Renesas Electronics America Inc - UPD78F0730MC-CAB-AX Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	78K/0
Core Size	8-Bit
Speed	16MHz
Connectivity	3-Wire SIO, UART/USART, USB
Peripherals	LVD, POR, PWM, WDT
Number of I/O	19
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3K x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	30-LSSOP (0.240", 6.10mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f0730mc-cab-ax

3.4.3 Direct addressing

[Function]

The memory to be manipulated is directly addressed with immediate data in an instruction word becoming an operand address.

This addressing can be carried out for all of the memory spaces.

[Operand format]

Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

MOV A, !0FE00H; when setting !addr16 to FE00H

Operation code	1 0 0 0 1 1 1 0	OP code
	0 0 0 0 0 0 0 0	00H
	1 1 1 1 1 1 1 0	FEH

[Illustration]

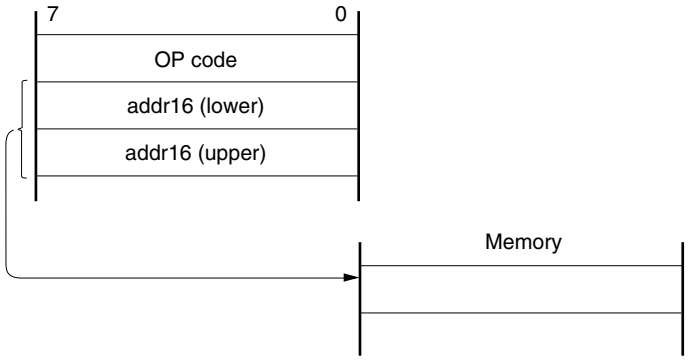
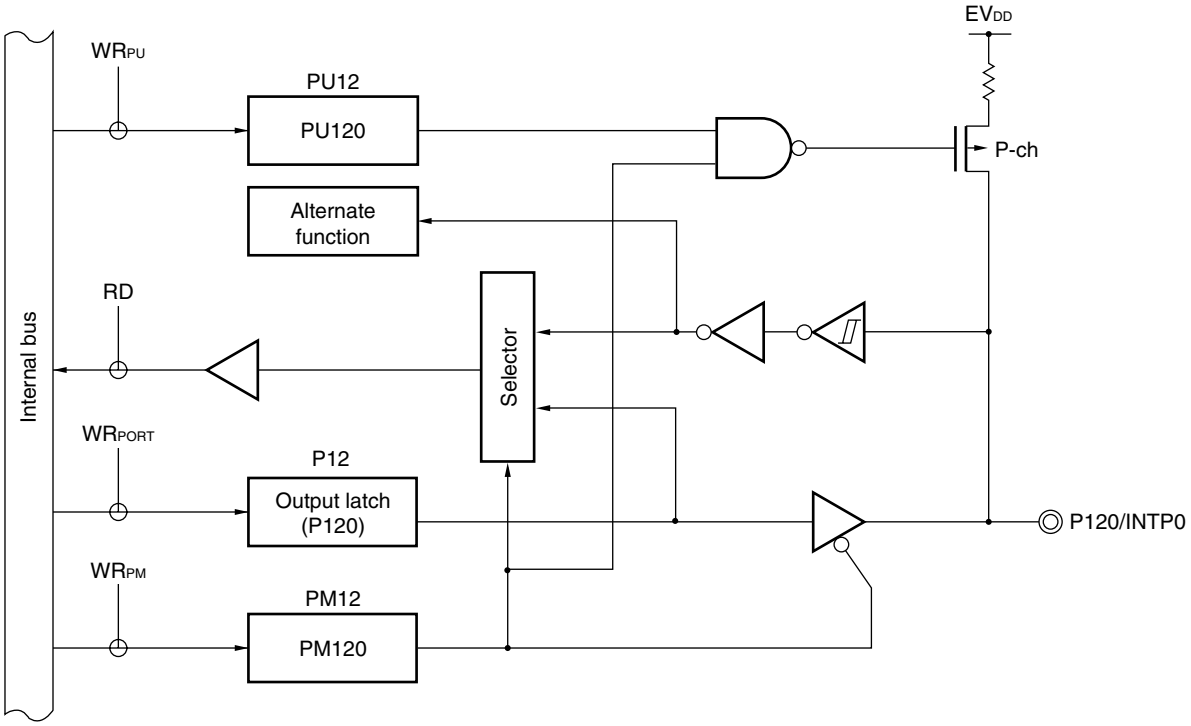


Figure 4-13. Block Diagram of P120



P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

(9) USB clock control register (UCKC)

This register controls the USB clock (f_{USB}) supplied to the USB function controller.

