To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

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

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

78K0/KB2

- μPD78F0500A, 78F0501A, 78F0502A, 78F0503A, 78F0503DA -

One-sheet Manual
Compact and Easily Readable

Outline of Functions

	Item		μPD78F0500A	μPD78F0501A	μPD78F0502A	μPD78F0503A	μPD78F0503DA
Internal memory	Flash memory (self-programming supported)		8 KB	16 KB	24 KB	32 KB	
	High-speed RAM		512 bytes	768 bytes	1 KB		
Memory space			64 KB				
Main system clock	High-speed system clock		X1(crystal/ceramic) oscillation, external system clock input (EXCLK) 1 to 20 MHz: VDD = 4.0 to 5.5 V, 1 to 10 MHz: VDD = 2.7 to 5.5 V, 1 to 5 MHz: VDD = 1.8 to 5.5 V				
	Internal high-speed oscillation clock		Internal oscillation 8 MHz (TYP.) : VDD = 1.8 to 5.5 V				
Internal low-speed oscillation clock			Internal oscillation 240 kHz (TYP.) : VDD = 1.8 to 5.5 V				
General-purpose registers			8 bits x 32 registers (8 bits x 8 registers x 4 banks)				
Minimum instruction execution time			0.1 μs (high-speed system clock: @ fXH = 20 MHz operation) 0.25 μs (internal high-speed oscillation clock: @ fRH = 8 MHz (TYP.) operation)				
Instruction set			• 8-bit operation, 16-bit operation • Multiply/divide (8 bits x 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc				
I/O ports			Total 23 • CMOS I/O: 21 • N-ch open-drain I/O (6.0 V tolerance) : 2				
Timers			• 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 2 channels • 8-bit timer: 2 channels • Watchdog timer: 1 channel				
	Timer outputs		5 (PWM output: 4, PPG output: 1)				
A/D converter			10-bit resolution x 4 channels (AVREF = 2.3 to 5.5 V)				
Serial interface			• UART supporting LIN-bus: 1 channel • 3-wire serial I/O/UART: 1 channel • I²C bus: 1 channel				
Vectored interrupt sources	Internal		14				
	External		6				
Reset			• Reset using RESET pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector				
On-chip debug function			None			Provided	
Power supply voltage			VDD = 1.8 to 5.5 V				
Operating ambient temperature			TA = −40 to +85°C				
Package			• 30-pin plastic SSOP (7.62 mm (300)) • 36-pin plastic FLGA (4x4)				

Remark Pin names in parentheses [ ] are only for μPD78F0503DA (products with on-chip debug function).

Document No. U17983EJ2V0IF00 (2nd edition)
Date Published February 2008 NS

Development Tools (1/5)

Remark For details about development tools, see the site for development tools at NEC Electronics Website.
NEC Electronics Website: http://www.necel.com/

(1) Software Tools

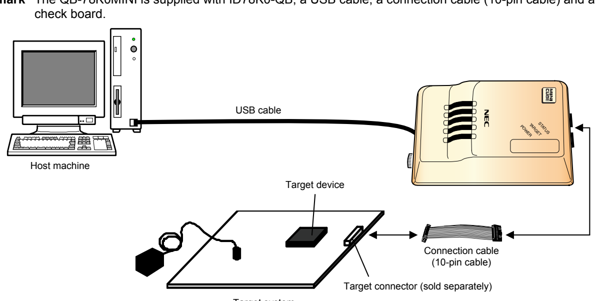
Host Machine	Software Tools
IBM PC/AT™ compatibles, PC98-NX series	Software package SP78K0
	Assembler package RA78K0
	C compiler CC78K0
	C library source file CC78K0-L
	Integrated debugger ID78K0-QB
	System simulator SM+ for 78K0/Kx2
	Device file DF780547

(2) Hardware Tools (1/3)

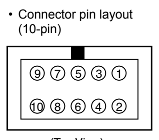
<1> On-chip debug emulator QB-78K0MINI (MINICUBE®)

On-chip Debug Emulator	Target Connector Specifications
QB-78K0MINI	10-pin general-purpose connector (2.54 mm pitch)

Remark The QB-78K0MINI is supplied with ID78K0-QB, a USB cable, a connection cable (10-pin cable) and a self-check board.



