TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/L1 Series

TMP91C824FG

Not Recommended for New Design

TOSHIBA CORPORATION

Semiconductor Company

Downloaded from Arrow.com.

Preface

Thank you very much for making use of Toshiba microcomputer LSIs.
Before use this LSI, refer the section, “Points of Note and Restrictions”.
Especially, take care below cautions.

CAUTION
How to release the HALT mode

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{NMI}$, INT0 to INT3, INTRTC, INTALM0 to INTALM4), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

Not Recommended for New Design

Downloaded from Arrow.com.

CMOS 16-Bit Microcontrollers

TMP91C824FG/JTMP91C824-S

1. Outline and Features

TMP91C824 is a high-speed 16-bit microcontroller designed for the control of various mid- to large-scale equipment.

TMP91C824FG comes in a 100-pin flat package. JTMP91C824-S is a chip form product.

Listed below are the features.

(1) High-speed 16-bit CPU (900/L1 CPU)
- Instruction mnemonics are upward-compatible with TLCS-90
- 16 Mbytes of linear address space
- General-purpose registers and register banks
- 16-bit multiplication and division instructions; bit transfer and arithmetic instructions
- Micro DMA: 4 channels (485 ns/2 bytes at 33 MHz)

(2) Minimum instruction execution time: 121 ns (at 33 MHz)

(3) Built-in RAM: 8 Kbytes
Built-in ROM: None

RESTRICTIONS ON PRODUCT USE

20070701-EN

- The information contained herein is subject to change without notice.
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc.
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in his document shall be made at the customer's own risk.
- The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patents or other rights of TOSHIBA or the third parties.
- Please contact your sales representative for product-by-product details in this document regarding RoHS compatibility. Please use these products in this document in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances. Toshiba assumes no liability for damage or losses occurring as a result of noncompliance with applicable laws and regulations.
- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions.

Not Recommended for New Design

Downloaded from Arrow.com.

(4) External memory expansion
- Expandable up to 106 Mbytes (shared program/data area)
- Can simultaneously support 8-/16-bit width external data bus
  Dynamic data bus sizing
- Separate bus system

(5) 8-bit timers: 4 channels

(6) General-purpose serial interface: 2 channels
- UART/Synchronous mode: 2 channels
- IrDA Ver.1.0 (115.2 kbps) mode selectable: 1 channel

(7) Serial bus interface: 1 channel
- I^2C bus mode/clock synchronous mode selectable

(8) Timer for real-time clock (RTC)
- Based on TC8521A

(9) 10-bit AD converter: 8 channels

(10) Watchdog timer

(11) Melody/alarm generator
- Melody: Output of clock 4 to 5461 Hz
- Alarm: Output of the 8 kinds of alarm pattern
- Output of the 5 kinds of interval interrupt

(12) Chip select/wait controller: 4 channels

(13) Memory management unit
- Expandable up to 106 Mbytes (4 local areas/8-bank method)

(14) Interrupts: 37 interrupts
- 9 CPU interrupts: Software interrupt instruction and illegal instruction
- 23 internal interrupts: 7 priority levels are selectable
- 5 external interrupts: 7 priority levels are selectable
  (among 4 interrupts are selectable edge mode)

(15) Input/output ports: 35 pins (at external 16-bit data bus memory)

(16) Standby function
Three HALT modes: IDLE2 (Programmable), IDLE1 and STOP

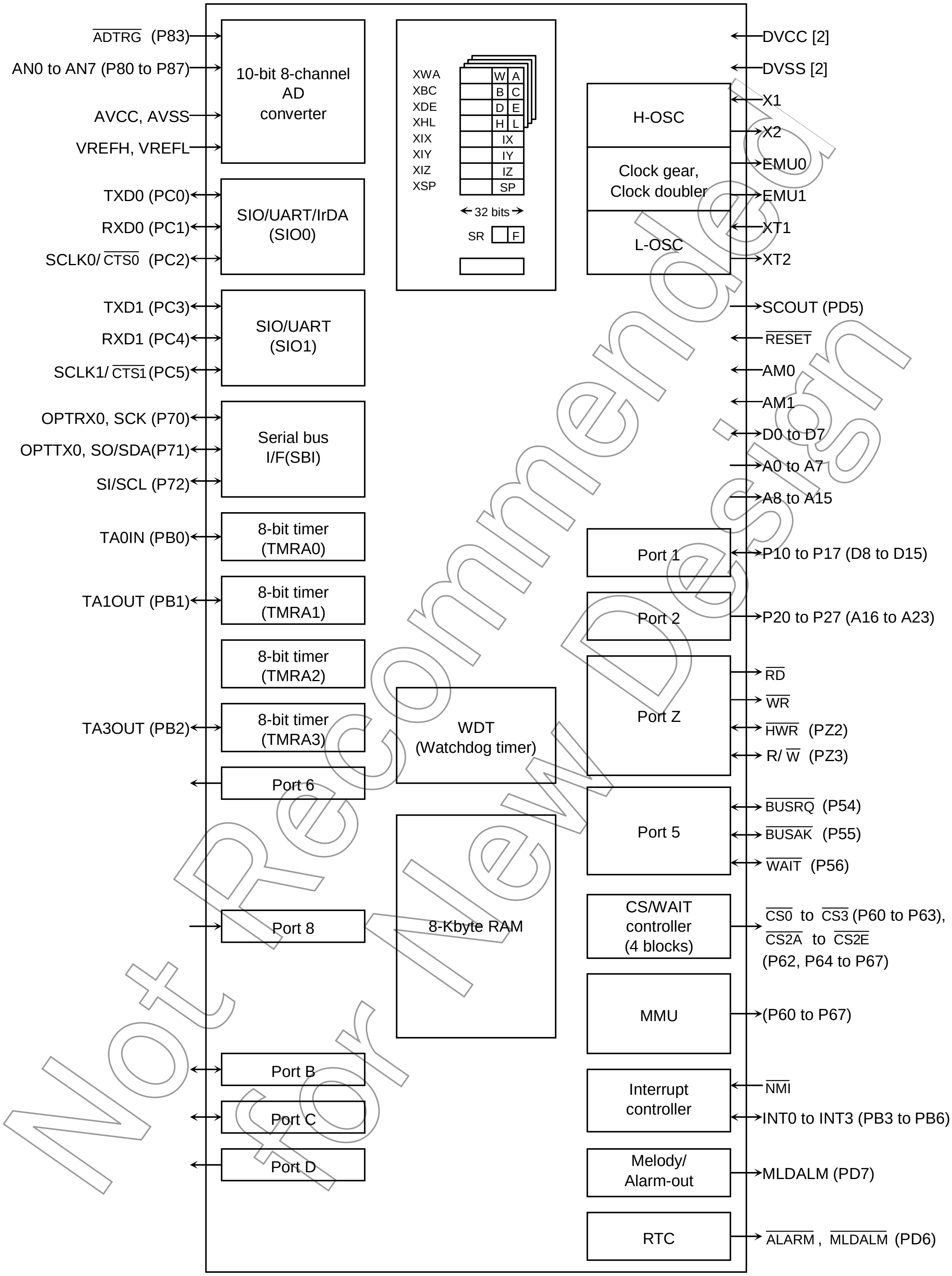
(17) Triple-clock controller
- Clock doubler (DFM) circuit is inside
- Clock gear function: Select a high-frequency clock fc/1 to fc/16
- Slow mode (fs = 32.768 kHz)

(18) Operating voltage
- V_{CC} = 2.7 V to 3.6 V (fc max = 33 MHz)
- V_{CC} = 1.8 V to 3.6 V (fc max = 10 MHz)

(19) Package
- 100-pin QFP: LQFP100-P-1414-0.50F
- Chip form supply also available. For details, contact your local Toshiba sales representative.

Not Recommended for New Design

Downloaded from Arrow.com.



( ): Initial function after reset

Figure 1.1 TMP91C824 Block Diagram

Downloaded from Arrow.com.

2. Pin Assignment and Functions

The assignment of input/output pins for the TMP91C824, their names and functions are as follows:

2.1 Pin Assignment Diagram

Figure 2.1 shows the pin assignment of the TMP91C824FG.

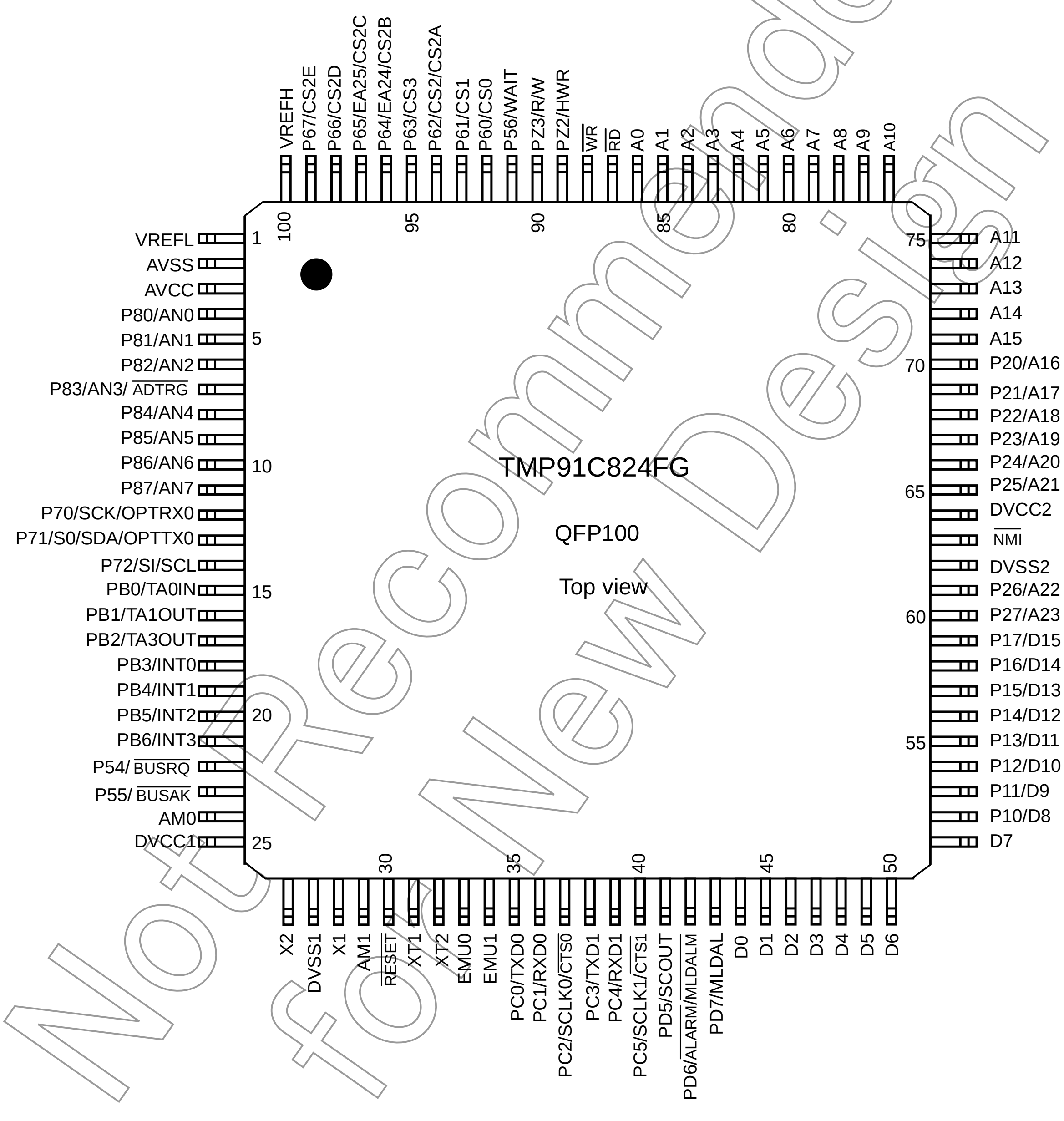


Figure 2.1 Pin Assignment Diagram (100-pin QFP)

Not Recommended for New Design

Downloaded from Arrow.com.

2.2 Pad Layout

(Chip size 4.37 mm × 4.37 mm) Unit: μm

Pin No.	Name	X Point	Y Point	Pin No.	Name	X Point	Y Point	Pin No.	Name	X Point	Y Point
1	VREFL	–2050	1721	35	PC0	–140	–2050	69	P21	2045	939
2	AVSS	–2050	1596	36	PC1	–14	–2050	70	P20	2045	1075
3	AVCC	–2050	1470	37	PC2	112	–2050	71	A15	2045	1207
4	P80	–2050	1337	38	PC3	238	–2050	72	A14	2045	1337
5	P81	–2050	1209	39	PC4	365	–2050	73	A13	2045	1464
6	P82	–2050	1076	40	PC5	491	–2050	74	A12	2045	1592
7	P83	–2050	943	41	PD5	618	–2050	75	A11	2045	1721
8	P84	–2050	810	42	PD6	744	–2050	76	A10	1720	2045
9	P85	–2050	677	43	PD7	871	–2050	77	A9	1591	2045
10	P86	–2050	544	44	D0	998	–2050	78	A8	1464	2045
11	P87	–2050	416	45	D1	1124	–2050	79	A7	1337	2045
12	P70	–2050	148	46	D2	1251	–2050	80	A6	1197	2045
13	P71	–2050	15	47	D3	1377	–2050	81	A5	1058	2045
14	P72	–2050	–118	48	D4	1504	–2050	82	A4	918	2045
15	PB0	–2050	–251	49	D5	1630	–2050	83	A3	778	2045
16	PB1	–2050	–384	50	D6	1757	–2050	84	A2	639	2045
17	PB2	–2050	–517	51	D7	2045	–1750	85	A1	499	2045
18	PB3	–2050	–650	52	P10	2045	–1614	86	A0	359	2045
19	PB4	–2050	–783	53	P11	2045	–1478	87	$\overline{RD}$	219	2045
20	PB5	–2050	–916	54	P12	2045	–1341	88	$\overline{WR}$	80	2045
21	PB6	–2050	–1049	55	P13	2045	–1205	89	PZ2	–59	2045
22	P54	–2050	–1182	56	P14	2045	–1069	90	PZ3	–199	2045
23	P55	–2050	–1315	57	P15	2045	–933	91	P56	–338	2045
24	AM0	–2050	–1448	58	P16	2045	–796	92	P60	–478	2045
25	VCC	–2050	–1581	59	P17	2045	–660	93	P61	–618	2045
26	X2	–1551	–2050	60	P27	2045	–524	94	P62	–757	2045
27	VSS	–1330	–2050	61	P26	2045	–388	95	P63	–897	2045
28	X1	–1205	–2050	62	VSS	2045	–234	96	P64	–1037	2045
29	AM1	–1075	–2050	63	$\overline{NMI}$	2045	–80	97	P65	–1176	2045
30	$\overline{RESET}$	–948	–2050	64	VCC	2045	240	98	P66	–1316	2045
31	XT1	–822	–2050	65	P25	2045	394	99	P67	–1456	2045
32	XT2	–520	–2050	66	P24	2045	530	100	VREFH	–1725	2045
33	EMU0	–394	–2050	67	P23	2045	666				
34	EMU1	–267	–2050	68	P22	2045	803				

Not Recommended for New Design

Downloaded from Arrow.com.

2.3 Pin Names and Functions

The names of the input/output pins and their functions are described below.

Table 2.3.1 Pin Names and Functions (1/3)

Pin Name	Number of Pins	I/O	Functions
D0 to D7	8	I/O	Data (Lower): Bits 0 to 7 of data bus
P10 to P17 D8 to D15	8	I/O I/O	Port 1: I/O port that allows I/O to be selected at the bit level (when used to the external 8-bit bus) Data (Upper): bits 8 to15 of data bus
P20 to P27 A16 to A23	8	Output Output	Port 2: Output port Address: Bits 16 to 23 of address bus
A8 to A15	8	Output	Address: Bits 8 to 15 of address bus
A0 to A7	8	Output	Address: Bits 0 to 7 of address bus
$\overline{RD}$	1	Output	Read: Strobe signal for reading external memory RD is outputted by setting PZ<RDE> to “0” even when read internal memory.
$\overline{WR}$	1	Output	Write: Strobe signal for writing data to pins D0 to D7
PZ2 $\overline{HWR}$	1	I/O Output	Port Z2: I/O port (with pull-up resistor) High write: Strobe signal for writing data to pins D8 to D15
PZ3 R/ $\overline{W}$	1	I/O Output	Port Z3: I/O port (with pull-up resistor) Read/write: 1 represents read or dummy cycle; 0 represents write cycle.
P54 $\overline{BUSRQ}$	1	I/O Output	Port 54: I/O port (with pull-up resistor) Bus request: High-impedance used to request bus release
P55 $\overline{BUSAK}$	1	I/O Output	Port 55: I/O port (with pull-up resistor) Bus acknowledge: Signal used to acknowledge bus release
P56 $\overline{WAIT}$	1	I/O Input	Port 56: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait ((1 + N) wait states)
P60 $\overline{CS0}$	1	Output Output	Port 60: Output port Chip select 0: Outputs 0 when address is within specified address area.
P61 $\overline{CS1}$	1	Output Output	Port 61: Output port Chip select 1: Outputs 0 when address is within specified address area
P62 $\overline{CS2}$ $\overline{CS2A}$	1	Output Output Output	Port 62: Output port Chip select 2: Outputs 0 when address is within specified address area Expand chip select 2A: Outputs 0 when address is within specified address area
P63 $\overline{CS3}$	1	Output Output	Port 63: Output port Chip select 3: Outputs 0 when address is within specified address area
P64 EA24 $\overline{CS2B}$	1	Output Output Output	Port 64: Output port Address 24: Expand address Expand chip select 2B: Outputs 0 when address is within specified address area
P65 EA25 $\overline{CS2C}$	1	Output Output Output	Port 65: Output port Address 25: Expand address Expand chip select 2C: Outputs 0 when address is within specified address area
P66 $\overline{CS2D}$	1	Output Output	Port 66: Output port Expand chip select 2D: Outputs 0 when address is within specified address area
P67 $\overline{CS2E}$	1	Output Output	Port 67: Outpt port Expand chip select 2E: Outputs 0 when address is within specified address area

Not Recommended for New Design

Downloaded from Arrow.com.

Table 2.3.2 Pin Names and Functions (2/3)

Pin Name	Number of Pins	I/O	Functions
P70 SCK OPTRX0	1	I/O I/O Input	Port 70: I/O port Serial bus interface clock I/O data at SIO mode Serial 0 receive data
P71 SO SDA OPTTX0	1	I/O Output I/O Output	Port 71: I/O port Serial bus interface send data at SIO mode Serial bus interface send/receive data at I^2C bus mode Open-drain output mode by programmable (with pull up) Serial 0 send data
P72 SI SCL	1	I/O Input I/O	Port 72: I/O port Serial bus interface recive data at SIO mode Serial bus interface clock I/O data at I^2C bus mode Open-drain output mode by programmable (with pull up)
P80 to P87 AN0 to AN7 $\overline{\text{ADTRG}}$	8	Input Input Input	Port 80 to 87 port: Pin used to input ports Analog input 0 to 7: Pin used to input to AD conveter AD trigger: Signal used to request AD start (with used to P83)
PB0 TA0IN	1	I/O Input	Port B0: I/O port 8-bit timer 0 input: Timer 0 input
PB1 TA1OUT	1	I/O Output	Port B1: I/O port 8-bit timer 1 output: Timer 0 output or timer 1 output
PB2 TA3OUT	1	I/O Output	Port B2: I/O port 8-bit timer 3 output: Timer 2 output or timer 3 output
PB3 INT0	1	I/O Input	Port B0: I/O port Interrupt request pin0: Interrupt request pin with programmable rising /falling edge
PB4 to PB6 INT1 to INT3	3	I/O Input	Port B4 to B6: I/O port Interrupt request pin1 to 3: Interrupt request pin with programmable rising /falling edge
PC0 TXD0	1	I/O Output	Port C0: I/O port Serial 0 send data: Open-drain output pin by programmable
PC1 RXD0	1	I/O Input	Port C1: I/O port Serial 0 receive data

Not Recommended for New Design

Downloaded from Arrow.com.

Table 2.3.3 Pin Names and Functions (3/3)

Pin Name	Number of Pins	I/O	Functions
PC2 SCLK0 $\overline{\text{CTS0}}$	1	I/O I/O Input	Port C2: I/O port Serial 0 clock Serial data send enable 0 (Clear to send)
PC3 TXD1	1	I/O Output	Port C3: I/O port Serial 1 send data Open-drain output pin by programmable
PC4 RXD1	1	I/O Input	Port C4: I/O port Serial 1 receive data
PC5 SCLK1 $\overline{\text{CTS1}}$	1	I/O I/O Input	Port C5: I/O port Serial clock I/O 1 Serial 1 data send enable (Clear to send)
XT1	1	Input	Low-frequency oscillator connecting pin
XT2	1	Output	Low-frequency oscillator connecting pin
PD5 SCOUT	1	Output Output	Port D5: Output port System clock output: f_{SYS} or fs output
PD6 $\overline{\text{ALARM}}$ $\overline{\text{MLDALM}}$	1	Output Output Output	Port D6: Output port RTC alarm output pin
PD7 MLDALM	1	Output Output	Port D7: Output port Melody/alarm output pin
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin: Interrupt request pin with programmable falling edge level or with both edge levels programmable
AM0 to AM1	2	Input	Operation mode: Fixed to AM1 = 0, AM0 = 1 16-bit external bus or 8-/16-bit dynamic sizing. Fixed to AM1 = 0, AM0 = 0 8-bit external bus fixed.
EMU0	1	Output	Open pin
EMU1	1	Output	Open pin
$\overline{\text{RESET}}$	1	Input	Reset: Initializes TMP91C824 (with pull-up resistor).
VREFH	1	Input	Pin for reference voltage input to AD converter (H)
VREFL	1	Input	Pin for reference voltage input to AD converter (L)
AVCC	1		Power supply pin for AD converter
AVSS	1		GND pin for AD converter (0 V)
X1/X2	2		High-frequency oscillator connection pins
DVCC	2		Power supply pins (All VCC pins should be connecyed with the power Supply pin.)
DVSS	2		GND pins (0 V) (All pins shuold be connected with GND (0 V).)

Not Recommended for New Design

Downloaded from Arrow.com.

3. Operation

This following describes block by block the functions and operation of the TMP91C824. Notes and restrictions for eatch book are outlined in 6 “Precautions and Restrictions” at the end of this manual.

3.1 CPU

The TMP91C824 incorporates a high-performance 16-bit CPU (the 900/L1 CPU). For CPU operation, see the TLCS-900/L1 CPU.

The following describe the unique function of the CPU used in the TMP91C824; these functions are not covered in the TLCS-900/L1 CPU section.

3.1.1 Reset

When resetting the TMP91C824 microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then set the $\overline{\text{RESET}}$ input to low level at least for 10 system clocks (10 μs at 33 MHz). Thus, when turn on the switch, be set to the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the $\overline{\text{RESET}}$ input to low level at least for 10 system clocks.

Clock gear is initialized 1/16 mode by reset operation. It means that the system clock mode f_{SYS} is set to fc/32 (= fc/16 × 1/2).

When the reset is accept, the CPU:

- Sets as follows the program counter (PC) in accordance with the reset vector stored at address FFFF00H to FFFF02H:

  PC<0:7> ← Value at FFFF00H address

  PC<15:8> ← Value at FFFF01H address

  PC<23:16> ← Value at FFFF02H address

- Sets the stack pointer (XSP) to 100H.
- Sets bits <IFF2:0> of the status register (SR) to 111. (Sets the interrupt level mask register to level 7.)
- Sets the <MAX> bit of the status register (SR) to 1 (MAX mode).
  (Note: As this product does not support MIN mode, do not write a 0 to the <MAX>)
- Clears bits <RFP2:0> of the status register (SR) to 000. (Sets the register bank to 0.)

  When reset is released,the CPU starts executing instructions in accordance with the program counter settings. CPU internal registers not mentioned above do not change when the reset is released.

  When the reset is accepted, the CPU sets internal I/O, ports, and other pins as follows.

- Initializes the internal I/O registers.
- Sets the port pins, including the pins that also act as internal I/O, to general-purpose input or output port mode.

Note: The CPU internal register (except to PC, SR, XSP) and internal RAM data do not change by resetting.

Figure 3.1.1 is a reset timing chart of the TMP91C824.

Not Recommended for New Design

Downloaded from Arrow.com.

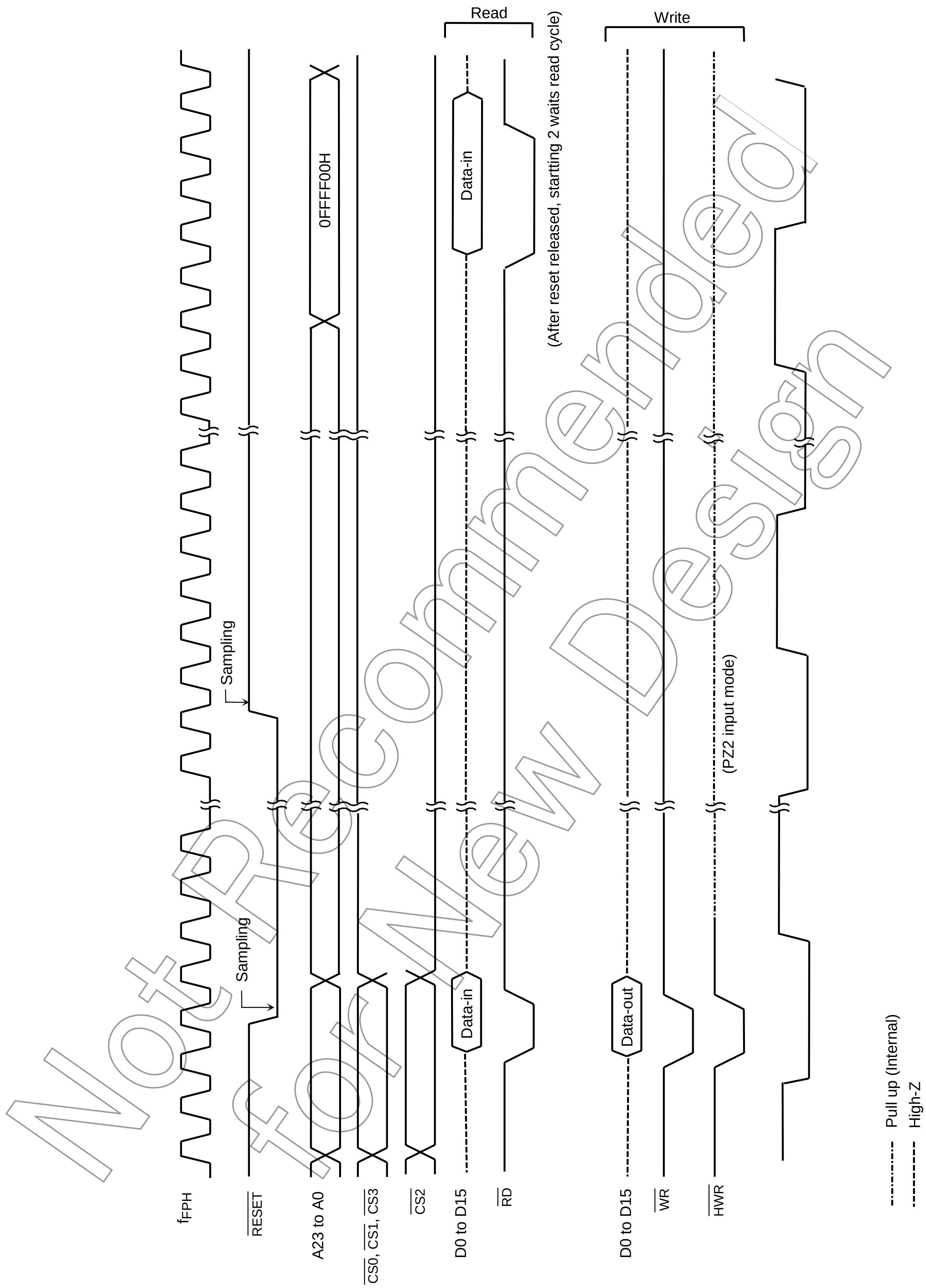


Figure 3.1.1 TMP91C824 Reset Timing Chart

Not Recommended for New Design

Downloaded from Arrow.com.

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP91C824.

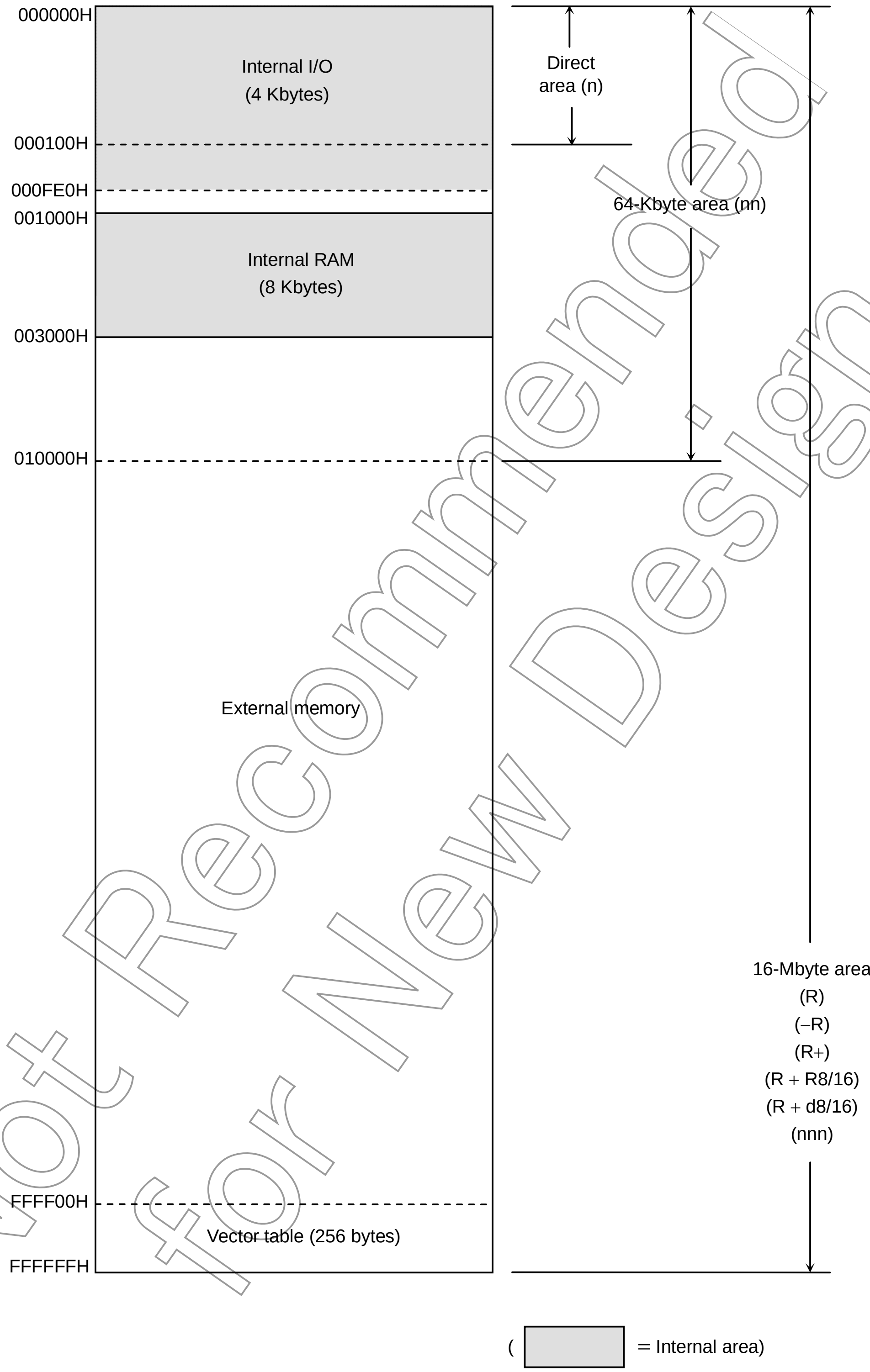


Figure 3.2.1 Memory Map

Note: Address 000FE0H to 00FFFH is assigned for the TOSHIBA reserve area, user can't use.

Not Recommended for New Design

Downloaded from Arrow.com.

3.3 Triple Clock Function and Standby Function

TMP91C824 contains (1) clock gear, (2) clock doubler (DFM), (3) standby controller and (4) noise-reduction circuit. It is used for low-power and low-noise systems.

This chapter is organized as follows:

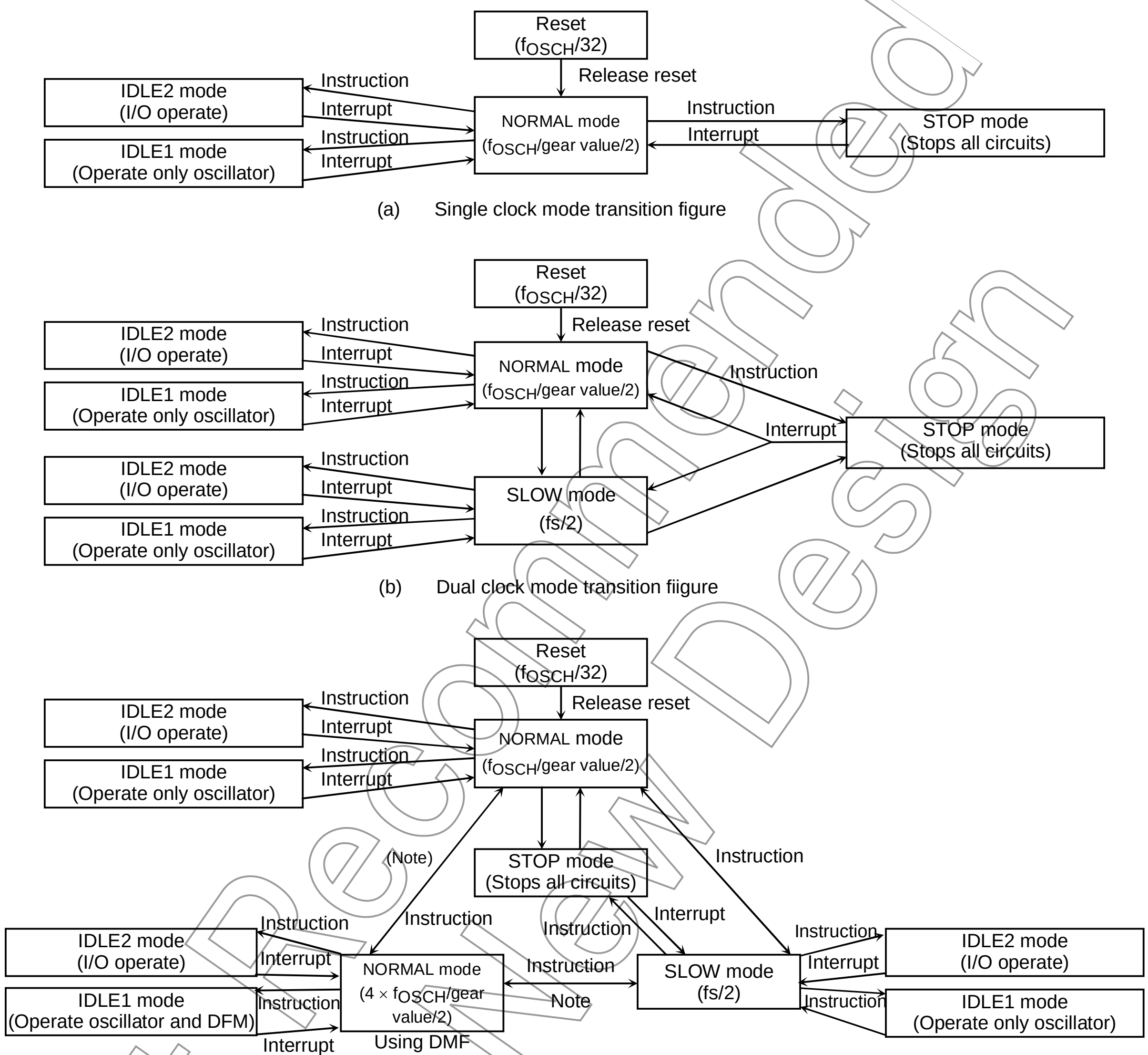
3.3.1 Block Diagram of System Clock
3.3.2 SFR
3.3.3 System Clock Controller
3.3.4 Prescaler Clock Controller
3.3.5 Clock Doubler (DFM)
3.3.6 Noise Reduction Circuits
3.3.7 Standby Controller

Not Recommended for New Design

Downloaded from Arrow.com.

The clock operating modes are as follows: (a) Single clock mode (X1, X2 pins only), (b) Dual clock mode (X1, X2, XT1 and XT2 pins) and (c) Triple clock mode (The X1, X2, XT1 and XT2 pins and DFM).

Figure 3.3.1 shows a transition figure.



(c) Triple clock mode trasision Figure

Note 1: It's prohiibited to control DFM in SLOW mode when shifting from SLOW mode to NORMAL mode with use of DFM. (DFM start up/stop/change write to DFMCR0<ACT1:0> register)

Note 2: If you shift from NORMAL mode with use of DFM to NORMAL mode, the instructions should be separated into two procedures as below. Change CPU clock → Stop DFM circuit.

Note 3: It's prohibited to shift from NORMAL mode with use of DFM to STOP mode directly. You should set NORMAL mode once, and then shift to STOP mode. (You should stop high-frequency oscillator after you stop DFM.)

Figure 3.3.1 System Clock Block Diagram

Note: The clock frequency input from the X1 and X2 pins is called f_{OSCH} and the clock frequency input from the XT1 and XT2 pins is called fs. The clock frequency selected by SYSCR1<SYSCK> is called the system clock f_{FPH}. The system clock f_{SYS} is defined as the divided clock of f_{FPH}, and one cycle of f_{SYS} is called one state.

Not Recommended for New Design
Downloaded from Arrow.com.

3.3.1 Block Diagram of System Clock

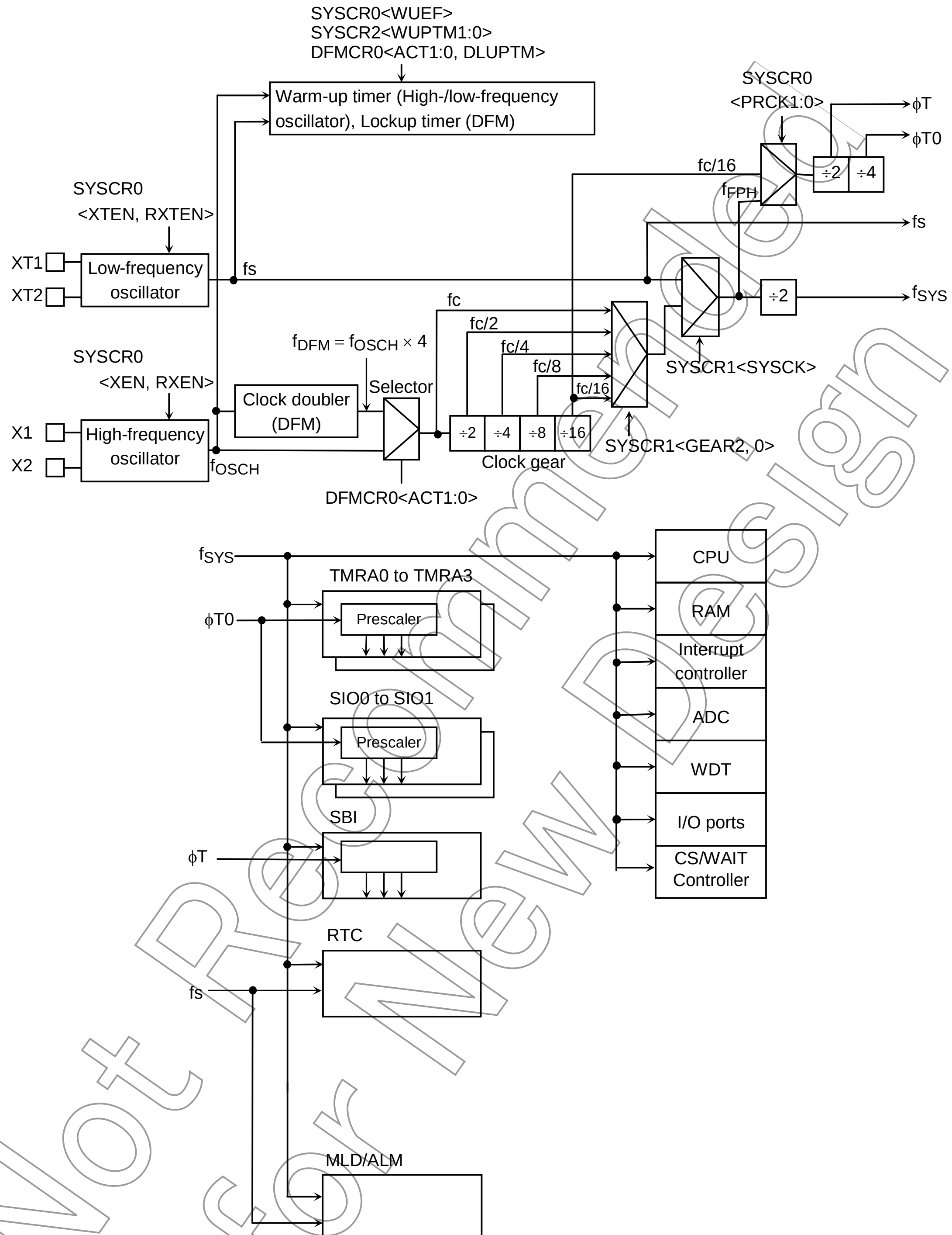


Figure 3.3.2 Block Diagram of System Clock

Not Recommended for New Design

Downloaded from Arrow.com.

3.3.2 SFR

		7	6	5	4	3	2	1	0
SYSCR0 (00E0H)	Bit symbol	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
	Read/Write	R/W							
	After reset	1	1	1	0	0	0	0	0
	Function	High-frequency oscillator (fc) 0: Stop 1: Oscillation	Low-frequency oscillator (fs) 0: Stop 1: Oscillation (Note 1)	High-frequency oscillator (fc) after release of STOP mode 0: Stop 1: Oscillation	Low-frequency oscillator (fs) after release of STOP mode 0: Stop 1: Oscillation	Selects clock after release of STOP mode 0: fc 1: fs	Warm-up timer 0: Write Don't care 1: Write start timer 0: Read end warm up 1: Read do not end warm up	Select prescaler clock 00: f_{FPH} (Note 2) 01: Reserved 10: fc/16 11: Reserved	

		7	6	5	4	3	2	1	0
SYSCR1 (00E1H)	Bit symbol					SYSCK	GEAR2	GEAR1	GEAR0
	Read/Write					R/W			
	After reset					0	1	0	0
	Function					Select system clock 0: fc 1: fs	Select gear value of high frequency (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (Reserved) 110: (Reserved) 111: (Reserved)		

		7	6	5	4	3	2	1	0
SYSCR2 (00E2H)	Bit symbol		SCOSEL	WUPTM1	WUPTM0	HALTM1	HALTM0	SELDRV	DRVE
	Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset		0	1	0	1	1	0	0
	Function		0: fs 1: f_{SYS}	Warm-up timer 00: Reserved 01: 2^8 inputted frequency 10: 2^{14} 11: 2^{16}		HALT mode 00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		<DRVE> mode select 0: STOP 1: IDLE1	Pin state control in STOP/IDLE1 mode 0: I/O off 1: Remains the state before halt

Note 1: By reset, low-frequency oscillator is enable.

Note 2: In case of using built-in SBI circuit, it must set SYSCR0<PRCK1:0> to 00.

Figure 3.3.3 SFR for System Clock

Not Recommended for New Design

Downloaded from Arrow.com.

		7	6	5	4	3	2	1	0
DFMCR0 (00E8H)	Bit symbol	ACT1	ACT0	DLUPFG	DLUPTM				
	Read/Write	R/W	R/W	R	R/W				
	After reset	0	0	0	0				
	Function	DFM LUP select f_{FPH} 00 STOP STOP f_{OSCH} 01 RUN RUN f_{OSCH} 10 RUN STOP f_{DFM} 11 RUN STOP f_{OSCH}		Lockup status flag 0: LUP end 1: LUP not end	Lockup time 0: $2^{12}/f_{OSCH}$ 1: $2^{10}/f_{OSCH}$				
DFMCR1 (00E9H)	Bit symbol	D7	D6	D5	D4	D3	D2	D1	D0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	1	0	0	1	1
	Function	DFM revision Input frequency 4 to 8.25 MHz (at 2.7 V to 3.6 V): write “0BH” Input frequency 2 to 2.5 MHz (at 2.0 V ± 10%): write “1BH”							

Figure 3.3.4 SFR for DFM

Limitation point on the use of DFM

1. It’s prohibited to execute DFM enable/disable control in the SLOW mode (fs). (Write to DFMCR0<ACT1:0> = “10”).
   You should control DFM in the NORMAL mode.

2. If you stop DFM operation during using DFM (DFMCR0<ACT1:0> = “10”), you shouldn’t execute that change the clock f_{DFM} to f_{OSCH} and stop the DFM at the same time. Therefore the above execution should be separated into two procedures as showing below.

```
LD    (DFMCR0), C0H    ;  Change the clock fDFM to fOSCH
LD    (DFMCR0), 00H    ;  DFM stop
```

3. If you stop high-frequency oscillator during using DFM (DFMCR0<ACT1:0> = “10”), you should stop DFM before you stop high-frequency oscillator.

Please refer to 3.3.5 “Clock Doubler (DFM)” for the details.

Not Recommended for New Design

Downloaded from Arrow.com.

		7	6	5	4	3	2	1	0
EMCCR0 (00E3H)	Bit symbol	PROTECT	–	–	–	–	EXTIN	DRVOSCH	DRVOSCL
	Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	1	0	0	0	1	1
	Function	Protect flag 0: OFF 1: ON	Always write 0	Always write 1	Always write 0	Always write 0	1: External clock	fc oscillator driver ability 1: Normal 0: Weak	fs oscillator driver ability 1: Normal 0: Weak
EMCCR1 (00E4H)	Bit symbol	Switching the protect ON/OFF by write to following 1st-KEY,2nd-KEY 1st-KEY: EMCCR1 = 5AH, EMCCR2 = A5H in succession write 2nd-KEY: EMCCR1 = A5H, EMCCR2 = 5AH in succession write							
	Read/Write								
	After reset								
	Function								
EMCCR2 (00E5H)	Bit symbol								
	Read/Write								
	After reset								
	Function								
EMCCR3 (00E6H)	Bit symbol		ENFROM	ENDROM	ENPROM		FFLAG	DFLAG	PFLAG
	Read/Write		R/W	R/W	R/W		R/W	R/W	R/W
	After reset		0	0	0		0	0	0
	Function		CS1A area detect control 0: Disable 1: Enable	CS2B-2G area detect control 0: Disable 1: Enable	CS2A area detect control 0: Disable 1: Enable		CS1A write operation flag	CS2B-2G write operation flag	CS2A write operation flag
							When reading 0: Not written 1: Written When writing 0: Clear flag		

Note1: In case restarting the oscillator in the stop oscillation state (e.g. Restart the oscillator in STOP mode), set EMCCR0<DRVOSCH>, <DRVOSCL>="1".

Note2: In case of Vcc = 2 V ± 10% use, fixed to EMCCR0<DRVOSCH> = 1.

Figure 3.3.5 SFR for Noise Reduction

Not Recommended for New Design

Downloaded from Arrow.com.

3.3.3 System Clock Controller

The system clock controller generates the system clock signal (f_{SYS}) for the CPU core and internal I/O. It contains two oscillation circuits and a clock gear circuit for high-frequency (fc) operation. The register SYSCR1<SYSCK> changes the system clock to either fc or fs, SYSCR0<XEN> and SYSCR0<XTEN> control enabling and disabling of each oscillator, and SYSCR1<GEAR0:2> sets the high-frequency clock gear to either 1, 2, 4, 8 or 16 (fc, fc/2, fc/4, fc/8 or fc/16). These functions can reduce the power consumption of the equipment in which the device is installed.

The combination of settings <XEN> = 1, <XTEN> = 0, <SYSCK> = 0 and <GEAR0:2> = 100 will cause the system clock (f_{SYS}) to be set to fc/32 (fc/16 × 1/2) after a reset.

For example, f_{SYS} is set to 1.03 MHz when the 33-MHz oscillator is connected to the X1 and X2 pins.

(1) Switching from NORMAL mode to SLOW mode

When the resonator is connected to the X1 and X2 pins, or to the XT1 and XT2 pins, the warm-up timer can be used to change the operation frequency after stable oscillation has been attained.

The warm-up time can be selected using SYSCR2<WUPTM0:1>.

This warm-up timer can be programmed to start and stop as shown in the following examples 1 and 2.

Table 3.3.1 shows the warm-up time.

Note 1: When using an oscillator (other than a resonator) with stable oscillation, a warm-up timer is not needed.

Note 2: The warm-up timer is operated by an oscillation clock. Hence, there may be some variation in warm-up time.

Table 3.3.1 Warm-up Times

Warm-up Time SYSCR2 <WUPTM1:0>	Change to NORMAL Mode	Change to SLOW Mode
01 (2^8/frequency)	8 (μs)	7.8 (ms)
10 (2^{14}/frequency)	0.496 (ms)	500 (ms)
11 (2^{16}/frequency)	1.986 (ms)	2000 (ms)

at f_{OSCH} = 33 MHz,
fs = 32.768 kHz

Not Recommended for New Design

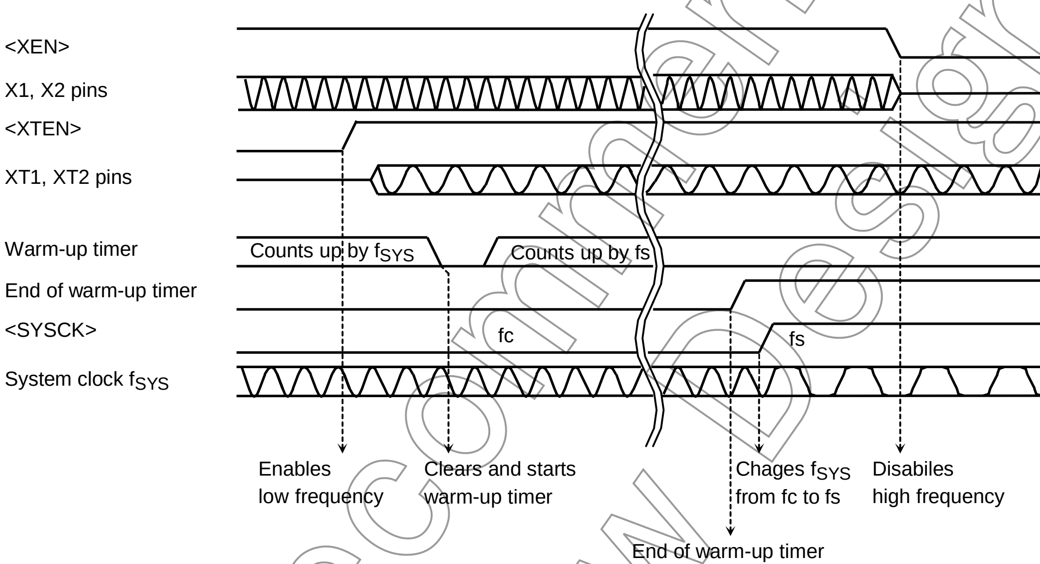
Downloaded from Arrow.com.

Example 1: Setting the clock

Changing from high frequency (fc) to low frequency (fs).

```
SYSCR0  EQU   00E0H
SYSCR1  EQU   00E1H
SYSCR2  EQU   00E2H
WDMOD   EQU   005CH
        LD    (SYSCR2), X–11– – – –B  ;  Sets warm-up time to 2^16/fs.
        SET   6, (SYSCR0)             ;  Enables low-frequency oscillation.
        SET   2, (SYSCR0)             ;  Clears and starts warm-up timer.
WUP:    BIT   2, (SYSCR0)             ; } Detects stopping of warm-up timer.
        JR    NZ, WUP                 ; }
        SET   3, (SYSCR1)             ;  Changes fSYS from fc to fs.
        RES   7, (SYSCR0)             ;  Disables high-frequency oscillation.
```

X: Don’t care, –: No change



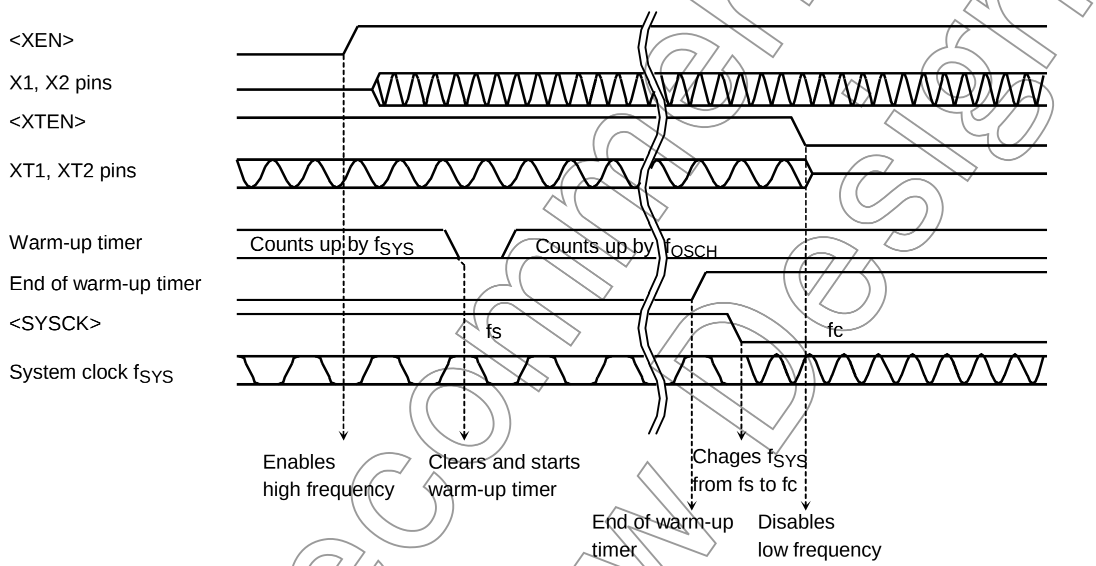
Not Recommended for New Design

Downloaded from Arrow.com.

Example 2: Setting the clock

Changing from low frequency (fs) to high frequency (fc).

```
SYSCR0   EQU    00E0H
SYSCR1   EQU    00E1H
SYSCR2   EQU    00E2H
         LD     (SYSCR2), X–10– – – –B   ;   Sets warm-up time to 214/fc.
         SET    7, (SYSCR0)              ;   Enables high-frequency oscillation.
         SET    2, (SYSCR0)              ;   Clears and starts warm-up timer.
WUP:     BIT    2, (SYSCR0)              ; }
         JR     NZ, WUP                  ; }  Detects stopping of warm-up timer.
         RES    3, (SYSCR1)              ;   Changes fSYS from fs to fc.
         RES    6, (SYSCR0)              ;   Disables low-frequency oscillation.
X: Don't care, –: No change
```



Not Recommended for New Design

Downloaded from Arrow.com.

(2) Clock gear controller

When the high-frequency clock fc is selected by setting SYSCR1<SYSCK> = 0, f_{FPH} is set according to the contents of the clock gear select register SYSCR1<GEAR0:2> to either fc, fc/2, fc/4, fc/8 or fc/16. Using the clock gear to select a lower value of f_{FPH} reduces power consumption.

Example 3:

Changing to a high-frequency gear

```
SYSCR1    EQU    00E1H
          LD     (SYSCR1),XXXX0000B     ;  Changes fSYS to fc/2.
X: Don't care
```

(High-speed clock gear changing)

To change the clock gear, write the register value to the SYSCR1<GEAR2:0> register. It is necessary the warm-up time until changing after writing the register value.

There is the possibility that the instruction next to the clock gear changing instruction is executed by the clock gear before changing. To execute the instruction next to the clock gear switching instruction by the clock gear after changing, input the dummy instruction as follows (Instruction to execute the write cycle).

```
Example:
SYSCR1    EQU    00E1H
          LD     (SYSCR1),XXXX0001B     ;  Changes fSYS to fc/4.
          LD     (DUMMY), 00H           ;  Dummy instruction
          Instruction to be executed after clock gear has changed
```

(3) Internal clock terminal out function

It can out internal clock (f_{SYS} or fs) from PD5/SCOUT.

PD5 pin function is set to SCOUT output by the following bit setting.

: PDFC<PD5F> = 1

Output clock select

: Refer to SYSCR2<SCOSEL> bit setting

<table>
<tr><th rowspan="2">HALT Mode
SCOUT Select</th><th rowspan="2">NORMAL
SLOW</th><th colspan="3">HALT Mode</th></tr>
<tr><th>IDLE2</th><th>IDLE1</th><th>STOP</th></tr>
<tr><td><SCOSEL> = 0</td><td colspan="3">fs clock out</td><td rowspan="2"></td></tr>
<tr><td><SCOSEL> = 1</td><td colspan="2">f_{SYS} clock out</td><td colspan="2">0 or 1 fix out</td></tr>
</table>

Not Recommended for New Design

Downloaded from Arrow.com.

3.3.4 Prescaler Clock Controller

For the internal I/O (TMRA01 to TMRA23, SIO0 to SIO1) there is a prescaler which can divide the clock.

The ϕT0 clock input to the prescaler is either the clock f_{FPH} divided by 4 or the clock fc/16 divided by 4. The setting of the SYSCR0<PRCK0:1> register determines which clock signal is input.

3.3.5 Clock Doubler (DFM)

DFM outputs the f_{DFM} clock signal, which is four times as fast as f_{OSCH}. It can use the low-frequency oscillator, even though the internal clock is high frequency.

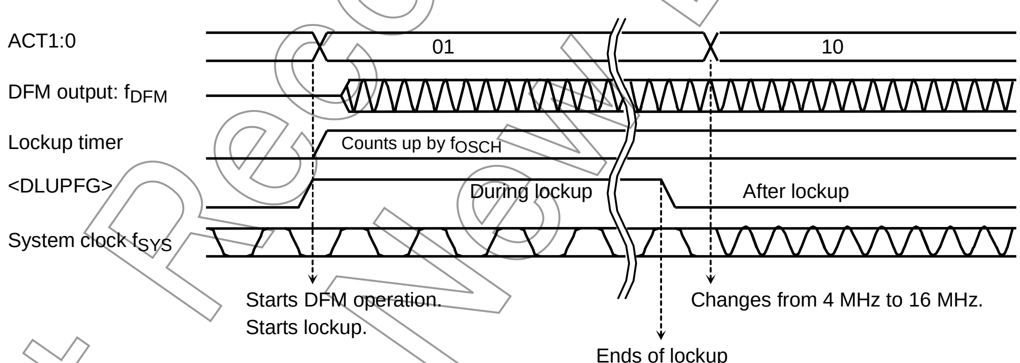
A reset initializes DFM to stop status, setting to DFMCR0 register is needed before use.

Like an oscillator, this circuit requires time to stabilize. This is called the lockup time.

The following example shows how DFM is used.

```
DFMCR0   EQU   00E8H
DFMCR1   EQU   00E9H
         LD    (DFMCR1), 00001011B      DFM parameter setting
         LD    (DFMCR0), 01−0XXXXB  ;   Set lockup time to 2^12/4 MHz
                                        Enables DFM operation and starts lockup
LUP:     BIT   5, (DFMCR0)          ;  } Detects end of lockup
         JR    NZ, LUP              ;  }
         LD    (DFMCR0), 10−0XXXXB  ;   Changes fc from 4 MHz to 16 MHz
                                        Changes fSYS from 2 MHz to 8 MHz
```

X: Don't care



Note: Input frequency limitation and correction for DFM

Recommend to use input frequency (High-speed oscillation) for DFM in the following condition.

f_{OSCH} = 4 to 8.25 MHz (Vcc = 2.7 V to 3.6 V): Write 0BH to DFMCR1

f_{OSCH} = 2 to 2.5 MHz (Vcc = 2.0 V ± 10%): Write 1BH to DFMCR1

Not Recommended for New Design

Downloaded from Arrow.com.

Limitation point on the use of DFM

1. it's prohibited to execute DFM enable/disable control in the SLOW mode (fs). You should control DFM in the NORMAL mode.
2. If you stop DFM operation during using DFM (DFMCR0<ACT1:0> = "10"), you shouldn't execute the commands that change the clock f_{DFM} to f_{OSCH} and stop the DFM at the same time. Therefore the above execution should be separated into two procedures as showing below.

```
      LD      (DFMCR0), C0H        ;  Change the clock fDFM to fOSCH
      LD      (DFMCR0), 00H        ;  DFM stop
```

3. If you stop high-frequency oscillator during using DFM (DFMCR0<ACT1:0> = "10"), you should stop DFM before you stop high-frequency oscillator.
4. More than 1 ms of interval time is required from stop of DFM to the next start up of DFM.

Examples of settings are below.

(1) Start up control

(OK) Low-frequency oscillator operation mode (fs) (High-frequency oscillator STOP) → High-frequency oscillator start up → High-frequency oscillator operation mode (f_{OSCH}) → DFM start up → DFM use mode (f_{DFM})

```
      LD      (SYSCR0),  11---1--B  ;  High-frequency oscillator start up/warm-up start
WUP:  BIT     2, (SYSCR0)           ;  } Check for the flag of warm-up end
      JR      NZ, WUP               ;  }
      LD      (SYSCR1),  ----0---B  ;  Change the system clock fs to fOSCH
      LD      (DFMCR0),  01-0----B  ;  DFM start up/lockup start
LUP:  BIT     5, (DFMCR0)           ;  } Check for the flag of lockup end
      JR      NZ, LUP               ;  }
      LD      (DFMCR0),  10-0----B  ;  Change the system clock fOSCH to fDFM
```

(OK) Low-frequency oscillator operation mode (fs) (High-frequency oscillator operate) → High-frequency oscillator operation mode (f_{OSCH}) → DFM start up → DFM use mode (f_{DFM})

```
      LD      (SYSCR1),  ----0---B  ;  Change the system clock fs to fOSCH
      LD      (DFMCR0),  01-0----B  ;  DFM start up/lockup start
LUP:  BIT     5, (DFMCR0)           ;  } Check for the flag of lockup end
      JR      NZ, LUP               ;  }
      LD      (DFMCR0),  10-0----B  ;  Change the system clock fOSCH to fDFM
```

(Error) Low-frequency oscillator operation mode (fs) (High-frequency oscillator STOP) → High-frequency oscillator start up → DFM start up → DFM use mode (f_{DFM})

```
      LD      (SYSCR0),  11---1--B  ;  High-frequency oscillator start up/warm-up start
WUP:  BIT     2, (SYSCR0)           ;  } Check for the flag of warm-up end
      JR      NZ, WUP               ;  }
      LD      (DFMCR0),  01-0----B  ;  DFM start up/lockup start
LUP:  BIT     5, (DFMCR0)           ;  } Check for the flag of lockup end
      JR      NZ, LUP               ;  }
      LD      (DFMCR0),  10-0----B  ;  Change the clock fOSCH to fDFM
      LD      (SYSCR1),  ----0---B  ;  Change the internal clock fs to fDFM
```

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Change/stop control

(OK) DFM use mode (f_{DFM}) → High-frequency oscillator operation mode (f_{OSCH}) → DFM stop → Low-frequency oscillator operation mode (fs) → High-frequency oscillator stop

```
LD    (DFMCR0),  11------B    ;  Change the system clock fDFM to fOSCH
LD    (DFMCR0),  00------B    ;  DFM stop
LD    (SYSCR1),  ----1---B    ;  Change the system clock fOSCH to fs
LD    (SYSCR0),  0-------B    ;  High-frequency oscillator stop
```

(Error) DFM use mode (f_{DFM}) → Low-frequency oscillator operation mode (fs) → DFM stop → High-frequency oscillator stop

```
LD    (SYSCR1),  ----1---B    ;  Change the system clock fDFM to fs
LD    (DFMCR0),  11------B    ;  Change the internal clock (fc) fDFM to fOSCH
LD    (DFMCR0),  00------B    ;  DFM stop
LD    (SYSCR0),  0-------B    ;  High-frequency oscillator stop
```

(OK) DFM use mode (f_{DFM}) → Set the STOP mode → High-frequency oscillator operation mode (f_{OSCH}) → DFM stop → HALT (High-frequency oscillator stop)

```
LD    (SYSCR2),  ----01--B    ;  Set the STOP mode
                                 (This command can execute before use of DFM)
LD    (DFMCR0),  11------B    ;  Change the system clock fDFM to fOSCH
LD    (DFMCR0),  00------B    ;  DFM stop
HALT                          ;  Shift to STOP mode
```

(Error) DFM use mode (f_{DFM}) → Set the STOP mode → HALT (High-frequency oscillator stop)

```
LD    (SYSCR2),  ----01--B    ;  Set the STOP mode
                                 (This command can execute before use of DFM)
HALT                          ;  Shift to STOP mode
```

Not Recommended for New Design

Downloaded from Arrow.com.

3.3.6 Noise Reduction Circuits

Noise reduction circuits are built in, allowing implementation of the following features.

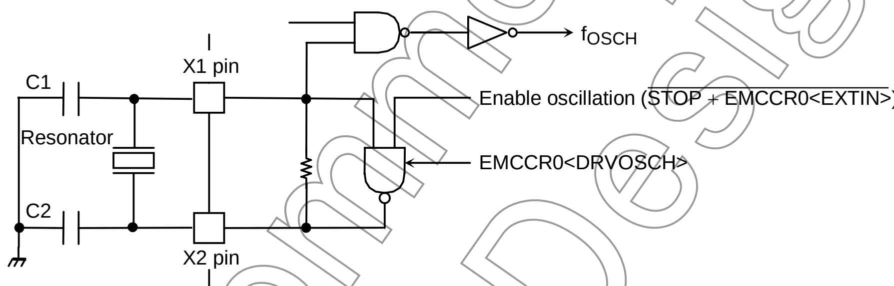
(1) Reduced drivability for high-frequency oscillator
(2) Reduced drivability for low-frequency oscillator
(3) Single drive for high-frequency oscillator
(4) SFR protection of register contents
(5) ROM protection of register contents

(1) Reduced drivability for high-frequency oscillator

(Purpose)

Reduces noise and power for oscillator when a resonator is used.

(Block diagram)



(Setting method)

The drivability of the oscillator is reduced by writing 0 to EMCCR0<DRVOSCH> register. By reset, <DRVOSCH> is initialized to 1 and the oscillator starts oscillation by normal-drivability when the power supply is on.

Don’t set <DRVOSCH> to 0 at Vcc = 2 V ± 10%.

Not Recommended for New Design

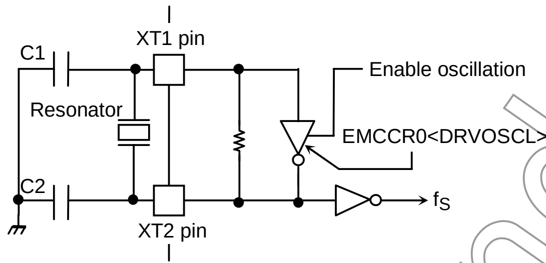
Downloaded from Arrow.com.

(2) Reduced drivability for low-frequency oscillator

(Purpose)

Reduces noise and power for oscillator when a resonator is used.

(Block diagram)



(Setting method)

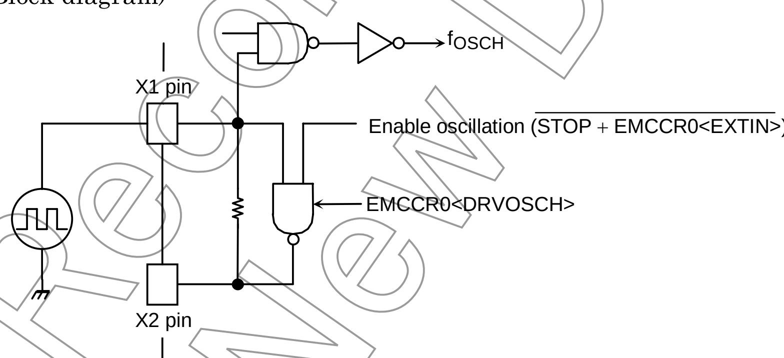
The drivability of the oscillator is reduced by writing 0 to the EMCCR0 <DRVOSCL> register. By reset, <DRVOSCL> is initialized to 1.

(3) Single drive for high-frequency oscillator

(Purpose)

Not need twin-drive and protect mistake operation by inputted noise to X2 pin when the external-oscillator is used.

(Block diagram)



(Setting method)

The oscillator is disabled and starts operation as buffer by writing 1 to EMCCR0<EXTIN> register. X2 pin is always outputted 1.

By reset, <EXTIN> is initialized to 0.

Note: Do not write EMCCR0<EXTIN> = “1” when using external resonator.

Not Recommended for New Design

Downloaded from Arrow.com.

(4) Runaway provision with SFR protection register

(Purpose)

Provision in runaway of program by noise mixing.

Write operation to specified SFR is prohibited so that provision program in runaway prevents that it is it in the state which is fetch impossibility by stopping of clock, memory control register (CS/WAIT controller, MMU) is changed.

And error handling in runaway becomes easy by INTP0 interruption.

Specified SFR list

1. CS/WAIT controller
   B0CS, B1CS, B2CS, B3CS, BEXCS,
   MSAR0, MSAR1, MSAR2, MSAR3,
   MAMR0, MAMR1, MAMR2, MAMR3
2. MMU
   LOCAL0/1/2/3
3. Clock gear (only EMCCR1, EMCCR2 can be written to)
   SYSCR0, SYSCR1, SYSCR2, EMCCR0, EMCCR3
4. DFM
   DFMCR0, DFMCR1

(Operation explanation)

Execute and release of protection (write operation to specified SFR) become possible by setting up a double key to EMCCR1 and EMCCR2 register.

(Double key)

1st-KEY: Succession writes in 5AH at EMCCR1 and A5H at EMCCR2

2nd-KEY: Succession writes in A5H at EMCCR1 and 5AH at EMCCR2

A state of protection can be confirmed by reading EMCCR0<PROTECT>.

By reset, protection becomes OFF.

And INTP0 interruption occurs when write operation to specified SFR was executed with protection ON state.

Not Recommended for New Design

Downloaded from Arrow.com.

(5) Runaway provision with ROM protection register

(Purpose)

Provision in runaway of program by noise mixing.

(Operation explanation)

When write operation was executed for external three kinds of ROM by runaway of program, INTP1 is occurred and detects runaway function.

Three kinds of ROM is fixed as for flash ROM (Option program ROM), data ROM, program ROM are as follows on the logical address memory map.

1. Flash ROM: Address 400000H to 7FFFFFH
2. Data ROM: Address 800000H to BFFFFFH
3. Program ROM: Address C00000H to FFFFFFH

For these address, admission/prohibition of detection of write operation sets it up with EMCCR3<ENFROM, ENDROM, ENPROM>. And INTP1 interruption occurred within which ROM can confirm each with EMCCR3<FFLAG, DFLAG, PFLAG>. This flag is cleared when write in 0.

Not Recommended for New Design

Downloaded from Arrow.com.

3.3.7 Standby Controller

(1) HALT modes

When the HALT instruction is executed, the operating mode switches to IDLE2, IDLE1 or STOP mode, depending on the contents of the SYSCR2<HALTM1:0> register.

The subsequent actions performed in each mode are as follows:

a. IDLE2: Only the CPU halts.
The internal I/O is available to select operation during IDLE2 mode. By setting the following register.
Table 3.3.2 shows the registers of setting operation during IDLE2 mode.

Table 3.3.2 SFR Seting Operation during IDLE2 Mode

Internal I/O	SFR
TMRA01	TA01RUN<I2TA01>
TMRA23	TA23RUN<I2TA23>
SIO0	SC0MOD1<I2S0>
SIO1	SC1MOD1<I2S1>
AD converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>
SBI	SBI0BR0<I2SBI0>

b. IDLE1: Only the oscillator and the RTC (Real time clock) and MLD continue to operate.

c. STOP: All internal circuits stop operating.