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

Reset signal generation sets this register to 00H.

Figure 5-10. Format of USB Clock Control Register (UCKC)

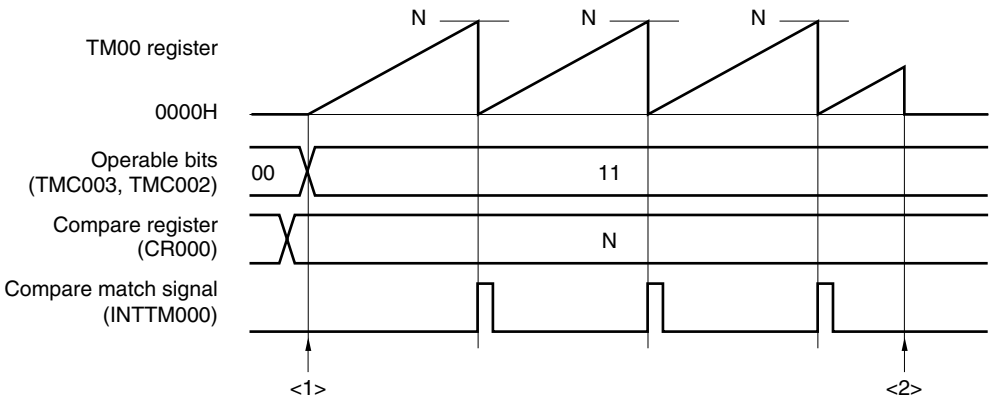
Address: FFA7H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
UCKC	UCKCNT	0	0	0	0	0	0	0

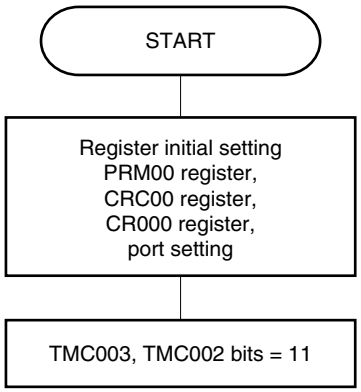
UCKCNT	Control of USB clock supplied to USB function controller
0	Stops USB clock supply.
1	Supplies USB clock.

Caution Before shifting to the STOP mode, stop the clock supply to the USB function controller. After the STOP mode is released, count the PLL oscillation stabilization time (800 µs) using software, and supply the clock to the USB function controller after the oscillation stabilization wait time has elapsed.

Figure 6-21. Example of Software Processing in External Event Counter Mode



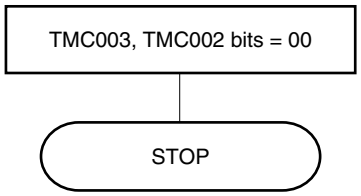
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMC003 and TMC002 bits to 11.

Starts count operation

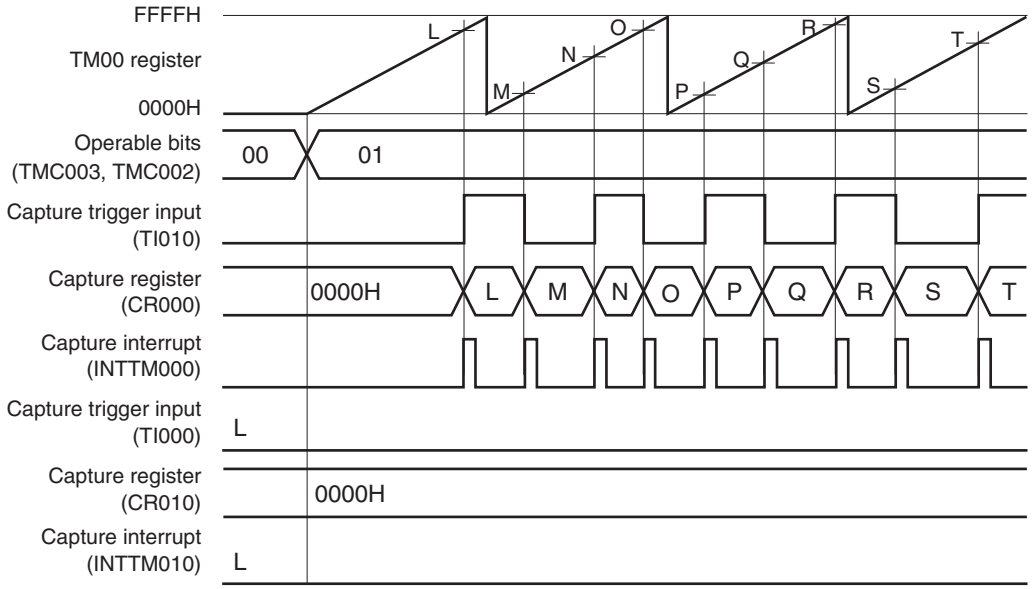
<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TMC003 and TMC002 bits to 00.

