W53300/W53320

Winbond Electronics Corp.

16" Voice / Melody / LCD Controller (ViewTalk™ Series)

GENERAL DESCRIPTION

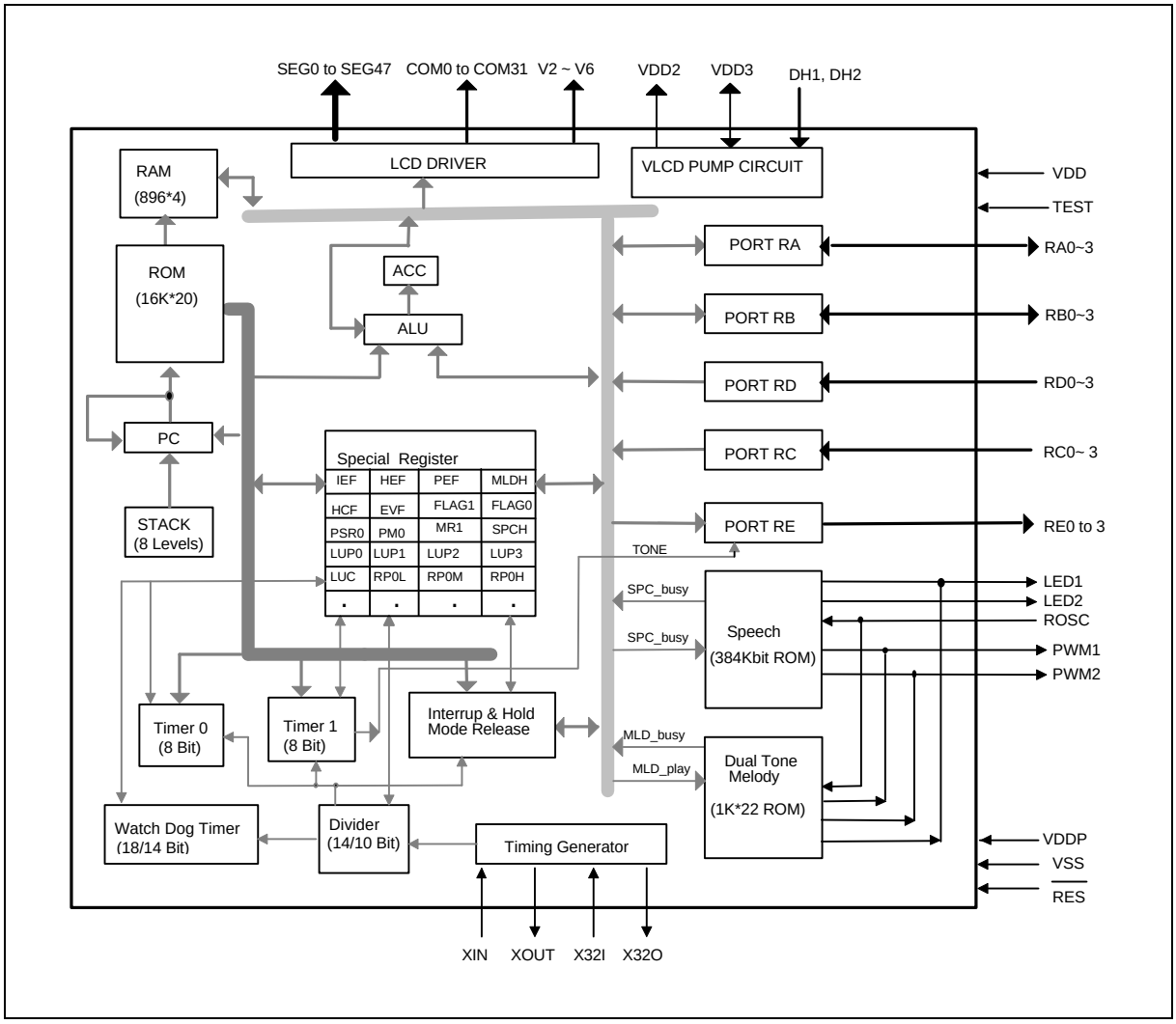
The W53300/W53320 is a high-performance 4-bit microcontroller (μC) with built-in speech, melody and 32*48 LCD driver which includes internal pump circuit. The 4-bit uc core contains dual clock source, 4-bit ALU, two 8-bit timers, one divider, 20 pin input or output, 7 interrupt sources and 8-level subroutine nesting for interrupt applications. Speech unit can be implemented with Winbond 16-sec Power Speech using ADPCM algorithm. Melody unit provides dual tone output and can store up to 1k notes. Power reduction mode is also built in to minimize power dissipation. It is ideal for games, educational toys, remote controllers, watches, clocks and other application products which incorporate both LCD display and melody.

FEATURES

- Operating voltage: 2.4~5.5Volt
- Dual clock operating system
  - RC/Crystal (400 KHz to 4 MHz) for main clock
  - 32.768 KHz crystal oscillation circuit for sub-oscillator
- Memory
  - 16k × 20 bit program ROM
  - 512 × 4 bit (W53320) / 704 × 4 bit (W53300) general data RAM
  - 384 × 4 bit (W53320) / 192 × 4 bit (W53300) LCD data RAM
- 20 input/output pins
  - Ports for input only: 2 ports/8 pins
  - Input/output ports: 2 ports/8 pins
  - Port for output only: 1 port /4 pins
- Power-down mode
  - Hold function: no operation (except for oscillator)
- Seven types of interrupts
  - Five internal interrupts (Divider ,Timer 0, Timer 1, Speech, Melody)
  - Two external interrupts (Port RC, Port RD)
- One built-in 14-bit clock frequency divider circuit
- Two built-in 8-bit programmable countdown timers
  - Timer 0: one of two internal clock frequencies (FOSC/4 or FOSC/1024) can be selected
  - Timer 1: built-in auto-reload function includes internal timer, external event counter from RC.0 or TONE output function (can be used as IR carrier output if main clock is 455kHz)
- Built-in 18/14-bit watchdog timer for system reset by mask code option
- Over 500 powerful instruction sets
- 8-level subroutine (including interrupt) nesting

- LCD driver output
  - 32 common × 48 segment (W53320), 16 common × 48 segment (W53300)
  - 1/16 or 1/32 duty, 1/5 or 1/7 bias, internal pump circuit option by special register
- Speech function
  - Provides 384 kbits dedicated speech ROM
  - Direct driving output for speaker
  - Maximum 256 sections available
- Melody function
  - Provides 22 kbits dedicated melody ROM
  - Provides 6 kinds of beat, 16 kinds of tempo, and pitch range from G3# to C7
  - Tremolo, triple frequency and 3 kinds of percussion available
  - Direct driving output for speaker
  - Maximum 32 scores available
- Mix speech with melody available
- Multi-engine controller
- PWM output current option
- Chip On Board available

BLOCK DIAGRAM



PIN DESCRIPTION

SYMBOL	I/O	FUNCTION
XIN	I	Input pin for oscillator. It can be connected to crystal, or can connect a resistor to VDD to generate main system clock. Oscillator can be stopped when SCR.1 is set to logic 1.
XOUT	O	Output pin for oscillator which is connected to another crystal pin.
X32I	I	32.768 KHz crystal input pin.
X32O	O	32.768 KHz crystal output pin.
RA0 ~ RA3	I/O	General Input/Output port specified by PM1 register. If output mode is selected, PM0 register can be used to specify CMOS/NMOS driving capability option. Initial state is input mode.
RB0 ~ RB3	I/O	General Input/Output port specified by PM2 register. If output mode is selected, PM0 register can be used to specify CMOS/NMOS driving capability option. Initial state is input mode.
RC0 ~ RC3	I	4-bit schmitt input port with internal pull high option specified by PM0 register. RC0 can be used as clock source for Timer 1. Each pin has an independent interrupt capability specified by PEFL special register.
RD0 ~ RD3	I	4-bit schmitt input port with internal pull high option specified by PM0 register. Each pin has an independent interrupt capability specified by PEFH special register.
RE0~ RE3/TONE	O	Output port only. RE3 may use as TONE if bit 0 of MR0 special register is set to logic 1.
$\overline{\text{RES}}$	I	System reset pin with internal pull-high resistor is active low.
TEST	I	Test pin. Connected to low for normal use.
ROSC	I	Connects resistor to VDD to generate speech or melody clock source.
VDDP	I	Power source for PWM output.
LED1	O	Synchronous LED1 output while speech play/melody is active.
LED2	O	Synchronous LED2 output only while speech play is active.
PWM1	O	Speaker direct driving output 1 while speech or melody is active.
PWM2	O	Speaker direct driving output 2 while speech or melody is active.
SEG0–SEG47	O	LCD segment output pins.
COM0–COM31	O	LCD common signal output pins. The LCD alternating frequency is fixed at 64Hz.
DH1, DH2	I	Connection terminals for voltage doubler capacitor.
VDD2	O	Connects a 1uF capacitor to VSS to double VDD voltage output if triple pump option is enabled. Otherwise, VDD2 connects to VDD directly if double pump option is enabled.

VDD3	O/I	An output if internal pump circuit is enabled. It connects a 1uF capacitor to VSS. Triple VDD voltage will be output if triple pump option is enabled. Otherwise, double VDD voltage will be output if double pump option is enabled. An input if internal pump voltage is disabled.
V2 ~ V5	O	LCD COM/SEG output driving voltage. If internal shunt resistor is disabled, external resistors need to be supplied to V2, V3, V4, V5. A capacitor is suggestted for stable LCD voltage level.
V6	I	External variable resistor connects between VDD3 and V6 to adjust maximun LCD output voltage level.
VDD	I	Microcontroller Positive power supply (+).
VSS1	I	Negative power supply (-).
VSS2	I	Negative power supply (-).

FUNCTIONAL DESCRIPTION

There are four main units in W53300/W53320 : 4 bit uC, Power Speech, dual-tone melody and 16com/32 com * 48 seg LCD driver. The 4 bit uC is modified from Winbond W741C260 with many features enhanced, such as larger ROM and RAM space, addressing capability, more instruction sets, 7 interrupt sources, speech control, capability for playing melody directly to drive speaker, and so on. The following sections, Parts A, B, and C will explain the functions in detail.

PART A: UC FUNCTION

Program Counter (PC)

Organized as a 14-bit binary counter (PC0 to PC13), the program counter generates the addresses of the 16K x20 on-chip ROM containing the program instructions. When jump instructions, subroutine calls, interrupts, or initial reset conditions are executed, the address corresponding to the next instruction will be loaded into the program counter. Table 1 lists the formats used.

ITEM	ADDRESS	INTERRUPT PRIORITY
Initial Reset	0000H	-
INT 0 (DIV)	0004H	1st
INT 1 (TM 0)	0008H	2nd
INT 2 (RC)	000CH	3rd
INT 3 (RD)	0010H	4th
INT 4 (Reserved)	-	-
INT 5 (SPEECH)	0018H	5th
INT 6 (MELODY)	001CH	6th
INT 7 (TM 1)	0020H	7th
JP Instruction	XXXXH	-
Subroutine Call	XXXXH	-

Table 1: Interrup Address Assignment & Priority

Stack Register (STACK)

The stack register is organized as 14bits × 8 levels (first-in, last-out). When a subroutine call or an interrupt is executed, the program counter will be pushed onto the stack register automatically. At the end of a subroutine call or an interrupt service subroutine, the *RTN* instruction must be executed to pop the content of the stack register into the program counter. When the stack register is pushed over the eighth level, the content of the first level will be lost. In other words, a maximum of 8 subroutine nestings are allowed in the stack register.

Program Memory (ROM)

1. Architecture

The read-only memory (ROM) of size 16K x20 bit is used to store program codes addressing PC from 0000H to 3FFFH. Locations 0000H through 0020H are reserved for interrupt service as shown in Figure 1. All instruction sets are one word, one cycle. Look-up table function is provided to access ROM code of all 16k ROM spaces. The program memory map is shown in Figure 1.

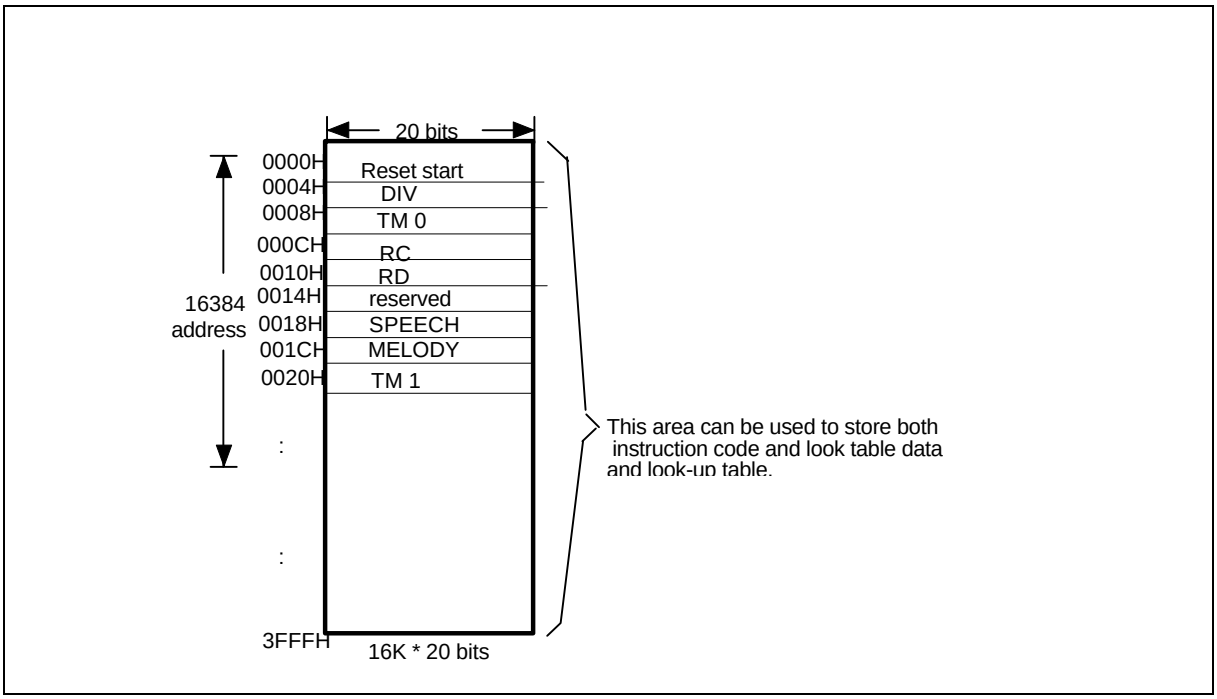


Figure 1. Program Memory Map

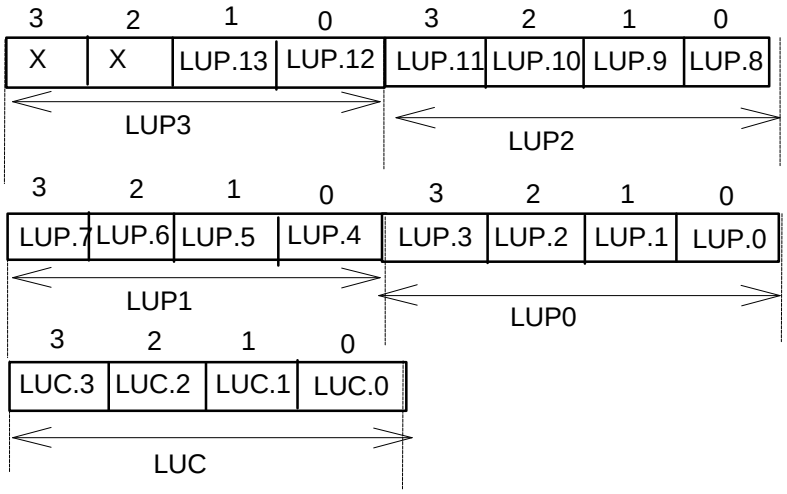
2. Look-Up Table Pointer Register(LUP3, LUP2, LUP1, LUP0 and LUC)

The LUPC (Look-up table address Pointer Counter) is used to access data in the 16K ROM space. It includes 5 consecutive registers, LUP3 (Look-Up table Pointer), LUP2, LUP1, LUP0, and LUC (Look-Up table data Counter). LUP3, LUP2, LUP1 and LUP0 together store the 14-bit ROM address for accessing data in the 16k-word ROM, and every single 20-bit ROM word is separated into 5 nibbles. As described in the following equation,

LUPC = LUPC.13~LUPC.0 + LUC.3~LUC.0

LUPC.13~LUPC.0 mapping from 0000H to 3FFFH is used as word address of 16K ROM. LUC is used to determine which nibble of the word data is accessed, and counts from 0 to 4 cyclically. The instruction *MOV LUPn, ACC* can write initial address pointer of look-up table into LUPC, and resets LUC register to 0.

LUPC is increased by 1 when LUC is increased by 1, and LUP0 is increased by 1 when LUC counts from 4 to 0. LUP1 is increased by 1 when LUP0 is counted from 0xF to 0. LUP3 and LUP2 follow the same rule as LUP1. The LUC will be increased by 1 automatically when symbol @LUPC++ is used. Registers LUP3~LUP0 can be read/write by user, but LUC register is read only. At initial reset, all registers are initialized to 0000B.



