十速

TM57ML40

DATA SHEET

Rev V1.9

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

AMENDMENT HISTORY

Version	Date	Description
V1.0	Mar, 2011	New release.
V1.1	Dec, 2011	Add ordering information.
V1.2	Jan, 2012	1. Add Electrical Characteristics in Features section. 2. Add LVR description in Reset section.
V1.3	Jul, 2012	Modify mnemonic description. Add pin current data in Electrical Characteristics section.
V1.4	Mar, 2013	Modify LCD description.
V1.5	Jun, 2013	Add supported EV board on ICE. Modify System block diagram.
V1.6	Jul, 2013	Modify fast clock selection and operation speed in Features section.
V1.7	Dec, 2013	Add I/O port section.
V1.8	Jan, 2016	1. LVR table update (p18) 2. DC Characteristics update (p77) 3. New LVR vs Temperature in Characteristic Graphs(p77)
V1.9	July, 2017	1. Add LQFP-64 Package information (p10, p79~80)

CONTENTS

FEATURES

1. 37 Instructions, two clock cycles execution

2. 4Kx14 MTP (Multi-Time Programmable) ROM

3. 368-byte SRAM

- 176 bytes on F-Plane
- 160 bytes on R-Plane
- 32 bytes LCD RAM

4. 8-level Stack

5. ISP (In-System Programming) uses 5 wires (V_{DD}, GND, PA1 (SCL), PA0 (SDA), V_{PP})

6. Individual Interrupt Vector

7. Auto Push/Pop WREG and STATUS (selectable by register control)

8. 2 Independent 8-bit PWMs

- PWM0 with prescaler/period-adjustment/buffer-reload/pos-neg-output (REM)
- PWM1 is simple duty controlled PWM (LED)

9. Buzzer output

10. Independent RC Oscillating WatchDog Timer (or Wake-up Timer) with 4 adjustable Reset/Interrupt Time

- 128 ms/32 ms/2 ms/1 ms

11. Independent Timers

Timer0 is 8-bit with 8-bit prescaler, Counter/Reload/Read/Write/Capture/Interrupt function Timer1 is 16-bit with Capture/Reload/Interrupt/Read/clear/set/stop function

Timer2 is used for LCD clock generation and real time 32768 Hz interrupt with clear function

12. Resistance to Frequency Converter (RFC)

- R type sensor
- 2 input pins (CX0, CX1), 4 output pins (RFC0~3)
- Built-in 16-bit RFC counter with Read/clear/stop/interrupt function
- With Timer0/Timer1 precision control mode

13. Max. 33 Programmable I/O pin

- CMOS Output
- Pseudo-Open-Drain or Open-Drain Output
- Schmitt Trigger Input

14. 4 Fast Clock selections

- Fast Crystal (1~12 MHz)
- PLL, PLL freq. = [32768*4*(64+M)] / N, (N=1, 2, 4, 8), (M=0~63)
- Fast Internal RC (8M/4M/2M/512 KHz)
- External RC

15. 4 Slow Clock selections

- Slow Crystal (32768 Hz)
- External RC
- Low speed Internal RC
- Touch Key

16. 4 Power Saving Operation Modes

- Fast Mode: Slow Clock can disable or enable
- Slow Mode: Fast Clock stop, CPU running
- Idle Mode: Slow Clock running, CPU off, LCD can be disabled or enabled, Timer2 running
- Stop Mode: All Clock stop, wake-up Timer disable or enable

17. 15 Maskable Interrupt Sources

- 4 External Interrupt pins: 2-pin negative edge trigger, 2-pin positive or negative edge trigger
- Timer0, Timer1, Timer2, Wake-up Timer Interrupt
- PWM Interrupt
- RFC Interrupt

18. Pin wake up function (PB7~PB2)

19. LCD/LED Controller/Driver

LCD

- 8com x 28seg/7com x 29seg/6com x 30seg/5com x 31seg
  4com x 32seg/3com x 32seg/2com x 32seg/1com x 32seg
- 1/2, 1/3 Bias
- 4 Brightness Level selections
- 4 Current Drive selections

LED

- 8com~1com x 28seg

20. 15-channel Touch Key

21. 2-Level Low Voltage Reset: 1.7V/2.5V (Can be disabled)

