 MHz)

○ Small package versions available (14 x 14 mm fine-pitch TQFP)                                  ★

---

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

**Ordering Information**

| Part Number | Package | Max. operating freq. (MHz) |
|---|---|---|
| µPD70732GD-16-LBB | 120-pin plastic QFP (28 x 28 mm) | 16 |
| µPD70732GD-20-LBB | 120-pin plastic QFP (28 x 28 mm) | 20 |
| µPD70732GD-25-LBB | 120-pin plastic QFP (28 x 28 mm) | 25 |
| ★ µPD70732GC-25-9EV | 120-pin plastic TQFP (Fine pitch) (14 x 14 mm) | 25 |
| µPD70732R-25 | 176-pin ceramic PGA (Seam weld) | 25 |

**Pin Outline**
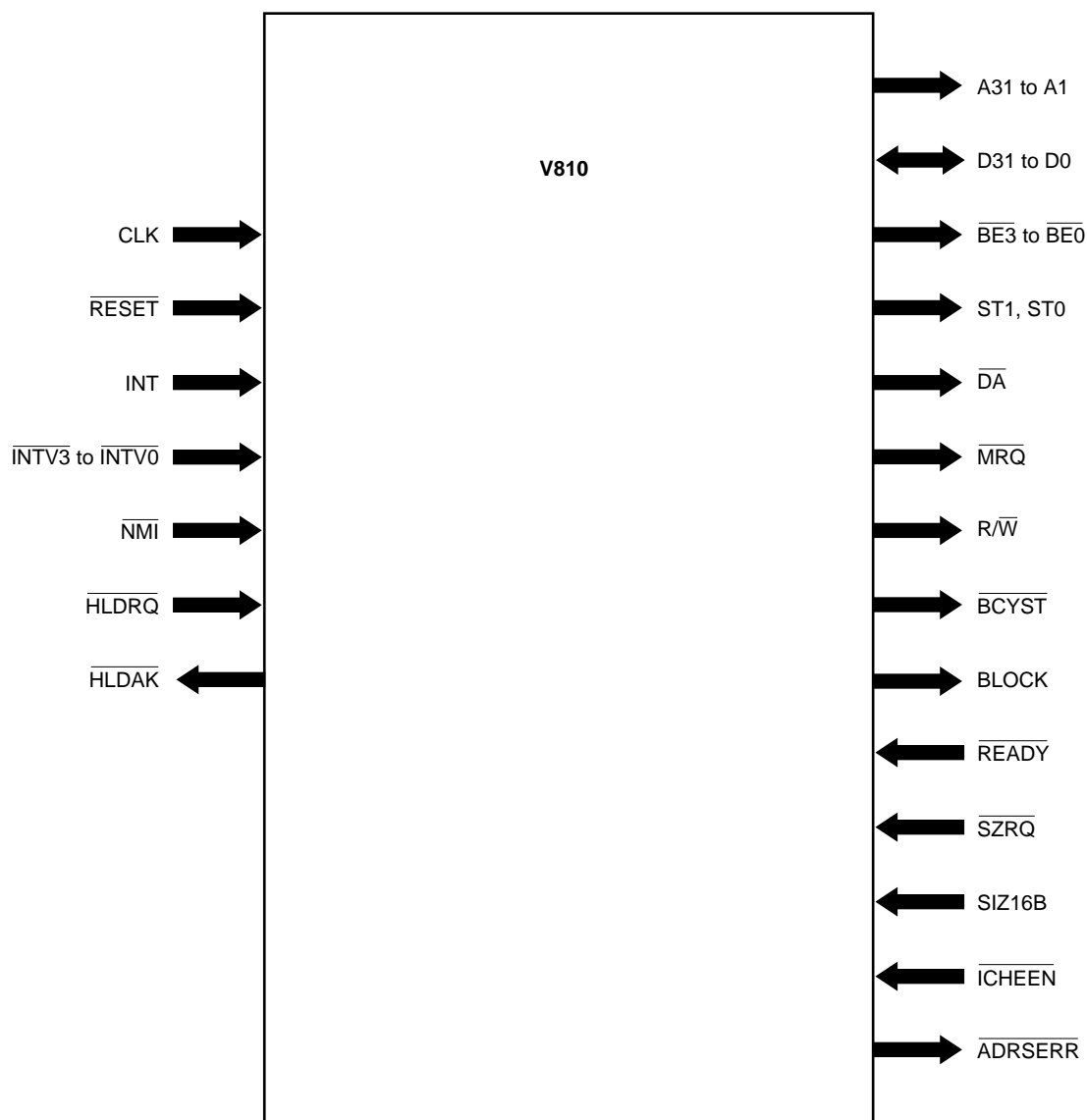
**Pin Configuration**

• **120-pin plastic QFP (28 x 28 mm) (Top View)**
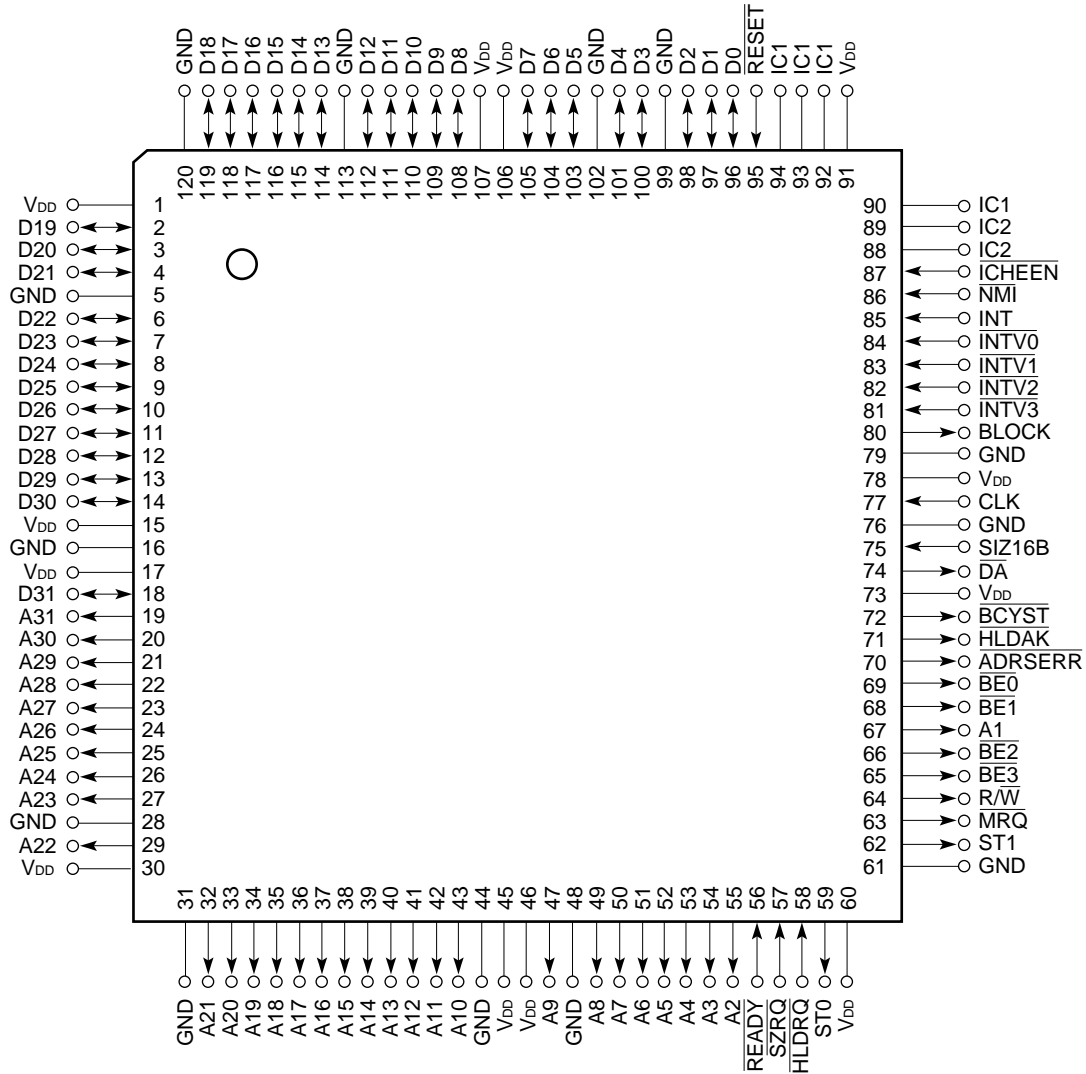
μPD70732GD-xx-LBB
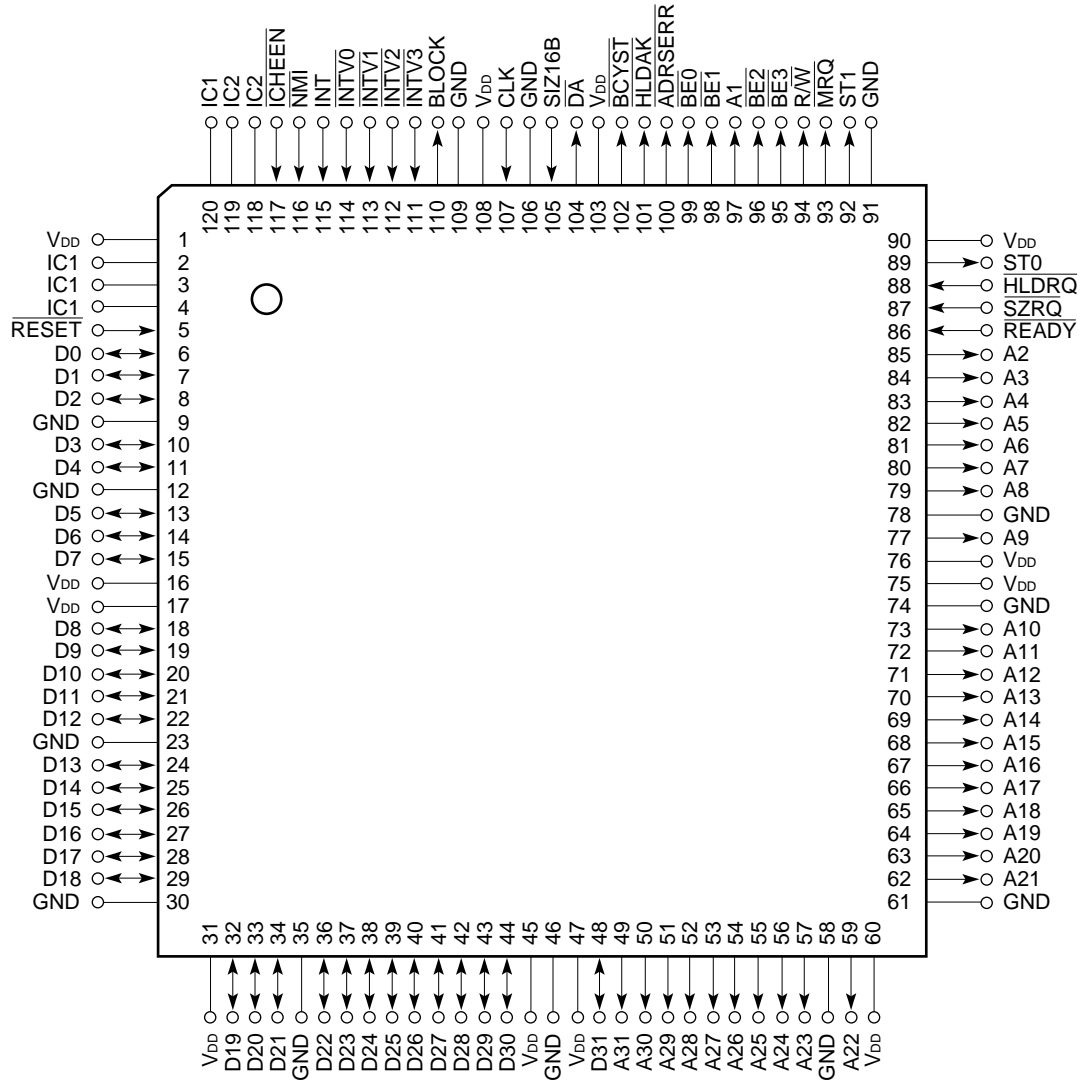


**Cautions 1. Leave the IC1 pin open.**

**2. Connect the IC2 pin to GND.**
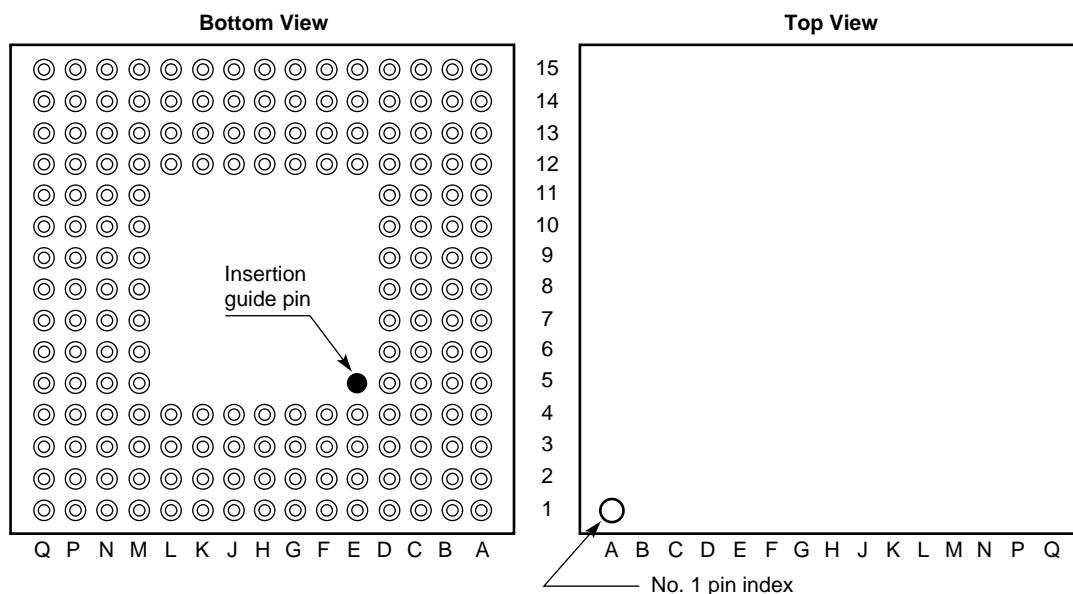
**Remark** IC: Internally Connected

★ • **120-pin plastic TQFP (Fine pitch) (14 x 14 mm) (Top View)**

μPD70732GC-25-9EV



**Cautions 1.** V_DD is power supply pin. All V_DD pins should be connected to a +5V power supply (the same power supply).

**2.** GND is ground pin. All GND pins should be connected to the same GND.

**3.** Leave the IC1 pin open.

**4.** Connect the IC2 pin to GND.

**Remark** IC: Internally Connected

- **176-pin ceramic PGA (Seam weld)**
  µPD70732R-25

**Bottom View**                                                    **Top View**

Insertion guide pin

The insertion guide pin is not included in the number of pins.

No. 1 pin index

Q P N M L K J H G F E D C B A          A B C D E F G H J K L M N P Q

**Remark**   The insertion guide pin is not included in the number of pins.

| No. | Signal | No. | Signal | No. | Signal | No. | Signal |
|-----|--------|-----|--------|-----|--------|-----|--------|
| A1 | IC2 | B3 | GND | C5 | $V_{DD}$ | D7 | $V_{DD}$ |
| A2 | D12 | B4 | D11 | C6 | D8 | D8 | $V_{DD}$ |
| A3 | D13 | B5 | GND | C7 | $V_{DD}$ | D9 | GND |
| A4 | D10 | B6 | D7 | C8 | D4 | D10 | IC3 |
| A5 | GND | B7 | $V_{DD}$ | C9 | D2 | D11 | IC2 |
| A6 | D6 | B8 | D3 | C10 | IC3 | D12 | GND |
| A7 | IC2 | B9 | GND | C11 | $V_{DD}$ | D13 | INT |
| A8 | D5 | B10 | D0 | C12 | IC1 | D14 | $\overline{INTV1}$ |
| A9 | IC2 | B11 | GND | C13 | IC2 | D15 | GND |
| A10 | D1 | B12 | IC1 | C14 | $V_{DD}$ | E1 | D27 |
| A11 | $V_{DD}$ | B13 | GND | C15 | $\overline{NMI}$ | E2 | D25 |
| A12 | $\overline{RESET}$ | B14 | IC1 | D1 | D23 | E3 | D21 |
| A13 | IC1 | B15 | $\overline{ICHEEN}$ | D2 | D22 | E4 | D19 |
| A14 | IC1 | C1 | $V_{DD}$ | D3 | D20 | E12 | IC3 |
| A15 | IC2 | C2 | $V_{DD}$ | D4 | GND | E13 | $\overline{INTV0}$ |
| B1 | D17 | C3 | D16 | D5 | D15 | E14 | IC3 |
| B2 | D18 | C4 | D14 | D6 | D9 | E15 | IC1 |

| No. | Signal | No. | Signal | No. | Signal | No. | Signal |
|---|---|---|---|---|---|---|---|
| F1 | $V_{DD}$ | J4 | $V_{DD}$ | M7 | $V_{DD}$ | P4 | A12 |
| F2 | D26 | J12 | IC2 | M8 | A5 | P5 | GND |
| F3 | D24 | J13 | IC2 | M9 | $V_{DD}$ | P6 | A8 |
| F4 | GND | J14 | IC1 | M10 | ST1 | P7 | GND |
| F12 | $\overline{INTV2}$ | J15 | IC1 | M11 | A1 | P8 | A6 |
| F13 | $\overline{INTV3}$ | K1 | IC2 | M12 | GND | P9 | GND |
| F14 | $V_{DD}$ | K2 | A27 | M13 | $\overline{BCYST}$ | P10 | $\overline{SZRQ}$ |
| F15 | GND | K3 | A25 | M14 | $\overline{DA}$ | P11 | GND |
| G1 | D29 | K4 | A24 | M15 | SIZ16B | P12 | $\overline{MRQ}$ |
| G2 | D28 | K12 | GND | N1 | $V_{DD}$ | P13 | GND |
| G3 | IC2 | K13 | BLOCK | N2 | $V_{DD}$ | P14 | $\overline{ADRSERR}$ |
| G4 | IC2 | K14 | $V_{DD}$ | N3 | A17 | P15 | $\overline{BE0}$ |
| G12 | $V_{DD}$ | K15 | $V_{DD}$ | N4 | A15 | Q1 | IC2 |
| G13 | IC2 | L1 | A28 | N5 | $V_{DD}$ | Q2 | A13 |
| G14 | IC1 | L2 | A26 | N6 | A9 | Q3 | A14 |
| G15 | IC1 | L3 | A22 | N7 | $V_{DD}$ | Q4 | A11 |
| H1 | A31 | L4 | A20 | N8 | $V_{DD}$ | Q5 | GND |
| H2 | D30 | L12 | $\overline{HLDAK}$ | N9 | A3 | Q6 | A7 |
| H3 | GND | L13 | $V_{DD}$ | N10 | $\overline{HLDRQ}$ | Q7 | IC2 |
| H4 | D31 | L14 | IC1 | N11 | $V_{DD}$ | Q8 | A4 |
| H12 | GND | L15 | IC1 | N12 | $\overline{BE2}$ | Q9 | IC2 |
| H13 | CLK | M1 | GND | N13 | $\overline{BE1}$ | Q10 | A2 |
| H14 | IC1 | M2 | A23 | N14 | $V_{DD}$ | Q11 | $\overline{READY}$ |
| H15 | IC2 | M3 | A21 | N15 | IC1 | Q12 | ST0 |
| J1 | A30 | M4 | GND | P1 | A18 | Q13 | $\overline{BE3}$ |
| J2 | A29 | M5 | A16 | P2 | A19 | Q14 | $R/\overline{W}$ |
| J3 | IC2 | M6 | A10 | P3 | GND | Q15 | IC2 |

