Renesas Electronics America Inc - UPD78F0462GB-GAH-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/0
Core Size	8-Bit
Speed	10MHz
Connectivity	3-Wire SIO, LINbus, UART/USART
Peripherals	LCD, LVD, POR, PWM, WDT
Number of I/O	46
Program Memory Size	24KB (24K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b, 3x16b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f0462gb-gah-ax

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

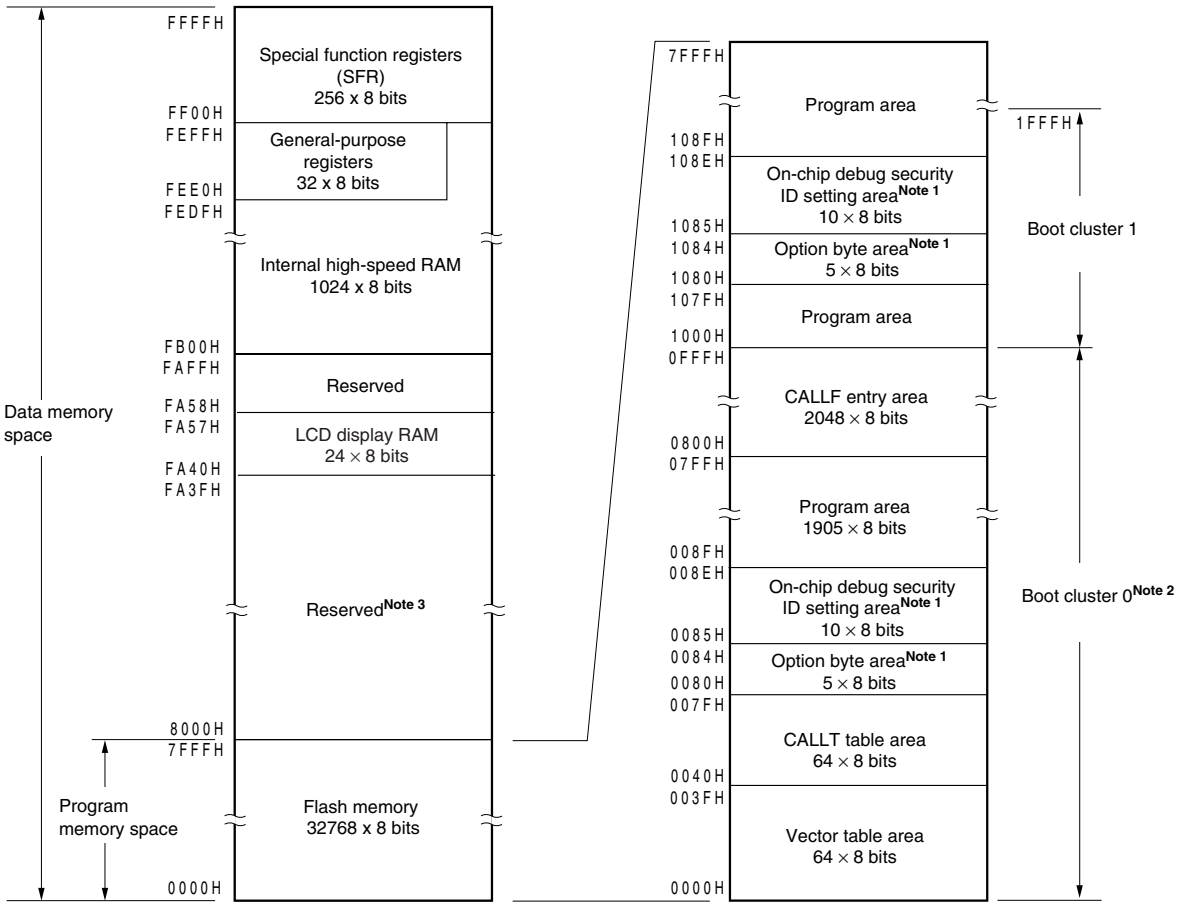
Each products in the 78K0/LE3 can access a 64 KB memory space. Figures 3-1 to 3-10 show the memory maps.

Caution Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) and internal expansion RAM size switching register (IXS) of all products in the 78K0/LE3 are fixed (IMS = CFH, IXS = 0CH). Therefore, set the value corresponding to each product as indicated below.

Table 3-1. Set Values of Internal Memory Size Switching Register (IMS) and Internal Expansion RAM Size Switching Register (IXS)

Flash Memory Version (78K0/LE3)	IMS	IXS	ROM Capacity	Internal High-Speed RAM Capacity	Internal Expansion RAM Capacity
μPD78F0441, 78F0451, 78F0461	04H	0CH	16 KB	768 bytes	–
μPD78F0442, 78F0452, 78F0462	C6H		24 KB	1 KB	
μPD78F0443, 78F0453, 78F0463	C8H		32 KB		
μPD78F0444, 78F0454, 78F0464	CCH	0AH	48 KB		1 KB
μPD78F0445, 78F0455, 78F0465	CFH		60 KB		

Figure 3-6. Memory Map (μPD78F0463)



Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **27.8 Security Setting**).

<R> **3.** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.

