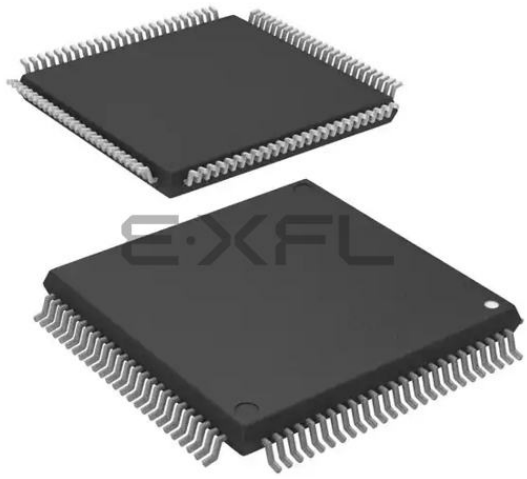
Renesas Electronics America Inc - UPD78F1166AGC-UEU-AX Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	78K/0R
Core Size	16-Bit
Speed	20MHz
Connectivity	3-Wire SIO, I²C, LINbus, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	83
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 16x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f1166agc-ueu-ax

Table 3-5. SFR List (5/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFF0H	Multiplication input data register A	MULA		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiplication input data register B	MULB		R/W	–	–	√	0000H
FFFF3H								
FFFF4H	Higher multiplication result storage register	MULOH		R	–	–	√	0000H
FFFF5H								
FFFF6H	Lower multiplication result storage register	MULOL		R	–	–	√	0000H
FFFF7H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H
FFFFFH	Memory extension mode control register	MEM		R/W	√	√	–	00H

Remark For extended SFRs (2nd SFRs), see **Table 3-6 Extended SFR (2nd SFR) List**.

Table 3-6. Extended SFR (2nd SFR) List (3/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	–	√	√	0000H
F014DH		–			–	–		
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	–	√	√	0000H
F014FH		–			–	–		
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F0153H								
F0154H	Serial mode register 12	SMR12		R/W	–	–	√	0020H
F0155H								
F0156H	Serial mode register 13	SMR13		R/W	–	–	√	0020H
F0157H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H
F015BH								
F015CH	Serial communication operation setting register 12	SCR12		R/W	–	–	√	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	–	–	√	0087H
F015FH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		–			–	–		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		–			–	–		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		–			–	–		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	–	√	√	0000H
F0167H		–			–	–		
F0168H	Serial output register 1	SO1		R/W	–	–	√	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1L	R/W	√	√	√	0000H
F016BH		–			–	–		
F0174H	Serial output level register 1	SOL1L	SOL1L	R/W	–	√	√	0000H
F0175H		–			–	–		
F0180H	Timer counter register 00	TCR00		R	–	–	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	–	–	√	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	–	–	√	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	–	–	√	FFFFH
F0187H								

11.5.2 Registers used by temperature sensors

The following four types of registers are used when using a temperature sensor.

- Peripheral enable register 0 (PER0)
- A/D converter mode register (ADM)
- Analog input channel specification register (ADS)
- 10-bit A/D conversion result register (ADCR)

Caution Setting of the A/D port configuration register (ADPC), port mode register 2 (PM2) and port register 2 (P2) is not required when using the temperature sensor. There is no problem if the pin function is set as digital I/O.

(1) Peripheral enable register 0 (PER0)

Use the PER0 register in the same manner as during A/D converter basic operation (see **11.3 (1) Peripheral enable register 0 (PER0)**).

(2) A/D converter mode register (ADM)

Use the ADM register in the same manner as during A/D converter basic operation (see **11.3 (2) A/D converter mode register (ADM)**).

However, selection of the A/D conversion time when a temperature sensor is used varies as shown in Table 11-5.