**Cautions 1. Leave the IC1 pin open.**

**2. Connect the IC2 pin to GND.**

**3. Connect the IC3 pin to power supply.**

**Remark** IC: Internally Connected

# CONTENTS

# 1. PIN FUNCTIONS

## 1.1 Pin Function List

| Name | I/O | Function | Bus hold status during operation | Bus hold status at reset | Bus idle status at reset |
|---|---|---|---|---|---|
| A31 to A1 (Address Bus) | 3-state output | Address bus | Hi-Z | Hi-Z | H[Note] |
| D31 to D0 (Data Bus) | 3-state I/O | Bidirectional data bus | Hi-Z | Hi-Z | Hi-Z |
| $\overline{BE3}$ to $\overline{BE0}$ (Byte Enable) | 3-state output | Indicates valid data bus when data is accessed | Hi-Z | Hi-Z | H |
| ST1, ST0 (Status) | 3-state output | Indicates type of bus cycle | Hi-Z | Hi-Z | H |
| $\overline{DA}$ (Data Access) | 3-state output | Strobe signal for bus cycle | Hi-Z | Hi-Z | H |
| $\overline{MRQ}$ (Memory Request) | 3-state output | Indicates memory access | Hi-Z | Hi-Z | H |
| R/$\overline{W}$ (Read/Write) | 3-state output | Distinguishes between read access and write access | Hi-Z | Hi-Z | H |
| $\overline{BCYST}$ (Bus Cycle Start) | 3-state output | Indicates start of bus cycle | Hi-Z | Hi-Z | H |
| $\overline{READY}$ (Ready) | Input | Extends bus cycle | — | — | — |
| $\overline{HLDRQ}$ (Hold Request) | Input | Requests bus mastership | — | — | — |
| $\overline{HLDAK}$ (Hold Acknowledge) | Output | Acknowledges $\overline{HLDRQ}$ | L | L | H |
| $\overline{SZRQ}$ (Bus Sizing Request) | Input | Requests bus sizing | — | — | — |
| SIZ16B (Bus Size 16 Bit) | Input | Fixes external data bus width to 16 bits | — | — | — |
| BLOCK (Bus Lock) | Output | Requests to inhibit use of bus | L | L | L |
| $\overline{ICHEEN}$ (Instruction Cache Enable) | Input | Operates instruction cache | — | — | — |
| INT (Maskable Interrupt) | Input | Interrupt request | — | — | — |
| $\overline{INTV3}$ to $\overline{INTV0}$ (Interrupt Level) | Input | Interrupt level | — | — | — |

**Note**    A1 pin is "H" in the 16-bit bus fixed mode; otherwise, it is "L".

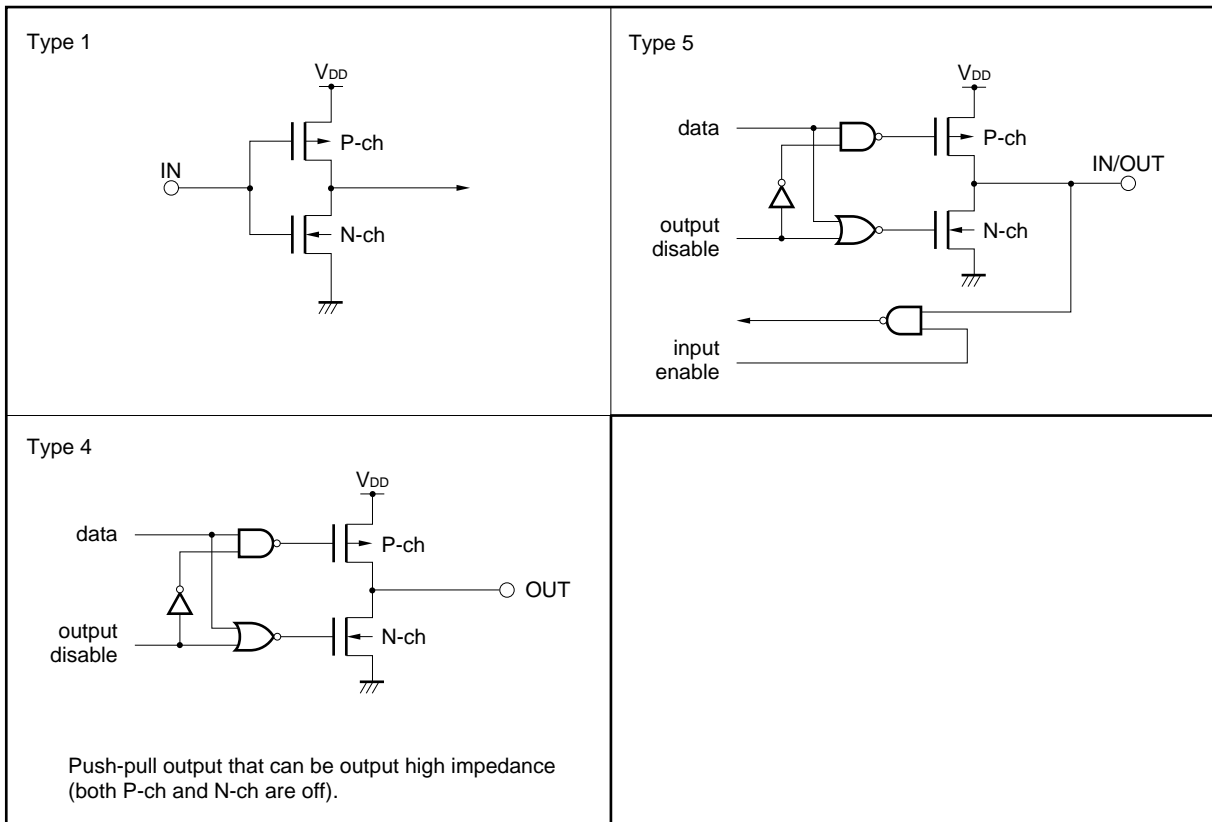| Name | I/O | Function | Bus hold status during operation | Bus hold status at reset | Bus idle status at reset |
|------|-----|----------|------------------|---------------|---------------|
| $\overline{\text{NMI}}$ (Non-Maskable Interrupt) | Input | Non-maskable interrupt request | — | — | — |
| CLK | Input | CPU clock input | — | — | — |
| $\overline{\text{RESET}}$ (Reset) | Input | Resets internal status | — | — | — |
| $\overline{\text{ADRSERR}}$ (Address Error) | Output | Indicates that data alignment is illegal | Not affected | H | H |
| $V_{DD}$ (Power Supply) | — | Positive power supply | — | — | — |
| GND (Ground) | — | Ground potential (0 V) | — | — | — |
| IC1 (Internally Connected 1) | — | Internally connected (Leave this pin open.) | — | — | — |
| IC2 (Internally Connected 2) | — | Internally connected (Ground this pin.) | — | — | — |
| IC3 (Internally Connected 3) | — | Internally connected (Connect this pin to power supply.) | — | — | — |

★    **1.2   Pin I/O Circuits and Recommended Connection of Unused Pins**

The I/O circuit type of each pin and recommended connection of unused pins are shown in Table 1-1.  Figure
1-1 shows the I/O circuit of each type.

**Table 1-1.  Pin I/O Circuit Types and Recommended Connection Method of Unused Pins**

| Pin | I/O Circuit Type | Recommended Connection Method |
|---|---|---|
| D31 to D0 | 5 | Open |
| A31 to A1 | 4 | |
| $\overline{BE3}$ to $\overline{BE0}$ | | |
| ST1, ST0 | | |
| $\overline{DA}$ | | |
| $\overline{MRQ}$ | | |
| R/$\overline{W}$ | | |
| $\overline{BCYST}$ | | |
| $\overline{READY}$ | 1 | Connect to GND via resistor |
| $\overline{HLDRQ}$ | | Connect to $V_{DD}$ via resistor |
| $\overline{HLDAK}$ | 4 | Open |
| $\overline{SZRQ}$ | 1 | Connect to $V_{DD}$ via resistor |
| SIZ16B | | Connect to GND via resistor |
| BLOCK | 4 | Open |
| $\overline{ICHEEN}$ | 1 | Connect to $V_{DD}$ via resistor |
| INT | | Connect to GND via resistor |
| $\overline{INTV3}$ to $\overline{INTV0}$ | | Connect to $V_{DD}$ via resistor |
| $\overline{NMI}$ | | |
| CLK | | — |
| $\overline{RESET}$ | | |
| $\overline{ADRSERR}$ | 4 | Open |
| IC1 | — | |
| IC2 | — | Connect to GND |
| IC3 | — | Connect to $V_{DD}$ |

**Figure 1-1.  Pin I/O Circuit**



Type 1

Type 5

Type 4

Push-pull output that can be output high impedance
(both P-ch and N-ch are off).

## ★ 2. REGISTER SET

The registers of the V810 can be classified into two types: general-purpose program register set and dedicated system register set.  All registers are 32 bits wide.

**Program register sets**

31                                              0

| Reg | Description |
|-----|-------------|
| r0 | Zero Register |
| r1 | Reserved for Address Generation |
| r2 | Handler Stack Pointer (hp) |
| r3 | Stack Pointer (sp) |
| r4 | Global Pointer (gp) |
| r5 | Text Pointer (tp) |
| r6 | |
| r7 | |
| r8 | |
| r9 | |
| r10 | |
| r11 | |
| r12 | |
| r13 | |
| r14 | |
| r15 | |
| r16 | |
| r17 | |
| r18 | |
| r19 | |
| r20 | |
| r21 | |
| r22 | |
| r23 | |
| r24 | |
| r25 | |
| r26 | String Destination Bit Offset |
| r27 | String Source Bit Offset |
| r28 | String Length |
| r29 | String Destination |
| r30 | String Source |
| r31 | Link Pointer (lp) |

31                                              0

| Reg | Description |
|-----|-------------|
| PC | Program Counter |

**System register sets**

31                                              0

| Reg | Description |
|-----|-------------|
| EIPC | Exception/Interrupt PC |
| EIPSW | Exception/Interrupt PSW |

31   0

| Reg | Description |
|-----|-------------|
| FEPC | Fatal Error PC |
| FEPSW | Fatal Error PSW |

31   0

| Reg | Description |
|-----|-------------|
| ECR | Exception Cause Register |

31   0

| Reg | Description |
|-----|-------------|
| PSW | Program Status Word |

31   0

| Reg | Description |
|-----|-------------|
| PIR | Processor ID Register |

31   0

| Reg | Description |
|-----|-------------|
| TKCW | Task Control Word |

31   0

| Reg | Description |
|-----|-------------|
| CHCW | Cache Control Word |

31   0

| Reg | Description |
|-----|-------------|
| ADTRE | Address Trap Register |

## 2.1  Program Register Set

The program register set is composed of general-purpose registers and a program counter.

### (1)  General-purpose registers

Thirty-two general-purpose registers, r0 to r31, are available.  All these registers can be used as data registers or address registers.

Of these registers, r0 and r26 through r30 are implicitly used by some instructions, and r1 through r5 and r31 are implicitly used by the assembler and C compiler.  Therefore, when using these registers, it is necessary to take special care such as saving these registers' contents to different areas before using these registers and restoring the contents after using them.

**Table 2-1.  Program Registers**

| Register | Application | Operation |
|---|---|---|
| r0 | Zero register | Always holds zeros. |
| r1 | Register reserved for assembler | Used as a working register to generate a 32-bit immediate data. |
| r2 | Handler stack pointer | Used as the stack pointer for the handler. |
| r3 | Stack pointer | Used to generate a stack frame at a function call. |
| r4 | Global pointer | Used to access a global variable in the data area. |
| r5 | Text pointer | Points the start address of the text area. |
| r6 to r25 | — | Stores address or data variables. |
| r26 | String destination bit offset | Used in a bit-string instruction execution. |
| r27 | String source bit offset | |
| r28 | String length register | |
| r29 | String destination address register | |
| r30 | String address register | |
| r31 | Link pointer | Stores the return address at execution of a JAL instruction. |

### (2)  Program Counter

The program counter (PC) indicates the address of the instruction currently executed by the program. Bit 0 of the PC is fixed to 0, and execution cannot branch to an odd address.  The contents of the PC are initialized to FFFFFFF0H at reset.

### 2.2 System Register Set

The system register set is composed of the following registers that perform operations such as CPU-status control and interrupt information holding.

**Table 2-2.  System Register Number**

| Number | Register Name | Application | Operation |
|---|---|---|---|
| 0 | EIPC | Status saving registers for exception/interrupt | The EIPC and EIPSW registers save the PC and PSW, respectively, when an exception or interrupt occurs.  Because in the V810 the registers incorporated for this purpose are these registers only, save the contents of these registers by means of programming if your application set can cause multiple interrupt requests to be issued in the V810. |
| 1 | EIPSW | | |
| 2 | FEPC | Status saving registers for NMI/duplexed exception | The FEPC and FEPSW registers save the PC and PSW, respectively, when an NMI or duplexed exception occurs. |
| 3 | FEPSW | | |
| 4 | ECR | Exception cause register | This register, when an exception, maskable interrupt, or NMI occurs, holds its cause.  This register consists of 32 bits. Its higher 16 bits, called FECC, hold the exception code for an NMI or duplexed exception, while the lower 16 bits, called EICC, hold the exception code for an exception or maskable interrupt. |
| 5 | PSW | Program status word | This register, also called the program status word, is a set of flags indicating the statuses of the CPU and program (instruction execution results). |
| 6 | PIR | Processor ID register | This register identifies the CPU type number. |
| 7 | TKCW | Task control word | This register controls floating-point operations. |
| 8 to 23 | Reserved | | |
| 24 | CHCW | Cache control word | This register controls the on-chip instruction cache. |
| 25 | ADTRE | Address trap register | This register holds an address and is used for address trapping. When the address in this register matches the PC value, the execution jumps to a predefined address. |
| 26 to 31 | Reserved | | |

To read or write one of the registers shown above, specify a system register number with the system register load (LDSR) or system register store (STSR) instruction.

## 3. DATA TYPES ★

### 3.1 Data Types

The data types supported by the V810 are as follows:

- Integer (8, 16, 32 bits)
- Unsigned integer (8, 16, 32 bits)
- Bit string
- Single-precision floating-point data (32 bits)

### 3.1.1 Data type and addressing

The V810 uses the little-endian data addressing. In this addressing, if a fixed-length data is located in a memory area, the data must be either of the data types shown below.
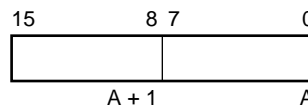
**(1)  Byte**

A byte is a consecutive 8-bit data whose first-bit address is aligned to a byte boundary. Each bit in a byte is numbered from 0 to 7: LSB (the least significant bit) is bit 0 and MSB (the most significant bit) is bit 7. To access a byte, specify address A. (See diagram below.)
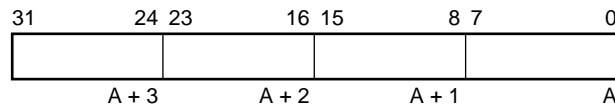
```
7           0
┌───────────┐
│           │
└───────────┘
      A
```
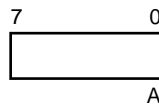
**(2)  Halfword**

A halfword is a consecutive 16-bit (= 2 bytes) data whose first-bit address is aligned to a halfword boundary. Each bit in a halfword is numbered from 0 to 15: LSB (the least significant bit) is bit 0 and MSB (the most significant bit) is bit 15. To access a halfword, specify the address A only (lowest bit must be 0).

```
15        8 7         0
┌──────────┬──────────┐
│          │          │
└──────────┴──────────┘
   A + 1        A
```

**(3)  Word/short real**

A word, also called short real, is a consecutive 32-bit (= 4 bytes) data whose first-bit address is aligned to a word boundary. Each bit in a word is numbered from 0 to 31: LSB (the least significant bit) is bit 0 and MSB (the most significant bit) is bit 31. To access a word or short real, specify the address A only (lower two bits must be 0).

```
31       24 23      16 15       8 7        0
┌──────────┬──────────┬──────────┬──────────┐
│          │          │          │          │
└──────────┴──────────┴──────────┴──────────┘
   A + 3       A + 2      A + 1        A
```

### 3.1.2  Integer

In the V810, all integers are expressed in the two's-complement binary notation, and are composed of either 8 bits, 16 bits, or 32 bits.  Regardless of the data length, bit 0 is the least significant bit, and higher-numbered bits express higher digits of the integer with the highest bit expressing its sign.

| Data Length | | Range |
|---|---|---|
| Byte | 8 bits | −128 to +127 |
| Halfword | 16 bits | −32768 to +32767 |
| Word | 32 bits | −2147483648 to +2147483647 |

### 3.1.3  Unsigned integer

An unsigned integer is either zero or a positive integer unlike the integer explained in section 3.1.2 which can be negative as well as zero and positive.  Unsigned integers are expressed in the binary notation in the same way as integers, and are either 8 bits, 16 bits, or 32 bits long.  Regardless of the data length, the bit assignments are the same as in the case of integers except that unsigned integers do not include a sign bit; the highest bit is also a part of the integer.

| Data Length | | Range |
|---|---|---|
| Byte | 8 bits | 0 to 255 |
| Halfword | 16 bits | 0 to 65535 |
| Word | 32 bits | 0 to 4294967295 |

### 3.1.4  Bit string

A bit string is a type of data whose bit length is variable from 0 to $2^{32} - 1$.  To specify a bit-string data, define the following three attributes.

- A : address of the string data's first word (lower two bits must be 0.)
- B : in-word bit offset in the string data (0 to 31)
- M : bit length of the string data (0 to $2^{32} - 1$)

The above three attributes may vary depending on the bit-string data manipulation direction: upward or downward, as shown below. The former is the direction from lower addresses to higher addresses while the latter is the direction from higher to lower addresses.
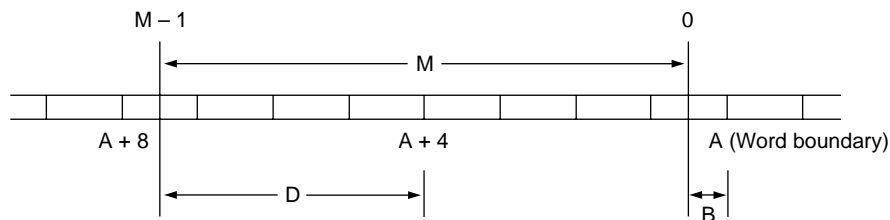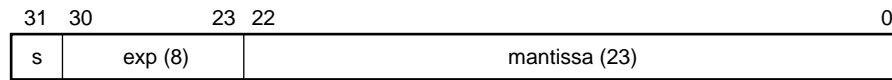


| Attribute | Upward | Downward |
|---|---|---|
| First-word address (0s in bits 1 and 0) | A | A + 4 |
| In-word bit offset (0 to 31) | B | D |
| Bit length (0 to $2^{32} - 1$) | M | M |

**3.1.5  Single-precision floating-point data**

This data type is 32 bits long and its bit allocation complies with the IEEE single format.  A single-precision floating-point data consists of 1-bit mantissa sign bit, 8-bit exponent, and 23-bit mantissa.  The exponent is offset-expressed from the bias value – 127, and the mantissa is binary-expressed with the integer part omitted.

| 31 | 30        23 | 22                                          0 |
|----|--------------|-----------------------------------------------|
| s  | exp (8)      | mantissa (23)                                 |

**3.2  Data Alignment**

In the V810, a word data must be aligned to a word boundary (with the lowest two bits of the address fixed to 0s), and a halfword data to a halfword boundary (with the lowest bit of the address fixed to 0).  If a data is not aligned as specified, the lowest one bit (in the case of word) or two bits (in the case of halfword) of its address will forcibly be masked with 0s when the data is accessed.

★ **4. ADDRESS SPACE**

The V810 supports 4 Gbytes of linear memory space and I/O space.  The CPU outputs 32-bit addresses to the memory and I/Os; therefore, the addresses are from 0 to $2^{32} - 1$.

Bit number 0 of each byte data is defined as the LSB (Least Significant Bit), and bit number 7 is the MSB (Most Significant Bit).  Unless otherwise specified, the byte data at the lower address side of data consisting of two or more bytes is the LSB, and the byte data at the higher address side is the MSB (little endian).

