To all our customers

Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

EOL announced

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

M37735MHBXXXFP

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37735MHBXXXFP is a single-chip microcomputer using the 7700 Family core. This single-chip microcomputer has a CPU and a bus interface unit. The CPU is a 16-bit parallel processor that can be an 8-bit parallel processor, and the bus interface unit enhances the memory access efficiency to execute instructions fast. This microcomputer also includes a 32 kHz oscillation circuit, in addition to the ROM, RAM, multiple-function timers, serial I/O, A-D converter, and so on.

FEATURES

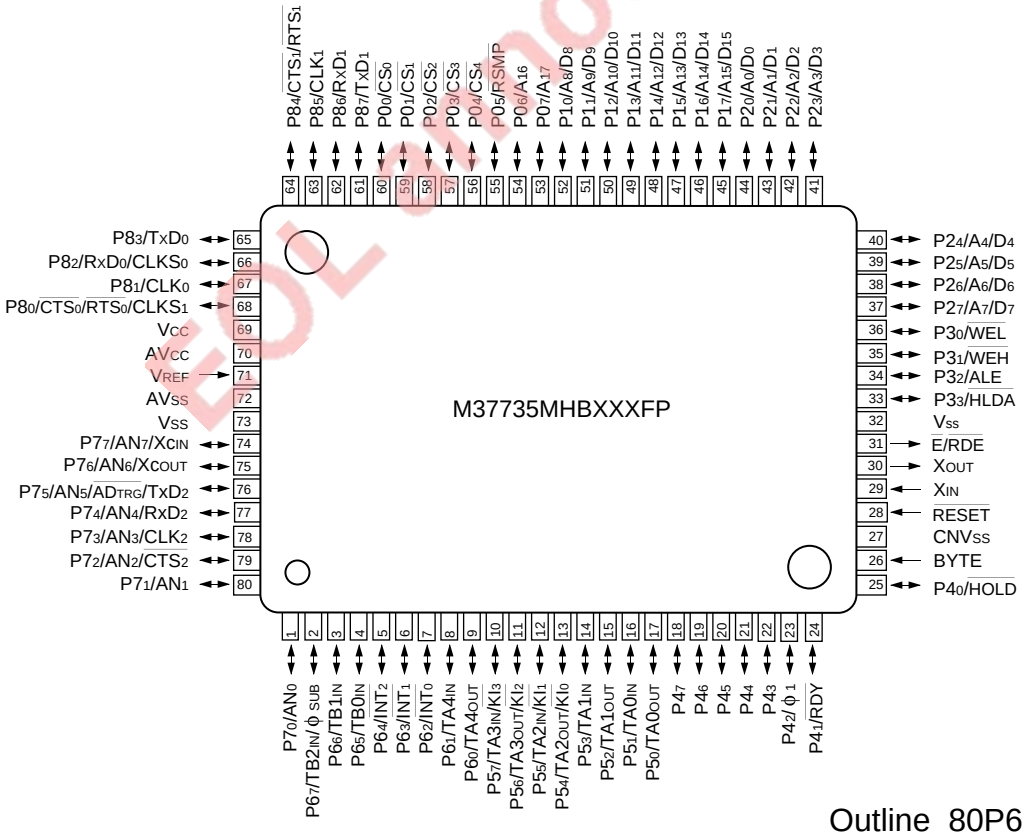
- Number of basic instructions 103
- Memory size ROM ..... 124 Kbytes
  RAM ..... 3968 bytes
- Instruction execution time
  The fastest instruction at 25 MHz frequency ..... 160 ns
- Single power supply ..... 5 V ± 10%
- Low power dissipation (at 25 MHz frequency)
  .....47.5 mW (Typ.)
- Interrupts ..... 19 types, 7 levels
- Multiple-function 16-bit timer ..... 5 + 3
- Serial I/O (UART or clock synchronous) 3
- 10-bit A-D converter ..... 8-channel inputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P0, P1, P2, P3, P4, P5, P6, P7, P8) 68
- Clock generating circuit ..... 2 circuits built-in

APPLICATION

Control devices for general commercial equipment such as office automation, office equipment, and so on.
Control devices for general industrial equipment such as communication equipment, and so on.

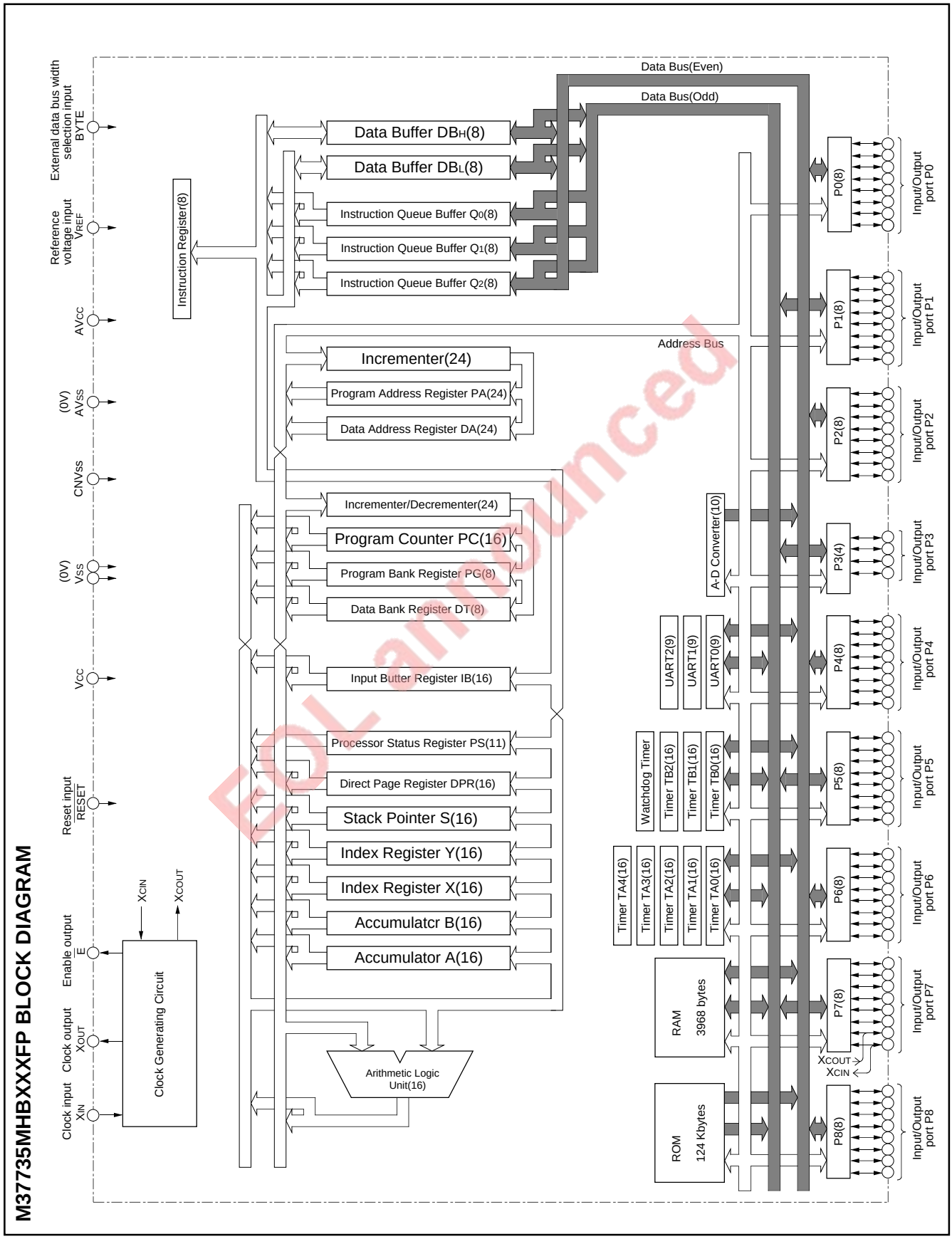
EOL announced

PIN CONFIGURATION (TOP VIEW)



Outline 80P6N-A

M37735MHBXXXFP BLOCK DIAGRAM



EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

FUNCTIONS OF M37735MHBXXXFP

Parameter		Functions
Number of basic instructions		103
Instruction execution time		160 ns (the fastest instruction at external clock 25 MHz frequency)
Memory size	ROM	124 Kbytes
	RAM	3968 bytes
Input/Output ports	P0 – P2, P4 – P8	8-bit × 8
	P3	4-bit × 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit × 5
	TB0, TB1, TB2	16-bit × 3
Serial I/O		(UART or clock synchronous serial I/O) × 3
A-D converter		10-bit × 1 (8 channels)
Watchdog timer		12-bit × 1
Interrupts		3 external types, 16 internal types Each interrupt can be set to the priority level (0 – 7.)
Clock generating circuit		2 circuits built-in (externally connected to a ceramic resonator or a quartz-crystal oscillator)
Supply voltage		5 V ± 10%
Power dissipation		47.5 mW (at external clock 25 MHz frequency)
Input/Output characteristic	Input/Output voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 1 Mbytes
Operating temperature range		–20 to 85 °C
Device structure		CMOS high-performance silicon gate process
Package		80-pin plastic molded QFP (80P6N-A)

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

PIN DESCRIPTION

Pin	Name	Input/Output	Functions
Vcc, Vss	Power source		Apply 5 V ± 10% to Vcc and 0 V to Vss.
CNVss	CNVss input	Input	This pin controls the processor mode. Connect to Vss for the single-chip mode and the memory expansion mode, and to Vcc for the microprocessor mode.
$\overline{RESET}$	Reset input	Input	When "L" level is applied to this pin, the microcomputer enters the reset state.
X_{IN}	Clock input	Input	These are pins of main-clock generating circuit. Connect a ceramic resonator or a quartz-crystal oscillator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin, and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	This pin functions as the enable signal output pin which indicates the access status in the internal bus. In the memory expansion mode or the microprocessor mode, this pin functions as the $\overline{RDE}$ signal output pin.
BYTE	External data bus width selection input	Input	In the memory expansion mode or the microprocessor mode, this pin determines whether the external data bus has an 8-bit width or a 16-bit width. The data bus has a 16-bit width when "L" signal is input and an 8-bit width when "H" signal is input.
AVcc, AVss	Analog power source input		Power source input pin for the A-D converter. Externally connect AVcc to Vcc and AVss to Vss.
V_{REF}	Reference voltage input	Input	This is reference voltage input pin for the A-D converter.
$P0_0$ – $P0_7$	I/O port P0	I/O	In the single-chip mode, port P0 becomes an 8-bit I/O port. An I/O direction register is available so that each pin can be programmed for input or output. These ports are in the input mode when reset. In the memory expansion mode or the microprocessor mode, these pins output $\overline{CS_0}$ – $\overline{CS_4}$, $\overline{RSMP}$ signals, and address (A_{16}, A_{17}).
$P1_0$ – $P1_7$	I/O port P1	I/O	In the single-chip mode, these pins have the same functions as port P0. When the BYTE pin is set to "L" in the memory expansion mode or the microprocessor mode and external data bus has a 16-bit width, high-order data (D_8 – D_{15}) is input/output or an address (A_8 – A_{15}) is output. When the BYTE pin is "H" and an external data bus has an 8-bit width, only address (A_8 – A_{15}) is output.
$P2_0$ – $P2_7$	I/O port P2	I/O	In the single-chip mode, these pins have the same functions as port P0. In the memory expansion mode or the microprocessor mode, low-order data (D_0 – D_7) is input/output or an address (A_0 – A_7) is output .
$P3_0$ – $P3_3$	I/O port P3	I/O	In the single-chip mode, these pins have the same function as port P0. In the memory expansion mode or the microprocessor mode, $\overline{WEL}$, $\overline{WEH}$, ALE, and $\overline{HLDA}$ signals are output.
$P4_0$ – $P4_7$	I/O port P4	I/O	In the single-chip mode, these pins have the same functions as port P0. In the memory expansion mode or the microprocessor mode, $P4_0$, $P4_1$ and $P4_2$ become $\overline{HOLD}$ and $\overline{RDY}$ input pins, and a clock ϕ_1 output pin, respectively. Functions of the other pins are the same as in the single-chip mode. However, in the memory expansion mode, $P4_2$ can be selected as an I/O port.
$P5_0$ – $P5_7$	I/O port P5	I/O	In addition to having the same functions as port P0 in the single-chip mode, these pins also function as I/O pins for timers A0 to A3 and input pins for key input interrupt input ($\overline{KI_0}$ – $\overline{KI_3}$).
$P6_0$ – $P6_7$	I/O port P6	I/O	In addition to having the same functions as port P0 in the single-chip mode, these pins also function as I/O pins for timer A4, input pins for external interrupt input ($\overline{INT_0}$ – $\overline{INT_2}$) and input pins for timers B0 to B2. $P6_7$ also functions as a sub-clock ϕ_{SUB} output pin.
$P7_0$ – $P7_7$	I/O port P7	I/O	In addition to having the same functions as port P0 in the single-chip mode, these pins function as input pins for A-D converter. $P7_2$ to $P7_5$ also function as I/O pins for UART2. Additionally, $P7_6$ and $P7_7$ have the function as the output pin (X_{COUT}) and the input pin (X_{CIN}) of the sub-clock (32 kHz) oscillation circuit, respectively. When $P7_6$ and $P7_7$ are used as the X_{COUT} and X_{CIN} pins, connect a resonator or an oscillator between the both.
$P8_0$ – $P8_7$	I/O port P8	I/O	In addition to having the same functions as port P0 in the single-chip mode, these pins also function as I/O pins for UART 0 and UART 1.

Engineering Advanced

BASIC FUNCTION BLOCKS

The M37735MHBXXXFP contains the following peripheral devices on a single chip: ROM, RAM, CPU, bus interface unit, timers, serial I/O, A-D converter, I/O ports, clock generating circuit and others. Each of these devices is described below.

MEMORY

The memory map is shown in Figure 1. The address space has a capacity of 16 Mbytes and is allocated to addresses from 0_{16} to $FFFFFF_{16}$. The address space is divided by 64-Kbyte unit called bank. The banks are numbered from 0_{16} to FF_{16}.

However, banks 10_{16} – FF_{16} of the 7735 group cannot be accessed. Built-in ROM, RAM and control registers for internal peripheral devices are assigned to banks 0_{16} and 1_{16}.

The 124-Kbyte area from addresses 1000_{16} to $1FFFF_{16}$ is the built-in ROM. Addresses $FFD6_{16}$ to $FFFF_{16}$ are the RESET and interrupt vector addresses and contain the interrupt vectors. Refer to the section on interrupts for details.

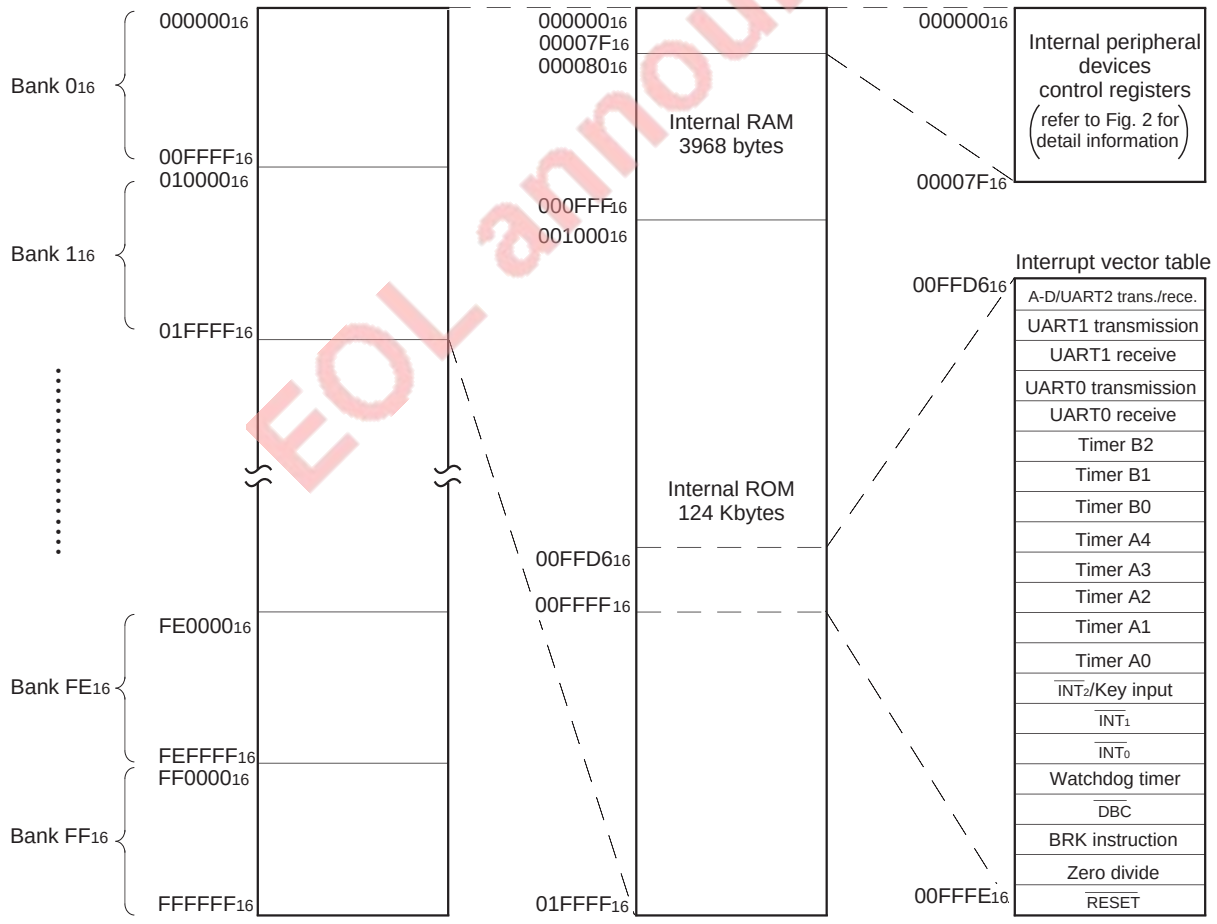
The 3968-byte area allocated to addresses from 80_{16} to FFF_{16} is the built-in RAM. In addition to storing data, the RAM is used as stack during a subroutine call or interrupts.

Peripheral devices such as I/O ports, A-D converter, serial I/O, timer, and interrupt control registers are allocated to addresses from 0_{16} to $7F_{16}$.

Additionally, the internal ROM and RAM area can be modified by software. Refer to the section on ROM area modification function for details.

A 256-byte direct page area can be allocated anywhere in bank 0_{16} by using the direct page register (DPR). In the direct page addressing mode, the memory in the direct page area can be accessed with two words. Hence program steps can be reduced.

EOL announced



Notes 1. Internal ROM and RAM area can be modified. (Refer to the section on ROM area modification function.)
2. Banks 10_{16} – FF_{16} cannot be accessed in the 7735 group.

Fig. 1 Memory map

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Address (Hexadecimal notation)	
000000	
000001	
000002	Port P0 register
000003	Port P1 register
000004	Port P0 direction register
000005	Port P1 direction register
000006	Port P2 register
000007	Port P3 register
000008	Port P2 direction register
000009	Port P3 direction register
00000A	Port P4 register
00000B	Port P5 register
00000C	Port P4 direction register
00000D	Port P5 direction register
00000E	Port P6 register
00000F	Port P7 register
000010	Port P6 direction register
000011	Port P7 direction register
000012	Port P8 register
000013	
000014	Port P8 direction register
000015	
000016	
000017	
000018	
000019	
00001A	
00001B	
00001C	Reserved area (Note)
00001D	Reserved area (Note)
00001E	A-D control register 0
00001F	A-D control register 1
000020	A-D register 0
000021	
000022	A-D register 1
000023	
000024	A-D register 2
000025	
000026	A-D register 3
000027	
000028	A-D register 4
000029	
00002A	A-D register 5
00002B	
00002C	A-D register 6
00002D	
00002E	A-D register 7
00002F	
000030	UART 0 transmit/receive mode register
000031	UART 0 baud rate register (BRG0)
000032	UART 0 transmission buffer register
000033	
000034	UART 0 transmit/receive control register 0
000035	UART 0 transmit/receive control register 1
000036	UART 0 receive buffer register
000037	
000038	UART 1 transmit/receive mode register
000039	UART 1 baud rate register (BRG1)
00003A	UART 1 transmission buffer register
00003B	
00003C	UART 1 transmit/receive control register 0
00003D	UART 1 transmit/receive control register 1
00003E	UART 1 receive buffer register
00003F	

Address (Hexadecimal notation)	
000040	Count start flag
000041	
000042	One-shot start flag
000043	
000044	Up-down flag
000045	
000046	Timer A0 register
000047	
000048	Timer A1 register
000049	
00004A	Timer A2 register
00004B	
00004C	Timer A3 register
00004D	
00004E	Timer A4 register
00004F	
000050	Timer B0 register
000051	
000052	Timer B1 register
000053	
000054	Timer B2 register
000055	
000056	Timer A0 mode register
000057	Timer A1 mode register
000058	Timer A2 mode register
000059	Timer A3 mode register
00005A	Timer A4 mode register
00005B	Timer B0 mode register
00005C	Timer B1 mode register
00005D	Timer B2 mode register
00005E	Processor mode register 0
00005F	Processor mode register 1
000060	Watchdog timer register
000061	Watchdog timer frequency selection flag
000062	Reserved area (Note)
000063	Memory allocation control register
000064	UART 2 transmit/receive mode register
000065	UART 2 baud rate register (BRG2)
000066	UART 2 transmission buffer register
000067	
000068	UART 2 transmit/receive control register 0
000069	UART 2 transmit/receive control register 1
00006A	UART 2 receive buffer register
00006B	
00006C	Oscillation circuit control register 0
00006D	Port function control register
00006E	Serial transmit control register
00006F	Oscillation circuit control register 1
000070	A-D/UART 2 trans./rece. interrupt control register
000071	UART 0 transmission interrupt control register
000072	UART 0 receive interrupt control register
000073	UART 1 transmission interrupt control register
000074	UART 1 receive interrupt control register
000075	Timer A0 interrupt control register
000076	Timer A1 interrupt control register
000077	Timer A2 interrupt control register
000078	Timer A3 interrupt control register
000079	Timer A4 interrupt control register
00007A	Timer B0 interrupt control register
00007B	Timer B1 interrupt control register
00007C	Timer B2 interrupt control register
00007D	$\overline{INT_0}$ interrupt control register
00007E	$\overline{INT_1}$ interrupt control register
00007F	$\overline{INT_2}$/Key input interrupt control register

Note. Do not write to this address.

EOL announced

Fig. 2 Location of internal peripheral devices and interrupt control registers

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

CENTRAL PROCESSING UNIT (CPU)

The CPU has ten registers and is shown in Figure 3. Each of these registers is described below.

ACCUMULATOR A (A)

Accumulator A is the main register of the microcomputer. It consists of 16 bits and the low-order 8 bits can be used separately. The data length flag (m) determines whether the register is used as a 16-bit register or as an 8-bit register. It is used as a 16-bit register when flag m is “0” and as an 8-bit register when flag m is “1”. Flag m is a part of the processor status register (PS) which is described later.

Data operations such as arithmetic operation, data transfer, input/output, etc., are executed mainly through the accumulator A.

ACCUMULATOR B (B)

Accumulator B has the same functions as accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A.

INDEX REGISTER X (X)

Index register X consists of 16 bits and the low-order 8 bits can be used separately. The index register length flag (x) determines whether the register is used as a 16-bit register or as an 8-bit register. It is used as a 16-bit register when flag x is “0” and as an 8-bit register when flag x is “1”. Flag x is a part of the processor status register (PS) which is described later.

In an index addressing mode where register X is used as the index register, the contents of this address is added to obtain the real address.

Also, when executing a block transfer instruction (MVP, MVN), the contents of index register X indicates the low-order 16 bits of the source data address. The third byte of the MVP or MVN is the high-order 8 bits of the source data address.

INDEX REGISTER Y (Y)

Index register Y consists of 16 bits and the low-order 8 bits can be used separately. The index register length flag (x) determines whether the register is used as a 16-bit register or as an 8-bit register. It is used as a 16-bit register when flag x is “0” and as an 8-bit register when flag x is “1”. Flag x is a part of the processor status register (PS) which is described later.

In an index addressing mode where register Y is used as the index register, the contents of this address is added to obtain the real address.

Also, when executing a block transfer instruction (MVP, MVN), the contents of index register Y indicates the low-order 16 bits of the destination data address. The second byte of the MVP or MVN is the high-order 8 bits of the destination data address.

EOL announced

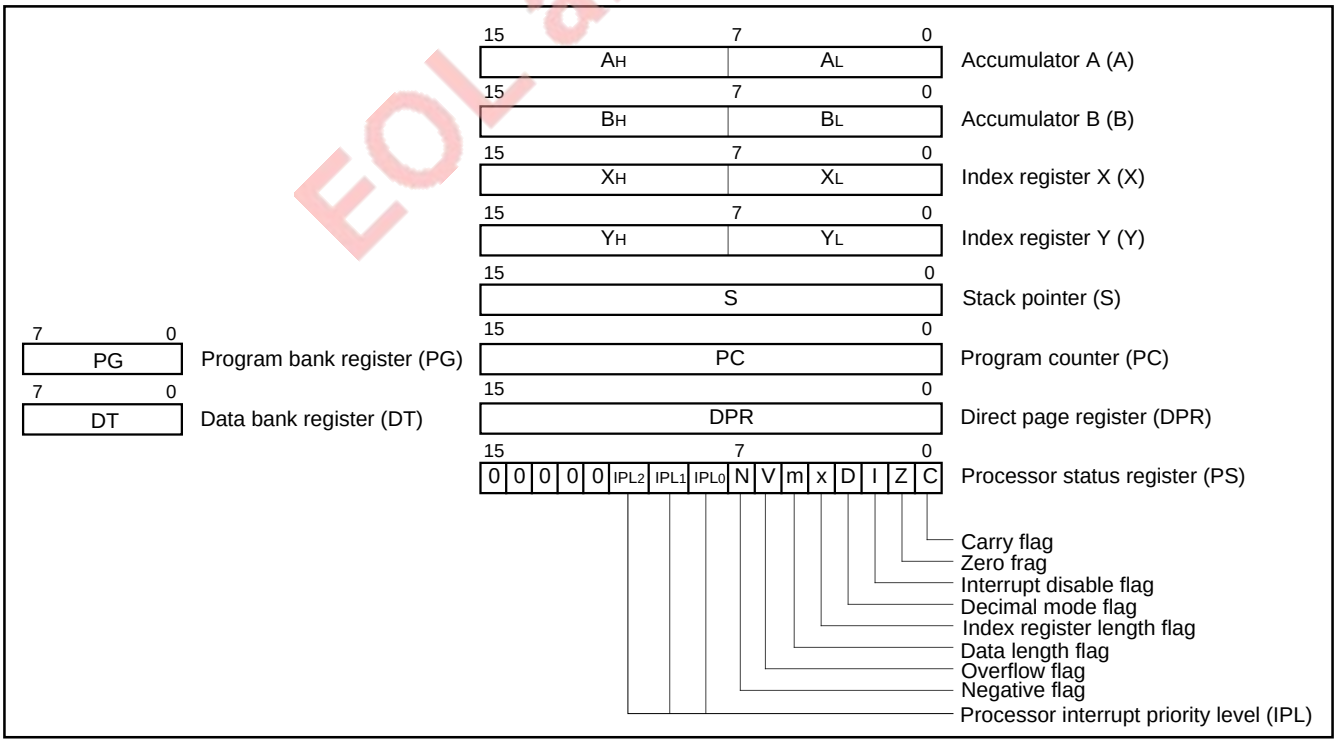


Fig. 3 Register structure

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

STACK POINTER (S)

Stack pointer (S) is a 16-bit register. It is used during a subroutine call or interrupts. It is also used during stack, stack pointer relative, or stack pointer relative indirect indexed Y addressing modes.

PROGRAM COUNTER (PC)

Program counter (PC) is a 16-bit counter that indicates the low-order 16 bits of the next program memory address to be executed. There is a bus interface unit between the program memory and the CPU, so that the program memory is accessed through the bus interface unit. This is described later.

PROGRAM BANK REGISTER (PG)

Program bank register (PG) is an 8-bit register that indicates the high-order 8 bits of the next program memory address to be executed. When a carry occurs by incrementing the contents of the program counter, the contents of the program bank register (PG) is incremented by 1. Also, when a carry or borrow occurs after adding or subtracting the offset value to or from the contents of the program counter (PC) by using a branch instruction, the contents of the program bank register (PG) is incremented or decremented by 1 so that programs can be written without worrying about bank boundaries.

DATA BANK REGISTER (DT)

Data bank register (DT) is an 8-bit register. With some addressing modes, a part of the data bank register (DT) is used to specify a memory address. The contents of data bank register (DT) is used as the high-order 8 bits of a 24-bit address. Addressing modes that use the data bank register (DT) to specify the address are direct indirect, direct indexed X indirect, direct indirect indexed Y, absolute, absolute bit, absolute indexed X, absolute indexed Y, absolute bit relative, and stack pointer relative indirect indexed Y.

DIRECT PAGE REGISTER (DPR)

Direct page register (DPR) is a 16-bit register. Its contents is used as the base address of a 256-byte direct page area. The direct page area is allocated in bank 0_{16}, but when the contents of DPR is $FF01_{16}$ or more, the direct page area spans across bank 0_{16} and bank 1_{16}. All direct addressing modes use the contents of the direct page register (DPR) to generate the data address. If the low-order 8 bits' contents of the direct page register (DPR) is "00_{16}", the number of cycles required to generate an address is minimized. Hence the low-order 8 bits' contents of the direct page register (DPR) is usually set to "00_{16}".

PROCESSOR STATUS REGISTER (PS)

Processor status register (PS) is an 11-bit register. It consists of flags which indicate the result of operation and the processor interrupt priority level (IPL).

Branch operations can be performed by testing flags C, Z , V, and N. The details of each processor status register bit are described below.

1. Carry flag (C)

The carry flag contains the carry or borrow generated by the ALU after an arithmetic operation. This flag is also affected by shift or rotate instruction. This flag can be set or reset directly with the SEC, CLC instructions or with the SEP, CLP instructions.

2. Zero flag (Z)

This zero flag is set when the result of an arithmetic operation or data transfer is zero and reset when it is not. This flag can be set or reset directly with the SEP or CLP instruction.

3. Interrupt disable flag (I)

When the interrupt disable flag is "1", all interrupts except watchdog timer, $\overline{DBC}$, and software interrupt are disabled. This flag is automatically set to "1" when an interrupt is accepted. It can be set or reset directly with the SEI, CLI instructions or SEP and CLP instructions.

4. Decimal mode flag (D)

The decimal mode flag determines whether addition and subtraction are performed in the binary or the decimal system. Binary arithmetic is performed when this flag is "0". If it is "1", decimal arithmetic is performed with each word treated as the 2- or 4-digit number. Arithmetic operation is performed with 4-digit number when the data length flag (m) is "0" and with 2-digit number when it is "1". Decimal correction is automatically performed. (Decimal operation is possible only with the ADC and SBC instructions.) This flag can be set or reset with the SEP or CLP instruction.

5. Index register length flag (x)

The index register length flag determines whether index register X and index register Y are used as 16-bit registers or as 8-bit registers. The registers are used as 16-bit registers when flag x is "0" and as 8-bit registers when it is "1". This flag can be set or reset with the SEP or CLP instruction.

6. Data length flag (m)

The data length flag determines whether the data has a length of 16 bits or that of 8 bits. The 16-bit length is selected when flag m is "0" and the 8-bit length is selected when it is "1". This flag can be set or reset with the SEM, CLM instructions or with the SEP, CLP instructions.

EOL announced

7. Overflow flag (V)

The overflow flag is effective only when addition or subtraction is performed with treating a word as a signed binary number. When the data length flag (m) is “0”, the overflow flag is set if the result of addition or subtraction is outside the range between – 32768 and +32767. When the data length flag (m) is “1”, the overflow flag is set if the result of addition or subtraction is outside the range between –128 and +127. It is reset in the other cases. The overflow flag can also be set or reset directly with the SEP or CLV, CLP instructions.

8. Negative flag (N)

The negative flag is set when the result of arithmetic operation or data transfer is negative (If data length flag (m) is “0”, data bit 15 is “1”. If data length flag (m) is “1”, data bit 7 is “1”.) It is reset in the other cases. It can also be set or reset with the SEP or CLP instructions.

9. Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) consists of 3 bits and determines the processor interrupt priority level (0 to 7). Interrupt is enabled when the interrupt priority level of the device requesting interrupt (the priority can be set using the interrupt control register) is higher than the processor interrupt priority level. When interrupt is enabled, the current processor interrupt priority level is saved in a stack and the processor interrupt priority level is replaced by the interrupt priority level of the device requesting the interrupt. Refer to the section on interrupts for more details.

BUS INTERFACE UNIT

The CPU operates on an internal clock ϕ ’s frequency. Internal clock ϕ ’s frequency is twice the bus cycle frequency. In order to speed up processing, a bus interface unit is used to pre-fetch instructions when the data bus is idle. The bus interface unit synchronizes the CPU and the bus and pre-fetches instructions. Figure 4 shows the relationship between the CPU and the bus interface unit. The bus interface unit has a program address register, a 3-byte instruction queue buffer, a data address register, and a 2-byte data buffer.
The bus interface unit obtains an instruction code from the memory and stores it in the instruction queue buffer, obtains data from the memory and stores it in the data buffer, or writes the data from the data buffer to the memory.

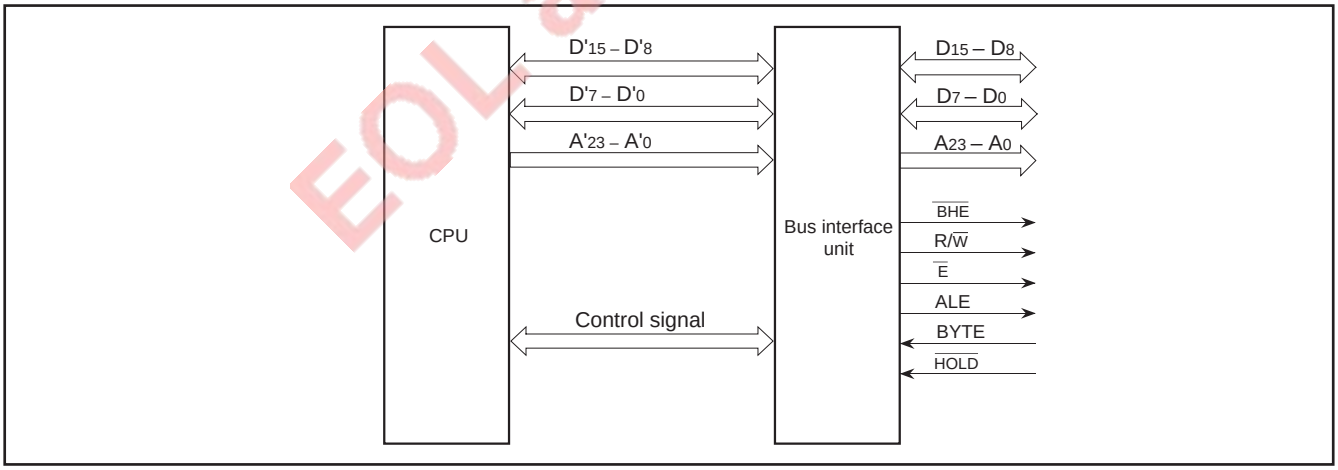


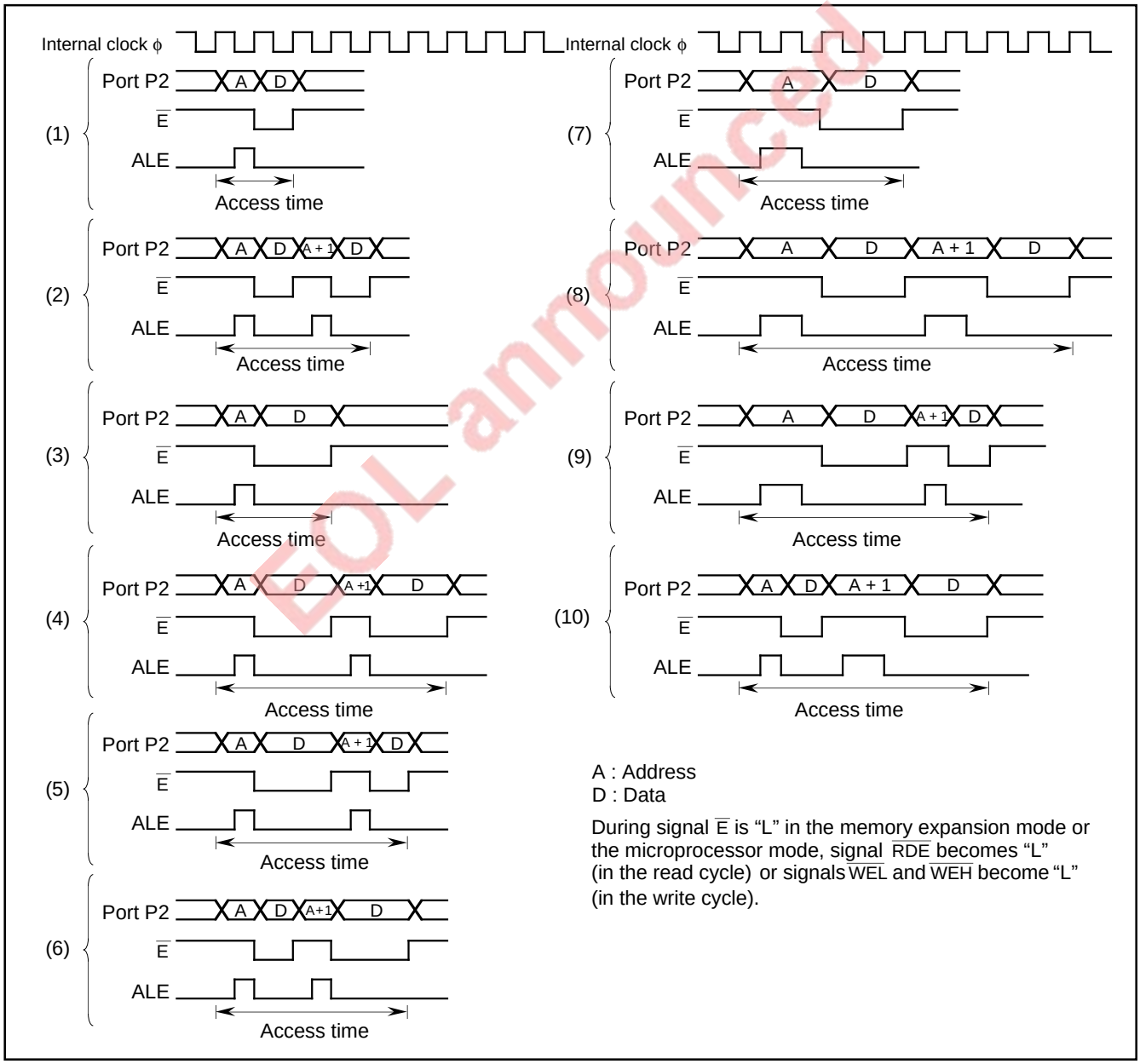
Fig. 4 Relationship between the CPU and the bus interface unit

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

The bus interface unit operates using one of the waveforms (1) to (10) shown in Figure 5. The standard waveforms are (1) and (2).
The ALE signal is used to latch only the address signal from the multiplexed signal containing data and address.
The $\overline{E}$ signal becomes "L" when the bus interface unit reads an instruction code or data from the memory or when it writes data to the memory. Whether to perform read or write is controlled by the R/$\overline{W}$ signal. The $\overline{E}$ (except for that in the single-chip mode) and R/$\overline{W}$ signals are not directly output to the outside. In the memory expansion mode or the microprocessor mode, read signal $\overline{RDE}$ and write signals $\overline{WEL}$, $\overline{WEH}$ are output, instead of $\overline{E}$ and R/$\overline{W}$, to the outside of the chip. During signal $\overline{E}$ is "L", signal $\overline{RDE}$ becomes "L" (in the read cycle) or signals $\overline{WEL}$ and $\overline{WEH}$ become "L" (in the write cycle).

Waveform (1) in Figure 5 is used to access a single byte or two bytes simultaneously. To read or write two bytes simultaneously, the first address accessed must be even. Furthermore, when accessing an external memory area in the memory expansion mode or the microprocessor mode, set the bus width selection input pin (BYTE) to "L" (external data bus has a width of 16 bits). The data bus in the internal memory area is always treated as the 16-bit bus independent of BYTE.



EOL announced

Fig. 5 Bus access timing

When performing 16-bit data read or write, waveform (2) is used to access each byte one by one if the conditions for simultaneously accessing two bytes are not satisfied. However, when prefetching the instruction code, if the address of the instruction code is odd, waveform (1) is used, and only one byte is read in the instruction queue buffer.

Access to the even/odd address is controlled by signals $\overline{BHE}$ and A_0. Signal $\overline{BHE}$ is not directly output to the outside, but write signals ($\overline{WEL}$, $\overline{WEH}$) are generated corresponding to the accessed address (even or odd).

Bit 2 of processor mode register 0 (address $5E_{16}$) is the wait bit. When the external memory area is accessed in the memory expansion mode or the microprocessor mode with this bit set to “0”, the width of the $\overline{E}$ signal is extended and access time can be extended.

There are two ways to extend the access time and they are selected with bit 0 of the processor mode register 1 (address $5F_{16}$).

When this bit is set to “1”, the “L” width of the $\overline{E}$ signal in (1) becomes twice as long as in (3) and the access time becomes 1.5 times (wait 1). When this bit is set to “0”, the ALE signal and $\overline{E}$ signal in (1) are extended as in (7) and the access time is doubled (wait 0).

However, these signals are not extended when accessing the internal memory area.

When the wait bit is set to “1”, these signals are not extended when accessing any memory area regardless of the bit 0 of the processor mode register 1.

Waveforms (4), (5), and (6) show the entire waveform, first half, and last half respectively of waveform (2) for wait 1.

Waveforms (8), (9), and (10) show the entire waveform, first half, and last half respectively of waveform (2) for wait 0.

Instruction code read, data read, and data write are described below. Instruction code read will be described first.

The CPU obtains instruction codes from the instruction queue buffer and executes them. The CPU notifies the bus interface unit that it is requesting an instruction code during an instruction code request cycle. If the requested instruction code is not yet stored in the instruction queue buffer, the bus interface unit halts the CPU until more instructions than requested is stored in the instruction queue buffer. Even if there is no instruction code request from the CPU, the bus interface unit reads instruction codes from the memory and stores them in the instruction queue buffer when the instruction queue buffer is empty or when only one instruction code is stored and the bus is idle on the next cycle. This is referred to as instruction pre-fetching.

Normally , when reading an instruction code from the memory, if the accessed address is even, the next odd address is read together with the instruction code and stored in the instruction queue buffer.

However, in the memory expansion mode or the microprocessor mode, only one byte is read and stored in the instruction queue buffer if the following conditions are satisfied.

- The address to be read is in the external memory area when the external data bus has an 8-bit width (BYTE = “H”).
- The address to be read is odd.

Therefore, waveform (1), (3) or (7) in Figure 5 is used for instruction code read. Data read and write are described below.

The CPU notifies the bus interface unit when performing data read or write. At this time, the bus interface unit halts the CPU if the bus interface unit is already using the bus or if there is a request with higher priority. When data read or write is enabled, the bus interface unit uses one of the waveforms from (1) to (10) in Figure 5 to perform the operation.

During data read, the CPU waits until the entire data is stored in the data buffer. The bus interface unit sends the address received from the CPU to the address bus. Then it reads the memory when the $\overline{E}$ signal is “L” and stores the result in the data buffer.

During data write, the CPU writes the data in the data buffer and the bus interface unit writes it to the memory . Therefore, the CPU can proceed to the next step without waiting for write completed. The bus interface unit sends the address received from the CPU to the address bus. Then when the $\overline{E}$ signal is “L”, the bus interface unit sends the data in the data buffer to the data bus and writes it to the memory.

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

INTERRUPTS

Table 1 shows the interrupt sources and the corresponding interrupt vector addresses. Reset is also treated as a source of interrupt and is described in this section.

$\overline{DBC}$ is an interrupt used only for debugging.

Interrupts other than reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction all have their respective interrupt control registers. Table 2 shows the addresses of the interrupt control registers and Figure 6 shows the bit configuration of the interrupt control register.

The interrupt request bit is automatically cleared by hardware during reset or when processing an interrupt. Also, interrupt request bits other than $\overline{DBC}$ and watchdog timer can be cleared by software.

$\overline{INT_0}$ to $\overline{INT_2}$ are external interrupts, and whether to cause an interrupt at the input level (level sense) or at the edge (edge sense) can be selected with the level sense/edge sense selection bit. Furthermore, the polarity of the interrupt input can be selected with a polarity selection bit.

In the $\overline{INT_2}$/Key input interrupt, whether to input an interrupt request from the $\overline{INT_2}$ pin or the $\overline{KI_0}$ – $\overline{KI_3}$ pins can be selected by bit 7 of the port function control register (refer to Figure 11).

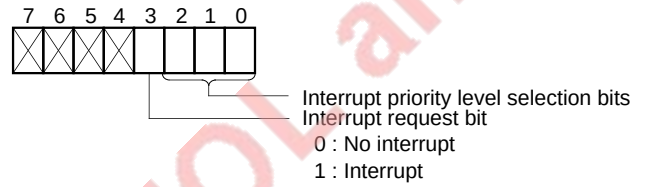
Timer and UART interrupts are described in the respective section.

