



4-BIT MICROCONTROLLER

1. GENERAL DESCRIPTION

The W742C800 is a high-performance 4-bit microcontroller (μC) that provides an LCD driver. The device contains a 4-bit ALU, two 8-bit timers, two dividers (for two oscillators) in dual-clock operation, a 40×8 LCD driver, two 4-bit I/O ports, and one channel PWM output for buzzer application. There are also six interrupt sources and 8-level subroutine nesting for interrupt applications. The W742C800 operates on very low current and has one power reduction modes, that is the dual-clock slow operation, which help to minimize power dissipation.

2. FEATURES

- Operating voltage: 2.4V - 3.8V
- Dual-clock operation
- Main oscillator
 - Crystal or RC oscillator can be selected by code option
 - Oscillation frequency range: 400 KHz to 4.19 MHz (Crystal)
: 2 MHz typically (RC)
- Sub-oscillator
 - Connect to 32768 Hz crystal only
- Memory
 - 8192 x 16 bit program ROM (including 32K x 4 bit look-up table)
 - 1024 x 4 bit data RAM (including 16 nibbles x 16 pages working registers)
 - 80 x 4 LCD data RAM
- 8 input/output pins
 - Input/output ports: 2 ports/8 pins
- Power-down mode
 - Hold function: no operation (excluding main oscillator and sub-oscillator)
 - Stop function: no operation (excluding sub-oscillator)
 - Dual-clock slow operation mode: system is operated by the sub-oscillator ($F_{\text{OSC}}=F_s$ and F_m is stopped)
- Six types of interrupts
 - Five internal interrupts (Divider0, Divider1, Timer 0, Timer 1, f_{sampling} : 6K/8KHz)
 - One external interrupts (RC Port)
- LCD driver output
 - 40 segments \times 8 commons & 1/8 duty 1/4 bias driving mode
 - SEG0 - SEG7 can be used as DC output by software controlling
 - Clock source should be the sub-oscillator clock

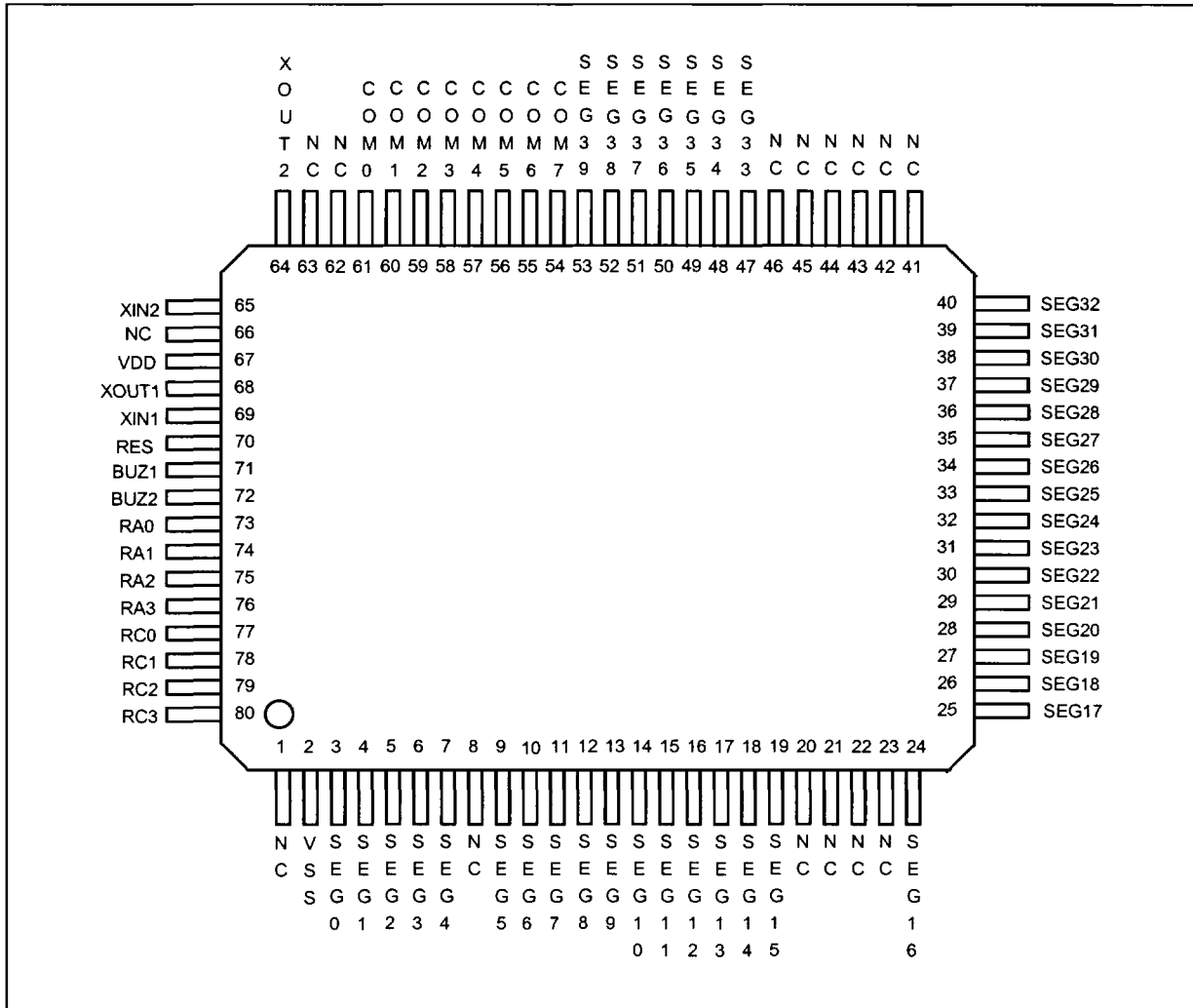


- BUZ1 & BUZ2 output pins
 - Output is software selectable as modulating or nonmodulating frequency
 - Works as frequency output specified by Timer 1
 - Pulse width modulation output pins for buzzer application
- Two built-in 14-bit frequency dividers
 - Divider0: the clock source is the output of the main oscillator
 - Divider1: the clock source is the output of the sub-oscillator
- Two built-in 8-bit programmable countdown timers
 - Timer 0: one of two internal clock frequencies ($F_{OSC}/4$ or $F_{OSC}/1024$) can be selected
 - Timer 1: with auto-reload function and one of three internal clock frequencies (F_{OSC} or $F_{OSC}/64$ or F_s) can be selected

; and it can output the programmable frequency signal through the BUZ1 pin
- Built-in 18/14-bit watchdog timer selectable for system reset
 - Enable/disable the watch dog timer is determined by code option
- Powerful instruction set: 131 instructions
- 8-level subroutine (include interrupt) nesting



3. PIN CONFIGURATION



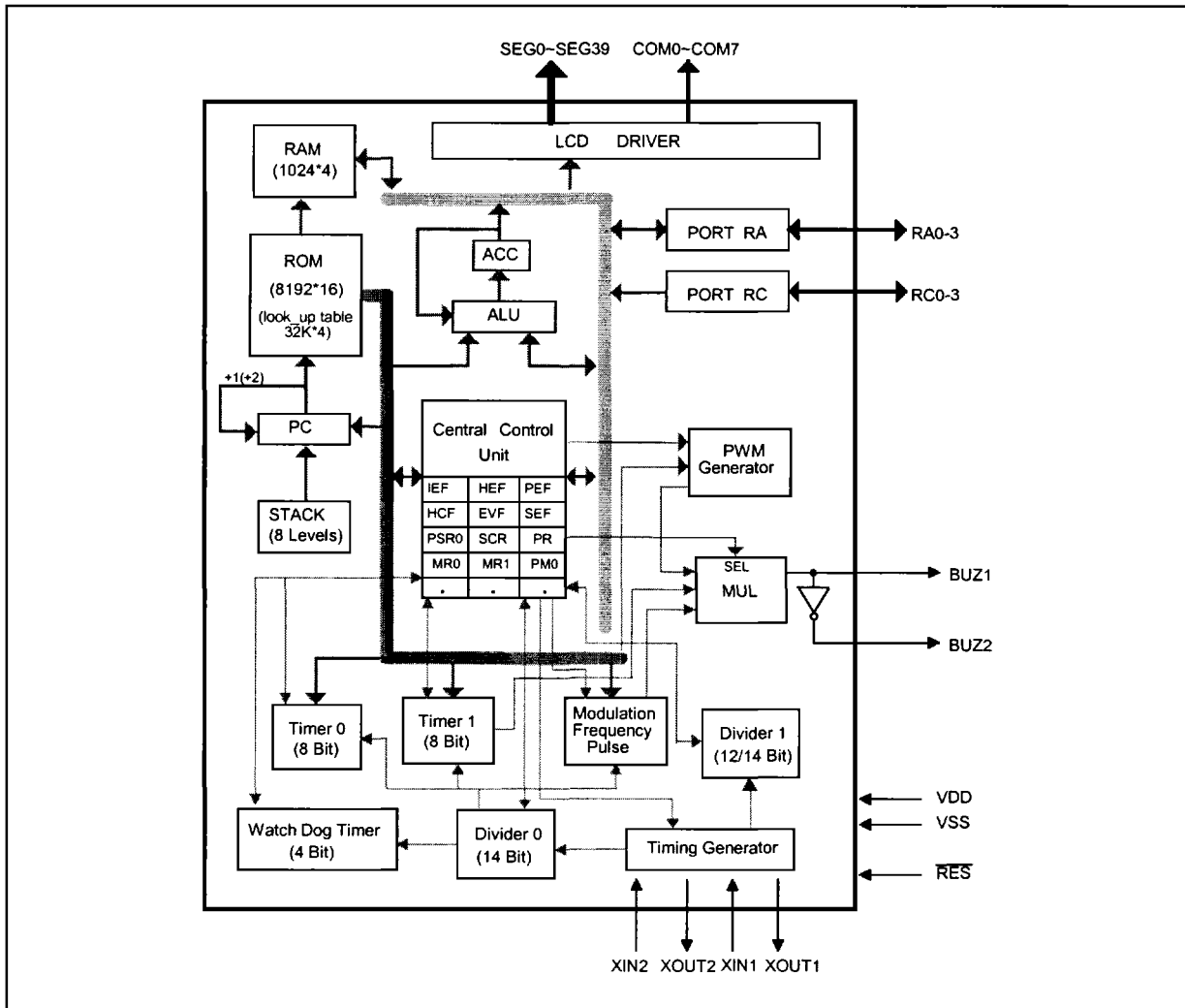


4. PIN DESCRIPTION

SYMBOL	I/O	FUNCTION
XIN2	I	Input pin for sub-oscillator. Connected to 32.768 Khz crystal only.
XOUT2	O	Output pin for sub-oscillator with internal oscillation capacitor. Connected to 32.768 Khz crystal only.
XIN1	I	Input pin for main-oscillator. Connected to resistor or crystal to generate system clock.
XOUT1	O	Output pin for main-oscillator. Connected to resistor or crystal to generate system clock.
RA0-RA3	I/O	Input/Output port. Input/output mode specified by port mode 1 register (PM1).
RC0-RC3	I/O	Input/Output port. Input/output mode specified by port mode 4 register (PM4). Each pin has an independent interrupt capability. And the built-in pull-high resistors can be enabled or not by port mode 0 register (PM0).
BUZ1 & BUZ2	O	Output pins only. These pins can output modulating or nonmodulating frequency, or PWM signal. The BUZCR register can be used to select which output signal is active.
RES	I	System reset pin with pull-high resistor.
SEG0-SEG39	O	LCD segment output pins. SEG0-SEG7 can be used as DC output port by code option.
COM0 - COM7	O	LCD common signal output pins.
VDD	I	Positive power supply (+).
VSS	I	Negative power supply (-).



5. BLOCK DIAGRAM





6. FUNCTIONAL DESCRIPTION

6.1 Program Counter (PC)

Organized as an 13-bit binary counter (PC0 to PC12), the program counter generates the addresses of the 8192×16 on-chip ROM containing the program instruction words. Before the jump or subroutine call instructions are to be executed, the destination ROM page must be determined firstly. The confirmation of the ROM page can be done by executing the MOV ROMPR, #I instruction. When the interrupt or initial reset conditions are to be executed, the corresponding address will be loaded into the program counter directly. The format used is shown below.

ITEM	ADDRESS	INTERRUPT PRIORITY
Initial Reset	0000H	-
INT 0 (Divider0)	0004H	1st
INT 1 (Timer 0)	0008H	2nd
INT 2 (Port RC)	000CH	3rd
INT 3 (fsampling)	0010H	4th
INT 4 (Divider1)	0014H	5th
INT 5 (Timer 1)	0020H	6th
JP Instruction	XXXXH	-
Subroutine Call	XXXXH	-

Table 1 Vector address and interrupt priority

6.2 Stack Register (STACK)

The stack register is organized as 13 bits x 8 levels (first-in, last-out). When either a call subroutine or an interrupt is executed, the program counter will be pushed onto the stack register automatically. At the end of a call subroutine or an interrupt service subroutine, the RTN instruction must be executed to pop the contents of the stack register into the program counter. When the stack register is pushed over the eighth level, the contents of the first level will be lost. In other words, the stack register is always eight levels deep.

6.3 Program Memory (ROM)

The read-only memory (ROM) is used to store program codes; and the look-up table is arranged as 32768×4 bits. The program ROM is divided into four pages; the size of each page is 2048×16 bits. So the total ROM size is 8192×16 bits. Before the jump or subroutine call instructions are to be executed, the destination ROM page must be determined firstly. The ROM page can be selected by executing the MOV ROMPR,#I instruction. But the branch decision instructions (e.g. JB0, SKB0, JZ, JC, ...) must jump to the same ROM page which the branch decision instructions are in. The whole ROM range can store both instruction codes and the look-up table. Each look-up table element is composed of 4 bits, so the look-up table can be addressed up to 32768 elements. Instruction MOVC R is used to read the look-up table content and transfer table data to the RAM. But before reading the addressed look-up table content, the content of the look-up table pointer (TAB) must be determined firstly. The address of the look-up table element is allocated by the content of TAB. The MOV TAB0 (TAB1, TAB2, TAB3), R instructions are used to allocate the address of the wanted look-up table element. The TAB0 register stores the LSB 4 bits of the look-up table address. The organization of the program memory is shown in Figure 6-1.