22. Operation voltage: Low Voltage Reset to 3.6V

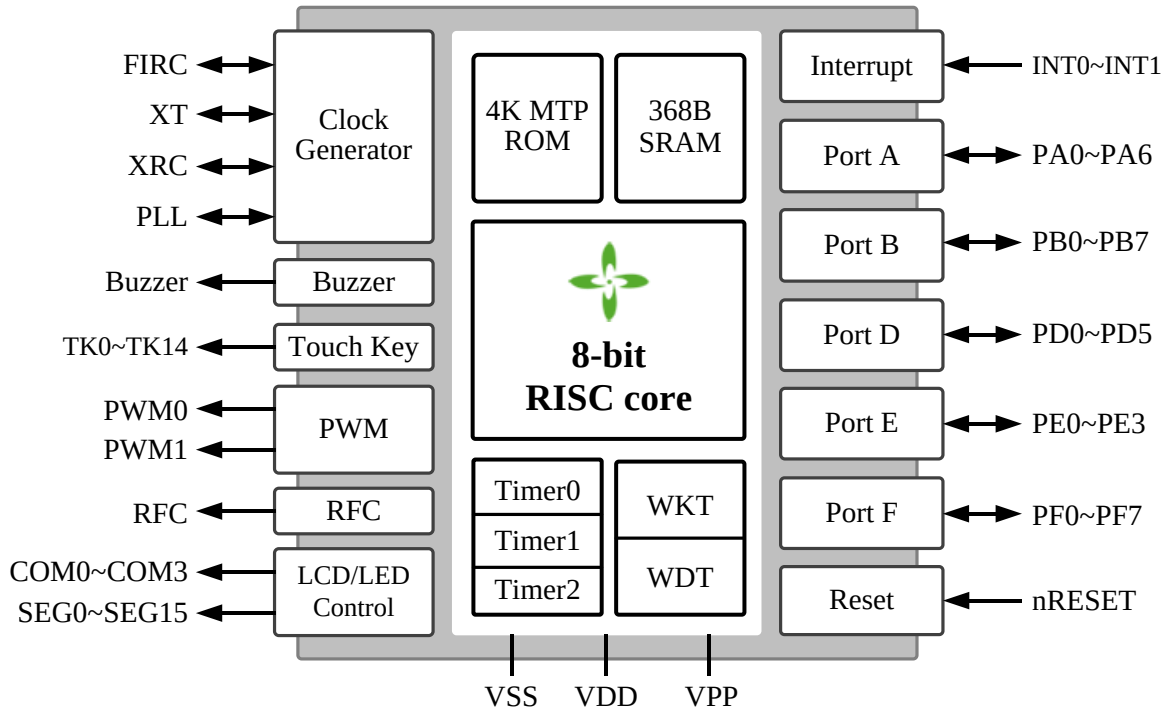
- fosc =4 MHz, 2.2V ~ 3.6V
- fosc =8 MHz, 2.3V ~ 3.6V
- fosc =12 MHz, 2.8V ~ 3.6V