The operation of each of the different HALT modes is described in Table 3.3.3.

Table 3.3.3 I/O Operation during HALT Modes

<table>
<tr><th colspan="2">HALT Mode</th><th>IDLE2</th><th>IDLE1</th><th>STOP</th></tr>
<tr><td colspan="2">SYSCR2<HALTM1:0></td><td>11</td><td>10</td><td>01</td></tr>
<tr><td rowspan="7">Block</td><td>CPU</td><td colspan="3">Stop</td></tr>
<tr><td>I/O ports</td><td>Keep the state when the HALT instruction was executed.</td><td colspan="2">See Table 3.3.6, Table 3.3.7</td></tr>
<tr><td>TMRA</td><td rowspan="5">Available to select operation block</td><td colspan="2" rowspan="4">Stop</td></tr>
<tr><td>SIO, SBI</td></tr>
<tr><td>AD converter</td></tr>
<tr><td>WDT</td></tr>
<tr><td>RTC, MLD</td><td>Possible to operate</td><td></td></tr>
</table>

Not Recommended for New Design

Downloaded from Arrow.com.

(2) How to release the HALT mode

These halt states can be released by resetting or requesting an interrupt. The HALT release sources are determined by the combination between the states of interrupt mask register <IFF2:0> and the HALT modes. The details for releasing the halt status are shown in Table 3.3.4.

- Released by requesting an interrupt

The operating released from the HALT mode depends on the interrupt enabled status. When the interrupt request level set before executing the HALT instruction exceeds the value of interrupt mask register, the interrupt due to the source is processed after releasing the HALT mode, and CPU status executing an instruction that follows the HALT instruction. When the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the HALT mode is not executed. (In non-maskable interrupts, interrupt processing is processed after releasing the HALT mode regardless of the value of the mask register.) However only for INT0 to INT3 and INTRTC and INTALM interrupts, even if the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the the HALT mode is executed. In this case, interrupt processing, and CPU starts executing the instruction next to the HALT instruction, but the interrupt request flag is held at 1.

Note: Usually, interrupts can release all halts status. However, the interrupts ($\overline{NMI}$, INT0 to INT3, INTRTC, INTALM0 to INTALM4) which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)
If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficulty. The priority of this interrupt is compared with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

- Releasing by resetting

Releasing all halt status is executed by resetting.

When the STOP mode is released by reset, it is necessry enough resetting time (See Table 3.3.5) to set the operation of the oscillator to be stable.

When releasing the HALT mode by resetting, the internal RAM data keeps the state before the HALT instruction is executed. However the other settings contents are initialized. (Releasing due to interrupts keeps the state before the HALT instruction is executed.)

Not Recommended for New Design

Downloaded from Arrow.com.

Table 3.3.4 Source of Halt State Clearance and Halt Clearance Operation

Status of Received Interrupt			Interrupt Enabled (Interrupt level) ≥ (Interrupt mask)			Interrupt Disabled (Interrupt level) < (Interrupt mask)		
HALT Mode			IDLE2	IDLE1	STOP	IDLE2	IDLE1	STOP
Source of Halt State Clearance	Interrupt	NMI	♦	♦	♦ *1	–	–	–
		INTWDT	♦	×	×	–	–	–
		INT0 to INT3 (Note 1)	♦	♦	♦ *1	○	○	○ *1
		INTALM0 to INTALM4	♦	♦	×	○	○	×
		INTTA0 to INTTA3	♦	×	×	×	×	×
		INTRX0 to INTRX1, TX0 to TX1	♦	×	×	×	×	×
		INTAD	♦	×	×	×	×	×
		INTRTC	♦	♦	×	○	○	×
		INTSBI	♦	×	×	×	×	×
	RESET		Reset initializes the LSI					

♦: After clearing the HALT mode, CPU starts interrupt processing.

○: After clearing the HALT mode, CPU resumes executing starting from instruction following the HALT instruction.

×: It can not be used to release the HALT mode.

–: The priority level (Interrupt request level) of non-maskable interrupts is fixed to 7, the highest priority level. There is not this combination type.

*1: Releasing the HALT mode is executed after passing the warm-up time.

Note: When the HALT mode is cleared by an INT0 interrupt of the level mode in the interrupt enabled status, hold level “H” until starting interrupt processing. If level “L” is set before holding level “L”, interrupt processing is correctly started.

(Example: Clearing IDLE1 mode)

An INT0 interrupt clears the halt state when the device is in IDLE1 mode.

```
Address
8200H   LD     (PBFC), 08H       ; Sets PB3 to INT0.
8203H   LD     (IIMC), 00H       ; Selects INT0 interrupt rising edge.
8206H   LD     (INTE0AD), 06H    ; Sets INT0 interrupt level to 6.
8209H   EI     5                 ; Sets interrupt level to 5 for CPU.
820BH   LD     (SYSCR2), 28H     ; Sets HALT mode to IDLE1 mode.
820EH   HALT                     ; Halts CPU.

   INT0                          →  INT0 interrupt routine

                                    RETI
820FH   LD     XX, XX
```

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Operation

a. IDLE2 mode

In IDLE2 mode only specific internal I/O operations, as designated by the IDLE2 setting register, can take place. Instruction execution by the CPU stops.

Figure 3.3.6 illustrates an example of the timing for clearance of the IDLE2 mode halt state by an interrupt.

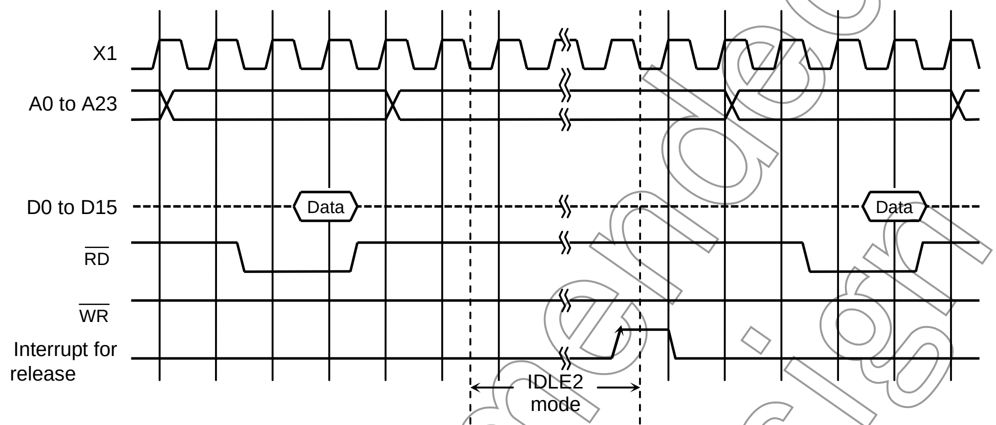


Figure 3.3.6 Timing Chart for IDLE2 Mode Halt State Cleared by Interrupt

b. IDLE1 mode

In IDLE1 mode, only the internal oscillator and the RTC, MLD continue to operate. The system clock in the MCU stops. The pin status in the IDLE1 mode is depended on setting the register SYSCR2<SELDRV, DRVE>. Table 3.3.6, Table 3.3.7 summarizes the state of these pins in the IDLE mode1.

In the halt state, the interrupt request is sampled asynchronously with the system clock; however, clearance of the halt state (e.g., restart of operation) is synchronous with it.

Figure 3.3.7 illustrates the timing for clearance of the IDLE1 mode halt state by an interrupt.

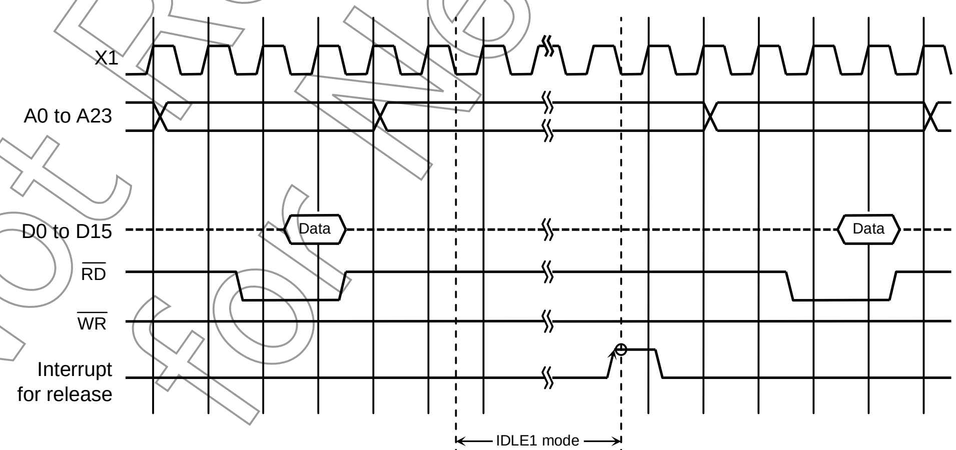


Figure 3.3.7 Timing Chart for IDLE1 Mode Halt State Cleared by Interrupt

Not Recommended for New Design

Downloaded from Arrow.com.

c. STOP mode

When STOP mode is selected, all internal circuits stop, including the internal oscillator pin status in STOP mode depends on the settings in the SYSCR2<DRVE> register. Table 3.3.6, Table 3.3.7 summarizes the state of these pins in STOP mode.

After STOP mode has been cleared system clock output starts when the warm-up time has elapsed, in order to allow oscillation to stabilize. After STOP mode has been cleared, either NORMAL mode or SLOW mode can be selected using the SYSCR0<RSYSCK> register. Therefore, <RSYSCK>, <RXEN> and <RXTEN> must be set See the sample warm-up times in Table 3.3.5.

Figure 3.3.8 illustrates the timing for clearance of the STOP mode halt state by an interrupt.

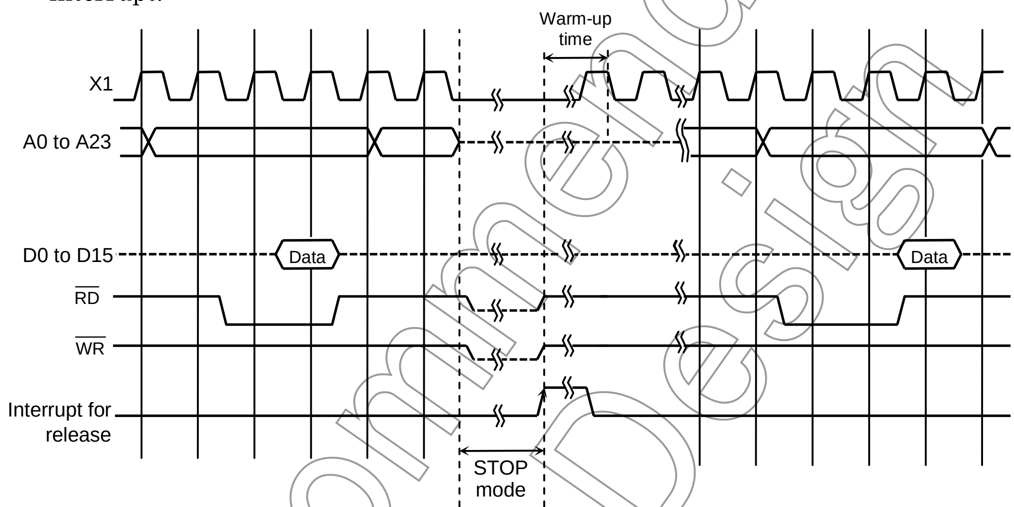


Figure 3.3.8 Timing Chart for STOP Mode Halt State Cleared by Interrupt

Table 3.3.5 Sample Warm-up Times after Clearance of STOP Mode

at f_{OSCH} = 33 MHz, fs = 32.768 kHz

SYSCR0 <RSYSCK>	SYSCR2<WUPTM1:0>		
	01 (2^8)	10 (2^{14})	11 (2^{16})
0 (fc)	8 μs	0.496 ms	1.986 ms
1 (fs)	7.8 ms	500 ms	2000 ms

Not Recommended for New Design

Downloaded from Arrow.com.

Example:

The STOP mode is entered when the low-frequency operates, and high-frequency operates after releasing due to NMI.

```
Address
              SYSCR0   EQU   00E0H
              SYSCR1   EQU   00E1H
              SYSCR2   EQU   00E2H
8FFDH                  LD    (SYSCR1), 08H          ; fSYS = fs/2
9000H                  LD    (SYSCR2), X–1001–1B    ; Sets warm-up time to 2^14/fOSCH
9002H                  LD    (SYSCR0), 011000 – –B  ; Operates high frequency after released

9005H                  HALT
              NMI                                     Clears and starts hit
                                                        warm-up timer
                                                      (High frequency)
                                                            End
                                                    NMI Interrupt routine

9006H                  LD    XX, XX                        RETI

 – : No change
```

Note: When different modes are used before and after STOP mode as the above mentioned, there is possible to release the HALT mode without changing the operation mode by acceptance of the halt release interrupt request during execution of HALT instruction (during 6 states). In the system which accepts the interrupts during execution HALT instruction, set the same operation mode before and after the STOP mode.

Not Recommended for New Design

Downloaded from Arrow.com.

Table 3.3.6 Input Buffer State Table

<table>
<tr><th rowspan="4">Port Name</th><th rowspan="4">Input Function Name</th><th colspan="9">Input Buffer State</th></tr>
<tr><th rowspan="3">During Reset</th><th colspan="2" rowspan="2">When the CPU is operating</th><th colspan="2" rowspan="2">In HALT mode(IDLE2)</th><th colspan="4">In HALT mode(IDLE1/STOP)</th></tr>
<tr><th colspan="2">Condition A (Note)</th><th colspan="2">Condition B (Note)</th></tr>
<tr><th>When Used as function Pin</th><th>When Used as Input Port</th><th>When Used as function Pin</th><th>When Used as Input Port</th><th>When Used as function Pin</th><th>When Used as Input Port</th><th>When Used as function Pin</th><th>When Used as Input Port</th></tr>
<tr><td>D0-D7</td><td>–</td><td rowspan="2">OFF</td><td rowspan="2">ON upon external read</td><td>–</td><td rowspan="2">OFF</td><td>–</td><td rowspan="3">OFF</td><td>–</td><td rowspan="2">OFF</td><td>–</td></tr>
<tr><td>P10-P17</td><td>D8-D15</td><td rowspan="3">ON</td><td>OFF</td><td rowspan="3">OFF</td><td>OFF</td></tr>
<tr><td>P54(*1)</td><td>$\overline{\text{BUSRQ}}$</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td></tr>
<tr><td>P55(*1)</td><td>–</td><td>OFF</td><td>–</td><td>–</td><td>OFF</td><td>–</td><td>–</td><td>OFF</td></tr>
<tr><td>P56(*1)</td><td>$\overline{\text{WAIT}}$</td><td rowspan="4">ON</td><td rowspan="4">ON</td><td rowspan="4">ON</td><td rowspan="4">ON</td><td rowspan="4">ON</td><td rowspan="3">OFF</td><td rowspan="3">OFF</td><td rowspan="4">ON</td><td rowspan="4">ON</td></tr>
<tr><td>P70</td><td>SCK
OPTRX0</td></tr>
<tr><td>P71(*1)</td><td>SDA</td></tr>
<tr><td>P72(*1)</td><td>SI
SCL</td><td>ON</td><td>ON</td></tr>
<tr><td>P80-P82(*2)</td><td>–</td><td rowspan="3">OFF</td><td>–</td><td rowspan="3">ON upon port read</td><td>–</td><td rowspan="3">OFF</td><td>–</td><td rowspan="6">OFF</td><td>–</td><td rowspan="3">OFF</td></tr>
<tr><td>P83(*2)</td><td>$\overline{\text{ADTRG}}$</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td></tr>
<tr><td>P84-P87(*2)</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>PB0</td><td>TA0IN</td><td rowspan="4">ON</td><td>ON</td><td rowspan="15">ON</td><td>ON</td><td rowspan="4">ON</td><td>OFF</td><td>ON</td><td rowspan="4">ON</td></tr>
<tr><td>PB1</td><td>–</td><td rowspan="2">–</td><td rowspan="2">–</td><td rowspan="2">–</td><td rowspan="2">–</td></tr>
<tr><td>PB2</td><td>–</td></tr>
<tr><td>PB3</td><td>INT0</td><td rowspan="4">ON</td><td rowspan="4">ON</td><td rowspan="4">ON</td><td>ON</td><td rowspan="4">ON</td></tr>
<tr><td>PB4</td><td>INT1</td><td rowspan="3">OFF</td><td rowspan="3">OFF</td><td rowspan="11">OFF</td><td rowspan="3">OFF</td></tr>
<tr><td>PB5</td><td>INT2</td></tr>
<tr><td>PB6</td><td>INT3</td></tr>
<tr><td>PC0</td><td>–</td><td rowspan="6">ON</td><td>–</td><td>–</td><td rowspan="6">ON</td><td>–</td><td>–</td><td rowspan="6">ON</td></tr>
<tr><td>PC1</td><td>RXD0</td><td rowspan="2">ON</td><td rowspan="2">ON</td><td rowspan="2">OFF</td><td rowspan="2">ON</td></tr>
<tr><td>PC2</td><td>SCLK0
CTS0</td></tr>
<tr><td>PC3</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>PC4</td><td>RXD1</td><td rowspan="2">ON</td><td rowspan="2">ON</td><td rowspan="2">OFF</td><td rowspan="2">ON</td></tr>
<tr><td>PC5</td><td>SCLK1
CTS1</td></tr>
<tr><td>PZ2(*1)</td><td>–</td><td rowspan="2">OFF</td><td rowspan="2">–</td><td rowspan="2">–</td><td rowspan="2">OFF</td><td rowspan="2">–</td><td rowspan="2">–</td><td rowspan="2">OFF</td></tr>
<tr><td>PZ3(*1)</td><td>–</td></tr>
<tr><td>NMI</td><td>–</td><td rowspan="4">ON</td><td rowspan="4">ON</td><td rowspan="4">–</td><td rowspan="4">ON</td><td rowspan="4">–</td><td rowspan="3">ON</td><td rowspan="3">–</td><td rowspan="3">ON</td><td rowspan="3">–</td></tr>
<tr><td>RESET</td><td>–</td></tr>
<tr><td>AM0,AM1</td><td>–</td></tr>
<tr><td>X1,XT1</td><td>–</td><td colspan="4">IDLE1 : ON , STOP : OFF</td></tr>
</table>

ON: The buffer is always turned on. A current flows the input buffer if the input pin is not driven.

OFF: The buffer is always turned off.

–: No applicable

*1: Port having a pull-up/pull-down resistor.

*2: AIN input does not cause a current to flow through the buffer.

Note: Condition A/B are as follows.

<table>
<tr><th colspan="2">SYSCR2 register setting</th><th colspan="2">HALT mode</th></tr>
<tr><th><DRVE></th><th><SELDRV></th><th>IDLE1</th><th>STOP</th></tr>
<tr><td>0</td><td>0</td><td>Condition B</td><td rowspan="2">Condition A</td></tr>
<tr><td>0</td><td>1</td><td>Condition A</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Condition B</td><td rowspan="2">Condition B</td></tr>
<tr><td>1</td><td>1</td></tr>
</table>

Not Recommended for New Design

Downloaded from Arrow.com.

Table 3.3.7 Output buffer State Table

Port Name	Output Function Name	Output Buffer State: During Reset	When the CPU is Operating: When Used as function Pin	When the CPU is Operating: When Used as Output Port	In HALT mode(IDLE2): When Used as function Pin	In HALT mode(IDLE2): When Used as Output Port	In HALT mode (IDLE1/STOP) Condition A (Note): When Used as function Pin	In HALT mode (IDLE1/STOP) Condition A (Note): When Used as Output Port	In HALT mode (IDLE1/STOP) Condition B (Note): When Used as function Pin	In HALT mode (IDLE1/STOP) Condition B (Note): When Used as Output Port
D0-D7	–	OFF	ON upon external write	–	OFF	–	OFF	–	OFF	–
P10-P17	D8-15			ON		ON		OFF		ON
P20-P27	A16-23	ON	ON		ON				ON	
A0-A15	–			–		–		–		–
RD	–									
WR	–									
P54(*1)	–	OFF	–	ON	–	ON	–	OFF	–	ON
P55(*1)	BUSAK		ON		ON		OFF		ON	
P56(*1)	–		–		–		–		–	
P60	CS0	ON	ON		ON		OFF		ON	
P61	CS1									
P62	CS2 , CS2A									
P63	CS3									
P64	EA24 CS2B									
P65	EA25 CS2C									
P66	CS2D									
P67	CS2E									
P70	SCK	OFF								
P71(*1)	SDA SO OPTTX0									
P72(*1)	SCL									
PB0	–		–		–		–		–	
PB1	TA1OUT		ON		ON		OFF		ON	
PB2	TA3OUT									
PB3-PB6	–		–		–		–		–	
PC0	TXD0		ON		ON		OFF		ON	
PC1	–		–		–		–		–	
PC2	SCLK0		ON		ON		OFF		ON	
PC3	TXD1									
PC4	–		–		–		–		–	
PC5	SCLK1		ON		ON		OFF		ON	
PD5	SCOUT	ON								
PD6	ALARM MLDALM									
PD7	MLDALM									
PZ2(*1)	HWR	OFF								
PZ3(*1)	R/W									
X2	–	ON		–		–	IDLE1 : ON , STOP : output "H" level			
XT2	–						IDLE1 : ON , STOP : High-Z			

ON: The buffer is always turned on. When the bus is released, however, output buffers for some pins are turned off.

OFF: The buffer is always turned off.

–: No applicable

*1: Port having a pull-up/pull-down resistor.

Note: Condition A/B are as follows.

SYSCR2 register setting		HALT mode	
<DRVE>	<SELDRV>	IDLE1	STOP
0	0	Condition B	Condition A
0	1	Condition A	
1	0	Condition B	Condition B
1	1		

Downloaded from Arrow.com.

3.4 Interrupts

Interrupts are controlled by the CPU interrupt mask register SR<IFF2:0> and by the built-in interrupt controller.

The TMP91C824 has a total of 37 interrupts divided into the following five types:

- Interrupts generated by CPU: 9 sources
  (Software interrupts, illegal instruction interrupt)
- Internal interrupts: 23 sources
- Interrupts on external pins ($\overline{\text{NMI}}$ and INT0 to INT3): 5 sources

A (Fixed) individual interrupt vector number is assigned to each interrupt.

One of six (Variable) priority level can be assigned to each maskable interrupt.

The priority level of non-maskable interrupts are fixed at 7 as the highest level.

When an interrupt is generated, the interrupt controller sends the piority of that interrupt to the CPU. If multiple interrupts are generated simultaneously, the interrupt controller sends the interrupt with the highest priority to the CPU. (The highest priority is level 7 using for non-maskable interrupts.)

The CPU compares the priority level of the interrupt with the value of the CPU interrupt mask register <IFF2:0>. If the priority level of the interrupt is higher than the value of the interrupt mask register, the CPU accepts the interrupt.

The interrupt mask register <IFF2:0> value can be updated using the value of the EI instruction (EI num sets <IFF2:0> data to num).

For example, specifying EI 3 enables the maskable interrupts which priority level set in the interrupt controller is 3 or higher, and also non-maskable interrupts.

Operationally, the DI instruction (<IFF2:0> = 7) is identical to the EI7 instruction. DI instruction is used to disable maskable interrupts because of the priority level of maskable interrupts is 1 to 6. The EI instruction is vaild immediately after execution.

In addition to the above general-purpose interrupt processing mode, TLCS-900/L1 has a micro DMA interrupt processing mode as well. The CPU can transfer the data (1/2/4 bytes) automatically in micro DMA mode, therefore this mode is used for speed-up interrupt processing, such as transferring data to the internal or external peripheral I/O. Moreover, TMP91C824 has software start function for micro DMA processing request by the software not by the hardware interrupt.

Figure 3.4.1 shows the overall interrupt processing flow.

Not Recommended for New Design

Downloaded from Arrow.com.

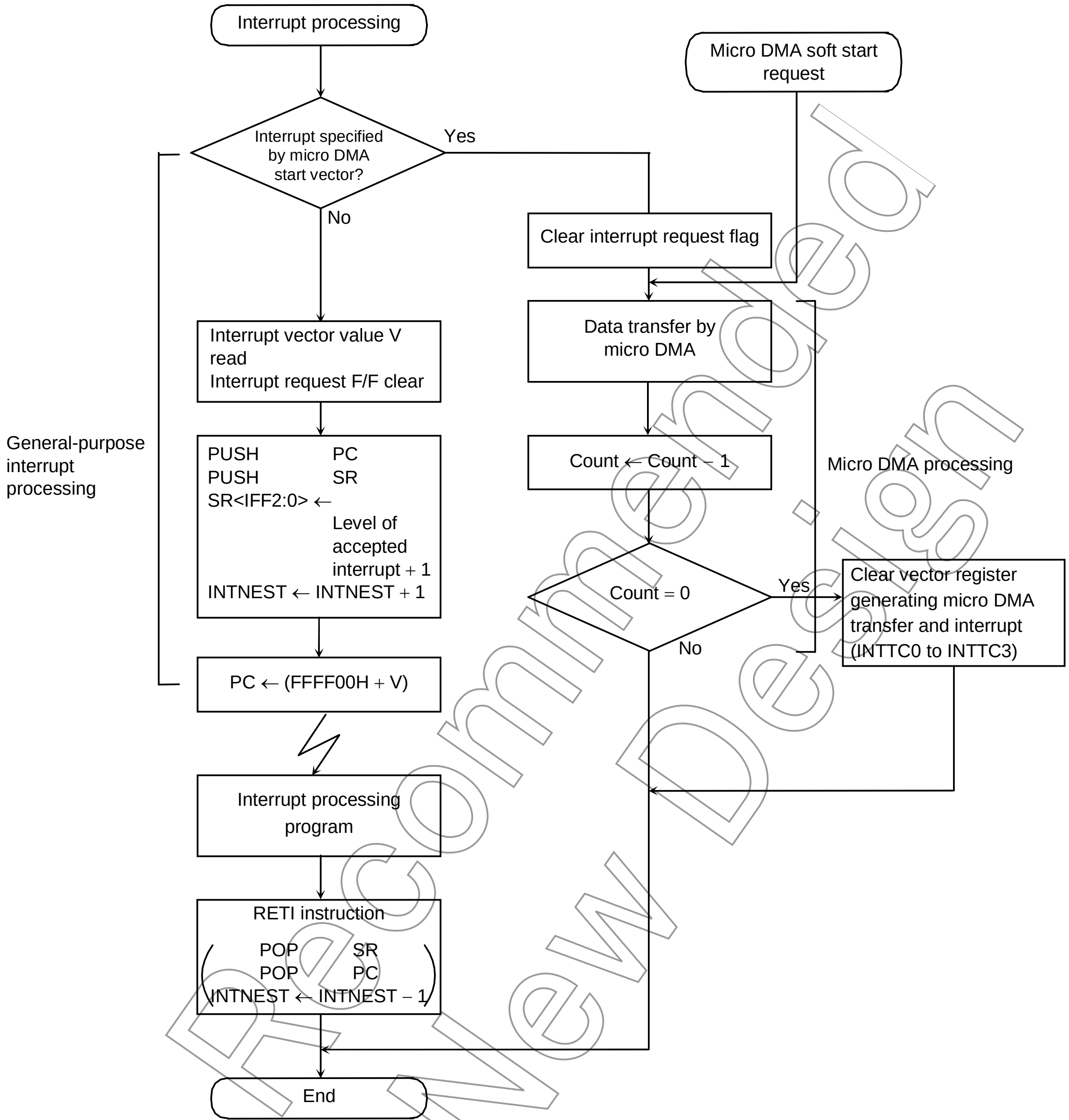


Figure 3.4.1 Overall Interrupt Processing Flow

Not Recommended for New Design

Downloaded from Arrow.com.

3.4.1 General-purpose Interrupt Processing

When the CPU accepts an interrupt, it usually performs the following sequence of operations. That is also the same as TLCS-900/L and TLCS-900/H.

(1) The CPU reads the interrupt vector from the interrupt controller.
If the same level interrupts occur simultaneously, the interrupt controller generates an interrupt vector in accordance with the default priority and clears the interrupt request.
(The default priority is already fixed for each interrupt: The smaller vector value has the higher priority level.)

(2) The CPU pushes the value of program counter (PC) and status register (SR) onto the stack area (Indicated by XSP).

(3) The CPU sets the value which is the priority level of the accepted interrupt plus 1 (+1) to the interrupt mask register <IFF2:0>. However, if the priority level of the accepted interrupt is 7, the register's value is set to 7.

(4) The CPU increases the interrupt nesting counter INTNEST by 1 (+1).

(5) The CPU jumps to the address indicated by the data at address FFFF00H + interrupt vector and starts the interrupt processing routine.

(6) The above processing time is 18-states (1.09 μs at 33 MHz) as the best case (16 bits data bus width and 0 waits).

When the CPU compled the interrupt processing, use the RETI instruction to return to the main routine. RETI restores the contents of program counter (PC) and status register (SR) from the stack and decreases the interrupt nesting counter INTNEST by 1 (–1).

Non-maskable interrupts cannot be disabled by a user program. Maskable interrupts, however, can be enabled or disabled by a user program. A program can set the priority level for each interrupt source. (A priority level setting of 0 or 7 will disable an interrupt request.)

If an interrupt request which has a priority level equal to or greater than the value of the CPU interrupt mask register <IFF2:0> comes out, the CPU accepts its interrupt. Then, the CPU interrupt mask register <IFF2:0> is set to the value of the priority level for the accepted interrupt plus 1 (+1).

Therefore, if an interrupt is generated with a higher level than the current interrupt during its processing, the CPU accepts the later interrupt and goes to the nesting status of interrupt processing.

Moreover, if the CPU receives another interrupt request while performing the said (1) to (5) processing steps of the current interrupt, the latest interrupt request is sampled immediately after execution of the first instruction of the current interrupt processing routine. Specifying DI as the start instruction disables maskable interrupt nesting.

A reset initializes the interrupt mask register <IFF2:0> to 111, disabling all maskable interrupts.

Table 3.4.1 shows the TMP91C824 interrupt vectors and micro DMA start vectors. The address FFFF00H to FFFFFFH (256 bytes) is assigned for the interrupt vector area.

Not Recommended for New Design

Downloaded from Arrow.com.

Table 3.4.1 TMP91C824 Interrupt Vectors Table

Default Priority	Type	Interrupt Source and Source of Micro DMA Request	Vector Value (V)	Vector Reference Address	Micro DMA Start Vector
1	Non maskable	Reset or “SWI 0” instruction	0000H	FFFF00H	–
2		“SWI 1” instruction	0004H	FFFF04H	–
3		INTUNDEF: Illegal instruction or “SWI 2” instruction	0008H	FFFF08H	–
4		“SWI 3” instruction	000CH	FFFF0CH	–
5		“SWI 4” instruction	0010H	FFFF10H	–
6		“SWI 5” instruction	0014H	FFFF14H	–
7		“SWI 6” instruction	0018H	FFFF18H	–
8		“SWI 7” instruction	001CH	FFFF1CH	–
9		$\overline{\text{NMI}}$ pin	0020H	FFFF20H	–
10		INTWD: Watchdog timer	0024H	FFFF24H	–
–	Maskable	Micro DMA (MDMA)	–	–	–
11		INT0 pin	0028H	FFFF28H	0AH
12		INT1 pin	002CH	FFFF2CH	0BH
13		INT2 pin	0030H	FFFF30H	0CH
14		INT3 pin	0034H	FFFF34H	0DH
15		INTALM0: ALM0 (8 kHz)	0038H	FFFF38H	0EH
16		INTALM1: ALM1 (512 Hz)	003CH	FFFF3CH	0FH
17		INTALM2: ALM2 (64 Hz)	0040H	FFFF40H	10H
18		INTALM3: ALM3 (2 Hz)	0044H	FFFF44H	11H
19		INTALM4: ALM4 (1 Hz)	0048H	FFFF48H	12H
20		INTTA0: 8-bit timer 0	004CH	FFFF4CH	13H
21		INTTA1: 8-bit timer 1	0050H	FFFF50H	14H
22		INTTA2: 8-bit timer 2	0054H	FFFF54H	15H
23		INTTA3: 8-bit timer 3	0058H	FFFF58H	16H
24		INTRX0: Serial reception (Channel 0)	005CH	FFFF5CH	17H
25		INTTX0: Serial transmission (Channel 0)	0060H	FFFF60H	18H
26		INTRX1: Serial reception (Channel 1)	0064H	FFFF64H	19H
27		INTTX1: Serial transmission (Channel 1)	0068H	FFFF68H	1AH
28		INTAD: AD conversion end	006CH	FFFF6CH	1BH
29		INTRTC: RTC (Alarm interrupt)	0074H	FFFF74H	1DH
30		INTSBI: SBI interrupt	0078H	FFFF78H	1EH
31		INTP0: Protect 0 (WR to special SFR)	0080H	FFFF80H	20H
32		INTP1: Protect 1 (WR to ROM)	0084H	FFFF84H	21H
33		INTTC0: Micro DMA end (Channel 0)	0088H	FFFF88H	–
34		INTTC1: Micro DMA end (Channel 1)	008CH	FFFF8CH	–
35		INTTC2: Micro DMA end (Channel 2)	0090H	FFFF90H	–
36		INTTC3: Micro DMA end (Channel 3)	0094H	FFFF94H	–
		(Reserved) : (Reserved)	0098H : 00FCH	FFFF98H : FFFFFCH	– : –

Not Recommended for New Design

Downloaded from Arrow.com.

3.4.2 Micro DMA Processing

In addition to general-purpose interrupt processing, the TMP91C824 supprots a micro DMA function. Interrupt requests set by micro DMA perform micro DMA processing at the highest priority level (Level 6) among maskable interrupts, regardless of the priority level of the particular interrupt source. Micro. The micro DMA has 4 channels and is possible continuous transmission by specifing the say later burst mode.

Because the micro DMA function has been implemented with the cooperative operation of CPU, when CPU goes to a standby mode by HALT instruction, the requirement of micro DMA will be ignored (Pending).

(1) Micro DMA operation

When an interrupt request specified by the micro DMA start vector register is generated, the micro DMA triggers a micro DMA request to the CPU at interrupt priority level 6 and starts processing the request in spite of any interrupt source's level. The micro DMA is ignored on <IFF2:0> = 7.

The 4 micro DMA channels allow micro DMA processing to be set for up to 4 types of interrupts at any one time. When micro DMA is accepted, the interrupt request flip-flop assigned to that channel is cleared.

The data are automatically transferred once (1/2/4 bytes) from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decreased by 1 (−1).

If the decreased result is 0, the micro DMA transfer end interrupt (INTTC0 to INTTC3) passes from the CPU to the interrupt controller. In addition, the micro DMA start vector register DMAnV is cleared to 0, the next micro DMA is disabled and micro DMA processing completes. If the decreased result is other than 0, the micro DMA processing completes if it isn't specified the say later burst mode. In this case, the micro DMA transfer end interrupt (INTTC0 to INTTC3) aren't generated.

If an interrupt request is triggered for the interrupt source in use during the interval between the clearing of the micro DMA start vector and the next setting, general-purpose interrupt processing executes at the interrupt level set. Therefore, if only using the interrupt for starting the micro DMA (Not using the interrupts as a general-purpose interrupt: Level 1 to 6), first set the interrupt level to 0 (Interrupt requests disabled).

If using micro DMA and general-purpose interrupts together, first set the level of the interrupt used to start micro DMA processing lower than all the other interrupt levels. In this case, the cause of general interrupt is limited to the edge interrupt. (Note)

The priority of the micro DMA transfer end interrupt (INTTC0 to INTTC3) is defined by the interrupt level and the default priority as the same as the other maskable interrupt.

Note: If the priority level of micro DMA is set higher than that of other interrupts, CPU operates as follows.
In case INTxxx interrupt is generated first and then INTyyy interrupt is generated between checking "Interrupt specified by micro DMA start vector" (in the Figure 3.4.1) and reading interrupt vector with setting below. The vector shifts to that of INTyyy at the time.
This is because the priority level of INTyyy is higher than that of INTxxx.
In the interrupt routine, CPU reads the vector of INTyyy because cheking of micro DMA has finished.
And INTyyy is generated regardless of transfer counter of micro DMA.
INTxxx: level 1 without micro DMA
INTyyy: level 6 with micro DMA

Not Recommended for New Design

Downloaded from Arrow.com.

If a micro DMA request is set for more than one channel at the same time, the priority is not based on the interrupt priority level but on the channel number. The smaller channel number has the higher priority (Channel 0 (High) > Channel 3 (Low)).

While the register for setting the transfer source/transfer destination addresses is a 32-bit control register, this register can only effectively output 24-bit addresses. Accordingly, micro DMA can access 16 Mbytes (The upper 8 bits of the 32 bits are not valid).

Three micro DMA transfer modes are supported: 1-byte transfer, 2-byte (One word) transfer, and 4-byte transfer. After a transfer in any mode, the transfer source/destination addresses are increased, decreased, or remain unchanged.

This simplifies the transfer of data from I/O to memory, from memory to I/O, and from I/O to I/O. For details of the transfer modes, see 3.4.2 (4) “Detailed description of the transfer mode register”. As the transfer counter is a 16-bit counter, micro DMA processing can be set for up to 65536 times per interrupt source. (The micro DMA processing count is maximized when the transfer counter initial value is set to 0000H.)

Micro DMA processing can be started by the 24 interrupts shown in the micro DMA start vectors of Table 3.4.1 and by the micro DMA soft start, making a total of 25 interrupts.

Figure 3.4.2 shows the word transfer micro DMA cycle in transfer destination address INC mode (except for counter mode, the same as for other modes).

(The conditions for this cycle are based on an external 16-bit bus, 0 waits, transfer source/transfer destination addresses both even-numberd values.)

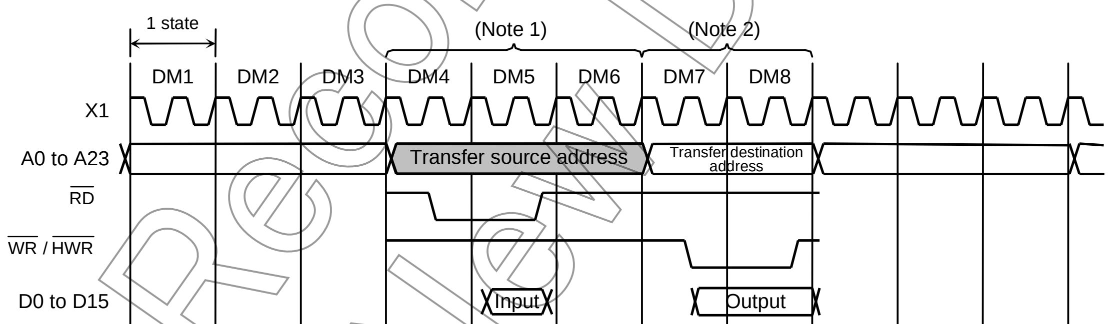


Figure 3.4.2 Timing for Micro DMA Cycle

States 1 to 3: Instruction fetch cycle (gets next address code).
If 3 bytes and more instruction codes are inserted in the instruction queue buffer, this cycle becomes a dummy cycle.

States 4 to 5: Micro DMA read cycle

State 6: Dummy cycle (The address bus remains unchanged from state 5)

States 7 to 8: Micro DMA write cycle

Note 1: If the source address area is an 8-bit bus, it is increased by two states.
If the source address area is a 16-bit bus and the address starts from an odd number, it is increased by two states.

Note 2: If the destination address area is an 8-bit bus, it is increased by two states.
If the destination address area is a 16-bit bus and the address starts from an odd number, it is increased by two states.

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Soft start function

In addition to starting the micro DMA function by interrupts, TMP91C824 includes a micro DMA software start function that starts micro DMA on the generation of the write cycle to the DMAR register.

Writing “1” to each bit of DMAR register causes micro DMA once (If write “0” to each bit, micro DMA doesn’t operate). At the end of transfer, the corresponding bit of the DMAR register which support the end channel are automatically cleared to “0”.

Only one-channel can be set for DMA request at once. (Do not write “1” to plural bits.)

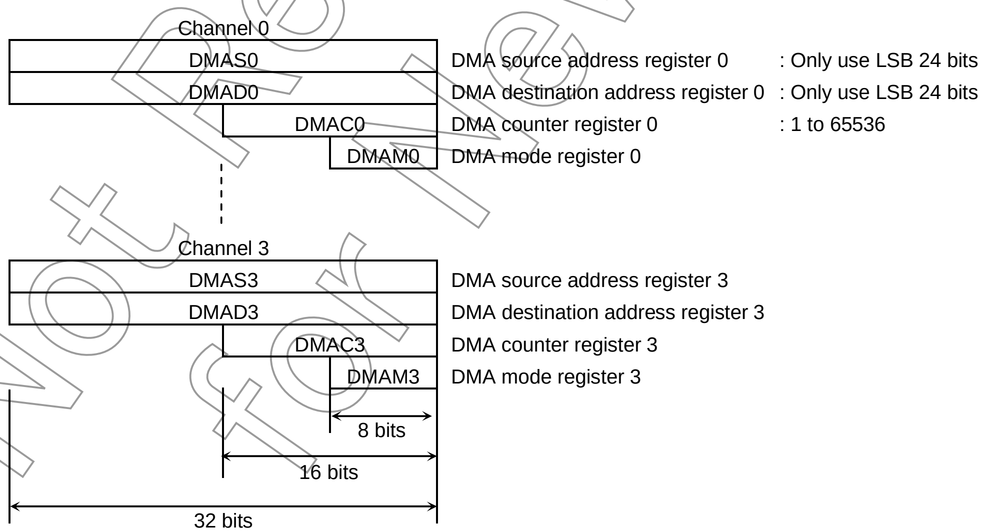
When writing again “1” to the DMAR register, check whether the bit is “0” before writing “1”. If read “1”, micro DMA transfer isn’t started yet.

When a burst is specified by DMAB register, data is continuously transferred until the value in the micro DMA transfer counter is “0” after start up of the micro DMA. If execute soft start during micro DMA transfer by interrupt source, micro DMA transfer counter doesn’t change. Don’t use Read-modify-write instruction to avoid writing to other bits by mistake.

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA request register	89H (Prohibit RMW)					DMAR3	DMAR2	DMAR1	DMAR0
							R/W			
							0	0	0	0
							DMA request			

(3) Transfer control registers

The transfer source address and the transfer destination address are set in the following registers in CPU. Data setting for these registers is done by an LDC cr, r instruction.



Not Recommended for New Design

Downloaded from Arrow.com.

(4) Detailed description of the transfer mode register

DMAM0 to DMAM3: 8 bits — 0 | 0 | 0 | Mode

Note: When setting a value in this register, write 0 to the upper 3 bits.

			Number of Transfer Bytes	Mode Description	Number of Execution States	Minimum Execution Time at fc = 33 MHz
000 (Fixed)	000	00	Byte transfer	Transfer destination address INC mode .... I/O to memory (DMADn+) ← (DMASn) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	485 ns
		01	Word transfer		12 states	727 ns
		10	4-byte transfer			
	001	00	Byte transfer	Transfer destination address DEC mode .... I/O to memory (DMADn–) ← (DMASn) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	485 ns
		01	Word transfer		12 states	727 ns
		10	4-byte transfer			
	010	00	Byte transfer	Transfer source address INC mode .... Memory to I/O (DMADn) ← (DMASn+) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	485 ns
		01	Word transfer		12 states	727 ns
		10	4-byte transfer			
	011	00	Byte transfer	Transfer source address DEC mode .... Memory to I/O (DMADn) ← (DMASn–) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	485 ns
		01	Word transfer		12 states	727 ns
		10	4-byte transfer			
	100	00	Byte transfer	Fixed address mode .... I/O to I/O (DMADn) ← (DMASn–) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	485 ns
		01	Word transfer		12 states	727 ns
		10	4-byte transfer			
	101	00	Counter mode .... for counting number of times interrupt is generated DMASn ← DMASn + 1 DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.		5 states	303 ns

Note 1: "n" is the corresponding micro DMA channels 0 to 3
DMADn+/DMASn+: Post-increment (Increment register value after transfer)
DMADn–/DMASn–: Post-decrement (Decrement register value after transfer)
The I/Os in the table mean fixed address and the memory means increment (INC) or decrement (DEC) addresses.

Note 2: Execution time is under the condition of:
16-bit bus width (Both translation and destination address area)/0 waits/fc = 33 MHz/selected high-frequency mode (fc × 1)

Note 3: Do not use an undefined code for the transfer mode register except for the defined codes listed in the above table.

Not Recommended for New Design

Downloaded from Arrow.com.

3.4.3 Interrupt Controller Operation

The block diagram in Figure 3.4.3 shows the interrupt circuits. The left-hand side of the diagram shows the interrupt controller circuit. The right-hand side shows the CPU interrupt request signal circuit and the halt release circuit.

For each of the 36 interrupt channels there is an interrupt request flag (Consisting of a flip-flop), an interrupt priority setting register and a micro DMA start vector register. The interrupt request flag latches interrupt requests from the peripherals. The flag is cleared to 0 in the following cases:

- when reset occurs
- when the CPU reads the channel vector after accepted its interrupt
- when executing an instruction that clears the interrupt (Write DMA start vector to INTCLR register)
- when the CPU receives a micro DMA request (when micro DMA is set)
- when the micro DMA burst transfer is terminated

An interrupt priority can be set independently for each interrupt source by writing the priority to the interrupt priority setting register (e.g., INTE0AD or INTE12). 6 interrupt priorities levels (1 to 6) are provided. Setting an interrupt source's priority level to 0 (or 7) disables interrupt requests from that source. The priority of non-maskable interrupts (NMI pin interrupts and watchdog timer interrupts) is fixed at 7. If interrupt request with the same level are generated at the same time, the default priority (The interrupt with the lowest priority or, in other words, the interrupt with the lowest vector value) is used to determine which interrupt request is accepted first.

The 3rd and 7th bits of the interrupt priority setting register indicate the state of the interrupt request flag and thus whether an interrupt request for a given channel has occurred.

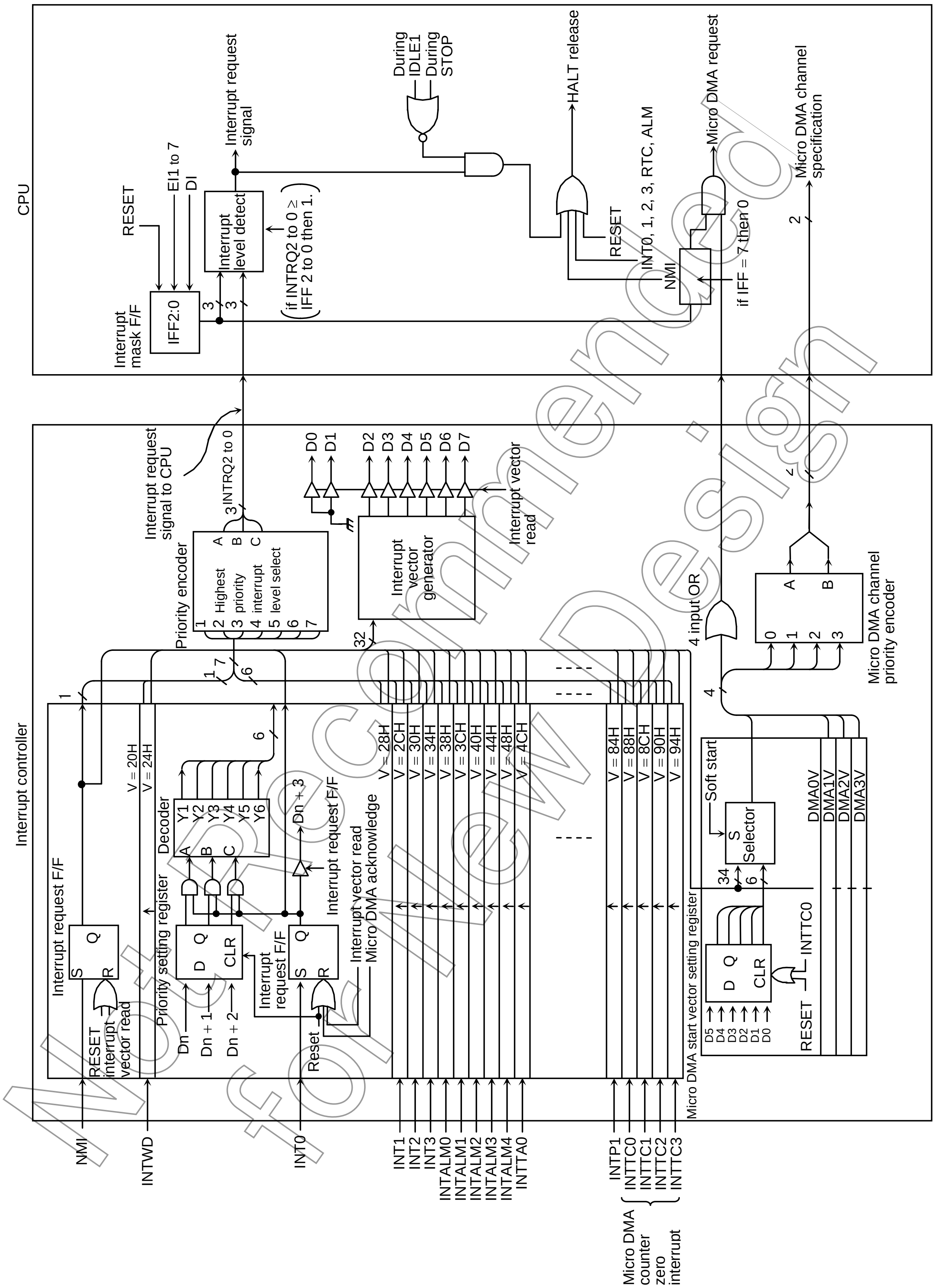
The interrupt controller sends the interrupt request with the highest priority among the simulateous interrupts and its vector address to the CPU. The CPU compares the priority value <IFF2:0> in the status register by the interrupt request signal with the priority value set;if the latter is higher, the interrupt is accepted. Then the CPU sets a value higher than the priority value by 1 (+1) in the CPU SR<IFF2:0>. Interrupt request where the priority value equals or is higher than the set value are accepted simultaneously during the previous interrupt routine.

When interrupt processing is completed (after execution of the RETI instruction), the CPU restores the priority value saved in the stack before the interrupt was generated to the CPU SR<IFF2:0>.

The interrupt controller also has registers (4 channels) used to store the micro DMA start vector. Writing the start vector of the interrupt source for the micro DMA processing (See Table 3.4.1), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter register (e.g., DMAS and DMAD) prior to the micro DMA processing.

Not Recommended for New Design

Downloaded from Arrow.com.



Not Recommended for New Design

Figure 3.4.3 Block Diagram of Interrupt Controller

Downloaded from Arrow.com.

(1) Interrupt level setting registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 & INTAD enable	90H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE12	INT1 & INT2 enable	91H	INT2				INT1			
			I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE3ALM4	INT3 & INTALM4 enable	92H	INTALM4				INT3			
			IA4C	IA4M2	IA4M1	IA4M0	I3C	I3M2	I3M1	I3M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEALM01	INTALM0 & INTALM1 enable	93H	INTALM1				INTALM0			
			IA1C	IA1M2	IA1M1	IA1M0	IA0C	IA0M2	IA0M1	IA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEALM23	INTALM2 & INTALM3 enable	94H	INTALM3				INTALM2			
			IA3C	IA3M2	IA3M1	IA3M0	IA2C	IA2M2	IA2M1	IA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETA01	INTTA0 & INTTA1 enable	95H	INTTA1 (TMRA1)				INTTA0 (TMRA0)			
			ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETA23	INTTA2 & INTTA3 enable	96H	INTTA3 (TMRA3)				INTTA2 (TMRA2)			
			ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTERTC	INTRTC enable	97H					INTRTC			
							IRC	IRM2	IRM1	IRM0
							R	R/W		
							0	0	0	0

Interrupt request flag

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt requests
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt requests

Not Recommended for New Design

Downloaded from Arrow.com.

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTES0	Interrupt Enable serial 0	98H	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES1	INTRX1 & INTTX1 enable	99H	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES2	INTESBI enable	9AH					INTSBI			
							ISBIC	ISBIM2	ISBIM1	ISBIM0
							R	R/W		
							0	0	0	0
INTETC01	INTTC0 & INTTC1 enable	9BH	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC23	INTTC2 & NTTC3 enable	9CH	INTTC3				INTTC2			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEP01	INTP0 & NTP1 enable	9DH	INTP1				INTP0			
			IP1C	IP1M2	IP1M1	IP1M0	IP0C	IP0M2	IP0M1	IP0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Interrupt request flag

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt requests
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt requests

Not Recommended for New Design

Downloaded from Arrow.com.

(2) External interrupt control

Symbol	Name	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt input mode control	8CH (Prohibit RMW)	–	–	I3EDGE	I2EDGE	I1EDGE	I0EDGE	I0LE	NMIREE
			W							
			0	0	0	0	0	0	0	0
			Always write 0	Always write 0	INT3EDGE 0: Rising 1: Falling	INT2EDGE 0: Rising 1: Falling	INT1EDGE 0: Rising 1: Falling	INT0EDGE 0: Rising 1: Falling	INT0 mode 0: Edge 1: Level	1: Operates even on rising/ falling edge of $\overline{\text{NMI}}$

INT0 level enable

0	edge detect INT
1	H level INT

$\overline{\text{NMI}}$ rising edge enable

0	INT request generation at falling edge
1	INT request generation at rising/falling edge

(3) Interrupt request flag clear register

The interrupt request flag is cleared by writing the appropriate micro DMA start vector, as given in Table 3.4.1, to the register INTCLR.

For example, to clear the interrupt flag INT0, perform the following register operation after execution of the DI instruction.

INTCLR ← 0AH: Clears interrupt request flag INT0.

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt clear control	88H (Prohibit RMW)			CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
					W					
					0	0	0	0	0	0
					Interrupt vector					

(4) Micro DMA start vector registers

This register assigns micro DMA processing to which interrupt source. The interrupt source with a micro DMA start vector that matches the vector set in this register is assigned as the micro DMA start source.

When the micro DMA transfer counter value reaches zero, the micro DMA transfer end interrupt corresponding to the channel is sent to the interrupt controller, the micro DMA start vector register is cleared, and the micro DMA start source for the channel is cleared. Therefore, to continue micro DMA processing, set the micro DMA start vector register again during the processing of the micro DMA transfer end interrupt.

If the same vector is set in the micro DMA start vector registers of more than one channel, the channel with the lowest number has a higher priority.

Accordingly, if the same vector is set in the micro DMA start vector registers of two channels, the interrupt generated in the channel with the lower number is executed until micro DMA transfer is complete. If the micro DMA start vector for this channel is not set again, the next micro DMA is started for the channel with the higher number (Micro DMA chaining).

Not Recommended for New Design

Downloaded from Arrow.com.

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA0 start vector	80H			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
					DMA0 start vector					
DMA1V	DMA1 start vector	81H			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
					DMA1 start vector					
DMA2V	DMA2 start vector	82H			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
					DMA2 start vector					
DMA3V	DMA3 start vector	83H			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
					0	0	0	0	0	0
					DMA3 start vector					

(5) Micro DMA burst specification

Specifying the micro DMA burst continues the micro DMA transfer until the transfer counter register reaches zero after micro DMA start. Setting a bit which corresponds to the micro DMA channel of the DMAB registers mentioned below to 1 specifies a burst.

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA software request register	89H (Prohibit RMW)					DMAR3	DMAR2	DMAR1	DMAR0
							R/W	R/W	R/W	R/W
							0	0	0	0
							1: DMA software request			
DMAB	DMA burst register	8AH					DMAB3	DMAB2	DMAB1	DMAB0
							R/W			
							0	0	0	0
							1: DMA burst request			

Not Recommended for New Design

Downloaded from Arrow.com.

(6) Attention point

The instruction execution unit and the bus interface unit of this CPU operate independently. Therefore, immediately before an interrupt is generated, if the CPU fetches an instruction that clears the corresponding interrupt request flag, the CPU may execute the instruction that clears the interrupt request flag (Note) between accepting and reading the interrupt vector. In this case, the CPU reads the default vector 0008H and reads the interrupt vector address FFFF08H.

To avoid the avobe plogram, place instructions that clear interrupt request flags after a DI instruction. And in the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing and more than 1-instructions (e.g., “NOP” × 1 times).

In the case of changing the value of the interrupt mask register <IFF2:0> by execution of POP SR instruction, disable an interrupt by DI instruction before execution of POP SR instruction.

In addition, take care as the following 2 circuits are exceptional and demand special attention.

INT0 Level Mode	In level mode INT0 is not an edge-triggered interrupt. Hence, in level mode the interrupt request flip-flop for INT0 does not function. The peripheral interrupt request passes through the S input of the flip-flop and becomes the Q output. If the interrupt input mode is changed from edge mode to level mode, the interrupt request flag is cleared automatically.

	If the CPU enters the interrupt response sequence as a result of INT0 going from 0 to 1, INT0 must then be held at 1 until the interrupt response sequence has been completed. If INT0 is set to level mode so as to release a halt state, INT0 must be held at 1 from the time INT0 changes from 0 to 1 until the halt state is released. (Hence, it is necessary to ensure that input noise is not interpreted as a 0, causing INT0 to revert to 0 before the halt state has been released.)
	When the mode changes from level mode to edge mode, interrupt request flags which were set in level mode will not be cleared. Interrupt request flags must be cleared using the following sequence.
	DI
	LD (IIMC), 00H; Switches interrupt input mode from level mode to edge mode.
	LD (INTCLR), 0AH; Clears interrupt request flag.
	NOP ; Wait EI instruction
	EI

INTRX	The interrupt request flip-flop can only be cleared by a reset or by reading the serial channel receive buffer. It cannot be cleared by writing INTCLR register.

Note: The following instructions or pin input state changes are equivalent to instructions that clear the interrupt request flag.

INT0: Instructions which switch to level mode after an interrupt request has been generated in edge mode.
The pin input change from high to low after interrupt request has been generated in level mode. (H → L)

INTRX: Instruction which read the receive buffer

Downloaded from Arrow.com.

Not Recommended for New Design

3.5 Port Functions

The TMP91C824 features 56-bit settings which relate to the various I/O ports.

As well as general-purpose I/O port functionality, the port pins also have I/O functions which relate to the built-in CPU and internal I/Os. Table 3.5.1 lists the functions of each port pin. Table 3.5.2 lists I/O registers and their specifications.

Table 3.5.1 Port Functions

(R: PU = with programmable pull-up resistor/U = with pull-up resistor)

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port 1	P10 to P17	8	I/O	–	Bit	D8 to D15
Port 2	P20 to P27	8	Output	–	(Fixed)	A16 to A23
Port 5	P54	1	I/O	PU	Bit	$\overline{\text{BUSRQ}}$
	P55	1	I/O	PU	Bit	$\overline{\text{BUSAK}}$
	P56	1	I/O	PU	Bit	$\overline{\text{WAIT}}$
Port 6	P60	1	Output	–	(Fixed)	$\overline{\text{CS0}}$
	P61	1	Output	–	(Fixed)	$\overline{\text{CS1}}$
	P62	1	Output	–	(Fixed)	$\overline{\text{CS2}}$, $\overline{\text{CS2A}}$
	P63	1	Output	–	(Fixed)	$\overline{\text{CS3}}$
	P64	1	Output	–	(Fixed)	EA24, $\overline{\text{CS2B}}$
	P65	1	Output	–	(Fixed)	EA25, $\overline{\text{CS2C}}$
	P66	1	Output	–	(Fixed)	$\overline{\text{CS2D}}$
	P67	1	Output	–	(Fixed)	$\overline{\text{CS2E}}$
Port 7	P70	1	I/O	–	Bit	SCK,OPTRX0
	P71	1	I/O	PU	Bit	SO/SDA,OPTTX0
	P72	1	I/O	PU	Bit	SI/SCL
Port 8	P80 to P87	8	Input	–	(Fixed)	AN0 to AN7, $\overline{\text{ADTRG}}$ (P83)
Port B	PB0	1	I/O	–	Bit	TA0IN
	PB1	1	I/O	–	Bit	TA1OUT
	PB2	1	I/O	–	Bit	TA3OUT
	PB3	1	I/O	–	Bit	INT0
	PB4	1	I/O	–	Bit	INT1
	PB5	1	I/O	–	Bit	INT2
	PB6	1	I/O	–	Bit	INT3
Port C	PC0	1	I/O	–	Bit	TXD0
	PC1	1	I/O	–	Bit	RXD0
	PC2	1	I/O	–	Bit	SCLK0/ $\overline{\text{CTS0}}$
	PC3	1	I/O	–	Bit	TXD1
	PC4	1	I/O	–	Bit	RXD1
	PC5	1	I/O	–	Bit	SCLK1/ $\overline{\text{CTS1}}$
Port D	PD5	1	Output	–	(Fixed)	SCOUT
	PD6	1	Output	–	(Fixed)	$\overline{\text{ALARM}}$, $\overline{\text{MLDALM}}$
	PD7	1	Output	–	(Fixed)	MLDALM
Port Z	PZ2	1	I/O	PU	Bit	$\overline{\text{HWR}}$
	PZ3	1	I/O	PU	Bit	R / $\overline{\text{W}}$

Not Recommended for New Design

Downloaded from Arrow.com.

Table 3.5.2 I/O Registers and Specifications (1/2)

Port	Pin Name	Specification	I/O Register			
			Pn	PnCR	PnFC	PnFC2
Port 1 (Note 1)	P10 to P17	Input port	X	0	None	None
		Output port	X	1		
		D8 to D15 bus	X	X		
Port 2	P20 to P27	Output port	X	None	0	
		A16 to A23 output	X		1	
Port 5	P54 to P56	Input port (without PU)	0	0	0	
		Input port (with PU)	1	0	0	
		Output port	X	1	0	
	P54	$\overline{\text{BUSRQ}}$ input (without PU)	0	0	1	
		$\overline{\text{BUSRQ}}$ input (with PU)	1	0	1	
	P55	$\overline{\text{BUSAK}}$ output	X	1	1	
	P56	$\overline{\text{WAIT}}$ input (without PU)	0	0	None	
		$\overline{\text{WAIT}}$ input (with PU)	1	0		
Port 6	P60 to P64	Output port	X	None	0	0
	P60	$\overline{\text{CS0}}$ output	X		1	None
	P61	$\overline{\text{CS1}}$ output	X		1	
	P62	$\overline{\text{CS2}}$ output	X		1	0
		$\overline{\text{CS2A}}$ output	X		X	1
	P63	$\overline{\text{CS3}}$ output	X		1	None
	P64	EA24 output	X		1	0
		$\overline{\text{CS2B}}$ output	X		X	1
	P65	EA25 output	X		1	0
		$\overline{\text{CS2C}}$ output	X		X	1
	P66	$\overline{\text{CS2D}}$ output	X		0	1
	P67	$\overline{\text{CS2E}}$ output	X		0	1
Port 7	P70 to P72	Input port (without PU)	0	0	0	0
		Input port (with PU)	1	0	0	0
		Output port	X	1	0	0
	P70	SCK input	X	0	0	0
		SCK output	X	1	1	0
		OPTRX0 input (Note 2)	1	0	X	1
	P71	SDA input	X	0	0	0
		SDA output (Note 3)	X	1	1	0
		SO output	X	1	1	0
		OPTTX0 output (Note 2)	1	1	X	1
	P72	SI input	X	0	0	0
		SCL input	X	0	0	0
		SCL output (Note 3)	X	1	1	0

X: Don't care

Not Recommended for New Design

Downloaded from Arrow.com.

Table 3.5.3 I/O Registers and Specifications (2/2)

Port	Pin Name	Specification	I/O Register Pn	PnCR	PnFC	PnFC2
Port 8	P80 to P87	Input port	X	None		None
		AN0 to 7 input (Note 4)	X			
	P83	$\overline{\text{ADTRG}}$ input (Note 5)	X			
Port B	PB0 to PB6	Input port	X	0	0	
		Output port	X	1	0	
	PB0	TA0IN input	X	0	None	
	PB1	TA1OUT output	X	1	1	
	PB2	TA3OUT output	X	1	1	
	PB3	INT0 input	X	0	1	
	PB4	INT1 input	X	0	1	
	PB5	INT2 input	X	0	1	
	PB6	INT3 input	X	0	1	
Port C	PC0 to PC5	Input port	X	0	0	
		Output port	X	1	0	
	PC0	TXD0 output (Note 2)	1	1	1	
	PC1	RXD0 input (Note 2) (Note 6)	1	0	None	
	PC2	SCLK0 input (Note 2)	1	0	0	
		SCLK0 output (Note 2)	1	1	1	
		$\overline{\text{CTS0}}$ input (Note 2)	1	0	0	
	PC3	TXD1 output (Note 2)	1	1	1	
	PC4	RXD1 input (Note 2)	1	0	None	
	PC5	SCLK1 input (Note 2)	1	0	0	
		SCLK1 output (Note 2)	1	1	1	
		$\overline{\text{CTS1}}$ input (Note 2)	1	0	0	
Port D	PD5 to PD7	Output port	X	None	0	
	PD5	SCOUT output	X		1	
	PD6	$\overline{\text{ALARM}}$ output	1		1	
		$\overline{\text{MLDALM}}$ output	0		1	
	PD7	MLDALM output	X		1	
Port Z	PZ2 to PZ3	Input port (without PU)	0	0	0	
		Input port (with PU)	1	0	0	
		Output port	X	1	0	
	PZ2	$\overline{\text{HWR}}$ output	X	1	1	
	PZ3	R/$\overline{\text{W}}$ output	X	1	1	

X: Don't care

Note 1: Port 1 is only use for Port or DATA bus (D8 to D15) by setting AM1 and AM0 pins.

Note 2: As for input ports of SIO0 and SIO1 (OPTRX0, OPTTX0, TXD0, RXD0, SCLK0, $\overline{\text{CTS0}}$, TXD1, RXD1, SCLK1, $\overline{\text{CTS1}}$), logical selection for output data or input data is determined by the output latch register Pn of each port.

Note 3: In case using P71 and P72 for SDA and SCL as open-drain ports, set to P7ODE <ODEP71:72>.

Note 4: In case using P80 to P87 for analog input ports of AD converter, set to ADMOD1 <ADCH2:0>.

Note 5: In case using P83 for $\overline{\text{ADTRG}}$ input port, set to ADMOD1<ADTRGE>.

Note 6: In case using PC1 for RXD0 port, set 0 to P7FC2<P70F2>.

Not Recommended for New Design

Downloaded from Arrow.com.

Note about bus release and programmable pull-up I/O port pins

When the bus is released (e.g., when $\overline{BUSAK}$ = 0), the output buffers for D0 to D15, A0 to A23, and the control signals ($\overline{RD}$, $\overline{WR}$, $\overline{HWR}$, R/$\overline{W}$ and $\overline{CS0}$ to $\overline{CS3}$, EA24, EA25, $\overline{CS2A}$ to $\overline{2E}$) are off and are set to high impedance.

However, the output of built-in programmable pull-up resistors are kept before the bus is released. These programmable pull-up resistors can be selected ON/OFF by programmable when they are used as the input ports.

When they are used as output ports, they cannot be turned ON/OFF in software.

Table 3.5.4 shows the pin states after the bus has been released.

Table 3.5.4 Pin States (after bus release)

Pin Name	The Pin State (when the bus is released)	
	Port Mode	Function Mode
D0 to D7		Become high impedance (High-Z).
D8 to D15 (P10 to P17)	The state is not changed. (Do not become to high impedance (High-Z).)	↑
A0 to A15		First sets all bits to high, then sets them to high impedance (High-Z).
A16 to 23 (P20 to P27)	The state is not changed. (Do not become to high impedance (High-Z).)	↑
$\overline{RD}$ $\overline{WR}$		↑
PZ2 ($\overline{HWR}$), PZ3 (R/$\overline{W}$),	The state is not changed. (Do not become to high impedance (High-Z).)	First sets all bits to high, then the output buffer is OFF. The programmable pull-up resistor is ON irrespective of the output latch.
P60 ($\overline{CS0}$), P61 ($\overline{CS1}$), P62 ($\overline{CS2}$, $\overline{CS2A}$), P63 ($\overline{CS3}$),	↑	First sets all bits to high, then sets them to high impedance (High-Z).

Not Recommended for New Design

Downloaded from Arrow.com.

3.5.1 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O port. Each bit can be set individually for input or output using the control register P1CR. Resetting the control register P1CR to 0 and sets port 1 to input mode.

In addition to functioning as a general-purpose I/O port, port 1 can also function as an address data bus (D8 to D15).

When AM1 = 0 and AM0 = 1, port 10 to 17 always operate data bus function even if it changes P1CR setting.

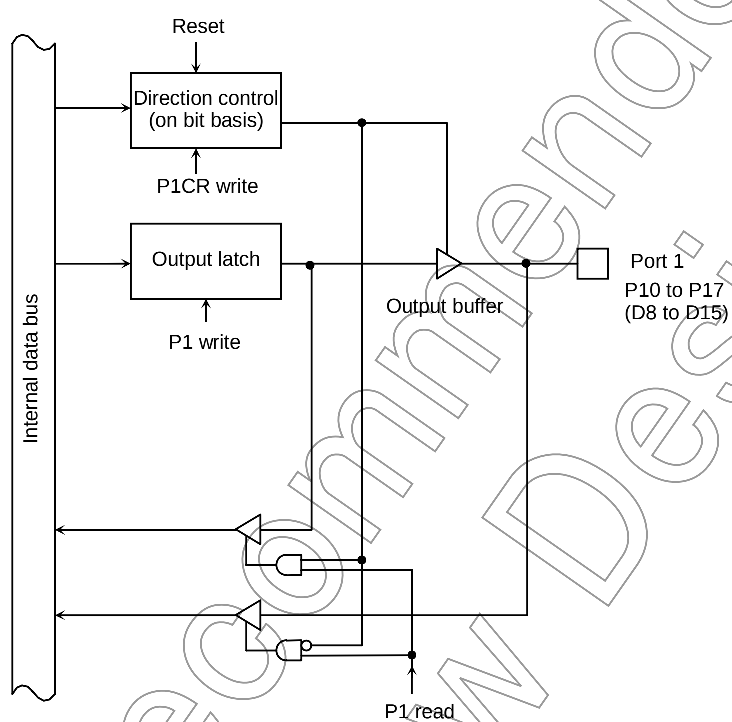


Figure 3.5.1 Port 1

Not Recommended for New Design

Downloaded from Arrow.com.

3.5.2 Port 2 (P20 to P27)

Port 2 is an 8-bit output port. In addition to functioning as a output port, port 2 can also function as an address bus (A16 to A23).

Each bit can be set individually for address bus using the function register P2FC. Resetting sets all bits of the function register P2FC to 1 and sets port 2 to address bus.

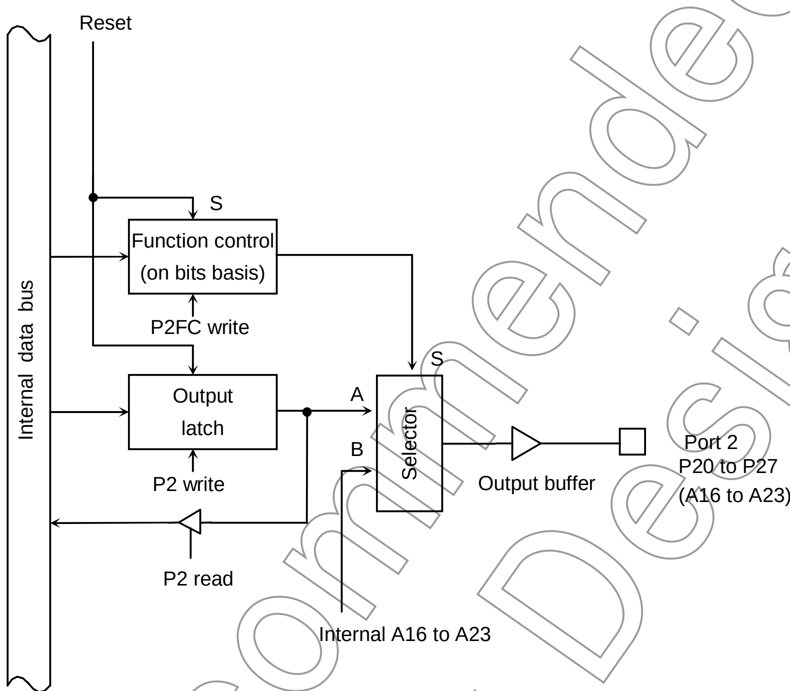


Figure 3.5.2 Port 2

Not Recommended for New Design

Downloaded from Arrow.com.