Table 11-5. Selection of A/D Conversion Time When Using Temperature Sensor

(1) 2.7 V ≤ AV_{REF0} ≤ 5.5 V

A/D Converter Mode Register (ADM)					Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 8 MHz	f_{CLK} = 20 MHz	
0	0	0	0	1	480/f_{CLK}	Setting prohibited	60.0 μs	24.0 μs	f_{CLK}/12
0	0	1	0	1	320/f_{CLK}		40.0 μs	Setting prohibited	f_{CLK}/8
0	1	0	0	1	240/f_{CLK}		30.0 μs		f_{CLK}/6
0	1	1	0	1	160/f_{CLK}		Setting prohibited		f_{CLK}/4
1	0	0	0	1	120/f_{CLK}	60.0 μs			f_{CLK}/3
1	0	1	0	1	80/f_{CLK}	40.0 μs			f_{CLK}/2
1	1	1	0	1	40/f_{CLK}	Setting prohibited			f_{CLK}
Other than above					Setting prohibited				

Cautions 1. Set the conversion times so as to satisfy the following condition.
f_{AD} = 0.6 to 1.8 MHz

2. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion (ADCS = 0) beforehand.

3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

(9) Serial channel start register m (SSm)

SSm is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1. Because SSmn is a trigger bit, it is cleared immediately when SEmn = 1.

SSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SSm can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears this register to 0000H.

Figure 13-12. Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H (SS0), F0162H, F0163H (SS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3	SSm2	SSm1	SSm0

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets SEmn to 1 and enters the communication wait status (if a communication operation is already under execution, the operation is stopped and the start condition is awaited).

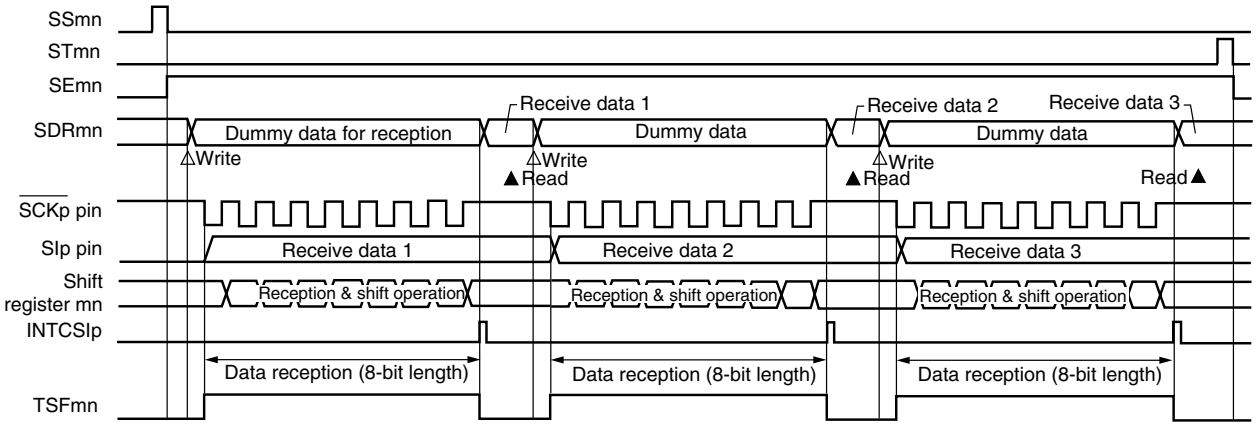
Caution Be sure to clear bits 15 to 4 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

2. When the SSm register is read, 0000H is always read.

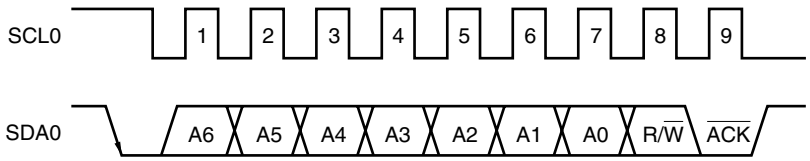
(3) Processing flow (in single-reception mode)

Figure 13-36. Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), p: CSI number (p = 00, 01, 10, 20)

Figure 14-17. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated, regardless of the value of ACKE0. When an address other than that of the local address is received, $\overline{\text{ACK}}$ is not generated (NACK).