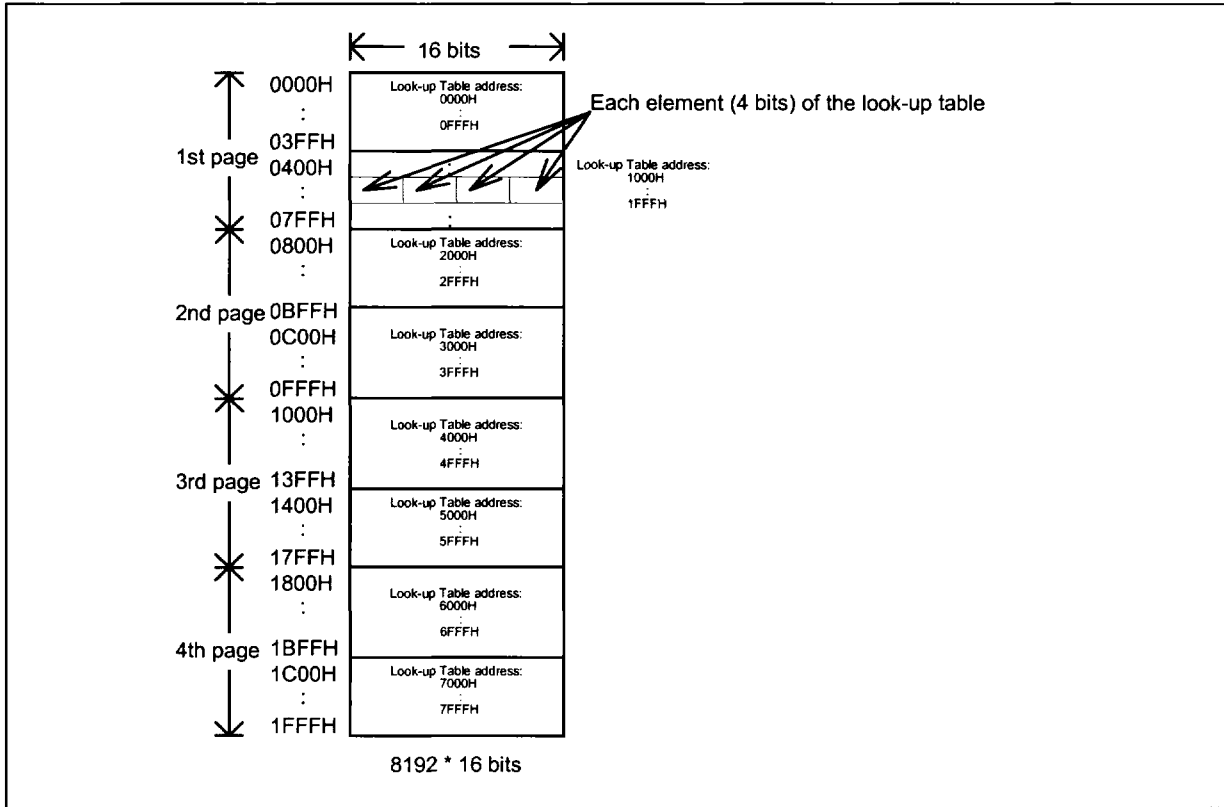
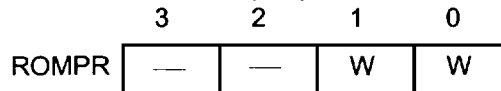


Figure 6-1 Program Memory Organization

6.3.1 ROM Page Register (ROMPR)

The ROM page register is organized as a 4-bit binary register. The bit descriptions are as follows:



Note: W means write only.

Bit 3 & Bit 2 is reserved.

Bit 1, Bit 0 ROM page preselect bits:

00 = ROM page 0 (0000H - 07FFH)

01 = ROM page 1 (0800H - 0FFFH)

10 = ROM page 2 (1000H - 17FFH)

11 = ROM page 3 (1800H - 1FFFH)



6.4 Data Memory (RAM)

6.4.1 Architecture

The static data memory (RAM) used to store data is arranged as 1024 × 4 bits. The data RAM is divided into eight banks; each bank has 128 × 4 bits. Executing the MOV DBKR,WR or MOV DBKR,#I instruction can determine which data bank is used. The data memory can be addressed directly or indirectly. But the data bank must be confirmed firstly; and the page in the data bank will be done in the indirect addressing mode, too. In indirect addressing mode, each data bank will be divided into eight

pages. Before the data memory is addressed indirectly, the page which the data memory is in must be confirmed. The organization of the data memory is shown in Figure 6-2.

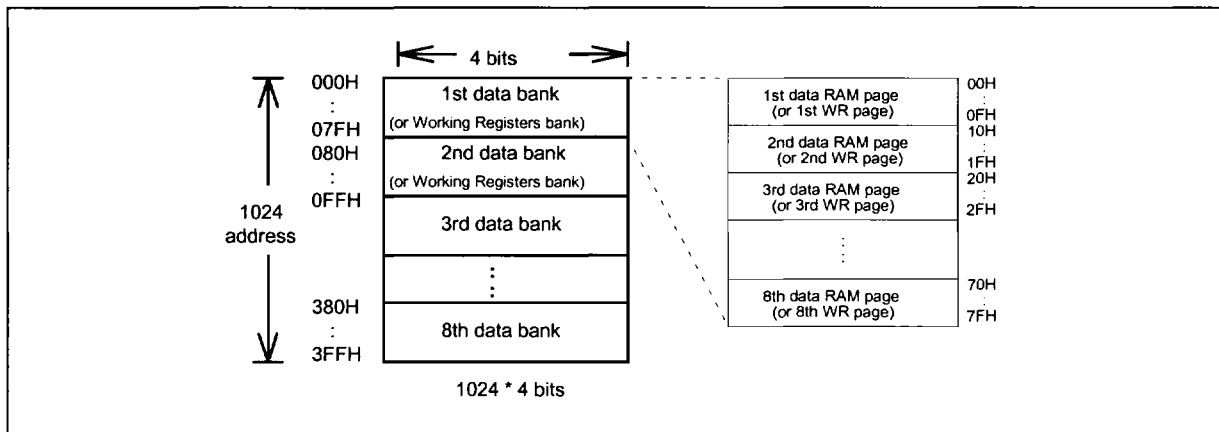
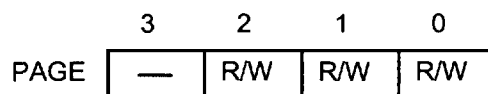


Figure 6-2 Data Memory Organization

The 1st and 2nd data bank (00H to 7FH & 80H to FFH) in the data memory can also be used as the working registers (WR). It is also divided into sixteen pages. Each page contains 16 working registers. When one page is used as WR, the others can be used as the normal data memory. The WR page can be switched by executing the MOV WRP,R or MOV WRP,#I instruction. The data memory cannot operate directly with immediate data, but the WR can do. The relationship between data memory locations and the page register (PAGE) in indirect addressing mode is described in the next sub-section.

6.4.2 Page Register (PAGE)

The page register is organized as a 4-bit binary register. The bit descriptions are as follows:



Note: R/W means read/write available.

Bit 3 is reserved.

Bit 2, Bit 1, Bit 0 Indirect addressing mode preselect bits in one data bank:

000 = Page 0 (00H - 0FH)

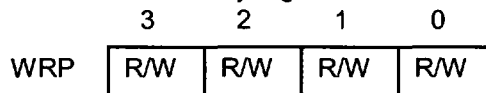
001 = Page 1 (10H - 1FH)



- 010 = Page 2 (20H - 2FH)
- 011 = Page 3 (30H - 3FH)
- 100 = Page 4 (40H - 4FH)
- 101 = Page 5 (50H - 5FH)
- 110 = Page 6 (60H - 6FH)
- 111 = Page 7 (70H - 7FH)

6.4.3 WR Page Register (WRP)

The WR page register is organized as a 4-bit binary register. The bit descriptions are as follows:



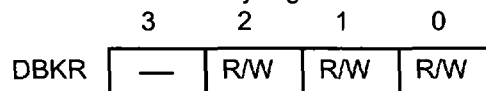
Note: R/W means read/write available.

Bit 3, Bit 2, Bit 1, Bit 0 Working registers page preselect bits:

- 0000 = WR Page 0 (00H - 0FH)
- 0001 = WR Page 1 (10H - 1FH)
- 0010 = WR Page 2 (20H - 2FH)
- 0011 = WR Page 3 (30H - 3FH)
- 0100 = WR Page 4 (40H - 4FH)
- 0101 = WR Page 5 (50H - 5FH)
- 0110 = WR Page 6 (60H - 6FH)
- 0111 = WR Page 7 (70H - 7FH)
- 1000 = WR Page 8 (80H - 8FH)
- 1001 = WR Page 9 (90H - 9FH)
- 1010 = WR Page A (A0H - AFH)
- 1011 = WR Page B (B0H - BFH)
- 1100 = WR Page C (C0H - CFH)
- 1101 = WR Page D (D0H - DFH)
- 1110 = WR Page E (E0H - EFH)
- 1111 = WR Page F (F0H - FFH)

6.4.4 Data Bank Register (DBKR)

The data bank register is organized as a 4-bit binary register. The bit descriptions are as follows:



Note: R/W means read/write available.

Bit 3 is reserved.

Bit 2, Bit 1, Bit 0 Data memory bank preselect bits:

- 000 = Data bank 0 (000H - 07FH)
- 001 = Data bank 1 (080H - 0FFH)
- 010 = Data bank 2 (100H - 17FH)
- 011 = Data bank 3 (180H - 1FFH)



- 100 = Data bank 4 (200H - 27FH)
- 101 = Data bank 5 (280H - 2FFH)
- 110 = Data bank 6 (300H - 37FH)
- 111 = Data bank 7 (380H - 3FFH)

6.5 Accumulator (ACC)

The accumulator (ACC) is a 4-bit register used to hold results from the ALU and transfer data between the memory, I/O ports, and registers.

6.6 Arithmetic and Logic Unit (ALU)

This is a circuit which performs arithmetic and logic operations. The ALU provides the following functions:

- Logic operations: ANL, XRL, ORL
- Branch decisions: JB0, JB1, JB2, JB3, JNZ, JZ, JC, JNC, DSKZ, DSKNZ, SKB0, SKB1, SKB2, SKB3
- Shift operations: SHRC, RRC, SHLC, RLC
- Binary additions/subtractions: ADC, SBC, ADD, SUB, ADU, DEC, INC

After any of the above instructions are executed, the status of the carry flag (CF) and zero flag (ZF) is stored in the internal registers. CF can be read out by executing MOV R, CF.

6.7 Main Oscillator

The W742C800 provides a crystal/RC oscillation circuit to generate the system clock through external connections. The requirement of crystal or RC oscillator can be selected by option code. If a crystal oscillator is used, a crystal or a ceramic resonator must be connected to XIN1 and XOUT1, and a capacitor must be connected to XIN1 and V_{SS} if an accurate frequency is needed. The suitable frequency range for crystal oscillator is from 400 KHz to 4.19 Mhz. If the RC oscillator is used, a resistor must be connected to XIN1 and XOUT1. The typical frequency of the RC oscillator is 2 Mhz.

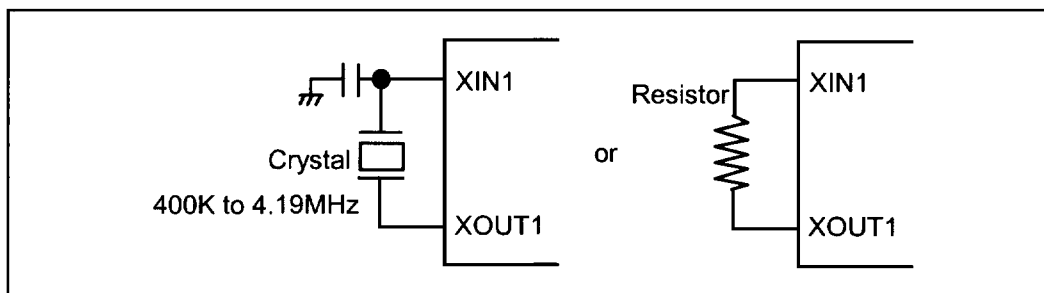


Figure 6-3 System clock oscillator Configuration

6.8 Sub-Oscillator

The sub-oscillator is used for dual-clock operation mode. In the sub-oscillator application, just only the 32768 Hz crystal could be connected to XIN2 and XOUT2, and a capacitor must be connected to XIN2 and V_{SS} if an accurate frequency is needed. The sub-oscillator would be oscillatory continuously in STOP mode.



6.9 Dividers

Each divider is organized as a 14-bit binary up-counter designed to generate periodic interrupts. When the main oscillator starts action, the Divider0 is incremented by each clock (FOSC). When an overflow occurs, the Divider0 event flag is set to 1 (EVF.0 = 1). Then, if the Divider0 interrupt enable flag has been set (IEF.0 = 1), the interrupt is executed, while if the hold release enable flag has been set (HEF.0 = 1), the hold state is terminated. And the last 4-stage of the Divider0 can be reset by executing CLR DIVR0 instruction. If the main oscillator is connected to the 32768 Hz crystal, the EVF.0 will be set to 1 periodically at the period of 500 mS.