Data Memory (RAM)

1. Architecture

The static data memory (RAM) is arranged as (512+384) × 4 bits. Data memory can be addressed directly or indirectly. The mapping of the data memory, using W53320 as example, is shown in Figure 2. The first 512 nibbles RAM mapping from 000H to 1FFH are allocated for general data memory. Data memory from 200H to 37FH can be allocated as an LCD dedicated pattern data memory as mapping in Table 5, or as a general data memory, since they both have the same addressing capability. There are two data memory address points, RP0 (RP0H+RP0M+RP0L) and RP1 (RP1H+RP1M+RP1L), that can be operated by indirect addressing instructions, such as *MOV ACC, @RP0* and *MOV @RP1, @RP0*, to move data between different data memory ranges and ACC. Instructions for moving data between ROM and RAM are also provided, such as *MOV @RP0, @LUPC*, so that users can move look-up table data in ROM to general RAM easily. The instruction *MOV @RP0++, @LUPC++* is also available for automatically incrementing point counter by 1 once the instruction is completed. Please refer to instruction sets description for more details.

The first 16 addresses (00H to 3FH) in the data memory are known as the page 0 working registers. Only working registers can directly operate with immediate data. A special register, WRPAGE, with ranges from 0H to 0DH, is used for selecting working register page.

2. Working Register Page (WRPAGE with SR=30H)

WRPAGE, a 4-bit special register, that counts from 0 to 0DH, divides 896 nibbles RAM into 14 pages. Every page consists of 64 nibbles. The bit descriptions are as follow:

	3	2	1	0
WRPAGE	R/W	R/W	R/W	R/W

Bit 3~0: 0000~1011 Page 0 to Page 0DH

Bit3~0: 0000~1111 is inhibited. All bits can be read/write. At initial reset, the WRPAGE is initiated to

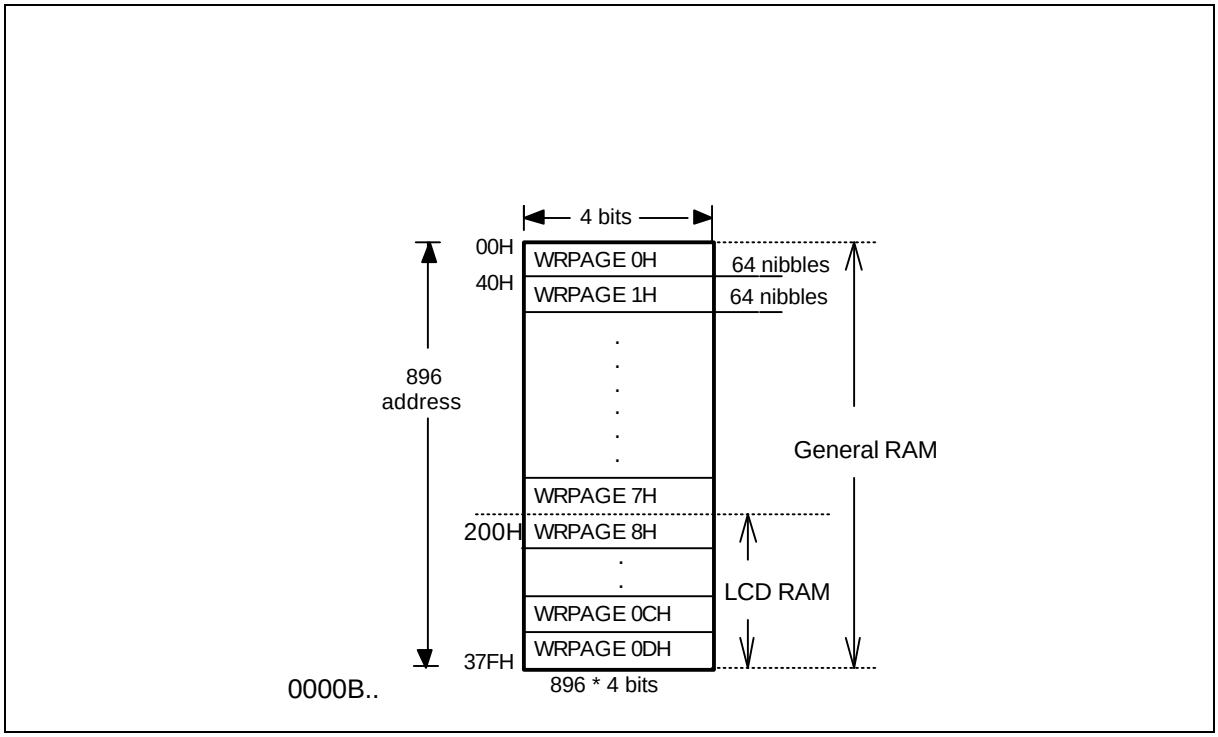


Figure 2. Data Memory Mapping of W53320

3. RAM Point Register (RP0L, RP0M, RP0H, RP1L, RP1M, RP1L)

There are two RAM points 0 and 1 that can be utilized for easily accessing data, either by direct or by indirect addressing. RAM Point 0 (RP0) is configured as 10-bit format (RP0.9~RP0.0) represented by 3 special registers, RP0L, RP0M and RP0H. RAM Point 1 (RP1) is configured as the same structure as RP0, thus are represented by RP1L, RP1M and RP1H.

	11	10	9	8	7	6	5	4	3	2	1	0
RP0	X	X	RP0.9	RP0.8	RP0.7	RP0.6	RP0.5	RP0.4	RP0.3	RP0.2	RP0.1	RP0.0
	RP0H				RP0M				RP0L			
	11	10	9	8	7	6	5	4	3	2	1	0
RP1	X	X	RP1.9	RP1.8	RP1.7	RP1.6	RP1.5	RP1.4	RP1.3	RP1.2	RP1.1	RP1.0
	RP1H				RP1M				RP1L			

All bits of RPn are allowed for read/write operation. At initial reset, all RPn are cleared to 0000B.

Special Register and Special Register Pair (SR & SRP)

Some special registers are formatted as 4-bit in length, as shown in Table 2. Chip operating condition depends on the value of the special register. Commands such as *MOV SR, #I*, *CLR SR* and *SET SR* write the appropiate value to control chip operating state. Some special registers, such as HEF, IEF and HCF, can write 8-bit immediate data simultaneously by using commands associated with Special Register Pair (SRP) like *MOV SRP, #I*. All special register functions will be described in detail when the close relation function is introduced.

SRP	SR	SR Symbol	Function	Bit 3 ~ 0 assignment
	00	-----		
	01	-----		
	02	TM0L(w)	low nibble of Timer 0	TM0.3~TM0.0
	03	TM0H(w)	high nibble of Timer 0	TM0.7~TM0.4
	04	TM1L(w)	low nibble of Timer 1	TM1.3~TM1.0
	05	TM1H(w)	high nibble of Timer 1	TM1.7~TM1.4
	06	TMC1L(r)	low nibble of Timer 1	TM1.3~TM1.0
	07	TMC1H(r)	high nibble of Timer 1	TM1.7~TM1.4
	08	-----		
	09	-----		
	0A	EVFL(r,c)	Event Flag (set by chip hardware	RD,RC,TM0,DIV
	0B	EVFH(r,c)	if interrupt is occurred)	TM1,SPEECH,MELODY,X
06H	0C	HEFL(r/w,s/c)	Hold mode release Enable Flag	RD,RC,TM0,DIV
	0D	HEFH(r/w,s/c)		TM1,SPEECH,MELODY,X
07H	0E	IEFL(r/w,s/c)	Interrup Enable Flag	RD,RC,TM0,DIV
	0F	IEFH(r/w,s/c)		TM1,SPEECH,MELODY,X
	10	HCFL(r)	Hold mode release Condition Flag	RD,RC,TM0,DIV
	11	HCFH(r)	(set by H/W if hold mode is released)	TM1,SPEECH,MELODY,X
	12	-----		
	13	-----		
	14	PEFL(r/w,s/c)	Port Enable Flag for hold mode	RC.3, RC.2, RC.1, RC.0
	15	PEFH(r/w,s/c)	release or interrupt function	RD.3, RD.2, RD.1, RD.0
	16	RP0L(r/w)	RAM address Pointer 0 Low nibble	RP0.3~RP0.0
	17	RP0M(r/w)	RAM address Pointer 0 Middle nibble	RP0.7~RP0.4
	18	RP1L(r/w)	RAM address Pointer 1 Low nibble	RP1.3~RP1.0
	19	RP1M(r/w)	RAM address Pointer 1 Middle nibble	RP1.7~RP1.4
	1A	RP0H(r/w)	RAM address Pointer 0 High nibble	X,X, RP0.9,RP0.8
	1B	RP1H(r/w)	RAM address Pointer 1 High nibble	X,X, RP1.9,RP1.8
	1C	MLDL(w)	MeLoDy score address Low nibble	MLD.3~MLD.0
	1D	MLDH(w)	MeLoDy score address High nibble	MLED1,MLED0, OSB,MLD.4
	1E	SPCL(w)	SPeeCh section address Low nibble	SPC.3~SPC.0
	1F	SPCH(w)	SPeeCh section address High nibble	SPC.7~SPC.4

	20	CF(r,s/c)	Carrier Flag	X,X,X,CF
	21	-----		
	22	FLAG0(r/w,s/c)	melody/speech busy and play flag	MLD_busy,SPC_busy,MLD_play,SPC_play
	23	FLAG1(c)	reset flag for Divider/WatchDog	X,DIVR,WDTR,X
	24	-----		
	25	-----		
	26	MR2(r/w,s/c)	special register for interrupt enable	X,X,X,INTEN
	27	MR3(w)	Mode register 3 for pump voltage and PWM current	VLCDEXT,X,CUR1,CUR0
	28	MR0(r/w,s/c)	Mode Register 0 for timer	TM0EN,TM1EN,LCDEN,TONE
	29	MR1(w)	Mode Register 1 for timer	WDTCK,TM0CK,TM1SR,TM1CK
	2A	LCDM1(w)	LCD Mode register 1	COM32B,BIAS7B,PUPV3B,INTSRB
	2B	SCR(r/w,s/c)	System Control Register	DIV5MB,FMRCB,FMEN,F32IN
	2C	LUP0(r/w)	Look UP table address pointer 1st nibble	LUPC.3~LUPC.0
	2D	LUP1(r/w)	Look UP table address pointer 2nd nibble	LUPC.7~LUPC.4
	2E	LUP2(r/w)	Look UP table address pointer 3th nibble	LUPC.11~LUPC.8
	2F	LUP3(r/w)	Look UP table address pointer 4th nibble	X,X,LUPC.13,LUPC.12
	30	WRPAGE(r/w)	Working Register PAGE register	0000~1101H
	31	LUC(r)	LUPC nibble counter	0000~0100H
	32	PM0(r/w,s/c)	Port Mode Register 0	RD_PH,RC_PH,RB_NM,RA_NM
	33	-----		
	34	PSR0(r,clr-all)	Port RC Status change Register	RC3EG~RC0EG
	35	PSR1(r,clr-all)	Port RD Status change Register	RD3EG~RD0EG
	36	PM1(r/w,s/c)	Port RA I/O Mode select Register	RA3IN~RA0IN
	37	PM2(r/w,s/c)	Port RB I/O Mode select Register	RB3IN~RB0IN
	38	PORTA(r/w)	PORT data of RA	RA3~RA0
	39	PORTB(r/w)	PORT data of RB	RB3~RB0
	3A	PORTC(r)	PORT data of RC	RC3~RC0
	3B	PORTD(r)	PORT data of RD	RD3~RD0
	3C	PORTE(w)	PORT data of RE	RE3~RE0
	3D~3F	------		

Note 1: "r, w, s, c " means "read, write, set, and clear" (respectively)
Note 2: "clr-all " means all 4 bits will be cleared simultaneously.
Note 3: X means don't care bit

Table 2: Special Register address mapping

Accumulator (ACC)

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

Arithmetic and Logic Unit (ALU)

ALU is a circuit which performs arithmetic and logic operations. It 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, JNB0, JNB1, JNB2, JNB3, SKNB0, SKNB1, SKNB2, SKNB3
- Shift operations: SHRC, RRC, SHLC, RLC
- Binary additions/subtractions: ADDC, ADD, ADDU, SUB, SUBB, DEC, INC

After any of the above instructions is executed, the status of the carry flag (CF) will be affected. The CF will be stored into internal register, and read out by *MOVA R, CF* or *MOV CF, R*.

Carry Flag Register (CF with SR=20H)

The CF register stores only the CF signal state. Please refer to instruction sets for CF signal status.

	3	2	1	0
CF	X	X	X	CF

Clock Generator

W53300/W53320 provides two oscillation circuits: main-oscillator (FM) and sub-oscillator (FS). The SCR (System Control Register) is incorporated to specify clock operation condition. Either main-oscillator or sub-clock can be the system clock (FOSC) by setting F32IN option bit (bit 0 of SCR special register). The main-oscillator starts oscillation when FMEN (bit 1 of SCR) is set to 1. Main-oscillator can select crystal or RC oscillation through external connections by setting special register FMRCB bit (bit 2 of SCR). If a crystal oscillator is chosen, a crystal or a ceramic resonator must be connected between XIN and XOUT, and a capacitor must be connected if accurate frequency is required. The oscillator supports clock frequency from 400KHz to 4 MHz. A 455 KHz ceramic resonator should be selected if an IR carrier output from RE3/TONE is needed. On the other hand, if the RC oscillator is selected, a resistor must be connected between XIN and VDD. The sub-oscillator must be connected to a 32.768 KHz crystal between X32I and X32O. The connection is shown in Figure 3. One machine cycle consists of a four-state system clock sequence, and can run up to 1 µS with a 4 MHz system clock.

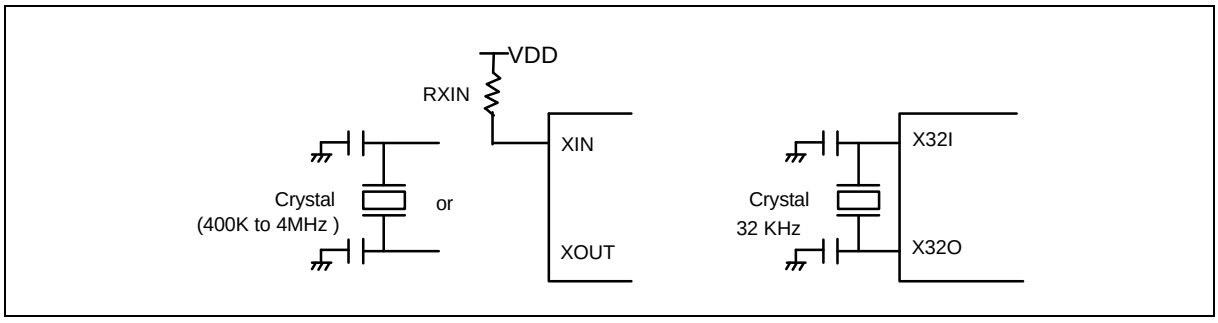


Figure 3. Oscillator Configuration

Dual-clock operation

This operation mode is dual-clock mode when FMEN bit is enabled. LCD operation clock source must be the sub-oscillator clock (32768 Hz) only. Sub-clock is used as system clock at initial reset, for example, power on or if reset pin is actived, since SCR special register is set to 0001B. However, if high frequency clock is required, that is, where main-clock should be selected as the system clock, a proper value should to be written to F32IN.

The exchange of the main-clock and sub-clock operation is performed by resetting or setting F32IN. If F32IN is reset to 0, the clock source of the system clock generator is the main-oscillator clock; if the F32IN is set to 1, the clock source of the system clock generator is the sub-oscillator clock. The main-oscillator will stop oscillating when FMEN is reset to 0.

Please attend to the setting or resetting of SCR register when:

1. XX10B → XX01B: Disabling the main-oscillator (FM) should not be done at the same time as changing the system clock source (FOSC) from FM to FS. The FOSC should be changed from FM to FS before the main-oscillator (FM) is disabled. The correct seqence is: XX10B→XX11B→XX01B.
2. XX01B → XX10B: Enabling the main-oscillator (FM) should not be done at the same time as changing the system clock source (FOSC) from FS into FM. The main-oscillator (FM) should be enabled before a delay subroutine is called for stabilizing the main-oscillator; the FOSC can then be changed from FS into FM. The correct sequence is therefore XX01B→XX11B→delay subroutine→XX10B. The suggested delay for FM is 3.5 mS for 455 KHz ceramic resonator and 0.8 mS for 4 MHz crystal.

Remember that bits FMEN and F32IN of SCR register can not be reset (XX00B) at the same time, as this will cause a system shutdown. The organization of the dual-clock operation mode is shown below.