Data consisting of 2 bytes is called a halfword, and data consisting of 4 bytes is called a word.  The lower address of memory or I/O data of two or more bytes, here, is shown on the right, and the higher address is shown on the left, as follows:
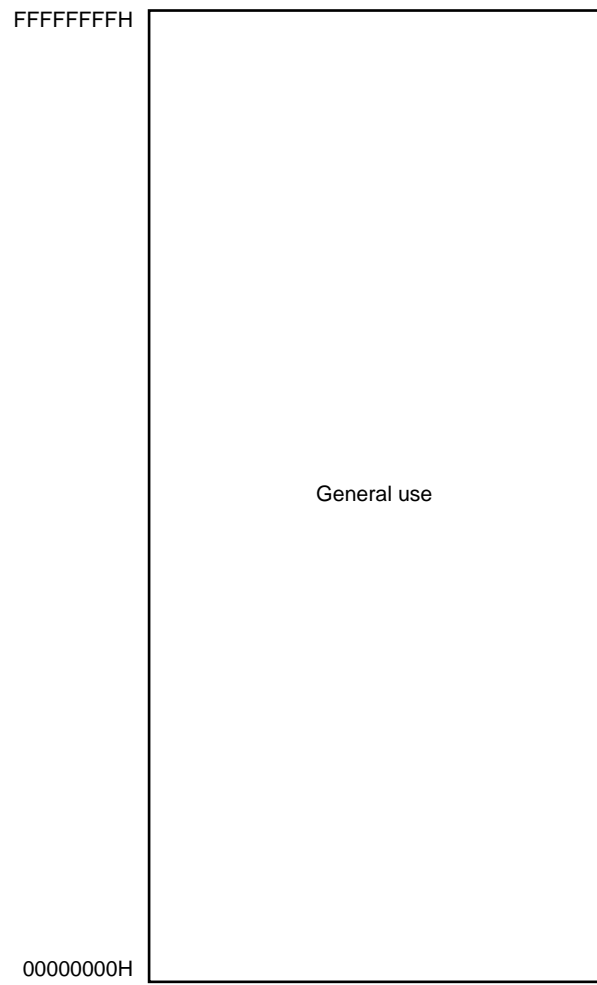
Byte of address A ·············································································· 
```
 7        0
┌─────────┐
│         │
└─────────┘
A (address)
```

Halfword of address A ·············································· 
```
15      8 7        0
┌────────┬─────────┐
│        │         │
└────────┴─────────┘
  A + 1    A (address)
```

Word/short real of address A ·················· 
```
31     24 23     16 15      8 7        0
┌────────┬────────┬─────────┬─────────┐
│        │        │         │         │
└────────┴────────┴─────────┴─────────┘
  A + 3     A + 2     A + 1      A
```
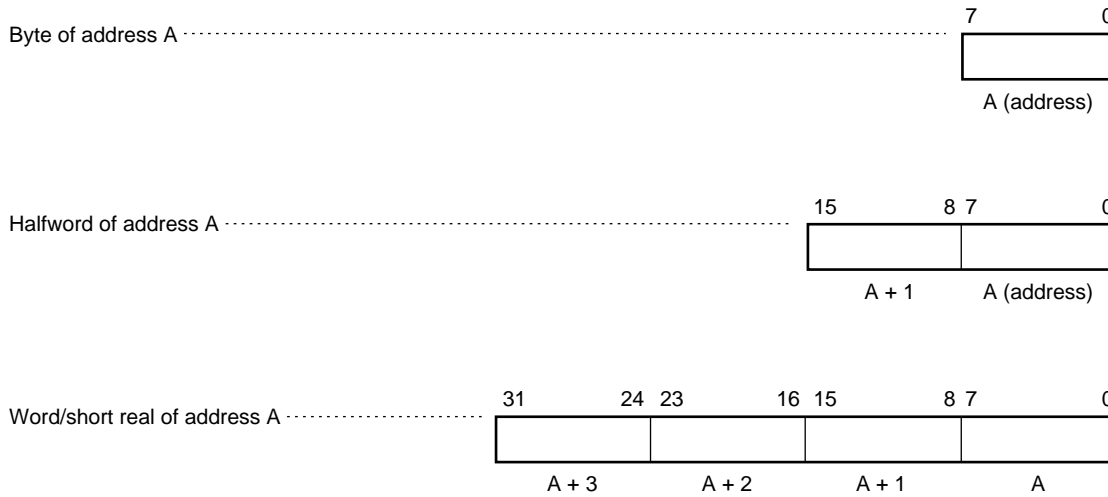
Figure 4-1 shows the memory map of the V810, and Figure 4-2 shows the I/O map.

**Figure 4-1.  Memory Map**



FFFFFFFFH

Interrupt handler table**Note**

FFFFFE00H
FFFFFDFFH

General use

00000000H
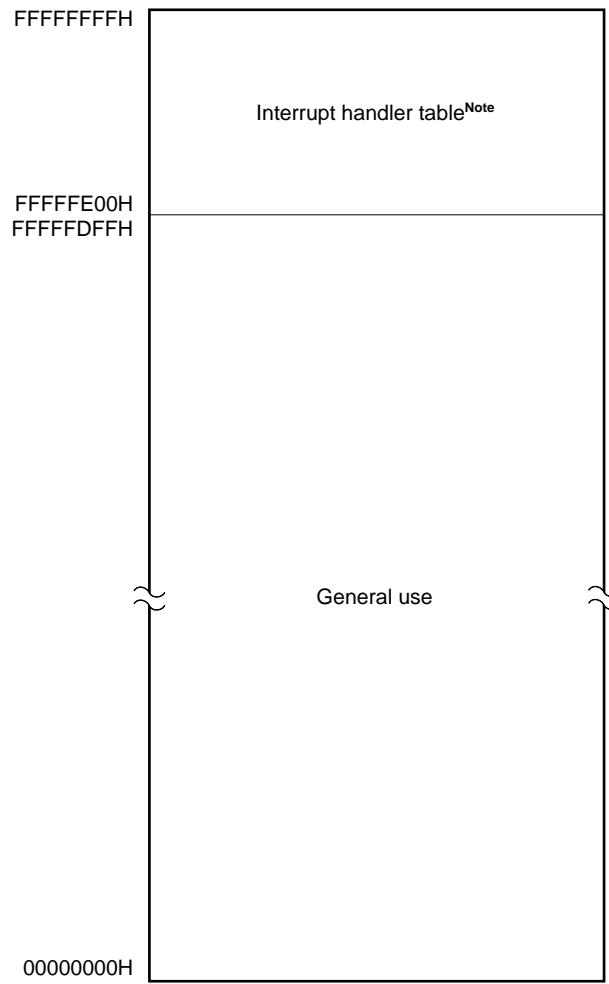
**Note**    For the details, refer to **Table 6-1 Exception Codes**.

**Figure 4-2.  I/O Map**

FFFFFFFFH

General use

00000000H

## 5. BUS INTERFACE FUNCTION                                                ★

The V810 is equipped with a 32-bit data bus.

In the bus interface, there are two modes: 32-bit bus mode which uses the data bus in 32 bits and 16-bit bus fixed mode which fixes the bus in 16 bits. Modes can be switched only at reset using the SIZ16B signal.

The 32-bit bus mode has a dynamic bus sizing function which uses the data bus in 16-bit bus width to access the 16-bit peripherals. This function can be used by setting the $\overline{SZRQ}$ signal active. Access to word data (32-bit data) in the dynamic bus sizing is executed by loading/storing a 16-bit data twice.

In the 16-bit bus fixed mode, access to word data (32-bit data) is executed by activating a bus cycle twice. The control signal and the A1 signal output values according to the 16-bit system.

The relationship between the external access and byte enable signals ($\overline{BE3}$ to $\overline{BE0}$) during the 32-bit bus mode and the 16-bit bus fixed mode is shown below.

**Table 5-1.  Relationship among Address, Data Length, Byte Enable Signals and A1 (32-bit bus mode)**

| Data length | Operand address | | Byte enable | | | | A1 | Bus cycle sequence |
|---|---|---|---|---|---|---|---|---|
| | Bit 1 | Bit 0 | $\overline{BE3}$ | $\overline{BE2}$ | $\overline{BE1}$ | $\overline{BE0}$ | | |
| Byte | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| Halfword | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| Word | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | | 0 | 0 | 1 | 1 | 1 | 2[Note] |

**Note**    Bus cycle added by dynamic bus sizing

**Table 5-2.  Relationship among Address, Data Length, Byte Enable Signals and A1 (16-bit bus fixed mode)**

| Data length | Operand address | | Byte enable | | | | A1 | Bus cycle sequence |
|---|---|---|---|---|---|---|---|---|
| | Bit 1 | Bit 0 | $\overline{BE3}$ | $\overline{BE2}$ | $\overline{BE1}$ | $\overline{BE0}$ | | |
| Byte | 0 | 0 | Hi-Z | Hi-Z | 1 | 0 | 0 | 1 |
| | 0 | 1 | Hi-Z | Hi-Z | 0 | 1 | 0 | 1 |
| | 1 | 0 | Hi-Z | Hi-Z | 1 | 0 | 1 | 1 |
| | 1 | 1 | Hi-Z | Hi-Z | 0 | 1 | 1 | 1 |
| Halfword | 0 | 0 | Hi-Z | Hi-Z | 0 | 0 | 0 | 1 |
| | 1 | 0 | Hi-Z | Hi-Z | 0 | 0 | 1 | 1 |
| Word | 0 | 0 | Hi-Z | Hi-Z | 0 | 0 | 0 | 1 |
| | | | Hi-Z | Hi-Z | 0 | 0 | 1 | 2[Note] |

**Note**    Added bus cycle

★  **6.   INTERRUPT AND EXCEPTION**

Interrupts are events that take place independently of the program execution and can be classified into maskable interrupts and a non-maskable interrupt. An exception is an event that takes place depending upon the program execution. There is little difference between the interrupt and exception in terms of flow, but the interrupt takes precedence over the exception.

The V810 architecture is provided with the interrupts and exceptions listed in the table below. If an exception, a maskable interrupt or NMI occurs, control is transferred to a handler whose address is determined by the source of the interrupt or exception. The exception source can be checked by examining an exception code stored in the ECR (Exception Code Register). Each handler analyzes the contents of the ECR and performs appropriate exception/interrupt servicing.

**Table 6-1.  Exception Codes**

| Exception and interrupt | Classification | Exception code | Handler address | Restore PC[Note 1] |
|---|---|---|---|---|
| Reset | Interrupt | F F F 0 | F F F F F F F 0 | **Note 2** |
| NMI | Interrupt | F F D 0 | F F F F F F D 0 | next PC[Note 3] |
| Duplexed exception | Exception | **Note 4** | F F F F F F D 0 | current PC |
| Address trap | Exception | F F C 0 | F F F F F F C 0 | current PC |
| Trap instruction (parameter is 0x1n) | Exception | F F B n | F F F F F F B 0 | next PC |
| Trap instruction (parameter is 0x0n) | Exception | F F A n | F F F F F F A 0 | next PC |
| Invalid instruction code | Exception | F F 9 0 | F F F F F F 9 0 | current PC |
| Zero division | Exception | F F 8 0 | F F F F F F 8 0 | current PC |
| FIV (floating-point invalid operation) | Exception | F F 7 0 | F F F F F F 6 0 | current PC |
| FZD (floating-point zero division) | Exception | F F 6 8 | F F F F F F 6 0 | current PC |
| FOV (floating-point overflow) | Exception | F F 6 4 | F F F F F F 6 0 | current PC |
| FUD (floating-point underflow)[Note 5] | Exception | F F 6 2 | F F F F F F 6 0 | current PC |
| FPR (floating-point precision degradation)[Note 5] | Exception | F F 6 1 | F F F F F F 6 0 | current PC |
| FRO (floating-point reserved operand) | Exception | F F 6 0 | F F F F F F 6 0 | current PC |
| INT level n (n = 0 to 15) | Interrupt | F E n 0 | F F F F F E n 0 | next PC[Note 3] |

**Notes 1.** PC to be saved to EIPC or FEPC.
   **2.** EIPC and FEPC are undefined.
   **3.** While an instruction whose execution is aborted by an interrupt (DIV/DIVU, single-precision floating-point data, bit string instruction) is executed, restore PC = current PC.
   **4.** The exception code of the exception that occurs for the first time is stored to the lower 16 bits of the ECR, and that of the second exception is stored in the higher 16 bits.
   **5.** In the V810, the floating-point underflow exception and floating-point precision degradation exception do not occur.
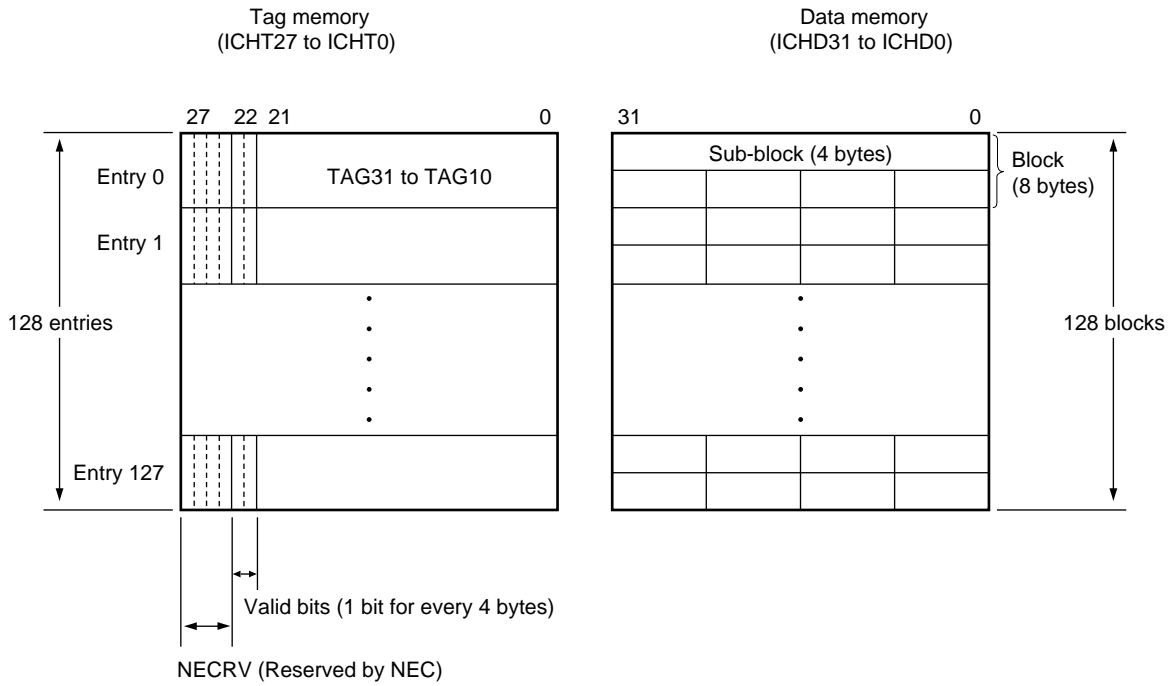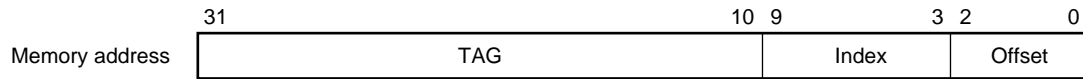
## 7. CACHE                                                                              ★

Figure 7-1 shows the instruction cache configuration provided to the V810.

**Figure 7-1.  Cache Configuration**

| | |
|---|---|
| Capacity | : 1 Kbytes |
| Mapping system | : direct map |
| Block size | : 8 bytes |
| Sub-block size | : 4 bytes |

Memory address

| 31 | 10 | 9 | 3 | 2 | 0 |
|---|---|---|---|---|---|
| TAG | | Index | | Offset | |

Tag memory
(ICHT27 to ICHT0)

Data memory
(ICHD31 to ICHD0)

Entry 0
Entry 1
128 entries
Entry 127

27  22  21                          0

TAG31 to TAG10

31                                  0

Sub-block (4 bytes)

Block
(8 bytes)

128 blocks

Valid bits (1 bit for every 4 bytes)

NECRV (Reserved by NEC)

★ **8. RESET**

A low-level input detection on the $\overline{\text{RESET}}$ pin always triggers a system reset. Consequently, all the hardware-controlling registers are initialized as shown in Table 8-1. After the initialization procedure is completed and the $\overline{\text{RESET}}$ pin returns to the high level, the device is released from the resetting state and starts the implementation of a program. Then, if necessary, set some registers to user-desired values in the first stage of the program.

**Table 8-1. Register State after Reset**

| Hardware (Symbol) | | State after Reset |
|---|---|---|
| Program counter | PC | FFFFFFF0H |
| Status saving register for interrupt | EIPC | Undefind |
| | EIPSW | |
| Status saving register for NMI | FEPC | Undefind |
| | FEPSW | |
| Interrupt cause register | FECC | 0000H |
| | EICC | FFF0H |
| Program status word | PSW | 00008000H |
| General-purpose register | r0 | Fixed to 00000000H |
| | r1 to r31 | Undefind |

## 9. INSTRUCTION SET ★

### 9.1 Instruction Format

The V810 instructions are formatted in either 16 bits or 32 bits.  Examples of the 16-bit format instruction are binomial operation, control, and conditional branch; those for the 32-bit format are load/store, I/O manipulate, 16-bit immediate, jump & link, and extended operations.

Some instructions have an unused field.  However, do not write a program that uses this field because it is reserved for future use.  This unused field must be set to zeros.

Instructions are stored in memory in the following manner.

- The lower half of an instruction, that is, the half which includes bit 0, is stored at the lower address.
- The higher half of an instruction, that is, the half which includes bit 15 or 31, is stored at the higher address.

**(1)  reg-reg instruction format (Format I)**

This format consists of one 6-bit field to hold an operation code and two 5-bit fields to specify general-purpose registers as instruction's operands.  16-bit instructions use this format.

```
15          10 9        5 4        0
+-------------+----------+----------+
|   opcode    |   reg2   |   reg1   |
+-------------+----------+----------+
```
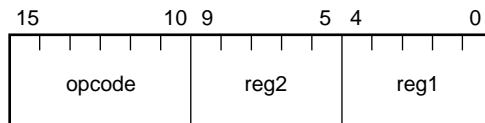
**(2)  imm-reg instruction format (Format II)**

This format consists of one 6-bit field to hold an operation code, one 5-bit field to hold an immediate data, and one field to specify a general-purpose register as an operand.  16-bit instructions use this format.

```
15          10 9        5 4        0
+-------------+----------+----------+
|   opcode    |   reg2   |   imm    |
+-------------+----------+----------+
```
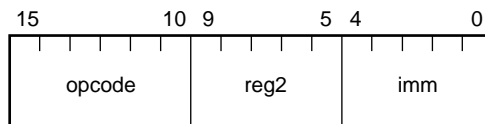
**(3)  Conditonal branch instruction format (Format III)**

This format consists of one 3-bit field to hold an operation code, one 4-bit field to hold a condition code, and one 9-bit field to hold a branch displacement (with its LSB masked to 0).  16-bit instructions use this format.

```
15   13 12      9 8              0
+------+---------+----------------+
|opcode|  cond   |   disp       0 |
+------+---------+----------------+
```
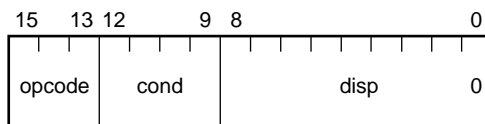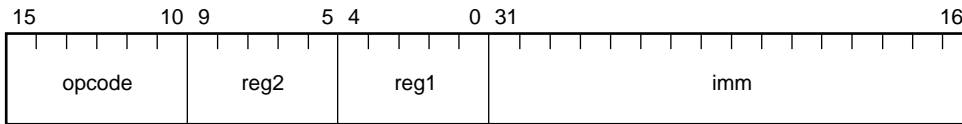
**(4) Intermediate jump instruction format (Format IV)**

This format consists of one 6-bit field to hold an operation code and one 26-bit field to hold a displacement (with its LSB masked to 0).  32-bit instructions use this format.
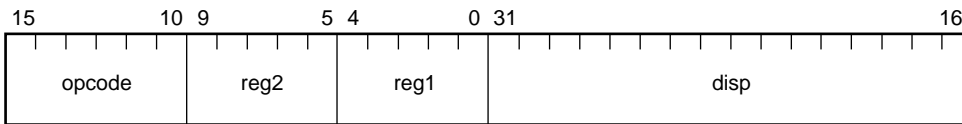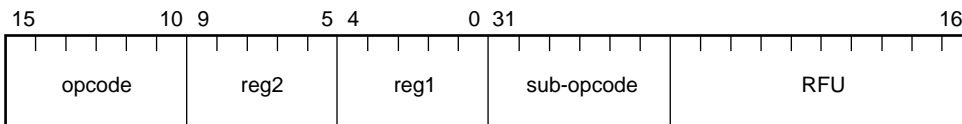
```
15        10 9              0 31                              16
 ┌─────────┬────────────────────┬──────────────────────────────┐
 │ opcode  │               disp                              │ 0 │
 └─────────┴────────────────────┴──────────────────────────────┘
```
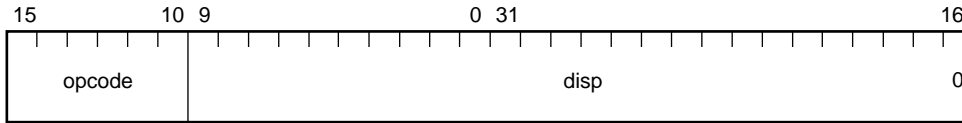
**(5) 3-operand instruction format (Format V)**

This format consists of one 6-bit field to hold an operation code, two fields to specify general-purpose registers as operands, and one 16-bit field to hold an immediate data.  32-bit instructions use this format.

```
15        10 9    5 4      0 31                              16
 ┌─────────┬───────┬────────┬──────────────────────────────────┐
 │ opcode  │ reg2  │ reg1   │              imm                  │
 └─────────┴───────┴────────┴──────────────────────────────────┘
```

**(6) Load/store instruction format (Format VI)**

This format consists of one 6-bit field to hold an operation code, two fields to specify a general-purpose register, and one 16-bit field to hold a displacement.  32-bit instructions use this format.

```
15        10 9    5 4      0 31                              16
 ┌─────────┬───────┬────────┬──────────────────────────────────┐
 │ opcode  │ reg2  │ reg1   │              disp                 │
 └─────────┴───────┴────────┴──────────────────────────────────┘
```

**(7) Extension instruction format (Format VII)**

This format consists of one 6-bit field to hold an operation code, two 5-bit fields to specify general-purpose registers as operands, and one 6-bit field to hold an sub-operation code.  The remaining 10 bits are reserved for future use and must be set to zeros.  32-bit instructions use this format.

```
15        10 9    5 4      0 31                              16
 ┌─────────┬───────┬────────┬──────────────┬────────────────────┐
 │ opcode  │ reg2  │ reg1   │  sub-opcode  │        RFU         │
 └─────────┴───────┴────────┴──────────────┴────────────────────┘
```

### 9.2 Instruction Mnemonic (in alphabetical order)

The list of mnemonics is shown below.