When an extension code is received, $\overline{\text{ACK}}$ is generated if ACKE0 is set to 1 in advance.

How $\overline{\text{ACK}}$ is generated when data is received differs as follows depending on the setting of the wait timing.

- When 8-clock wait state is selected (bit 3 (WTIM0) of IICC0 register = 0):
  By setting ACKE0 to 1 before releasing the wait state, $\overline{\text{ACK}}$ is generated at the falling edge of the eighth clock of the SCL0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICC0 register = 1):
  $\overline{\text{ACK}}$ is generated by setting ACKE0 to 1 in advance.

14.5.7 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0 pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 14-19. Wait (1/2)

(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0 = 1)

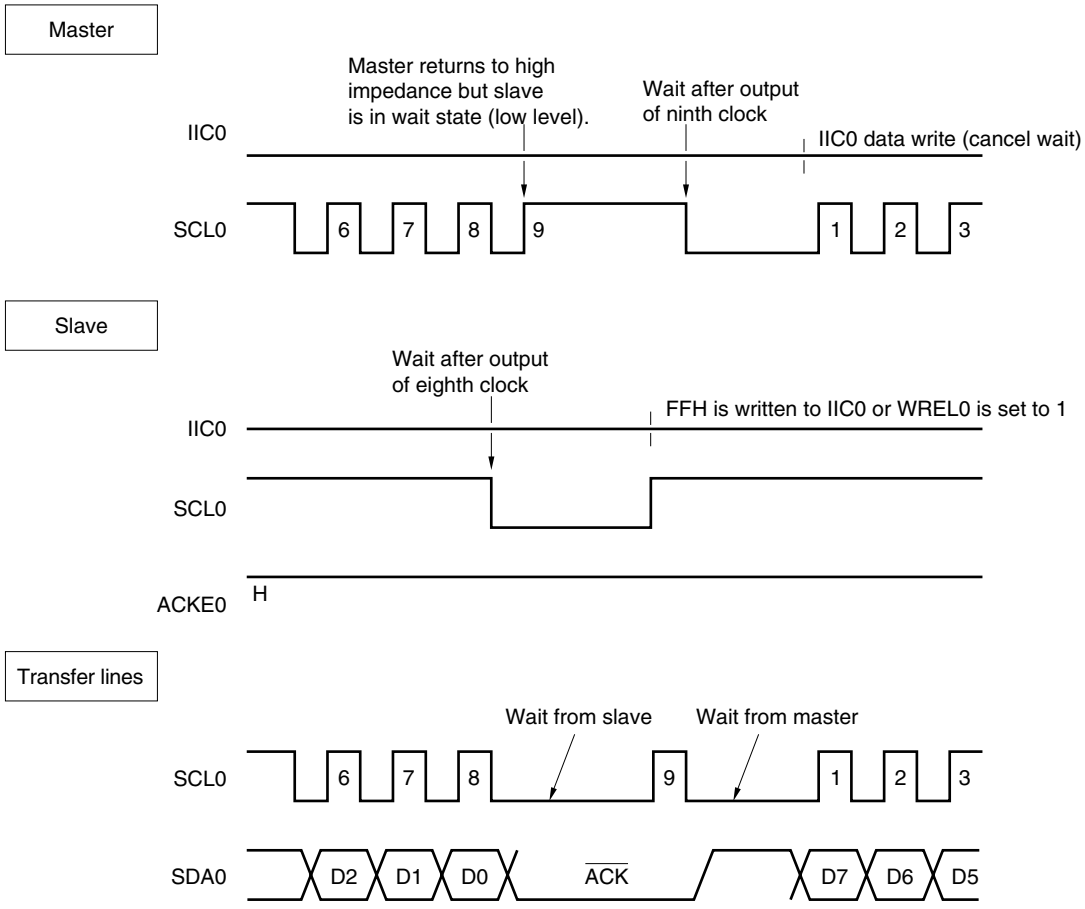


Figure 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF2H	0	0	0	0	0	0	0	PIF11

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. Be sure to clear bits 1 to 7 of IF2H to 0.

2. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

3. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H can be set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, MK1L and MK1H, and MK2L and MK2H are combined to form 16-bit registers MK0, MK1, and MK2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Table 20-2. Hardware Statuses After Reset Acknowledgment (2/3)

Hardware		Status After Reset Acknowledgment[Note 1]
Real-time counter	Subcount register (RSUBC)	0000H
	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Day count register (DAY)	01H
	Week count register (WEEK)	00H
	Month count register (MONTH)	01H
	Year count register (YEAR)	00H
	Watch error correction register (SUBCUD)	00H
	Alarm minute register (ALARMWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register ALARMWW)	00H
	Real-time counter control register 0 (RTCC0)	00H
	Real-time counter control register 1 (RTCC1)	00H
	Real-time counter control register 2 (RTCC2)	00H
Clock output/buzzer output controller	Clock output select registers 0, 1 (CKS0, CKS1)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 2]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	10H
D/A converter	8-bit D/A conversion value setting registers 0, 1 (DACS0, DACS1)	00H
	D/A converter mode register (DAM)	00H
Serial array unit (SAU)	Serial data registers 00, 01, 02, 03, 10, 11, 12, 13 (SDR00, SDR01, SDR02, SDR03, SDR10, SDR11, SDR12, SDR13)	0000H
	Serial status registers 00, 01, 02, 03, 10, 11, 12, 13 (SSR00, SSR01, SSR02, SSR03, SSR10, SSR11, SSR12, SSR13)	0000H
	Serial flag clear trigger registers 00, 01, 02, 03, 10, 11, 12, 13 (SIR00, SIR01, SIR02, SIR03, SIR10, SIR11, SIR12, SIR13)	0000H
	Serial mode registers 00, 01, 02, 03, 10, 11, 12, 13 (SMR00, SMR01, SMR02, SMR03, SMR10, SMR11, SMR12, SMR13)	0020H
	Serial communication operation setting registers 00, 01, 02, 03, 10, 11, 12, 13 (SCR00, SCR01, SCR02, SCR03, SCR10, SCR11, SCR12, SCR13)	0087H
	Serial channel enable status registers 0, 1 (SE0, SE1)	0000H
	Serial channel start registers 0, 1 (SS0, SS1)	0000H
	Serial channel stop registers 0, 1 (ST0, ST1)	0000H
	Serial clock select registers 0, 1 (SPS0, SPS1)	0000H
	Serial output registers 0, 1 (SO0, SO1)	0F0FH
	Serial output registers 0, 1 (SO0, SO1)	0000H
	Serial output level registers 0, 1 (SOL0, SOL1)	0000H
	Input switch control register (ISC)	00H

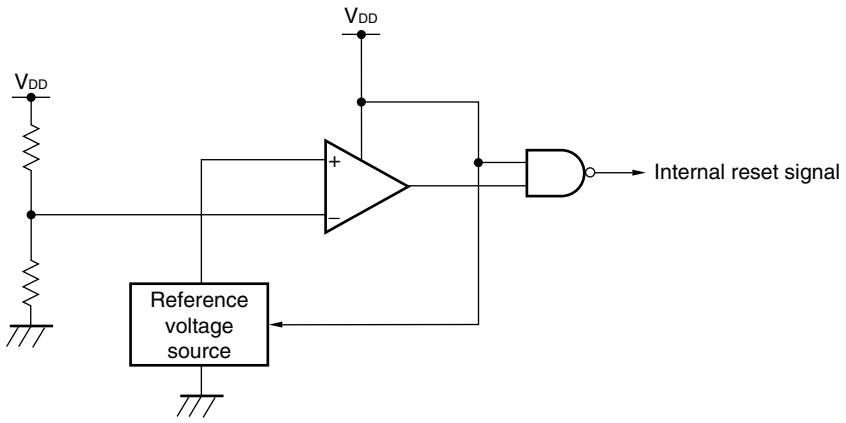
Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. The reset value of WDTE is determined by the option byte setting.