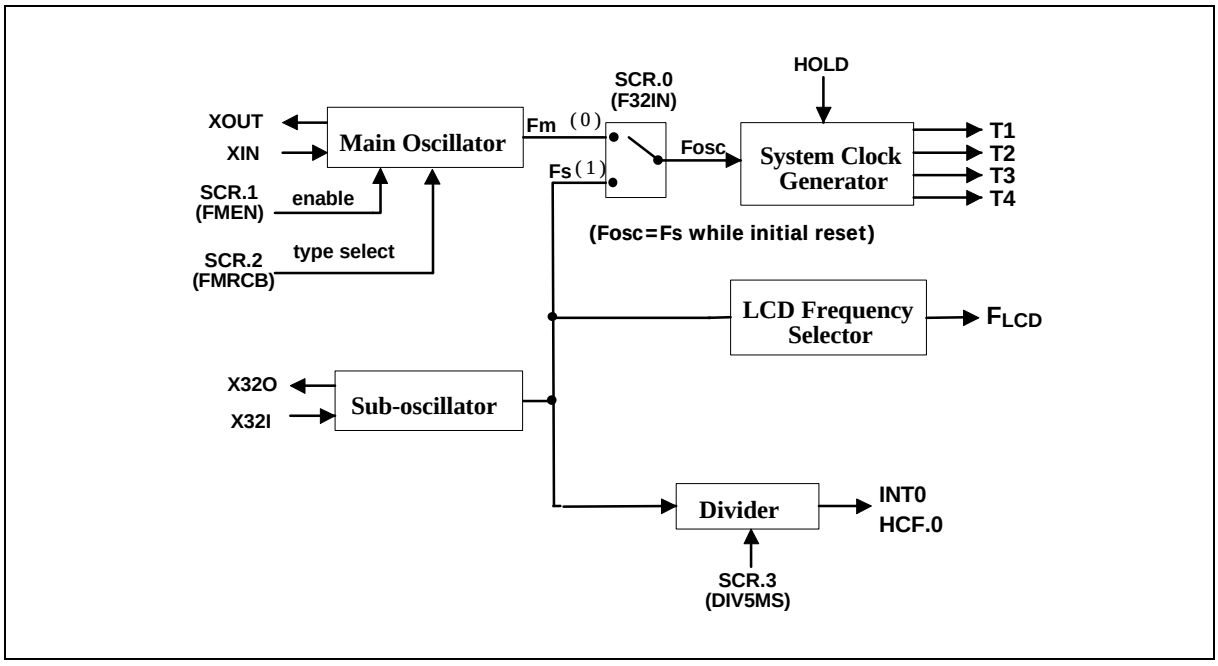


Figure 4. The Dual Clock Operation Mode Control Diagram

System Control Register (SCR with SR=2BH)

SCR is a 4 bit register (SCR.3~SCR.0). The functions of the bits are shown below.

	3	2	1	0
SCR	DIV5MB	FMRCB	FMEN	F32IN

F32IN =0: F_M is used as F_{OSC} input
=1: F_S is used as F_{OSC} input
FMEN =0: F_M oscillation is disabled
=1: F_M oscillation is enabled
FMRCB =0: F_M type is RC oscillation
=1: F_M type is XTAL oscillation
DIV5MB =0 : Divider overflows periodically every 0.5 sec.
=1: Divider overflows periodically every 0.125 sec.

All bits can be read/write, set/clear by user. At initial reset, the SCR is initiated to 0001B.

Divider

There is one divider, a 14-bit/12bit binary up-counter designed to generate periodic interrupts. The divider is incremented by each clock (Fs). When an overflow occurrs, the divider event flag is set (EVF.0 = 1). The interrupt is executed if the divider interrupt enable flag is set (IEF.0 = 1), and the hold state is terminated when the hold release enable flag is set (HEF.0 = 1). There are two time periods (500mS & 125 mS) that can be selected by DIV5MB bit. DIV5MB is reset to 0 (default) for selecting 500 mS period time, and is set to 1 for 125 mS.

Watchdog Timer (WDT)

The watchdog timer (WDT) is incorporated to prevent the program from being affected by unexpected errors. The WDT function can be enabled by mask option, and the clock source can be chosen between Fosc/1024 and Fosc/16384 by WDTCK (bit 3 of MR1 special register). At initial reset, the WDTCK derives from F_{OSC}/1024. WDT overflow occurrs when chip operation is out of control. In this case, the whole system will be reset. The content of the WDT can be cleared by the instruction *CLR FLAG1, #0010B* or *CLR WDT*. The input clock of the WDT can be switched to F_{OSC}/16384 (or F_{OSC}/1024) when WDTCK is written 1 (or 0). In normal operation, the application program must reset WDT (by CLR WDT) before it overflows. The WDT minimum overflow period is 500mS when the system clock (F_{OSC}) is 32 KHz and WDT clock input is F_{OSC}/1024. The organization of the watchdog timer is shown in Figure 5.

FLAG1 Register (FLAG1 with SR=23H)

Divider and watchdog counter can be reset by *CLR FLAG1, #I* instruction. Instructions *CLR DIV* and *CLR WDT* can be used to clear DIVR bit and WDTR bit respectively. The bit descriptions are as follows.

	3	2	1	0
FLAG1	X	DIVR	WDTR	X

DIVR =0 no influence

=1 Divider counter is clear

WDTR=0 no influence

=1 Watchdog timer is clear

X means don't care value

All bits can be cleared only. At initial reset, FLAG1 is set to 0000B.

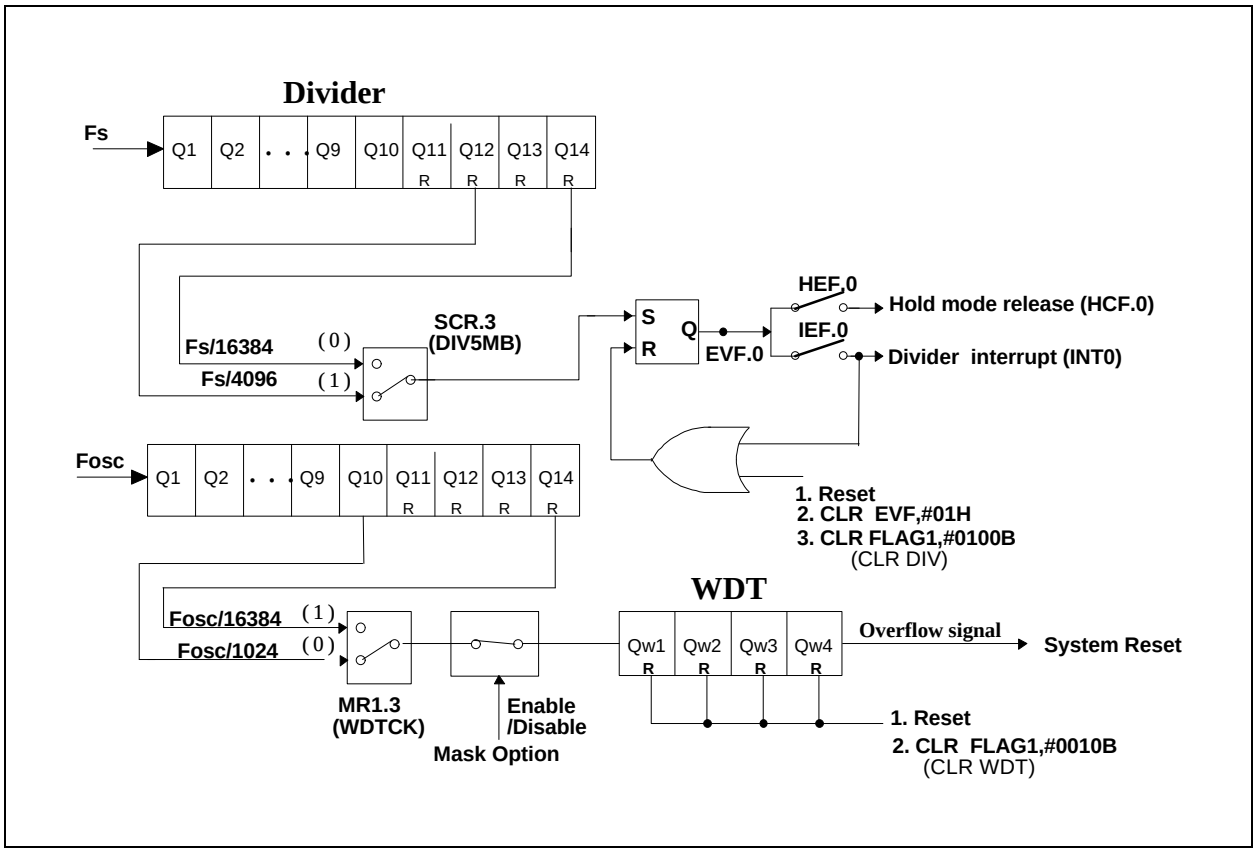


Figure 5. Organization of Divider and Watchdog Timer

Timer/Counter

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* instruction. If the TM0 is down-counting, executing *MOV TM0L(TM0H),R* instruction will stop TM0, reset TM0EN option bit (bit 3 of MR0 special register) to 0, and load specified value to TM0. When TM0EN is set to 1, the event flag 1 (EVF.1) is reset and the TM0 starts to count. Timer 0 stops operating and generates an underflow (EVF.1 = 1) while it decreases to FFH. The interrupt is executed if the Timer 0 interrupt enable flag is set (IEF.1 = 1); the hold state is terminated if the hold release enable flag 1 is set (HEF.1 = 1). The Timer 0 clock input can select either FOSC/1024 or FOSC/4 by setting TM1CK (bit 2 of MR1 special register) to 1 or resetting TM1CK to 0. The organization of Timer 0 is shown in Figure 6.

Example:

If the Timer 0 clock input is FOSC/4 :

Desired Time 0 interval = (preset value +1) × 4 × 1/FOSC

If the Timer 0 clock input is FOSC/1024 :

Desired Time 0 interval = (preset value +1) × 1024 × 1/FOSC

Preset value: Decimal number of Timer 0 preset value

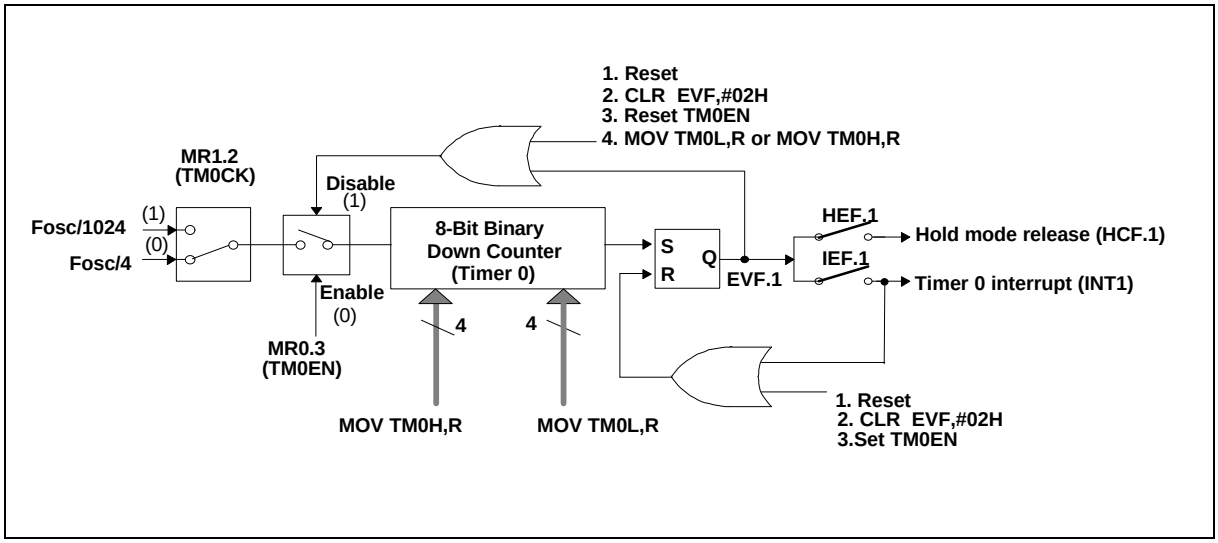


Figure 6. Organization of Timer 0

2. Timer 1 (TM1)

Timer 1 (TM1) is also a programmable 8-bit binary down counter, as shown in Figure 7. Timer 1 can be used as a counter to count external events or to output an arbitrary frequency to the RE3/TONE pin. The input clock source of Timer 1 can be internal or external, determined by TM1SR option bit (bit 1 of MR1 special register). The internal clock can be FOSC/64 or FOSC, selected by TM1CK option bit (bit 0 of MR1 special register). At initial reset, the Timer 1 clock input is FOSC. An external clock is attached via RC0 input pin. If an external clock is selected as the clock source of Timer 1, the content of Timer 1 is decreased by 1 at the falling edge of RC.0. Executing *MOV TM1L, R* or *MOV TM1H,R*

instruction will load specified data to the auto-reload buffer and disable TM1 down-counting (i.e. TM1EN is reset to 0). If TM1EN is set to 1, the contents of the auto-reload buffer will be loaded into the TM1 down counter to start counting and reset the event flag 7 (EVF.7 = 0). When the timer decreases to FFH, it will generate an underflow (EVF.7 = 1) and will be auto-reloaded with the specified data, after which it will continue to count down. An interrupt is executed if the interrupt enable flag 7 is set to 1 (IEF.7 = 1), and the hold state is terminated if the hold mode release enable flag 7 is set to 1 (HEF.7 = 1).

The specified frequency of Timer 1 can also be output to the RE3/TONE pin by TONE option bit (bit 0 of MR0).

Example:

If the Timer 1 clock input is F_T :

Desired Timer 1 interval = (preset value +1) / F_T

Desired frequency for RE3/TONE output pin = $F_T \div$ (preset value + 1) $\div$ 2 (Hz)

Preset value: Decimal number of Timer 1 preset value

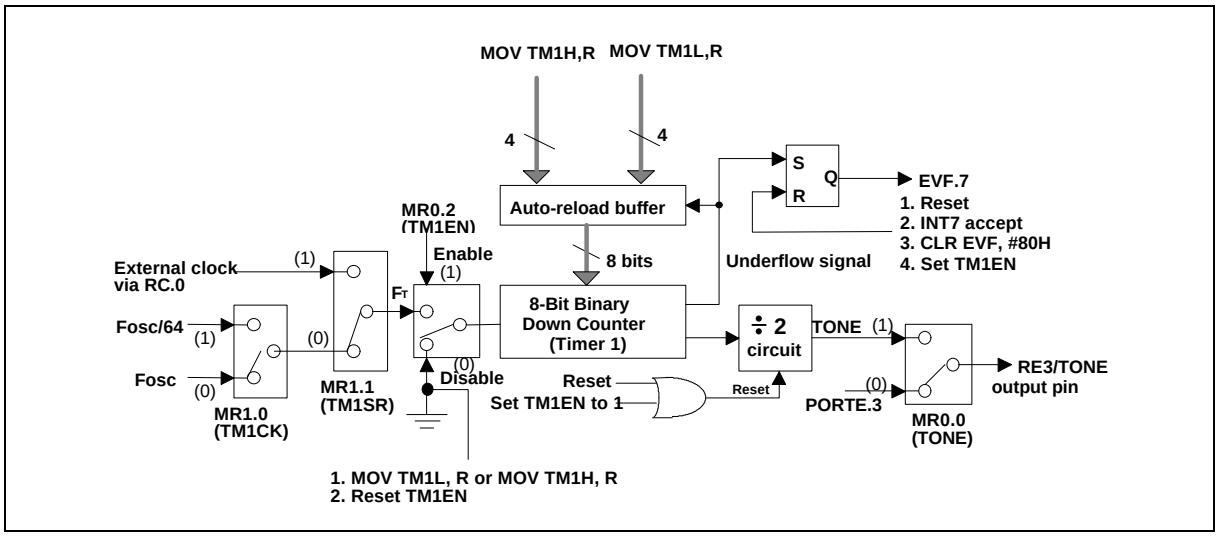


Figure 7. Organization of Timer 1

For example, when F_T equals 455kHz, depending on the preset value of TM1, the RE3/TONE pin will output a single tone signal in the tone frequency range from 889Hz to 227.5kHz. The relation between the tone frequency and the preset value of TM1 is shown in Table 3.

F_T frequency	400KHz	455KHz
TM1 Present value	04	05
TONE output frequency	40KHz	37.9KHz

Table 3: TONE output

Mode Register 0 (MR0 with SR=28H)

Mode Register 0 is a 4-bit binary register (MR0.0 to MR0.3). The bit descriptions are as follows:

	3	2	1	0
MR0	TM0EN	TM1EN	LCDEN	TONE

TONE = 0 RE3 as the data output of PORTE.3.
= 1 RE3 will be as TONE signal output generated from Timer 1.

LCDEN =0 LCD display OFF.
=1 LCD display ON .

TM1EN =0 Timer 1 counting is disabled.
=1 Timer 1 counting is enabled.

TM0EN=0 Timer 0 counting is disabled.
=1 Timer 0 counting is enabled.

User can read/write and set/clear all bits. At initial reset, MR0 is initiated to 0000B.

Mode 1 Register (MR1 with SR=29H)

Mode Register 1 is a 4-bit binary register (MR1.0 to MR1.3) . The bit descriptions are as follows:

	3	2	1	0
MR1	WDTCK	TM0CK	TM1SR	TM1CK

TM1CK= 0 The internal Timer 1 clock rate is Fosc.
= 1 The internal Timer 1 clock rate is Fosc/64.

TM1SR=0 Timer 1 with internal clock source (depended on TM1CK).
=1 Timer 1 with external clock source from RC0 pin.

TM0CK= 0 The internal Timer 0 clock rate is Fosc/4.
= 1 The internal Timer 0 clock rate is Fosc/1024.

WDTCK= 0 The watchdog timer clock rate is Fosc/1K.
= 1 The watchdog timer clock rate is Fosc/16K.

All bits can be written to by user. At initial reset, MR1 is set to 0000B.