If the sub-oscillator starts action, the Divider1 is incremented by each clock (Fs). When an overflow occurs, the Divider1 event flag is set to 1 (EVF.4 = 1). Then, if the Divider1 interrupt enable flag has been set (IEF.4 = 1), the interrupt is executed, while if the hold release enable flag has been set (HEF.4 = 1), the hold state is terminated. And the last 4-stage of the Divider1 can be reset by executing CLR DIVR1 instruction. Same as EVF.0, the EVF.4 is set to 1 periodically. But there are two period time (125 mS & 500mS) that can be selected by setting the SCR.3 bit. When SCR.3 = 0 (default), the 500 mS period time is selected; SCR.3 = 1, the 125 mS period time is selected.

6.10 Dual-clock operation

In the dual-clock mode, the clock source of the LCD frequency selector should be the sub-oscillator clock (32768 Hz) only. So when the STOP instruction is executing, the LCD will keep working in the dual-clock mode.

In this dual-clock mode, the normal operation is performed by generating the system clock from the main-oscillator clock (Fm). As required, the slow operation can be performed by generating the system clock from the sub-oscillator clock (Fs). The exchange of the normal operation and the slow operation is performed by resetting or setting the bit 0 of the System clock Control Register (SCR). If the SCR.0 is reset to 0, the clock source of the system clock generator is main-oscillator clock; if the SCR.0 is set to 1, the clock source of the system clock generator is sub-oscillator clock. In the dual-clock mode, the main-oscillator can stop oscillating when the STOP instruction is executing or the SCR.1 is set to 1.

When the SCR is set or reset, we must care the following cases:

1. X000B → X011B: we should not exchange the FOSC from Fm into Fs and disable Fm simultaneously. We could first exchange the FOSC from Fm into Fs, then disable the main-oscillator. So it should be X000B→X001B→X011B.

2. X011B → X000B: we should not enable Fm and exchange the FOSC from Fs into Fm simultaneously. We could first enable the main-oscillator; the 2nd step is calling a delay subroutine to wait the main-oscillator oscillating stably; then exchange the

FOSC from Fs into Fm is the last step. So it should be X011B→X001B→delay the Fm oscillating stably time→X000B. The suggestion of the Fm oscillating stably time is 3.5ms for 455K Hz and 0.8ms for 4M Hz.

We must remember that the X010B state is inhibitive, because it will induce the system shutdown.

The organization of the dual-clock operation mode is shown in Figure 6-4.

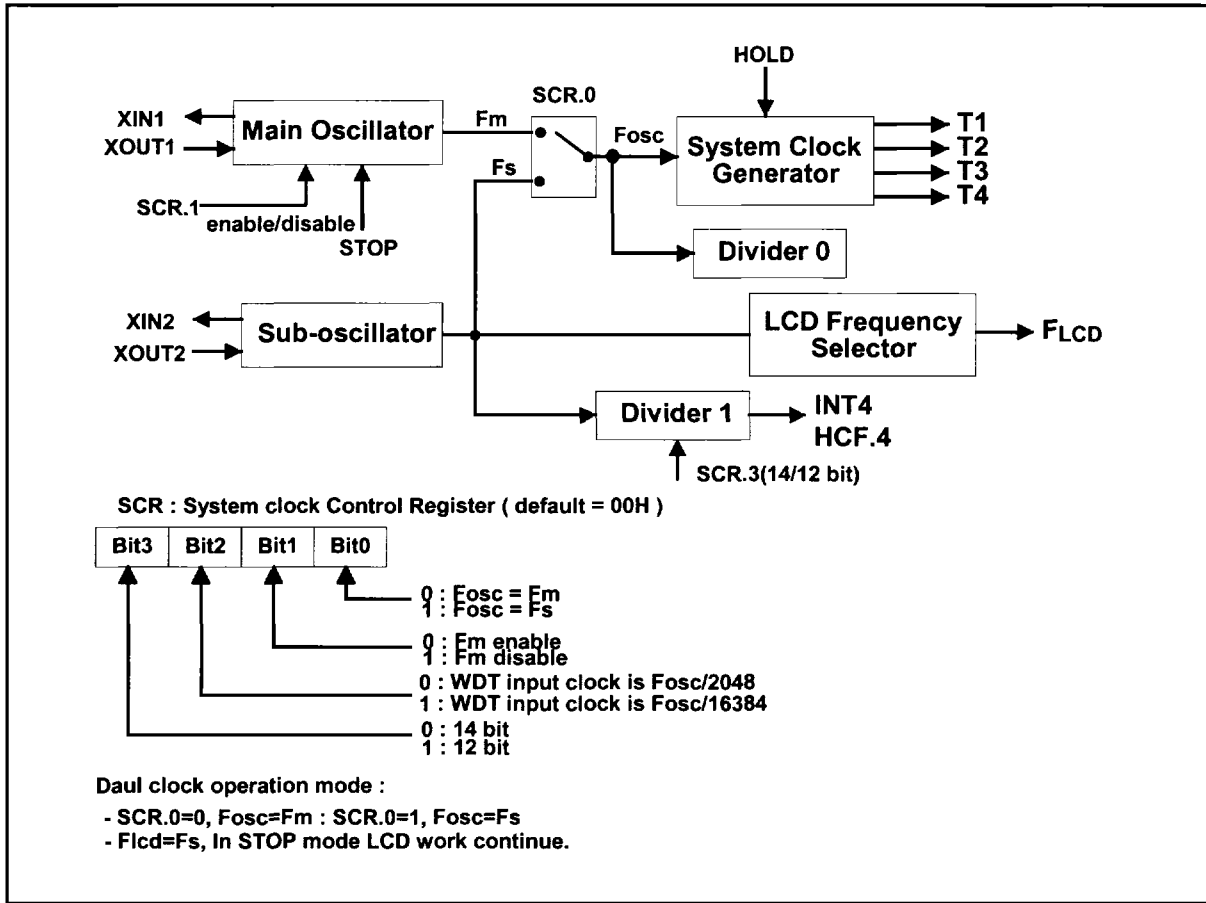


Figure 6-4 Organization of the dual-clock operation mode

6.11 Watchdog Timer (WDT)

The watchdog timer (WDT) is organized as a 4-bit up counter designed to prevent the program from unknown errors. When the corresponding option code bit of the WDT set to 1, the WDT is enabled, and if the WDT overflows, the chip will be reset. At initial reset, the input clock of the WDT is $F_{osc}/2048$. The input clock of the WDT can be switched to $F_{osc}/16384$ (or $F_{osc}/2048$) by setting SCR.2 to 1 (or clearing SCR.2 to 0). The contents of the WDT can be reset by the instruction CLR WDT. In normal operation, the application program must reset WDT before it overflows. A WDT overflow indicates that operation is not under control and the chip will be reset. The WDT overflow period is 1000 mS when the system clock (F_{osc}) is 32 KHz and WDT clock input is $F_{osc}/2048$. When the corresponding option code bit of the WDT set to 0, the WDT function is disabled. The organization of the Divider0 and watchdog timer is shown in Figure 6-5.

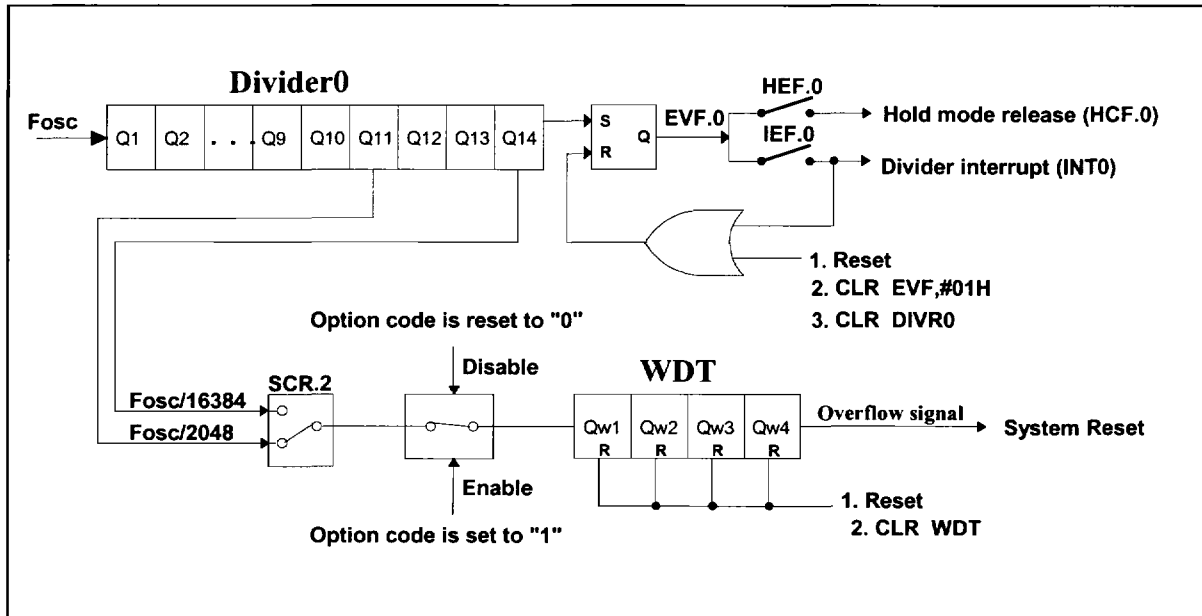


Figure 6-5 Organization of Divider0 and watchdog timer

6.12 Timer/Counter

6.12.1 Timer 0 (TM0)

Timer 0 (TM0) is a programmable 8-bit binary down-counter. The specified value can be loaded into TM0 by executing the MOV TM0L(TM0H),R instructions. When the MOV TM0L(TM0H),R instructions are executed, it will stop the TM0 down-counting (if the TM0 is down-counting) and reset the MR0.3 to 0, and the specified value can be loaded into TM0. Then we can set MR0.3 to 1, that will cause the event flag 1 (EVF.1) is reset and the TM0 starts to count. When it decrements to FFH, Timer 0 stops operating and generates an underflow (EVF.1 = 1). Then, if the Timer 0 interrupt enable flag has been set (IEF.1 = 1), the interrupt is executed, while if the hold release enable flag 1 has been set (HEF.1 = 1), the hold state is terminated. The Timer 0 clock input can be set as Fosc/1024 or Fosc/4 by setting MR0.0 to 1 or resetting MR0.0 to 0. The default timer value is Fosc/4. The organization of Timer 0 is shown in Figure 6-6.

If the Timer 0 clock input is Fosc/4:

$$\text{Desired Timer 0 interval} = (\text{preset value} + 1) \times 4 \times 1/\text{Fosc}$$

If the Timer 0 clock input is Fosc/1024:

$$\text{Desired Timer 0 interval} = (\text{preset value} + 1) \times 1024 \times 1/\text{Fosc}$$

Preset value: Decimal number of Timer 0 preset value

Fosc: Clock oscillation frequency

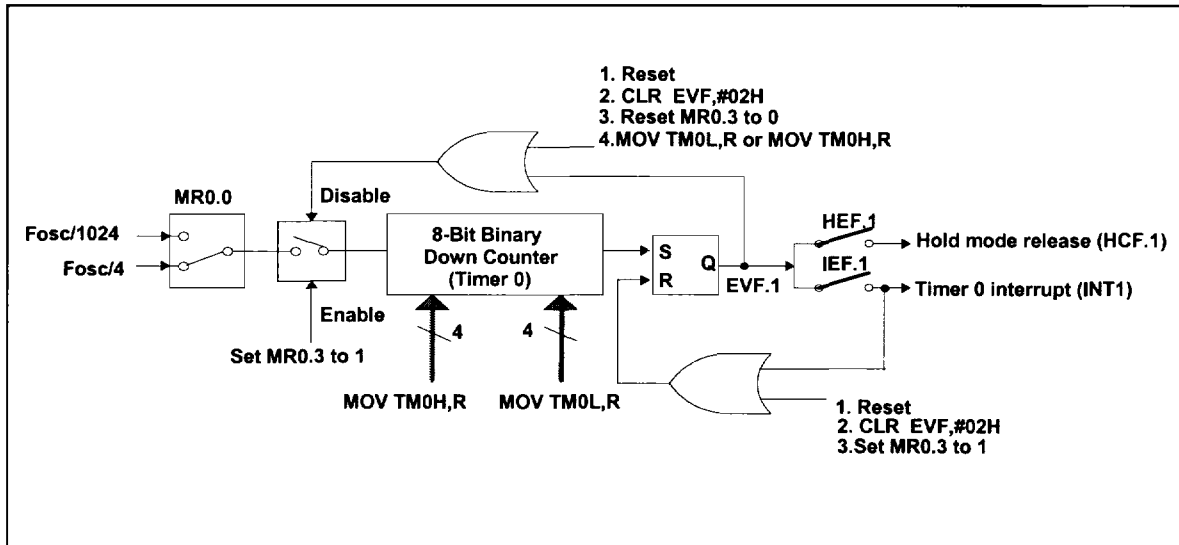


Figure 6-6 Organization of Timer 0

6.12.2 Timer 1 (TM1)

Timer 1 (TM1) is also a programmable 8-bit binary down counter, as shown in Figure 6-7. Timer 1 can be used as to output an arbitrary frequency to the BUZ1 pin. The input clock of Timer 1 can be one of three sources: Fosc/64, Fosc or Fs. The source can be selected by setting bit 0 and bit1 of mode register 1 (MR1). At initial reset, the Timer 1 clock input is Fosc. When the MOV TM1L, R or MOV TM1H,R instruction is executed, the specified data are loaded into the auto-reload buffer and the TM1 down-counting will be disabled that is MR1.3 is reset to 0 at the same time. If the bit 3 of MR1 is set (MR1.3 = 1), the content of the auto-reload buffer will be loaded into the TM1 down counter, and Timer 1 starts to down count, and the event flag 7 is reset (EVF.7=0). When the timer decrements to FFH, it will generate an underflow (EVF.7 = 1) and be auto-reloaded with the specified data, after which it will continue to count down. Then, if interrupt enable flag 7 has been set to 1 (IEF.7 = 1), an interrupt is executed; if hold mode release enable flag 7 is set to 1 (HEF.7 = 1), the hold state is terminated. The specified frequency of Timer 1 can be delivered to the BUZ1 output pin by programming bit0 and bit1 of BUZCR. Bit 3 of MR1 can be used to make Timer 1 stop or start counting.