21.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 21-1.

Figure 21-1. Block Diagram of Power-on-Clear Circuit



21.3 Operation of Power-on-Clear Circuit

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{POC} = 1.59 V ±0.09 V), the reset status is released.

  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V.

- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.09 V) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

(2) When detecting level of input voltage from external input pin (EXLVI)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
    Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for the following periods of time (Total 410 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
    - Detection delay time (200 μs (MAX.))
  - <5> Confirm that "input voltage from external input pin (EXLVI) $\geq$ detection voltage (V_{EXLVI} = 1.21 V (TYP.))" when detecting the falling edge of EXLVI, or "input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI} = 1.21 V (TYP.))" when detecting the rising edge of EXLVI, at bit 0 (LVIF) of LVIM.
  - <6> Clear the interrupt request flag of LVI (LVIIF) to 0.
  - <7> Release the interrupt mask flag of LVI (LVIMK).
  - <8> Execute the EI instruction (when vector interrupts are used).

  Figure 22-10 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

  Caution The input voltage from the external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Table 28-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, ES:[HL + byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL + byte], AX	3	2	–	((ES, HL) + byte) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
	XCHW	AX, rp Note 3	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
2. When the program memory area is accessed.
3. Except rp = AX
4. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).
2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r – CY	×	×	×
		r, A	2	1	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16) – CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL) – CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte) – CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B) – CY	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES:HL) + C) – CY	×	×	×
	AND	A, #byte	2	1	–	A ← A ∧ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	–	A ← A ∧ r	×		
		r, A	2	1	–	r ← r ∧ A	×		
		A, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		

Notes 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

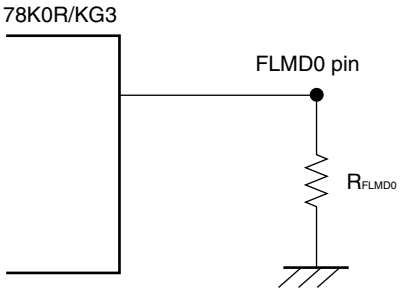
3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

DC Characteristics (7/16)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 1.8 V ≤ AV_{REF0} ≤ V_{DD}, 1.8 V ≤ AV_{REF1} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
On-chip pull-up resistance	R_U	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P120, P131, P140 to P145	V_I = V_{SS}, In input port	10	20	100	kΩ
FLMD0 pin external pull-down resistance[Note]	R_{FLMD0}	When enabling the self-programming mode setting with software		100			kΩ

Note It is recommended to leave the FLMD0 pin open. If the pin is required to be pulled down externally, set R_{FLMD0} to 100 kΩ or more.



Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(3) Serial interface: Serial array unit (14/18)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

<R> **(g) During communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{SCKp}$... external clock input)**