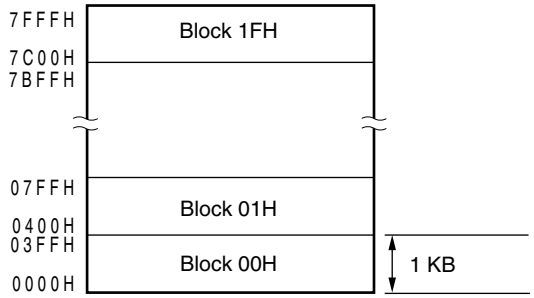
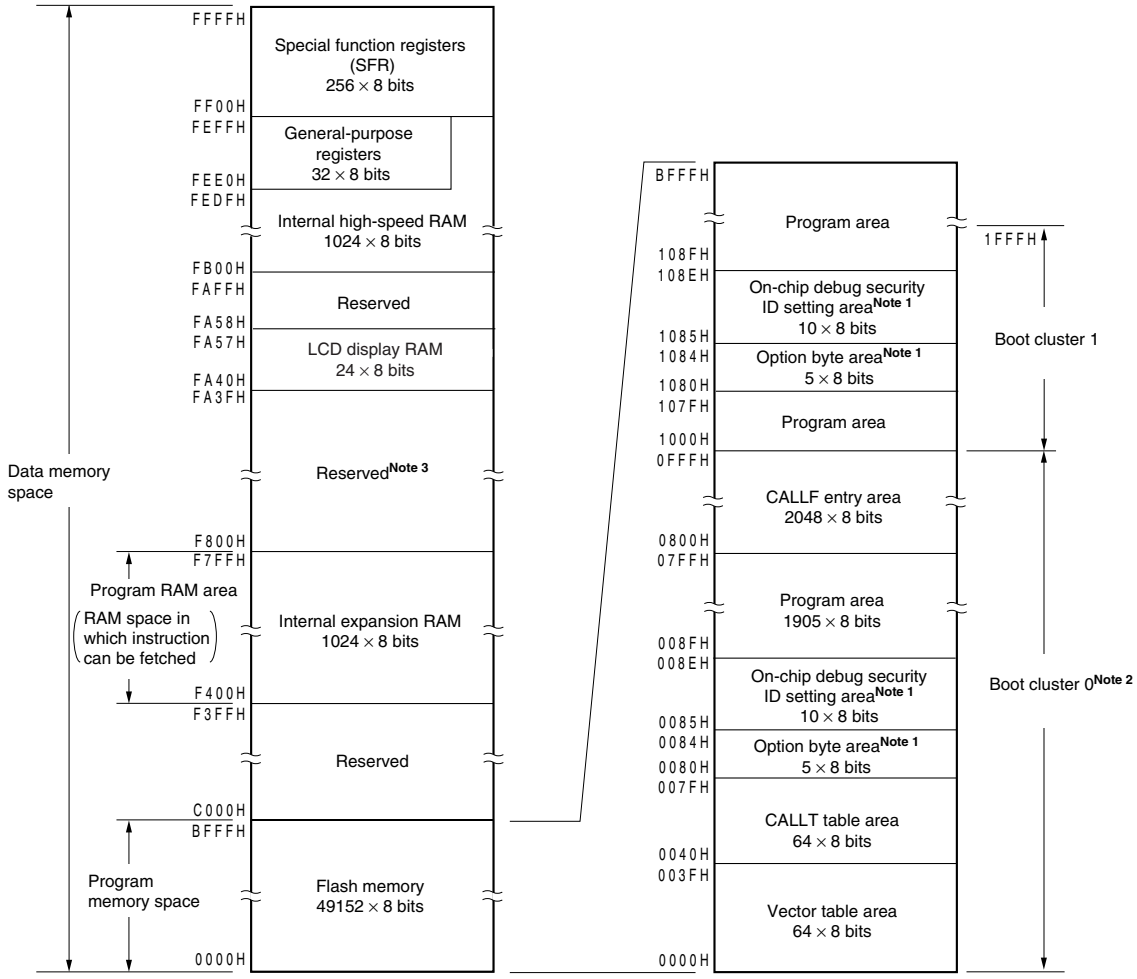


Figure 3-8. Memory Map (μPD78F0464)



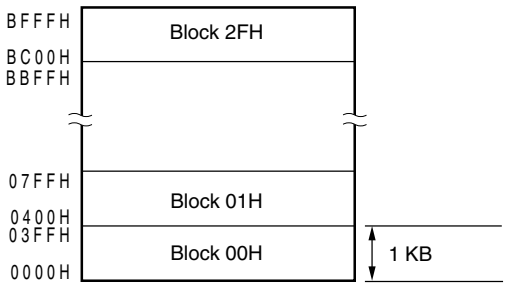
Notes 1. When boot swap is not used: Set the option bytes to 0080H to 0084H, and the on-chip debug security IDs to 0085H to 008EH.

When boot swap is used: Set the option bytes to 0080H to 0084H and 1080H to 1084H, and the on-chip debug security IDs to 0085H to 008EH and 1085H to 108EH.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **27.8 Security Setting**).

<R> **3.** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-2 Correspondence Between Address Values and Block Numbers in Flash Memory**.



(2) CALLT instruction table area

The 64-byte area 0040H to 007FH can store the subroutine entry address of a 1-byte call instruction (CALLT).

(3) Option byte area

A 5-byte area of 0080H to 0084H and 1080H to 1084H can be used as an option byte area. Set the option byte at 0080H to 0084H when the boot swap is not used, and at 0080H to 0084H and 1080H to 1084H when the boot swap is used. For details, see **CHAPTER 26 OPTION BYTE**.

(4) CALLF instruction entry area

The area 0800H to 0FFFH can perform a direct subroutine call with a 2-byte call instruction (CALLF).

(5) On-chip debug security ID setting area

A 10-byte area of 0085H to 008EH and 1085H to 108EH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 0085H to 008EH when the boot swap is not used and at 0085H to 008EH and 1085H to 108EH when the boot swap is used. For details, see **CHAPTER 28 ON-CHIP DEBUG FUNCTION**.

