PRELIMINARY PRODUCT INFORMATION

MOS INTEGRATED CIRCUIT

μPD789046

8-BIT SINGLE-CHIP MICROCONTROLLER

The μPD789046 is a μPD789046 sub-series product of the 78K/0S series.

The μPD789046 features an 8-bit CPU, I/O ports, timers, a serial interface, and interrupt control circuits.

In addition, a flash memory product (μPD78F9046) that can operate within the same voltage range as the masked ROM models, and a range of related development tools are being developed.

The functions of the μPD789046 are described in the following user's manuals. Refer to these manuals when designing a system based on the μPD789046.

μPD789046 Sub-Series User's Manual	**:**	**Under creation**
78K/0S Series User's Manual, Instruction	**:**	**U11047E**

FEATURES

- ROM and RAM sizes

  Internal ROM: 16 Kbytes

  Internal high-speed RAM: 512 bytes
- Variable minimum instruction execution time: From high-speed (0.4 μs: With the main system clock running at 5.0 MHz) to very low-speed (122 μs: With the subsystem clock running at 32.768 kHz)
- 34 I/O ports
- Serial interface channel:

  Switchable between three-wire serial I/O and UART modes
- Four timers:
  - 16-bit timer counter
  - 8-bit timer/event counter
  - Clock timer
  - Watchdog timer
- Power supply voltage: V_{DD} = 1.8 to 5.5 V

APPLICATIONS

Cordless phones, etc.

The information contained in this document is being issued in advance of the production cycle for the device. The parameters for the device may change before final production or NEC Corporation, at its own discretion, may withdraw the device prior to its production.

Document No. U13380EJ1V0PM00 (1st edition)
Date Published May 1998 J CP(K)
Printed in Japan

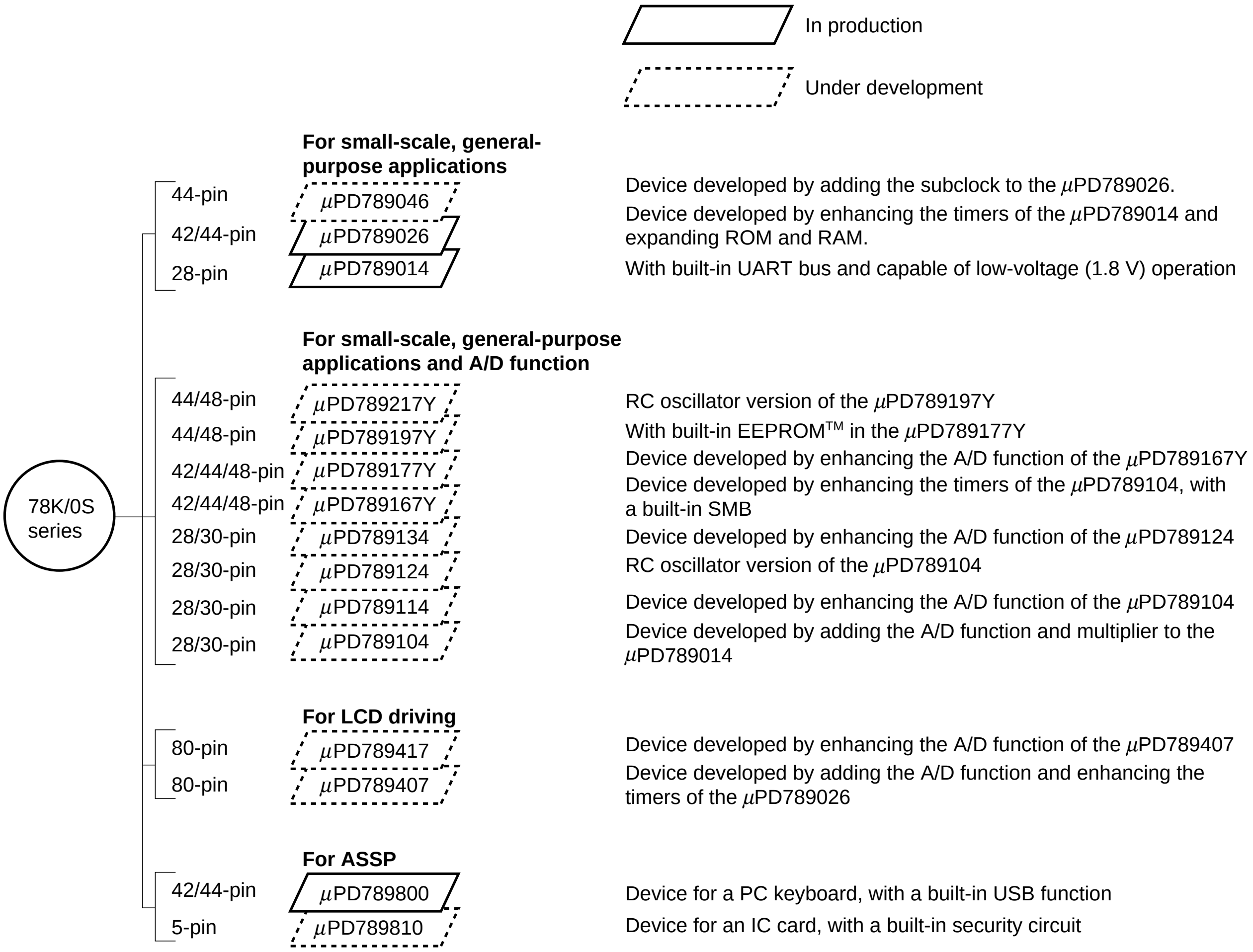
ORDERING INFORMATION

Part number	Package
μPD789046GB-××× -8ES	44-pin plastic QFP (10 × 10 mm)

Remark ××× indicates ROM code suffix.

78K/0S SERIES DEVELOPMENT

The 78K/0S series products are shown below. The sub-series names are indicated in frames.



In production

Under development

78K/0S series

For small-scale, general-purpose applications

44-pin	μPD789046	Device developed by adding the subclock to the μPD789026.
42/44-pin	μPD789026	Device developed by enhancing the timers of the μPD789014 and expanding ROM and RAM.
28-pin	μPD789014	With built-in UART bus and capable of low-voltage (1.8 V) operation

For small-scale, general-purpose applications and A/D function

44/48-pin	μPD789217Y	RC oscillator version of the μPD789197Y
44/48-pin	μPD789197Y	With built-in EEPROM™ in the μPD789177Y
42/44/48-pin	μPD789177Y	Device developed by enhancing the A/D function of the μPD789167Y
42/44/48-pin	μPD789167Y	Device developed by enhancing the timers of the μPD789104, with a built-in SMB
28/30-pin	μPD789134	Device developed by enhancing the A/D function of the μPD789124
28/30-pin	μPD789124	RC oscillator version of the μPD789104
28/30-pin	μPD789114	Device developed by enhancing the A/D function of the μPD789104
28/30-pin	μPD789104	Device developed by adding the A/D function and multiplier to the μPD789014

For LCD driving

80-pin	μPD789417	Device developed by enhancing the A/D function of the μPD789407
80-pin	μPD789407	Device developed by adding the A/D function and enhancing the timers of the μPD789026

For ASSP

42/44-pin	μPD789800	Device for a PC keyboard, with a built-in USB function
5-pin	μPD789810	Device for an IC card, with a built-in security circuit

The following table lists the major differences in functions between the sub-series.

Function Sub-series		ROM size	Timer				8-bit A/D	10-bit A/D	Serial interface	I/O	Minimum V_{DD} value	Remarks
			8-bit	16-bit	Clock	WDT						
Small-scale, general-purpose applications	μPD789046	16 K	1 ch	1 ch	1 ch	1 ch	–	–	1 ch (UART: 1 ch)	34 pins	1.8 V	–
	μPD789026	4 K-16 K			–							
	μPD789014	2 K-4 K	2 ch	–						22 pins		
Small-scale, general-purpose applications and A/D function	μPD789217Y	16 K-24 K	3 ch	1 ch	1 ch	1 ch	–	8 ch	2 ch UART: 1 ch SMB : 1 ch	31 pins	1.8 V	RC-oscillator version, with built-in EEPROM
	μPD789197Y											With built-in EEPROM
	μPD789177Y											–
	μPD789167Y						8 ch	–				
	μPD789134	2 K-8 K	1 ch		–		–	4 ch	1 ch (UART: 1 ch)	20 pins		RC-oscillator version
	μPD789124						4 ch	–				
	μPD789114						–	4 ch				–
	μPD789104						4 ch	–				
LCD driving	μPD789417	12 K-24 K	3 ch	1 ch	1 ch	1 ch	–	7 ch	1 ch (UART: 1 ch)	43 pins	1.8 V	–
	μPD789407						7 ch	–				
ASSP	μPD789800	8 K	2 ch	–	–	1 ch	–	–	2 ch (USB: 1 ch)	31 pins	4.0 V	–
	μPD789810	6 K	–						–	1 pin	1.8 V	With built-in EEPROM

FUNCTIONS

Item		Function
Internal memory	ROM	16 Kbytes
	High-speed RAM	512 bytes
Minimum instruction execution time		• 0.4/1.6 μs (operation with main system clock running at 5.0 MHz) • 122 μs (operation with subsystem clock running at 32.768 kHz)
General-purpose registers		8 bits × 8 registers
Instruction set		• 16-bit operations • Bit manipulations (such as set, reset, and test)
I/O ports		34 CMOS input/output pins
Serial interface		Switchable between three-wire serial I/O and UART modes
Timers		• 16-bit timer counter • 8-bit timer/event counter • Clock timer • Watchdog timer
Timer output		Two outputs
Vectored interrupt sources	Maskable	Seven internal and four external Interrupts
	Nonmaskable	Internal Interrupt
Power supply voltage		V_{DD} = 1.8 to 5.5 V
Operating ambient temperature		T_A = –40°C to +85°C
Package		44-pin plastic QFP (10 × 10 mm)

CONTENTS

1. PIN CONFIGURATION (TOP VIEW) 8

2. BLOCK DIAGRAM 9

3. PIN FUNCTIONS 10

3.1 Port Pins 10
3.2 Non-Port Pins 11
3.3 Pin Input/Output Circuits and Handling of Unused Pins 12

4. CPU ARCHITECTURE 14

4.1 Memory Space 14
4.2 Data Memory Addressing 15
4.3 Processor Registers 16

5. PERIPHERAL HARDWARE FUNCTIONS 20

5.1 Ports 20
5.2 Clock Generator 26
5.3 16-Bit Timer Counter 30
5.4 8-Bit Timer/Event Counter 37
5.5 Clock Timer 41
5.6 Watchdog Timer 44
5.7 Serial Interface 47

6. INTERRUPT FUNCTIONS 58

6.1 Interrupt Function Types 58
6.2 Interrupt Sources and Configuration 58
6.3 Interrupt Function Control Registers 61

7. STANDBY FUNCTION 67

7.1 Standby Function 67
7.2 Standby Function Control Register 70

8. RESET FUNCTIONS 71

9. INSTRUCTION SET OVERVIEW 73

9.1 Legend 73
9.2 Operations 75

10. ELECTRICAL CHARACTERISTICS 80

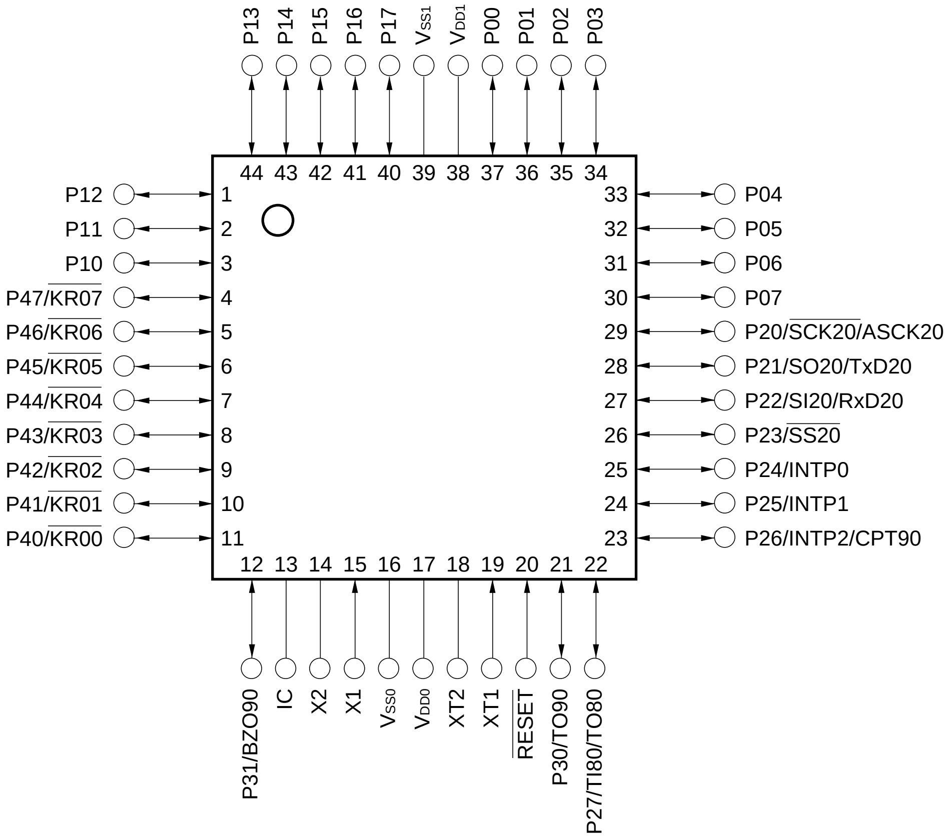
11. PACKAGE DRAWINGS 91

APPENDIX A DEVELOPMENT TOOLS 92

APPENDIX B RELATED DOCUMENTS 94

1. PIN CONFIGURATION (TOP VIEW)

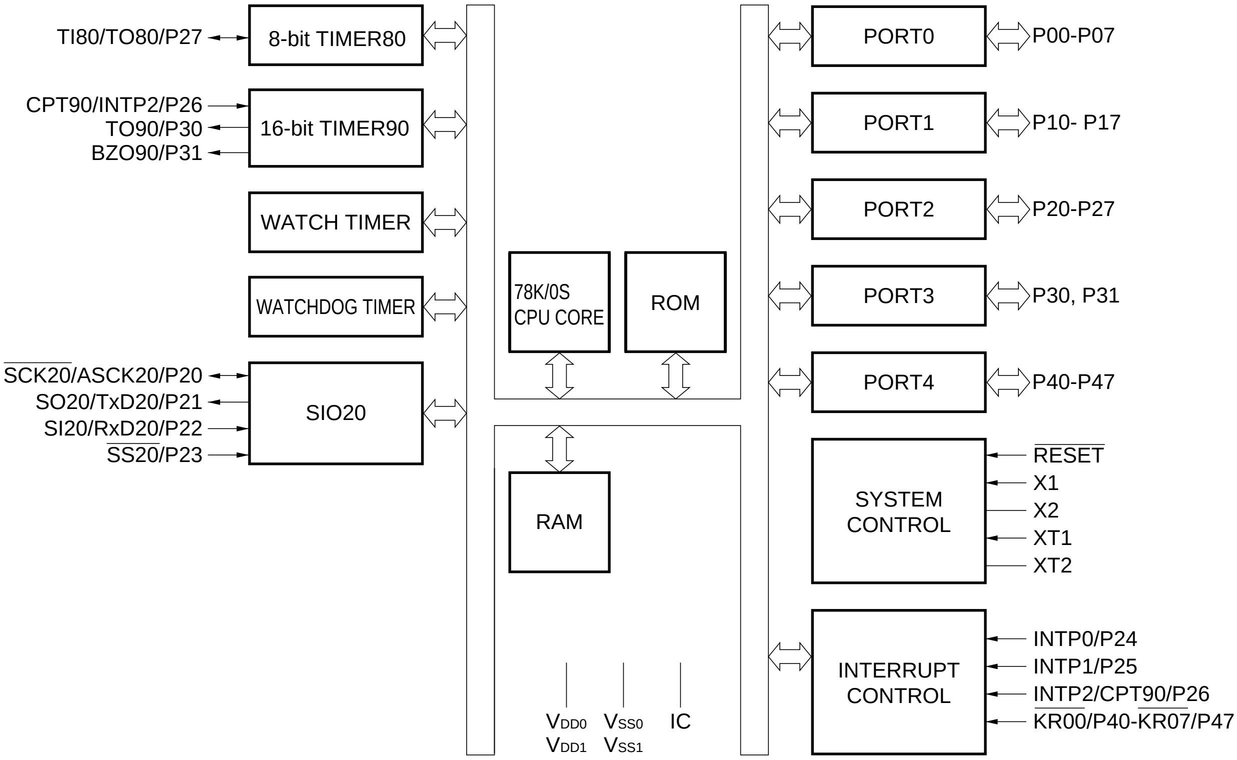
- **44-pin plastic QFP (10 × 10 mm)**
  μPD789046GB-×××-8ES



Caution Connect the IC (internally connected) pin directly to the V_{SS0} or V_{SS1} pin.

ASCK20	: Asynchronous Serial Input	RxD20	: Receive Data
BZO90	: Buzzer Output	SCK20	: Serial Clock
CPT90	: Capture Trigger Input	SI20	: Serial Input
IC	: Internally Connected	SO20	: Serial Output
INTP0-INTP2	: Interrupt from Peripherals	SS20	: Chip Select Input
KR00-KR07	: Key Return	TI80	: Timer Input
P00-P07	: Port 0	TO80, TO90	: Timer Output
P10-P17	: Port 1	TxD20	: Transmit Data
P20-P27	: Port 2	V_{DD0}, V_{DD1}	: Power Supply
P30, P31	: Port 3	V_{SS0}, V_{SS1}	: Ground
P40-P47	: Port 4	X1, X2	: Crystal (Main System Clock)
RESET	: Reset	XT1, XT2	: Crystal (Subsystem Clock)

2. BLOCK DIAGRAM



3. PIN FUNCTIONS

3.1 Port Pins

Pin name	I/O	Function	When reset	Also used as
P00-P07	I/O	Port 0 8-bit input/output port Can be set to either input or output in 1-bit units When used as an input port, whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register 0 (PU0).	Input	–
P10-P17	I/O	Port 1 8-bit input/output port Can be set to either input or output in 1-bit units When used as an input port, whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register 0 (PU0).	Input	–
P20	I/O	Port 2 8-bit input/output port Can be set to either input or output in 1-bit units Whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register B2 (PUB2).	Input	$\overline{\text{SCK20}}$/ASCK20
P21				SO20/TxD20
P22				SI20/RxD20
P23				$\overline{\text{SS20}}$
P24				INTP0
P25				INTP1
P26				INTP2/CPT90
P27				TI80/TO80
P30	I/O	Port 3 2-bit input/output port Can be set to either input or output in 1-bit units When used as an input port, whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register 0 (PU0).	Input	TO90
P31				BZO90
P40-P47	I/O	Port 4 8-bit input/output port Can be set to either input or output in 1-bit units When used as an input port, whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register 0 (PU0).	Input	$\overline{\text{KR00}}$-$\overline{\text{KR07}}$

3.2 Non-Port Pins

Pin name	I/O	Function	When reset	Also used as
INTP0	Input	External interrupt input for which effective edges (rising and/or falling edges) can be set	Input	P24
INTP1				P25
INTP2				P26/CPT90
$\overline{\text{KR00}}$-$\overline{\text{KR07}}$	Input	Detection of key return signal	Input	P40-P47
SI20	Input	Serial data input to serial interface	Input	P22/RxD20
SO20	Output	Serial data output from serial interface	Input	P21/TxD20
$\overline{\text{SCK20}}$	I/O	Serial clock input/output for serial interface	Input	P20/ASCK20
$\overline{\text{SS20}}$	Input	Chip select input to serial interface	Input	P23
ASCK20	Input	Serial clock input to asynchronous serial interface	Input	P20/$\overline{\text{SCK20}}$
RxD20	Input	Serial data input to asynchronous serial interface	Input	P22/SI20
TxD20	Output	Serial data output from asynchronous serial interface	Input	P21/SO20
TI80	Input	External count clock input to 8-bit timer (TM80)	Input	P27/TO80
TO80	Output	8-bit timer (TM80) output	Input	P27/TI80
TO90	Output	16-bit timer (TM90) output	Input	P30
BZO90	Output	16-bit timer (TM90) buzzer output	Input	P31
CPT90	Input	Capture edge input	Input	P26/INTP2
X1	Input	Connected to crystal for main system clock oscillation	–	–
X2	–		–	–
XT1	Input	Connected to crystal for subsystem clock oscillation	–	–
XT2	–		–	–
$\overline{\text{RESET}}$	Input	System reset input	Input	–
V_{DD0}	–	Positive supply voltage for ports	–	–
V_{DD1}	–	Positive supply voltage (for circuits other than ports)	–	–
V_{SS0}	–	Ground potential for ports	–	–
V_{SS1}	–	Ground potential (for circuits other than ports)	–	–
IC	–	This pin is internally connected. Connect this pin directly to the V_{SS0} or V_{SS1} pin.	–	–

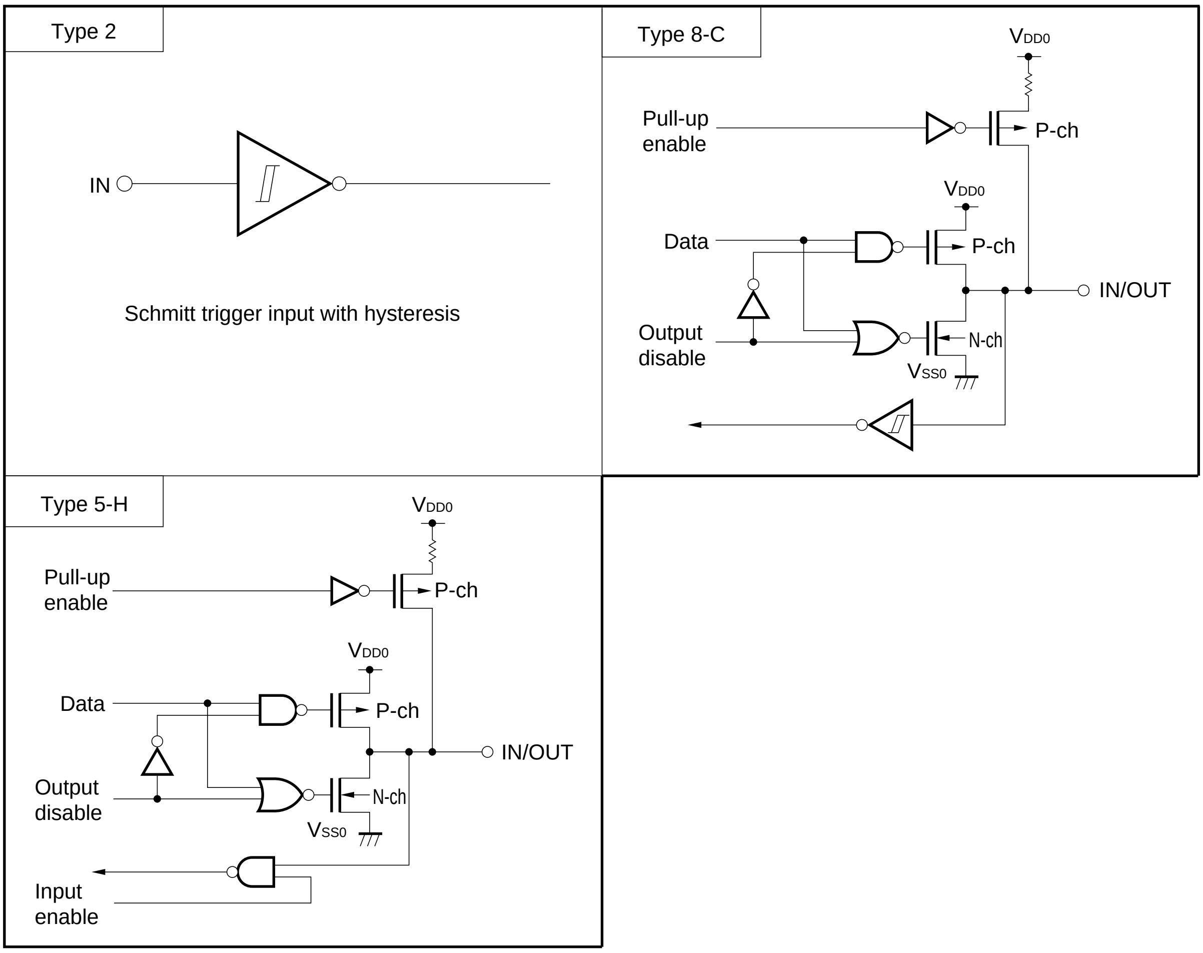
3.3 Pin Input/Output Circuits and Handling of Unused Pins

Table 3-1 lists the types of input/output circuits for each pin and explains how unused pins are handled. Figure 3-1 shows the configuration of each type of input/output circuit.

