FUJITSU SEMICONDUCTOR
DATA SHEET

DS07-13506-1E

16-bit Proprietary Microcontroller

CMOS

F²MC-16F MB90242A Series

MB90242A

■ DESCRIPTION

The MB90242A is a 16-bit microcontroller optimized for “mechatronics” control applications such as hard disk drive unit control.

The instruction set is based on the AT architecture of the F²MC*-16, 16H family, with additional high-level language supporting instruction, expanded addressing modes, enhanced multiplication and division instructions, and improved bit processing instructions. In addition, long-word data can now be processed due to the inclusion of a 32-bit accumulator.

The MB90242A has a multiply/accumulate unit as a peripheral resource, allowing easy realization of digital filters such as IIR or FIR. The MB90242A has abundant embedded peripheral features, such as 6-channel 8/10-bit A/D converter, UART, 2-channel + 1-channel timer, 4-channel input capture and 4-channel external interrupt.

*1: F²MC stands for FUJITSU Flexible Microcontroller.

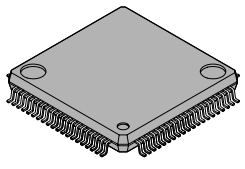
■ FEATURES

- F²MC-16F CPU
  Minimum execution time: 62.5 ns (32 MHz oscillation: 5.0 V ±10%)
  Instruction set optimized for controller applications
  Improved instruction set applicable to high-level language (C) and multitasking
  Improved execution speed: 8-byte queue
  Powerful interrupt fuctions (interrupt processing time: 1.0 μs 32 MHz oscillation)
  Automatic transfer function independent of instructions
  Extended intelligent I/O Service

(Continued)

■ PACKAGE

80-pin Plastic LQFP



(FPT-80P-M05)

(Continued)

- DSP unit
  Specific function for calculations of IIR
  A maximum of 8 product resulted from signed 16-bit × 16-bit multiplications can be accumulated.

  $Y_k = \sum_{n=0}^{N} b_n Y_{k-n} + \sum_{m=0}^{M} a_m X_{k-m}$ is executed in 0.625 µs (at oscillation of 32 MHz, N = M = 3)

  The N and M value is set to a maximum of 3, independently.
- Internal RAM: 2 Kbytes (MB90242A)
  Depending on mode settings, data stored on RAM can be executed as CPU instructions.
- General-purpose ports: max. 38 channels
- A/D converter (analog inputs: 6 channels)
  Resolution: 10 bits
  Conversion time: min. 1.25 µs
  Switchable to 8/10 bits
  Number of registers for storing conversion results: 4
- 8-bit UART: 1 channel
- 8/16-bit I/O simple serial interface (8 Mbps max.): 2 channels
- 16-bit free-run timer: Operating clock cycle 0.25 µs
- 16-bit input capture: 4 channels
  Activated by selected edges
- 16-bit reload timer: 2 channels
- External interrupts: 4 channels
- Timebase timer: 18 bits
- Watchdog timer
- Clock gear function
- Low-power consumption modes
  Sleep mode
  Stop mode
  Hardware standby mode
- Packages: LQFP-80
- CMOS 0.8 µm technology

MB90242A Series

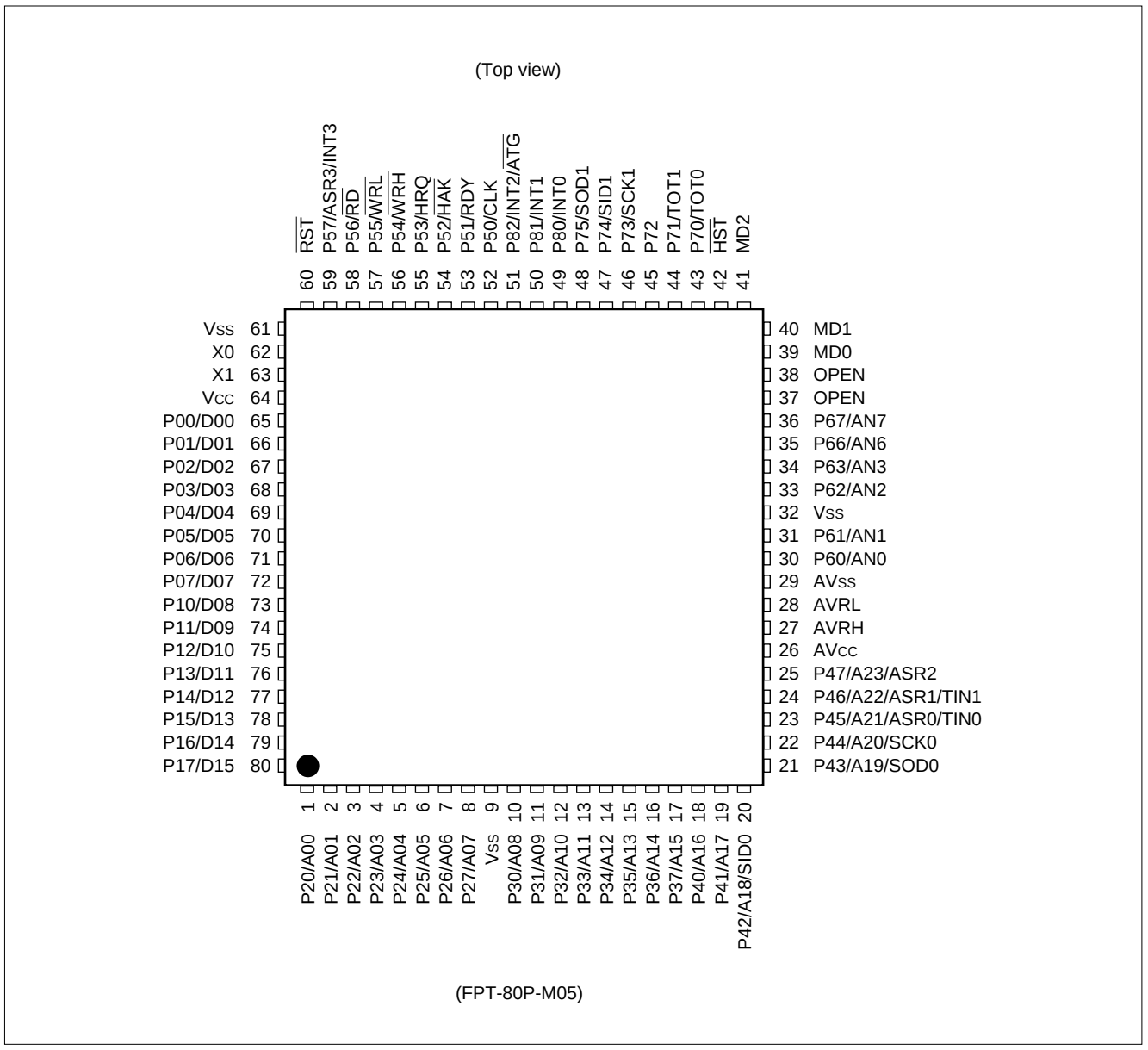
■ PRODUCT LINEUP

Parameter / Part number	MB90242A
Classification	Mass production device
CPU	F²MC-16F CPU core
DSP unit	Built-in
Internal RAM*	2 Kbytes
General-purpose ports	Max. 38 channels
A/D converter	10-bit resolution, analog inputs: 6 channels
D/A converter	None
UART	8 bits: 1 channel
8/16-bit serial I/O	8/16 bits: 1 channel Transfer direction switching function available
16-bit free-run timer	Built-in
16-bit input capture	4 channels
16-bit reload timer	2 channels
External interrupts	4 channels
Timebase timer	Built-in
Watchdog timer	Built-in
Clock gear function	Built-in
Package	FPT-80P-M05

* : The RAM has an extra 64-byte area reserved for multiply/accumulate operations.

MB90242A Series

■ PIN ASSIGNMENT



(FPT-80P-M05)

■ PIN DESCRIPTION

Pin no. LQFP*	Pin name	Circuit type	Function
1 to 8	P20 to P27	F	These pins cannot be used as general-purpose ports.
	A00 to A07		Output pins for the lower 8 bits of the external address bus
10 to 17	P30 to P37	F	These pins cannot be used as general-purpose ports.
	A08 to A15		Output pins for the middle 8 bits of the external address bus
18	P40	F	General-purpose I/O port This function is available when corresponding bit of the upper address control register specifies port.
	A16		External address bus output pin bit 16 This function is available when corresponding bit of the upper address control register specifies address.
19	P41	F	General-purpose I/O port This function is available when corresponding bit of the upper address control register specifies port.
	A17		External address bus output pin bit 17 This function is available when corresponding bit of the upper address control register specifies address.
20	P42	F	General-purpose I/O port This function is available when corresponding bit of the upper address control register specifies port.
	A18		External address bus output pin bit 18 This function is available when corresponding bit of the upper address control register specifies address.
	SID0		UART #0 data input pin This pin, as required, is used for input during UART #0 input operation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.
21	P43	F	General-purpose I/O port This function is available when data output of UART #0 is disabled and corresponding bit of the upper address control register specifies port.
	A19		External address bus output pin bit 19 This function is available when data output of UART #0 is disabled and corresponding bit of the upper address control register specifies address.
	SOD0		UART #0 data output pin This function is available when data output of UART #0 is enabled.

* : FPT-80P-M05

(Continued)

Pin no. LQFP*	Pin name	Circuit type	Function
22	P44	F	General-purpose I/O port This function is available when clock output of UART #0 and SSI #2 are disabled and corresponding bit of the upper address control register specifies port.
	A20		External address bus output pin bit 20 This function is available when clock output of UART #0 is disabled and corresponding bit of the upper address control register specifies address.
	SCK0		UART #0 clock input pin This function is available when the UART #0 clock output is enabled.
23	P45	F	General-purpose I/O port This function is available when data output of SSI #2 is disabled and corresponding bit of the upper address control register specifies port.
	A21		External address bus output pin bit 21 This function is available when data output of SSI #2 is disabled and corresponding bit of the upper address control register specifies address.
	ASR0		Input capature #0 data input pin This pin, as required, is used for input during input capture #0 input operation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.
	TIN0		16-bit timer #0 data input pin This pin, as required, is used for input during 16-bit timer #0 input operation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.
24	P46	F	General-purpose I/O port This function is available when corresponding bit of the upper address control register specifies port.
	A22		External address bus output pin bit 22 This function is available when corresponding bit of the upper address control register specifies address.
	ASR1		Input capature #1 data input pin This pin, as required, is used for input during input capture #1 input operation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.
	TIN1		16-bit timer #1 data input pin This pin, as required, is used for input during 16-bit timer #1 input operation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.

* : FPT-80P-M05

(Continued)

Pin no. LQFP*	Pin name	Circuit type	Function
25	P47	F	General-purpose I/O port This function is available when corresponding bit of the upper address control register specifies port.
	A23		External address bus output pin bit 23 This function is available when corresponding bit of the upper address control register specifies address.
	ASR2		Input capature #2 data input pin This pin, as required, is used for input during input capture #2 input operation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.
26	AV_{CC}	Power supply	Analog circuit power supply pin This power supply must only be turned on or off when electric potential of AV_{CC} or greater is applied to V_{CC}.
27	AVRH	Power supply	A/D converter external reference voltage input pin This pin must only be trendy on or off when electric potential of AVRH or greater is applied to AV_{CC}.
28	AVRL	Power supply	A/D converter external reference voltage input pin
29	AV_{SS}	Power supply	Analog circuit power supply (GND) pin
30, 31	P60, P61	H	N-ch open-drain I/O ports When corresponding bit of the ADER are set to “0,” reading data register with an instruction other than read-modify-write group instructions reads the level on these pins, while data written on the data register is output on these pins directly.
	AN0, AN1		A/D converter analog input pins Set corresponding bit of the ADER to “1,” and corresponding bit of the data register to “1.”
33, 34	P62, P63	H	N-ch open-drain I/O ports When corresponding bit of the ADER are set to “0,” reading data register with an instruction other than read-modify-write group instructions reads the level on these pins, while data written on the data register is output on these pins directly.
	AN2, AN3		A/D converter analog input pins Set corresponding bit of the ADER to “1,” and corresponding bit of the data register to “1.”
35, 36	P66, P67	H	N-ch open-drain I/O ports When corresponding bit of the ADER are set to “0,” reading data register with an instruction other than read-modify-write group instructions reads the level on these pins, while data written on the data register is output on these pins directly.
	AN6, AN7		A/D converter analog input pins Set corresponding bit of the ADER to “1,” and corresponding bit of the data register to “1.”
37, 38	OPEN	—	Open pins No internal connections are made.

* : FPT-80P-M05

(Continued)

Pin no. LQFP*	Pin name	Circuit type	Function
39 to 41	MD0 to MD2	C	Operating mode selection input pins Connect directly to V_{CC} or V_{SS}.
42	$\overline{HST}$	D	Hardware standby input pin
43, 44	P70, P71	F	General-purpose I/O ports This function is available when neither output of 16-bit timer #0 nor #1 is enabled.
	TOT0, TOT1		16-bit timer output pins This function is available when outputs of both 16-bit timer #0 and #1 are enabled.
45	P72	F	General-purpose I/O port
46	P73	F	General-purpose I/O port This function is available when clock output of SSI #1 is disabled.
	SCK1		SSI #1 clock I/O pin
47	P74	F	General-purpose I/O port This function is always valid.
	SID1		SSI #1 data input pin This pin, as required, is used for input during SSI #1 input operation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.
48	P75	F	General-purpose I/O port This function is available when data output of SSI #1 is disabled.
	SOD1		SSI #1 data output pin This function is available when data output of SSI #1 is enabled.
49, 50	P80, P81	G	General-purpose I/O ports This function is always valid.
	INT0, INT1		External interrupt input pins These pins, as required, are used for input while external interrupt is enabled, and it is necessary to disable input/output for other functions from these pins unless such input/output is made intentionally.
51	P82	F	General-purpose I/O port This function is always valid.
	INT2		External interrupt input pin This pin, as required, is used for input while external interrupt is enabled, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally. This pin is clamped to “LOW” level when CPU is in the “STOP” status. Use INT0 or INT1 to resume operation.
	$\overline{ATG}$		A/D converter activation trigger input pin This pin, as required, is used for input while A/D converter is waiting for activation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.

* : FPT-80P-M05

(Continued)

Pin no. LQFP*	Pin name	Circuit type	Function
52	P50	F	General-purpose I/O port This function is available when CLK output is disabled.
	CLK		CLK output pin This function is available when CLK output is enabled.
53	P51	E	General-purpose I/O port This function is available when ready function is disabled.
	RDY		Ready input pin This function is available when ready function is enabled.
54	P52	E	General-purpose I/O port This function is available when hold function is disabled.
	$\overline{\text{HAK}}$		Hold acknowledge output pin This function is available when hold function is enabled.
55	P53	E	General-purpose I/O port This function is available when hold function is disabled.
	HRQ		Hold request input pin This function is available when hold function is enabled.
56	P54	F	General-purpose I/O port This function is available when the external bus 8-bit mode is selected or $\overline{\text{WRH}}$ pin output is disabled.
	$\overline{\text{WRH}}$		Write strobe output pin for the upper eight bits of the data bus This function is available when the external bus 16-bit mode is selected and $\overline{\text{WRH}}$ pin output is enabled.
57	P55	F	General-purpose I/O port This function is available when $\overline{\text{WRL}}$ pin output is disabled.
	$\overline{\text{WRL}}$		Write strobe output pin for the lower eight bits of the data bus This function is available when $\overline{\text{WRL}}$ pin output is enabled.
58	P56	F	This pin cannot be used as a general-purpose port.
	$\overline{\text{RD}}$		Read strobe output pin for the data bus
59	P57	F	General-purpose I/O port
	ASR3		Input capture #3 data input pin This pin, as required, is used for input during input capture #3 input operation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.
	INT3		External interrupt #3 data input pin This pin, as required, is used for input during external interrupt #3 input operation, and it is necessary to disable input/output for other functions from this pin unless such input/output is made intentionally.
60	$\overline{\text{RST}}$	B	External reset request input pin
62, 63	X0, X1	A	Crystal oscillator pins (32 MHz)

* : FPT-80P-M05

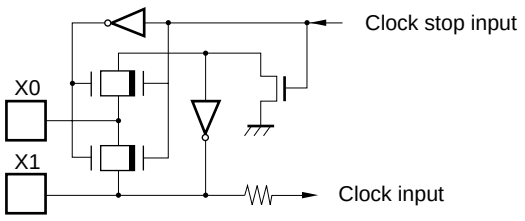
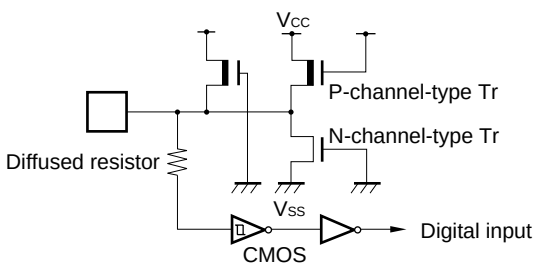
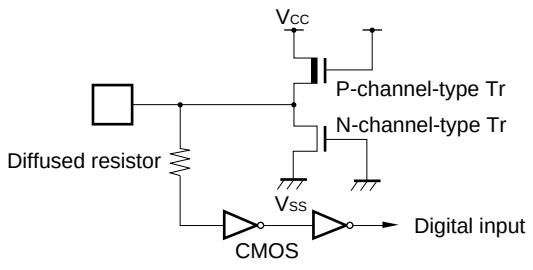
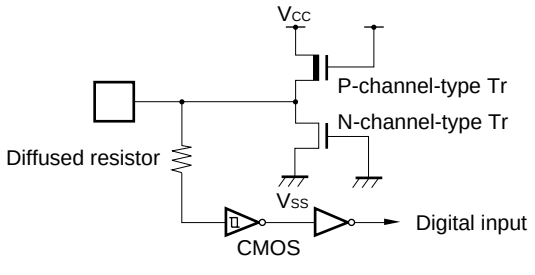
(Continued)

(Continued)

Pin no. LQFP*	Pin name	Circuit type	Function
64	V_{CC}	Power supply	Digital circuit power supply pin
9, 32, 61	V_{SS}	Power supply	Digital circuit power supply (GND) pins
65 to 72	P00 to P07	E	These pins cannot be used as general-purpose ports.
	D00 to D07		I/O pins for the lower 8 bits of the external data bus
73 to 80	P10 to P17	E	General-purpose I/O ports This function is available when the external bus 8-bit mode is selected.
	D08 to D15		I/O pins for the upper 8 bits of the external data bus This function is available when the 16-bit bus mode is selected.