The priority of interrupts when multiple interrupts are caused simultaneously is partially fixed by hardware, but it can also be adjusted by software as shown in Figure 7. The hardware priority is fixed as follows:

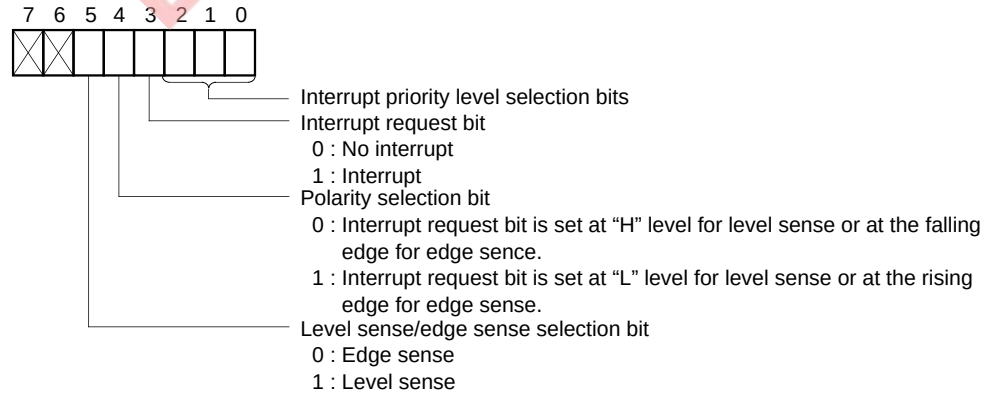
reset > $\overline{DBC}$ > watchdog timer > other interrupts

Table 1. Interrupt sources and the interrupt vector addresses

Interrupts	Vector addresses
A-D/UART2 trans./rece.	00FFD6$_{16}$ 00FFD7$_{16}$
UART1 transmit	00FFD8$_{16}$ 00FFD9$_{16}$
UART1 receive	00FFDA$_{16}$ 00FFDB$_{16}$
UART0 transmit	00FFDC$_{16}$ 00FFDD$_{16}$
UART0 receive	00FFDE$_{16}$ 00FFDF$_{16}$
Timer B2	00FFE0$_{16}$ 00FFE1$_{16}$
Timer B1	00FFE2$_{16}$ 00FFE3$_{16}$
Timer B0	00FFE4$_{16}$ 00FFE5$_{16}$
Timer A4	00FFE6$_{16}$ 00FFE7$_{16}$
Timer A3	00FFE8$_{16}$ 00FFE9$_{16}$
Timer A2	00FFEA$_{16}$ 00FFEB$_{16}$
Timer A1	00FFEC$_{16}$ 00FFED$_{16}$
Timer A0	00FFEE$_{16}$ 00FFEF$_{16}$
$\overline{INT_2}$/Key input	00FFF0$_{16}$ 00FFF1$_{16}$
$\overline{INT_1}$	00FFF2$_{16}$ 00FFF3$_{16}$
$\overline{INT_0}$	00FFF4$_{16}$ 00FFF5$_{16}$
Watchdog timer	00FFF6$_{16}$ 00FFF7$_{16}$
$\overline{DBC}$ (unusable)	00FFF8$_{16}$ 00FFF9$_{16}$
BRK instruction	00FFFA$_{16}$ 00FFFB$_{16}$
Zero divide	00FFFC$_{16}$ 00FFFD$_{16}$
Reset	00FFFE$_{16}$ 00FFFF$_{16}$



Interrupt control register configuration for timers A0 to A4, timers B0 to B2, UART0, UART1 and A-D/UART2 trans./rece.



Interrupt control register configuration for $\overline{INT_0}$ to $\overline{INT_2}$/Key input.

Fig. 6 Interrupt control register bit configuration

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Table 2. Addresses of interrupt control registers

Interrupt control registers	addresses
A-D/UART2 trans./rece. interrput control register	000070₁₆
UART0 transmit interrput control register	000071₁₆
UART0 receive interrput control register	000072₁₆
UART1 transmit interrput control register	000073₁₆
UART1 receive interrupt control register	000074₁₆
Timer A0 interrupt control register	000075₁₆
Timer A1 interrupt control register	000076₁₆
Timer A2 interrupt control register	000077₁₆
Timer A3 interrupt control register	000078₁₆
Timer A4 interrupt control register	000079₁₆
Timer B0 interrupt control register	00007A₁₆
Timer B1 interrupt control register	00007B₁₆
Timer B2 interrupt control register	00007C₁₆
$\overline{INT_0}$ interrupt control register	00007D₁₆
$\overline{INT_1}$ interrupt control register	00007E₁₆
$\overline{INT_2}$/Key input interrupt control register	00007F₁₆

Interrupts caused by a BRK instruction and when dividing by zero are software interrupts and are not included in this list.

Other interrupts previously mentioned are A-D converter, UART, Timer, INT interrupts. The priority of these interrupts can be changed by changing the interrupt priority level selection bits of the corresponding interrupt control register with software.

Figure 8 shows a diagram of the interrupt priority detection circuit. When an interrupt is caused, the each interrupt device compares its own priority with the priority from above and if its own priority is higher, then it sends the priority below and requests the interrupt. If the priorities are the same, the one above has priority.

This comparison is repeated to select the interrupt with the highest priority among the interrupts that are being requested. Finally the selected interrupt is compared with the processor interrupt priority level (IPL) contained in the processor status register (PS), and the request is accepted if it is higher than IPL and the interrupt disable flag (I) is "0". The request is not accepted if flag I is "1". The reset, $\overline{DBC}$, and watchdog timer interrupts are not affected by the interrupt disable flag (I).

When an interrupt is accepted, the contents of the processor status register (PS) is saved to the stack and the interrupt disable flag (I) is set to "1".

Furthermore, the interrupt request bit of the accepted interrupt is cleared to "0" and the processor interrupt priority level (IPL) in the processor status register (PS) is replaced by the priority level of the accepted interrupt.

Therefore, multiple interrupts are possible by resetting the interrupt disable flag (I) to "0" and enable further interrupts.

For reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 3.

Priority detection is performed by latching the interrupt request bit and interrupt priority level selection bits so that they do not change. They are sampled at the first half and latched at the last half of the operation code fetch cycle.

Because priority detection takes some time, no sampling pulse is generated for a certain interval even if it is the next operation code fetch cycle.

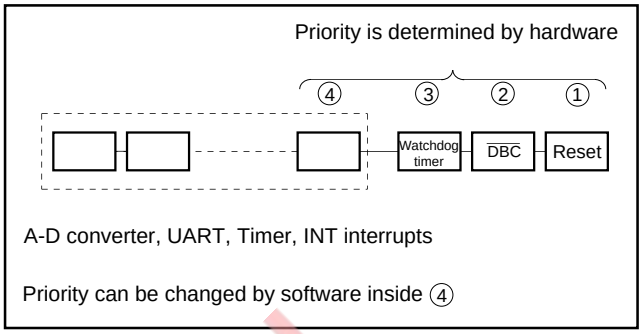


Fig. 7 Interrupt priority

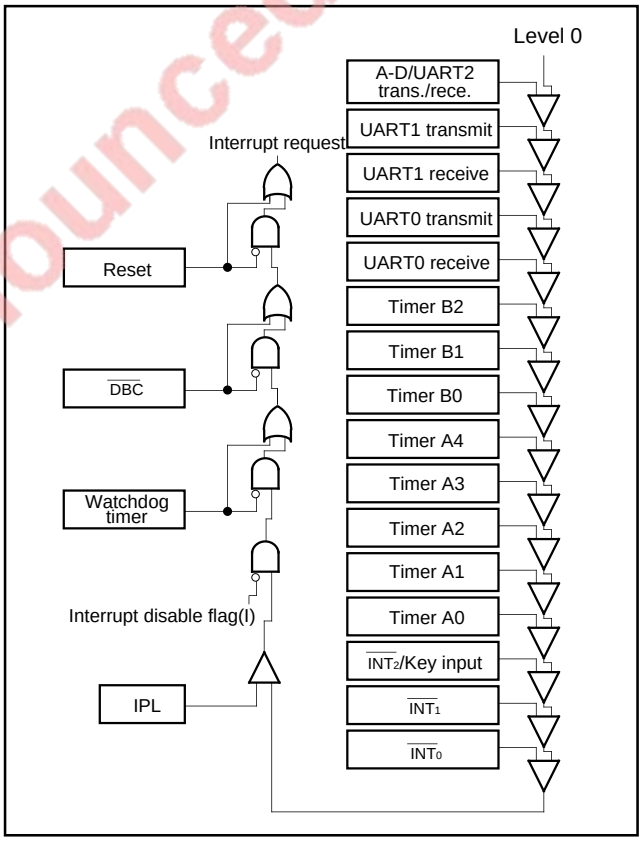


Fig. 8 Interrupt priority detection circuit

Renesas Technology Corp.

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

As shown in Figure 9, there are three different interrupt priority detection time from which one is selected by software. After the selected time has elapsed, the interrupt which has the highest priority is determined and is processed after the current instruction execution has been completed.

The time is selected with bits 4 and 5 of the processor mode register 0 (address $5E_{16}$) shown in Figure 10. Table 4 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register 0 is initialized to "00_{16}". Therefore, the longest time is selected. However, the shortest time should be selected by software.

Table 3. Value set in processor interrupt level (IPL) during an interrupt

Interrupt types	Setting value
Reset	0
$\overline{DBC}$	7
Watchdog timer	7
Zero divide	Not change value of IPL.
BRK instruction	Not change value of IPL.

Table 4. Relationship between interrupt priority detection time selection bits and number of cycles

Interrupt priority detection time selection bits		Number of cycles
Bit 5	Bit 4	
0	0	7 cycles of ϕ
0	1	4 cycles of ϕ
1	0	2 cycles of ϕ

ϕ : internal clock

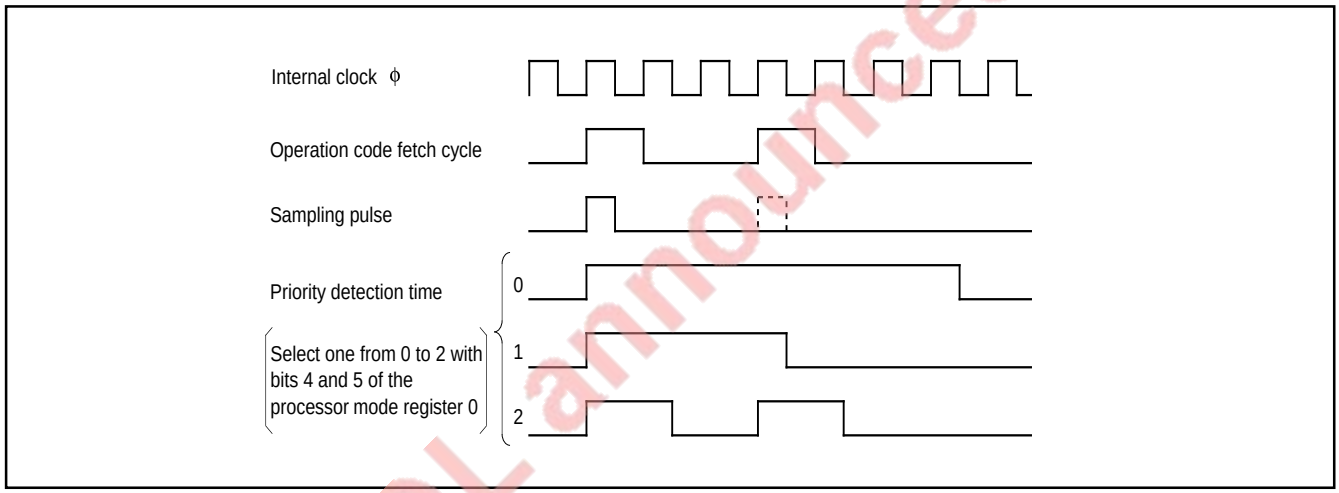


Fig. 9 Interrupt priority detection time

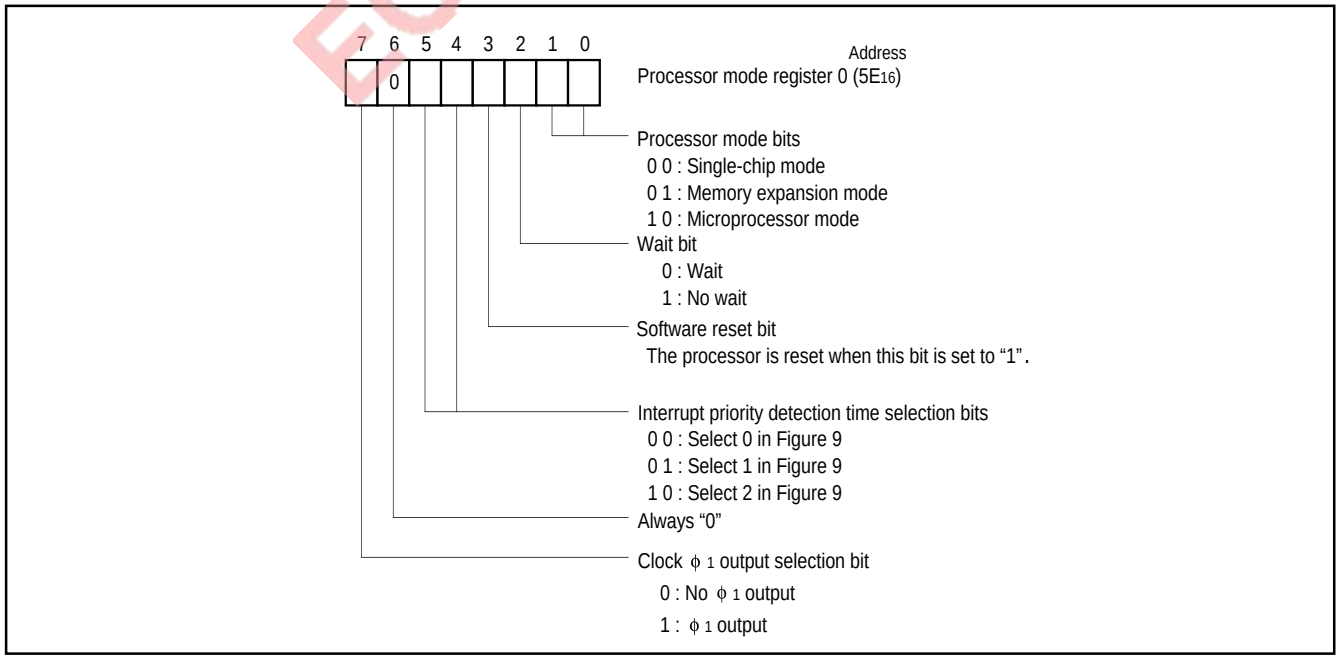


Fig. 10 Processor mode register 0 configuration

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

By setting the port function control register, the $\overline{INT_2}$/Key input interrupt function can be switched to the key input interrupt function which uses the $\overline{KI_0}$ to $\overline{KI_3}$ inputs. Figure 11 shows the bit configuration of the port function control register, and Figure 12 shows the $\overline{INT_2}$/Key input interrupt input circuit block diagram.

When the key input interrupt selection bit of the port function control register is "0", a signal is input from the $\overline{INT_2}$ pin to the $\overline{INT_2}$/Key input interrupt control circuit and the $\overline{INT_2}$ interrupt is normally performed.

When the key input interrupt selection bit is "1", signals input from the $\overline{KI_0}$ to $\overline{KI_3}$ pins are inverted, and then the logical sum of these signals is input to the $\overline{INT_2}$ interrupt control circuit. In this case, the external interrupt which uses the $\overline{KI_0}$ to $\overline{KI_3}$ pins is performed. (Pins $\overline{KI_0}$ to $\overline{KI_3}$ correspond to ports $P5_4$ to $P5_7$, respectively.) Additionally, by setting the port P6 pull-up selection bit 1 to "1", the $\overline{INT_2}$ input is added to that logical sum, so that the external interrupt which uses the inputs $\overline{KI_0}$ to $\overline{KI_3}$ and $\overline{INT_2}$ is performed. When using the key input interrupt, it is necessary to select the edge sense which uses the falling edge by setting the $\overline{INT_2}$/Key input interrupt control register. Because of this selection, a key input interrupt request occurs when "L" is input to one of the $\overline{KI_0}$ to $\overline{KI_3}$ and $\overline{INT_2}$ pins. The interrupt vector and the interrupt control register are common to the $\overline{INT_2}$ and key input interrupts.

Pull-up resistors (transistors) can be added to the $\overline{KI_0}$ to $\overline{KI_3}$ pins by setting "1" to the port P5 pull-up selection bit and "0" to the contents of the port $P5_i$ (i = 4 to 7) direction register. Similarly, a pull-up resistor can be added to the $\overline{INT_2}$ pin by setting "1" to the port P6 pull-up selection bit 1 and "0" to the content of the port $P6_4$ direction register. With the key input interrupt and the pull-up function, the key input circuit is easily composed.

EOL announced

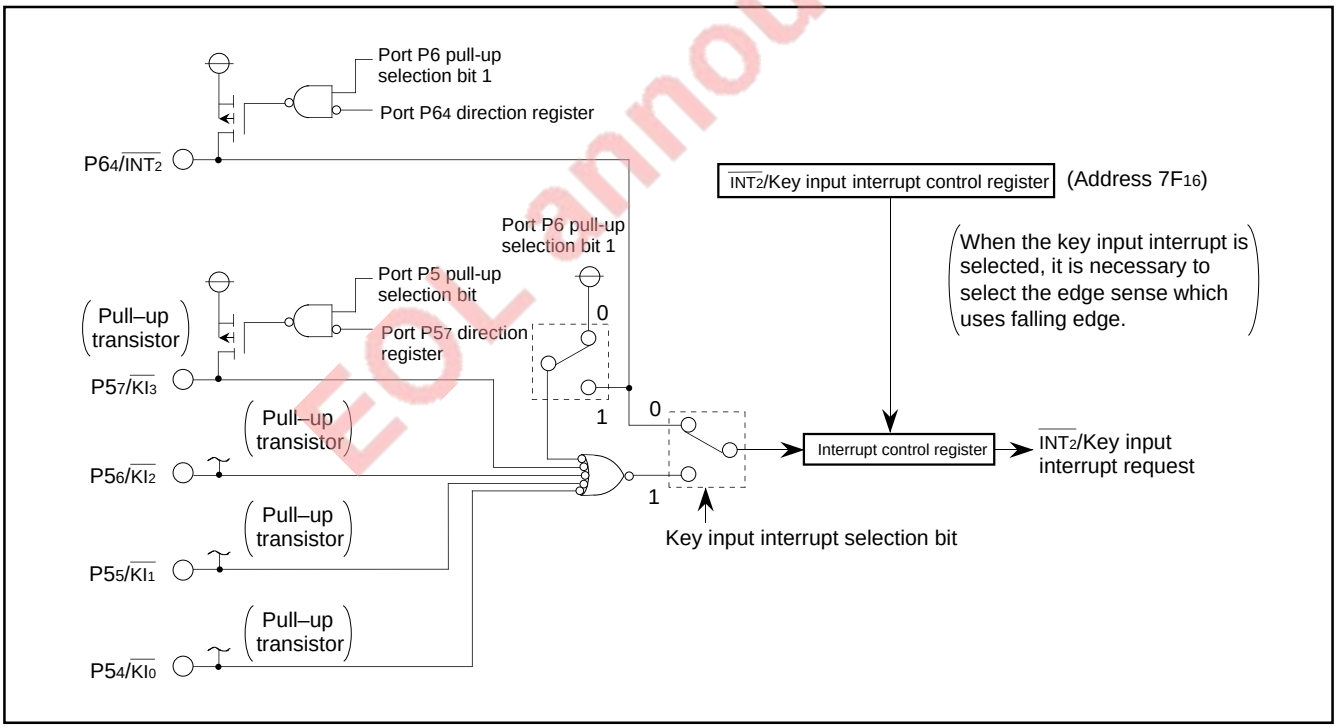
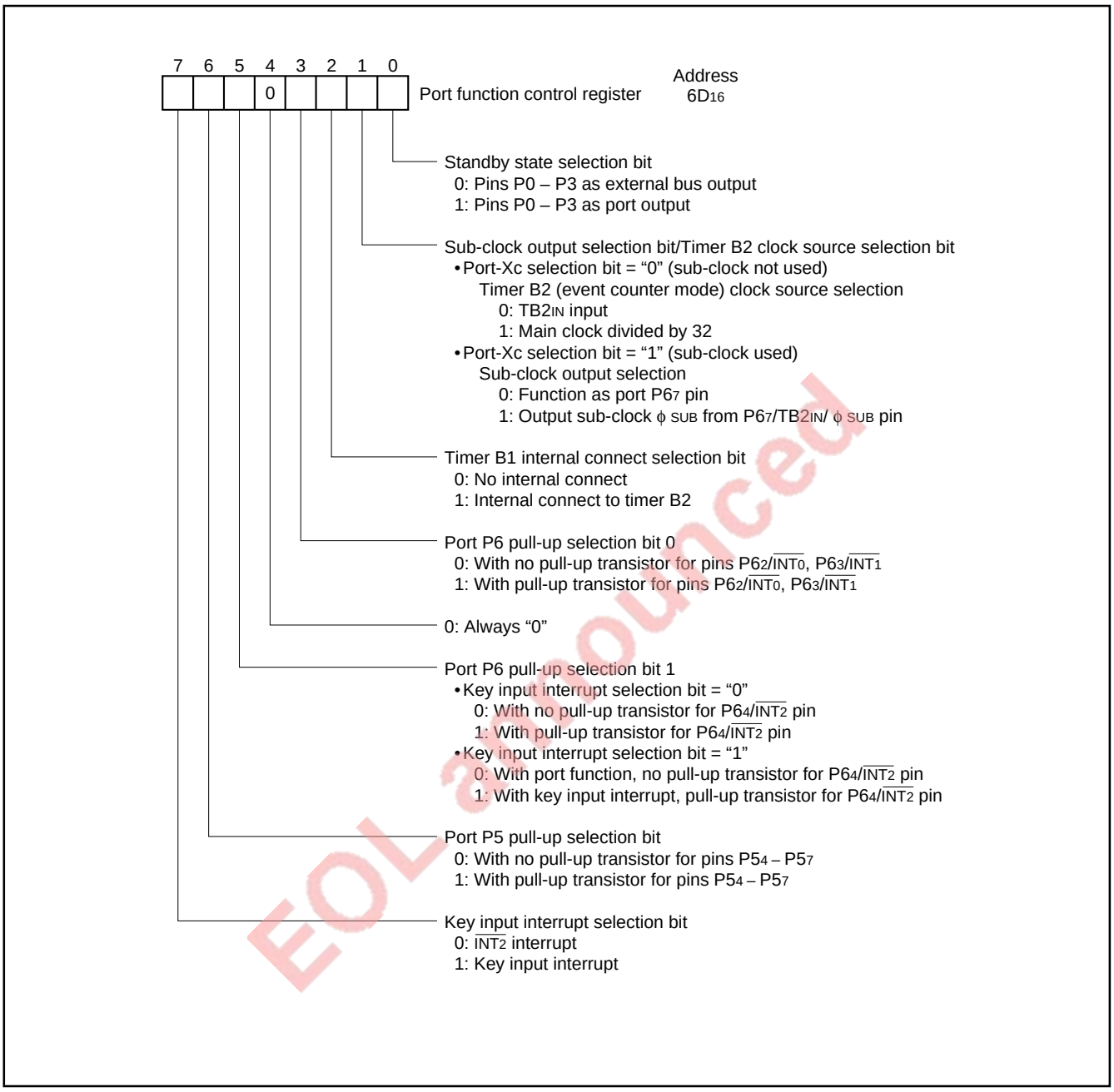


Fig. 12 $\overline{INT_2}$/Key input interrupt input circuit block diagram

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.



EOL announced

Fig. 11 Bit configuration of port function control register

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

TIMER

There are eight 16-bit timers. They are divided by type into timer A(5) and timer B(3).

The timer I/O pins are also used as I/O pins for ports P5 and P6. To use these pins as timer input pins, the port direction register bit corresponding to the pin must be cleared to “0” to specify the input mode.

TIMER A

Figure 13 shows a block diagram of timer A.

Timer A has four modes; timer mode, event counter mode, one-shot pulse mode, and pulse width modulation mode. The mode is selected with bits 0 and 1 of the timer Ai mode register (i = 0 to 4). Each of these modes is described below.

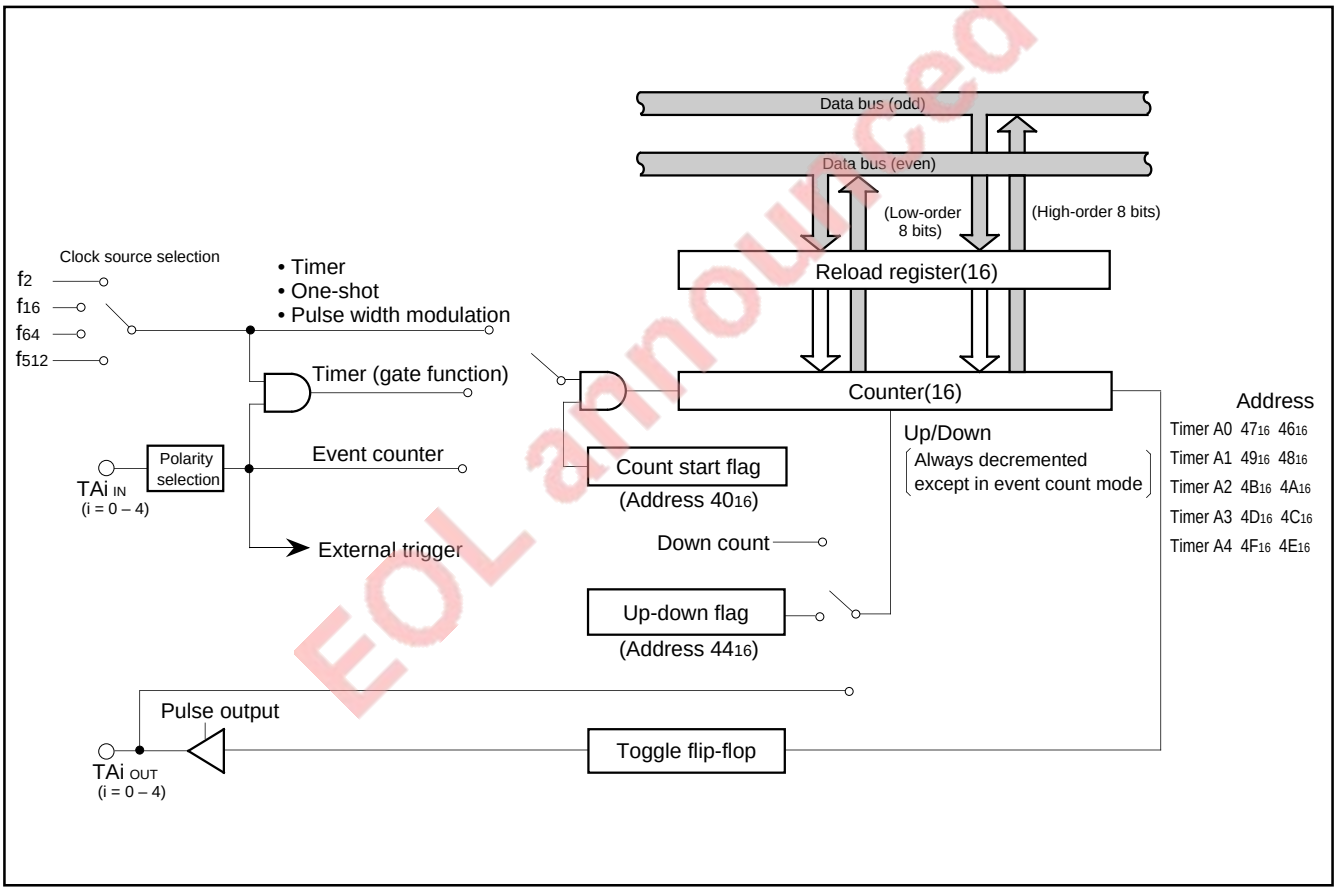
(1) Timer mode [00]

Figure 14 shows the bit configuration of the timer Ai mode register during timer mode. Bits 0, 1, and 5 of timer Ai mode register must always be “0” in the timer mode.

Bit 3 is ignored if bit 4 is “0”.

Bits 6 and 7 are used to select the timer counter source. The counting of the selected clock starts when the count start flag is “1” and stops when it is “0”.

Figure 15 shows the bit configuration of the count start flag. The counter is decremented. An interrupt is caused and the interrupt request bit of the timer Ai interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register are transferred to the counter, and count is continued.



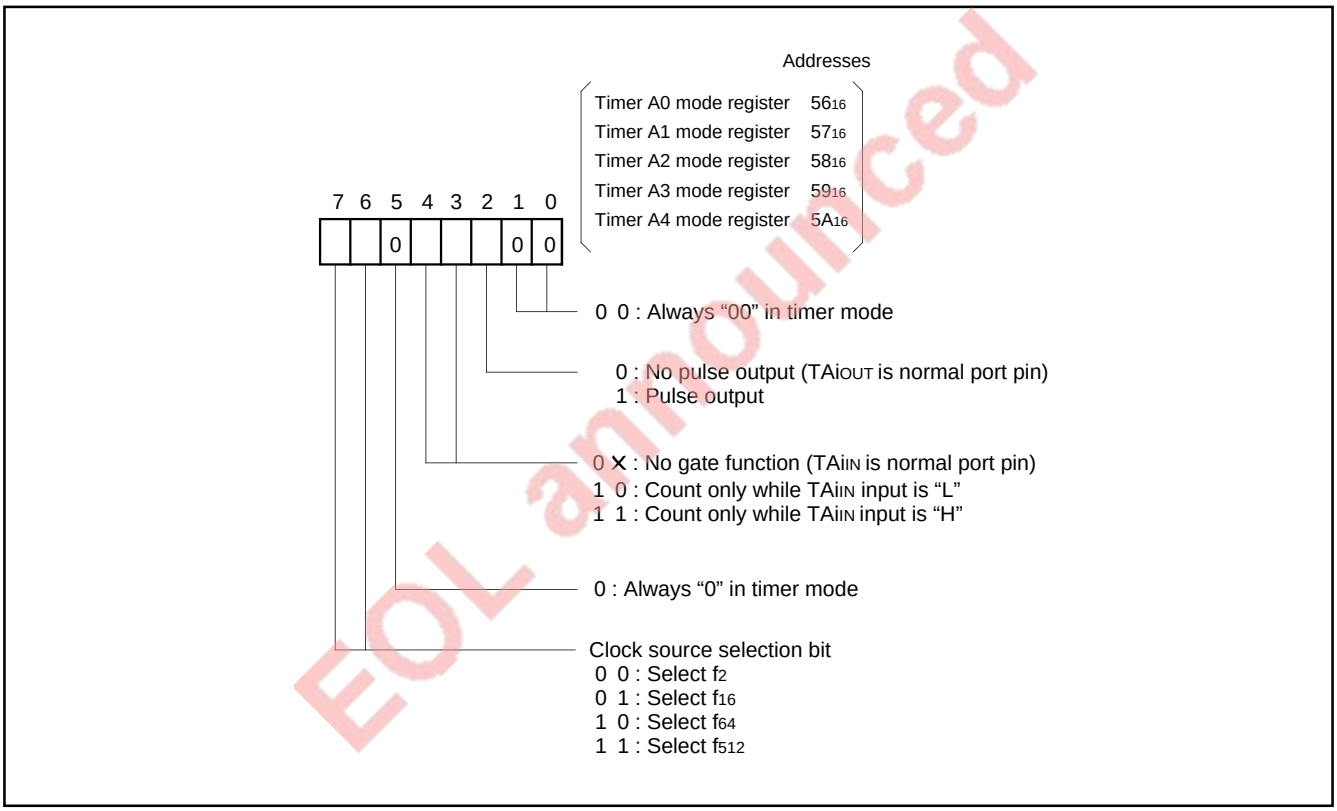
EOL announced

Fig. 13 Block diagram of timer A

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

When bit 2 of the timer Ai mode register is “1”, the output is generated from TAiOUT pin. The output is toggled each time the contents of the counter reaches to 0000_{16}. When the contents of the count start flag is “0”, “L” is output from TAiOUT pin.
When bit 2 is “0”, TAiOUT can be used as a normal port pin.
When bit 4 is “0”, TAiIN can be used as a normal port pin.
When bit 4 is “1”, counting is performed only while the input signal from the TAiIN pin is “H” or “L” as shown in Figure 16. Therefore, this can be used to measure the pulse width of the TAiIN input signal. Whether to count while the input signal is “H” or while it is “L” is determined by bit 3. If bit 3 is “1”, counting is performed while the TAiIN pin input signal is “H” and if bit 3 is “0”, counting is performed while it is “L”.

Note that the duration of “H” or “L” on the TAiIN pin must be two or more cycles of the timer count sourse.
When data is written to the timer Ai register with timer Ai halted, the same data is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The contents of the counter can be read at any time.
When the value set in the timer Ai register is n, the timer frequency dividing ratio is 1/(n + 1).



EOL announced

Fig. 14 Timer Ai mode register bit configuration during timer mode

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

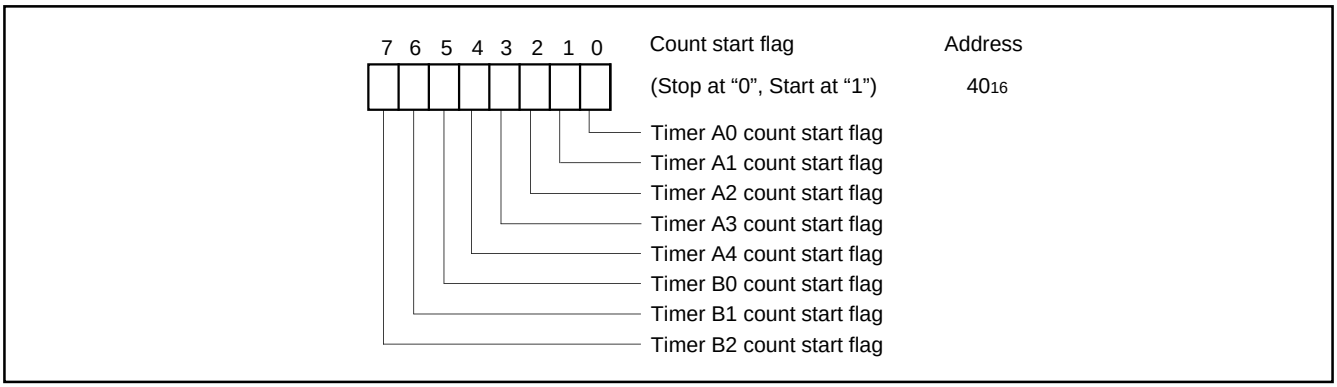
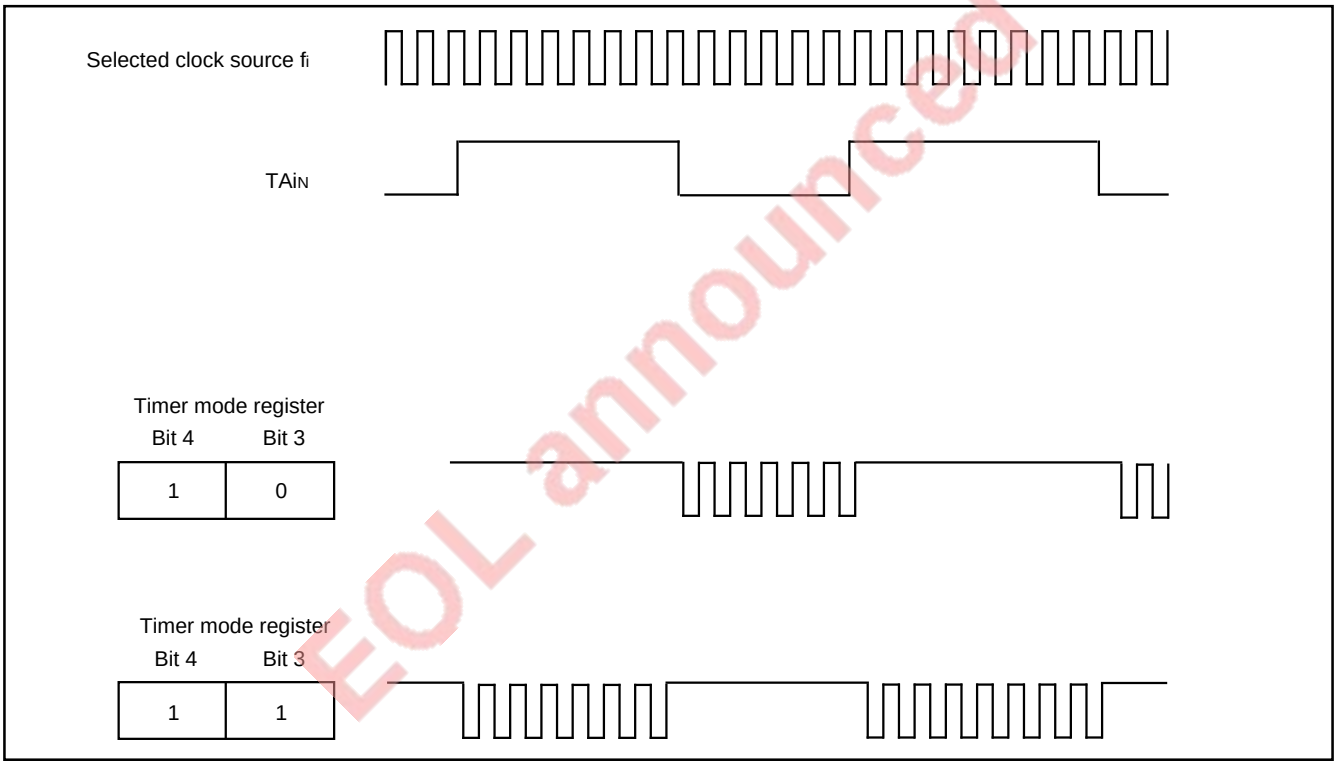


Fig. 15 Count start flag bit configuration



EOL announced

Fig. 16 Count waveform when gate function is available

(2) Event counter mode [01]

Figure 17 shows the bit configuration of the timer Ai mode register during the event counter mode. In the event counter mode, the bit 0 of the timer Ai mode register must be “1” and bits 1 and 5 must be “0”.

The input signal from the TAiIN pin is counted when the count start flag shown in Figure 15 is “1” and counting is stopped when it is “0”. Count is performed at the fall of the input signal when bit 3 is “0” and at the rise of the signal when it is “1”.

In the event counter mode, whether to increment or decrement the count can be selected with the up-down flag or the input signal from the TAiOUT pin.

When bit 4 of the timer Ai mode register is “0”, the up-down flag is used to determine whether to increment or decrement the count (decrement when the flag is “0” and increment when it is “1”). Figure 18 shows the bit configuration of the up-down flag.

When bit 4 of the timer Ai mode register is “1”, the input signal from the TAiOUT pin is used to determine whether to increment or decrement the count. However, note that bit 2 must be “0” if bit 4 is “1”. Because TAiOUT pin becomes an output pin with pulse output if bit 2 is “1”.

The count is decremented when the input signal from the TAiOUT pin is “L” and incremented when it is “H”. Determine the level of the input signal from the TAiOUT pin before an effective edge is input to the TAiIN pin.

An interrupt request signal is generated and the interrupt request bit of the timer Ai interrupt control register is set when the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count). At the same time, timers A0 and A1 transfer the contents of the reload register to the counter and continue counting.

Timers A2, A3, and A4 transfer the contents of the reload register to the counter and continue count when bit 6 of the corresponding timer Ai mode register is “0”, but when bit 6 is “1”, they continue counting without transferring the contents of the reload register to the counter.

When bit 2 is “1”, the waveform of which polarity is reversed each time the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count) is output from TAiOUT pin. If bit 2 is “0”, the TAiOUT pin can be used as a normal port pin. However, if bit 4 is “1” and the TAiOUT pin is used as an output pin, the output from the TAiOUT pin changes the count direction. Therefore, bit 4 must be “0” unless the output from the TAiOUT pin is used to select the count direction.

Data write and data read are performed in the same way as for the timer mode. That is, when data is written to timer Ai which is halted, it is also written to the reload register and the counter.

When data is written to timer Ai which is busy, the data is written to the reload register, but not the counter. The counter is reloaded with new data from the reload register at the next reload time and continues counting. For timers A2, A3, and A4, the contents of the reload register is not reloaded in the counter when bit 6 of the corresponding timer Ai mode register is “1”. The contents of the counter can be read at any time.

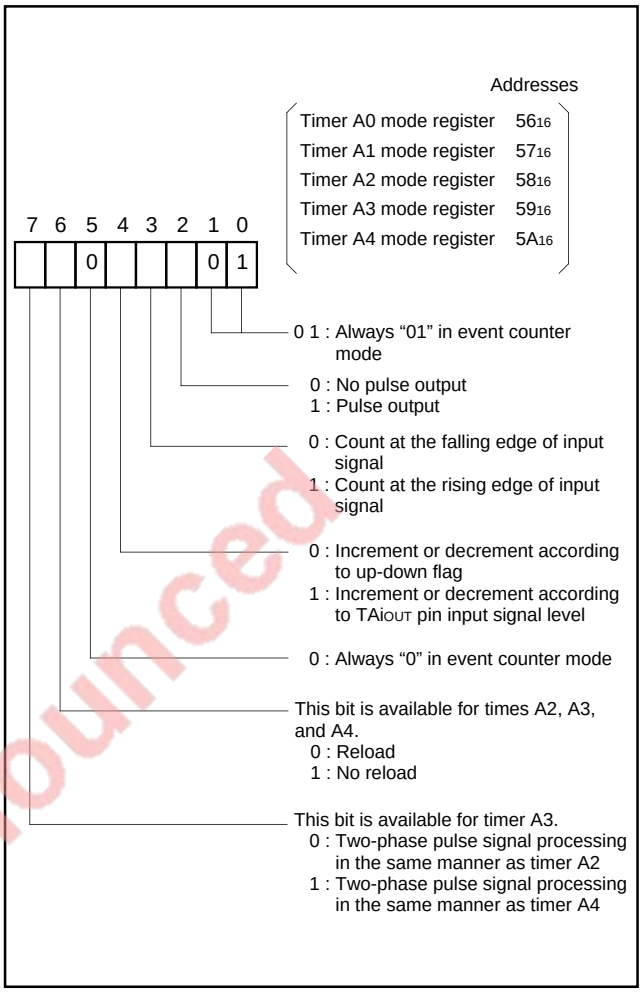


Fig. 17 Timer Ai mode register bit configuration during event counter mode

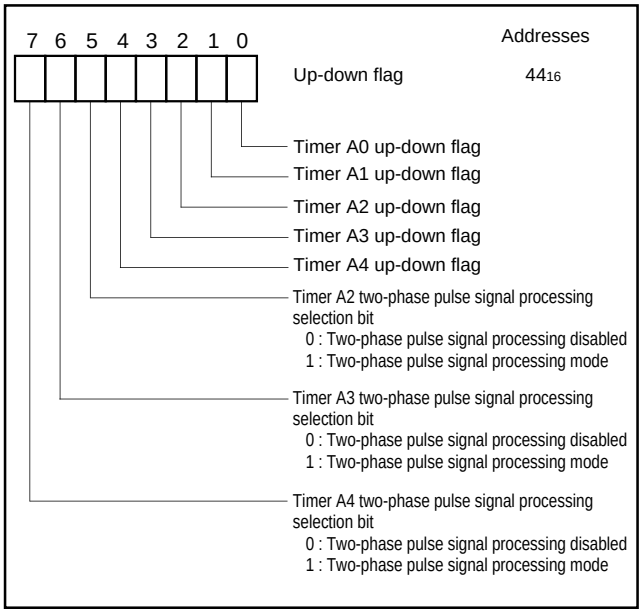


Fig. 18 Up-down flag bit configuration

Not Recommended for New Design

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Furthermore, in the event counter mode, whether to increment or decrement the counter can also be determined by supplying two kinds of pulses of which phases differ by 90° to timer A2, A3, or A4. There are two types of two-phase pulse signal processing operations. One uses timer A2 and the other uses timer A4. Timer A3 can select one of these two operations with bit 7 of the timer A3 mode register. In both processing operations, two kinds of pulses of which phases differ by 90° are input to the TAjOUT (j = 2 to 4) pin and TAjIN pin respectively.

After the level of the TA2OUT pin changes from "L" to "H" with timer A2 used, as shown in Figure 19, the count is incremented when a rising edge is input to the TA2IN pin and the count is decremented when the falling edge is input.

For timer A4, as shown in Figure 20, when a phase related pulse with a rising edge input to the TA4IN pin is input after the level of TA4OUT pin changes from "L" to "H", the count is incremented at the respective rising edge and falling edge of the TA4OUT pin and TA4IN pin. When a phase related pulse with a falling edge input to the TA4OUT pin is input after the level of TA4IN pin changes from "H" to "L", the count is decremented at the respective rising edge and falling edge of the TA4IN pin and TA4OUT pin.

When performing this two-phase pulse signal processing, bits 0 and 4 of the timer Aj mode register must be set to "1" and bits 1, 2, 3, and 5 must be set to "0" as shown in Figure 21.

Bit 7 is used to select whether to perform two-phase pulse signal processing for timer A3 in the same manner as timer A2 or as timer A4. When this bit is "0", two-phase pulse signal processing for timer A3 is performed in the same manner as timer A2 and when it is "1", it is performed in the same manner as timer A4. This bit is ignored for timers A2 and A4.

Note that bits 5, 6, and 7 of the up-down flag (address 44_{16}) are the two-phase pulse signal processing selection bits for timers A2, A3, and A4, respectively.