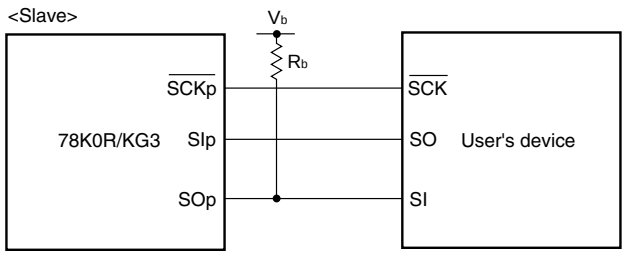
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	13.6 MHz < f_{MCK}	10/f_{MCK}			ns
			6.8 MHz < f_{MCK} ≤ 13.6 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 6.8 MHz	6/f_{MCK}			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	18.5 MHz < f_{MCK}	16/f_{MCK}			ns
			14.8 MHz < f_{MCK} ≤ 18.5 MHz	14/f_{MCK}			ns
			11.1 MHz < f_{MCK} ≤ 14.8 MHz	12/f_{MCK}			ns
			7.4 MHz < f_{MCK} ≤ 11.1 MHz	10/f_{MCK}			ns
			3.7 MHz < f_{MCK} ≤ 7.4 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 3.7 MHz	6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 – 35			ns
SIp setup time (to $\overline{SCKp}$↑)[Note 1]	t_{SIK2}			90			ns
SIp hold time (from $\overline{SCKp}$↑)[Note 2]	t_{KSI2}			1/f_{MCK} + 50			ns
Delay time from $\overline{SCKp}$↓ to SOp output[Note 3]	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ				2/f_{MCK} + 120	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ				2/f_{MCK} + 230	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (during communication at different potential)



(Caution and Remarks are given on the next page.)

(3) Serial interface: Serial array unit (11/18)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

<R> **(f) During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCKp}$... internal clock output) (1/2)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	400 Note 1			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	800 Note 1			ns
$\overline{SCKp}$ high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 75			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 170			ns
$\overline{SCKp}$ low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 35			ns
SIp setup time (to $\overline{SCKp}$↑) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	150			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	275			ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	30			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 2	t_{KSO1}	4.0 V ≤ V_{DD} < 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			120	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			215	ns

Notes 1. The value must also be 4/f_{CLK} or more.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Caution Select the TTL input buffer for SIp and the N-ch open-drain output (V_{DD} tolerance) mode for SOp and $\overline{SCKp}$ by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 01, 10, 20), g: PIM and POM number (g = 0, 4, 14)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2)

3. R_b[Ω]: Communication line ($\overline{SCKp}$, SOp) pull-up resistance,
C_b[F]: Communication line ($\overline{SCKp}$, SOp) load capacitance, V_b[V]: Communication line voltage

4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

5. CSI00 cannot communicate at different potential. Use CSI01, CSI10, and CSI20 for communication at different potential.

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 7	Soft	Timer array unit	TOM0: Timer output mode register 0	Be sure to clear bits 15 to 8 to “0”.	p.253 □
			ISC: Input switch control register	Be sure to clear bits 7 to 2 to “0”.	p.254 □
			Channel output (TO0n pin) operation	(1) Changing values set in registers TO0, TOE0, TOL0, and TOM0 during timer operation Since the timer operations (operations of TCR0n and TDR0n) are independent of the TO0n output circuit and changing the values set in TO0, TOE0, TOL0, and TOM0 does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TO0n pin by timer operation, however, set TO0, TOE0, TOL0, and TOM0 to the values stated in the register setting example of each operation. When the values set in TOE0, TOL0, and TOM0 (except for TO0) are changed close to the timer interrupt (INTTM0n), the waveform output to the TO0n pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTM0n) signal generation timing.	p.258 □
				(2) Default level of TO0n pin and output level after timer operation start The following figure shows the TO0n pin output level transition when writing has been done in the state of TOE0n = 0 before port output is enabled and TOE0n = 1 is set after changing the default level. (a) When operation starts with TOM0n = 0 setting (toggle output) The setting of TOL0n is invalid when TOM0n = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TO0n pin is reversed. (b) When operation starts with TOM0n = 1 setting (combination operation mode (PWM output)) When TOM0n = 1, the active level is determined by TOL0n setting.	pp.259, 260 □
				(3) Operation of TO0n pin in combination operation mode (TOM0n = 1) (a) When TOL0n setting has been changed during timer operation When the TOL0n setting has been changed during timer operation, the setting becomes valid at the generation timing of TO0n change condition. Rewriting TOL0n does not change the output level of TO0n. The following figure shows the operation when the value of TOL0n has been changed during timer operation (TOM0n = 1). (b) Set/reset timing To realize 0%/100% output at PWM output, the TO0n pin/TO0n set timing at master channel timer interrupt (INTTM0n) generation is delayed by 1 count clock by the slave channel. If the set condition and reset condition are generated at the same time, a higher priority is given to the latter. Figure 7-29 shows the set/reset operating statuses where the master/slave channels are set as follows.	pp.260, 261 □
			Collective manipulation of TO0n bits	When TOE0n = 1, even if the output by timer interrupt of each timer (INTTM0n) contends with writing to TO0n, output is normally done to TO0n pin.	p.263 □