Table 3-8. Special Function Register List (3/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF60H	Sub-count register	RSUBC	R	–	–	√	0000H
FF61H							
FF62H	Second count register	SEC	R/W	–	√	–	00H
FF63H	Minute count register	MIN	R/W	–	√	–	00H
FF64H	Hour count register	HOUR	R/W	–	√	–	12H
FF65H	Week count register	WEEK	R/W	–	√	–	00H
FF66H	Day count register	DAY	R/W	–	√	–	01H
FF67H	Month count register	MONTH	R/W	–	√	–	01H
FF68H	Year count register	YEAR	R/W	–	√	–	00H
FF69H	8-bit timer H mode register 0	TMHMD0	R/W	√	√	–	00H
FF6AH	Timer clock selection register 50	TCL50	R/W	√	√	–	00H
FF6BH	8-bit timer mode control register 50	TMC50	R/W	√	√	–	00H
FF6CH	8-bit timer H mode register 1	TMHMD1	R/W	√	√	–	00H
FF6DH	8-bit timer H carrier control register 1	TMCYC1	R/W	√	√	–	00H
FF6EH	Key return mode register	KRM	R/W	√	√	–	00H
FF6FH	8-bit timer counter 51	TM51	R	–	√	–	00H
FF70H	Asynchronous serial interface operation mode register 0	ASIM0	R/W	√	√	–	01H
FF71H	Baud rate generator control register 0	BRGC0	R/W	–	√	–	1FH
FF72H	Receive buffer register 0	RXB0	R	–	√	–	FFH
FF73H	Asynchronous serial interface reception error status register 0	ASIS0	R	–	√	–	00H
FF74H	Transmit shift register 0	TXS0	W	–	√	–	FFH
FF75H	16-bit ΔΣ A/D conversion end channel register[Note 1]	ADDSTR	R	–	√	–	00H
FF7CH	ΔΣ A/D converter control register 0[Note 1]	ADDCTL0	R/W	√	√	–	00H
FF7DH	ΔΣ A/D converter control register 1[Note 1]	ADDCTL1	R/W	√	√	–	00H
FF7EH	16-bit ΔΣ A/D conversion result register[Note 1]	ADDCR	R	–	–	√	0000H
FF7FH	8-bit ΔΣ A/D conversion result register[Note 1]	ADDCRH	R	–	√	–	00H
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H
FF82H	Watch error correction register	SUBCUD	R/W	√	√	–	00H
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	00H
FF86H	Alarm minute register	ALARMWM	R/W	–	√	–	00H
FF87H	Alarm hour register	ALARMWH	R/W	–	√	–	12H
FF88H	Alarm week register	ALARMWW	R/W	–	√	–	00H
FF89H	Real-time counter control register 0	RTCC0	R/W	√	√	–	00H
FF8AH	Real-time counter control register 1	RTCC1	R/W	√	√	–	00H
FF8BH	Real-time counter control register 2	RTCC2	R/W	√	√	–	00H
FF8CH	Timer clock selection register 51	TCL51	R/W	√	√	–	00H
FF8DH	A/D converter mode register[Note 2]	ADM	R/W	√	√	–	00H
FF8EH	Analog input channel specification register[Note 2]	ADS	R/W	√	√	–	00H
FF8FH	A/D port configuration register 0[Note 2]	ADPC0	R/W	√	√	–	08H

Notes 1. μPD78F046x only.

2. μPD78F045x and 78F046x only.

<R> **(12) Reading of 16-bit timer counter 00 (TM00)**

TM00 can be read without stopping the actual counter, because the count values captured to the buffer are fixed when it is read. The buffer, however, may not be updated when it is read immediately before the counter counts up, because the buffer is updated at the timing the counter counts up.

Figure 6-62. 16-bit Timer Counter 00 (TM00) Read Timing

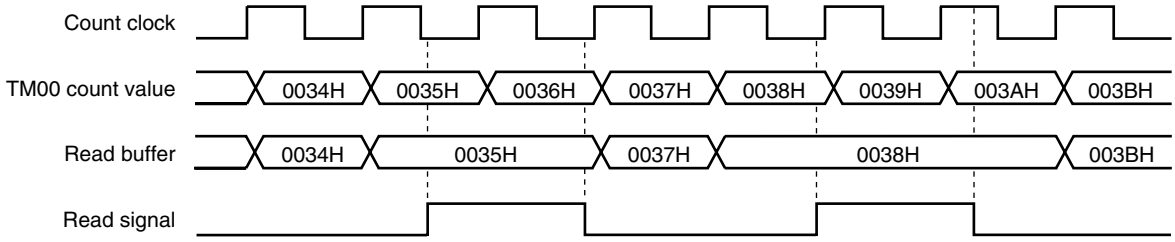


Figure 7-8. Format of Timer Clock Selection Register 52 (TCL52)

Address: FF5BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL52	0	0	0	0	0	TCL522	TCL521	TCL520

TCL522	TCL521	TCL520	Count clock selection[Note 1]			
				f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 10 MHz
0	0	0	Falling edge of clock selected by ISC2			
0	0	1	Rising edge of clock selected by ISC2			
0	1	0	f_{PRS}[Note 2]	2 MHz	5 MHz	10 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz
1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz
1	1	1	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz

Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.

- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of TCL522, TCL521, TCL520 = 0, 1, 0 (count clock: f_{PRS}) is prohibited.

Cautions 1. When rewriting TCL52 to other data, stop the timer operation beforehand.