Each timer operates in the normal event counter mode when the corresponding bit is "0" and performs two-phase pulse signal processing when it is "1".

Count is started by setting the count start flag to "1". Data write and read are performed in the same way as for the normal event counter mode. Note that the port direction register of the input port must be set to the input mode because two-phase pulse signal is input. Also, there can be no pulse output in this mode.

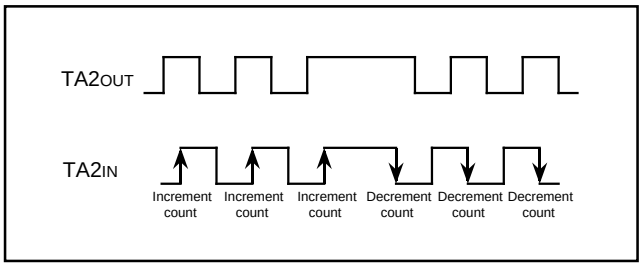


Fig. 19 Two-phase pulse signal processing operation of timer A2

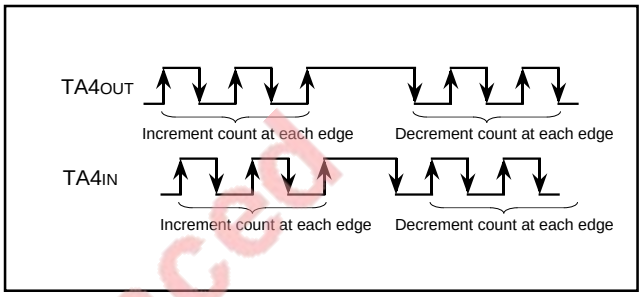


Fig. 20 Two-phase pulse signal processing operation of timer A4

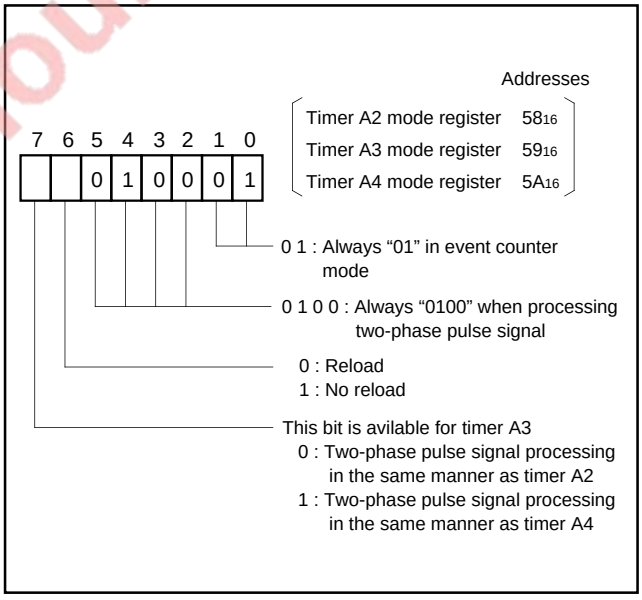


Fig. 21 Timer Aj mode register bit configuration when performing two-phase pulse signal processing in event counter mode

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

(3) One-shot pulse mode [10]

Figure 22 shows the bit configuration of the timer Ai mode resister during the one-shot pulse mode. In the one-shot pulse mode, bit 0 and bit 5 must be “0” and bit 1 and bit 2 must be “1”.

The trigger is enabled when the count start flag is “1”. The trigger can be generated by software, or it can be input from the TAiIN pin. Software trigger is selected when bit 4 is “0”, and the input signal from the TAiIN pin is used as the trigger when bit 4 is “1”.

Bit 3 is used to determine whether to trigger at the fall of the trigger signal or at the rise. The trigger is at the fall of the trigger signal when bit 3 is “0” and at the rise of the trigger signal when bit 3 is “1”.

Software trigger is generated by setting the bit of the one-shot start flag corresponding to each timer.

Figure 23 shows the bit configuration of the one-shot start flag.

As shown in Figure 24, when a trigger signal is received, the counter counts the clock selected by bits 6 and 7.

If the contents of the counter is not 0000_{16}, the TAiOUT pin goes “H” when a trigger signal is received. The count direction is decrement. When the counter reaches 0001_{16}, the TAiOUT pin goes “L” and count is stopped. The contents of the reload register is transferred to the counter. At the same time, an interrupt request signal is generated, and the interrupt request bit of the timer Ai interrupt control register is set. This is repeated each time a trigger signal is received. The output pulse width is

$$\frac{1}{\text{pulse frequency of the selected clock}} \times \text{(counter's value at the time of trigger)}.$$

If the count start flag is “0”, the level of the TAiOUT pin goes “L”. Therefore, the counter’s value corresponding to the desired pulse width must be written to timer Ai before setting “1” to the timer Ai count start flag.

As shown in Figure 25, a trigger signal can be received before the operation for the previous trigger signal is completed. In this case, the contents of the reload register is transferred to the counter by the trigger, and then that value is decremented.

Except when retriggering while operating, the contents of the reload register is not transferred to the counter by triggering.

When retriggering, there must be at least two timer count source cycles before a new trigger can be issued.

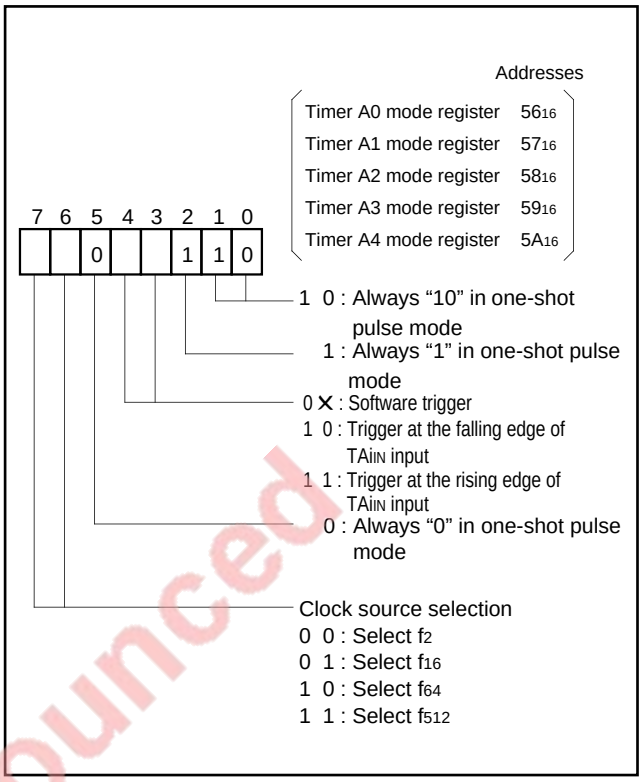


Fig. 22 Timer Ai mode register bit configuration during one-shot pulse mode

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Data write is performed in the same way as for the timer mode. When data is written in timer Ai halted, it is also written to the reload register and the counter.
When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time and continues counting.
Undefined data is read when timer Ai is read.

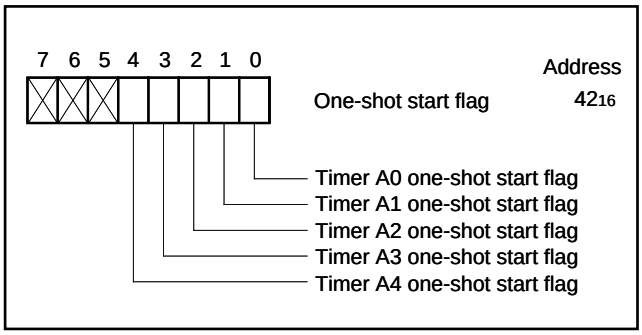


Fig. 23 One-shot start flag bit configuration

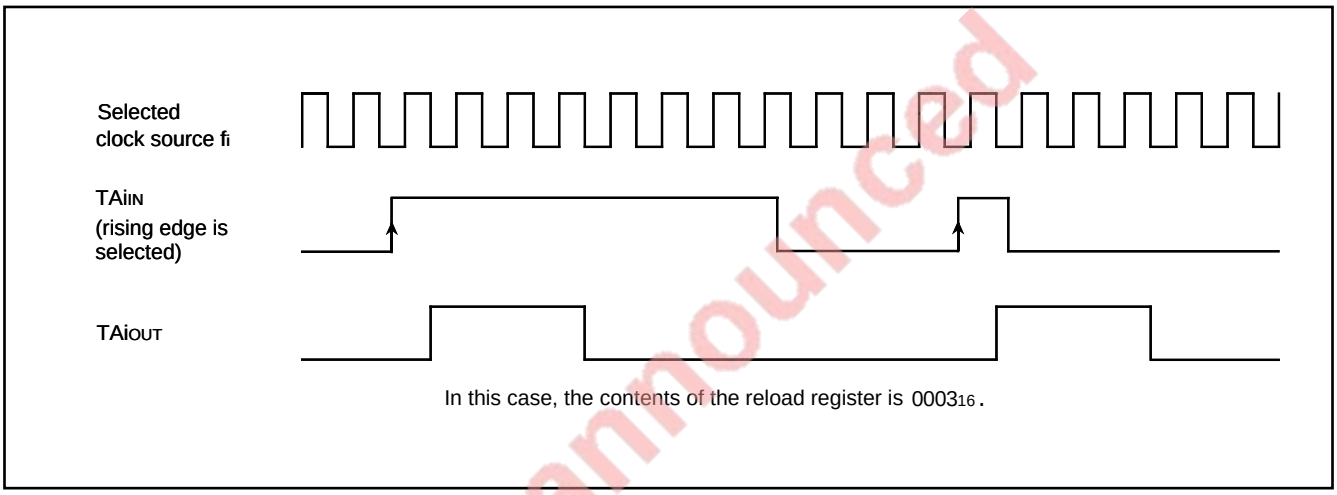


Fig. 24 Pulse output example when external rising edge is selected

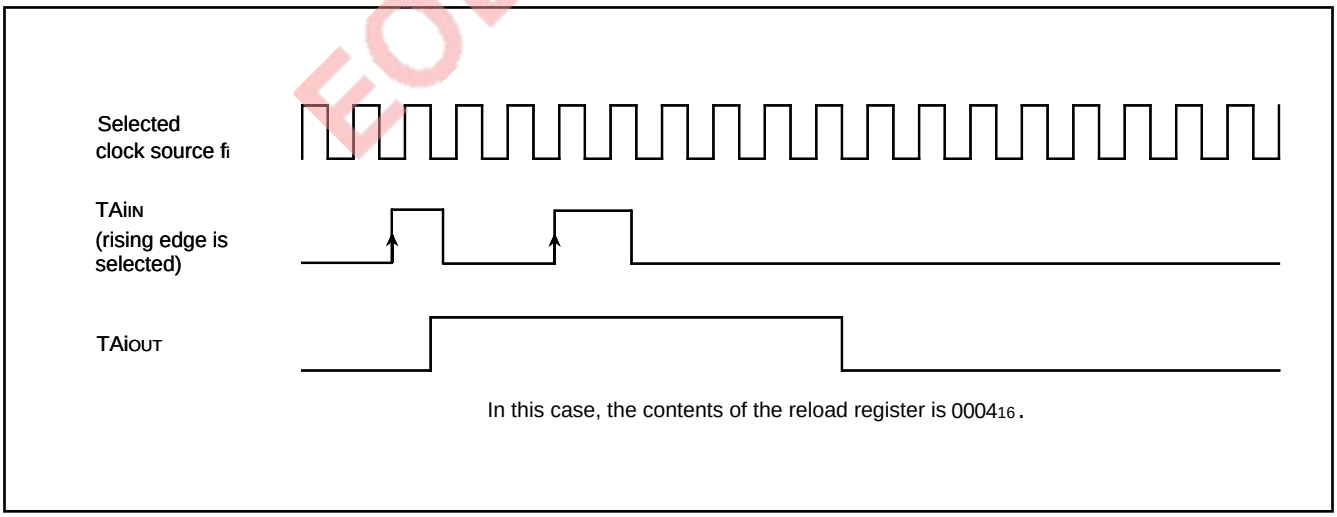


Fig. 25 Example when trigger is re-issued during pulse output

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

(4) Pulse width modulation mode [11]

Figure 26 shows the bit configuration of the timer Ai mode register during the pulse width modulation mode. In the pulse width modulation mode, bits 0, 1, and 2 must be set to “1”.

Bit 5 is used to determine whether to perform as the 16-bit length pulse width modulator or the 8-bit length pulse width modulator. 16-bit length pulse width modulator is selected when bit 5 is “0” and 8-bit length pulse width modulator is selected when bit 5 is “1”. The 16-bit length pulse width modulator is described first.

The pulse width modulator can be started with a software trigger or with an input signal from a TAi_{IN} pin (external trigger).

The software trigger mode is selected when bit 4 is “0”. Pulse width modulator is started and pulse is output from the TAi_{OUT} pin when the timer Ai start flag is set to “1”.

The external trigger mode is selected when bit 4 is “1”. Pulse width modulator starts when a trigger signal is input from the TAi_{IN} pin when the timer Ai start flag is “1”. Whether to trigger at the fall or rise of the trigger signal is determined by bit 3. The trigger is at the fall of the trigger signal when bit 3 is “0” and at the rise when it is “1”.

When data is written to timer Ai with the pulse width modulator halted, it is written to the reload register and the counter.

Then when the timer Ai start flag is set to “1” and a software trigger or an external trigger is issued to start modulation, the waveform shown in Figure 27 is output continuously. Once modulation is started, triggers are not accepted. If the value in the reload register is m, the duration “H” of pulse is

$$\frac{1}{\text{selected clock frequency}} \times m$$

and the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (2^{16} - 1).$$

An interrupt request signal is generated and the interrupt request bit of the timer Ai interrupt control register is set at each fall of the output pulse.

The width of the output pulse is changed by updating timer data. The update can be performed at any time. The output pulse width is changed at the rise of the pulse after data is written to the timer.

The contents of the reload register is transferred to the counter just before the rise of the next output pulse so that the pulse width is changed from the next output pulse.

Undefined data is read when timer Ai is read.

The 8-bit length pulse width modulator is described next.

The 8-bit length pulse width modulator is selected when bit 5 of the timer Ai mode register is “1”.

The reload register and the counter are both divided into 8-bit halves. The low-order 8 bits function as a prescaler and the high-order 8 bits function as the 8-bit length pulse width modulator. The prescaler counts the clock selected by bits 6 and 7. A pulse is generated when the counter reaches 0000_{16} as shown in Figure 28. At the same time, the contents of the reload register is transferred to the counter, and count is continued.

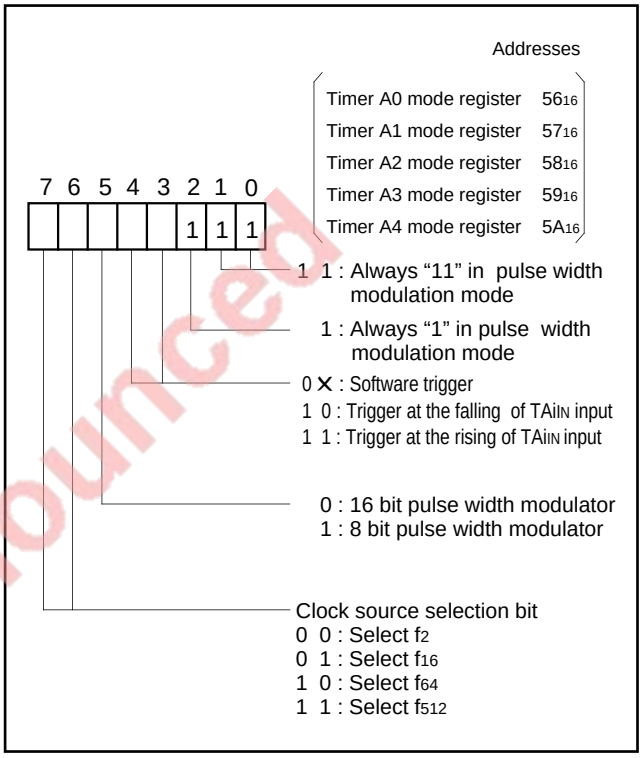


Fig. 26 Timer Ai mode register bit configuration during pulse width modulation mode

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Therefore, if the low-order 8 bits of the reload register is n, the period of the generated pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n + 1).$$

The high-order 8 bits function as an 8-bit length pulse width modulator using this pulse as input. Its operation is the same as for 16-bit length pulse width modulator except it has a length of 8 bits. If the high-order 8 bits' contents of the reload register is m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n + 1) \times m.$$

And the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (n + 1) \times (2^8 - 1).$$

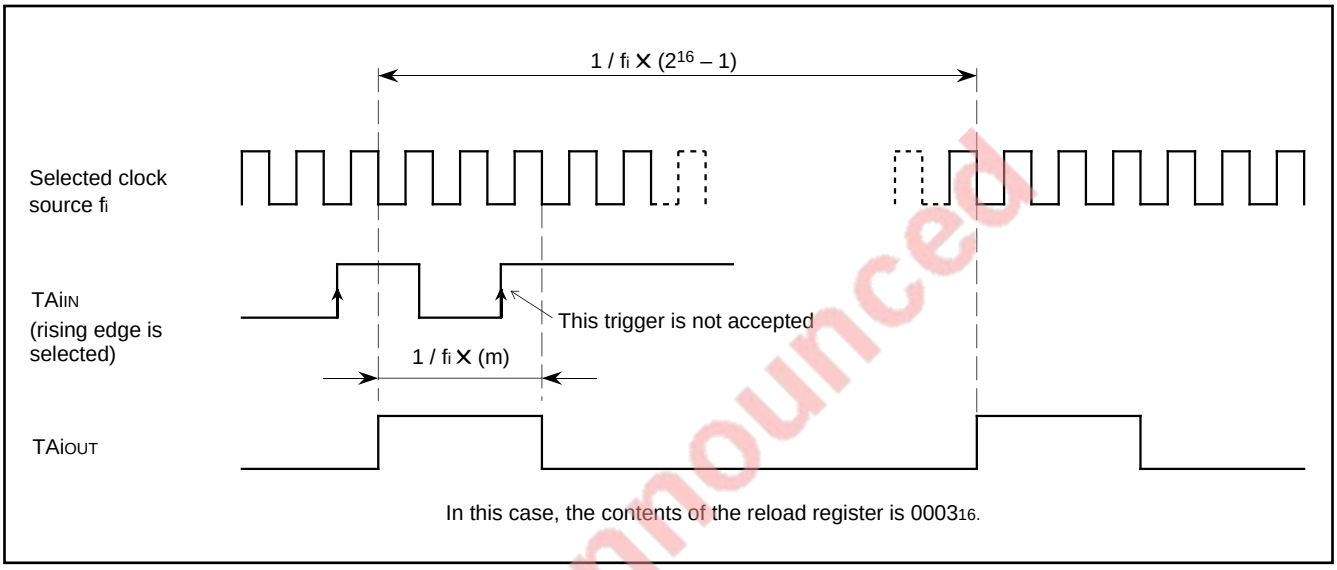


Fig. 27 16-bit length pulse width modulator output pulse example

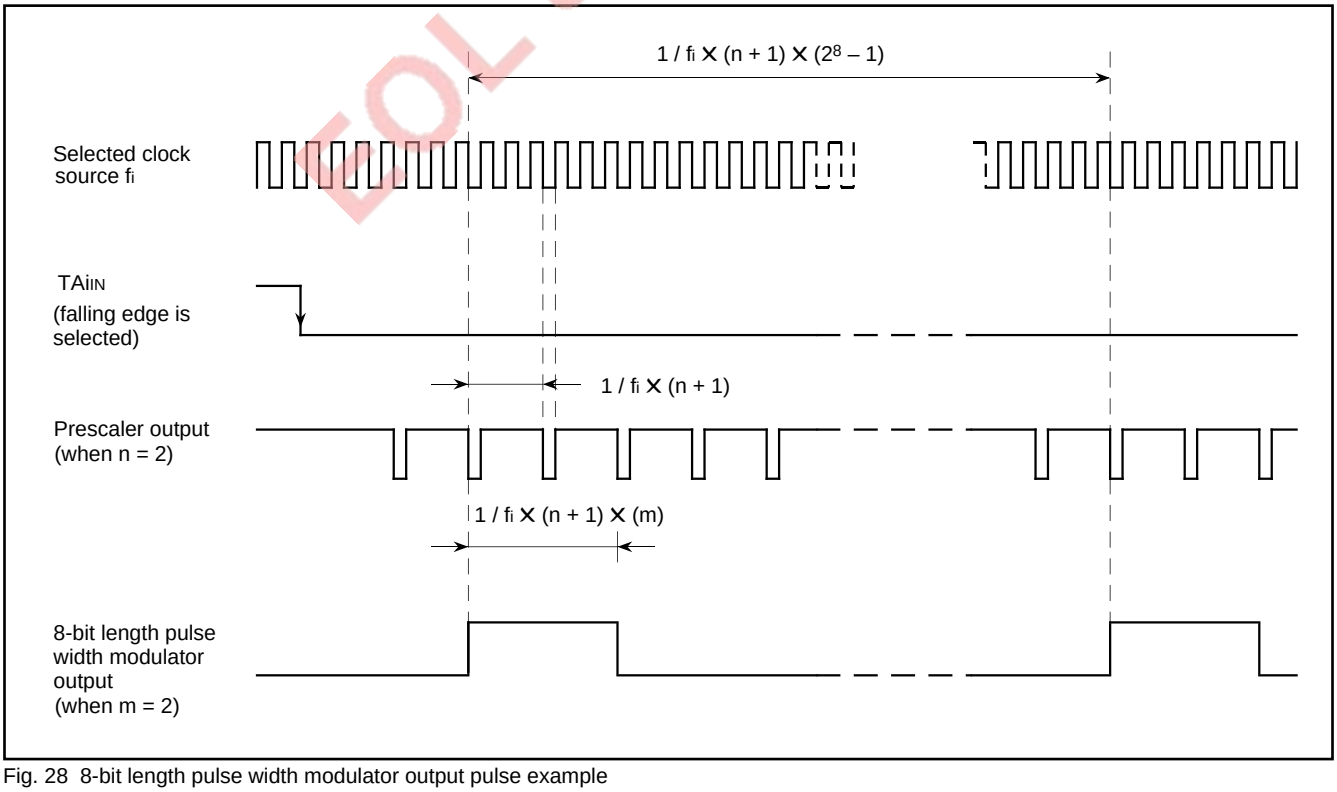


Fig. 28 8-bit length pulse width modulator output pulse example

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

TIMER B

Figure 29 shows a block diagram of timer B.

Timer B has three modes; timer mode, event counter mode, and pulse period measurement/pulse width measurement mode. The mode is selected with bits 0 and 1 of the timer Bi mode register (i = 0 to 2). Timer B2 can also be used as the clock timer of which clock source is the main clock or the sub-clock divided by 32. Additionally, timer B2 can be internally connected to timer B1 (cascade connection). Each of these modes is described below.

(1) Timer mode [00]

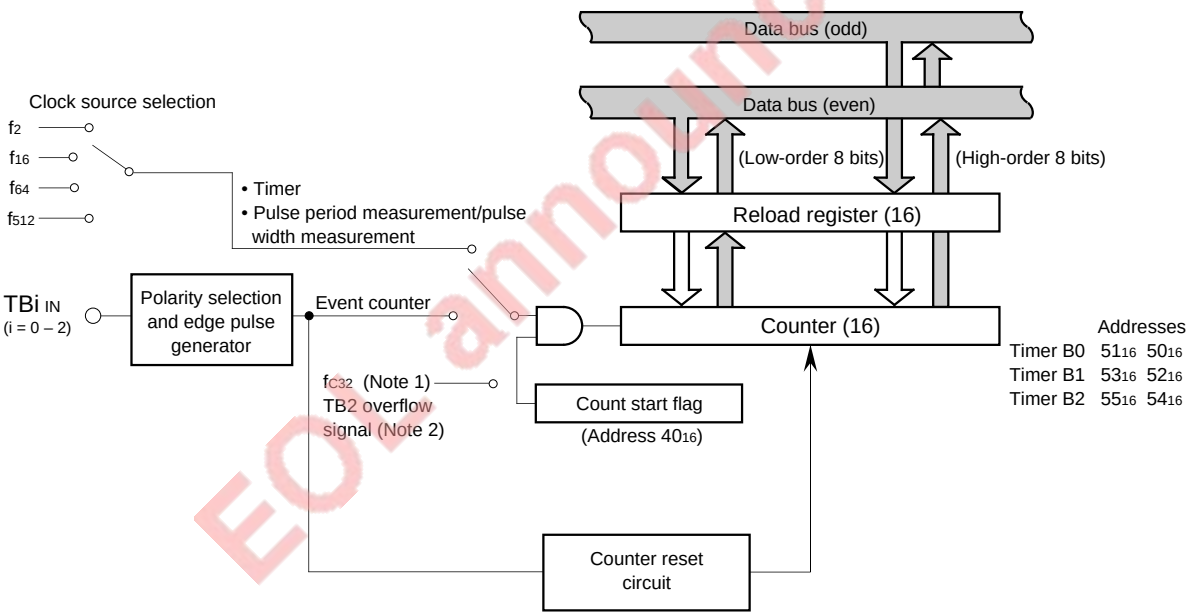
Figure 30 shows the bit configuration of the timer Bi mode register during the timer mode. Bits 0 and 1 of the timer Bi mode register must always be “0” in the timer mode.

Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start flag is “1” and stops when it is “0”.

As shown in Figure 15, the timer Bi count start flag is at the same address as the timer Ai count start flag. The count is decremented. When the contents of the counter becomes 0000_{16}, an interrupt request occurs and the interrupt request bit of the timer Bi interrupt control register is set. At the same time, the contents of the reload register is stored in the counter, and count is continued.

Timer Bi does not have a pulse output function or a gate function like timer A.

When data is written to timer Bi halted, it is written to the reload register and the counter. When data is written to timer Bi which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time and continues counting. The contents of the counter can be read at any time.



Notes 1. Clock source of clock timer; Only Timer B2 can select it (refer to Fig. 65)
2. Only Timer B1 can select it (internal connect mode)

Fig. 29 Timer B block diagram

EO announced

(2) Event counter mode [01]

Figure 31 shows the bit configuration of the timer Bi mode register during the event counter mode. In the event counter mode, the bit 0 of the timer Bi mode register must be “1” and bit 1 must be “0”.

The input signal from the TBiIN pin is counted when the count start flag is “1”, and counting is stopped when it is “0”. Counting is performed at the fall of the input signal when bits 2 and 3 are “0” and at the rise of the input signal when bit 3 is “0” and bit 2 is “1”.

When bit 3 is “1” and bit 2 is “0”, counting is performed at the rise and fall of the input signal.

When the sub-clock (32 kHz) oscillation circuit is used and others, and the event counter mode is selected, timer B2 functions as the clock timer and the original functions as timer B2 in the event counter mode are lost. For details, refer to “(4) Clock timer”.

When the internal connect mode which connects timer B1 to timer B2 is selected, the original function as timer B1 in the event counter mode is lost. For details, refer to “(5) Internal connect mode”.

Data write, data read, and interrupt generation are performed in the same way as for the timer mode.

(3) Pulse period measurement/pulse width measurement mode [10]

Figure 32 shows the bit configuration of the timer Bi mode register during the pulse period measurement/pulse width measurement mode.

In the pulse period measurement/pulse width measurement mode, bit 0 must be “0” and bit 1 must be “1”. Bits 6 and 7 are used to select the clock source. The selected clock is counted when the count start flag is “1”, and counting stops when it is “0”.

The pulse period measurement mode is selected when bit 3 is “0”. In the pulse period measurement mode, the selected clock is counted during the interval starting at the fall of the input signal from the TBiIN pin to the next fall or at the rise of the input signal to the next rise. And then, the result is stored in the reload register. In this case, the reload register acts as a buffer register.

When bit 2 is “0”, the clock is counted from the fall of the input signal to the next fall. When bit 2 is “1”, the clock is counted from the rise of the input signal to the next rise.

In the case of counting from the fall of the input signal to the next fall, counting is performed as follows. As shown in Figure 33, when the fall of the input signal from TBiIN pin is detected, the contents of the counter is transferred to the reload register. Next the counter is cleared and count is started from the next clock. When the fall of the next input signal is detected, the contents of the counter is transferred to the reload register once more, the counter is cleared, and counting is started. The period from the fall of the input signal to the next fall is measured in this way.

After the contents of the counter is transferred to the reload register, an interrupt request signal is generated and the interrupt request bit of the timer Bi interrupt control register is set. However, no interrupt request signal is generated when the contents of the counter is transferred first time to the reload register after the count start flag is set to “1”.

When bit 3 is “1”, the pulse width measurement mode is selected. The pulse width measurement mode is similar to the pulse period measurement mode except that the clock is counted from the fall of the TBiIN pin input signal to the next rise or from the rise of the input signal to the next fall as shown in Figure 34.

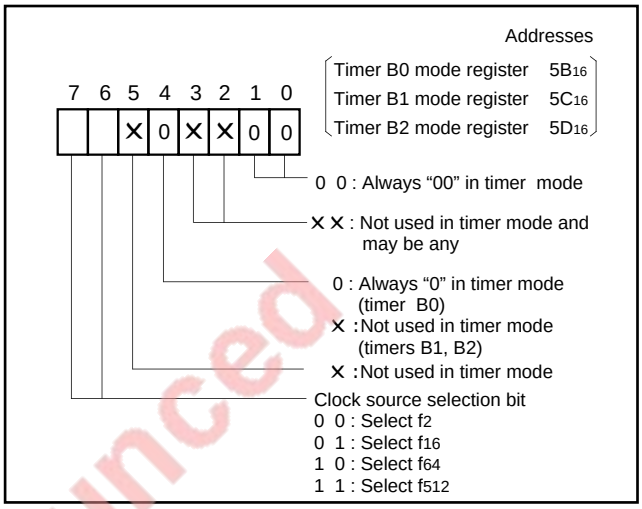


Fig. 30 Timer Bi mode register bit configuration during timer mode

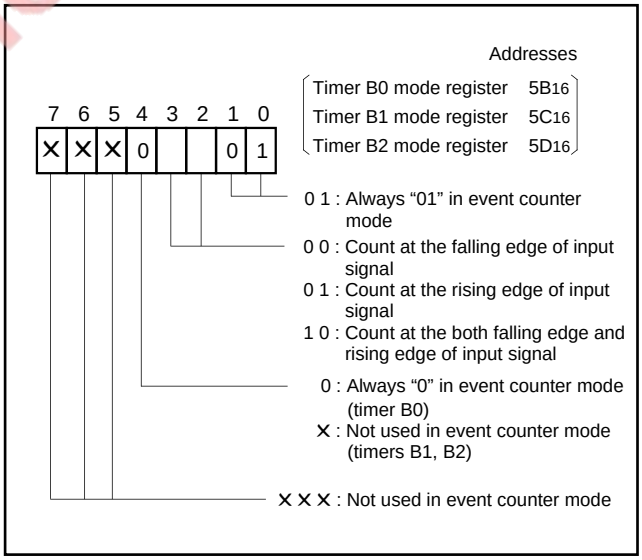


Fig. 31 Timer Bi mode register bit configuration during event counter mode

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

When timer Bi is read, the contents of the reload register is read.
Note that, in this mode, the interval from the fall of the TBiIN pin input signal to the next rise or from the rise to the next fall must be at least two cycles of the timer count source.
Timer Bi overflow flag which is bit 5 of the timer Bi mode register is set to "1" when the timer Bi counter reaches 0000_{16}. This flag is cleared by writing to the corresponding timer Bi mode register. By reading this flag, the reason why the interrupt request signal is generated, which is the completion of measurement or the counter overflow, can be detected. An interrupt request signal may occur because the counter value is particularly undefined just after counting starts.
Accordingly, make sure to detect the occurrence reason of an interrupt request signal with the timer Bi overflow flag. This flag is "1" at reset.
When using timer B2 as the clock timer and using timer B1 in the internal connect mode, functions in this mode are lost.

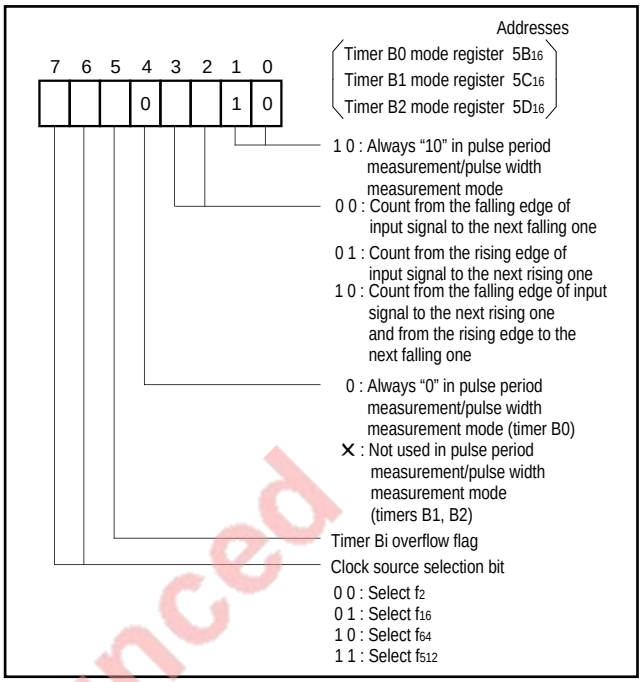


Fig. 32 Timer Bi mode register bit configuration during pulse period measurement/pulse width measurement mode

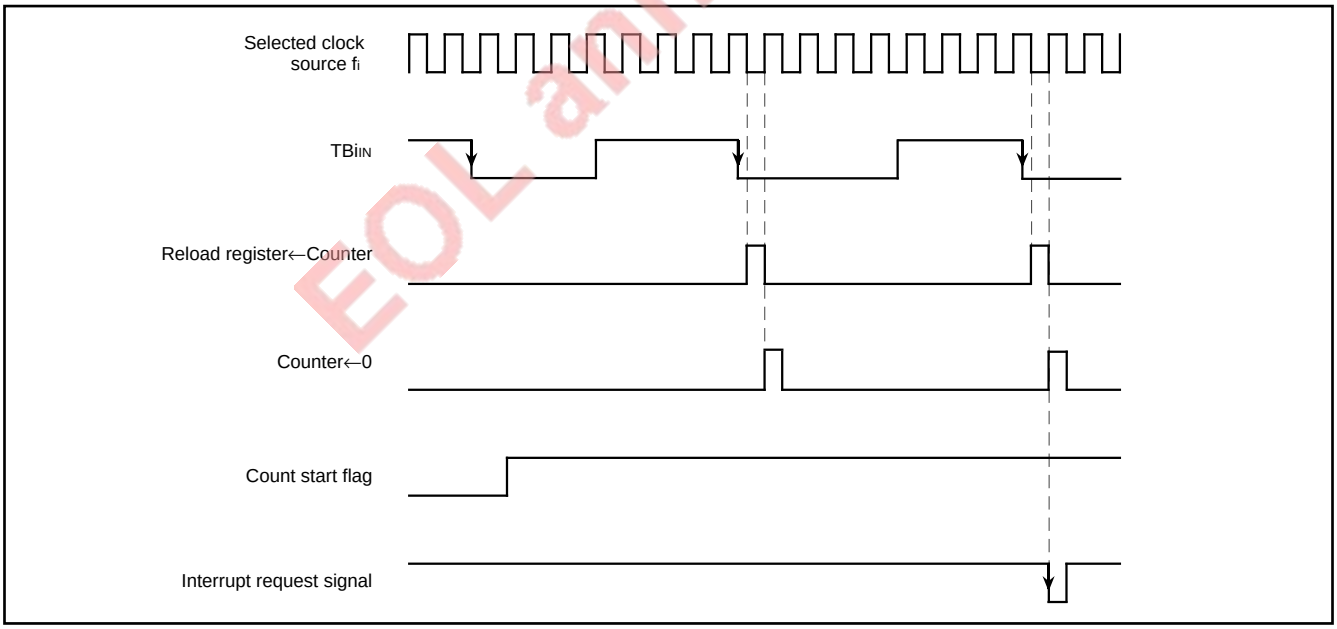


Fig. 33 Pulse period measurement mode operation (example of measuring the interval from the falling edge to next falling one)

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

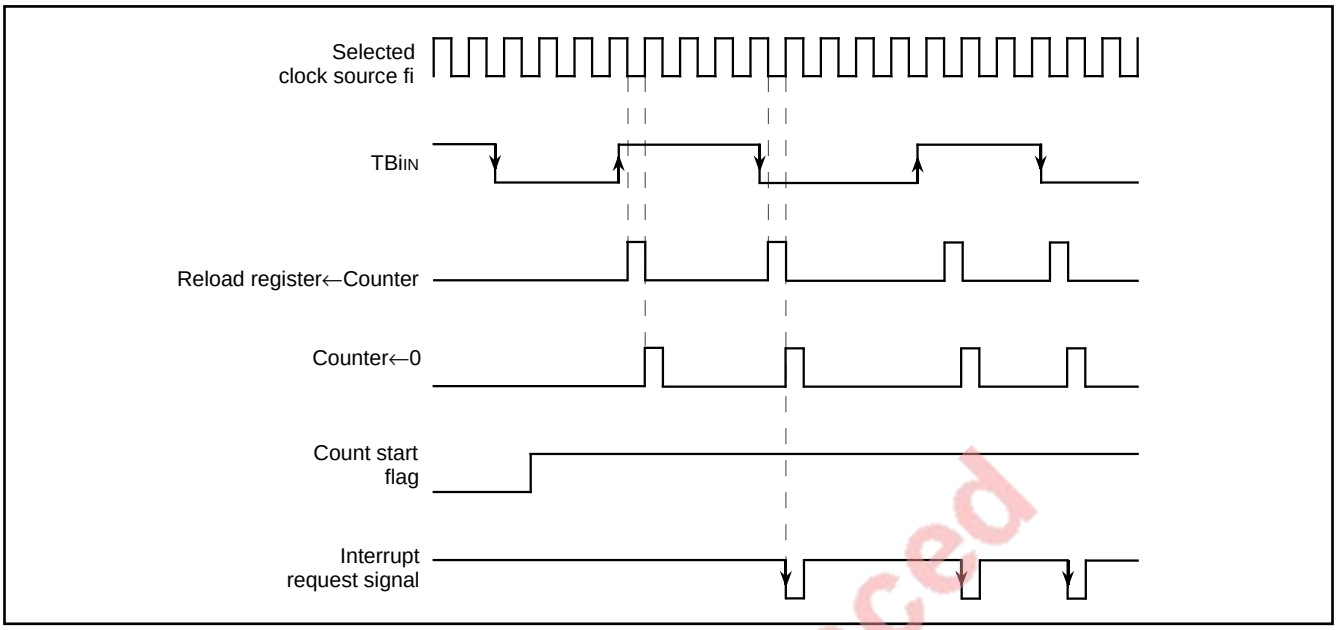


Fig. 34 Pulse width measurement mode operation

(4) Clock timer

When the port-Xc selection bit of the oscillation circuit control register 0 (refer to Figure 63) is set to "1" to make the sub-clock oscillation circuit active, timer B2 can function as the clock timer, which uses clock f_{C32} as the clock source. Clock f_{C32} is the sub clock (32 kHz) divided by 32.

Additionally, when the port-Xc selection bit is set to "0" not to use the sub-clock and the timer B2 clock source selection bit of the port function control register (refer to Figure 11) is set to "1", timer B2 can functions as the clock timer, which uses clock f_{C32} as the clock source. Clock f_{C32} is the main clock divided by 32.

Figure 35 shows the timer B2 mode register bit configuration when timer B2 is used as the clock timer. As shown in Figure 35, the event counter mode must be selected for timer B2.

For how to use the clock timer, refer to the section on clock generating circuit.

(5) Internal connect mode

When the timer B1 internal connect selection bit of the port function control register (refer to Figure 11) is set to "1", timer B1 uses the timer B2's overflow signal as the clock source and timer B1 is internally connected to timer B2 (cascade connection).

The internal connect mode makes timers B1 and B2 function as 16 + 16 bit-timer with the timer B2's clock source.

Figure 35 shows the timer B1 mode register bit configuration when using timer B1 in the internal connect mode. Set timer B1 in the event counter mode as shown in Figure 35.

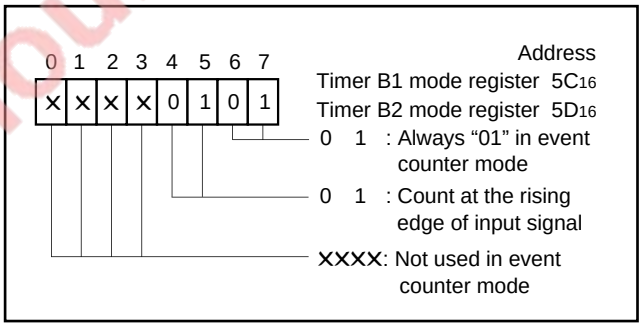


Fig. 35 Timer B1 mode register bit configuration when timer B1 is used in the internal connect mode and timer B2 mode register bit configuration when timer B2 is used as clock timer

EOL announced

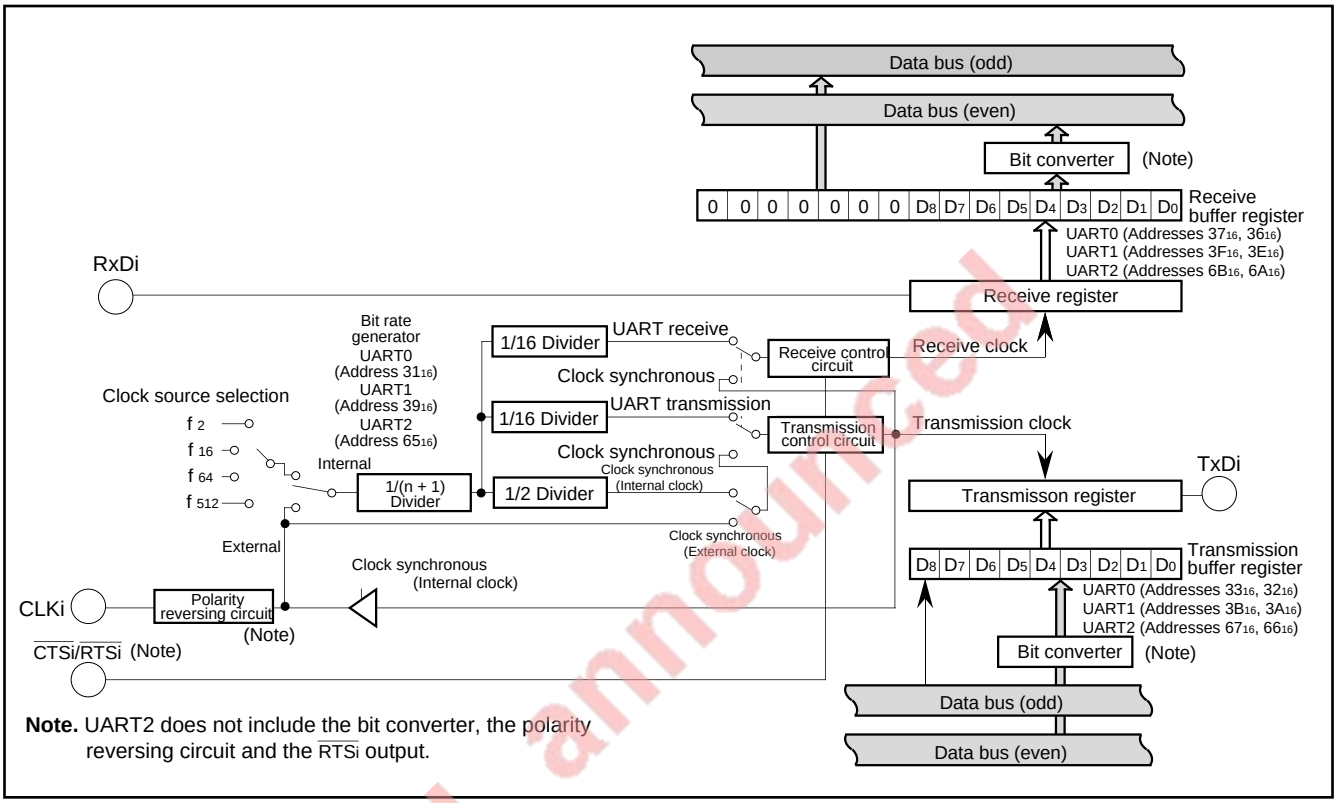
SERIAL I/O PORTS

Three independent serial I/O ports are provided.

Figure 36 shows a block diagram of the serial I/O ports.

