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/upd78f9328gb-8et-a

Documents Related to Flash Memory Writing

Document Name	Document No.
PG-FP3 Flash Memory Programmer User's Manual	U13502E
PG-FP4 Flash Memory Programmer User's Manual	U15260E

Other Related Documents

Document Name	Document No.
★ SEMICONDUCTOR SELECTION GUIDE - Products and Packages -	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

★ **Note** See the “Semiconductor Device Mount Manual” website (http://www.necel.com/pkg/en/mount/index.html)

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document for designing.

★ 1.3 Ordering Information

Part Number	Package	Internal ROM
μPD789322GB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD789324GB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD789326GB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD789327GB-×××-8ET	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD78F9328GB-8ET	52-pin plastic LQFP (10 × 10)	Flash memory
μPD789322GB-×××-8ET-A	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD789324GB-×××-8ET-A	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD789326GB-×××-8ET-A	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD789327GB-×××-8ET-A	52-pin plastic LQFP (10 × 10)	Mask ROM
μPD78F9328GB-8ET-A	52-pin plastic LQFP (10 × 10)	Flash memory

Remarks 1. ××× indicates ROM code suffix.

2. Products that have the part numbers suffixed by "-A" are lead-free products.

The major differences between subseries are shown below.

Series for General-Purpose and LCD Drive

Subseries \ Function		ROM Capacity (Bytes)	Timer				8-Bit A/D	10-Bit A/D	Serial Interface	I/O	V_{DD} MIN.Value	Remarks
			8-Bit	16-Bit	Watch	WDT						
Small-scale package, general-purpose applications	μPD789046	16 K	1 ch	1 ch	1 ch	1 ch	–	–	1 ch (UART: 1 ch)	34	1.8 V	–
	μPD789026	4 K to 16 K			–							
	μPD789088	16 K to 32 K	3 ch							24		
	μPD789074	2 K to 8 K	1 ch									
	μPD789062	4 K	2 ch	–					–	14		RC-oscillation version
	μPD789052											–
Small-scale package, general-purpose applications + A/D converter	μPD789177	16 K to 24 K	3 ch	1 ch	1 ch	1ch	–	8 ch	1 ch (UART: 1 ch)	31	1.8 V	–
	μPD789167						8 ch	–				
	μPD789134A	2 K to 8 K	1 ch		–		–	4 ch		20		RC-oscillation version
	μPD789124A						4 ch	–				
	μPD789114A						–	4 ch				–
	μPD789104A						4 ch	–				
LCD drive	μPD789835	24 K to 60 K	6 ch	–	1 ch	1 ch	3 ch	–	1 ch (UART: 1 ch)	37	1.8 V[Note]	Dot LCD supported
	μPD789830	24 K	1 ch	1 ch			–			30	2.7 V	
	μPD789489	32 K to 48 K	3 ch					8 ch	2 ch (UART: 1 ch)	45	1.8 V	–
	μPD789479	24 K to 48 K					8 ch	–				
	μPD789417A	12 K to 24 K					–	7 ch	1 ch (UART: 1 ch)	43		
	μPD789407A						7 ch	–				
	μPD789456	12 K to 16 K	2 ch				–	6 ch		30		
	μPD789446						6 ch	–				
	μPD789436						–	6 ch		40		
	μPD789426						6 ch	–				
	μPD789316	8 K to 16 K					–		2 ch (UART: 1 ch)	23		RC-oscillation version
	μPD789306											–
	μPD789467	4 K to 24 K		–			1 ch		–	18		
	μPD789327						–		1 ch	21		

Note Flash memory version: 3.0 V

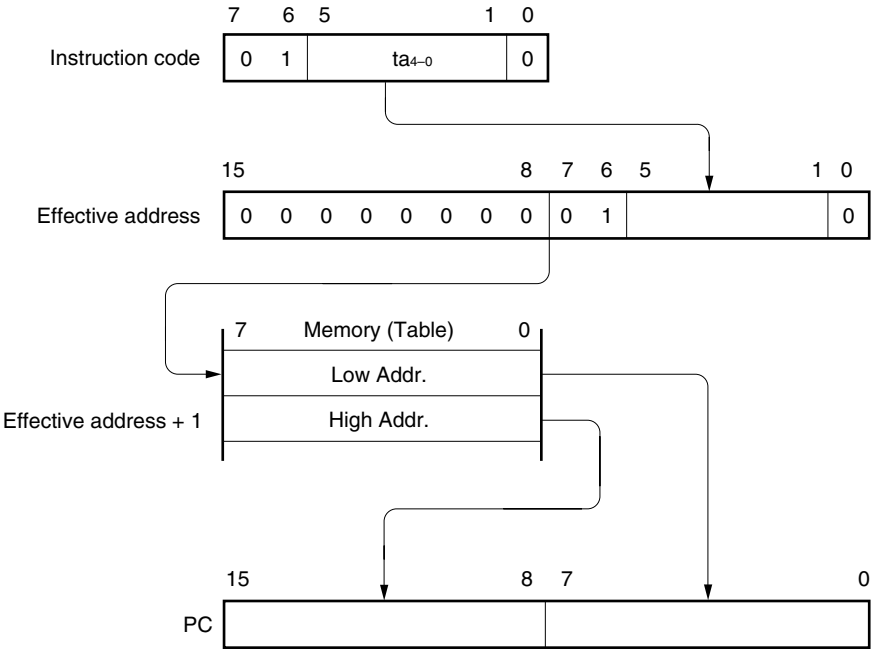
3.3.3 Table indirect addressing

[Function]

Table contents (branch destination address) of the particular location to be addressed by the lower 5-bit immediate data of an instruction code from bit 1 to bit 5 are transferred to the program counter (PC) and branched.

This function is carried out when the CALLT [addr5] instruction is executed. The instruction enables a branch to any location in the memory space by referring to the addresses stored in the memory table at 40H to 7FH.

[Illustration]



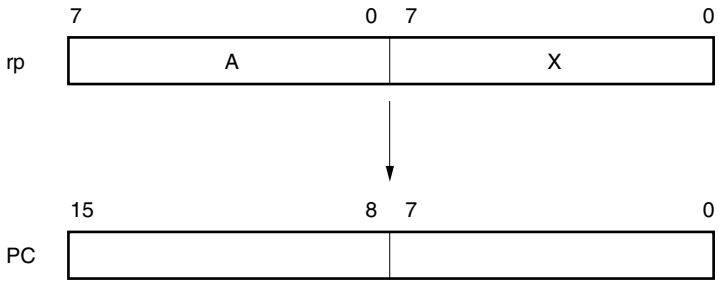
3.3.4 Register addressing

[Function]

The register pair (AX) contents to be specified with an instruction word are transferred to the program counter (PC) and branched.

