Renesas Electronics America Inc - UPD70F3017AYGC-8EU-A Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3017aygc-8eu-a

Related Documents The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Documents related to V850/SA1

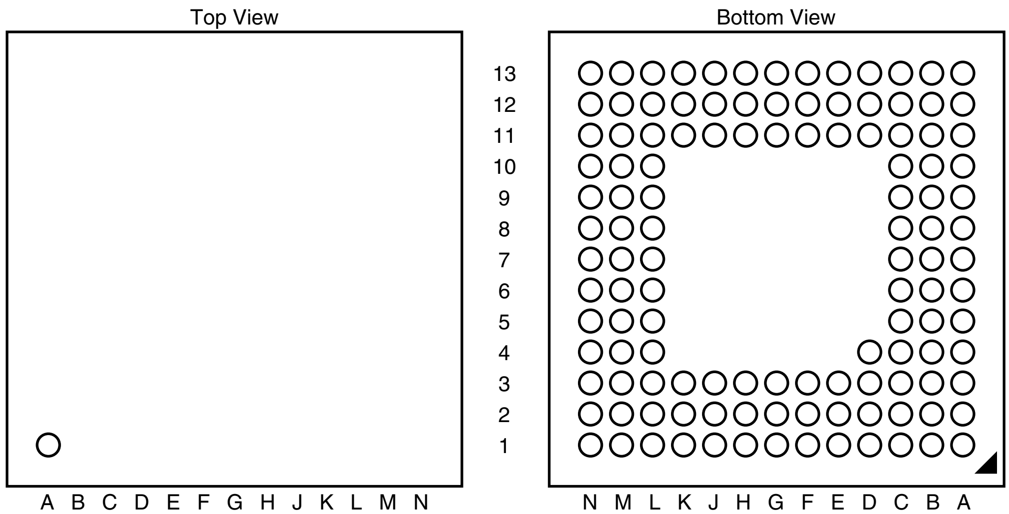
Document Name	Document No.
V850 Series Architecture User's Manual	U10243E
V850/SA1 Application Note	U13851E
V850/SA1 Hardware User's Manual	This manual
V850 Series Flash Memory Self Programming User's Manual	U15673E

Documents related to development tools (user's manuals)

Document Name		Document No.
IE-703002-MC (In-Circuit Emulator)		U11595E
IE-703017-MC-EM1 (In-Circuit Emulator Option Board)		U12898E
CA850 Ver. 2.40 or Later C Compiler Package	Operation	U15024E
	C Language	U15025E
	Project Manager	U15026E
	Assembly Language	U15027E
ID850 Ver. 2.40 Integrated Debugger	Operation (Windows™ Based)	U15181E
SM850 Ver. 2.40 System Simulator	Operation (Windows Based)	U15182E
SM850 Ver. 2.00 or Later System Simulator	External Part User Open Interface Specifications	U14873E
RX850 Ver. 3.13 or Later Real-time OS	Basic	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Ver. 3.13 or Later Real-time OS	Basic	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 Ver. 3.01 Task Debugger		U13737E
RD850 Pro Ver. 3.01 Task Debugger		U13916E
AZ850 Ver. 3.0 System Performance Analyzer		U14410E
PG-FP3 Flash Memory Programmer		U13502E
PG-FP4 Flash Memory Programmer		U15260E

★ 121-pin plastic FBGA (12 × 12)

- μPD703014AF1-×××-EA6
- μPD703014AF1-×××-EA6-A
- μPD703014AYF1-×××-EA6
- μPD703014AYF1-×××-EA6-A
- μPD703014BF1-×××-EA6-A
- μPD703014BYF1-×××-EA6-A
- μPD703015AF1-×××-EA6
- μPD703015AF1-×××-EA6-A
- μPD703015AYF1-×××-EA6
- μPD703015AYF1-×××-EA6-A
- μPD703015BF1-×××-EA6-A
- μPD703015BYF1-×××-EA6-A
- μPD703017AF1-×××-EA6
- μPD703017AF1-×××-EA6-A
- μPD703017AYF1-×××-EA6
- μPD703017AYF1-×××-EA6-A
- μPD70F3015BF1-EA6-A
- μPD70F3015BYF1-EA6-A
- μPD70F3017AF1-EA6
- μPD70F3017AF1-EA6-A
- μPD70F3017AYF1-EA6
- μPD70F3017AYF1-EA6-A



Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name	Pin No.	Name
A1	P20	B8	P83	D2	V_{DD}	G11	P60	K13	BV_{DD}	M7	V_{SS}
A2	P15	B9	P80	D3	V_{SS}	G12	P56	L1	P104	M8	V_{SS}
A3	V_{SS}	B10	P75	D11	AV_{DD}	G13	P57	L2	P105	M9	P92
A4	P13	B11	AV_{SS}	D12	AV_{DD}	H1	P34	L3	$\overline{RESET}$	M10	P95
A5	P11	B12	AV_{SS}	D13	AV_{DD}	H2	P37	L4	V_{DD}	M11	P41
A6	P06	B13	P71	E1	P25	H3	P35	L5	V_{SS}	M12	P45
A7	P03	C1	P22	E2	V_{DD}	H11	P55	L6	X2	M13	P44
A8	P00	C2	P23	E3	P30	H12	P53	L7	P90	N1	P107
A9	P81	C3	V_{SS}	E11	AV_{DD}	H13	P54	L8	P120	N2	P110
A10	P76	C4	P24	E12	P64	J1	IC/V_{PP}[Note]	L9	P93	N3	P112
A11	P73	C5	P07	E13	P65	J2	IC/V_{PP}[Note]	L10	P96	N4	V_{DD}
A12	P72	C6	P04	F1	P26	J3	P100	L11	BV_{SS}	N5	XT1
A13	AV_{SS}	C7	P01	F2	P27	J11	P52	L12	BV_{SS}	N6	V_{SS}
B1	P21	C8	P82	F3	P33	J12	P50	L13	BV_{SS}	N7	V_{SS}
B2	P14	C9	P77	F11	P63	J13	P51	M1	P106	N8	CLKOUT
B3	V_{SS}	C10	P74	F12	P61	K1	P101	M2	P111	N9	P91
B4	P12	C11	AV_{SS}	F13	P62	K2	P102	M3	P113	N10	P94
B5	P10	C12	P70	G1	P31	K3	P103	M4	V_{DD}	N11	P40
B6	P05	C13	AV_{REF}	G2	P32	K11	P46	M5	XT2	N12	P42
B7	P02	D1	V_{DD}	G3	P36	K12	P47	M6	X1	N13	P43