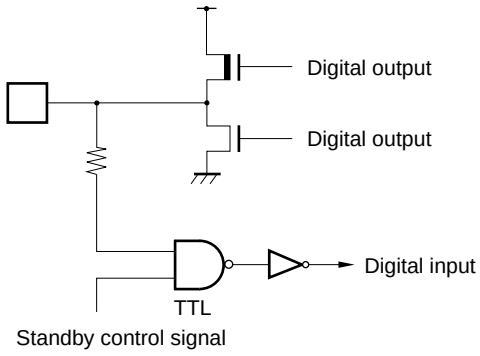
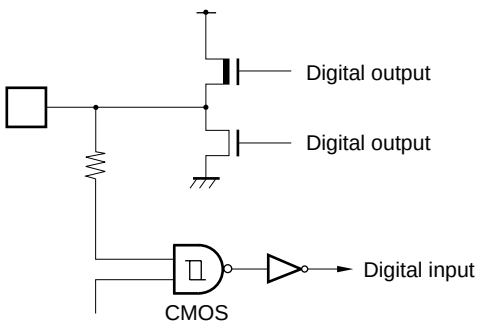
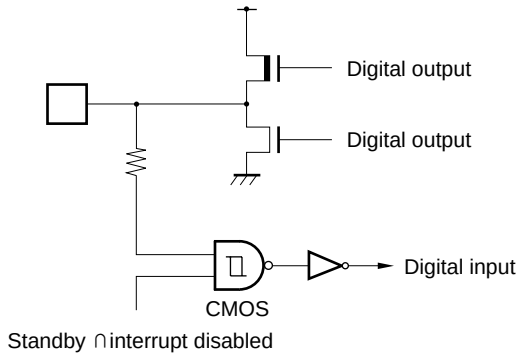
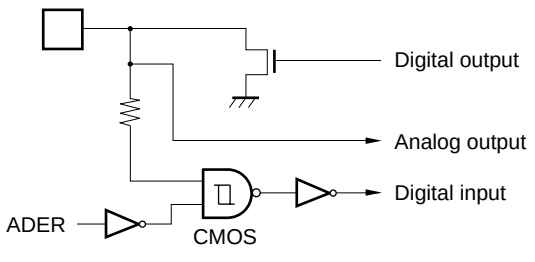
* : FPT-80P-M05

■ I/O CIRCUIT TYPE

Type	Circuit	Remarks
A		• 32 MHz • Oscillation feedback resistor: approx. 1 MΩ
B		• CMOS-level hysteresis input (without standby control) Pull-up resistor: approx. 50 kΩ
C		• CMOS-level input (without standby control)
D		• CMOS-level hysteresis input (without standby control)

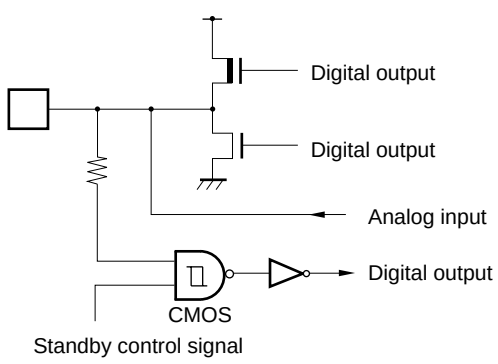
(Continued)

MB90242A Series

Type	Circuit	Remarks
E		• CMOS-level output • TTL-Level input (with standby control)
F		• CMOS-level input CMOS-level hysteresis input (with standby control)
G		• CMOS-level output CMOS-level hysteresis input Standby control (when interrupt disabled) available
H		• N-ch open-drain CMOS-level output CMOS-level hysteresis input Analog input (with analog input control)

(Continued)

(Continued)

Type	Circuit	Remarks
I		• CMOS-level input Analog input • CMOS-level hysteresis input (with standby control)

■ HANDLING DEVICES

1. Preventing Latchup

Latchup may occur on CMOS ICs if voltage higher than V_{CC} or lower than V_{SS} is applied to the input or output pins other than medium-and high voltage pins or if higher than the voltage is applied between V_{CC} and V_{SS}.

When latchup occurs, power supply current increases rapidly might thermally damage elements. When using, take great care not to exceed the absolute maximum ratings.

In addition, for the same reasons take care to prevent the analog power supply from exceeding the digital power supply.

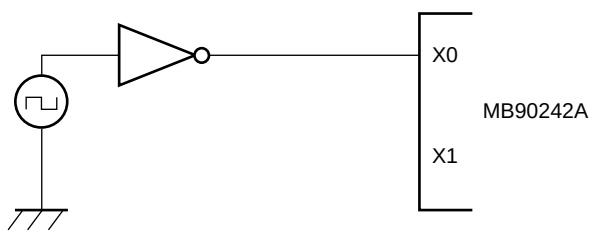
2. Treatment of Unused Pins

Leaving unused input pins open could cause malfunctions. They should be connected to a pull-up or pull-down resistors.

3. Precautions when Using an External Clock

When an external clock is used, drive X0 only and X1 should be left open.

- **Using an External Clock**



4. Power Supply Pins

When there are several V_{CC} and V_{SS} pins, those pins that should have the same electric potential are connected within the device when the device is designed in order to prevent misoperation, such as latchup. However, all of those pins must be connected to the power supply and ground externally in order to reduce unnecessary emissions, prevent misoperation of strobe signals due to an increase in the ground level, and to observe the total output current standards.

In addition, give a due consideration to the connection in that current supply be connected to V_{CC} and V_{SS} with the lowest possible impedance.

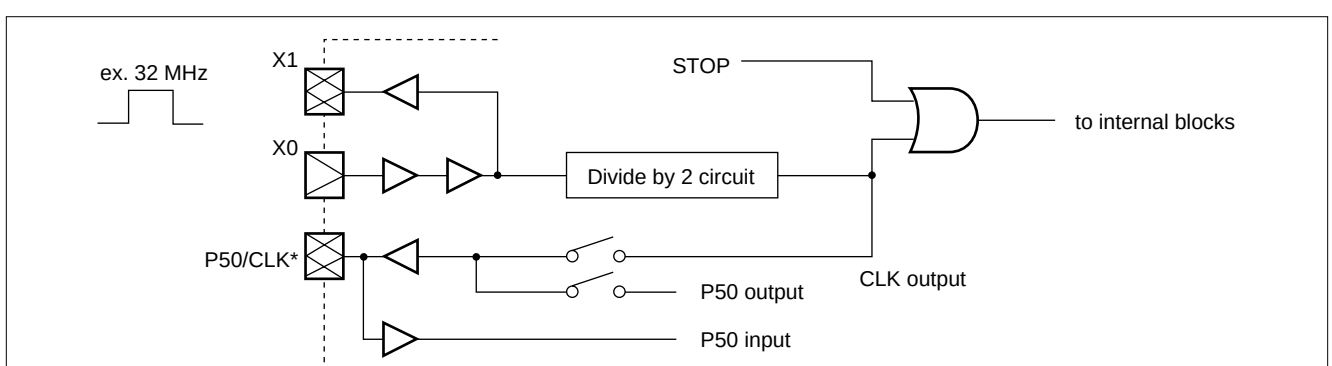
Finally, it is recommended to connect a capacitor of about 0.1 µF between V_{CC} and V_{SS} near this device as a bypass capacitor.

5. Crystal Oscillation Circuit

Noise in the vicinity of the X0 and X1 pins will cause this device to operate incorrectly. Design the printed circuit board so that the bypass capacitor connecting X0, X1 and the crystal oscillator (or ceramic oscillator) to ground is located as close to the device as possible.

In addition, because printed circuit board artwork in which the area around the X0 and X1 pins is surrounded by ground provides stable operation, such an arrangement is strongly recommended.

6. CLK Pin



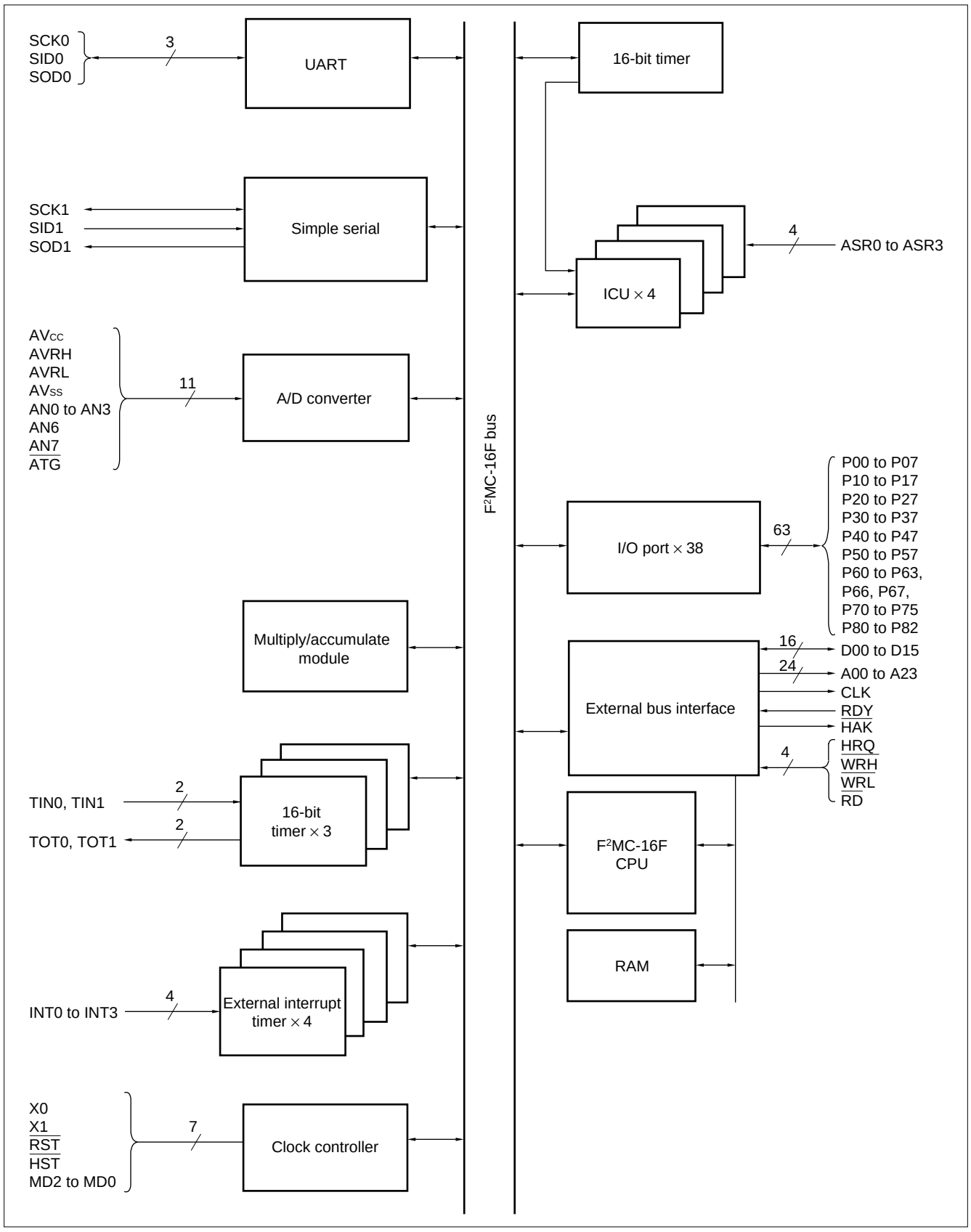
* : In the external bus mode, the P50/CLK pin is initially configured as a CLK output pin.

7. Cautions in Applying Power Supply

Hold the $\overline{\text{HST}}$ pin to the “H” level when applying power supply.
When the $\overline{\text{RST}}$ pin is in the “L” level, do not hold the $\overline{\text{HST}}$ pin to “L” level.

MB90242A Series

■ BLOCK DIAGRAM



■ ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings

(V_{SS} = AV_{SS} = 0.0 V)

Parameter	Symbol	Value		Unit	Remarks
		Min.	Max.		
Power supply voltage	V_{CC}	V_{SS} – 0.3	V_{SS} + 7.0	V	
	AV_{CC}	V_{CC} – 0.3	V_{CC} + 7.0	V	
Input voltage	V_I*	V_{SS} – 0.3	V_{CC} + 0.3	V	
Output voltage	V_O*	V_{SS} – 0.3	V_{CC} + 0.3	V	
“L” level output current	I_{OL}	—	10	mA	
“L” level average output current	I_{OLAV}	—	4	mA	
“L” level total average output current	ΣI_{OLAV}	—	50	mA	
“H” level output current	I_{OH}	—	–10	mA	
“H” level average output current	I_{OHAV}	—	–4	mA	
“H” level total average output current	ΣI_{OHAV}	—	–48	mA	
Power consumption	P_D	—	600	mW	
Operating temperature	T_A	–30	+70	°C	
Storage temperature	T_{stg}	–55	+150	°C	

* : V_I and V_O must not exceed V_{CC} + 0.3 V.

WARNING: Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc.) in excess of absolute maximum ratings. Do not exceed these ratings.

2. Recommended Operating Conditions

(V_{SS} = AV_{SS} = 0.0 V)

Parameter	Symbol	Value		Unit	Remarks
		Min.	Max.		
Power supply voltage	V_{CC}	4.5	5.5	V	
		2.0	5.5	V	For retaining RAM data in the stop mode
Operating temperature	T_A	–30	+70	°C	External bus mode

WARNING: Recommended operating conditions are normal operating ranges for the semiconductor device. All the device’s electrical characteristics are warranted when operated within these ranges.

Always use semiconductor devices within the recommended operating conditions. Operation outside these ranges may adversely affect reliability and could result in device failure.

No warranty is made with respect to uses, operating conditions, or combinations not represented on the data sheet. Users considering application outside the listed conditions are advised to contact their FUJITSU representative beforehand.

3. DC Characteristics

(V_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value			Unit	Remarks
				Min.	Typ.	Max.		
"H" level input voltage	V_{IH1}	—	—	0.7 V_{CC}	—	V_{CC} + 0.3	V	CMOS input
	V_{IH2}	—	—	2.2	—	V_{CC} + 0.3	V	TTL input
	V_{IHIS}	—	—	0.8 V_{CC}	—	V_{CC} + 0.3	V	Hysteresis input
	V_{IHM}	MD0 to MD2	—	V_{CC} – 0.3	—	V_{CC} + 0.3	V	
"L" level input voltage	V_{IL1}	—	—	V_{SS} – 0.3	—	0.3 V_{CC}	V	CMOS input
	V_{IL2}	—	—	V_{SS} – 0.3	—	0.8	V	TTL input
	V_{ILIS}	—	—	V_{SS} – 0.3	—	0.2 V_{CC}	V	Hysteresis input
	V_{ILM}	MD0 to MD2	—	V_{SS} – 0.3	—	V_{SS} + 0.3	V	
"H" level output voltage	V_{OH}	All ports except P60 to P63, P66, P67	V_{CC} = 4.5 V I_{OH} = –4.0 mA	V_{CC} – 0.5	—	—	V	
"L" level output voltage	V_{OL}	All ports	V_{CC} = 4.5 V I_{OL} = 4.0 mA	—	—	0.4	V	
"H" level input current	I_{IH1}	Except $\overline{RST}$	V_{CC} = 5.5 V V_{IH} = 0.7 V_{CC}	—	—	–10	µA	CMOS input
	I_{IH2}	—	V_{CC} = 5.5 V V_{IH} = 2.2 V	—	—	–10	µA	TTL input
	I_{IH3}	—	V_{CC} = 5.5 V V_{IH} = 0.8 V_{CC}	—	—	–10	µA	Hysteresis input
"L" level input current	I_{IL1}	Except $\overline{RST}$	V_{CC} = 5.5 V V_{IL} = 0.3 V_{CC}	—	—	10	µA	CMOS input
	I_{IL2}	—	V_{CC} = 5.5 V V_{IL} = 0.8 V_{CC}	—	—	10	µA	TTL input
	I_{IL3}	—	V_{CC} = 5.5 V V_{IL} = 0.2 V_{CC}	—	—	10	µA	Hysteresis input
Pull-up resistor	R_{PULL}	$\overline{RST}$	V_{CC} = 5.0 V	22	—	110	kΩ	
Power supply current	I_{CC}	V_{CC}	V_{CC} = 5.0 V ±10% F_C = 32 MHz	—	80	100	mA	In operation mode
	I_{CCS}	V_{CC}	V_{CC} = 5.0 V ±10% F_C = 32 MHz In sleep mode	—	30	50	mA	
	I_{CCH}	V_{CC}	V_{CC} = 5.0 V ±10% T_A = +25°C In stop mode	—	0.1	10	µA	
Input capacitance	C_{IN}	Except V_{CC}, V_{SS}	—	—	10	—	pF	
Open-drain output leakage current	I_{LEAK}	P60 to P63, P66, P67	—	—	0.1	10	µA	

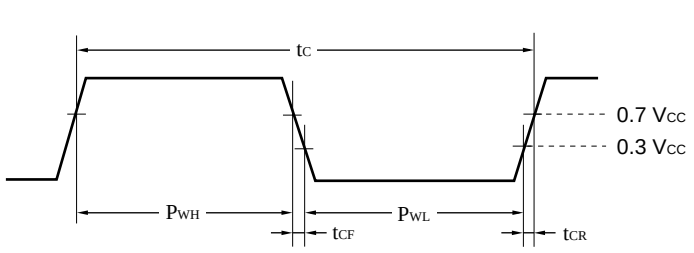
4. AC Characteristics

(1) Clock Timing

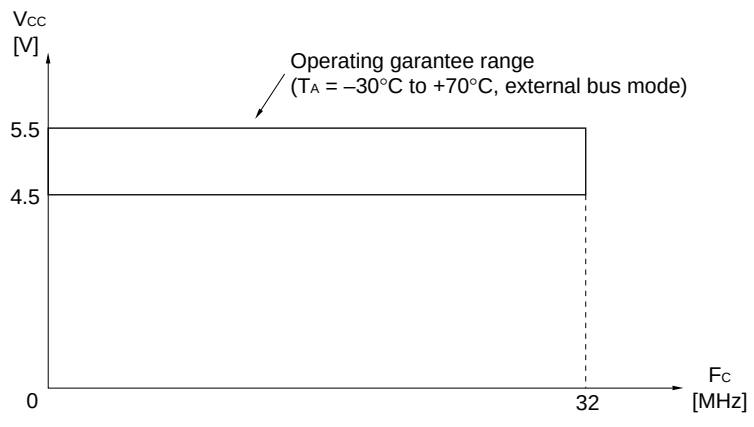
(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Clock frequency	F_C	X0 X1	—	—	32	MHz	
Clock cycle time	t_C	X0 X1	—	1/F_C	—	ns	
Input clock pulse width	P_{WH} P_{WL}	X0	—	10	—	ns	
Input clock rising/ falling time	t_{CR} t_{CF}	X0	—	—	8	ns	