Figure 6-37. Timing Example of Free-Running Timer Mode (CR000: Capture Register, CR010: Capture Register) (2/2)

(b) TOC00 = 13H, PRM00 = C0H, CRC00 = 05H, TMC00 = 04H



This is an application example where both the edges of the TI010 pin are detected and the count value is captured to CR000 in the free-running timer mode.

When both CR000 and CR010 are used as capture registers and when the valid edge of only the TI010 pin is to be detected, the count value cannot be captured to CR010.

(1) 8-bit timer counter 5n (TM5n)

TM5n is an 8-bit register that counts the count pulses and is read-only.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 7-3. Format of 8-Bit Timer Counter 5n (TM5n)

Address: FF16H (TM50), FF1FH (TM51) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM5n (n = 0, 1)								

In the following situations, the count value is cleared to 00H.

<1> Reset signal generation

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in the mode in which clear & start occurs upon a match of the TM5n and CR5n.

(2) 8-bit timer compare register 5n (CR5n)

CR5n can be read and written by an 8-bit memory manipulation instruction.

Except in PWM mode, the value set in CR5n is constantly compared with the 8-bit timer counter 5n (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match.

In the PWM mode, the TO5n pin becomes inactive when the values of TM5n and CR5n match, but no interrupt is generated.

The value of CR5n can be set within 00H to FFH.

Reset signal generation sets CR5n to 00H.

Figure 7-4. Format of 8-Bit Timer Compare Register 5n (CR5n)

Address: FF17H (CR50), FF41H (CR51) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR5n (n = 0, 1)								

Cautions 1. In the mode in which clear & start occurs on a match of TM5n and CR5n (TMC5n6 = 0), do not write other values to CR5n during operation.

2. In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

Figure 8-4. Format of 8-Bit Timer H Mode Register 1 (TMHMD1)

Address: FF6CH After reset: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS12	CKS11	CKS10	Count clock selection		
				f_{PRS} = 12 MHz	f_{PRS} = 16 MHz
0	0	0	f_{PRS}	12 MHz	16 MHz
0	0	1	$f_{PRS}/2^2$	3 MHz	4 MHz
0	1	0	$f_{PRS}/2^4$	750 kHz	1 MHz
0	1	1	$f_{PRS}/2^6$	187.5 kHz	250 kHz
1	0	0	$f_{PRS}/2^{12}$	2.93 kHz	3.91 kHz
1	0	1	$f_{RL}/2^7$	1.88 kHz (TYP.)	
1	1	0	$f_{RL}/2^9$	0.47 kHz (TYP.)	
1	1	1	f_{RL}	240 kHz (TYP.)	

TMMD11	TMMD10	Timer operation mode
0	0	Interval timer mode
0	1	Carrier generator mode
1	0	PWM output mode
1	1	Setting prohibited

TOLEV1	Timer output level control (in default mode)
0	Low level
1	High level

TOEN1	Timer output control
0	Disables output
1	Enables output

Cautions 1. When TMHE1 = 1, setting the other bits of TMHMD1 is prohibited.

2. In the PWM output mode and carrier generator mode, be sure to set 8-bit timer H compare register 11 (CMP11) when starting the timer count operation (TMHE1 = 1) after the timer count operation was stopped (TMHE1 = 0) (be sure to set again even if setting the same value to CMP11).

3. When the carrier generator mode is used, set so that the count clock frequency of TMH1 becomes more than 6 times the count clock frequency of TM51.

Remarks 1. f_{PRS}: Peripheral hardware clock frequency

2. f_{RL}: Internal low-speed oscillation clock frequency

Setting

<1> Set each register.

Figure 8-12. Register Setting in Carrier Generator Mode

(i) Setting 8-bit timer H mode register 1 (TMHMD1)

(ii) CMP01 register setting

- Compare value

(iii) CMP11 register setting

- Compare value

(iv) TMCYC1 register setting

- RMC1 = 1 ... Remote control output enable bit
- NRZB1 = 0/1 ... carrier output enable bit

(v) TCL51 and TMC51 register setting

- See **7.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51**.