• Connector pin layout (10-pin)



(Top View)

• Connector pin configuration (10-pin)

Pin No.	Pin Name Note 1	IN/OUT Note 2	Description
1	RESET_IN	IN	Pin used to input reset signal from the target system
2	$\overline{\text{RESET\_OUT}}$	OUT	Pin used to output reset signal to the target device
3	FLMD0	OUT	Output pin used to control on-chip debugging functions
4	VDD	IN	Input pin for when using power supply of the target system
5	X2/DATA	IN/OUT	Pin used to input/output for data communication during debugging
6	GND	–	Connected to GND.
7	X1/CLK	OUT	Pin used to output clock signal to the target device
8	GND	–	Connected to GND.
9	RESERVED	–	Open
10	RESERVED	–	Open

Notes 1. Signal names in MINICUBE
2. As seen from MINICUBE.

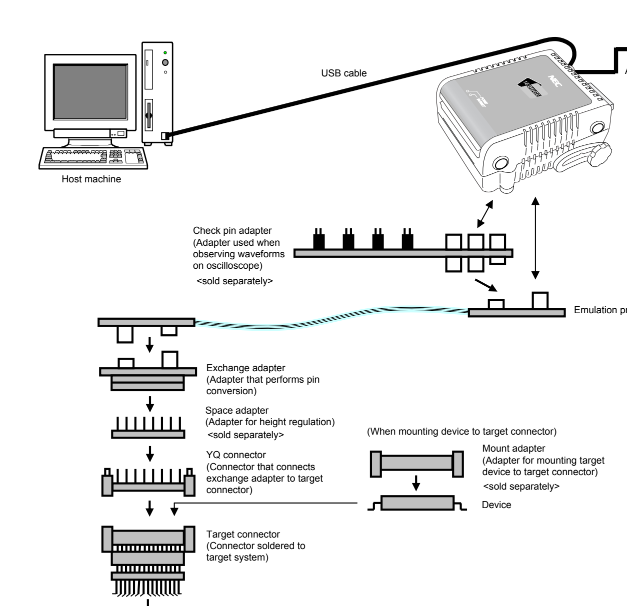
Development Tools (2/5)

(2) Hardware Tools (2/3)

<2> In-circuit emulator QB-78K0KX2 (IECUBE®)

In-Circuit Emulator	Package	Check Pin Adapter	Emulation Probe	Exchange Adapter	Space Adapter	YQ Connector	Mount Adapter	Target connector
QB-78K0KX2	30-pin plastic SSOP (7.62 mm (300))	QB-144-CA-01	QB-80-EP-01T	QB-30MC-EA-02T	QB-30MC-YQ-01T	QB-30MC-YQ-01T	QB-30MC-HQ-01T	QB-30MC-NQ-01T
	36-pin plastic FLGA (4x4)			QB-36FC-EA-01T	–	–	–	QB-36FC-NQ-01T

Remark The QB-78K0KX2 is supplied with ID78K0-QB, a USB cable, a power supply unit, QB-MINI2, connection cables (10-pin cable and 16-pin cable) and the 78K0-OCD board.



Development Tools (3/5)

(2) Hardware Tools (3/3)

<3> On-chip debug emulator with programming function QB-MINI2 (MINICUBE2) : for on-chip debugging

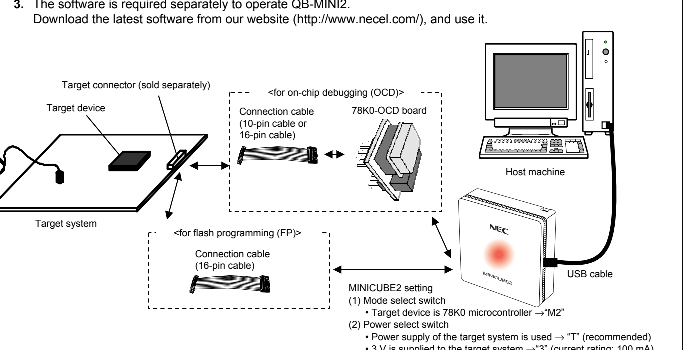
On-Chip Debug Emulator with Programming Function	Target Connector Specifications	
QB-MINI2	10-pin general-purpose connector (2.54 mm pitch)	When using 10-pin cable
	16-pin general-purpose connector (2.54 mm pitch)	When using 16-pin cable

(3) Flash Memory Write Tools (1/3)