Table 3-1. Type of Input/Output Circuit for Each Pin and Handling of Unused Pins

Pin name	I/O circuit type	I/O	Recommended connection of unused pins
P00-P07	5-H	I/O	Connect these pins to the V_{DD0}, V_{DD1}, V_{SS0}, or V_{SS1} pin via respective resistors.
P10-P17			
P20/$\overline{\text{SCK20}}$/ASCK20	8-C		
P21/SO20/TxD20			
P22/SI20/RxD20			
P23/$\overline{\text{SS20}}$			
P24/INTP0			
P25/INTP1			
P26/INTP2/CPT90			
P27/TI80/TO80			
P30/TO90	5-H		
P31/BZO90			
P40/$\overline{\text{KR00}}$-P47/$\overline{\text{KR07}}$	8-C		
XT1	–	Input	Connect this pin to the V_{SS0} or V_{SS1} pin via a resistor.
XT2		–	Leave this pin open.
$\overline{\text{RESET}}$	2	Input	–
IC	–	–	Connect this pin directly to the V_{SS0} or V_{SS1} pin.

Figure 3-1. Pin Input/Output Circuits

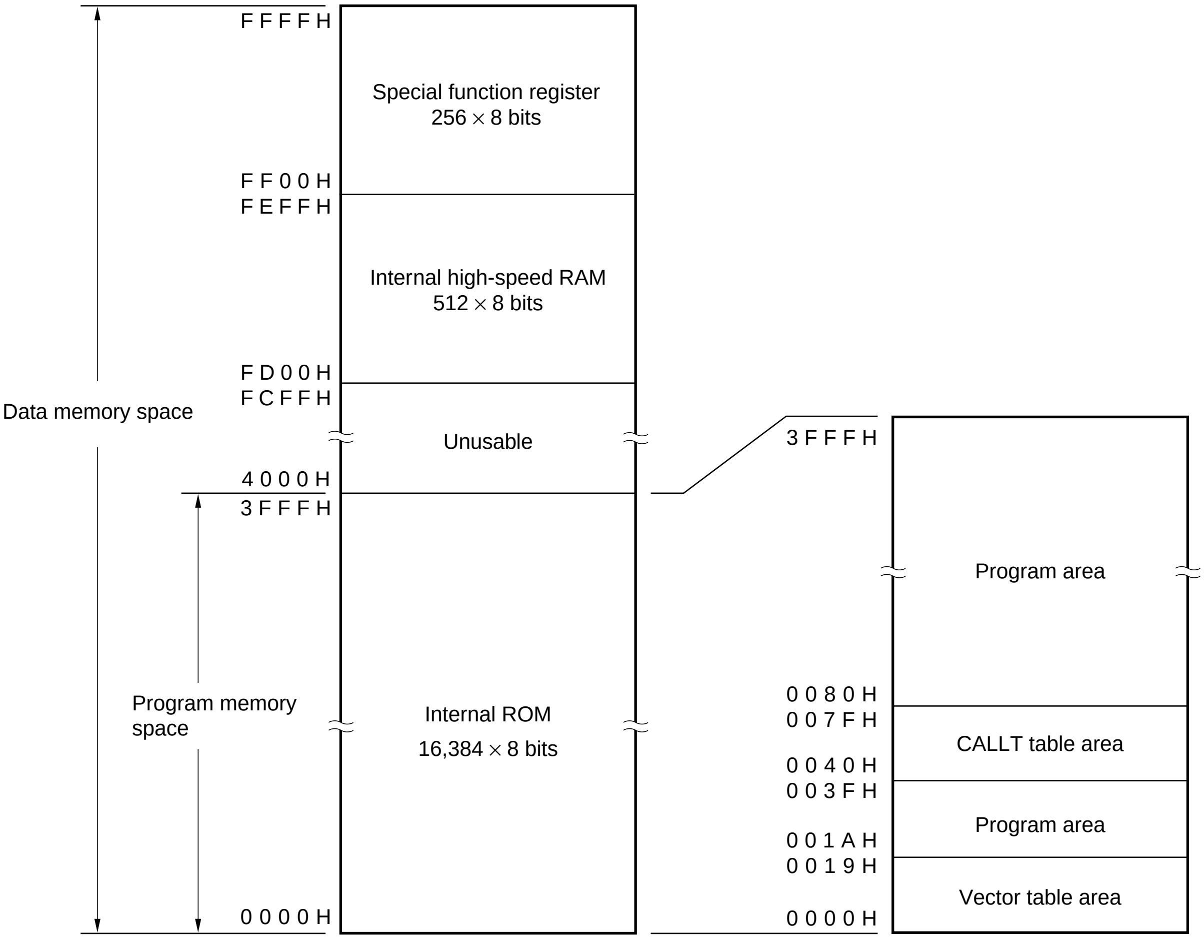


4. CPU ARCHITECTURE

4.1 Memory Space

The μPD789046 can access up to 64 Kbytes of memory space. Figure 4-1 shows the memory map.

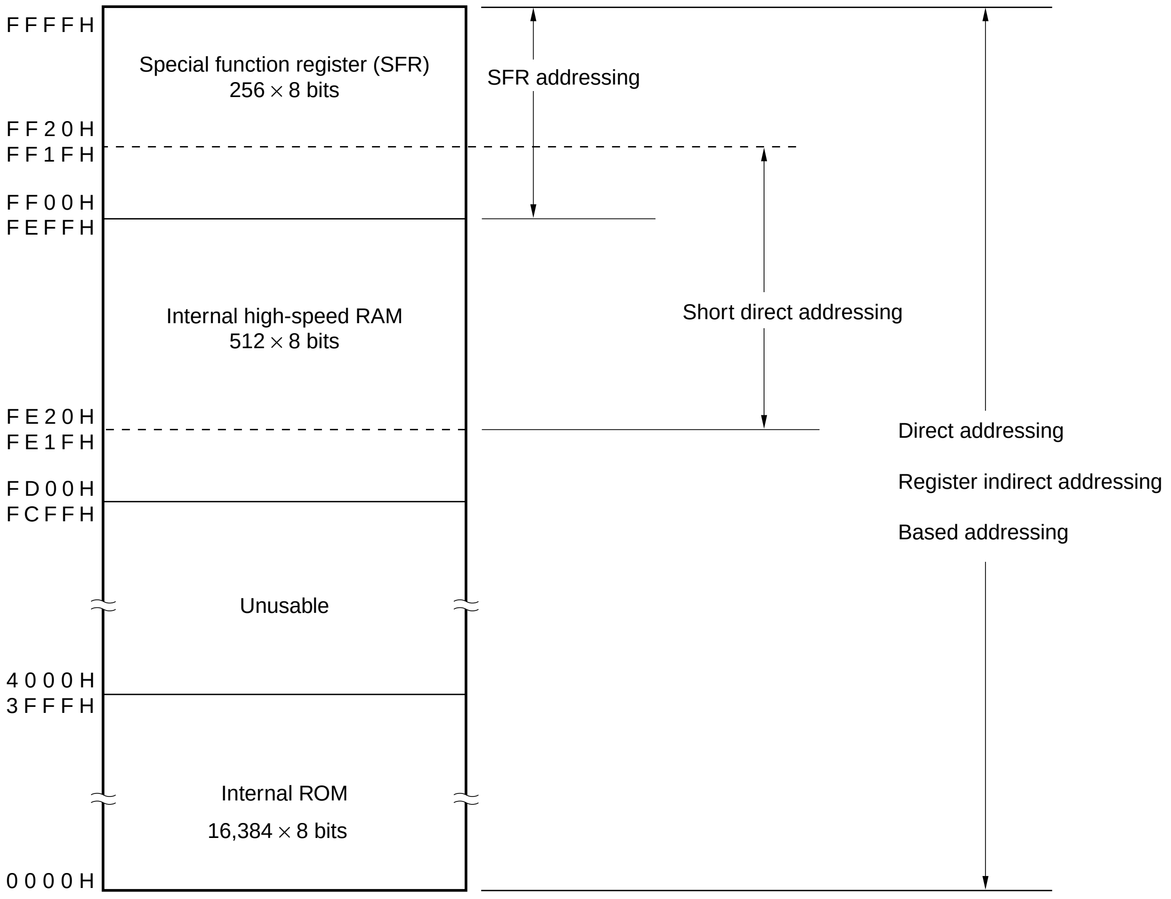
Figure 4-1. Memory Map



4.2 Data Memory Addressing

The μPD789046 is provided with a wide range of addressing modes to make memory manipulation as efficient as possible. A data memory area (FD00H to FFFFH) can be accessed using a unique addressing mode according to its use, such as a special function register (SFR). Figure 4-2 illustrates the data memory addressing modes.

Figure 4-2. Data Memory Addressing Modes



4.3 Processor Registers

4.3.1 Controller registers

(1) Program counter (PC)

The PC is a 16-bit register for holding address information that indicates the next program to be executed.

Figure 4-3. Program Counter Configuration

	15															0
PC	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

(2) Program status word (PSW)

The PSW is an 8-bit register for holding the status of the CPU according to the results of instruction execution.

Figure 4-4. Program Status Word Configuration

	7							0
PSW	IE	Z	0	AC	0	0	1	CY

(a) Interrupt enable flag (IE)

IE is used to control whether interrupt requests are to be accepted by the CPU.

(b) Zero flag (Z)

Z is set (1) if the result of operation is zero. Otherwise, it is reset (0).

(c) Auxiliary carry flag (AC)

AC is set (1) if the result of the operation has a carry from bit 3 or a borrow to bit 3. Otherwise, it is reset (0).

(d) Carry flag (CY)

CY is used to indicate whether an overflow or underflow has occurred during the execution of a subtract or add instruction.

(3) Stack pointer (SP)

SP is a 16-bit register for holding the start address of a stack area. The stack area can be specified only in an area (FD00H to FEFFH) of internal high-speed RAM.

Figure 4-5. Stack Pointer Configuration

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

Caution A $\overline{\text{RESET}}$ input makes the SP content undefined. Before executing an instruction, always initialize the SP.

4.3.2 General-purpose registers

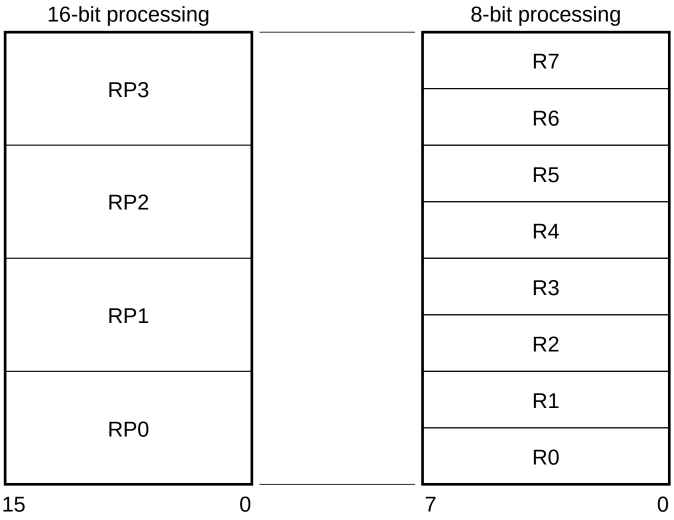
The μPD789046 has eight 8-bit general-purpose registers (X, A, C, B, E, D, L, and H).

These registers can be used as 16-bit registers (two 8-bit registers used in pairs like AX, BC, DE, and HL) as well as ordinary 8-bit registers.

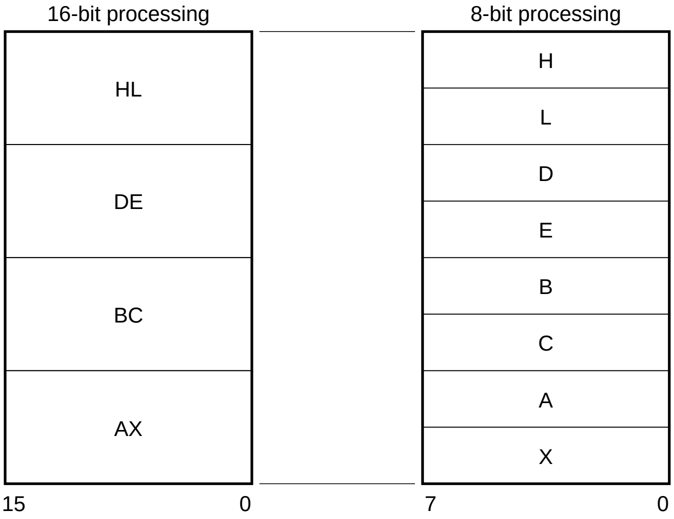
These registers are identified using functional register names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute register names (R0 to R7 and RP0 to RP3).

Figure 4-6. General-Purpose Register Configuration

(a) Absolute register names



(b) Functional register names



4.3.3 Special function registers (SFRs)

The SFRs are used as peripheral hardware mode registers and control registers. They are mapped in a 256-byte space, from FF00H to FFFFH.

If a bit name is specified as a reserved word in the RA78K/0S series, and is defined in header file sfrbit.h in the CC78K/0S series, the corresponding bit number in the format of the register is circled. See the format of each register, explained in **Chapter 5**.

Table 4-1. Special Function Registers (1/2)

Address	Special function register (SFR) name	Symbol	R/W	Number of bits manipulated simultaneously			When reset
				1 bit	8 bits	16 bits	
FF00H	Port 0	P0	R/W	O	O	–	00H
FF01H	Port 1	P1		O	O	–	
FF02H	Port 2	P2		O	O	–	
FF03H	Port 3	P3		O	O	–	
FF04H	Port 4	P4		O	O	–	
FF16H	16-bit compare register 90	CR90	W	–	O[Note 1]	O[Note 2]	FFFFH
FF17H							
FF18H	16-bit timer register 90	TM90	R	–	O[Note 1]	O[Note 2]	0000H
FF19H							
FF1AH	16-bit capture register 90	TCP90		–	O[Note 1]	O[Note 2]	Undefined
FF1BH							
FF20H	Port mode register 0	PM0	R/W	O	O	–	FFH
FF21H	Port mode register 1	PM1		O	O	–	
FF22H	Port mode register 2	PM2		O	O	–	
FF23H	Port mode register 3	PM3		O	O	–	
FF24H	Port mode register 4	PM4		O	O	–	
FF32H	Pull-up resistor option register B2	PUB2		O	O	–	00H
FF42H	Timer clock selection register 2	TCL2		–	O	–	
FF48H	16-bit timer mode control register 90	TMC90		O	O	–	
FF49H	Buzzer output control register 90	BZC90		O	O	–	
FF4AH	Clock timer mode control register	WTM		O	O	–	
FF50H	8-bit compare register 80	CR80	W	–	O	–	Undefined
FF51H	8-bit timer register 80	TM80	R	–	O	–	00H
FF53H	8-bit timer mode control register 80	TMC80	R/W	O	O	–	
FF70H	Asynchronous serial interface mode register 20	ASIM20		O	O	–	
FF71H	Asynchronous serial interface status register 20	ASIS20	R	O	O	–	
FF72H	Serial operation mode register 20	CSIM20	R/W	O	O	–	
FF73H	Baud rate generator control register 20	BRGC20		–	O	–	

Notes 1. CR90, TM90, and TCP90 are designed only for 16-bit access. In direct addressing, however, 8-bit access can also be performed.

2. 16-bit access is allowed only in short direct addressing.

Table 4-1. Special Function Registers (2/2)

Address	Special function register (SFR) name	Symbol		R/W	Number of bits manipulated simultaneously			When reset
					1 bit	8 bits	16 bits	
FF74H	Transmission shift register 20	TXS20	SIO20	W	–	○	–	FFH
	Reception buffer register 20	RXB20		R	–	○	–	Undefined
FFE0H	Interrupt request flag register 0	IF0		R/W	○	○	–	00H
FFE1H	Interrupt request flag register 1	IF1			○	○	–	
FFE4H	Interrupt mask flag register 0	MK0			○	○	–	FFH
FFE5H	Interrupt mask flag register 1	MK1			○	○	–	
FFECH	External interrupt mode register 0	INTM0			–	○	–	00H
FFF0H	Suboscillation mode register	SCKM			○	○	–	
FFF2H	Subclock control register	CSS			○	○	–	
FFF5H	Key return mode register 00	KRM00			○	○	–	
FFF7H	Pull-up resistor option register 0	PU0			○	○	–	
FFF9H	Watchdog timer mode register	WDTM			○	○	–	
FFFAH	Oscillation settling time selection register	OSTS			–	○	–	04H
FFFBH	Processor clock control register	PCC			○	○	–	02H

5. PERIPHERAL HARDWARE FUNCTIONS

5.1 Ports

5.1.1 Port functions

The μPD789046 is provided with the ports shown in Figure 5-1. These ports are used to enable several types of control. Table 5-1 lists the functions of each port.

These ports, while originally designed as digital input/output ports, can also be used for other functions, as summarized in **Chapter 3**.

Figure 5-1. Port Types

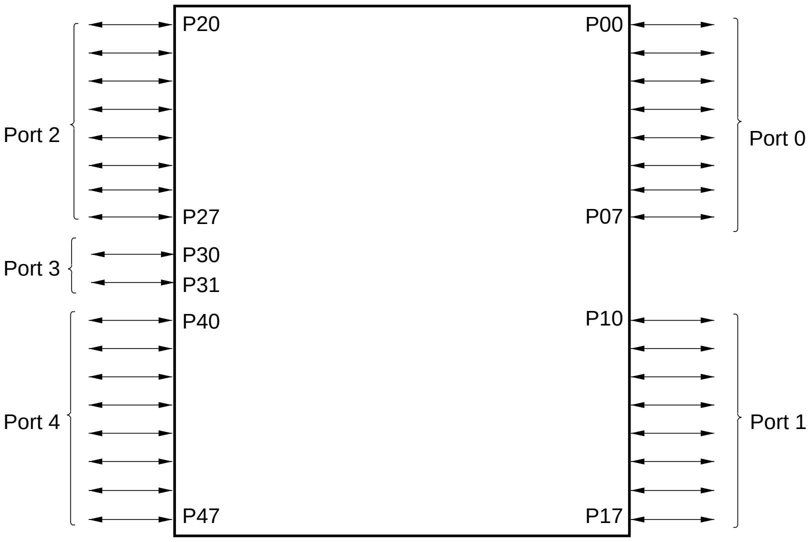


Table 5-1. Port Functions

Port name	Pin name	Description
Port 0	P00-P07	Input/output port. Can be set to either input or output in 1-bit units. When used as an input port, whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register 0 (PU0).
Port 1	P10-P17	Input/output port. Can be set to either input or output in 1-bit units. When used as an input port, whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register 0 (PU0).
Port 2	P20-P27	Input/output port. Can be set to either input or output in 1-bit units. Whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register B2 (PUB2).
Port 3	P30, P31	Input/output port. Can be set to either input or output in 1-bit units. When used as an input port, whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register 0 (PU0).
Port 4	P40-P47	N-channel open-drain input/output port. Can be set to either input or output in 1-bit units. When used as an input port, whether the on-chip pull-up resistor is to be used can be specified by pull-up resistor option register 0 (PU0).