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 29	Hard	Electrical specifications (standard products)	Absolute maximum ratings	Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.	pp.763, 764 ☐
			X1 oscillator characteristics	When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. • Keep the wiring length as short as possible. • Do not cross the wiring with the other signal lines. • Do not route the wiring near a signal line through which a high fluctuating current flows. • Always make the ground point of the oscillator capacitor the same potential as V_{SS}. • Do not ground the capacitor to a ground pattern through which a high current flows. • Do not fetch signals from the oscillator.	p.765 ☐
				Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.	p.765 ☐
			XT1 oscillator characteristics	When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. • Keep the wiring length as short as possible. • Do not cross the wiring with the other signal lines. • Do not route the wiring near a signal line through which a high fluctuating current flows. • Always make the ground point of the oscillator capacitor the same potential as V_{SS}. • Do not ground the capacitor to a ground pattern through which a high current flows. • Do not fetch signals from the oscillator.	p.767 ☐
				The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.	p.767 ☐
			Recommended oscillator constants	The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit. When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KG3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.	pp.768 to 771 ☐
				The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit. When doing so, check the conditions for using the RMC register, and whether to enter or exit the STOP mode. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KG3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.	p.772 ☐

(4/5)

Page	Description	Classification
CHAPTER 16 DMA CONTROLLER (continuation)		
p.627	Addition of **Note** to **Figure 16-12. Example of Setting for Holding DMA Transfer Pending by DWAITn**	(c)
pp.628, 629	Change of **16.5.7 Forced termination by software**	(c)
p.630	Change of **(1) Priority of DMA** in **16.6 Cautions on Using DMA Controller**	(c)
p.631	Change of **(2) DMA response time** in **16.6 Cautions on Using DMA Controller**	(c)
p.632	Change of description in **(4) DMA pending instruction** in **16.6 Cautions on Using DMA Controller**	(c)
CHAPTER 17 INTERRUPT FUNCTIONS		
p.636	Change of **(B) External maskable interrupt (INTPn)** in **Figure 17-1. Basic Configuration of Interrupt Function**	(c)
p.637	Addition of **(C) External maskable interrupt (INTKR)** to **Figure 17-1. Basic Configuration of Interrupt Function**	(c)
p.654	Addition of **instruction** to **17.4.4 Interrupt request hold**	(c)
CHAPTER 18 KEY INTERRUPT FUNCTION		
p.655	Change of **Table 18-2. Configuration of Key Interrupt**	(c)
p.656	Addition of **18.3 (2) Port mode register 7 (PM7)**	(c)
CHAPTER 27 BCD CORRECTION CIRCUIT		
p.740	Change of **27.3 BCD Correction Circuit Operation**	(a)
CHAPTER 28 INSTRUCTION SET		
p.745	Change of description in **28.1.4 PREFIX instruction**	(c)
p.761	Change of **Clocks** of **BT Mnemonic** in **Table 28-5. Operation List (16/17)**	(c)
p.762	Change of **Clocks** of **BF Mnemonic** in **Table 28-5. Operation List (17/17)**	(c)
CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)		
p.765	Deletion of **Remark** in **X1 Oscillator Characteristics**	(a)
p.767	Deletion of **Remark** in **XT1 Oscillator Characteristics**	(a)
pp.768, 770, 772	Change of **Caution** in **Recommended Oscillator Constants**	(c)
pp.769, 771	Addition of KYOCERA KINSEKI Co., Ltd. to **Recommended Oscillator Constants**	(c)
pp.780 to 788	Addition of **Remark** to Supply current in **DC Characteristics**	(c)
p.801	Change of **(b) During communication at same potential (CSI mode) (master mode, $\overline{\text{SCK}}$p... internal clock output)** in **Serial interface: Serial array unit**	(b)
p.802	Change of **(c) During communication at same potential (CSI mode) (slave mode, $\overline{\text{SCK}}$p... external clock input)** in **Serial interface: Serial array unit**	(b)
p.804	Addition of **Note** to **(d) During communication at same potential (simplified I^2C mode)** in **Serial interface: Serial array unit**	(c)
pp.810, 811	Change of **(f) During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{\text{SCK}}$p... internal clock output)** in **Serial interface: Serial array unit**	(b)
p.813	Change of **(g) During communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{\text{SCK}}$p... external clock input)** in **Serial interface: Serial array unit**	(b)
p.816	Addition of **Note** to **(h) During communication at different potential (2.5 V, 3 V) (simplified I^2C mode)** in **Serial interface: Serial array unit**	(b)

