Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP4 Flash Memory Programmer User's Manual	U15260E
PG-FP5 Flash Memory Programmer	U18865E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

Other Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE – Products and Packages –	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the "Semiconductor Device Mount Manual" website (http://www.necel.com/pkg/en/mount/index.html).

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

3.1.2 Mirror area

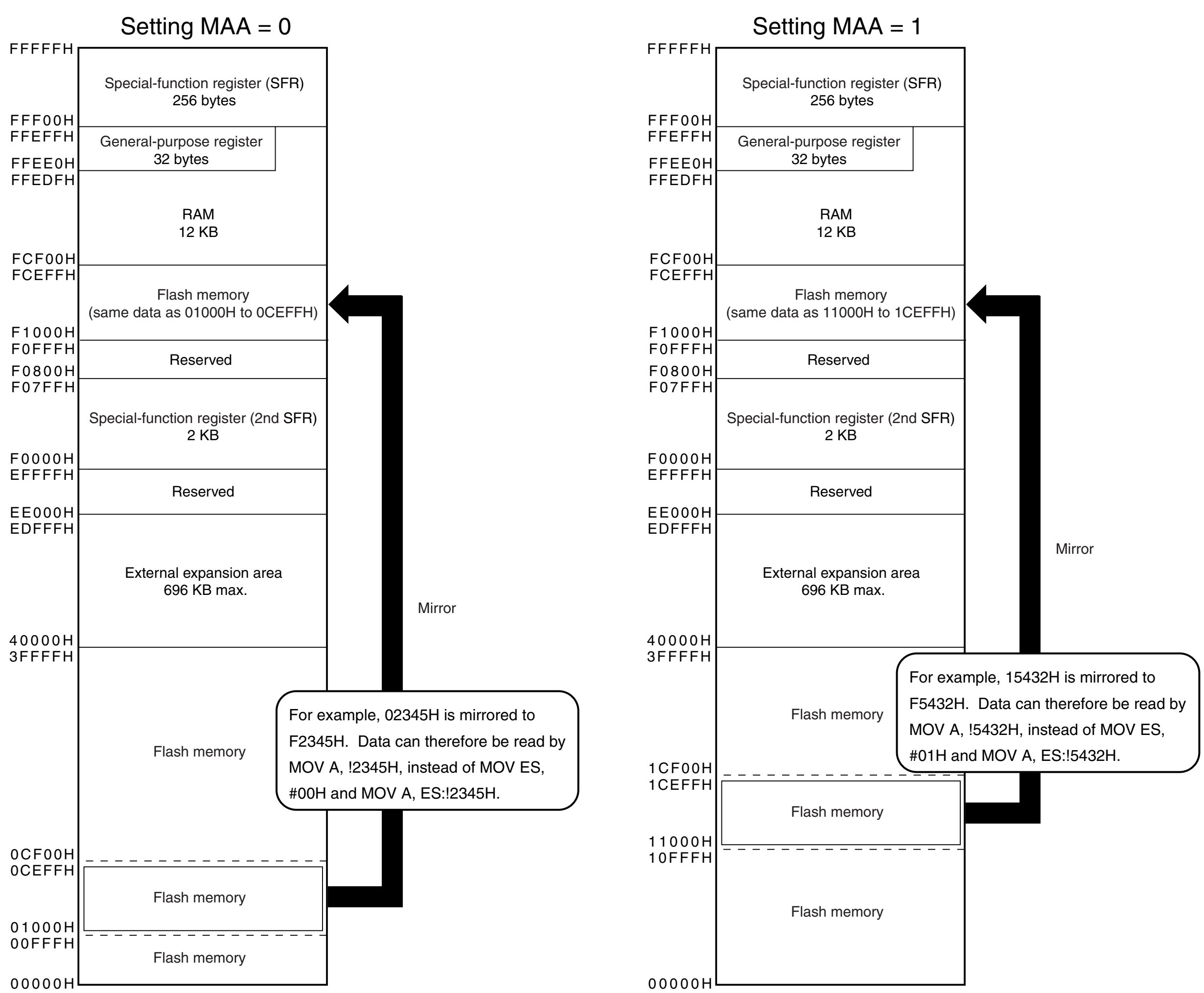
The 78K0R/KH3 mirrors the data flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the data flash area to be mirrored is set by the processor mode control register (PMC)).

By reading data from F0000H to FFFFFH, an instruction that does not have the ES registers as an operand can be used, and thus the contents of the data flash can be read with the shorter code. However, the data flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

See **3.1 Memory Space** for the mirror area of each product.
The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

Example μPD78F1176, 78F1176A (Flash memory: 256 KB, RAM: 12 KB)



Remark MAA: Bit 0 of the processor mode control register (PMC).

PMC register is described below.

3.1.6 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the 78K0R/KH3, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 3-7 to 3-11 show correspondence between data memory and addressing.