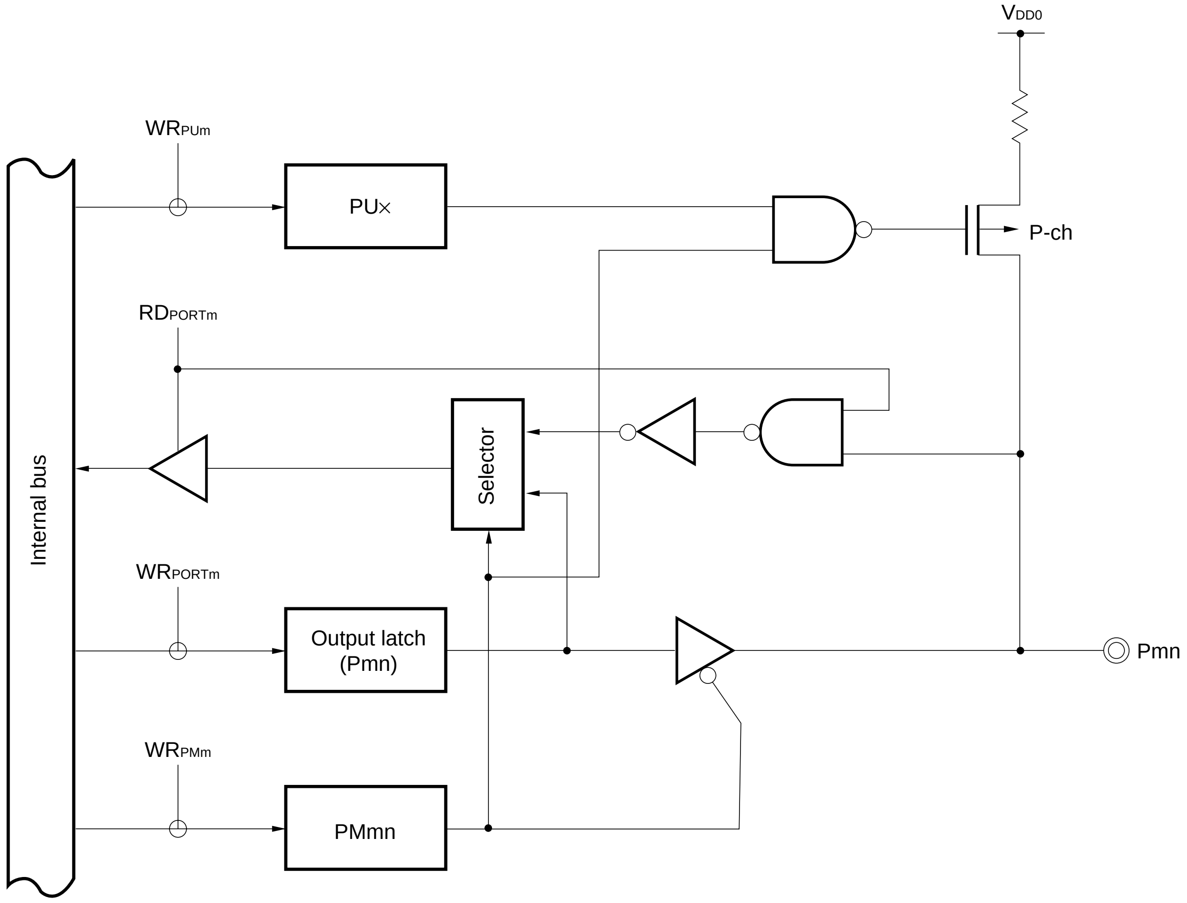
5.1.2 Port configuration

The hardware configuration of the ports is as follows.

Table 5-2. Port Configuration

Item	Configuration
Control register	Port mode register (PMm, where m = 0 to 4) Pull-up resistor option registers (PU0, PUB2)
Port pins	Total: 34 (34 CMOS input/output pins)
Pull-up resistors	Total: 34 (on-chip pull-up resistors can be used as specified by software)

Figure 5-2. Basic CMOS Port Configuration



Caution **Figure 5-2 shows the basic configuration of the CMOS input/output ports. The configuration differs depending on the pull-up option resistor and the functions assigned to the dual-function pins.**

Remark PU× : Pull-up resistor option register (× = 0, B2)
PMmn : Bit n of port mode register m, where m = 0 to 4 and n = 0 to 7
Pmn : Bit n of port m
RD : Port read signal
WR : Port write signal
For details, see **(2)** in **Section 5.1.3**.

5.1.3 Port function control registers

The following two types of registers are used to control the ports.

- Port mode registers (PM0 to PM4)
- Pull-up resistor option registers (PU0, PUB2)

(1) Port mode registers (PM0 to PM4)

The port mode registers separately set each port bit to either input or output.

Each port mode register is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input writes FFH into the port mode registers.

When port pins are used for secondary functions, the corresponding port mode register and output latch must be set or reset as described in Table 5-3.

Caution When port 2 is acting as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as the input for an external interrupt. To use port 3 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

Table 5-3. Port Mode Register and Output Latch Settings for Using Secondary Functions

Pin name	Secondary function		PM××	P××
	Name	Input/output		
P24	INTP0	Input	1	×
P25	INTP1	Input	1	×
P26	INTP2	Input	1	×
	CPT90	Input	1	×
P27	TI80	Input	1	×
	TO80	Output	0	0
P30	TO90	Output	0	0
P31	BZO90	Output	0	0
P40-P47[Note]	$\overline{\text{KR00}}$-$\overline{\text{KR07}}$	Input	1	×

Note To use the P40 to P47 pins for the secondary function, set the corresponding bits of key return mode register 00 (KRM00) to 1. (See **(5)** in **Section 6.3**.)

Caution When port 2 is being used as a serial interface, it is necessary to specify whether the port is an input or output port, and to set the output latch accordingly. See Table 5-11 for an explanation of how to make this specification.

Remark		
×	:	Don’t care
PM××	:	Port mode register
P××	:	Port output latch

Figure 5-3. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W

PMmn	Pmn pin input/output mode selection (m = 0 to 2, 4 n = 0 to 7 m = 3 n = 0, 1)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

(2) Pull-up resistor option registers (PU0, PUB2)

These registers are used to specify pull-up resistor connection on a port-by-port basis and bit-by-bit basis.

The method of pull-up resistor connection varies, depending on whether a connection is made on a port-by-port basis or bit-by-bit basis as described below.

(a) Pull-up resistor option register 0 (PU0)

This register is used to specify whether the on-chip pull-up resistors for ports 0 to 2 and 4 are to be used for each port. An on-chip pull-up resistor can be used only for those bits set to the input mode of a port for which the use of the on-chip pull-up resistor is specified using PU0. For those bits set to the output mode, on-chip pull-up resistors cannot be used, regardless of the setting of PU0. This also applies to a dual-function pin used as an output pin.

A $\overline{\text{RESET}}$ input clears PU0 to 00H.

(b) Pull-up resistor option register B2 (PUB2)

These registers are used to specify whether the on-chip pull-up resistor for each pin of port 2 is to be used in 1-bit units. By setting PUB2, an on-chip pull-up resistor can be used, regardless of the setting of the port mode register.

A $\overline{\text{RESET}}$ input clears PUB2 to 00H.

Figure 5-4. Format of Pull-Up Resistor Option Register 0

Symbol	7	6	5	④	③	2	①	⓪	Address	When reset	R/W
PU0	0	0	0	PU04	PU03	0	PU01	PU00	FFF7H	00H	R/W

PU0m	Pm on-chip pull-up resistor selection[Note] (m = 0, 1, 3, or 4)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Note For each port, PU0 selects whether on-chip pull-up resistors are to be used.

Caution Bits 2, and 5 to 7 must be fixed to 0.

Figure 5-5. Format of Pull-Up Resistor Option Register B2

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	When reset	R/W
PUB2	PUB27	PUB26	PUB25	PUB24	PUB23	PUB22	PUB21	PUB20	FF32H	00H	R/W

PUB2m	P2m on-chip pull-up resistor selection[Note] (m = 0 to 7)
0	On-chip pull-up resistor not used
1	On-chip pull-up resistor used

Note PUB2 selects whether on-chip pull-up resistors are to be used in 1-bit units.

5.2 Clock Generator

5.2.1 Clock generator functions

The clock generator generates the clock pulse to be supplied to the CPU and peripheral hardware. There are two types of system clock oscillators:

- Main system clock oscillator
  This circuit generates a frequency of 1.0 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction or by using the processor clock control register.
- Subsystem clock oscillator
  This circuit generates 32.768 kHz. Oscillation can be stopped by using the suboscillation mode register.

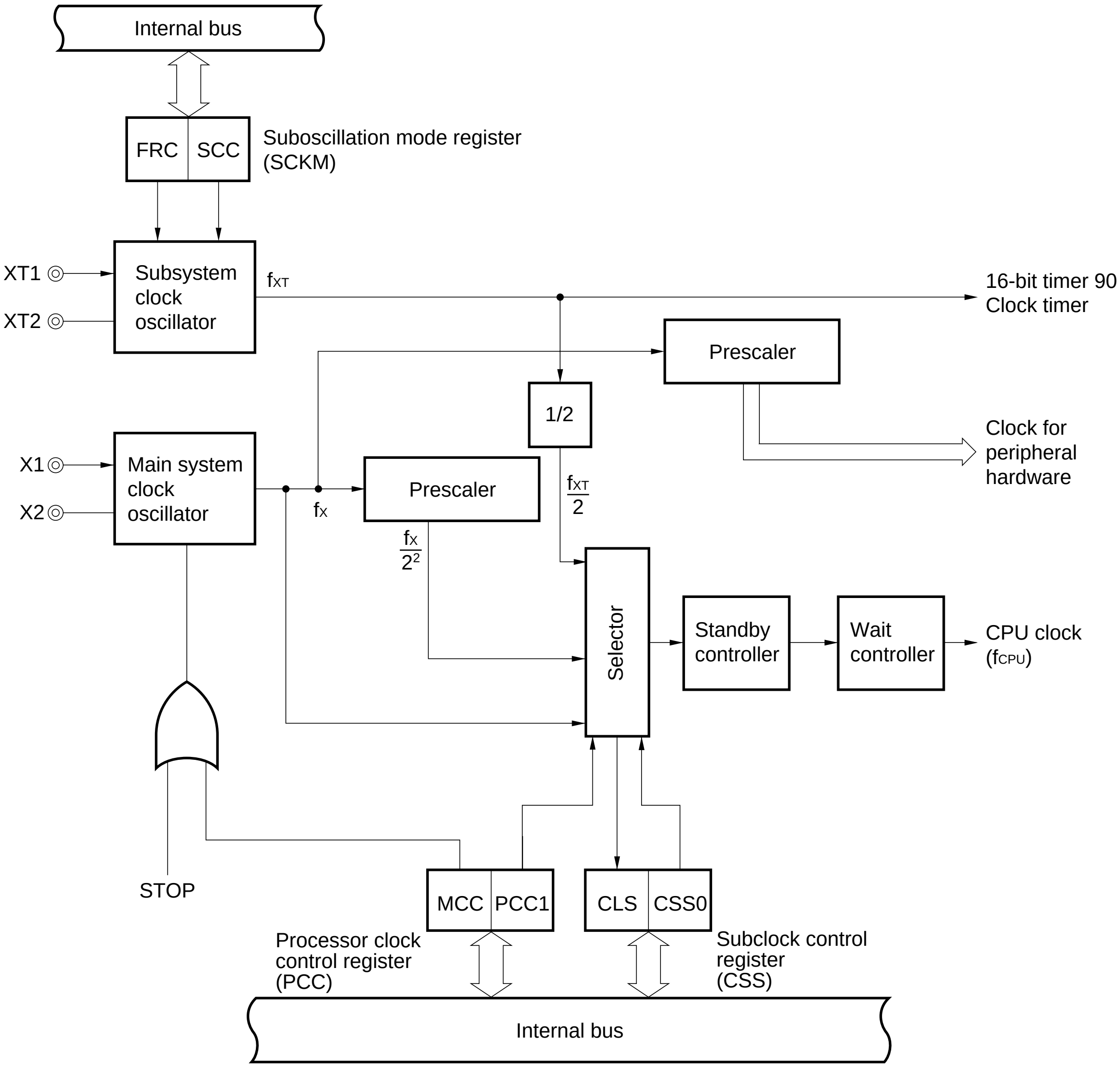
5.2.2 Clock generator configuration

The clock generator consists of the following hardware.

Table 5-4. Clock Generator Configuration

Item	Configuration
Control register	Processor clock control register (PCC) Suboscillation mode register (SCKM) Subclock control register (CSS)
Oscillator	Main system clock oscillator Subsystem clock oscillator

Figure 5-6. Block Diagram of Clock Generator



5.2.3 Clock generator control registers

The clock generator is controlled using the following registers.

- Processor clock control register (PCC)
- Suboscillation mode register (SCKM)
- Subclock control register (CSS)

(1) Processor clock control register (PCC)

The PCC selects a CPU clock and specifies a corresponding frequency division ratio.

It is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input loads 02H into the PCC.

Figure 5-7. Format of Processor Clock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
PCC	MCC	0	0	0	0	0	PCC1	0	FFFBH	02H	R/W

MCC	Control of main system clock oscillator operation
0	Operation enabled
1	Operation disabled

CSS0	PCC1	CPU clock (f_{CPU}) selection[Note]
0	0	f_X (0.2 μs)
0	1	$f_X/2^2$ (0.8 μs)
1	0	$f_{XT}/2$ (61 μs)
1	1	

Note A CPU clock is selected by a combination of the PCC1 flag in the processor clock control register (PCC) and the CSS0 flag in the subclock control register (CSS). (See **(3)** in **Section 5.2.3**.)

Cautions 1. Bit 0 and bits 2 to 6 must be fixed to 0.

2. MCC can be set only when the subsystem clock is selected as the CPU clock.

3. Never set MCC when an external clock is applied. This is because the X2 pin is pulled up to V_{DD0} or V_{DD1}.

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

4. Minimum instruction execution time: 2 f_{CPU}

- f_{CPU} = 0.2 μs : 0.4 μs
- f_{CPU} = 0.8 μs : 1.6 μs
- f_{CPU} = 61 μs : 122 μs

(2) Suboscillation mode register (SCKM)

The SCKM selects a feedback resistor for the subsystem clock, and controls the oscillation of the clock.

It is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears SCKM to 00H.

Figure 5-8. Format of Suboscillation Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
SCKM	0	0	0	0	0	0	FRC	SCC	FFF0H	00H	R/W

FRC	Feedback resistor selection
0	Internal feedback resistor used
1	Internal feedback resistor not used

SCC	Control of subsystem clock oscillator operation
0	Operation enabled
1	Operation disabled

Caution Bits 2 to 7 must be fixed to 0.

(3) Subclock control register (CSS)

The CSS specifies whether the main system or subsystem clock oscillator is to be selected. It also specifies how the CPU clock operates.

It is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears CSS to 00H.

Figure 5-9. Format of Subclock Control Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
CSS	0	0	CLS	CSS0	0	0	0	0	FFF2H	00H	R/W[Note]

CLS	CPU clock operation status
0	Operation based on the output of the divided main system clock
1	Operation based on the subsystem clock

CSS0	Selection of the main system or subsystem clock oscillator
0	Divided output from the main system clock oscillator
1	Output form the subsystem clock oscillator

Note Bit 5 is read-only.

Caution Bits 0 to 3, 6, and 7 must be fixed to 0.

5.3 16-Bit Timer Counter

5.3.1 16-bit timer counter functions

16-bit timer counter 90 (TM90) has the following functions.

(1) Timer interrupt

An interrupt is generated if the TM90 count matches a comparison value.

(2) Timer output

The timer output can be controlled if the count matches a comparison value.

(3) Count capture

The count in TM90 is captured into the capture register in synchronization with a capture trigger.

(4) Buzzer output

The buzzer output can be controlled if the count matches the comparison value.

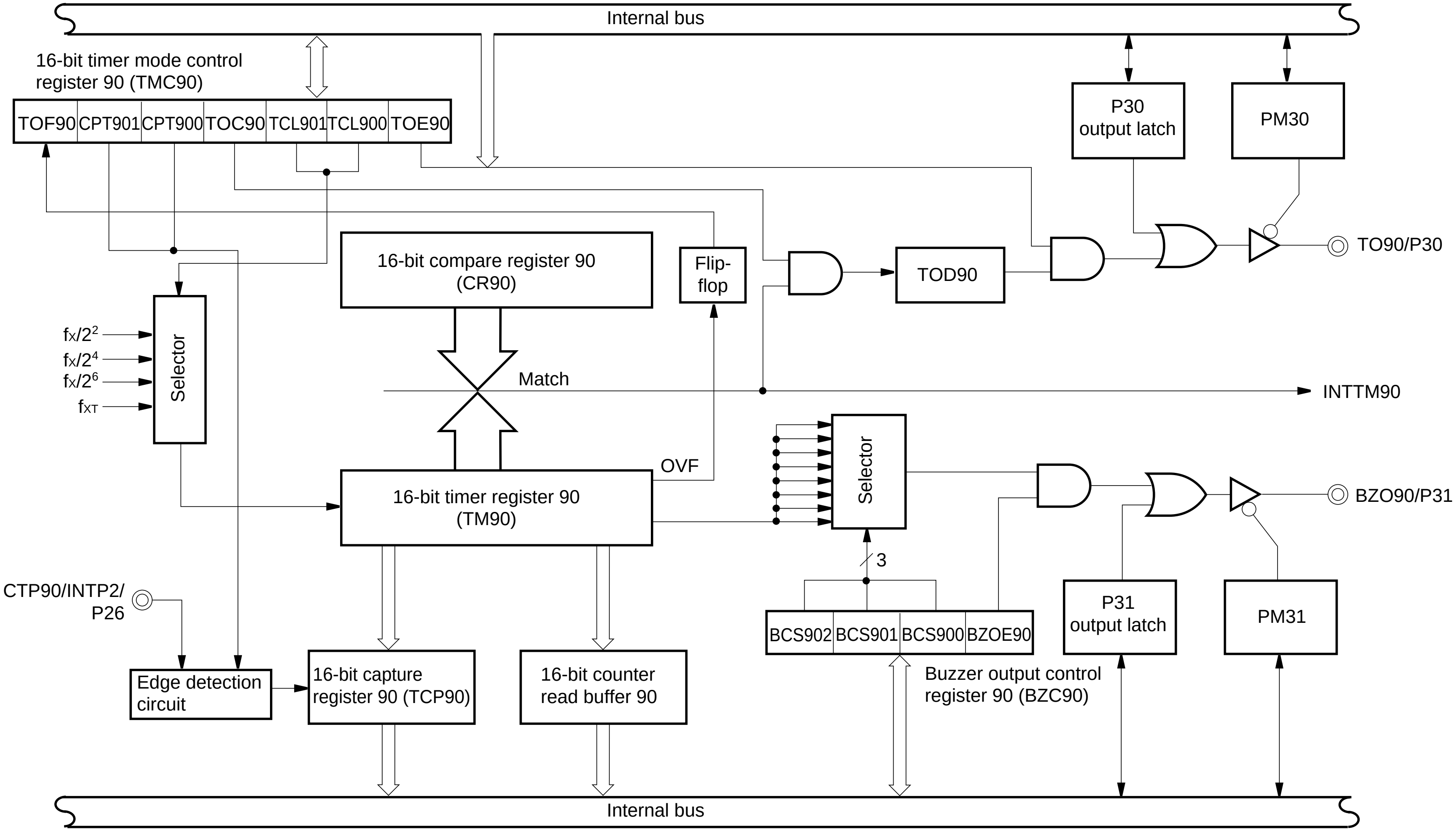
5.3.2 16-bit timer counter configuration

16-bit timer counter 90 (TM90) consists of the following hardware.

Table 5-5. 16-Bit Timer Counter 90 Configuration

Item	Configuration
Timer register	16 bits × 1 (TM90)
Register	Compare register 90 : 16 bits × 1 (CR90) Capture register 90 : 16 bits × 1 (TCP90)
Timer output	1 (TO90)
Control register	16-bit timer mode control register 90 (TMC90) Buzzer output control register 90 (BZC90) Port mode register 3 (PM3)

Figure 5-10. Block Diagram of 16-Bit Timer Counter



(1) 16-bit compare register 90 (CR90)

A value specified in CR90 is compared with the count in 16-bit timer register 90 (TM90). If they match, an interrupt request (INTTM90) is issued.

CR90 is manipulated using an 8-bit or 16-bit memory manipulation instruction. Any value from 0000H to FFFFH can be set.

A $\overline{\text{RESET}}$ input loads FFFFH into CR90.

Cautions 1. CR90 is designed to be manipulated using a 16-bit memory manipulation instruction. It can also be manipulated using 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to set CR90, it must be in a direct addressing access mode. When a 16-bit memory manipulation instruction is used, this register can be accessed only in short direct addressing.

2. To re-set CR90 during count operation, it is necessary to disable interrupts in advance, using an interrupt mask flag register 1 (MK1). It is also necessary to disable inversion of the timer output data, using 16-bit timer mode control register 90 (TMC90).

(2) 16-bit timer register 90 (TM90)

TM90 is used to count the number of pulses.

The contents of TM90 are read using an 8-bit or 16-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TM90 to 0000H.

Cautions 1. The count becomes undefined when STOP mode is deselected, because the count operation is performed before oscillation settles.

2. TM90 is designed to be manipulated using a 16-bit memory manipulation instruction. It can also be manipulated using 8-bit memory manipulation instructions, however. When an 8-bit memory instruction is used to manipulate TM90, it must be in a direct addressing access mode. When a 16-bit memory manipulation instruction is used, this register can be accessed only in short direct addressing.