- **Clock Timing**



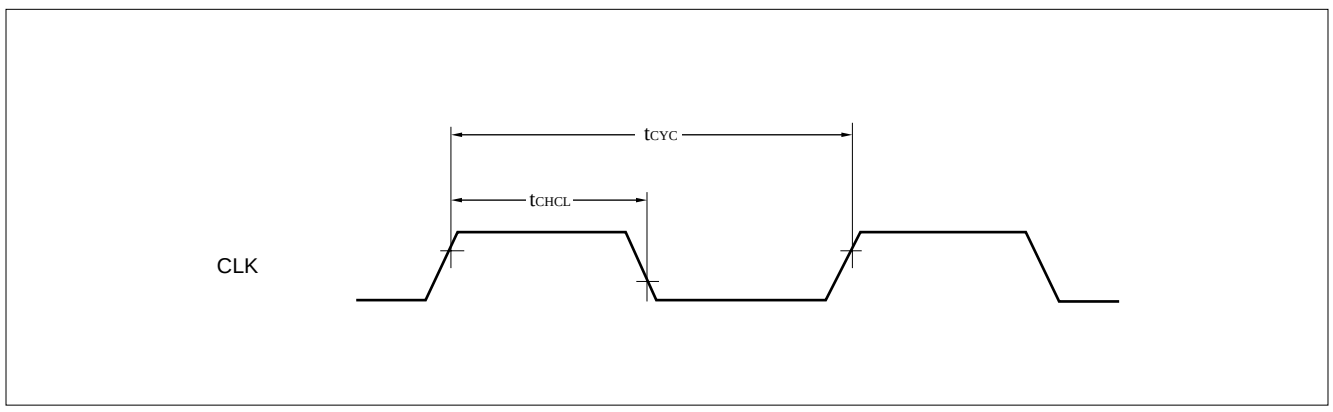
- **Relationship between Clock Frequency and Supply Voltage**



(2) Clock Output Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Machine cycle time	t_{CYC}	CLK	—	$t_C \times 2$	—	ns	
CLK ↑ → CLK↓	t_{CHCL}	CLK	—	$t_{CYC}/2 - 20$	$t_{CYC}/2$	ns	

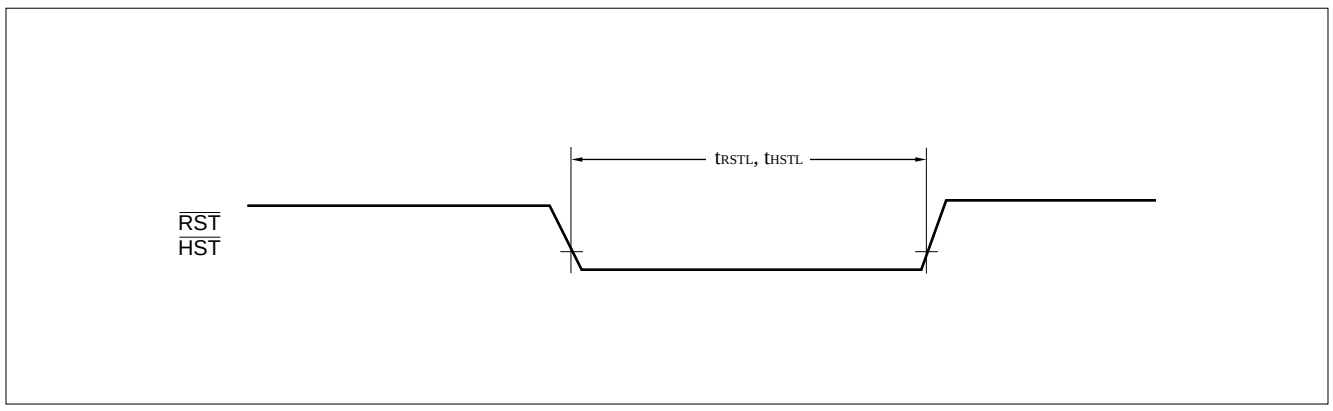


(3) Reset and Hardware Standby Input

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Reset input time	t_{RSTL}	$\overline{RST}$	—	$t_{CYC} \times 5$	—	ns	
Hardware standby input time	t_{HSTL}	$\overline{HST}$	—	$t_{CYC} \times 5$	—	ns	

Note: The machine cycle time (t_{CYC}) at hardware standby is set to 1/32 divided oscillation.

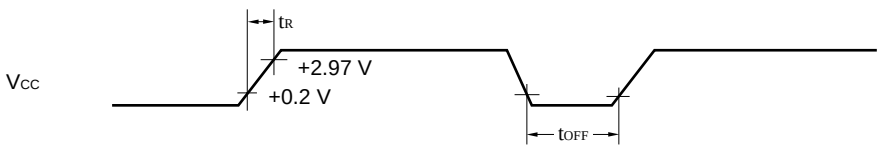


(4) Power-on Reset

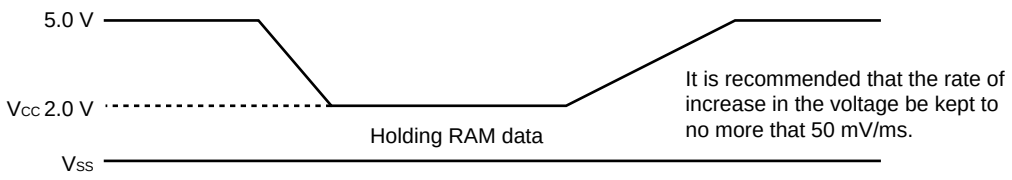
(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Power supply rising time	t_R	V_{CC}	—	—	30	ms	V_{CC} must be lower than 0.2 V before power is applied.
Power supply cut-off time	t_{OFF}	V_{CC}	—	1	—	ms	

Note: The above standards are the values needed in order to activate a power-on reset.



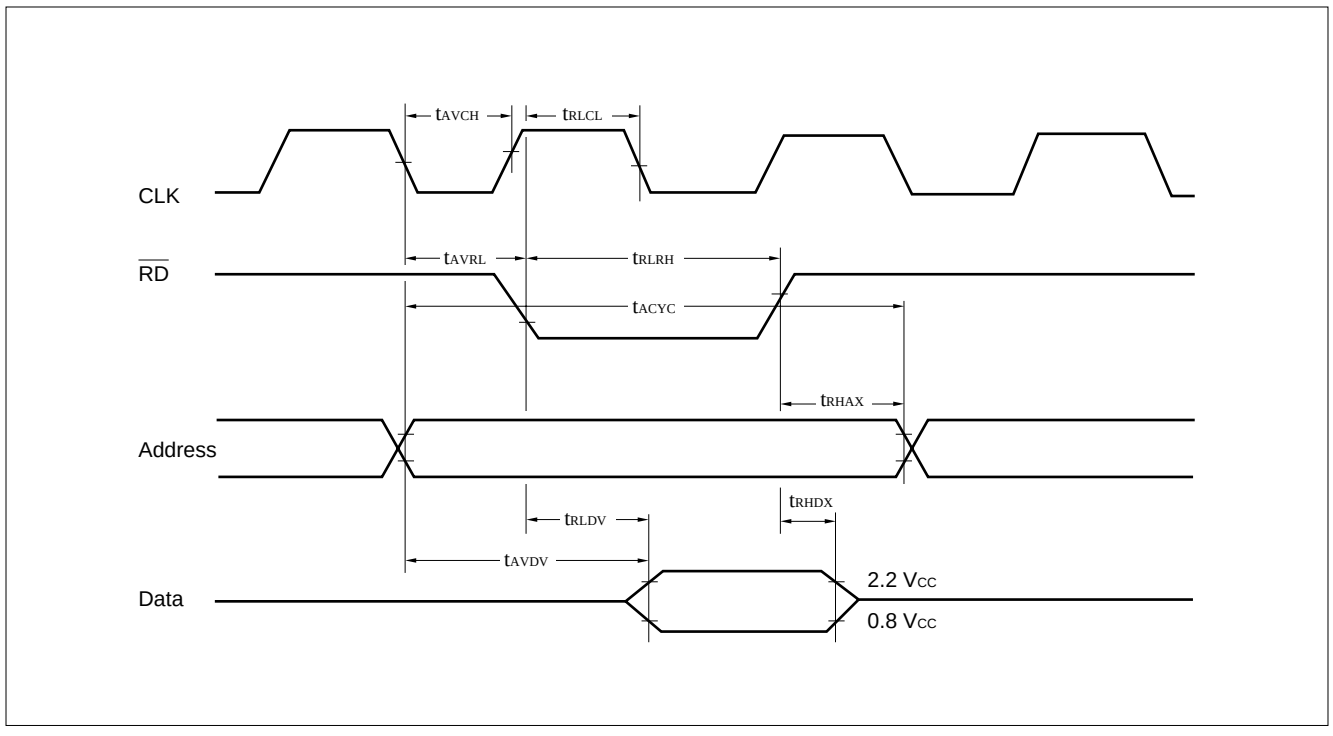
If power supply voltage needs to be changed in the course of operation, a smooth voltage rise is recommended by suppressing the voltage variation as shown below.



(5) Bus Read Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

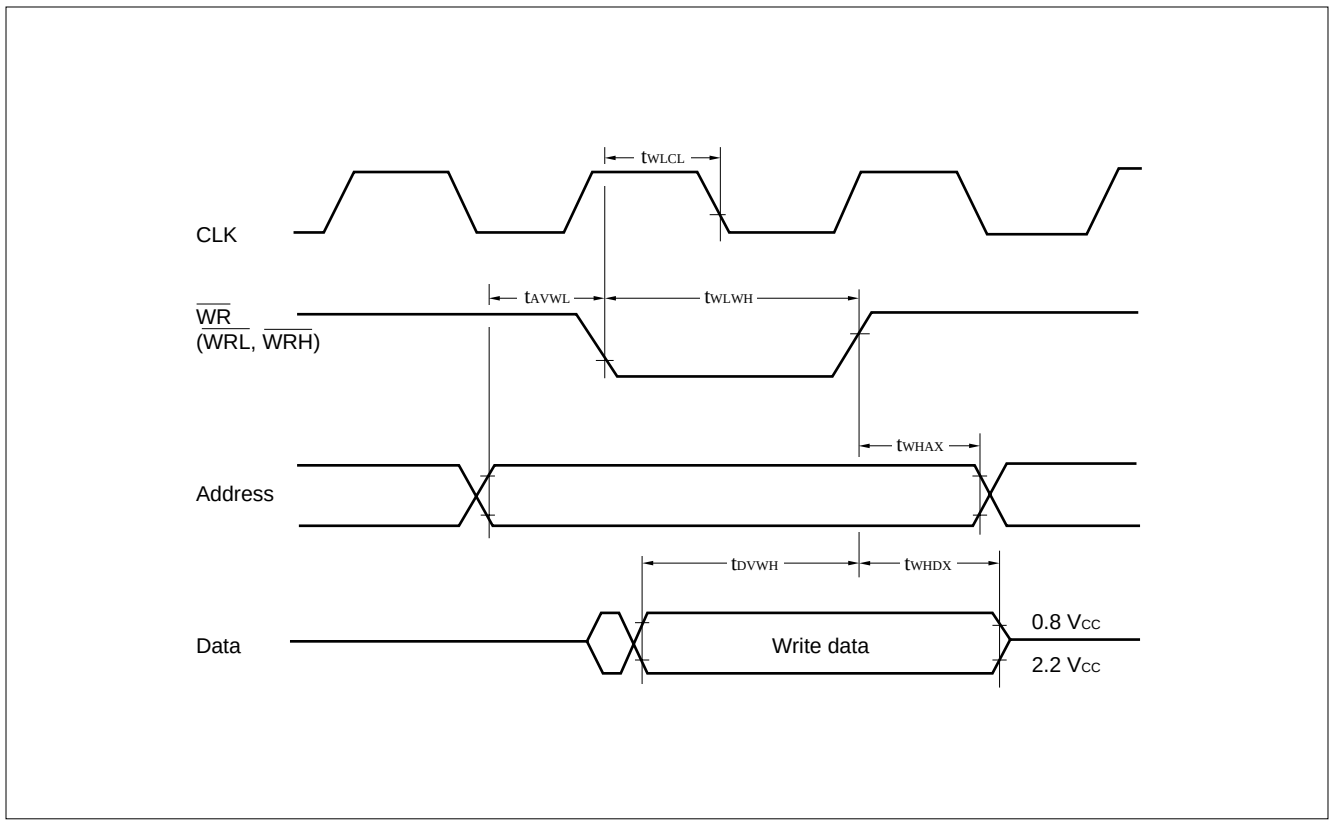
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Address cycle time	t_{ACYC}	Address	—	2 t_{CYC} – 10	—	ns	
Valid address → $\overline{RD}$ ↓ time	t_{AVRL}	Address	—	t_{CYC}/2 – 15	—	ns	
$\overline{RD}$ pulse width	t_{RLRH}	$\overline{RD}$	—	t_{CYC} – 25	—	ns	
$\overline{RD}$ ↓ → Valid data input	t_{RLDV}	D00 to D15	—	—	t_{CYC}/ – 30	ns	
$\overline{RD}$ ↑ → data hold time	t_{RHDX}		—	0	—	ns	
Valid address → Valid data input	t_{AVDV}		—	—	3 t_{CYC}/2 – 40	ns	
$\overline{RD}$ ↑ → Address valid time	t_{RHAX}	Address	—	t_{CYC}/2 – 20	—	ns	
Valid address → CLK ↑ time	t_{AVCH}	Address CLK	—	t_{CYC}/2 – 25	—	ns	
$\overline{RD}$ ↓ → CLK ↓ time	t_{RLCL}	$\overline{RD}$, CLK	—	t_{CYC}/2 – 25	—	ns	



(6) Bus Write Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Valid address → $\overline{WR}$ ↓ time	t_{AVWL}	Address	—	$t_{CYC}/2 - 15$	—	ns	
$\overline{WR}$ pulse width	t_{WLWH}	$\overline{WRL}$, $\overline{WRH}$	—	$t_{CYC} - 25$	—	ns	
Write data → $\overline{WR}$ ↑ time	t_{DVWH}	D00 to D15	—	$t_{CYC} - 40$	—	ns	
$\overline{WR}$ ↑ → Data hold time	t_{WHDX}	D00 to D15	—	$t_{CYC}/2 - 15$	—	ns	
$\overline{WR}$ ↑ → Address invalid time	t_{WHAX}	Address	—	$t_{CYC}/2 - 15$	—	ns	
$\overline{WR}$ ↓ → CLK ↑ time	t_{WLCL}	$\overline{WRL}$, $\overline{WRH}$, CLK	—	$t_{CYC}/2 - 25$	—	ns	

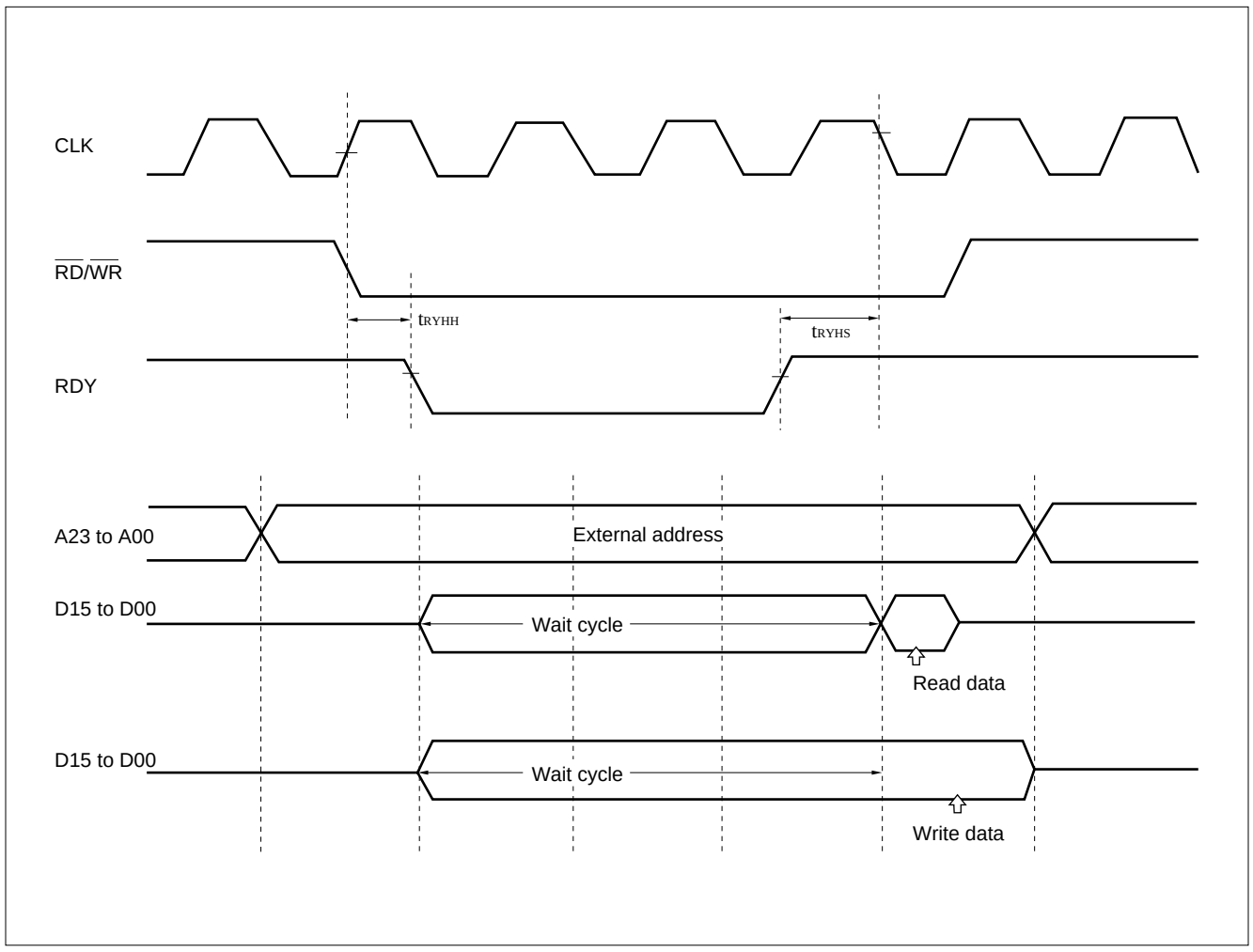


(7) Ready Input Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
RDY setup time	t_{RYHS}	RDY	At 32 MHz oscillation	15	60	ns	
RDY hold time	t_{RYHH}	RDY		0	60	ns	

Note: If the setup time of RDY on a falling edge is insufficient, use the auto ready function.