<R> **Figure 3-7. Correspondence Between Data Memory and Addressing (μPD78F1174, 78F1174A)**

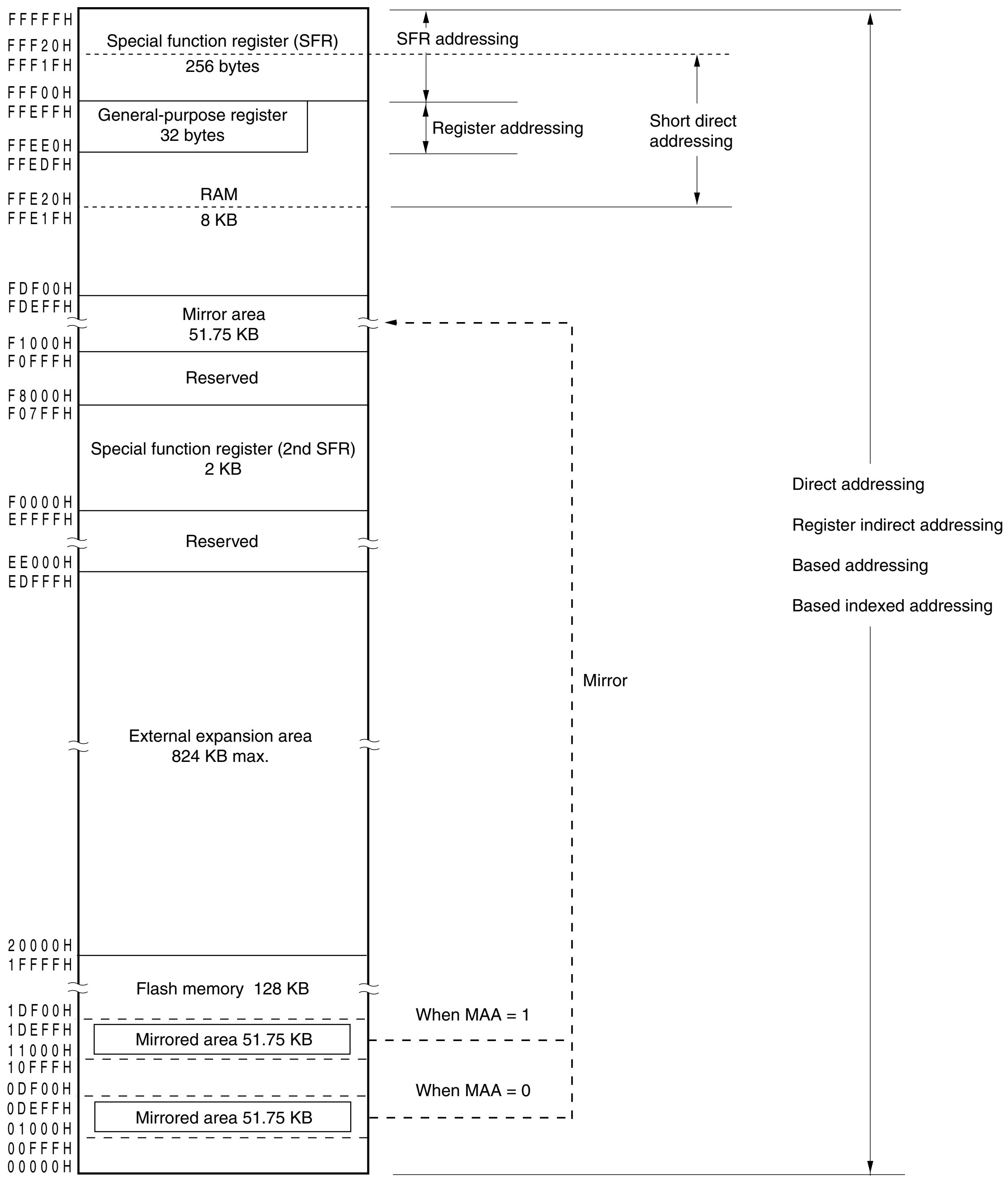
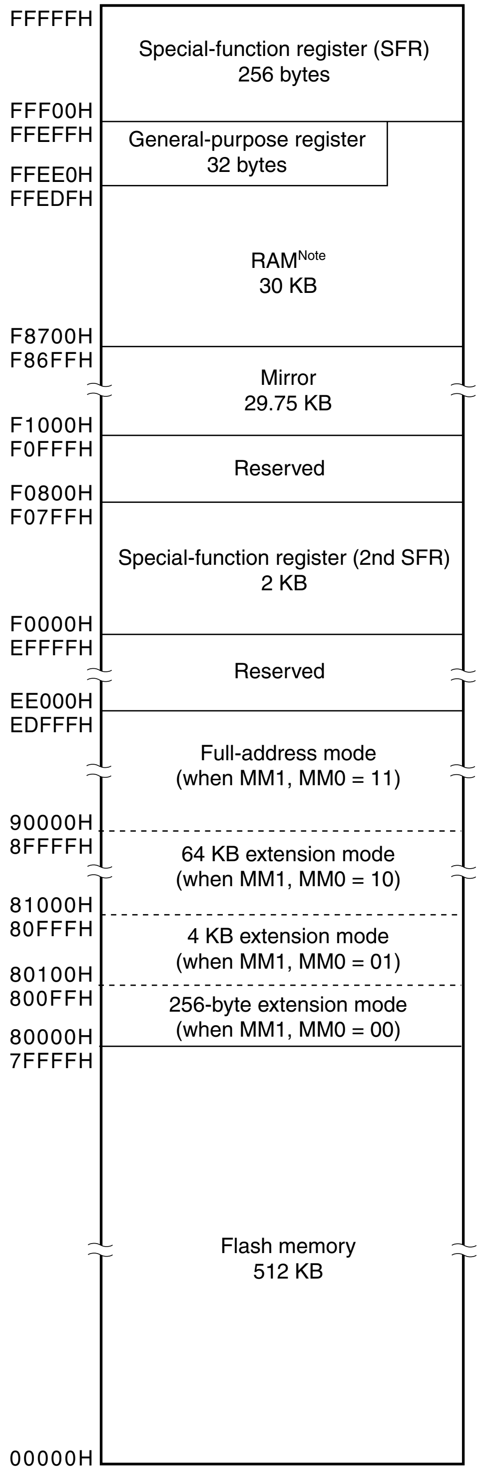