3. When an 8-bit memory manipulation instruction is used to manipulate TM90, the lower and upper bytes must be read as a pair, in this order.

(3) 16-bit capture register 90 (TCP90)

TCP90 captures the contents of 16-bit timer register 90 (TM90).

It is manipulated using an 8-bit or 16-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input makes TCP90 undefined.

Caution TCP90 is designed to be manipulated using a 16-bit memory manipulation instruction. It can also be manipulated using 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to manipulate TCP90, it must be in a direct addressing access mode. When a 16-bit memory manipulation instruction is used, this register can be accessed only in short direct addressing.

(4) 16-bit counter read buffer 90

This buffer is used to latch and hold the count for 16-bit timer register 90 (TM90).

5.3.3 16-bit timer counter control registers

The following three types of registers are used to control 16-bit timer counter 90 (TM90).

- 16-bit timer mode control register 90 (TMC90)
- Buzzer output control register 90 (BZC90)
- Port mode register 3 (PM3)

(1) 16-bit timer mode control register 90 (TMC90)

TMC90 controls the count clock and capture edge settings.
It is manipulated using a 1-bit or 8-bit memory manipulation instruction.
A $\overline{\text{RESET}}$ input clears TMC90 to 00H.

Figure 5-11. Format of 16-Bit Timer Mode Control Register 90

Symbol	7	⑥	5	4	3	2	1	⓪	Address	When reset	R/W
TMC90	TOD90	TOF90	CPT901	CPT900	TOC90	TCL901	TCL900	TOE90	FF48H	00H	R/W[Note]

TOD90	Timer output data
0	Timer output of 0
1	Timer output of 1

TOF90	Overflow flag control
0	Reset or cleared by software
1	Set when the 16-bit timer overflows

CPT901	CPT900	Capture edge selection
0	0	Capture operation disabled
0	1	Captured at the rising edge at the CPT90 pin
1	0	Captured at the falling edge at the CPT90 pin
1	1	Captured at both the rising and falling edges at the CPT90 pin

TOC90	Timer output data inversion control
0	Inversion disabled
1	Inversion enabled

TCL901	TCL900	16-bit timer counter 90 count clock selection
0	0	$f_X/2^2$ (1.25 MHz)
0	1	$f_X/2^6$ (78.125 kHz)
1	0	$f_X/2^4$ (312.5 kHz)
1	1	f_{XT} (32.768 kHz)

TOE90	16-bit timer counter 90 output control
0	Output disabled (port mode)
1	Output enabled

Note Bit 7 is read-only.

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XT} : Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

(2) Buzzer output control register 90 (BZC90)

Based on the count clock (fcl) selected with the count clock selection bits (TCL901 and TCL900), this register sets a buzzer frequency and controls square wave output.

BZC90 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears BZC90 to 00H.

Figure 5-12. Format of Buzzer Output Control Register 90

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
BZC90	0	0	0	0	BCS902	BCS901	BCS900	BZOE90	FF49H	00H	R/W

BZOE90	Buzzer port output control
0	Disables buzzer port output.
1	Enables buzzer port output.

BCS902	BCS901	BCS900	Buzzer frequency			
			fcl = $f_X/2^2$	fcl = $f_X/2^6$	fcl = $f_X/2^4$	fcl = f_{XT}
0	0	0	fcl/2^4 (78.1 kHz)	fcl/2^4 (4.88 kHz)	fcl/2^4 (19.5 kHz)	fcl/2^4 (2.05 kHz)
0	0	1	fcl/2^5 (39.1 kHz)	fcl/2^5 (2.44 kHz)	fcl/2^5 (9.77 kHz)	fcl/2^5 (1.02 kHz)
0	1	0	fcl/2^8 (4.88 kHz)	fcl/2^8 (305 Hz)	fcl/2^8 (1.22 kHz)	fcl/2^8 (128 Hz)
0	1	1	fcl/2^9 (2.44 kHz)	fcl/2^9 (153 Hz)	fcl/2^9 (610 Hz)	fcl/2^9 (64 Hz)
1	0	0	fcl/2^{10} (1.22 kHz)	fcl/2^{10} (76 Hz)	fcl/2^{10} (305 Hz)	fcl/2^{10} (32 Hz)
1	0	1	fcl/2^{11} (610 Hz)	fcl/2^{11} (38 Hz)	fcl/2^{11} (153 Hz)	fcl/2^{11} (16 Hz)
1	1	0	fcl/2^{12} (305 Hz)	fcl/2^{12} (19 Hz)	fcl/2^{12} (76.3 Hz)	fcl/2^{12} (8 Hz)
1	1	1	fcl/2^{13} (153 Hz)	fcl/2^{13} (10 Hz)	fcl/2^{13} (38.1 Hz)	fcl/2^{13} (4 Hz)

Cautions 1. Bits 4 to 7 must be fixed to 0.

2. If the subclock is selected as the count clock (TCL901 = 1, TCL900 = 1: see Figure 5-11), the subclock is not synchronized when buzzer port output is enabled. In this case, the capture function and TM90 register read function are disabled. In addition, the count value of the TM90 register is undefined.

Remarks 1. f_X : Main system clock oscillation frequency

2. f_{XT}: Subsystem clock oscillation frequency

3. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

(3) Port mode register 3 (PM3)

PM3 separately sets each bit of port 3 to either input or output.

When the P30/TO90 pin is used for timer output, set 0 in the output latch of PM30 and P30. When the P31/BZO90 pin is used for buzzer output, set 0 in the output latch of the PM31 and P31.

PM3 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input loads FFH into PM3.

Figure 5-13. Format of Port Mode Register 3

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FF23H	FFH	R/W

PM3n	P3n pin I/O mode (n = 0 or 1)
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

5.4 8-Bit Timer/Event Counter

5.4.1 8-bit timer/event counter functions

The 8-bit timer/event counter 80 (TM80) has the following functions.

(1) 8-bit interval timer

This timer causes interrupts to be issued at specified intervals.

(2) External event counter

This counter is used to count the number of pulses input from an external source.

(3) Square wave output

A square wave of any frequency can be output.

(4) PWM output

PWM output with an 8-bit resolution is supported.

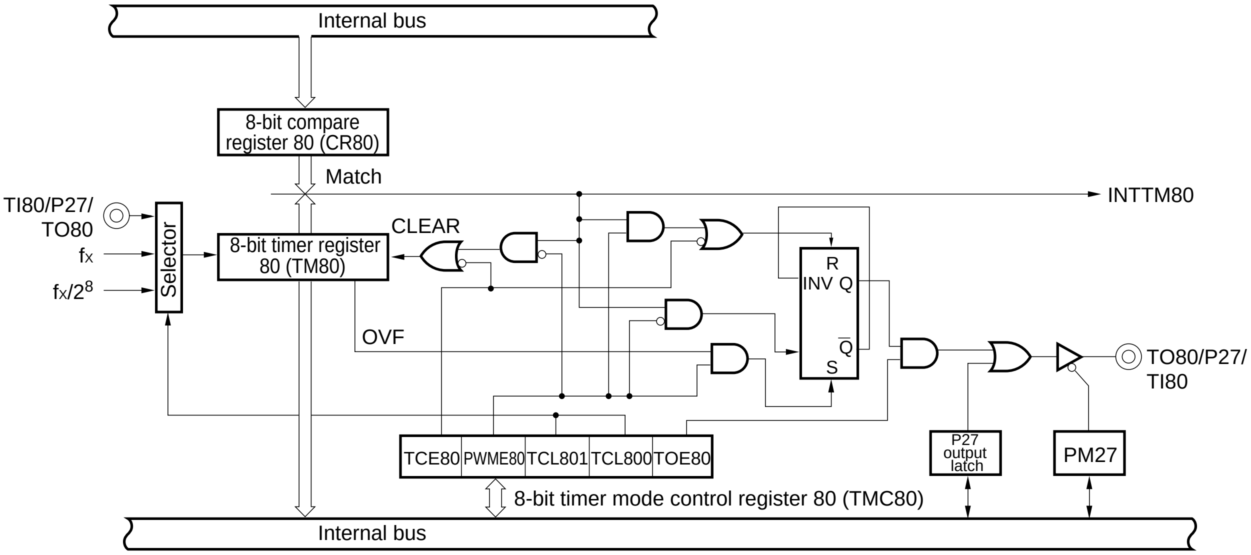
5.4.2 8-bit timer/event counter configuration

The 8-bit timer/event counter 80 (TM80) consists of the following hardware.

Table 5-6. 8-Bit Timer/Event Counter Configuration

Item	Configuration
Timer register	8 bits × 1 (TM80)
Register	Compare register: 8 bits × 1 (CR80)
Timer output	1 (TO80)
Control register	8-bit timer mode control register 80 (TMC80) Port mode register 2 (PM2)

Figure 5-14. Block Diagram of 8-Bit Timer/Event Counter



(1) 8-bit compare register 80 (CR80)

A value specified in CR80 is compared with the count in 8-bit timer register 80 (TM80). If they match, an interrupt request (INTTM80) is issued.

CR80 is manipulated using an 8-bit memory manipulation instruction. Any value from 00H to FFH can be set.

A $\overline{\text{RESET}}$ input makes CR80 undefined.

(2) 8-bit timer register 80 (TM80)

TM80 is used to count the number of pulses.

Its contents are read using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TM80 to 00H.

5.4.3 8-bit timer/event counter control registers

The following two types of registers are used to control the 8-bit timer/event counter 80 (TM80).

- 8-bit timer mode control register 80 (TMC80)
- Port mode register 2 (PM2)

(1) 8-bit timer mode control register 80 (TMC80)

TMC80 determines whether to enable or disable 8-bit timer register 80 (TM80) and specifies the count clock for TM80. It also controls the operation of the output control circuit of 8-bit timer counter 80.

TMC80 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TMC80 to 00H.

Figure 5-15. Format of 8-Bit Timer Mode Control Register 80

Symbol	⑦	6	5	4	3	2	1	⓪	Address	When reset	R/W
TMC80	TCE80	PWME80	0	0	0	TCL801	TCL800	TOE80	FF53H	00H	R/W

TCE80	8-bit timer register 80 operation control
0	Operation disabled (TM80 is cleared to 0.)
1	Operation enabled

PWME80	PWM output selection
0	Counter operation
1	PWM output

TCL801	TCL800	8-bit timer/event counter 80 count clock selection
0	0	f_X (5.0 MHz)
0	1	$f_X/2^8$ (19.5 kHz)
1	0	Rising edge of TI80 Note
1	1	Falling edge of TI80 Note

TOE80	8-bit timer/event counter 80 output control
0	Output disabled (port mode)
1	Output enabled

Note When an external clock is used, the timer output cannot be used.

Cautions 1. Always stop the timer before setting TMC80.

2. For PWM mode operation, TMMK80 (bit 0 of the interrupt mask flag register (MK1)) must be set.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) Port mode register 2 (PM2)

PM2 separately sets each bit of port 2 to either input or output.

To use the P27/TO80/TI80 pin for timer output, the PM27 and P27 output latches must be reset to 0.

PM2 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input loads FFH into PM2.

Figure 5-16. Format of Port Mode Register 2

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W

PM27	P27 pin input/output mode selection
0	Output mode (output buffer ON)
1	Input mode (output buffer OFF)

5.5 Clock Timer

5.5.1 Clock timer functions

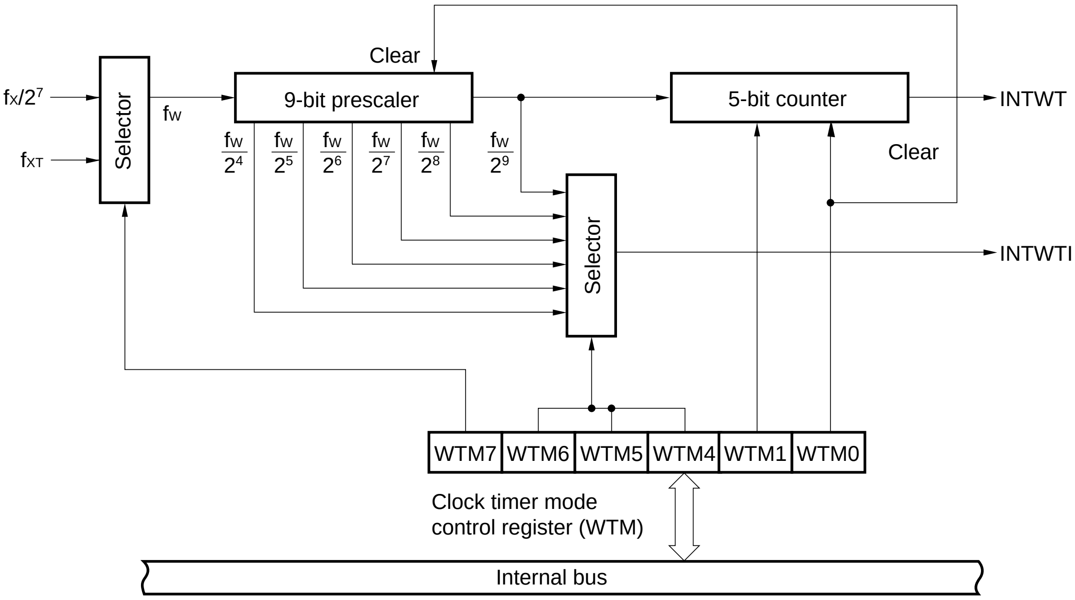
The clock timer has the following functions.

- Clock timer
- Interval timer

The clock and interval timers can be used at the same time.

Figure 5-17 is a block diagram of the clock timer.

Figure 5-17. Block Diagram of Clock Timer



(1) Clock timer

The 4.19-MHz main system clock or 32.768-kHz subsystem clock is used to issue an interrupt request (INTWT) at 0.5-second intervals.

Caution When the main system clock is operating at 5.0 MHz, it cannot be used to generate a 0.5-second interval. In this case, the subsystem clock, which operates at 32.768 kHz, should be used instead.

(2) Interval timer

The interval timer is used to generate an interrupt request (INTWTI) at specified intervals.

Table 5-7. Interval Generated Using the Interval Timer

Interval	Operation at f_X = 5.0 MHz	Operation at f_X = 4.19 MHz	Operation at f_{XT} = 32.768 kHz
$2^4 \times 1/f_W$	409.6 μs	489 μs	488 μs
$2^5 \times 1/f_W$	819.2 μs	978 μs	977 μs
$2^6 \times 1/f_W$	1.64 ms	1.96 ms	1.95 ms
$2^7 \times 1/f_W$	3.28 ms	3.91 ms	3.91 ms
$2^8 \times 1/f_W$	6.55 ms	7.82 ms	7.81 ms
$2^9 \times 1/f_W$	13.1 ms	15.6 ms	15.6 ms

Remark f_W : Clock timer clock frequency ($f_X/2^7$ or f_{XT})
f_X : Main system clock oscillation frequency
f_{XT} : Subsystem clock oscillation frequency

5.5.2 Clock timer configuration

The clock timer consists of the following hardware.

Table 5-8. Clock Timer Configuration

Item	Configuration
Counter	5 bits × 1
Prescaler	9 bits × 1
Control register	Clock timer mode control register (WTM)

5.5.3 Clock timer control register

The following register is used to control the clock timer.

- Clock timer mode control register (WTM)

  The WTM selects a count clock for the clock timer and specifies whether to enable clocking of the timer. It also specifies the prescaler interval and how the 5-bit counter is controlled.

  The WTM is manipulated using a 1-bit or 8-bit memory manipulation instruction.

  A $\overline{\text{RESET}}$ input clears the WTM to 00H.

Figure 5-18. Format of Clock Timer Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
WTM	WTM7	WTM6	WTM5	WTM4	0	0	WTM1	WTM0	FF4AH	00H	R/W

WTM7	Clock timer count clock selection
0	$f_X/2^7$ (39.1 kHz)
1	f_{XT} (32.768 kHz)

WTM6	WTM5	WTM4	Prescaler interval selection
0	0	0	$2^4/f_W$
0	0	1	$2^5/f_W$
0	1	0	$2^6/f_W$
0	1	1	$2^7/f_W$
1	0	0	$2^8/f_W$
1	0	1	$2^9/f_W$
Other settings			Not to be set

WTM1	5-bit counter operation control
0	Cleared after stop
1	Started

WTM0	Clock timer operation
0	Operation disabled (both prescaler and timer cleared)
1	Operation enabled

Remarks 1. f_W : Clock timer clock frequency ($f_X/2^7$ or f_{XT})

2. f_X : Main system clock oscillation frequency

3. f_{XT}: Subsystem clock oscillation frequency

4. The parenthesized values apply to operation at f_X = 5.0 MHz or f_{XT} = 32.768 kHz.

5.6 Watchdog Timer

5.6.1 Watchdog timer functions

The watchdog timer has the following functions.

(1) Watchdog timer

The watchdog timer is used to detect unintended program loops. If an unintended program loop is detected, a nonmaskable interrupt or $\overline{\text{RESET}}$ signal is generated.

(2) Interval timer

The interval timer is used to generate interrupts at specified intervals.

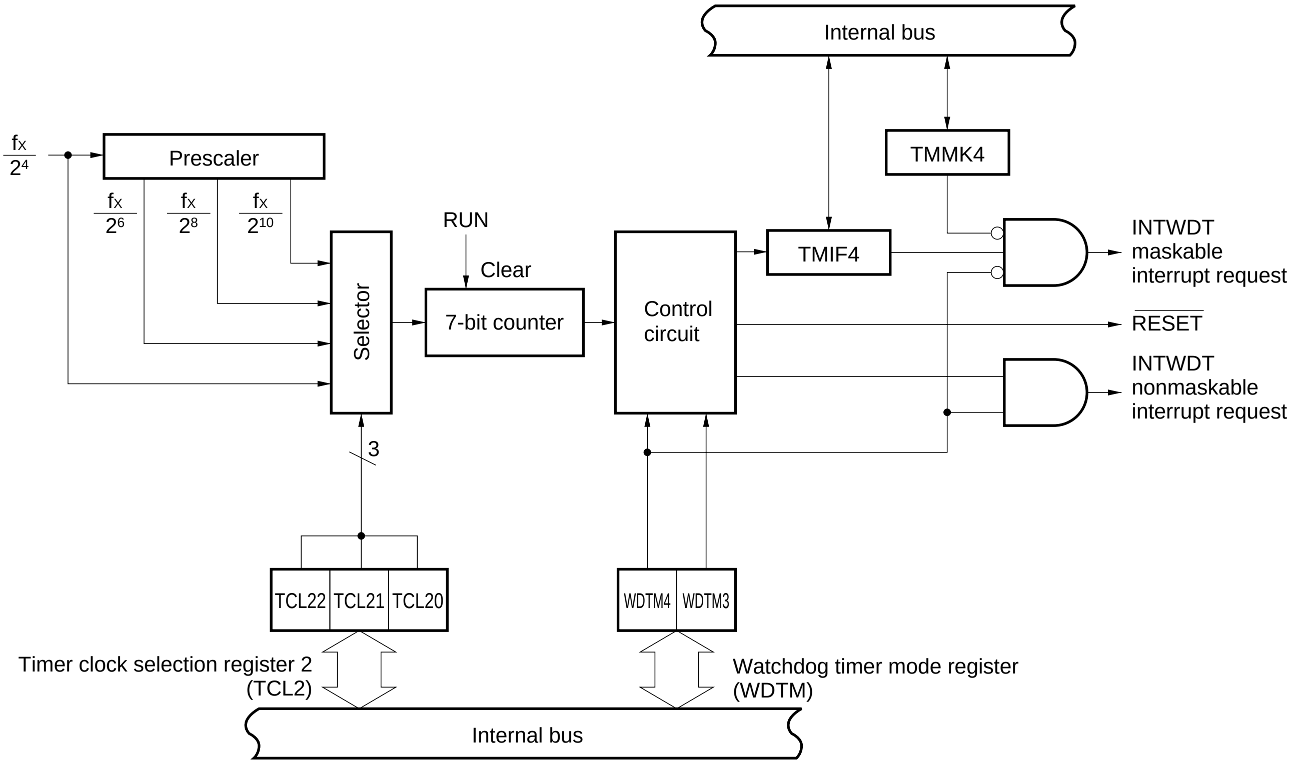
5.6.2 Watchdog timer configuration

The watchdog timer consists of the following hardware.

Table 5-9. Watchdog Timer Configuration

Item	Configuration
Control register	Timer clock selection register 2 (TCL2) Watchdog timer mode register (WDTM)

Figure 5-19. Block Diagram of Watchdog Timer



Internal bus
$f_X/2^4$
Prescaler
$f_X/2^6$
$f_X/2^8$
$f_X/2^{10}$
TMMK4
RUN
Clear
Selector
7-bit counter
Control circuit
TMIF4
INTWDT maskable interrupt request
$\overline{\text{RESET}}$
INTWDT nonmaskable interrupt request
3
TCL22
TCL21
TCL20
WDTM4
WDTM3
Timer clock selection register 2 (TCL2)
Watchdog timer mode register (WDTM)
Internal bus

5.6.3 Watchdog timer control registers

The following two types of registers are used to control the watchdog timer.

- Timer clock selection register 2 (TCL2)
- Watchdog timer mode register (WDTM)

(1) Timer clock selection register 2 (TCL2)

TCL2 specifies the count clock for the watchdog timer.

TCL2 is manipulated using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TCL2 to 00H.

Figure 5-20. Format of Timer Clock Selection Register 2

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
TCL2	0	0	0	0	0	TCL22	TCL21	TCL20	FF42H	00H	R/W

TCL22	TCL21	TCL20	Watchdog timer count clock selection	Interval time
0	0	0	$f_X/2^4$ (312.5 kHz)	$2^{11}/f_X$ (410 μs)
0	1	0	$f_X/2^6$ (78.1 kHz)	$2^{13}/f_X$ (1.64 ms)
1	0	0	$f_X/2^8$ (19.5 kHz)	$2^{15}/f_X$ (6.55 ms)
1	1	0	$f_X/2^{10}$ (4.88 kHz)	$2^{17}/f_X$ (26.2 ms)
Other settings			Not to be set	

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

(2) Watchdog timer mode register (WDTM)

The WDTM specifies the watchdog timer operation mode and whether to enable or disable counting.