In a case where Timer 1 clock input is FT:

$$\text{Desired Timer 1 interval} = (\text{preset value} + 1) / FT$$

$$\text{Desired frequency for MFP output pin} = FT \div (\text{preset value} + 1) \div 2 \text{ (Hz)}$$

Preset value: Decimal number of Timer 1 preset value

Fosc: Clock oscillation frequency

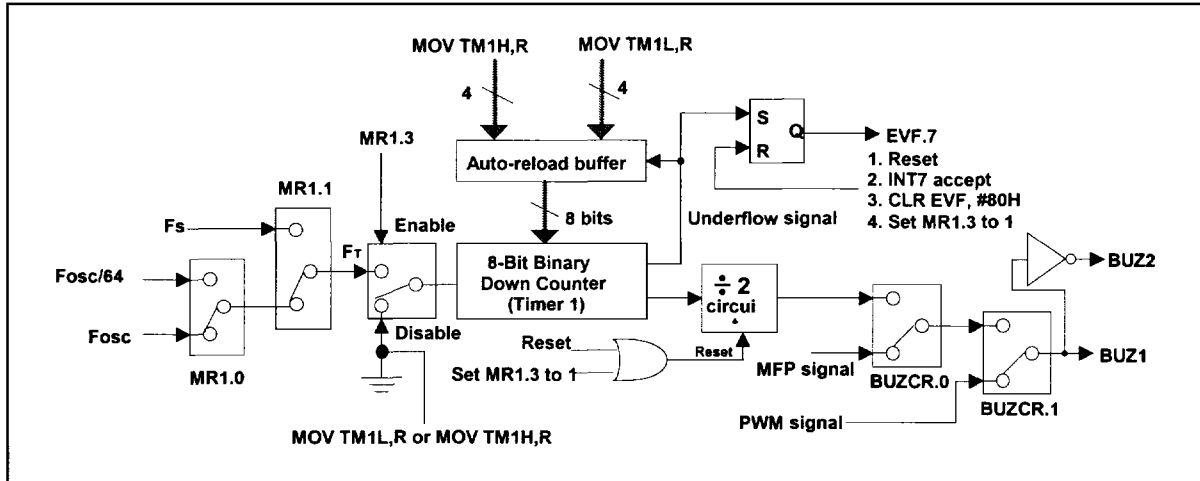


Figure 6-7 Organization of Timer 1

For example, when Ft equals 32768 Hz, depending on the preset value of TM1, the BUZ1 pin will output a single tone signal in the tone frequency range from 64 Hz to 16384 Hz. The relation between the tone frequency and the preset value of TM1 is shown in the table below.

		3rd octave		4th octave		5th octave				
		Tone frequency	TM1 preset value & MFP frequency	Tone frequency	TM1 preset value & MFP frequency	Tone frequency	TM1 preset value & MFP frequency			
TONE	C	130.81	7CH	131.07	261.63	3EH	260.06	523.25	1EH	528.51
	C#	138.59	75H	138.84	277.18	3AH	277.69	554.37	1CH	564.96
	D	146.83	6FH	146.28	293.66	37H	292.57	587.33	1BH	585.14
	D#	155.56	68H	156.03	311.13	34H	309.13	622.25	19H	630.15
	E	164.81	62H	165.49	329.63	31H	327.68	659.26	18H	655.36
	F	174.61	5DH	174.30	349.23	2EH	372.36	698.46	16H	712.34
	F#	185.00	58H	184.09	369.99	2BH	390.09	739.99	15H	744.72
	G	196.00	53H	195.04	392.00	29H	420.10	783.99	14H	780.19
	G#	207.65	4EH	207.39	415.30	26H	443.81	830.61	13H	819.20
	A	220.00	49H	221.40	440.00	24H	442.81	880.00	12H	862.84
	A#	233.08	45H	234.05	466.16	22H	468.11	932.23	11H	910.22
	B	246.94	41H	248.24	493.88	20H	496.48	987.77	10H	963.76

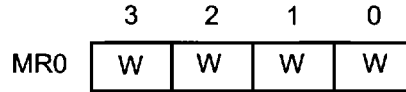
Note: Central tone is A4 (440 Hz).

Table 2 The relation between the tone frequency and the preset value of TM1



6.12.3 Mode Register 0 (MR0)

Mode Register 0 is organized as a 4-bit binary register (MR0.0 to MR0.3). MR0 can be used to control the operation of Timer 0. The bit descriptions are as follows:

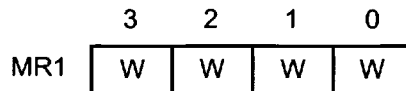


Note: W means write only.

- Bit 0 = 0 The fundamental frequency of Timer 0 is $F_{osc}/4$.
- = 1 The fundamental frequency of Timer 0 is $F_{osc}/1024$.
- Bit 1 & Bit 2 are reserved
- Bit 3 = 0 Timer 0 stops down-counting.
- = 1 Timer 0 starts down-counting.

6.12.4 Mode Register 1 (MR1)

Mode Register 1 is organized as a 4-bit binary register (MR1.0 to MR1.3). MR1 can be used to control the operation of Timer 1. The bit descriptions are as follows:



Note: W means write only.

- Bit 0 = 0 The internal fundamental frequency of Timer 1 is F_{osc} .
- = 1 The internal fundamental frequency of Timer 1 is $F_{osc}/64$.
- Bit 1 = 0 The fundamental frequency source of Timer 1 is the internal clock.
- = 1 The fundamental frequency source of Timer 1 is the sub-oscillator frequency F_s (32.768 KHz).
- Bit 2 is reserved.
- Bit 3 = 0 Timer 1 stops down-counting.
- = 1 Timer 1 starts down-counting.

6.13 Interrupts

The W742C800 provides five internal interrupt sources (Divider 0, Divider 1, Timer 0, Timer 1, fsampling) and one external interrupt source (port RC). Vector addresses for each of the interrupts are located in the range of program memory (ROM) addresses 004H to 020H. The flags IEF, PEF, and EVF are used to control the interrupts. When EVF is set to "1" by hardware and the corresponding bits of IEF and PEF have been set by software, an interrupt is generated. When an interrupt occurs, all of the interrupts are inhibited until the EN INT or MOV IEF,#1 instruction is invoked. The interrupts can also be disabled by executing the DIS INT instruction. When an interrupt is generated in hold mode, the hold mode will be released momentarily and interrupt subroutine will be executed. After the RTN instruction is executed in an interrupt subroutine, the μC will enter hold mode again. The operation flow chart is shown in Figure 6-9. The control diagram is shown below.

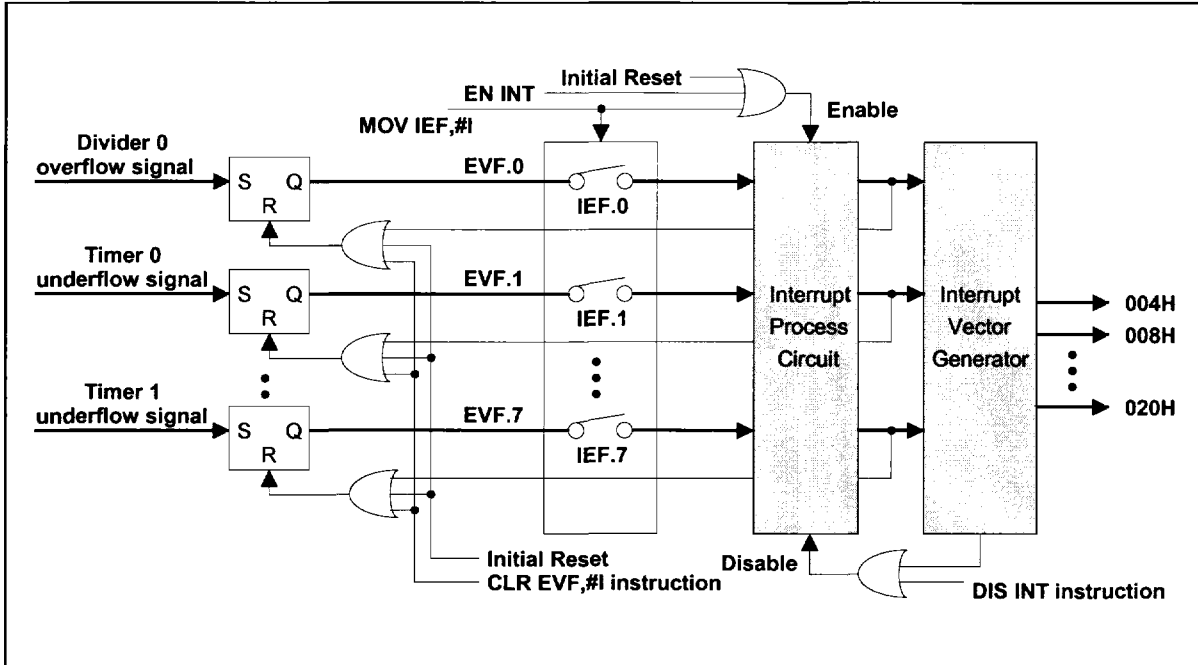


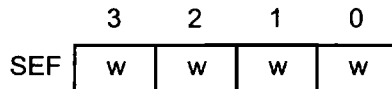
Figure 6-8 Interrupt event control diagram

6.14 Stop Mode Operation

In stop mode, all operations of the μC cease (excluding the operation of the sub-oscillator, Divider 1 and LCD driver), and the BUZ1 & BUZ2 pins are kept to low. The μC enters stop mode when the STOP instruction is executed and exits stop mode when an external trigger is activated (by a falling signal on the RC). When the designated signal is accepted, the μC awakens and executes the next instruction. To prevent erroneous execution, the NOP instruction should follow the STOP command. In the dual-clock slow operation mode, the STOP instruction will disable the main-oscillator oscillating; but the μC system is still operated by the sub-oscillator.

6.14.1 Stop Mode Wake-up Enable Flag for RC Port (SEF)

The stop mode wake-up flag for port RC is organized as an 4-bit binary register (SEF.0 to SEF.3). Before port RC may be used to make the device exit the stop mode, the content of the SEF must be set first. The SEF is controlled by the MOV SEF, #I instruction. The bit descriptions are as follows:



Note: W means write only.

SEF.0 = 1 Device will exit stop mode when falling edge signal is applied to pin RC.0

SEF.1 = 1 Device will exit stop mode when falling edge signal is applied to pin RC.1

SEF.2 = 1 Device will exit stop mode when falling edge signal is applied to pin RC.2



SEF.3 = 1 Device will exit stop mode when falling edge signal is applied to pin RC.3

6.15 Hold Mode Operation

In hold mode, all operations of the μC cease, except for the operation of the oscillator, Timer, Divider, LCD driver and PWM/MFP output. The μC enters hold mode when the HOLD instruction is executed. The hold mode can be released in one of six ways: by the action of timer 0, timer 1, divider 0, divider 1, fsampling, the RC port. Before the device enters the hold mode, the HEF, PEF, and IEF flags must be set to define the hold mode release conditions. For more details, refer to the instruction-set table and the following flow chart.

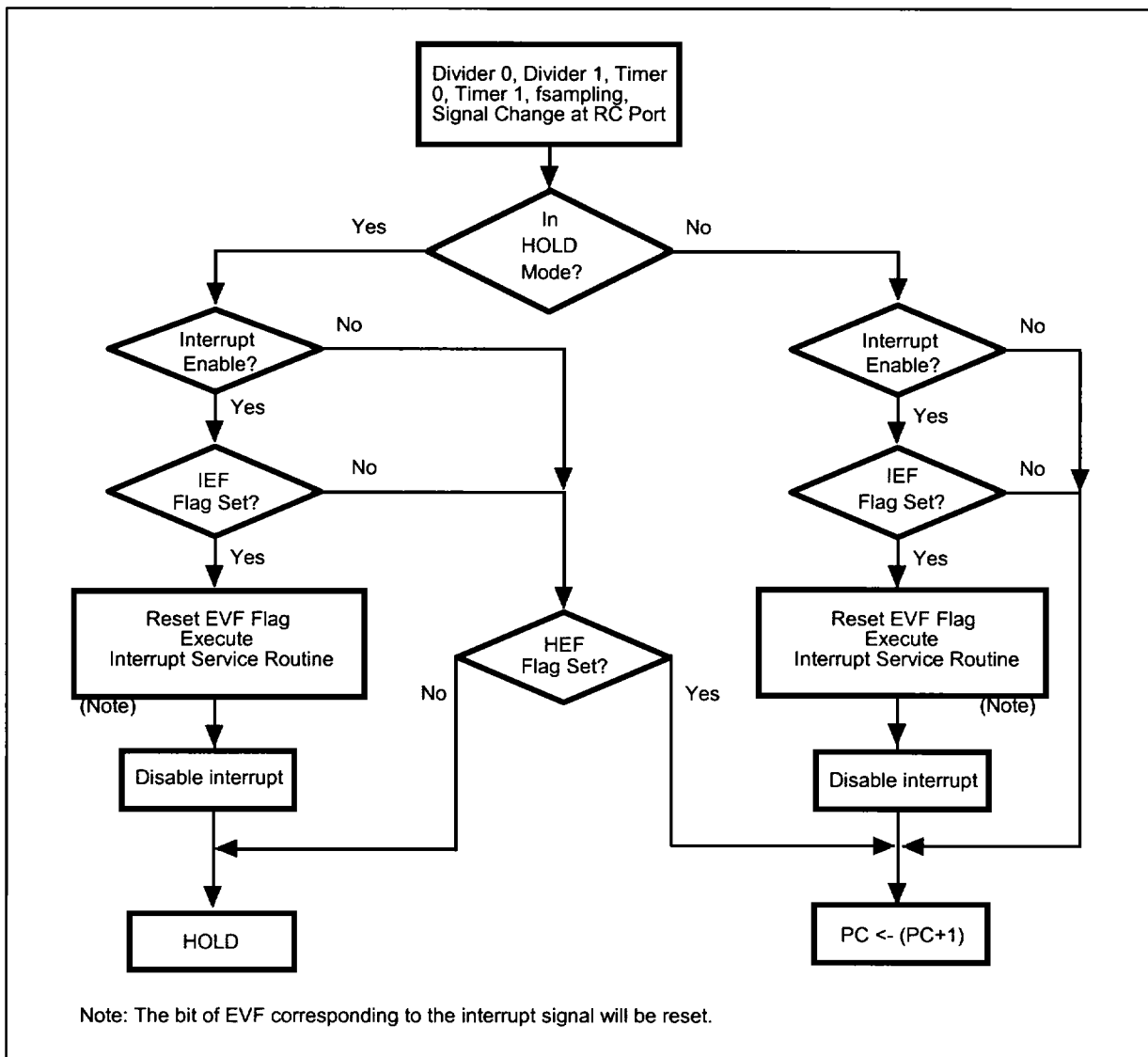


Figure 6-9 Hold Mode and Interrupt Operation Flow Chart



6.15.1 Hold Mode Release Enable Flag (HEF)

The hold mode release enable flag is organized as an 8-bit binary register (HEF.0 to HEF.7). The HEF is used to control the hold mode release conditions. It is controlled by the MOV HEF, #I instruction. The bit descriptions are as follows:

	7	6	5	4	3	2	1	0
HEF	w	—	—	w	w	w	w	w

Note: W means write only.

- HEF.0 = 1 Overflow from the Divider 0 causes Hold mode to be released.
- HEF.1 = 1 Underflow from Timer 0 causes Hold mode to be released.
- HEF.2 = 1 Signal change at port RC causes Hold mode to be released.
- HEF.3 = 1 6K/8K Hz sampling rate for PWM function.
- HEF.4 = 1 Overflow from the Divider 1 causes Hold mode to be released.
- HEF.5 & HEF.6 are reserved.
- HEF.7 = 1 Underflow from Timer 1 causes Hold mode to be released.

6.15.2 Interrupt Enable Flag (IEF)

The interrupt enable flag is organized as a 8-bit binary register (IEF.0 to IEF.7). These bits are used to control the interrupt conditions. It is controlled by the MOV IEF, #I instruction. When one of these interrupts is accepted, the corresponding to the bit of the event flag will be reset, but the other bits are unaffected. In interrupt subroutine, these interrupts will be disable till the instruction MOV IEF, #I or EN INT is executed again. Otherwise, these interrupts can be disable by executing DIS INT instruction. The bit descriptions are as follows:

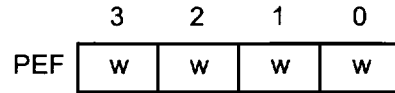
	7	6	5	4	3	2	1	0
IEF	w	—	—	w	w	w	w	w

Note: W means write only.

- IEF.0 = 1 Interrupt 0 is accepted by overflow from the Divider 0.
- IEF.1 = 1 Interrupt 1 is accepted by underflow from the Timer 0.
- IEF.2 = 1 Interrupt 2 is accepted by a signal change at port RC.
- IEF.3 = 1 Interrupt 3 is accepted by the sampling rate for the PWM function.
- IEF.4 = 1 Interrupt 4 is accepted by overflow from the Divider 1.
- IEF.5 & IEF.6 are reserved.
- IEF.7 = 1 Interrupt 7 is accepted by underflow from Timer 1.

6.15.3 Port Enable Flag (PEF)

The port enable flag is organized as 4-bit binary register (PEF.0 to PEF.3). Before port RC (used as input mode) may be used to release the hold mode or preform interrupt function, the content of the PEF must be set first. The PEF is controlled by the MOV PEF, #I instruction. The bit descriptions are as follows:



Note: W means write only.

PEF.0: Enable/disable the signal change at pin RC.0 to release hold mode or perform interrupt.

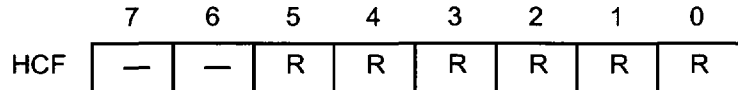
PEF.1: Enable/disable the signal change at pin RC.1 to release hold mode or perform interrupt.

PEF.2: Enable/disable the signal change at pin RC.2 to release hold mode or perform interrupt.

PEF.3: Enable/disable the signal change at pin RC.3 to release hold mode or perform interrupt.

6.15.4 Hold Mode Release Condition Flag (HCF)

The hold mode release condition flag is organized as a 8-bit binary register (HCF.0 to HCF.7). It indicates by which interrupt source the hold mode has been released, and is loaded by hardware. The HCF can be read out by the MOVA R, HCFL and MOVA R, HCFH instructions. When any of the HCF bits is "1," the hold mode will be released and the HOLD instruction is invalid. The HCF can be reset by the CLR EVF or MOV HEF,#I (HEF = 0) instructions. When EVF and HEF have been reset, the corresponding bit of HCF is reset simultaneously. The bit descriptions are as follows:



Note: R means read only.

HCF.0 = 1 Hold mode was released by overflow from the divider 0.

HCF.1 = 1 Hold mode was released by underflow from the timer 0.

HCF.2 = 1 Hold mode was released by a signal change at port RC.

HCF.3 = 1 Hold mode was released by the sampling rate for the PWM function.

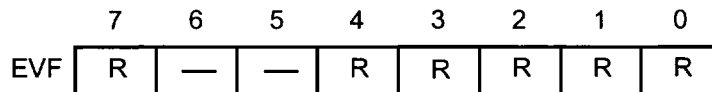
HCF.4 = 1 Hold mode was released by overflow from the divider 1.

HCF.5 = 1 Hold mode was released by underflow from the timer 1.

HCF.6 and HCF.7 are reserved.

6.15.5 Event Flag (EVF)

The event flag is organized as a 8-bit binary register (EVF.0 to EVF.7). It is set by hardware and reset by CLR EVF,#I instruction or the occurrence of an interrupt. The bit descriptions are as follows:



Note: R means read only.

EVF.0 = 1 Overflow from divider 0 occurred.

EVF.1 = 1 Underflow from timer 0 occurred.

EVF.2 = 1 Signal change at port RC occurred.



- EVF.3 = 1 PWM sampling time is up.
- EVF.4 = 1 Overflow from divider 1 occurred.
- EVF.5 & EVF.6 are reserved.
- EVF.7 = 1 Underflow from Timer 1 occurred.

6.16 Reset Function

The W742C800 is reset either by a power-on reset or by using the external $\overline{\text{RES}}$ pin. The initial state of the W742C800 after the reset function is executed is described below.

Program Counter (PC)	000H
TM0, TM1	Reset
MR0, MR1, PAGE registers	Reset
PSR0, SCR registers	Reset
IEF, HEF, HCF, PEF, EVF, SEF flags	Reset
WRP, DBKR register	Reset
Timer 0 input clock	FOSC/4
Timer 1 input clock	FOSC
MFP output	Low
PWM & PWMB output	Low
Input/output ports RA, RC	Input mode
RA ports output type	CMOS type
PWM fuction	Disable
Input clock of the watchdog timer	FOSC/2048
LCD display	OFF

Table 3 The initial state after the reset function is executed

6.17 Input/Output Ports RA, RC

Port RA consists of pins RA.0 to RA.3. Port RC consists of pins RC.0 to RC.3. At initial reset, input/output ports RA, RC are all in input mode. When RA is used as output port, CMOS or NMOS open drain output type can be selected by the PM0 register. But when RC is used as output port, the output type is just fixed to be CMOS output type. Each pin of port RA, RC can be specified as input or output mode independently by the PM1, PM4 registers. The MOVA R, RA or MOVA R, RC instructions operate the input functions and the MOV RA, R or MOV RC, R operate the output functions. Port RA and RC have the internal pull-high resistors that can be enabled/disabled by specifying the PM0.4 and PM0.2 respectively. When RC port is used as input mode and the PEF, HEF, and IEF corresponding to the RC port are set, a signal change at the specified pins of port RC will execute the hold mode release or interrupt subroutine. Port status register 0 (PSR0) records the status of ports RC, i.e., any signal changes on the pins that make up the ports. PSR0 can be read out and cleared by the MOV R, PSR0, and CLR PSR0 instructions. In addition, the falling edge signal on

the pin of port RC specified by the instruction MOV SEF, #I will cause the device to exit the stop mode. For more details, refer to the instruction table and Figure 6-10, Figure 6-11 and Figure 6-12.