Table 3-6. Extended SFR (2nd SFR) List (2/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–			–	–		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–			–	–		
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	–	–	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	–	–	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	–	–	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	–	–	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		–			–	–		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		–			–	–		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		–			–	–		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	–	√	√	0000H
F0127H		–			–	–		
F0128H	Serial output register 0	SO0		R/W	–	–	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		–			–	–		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	–	√	√	0000H
F0135H		–			–	–		
F0140H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H
F0141H		–			–	–		
F0142H	Serial status register 11	SSR11L	SSR11	R	–	√	√	0000H
F0143H		–			–	–		
F0144H	Serial status register 12	SSR12L	SSR12	R	–	√	√	0000H
F0145H		–			–	–		
F0146H	Serial status register 13	SSR13L	SSR13	R	–	√	√	0000H
F0147H		–			–	–		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	–	√	√	0000H
F0149H		–			–	–		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	–	√	√	0000H
F014BH		–			–	–		

Figure 5-1. Memory Map When Using External Bus Interface Function (3/3)

(e) Memory map of μPD78F1178, 78F1178A



Note Use of the area F8700H to F8EFFH is prohibited when using the self-programming function. Since this area is used for self-programming library.

7.4.4 Collective manipulation of TOmn bits

In the TOm register, the setting bits for all the channels are located in one register in the same way as the TSm register (channel start trigger). Therefore, TOmn of all the channels can be manipulated collectively. Only specific bits can also be manipulated by setting the corresponding TOEmn = 0 to a target TOmn (channel output).

Figure 7-32. Example of TO0n Bits Collective Manipulation

Before writing

TO0	0	0	0	0	0	0	0	0	TO07 0	TO06 0	TO05 1	TO04 0	TO03 0	TO02 0	TO01 1	TO00 0
TOE0	0	0	0	0	0	0	0	0	TOE07 0	TOE06 0	TOE05 1	TOE04 0	TOE03 1	TOE02 1	TOE01 1	TOE00 1

Data to be written

0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1

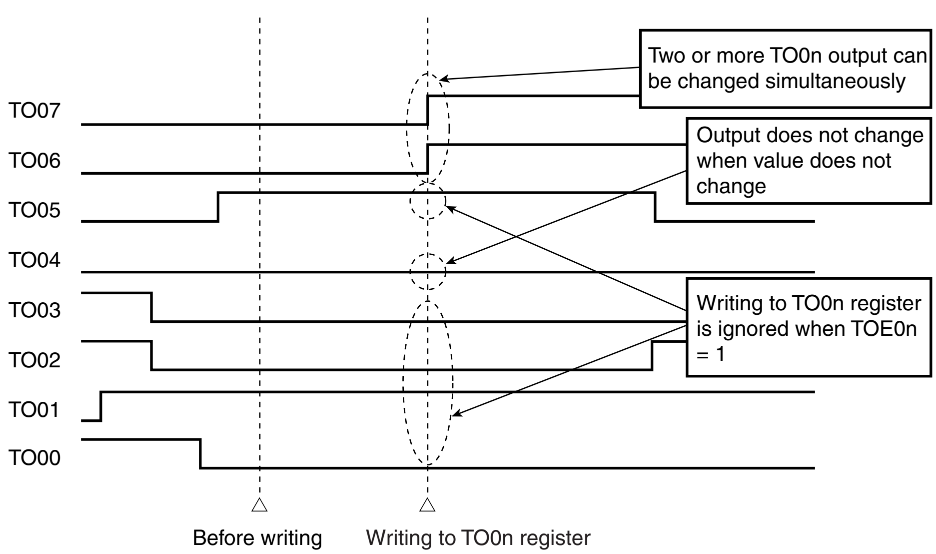
After writing

TO0	0	0	0	0	0	0	0	0	TO07 1	TO06 1	TO05 1	TO04 0	TO03 0	TO02 0	TO01 1	TO00 0

Writing is done only to TOmn bits with TOEmn = 0, and writing to TOmn bits with TOEmn = 1 is ignored.

TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to TOmn, it is ignored and the output change by timer operation is normally done.

Figure 7-33. TO0n Pin Statuses by Collective Manipulation of TO0n Bits



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

Figure 7-39. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/3)

(2) When f_{SUB}/4 is selected as count clock

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn			CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

f_{SUB}/4 edge selection
00B: Detects falling edge (counts on f_{SUB}/4 cycles).
01B: Detects rising edge (counts on f_{SUB}/4 cycles).
10B: Detects both edges (counts on f_{SUB}/2 cycles).
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
1: Selects subsystem clock divided by four (f_{SUB}/4).

Operation clock selection
0: Selects CKm0 as operation clock of channel n.
1: Selects CKm1 as operation clock of channel n.
f_{CLK} (no division) is selected as selected operation clock by TPSm register.

(b) Timer clock select register m (TPSm)

	Bits 7 to 4, 3 to 0	
TPSm	PRSmk3 to PRSmk0	0000B: Selects f_{CLK} (no division) as operation clock selected by CKSmn of TMRmn register.
	0000	k = 0 (bits 0 to 3) when CKm0 is selected and k = 1 (bits 4 to 7) when CKm1 is selected

(c) Timer input select register m (TISm)

	Bit n	
TISm	TISmn	1: Selects subsystem clock divided by four (f_{SUB}/4).
	1	

(d) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	1/0	1: Outputs 1 from TOmn.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13, k =0, 1

2. f_{SUB}: Subsystem clock oscillation frequency

11.4 A/D Converter Operations

11.4.1 Basic operations of A/D converter

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the A/D converter.

<2> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1 to start the operation of the A/D voltage comparator.

<3> Set channels for A/D conversion to analog input by using the A/D port configuration register (ADPC) and set to input mode by using port mode registers 2 and 15 (PM2, PM15).

<4> Set A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM.

<5> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<6> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1.
(<7> to <13> are operations performed by hardware.)

<7> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<8> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<9> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF0} by the tap selector.

<10> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the A/D voltage comparator. If the analog input is greater than (1/2) AV_{REF0}, the MSB of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF0}, the MSB is reset to 0.

<11> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.
- Bit 9 = 1: (3/4) AV_{REF0}
- Bit 9 = 0: (1/4) AV_{REF0}

The voltage tap and sampled voltage are compared and bit 8 of SAR is manipulated as follows.
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<12> Comparison is continued in this way up to bit 0 of SAR.

<13> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<14> Repeat steps <7> to <13>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <6>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <6>. To change a channel of A/D conversion, start from <5>.

Caution Make sure the period of <2> to <6> is 1 μs or more.

Remark Two types of A/D conversion result registers are available.
- ADCR (16 bits): Store 10-bit A/D conversion value
- ADCRH (8 bits): Store 8-bit A/D conversion value

(7) Serial flag clear trigger register mn (SIRmn)

SIRmn is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because SIRmn is a trigger register, it is cleared immediately when the corresponding bit of SSRmn is cleared.

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

The lower 8 bits of SIRmn can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears this register to 0000H.

Figure 13-10. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03), After reset: 0000H R/W
F0148H, F0149H (SIR10), F014AH, F014BH (SIR11),
F014CH, F014DH (SIR12), F014EH, F014FH (SIR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn	PECTmn	OVCTmn

FECTmn	Clear trigger of framing error of channel n
0	No trigger operation
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	No trigger operation
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	No trigger operation
1	Clears the OVFmn bit of the SSRmn register to 0.