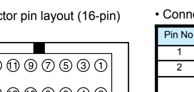
<1> On-chip debug emulator with programming function QB-MINI2 (MINICUBE2) : for flash programming

On-Chip Debug Emulator with Programming Function	Target Connector Specifications	
QB-MINI2	16-pin general-purpose connector (2.54 mm pitch)	When using 16-pin cable

Remarks 1. The QB-MINI2 is supplied with a USB cable, connection cables (10-pin cable and 16-pin cable) and the 78K0-OCD board.
2. A connection cable (10-pin cable) and the 78K0-OCD board are used only when using the on-chip debug function.
3. The software is required separately to operate QB-MINI2.
Download the latest software from our website (http://www.necel.com/), and use it.



• Connector pin layout (16-pin)



(Top View)

• Connector pin configuration (16-pin)

Pin No.	Pin Name Note 1	IN/OUT Note 2	Description
1	GND	–	Connected to GND.
2	$\overline{\text{RESET\_OUT}}$	OUT	Pin used to output reset signal to the target device
3	RESERVED (during OCD)	–	Open
	RxD (during FP)	IN	Pin used to input for command/data communication from the target device
4	VDD	IN	Input pin for when using power supply of the target system
5	RESERVED (during OCD)	–	Open
	TxD (during FP)	OUT	Pin used to output for command/data communication to the target device
6-8	RESERVED	–	Open
9	CLK	OUT	Pin used to output clock signal to the target device
10-12	RESERVED	–	Open
13	DATA (during OCD)	IN/OUT	Pin used to input/output for data communication during debugging
	RESERVED (during FP)	–	Open
14	FLMD0	OUT	Pin used to set to debug mode or programming mode
15	RESET_IN (during OCD)	IN	Pin used to input reset signal from the target system
	RESERVED (during FP)	–	Open
16	RESERVED	–	Open

Notes 1. Signal names in MINICUBE2
2. As seen from MINICUBE2

Remark The 10-pin target connector is the same as it of MINICUBE. See the description of the target connector of MINICUBE.

Development Tools (4/5)

(3) Flash Memory Write Tools (2/3)

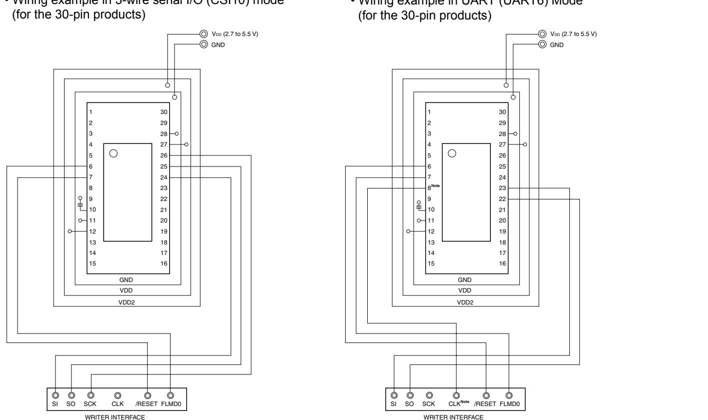
<2> Flash memory programmer PG-FP5, FL-PR5, PG-FP4, FL-PR4, PG-FPL3, FP-LITE3 (1/2)

Flash Memory Programmer	Flash Memory Write Adapter	
Flash memory programmer PG-FP5, FL-PR5, PG-FP4, FL-PR4 Simple flash memory programmer PG-FPL3 Note, FP-LITE3	FA-30MC-5A4-A, FA-78F0503MC-5A4-MX	30-pin plastic SSOP (7.62 mm (300))
	FA-78F0503FC-AA3-MX	36-pin plastic FLGA (4x4)

Note Phase-out

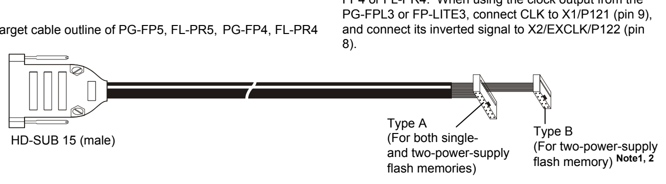
Remarks 1. FL-PR5, FL-PR4, FP-LITE3, FA-64GB-8BU-A, FA-78F0537GB-UEU-MX, FA-64GC-8BS-A, FA-78F0537GC-UBS-MX, FA-64GK-9ET-A, FA-78F0537GK-UET-MX, FA-78F0537GA-8EV-MX, and FA-78F0537FC-AA1-MX are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-42-750-4172 Naito Densei Machida Mfg. Co., Ltd.
2. Use the latest version of the flash memory programming adapter.