23. Operation speed: 12 MHz @Vdd=3.3V; 8 MHz @Vdd=2.0V

24. Supported EV board on ICE

EV board: EV2785

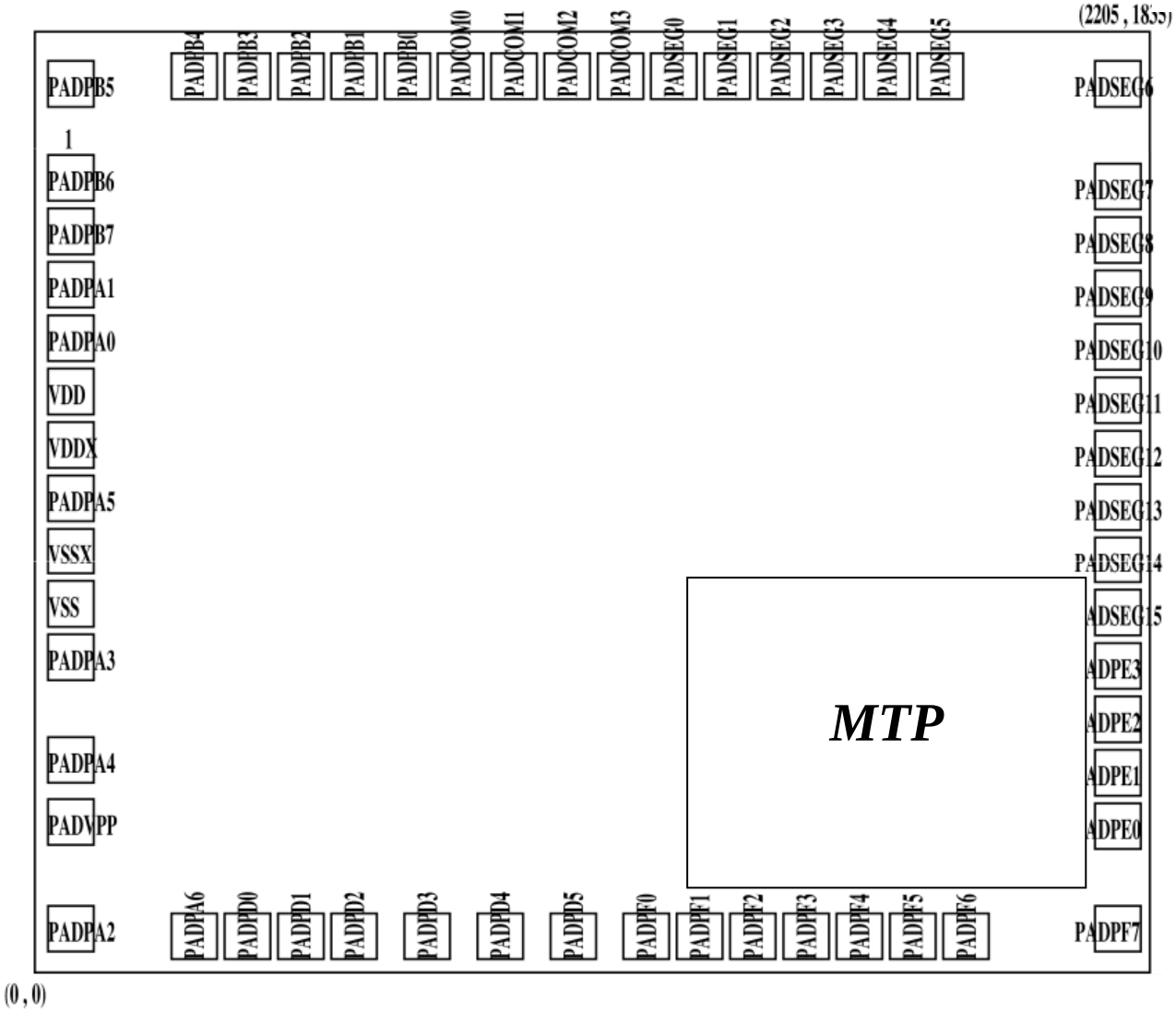
System Block Diagram



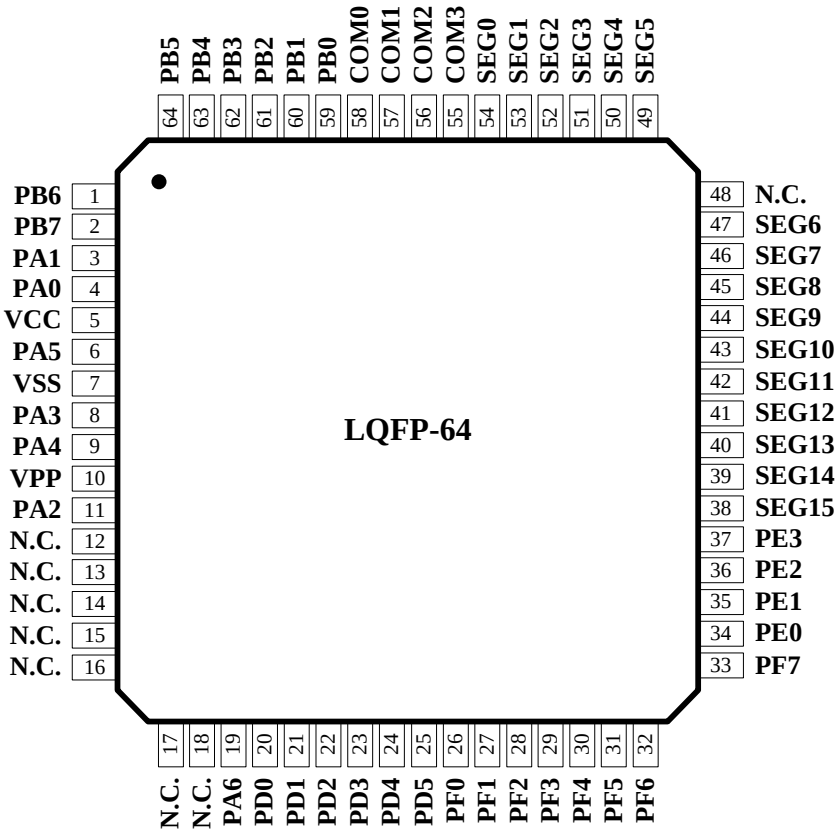
TM57ML40 Block Diagram

< Figure 1 System Block Diagram >

Pad Bonding Diagram



PIN ASSIGNMENT



Pin Descriptions

	PIN NAME	Type	Output			Ext.Interrupt	LCD/LED	Touch-Key	RFC	Misc
			OD	POD	PP					
1	PB6/RFC3/TK6	I/O	O		O			O	O	
2	PB7/CX1/TK7	I/O	O		O			O	O	
3	PA1/INT1/TK8	I/O		O	O	O		O		
4	PA0/INT0/TK9	I/O		O	O	O		O		
5	VDD	P								
6	VDDX	P								
7	PA5/FLT	I/O	O		O					
8	VSSX	P								
9	VSS	P								
10	PA3/XO	I/O	O		O					
11	PA4/XRC/XI	I/O	O		O					
12	VPP/RSTN	I								
13	PA2/T0CKI/TK10	I/O		O	O			O		T0CKI
14	PA6/PWM0	I/O	O		O					PWM0
15	PD0/PWM1	I/O	O		O					PWM1
16	PD1/BUZ	I/O	O		O					BUZ
17	PD2/COM4/SEG31	I/O	O		O		O			
18	PD3/COM5/SEG30	I/O	O		O		O			
19	PD4/COM6/SEG29	I/O	O		O		O			
20	PD5/COM7/SEG28	I/O	O		O		O			
21	PF0/TK11/SEG27	I/O	O		O		O	O		