Note μPD703014A, 703014AY, 703014B, 703014BY, 703015A, 703015AY, 703015B, 703015BY, μPD703017A, 703017AY: IC (connect directly to V_{SS})
μPD70F3015B, 70F315BY, 70F3017A, 70F3017AY: V_{PP} (connect to V_{SS} in normal operation mode)

Remarks **1.** Alternate pin names are omitted. Alternate pins are identical to the 100-pin plastic LQFP. However, SCL and SDA are available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 70F3015BY, 703017AY, and 70F3017AY.
2. Connect the D4 pin directly to V_{SS}.

3.2.1 Program register set

The program register set includes general-purpose registers and a program counter.

(1) General-purpose registers

Thirty-two general-purpose registers, r0 to r31, are available. Any of these registers can be used as a data variable or address variable.

However, r0 and r30 are implicitly used by instructions, so care must be exercised when using these registers. Also, r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost. The contents must be restored to the registers after the registers have been used. r2 is sometimes used by the real-time OS. When the real-time OS to be used is not using r2, r2 can be used as a variable register.

Table 3-1. Program Registers

Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating 32-bit immediate
r2	Address/data variable register (when the real-time OS to be used is not using r2)	
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	Used to access global variable in data area
r5	Text pointer	Register to indicate the start of the text area[Note]
r6 to r29	Address/data variable registers	
r30	Element pointer	Base pointer when memory is accessed
r31	Link pointer	Used by compiler when calling function
PC	Program counter	Holds instruction address during program execution

Note Area in which program code is mapped.

(2) Program counter (PC)

This register holds the address of the instruction under execution. The lower 24 bits of this register are valid, and bits 31 to 24 are fixed to 0. If a carry occurs from bit 23 to bit 24, it is ignored.

Bit 0 is fixed to 0, and branching to an odd address cannot be performed.

After reset: 00000000H

Symbol	31 … 24	23 … 1	0
PC	Fixed to 0	Instruction address under execution	0

(2/2)

SAT[Note]	Saturation detection of operation result of saturation operation instruction
0	Not saturated. This flag is not cleared (0) if the result of saturated operation instruction execution is not saturated while this flag is set (1). To clear (0) this flag, write the PSW directly.
1	Saturated.

CY	Detection of carry or borrow of operation result
0	Overflow has not occurred.
1	Overflow occurred.

OV[Note]	Detection of overflow during operation
0	Overflow has not occurred.
1	Overflow occurred.

S[Note]	Detection of operation result positive/negative
0	The operation result was positive or 0.
1	The operation result was negative.

Z	Detection of operation result zero
0	The operation result was not 0.
1	The operation result was 0.

Note The result of a saturation-processed operation is determined by the contents of the OV and S bits in the saturation operation. Simply setting (1) the OV bit will set (1) the SAT bit in a saturation operation.

Status of operation result	Flag status			Saturation-processed operation result
	SAT	OV	S	
Maximum positive value exceeded	1	1	0	7FFFFFFFH
Maximum negative value exceeded	1	1	1	80000000H
Positive (not exceeding the maximum)	Retains the value before operation	0	0	Operation result itself
Negative (not exceeding the maximum)			1	