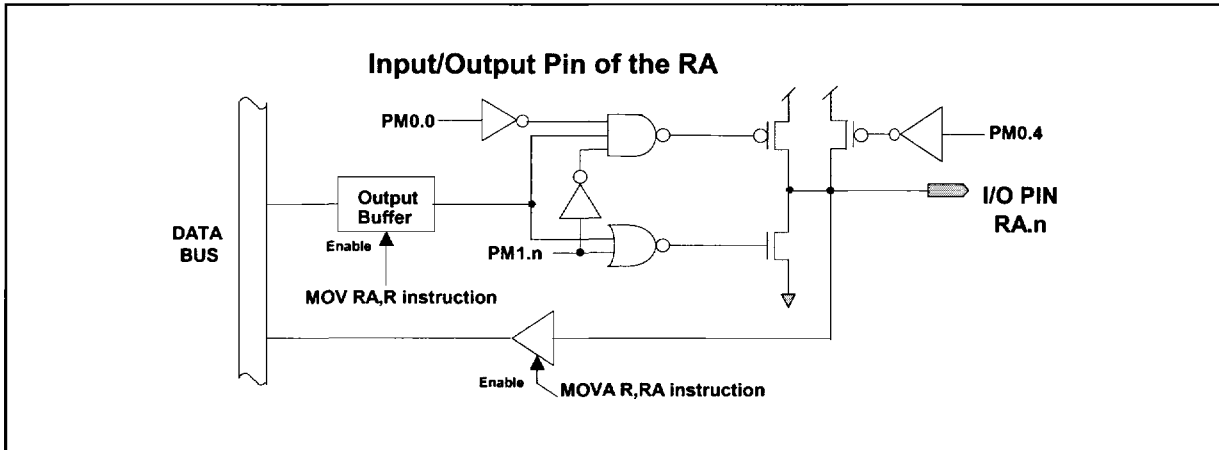


Figure 6-10 Architecture of RA Input/Output Pins

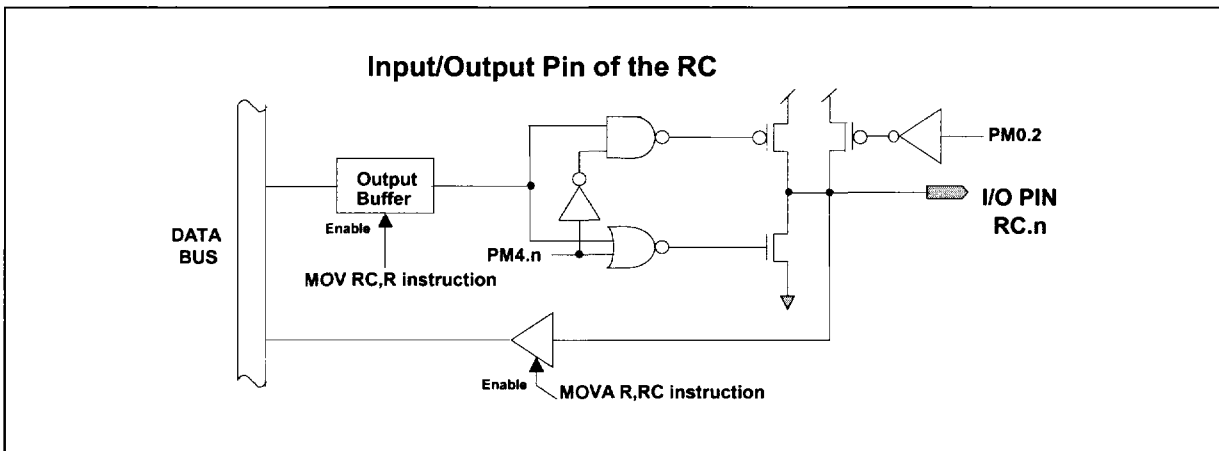


Figure 6-11 Architecture of RC Input/Output pins

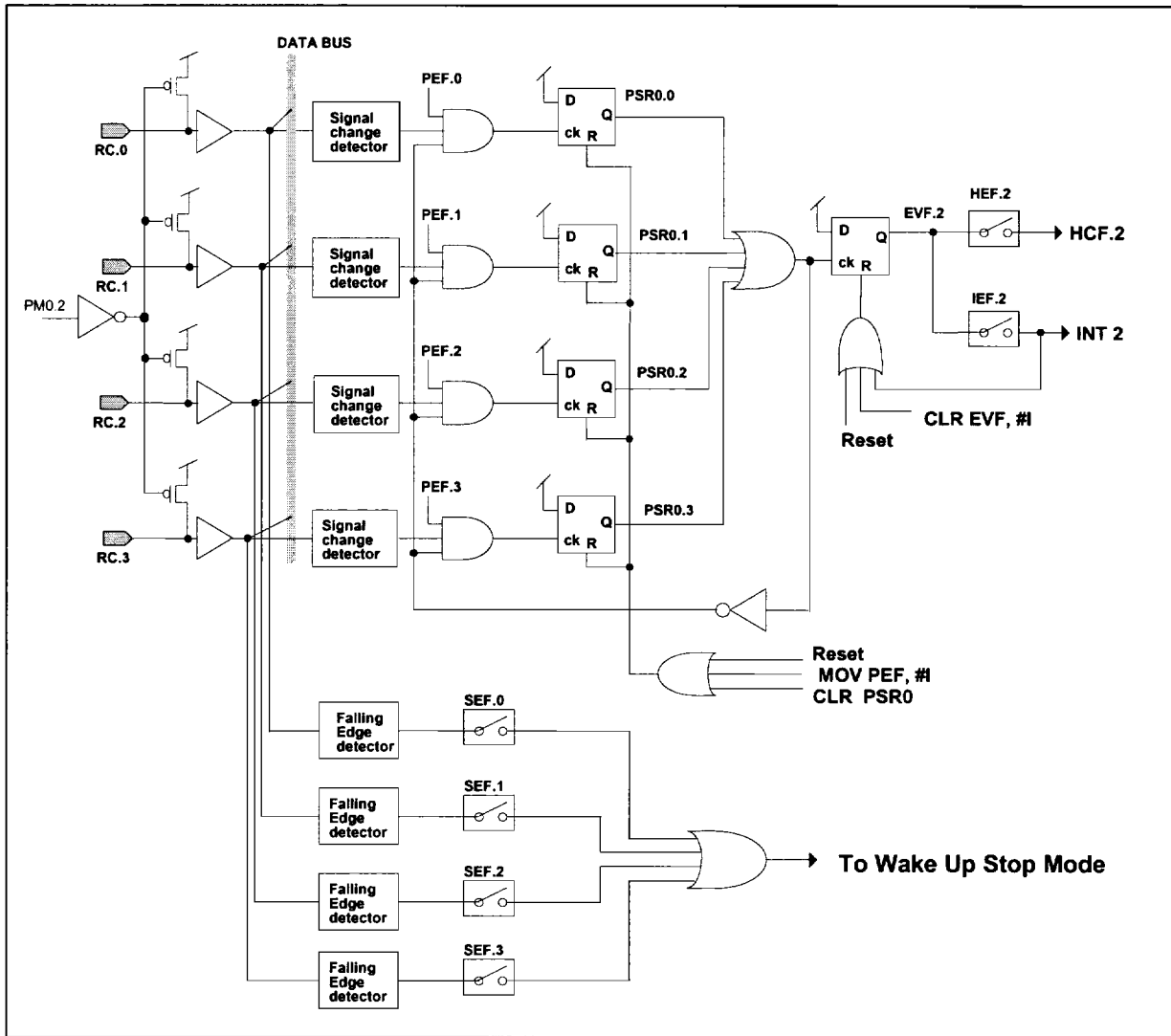
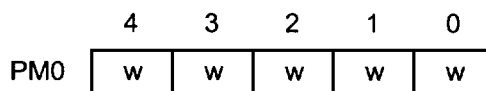


Figure 6-12 Architecture of Input Ports RC



6.17.1 Port Mode 0 Register (PM0)

The port mode 0 register is organized as 5-bit binary register (PM0.0 to PM0.4). PM0 can be used to determine the structure of the input/output ports; it is controlled by the MOV PM0, #I instruction. The bit descriptions are as follows:



Note: W means write only.

Bit 0 = 0 RA port is CMOS output type. Bit 0 = 1 RA port is NMOS open drain output type.

Bit 1 is reserved.

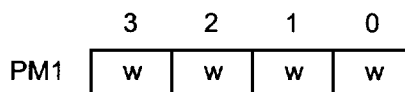
Bit 2 = 0 RC port pull-high resistor is disabled. Bit 2 = 1 RC port pull-high resistor is enabled.

Bit 3 are reserved.

Bit 4 = 0 RA port pull-high resistor is disabled. Bit 4 = 1 RA port pull-high resistor is enabled.

6.17.2 Port Mode 1 Register (PM1)

The port mode 1 register is organized as 4-bit binary register (PM1.0 to PM1.3). PM1 can be used to control the input/output mode of port RA. PM1 is controlled by the MOV PM1, #I instruction. The bit descriptions are as follows:



Note: W means write only.

Bit 0 = 0 RA.0 works as output pin; Bit 0 = 1 RA.0 works as input pin

Bit 1 = 0 RA.1 works as output pin; Bit 1 = 1 RA.1 works as input pin

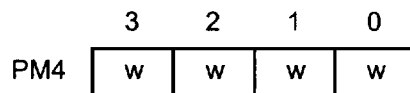
Bit 2 = 0 RA.2 works as output pin; Bit 2 = 1 RA.2 works as input pin

Bit 3 = 0 RA.3 works as output pin; Bit 3 = 1 RA.3 works as input pin

At initial reset, port RA is input mode (PM1 = 1111B).

6.17.3 Port Mode 4 Register (PM4)

The port mode 4 register is organized as 4-bit binary register (PM4.0 to PM4.3). PM4 can be used to control the input/output mode of port RC. PM4 is controlled by the MOV PM4, #I instruction. The bit descriptions are as follows:



Note: W means write only.

Bit 0 = 0 RC.0 works as output pin; Bit 0 = 1 RC.0 works as input pin

Bit 1 = 0 RC.1 works as output pin; Bit 1 = 1 RC.1 works as input pin

Bit 2 = 0 RC.2 works as output pin; Bit 2 = 1 RC.2 works as input pin

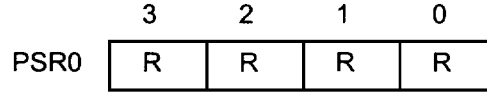
Bit 3 = 0 RC.3 works as output pin; Bit 3 = 1 RC.3 works as input pin

At initial reset, the port RC is input mode (PM4 = 1111B).



6.17.4 Port Status Register 0 (PSR0)

Port status register 0 is organized as 4-bit binary register (PSR0.0 to PSR0.3). PSR0 can be read or cleared by the MOVA R, PSR0, and CLR PSR0 instructions. The bit descriptions are as follows:



Note: R means read only.

- Bit 0 = 1 Signal change at RC.0
- Bit 1 = 1 Signal change at RC.1
- Bit 2 = 1 Signal change at RC.2
- Bit 3 = 1 Signal change at RC.3

6.18 BUZ1 & BUZ2 Output Pin

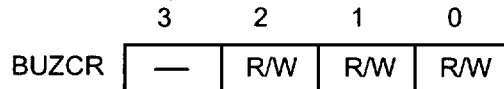
The BUZ2 pin output is the inverse output of the BUZ1 pin, which are connected to the output of the internal CMOS output buffer. There are three sources that can deliver to BUZ1 and BUZ2 output pins. The sources are MFP, programmable single tone of TM1 and PWM. The bit0 and bit1 of the buzzer control register (BUZCR) can determine which one can be delivered to BUZ1 and BUZ2 pins. When the bit1 and bit0 of BUZCR are all zero, the MFP signal will be delivered to BUZ output pins. When the bit1 is 0 and bit0 is 1, the programmable single tone of TM1 will be delivered to BUZ output pins. And the bit1 of BUZCR is 1, the PWM signal will be delivered to BUZ output pins. This signal source selection of BUZ1 and BUZ2 output pins can refer to Figure 6-7. The BUZ1 and BUZ2 pins can be enabled or disabled by using the EN BUZ or DIS BUZ instructions. When DIS BUZ instruction is executing, the BUZ1 and BUZ2 pins will all be kept low.

6.18.1 Pulse Width Modulation (PWM) Output

When the output of BUZ pins is PWM signal, the MOV PWM, R instruction can load the content of the ACC (low nibble) and R (high nibble) into the 8-bit PWM register in order to output the voice data through the PWM generator. There are two voice output sampling rates (6K or 8K Hz) that can be controlled by the bit2 of BUZCR.

6.18.2 Buzzer Control Register (BUZCR)

Buzzer control register is organized as 4-bit binary register (BUZCR.0 to BUZCR.3). BUZCR can be read by the MOV R, BUZCR instruction. And this register can be loaded by the MOV BUZCR, #I or MOV BUZCR, R instructions. The bit descriptions are as follows:



Note: R/W means read or write.

- Bit 0 = 0 MFP signal is delivered to BUZ pins.
- = 1 TM1 programmable single tone is delivered to BUZ pins.
- Bit 1 = 0 MFP or TM1 programmable single tone is selected.
- = 1 PWM signal is delivered to BUZ pins.
- Bit 2 = 0 PWM sampling rate is 6 Khz.



= 1 PWM sampling rate is 8 KHz.

Bit 3 is reserved.

6.18.3 MFP Output

When the output of BUZ pins is MFP signal, the BUZ pins can deliver a modulation output in any combination of one signal from among DC, 4096Hz, 2048Hz, and one or more signals from among 128 Hz, 64 Hz, 8 Hz, 4 Hz, 2 Hz, or 1 Hz (when using a 32.768 KHz crystal). The MOV MFP, #I instruction is used to specify the modulation output combination. The data specified by the 8-bit operand and the MFP output signal are shown in Table 4.

(Fosc = 32.768 KHz)

R7 R6	R5	R4	R3	R2	R1	R0	FUNCTION
0 0	0	0	0	0	0	0	Low level
	0	0	0	0	0	1	128 Hz
	0	0	0	0	1	0	64 Hz
	0	0	0	1	0	0	8 Hz
	0	0	1	0	0	0	4 Hz
	0	1	0	0	0	0	2 Hz
	1	0	0	0	0	0	1 Hz
0 1	0	0	0	0	0	0	High level
	0	0	0	0	0	1	128 Hz
	0	0	0	0	1	0	64 Hz
	0	0	0	1	0	0	8 Hz
	0	0	1	0	0	0	4 Hz
	0	1	0	0	0	0	2 Hz
	1	0	0	0	0	0	1 Hz
1 0	0	0	0	0	0	0	2048 Hz
	0	0	0	0	0	1	2048 Hz * 128 Hz
	0	0	0	0	1	0	2048 Hz * 64 Hz
	0	0	0	1	0	0	2048 Hz * 8 Hz
	0	0	1	0	0	0	2048 Hz * 4 Hz
	0	1	0	0	0	0	2048 Hz * 2 Hz
	1	0	0	0	0	0	2048 Hz * 1 Hz
1 1	0	0	0	0	0	0	4096 Hz
	0	0	0	0	0	1	4096 Hz * 128 Hz
	0	0	0	0	1	0	4096 Hz * 64 Hz
	0	0	0	1	0	0	4096 Hz * 8 Hz
	0	0	1	0	0	0	4096 Hz * 4 Hz
	0	1	0	0	0	0	4096 Hz * 2 Hz
	1	0	0	0	0	0	4096 Hz * 1 Hz

Table 4 The relation between the MFP output frequency and the data specified by 8-bit operand



6.19 LCD Controller/Driver

The W742C800 can directly drive an LCD pannel with 40 segments × 8 commons. The LCD driving mode is 1/4 bias 1/8 duty. The alternating frequency of the LCD comes from sub-oscillator and is fixed at 64Hz.

Corresponding to the 40 LCD drive output pins, there are 80 LCD data RAMs. Instructions such as MOV LP0,R, MOV LP1,R, MOV @LP,R, and MOV R,@LP are used to control the LCD data RAM. The data in the LCD data RAM are transferred to the segment output pins automatically without program control. When the bit value of the LCD data RAM is "1," the LCD is turned on. When the bit value of the LCD data RAM is "0," LCD is turned off. The contents of the LCD data RAM (LCDR) are sent out through the segment0 to segment39 pins by a direct memory access. The relation between the LCD data RAM and segment/common pins is shown below.

OUTPUT PIN	LCD RAM	COM7 BIT3	COM6 BIT2	.	COM4 BIT0	LCD RAM	COM3 BIT3	COM2 BIT2	.	COM0 BIT0
SEG0	LCDR01	0/1	0/1	.	0/1	LCDR00	0/1	0/1	.	0/1
SEG1	LCDR03	0/1	0/1	.	0/1	LCDR02	0/1	0/1	.	0/1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
SEG38	LCDR4D	0/1	0/1	.	0/1	LCDR4C	0/1	0/1	.	0/1
SEG39	LCDR4F	0/1	0/1	.	0/1	LCDR4E	0/1	0/1	.	0/1

Table 5 The reation between the Lcdr and segments for 40 segments × 8 commons driving type

The LCDON instruction turns the LCD display on (even in HOLD mode), and the LCDOFF instruction turns the LCD display off. At initial reset, all the LCD segments are unlit. When the initial reset state ends, the LCD display is turned off automatically. To turn on the LCD display, the instruction LCDON must be executed.

6.19.1 LCD RAM addressing method

There are 80 LCD RAMs (LCDR00 - Lcdr4F) that should be indirectly addressed. The LCD RAM pointer (LP) is used to point to the address of the wanted LCD RAM. The LP is organized as 6-bit binary register. The MOV LP0,R and MOV LP1,R instructions can load the LCD RAM address to the LP from R. The MOV @LP,R and MOV R,@LP instructions can access the pointed LCD RAM content.