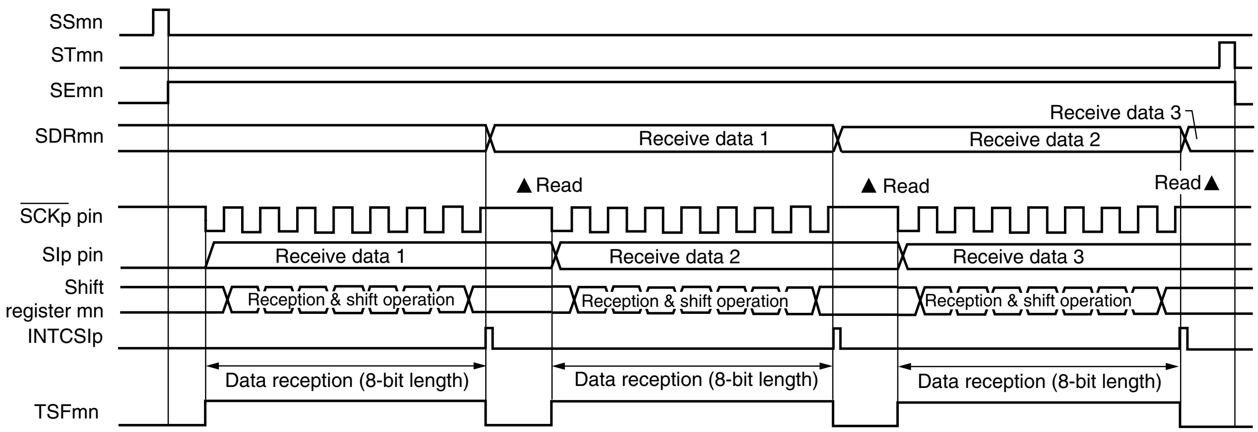
Caution Be sure to clear bits 15 to 3 to “0”.

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

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

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

Figure 13-60. Timing Chart of Slave Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

(1) Register setting

Figure 13-62. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3	CKOm2	CKOm1	CKOm0					SOm3	SOm2	SOm1	SOm0
	0	0	0	0	×	×	×	×	0	0	0	0	0/1	0/1	0/1	0/1

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

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

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0		DLSmn2	DLSmn1	DLSmn0
	1	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15–9	8	7–0
SDRmn	0000000 (baud rate setting)	0	Transmit data setting/receive data register (SIOp)

<R> **Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)
□ : Setting is fixed in the CSI slave transmission/reception mode, ▒ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

13.6 Operation of UART (UART0, UART1, UART2, UART3) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[Data transmission/reception]

- Data length of 5, 7, or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is supported in UART3 (2, 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

$\}$ External interrupt (INTP0) or timer array unit (TAU) is used.

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 2 and 3 of SAU0.
UART2 uses channels 0 and 1 of SAU1.
UART3 uses channels 2 and 3 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

Caution When using serial array units 0 and 1 as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.

UART performs the following four types of communication operations.

- UART transmission (See **13.6.1**.)
- UART reception (See **13.6.2**.)
- LIN transmission (UART3 only) (See **13.6.3**.)
- LIN reception (UART 3 only) (See **13.6.4**.)