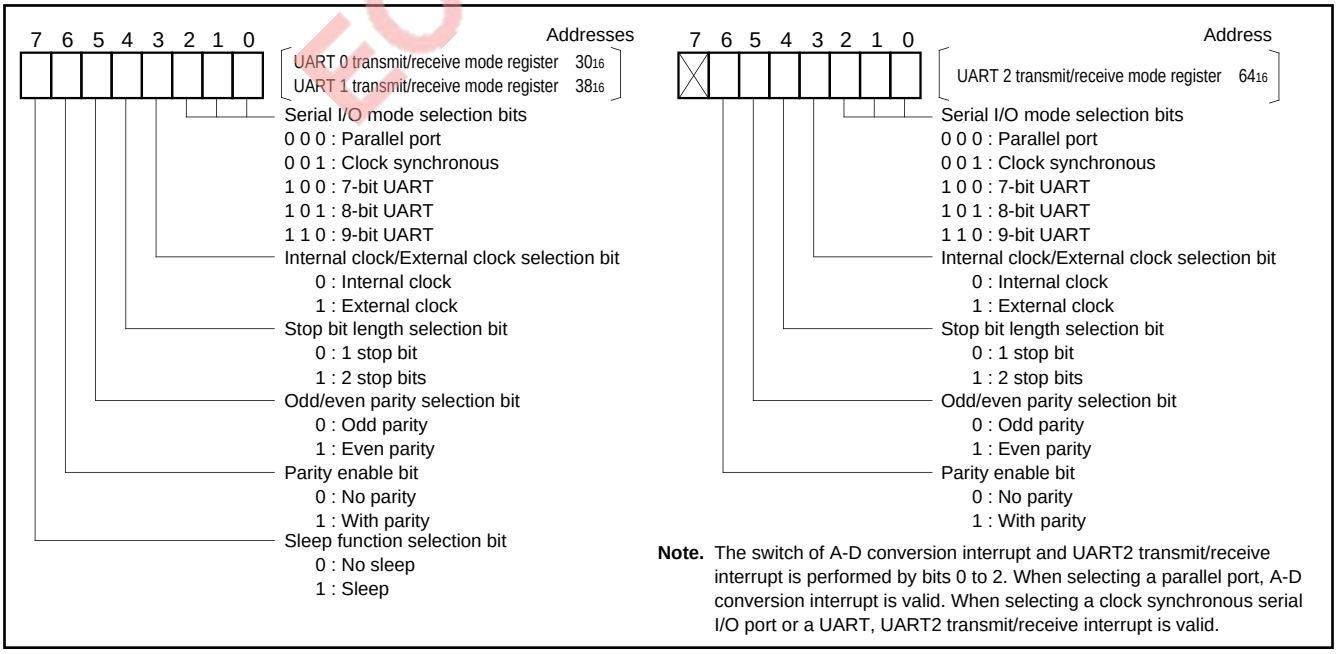
Table 5 shows the functional differences of three serial I/O ports (UART 0, 1, 2).

Bits 0, 1, and 2 of the UARTi (i = 0, 1, 2) transmit/receive mode register shown in Figure 37 are used to determine whether to use port P8 or port P7 as a parallel port, a clock synchronous serial I/O port, or an asynchronous serial I/O port (UART) using start and stop bits.



Note. UART2 does not include the bit converter, the polarity reversing circuit and the $\overline{RTSi}$ output.

Fig. 36 Serial I/O port block diagram



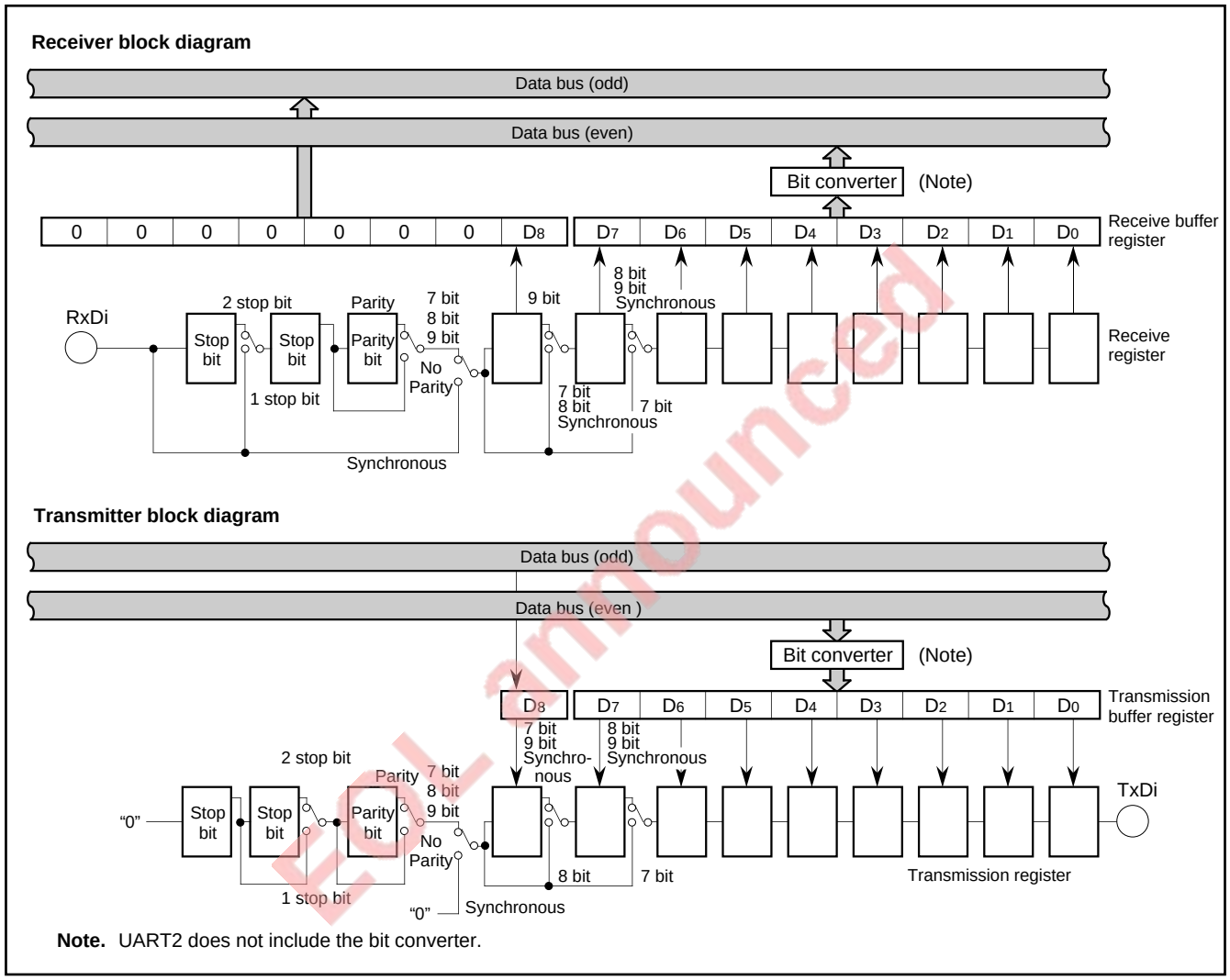
Note. The switch of A-D conversion interrupt and UART2 transmit/receive interrupt is performed by bits 0 to 2. When selecting a parallel port, A-D conversion interrupt is valid. When selecting a clock synchronous serial I/O port or a UART, UART2 transmit/receive interrupt is valid.

Fig. 37 UARTi transmit/receive mode register bit configuration

FOL announced

The interrupt vector and the interrupt control register are common to the A-D conversion interrupt and UART2 transmit/receive interrupt. It is switched by a selection of UART2 function as shown in Figure 37 and Table 5.
Figure 38 shows the connections of receiver/transmitter.

Figures 39 and 40 show the bit configuration of the UARTi transmit/receive control register.
Each communication method is described below.



Note. UART2 does not include the bit converter.

Fig. 38 Receiver and transmitter block diagram

Table 5. Differences between UART0, UART1 and UART2

	Communication method	$\overline{CTS}$ input/ $\overline{RTS}$ output	Interrupt	Selection of data output, CLK polarity, transfer format	Multiple clocks output	Sleep function
UART0	Selection of clock synchronous or asynchronous (UART)	Both $\overline{CTS}$ input and $\overline{RTS}$ output	Transmit and receive (2 systems)	Available	Available	Available
UART1	Selection of clock synchronous or asynchronous (UART)	Both $\overline{CTS}$ input and $\overline{RTS}$ output	Transmit and receive (2 systems)	Available	Nothing	Available
UART2	Selection of clock synchronous or asynchronous (UART)	Only $\overline{CTS}$ input	Transmit/receive (1 system) (Note)	Nothing	Nothing	Nothing

Note. The interrupt vector and the interrupt control register are common to the A-D conversion interrupt and UART2 transmit/receive interrupt. It is switched by a selection of UART2 function.

Not announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

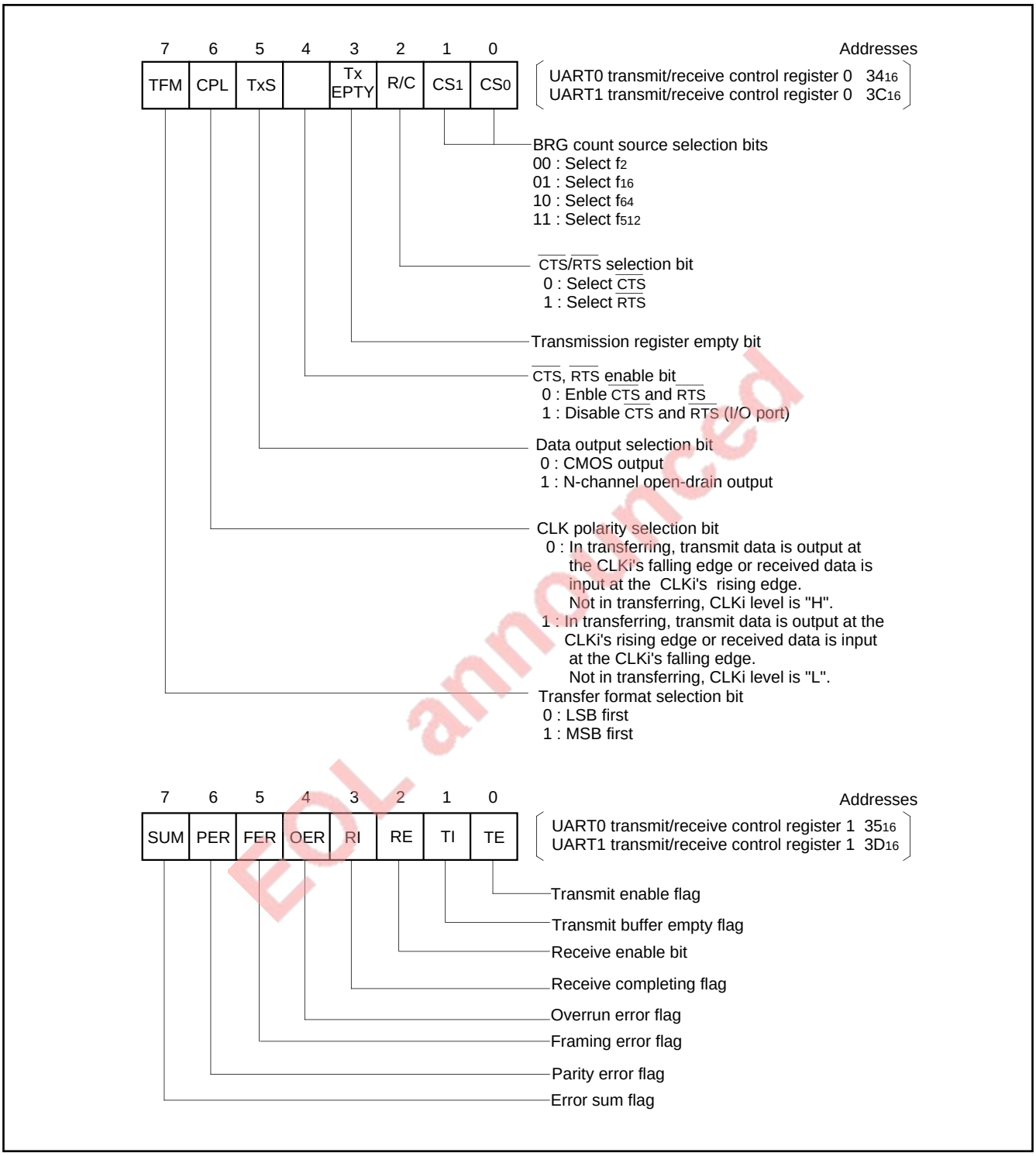


Fig. 39 UART0, UART1 transmit/receive control registers bit configuration

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

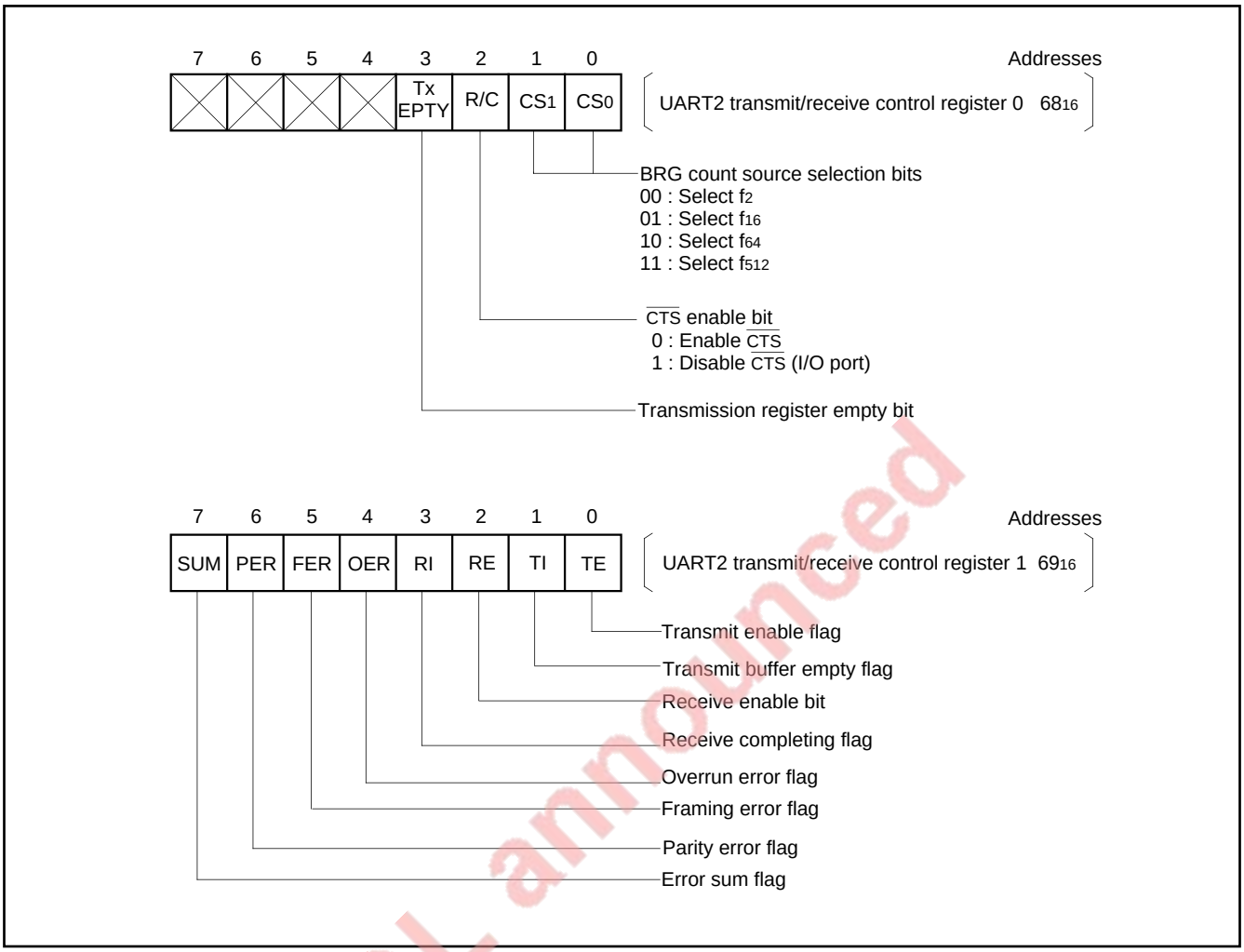


Fig. 40 UART2 transmit/receive control register bit configuration

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 41 will be described. (The transmission side will be denoted by subscript j and the receiving side will be denoted by subscript k.)

Bit 0 of the UARTj transmit/receive mode register and UARTk transmit/receive mode register must be set to "1", and bits 1 and 2 must be "0". The length of the transmission data is 8 bits.

Bit 3 of the UARTj transmit/receive mode register of the clock sending side is cleared to "0" to select the internal clock. Bit 3 of the UARTk transmit/receive mode register of the clock receiving side is set to "1" to select the external clock. Bits 4, 5 and 6 are ignored in the clock synchronous mode. Bit 7 must always be "0".

The clock source is selected by bit 0 (CS_0) and bit 1 (CS_1) of the clock sending side UARTj transmit/receive control register 0. If the contents of the bit rate genarator is n, as shown in Figure 36, the selected clock is divided by (n + 1), then by 2, passed through a transmission control circuit, and output as transmission clock CLK_j. Therefore, when the selected clock is f_i,

$$\text{Bit Rate} = f_i / \{(n + 1) \times 2\}$$

On the clock receiving side, the CS_0 and CS_1 bits are ignored because an external clock is selected.

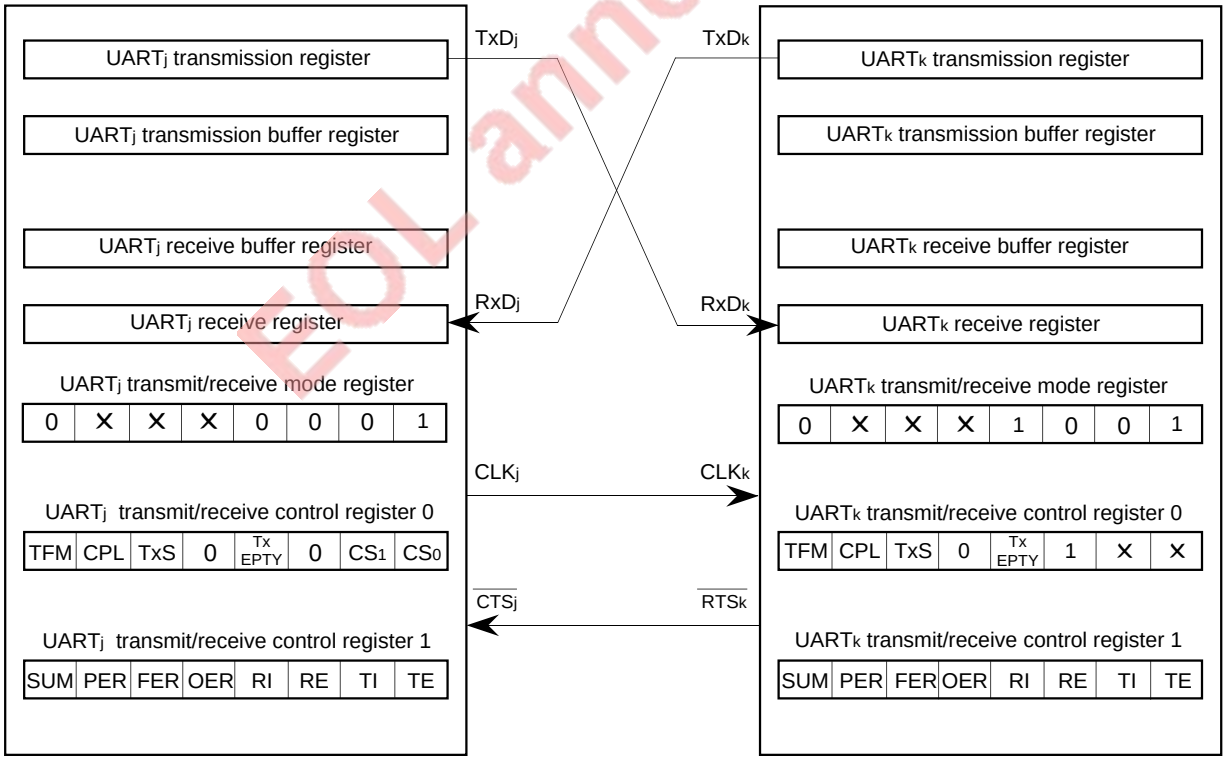
The bit 2 of the clock sending side UARTj transmit/receive control register 0 is cleared to "0" to select $\overline{CTS_j}$ input. The bit 2 of the clock receiving side is set to "1" to select $\overline{RTS_k}$ output.

Whether to use the $\overline{CTS}$ and $\overline{RTS}$ signals is determined by bit 4 of the UART transmit/receive control register 0. Set bit 4 to "0" when $\overline{CTS}$ and $\overline{RTS}$ signals are used, and to "1" when they are not used.

UART2 has the $\overline{CTS}$ input function, but that does not have the $\overline{RTS}$ output function (refer to Figure 40.)

When $\overline{CTS}$ and $\overline{RTS}$ signals are not used, the $\overline{CTS}/\overline{RTS}$ pin can be used as a normal port. The following describes the case when the $\overline{CTS}$ and $\overline{RTS}$ signals are used. If $\overline{CTS}$ and $\overline{RTS}$ signals are not used, the $\overline{CTS_j}$ input condition is unnecessary and there is no $\overline{RTS_k}$ output.

Output driver format of the transmit data output pin (TxD_j), which is the CMOS output or the N-channel open-drain output, is selected with bit 5 (TxS) of the UARTj transmit/receive control register 0. When bit 5 is "0", the CMOS output format is selected. When bit 5 is "1", the N-channel open-drain output format is selected. When the N-channel open-drain output format is selected, make sure to pull-up the data line using a pull-up resistor.



Note. UART2 does not include $\overline{RTS}$ output. The UART2 transmit/receive control register 0's bit configuration is partialy different.

Fig. 41 Clock synchronous serial communication

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

The internal/external clock polarity is selected with bit 6 (CPL) of the UART$_j$ transmit/receive control register 0. When bit 6 is "0", transmit data is output at the CLK$_j$'s falling edge in transmitting, received data is input at the CLK$_k$'s rising edge in receiving, and the CLK$_i$ level is "H" not in transferring (transmitting/receiving). When bit 6 is "1", reversely, transmit data is output at the CLK$_j$'s rising edge in transmitting, received data is input at the CLK$_k$'s falling edge in receiving, and the CLK$_i$ level is "L" not in transferring. Bit transfer order of transmit/received data, which is LSB first or MSB first (Note), is selected with bit 7 (TFM) of the UART$_j$ transmit/receive control register 0. LSB first is selected when bit 7 is "0", and MSB first is selected when bit 7 is "1". However, UART2's function is fixed to the function specified by TxS=CPL=TFM="0", and it cannot be changed. Note that, only in the UART$_0$ transmission mode, the transmission clock can be output not only from the CLK$_0$ pin but also from the other output pins (CLKS$_0$, CLKS$_1$). Transmission clock output multiple-selection mode is set with the serial transmit control register and others. For details, refer to the section on transmission.

Note. When LSB first is selected, data is transmitted/received beginning at the least significant bit (LSB). When MSB first is selected, data is transmitted/received beginning at the most significant bit (MSB).

Transmission

Transmission is started when the bit 0 (TE$_j$ flag) of the UART$_j$ transmit/receive control register 1 is "1", bit 1 (TI$_j$ flag) of one is "0", and the $\overline{\text{CTS}_j}$ input is "L".

Transmit data is output each time when the transmission clock (CLK$_j$) level changes from "H" to "L" with bit 6 (CPL) of the UART$_j$ transmit/receive control register 0 "0" or is output each time when the CLK$_j$ level changes from "L" to "H" with CPL "1". For details, refer to Figure 42. In addition, transmit data is output beginning at the least significant bit (LSB) with bit 7 (TFM) of the UART$_j$ transmit/receive control register "0" or is output beginning at the most significant bit (MSB) with TFM "1".

The TI$_j$ flag indicates whether the transmission buffer register is empty or not. It is cleared to "0" when date is written in the transmission buffer register and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.

EOL announced

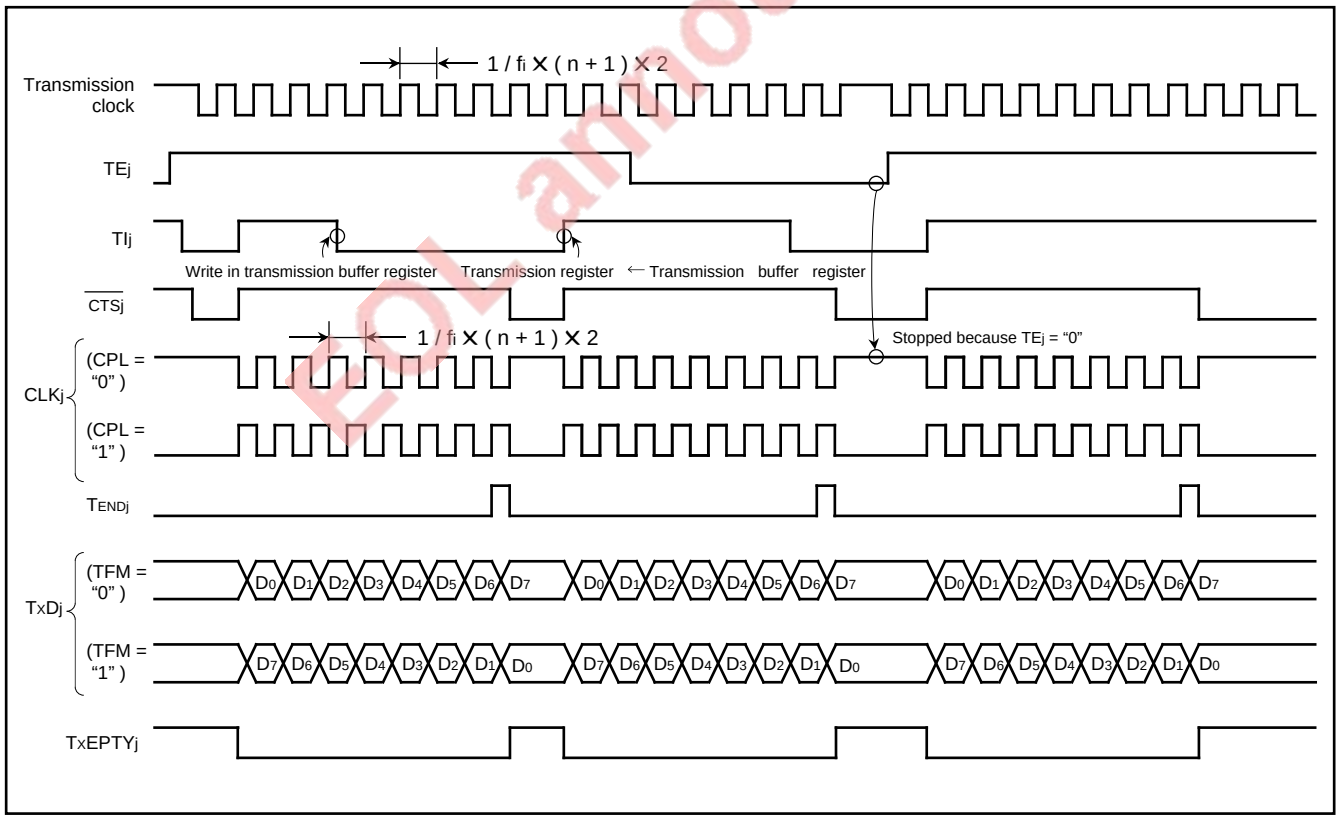


Fig. 42 Clock synchronous serial I/O timing

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

When the transmission register becomes empty after its contents has been transmitted, data is automatically transferred from the transmission buffer register to the transmission register if the next transmission start condition is satisfied. If bit 2 of the UART$_j$ transmit/receive control register 0 is "1", $\overline{CTS}_j$ input is ignored and transmission start is controlled only by the TE$_j$ flag and TI$_j$ flag. Once transmission has started, the TE$_j$ flag, TI$_j$ flag, and $\overline{CTS}_j$ signals are ignored until data transmission completes. Therefore, transmission is not interrupt even when $\overline{CTS}_j$ input is changed to "H" during transmission.

As shown in Figure 42, $\overline{CTS}_j$ and flags TE$_j$ and TI$_j$, which indicate the transmission start condition, are checked while the TEND$_j$ signal is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and the TI$_j$ flag is cleared to "0" before the TEND$_j$ signal level becomes H". The bit 3 (TxEPTY$_j$ flag) of the UART$_j$ transmit/receive control register 0 changes to "1" at the next cycle after the TEND$_j$ signal level becomes "H". Furthermore, the TxEPTY$_j$ flag changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission has been completed.

When the TI$_j$ flag changes from "0" to "1", the interrupt request bit in the UART$_j$ transmission (transmit/receive in UART2) interrupt control register is set to "1".

Since UART0 has three output pins (CLK$_0$, CLKS$_0$, and CLKS$_1$) for the transmission clock, the user can select one from these pins when using the internal clock. Accordingly, data can be transmitted to three external receive devices which will not receive data at the same time. Figure 43 shows the extrnal connection diagram example.

To select the transmission clock output multiple-selection mode, it is necessary to set bits 5 and 4 of the serial transmit control register. In addition, it is necessary to select the internal clock, to disable $\overline{CTS}$ and $\overline{RTS}$, and disable reception, with the UART0 transmit/receive mode register and the UART0 transmit/receive control register 0/1. Figure 44 shows the bit configuration of the serial transmit control register and Figure 45 shows the bit configuration of the UART0 transmit/receive mode register and the UART0 transmit/receive control register 0/1 in the transmission clock output multiple-selection mode. Furthermore, Table 6 shows the function of bits 5 and 4 (Transmission clock output pin selection bits, TC$_1$ and TC$_0$) of the serial transmit control register. As shown in Table 5, the transmission clock is output from the CLK$_0$, CLKS$_0$, or CLKS$_1$ pin depending on TC$_1$, TC$_0$. Do not change the value of TC$_1$ and TC$_0$ during transferring. The transmission clock polarity also depends on bit 6 (CPL) of the UART0 transmit/receive control register 0.

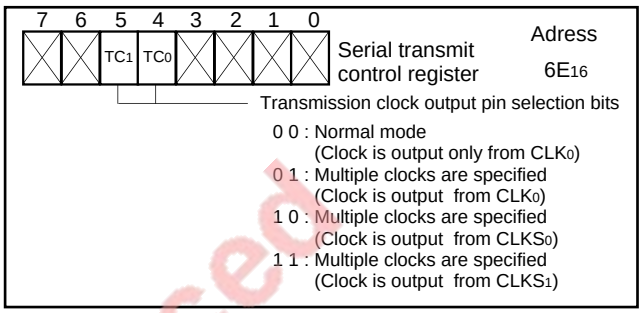


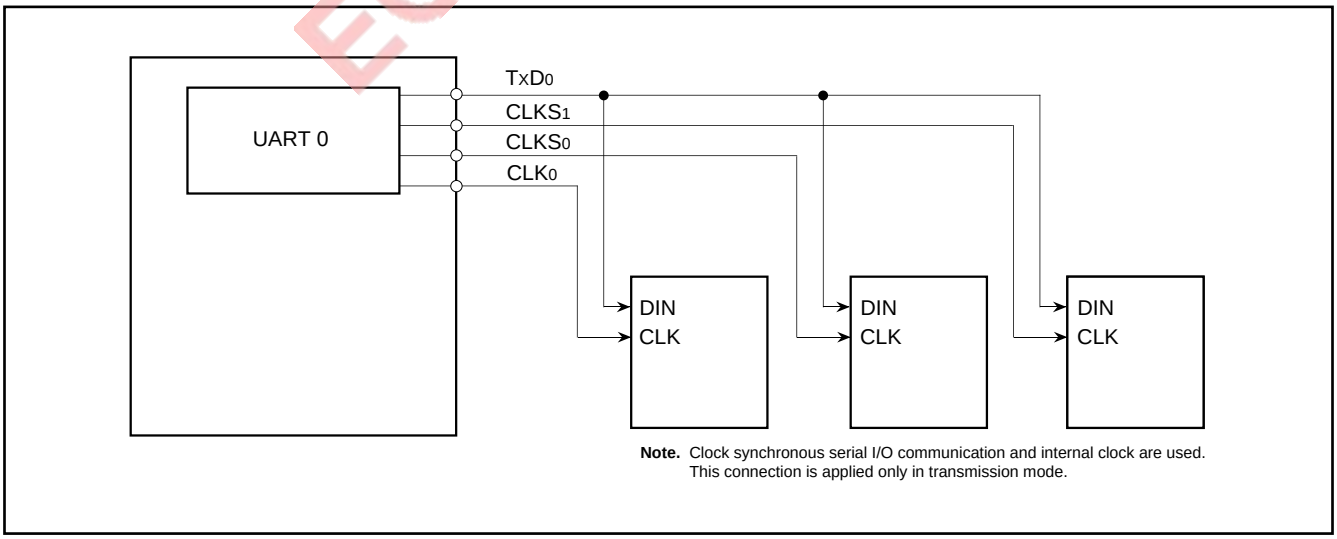
Fig. 44 Bit configuration of serial transmit control register

Table 6. Relationship between transmission clock output pin selection bits and pin functions

Transmission clock output pin selection bits		P8$_1$	P8$_2$	P8$_0$
			RxD$_0$	$\overline{CTS}_0/\overline{RTS}_0$
TC$_1$	TC$_0$	CLK$_0$	CLKS$_0$	CLKS$_1$
0	0	CLK$_0$	RxD$_0$	P8/$\overline{CTS}_0/\overline{RTS}_0$
0	1	CLK$_0$	"H" (Note2)	P8$_0$
1	0	"H"	CLKS$_0$	P8$_0$
1	1	"H"	"H" (Note2)	CLKS$_1$

Notes 1. In this table, the CLK polarity selection bit (CPL) is "0". When CPL is "1", "H" in this table becomes "L". The polarity of CLK$_0$, CLKS$_0$, or CLKS$_1$ also depends on CPL.

2. When bit 2 of the port P8 direction register is "1", "H" is output. When this bit is "0", floating is entered.



Note. Clock synchronous serial I/O communication and internal clock are used. This connection is applied only in transmission mode.

Fig. 43 External connection diagram example in the transmission clock output multiple-selection mode

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

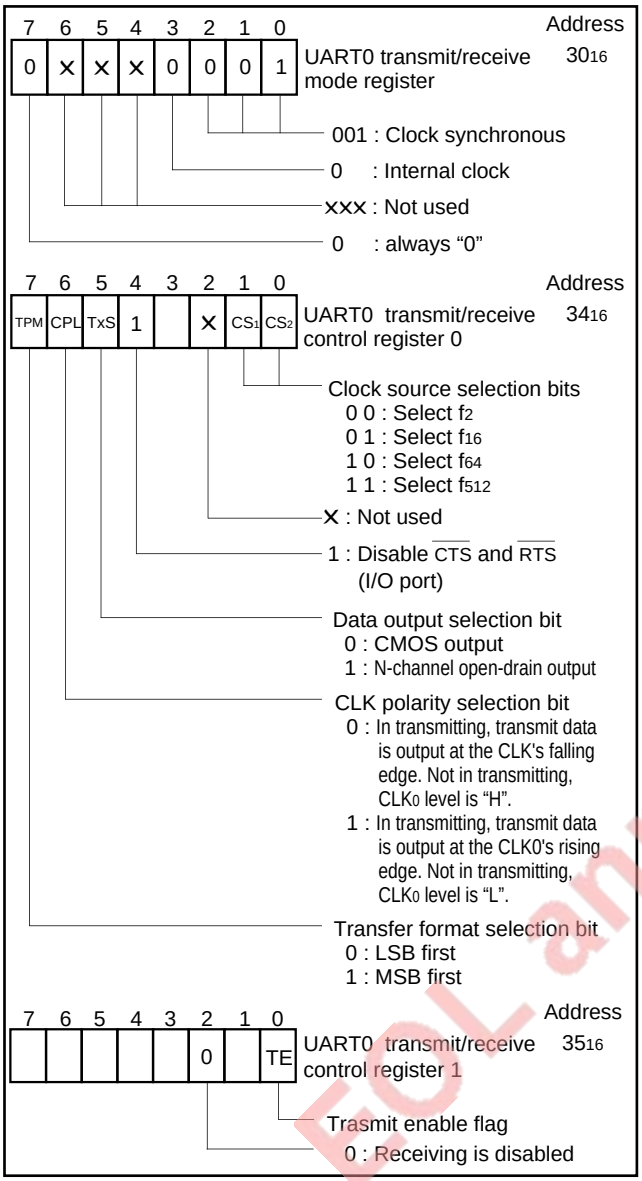


Fig. 45 Bit configuration of UART0 transmit/receive mode register and UART0 transmit/receive control register 0/1 in the transmission clock output multiple-selection mode

Receive

Receive starts when the bit 2 (REk flag) of the UARTk transmit/receive control register 1 is set to "1".

The $\overline{RTS_k}$ output level is "H" when the REk flag is "0", but it is "L" when the REk flag is "1" and the TIk flag is "0". Furthermore, the $\overline{RTS_k}$ output level is "H" again when receiving restarts. The TIk flag is cleared to "0" by writing dummy data into the transmission buffer register. When the $\overline{RTS_k}$ output level is "L", receiving for the receive register is enabled. UART2 does not have the $\overline{RTS}$ output function.

When bit 6 (CPL) of the UARTk transmit/receive control register 0 is "0", the contents of the receive register is shifted by 1 bit each time when the receive clock (CLKk) changes from "L" to "H". When CPL is "1", the contents is shifted by 1 bit each time when CLKk changes from "H" to "L". These shifts are performed simultaneously with the data reception from the RxDk pin. When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and the bit 3 (RIk flag) of the UARTk transmit/receive control register 1 is set to "1". In other words, the setting of the RIk flag to "1" indicates that the receive buffer register contains the received data. When the TIk flag goes "0", $\overline{RTS_k}$ output level goes "L" to indicate that the next data can be received. When the RIk flag changes from "0" to "1", the interrupt request bit of the UARTk receive (transmit/receive in UART2) interrupt control register is set to "1". Bit 4 (OERk flag) of the UARTk transmit/receive control register is set to "1" when the next data is transferred from the receive register to the receive buffer register while RIk flag is "1", and the OERk flag indicates that the next data was transferred to the receive buffer register before the contents of the receive buffer register was read.

The RIk flag is cleared to "0" when reading the low-order byte to the receive buffer, when writing "0" to the REk flag, or when setting to be a parallel port. The OERk flag is cleared to "0" when writing "0" to the REk flag or when setting to be a parallel port. The FERk, PERk, and SUMk flags are ineffective in the clock synchronous communication.

The received data in the receive buffer register is read into the data bus according to the LSB first (beginning at the least significant bit) when bit 7 (TEM) of the UARTk transmit/receive control register 0 is "0" or according to the MSB first (beginning at the most significant bit) when bit 7 is "1".

As shown in Figure 36, with clock synchronous serial communication, data cannot be received unless the transmitter is operating because the receive clock is created from the transmission clock. Therefore, the transmitter must be operating even when there is no data to be sent from UARTk to UARTj.

EOL announced

ASYNCHRONOUS SERIAL COMMUNICATION (UART)

Asynchronous serial communication can be performed using 7-, 8-, or 9-bit length data. The operation is the same for all data lengths. The following is the description for 8-bit asynchronous communication.

With 8-bit asynchronous communication, the bits 2 to 0 of the UARTi transmit/receive mode register must be “101”.

Bit 3 is used to select an internal clock or an external clock. If bit 3 is “0”, an internal clock is selected and if bit 3 is “1”, then external clock is selected. If an internal clock is selected, the bit 0 (CS_0) and bit 1 (CS_1) of UARTi transmit/receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLKi pin can be used as a normal port.

If the content of the bit rate generator is n, the selected internal or external clock is divided by (n + 1), then by 16, and passed through a control circuit to create the UART transmission clock or the UART receive clock.

If the selected clock is an internal clock fi or an external clock f_{EXT},

$$\text{Bit Rate} = (\text{fi or } f_{EXT}) / \{(n + 1) \times 16\}$$

Bit 4 selects 1 stop bit or 2 stop bits.

The bit 5 is a selection bit of odd parity or even parity.

In the odd parity mode, the parity bit is adjusted so that the sum of the 1’s in the data and parity bit is always odd.

In the even parity mode, the parity bit is adjusted so that the sum of the 1’s in the data and parity bit is always even.

EOL announced

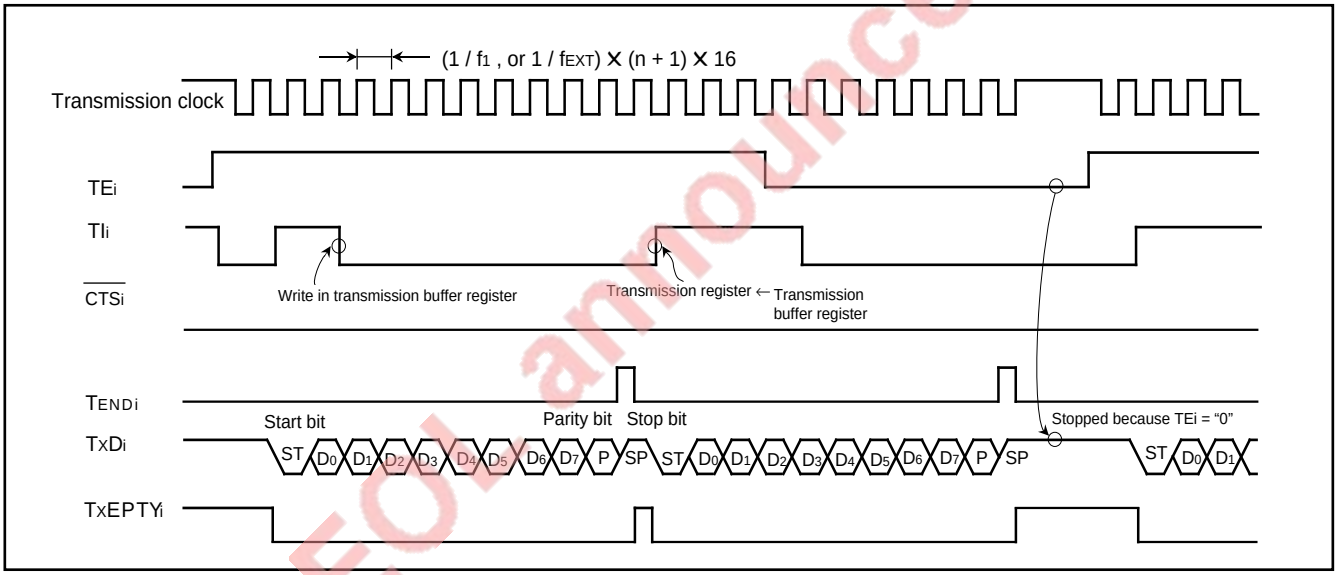


Fig. 46 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit is selected

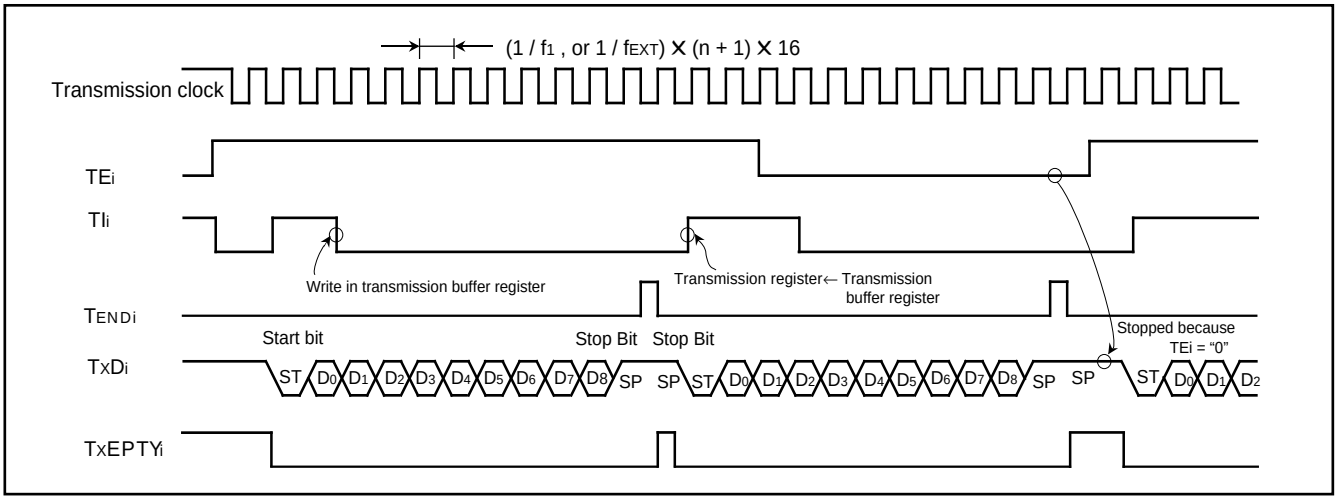


Fig. 47 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits is selected

Bit 6 is the parity enable bit which indicates whether to add parity bit or not.
Bits 4 to 6 should be set or reset according to the data format of the communicating devices.
Bit 7 is the sleep selection bit (refer to the next page).
Bit 2 of the UART$_i$ transmit/receive control register 0 is used to determine whether to use $\overline{CTS}_i$ input or $\overline{RTS}_i$ output. $\overline{CTS}_i$ input is used if bit 2 is "0" and $\overline{RTS}_i$ output is used if bit 2 is "1".
If $\overline{CTS}_i$ input is selected, the user can control whether to stop or start transmission with external $\overline{CTS}_i$ input.
Whether to use $\overline{CTS}$ and $\overline{RTS}$ signals is determined by bit 4 of the UART transmit/receive control register 0. Set bit 4 to "0" when $\overline{CTS}$ and $\overline{RTS}$ signals are used, and to "1" when they are not used.
UART2 has the $\overline{CTS}$ input function, but that does not have the $\overline{RTS}$ output function (refer to Figure 40.)
When $\overline{CTS}$ and $\overline{RTS}$ signals are not used, the $\overline{CTS}/\overline{RTS}$ pin can be used as a normal port. The following describes the case when the $\overline{CTS}$ and $\overline{RTS}$ signals are used. If $\overline{CTS}$ and $\overline{RTS}$ signals are not used, the $\overline{CTS}_i$ input condition is unnecessary and there is no $\overline{RTS}_i$ output.
In addition, output driver format of the transmission data output pin (TxD$_j$), which is CMOS output or N-channel open-drain output, is selected with bit 5 (TxS) of the UART$_j$ transmit/receive control register 0. CMOS output format is selected when bit 5 is "0", and N-channel open-drain output format is selected when bit 5 is "1". When N-channel open-drain output format is selected, make sure to pull-up the data line using a pull-up resistor.
However, UART2 does not have bit 5 (TxS) and the format is always CMOS output.
In asynchronous serial communication, bits 6 and 7 of the UART$_j$ transmit/receive control register 0 must be "0".