Interrupts

W53300/W53320 provides five internal interrupt sources (Divider, TM0, SPEECH, MELODY and TM1) and two external interrupt sources (port RC and port RD). Vector addresses for each of the interrupts are located in the range of program memory (ROM) addresses from 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 are set by software, an interrupt is generated. When an interrupt occurs, all of the interrupts are inhibited until the *EN INT* or *MOV IEF,#I* 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 control circuit diagram and operation flow chart are shown in Figure 8 and Figure 9 respectively.

Mode Register 2 Register (MR2 with SR=26H)

Mode Register 2 is a 1-bit only register . The INTEN bit of MR2 is used to disable/enable interrupt function. Instruction *DIS EN* can reset INTEN bit to logic 0, and *EN INT* sets INTEN bit to 1.

	3	2	1	0
MR2	X	X	X	INTEN

INTEN = 0 Disable any interrupt process.

= 1 Enable interrupt process where IEF.n is set by 1.

X means don't care.

User can use EN INT to set INTEN=1, and DIS INT to clear INTEN=0. At initial reset, MR2 is initialized to 0001B.

Interrupt Enable Flag Register (IEF with SRP=07H)

The interrupt enable flag (IEF) is an 8-bit binary register (IEF.0 to IEF.7), and comprises 2 registers IEFL (IEF.0~IEF.3) and IEFH (IEF.4~IEF.7) which are used for controlling the interrupt conditions. Instruction *MOV IEF, #I* deals with 8 bit immediate data, whereas *MOV IEFH, #I and MOV IEFL, #I* instructions deal with 4 bit immediate data. When one of the interrupts is accepted, the corresponding bit of the event flag will be reset by hardware, but the other bits are unaffected. In the interrupt subroutine, these interrupts will be disabled until the instruction *MOV IEF, #I* or *EN INT* is executed. Therefore, to enable these interrupts, the instructions *MOV IEF, #I* or *EN INT* must be executed again. These interrupts can be disabled by executing *DIS INT* instruction. The bit descriptions are as follows:

	7	6	5	4	3	2	1	0
IEF	TM1	Melody	Speech	X	RD	RC	TM0	DIV

IEFH

IEFL

IEF.0 = 1 Interrupt 0 is accepted by overflow from the Divider.

IEF.1 = 1 Interrupt 1 is accepted by underflow from the Timer 0.

IEF.2 = 1 Interrupt 2 is accepted by a signal change on port RC.

IEF.3 = 1 Interrupt 3 is accepted by a signal change on port RD.

IEF.4 Reserved

IEF.5 = 1 Interrupt 5 is accepted when end of speech play with SPC_busy falling edge.

IEF.6 = 1 Interrupt 6 is accepted when end of melody play with MLD_busy falling edge.

IEF.7 = 1 Interrupt 7 is accepted by underflow from Timer 1.

All bits can be read/write and set/clear by user.

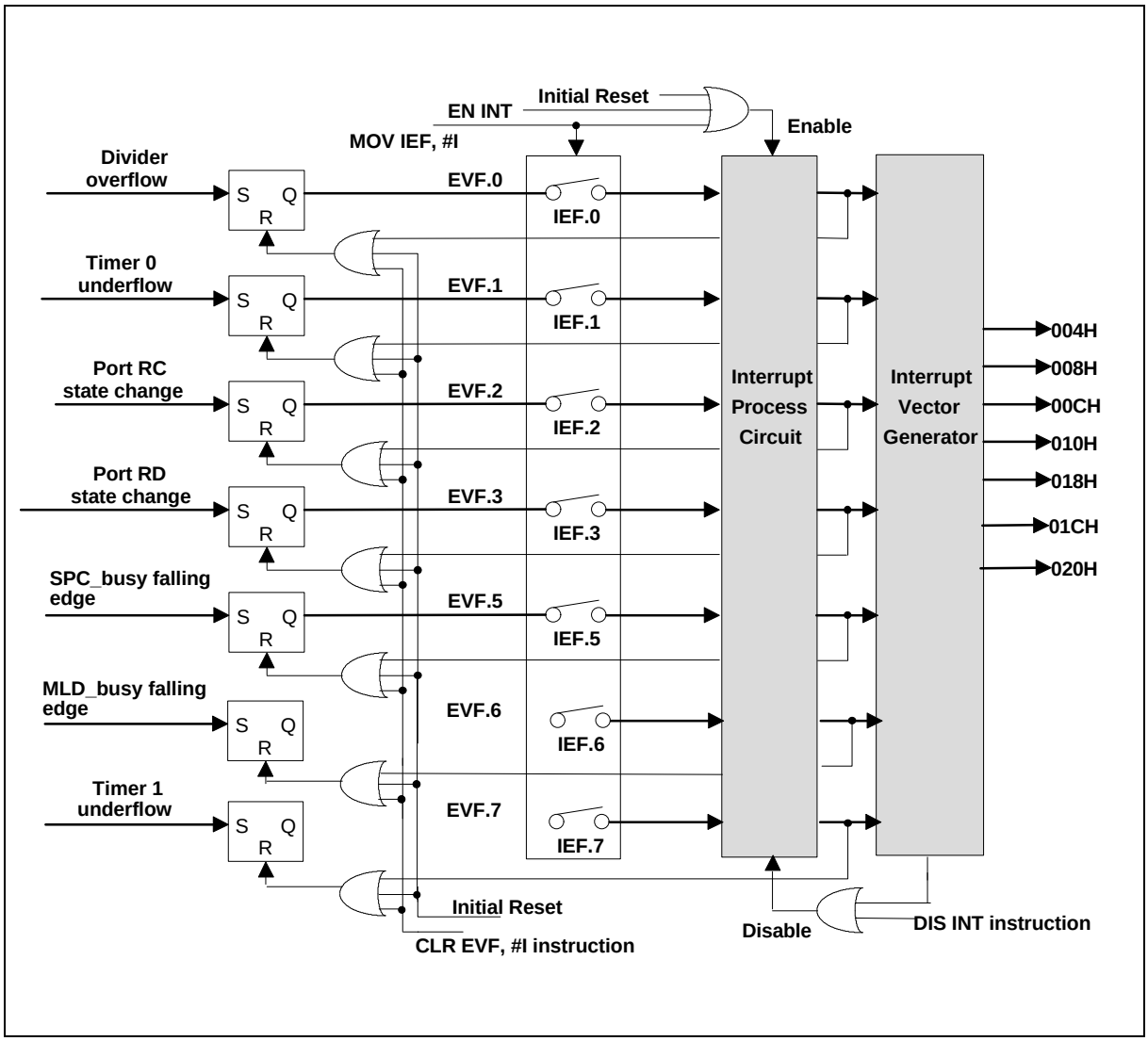


Figure 8. Interrupt Event Control Diagram

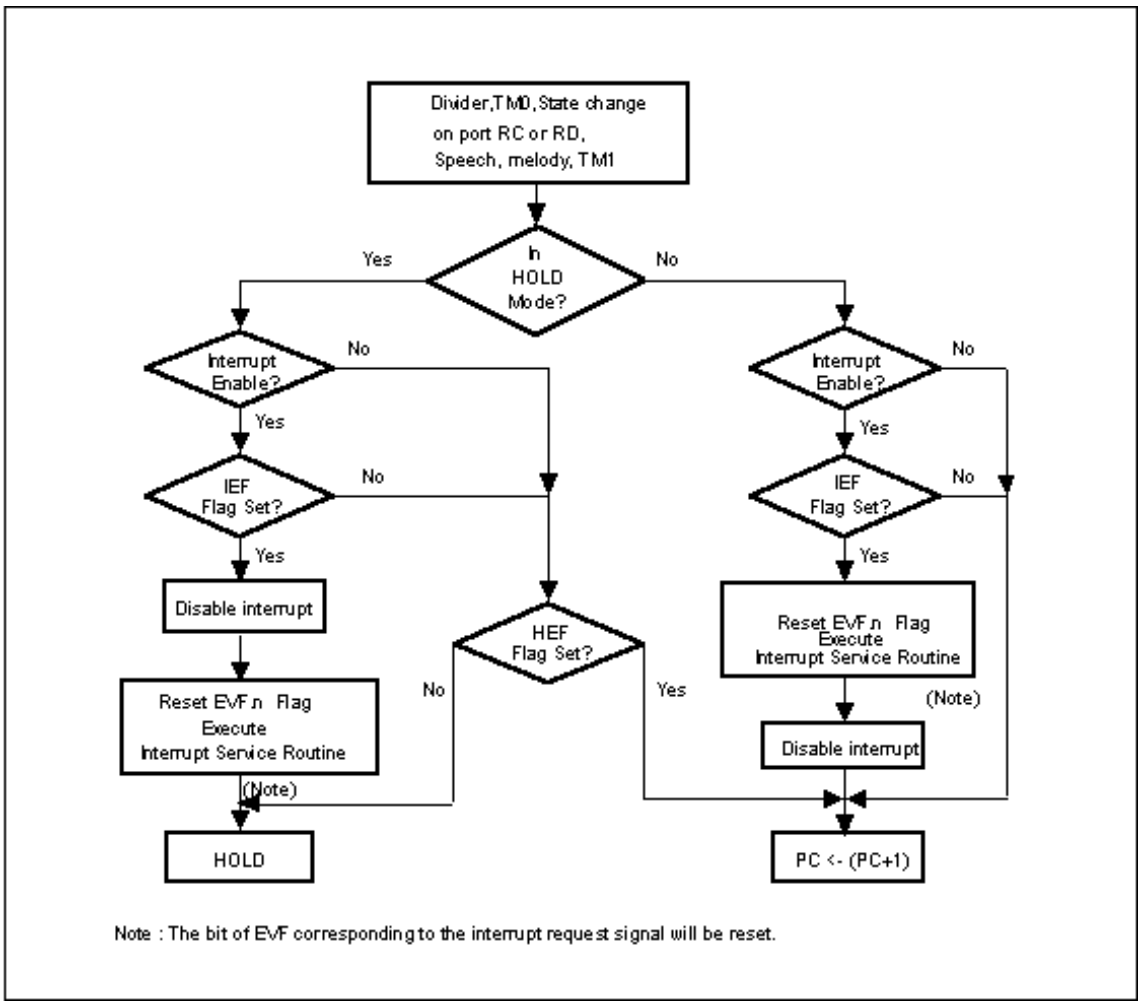


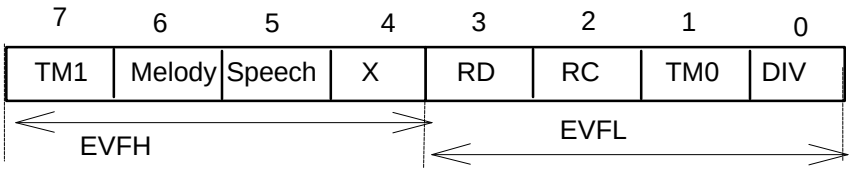
Figure 9. Hold Mode and Interrupt Operation Flow Chart

Hold Mode Operation

In hold mode, all operations of the μC cease except for oscillator, timer, divider and LCD driver . The μC enters hold mode once the HOLD instruction is executed. The hold mode can be released inone of seven ways, which are timer 0 underflow, timer 1 underflow, divider overflow, speech playing finished, melody playing finished, RC port pin state changed and RD port pin state changed. Before the device enters the hold mode, HEF, PEF, and IEF flags must be set to define the hold mode release conditions. For more details, refer to the instruction sets and Figure 9.

Event Flag Register (EVFL, EVFH with SR=0AH & 0BH)

EVFL (EVF.0 ~ EVF.3) and EVFH (EVF.4 ~ EVF.7) are 4 bit registers. It is set by hardware, and reset by instructions *CLR EVFL,#I* and *MOV EVFH,#I* or the occurrence of an interrupt. The bit descriptions are as follows:



EVF.0 = 1 Overflow from Divider.

EVF.1 = 1 Underflow from Timer 0.

EVF.2 = 1 State change on port RC.

EVF.3 = 1 State change on port RD.

EVF.4 Reserved.

EVF.5 = 1 End of speech play with SPC_busy flag falling edge.

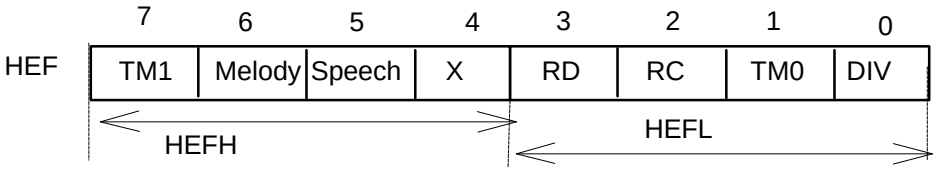
EVF.6 = 1 End of melody play with MLD_busy flag falling edge.

EVF.7 = 1 Underflow from Timer 1 .

All bits can be read and clear only by users.

Hold Mode Release Enable Flag Register (HEF with SRP=06H)

The hold mode release enable flag is a 8-bit register (HEF.0 to HEF.7), and comprises two registers HEFL (HEF.0 ~ HEF.3) and HEFH (HEF.4~ HEF.7). Register HEF is used to control the hold mode release conditions, through the use of instruction *MOV HEF, #I* with 8 bit immediate data, or *MOV HEFH,#I* and *MOV HEFL,#I* with 4 bit immediate data. The bit descriptions are as follows:



HEF.0 = 1 Overflow from the Divider causes hold mode to be released.

HEF.1 = 1 Underflow from Timer 0 causes hold mode to be released.

HEF.2 = 1 State change on port RC causes hold mode to be released.

HEF.3 =1 State change on port RD causes hold mode to be released

HEF.5 = 1 End of speech play with SPC_busy flag falling edge causes hold mode to be released

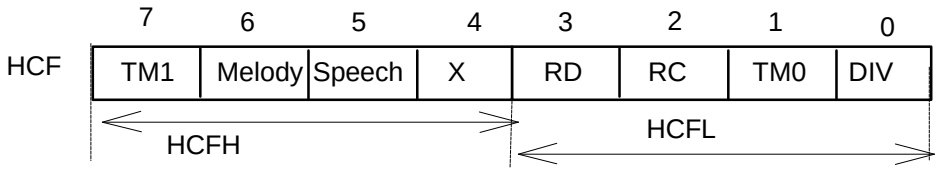
HEF.6 =1 End of melody play with MLD_busy flag falling edge causes hold mode to be released

HEF.7 = 1 Underflow from Timer 1 causes hold mode to be released.

All bits can be read/write and set/clear by users.

Hold Mode Release Condition Flag Register (HCFL, HCFH with SR=10H & 11H)

HCF is a 8-bit register (HCF.0 to HCF.7), and consists of HCFL (HCF.0~HCF.3) and HCFH registers (HCF.4~HCF.7). The hold mode is released, and loaded by hardware. The content of HCF can be read by using 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 will be invalid. HCF can be reset by instruction *CLR EVFL/EVFH,#I* (EVF.n = 0). When EVF.n or HEF.n is reset, the corresponding bits of HCF is reset simultneously by hardware. The bit descriptions are as follows:



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

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

HCF.2 = 1 Hold mode was released by a state change on port RC.

HCF.3 = 1 Hold mode was released by a state change on port RD.

HCF.4 reserved

HCF.5 = 1 Hold mode was released when end of speech play with SPC_busy falling edge

HCF.6 = 1 Hold mode was released when end of melody play with MLD_busy falling edge

HCF.7 = 1 Hold mode was released by underflow from the Timer 1

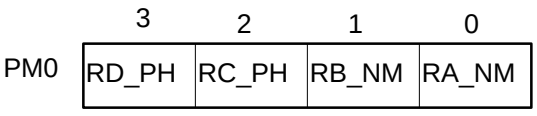
All bits are read only by users, and set/clear by chip hardware.

Input/Output Ports RA, RB

Port RA consists of 4 pins RA0 to RA3, and port RB also consists of 4 pins RB0 to RB3. At initial reset, RA and RB input/output ports are both in input mode. When RA and RB are used as output ports, CMOS or NMOS open drain output type can be selected by the PM0 special register. Each pin of port RA or RB can be specified as input or output mode independently by PM1 and PM2 special registers. The *MOVA R, PORTA* and *MOVA R, RORTB* instructions operate the input functions, as *MOV PORTA, R* and *MOV PORTB, R* operate the output functions. For more details, refer to the instruction table and Figure 10.

Port Mode 0 Register (PM0 with SR=32H)

PM0 is a 4-bit register (PM0.0 to PM0.3). It 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:



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

RB_NM = 0 RB port is CMOS output type.
= 1 RB port is NMOS open drain output type.

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

RD_PH = 0 RD port pull-high resistor is disabled.
= 1 RD port pull-high resistor is enabled.

All bits can be read/write and set/clear by user. At initial reset, PM1 is initialized to 0000B , where port RA and RB are of type CMOS input mode; port RC and RD are disabled pull-high resistors.