Port 1 Register

P1 (0001H)		7	6	5	4	3	2	1	0
	Bit symbol	P17	P16	P15	P14	P13	P12	P11	P10
	Read/Write	R/W							
	After reset	Data from external port (Output latch register is cleared to 0.)							

Port 1 Control Register

P1CR (0004H)		7	6	5	4	3	2	1	0
	Bit symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

Port 1 I/O setting

0	Input
1	Output

Port 2 Register

P2 (0006H)		7	6	5	4	3	2	1	0
	Bit symbol	P27	P26	P25	P24	P23	P22	P21	P20
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Port 2 Function Register

P2FC (0009H)		7	6	5	4	3	2	1	0
	Bit symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
	Read/Write	W							
	After reset	1	1	1	1	1	1	1	1
	Function	0: Port 1: Address bus (A23 to A16)							

Note: Read-modify-write is prohibited for P1CR and P2FC.

Figure 3.5.3 Registers for Ports 1 and 2

Not Recommended for New Design

Downloaded from Arrow.com.

3.5.3 Port 5 (P54 to P56)

Port 5 is an 3-bit general-purpose I/O port. I/O is set using control register P5CR and P5FC. Resetting resets all bits of the output latch P5 to 1, the control register P5CR and the function register P5FC to 0 and sets P54 to P56 to input mode with pull-up resistor.

In addition to functioning as a general-purpose I/O port, port 5 also functions as I/O for the CPU's control/status signal.

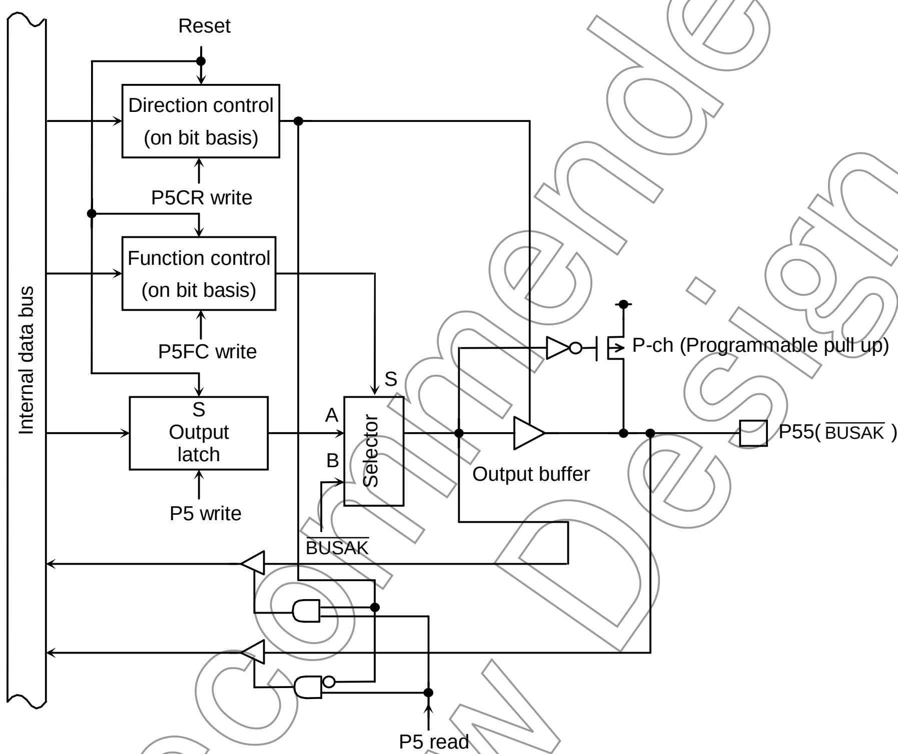


Figure 3.5.4 Port 5 (P55)

Not Recommended for New Design

Downloaded from Arrow.com.

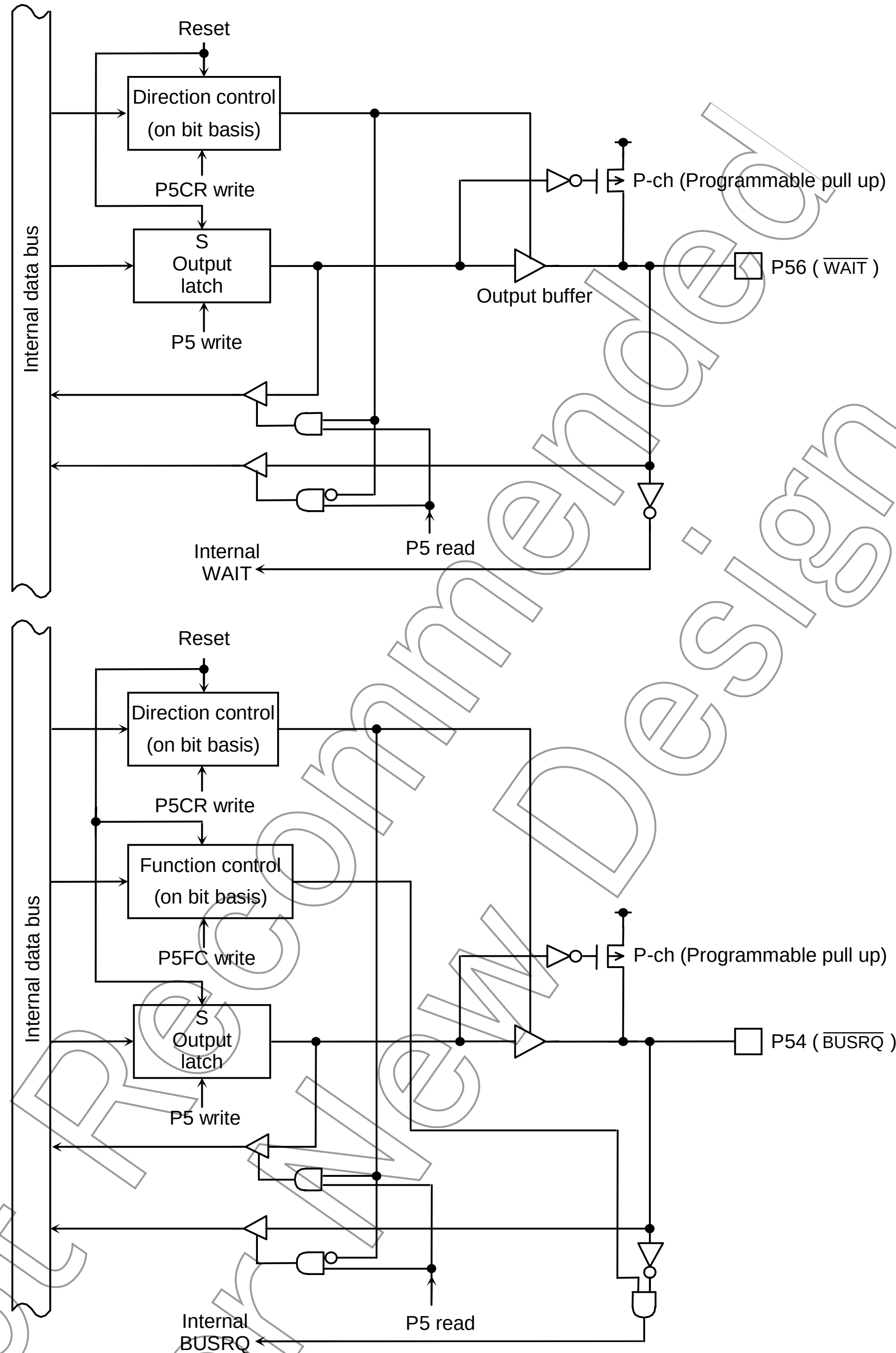


Figure 3.5.5 Port 5 (P56, P54)

Not Recommended for New Design

Downloaded from Arrow.com.

Port 5 Register

P5 (000DH)

	7	6	5	4	3	2	1	0
Bit symbol		P56	P55	P54				
Read/Write		R/W						
After reset		Data from external port (Output latch register is set to “1”.)						
Function		0(Output latch register) : Pull-up resistor OFF 1(Output latch register) : Pull-up resistor ON						

Port 5 Control Register

P5CR (000AH)

	7	6	5	4	3	2	1	0
Bit symbol		P56C	P55C	P54C				
Read/Write		W						
After reset		0	0	0				
Function		0: Input 1: Output						

I/O setting

0	Input
1	Output

Port 5 Function Register

P5FC (000BH)

	7	6	5	4	3	2	1	0
Bit symbol			P55F	P54F				
Read/Write			W					
After reset			0	0				
Function			0: Port 1: $\overline{\text{BUSAK}}$	0: Port 1: $\overline{\text{BUSRQ}}$				

Note 1: Read-modify-write is prohibited for register P5CR, P5FC.

Note 2: When port 5 is used in the input mode, P5 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Note 3: When P56 pin is used as a $\overline{\text{WAIT}}$ pin, set P5CR<P56C> to 0 and chip select/WAIT control register BnCS<BnW2:0> to 010.

Figure 3.5.6 Register for Port 5

Not Recommended for New Design

Downloaded from Arrow.com.

3.5.4 Port 6 (P60 to P67)

Port 60 to 67 are 8-bit output ports. Resetting sets output latch of P62 to 0 and output latches of P60 to P61, P63 to P67 to 1.

Port 6 also function as chip-select output ($\overline{CS0}$ to $\overline{CS3}$), extend address output (EA24).

Writing 1 in the corresponding bit of P6FC, P6FC2 enables the respective functions.

Resetting resets the P6FC, P6FC2 to 0, and sets all bits to output ports.

If set port 6, be careful of a setting because of chip select function.

Starting memory connects to CS2 pin, but this signal function as P62 after reset. Therefore initialized value of output data of P62 is set to "0". If manage chip select by connection many memory to outside, after program started, must to change port function to chip select function in this program. If outputted "1" remain port function, program is not run. Therefore data setting (P6) must to execute after function changing (P6FC).

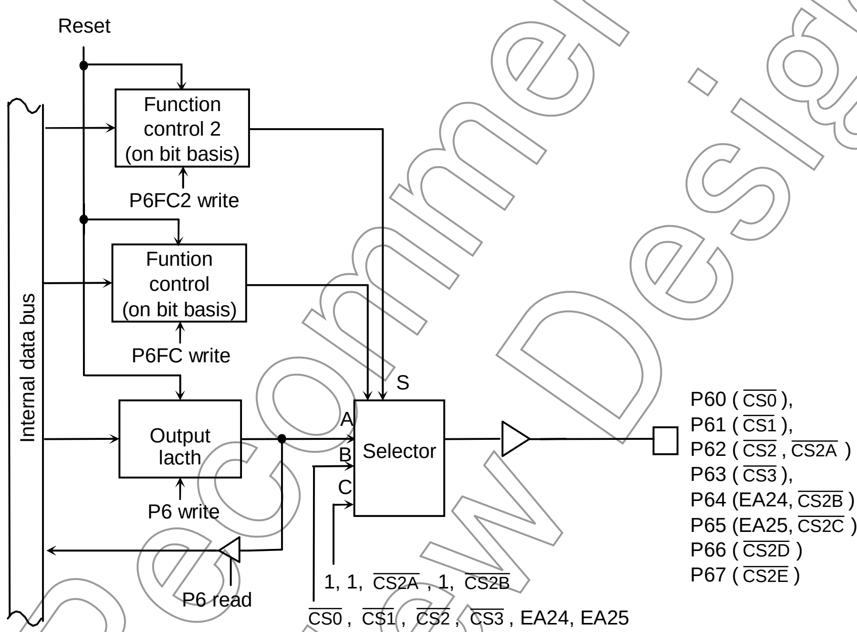


Figure 3.5.7 Port 6

Not Recommended for New Design

Downloaded from Arrow.com.

Port 6 Register

		7	6	5	4	3	2	1	0
P6 (0012H)	Bit symbol	P67	P66	P65	P64	P63	P62	P61	P60
	Read/Write	R/W							
	After reset	1	1	1	1	1	0	1	1

Port 6 Function Register

		7	6	5	4	3	2	1	0
P6FC (0015H)	Bit symbol	–	–	P65F	P64F	P63F	P62F	P61F	P60F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Always write 0		0:Port 1:EA25	0: Port 1: EA24	0: Port 1: $\overline{CS3}$	0: Port 1: $\overline{CS2}$	0: Port 1: $\overline{CS1}$	0: Port 1: $\overline{CS0}$

Port 6 Function Register 2

		7	6	5	4	3	2	1	0
P6FC2 (001BH)	Bit symbol	P67F2	P66F2	P65F2	P64F2	–	P62F2	–	–
	Read/Write	W				W	W	W	W
	After reset	0	0	0	0	0	0	0	0
	Function	0: <P67F> 1: $\overline{CS2E}$	0: <P66F> 1: $\overline{CS2D}$	0: <P65F> 1: $\overline{CS2C}$	0: <P64F> 1: $\overline{CS2B}$	Always write 0	0: <P62F> 1: $\overline{CS2A}$	Always write 0	

Note: Read-modify-write is prohibited for P6FC and P6FC2.

Figure 3.5.8 Register for Port 6

Not Recommended for New Design

Downloaded from Arrow.com.

3.5.5 Port 7 (P70 to P72)

Port 7 is a 3-bit general-purpose I/O port. I/O can be set on bit basis using the control register. Resetting sets port 7 to input port and all bits of output latch to 1.

In addition to functioning as a general-purpose I/O port, port 7 also functions as follows.

1. Input/output function for serial bus interface (SCK, SO/SDA,SI/SCL)
2. Input/output function for IrDA (OPTRX0, OPTTX0)

Writing 1 in the corresponding bit of P7FC, P7FC2 enables the respective functions.

Resetting resets the P7FC, P7FC2, and P7CR to 0, and sets all bits to input ports.

(1) Port 70 (SCK, OPTRX0)

Port 70 is a general-purpose I/O port. It is also used as SCK (Clock signal for SIO mode) and OPTRX0 (Receive input for IrDA mode of SIO0).

Used as OPTRX0, it is possible to logical invert by P7<P70> = 0.

For port C1, RXD0 or OPTRX0 is used P7FC2<P70F2>.

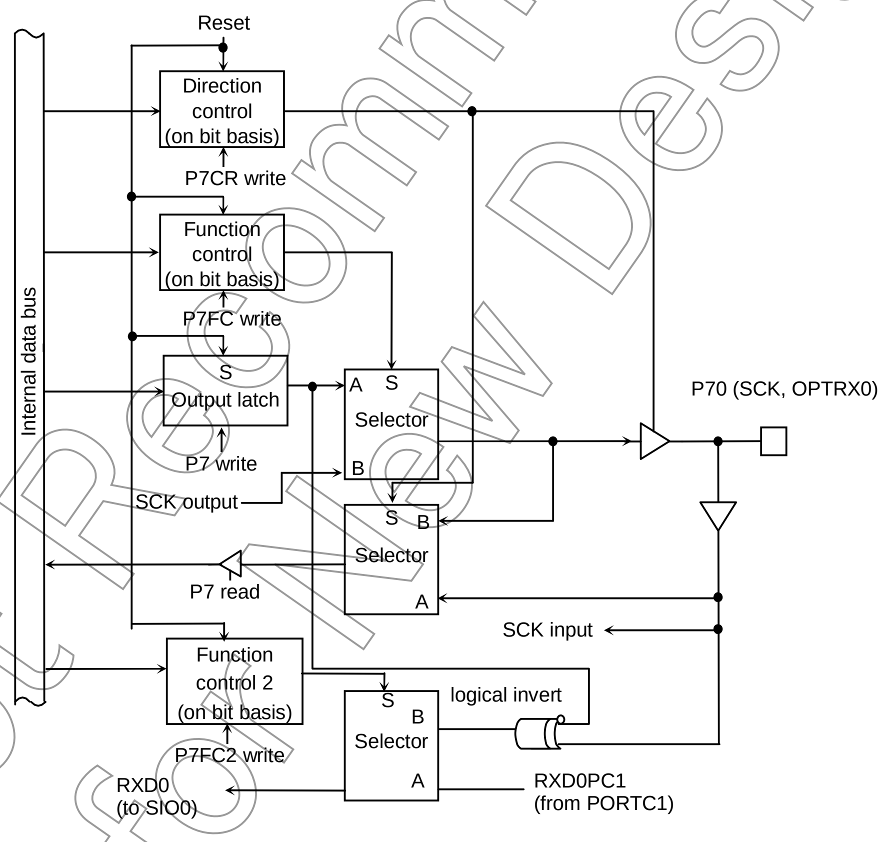


Figure 3.5.9 Port 70

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Port 71 (SO/SDA/OPTTX0)

Port 71 is a general-purpose I/O port. It is also used as SDA (Data input for I²C mode), SO (Data output for SIO mode) for serial bus interface and OPTTX0 (Transmit output for IrDA mode of SIO0).

Used as OPTTX0, it is possible to logical invert by P7<P71> = 0.

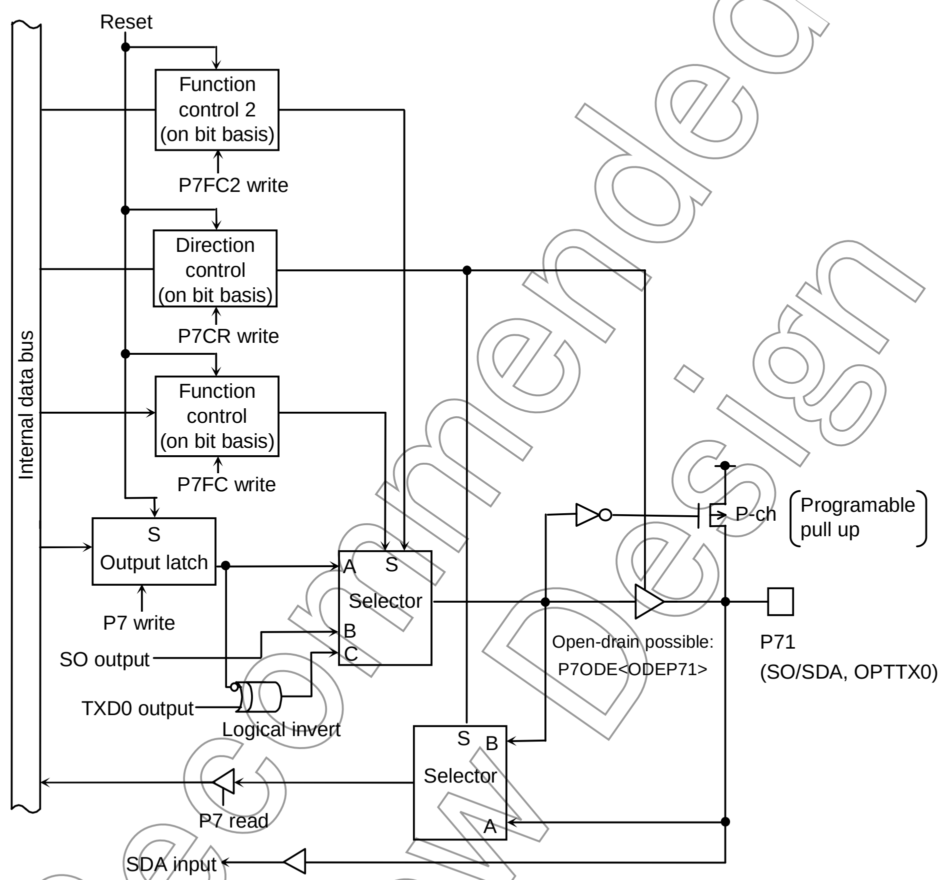


Figure 3.5.10 Port 71

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Port 72 (SI/SCL)

Port 72 is a general-purpose I/O port. It is also used as SI (Data input for SIO mode), SCL (Clock input/output for I²C mode) for serial bus interface and input for release hard protect.

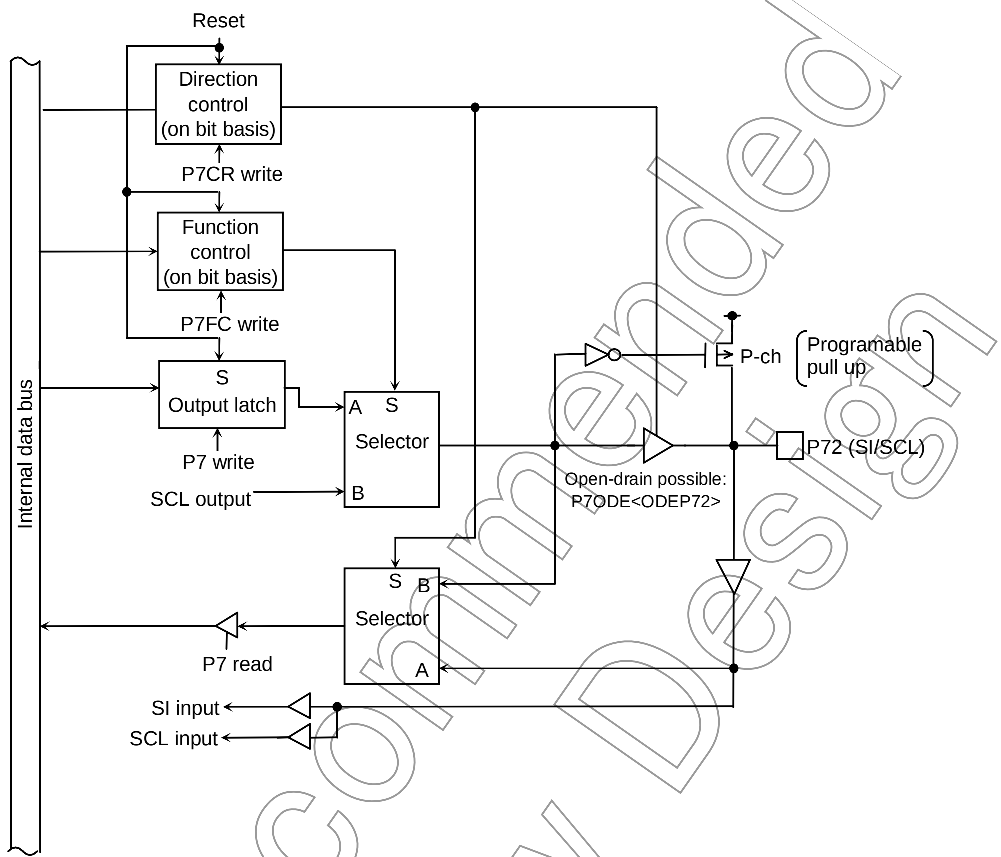


Figure 3.5.11 Port 72

Not Recommended for New Design

Downloaded from Arrow.com.

Port 7 Register

P7 (0013H)

	7	6	5	4	3	2	1	0
Bit symbol						P72	P71	P70
Read/Write						R/W		
After reset						Data from external port (Output latch register is set to “1”.)		
Function						0(Output latch register) : Pull-up resistor ON 1(Output latch register) : Pull-up resistor OFF		–

Port 7 Control Register

P7CR (0016H)

	7	6	5	4	3	2	1	0
Bit symbol						P72C	P71C	P70C
Read/Write						W		
After reset						0	0	0
Function						0: Input 1: Output		

Port 7 Function Register

P7FC (0017H)

	7	6	5	4	3	2	1	0
Bit symbol						P72F	P71F	P70F
Read/Write						W		
After reset						0	0	0
Function						0: Port 1: SCL output	0: Port 1: SDA/SO output	0: Port 1: SCK output

Port 7 Function Register 2

P7FC2 (001CH)

	7	6	5	4	3	2	1	0
Bit symbol						–	P71F2	P70F2
Read/Write						W		
After reset						0	0	0
Function						Always write 0	0: <P71F> 1: OPTTX0 output	SIO0 RXD pin select 0: RXD0 (PC1) 1: OPTRX0 (P70)

Port 7 ODE Register

P7ODE (001FH)

	7	6	5	4	3	2	1	0
Bit symbol						ODEP72	ODEP71	
Read/Write						W		
After reset						0	0	
Function						0: 3 states 1: Open drain		

Note: Read-modify-write is prohibited for P7CR, P7FC, P7FC2 and P7ODE.

Figure 3.5.12 Register for Port 7

Downloaded from Arrow.com.

3.5.6 Port 8 (P80 to P87)

Port 8 is an 8-bit input port and can also be used as the analog input pins for the internal AD converter. P83 can also be used as ADTRG pin for the AD converter.

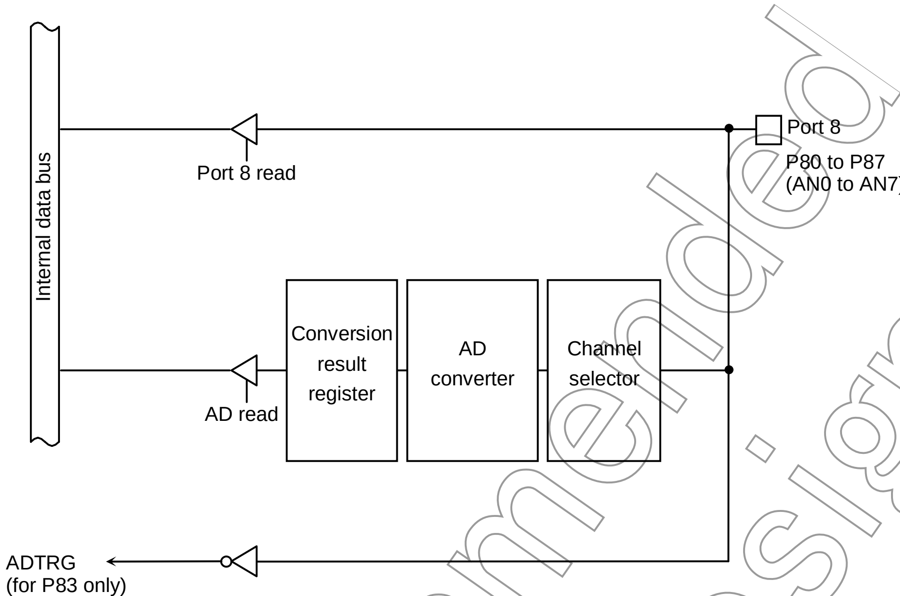


Figure 3.5.13 Port 8

Port 8 Register

P8 (0018H)		7	6	5	4	3	2	1	0
	Bit symbol	P87	P86	P85	P84	P83	P82	P81	P80
	Read/Write	R							
	After reset	Data from external port							

Note: The input channel selection of AD converter and the permission of ADTRG input are set by AD converter mode register ADMOD1.

Figure 3.5.14 Register for Port 8

Not Recommended for New Design

Downloaded from Arrow.com.

3.5.7 Port B (PB0 to PB6)

Port B0 to PB6 is a 7-bit general-purpose I/O port. Each bit can be set individually for input or output. Resetting sets port B to be an input port.

In addition to functioning as a general-purpose I/O port, port B0 has clock input terminal TA0IN of 8-bit timer 0, and port B1, B2 each has facility of 8-bit timer listing TA1OUT, TA3OUT terminal. And, port B3 to B6 has each external interruption input facility of INT0 to INT3. Edge selection of external interruption is establishes by IIMC register in the interrupt controller.

Timer output function and external interrupt function can be enabled by writing 1 to the corresponding bits in the port B function register (PBFC). Resetting resets all bits of the registers PBCR and PBFC to 0, and sets all bits to be input ports.

(1) PB0 to PB2

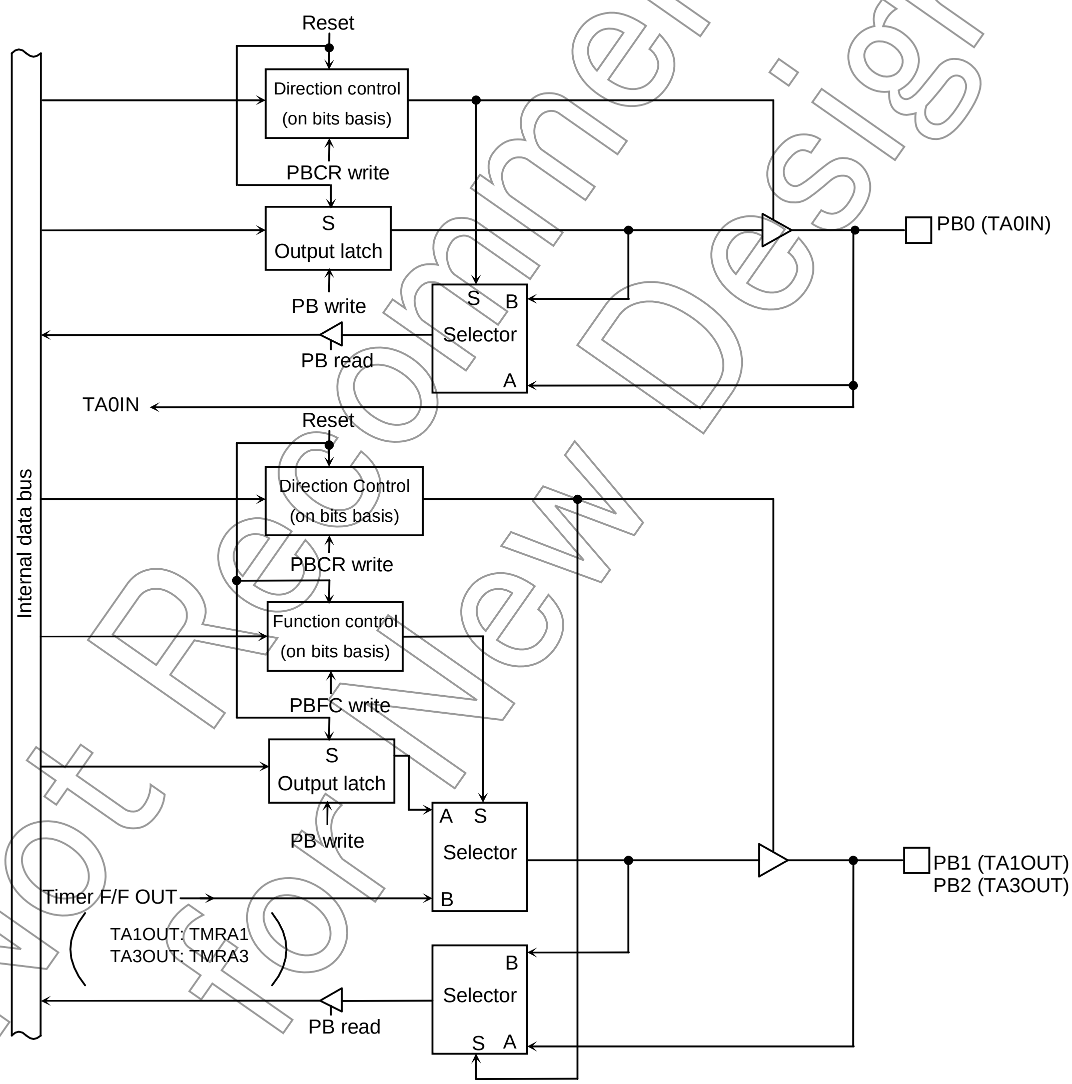


Figure 3.5.15 Port B0 to B2

Downloaded from Arrow.com.

(2) PB3 (INT0), PB4 (INT1) to PB6 (INT3)

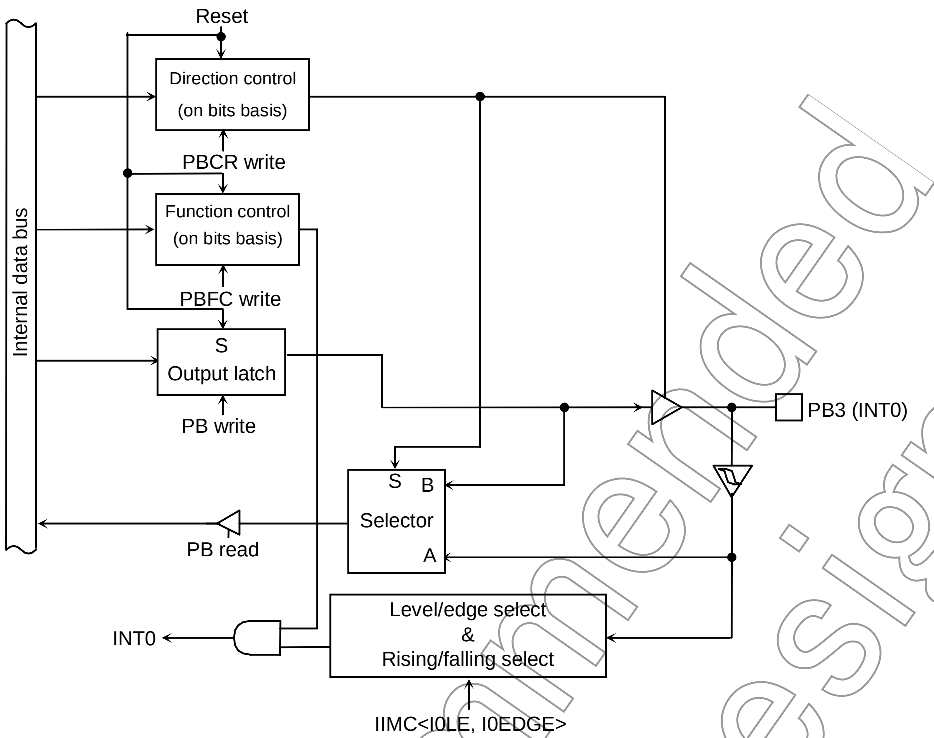


Figure 3.5.16 Port B3

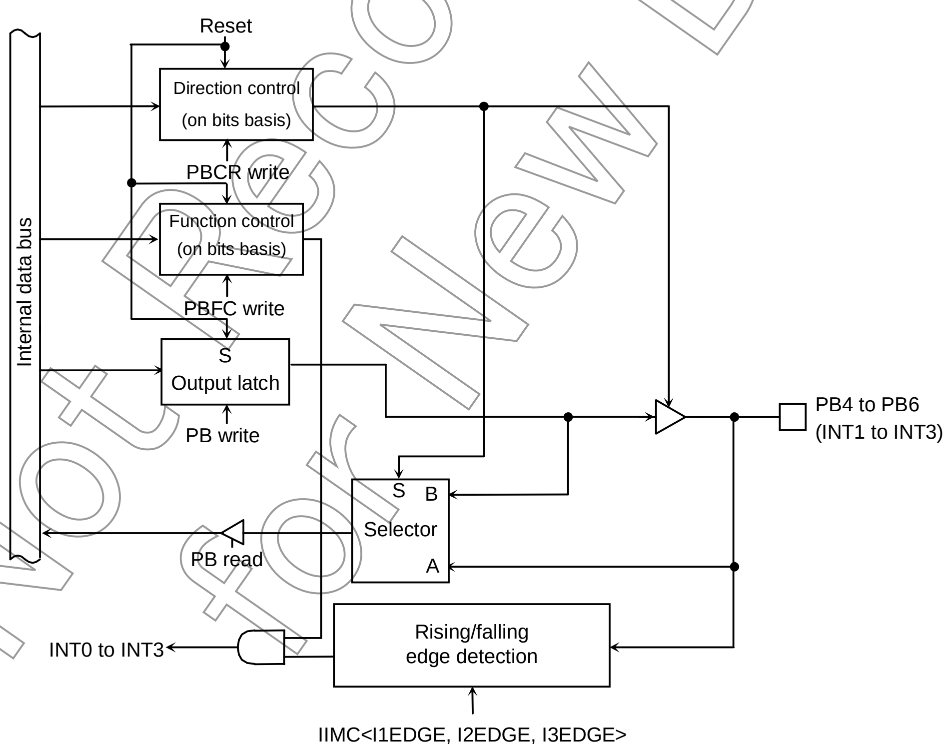


Figure 3.5.17 Port B4 to B6

Not Recommended for New Design

Downloaded from Arrow.com.

Port B Register

PB (0022H)		7	6	5	4	3	2	1	0
	Bit symbol		PB6	PB5	PB4	PB3	PB2	PB1	PB0
	Read/Write		R/W						
	After reset		Data from external port (Output latch register is set to "1".)						

Port B Control Register

PBCR (0024H)		7	6	5	4	3	2	1	0
	Bit symbol		PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
	Read/Write		W						
	After reset		0	0	0	0	0	0	0
	Function		0: Input 1: Output						

Port B Function Register

PBFC (0025H)		7	6	5	4	3	2	1	0
	Bit symbol		PB6F	PB5F	PB4F	PB3F	PB2F	PB1F	
	Read/Write		W						
	After reset		0	0	0	0	0	0	
	Function		0: Port 1: INT3	0: Port 1: INT2	0: Port 1: INT1	0: Port 1: INT0	0: Port 1: TA3OUT	0: Port 1: TA1OUT	

Note 1: Read-Modify-Write is prohibited for the registers PBCR and PBFC.

Note 2: PB0/TA0IN pin does not have a register changing PORT/FUNCTION.
For example, when it is used as an input port, the input signal is inputted to 8-bit timer.

Figure 3.5.18 Register for Port B

Not Recommended for New Design

Downloaded from Arrow.com.

3.5.8 Port C (PC0 to PC5)

Port C0 to C5 are 6-bit general-purpose I/O ports. Each bit can be set individually for input or output. Resetting sets PC0 to PC5 to be an input ports. It also sets all bits of the output latch register to 1.

In addition to functioning as general-purpose I/O port pins, PC0 to PC5 can also function as the I/O for serial channels 0 and 1. A pin can be enabled for I/O by writing 1 to the corresponding bit of the port C function register (PCFC).

Resetting resets all bits of the registers PCCR and PCFC to 0 and sets all pins to be input ports .

(1) Port C0, C3 (TXD0/TXD1)

As well as functioning as I/O port pins, port C0 and C3 can also function as serial channel TXD output pins. In case of use TXD0/TXD1, it is possible to logical invert by setting the register PC<PC0, 3>.

And port C0 to C3 have a programmable open-drain function which can be controlled by the register PCODE<ODEPC0, 3>.

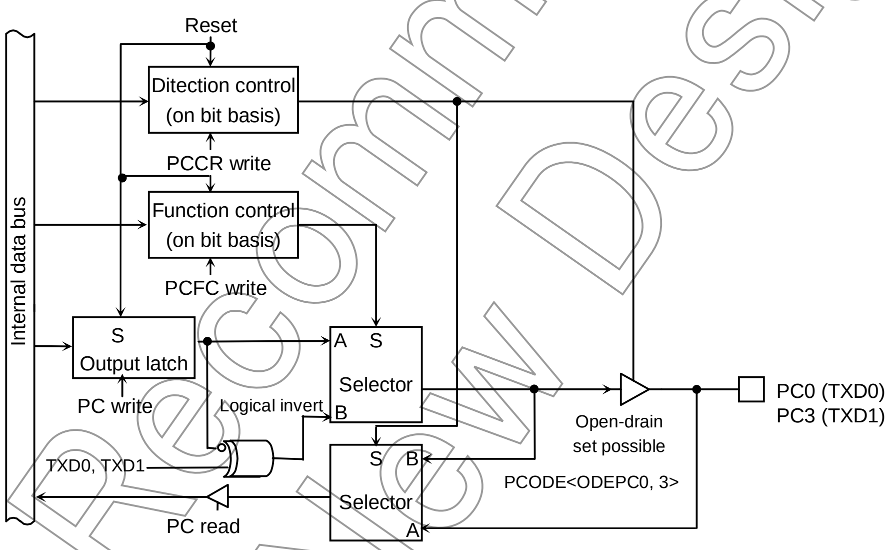


Figure 3.5.19 Port C0 and C3

Downloaded from Arrow.com.

(2) Port C1, C4 (RXD0, RXD1)

Port C1 and C4 are I/O port pins and can also is used as RXD input for the serial channels. In case of use RXD0/RXD1, it is possible to logical invert by setting the register PC<PC1, 4>.

And input data of SIO0 can be select from RXD0/PC1 pin or OPTRX0/P70 by setting the register PCFC2<P70F2>.

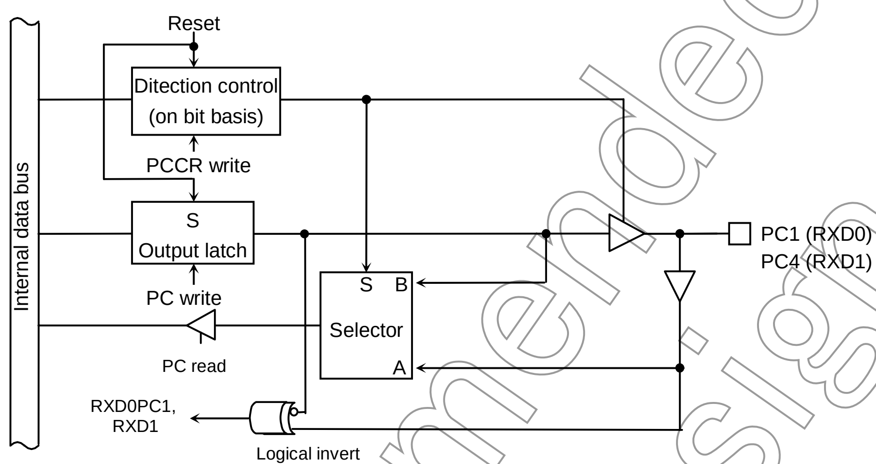


Figure 3.5.20 Port C1 and C4

(3) Port C2 ($\overline{CTS0}$, SCLK0), C5 ($\overline{CTS1}$, SCLK1)

Port C2 and C4 are I/O port pins and can also is used as $\overline{CTS}$ input or SCLK input/output for the serial channels. In case of use $\overline{CTS}$, SCLK, it is possible to logical invert by setting the register PC<PC2, 5>.

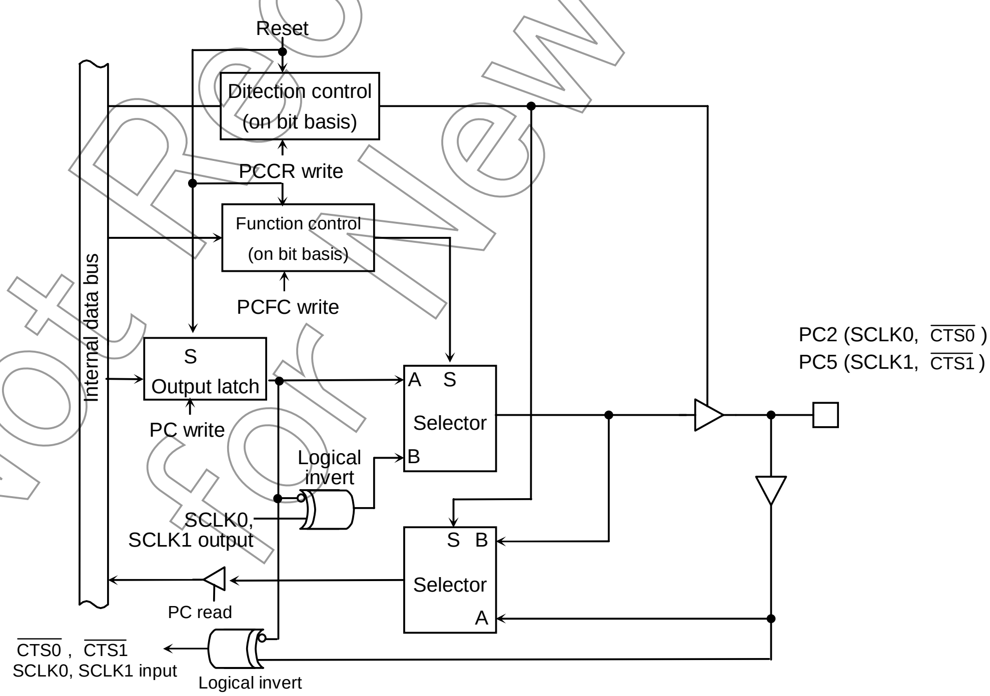


Figure 3.5.21 Port C2 and C5

Downloaded from Arrow.com.

Not Recommended for New Design

Port C Register

PC (0023H)		7	6	5	4	3	2	1	0
	Bit symbol			PC5	PC4	PC3	PC2	PC1	PC0
	Read/Write			R/W					
	After reset			Data from external port (Output latch register is set to “1”.)					

Port C Control Register

PCCR (0026H)		7	6	5	4	3	2	1	0
	Bit symbol			PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
	Read/Write			W					
	After reset			0	0	0	0	0	0
	Function			0: Input 1: Output					

Port C Functon Register

PCFC (0027H)		7	6	5	4	3	2	1	0
	Bit symbol			PC5F		PC3F	PC2F		PC0F
	Read/Write			W		W	W		W
	After reset			0		0	0		0
	Function			0: Port 1: SCLK1 output		0: Port 1: TXD1	0: Port 1: SCLK0 output		0: Port 1: TXD0

Port C ODE Register

PCODE (0028H)		7	6	5	4	3	2	1	0
	Bit symbol					ODEPC3			ODEPC0
	Read/Write					W			W
	After reset					0			0
	Function					TXD1 0: CMOS 1: Open drain			TXD0 0: CMOS 1: Open drain

Note 1: Read-modify-write is prohibited for the registers PCCR, PCFC and PCODE.

Note 2: PC1/RXD0, PC4/RXD1 pins do not have a register changing PORT/FUNCTION. For example, when it is used as an input port, the input signal is inputted to SIO as the cereal receive data.

Figure 3.5.22 Register for Port C

Not Recommended for New Design

Downloaded from Arrow.com.

3.5.9 Port D (PD0 to PD7)

Port D is an 8-bit output port. Resetting sets the output latch PD to 1, and PD5 to PD7 pin output 1.

In addition to functioning as output port, port D also function as output pin for output pin for internal clock (SCOUT), output pin for RTC alarm ($\overline{\text{ALARM}}$) and output pin for melody/alarm generator (MLDALM, $\overline{\text{MLDALM}}$). Above setting is used the function register PDFC.

Only PD6 has two output functions which $\overline{\text{ALARM}}$ and $\overline{\text{MLDALM}}$. This selection is used PD<PD6>. Resetting resets the function register PDFC to 0, and sets all ports to output ports.

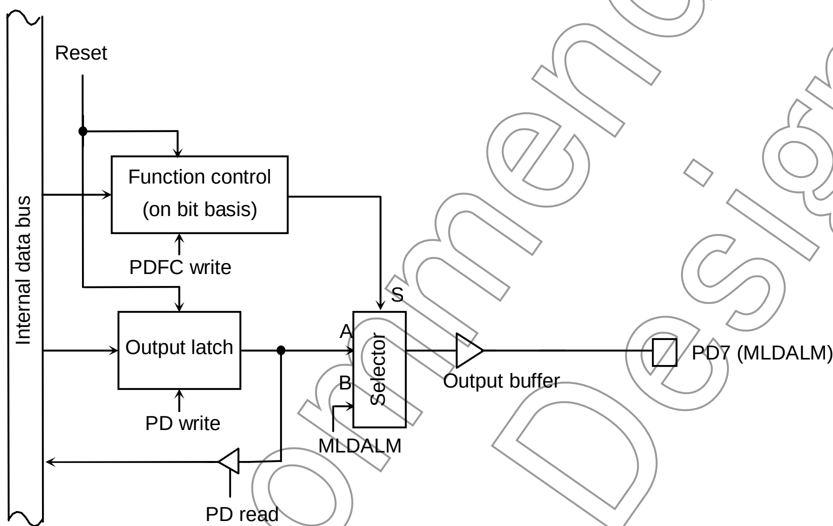


Figure 3.5.23 Port D

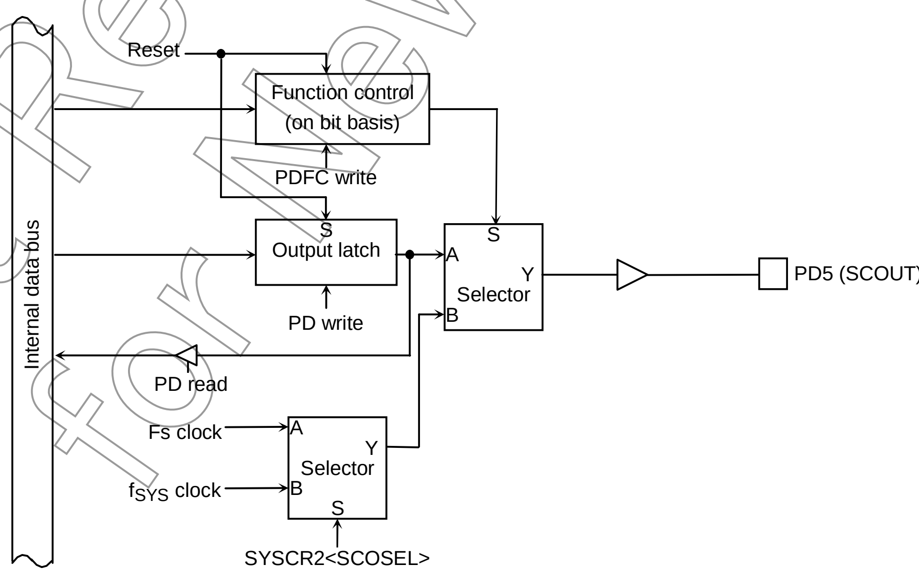


Figure 3.5.24 Port D

Not Recommended for New Design

Downloaded from Arrow.com.

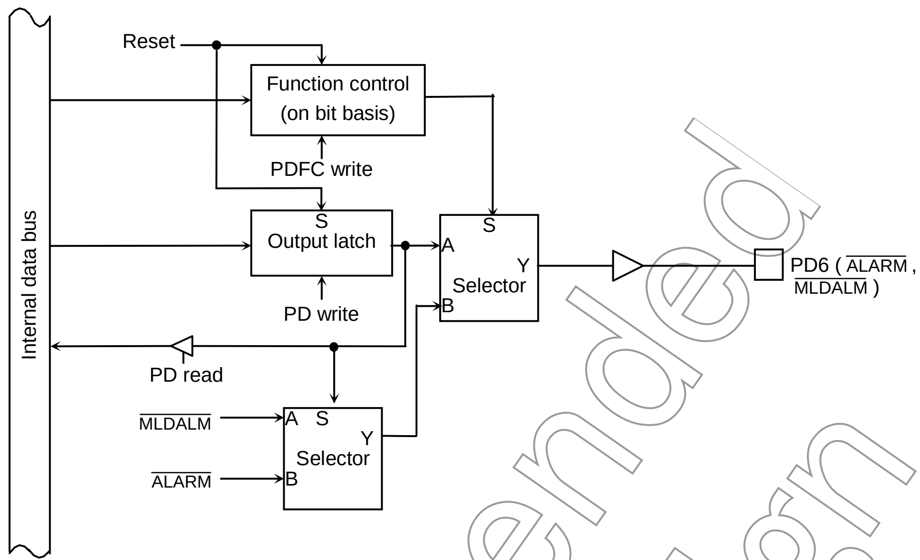


Figure 3.5.25 Port D

Port D Register

PD (0029H)		7	6	5	4	3	2	1	0
	Bit symbol	PD7	PD6	PD5					
	Read/Write	R/W							
	After reset	1	1	1					

Port D Function Register

PDFC (002AH)		7	6	5	4	3	2	1	0
	Bit symbol	PD7F	PD6F	PD5F					
	Read/Write	W							
	After reset	0	0	0					
	Function	0: Port 1: MLDALM	0: Port 1: ALARM at <PD6>=1 1: MLDALM at <PD6>=0	0: Port 1: SCOUT					

Note: Read-modify-write is prohibited for the registers PDFC.

Figure 3.5.26 Register for Port D

Not Recommended for New Design

Downloaded from Arrow.com.

3.5.10 Port Z (PZ2 to PZ3)

Port Z is the 2-bit general-purpose I/O port. I/O is set using control register PZCR and PZFC. Resetting resets all bits of the output latch PZ to 1.

In addition to functioning as a general-purpose I/O port, port Z also functions as output for the CPU’s control/status signal.

Resetting initializes PZ2 and PZ3 pins to input mode with pull-up resistor.

When the PZ<RDE> register clearing to 0,outputs the $\overline{\text{RD}}$ strobe (used for the peused static RAM) of the $\overline{\text{RD}}$ pin even when the internal addressed.

If the <RDE> remains 1, the $\overline{\text{RD}}$ strobe signal is output only when the external address are is accessed.

Not Recommended for New Design

Downloaded from Arrow.com.

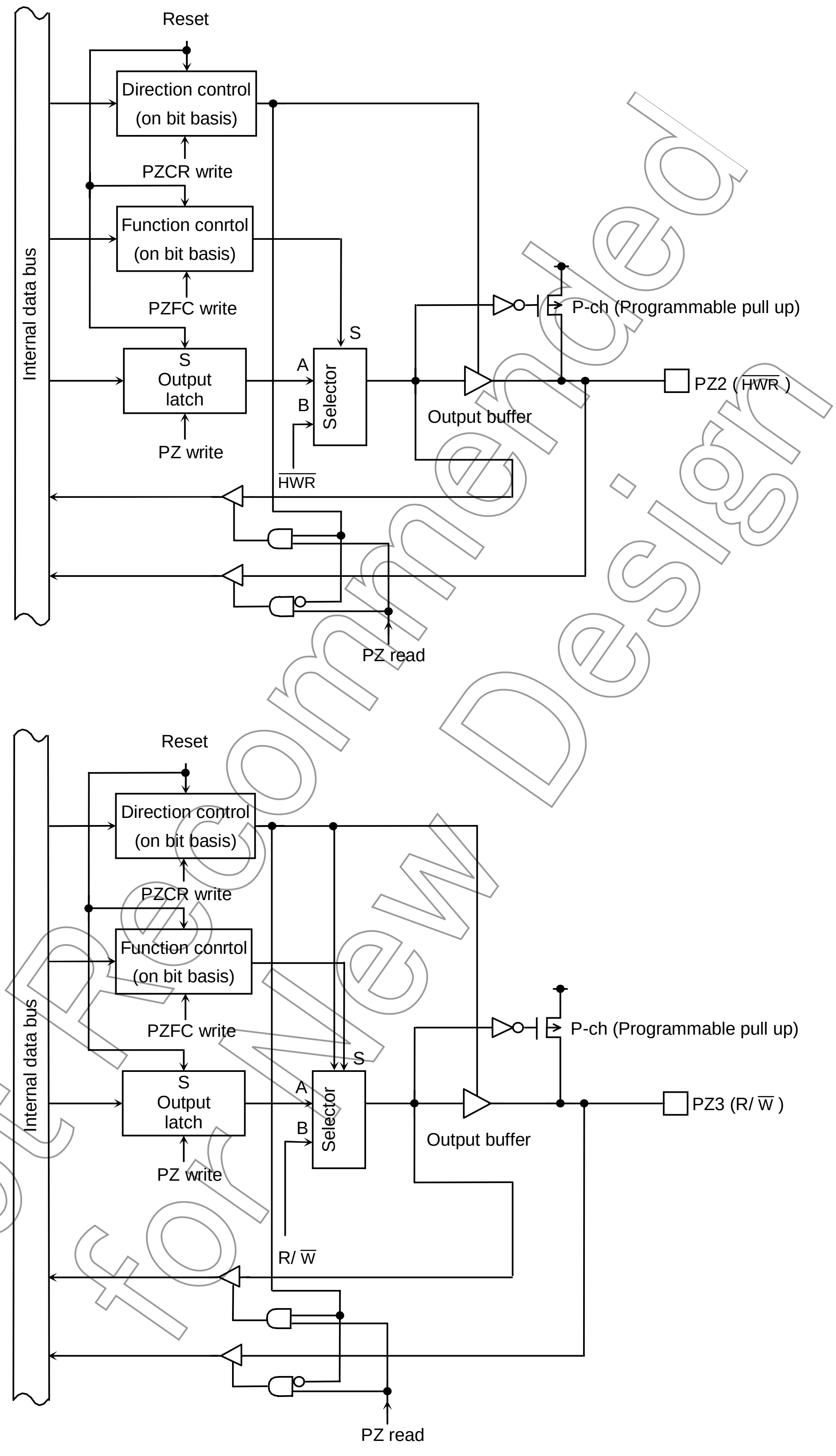


Figure 3.5.27 Port Z (PZ2, PZ3)

Downloaded from Arrow.com.

Port Z Register

		7	6	5	4	3	2	1	0
PZ (007DH)	Bit symbol					PZ3	PZ2		RDE
	Read/Write					R/W			R/W
	After reset					Data from external port (Output latch register is set to "1".)			1
	Function					0(Output latch register) : Pull-up resistor OFF 1(Output latch register) : Pull-up resistor ON			

Port Z Control Register

		7	6	5	4	3	2	1	0
PZCR (007EH)	Bit symbol					PZ3C	PZ2C		
	Read/Write					W			
	After reset					0	0		
	Function					0: Input 1: Output			

Port Z Function Register

		7	6	5	4	3	2	1	0
PZFC (007FH)	Bit symbol	–				PZ3F	PZ2F		
	Read/Write	W				W			
	After reset	0				0	0		
	Function	Always write 0				0: Port 1: R/$\overline{W}$	0: Port 1: $\overline{HWR}$		

$\overline{HWR}$ setting

PZFC<PZ2F>	1
PZCR<PZ2C>	1

Note 1: Read-modify-write is prohibited for registers PZCR and PZFC.

Note 2: When port Z is used in input mode, the PZ register controls the built-in pull-up resistor. Read-modify-write is prohibited in input mode or I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Figure 3.5.28 Port Register for Port Z

Not Recommended for New Design

Downloaded from Arrow.com.

3.6 Chip Select/Wait Controller

On the TM91C824, four user-specifiable address areas (CS0 to CS3) can be set. The data bus width and the number of waits can be set independently for each address area (CS0 to CS3 and others).

The pins $\overline{CS0}$ to $\overline{CS3}$ (which can also function as port pins P60 to P63) are the respective output pins for the areas CS0 to CS3. When the CPU specifies an address in one of these areas, the corresponding $\overline{CS0}$ to $\overline{CS3}$ pin outputs the chip select signal for the specified address area (in ROM or SRAM). However, in order for the chip select signal to be output, the port 6 function register P6FC must be set.

$\overline{CS2A}$ To $\overline{CS2E}$ (CS pin except $\overline{CS0}$ to $\overline{CS3}$) are made by MMU.

These pins are $\overline{CS}$ pin that area and BANK value is fixed without concern in setting of CS/WAIT controller.

The areas CS0 to CS3 are defined by the values in the memory start address registers MSAR0 to MSAR3 and the memory address mask registers MAMR0 to MAMR3.

The chip select/wait control registers B0CS to B3CS and BEXCS should be used to specify the master enable/disable status the data bus width and the number of waits for each address area.

The input pin controlling these states is the bus wait request pin ($\overline{WAIT}$).

3.6.1 Specifying an Address Area

The CS0 to CS3 address areas are specified using the start address registers (MSAR0 to MSAR3) and memory address mask registers (MAMR0 to MAMR3).

At each bus cycle, a compare operation is performed to determine if the address on the specified a location in the CS0 to CS3 area. If the result of the comparison is a match, this indicates an access to the corresponding CS area. In this case, the $\overline{CS0}$ to $\overline{CS3}$ pin outputs the chip select signal and the bus cycle operates in accordance with the settings in chip select/wait control register B0CS to B3CS. (See 3.6.2 “Chip Select/Wait Control Registers”.)

Not Recommended for New Design

Downloaded from Arrow.com.

(1) Memory start address registers

Figure 3.6.1 shows the memory start address registers. The memory start address registers MSAR0 to MSAR3 set the start addresses for the CS0 to CS3 areas. Set the upper 8 bits (A23 to A16) of the start address in <S23:16>. The lower 16 bits of the start address (A15 to A0) are permanently set to 0. Accordingly, the start address can only be set in 64-Kbyte increments, starting from 000000H. Figure 3.6.2 shows the relationship between the start address and the start address register value.

Memory Start Address Registers (for areas CS0 to CS3)

MSAR0 (00C8H) / MSAR1 (00CAH)
MSAR2 (00CCH) / MSAR3 (00CEH)

	7	6	5	4	3	2	1	0
Bit symbol	S23	S22	S21	S20	S19	S18	S17	S16
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Determines A23 to A16 of start address.							

→ Sets start addresses for areas CS0 to CS3.

Figure 3.6.1 Memory Start Address Register

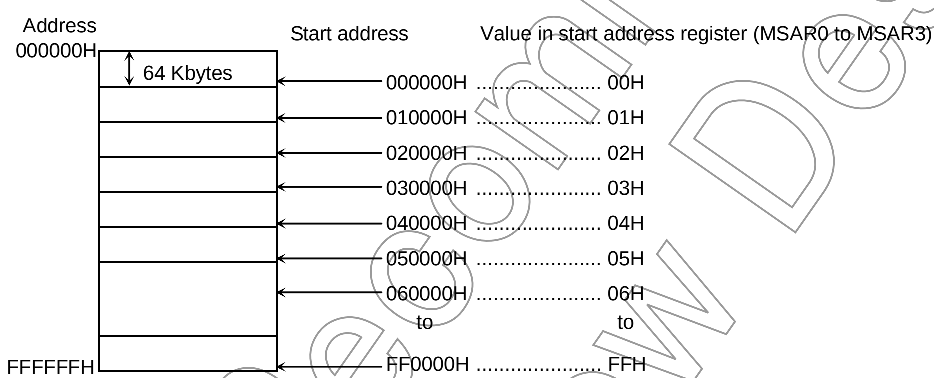


Figure 3.6.2 Relationship between Start Address and Start Address Register Value

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Memory address mask registers

Figure 3.6.3 shows the memory address mask registers. Memory address mask registers MAMR0 to MAMR3 are used to set the size of the CS0 to CS3 areas by specifying a mask for each bit of the start address set in memory start address registers MAMR0 to MAMR3. The compare operation used to determine if an address is in the CS0 to CS3 areas is only performed for bus address bits corresponding to bits set to 0 in these registers. Also, the address bits that can be masked by MAMR0 to MAMR3 differ between CS0 to CS3 areas. Accordingly, the size that can be each area is different.

Memory Address Mask Register (for CS0 area)

MAMR0 (00C9H)

	7	6	5	4	3	2	1	0
Bit symbol	V20	V19	V18	V17	V16	V15	V14 to V9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS0 area 0: Used for address compare							

Range of possible settings for CS0 area size: 256 bytes to 2 Mbytes

Memory Address Mask Register (CS1)

MAMR1 (00CBH)

	7	6	5	4	3	2	1	0
Bit symbol	V21	V20	V19	V18	V17	V16	V15 to V9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS1 area 0: Used for address compare							

Range of possible settings for CS1 area size: 256 bytes to 4 Mbytes.

Memory Address Mask Register (CS2, CS3)

MAMR2 (00CDH) / MAMR3 (00CFH)

	7	6	5	4	3	2	1	0
Bit symbol	V22	V21	V20	V19	V18	V17	V16	V15
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS2 or CS3 area 0: Used for address compare							

Range of possible settings for CS2 and CS3 area sizes: 32 Kbytes to 8 Mbytes.

Figure 3.6.3 Memory Address Mask Registers

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Setting memory start addresses and address areas

Figure 3.6.4 show an example of specifying a 64-Kbyte address area starting from 010000H using the CS0 areas.

Set 01H in memory start address register MSAR0<S23:16> (Corresponding to the upper 8 bits of the start address). Next, calculate the difference between the start address and the anticipated end address (01FFFFH). Bits 20 to 8 of the result correspond to the mask value to be set for the CS0 area. Setting this value in memory address mask register MAMR0<V20:8> sets the area size This example sets 07H in MAMR0 to specify a 64-Kbyte area.

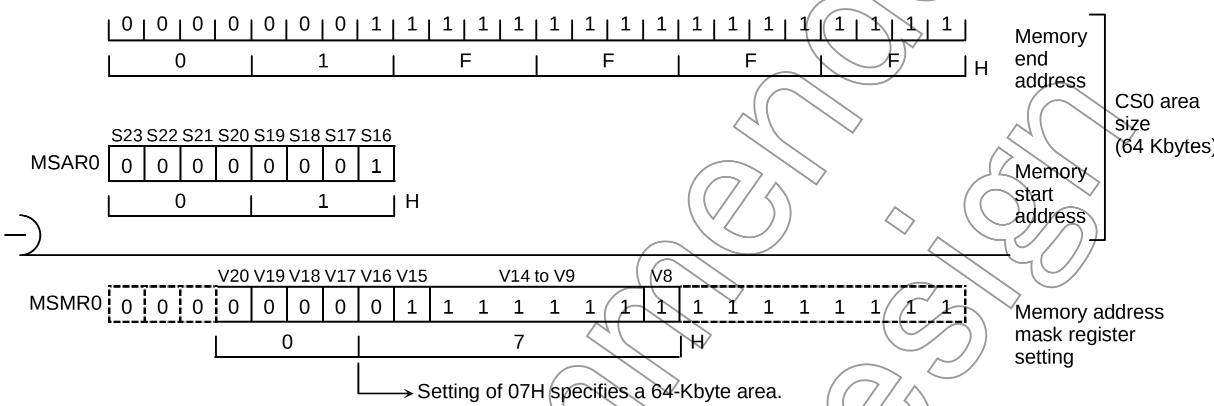


Figure 3.6.4 Example Showing How to Set the CS0 Area

After a reset, MSAR0 to MSAR3 and MAMR0 to MAMR3 are set to FFH. B0CS<B0E>, B1CS<B1E> and B3CS<B3E> are reset to 0. This disabling the CS0, CS1 and CS3 areas. However, as B2CS<B2M> to 0 and B2CS<B2E> to 1, CS2 is enabled from 000FE0H to 000FFFH to 003000H to FFFFFFH in TMP91C824. Also, the bus width and number of waits specified in BEXCS are used for accessing addresses outside the specified CS0 to CS3 area. (See 3.6.2 “Chip Select/Wait Control Registers”.)

Not Recommended for New Design

Downloaded from Arrow.com.

(4) Address area size specification

Table 3.6.1 shows the relationship between CS area and area size. “Δ” indicates areas that cannot be set by memory start address register and address mask register combinations. When setting an area size using a combination indicated by “Δ”, set the start address mask register in the desired steps starting from 000000H.

If the CS2 area is set to 16-Mbytes or if two or more areas overlap, the smaller CS area number has the higher priority.

Example: To set the area size for CS0 to 128 Kbytes:

a. Valid start addresses

000000H
128 Kbytes
020000H
128 Kbytes
040000H
128 Kbytes
060000H
⋮

Any of these addresses may be set as the start address.

b. Invalid start addresses

000000H
64 Kbytes ← This is not an integer multiple of the desired area size setting. Hence, none of these addresses can be set as the start address.
010000H
128 Kbytes
030000H
128 Kbytes
050000H
⋮

Table 3.6.1 Valid Area Sizes for Each CS Area

CS Area \ Size (Bytes)	256	512	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M
CS0	○	○	○	○	Δ	Δ	Δ	Δ	Δ		
CS1	○	○		○	Δ	Δ	Δ	Δ	Δ	Δ	
CS2			○	○	Δ	Δ	Δ	Δ	Δ	Δ	Δ
CS3			○	○	Δ	Δ	Δ	Δ	Δ	Δ	Δ

Note: “Δ” indicates areas that cannot be set by memory start address register and address mask register combinations.

3.6.2 Chip Select/Wait Control Registers

Figure 3.6.5 lists the chip select/wait control registers.

The master enable/disable, chip select output waveform, data bus width and number of wait states for each address area (CS0 to CS3 and others) are set in their respective chip select/wait control registers, B0CS to B3CS and BEXCS.

Not Recommended for New Design

Downloaded from Arrow.com.

Register		7	6	5	4	3	2	1	0
B0CS (00C0H)	Bit symbol	B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: 10: } Don't care 11:		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
B1CS (00C1H)	Bit symbol	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: 10: } Don't care 11:		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
B2CS (00C2H)	Bit symbol	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
	Read/Write	W							
	After reset	1	0	0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Functions	0: Disable 1: Enable	CS2 area selection 0: 16-Mbyte area 1: CS area	Chip select output waveform selection 00: For ROM/SRAM 01: 10: } Don't care 11:		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
B3CS (00C3H)	Bit symbol	B3E		B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Functions	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: 10: } Don't care 11:		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
BEXCS (00C7H)	Bit symbol					BEXBUS	BEXW2	BEXW1	BEXW0
	Read/Write					W			
	After reset					0	0	0	0
Read-modify-write instructions are prohibited.	Functions					Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	

Master enable bit

0	Enable
1	Disable

CS2 area selection

0	16-Mbyte area
1	Specified address area

Chip select output waveform selection

00	For ROM/SRAM
01 10 11	Don't care

Number of address area waits
(See 3.6.2, (3) Wait control.)

Data bus width selection

0	16-bit data bus
1	8-bit data bus

Figure 3.6.5 Chip Select/Wait Control Registers

Not Recommended for New Design

Downloaded from Arrow.com.

(1) Master enable bits

Bit 7 (<B0E>, <B1E>, <B2E> or <B3E>) of a chip select/wait control register is the master bit which is used to enable or disable settings for the corresponding address area. Writing 1 to this bit enables the settings. Reset disables (Sets to 0) <B0E>, <B1E> and <B3E>, and enabled (Sets to 1) <B2E>. This enables area CS2 only.

(2) Data bus width selection

Bit 3 (<B0BUS>, <B1BUS>, <B2BUS>, <B3BUS> or <BEXBUS>) of a chip select/wait control register specifies the width of the data bus. This bit should be set to 0 when memory is to be accessed using a 16-bit data bus and to 1 when an 8-bit data bus is to be used.

This process of changing the data bus width according to the address being accessed is known as dynamic bus sizing. For details of this bus operation see Table 3.6.2.

Table 3.6.2 Dynamic Bus Sizing

Operand Data Bus Width	Operand Start Address	Memory Data Bus Width	CPU Address	CPU Data	
				D15 to D8	D7 to D0
8 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
		16 bits	2n + 0	xxxxx	b7 to b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
		16 bits	2n + 1	b7 to b0	xxxxx
16 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
		16 bits	2n + 0	b15 to b8	b7 to b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	xxxxx	b15 to b8
32 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
			2n + 2	xxxxx	b23 to b16
			2n + 3	xxxxx	b31 to b24
		16 bits	2n + 0	b15 to b8	b7 to b0
			2n + 2	b31 to b24	b23 to b16
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
			2n + 3	xxxxx	b23 to b16
			2n + 4	xxxxx	b31 to b24
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	b23 to b16	b15 to b8
			2n + 4	xxxxx	b31 to b24

Note: xxxxx indicates that the input data from these bits are ignored during a read. During a write, indicates that the bus for these bits goes too high impedance; also, that the write strobe signal for the bus remains inactive.

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Wait control

Bits 0 to 2 (<B0W0:2>, <B1W0:2>, <B2W0:2>, <B3W0:2>, <BEXW0:2>) of a chip select/wait control register specify the number of waits that are to be inserted when the corresponding memory area is accessed.

The following types of wait operation can be specified using these bits. Bit settings other than those listed in the table should not be made.

Table 3.6.3 Wait Operation Settings

<BxW2:0>	Number of Waits	Wait Operation
000	2	Inserts a wait of 2 states, irrespective of the $\overline{\text{WAIT}}$ pin state.
001	1	Inserts a wait of 1 state, irrespective of the $\overline{\text{WAIT}}$ pin state.
010	(1 + N)	Samples the state of the $\overline{\text{WAIT}}$ pin after inserting a wait of 1 state. If the $\overline{\text{WAIT}}$ pin is low, the waits continue and the bus cycle is extended until the pin goes high.
011	0	Ends the bus cycle without a wait, regardless of the $\overline{\text{WAIT}}$ pin state.
100	Reserved	Invalid setting
101	3	Inserts a wait of 3 states, irrespective of the $\overline{\text{WAIT}}$ pin state.
110	4	Inserts a wait of 4 states, irrespective of the $\overline{\text{WAIT}}$ pin state.
111	8	Inserts a wait of 8 states, irrespective of the $\overline{\text{WAIT}}$ pin state.