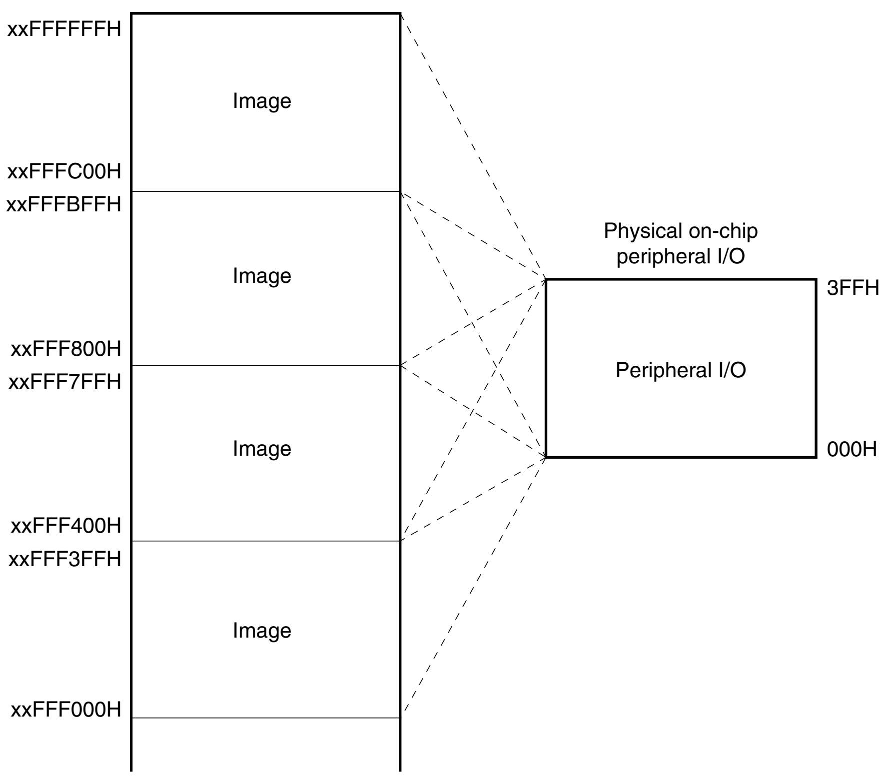
(3) On-chip peripheral I/O area

A 4 KB area of addresses FFF000H to FFFFFFH is reserved as an on-chip peripheral I/O area.

The V850/SA1 is provided with a 1 KB area of addresses FFF000H to FFF3FFH as a physical on-chip peripheral I/O area, and its image can be seen on the rest of the area (FFF400H to FFFFFFH).

Peripheral I/O registers associated with operation mode specification and state monitoring for the on-chip peripherals are all memory-mapped to the on-chip peripheral I/O area. Program fetches are not allowed in this area.

Figure 3-12. On-Chip Peripheral I/O Area



Cautions 1. The least significant bit of an address is not decoded since all registers reside at an even address. If an odd address (2n + 1) in the peripheral I/O area is referenced (accessed in byte units), the register at the next lowest even address (2n) will be accessed.

2. If a register that can be accessed in byte units is accessed in halfword units, the higher 8 bits become undefined, if the access is a read operation. If a write access is made, only the data in the lower 8 bits is written to the register.

3. If a register at address n that can be accessed only in halfword units is accessed in word units, the operation is replaced with two halfword operations. The first operation (lower 16 bits) accesses the register at address n and the second operation (higher 16 bits) accesses the register at address n + 2.

4. If a register at address n that can be accessed in word units is accessed with a word operation, the operation is replaced with two halfword operations. The first operation (lower 16 bits) accesses the register at address n and the second operation (higher 16 bits) accesses the register at address n + 2.

5. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.

CHAPTER 4 BUS CONTROL FUNCTION

The V850/SA1 is provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

4.1 Features

- Address bus (capable of separate output)
- 16-bit data bus
- Able to be connected to external devices via the pins that have alternate functions as ports
- Wait function
  - Programmable wait function, capable of inserting up to 3 wait states per 2 blocks
  - External wait control through $\overline{\text{WAIT}}$ input pin
- Idle state insertion function
- Bus hold function

4.2 Bus Control Pins and Control Register

4.2.1 Bus control pins

The following pins are used for interfacing with external devices.

Table 4-1. Bus Control Pins

External Bus Interface Function	Corresponding Port (Pins)
Address/data bus (AD0 to AD7)	Port 4 (P40 to P47)
Address/data bus (AD8 to AD15)	Port 5 (P50 to P57)
Address bus (A1 to A4)	Port 11 (P110 to P113)
Address bus (A5 to A12)	Port 10 (P100 to P107)
Address bus (A13 to A15)	Port 3 (P34 to P36)
Address bus (A16 to A21)	Port 6 (P60 to P65)
Read/write control ($\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$, R/$\overline{\text{W}}$, $\overline{\text{DSTB}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{RD}}$)	Port 9 (P90 to P93)
Address strobe (ASTB)	Port 9 (P94)
Bus hold control ($\overline{\text{HLDRQ}}$, $\overline{\text{HLDAK}}$)	Port 9 (P95, P96)
External wait control ($\overline{\text{WAIT}}$)	Port 12 (P120)

The bus interface function of each pin is enabled by specifying the memory expansion mode register (MM) or the memory address output mode register (MAM). For the details of specifying an operation mode of the external bus interface, refer to **3.4.6 (1) Memory expansion mode register (MM)** and for **(2) Memory address output mode register (MAM).**

Caution For debugging using the separate bus, refer to IE-703017-MC-EM1 User's Manual.

5.1 Outline

The V850/SA1 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and realizes a high-powered interrupt function that can service interrupt requests from a total of 30 sources.

An interrupt is an event that occurs independently of program execution, and an exception is an event that is dependent on program execution.

The V850/SA1 can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (fetching of an illegal opcode) (exception trap).

5.1.1 Features

- Interrupts
  - External interrupts: 8 sources (5 sources[Note])
  - Internal interrupts: 24 sources
  - 8 levels of programmable priorities
  - Mask specification for interrupt requests according to priority
  - Masks can be specified for each maskable interrupt request.
  - Noise elimination, edge detection, and valid edge of external interrupt request signal can be specified.

Note Number of external interrupts that can release the software STOP mode.

- Exceptions
  - Software exceptions: 32 sources
  - Exception trap: 1 source (illegal opcode exception)

The interrupt/exception sources are listed in Table 5-1.

5.4 Software Exceptions

A software exception is generated when the CPU executes the TRAP instruction, and can be always acknowledged.