• Wiring example in 3-wire serial I/O (CSI10) mode (for the 30-pin products) • Wiring example in UART (UART6) Mode (for the 30-pin products)



Note The above figure illustrates an example of wiring when using the clock output from the PG-FP5, FL-PR5, PG-FP4 or FL-PR4. When using the clock output from the PG-FPL3 or FP-LITE3, connect CLK to X1/P121 (pin 9), and connect its inverted signal to X2/EXCLK/P122 (pin 8).

• Target cable outline of PG-FP5, FL-PR5, PG-FP4, FL-PR4



Notes 1. The target cable of PG-FP5 and FL-PR5 is not equipped with Type B.
2. Type B is not used to connect with 78K0/KB2 because 78K0/KB2 incorporates the single-power-supply flash memory.

Development Tools (5/5)

(3) Flash Memory Write Tools (3/3)

<2> Flash memory programmer PG-FP5, FL-PR5, PG-FP4, FL-PR4, PG-FPL3, FP-LITE3 (2/2)

• Connector pin layout of PG-FP5, FL-PR5, PG-FP4 and FL-PR4 (view from socket side)

1	3	5	7	9	11	13	15
2	4	6	8	10	12	14	16

Type A (16-pin)

• Connector pin configuration of PG-FP5, FL-PR5, PG-FP4 and FL-PR4

Signal Name of PG-FP4	Target Connector Type A: Signal (16-Pin)
GND	1
$\overline{\text{RESET}}$	2
SI/RxD	3
VDD	4
SO/TxD	5
(VPP) Note	6
SCK	7
(H/S) Note	8
CLK	9
(VDE) Note	10
(VDD2) Note	11
(FLMD1) Note	12
(RFU-1) Note	13
FLMD0	14
(Not used) Note	15, 16

Note Signals in parentheses and the corresponding pins are not used with 78K0/KB2.

Operation List (1/6)

• Operand identifiers and specification methods

Identifier	Specification Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special function register symbol Note Special function register symbol (16-bit manipulatable register even addresses only) Note
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1FH Immediate data or labels (even address only)
addr16 addr11 addr5	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions) 0800H to 0FFFH Immediate data or labels 0040H to 007FH Immediate data or labels (even address only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Operation List (2/6)

• Description of operation column

A: A register (8-bit accumulator), X: X register, B: B register, C: C register, D: D register, E: E register, H: H register, L: L register, AX: AX register pair (16-bit accumulator), BC: BC register pair, DE: DE register pair, HL: HL register pair, PC: Program counter, SP: Stack pointer, PSW: Program status word, CY: Carry flag, AC: Auxiliary carry flag, Z: Zero flag, RBS: Register bank select flag, IE: Interrupt request enable flag,
( ): Memory contents indicated by address or register contents in parentheses,
X_H, X_L: Higher 8 bits and lower 8 bits of 16-bit register, ∧: Logical product (AND), ∨: Logical sum (OR),
⊻: Exclusive logical sum (exclusive OR), ¯¯¯: Inverted data, addr16: 16-bit immediate data or label,
jdisp8: Signed 8-bit data (displacement value)

• Description of flag operation column

(Blank): Not affected, 0: Cleared to 0, 1: Set to 1, ×: Set/cleared according to the result, R: Previously saved value is restored