A reset sets these bits to 000 (2 waits).

(4) Bus width and wait control for an area other than CS0 to CS3

The chip select/wait control register BEXCS controls the bus width and number of waits when memory locations which are not in one of the four user-specified address areas (CS0 to CS3) are accessed. The BEXCS register settings are always enabled for areas other than CS0 to CS3.

(5) Selecting 16-Mbyte area/specified address area

Setting B2CS<B2M> (Bit 6 of the chip select/wait control register for CS2) to 0 designates the 16-Mbyte area 000FE0H to 000FFFH, 003000H to FFFFFFH as the CS2 area. Setting B2CS<B2M> to 1 designates the address area specified by the start address register MSAR2 and the address mask register MAMR2 as CS2 (e.g., if B2CS<B2M> = 1, CS2 is specified in the same manner as CS0, CS1 and CS3 are).

A reset clears this bit to 0, specifying CS2 as a 16-Mbyte address area.

Not Recommended for New Design

Downloaded from Arrow.com.

(6) Procedure for setting chip select/wait control

When using the chip select/wait control function, set the registers in the following order:

1. Set the memory start address registers MSAR0 to MSAR3.
   Set the start addresses for CS0 to CS3.
2. Set the memory address mask registers MAMR0 to MAMR3.
   Set the sizes of CS0 to CS3.
3. Set the chip select/wait control registers B0CS to B3CS.

   Set the chip select output waveform, data bus width, number of waits and master enable/disable status for $\overline{CS0}$ to $\overline{CS3}$.

   The CS0 to CS3 pins can also function as pins P60 to P63. To output a chip select signal using one of these pins, set the corresponding bit in the port 6 function register P6FC to 1.

   If a CS0 to CS3 address is specified which is actually an internal I/O and RAM area address, the CPU accesses the internal address area and no chip select signal is output on any of the $\overline{CS0}$ to $\overline{CS3}$ pins.

Setting example:

In this example CS0 is set to be the 64-Kbyte area 010000H to 01FFFFH. The bus width is set to 16 bits and the number of waits is set to 0.

MSAR0 = 01H	Start address: 010000H
MAMR0 = 07H	Address area: 64 Kbytes
B0CS = 83H	ROM/SRAM, 16-bit data bus, 0 waits, CS0 area settings enabled.

Not Recommended for New Design

Downloaded from Arrow.com.

3.6.3 Connecting External Memory

Figure 3.6.6 shows an example of how to connect external memory to the TMP91C824.

In this example the ROM is connected using a 16-bit bus. The RAM and I/O are connected using an 8-bit bus.

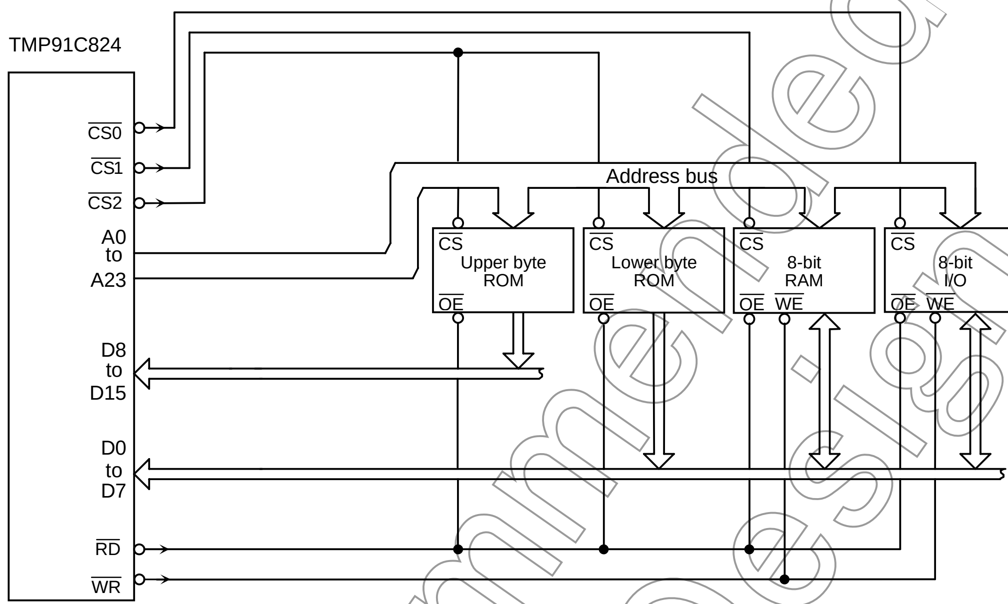


Figure 3.6.6 Example of External Memory Connection
(ROM uses 16-bit bus; RAM and I/O use 8-bit bus.)

A reset clears all bits of the port 6 control register P6CR and the port 6 function register P6FC to 0 and disables output of the CS signal. To output the CS signal, the appropriate bit must be set to 1.

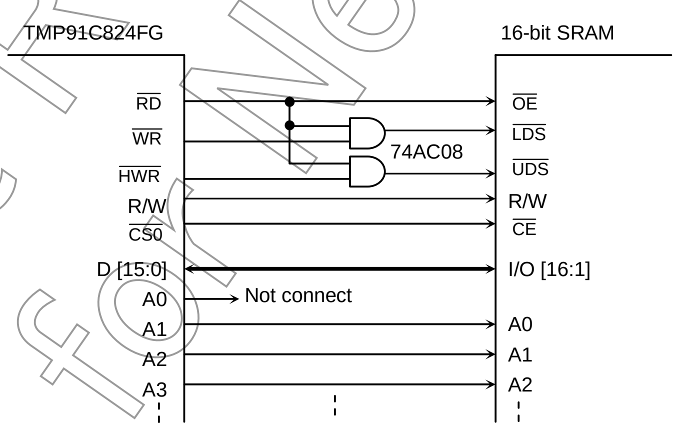


Figure 3.6.7 Example of External Memory Connection (RAM and I/O use 16-bit bus)

Not Recommended for New Design

Downloaded from Arrow.com.

3.7 8-Bit Timers (TMRA)

The TMP91C824 features 4 channel (TMRA0 to TMRA3) built-in 8-bit timers.

These timers are paired into 2 modules: TMRA01 and TMRA23. Each module consists of 2 channels and can operate in any of the following 4 operating modes.

- 8-bit interval timer mode
- 16-bit interval timer mode
- 8-bit programmable square wave pulse generation output mode (PPG: Variable duty cycle with variable period)
- 8-bit pulse width modulation output mode (PWM: Variable duty cycle with constant period)

Figure 3.7.1 to Figure 3.7.2 show block diagrams for TMRA01 and TMRA23.

Each channel consists of an 8-bit up counter, an 8-bit comparator and an 8-bit timer register. In addition, a timer flip-flop and a prescaler are provided for each pair of channels.

The operation mode and timer flip-flop condition are controlled by 5-byte registers.

We call control registers SFRs: Special function registers.

Each of the two modules (TMRA01 and TMRA23) can be operated independently. All modules operate in the same manner; hence only the operation of TMRA01 is explained here.

The contents of this chapter are as follows.

Table 3.7.1 Registers and Pins for Each Module

Module		TMRA01	TMRA23
External pin	Input pin for external clock	TA0IN (shared with PB0)	None
	Output pin for timer flip-flop	TA1OUT (shared with PB1)	TA3OUT (shared with PB2)
SFR (Address)	Timer run register	TA01RUN (0100H)	TA23RUN (0108H)
	Timer register	TA0REG (0102H) TA1REG (0103H)	TA2REG (010AH) TA3REG (010BH)
	Timer mode register	TA01MOD (0104H)	TA23MOD (010CH)
	Timer flip-flop control register	TA1FFCR (0105H)	TA3FFCR (010DH)

Not Recommended for New Design

Downloaded from Arrow.com.

3.7.1 Block Diagrams

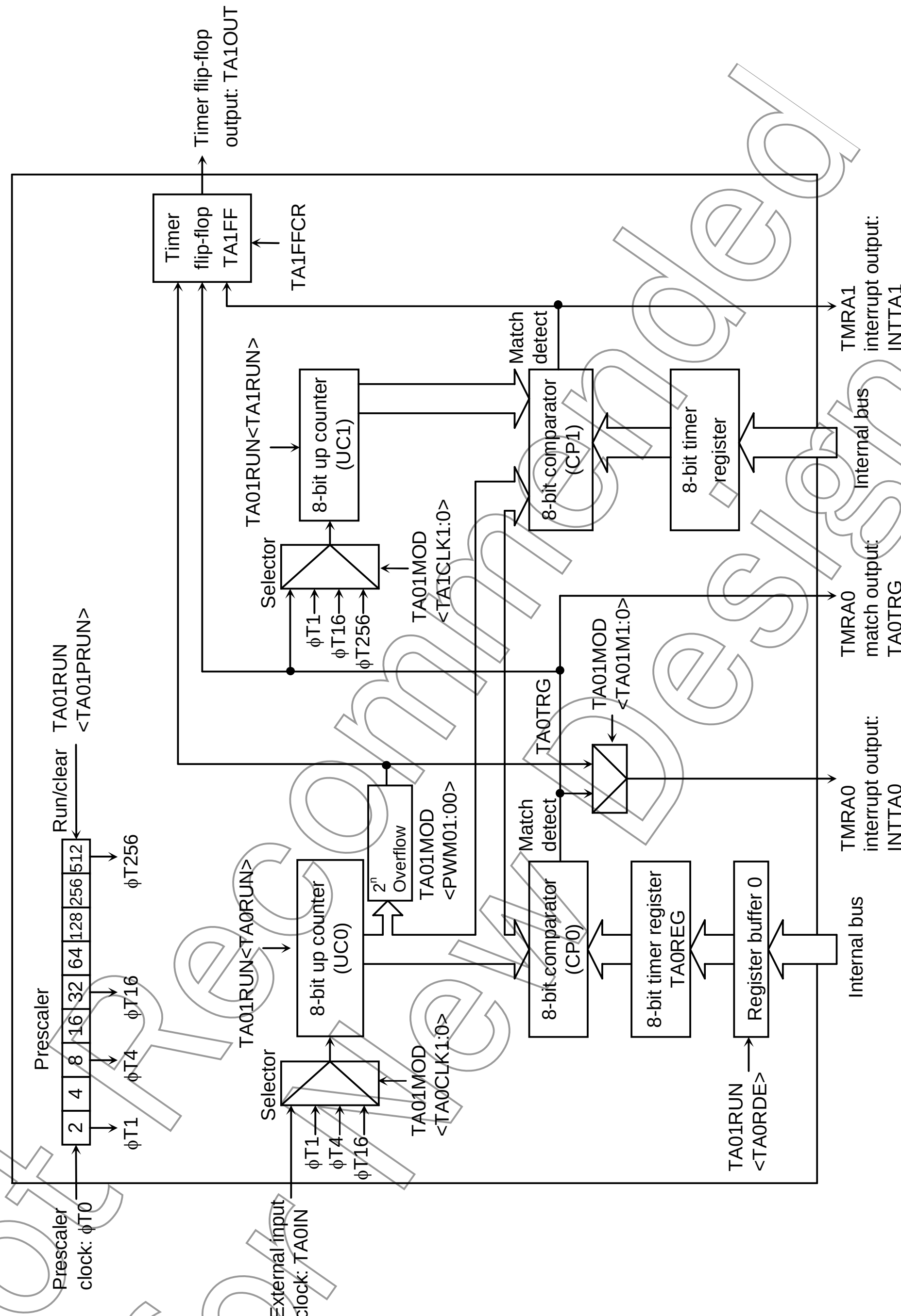


Figure 3.7.1 TMRA01 Block Diagram

Not Recommended for New Design

Downloaded from Arrow.com.

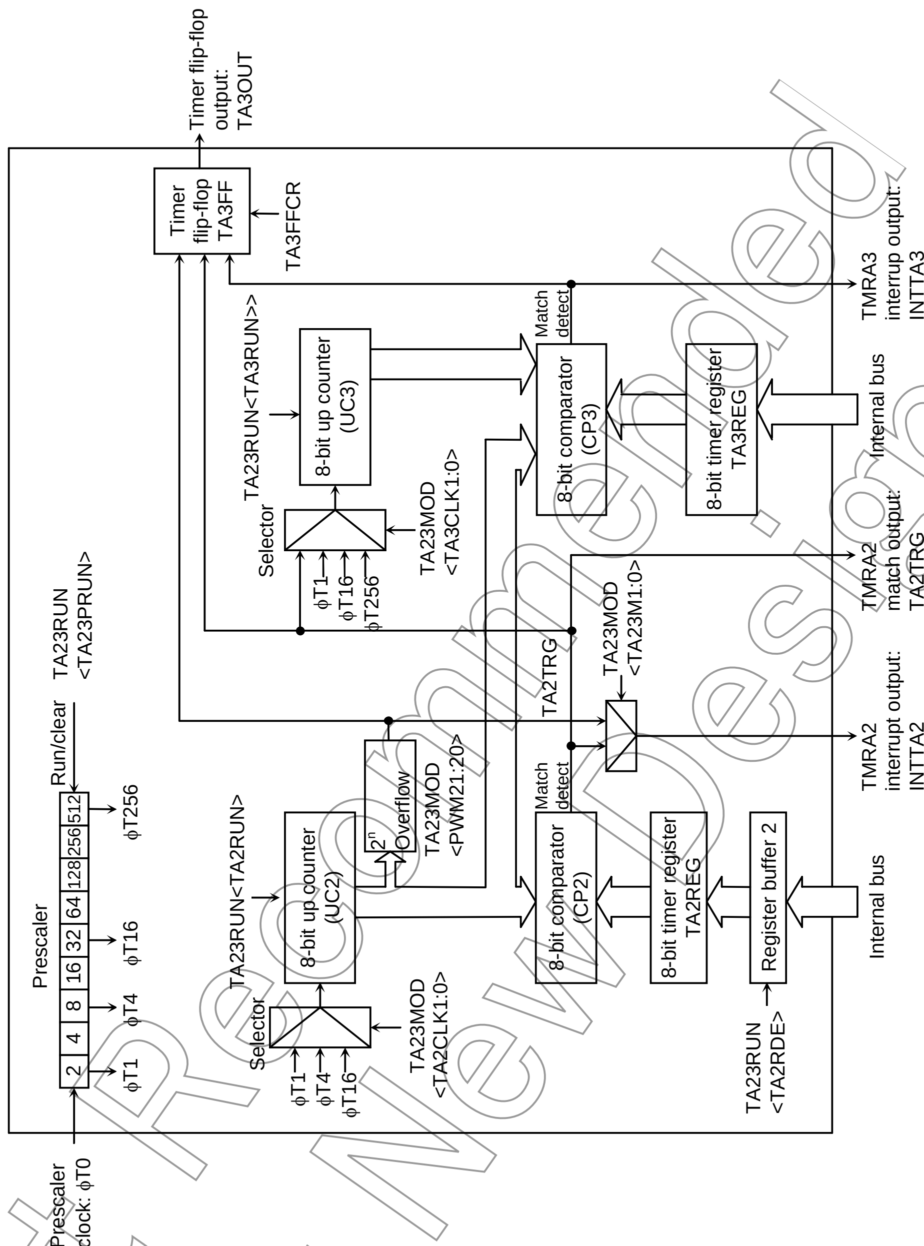


Figure 3.7.2 TMRA23 Block Diagram

Not Recommended for New Design

Downloaded from Arrow.com.

3.7.2 Operation of Each Circuit

(1) Prescalers

A 9-bit prescaler generates the input clock to TMRA01.

The ϕT0 as the input clock to prescaler is a clock divided by 4 which selected using the prescaler clock selection register SYSCR0<PRCK1:0>.

The prescaler's operation can be controlled using TA01RUN<TA01PRUN> in the timer control register. Setting <TA01PRUN> to 1 starts the count; setting <TA01PRUN> to 0 clears the prescaler to zero and stops operation. Table 3.7.2 shows the various prescaler output clock resolutions.

Table 3.7.2 Prescaler Output Clock Resolution

at fc = 33 MHz, fs = 32.768 kHz

System Clock Selection SYSCR1 <SYSCK>	Prescaler Clock Selection SYSCR0 <PRCK1:0>	Gear Value SYSCR1 <GEAR2:0>	Prescaler Output Clock Resolution			
			ϕT1	ϕT4	ϕT16	ϕT256
1 (fs)	00 (f_{FPH})	XXX	2^3/fs (244 μs)	2^5/fs (977 μs)	2^7/fs (3.9 ms)	2^{11}/fs (62.5 ms)
0 (fc)		000 (fc)	2^3/fc (0.2 μs)	2^5/fc (1.0 μs)	2^7/fc (3.9μs)	2^{11}/fc (62.1 μs)
		001 (fc/2)	2^4/fc (0.5 μs)	2^6/fc (1.9 μs)	2^8/fc (7.8 μs)	2^{12}/fc (248.2 μs)
		010 (fc/4)	2^5/fc (1.0 μs)	2^7/fc (3.9 μs)	2^9/fc (15.5 μs)	2^{13}/fc (496.5 μs)
		011 (fc/8)	2^6/fc (1.9 μs)	2^8/fc (7.8 μs)	2^{10}/fc (31.0 μs)	2^{14}/fc (1024 μs)
		100 (fc/16)	2^7/fc (3.9 μs)	2^9/fc (15.5 μs)	2^{11}/fc (62.1 μs)	2^{15}/fc (993 μs)
	10 (fc/16 clock)	XXX	2^7/fc (3.9 μs)	2^9/fc (15.5 μs)	2^{11}/fc (62.1 μs)	2^{15}/fc (993 μs)

xxx: Don't care

(2) Up counters (UC0 and UC1)

These are 8-bit binary counters which count up the input clock pulses for the clock specified by TA01MOD.

The input clock for UC0 is selectable and can be either the external clock input via the TA0IN pin or one of the three internal clocks ϕT1, ϕT4 or ϕT16. The clock setting is specified by the value set in TA01MOD<TA0CLK1:0>.

The input clock for UC1 depends on the operation mode. In 16-bit timer mode, the overflow output from UC0 is used as the input clock. In any mode other than 16-bit timer mode, the input clock is selectable and can either be one of the internal clocks ϕT1, ϕT16 or ϕT256, or the comparator output (The match detection signal) from TMRA0.

For each interval timer the timer operation control register bits TA01RUN<TA0RUN> and TA01RUN<TA1RUN> can be used to stop and clear the up counters and to control their count. A reset clears both up counters, stopping the timers.

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Timer registers (TA0REG and TA1REG)

These are 8-bit registers which can be used to set a time interval. When the value set in the timer register TA0REG or TA1REG matches the value in the corresponding up counter, the comparator match detect signal goes active. If the value set in the timer register is 00H, the signal goes active when the up counter overflows.

The TA0REG are double buffer structure, each of which makes a pair with register buffer.

The setting of the bit TA01RUN<TA0RDE> determines whether TA0REG's double buffer structure is enabled or disabled. It is disabled if <TA0RDE> = 0 and enabled if <TA0RDE> = 1.

When the double buffer is enabled, data is transferred from the register buffer to the timer register when a 2^n overflow occurs in PWM mode, or at the start of the PPG cycle in PPG mode. Hence the double buffer cannot be used in timer mode.

A reset initializes <TA0RDE> to 0, disabling the double buffer. To use the double buffer, write data to the timer register, set <TA0RDE> to 1, and write the following data to the register buffer. Figure 3.7.3 show the configuration of TA0REG.

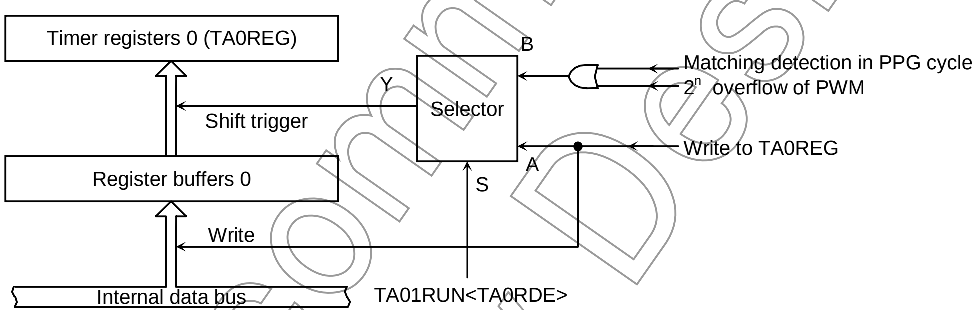


Figure 3.7.3 Configuration of TA0REG

Note: The same memory address is allocated to the timer register and the register buffer. When <TA0RDE> = 0, the same value is written to the register buffer and the timer register; when <TA0RDE> = 1, only the register buffer is written to.

The address of each timer register is as follows.

TA0REG: 000102H TA1REG: 000103H

TA2REG: 00010AH TA3REG: 00010BH

All these registers are write only and cannot be read.

Downloaded from Arrow.com.

(4) Comparator (CP0)

The comparator compares the value in an up counter with the value set in a timer register. If they match, the up counter is cleared to zero and an interrupt signal (INTTA0 or INTTA1) is generated. If timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

(5) Timer flip-flop (TA1FF)

The timer flip-flop (TA1FF) is a flip-flop inverted by the match detects signal (8-bit comparator output) of each interval timer.

Whether inversion is enabled or disabled is determined by the setting of the bit TA1FFCR<TA1FFIE> in the timer flip-flop control register.

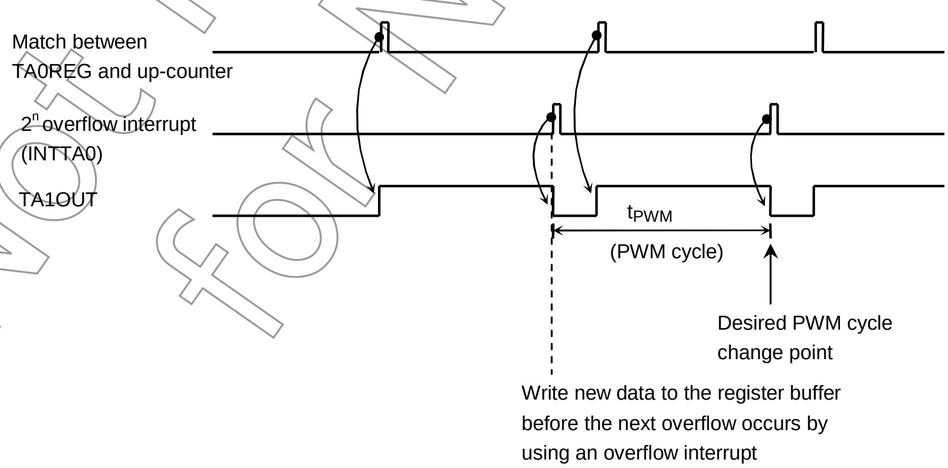
A reset clears the value of TA1FF1 to 0.

Writing 01 or 10 to TA1FFCR<TA1FFC1:0> sets TA1FF to 0 or 1. Writing 00 to these bits inverts the value of TA1FF. (This is known as software inversion.)

The TA1FF signal is output via the TA1OUT pin (Concurrent with PB1). When this pin is used as the timer output, the timer flip-flop should be set beforehand using the port B function register PBCR, PBFC.

Note: When the double buffer is enabled for an 8-bit timer in PWM or PPG mode, caution is required as explained below.
If new data is written to the register buffer immediately before an overflow occurs by a match between the timer register value and the up-counter value, the timer flip-flop may output an unexpected value.
For this reason, make sure that in PWM mode new data is written to the register buffer by six cycles ($f_{SYS} \times 6$) before the next overflow occurs by using an overflow interrupt.
In the case of using PPG mode, make sure that new data is written to the register buffer by six cycles before the next cycle compare match occurs by using a cycle compare match interrupt.

Example when using PWM mode



Downloaded from Arrow.com.
Not Recommended for New Design

3.7.3 SFRs

TMRA01 Run Register

TA01RUN (0100H)

	7	6	5	4	3	2	1	0
Bit symbol	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
Read/Write	R/W				R/W			
After reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

TA0REG double buffer control

0	Disable
1	Enable

Timer run/stop control

0	Stop and clear
1	Run (Count up)

I2TA01: Operation in IDLE2 mode
TA01PRUN: Run prescaler
TA1RUN: Run TMRA1
TA0RUN: Run TMRA0

Note: The values of bits 4, 5, 6 of TA01RUN are undefined when read.

TMRA23 Run Register

TA23RUN (0108H)

	7	6	5	4	3	2	1	0
Bit symbol	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
Read/Write	R/W				R/W			
After reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

TA2REG double buffer control

0	Disable
1	Enable

Timer run/stop control

0	Stop and clear
1	Run (Count up)

I2TA23: Operation in IDLE2 mode
TA23PRUN: Run prescaler
TA3RUN: Run TMRA3
TA2RUN: Run TMRA2

Note: The values of bits 4, 5, 6 of TA23RUN are undefined when read.

Figure 3.7.4 TMRA Registers

Not Recommended for New Design

Downloaded from Arrow.com.

TMRA01 Mode Register

		7	6	5	4	3	2	1	0
TA01MOD (0104H)	Bit symbol	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Operation mode 00: 8-bit timer mode 01: 16-bit timer mode 10: 8-bit PPG mode 11: 8-bit PWM mode		PWM cycle 00: Reserved 01: 2^6 10: 2^7 11: 2^8		Source clock for TMRA1 00: TA0TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock for TMRA0 00: TA0IN pin 01: ϕT1 10: ϕT4 11: ϕT16	

TMRA0 source clock selection

00	TA0IN (External input)
01	ϕT1 (Prescaler)
10	ϕT4 (Prescaler)
11	ϕT16 (Prescaler)

TMRA1 source clock selection

	TA01MOD <TA01M1:0> $\neq$ 01	TA01MOD <TA01M1:0> = 01
00	Comparator output from TMRA0	Overflow output from TMRA0
01	ϕT1	
10	ϕT16	
11	ϕT256	(16-bit timer mode)

PWM cycle selection

00	Reserved
01	$2^6 \times$ source clock
10	$2^7 \times$ source clock
11	$2^8 \times$ source clock

TMRA01 operation mode selection

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA0) + 8-bit timer (TMRA1)

Figure 3.7.5 TMRA Registers

Not Recommended for New Design

Downloaded from Arrow.com.

TMRA23 Mode Register

TA23MOD (010CH)

	7	6	5	4	3	2	1	0
Bit symbol	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-bit timer mode 01: 16-bit timer mode 10: 8-bit PPG mode 11: 8-bit PWM mode		PWM cycle 00: Reserved 01: 2^6 10: 2^7 11: 2^8		TMRA3 clock for TMRA3 00: TA2TRG 01: ϕT1 10: ϕT16 11: ϕT256		TMRA2 clock for TMRA2 00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	

→ TMRA2 source clock selection

00	Do not set
01	ϕT1 (Prescaler)
10	ϕT4 (Prescaler)
11	ϕT16 (Prescaler)

→ TMRA3 source clock selection

	TA23MOD <TA23M1:0> $\neq$ 01	TA23MOD <TA23M1:0> $=$ 01
00	Comparator output from TMRA2	Overflow output from TMRA2
01	ϕT1	
10	ϕT16	
11	ϕT256	(16-bit timer mode)

→ PWM cycle selection

00	Reserved
01	$2^6 \times$ source clock
10	$2^7 \times$ source clock
11	$2^8 \times$ source clock

→ TMRA23 operation mode selection

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA2) + 8-bit timer (TMRA3)

Figure 3.7.6 TMRA Registers

Not Recommended for New Design

Downloaded from Arrow.com.

TMRA1 Flip-Flop Control Register

TA1FFCR (0105H)

Read-modify-write instructions are prohibited.

	7	6	5	4	3	2	1	0
Bit symbol					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
Read/Write					R/W		R/W	
After reset					1	1	0	0
Function					00: Invert TA1FF 01: Set TA1FF 10: Clear TA1FF 11: Don't care		TA1FF control for inversion 0: Disable 1: Enable	TA1FF inversion select 0: TMRA0 1: TMRA1

Inverse signal for timer flip-flop 1 (TA1FF)
(Don't care except in 8-bit timer mode)

0	Inversion by TMRA0
1	Inversion by TMRA1

Inversion of TA1FF

0	Disabled
1	Enabled

Control of TA1FF

00	Inverts the value of TA1FF
01	Sets TA1FF to 1
10	Clears TA1FF to 0
11	Don't care

Figure 3.7.7 TMRA Registers

Not Recommended for New Design

Downloaded from Arrow.com.

TMRA3 Flip-Flop Control Register

TA3FFCR
(010DH)

Read-modify-write instructions are prohibited.

	7	6	5	4	3	2	1	0
Bit symbol					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
Read/Write					R/W		R/W	
After reset					1	1	0	0
Function					00: Invert TA3FF 01: Set TA3FF 10: Clear TA3FF 11: Don't care		TA3FF control for inversion 0: Disable 1: Enable	TA3FF inversion select 0: TMRA2 1: TMRA3

Inverse signal for timer flip-flop 3 (TA3FF)
(Don't care except in 8-bit timer mode)

0	Inversion by TMRA2
1	Inversion by TMRA3

Inversion of TA3FF

0	Disabled
1	Enabled

Control of TA3FF

00	Inverts the value of TA3FF
01	Sets TA3FF to 1
10	Clears TA3FF to 0
11	Don't care

Figure 3.7.8 TMRA Registers

Not Recommended for New Design

Downloaded from Arrow.com.

TMRA register

		7	6	5	4	3	2	1	0
TA0REG (0102H)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TA1REG (0103H)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TA2REG (010AH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TA3REG (010BH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							

Note: The above registers are prohibited read-modify-write instruction.

Figure 3.7.9 TMRA Registers

Not Recommended for New Design

Downloaded from Arrow.com.

3.7.4 Operation in Each Mode

(1) 8-bit timer mode

Both TMRA0 and TMRA1 can be used independently as 8-bit interval timers.

Setting its function or counter data for TMRA0 and TMRA1 after stop these registers.

a. Generating interrupts at a fixed interval (using TMRA1)

To generate interrupts at constant intervals using TMRA1 (INTTA1), first stop TMRA1 then set the operation mode, input clock and a cycle to TA01MOD and TA1REG register, respectively. Then, enable the interrupt INTTA1 and start TMRA1 counting.

Example: To generate an INTTA1 interrupt every 10 μseconds at fc = 33 MHz, set each register as follows:

* Clock state — System clock: High frequency (fc); Prescaler clock: f_{FPH}

		MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	←	–	X	X	X	–	–	0	–	Stop TMRA1 and clear it to 0.
TA01MOD	←	0	0	X	X	1	0	X	X	Select 8-bit timer mode and select φT1 ((2^3/fc)s at fc = 33 MHz) as the input clock.
TA1REG	←	0	0	1	0	1	0	0	0	Set TA1REG to 10 μs ÷ φT1 (2^3/fc)s ≈ 40 = 28H.
INTETA01	←	–	1	0	1	–	–	–	–	Enable INTTA1 and set it to level 5.
TA01RUN	←	–	X	X	X	–	1	1	–	Start TMRA1 counting.

X: Don't care, –: No change

Select the input clock using in Table 3.7.2.

Note: The input clocks for TMRA0 and TMRA1 are different from as follows.
TMRA0: TA0IN input, φT1, φT4 or φT16
TMRA1: Match output of TMRA0, φT1, φT16, φT256

Downloaded from Arrow.com.

b. Generating a 50% duty ratio square wave pulse

The state of the timer flip-flop (TA1FF) is inverted at constant intervals and its status output via the timer output pin (TA1OUT).

Example: To output a 1.5 μs square wave pulse from the TA1OUT pin at fc = 33 MHz, use the following procedure to make the appropriate register settings. This example uses TMRA1; however, either TMRA0 or TMRA1 may be used.

* Clock state

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

		7	6	5	4	3	2	1	0	
TA01RUN	←	–	X	X	X	–	–	0	–	Stop TMRA1 and clear it to 0.
TA01MOD	←	0	0	X	X	0	1	–	–	Select 8-bit timer mode and select φT1 ((2^3/fc)s at fc = 33 MHz) as the input clock.
TA1REG	←	0	0	0	0	0	0	1	1	Set the timer register to 1.5 μs ÷ φT1(2^3/fc)s ÷ 2 ≈ 3.
TA1FFCR	←	X	X	X	X	1	0	1	1	Clear TA1FF to 0 and set it to invert on the match detects signal from TMRA1.
PBCR	←	X	–	–	–	–	–	1	–	Set PB1 to function as the TA1OUT pin.
PBFC	←	X	–	–	–	–	–	1	X	
TA01RUN	←	–	X	X	X	–	1	1	–	Start TMRA1 counting.

X: Don't care, –: No change

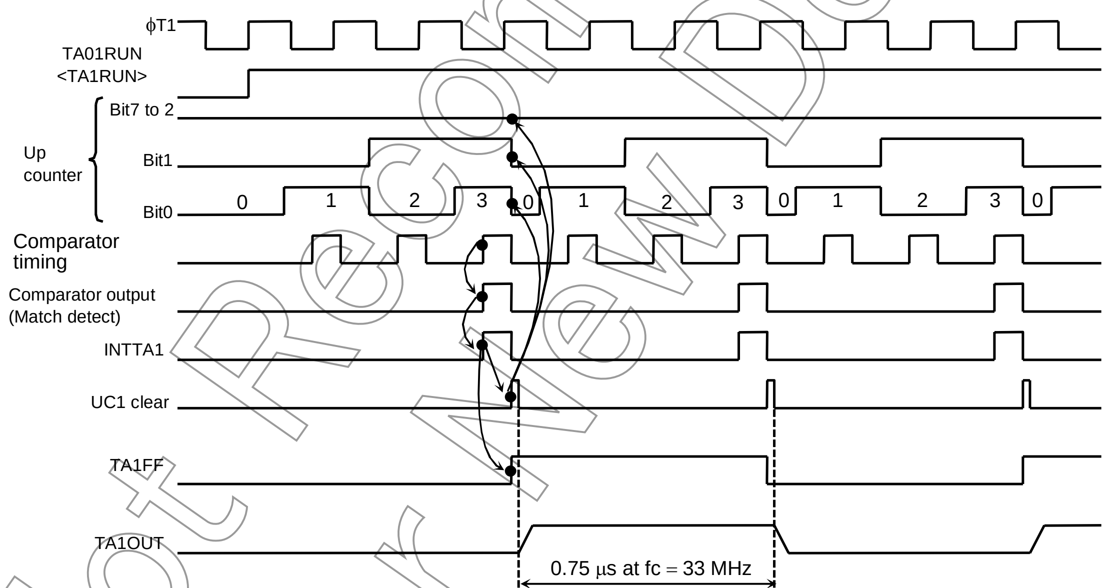


Figure 3.7.10 Square Wave Output Timing Chart (50% duty)

Not Recommended for New Design

Downloaded from Arrow.com.

c. Making TMRA1 count up on the match signal from the TMRA0 comparator

Select 8-bit timer mode and set the comparator output from TMRA0 to be the input clock to TMRA1.

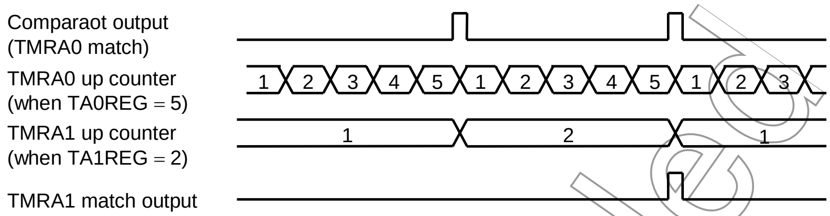


Figure 3.7.11 TMRA1 Count up on Signal from TMRA0

Not Recommended for New Design

Downloaded from Arrow.com.

(2) 16-bit timer mode

A 16-bit interval timer is configured by pairing the two 8-bit timers TMRA0 and TMRA1.

To make a 16-bit interval timer in which TMRA0 and TMRA1 are cascaded together, set TA01MOD<TA01M1:0> to 01.

In 16-bit timer mode, the overflow output from TMRA0 is used as the input clock for TMRA1, regardless of the value set in TA01MOD<TA01CLK1:0>. Table 3.7.2 shows the relationship between the timer (Interrupt) cycle and the input clock selection.

LSB 8 bits set to TA0REG and MSB 8 bits set to TA1REG. Please keep setting TA0REG first because setting data for TA0REG inhibit its compare function and setting data for TA1REG permit it.

Example: To generate an INTTA1 interrupt every 0.24 [s] at fc = 33 MHz, set the timer registers TA0REG and TA1REG as follows:

* Clock state
- System clock: High frequency (fc)
- Clock gear: 1 (fc)
- Prescaler clock: f_{FPH}

If ϕT16 ((2^7/fc)s at 33 MHz) is used as the input clock for counting, set the following value in the registers: 0.24 s ÷ (2^7/fc)s ≈ 62500 = F424H

(e.g., set TA1REG to F4H and TA0REG to 24H).

As a result, INTTA1 interrupt can be generated every 0.24 [s].

The comparator match signal is output from TMRA0 each time the up counter UC0 matches TA0REG, though the up counter UC0 is not be cleared and also INTTA0 is not generated.

In the case of the TMRA1 comparator, the match detect signal is output on each comparator pulse on which the values in the up counter UC1 and TA1REG match. When the match detect signal is output simultaneously from both the comparators TMRA0 and TMRA1, the up counters UC0 and UC1 are cleared to 0 and the interrupt INTTA1 is generated. Also, if inversion is enabled, the value of the timer flip-flop TA1FF is inverted.

Example: When TA1REG = 04H and TA0REG = 80H

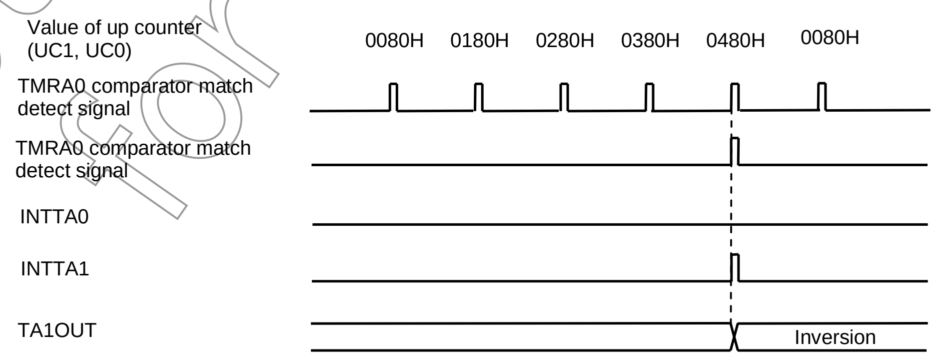


Figure 3.7.12 Timer Output by 16-Bit Timer Mode

Not Recommended for New Design

Downloaded from Arrow.com.

(3) 8-bit PPG (Programmable pulse generation) output mode

Square wave pulses can be generated at any frequency and duty ratio by TMRA0. The output pulses may be active-Low or active-High. In this mode TMRA1 cannot be used.

TMRA0 outputs pulses on the TA1OUT pin.

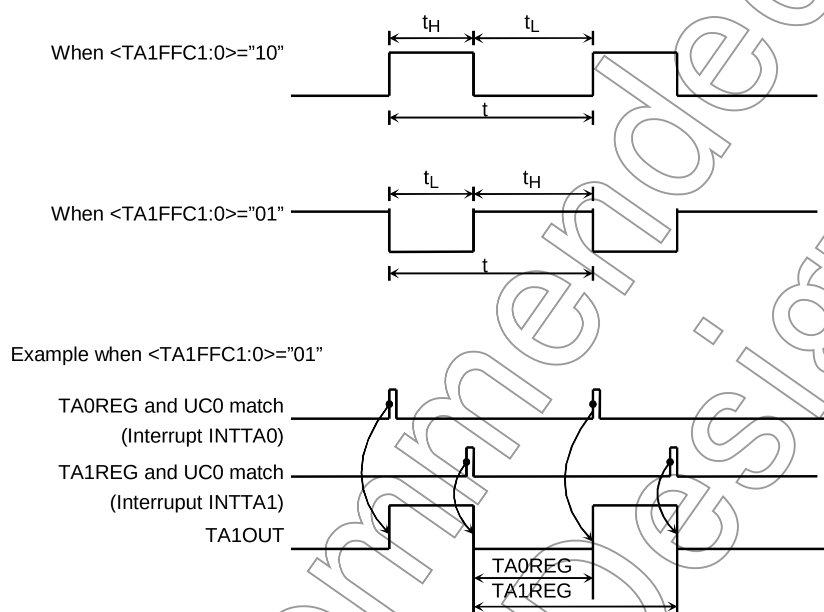


Figure 3.7.13 8-Bit PPG Output Waveforms

Not Recommended for New Design

Downloaded from Arrow.com.

In this mode, a programmable square wave is generated by inverting the timer output each time the 8-bit up counter (UC0) matches the value in one of the timer registers TA0REG or TA1REG.

The value set in TA0REG must be smaller than the value set in TA1REG.

Although the up counter for TMRA1 (UC1) is not used in this mode, TA01RUN<TA1RUN> should be set to 1, so that UC1 is set for counting.

Figure 3.7.14 shows a block diagram representing this mode.

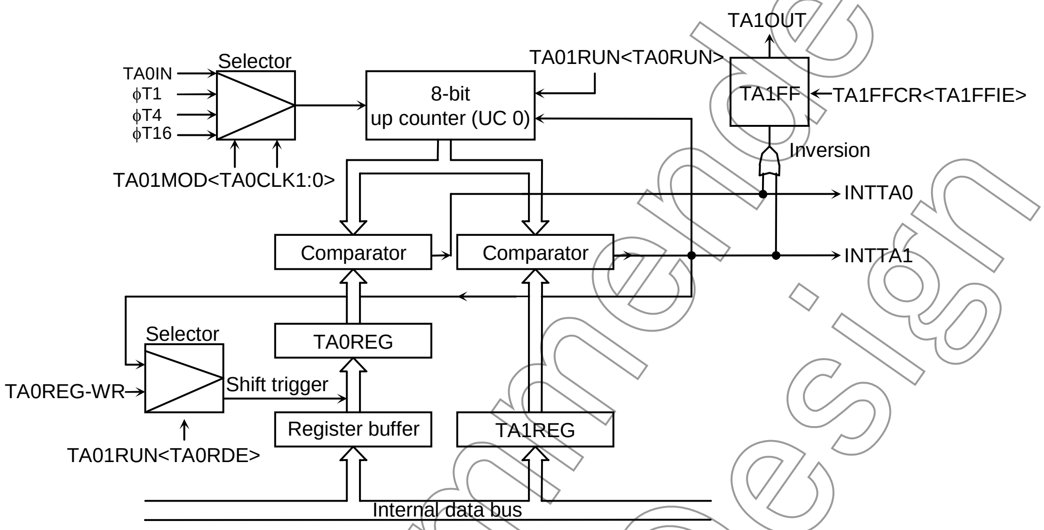


Figure 3.7.14 Block Diagram of 8-Bit PPG Output Mode

If the TA0REG double buffer is enabled in this mode, the value of the register buffer will be shifted into TA0REG each time TA1REG matches UC0.

Use of the double buffer facilitates the handling of low-duty waves (when duty is varied).

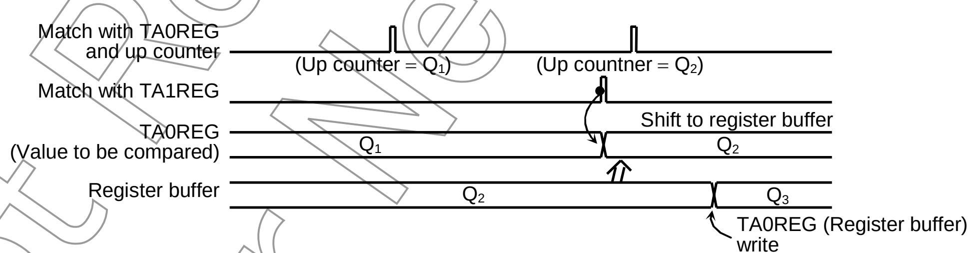
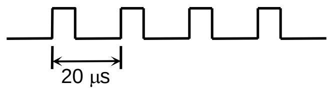


Figure 3.7.15 Operation of Register Buffer

Not Recommended for New Design

Downloaded from Arrow.com.

Example: To generate 1/4-duty 50-kHz pulses (at fc = 33 MHz)



* Clock state

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

Calculate the value which should be set in the timer register.
To obtain a frequency of 50 kHz, the pulse cycle t should be: t = 1/50 kHz = 20 µs
ϕT1 = (2^3/fc)s (at 33 MHz);

20 µs ÷ (2^3/fc)s ≈ 83

Therefore set TA1REG = 83 = 53H
The duty is to be set to 1/4: t × 1/4 = 20 µs × 1/4 = 5 µs

5 µs ÷ (2^3/fc)s ≈ 10

Therefore, set TA0REG = 21 =15H.

		7	6	5	4	3	2	1	0	
TA01RUN	←	–	X	X	X	–	0	0	0	Stop TMRA0 and TMRA01 and clear it to 0.
TA01MOD	←	1	0	X	X	X	X	0	1	Set the 8-bit PPG mode, and select ϕT1 as input clock.
TA0REG	←	0	0	0	1	0	1	0	1	Write 15H
TA1REG	←	0	1	0	1	0	0	1	1	Write 53H
TA1FFCR	←	X	X	X	X	0	1	1	X	Set TA1FF, enabling both inversion and the double buffer. Writing 10 provides negative logic pulse.
PBCR	←	X	–	–	–	–	–	1	–	Set PB1 as the TA1OUT pin.
PBFC	←	X	–	–	–	–	–	1	X	
TA01RUN	←	1	X	X	X	–	1	1	1	Start TMRA0 and TMRA01 counting.

X: Don't care, –: No change

Not Recommended for New Design

Downloaded from Arrow.com.

(4) 8-bit PWM (Pulse width modulation) output mode

This mode is only valid for TMRA0. In this mode, a PWM pulse with the maximum resolution of 8 bits can be output.

When TMRA0 is used the PWM pulse is output on the TA1OUT pin. TMRA1 can also be used as an 8-bit timer.

The timer output is inverted when the up counter (UC0) matches the value set in the timer register TA0REG or when 2^n counter overflow occurs (n = 6, 7 or 8 as specified by TA01MOD<PWM01:00>). The up counter UC0 is cleared when 2^n counter overflow occurs.

The following conditions must be satisfied before this PWM mode can be used.

Value set in TA0REG < Value set for 2^n counter overflow
Value set in TA0REG ≠ 0

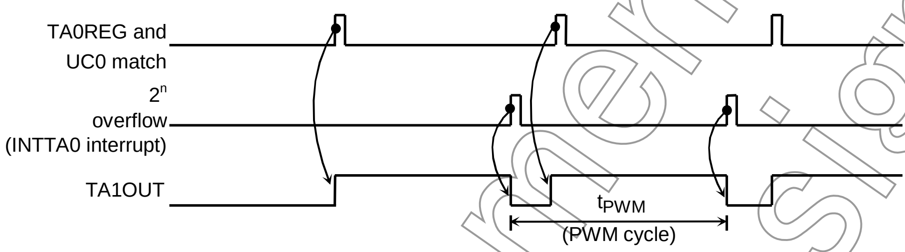


Figure 3.7.16 8-Bit PWM Waveforms

Figure 3.7.17 shows a block diagram representing this mode.

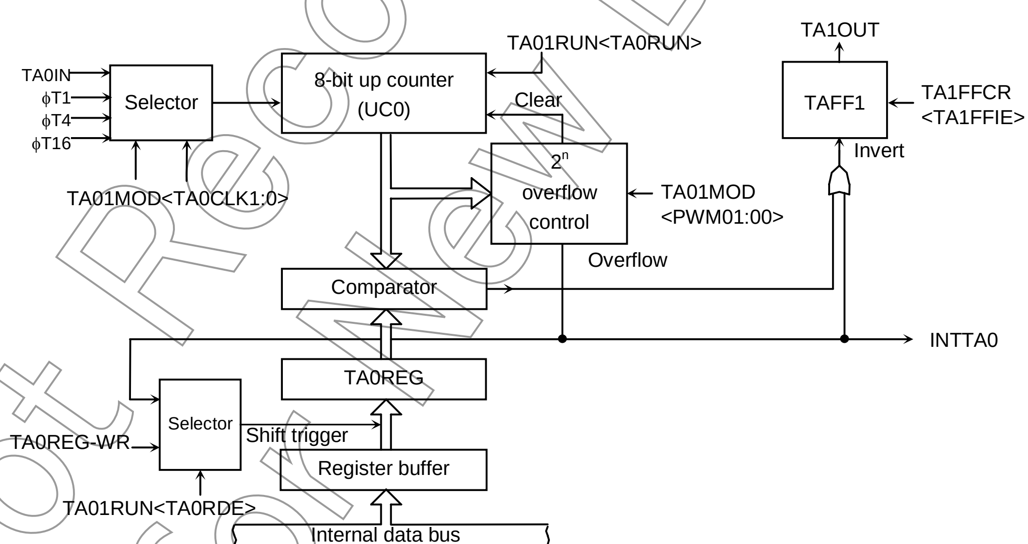


Figure 3.7.17 Block Diagram of 8-Bit PWM Mode

Not Recommended for New Design

Downloaded from Arrow.com.

In this mode, the value of the register buffer will be shifted into TA0REG if 2^n overflow is detected when the TA0REG double buffer is enabled.

Use of the double buffer facilitates the handling of low duty ratio waves.

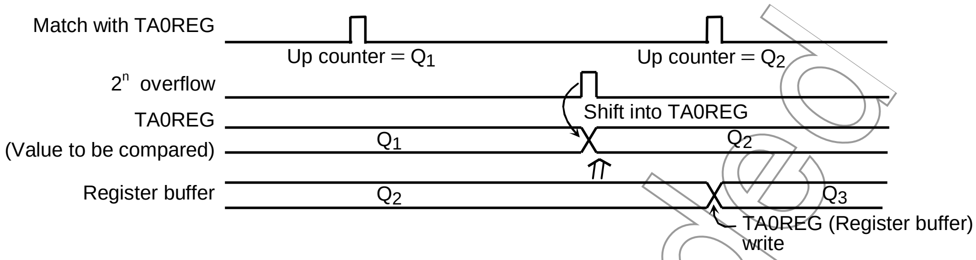
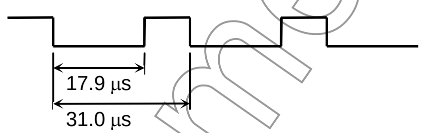


Figure 3.7.18 Register Buffer Operation

Example: To output the following PWM waves on the TA1OUT pin at fc = 33MHz:



* Clock state

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

To achieve a 31.0 μs PWM cycle by setting $\phi T1 = (2^3/fc)s$ (at fc = 33 MHz):

$31.0\ \mu s \div (2^3/fc)s \approx 128 = 2^n$

Therefore n should be set to 7.

Since the low-level period is 37.0 μs when $\phi T1 = (2^3/fc)s$,

set the following value for TA0REG:

$17.9\ \mu s \div (2^3/fc)s \approx 74 = 4AH$

		MSB 7	6	5	4	3	2	1	LSB 0	
TA01RUN	←	–	X	X	X	–	–	–	0	Stop TMRA0 and clear it to 0.
TA01MOD	←	1	1	1	0	–	–	0	1	Select 8-bit PWM mode (Cycle: 2^7) and select ϕT1 as the input clock.
TA0REG	←	0	1	0	0	1	0	1	0	Write 4AH.
TA1FFCR	←	X	X	X	X	1	0	1	X	Clear TA1FF to 0, enable the inversion and double buffer.
PBCR	←	X	–	–	–	–	–	1	–	Set PB1 and the TA1OUT pin.
PBFC	←	X	–	–	–	–	–	1	X	
TA01RUN	←	1	X	X	X	–	1	–	1	Start TMRA0 counting.

X: Don't care, –: No change

Not Recommended for New Design

Downloaded from Arrow.com.

Table 3.7.3 PWM Cycle

at fc = 33 MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	PWM Cycle								
			2^6			2^7			2^8		
			ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16
1 (fs)	00 (f_{FPH})	XXX	15.6 ms	62.5 ms	250 ms	31.3 ms	125 ms	500 ms	62.5 ms	250 ms	1000 ms
0 (fc)		000 (fc)	15.5 μs	62.1 μs	248.2 μs	31.0 μs	124.1 μs	496.5 μs	62.1 μs	248.2 μs	993.0 μs
		001 (fc/2)	31.0 μs	124.1 μs	496.5 μs	62.1 μs	248.2 μs	993.0 μs	124.1 μs	496.5 μs	1986 μs
		010 (fc/4)	32.1 μs	248.2 μs	993.0 μs	124.1 μs	496.5 μs	1986 μs	248.2 μs	993.0 μs	3972 μs
		011 (fc/8)	124.1 μs	496.5 μs	1986 μs	248.2 μs	993.0 μs	3972 μs	496.5 μs	1986 μs	7944 μs
		100 (fc/16)	248.2 μs	993.0 μs	3972 μs	496.5 μs	1986 μs	7944 μs	993 μs	3972 μs	15888 μs
	10 (fc/16 clock)	XXX	248.2 μs	993.0 μs	3972 μs	496.5 μs	1986 μs	7944 μs	993 μs	3972 μs	15888 μs

XXX: Don't care

(5) Settings for each mode

Table 3.7.4 shows the SFR settings for each mode.

Table 3.7.4 Timer Mode Setting Registers

Register Name	TA01MOD				TA1FFCR
<Bit Symbol>	<TA01M1:0>	<PWM01:00>	<TA1CLK1:0>	<TA0CLK1:0>	TA1FFIS
Function	Timer Mode	PWM Cycle	Upper Timer Input Clock	Lower Timer Input Clock	Timer F/F Invert Signal Select
8-bit timer × 2 channels	00	–	Lower timer match ϕT1, ϕT16, ϕT256 (00, 01, 10, 11)	External clock ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	0: Lower timer output 1: Upper timer output
16-bit timer mode	01	–	–	External clock ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit PPG × 1 channel	10	–	–	External clock ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit PWM × 1 channel	11	2^6, 2^7, 2^8 (01, 10, 11)	–	External clock ϕT1, ϕT4, ϕT16 (00, 01, 10, 11)	–
8-bit Timer × 1 channel	11	–	ϕT1, ϕT16, ϕT256 (01, 10, 11)	–	Output disabled

–: Don't care

Not Recommended for New Design

Downloaded from Arrow.com.

3.8 External Memory Extension Function (MMU)

This is MMU function which can expand program/data area to 106 Mbytes by having 4 local areas.

Address pins to external memory are 2 extended address bus pins (EA24, EA25) and 8 extended chip select pins ($\overline{CS2A}$ to $\overline{CS2E}$) in addition to 24 address bus pins (A0 to A23) which are common specification of TLCS-900 family and 4 chip select pins ($\overline{CS0}$ to $\overline{CS3}$) output from CS/WAIT controller.

The feature and the recommendation setting method of two types are shown below.

In addition, AH in the table is the value which number address 23 to 16 displayed as hex.

Purpose	Item	(A): For Standard Extended Memory	(B): For Many Pieces Extended Memory
Program ROM	Maximum memory size	2 Mbytes: COMMON2 + 14 Mbytes: bank (16 Mbytes × 1 pcs)	
	Used local area, BANK number	LOCAL2 (AH = C0 – DF: 2 Mbytes × 7 BANK)	
	Setting CS/WAIT	Setup AH = C0 – FF to CS2	Setup AH = 80 – FF to CS2
	Used $\overline{CS}$ pin	$\overline{CS2}$	$\overline{CS2A}$
Data ROM	Maximum memory size	64 Mbytes (64 Mbytes × 1 pcs)	64 Mbytes (16 Mbytes × 4 pcs)
	Used local area, BANK number	LOCAL3 (AH = 80 – BF: 4 Mbytes × 16 BANK)	LOCAL3 (AH = 80 – BF: 4 Mbytes × 16 BANK)
	Setting CS/WAIT	Setup AH = 80 – BF to CS3	Setup AH = 80 – FF to CS2
	Used $\overline{CS}$ pins	$\overline{CS3}$, EA24, EA25	$\overline{CS2B}$, $\overline{CS2C}$, $\overline{CS2D}$, $\overline{CS2E}$
Option program ROM	Maximum memory size	2 Mbytes: COMMON1 + 14 Mbytes : bank (16 Mbytes × 1 pcs)	
	Used local area, BANK number	LOCAL1 (AH = 40 – 5F: 2 Mbytes × 7 BANK)	
	Setting CS/WAIT	Setup AH = 40 – 7F to CS1	
	Used $\overline{CS}$ pin	$\overline{CS1}$	
Data RAM	Maximum memory size	1 Mbyte : COMMON0 + 7 Mbytes: bank (8 Mbytes × 1 pcs)	
	Used local area, BANK number	LOCAL0 (AH = 10 – 1F: 1 Mbyte × 7 BANK)	
	Setting CS/WAIT	Setup AH = 00 – 3F to CS0	Setup AH = 00 – 1F to CS3
	Used $\overline{CS}$ pin	$\overline{CS0}$	$\overline{CS3}$
Extended memory 1	Maximum memory size	COMMON0α	2 Mbytes (2 Mbytes × 1 pcs)
	Used local area, BANK number	Overlapped data RAM	None
	Setting CS/WAIT	Setup AH = 00 – 3F to CS0	Setup AH = 20 – 3F to CS0
	Used $\overline{CS}$ pin	$\overline{CS0}$	$\overline{CS0}$
Total memory size		16M + 64M + 16M + 8M = 104 Mbytes	16M + (16M + 16M + 16M + 16M) + 16M + 8M + 2M = 106 Mbytes

Not Recommended for New Design

Downloaded from Arrow.com.

3.8.1 Recommendable Memory Map

The recommendation logic address memory map at the time of varieties extension memory correspondence is shown in Figure 3.8.1. And, a physical-address map is shown in Figure 3.8.2.

However, when memory area is less than 16 Mbytes and is not expanded, please refer to section of CS/WAIT controller. Setting of register in MMU is not necessary.

Since it is being fixed, the address of a local area cannot be changed.

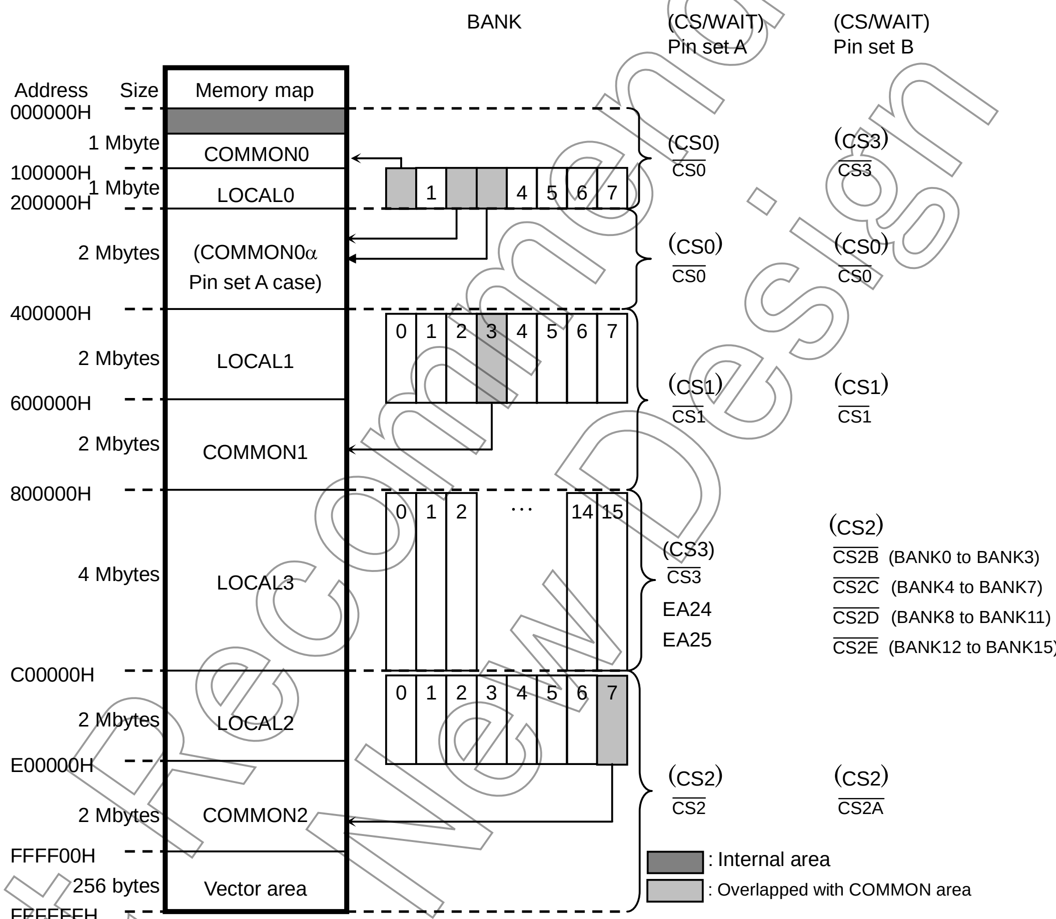


Figure 3.8.1 Logical Address Map

Downloaded from Arrow.com.

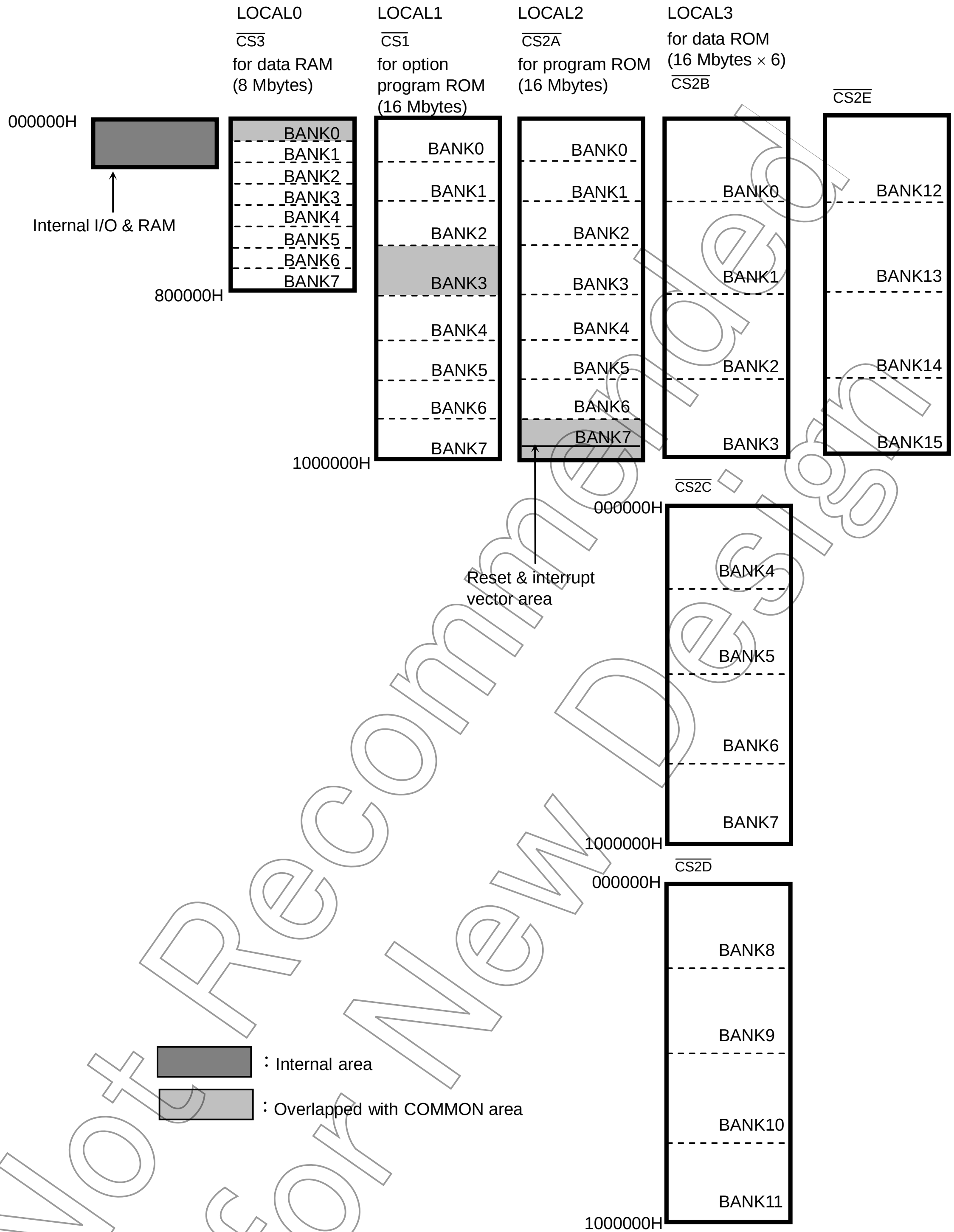


Figure 3.8.2 Physical Address Map

Not Recommended for New Design

Downloaded from Arrow.com.

3.8.2 Control Registers

Setup bank value and bank use in bank setting register of each local area of LOCAL register in common area. Moreover, in that case, a combination pin is set up and mapping is simultaneously setup by the CS/WAIT controller. When CPU outputs logical address of the local area, MMU outputs physical address to the outside address bus pin according to value of bank setting register. Access of external memory becomes possible therefore.

LOCAL0 Register

LOCAL0 (0350H)		7	6	5	4	3	2	1	0
	Bit symbol	L0E					L0EA22	L0EA21	L0EA20
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	BANK for LOCAL0 0: Disable 1: Enable					Setting BANK number for LOCAL0 “000” setting is prohibited because it pretend COMMON 0 area		

LOCAL1 Register

LOCAL1 (0351H)		7	6	5	4	3	2	1	0
	Bit symbol	L1E					L1EA23	L1EA22	L1EA21
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	BANK for LOCAL1 0: Disable 1: Enable					Setting BANK number for LOCAL1 “001” setting is prohibited because it pretend COMMON 0 area		

LOCAL2 Register

LOCAL2 (0352H)		7	6	5	4	3	2	1	0
	Bit symbol	L2E					L2EA23	L2EA22	L2EA21
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	BANK for LOCAL2 0: Disable 1: Enable					Setting BANK number for LOCAL2 “111” setting is prohibited because it pretend COMMON 0 area		

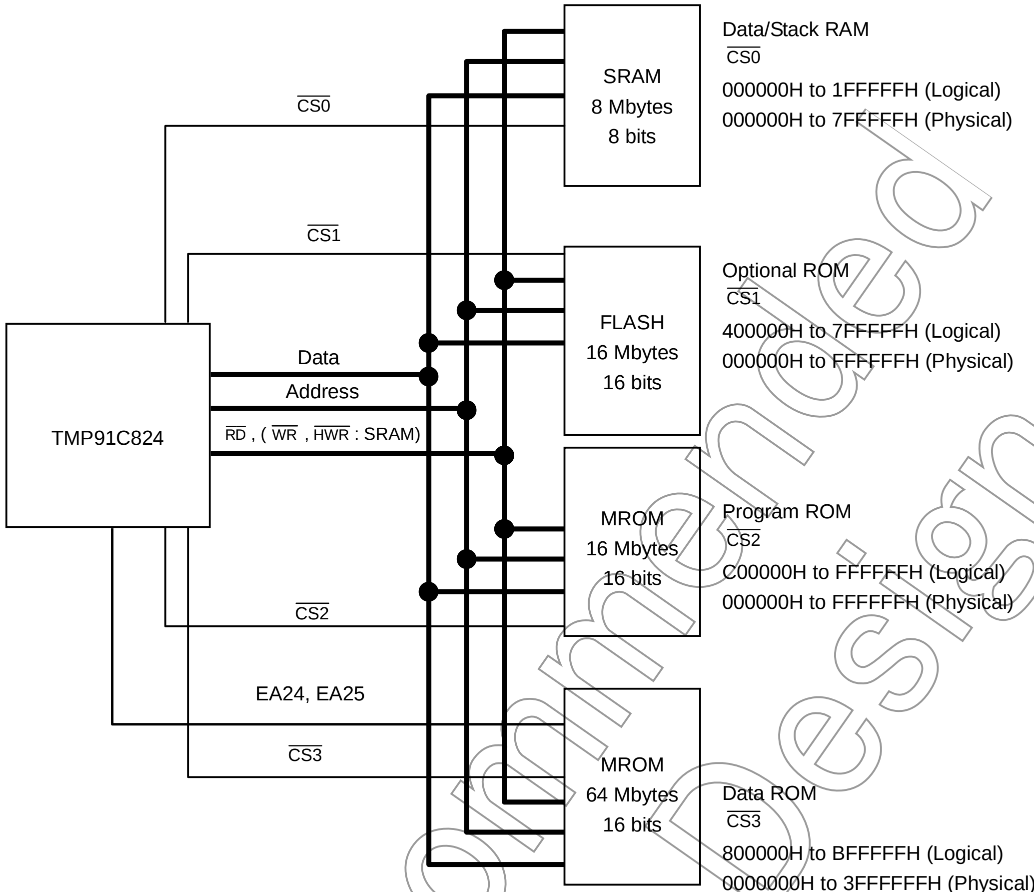
LOCAL3 Register

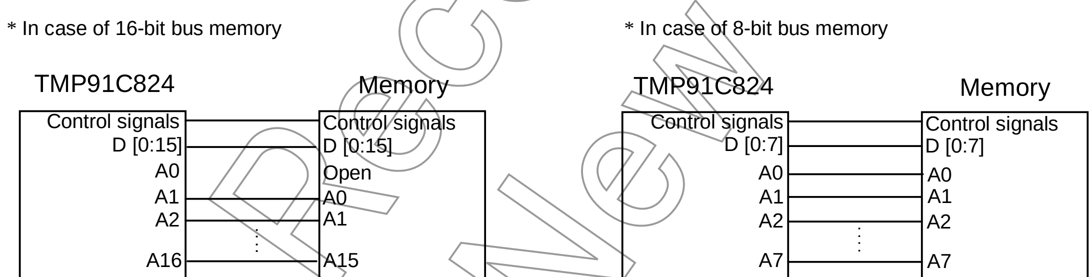
LOCAL3 (0353H)		7	6	5	4	3	2	1	0
	Bit symbol	L3E			L3EA26	L3EA25	L3EA24	L3EA23	L3EA22
	Read/Write	R/W			R/W	R/W	R/W	R/W	R/W
	After reset	0			0	0	0	0	0
	Function	BANK for LOCAL3 0: Disable 1: Enable			01000 to 01011: $\overline{CS2D}$ 01100 to 01111: $\overline{CS2E}$ 00000 to 00011: $\overline{CS2B}$ 00100 to 00111: $\overline{CS2C}$ 10000 to 11111: Set prohibition				

Note: In case of this TMP91C824, because most upper address bit of physical address is EA25, most upper address bit of BANK register is meaningless. 4 bits of upper 5-bit address means 16 BANKs.

Not Recommended for New Design

Downloaded from Arrow.com.





* In case of 16-bit bus memory, address connection is ... : CPU A1 = Memory A0, CPU A2 = Memory A1...

* In case of 8-bit bus memory, address connection is ... : CPU A0 = Memory A0, CPU A1 = Memory A1...

Figure 3.8.3 H/W Setting Example

Not Recommended for New Design

Downloaded from Arrow.com.

At Figure 3.8.3, it shows example of connection TMP91C824 and some memories: Program ROM: MROM, 16 Mbytes, data ROM: MROM, 64 Mbytes, data RAM: SRAM, 8 Mbytes, 8-bit bus, option ROM: Flash, 16 Mbytes.

In case of 16-bit bus memory connection, it need to shift 1-bit address bus from TMP91C824 and 8-bit bus case, direct connection address bus from TMP91C824.

In that figure, logical address and physical address are shown. And each memory allot each chip select signal, RAM: $\overline{CS0}$, $\overline{FLASH}$ ROM: $\overline{CS1}$, program MROM: $\overline{CS2}$, data MROM: $\overline{CS3}$. In case of this example, as data MROM is 64 Mbytes, this MROM connect to EA24 and EA25.