This function is carried out when the BR AX instruction is executed.

[Illustration]



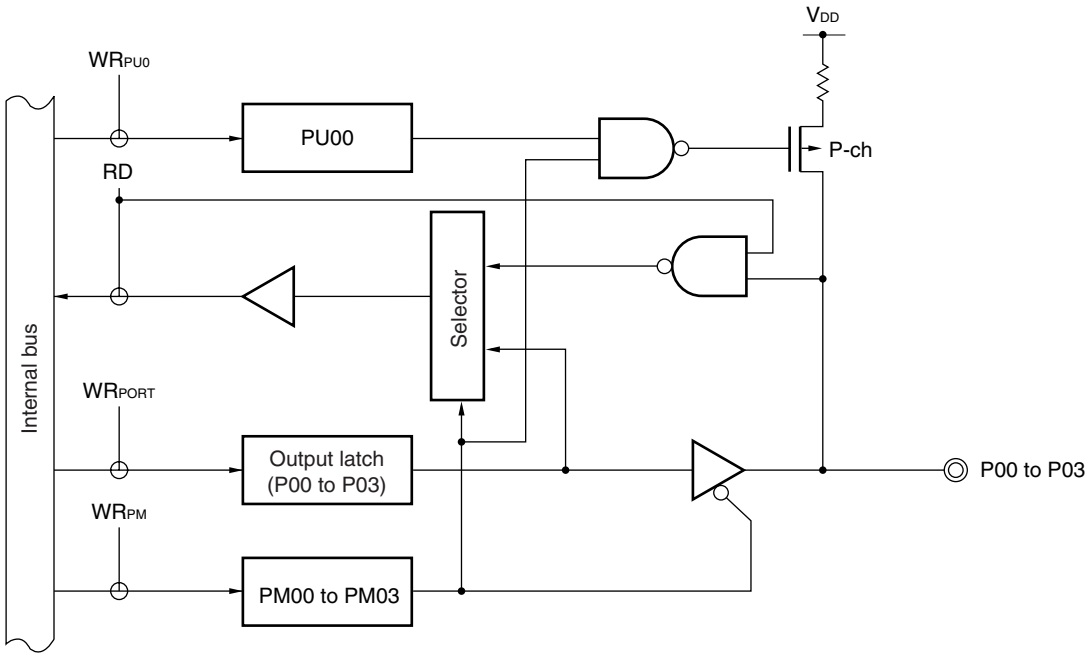
4.2.1 Port 0

Port 0 is a 4-bit I/O port with an output latch. It can be specified in the input or output mode in 1-bit units by using the port mode register 0 (PM0). When the P00 to P03 pins are used as input port pins, on-chip pull-up resistors can be connected in 4-bit units by using pull-up resistor option register 0 (PU0).

$\overline{\text{RESET}}$ input sets port 0 in the input mode.

Figure 4-2 shows a block diagram of port 0.

Figure 4-2. Block Diagram of P00 to P03



PU0: Pull-up resistor option register 0
PM: Port mode register
RD: Port 0 read signal
WR: Port 0 write signal

Figure 5-1. Block Diagram of Clock Generator

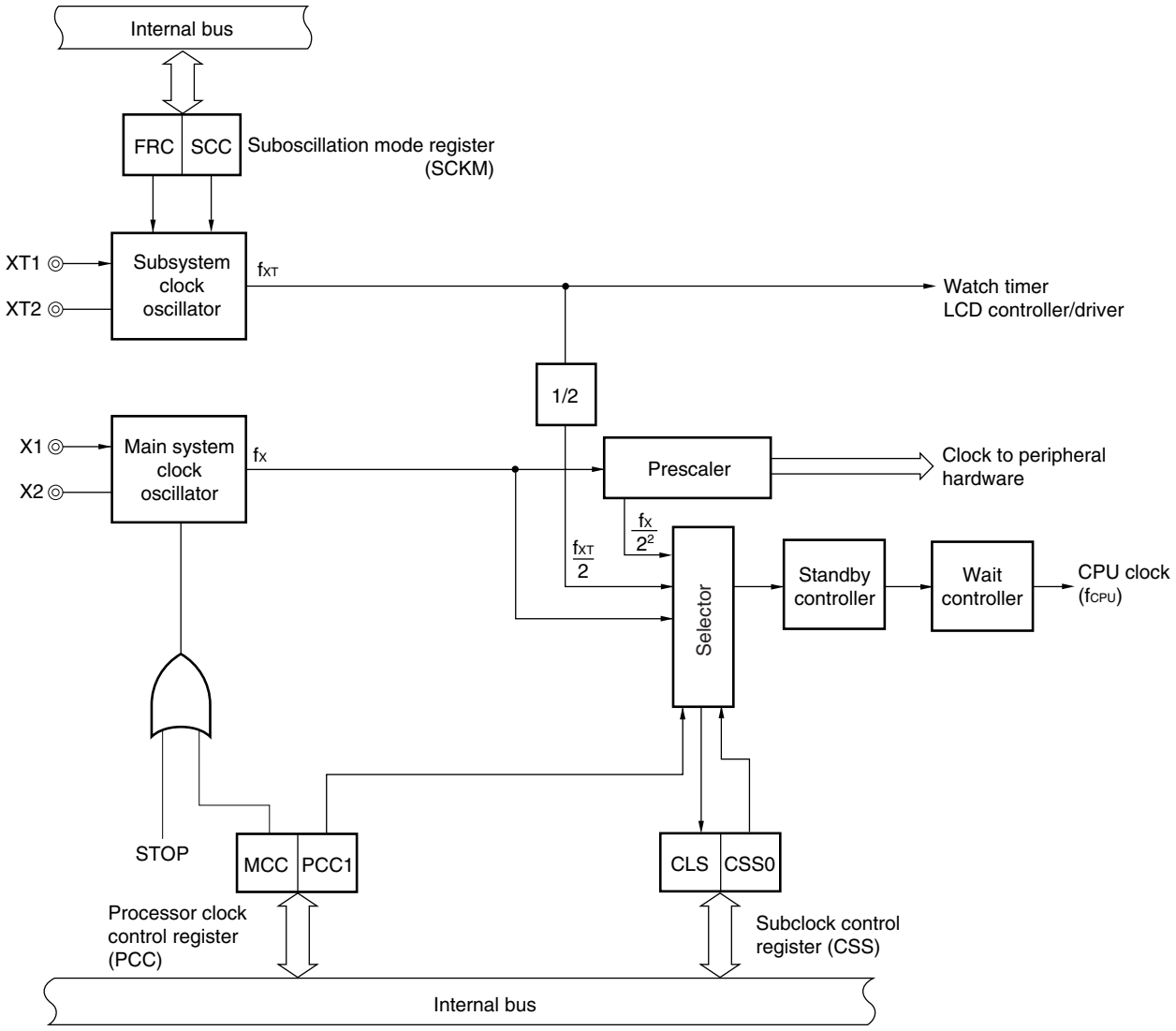


Table 6-3. Interval Time of Timer 30 (at fx = 5.0 MHz Operation)