2. Be sure to clear bits 3 to 7 to 0.

Remark f_{PRS}: Peripheral hardware clock frequency

Figure 12-9. Format of A/D Port Configuration Register 0 (ADPC0)

Address: FF8FH After reset: 08H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	ADPC03	ADPC02	ADPC01	ADPC00

<μPD78F045x>

ADPC03	ADPC02	ADPC01	ADPC00	Digital I/O (D)/analog input (A) switching							
				P27/ ANI7/ SEG24	P26/ ANI6/ SEG25	P25/ ANI5/ SEG26	P24/ ANI4/ SEG27	P23/ ANI3/ SEG28	P22/ ANI2/ SEG29	P21/ ANI1/ SEG30	P20/ ANI0/ SEG31
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	A	A	A	A	A	A	A	D
0	0	1	0	A	A	A	A	A	A	D	D
0	0	1	1	A	A	A	A	A	D	D	D
0	1	0	0	A	A	A	A	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
Other than above				Setting prohibited							

<μPD78F046x>

ADPC03	ADPC02	ADPC01	ADPC00	Digital I/O (D)/analog input (A: successive approximation type, Δ: ΔΣ type) switching							
				P27/ ANI7/ REF+	P26/ ANI6/ REF-	P25/ ANI5/ DS2+	P24/ ANI4/ DS2-	P23/ ANI3/ DS1+	P22/ ANI2/ DS1-	P21/ ANI1/ DS0+	P20/ ANI0/ DS0-
0	0	0	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ
0	0	0	1	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A	D
0	0	1	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	D	D
0	0	1	1	A/Δ	A/Δ	A/Δ	A/Δ	A	D	D	D
0	1	0	0	A/Δ	A/Δ	A/Δ	A/Δ	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
Other than above				Setting prohibited							

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode register 2 (PM2).

2. Do not set the pin set by ADPC0 as digital I/O by ADS, ADDS1, or ADDS0.

3. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 33 CAUTIONS FOR WAIT.

4. If pins ANI0/P20/SEG31 to ANI7/P27/SEG24 are set to segment output pins via the PF2 register, output is set to segment output, regardless of the ADPC0 setting (for μPD78F045x only).

12.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

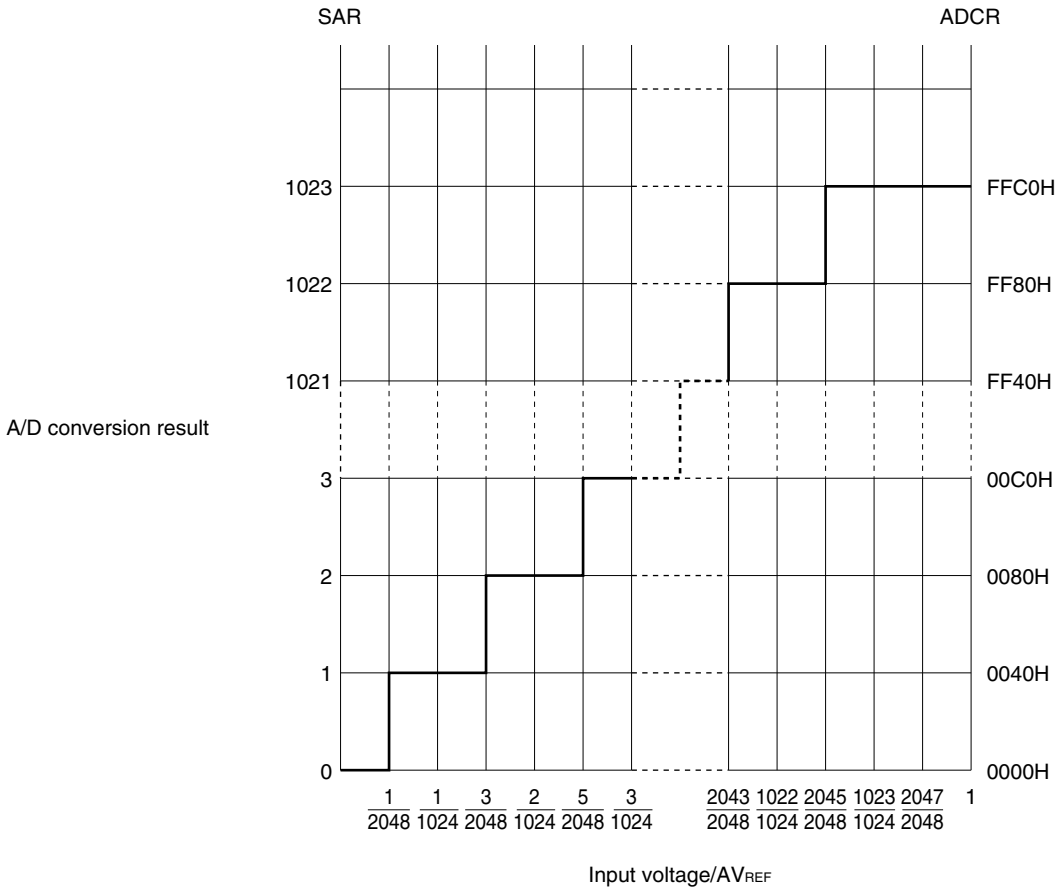
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 12-12 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 12-12. Relationship Between Analog Input Voltage and A/D Conversion Result



14.4.2 Asynchronous serial interface (UART) mode

In this mode, 1-byte data is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 0 (ASIM0)
- Asynchronous serial interface reception error status register 0 (ASIS0)
- Baud rate generator control register 0 (BRGC0)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the BRGC0 register (see **Figure 14-4**).
<2> Set bits 1 to 4 (SL0, CL0, PS00, and PS01) of the ASIM0 register (see **Figure 14-2**).
<3> Set bit 7 (POWER0) of the ASIM0 register to 1.
<4> Set bit 6 (TXE0) of the ASIM0 register to 1. → Transmission is enabled.
Set bit 5 (RXE0) of the ASIM0 register to 1. → Reception is enabled.
<5> Write data to the TXS0 register. → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 14-2. Relationship Between Register Settings and Pins