- TRAP instruction format: TRAP vector (where vector is 0 to 1FH)

For details of the instruction function, refer to the **V850 Series Architecture User's Manual.**

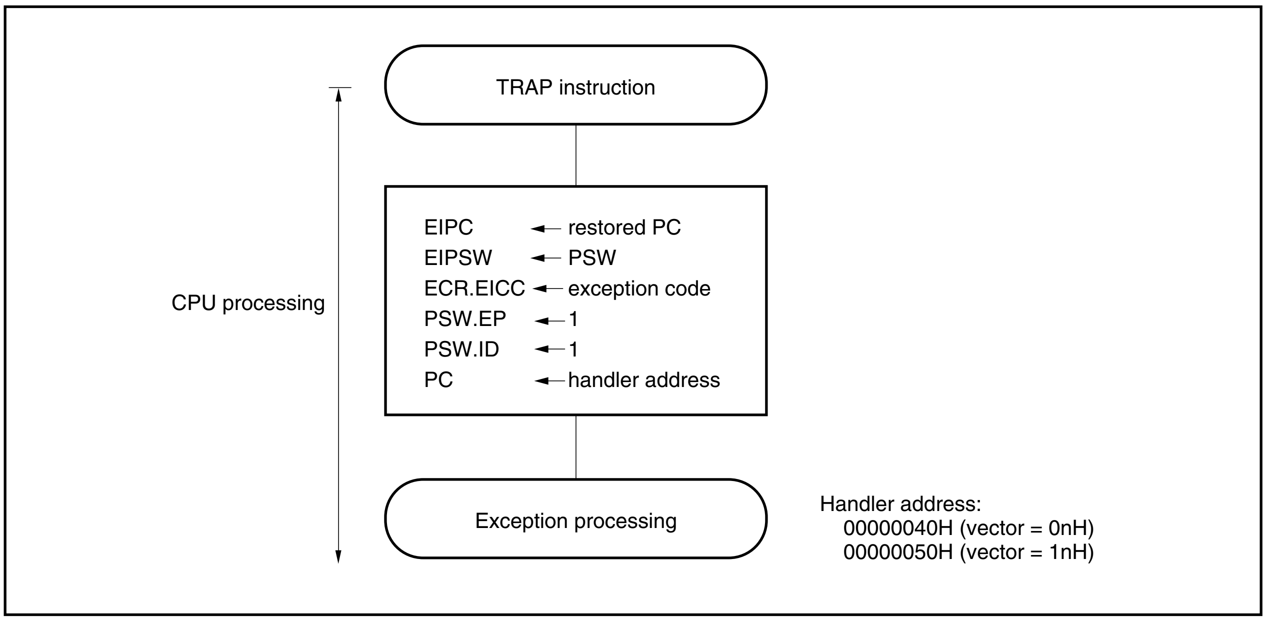
5.4.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine.

(1) Saves the restored PC to EIPC.
(2) Saves the current PSW to EIPSW.
(3) Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
(4) Sets the EP and ID bits of PSW.
(5) Loads the handler address (00000040H or 00000050H) of the software exception routine in the PC, and transfers control.

How a software exception is processed is shown below.

Figure 5-10. Software Exception Processing



★ 5.8.1 Interrupt request valid timing after EI instruction

When an interrupt request signal is generated (IF flag = 1) in the status in which the DI instruction is executed (interrupts disabled) and interrupts are not masked (MK flag = 0), seven system clocks are required from the execution of the EI instruction (interrupts enabled) to the interrupt request acknowledgement by the CPU. The CPU does not acknowledge interrupt requests if the DI instruction (interrupts disabled) is executed during the seven system clocks.

Therefore, seven system clocks worth of instruction execution clocks must be inserted after the EI instruction (interrupts enabled). However, under the following conditions, interrupt requests cannot be acknowledged even if the seven system clocks are secured, so securing under the following conditions is prohibited.

- In IDLE/software STOP mode
- An interrupt request non-sampling instruction (instruction to manipulate the PSW.ID bit) is executed
- An interrupt request control register (xxICn) is accessed

The following shows an example of program processing.

[Program processing example]

```
        DI
        :               ; (MK flag = 0)
        :               ; ← Interrupt request occurs (IF flag = 1)
        EI              ; EI instruction executed
        NOP             ; 1 system clock                           ┐
        NOP             ; 1 system clock                           │
        NOP             ; 1 system clock                           ├ Note
        NOP             ; 1 system clock                           │
        JR    LP1       ; 3 system clocks (branch to LP1 routine)  ┘
        :
   LP1  :               ; LPI routine
        DI              ; After EI instruction execution, NOP instruction is
                          executed four times, and DI
                          instruction is executed at the eighth clock by JR instruction
```

Note Do not execute the DI instruction (PSW.ID = 1) during this period.

Remarks 1. In this example, the DI instruction is executed at the eighth clock after execution of the EI instruction, so the CPU acknowledges an interrupt request signal and performs interrupt servicing.

2. The interrupt servicing routine instructions are not executed at the eighth clock after the EI instruction execution. The interrupt servicing routine instructions are executed the four system clocks after the CPU acknowledges the interrupt request signal.

3. This example shows the case in which an interrupt request signal is generated (IF flag = 1) before the EI instruction is executed. If an interrupt request signal is generated after the EI instruction is executed, the CPU does not acknowledge the interrupt request signal if interrupts are disabled (PSW.ID = 1) for seven clocks after the IF flag is set (1).