<2> When TMHE1 = 1, 8-bit timer H1 starts counting.

<3> When TCE51 of 8-bit timer mode control register 51 (TMC51) is set to 1, 8-bit timer/event counter 51 starts counting.

<4> After the count operation is enabled, the first compare register to be compared is the CMP01 register. When the count value of 8-bit timer counter H1 and the CMP01 register value match, the INTTMH1 signal is generated, 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with 8-bit timer counter H1 is switched from the CMP01 register to the CMP11 register.

<5> When the count value of 8-bit timer counter H1 and the CMP11 register value match, the INTTMH1 signal is generated, 8-bit timer counter H1 is cleared. At the same time, the compare register to be compared with 8-bit timer counter H1 is switched from the CMP11 register to the CMP01 register.

<6> By performing procedures <4> and <5> repeatedly, a carrier clock is generated.

<7> The INTTM51 signal is synchronized with count clock of the 8-bit timer H1 and output as the INTTM5H1 signal. The INTTM5H1 signal becomes the data transfer signal for the NRZB1 bit, and the NRZB1 bit value is transferred to the NRZ1 bit.

<8> Write the next value to the NRZB1 bit in the interrupt servicing program that has been started by the INTTM5H1 interrupt or after timing has been checked by polling the interrupt request flag. Write data to count the next time to the CR51 register.

<9> When the NRZ1 bit is high level, a carrier clock is output from the TOH1 pin.

11.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

In this mode, communication is executed by using three lines: the serial clock ($\overline{SCK10}$), serial output (SO10), and serial input (SI10) lines.

(1) Registers used

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the CSIC10 register (see **Figure 11-3**).
<2> Set bits 0, 4 and 6 (CSOT10, DIR10, and TRMD10) of the CSIM10 register (see **Figure 11-2**).
<3> Set bit 7 (CSIE10) of the CSIM10 register to 1. → Transmission/reception is enabled.
<4> Write data to transmit buffer register 10 (SOTB10). → Data transmission/reception is started.
Read data from serial I/O shift register 10 (SIO10). → Data reception is started.

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

12.2 Configuration

USB function controller USBF includes the following hardware.

Table 12-1. Configuration of USB Function Controller USBF (1/2)

Item	Configuration
USB port pins	USBP (+), USBM (–)
Control registers	UF0 EP0NAK register (UF0E0N) UF0 EP0NAKALL register (UF0E0NA) UF0 EPNAK register (UF0EN) UF0 EPNAK mask register (UF0ENM) UF0 SNDSIE register (UF0SDS) UF0 CLR request register (UF0CLR) UF0 SET request register (UF0SET) UF0 EP status 0 register (UF0EPS0) UF0 EP status 1 register (UF0EPS1) UF0 EP status 2 register (UF0EPS2) UF0 INT status 0 register (UF0IS0) UF0 INT status 1 register (UF0IS1) UF0 INT status 2 register (UF0IS2) UF0 INT status 3 register (UF0IS3) UF0 INT status 4 register (UF0IS4) UF0 INT mask 0 register (UF0IM0) UF0 INT mask 1 register (UF0IM1) UF0 INT mask 2 register (UF0IM2) UF0 INT mask 3 register (UF0IM3) UF0 INT mask 4 register (UF0IM4) UF0 INT clear 0 register (UF0IC0) UF0 INT clear 1 register (UF0IC1) UF0 INT clear 2 register (UF0IC2) UF0 INT clear 3 register (UF0IC3) UF0 INT clear 4 register (UF0IC4) UF0 FIFO clear 0 register (UF0FIC0) UF0 FIFO clear 1 register (UF0FIC1) UF0 data end register (UF0DEND) UF0 GPR register (UF0GPR) UF0 mode control register (UF0MODC) UF0 mode status register (UF0MODS) UF0 active interface number register (UF0AIFN) UF0 active alternative setting register (UF0AAS) UF0 alternative setting status register (UF0ASS) UF0 endpoint 1 interface mapping register (UF0E1IM) UF0 endpoint 2 interface mapping register (UF0E2IM)

12.4 Register Configuration

12.4.1 Control registers

(1) UF0 EP0NAK register (UF0E0N)

This register controls NAK of Endpoint0 (except an automatically executed request).

This register can be read or written in 8-bit units (however, bit 0 can only be read).