This section lists the instructions incorporated in the V810 along with their operations.  The instructions are listed in the instruction mnemonic's alphabetical order to allow users to use this section as a quick reference or dictionary. The conventions used in the list are shown below.

| Instruction Mnemonic | Operand (s) | Format | CY OV S Z | Instruction Function |
|---|---|---|---|---|

**Legend**

| ADD | reg1, reg2 | I | * * * * |
|---|---|---|---|

| Mnemonic of instruction | Identifier of operand | Instruction format (Refer to **9.1**.) | Flag operation |
|---|---|---|---|

Flag operation:

– Remains unchanged
* Inverts the previous value
0 Changes to 0
1 Changes to 1
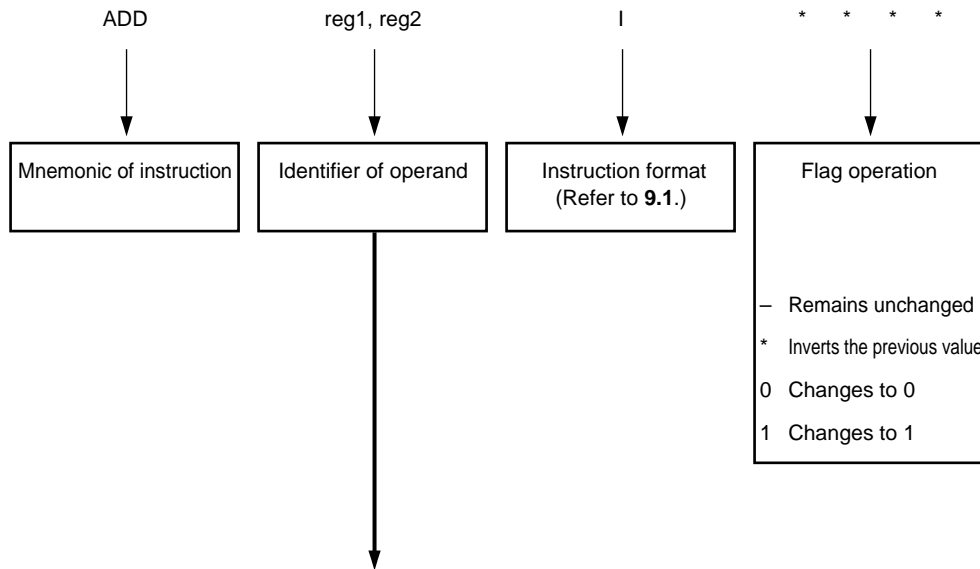
| Identifier | Description |
|---|---|
| reg1 | General-purpose register (Used as a source register) |
| reg2 | General-purpose register (Used mainly as a destination register and occasionally as a source register) |
| imm5 | 5-bit immediate |
| imm16 | 16-bit immediate |
| disp9 | 9-bit displacement |
| disp16 | 16-bit displacement |
| disp26 | 26-bit displacement |
| regID | System register number |
| vector adr | Trap handler address that corresponds to a trap vector |

**Table 9-1.  Instruction Mnemonics (in alphabetical order) (1/9)**

| Instruction Mnemonic | Operand (s) | Format | CY | OV | S | Z | Instruction Function |
|---|---|---|---|---|---|---|---|
| ADD | reg1, reg2 | I | * | * | * | * | Addition:<br>Adds the word data in the reg2-specified register and the word data in the reg1-specified register, then stores the result into the reg2-specified register. |
| ADD | imm5, reg2 | II | * | * | * | * | Addition:<br>Sign-extends the 5-bit immediate data to 32 bits, and adds the extended immediate data and the word data in the reg2-specified register, then stores the result into the reg2-specified register. |
| ADDF.S | reg1, reg2 | VII | * | 0 | * | * | Floating-point addition:<br>Adds the single-precision floating-point data in the reg2-specified register and the single-precision floating-point data in the reg1-specified register, then restores the result into the reg2-specified register while changing flags according to the result. |
| ADDI | imm16, reg1, reg2 | V | * | * | * | * | Addition:<br>Sign-extends the 16-bit immediate data to 32 bits, and adds the extended immediate data and the word data in the reg1-specified register, then stores the result into the reg2-specified register. |
| AND | reg1, reg2 | I | – | 0 | * | * | AND:<br>Performs the logical AND operation on the word data in the reg2-specified register and the word data in the reg1-specified register, then stores the result into the reg2-specified register. |
| ANDBSU | – | II | – | – | – | – | Transfer after ANDing bit strings:<br>Performs a logical AND operation on a source bit string and a destination bit string, then transfers the result to the destination bit string. |
| ANDI | imm16, reg1, reg2 | V | – | 0 | 0 | * | AND:<br>Sign-extends the 16-bit immediate data to 32 bits, and performs a logical AND operation on the extended immediate data and the word data in the reg1-specified register, then stores the result into the reg2-specified register. |
| ANDNBSU | – | II | – | – | – | – | Transfer after NOTting a bit string then ANDing it with another bit string:<br>Performs a logical AND operation on a destination bit string and the 1's complement of a source bit string, then transfers the result to the destination bit string. |
| BC | disp9 | III | – | – | – | – | Conditional branch (if Carry):<br>PC relative branch |
| BE | disp9 | III | – | – | – | – | Conditional branch (if Equal):<br>PC relative branch |
| BGE | disp9 | III | – | – | – | – | Conditional branch (if Greater than or Equal):<br>PC relative branch |
| BGT | disp9 | III | – | – | – | – | Conditional branch (if Greater than):<br>PC relative branch |

**Table 9-1.  Instruction Mnemonics (in alphabetical order) (2/9)**

| Instruction Mnemonic | Operand (s) | Format | CY | OV | S | Z | Instruction Function |
|---|---|---|---|---|---|---|---|
| BH | disp9 | III | – | – | – | – | Conditional branch (if Higher): PC relative branch |
| BL | disp9 | III | – | – | – | – | Conditional branch (if Lower): PC relative branch |
| BLE | disp9 | III | – | – | – | – | Conditional branch (if Less than or Equal): PC relative branch |
| BLT | disp9 | III | – | – | – | – | Conditional branch (if Less than): PC relative branch |
| BN | disp9 | III | – | – | – | – | Conditional branch (if Negative): PC relative branch |
| BNC | disp9 | III | – | – | – | – | Conditional branch (if Not Carry): PC relative branch |
| BNE | disp9 | III | – | – | – | – | Conditional branch (if Not Equal): PC relative branch |
| BNH | disp9 | III | – | – | – | – | Conditional branch (if Not Higher): PC relative branch |
| BNL | disp9 | III | – | – | – | – | Conditional branch (if Not Lower): PC relative branch |
| BNV | disp9 | III | – | – | – | – | Conditional branch (if Not Overflow): PC relative branch |
| BNZ | disp9 | III | – | – | – | – | Conditional branch (if Not Zero): PC relative branch |
| BP | disp9 | III | – | – | – | – | Conditional branch (if Positive): PC relative branch |
| BR | disp9 | III | – | – | – | – | Unconditional branch: PC relative branch |
| BV | disp9 | III | – | – | – | – | Conditional branch (if Overflow): PC relative branch |
| BZ | disp9 | III | – | – | – | – | Conditional branch (if Zero): PC relative branch |
| CAXI | disp16 [reg1], reg2 | VI | * | * | * | * | Inter-processor synchronization in a multi-processor system. |
| CMP | reg1, reg2 | I | * | * | * | * | Comparison: Subtracts the word data in the reg1-specified register from that for reg2 for comparison, then changes flags according to the result. |
| CMP | imm5, reg2 | II | * | * | * | * | Comparison: Sign-extends the 5-bit immediate data to 32 bits, and subtracts the extended immediate data from the word data in the reg2-specified register for comparison, then changes flags according to the result. |
| CMPF.S | reg1, reg2 | VII | * | 0 | * | * | Floating-point comparison: Subtracts the single-precision floating-point data in the reg1-specified register from that for reg2 for comparison, then changes flags according to the result. |

**Table 9-1.  Instruction Mnemonics (in alphabetical order) (3/9)**

| Instruction Mnemonic | Operand (s) | Format | CY | OV | S | Z | Instruction Function |
|---|---|---|---|---|---|---|---|
| CVT.SW | reg1, reg2 | VII | – | 0 | * | * | Data conversion from floating-point to integer: Converts the single-precision floating-point data in the reg1-specified register into an integer data, then stores the result into the reg2-specified register while changing flags according to the result. |
| CVT.WS | reg1, reg2 | VII | * | 0 | * | * | Data conversion from integer to floating-point: Converts the integer data in the reg1-specified register into a single-precision floating-point data, then stores the result into the reg2-specified register while changing flags according to the result. |
| DIV | reg1, reg2 | I | – | * | * | * | Signed division: Divides the word data in the reg2-specified register by that for reg1 with their sign bits validated, then stores the quotient into the reg2-specified register and the remainder into r30.  Division is performed so that the sign of the remainder matches that of the dividend. |
| DIVF.S | reg1, reg2 | VII | * | 0 | * | * | Floating-point division: Divides the single-precision floating-point data in the reg2-specified register by that for reg1, then stores the result into the reg2-specified register while changing flags according to the result. |
| DIVU | reg1, reg2 | I | – | 0 | * | * | Unsigned division: Divides the word data in the reg2-specified register by that for reg1 with their data handled as unsigned data, then stores the quotient into the reg2-specified register and the remainder into r30.  Division is performed so that the sign of the remainder matches that of the dividend. |
| HALT | – | II | – | – | – | – | Processor stop |
| IN.B | disp16 [reg1], reg2 | VI | – | – | – | – | Port input: Sign-extends the 16-bit displacement to 32 bits, and adds the extended displacement and the content of the reg1-specified register to generate a 32-bit unsigned port address, then reads the byte data located at the generated port address, zero-extends the byte data to 32 bits, and stores the result into the reg2-specified register. |
| IN.H | disp16 [reg1], reg2 | VI | – | – | – | – | Port input: Sign-extends the 16-bit displacement to 32 bits, and adds the extended displacement and the content of the reg1-specified register to generate a 32-bit unsigned port address, then reads the halfword data located at the generated port address while masking the address's bit 0 to 0, zero-extends the halfword data to 32 bits, and stores the result into the reg2-specified register. |

**Table 9-1. Instruction Mnemonics (in alphabetical order) (5/9)**

| Instruction Mnemonic | Operand (s) | Format | CY | OV | S | Z | Instruction Function |
|---|---|---|---|---|---|---|---|
| LDSR | reg2, regID | II | * | * | * | * | Loading system register:<br>Transfers the word data in the reg2-specified register to the system register specified with the system register number (regID). |
| MOV | reg1, reg2 | I | – | – | – | – | Transferring data:<br>Loads the reg2-specified register with the word data in of the reg1-specified register. |
| MOV | imm5, reg2 | II | – | – | – | – | Transferring data:<br>Sign-extends the 5-bit immediate data to 32 bits, then loads the reg2-specified register with the extended immediate data. |
| MOVBSU | – | II | – | – | – | – | Transferring bit strings:<br>Loads the destination bit string with the source bit string. |
| MOVEA | imm16, reg1, reg2 | V | – | – | – | – | Addition:<br>Sign-extends the 16-bit immediate data to 32 bits, adds it with the word data in the reg1-specified register, then stores the addition result into reg2. |
| MOVHI | imm16, reg1, reg2 | V | – | – | – | – | Addition:<br>Appends 16-bit zeros below the 16-bit immediate data to form a 32-bit word data, then adds it with the word data in the reg1-specified register, and stores the result into the reg2-specified register. |
| MUL | reg1, reg2 | I | – | * | * | * | Signed multiplication:<br>Signed-multiplies the word data in the reg2-specified register by that for reg1, then separates the 64-bit (double-word) result into two 32-bit data, and stores the higher 32 bits into r30 and the lower 32 bits into the reg2-specified register. |
| MULF.S | reg1, reg2 | VII | * | 0 | * | * | Floating-point multiplication:<br>Multiplies the single-precision floating-point data in the reg2-specified register by that for reg1, then stores the result into the reg2-specified register while changing flags according to the result. |
| MULU | reg1, reg2 | I | – | * | * | * | Unsigned multiplication:<br>Multiplies the word data in the reg2-specified register by that for reg1 while handling these data as unsigned data, then separates the 64-bit (double-word) result into two 32-bit data, and stores the higher 32 bits into r30 and the lower 32 bits into the reg2-specified register. |
| NOP | – | III | – | – | – | – | No operation:<br>Makes no changes or operations while spending one instruction cycle. |
| NOT | reg1, reg2 | I | – | 0 | * | * | Logical NOT:<br>Obtains the 1's complement (logical NOT) of the content of the reg1-specified register, then stores the result into the reg2-specified register. |

**Table 9-1.  Instruction Mnemonics (in alphabetical order) (6/9)**

| Instruction Mnemonic | Operand (s) | Format | CY | OV | S | Z | Instruction Function |
|---|---|---|---|---|---|---|---|
| NOTBSU | – | II | – | – | – | – | Transfer after NOTting a bit string: Obtains the 1's complement (all bits inverted) of the source bit string, then transfers the result to the destination bit string. |
| OR | reg1, reg2 | I | – | 0 | * | * | OR: Performs a logical OR operation on the word data in the reg2-specified register and that for reg1, then stores the result into the reg2-specified register. |
| ORBSU | – | II | – | – | – | – | Transfer after ORing bit strings: Performs a logical OR operation on the source and destination bit strings, then transfers the result to the destination bit string. |
| ORI | imm16, reg1, reg2 | V | – | 0 | * | * | OR: Zero-extends the 16-bit immediate data to 32 bits, performs a logical OR operation on the extended data and the word data in the reg1-specified register, then stores the result into the reg2-specified register. |
| ORNBSU | – | II | – | – | – | – | Transfer after NOTting a bit string and ORing it with another bit string: Obtains the 1's complement (logical NOT) of the source bit string, performs a logical OR operation on the NOTted bit string and the destination bit string, then transfers the result to the destination bit string. |
| OUT.B | reg2, disp16 [reg1] | VI | – | – | – | – | Port output: Sign-extends the 16-bit displacement to 32 bits, adds the extended value and the content of the reg1-specified register to generate a 32-bit unsigned port address, then outputs the lowest 8 bits (= 1 byte) of the reg2-specified register onto the port pins corresponding to the generated port address. |
| OUT.H | reg2, disp16 [reg1] | VI | – | – | – | – | Port output: Sign-extends the 16-bit displacement to 32 bits, adds the extended value and the content of the reg1-specified register to generate a 32-bit unsigned port address with its bit 0 masked to 0, then outputs the lowest 16 bits (= 1 halfword) of the reg2-specified register onto the port pins corresponding to the generated port address. |
| OUT.W | reg2, disp16 [reg1] | VI | – | – | – | – | Port output: Sign-extends the 16-bit displacement to 32 bits, adds the extended value and the content of the reg1-specified register to generate a 32-bit unsigned port address with its bits 0 and 1 masked to 0, then outputs the 32 bits (= 1 word) of the reg2-specified register onto the port pins corresponding to the generated port address. |

**Table 9-1. Instruction Mnemonics (in alphabetical order) (7/9)**

| Instruction Mnemonic | Operand (s) | Format | CY | OV | S | Z | Instruction Function |
|---|---|---|---|---|---|---|---|
| RETI | – | II | * | * | * | * | Return from a trap or interrupt routine: Reads the restore PC and PSW from the system registers and loads them to the due places to return from a trap or interrupt routine to the original operation flow. |
| SAR | reg1, reg2 | I | * | 0 | * | * | Arithmetic right shift: Shifts every bit of the word data in the reg2-specified register to the right by the number of times specified with the reg1-specified register's lowest 5 bits, then stores the result into the reg2-specified register. In arithmetic right shift operations, the MSB is loaded with the LSB value at each shift. |
| SAR | imm5, reg2 | II | * | 0 | * | * | Arithmetic right shift: Zero-extends the 5-bit immediate data to 32 bits, shifts every bit of the word data in the reg2-specified register to the right by the number of times specified with the extended immediate data, then stores the result into the reg2-specified register. In arithmetic right shift operations, the MSB is loaded with the LSB value at each shift. |
| SCH0BSU SCH0BSD | – – | II II | – – | – – | – – | * * | Searching 0s in a bit string: Searches "0" bits in the source bit string, and loads r30 and r27 with the address of the bit next to the first detected "0" bit, then r29 with the number of bits skipped until the first "0" bit is detected, and r28 with the value subtracted by the r29 value. |
| SCH1BSU SCH1BSD | – – | II II | – – | – – | – – | – – | Searching 1s in a bit string: Searches 1s in the source bit string, and loads r30 and r27 with the bit address next to the first detected "1" bit, then r29 with the number of bits skipped until the first "1" is detected, and r28 with the value subtracted by the r29 value. |
| SETF | imm5, reg2 | II | – | – | – | – | Flag condition setting: Sets the reg2-specified register to 1 if the condition flag value matches the lowest 4 bits of the 5-bit immediate data, and sets the reg2-specified register to 0 when they do not match. |
| SHL | reg1, reg2 | I | * | 0 | * | * | Logical left shift: Shifts every bit of the word data in the reg2-specified register to the left by the number of times specified with the reg1-specified register's lowest 5 bits, then stores the result into the reg2-specified register. In logical left shift operations, the LSB is loaded with 0 at each shift. |

**Table 9-1.  Instruction Mnemonics (in alphabetical order) (8/9)**

| Instruction Mnemonic | Operand (s) | Format | CY | OV | S | Z | Instruction Function |
|---|---|---|---|---|---|---|---|
| SHL | imm5, reg2 | II | * | 0 | * | * | Logical left shift:<br>Zero-extends the 5-bit immediate data to 32 bits, shifts every bit of the word data in the reg2-specified register to the left by the number of times specified by the extended immediate data, then stores the result into the reg2-specified register. In logical left shift operations, the LSB is loaded with 0 at each shift. |
| SHR | reg1, reg2 | I | * | 0 | * | * | Logical right shift:<br>Shifts every bit of the word data in the reg2-specified register to the right by the number of times specified with the reg1-specified register's lowest 5 bits, then stores the result into the reg2-specified register. In logical right shift operations, the MSB is loaded with 0 at each shift. |
| SHR | imm5, reg2 | II | * | 0 | * | * | Logical right shift:<br>Zero-extends the 5-bit immediate data to 32 bits, shifts every bit of the word data in the reg2-specified register to the right by the number of times specified by the extended immediate data, then stores the result into the reg2-specified register. In logical right shift operations, the MSB is loaded with 0 at each shift. |
| ST.B | reg2, disp16 [reg1] | VI | – | – | – | – | Byte store:<br>Sign-extends the 16-bit displacement to 32 bits and adds the 32-bit displacement and the content of the reg1-specified register to generate a 32-bit unsigned address, then transfers the reg2-specified register's lowest 8 bits to the generated address. |
| ST.H | reg2, disp16 [reg1] | VI | – | – | – | – | Halfword store:<br>Sign-extends the 16-bit displacement to 32 bits with its bit 0 masked to 0, and adds the content of the reg1-specified register and the 32-bit displacement to generate a 32-bit unsigned address, then transfers the reg2-specified register's lower 16 bits to the generated address. |
| ST.W | reg2, disp16 [reg1] | VI | – | – | – | – | Word store:<br>Sign-extends the 16-bit displacement to 32 bits with its bits 0 and 1 masked to 0, and adds the reg1-specified register and the 32-bit displacement to generate a 32-bit unsigned address, then transfers the content of the reg1-specified register to the generated address. |
| STSR | regID, reg2 | II | – | – | – | – | Storing system register contents:<br>Loads the reg2-specified register with the content of the system register specified by the system register number (regID). |
| SUB | reg1, reg2 | I | * | * | * | * | Subtraction:<br>Subtracts the content of the reg1-specified register from the content of the reg2-specified register, then stores the result into the reg2-specified register. |