POWER0	TXE0	RXE0	PM13	P13	PM12	P12	UART0 Operation	Pin Function	
								TxD0/SO10 /<TxD6>/P13	RxD0/SI10 /<RxD6>/P12
0	0	0	×Note	×Note	×Note	×Note	Stop	SO10/<TxD6>/P13	SI10/<RxD6>/P12
1	0	1	×Note	×Note	1	×	Reception	SO10/P13	RxD0
	1	0	0	×	×Note	×Note	Transmission	TxD0	SI10/P12
	1	1	0	×	1	×	Transmission/ reception	TxD0	RxD0

Note Can be set as port function, serial interface CSI10, or serial interface UART6 (only when UART0 is stopped).

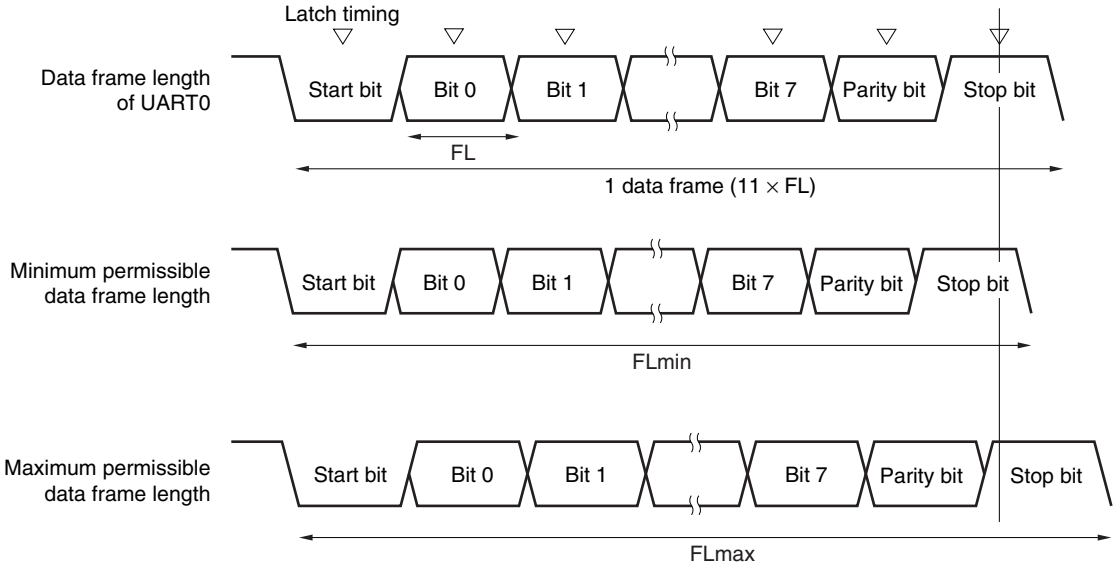
Remark ×: don't care
POWER0: Bit 7 of asynchronous serial interface operation mode register 0 (ASIM0)
TXE0: Bit 6 of ASIM0
RXE0: Bit 5 of ASIM0
PM1×: Port mode register
P1×: Port output latch

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 14-13. Permissible Baud Rate Range During Reception



As shown in Figure 14-13, the latch timing of the receive data is determined by the counter set by baud rate generator control register 0 (BRGC0) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.

Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$FL = (Brate)^{-1}$

Brate: Baud rate of UART0
k: Set value of BRGC0
FL: 1-bit data length
Margin of latch timing: 2 clocks

(d) Continuous transmission

The next transmit data can be written to transmit buffer register 6 (TXB6) as soon as transmit shift register 6 (TXS6) has started its shift operation. Consequently, even while the INTST6 interrupt is being serviced after transmission of one data frame, data can be continuously transmitted and an efficient communication rate can be realized. In addition, the TXB6 register can be efficiently written twice (2 bytes) without having to wait for the transmission time of one data frame, by reading bit 0 (TXSF6) of asynchronous serial interface transmission status register 6 (ASIF6) when the transmission completion interrupt has occurred.

To transmit data continuously, be sure to reference the ASIF6 register to check the transmission status and whether the TXB6 register can be written, and then write the data.

Cautions 1. The TXBF6 and TXSF6 flags of the ASIF6 register change from "10" to "11", and to "01" during continuous transmission. To check the status, therefore, do not use a combination of the TXBF6 and TXSF6 flags for judgment. Read only the TXBF6 flag when executing continuous transmission.

2. When the device is use in LIN communication operation, the continuous transmission function cannot be used. Make sure that asynchronous serial interface transmission status register 6 (ASIF6) is 00H before writing transmit data to transmit buffer register 6 (TXB6).

TXBF6	Writing to TXB6 Register
0	Writing enabled
1	Writing disabled

Caution To transmit data continuously, write the first transmit data (first byte) to the TXB6 register. Be sure to check that the TXBF6 flag is "0". If so, write the next transmit data (second byte) to the TXB6 register. If data is written to the TXB6 register while the TXBF6 flag is "1", the transmit data cannot be guaranteed.

The communication status can be checked using the TXSF6 flag.

TXSF6	Transmission Status
0	Transmission is completed.
1	Transmission is in progress.

Cautions 1. To initialize the transmission unit upon completion of continuous transmission, be sure to check that the TXSF6 flag is "0" after generation of the transmission completion interrupt, and then execute initialization. If initialization is executed while the TXSF6 flag is "1", the transmit data cannot be guaranteed.