Initial condition after reset, because TMP91C824 access from $\overline{CS2}$ area, $\overline{CS2}$ area allot to program ROM. It can set free setting except program ROM.

```
;Initial Setting
;CS0
      LD      (MSAR0),00H     ; Logical address area: 000000H to 1FFFFFH
      LD      (MAMR0),FFH     ; Logical address size: 2 Mbytes
      LD      (B0CS),89H      ; Condition: 8 bits, 1 wait (8 Mbytes, SRAM)
;CS1
      LD      (MSAR1),40H     ; Logical address area: 400000H to 7FFFFFH
      LD      (MAMR1),FFH     ; Logical address size: 4 Mbytes
      LD      (B1CS),80H      ; Condition: 16 bits, 2 waits (16 Mbytes, Flash ROM)
;CS2
      LD      (MSAR2),C0H     ; Logical address area: C00000H to FFFFFFH
      LD      (MAMR2),7FH     ; Logical address size: 4 Mbytes
      LD      (B2CS),C3H      ; Condition: 16 bits, 0 waits (16 Mbytes, MROM)
;CS3
      LD      (MSAR3),80H     ; Logical address area: 800000H to BFFFFFH
      LD      (MAMR3),7FH     ; Logical address size: 4 Mbytes
      LD      (B3CS),85H      ; Condition: 16 bits, 3 waits (64 Mbytes, MROM)
;CSX
      LD      (BEXCS),00H     ; Other : 16 bits, 2 waits (Don't care)
;Port
      LD      (P6FC), 3FH     ; CS0 to CS3, EA24, EA25: Port 6 setting
```

Figure 3.8.4 BANK Operation S/W Example 1

Secondly, it shows example of initial setting at Figure 3.8.4.

Because $\overline{CS0}$ connect to RAM: 8-bit bus, 8 Mbytes, it need to set 8-bit bus. At this example, it set 1-wait setting. In the same way $\overline{CS1}$ set to 16-bit bus and 2 waits, $\overline{CS2}$ set 16-bit bus and 0 waits, $\overline{CS3}$ set 16-bit bus and 3 waits.

By CS/WAIT controller, each chip selection signal's memory size, don't set actual connect memory size, need to set that logical address size: fitting to each local area. Actual physical address is set by each area's BANK register setting.

CSX setting of CS/WAIT controller is except above CS0 to CS3's setting. This program example isn't used CSX setting.

Finally pin condition is set. Port 60 to 65 set to $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, EA24, EA25.

Not Recommended for New Design

Downloaded from Arrow.com.

```
;BANK Operation
;***** CS2 *****
ORG   000000H                 ; Program ROM: Start address at BANK0 of LOCAL2
ORG   200000H                 ; Program ROM: Start address at BANK1 of LOCAL2
ORG   400000H                 ; Program ROM: Start address at BANK2 of LOCAL2
ORG   600000H                 ; Program ROM: Start address at BANK3 of LOCAL2
ORG   800000H                 ; Program ROM: Start address at BANK4 of LOCAL2
ORG   a00000H                 ; Program ROM: Start address at BANK5 of LOCAL2
ORG   c00000H                 ; Program ROM: Start address at BANK6 of LOCAL2

ORG   E00000H                 ; Program ROM: Start address at BANK7 (= COMMON2) of LOCAL2
                              ; Logical address E00000H to FFFFFFH
                              ; Physical address 0E00000H to 0FFFFFFH
      LD    (LOCAL3),85H      ; LOCAL3 BANK5 set 14xxxxH
      LDW   HL,(800000H)      ; Load data (5555H) form BANK5
                                          (140000H: Physical address) of LOCAL3 (CS3)
      LD    (LOCAL3),88H      ; LOCAL3 BANK8 set 20xxxxH
      LDW   BC,(800000H)      ; Load data (AAAAH) form BANK8
                                          (200000H: Physical address) of LOCAL3 (CS3)
~
ORG   FFFFFFH                 ; Program ROM: End address at BANK7 (= COMMON2) of LOCAL2
```

```
;***** CS3 *****
ORG   0000000H                ; Data ROM: Start address at BANK0 of LOCAL3
ORG   0400000H                ; Data ROM: Start address at BANK1 of LOCAL3
ORG   0800000H                ; Data ROM: Start address at BANK2 of LOCAL3
ORG   0C00000H                ; Data ROM: Start address at BANK3 of LOCAL3
ORG   1000000H                ; Data ROM: Start address at BANK4 of LOCAL3
ORG   1400000H                ; Data ROM: Start address at BANK5 of LOCAL3
      dw    5555H
~
ORG   1800000H                ; Data ROM: Start address at BANK6 of LOCAL3
ORG   1C00000H                ; Data ROM: Start address at BANK7 of LOCAL3
ORG   2000000H                ; Data ROM: Start address at BANK8 of LOCAL3
      dw    AAAAH
~
ORG   2400000H                ; Data ROM: Start address at BANK9 of LOCAL3
ORG   2800000H                ; Data ROM: Start address at BANK10 of LOCAL3
ORG   2C00000H                ; Data ROM: Start address at BANK11 of LOCAL3
ORG   3000000H                ; Data ROM: Start address at BANK12 of LOCAL3
ORG   3400000H                ; Data ROM: Start address at BANK13 of LOCAL3
ORG   3800000H                ; Data ROM: Start address at BANK14 of LOCAL3
ORG   3C00000H                ; Data ROM: Start address at BANK15 of LOCAL3
ORG   3FFFFFFH                ; Data ROM: End address at BANK15 of LOCAL3
```

Figure 3.8.5 BANK Operation S/W Example 2

Here shows example of data access between one BANK and other BANK. Figure 3.8.5 is one software example. A dot line square area shows one memory and each dot line square shows $\overline{\text{CS2}}$'s program ROM and $\overline{\text{CS3}}$'s data ROM. Program start from E00000H address, firstly, write to BANK register of LOCAL3 area upper 5-bit address of access point.

In case of this TMP91C824, because most upper address bit of physical address is EA25, most upper address bit of BANK register is meaningless. 4 bits of upper 5-bit address means 16 BANKs. After setting BANK5, accessing 800000H to BFFFFFH address: Logical LOCAL3 address, actually access to physical 1400000H to 1700000H address.

Downloaded from Arrow.com.

Not Recommended for New Design

```
;BANK Operation
;***** CS2 *****
ORG   000000H                 ; Program ROM: Start address at BANK0 of LOCAL2
ORG   200000H                 ; Program ROM: Start address at BANK1 of LOCAL2
      NOP                     ; Operation at BANK1of LOCAL2
~
      JP  E00100H             ; Jump to BANK7 (= COMMON2) of LOCAL2
ORG   400000H                 ; Program ROM: Start address at BANK2 of LOCAL2
ORG   600000H                 ; Program ROM: Start address at BANK3 of LOCAL2
      NOP                     ; Operation at BANK3 of  LOCAL2
~
      JP  E00200H             ; Jump to BANK7 (= COMMON2) of LOCAL2
ORG   800000H                 ; Program ROM: Start address at BANK4 of LOCAL2
ORG   a00000H                 ; Program ROM: Start address at BANK5 of LOCAL2
ORG   c00000H                 ; Program ROM: Start address at BANK6 of LOCAL2

!!!! Program Start !!!!
ORG   E00000H                 ; Program ROM: Start address at BANK7 (= COMMON2) of LOCAL2
                              ; Logical address E00000H to FFFFFFH
                              ; Physical address 0E00000H to 0FFFFFFH
      LD    (LOCAL2),81H      ; LOCAL2 BANK1 set 20xxxxH
      JP    C00000H           ; Jump to BANK1 (200000H: Physical address) of LOCAL2
~
ORG   E00100H
      LD    (LOCAL2),83H      ; LOCAL2 BANK3 set 60xxxxH
      JP    C00000H           ; Jump to BANK3 (600000H: Physical address) of LOCAL2
~
ORG   E00200H
      LD    (LOCAL1),84H      ; LOCAL1 BANK4 set 80xxxxH
      JP    400000H           ; Jump to BANK4 (800000H: Physical address) of LOCAL1
ORG   FFFFFFH                 ; Program ROM: End address at BANK7 (= COMMON2) of LOCAL2
```

```
;***** CS1 *****
ORG   000000H                 ; Program ROM: Start address at BANK0 of LOCAL1
ORG   200000H                 ; Program ROM: Start address at BANK1 of LOCAL1
ORG   400000H                 ; Program ROM: Start address at BANK2 of LOCAL1
ORG   600000H                 ; Program ROM: Start address at BANK3 (= COMMON1) of LOCAL1
      LD    (LOCAL1),87H      ; LOCAL1 BANK7 set E0xxxxH
      JP    400000H           ; Jump to BANK7 (E00000H: Physical address) of LOCAL1
ORG   800000H                 ; Program ROM: Start address at BANK4 of LOCAL1
      NOP                     ; Operation at BANK4 of LOCAL1
~
      JP  600000H             ; Jump to BANK3 (= COMMON1) of LOCAL1
ORG   a00000H                 ; Program ROM: Start address at BANK5 of LOCAL1
ORG   c00000H                 ; Program ROM: Start address at BANK6 of LOCAL1
ORG   E00000H                 ; Program ROM: Start address at BANK7 of LOCAL1
      LD    (LOCAL1),80H      ; LOCAL1 BANK0 set 00xxxxH
      JP    400000H           ; Jump to BANK0 (000000H: Physical address) of LOCAL1

      It's prohibit to set other BANK setting in except common area
      Program run away

ORG   FFFFFFH                 ; Program ROM: End address at BANK7 of LOCAL1
```

Figure 3.8.6 BANK Operation S/W Example 3

Not Recommended for New Design

Downloaded from Arrow.com.

At Figure 3.8.6, it shows example of program jump.

In the same way with before example, two dot line squares show each $\overline{CS2}$'s program ROM and $\overline{CS1}$'s option ROM. Program start from E00000H common address, firstly, write to BANK register of LOCAL2 area upper 3-bit address of jumping point.

After setting BANK1, jumping C00000H to DFFFFFH address: Logical LOCAL2 address, actually jump to physical 2000000H to 3FFFFFH address. When return to common area, it can only jump to E00000H to FFFFFFH without writing to BANK register of LOCAL2 area.

By a way of setting of BANK register, the setting that BANK address and common address conflict with is possible. When two kinds or more logical addresses to show common area exist, management of BANK is confused. We recommend not using The BANK setting, BANK address and common address conflict with.

When it jumps to one memory from other different memory, it can set same as the last time setting. It needs to write to BANK register of local1 area upper 3-bit address of jumping point. After setting BANK4, jumping 400000H to 5FFFFFH address: Logical LOCAL1 address, actually jump to physical 8000000H to 9FFFFFH address.

It is a mark paid attention to here, it needs to go by way of common area by all means when moves from a bank to a bank. In other words, it must write to BANK register only in common area and It is prohibit writing the BANK register in BANK area. If it modify the BANK register's data in BANK area, program run-away.

Not Recommended for New Design

Downloaded from Arrow.com.

3.9 Serial Channels

TMP91C824 includes 2 serial I/O channels. For both channels either UART mode (Asynchronous transmission) or I/O interface mode (Synchronous transmission) can be selected.

- I/O interface mode —— Mode 0: For transmitting and receiving I/O data using the synchronizing signal SCLK for extending I/O.
- UART mode
  - Mode 1: 7-bit data
  - Mode 2: 8-bit data
  - Mode 3: 9-bit data

In mode 1 and mode 2, a parity bit can be added. Mode 3 has a wakeup function for making the master controller start slave controllers via a serial link (A multi-controller system).

Figure 3.9.2, Figure 3.9.3 are block diagrams for each channel.

Serial channels 0 and 1 can be used independently.

Both channels operate in the same fashion except for the following points; hence only the operation of channel 0 is explained below.

Table 3.9.1 Differences between Channels 0 to 1

	Channel 0	Channel 1
Pin Name	TXD0 (PC0) RXD0 (PC1) $\overline{\text{CTS0}}$ /SCLK0 (PC2)	TXD1 (PC3) RXD1 (PC4) $\overline{\text{CTS1}}$ /SCLK1 (PC5)
IrDA Mode	Yes	No

This chapter contains the following sections:

3.9.1 Block Diagrams
3.9.2 Operation of Each Circuit
3.9.3 SFRs
3.9.4 Operation in Each Mode
3.9.5 Support for IrDA

Not Recommended for New Design

Downloaded from Arrow.com.

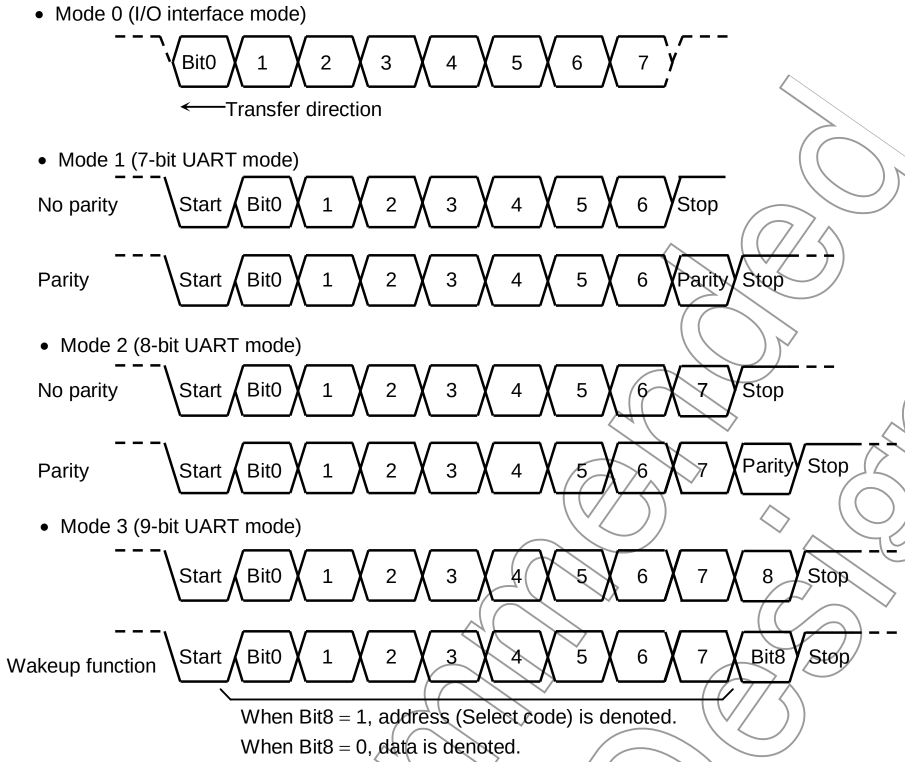


Figure 3.9.1 Data Formats

Not Recommended for New Design

Downloaded from Arrow.com.

3.9.1 Block Diagrams

Figure 3.9.2 is a block diagram representing serial channel 0.

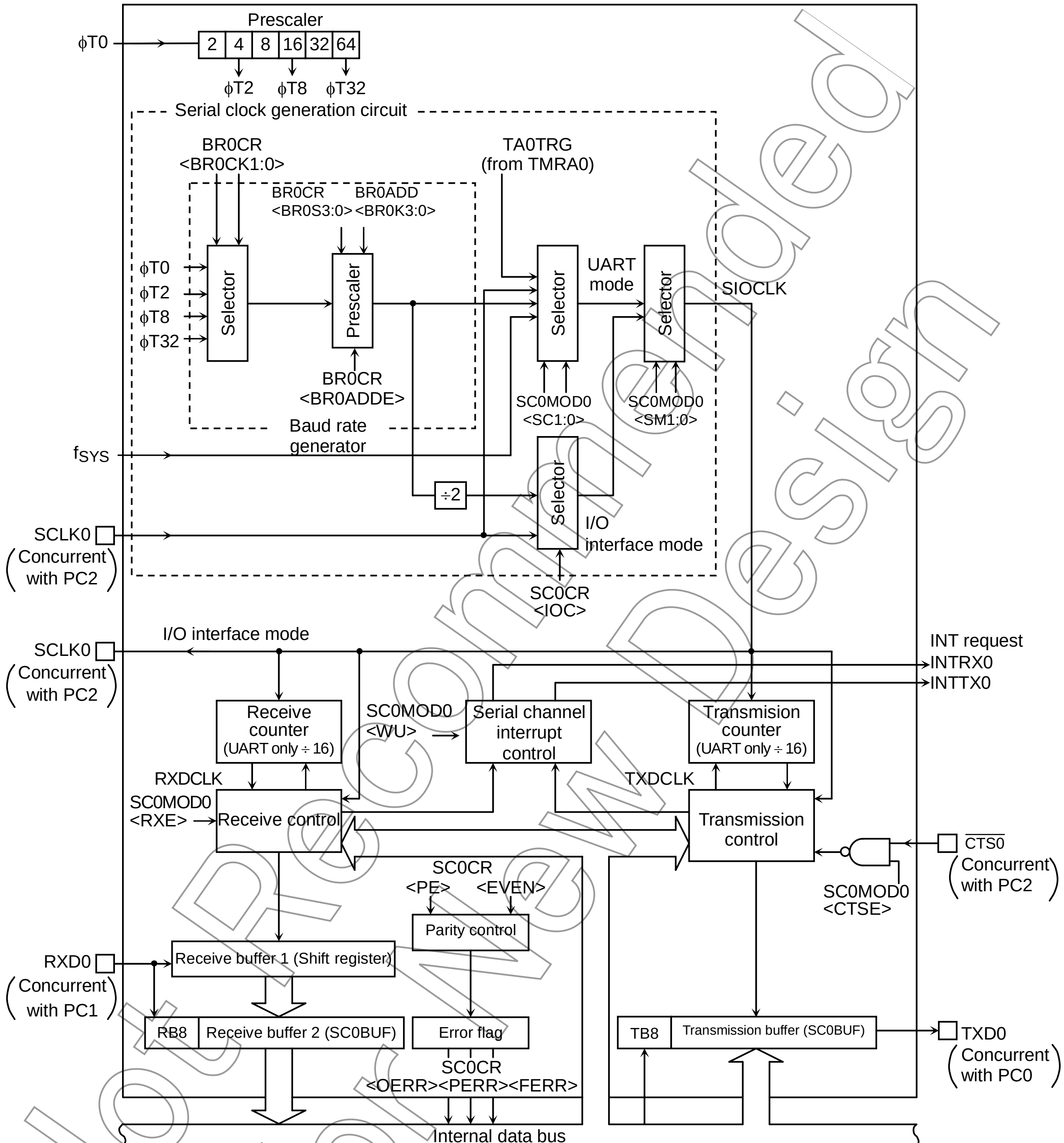


Figure 3.9.2 Block Diagram of the Serial Channel 0 (SIO0)

Not Recommended for New Design

Downloaded from Arrow.com.

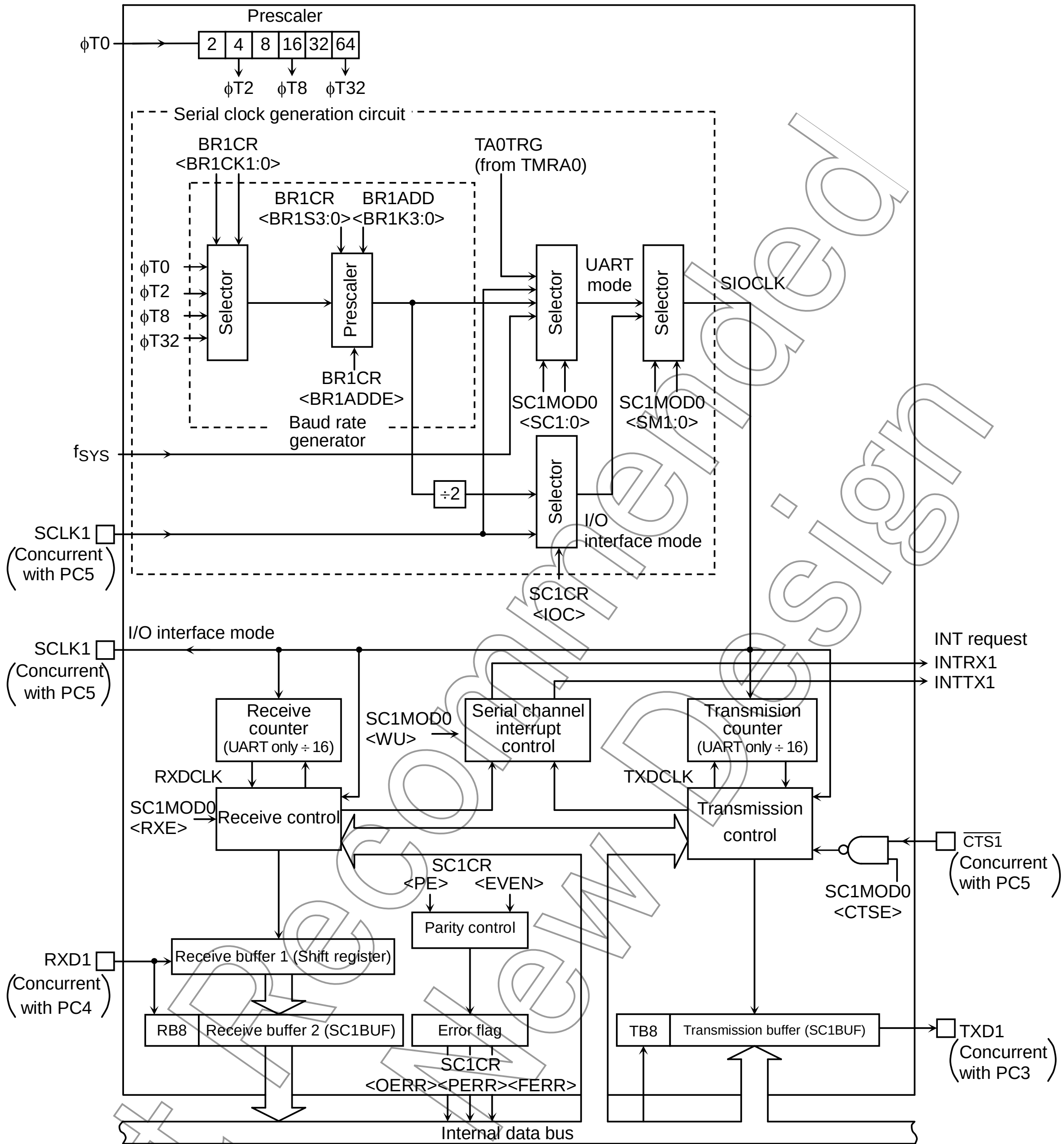


Figure 3.9.3 Block Diagram of the Serial Channel 1 (SIO1)

Not Recommended for New Design

Downloaded from Arrow.com.

3.9.2 Operation of Each Circuit

(1) Prescaler

There is a 6-bit prescaler for generating a clock to SIO0. The clock selected using SYSCR<PRCK1:0> is divided by 4 and input to the prescaler as φT0. The prescaler can be run by selecting the baud rate generator as the serial transfer clock.

Table 3.9.2 shows prescaler clock resolution into the baud rate generator.

Table 3.9.2 Prescaler Clock Resolution to Baud Rate Generator

Select System Clock SYSCR1 <SYSCK>	Select Prescaler Clock SYSCR0 <PRCK1:0>	Gear Value SYSCR1<GEAR2:0>	Prescaler Output Clock Resolution			
			φT0	φT2	φT8	φT32
1 (fs)	00 (f_{FPH})	XXX	2^2/fs	2^4/fs	2^6/fs	2^8/fs
0 (fc)		000 (fc)	2^2/fc	2^4/fc	2^6/fc	2^8/fc
		001 (fc/2)	2^3/fc	2^5/fc	2^7/fc	2^9/fc
		010 (fc/4)	2^4/fc	2^6/fc	2^8/fc	2^{10}/fc
		011 (fc/8)	2^5/fc	2^7/fc	2^9/fc	2^{11}/fc
		100 (fc/16)	2^6/fc	2^8/fc	2^{10}/fc	2^{12}/fc
	10 (fc/16 clock)	XXX	–	2^8/fc	2^{10}/fc	2^{12}/fc

X: Don't care, –: Cannot be used

The baud rate generator selects between 4-clock inputs: φT0, φT2, φT8, and φT32 among the prescaler outputs.

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Baud rate generator

The baud rate generator is the circuit which generates transmission and receiving clocks which determine the transfer rate of the serial channels.

The input clock to the baud rate generator, ϕT0, ϕT2, ϕT8 or ϕT32, is generated by the 6-bit prescaler which is shared by the timers. One of these input clocks is selected using the BR0CR<BR0CK1:0> field in the baud rate generator control register.

The baud rate generator includes a frequency divider, which divides the frequency by 1 or N + (16 – K)/16 to 16 values, determining the transfer rate.

The transfer rate is determined by the settings of BR0CR<BR0ADDE, BR0S3:0> and BR0ADD<BR0K3:0>.

- In UART mode

(1) When BR0CR<BR0ADDE> = 0

The settings BR0ADD<BR0K3:0> are ignored. The baud rate generator divides the selected prescaler clock by N, which is set in BR0CK<BR0S3:0>. (N = 1, 2, 3 ... 16)

(2) When BR0CR<BR0ADDE> = 1

The N + (16 – K)/16 division function is enabled. The baud rate generator divides the selected prescaler clock by N + (16 – K)/16 using the value of N set in BR0CR<BR0S3:0> (N = 2, 3 ... 15) and the value of K set in BR0ADD<BR0K3:0> (K = 1, 2, 3 ... 15)

Note: If N = 1 or N = 16, the N + (16 – K)/16 division function is disabled. Set BR0CR<BR0ADDE> to 0.

- In I/O interface mode

The N + (16 – K)/16 division function is not available in I/O interface mode. Set BR0CR<BR0ADDE> to 0 before dividing by N.

The method for calculating the transfer rate when the baud rate generator is used is explained below.

- In UART mode

$$\text{Baud rate} = \frac{\text{Input clock of baud rate generator}}{\text{Frequency divider for baud rate generator}} \div 16$$

- In I/O interface mode

$$\text{Baud rate} = \frac{\text{Input clock of baud rate generator}}{\text{Frequency divider for baud rate generator}} \div 2$$

Not Recommended for New Design

Downloaded from Arrow.com.

- Integer divider (N divider)

For example, when the source clock frequency (fc) = 12.288 MHz, the input clock frequency = ϕT2 (fc/16), the frequency divider N (BR0CR<BR0S3:0>) = 5, and BR0CR<BR0ADDE> = 0, the baud rate in UART mode is as follows:

* Clock state

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: System clock

$$\text{Baud rate} = \frac{fc/16}{5} \div 16$$

$$= 12.288 \times 10^6 \div 16 \div 5 \div 16 = 9600 \text{ (bps)}$$

Note: The N + (16 – K)/16 division function is disabled and setting BR0ADD<BR0K3:0> is invalid.

- N + (16 – K)/16 divider (UART mode only)

Accordingly, when the source clock frequency (fc) = 4.8 MHz, the input clock frequency = ϕT0, the frequency divider N (BR0CR<BR0S3:0>) = 7, K (BR0ADD<BR0K3:0>) = 3, and BR0CR <BR0ADDE> = 1, the baud rate in UART mode is as follows:

* Clock state

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: System clock

$$\text{Baud rate} = \frac{fc/4}{7 + (16 - 3)/16} \div 16$$

$$= 4.8 \times 10^6 \div 4 \div (7 + 13/16) \div 16 = 9600 \text{ (bps)}$$

Table 3.9.3 show examples of UART mode transfer rates.

Additionally, the external clock input is available in the serial clock (Serial channels 0, 1). The method for calculating the baud rate is explained below:

- In UART mode

  Baud rate = External clock input frequency ÷ 16

  It is necessary to satisfy (External clock input cycle) ≥ 4/fc

- In I/O interface mode

  Baud rate = External clock input frequency

  It is necessary to satisfy (External clock input cycle) ≥ 16/fc

Not Recommended for New Design

Downloaded from Arrow.com.

Table 3.9.3 Transfer Rate Selection

(when baud rate generator Is used and BR0CR<BR0ADDE> = 0)

Unit (kbps)

fc [MHz]	Input Clock / Frequency Divider (set to BR1CR<BR1S3:0>)	ϕT0	ϕT2	ϕT8	ϕT32
9.830400	2	76.800	19.200	4.800	1.200
↑	4	38.400	9.600	2.400	0.600
↑	8	19.200	4.800	1.200	0.300
↑	0	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
↑	A	19.200	4.800	1.200	0.300
14.745600	2	115.200	28.800	7.200	1.800
↑	3	76.800	19.200	4.800	1.200
↑	6	38.400	9.600	2.400	0.600
↑	C	19.200	4.800	1.200	0.300
19.6608	1	307.200	76.800	19.200	4.800
↑	2	153.600	38.400	9.600	2.400
↑	4	76.800	19.200	4.800	1.200
↑	8	38.400	9.600	2.400	0.600
↑	10	19.200	4.800	1.200	0.300
22.1184	3	115.200	28.800	7.200	1.800
24.576	1	384.000	96.000	24.000	6.000
↑	2	192.000	48.000	12.000	3.000
↑	4	96.000	24.000	6.000	1.500
↑	5	76.800	19.200	4.800	1.200
↑	8	48.000	12.000	3.000	0.750
↑	A	38.400	9.600	2.400	0.600
↑	10	24.000	6.000	1.500	0.375
27.0336	B	38.400	9.600	2.400	0.600
29.4912	1	460.800	115.200	28.800	7.200
↑	3	153.600	38.400	9.600	2.400
↑	4	115.200	28.800	7.200	1.800
↑	6	76.800	19.200	4.800	1.200
↑	9	51.200	12.800	3.200	0.800
↑	C	38.400	9.600	2.400	0.600
↑	F	30.720	7.680	1.920	0.480
↑	10	28.800	7.200	1.800	0.450
31.9488	D	38.400	9.600	2.400	0.600

Note 1: Transfer rates in I/O interface mode are eight times faster than the values given above.

Note 2: The values in this table are calculated for when fc is selected as the system clock, the clock gear is set for fc/1 and the system clock is the prescaler clock input f_{FPH}.

Timer out clock (TA0TRG) can be used for source clock of UART mode only.

Calculation method the frequency of TA0TRG

Frequency of TA0TRG = Baud rate × 16

Note 1:The TMRA0 match detects signal cannot be used as the transfer clock in I/O interface mode.

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Serial clock generation circuit

This circuit generates the basic clock for transmitting and receiving data.

- In I/O interface mode

In SCLK output mode with the setting SC0CR<IOC> = 0, the basic clock is generated by dividing the output of the baud rate generator by 2, as described previously.

In SCLK input mode with the setting SC0CR<IOC> = 1, the rising edge or falling edge will be detected according to the setting of the SC0CR<SCLKS> register to generate the basic clock.

- In UART mode

The SC0MOD0<SC1:0> setting determines whether the baud rate generator clock, the internal system clock f_{SYS}, the match detect signal from timer TMRA0 or the external clock (SCLK0) is used to generate the basic clock SIOCLK.

(4) Receiving counter

The receiving counter is a 4-bit binary counter used in UART mode which counts up the pulses of the SIOCLK clock. It takes 16 SIOCLK pulses to receive 1 bit of data; each data bit is sampled three times – on the 7th, 8th and 9th clock cycles.

The value of the data bit is determined from these three samples using the majority rule.

For example, if the data bit is sampled respectively as 1, 0 and 1 on 7th, 8th and 9th clock cycles, the received data bit is taken to be 1. A data bit sampled as 0, 0 and 1 is taken to be 0.

(5) Receiving control

- In I/O interface mode

In SCLK output mode with the setting SC0CR<IOC> = 0, the RXD0 signal is sampled on the rising edge or falling edge of the shift clock which is output on the SCLK0 pin, according to the SC0CR<SCLKS> setting.

In SCLK input mode with the setting SC0CR<IOC> = 1, the RXD0 signal is sampled on the rising or falling edge of the SCLK0 input, according to the SC0CR<SCLKS> setting.

- In UART mode

The receiving control block has circuit which detects a start bit using the majority rule. Received bits are sampled three times; when two or more out of three samples are 0, the bit is recognized as the start bit and the receiving operation commences.

The values of the data bits that are received are also determined using the majority rule.

Not Recommended for New Design

Downloaded from Arrow.com.

(6) The receiving buffers

To prevent overrun errors, the receiving buffers are arranged in a double-buffer structure.

Received data is stored one bit at a time in receiving buffer 1 (which is a shift register). When 7 or 8 bits of data have been stored in receiving buffer 1, the stored data is transferred to receiving buffer 2 (SC0BUF); this cause an INTRX0 interrupt to be generated. The CPU only reads receiving buffer 2 (SC0BUF). Even before the CPU reads receiving buffer 2 (SC0BUF), the received data can be stored in receiving buffer 1. However, unless receiving buffer 2 (SC0BUF) is read before all bits of the next data are received by receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of receiving buffer 1 will be lost, although the contents of receiving buffer 2 and SC0CR<RB8> will be preserved.

SC0CR<RB8> is used to store either the parity bit – added in 8-bit UART mode – or the most significant bit (MSB) – in 9-bit UART mode.

In 9-bit UART mode the wakeup function for the slave controller is enabled by setting SC0MOD0<WU> to 1; in this mode INTRX0 interrupts occur only when the value of SC0CR<RB8> is 1.

(7) Transmission counter

The transmission counter is a 4-bit binary counter which is used in UART mode and which, like the receiving counter, counts the SIOCLK clock pulses; a TXDCLK pulse is generated every 16 SIOCLK clock pulses.

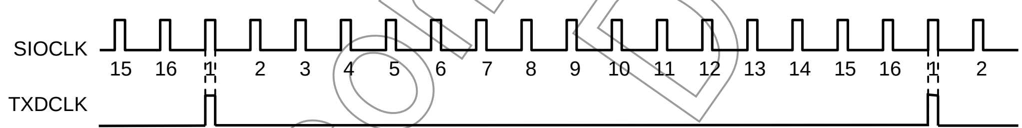


Figure 3.9.4 Generation of the Transmission Clock

(8) Transmission controller

- In I/O interface mode

In SCLK output mode with the setting SC0CR<IOC> = 0, the data in the transmission buffer is output one bit at a time to the TXD0 pin on the rising edge or falling edge of the shift clock which is output on the SCLK0 pin, according to the SC0CR<SCLKS> setting..

In SCLK input mode with the setting SC0CR<IOC> = 1, the data in the transmission buffer is output one bit at a time on the TXD0 pin on the rising or falling edge of the SCLK0 input, according to the SC0CR<SCLKS> setting.

- In UART mode

When transmission data sent from the CPU is written to the transmission buffer, transmission starts on the rising edge of the next TXDCLK.

Not Recommended for New Design

Downloaded from Arrow.com.

Handshake function

Use of $\overline{\text{CTS}}$ pin allows data can be sent in units of one frame; thus, overrun errors can be avoided. The handshake functions is enabled or disabled by the SC0MOD<CTSE> setting.

When the $\overline{\text{CTS0}}$ pin goes high on completion of the current data send, data transmission is halted until the $\overline{\text{CTS0}}$ pin goes low again. However, the INTTX0 interrupt is generated, it requests the next data send to the CPU. The next data is written in the transmission buffer and data sending is halted.

Though there is no RTS pin, a handshake function can be easily configured by setting any port assigned to be the $\overline{\text{RTS}}$ function. The $\overline{\text{RTS}}$ should be output high to request send data halt after data receive is completed by software in the $\overline{\text{RXD}}$ interrupt routine.

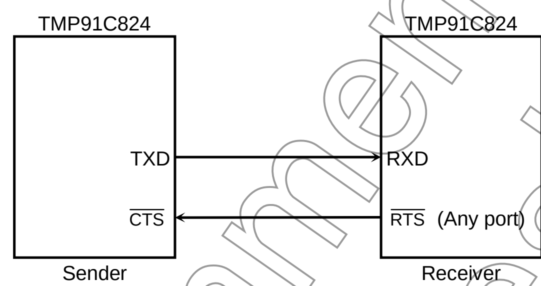
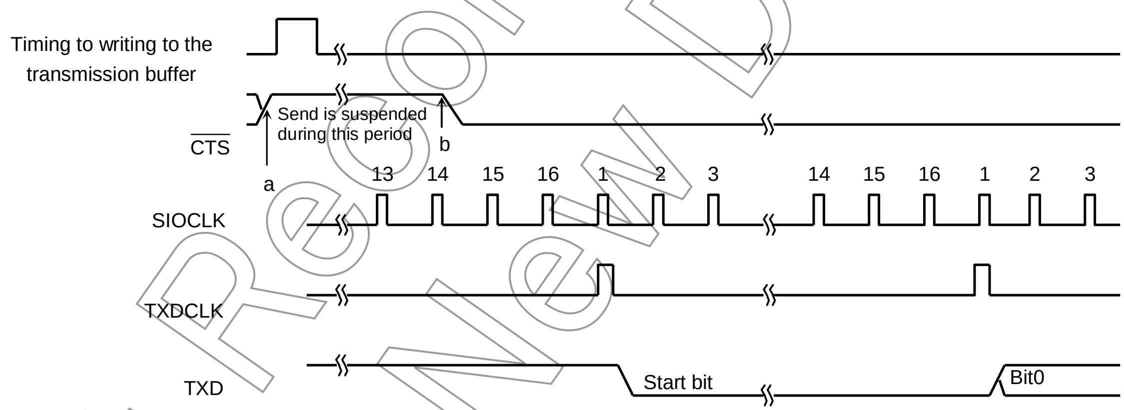


Figure 3.9.5 Handshake Function



Note 1: If the $\overline{\text{CTS}}$ signal goes high during transmission, no more data will be sent after completion of the current transmission.

Note 2: Transmission starts on the first falling edge of the TXDCLK clock after the $\overline{\text{CTS}}$ signal has fallen.

Figure 3.9.6 $\overline{\text{CTS}}$ (Clear to send) Timing

Not Recommended for New Design

Downloaded from Arrow.com.

(9) Transmission buffer

The transmission buffer (SC0BUF) shifts out and sends the transmission data written from the CPU form the least significant bit (LSB) in order. When all the bits are shifted out, the transmission buffer becomes empty and generates an INTTX0 interrupt.

(10) Parity control circuit

When SC0CR<PE> in the serial channel control register is set to 1, it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART mode or 8-bit UART mode. The SC0CR<EVEN> field in the serial channel control register allows either even or odd parity to be selected.

In the case of transmission, parity is automatically generated when data is written to the transmission buffer SC0BUF. The data is transmitted after the parity bit has been stored in SC0BUF<TB7> in 7-bit UART mode or in SC0MOD0<TB8> in 8-bit UART mode. SC0CR<PE> and SC0CR<EVEN> must be set before the transmission data is written to the transmission buffer.

In the case of receiving, data is shifted into receiving buffer 1, and the parity is added after the data has been transferred to receiving buffer 2 (SC0BUF), and then compared with SC0BUF<RB7> in 7-bit UART mode or with SC0CR<RB8> in 8-bit UART mode. If they are not equal, a parity error is generated and the SC0CR<PERR> flag is set.

(11) Error flags

Three error flags are provided to increase the reliability of data reception.

1. Overrun error <OERR>

   If all the bits of the next data item have been received in receiving buffer 1 while valid data still remains stored in receiving buffer 2 (SC0BUF), an overrun error is generated.

   The below is a recommended flow when the overrun error is generated.

   (INTRX interrupt routine)

   1) Read receiving buffer

   2) Read error flag

   3) If <OERR> = 1

      then

      a) Set to disable receiving (Write 0 to SC0MOD0<RXE>)

      b) Wait to terminate current frame

      c) Read receiving buffer

      d) Read error flag

      e) Set to enable receiving (Write 1 to SC0MOD0<RXE>)

      f) Request to transmit again

   4) Other

2. Parity error <PERR>

   The parity generated for the data shifted into receiving buffer 2 (SC0BUF) is compared with the parity bit received via the RXD pin. If they are not equal, a parity error is generated.

3. Framing error <FERR>

   The stop bit for the received data is sampled three times around the center. If the majority of the samples are 0, a Framing error is generated.

Not Recommended for New Design

Downloaded from Arrow.com.

(12) Timing generation

a. In UART mode

Receiving

Mode	9 Bits (Note)	8 Bits + Parity (Note)	8 Bits, 7 Bits + Parity, 7 Bits
Interrupt timing	Center of last bit (Bit8)	Center of last bit (Parity bit)	Center of stop bit
Framing error timing	Center of stop bit	Center of stop bit	Center of stop bit
Parity error timing	–	Center of last bit (Parity bit)	Center of stop bit
Overrun error timing	Center of last bit (Bit8)	Center of last bit (Parity bit)	Center of stop bit

Note: In 9-Bit and 8-Bit+Parity mode, interrupts coincide with the ninth bit pulse.Thus, when servicing the interrupt, it is necessary to wait for a 1-bit period (to allow the stop bit to be transferred) to allow checking for a framing error.

Transmitting

Mode	9 Bits	8 Bits + Parity	8 Bits, 7 Bits + Parity, 7 Bits
Interrupt timing	Just before stop bit is transmitted	Just before stop bit is transmitted	Just before stop bit is transmitted

b. I/O interface

Transmission interrupt timing	SCLK output mode	Immediately after the last bit. (See Figure 3.9.19.)
	SCLK input mode	Immediately after rise of last SCLK signal rising mode, or immediately after fall in falling mode. (See Figure 3.9.20.)
Receiving interrupt timing	SCLK output mode	Timing used to transfer received to data receive buffer 2 (SC0BUF) (e.g., immediately after last SCLK). (See Figure 3.9.21.)
	SCLK input mode	Timing used to transfer received data to receive buffer 2 (SC0BUF) (e.g., immediately after last SCLK). (See Figure 3.9.22.)

Not Recommended for New Design

Downloaded from Arrow.com.

3.9.3 SFRs

SC0MOD0 (0202H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data bit8	Handshake 0: CTS disable 1: CTS enable	Receive function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transmission clock (UART) 00: TMRA0 trigger 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clcok (SCLK0 input)	

Serial transmission clock source (UART)

00	Timer TMRA0 match detect signal
01	Baud rate generator
10	Internal clock f_{SYS}
11	External clock (SCLK0 input)

Note: The clock selection for the I/O interface mode is controlled by the serial control register (SC0CR).

Serial transmission mode

00	I/O Interface mode	
01	UART mode	7-bit mode
10		8-bit mode
11		9-bit mode

Wakeup function

	9-bit UART	Other modes
0	Interrupt generated when data is received	Don't care
1	Interrupt generated only when RB8 = 1	

Receiving function

0	Receive disabled
1	Receive enabled

Handshake function ($\overline{CTS}$ pin)

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

Figure 3.9.7 Serial Mode Control Register (SIO0, SC0MOD0)

Not Recommended for New Design

Downloaded from Arrow.com.

SC1MOD0 (020AH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data bit8	Handshake 0: CTS disable 1: CTS enable	Receive function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transmission clock (UART) 00: TMRA0 trigger 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clcok (SCLK1 input)	

Serial transmission clock source (for UART)

00	Timer TMRA0 match detect signal
01	Baud rate generator
10	Internal clock f_{SYS}
11	External clock (SCLK1 input)

Note: The clock selection for the I/O interface mode is controlled by the serial control register (SC1CR).

Serial transmission mode

00	I/O Interface mode	
01	UART mode	7-bit mode
10		8-bit mode
11		9-bit mode

Wakeup function

	9-bit UART	Other modes
0	Interrupt generated when data is received	Don’t care
1	Interrupt generated only when RB8 = 1	

Receiving function

0	Receive disabled
1	Receive enabled

Handshake function ($\overline{CTS}$ pin)

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

Figure 3.9.8 Serial Mode Control Register (SIO1, SC1MOD0)

Not Recommended for New Design

Downloaded from Arrow.com.

SC0CR (0201H)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error Overrun	Parity	Framing	0: SCLK0 1: SCLK0	0: Baud rate generator 1: SCLK0 pin input

I/O interface input clock selection

0	Baud rate generator
1	SCLK0 pin input

Edge selection for SCLK pin (I/O mode)

0	Transmits and receivers data on rising edge of SCLK0.
1	Transmits and receivers data on falling edge SCLK0.

Framing error flag
Parity error flag
Overrun error flag
Cleared to 0 when read

Parity addition enable

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data 8

Note: As all error flags are cleared after reading do not test only a single bit with a bit-testing instruction.

Figure 3.9.9 Serial Control Register (SIO0, SC0CR)

Not Recommended for New Design

Downloaded from Arrow.com.

SC1CR
(0209H)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to 0 when)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error Overrun	Parity	Framing	0: SCLK1 1: SCLK1	0: Baud rate generator 1: SCLK1 pin input

I/O interface input clock select

0	Baud rate generator
1	SCLK1 pin input

Edge selection for SCKL pin (I/O mode)

0	Transmits and receives data on rising edge of SCLK1.
1	Transmits and receives data on falling edge of SCLK1.

Framing error flag
Parity error flag
Overrun error flag
Cleared to 0 when read

Parity addition enable

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data bit8

Note: As all error flags are cleared after reading do not test only a single bit with a bit-testing instruction.

Figure 3.9.10 Serial Control Register (SIO1, SC1CR)

Not Recommended for New Design

Downloaded from Arrow.com.

BR0CR (0203H)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Always write 0	+ (16 – K)/16 division 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting the divided frequency "N" (0 to F)			

+ (16 – K)/16 division enable

0	Disable
1	Enable

Setting the input clock of baud rate generator

00	Internal clock ϕT0
01	Internal clock ϕT2
10	Internal clock ϕT8
11	Internal clock ϕT32

BR0ADD (0204H)

	7	6	5	4	3	2	1	0
Bit symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets the frequency divisor "K" (Divided by N + (16 – K)/16)			

Sets baud rate generator frequency divisor

BR0ADD <BR0K3:0> \ BR0CR <BR0S3:0>	BR0CR<BR0ADDE> = 1: 0000 (N = 16) or 0001 (N = 1)	BR0CR<BR0ADDE> = 1: 0010 (N = 2) to 1111 (N = 15)	BR0CR<BR0ADDE> = 0: 0001 (N = 1) (UART only) to 1111(N = 15) 0000(N = 16)
0000	Disable	Disable	Divided by N
0001(K = 1) to 1111(K = 15)	Disable	Divided by N + (16 – K)/16	Divided by N

Note1: Availability of +(16-K)/16 division function

N	UART Mode	I/O Mode
2 to 15	○	×
1, 16	×	×

The baud rate generator can be set "1" in UART mode and disable +(16-K)/16 division function.Don't use in I/O interface mode.

Note2: Set BR0CR <BR0ADDE> to 1 after setting K (K = 1 to 15) to BR0ADD<BR0K3:0> when +(16-K)/16 division function is used. Writes to unused bits in the BR0ADD register do not affext operation, and undefined data is read from these unused bits.

Figure 3.9.11 Baud Rate Generator Control (SIO0, BR0CR, BR0ADD)

Not Recommended for New Design

Downloaded from Arrow.com.

BR1CR (020BH)

	7	6	5	4	3	2	1	0
Bit symbol	–	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Always write 0	+ (16 − K)/16 division 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting the divided frequency “N” (0 to F)			

+ (16 − K)/16 division enable

0	Disabled
1	Enabled

Input clock selection for baud rate generator

00	Internal clock ϕT0
01	Internal clock ϕT2
10	Internal clock ϕT8
11	Internal clock ϕT32

BR1ADD (020CH)

	7	6	5	4	3	2	1	0
Bit symbol					BR1K3	BR1K2	BR1K1	BR1K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets the frequency divisor “K” (Divided by N + (16 − K)/16)			

Baud rate generator frequency divisor setting

	BR1CR<BR1ADDE> = 1		BR1CR<BR1ADDE> = 0
BR0CR <BR1S3:0> / BR1ADD <BR1K3:0>	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) (UART only) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	Divided by N
0001 (K = 1) to 1111 (K = 15)	Disable	Disabled by N + (16 − K)/16	

Note1: Availability of +(16-K)/16 division function

N	UART Mode	I/O Mode
2 to 15	○	×
1, 16	×	×

The baud rate generator can be set “1” in UART mode and disable +(16-K)/16 division function.Don’t use in I/O interface mode.

Note2: Set BR1CR <BR1ADDE> to 1 after setting K (K = 1 to 15) to BR1ADD<BR1K3:0> when +(16-K)/16 division function is used. Writes to unused bits in the BR0ADD register do not affext operation, and undefined data is read from these unused bits.

Figure 3.9.12 Baud Rate Generator Control (SIO1, BR1CR, BR1ADD)

Not Recommended for New Design

Downloaded from Arrow.com.

SC0BUF (0200H)

	7	6	5	4	3	2	1	0	
	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receiving)

Note: Prohibit read modify write for SC0BUF.

Figure 3.9.13 Serial Transmission/Receiving Buffer Registers (SIO0, SC0BUF)

SC0MOD1 (0205H)

	7	6	5	4	3	2	1	0
Bit symbol	I2S0	FDPX0						
Read/Write	R/W	R/W						
After reset	0	0						
Function	IDLE2 0: Stop 1: Run	Duplex 0: Half 1: Full						

Figure 3.9.14 Serial Mode Control Register 1 (SIO0, SC0MOD1)

Not Recommended for New Design

Downloaded from Arrow.com.

SC1BUF (0208H)

	7	6	5	4	3	2	1	0	
	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receiving)

Note: Prohibit read modify write for SC1BUF.

Figure 3.9.15 Serial Transmission/Receiving Buffer Registers (SIO1, SC1BUF)

SC1MOD1 (020DH)

	7	6	5	4	3	2	1	0
Bit symbol	I2S1	FDPX1						
Read/Write	R/W	R/W						
After reset	0	0						
Function	IDLE2 0: Stop 1: Run	Duplex 0: Half 1: Full						

Figure 3.9.16 Serial Mode Control Register 1 (SIO1, SC1MOD1)

Not Recommended for New Design

Downloaded from Arrow.com.

3.9.4 Operation in Each Mode

(1) Mode 0 (I/O interface mode)

This mode allows an increase in the number of I/O pins available for transmitting data to or receiving data from an external shift register.

This mode includes the SCLK output mode to output synchronous clock SCLK and SCLK input mode to input external synchronous clock SCLK.

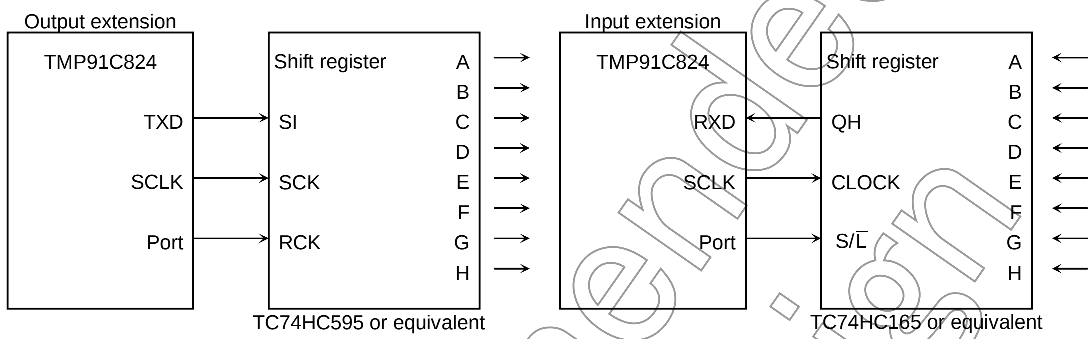


Figure 3.9.17 SCLK Output Mode Connection Example

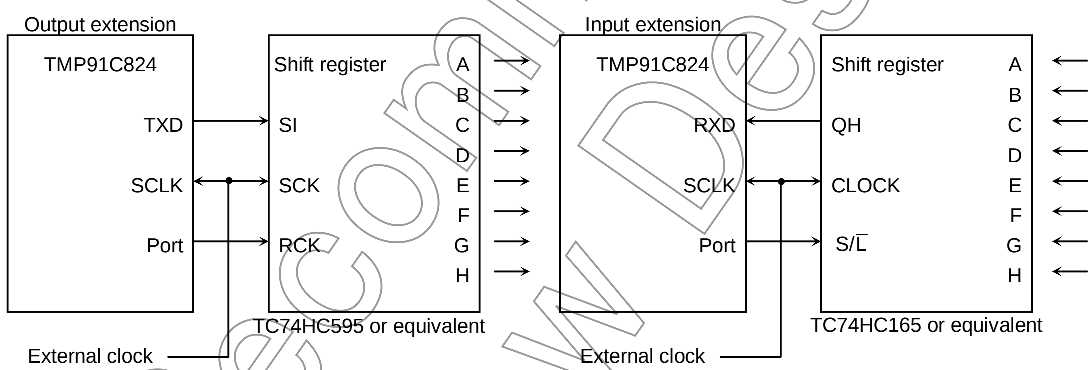


Figure 3.9.18 SCLK Input Mode Connection Example

Not Recommended for New Design

Downloaded from Arrow.com.

a. Transmission

In SCLK output mode 8-bit data and a synchronous clock are output on the TXD0 and SCLK0 pins respectively each time the CPU writes the data to the transmission buffer. When all data is output, INTES0<ITX0C> will be set to generate the INTTX0 interrupt.

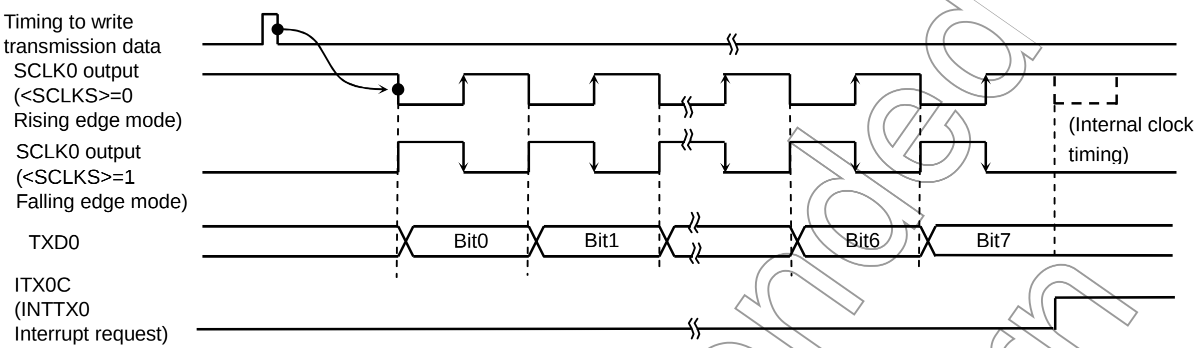


Figure 3.9.19 Transmitting Operation in I/O Interface Mode (SCLK0 output mode)

In SCLK input mode, 8-bit data is output on the TXD0 pin when the SCLK0 input becomes active after the data has been written to the transmission buffer by the CPU.

When all data is output, INTES0<ITX0C> will be set to generate INTTX0 interrupt.

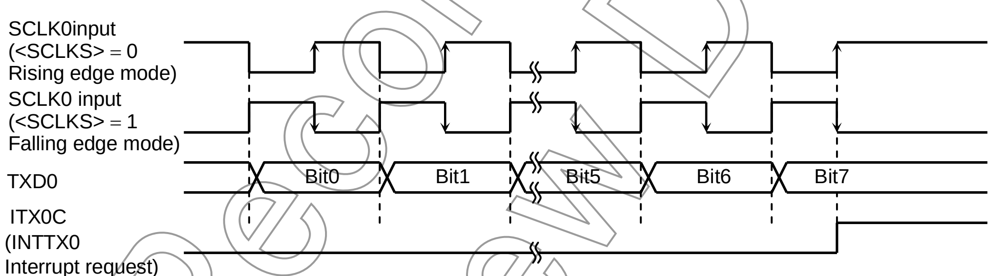


Figure 3.9.20 Transmitting Operation in I/O Interface Mode (SCLK0 input mode)

Not Recommended for New Design

Downloaded from Arrow.com.

b. Receiving

In SCLK output mode, the synchronous clock is outputted from SCLK0 pin and the data is shifted to receiving buffer 1. This starts when the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred to receiving buffer 2 (SC0BUF according to the timing shown below) and INTES0<IRX0C> will be set to generate INTRX0 interrupt.

The outputting for the first SCLK0 starts by setting SC0MOD0<RXE> to 1.

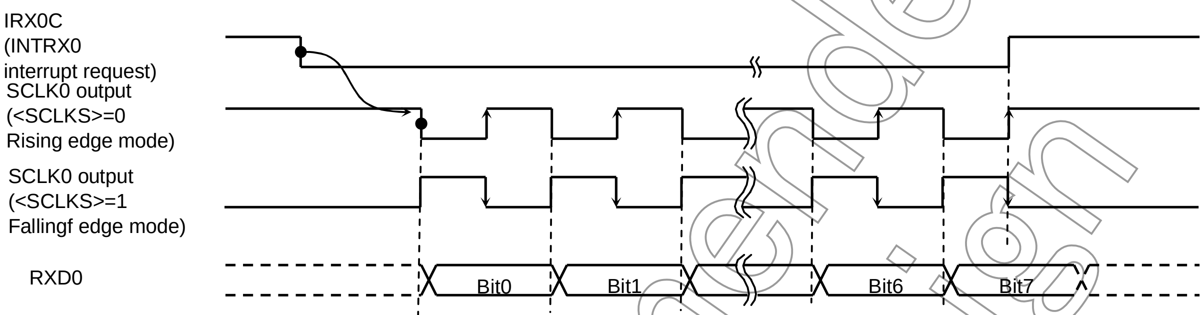


Figure 3.9.21 Receiving Operation in I/O Interface Mode (SCLK0 output mode)

In SCLK input mode, the data is shifted to receiving buffer 1 when the SCLK input becomes active after the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data is received, the data will be shifted to receiving buffer 2 (SC0BUF according to the timing shown below) and INTES0<IRX0C> will be set again to be generate INTRX0 interrupt.

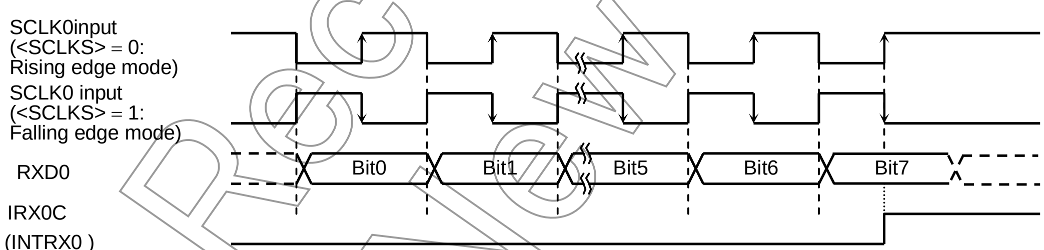


Figure 3.9.22 Receiving Operation in I/O Interface Mode (SCLK0 input mode)

Note: The system must be put in the receive enable state (SCMOD0<RXE> = 1) before data can be received.

Not Recommended for New Design

Downloaded from Arrow.com.

c. Transmission and receiving (Full duplex mode)

When the full duplex mode is used, set the level of Receive Interrupt to 0 and set enable the interrupt level (1 to 6) to the transfer interrupt. In the transfer interrupt program, The receiving operation should be done like the above example before setting the next transfer data.

Example: Channel 0, SCLK output
Baud rate = 9600 bps
fc = 14.7456 MHz

* Clock state

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

Main routine

	7	6	5	4	3	2	1	0	
INTES0	–	0	0	1	–	0	0	0	Set the INTTX0 level to 1. Set the INTRX0 level to 0.
PCCR	–	–	–	–	–	1	0	1	Set PC0, PC1 and PC2 to function as the TXD0, RXD0 and SCLK0 pins respectively.
PCFC	X	X	–	X	–	1	X	1	
SC0MOD0	–	–	–	–	0	0	–	–	Select I/O interface mode.
SC0MOD1	1	1	X	X	X	X	X	X	Select full duplex mode.
SC0CR	–	–	–	–	–	–	0	–	SCLK output, transmit on negative edge, receive on positive edge
BR0CR	0	0	1	1	0	0	1	1	Baud rate = 9600 bps
SC0MOD0	–	–	1	–	–	–	–	–	Enable receiving
SC0BUF	*	*	*	*	*	*	*	*	Set the transmit data and start.
INTTX0 interrupt routine									
Acc SC0BUF									Read the receiving buffer.
SC0BUF	*	*	*	*	*	*	*	*	Set the next transmit data.

X: Don't care, –: No change

Downloaded from Arrow.com.

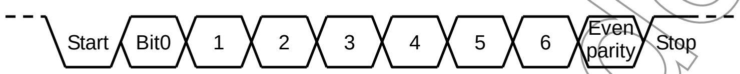
Not Recommended for New Design

(2) Mode 1 (7-bit UART mode)

7-bit UART mode is selected by setting serial channel mode register SC0MOD0<SM1:0> to 01.

In this mode, a parity bit can be added. Use of a parity bit is enabled or disabled by the setting of the serial channel control register SC0CR<PE> bit; whether even parity or odd parity will be used is determined by the SC0CR<EVEN> setting when SC0CR<PE> is set to 1 (Enabled).

Example: When transmitting data of the following format, the control registers should be set as described below. This explanation applies to channel 0.



Transmission direction (Transmission rate: 2400 bps at fc = 12.288 MHz)

* Clock state

System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: System clock

		7	6	5	4	3	2	1	0	
PCCR	←	X	X	–	–	–	–	–	1	Set PC0 to function as the TXD0 pin.
PCFC	←	X	X	–	X	–	–	X	1	
SC0MOD0	←	–	–	–	–	0	1	0	1	Select 7-bit UART mode.
SC0CR	←	–	1	1	–	–	–	–	–	Add even parity.
BR0CR	←	0	0	1	0	0	1	0	1	Set the transfer rate to 2400 bps.
INTES0	←	–	1	0	0	–	–	–	–	Enable the INTTX0 interrupt and set it to interrupt level 4.
SC0BUF	←	*	*	*	*	*	*	*	*	Set data for transmission.

X: Don't care, –: No change

(3) Mode 2 (8-bit UART mode)

8-bit UART mode is selected by setting SC0MOD0<SM1:0> to 10. In this mode, a parity bit can be added (use of a parity bit is enabled or disabled by the setting of SC0CR<PE>); whether even parity or odd parity will be used is determined by the SC0CR<EVEN> setting when SC0CR<PE> is set to 1 (Enabled).

Example: When receiving data of the following format, the control registers should be set as described below.

Transmission direction (Transmission rate: 9600 bps at fc = 12.288 MHz)

Not Recommended for New Design

Downloaded from Arrow.com.

* Clock state

System clock:	High frequency (fc)
Clock gear:	1 (fc)
Prescaler clock:	System clock

Main settings

		7	6	5	4	3	2	1	0	
PCCR	←	X	X	–	–	–	–	0	–	Set PC1 to function as the RXD0 pin.
SC0MOD0	←	–	–	1	–	1	0	0	1	Enable receiving in 8-bit UART mode.
SC0CR	←	–	0	1	–	–	–	–	–	Add even parity.
BR0CR	←	0	0	0	1	0	1	0	1	Set the transfer rate to 9600 bps.
INTES0	←	–	–	–	–	–	1	0	0	Enable the INTTX0 interrupt and set it to interrupt level 4.

Interrupt processing

Acc	←	SC0CR AND 00011100	Check for errors.
if Acc	≠	0 then ERROR	
Acc	←	SC0BUF	Read the received data.

X: Don't care, –: No change

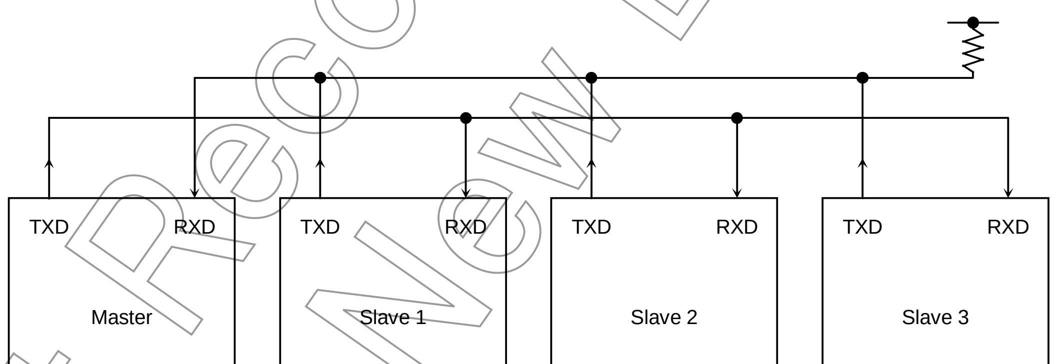
(4) Mode 3 (9-bit UART mode)

9-bit UART mode is selected by setting SC0MOD0<SM1:0> to 11. In this mode parity bit cannot be added.

In the case of transmission the MSB (9th bit) is written to SC0MOD0<TB8>. In the case of receiving it is stored in SC0CR<RB8>. When the buffer is written and read, the MSB is read or written first, before the rest of the SC0BUF data.

Wakeup function

In 9-bit UART mode, the wakeup function for slave controllers is enabled by setting SC0MOD0<WU> to 1. The interrupt INTRX0 occurs only when<RB8> = 1.



Note: The TXD pin of each slave controller must be in Open-drain output mode.

Figure 3.9.23 Serial Link Using Wakeup Function

Not Recommended for New Design

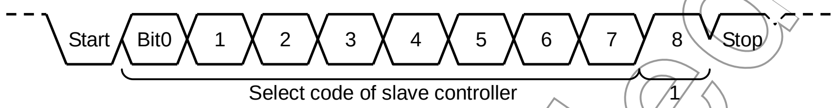
Downloaded from Arrow.com.

Protocol

(1) Select 9-bit UART mode on the master and slave controllers.

(2) Set the SC0MOD0<WU> bit on each slave controller to 1 to enable data receiving.

(3) The master controller transmits one-frame data including the 8-bit select code for the slave controllers. The MSB (Bit8) <TB8> is set to 1.



(4) Each slave controller receives the above frame. Each controller checks the above select code against its own select code. The controller whose code matches clears its WU bit to 0.

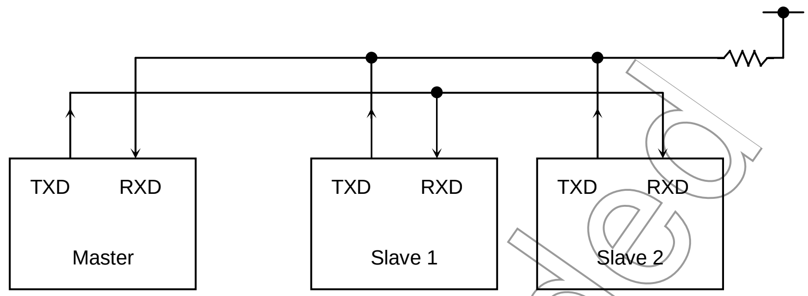
(5) The master controller transmits data to the specified slave controller whose SC0MOD<WU> bit is cleared to 0. The MSB (Bit8) <TB8> is cleared to 0.

(6) The other slave controllers (whose <WU> bits remain at 1) ignore the received data because their MSBs (Bit8 or <RB8>) are set to 0, disabling INTRX0 interrupts.
The slave controller (WU bit = 0) can transmit data to the master controller, and it is possible to indicate the end of data receiving to the master controller by this transmission.

Not Recommended for New Design

Downloaded from Arrow.com.

Example: To link two slave controllers serially with the master controller using the internal clock f_{SYS} as the transfer clock.



Since serial channels 0 and 1 operate in exactly the same way, channel 0 only is used for the purposes of this explanation.

- Setting the master controller

```
Main

PCCR    ← X X – – – – 0 1  } Set PC0 and PC1 to function as the TXD0 and RXD0 pins
PCFC    ← X X – X – – X 1  } respectively.
INTES0  ← – 1 0 0 – 1 0 1    Enable the INTTX0 interrupt and set it to interrupt level 4.
                             Enable the INTRX0 interrupt and set it to interrupt level 5.
SC0MOD0 ← 1 – 1 – 1 1 1 0    Set fSYS as the transmission clock for 9-bit UART mode.
SC0BUF  ← 0 0 0 0 0 0 0 1    Set the select code for slave controller 1.

INTTX0 interrupt

SC0MOD0 ← 0 – – – – – – –    Set TB8 to 0.
SC0BUF  ← * * * * * * * *    Set data for transmission.
```

- Setting the slave controller

```
Main

PCCR    ← X X – – – – 0 1  }
PCFC    ← X X – X – – X 1  } Set PC1 to RXD0 and PC0 to TXD0 (Open-drain output).
PCODE   ← X X X X – X X 1  }
INTES0  ← – 1 0 1 – 1 1 0    Enable INTRX0 and INTTX0.
SC0MOD0 ← – – 1 1 1 1 1 0    Set <WU> to 1 in 9-bit UART transmission mode using fSYS
                             as the transfer clock.

INTRX0 interrupt

Acc ← SC0BUF
if Acc = select code
Then SC0MOD0 ← – – – 0 – – – – Clear <WU> to 0.
```

Not Recommended for New Design

Downloaded from Arrow.com.

3.9.5 Support for IrDA

SIO0 includes support for the IrDA 1.0 infrared data communication specification. Figure 3.9.24 shows the block diagram.

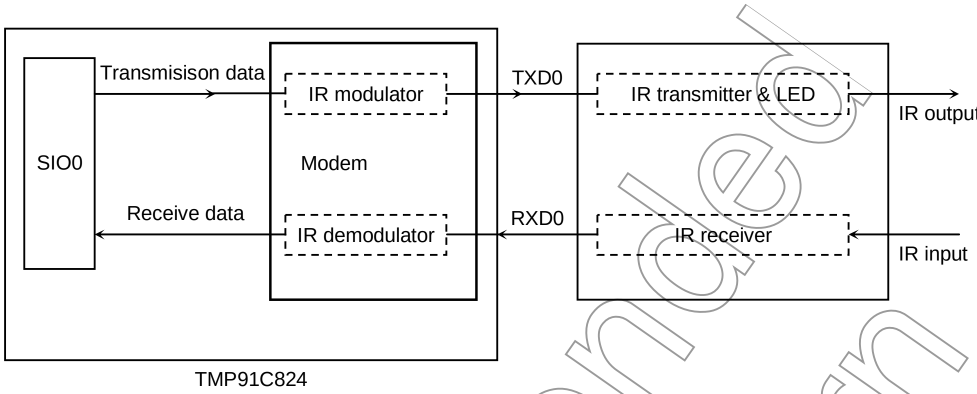


Figure 3.9.24 IrDA Block Diagram

(1) Modulation of the transmission data

When the transfer data is 0, the modem outputs 1 to TXD0 pin with either 3/16 or 1/16 times for width of baud rate. The pulse width is selected by the SIRCR<PLSEL>. When the transfer data is 1, the modem outputs 0.

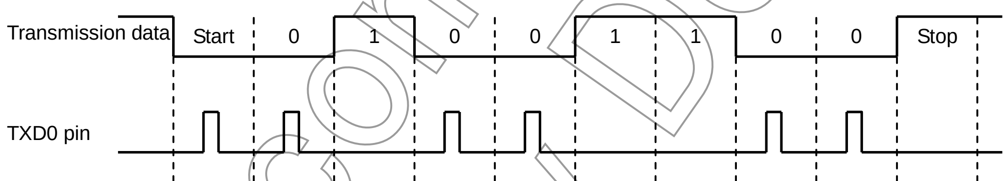


Figure 3.9.25 Modulation Example of Transfer Data

(2) Modulation of the receive data

When the receive data has the effective high level pulse width (Software selectable), the modem outputs 0 to SIO0. Otherwise the modem outputs 1 to SIO0. The receive pulse logic is also selectable by SIRCR<RXSEL>.