TCL301	TCL300	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	2^6/fx (12.8 μs)	2^{14}/fx (3.28 ms)	2^4/fx (12.8 μs)
0	1	2^8/fx (51.2 μs)	2^{16}/fx (13.1 ms)	2^8/fx (51.2 μs)
1	0	Input cycle of timer 40 match signal	Input cycle of timer 40 match signal $\times 2^8$	Input cycle of timer 40 match signal
1	1	Carrier clock cycle created with timer 40	Carrier clock cycle created with timer 40 $\times 2^8$	Carrier clock cycle created with timer 40

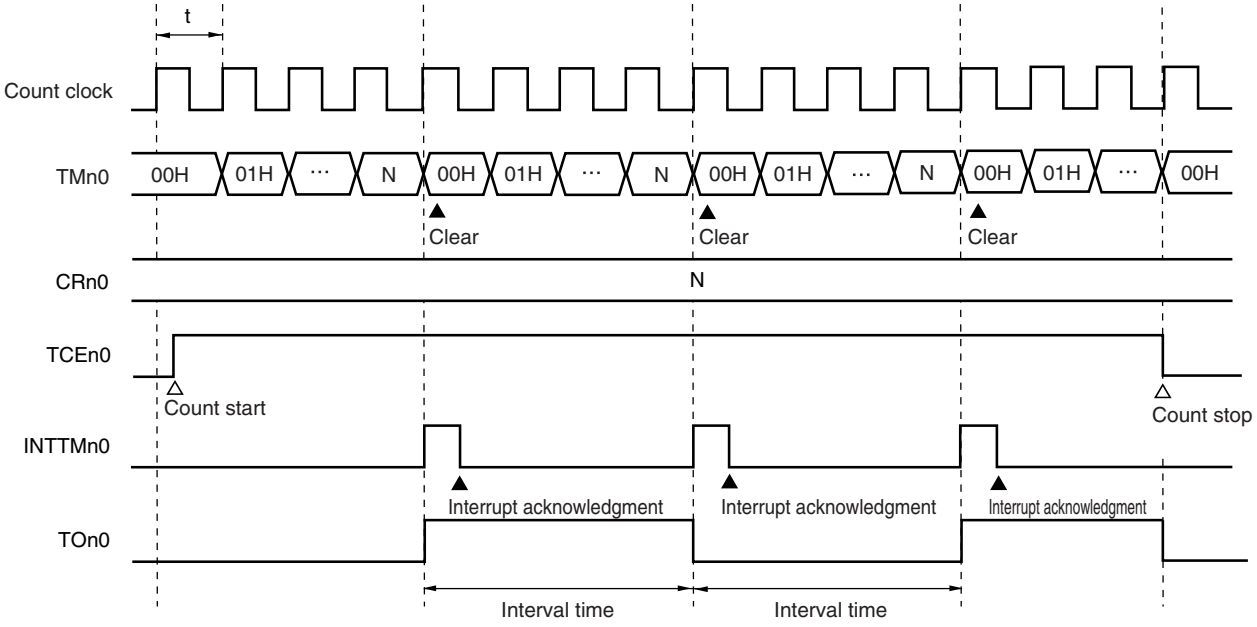
Remark fx: Main system clock oscillation frequency

Table 6-4. Interval Time of Timer 40 (at fx = 5.0 MHz Operation)

TCL402	TCL401	TCL400	Minimum Interval Time	Maximum Interval Time	Resolution
0	0	0	1/fx (0.2 μs)	2^8/fx (51 μs)	1/fx (0.2 μs)
0	0	1	2^2/fx (0.8 μs)	2^{10}/fx (205 μs)	2^2/fx (0.8 μs)
0	1	0	2/fx (0.4 μs)	2^9/fx (102 μs)	2/fx (0.4 μs)
0	1	1	2^2/fx (0.8 μs)	2^{10}/fx (205 μs)	2^2/fx (0.8 μs)
1	0	0	2^3/fx (1.6 μs)	2^{11}/fx (410 μs)	2^3/fx (1.6 μs)
1	0	1	2^4/fx (3.2 μs)	2^{12}/fx (819 μs)	2^4/fx (3.2 μs)