Instruction Group	Mnemonic	Operands	Bytes	Clocks Note 1	Clocks Note 2	Operation	Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	4	–	r ← byte			
		saddr, #byte	3	6	7	(saddr) ← byte			
		sfr, #byte	3	–	7	sfr ← byte			
		A, r Note 3	1	2	–	A ← r			
		r, A Note 3	1	2	–	r ← A			
		A, saddr	2	4	5	A ← (saddr)			
		saddr, A	2	4	5	(saddr) ← A			
		A, sfr	2	–	5	A ← sfr			
		sfr, A	2	–	5	sfr ← A			
		A, !addr16	3	8	9	A ← (addr16)			
		!addr16, A	3	8	9	(addr16) ← A			
		PSW, #byte	3	–	7	PSW ← byte	×	×	×
		A, PSW	2	–	5	A ← PSW			
		PSW, A	2	–	5	PSW ← A	×	×	×
		A, [DE]	1	4	5	A ← (DE)			
		[DE], A	1	4	5	(DE) ← A			
		A, [HL]	1	4	5	A ← (HL)			
		[HL], A	1	4	5	(HL) ← A			
		A, [HL+byte]	2	8	9	A ← (HL+byte)			
		[HL+byte], A	2	8	9	(HL+byte) ← A			
		A, [HL+B]	1	6	7	A ← (HL+B)			
		[HL+B], A	1	6	7	(HL+B) ← A			
		A, [HL+C]	1	6	7	A ← (HL+C)			
		[HL+C], A	1	6	7	(HL+C) ← A			
	XCH	A, r Note 3	1	2	–	A ←→ r			
		A, saddr	2	4	6	A ←→ (saddr)			
		A, sfr	2	–	6	A ←→ sfr			
		A, !addr16	3	8	10	A ←→ (addr16)			
		A, [DE]	1	4	6	A ←→ (DE)			
		A, [HL]	1	4	6	A ←→ (HL)			
		A, [HL+byte]	2	8	10	A ←→ (HL+byte)			
		A, [HL+B]	2	8	10	A ←→ (HL+B)			
		A, [HL+C]	2	8	10	A ←→ (HL+C)			
16-bit data transfer	MOVW	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access
2. When an area except the internal high-speed RAM area is accessed
3. Except "r = A"

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
2. This clock cycle applies to the internal ROM program.

Operation List (3/6)

Instruction Group	Mnemonic	Operands	Bytes	Clocks Note 1	Clocks Note 2	Operation	Z	AC	CY
16-bit data transfer	MOVW	AX, rp Note 3	1	4	–	AX ← rp			
		rp, AX Note 3	1	4	–	rp ← AX			
		AX, !addr16	3	10	12	AX ← (addr16)			
		!addr16, AX	3	10	12	(addr16) ← AX			
	XCHW	AX, rp Note 3	1	4	–	AX ←→ rp			
8-bit operation	ADD	A, #byte	2	4	–	A, CY ← A+byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) +byte	×	×	×
		A, r Note 4	2	4	–	A, CY ← A+r	×	×	×
		r, A	2	4	–	r, CY ← r+A	×	×	×
		A, saddr	2	4	5	A, CY ← A+ (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A+ (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A+ (HL)	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A+ (HL+byte)	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A+ (HL+B)	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A+ (HL+C)	×	×	×
	ADDC	A, #byte	2	4	–	A, CY ← A+byte+CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) +byte+CY	×	×	×
		A, r Note 4	2	4	–	A, CY ← A+r+CY	×	×	×
		r, A	2	4	–	r, CY ← r+A+CY	×	×	×
		A, saddr	2	4	5	A, CY ← A+ (saddr) +CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A+ (addr16) +CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A+ (HL) +CY	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A+ (HL+byte) +CY	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A+ (HL+B) +CY	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A+ (HL+C) +CY	×	×	×
	SUB	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 4	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A – (HL+byte)	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A – (HL+B)	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A – (HL+C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 4	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL+byte]	2	8	9	A, CY ← A – (HL+byte) – CY	×	×	×
		A, [HL+B]	2	8	9	A, CY ← A – (HL+B) – CY	×	×	×
		A, [HL+C]	2	8	9	A, CY ← A – (HL+C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 4	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∧ (HL)	×		
		A, [HL+byte]	2	8	9	A ← A ∧ (HL+byte)	×		
		A, [HL+B]	2	8	9	A ← A ∧ (HL+B)	×		
		A, [HL+C]	2	8	9	A ← A ∧ (HL+C)	×		

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access
2. When an area except the internal high-speed RAM area is accessed
3. Only when rp = BC, DE or HL
4. Except "r = A"

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
2. This clock cycle applies to the internal ROM program.