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the UF0FIC0 and UF0FIC1 registers from a read signal that accesses the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read and Endpoint2, a write access to the EP0NKR bit is ignored.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0N	0	0	0	0	0	0	EP0NKR	EP0NKW	FF60H	00H

Bit position	Bit name	Function
1	EP0NKR	This bit controls NAK to the OUT token to Endpoint0 (except an automatically executed request). It is automatically set to 1 by hardware when Endpoint0 has correctly received data. It is also cleared to 0 by hardware when the data of the UF0E0R register has been read by FW (counter value = 0). 1: Transmit NAK. 0: Do not transmit NAK (default value). Set this bit to 1 by FW when data should not be received from the USB bus for some reason even when USBF is ready for receiving data. In this case, USBF continues transmitting NAK until this bit is cleared to 0 by FW. This bit is also cleared to 0 as soon as the UF0E0R register has been cleared.
0	EP0NKW	This bit indicates how NAK to the IN token to Endpoint0 is controlled (except an automatically executed request). This bit is automatically cleared to 0 by hardware when the data of Endpoint0 is transmitted and the host correctly receives the transmitted data. The data of the UF0E0W register is retained until this bit is cleared. Therefore, it is not necessary to rewrite this bit even in the case of a retransmission request that is made if the host could not receive data correctly. To send a short packet, be sure to set the E0DED bit of the UF0DEND register to 1. This bit is automatically set to 1 when the FIFO is full. As soon as the E0DED bit of the UF0DEND register is set to 1, the EP0NKW bit is automatically set to 1 at the same time. 1: Do not transmit NAK. 0: Transmit NAK (default value). If control transfer enters the status stage while ACK cannot be correctly received in the data stage, this bit is cleared to 0 as soon as the UF0E0W register is cleared. This bit is also cleared to 0 when UF0E0W is cleared by FW.

(33) UF0 active alternative setting register (UF0AAS)

This register specifies a link between the Interface number and Alternative Setting.

This register can be read or written in 8-bit units.

USBF of the µPD78F0730 can set a five-series Alternative Setting (Alternate Setting 0, 1, 2, 3, and 4 can be defined) and a two-series Alternative Setting (Alternative Setting 0 and 1 can be defined) for one Interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0AAS	ALT2	IFAL21	IFAL20	ALT2EN	ALT5	IFAL51	IFAL50	ALT5EN	FF71H	00H

Bit position	Bit name	Function
7, 3	ALTn	These bits specify whether an n-series Alternative Setting is linked with Interface 0. When these bits are set to 1, the setting of the IFALn1 and IFALn0 bits is invalid. 1: Link n-series Alternative Setting with Interface 0. 0: Do not link n-series Alternative Setting with Interface 0 (default value).
6, 5, 2, 1	IFALn1, IFALn0	These bits specify the Interface number to be linked with the n-series Alternative Setting. If the linked Interface number is outside the range specified by the UF0AIFN register, the n-series Alternative Setting is invalid (ALTnEN bit = 0).

IFALn1	IFALn0	Interface number to be linked
1	1	Links Interface 4.
1	0	Links Interface 3.
0	1	Links Interface 2.
0	0	Links Interface 1.

Do not link a five-series Alternative Setting and a two-series Alternative Setting with the same Interface number.

4, 0	ALTnEN	These bits validate the n-series Alternative Setting. Unless these bits are set to 1, the setting of the ALTn, IFALn1, and IFALn0 bits is invalid. 1: Validate the n-series Alternative Setting. 0: Do not validate the n-series Alternative Setting (default value).

Remark n = 2, 5

For example, when the UF0AIFN register is set to 82H and the UF0AAS register is set to 15H, Interfaces 0, 1, 2, and 3 are valid. Interfaces 0 and 2 support only Alternative Setting 0. Interface 1 supports Alternative Setting 0 and 1, and Interface 3 supports Alternative Setting 0, 1, 2, 3, and 4. With this setting, requests GET_INTERFACE wIndex = 0/1/2/3, SET_INTERFACE wValue = 0 & wIndex = 0/2, SET_INTERFACE wValue = 0/1 & wIndex = 1, and SET_INTERFACE wValue = 0/1/2/3/4 & wIndex = 3 are automatically responded to, and a STALL response is made to the other GET/SET_INTERFACE requests.