**Table 9-1.  Instruction Mnemonics (in alphabetical order) (9/9)**

| Instruction Mnemonic | Operand (s) | Format | CY | OV | S | Z | Instruction Function |
|---|---|---|---|---|---|---|---|
| SUBF.S | reg1, reg2 | VII | * | 0 | * | * | Floating-point subtraction:<br>Subtracts the single-precision floating-point data in the reg1-specified register from that for reg2, then stores the result into the reg2-specified register while changing flags according to the result. |
| TRAP | vector | II | – | – | – | – | Software trap:<br>Jumps to a trap handler address according to the vector-specified trap vector (from 0 to 31) to start an exception handling after completing all necessary saving and presetting procedures as follows:<br>(1) Saving the restore PC and PSW into the FEPC and FEPSW system registers, respectively, if the PSW's EP flag = 1, or into the EIPC and EIPSW system registers, respectively, if EP = 0<br>(2) Setting an exception code into the ECR's FECC and FESW flags if the PSW's EP flag = 1, or into the ECR's EICC if EP = 0<br>(3) Setting the PSW's ID flag and clearing the PSW's AE flag<br>(4) Setting the PSW's NP flag if the PSW's EP flag = 1, or setting the PSW's ID flag if EP = 0 |
| TRNC.SW | reg1, reg2 | VII | – | 0 | * | * | Conversion from floating-point data to integer:<br>Converts the single-precision floating-point data in the reg1-specified register into an integer data, then stores the result into the reg2-specified register while changing flags according to the result. |
| XOR | reg1, reg2 | I | – | 0 | * | * | Exclusive OR:<br>Performs a logical exclusive-OR operation on the word data in the reg2-specified register and that for reg1, then stores the result into the reg2-specified register. |
| XORBSU | – | II | – | – | – | – | Transfer of exclusive ORed bit string:<br>Performs a logical exclusive-OR operation on the source and destination bit strings, then transfers the result to the destination bit string. |
| XORI | imm16, reg1, reg2 | V | – | 0 | * | * | Exclusive OR:<br>Zero-extends the 16-bit immediate data to 32 bits and performs a logical exclusive-OR operation on the extended immediate data and the word data in the reg2-specified register, then stores the result into the reg2-specified register. |
| XORNBSU | – | II | – | – | – | – | Transfer after exclusive-ORing a NOTted bit string and another bit string:<br>Obtains the 1's complement (NOT) of the source bit string, and exclusive-ORs it with the destination bit string, then transfers the result to the destination bit string. |

## 10. ELECTRICAL SPECIFICATIONS

**Supported Electrical Specifications**

| Operating Supply Voltage | Operating Ambient Temperature ($T_A$) | μPD70732-16 | μPD70732-20 | μPD70732-25 | | |
|---|---|---|---|---|---|---|
| | | 120-pin Plastic QFP | | | 120-pin Plastic TQFP | 176-pin Ceramic PGA |
| $V_{DD}$ = +5 V ± 10% | −10 to +70°C | ◯ (16 MHz) | ◯ (20 MHz) | ◯ (25 MHz) | ◯ (25 MHz) | ◯ (25 MHz) |
| | −40 to +85°C | — | — | ◯ (20 MHz) | ◯ (20 MHz) | — |
| $V_{DD}$ = 2.7 to 3.6 V | −40 to +85°C | — | — | ◯ (16 MHz) | ◯ (16 MHz) | — |
| $V_{DD}$ = 2.2 to 3.6 V | −40 to +85°C | — | — | ◯ (10 MHz) | ◯ (10 MHz) | — |

★

**Remarks 1.** ◯ : with electrical specifications

— : without electrical specifications

**2.** ( ) : maximum operating frequency

### 10.1 Specifications When V$_{DD}$ = +5 V ± 10%

### (1) T$_A$ = −10 to +70°C

**Absolute Maximum Ratings (T$_A$ = 25°C)**

| Parameter | Symbol | Test Conditions | Rating | Unit |
|---|---|---|---|---|
| Supply voltage | V$_{DD}$ | | −0.5 to +7.0 | V |
| Input voltage | V$_I$ | V$_{DD}$ = +5 V ± 10% | −0.5 to V$_{DD}$ + 0.3 | V |
| Clock Input voltage | V$_K$ | V$_{DD}$ = +5 V ± 10% | −0.5 to V$_{DD}$ + 0.3 | V |
| Output voltage | V$_O$ | V$_{DD}$ = +5 V ± 10% | −0.5 to V$_{DD}$ + 0.3 | V |
| Operating ambient temperature | T$_A$ | | −10 to +70 | °C |
| Storage temperature | T$_{stg}$ | | −65 to +150 | °C |

**Cautions   1.   Do not directly interconnect IC product output (or input/output) pins, or directly connect V$_{DD}$ or V$_{CC}$ to GND. However, open-drain pins and open-collector pins can be interconnected. Direct connection is also possible for an external circuit using timing design that avoids output collision with a pin that becomes high-impedance.**

**2.   Product quality may suffer if the absolute maximum rating is exceeded for even a single parameter, or even momentarily.**
**In other words, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore, the product must be used under conditions which ensure that the absolute maximum ratings are not exceeded.  As far as possible, the product should be used in a state in which the rated value is not approached. The ratings and test conditions shown in the DC characteristics and AC characteristics are the normal operation and quality assurance ranges of the product.**

**DC Characteristics (T$_A$ = −10 to +70°C, V$_{DD}$ = +5V ± 10%)**

| Parameter | Symbol | Test Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Clock input voltage, high | V$_{KH}$ | | 4.0 | | V$_{DD}$ + 0.3 | V |
| Clock input voltage, low | V$_{KL}$ | | −0.5 | | +0.6 | V |
| Input voltage, high | V$_{IH}$ | | 2.2 | | V$_{DD}$ + 0.3 | V |
| Input voltage, low | V$_{IL}$ | | −0.5 | | +0.8 | V |
| Output voltage, high | V$_{OH}$ | I$_{OH}$ = −400 µA | 2.4 | | | V |
| Output voltage, low | V$_{OL}$ | I$_{OL}$ = 3.2 mA | | | 0.45 | V |
| Input leak current, high | I$_{LIH}$ | V$_{IN}$ = V$_{DD}$ | | | 10 | µA |
| Input leak current, low | I$_{LIL}$ | V$_{IN}$ = 0 V | | | −10 | µA |
| Output leak current, high | I$_{LOH}$ | V$_O$ = V$_{DD}$ | | | 10 | µA |
| Output leak current, low | I$_{LOL}$ | V$_O$ = 0 V | | | −10 | µA |
| Supply current | I$_{DD}$ | f = 16 MHz | | 64[Note 2] | 160 | mA |
| | | f = 20 MHz | | 80[Note 2] | 200 | |
| | | f = 25 MHz | | 100[Note 2] | 240 | |
| | | Stopping clock[Note 1] | | 5 | | µA |

**Notes   1.   V$_{IL}$ = 0 V, V$_{IH}$ = V$_{DD}$ applied**
**2.   In general benchmark test (Output pins are open.)**

**Remark**   Operating supply current is approximately proportional to operating clock frequency.

**Capacitance (T$_A$ = 25°C, V$_{DD}$ = +5 V ± 10%)**

| Parameter | Symbol | Test Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| Input capacitance | C$_I$ | f$_C$ = 1 MHz | | 15 | pF |
| I/O capacitance | C$_{IO}$ | | | 15 | pF |

**AC Characteristics (T$_A$ = –10 to +70°C, V$_{DD}$ = +5V ± 10%)**

**Clock Input**

| Parameter | Symbol | Test Conditions | μPD70732-16 | | μPD70732-20 | | μPD70732-25 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| Clock cycle | t$_{CYK}$ | | 62.5 | | 50 | | 40 | | ns |
| Clock pulse high-level width | t$_{KKH}$ | | 26 | | 21 | | 17 | | ns |
| Clock pulse low-level width | t$_{KKL}$ | | 26 | | 21 | | 17 | | ns |
| Clock rise time | t$_{KR}$ | | | 5 | | 4 | | 3 | ns |
| Clock fall time | t$_{KF}$ | | | 5 | | 4 | | 3 | ns |

**Reset**

| Parameter | Symbol | Test Conditions | μPD70732-16 | | μPD70732-20 | | μPD70732-25 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| $\overline{\text{RESET}}$ hold time (from V$_{DD}$ VALID) | t$_{HVR}$ | | 1000 + 20 t$_{CYKR}$ | | 1000 + 20 t$_{CYKR}$ | | 1000 + 20 t$_{CYKR}$ | | ns |
| Clock cycle (at reset) | t$_{CYKR}$ | | 62.5 | 1000 | 50 | 1000 | 40 | 1000 | ns |
| Clock high-level time (at reset) | t$_{KKHR}$ | | 26 | | 21 | | 17 | | ns |
| Clock low-level time (at reset) | t$_{KKLR}$ | | 26 | | 21 | | 17 | | ns |
| $\overline{\text{RESET}}$ setup time (to CLK↓, active) | t$_{SRKF}$ | | 10 | | 10 | | 10 | | ns |
| $\overline{\text{RESET}}$ setup time (to CLK↓, inactive) | t$_{SRKR}$ | | 10 | | 10 | | 10 | | ns |
| $\overline{\text{RESET}}$ hold time (from CLK↓) | t$_{HKR}$ | | 10 | | 10 | | 10 | | ns |
| $\overline{\text{RESET}}$ pulse low-level width (to CLK↓) | t$_{WRL}$ | | 20 t$_{CYKR}$ | | 20 t$_{CYKR}$ | | 20 t$_{CYKR}$ | | ns |

**Memory, I/O Access**

| Parameter | Symbol | Test Conditions | µPD70732-16 | | µPD70732-20 | | µPD70732-25 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| Address, etc. output delay time (from CLK↑) | $t_{DKA}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| Address, etc. ouput hold time (from CLK↑) | $t_{HKA}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| $\overline{BCYST}$ output delay time (from CLK↑) | $t_{DKBC}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| $\overline{BCYST}$ output hold time (from CLK↑) | $t_{HKBC}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| $\overline{DA}$ output delay time (from CLK↑) | $t_{DKDA}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| $\overline{DA}$ output hold time (from CLK↑) | $t_{HKDA}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| $\overline{READY}$ setup time (to CLK↓) | $t_{SRYK}$ | | 6 | | 5 | | 4 | | ns |
| $\overline{READY}$ hold time (from CLK↓) | $t_{HKRY}$ | | 5 | | 5 | | 4 | | ns |
| Data setup time (to CLK↑) | $t_{SDK}$ | | 6 | | 5 | | 4 | | ns |
| Data hold time (from CLK↑) | $t_{HKD}$ | | 5 | | 5 | | 4 | | ns |
| Data output delay time (from active, from CLK↓) | $t_{DKDT}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| Data output hold time (to active, from CLK↓) | $t_{HKDT}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| Data output delay time (from float, from CLK↓) | $t_{LZKDT}$ | | 5 | 25 | 5 | 20 | 5 | 20 | ns |
| Data output hold time (to float, from CLK↓) | $t_{HZKDT}$ | | 5 | 25 | 5 | 20 | 5 | 20 | ns |

**Dynamic Bus Sizing**

| Parameter | Symbol | Test Conditions | µPD70732-16 | | µPD70732-20 | | µPD70732-25 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| $\overline{SZRQ}$ setup time (to CLK↓) | $t_{SSZK}$ | | 6 | | 5 | | 4 | | ns |
| $\overline{SZRQ}$ hold time (from CLK↓) | $t_{HKSZ}$ | | 5 | | 5 | | 4 | | ns |

**Interrupt**

| Parameter | Symbol | Test Conditions | µPD70732-16 | | µPD70732-20 | | µPD70732-25 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| $\overline{NMI}$ setup time (to CLK↓) | $t_{SNK}$ | | 6 | | 5 | | 4 | | ns |
| $\overline{NMI}$ hold time (from CLK↓) | $t_{HKN}$ | | 5 | | 5 | | 4 | | ns |
| INT, etc. setup time (to CLK↑) | $t_{SIK}$ | | 6 | | 5 | | 4 | | ns |
| INT, etc. hold time (from CLK↑) | $t_{HKI}$ | | 5 | | 5 | | 4 | | ns |

**Bus Hold**

| Parameter | Symbol | Test Conditions | µPD70732-16 | | µPD70732-20 | | µPD70732-25 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| $\overline{\text{HLDRQ}}$ setup time (to CLK↓) | $t_{SHQK}$ | | 6 | | 5 | | 4 | | ns |
| $\overline{\text{HLDRQ}}$ hold time (from CLK↓) | $t_{HKHQ}$ | | 5 | | 5 | | 4 | | ns |
| $\overline{\text{HLDAK}}$ output delay time (from CLK↑) | $t_{DKHA}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| $\overline{\text{HLDAK}}$ output hold time (from CLK↑) | $t_{HKHA}$ | | 2 | 20 | 2 | 15 | 2 | 15 | ns |
| Address, etc. delay time (from active, from CLK↑) | $t_{HZKA}$ | | 2 | 25 | 2 | 20 | 2 | 20 | ns |
| Address, etc. delay time (from float, from CLK↑) | $t_{LZKA}$ | | 2 | 25 | 2 | 20 | 2 | 20 | ns |
| Data delay time (from active, from CLK↓) | $t_{HZKD}$ | | 5 | 25 | 5 | 20 | 5 | 20 | ns |
| Data delay time (from float, from CLK↓) | $t_{LZKD}$ | | 5 | 25 | 5 | 20 | 5 | 20 | ns |
| $\overline{\text{BCYST}}$ delay time (from active, from CLK↑) | $t_{HZKBC}$ | | 2 | 25 | 2 | 20 | 2 | 20 | ns |
| $\overline{\text{BCYST}}$ delay time (from float, from CLK↑) | $t_{LZKBC}$ | | 2 | 25 | 2 | 20 | 2 | 20 | ns |
| $\overline{\text{DA}}$ delay time (from active, from CLK↑) | $t_{HZKDA}$ | | 2 | 25 | 2 | 20 | 2 | 20 | ns |
| $\overline{\text{DA}}$ delay time (from float, from CLK↑) | $t_{LZKDA}$ | | 2 | 25 | 2 | 20 | 2 | 20 | ns |

**AC Test Input Waveform (Except CLK)**



**AC Test Input Waveform (CLK)**



**AC Test Output Test Points**

**Load Conditions**

V810
output pin ◯————————┐
                     │
                   ──┴── $C_L$ = 100 pF
                   ──┬──
                     │
                    ⏚
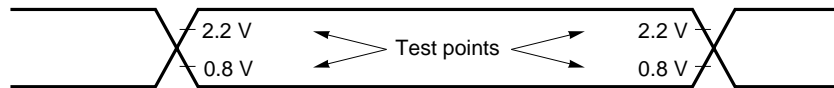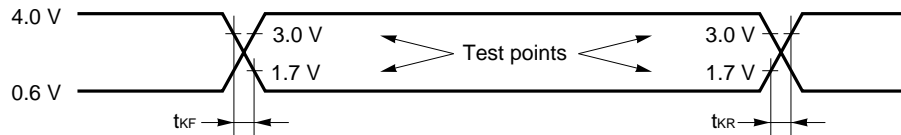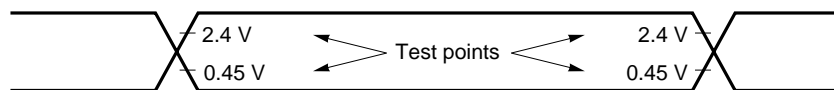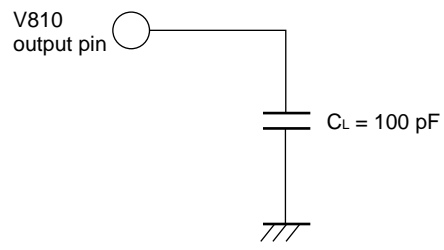
**(2) $T_A$ = –40 to +85°C**

**Absolute Maximum Ratings ($T_A$ = 25°C)**

| Parameter | Symbol | Test Conditions | Rating | Unit |
|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | | –0.5 to +7.0 | V |
| Input voltage | $V_I$ | $V_{DD}$ = +5 V ± 10% | –0.5 to $V_{DD}$ + 0.3 | V |
| Clock Input voltage | $V_K$ | $V_{DD}$ = +5 V ± 10% | –0.5 to $V_{DD}$ + 0.3 | V |
| Output voltage | $V_O$ | $V_{DD}$ = +5 V ± 10% | –0.5 to $V_{DD}$ + 0.3 | V |
| Operating ambient temperature | $T_A$ | | –40 to +85 | °C |
| Storage temperature | $T_{stg}$ | | –65 to +150 | °C |

**Cautions 1. Do not directly interconnect IC product output (or input/output) pins, or directly connect $V_{DD}$ or $V_{CC}$ to GND. However, open-drain pins and open-collector pins can be interconnected. Direct connection is also possible for an external circuit using timing design that avoids output collision with a pin that becomes high-impedance.**

**2. Product quality may suffer if the absolute maximum rating is exceeded for even a single parameter, or even momentarily.**
**In other words, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore, the product must be used under conditions which ensure that the absolute maximum ratings are not exceeded. As far as possible, the product should be used in a state in which the rated value is not approached. The ratings and test conditions shown in the DC characteristics and AC characteristics are the normal operation and quality assurance ranges of the product.**

**DC Characteristics ($T_A$ = –40 to +85°C, $V_{DD}$ = +5V ± 10%)**

| Parameter | Symbol | Test Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Clock input voltage, high | $V_{KH}$ | | 4.0 | | $V_{DD}$ + 0.3 | V |
| Clock input voltage, low | $V_{KL}$ | | –0.5 | | +0.6 | V |
| Input voltage, high | $V_{IH}$ | | 2.2 | | $V_{DD}$ + 0.3 | V |
| Input voltage, low | $V_{IL}$ | | –0.5 | | +0.8 | V |
| Output voltage, high | $V_{OH}$ | $I_{OH}$ = –400 µA | 2.4 | | | V |
| Output voltage, low | $V_{OL}$ | $I_{OL}$ = 3.2 mA | | | 0.45 | V |
| Input leak current, high | $I_{LIH}$ | $V_{IN}$ = $V_{DD}$ | | | 10 | µA |
| Input leak current, low | $I_{LIL}$ | $V_{IN}$ = 0 V | | | –10 | µA |
| Output leak current, high | $I_{LOH}$ | $V_O$ = $V_{DD}$ | | | 10 | µA |
| Output leak current, low | $I_{LOL}$ | $V_O$ = 0 V | | | –10 | µA |
| Supply current | $I_{DD}$ | f = 20 MHz | | 80[Note 2] | 200 | mA |
| | | Stopping clock[Note 1] | | 5 | | µA |

**Notes 1.** $V_{IL}$ = 0 V, $V_{IH}$ = $V_{DD}$ applied
**2.** In general benchmark test (Output pins are open.)

**Remark** Operating supply current is approximately proportional to operating clock frequency.

**Capacitance (T$_A$ = 25°C, V$_{DD}$ = +5 V ± 10%)**

| Parameter | Symbol | Test Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| Input capacitance | C$_I$ | f$_C$ = 1 MHz | | 15 | pF |
| I/O capacitance | C$_{IO}$ | | | 15 | pF |