Remark fx: Main system clock oscillation frequency

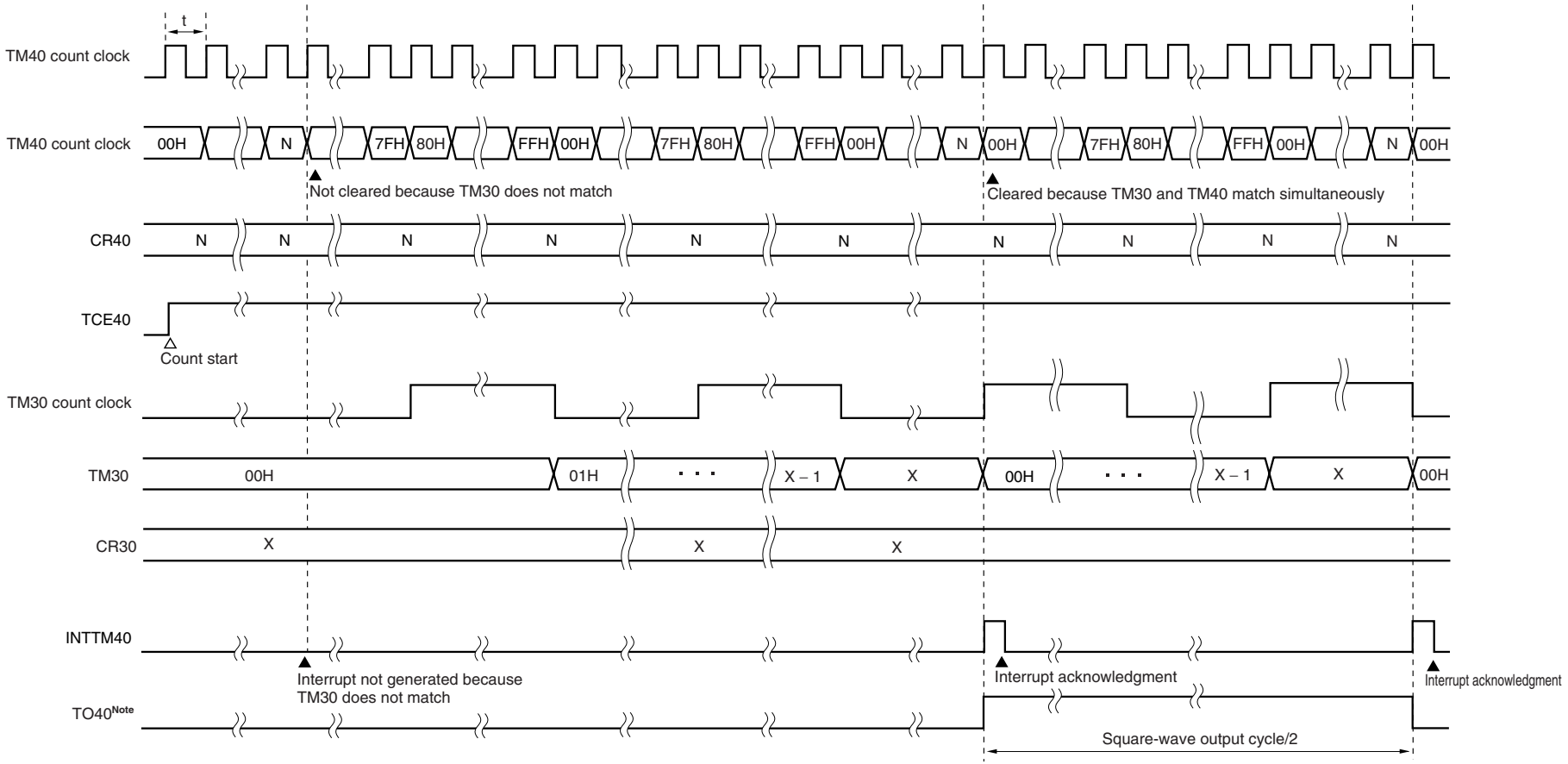
Figure 6-8. Timing of Interval Timer Operation with 8-Bit Resolution (Basic Operation)



Remarks 1. Interval time = (N + 1) × t: N = 00H to FFH

2. n = 3, 4

Figure 6-16. Timing of Square-Wave Output with 16-Bit Resolution



Note The initial value of TO40 is low level when output is enabled (TOE40 = 1).

Remark Square-wave output cycle = 2 (256X + N + 1) × t: X = 00H to FFH, N = 00H to FFH

6.4.3 Operation as carrier generator

An arbitrary carrier clock generated by TM40 can be output in the cycle set in TM30.

To operate timers 30 and 40 as carrier generators, settings must be made in the following sequence.

<1> Disable operation of TM30 and TM40 (TCE30 = 0, TCE40 = 0).

<2> Disable timer output of TO40 (TOE40 = 0).

<3> Set count values in CR30, CR40, and CRH40.

<4> Set the operation mode of timer 30 and timer 40 to carrier generator mode (see **Figures 6-4** and **6-5**).

<5> Set the count clock for timer 30 and timer 40.

<6> Set remote control output to carrier pulse (RMC40 (bit 2 of carrier generator output control register 40 (TCA40)) = 0).
Input the required value to NRZB40 (bit 1 of TCA40) by program.
Input a value to NRZ40 (bit 0 of TCA40) before it is reloaded from NRZB40.

<7> Set P60 to output mode (PM60 = 0) and the P60 output latch to 0 and enable TO40 output by setting TOE40 to 1.

<8> Enable the operation of TM30 and TM40 (TCE30 = 1, TCE40 = 1).

★ <9> Save the NRZB40 value to a general-purpose register.

★ <10> When INTTM30 rises, the NRZB40 value is transferred to NRZ40. After that, rewrite TCA40 using an 8-bit memory manipulation instruction. Input the value to be transferred next to NRZ40 to NRZB40, and input the value saved in step <9> to NRZ40.

★ <11> Generate the desired carrier signal by repeating steps <9> and <10>.

The operation of the carrier generator is as follows.

<1> When the count the value of TM40 matches the value set in CR40, an interrupt request signal (INTTM40) is generated and the output of timer 40 is inverted, which makes the compare register switch from CR40 to CRH40.

<2> After that, when the count the value of TM40 matches the value set in CRH40, an interrupt request signal (INTTM40) is generated and the output of timer 40 is inverted again, which makes the compare register switch from CRH40 to CR40.

<3> The carrier clock is generated by repeating <1> and <2> above.

<4> When the count value of TM30 matches the value set in CR30, an interrupt request signal (INTTM30) is generated. The rising edge of INTTM30 is the data reload signal of NRZB40 and is transferred to NRZ40.

<5> When NRZ40 is 1, a carrier clock is output from TO40 pin.

Cautions

1. **TCA40 cannot be set with a 1-bit memory manipulation instruction. Be sure to use an 8-bit memory manipulation instruction.**
2. **The NRZ40 flag can be rewritten only when the carrier generator output is stopped (TOE40 = 0). The data of the flag is not changed even if a write instruction is executed while TOE40 = 1.**
3. **When the carrier generator is stopped once and then started again, NRZB40 does not hold the previous data. Re-set data to NRZB40. At this time, a 1-bit memory manipulation instruction must not be used. Be sure to use an 8-bit memory manipulation instruction.**
4. **To enable operation in the carrier generator mode, set a value to the compare registers (CR30, CR40, and CRH40), and input the necessary value to the NRZB40 and NRZ40 flags in advance. Otherwise, the signal of the timer match circuit will become unstable and the NRZ40 flag will be undefined.**

Figures 6-17 to 6-19 show the operation timing of the carrier generator.