Operation List (4/6)

Instruction Group	Mnemonic	Operands	Bytes	Clocks Note 1	Clocks Note 2	Operation	Z	AC	CY
8-bit operation	OR	A, #byte	2	4	–	A ← A ∨ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	4	–	A ← A ∨ r	×		
		r, A	2	4	–	r ← r ∨ A	×		
		A, saddr	2	4	5	A ← A ∨ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∨ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∨ (HL)	×		
		A, [HL+byte]	2	8	9	A ← A ∨ (HL+byte)	×		
		A, [HL+B]	2	8	9	A ← A ∨ (HL+B)	×		
		A, [HL+C]	2	8	9	A ← A ∨ (HL+C)	×		
	XOR	A, #byte	2	4	–	A ← A ⊻ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⊻ byte	×		
		A, r Note 3	2	4	–	A ← A ⊻ r	×		
		r, A	2	4	–	r ← r ⊻ A	×		
		A, saddr	2	4	5	A ← A ⊻ (saddr)	×		
		A, !addr16	3	8	9	A ← A ⊻ (addr16)	×		
		A, [HL]	1	4	5	A ← A ⊻ (HL)	×		
		A, [HL+byte]	2	8	9	A ← A ⊻ (HL+byte)	×		
		A, [HL+B]	2	8	9	A ← A ⊻ (HL+B)	×		
		A, [HL+C]	2	8	9	A ← A ⊻ (HL+C)	×		
	CMP	A, #byte	2	4	–	A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A – r	×	×	×
		r, A	2	4	–	r – A	×	×	×
		A, saddr	2	4	5	A – (saddr)	×	×	×
		A, !addr16	3	8	9	A – (addr16)	×	×	×
		A, [HL]	1	4	5	A – (HL)	×	×	×
		A, [HL+byte]	2	8	9	A – (HL+byte)	×	×	×
		A, [HL+B]	2	8	9	A – (HL+B)	×	×	×
		A, [HL+C]	2	8	9	A – (HL+C)	×	×	×
16-bit operation	ADDW	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/divide	MULU	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/decrement	INC	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp+1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	ROR	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access
2. When an area except the internal high-speed RAM area is accessed
3. Except "r = A"

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
2. This clock cycle applies to the internal ROM program.

Operation List (5/6)

Instruction Group	Mnemonic	Operands	Bytes	Clocks Note 1	Clocks Note 2	Operation	Z	AC	CY
BCD adjustment	ADJBA		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	MOV1	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL]. bit, CY	2	6	8	(HL).bit ← CY			
	AND1	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{CY}$			×
Call/return	CALL	!addr16	3	7	–	(SP – 1) ← $(PC+3)_H$, (SP – 2) ← $(PC+3)_L$, PC ← addr16, SP ← SP – 2			
	CALLF	!addr11	2	5	–	(SP – 1) ← $(PC+2)_H$, (SP – 2) ← $(PC+2)_L$, PC_{15-11} ← 00001, PC_{10-0} ← addr11, SP ← SP – 2			
	CALLT	[addr5]	1	6	–	(SP – 1) ← $(PC+1)_H$, (SP – 2) ← $(PC+1)_L$, PC_H ← (addr5+1), PC_L ← (addr5), SP ← SP – 2			
	BRK		1	6	–	(SP – 1) ← PSW, (SP – 2) ← $(PC+1)_H$, (SP – 3) ← $(PC+1)_L$, PC_H ← (003FH), PC_L ← (003EH), SP ← SP – 3, IE ← 0			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access
2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
2. This clock cycle applies to the internal ROM program.

Operation List (6/6)

Instruction Group	Mnemonic	Operands	Bytes	Clocks Note 1	Clocks Note 2	Operation	Z	AC	CY
Call/return	RET		1	6	–	PC_H ← (SP+1), PC_L ← (SP), SP ← SP+2			
	RETI		1	6	–	PC_H ← (SP+1), PC_L ← (SP), PSW ← (SP+2), SP ← SP+3	R	R	R
	RETB		1	6	–	PC_H ← (SP+1), PC_L ← (SP), PSW ← (SP+2), SP ← SP+3	R	R	R
Stack manipulate	PUSH	PSW	1	2	–	(SP – 1) ← PSW, SP ← SP – 1			
		rp	1	4	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2			
	POP	PSW	1	2	–	PSW ← (SP), SP ← SP+1	R	R	R
		rp	1	4	–	rp_H ← (SP – 1), rp_L ← (SP), SP ← SP+2			
	MOVW	SP, #word	4	–	10	SP ← word			
		SP, AX	2	–	8	SP ← AX			
		AX, SP	2	–	8	AX ← SP			
Unconditional branch	BR	!addr16	3	6	–	PC ← addr16			
		$addr16	2	6	–	PC ← PC + 2 + jdisp8			
		AX	2	8	–	PC_H ← A, PC_L ← X			
Conditional branch	BC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr16	2	6	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BT	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1			
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0			
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)			
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
		C, $addr16	2	6	–	C ← C – 1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
CPU control	SEL	RBn	2	4	–	RBS1, 0 ← n			
	NOP		1	2	–	No Operation			
	EI		2	–	6	IE ← 1 (Enable Interrupt)			
	DI		2	–	8	IE ← 0 (Disable Interrupt)			
	HALT		2	6	–	Set HALT Mode			
	STOP		2	6	–	Set STOP Mode			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access
2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).
2. This clock cycle applies to the internal ROM program.