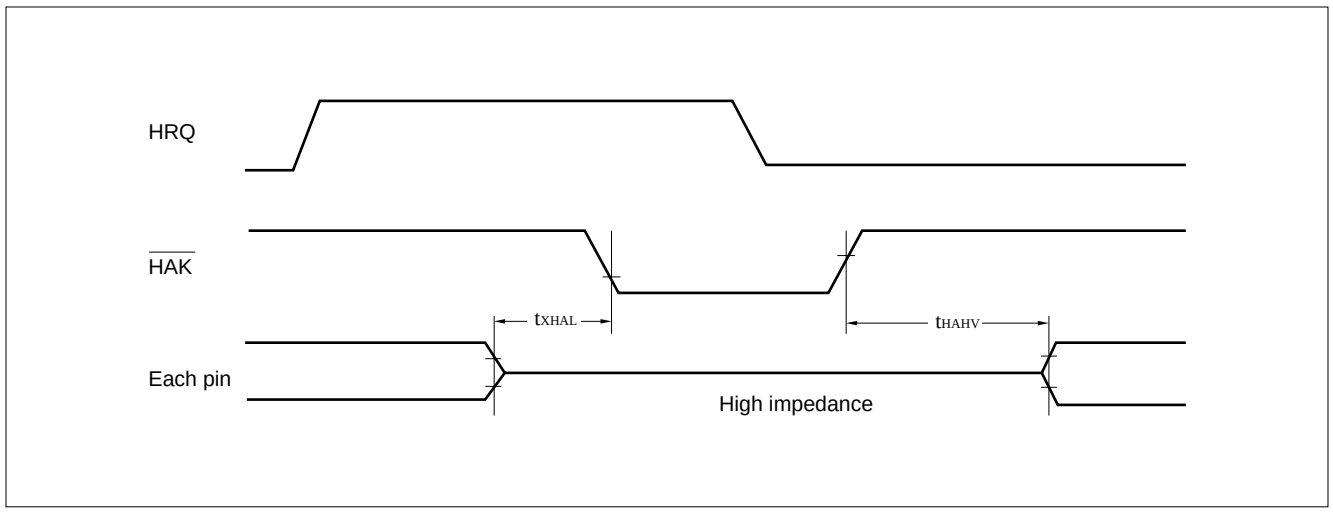


(8) Hold Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Pin floating → $\overline{HAK}$ ↓ time	t_{XHAL}	$\overline{HAK}$	—	30	t_{CYC}	ns	
$\overline{HAK}$ ↑ time → Pin valid time	t_{HAHV}	$\overline{HAK}$	—	t_{CYC}	2 t_{CYC}	ns	

Note: At least one cycle is required from the time when HRQ is fetched until $\overline{HAK}$ changes.



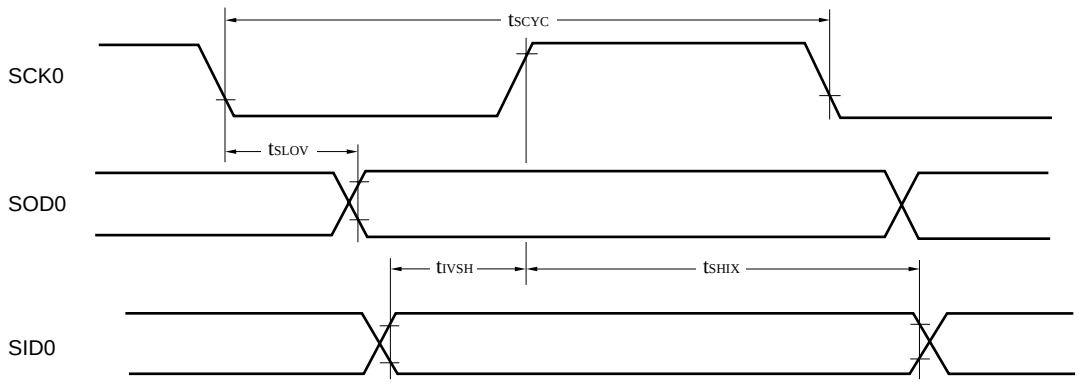
(9) UART Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

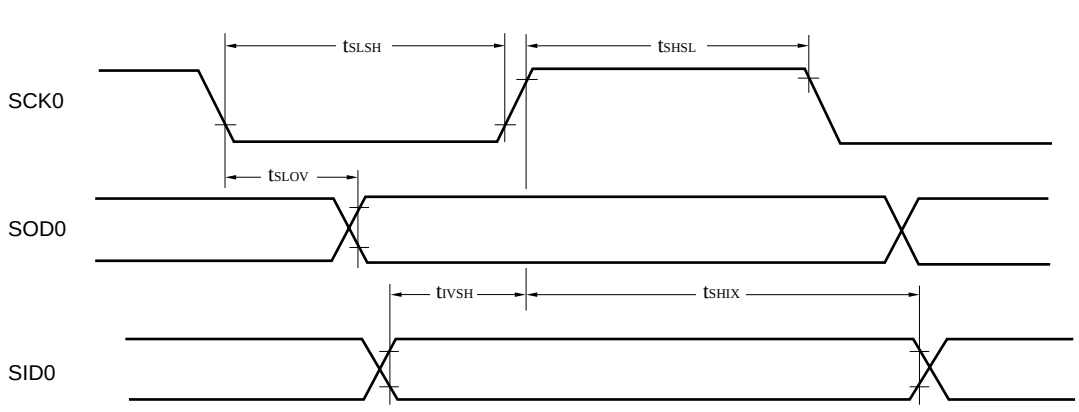
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Serial clock cycle time	t_{SCYC}	—	—	8 t_{CYC}	—	ns	For internal shift clock mode output pin, C_L = 80 pF
SCK ↓ → SOD delay time	t_{SLOV}	—	—	–80	80	ns	
Valid SID → SCK ↑	t_{IVSH}	—	—	100	—	ns	
SCK ↑ → Valid SID hold time	t_{SHIX}	—	—	60	—	ns	
Serial clock “H” pulse width	t_{SHSL}	—	—	4 t_{CYC}	—	ns	For external shift clock mode output pin, C_L = 80 pF
Serial clock “L” pulse width	t_{SLSH}	—	—	4 t_{CYC}	—	ns	
SCK ↓ → SOD delay time	t_{SLOV}	—	—	—	150	ns	
Valid SID → SCK ↑	t_{IVSH}	—	—	60	—	ns	
SCK ↑ → Valid SID hold time	t_{SHIX}	—	—	60	—	ns	

Notes: • These are the AC characteristics for CLK synchronous mode.
• C_L is the load capacitance added to pins during testing.
• t_{CYC} is the machine cycle time (unit: ns).

- **Internal Shift Clock Mode**



- **External Shift Clock Mode**



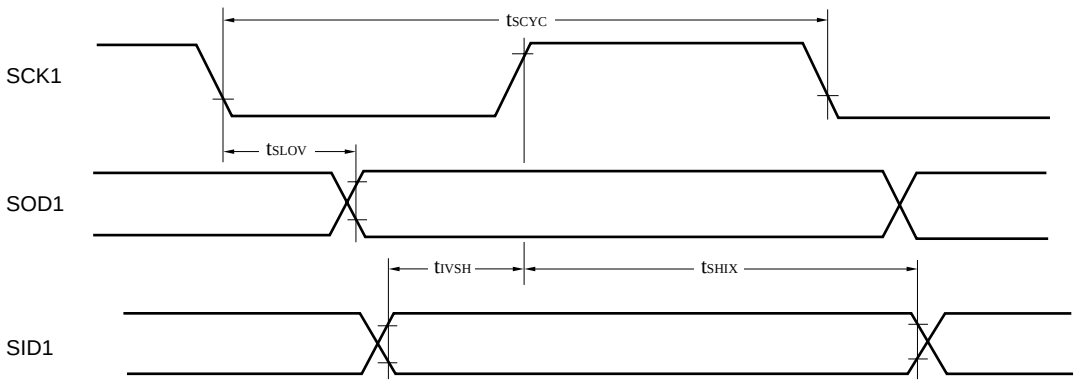
(10) Simple Serial Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Serial clock cycle time	t_{SCYC}	—	—	2 t_{CYC}	—	ns	For operation output pin, C_L = 80 pF
SCK ↓ → SOD delay time	t_{SLOV}	—	—	—	t_{CYC}/2	ns	
Valid SID → SCK ↑	t_{IVSH}	—	—	1 t_{CYC}	—	ns	
SCK ↑ → Valid SID hold time	t_{SHIX}	—	—	1 t_{CYC}	—	ns	

Notes: • C_L is the load capacitance added to pins during testing.
• t_{CYC} is the machine cycle time (unit: ns).

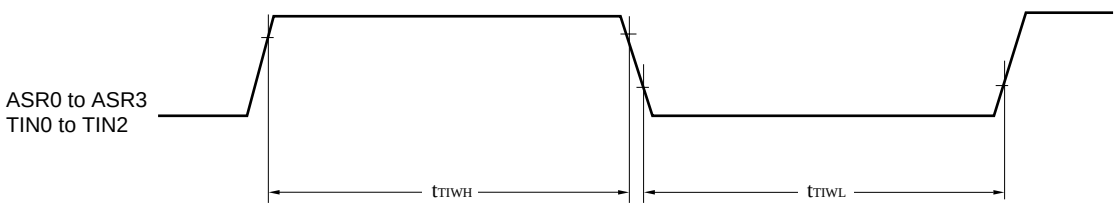
- **Internal Shift Clock Mode**



(11) Timer Input Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

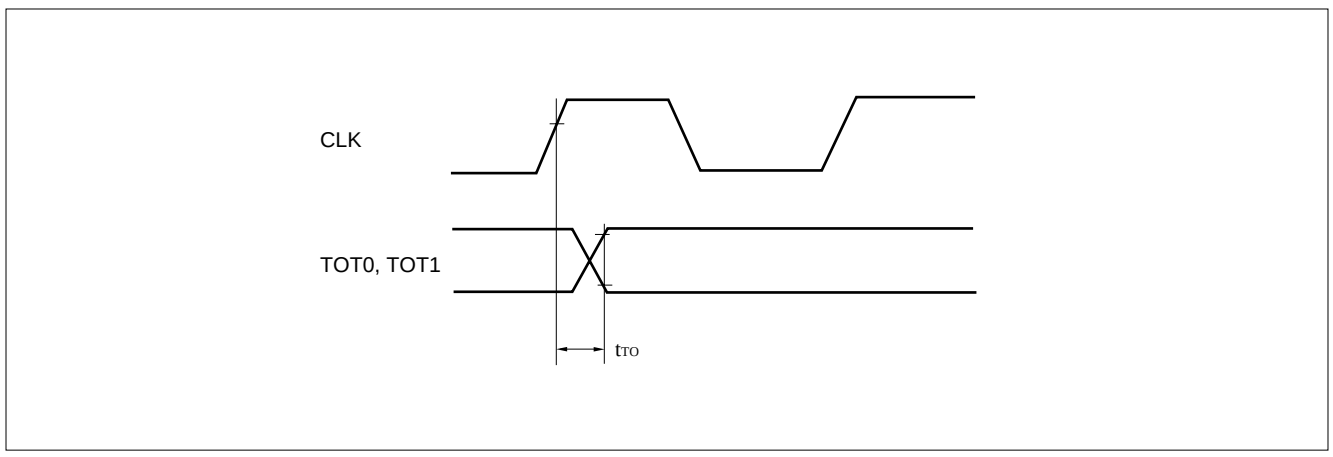
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Input pulse width	t_{TIWH} t_{TIWL}	ASR0 to ASR3, TIN0 to TIN2	—	4 t_{CYC}	—	ns	



(12) Timer Ouput Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

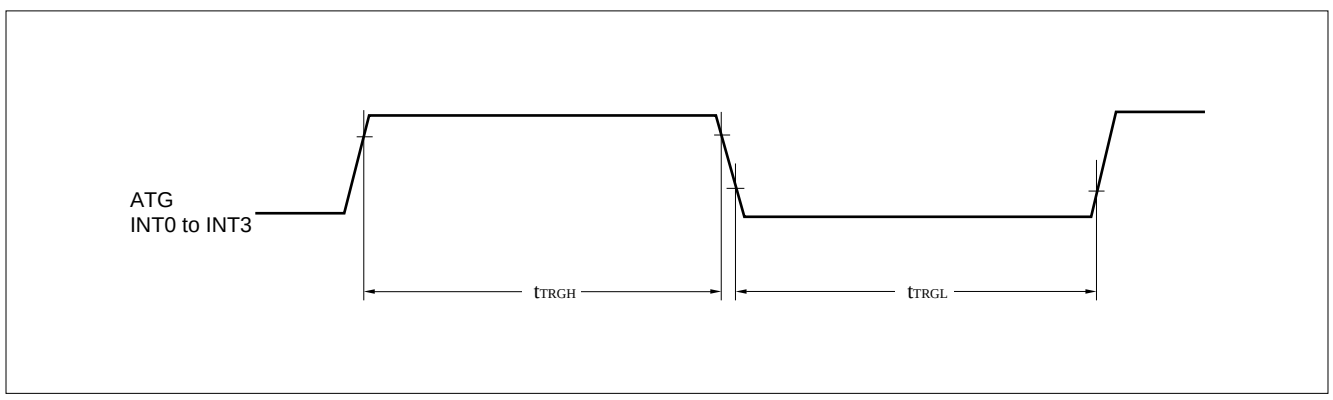
Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
SCK ↑ → Change time	t_{TO}	TOT0, TOT1	V_{CC} = 5.0 V ±10%	—	40	ns	



(13) Trigger Input Timing

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

Parameter	Symbol	Pin name	Condition	Value		Unit	Remarks
				Min.	Max.		
Input pulse width	t_{TRGH} t_{TRGL}	$\overline{ATG}$, INT0 to INT3	—	5 t_{CYC}	—	ns	



5. A/D Converter Electrical Characteristics

(V_{CC} = 5.0 V ±10%, V_{SS} = 0.0 V, T_A = –30°C to +70°C)

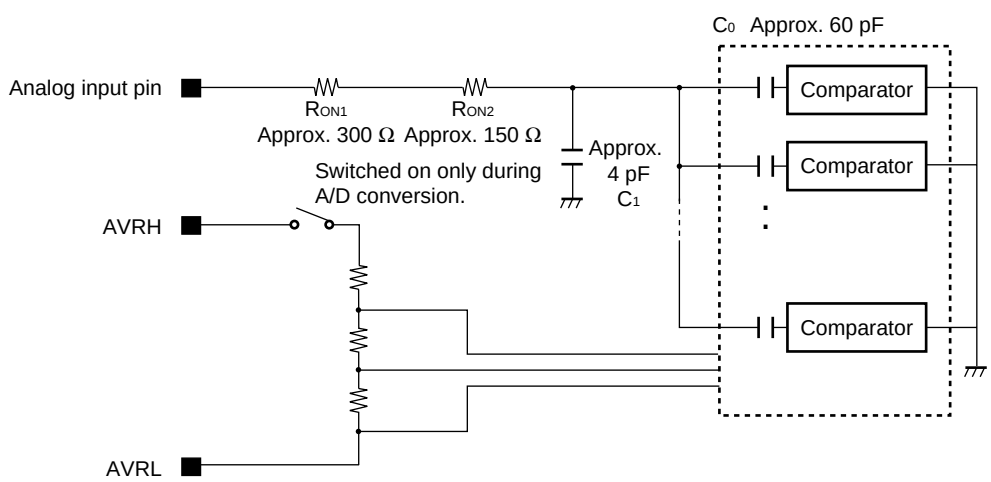
Parameter	Symbol	Pin name	Value			Unit	Remarks
			Min.	Typ.	Max.		
Resolution	—	—	—	8, 10	10	bit	
Total error	—	—	—	—	±3.0	LSB	
Linearity error	—	—	—	—	±2.0	LSB	
Differential linearity error	—	—	—	—	±1.9	LSB	
Zero transition voltage	V_{Ot}	AN0 to AN3 AN6, AN7	AVRL – 1.0	AVRL + 1.0	AVRL + 3.0	LSB	
Full-scale transition voltage	V_{FST}	AN0 to AN3 AN6, AN7	AVRH – 4.0	AVRH – 1.0	AVRH + 1.0	LSB	
Conversion time	—	—	1.25	—	—	µs	Specified by the ADCT register settings.*1 V_{CC} = 5.0 V±10%
Sampling period	—	—	560	—	—	ns	
Conversion period a	—	—	125	—	—	ns	
Conversion period b	—	—	125	—	—	ns	
Conversion period c	—	—	250	—	—	ns	
Analog port input current	I_{AIN}	AN0 to AN3 AN6, AN7	—	0.1	3	µA	
Analog input voltage	—	AN0 to AN3 AN6, AN7	AVRL	—	AVRH	V	
Reference voltage	—	AVRH	AVRL + 2.7	—	AV_{CC}	V	AVRH – AVRL ≥ 2.7
	—	AVRL	0	—	AVRH – 2.7	V	
Power supply current	I_A	AV_{CC}	—	15	20	mA	
	I_{AS}*2		—	—	5	µA	AV_{CC} = 5.5 V in stop mode
Reference voltage supply current	I_R	AVRH	—	1.5	2	mA	
	I_{RS}*2		—	—	5	µA	AV_{CC} = 5.5 V in stop mode
Interchannel disparity	—	AN0 to AN3 AN6, AN7	—	—	4	LSB	

*1: When F_C = 32 MHz, and the machine cycle is 62.5 ns.