Table 12-8. Register Values in Specific Status (2/2)

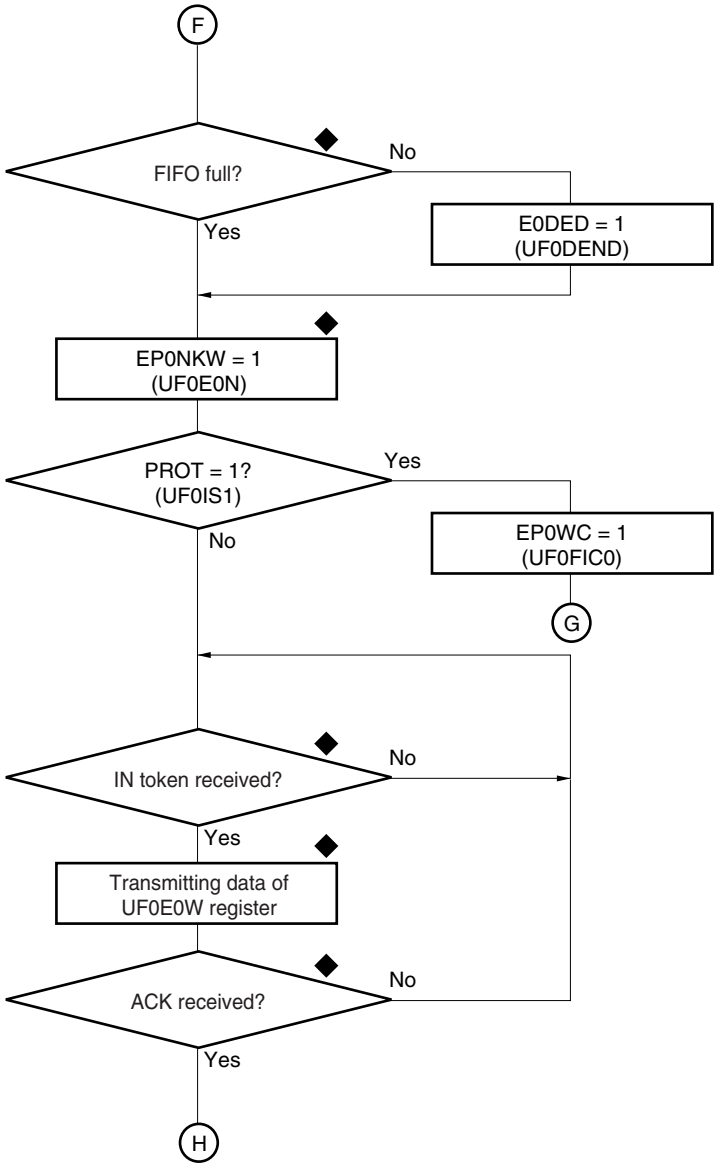
Register Name	After CPU Reset ($\overline{\text{RESET}}$)	After Bus Reset
UF0E0R register	Undefined[Note 1]	Value is held.
UF0E0L register	00H	Value is held.
UF0E0ST register	00H	00H
UF0E0W register	Undefined[Note 1]	Value is held.
UF0BO1 register	Undefined[Note 1]	Value is held.
UF0BO1L register	00H	Value is held.
UF0BI1 register	Undefined[Note 1]	Value is held.
UF0DSTL register	00H	00H
UF0E0SL register	00H	00H
UF0E1SL register	00H	00H
UF0E2SL register	00H	00H
UF0ADRS register	00H	00H
UF0CNF register	00H	00H
UF0IF0 register	00H	00H
UF0IF1 register	00H	00H
UF0IF2 register	00H	00H
UF0IF3 register	00H	00H
UF0IF4 register	00H	00H
UF0DSCL register	00H	Value is held.
UF0DDn register (n = 0 to 17)	**Note 2**	**Note 2**
UF0CIEn register (n = 0 to 255)	**Note 2**	**Note 2**

Notes 1. This register can be cleared to 0 by the $\overline{\text{RESET}}$ signal because its write pointer, counter, and read pointer are cleared to 0 when the $\overline{\text{RESET}}$ signal becomes active, in the same manner as clearing by the UF0FICn register, as the register is controlled by FIFO.

2. This register cannot be cleared to 0. Because data can be written to it by FW, however, any value can be written to the register (before doing so, however, be sure to set the EP0NKA bit of the UF0E0NA register to 1).