Special Function Register (SFR) List (1/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	1 Bit	8 Bits	16 Bits	After Reset
FF00H	Port register 0	P0	R/W	√	√	–	00H
FF01H	Port register 1	P1	R/W	√	√	–	00H
FF02H	Port register 2	P2	R/W	√	√	–	00H
FF03H	Port register 3	P3	R/W	√	√	–	00H
FF06H	Port register 6	P6	R/W	√	√	–	00H
FF08H	10-bit A/D conversion result register	ADCR	R	–	–	√	0000H
FF09H	8-bit A/D conversion result register	ADCRH	R	–	√	–	00H
FF0AH	Receive buffer register 6	RXB6	R	–	√	–	FFH
FF0BH	Transmit buffer register 6	TXB6	R/W	–	√	–	FFH
FF0CH	Port register 12	P12	R/W	√	√	–	00H
FF0FH	Serial I/O shift register 10	SIO10	R	–	√	–	00H
FF10H	16-bit timer counter 00	TM00	R	–	–	√	0000H
FF11H							
FF12H	16-bit timer capture/compare register 000	CR000	R/W	–	–	√	0000H
FF13H							
FF14H	16-bit timer capture/compare register 010	CR010	R/W	–	–	√	0000H
FF15H							
FF16H	8-bit timer counter 50	TM50	R	–	√	–	00H
FF17H	8-bit timer compare register 50	CR50	R/W	–	√	–	00H
FF18H	8-bit timer H compare register 00	CMP00	R/W	–	√	–	00H
FF19H	8-bit timer H compare register 10	CMP10	R/W	–	√	–	00H
FF1AH	8-bit timer H compare register 01	CMP01	R/W	–	√	–	00H
FF1BH	8-bit timer H compare register 11	CMP11	R/W	–	√	–	00H
FF1FH	8-bit timer counter 51	TM51	R	–	√	–	00H
FF20H	Port mode register 0	PM0	R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1	R/W	√	√	–	FFH
FF22H	Port mode register 2	PM2	R/W	√	√	–	FFH
FF23H	Port mode register 3	PM3	R/W	√	√	–	FFH
FF26H	Port mode register 6	PM6	R/W	√	√	–	FFH
FF28H	A/D converter mode register	ADM	R/W	√	√	–	00H
FF29H	Analog input channel specification register	ADS	R/W	√	√	–	00H
FF2CH	Port mode register 12	PM12	R/W	√	√	–	FFH
FF2FH	A/D port configuration register	ADPC	R/W	√	√	–	00H
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
FF31H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
FF3CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H

Special Function Register (SFR) List (2/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	1 Bit	8 Bits	16 Bits	After Reset
FF41H	8-bit timer compare register 51	CR51	R/W	–	√	–	00H
FF43H	8-bit timer mode control register 51	TMC51	R/W	√	√	–	00H
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	–	00H
FF49H	External interrupt falling edge enable register	EGN	R/W	√	√	–	00H
FF4FH	Input switch control register	ISC	R/W	√	√	–	00H
FF50H	Asynchronous serial interface operation mode register 6	ASIM6	R/W	√	√	–	01H
FF53H	Asynchronous serial interface reception error status register 6	ASIS6	R	–	√	–	00H
FF55H	Asynchronous serial interface transmission status register 6	ASIF6	R	–	√	–	00H
FF56H	Clock selection register 6	CKSR6	R/W	–	√	–	00H
FF57H	Baud rate generator control register 6	BRGC6	R/W	–	√	–	FFH
FF58H	Asynchronous serial interface control register 6	ASICL6	R/W	√	√	–	16H
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
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
FF80H	Serial operation mode register 10	CSIM10	R/W	√	√	–	00H
FF81H	Serial clock selection register 10	CSIC10	R/W	√	√	–	00H
FF84H	Transmit buffer register 10	SOTB10	R/W	–	√	–	00H
FF8CH	Timer clock selection register 51	TCL51	R/W	√	√	–	00H
FF99H	Watchdog timer enable register	WDTE	R/W	–	√	–	1AH/9AH Note 1
FF9FH	Clock operation mode select register	OSCCTL	R/W	√	√	–	00H
FFA0H	Internal oscillation mode register	RCM	R/W	√	√	–	80H Note 2
FFA1H	Main clock mode register	MCM	R/W	√	√	–	00H
FFA2H	Main OSC control register	MOC	R/W	√	√	–	80H