22	PF1/TK12/SEG26	I/O	O		O		O	O		
23	PF2/TK13/SEG25	I/O	O		O		O	O		
24	PF3/TK14/SEG24	I/O	O		O		O	O		
25	PF4/SEG23	I/O	O		O		O			
26	PF5/SEG22	I/O	O		O		O			
27	PF6/SEG21	I/O	O		O		O			
28	PF7/SEG20	I/O	O		O		O			
29	PE0/SEG19	I/O	O		O		O			
30	PE1/SEG18	I/O	O		O		O			
31	PE2/SEG17	I/O	O		O		O			
32	PE3/SEG16	I/O	O		O		O			
33	SEG15	O					O			

	PIN NAME	Type	Output			Ext.Interrupt	LCD/LED	Touch-Key	RFC	Misc
			OD	POD	PP					
34	SEG14	O					O			
35	SEG13	O					O			
36	SEG12	O					O			
37	SEG11	O					O			
38	SEG10	O					O			
39	SEG9	O					O			
40	SEG8	O					O			
41	SEG7	O					O			
42	SEG6	O					O			
43	SEG5	O					O			
44	SEG4	O					O			
45	SEG3	O					O			
46	SEG2	O					O			
47	SEG1	O					O			
48	SEG0	O					O			
49	COM3	O					O			
50	COM2	O					O			
51	COM1	O					O			
52	COM0	O					O			
53	PB0/TK0/INT2	I/O	O		O	O		O		
54	PB1/TK1/INT3/CAPT	I/O	O		O	O		O		Capture
55	PB2/TK2/CX0	I/O	O		O			O	O	
56	PB3/TK3/RFC0	I/O	O		O			O	O	
57	PB4/TK4/RFC1	I/O	O		O			O	O	
58	PB5/TK5/RFC2	I/O	O		O			O	O	