The WDTM is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears the WDTM to 00H.

Figure 5-21. Format of Watchdog Timer Mode Register

Symbol	⑦	6	5	4	3	2	1	0	Address	When reset	R/W
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0	FFF9H	00H	R/W

RUN	Watchdog timer operation selection[Note 1]
0	Stops counting.
1	Clears the counter and causes it to start.

WDTM4	WDTM3	Watchdog timer operation mode selection[Note 2]
0	0	Operation disabled
0	1	Internal timer mode (When an overflow occurs, a maskable interrupt is issued.)[Note 3]
1	0	Watchdog timer mode 1 (When an overflow occurs, a nonmaskable interrupt is issued.)
1	1	Watchdog timer mode 2 (When an overflow occurs, a reset operation is started.)

Notes 1. Once the RUN bit has been set (1), it is impossible to zero-clear it by software. So, once counting begins, it cannot be stopped by any means other than a $\overline{\text{RESET}}$ input.

2. Once WDTM3 and WDTM4 have been set (1), it is impossible to zero-clear them by software.

3. The interval timer starts operating when the RUN bit is set to 1.

Cautions 1. If the RUN bit is set to 1, and the watchdog timer is cleared, the actual overflow time becomes 0.8% (maximum) less than the time specified in timer clock selection register 2.

2. To use watchdog timer mode 1 or 2, ensure that TMIF4 (bit 0 of the interrupt request flag register 0 (IF0)) is set to 0, before setting WDTM4 (bit 0 of the interrupt mask flag register 0 (MK0)) to 1. If TMIF4 is set to 1, selecting mode 1 or 2 causes a nonmaskable interrupt to be issued at the instant rewriting ends.

5.7 Serial Interface

5.7.1 Serial interface 20 functions

Serial interface 20 has the following three types of modes.

- Operation stopped mode
- Asynchronous serial interface (UART) mode
- Three-wire serial I/O mode

(1) Operation stopped mode

This mode is used when serial transfer is not performed. Power consumption is minimized in this mode.

(2) Asynchronous serial interface (UART) mode

This mode is used to send and receive the one byte of data that follows a start bit. It supports full-duplex communication.

Serial interface 20 contains a dedicated UART baud rate generator, enabling communication over a wide range of baud rates. It is also possible to define baud rates by dividing the frequency of the input clock pulse at the ASCK20 pin.

(3) Three-wire serial I/O mode (switchable between MSB-first and LSB-first transmission)

This mode is used to transmit 8-bit data, using three lines: a serial clock ($\overline{\text{SCK20}}$) line and two serial data lines (SI20 and SO20).

As it supports simultaneous transmission and reception, three-wire serial I/O mode requires less processing time for data transmission than asynchronous serial interface mode.

Because, in three-wire serial I/O mode, it is possible to select whether 8-bit data transmission begins with the MSB or LSB, serial interface 20 can be connected to any device regardless of whether that device is designed for MSB-first or LSB-first transmission.

Three-wire serial I/O mode is useful for connecting peripheral I/O circuits and display controllers having conventional clock synchronous serial interfaces, such as those of the 75X/XL, 78K, and 17K series devices.

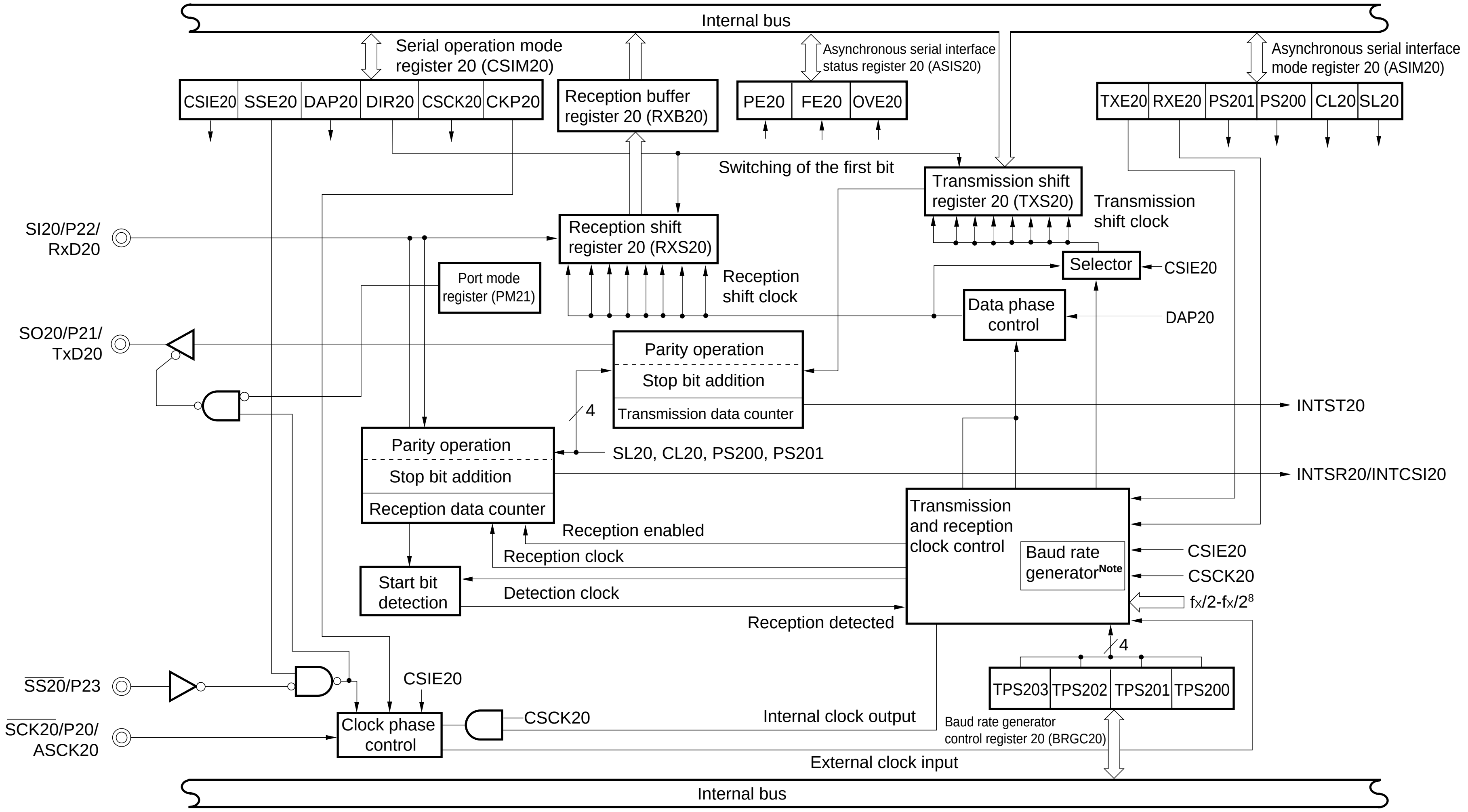
5.7.2 Serial interface 20 configuration

Serial interface 20 consists of the following hardware.

Table 5-10. Serial Interface 20 Configuration

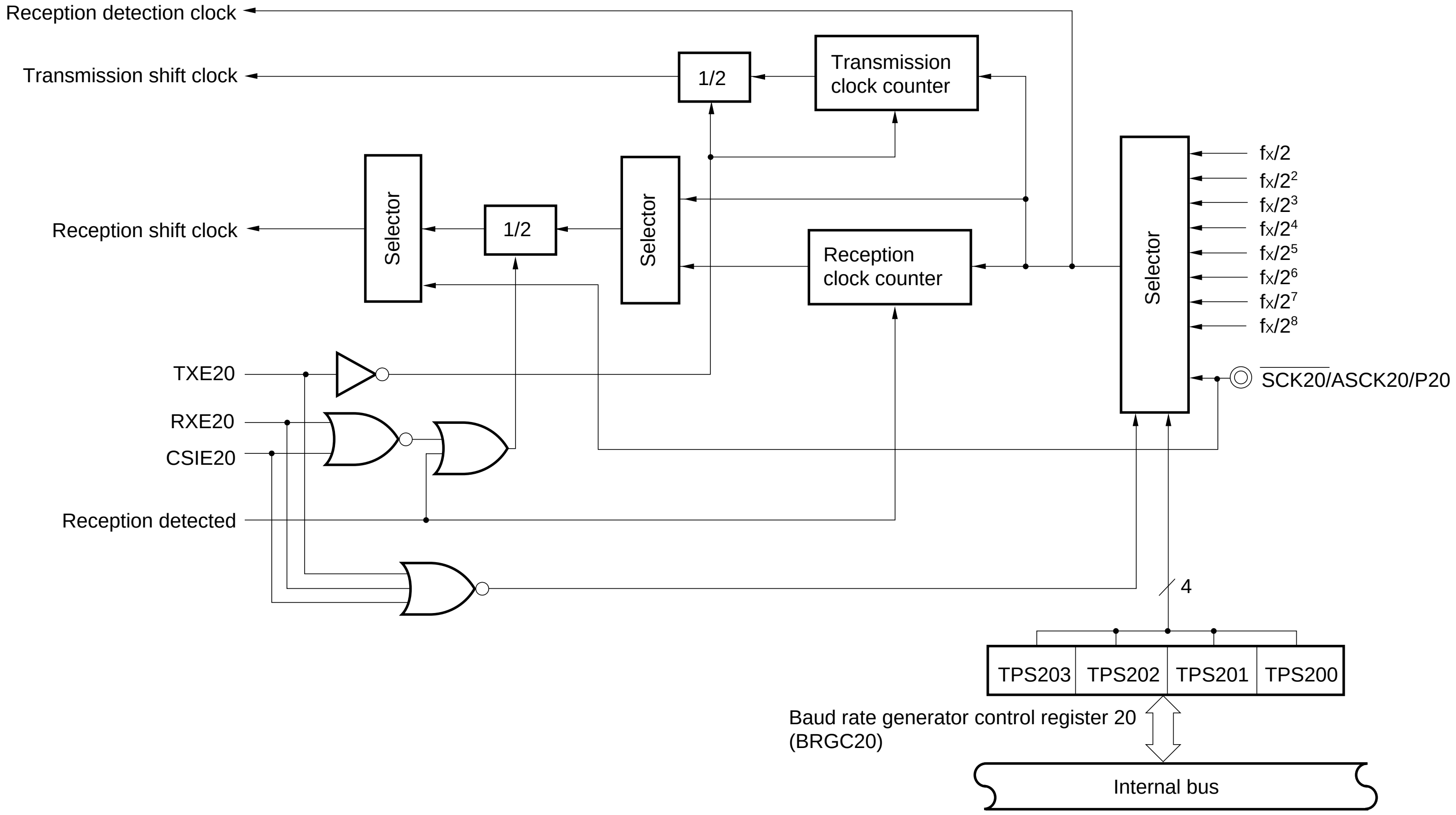
Item	Configuration
Register	Transmission shift register 20 (TXS20) Reception shift register 20 (RXS20) Reception buffer register 20 (RXB20)
Control register	Serial operation mode register 20 (CSIM20) Asynchronous serial interface mode register 20 (ASIM20) Asynchronous serial interface status register 20 (ASIS20) Baud rate generator control register 20 (BRGC20)

Figure 5-22. Block Diagram of Serial Interface 20



Note See **Figure 5-23** for the configuration of the baud rate generator.

Figure 5-23. Block Diagram of Baud Rate Generator 20



(1) Transmission shift register 20 (TXS20)

TXS20 is a register in which transmission data is prepared. The transmission data is output from the TXS20 bit-serially.

When the data length is seven bits, bits 0 to 6 of the data in TXS20 will be transmission data. Writing data to TXS20 triggers transmission.

TXS20 can be write-accessed, using an 8-bit memory manipulation instruction, but cannot be read-accessed.

A $\overline{\text{RESET}}$ input loads FFH into TXS20.

Caution Do not write to TXS20 during transmission.

TXS20 and the reception buffer register 20 (RXB20) are mapped at the same address, such that any attempt to read from TXS20 results in a value being read from the RXB.

(2) Reception shift register 20 (RXS20)

RXS20 is a register in which serial data, received at the RxD20 pin, is converted to parallel data. Once one entire byte has been received, RXS20 feeds the reception data to the reception buffer register 20 (RXB20). RXS20 cannot be manipulated directly by a program.

(3) Reception buffer register 20 (RXB20)

RXB20 is used to hold reception data. Once RXS20 has received one entire byte of data, it feeds that data into RXB20.

When the data length is seven bits, the reception data is sent to bits 0 to 6 of RXB20, in which the MSB is fixed to 0.

RXB20 can be read-accessed, using an 8-bit memory manipulation instruction, but cannot be write-accessed.

A $\overline{\text{RESET}}$ input makes RXB20 undefined.

Caution RXB20 and the transmission shift register 20 (TXS20) are mapped at the same address, such that any attempt to write to RXB20 results in a value being written to TXS20.

(4) Transmission control circuit

The transmission control circuit controls transmission. For example, it adds start, parity, and stop bits to the data in the transmission shift register 20 (TXS20), according to the setting of the asynchronous serial interface mode register 20 (ASIM20).

(5) Reception control circuit

The reception control circuit controls reception according to the setting of the asynchronous serial interface mode register 20 (ASIM20). It also checks for errors, such as parity errors, during reception. If an error is detected, the asynchronous serial interface status register 20 (ASIS20) is set according to the status of the error.

5.7.3 Serial interface 20 control registers

The following four types of registers are used to control serial interface 20.

- Serial operation mode register 20 (CSIM20)
- Asynchronous serial interface mode register 20 (ASIM20)
- Asynchronous serial interface status register 20 (ASIS20)
- Baud rate generator control register 20 (BRGC20)

(1) Serial operation mode register 20 (CSIM20)

CSIM20 is used to make the settings related to three-wire serial I/O mode.

CSIM20 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears CSIM20 to 00H.

Figure 5-24. Format of Serial Operation Mode Register 20

Symbol	⑦	6	5	4	3	2	1	0	Address	When reset	R/W
CSIM20	CSIE20	SSE20	0	0	DAP20	DIR20	CSCK20	CKP20	FF72H	00H	R/W

CSIE20	Three-wire serial I/O mode operation control
0	Operation disabled
1	Operation enabled

SSE20	$\overline{\text{SS20}}$-pin selection	Functions of the $\overline{\text{SS20}}$/P23 pin	Communication status
0	Not used	Port function	Communication enabled
1	Used	0	Communication enabled
		1	Communication disabled

DAP20	Three-wire serial I/O mode data phase selection
0	Outputs at the falling edge of $\overline{\text{SCK20}}$.
1	Outputs at the rising edge of $\overline{\text{SCK20}}$.

DIR20	First-bit specification
0	MSB
1	LSB

CSCK20	Three-wire serial I/O mode clock selection
0	External clock pulse input to the $\overline{\text{SCK20}}$ pin
1	Output of the dedicated baud rate generator

CKP20	Three-wire serial I/O mode clock phase selection
0	Clock is low active; $\overline{\text{SCK20}}$ is high in the idle state
1	Clock is high active; $\overline{\text{SCK20}}$ is low in the idle state

Cautions 1. Bits 4 and 5 must be fixed to 0.

2. CSIM20 must be cleared to 00H, if UART mode is selected.

(2) Asynchronous serial interface mode register 20 (ASIM20)

ASIM20 is used to make the settings related to serial interface 20 used in asynchronous serial interface mode.

ASIM20 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears ASIM20 to 00H.

Figure 5-25. Format of Asynchronous Serial Interface Mode Register 20

Symbol	⑦	⑥	5	4	3	2	1	0	Address	When reset	R/W
ASIM20	TXE20	RXE20	PS201	PS200	CL20	SL20	0	0	FF70H	00H	R/W

TXE20	Transmission control
0	Transmission disabled
1	Transmission enabled

RXE20	Reception control
0	Reception disabled
1	Reception enabled

PS201	PS200	Parity bit specification
0	0	No parity
0	1	At transmission, the parity bit is fixed to 0. At reception, a parity check is not made; no parity error is reported.
1	0	Odd parity
1	1	Even parity

CL20	Transmission data character length specification
0	7 bits
1	8 bits

SL20	Transmission data stop bit length specification
0	1 bit
1	2 bits

Cautions 1. Bits 0 and 1 must be fixed to 0.

2. If three-wire serial I/O mode is selected, ASIM20 must be cleared to 00H.

3. Switch operation mode from one mode to another after stopping both serial transmission and reception.

Table 5-11. Serial Interface 20 Operation Mode Settings

(1) Operation stopped mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First bit	Shift clock	P22/SI20/RxD20 pin function	P21/SO20/TxD20 pin function	P20/$\overline{\text{SCK20}}$/ASCK20 pin function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
0	0	0	×	×	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	-	-	P22	P21	P20
Other settings											Not to be set				

(2) Three-wire serial I/O mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First bit	Shift clock	P22/SI20/RxD20 pin function	P21/SO20/TxD20 pin function	P20/$\overline{\text{SCK20}}$/ASCK20 pin function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
0	0	1	0	0	1[Note 2]	×[Note 2]	0	1	1	×	MSB	External clock	SI20[Note 2]	SO20 (CMOS output)	$\overline{\text{SCK20}}$ input
				1					0	1		Internal clock			$\overline{\text{SCK20}}$ output
		1	1	0					1	×	LSB	External clock			$\overline{\text{SCK20}}$ input
				1					0	1		Internal clock			$\overline{\text{SCK20}}$ output
Other settings											Not to be set				

(3) Asynchronous serial interface mode

ASIM20		CSIM20			PM22	P22	PM21	P21	PM20	P20	First bit	Shift clock	P22/SI20/RxD20 pin function	P21/SO20/TxD20 pin function	P20/$\overline{\text{SCK20}}$/ASCK20 pin function
TXE20	RXE20	CSIE20	DIR20	CSCK20											
1	0	0	0	0	×[Note 1]	×[Note 1]	0	1	1	×	LSB	External clock	P22	TxD20 (CMOS output)	ASCK20 input
									×[Note 1]	×[Note 1]		Internal clock			P20
0	1	0	0	0	1	×	×[Note 1]	×[Note 1]	1	×		External clock	RxD20	P21	ASCK20 input
									×[Note 1]	×[Note 1]		Internal clock			P20
1	1	0	0	0	1	×	0	1	1	×		External clock		TxD20 (CMOS output)	ASCK20 input
									×[Note 1]	×[Note 1]		Internal clock			P20
Other settings											Not to be set				

Notes 1. These pins can be used for port functions.

2. When only transmission is used, these pins can be used as P22 (CMOS input/output).

Remark ×: Don't care.

(3) Asynchronous serial interface status register 20 (ASIS20)

ASIS20 is used to display the type of a reception error, if it occurs while asynchronous serial interface mode is set.

ASIS20 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

The contents of ASIS20 are undefined in three-wire serial I/O mode.

A $\overline{\text{RESET}}$ input clears ASIS20 to 00H.

Figure 5-26. Format of Asynchronous Serial Interface Status Register 20

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
ASIS20	0	0	0	0	0	PE20	FE20	OVE20	FF71H	00H	R

PE20	Parity error flag
0	No parity error has occurred.
1	A parity error has occurred (parity mismatch in transmission data).

FE20	Framing error flag
0	No framing error has occurred.
1	A framing error has occurred (no stop bit detected).Note 1

OVE20	Overrun error flag
0	No overrun error has occurred.
1	An overrun error has occurred.Note 2 (Before data was read from the reception buffer register, the subsequent recepiton sequence was completed.)

Notes 1. Even if 2 is specified for the number of stop bits (using bit 2 (SL20) of ASIM20), only one stop bit is detected at reception.

2. After an overrun occurs, read-access the reception buffer register 20 (RXB20). Otherwise, the overrun error will recur each time data is received.

(4) Baud rate generator control register 20 (BRGC20)

BRGC20 is used to specify the serial clock for serial interface 20.

BRGC20 is manipulated using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears BRGC20 to 00H.

Figure 5-27. Format of Baud Rate Generator Control Register 20

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
BRGC20	TPS203	TPS202	TPS201	TPS200	0	0	0	0	FF73H	00H	R/W

TPS203	TPS202	TPS201	TPS200	3-bit counter source clock selection	n
0	0	0	0	$f_X/2$ (2.5 MHz)	1
0	0	0	1	$f_X/2^2$ (1.25 MHz)	2
0	0	1	0	$f_X/2^3$ (625 kHz)	3
0	0	1	1	$f_X/2^4$ (313 kHz)	4
0	1	0	0	$f_X/2^5$ (156 kHz)	5
0	1	0	1	$f_X/2^6$ (78.1 kHz)	6
0	1	1	0	$f_X/2^7$ (39.1 kHz)	7
0	1	1	1	$f_X/2^8$ (19.5 kHz)	8
1	0	0	0	External clock pulse input at the ASCK20 pin[Note]	-
Other settings				Not to be set	

Note An external clock can be used only in UART mode.

Cautions 1. Any attempt to write to BRGC20 during communication adversely affects the output of the baud rate generator, thus hampering normal operation. Therefore, do not write to BRGC20 during communication.

2. Do not select n = 1 during operation at f_X = 5.0 MHz, as n = 1 causes the rated baud rate to be exceeded.

3. When the external input clock is selected, set P20 in input mode (PM20 (bit 0 of the port mode register 2) = 1)

Remarks 1. f_X : Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

The transmission and reception clock pulses used to generate the baud rate are obtained by dividing the frequency of the main system clock pulse or a signal input to the ASCK20 pin.