Table 6-1. Operating Statuses in HALT Mode (1/2)

Item (HALT Mode Setting)		When CPU Operates with Main Clock		When CPU Operates with Subclock	
		When subclock does not exist	When subclock exists	When main clock's oscillation continues	When main clock's oscillation is stopped
CPU		Stopped			
Clock generator		Oscillation for main clock and subclock Clock supply to CPU is stopped			
16-bit timer (TM0)		Operating			Operates when INTWTI is selected as count clock (f_{XT} is selected for watch timer)
16-bit timer (TM1)		Operating			Stopped
8-bit timer (TM2)		Operating			Stopped
8-bit timer (TM3)		Operating			Stopped
8-bit timer (TM4)		Operating			Operates when f_{XT} is selected for count clock
8-bit timer (TM5)		Operating			Operates when f_{XT} is selected for count clock
Watch timer		Operates when $f_{XX}/2^9$ is selected for count clock	Operating		Operates when f_{XT} is selected for count clock
Watchdog timer		Operating (interval timer only)			
Serial interface	CSI0 to CSI2	Operating			Operates when an external clock is selected as the serial clock
	I^{2}C[Note]	Operating			Stopped
	UART0, UART1	Operating			Operates when an external clock is selected as the baud rate clock (transmit only)
A/D converter		Operating			Stopped
DMA0 to DMA2		Operating			
Real-time output		Operating			
Port function		Held			
External bus interface		Only bus hold function operates			
External interrupt request	NMI	Operating			
	INTP0 to INTP3	Operating			
	INTP4 to INTP6	Operating			Stopped

★ **Note** Available only in the μPD703014AY, 703014BY, 703015AY, 703015BY, 703017AY, 70F3015BY, and 70F3017AY

CHAPTER 7 TIMER/COUNTER FUNCTION

7.1 16-Bit Timers (TM0, TM1)

7.1.1 Outline

- 16-bit capture/compare registers: 2 (CRn0, CRn1)
- Independent capture/trigger inputs: 2 (TIn0, TIn1)
- Support of output of capture/match interrupt request signals (INTTMn0, INTTMn1)
- Event input (shared with TIn0) via digital noise eliminator and support of edge specification
- Timer output operated by match detection: 1 each (TOn)
  When using the P34/TO0 and P35/TO1 pins as the TO0 and TO1 pins (timer output), set the value of port 3 (P3) to 0 (low-level output) and the port 3 mode register (PM3) to 0 (port output mode). The logical sum (ORed) value of the output of the port and the timer is output.

Remark n = 0, 1

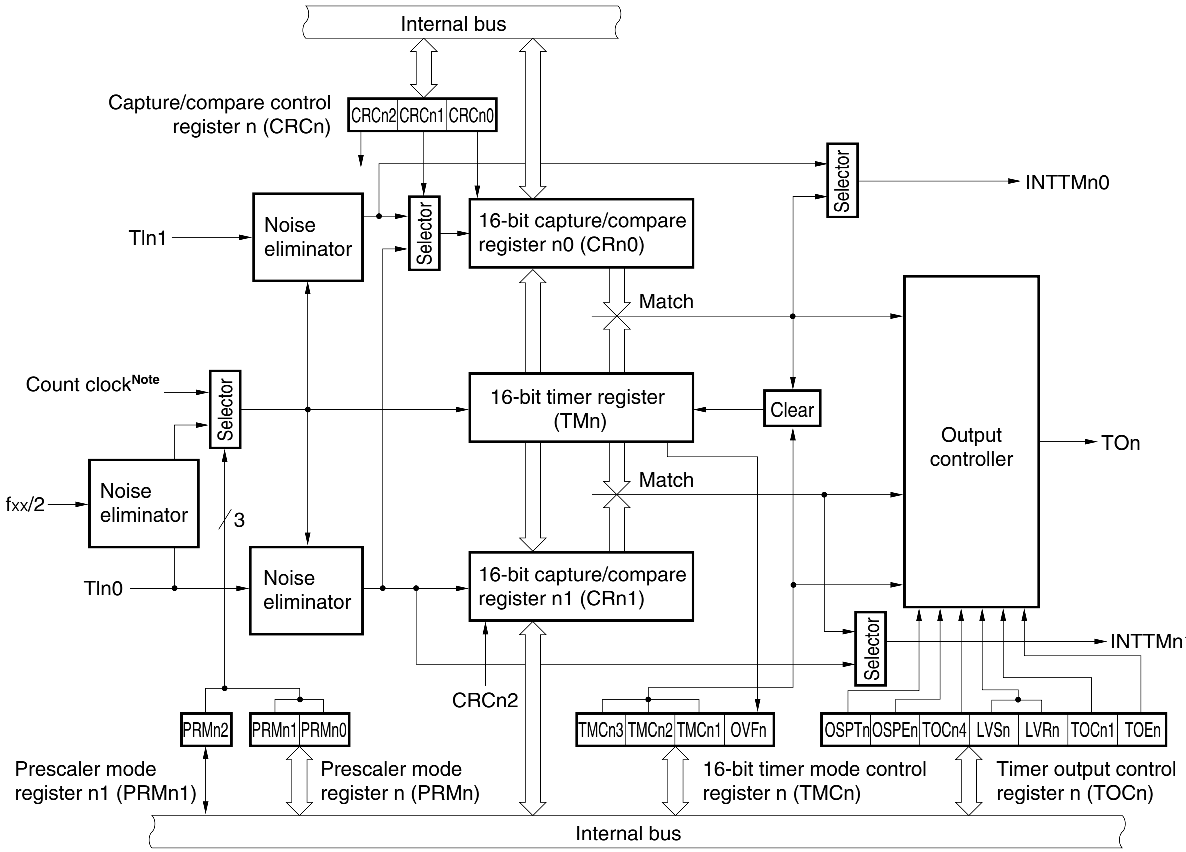
7.1.2 Functions

TM0 and TM1 have the following functions.

- Interval timer
- PPG output
- Pulse width measurement
- External event counter
- Square wave output
- One-shot pulse output

The block diagram is shown below.

Figure 7-1. Block Diagram of TM0 and TM1



Note The count clock is set by the PRMn and PRMn1 registers.