Port Mode Register 1 and 2 (PM1, PM2 with SR=36H, 37H)

PM1 (PM1.0 ~ PM1.3) and PM2 (PM2.0~PM2.3) are 4-bit registers. PM1 is used to control the input/output mode of port RA, using the instruction *MOV PM1, #I*. PM2 is used to control the RB input/output mode of port RB, using the instruction *MOV PM2, #I.* The bit descriptions are as follows:

	3	2	1	0
PM1	RA3IN	RA2IN	RA1IN	RA0IN

RA0IN = 0 RB0 works as output pin;
= 1 RB.0 works as input pin

RA1IN = 0 RB1 works as output pin;
= 1 RB.1 works as input pin

RA2IN = 0 RB2 works as output pin;
=1 RB.2 works as input pin

RA3IN = 0 RB3 works as output pin;
= 1 RB.3 works as input pin

	3	2	1	0
PM2	RB3IN	RB2IN	RB1IN	RB0IN

RB0IN = 0 RB0 works as output pin;
= 1 RB.0 works as input pin

RB1IN = 0 RB1 works as output pin;
= 1 RB.1 works as input pin

RB2IN = 0 RB2 works as output pin;
=1 RB.2 works as input pin

RB3IN = 0 RB3 works as output pin;
= 1 RB.3 works as input pin

All bits can be read/write and set/clear by users. At initial reset, port RA, RB is in input mode (PM1 = PM2 = 1111B).

Port A Register (PORTA with SR=38H)

PORTA register stores the current port RA pin state, and can be operated by *MOV PORTA, R* and *MOV R, PORTA* instructions. When port A is in input mode, it is read only. Otherwise a write operation is performed during RA ouput mode.

	3	2	1	0
PORTA	PA3	PA2	PA1	PA0

Port B Register (PORTB with SR=39H)

PORTB register stores the current port RB pin state, and can be operated by *MOV PORTB, R* and *MOV R, PORTB* instructions. When port B is in input mode, it is read only. Otherwise a write operation is performed during RB ouput mode.

	3	2	1	0
PORTB	PB3	PB2	PB1	PB0

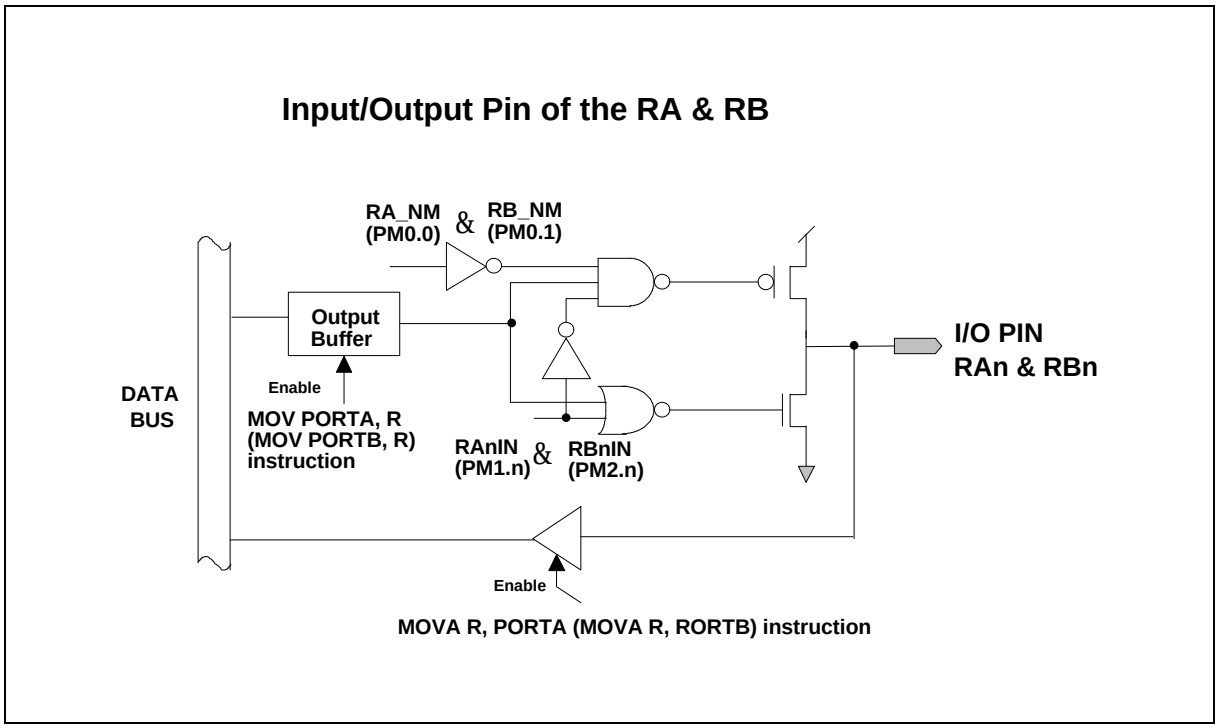


Figure 10. Architecture of RA (RB) Input/Output Pins

Input Ports RC, RD

Port RC consists of 4 pins RC0 to RC3, just as port RD consists of 4 pins RD0 to RD3. Each pin of port RC and port RD can be connected to a pull-up resistor, which is controlled by the port mode 0 register (PM0). When the bits of PEF, HEF, and IEF corresponding to the RC (RD) port are set, a state change at the specified pins of port RC (RD) will execute the hold mode release or interrupt subroutine. PSR0 and PSR1 record the signal changing status of the port RC and RD. PSR0 can be read and cleared by the *MOVA R, PSR0* and *CLR PSR0* instructions. PSR1 can be read and cleared by the *MOVA R, PSR1* and *CLR PSR1* instructions. Refer to Figure 11 and the instruction sets for more details.

Port Enable Flag for hold mode (PEFL, PEFH with SR=14H, 15H)

PEFL (PEFL.3~PEFL.0) and PEFH (PEFH.3~PEFH.0) are 4 bit registers. PEFL controls port RC status, and PEFH is responsed for port RD status. Before port RC or RD is used to release the hold mode or to perform an interrupt function, the content of the PEFL/PEFH must be set first. If PEFL/PEFH is written to "1", the function will be enabled. Instructions *MOV PEFH,#I* and *MOV PEFH,#I* can be used with 4 bit immediate data. The bit descriptions are as follows:

	3	2	1	0
PEFL	PEFL.3	PEFL.2	PEFL.1	PEFL.0

PEFL.0 =1 : State change on pin RC0 to release hold mode or perform interrupt

PEFL.1 =1 : State change on pin RC1 to release hold mode or perform interrupt

PEFL.2 =1 : State change on pin RC2 to release hold mode or perform interrupt

PEFL.3 =1 : State change on pin RC3 to release hold mode or perform interrupt

	3	2	1	0
PEFH	PEFH.3	PEFH.2	PEFH.1	PEFH.0

PEFH.0 =1 : State change on pin RD0 to release hold mode or perform interrupt

PEFH.1 =1 : State change on pin RD1 to release hold mode or perform interrupt

PEFH.2 =1 : State change on pin RD2 to release hold mode or perform interrupt

PEFH.3 =1 : State change on pin RD3 to release hold mode or perform interrupt

All bits can be read/write and set/clear by users.

Port Status Register 0 and 1 (PSR0, PSR1 with 34H, 35H)

PSR 0 (PSR0.0 to PSR0.3) and PSR1 (PSR1.0 to PSR1.3) are 4-bit registers. They are set to "1" if PEF.n is enabled and RCn (RDn) input state is changed. Then hold mode or interupt will occur. Refer to Figure 10. PSR0 can be read or cleared by instructions *MOVA R, PSR0*, and *CLR PSR0*. PSR1 can be read or cleared by instructions *MOVA R, PSR1*, and *CLR PSR1*. The bit descriptions are as follows:

	3	2	1	0
PSR0	RC3EG	RC2EG	RC1EG	RC0EG

Bit 0 = 1 : RC0 input signal state is changed
= 0 : RC0 input signal state isn't changed
Bit 1 = 1 : RC1 input signal state is changed
= 0 : RC1 input signal state isn't changed

Bit 2 = 1 : RC2 input signal state is changed
= 0 : RC2 input signal state isn't changed
Bit 3 = 1 : RC3 input signal state is changed
= 0 : RC3 input signal state isn't changed

All bits are read only, and are cleared simultaneously. At initial reset , PSR1 is set to 0000B.

	3	2	1	0
PSR1	RD3EG	RD2EG	RD1EG	RD0EG

Bit 0 = 1 : RD0 input signal state is changed
= 0 : RD0 input signal state isn't changed
Bit 1 = 1 : RD1 input signal state is changed
= 0 : RD1 input signal state isn't changed
Bit 2 = 1 : RD2 input signal state is changed
= 0 : RD2 input signal state isn't changed
Bit 3 = 1 : RD3 input signal state is changed
= 0 : RD3 input signal state isn't changed

All bits are read only, and are cleared simultaneously. At initial reset , PSR1 is set to 0000B.

Port C Register (PORTC with SR=3AH)

This register stores the port RC current input state, which is specified by *MOV R, PORTC* instruction.

	3	2	1	0
PORTC	PC3	PC2	PC1	PC0

Port D Register (PORTD with SR=3BH)

This register stores the port RD current input state, which is specified by *MOV R, PORTD* instruction.

	3	2	1	0
PORTD	PD3	PD2	PD1	PD0

Output Port RE

When instruction *MOV PORTE, R* is executed, the data in the RAM will be output to port RE. The RE3 pin can be used to output TONE from Timer 1 if TONE option bit is set to 1.

Port E Register (PORTE with SR=3CH)

This register stores the current Port RE output state, which is specified by the *MOV PORTE, R* instruction.

	3	2	1	0
PORTE	PE3	PE2	PE1	PE0

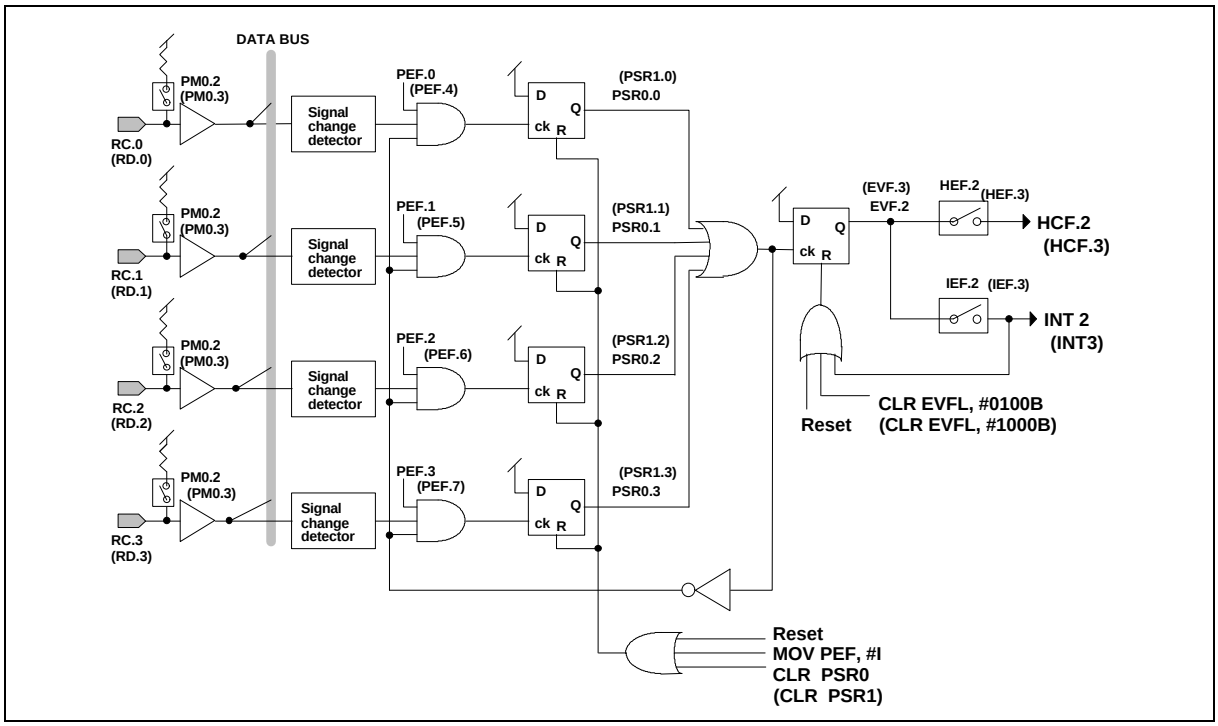


Figure 11. Architecture of Input Ports RC (RD)

Reset Function

W53300/W53320 is reset either by a power-on reset or $\overline{RES}$ active low pulse. Table 4 shows the initial reset state of internal special register and Input/Output.

Program Counter (PC)	0000B
Input/output ports RA, RB	Input mode
Output port RE	0000B
RA & RB ports output type	CMOS type
RC & RD ports pull-high resistors	Disable
System Clock Input	Fs (32768HZ)
Timer 0 input clock	Fosc/4
Timer 1 input clock	Fosc
Input clock of the watchdog timer	Fosc/1024
LCD display	OFF
LCD Bias	1/7 bias
LCD Duty	32 duty
LCD Internal Pump Circuit	Enable
LCD Pump Voltage	Triple pump
SCR register	0001B
MR2 register (INTEN flag)	0001B
MR3 register	0000B
PM1,PM2 register	1111B
Others Registers	0000B

Table 4: Default value at initial reset

PART B: SPEECH and MELODY FUNCTION

Both speech and melody use the same clock source from ROSC pin. When speech or melody is playing, the ROSC clock is enabled, otherwise it is disabled for power saving. Speech synthesizer, just like melody sound tone, can be output to PWM1 and PWM2, thus directly driving the speaker. MR3 register assigns 2 bits (CUR1~CUR0) to control PWM current option. Speech coding can select whether two LED output pins (LED1 and LED2) will be active. LED1 can also be activated by melody depending on its output volume. However, speech and melody can not play at the same time. Therefore, users should require W53300/W53320 to play either speech sound or melody sound.

Mode Register 3 (MR3 with SR=27H)

The MR3 register is used to enable internal LCD voltage pump circuit and PWM current output option. While VLCDEXT is disabled, PMPV3B bit option has meaning. On clearing VLCDEXT bit, the built-in LCD voltage pump circuit generates VDD2 and VDD3 from VDD. However, if VLCDEXT is set, an external power source needs to be added to VDD3 pin to generate common and segment output voltage.

	3	2	1	0
MR3	VLCDEXT	X	CUR1	CUR0

.

CUR1~0: select PWM output driving current type

00: 5mA driving capability

01: 10mA driving capability

10: 15mA driving capability

11: 20mA driving capability

VLCDEXT = 0 VDD3 is output pin, enabling internal LCD voltage pump circuit

= 1 VDD3 is input pin, connecting external power source to VDD3 for com/segoutput

MR3 register is write only. At initial reset, MR3 bit is set to 0000B.

FLAG0 Register (FLAG0 with SR=22H)

FLAG0 is a 4-bit register used to control the speech and melody synthesizers. FLAG0.1~0 can be read or written to, set or cleared by user, but FLAG0.3 ~ FLAG0.2 are set and cleared by chip hardware. At initial reset, FLAG0 is initialized to 0000B. Bits are defined as follows.

	3	2	1	0
FLAG0	MLD_busy	SPC_busy	MLD_play	SPC_play

SPC_paly =0 : Speech play is disabled.

=1 : Speech play is enabled while SPC_play high keeps 8 Tcyc minimum.

MLD_paly =0 : Melody play is disabled.

=1 : Melody play is enabled while MLD_play high keeps 30ms minimum.

SPC_busy =0 : Speech play is finished.

=1 : Speech play is processing.

MLD_busy =0 : Melody play is finished.

=1 : melody play is processing.

SPEECH Function

Speech synthesizer is built-in 384 K bits dedicated speech ROM, which can be separated by Winbond ADPCM Power Speech Coding System to a maximum of 256 sections with different voices. In order to play speech voice, the playing section number needs to be written to SPCH and SPCL, and SPC_play option bit (bit 0 of FLAG0 special register) set to 1. Then SPC_busy bit (bit 2 of FLAG0) will be changed from low to high, and remains high till speech play comes to an end. If interrupt flag (IEF.5) or hold mode flag (HEF.5, HCF.5) is set, interrupt or hold mode release will be processed when SPC_busy falling edge occurs. The circuit structure is shown in Figure 12. SPC_play bit can be set to "1" again, after finishing transfer of section number via parallel to serial interface during previous SPC_play edge. There is minimum 8-instruction delay between two continuous SPC_play rising.

Two LED outputs with 3Hz frequency can be used to drive external LED during speech playing. The SPCH and SPCL will be latched during SPC_play risng edge. The speech synthesizer is disabled when MLD_busy bit (bit 3 of FLAG0) is 1, and so is the melody synthesizer when SPC_busy bit is 1.

The SPC_play is set to 1 to activate the speech synthesizer. The speech synthesizer receives the rising edge of SPC_play, then plays the voice section pointed by SPC.7~SPC.0 and sets SPC_busy to logic 1. The SPC_busy is cleared by hardware if the speech synthesizer finishes its tasks or executes an END section. The END section number can be defined by speech programmer at any section.