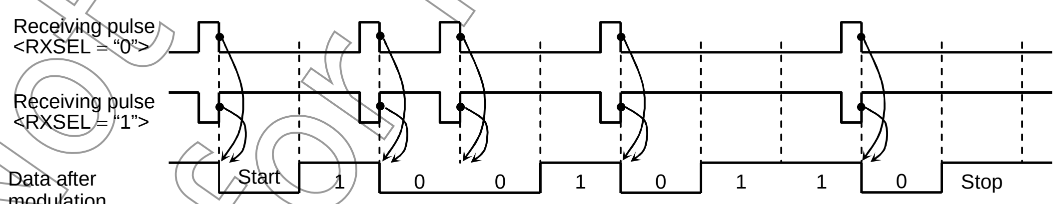


Figure 3.9.26 Demodulation Example of Receive Data

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Data format

The data format is fixed as follows:

- Data length: 8-bit
- Parity bits: none
- Stop bits: 1

Any other settings don't guarantee the normal operation.

(4) SFR

Figure 3.9.27 shows the control register SIRCR. Set the data SIRCR during SIO0 is inhibited (Both TXEN and RXEN of this register should be set to 0).

Any changing for this register during transmission or receiving operation don't guarantee the normal operation.

The following example describes how to set this register:

```
1)  SIO setting                ; Set the SIO to UART Mode.
        ↓
2)  LD (SIRCR), 07H            ; Set the receive data pulse width to 16×.
3)  LD (SIRCR), 37H            ; TXEN, RXEN Enable the Transmission and receiving of SIO.
        ↓
4)  Start transmission         ; The modem operates as follows:
    and receiving for SIO0     • SIO0 starts transmitting.
                               • IR receiver starts receiving.
```

(5) Notes

1) Baud rate generator for IrDA

To generate baud rate for IrDA, use baud rate generator in SIO0 by setting 01 to SC0MOD0<SC1:0>. To use another source (TA0TRG, f_{SYS} and SCLK0 input) are not allowed.

2) As the IrDA 1.0 physical layer specification, the data transfer speed and infra-red pulse width is specified.

Table 3.9.4 Baud Rate and Pulse Width Specifications

Baud Rate	Modulation	Rate Tolerance (% of rate)	Pulse Width (Minimum)	Pulse Width (Typical)	Pulse Width (Maximum)
2.4 kbps	RZI	±0.87	1.41 μs	78.13 μs	88.55 μs
9.6 kbps	RZI	±0.87	1.41 μs	19.53 μs	22.13 μs
19.2 kbps	RZI	±0.87	1.41 μs	9.77 μs	11.07 μs
38.4 kbps	RZI	±0.87	1.41 μs	4.88 μs	5.96 μs
57.6 kbps	RZI	±0.87	1.41 μs	3.26 μs	4.34 μs
115.2 kbps	RZI	±0.87	1.41 μs	1.63 μs	2.23 μs

The infra-red pulse width is specified either baud rate T × 3/16 or 1.6 μs (1.6 μs is equal to 3/16 pulse width when baud rate is 115.2 kbps).

The TMP91C824FG has the function selects the pulse width on the transmission either 3/16 or 1/16. But 1/16 pulse width can be selected when the baud rate is equal or less than 38.4 kbps only. When 57.6 kbps and 115.2 kbps, the output pulse width should not be set to T × 1/16.

Not Recommended for New Design

Downloaded from Arrow.com.

As the same reason, + (16 − k)/16 division functions in the baud rate generator of SIO0 can not be used to generate 115.2 kbps baud rate.

Also when the 38.4 kbps and 1/16 pulse width, + (16 − k)/16 division function can not be used. Table 3.9.5 shows Baud rate and pulse width for (16 − k)/16 division function.

Table 3.9.5 Baud Rate and Pulse Width for (16 – k)/16 Division Function

Pulse Width	Baud Rate					
	115.2 kbps	57.6 kbps	38.4 kbps	19.2 kbps	9.6 kbps	2.4 kbps
T × 3/16	×	○	○	○	○	○
T × 1/16	–	–	×	○	○	○

○: Can be used (16 – k)/16 division function
×: Can not be used (16 – k)/16 division function
–: Can not be set to 1/16 pulse width

Not Recommended for New Design

Downloaded from Arrow.com.

SIRCR (0207H)

	7	6	5	4	3	2	1	0
Bit symbol	PLSEL	RXSEL	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Select transmit pulse width 0: 3/16 1: 1/16	Receive data 0: H pulse 1: L pulse	Transmit 0: Disable 1: Enable	Receive 0: Disable 1: Enable	Select receive pulse width Set effective pulse width for equal or more than $2x \times (\text{value} + 1) + 100\text{ns}$ Can be set: 1 to 4 Can not be set: 0, 15			

Select receive pulse width
Formula: Effective pulse width $\geq 2x \times (\text{Value} + 1) + 100\text{ns}$
$x = 1/f_{FPH}$

0000	Cannot be set
0001 to 1110	Equal or more than 4x + 100 ns Equal or more than 30x + 100 ns
1111	Can not be set

Receive operation

0	Disabled
1	Enabled

Transmit operation

0	Disabled
1	Enabled

Select transmit pulse width

0	3/16
1	1/16

Figure 3.9.27 IrDA Control Register

Not Recommended for New Design

Downloaded from Arrow.com.

3.10 Serial Bus Interface (SBI)

The TMP91C824FG has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit SIO mode and an I²C bus mode.

The serial bus interface is connected to an external device through P71 (SDA) and P72 (SCL) in the I²C bus mode; and through P70 (SCK), P71 (SO) and P72 (SI) in the clocked-synchronous 8-bit SIO mode.

Each pin is specified as follows.

	P7ODE<ODE72:71>	P7CR<P72C:70C>	P7FC<P72F:70F>
I²C bus mode	11	11X	11X
Clocked synchronous 8-bit SIO mode	XX	011 010	111

X: Don't care

3.10.1 Configuration

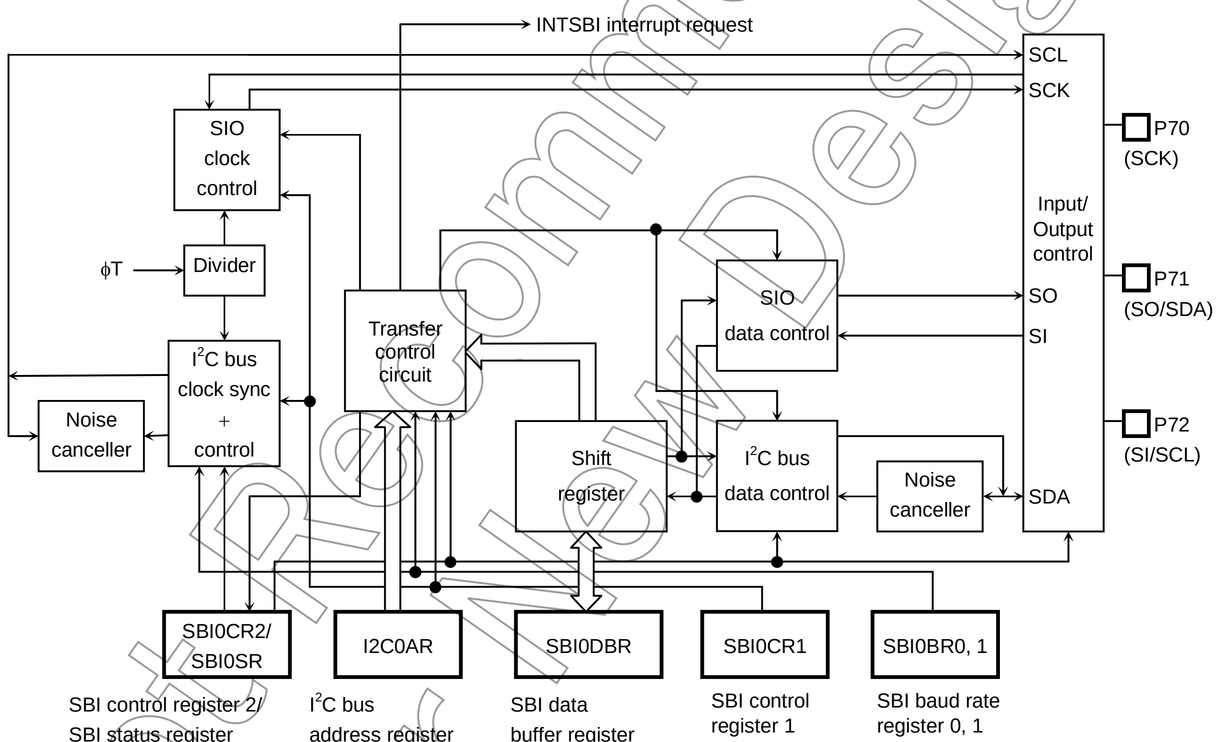


Figure 3.10.1 Serial Bus Interface (SBI)

Downloaded from Arrow.com.

3.10.2 Serial Bus Interface (SBI) Control

The following registers are used to control the serial bus interface and monitor the operation status.

- Serial bus interface control register 1 (SBI0CR1)
- Serial bus interface control register 2 (SBI0CR2)
- Serial bus interface data buffer register (SBI0DBR)
- I²C bus address register (I2C0AR)
- Serial bus interface status register (SBI0SR)
- Serial bus interface baud rate register 0 (SBI0BR0)
- Serial bus interface baud rate register 1 (SBI0BR1)

The above registers differ depending on a mode to be used.

Refer to section 3.10.4 "I²C Bus Mode Control" and 3.10.7 "Clocked Synchronous 8-Bit SIO Mode Control".

3.10.3 The Data Formats in the I²C Bus Mode

The data formats in the I²C bus mode is shown below.

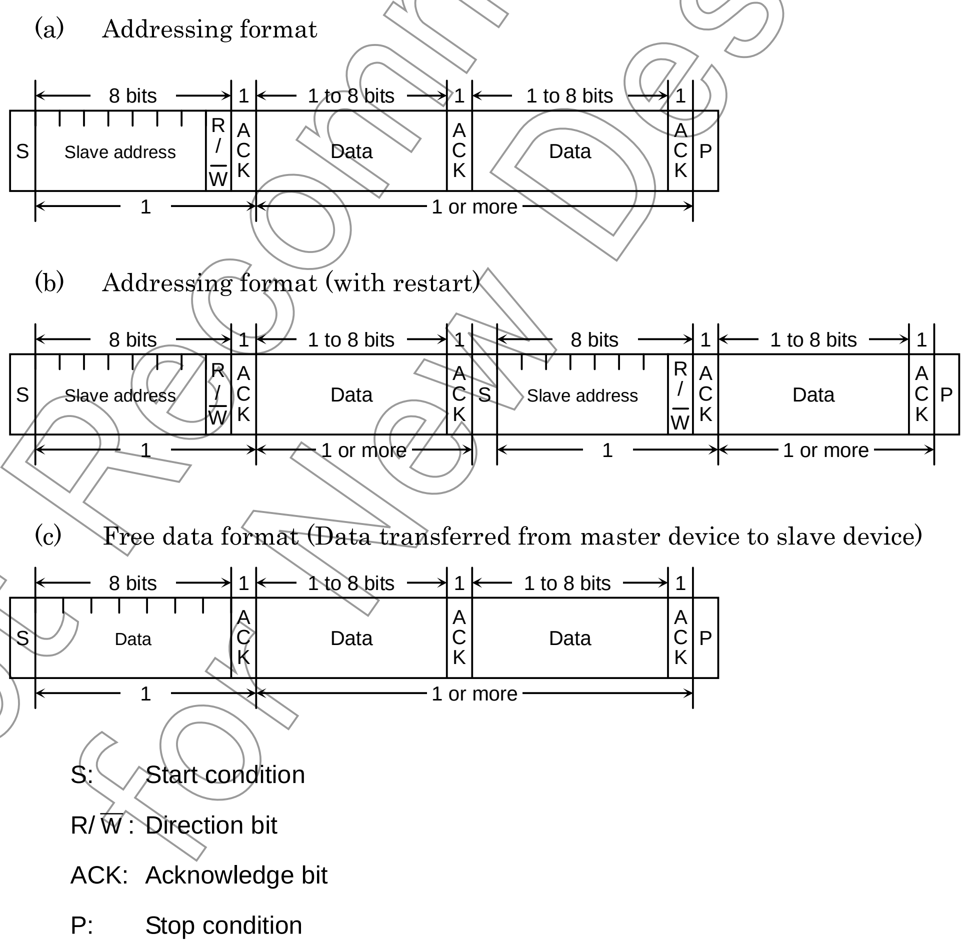


Figure 3.10.2 Data Format in the I²C Bus Mode

Downloaded from Arrow.com.

3.10.4 I²C Bus Mode Control

The following registers are used to control and monitor the operation status when using the serial bus interface (SBI) in the I²C bus mode.

Seirial Bus Interface Conrol Register 1

		7	6	5	4	3	2	1	0
SBI0CR1 (0240H)	Bit symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
	Read/Write	W			R/W		W		R/W
	After reset	0	0	0	0		0	0	0/1 (Note 3)
Prohibit read-modify-write	Function	Number of transferred bits (Note 1)			Acknowledge mode specification 0: Not generate 1: Generate		Internal serial clock selection and software reset monitor (Note 2)		

Internal serial clock selection <SCK2:0> at write

000	n = 5	- Note 4	System clock: fc Clock gear: fc/1 fc = 33 MHz (internal SCL output) $fscl = \frac{fc}{2^n + 8}$ [Hz]
001	n = 6	- Note 4	
010	n = 7	- Note 4	
011	n = 8	- Note 4	
100	n = 9	63.5 kHz	
101	n = 10	32.0 kHz	
110	n = 11	16.1 kHz	
111		(Reserved)	

Software reset state monitor <SWRMON> at read

0	During software reset
1	Initial data

Acknowledge mode specification

0	Not generate clock pulse for acknowledge signal
1	Generate clock pulse for acknowledge signal

Number of bits transferred

<BC2:0>	<ACK> = 0		<ACK> = 1	
	Number of clock pulses	Bits	Number of clock pulses	Bits
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

Note 1: Set the <BC2:0> to 000 before switching to a clock-synchronous 8-bit SIO mode.

Note 2: For the frequency of the SCL line clock, see 3.10.5 (3) Serial clock.

Note 3: Initial data of SCK0 is "0", SWRMON is "1".

Note 4: This I²C bus circuit does not support fast mode, it supports standard mode only. Although the I²C bus circuit itself allows the setting of a baud rate over 100kbps, the compliance with the I²C specification is not guraranteed in that case.

Figure 3.10.3 Registers for the I²C Bus Mode

Not Recommended for New Design

Downloaded from Arrow.com.

Serial Bus Interface Control Register 2

		7	6	5	4	3	2	1	0
SBI0CR2 (0243H)	Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
	Read/Write	W				W (Note 1)		W (Note 1)	
	After reset	0	0	0	1	0	0	0	0
Prohibit read-modify-write	Function	Master/slave selection	Transmitter/ receiver selection	Start/stop condition generation	Cancel INTSBI interrupt request	Serial bus interface operating mode selection (Note2) 00: Port mode 01: SIO mode 10: I^2C bus mode 11: (Reserved)		Software reset generate write 10 and 01, then an internal reset signal is generated.	

Serial bus interface operating mode selection (Note 2)

00	Port mode (Serial bus interface output disabled)
01	Clocked synchronous 8-bit SIO mode
10	I^2C bus mode
11	(Reserved)

INTSBI interrupt request

0	Don’t care
1	Cancel interrupt request

Start/stop condition generation

0	Generates the stop condition
1	Generates the start condition

Transmitter/receiver selection

0	Receiver
1	Transmitter

Master/slave selection

0	Slave
1	Master

Note 1: Reading this register function as SBI0SR register.

Note 2: Switch a mode to port mode after confirming that the bus is free.
Switch a mode between I^2C bus mode and clock-synchronous 8-bit SIO mode after confirming that input signals via port are high level.

Figure 3.10.4 Registers for the I^2C Bus Mode

Not Recommended for New Design

Downloaded from Arrow.com.

Serial Bus Interface Status Register

		7	6	5	4	3	2	1	0
SBI0SR (0243H)	Bit symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
	Read/Write	R							
	After reset	0	0	0	1	0	0	0	0
Prohibit read-modify-write	Function	Master/ slave status monitor	Transmitter/ receiver status monitor	I^2C bus status monitor	INTSBI interrupt request monitor	Arbitration lost detection monitor 0: – 1: Detected	Slave address match detection monitor 0: Undetected 1: Detected	GENERAL CALL detection monitor 0: Undetected 1: Detected	Last received bit monitor 0: 0 1: 1

Last received bit monitor

0	Last received bit was 0
1	Last received bit was 1

GENERAL CALL detection monitor

0	Undetected
1	GENERAL CALL detected

Slave address match detection monitor

0	Undetected
1	Slave address match or GENERAL CALL detected

Arbitration lost detection monitor

0	–
1	Arbitration lost

INTSBI interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I^2C bus status monitor

0	Free
1	Busy

Transmitter/receiver status monitor

0	Receiver
1	Transmitter

Master/slave status monitor

0	Slave
1	Master

Note: Writing in this register functions as SBI0CR2.

Figure 3.10.5 Registers for the I^2C Bus Mode

Not Recommended for New Design

Downloaded from Arrow.com.

Serial Bus Interface Baud Rate Regster 0

SBI0BR0 (0244H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	–	I2SBI0						
Read/Write	W	R/W						
After reset	0	0						
Function	Always write 0	IDLE2 0: Stop 1: Run						

Operation during IDLE 2 mode

0	Stop
1	Operation

Serial Bus Interface Baud Rate Register 1

SBI0BR1 (0245H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Operate	Always write 0						

Baud rate clock control

0	Stop
1	Operate

Sirial Bus Interface Data Buffer Register

SBI0DBR (0241H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Received)/W (Transfer)							
After reset	Undefined							

Note 1: When writing transmitted data, start from the MSB (Bit7). Receiving data is placed from LSB (Bit0).

Note 2: SBIDBR can't be read the written data. Therefore read-modify-write instruction (e.g., "BIT" instruction) is prohibitted.

Note 3: Written data in SBI0DBR is cleared by INTSBI signal.

I²C Bus Address Register

I2C0AR (0242H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Slave address selection for when device is operating as slave device							Address recognition mode specification

Address recognition mode specification

0	Slave address recognition
1	Non slave address recognition

Figure 3.10.6 Registers for the I²C Bus Mode

Downloaded from Arrow.com.

Not Recommended for New Design

3.10.5 Control in I²C Bus Mode

(1) Acknowledge mode specification

Set the SBI0CR1<ACK> to 1 for operation in the acknowledge mode. The TMP91C824 generates an additional clock pulse for an acknowledge signal when operating in master mode. In the transmitter mode during the clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low in order to generate the acknowledge signal.

Clear the <ACK> to 0 for operation in the non-acknowledge mode, The TMP91C824 does not generate a clock pulse for the acknowledge signal when operating in the master mode.

(2) Number of transfer bits

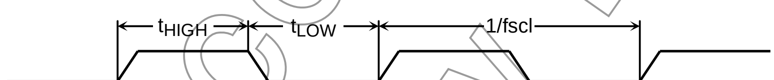
The SBI0CR1<BC2:0> is used to select a number of bits for next transmitting and receiving data.

Since the <BC2:0> is cleared to 000 as a start condition, a slave address and direction bit transmission are executed in 8 bits. Other than these, the <BC2:0> retains a specified value.

(3) Serial clock

a. Clock source

The SBI0CR1<SCK2:0> is used to select a maximum transfer frequency outputted on the SCL pin in master mode. Set a communication baud rate that meets the I²C bus specification, such as the shortest pulse width of t_{LOW}, based on the equations shown below.



$$t_{LOW} = 2^{n-1}/f_{SBI}$$
$$t_{HIGH} = 2^{n-1}/f_{SBI} + 8/f_{SBI}$$
$$fscl = 1/(t_{Low} + t_{HIGH}) = \frac{f_{SBI}}{2^n + 8}$$

SBI0CR1<SCK2:0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Note 1: f_{SBI} is the clock f_{FPH}.

Note 2: It's prohibited to use fc/16 prescaler clock when using SBI block. (I²C bus and clock synchronous.)

Figure 3.10.7 Clock Source

Not Recommended for New Design

Downloaded from Arrow.com.

b. Clock synchronization

In the I²C bus mode, in order to wired-AND a bus, a master device which pulls down a clock line to low level, in the first place, invalidate a clock pulse of another master device which generates a high-level clock pulse. The master device with a high-level clock pulse needs to detect the situation and implement the following procedure.

The TMP91C824 has a clock synchronization function for normal data transfer even when more than one master exists on the bus.

The example explains the clock synchronization procedures when two masters simultaneously exist on a bus.

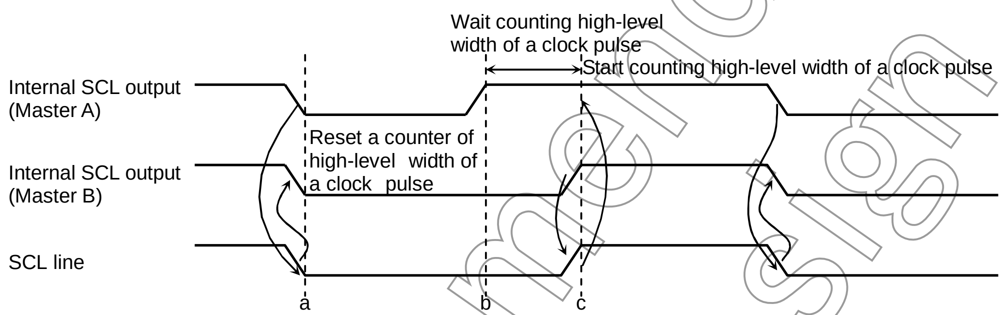


Figure 3.10.8 Clock Synchronization

As master A pulls down the internal SCL output to the low level at point a, the SCL line of the bus becomes the low level. After detecting this situation, master B resets a counter of high-level width of an own clock pulse and sets the internal SCL output to the low level.

Master A finishes counting low-level width of an own clock pulse at point b and sets the internal SCL output to the high level. Since master B holds the SCL line of the bus at the low level, master A wait for counting high-level width of an own clock pulse. After master B finishes counting low-level width of an own clock pulse at point c and master A detects the SCL line of the bus at the high level, and starts counting high level of an own clock pulse. The clock pulse on the bus is determined by the master device with the shortest high-level width and the master device with the longest low-level width from among those master devices connected to the bus.

(4) Slave address and address recognition mode specification

When the TMP91C824 is used as a slave device, set the slave address <SA6:0> and <ALS> to the I2C0AR. Clear the <ALS> to 0 for the address recognition mode.

(5) Master/slave selection

Set the SBI0CR2<MST> to 1 for operating the TMP91C824 as a master device. Clear the SBI0CR2<MST> to 0 for operation as a slave device. The <MST> is cleared to 0 by the hardware after a stop condition on the bus is detected or arbitration is lost.

Not Recommended for New Design

Downloaded from Arrow.com.

(6) Transmitter/receiver selection

Set the SBI0CR2<TRX> to 1 for operating the TMP91C824 as a transmitter. Clear the <TRX> to 0 for operation as a receiver. When data with an addressing format is transferred in slave mode, when a slave address with the same value that an I2C0AR or a GENERAL CALL is received (All 8-bit data are 0 after a start condition), the <TRX> is set to 1 by the hardware if the direction bit ($R/\overline{W}$) sent from the master device is 1, and is cleared to 0 by the hardware if the bit is 0. In the master mode, after an acknowledge signal is returned from the slave device, the <TRX> is cleared to 0 by the hardware if a transmitted direction bit is 1, and is set to 1 by the hardware if it is 0. When an acknowledge signal is not returned, the current condition is maintained.

The <TRX> is cleared to 0 by the hardware after a stop condition on the I^{2}C bus is detected or arbitration is lost.

(7) Start/stop condition generation

When the SBI0SR<BB> is 0, slave address and direction bit which are set to SBI0DBR are output on a bus after generating a start condition by writing 1 to the SBI0CR2<MST, TRX, BB, PIN>. It is necessary to set transmitted data to the data buffer register SBI0DBR and set 1 to <ACK> beforehand.

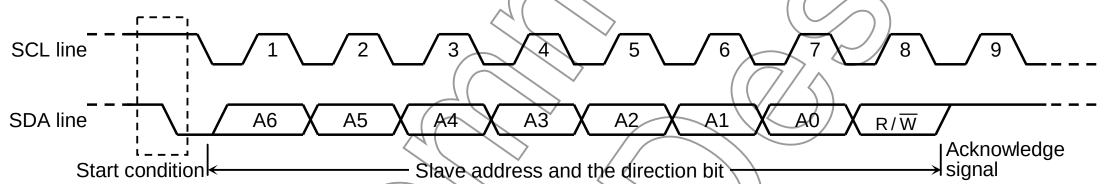


Figure 3.10.9 Start Condition Generation and Slave Address Generation

When the <BB> is 1, a sequence of generating a stop condition is started by writing 1 to the <MST, TRX, PIN>, and 0 to the <BB>. Do not modify the contents of <MST, TRX, BB, PIN> until a stop condition is generated on a bus.

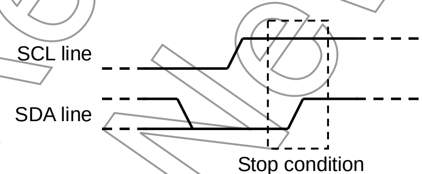


Figure 3.10.10 Stop Condition Generation

The state of the bus can be ascertained by reading the contents of SBI0SR<BB>. SBI0SR<BB> will be set to 1 if a start condition has been detected on the bus, and will be cleared to 0 if a stop condition has been detected.

And about generation of stop condition in master mode, there are some limitation points. Please refer to the 3.10.6 (4) "Stop condition generation".

Not Recommended for New Design

Downloaded from Arrow.com.

(8) Interrupt service requests and interrupt cancellation

When a serial bus interface interrupt request (INTSBI) occurs, the SBI0CR2<PIN> is cleared to 0. During the time that the SBI0CR2<PIN> is 0, the SCL line is pulled down to the low level.

The <PIN> is cleared to 0 when a 1 word of data is transmitted or received. Either writing/reading data to/from SBI0DBR sets the <PIN> to 1.

The time from the <PIN> being set to 1 until the SCL line is released takes t_{LOW}.

In the address recognition mode (<ALS> = 0), <PIN> is cleared to 0 when the received slave address is the same as the value set at the I2C0AR or when a GENERAL CALL is received (All 8-bit data are 0 after a start condition). Although SBI0CR2<PIN> can be set to 1 by the program, the <PIN> is not clear it to 0 when it is written 0.

(9) Serial bus interface operation mode selection

SBI0CR2<SBIM1:0> is used to specify the serial bus interface operation mode. Set SBI0CR2<SBIM1:0> to 10 when the device is to be used in I²C bus mode after confirming pin condition of serial bus interface to "H".

Switch a mode to port after confirming a bus is free.

(10) Arbitration lost detection monitor

Since more than one master device can exist simultaneously on the bus in I²C bus mode, a bus arbitration procedure has been implemented in order to guarantee the integrity of transferred data.

Data on the SDA line is used for I²C bus arbitration.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on the bus. Master A and master B output the same data until point a. After master A outputs "L" and master B, "H", the SDA line of the bus is wire-AND and the SDA line is pulled down to the low level by master A. When the SCL line of the bus is pulled up at point b, the slave device reads the data on the SDA line, that is, data in master A. A data transmitted from master B becomes invalid. The state in master B is called arbitration lost. Master B device which loses arbitration releases the internal SDA output in order not to affect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

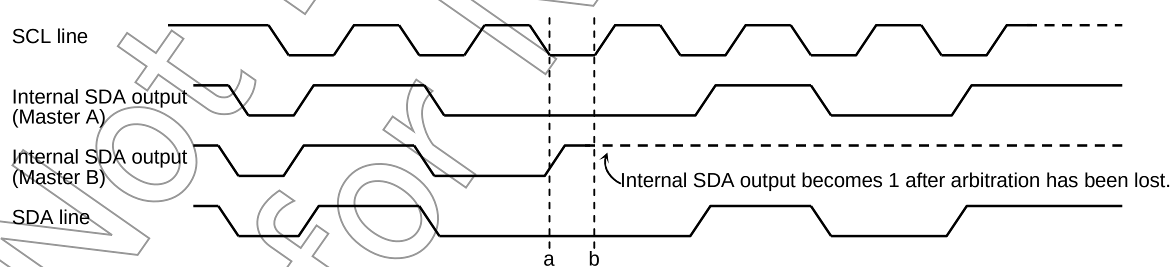


Figure 3.10.11 Arbitration Lost

Not Recommended for New Design

Downloaded from Arrow.com.

The TMP91C824 compares the levels on the bus's SDA line with those of the internal SDA output on the rising edge of the SCL line. If the levels do not match, arbitration is lost and SBI0SR<AL> is set to 1.

When SBI0SR<AL> is set to 1, SBI0SR<MST, TRX> are cleared to 00 and the mode is switched to slave receiver mode. Thus, clock output is stopped in data transfer after setting <AL> = "1".

SBI0SR<AL> is cleared to 0 when data is written to or read from SBI0DBR or when data is written to SBI0CR2.

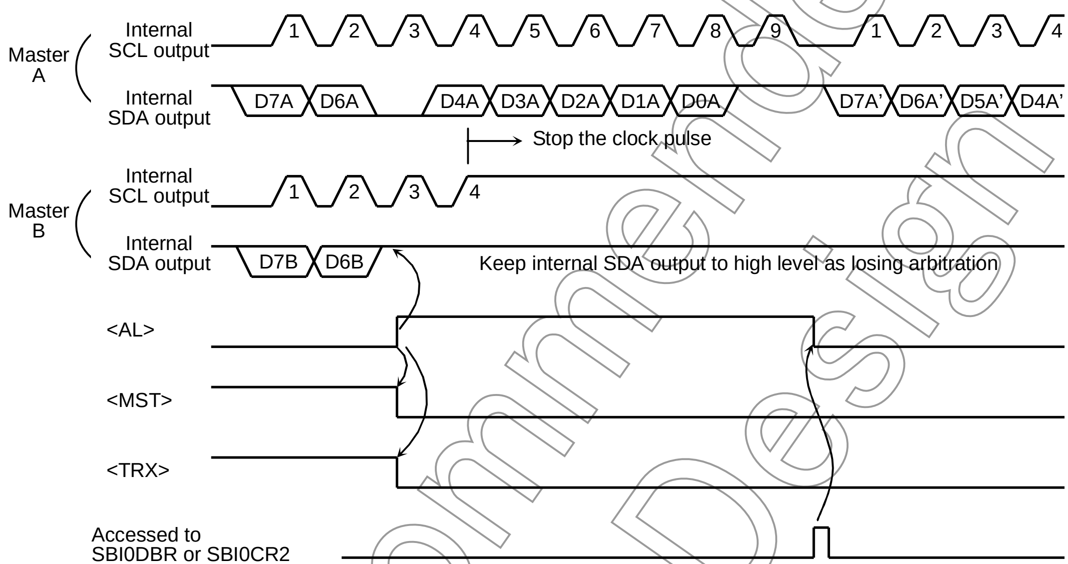


Figure 3.10.12 Example of when TMP91CW12 is a Master Device B
(D7A = D7B, D6A = D6B)

(11) Slave address match detection monitor

SBI0SR<AAS> is set to 1 in slave mode, in address recognition mode (e.g., when I2C0AR<ALS> = 0), when a GENERAL CALL is received, or when a slave address matches the value set in I2C0AR. When I2C0AR<ALS> = 1, SBI0SR<AAS> is set to 1 after the first word of data has been received. SBI0SR<AAS> is cleared to 0 when data is written to or read from the data buffer register SBI0DBR.

(12) GENERAL CALL detection monitor

SBI0SR<AD0> is set to 1 in slave mode, when a GENERAL CALL is received (All 8-bit received data is 0, after a start condition). SBI0SR<AD0> is cleared to 0 when a start condition or stop condition is detected on the bus.

(13) Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is set to the SBI0SR<LRB>. In the acknowledge mode, immediately after an INTSBI interrupt request is generated, an acknowledge signal is read by reading the contents of the SBI0SR<LRB>.

Not Recommended for New Design

Downloaded from Arrow.com.

(14) Software reset function

The software reset function is used to initialize the SBI circuit, when SBI is locked by external noises, etc.

An internal reset signal pulse can be generated by setting SBI0CR2<SWRST1:0> to 10 and 01. This initializes the SBI circuit internally. All command (except SBI0CR2<SBIM1:0>) registers and status registers are initialized as well.

SBI0CR1<SWRMON> is automatically set to “1” after the SBI circuit has been initialized.

(15) Serial bus interface data buffer register (SBI0DBR)

The received data can be read and transferred data can be written by reading or writing the SBI0DBR.

In the master mode, after the start condition is generated the slave address and the direction bit are set in this register.

(16) I²C bus address register (I2C0AR)

I2C0AR<SA6:0> is used to set the slave address when the TMP91C824 functions as a slave device.

The slave address output from the master device is recognized by setting the I2C0AR<ALS> to 0. The data format is the addressing format. When the slave address is not recognized at the <ALS> = 1, the data format is the free data format.

(17) Baud rate register (SBI0BR1)

Write 1 to SBI0BR1<P4EN> before operation commences.

(18) Setting register for IDLE2 mode operation (SBI0BR0)

SBI0BR0<I2SBI0> is the register setting operation/stop during IDLE2 mode. Therefore, setting <I2SBI0> is necessary before the HALT instruction is executed.

Not Recommended for New Design

Downloaded from Arrow.com.

3.10.6 Data Transfer in I^{2}C Bus Mode

(1) Device initialization

Set the SBI0BR1<P4EN>, SBI0CR1<ACK, SCK2:0>, Set SBI0BR1 to 1 and clear bits 7 to 5 and 3 in the SBI0CR1 to 0.

Set a slave address <SA6:0> and the <ALS> (<ALS> = 0 when an addressing format) to the I2C0AR.

For specifying the default setting to a slave receiver mode, clear 0 to the <MST, TRX, BB> and set 1 to the <PIN>, 10 to the <SBIM1:0>.

(2) Start condition and slave address generation

a. Master mode

In the master mode, the start condition and the slave address are generated as follows.

Check a bus free status (when <BB> = 0).

Set the SBI0CR1<ACK> to 1 (Acknowledge mode) and specify a slave address and a direction bit to be transmitted to the SBI0DBR.

When SBI0CR2<BB> = 0, the start condition are generated by writing 1111 to SBI0CR2<MST, TRX, BB, PIN>. Subsequently to the start condition, nine clocks are output from the SCL pin. While eight clocks are output, the slave address and the direction bit which are set to the SBI0DBR. At the 9th clock, the SDA line is released and the acknowledge signal is received from the slave device.

An INTSBI interrupt request occurs at the falling edge of the 9th clock. The <PIN> is cleared to 0. In the master mode, the SCL pin is pulled down to the low level while <PIN> is 0. When an interrupt request occurs, the <TRX> is changed according to the direction bit only when an acknowledge signal is returned from the slave device.

b. Slave mode

In the slave mode, the start condition and the slave address are received.

After the start condition is received from the master device, while eight clocks are output from the SCL pin, the slave address and the direction bit which are output from the master device are received.

When a GENERAL CALL or the same address as the slave address set in I2C0AR is received, the SDA line is pulled down to the low level at the 9th clock, and the acknowledge signal is output.

An INTSBI interrupt request occurs on the falling edge of the 9th clock. The <PIN> is cleared to 0. In slave mode the SCL line is pulled down to the low level while the <PIN> = 0.

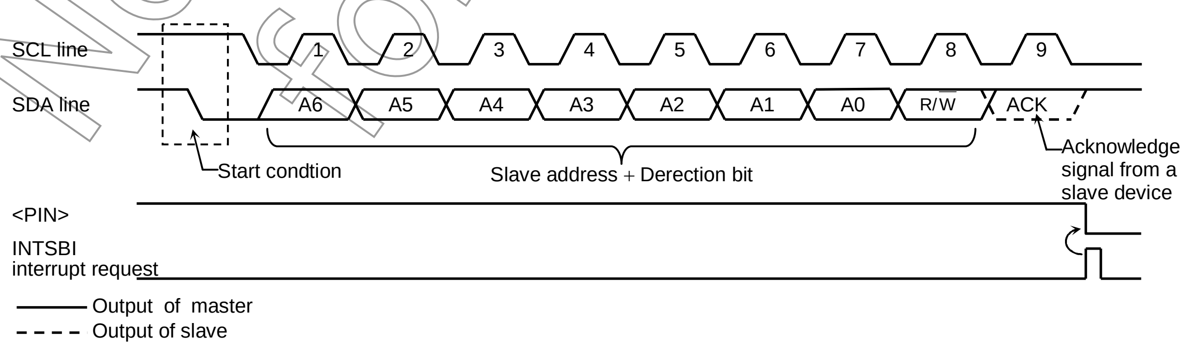


Figure 3.10.13 Start Condition Generation and Slave Address Transfer

Not Recommended for New Design

Downloaded from Arrow.com.

(3) 1-word data transfer

Check the <MST> by the INTSBI interrupt process after the 1-word data transfer is completed, and determine whether the mode is a master or slave.

a. If <MST> = 1 (Master mode)

Check the <TRX> and determine whether the mode is a transmitter or receiver.

When the <TRX> = 1 (Transmitter mode)

Check the <LRB>. When <LRB> is 1, a receiver does not request data. Implement the process to generate a stop condition (Refer to 3.10.6 (4)) and terminate data transfer.

When the <LRB> is 0, the receiver is requests new data. When the next transmitted data is 8 bits, write the transmitted data to SBI0DBR. When the next transmitted data is other than 8 bits, set the BC<2:0> <ACK> and write the transmitted data to SBI0DBR. After written the data, <PIN> becomes 1, a serial clock pulse is generated for transferring a new 1 word of data from the SCL pin, and then the 1-word data is transmitted. After the data is transmitted, an INTSBI interrupt request occurs. The <PIN> becomes 0 and the SCL line is pulled down to the low level. If the data to be transferred is more than 1 word in length, repeat the procedure from the <LRB> checking above.

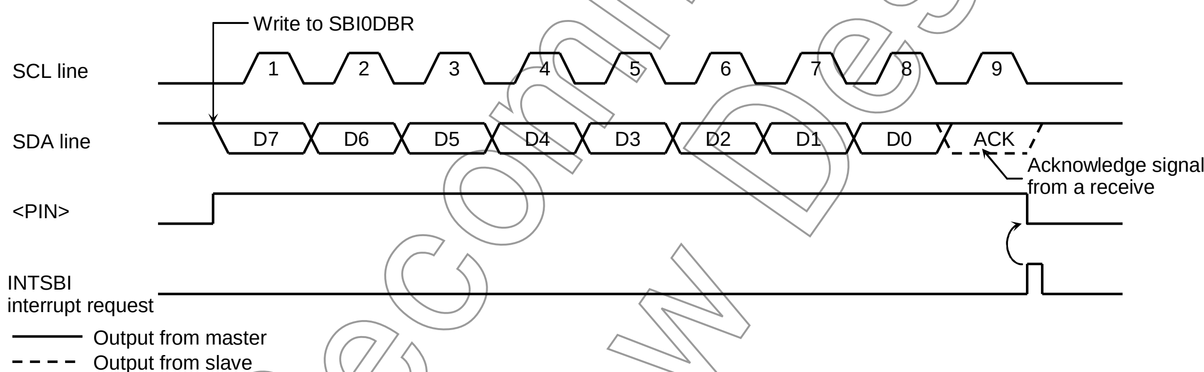


Figure 3.10.14 Example in which BC<2:0> = 000 and <ACK> = 1 in Transmitter Mode

Not Recommended for New Design

Downloaded from Arrow.com.

When the <TRX> is 0 (Receiver mode)

When the next transmitted data is other than 8 bits, set <BC2:0> <ACK> and read the received data from SBI0DBR to release the SCL line (data which is read immediately after a slave address is sent is undefined). After the data is read, <PIN> becomes 1. Serial clock pulse for transferring new 1 word of data is defined SCL and outputs "L" level from SDA pin with acknowledge timing.

An INTSBI interrupt request then occurs and the <PIN> becomes 0, Then the TMP91C824FG pulls down the SCL pin to the low level. The TMP91C824 outputs a clock pulse for 1 word of data transfer and the acknowledge signal each time that received data is read from the SBI0DBR.

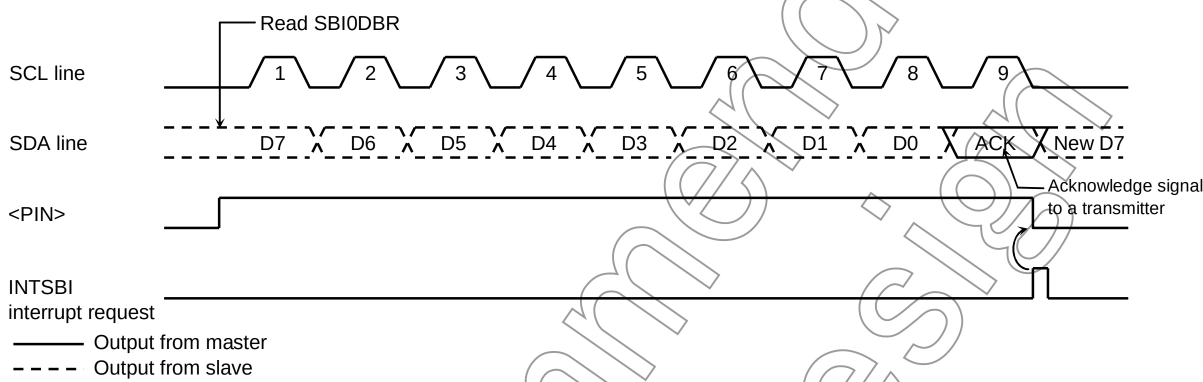


Figure 3.10.15 Example of when <BC2:0> = 000, <ACK> = 1 in Receiver Mode

In order to terminate the transmission of data to a transmitter, clear <ACK> to 0 before reading data which is 1 word before the last data to be received. The last data word does not generate a clock pulse as the acknowledge signal. After the data has been transmitted and an interrupt request has been generated, set BC<2:0> to 001 and read the data. The TMP91C824 generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA line on the bus remains high. The transmitter interprets the high signal as an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After the one data bit has been received and an interrupt request been generated, the TMP91C824 generates a stop condition (See Section 3.10.6 (4)) and terminates data transfer.

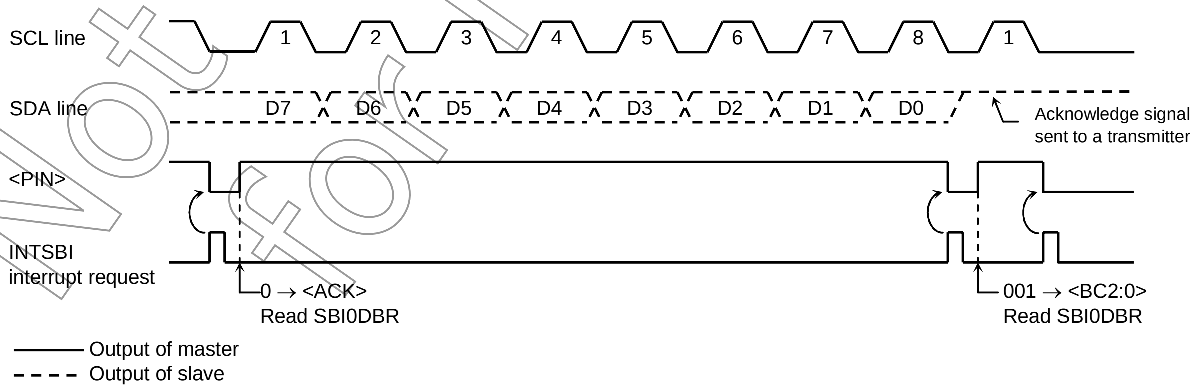


Figure 3.10.16 Termination of Data Transfer in Master Receiver Mode

Downloaded from Arrow.com.

b. If <MST> = 0 (Slave mode)

In the slave mode the TMP91C824 operates either in normal slave mode or in slave mode after losing arbitration.

In the slave mode, an INTSBI interrupt request occurs when the TMP91C824 receives a slave address or a GENERAL CALL from the master device, or when a GENERAL CALL is received and data transfer is complete, or after matching received address. In the master mode, the TMP91C824 operates in a slave mode if it losing arbitration. An INTSBI interrupt request occurs when a word data transfer terminates after losing arbitration. When an INTSBI interrupt request occurs the <PIN> is cleared to 0 and the SCL pin is pulled down to the low level. Either reading/writing from/to the SBI0DBR or setting the <PIN> to 1 will release the SCL pin after taking t_{LOW} time.

Check the SBI0SR<AL>, <TRX>, <AAS>, and <AD0> and implements processes according to conditions listed in the next table.

Table 3.10.1 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	The TMP91C824 loses arbitration when transmitting a slave address and receives a slave address for which the value of the direction bit sent from another master is 1.	Set the number of bits a word in <BC2:0> and write the transmitted data to SBI0DBR
	0	1	0	In salve receiver mode the TMP91C824 receives a slave address for which the value of the direction bit sent from the master is 1.	
		0	0	In salve transmitter mode a single word of is transmitted. Set BC<2:0> to the number of bits in a word.	Check the <LRB> setting. If <LRB> is set to 1, set <PIN> to 1 since the receiver win no request the data which follows. Then, cleat <TRX> to 0 to release the bus. If <LRB> is cleared to 0 of and write the transmitted data to SBI0DBR since the receiver requests next data.
0	1	1	1/0	The TMP91C824 loses arbitration when transmitting a slave address and receives a slave address or GENERAL CALL for which the value of the direction bit sent from another master is 0.	Read the SBI0DBR for setting the <PIN> to 1 (Reading dummy data) or set the <PIN> to 1.
		0	0	The TMP91C824 loses arbitration when transmitting a slave address or data and terminates word data transfer.	
	0	1	1/0	In slave receiver mode the TMP91C824 receives a slave address or GENERAL CALL for which the value of the direction bit sent from the master is 0.	
		0	1/0	In slave receiver mode the TMP91C824 terminates receiving word data.	Set BC<2:0> to the number of bits in a word and read the received data from SBI0DBR.

Not Recommended for New Design

Downloaded from Arrow.com.

(4) Stop condition generation

When SBI0SR<BB> = 1, the sequence for generating a stop condition can be initiated by writing 1 to SBI0CR2<MST, TRX, PIN> and 0 to SBI0CR2<BB>. Do not modify the contents of SBI0CR2<MST, TRX, PIN, BB> until a stop condition has been generated on the bus. When the bus's SCL line has been pulled low by another device, the TMP91C824 generates a stop condition when the other device has released the SCL line.

When SBI0CR2<MST, TRX, PIN> are written 1 and <BB> is written 0, <BB> changes to 0 by internal SCL changes to 1, without waiting stop condition.

To check whether SCL and SDA pin are 1 by sensing their ports is needed to detect bus free condition.

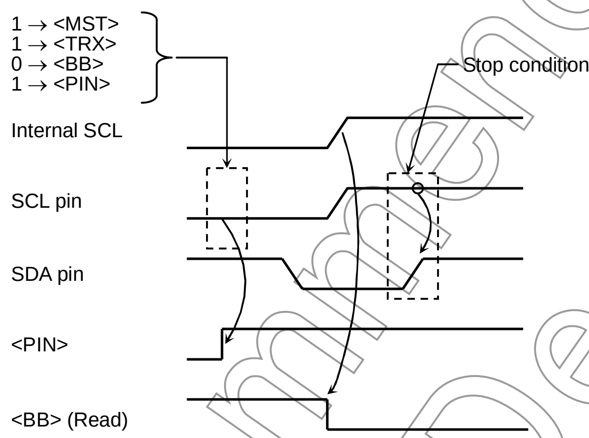


Figure 3.10.17 Stop Condition Generation (Single master)

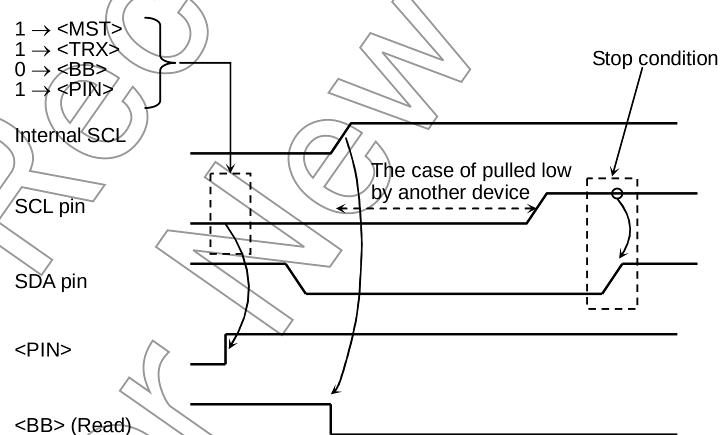


Figure 3.10.18 Stop Condition Generation (Multi master)

Not Recommended for New Design

Downloaded from Arrow.com.

(5) Restart

Restart is used during data transfer between a master device and a slave device to change the data transfer direction. The following description explains how to restart when the TMP91C824 is in master mode.

Clear SBI0CR2<MST, TRX, BB> to 0 and set SBI0CR2<PIN> to 1 to release the bus. The SDA line remains high and the SCL pin is released. Since a stop condition has not been generated on the bus, other devices assume the bus to be in busy state. Monitor the value of SBI0SR<BB> until it becomes 0 so as to ascertain when the TMP91C824's SCL pin is released. Check the <LRB> until it becomes 1 to check that the SCL line on a bus is not pulled down to the low level by other devices. After confirming that the bus remains in a free state, generate a start condition using the procedure described in 3.10.6 (2).

In order to satisfy the setup time requirements when restarting, take at least 4.7 μs of waiting time by software from the time of restarting to confirm that the bus is free until the time to generate the start condition.

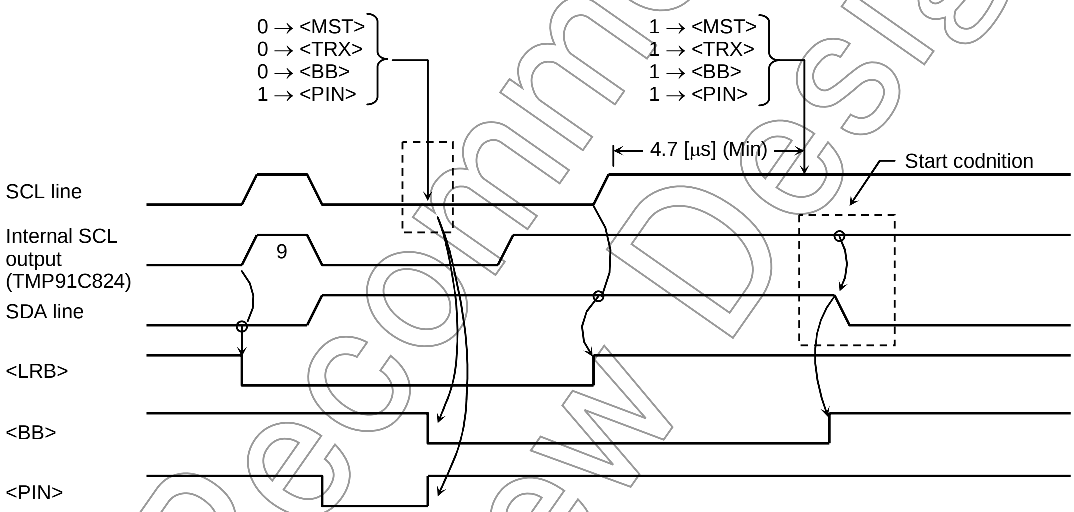


Figure 3.10.19 Timing Diagram for TMP91C824FG Restart

Not Recommended for New Design

Downloaded from Arrow.com.

3.10.7 Clocked Synchronous 8-Bit SIO Mode Control

The following registers are used to control and monitor the operation status when the serial bus interface (SBI) is being operated in clocked synchronous 8-bit SIO mode.

Serial Bus Interface Control Register 1

SBI0CR1 (0240H)

	7	6	5	4	3	2	1	0
Bit symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
Read/Write	W					W		W
After reset	0	0	0	0		0	0	0
Function	Transfer start 0: Stop 1: Start	Continue/ abort transfer 0: Continue transfer 1: Abort transfer	Transfer mode select 00: Transmit mode 01: (Reserved) 10: Transmit/receive mode 11: Receive mode			Serial clock selection and reset monitor		

Prohibit read-modify-write

Serial clock selection <SCK2:0> at write

000	n = 4	2.1 MHz	System clock: fc Clock gear: fc/1 fc = 33 MHz (Output to SCK pin) $fscl = \frac{fc}{2^n}$ [Hz]
001	n = 5	1031.3 kHz	
010	n = 6	515.6 kHz	
011	n = 7	257.8 kHz	
100	n = 8	128.9 kHz	
101	n = 9	64.5 kHz	
110	n = 10	32.2 kHz	
111	–	External mode	(Input from SCK terminal)

Transfer mode selection

00	8-bit transmit mode
01	(Reserved)
10	8-bit transmit/received mode
11	8-bit received mode

Continue/abort transfer

0	Continue transfer
1	Abort transfer (Automatically cleared after transfer aborted)

Transfer start/stop

0	Stopped
1	Started

Note: Set the tranfer mode and the serial clock after setting <SIOS> to 0 and <SIOINH> to 1.

Serial Bus Interface Data Buffer Register

SBI0DBR (0241H)

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receiver)/W (Transfer)							
After reset	Undefined							

Prohibit read-modify-write

Figure 3.10.20 Register for the SIO Mode

Not Recommended for New Design

Downloaded from Arrow.com.

Serial Bus Interface Control Register 2

SBI0CR2 (0243H)		7	6	5	4	3	2	1	0
	Bit symbol					SBIM1	SBIM0	–	–
	Read/Write					W		W	W
	After reset					0	0	0	0
Prohibit read-modify-write	Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I^2C bus mode 11: (Reserved)		(Note 2)	(Note 2)

Serial bus interface operation mode selection

00	Port mode (Serial bus interface output disabled)
01	Clocked synchronous 8-bit SIO mode
10	I^2C bus mode
11	(Reserved)

Note 1: Set the SBI0CR1<BC2:0> 000 before switching to a clocked synchronous 8-bit SIO mode.

Note 2: Please always write SBICR2<1:0> to “00”.

Serial Bus Interface Status Register

SBI0SR (0243H)		7	6	5	4	3	2	1	0
	Bit symbol					SIOF	SEF		
	Read/Write					R			
	After reset					0	0		
	Function					Serial transfer operation status monitor	Shift operation status monitor		

Shift operation status monitor

0	Shift operation terminated
1	Shift operation in progress

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in progress

Figure 3.10.21 Registers for the SIO Mode

Not Recommended for New Design

Downloaded from Arrow.com.

Serial Bus Interface Baud Rate Register 0

SBI0BR0 (0244H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	–	I2SBI0						
Read/Write	W	R/W						
After reset	0	0						
Function	Always write 0	IDLE2 0: Stop 1: Operate						

Operation in IDLE2 mode

0	Stop
1	Operate

Serial Bus Interface Baud Rate Register 1

SBI0BR1 (0245H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Operate	Always write 0						

Baud rate clock control

0	Stop
1	Operate

Figure 3.10.22 Registers for the SIO Mode

Not Recommended for New Design

Downloaded from Arrow.com.

(1) Serial clock

a. Clock source

SBI0CR1<SCK2:0> is used to select the following functions:

<u>Internal clock</u>

In internal clock mode one of seven frequencies can be selected. The serial clock signal is output to the outside on the SCK pin. The SCK pin goes high when data transfer starts. When the device is writing (in transmit mode) or reading (in receive mode), data cannot follow the serial clock rate, so an automatic wait function is executed which automatically stops the serial clock and holds the next shift operation until reading or writing has been completed.

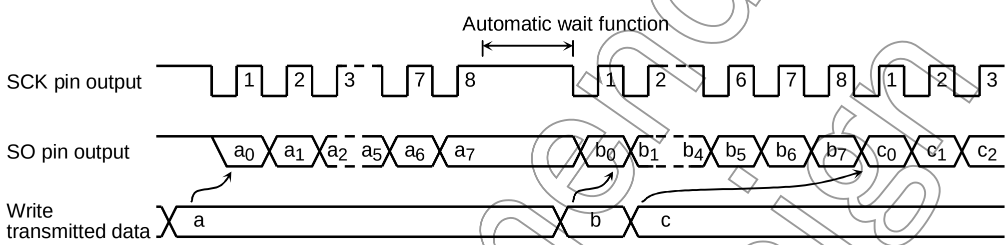


Figure 3.10.23 Automatic Wait Function

<u>External clock (<SCK2:0> = 111)</u>

An external clock input via the SCK pin is used as the serial clock. In order to ensure the integrity of shift operations, both the high and low-level serial clock pulse widths shown below must be maintained. The maximum data transfer frequency is 2.1 MHz (when fc = 33 MHz).

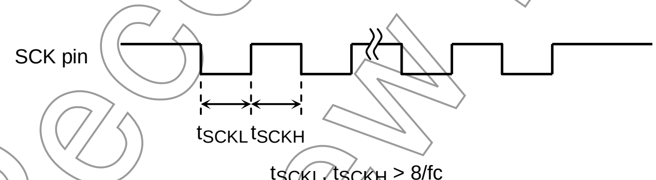


Figure 3.10.24 Maximum Data Transfer Frequency when External Clock Input Used

Not Recommended for New Design

Downloaded from Arrow.com.

b. Shift edge

Data is transmitted on the leading edge of the clock and received on the trailing edge.

Leading edge shift

Data is shifted on the leading edge of the serial clock (on the falling edge of the SCK pin input/output).

Trailing edge shift

Data is shifted on the trailing edge of the serial clock (on the rising edge of the SCK pin input/output).

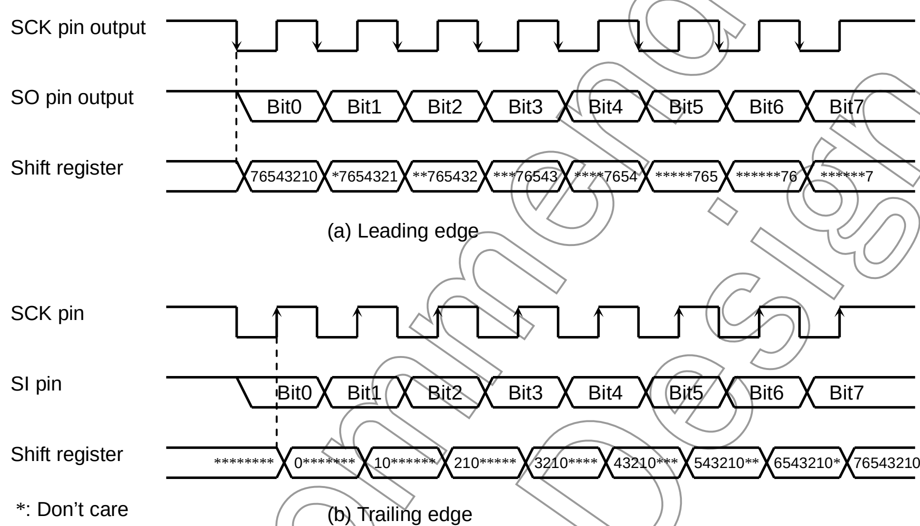


Figure 3.10.25 Shift Edge

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Transfer modes

The SBI0CR1<SIOM1:0> is used to select a transmit, receive or transmit/receive mode.

a. 8-bit transmit mode

Set a control register to a transmit mode and write transmit data to the SBI0DBR.

After the transmit data is written, set the SBI0CR1<SIOS> to 1 to start data transfer. The transmitted data is transferred from SBI0DBR to the shift register and output to the SO pin in synchronized with the serial clock, starting from the least significant bit (LSB), When the transmission data is transferred to the shift register, the SBI0DBR becomes empty. An INTSBI (Buffer empty) interrupt request is generated to request new data.

When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated if new data is not loaded to the data buffer register after the specified 8-bit data is transmitted. When new transmit data is written, automatic-wait function is canceled.

When the external clock is used, data should be written to SBI0DBR before new data is shifted. The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written to SBI0DBR by the interrupt service program.

When the transmit is started, after the SBI0SR<SIOF> goes 1 output from the SO pin holds final bit of the last data until falling edge of the SCK.

Transmitting data is ended by clearing the <SIOS> to 0 by the buffer empty interrupt service program or setting the <SIOINH> to 1. When the <SIOS> is cleared, the transmitted mode ends when all data is output. In order to confirm if data is surely transmitted by the program, set the <SIOF> (Bit3 of SBI0SR) to be sensed. The SBI0SR<SIOF> is cleared to 0 when transmitting is complete. When the <SIOINH> is set to 1, transmitting data stops. SBI0SR<SIOF> turns 0.

When an external clock is used, it is also necessary to clear SBI0SR<SIOS> to 0 before new data is shifted; otherwise, dummy data is transmitted and operation ends.

Not Recommended for New Design

Downloaded from Arrow.com.

Example: Program to stop data transmission (when an external clock is used)

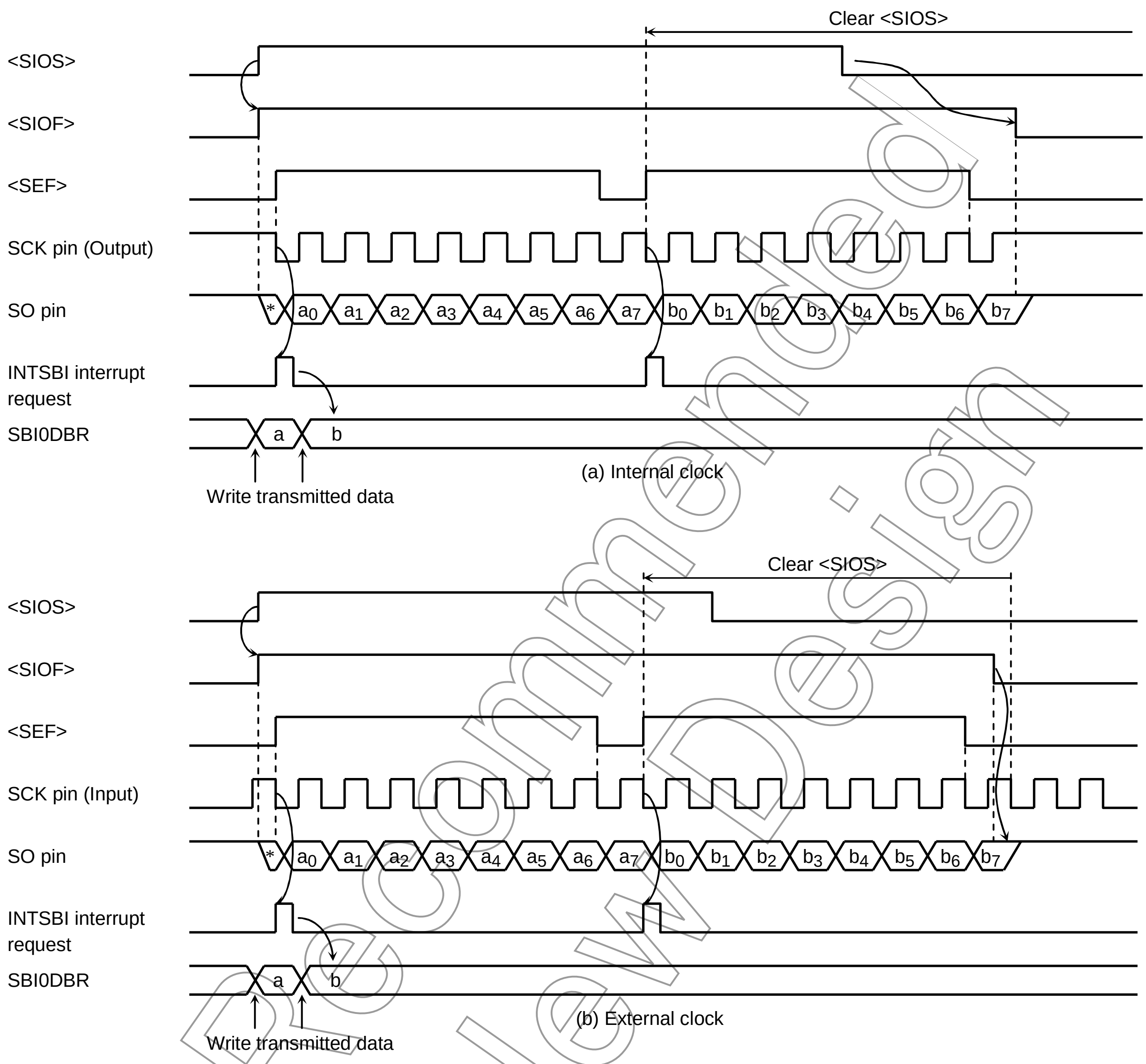


Figure 3.10.26 Transfer Mode

```
STEST1:   BIT  2, (SBI0SR)              ; If <SEF> = 1 then loop
          JR   NZ, STEST1
STEST2:   BIT  0, (P7)                  ; If SCK = 0 then loop
          JR   Z, STEST2
          LD   (SBI0CR1), 00000111B     ; <SIOS> ← 0
```

Not Recommended for New Design

Downloaded from Arrow.com.

b. 8-bit receive mode

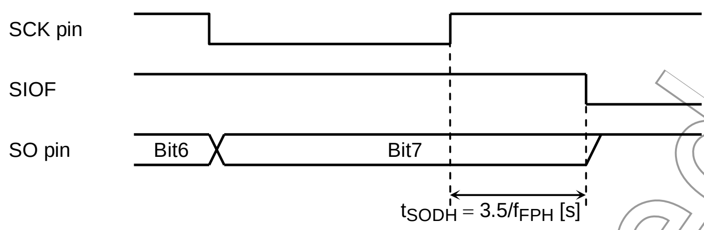


Figure 3.10.27 Transmitted Data Hold Time at End of Transmission

Set the control register to receive mode and set SBI0CR1<SIOS> to 1 for switching to receive mode. Data is received into the shift register via the SI pin and synchronized with the serial clock, starting from the least significant bit (LSB). When 8-bit data is received, the data is transferred from the shift register to SBI0DBR. An INTSBI (Buffer full) interrupt request is generated to request that the received data be read. The data is then read from SBI0DBR by the interrupt service program.

When an internal clock is used, the serial clock will stop and the automatic wait function will be in effect until the received data has been read from SBI0DBR.

When an external clock is used, since shift operation is synchronized with an external clock pulse, the received data should be read from SBI0DBR before the next serial clock pulse is input. If the received data is not read, any further data which is to be received is canceled. The maximum transfer speed when an external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when the received data is read.

Receiving of data ends when <SIOS> is cleared to 0 by the buffer full interrupt service program or when <SIOINH> is set to 1. If <SIOS> is cleared to 0, received data is transferred to SBI0DBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm whether data is being received properly by the program, set SBI0SR<SIOF> to be sensed. <SIOF> is cleared to 0 when receiving has been completed. When it is confirmed that receiving has been completed, the last data is read. When <SIOINH> is set to 1, data receiving stops. <SIOF> is cleared to 0 (The received data becomes invalid, therefore no need to read it).

Note: When the transfer mode is changed, the contents of SBI0DBR will be lost. If the mode must be changed, conclude data receiving by clearing <SIOS> to 0, read the last data, then change the mode.

Not Recommended for New Design

Downloaded from Arrow.com.

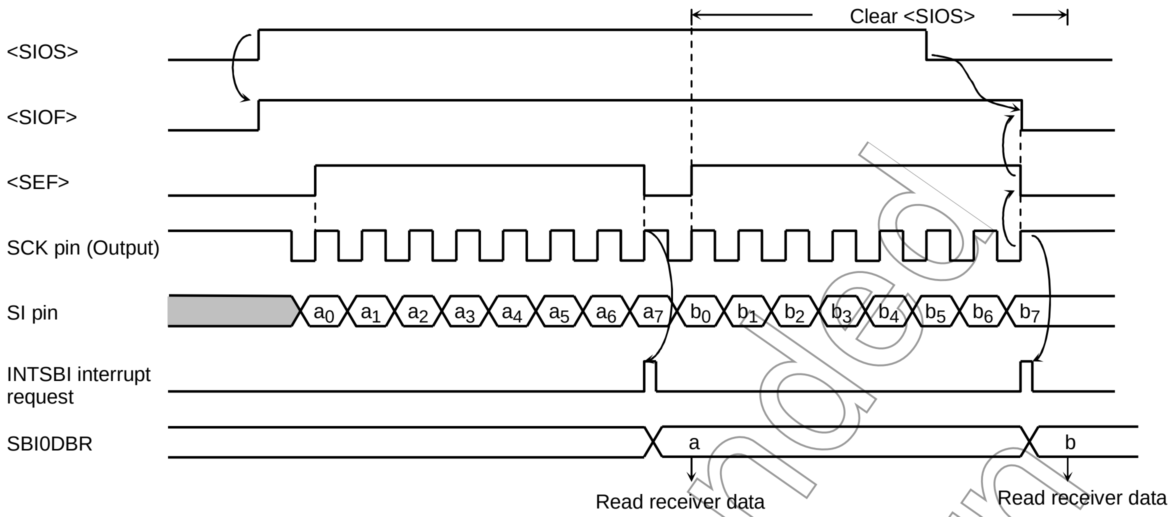


Figure 3.10.28 Receiver Mode (Example: Internal clock)

c. 8-bit transmit/receive mode

Set a control register to a transmit/receive mode and write data to SBI0DBR. After the data has been written, set SBI0CR<SIOS> to 1 to start transmitting/receiving. When data is transmitted, the data is output via the SO pin, starting from the least significant bit (LSB) and synchronized with the leading edge of the serial clock signal. When data is received, the data is input via the SI pin on the trailing edge of the serial clock signal. 8-bit data is transferred from the shift register to SBI0DBR and an INTSBI interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the data which is to be transmitted. SBI0DBR is used for both transmitting and receiving. Transmitted data should always be written after received data has been read.

When an internal clock is used, the automatic wait function will be in effect until the received data has been read and the next data has been written.

When an external clock is used, since the shift operation is synchronized with the external clock, received data is read and transmitted data is written before a new shift operation is executed. The maximum transfer speed when an external clock is used is determined by the delay time between the time when an interrupt request is generated and the time at which received data is read and transmitted data is written.

When the transmit is started, after the SBI0SR<SIOF> goes 1 output from the SO pin holds final bit of the last data until falling edge of the SCK.

Transmitting/receiving data ends when <SIOS> is cleared to 0 by the INTS2 interrupt service program or when SBI0CR1<SIOINH> is set to 1. When <SIOS> is cleared to 0, received data is transferred to SBI0DBR in complete blocks. The transmit/receive mode ends when the transfer is complete. In order to confirm whether data is being transmitted/received properly by the program, set SBI0SR to be sensed. <SIOF> is set to 0 when transmitting/receiving has been completed. When <SIOINH> is set to 1, data transmitting/receiving stops. <SIOF> is then cleared to 0.

Note: When the transfer mode is changed, the contents of SBI0DBR will be lost. If the mode must be changed, conclude data transmitting/receiving by clearing <SIOS> to 0, read the last data, then change the transfer mode.

Not Recommended for New Design

Downloaded from Arrow.com.