Symbol: O.D. = Open Drain
P.O.D. = Pseudo Open Drain
P.P. = Push-Pull Output

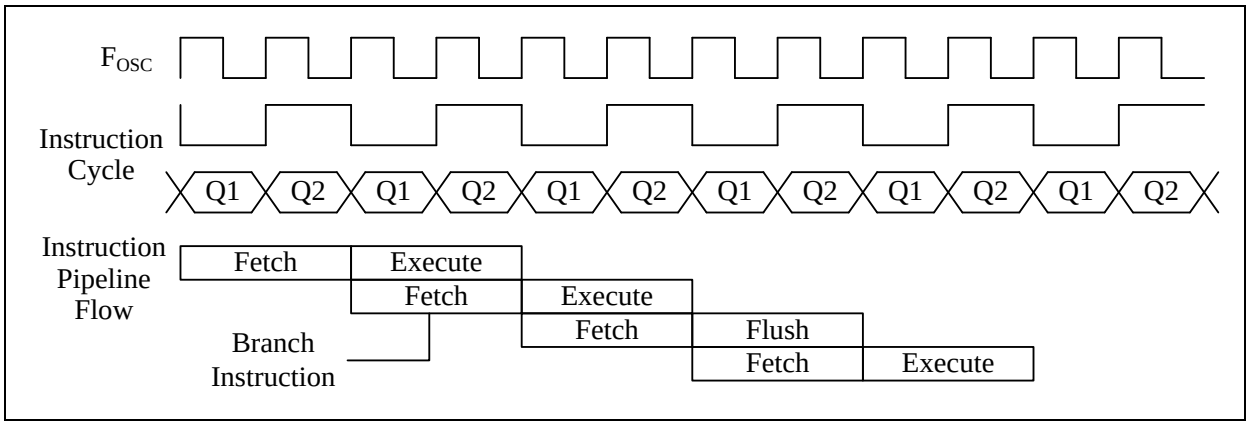
Name	In/Out	Pin Description
PA2–PA0	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "**pseudo-open-drain**" output. Pull-up resistors are assignable by software.
PA6-PA3 PB7-PB0 PD5-PD0 PE3-PE0 PF7-PF0	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or **open-drain** output. Pull-up resistors are assignable by software.
RSTN	I	External active low reset
XI, XO	-	Crystal/Resonator oscillator connection for system clock
XRC	-	External RC oscillator connection for system clock
VDD,VDDX VSS,VSSX	P	Power input pin and ground
INT0~INT3	I	External interrupt input
TK0~TK14	I	Touch Key input
COM0~COM7 SEG0~SEG23	O	LCD/LED common and segment output
T0CKI	I	Timer0's input in counter mode
CAPT	I	Timer0/Timer1 Capture input
PWM0	O	PWM0 positive and negative outputs (Period/Duty adjustable)
PWM1	O	PWM1 output (fixed period, duty adjustable)
FLT	I	PLL FLT input
CX0~CX1	I	Resistance to frequency converter input
RFC0~RFC3	O	Resistance to frequency converter output

Functional Description

1. CPU Core

1.1 Clock Scheme and Instruction Cycle

The system clock is internally divided by two to generate Q1 state and Q2 state for each instruction cycle. The Programming Counter (PC) is updated at Q1 and the instruction is fetched from program ROM and latched into the instruction register in Q2. It is then decoded and executed during the following Q1-Q2 cycle. Branch instructions take two cycles since the fetch instruction is 'flushed' from the pipeline, while the new instruction is being fetched and then executed.

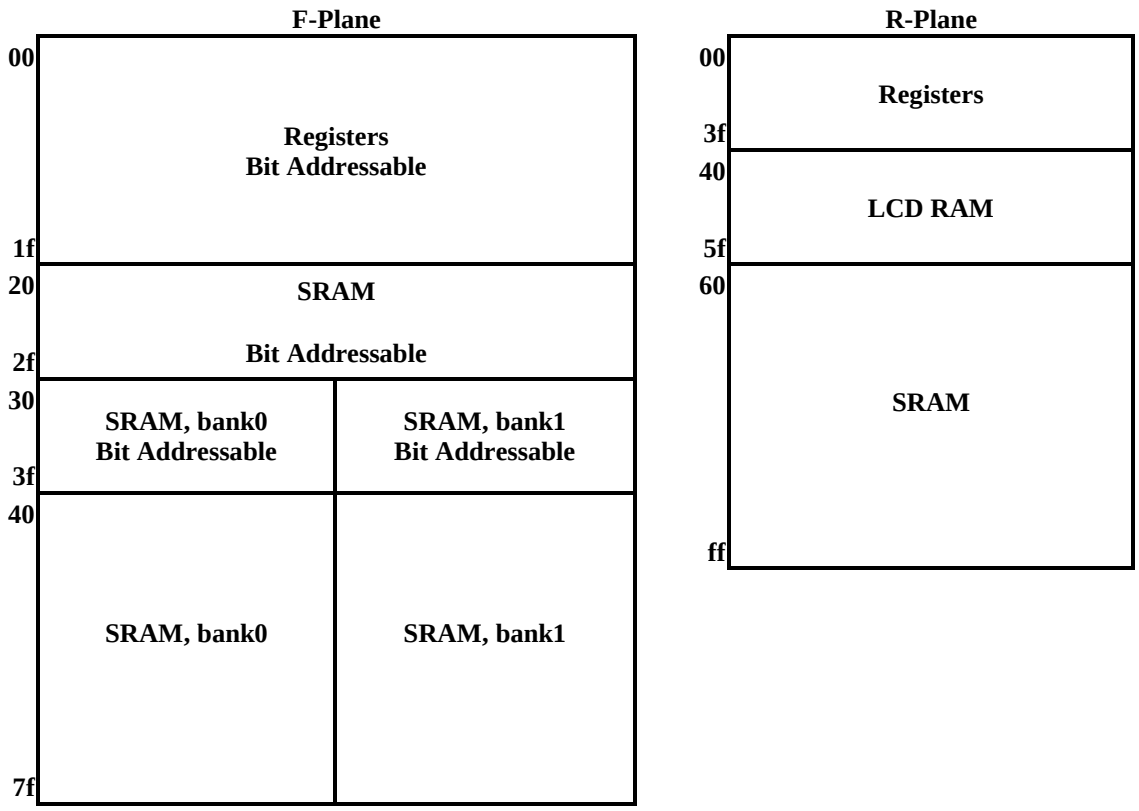


1.2 Addressing Mode

There are two Data Memory Planes in CPU, R-Plane and F-Plane. The F-Plane supports rich instructions operation, such as ADDWF, INCF, MOVWF,..., while the R-Plane only supports MOVWR and MOVRW instructions to exchange data between R-Plane and W-Register.