(a) Generating transmission and reception clock pulses for baud rates based on the main system clock

The frequency of the main system clock is divided to generate the transmission and reception clock pulses. The baud rate generated based on the main system clock is determined using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} \ [\text{Hz}]$$

f_X : Main system clock oscillation frequency

Table 5-12. Relationships between Main System Clock Frequencies and Baud Rates (Example)

Baud rate (bps)	n	BRGC20 setting	Error (%)	
			f_X = 5.0 MHz	f_X = 4.9152 MHz
1,200	8	70H	1.73	0
2,400	7	60H		
4,800	6	50H		
9,600	5	40H		
19,200	4	30H		
38,400	3	20H		
76,800	2	10H		

Caution Do not select n = 1 during operation at f_X = 5.0 MHz, as n = 1 causes the rated baud rate to be exceeded.

(b) Generating transmission and reception clock pulses for baud rates based on an external clock pulse received at the ASCK20 pin

The frequency of an external clock pulse received at the ASCK20 pin is used to generate the transmission and reception clock pulses. The baud rate generated based on the external clock pulse received at the ASCK20 pin is determined using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16} \ [\text{Hz}]$$

f_{ASCK}: Frequency of clock pulse received at the ASCK20 pin

Table 5-13. Relationships between ASCK20 Pin Input Frequencies and Baud Rates (When BRGC20 = 80H)

Baud rate (bps)	ASCK20 pin input frequency (kHz)
75	1.2
150	2.4
300	4.8
600	9.6
1,200	19.2
2,400	38.4
4,800	76.8
9,600	153.6
19,200	307.2
31,250	500.0
38,400	614.4

6. INTERRUPT FUNCTIONS

6.1 Interrupt Function Types

Two types of interrupt function are supported.

(1) Nonmaskable interrupt

A nonmaskable interrupt request is accepted unconditionally, that is, even when interrupts are disabled. A nonmaskable interrupt takes precedence over all other interrupts; it is not subjected to interrupt priority control.

A nonmaskable interrupt causes the standby release signal to be generated.

The μPD789046 supports one nonmaskable interrupt source namely, the watchdog timer interrupt.

(2) Maskable interrupt

Maskable interrupts are those which are subjected to mask control. If two or more maskable interrupts occur simultaneously, the default priority listed in Table 6-1 applies.

The maskable interrupts cause the standby release signal to be generated.

The maskable interrupts supported by the μPD789046 include 4 external interrupt sources and 7 internal interrupt sources.

6.2 Interrupt Sources and Configuration

The μPD789046 supports a total of 12 maskable and nonmaskable interrupt sources. (See **Table 6-1**.)

Table 6-1. Interrupt Sources

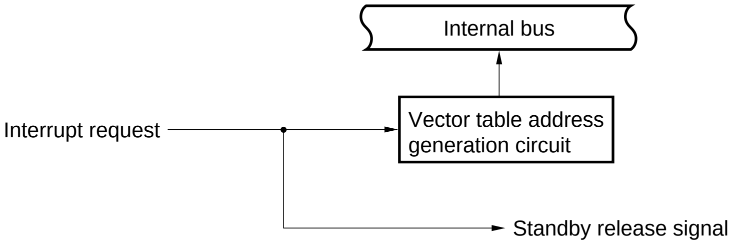
Interrupt type	Priority[Note 1]	Interrupt source		Internal/external	Vector table address	Basic configuration type[Note 2]
		Name	Trigger			
Nonmaskable interrupt	–	INTWDT	Watchdog timer overflow (when watchdog timer mode 1 is selected)	Internal	0004H	(A)
Maskable interrupt	0	INTWDT	Watchdog timer overflow (when the interval timer mode is selected)			(B)
	1	INTP0	Pin input edge detection	External	0006H	(C)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTSR20	End of UART reception on serial interface 20	Internal	000CH	(B)
		INTCSI20	End of three-wire SIO transfer reception on serial interface 20			
	5	INTST20	End of UART transmission on serial interface 20		000EH	
	6	INTWT	Clock timer interrupt		0010H	
	7	INTWTI	Interval timer interrupt		0012H	
	8	INTTM80	Generation of match signal for 8-bit timer/event counter 80		0014H	
	9	INTTM90	Generation of match signal for 16-bit timer counter 90		0016H	
	10	INTKR00	Detection of key return signal	External	0018H	(C)

Notes 1. The priority regulates which maskable interrupt is higher, when two or more maskable interrupts are requested simultaneously. Zero signifies the highest priority, while 10 is the lowest.

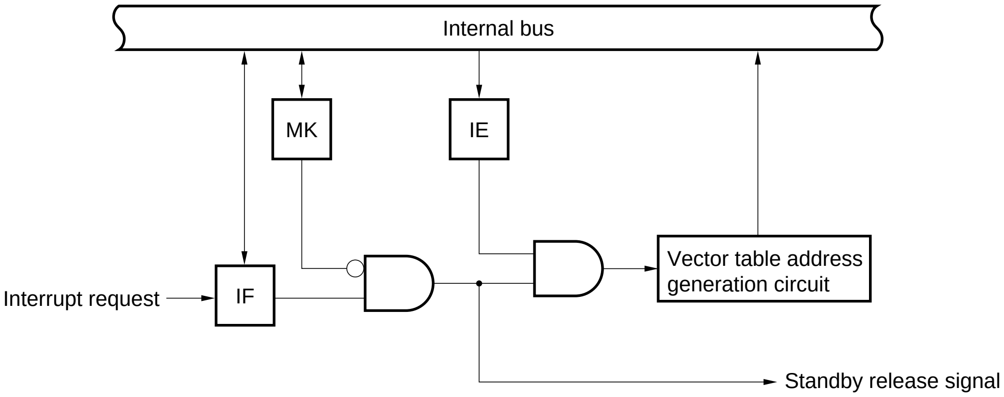
2. Basic configuration types (A), (B), and (C) correspond to (A), (B), and (C) in Figure 6-1, respectively.

Figure 6-1. Basic Configuration of Interrupt Functions

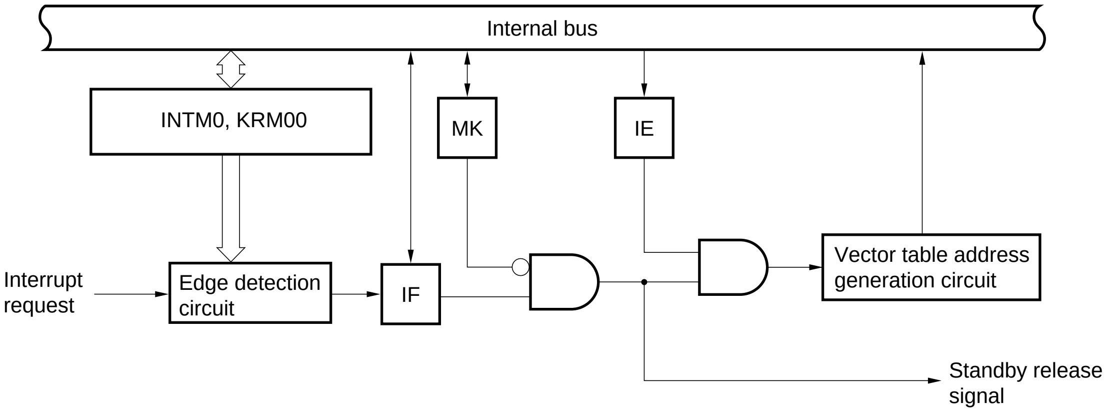
(A) Internal nonmaskable interrupt



(B) Internal maskable interrupt



(C) External maskable interrupt



INTM0	:	External interrupt mode register 0
KRM00	:	Key return mode register 00
IF	:	Interrupt request flag
IE	:	Interrupt enable flag
MK	:	Interrupt mask flag

6.3 Interrupt Function Control Registers

The interrupt functions are controlled by the following registers.

- Interrupt request flag registers 0 and 1 (IF0 and IF1)
- Interrupt mask flag registers 0 and 1 (MK0 and MK1)
- External interrupt mode register 0 (INTM0)
- Program status word (PSW)
- Key return mode register 00 (KRM00)

Table 6-2 lists interrupt requests, the corresponding interrupt request flags, and interrupt mask flags.

Table 6-2. Interrupt Request Signals and Corresponding Flags

Interrupt request signal	Interrupt request flag	Interrupt mask flag
INTWDT	TMIF4	TMMK4
INTP0	PIF0	PMK0
INTP1	PIF1	PMK1
INTP2	PIF2	PMK2
INTSR20/INTCSI20	SRIF20	SRMK20
INTST20	STIF20	STMK20
INTWT	WTIF	WTMK
INTWTI	WTIIF	WTIMK
INTTM80	TMIF80	TMMK80
INTTM90	TMIF90	TMMK90
INTKR00	KRIF00	KRMK00

(1) Interrupt request flag registers (IF0 and IF1)

An interrupt request flag is set (1), when the corresponding interrupt request is issued, or when the related instruction is executed. It is cleared (0), when the interrupt request is accepted, when a $\overline{\text{RESET}}$ signal is input, or when a related instruction is executed.

IF0 and IF1 are manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears IF0 and IF1 to 00H.

Figure 6-2. Format of Interrupt Request Flag Register

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	When reset	R/W
IF0	WTIIF	WTIF	STIF20	SRIF20	PIF2	PIF1	PIF0	TMIF4	FFE0H	00H	R/W
	7	6	5	4	3	②	①	⓪			
IF1	0	0	0	0	0	KRIF00	TMIF90	TMIF80	FFE1H	00H	R/W

XXIFX	Interrupt request flag
0	No interrupt request signal has been issued.
1	An interrupt request signal has been issued; an interrupt request has been made.

Cautions 1. Bits 3 to 7 of IF1 must be fixed to 0.

2. The TMIF4 flag can be read- and write-accessed only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.

3. When port 2 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 2 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

(2) Interrupt mask flag registers (MK0 and MK1)

The interrupt mask flags are used to enable and disable the corresponding maskable interrupts.

MK0 and MK1 are manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input loads FFH into MK0 and MK1.

Figure 6-3. Format of Interrupt Mask Flag Register

Symbol	⑦	⑥	⑤	④	③	②	①	⓪	Address	When reset	R/W
MK0	WTIMK	WTMK	STMK20	SRMK20	PMK2	PMK1	PMK0	TMMK4	FFE4H	FFH	R/W
	7	6	5	4	3	②	①	⓪			
MK1	1	1	1	1	1	KRMK00	TMMK90	TMMK80	FFE5H	FFH	R/W

XXMKX	Interrupt handling control
0	Enable interrupt handling.
1	Disable interrupt handling.

Cautions 1. Bits 3 to 7 of MK1 must be fixed to 1.

2. When the watchdog timer is being used in watchdog timer mode 1 or 2, any attempt to read TMMK4 flag results in an undefined value being detected.

3. When port 2 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 2 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.

(3) External interrupt mode register 0 (INTM0)

INTM0 is used to specify an effective edge for INTP0 to INTP2.

INTM0 is manipulated using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears INTM0 to 00H.

Figure 6-4. Format of External Interrupt Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
INTM0	ES21	ES20	ES11	ES10	ES01	ES00	0	0	FFECH	00H	R/W

ES21	ES20	INTP2 effective edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Not to be set
1	1	Both rising and falling edges

ES11	ES10	INTP1 effective edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Not to be set
1	1	Both rising and falling edges

ES01	ES00	INTP0 effective edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Not to be set
1	1	Both rising and falling edges

Cautions 1. Bits 0 and 1 must be fixed to 0.

2. Before setting INTM0, set the corresponding interrupt mask flag to 1 to disable interrupts. To enable interrupts, clear (0) the corresponding interrupt request flag, then the corresponding interrupt mask flag.

(4) Program status word (PSW)

The program status word is used to hold the instruction execution result and the current status of the interrupt requests. The IE flag, used to enable and disable maskable interrupts, is mapped to the PSW.

The PSW can be read- and write-accessed in 8-bit units, as well as in 1-bit units when using bit manipulation instructions and dedicated instructions (EI and DI). When a vector interrupt is accepted, the PSW is automatically saved to a stack, and the IE flag is reset (0).

A $\overline{\text{RESET}}$ input loads 02H into the PSW.

Figure 6-5. Program Status Word Configuration

Symbol	7	6	5	4	3	2	1	0	When reset
PSW	IE	Z	0	AC	0	0	1	CY	02H

Z, AC, CY → Used in the execution of ordinary instructions

IE	Whether to enable/disable interrupt acceptance
0	Disable
1	Enable

(5) Key return mode register 00 (KRM00)

KRM00 is used to specify pins for which the key return signals are detected.

KRM00 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

Bit 0 (KRM000) specifies whether the detection is performed for four pins from $\overline{\text{KR00}}$/P40 to $\overline{\text{KR03}}$/P43 together.

Bits 4 to 7 (KRM004 to KRM007) specify whether the detection is performed for the $\overline{\text{KR04}}$/P44 to $\overline{\text{KR07}}$/P47 pins individually.

A $\overline{\text{RESET}}$ input clears KRM00 to 00H.

Figure 6-6. Format of Key Return Mode Register 00

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
KRM00	KRM007	KRM006	KRM005	KRM004	0	0	0	KRM000	FFF5H	00H	R/W

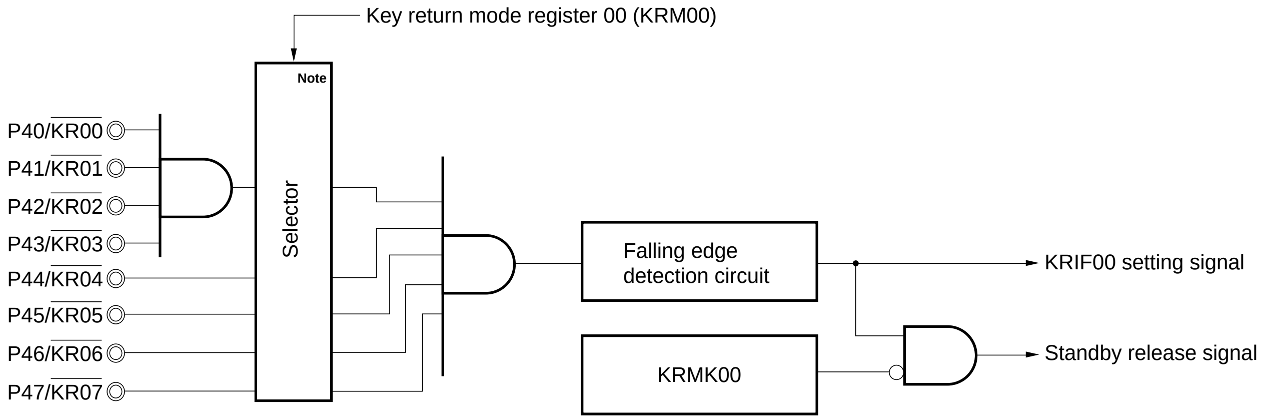
KRMn	Key return signal detection selection
0	Not detected
1	Detected (Falling edges of port 4)

Cautions 1. Bits 1 to 3 must be fixed to 0.

2. When a bit of KRM00 is set to 1, the pull-up resistor is forcibly connected to the corresponding pin. If the pin is set to output mode, however, the pull-up resistor is left disconnected.

3. Before setting KRM00, set bit 2 of MK1 to 1 (KRMK00 = 1) to 1 to disable interrupts. After KRM00 is set, clear bit 2 of IF1 (KRIF = 0), then clear KRMK00 (KRMK00 = 0) to enable interrupts.

Figure 6-7. Block Diagram of Falling Edge Detection Circuit



Note This selector selects pins to be used for falling-edge inputs.

7. STANDBY FUNCTION

7.1 Standby Function

The standby function is supported to minimize the system's power consumption. There are two standby modes: HALT and STOP.

HALT and STOP modes are selected using the HALT and STOP instructions, respectively.

(1) HALT mode

In HALT mode, the CPU clock is stopped. Interleaving normal mode with HALT mode can reduce the average power consumption.

(2) STOP mode

In STOP mode, the main system clock is stopped. As a result, main system clock-based operation is also stopped, thus minimizing power consumption.

Caution Before shifting to STOP mode, first stop the operation of the peripheral hardware, then execute the STOP instruction.

Table 7-1. Operation Statuses in HALT Mode

Item	HALT mode operation status while the main system clock is running		HALT mode operation status while the subsystem clock is running	
	While the subsystem clock is running	While the subsystem clock is not running	While the main system clock is running	While the main system clock is not running
Main system clock generator	Oscillation enabled			Oscillation disabled
CPU	Operation disabled			
Port (output latch)	Remains in the state existing before the selection of HALT mode.			
16-bit timer counter	Operation enabled	Operation enabled[Note 1]	Operation enabled	Operation enabled[Note 2]
8-bit timer/event counter	Operation enabled			Operation enabled[Note 3]
Clock timer	Operation enabled	Operation enabled[Note 1]	Operation enabled	Operation enabled[Note 2]
Watchdog timer	Operation enabled		Operation disabled	
Serial interface	Operation enabled			Operation enabled[Note 4]
External interrupt	Operation enabled[Note 5]			

Notes

1. Operation is enabled while the main system clock is selected.
2. Operation is enabled while the subsystem clock is selected.
3. Operation is enabled only when TI80 is selected as the count clock.
4. Operation is enabled in both three-wire serial I/O and UART modes while an external clock is being used.
5. Maskable interrupt that is not masked

Table 7-2. Operation Statuses in STOP Mode

Item	STOP mode operation status while the main system clock is running	
	While the subsystem clock is running	While the subsystem clock is not running
Main system clock generator	Oscillation disabled	
CPU	Operation disabled	
Port (output latch)	Remains in the state existing before the selection of STOP mode.	
16-bit timer counter	Operation enabled[Note 1]	Operation disabled
8-bit timer/event counter	Operation enabled[Note 2]	
Clock timer	Operation enabled[Note 1]	Operation disabled
Watchdog timer	Operation disabled	
Serial interface	Operation enabled[Note 3]	
External interrupt	Operation enabled[Note 4]	

Notes **1.** Operation is enabled while the subsystem clock is selected.

2. Operation is enabled only when TI80 is selected as the count clock.

3. Operation is enabled in both three-wire serial I/O and UART modes while an external clock is being used.

4. Maskable interrupt that is not masked

7.2 Standby Function Control Register

The oscillation settling time selection register (OSTS) is used to control the wait time, from the time STOP mode is deselected by an interrupt request, until oscillation settles.

The OSTS is manipulated using an 8-bit memory manipulation instruction.

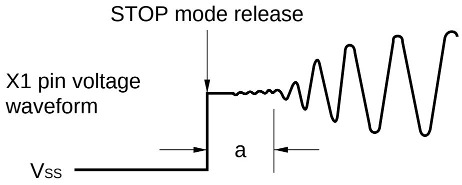
A $\overline{\text{RESET}}$ input loads 04H into the OSTS. If a $\overline{\text{RESET}}$ input is used to deselect STOP mode, the time required for oscillation to settle will be $2^{15}/f_X$, rather than $2^{17}/f_X$.

Figure 7-1. Format of Oscillation Settling Time Selection Register

Symbol	7	6	5	4	3	2	1	0	Address	When reset	R/W
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0	FFFAH	04H	R/W

OSTS2	OSTS1	OSTS0	Oscillation settling time selection
0	0	0	$2^{12}/fx$ (819 μs)
0	1	0	$2^{15}/fx$ (6.55 ms)
1	0	0	$2^{17}/fx$ (26.2 ms)
Other settings			Not to be set

Caution **The wait time required to deselect STOP mode does not include the time ("a" in the following figure) required for the clock oscillation to settle after STOP mode is deselected, regardless of whether STOP mode is deselected by a $\overline{\text{RESET}}$ input or interrupt.**



Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

8. RESET FUNCTIONS

The μPD789046 can be reset using the following signals.

(1) External reset signal input to the $\overline{\text{RESET}}$ pin

(2) Internal reset signal generated upon the elapse of the period set in the watchdog timer, used for detecting an unintended program loop

The external and internal reset signals are functionally equivalent. When $\overline{\text{RESET}}$ is input, they cause program execution to begin at the addresses indicated at addresses 0000H and 0001H, respectively.

If a low level signal is applied to the $\overline{\text{RESET}}$ pin, or if the watchdog timer overflows, a reset occurs, causing each piece of the hardware to enter the states listed in Table 8-1. While a reset signal is being input, or while the oscillation frequency is settling immediately after the end of a reset sequence, each pin remains in the high-impedance state.

If a high level signal is applied to the $\overline{\text{RESET}}$ pin, a reset sequence is terminated, and program execution begins once the oscillation settling time ($2^{15}/f_X$) elapses. A watchdog timer overflow-based reset sequence is terminated automatically. Similarly, program execution begins upon the elapse of the oscillation settling time ($2^{15}/f_X$).

Cautions 1. To use an external reset sequence, supply a low level signal to the $\overline{\text{RESET}}$ pin and maintain the signal for at least 10 μs.

2. When a reset is used to deselect STOP mode, the information related to STOP mode is held during the reset sequence, that is, while the reset signal is applied. The port pins remain in the high-impedance state, however.