Remark n = 0, 1

(1) Interval timer

Generates an interrupt at predetermined time intervals.

(2) PPG output

Can output a square wave whose frequency and output pulse width can be changed arbitrarily.

(3) Pulse width measurement

Can measure the pulse width of a signal input from an external source.

(4) External event counter

Can measure the number of pulses of a signal input from an external source.

(5) Square wave output

Can output a square wave of any frequency.

(2) Capture/compare control registers 0, 1 (CRC0, CRC1)

CRCn controls the operation of capture/compare register n (CRn0 and CRn1).

CRCn is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CRC0 and CRC1 to 00H.

After reset: 00H R/W Address: FFFFF20AH, FFFFF21AH

	7	6	5	4	3	2	1	0
CRCn	0	0	0	0	0	CRCn2	CRCn1	CRCn0

(n = 0, 1)

CRCn2	Selection of Operation Mode of CRn1
0	Operates as compare register
1	Operates as capture register

CRCn1	Selection of Capture Trigger of CRn0
0	Captured at valid edge of TIn1
1	Captured in reverse phase of valid edge of TIn0

CRCn0	Selection of Operation Mode of CRn0
0	Operates as compare register
1	Operates as capture register

Cautions 1. Before setting CRCn, be sure to stop the timer operation.

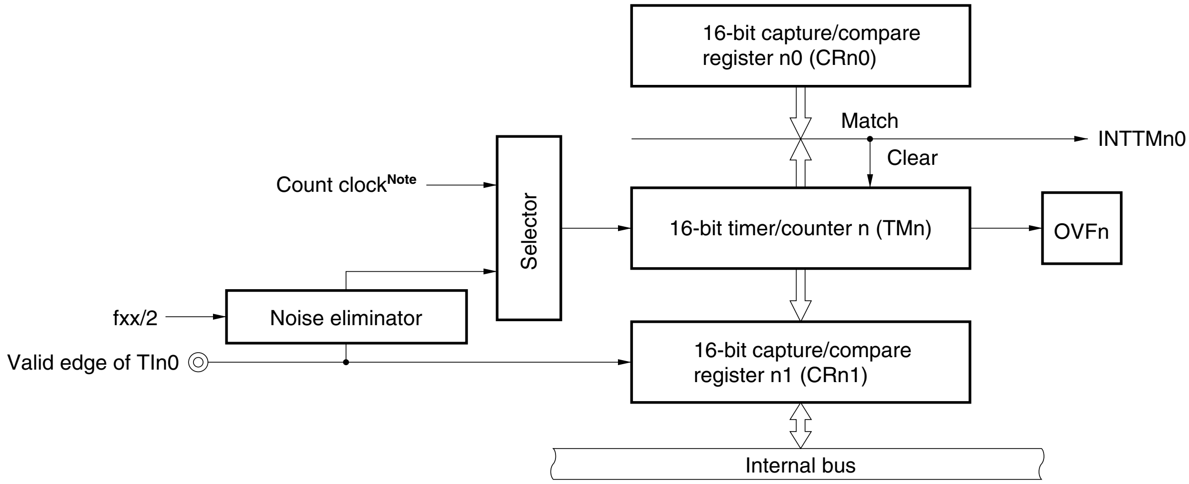
2. When the mode in which the timer is cleared and started on a match between TMn and CRn0 is selected by 16-bit timer mode control register n (TMCn), do not specify CRn0 as a capture register.

3. When both the rising edge and falling edge are specified for the TIn0 valid edge, the capture operation does not work.

4. To ensure that the capture trigger captures the signals from TIn0 and TIn1 correctly, a pulse longer than two of the count clocks selected by prescaler mode registers n and n1 (PRMn, PRMn1) is required.

5. Be sure to set bits 3 to 7 to 0.

Figure 7-17. Configuration of External Event Counter



Note The count clock is set by the PRMn and PRMn1 registers.

Remarks 1. "◎——" indicates a signal that can be directly connected to a port.

2. n = 0, 1

Figure 7-18. Timing of External Event Counter Operation (with Rising Edge Specified)

Caution Read TMn when reading the count value of the external event counter.

Remark n = 0, 1

(2/3)

EXC	Detection of Extension Code Reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC = 0)	Condition for setting (EXC = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 • When IICE changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the higher four bits of the received address data are either “0000” or “1111” (set at the rising edge of the eighth clock).

COI	Detection of Matching Addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI = 0)	Condition for setting (COI = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 • When IICE changes from 1 to 0 • When $\overline{\text{RESET}}$ is input	• When the received address matches the local address (SVA0) (set at the rising edge of the eighth clock).

TRC	Detection of Transmit/Receive Status
0	Receive status (other than transmit status). The SDA line is set to high impedance.
1	Transmit status. The value in the SO latch is enabled for output to the SDA line (valid starting at the rising edge of the first byte’s ninth clock).

Condition for clearing (TRC = 0)	Condition for setting (TRC = 1)
• When a stop condition is detected • Cleared by LREL = 1 • When IICE changes from 1 to 0 • Cleared by WREL = 1[Note] • When $\overline{\text{ALD}}$ changes from 0 to 1 • When $\overline{\text{RESET}}$ is input Master • When “1” is output to the first byte’s LSB (transfer direction specification bit) Slave • When a start condition is detected When not used for communication	Master • When a start condition is generated Slave • When “1” is input by the first byte’s LSB (transfer direction specification bit)

Note When bit 3 (TRC) of IIC status register 0 (IICS0) is 1, if a wait is released by setting bit 5 (WREL) of IIC control register 0 (IICC0) at the 9th clock, the SDA line becomes high impedance after TRC is cleared.