**AC Characteristics (T$_A$ = −40 to +85°C, V$_{DD}$ = +5V ± 10%)**

**Clock Input**

| Parameter | Symbol | Test Conditions | μPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| Clock cycle | t$_{CYK}$ | | 50 | | ns |
| Clock pulse high-level width | t$_{KKH}$ | | 21 | | ns |
| Clock pulse low-level width | t$_{KKL}$ | | 21 | | ns |
| Clock rise time | t$_{KR}$ | | | 4 | ns |
| Clock fall time | t$_{KF}$ | | | 4 | ns |

**Reset**

| Parameter | Symbol | Test Conditions | μPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{\text{RESET}}$ hold time (from V$_{DD}$ VALID) | t$_{HVR}$ | | 1000 + 20 t$_{CYKR}$ | | ns |
| Clock cycle (at reset) | t$_{CYKR}$ | | 50 | 1000 | ns |
| Clock high-level time (at reset) | t$_{KKHR}$ | | 21 | | ns |
| Clock low-level time (at reset) | t$_{KKLR}$ | | 21 | | ns |
| $\overline{\text{RESET}}$ setup time (to CLK↓, active) | t$_{SRKF}$ | | 10 | | ns |
| $\overline{\text{RESET}}$ setup time (to CLK↓, inactive) | t$_{SRKR}$ | | 10 | | ns |
| $\overline{\text{RESET}}$ hold time (from CLK↓) | t$_{HKR}$ | | 10 | | ns |
| $\overline{\text{RESET}}$ pulse low-level width (to CLK↓) | t$_{WRL}$ | | 20 t$_{CYKR}$ | | ns |

**Memory, I/O Access**

| Parameter | Symbol | Test Conditions | μPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| Address, etc. ouput delay time (from CLK↑) | $t_{DKA}$ | | 1 | 15 | ns |
| Address, etc. ouput hold time (from CLK↑) | $t_{HKA}$ | | 1 | 15 | ns |
| $\overline{BCYST}$ output delay time (from CLK↑) | $t_{DKBC}$ | | 1 | 15 | ns |
| $\overline{BCYST}$ output hold time (from CLK↑) | $t_{HKBC}$ | | 1 | 15 | ns |
| $\overline{DA}$ output delay time (from CLK↑) | $t_{DKDA}$ | | 1 | 15 | ns |
| $\overline{DA}$ output hold time (from CLK↑) | $t_{HKDA}$ | | 1 | 15 | ns |
| $\overline{READY}$ setup time (to CLK↓) | $t_{SRYK}$ | | 5 | | ns |
| $\overline{READY}$ hold time (from CLK↓) | $t_{HKRY}$ | | 5 | | ns |
| Data setup time (to CLK↑) | $t_{SDK}$ | | 5 | | ns |
| Data hold time (from CLK↑) | $t_{HKD}$ | | 5 | | ns |
| Data output delay time (from active, from CLK↓) | $t_{DKDT}$ | | 1 | 15 | ns |
| Data output hold time (to active, from CLK↓) | $t_{HKDT}$ | | 1 | 15 | ns |
| Data output delay time (from float, from CLK↓) | $t_{LZKDT}$ | | 5 | 20 | ns |
| Data output hold time (to float, from CLK↓) | $t_{HZKDT}$ | | 5 | 20 | ns |

**Dynamic Bus Sizing**

| Parameter | Symbol | Test Conditions | μPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{SZRQ}$ setup time (to CLK↓) | $t_{SSZK}$ | | 5 | | ns |
| $\overline{SZRQ}$ hold time (from CLK↓) | $t_{HKSZ}$ | | 5 | | ns |

**Interrupt**

| Parameter | Symbol | Test Conditions | μPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{NMI}$ setup time (to CLK↓) | $t_{SNK}$ | | 5 | | ns |
| $\overline{NMI}$ hold time (from CLK↓) | $t_{HKN}$ | | 5 | | ns |
| INT, etc. setup time (to CLK↑) | $t_{SIK}$ | | 5 | | ns |
| INT, etc. hold time (from CLK↑) | $t_{HKI}$ | | 5 | | ns |

**Bus Hold**

| Parameter | Symbol | Test Conditions | µPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{\text{HLDRQ}}$ setup time (to CLK↓) | $t_{SHQK}$ | | 5 | | ns |
| $\overline{\text{HLDRQ}}$ hold time (from CLK↓) | $t_{HKHQ}$ | | 5 | | ns |
| $\overline{\text{HLDAK}}$ output delay time (from CLK↑) | $t_{DKHA}$ | | 1 | 15 | ns |
| $\overline{\text{HLDAK}}$ output hold time (from CLK↑) | $t_{HKHA}$ | | 1 | 15 | ns |
| Address, etc. delay time (from active, from CLK↑) | $t_{HZKA}$ | | 2 | 20 | ns |
| Address, etc. delay time (from float, from CLK↑) | $t_{LZKA}$ | | 2 | 20 | ns |
| Data delay time (from active, from CLK↓) | $t_{HZKD}$ | | 5 | 20 | ns |
| Data delay time (from float, from CLK↓) | $t_{LZKD}$ | | 5 | 20 | ns |
| $\overline{\text{BCYST}}$ delay time (from active, from CLK↑) | $t_{HZKBC}$ | | 2 | 20 | ns |
| $\overline{\text{BCYST}}$ delay time (from float, from CLK↑) | $t_{LZKBC}$ | | 2 | 20 | ns |
| $\overline{\text{DA}}$ delay time (from active, from CLK↑) | $t_{HZKDA}$ | | 2 | 20 | ns |
| $\overline{\text{DA}}$ delay time (from float, from CLK↑) | $t_{LZKDA}$ | | 2 | 20 | ns |

**AC Test Input Waveform (Except CLK)**

2.2 V  Test points  2.2 V
0.8 V  0.8 V

**AC Test Input Waveform (CLK)**

4.0 V
3.0 V  Test points  3.0 V
1.7 V  1.7 V
0.6 V
$t_{KF}$  $t_{KR}$

**AC Test Output Test Points**

2.4 V  Test points  2.4 V
0.45 V  0.45 V

**Load Conditions**

V810 output pin

$C_L = 100$ pF
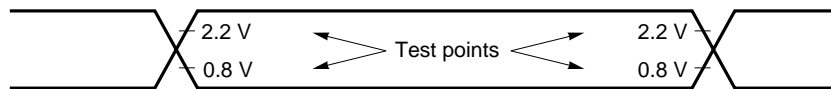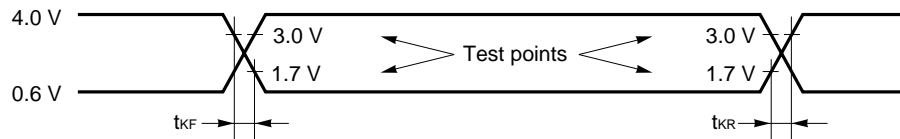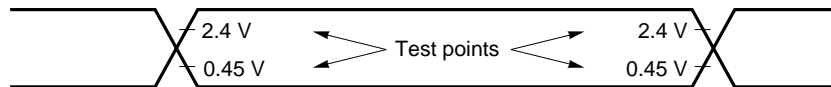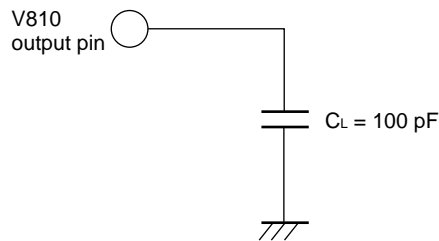
## 10.2 Specifications When $V_{DD}$ = 2.7 to 3.6 V

**Absolute Maximum Ratings ($T_A$ = 25°C)**

| Parameter | Symbol | Test Conditions | Rating | Unit |
|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | | −0.5 to +7.0 | V |
| Input voltage | $V_I$ | $V_{DD}$ = 2.7 to 3.6 V | −0.5 to $V_{DD}$ + 0.3 | V |
| Clock Input voltage | $V_K$ | $V_{DD}$ = 2.7 to 3.6 V | −0.5 to $V_{DD}$ + 0.3 | V |
| Output voltage | $V_O$ | $V_{DD}$ = 2.7 to 3.6 V | −0.5 to $V_{DD}$ + 0.3 | V |
| Operaitng ambient temperature | $T_A$ | | −40 to +85 | °C |
| Storage temperature | $T_{stg}$ | | −65 to +150 | °C |

**Cautions 1. Do not directly interconnect IC product output (or input/output) pins, or directly connect $V_{DD}$ or $V_{CC}$ to GND. However, open-drain pins and open-collector pins can be interconnected. Direct connection is also possible for an external circuit using timing design that avoids output collision with a pin that becomes high-impedance.**

**2. Product quality may suffer if the absolute maximum rating is exceeded for even a single parameter, or even momentarily.**
**In other words, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore, the product must be used under conditions which ensure that the absolute maximum ratings are not exceeded. As far as possible, the product should be used in a state in which the rated value is not approached. The ratings and test conditions shown in the DC characteristics and AC characteristics are the normal operation and quality assurance ranges of the product.**

**DC Characteristics ($T_A$ = −40 to +85°C, $V_{DD}$ = 2.7 to 3.6 V)**

| Parameter | Symbol | Test Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Clock input voltage, high | $V_{KH}$ | | 0.8 $V_{DD}$ | | $V_{DD}$ + 0.3 | V |
| Clock input voltage, low | $V_{KL}$ | | −0.5 | | +0.2 $V_{DD}$ | V |
| Input voltage, high | $V_{IH}$ | | 2.0 | | $V_{DD}$ + 0.3 | V |
| Input voltage, low | $V_{IL}$ | | −0.5 | | +0.6 | V |
| Output voltage, high | $V_{OH}$ | $I_{OH}$ = −2.0 mA | 0.85 $V_{DD}$ | | | V |
| | | $I_{OH}$ = −100 µA | $V_{DD}$ − 0.2 | | | V |
| Output voltage, low | $V_{OL}$ | $I_{OL}$ = 3.2 mA | | | 0.4 | V |
| Input leak current, high | $I_{LIH}$ | $V_{IN}$ = $V_{DD}$ | | | 5 | µA |
| Input leak current, low | $I_{LIL}$ | $V_{IN}$ = 0 V | | | −5 | µA |
| Output leak current, high | $I_{LOH}$ | $V_O$ = $V_{DD}$ | | | 5 | µA |
| Output leak current, low | $I_{LOL}$ | $V_O$ = 0 V | | | −5 | µA |
| Supply current | $I_{DD}$ | f = 16 MHz | | 38[Note 2] | 100 | mA |
| | | Stopping clock[Note 1] | | 3 | 30 | µA |

**Notes 1.** $V_{IL}$ = 0 V, $V_{IH}$ = $V_{DD}$ applied

**2.** In general benchmark test (Output pins are open.)

**Remark** Operating supply current is approximately proportional to operating clock frequency.

**Capacitance (T$_A$ = 25°C, V$_{DD}$ = 2.7 to 3.6 V)**

| Parameter | Symbol | Test Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| Input capacitance | C$_I$ | f$_C$ = 1 MHz | | 15 | pF |
| I/O capacitance | C$_{IO}$ | | | 15 | pF |

**AC Characteristics (T$_A$ = –40 to +85°C, V$_{DD}$ = 2.7 to 3.6 V)**

**Clock Input**

| Parameter | Symbol | Test Conditions | μPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| Clock cycle | t$_{CYK}$ | | 62.5 | | ns |
| Clock pulse high-level width | t$_{KKH}$ | | 26 | | ns |
| Clock pulse low-level width | t$_{KKL}$ | | 26 | | ns |
| Clock rise time | t$_{KR}$ | | | 5 | ns |
| Clock fall time | t$_{KF}$ | | | 5 | ns |

**Reset**

| Parameter | Symbol | Test Conditions | μPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{RESET}$ hold time (from V$_{DD}$ VALID) | t$_{HVR}$ | | 1000 + 20t$_{CYKR}$ | | ns |
| Clock cycle (at reset) | t$_{CYKR}$ | | 62.5 | 1000 | ns |
| Clock high-level time (at reset) | t$_{KKHR}$ | | 26 | | ns |
| Clock low-level time (at reset) | t$_{KKLR}$ | | 26 | | ns |
| $\overline{RESET}$ setup time (to CLK↓, active) | t$_{SRKF}$ | | 10 | | ns |
| $\overline{RESET}$ setup time (to CLK↓, inactive) | t$_{SRKR}$ | | 10 | | ns |
| $\overline{RESET}$ hold time (from CLK↓) | t$_{HKR}$ | | 10 | | ns |
| $\overline{RESET}$ pulse low-level width (to CLK↓) | t$_{WRL}$ | | 20t$_{CYKR}$ | | ns |

**Memory, I/O Access**

| Parameter | Symbol | Test Conditions | µPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| Address etc. output delay time (from CLK↑) | $t_{DKA}$ | | 1 | 25 | ns |
| Address etc. output hold time (from CLK↑) | $t_{HKA}$ | | 1 | 25 | ns |
| $\overline{BCYST}$ output delay time (from CLK↑) | $t_{DKBC}$ | | 1 | 25 | ns |
| $\overline{BCYST}$ output hold time (from CLK↑) | $t_{HKBC}$ | | 1 | 25 | ns |
| $\overline{DA}$ output delay time (from CLK↑) | $t_{DKDA}$ | | 1 | 25 | ns |
| $\overline{DA}$ output hold time (from CLK↑) | $t_{HKDA}$ | | 1 | 25 | ns |
| $\overline{READY}$ setup time (to CLK↓) | $t_{SRYK}$ | | 8 | | ns |
| $\overline{READY}$ hold time (from CLK↓) | $t_{HKRY}$ | | 5 | | ns |
| Data setup time (to CLK↑) | $t_{SDK}$ | | 8 | | ns |
| Data hold time (from CLK↑) | $t_{HKD}$ | | 5 | | ns |
| Data output delay time (from active, from CLK↓) | $t_{DKDT}$ | | 1 | 35 | ns |
| Data output hold time (to active, from CLK↓) | $t_{HKDT}$ | | 1 | 35 | ns |
| Data output delay time (from float, from CLK↓) | $t_{LZKDT}$ | | 3 | 40 | ns |
| Data output hold time (to float, from CLK↓) | $t_{HZKDT}$ | | 3 | 40 | ns |

**Dynamic Bus Sizing**

| Parameter | Symbol | Test Conditions | µPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{SZRQ}$ setup time (to CLK↓) | $t_{SSZK}$ | | 8 | | ns |
| $\overline{SZRQ}$ hold time (from CLK↓) | $t_{HKSZ}$ | | 5 | | ns |

**Interrupt**

| Parameter | Symbol | Test Conditions | µPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{NMI}$ setup time (to CLK↓) | $t_{SNK}$ | | 8 | | ns |
| $\overline{NMI}$ hold time (from CLK↓) | $t_{HKN}$ | | 5 | | ns |
| INT etc. setup time (to CLK↑) | $t_{SIK}$ | | 8 | | ns |
| INT etc. hold time (from CLK↑) | $t_{HKI}$ | | 5 | | ns |

**Bus Hold**

| Parameter | Symbol | Test Conditions | µPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{\text{HLDRQ}}$ setup time (to CLK↓) | $t_{SHQK}$ | | 8 | | ns |
| $\overline{\text{HLDRQ}}$ hold time (from CLK↓) | $t_{HKHQ}$ | | 5 | | ns |
| $\overline{\text{HLDAK}}$ output delay time (from CLK↑) | $t_{DKHA}$ | | 1 | 25 | ns |
| $\overline{\text{HLDAK}}$ output hold time (from CLK↑) | $t_{HKHA}$ | | 1 | 25 | ns |
| Address, etc. delay time (from active, from CLK↑) | $t_{HZKA}$ | | 3 | 30 | ns |
| Address, etc. delay time (from float, from CLK↑) | $t_{LZKA}$ | | 3 | 30 | ns |
| Data delay time (from active, from CLK↓) | $t_{HZKD}$ | | 3 | 40 | ns |
| Data delay time (from float, from CLK↓) | $t_{LZKD}$ | | 3 | 40 | ns |
| $\overline{\text{BCYST}}$ delay time (from active, from CLK↑) | $t_{HZKBC}$ | | 3 | 30 | ns |
| $\overline{\text{BCYST}}$ delay time (from float, from CLK↑) | $t_{LZKBC}$ | | 3 | 30 | ns |
| $\overline{\text{DA}}$ delay time (from active, from CLK↑) | $t_{HZKDA}$ | | 3 | 30 | ns |
| $\overline{\text{DA}}$ delay time (from float, from CLK↑) | $t_{LZKDA}$ | | 3 | 30 | ns |

**AC Test Input Waveform (Except CLK)**
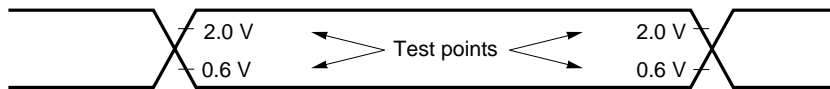


**AC Test Input Waveform (CLK)**
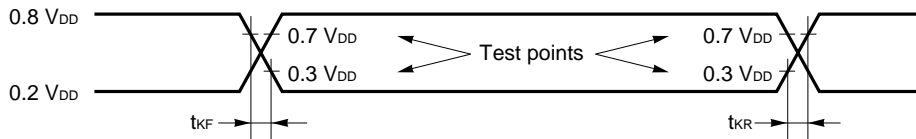


**AC Test Output Test Points**
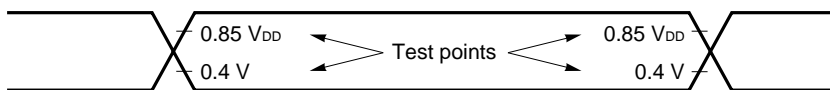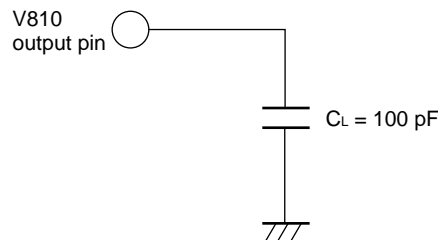


**Load Conditions**

### 10.3  Specifications When $V_{DD}$ = 2.2 to 3.6 V

**Absolute Maximum Ratings ($T_A$ = 25°C)**

| Parameter | Symbol | Test Conditions | Rating | Unit |
|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | | −0.5 to +7.0 | V |
| Input voltage | $V_I$ | $V_{DD}$ = 2.2 to 3.6 V | −0.5 to $V_{DD}$ + 0.3 | V |
| Clock Input voltage | $V_K$ | $V_{DD}$ = 2.2 to 3.6 V | −0.5 to $V_{DD}$ + 0.3 | V |
| Output voltage | $V_O$ | $V_{DD}$ = 2.2 to 3.6 V | −0.5 to $V_{DD}$ + 0.3 | V |
| Operaitng ambient temperature | $T_A$ | | −40 to +85 | °C |
| Storage temperature | $T_{stg}$ | | −65 to +150 | °C |

**Cautions   1.  Do not directly interconnect IC product output (or input/output) pins, or directly connect $V_{DD}$ or $V_{CC}$ to GND. However, open-drain pins and open-collector pins can be interconnected. Direct connection is also possible for an external circuit using timing design that avoids output collision with a pin that becomes high-impedance.**

**2.  Product quality may suffer if the absolute maximum rating is exceeded for even a single parameter, or even momentarily.**

**In other words, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore, the product must be used under conditions which ensure that the absolute maximum ratings are not exceeded.  As far as possible, the product should be used in a state in which the rated value is not approached. The ratings and test conditions shown in the DC characteristics and AC characteristics are the normal operation and quality assurance ranges of the product.**

**DC Characteristics ($T_A$ = −40 to +85°C, $V_{DD}$ = 2.2 to 3.6 V)**

| Parameter | Symbol | Test Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Clock input voltage, high | $V_{KH}$ | | 0.8 $V_{DD}$ | | $V_{DD}$ + 0.3 | V |
| Clock input voltage, low | $V_{KL}$ | | −0.5 | | +0.2 $V_{DD}$ | V |
| Input voltage, high | $V_{IH}$ | $V_{DD}$ • 2.5 V | 2.0 | | $V_{DD}$ + 0.3 | V |
| | | $V_{DD}$ - 2.5 V | 0.8 $V_{DD}$ | | $V_{DD}$ + 0.3 | V |
| Input voltage, low | $V_{IL}$ | | −0.5 | | +0.2 $V_{DD}$ | V |
| Output voltage, high | $V_{OH}$ | $I_{OH}$ = −2.0 mA | 0.85 $V_{DD}$ | | | V |
| | | $I_{OH}$ = −100 µA | $V_{DD}$ − 0.2 | | | V |
| Output voltage, low | $V_{OL}$ | $I_{OL}$ = 3.2 mA | | | 0.4 | V |
| Input leak current, high | $I_{LIH}$ | $V_{IN}$ = $V_{DD}$ | | | 5 | µA |
| Input leak current, low | $I_{LIL}$ | $V_{IN}$ = 0 V | | | −5 | µA |
| Output leak current, high | $I_{LOH}$ | $V_O$ = $V_{DD}$ | | | 5 | µA |
| Output leak current, low | $I_{LOL}$ | $V_O$ = 0 V | | | −5 | µA |
| Supply current | $I_{DD}$ | f = 10 MHz | | 24[Note 2] | 70 | mA |
| | | Stopping clock[Note 1] | | 3 | 30 | µA |