Notes 1. The reset value of WDTE is determined by setting of option byte.
2. The value of this register is 00H immediately after a reset release but automatically changes to 80H after oscillation accuracy stabilization of high-speed internal oscillator has been waited.

Special Function Register (SFR) List (3/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	1 Bit	8 Bits	16 Bits	After Reset
FFA3H	Oscillation stabilization time counter status register	OSTC		R	√	√	–	00H
FFA4H	Oscillation stabilization time select register	OSTS		R/W	–	√	–	05H
FFA5H	IIC shift register 0	IIC0		R/W	–	√	–	00H
FFA6H	IIC control register 0	IICC0		R/W	√	√	–	00H
FFA7H	Slave address register 0	SVA0		R/W	–	√	–	00H
FFA8H	IIC clock selection register 0	IICCL0		R/W	√	√	–	00H
FFA9H	IIC function expansion register 0	IICX0		R/W	√	√	–	00H
FFAAH	IIC status register 0	IICS0		R	√	√	–	00H
FFABH	IIC flag register 0	IICF0		R/W	√	√	–	00H
FFACH	Reset control flag register	RESF		R	–	√	–	00H Note
FFBAH	16-bit timer mode control register 00	TMC00		R/W	√	√	–	00H
FFBBH	Prescaler mode register 00	PRM00		R/W	√	√	–	00H
FFBCH	Capture/compare control register 00	CRC00		R/W	√	√	–	00H
FFBDH	16-bit timer output control register 00	TOC00		R/W	√	√	–	00H
FFBEH	Low-voltage detection register	LVIM		R/W	√	√	–	00H Note
FFBFH	Low-voltage detection level selection register	LVIS		R/W	√	√	–	00H Note
FFE0H	Interrupt request flag register 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H	R/W	√	√		00H
FFE2H	Interrupt request flag register 1L	IF1	IF1L	R/W	√	√	√	00H
FFE3H	Interrupt request flag register 1H		IF1H	R/W	√	√		00H
FFE4H	Interrupt mask flag register 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H	R/W	√	√		FFH
FFE6H	Interrupt mask flag register 1L	MK1	MK1L	R/W	√	√	√	FFH
FFE7H	Interrupt mask flag register 1H		MK 1H	R/W	√	√		FFH
FFE8H	Priority specification flag register 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	Priority specification flag register 0H		PR0H	R/W	√	√		FFH
FFEAH	Priority specification flag register 1L	PR1	PR1L	R/W	√	√	√	FFH
FFEBH	Priority specification flag register 1H		PR1H	R/W	√	√		FFH

Note The reset values of RESF, LVIM, and LVIS vary depending on the reset source.

Special Function Register (SFR) List (4/4)

Address	Special Function Register (SFR) Name	Symbol	R/W	1 Bit	8 Bits	16 Bits	After Reset
FFF0H	Internal memory size switching register Note 1	IMS	R/W	–	√	–	CFH
FFFBH	Processor clock control register	PCC	R/W	√	√	–	01H

Notes 1. Regardless of the internal memory capacity, the initial values of the internal memory size switching register (IMS) of all products in the 78K0/KB2 are fixed (IMS = CFH). Therefore, set the value corresponding to each product as indicated below.

78K0/KB2	IMS	ROM Capacity	Internal High-Speed RAM Capacity
μPD78F0500A	42H	8 KB	512 bytes
μPD78F0501A	04H	16 KB	768 bytes
μPD78F0502A	C6H	24 KB	1 KB
μPD78F0503A, 78F0503DA Note 2	C8H	32 KB	

2. The ROM and RAM capacities of the products with the on-chip debug function can be debugged according to the debug target products. Set IMS according to the debug target products.

• This document is a reference.