Figure 6-17. Timing of Carrier Generator Operation (When CR40 = N, CRH40 = M (M > N))

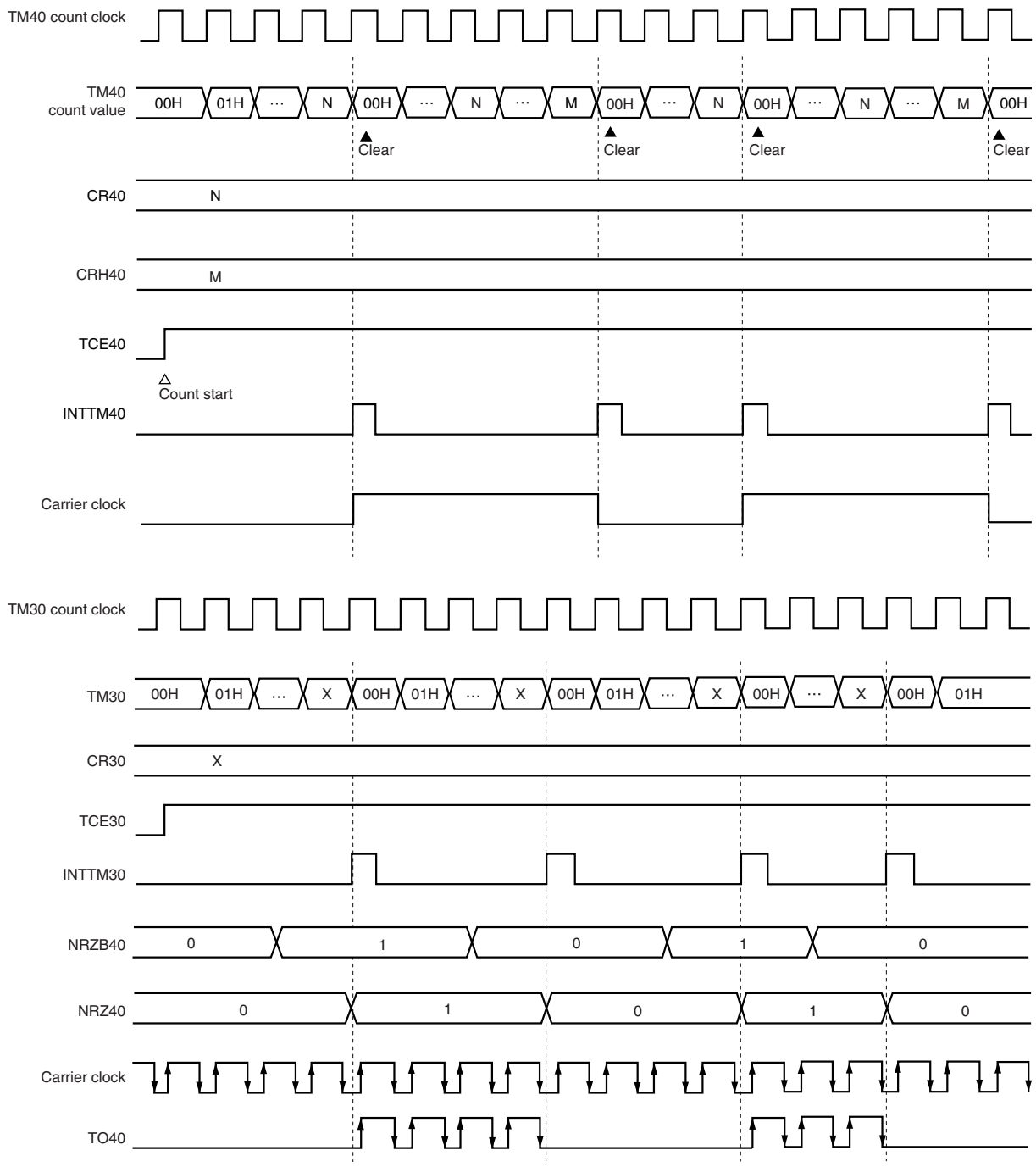
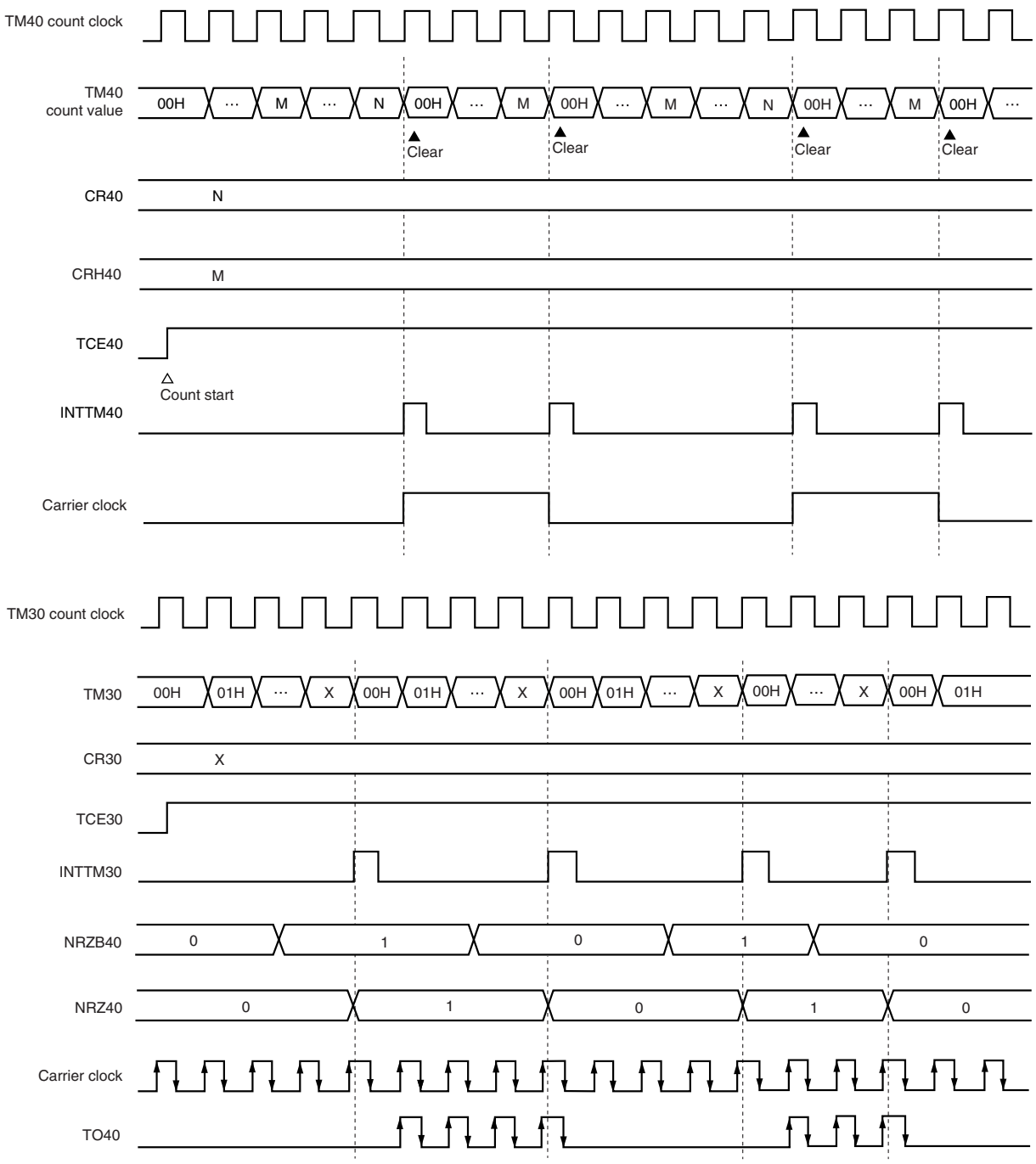
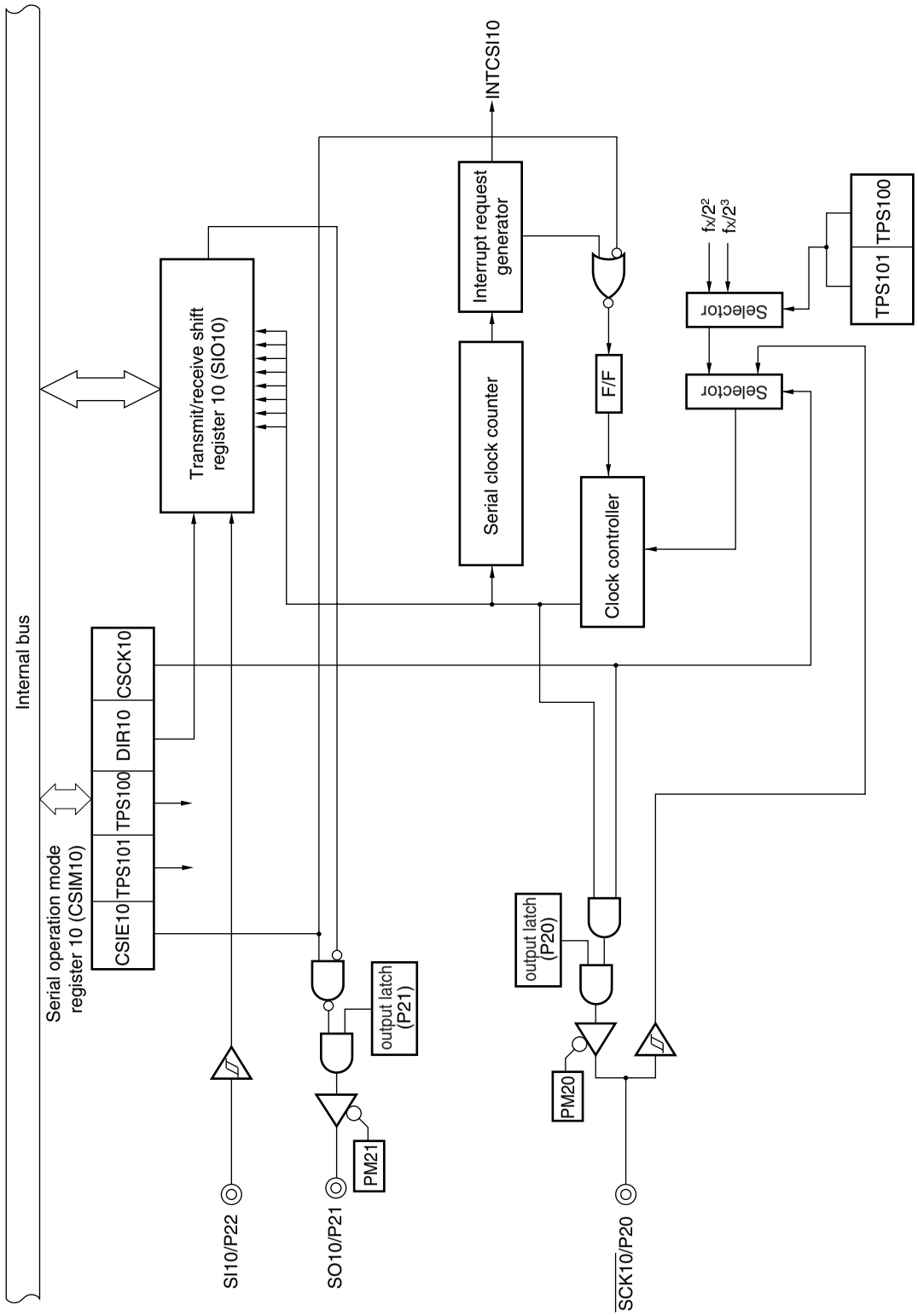