**Notes  1.**  $V_{IL}$ = 0 V, $V_{IH}$ = $V_{DD}$ applied

**2.**  In general benchmark test (Output pins are open.)

**Remark**   Operating supply current is approximately proportional to operating clock frequency.

### Capacitance (T$_A$ = 25°C, V$_{DD}$ = 2.2 to 3.6 V)

| Parameter | Symbol | Test Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| Input capacitance | C$_I$ | f$_C$ = 1 MHz | | 15 | pF |
| I/O capacitance | C$_{IO}$ | | | 15 | pF |

### AC Characteristics (T$_A$ = –40 to +85°C, V$_{DD}$ = 2.2 to 3.6 V)

### Clock Input

| Parameter | Symbol | Test Conditions | µPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| Clock cycle | t$_{CYK}$ | | 100 | | ns |
| Clock pulse high-level width | t$_{KKH}$ | | 40 | | ns |
| Clock pulse low-level width | t$_{KKL}$ | | 40 | | ns |
| Clock rise time | t$_{KR}$ | | | 10 | ns |
| Clock fall time | t$_{KF}$ | | | 10 | ns |

### Reset

| Parameter | Symbol | Test Conditions | µPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{RESET}$ hold time (from V$_{DD}$ VALID) | t$_{HVR}$ | | 1000 + 20t$_{CYKR}$ | | ns |
| Clock cycle (at reset) | t$_{CYKR}$ | | 100 | 1000 | ns |
| Clock high-level time (at reset) | t$_{KKHR}$ | | 40 | | ns |
| Clock low-level time (at reset) | t$_{KKLR}$ | | 40 | | ns |
| $\overline{RESET}$ setup time (to CLK↓, active) | t$_{SRKF}$ | | 10 | | ns |
| $\overline{RESET}$ setup time (to CLK↓, inactive) | t$_{SRKR}$ | | 10 | | ns |
| $\overline{RESET}$ hold time (from CLK↓) | t$_{HKR}$ | | 15 | | ns |
| $\overline{RESET}$ pulse low-level width (to CLK↓) | t$_{WRL}$ | | 20t$_{CYKR}$ | | ns |

**Memory, I/O Access**

| Parameter | Symbol | Test Conditions | μPD70732-25 MIN. | μPD70732-25 MAX. | Unit |
|---|---|---|---|---|---|
| Address, etc. output delay time (from CLK↑) | $t_{DKA}$ | | 1 | 35 | ns |
| Address, etc. output hold time (from CLK↑) | $t_{HKA}$ | | 1 | 35 | ns |
| $\overline{BCYST}$ output delay time (from CLK↑) | $t_{DKBC}$ | | 1 | 35 | ns |
| $\overline{BCYST}$ output hold time (from CLK↑) | $t_{HKBC}$ | | 1 | 35 | ns |
| $\overline{DA}$ output delay time (from CLK↑) | $t_{DKDA}$ | | 1 | 35 | ns |
| $\overline{DA}$ output hold time (from CLK↑) | $t_{HKDA}$ | | 1 | 35 | ns |
| $\overline{READY}$ setup time (to CLK↓) | $t_{SRYK}$ | | 15 | | ns |
| $\overline{READY}$ hold time (from CLK↓) | $t_{HKRY}$ | | 5 | | ns |
| Data setup time (to CLK↑) | $t_{SDK}$ | | 15 | | ns |
| Data hold time (from CLK↑) | $t_{HKD}$ | | 5 | | ns |
| Data output delay time (from active, from CLK↓) | $t_{DKDT}$ | | 1 | 50 | ns |
| Data output hold time (to active, from CLK↓) | $t_{HKDT}$ | | 1 | 50 | ns |
| Data output delay time (from float, from CLK↓) | $t_{LZKDT}$ | | 3 | 50 | ns |
| Data output hold time (to float, from CLK↓) | $t_{HZKDT}$ | | 3 | 50 | ns |

**Dynamic Bus Sizing**

| Parameter | Symbol | Test Conditions | μPD70732-25 MIN. | μPD70732-25 MAX. | Unit |
|---|---|---|---|---|---|
| $\overline{SZRQ}$ setup time (to CLK↓) | $t_{SSZK}$ | | 15 | | ns |
| $\overline{SZRQ}$ hold time (from CLK↓) | $t_{HKSZ}$ | | 5 | | ns |

**Interrupt**

| Parameter | Symbol | Test Conditions | μPD70732-25 MIN. | μPD70732-25 MAX. | Unit |
|---|---|---|---|---|---|
| $\overline{NMI}$ setup time (to CLK↓) | $t_{SNK}$ | | 15 | | ns |
| $\overline{NMI}$ hold time (from CLK↓) | $t_{HKN}$ | | 5 | | ns |
| INT, etc. setup time (to CLK↑) | $t_{SIK}$ | | 15 | | ns |
| INT, etc. hold time (from CLK↑) | $t_{HKI}$ | | 5 | | ns |

**Bus Hold**

| Parameter | Symbol | Test Conditions | µPD70732-25 | | Unit |
|---|---|---|---|---|---|
| | | | MIN. | MAX. | |
| $\overline{\text{HLDRQ}}$ setup time (to CLK↓) | $t_{SHQK}$ | | 15 | | ns |
| $\overline{\text{HLDRQ}}$ hold time (from CLK↓) | $t_{HKHQ}$ | | 5 | | ns |
| $\overline{\text{HLDAK}}$ output delay time (from CLK↑) | $t_{DKHA}$ | | 1 | 35 | ns |
| $\overline{\text{HLDAK}}$ output hold time (from CLK↑) | $t_{HKHA}$ | | 1 | 35 | ns |
| Address, etc. delay time (from active, from CLK↑) | $t_{HZKA}$ | | 3 | 35 | ns |
| Address, etc. delay time (from float, from CLK↑) | $t_{LZKA}$ | | 3 | 35 | ns |
| Data delay time (from active, from CLK↓) | $t_{HZKD}$ | | 3 | 50 | ns |
| Data delay time (from float, from CLK↓) | $t_{LZKD}$ | | 3 | 50 | ns |
| $\overline{\text{BCYST}}$ delay time (from active, from CLK↑) | $t_{HZKBC}$ | | 3 | 35 | ns |
| $\overline{\text{BCYST}}$ delay time (from float, from CLK↑) | $t_{LZKBC}$ | | 3 | 35 | ns |
| $\overline{\text{DA}}$ delay time (from active, from CLK↑) | $t_{HZKDA}$ | | 3 | 35 | ns |
| $\overline{\text{DA}}$ delay time (from float, from CLK↑) | $t_{LZKDA}$ | | 3 | 35 | ns |

**AC Test Input Waveform (Except CLK)**
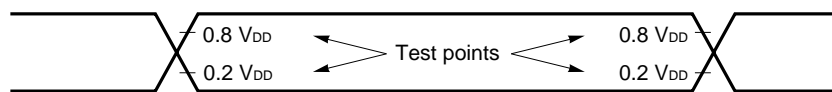


**AC Test Input Waveform (CLK)**
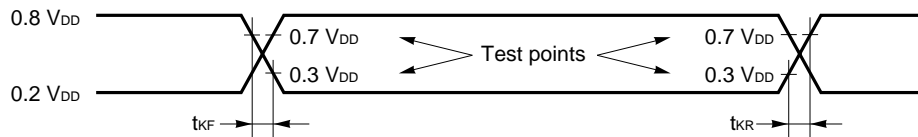


**AC Test Output Test Points**
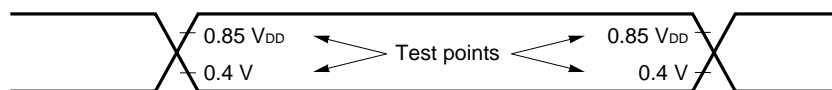


**Load Conditions**
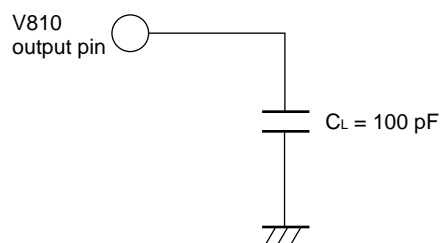
**Clock Timing**



**Reset Timing**



Clock stopping exception period
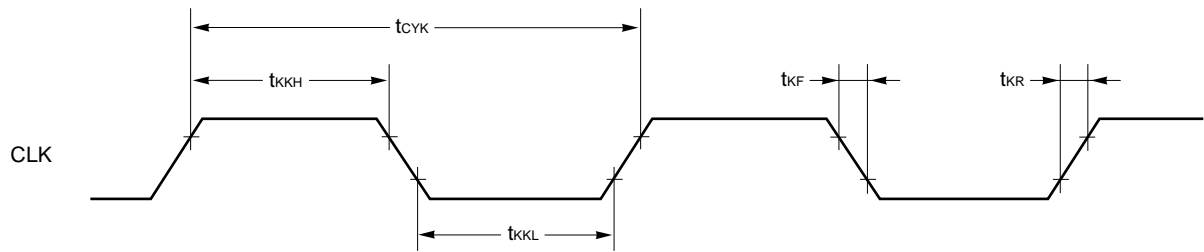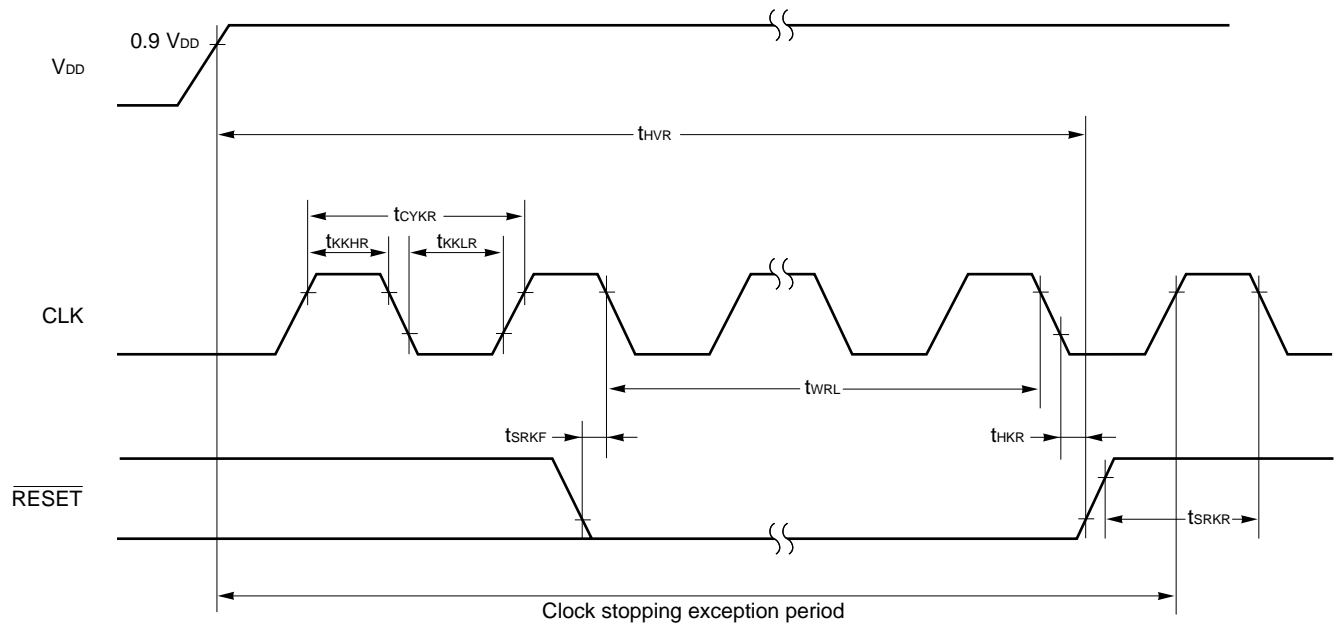
**Memory, I/O Access Timing**
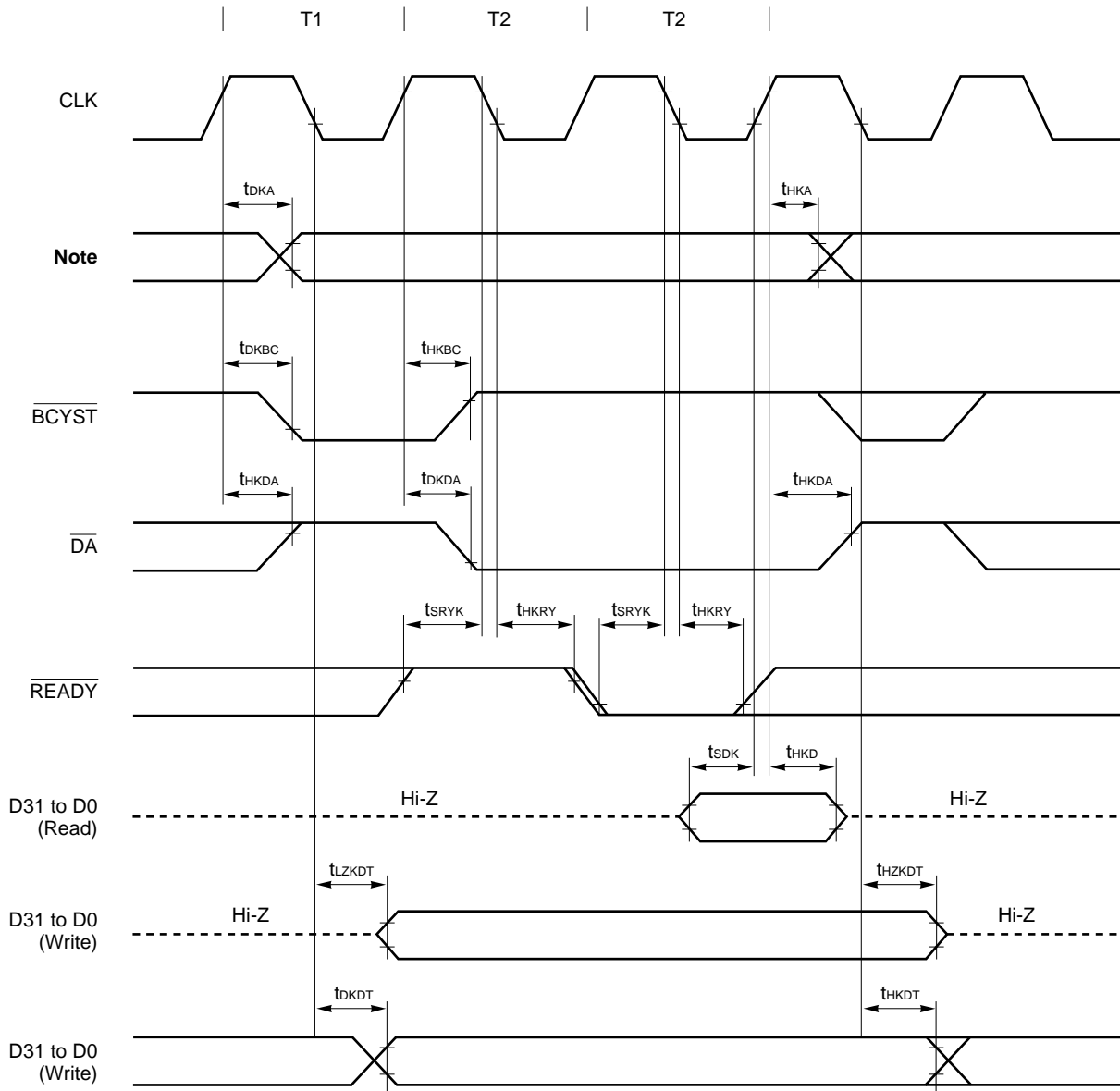


**Note** A31 to A1, $\overline{\text{BE3}}$ to $\overline{\text{BE0}}$, R/$\overline{\text{W}}$, $\overline{\text{MRQ}}$, ST1, ST0, BLOCK, $\overline{\text{ADRSERR}}$