*2: I_{AS} and I_{RS} are current when the A/D converter is not operating and the CPU is stopped.

Notes:
- The smaller | AVRH – AVRL |, the greater the error would become relatively.
- If the output impedance of the external circuit of an analog input is too high, an analog voltage sampling time might be insufficient. When the sampling period close to the minimum value is used, the output impedance of the external circuit should be less than approximately 300 Ω.

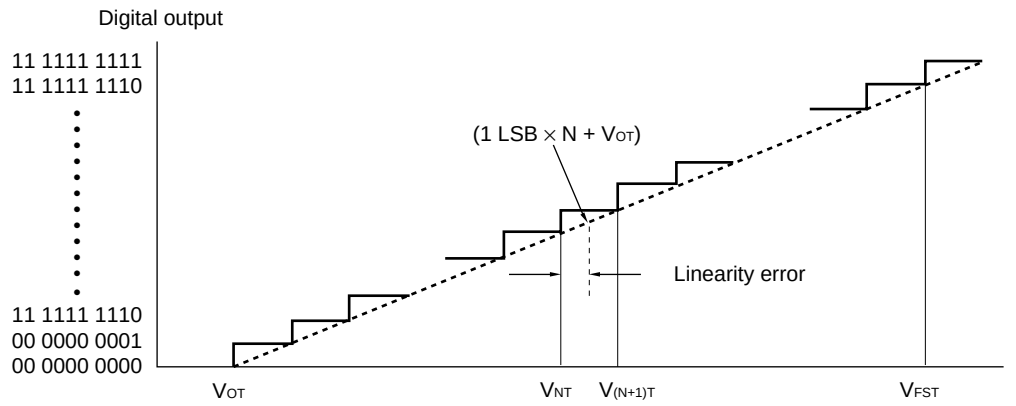
- **Analog Input Circuit Model Diagram**



Note: Use the values shown as guides only.

6. A/D Converter Glossary

- Resolution
  Analog changes that are identifiable with the A/D converter.
  If the resolution is 10 bits, the analog voltage can be resolved into 2^{10}.
- Total error
  The difference between theoretical and actual conversion values caused by the zero transition error, full-scale transition error, non-linearity error, differential linearity error, and noise.
- Linearity error
  The deviation of the straight line connecting the zero transition point (“00 0000 0000” ↔ “00 0000 0001”) with the full-scale transition point (“11 1111 1110” ↔ “11 1111 1111”) from actual conversion characteristics.
- Differential linearity error
  The deviation of input voltage needed to change the output by 1 LSB from the theoretical value.



- 1 LSB = $\frac{V_{FST} - V_{OT}}{1022}$
- Linearity error = $\frac{V_{NT} - (1\ LSB \times N + V_{OT})}{1\ LSB}$ (LSB)
- Differential linearity error = $\frac{V_{(N+1)T} - V_{NT}}{1\ LSB} - 1$ (LSB)

■ INSTRUCTION SET (412 INSTRUCTIONS)

Table 1 Explanation of Items in Table of Instructions

Item	Explanation
Mnemonic	Upper-case letters and symbols: Represented as they appear in assembler Lower-case letters: Replaced when described in assembler. Numbers after lower-case letters: Indicate the bit width within the instruction.
#	Indicates the number of bytes.
~	Indicates the number of cycles. See Table 4 for details about meanings of letters in items.
B	Indicates the correction value for calculating the number of actual cycles during execution of instruction. The number of actual cycles during execution of instruction is summed with the value in the “cycles” column.
Operation	Indicates operation of instruction.
LH	Indicates special operations involving the bits 15 through 08 of the accumulator. Z: Transfers “0”. X: Extends before transferring. —: Transfers nothing.
AH	Indicates special operations involving the high-order 16 bits in the accumulator. *: Transfers from AL to AH. —: No transfer. Z: Transfers 00_H to AH. X: Transfers 00_H or FF_H to AH by extending AL.
I S T N Z V C	Indicates the status of each of the following flags: I (interrupt enable), S (stack), T (sticky bit), N (negative), Z (zero), V (overflow), and C (carry). *: Changes due to execution of instruction. —: No change. S: Set by execution of instruction. R: Reset by execution of instruction.
RMW	Indicates whether the instruction is a read-modify-write instruction (a single instruction that reads data from memory, etc., processes the data, and then writes the result to memory.). *: Instruction is a read-modify-write instruction —: Instruction is not a read-modify-write instruction Note: Cannot be used for addresses that have different meanings depending on whether they are read or written.

Table 2 Explanation of Symbols in Table of Instructions

Symbol	Explanation
A	32-bit accumulator The number of bits used varies according to the instruction. Byte: Low order 8 bits of AL Word: 16 bits of AL Long: 32 bits of AL, AH
AH	High-order 16 bits of A
AL	Low-order 16 bits of A
SP	Stack pointer (USP or SSP)
PC	Program counter
SPCU	Stack pointer upper limit register
SPCL	Stack pointer lower limit register
PCB	Program bank register
DTB	Data bank register
ADB	Additional data bank register
SSB	System stack bank register
USB	User stack bank register
SPB	Current stack bank register (SSB or USB)
DPR	Direct page register
brg1	DTB, ADB, SSB, USB, DPR, PCB, SPB
brg2	DTB, ADB, SSB, USB, DPR, SPB
Ri	R0, R1, R2, R3, R4, R5, R6, R7
RWi	RW0, RW1, RW2, RW3, RW4, RW5, RW6, RW7
RWj	RW0, RW1, RW2, RW3
RLi	RL0, RL1, RL2, RL3
dir addr16 addr24 addr24 0 to 15 addr24 16 to 23	Compact direct addressing Direct addressing Physical direct addressing Bits 0 to 15 of addr24 Bits 16 to 23 of addr24
io	I/O area (000000_H to $0000FF_H$)

(Continued)

(Continued)

Symbol	Explanation
#imm4 #imm8 #imm16 #imm32 ext (imm8)	4-bit immediate data 8-bit immediate data 16-bit immediate data 32-bit immediate data 16-bit data signed and extended from 8-bit immediate data
disp8 disp16	8-bit displacement 16-bit displacement
bp	Bit offset value
vct4 vct8	Vector number (0 to 15) Vector number (0 to 255)
( )b	Bit address
rel ear eam	Branch specification relative to PC Effective addressing (codes 00 to 07) Effective addressing (codes 08 to 1F)
rlst	Register list

Table 3 Effective Address Fields

Code	Notation	Address format	Number of bytes in address extemsion*
00 01 02 03 04 05 06 07	R0 RW0 RL0 R1 RW1 (RL0) R2 RW2 RL1 R3 RW3 (RL1) R4 RW4 RL2 R5 RW5 (RL2) R6 RW6 RL3 R7 RW7 (RL3)	Register direct “ea” corresponds to byte, word, and long-word types, starting from the left	—
08 09 0A 0B	@RW0 @RW1 @RW2 @RW3	Register indirect	0
0C 0D 0E 0F	@RW0 + @RW1 + @RW2 + @RW3 +	Register indirect with post-increment	0
10 11 12 13 14 15 16 17	@RW0 + disp8 @RW1 + disp8 @RW2 + disp8 @RW3 + disp8 @RW4 + disp8 @RW5 + disp8 @RW6 + disp8 @RW7 + disp8	Register indirect with 8-bit displacement	1
18 19 1A 1B	@RW0 + disp16 @RW1 + disp16 @RW2 + disp16 @RW3 + disp16	Register indirect with 16-bit displacemen	2
1C 1D 1E 1F	@RW0 + RW7 @RW1 + RW7 @PC + dip16 addr16	Register indirect with index Register indirect with index PC indirect with 16-bit displacement Direct address	0 0 2 2

* : The number of bytes for address extension is indicated by the “+” symbol in the “#” (number of bytes) column in the Table of Instructions.

Table 4 Number of Execution Cycles for Each Form of Addressing

Code	Operand	(a)*
		Number of execution cycles for each from of addressing
00 to 07	Ri RWi RLi	Listed in Table of Instructions
08 to 0B	@RWj	1
0C to 0F	@RWj +	4
10 to 17	@RWi + disp8	1
18 to 1B	@RWj + disp16	1
1C 1D 1E 1F	@RW0 + RW7 @RW1 + RW7 @PC + dip16 @addr16	2 2 2 1

* : “(a)” is used in the “cycles” (number of cycles) column and column B (correction value) in the Table of Instructions.

Table 5 Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles

Operand	(b)*		(c)*		(d)*	
	byte		word		long	
Internal register	+	0	+	0	+	0
Internal RAM even address	+	0	+	0	+	0
Internal RAM odd address	+	0	+	1	+	2
Even address not in internal RAM	+	1	+	1	+	2
Odd address not in internal RAM	+	1	+	3	+	6
External data bus (8 bits)	+	1	+	3	+	6

* : “(b)”, “(c)”, and “(d)” are used in the “cycles” (number of cycles) column and column B (correction value) in the Table of Instructions.

Table 6 Transfer Instructions (Byte) [50 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOV A, dir	2	2	(b)	byte (A) ← (dir)	Z	*	–	–	–	*	*	–	–	–
MOV A, addr16	3	2	(b)	byte (A) ← (addr16)	Z	*	–	–	–	*	*	–	–	–
MOV A, Ri	1	1	0	byte (A) ← (Ri)	Z	*	–	–	–	*	*	–	–	–
MOV A, ear	2	1	0	byte (A) ← (ear)	Z	*	–	–	–	*	*	–	–	–
MOV A, eam	2+	2+ (a)	(b)	byte (A) ← (eam)	Z	*	–	–	–	*	*	–	–	–
MOV A, io	2	2	(b)	byte (A) ← (io)	Z	*	–	–	–	*	*	–	–	–
MOV A, #imm8	2	2	0	byte (A) ← imm8	Z	*	–	–	–	*	*	–	–	–
MOV A, @A	2	2	(b)	byte (A) ← ((A))	Z	–	–	–	–	*	*	–	–	–
MOV A, @RLi+disp8	3	6	(b)	byte (A) ← ((RLi))+disp8)	Z	*	–	–	–	*	*	–	–	–
MOV A, @SP+disp8	3	3	(b)	byte (A) ← ((SP)+disp8)	Z	*	–	–	–	*	*	–	–	–
MOVP A, addr24	5	3	(b)	byte (A) ←(addr24)	Z	*	–	–	–	*	*	–	–	–
MOVP A, @A	2	2	(b)	byte (A) ← ((A))	Z	–	–	–	–	*	*	–	–	–
MOVN A, #imm4	1	1	0	byte (A) ← imm4	Z	*	–	–	–	R	*	–	–	–
MOVX A, dir	2	2	(b)	byte (A) ← (dir)	X	*	–	–	–	*	*	–	–	–
MOVX A, addr16	3	2	(b)	byte (A) ← (addr16)	X	*	–	–	–	*	*	–	–	–
MOVX A, Ri	2	1	0	byte (A) ← (Ri)	X	*	–	–	–	*	*	–	–	–
MOVX A, ear	2	1	0	byte (A) ← (ear)	X	*	–	–	–	*	*	–	–	–
MOVX A, eam	2+	2+ (a)	(b)	byte (A) ← (eam)	X	*	–	–	–	*	*	–	–	–
MOVX A, io	2	2	(b)	byte (A) ← (io)	X	*	–	–	–	*	*	–	–	–
MOVX A, #imm8	2	2	0	byte (A) ← imm8	X	*	–	–	–	*	*	–	–	–
MOVX A, @A	2	2	(b)	byte (A) ← ((A))	X	–	–	–	–	*	*	–	–	–
MOVX A,@RWi+disp8	2	3	(b)	byte (A) ← ((RWi))+disp8)	X	*	–	–	–	*	*	–	–	–
MOVX A, @RLi+disp8	3	6	(b)	byte (A) ← ((RLi))+disp8)	X	*	–	–	–	*	*	–	–	–
MOVX A, @SP+disp8	3	3	(b)	byte (A) ← ((SP)+disp8)	X	*	–	–	–	*	*	–	–	–
MOVPX A, addr24	5	3	(b)	byte (A) ←(addr24)	X	*	–	–	–	*	*	–	–	–
MOVPX A, @A	2	2	(b)	byte (A) ← ((A))	X	–	–	–	–	*	*	–	–	–
MOV dir, A	2	2	(b)	byte (dir) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV addr16, A	3	2	(b)	byte (addr16) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV Ri, A	1	1	0	byte (Ri) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV ear, A	2	2	0	byte (ear) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV eam, A	2+	2+ (a)	(b)	byte (eam) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV io, A	2	2	(b)	byte (io) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV @RLi+disp8, A	3	6	(b)	byte ((RLi)) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV @SP+disp8, A	3	3	(b)	byte ((SP)+disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVP addr24, A	5	3	(b)	byte (addr24) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV Ri, ear	2	2	0	byte (Ri) ← (ear)	–	–	–	–	–	*	*	–	–	–
MOV Ri, eam	2+	3+ (a)	(b)	byte (Ri) ← (eam)	–	–	–	–	–	*	*	–	–	–
MOVP @A, Ri	2	3	(b)	byte ((A)) ← (Ri)	–	–	–	–	–	*	*	–	–	–
MOV ear, Ri	2	3	0	byte (ear) ← (Ri)	–	–	–	–	–	*	*	–	–	–
MOV eam, Ri	2+	3+ (a)	(b)	byte (eam) ← (Ri)	–	–	–	–	–	*	*	–	–	–
MOV Ri, #imm8	2	2	0	byte (Ri) ← imm8	–	–	–	–	–	*	*	–	–	–
MOV io, #imm8	3	3	(b)	byte (io) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV dir, #imm8	3	3	(b)	byte (dir) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV ear, #imm8	3	2	0	byte (ear) ← imm8	–	–	–	–	–	*	*	–	–	–
MOV eam, #imm8	3+	2+ (a)	(b)	byte (eam) ← imm8	–	–	–	–	–	–	–	–	–	–
MOV @AL, AH	2	2	(b)	byte ((A)) ← (AH)	–	–	–	–	–	*	*	–	–	–

(Continued)

(Continued)

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
XCH A, ear	2	3	0	byte (A) ↔ (ear)	Z	–	–	–	–	–	–	–	–	–
XCH A, eam	2+	3+ (a)	2× (b)	byte (A) ↔ (eam)	Z	–	–	–	–	–	–	–	–	–
XCH Ri, ear	2	4	0	byte (Ri) ↔ (ear)	–	–	–	–	–	–	–	–	–	–
XCH Ri, eam	2+	5+ (a)	2× (b)	byte (Ri) ↔ (eam)	–	–	–	–	–	–	–	–	–	–

For an explanation of “(a)” and “(b)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 7 Transfer Instructions (Word) [40 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVW A, dir	2	2	(c)	word (A) ← (dir)	–	*	–	–	–	*	*	–	–	–
MOVW A, addr16	3	2	(c)	word (A) ← (addr16)	–	*	–	–	–	*	*	–	–	–
MOVW A, SP	1	2	0	word (A) ← (SP)	–	*	–	–	–	*	*	–	–	–
MOVW A, RWi	1	1	0	word (A) ← (RWi)	–	*	–	–	–	*	*	–	–	–
MOVW A, ear	2	1	0	word (A) ← (ear)	–	*	–	–	–	*	*	–	–	–
MOVW A, eam	2+	2+ (a)	(c)	word (A) ← (eam)	–	*	–	–	–	*	*	–	–	–
MOVW A, io	2	2	(c)	word (A) ← (io)	–	*	–	–	–	*	*	–	–	–
MOVW A, @A	2	2	(c)	word (A) ← ((A))	–	–	–	–	–	*	*	–	–	–
MOVW A, #imm16	3	2	0	word (A) ← imm16	–	*	–	–	–	*	*	–	–	–
MOVW A, @RWi+disp8	2	3	(c)	word (A) ← ((RWi) +disp8)	–	*	–	–	–	*	*	–	–	–
MOVW A, @RLi+disp8	3	6	(c)	word (A) ← ((RLi) +disp8)	–	*	–	–	–	*	*	–	–	–
MOVW A, @SP+disp8	3	3	(c)	word (A) ← ((SP) +disp8	–	*	–	–	–	*	*	–	–	–
MOVPW A, addr24	5	3	(c)	word (A) ← (addr24)	–	*	–	–	–	*	*	–	–	–
MOVPW A, @A	2	2	(c)	word (A) ← ((A))	–	–	–	–	–	*	*	–	–	–
MOVW dir, A	2	2	(c)	word (dir) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW addr16, A	3	2	(c)	word (addr16) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW SP, # imm16	4	2	0	word (SP) ← imm16	–	–	–	–	–	*	*	–	–	–
MOVW SP, A	1	2	0	word (SP) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, A	1	1	0	word (RWi) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW ear, A	2	2	0	word (ear) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW eam, A	2+	2+ (a)	(c)	word (eam) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW io, A	2	2	(c)	word (io) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW @RWi+disp8, A	2	3	(c)	word ((RWi) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW @RLi+disp8, A	3	6	(c)	word ((RLi) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVW @SP+disp8, A	3	3	(c)	word ((SP) +disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVPW addr24, A	5	3	(c)	word (addr24) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVPW @A, RWi	2	3	(c)	word ((A)) ← (RWi)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, ear	2	2	0	word (RWi) ← (ear)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, eam	2+	3+ (a)	(c)	word (RWi) ← (eam)	–	–	–	–	–	*	*	–	–	–
MOVW ear, RWi	2	3	0	word (ear) ← (RWi)	–	–	–	–	–	*	*	–	–	–
MOVW eam, RWi	2+	3+ (a)	(c)	word (eam) ← (RWi)	–	–	–	–	–	*	*	–	–	–
MOVW RWi, #imm16	3	2	0	word (RWi) ← imm16	–	–	–	–	–	*	*	–	–	–
MOVW io, #imm16	4	3	(c)	word (io) ← imm16	–	–	–	–	–	–	–	–	–	–
MOVW ear, #imm16	4	2	0	word (ear) ← imm16	–	–	–	–	–	*	*	–	–	–
MOVW eam, #imm16	4+	2+ (a)	(c)	word (eam) ← imm16	–	–	–	–	–	–	–	–	–	–
MOVW @AL, AH	2	2	(c)	word ((A)) ← (AH)	–	–	–	–	–	*	*	–	–	–
XCHW A, ear	2	3	0	word (A) ↔ (ear)	–	–	–	–	–	–	–	–	–	–
XCHW A, eam	2+	3+ (a)	2× (c)	word (A) ↔ (eam)	–	–	–	–	–	–	–	–	–	–
XCHW RWi, ear	2	4	0	word (RWi) ↔ (ear)	–	–	–	–	–	–	–	–	–	–
XCHW RWi, eam	2+	5+ (a)	2× (c)	word (RWi) ↔ (eam)	–	–	–	–	–	–	–	–	–	–