Transmission

Transmission is started when the bit 0 (TE$_i$ flag) of UART$_i$ transmit/receive control register 1 is "1", the bit 1 (TI$_i$ flag) is "0", and $\overline{CTS}_i$ input is "L" if $\overline{CTS}_i$ input is selected. As shown in Figures 46 and 47, data is output from the TxD$_i$ pin with the start bit and the stop bit or parity bit specified by the bits 4 to 6 of UART$_i$ transmit/receive mode register. The data is output beginning at the least significant bit.
The TI$_i$ flag indicates whether the transmission butter is empty or not. It is cleared to "0" when data is written in the transmission buffer and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.
When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically from the transmission buffer register to the transmission register if the next transmission start condition is satisfied.
Once transmission has started, the TE$_i$ flag, TI$_i$ flag, and $\overline{CTS}_i$ signal (if $\overline{CTS}_i$ input is selected) are ignored until data transmission is completed.
Therefore, transmission does not stop until it completes even if the TE$_i$ flag is cleared during transmission.
As shown in Figure 46, $\overline{CTS}_i$ input and flags TE$_i$ and TI$_i$, which indicate the transmission start condition, are checked while the TEND$_i$ signal is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and TI$_i$ flag is cleared to 0 before the TEND$_i$ signal goes "H".
The bit 3 (TxEPTY$_i$ flag) of the UART$_i$ transmit/receive control register 0 changes to "1" at the next cycle after the TEND$_i$ signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission is completed.
When the TIi flag changes from "0" to "1", the interrupt request bit of the UART$_i$ transmission (transmit/receive in UART2) interrupt control register is set to "1".

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Receive

Receive is enabled when bit 2 (REi flag) of the UARTi transmit/receive control register 1 is set to "1". As shown in Figure 48, the frequency divider circuit at the receiving end begin to work when a start bit is arrived and the data is received.

If $\overline{RTS_i}$ output is selected by setting bit 2 of the UARTi transmit/receive control register 0 to "1", the $\overline{RTS_i}$ output is "H" when the REi flag is "0". When the REi flag changes to "1", the $\overline{RTS_i}$ output goes "L" to indicate receive ready and returns to "H" once receive has started. In other words, $\overline{RTS_i}$ output can be used to determine externally whether the receive register is ready to receive. (UART2 does not have the $\overline{RTS}$ output function.)

The entire transmission data bits are received when the start bit passes the final bit of the receive register of the receive block shown in Figure 38. At this point, the contents of the receive register is transferred to the receive buffer register and the bit 3 of the UARTi transmit/receive control register 1 (RIi flag) is set. In other words, the RIi flag indicates that the receive buffer register contains data when it is set. If $\overline{RTS_i}$ output is selected, $\overline{RTS_i}$ output goes "L" to indicate that the register is ready to receive the next data.

The interrupt request bit of the UARTi receive (transmit/receive in UART2) interrupt control register is set when the RIi flag changes from "0" to "1".

The bit 4 (OERi flag) of the UARTi transmission control register 1 is set when the next data is transferred from the receive register to the receive buffer register while the RIi flag is "1". In other words when an overrun error occurs. If the OERi flag is "1", it indicates that the next data has been transferred to the receive buffer register before the contents of the receive butter register has been read.

Bit 5 (FERi flag) is set when the number of stop bits is less than required (framing error).

Bit 6 (PERi flag) is set when a parity error occurs.

Bit 7 (SUMi flag) is set when either the OERi flag, FERi flag, or the PERi flag is set. Therefore, the SUMi flag can be used to determine whether there is an error.

The setting of the RIi flag, OERi flag, FERi flag, and the PERi flag is performed while transferring the contents of the receive register to the receive buffer register. The RIi, FERi, and PERi flags are cleared when reading the low-order byte of the receive buffer register or when writing "0" to the REi flag or when setting to be a parallel port. The OERi and SUMi flags are cleared when writing "0" to the REi flag or when the setting to be a parallel port.

Sleep mode

The sleep mode is used to communicate only between certain microcomputers when multiple microcomputers are connected through serial I/O.

The sleep mode is entered when bit 7 of the UARTi transmit/receive mode register is set.

UART2 does not have the sleep mode.

The operation of the sleep mode for an 8-bit asynchronous communication is described below.

When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asychronous communication) of the received data is "0". Also the RIi, OERi, FERi, PERi, and the SUMi flag are unchanged. Therefore, the interrupt request bit of the UARTi receive interrupt control register is also unchanged.

Normal receive operation takes place when bit 7 of the received data is "1".

The following is an example of how the sleep mode can be used.

The main microcomputer first sends data with bit 7 set to "1" and bits 0 to 6 set to the address of the subordinate microcomputer which wants to communicate with. Then all subordinate microcomputers receive the same data. Each subordinate microcomputer checks the received data, clears the sleep function selection bit if bits 0 to 6 are its own address and sets the sleep bit if not. Next the main microcomputer sends data with bit 7 cleared. Then the microcomputer with the sleep bit cleared will receive the data, but the microcomputer with the sleep bit set will not. In this way, the main microcomputer is able to communicate only with the designated microcomputer.

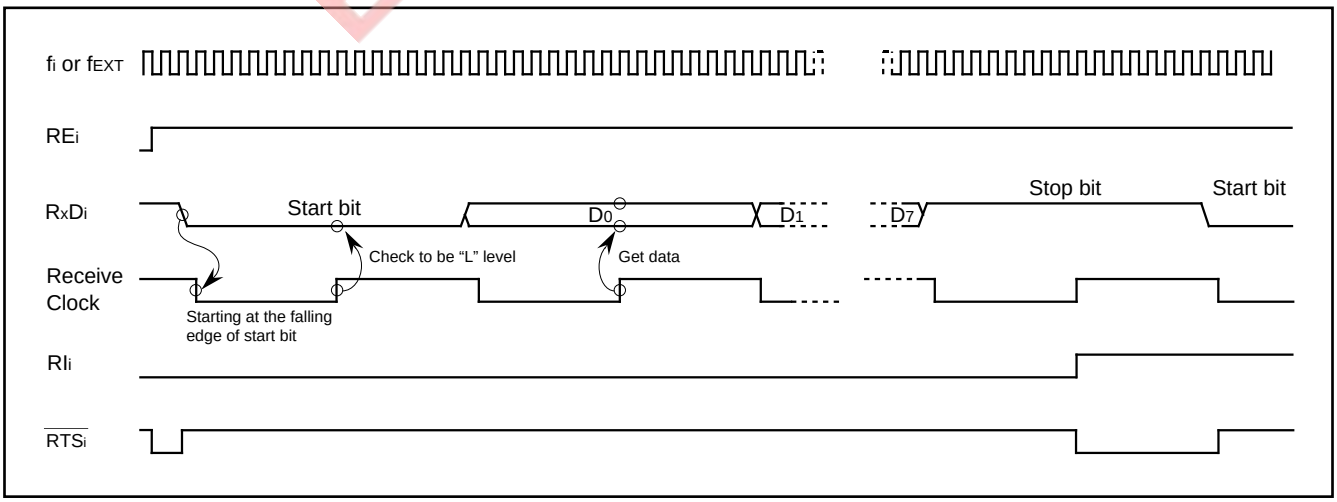


Fig. 48 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit is selected

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

A-D CONVERTER

The A-D converter is an 10-bit successive approximation converter. Figure 49 shows a block diagram of the A-D converter and Figure 50 shows the configuration of the A-D control register 0 (address $1E_{16}$) and A-D control register 1 (address $1F_{16}$).

The frequency of the A-D converter operating clock ϕ_{AD} is selected by bit 7 of the A-D control register 0. When bit 7 is "0", ϕ_{AD} is the clock frequency divided by 4. That is, $\phi_{AD} = f_2/4$. When bit 7 is "1", ϕ_{AD} is the clock frequency divided by 2 and $\phi_{AD} = f_2/2$.

The ϕ_{AD} during A-D conversion must be 250 kHz or more because the comparator uses a capacity coupling amplifier.

Bit 3 of A-D control register 1 is used to select whether to use the conversion result as 10 bits or as 8 bits. The conversion result is used as 10 bits when bit 3 is "1" and as 8 bits when bit 3 is "0".

When the conversion result is used as 10 bits, the low-order 8 bits of the conversion result is stored in the even address of the corresponding A-D register and the high-order two bits are stored in bits 0 and 1 of the odd address of the corresponding A-D register. Bits 2 to 7 of the A-D register odd address return "000000_2" when read.

When the conversion result is used as 8 bits, the high-order 8 bits of the 10-bit A-D conversion are stored in even address of the corresponding A-D register. In this case, the A-D register odd address returns "00_{16}" when read.

The operating mode is selected by bits 3 and 4 of A-D control register 0. The available operating modes are one-shot, repeat, single sweep, repeat sweep.

Whether to connect the reference voltage input pin (V_{REF}) with the ladder network or not depends on bit 5 of the A-D control register 1. The V_{REF} pin is connected when bit 5 is "0" and is disconnected when bit 5 is "1" (High impedance state). When A-D conversion is not performed, current from the V_{REF} pin to the ladder network can be cut off by disconnecting ladder network from the V_{REF} pin.

Before starting A-D conversion, wait for 1 μs or more after clearing bit 5 to "0".

The bit of the port direction register corresponding to the analog input pin to be used must be "0" (input mode) because the analog input pin is also used as port P7.

Note that when using the sub-clock (X_{CIN} - X_{COUT}) or UART2, the analog pins shared with those functions cannot be used.

The operation of each mode is described below.

The interrupt vector and the interrupt control register are common to the A-D conversion interrupt and UART2 transmit/receive interrupt. It is switched by a selection of UART2 function as shown in Figure 37's note.

ROLannounced

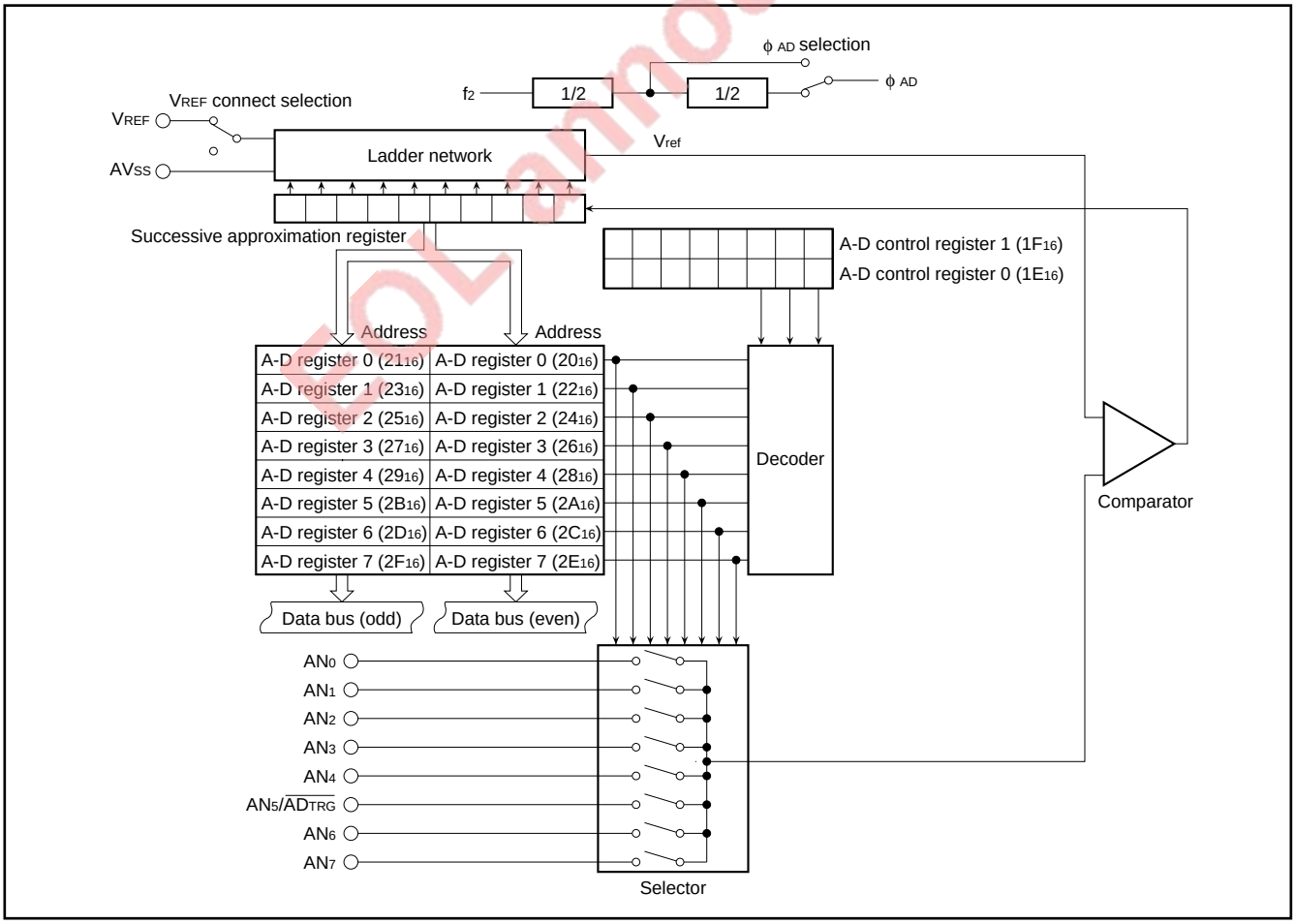


Fig. 49 A-D converter block diagram

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

(1) One-shot mode

One-shot mode is selected when bits 3 and 4 of A-D control register 0 are “0”. The analog input pin (AN_0 – AN_7) is selected with bits 0 to 2 of A-D control register 0. A-D conversion can be started by a software trigger or by an external trigger.

A software trigger is selected when bit 5 of A-D control register 0 is “0” and an external trigger is selected when it is “1”. When a software trigger is selected, A-D conversion is started when bit 6 (A-D conversion start flag) is set to “1”. A-D conversion ends after 59 ϕ_{AD} cycles and an interrupt request bit of the A-D conversion interrupt control register is set to “1”. At the same time, the A-D conversion start flag (bit 6 of the A-D control register 0) is cleared and A-D conversion stops. The result of A-D conversion is stored in the A-D register corresponding to the selected pin.

If an external trigger is selected, A-D conversion starts when the A-D conversion start flag is “1” and the $\overline{ADTRG}$ input changes from “H” to “L”. In this case, the pins that can be used for A-D conversion are AN_0 to AN_4, AN_6 and AN_7 (a total of 7) because the $\overline{ADTRG}$ pin is also used as the analog voltage input pin (AN_5). The operation is the same as with software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(2) Repeat mode

Repeat mode is selected when bit 3 of A-D control register 0 is “1” and bit 4 is “0”. The operation of this mode is the same as the operation of one-shot mode except that when A-D conversion of the selected pin is complete and the result is stored in the A-D register, conversion does not stop, but is repeated.

No interrupt request is issued in this mode. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared.

The contents of the A-D register can be read at any time.

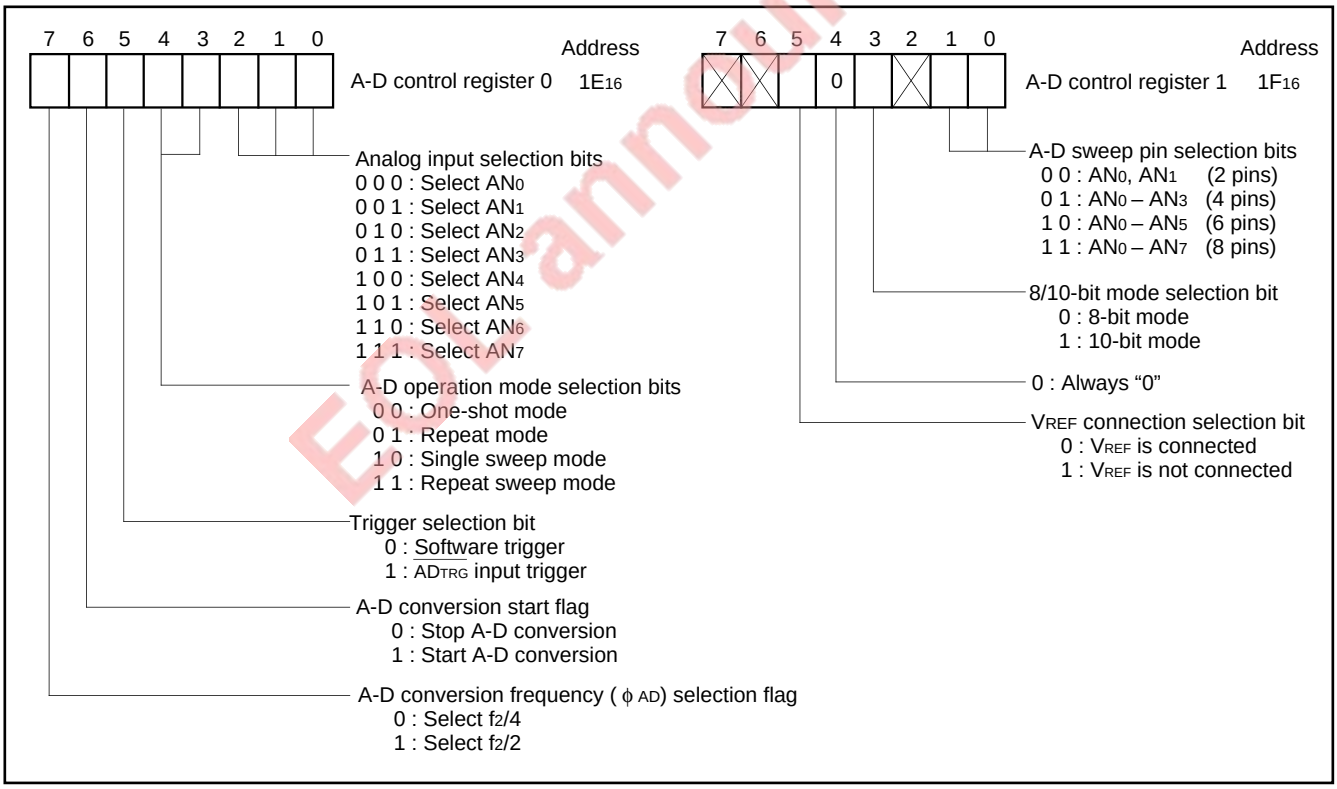


Fig. 50 A-D control register bit configuration

Under Announced

(3) Single sweep mode

Single sweep mode is selected when bit 3 of A-D control register 0 is "0" and bit 4 is "1".

In the single sweep mode, the number of analog input pins to be swept can be selected. Analog input pins are selected by bits 1 and 0 of the A-D control register 1 (address $1F_{16}$). Two pins, four pins, six pins or eight pins can be selected as analog input pins, depending on the contents of these bits.

A-D conversion is performed only for selected input pins. After A-D conversion is performed for input of AN_0 pin, the conversion result is stored in A-D register 0, and in the same way, A-D conversion is performed for selected pins one after another. After A-D conversion is performed for all selected pins, the sweep is stopped.

A-D conversion can be started with a software trigger or with an external trigger input. A software trigger is selected when bit 5 is "0" and an external trigger is selected when it is "1".

When a software trigger is selected, A-D conversion is started when A-D control register 0 bit 6 (A-D conversion start flag) is set to "1". When A-D conversion of all selected pins ends, an interrupt request bit of the A-D conversion interrupt control register is set to "1". At the same time, A-D conversion start flag is cleared and A-D conversion stops.

If an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{ADTRG}$ input changes from "H" to "L". In this case, the A-D conversion result which is stored in the A-D register 5 becomes invalid because the $\overline{ADTRG}$ pin is also used as the AN_5 pin.

The operation by external trigger is the same as that done by software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(4) Repeat sweep mode

Repeat sweep mode 0 is selected when bit 3 of A-D control register 0 is "1" and bit 4 is "1".

The difference from the single sweep mode is that A-D conversion does not stop after converting from the AN_0 pin to the selected pins, but repeats again from the AN_0 pin. The repeat is performed among the selected pins. Also, no interrupt request is generated. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared. The A-D register can be read at any time.

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software runaway.

Figure 51 shows a block diagram of the watchdog timer. The watchdog timer includes a 12-bit binary counter.

The watchdog timer counts divided clock f_{32} or f_{512}. Whether to count f_{32} or f_{512} is determined by the watchdog timer frequency selection flag shown in Figure 52. For divided clocks f_{32} and f_{512}, refer to the section on clock generating circuit. f_{512} is selected when the flag is "0" and f_{32} is selected when it is "1". The flag is cleared after reset. "FFF_{16}" is set in the watchdog timer when "L" or 2 Vcc is applied to the $\overline{\text{RESET}}$ pin, STP instruction is executed, data is written to the watchdog timer register, or the most significant bit of the watchdog timer becomes "0".

After "FFF_{16}" is set in the watchdog timer, the contents of the watchdog timer is decremented by one at every cycle of f_{32} or f_{512}. After 2048 counts, the most significant bit of the watchdog timer becomes "0", and a watchdog timer interrupt request bit is set, and "FFF_{16}" is set in the watchdog timer.

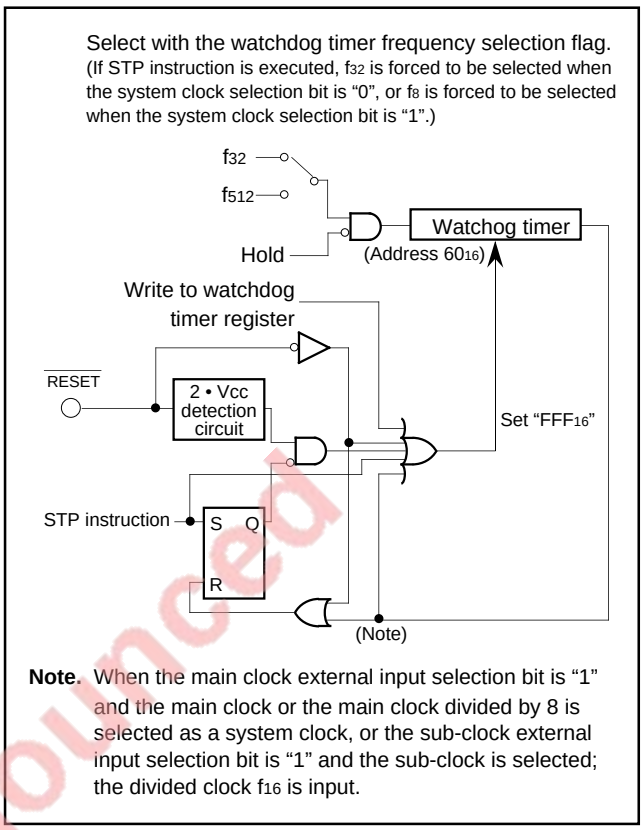
Normally, a program is written so that data is written in the watchdog timer register before the most significant bit of the watchdog timer becomes "0". If this routine is not executed due to unexpected program runaway, the most significant bit of the watchdog timer becomes eventually "0" and an interrupt is generated.

The processor can be reset by setting "1" to the software reset bit (bit 3 of the processor mode register 0) described in Figure 10 on the interrupt section and generating a reset pulse.

The watchdog timer stops its function when the $\overline{\text{RESET}}$ pin voltage is raised to double the Vcc voltage.

The watchdog timer can also be used to recover from when the clock is stopped by the STP instruction. Refer to the section on stand-by function for more details.

The watchdog timer hold the contents during a hold state and the input of the divided clock is stopped.



Note. When the main clock external input selection bit is "1" and the main clock or the main clock divided by 8 is selected as a system clock, or the sub-clock external input selection bit is "1" and the sub-clock is selected; the divided clock f_{16} is input.

Fig. 51 Watchdog timer block diagram

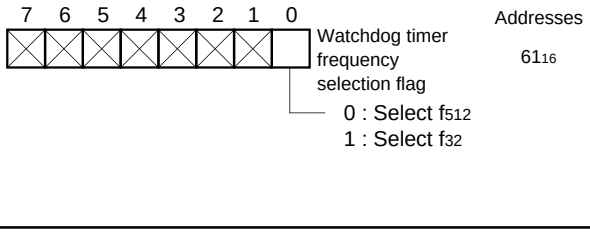


Fig. 52 Watchdog timer frequency selection flag

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

RESET CIRCUIT

The microcomputer is released from the reset state when the $\overline{\text{RESET}}$ pin is returned to "H" level after holding it at "L" level with the power source voltage at 5 V ± 10%. Program execution starts at the address formed by setting address A_{23} – A_{16} to 00_{16}, A_{15} – A_8 to the contents of address $FFFF_{16}$, and A_7 – A_0 to the contents of address $FFFE_{16}$. Figure 53 shows the status of the internal registers during reset.

	Address	
Port P0 direction register	(04₁₆)•••	00₁₆
Port P1 direction register	(05₁₆)•••	00₁₆
Port P2 direction register	(08₁₆)•••	00₁₆
Port P3 direction register	(09₁₆)•••	X X X X 0 0 0 0
Port P4 direction register	(0C₁₆)•••	00₁₆
Port P5 direction register	(0D₁₆)•••	00₁₆
Port P6 direction register	(10₁₆)•••	00₁₆
Port P7 direction register	(11₁₆)•••	00₁₆
Port P8 direction register	(14₁₆)•••	00₁₆
A-D control register 0	(1E₁₆)•••	0 0 0 0 0 ? ? ?
A-D control register 1	(1F₁₆)•••	X X 0 0 0 X 1 1
UART 0 Transmit/Receive mode register	(30₁₆)•••	00₁₆
UART 1 Transmit/Receive mode register	(38₁₆)•••	00₁₆
UART 0 transmit/receive control register 0	(34₁₆)•••	0 0 0 0 1 0 0 0
UART 1 transmit/receive control register 0	(3C₁₆)•••	0 0 0 0 1 0 0 0
UART 0 transmit/receive control register 1	(35₁₆)•••	0 0 0 0 0 0 1 0
UART 1 transmit/receive control register 1	(3D₁₆)•••	0 0 0 0 0 0 1 0
Count start flag	(40₁₆)•••	00₁₆
One- shot start flag	(42₁₆)•••	X X 0 0 0 0 0
Up-down flag	(44₁₆)•••	00₁₆
Timer A0 mode register	(56₁₆)•••	00₁₆
Timer A1 mode register	(57₁₆)•••	00₁₆
Timer A2 mode register	(58₁₆)•••	00₁₆
Timer A3 mode register	(59₁₆)•••	00₁₆
Timer A4 mode register	(5A₁₆)•••	00₁₆
Timer B0 mode register	(5B₁₆)•••	0 0 1 0 0 0 0 0
Timer B1 mode register	(5C₁₆)•••	0 0 1 X 0 0 0 0
Timer B2 mode register	(5D₁₆)•••	0 0 1 X 0 0 0 0
Processor mode register 0	(5E₁₆)•••	00₁₆
Processor mode register 1	(5F₁₆)•••	X X X X X X X 0
Watchdog timer register	(60₁₆)•••	FFF₁₆

	Address	
Watchdog timer frequency selection flag	(61₁₆)•••	X X X X X X X 0
Memory allocation control	(63₁₆)•••	X X X 0 0 0 0 0
UART2 transmit/receive mode register	(64₁₆)•••	X 0 0 0 0 0 0 0
UART2 transmit/receive control register 0	(68₁₆)•••	X X X X 1 0 0 0
UART2 transmit/receive control register 1	(69₁₆)•••	0 0 0 0 0 0 1 0
Oscillation circuit control register 0	(6C₁₆)•••	X 0 0 0 0 0 X 1
Port function control register	(6D₁₆)•••	00₁₆
Serial transmit control register	(6E₁₆)•••	X X 0 0 X X X X
Oscillation circuit control register 1	(6F₁₆)•••	0 X X 0 0 0 0 0
A-D/UART2 trans./rece. interrupt control register	(70₁₆)•••	X X X X 0 0 0 0
UART 0 transmission interrupt control register	(71₁₆)•••	X X X X 0 0 0 0
UART 0 receive interrupt control register	(72₁₆)•••	X X X X 0 0 0 0
UART 1 transmission interrupt control register	(73₁₆)•••	X X X X 0 0 0 0
UART 1 receive interrupt control register	(74₁₆)•••	X X X X 0 0 0 0
Timer A0 interrupt control register	(75₁₆)•••	X X X X 0 0 0 0
Timer A1 interrupt control register	(76₁₆)•••	X X X X 0 0 0 0
Timer A2 interrupt control register	(77₁₆)•••	X X X X 0 0 0 0
Timer A3 interrupt control register	(78₁₆)•••	X X X X 0 0 0 0
Timer A4 interrupt control register	(79₁₆)•••	X X X X 0 0 0 0
Timer B0 interrupt control register	(7A₁₆)•••	X X X X 0 0 0 0
Timer B1 interrupt control register	(7B₁₆)•••	X X X X 0 0 0 0
Timer B2 interrupt control register	(7C₁₆)•••	X X X X 0 0 0 0
$\overline{\text{INT}}_0$ interrupt control register	(7D₁₆)•••	X X 0 0 0 0 0 0
$\overline{\text{INT}}_1$ interrupt control register	(7E₁₆)•••	X X 0 0 0 0 0 0
$\overline{\text{INT}}_2$/Key input interrupt control register	(7F₁₆)•••	X X 0 0 0 0 0 0
Processor status register (PS)		0 0 0 ? ? 0 0 0 1 ? ?
Program bank register (PG)		00₁₆
Program counter (PC_H)		Content of FFFF₁₆
Program counter (PC_L)		Content of FFFE₁₆
Direct page register (DPR)		0000₁₆
Data bank register (DT)		00₁₆

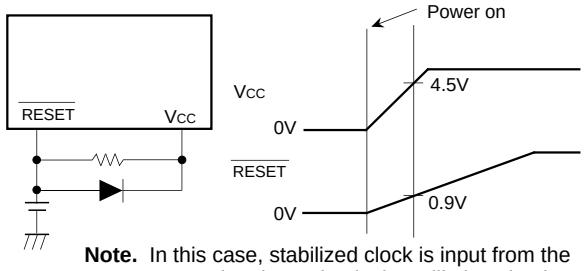
Contents of other registers and RAM are undefined during reset. Initialize them by software.

Fig. 53 Microcomputer internal status during reset

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Figure 54 shows an example of a reset circuit. If the stabilized clock is input from the external to the main-clock oscillation circuit, the reset input voltage must be 0.9 V or less when the power source voltage reaches 4.5 V. If a resonator/oscillator is connected to the main-clock oscillation circuit, change the reset input voltage from “L” to “H” after the main-clock oscillation is fully stabilized.



Note. In this case, stabilized clock is input from the external to the main-clock oscillation circuit. Perform careful evaluation at the system design level before using.

Fig. 54 Example of a reset circuit

INPUT / OUTPUT PINS

Ports P0 to P8 all have a port direction register and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding bit of the port direction register is set to “1” and an input pin when the bit is cleared to “0”.

When a pin is programmed for output, the data is written to the port latch and is output, and the contents of the port latch is read instead of the value of the pin. Therefore, a previously output value can be read correctly even when the output “L” voltage is raised by directly driving an LED or others.

A pin programmed for input is floating and the value input to the pin can be read. When a pin is programmed for input, the data is written only in the port latch and the pin retains floating.

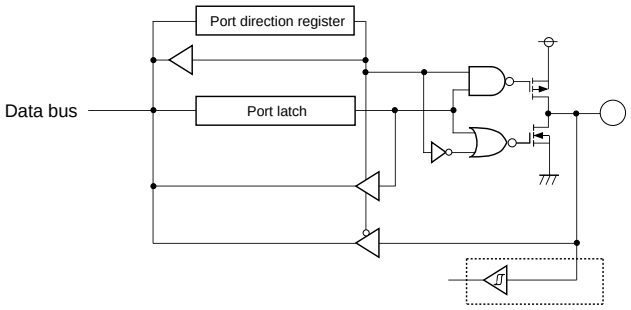
Ports $P5_4$ to $P5_7$, and $P6_2$ – $P6_4$, however, have pull-up transistors and the port’s pull-up function can be selected by setting “1” to bits 6, 5, 3 of the port function control register (reffer to Figure 11.) A port which corresponds to a port direction register’s bit set to “0” is pulled up. A port which corresponds to a bit set to “1” is an output pin and it is not pulled up.

Figures 55 and 56 show the block diagram of ports P0 to P8 and the E pin output format.

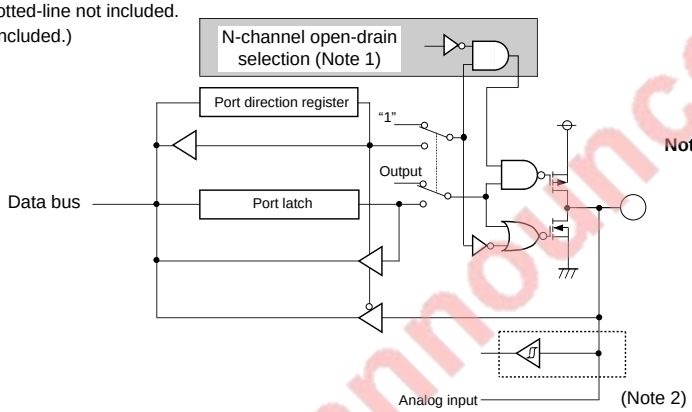
In the memory expansion mode and the microprocessor mode, ports P0 to P4 are also used as address, data, and control signal pins. Refer to the section on the processor modes for more details.

EOL announced

• Ports $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_0$ – $P3_3$, $P4_3$ – $P4_6$ (Inside dotted-line not included)
Ports $P4_0$, $P4_1$, $P4_7$, $P5_1$, $P5_3$, $P6_1$, $P6_5$ – $P6_7$, $P8_6$ (Inside dotted-line included)

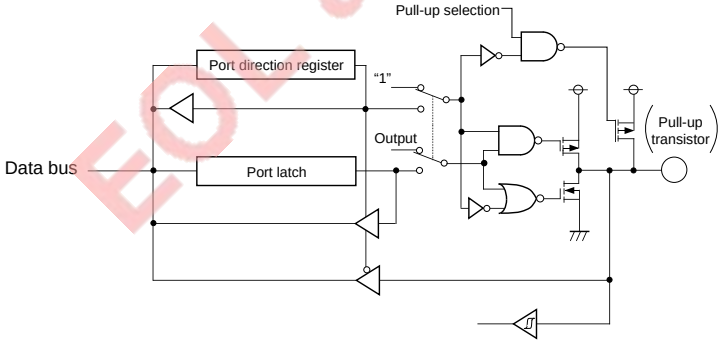


• Ports $P8_3$, $P8_7$ (Inside dotted-line not included. Shaded area included.)
Ports $P5_0$, $P5_2$, $P6_0$, $P7_5$, $P8_2$ (Inside dotted-line included. Shaded area not included.)
Port $P4_2$ (Inside dotted-line not included.
Shaded area not included.)

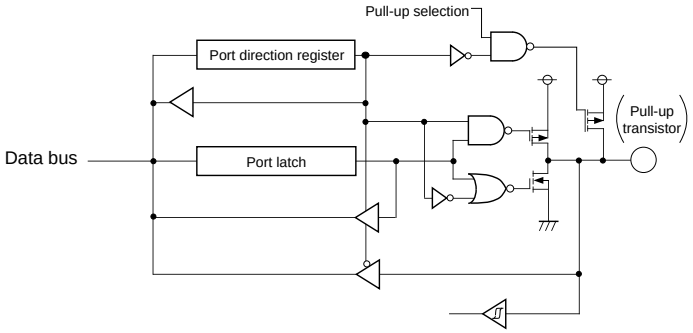


Notes 1. Valid only when pins are used as TxDj pins for serial I/O communication
2. Only $P7_5$ as analog input

• Ports $P5_4$, $P5_6$



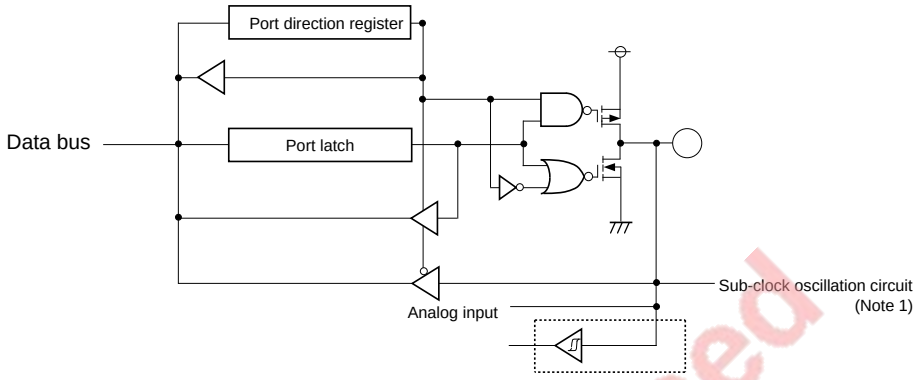
• Ports $P5_5$, $P5_7$, $P6_2$ – $P6_4$



EOL announced

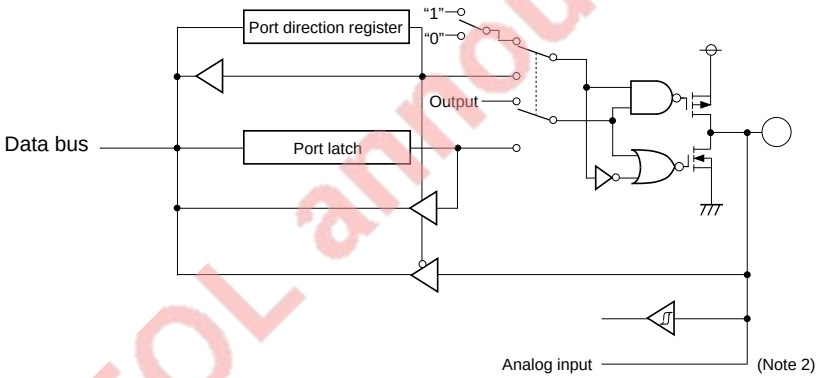
Fig. 55 Block diagram for ports P0 to P8 and the $\overline{E}$ pin output format (1)

• Ports $P7_0$, $P7_1$, $P7_6$, $P7_7$ (Inside dotted-line not included.)
Ports $P7_2$, $P7_4$ (Inside dotted-line included.)



Note 1. Only $P7_6$, $P7_7$ as sub-clock oscillation circuit

• Ports $P5_5$, $P7_3$, $P8_0$, $P8_1$, $P8_4$



Note 2. Only $P7_3$ as analog input

• $\overline{E}/\overline{RDE}$

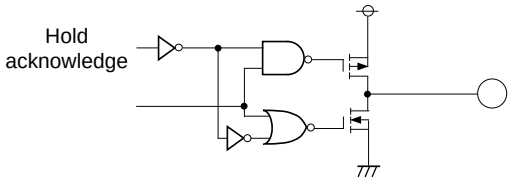


Fig. 56 Block diagram for ports P0 to P8 and the $\overline{E}$ pin output format (2)

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

PROCESSOR MODE

The bits 0 and 1 of processor mode register 0 shown in Figure 57 are used to select any mode of the single-chip mode, the memory expansion mode, and the microprocessor mode.

Ports P0 to P3 and a part of port P4 are used as I/O pins of address, data, and control signals except in the single-chip mode.

Figure 58 shows the functions of ports P4 to P0 in each mode.

The external memory area changes when the mode changes.

Figure 59 shows the memory map for each mode. Refer to Figure 1 for the addresses of RAM and ROM. The external memory area can be accessed except in the single-chip mode. The accessing of the external memory is affected by the BYTE pin, the wait bit (bit 2 of the processor mode register 0), and the wait selection bit (bit 0 of the processor mode register 1) .These will be described next.

• BYTE pin

When accessing the external memory, the level of the BYTE pin is used to determine whether to use the data bus as 8-bit width or 16-bit width.

The data bus has a width of 8 bits when level of the BYTE pin is “H”, and port P2 becomes the data I/O pin.

The data bus has a width of 16 bits when the level of the BYTE pin is “L”, and ports P1 and P2 become the data I/O pins.

When accessing the internal memory, the data bus always has a width of 16 bits regardless of the BYTE pin level.

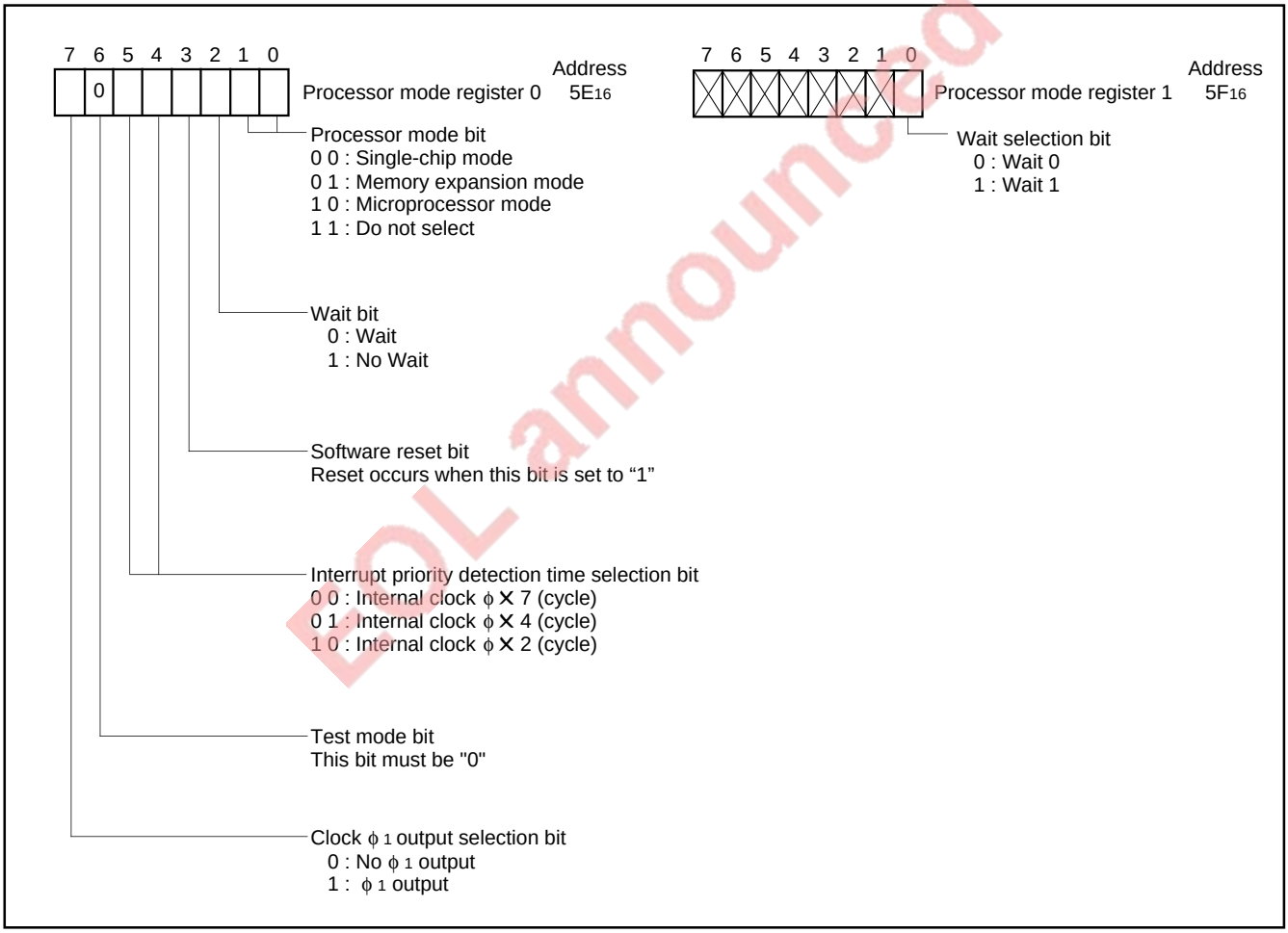


Fig. 57 Processor mode register bit configuration

EOL announced

Port \ Mode	Single-chip mode	Memory expansion mode (Note 1)	Microprocessor mode (Note 1)
PM1	0	0	1
PM0	0	1	0
$\overline{E}$/$\overline{RDE}$	$\overline{E}$ (Note 2)	$\overline{E}$, $\overline{RDE}$ (Note 2)	Same as left
Port P0	$\overline{E}$, P00 to P07: I/O Port	$\overline{E}$, P00 to P04: $\overline{CS0}$ – $\overline{CS4}$, P05: $\overline{RSMP}$, P06 P07: Address A17, A16	Same as left
Port P1 BYTE = "L"	$\overline{E}$	$\overline{E}$, P10 to P17: A8 to A15, Address, Data(odd)	Same as left
Port P1 BYTE = "H"	P10 to P17: I/O Port	$\overline{E}$, P10 to P17: Address A8 – A15	Same as left
Port P2 BYTE = "L"	$\overline{E}$	$\overline{E}$, P20 to P27: Address, Data(even)	Same as left
Port P2 BYTE = "H"	P20 to P27: I/O Port	$\overline{E}$, P20 to P27: Address, Data (odd,even)	Same as left
Port P3	$\overline{E}$, P30 to P33: I/O Port	$\overline{E}$, P30: $\overline{WEL}$, P31: $\overline{WEH}$, P32: ALE, P33: $\overline{HLDA}$ (Note 2)	Same as left
Port P4	$\overline{E}$, P40 to P47: I/O Port • In this case, bit 7 of the processor mode register 0 is "0"	$\overline{E}$, P40: $\overline{HOLD}$, P41: $\overline{RDY}$, P42 to P47: I/O Port • In this case, bit 7 of the processor mode register 0 is "0"	Same as left except for port P42 which outputs ϕ1 independent of the bit 7 of the processor mode register 0 (Note 2)
	P42: ϕ1 Same as above except for P42 • In this case, bit 7 of the processor mode register 0 is "1"	P42: ϕ1 Same as above except for P42 • In this case, bit 7 of the processor mode register 0 is "1"	

EOL announced

Fig. 58 Relationship between ports P0 to P4, pin $\overline{E}$/$\overline{RDE}$ and processor modes

Notes 1. In the memory expansion mode and the microprocessor mode, signal $\overline{E}$ is not output.