2. During continuous transmission, the next transmission may complete before execution of INTST6 interrupt servicing after transmission of one data frame. As a countermeasure, detection can be performed by developing a program that can count the number of transmit data and by referencing the TXSF6 flag.

(4) Port mode register 1 (PM1)

This register sets port 1 input/output in 1-bit units.

When using P11/$\overline{SCK10}$ as the clock output pin of the serial interface, clear PM11 to 0, and set the output latches of P11 to 1.

When using P13/SO10/TxD0/<TxD6> as the data output pin of the serial interface, clear PM13 and the output latches of P13 to 0.

When using P11/$\overline{SCK10}$ as the clock input pin of the serial interface and P12/SI10/RxD0/<RxD6> as the data input pin, set PM11 and PM12 to 1. At this time, the output latches of P11 and P12 may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 16-5. Format of Port Mode Register 1 (PM1)

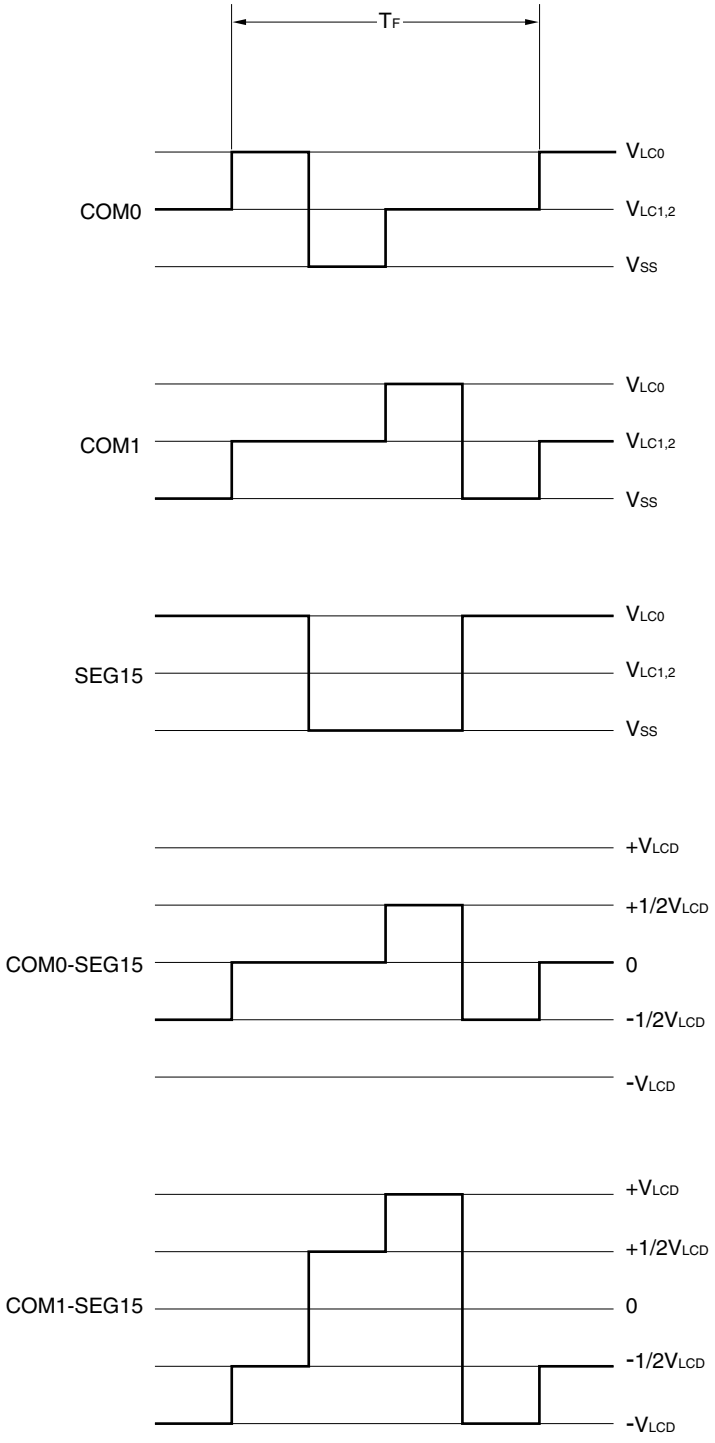
Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	1	1	PM14	PM13	PM12	PM11	1

PM1n	P1n pin I/O mode selection (n = 1 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 17-22. Two-Time-Slice LCD Drive Waveform Examples (1/2 Bias Method)

(a) When segment key scan function is not used (KSON = 0)