Note: For an explanation of “(a)” and “(c)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 8 Transfer Instructions (Long Word) [11 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVL A, ear	2	1	0	long (A) ← (ear)	–	–	–	–	–	*	*	–	–	–
MOVL A, eam	2+	3+ (a)	(d)	long (A) ← (eam)	–	–	–	–	–	*	*	–	–	–
MOVL A, # imm32	5	3	0	long (A) ← imm32	–	–	–	–	–	*	*	–	–	–
MOVL A, @SP + disp8	3	4	(d)	long (A) ← ((SP) +disp8)	–	–	–	–	–	*	*	–	–	–
MOVPL A, addr24	5	4	(d)	long (A) ← (addr24)	–	–	–	–	–	*	*	–	–	–
MOVPL A, @A	2	3	(d)	long (A) ← ((A))	–	–	–	–	–	*	*	–	–	–
MOVPL @A, RLi	2	5	(d)	long ((A)) ← (RLi)	–	–	–	–	–	*	*	–	–	–
MOVL @SP + disp8, A	3	4	(d)	long ((SP) + disp8) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVPL addr24, A	5	4	(d)	long (addr24) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVL ear, A	2	2	0	long (ear) ← (A)	–	–	–	–	–	*	*	–	–	–
MOVL eam, A	2+	3+ (a)	(d)	long (eam) ← (A)	–	–	–	–	–	*	*	–	–	–

For an explanation of “(a)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 9 Addition and Subtraction Instructions (Byte/Word/Long Word) [42 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ADD A, #imm8	2	2	0	byte (A) ← (A) +imm8	Z	–	–	–	–	*	*	*	*	–
ADD A, dir	2	3	(b)	byte (A) ← (A) +(dir)	Z	–	–	–	–	*	*	*	*	–
ADD A, ear	2	2	0	byte (A) ← (A) +(ear)	Z	–	–	–	–	*	*	*	*	–
ADD A, eam	2+	3+ (a)	(b)	byte (A) ← (A) +(eam)	Z	–	–	–	–	*	*	*	*	–
ADD ear, A	2	2	0	byte (ear) ← (ear) + (A)	–	–	–	–	–	*	*	*	*	*
ADD eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) + (A)	Z	–	–	–	–	*	*	*	*	*
ADDC A	1	2	0	byte (A) ← (AH) + (AL) + (C)	Z	–	–	–	–	*	*	*	*	–
ADDC A, ear	2	2	0	byte (A) ← (A) + (ear) + (C)	Z	–	–	–	–	*	*	*	*	–
ADDC A, eam	2+	3+ (a)	(b)	byte (A) ← (A) + (eam) + (C)	Z	–	–	–	–	*	*	*	*	–
ADDDC A	1	3	0	byte (A) ← (AH) + (AL) + (C) (Decimal)	Z	–	–	–	–	*	*	*	*	–
SUB A, #imm8	2	2	0	byte (A) ← (A) –imm8	Z	–	–	–	–	*	*	*	*	–
SUB A, dir	2	3	(b)	byte (A) ← (A) – (dir)	Z	–	–	–	–	*	*	*	*	–
SUB A, ear	2	2	0	byte (A) ← (A) – (ear)	Z	–	–	–	–	*	*	*	*	–
SUB A, eam	2+	3+ (a)	(b)	byte (A) ← (A) – (eam)	Z	–	–	–	–	*	*	*	*	–
SUB ear, A	2	2	0	byte (ear) ← (ear) – (A)	–	–	–	–	–	*	*	*	*	*
SUB eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) – (A)	–	–	–	–	–	*	*	*	*	*
SUBC A	1	2	0	byte (A) ← (AH) – (AL) – (C)	Z	–	–	–	–	*	*	*	*	–
SUBC A, ear	2	2	0	byte (A) ← (A) – (ear) – (C)	Z	–	–	–	–	*	*	*	*	–
SUBC A, eam	2+	3+ (a)	(b)	byte (A) ← (A) – (eam) – (C)	Z	–	–	–	–	*	*	*	*	–
SUBDC A	1	3	0	byte (A) ← (AH) – (AL) – (C) (Decimal)	Z	–	–	–	–	*	*	*	*	–
ADDW A	1	2	0	word (A) ← (AH) + (AL)	–	–	–	–	–	*	*	*	*	–
ADDW A, ear	2	2	0	word (A) ← (A) +(ear)	–	–	–	–	–	*	*	*	*	–
ADDW A, eam	2+	3+ (a)	(c)	word (A) ← (A) +(eam)	–	–	–	–	–	*	*	*	*	–
ADDW A, #imm16	3	2	0	word (A) ← (A) +imm16	–	–	–	–	–	*	*	*	*	–
ADDW ear, A	2	2	0	word (ear) ← (ear) + (A)	–	–	–	–	–	*	*	*	*	*
ADDW eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) + (A)	–	–	–	–	–	*	*	*	*	*
ADDCW A, ear	2	2	0	word (A) ← (A) + (ear) + (C)	–	–	–	–	–	*	*	*	*	–
ADDCW A, eam	2+	3+ (a)	(c)	word (A) ← (A) + (eam) + (C)	–	–	–	–	–	*	*	*	*	–
SUBW A	1	2	0	word (A) ← (AH) – (AL)	–	–	–	–	–	*	*	*	*	–
SUBW A, ear	2	2	0	word (A) ← (A) – (ear)	–	–	–	–	–	*	*	*	*	–
SUBW A, eam	2+	3+ (a)	(c)	word (A) ← (A) – (eam)	–	–	–	–	–	*	*	*	*	–
SUBW A, #imm16	3	2	0	word (A) ← (A) –imm16	–	–	–	–	–	*	*	*	*	–
SUBW ear, A	2	2	0	word (ear) ← (ear) – (A)	–	–	–	–	–	*	*	*	*	*
SUBW eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) – (A)	–	–	–	–	–	*	*	*	*	*
SUBCW A, ear	2	2	0	word (A) ← (A) – (ear) – (C)	–	–	–	–	–	*	*	*	*	–
SUBCW A, eam	2+	3+ (a)	(c)	word (A) ← (A) – (eam) – (C)	–	–	–	–	–	*	*	*	*	–
ADDL A, ear	2	5	0	long (A) ← (A) + (ear)	–	–	–	–	–	*	*	*	*	–
ADDL A, eam	2+	6+ (a)	(d)	long (A) ← (A) + (eam)	–	–	–	–	–	*	*	*	*	–
ADDL A, #imm32	5	4	0	long (A) ← (A) +imm32	–	–	–	–	–	*	*	*	*	–
SUBL A, ear	2	5	0	long (A) ← (A) – (ear)	–	–	–	–	–	*	*	*	*	–
SUBL A, eam	2+	6+ (a)	(d)	long (A) ← (A) – (eam)	–	–	–	–	–	*	*	*	*	–
SUBL A, #imm32	5	4	0	long (A) ← (A) –imm32	–	–	–	–	–	*	*	*	*	–

For an explanation of “(a)”, “(b)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 10 Increment and Decrement Instructions (Byte/Word/Long Word) [12 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
INC ear	2	2	0	byte (ear) ← (ear) +1	–	–	–	–	–	*	*	*	–	*
INC eam	2+	3+ (a)	2× (b)	byte (eam) ← (eam) +1	–	–	–	–	–	*	*	*	–	*
DEC ear	2	2	0	byte (ear) ← (ear) –1	–	–	–	–	–	*	*	*	–	*
DEC eam	2+	3+ (a)	2× (b)	byte (eam) ← (eam) –1	–	–	–	–	–	*	*	*	–	*
INCW ear	2	2	0	word (ear) ← (ear) +1	–	–	–	–	–	*	*	*	–	*
INCW eam	2+	3+ (a)	2× (c)	word (eam) ← (eam) +1	–	–	–	–	–	*	*	*	–	*
DECW ear	2	2	0	word (ear) ← (ear) –1	–	–	–	–	–	*	*	*	–	*
DECW eam	2+	3+ (a)	2× (c)	word (eam) ← (eam) –1	–	–	–	–	–	*	*	*	–	*
INCL ear	2	4	0	long (ear) ← (ear) +1	–	–	–	–	–	*	*	*	–	*
INCL eam	2+	5+ (a)	2× (d)	long (eam) ← (eam) +1	–	–	–	–	–	*	*	*	–	*
DECL ear	2	4	0	long (ear) ← (ear) –1	–	–	–	–	–	*	*	*	–	*
DECL eam	2+	5+ (a)	2× (d)	long (eam) ← (eam) –1	–	–	–	–	–	*	*	*	–	*

For an explanation of “(a)”, “(b)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 11 Compare Instructions (Byte/Word/Long Word) [11 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
CMP A	1	2	0	byte (AH) – (AL)	–	–	–	–	–	*	*	*	*	–
CMP A, ear	2	2	0	byte (A) – (ear)	–	–	–	–	–	*	*	*	*	–
CMP A, eam	2+	2+ (a)	(b)	byte (A) – (eam)	–	–	–	–	–	*	*	*	*	–
CMP A, #imm8	2	2	0	byte (A) – imm8	–	–	–	–	–	*	*	*	*	–
CMPW A	1	2	0	word (AH) – (AL)	–	–	–	–	–	*	*	*	*	–
CMPW A, ear	2	2	0	word (A) – (ear)	–	–	–	–	–	*	*	*	*	–
CMPW A, eam	2+	2+ (a)	(c)	word (A) – (eam)	–	–	–	–	–	*	*	*	*	–
CMPW A, #imm16	3	2	0	word (A) – imm16	–	–	–	–	–	*	*	*	*	–
CMPL A, ear	2	3	0	long (A) – (ear)	–	–	–	–	–	*	*	*	*	–
CMPL A, eam	2+	4+ (a)	(d)	long (A) – (eam)	–	–	–	–	–	*	*	*	*	–
CMPL A, #imm32	5	3	0	long (A) – imm32	–	–	–	–	–	*	*	*	*	–

For an explanation of “(a)”, “(b)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 12 Unsigned Multiplication and Division Instructions (Word/Long Word) [11 Instructions]

Mnemonic		#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
DIVU	A	1	*1	0	word (AH) /byte (AL) Quotient → byte (AL) Remainder → byte (AH)	–	–	–	–	–	–	–	*	*	–
DIVU	A, ear	2	*2	0	word (A)/byte (ear) Quotient → byte (A) Remainder → byte (ear)	–	–	–	–	–	–	–	*	*	–
DIVU	A, eam	2+	*3	*6	word (A)/byte (eam) Quotient → byte (A) Remainder → byte (eam)	–	–	–	–	–	–	–	*	*	–
DIVUW	A, ear	2	*4	0	long (A)/word (ear) Quotient → word (A) Remainder → word (ear)	–	–	–	–	–	–	–	*	*	–
DIVUW	A, eam	2+	*5	*7	long (A)/word (eam) Quotient → word (A) Remainder → word (eam)	–	–	–	–	–	–	–	*	*	–
MULU	A	1	*8	0	byte (AH) × byte (AL) → word (A)	–	–	–	–	–	–	–	–	–	–
MULU	A, ear	2	*9	0	byte (A) × byte (ear) → word (A)	–	–	–	–	–	–	–	–	–	–
MULU	A, eam	2+	*10	(b)	byte (A) × byte (eam) → word (A)	–	–	–	–	–	–	–	–	–	–
MULUW	A	1	*11	0	word (AH) × word (AL) → long (A)	–	–	–	–	–	–	–	–	–	–
MULUW	A, ear	2	*12	0	word (A) × word (ear) → long (A)	–	–	–	–	–	–	–	–	–	–
MULUW	A, eam	2+	*13	(c)	word (A) × word (eam) → long (A)	–	–	–	–	–	–	–	–	–	–

For an explanation of “(b)” and “(c), refer to Table 5, “Correction Values for Number of Cycle Used to Calculate Number of Actual Cycles.”

*1: 3 when dividing into zero, 6 when an overflow occurs, and 14 normally.
*2: 3 when dividing into zero, 5 when an overflow occurs, and 13 normally.
*3: 5 + (a) when dividing into zero, 7 + (a) when an overflow occurs, and 17 + (a) normally.
*4: 3 when dividing into zero, 5 when an overflow occurs, and 21 normally.
*5: 4 + (a) when dividing into zero, 7 + (a) when an overflow occurs, and 25 + (a) normally.
*6: (b) when dividing into zero or when an overflow occurs, and 2 × (b) normally.
*7: (c) when dividing into zero or when an overflow occurs, and 2 × (c) normally.
*8: 3 when byte (AH) is zero, and 7 when byte (AH) is not 0.
*9: 3 when byte (ear) is zero, and 7 when byte (ear) is not 0.
*10: 4 + (a) when byte (eam) is zero, and 8 + (a) when byte (eam) is not 0.
*11: 3 when word (AH) is zero, and 11 when word (AH) is not 0.
*12: 3 when word (ear) is zero, and 11 when word (ear) is not 0.
*13: 4 + (a) when word (eam) is zero, and 12 + (a) when word (eam) is not 0.

Table 13 Signed Multiplication and Division Instructions (Word/Long Word) [11 Insturctions]

Mnemonic		#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
DIV	A	2	*1	0	word (AH) /byte (AL) Quotient → byte (AL) Remainder → byte (AH)	Z	–	–	–	–	–	–	*	*	–
DIV	A, ear	2	*2	0	word (A)/byte (ear) Quotient → byte (A) Remainder → byte (ear)	Z	–	–	–	–	–	–	*	*	–
DIV	A, eam	2+	*3	*6	word (A)/byte (eam) Quotient → byte (A) Remainder → byte (eam)	Z	–	–	–	–	–	–	*	*	–
DIVW	A, ear	2	*4	0	long (A)/word (ear) Quotient → word (A) Remainder → word (ear)	–	–	–	–	–	–	–	*	*	–
DIVW	A, eam	2+	*5	*7	long (A)/word (eam) Quotient → word (A) Remainder → word (eam)	–	–	–	–	–	–	–	*	*	–
MUL	A	2	*8	0	byte (AH) × byte (AL) → word (A)	–	–	–	–	–	–	–	–	–	–
MUL	A, ear	2	*9	0	byte (A) × byte (ear) → word (A)	–	–	–	–	–	–	–	–	–	–
MUL	A, eam	2+	*10	(b)	byte (A) × byte (eam) → word (A)	–	–	–	–	–	–	–	–	–	–
MULW	A	2	*11	0	word (AH) × word (AL) → long (A)	–	–	–	–	–	–	–	–	–	–
MULW	A, ear	2	*12	0	word (A) × word (ear) → long (A)	–	–	–	–	–	–	–	–	–	–
MULW	A, eam	2+	*13	(b)	word (A) × word (eam) → long (A)	–	–	–	–	–	–	–	–	–	–

For an explanation of “(b)” and “(c)”, refer to Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

*1: 3 when dividing into zero, 8 or 18 when an overflow occurs, and 18 normally.

*2: 3 when dividing into zero, 10 or 21 when an overflow occurs, and 22 normally.

*3: 4 + (a) when dividing into zero, 11 + (a) or 22 + (a) when an overflow occurs, and 23 + (a) normally.

*4: When the dividend is positive: 4 when dividing into zero, 10 or 29 when an overflow occurs, and 30 normally.
When the dividend is negative: 4 when dividing into zero, 11 or 30 when an overflow occurs, and 31 normally.

*5: When the dividend is positive: 4 + (a) when dividing into zero, 11 + (a) or 30 + (a) when an overflow occurs, and 31 + (a) normally.
When the dividend is negative: 4 + (a) when dividing into zero, 12 + (a) or 31 + (a) when an overflow occurs, and 32 + (a) normally.

*6: (b) when dividing into zero or when an overflow occurs, and 2 × (b) normally.

*7: (c) when dividing into zero or when an overflow occurs, and 2 × (c) normally.

*8: 3 when byte (AH) is zero, 12 when the result is positive, and 13 when the result is negative.

*9: 3 when byte (ear) is zero, 12 when the result is positive, and 13 when the result is negative.

*10: 4 + (a) when byte (eam) is zero, 13 + (a) when the result is positive, and 14 + (a) when the result is negative.

*11: 3 when word (AH) is zero, 12 when the result is positive, and 13 when the result is negative.

*12: 3 when word (ear) is zero, 16 when the result is positive, and 19 when the result is negative.

*13: 4 + (a) when word (eam) is zero, 17 + (a) when the result is positive, and 20 + (a) when the result is negative.

Note: Which of the two values given for the number of execution cycles applies when an overflow error occurs in a DIV or DIVW instruction depends on whether the overflow was detected before or after the operation.