Remark LREL: Bit 6 of IIC control register 0 (IICC0)
IICE: Bit 7 of IIC control register 0 (IICC0)

Table 10-5. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{ACK}$ signal transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is output (when SPIE = 1)[Note 2]
When data is at low level while attempting to output a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to output a restart condition	When stop condition is output (when SPIE = 1)[Note 2]
When data is at low level while attempting to output a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCL is at low level while attempting to output a restart condition	

Notes 1. When WTIM (bit 3 of IIC control register 0 (IICC0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a possibility that arbitration will occur, set SPIE = 1 for master device operation.

Remark SPIE: Bit 5 of IIC control register 0 (IICC0)

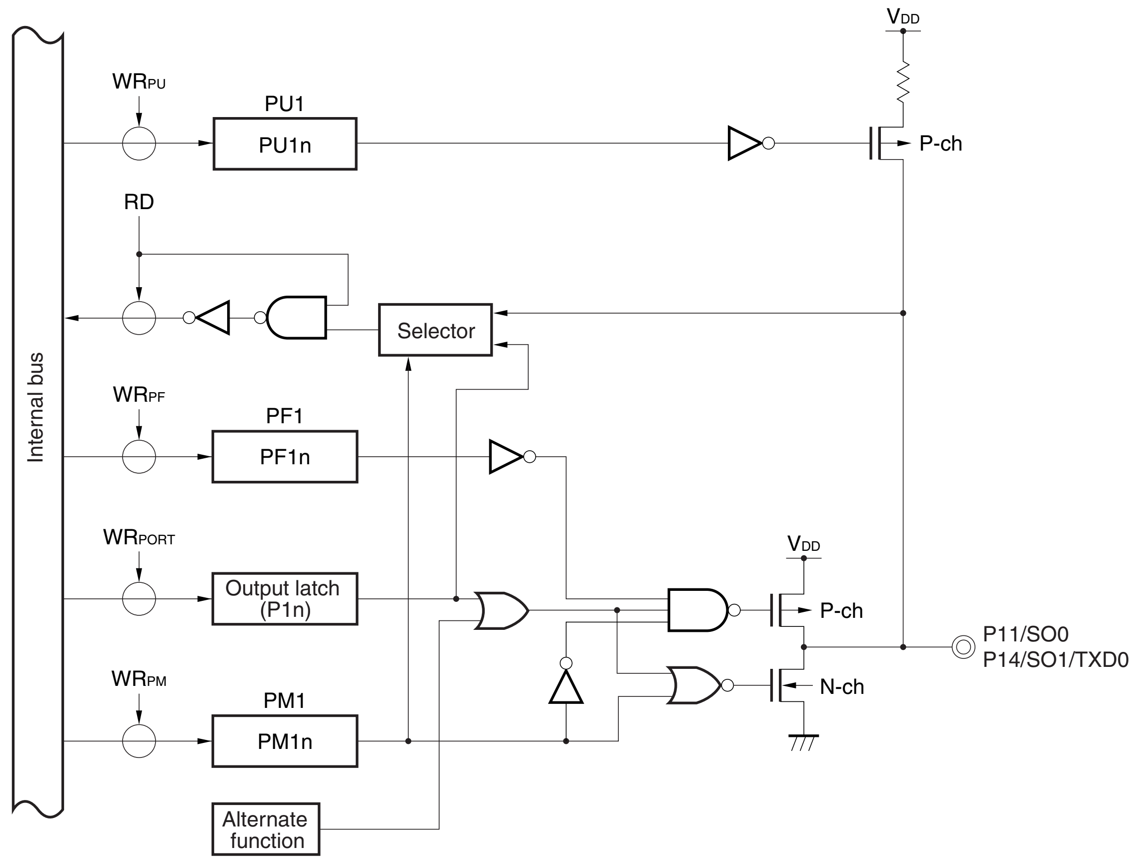
10.3.11 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request (INTIIC0) when a local address and extension code have been received. This function makes processing more efficient by preventing unnecessary interrupt requests from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has output a start condition) to a slave device.

However, when a stop condition is detected, bit 5 (SPIE) of IIC control register 0 (IICC0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