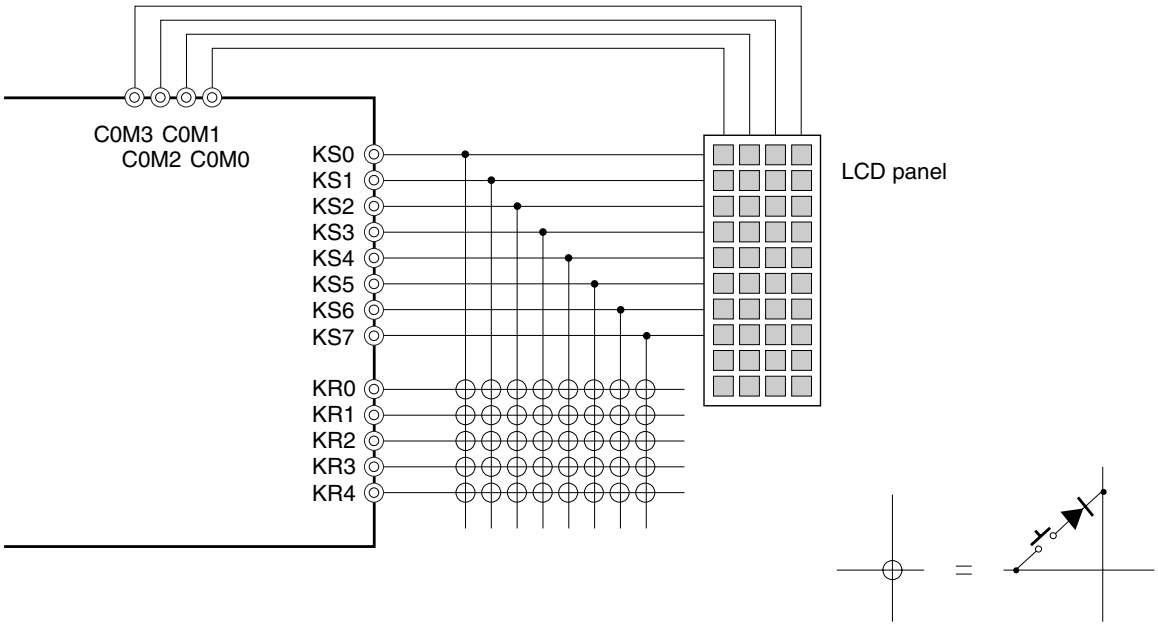
17.8 Operation of Segment Key Scan Function

The segment key scan function is used to reduce the number of pins used by outputting LCD display segment output and key scan signals from the same pin.

Caution This function may affect the LCD panel, depending on how it is used.
Use the function after thorough evaluation.

17.8.1 Circuit configuration example

Figure 17-33. Circuit configuration example



CHAPTER 18 MANCHESTER CODE GENERATOR

18.1 Functions of Manchester Code Generator

The following three types of modes are available for the Manchester code generator.

(1) Operation stop mode

This mode is used when output by the Manchester code generator/bit sequential buffer is not performed. This mode reduces the power consumption.

For details, refer to **18.4.1 Operation stop mode**.

(2) Manchester code generator mode

This mode is used to transmit Manchester code from the MCGO pin.

The transfer bit length can be set and transfers of various bit lengths are enabled. Also, the output level of the data transfer and LSB- or MSB-first can be set for 8-bit transfer data.

(3) Bit sequential buffer mode

This mode is used to transmit bit sequential data from the MCGO pin.

The transfer bit length can be set and transfers of various bit lengths are enabled. Also, the output level of the data transfer and LSB- or MSB-first can be set for 8-bit transfer data.

18.2 Configuration of Manchester Code Generator

The Manchester code generator includes the following hardware.

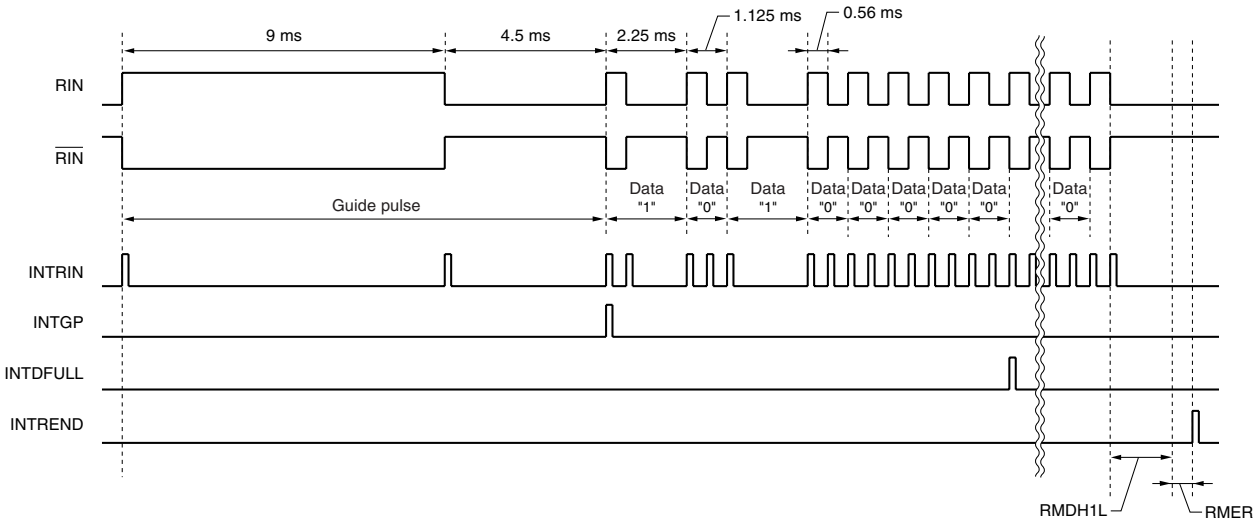
Table 18-1. Configuration of Manchester Code Generator

Item	Configuration
Registers	MCG transmit buffer register (MC0TX) MCG transmit bit count specification register (MC0BIT)
Control registers	MCG control register 0 (MC0CTL0) MCG control register 1 (MC0CTL1) MCG control register 2 (MC0CTL2) MCG status register (MC0STR) Port mode register 3 (PM3) Port register 3 (P3)

19.4.3 Format of type B reception mode

Figure 19-8 shows the data format for type B.

Figure 19-8. Example of Type B Data Format



Remark $\overline{\text{RIN}}$ is the internally inverted signal of RIN.

19.4.4 Operation flow of type B reception mode

Figure 19-9 shows the operation flow.

Cautions 1. When INTRERR is generated, RMSR and RMSCR are automatically cleared immediately.