Table 14 Logical 1 Instructions (Byte, Word) [39 Instructions]

Mnemonic		#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
AND	A, #imm8	2	2	0	byte (A) ← (A) and imm8	–	–	–	–	–	*	*	R	–	–
AND	A, ear	2	2	0	byte (A) ← (A) and (ear)	–	–	–	–	–	*	*	R	–	–
AND	A, eam	2+	3+ (a)	(b)	byte (A) ← (A) and (eam)	–	–	–	–	–	*	*	R	–	–
AND	ear, A	2	3	0	byte (ear) ← (ear) and (A)	–	–	–	–	–	*	*	R	–	*
AND	eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) and (A)	–	–	–	–	–	*	*	R	–	*
OR	A, #imm8	2	2	0	byte (A) ← (A) or imm8	–	–	–	–	–	*	*	R	–	–
OR	A, ear	2	2	0	byte (A) ← (A) or (ear)	–	–	–	–	–	*	*	R	–	–
OR	A, eam	2+	3+ (a)	(b)	byte (A) ← (A) or (eam)	–	–	–	–	–	*	*	R	–	–
OR	ear, A	2	3	0	byte (ear) ← (ear) or (A)	–	–	–	–	–	*	*	R	–	*
OR	eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) or (A)	–	–	–	–	–	*	*	R	–	*
XOR	A, #imm8	2	2	0	byte (A) ← (A) xor imm8	–	–	–	–	–	*	*	R	–	–
XOR	A, ear	2	2	0	byte (A) ← (A) xor (ear)	–	–	–	–	–	*	*	R	–	–
XOR	A, eam	2+	3+ (a)	(b)	byte (A) ← (A) xor (eam)	–	–	–	–	–	*	*	R	–	–
XOR	ear, A	2	3	0	byte (ear) ← (ear) xor (A)	–	–	–	–	–	*	*	R	–	*
XOR	eam, A	2+	3+ (a)	2× (b)	byte (eam) ← (eam) xor (A)	–	–	–	–	–	*	*	R	–	*
NOT	A	1	2	0	byte (A) ← not (A)	–	–	–	–	–	*	*	R	–	–
NOT	ear	2	2	0	byte (ear) ← not (ear)	–	–	–	–	–	*	*	R	–	*
NOT	eam	2+	3+ (a)	2× (b)	byte (eam) ← not (eam)	–	–	–	–	–	*	*	R	–	*
ANDW	A	1	2	0	word (A) ← (AH) and (A)	–	–	–	–	–	*	*	R	–	–
ANDW	A, #imm16	3	2	0	word (A) ← (A) and imm16	–	–	–	–	–	*	*	R	–	–
ANDW	A, ear	2	2	0	word (A) ← (A) and (ear)	–	–	–	–	–	*	*	R	–	–
ANDW	A, eam	2+	3+ (a)	(c)	word (A) ← (A) and (eam)	–	–	–	–	–	*	*	R	–	–
ANDW	ear, A	2	3	0	word (ear) ← (ear) and (A)	–	–	–	–	–	*	*	R	–	*
ANDW	eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) and (A)	–	–	–	–	–	*	*	R	–	*
ORW	A	1	2	0	word (A) ← (AH) or (A)	–	–	–	–	–	*	*	R	–	–
ORW	A, #imm16	3	2	0	word (A) ← (A) or imm16	–	–	–	–	–	*	*	R	–	–
ORW	A, ear	2	2	0	word (A) ← (A) or (ear)	–	–	–	–	–	*	*	R	–	–
ORW	A, eam	2+	3+ (a)	(c)	word (A) ← (A) or (eam)	–	–	–	–	–	*	*	R	–	–
ORW	ear, A	2	3	0	word (ear) ← (ear) or (A)	–	–	–	–	–	*	*	R	–	*
ORW	eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) or (A)	–	–	–	–	–	*	*	R	–	*
XORW	A	1	2	0	word (A) ← (AH) xor (A)	–	–	–	–	–	*	*	R	–	–
XORW	A, #imm16	3	2	0	word (A) ← (A) xor imm16	–	–	–	–	–	*	*	R	–	–
XORW	A, ear	2	2	0	word (A) ← (A) xor (ear)	–	–	–	–	–	*	*	R	–	–
XORW	A, eam	2+	3+ (a)	(c)	word (A) ← (A) xor (eam)	–	–	–	–	–	*	*	R	–	–
XORW	ear, A	2	3	0	word (ear) ← (ear) xor (A)	–	–	–	–	–	*	*	R	–	*
XORW	eam, A	2+	3+ (a)	2× (c)	word (eam) ← (eam) xor (A)	–	–	–	–	–	*	*	R	–	*
NOTW	A	1	2	0	word (A) ← not (A)	–	–	–	–	–	*	*	R	–	–
NOTW	ear	2	2	0	word (ear) ← not (ear)	–	–	–	–	–	*	*	R	–	*
NOTW	eam	2+	3+ (a)	2× (c)	word (eam) ← not (eam)	–	–	–	–	–	*	*	R	–	*

For an explanation of “(a)”, “(b)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 15 Logical 2 Instructions (Long Word) [6 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ANDL A, ear	2	5	0	long (A) ← (A) and (ear)	–	–	–	–	–	*	*	R	–	–
ANDL A, eam	2+	6+ (a)	(d)	long (A) ← (A) and (eam)	–	–	–	–	–	*	*	R	–	–
ORL A, ear	2	5	0	long (A) ← (A) or (ear)	–	–	–	–	–	*	*	R	–	–
ORL A, eam	2+	6+ (a)	(d)	long (A) ← (A) or (eam)	–	–	–	–	–	*	*	R	–	–
XORL A, ear	2	5	0	long (A) ← (A) xor (ear)	–	–	–	–	–	*	*	R	–	–
XORL A, eam	2+	6+ (a)	(d)	long (A) ← (A) xor (eam)	–	–	–	–	–	*	*	R	–	–

For an explanation of “(a)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 16 Sign Inversion Instructions (Byte/Word) [6 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
NEG A	1	2	0	byte (A) ← 0 – (A)	X	–	–	–	–	*	*	*	*	–
NEG ear	2	2	0	byte (ear) ← 0 – (ear)	–	–	–	–	–	*	*	*	*	*
NEG eam	2+	3+ (a)	2× (b)	byte (eam) ← 0 – (eam)	–	–	–	–	–	*	*	*	*	*
NEGW A	1	2	0	word (A) ← 0 – (A)	–	–	–	–	–	*	*	*	*	–
NEGW ear	2	2	0	word (ear) ← 0 – (ear)	–	–	–	–	–	*	*	*	*	*
NEGW eam	2+	3+ (a)	2× (c)	word (eam) ← 0 – (eam)	–	–	–	–	–	*	*	*	*	*

For an explanation of “(a)”, “(b)” and “(c)” and refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

Table 17 Absolute Value Instructions (Byte/Word/Long Word) [3 Insturctions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ABS A	2	2	0	byte (A) ← absolute value (A)	Z	–	–	–	–	*	*	*	–	–
ABSW A	2	2	0	word (A) ← absolute value (A)	–	–	–	–	–	*	*	*	–	–
ABSL A	2	4	0	long (A) ← absolute value (A)	–	–	–	–	–	*	*	*	–	–

Table 18 Normalize Instructions (Long Word) [1 Instruction]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
NRML A, R0	2	*	0	long (A) ← Shifts to the position at which “1” was set first byte (R0) ← current shift count	–	–	–	–	*	–	–	–	–	–

* : 5 when the contents of the accumulator are all zeroes, 5 + (R0) in all other cases.

Table 19 Shift Instructions (Byte/Word/Long Word) [27 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
RORC A	2	2	0	byte (A) ← Right rotation with carry	–	–	–	–	–	*	*	–	*	–
ROLC A	2	2	0	byte (A) ← Left rotation with carry	–	–	–	–	–	*	*	–	*	–
RORC ear	2	2	0	byte (ear) ← Right rotation with carry	–	–	–	–	–	*	*	–	*	*
RORC eam	2+	3+ (a)	2× (b)	byte (eam) ← Right rotation with carry	–	–	–	–	–	*	*	–	*	*
ROLC ear	2	2	0	byte (ear) ← Left rotation with carry	–	–	–	–	–	*	*	–	*	*
ROLC eam	2+	3+ (a)	2× (b)	byte (eam) ← Left rotation with carry	–	–	–	–	–	*	*	–	*	*
ASR A, R0	2	*1	0	byte (A) ← Arithmetic right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSR A, R0	2	*1	0	byte (A) ← Logical right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSL A, R0	2	*1	0	byte (A) ← Logical left barrel shift (A, R0)	–	–	–	–	–	*	*	–	*	–
ASR A, #imm8	3	*3	0	byte (A) ← Arithmetic right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSR A, #imm8	3	*3	0	byte (A) ← Logical right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSL A, #imm8	3	*3	0	byte (A) ← Logical left barrel shift (A, imm8)	–	–	–	–	–	*	*	–	*	–
ASRW A	1	2	0	word (A) ← Arithmetic right shift (A, 1 bit)	–	–	–	–	*	*	*	–	*	–
LSRW A/SHRW A	1	2	0	word (A) ← Logical right shift (A, 1 bit)	–	–	–	–	*	R	*	–	*	–
LSLW A/SHLW A	1	2	0	word (A) ← Logical left shift (A, 1 bit)	–	–	–	–	–	*	*	–	*	–
ASRW A, R0	2	*1	0	word (A) ← Arithmetic right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSRW A, R0	2	*1	0	word (A) ← Logical right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSLW A, R0	2	*1	0	word (A) ← Logical left barrel shift (A, R0)	–	–	–	–	–	*	*	–	*	–
ASRW A, #imm8	3	*3	0	word (A) ← Arithmetic right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSRW A, #imm8	3	*3	0	word (A) ← Logical right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSLW A, #imm8	3	*3	0	word (A) ← Logical left barrel shift (A, imm8)	–	–	–	–	–	*	*	–	*	–
ASRL A, R0	2	*2	0	long (A) ← Arithmetic right shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSRL A, R0	2	*2	0	long (A) ← Logical right barrel shift (A, R0)	–	–	–	–	*	*	*	–	*	–
LSLL A, R0	2	*2	0	long (A) ← Logical left barrel shift (A, R0)	–	–	–	–	–	*	*	–	*	–
ASRL A, #imm8	3	*4	0	long (A) ← Arithmetic right shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSRL A, #imm8	3	*4	0	long (A) ← Logical right barrel shift (A, imm8)	–	–	–	–	*	*	*	–	*	–
LSLL A, #imm8	3	*4	0	long (A) ← Logical left barrel shift (A, imm8)	–	–	–	–	–	*	*	–	*	–

For an explanation of “(a)” and “(b)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”
*1: 3 when R0 is 0, 3 + (R0) in all other cases.
*2: 3 when R0 is 0, 4 + (R0) in all other cases.
*3: 3 when imm8 is 0, 3 + (imm8) in all other cases.
*4: 3 when imm8 is 0, 4 + (imm8) in all other cases.

Table 20 Branch 1 Instructions [31 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
BZ/BEQ rel	2	*1	0	Branch when (Z) = 1	–	–	–	–	–	–	–	–	–	–
BNZ/BNE rel	2	*1	0	Branch when (Z) = 0	–	–	–	–	–	–	–	–	–	–
BC/BLO rel	2	*1	0	Branch when (C) = 1	–	–	–	–	–	–	–	–	–	–
BNC/BHS rel	2	*1	0	Branch when (C) = 0	–	–	–	–	–	–	–	–	–	–
BN rel	2	*1	0	Branch when (N) = 1	–	–	–	–	–	–	–	–	–	–
BP rel	2	*1	0	Branch when (N) = 0	–	–	–	–	–	–	–	–	–	–
BV rel	2	*1	0	Branch when (V) = 1	–	–	–	–	–	–	–	–	–	–
BNV rel	2	*1	0	Branch when (V) = 0	–	–	–	–	–	–	–	–	–	–
BT rel	2	*1	0	Branch when (T) = 1	–	–	–	–	–	–	–	–	–	–
BNT rel	2	*1	0	Branch when (T) = 0	–	–	–	–	–	–	–	–	–	–
BLT rel	2	*1	0	Branch when (V) xor (N) = 1	–	–	–	–	–	–	–	–	–	–
BGE rel	2	*1	0	Branch when (V) xor (N) = 0	–	–	–	–	–	–	–	–	–	–
BLE rel	2	*1	0	((V) xor (N)) or (Z) = 1	–	–	–	–	–	–	–	–	–	–
BGT rel	2	*1	0	((V) xor (N)) or (Z) = 0	–	–	–	–	–	–	–	–	–	–
BLS rel	2	*1	0	Branch when (C) or (Z) = 1	–	–	–	–	–	–	–	–	–	–
BHI rel	2	*1	0	Branch when (C) or (Z) = 0	–	–	–	–	–	–	–	–	–	–
BRA rel	2	*1	0	Branch unconditionally	–	–	–	–	–	–	–	–	–	–
JMP @A	1	2	0	word (PC) ← (A)	–	–	–	–	–	–	–	–	–	–
JMP addr16	3	2	0	word (PC) ← addr16	–	–	–	–	–	–	–	–	–	–
JMP @ear	2	3	0	word (PC) ← (ear)	–	–	–	–	–	–	–	–	–	–
JMP @eam	2+	4+ (a)	(c)	word (PC) ← (eam)	–	–	–	–	–	–	–	–	–	–
JMPP @ear *3	2	3	0	word (PC) ← (ear), (PCB) ← (ear +2)	–	–	–	–	–	–	–	–	–	–
JMPP @eam *3	2+	4+ (a)	(d)	word (PC) ← (eam), (PCB) ← (eam +2)	–	–	–	–	–	–	–	–	–	–
JMPP addr24	4	3	0	word (PC) ← ad24 0 to 15 (PCB) ← ad24 16 to 23	–	–	–	–	–	–	–	–	–	–
CALL @ear *4	2	4	(c)	word (PC) ← (ear)	–	–	–	–	–	–	–	–	–	–
CALL @eam *4	2+	5+ (a)	2× (c)	word (PC) ← (eam)	–	–	–	–	–	–	–	–	–	–
CALL addr16 *5	3	5	(c)	word (PC) ← addr16	–	–	–	–	–	–	–	–	–	–
CALLV #vct4 *5	1	5	2× (c)	Vector call linstruction	–	–	–	–	–	–	–	–	–	–
CALLP @ear *6	2	7	2× (c)	word (PC) ← (ear) 0 to 15, (PCB) ← (ear) 16 to 23	–	–	–	–	–	–	–	–	–	–
CALLP @eam *6	2+	8+ (a)	*2	word (PC) ← (eam) 0 to 15, (PCB) ← (eam) 16 to 23	–	–	–	–	–	–	–	–	–	–
CALLP addr24 *7	4	7	2× (c)	word (PC) ← addr 0 to 15, (PCB) ← addr 16 to 23	–	–	–	–	–	–	–	–	–	–

For an explanation of “(a)”, “(c)” and “(d)”, refer to Table 4, “Number of Execution Cycles for Each Form of Addressing,” and Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

*1: 3 when branching, 2 when not branching.
*2: 3 × (c) + (b)
*3: Read (word) branch address.
*4: W: Save (word) to stack; R: Read (word) branch address.
*5: Save (word) to stack.
*6: W: Save (long word) to W stack; R: Read (long word) branch address.
*7: Save (long word) to stack.

Table 21 Branch 2 Instructions [20 Instructions]

Mnemonic	#	cycle	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
CBNE A, #imm8, rel	3	*1	0	Branch when byte (A) ≠ imm8	–	–	–	–	–	*	*	*	*	–
CWBNE A, #imm16, rel	4	*1	0	Branch when byte (A) ≠ imm16	–	–	–	–	–	*	*	*	*	–
CBNE ear, #imm8, rel	4	*1	0	Branch when byte (ear) ≠ imm8	–	–	–	–	–	*	*	*	*	–
CBNE eam, #imm8, rel	4+	*3	(b)	Branch when byte (eam) ≠ imm8	–	–	–	–	–	*	*	*	*	–
CWBNE ear, #imm16, rel	5	*1	0	Branch when word (ear) ≠ imm16	–	–	–	–	–	*	*	*	*	–
CWBNE eam, #imm16, rel	5+	*3	(c)	Branch when word (eam) ≠ imm16	–	–	–	–	–	*	*	*	*	–
DBNZ ear, rel	3	*2	0	Branch when byte (ear) = (ear) – 1, and (ear) ≠ 0	–	–	–	–	–	*	*	*	–	–
DBNZ eam, rel	3+	*4	2× (b)	Branch when byte (ear) = (eam) – 1, and (eam) ≠ 0	–	–	–	–	–	*	*	*	–	*
DWBNZ ear, rel	3	*2	0	Branch when word (ear) = (ear) – 1, and (ear) ≠ 0	–	–	–	–	–	*	*	*	–	–
DWBNZ eam, rel	3+	*4	2× (c)	Branch when word (eam) = (eam) – 1, and (eam) ≠ 0	–	–	–	–	–	*	*	*	–	*
INT #vct8	2	14	8× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
INT addr16	3	12	6× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
INTP addr24	4	13	6× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
INT9	1	14	8× (c)	Software interrupt	–	–	R	S	–	–	–	–	–	–
RETI	1	9	6× (c)	Return from interrupt	–	–	*	*	*	*	*	*	*	–
RETIQ *6	2	11	*5	Return from interrupt	–	–	*	*	*	*	*	*	*	–
LINK #imm8	2	6	(c)	At constant entry, save old frame pointer to stack, set new frame pointer, and allocate local pointer area	–	–	–	–	–	–	–	–	–	–
UNLINK	1	5	(c)	At constant entry, retrieve old frame pointer from stack.	–	–	–	–	–	–	–	–	–	–
RET *7	1	4	(c)	Return from subroutine	–	–	–	–	–	–	–	–	–	–
RETP *8	1	5	(d)	Return from subroutine	–	–	–	–	–	–	–	–	–	–

For an explanation of “(b)”, “(c)” and “(d)”, refer to Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”

*1: 4 when branching, 3 when not branching
*2: 5 when branching, 4 when not branching
*3: 5 + (a) when branching, 4 + (a) when not branching
*4: 6 + (a) when branching, 5 + (a) when not branching
*5: 3 × (b) + 2 × (c) when an interrupt request is generated, 6 × (c) when returning from the interrupt.
*6: High-speed interrupt return instruction. When an interrupt request is detected during this instruction, the instruction branches to the interrupt vector without performing stack operations when the interrupt is generated.
*7: Return from stack (word)
*8: Return from stack (long word)