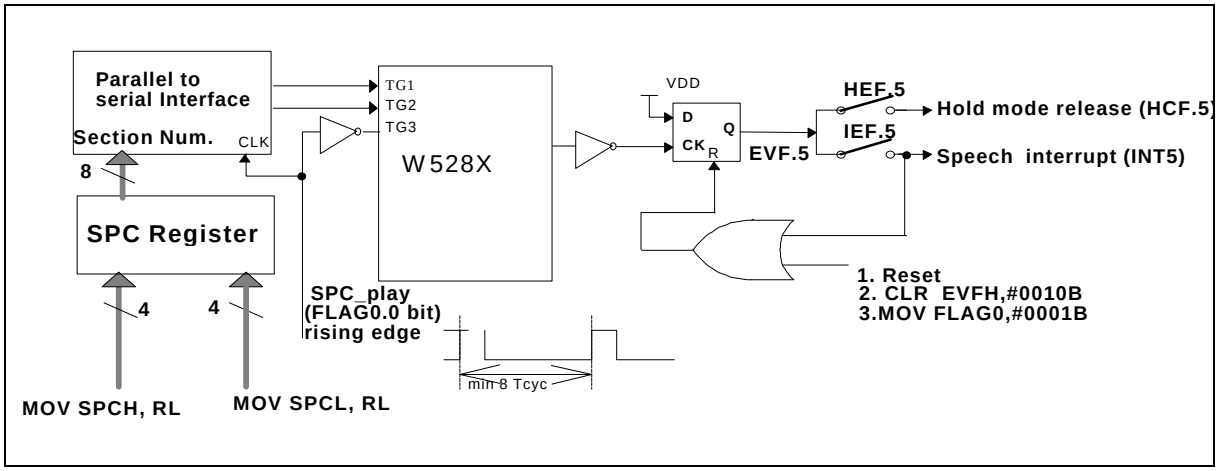
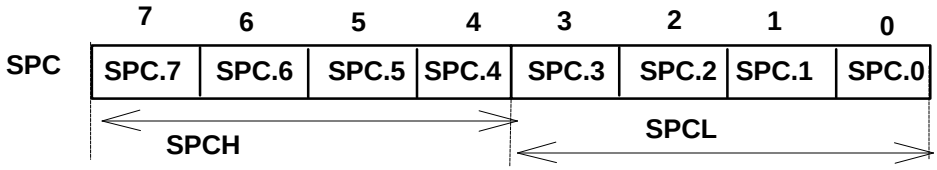


Figure 12. Speech Circuit Diagram

Speech Section Register (SPCL, SPCH with SR=1EH & 1FH)

The SPCH and SPCL registers named as SPC.7 ~SPC.0 define the speech section that the speech synthesizer is to play. SPCH (SPC.7 ~ SPC.4) represents the high nibble as SPCL (SPC.3 ~ SPC.0) represents the low nibble. When actived, the speech synthesizer plays the voice section pointed by SPC.7~SPC.0 with maximum 256 sections (00h to FFh).



Melody Function

Dual tone melody is dedicated ROM with built-in tremolo function , 3 percussions, 6 beats, 41 pitches from G3# to C7, 16 kinds of tempo and 1K notes (22 bits per note), which can be separated to a maximum of 32 different scores. The dual tone melody is played using the same methodology as speech, but including END command. The melody scores can be written to MLDH and MLDL registers. Then MLD_play is enabled high to play melody, and the MLD_busy bit will be changed from low to high and remains high till melody play is ended. If interrupt flag (IEF.6) or hold mode flag (HEF.6, HCF.6) is set, interrupt or hold mode release will be processed when MLD_busy falling edge occurs. The MLDH and MLDL will be latched during MLD_play rising edge. Users can select melody play mode using the OSB bit (bit 1 of MLDH). In one-shot trigger mode (OSB=0), the melody synthesizer receives a rising edge of MLD_play, then plays the score pointed by MLD4~MLD.0, and sets the MLD_busy to logic 1. When the melody synthesizer finishes its tasks or receives a rising edge of MLD_play with score number with *END* command, the melody synthesizer enters the standby mode and MLD_busy is pulled to logic 0. In level-trig mode (OSB=1), the melody synthesizer plays the pointed score when MLD_busy is set to 1. The MLD pointed score is repeatedly played and the MLD_busy is pulled high until the MLD_play is cleared by user. Bits MLED1 and MLED0 of the MLDH register can be used to control the LED1 pin output timing while melody is playing, where LED1 output is a volume control by melody tone.

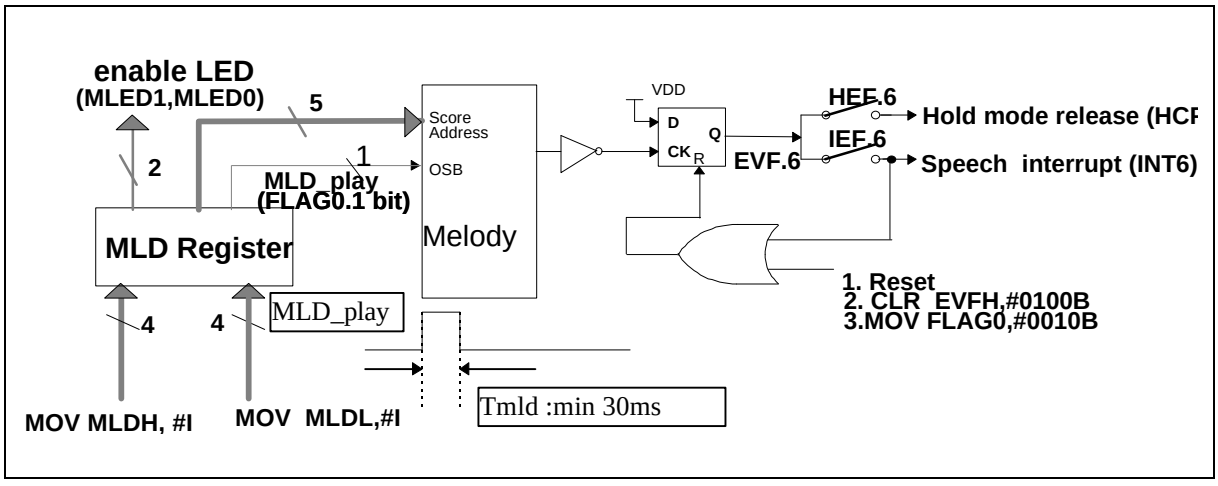
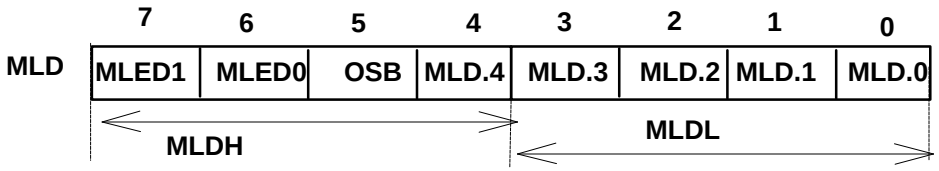


Figure 13. Melody Circuit Diagram

Melody Scores Register (MLDL, MLDH with SR=1CH, 1DH)

MLD register is made up of two 4-bit registers, MLDH and MLDL. MLDH represents the high nibble (MLED1, MLED0 , OSB, MLD.4), while MLDL represents the low nibble (MLD.3 ~ MLD.0). Bit MLD.4 to bit MLD.0 together form a 5-bit pointer of scores. MLED1 and MLED0 is used to control LED1 pin active type during melody playing. When actived, the melody synthesizer plays the score section pointed by the MLD.4 ~ MLD.0 . 31 scores, from score 01H to score 1FH, can be pointed by the MLD register. When the melody synthesizer is on one-shot trigger mode and MLD.4 ~ MLD.0 is set to 00H, a melody-play command becomes a melody-stop command.



MLD.4 ~ MLD.0 are the melody score number with maximum 31 scores.

OSB=0 : Melody play mode by one shot trigger

=1 : Melody play mode by level trigger

MLED1~0: Select LED1 output pin active type while melody is playing.

00: LED1 is disabled while melody is playing

01: LED1 will be active when melody volume is higher than low level

10: LED1 will be active when melody volume is higher than middle level

11: LED1 will be active when melody volume is higher than high level

PART III: LCD FUNCTION

W53300/W53320 can directly drive an LCD panel with 16/32 common output pins and 48 segment output pins for a total of 32 × 48 dots. The updating rate per frame is 64 Hz. Two registers, LCDM1 and **MR3.3** (refer to page MR3 description) can be used to select different LCD operating types, such as duty cycle, bias ratio, maximum pump voltage, internal shunt resistor, and enable LCD pump voltage circuit, by using *instructions MOV LCDM1, #I*, *MOV LCDM1, RL* (where RL is the low 9 bit of RAM address) and *MOV LCDM1, ACC*. For power saving, one can use *LCDON* and *LCDOFF* instructions, to turn the LCD display on and off. This is done by controlling the LCDEN bit (bit 1 of MR0 register). The LCD display can be turned on even in HOLD mode. At initial reset, LCDM1 register is initiated to 0000B. This sets the LCD operating condition to 1/16 or 1/32 duty, 1/5 or 1/7 bias, triple pump voltage with internal shunt resistor, and lights up all the LCD segments. Upon the end of the initial reset state, the LCD display is turned off automatically. The circuit architecture is shown in Figure 14. Different application conditions are shown in Figures 15 to 19.

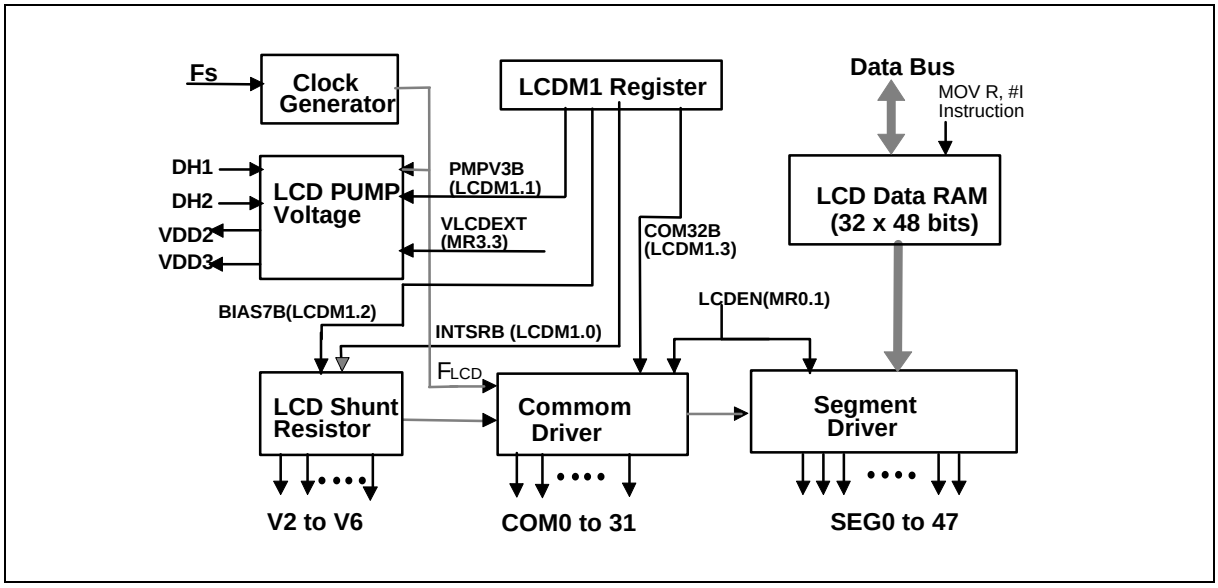


Figure 14. LCD Driver Circuit Diagram

LCD Pattern RAM (LCDR000H~LCDR17FH)

Corresponding to 48 LCD drive output pins, there are 384 LCD data RAM (LCDR). The address of LCD data RAM, which can be addressed by LCDR000 to LCDR17F, is mapping to address from 200H to 37FH of general purpose RAM. In fact, LCD data RAM are also general purpose RAM, and all instructions that are applied to RAM 00H~1FFH are also applied to LCDR. Therefore, the unused portions of LCD RAM can be used as general purpose data RAM. Instructions such as *MOV LCDR,#I*, *MOV WR, LCDR, MOV LCDR,WR* and *MOV LCDR, ACC* are provided to control the LCD data RAM, as the assembler will interpret LCDR as RAM by automatically adding 1FFH. Writing 1 to bit value of LCD data RAM will turn the LCD dot on; nevertheless 0 turns off the LCD dot. The contents of the LCD data RAM are sent out to the SEG0~SEG47 pins by a direct memory access. The relation between the LCD data RAM and segment/common pins is shown in Table 5.

LCD DATA RAM	OUTPUT PIN	BIT3	BIT 2	BIT 1	BIT 0
LCDR000 (RAM200)	SEG0	COM3	COM2	COM1	COM0
LCDR001 (RAM201)		COM7	COM6	COM5	COM4
LCDR002 (RAM202)		COM11	COM10	COM9	COM8
LCDR003 (RAM203)		COM15	COM14	COM13	COM12
LCDR004 (RAM204)		COM19	COM18	COM17	COM16
LCDR005 (RAM205)		COM23	COM22	COM21	COM20
LCDR006 (RAM206)		COM27	COM26	COM25	COM24
LCDR007 (RAM207)		COM31	COM30	COM29	COM28
LCDR008 (RAM208)	SEG1	COM3	COM2	COM1	COM0
LCDR009 (RAM209)		COM7	COM6	COM5	COM4
LCDR00A (RAM20A)		COM11	COM10	COM9	COM8
LCDR00B (RAM20B)		COM15	COM14	COM13	COM12
LCDR00C (RAM20C)		COM19	COM18	COM17	COM16
LCDR00D (RAM20D)		COM23	COM22	COM21	COM20
LCDR00E (RAM20E)		COM27	COM26	COM25	COM24
LCDR00F (RAM20F)		COM31	COM30	COM29	COM28
:	:	:	:	:	:
:	:	:	:	:	:
LCDR378 (RAM378)	SEG47	COM3	COM2	COM1	COM0
LCDR379 (RAM379)		COM7	COM6	COM5	COM4
LCDR37A (RAM37A)		COM11	COM10	COM9	COM8
LCDR37B (RAM37B)		COM15	COM14	COM13	COM12
LCDR37C (RAM37C)		COM19	COM18	COM17	COM16
LCDR37D (RAM37D)		COM23	COM22	COM21	COM20
LCDR37E (RAM37E)		COM27	COM26	COM25	COM24
LCDR37F (RAM37F)		COM31	COM30	COM29	COM28

Table5. LCD RAM mapping to segment and common output pins

LCD Mode Register 1 (LCDM1 with SR=2AH)

LCDM1 is a 4-bit register (LCDM1.0 ~ LCDM1.3) for selecting LCD duty cycle, bias ratio, pump voltage and internal shunt resistor, by using the instructions *MOV LCDM1, #I*, *MOV LCDM1, RL*" (where RL is the low 9 bit of RAM address) and *MOV LCDM1, ACC*.

Bit 3 of LCDM1, COM32B defines the duty cycle. Bit 2, known as BIAS7B, controls bias ratio to match the characteristics of the LCD panel. Bit 1, PMPV3B, is used to choose the maximum output voltage (either a doubler or a tripler) of COM/SEG, when the build-in LCD voltage pump circuit is enabled. The voltage tripler should be enabled for 3V operating voltage, and the voltage doubler for 4.5V operating voltage. Bit 0, defined as INTSRB, is used to select the internal shunt reistor for V2~V6 output power. Diagrams below show the application circuits; output waveforms of the five LCD driving modes are shown in Figures 14 to 25.

	3	2	1	0
LCDM1	COM32B	BIAS7B	PMPV3B	INTSRB

INTSRB = 0 Internal shunt resistor is available between V2~V6

= 1 External shunt resistor is needed between V2 ~V6.