Figure 8-1. Reset Function Block Diagram

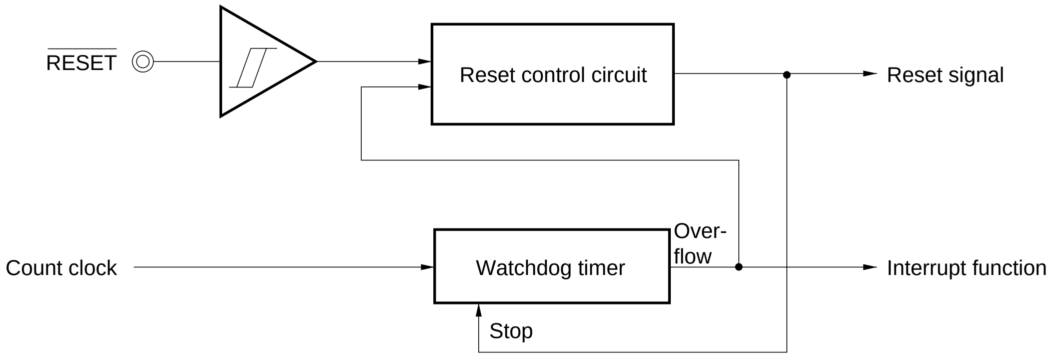


Table 8-1. State of the Hardware after a Reset

Hardware		State after reset
Program counter (PC)[Note 1]		Loaded with the contents of the reset vector table (0000H, 0001H)
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose register	Undefined[Note 2]
Ports (P0 to P4) (output latch)		00H
Port mode registers (PM0 to PM4)		FFH
Pull-up resistor option registers (PU0, PUB2)		00H
Processor clock control register (PCC)		02H
Suboscillation mode register (SCKM)		00H
Subclock control register (CSS)		00H
Oscillation settling time selection register (OSTS)		04H
16-bit timer counter	Timer register (TM90)	0000H
	Compare register (CR90)	FFFFH
	Capture register (TCP90)	Undefined
	Mode control register (TMC90)	00H
	Buzzer output control register (BZC90)	00H
8-bit timer/event counter	Timer register (TM80)	00H
	Compare register (CR80)	Undefined
	Mode control register (TMC80)	00H
Clock timer	Mode control register (WTM)	00H
Watchdog timer	Timer clock selection register (TCL2)	00H
	Mode register (WDTM)	00H
Serial interface	Mode register (CSIM20)	00H
	Asynchronous serial interface mode register (ASIM20)	00H
	Asynchronous serial interface status register (ASIS20)	00H
	Baud rate generator control register (BRGC20)	00H
	Transmission shift register (TXS20)	FFH
	Reception buffer register (RXB20)	Undefined
Interrupts	Request flag registers (IF0, IF1)	00H
	Mask flag registers (MK0, MK1)	FFH
	External interrupt mode register (INTM0)	00H
	Key return mode register (KRM00)	00H

Notes 1. While a reset signal is being input, and during the oscillation settling period, the contents of the PC will be undefined, while the remainder of the hardware will be the same as after the reset.

2. In standby mode, the RAM enters the hold state after a reset.

9. INSTRUCTION SET OVERVIEW

The instruction set for the μPD789046 is listed later.

9.1 Legend

9.1.1 Operand formats and descriptions

The description made in the operand field of each instruction conforms to the operand format for the instructions listed below (the details conform with the assembly specification). If more than one operand format is listed for an instruction, one is selected. Uppercase letters, #, !, $, and a pair of [and] are used to specify keywords, which must be written exactly as they appear. The meanings of these special characters are as follows:

- #: Immediate data specification
- $: Relative address specification
- !: Absolute address specification
- [and]: Indirect address specification

Immediate data should be described using appropriate values or labels. The specification of values and labels must be accompanied by #, !, $, or a pair of [and].

Operand registers, expressed as r or rp in the formats, can be described using both functional names (X, A, C, etc.) and absolute names (R0, R1, R2, and other names listed in Table 9-1).

Table 9-1. Operand Formats and Descriptions

Format	Description
r rp sfr	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special function register symbol
saddr saddrp	FE20H to FF1FH: Immediate data or label FE20H to FF1FH: Immediate data or label (even addresses only)
addr16 addr5	0000H to FFFFH: Immediate data or label (only even address for 16-bit data transfer instructions) 0040H to 007FH: Immediate data or label (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label

Remark For the special function register symbols, see **Table 4-1**.

9.1.2 Descriptions of the operation field

A	:	A register (8-bit accumulator)
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
AX	:	AX register pair (16-bit accumulator)
BC	:	BC register pair
DE	:	DE register pair
HL	:	HL register pair
PC	:	Program counter
SP	:	Stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Auxiliary carry flag
Z	:	Zero flag
IE	:	Interrupt request enable flag
NMIS	:	Flag to indicate that a nonmaskable interrupt is being handled
()	:	Contents of a memory location indicated by a parenthesized address or register name
X_H, X_L	:	Upper and lower 8 bits of a 16-bit register
^	:	Logical product (AND)
∨	:	Logical sum (OR)
∀	:	Exclusive OR
—	:	Inverted data
addr16	:	16-bit immediate data or label
jdisp8	:	Signed 8-bit data (displacement value)

9.1.3 Description of the flag operation field

(blank)	:	No change
0	:	To be cleared to 0
1	:	To be set to 1
×	:	To be set or cleared according to the result
R	:	To be restored to the previous value

9.2 Operations

Mnemonic	Operand	Byte	Clock	Operation	Flag		
					Z	AC	CY
MOV	r, #byte	3	6	r ← byte			
	saddr, #byte	3	6	(saddr) ← byte			
	sfr, #byte	3	6	sfr ← byte			
	A, r Note 1	2	4	A ← r			
	r, A Note 1	2	4	r ← A			
	A, saddr	2	4	A ← (saddr)			
	saddr, A	2	4	(saddr) ← A			
	A, sfr	2	4	A ← sfr			
	sfr, A	2	4	sfr ← A			
	A, !addr16	3	8	A ← (addr16)			
	!addr16, A	3	8	(addr16) ← A			
	PSW, #byte	3	6	PSW ← byte	×	×	×
	A, PSW	2	4	A ← PSW			
	PSW, A	2	4	PSW ← A	×	×	×
	A, [DE]	1	6	A ← (DE)			
	[DE], A	1	6	(DE) ← A			
	A, [HL]	1	6	A ← (HL)			
	[HL], A	1	6	(HL) ← A			
	A, [HL + byte]	2	6	A ← (HL + byte)			
	[HL + byte], A	2	6	(HL + byte) ← A			
XCH	A, X	1	4	A ↔ X			
	A, r Note 2	2	6	A ↔ r			
	A, saddr	2	6	A ↔ (saddr)			
	A, sfr	2	6	A ↔ (sfr)			
	A, [DE]	1	8	A ↔ (DE)			
	A, [HL]	1	8	A ↔ (HL)			
	A, [HL + byte]	2	8	A ↔ (HL + byte)			
MOVW	rp, #word	3	6	rp ← word			
	AX, saddrp	2	6	AX ← (saddrp)			
	saddrp, AX	2	8	(saddrp) ← AX			
	AX, rp Note 3	1	4	AX ← rp			
	rp, AX Note 3	1	4	rp ← AX			

Notes **1.** Except when r = A.

2. Except when r = A or X.

3. Only when rp = BC, DE, or HL.

Remark The instruction clock cycle is based on the CPU clock (f_{CPU}), specified in the processor clock controller register (PCC).

Mnemonic	Operand	Byte	Clock	Operation	Flag		
					Z	AC	CY
XCHW	AX, rp[Note]	1	8	AX ↔ rp			
ADD	A, #byte	2	4	A, CY ← A + byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte	×	×	×
	A, r	2	4	A, CY ← A + r	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte)	×	×	×
ADDC	A, #byte	2	4	A, CY ← A + byte + CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) + byte + CY	×	×	×
	A, r	2	4	A, CY ← A + r + CY	×	×	×
	A, saddr	2	4	A, CY ← A + (saddr) + CY	×	×	×
	A, !addr16	3	8	A, CY ← A + (addr16) + CY	×	×	×
	A, [HL]	1	6	A, CY ← A + (HL) + CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A + (HL + byte) + CY	×	×	×
SUB	A, #byte	2	4	A, CY ← A – byte	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte	×	×	×
	A, r	2	4	A, CY ← A – r	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr)	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16)	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL)	×	×	×
	A, [HL + byte]	2	6	A, CY ← A – (HL + byte)	×	×	×
SUBC	A, #byte	2	4	A, CY ← A – byte – CY	×	×	×
	saddr, #byte	3	6	(saddr), CY ← (saddr) – byte – CY	×	×	×
	A, r	2	4	A, CY ← A – r – CY	×	×	×
	A, saddr	2	4	A, CY ← A – (saddr) – CY	×	×	×
	A, !addr16	3	8	A, CY ← A – (addr16) – CY	×	×	×
	A, [HL]	1	6	A, CY ← A – (HL) – CY	×	×	×
	A, [HL + byte]	2	6	A, CY ← A – (HL + byte) – CY	×	×	×
AND	A, #byte	2	4	A ← A ∧ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∧ byte	×		
	A, r	2	4	A ← A ∧ r	×		
	A, saddr	2	4	A ← A ∧ (saddr)	×		
	A, !addr16	3	8	A ← A ∧ (addr16)	×		
	A, [HL]	1	6	A ← A ∧ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∧ (HL + byte)	×		

Note Only when rp = BC, DE, or HL.

Remark The instruction clock cycle is based on the CPU clock (f_{CPU}), specified in the processor clock controller register (PCC).

Mnemonic	Operand	Byte	Clock	Operation	Flag		
					Z	AC	CY
OR	A, #byte	2	4	A ← A ∨ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∨ byte	×		
	A, r	2	4	A ← A ∨ r	×		
	A, saddr	2	4	A ← A ∨ (saddr)	×		
	A, !addr16	3	8	A ← A ∨ (addr16)	×		
	A, [HL]	1	6	A ← A ∨ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∨ (HL + byte)	×		
XOR	A, #byte	2	4	A ← A ∀ byte	×		
	saddr, #byte	3	6	(saddr) ← (saddr) ∀ byte	×		
	A, r	2	4	A ← A ∀ r	×		
	A, saddr	2	4	A ← A ∀ (saddr)	×		
	A, !addr16	3	8	A ← A ∀ (addr16)	×		
	A, [HL]	1	6	A ← A ∀ (HL)	×		
	A, [HL + byte]	2	6	A ← A ∀ (HL + byte)	×		
CMP	A, #byte	2	4	A – byte	×	×	×
	saddr, #byte	3	6	(saddr) – byte	×	×	×
	A, r	2	4	A – r	×	×	×
	A, saddr	2	4	A – (saddr)	×	×	×
	A, !addr16	3	8	A – (addr16)	×	×	×
	A, [HL]	1	6	A – (HL)	×	×	×
	A, [HL + byte]	2	6	A – (HL + byte)	×	×	×
ADDW	AX, #word	3	6	AX, CY ← AX + word	×	×	×
SUBW	AX, #word	3	6	AX, CY ← AX – word	×	×	×
CMPW	AX, #word	3	6	AX – word	×	×	×
INC	r	2	4	r ← r + 1	×	×	
	saddr	2	4	(saddr) ← (saddr) + 1	×	×	
DEC	r	2	4	r ← r – 1	×	×	
	saddr	2	4	(saddr) ← (saddr) – 1	×	×	
INCW	rp	1	4	rp ← rp + 1			
DECW	rp	1	4	rp ← rp – 1			
ROR	A, 1	1	2	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
ROL	A, 1	1	2	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
RORC	A, 1	1	2	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
ROLC	A, 1	1	2	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×

Remark The instruction clock cycle is based on the CPU clock (f_{CPU}), specified in the processor clock controller register (PCC).

Mnemonic	Operand	Byte	Clock	Operation	Flag		
					Z	AC	CY
SET1	saddr. bit	3	6	(saddr. bit) ← 1			
	sfr. bit	3	6	sfr. bit ← 1			
	A. bit	2	4	A. bit ← 1			
	PSW. bit	3	6	PSW. bit ← 1	×	×	×
	[HL]. bit	2	10	(HL). bit ← 1			
CLR1	saddr. bit	3	6	(saddr. bit) ← 0			
	sfr. bit	3	6	sfr. bit ← 0			
	A. bit	2	4	A. bit ← 0			
	PSW. bit	3	6	PSW. bit ← 0	×	×	×
	[HL]. bit	2	10	(HL). bit ← 0			
SET1	CY	1	2	CY ← 1			1
CLR1	CY	1	2	CY ← 0			0
NOT1	CY	1	2	CY ← $\overline{\text{CY}}$			×
CALL	!addr16	3	6	(SP – 1) ← $(\text{PC} + 3)_H$, (SP – 2) ← $(\text{PC} + 3)_L$, PC ← addr16, SP ← SP – 2			
CALLT	[addr5]	1	8	(SP – 1) ← $(\text{PC} + 1)_H$, (SP – 2) ← $(\text{PC} + 1)_L$, PC_H ← (00000000, addr5 + 1), PC_L ← (00000000, addr5), SP ← SP – 2			
RET		1	6	PC_H ← (SP + 1), PC_L ← (SP), SP ← SP + 2			
RETI		1	8	PC_H ← (SP + 1), PC_L ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0	R	R	R
PUSH	PSW	1	2	(SP – 1) ← PSW, SP ← SP – 1			
	rp	1	4	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2			
POP	PSW	1	4	PSW ← (SP), SP ← SP + 1	R	R	R
	rp	1	6	rp_H ← (SP + 1), rp_L ← (SP), SP ← SP + 2			
MOVW	SP, AX	2	8	SP ← AX			
	AX, SP	2	6	AX ← SP			
BR	!addr16	3	6	PC ← addr16			
	$addr16	2	6	PC ← PC + 2 + jdisp8			
	AX	1	6	PC_H ← A, PC_L ← X			

Remark The instruction clock cycle is based on the CPU clock (f_{CPU}), specified in the processor clock controller register (PCC).

Mnemonic	Operand	Byte	Clock	Operation	Flag		
					Z	AC	CY
BC	$addr16	2	6	PC ← PC + 2 + jdisp8 if CY = 1			
BNC	$addr16	2	6	PC ← PC + 2 + jdisp8 if CY = 0			
BZ	$addr16	2	6	PC ← PC + 2 + jdisp8 if Z = 1			
BNZ	$addr16	2	6	PC ← PC + 2 + jdisp8 if Z = 0			
BT	saddr. bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr. bit) = 1			
	sfr. bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr. bit = 1			
	A. bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A. bit = 1			
	PSW. bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW. bit = 1			
BF	saddr. bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if (saddr. bit) = 0			
	sfr. bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if sfr. bit = 0			
	A. bit, $addr16	3	8	PC ← PC + 3 + jdisp8 if A. bit = 0			
	PSW. bit, $addr16	4	10	PC ← PC + 4 + jdisp8 if PSW. bit = 0			
DBNZ	B, $addr16	2	6	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
	C, $addr16	2	6	C ← C – 1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
	saddr, $addr16	3	8	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
NOP		1	2	No Operation			
EI		3	6	IE ← 1 (Enable Interrupt)			
DI		3	6	IE ← 0 (Disable Interrupt)			
HALT		1	2	Set HALT Mode			
STOP		1	2	Set STOP Mode			

Remark The instruction clock cycle is based on the CPU clock (f_{CPU}), specified in the processor clock controller register (PCC).

10. ELECTRICAL CHARACTERISTICS

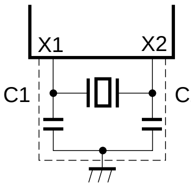
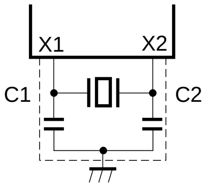
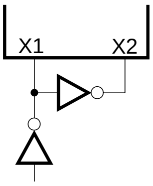
ABSOLUTE MAXIMUM RATINGS (T_A = 25°C)

Parameter	Symbol	Conditions	Rated value	Unit
Supply voltage	V_{DD}		–0.3 to +6.5	V
Input voltage	V_I		–0.3 to V_{DD} + 0.3	V
Output voltage	V_O		–0.3 to V_{DD} + 0.3	V
High-level output current	I_{OH}	Each pin	–10	mA
		Total for all pins	–30	mA
Low-level output current	I_{OL}	Each pin	30	mA
		Total for all pins	160	mA
Operating ambient temperature	T_A		–40 to +85	°C
Storage temperature	T_{stg}		–65 to +150	°C

Caution **Absolute maximum ratings are rated values beyond which physical damage will be caused to the product; if the rated value of any of the parameters in the above table is exceeded, even momentarily, the quality of the product may deteriorate. Always use the product within its rated values.**

Remark The characteristic of a dual-function pin does not differ between the port function and the secondary function, unless otherwise stated.

CHARACTERISTICS OF THE MAIN SYSTEM CLOCK OSCILLATION CIRCUIT
(T_A = –40°C to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillator frequency (f_X)[Note 1]	V_{DD} = oscillation voltage range	1.0		5.0	MHz
		Oscillation settling time[Notes 2, 3]	Reset by $\overline{\text{RESET}}$		$2^{15}/f_X$		ms
			Reset by an interrupt		Note 4		
Crystal		Oscillator frequency (f_X)[Note 1]		1.0		5.0	MHz
		Oscillation settling time[Note 2]	V_{DD} = 4.5 to 5.5 V			10	ms
						30	
External clock		X1 input frequency (f_X)[Note 1]		1.0		5.0	MHz
		X1 input high/low level width (t_{XH}, t_{XL})		85		500	ns

Notes 1. Only the characteristic of the oscillation circuit is indicated. See the description of the AC characteristic for the instruction execution time.

2. Time required for oscillation to settle once a reset sequence ends or STOP mode is deselected

3. Time after V_{DD} reaches MIN. of the oscillation voltage range

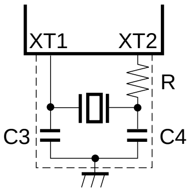
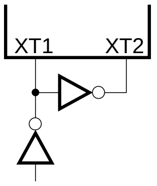
4. Bits 0 to 2 (OSTS0 to OSTS2) of the oscillation settling time selection register can be used to select $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$.

Cautions 1. When using the main system clock oscillation circuit, observe the following conditions for the wiring of that section enclosed in dotted lines in the above diagrams, so as to avoid the influence of the wiring capacitance.

- **Keep the wiring as short as possible.**
- **Do not allow signal wires to cross one another.**
- **Keep the wiring away from wires that carry a high, non-stable current.**
- **Keep the grounding point of the capacitors at the same level as V_{SS}.**
- **Do not connect the grounding point to a grounding wire that carries a high current.**
- **Do not extract a signal from the oscillation circuit.**

2. Before switching from the subsystem clock back to the main system clock, always allow sufficient time for the oscillation to settle by specifying it in the program.

CHARACTERISTICS OF THE SUBSYSTEM CLOCK OSCILLATION CIRCUIT
(T_A = –40°C to +85°C, V_{DD} = 1.8 to 5.5 V)

Resonator	Recommended circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal		Oscillator frequency (f_{XT})[Note 1]		32	32.768	35	kHz
		Oscillation settling time[Note 2]	V_{DD} = 4.5 to 5.5 V		1.2	2	s
						10	
External clock		XT1 input frequency (f_{XT})[Note 1]		32		35	kHz
		XT1 input high/low level width (t_{XTH}, t_{XTL})		14.3		15.6	μs

Notes 1. Only the characteristic of the oscillation circuit is indicated. See the description of the AC characteristic for the instruction execution time.

2. Time required for oscillation to settle after V_{DD} reaches the MIN. value of the oscillation voltage range.

Cautions 1. When using the subsystem clock oscillation circuit, observe the following conditions for the wiring of that section enclosed in dotted lines in the above diagrams, so as to avoid the influence of the wiring capacitance.

- **Keep the wiring as short as possible.**
- **Do not allow signal wires to cross one another.**
- **Keep the wiring away from wires that carry a high, non-stable current.**
- **Keep the grounding point of the capacitors at the same level as V_{SS}.**
- **Do not connect the grounding point to a grounding wire that carries a high current.**
- **Do not extract a signal from the oscillation circuit.**

2. The subsystem clock oscillation circuit is designed to have a low amplification degree so as to maintain a low current drain. Therefore, it is more likely to malfunction as a result of noise than the main system clock oscillation circuit. When using the subsystem clock, therefore, pay particularly careful attention to how it is wired.

DC CHARACTERISTICS (T_A = –40°C to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-level output current	I_{OL}	Each pin				Undefined	mA
		All pins				80	mA
High-level output current	I_{OH}	Each pin				Undefined	mA
		All pins				–15	mA
High-level input voltage	V_{IH1}	P00 to P07, P10 to P17, P30, P31	V_{DD} = 2.7 to 5.5 V	0.7V_{DD}		V_{DD}	V
				0.9V_{DD}		V_{DD}	V
	V_{IH2}	$\overline{\text{RESET}}$, P20 to P27, P40 to P47	V_{DD} = 2.7 to 5.5 V	0.8V_{DD}		V_{DD}	V
				0.9V_{DD}		V_{DD}	V
	V_{IH3}	X1, X2		V_{DD} – 0.1		V_{DD}	V
	V_{IH4}	XT1, XT2		V_{DD} – 0.1		V_{DD}	V
Low-level input voltage	V_{IL1}	P00 to P07, P10 to P17, P30, P31	V_{DD} = 2.7 to 5.5 V	0		0.3V_{DD}	V
				0		0.1V_{DD}	V
	V_{IL2}	$\overline{\text{RESET}}$, P20 to P27, P40 to P47	V_{DD} = 2.7 to 5.5 V	0		0.2V_{DD}	V
				0		0.1V_{DD}	V
	V_{IL3}	X1, X2		0		0.1	V
	V_{IL4}	XT1, XT2		0		0.1	V
High-level output voltage	V_{OH}	V_{DD} = 4.5 to 5.5 V, I_{OH} = –1 mA		V_{DD} – 1.0			V
		I_{OH} = –100 μA		V_{DD} – 0.5			V
Low-level output voltage	V_{OL}	V_{DD} = 4.5 to 5.5 V, I_{OL} = –1 mA				1.0	V
		I_{OL} = 400 μA				0.5	V
High-level input leakage current	I_{LIH1}	V_{IN} = V_{DD}	Pins other than the X1 pin or X2 pin			3	μA
	I_{LIH2}		X1, X2			20	μA
Low-level input leakage current	I_{LIL1}	V_{IN} = 0 V	Pins other than the X1 pin or X2 pin			–3	μA
	I_{LIL2}		X1, X2			–20	μA
High-level output leakage current	I_{LOH}	V_{OUT} = V_{DD}				3	μA
Low-level output leakage current	I_{LOL}	V_{OUT} = 0 V				–3	μA
Software-specified pull-up resistor	R	V_{IN} = 0 V		50	100	200	kΩ

Remark The characteristic of a dual-function pin does not differ between the port function and the secondary function, unless otherwise stated.