Table 22 Other Control Instructions (Byte/Word/Long Word) [36 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
PUSHW A	1	3	(c)	word (SP) ← (SP) –2, ((SP)) ← (A)	–	–	–	–	–	–	–	–	–	–
PUSHW AH	1	3	(c)	word (SP) ← (SP) –2, ((SP)) ← (AH)	–	–	–	–	–	–	–	–	–	–
PUSHW PS	1	3	(c)	word (SP) ← (SP) –2, ((SP)) ← (PS)	–	–	–	–	–	–	–	–	–	–
PUSHW rlst	2	*3	*4	(SP) ← (SP) –2n, ((SP)) ← (rlst)	–	–	–	–	–	–	–	–	–	–
POPW A	1	3	(c)	word (A) ← ((SP)), (SP) ← (SP) +2	–	*	–	–	–	–	–	–	–	–
POPW AH	1	3	(c)	word (AH) ← ((SP)), (SP) ← (SP) +2	–	–	–	–	–	–	–	–	–	–
POPW PS	1	3	(c)	word (PS) ← ((SP)), (SP) ← (SP) +2	–	–	*	*	*	*	*	*	*	–
POPW rlst	2	*2	*4	(rlst) ← ((SP)) , (SP) ← (SP)	–	–	–	–	–	–	–	–	–	–
JCTX @A	1	9	6× (c)	Context switch instruction	–	–	*	*	*	*	*	*	*	–
AND CCR, #imm8	2	3	0	byte (CCR) ← (CCR) and imm8	–	–	*	*	*	*	*	*	*	–
OR CCR, #imm8	2	3	0	byte (CCR) ← (CCR) or imm8	–	–	*	*	*	*	*	*	*	–
MOV RP, #imm8	2	2	0	byte (RP) ←imm8	–	–	–	–	–	–	–	–	–	–
MOV ILM, #imm8	2	2	0	byte (ILM) ←imm8	–	–	–	–	–	–	–	–	–	–
MOVEA RWi, ear	2	3	0	word (RWi) ←ear	–	–	–	–	–	–	–	–	–	–
MOVEA RWi, eam	2+	2+ (a)	0	word (RWi) ←eam	–	–	–	–	–	–	–	–	–	–
MOVEA A, ear	2	2	0	word(A) ←ear	–	*	–	–	–	–	–	–	–	–
MOVEA A, eam	2+	1+ (a)	0	word (A) ←eam	–	*	–	–	–	–	–	–	–	–
ADDSP #imm8	2	3	0	word (SP) ← ext (imm8)	–	–	–	–	–	–	–	–	–	–
ADDSP #imm16	3	3	0	word (SP) ← imm16	–	–	–	–	–	–	–	–	–	–
MOV A, brgl	2	*1	0	byte (A) ← (brgl)	Z	*	–	–	–	*	*	–	–	–
MOV brg2, A	2	1	0	byte (brg2) ← (A)	–	–	–	–	–	*	*	–	–	–
MOV brg2, #imm8	3	2	0	byte (brg2) ← imm8	–	–	–	–	–	*	*	–	–	–
NOP	1	1	0	No operation	–	–	–	–	–	–	–	–		–
ADB	1	1	0	Prefix code for AD space access	–	–	–	–	–	–	–	–	–	–
DTB	1	1	0	Prefix code for DT space access	–	–	–	–	–	–	–	–	–	–
PCB	1	1	0	Prefix code for PC space access	–	–	–	–	–	–	–	–	–	–
SPB	1	1	0	Prefix code for SP space access	–	–	–	–	–	–	–	–	–	–
NCC	1	1	0	Prefix code for no flag change	–	–	–	–	–	–	–	–	–	–
CMR	1	1	0	Prefix code for the common register bank	–	–	–	–	–	–	–	–	–	–
MOVW SPCU, #imm16	4	2	0	word (SPCU) ← (imm16)	–	–	–	–	–	–	–	–	–	–
MOVW SPCL, #imm16	4	2	0	word (SPCL) ← (imm16)	–	–	–	–	–	–	–	–	–	–
SETSPC	2	2	0	Stack check ooperation enable	–	–	–	–	–	–	–	–	–	–
CLRSPC	2	2	0	Stack check ooperation disable	–	–	–	–	–	–	–	–	–	–
BTSCN A	2	*5	0	byte (A)← position of "1" bit in word (A)	Z	–	–	–	–	–	*	–	–	–
BTSCNS A	2	*6	0	byte (A)← position of "1" bit in word (A) ×2	Z	–	–	–	–	–	*	–	–	–
BTSCND A	2	*7	0	byte (A)← position of "1" bit in word (A) ×4	Z	–	–	–	–	–	*	–	–	–

For an explanation of "(a)" and "(c)", refer to Tables 4 and 5.

*1: PCB, ADB, SSB, USB, and SPB: 1 cycle
DTB: 2 cycles
DPR: 3 cycles

*2: 3 + 4 × (pop count)

*3: 3 + 4 × (push count)

*4: Pop count × (c), or push count × (c)

*5: 3 when AL is 0, 5 when AL is not 0.

*6: 4 when AL is 0, 6 when AL is not 0.

*7: 5 when AL is 0, 7 when AL is not 0.

Table 23 Bit Manipulation Instructions [21 Instructions]

Mnemonic		#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVB	A, dir:bp	3	3	(b)	byte (A) ← (dir:bp) b	Z	*	–	–	–	*	*	–	–	–
MOVB	A, addr16:bp	4	3	(b)	byte (A) ← (addr16:bp) b	Z	*	–	–	–	*	*	–	–	–
MOVB	A, io:bp	3	3	(b)	byte (A) ← (io:bp) b	Z	*	–	–	–	*	*	–	–	–
MOVB	dir:bp, A	3	4	2× (b)	bit (dir:bp) b ← (A)	–	–	–	–	–	*	*	–	–	*
MOVB	addr16:bp, A	4	4	2× (b)	bit (addr16:bp) b ← (A)	–	–	–	–	–	*	*	–	–	*
MOVB	io:bp, A	3	4	2× (b)	bit (io:bp) b ← (A)	–	–	–	–	–	*	*	–	–	*
SETB	dir:bp	3	4	2× (b)	bit (dir:bp) b ← 1	–	–	–	–	–	–	–	–	–	*
SETB	addr16:bp	4	4	2× (b)	bit (addr16:bp) b ← 1	–	–	–	–	–	–	–	–	–	*
SETB	io:bp	3	4	2× (b)	bit (io:bp) b ← 1	–	–	–	–	–	–	–	–	–	*
CLRB	dir:bp	3	4	2× (b)	bit (dir:bp) b ← 0	–	–	–	–	–	–	–	–	–	*
CLRB	addr16:bp	4	4	2× (b)	bit (addr16:bp) b ← 0	–	–	–	–	–	–	–	–	–	*
CLRB	io:bp	3	4	2× (b)	bit (io:bp) b ← 0	–	–	–	–	–	–	–	–	–	*
BBC	dir:bp, rel	4	*1	(b)	Branch when (dir:bp) b = 0	–	–	–	–	–	–	*	–	–	–
BBC	addr16:bp, rel	5	*1	(b)	Branch when (addr16:bp) b = 0	–	–	–	–	–	–	*	–	–	–
BBC	io:bp, rel	4	*1	(b)	Branch when (io:bp) b = 0	–	–	–	–	–	–	*	–	–	–
BBS	dir:bp, rel	4	*1	(b)	Branch when (dir:bp) b = 1	–	–	–	–	–	–	*	–	–	–
BBS	addr16:bp, rel	5	*1	(b)	Branch when (addr16:bp) b = 1	–	–	–	–	–	–	*	–	–	–
BBS	io:bp, rel	4	*1	(b)	Branch when (io:bp) b = 1	–	–	–	–	–	–	*	–	–	–
SBBS	addr16:bp, rel	5	*2	2× (b)	Branch when (addr16:bp) b = 1, bit = 1	–	–	–	–	–	–	*	–	–	*
WBTS	io:bp	3	*3	*4	Wait until (io:bp) b = 1	–	–	–	–	–	–	–	–	–	–
WBTC	io:bp	3	*3	*4	Wait until (io:bp) b = 0	–	–	–	–	–	–	–	–	–	–

For an explanation of “(b)”, refer to Table 5, “Correction Values for Number of Cycles Used to Calculate Number of Actual Cycles.”
*1: 5 when branching, 4 when not branching
*2: 7 when condition is satisfied, 6 when not satisfied
*3: Undefined count
*4: Until condition is satisfied

Table 24 Accumulator Manipulation Instructions (Byte/Word) [6 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
SWAP	1	3	0	byte (A) 0 to 7 ← → (A) 8 to 15	–	–	–	–	–	–	–	–	–	–
SWAPW	1	2	0	word (AH) ← → (AL)	–	*	–	–	–	–	–	–	–	–
EXT	1	1	0	Byte code extension	X	–	–	–	–	*	*	–	–	–
EXTW	1	2	0	Word code extension	–	X	–	–	–	*	*	–	–	–
ZEXT	1	1	0	Byte zero extension	Z	–	–	–	–	R	*	–	–	–
ZEXTW	1	2	0	Word zero extension	–	Z	–	–	–	R	*	–	–	–

Table 25 String Instructions [10 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVS/MOVSI	2	*2	*3	Byte transfer @AH+ ← @AL+, counter = RW0	–	–	–	–	–	–	–	–	–	–
MOVSD	2	*2	*3	Byte transfer @AH– ← @AL–, counter = RW0	–	–	–	–	–	–	–	–	–	–
SCEQ/SCEQI	2	*1	*4	Byte retrieval @AH+ – AL, counter = RW0	–	–	–	–	–	*	*	*	*	–
SCEQD	2	*1	*4	Byte retrieval @AH– – AL, counter = RW0	–	–	–	–	–	*	*	*	*	–
FILS/FILSI	2	5m +3	*5	Byte filling @AH+ ← AL, counter = RW0	–	–	–	–	–	*	*	–	–	–
MOVSW/MOVSWI	2	*2	*6	Word transfer @AH+ ← @AL+, counter = RW0	–	–	–	–	–	–	–	–	–	–
MOVSWD	2	*2	*6	Word transfer @AH– ← @AL–, counter = RW0	–	–	–	–	–	–	–	–	–	–
SCWEQ/SCWEQI	2	*1	*7	Word retrieval @AH+ – AL, counter = RW0	–	–	–	–	–	*	*	*	*	–
SCWEQD	2	*1	*7	Word retrieval @AH– – AL, counter = RW0	–	–	–	–	–	*	*	*	*	–
FILSW/FILSWI	2	5m +3	*8	Word filling @AH+ ← AL, counter = RW0	–	–	–	–	–	*	*	–	–	–

m: RW0 value (counter value)
*1: 3 when RW0 is 0, 2 + 6 × (RW0) for count out, and 6n + 4 when match occurs
*2: 4 when RW0 is 0, 2 + 6 × (RW0) in any other case
*3: (b) × (RW0)
*4: (b) × n
*5: (b) × (RW0)
*6: (c) × (RW0)
*7: (c) × n
*8: (c) × (RW0)

Table 26 Multiple Data Transfer Instructions [18 Instructions]

Mnemonic	#	cycles	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVM @A, @RLi, #imm8	3	*1	*3	Multiple data trasfer byte ((A)) ← ((RLi))	–	–	–	–	–	–	–	–	–	–
MOVM @A, eam, #imm8	3+	*2	*3	Multiple data trasfer byte ((A)) ← (eam)	–	–	–	–	–	–	–	–	–	–
MOVM addr16, @RLi, #imm8	5	*1	*3	Multiple data trasfer byte (addr16) ← ((RLi))	–	–	–	–	–	–	–	–	–	–
MOVM addr16, eam, #imm8	5+	*2	*3	Multiple data trasfer byte (addr16) ← (eam)	–	–	–	–	–	–	–	–	–	–
MOVMW @A, @RLi, #imm8	3	*1	*4	Multiple data trasfer word ((A)) ← ((RLi))	–	–	–	–	–	–	–	–	–	–
MOVMW @A, eam, #imm8	3+	*2	*4	Multiple data trasfer word ((A)) ← (eam)	–	–	–	–	–	–	–	–	–	–
MOVMW addr16, @RLi, #imm8	5	*1	*4	Multiple data trasfer word (addr16) ← ((RLi))	–	–	–	–	–	–	–	–	–	–
MOVMW addr16, eam, #imm8	5+	*2	*4	Multiple data trasfer word (addr16) ← (eam)	–	–	–	–	–	–	–	–	–	–
MOVM @RLi, @A, #imm8	3	*1	*3	Multiple data trasfer byte ((RLi)) ← ((A))	–	–	–	–	–	–	–	–	–	–
MOVM eam, @A, #imm8	3+	*2	*3	Multiple data trasfer byte (eam) ← ((A))	–	–	–	–	–	–	–	–	–	–
MOVM @RLi, addr16, #imm8	5	*1	*3	Multiple data transfer byte ((RLi)) ← (addr16)	–	–	–	–	–	–	–	–	–	–
MOVM eam, addr16, #imm8	5+	*2	*3	Multiple data transfer byte (eam) ← (addr16)	–	–	–	–	–	–	–	–	–	–
MOVMW @RLi, @A, #imm8	3	*1	*4	Multiple data trasfer word ((RLi)) ← ((A))	–	–	–	–	–	–	–	–	–	–
MOVMW eam, @A, #imm8	3+	*2	*4	Multiple data trasfer word (eam) ← ((A))	–	–	–	–	–	–	–	–	–	–
MOVMW @RLi, addr16, #imm8	5	*1	*4	Multiple data transfer word ((RLi)) ← (addr16)	–	–	–	–	–	–	–	–	–	–
MOVMW eam, addr16, #imm8	5+	*2	*4	Multiple data transfer word (eam) ← (addr16)	–	–	–	–	–	–	–	–	–	–
MOVM bnk : addr16, *5 bnk : addr16, #imm8	7	*1	*3	Multiple data transfer byte (bnk:addr16) ← (bnk:addr16)	–	–	–	–	–	–	–	–	–	–
MOVMW bnk : addr16, *5 bnk : addr16, #imm8	7	*1	*4	Multiple data transfer word (bnk:addr16) ← (bnk:addr16)	–	–	–	–	–	–	–	–	–	–

*1: 5 + imm8 × 5, 256 times when imm8 is zero.
*2: 5 + imm8 × 5 + (a), 256 times when imm8 is zero.
*3: Number of transfers × (b) × 2
*4: Number of transfers × (c) × 2
*5: The bank register specified by “bnk” is the same as for the MOVS instruction.

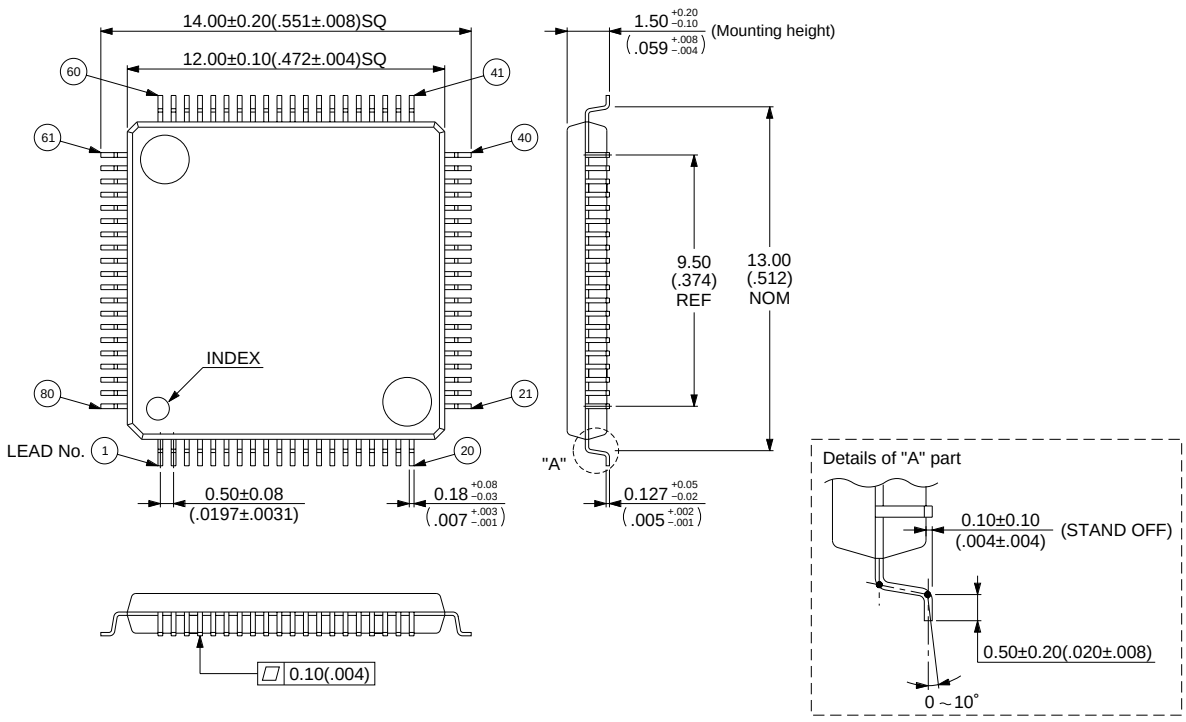
■ ORDERING INFORMATION

Part number	Package	Remarks
MB90242A	80-pin Plastic LQFP (FPT-80P-M05)	

MB90242A Series

■ PACKAGE DIMENSIONS

80-pin Plastic LQFP
(FPT-80P-M05)



Dimensions in mm (inches)

MB90242A Series

FUJITSU LIMITED

For further information please contact:

Japan
FUJITSU LIMITED
Corporate Global Business Support Division
Electronic Devices
KAWASAKI PLANT, 4-1-1, Kamikodanaka
Nakahara-ku, Kawasaki-shi
Kanagawa 211-88, Japan
Tel: (044) 754-3763
Fax: (044) 754-3329

North and South America
FUJITSU MICROELECTRONICS, INC.
Semiconductor Division
3545 North First Street
San Jose, CA 95134-1804, U.S.A.
Tel: (408) 922-9000
Fax: (408) 432-9044/9045

Europe
FUJITSU MIKROELEKTRONIK GmbH
Am Siebenstein 6-10
63303 Dreieich-Buchschlag
Germany
Tel: (06103) 690-0
Fax: (06103) 690-122

Asia Pacific
FUJITSU MICROELECTRONICS ASIA PTE. LIMITED
#05-08, 151 Lorong Chuan
New Tech Park
Singapore 556741
Tel: (65) 281-0770
Fax: (65) 281-0220

F9704

The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.

The information and circuit diagrams in this document presented as examples of semiconductor device applications, and are not intended to be incorporated in devices for actual use. Also, FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of this information or circuit diagrams.

FUJITSU semiconductor devices are intended for use in standard applications (computers, office automation and other office equipment, industrial, communications, and measurement equipment, personal or household devices, etc.).
CAUTION:
Customers considering the use of our products in special applications where failure or abnormal operation may directly affect human lives or cause physical injury or property damage, or where extremely high levels of reliability are demanded (such as aerospace systems, atomic energy controls, sea floor repeaters, vehicle operating controls, medical devices for life support, etc.) are requested to consult with FUJITSU sales representatives before such use. The company will not be responsible for damages arising from such use without prior approval.

Any semiconductor devices have inherently a certain rate of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Control Law of Japan, the prior authorization by Japanese government should be required for export of those products from Japan.