Figure 12-19. CPUDEC Request for Control Transfer (4/12)

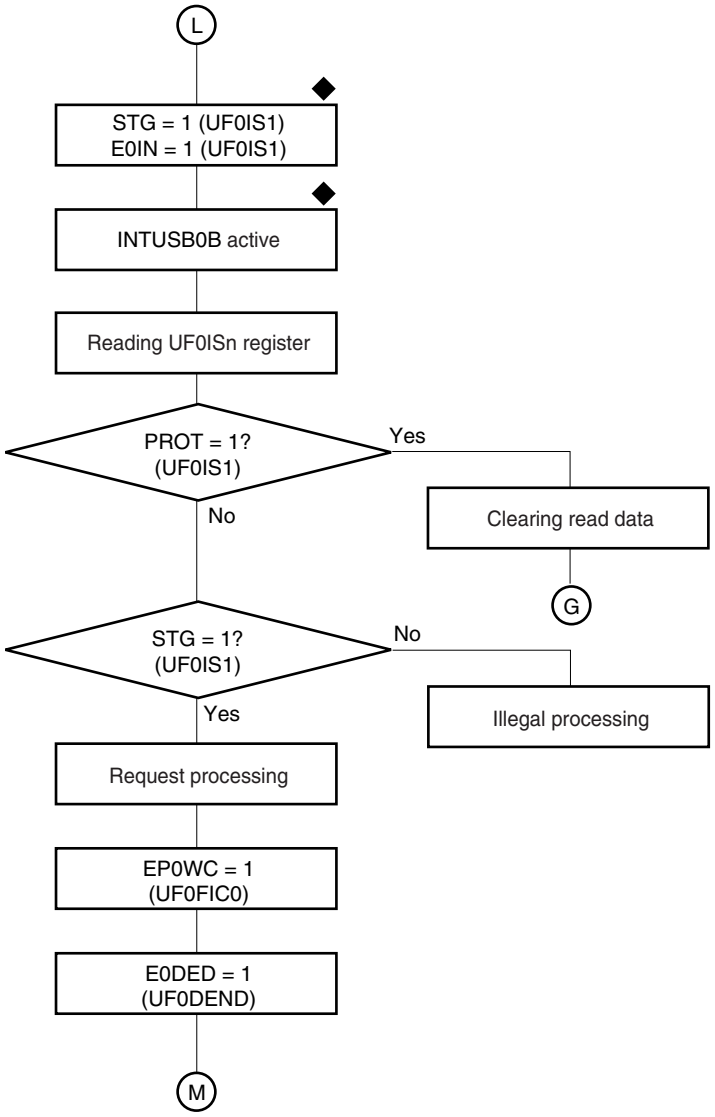
(b) Control transfer (read) (2/4)



Remark ♦: Processing by hardware

Figure 12-19. CPUDEC Request for Control Transfer (9/12)

(c) Control transfer (write) (3/4)

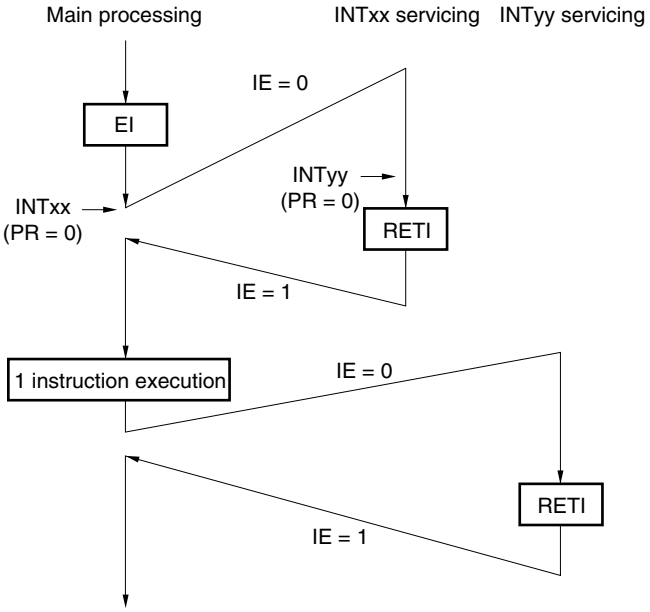


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 13-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledgement disabled

19.7.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the µPD78F0730 in the flash memory programming mode. To set the mode, set the FLMD0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 19-13. Flash Memory Programming Mode