**Dynamic Bus Sizing Timing**

CLK

T2

$t_{SSZK}$  $t_{HKSZ}$

$\overline{SZRQ}$

**Interrupt Timing**

CLK

$t_{SNK}$  $t_{HKN}$

$\overline{NMI}$

$t_{SIK}$  $t_{HKI}$

$\overline{INT}$,
$\overline{INTV3}$ to $\overline{INTV0}$

**Bus Hold Timing**

| T1 | T2 | TI | TH | TH | TH | TI | T1 |

CLK

$t_{SHQK}$   $t_{HKHQ}$   $t_{SHQK}$

$\overline{HLDRQ}$

$t_{DKHA}$   $t_{HKHA}$

$\overline{HLDAK}$

$t_{HZKA}$   $t_{LZKA}$

Note 1   Note 2

$t_{HZKD}$   $t_{LZKD}$

D31 to D0 (Write)   Note 2

$t_{HZKBC}$   $t_{LZKBC}$

$\overline{BCYST}$

$t_{HZKDA}$   $t_{LZKDA}$

$\overline{DA}$
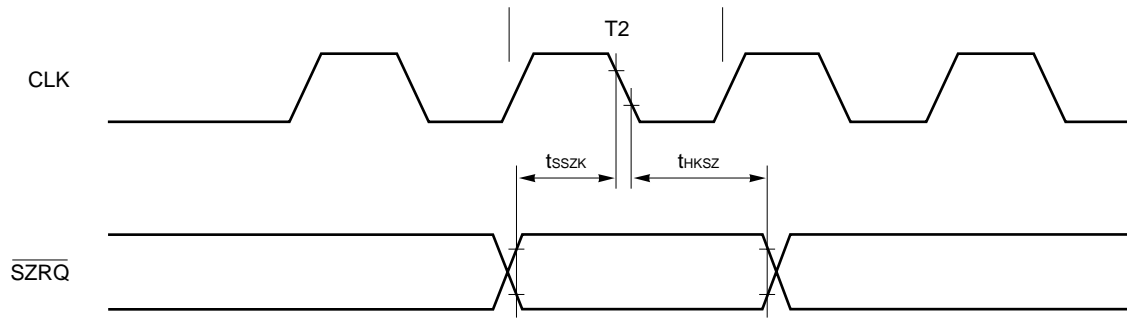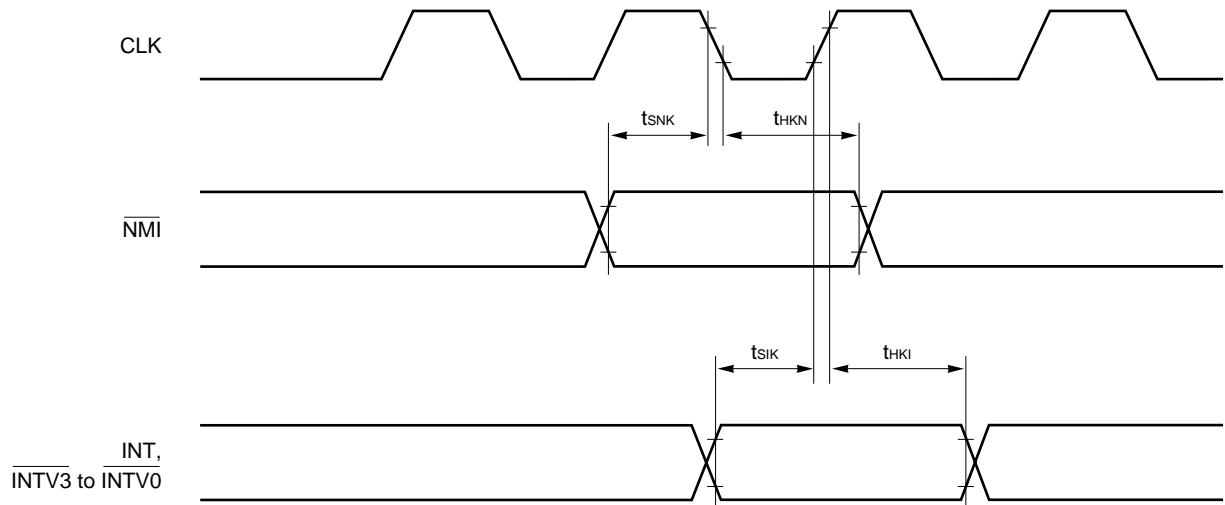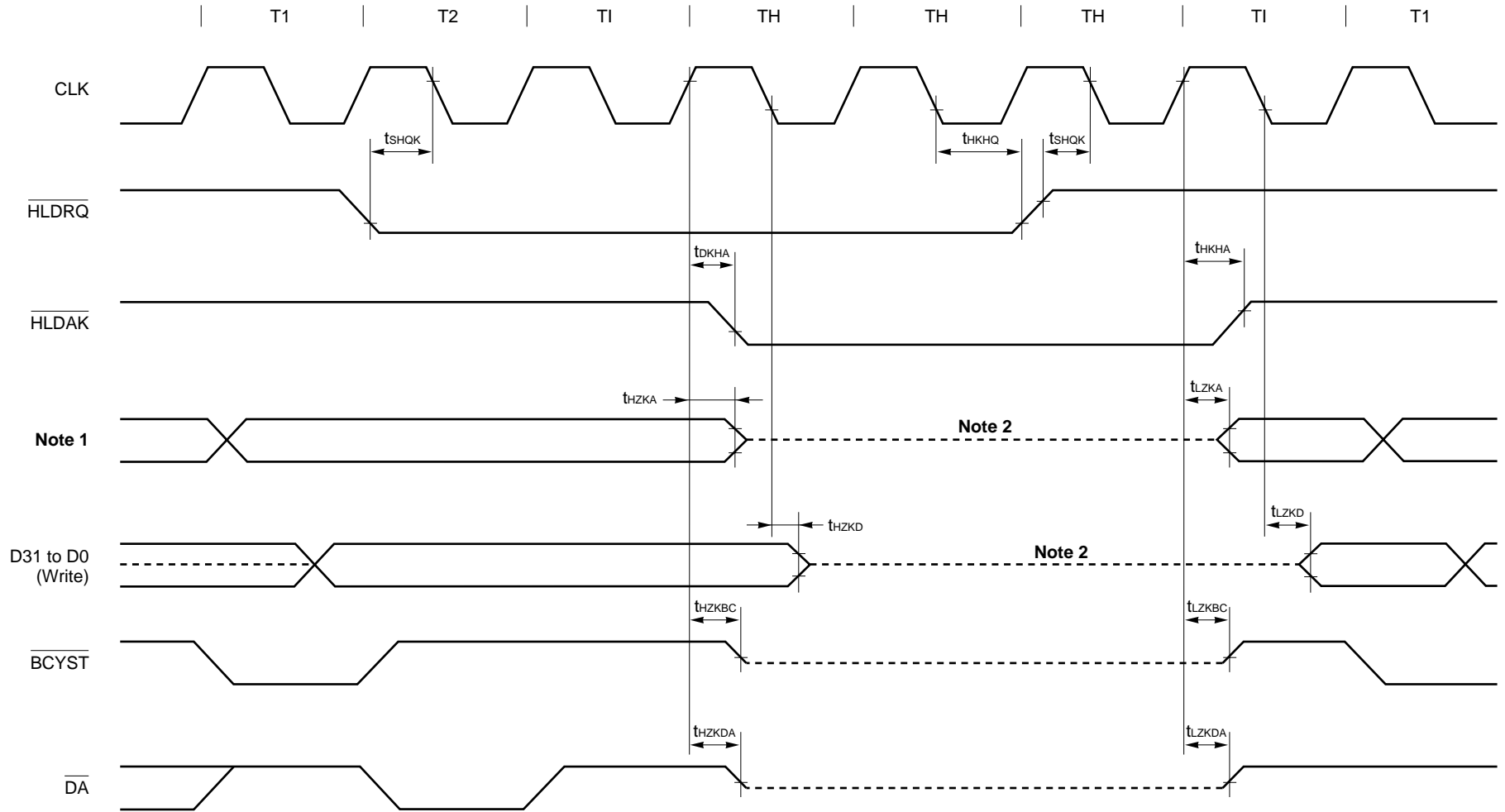
Notes 1. A31 to A1, $\overline{BE3}$ to $\overline{BE0}$, R/$\overline{W}$, $\overline{MRQ}$, ST1, ST0

2. The level immediately before the high-impedance state has been stored internally.

Remark A dashed line indicates high impedance.

## 11. PACKAGE DRAWINGS

**120-pin plastic QFP (28 x 28)**



detail of lead end
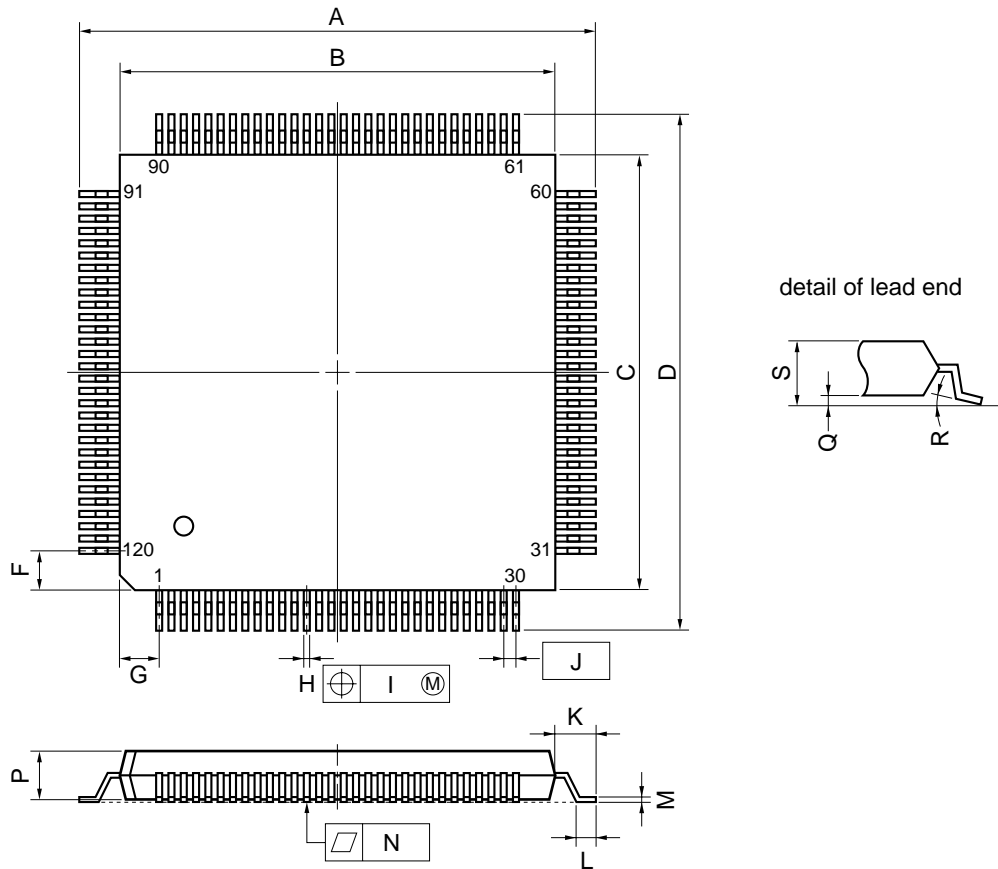
**NOTE**

Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|------|-------------|--------|
| A | 32.0±0.3 | 1.260±0.012 |
| B | 28.0±0.2 | $1.102^{+0.009}_{-0.008}$ |
| C | 28.0±0.2 | $1.102^{+0.009}_{-0.008}$ |
| D | 32.0±0.3 | 1.260±0.012 |
| F | 2.4 | 0.094 |
| G | 2.4 | 0.094 |
| H | 0.35±0.10 | $0.014^{+0.004}_{-0.005}$ |
| I | 0.15 | 0.006 |
| J | 0.8 (T.P.) | 0.031 (T.P.) |
| K | 2.0±0.2 | $0.079^{+0.009}_{-0.008}$ |
| L | 0.8±0.2 | $0.031^{+0.009}_{-0.008}$ |
| M | $0.15^{+0.10}_{-0.05}$ | $0.006^{+0.004}_{-0.003}$ |
| N | 0.1 | 0.004 |
| P | 3.2 | 0.126 |
| Q | 0.1±0.1 | 0.004±0.004 |
| R | 5°±5° | 5°±5° |
| S | 3.5 MAX. | 0.138 MAX. |

**P120GD-80-LBB, MBB-1**

★ **120-pin plastic TQFP (Fine pitch) (14 x 14)**



detail of lead end
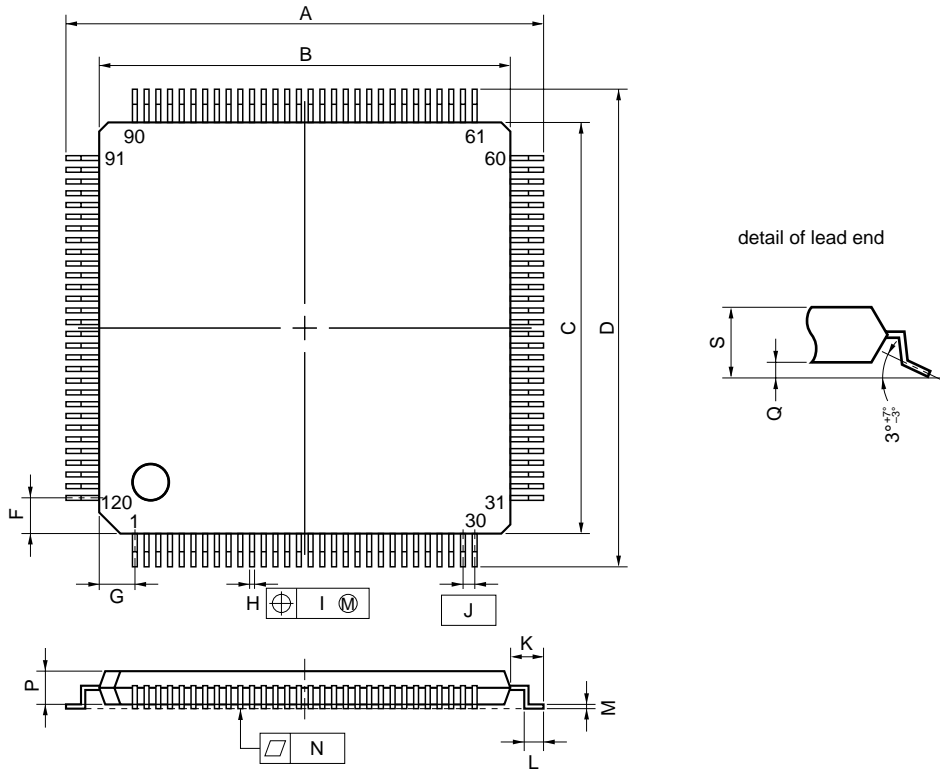
**NOTE**

Each lead centerline is located within 0.09 mm (0.004 inch) of its true position (T.P.) at maximum material condition.

**S120GC-40-9EV**

| ITEM | MILLIMETERS | INCHES |
|------|-------------|--------|
| A | 16.0±0.2 | 0.630±0.008 |
| B | 14.0±0.2 | $0.551^{+0.009}_{-0.008}$ |
| C | 14.0±0.2 | $0.551^{+0.009}_{-0.008}$ |
| D | 16.0±0.2 | 0.630±0.008 |
| F | 1.2 | 0.047 |
| G | 1.2 | 0.047 |
| H | 0.18±0.05 | 0.007±0.002 |
| I | 0.09 | 0.004 |
| J | 0.4 (T.P.) | 0.016 (T.P.) |
| K | 1.0±0.2 | $0.039^{+0.009}_{-0.008}$ |
| L | 0.5±0.2 | $0.020^{+0.008}_{-0.009}$ |
| M | 0.145±0.05 | $0.006^{+0.002}_{-0.003}$ |
| N | 0.08 | 0.003 |
| P | 1.0±0.1 | $0.039^{+0.005}_{-0.004}$ |
| Q | 0.1±0.05 | 0.004±0.002 |
| S | 1.2 MAX. | 0.048 MAX. |

**176-pin ceramic PGA (Seamweld)**



Index mark

(Bottom View)

Orientation pin
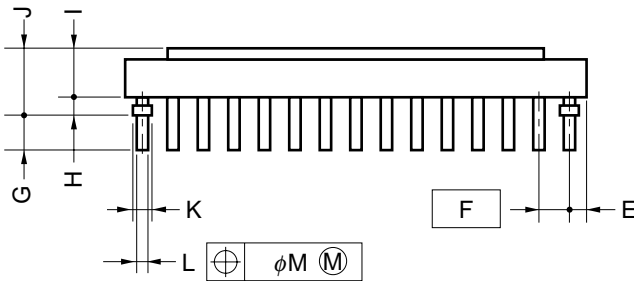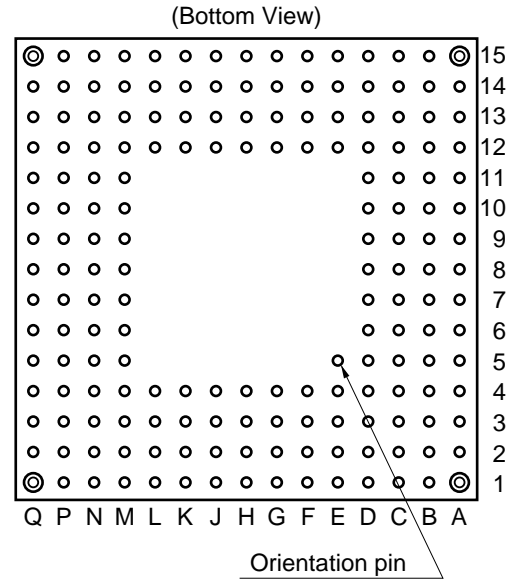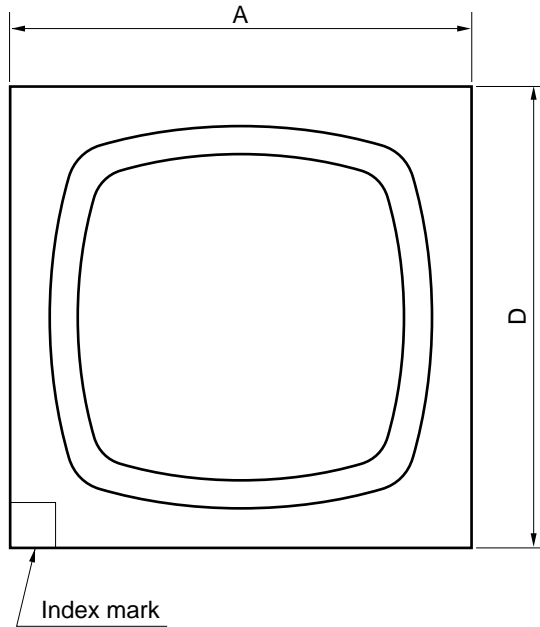
**NOTE**

Each lead centerline is located within $\phi$0.5 mm ($\phi$ 0.020 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|------|-------------|--------|
| A | 38.1±0.4 | $1.500^{+0.016}_{-0.015}$ |
| D | 38.1±0.4 | $1.500^{+0.016}_{-0.015}$ |
| E | 1.27 | 0.050 |
| F | 2.54 (T.P.) | 0.100 (T.P.) |
| G | 2.8±0.3 | $0.110^{+0.012}_{-0.011}$ |
| H | 0.5 MIN. | 0.019 MIN. |
| I | 2.81 | 0.111 |
| J | 4.57 MAX. | 0.180 MAX. |
| K | $\phi$1.2±0.2 | $\phi0.047^{+0.008}_{-0.007}$ |
| L | $\phi$0.46±0.05 | $\phi0.018^{+0.002}_{-0.001}$ |
| M | 0.5 | 0.020 |

**X176R-100A-1**

## 12. RECOMMENDED SOLDERING CONDITIONS

The µPD70732 should be soldered and mounted under the conditions recommended in the table below.

For details of recommended soldering conditions, refer to the information document **"Semiconductor Device Mounting Technology Manual" (C10535E).**

For soldering methods and conditions other than those recommended below, contact an NEC sales representative.

**Table 12-1.  Surface Mounting Type Soldering Conditions**

**(1)  µPD70732GD-16-LBB : 120-pin plastic QFP (28 x 28 mm)**
**µPD70732GD-20-LBB : 120-pin plastic QFP (28 x 28 mm)**
**µPD70732GD-25-LBB : 120-pin plastic QFP (28 x 28 mm)**

**E specification model only**

| Soldering Method | Soldering Conditions | Recommended Condition Symbol |
|---|---|---|
| Infrared reflow | Package peak temperature: 235°C, Duration: 30 sec. Max. (at 210°C or above), Number of times: Twice Max., Time limit: 7 days[Note] (thereafter 36 hours prebaking required at 125°C) | IR35-367-2 |
| VPS | Package peak temperature: 215°C, Duration: 40 sec. Max. (at 200°C or above), Number of times: Twice Max., Time limit: 7 days[Note] (thereafter 36 hours prebaking required at 125°C) | VP15-367-2 |
| Wave soldering | Solder bath temperature: 260°C Max., Duration: 10 sec. Max., Number of times: Once, Time limit: 7 days[Note] (thereafter 36 hours prebaking required at 125°C), Preliminary heat temperature: 120°C Max. (Package surface temperature) | WS60-367-1 |
| Partial heating | Pin temperature: 300°C Max., Duration: 3 sec. Max. (per device side) | — |

**Note**  For the storage period after dry-pack decapsulation, storage conditions are Max. 25°C, 65% RH.

**Caution    Use of more than one soldering method should be avoided (except for partial heating).**

**(2)  µPD70732GC-25-9EV: 120-pin plastic TQFP (Fine pitch) (14 x 14 mm)**                    ★

| Soldering Method | Soldering Conditions | Recommended Condition Symbol |
|---|---|---|
| Infrared reflow | Package peak temperature: 235°C, Duration: 30 sec. Max. (at 210°C or above), Number of times: Twice Max., Time limit: 7 days**Note** (thereafter 10 hours prebaking required at 125°C) | IR35-107-2 |
| VPS | Package peak temperature: 215°C, Duration: 40 sec. Max. (at 200°C or above), Number of times: Twice Max., Time limit: 7 days**Note** (thereafter 10 hours prebaking required at 125°C) | VP15-107-2 |
| Partial heating | Pin temperature: 300°C Max., Duration: 3 sec. Max. (per device side) | — |

**Note**    For the storage period after dry-pack decapsulation, storage conditions are Max. 25°C, 65% RH.

**Caution    Use of more than one soldering method should be avoided (except for partial heating).**

**Table 12-2.  Insertion Type Soldering Conditions**

**µPD70732R-25: 176-pin ceramic PGA (Seam weld)**

| Soldering Method | Soldering Conditions |
|---|---|
| Wave soldering (Pin only) | Solder bath temperature: 260°C Max., Duration: 10 sec. Max. |
| Partial heating | Pin temperature: 300°C Max., Duration: 3 sec. Max. (per one pin) |

**Caution    Apply wave soldering only to the pins and be careful not to bring solder into direct contact with the package.**

---

## 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.**

# Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability

- Ordering information

- Product release schedule

- Availability of related technical literature

- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)

- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

**NEC Electronics Inc. (U.S.)**
Mountain View, California
Tel: 800-366-9782
Fax: 800-729-9288

**NEC Electronics (Germany) GmbH**
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

**NEC Electronics (UK) Ltd.**
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

**NEC Electronics Italiana s.r.1.**
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

**NEC Electronics (Germany) GmbH**
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

**NEC Electronics (France) S.A.**
France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

**NEC Electronics (France) S.A.**
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

**NEC Electronics (Germany) GmbH**
Scandinavia Office
Taeby Sweden
Tel: 8-63 80 820
Fax: 8-63 80 388

**NEC Electronics Hong Kong Ltd.**
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

**NEC Electronics Hong Kong Ltd.**
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

**NEC Electronics Singapore Pte. Ltd.**
United Square, Singapore 1130
Tel: 253-8311
Fax: 250-3583

**NEC Electronics Taiwan Ltd.**
Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

**NEC do Brasil S.A.**
Sao Paulo-SP, Brasil
Tel: 011-889-1680
Fax: 011-889-1689

**J96. 3**

**Reference**: Electrical Characteristics for Microcomputer (IEI-601)

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

**V805, V810, and V810 Family are trademarks of NEC Corporation.**

M4 96.5