2. The signal output disable selection bit (bit 6 of the oscillation circuit control register 0) can stop the $\overline{E}$ signal output in the single-chip mode and the ϕ1 output in the microprocessor mode. In the memory expansion mode or the microprocessor mode, signals $\overline{RDE}$, $\overline{WEL}$, $\overline{WEH}$ can also be fixed to "H" when the internal memory area is accessed.

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

• Wait bit

As shown in Figure 60, when the external memory area is accessed with the wait bit (bit 2 of the processor mode register 0 at address $5E_{16}$) cleared to "0", the access time can be extended compared with no wait (the wait bit is "1").

The access time is extended in two ways and this is selected with the wait selection bit (bit 0 of the processor mode register 1 at address $5F_{16}$).

When this bit is "1", the access time is 1.5 times compared to that for no wait. When this bit is "0", the access time is twice compared to that for no wait.

At reset, the wait bit and the wait selection bit are "0".

The accessing of internal memory area is always performed in the no wait mode regardless of the wait bit.

The processor modes are described below.

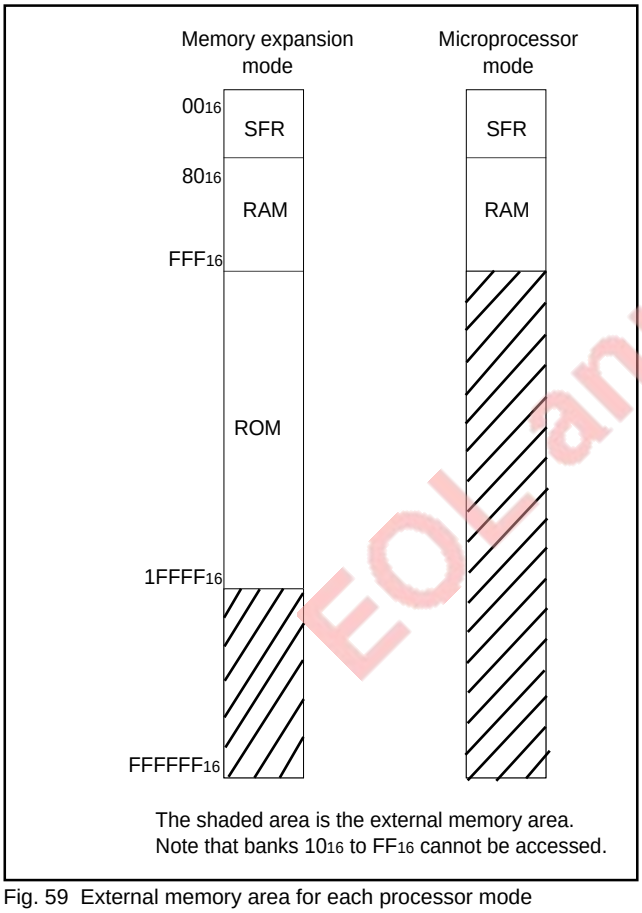


Fig. 59 External memory area for each processor mode

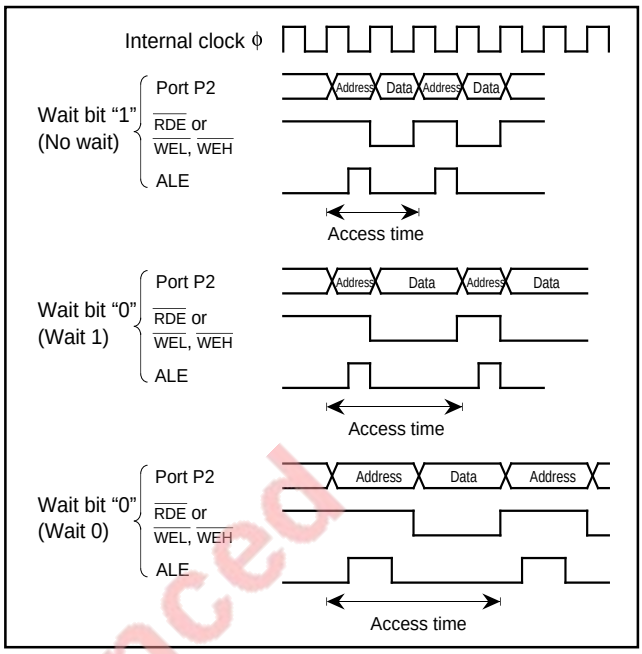


Fig. 60 Relationship between wait bit, wait selection bit, and access time

(1) Single-chip mode [00]

Single-chip mode is entered by connecting the CNVss pin to Vss and starting from reset. Ports P0 to P4 all function as normal I/O ports. Port $P4_2$ can output clock ϕ_1 by setting bit 7 of the processor mode register 0 to "1". For clock ϕ_1, refer to Figure 65.

In this mode, signal $\overline{E}$ is output from pin $\overline{E}/\overline{RDE}$. Signal $\overline{E}$ output, however, can be stopped by setting the signal output disable selection bit (bit 6 of the oscillation circuit control register 0) to "1" to switch the $\overline{E}/\overline{RDE}$ pin function to "L" output. Table 7 shows the function of the signal output disable selection bit.

(2) Memory expansion mode [01]

Memory expansion mode is entered by setting the processor mode bits to "01" after connecting the CNVss pin to Vss and starting from reset.

Pin $\overline{E}/\overline{RDE}$ becomes the output pin for $\overline{RDE}$.

$\overline{RDE}$ is a read-enable signal and is "L" during the data read term in the read cycle. When the internal memory area is read, $\overline{RDE}$ can be fixed to "H" by setting the signal output disabe selection bit (bit 6 of the oscillation circuit control register) to "1".

Ports $P0_6$ and $P0_7$ become the output pins for addresses A_{16} and A_{17}, respectively. Similarly, port $P0_5$ becomes the output pin for $\overline{RSMP}$, and ports $P0_0$ to $P0_4$ become the output pins for $\overline{CS_0}$ to $\overline{CS_4}$, respectively. In this case, their functions as I/O ports are lost.

EOL announced

$\overline{CS_0}$ to $\overline{CS_4}$ are the chip select signals and are "L" when the address shown in Table 7 is accessed. $\overline{RSMP}$ is the ready-sampling signal which is output for the $\overline{RDY}$ input described later when the external memory area is accessed. By inputting logical AND of $\overline{RSMP}$ and $\overline{CS_n}$ (n = 0 to 4) to the $\overline{RDY}$ pin, read/write term for any address areas can be extended by 1 cycle of clock ϕ_1. In addition, the read/write term can also be extended by 2 cycles of clock ϕ_1 if the above function and wait 0/1 function specified with the wait bit are used together.

Port P1 has two functions depending on the level of the BYTE pin. In bose cases, the I/O port function is lost.

When the BYTE pin level is "L", port P1 functions as an address (A_{15} to A_8) output pin while $\overline{RDE}$ or $\overline{WEL}$, $\overline{WEH}$ are "H" and as an odd address data I/O pin while these signals are "L". However, if an internal memory is read, external data is ignored while $\overline{RDE}$ is "L".

When the BYTE pin level is "H", port P1 functions as an address output pin.

Port P2 has two functions depending on the level of the BYTE pin. In bose cases, the I/O port function is lost.

When the BYTE pin level is "L", port P2 functions as an address (A_0 to A_7) output pin while $\overline{RDE}$ or $\overline{WEL}$, $\overline{WEH}$ are "H" and as an even address data I/O pin while these signals are "L". However, if an internal memory is read, external data is ignored while $\overline{RDE}$ is "L".

When the BYTE pin level is "H", port P2 functions as an address (A_0 to A_7) output pin while $\overline{RDE}$ or $\overline{WEL}$, $\overline{WEH}$ are "H" and as an even and odd address data I/O pin while these signals are "L". However, if an internal memory is read, external data is ignored while $\overline{RDE}$ is "L".

Ports $P3_0$, $P3_1$, $P3_2$, and $P3_3$ become $\overline{WEL}$, $\overline{WEH}$, ALE, and $\overline{HLDA}$ output pins, respectively and lose their I/O port functions.

$\overline{WEL}$, $\overline{WEH}$ are the write-enable low signal and the write-enable high signal, respectively. These signals go "L" during the data write term of the write cycle, but their operations differ depending on the BYTE pin level.

In the case the BYTE pin level is "L", $\overline{WEL}$ is "L" when writing to an even address, $\overline{WEH}$ is "L" when writing to an odd address, and both $\overline{WEL}$ and $\overline{WEH}$ are "L" when writing to even and odd addresses.

In the case the BYTE pin level is "H", regardless of address, only $\overline{WEL}$ is "L", and $\overline{WEH}$ retains "H". $\overline{WEL}$ and $\overline{WEH}$ can also be fixed to "H" when the internal memory is accessed, same as $\overline{RDE}$, by writing "1" to the signal output disable selection bit.

ALE is an address latch enable signal used to latch the address signal from a multiplexed signal of address and data. The latch is transparent while ALE is "H" to let the address signal pass through and held while ALE is "L".

$\overline{HLDA}$ is a hold acknowledge signal and is used to notify externally when the microcomputer receives $\overline{HOLD}$ input and enters into hole state.

Ports $P4_0$ and $P4_1$ become $\overline{HOLD}$ and $\overline{RDY}$ input pin, respectively, and lose their output pin function.

$\overline{HOLD}$ is a hold request signal. It is an input signal used to put the microcomputer in hold state. $\overline{HOLD}$ input is accepted when the internal clock ϕ falls from "H" level to "L" level while the bus is not used. Ports P0, P1, P2, $P3_0$, $P3_1$, and pin $\overline{E}/\overline{RDE}$ are floating while the microcomputer stays in hold state. These ports become floating after one cycle of internal clock ϕ later than $\overline{HLDA}$ signal changes to "L" level. At releasing hold state, these ports are released from floating state after one cycle of internal clock ϕ later than $\overline{HLDA}$ signal changes to "H" level.

$\overline{RDY}$ is a ready signal. If this signal goes "L", the internal clock ϕ stops at "L". $\overline{RDY}$ is used when slow external memory is attached. Port $P4_2$ becomes a normal I/O port when bit 7 of the processor mode register 0 is "0" and becomes an output pin for clock ϕ_1 when bit 7 is "1". The ϕ_1 output is independent of $\overline{RDY}$ and does not stop even when internal clock ϕ stops because of "L" input to the $\overline{RDY}$ pin.

EOL announced

(3) Microprocessor mode [10]

Microprocessor mode is entered by connecting the CNVss pin to Vcc and starting from reset. It can also be entered by programming the processor mode bits to "10" after connecting the CNVss pin to Vss and starting from reset. This mode is similar to the memory expansion mode except that internal ROM is disabled and an external memory is required, and clock ϕ_1 from port $P4_2$ is always output independently of bit 7 of the processor mode register 0.

As shown in Table 8, ϕ_1 output can also be stopped with the signal output disable selection bit "1". In this case, write "1" to the port $P4_2$ direction register.

Table 9 shows the relationship between the CNVss pin input level and the processor modes.

Table 9. Relationship between CNVss pin input levels and processor modes

CNVss	Mode	Description
Vss	• Single-chip • Memory expansion • Microprocessor	Single-chip mode upon starting after reset. Each mode can be selected by changing the processor mode bits by software.
Vcc	• Microprocessor	Microprocessor mode upon starting after reset.

Table 7. Relationship between access addresses and chip-select signals $\overline{CS_0}$ to $\overline{CS_4}$

Chip-select signal	Area	Access address	
		Memory expansion mode	Microprocessor mode
$\overline{CS_0}$	The first half of bank 00_{16} except for internal momory area	————	00 1000_{16} to 00 $7FFF_{16}$
$\overline{CS_1}$	The latter half of bank 00_{16} except for internal memory area and banks 01_{16} to 03_{16}.	02 0000_{16} (Note) to 03 $FFFF_{16}$	00 8000_{16} to 03 $FFFF_{16}$
$\overline{CS_2}$	Banks 04_{16} to 07_{16}	04 0000_{16} to 07 $FFFF_{16}$	04 0000_{16} to 07 $FFFF_{16}$
$\overline{CS_3}$	Banks 08_{16} to $0B_{16}$	08 0000_{16} to 0B $FFFF_{16}$	08 0000_{16} to 0B $FFFF_{16}$
$\overline{CS_4}$	Banks $0C_{16}$ to $0F_{16}$	0C 0000_{16} to 0F $FFFF_{16}$	0C 0000_{16} to 0F $FFFF_{16}$

Note. This applies when both bits 1 and 0 of the memory allocation control register is "0". Refer to on the section ROM AREA MODIFICATION FUNCTION.

Table 8. Function of signal output disable selection bit CM_6 (bit 6 of oscillation circuit control register 0)

Processor mode	Pin	Function	
		CM_6 = "0"	CM_6 = "1"
Single-chip mode	$\overline{E}$	Enable signal $\overline{E}$ is output.	"L" is output.
Memory expansion mode, Microprocessor mode	$\overline{RDE}$, $\overline{WEL}$, $\overline{WEH}$	$\overline{RDE}$, $\overline{WEL}$, $\overline{WEH}$ are output when the internal/external memory area is accessed.	$\overline{RDE}$, $\overline{WEL}$, $\overline{WEH}$ are output only when the external memory area is accessed.
	$\overline{RDE}$	After WIT/STP instruction is executed, "H" is output.	"L" is output after WIT/STP instruction is executed. ∗ Standby state selection bit (bit 0 of port function control register) must be set to "1".
Microprocessor mode	ϕ_1	Clock ϕ_1 is output independent of ϕ_1 output selection bit.	"H"or "L" is output. (Output the content of $P4_2$ latch.) ∗ Port $P4_2$ direction register must be set to "1".

Note. Functions shown in Table 7 cannot be emulated in a debugger. For the oscillation circuit control register 0, refer to Figure 63. For the port function control register, refer to Figure 11.

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

OSCILLATION CIRCUIT

In the oscillation circuit, two kinds of clock circuits are built-in. One is the main-clock oscillation circuit which uses the XIN and XOUT pins, and the other is the sub-clock (32 kHz) oscillation circuit which uses the XCIN and the XCOUT pins.

Either of these two oscillation circuits can output the system clock, and it can be selected.

Figure 61 shows the oscillation circuit example with a ceramic resonator or a quartz-crystal oscillator connected. The circuit constants such as capacitance depend on a resonator/oscillator, and these constants shall be set to the resonator/oscillator manufacture's recommended value.

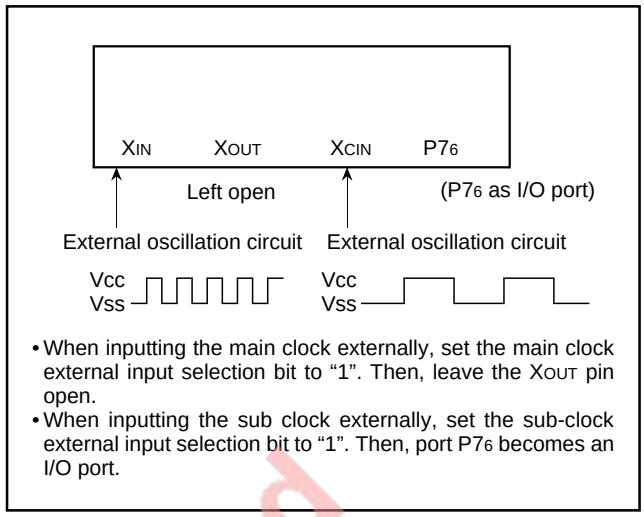
Figure 62 shows the example of the external clock input circuit.

When inputting the main clock externally, the main-clock oscillation circuit stops operating and power dissipation could be conserved by setting the main clock external input selection bit (bit 1 of the oscillation circuit control register 1, refer to Figure 63) to "1". Note that this bit also has the function to select a return factor from STP state (refer to the section on the STANDBY FUNCTION.) Additionally, write to the oscillation circuit control register 1 as the flow shown in Figure 64.

Pins XCIN and XCOUT of the sub-clock oscillation circuit are also used as I/O ports P77 and P76, and these functions are selected with the port-Xc selection bit described below.

From the time during reset to the time after releasing reset, only the main-clock oscillation circuit operates and the main clock is selected as the system clock. Furthermore, at this time, the sub-clock oscillation circuit stops and pins XCIN and XCOUT become I/O ports (P77, P76).

When the port-Xc selection bit (bit 4 of the oscillation circuit control register 0) is set to "1" in this condition, I/O ports P77 and P76 are switched to pins XCIN and XCOUT, and then, oscillation starts in the sub-clock oscillation circuit.



- When inputting the main clock externally, set the main clock external input selection bit to "1". Then, leave the XOUT pin open.
- When inputting the sub clock externally, set the sub-clock external input selection bit to "1". Then, port P76 becomes an I/O port.

Fig. 62 External clock input circuit

When inputting the sub clock externally, set the sub-clock external input selection bit (bit 2 of the oscillation circuit control register 1) to "1" before selecting pins XCIN and XCOUT with the port-Xc selection bit. When the sub-clock external input selection bit is set to "1", port P76 becomes an I/O port (or an analog input AN6). Note that this bit also has the function to select a return factor from STP state (refer to the section on the STANDBY FUNCTION.)

When the sub-clock output selection bit (bit 1 of the port function control register, refer to Figure 11) is set to "1" under the condition of the port-Xc selection bit = "1", the sub-clock ϕ_{SUB} is output from port P67. Accordingly, the sub-clock 32 kHz can be supplied for external devices.

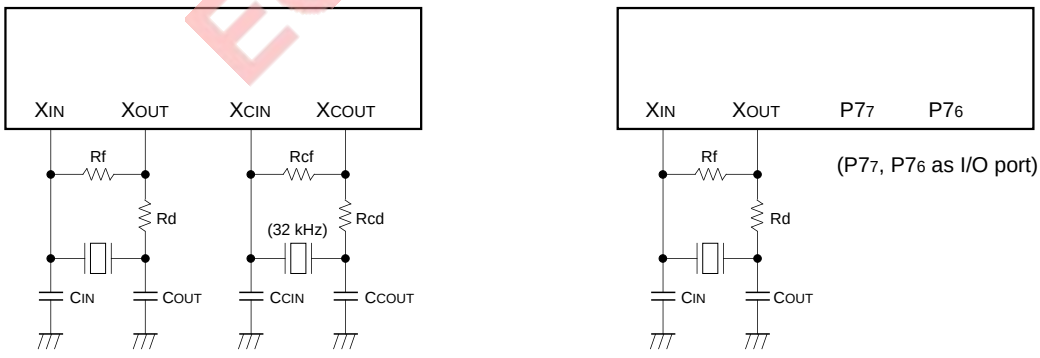


Fig. 61 Oscillation circuit example with external resonator or quartz-crystal oscillator

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

CLOCK GENERATING CIRCUIT

Figures 63 and 65 show the bit configuration of the oscillation circuit control registers 0, 1 and the clock generating circuit diagram. The clock generating circuit consists of main- and sub-clock oscillation circuits, system clock switch circuit, clock dividing circuit, standby control circuit, and others. The oscillation circuit control registers are some of the control registers for the clock generating circuit.

Clocks ϕ, f_2 to f_{512}, f_{C32}, and ϕ_1 are used in CPU and internal peripheral devices or are output from pins, and they are made of the main or sub clock, as shown in Figure 65.

The system clock and the clock f_2 can be switched to high-speed clocks or low-speed clocks shown in Table 10. When using the sub clock, it is possible to select one of 3 types: the main clock divided by 2, the direct main clock (not divided) and the sub clock divided by 2 as the clock f_2.

When not using the sub clock, it is possible to select one of 4 types: the main clock divided by 2, divided by 8, divided by 16 and the direct main clock (not divided) as the clock f_2.

This function of clocks switch make it possible to adapt power control to the system operation.

Bits 0 to 4 of the oscillation circuit control register 0 and bit 0 of the oscillation circuit control register 1 control sub-clock oscillation start, system clock selection, stop/restart of main-clock oscillation, sub-clock drivability selection and the main clock division selection.

The method of clocks switch is described bellow.

When selecting the main clock as the system clock, the main clock division selection bit (bit 0 of the oscillation circuit control register 1) selects either the main clock divided by 2 or the direct main clock as the clock f_2. When this bit is “1”, the clock f_2 is the direct main clock which is not divided, so that a half external input frequency is enough to perform the same operation speed. Consequently, power dissipation could be conserved (refer to Figure 69.) The main clock division selection bit is valid regardless of either using the sub clock or not.

Figure 66 shows the system clock state transition when using the sub clock.

From the time during reset to the time reset is released, only the main clock, which is selected as the system clock, oscillates.

If the port-X_C selection bit is set to “1” in this term, the sub-clock oscillation circuit starts oscillation. When the sub clock is not used, fix the port-X_C selection bit to “0” (“0” at reset) and use the $P7_7/AN_7$ X_{CIN} and $P7_6/AN_6/X_{COUT}$ pins as I/O ports $P7_7$ and $P7_6$ or analog inputs AN_7 and AN_6, respectively.

Table 10. Selection of system clock and clock f_2

Sub clock	Port-Xc selection bit (CM_4)	System clock selection bit (CM_3)	Main clock division selection bit (CC_0)	System clock	Clock f_2
Not used	0	0	0	Main clock	Main clock divided by 2
	0	0	1	Main clock	Main clock
	0	1	0	Main clock divided by 8	Main clock divided by 16
	0	1	1	Main clock divided by 8	Main clock divided by 8
Used	1	0	0	Main clock	Main clock divided by 2
	1	0	1	Main clock	Main clock
	1	1	0	Sub clock	Sub clock divided by 2
	1	1	1	Sub clock	Sub clock divided by 2

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

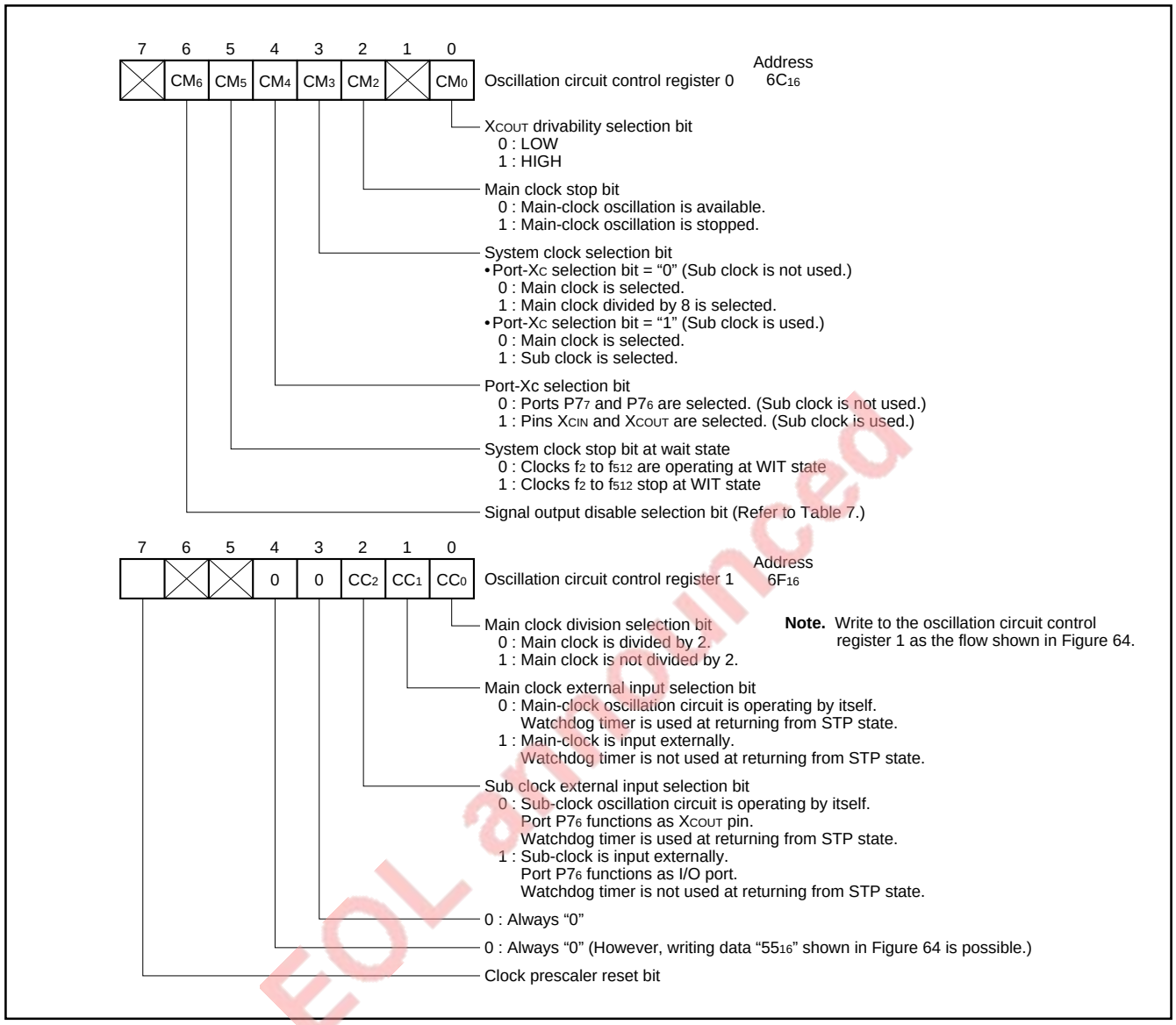


Fig. 63 Bit configuration of oscillation circuit control registers 0, 1

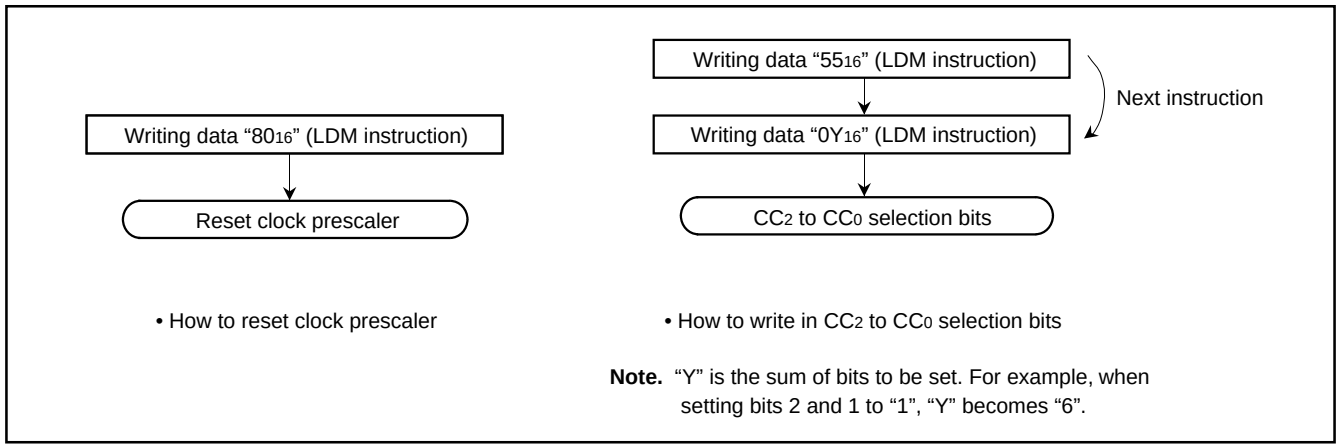


Fig. 64 How to write data in oscillation circuit control register 1

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

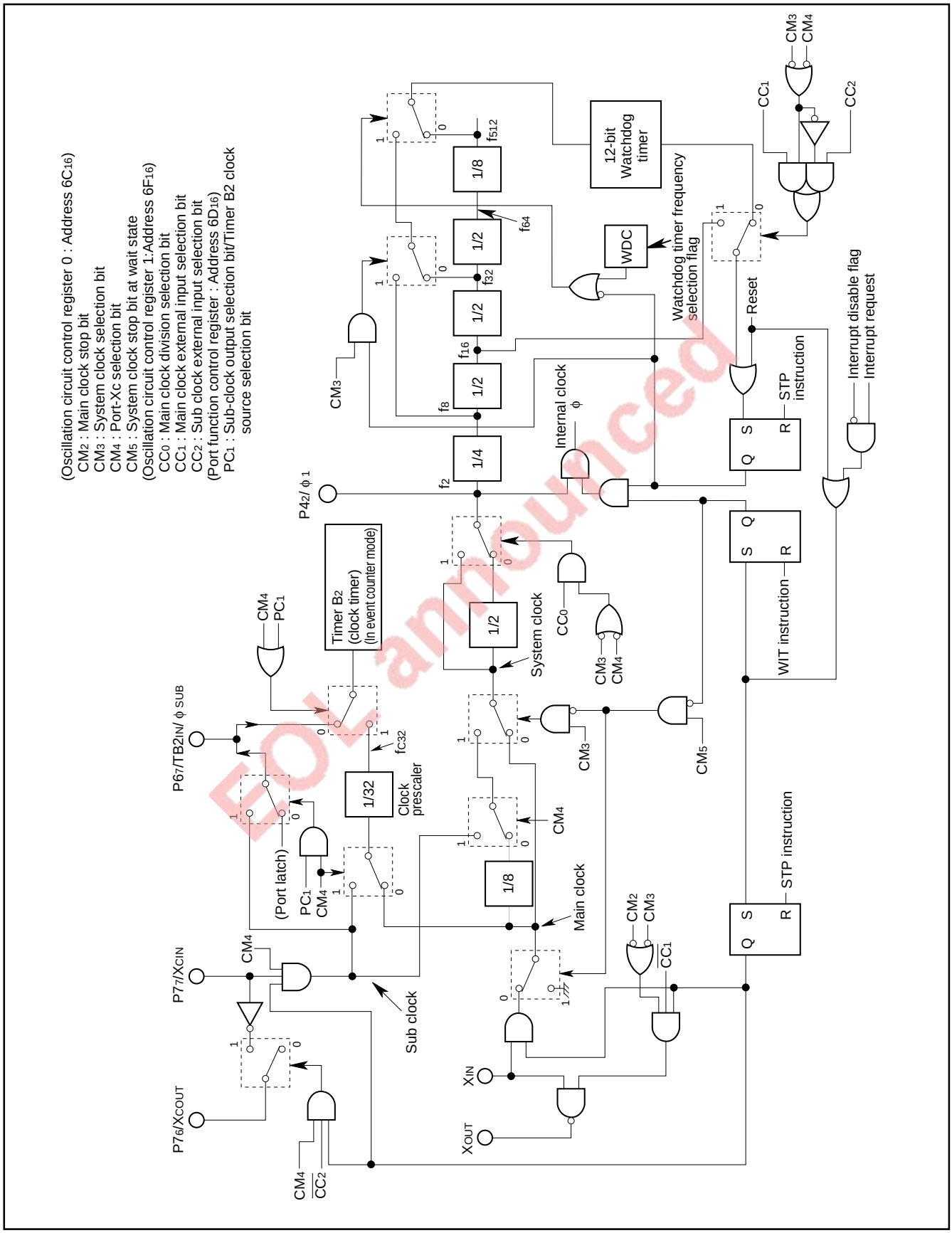
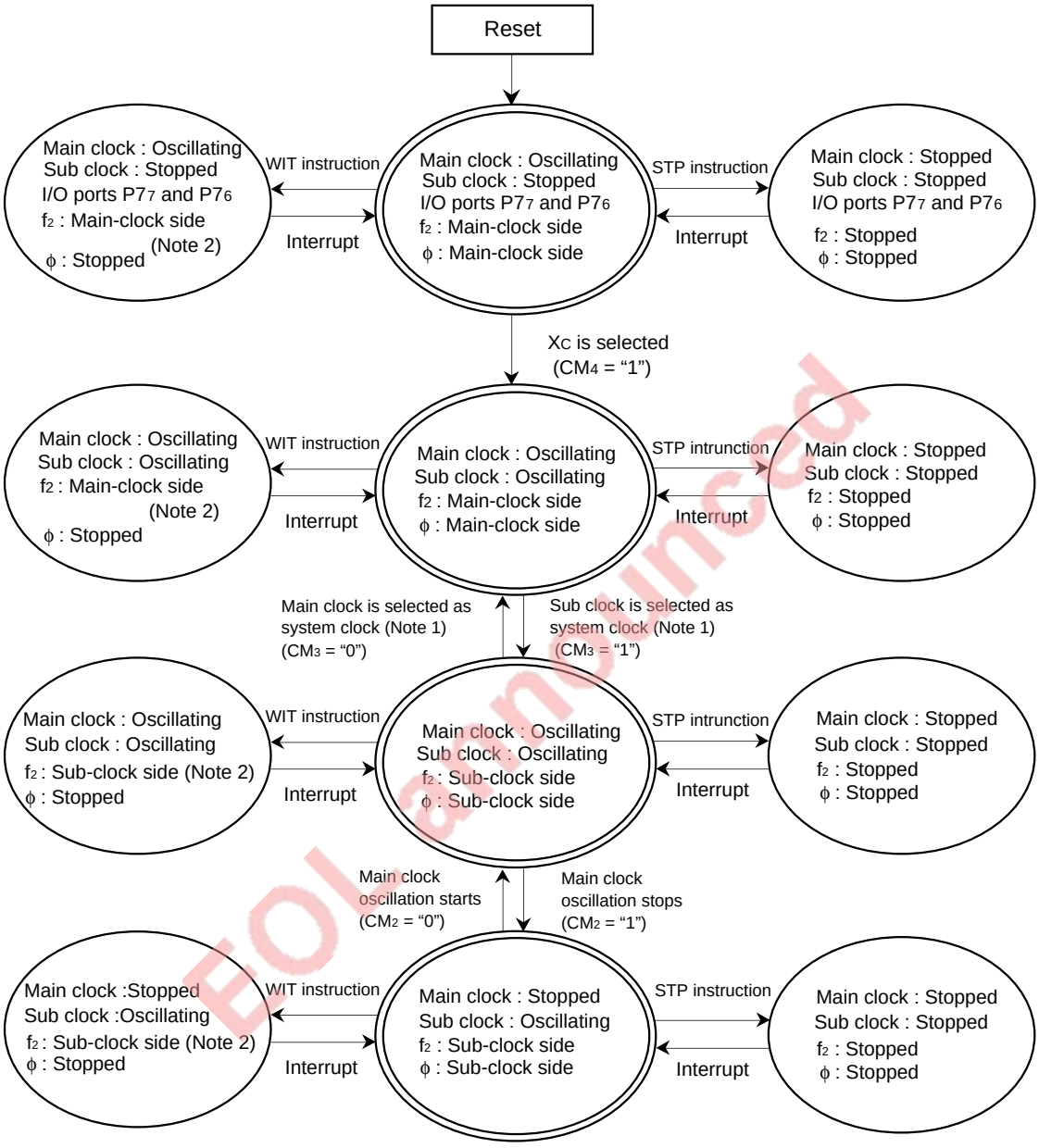


Fig. 65 Block diagram of clock generating circuit

EOL announced



Notes 1. Before selecting the main/sub clock of which oscillation already starts as the system clock, use software to fully stabilize the oscillation.
2. When the system clock stop bit at wait state (CM 5) is "1", f2 stops at wait state.

Fig. 66 System clock state transition

EOL announced

Note that the port-Xc selection bit cannot be cleared by software when it is once set to “1”. The bit can be cleared only by reset. It is impossible to write “1” to the port-Xc selection bit and the system clock selection bit at the same time. In addition, the contents of the main clock stop bit and the XCOUT drivability selection bit cannot be changed when the port-Xc selection bit is “0”.

Figure 67 shows the system clock selection change example when using the sub clock. When the system clock selection bit is “1” after sub-clock oscillation starts, the sub clock is selected as the system clock. Make sure to select the sub clock after the sub-clock oscillation is fully stabilized.

When the main clock stop bit is set to “1” after the sub clock is selected, the main-clock oscillation/input stops. By stopping the main-clock oscillation, current consumption can be further restricted.

When the main clock stop bit is cleared to “0” after the main-clock oscillation stops, the main-clock oscillation/input restarts.

When the system clock selection bit is “0” after the main-clock oscillation restarts, the main clock is selected as the system clock again. Make sure to select the main clock after the main-clock oscillation restarts and is fully stabilized.

The XCOUT drivability selection bit is a bit to select the drivability of the sub-clock oscillation circuit and is set to “1” (HIGH) after reset is released. Make sure to clear the XCOUT drivability selection bit to “0” (LOW) after the sub-clock oscillation is fully stabilized.

EOL announced

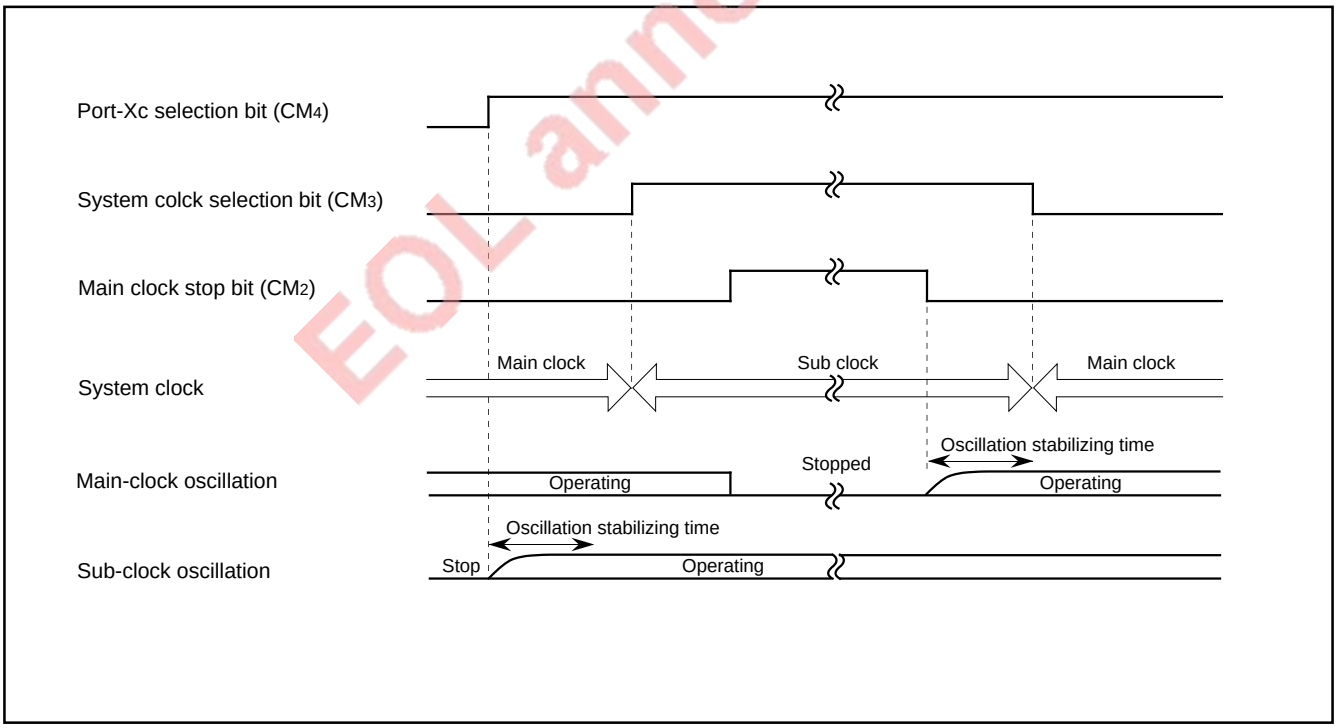


Fig. 67 System clock selection change example

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

When the port-Xc selection bit is set to "1" to use sub-clock oscillation and timer B2 is set to be in the event count mode, clock f_{C32} which is the sub clock (32 kHz) divided by 32 is selected as the count source of timer B2. By this selection, timer B2 can be used as the clock timer. For setting of timer B2 related registers, refer to the section on clock timer mode of timer B2.

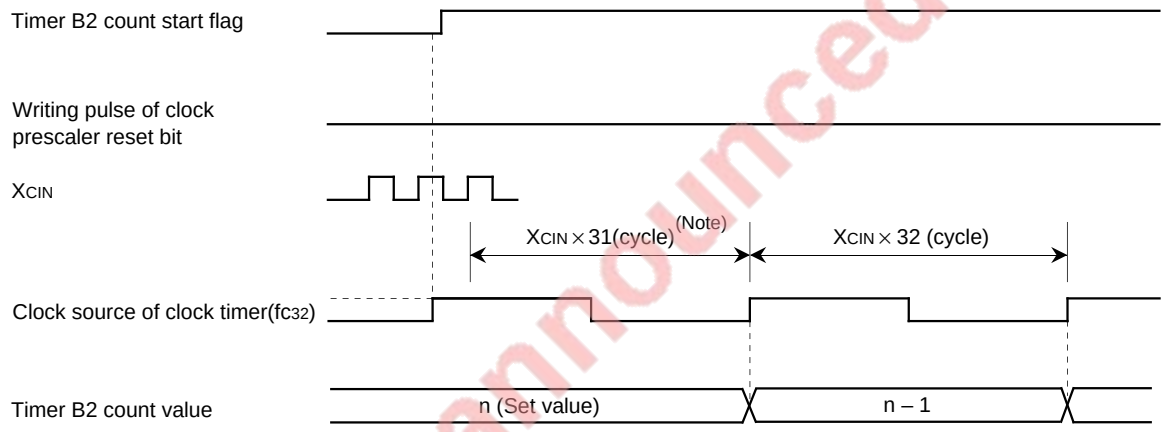
The clock prescaler in which the sub clock is divided by 32 is reset by writing "1", in dummy, into bit 7 (clock prescaler reset bit) of the oscillation circuit control register 1.

When the main clock is selected, by this function, clock f_{C32} of clock timer B2 can be synchronized with software. Figure 68 shows the operation timing for clock prescaler and clock timer B2.

Figure 69 shows the clock f_2 state transition when the port-Xc selection bit is "0" and the sub clock is not used.

From the time during reset to the time reset is released, the main clock divided by 2 is being selected as the clock f_2. When the system clock selection bit is set "1" in that condition, the main clock divided by 16 is selected as the clock f_2 and the clock frequency supplied for the CPU and internal peripheral devices is divided by 8 more. It makes current consumption restrict, although the operation speed slows.

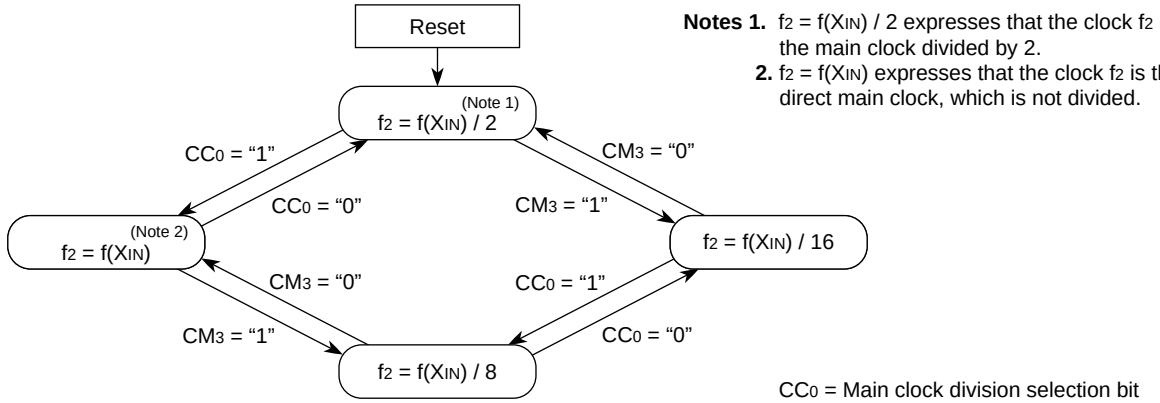
When the timer B2 clock source selection bit (bit 1 of the port function control register) is set to "1" and event counter mode is selected in timer B2 under the condition which the port-Xc selection bit is "0"; f_{C32}, which is the main clock divided by 32, is connected as a timer B's count source. Accordingly, timer B2 can be used as a clock timer which always operates with a regular clock source shown in Figure 69. For details relating to register setting of timer B2, refer to the section "Clock timer" on timer B.



This applies when the main clock is selected as the system clock (System clock selection bit (CM_3) = "0").

Note. Period of f_{C32} = X_{CIN} × 31 (cycle) (only in this term)
Period of f_{C32} = X_{CIN} × 32 (cycle) (after this term)

Fig. 68 Operation timing for clock prescaler and clock timer B2



Notes 1. f_2 = f(X_{IN}) / 2 expresses that the clock f_2 is the main clock divided by 2.
2. f_2 = f(X_{IN}) expresses that the clock f_2 is the direct main clock, which is not divided.

• In the case of not using sub clock (CM_4 = "0")

CC_0 = Main clock division selection bit
CM_3 = System clock selection bit
CM_4 = Port-Xc selection bit

Fig. 69 Clock f_2 state transition (when the sub clock is not used.)

EO announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

STANDBY FUNCTION

The WIT and the STP instructions make the microcomputer standby state.