Figure 13-74. Procedure for Resuming UART Transmission

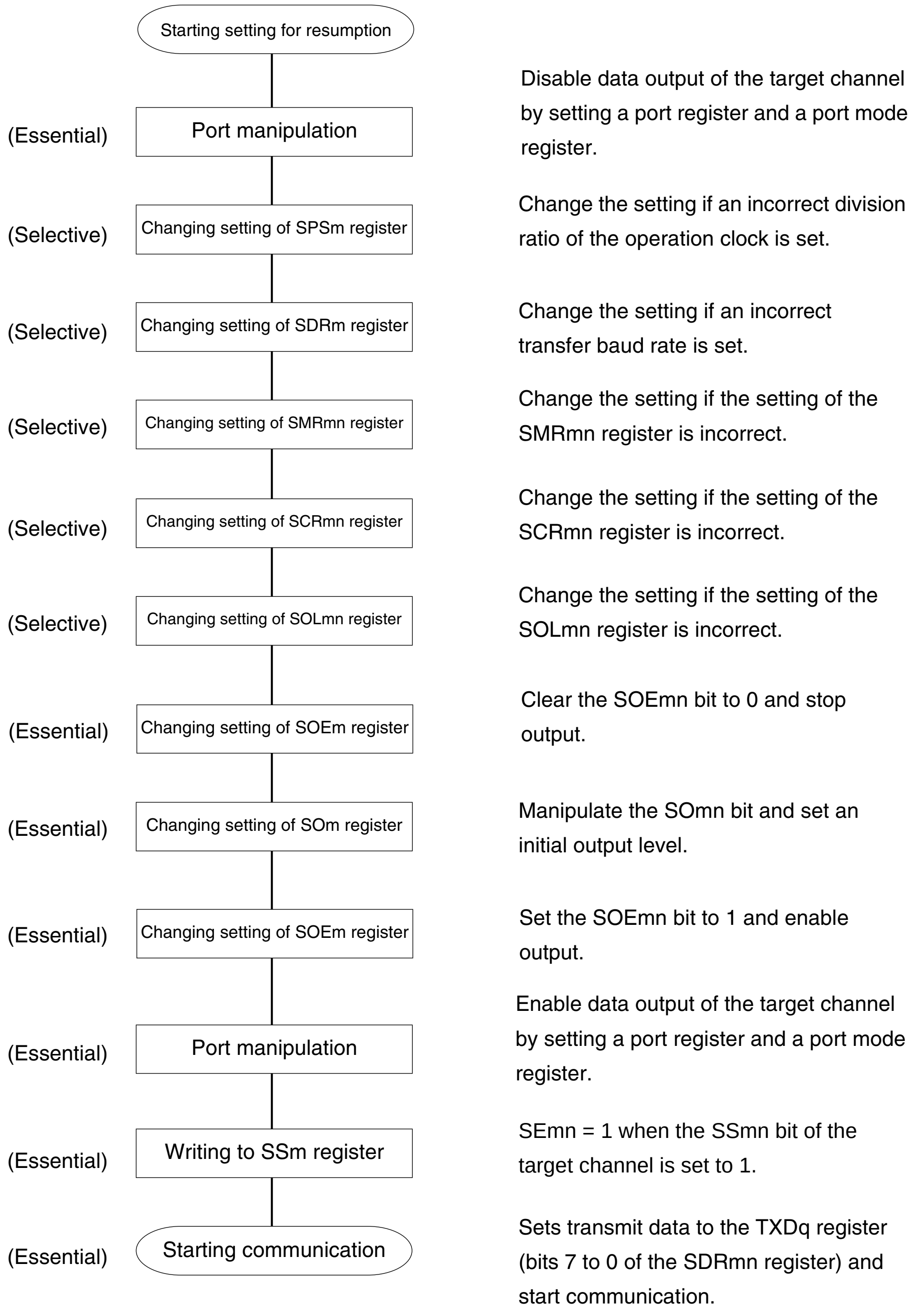
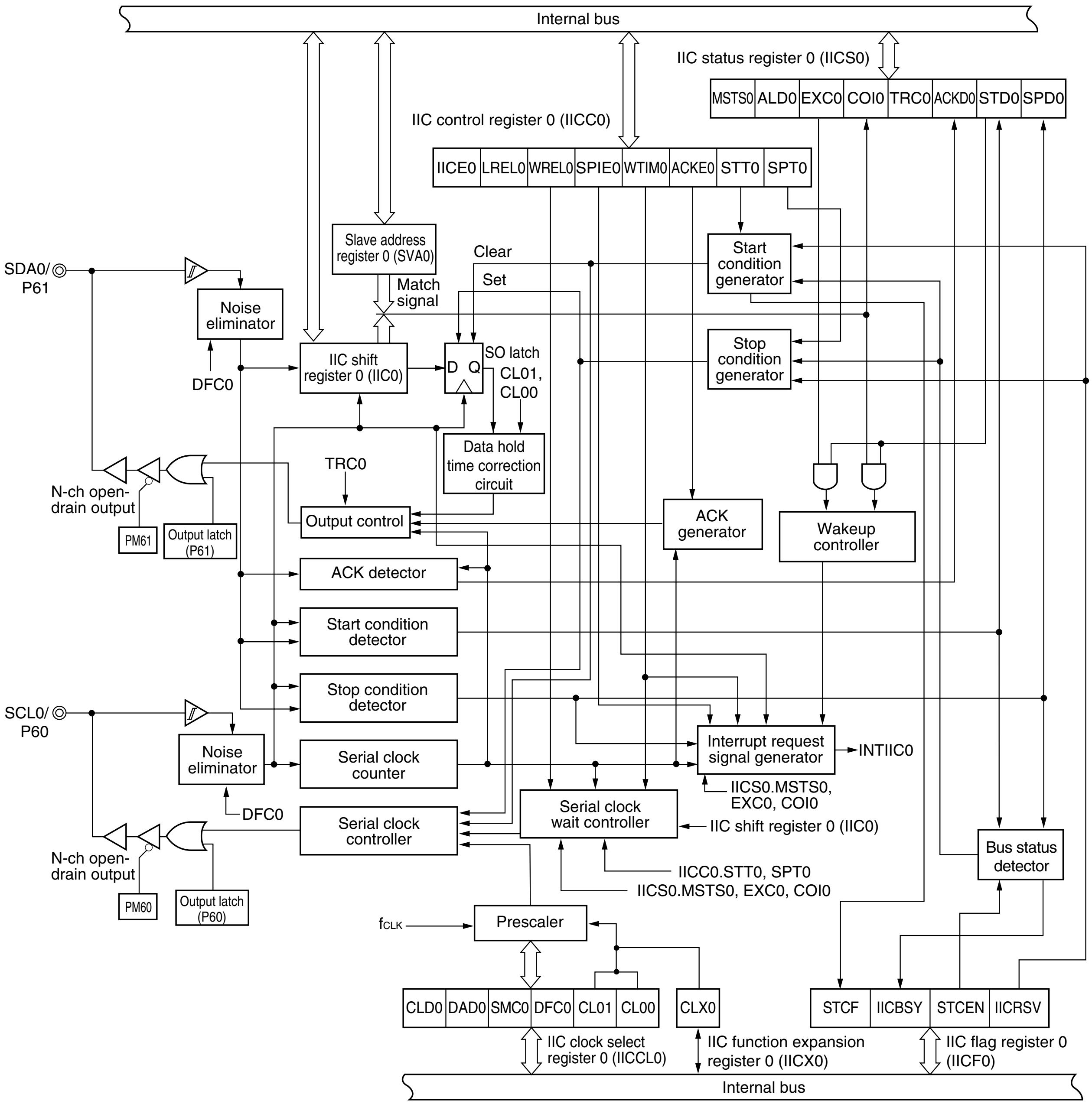