The lower locations of R-Plane are reserved for the read only registers. Above the registers are the LCD RAM and static RAM. R-plane can be indirect accessed via RSR register (F-plane 07h) and INDR (R-plane 00h).The INDR register is not a physical register. Addressing INDR actually addresses the register whose address is contained in the RSR register (RSR is a pointer).

The lower locations of F-Plane are reserved for the SFR. Above the SFR is General Purpose Data Memory, implemented as static RAM. F-Plane can be addressed directly or indirectly. Indirect Addressing is made by INDF register. The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a pointer). The first half of F-Plane is bit-addressable, while the second half of F-Plane is not bit-addressable.



1.3 Programming Counter (PC) and Stack

The Programming Counter is 12-bit wide capable of addressing a 4K x 14 program ROM. As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC value is normally increased by one except the followings. The Reset Vector (000h) and the Interrupt Vectors (from 001h to 008h) are provided for PC initialization and Interrupts. For CALL/GOTO instructions, PC loads the 12 bits address from instruction word. For RET/RETI/RETLW instructions, PC retrieves its content from the top level STACK. For the other instructions updating PC [7:0], the PC [12:8] keeps unchanged. The STACK is 12-bit wide and 8-level in depth. The CALL instruction and Hardware interrupt will push STACK level in order. While the RET/RETI/RETLW instruction pops the STACK level in order.

1.4 ALU and Working (W) Register

The ALU is 8-bit wide and capable of addition, subtraction, shift and logical operations. In two-operand instructions, typically one operand is the W register, which is an 8-bit non-addressable register used for ALU operations. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either W register or a file register. Depending on the instruction executed, the ALU may affect the values of Carry (C), Digit Carry (DC), and Zero (Z) Flags in the STATUS register. The C and DC flags operate as a /Borrow and /Digit Borrow, respectively, in subtraction.

Note: /Borrow represents inverted of Borrow register.

/Digit Borrow represents inverted of Digit Borrow register.

The W register can be automatically stored into the internal memory when interrupt and recall when exit from interrupt. This functionality is optional and can be enabled or disabled via ATOSAVE (R-Plane CLKCTRL.4) bit.

1.5 STATUS Register

This register contains the arithmetic status of ALU and the Reset status. The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. It is recommended, therefore, that only BCF, BSF and MOVWF instructions be used to alter the STATUS Register because these instructions do not affect those bits.

PD bit is '1' when SLEEP instruction is executed. It can be cleared either power off-on to generate Power-On Reset or by executing CLRWDT.

TO bit is '1' when WDT Timeout is happened. It can be cleared if SLEEP, power off-on, or CLRWDT is executed.

The STATUS register can be automatically stored into the internal memory when interrupt and restored when exit from interrupt. This functionality is optional and can be enabled or disabled via ATOSAVE (R-plane CLKCTRL.4) bit.

STATUS	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset Value	**–**	**0**	**0**	**–**	**–**	**0**	**0**	**0**
R/W	**–**	**R/W**	**R/W**	**R**	**R**	**R/W**	**R/W**	**R/W**