Remark "Classification" in the above table classifies revisions as follows.

(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(6/24)

Edition	Description	Chapter
3rd edition	Addition of **Cautions 3** to **Figure 6-3 Format of Clock Operation Status Control Register (CSC)**	CHAPTER 6 CLOCK GENERATOR
	Modification of description in **6.3 (3) Oscillation stabilization time counter status register (OSTC)**	
	Modification of **Cautions 2** in **Figure 6-4 Format of Oscillation Stabilization Time Counter Status Register (OSTC)**	
	Modification of **Cautions 5** in **Figure 6-5 Format of Oscillation Stabilization Time Select Register (OSTS)**	
	Modification of **Cautions 3** in **Figure 6-6 Format of System Clock Control Register (CKC)**	
	Modification of **Cautions 1 to 3** in **Figure 6-8 Format of Operation Speed Mode Control Register (OSMC)**	
	Addition of **Figure 6-14 Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))** and description	
	Addition of **Figure 6-15 Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))** and description	
	Modification of **Cautions 1** in **6.6.1 (1) Example of setting procedure when oscillating the X1 clock**	
	Modification of register name in title of **6.6.1 (2) <2>**	
	Addition of <2> to **6.6.1 (4) (b)**	
	Addition of **Caution** to **6.6.2 (2) (b)**	
	Modification of **Caution** in **6.6.3 Example of controlling subsystem clock**	
	Modification of **Caution** in **6.6.3 (1) Example of setting procedure when oscillating the subsystem clock**	
	Modification of bit name in **6.6.3 (2) <2> Setting the subsystem clock as the source clock of the CPU clock (CKC register)**	
	Modification of **Caution** in **6.6.3 (2) Example of setting procedure when using the subsystem clock as the CPU clock**	
	Modification of register name in title of **6.6.3 (3) <2>**	
	Addition of an arrow from (C) to (B) in **Figure 6-16 CPU Clock Status Transition Diagram**	
	Modification of **Table 6-4 CPU Clock Transition and SFR Register Setting Examples**	
	Addition of description to **Table 6-5 Changing CPU Clock**	
	Modification of description in **6.6.7 Time required for switchover of CPU clock and main system clock**	
	Deletion of **Caution** in **Table 6-8 Maximum Number of Clocks Required in Type 2**	
	Change of bit name of TIS0n0 and TIS0n1 bits to CIS0n0 and CIS0n1 bits in CHAPTER 7	CHAPTER 7 TIMER ARRAY UNIT
	Addition of description in **7.1.1 Functions of each channel when it operates independently**	
	Addition of description in **7.1.2 Functions of each channel when it operates with another channel**	
	Addition of description and table to **7.2 (1) Timer/counter register 0n (TCR0n)**	
	Deletion of **Caution** in **7.2 (2) Timer data register 0n (TDR0n)**	