PMPV3B = 0 Triple pump voltage available (recommended when VDD=3v)

= 1 Double pump voltage available (recommended when VDD=4.5v)

BIAS7B = 0 1/7 bias available (recommended for 32 common)

= 1 1/5 bias available (recommended for 16 common)

COM32B =0 1/32 duty, COM0~COM31 output available

=1 1/16 duty, COM0~ COM15 output available

All bits are write only. At initial state, LCDM1 is cleared to 0000B**.**

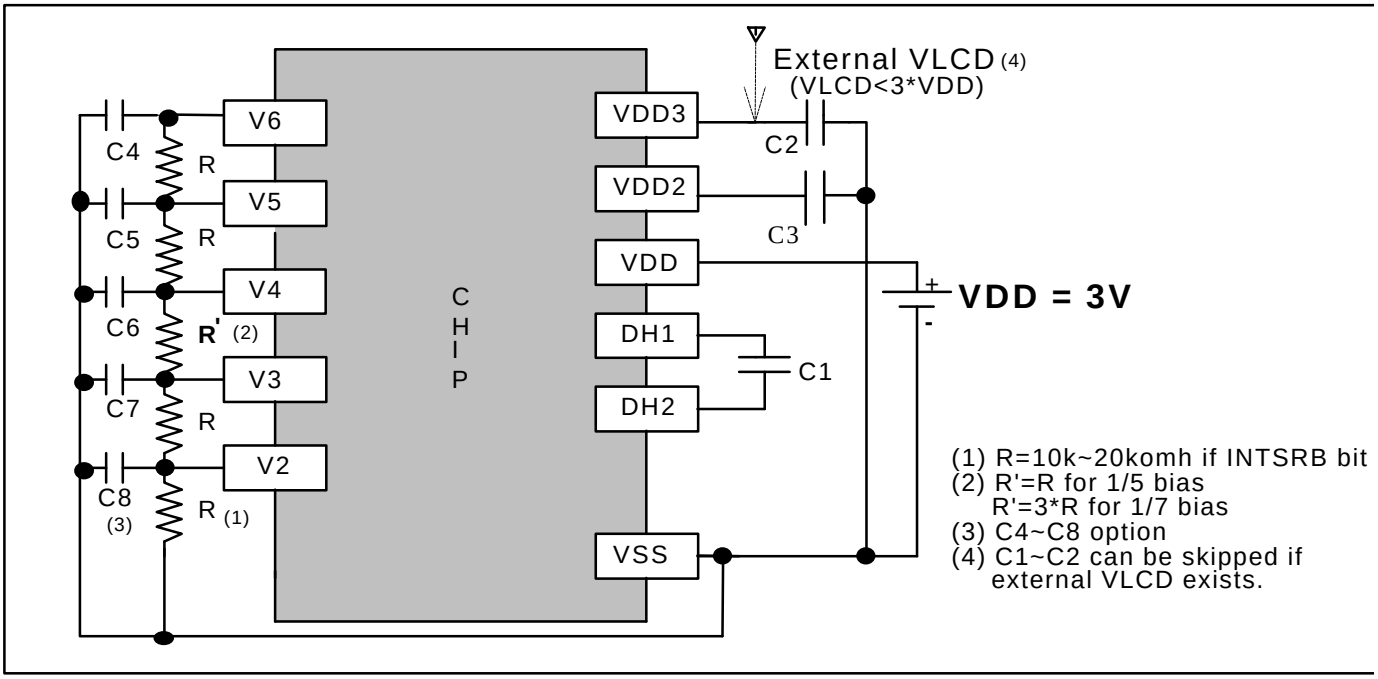


Figure 15. Triple pump voltage

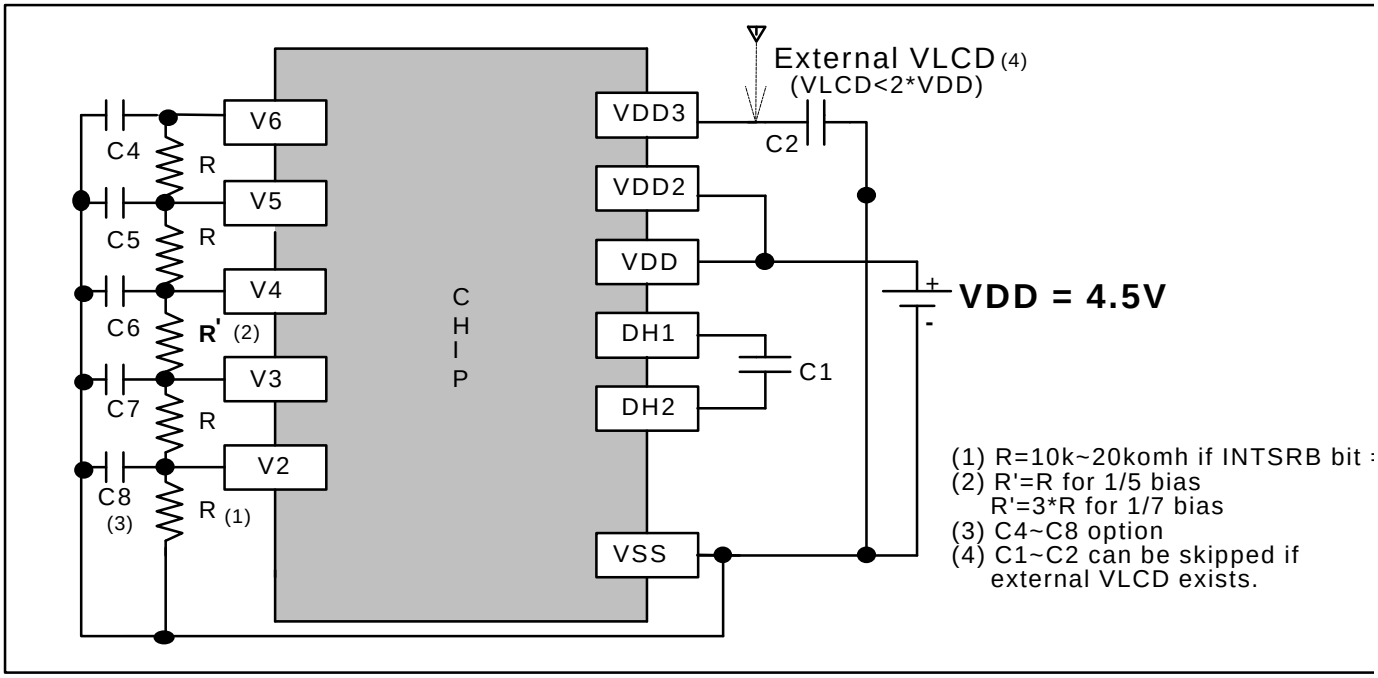


Figure 16. Double pump voltage

LCD Example and Waveform:

On the basis of the Figure 17 pattern assignment, the common, segment output waveform can be obtained as shown in Figure 18 for 1/7 bias, and in Figure 19 for 1/5 bias.

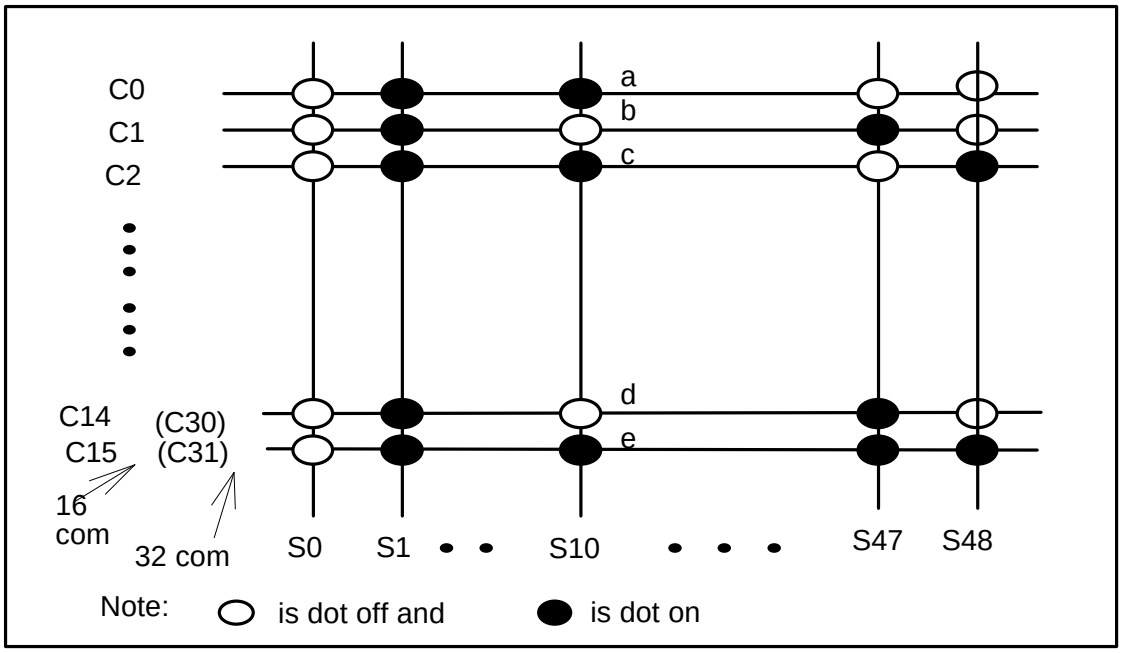


Figure 17. Common/Segment driving pattern

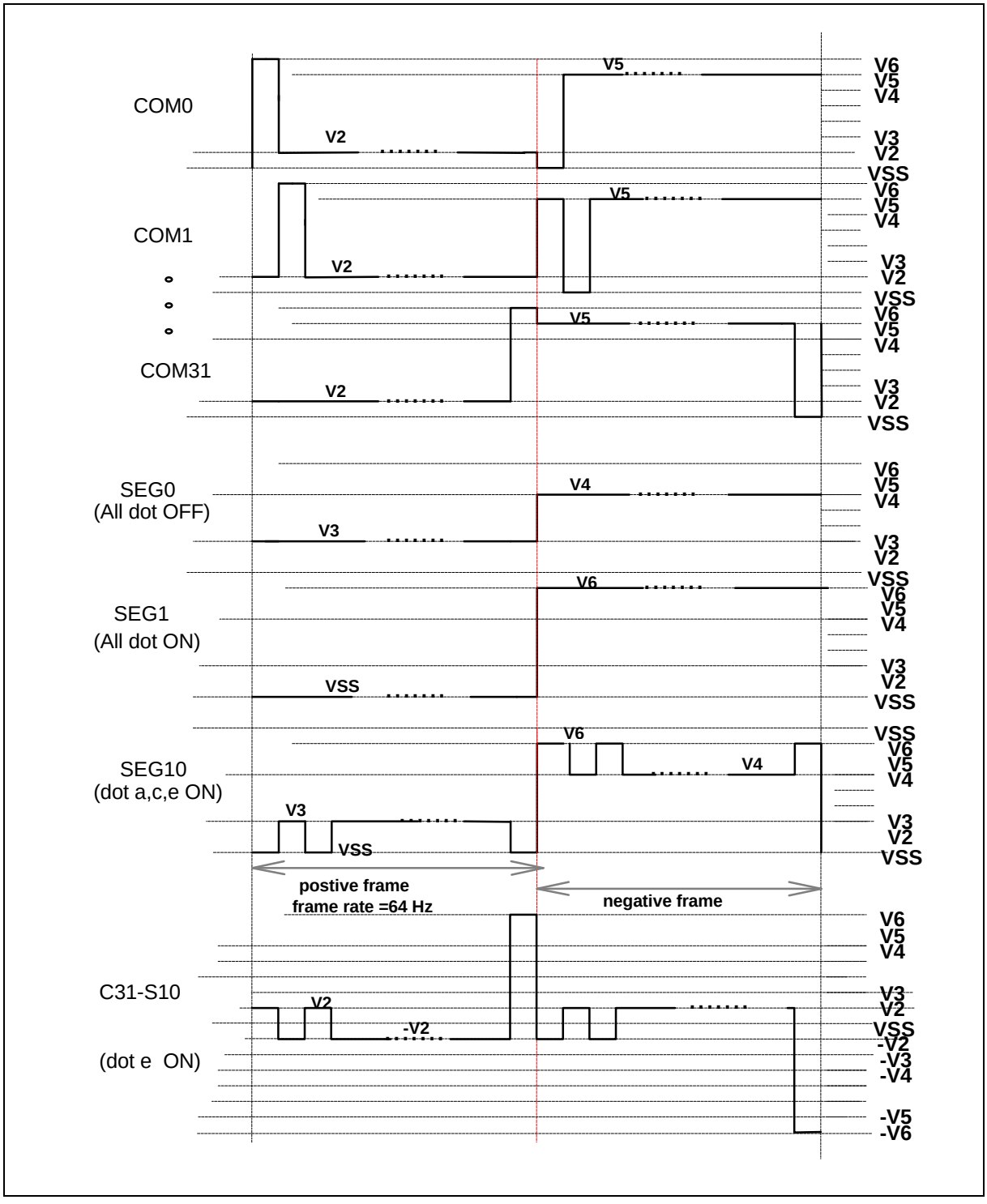


Figure 18. 1/7 bias, 1/32 duty driving waveform

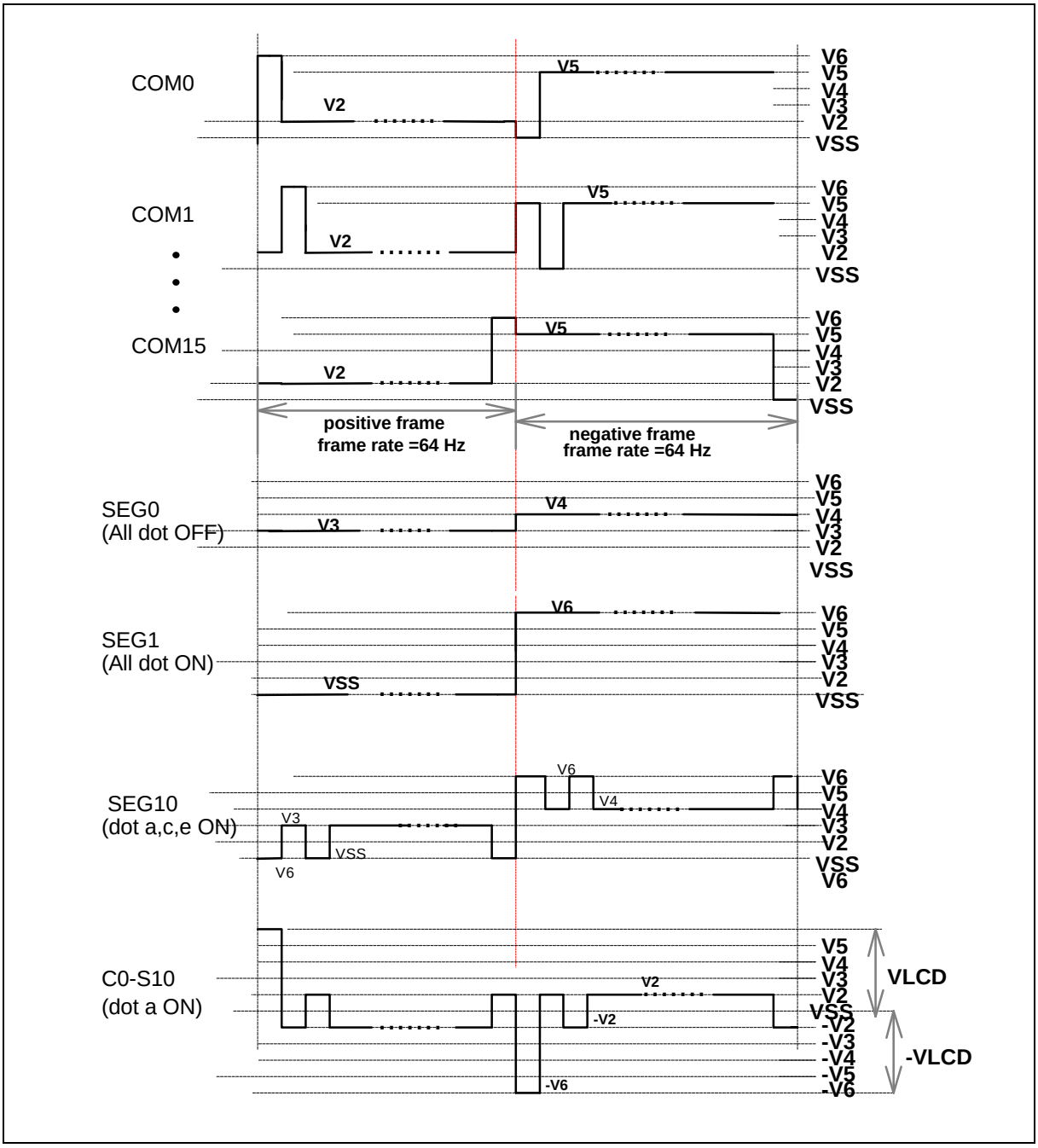
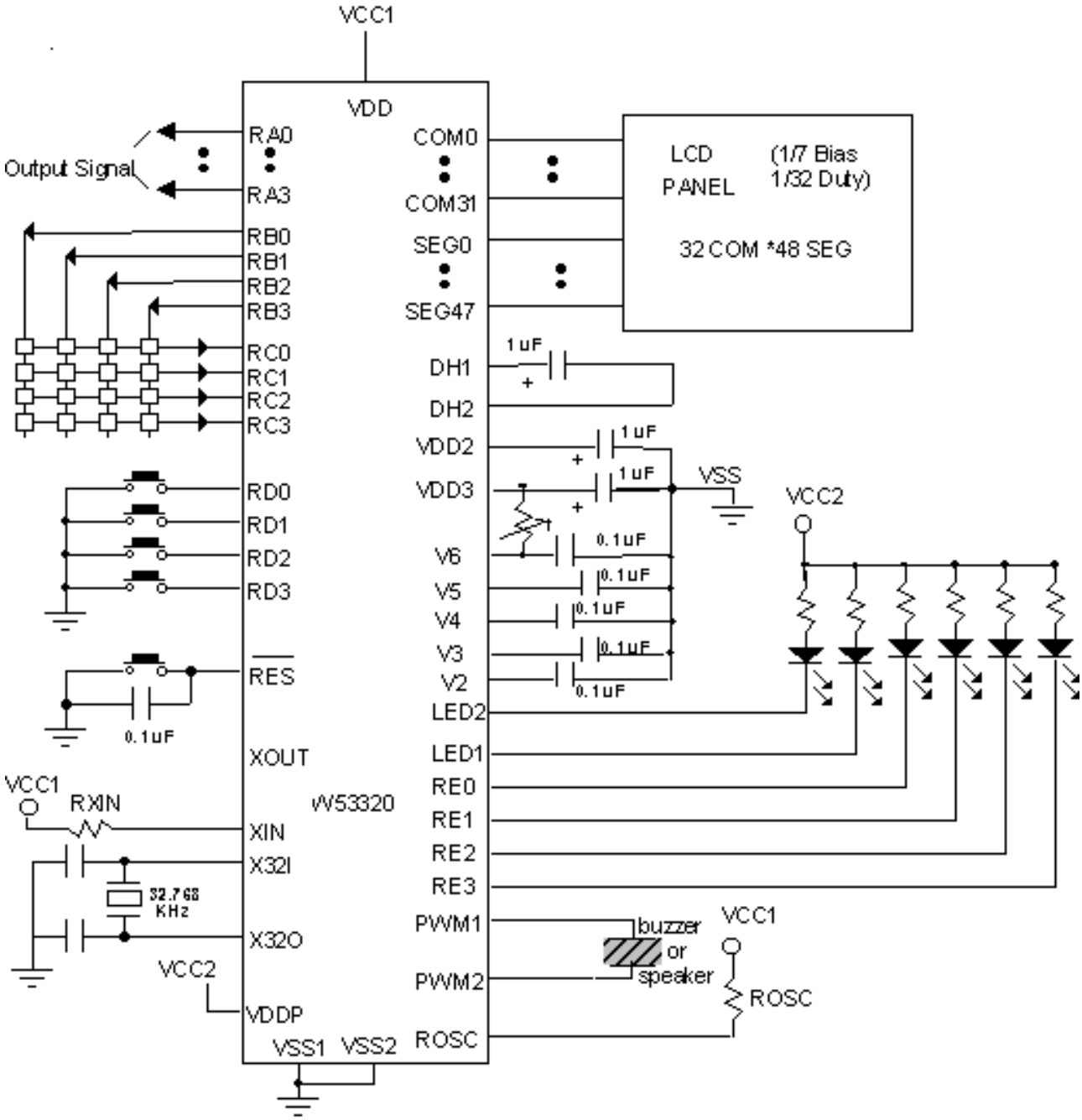