Table 11 shows the relationship between standby state and each block's operation.

When the WIT instruction is executed with the system clock stop bit at wait state (bit 5 of the oscillation circuit control register 0) = "0", internal clock ϕ is stopped being at "L", but the oscillation circuit, system clock, and divided clocks f_2 to f_{512} are not stopped.

Because divided clocks f_2 to f_{512} are not stopped, a part of internal peripheral devices which use these divided clocks can operate even at wait state.

Otherwise, when the WIT instruction is executed with the system clock stop bit at wait state = "1", the oscillation circuit is not stopped, but the system clock, divided clocks, and internal clock ϕ are stopped. Accordingly, in this case, all of the internal peripheral devices which use divided clocks f_2 to f_{512}, including the watchdog timer, are stopped.

When port-Xc selection bit is "1" to operate the sub-clock oscillation circuit, however, clock timer B2 can operate because clock f_{C32} for the clock timer is not stopped.

When internal peripheral devices are not used, later wait state (System clock stop bit at wait state = "1") is more effective to restrict the current consumption.

Make sure to set the system clock stop bit at wait state to "1" immediately before the WIT instruction execution and clear the bit to "0" immediately after the wait state is terminated.

The wait state is terminated when an interrupt request is accepted, and the internal clock ϕ operation is restarted. At this time, interrupt processing can immediately be executed because oscillation circuit's operation is not stopped during the wait state.

When the STP instruction is executed, the oscillation circuit is stopped with internal clock ϕ stopped at "L". Furthermore, "FFF_{16}" is automatically set into the watchdog timer, and the clock source of the watchdog timer is forced to connect with f_{32} when the main clock is selected or f_8 when the sub clock is selected. This connection is cut off when the most significant bit of the watchdog timer is cleared to "0" or the microcomputer is reset, and the clock source is connected with the input depending on the content of the watchdog timer frequency selection flag. In the stop state, internal peripheral devices using divided clocks f_2 to f_{512} are stopped.

The stop state is terminated by system reset or interrupt request acceptance, and then oscillation is restarted. At this time, supply of system clock and divided clocks f_2 to f_{512} is restarted.

In that condition, when the main clock external input selection bit is "0" and the main clock is being selected as a system clock, or when the sub clock external input selection bit is "0" and the sub clock is being selected as a system clock, internal clock ϕ is stopped at "L" till the most significant bit of the watchdog timer decremented with divided clock f_{32} or f_8 becomes "0". However, supply of internal clock ϕ is restarted immediately after the oscillation restarts by reset. Accordingly, in this case, it is necessary to wait for the oscillation stabilized before making the reset input "H".

Otherwise in that condition, when the main clock external input selection bit is "1" and the main clock is being selected as a system clock, or when the sub clock external input selection bit is "1" and the sub clock is being selected as a system clock, supply of internal clock ϕ is restarted from the seventh clock of clock f_2 after the oscillation restarts. By this function, the microcomputer can immediately return from the stop state when the clock supply input from the external is stabilized.

Even though the main clock or the sub clock is input externally, make sure to clear the main clock external input selection bit or the sub clock external input selection bit to "0" before executing the STP instruction if this external clock is unstable for a short time at a return from the stop state.

EOL announced

Table 11. Relationship between standby state and each block's operation

Instruction	System clock stop bit at wait state	Operation at WIT/STP state					
		Oscillation circuit	System clock	f_2 – f_{512}	Clock output ϕ_1	Internal clock ϕ	Internal peripheral devices using f_2 – f_{512}
WIT	"0"	Operating (Note)	Operating	Operating	Operating	Stopped ("L")	Operation enabled (Watchdog timer is operating)
	"1"	Operating (Note)	Stopped	Stopped	Stopped ("L")	Stopped ("L")	Operation disabled (Watchdog timer is stopped) (Clock timer's operation is enabled)
STP	—	Stopped	Stopped	Stopped	Stopped ("L")	Stopped ("L")	Operation disabled (Watchdog timer is stopped)

Note. When the main clock external input selection bit is "1", the main clock oscillation circuit stops. When the sub clock external input selection bit is "1", the sub-clock oscillation circuit stops. (In both cases, the external clock can be input.)

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

The wait/stop state is terminated by interrupt acceptance or reset. Accordingly, it is necessary to prepare the state in which any interrupt can be accepted before the WIT/STP instruction is executed. Additionally, it is necessary to set the system clock stop bit at wait state before the WIT instruction is executed.

When the WIT/STP instruction is executed in a bus access cycle, the bus enters the non-access state (each signal of $\overline{E}$, $\overline{RDE}$, $\overline{WEL}$ and $\overline{WEH}$ is at "H") because internal clock ϕ (or oscillation) is stopped after the read/write in this cycle is finished. Pins $P0_0/\overline{CS_0}$ to $P3_3/\overline{HLDA}$ normally retain the state at which internal clock ϕ is stopped in the wait/stop state.

However, only in the memory expansion mode and the microprocessor mode, arbitrary data which is set in the port P0 to P3 latches can be output from pins $P0_0/\overline{CS_0}$ to $P3_3/\overline{HLDA}$ even at the wait/stop state when the following conditions are satisfied before the WIT/STP instruction execution.

- The standby state selection bit (bit 0 of the port function control register) is set to "1".
- "FF_{16}" is set into the port P0 to P3 direction registers.

Furthermore, when the standby state selection bit is set to "1" and bit 6 of the oscillation circuit control register 0 (signal output disable selection bit) is set to "1", "L" level can be output from the $\overline{E}/\overline{RDE}$ pin at the wait/stop state. For the signal output disable selection bit, refer to Table 8 on the processor mode section.

Note that the function of arbitrary data output cannot be emulated using a debugger.

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

ROM AREA MODIFICATION FUNCTION

The internal ROM size and RAM size of the M37735MHBXXXFP can be modified by the memory allocation control register's bits 0,1 and 2 shown in Figure 70.

Figure 72 shows the memory allocation in which the internal ROM size and RAM size are modified.

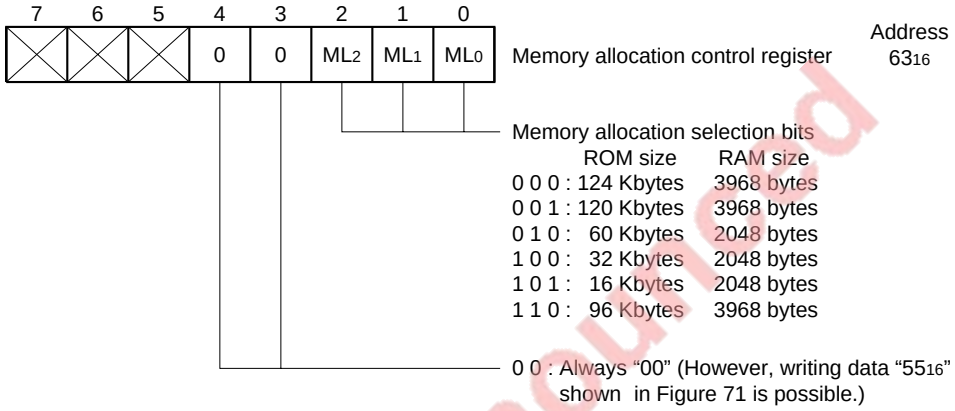
Make sure to write data in the memory allocation control register as the flow shown in Figure 71.

This ROM area modification function is valid in memory expansion mode and single-chip mode.

Table 12 shows the relationship between the memory allocation selection bits and addresses corresponding to chip-select signals $\overline{CS_0}$ and $\overline{CS_1}$.

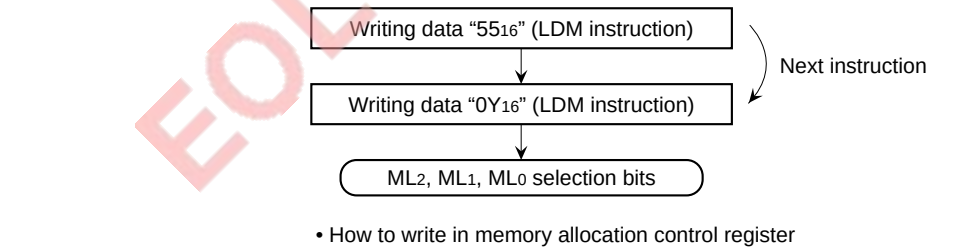
When ordering a mask ROM, Mitsubishi Electric corp. produces the mask ROM using the data within 128 Kbytes (addresses 000000_{16} – $01FFFF_{16}$). It is regardless of the selected ROM size (refer to MASK ROM ORDER CONFIRMATION FORM.) Therefore, program "FF_{16}" to the addresses out of the selected ROM area in the EPROM which you tender when ordering a mask ROM.

Address $01FFFF_{16}$ of this microcomputer corresponds to the lowest address of the EPROM which you tender.



Note. Write to the memory allocation control register as the flow shown in Figure 71.

Fig. 70 Bit configuration of memory allocation control register



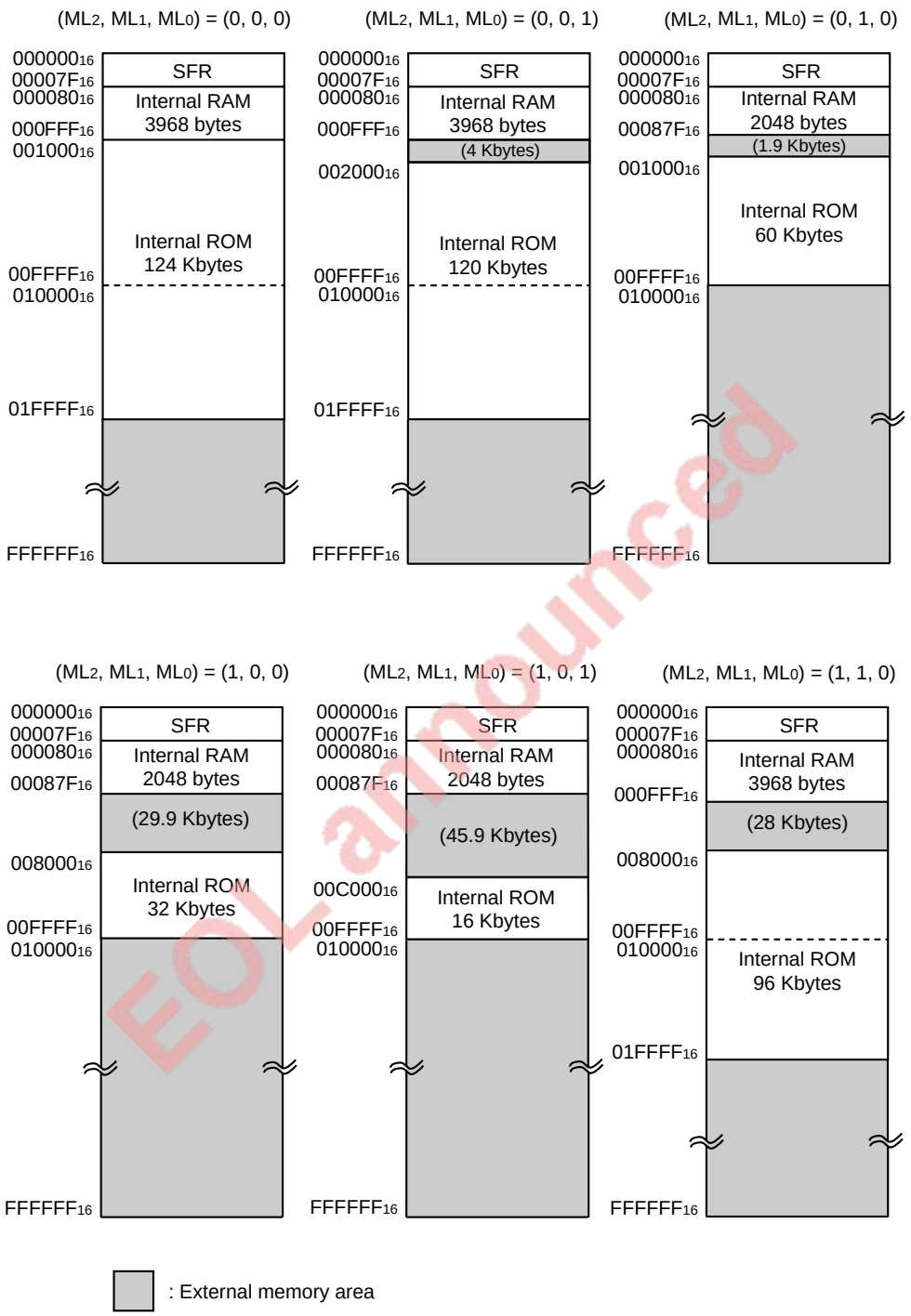
Note. "Y" is the sum of bits to be set. For example, when setting bit 1 to "1", "Y" becomes "2".

Fig. 71 How to write data in memory allocation control register

Table 12 Relationship between memory allocation selection bits and addresses corresponding to chip-select signals $\overline{CS_0}$ and $\overline{CS_1}$

Memory allocation selection bits			Internal ROM area	Access addresses	
ML_2	ML_1	ML_0		$\overline{CS_0}$	$\overline{CS_1}$
0	0	0	001000_{16} – $01FFFF_{16}$	————	020000_{16} – $03FFFF_{16}$
0	0	1	002000_{16} – $01FFFF_{16}$	001000_{16} – $001FFF_{16}$	020000_{16} – $03FFFF_{16}$
0	1	0	001000_{16} – $00FFFF_{16}$	000880_{16} – $000FFF_{16}$	010000_{16} – $03FFFF_{16}$
1	0	0	008000_{16} – $00FFFF_{16}$	000880_{16} – $007FFF_{16}$	010000_{16} – $03FFFF_{16}$
1	0	1	$00C000_{16}$ – $00FFFF_{16}$	000880_{16} – $007FFF_{16}$	008000_{16} – $00BFFF_{16}$ 010000_{16} – $03FFFF_{16}$
1	1	0	008000_{16} – $01FFFF_{16}$	001000_{16} – $007FFF_{16}$	020000_{16} – $03FFFF_{16}$

EOL announced



Note. Banks 10_{16}–FF_{16} cannot be accessed in the 7735 group.

Fig. 72 Memory allocation (modification of internal ROM and RAM area by memory allocation selection bits)

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

ADDRESSING MODES

The M37735MHBXXXFP has 28 powerful addressing modes. Refer to the “7700 Family Software Manual” for the details.

MACHINE INSTRUCTION LIST

The M37735MHBXXXFP has 103 machine instructions. Refer to the “7700 Family Software Manual” for the details.

DATA REQUIRED FOR MASK ROM ORDERING

Please send the following data for mask orders.
(1) M37735MHBXXXFP mask ROM order confirmation form
(2) 80P6N mark specification form
(3) ROM data (EPROM 3 sets)

EOL announced

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
Vcc	Power source voltage		–0.3 to +7	V
AVcc	Analog power source voltage		–0.3 to +7	V
V_I	Input voltage $\overline{RESET}$, CNVss, BYTE		–0.3 to +12	V
V_I	Input voltage $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_7$, $P5_0$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$, V_{REF}, X_{IN}		–0.3 to Vcc + 0.3	V
V_O	Output voltage $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_7$, $P5_0$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$, X_{OUT}, $\overline{E}$		–0.3 to Vcc + 0.3	V
P_d	Power dissipation	Ta = 25 °C	300	mW
T_{opr}	Operating temperature		–20 to +85	°C
T_{stg}	Storage temperature		–40 to +150	°C

RECOMMENDED OPERATING CONDITIONS (Vcc = 5 V ± 10%, Ta = –20 to +85 °C, unless otherwise noted)

Symbol	Parameter		Limits Min.	Limits Typ.	Limits Max.	Unit
Vcc	Power source voltage	f(X_{IN}) : Operating	4.5	5.0	5.5	V
		f(X_{IN}) : Stopped, f(X_{CIN}) = 32.768 kHz	2.7		5.5	
AVcc	Analog power source voltage			Vcc		V
Vss	Power source voltage			0		V
AVss	Analog power source voltage			0		V
V_{IH}	High-level input voltage $P0_0$ – $P0_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_7$, $P5_0$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE, X_{CIN} (Note 3)		0.8 Vcc		Vcc	V
V_{IH}	High-level input voltage $P1_0$ – $P1_7$, $P2_0$ – $P2_7$ (in single-chip mode)		0.8 Vcc		Vcc	V
V_{IH}	High-level input voltage $P1_0$ – $P1_7$, $P2_0$ – $P2_7$ (in memory expansion mode and microprocessor mode)		0.5 Vcc		Vcc	V
V_{IL}	Low-level input voltage $P0_0$ – $P0_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_7$, $P5_0$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE, X_{CIN} (Note 3)		0		0.2Vcc	V
V_{IL}	Low-level input voltage $P1_0$ – $P1_7$, $P2_0$ – $P2_7$ (in single-chip mode)		0		0.2Vcc	V
V_{IL}	Low-level input voltage $P1_0$ – $P1_7$, $P2_0$ – $P2_7$ (in memory expansion mode and microprocessor mode)		0		0.16Vcc	V
$I_{OH(peak)}$	High-level peak output current $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_7$, $P5_0$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$				–10	mA
$I_{OH(avg)}$	High-level average output current $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_7$, $P5_0$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$				–5	mA
$I_{OL(peak)}$	Low-level peak output current $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_3$, $P5_4$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$				10	mA
$I_{OL(peak)}$	Low-level peak output current $P4_4$ – $P4_7$, $P5_0$ – $P5_3$				20	mA
$I_{OL(avg)}$	Low-level average output current $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_3$, $P5_4$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$				5	mA
$I_{OL(avg)}$	Low-level average output current $P4_4$ – $P4_7$, $P5_0$ – $P5_3$				15	mA
f(X_{IN})	Main-clock oscillation frequency (Note 4)				25	MHz
f(X_{CIN})	Sub-clock oscillation frequency			32.768	50	kHz

Notes 1. Average output current is the average value of a 100 ms interval.

2. The sum of $I_{OL(peak)}$ for ports P0, P1, P2, P3, and P8 must be 80 mA or less,
the sum of $I_{OH(peak)}$ for ports P0, P1, P2, P3, and P8 must be 80 mA or less,
the sum of $I_{OL(peak)}$ for ports P4, P5, P6, and P7 must be 100 mA or less, and
the sum of $I_{OH(peak)}$ for ports P4, P5, P6, and P7 must be 80 mA or less.

3. Limits V_{IH} and V_{IL} for X_{CIN} are applied when the sub clock external input selection bit = “1”.

4. The maximum value of f(X_{IN}) = 12.5 MHz when the main clock division selection bit = “1”.

Eol announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

ELECTRICAL CHARACTERISTICS (Vcc = 5 V, Vss = 0 V, Ta = –20 to 85 °C, f(XIN) = 25 MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	High-level output voltage $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_3$, $P4_0$ – $P4_7$, $P5_0$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$	I_{OH} = –10 mA	3			V
V_{OH}	High-level output voltage $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_3$	I_{OH} = –400 μA	4.7			V
V_{OH}	High-level output voltage $P3_0$ – $P3_2$	I_{OH} = –10 mA	3.1			V
		I_{CH} = –400 μA	4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH} = –10 mA	3.4			V
		I_{OH} = –400 μA	4.8			
V_{OL}	Low-level output voltage $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_3$, $P4_0$ – $P4_3$, $P5_4$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_5$, $P8_0$ – $P8_7$	I_{OL} = 10 mA			2	V
V_{OL}	Low-level output voltage $P4_4$ – $P4_7$, $P5_0$ – $P5_3$	I_{OL} = 20 mA			2	V
V_{OL}	Low-level output voltage $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_3$	I_{OL} = 2 mA			0.45	V
V_{OL}	Low-level output voltage $P3_0$ – $P3_2$	I_{OL} = 10 mA			1.9	V
		I_{OL} = 2 mA			0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL} = 10 mA			1.6	V
		I_{OL} = 2 mA			0.4	
$V_{T+} – V_{T–}$	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$ – $TA4_{IN}$, $TB0_{IN}$ – $TB2_{IN}$, $\overline{INT0}$ – $\overline{INT2}$, $\overline{ADTRG}$, $\overline{CTS0}$, $\overline{CTS1}$, $\overline{CTS2}$, CLK_0, CLK_1, CLK_2, $\overline{KI0}$ – $\overline{KI3}$		0.4		1	V
$V_{T+} – V_{T–}$	Hysteresis $\overline{RESET}$		0.2		0.5	V
$V_{T+} – V_{T–}$	Hysteresis X_{IN}		0.1		0.4	V
$V_{T+} – V_{T–}$	Hysteresis X_{CIN} (When external clock is input)		0.1		0.4	V
I_{IH}	High-level input current $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_7$, $P5_0$ – $P5_7$, $P6_0$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I = 5 V			5	μA
I_{IL}	Low-level input current $P0_0$ – $P0_7$, $P1_0$ – $P1_7$, $P2_0$ – $P2_7$, $P3_0$ – $P3_3$, $P4_0$ – $P4_7$, $P5_0$ – $P5_3$, $P6_0$, $P6_1$, $P6_5$ – $P6_7$, $P7_0$ – $P7_7$, $P8_0$ – $P8_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I = 0 V			–5	μA
I_{IL}	Low-level input current $P5_4$ – $P5_7$, $P6_2$ – $P6_4$	V_I = 0 V, without a pull-up transistor			–5	μA
		V_I = 0 V, with a pull-up transistor	–0.25	–0.5	–1.0	mA
V_{RAM}	RAM hold voltage	When clock is stopped.	2			V

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

ELECTRICAL CHARACTERISTICS (Vcc = 5 V, Vss = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
Icc	Power source current	In single-chip mode, output pins are open, and other pins are Vss.	Vcc = 5 V, f(XIN) = 25 MHz (square waveform), f(f2) = 12.5 MHz, f(XCIN) = 32.768 kHz, in operating (Note 1)		9.5	19	mA
			Vcc = 5 V, f(XIN) = 25 MHz (square waveform), (f(f2) = 1.5625 MHz), f(XCIN) = Stopped, in operating (Note 1)		1.3	2.6	mA
			Vcc = 5V, f(XIN) = 25 MHz (square waveform), f(XCIN) = 32.768 kHz, when a WIT instruction is executed (Note 2)		10	20	μA
			Vcc = 5 V, f(XIN) : Stopped, f(XCIN) : 32.768 kHz, in operating (Note 3)		50	100	μA
			Vcc = 5 V, f(XIN) : Stopped, f(XCIN) : 32.768 kHz, when a WIT instruction is executed (Note 4)		5	10	μA
			Ta = 25 °C, when clock is stopped			1	μA
			Ta = 85 °C, when clock is stopped			20	μA

Notes 1. This applies when the main clock external input selection bit = "1", the main clock division selection bit = "0", and the signal output stop bit = "1".
2. This applies when the main clock external input selection bit = "1" and the system clock stop bit at wait state = "1".
3. This applies when CPU and the clock timer are operating with the sub clock (32.768 kHz) selected as the system clock.
4. This applies when the XCOUT drivability selection bit = "0" and the system clock stop bit at wait state = "1".

ECL announced

A–D CONVERTER CHARACTERISTICS

(Vcc = AVcc = 5 V, Vss = AVss = 0 V, Ta = –20 to 85 °C, f(XIN) = 25 MHz (Note), unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	VREF = Vcc			10	Bits
—	Absolute accuracy	VREF = Vcc			± 3	LSB
RLADDER	Ladder resistance	VREF = Vcc	10		25	kΩ
tCONV	Conversion time		9.44			μs
VREF	Reference voltage		2		Vcc	V
VIA	Analog input voltage		0		VREF	V

Note. This applies when the main clock division selection bit = "0" and f(f2) = 12.5 MHz.

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

TIMING REQUIREMENTS (V_{CC} = 5 V ± 10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 25 MHz, unless otherwise noted (Note))

Notes 1. This applies when the main clock division selection bit = "0" and f(f_2) = 12.5 MHz.

2. Input signal's rise/fall time must be 100 ns or less, unless otherwise noted.

External clock input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time (Note 3)	40		ns
$t_{w(H)}$	External clock input high-level pulse width (Note 4)	15		ns
$t_{w(L)}$	External clock input low-level pulse width (Note 4)	15		ns
t_r	External clock rise time		8	ns
t_f	External clock fall time		8	ns

Notes 3. When the main clock division selection bit = "1", the minimum value of tc = 80 ns.

4. When the main clock division selection bit = "1", values of $t_{w(H)}$ / tc and $t_{w(L)}$ / tc must be set to values from 0.45 through 0.55.

Single-chip mode

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(P0D–E)}$	Port P0 input setup time	60		ns
$t_{su(P1D–E)}$	Port P1 input setup time	60		ns
$t_{su(P2D-E)}$	Port P2 input setup time	60		ns
$t_{su(P3D–E)}$	Port P3 input setup time	60		ns
$t_{su(P4D–E)}$	Port P4 input setup time	60		ns
$t_{su(P5D–E)}$	Port P5 input setup time	60		ns
$t_{su(P6D–E)}$	Port P6 input setup time	60		ns
$t_{su(P7D–E)}$	Port P7 input setup time	60		ns
$t_{su(P8D–E)}$	Port P8 input setup time	60		ns
$t_{h(E–P0D)}$	Port P0 input hold time	0		ns
$t_{h(E–P1D)}$	Port P1 input hold time	0		ns
$t_{h(E–P2D)}$	Port P2 input hold time	0		ns
$t_{h(E–P3D)}$	Port P3 input hold time	0		ns
$t_{h(E–P4D)}$	Port P4 input hold time	0		ns
$t_{h(E–P5D)}$	Port P5 input hold time	0		ns
$t_{h(E–P6D)}$	Port P6 input hold time	0		ns
$t_{h(E–P7D)}$	Port P7 input hold time	0		ns
$t_{h(E–P8D)}$	Port P8 input hold time	0		ns

Memory expansion mode and microprocessor mode

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(D–RDE)}$	Data input setup time	32		ns
$t_{su(RDY–\phi 1)}$	$\overline{RDY}$ input setup time	55		ns
$t_{su(HOLD–\phi 1)}$	$\overline{HOLD}$ input setup time	55		ns
$t_{h(RDE–D)}$	Data input hold time	0		ns
$t_{h(\phi 1–RDY)}$	$\overline{RDY}$ input hold time	0		ns
$t_{h(\phi 1–HOLD)}$	$\overline{HOLD}$ input hold time	0		ns

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Timer A input (Count input in event counter mode)

Symbol	parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	80		ns
$t_{w(TAH)}$	TAiIN input high-level pulse width	40		ns
$t_{w(TAL)}$	TAiIN input low-level pulse width	40		ns

Timer A input (Gating input in timer mode)

Symbol	parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time (Note)	320		ns
$t_{w(TAH)}$	TAiIN input high-level pulse width (Note)	160		ns
$t_{w(TAL)}$	TAiIN input low-level pulse width (Note)	160		ns

Note. Limits change depending on f(XIN). Refer to “DATA FORMULAS” on page 72.

Timer A input (External trigger input in one-shot pulse mode)

Symbol	parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time (Note)	320		ns
$t_{w(TAH)}$	TAiIN input high-level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low-level pulse width	80		ns

Note. Limits change depending on f(XIN). Refer to “DATA FORMULAS” on page 72.

Timer A input (External trigger input in pulse width modulation mode)

Symbol	parameter	Limits		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input high-level pulse width	80		ns
$t_{w(TAL)}$	TAiIN input low-level pulse width	80		ns

Timer A input (Up-down input in event counter mode)

Symbol	parameter	Limits		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	2000		ns
$t_{w(UPH)}$	TAiOUT input high-level pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input low-level pulse width	1000		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	400		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	400		ns

Timer A input (Two-phase pulse input in event counter mode)

Symbol	parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAj input cycle time	800		ns
$t_{su(TAjIN-TAjOUT)}$	TAjIN input setup time	200		ns
$t_{su(TAjOUT-TAjIN)}$	TAjOUT input setup time	200		ns

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (one edge count)	80		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width (one edge count)	40		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width (one edge count)	40		ns
$t_{c(TB)}$	TBiIN input cycle time (both edges count)	160		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width (both edges count)	80		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width (both edges count)	80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (Note)	320		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width (Note)	160		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width (Note)	160		ns

Note. Limits change depending on f(X_{IN}). Refer to “DATA FORMULAS” on page 72.

Timer B input (Pulse width measurement mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (Note)	320		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width (Note)	160		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width (Note)	160		ns

Note. Limits change depending on f(X_{IN}). Refer to “DATA FORMULAS” on page 72.

A-D trigger input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	1000		ns
$t_{w(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	125		ns

Serial I/O

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input high-level pulse width	100		ns
$t_{w(CKL)}$	CLKi input low-level pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

External interrupt $\overline{INT_i}$ input, key input interrupt $\overline{KI_i}$ input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT_i}$ input high-level pulse width	250		ns
$t_{w(INL)}$	$\overline{INT_i}$ input low-level pulse width	250		ns
$t_{w(KIL)}$	$\overline{KI_i}$ input low-level pulse width	250		ns

SOL announced

DATA FORMULAS

Timer A input (Gating input in timer mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} input cycle time	$\frac{8 \times 10^9}{2 \cdot f(f_2)}$		ns
$t_{w(TAH)}$	TAi_{IN} input high-level pulse width	$\frac{4 \times 10^9}{2 \cdot f(f_2)}$		ns
$t_{w(TAL)}$	TAi_{IN} input low-level pulse width	$\frac{4 \times 10^9}{2 \cdot f(f_2)}$		ns

Timer A input (External trigger input in one-shot pulse mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} input cycle time	$\frac{8 \times 10^9}{2 \cdot f(f_2)}$		ns

Timer B input (In pulse period measurement mode or pulse width measurement mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	$\frac{8 \times 10^9}{2 \cdot f(f_2)}$		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	$\frac{4 \times 10^9}{2 \cdot f(f_2)}$		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	$\frac{4 \times 10^9}{2 \cdot f(f_2)}$		ns

Note. $f(f_2)$ represents the clock f_2 frequency.
For the relation to the main clock and sub clock, refer to Table 10.

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

SWITCHING CHARACTERISTICS (V_{CC} = 5 V ± 10%, V_{SS} = 0 V, Ta = –20 to 85°C, f(X_{IN}) = 25 MHz (Note), unless otherwise noted)

Symbol	Parameter	Test conditions	Limits		Unit
			Min.	Max.	
$t_{d(E-P0Q)}$	Port P0 data output delay time	Fig. 73		80	ns
$t_{d(E-P1Q)}$	Port P1 data output delay time			80	ns
$t_{d(E-P2Q)}$	Port P2 data output delay time			80	ns
$t_{d(E-P3Q)}$	Port P3 data output delay time			80	ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			80	ns

Note. This applies when the main clock division selection bit = "0" and f(f_2) = 12.5 MHz.

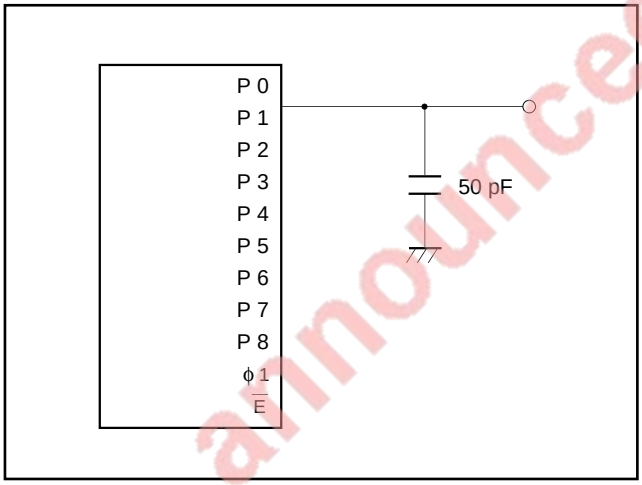


Fig. 73 Measuring circuit for ports P0 – P8 and φ 1

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Memory expansion mode and microprocessor mode

(V_{CC} = 5 V ± 10%, V_{SS} = 0 V, Ta = 25 °C, f(X_{IN}) = 25 MHz (Note 1), unless otherwise noted)

Symbol	Parameter	(Note 2) Wait mode	Test conditions	Limits Min.	Limits Max.	Unit
$t_{d(CS-WE)}$ $t_{d(CS-RDE)}$	Chip-select output delay time	No wait Wait 1	Fig. 73	12		ns
		Wait 0		87		ns
$t_{h(WE-CS)}$ $t_{h(RDE-CS)}$	Chip-select hold time			4		ns
$t_{d(An-WE)}$ $t_{d(An-RDE)}$	Address output delay time	No wait Wait 1		12		ns
		Wait 0		87		ns
$t_{d(A-WE)}$ $t_{d(A-RDE)}$	Address output delay time	No wait Wait 1		12		ns
		Wait 0		75		ns
$t_{h(WE-An)}$ $t_{h(RDE-An)}$	Address hold time			18		ns
$t_{w(ALE)}$	ALE pulse width	No wait Wait 1		22		ns
		Wait 0		57		ns
$t_{su(A-ALE)}$	Address output set up time	No wait Wait 1		5		ns
		Wait 0		45		ns
$t_{h(ALE-A)}$	Address hold time	No wait Wait 1		9		ns
		Wait 0		15		ns
$t_{d(ALE-WE)}$ $t_{d(ALE-RDE)}$	ALE output delay time	No wait Wait 1		4		ns
		Wait 0		10		ns
$t_{d(WE-DQ)}$	Data output delay time				45	ns
$t_{h(WE-DQ)}$	Data hold delay time			18		ns
$t_{w(WE)}$	$\overline{WEL}$/$\overline{WEH}$ pulse width	No wait		50		ns
		Wait 1 Wait 0		130		ns
$t_{pxz(RDE-DZ)}$	Floating start delay time				5	ns
$t_{pzx(RDE-DZ)}$	Floating release delay time			20		ns
$t_{w(RDE)}$	$\overline{RDE}$ pulse width	No wait		48		ns
		Wait 1 Wait 0		128		ns
$t_{d(RSMP-WE)}$ $t_{d(RSMP-RDE)}$	$\overline{RSMP}$ output delay time			10		ns
$t_{h(\phi 1-RSMP)}$	$\overline{RSMP}$ hold time			0		ns
$t_{d(WE-\phi 1)}$ $t_{d(RDE-\phi 1)}$	ϕ_1 output delay time			0	18	ns
$t_{d(\phi 1-HLDA)}$	$\overline{HLDA}$ output delay time				50	ns

Notes 1. This applies when the main clock division selection bit = "0" and f(f_2) = 12.5 MHz.

2. No wait : Wait bit = "1".
Wait 1 : The external memory area is accessed with wait bit = "0" and wait selection bit = "1".
Wait 0 : The external memory area is accessed with wait bit = "0" and wait selection bit = "0".

EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

Memory expansion mode and microprocessor mode

Bus timing data formulas (V_{CC} = 5 V ± 10%, V_{SS} = 0 V, Ta = –20 to 85 °C, f(X_{IN}) = 25 MHz (Max., Note1), unless otherwise noted)

Symbol	Parameter	Wait mode	Limits Min.	Limits Max.	Unit
$t_{d(CS–WE)}$ $t_{d(CS–RDE)}$	Chip-select output delay time	No wait Wait 1	$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 28$		ns
		Wait 0	$\frac{3 \times 10^9}{2 \cdot f(f_2)} - 33$		ns
$t_{h(WE–CS)}$ $t_{h(RDE–CS)}$	Chip-select hold time		4		ns
$t_{d(An–WE)}$ $t_{d(An–RDE)}$	Address output delay time	No wait Wait 1	$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 28$		ns
		Wait 0	$\frac{3 \times 10^9}{2 \cdot f(f_2)} - 33$		ns
$t_{d(A–WE)}$ $t_{d(A–RDE)}$	Address output delay time	No wait Wait 1	$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 28$		ns
		Wait 0	$\frac{3 \times 10^9}{2 \cdot f(f_2)} - 45$		ns
$t_{h(WE–An)}$ $t_{h(RDE–An)}$	Address hold time		$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 22$		ns
$t_{w(ALE)}$	ALE pulse width	No wait Wait 1	$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 18$		ns
		Wait 0	$\frac{2 \times 10^9}{2 \cdot f(f_2)} - 23$		ns
$t_{su(A–ALE)}$	Address output set up time	No wait Wait 1	$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 35$		ns
		Wait 0	$\frac{2 \times 10^9}{2 \cdot f(f_2)} - 35$		ns
$t_{h(ALE–A)}$	Address hold time	No wait Wait 1	9		ns
		Wait 0	$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 25$		ns
$t_{d(ALE–WE)}$ $t_{d(ALE–RDE)}$	ALE output delay time	No wait Wait 1	4		ns
		Wait 0	$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 30$		ns
$t_{d(WE–DQ)}$	Data output delay time			45	ns
$t_{h(WE–DQ)}$	Data hold time		$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 22$		ns
$t_{w(WE)}$	$\overline{WEL}$/$\overline{WEH}$ pulse width	No wait	$\frac{2 \times 10^9}{2 \cdot f(f_2)} - 30$		ns
		Wait 1 Wait 0	$\frac{4 \times 10^9}{2 \cdot f(f_2)} - 30$		ns
$t_{pxz(RDE–DZ)}$	Floating start delay time			5	ns
$t_{pzx(RDE–DZ)}$	Floating release delay time		$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 20$		ns
$t_{w(RDE)}$	$\overline{RDE}$ pulse width	No wait	$\frac{2 \times 10^9}{2 \cdot f(f_2)} - 32$		ns
		Wait 1 Wait 0	$\frac{4 \times 10^9}{2 \cdot f(f_2)} - 32$		ns
$t_{d(RSMP–WE)}$ $t_{d(RSMP–RDE)}$	$\overline{RSMP}$ output delay time		$\frac{1 \times 10^9}{2 \cdot f(f_2)} - 30$		ns
$t_{h(\phi-1–RSMP)}$	$\overline{RSMP}$ hold time		0		ns
$t_{d(WE–\phi 1)}$ $t_{d(RDE–\phi 1)}$	$\phi 1$ output delay time		0	18	ns

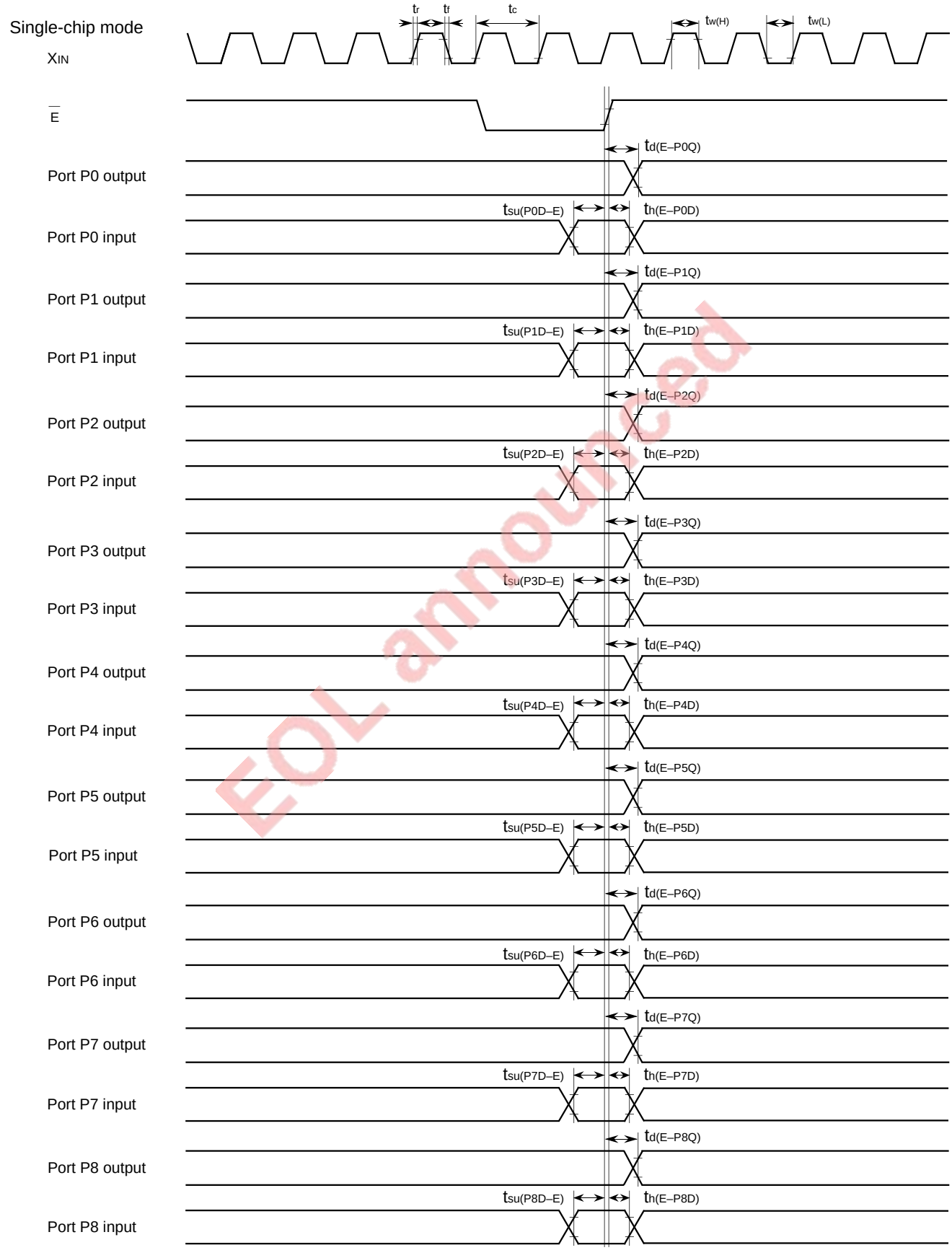
Notes 1. This applies when the main-clock division selection bit = “0”.

2. f(f_2) represents the clock f_2 frequency.
For the relation to the main clock and sub clock, refer to Table 10.

EOL announced

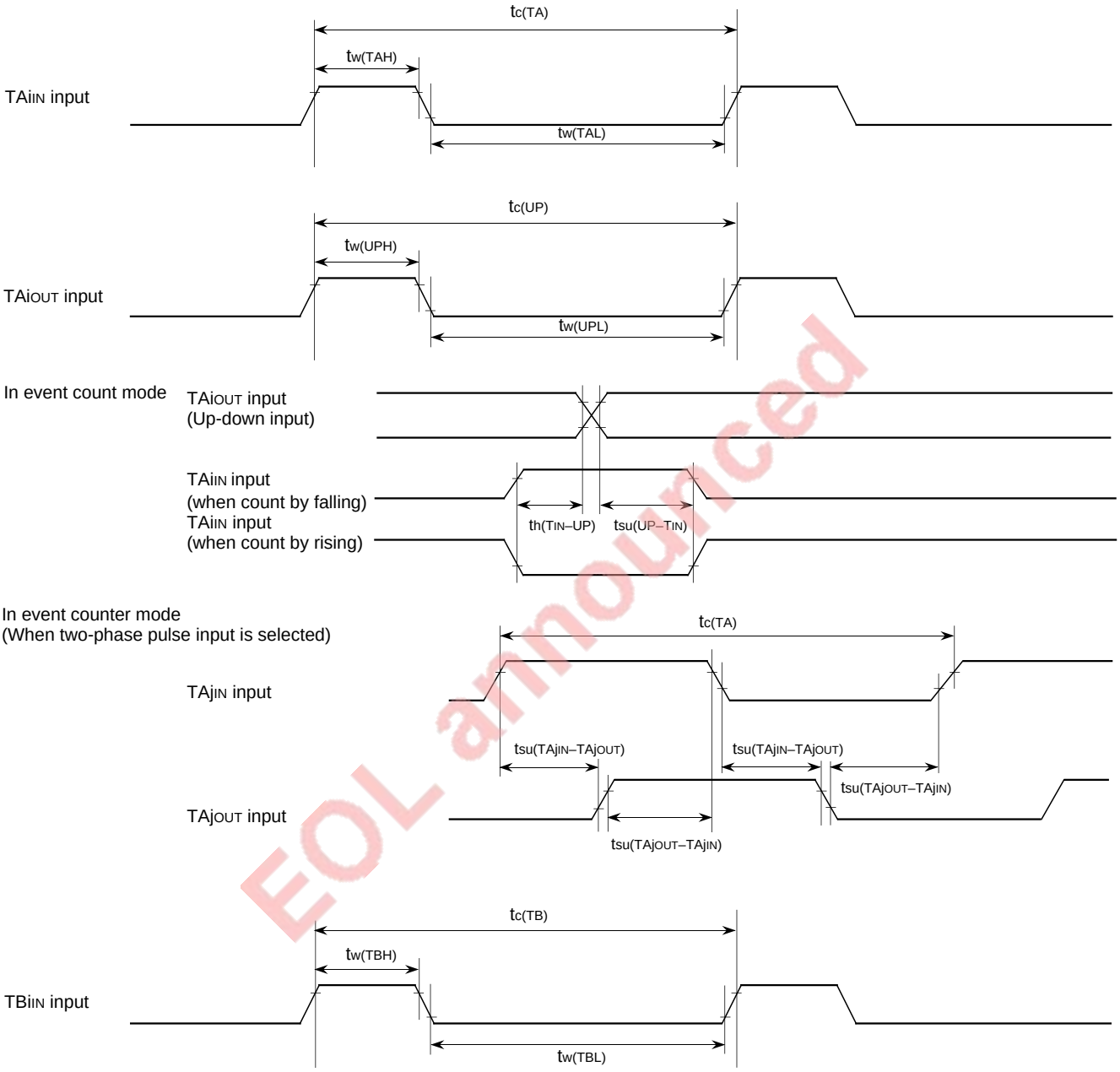
TIMING DIAGRAM

Single-chip mode



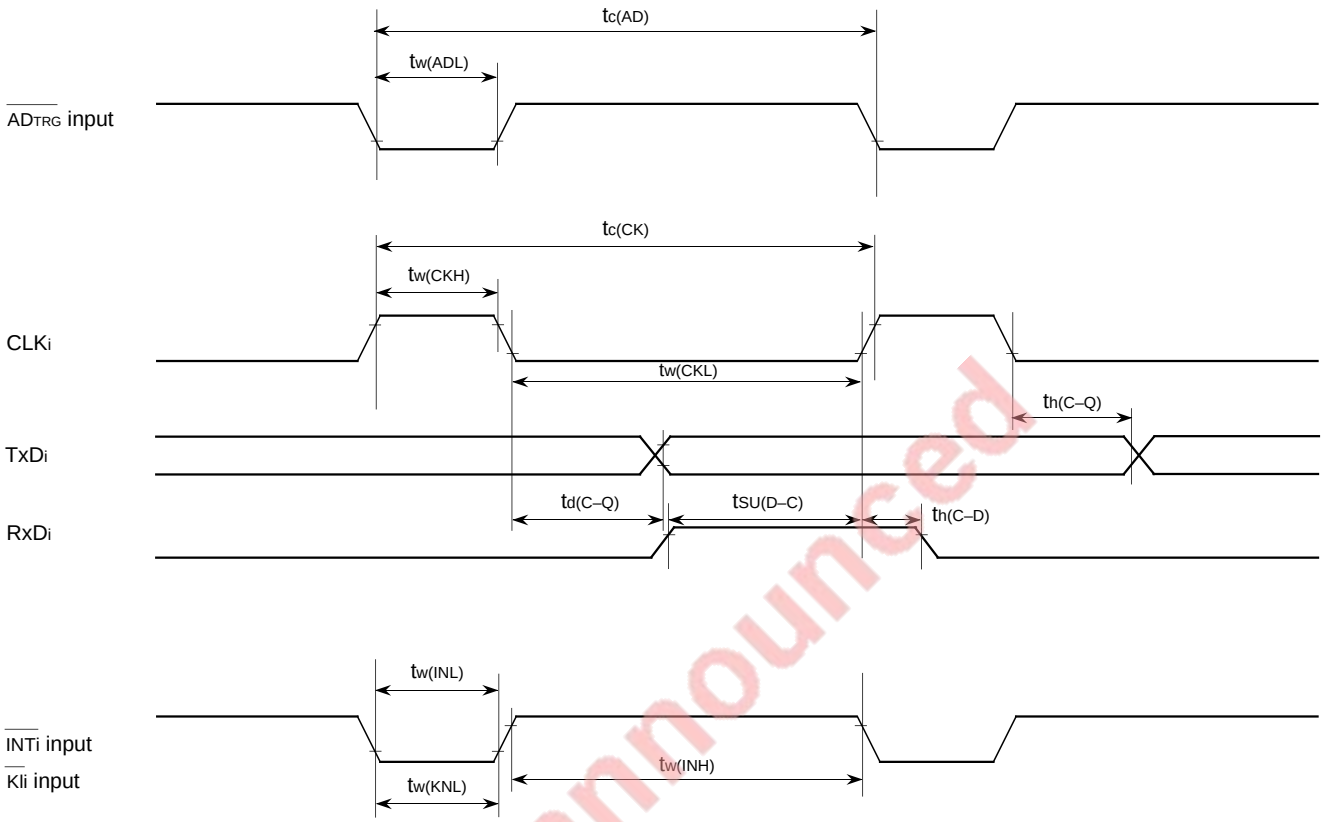
EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.