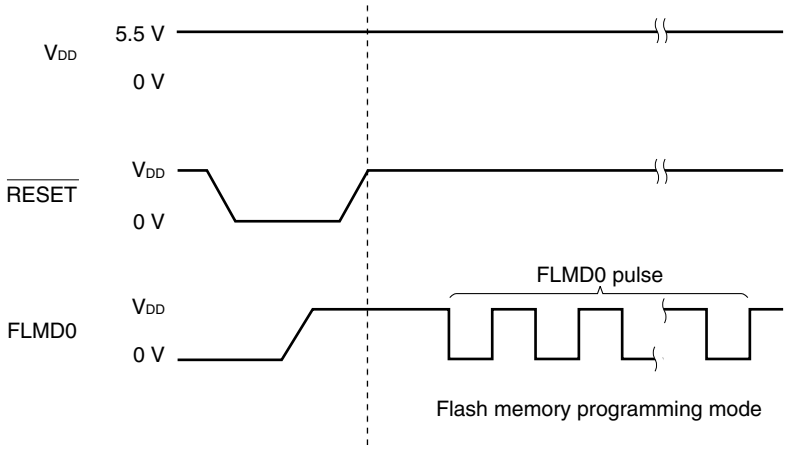


Table 19-4. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

DC Characteristics (1/3)

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00, P01, P10 to P17, P30 to P33, P120	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0	mA
		Total of P00, P01, P10 to P17, P30, P33, P120[Note 3]	4.0 V ≤ V_{DD} ≤ 5.5 V			–20.0	mA
		Total of P31 and P32[Note 3]	4.0 V ≤ V_{DD} ≤ 5.5 V			–6.0	mA
		Total[Note 3] of all pins	4.0 V ≤ V_{DD} ≤ 5.5 V			–26.0	mA
	I_{OH2}	Per pin for P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V			–100	μA
Output current, low[Note 2]	I_{OL1}	Per pin for P00, P01, P10 to P17, P30 to P33, P120	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
		Per pin for P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			15.0	mA
		Total of P00, P01, P10 to P17, P30, P33, P120[Note 3]	4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
		Total of P31, P32, P60, P61[Note 3]	4.0 V ≤ V_{DD} ≤ 5.5 V			45.0	mA
		Total of all pins[Note 3]	4.0 V ≤ V_{DD} ≤ 5.5 V			65.0	mA
	I_{OL2}	Per pin for P121, P122	4.0 V ≤ V_{DD} ≤ 5.5 V			400	μA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} to an output pin.

2. Value of current at which the device operation is guaranteed even if the current flows from an output pin to GND.

3. Specification under conditions where the duty factor is 70% (time for which current is output is 0.7 × t and time for which current is not output is 0.3 × t, where t is a specific time). The total output current of the pins at a duty factor of other than 70% can be calculated by the following expression.

- Where the duty factor of I_{OH} is n%: Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where the duty factor is 50%, I_{OH} = 20.0 mA

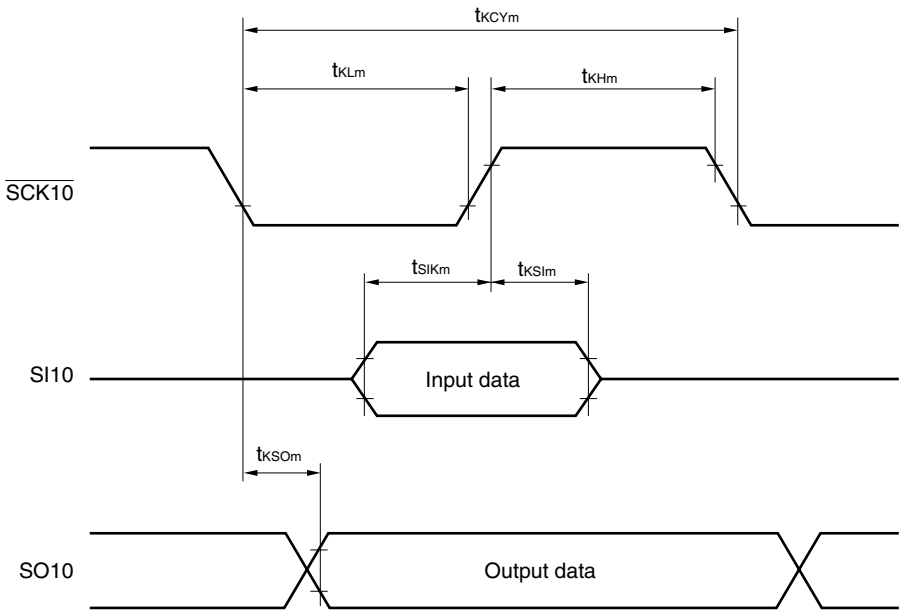
Total output current of pins = (20.0 × 0.7)/(50 × 0.01) = 28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Serial Transfer Timing

CSI10:



Remark m = 1, 2

C.2 Register Index (In Alphabetical Order with Respect to Register Symbol)