Figure 6-18. Timing of Carrier Generator Operation (When CR40 = N, CRH40 = M (M < N))



★ **Remark** This figure shows an example of when the NRZ40 value is changed while the carrier clock is high level.

★ **Figure 9-1. Block Diagram of Serial Interface 10**



9.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is useful for connection of peripheral I/O and display controllers, etc., which incorporate a conventional clocked serial interface, such as the 75XL Series, 78K Series, 17K Series.

Communication is performed using three lines: a serial clock line ($\overline{SCK10}$), serial output line (SO10), and serial input line (SI10).

(1) Register setting

3-wire serial I/O mode settings are performed using serial operation mode register 10 (CSIM10).

(a) Serial operation mode register 10 (CSIM10)

CSIM10 is set with a 1-bit or 8-bit memory manipulation instruction.

$\overline{RESET}$ input clears CSIM10 to 00H.

Symbol	<7>	6	5	4	3	2	1	0	Address	After reset	R/W
CSIM10	CSIE10	0	TPS101	TPS100	0	DIR10	CSCK10	0	FF72H	00H	R/W

CSIE10	Operation control in 3-wire serial I/O mode
0	Operation stopped
1	Operation enabled

TPS101	TPS100	Count clock selection when internal clock is selected
0	0	$f_X/2^2$ (1.25 MHz)
0	1	$f_X/2^3$ (625 kHz)
Other than above		Setting prohibited

DIR10	Start bit specification
0	MSB
1	LSB

CSCK10	SIO10 clock selection
0	Input clock to $\overline{SCK10}$ pin from external
1	Internal clock selected by TPS100, TPS101

Cautions 1. Bits 0, 3, and 6 must be set to 0.

2. Switch operation mode after stopping the serial transmit/receive operation.

Remarks 1. f_X: Main system clock oscillation frequency

2. The parenthesized values apply to operation at f_X = 5.0 MHz.

CHAPTER 10 LCD CONTROLLER/DRIVER

10.1 LCD Controller/Driver Functions

The functions of the LCD controller/driver of the μPD789327 Subseries are as follows.

(1) Automatic output of segment and common signals based on automatic display data memory read
(2) Two different display modes:
★ • Static
• 1/4 duty (1/3 bias)
(3) Four different frame frequencies, selectable in each display mode
(4) Up to 24 segment signal outputs (S0 to S23) and four common signal outputs (COM0 to COM3)
(5) Operation with a subsystem clock

Table 10-1 lists the maximum number of pixels that can be displayed in each display mode.

Table 10-1. Maximum Number of Pixels

Bias Mode	Number of Time Slices	Common Signals Used	Maximum Number of Pixels
★ –	Static	COM0 (COM1 to COM3)	24 (24 segments × 1 common)[Note 1]
1/3	4	COM0 to COM3	96 (24 segments × 4 commons)[Note 2]

Notes 1. 3-digit LCD panel, each digit having an 8-segment 8. configuration.
2. 12-digit LCD panel, each digit having a 2-segment 8. configuration.

10.2 LCD Controller/Driver Configuration

The LCD controller/driver consists of the following hardware.

Table 10-2. Configuration of LCD Controller/Driver

Item	Configuration
Display outputs	Segment signals: 24 Common signals: 4
Control registers	LCD display mode register 0 (LCDM0) LCD clock control register 0 (LCDC0) Port function register 8 (PF8)

CHAPTER 13 STANDBY FUNCTION

13.1 Standby Function and Configuration

The standby function is to reduce the power consumption of the system and can be effected in the following two modes:

(1) HALT mode

This mode is set when the HALT instruction is executed. The HALT mode stops the operation clock of the CPU. The system clock oscillator continues oscillating. This mode does not reduce the operating current as much as the STOP mode, but is useful for resuming processing immediately when an interrupt request is generated, or for intermittent operations.

(2) STOP mode

This mode is set when the STOP instruction is executed. The STOP mode stops the main system clock oscillator and stops the entire system. The power consumption of the CPU can be substantially reduced in this mode.

The data memory can be retained at the low voltage (V_{DD} = 1.8 V). Therefore, this mode is useful for retaining the contents of the data memory at an extremely low operating current.

The STOP mode can be released by an interrupt request, so that this mode can be used for intermittent operation. However, some time is required until the system clock oscillator stabilizes after the STOP mode has been released. If processing must be resumed immediately by using an interrupt request, therefore, use the HALT mode.

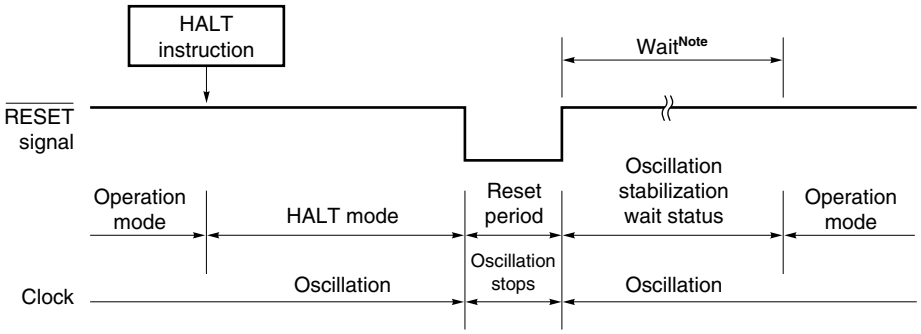
In both modes, the previous contents of the registers, flags, and data memory before setting the standby mode are all retained. In addition, the statuses of the output latch of the I/O ports and output buffer are also retained.

Caution To set the STOP mode, be sure to stop the operations of the peripheral hardware, and then execute the STOP instruction.

(c) Releasing by $\overline{\text{RESET}}$ input

When the HALT mode is released by the $\overline{\text{RESET}}$ signal, execution branches to the reset vector address in the same manner as the ordinary reset operation, and program execution is started.

Figure 13-3. Releasing HALT Mode by $\overline{\text{RESET}}$ Input



Note 2^{15}/fx: 6.55 ms (@ fx = 5.0 MHz operation)

Remark fx: Main system clock oscillation frequency

Table 13-3. Operation After Releasing HALT Mode

Releasing Source	MK××	IE	Operation
Maskable interrupt request	0	0	Executes next address instruction
	0	1	Executes interrupt servicing
	1	×	Retains HALT mode
Non-maskable interrupt request	–	×	Executes interrupt servicing
$\overline{\text{RESET}}$ input	–	–	Reset processing

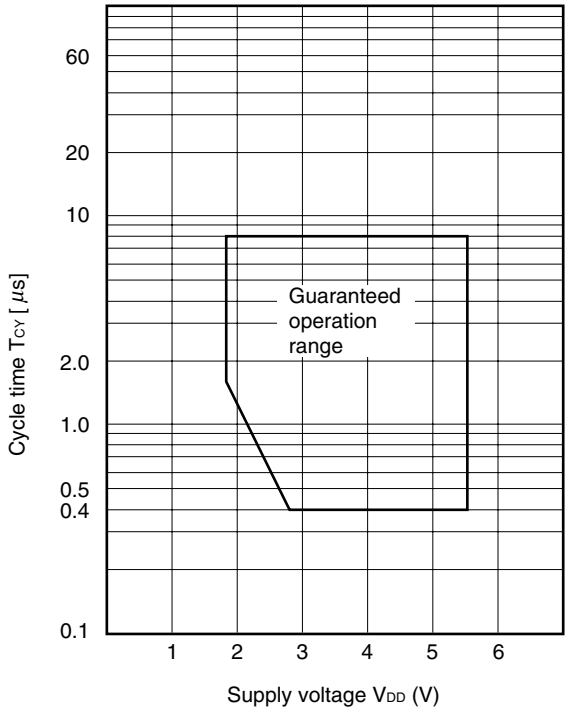
×: don't care

AC Characteristics

(1) Basic operation (T_A = –40 to +85°C, V_{DD} = 1.8 to 5.5 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Cycle time (Min. instruction execution time)	T_{CY}	2.7 ≤ V_{DD} ≤ 5.5 V	0.4		8.0	μs
		1.8 ≤ V_{DD} ≤ 5.5 V	1.6		8.0	μs
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INT	10			μs
Key return pin low-level width	t_{KRIL}	KR00 to KR03	10			μs
$\overline{RESET}$ low-level width	t_{RSL}		10			μs