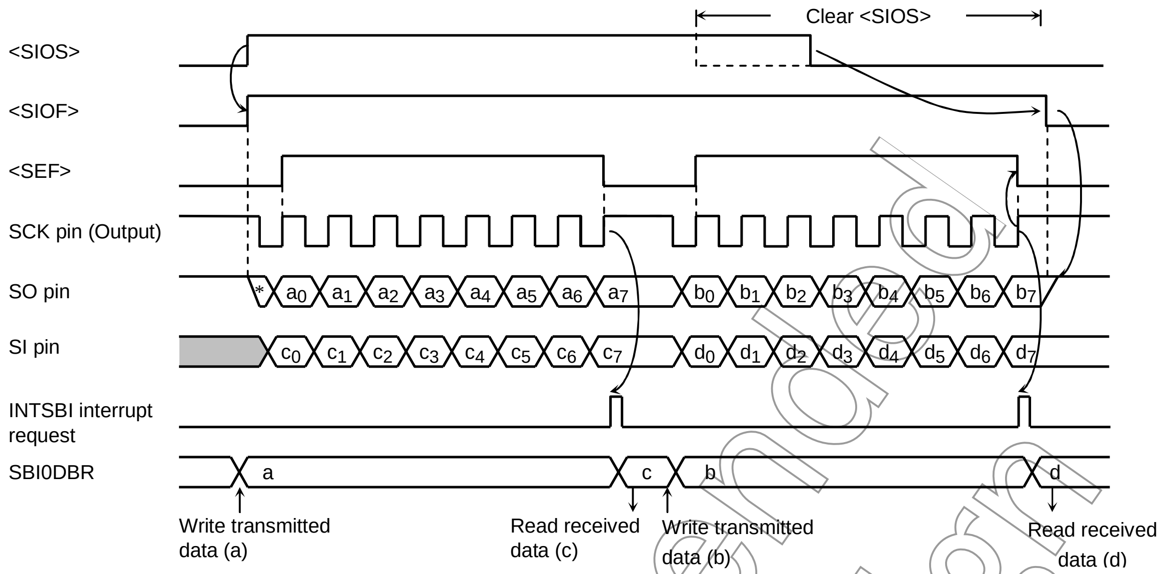


Figure 3.10.29 Transmit/Received Mode (Example using internal clock)

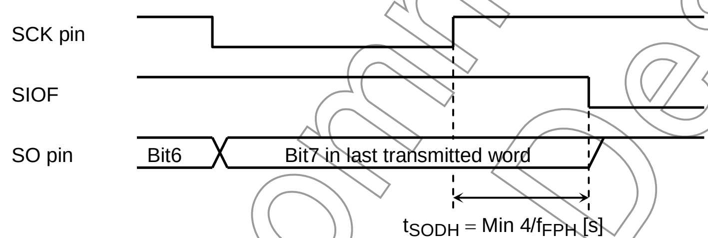


Figure 3.10.30 Transmitted Data Hold Time at End of Transmit/Receive

Not Recommended for New Design

Downloaded from Arrow.com.

3.11 Analog/Digital Converter

The TMP91C824 incorporates a 10-bit successive approximation-type analog/digital converter (AD converter) with 8-channel analog input.

Figure 3.11.1 is a block diagram of the AD converter. The 8-channel analog input pins (AN0 to AN7) are shared with the input only port 8 and can thus be used as an input port.

Note: When IDLE2, IDLE1 or STOP mode is selected, so as to reduce the power, with some timings the system may enter a standby mode even though the internal comparator is still enabled. Therefore be sure to check that AD converter operations are halted before a HALT instruction is executed.

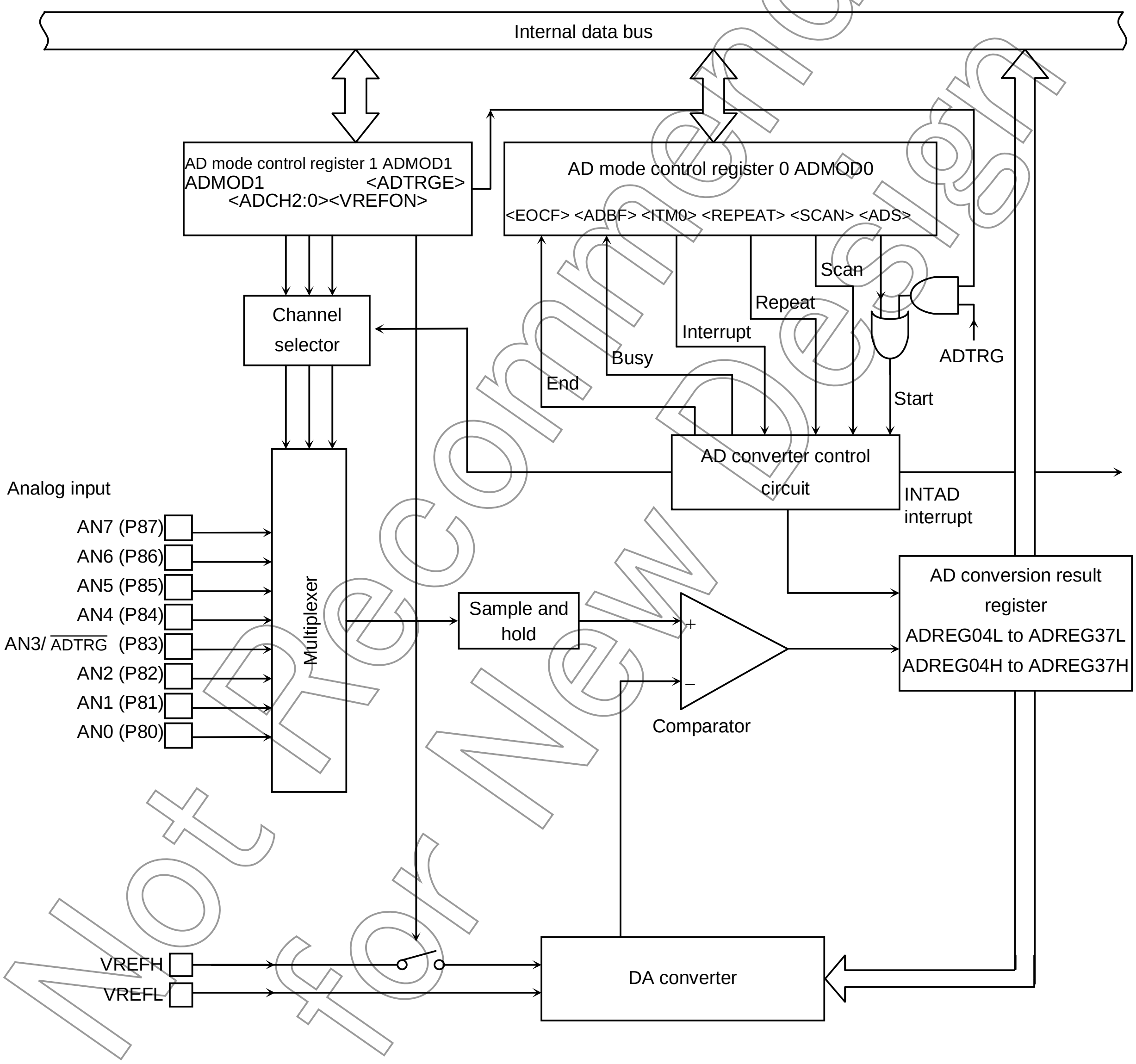


Figure 3.11.1 Block Diagram of AD Converter

Not Recommended for New Design

Downloaded from Arrow.com.

3.11.1 Analog/Digital Converter Registers

The AD converter is controlled by the two AD mode control registers: ADMOD0 and ADMOD1. The AD conversion results are stored in 8 kinds of AD conversion data upper and lower registers: ADREG04H/L, ADREG15H/L, ADREG26H/L and ADREG37H/L.

Figure 3.11.2 shows the registers related to the AD converter.

AD Mode Control Register 0

ADMOD0 (02B0H)

	7	6	5	4	3	2	1	0
Bit symbol	EOCF	ADBF	–	–	ITM0	REPEAT	SCAN	ADS
Read/Write	R		R/W					
After reset	0	0	0	0	0	0	0	0
Function	AD conversion end flag 0: Conversion in progress 1: Conversion complete	AD conversion busy flag 0: Conversion stopped 1: Conversion in progress	Always write 0	Always write 0	Interrupt specification in conversion channel fixed repeat mode 0: Every conversion 1: Every fourth conversion	Repeat mode specification 0: Single conversion 1: Repeat conversion mode	Scan mode specification 0: Conversion channel fixed mode 1: Conversion channel scan mode	AD conversion start 0: Don't care 1: Start conversion Always 0 when read

AD conversion start

0	Don't care
1	Start AD conversion

Note: Always read as 0.

AD scan mode setting

0	AD conversion channel fixed mode
1	AD conversion channel scan mode

AD repeat mode setting

0	AD single conversion mode
1	AD repeat conversion mode

Specify AD conversion interrupt for channel fixed repeat conversion mode

	Channel fixed repeat conversion mode <SCAN> = 0, <REPEAT> = 1
0	Generates interrupt every conversion.
1	Generates interrupt every fourth conversion.

AD conversion busy flag

0	AD conversion stopped
1	AD conversion in progress

AD conversion end flag

0	Before or during AD conversion
1	AD conversion complete

Figure 3.11.2 AD Converter Related Register

Not Recommended for New Design

Downloaded from Arrow.com.

AD Mode Control Register 1

ADMOD1 (02B1H)		7	6	5	4	3	2	1	0
	Bit symbol	VREFON	I2AD			ADTRGE	ADCH2	ADCH1	ADCH0
	Read/Write	R/W				R/W			
	After reset	0	0			0	0	0	0
	Function	VREF application control 0: OFF 1: ON	IDLE2 0: Stop 1: Operate			AD external trigger start control 0: Disable 1: Enable	Analog input channel selection		

Analog input channel selection

<SCAN> / <ADCH2:0>	0 (Channel fixed)	1 (Channel scanned)
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011 (Note)	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	AN6	AN4 → AN5 → AN6
111	AN7	AN4 → AN5 → AN6 → AN7

AD conversion start control by external trigger ($\overline{\text{ADTRG}}$ input)

0	Disabled
1	Enabled

IDLE2 control

0	Stopped
1	In operation

Control of application of reference voltage to AD converter

0	OFF
1	ON

Before starting conversion (before writing 1 to ADMOD0<ADS>), set the <VREFON> bit to 1.

Note: As pin AN3 also functions as the $\overline{\text{ADTRG}}$ input pin, do not set <ADCH2:0> = 011 when using $\overline{\text{ADTRG}}$ with <ADTRGE> = 0.

Figure 3.11.3 AD Converter Related Registers

Not Recommended for New Design

Downloaded from Arrow.com.

AD Conversion Data Low Register 0/4

ADREG04L (02A0H)		7	6	5	4	3	2	1	0
	Bit symbol	ADR01	ADR00						ADR0RF
	Read/Write	R							R
	After reset	Undefined							0
	Function	Stores lower 2 bits of AD conversion result							AD conversion data storage flag 1: Conversion result stored

AD Conversion Data Upper Register 0/4

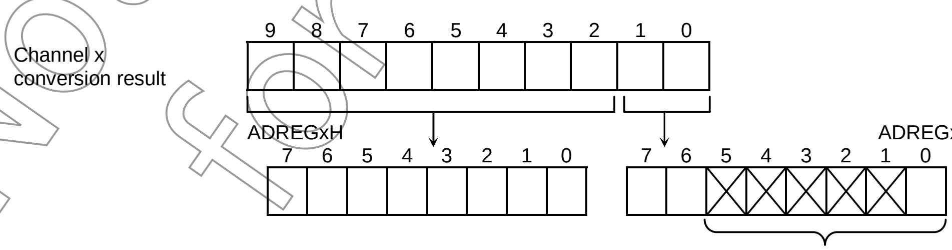
ADREG04H (02A1H)		7	6	5	4	3	2	1	0
	Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper 8 bits AD conversion result.							

AD Conversion Data Lower Register 1/5

ADREG15L (02A2H)		7	6	5	4	3	2	1	0
	Bit symbol	ADR11	ADR10						ADR1RF
	Read/Write	R							R
	After reset	Undefined							0
	Function	Stores lower 2 bits of AD conversion result							AD conversion result flag 1: Conversion result stored

AD Conversion Data Upper Register 1/5

ADREG15H (02A3H)		7	6	5	4	3	2	1	0
	Bit symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper 8 bits of AD conversion result.							



- Bits 5 to 1 are always read as 1.
- Bit0 is the AD conversion data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.11.4 AD Converter Related Registers

Not Recommended for New Design

Downloaded from Arrow.com.

AD Conversion Result Lower Register 2/6

ADREG26L (02A4H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR21	ADR20						ADR2RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							AD conversion data storage flag 1: Conversion result stored

AD Conversion Data upper Register 2/6

ADREG26H (02A5H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After reset	Undefined							
Function	Stores upper 8 bits of AD conversion result.							

AD Conversion Data Lower Register 3/7

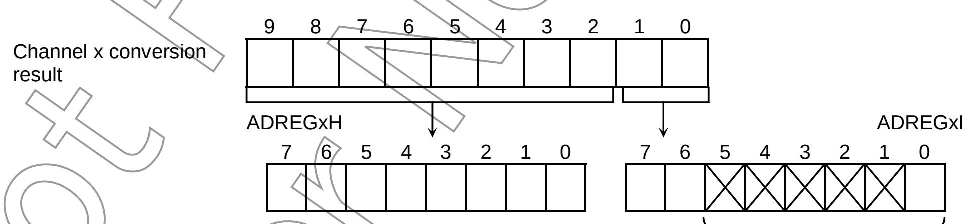
ADREG37L (02A6H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR31	ADR30						ADR3RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							AD Conversion Data Storage flag 1: conversion result stored

AD Conversion Result Upper Register 3/7

ADREG37H (02A7H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After reset	Undefined							
Function	Stores upper 8 bits of AD conversion result.							



- Bits 5 to1 are always read as 1.
- Bit0 is the AD conversion data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.11.5 AD Converter Related Registers

Not Recommended for New Design

Downloaded from Arrow.com.

3.11.2 Description of Operation

(1) Analog reference voltage

A high-level analog reference voltage is applied to the VREFH pin; a low-level analog reference voltage is applied to the VREFL pin. To perform AD conversion, the reference voltage as the difference between VREFH and VREFL is divided by 1024 using string resistance. The result of the division is then compared with the analog input voltage.

To turn off the switch between VREFH and VREFL, write 0 to ADMOD1 <VREFON> in AD mode control register 1. To start AD conversion in the off state, first write 1 to ADMOD1<VREFON>, wait 3 μs until the internal reference voltage stabilizes (This is not related to fc), then set ADMOD0<ADS> to 1.

(2) Analog input channel selection

The analog input channel selection varies depends on the operation mode of the AD converter.

- In analog input channel fixed mode (ADMOD0<SCAN> = 0)

Setting ADMOD1<ADCH2:0> selects one of the input pins AN0 to AN7 as the input channel.

- In analog input channel scan mode (ADMOD0<SCAN> = 1)

Setting ADMOD1<ADCH2:0> selects one of the 8 scan modes.

Table 3.11.1 illustrates analog input channel selection in each operation mode.

After reset, ADMOD0<SCAN> = 0 and ADMOD1<ADCH2:0> = 000. Thus pin AN0 is selected as the fixed input channel. Pins not used as analog input channels can be used as standard input port pins.

Table 3.11.1 Analog Input Channel Selection

<ADCH2:0>	Channel Fixed <SCAN> = 0	Channel Scan <SCAN> = 1
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	AN6	AN4 → AN5 → AN6
111	AN7	AN4 → AN5 → AN6 → AN7

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Starting AD conversion

To start AD conversion, write 1 to ADMOD0<ADS> in AD mode control register 0 or ADMOD1<ADTRGE> in AD mode control register 1 and input falling edge on $\overline{\text{ADTRG}}$ pin. When AD conversion starts, the AD conversion busy flag ADMOD0<ADBF> will be set to 1, indicating that AD conversion is in progress.

Writing 1 to ADMOD0<ADS> during AD conversion restarts conversion. At that time, to determine whether the AD conversion results have been preserved, check the value of the conversion data storage flag ADREGxL<ADRxRF>.

During AD conversion, a falling edge input on the $\overline{\text{ADTRG}}$ pin will be ignored.

(4) AD conversion modes and the AD conversion end interrupt

The 4 AD conversion modes are:

- Channel fixed single conversion mode
- Channel scan single conversion mode
- Channel fixed repeat conversion mode
- Channel scan repeat conversion mode

The ADMOD0<REPEAT> and ADMOD0<SCAN> settings in AD mode control register 0 determine the AD mode setting.

Completion of AD conversion triggers an INTAD AD conversion end interrupt request. Also, ADMOD0<EOCF> will be set to 1 to indicate that AD conversion has been completed.

a. Channel fixed single conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 00 selects channel fixed single conversion mode.

In this mode, data on one specified channel is converted once only. When the conversion has been completed, the ADMOD0<EOCF> flag is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

b. Channel scan single conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 01 selects channel scan single conversion mode.

In this mode, data on the specified scan channels is converted once only. When scan conversion has been completed, ADMOD0<EOCF> is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

Not Recommended for New Design

Downloaded from Arrow.com.

c. Channel fixed repeat conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 10 selects channel fixed repeat conversion mode.

In this mode, data on one specified channel is converted repeatedly. When conversion has been completed, ADMOD0<EOCF> is set to 1 and ADMOD0<ADBF> is not cleared to 0 but held 1. INTAD interrupt request generation timing is determined by the setting of ADMOD0<ITM0>.

Setting <ITM0> to 0 generates an interrupt request every time an AD conversion is completed.

Setting <ITM0> to 1 generates an interrupt request on completion of every fourth conversion.

d. Channel scan repeat conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 11 selects channel scan repeat conversion mode.

In this mode, data on the specified scan channels is converted repeatedly. When each scan conversion has been completed, ADMOD0<EOCF> is set to 1 and an INTAD interrupt request is generated. ADMOD0<ADBF> is not cleared to 0 but held 1.

To stop conversion in a repeat conversion mode (e.g., in cases c. and d.), write a 0 to ADMOD0<REPEAT>. After the current conversion has been completed, the repeat conversion mode terminates and ADMOD0<ADBF> is cleared to 0.

Switching to a halt state (IDLE2 mode with ADMOD1<I2AD> cleared to 0, IDLE1 mode or STOP mode) immediately stops operation of the AD converter even when AD conversion is still in progress. In repeat conversion modes (e.g., in cases c. and d.), when the halt is released, conversion restarts from the beginning. In single conversion modes (e.g., in cases a. and b.), conversion does not restart when the halt is released (The converter remains stopped).

Table 3.11.2 shows the relationship between the AD conversion modes and interrupt requests.

Table 3.11.2 Relationship between AD Conversion Modes and Interrupt Requests

Mode	Interrupt Request Generation	ADMOD0		
		<ITM0>	<REPEAT>	<SCAN>
Channel fixed single conversion mode	After completion of conversion	X	0	0
Channel scan single conversion mode	After completion of scan conversion	X	0	1
Channel fixed repeat conversion mode	Every conversion	0	1	0
	Every forth conversion	1		
Channel scan repeat conversion mode	After completion of every scan conversion	X	1	1

X: Don't care

Not Recommended for New Design

Downloaded from Arrow.com.

(5) AD conversion time

84 states (5.1 μs at f_{FPH} = 33 MHz) are required for the AD conversion for one channel.

(6) Storing and reading the results of AD conversion

The AD conversion data upper and lower registers (ADREG04H/L to ADREG37H/L) store the AD conversion results. (ADREG04H/L to ADREG37H/L are read-only registers.)

In channel fixed repeat conversion mode, the conversion results are stored successively in registers ADREG04H/L to ADREG37H/L. In other modes, the AN0 and AN4, AN1 and AN5, AN2 and AN6, and AN3 and AN7 conversion results are stored in ADREG04H/L, ADREG15H/L, ADREG26H/L and ADREG37H/L respectively.

Table 3.11.3 shows the correspondence between the analog input channels and the registers which are used to hold the results of AD conversion.

Table 3.11.3 Correspondence between Analog Input Channels and AD Conversion Result Registers

Analog Input Channel (Port 8)	AD Conversion Result Register	
	Conversion Modes Other than at Right	Channel Fixed Repeat Conversion Mode (<ITM0> = 1)
AN0	ADREG04H/L	ADREG04H/L
AN1	ADREG15H/L	↓ ADREG15H/L
AN2	ADREG26H/L	↓ ADREG26H/L
AN3	ADREG37H/L	↓ ADREG37H/L
AN4	ADREG04H/L	
AN5	ADREG15H/L	
AN6	ADREG26H/L	
AN7	ADREG37H/L	

<ADRxRF>, bit0 of the AD conversion data lower register, is used as the AD conversion data storage flag. The storage flag indicates whether the AD conversion result register has been read or not. When a conversion result is stored in the AD conversion result register, the flag is set to 1. When either of the AD conversion result registers (ADREGxH or ADREGxL) is read, the flag is cleared to 0.

Reading the AD conversion result also clears the AD conversion end flag ADMOD0<EOCF> to 0.

Not Recommended for New Design

Downloaded from Arrow.com.

Example:

a. Convert the analog input voltage on the AN3 pin and write the result, to memory address 0800H using the AD interrupt (INTAD) processing routine.

Main routine:

Register		7	6	5	4	3	2	1	0	Description
INTE0AD	←	–	1	0	0	–	–	–	–	Enable INTAD and set it to interrupt level 4.
ADMOD1	←	1	1	X	X	0	0	1	1	Set pin AN3 to be the analog input channel.
ADMOD0	←	–	–	0	0	X	0	0	1	Start conversion in channel fixed single conversion mode.

Interrupt routine processing example:

WA	← ADREG37	Read value of ADREG37L and ADREG37H into 16-bit general-purpose register WA.
WA	> > 6	Shift contents read into WA six times to right and zero-fill upper bits.
(0800H)	← WA	Write contents of WA to memory address 0800H.

b. This example repeatedly converts the analog input voltages on the three pins AN0, AN1 and AN2, using channel scan repeat conversion mode.

Register		7	6	5	4	3	2	1	0	Description
INTE0AD	←	–	0	0	0	–	–	–	–	Disable INTAD.
ADMOD1	←	1	–	X	X	0	0	1	0	Set pins AN0 to AN2 to be the analog input channels.
ADMOD0	←	–	–	0	0	X	1	1	1	Start conversion in channel scan repeat conversion mode.

X: Don't care; –: No change

Not Recommended for New Design

Downloaded from Arrow.com.

3.12 Watchdog Timer (Runaway detection timer)

The TMP91C824 features a watchdog timer for detecting runaway.

The watchdog timer (WDT) is used to return the CPU to normal state when it detects that the CPU has started to malfunction (Runaway) due to causes such as noise.

When the watchdog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU. Connecting the watchdog timer output to the reset pin internally forces a reset. (The level of external $\overline{\text{RESET}}$ pin is not changed)

3.12.1 Configuration

Figure 3.12.1 is a block diagram of he watchdog timer (WDT).

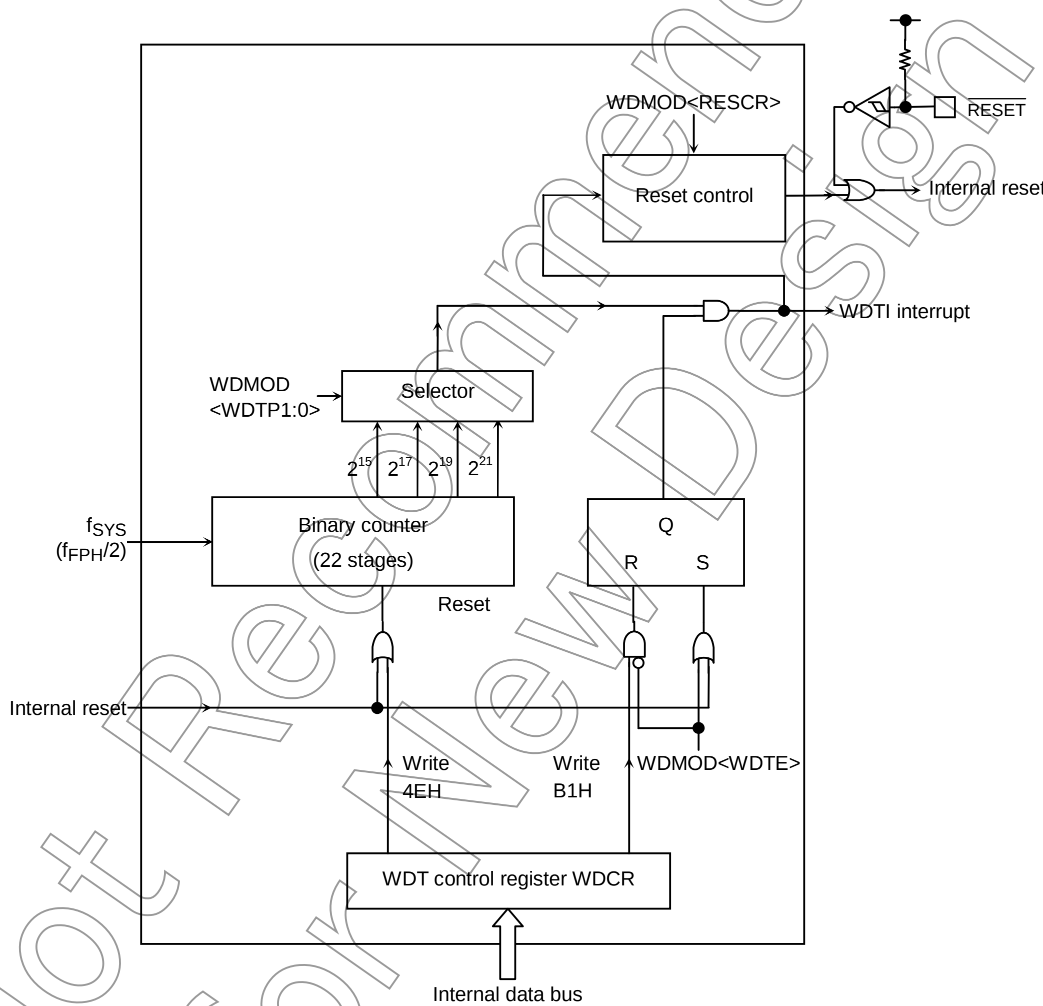


Figure 3.12.1 Block Diagram of Watchdog Timer

Note: It needs to care designing the total machine set, because watchdog timer can't operate completely by external noise.

Not Recommended for New Design

Downloaded from Arrow.com.

The watchdog timer consists of a 22-stage binary counter which uses the system clock (f_{SYS}) as the input clock. The binary counter can output $f_{SYS}/2^{15}$, $f_{SYS}/2^{17}$, $f_{SYS}/2^{19}$ and $f_{SYS}/2^{21}$.

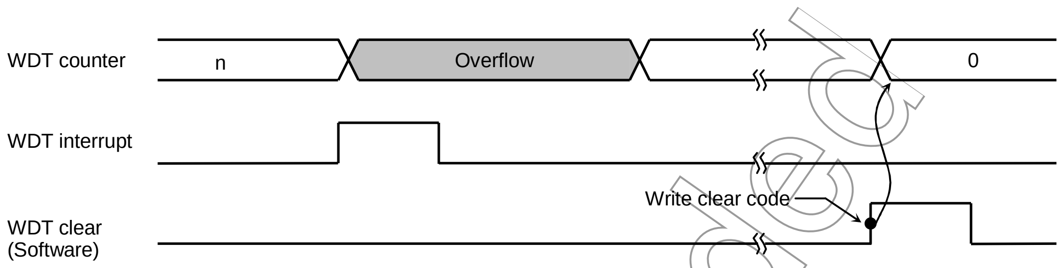


Figure 3.12.2 NORMAL Mode

The runaway is detected when an overflow occurs, and the watchdog timer can reset device. In this case, the reset time will be between 22 and 29 states (21.3~28.1 μs at f_{OSCH} = 33MHz, f_{FPH} = 2.2 MHz) is $f_{FPH}/2$, where f_{FPH} is generated by diving the high-speed oscillator clock (f_{OSCH}) by sixteen through the clock gear function.

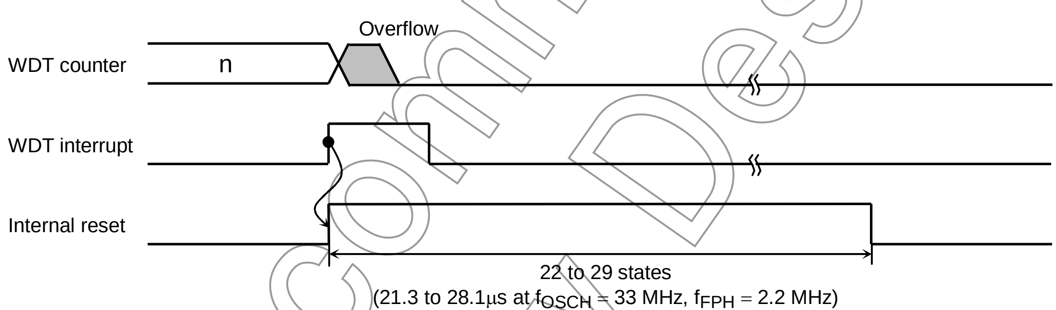


Figure 3.12.3 Reset Mode

Not Recommended for New Design

Downloaded from Arrow.com.

3.12.2 Control Registers

The watchdog timer WDT is controlled by two control registers WDMOD and WDCR.

(1) Watchdog timer mode register (WDMOD)

a. Setting the detection time for the watchdog timer in <WDTP1:0>

This 2-bit register is used for setting the watchdog timer interrupt time used when detecting runaway. After reset, this register is initialized to WDMOD<WDTP1:0> = 00.

The detection times for WDT are shown in Figure 3.12.4.

b. Watchdog timer enable/disable control register <WDTE>

After reset, WDMOD<WDTE> is initialized to 1, enabling the watchdog timer. To disable the watchdog timer, it is necessary to set this bit to 0 and to write the disable code (B1H) to the watchdog timer control register WDCR. This makes it difficult for the watchdog timer to be disabled by runaway.

However, it is possible to return the watchdog timer from the disabled state to the enabled state merely by setting <WDTE> to 1.

c. Watchdog timer out reset connection <RESCR>

This register is used to connect the output of the watchdog timer with the RESET terminal internally. Since WDMOD<RESCR> is initialized to 0 on reset, a reset by the watchdog timer will not be performed.

(2) Watchdog timer control register (WDCR)

This register is used to disable and clear the binary counter for the watchdog timer.

Disable control the watchdog timer can be disabled by clearing WDMOD<WDTE> to 0 and then writing the disable code (B1H) to the WDCR register.

```
WDMOD   ← 0 – – X X – – –     Clear WDMOD<WDTE> to 0.
WDCR    ← 1 0 1 1 0 0 0 1     Write the disable code (B1H).
```

- Enable control

Set WDMOD<WDTE> to 1.

- Watchdog timer clear control

To clear the binary counter and cause counting to resume, write the clear code (4EH) to the WDCR register.

```
WDCR    ← 0 1 0 0 1 1 1 0     Write the clear code (4EH).
```

Note1: If it is used disable control, set the disable code (B1H) to WDCR after write the clear code (4EH) once. (Please refer to setting example.)

Note2: If it is changed Watchdog timer setting, change setting after set to disable condition once.

Not Recommended for New Design

Downloaded from Arrow.com.

WDMOD
(0300H)

	7	6	5	4	3	2	1	0
Bit symbol	WDTE	WDTP1	WDTP0			I2WDT	RESCR	–
Read/Write	R/W	R/W				R/W		R/W
After reset	1	0	0			0	0	0
Function	WDT control 1: Enable	Select detecting time 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$				IDLE2 0: Stop 1: Operate	1: Internally connects WDT out to the reset pin	Always write 0

Watchdog timer out control

0	–
1	Connects WDT out to a reset

IDLE2 control

0	Stop
1	Operation

Watchdog timer detection time

at fc = 33 MHz, fs = 32.768 kHz

SYSCR1 System Clock Selection <SYSCK>	SYSCR1 Gear Value <GEAR2:0>	Watchdog Timer Detection Time WDMOD<WDTP1:0>			
		00	01	10	11
1 (fs)	XXX	2.0 s	8.0 s	32.0 s	128.0 s
0 (fc)	000 (fc)	1.99 ms	7.94 ms	31.78 ms	127.10 ms
	001 (fc/2)	3.97 ms	15.89 ms	63.55 ms	254.20 ms
	010 (fc/4)	7.94 ms	31.78 ms	127.10 ms	508.40 ms
	011 (fc/8)	15.89 ms	63.55 ms	254.20 ms	1016.80 ms
	100 (fc/16)	31.78 ms	127.10 ms	508.40 ms	2033.60 ms

Watchdog timer enable/disable control

0	Disabled
1	Enabled

Figure 3.12.4 Watchdog Timer Mode Register

Not Recommended for New Design

Downloaded from Arrow.com.

WDCR
(0301H)
Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	–							
Read/Write	W							
After reset	–							
Function	B1H: WDT disable code 4EH: WDT clear code							

Disable/clear WDT

B1H	Disable code
4EH	Clear code
Others	Don't care

Figure 3.12.5 Watchdog Timer Control Register

Not Recommended for New Design

Downloaded from Arrow.com.

3.12.3 Operation

The watchdog timer generates an INTWD interrupt when the detection time set in the WDMOD<WDTP1:0> has elapsed. The watchdog timer must be cleared 0 by software before an INTWD interrupt will be generated. If the CPU malfunctions (e.g., if runaway occurs) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter will overflow and an INTWD interrupt will be generated. The CPU will detect malfunction (Runaway) due to the INTWD interrupt and in this case it is possible to return to the CPU to normal operation by means of an anti-malfunction program.

The watchdog timer works immediately after reset.

The watchdog timer does not operate in IDLE1 or STOP mode, as the binary counter continues counting during bus release (when $\overline{\text{BUSAK}}$ goes low).

When the device is in IDLE2 mode, the operation of WDT depends on the WDMOD<I2WDT> setting. Ensure that WDMOD<I2WDT> is set before the device enters IDLE2 mode.

Example:

a. Clear the binary counter.

WDCR ← 0 1 0 0 1 1 1 0   Write the clear code (4EH).

b. Set the watchdog timer detection time to $2^{17}/f_{SYS}$.

WDMOD ← 1 0 1 X X – – –

c. Disable the watchdog timer.

WDMOD ← 0 – – X X – – –   Clear WDTE to 0.
WDCR ← 1 0 1 1 0 0 0 1   Write disable code (B1H).

Not Recommended for New Design

Downloaded from Arrow.com.

3.13 Real Time Clock (RTC)

3.13.1 Function Description for RTC

(1) Clock function (Second, minute, Hour, day of the week, day, Month and leap year)

(2) Calendar function

(3) 24- or 12-hour (AM/PM) clock function

(4) ± 30 second adjustment function (by software)

(5) Alarm output 1Hz/16Hz (from $\overline{\text{ALARM}}$ pin)

(6) Interrupt generate by Alarm output 1Hz/16Hz

3.13.2 Block Diagram

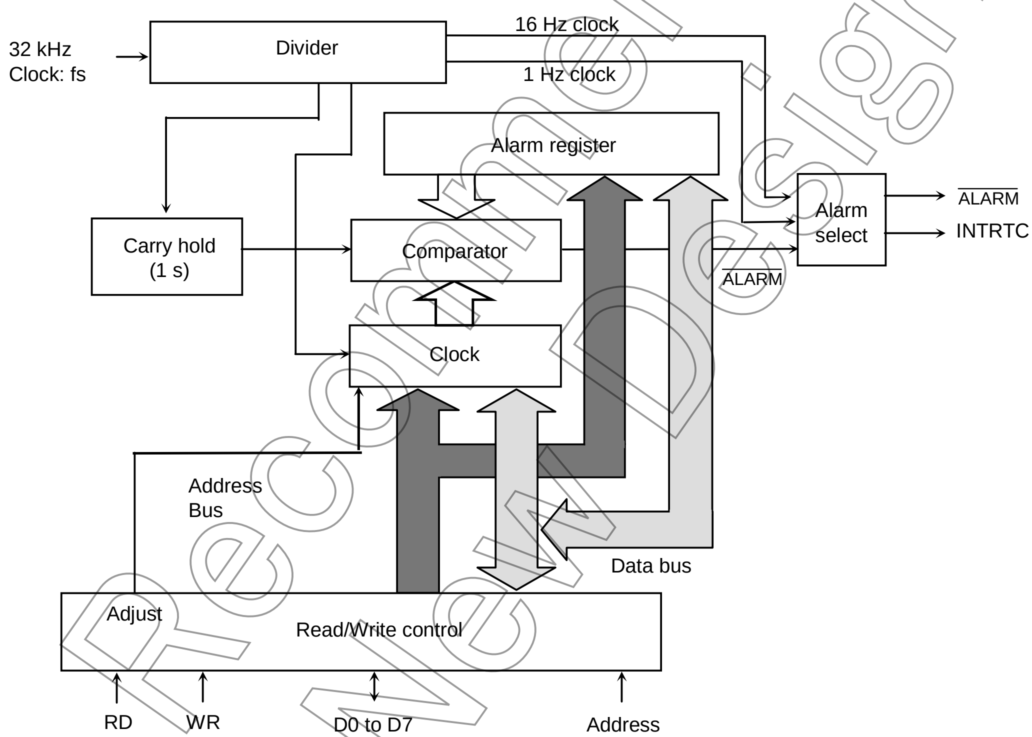


Figure 3.13.1 RTC Block Diagram

Note 1: The Christian era year column:
This product has year column toward only lower two columns. Therefore the next year in 99 works as 00 years. In system to use it, please manage upper two columns with the system side when handle year column in the Christian era.

Note 2: Leap year:
A leap year is the year, which is divisible with 4, but the year, which there is exception, and is divisible with 100 is not a leap year. However, the year, which is divisible with 400, is a leap year. But there is not this product for the correspondence to the above exception. Because there are only with the year which is divisible with 4 as a leap year, please cope with the system side if this function is problem.

Not Recommended for New Design

Downloaded from Arrow.com.

3.13.3 Control Registers

Table 3.13.1 PAGE 0 (Clock function) Registers

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
SECR	0320H		40 s	20 s	10 s	8 s	4 s	2 s	1 s	Second column	R/W
MINR	0321H		40 min.	20 min.	10 min.	8 min.	4 min.	2 min.	1 min.	Minute column	R/W
HOURR	0322H			20 /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hour	Hour column	R/W
DAYR	0323H						W2	W1	W0	Day of the week column	R/W
DATER	0324H			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
MONTHR	0325H				Oct.	Aug.	Apr.	Feb.	Jan.	Month column	R/W
YEARR	0326H	Year 80	Year 40	Year 20	Year 10	Year 8	Year 4	Year 2	Year 1	Year column (Lower two columns)	R/W
PAGER	0327H	Interrupt enable			Adjust-ment function	Clock enable	Alarm enable		PAGE setting	PAGE register	W, R/W
RESTR	0328H	1HZ enable	16HZ enable	Clock reset	Alarm reset	Always write “0”				Reset register	Write only

Note: As for SECR, MINR, HOURR, DAYR, MONTHR, YEARR of PAGE0, current state is read when read it.

Table 3.13.2 PAGE 1 (Alarm function) Registers

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
SECR	0320H										R/W
MINR	0321H		40 min.	20 min.	10 min.	8 min.	4 min.	2 min.	1 min.	Minute column for alarm	R/W
HOURR	0322H			20 /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hour	Hour column for alarm	R/W
DAYR	0323H						W2	W1	W0	Day of the week column for alarm	R/W
DATER	0324H			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column for alarm	R/W
MONTHR	0325H								24/12	24-hour clock mode	R/W
YEARR	0326H							Leap-year setting		Leap-year mode	R/W
PAGER	0327H	Interrupt enable				Clock enable	Alarm enable		PAGE setting	PAGE register	W, R/W
RESTR	0328H	1HZ enable	16HZ enable	Clock reset	Alarm reset	Always write “0”				Reset register	Write only

Note: As for MINR, HOURR, DAYR, MONTHR, YEARR of PAGE1, current state is read when read it.

Not Recommended for New Design

Downloaded from Arrow.com.

3.13.4 Detailed Explanation of Control Register

RTC is not initialized by reset.

Therefore, all registers must be initialized at the beginning of the program.

(1) Second column register (for PAGE0 only)

SECR (0320H)		7	6	5	4	3	2	1	0
	Bit symbol		SE6	SE5	SE4	SE3	SE2	SE1	SE0
	Read/Write		R/W						
	After reset		Undefined						
	Function	"0" is read.	40 sec. column	20 sec. column	10 sec. column	8 sec. column	4 sec. column	2 sec. column	1 sec. column

0	0	0	0	0	0	0	0 s
0	0	0	0	0	0	1	1 s
0	0	0	0	0	1	0	2 s
0	0	0	0	0	1	1	3 s
0	0	0	0	1	0	0	4 s
0	0	0	0	1	0	1	5 s
0	0	0	0	1	1	0	6 s
0	0	0	0	1	1	1	7 s
0	0	0	1	0	0	0	8 s
0	0	0	1	0	0	1	9 s
0	0	1	0	0	0	0	10 s
			:				
0	0	1	1	0	0	1	19 s
0	1	0	0	0	0	0	20 s
			:				
0	1	0	1	0	0	1	29 s
0	1	1	0	0	0	0	30 s
			:				
0	1	1	1	0	0	1	39 s
1	0	0	0	0	0	0	40 s
			:				
1	0	0	1	0	0	1	49 s
1	0	1	0	0	0	0	50 s
			:				
1	0	1	1	0	0	1	59 s

Note: Do not set the data other than showing above.

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Minute column register (for PAGE0/1)

MINR
(0321H)

	7	6	5	4	3	2	1	0
Bit symbol		MI6	MI5	MI4	MI3	MI2	MI1	MI0
Read/Write		R/W						
After reset		Undefined						
Function	"0" is read.	40 min, column	20 min, column	10 min, column	8 min, column	4 min, column	2 min, column	1 min, column

0	0	0	0	0	0	0	0 min.
0	0	0	0	0	0	1	1 min.
0	0	0	0	0	1	0	2 min.
0	0	0	0	0	1	1	3 min.
0	0	0	0	1	0	0	4 min.
0	0	0	0	1	0	1	5 min.
0	0	0	0	1	1	0	6 min.
0	0	0	0	1	1	1	7 min.
0	0	0	1	0	0	0	8 min.
0	0	0	1	0	0	1	9 min.
0	0	1	0	0	0	0	10 min.
			:				
0	0	1	1	0	0	1	19 min.
0	1	0	0	0	0	0	20 min.
			:				
0	1	0	1	0	0	1	29 min.
0	1	1	0	0	0	0	30 min.
			:				
0	1	1	1	0	0	1	39 min.
1	0	0	0	0	0	0	40 min.
			:				
1	0	0	1	0	0	1	49 min.
1	0	1	0	0	0	0	50 min.
			:				
1	0	1	1	0	0	1	59 min.

Note: Do not set the data other than showing above.

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Hour column register (for PAGE0/1)

a. In case of 24-hour clock mode (MONTHR<MO0>=1) of PAGE1

HOURR (0322H)

	7	6	5	4	3	2	1	0
Bit symbol			HO5	HO4	HO3	HO2	HO1	HO0
Read/Write			R/W					
After reset			Undefined					
Function	"0" is read.		20 hour column	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

0	0	0	0	0	0	0 o'clock
0	0	0	0	0	1	1 o'clock
0	0	0	0	1	0	2 o'clock
			:			
0	0	1	0	0	0	8 o'clock
0	0	1	0	0	1	9 o'clock
0	1	0	0	0	0	10 o'clock
			:			
0	1	1	0	0	1	19 o'clock
1	0	0	0	0	0	20 o'clock
			:			
1	0	0	0	1	1	23 o'clock

Note: Do not set the data other than showing above.

b. In case of 12-hour clock mode (MONTHR<MO0>=0) of PAGE1

HOURR (0322H)

	7	6	5	4	3	2	1	0
Bit symbol			HO5	HO4	HO3	HO2	HO1	HO0
Read/Write			R/W					
After reset			Undefined					
Function	"0" is read.		PM/AM	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

0	0	0	0	0	0	0 o'clock (AM)
0	0	0	0	0	1	1 o'clock
0	0	0	0	1	0	2 o'clock
			:			
0	0	1	0	0	1	9 o'clock
0	1	0	0	0	0	10 o'clock
0	1	0	0	0	1	11 o'clock
1	0	0	0	0	0	0 o'clock (PM)
1	0	0	0	0	1	1 o'clock

Note: Do not set the data other than showing above.

Not Recommended for New Design

Downloaded from Arrow.com.

(4) Day of the week column register (for PAGE0/1)

DAYR (0323H)		7	6	5	4	3	2	1	0
	Bit symbol						WE2	WE1	WE0
	Read/Write						R/W		
	After reset						Undefined		
	Function	"0" is read.					W2	W1	W0

0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday

Note: Do not set the data other than showing above.

(5) Day column register (for PAGE0/1)

DATER (0324H)		7	6	5	4	3	2	1	0
	Bit symbol			DA5	DA4	DA3	DA2	DA1	DA0
	Read/Write			R/W					
	After reset			Undefined					
	Function	"0" is read.		Day 20	Day 10	Day 8	Day 4	Day 2	Day 1

0	0	0	0	0	0	0
0	0	0	0	0	1	1st day
0	0	0	0	1	0	2nd day
0	0	0	0	1	1	3rd day
0	0	0	1	0	0	4th day
0	0	1	0	0	1	9th day
0	1	0	0	0	0	10th day
0	1	0	0	0	1	11th day
0	1	1	0	0	1	19th day
1	0	0	0	0	0	20th day
1	0	1	0	0	1	29th day
1	1	0	0	0	0	30th day
1	1	0	0	0	1	31st day

Note1: Do not set the data other than showing above.

Note2: Do not set the day which is not existed. (ex: 30th Feb)

Not Recommended for New Design

Downloaded from Arrow.com.

(6) Month column register (for PAGE0 only)

		7	6	5	4	3	2	1	0
MONTHR (0325H)	Bit symbol				MO4	MO4	MO2	MO1	MO0
	Read/Write				R/W				
	After reset				Undefined				
	Function	"0" is read.			10 months	8 months	4 months	2 months	1 month

0	0	0	0	1	January
0	0	0	1	0	February
0	0	0	1	1	March
0	0	1	0	0	April
0	0	1	0	1	May
0	0	1	1	0	June
0	0	1	1	1	July
0	1	0	0	0	August
0	1	0	0	1	September
1	0	0	0	0	October
1	0	0	0	1	November
1	0	0	1	0	December

Note: Do not set the data other than showing above.

(7) Select 24-hour clock or 12-hour clock (for PAGE1 only)

		7	6	5	4	3	2	1	0
MONTHR (0325H)	Bit symbol								MO0
	Read/Write								R/W
	After reset								Undefined
	Function	"0" is read.							1: 24-hour 0: 12-hour

Not Recommended for New Design

Downloaded from Arrow.com.

(8) Year column register (for PAGE0 only)

YEARR (0326H)

	7	6	5	4	3	2	1	0
Bit symbol	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
Read/Write	R/W							
After reset	Undefined							
Function	80 Years	40 Years	20 Years	10 Years	8 Years	4 Years	2 Years	1 Year

0	0	0	0	0	0	0	0	00 years
0	0	0	0	0	0	0	1	01 years
0	0	0	0	0	0	1	0	02 years
0	0	0	0	0	0	1	1	03 years
0	0	0	0	0	1	0	0	04 years
0	0	0	0	0	1	0	1	05 years
:								
1	0	0	1	1	0	0	1	99 years

Note: Do not set the data other than showing above.

(9) Leap-year register (for PAGE1 only)

YEARR (0326H)

	7	6	5	4	3	2	1	0
Bit symbol							LEAP1	LEAP0
Read/Write							R/W	
After reset							Undefined	
Function	0 is read.						00: Leap year 01: One year after leap year 10: Two years after leap year 11: Three years after leap year	

0	0	Current year is leap year
0	1	Present is next year of a leap year
1	0	Present is two years after a leap year
1	1	Present is three years after leap year

Not Recommended for New Design

Downloaded from Arrow.com.

(10) PAGE register setting (for PAGE0/1)

PAGER (0327H)
Read-modify write instruction are prohibited

	7	6	5	4	3	2	1	0
Bit symbol	INTENA			ADJUST	ENATMR	ENAALM		PAGE
Read/Write	R/W			W	R/W			R/W
After reset	0			Undefined	Undefined			Undefined
Function	INTRTC 0: Disable 1: Enable	"0" is read.		0:Don't care 1:Adjust	Clock 0: Disable 1: Enable	ALARM 0: Disable 1: Enable	"0" is read.	PAGE selection

Note: Pleas keep the setting order below and don't set same time.
(Set difference time to Clock/Alarm setting and interrupt setting)

(Example) Clock setting/Alarm setting

ld (pager), 0ch : Clock, Alarm enable

ld (pager), 8ch : Interrupt enable

PAGE	0	Select Page0
	1	Select Page1

ADJUST	0	Don't care
	1	Adjust sec. counter. When set this bit to "1" the sec. counter become to "0" when the value of sec. counter is 0 – 29. And in case that value of sec. counter is 30-59, min. counter is carried and become sec. counter to "0". Output Adjust signal during 1 cycle of f_{SYS}. After being adjusted once, Adjust is released automatically. (PAGE0 only)

(11) Reset register setting (for PAGE0/1)

RESTR (1328H)
Read-modify write instruction are prohibited

	7	6	5	4	3	2	1	0
Bit symbol	DIS1Hz	DIS16Hz	RSTTMR	RSTALM	RE3	RE2	RE1	RE0
Read/Write	W							
After reset	Undefined							
Function	1Hz 0: Enable 1: Disable	16Hz 0: Enable 1: Disable	1: Clock reset	1: Alarm reset	Always write "0"			

RSTALM	0	Unused
	1	Reset alarm register

Note: When write "1", reset alarm during 1 cycle of f_{SYS}. After that, reset is released automatically.

RSTTMR	0	Unused
	1	Reset divider

Note: When write "1", reset alarm during 1 cycle of f_{SYS}. After that, reset is released automatically.

<DIS1HZ>	<DIS16HZ>	(PAGER) <ENAALM>	Source signal
1	1	1	Alarm
0	1	0	1Hz
1	0	0	16Hz
Others			Output "0"

Downloaded from Arrow.com.
Not Recommended for New Design

3.13.5 Operational Description

(1) Reading Clock data

There is the case which reads wrong data when carry of the inside counter happens during the operation which Clock data reads. Therefore, please read two times with the following way for reading correct data.

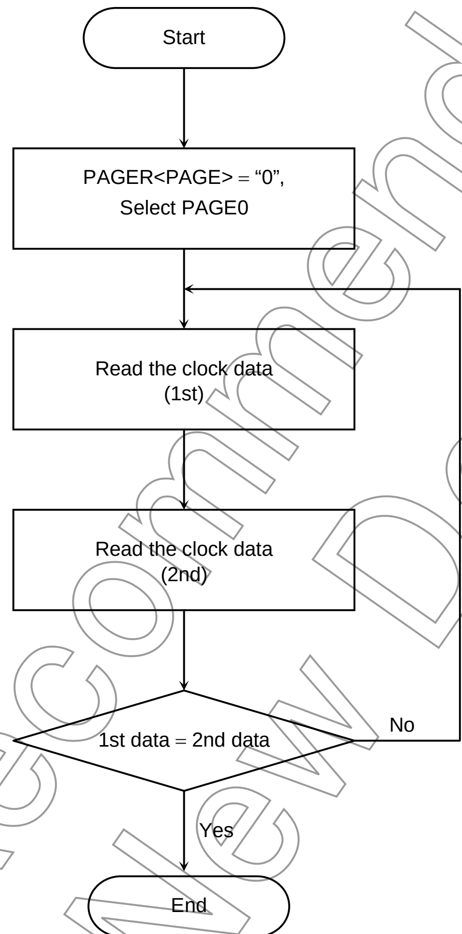


Figure 3.13.2 Flowchart of Clock Data Read

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Timing of INTRTC and Clock data

When time is read by interrupt, read clock data within 0.5s(s) after generating interrupt. This is because count up of clock data occurs by rising edge of 1Hz pulse cycle.

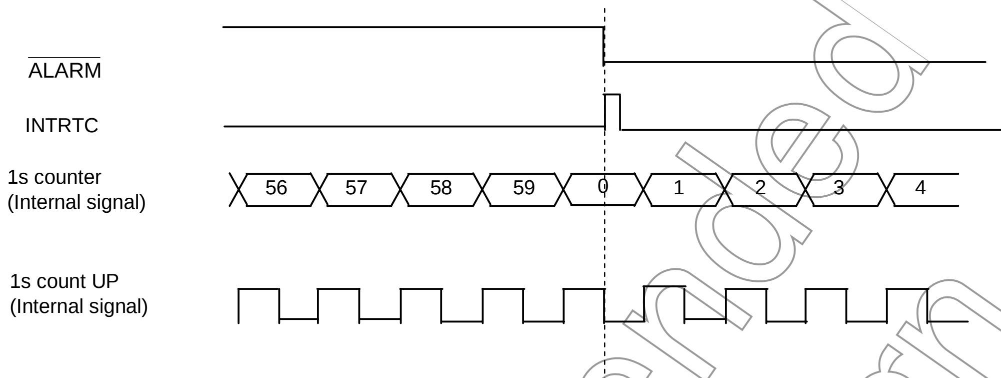


Figure 3.13.3 Timing of INTRTC and Clock data

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Writing Clock data

When there is carry on the way of write operation, expecting data can not be wrote exactly.

Therefore, in order to write in data exactly please follow the below way.

1. Reset for a divider

Inside of RTC, there is 15-stage divider which generates 1 Hz clock from 32.768 kHz. Carry of a Clock is not done for 0.5 second when reset this divider. So write in data during this interval.

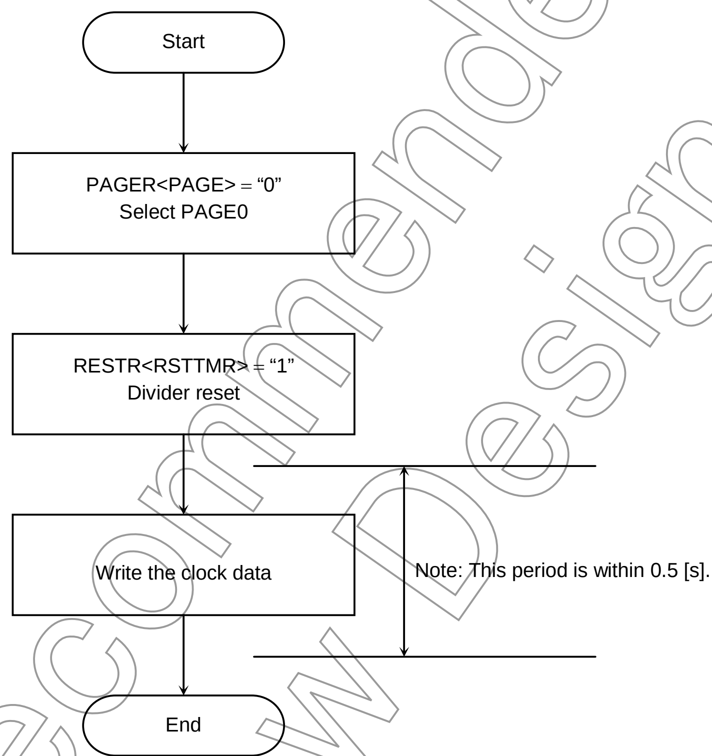


Figure 3.13.4 Flowchart of Data Write

Not Recommended for New Design

Downloaded from Arrow.com.

2. Disabling the Clock

Carry of a clock is prohibited when write “0” to PAGER<ENATMR> and can prevent malfunction by 1s carry hold circuit. During a clock prohibited, 1s carry hold circuit holds one second carry signal, which is generated from divider. After becoming clock enable state, output the carry signal to clock and revise time and continue operation. However, clock is late when clock-disabling state continues for one second or more.

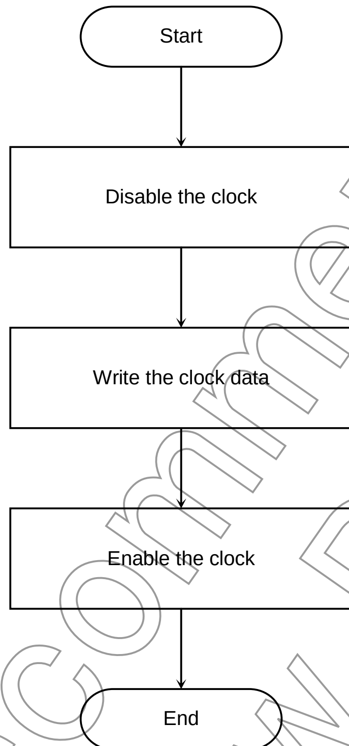


Figure 3.13.5 Flowchart of Clock Disable

Not Recommended for New Design

Downloaded from Arrow.com.

3.13.6 Explanation of the Alarm Function

Can use alarm function by setting of register of PAGE1 and output either of three signals from $\overline{\mathrm{ALARM}}$ pin as follows by write “1” to PAGER<PAGE>. INTRTC outputs 1shot pulse when the falling edge is detected. RTC is not initializes by RESET. Therefore, when clock or alarm function is used, clear interrupt request flag in INTC (interrupt controller).

(1) In accordance of alarm register and the Clock, output 0
(2) Output clock of 1 Hz
(3) Output clock of 16 Hz

(1) In accordance with alarm register and a clock, output “0”

When value of a clock of PAGE0 accorded with alarm register of PAGE1 with a state of PAGER<ENAALM>= “1”, output “0” to $\overline{\mathrm{ALARM}}$ pin and occur INTRTC.

Follows are ways using alarm.

Initialization of alarm is done by writing in “1” at RESTR<RSTALM>, setting value of all alarm becomes don’t care. In this case, always accorded with value of a clock and request INTRTC interrupt if PAGER<ENAALM> is “1”.

Setting alarm min., alarm hour, alarm day and alarm the day week are done by writing in data at each register of PAGE1.

When all setting contents accorded, RTC generates INTRTC interrupt, if PAGER<INTENA><ENAALM> is “1”. However, contents (don't care state) which does not set it up is considered to always accord.

The contents, which set it up once, cannot be returned to don't care state in independence. Initialization of alarm and resetting of alarm register set to “Don’t care”.

The following is an example program for outputting alarm from $\overline{\mathrm{ALARM}}$ -pin at noon (PM12:00) every day.

```
  LD    (PAGER), 09H      ;  Alarm disable, setting PAGE1
  LD    (RESTR), D0H      ;  Alarm initialize
  LD    (DAYR), 01H       ;  W0
  LD    (DATAR),01H       ;  1 day
  LD    (HOURR), 12H      ;  Setting 12 o’clock
  LD    (MINR), 00H       ;  Setting 00 min
                          ;  Set up time 31 µs (Note)
  LD    (PAGER), 0CH      ;  Alarm enable
( LD    (PAGER), 8CH      ;  Interrupt enable )
```

When CPU is operated by high frequency oscillation, it may take a maximum of one clock at 32 kHz (about 30μs) for the time register setting to become valid. In the above example, it is necessary to set 31μs of set up time between setting the time register and enabling the alarm register.

Note: This set up time is unnecessary when you use only internal interruption.

Not Recommended for New Design

Downloaded from Arrow.com.

(4) When output clock of 1 Hz

RTC outputs clock of 1 Hz to $\overline{\text{ALARM}}$ pin by setting up PAGER<ENAALM> = 0, RESTR<DIS1HZ> = 0, <DIS16HZ> = 1. And RTC generates INTRTC interrupt by falling edge of the clock.

(5) When output clock of 16 Hz

RTC outputs clock of 16 Hz to $\overline{\text{ALARM}}$ pin by setting up PAGER<ENAALM> = 0, RESTR<DIS1HZ> = 1, <DIS16HZ> = 0. And RTC generates INTRTC interrupt by falling edge of the clock.

Not Recommended for New Design

Downloaded from Arrow.com.

3.14 Melody/Alarm Generator (MLD)

TMP91C824 incorporates melody function and alarm function, both of which are output from the MLDALM pin. 5 kinds of fixed cycle interrupts are generated by the 15-bit free-run counter, which is used for alarm generator.

Features are as follows.

- Melody generator

The melody function generates signals of any frequency (4 Hz to 5461 Hz) based on low-speed clock (32.768 kHz) and outputs several signals from the MLDALM pin.

By connecting a loud speaker outside, melody tone can sound easily.

- Alarm generator

The alarm function generates 8 kinds of alarm waveform having a modulation frequency (4096 Hz) determined by the low-speed clock (32.768 kHz). And this waveform is able to invert by setting a value to a register.

By connecting a loud speaker outside, alarm tone can sound easily.

And also 5 kinds of fixed cycle (1 Hz, 2 Hz, 64 Hz, 512 Hz and 8192 Hz) interrupts are generated by the free-run counter which is used for alarm generator.

This section is constituted as follows.

3.14.1 Block Diagram
3.14.2 Control Registers
3.14.3 Operational Description
(1) Melody generator
(2) Alarm generator

Not Recommended for New Design

Downloaded from Arrow.com.

3.14.1 Block Diagram

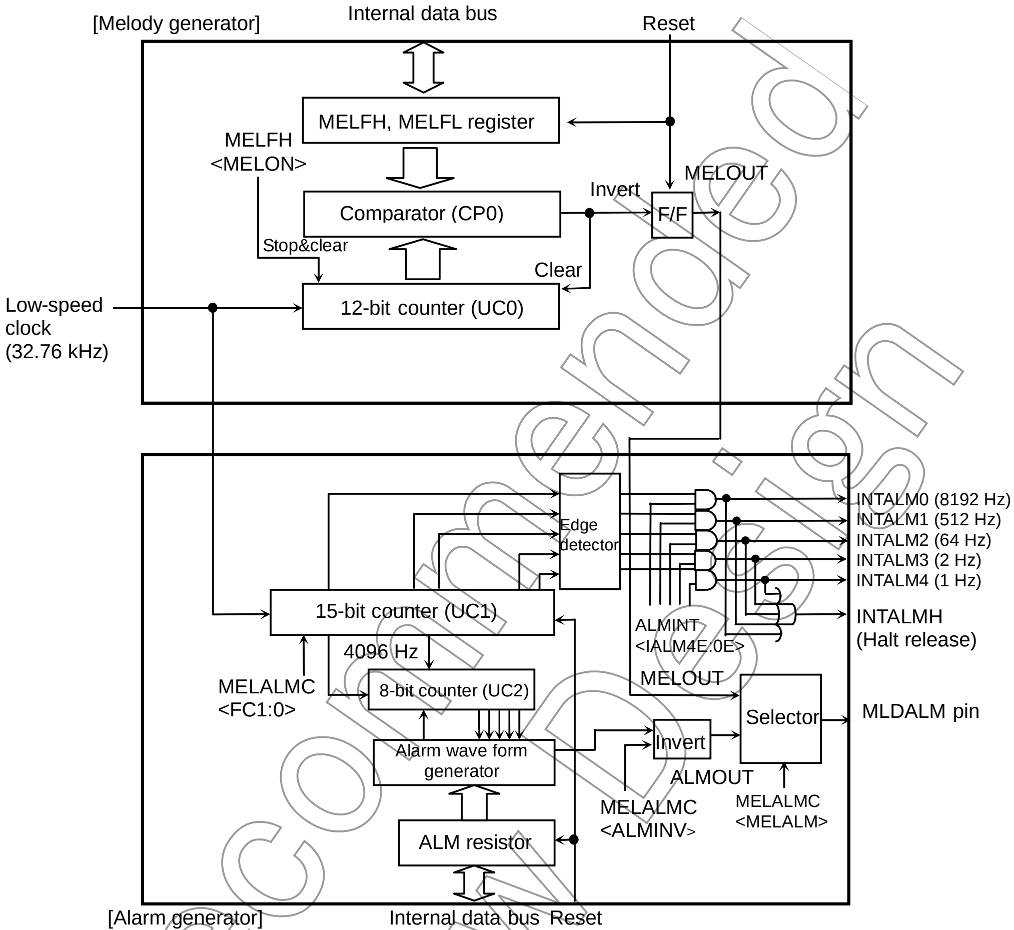


Figure 3.14.1 MLD Block Diagram

Not Recommended for New Design

Downloaded from Arrow.com.

3.14.2 Control Registers

ALM Register

ALM (0330H)

	7	6	5	4	3	2	1	0
Bit symbol	AL8	AL7	AL6	AL5	AL4	AL3	AL2	AL1
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Setting alarm pattern							

MELALMC Register

MELALMC (0331H)

	7	6	5	4	3	2	1	0
Bit symbol	FC1	FC0	ALMINV	–	–	–	–	MELALM
Read/Write	R/W		R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Free-run counter control 00: Hold 01: Restart 10: Clear 11: Clear and start		Alarm waveform invert 1: Invert	Always write 0				Output waveform select 0: Alarm 1: Melody

Note 1: MELALMEC<FC1> is read always 0.

Note 2: When setting MELALMC register except <FC1:0> during the free-run counter is running, <FC1:0> is kept 01.

MELFL Register

MELFL (0332H)

	7	6	5	4	3	2	1	0
Bit symbol	ML7	ML6	ML5	ML4	ML3	ML2	ML1	ML0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Setting melody frequency (Lower 8 bits)							

MELFH Register

MELFH (0333H)

	7	6	5	4	3	2	1	0
Bit symbol	MELON				ML11	ML10	ML9	ML8
Read/Write	R/W				R/W			
After reset	0				0	0	0	0
Function	Control melody counter 0: Stop & clear 1: Start				Setting melody frequency (Upper 4 bits)			

ALMINT Register

ALMINT (0334H)

	7	6	5	4	3	2	1	0
Bit symbol			–	IALM4E	IALM3E	IALM2E	IALM1E	IALM0E
Read/Write			R/W	R/W				
After reset			0	0	0	0	0	0
Function			Always write 0	1: Interrupt enable for INTALM4 to INTALM0				

Not Recommended for New Design

Downloaded from Arrow.com.

3.14.3 Operational Description

(1) Melody generator

The melody function generates signals of any frequency (4 Hz to 5461 Hz) based on low-speed clock (32.768 kHz) and outputs the signals from the MLDALM pin.

By connecting a loud speaker outside, melody tone can sound easily.

(Operation)

At first, MELALMC<MELALM> have to be set as 1 in order to select melody waveform as output waveform from MLDALM. Then melody output frequency has to be set to 12-bit register MELFH, MELFL.

Followings are setting example and calculation of melody output frequency.

(Formula for calculating of melody waveform frequency)

at fs = 32.768 [kHz]

melody output waveform f_{MLD} [Hz] = $32768/(2 \times N + 4)$

setting value for melody $N = (16384/f_{MLD}) - 2$

(notice: N = 1 to 4095(001H to FFFH), 0 is not acceptable)

(Example program)

In case of outputting La musical scale (440 Hz)

```
LD   (MELALMC), ---XXXX1B   ; select melody waveform
LD   (MELFL), 23H           ; N= 16384/440 – 2 = 35.2 = 023H
LD   (MELFH), 80H           ; start to generate waveform
```

(Refer to Basic musical scale setting table)

Scale	Frequency [Hz]	Register Value: N
C	264	03CH
D	297	035H
E	330	030H
F	352	02DH
G	396	027H
A	440	023H
B	495	01FH
C	528	01DH

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Alarm generator

The alarm function generates 8 kinds of alarm waveform having a modulation frequency 4096 Hz determined by the low-speed clock (32.768 kHz). And this waveform is reversible by setting a value to a register.

By connecting a loud speaker outside, alarm tone can sound easily.

5 kinds of fixed cycle (1 Hz, 2 Hz, 64 Hz, 512 Hz and 8192 Hz) interrupts are generated by the free-run counter, which is used for alarm generator.

(Operation)

At first, MELALMC<MELALM> have to be set as 0 in orders to select alarm waveform as output waveform from MLDALM. Then 10 be set on MELALMC<FC1:0> register, and clear internal counter. Finally alarm pattern has to be set on 8-bit register of ALM. If it is inverted output-data, set <ALMINV> as invert.

Followings are example program, setting value of alarm pattern and waveform of each setting value.

(Setting value of alarm pattern)

Setting Value for ALM Register	Alarm Waveform
00H	0 fixed
01H	AL1 pattern
02H	AL2 pattern
04H	AL3 pattern
08H	AL4 pattern
10H	AL5 pattern
20H	AL6pattern
40H	AL7 pattern
80H	AL8 pattern
Other	Undefined (do not set)

(Example program)

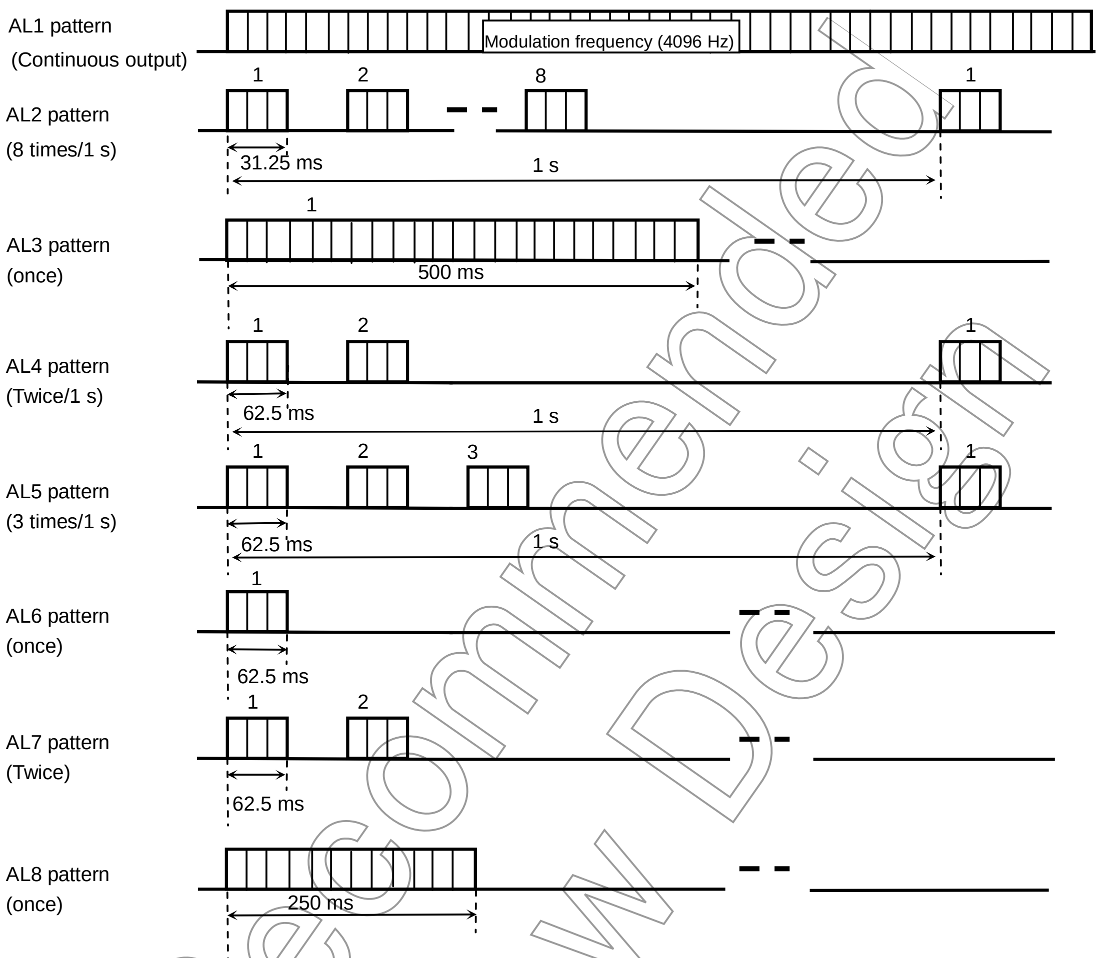
In case of outputting AL2 pattern (31.25 ms/8 times/1 s)

```
LD    (MELALMC), C0H        ; set output alarm waveform
                            ; free-run counter start
LD    (ALM), 02H            ; set AL2 pattern, start
```

Not Recommended for New Design

Downloaded from Arrow.com.

Example: Waveform of alarm pattern for each setting value: Not invert



Not Recommended for New Design

Downloaded from Arrow.com.

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Power supply voltage	Vcc	−0.5 to 4.0	V
Input voltage	VIN	−0.5 to Vcc + 0.5	V
Output current	IOL	2	mA
Output current	IOH	−2	mA
Output current (total)	ΣIOL	80	mA
Output current (total)	ΣIOH	−80	mA
Power dissipation (Ta = 85°C)	PD	600	mW
Soldering temperature (10 s)	TSOLDER	260	°C
Storage temperature	TSTG	−65 to 150	°C
Operating temperature	TOPR	−40 to 85	°C

Note: The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

Solderability of lead free products

Test parameter	Test condition	Note
Solderability	Use of Sn-37Pb solder Bath Solder bath temperature = 230°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: solderability rate until forming ≥ 95%
	Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux (use of lead free)	

Not Recommended for New Design

Downloaded from Arrow.com.