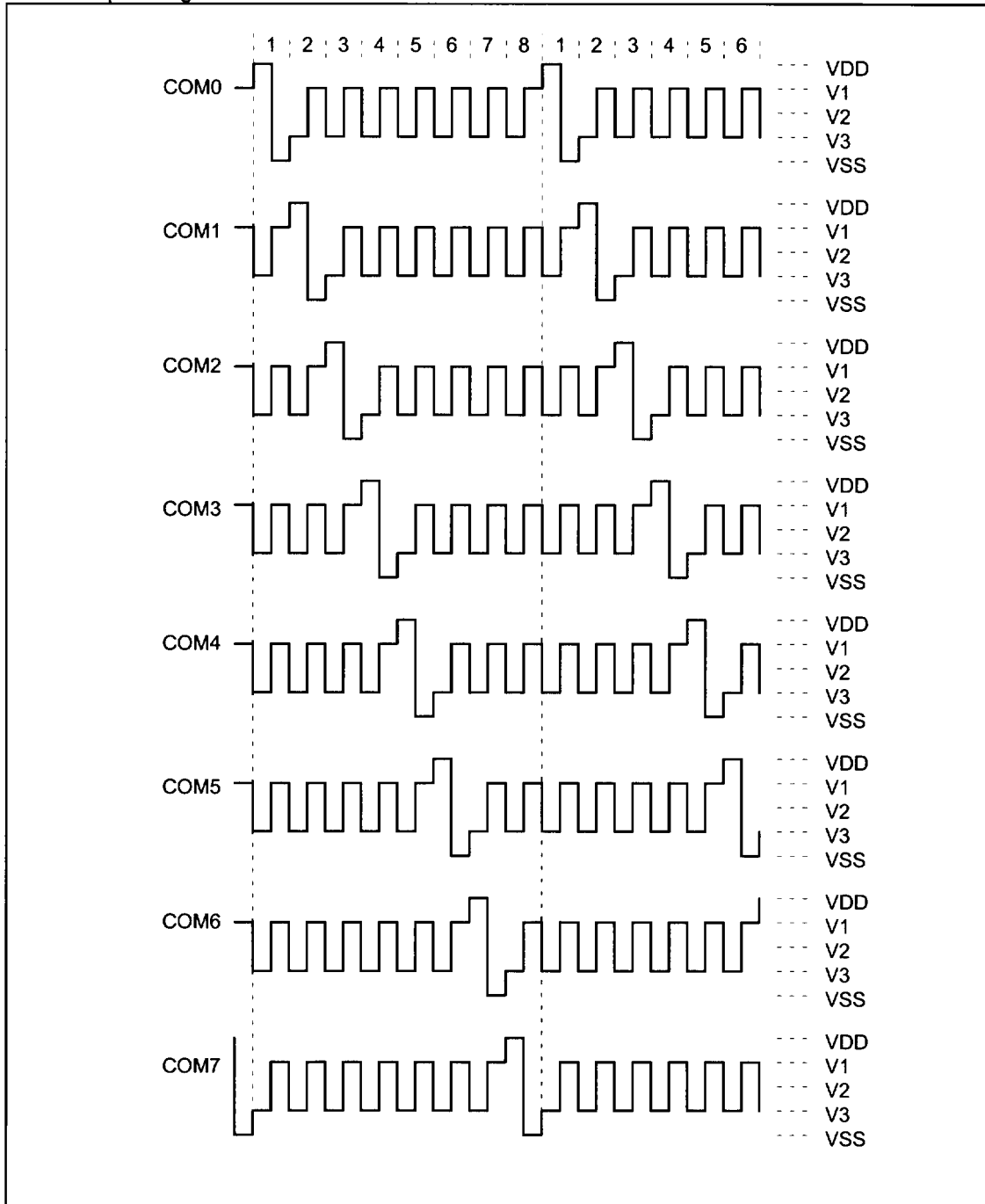
6.19.2 SEG0 - SEG7 using as DC output (NMOS open drain type)

Set the bit0 of the LCD control register (LCDCN) to 1 to change the SEG0 - SEG7 pins output type to DC output mode. The corresponding output registers are Lcdr00 and Lcdr01. Lcdr00 is used as low nibble (SEG0 - SEG3); and Lcdr01 is used as high nibble (SEG4 - SEG7). When SEG0 - SEG7 are used as DC output mode, their output stage is NMOS open drain type.



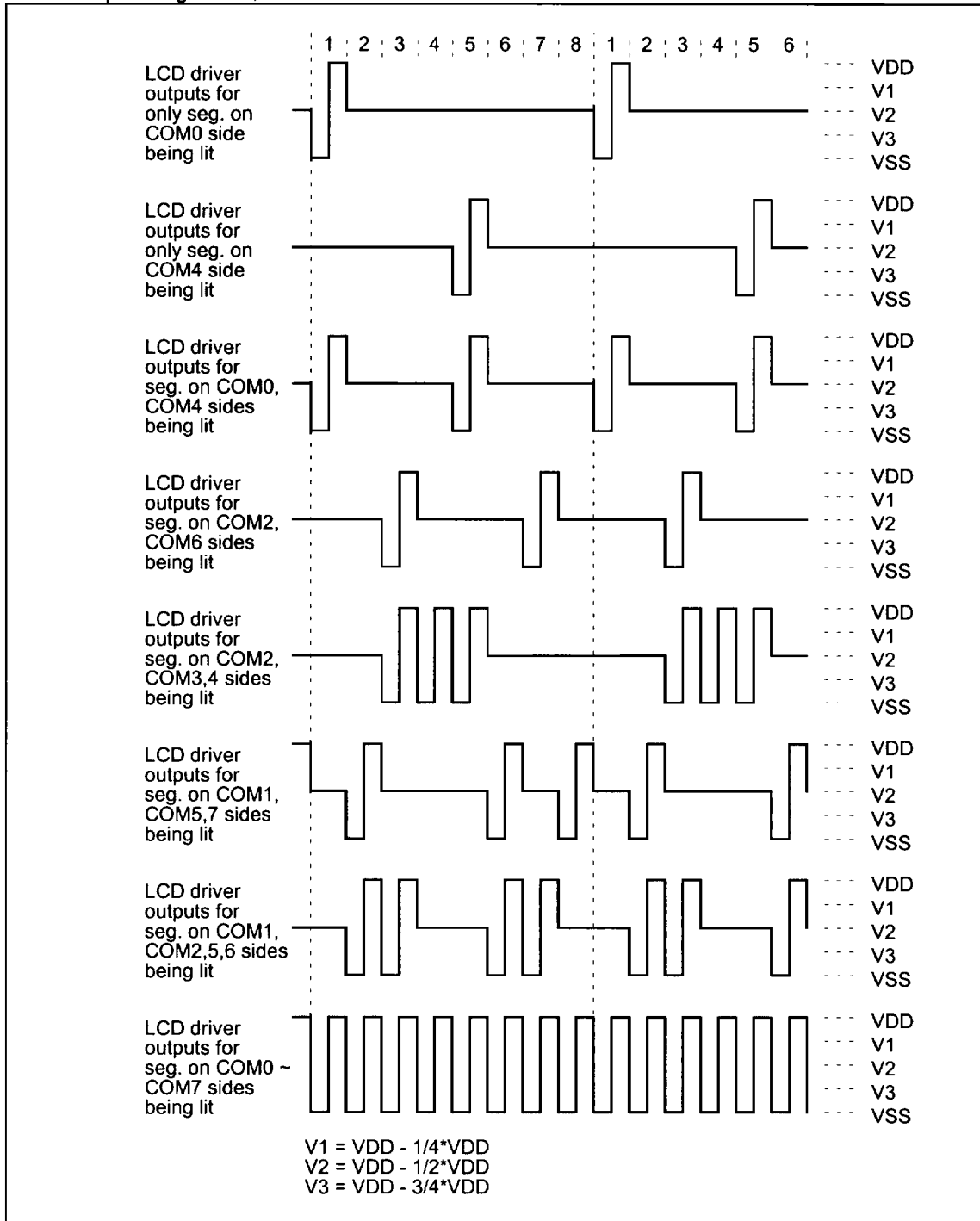
6.19.3 The output waveforms for the LCD driving mode

1/4 bias 1/8 duty Lighting System (Example)
Normal Operating Mode





Normal Operating Mode, continued





7. ABSOLUTE MAXIMUM RATINGS

PARAMETER	RATING	UNIT
Supply Voltage to Ground Potential	-0.3 to +7.0	V
Applied Input/Output Voltage	-0.3 to +7.0	V
Power Dissipation	120	mW
Ambient Operating Temperature	0 to +70	°C
Storage Temperature	-55 to +150	°C

Note: Exposure to conditions beyond those listed under Absolute Maximum Ratings may adversely affect the life and reliability of the device.

8. DC CHARACTERISTICS

(VDD-VSS = 3.0 V, Fm = 4.19MHz, Fs = 32.768 KHz, Ta = 25° C, LCD on; unless otherwise specified)

PARAMETER	SYM.	CONDITIONS	MIN.	TY P.	MAX.	UNIT
Op. Voltage	VDD	-	2.4	-	3.8	V
Op. Current (Crystal type)	IOP1	No load (Ext-V) In dual-clock normal operation	-	-	2.5	mA
Op. Current (Crystal type)	IOP3	No load (Ext-V) In dual-clock slow operation and Fm is stopped	-	-	20	μA
Hold Current (Crystal type)	IHM1	Hold mode No load (Ext-V) In dual-clock normal operation	-	-	450	μA
Hold Current (Crystal type)	IHM3	Hold mode No load (Ext-V) In dual-clock slow operation and Fm is stopped	-	6	15	μA
Stop Current (Crystal type)	ISM1	Stop mode No load (Ext-V) In dual-clock normal operation	-	6	15	μA
Input Low Voltage	VIL	-	VSS	-	0.3V _{DD}	V
Input High Voltage	VIH	-	0.7V _{DD}	-	VDD	V
BUZ Output Low Voltage	VML	IOL = 3.5mA	-	-	0.4	V
BUZ Output High Voltage	VMH	IOH = 3.5mA	2.4	-	-	V
Port RA, RC Output Low Voltage	VABL	IOL = 2.0mA	-	-	0.4	V
Port RA, RC Output high Voltage	VABH	IOH = 2.0mA	2.4	-	-	V
LCD Supply Current	ILCD	All Seg. ON	-	-	10	μA
SEG0-SEG39 Sink Current (Used as LCD output)	IOL1	VOL = 0.4V VLCD = 0.0V	90	-	-	μA
SEG0-SEG39 Drive Current (Used as LCD output)	IOH1	VOH = 2.4V VLCD = 3.0V	90	-	-	μA
RES Pull-up Resistor	RRES	-	20	100	500	KΩ



9. AC CHARACTERISTICS

PARAMETER	SYM.	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Op. Frequency	Fosc	RC type	-	2000	-	KHz
		Crystal type	400	-	4190	
Frequency deviation by voltage drop for RC oscillator	$\frac{\Delta f}{f}$	$\frac{f(3V) - f(2.4V)}{f(3V)}$	-	-	10	%
Instruction cycle time	Ti	One machine cycle	-	4/Fosc	-	S
Reset Active Width	TRAW	Fosc=32.768 KHz	1	-	-	μ S
Interrupt Active Width	TIAW	Fosc=32.768 KHz	1	-	-	μ S



10. INSTRUCTION SET TABLE

Symbol Description

ACC:	Accumulator
ACC.n:	Accumulator bit n
WR:	Working Register
WRP:	WR Page register
PAGE:	Page Register
DBKR:	Data Bank Register
ROMPR:	ROM Page Register
MR0:	Mode Register 0
MR1:	Mode Register 1
PM0:	Port Mode 0
PM1:	Port Mode 1
PM4:	Port Mode 4
PSR0:	Port Status Register 0
R:	Memory (RAM) of address R
LP:	LCD data RAM pointer
LPL:	Low nibble of the LCD data RAM pointer
LPH:	High nibble of the LCD data RAM pointer
R.n:	Memory bit n of address R
I:	Constant parameter
L:	Branch or jump address
CF:	Carry Flag
ZF:	Zero Flag
PC:	Program Counter
TM0L:	Low nibble of the Timer 0 counter
TM0H:	High nibble of the Timer 0 counter
TM1L:	Low nibble of the Timer 1 counter
TM1H:	High nibble of the Timer 1 counter
TAB0:	Look-up table address buffer 0
TAB1:	Look-up table address buffer 1
TAB2:	Look-up table address buffer 2
TAB3:	Look-up table address buffer 3
IEF.n:	Interrupt Enable Flag n
HCF.n:	HOLD mode release Condition Flag n
HEF.n:	HOLD mode release Enable Flag n
SEF.n:	STOP mode wake-up Enable Flag n



Continued

- PEF.n: Port Enable Flag n
- EVF.n: Event Flag n
- ! =: Not equal
- &: AND
- ^: OR
- EX: Exclusive OR
- ←: Transfer direction, result

[PAGE*10H+(): Contents of address PAGE(bit2, bit1, bit0)*10H+()