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



A.1 Software Package

SP78K0S Software package	Various software tools for 78K/0S Series development are integrated into one package. The following tools are included. RA78K0S, CC78K0S, ID78K0S-NS, SM78K0S, various device files
	Part number: μS××××SP78K0S

Remark ×××× in the part number differs depending on the operating system to be used.

μS××××SP78K0S

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Japanese Windows	CD-ROM
BB17		English Windows	

A.2 Language Processing Software

RA78K0S Assembler package	Program that converts program written in mnemonic into object codes that can be executed by microcontroller. In addition, automatic functions to generate symbol tables and optimize branch instructions are also provided. Used in combination with a device file (DF789328) (sold separately). **<Caution when used in PC environment>** The assembler package is a DOS-based application but may be used in the Windows environment by using the Project Manager of Windows (included in the assembler package).
	Part number: μS××××RA78K0S
CC78K0S C compiler package	Program that converts program written in C language into object codes that can be executed by microcontroller. Used in combination with an assembler package (RA78K0S) and device file (DF789328) (both sold separately). **<Caution when used in PC environment>** The C compiler package is a DOS-based application but may be used in the Windows environment by using the Project Manager of Windows (included in the assembler package).
	Part number: μS××××CC78K0S
DF789328 Note1 Device file	File containing the information specific to the device. Used in combination with the RA78K0S, CC78K0S, and SM78K0S (sold separately).
	Part number: μS××××DF789328
CC78K0S-L Note2 C library source file	Source file of functions for generating object library included in C compiler package. Necessary for changing object library included in C compiler package according to customer's specifications. Since this is a source file, its working environment does not depend on any particular operating system.
	Part number: μS××××CC78K0S-L

Notes 1. DF789328 is a common file that can be used with the RA78K0S, CC78K0S, ID78K0S-NS, and SM78K0S.

2. CC78K0S-L is not included in the software package (SP78K0S).

C.2 Revision History of Previous Editions

The revision history of this edition is listed in the table below. “Chapter” indicates the chapter of the previous edition where the revision was made.

(1/2)

Edition	Revisions	Chapter
2nd	Use of POC circuit of mask ROM product can be selected by mask option.	Throughout
	Deletion of description “input mode only” from condition of connecting on-chip pull-up resistor of port 2	**CHAPTER 2 PIN FUNCTIONS**
	Change of processing of V_{PP} pin	
	Change of processing of V_{LC0} pin when it is not used	
	Change of **Table 3-3 Special Function Register List**	**CHAPTER 3 CPU ARCHITECTURE**
	Deletion of description “input mode only” from condition of connecting on-chip pull-up resistor of port 2	**CHAPTER 4 PORT FUNCTIONS**
	Change of block diagram of P20, P21, P22, and P80 to P85	
	Modification of description on PM8 in **Figure 4-11 Format of Port Mode Register**	
	Addition of **Caution** in **Figure 4-13 Format of Pull-up Resistor Option Register B2**	
	Addition of note on feedback resistor	**CHAPTER 5 CLOCK GENERATOR**
	Modification of description on switching CPU clock	
	Modification of **Figure 6-6 Format of Carrier Generator Output Control Register 40**	**CHAPTER 6 8-BIT TIMERS 30 AND 40**
	Correction of value in **Table 6-3 Interval Time of Timer 30 (at f_X = 5.0 MHz Operation)**	
	Addition of **Caution** to **6.4.3 Operation as carrier generator**	
	Modification of **Figure 10-1 Block Diagram of LCD Controller/Driver**	**CHAPTER 10 LCD CONTROLLER/DRIVER**
	Modification of **Figure 10-2 Format of LCD Display Mode Register 0**	
	Modification of **Caution** in **Figure 10-3 Format of LCD Clock Control Register**	
	Revision of description in 1**0.4 Setting LCD Controller/Driver**	
	Correction of description in **Table 12-2 Flags Corresponding to Interrupt Request Signal Name**	**CHAPTER 12 INTERRUPT FUNCTIONS**
	Modification of bit 2 in **Figure 12-2 Format of Interrupt Request Flag Register 0** to CSIIF10 and addition of **Caution**	
	Modification of bit 2 in **Figure 12-3 Format of Interrupt Mask Flag Register 0** to CSIMK10	
	Modification of description in **12.3 (3) External interrupt mode register 0 (INTM0)** so that 1-bit memory manipulation instruction can be used	
	Addition of explanation to **Figure 12-6 Format of Key Return Mode Register 00**	
	Modification of description on OSTS in **13.2 Register Controlling Standby Function** so that 1-bit memory manipulation instruction can be used	**CHAPTER 13 STANDBY FUNCTION**

(2/2)

Edition	Revisions	Chapter
2nd	Addition of **Figure 14-5 Reset Timing by Power-on Clear**	**CHAPTER 14 RESET FUNCTION**
	Modification of value of port mode register 8 (PM8) in **Table 14-1 Hardware Status After Reset**	
	Addition of description on power-on clear circuit and oscillation stabilization wait time to **Table 15-1 Differences Between μPD78F9328 and Mask ROM Versions**	**CHAPTER 15 μPD78F9328**
	Total revision of description on flash memory programming as **15.1 Flash Memory Characteristics**	
	Addition of electrical specifications	**CHAPTER 18 ELECTRICAL SPECIFICATIONS**
	Addition of package drawing	**CHAPTER 19 PACKAGE DRAWING**
	Addition of recommended soldering conditions	**CHAPTER 20 RECOMMENDED SOLDERING CONDITIONS**
	Total revision of description of development tools Deletion of embedded software	**APPENDIX A DEVELOPMENT TOOLS**
	Addition of revision history	**APPENDIX C REVISION HISTORY**