Figure 14-1. Block Diagram of Serial Interface IIC0



(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
								△1

△1: IICS0 = 00000001B

Remark △: Generated only when SPIE0 = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIIC0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

(i) When WTIM0 = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1		▲2		▲3	△4

▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

28.2 Operation List

Table 28-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		A, r Note 3	1	1	–	A ← r			
		r, A Note 3	1	1	–	r ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			
		A, sfr	2	1	–	A ← sfr			
		sfr, A	2	1	–	sfr ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		PSW, #byte	3	3	–	PSW ← byte	×	×	×
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		ES, #byte	2	1	–	ES ← byte			
		ES, saddr	3	1	–	ES ← (saddr)			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		CS, #byte	3	1	–	CS ← byte			
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	–	(DE) ← A			
		[DE + byte], #byte	3	1	–	(DE + byte) ← byte			
		A, [DE + byte]	2	1	4	A ← (DE + byte)			
		[DE + byte], A	2	1	–	(DE + byte) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	–	(HL) ← A			
		[HL + byte], #byte	3	1	–	(HL + byte) ← byte			

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**.

Standard Products

DC Characteristics (8/12)

(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
Supply current	I_{DD1} Note 1	Operating mode	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V		Square wave input		7.0	12.2	mA
					Resonator connection		7.3	12.5	mA
			f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V		Square wave input		7.0	12.2	mA
					Resonator connection		7.3	12.5	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 5.0 V		Square wave input		3.8	6.2	mA
					Resonator connection		3.9	6.3	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 3.0 V		Square wave input		3.8	6.2	mA
					Resonator connection		3.9	6.3	mA
			f_{MX} = 5 MHz Notes 2, 3, V_{DD} = 3.0 V	Normal current mode	Square wave input		2.1	3.0	mA
					Resonator connection		2.2	3.1	mA
				Low consumption current mode Note 4	Square wave input		1.5	2.1	mA
					Resonator connection		1.5	2.1	mA
			f_{MX} = 5 MHz Notes 2, 3, V_{DD} = 2.0 V	Normal current mode	Square wave input		1.4	2.1	mA
					Resonator connection		1.4	2.1	mA
				Low consumption current mode Note 4	Square wave input		1.4	2.0	mA
					Resonator connection		1.4	2.0	mA
			f_{IH} = 8 MHz Note 5		V_{DD} = 5.0 V		3.1	5.0	mA
					V_{DD} = 3.0 V		3.1	5.0	mA

Notes 1. Total current flowing into V_{DD}, EV_{DD0}, EV_{DD1}, AV_{REF0}, and AV_{REF1}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. When internal high-speed oscillator and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FSEL (bit 0 of operation speed mode control register (OSMC)) = 0.

4. When the RMC register is set to 5AH.

5. When high-speed system clock and subsystem clock are stopped. When FSEL (bit 0 of operation speed mode control register (OSMC)) = 0 is set.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

f_{IH}: Internal high-speed oscillation clock frequency

2. For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

<R> **3.** Temperature condition of the TYP. value is T_A = 25°C

(A) Grade Products

DC Characteristics (10/12)

(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
Supply current	I_{DD2} [Note 1]	HALT mode	f_{MX} = 20 MHz [Note 2], V_{DD} = 5.0 V		Square wave input		1.0	2.7	mA
					Resonator connection		1.3	3.0	mA
			f_{MX} = 20 MHz [Note 2], V_{DD} = 3.0 V		Square wave input		1.0	2.7	mA
					Resonator connection		1.3	3.0	mA
			f_{MX} = 10 MHz [Notes 2, 3], V_{DD} = 5.0 V		Square wave input		0.52	1.4	mA
					Resonator connection		0.62	1.5	mA
			f_{MX} = 10 MHz [Notes 2, 3], V_{DD} = 3.0 V		Square wave input		0.52	1.4	mA
					Resonator connection		0.62	1.5	mA
			f_{MX} = 5 MHz [Notes 2, 3], V_{DD} = 3.0 V	Normal current mode	Square wave input		0.36	0.75	mA
					Resonator connection		0.41	0.8	mA
				Low consumption current mode [Note 4]	Square wave input		0.22	0.5	mA
					Resonator connection		0.27	0.55	mA
			f_{MX} = 5 MHz [Notes 2, 3], V_{DD} = 2.0 V	Normal current mode	Square wave input		0.22	0.5	mA
					Resonator connection		0.27	0.55	mA
				Low consumption current mode [Note 4]	Square wave input		0.22	0.5	mA
					Resonator connection		0.27	0.55	mA
			f_{IH} = 8 MHz [Note 5]		V_{DD} = 5.0 V		0.45	1.2	mA
					V_{DD} = 3.0 V		0.45	1.2	mA

Notes 1. Total current flowing into V_{DD}, EV_{DD0}, EV_{DD1}, AV_{REF0}, and AV_{REF1}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The maximum value include the peripheral operation current. However, not including the current flowing into the A/D converter, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. During HALT instruction execution by flash memory.