4.2 DC Characteristics (1/2)

Parameter		Symbol	Condition		Min	Typ. (Note)	Max	Unit
Power supply voltage (AVCC = DVCC) (AVSS = DVSS = 0 V)		VCC	fc = 2 to 33 MHz	fs = 30 to 34 kHz	2.7		3.6	V
			fc = 2 to 10 MHz		1.8			
Input low voltage	D0 to D15	VIL	Vcc ≥ 2.7V		−0.3		0.6	V
			Vcc < 2.7V				0.2 Vcc	
	P52 to PD7 (Except PB3)	VIL1	Vcc ≥ 2.7V				0.3 Vcc	
			Vcc < 2.7V				0.2 Vcc	
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, PB3 (INT0)	VIL2	Vcc ≥ 2.7V				0.25 Vcc	
			Vcc < 2.7V				0.15 Vcc	
	AM0 to AM1	VIL3	Vcc ≥ 2.7V				0.3	
			Vcc < 2.7V				0.3	
	X1	VIL4	Vcc ≥ 2.7V				0.2 Vcc	
			Vcc < 2.7V				0.1 Vcc	
Input high voltage	D0 to D15	VIH	3.6V ≥ Vcc > 3.3V		2.4		Vcc + 0.3	
			3.3V ≥ Vcc ≥ 2.7V		2.0			
			Vcc < 2.7V		0.7 Vcc			
	P52 to PD7 (Except PB3)	VIH1	Vcc ≥ 2.7V		0.7 Vcc			
			Vcc < 2.7V		0.8 Vcc			
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, PB3 (INT0)	VIH2	Vcc ≥ 2.7V		0.75 Vcc			
			Vcc < 2.7V		0.85 Vcc			
	AM0 to AM1	VIH3	Vcc ≥ 2.7V		Vcc − 0.3			
			Vcc < 2.7V		Vcc − 0.3			
	X1	VIH4	Vcc ≥ 2.7V		0.8 Vcc			
			Vcc < 2.7V		0.9 Vcc			
Output low voltage		VOL	IOL = 1.6 mA	Vcc ≥ 2.7V			0.45	V
			IOL = 0.4 mA	Vcc < 2.7V			0.15 Vcc	
Output high voltage		VOH	IOH = −400 μA	Vcc ≥ 2.7V	Vcc − 0.3			
			IOH = −200 μA	Vcc < 2.7V	0.8 Vcc			

Note: Typical values are for when Ta = 25°C and Vcc = 3.0 V uncles otherwise noted.

Not Recommended for New Design

Downloaded from Arrow.com.

4.2 DC Characteristics (2/2)

Parameter	Symbol	Condition	Min	Typ. (Note1)	Max	Unit
Input leakage current	ILI	0.0 ≤ VIN ≤ Vcc		0.02	±5	μA
Output leakage current	ILO	0.2 ≤ VIN ≤ Vcc − 0.2		0.05	±10	μA
Power down voltage (at STOP, RAM back up)	VSTOP	VIL2 = 0.2 Vcc, VIH2 = 0.8 Vcc	1.8		3.6	V
$\overline{\text{RESET}}$ pull-up resistor	RRST	3.6 V ≥ Vcc ≥ 2.7 V	80		400	kΩ
		Vcc = 2 V ± 10%	200		1000	
Pin capacitance	CIO	fc = 1 MHz			10	pF
Schmitt width $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0	VTH	Vcc ≥ 2.7 V	0.4	0.9		V
		Vcc < 2.7 V	0.3	0.7		
Programmable pull-up resistor	RKH	3.6 V ≥ Vcc ≥ 2.7 V	80		400	kΩ
		Vcc = 2 V ± 10%	200		1000	
NORMAL (Note 2)	Icc	3.6 V ≥ Vcc ≥ 2.7 V fc = 33 MHz		14.0	20.0	mA
IDLE2				4.0	6.1	
IDLE1				1.2	2.2	
NORMAL (Note 2)		Vcc = 2 V ± 10% fc = 10 MHz (Typ.: Vcc = 2.0 V)		2.6	3.0	
IDLE2				0.7	1.2	
IDLE1				0.2	0.4	
SLOW (Note 2)		3.6 V ≥ Vcc ≥ 2.7 V fs = 32.768 kHz		17.5	30.5	μA
IDLE2				7.0	13.5	
IDLE1				5.0	10.0	
SLOW (Note 2)		Vcc = 2 V ± 10% fs = 32.768 kHz (Typ.: Vcc = 2.0 V)		10.5	13.0	
IDLE2				4.5	6.5	
IDLE1				3.0	4.5	
STOP		3.6 V ≥ Vcc ≥ 1.8 V		0.2	15	

Note 1: Typical values are for when Ta = 25°C and Vcc = 3.0 V unless otherwise noted.

Note 2: Icc measurement conditions (NORMAL, SLOW):
All functions are operational; output pins are open and input pins are fixed. Data and address bus CL = 30 pF loaded.

Not Recommended for New Design

Downloaded from Arrow.com.

4.3 AC Characteristics

(1) Vcc = 3.0 V ± 10%

No.	Parameter	Symbol	Variable		f_{FPH} = 33 MHz		Unit
			Min	Max	Min	Max	
1	f_{FPH} period (= x)	t_{FPH}	30.3	31250	30.3		ns
2	A0 to A23 valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{AC}	x – 23		7		ns
3	$\overline{RD}$ rise → A0 to A23 hold	t_{CAR}	0.5x –13		2		ns
4	$\overline{WR}$ rise → A0 to A23 hold	t_{CAW}	x – 13		17		ns
5	A0 to A23 valid → D0 to D15 input	t_{AD}		3.5x – 24		82	ns
6	$\overline{RD}$ fall → D0 to D15 input	t_{RD}		2.5x – 24		51	ns
7	$\overline{RD}$ low width	t_{RR}	2.5x – 15		60		ns
8	$\overline{RD}$ rise → D0 to A15 hold	t_{HR}	0		0		ns
9	$\overline{WR}$ low width	t_{WW}	2.0x – 15		45		ns
10	D0 to D15 valid → $\overline{WR}$ rise	t_{DW}	1.5x – 35		10		ns
11	$\overline{WR}$ rise → D0 to D15 hold	t_{WD}	x – 25		5		ns
12	A0 to A23 valid → $\overline{WAIT}$ input (1 + N) WAIT mode	t_{AW}		3.5x – 60		46	ns
13	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold (1 + N) WAIT mode	t_{CW}	2.5x + 0		76		ns
14	A0 to A23 valid → Port input	t_{APH}		3.5x – 89		17	ns
15	A0 to A23 valid → Port hold	t_{APH2}	3.5x		106		ns
16	A0 to A23 valid → Port valid	t_{APO}		3.5x + 60		166	ns

AC measuring conditions

- Output level: High = 0.7 Vcc, Low = 0.3 Vcc, CL = 50 pF
- Input level: High = 0.9 Vcc, Low = 0.1 Vcc

Note: Symbol x in the above table means the period of clock f_{FPH}, it's half period of the system clock f_{SYS} for CPU core. The period of f_{FPH} depends on the clock gear setting or the selection of high/low oscillator frequency.

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Vcc = 2.0 V ± 10%

No.	Parameter	Symbol	Variable		10 MHz		Unit
			Min	Max	Min	Max	
1	f_{FPH} period (= x)	tFPH	100	31250	100		ns
2	A0 to A15 valid → $\overline{RD}$ / $\overline{WR}$ fall	tAC	x – 46		54		ns
3	$\overline{RD}$ rise → A0 to A23 hold	tCAR	0.5x – 30		20		ns
4	$\overline{WR}$ rise → A0 to A23 hold	tCAW	x – 26		74		ns
5	A0 to A23 valid → $\overline{RD}$ / $\overline{WR}$ fall	tAD		3.5x – 48		302	ns
6	$\overline{RD}$ fall → D0 to D15 input	tRD		2.5x – 48		202	ns
7	$\overline{RD}$ low width	tRR	2.5x – 30		220		ns
8	$\overline{RD}$ rise → D0 to D15 hold	tHR	0		0		ns
9	$\overline{WR}$ low width	tWW	2.0x – 30		170		ns
10	D0 to D15 valid → $\overline{WR}$ rise	tDW	1.5x – 70		80		ns
11	$\overline{WR}$ rise →D0 to D15 Hold	tWD	x – 50		50		ns
12	A0 to A23 valid → $\overline{WAIT}$ input (1 + N) WAIT mode	tAW		3.5x – 120		230	ns
13	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold (1 + N) WAIT mode	tCW	2.5x + 0		250		ns
14	A0 to A23 valid → Port input	tAPH		3.5x – 50		300	ns
15	A0 to A23 valid → Port hold	tAPH2	3.5x		350		ns
16	A0 to A23 valid → Port valid	tAPO		3.5x + 60		410	ns

AC measuring conditions

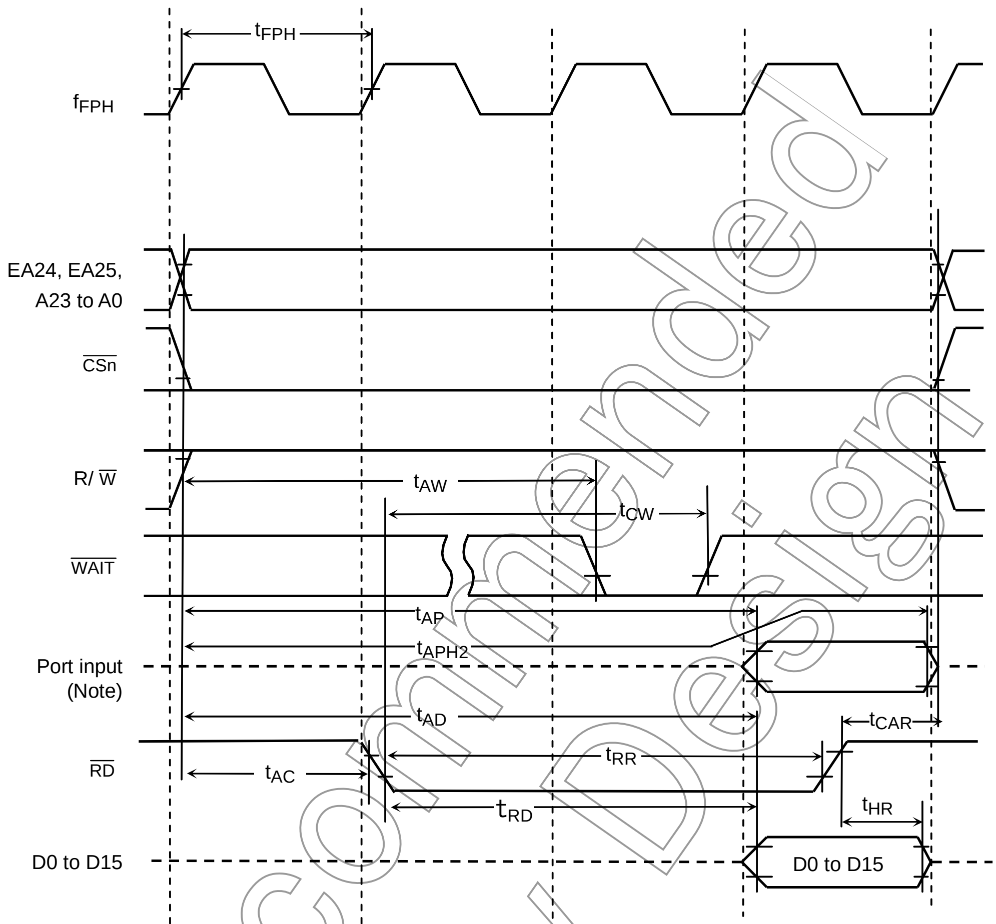
- Output level: High = 0.7 V, Low = 0.3 V, CL = 50 pF
- Input level: High = 0.9 V, Low = 0.1V

Note: Symbol x in the above table means the period of clock f_{FPH}, it's half period of the system clock f_{SYS} for CPU core. The period of f_{FPH} depends on the clock gear setting or the selection of high/low oscillator frequency.

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Read cycle

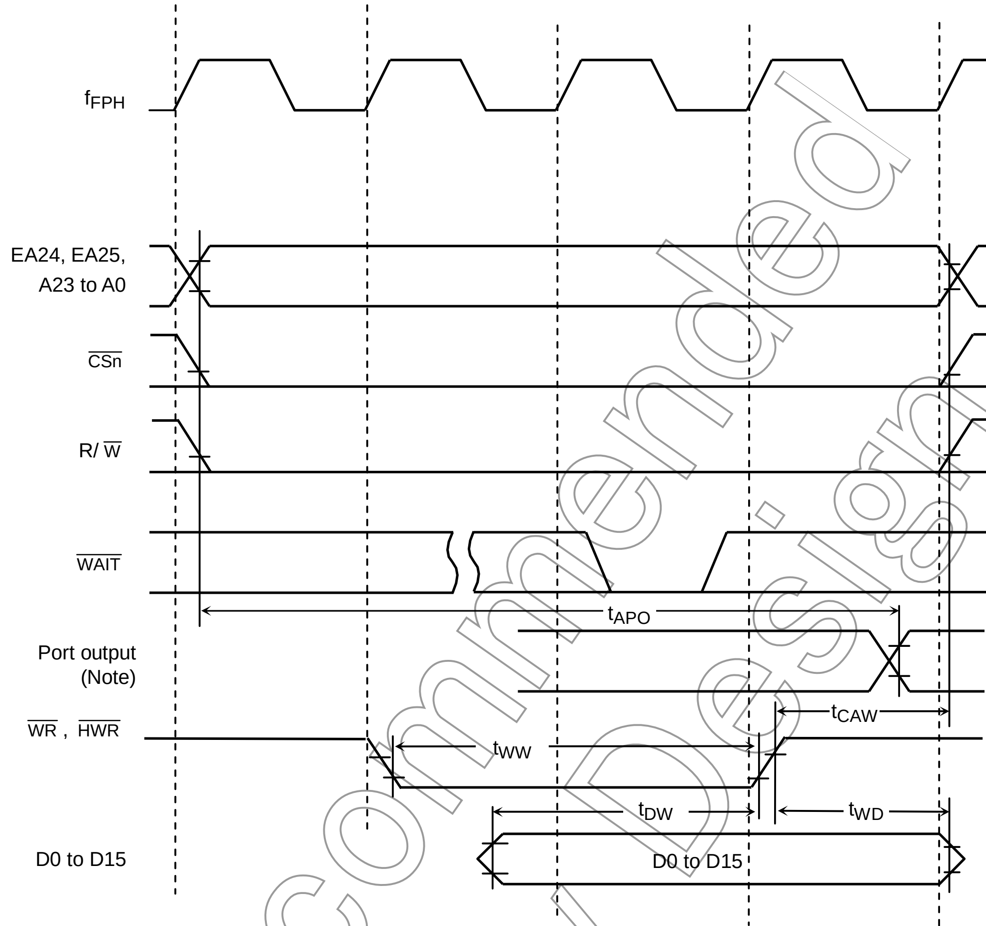


Note: Since the CPU accesses the internal area to read data from a port, the control signals of external pins such as RD and CS are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

Not Recommended for New Design

Downloaded from Arrow.com.

(4) Write cycle



Note: Since the CPU accesses the internal area to write data to a port, the control signals of external pins such as $\overline{WR}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

Not Recommended for New Design

Downloaded from Arrow.com.

4.4 AD Conversion Characteristics

AVcc = Vcc, AVss = Vss

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage (+)	VREFH	3.6 V ≥ V_{CC} ≥ 2.7 V	V_{CC} − 0.2 V	Vcc	Vcc	V
		V_{CC} = 2 V ± 10%	V_{CC}	Vcc	Vcc	
Analog reference voltage (−)	VREFL	3.6 V ≥ V_{CC} ≥ 2.7 V	V_{SS}	Vss	Vss + 0.2 V	
		V_{CC} = 2 V ± 10%	V_{SS}	Vss	Vss	
Analog input voltage range	VAIN		V_{REFL}		V_{REFH}	
Analog current for analog reference voltage <VREFON> = 1	IREF (VREFL = 0 V)	3.6 V ≥ V_{CC} ≥ 2.7 V		0.94	1.35	mA
		V_{CC} = 2 V ± 10%		0.65	0.90	
<VREFON> = 0		3.6 V ≥ V_{CC} ≥ 2.7 V		0.02	5.0	μA
Error (Not including quantizing errors)	−	3.6 V ≥ V_{CC} ≥ 2.7 V		±1.0	±4.0	LSB
		V_{CC} = 2 V ± 10%		±1.0	±4.0	

Note 1: 1 LSB = (VREFH − VREFL)/1024 [V]

Note 2: The operation above is guaranteed for f_{FPH} ≥ 4 MHz.

Note 3: The value for I_{CC} includes the current which flows through the AVCC pin.

Not Recommended for New Design

Downloaded from Arrow.com.

4.5 Serial Channel Timing (I/O Internal Mode)

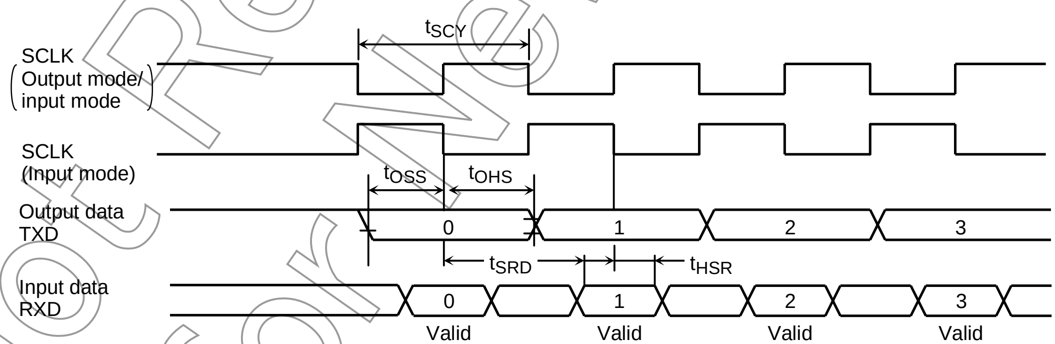
(1) SCLK input mode

Parameter		Symbol	Variable Min	Variable Max	10 MHz Min	10 MHz Max	27 MHz Min	27 MHz Max	Unit
SCLK period		t_{SCY}	16X		1.6		0.59		μs
Output data → SCLK rising/falling edge*	Vcc = 3 V ± 10%	t_{OSS}	t_{SCY}/2 − 4X − 110		290		38		ns
	Vcc = 2 V ± 10%		t_{SCY}/2 − 4X − 180		220		−		ns
SCLK rising/falling edge* → Output data hold		t_{OHS}	t_{SCY}/2 + 2X + 0		1000		370		ns
SCLK rising/falling edge* → Input data hold		t_{HSR}	3X + 10		310		121		ns
SCLK rising/falling edge* → Valid data input		t_{SRD}		t_{SCY} − 0		1600		592	ns
Valid data input → SCLK rising/falling edge*		t_{RDS}	0		0		0		ns

Note: SCLK rising/falling edge: The rising edge is used in SCLK rising mode.
The falling edge is used in SCLK falling mode.

(2) SCLK output mode

Parameter	Symbol	Variable Min	Variable Max	10 MHz Min	10 MHz Max	27 MHz Min	27 MHz Max	Unit
SCLK period	t_{SCY}	16X	8192X	1.6	819	0.59	303	μs
Output data → SCLK rising/falling edge*	t_{OSS}	t_{SCY}/2 − 40		760		256		ns
SCLK rising/falling edge* → Output data hold	t_{OHS}	t_{SCY}/2 − 40		760		256		ns
SCLK rising/falling edge* → Input data hold	t_{HSR}	0		0		0		ns
SCLK rising/falling edge* → Valid data input	t_{SRD}		t_{SCY} − 1X − 180		1320		375	ns
Valid data input → SCLK rising/falling edge*	t_{RDS}	1X + 180		280		217		ns



Not Recommended for New Design

Downloaded from Arrow.com.

4.6 Event Counter (TA0IN)

Parameter	Symbol	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock period	t_{VCK}	8X + 100		900		396		ns
Clock low level width	t_{VCKL}	4X + 40		440		188		ns
Clock high level width	t_{VCKH}	4X + 40		440		188		ns

4.7 Interrupt, Capture

(1) $\overline{\text{NMI}}$, INT0 to INT3 interrupts

Parameter	Symbol	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{\text{NMI}}$, INT0 to INT3 low level width	t_{INTAL}	4X + 40		440		188		ns
$\overline{\text{NMI}}$, INT0 to INT3 high level width	t_{INTAH}	4X + 40		440		188		ns

4.8 SCOUT Pin AC Characteristics

Parameter	Symbol	Variable		10MHz		27 MHz		Condition	Unit
		Min	Max	Min	Max	Min	Max		
Low level width	t_{SCH}	0.5T – 10		90		27		Vcc ≥ 2.7 V	ns
		0.5T – 30		70		–		Vcc < 2.7 V	
High level width	t_{SCL}	0.5T – 10		90		27		Vcc ≥ 2.7 V	ns
		0.5T – 30		70		–		Vcc < 2.7 V	

Note: T = Period of SCOUT

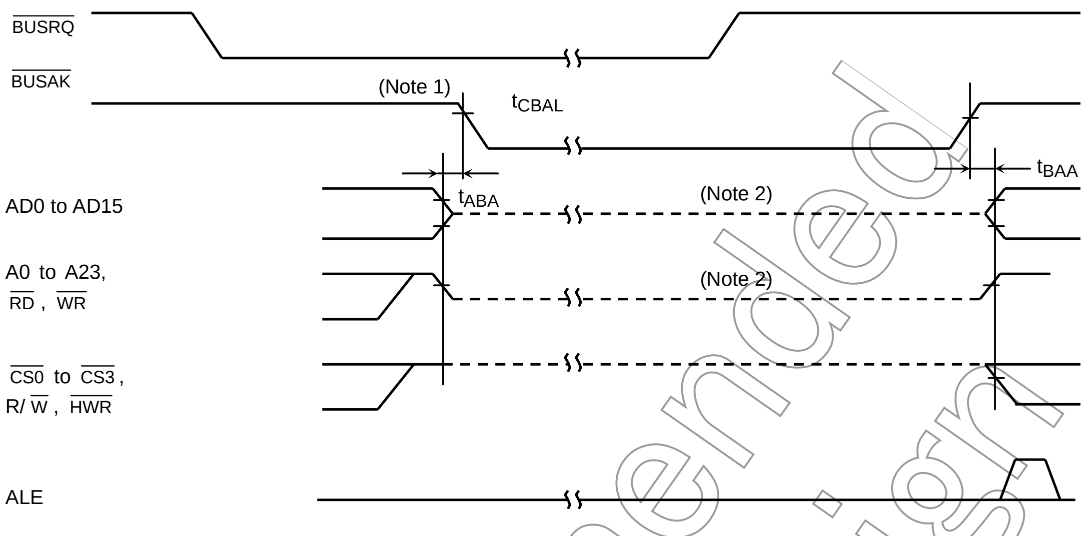
Measuring conditions

- Output level: High = 0.7 V, Low = 0.3 V, CL = 10 pF

Not Recommended for New Design

Downloaded from Arrow.com.

4.9 Bus Request/Bus Acknowledge



Symbol	Parameter	Variable		f_{FPH} = 4 MHz		f_{FPH} = 16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{ABA}	Output buffer off to $\overline{\text{BUSAK}}$ low	0	80	0	80	0	80	ns
t_{BAA}	$\overline{\text{BUSAK}}$ high to output buffer on	0	80	0	80	0	80	ns

Note 1: Even if the $\overline{\text{BUSRQ}}$ signal goes low, the bus will not be released while the $\overline{\text{WAIT}}$ signal is low. The bus will only be released when $\overline{\text{BUSRQ}}$ goes low while $\overline{\text{WAIT}}$ is high.

Note 2: This line shows only that the output buffer is in the off state.
It does not indicate that the signal level is fixed.
Just after the bus is released, the signal level set before the bus was released is maintained dynamically by the external capacitance. Therefore, to fix the signal level using an external resister during bus release, careful design is necessary, since fixing of the level is delayed. The internal programmable pull-up/pull-down resistor is switched between the active and non-active states by the internal signal.

Not Recommended for New Design

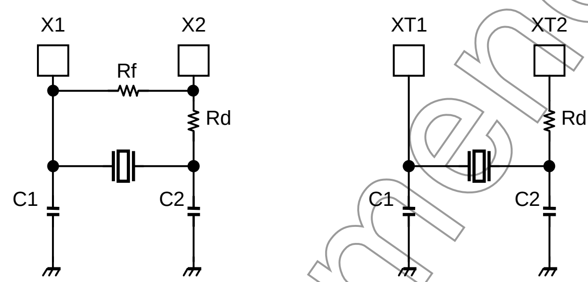
Downloaded from Arrow.com.

4.10 Recommended Crystal Oscillation Circuit

TMP91C824 is evaluated by below oscillator vender. When selecting external parts, make use of this information.

Note: Total loads value of oscillator is sum of external loads (C1 and C2) and floating loads of actual assemble board. There is a possibility of miss-operating using C1 and C2 value in below table. When designing board, it should design minimum length pattern around oscillator. And we recommend that oscillator evaluation try on your actual using board.

(1) Connection example



High-frequency oscillator Low-frequency oscillator

Not Recommended for New Design

Downloaded from Arrow.com.

(2) TMP91C824 recommended ceramic oscillator: Murata Manufacturing Co., Ltd. (JAPAN)

Circuit parameter recommended

MCU	Oscillation Frequency [MHz]	Item of Oscillator	Parameter of Elements				Running Condition	
			C1 [pF]	C2 [pF]	Rf [Ω]	Rd [Ω]	Voltage of Power [V]	Tc [°C]
TMP91C824	2.00	CSTLS2M00G56-B0	(47)	(47)	Open	0	1.8 to 2.2	−40 to +85
	2.50	CSTLS2M50G56-B0	(47)	(47)	Open	0		
	10.00	CSTS1000MG03 *CSTLS10M0G53-B0	(15)	(15)	Open	0		
	12.50	CSA12.5MTZ093 *CSALA12M5T55093-B0	30	30	Open	0		
		CST12.0MTW093 *CSTLA12M5T55093-B0	(30)	(30)	Open	0		

MCU	Oscillation Frequency [MHz]	Item of Oscillator	Parameter of Elements				Running Condition	
			C1 [pF]	C2 [pF]	Rf [Ω]	Rd [Ω]	Voltage of Power [V]	Tc [°C]
TMP91C824	4.00	CSTS0400MG06 *CSTLS4M00G56-B0	(47)	(47)	Open	0	2.7 to 3.6	−40 to +85
	6.750	CSTS0675MG06 *CSTLS6M75G56-B0	(47)	(47)	Open	0		
	12.50	CSA12.5MTZ *CSALA12M5T55-B0	30	30	Open	0		
		CST12.0MTW *CSTLA12M5T55-B0	(30)	(30)	Open	0		
	20.00	CSALS20M0X53-B0	5	5	Open	0		
		CSTLS20M0X51-B0	(5)	(5)	Open	0		
	27.00	CSALS27M0X51-B0	Open	Open	10k	0		
	32.00	CSALA32M0X51-B0	3	3	Open	0		

NOTE: In CST ***type oscillator, Capacitance C1, C2 is built in

* After 2001/06,new products will be made, and the old products (Now in production) will not be made in Murata Manufacturing Co., Ltd. (JAPAN)

- The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change.
  For up-to-date information, please refer to the following URL:
  http://www.murata.co.jp/search/index.html

Not Recommended for New Design

Downloaded from Arrow.com.

5. Table of SFRs

The SFRs (Special function registers) include the I/O ports and peripheral control registers allocated to the 4-Kbyte address space from 000FE0H to 000FFFH.

(1) I/O port

(2) I/O port control

(3) Interrupt control

(4) Chip select/wait control

(5) Clock gear

(6) DFM (Clock doubler)

(7) 8-bit timer

(8) UART/Serial channel

(9) I²C bus/serial channel

(10) AD converter

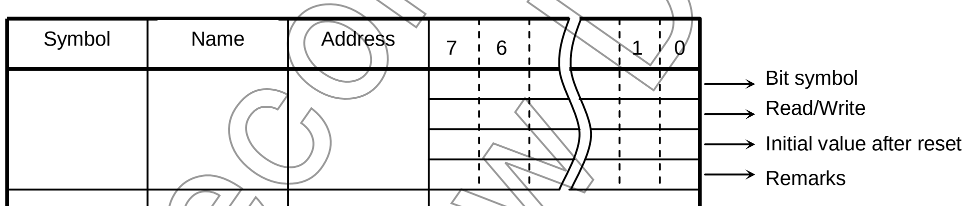
(11) Watchdog timer

(12) RTC (Real time clock)

(13) Melody/alarm generator

(14) MMU

Table layout



Note: Prohibit RMW in the table means that you cannot use RMW instructions on these register.

Example: When setting bit0 only of the register PxCR, the instruction SET 0, (PxCR) cannot be used. The LD (Transfer) instruction must be used to write all eight bits.

Read/Write

R/W: Both read and write are possible.

R: Only read is possible.

W: Only write is possible.

W*: Both read and write are possible (when this bit is read as 1).

Prohibit RMW: Read-modify-write instructions are prohibited. (The EX, ADD, ADC, BUS, SBC, INC, DEC, AND, OR, XOR, STCF, RES, SET, CHG, TSET, RLC, RRC, RL, RR, SLA, SRA, SLL, SRL, RLD and RRD instruction are read-modify-write instructions.)

R/W*: Read-modify-write instructions are prohibited when controlling the pull-up resistor.

Not Recommended for New Design

Downloaded from Arrow.com.

Table 5.1 Address Map SFRs

[1], [2] Port

Address	Name
0000H	
1H	P1
2H	
3H	
4H	P1CR
5H	
6H	P2
7H	
8H	
9H	P2FC
AH	P5CR
BH	P5FC
CH	
DH	P5
EH	
FH	

Address	Name
0010H	
1H	
2H	P6
3H	P7
4H	
5H	P6FC
6H	P7CR
7H	P7FC
8H	P8
9H	
AH	
BH	P6FC2
CH	P7FC2
DH	
EH	
FH	P7ODE

Address	Name
0022H	
1H	
2H	PB
3H	PC
4H	PBCR
5H	PBFC
6H	PCCR
7H	PCFC
8H	PCODE
9H	PD
AH	PDFC
BH	
CH	
DH	
EH	
FH	

Address	Name
0070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	PZ
EH	PZCR
FH	PZFC

[3] INTC

Address	Name
0080H	DMA0V
1H	DMA1V
2H	DMA2V
3H	DMA3V
4H	
5H	
6H	
7H	
8H	INTCLR
9H	DMAR
AH	DMAB
BH	
CH	IIMC
DH	
EH	
FH	

Address	Name
0090H	INTE0AD
1H	INTE12
2H	INTE3ALM4
3H	INTEALM01
4H	INTEALM23
5H	INTETA01
6H	INTETA23
7H	INTERTC
8H	INTES0
9H	INTES1
AH	INTES2
BH	INTETC01
CH	INTETC23
DH	INTEP01
EH	
FH	

[4] CS/WAIT

Address	Name
00C0H	B0CS
1H	B1CS
2H	B2CS
3H	B3CS
4H	
5H	
6H	
7H	BEXCS
8H	MSAR0
9H	MAMR0
AH	MSAR1
BH	MAMR1
CH	MSAR2
DH	MAMR2
EH	MSAR3
FH	MAMR3

[5], [6] CGEAR, DFM

Address	Name
00E0H	SYSCR0
1H	SYSCR1
2H	SYSCR2
3H	EMCCR0
4H	EMCCR1
5H	EMCCR2
6H	EMCCR3
7H	
8H	DFMCR0
9H	DFMCR1
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses, e.g., addresses to which no register has been allocated.

Not Recommended for New Design

Downloaded from Arrow.com.

Table 5.2 Address Map SFRs

[7] TMRA

Address	Name
0100H	TA01RUN
1H	
2H	TA0REG
3H	TA1REG
4H	TA01MOD
5H	TA01FFCR
6H	
7H	
8H	TA23RUN
9H	
AH	TA2REG
BH	TA3REG
CH	TA23MOD
DH	TA3FFCR
EH	
FH	

[8] UART/SIO

Address	Name
0200H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	
7H	SIRCR
8H	SC1BUF
9H	SC1CR
AH	SC1MOD0
BH	BR1CR
CH	BR1ADD
DH	SC1MOD1
EH	
FH	

[9] I^{2}C bus/SIO

Address	Name
0240H	SBI0CR1
1H	SBI0DBR
2H	I2C0AR
3H	SBI0CR2/SBI0SR
4H	SBI0BR0
5H	SBI0BR1
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[10] 10-bit ADC

Address	Name
02A0H	ADREG04L
1H	ADREG04H
2H	ADREG15L
3H	ADREG15H
4H	ADREG26L
5H	ADREG26H
6H	ADREG37L
7H	ADREG37H
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Name
02B0H	ADMOD0
1H	ADMOD1
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses, e.g., addresses to which no register has been allocated.

Not Recommended for New Design

Downloaded from Arrow.com.

Table 5.3 Address Map SFRs

[11] WDT

Address	Name
0300H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[12] RTC

Address	Name
0320H	SECR
1H	MINR
2H	HOURR
3H	DAYR
4H	DATER
5H	MONTHR
6H	YEARR
7H	PAGER
8H	RESTR
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[13] MLD

Address	Name
0330H	ALM
1H	MELALMC
2H	MELFL
3H	MELFH
4H	ALMINT
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[14] MMU

Address	Name
0350H	LOCAL0
1H	LOCAL1
2H	LOCAL2
3H	LOCAL3
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses, e.g., addresses to which no register has been allocated.

Not Recommended for New Design

Downloaded from Arrow.com.

(1) I/O ports

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1	Port 1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Data from external port (Output latch register cleared to "0".)							
P2	Port 2	06H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			1	1	1	1	1	1	1	1
P5	Port 5	0DH		P56	P55	P54				
				R/W						
				Data from external port (Output latch register is set to "1".)						
				0 (Output latch register) : Pull-up resistor OFF 1 (Output latch register) : Pull-up resistor ON						
P6	Port 6	12H	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			1	1	1	1	1	0	1	1
P7	Port 7	13H						P72	P71	P70
								R/W		
								Data from external port (Output latch register is set to "1".)		
								0 (Output latch register) : Pull-up resistor OFF 1 (Output latch register) : Pull-up resistor ON		–
P8	Port 8	18H	P87	P86	P85	P84	P83	P82	P81	P80
			R							
			Data from external port							
PB	Port B	22H		PB6	PB5	PB4	PB3	PB2	PB1	PB0
				R/W						
				Data from external port (Output latch register is set to "1".)						
PC	Port C	23H			PC5	PC4	PC3	PC2	PC1	PC0
					R/W					
					Data from external port (Output latch register is set to "1".)					
PD	Port D	29H	PD7	PD6	PD5					
			R/W							
			1	1	1					
PZ	Port Z	7DH					PZ3	PZ2		RDE
							R/W			R/W
							Data from external port (Output latch register is set to "1".)			1
							0 (Output latch register) : Pull-up resistor OFF 1 (Output latch register) : Pull-up resistor ON			–

Not Recommended for New Design

Downloaded from Arrow.com.

(2) I/O port control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1CR	Port 1 control	04H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P2FC	Port 2 function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			1	1	1	1	1	1	1	1
			0: Port 1: Address bus (A23 to A16)							
P5CR	Port 5 control	0AH (Prohibit RMW)		P56C	P55C	P54C				
				W						
				0	0	0				
				0: Input 1: Output						
P5FC	Port 5 function	0BH (Prohibit RMW)			P55F	P54F				
					W					
					0	0				
					0: Port 1: $\overline{BUSAK}$	0: Port 1: $\overline{BUSRQ}$				
P6FC	Port 6 function	15H (Prohibit RMW)	–	–	P65F	P64F	P63F	P62F	P61F	P60F
			W	W	W					
			0	0	0	0	0	0	0	0
			Always write 0		0: Port 1: EA25	0: Port 1: EA24	0: Port 1: $\overline{CS3}$	0: Port 1: $\overline{CS2}$	0: Port 1: $\overline{CS1}$	0: Port 1: $\overline{CS0}$
P6FC2	Port 6 function 2	1BH (Prohibit RMW)	P67F2	P66F2	P65F2	P64F2	–	P62F2	–	–
			W				W	W	W	W
			0	0	0	0	0	0	0	0
			0: <P67F> 1: $\overline{CS2E}$	0: <P66F> 1: $\overline{CS2D}$	0: <P65F> 1: $\overline{CS2C}$	0: <P64F> 1: $\overline{CS2B}$	Always write 0	0: <P62F> 1: $\overline{CS2A}$	Always write 0	
P7CR	Port 7 control	16H (Prohibit RMW)						P72C	P71C	P70C
								W		
								0	0	0
								0: Input 1: Output		
P7FC	Port 7 function	17H (Prohibit RMW)						P72F	P71F	P70F
								W		
								0	0	0
								0: Port 1: SCL	0: Port 1: SDA/SO	0: Port 1: SCK
P7FC2	Port 7 function 2	1CH (Prohibit RMW)						–	P71F2	P70F2
								W		
								0	0	0
								Always write 0	0: <P71F> 1: OPTTX0	PIN SELECT 0: RXD0 (PC1) 1: PTRX0 (P70)

Not Recommended for New Design

Downloaded from Arrow.com.

(2) I/O port control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P7ODE	Port 7 open drain	1FH (Prohibit RMW)						ODEP72	ODEP71	
								W		
								0	0	
								0: 3 states 1: Open drain		
PBCR	Port B control	24H (Prohibit RMW)		PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
				W						
				0	0	0	0	0	0	0
				0: Input 1: Output						
PBFC	Port B function	25H (Prohibit RMW)		PB6F	PB5F	PB4F	PB3F	PB2F	PB1F	
				W	W	W	W	W	W	
				0	0	0	0	0	0	
				0: Port 1: INT3	0: Port 1: INT2	0: Port 1: INT1	0: Port 1: INT0	0: Port 1: TA3OUT	0: Port 1: TA1OUT	
PCCR	Port C control	26H (Prohibit RMW)			PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
					W					
					0	0	0	0	0	0
					0: Input 1: Output					
PCFC	Port C function	27H (Prohibit RMW)			PC5F		PC3F	PC2F		PC0F
					W		W	W		W
					0		0	0		0
					0: Port 1: SCLK1		0: Port 1: TXD1	0: Port 1: SCLK0		0: Port 1: TXD0
PCODE	Port C open drain	28H (Prohibit RMW)					ODEPC3			ODEPC0
							W			W
							0			0
							0: CMOS 1: Open drain			0: CMOS 1: Open drain
PDFC	Port D function	2AH (Prohibit RMW)	PD7F	PD6F	PD5F					
			W	W	W					
			0	0	0					
			0: Port 1: MLDALM	0: Port 1: $\overline{\text{ALARM}}$ @<PD6>=1 $\overline{\text{MLDALM}}$ @<PD6>=0	0: Port 1: SCOUT					
PZCR	Port Z control	7EH (Prohibit RMW)					PZ3C	PZ2C		
							W			
							0	0		
							0: Input 1: Output			
PZFC	Port Z function	7FH (Prohibit RMW)					PZ3F	PZ2F		
							W			
							0	0		
							0: Port 1: R/$\overline{\text{W}}$	0: Port 1: $\overline{\text{HWR}}$		

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Interrupt control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 and INTAD enable	90H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTAD	Interrupt level			1: INT0	Interrupt level		
INTE12	INT1 and INT2 enable	91H	INT2				INT1			
			I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INT2	Interrupt level			1: INT1	Interrupt level		
INTE3ALM4	INT3 and INTALM 4 enable	92H	INTALM4				INT3			
			IA4C	IA4M2	IA4M1	IA4M0	I3C	I3M2	I3M1	I3M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTALM4	Interrupt level			1: INT3	Interrupt level		
INTEALM01	INTALM 0 and INTALM 1 enable	93H	INTALM1				INTALM0			
			IA1C	IA1M2	IA1M1	IA1M0	IA0C	IA0M2	IA0M1	IA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTALM1	Interrupt level			1:INTALM0	Interrupt level		
INTEALM23	INTALM2 and INTALM3 enable	94H	INTALM3				INTALM2			
			IA3C	IA3M2	IA3M1	IA3M0	IA2C	IA2M2	IA2M1	IA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTALM3	Interrupt level			1:INTALM2	Interrupt level		
INTETA01	INTTA0 and INTTA1 enable	95H	INTTA1 (TMRA1)				INTTA0 (TMRA0)			
			ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTA1	Interrupt level			1: INTTA0	Interrupt level		
INTETA23	INTTA2 and INTTA3 enable	96H	INTTA3 (TMRA3)				INTTA2 (TMRA2)			
			ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTA3	Interrupt level			1: INTTA2	Interrupt level		
INTERTC	INTRTC enable	97H	–				INTRTC			
			–	–	–	–	IRC	IRM2	IRM1	IRM0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”				1: INTRTC	Interrupt level		

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Interrupt control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTES0	INTTX0 and INTTRX0 enable	98H	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTX0	Interrupt level			1: INTRX0	Interrupt level		
INTES1	INTTX1 and INTTRX1 enable	99H	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTX1	Interrupt level			1: INTRX1	Interrupt level		
INTES2	INTESBI enable	9AH	–				INTSBI			
			–	–	–	–	ISBIC	ISBIM2	ISBIM1	ISBIM0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”				1: INTSBI	Interrupt level		
INTETC01	INTTC0 and INTTC1 enable	9BH	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC23	INTTC2 and INTTC3 enable	9CH	INTTC3				INTTC2			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEP01	INTP0 and INTP1 enable	9DH	INTP1				INTP0			
			IP1C	IP1M2	IP1M1	IP1M0	IP0C	IP0M2	IP0M1	IP0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Not Recommended for New Design

Downloaded from Arrow.com.

(3) Interrupt control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA 0 request vector	80H			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
					DMA0 start vector					
DMA1V	DMA 1 request vector	81H			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
					DMA1 start vector					
DMA2V	DMA 2 request vector	82H			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
					DMA2 start vector					
DMA3V	DMA 3 request vector	83H			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
					0	0	0	0	0	0
					DMA3 start vector					
INTCLR	Interrupt clear control	88H (Prohibit RMW)			CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
					W					
					0	0	0	0	0	0
					Clears interrupt request flag by writing to DMA start vector					
DMAR	DMA software request register	89H (Prohibit RMW)					DMAR3	DMAR2	DMAR1	DMAR0
							R/W	R/W	R/W	R/W
							0	0	0	0
							1: DMA request in software			
DMAB	DMA burst request register	8AH					DMAB3	DMAB2	DMAB1	DMAB0
							R/W	R/W	R/W	R/W
							0	0	0	0
							1 : DMA request on burst mode			
IIMC	Interrupt input mode control	8CH (Prohibit RMW)	–	–	I3EDGE	I2EDGE	I1EDGE	I0EDGE	I0LE	NMIREE
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Always write 0	Always write 0	INT3 edge 0: Rising 1: Falling	INT2 edge 0: Rising 1: Falling	INT1 edge 0: Rising 1: Falling	INT0 edge 0: Rising 1: Falling	INT0 0: Edge 1: Level	1: Operation even on $\overline{\text{NMI}}$ rising edge

Not Recommended for New Design

Downloaded from Arrow.com.

(4) Chip select/wait control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT control register	C0H (Prohibit RMW)	B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: 10: } Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
B1CS	Block 1 CS/WAIT control register	C1H (Prohibit RMW)	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: 10: } Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
B2CS	Block 2 CS/WAIT control register	C2H (Prohibit RMW)	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
			W	W	W	W	W	W	W	W
			1	0	0	0	0	0	0	0
			0: Disable 1: Enable	0: 16 M area 1: Area set	00: ROM/SRAM 01: 10: } Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
B3CS	Block 3 CS/WAIT control register	C3H (Prohibit RMW)	B3E		B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: 10: } Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
BEXCS	External CS/WAIT control register	C7H (Prohibit RMW)					BEXBUS	BEXW2	BEXW1	BEXW0
							W	W	W	W
							0	0	0	0
							Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
MSAR0	Memory start address register 0	C8H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR0	Memory address mask register 0	C9H	V20	V19	V18	V17	V16	V15	V14 to V9	V8
			R/W							
			1	1	1	1	1	1	1	1
			CS0 area size 0: enable to address comparison							
MSAR1	Memory start address register 1	CAH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR1	Memory address mask register 1	CBH	V21	V20	V19	V18	V17	V16	V15 to V9	V8
			R/W							
			1	1	1	1	1	1	1	
			CS1 area size 0: Enable to address comparison							

Not Recommended for New Design

Downloaded from Arrow.com.

(4) Chip select/wait control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
MSAR2	Memory start address register 2	CCH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR2	Memory address mask register 2	CDH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS2 area size 0: Enable to address comparison							
MSAR3	Memory start address register 3	CEH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR3	Memory address mask register 3	CFH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS3 area size 0: Enable to address comparison							

Not Recommended for New Design

Downloaded from Arrow.com.

(5) Clock gear (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SYSCR0	System clock control register 0	E0H	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
			R/W							
			1	1	1	0	0	0	0	0
			High-frequency oscillator (fc) 0: Stopped 1: Oscillation	Low-frequency oscillator (fs) 0: Stopped 1: Oscillation	High-frequency oscillator (fc) after release of STOP mode 0: Stopped 1: Oscillation	Low-frequency oscillator (fs) after release of STOP mode 0: Stopped 1: Oscillation	Select clock after release of STOP mode 0: fc 1: fs	Warm-up timer 0 write: Don't care 1 write: Start timer 0 read: End warm-up 1 read: Not end warm-up	Select prescaler clock 00: f_{FPH} 01: Reserved 10: fc/16 11: Reserved	
SYSCR1	System clock control register 1	E1H					SYSCK	GEAR2	GEAR1	GEAR0
							R/W			
							0	1	0	0
							System clock selection 0: fc 1: fs (Note 2)	High-frequency gear value selection (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (Reserved) 110: (Reserved) 111: (Reserved)		
SYSCR2	System clock control register 2	E2H		SCOSEL	WUPTM1	WUPTM0	HALTM1	HALTM0	SELDRV	DRVE
				R/W	R/W	R/W	R/W	R/W	R/W	R/W
				0	1	0	1	1	0	0
				0: fs 1: f_{FPH}	Warm-up time 00: Reserved 01: 2^8 input frequency 10: 2^{14} 11: 2^{16}		00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		<Drive> mode select 0: STOP 1: IDLE	1: Drive IDLE1 mode

Not Recommended for New Design

Downloaded from Arrow.com.

(5) Clock gear (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
EMCCR0	EMC control register 0	E3H	PROTECT	–	–	–	–	EXTIN	DRVOSCH	DRVOSCL
			R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	1	1	0	0	0	1	1
			Protection flag 0: OFF 1: ON	Always write 0	Always write 1	Always write 0	Always write 0	1: External	fc oscillator drivability 1: Normal 0: Weak	fs oscillator driver ability 1: Normal 0: Weak
EMCCR1	EMC control register 1	E4H	Switching the protect ON/OFF by write to following : 1st-KEY and 2nd-KEY 1st-KEY: EMCCR1 = 5AH, EMCCR2 = A5H in succession write 2nd-KEY: EMCCR1 = A5H, EMCCR2 = 5AH in succession write							
EMCCR2	EMC control register 2	E5H								
EMCCR3	EMC control register 3	E6H		ENFROM	ENDROM	ENPROM		FFLAG	DFLAG	PFLAG
				R/W	R/W	R/W		R/W	R/W	R/W
				0	0	0		0	0	0
				CS1A area detect enable 0: Disable 1: Enable	CS2B-2G area detect enable 0: Disable 1: Enable	CS2A area detect enable 0: Disable 1: Enable		CS1A Write operation flag	CS2B-2G Write operation flag	CS2A Write operation flag
								When read mode 0: No write 1: Write When write mode 0: Flag area		

Not Recommended for New Design

Downloaded from Arrow.com.

(6) DFM (Clock doubler)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DFMCR0	DFM control register 0	E8H	ACT1	ACT0	DLUPFG	DLUPTM				
			R/W	R/W	R	R/W				
			0	0	0	0				
			(ACT1, ACT0): DFM / LUP / f_{FPH} 00: STOP / STOP / f_{OSCH} 01: RUN / RUN / f_{OSCH} 10: RUN / STOP / f_{DFM} 11: RUN / STOP / f_{OSCH}		Lockup flag 0: End LUP 1: Do not	Lockup time 0: $2^{12}/f_{OSCH}$ 1: $2^{10}/f_{OSCH}$				
DFMCR1	DFM control register 1	E9H	D7	D6	D5	D4	D3	D2	D1	D0
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	1	0	0	1	1
			DFM correction Input frequency 4 to 8.25 MHz (at 2.7 to 3.6 V): Write 0BH Input frequency 2 to 2.5 MHz (at 2.0V ± 10%): Write 1BH							

Not Recommended for New Design

Downloaded from Arrow.com.

(7) 8-bit timer

(7–1) TMRA01

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA01RUN	8-bit timer RUN register	100H	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		
TA0REG	8-bit timer register 0	102H (Prohibit RMW)	–							
			W							
			Undefined							
TA1REG	8-bit timer register 1	103H (Prohibit RMW)	–							
			W							
			Undefined							
TA01MOD	8-bit timer source CLK & mode	104H	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		00: Reserved 01: 2^6 PWM cycle 10: 2^7 11: 2^8		00: TA0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TA0IN pin 01: ϕT1 10: ϕT4 11: ϕT16	
TA1FFCR	8-bit timer flip-flop control	105H (Prohibit RMW)					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA1FF 01: Set TA1FF 10: Clear TA1FF 11: Don't care		1: TA1FF enable	0: TMRA0 1: TMRA1 inversion

(7–2) TMRA23

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA23RUN	8-bit timer RUN register	108H	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		
TA2REG	8-bit timer register 0	10AH (Prohibit RMW)	–							
			W							
			Undefined							
TA3REG	8-bit timer register 1	10BH (Prohibit RMW)	–							
			W							
			Undefined							
TA23MOD	8-bit timer source CLK & mode	10CH	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		00: Reserved 01: 2^6 PWM cycle 10: 2^7 11: 2^8		00: TA2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	
TA3FFCR	8-bit timer flip-flop control	10DH (Prohibit RMW)					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA3FF 01: Set TA3FF 10: Clear TA3FF 11: Don't care		1: TA3FF invert enable	0: TMRA2 1: TMRA3 inversion

Not Recommended for New Design

Downloaded from Arrow.com.

(8) UART/serial channel (1/2)

(8-1) UART/SIO channel 0

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial channel 0 buffer	200H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC0CR	Serial channel 0 control	201H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity Enable	1: Error Overrun	Parity	Framing	0: SCLK0↑ 1: SCLK0↓	1: Input SCLK0
SC0MOD0	Serial channel 0 mode0	202H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit8	1: CTS enable	1: Receive enable	1: Wakeup enable	00: I/O interface 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TA0TRG 01: Baud rate 10: Internal clock f_{SYS} 11: External clock	
BR0CR	Baud rate control	203H	–	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W							
			0	0	0		0	0	0	0
			Always write 0	1: (16 – K)/16 divided	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting the divided frequency "N" (0 to F)			
BR0ADD	Serial channel 0 K setting register	204H					BR0K3	BR0K2	BR0K1	BR0K0
							R/W			
							0	0	0	0
							Sets the frequency divisor "K" (Divided by N + (16 – K)/16)			
SC0MOD1	Serial channel 0 mode1	205H	I2S0	FDPX0						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	Duplex 0: Half 1: Full						

(8-2) IrDA

Symbol	Name	Address	7	6	5	4	3	2	1	0
SIRCR	IrDA control register	207H	PLSEL	RXSEL	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0
			R/W	R/W	R/W	R/W	R/W			
			0	0	0	0	0	0	0	0
			Transmission pulse width 0: 3/16 1: 1/16	Receiving data 0: H pulse 1: L pulse	Transmission 0: Disable 1: Enable	Receiving 0: Disable 1: Enable	Set the effective SIRRxD pulse width Pulse width more than 2x × (Set value + 1) + 100ns Possible: 1 to 14 Not possible: 0, 15			

Not Recommended for New Design

Downloaded from Arrow.com.

(8) UART/serial channel (2/2)

(8-3) UART/SIO channel1

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC1BUF	Serial channel 1 buffer	208H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC1CR	Serial channel 1 control	209H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity Enable	1: Error Overrun	Parity	Framing	0: SCLK1↑ 1: SCLK1↓	1: Input
SC1MOD0	Serial channel 1 mode	20AH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit8	1: CTS enable	1: Receive enable	1: Wakeup	00: I/O interface 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TA0TRG 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock SCLK1	
BR1CR	Baud rate control	20BH	–	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write 0	1: (16 – K)/16 divided enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting the divided frequency “N” (0 to F)			
BR1ADD	Serial channel 1 K setting register	20CH					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Sets the frequency divisor “K” (Divided by N + (16 – K)/16)			
SC1MOD1	Serial channel 1 mode1	20DH	I2S1	FDPX1						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	Duplex 0: Half 1: Full						

Not Recommended for New Design

Downloaded from Arrow.com.

(9) I²C bus/serial interface

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI0CR1	Serial bus interface control register 1	240H (I^2C bus mode) (Prohibit RMW)	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
			W			R/W		W	W	R/W
			0	0	0	0		0	0	0/1
			Number of transfer bits 000: 8, 001: 1, 010: 2 011: 3, 100: 4, 101: 5 110: 6, 111: 7			Acknowledge mode 0: Disable 1: Enable		Setting for the devisor value n 000: 4, 001: 5, 010: 6 011: 7, 100: 8, 101: 9 110: 10, 111: (Reserved)		
		240H (SIO mode) (Prohibit RMW)	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			Transfer 0: Stop 1: Start	Transfer 0: Continue 1: Abort	Transfer mode 00: 8-bit transmit mode 01: Reserved 10: 8-bit transmit/receive 11: 8-bit received mode			Setting for the divisor value n 000: 3, 001: 4, 010: 5 011: 6, 100: 7, 101: 8 110: 9, 111: SCK pin		
SBI0DBR	SBI buffer register	241H (Prohibit RMW)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
			R (Receiving)/W (Transmission)							
			Undefined							
I2C0AR	I^2C bus address register	242H (Prohibit RMW)	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Setting slave address							Address recognition 0: Enable 1: Disable
When read SBI0SR	Serial bus interface status register	243H (I^2C bus mode) (Prohibit RMW)	MST	TRX	BB	PIN	AL/SBIM1	AAS/SBIM0	AD0/ SWRST	LRB/ SWRST0
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	1	0	0	0	0
			0: Slave 1: Master	0: Receiver 1: Transmit	Bus status monitor 0: Free 1: Busy	INTSBI request monitor 0: Request 1: Cancel	Arbitration lost detection monitor 1: Detect	Slave address match detection monitor 1: Detect	GENERAL CALL detection monitor 1: Detect	Lost receive bit monitor 0: 0 1: 1
When write SBI0CR2	Serial bus interface control register 2				Start/stop condition generation 0: Start condition 1: Stop condition		SBI operating mode selection 00: Port mode 01: SIO mode 10: I2C bus mode 11: (Reserved)		Software reset generate write 10 and 01, then an internal reset signal is generated.	
When read SBI0SR	Serial bus interface status register	243H (SIO mode) (Prohibit RMW)					SIOF/SBIM1	SEF/SBIM2	–	–
							R/W	R/W	W	W
							0	0	0	0
							Transfer status monitor 0: Stopped 1: Terminated in process	Shift operation status monitor 0: Stopped 1: Terminated in process		
When write SBI0CR2	Serial bus interface control register 2						Serial bus interface operating mode selection 00: Port mode 01: SIO mode 10: I^2C bus mode 11: (Reserved)		Always write 0	Always write 0
SBI0BR0	Serial bus interface baud rate register 0	244H (Prohibit RMW)	–	I2SBI0						
			W	R/W						
			0	0						
			Always write 0	IDLE2 0: Abort 1: Operate						
SBI0BR1	Serial bus interface baud rate register 1	245H (Prohibit RMW)	P4EN	–						
			W	W						
			0	0						
			Internal Clock 0: Abort 1: Operate	Always write 0						

Not Recommended for New Design

Downloaded from Arrow.com.

(10) AD converter

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD0	AD MODE register 0	2B0H	EOCF	ADBF	–	–	ITM0	REPEAT	SCAN	ADS
			R		R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			AD conversion end flag 1: End	AD conversion burst flag 1: Busy	Always write 0	Always write 0	Interrupt in repeat mode	Repeat mode specification 1: Repeat	Scan mode specification 1: Scan	AD conversion Star 1: Start
ADMOD1	AD MODE register 1	2B1H	VREFON	I2AD			ADTRGE	ADCH2	ADCH1	ADCH0
			R/W	R/W			R/W	R/W	R/W	R/W
			0	0			0	0	0	0
			VREF control 1: VREF on	IDLE2 0: Abort 1: Operate			AD control 1: Enable for	Input channel 000: AN0 AN0 001: AN1 AN0 → AN1 010: AN2 AN0 → AN1 → AN2 011: AN3 AN0 → AN1 → AN2 → AN3 100: AN4 AN4 101: AN5 AN4 → AN5 110: AN6 AN4 → AN5 → AN6 111: AN7 AN4 → AN5 → AN6 → AN7		
ADREG04L	AD result register 0/4 low	2A0H	ADR01	ADR00						ADR0RF
			R							R
			Undefined							0
ADREG04H	AD result register 0/4 high	2A1H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
ADREG15L	AD result register 1/5 low	2A2H	ADR11	ADR10						ADR1RF
			R							R
			Undefined							0
ADREG15H	AD result register 1/5 high	2A3H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
ADREG26L	AD result register 2/6 low	2A4H	ADR21	ADR20						ADR2RF
			R							R
			Undefined							0
ADREG26H	AD result register 2/6 high	2A5H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
ADREG37L	AD result register 3/7 low	2A6H	ADR31	ADR30						ADR3RF
			R							R
			Undefined							0
ADREG37H	AD result register 3/7 high	2A7H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

Not Recommended for New Design

Downloaded from Arrow.com.

(11) Watchdog timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	WDT mode register	300H	WDTE	WDTP1	WDTP0			I2WDT	RESCR	–
			R/W	R/W	R/W			R/W	R/W	R/W
			1	0	0			0	0	0
			1: WDT enable	00: 2^{15}/f_{SYS} 01: 2^{17}/f_{SYS} 10: 2^{19}/f_{SYS} 11: 2^{21}/f_{SYS}				IDLE2 0: Abort 1: Operate	1: RESET	Always write 0
WDCR	WDT control	301H (Prohibit RMW)	–							
			W							
			–							
			B1H: WDT disable 4EH: WDT clear							

Not Recommended for New Design

Downloaded from Arrow.com.

(12) RTC (Real time clock)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SECR	Second register	320H		SE6	SE5	SE4	SE3	SE2	SE1	SE0
				R/W						
				Undefined						
			0 is read	40 s	20 s	10 s	8 s	4 s	2 s	1 s
MINR	Minute register	321H		MI6	MI5	MI4	MI3	MI2	MI1	MI0
				R/W						
				Undefined						
			0 is read	40 min	20 min	10 min	8 min	4 min	2 min	1min
HOURR	Hour register	322H			HO5	HO4	HO3	HO2	HO1	HO0
					R/W					
					Undefined					
			0 is read		20 H (PM/AM)	10 H	8 H	4 H	2 H	1 H
DAYR	Day register	323H						WE2	WE1	WE0
								R/W		
								Undefined		
			0 is read					W2	W1	W0
DATER	Date register	324H			DA5	DA4	DA3	DA2	DA1	DA0
					R/W					
					Undefined					
			0 is read		20 days	10 days	8 days	4 days	2 days	1 day
MONTHR	Month register	325H				MO4	MO3	MO2	MO1	MO0
						R/W				
						Undefined				
		Page 0	0 is read			10 month	8 month	4 month	2 month	1 month
		Page 1	0 is read							0: Indicator for 1: Indicator for
YEARR	Year register	326H	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
			R/W							
			Undefined							
		Page 0	80 years	40 years	20 years	10 years	8 years	4 years	2 years	1 year
		Page 1	0 is read						Leap year setting	
PAGER	Page register	327H (Prohibit RMW)	INTENA			Adjust	ENATMR	ENAALM		PAGE
			R/W			W	R/W			R/W
			0			Undefined	Undefined			Undefined
			INTRTC 0: Disable 1: Enable	0 is read		0:Don't care 1: Adjust	Clock 0: Disable 1: Enable	Alarm 0: Disable 1: Enable	0 is read	select PAGE
RESTR	Reset register	328H (Prohibit RMW)	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	RE3	RE2	RE1	RE0
			W							
			Undefined							
			1 Hz 0: Enable 1: Disable	16 Hz 0: Enable 1: Disable	1: Clock reset	1:Alarm reset	Always write 0			

Not Recommended for New Design

Downloaded from Arrow.com.

(13) Melody/alarm generator

Symbol	Name	Address	7	6	5	4	3	2	1	0
ALM	Alarm pattern register	330H	AL8	AL7	AL6	AL5	AL4	AL3	AL2	AL1
			R/W							
			0	0	0	0	0	0	0	0
			Alarm pattern set							
MELALMC	Melody/ alarm control register	331H	FC1	FC0	ALMINV	–	–	–	–	MELALM
			R/W		R/W	R/W	R/W	R/W	R/W	R/W
			0		0	0	0	0	0	0
			Free-run counter control 00: Hold 01: Restart 10: Clear 11: Clear and start		Alarm frequency invert 1: Invert	Always write 0				Output frequency 0: Alarm 1: Melody
MELFL	Melody frequency register-L	332H	ML7	ML6	ML5	ML4	ML3	ML2	ML1	ML0
			R/W							
			0	0	0	0	0	0	0	0
			Melody frequency set (Low 8 bits)							
MELFH	Melody frequency register-H	333H	MELON				ML11	ML10	ML9	ML8
			R/W				R/W			
			0				0	0	0	0
			Melody counter control 0: Stop and clear 1: Start				Melody frequency set (High 4 bits)			
ALMINT	Alarm interrupt enable register	334H			–	IALM4E	IALM3E	IALM2E	IALM1E	IALM0E
					R/W	R/W				
					0	0	0	0	0	0
					Always write 0	INTALM4 to INTALM0 alarm interrupt enable				

Not Recommended for New Design

Downloaded from Arrow.com.

(14) MMU

Symbol	Name	Address	7	6	5	4	3	2	1	0
LOCAL0	LOCAL0 control register	350H	L0E					L0EA22	L0EA21	L0EA20
			R/W					R/W		
			0					0	0	0
			BANK for LOCAL0 0: Disable 1: Enable					LOCAL0 area BANK set "000" setting is prohibited because it pretend COMMON 0 area		
LOCAL1	LOCAL1 control register	351H	L1E					L1EA23	L1EA22	L1EA21
			R/W					R/W		
			0					0	0	0
			BANK for LOCAL1 0: Disable 1: Enable					LOCAL1 area ANK set "001" setting is prohibited because it pretend COMMON 0 area		
LOCAL2	LOCAL2 control register	352H	L2E					L2EA23	L2EA22	L2EA21
			R/W					R/W		
			0					0	0	0
			BANK for LOCAL2 0: Disable 1: Enable					LOCAL2 area BANK set "111" setting is prohibited because it pretend COMMON 0 area		
LOCAL3	LOCAL3 control register	353H	L3E			L3EA26	L3EA25	L3EA24	L3EA23	L3EA22
			R/W			R/W				
			0			0	0	0	0	0
			BANK for LOCAL3 0: Disable 1: Enable			01000 to 01011: $\overline{CS2D}$ 00000 to 00011: $\overline{CS2B}$ 00100 to 00111: $\overline{CS2C}$ 01100 to 01111: $\overline{CS2E}$ 10000 to 11111: Set prohibition				

Not Recommended for New Design

Downloaded from Arrow.com.

6. Points of Note and Restrictions

(1) Notation

a. The notation for built-in I/O registers is as follows register symbol <Bit symbol>

e.g.) TA01RUN<TA0RUN> denotes bit TA0RUN of register TA01RUN.

b. Read-modify-write instructions

An instruction in which the CPU reads data from memory and writes the data to the same memory location in one instruction.

Example 1: SET 3, (TA01RUN) ... Set bit3 of TA01RUN.

Example 2: INC 1, (100H) ... Increment the data at 100H.

- Examples of read-modify-write instructions on the TLCS-900

Exchange instruction

EX (mem), R

Arithmetic operations

ADD	(mem), R/#	ADC	(mem), R/#
SUB	(mem), R/#	SBC	(mem), R/#
INC	#3, (mem)	DEC	#3, (mem)

Logic operations

AND	(mem), R/#	OR	(mem), R/#
XOR	(mem), R/#		

Bit manipulation operations

STCF	#3/A, (mem)	RES	#3, (mem)
SET	#3, (mem)	CHG	#3, (mem)
TSET	#3, (mem)		

Rotate and shift operations

RLC	(mem)	RRC	(mem)
RL	(mem)	RR	(mem)
SLA	(mem)	SRA	(mem)
SLL	(mem)	SRL	(mem)
RLD	(mem)	RRD	(mem)

c. fc, fs, f_{FPH}, f_{SYS} and one state

The clock frequency input on pins X1 and 2 is called f_{OSCH}. The clock selected by DFMCR0<ACT1:0> is called fc.

The clock selected by SYSCR1<SYSCK> is called f_{FPH}. The clock frequency give by f_{FPH} divided by 2 is called f_{SYS}.

One cycle of f_{SYS} is referred to as one state.

Not Recommended for New Design

Downloaded from Arrow.com.

(2) Points of note

a. AM0 and AM1 pins

This pin is connected to the VCC or the VSS pin. Do not alter the level when the pin is active.

b. EMU0 and EMU1

Open pins.

c. Reserved address areas

The TMP91C824 does not have any reserved areas.

d. HALT mode (IDLE1)

When IDLE1 mode is used (in which oscillator operation only occurs), set RTCCR <RTCRUN> to 0 stop the timer for the real-time clock before the HALT instructions is executed.

e. Warm-up counter

The warm-up counter operates when STOP mode is released, even if the system is using an external oscillator. As a result a time equivalent to the warm-up time elapses between input of the release request and output of the system clock.

f. Programmable pull-up resistance

The programmable pull-up resistor can be turned ON/OFF by a program when the ports are set for use as input ports. When the ports are set for use as output ports, they cannot be turned on/off by a program.

The data registers (e.g., Px) are used to turn the pull-up/pull-down resistors ON/OFF. Consequently read-modify-write instructions are prohibited.

g. Bus release function

It is described note point in 3.5 “Port Function” that pin’s conditions at bus release condition.

Please refer that.

h. Watchdog timer

The watchdog timer starts operation immediately after a reset is released. When the watchdog timer is not to be used, disable it.

When the bus is released, neither internal memory nor internal I/O can be accessed. However, the internal I/O continues to operate. Hence the watchdog timer continues to run. Therefore be careful about the bus releasing time and set the detection timer of watchdog timer.

i. AD converter

The string resistor between the VREFH and VREFL pins can be cut by a program so as to reduce power consumption. When STOP mode is used, disable the resistor using the program before the HALT instruction is executed.

j. CPU (Micro DMA)

Only the LDC cr, r and LDC r, cr instructions can be used to access the control registers in the CPU (e.g., the transfer source address register (DMASn)).

k. Undefined SFR

The value of an undefined bit in an SFR is undefined when read.

l. POP SR instruction

Please execute the POP SR instruction during DI condition.

Not Recommended for New Design

Downloaded from Arrow.com.

m. Releasing the HALT mode by requesting an interruption

Usually, interrupts can release all halts status. However, the interrupts ($\overline{NMI}$, INT0 to INT3, INTRTC, INTALM0 to INTALM4) which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficulty. The priority of this interrupt is compared with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

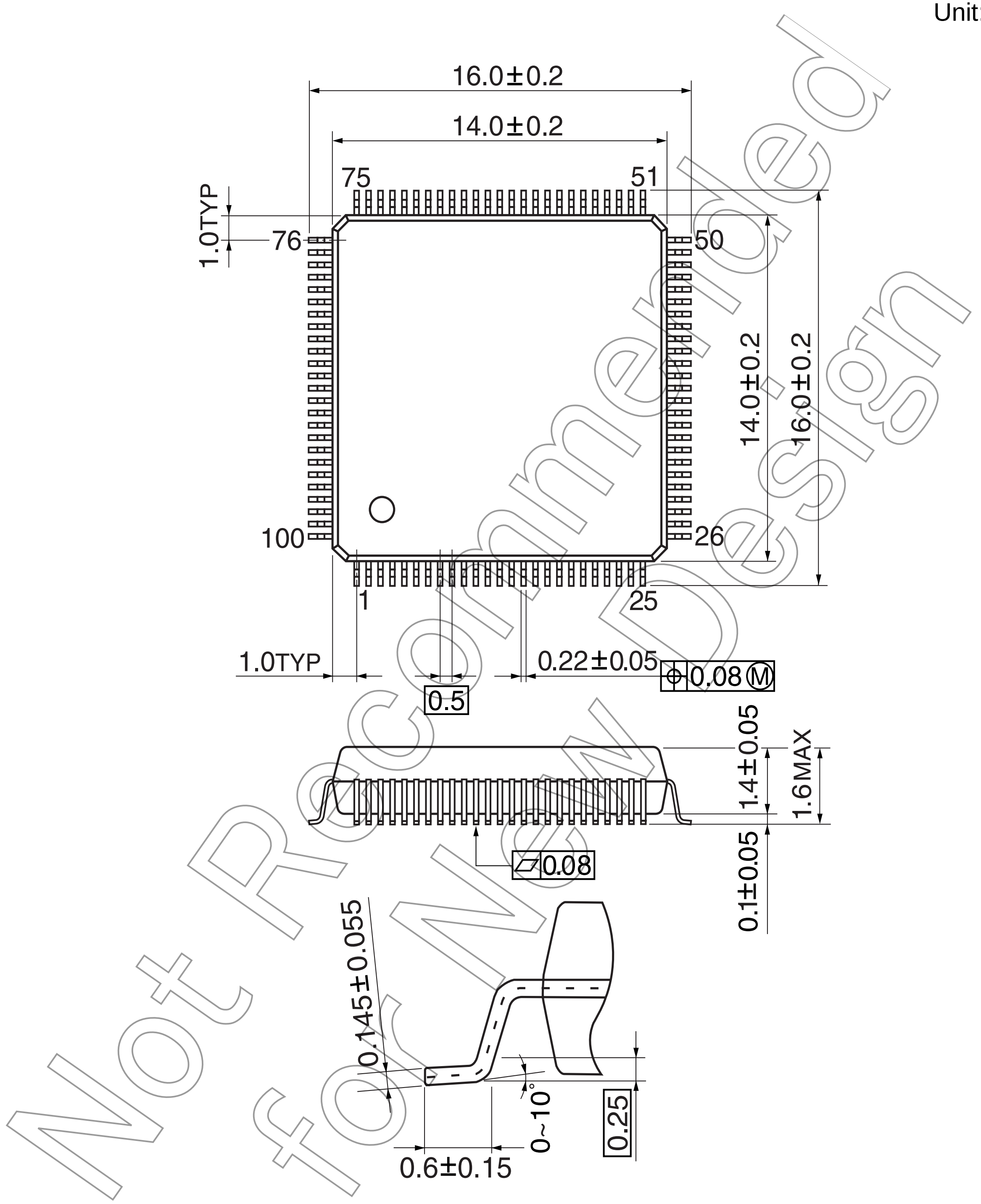
Not Recommended for New Design

Downloaded from Arrow.com.

7. Package Dimensions

LQFP100-P-1414-0.50F

Unit: mm



Downloaded from Arrow.com.