[P(): Contents of port P

Machine code	Mnemonic	Function	Flag affected	W/C
Arithmetic				
0001 1000 0xxx xxxx	ADD R, ACC	ACC←(R) + (ACC)	ZF, CF	1/1
0001 1100 iiiii nnnn	ADD WRn, #I	ACC←(WRn) + I	ZF, CF	1/1
0001 1001 0xxx xxxx	ADDR R, ACC	ACC, R←(R) + (ACC)	ZF, CF	1/1
0001 1101 iiiii nnnn	ADDR WRn, #I	ACC, WRn←(WRn) + I	ZF, CF	1/1
0000 1000 0xxx xxxx	ADC R, ACC	ACC←(R) + (ACC) + (CF)	ZF, CF	1/1
0000 1100 iiiii nnnn	ADC WRn, #I	ACC←(WRn) + I + (CF)	ZF, CF	1/1
0000 1001 0xxx xxxx	ADCR R, ACC	ACC, R←(R) + (ACC) + (CF)	ZF, CF	1/1
0000 1101 iiiii nnnn	ADCR WRn, #I	ACC, WRn←(WRn) + I + (CF)	ZF, CF	1/1
0010 1000 0xxx xxxx	ADU R, ACC	ACC←(R) + (ACC)	ZF	1/1
0010 1100 iiiii nnnn	ADU WRn, #I	ACC←(WRn) + I	ZF	1/1
0010 1001 0xxx xxxx	ADUR R, ACC	ACC, R←(R) + (ACC)	ZF	1/1
0010 1101 iiiii nnnn	ADUR WRn, #I	ACC, WRn←(WRn) + I	ZF	1/1
0001 1010 0xxx xxxx	SUB R, ACC	ACC←(R) - (ACC)	ZF, CF	1/1
0001 1110 iiiii nnnn	SUB WRn, #I	ACC←(WRn) - I	ZF, CF	1/1
0001 1011 0xxx xxxx	SUBR R, ACC	ACC, R←(R) - (ACC)	ZF, CF	1/1
0001 1111 iiiii nnnn	SUBR WRn, #I	ACC, WRn←(WR) - I	ZF, CF	1/1
0000 1010 0xxx xxxx	SBC R, ACC	ACC←(R) - (ACC) - (CF)	ZF, CF	1/1
0000 1110 iiiii nnnn	SBC WRn, #I	ACC←(WRn) - I - (CF)	ZF, CF	1/1
0000 1011 0xxxxxxx	SBCR R, ACC	ACC, R←(R) - (ACC) - (CF)	ZF, CF	1/1
0000 1111 iiiii nnnn	SBCR WRn, #I	ACC, WRn←(WRn) - I - (CF)	ZF, CF	1/1
0100 1010 0xxx xxxx	INC R	ACC, R←(R) + 1	ZF, CF	1/1
0100 1010 1xxx xxxx	DEC R	ACC, R←(R) - 1	ZF, CF	1/1



Instruction set, continued

Machine code	Mnemonic	Function	Flag affected	W/C
Logic				
0010 1010 0xxx xxxx	ANL R, ACC	ACC←(R) & (ACC)	ZF	1/1
0010 1110 iiiii nnnn	ANL WRn, #I	ACC←(WRn) & I	ZF	1/1
0010 1011 0xxx xxxx	ANLR R, ACC	ACC, R←(R) & (ACC)	ZF	1/1
0010 1111 iiiii nnnn	ANLR WRn, #I	ACC, WRn←(WRn) & I	ZF	1/1
0011 1010 0xxx xxxx	ORL R, ACC	ACC←(R) ^ (ACC)	ZF	1/1
0011 1110 iiiii nnnn	ORL WRn, #I	ACC←(WRn) ^ I	ZF	1/1
0011 1011 0xxx xxxx	ORLR R, ACC	ACC, R←(R) ^ (ACC)	ZF	1/1
0011 1111 iiiii nnnn	ORLR WRn, #I	ACC, WRn←(WRn) ^ I	ZF	1/1
0011 1000 0xxx xxxx	XRL R, ACC	ACC←(R) EX (ACC)	ZF	1/1
0011 1100 iiiii nnnn	XRL WRn, #I	ACC←(WRn) EX I	ZF	1/1
0011 1001 0xxx xxxx	XRLR R, ACC	ACC, R←(R) EX (ACC)	ZF	1/1
0011 1101 iiiii nnnn	XRLR WRn, #I	ACC, WRn←(WRn) EX I	ZF	1/1
Branch				
0111 0aaa aaaa aaaa	JMP L	PC12~PC0←(ROMPR)×800H+L10~L0		1/1
1000 0aaa aaaa aaaa	JB0 L	PC10~PC0←L10~L0; if ACC.0 = "1"		1/1
1001 0aaa aaaa aaaa	JB1 L	PC10~PC0←L10~L0; if ACC.1 = "1"		1/1
1010 0aaa aaaa aaaa	JB2 L	PC10~PC0←L10~L0; if ACC.2 = "1"		1/1
1011 0aaa aaaa aaaa	JB3 L	PC10~PC0←L10~L0; if ACC.3 = "1"		1/1
1110 0aaa aaaa aaaa	JZ L	PC10~PC0←L10~L0; if ACC = 0		1/1
1100 0aaa aaaa aaaa	JNZ L	PC10~PC0←L10~L0; if ACC != 0		1/1
1111 0aaa aaaa aaaa	JC L	PC10~PC0←L10~L0; if CF = "1"		1/1
1101 0aaa aaaa aaaa	JNC L	PC10~PC0←L10~L0; if CF != "1"		1/1
0100 1000 0xxx xxxx	DSKZ R	ACC, R←(R) - 1; PC ← (PC) + 2 if ACC = 0	ZF, CF	1/1
0100 1000 1xxx xxxx	DSKNZ R	ACC, R←(R) - 1; PC ← (PC) + 2 if ACC != 0	ZF, CF	1/1
1010 1000 0xxx xxxx	SKB0 R	PC ← (PC) + 2 if R.0 = "1"		1/1
1010 1000 1xxx xxxx	SKB1 R	PC ← (PC) + 2 if R.1 = "1"		1/1
1010 1001 0xxx xxxx	SKB2 R	PC ← (PC) + 2 if R.2 = "1"		1/1
1010 1001 1xxx xxxx	SKB3 R	PC ← (PC) + 2 if R.3 = "1"		1/1
Subroutine				
0110 0aaa aaaa aaaa	CALL L	STACK ← (PC)+1; PC12 ~ PC0 ← (ROMPR)×800H+L10 ~ L0		1/1
0000 0001 0000 0000	RTN	(PC)← STACK		1/1



Instruction set, continued

Machine code	Mnemonic	Function	Flag affected	W/C
Data move				
1110 1nnn nxxx xxxx	MOV WRn, R	WRn←(R)		1/1
1111 1nnn nxxx xxxx	MOV R, WRn	R←(WRn)		1/1
0110 1nnn nxxx xxxx	MOVA WRn, R	ACC, WRn←(R)	ZF	1/1
0111 1nnn nxxx xxxx	MOVA R, WRn	ACC, R←(WRn)	ZF	1/1
0101 1001 1xxx xxxx	MOV R, ACC	R←(ACC)		1/1
0100 1110 1xxx xxxx	MOV ACC, R	ACC←(R)	ZF	1/1
1011 1iii ixxx xxxx	MOV R, #I	R←I		1/1
1100 1nnn n000 qqqq	MOV WRn, @WRq	WRn←[(DBKR)×80H+(PAGE)×10H +(WRq)]		1/2
1101 1nnn n000 qqqq	MOV @WRq, WRn	[(DBKR)×80H+(PAGE)×10H +(WRq)]←WRn		1/2
1000 1100 0xxx xxxx	MOV TAB0, R	TAB0←(R)		1/1
1000 1100 1xxx xxxx	MOV TAB1, R	TAB1←(R)		1/1
1000 1110 0xxx xxxx	MOV TAB2, R	TAB2←(R)		1/1
1000 1110 1xxx xxxx	MOV TAB3, R	TAB3←(R)		1/1
1000 1101 0xxx xxxx	MOVC R	R←[(TAB3)×1000H+(TAB2)×100H+(TAB1)×10H +(TAB0)]		1/2
Input & Output				
0101 1011 0xxx xxxx	MOVA R, RA	ACC, R←[RA]	ZF	1/1
0100 1011 0xxx xxxx	MOVA R, RC	ACC, R←[RC]	ZF	1/1
0101 1010 0xxx xxxx	MOV RA, R	[RA]←(R)		1/1
1010 1100 0xxx xxxx	MOV RC, R	[RC]←(R)		1/1
0001 0010 iiiii iiiii	MOV MFP, #I	[MFP]←I		1/1
Flag & Register				
0101 1111 1xxx xxxx	MOVA R, PAGE	ACC, R←PAGE (Page Register)	ZF	1/1
0101 1110 1xxx xxxx	MOV PAGE, R	PAGE←(R)		1/1
0101 0110 1000 0iii	MOV PAGE, #I	PAGE←I		1/1
1001 1101 1xxx xxxx	MOV R, WRP	R←WRP		1/1
1001 1100 1xxx xxxx	MOV WRP, R	WRP←(R)		1/1
0011 0101 1000 liii	MOV WRP, #I	WRP←I		1/1
1001 1101 0000nnnn	MOV WRn,DBKR	WRn←DBKR		1/1
1001 1100 0000nnnn	MOV DBKR, WRn	DBKR←WRn		1/1
0011 0101 0000 0iii	MOV DBKR, #I	DBKR←I		1/1
0011 0100 0000 00ii	MOV ROMPR, #I	ROMPR←I		1/1
1000 1000 0xxx xxxx	MOV ROMPR, R	ROMPR←(R)		1/1
1000 1001 0xxx xxxx	MOV R, ROMPR	R←(ROMPR)		1/1



Instruction set, continued

Machine code	Mnemonic	Function	Flag affected	W/C
0001 0011 1000 i00i	MOV MR0, #i	MR0←i		1/1
0001 0011 0000 iiii	MOV MR1, #i	MR1←i		1/1
0101 1001 0xxx xxxx	MOVA R, CF	ACC.0, R.0←CF	ZF	1/1
0101 1000 0xxx xxxx	MOV CF, R	CF←(R.0)	CF	1/1
0100 1001 0xxx xxxx	MOVA R, HCF.L	ACC, R←HCF.0-HCF.3	ZF	1/1
0100 1001 1xxx xxxx	MOVA R, HCF.H	ACC, R←HCF.4-HCF.7	ZF	1/1
0101 0011 0000 iiii	MOV PM0, #i	Port Mode 0←i		1/1
0101 0111 0000 iiii	MOV PM1, #i	Port Mode 1←i		1/1
0011 0111 0000 iiii	MOV PM4, #i	Port Mode 4←i		1/1
0100 0000 i00i liii	CLR EVF, #i	Clear Event Flag if In = 1		1/1
0101 1101 0xxx xxxx	MOVA R, EVFL	R←EVF.0 - EVF.3		1/1
0101 1101 1xxx xxxx	MOVA R, EVFH	R←EVF.4 - EVF.7		1/1
0100 0001 i00i iiii	MOV HEF, #i	Set/Reset HOLD mode release Enable Flag		1/1
0101 0001 i00i iiii	MOV IEF, #i	Set/Reset Interrupt Enable Flag		1/1
0100 0011 0000 iiii	MOV PEF, #i	Set/Reset Port Enable Flag		1/1
0101 0010 0000 iiii	MOV SEF, #i	Set/Reset STOP mode wake-up Enable Flag for RC port		1/1
0101 0100 0000 iiii	MOV SCR, #i	SCR←i		1/1
0100 1111 0xxx xxxx	MOVA R, PSR0	ACC, R←Port Status Register 0	ZF	1/1
0100 0010 0000 0000	CLR PSR0	Clear Port Status Register 0		1/1
0101 0000 0100 0000	SET CF	Set Carry Flag	CF	1/1
0101 0000 0000 0000	CLR CF	Clear Carry Flag	CF	1/1
0001 0111 0000 0000	CLR DIVR0	Clear the last 4-bit of the Divider 0		1/1
0101 0101 1000 0000	CLR DIVR1	Clear the last 4-bit of the Divider 1		1/1
0001 0111 1000 0000	CLR WDT	Clear WatchDog Timer		1/1
Shift & Rotate				
0100 1101 0xxx xxxx	SHRC R	ACC.n, R.n←(R.n+1); ACC.3, R.3←0; CF←R.0	ZF, CF	1/1
0100 1101 1xxx xxxx	RRC R	ACC.n, R.n←(R.n+1); ACC.3, R.3←CF; CF←R.0	ZF, CF	1/1
0100 1100 0xxx xxxx	SHLC R	ACC.n, R.n←(R.n-1); ACC.0, R.0←0; CF←R.3	ZF, CF	1/1
0100 1100 1xxx xxxx	RLC R	ACC.n, R.n←(R.n-1); ACC.0, R.0←CF; CF←R.3	ZF, CF	1/1



Instruction set, continued

Machine code	Mnemonic	Function	Flag affected	W/C
LCD				
1001 1000 0xxx xxxx	MOV LP0, R	LP0←(R)		1/1
1001 1000 1xxx xxxx	MOV LP1, R	LP1←(R)		1/1
1001 1010 0xxx xxxx	MOV @LP, R	[(LP1)×10H+(LP0)]←(R)		1/1
1001 1011 0xxx xxxx	MOV R, @LP	R← [(LP1) ×10H+(LP0)]		1/1
0000 0010 0000 0000	LCDON	LCD ON		1/1
0000 0010 1000 0000	LCDOFF	LCD OFF		1/1
0000 0011 0000 0111	MOV LCDCN, #i	Select LCD output as DC output type		1/1
PWM				
1000 1010 1xxx xxxx	MOV PWM, R	PWM←(R)×10H + (ACC)		1/1
0011 0110 0000 11ii	MOV BUZCR, #i	BUZCR← i		1/1
1000 1010 0xxx xxxx	MOV BUZCR, R	BUZCR← (R)		1/1
1000 1011 0xxx xxxx	MOV R, BUZCR	R← (BUZCR)		1/1
0011 0110 1100 0000	EN BUZ	Enable buzzer output		1/1
0011 0110 1000 0000	DIS BUZ	Disable buzzer output		1/1
Timer				
1010 1010 0xxx xxxx	MOV TM0L, R	TM0L←(R)		1/1
1010 1010 1xxx xxxx	MOV TM0H, R	TM0H←(R)		1/1
1010 1011 0xxx xxxx	MOV TM1L, R	TM1L←(R)		1/1
1010 1011 1xxx xxxx	MOV TM1H, R	TM1H←(R)		1/1
Other				
0000 0000 1000 0000	HOLD	Enter Hold mode		1/1
0000 0000 1100 0000	STOP	Enter Stop mode		1/1
0000 0000 0000 0000	NOP	No operation		1/1
0101 0000 1100 0000	EN INT	Enable interrupt function		1/1
0101 0000 1000 0000	DIS INT	Disable interrupt function		1/1