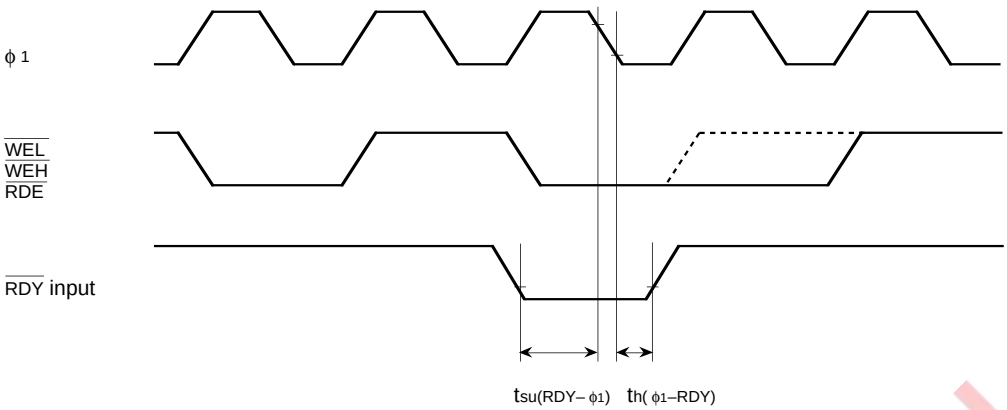
EOL announced

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

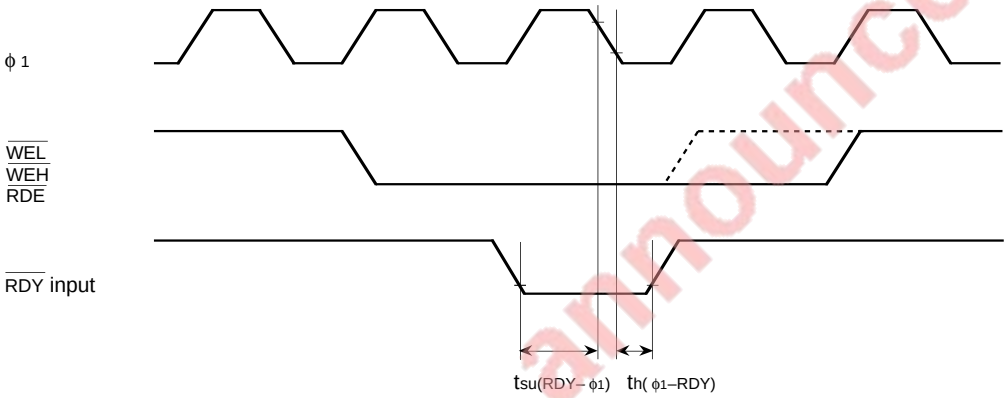


EOL announced

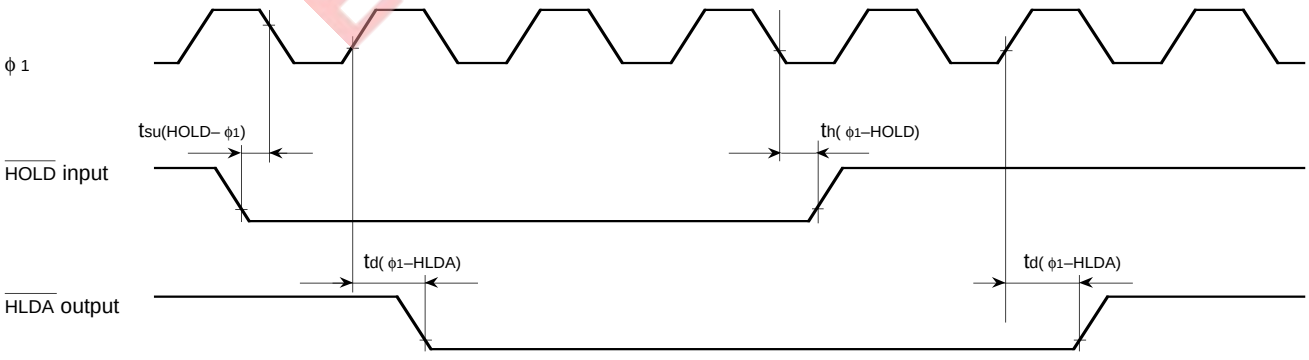
Memory expansion mode and microprocessor mode
(When wait bit = “1”)



(When wait bit = “0”)



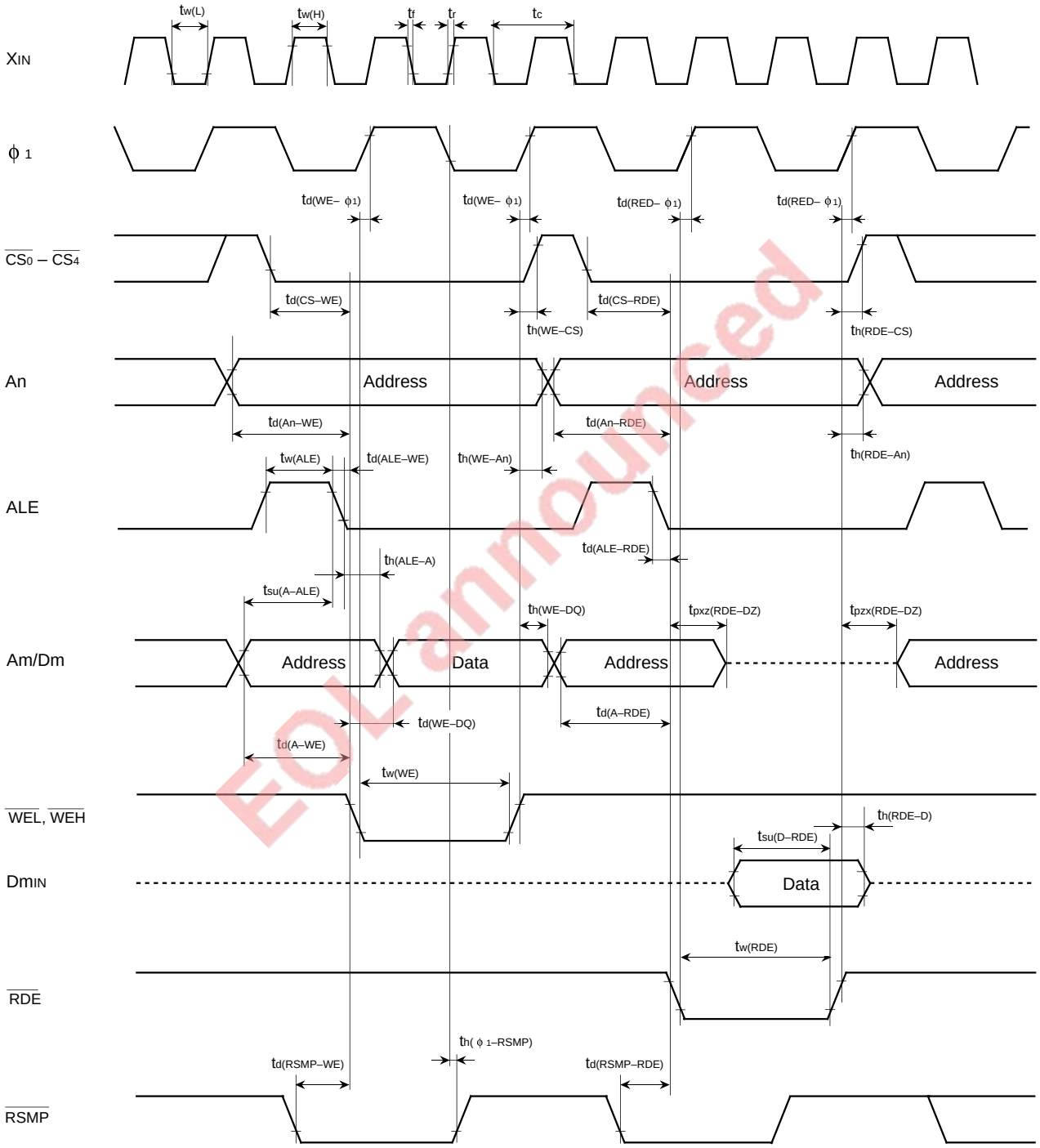
(When wait bit = “1” or “0” in common)



Test conditions
- V_{CC} = 5 V ± 10%
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V

EOL announced

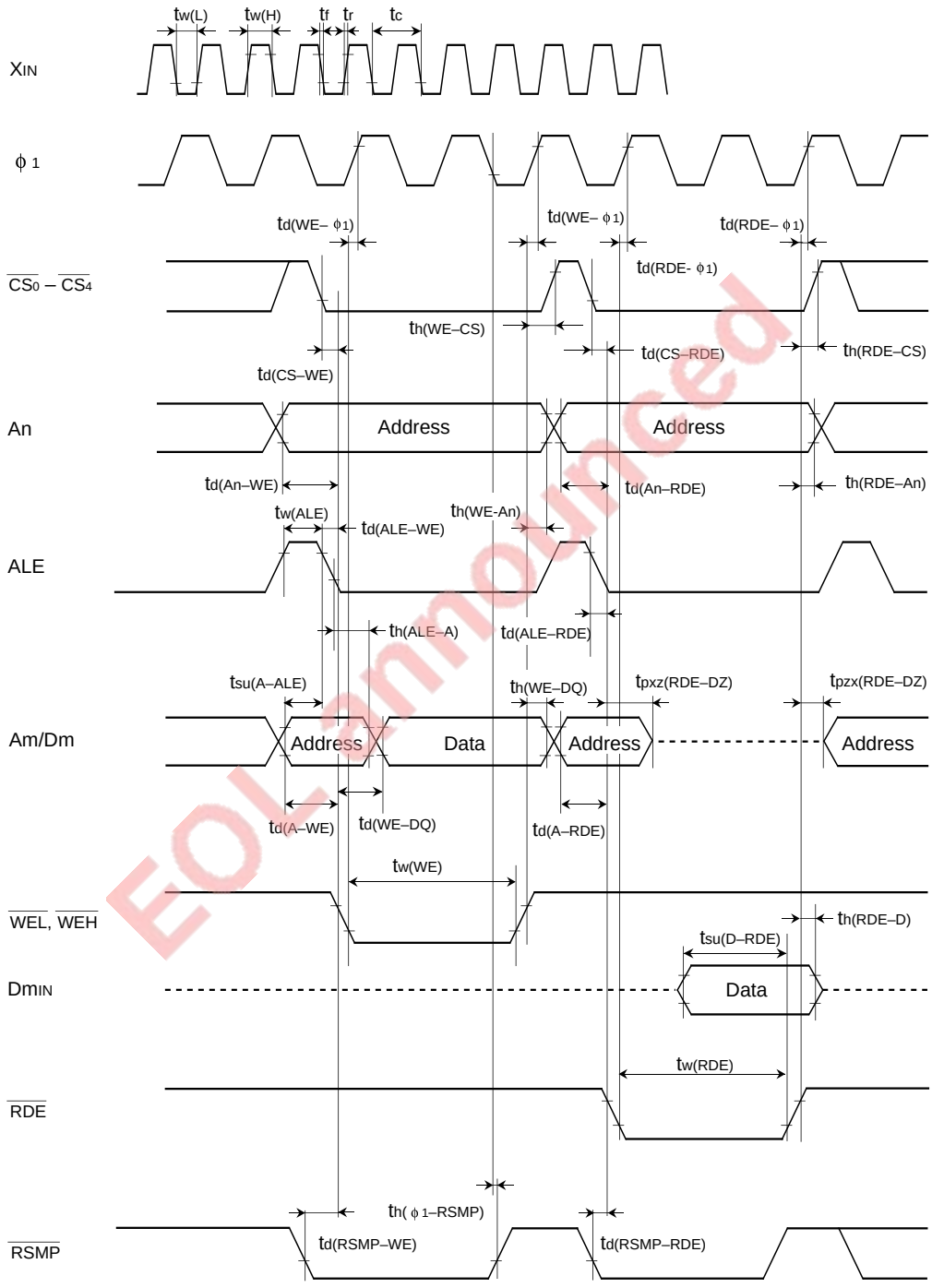
Memory expansion mode and microprocessor mode
(No wait : When wait bit = "1")



EOL announced

Test condition
- Vcc = 5 V ± 10%
- Output timing voltage : VIL = 0.8 V, VIH = 2.0 V
- Data input DmIN : VIL = 0.8 V, VIH = 2.5 V

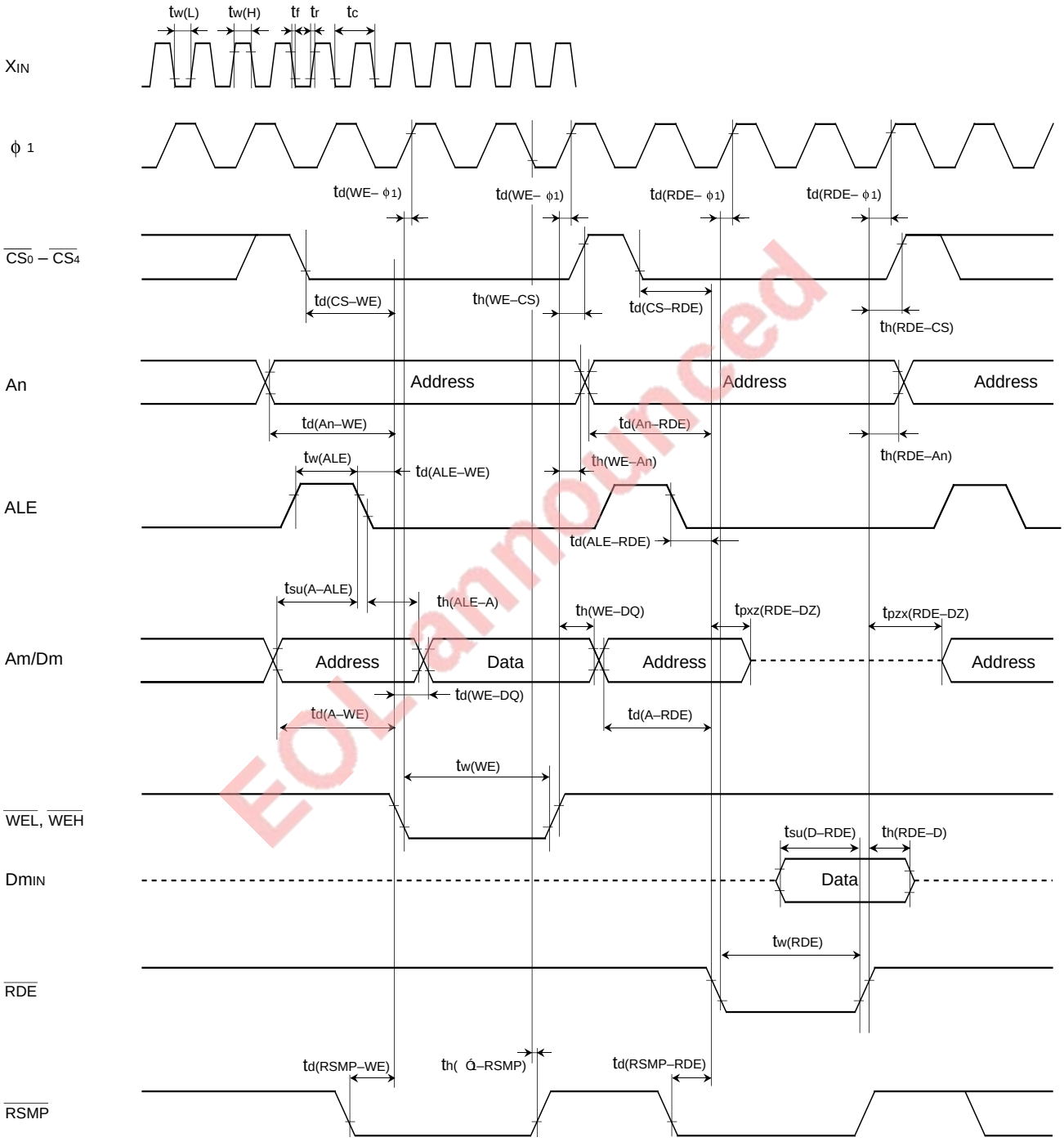
Memory expansion mode and microprocessor mode
(Wait 1 : The external area is accessed when wait bit = “0” and wait selection bit = “1”.)



EOL announced

Test condition
- Vcc = 5 V ± 10%
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V
- Data input D_{MIN} : V_{IL} = 0.8 V, V_{IH} = 2.5 V

Memory expansion mode and microprocessor mode
(Wait 0 : The external memory area is accessed when wait bit = “0” and wait selection bit = “0”.)



EOL announced

Test conditions
- Vcc = 5 V ± 10%
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V
- Data input DmIN : VIL = 0.8 V, VIH = 2.5 V

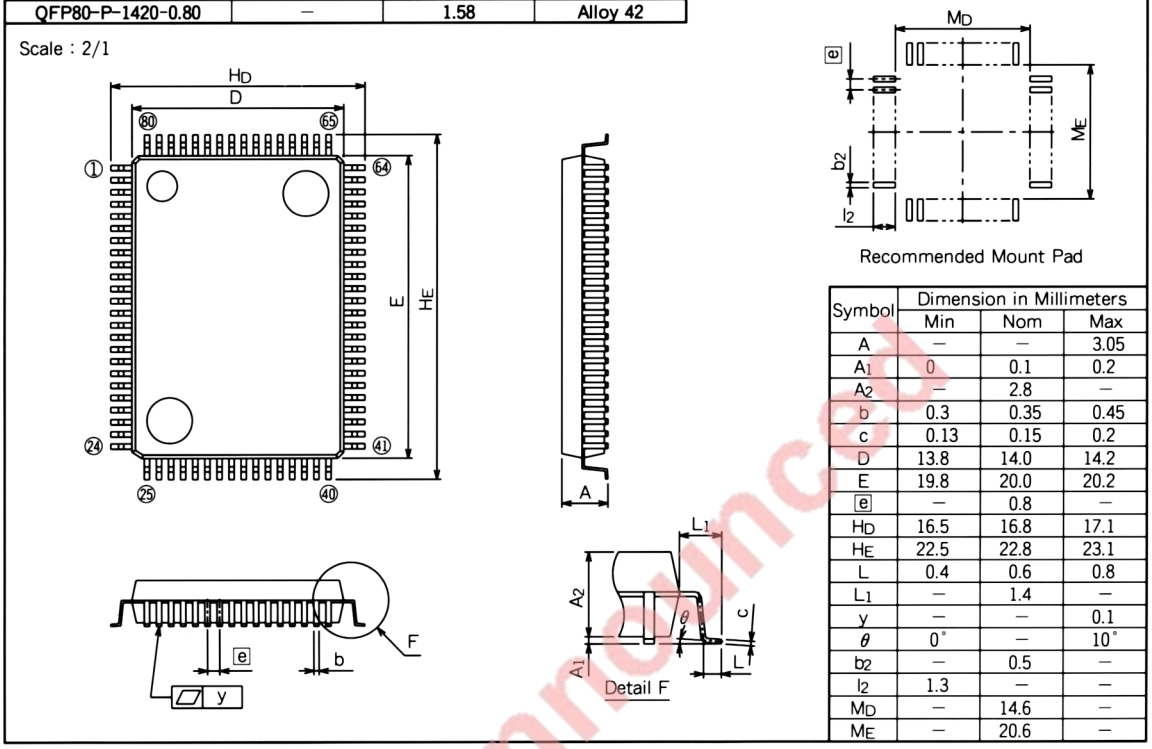
PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

PACKAGE OUTLINE

80P6N−A — Plastic 80pin 14×20mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP80-P-1420-0.80	—	1.58	Alloy 42

Scale : 2/1



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	—	—	3.05
A_1	0	0.1	0.2
A_2	—	2.8	—
b	0.3	0.35	0.45
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	—	0.8	—
H_D	16.5	16.8	17.1
H_E	22.5	22.8	23.1
L	0.4	0.6	0.8
L_1	—	1.4	—
y	—	—	0.1
θ	0°	—	10°
b_2	—	0.5	—
l_2	1.3	—	—
M_D	—	14.6	—
M_E	—	20.6	—

EOL announced

GZZ–SH00–79B<84A0>

Mask ROM number	

7700 FAMILY MASK ROM ORDER CONFIRMATION FORM
SINGLE-CHIP 16-BIT MICROCOMPUTER
M37735MHBXXXFP
MITSUBISHI ELECTRIC

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked ※

※	Customer	Company name	TEL ( )	Issuance signatures	Responsible officer	Supervisor
		Date issued	Date:			

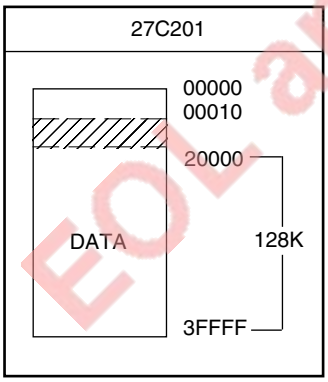
EOL announced

※1. Confirmation

Specify the name of the product being ordered.
Three sets of EPROMs are required for each pattern (Check @ in the appropriate box).
If at least two of the three sets of EPROMs submitted contain the identical data, we will produce masks based on this data.
We shall assume the responsibility for errors only if the mask ROM data on the products we produce differ from this data.
Thus, the customer must be especially careful in verifying the data contained in the EPROMs submitted.

Checksum code for entire EPROM areas [ ][ ][ ][ ] (hexadecimal notation)

EPROM Type :



Note : Make sure that address $01FFFF_{16}$ of the microcomputer's internal ROM corresponds to address $3FFFF_{16}$ of EPROM.

(1) Set "FF_{16}" in the shaded area.
(2) Address 0_{16} to 10_{16} are the area for storing the data on model designation and options.This area must be written with the data shown below.
Details for option data are given next in the section describing the STP instruction option.
Address and data are written in hexadecimal notation.

	Address		Address		Address
4D	0	42	8	Option data	10
33	1	FF	9		
37	2	FF	A		
37	3	FF	B		
33	4	FF	C		
35	5	FF	D		
4D	6	FF	E		
48	7	FF	F		

※2. STP instruction option

One of the following sets of data should be written to the option data address (10_{16}) of the EPROM you have ordered.
Check @ in the appropriate box.

☐ STP instruction enable 01_{16} Address 10_{16}
☐ STP instruction disable 00_{16} Address 10_{16}

※3. Mark specification

Mark specification must be submitted using the correct form for the type of package being ordered fill out the appropriate 80P6N Mark Specification Form (for M37735MHBXXXFP) and attach to the Mask ROM Order Confirmation Form.

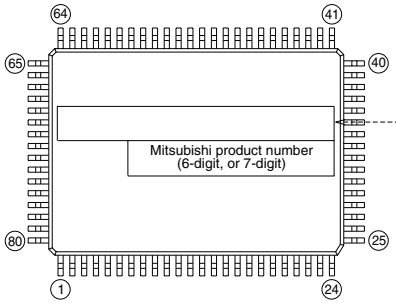
※4. Comments

80P6N (80-PIN QFP) MARK SPECIFICATION FORM

Mitsubishi IC catalog name

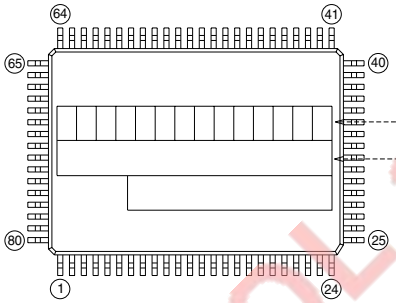
Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



Mitsubishi IC catalog name

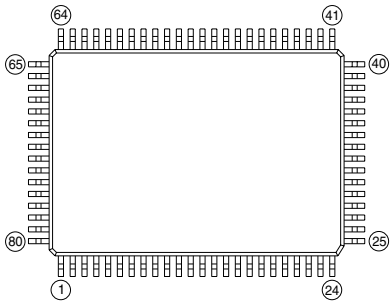
B. Customer's Parts Number + Mitsubishi IC Catalog Name



Customer's Parts Number
Note : The fonts and size of characters are standard Mitsubishi type.
Mitsubishi IC catalog name
Notes 1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 14 alphanumeric characters for capital letters, hyphens, commas, periods and so on.

EOL announced

C. Special Mark Required



Notes1 : If special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated technically as close as possible.
Mitsubishi product number (6-digit, or 7-digit) and Mask ROM number (3-digit) are always marked for sorting the products.
2 : If special character fonts (e,g., customer's trade mark logo) must be used in Special Mark, check the box below.
For the new special character fonts, a clean font original (ideally logo drawing) must be submitted.

Special character fonts required ☐

PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.

M37735MHBXXXFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

EOL announced

Renesas Technology Corp.
Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

Keep safety first in your circuit designs!

- Mitsubishi Electric Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

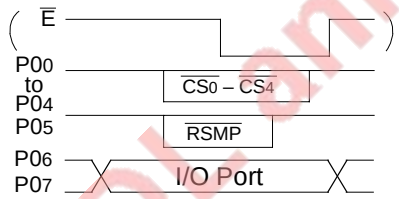
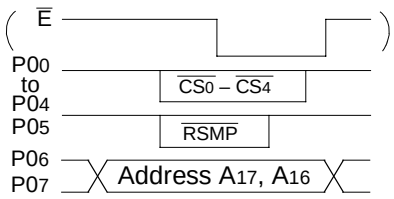
Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams and charts, represent information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

© 1996 MITSUBISHI ELECTRIC CORP.
H-LF425-A KI-9606 Printed in Japan (ROD) II
New publication, effective Jun. 1996.
Specifications subject to change without notice.

RENESAS
Renesas Technology Corp.

REVISION DESCRIPTION LIST	M37735MHBXXXFP Datasheet

<table>
<tr><th>Rev. No.</th><th>Revision Description</th><th>Rev. date</th></tr>
<tr><td>1.00</td><td>First Edition</td><td>970604</td></tr>
<tr><td>1.01</td><td>The following are added:
• MASK ROM ORDER CONFIRMATION FORM
• MARK SPECIFICATION FORM</td><td>980526</td></tr>
<tr><td>2.00</td><td>The following are revised:
<table>
<tr><th>Page</th><th>Previous Version</th><th>Revised Version</th></tr>
<tr><td>P5
Right column
Line 5</td><td>Additionally, the internal ROM area can be modified by software.</td><td>Additionally, the internal ROM and RAM area can be modified by software.</td></tr>
<tr><td>P5
Fig. 1</td><td>Note. Internal ROM area can be modified. (Refer to the section on ROM area modification function.)</td><td>Note. Internal ROM and RAM area can be modified. (Refer to the section on ROM area modification function.)</td></tr>
<tr><td>P9
Right column
Line 12</td><td>The CPU operates on an internal clock ϕ's frequency which is obtained by dividing the external clock frequency f(X_{IN}) by two.</td><td>The CPU operates on an internal clock ϕ's frequency.</td></tr>
<tr><td>P50
Fig. 58</td><td><Port P0> Memory expansion mode

</td><td><Port P0> Memory expansion mode

</td></tr>
<tr><td>P53
Table 7</td><td>
<table>
<tr><th colspan="2">Access address</th></tr>
<tr><th></th><th>Microprocessor mode</th></tr>
<tr><td></td><td>00 0880₁₆
to
00 7FFF₁₆</td></tr>
<tr><td></td><td>00 8000₁₆
to
03 FFFF₁₆</td></tr>
<tr><td></td><td>04 0000₁₆
to
07 FFFF₁₆</td></tr>
<tr><td></td><td>08 0000₁₆
to
0B FFFF₁₆</td></tr>
<tr><td></td><td>0C 0000₁₆
to
0F FFFF₁₆</td></tr>
</table>
</td><td>
<table>
<tr><th colspan="2">Access address</th></tr>
<tr><th></th><th>Microprocessor mode</th></tr>
<tr><td></td><td>00 1000₁₆
to
00 7FFF₁₆</td></tr>
<tr><td></td><td>00 8000₁₆
to
03 FFFF₁₆</td></tr>
<tr><td></td><td>04 0000₁₆
to
07 FFFF₁₆</td></tr>
<tr><td></td><td>08 0000₁₆
to
0B FFFF₁₆</td></tr>
<tr><td></td><td>0C 0000₁₆
to
0F FFFF₁₆</td></tr>
</table>
</td></tr>
</table>
</td><td>980731</td></tr>
</table>

EOL announced

REVISION DESCRIPTION LIST	M37735MHBXXXFP Datasheet

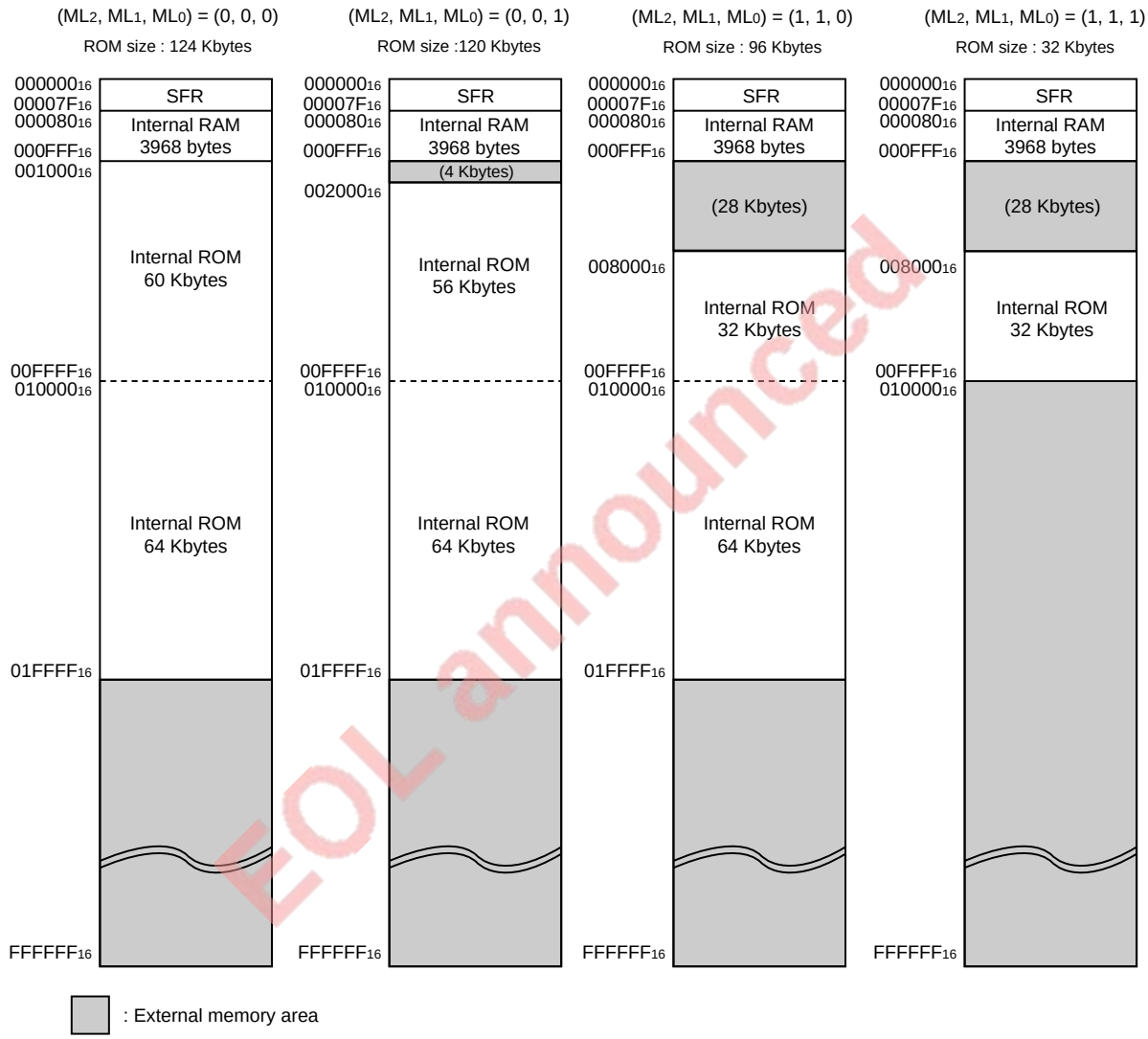
<table>
<tr><th>Rev. No.</th><th colspan="3">Revision Description</th><th>Rev. date</th></tr>
<tr><td>2.00</td><td>Page</td><td>Previous Version</td><td>Revised Version</td><td>980731</td></tr>
<tr><td></td><td>P63
Left column
Line 2</td><td>The internal ROM size and its address area of the M37735MHBXXXFP can be modified by the memory allocation control register's bits 0,1 and 2 shown in Figure 70.
Figure 72 shows the memory allocation in which the internal ROM size and its address area are modified.
Make sure to write data in the memory allocation control register as the flow shown in Figure 71.
This ROM area modification function is valid in memory expansion mode and single-chip mode.</td><td>The internal ROM size and RAM size of the M37735MHBXXXFP can be modified by the memory allocation control register's bits 0,1 and 2 shown in Figure 70.
Figure 72 shows the memory allocation in which the internal ROM size and RAM size are modified.
Make sure to write data in the memory allocation control register as the flow shown in Figure 71.
This ROM area modification function is valid in memory expansion mode and single-chip mode.
Table 12 shows the relationship between the memory allocation selection bits and addresses corresponding to chip-select signals $\overline{CS_0}$ and $\overline{CS_1}$.</td><td></td></tr>
<tr><td></td><td>P63
Fig. 70</td><td>Memory allocation selection bits
ROM size (ROM area)
0 0 0 : 124 Kbytes (addresses 001000_{16} – $01FFFF_{16}$)
0 0 1 : 120 Kbytes (addresses 002000_{16} – $01FFFF_{16}$)
1 1 0 : 96 Kbytes (addresses 008000_{16} – $01FFFF_{16}$)
1 1 1 : 32 Kbytes (addresses 008000_{16} – $00FFFF_{16}$)</td><td>Memory allocation selection bits
ROM size RAM size
0 0 0 : 124 Kbytes 3968 bytes
0 0 1 : 120 Kbytes 3968 bytes
0 1 0 : 60 Kbytes 2048 bytes
1 0 0 : 32 Kbytes 2048 bytes
1 0 1 : 16 Kbytes 2048 bytes
1 1 0 : 96 Kbytes 3968 bytes</td><td></td></tr>
<tr><td></td><td>P64, Fig. 72</td><td>Refer to page (3).</td><td>Refer to page (4).</td><td></td></tr>
<tr><td></td><td>P64
Table 12</td><td colspan="2">Previous Version
<table>
<tr><th colspan="3">Memory allocation selection bits</th><th rowspan="2">Internal ROM area</th><th colspan="2">Access address</th></tr>
<tr><th>ML_2</th><th>ML_1</th><th>ML_0</th><th>$\overline{CS_0}$</th><th>$\overline{CS_1}$</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>001000_{16} – $01FFFF_{16}$</td><td>————</td><td>020000_{16} – $03FFFF_{16}$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>002000_{16} – $01FFFF_{16}$</td><td>001000_{16} – $001FFF_{16}$</td><td>020000_{16} – $03FFFF_{16}$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>008000_{16} – $01FFFF_{16}$</td><td>001000_{16} – $007FFF_{16}$</td><td>020000_{16} – $03FFFF_{16}$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>008000_{16} – $00FFFF_{16}$</td><td>001000_{16} – $007FFF_{16}$</td><td>010000_{16} – $03FFFF_{16}$</td></tr>
</table>
Revised Version
<table>
<tr><th colspan="3">Memory allocation selection bits</th><th rowspan="2">Internal ROM area</th><th colspan="2">Access address</th></tr>
<tr><th>ML_2</th><th>ML_1</th><th>ML_0</th><th>$\overline{CS_0}$</th><th>$\overline{CS_1}$</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>001000_{16} – $01FFFF_{16}$</td><td>————</td><td>020000_{16} – $03FFFF_{16}$</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>002000_{16} – $01FFFF_{16}$</td><td>001000_{16} – $001FFF_{16}$</td><td>020000_{16} – $03FFFF_{16}$</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>001000_{16} – $00FFFF_{16}$</td><td>000880_{16} – $000FFF_{16}$</td><td>010000_{16} – $03FFFF_{16}$</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>008000_{16} – $00FFFF_{16}$</td><td>000880_{16} – $007FFF_{16}$</td><td>010000_{16} – $03FFFF_{16}$</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>$00C000_{16}$ – $00FFFF_{16}$</td><td>000880_{16} – $007FFF_{16}$</td><td>008000_{16} – $00BFFF_{16}$
010000_{16} – $03FFFF_{16}$</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>008000_{16} – $01FFFF_{16}$</td><td>001000_{16} – $007FFF_{16}$</td><td>020000_{16} – $03FFFF_{16}$</td></tr>
</table></td><td></td></tr>
<tr><td></td><td>P65
Right column
Line 3</td><td>Refer to the MITSUBISHI SEMICONDUCTORS DATA BOOK SINGLE-CHIP 16-BIT MICROCOMPUTERS for the details of each addressing mode.

MACHINE INSTRUCTION LIST
....Refer to the MITSUBISHI SEMICONDUCTORS DATA BOOK SINGLE-CHIP 16-BIT MICROCOMPUTERS for details.</td><td>Refer to the "7700 Family Software Manual" for the details.

MACHINE INSTRUCTION LIST
...Refer to the "7700 Family Software Manual" for the details.</td><td></td></tr>
<tr><td></td><td>P66
RECOMMENDED OPERATING CONDITIONS</td><td>Previous Version
<table>
<tr><td>$I_{OL(peak)}$</td><td>Low-level average output current I</td></tr>
<tr><td>$f(X_{IN})$</td><td>Main-clock oscillation frequency (f</td></tr>
<tr><td>$f(X_{CIN})$</td><td>Sub-clock oscillation frequency</td></tr>
</table></td><td>Revised Version
<table>
<tr><td>$I_{OL(avg)}$</td><td>Low-level average output current P</td></tr>
<tr><td>$f(X_{IN})$</td><td>Main-clock oscillation frequency (N</td></tr>
<tr><td>$f(X_{CIN})$</td><td>Sub-clock oscillation frequency</td></tr>
</table></td><td></td></tr>
</table>

EOL announced

Previous version

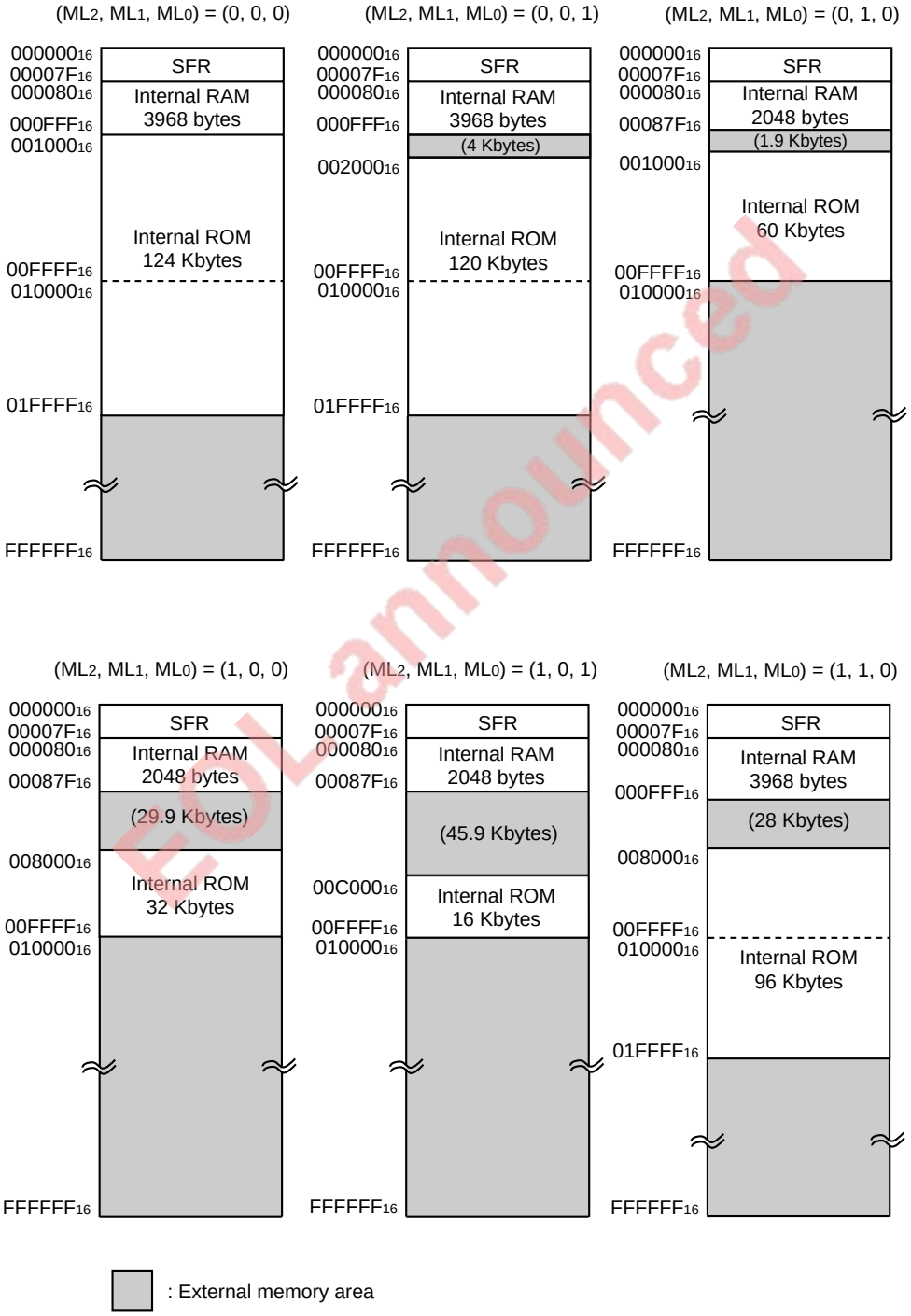


Note. Banks 10₁₆–FF₁₆ cannot be accessed in the 7735 group.

Fig. 72 Memory allocation (modification of internal ROM area by memory allocation selection bit)

EOL announced

Revised version



Note. Banks 10_{16}–FF_{16} cannot be accessed in the 7735 group.

Fig. 72 Memory allocation (modification of internal ROM and RAM area by memory allocation selection bits)

EOL announced