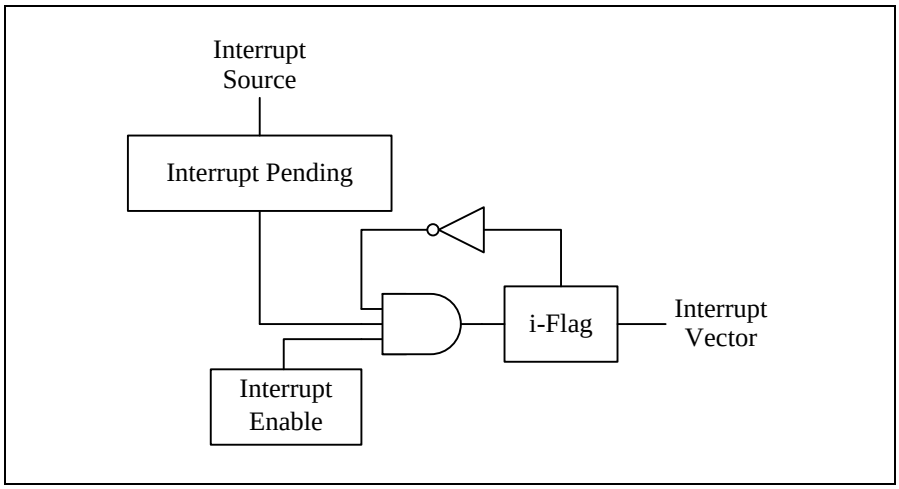
<table>
<tr><td>Bit</td><td colspan="2">Description</td></tr>
<tr><td>6</td><td colspan="2">GBIT: General Purpose Bit
No special function. User can use it as general purpose bit.</td></tr>
<tr><td>5</td><td colspan="2">RAMBK: RAM Bank
0: RAM Bank 0
1: RAM Bank 1</td></tr>
<tr><td>4</td><td colspan="2">TO: Time Out
0: after Power On Reset, LVR Reset, External active low reset or CLRWDT/SLEEP instruction
1: WDT time out occurred</td></tr>
<tr><td>3</td><td colspan="2">PD: Power Down
0: after Power On Reset, LVR Reset, External active low reset or CLRWDT instruction
1: after SLEEP instruction</td></tr>
<tr><td>2</td><td colspan="2">Z: Zero Flag
0: the result of a logic operation is not zero
1: the result of a logic operation is zero</td></tr>
<tr><td rowspan="3">1</td><td colspan="2">DC: Decimal Carry Flag or Decimal/Borrow Flag</td></tr>
<tr><td>ADD instruction</td><td>SUB instruction</td></tr>
<tr><td>1: a carry from the low nibble bits of the result occurs
0: no carry</td><td>1: no borrow
0: a borrow from the low nibble bits of the result occurs</td></tr>
<tr><td rowspan="3">0</td><td colspan="2">C: Carry Flag or Borrow Flag</td></tr>
<tr><td>ADD instruction</td><td>SUB instruction</td></tr>
<tr><td>1: a carry occurs from the MSB
0: no carry</td><td>1: no borrow
0: a borrow occurs from the MSB</td></tr>
</table>

1.6 Interrupt

The TM57ML40 has 1 level, 8 vectors and 10 interrupt sources. Each interrupt source has its own enable control bit. An interrupt event will set its individual interrupt flag, no matter its interrupt enable control bit is 0 or 1. Because TM57ML40 has 8 vectors, there is not an interrupt priority register. Priority of each interrupt is equal and the device does not support nested interrupt. Another interrupts can be executed only if the current interrupt is exited (that is, RETI instruction is executed). Although the interrupts do not have priority, however, when exit from current interrupt, if there are more than 2 interrupts happened, the priority of the interrupt is TM2 > TM1 > TM0 > XINTA > XINTB > WKT > PWM0 > RFC

No	Address	Source	Description	WakeUp
1	001	Timer2	Timer2 Count Match	Yes
2	002	Timer1	Timer1 Counter Overflow	
3	003	Timer0	Timer0 Counter Overflow	
4	004	PWM0	PWM0 Period Finish	
5	005	WKT	Wakeup Timer Match (if WDT disable)	Yes
6	006	XINTA	PA0, PA1, PB0 falling interrupt (PA0 is rising/falling selectable)	Yes
7	007	XINTB	PB1 rising/falling selectable interrupt	Yes
8	008	RFC	RFC Counter Overflow	Yes

If the corresponding interrupt enable bit has been set (INTE), it would trigger CPU to service the interrupt. CPU accepts interrupt in the end of current executed instruction cycle. In the mean while, a "CALL 00n" (n ranges from 1 to 8) instruction is inserted to CPU, and i-flag is set to prevent recursive interrupt nesting. The i-flag is cleared in the instruction after the "RETI" instruction. That is, at least one instruction in main program is executed before service the pending interrupt. The interrupt event is level triggered. F/W must clear the interrupt event register while serving the interrupt routine.



2. Chip Operation Mode

2.1 Reset

The TM57ML40 can be RESET in four ways.

- Power-On-Reset
- Low Voltage Reset (LVR)
- External Pin Reset (RSTN)
- Watchdog Reset (WDT)

After Power-On-Reset, all system and peripheral control registers are then set to their default hardware Reset values. And the clock source, LVR level and chip operation mode are selected by the SYSCFG register value.

The Low Voltage Reset features static reset when supply voltage is below a threshold level. There are two threshold levels can be selected. The LVR's operation mode is defined by the SYSCFG register.