Figure 19. 1/5 bias, 1/16 duty driving waveform

TYPICAL APPLICATION CIRCUIT



INSTRUCTION SET SUMMARY

Symbol Description

WR	Working RAM
SR	Special register
SRP	Special register pair
ACC	Accumulator
acc.n	Accumulator bit n
R	Data Memory address from RAM000~RAM3FF
R.n	Bit n of data memory from RAM000~RAM3FF
RL	Data Memory address from RAM000~RAM1FF
RL.n	Bit n of data memory from RAM000~RAM1FF
P	RAM pointer rp0/rp1
@P	Memory (RAM) addressed by pointer rp0/rp1
@P.n	Memory (RAM) bit n addressed by pointer rp0/rp1
LUPC	ROM pointer, for use with look-up-table
@LUPC	Memory (ROM) addressed by pointer lupc
I or i:	Constant parameter
L	Branch or jump address
CF	Carry Flag
PC	Program Counter
==	Equal
! =	Not equal
$\wedge$	AND
$\vee$	OR
$\oplus$	Exclusive OR
$\leftarrow$	Transfer direction, result

Mnemonic	Des/Src operand	Affected flag	function description
ADD	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I*** ***ACC, #I***	CF	ACC ← R + ACC. ACC ← @P + ACC. ACC ← WR + I. ACC ← @P + I. ACC ← ACC + I.
ADDC	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I*** ***ACC, #I***	CF	ACC ← R + ACC + CF. ACC ← @P + ACC + CF. ACC ← WR + I + CF. ACC ← @P + I + CF. ACC ← ACC + I + CF.
ADDU	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I*** ***ACC, #I***	--	ACC ← R + ACC. ACC ← @P + ACC. ACC ← WR + I. ACC ← @P + I. ACC ← ACC + I.
ADDUR	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I***	--	ACC, R ← R + ACC. ACC, @P ← @P + ACC. ACC, WR ← WR + I. ACC, @P ← @P + I.
ADDR	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I***	CF	ACC, R ← R + ACC. ACC, @P ← @P + ACC. ACC, WR ← WR + I. ACC, @P ← @P + I.
ADDCR	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I***	CF	ACC, R ← R + ACC + CF. ACC, @P ← @P + ACC + CF. ACC, WR ← WR + I + CF. ACC, @P ← @P + I + CF.

SUB	***R, ACC***	CF	ACC ← R - ACC.
	@P, ACC		ACC ← @P - ACC.
	WR, #I		ACC ← WR - I.
	@P, #I		ACC ← @P - I.
	ACC, #I		ACC ← ACC - I.
SUBB	***R, ACC***	CF	ACC ← R - ACC - CF.
	@P, ACC		ACC ← @P - ACC - CF.
	WR, #I		ACC ← WR - I - CF.
	@P, #I		ACC ← @P - I - CF.
	ACC, #I		ACC ← ACC - I - CF.
SUBR	***R, ACC***	CF	ACC, R ← R - ACC.
	@P, ACC		ACC, @P ← @P - ACC.
	WR, #I		ACC, WR ← WR - I.
	@P, #I		ACC, @P ← @P - I.
SUBBR	***R, ACC***	CF	ACC, R ← R - ACC - CF.
	@P, ACC		ACC, @P ← @P - ACC - CF.
	WR, #I		ACC, WR ← WR - I - CF.
	@P, #I		ACC, @P ← @P - I - CF.
ANL	***R, ACC***	--	ACC ← R ∧ ACC.
	@P, ACC		ACC ← @P ∧ ACC.
	WR, #I		ACC ← WR ∧ I.
	@P, #I		ACC ← @P ∧ I.
	ACC, #I		ACC ← ACC ∧ I.
ORL	***R, ACC***	--	ACC ← R ∨ ACC.
	@P, ACC		ACC ← @P ∨ ACC.
	WR, #I		ACC ← WR∨ I.
	@P, #I		ACC ← @P∨ I.
	ACC, #I		ACC ← ACC ∨ I.

XRL	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I*** ***ACC, #I***	--	ACC ← R ⊕ ACC. ACC ← @P ⊕ ACC. ACC ← WR ⊕ I. ACC ← @P ⊕ I. ACC ← ACC ⊕ I.
ANLR	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I***	--	ACC, R ← R ∧ACC. ACC, @P ← @P∧ ACC. ACC, WR ← WR∧ I. ACC, @P ← @P ∧ I.
ORLR	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I***	--	ACC, R ← R∨ ACC. ACC, @P ← @P ∨ ACC. ACC, WR ← WR ∨ I. ACC, @P ← @P∨ I.
XRLR	***R, ACC*** ***@P, ACC*** ***WR, #I*** ***@P, #I***	--	ACC, R ← R ⊕ ACC. ACC, @P ← @P ⊕ ACC. ACC, WR ← WR ⊕ I. ACC, @P ← @P ⊕ I.
DEC	***R*** ***@P*** ***ACC***	CF	ACC, R ← R-1. ACC, @P ← @P-1. ACC ← ACC-1.
INC	***R*** ***@P*** ***ACC***	CF	ACC, R ← R+1. ACC, @P ← @P+1. ACC ← ACC+1.
SHLC	***R*** ***@P*** ***ACC***	CF	ACC.n, R.n ← R.n-1; ACC.0, R.0 ← 0; CF ← R.3. ACC.n, @P.n ← @P.n-1; ACC.0, @P.0 ← 0; CF ← @P.3. ACC.n ← ACC.n-1; ACC.0 ← 0; CF ← ACC.3.
SHRC	***R*** ***@P*** ***ACC***	CF	ACC.n, R.n ← R.n+1; ACC.3, R.3 ← 0; CF ← R.0. ACC.n, @P.n ← @P.n+1; ACC.3, @P.3 ← 0; CF ← @P.0. ACC.n ← ACC.n+1; ACC.3 ← 0; CF ← ACC.0.
RLC	***R*** ***@P*** ***ACC***	CF	ACC.n, R.n ← R.n-1; ACC.0, R.0 ← CF; CF ← R.3. ACC.n, @P.n ← @P.n-1; ACC.0, @P.0 ← CF; CF ← @P.3. ACC.n ← ACC.n-1; ACC.0 ← CF; CF ← ACC.3.

RRC	***R*** ***@P*** ***ACC***	CF	ACC.n, R.n ← R.n+1; ACC.3, R.3 ← CF; CF ← R.0. ACC.n, @P.n ← @P.n+1; ACC.3, @P.3 ← CF; CF ← @P.0. ACC.n ← ACC.n+1; ACC.3 ← CF; CF ← ACC.0.
SKBn ***(0 £ n £ 3)***	***R*** ***@P*** ***ACC***	------	IF R.n == 1, PC ← PC + 2. IF @P.n == 1, PC ← PC + 2. IF ACC.n == 1, PC ← PC + 2.
SKNBn ***(0 £ n £ 3)***	***R*** ***@P*** ***ACC***	------	IF R.n == 0, PC ← PC + 2. IF @P.n == 0, PC ← PC + 2. IF ACC.n == 0, PC ← PC + 2.
DSKZ	***R*** ***@P*** ***ACC***	--	ACC, R ← R-1; IF ACC == 0, PC ← PC+2. ACC, @P ← @P-1; IF ACC == 0, PC ← PC+2. ACC ← ACC-1; IF ACC == 0, PC ← PC+2.
DSKNZ	***R*** ***@P*** ***ACC***	--	ACC, R ← R-1; IF ACC != 0, PC ← PC+2. ACC, @P ← @P-1; IF ACC != 0, PC ← PC+2. ACC ← ACC-1; IF ACC != 0, PC ← PC+2.
JP	***L***	-----	PC ← L.
JC	***L***	-----	IF CF == 1, PC ← L.
JNC	***L***	-----	IF CF == 0, PC ← L.
JZ	***L***	-----	IF ACC == 0, PC ← L.
JNZ	***L***	-----	IF ACC != 0, PC ← L.
JBn	***L***	-----	IF ACC.n == 1, PC ← L.
JNBn	***L***	-----	IF ACC.n == 0, PC ← L.
CALL	***L***	-----	STACK ← PC+1; PC ← L; STACK ← STACK+1.
RTN		-----	STACK ← STACK-1; PC ← STACK;
HOLD		-----	ENTER THE HOLD MODE FOR POWER SAVING.
NOP		-----	NO OPERATION.
SET	***SR, #I***	-----	IF I.n == 1, SR.n ← 1.
CLR	***SR, #I***	-----	IF I.n == 1, SR.n ← 0.

MOV	***SR, ACC***	-----	SR ← ACC.
	SRP, #I		SRP ← #I.
	SR, #I		SR ← #I.
MOV	***SR, RL***	-----	SR ← RL.
	RL, SR		RL ← SR.
MOVA	***SR, RL***	-----	ACC, SR ← RL.
	RL, SR		ACC, RL ← SR.
MOV	***R, ACC***	-----	R ← ACC.
	ACC, R		ACC ← R.
	ACC, @LUPC		ACC ← @LUPC.
	ACC, @LUPC++		ACC ← @LUPC; LUPC ← LUPC+1.
MOV	***R, #I***	-----	R ← I.
	@P, #I		@P ← I.
MOV	***WR, R***	-----	WR ← R.
	R, WR		R ← WR.
MOV	***R, @P***	-----	R ← @P.
	@P, R		@P ← R.
	@RP1, @RP0		@RP1 ← @RP0.
	@RP0, @RP1		@RP0 ← @RP1.
MOV	***R, @P++***	-----	R ← @P; P ← P+1.
	@P++, R		@P ← R; P ← P+1.
	@RP1++, @RP0++		@RP1 ← @RP0; RP0 ← RP0+1; RP1 ← RP1+1.
	@RP0++, @RP1++		@RP0 ← @RP1; RP0 ← RP0+1; RP1 ← RP1+1.
MOV	***R, @LUPC***	-----	R ← @LUPC.
	@P, @LUPC		@P ← @LUPC.
	R, @LUPC++		R ← @LUPC; LUPC ← LUPC+1.
	@P++, @LPUC++		@P ← @LUPC; LUPC ← LUPC+1; P ← P+1.
MOVA	***R, #I***	-----	ACC, R ← I.
	@P, #I		ACC, @P ← I.
MOVA	***WR, R***	-----	ACC, WR ← R.
	R, WR		ACC, R ← WR.

MOVA	***R, @P*** ***@P, R*** ***@RP1, @RP0*** ***@RP0, @RP1***	-----	ACC, R ← @P. ACC, @P ← R. ACC, @RP1 ← @RP0. ACC, @RP0 ← @RP1.
MOVA	***R, @P++*** ***@P++, R*** ***@RP1++, @RP0++*** ***@RP0++, @RP1++***	-----	ACC, R ← @P; P ← P+1. ACC, @P ← R; P ← P+1. ACC, @RP1 ← @RP0; RP0 ← RP0+1; RP1 ← RP1+1. ACC, @RP0 ← @RP1; RP0 ← RP0+1; RP1 ← RP1+1.
MOVA	***R, @LUPC*** ***@P, @LUPC*** ***R, @LUPC++*** ***@P++, LPUC++***	-----	ACC, R ← @LUPC. ACC, @P ← @LUPC. ACC, R ← @LUPC; LUPC ← LUPC+1. ACC, @P ← @LUPC; LUPC ← LUPC+1; P ← P+1.
INC	***RP0*** ***RP1*** ***LUPC***	-----	RP0 ← RP0+1. RP1 ← RP1+1. LUPC ← LUPC+1.
CLR	***DIV***	-----	FLAG1.2 (DIVR bit) ← 0.
CLR	***WDT***	-----	FLAG1.1 (WDTR bit) ← 0.
EN	***INT***	-----	MR2.0 (INTEN bit) ← 1.
DIS	***INT***	-----	MR2.0 (INTEN bit) ← 0.
LCDON		-----	MR0.1 (LCDEN bit) ← 1.
LCDOFF		-----	MR0.1 (LCDEN bit) ← 0.

LD	***LUP***	***#i3i2i1i0***	MOV LUP3,#i3
			MOV LUP2,#i2
			MOV LUP1,#i1
			MOV LUP0,#i0

LD	***RP0***	***#i2i1i0***	MOV RP0H,#i2
			MOV RP0M,#i1
			MOV RP0L,#i0

LD	***RP1***	***#i2i1i0***	MOV RP1H,#i2
			MOV RP1M,#i1
			MOV RP1L,#i0

•Replacement table for instruction set

Item	Replacement
WR	WR00, WR01, ..., WR0F, WR10, WR11, ..., WR3F
R	RAM000, RAM001, ..., RAM00F, RAM010, ..., RAM37F
RL	RAM000, ..., RAM1FE, RAM1FF
SR	TM0H, TM0L, TM1H, TM1L, TMC1H, TMC1L, EVFH, EVFL, HEFH, HEFL, IEFH, IEFL, HCFH, HCFL, PEFH, PEFL, RP0H, RP0M, RP0L, RP1H, RP1M, RP1L, MLDH, MLDL, SPCH, SPCL, LCDM2, LCDM1, CF, FLAG1, FLAG0, MR2, MR1, MR0, SCR, LUP3, LUP2, LUP1, LUP0, LUC, WRPAGE, PSR1, PSR0, PM2, PM1, PM0, PORTA, PORTB, PORTC, PORTD, PORTE
SRP	HEF, IEF
P	RP0, RP1
I	8-bits or 4-bits immediate data

•Read/write property of each register

Name	Type	Name	Type	Name	Type
TM0H	w	TM0L	w	TM1H	w
TM1L	w	TMC1H	r	TMC1L	r
EVFH	r,c	EVFL	r,c	HEFH	r,w,s,c
HEFL	r,w,s,c	IEFH	r,w,s,c	IEFL	r,w,s,c
HCFH	r	HCFL	r	PEFH	r,w,s,c
PEFL	r,w,s,c	RP0H	r,w	RP0M	r,w
RP0L	r,w	RP1H	r,w	RP1M	r,w
RP1L	r,w	MLDH	w	MLDL	w
SPCH	w	SPCL	w	LCDM1	w
CF	r,s,c	FLAG1	c	FLAG0	r,w,s,c
MR3	w	MR2	s,c	MR1	w
MR0	r,w,s,c	SCR	r,w,s,c	LUP3	r,w
LUP2	r,w	LUP1	r,w	LUP0	r,w
LUC	r	WRPAGE	r,w	PM2	r,w,s,c
PM1	r,w,s,c	PM0	r,w,s,c	PSR1	r, clr-all
PSR0	r, clr-all	PORTA	r, w	PORTB	r,w
PORTC	r	PORTD	r	PORTE	w

w can use "*MOV SR, R* " instruction.. r can use "*MOV R, SR* " instruction

c can use " *CLR SR* ' instruction. s can use " *SET SR* " instruction

Winbond Electronics Corp.

Winbond Electronics Corp.

Headquarters
No. 4, Creation Rd. III,
Science-Based Industrial Park,
Hsinchu, Taiwan
TEL: 886-3-5770066
FAX: 886-3-5792697
http://www.winbond.com.tw/
Voice & Fax-on-demand: 886-2-7197006

Taipei Office
11F, No. 115, Sec. 3, Min-Sheng East Rd.,
Taipei, Taiwan
TEL: 886-2-7190505
FAX: 886-2-7197502

Winbond Electronics (H.K.) Ltd.
Rm. 803, World Trade Square, Tower II,
123 Hoi Bun Rd., Kwun Tong,
Kowloon, Hong Kong
TEL: 852-27513100
FAX: 852-27552064

Winbond Electronics North America Corp.
Winbond Memory Lab.
Winbond Microelectronics Corp.
Winbond Systems Lab.
2730 Orchard Parkway, San Jose,
CA 95134, U.S.A.
TEL: 1-408-9436666
FAX: 1-408-9436668