DC CHARACTERISTICS (T_A = –40°C to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Power supply current[Note 1]	I_{DD1}	5.0-MHz crystal oscillation operating mode	V_{DD} = 5.0 V ± 10%[Note 2]		1.1	2.0	mA
			V_{DD} = 3.0 V ± 10%[Note 3]		0.3	0.45	mA
			V_{DD} = 2.0 V ± 10%[Note 3]		0.2	0.35	mA
	I_{DD2}	5.0-MHz crystal oscillation HALT mode	V_{DD} = 5.0 V ± 10%[Note 3]		0.6	0.85	mA
			V_{DD} = 3.0 V ± 10%[Note 3]		0.2	0.35	mA
			V_{DD} = 2.0 V ± 10%[Note 3]		0.1	0.18	mA
	I_{DD3}	32.768-kHz crystal oscillation operating mode[Note 4]	V_{DD} = 5.0 V ± 10%		100	200	μA
			V_{DD} = 3.0 V ± 10%		70	140	μA
			V_{DD} = 2.0 V ± 10%		50	100	μA
	I_{DD4}	32.768-kHz crystal oscillation HALT mode[Note 4]	V_{DD} = 5.0 V ± 10%		25	55	μA
			V_{DD} = 3.0 V ± 10%		5	25	μA
			V_{DD} = 2.0 V ± 10%		2.5	12.5	μA
	I_{DD5}	32.768-kHz crystal stop STOP mode	V_{DD} = 5.0 V ± 10%		0.1	30	μA
			V_{DD} = 3.0 V ± 10%		0.05	10	μA
			V_{DD} = 2.0 V ± 10%		0.05	10	μA

Notes 1. The power supply current does not include the port current (including the current flowing through the on-chip pull-up resistor).

2. During high-speed mode operation (when the processor clock control register (PCC) is cleared to 00H)

3. During low-speed mode operation (when PCC is set to 02H)

4. When the main system clock is not running

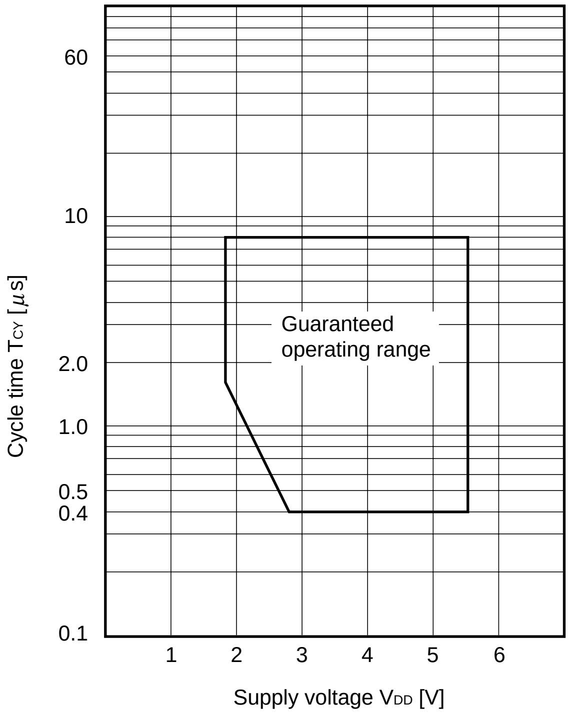
Remark The characteristic of a dual-function pin does not differ between the port function and the secondary function, unless otherwise stated.

AC CHARACTERISTICS

(1) Basic operations (T_A = –40°C to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Cycle time (minimum instruction execution time)	T_{CY}	Operation based on the main system clock	V_{DD} = 2.7 to 5.5 V	0.4		8	μs
				1.6		8	μs
		Operation based on the subsystem clock			122		μs
TI80 input high/low level width	t_{TIH}, t_{TIL}	V_{DD} = 2.7 to 5.5 V		0.1			μs
				1.8			μs
TI80 input frequency	f_{TI}	V_{DD} = 2.7 to 5.5 V		0		4	MHz
				0		275	kHz
Interrupt input high/low level width	t_{INTH}, t_{INTL}	INTP0 to INTP2	V_{DD} = 2.7 to 5.5 V	10			μs
				20			μs
$\overline{\text{RESET}}$ low level width	t_{RST}	V_{DD} = 2.7 to 5.5 V		10			μs
				20			μs

T_{CY} vs V_{DD} (main system clock)



(2) SERIAL INTERFACE (T_A = –40°C to +85°C, V_{DD} = 1.8 to 5.5 V)

(a) Three-wire serial I/O mode ($\overline{\text{SCK}}$...Internal clock output)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK}}$ cycle time	t_{KCY1}	V_{DD} = 2.7 to 5.5 V		800			ns
				3,200			ns
$\overline{\text{SCK}}$ high/low level width	t_{KH1}, t_{KL1}	V_{DD} = 2.7 to 5.5 V		t_{KCY1}/2–50			ns
				t_{KCY1}/2–150			ns
SI setup time (for $\overline{\text{SCK}}$ ↑)	t_{SIK1}	V_{DD} = 2.7 to 5.5 V		150			ns
				500			ns
SI hold time (for $\overline{\text{SCK}}$ ↑)	t_{KSI1}	V_{DD} = 2.7 to 5.5 V		400			ns
				600			ns
Delay from $\overline{\text{SCK}}$ ↓ to SO output	t_{KSO1}	R = 1 kΩ, C = 100 pF Note	V_{DD} = 2.7 to 5.5 V	0		250	ns
				0		1,000	ns

Note R and C are the resistance and capacitance of the SO output line, respectively.

(b) Three-wire serial I/O mode ($\overline{\text{SCK}}$...External clock output)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK}}$ cycle time	t_{KCY2}	V_{DD} = 2.7 to 5.5 V		800			ns
				3,200			ns
$\overline{\text{SCK}}$ high/low level width	t_{KH2}, t_{KL2}	V_{DD} = 2.7 to 5.5 V		400			ns
				1,600			ns
SI setup time (for $\overline{\text{SCK}}$ ↑)	t_{SIK2}	V_{DD} = 2.7 to 5.5 V		100			ns
				150			ns
SI hold time (for $\overline{\text{SCK}}$ ↑)	t_{KSI2}	V_{DD} = 2.7 to 5.5 V		400			ns
				600			ns
Delay from $\overline{\text{SCK}}$ ↓ to SO output	t_{KSO2}	R = 1 kΩ, C = 100 pF Note	V_{DD} = 2.7 to 5.5 V	0		300	ns
				0		1,000	ns

Note R and C are the resistance and capacitance of the SO output line, respectively.

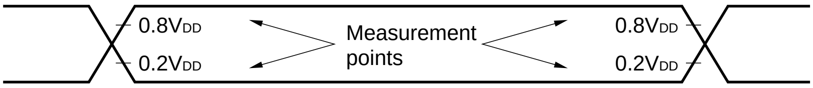
(c) UART mode (internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		V_{DD} = 2.7 to 5.5 V			78,125	bps
					19,531	bps

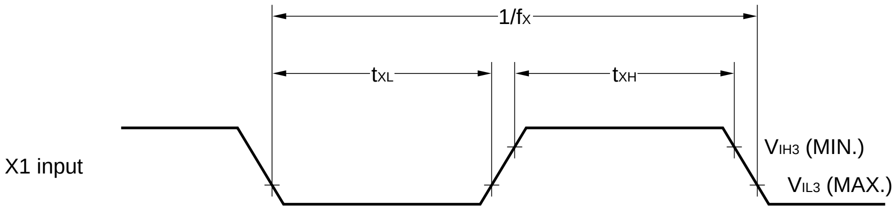
(d) UART mode (external clock input)

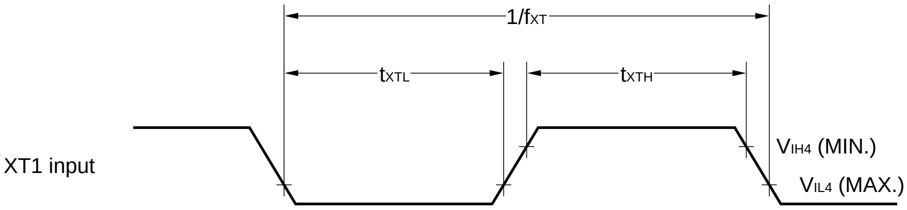
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ASCK cycle time	t_{KCY3}	V_{DD} = 2.7 to 5.5 V	800			ns
			3,200			ns
ASCK high/low level width	t_{KH3}, t_{KL3}	V_{DD} = 2.7 to 5.5 V	400			ns
			1,600			ns
Transfer rate		V_{DD} = 2.7 to 5.5 V			39,063	bps
					9,766	bps
ASCK rising time, falling time	t_R, t_F				1	μs

AC TIMING MEASUREMENT POINTS (except the X1 and XT1 inputs)

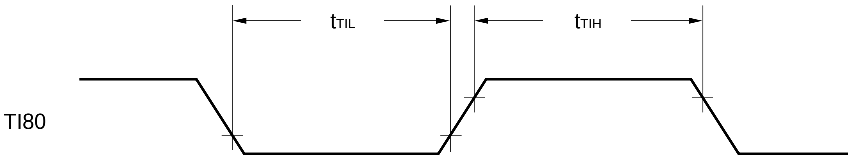


CLOCK TIMING

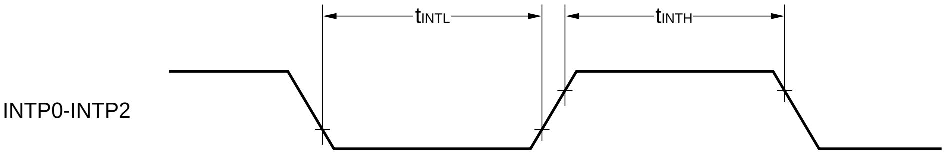




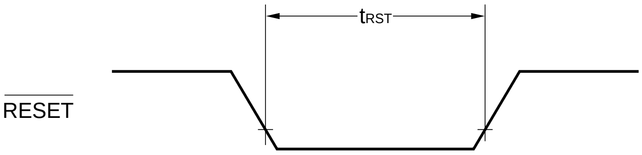
TI TIMING



INTERRUPT INPUT TIMING

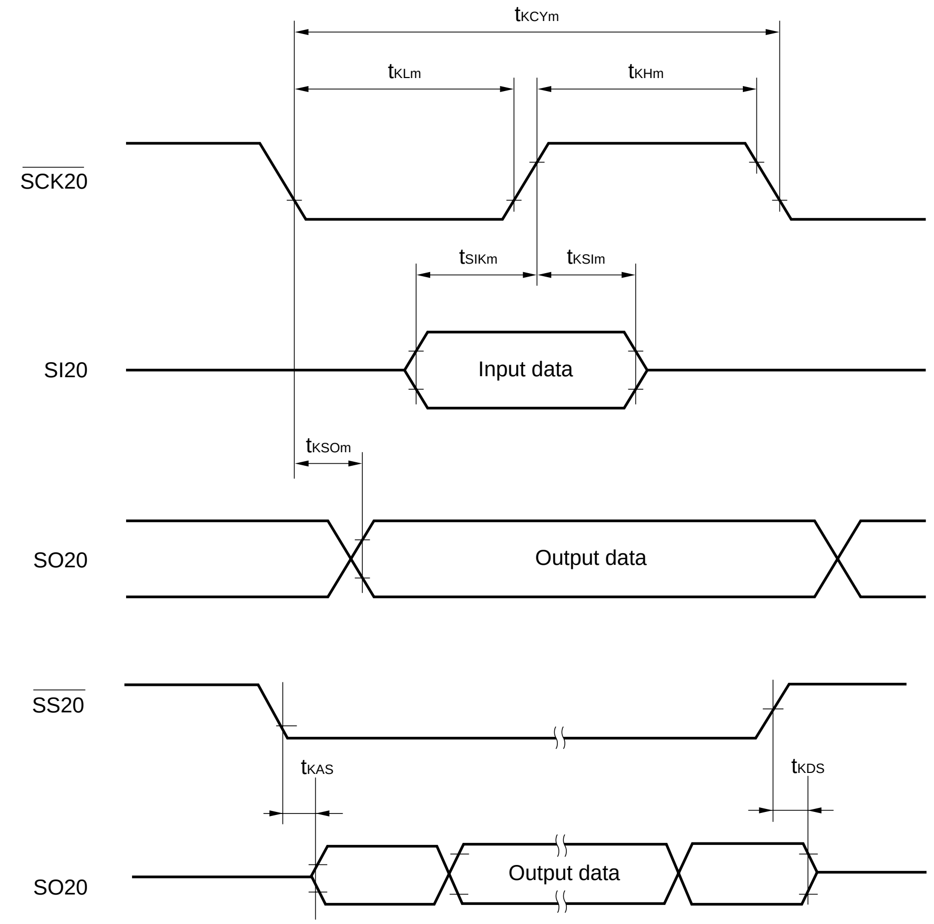


$\overline{\text{RESET}}$ INPUT TIMING



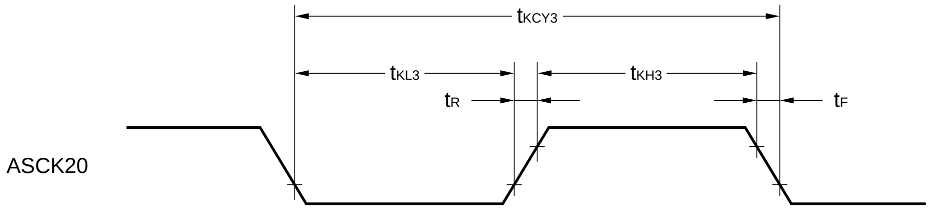
SERIAL TRANSFER TIMING

Three-Wire Serial I/O Mode:



m = 1, 2

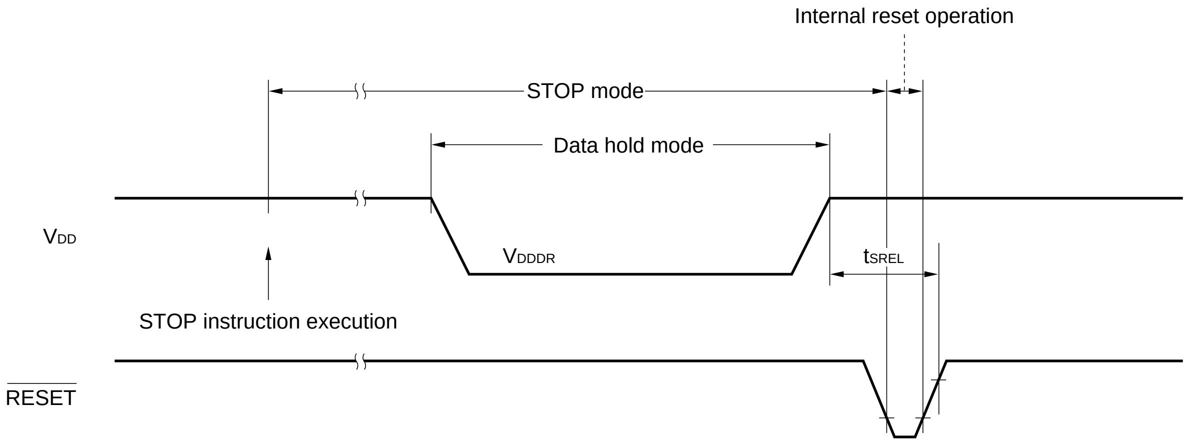
UART Mode (External Clock Input):



DATA MEMORY STOP MODE LOW POWER SUPPLY VOLTAGE DATA HOLD CHARACTERISTICS (T_A = –40°C to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data hold supply voltage	V_{DDDR}		1.8		5.5	V
Release signal set time	t_{SREL}		0			μs

DATA HOLD TIMING (STOP mode release by $\overline{\text{RESET}}$)



11. PACKAGE DRAWINGS

44-PIN PLASTIC QFP (10 × 10)

To be decided

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for developing systems using the μPD789046.

LANGUAGE PROCESSING SOFTWARE

RA78K0S[Notes 1, 2, 3]	Assembler package common to the 78K/0S series
CC78K0S[Notes 1, 2, 3]	C compiler package common to the 78K/0S series
DF789046[Notes 1, 2, 3, 5]	Device file for the μPD789046 sub-series
CC78K0S-L[Notes 1, 2, 3, 5]	C compiler library source file common to the 78K/0S series

FLASH MEMORY WRITE TOOLS

Flashpro II[Note 4]	Dedicated flash writer (formerly, Flashpro)
FA-44GB[Note 4]	Flash memory write adapter

DEBUGGING TOOLS

IE-78K0S-NS In-circuit emulator	This in-circuit emulator is used to debug hardware or software when application systems which use the 78K/0S series are developed. The IE-78K0S-NS supports the integrated debugger (ID78K0S-NS). The IE-78K0S-NS is used in combination with an interface adapter for connection to an AC adapter, emulation probe, or host machine.
IE-70000-MC-PS-B AC adapter	This adapter is used to supply power from a 100-VAC outlet.
IE-70000-98-IF-C Interface adapter	This adapter is required when a PC-9800 series computer (other than a notebook type) is used as the host machine for the IE-78K0S-NS.
IE-70000-CD-IF PC card/interface	These PC card and interface cable are required when a PC-9800 series computer is used as the host machine for the IE-78K0S-NS.
IE-70000-PC-IF-C Interface adapter	This adapter is required when an IBM PC/AT™ or compatible is used as the host machine for the IE-78K0S-NS.
IE-789046-NS-EM1[Note 5] Emulation board	This board is used to emulate the peripheral hardware specific to the device. The IE-789046-NS-EM1 is used in combination with the in-circuit emulator.
NP-44GB[Note 4] Emulation probe	This probe is used to connect an in-circuit emulator to the target system. The probe is dedicated to the 44-pin plastic QFP.
SM78K0S[Notes 1, 2]	System simulator common to all 78K/0S series units
DF789046[Notes 1, 2, 5]	Device file for the μPD789046 sub-series

Notes 1. Based on the PC-9800 series (MS-DOS™ + Windows™)

2. Based on the IBM PC/AT and compatibles (Japanese/English Windows)

3. Based on the HP9000 series 700™ (HP-UX™), SPARCstation™ (SunOS™), and NEWS™ (NEWS-OS™)

4. Product manufactured by and available from Naito Densei Machida Mfg. Co., Ltd. (044-822-3813).

5. Under development

Remark The RA78K0S, CC78K0S, and SM78K0S can be used in combination with the DF789046.

REAL-TIME OS

MX78K0S^Notes 1, 2^	OS for the 78K/0S series

Notes **1.** Based on the PC-9800 series (MS-DOS^{TM} + Windows^{TM})

2. Based on the IBM PC/AT and compatibles (Japanese/English Windows)

Remark The RA78K0S, CC78K0S, and SM78K0S can be used in combination with the DF789046.

APPENDIX B RELATED DOCUMENTS

DOCUMENTS RELATED TO DEVICES

Document name	Document No.	
	Japanese	English
μPD789046 Preliminary Product Information	U13380J	This manual
μPD78F9046 Preliminary Product Information	To be created	To be created
μPD789046 Sub-Series User's Manual	To be created	To be created
78K/0S Series User's Manual, Instruction	U11047J	U11047E
78K/0S Series Instruction Summary Sheet	To be created	–
78K/0S Series Instruction Set	To be created	–

DOCUMENTS RELATED TO DEVELOPMENT TOOLS (USER'S MANUAL)

Document name		Document No.	
		Japanese	English
RA78K0S Assembler Package	Operation	U11622J	U11622E
	Assembly Language	U11599J	U11599E
	Structured Assembly Language	U11623J	U11623E
CC78K/0S C Compiler	Operation	U11816J	U11816E
	Language	U11817J	U11817E
SM78K0S System Simulator Windows Base	Reference	U11489J	U11489E
SM78K Series System Simulator	External Parts User Open Interface Specifications	U10092J	U10092E
ID78K0S-NS Integrated Debugger Windows Base	Reference	U12901J	To be created

DOCUMENTS RELATED TO SOFTWARE TO BE INCORPORATED INTO THE PRODUCT (USER'S MANUAL)

Document name		Document No.	
		Japanese	English
OS for 78K/0S Series MX78K0S	Fundamental	U12938J	U12938E

Caution The above documents may be revised without notice. Use the latest versions when you design application systems.

OTHER DOCUMENTS

Document name	Document No.	
	Japanese	English
IC PACKAGE MANUAL	C10943X	
Semiconductor Device Mounting Technology Manual	C10535J	C10535E
Quality Grades on NEC Semiconductor Device	C11531J	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983J	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892J	C11892E
Semiconductor Device Quality Control/Reliability Handbook	C12769J	–
Guide for Products Related to Microcomputer: Other Companies	U11416J	–

Caution The above documents may be revised without notice. Use the latest versions when you design application systems.

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

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

EEPROM is a trademark of NEC Corporation.

MS-DOS and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

PC/AT is a trademark of IBM Corporation.

HP9000 series 700 and HP-UX are trademarks of Hewlett-Packard Company.

SPARCstation is a trademark of SPARC International, Inc.

SunOS is a trademark of Sun Microsystems, Inc.

NEWS and NEWS-OS are trademarks of SONY Corporation.

Regional Information

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

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

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

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

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

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

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

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

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

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

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

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

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

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 65-253-8311
Fax: 65-250-3583

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

NEC do Brasil S.A.
Cumbica-Guarulhos-SP, Brasil
Tel: 011-6465-6810
Fax: 011-6465-6829

J98. 2

[MEMO]

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

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

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

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

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

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

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

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

Anti-radioactive design is not implemented in this product.

M4 96.5