2. When internal high-speed oscillator and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FSEL (bit 0 of operation speed mode control register (OSMC)) = 0.

4. When the RMC register is set to 5AH.

5. When high-speed system clock and subsystem clock are stopped. When FSEL (bit 0 of operation speed mode control register (OSMC)) = 0 is set.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

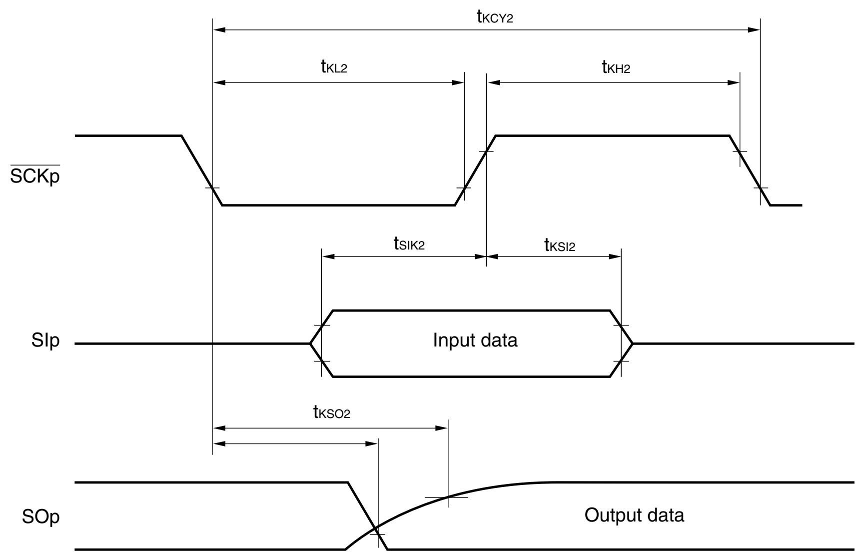
f_{IH}: Internal high-speed oscillation clock frequency

2. For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

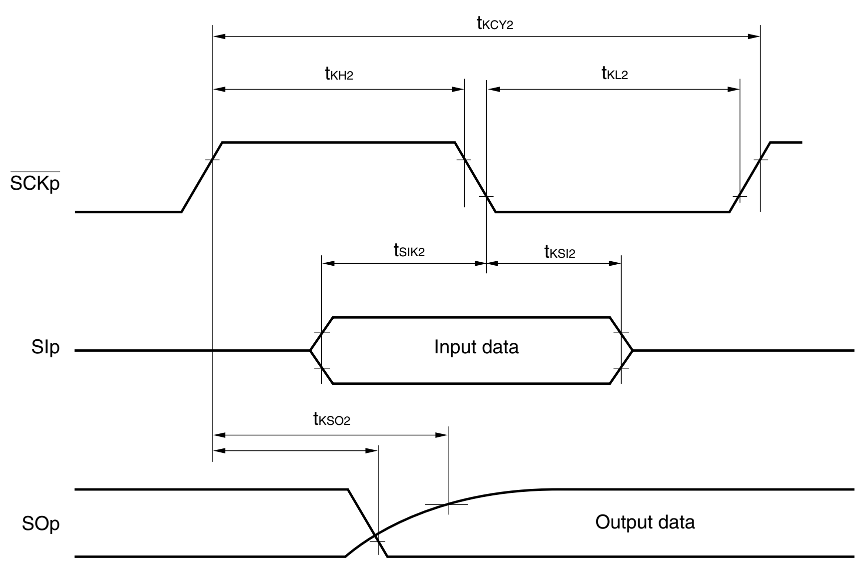
<R> **3.** Temperature condition of the TYP. value is T_A = 25°C

(3) Serial interface: Serial array unit (16/18)

CSI mode serial transfer timing (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



CSI mode serial transfer timing (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



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

Remarks 1. p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)

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

3. CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.

A.5.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0R microcontrollers. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use 78K0R/KH3, use USB interface cable and 16-pin connection cable.

Remark Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html).

A.6 Debugging Tools (Software)

SM+ for 78K0R System simulator	SM+ for 78K0R is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of SM+ for 78K0R allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. SM+ for 78K0R should be used in combination with the device file (DF781188) .
	Part number: μS××××SM781000
ID78K0R-QB Integrated debugger	This debugger supports the in-circuit emulators for the 78K0R microcontrollers. The ID78K0R-QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file.
	Part number: μS××××ID78K0R-QB

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM781000
μS××××ID78K0R-QB

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	