2. When data has been set to all the bits of RMSR, the following processing is automatically performed.

- **The value of RMSR is transferred to RMDR.**
- **INTDFULL is generated.**
- **RMSR is cleared.**

RMDR must then be read before the next data is set to all the bits of RMSR.

3. When INTREND has been generated, read RMSCR first followed by RMSR.
When RMSR has been read, RMSCR and RMSR are automatically cleared.
If INTREND is generated, the next data cannot be received until RMSR is read.

4. RMSR, RMSCR, and RMDR are cleared simultaneously to operation termination (RMEN = 0).

Table 27-13. Processing Time and Interrupt Acknowledgment (4/4)
(When Static Model Library and Entry RAM Are Allocated Within Short Direct Addressing Range)

Function Name		Processing Time (Unit: μs)			Interrupt Acknowledgment
		RSTOP = 0 and RSTS = 1 (during stable operation of internal high-speed oscillator)		RSTOP = 1 (internal high-speed oscillator stopped)[Note]	
		MCS = 0 (CPU operates with internal high-speed oscillation clock)	MCS = 1 (CPU operates with high-speed system clock)	MCS = 1 (CPU operates with high-speed system clock)	
Self programming start function		34/f_{CPU}	34/f_{CPU}	34/f_{CPU}	Disabled
Self programming end function		34/f_{CPU}	34/f_{CPU}	34/f_{CPU}	Disabled
Initialize function		55/f_{CPU}+462	55/f_{CPU}+462	55/f_{CPU}+473	Disabled
Block erase function		136/f_{CPU}+352516	136/f_{CPU}+352516	136/f_{CPU}+352528	Enabled
Word write function		272/f_{CPU}+477+2142×W	272/f_{CPU}+477+2142×W	272/f_{CPU}+488+2142×W	Enabled
Block verify function		136/f_{CPU}+24918	136/f_{CPU}+24918	136/f_{CPU}+24930	Enabled
Block blank check function		136/f_{CPU}+12128	136/f_{CPU}+12128	136/f_{CPU}+12139	Enabled
Get information function	Option value: 03H	134/f_{CPU}+388	134/f_{CPU}+388	134/f_{CPU}+399	Disabled
	Option value: 04H	144/f_{CPU}+378	144/f_{CPU}+378	144/f_{CPU}+390	Disabled
	Option value: 05H	304/f_{CPU}+363	304/f_{CPU}+363	304/f_{CPU}+375	Disabled
Set information function		72/f_{CPU}+752540	72/f_{CPU}+752540	72/f_{CPU}+753654	Enabled
Mode check function		30/f_{CPU}+274	30/f_{CPU}+274	30/f_{CPU}+286	Disabled
EEPROM write function		268/f_{CPU}+619+2286×W	268/f_{CPU}+619+2286×W	268/f_{CPU}+630+2286×W	Enabled

Note This is the function processing time when the function is executed immediately after the self programming start function has been executed. The processing time after a function other than the self programming start function has been executed is the same as that of RSTOP = 0.

Remark RSTOP: Bit 0 of the internal oscillation mode register (RCM)
RSTS: Bit 7 of RCM
MCS: Bit 1 of main clock mode register (MCM)
f_{CPU}: CPU clock frequency
W: Number of words to be written (1 word = 4 bytes)

29.3 Instructions Listed by Addressing Type

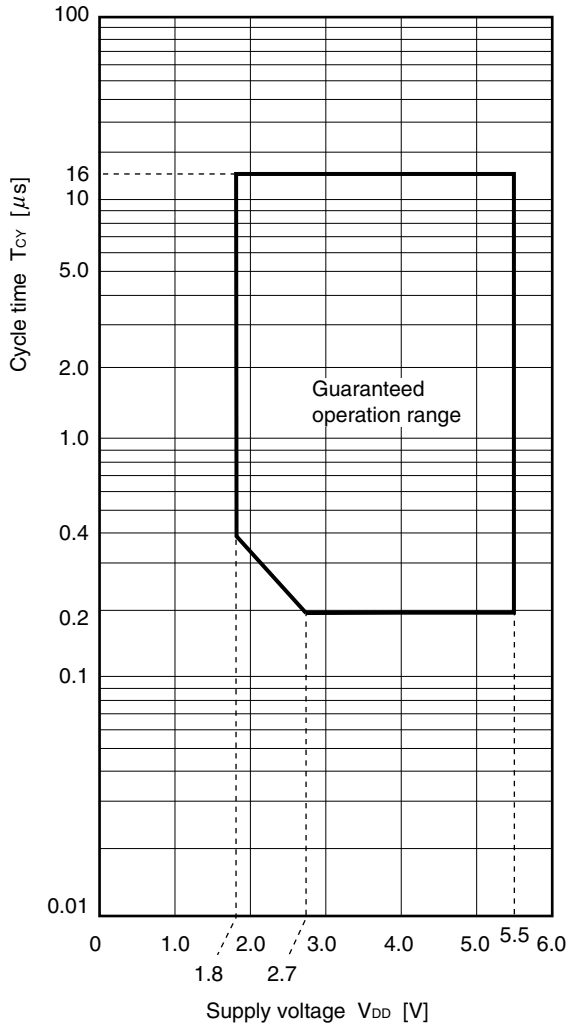
(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

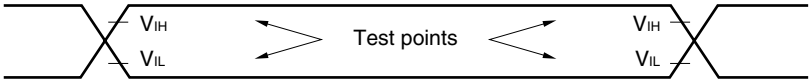
First Operand \ Second Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except “r = A”

T_{CY} vs. V_{DD} (Main System Clock Operation)



AC Timing Test Points (Excluding External Main System Clock)



External Main System Clock Timing