Figure 14-3. Block Diagram of P11 and P14



Remarks **1.** PU1: Pull-up resistor option register 1

PF1: Port 1 function register

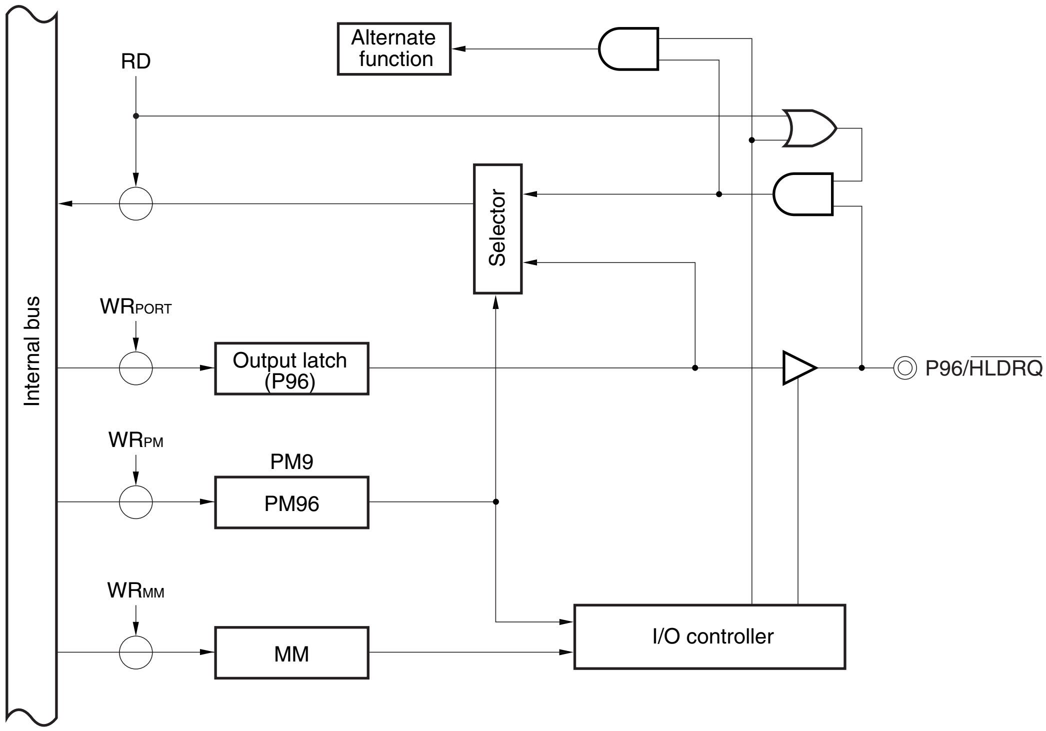
PM1: Port 1 mode register

RD: Read signal of port 1

WR: Write signal of port 1

2. n = 1, 4

Figure 14-17. Block Diagram of P96



Remark PM9: Port 9 mode register
MM: Memory expansion mode register
RD: Read signal of port 9
WR: Write signal of port 9

★ 14.4 Operation of Port Function

The operation of a port differs depending on whether the port is in the input or output mode, as described below.

14.4.1 Writing data to I/O port

(1) In output mode

A value can be written to the output latch by using a transfer instruction. The contents of the output latch are output from the pin. Once data has been written to the output latch, it is retained until new data is written to the output latch.

(2) In input mode

A value can be written to the output latch by using a transfer instruction. Because the output buffer is off, however, the status of the pin does not change.

Once data has been written to the output latch, it is retained until new data is written to the output latch.

Caution A bit manipulation instruction (CLR1, SET1, NOT1) manipulates 1 bit but accesses a port in 8-bit units. If this instruction is executed to manipulate a port with a mixture of input and output bits, the contents of the output latch of a pin set in the input mode, in addition to the bit to be manipulated, are overwritten to the current input pin status and become undefined.

14.4.2 Reading data from I/O port

(1) In output mode

The contents of the output latch can be read by using a transfer instruction. The contents of the output latch do not change.

(2) In input mode

The status of the pin can be read by using a transfer instruction. The contents of the output latch do not change.

(5/7)

Edition	Major Revision from Previous Edition	Applied to:
3rd edition	**Figure 12-3 Correspondence Between DRAn Setup Value and Internal RAM Area** Addition	**CHAPTER 12 DMA FUNCTIONS**
	Figure 12-5 DMA Channel Control Registers 0 to 2 (DCHC0 to DCHC2) Deletion and addition of products in Note 2	
	14.2.1 (4) Block diagram (port 0) Addition	**CHAPTER 14 PORT FUNCTION**
	14.2.2 (3) Block diagrams (port 1) Addition	
	14.2.3 (3) Block diagrams (port 2) Addition	
	14.2.4 (3) Block diagrams (port 3) Addition	
	14.2.5 (1) Functions of P4 and P5 pins Modification of description	
	14.2.5 (3) Block diagram (port 4, port 5) Addition	
	14.2.6 (3) Block diagram (port 6) Addition	
	14.2.7 (2) Block diagram (port 7, port 8) Addition	
	14.2.8 (1) Function of P9 pins Modification of description	
	14.2.8 (3) Block diagrams (port 9) Addition	
	14.2.9 (3) Block diagram (port 10) Addition	
	14.2.10 (3) Block diagrams (port 11) Addition	
	14.2.11 (1) Function of P12 pin Deletion of description	
	14.2.11 (3) Block diagram (port 12) Addition	
	14.3 Setting When Port Pin Is Used for Alternate Function Addition	
	16.1.1 Erase units Addition	**CHAPTER 16 FLASH MEMORY (μPD70F3017A, 70F3017AY)**
	16.4 (3) CSI0 + HS Addition	
	Table 16-1 Signal Generation of Dedicated Flash Programmer (PG-FP3) Addition of CSI0 + HS	
	Table 16-2 Pins Used by Each Serial Interface Addition of CSI0 + HS	
	Table 16-3 List of Communication Systems Addition of CSI0 + HS	
4th edition	Addition of μPD703014B, 703014BY, 703015B, 703015BY, 70F3015B, and 70F3015BY Deletion of μPD703014AGC, 703014AYGC, 703015AGC, and 703015AYGC	Throughout
	Addition of **Table 1-1 List of V850/SA1 Products**	**INTRODUCTION**
	Addition of description to the minimum instruction execution time in **1.2 Features**	
	Deletion and addition of products in **1.4 Ordering Information**	
	Deletion and addition of products in **1.5 Pin Configuration**	
	Deletion of description in **1.6.2 (2) Bus control unit (BCU)**	
	Addition of **Table 2-1 Pin I/O Buffer Power Supplies**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of description in **Table 2-2 Operating States of Pins in Each Operating Mode**	
	Modification of description in **2.3 (7) P60 to P65 (Port 6)**	
	Addition of **2.3 (13) CLKOUT (Clock Out)**	
	Addition and modification of description in **2.4 Pin I/O Circuits and Recommended Connection of Unused Pins**	
	Modification of **2.5 Pin I/O Circuits**	