There are two voltage selections for the LVR threshold level, one is higher level which is suitable for application with V_{DD} is more than 3.3V, while another one is suitable for application with V_{DD} is less than 3.3V. See the following LVR Selection Table; user must also consider the lowest operating voltage of operating frequency.

LVR Selection Table:

LVR Threshold Level	Consider the operating voltage to choose LVR
LVR2.5	3.6V > V_{DD} > 3.3V
LVR1.7	V_{DD} is wide voltage range

Different Fsys have different system minimum operating voltage, reference to Operating Voltage of DC characteristics, if current system voltage is low than minimum operating voltage and lower LVR is selected, then the system maybe enter dead-band and error occur.

The External Pin Reset and Watchdog Reset can be disabled or enabled by the SYSCFG register. These two resets also set all the control registers to their default reset value. The TO/PD flag is not affected by these resets.

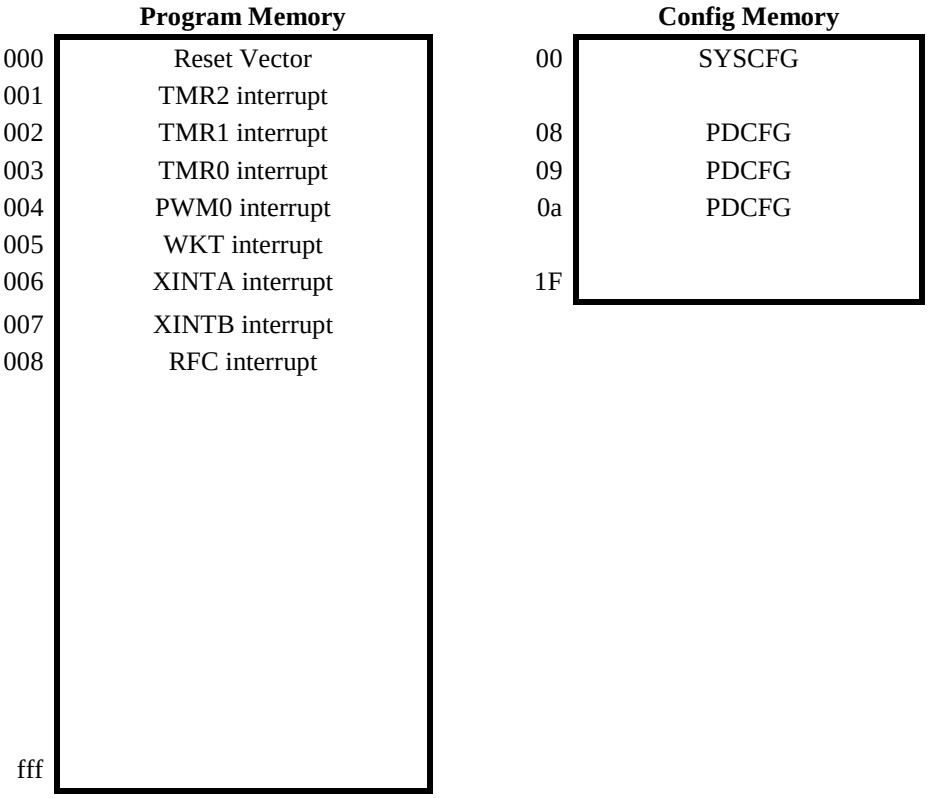
2.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at MTP INFO area. The SYSCFG determines the option for initial condition of MCU. It is written by MTP Writer only. User can select clock source, LVR threshold voltage and chip operation mode by SYSCFG register. The 13th bit of SYSCFG is code protection selection bit. If this bit is 1, the data in MTP will be protected, when user reads MTP.

Bit	**13~0**	
Default Value	**00_0000_0000_0000**	
Bit	Description	
13	**PROTECT**: Code Protection Selection	
	1	Code protection
	0	No protect
12	Not used	
11-10	**LVR**: LV reset mode	
	11	LVR threshold is 1.7V, always enabled
	10	LVR threshold is 1.7V, disabled at Sleep mode
	01	LVR threshold is 2.5V, always enabled
	00	LVR disable
9-8	**CLKS**: Fast Clock Source Selection	
	11	Fast Xtal
	10	PLL
	01	Fast Internal RC (512 kHz~8 MHz)
	00	External RC
7	Not used	
6	**WDTE**: WDT Reset Enable	
	1	Enable WDT Reset, Disable WKT Timer
	0	Disable WDT Reset, Enable WKT Timer
5	Not used	
4-0	**FIRCF**: Fast Internal RC Frequency adjustment control	

2.3 MTP ROM

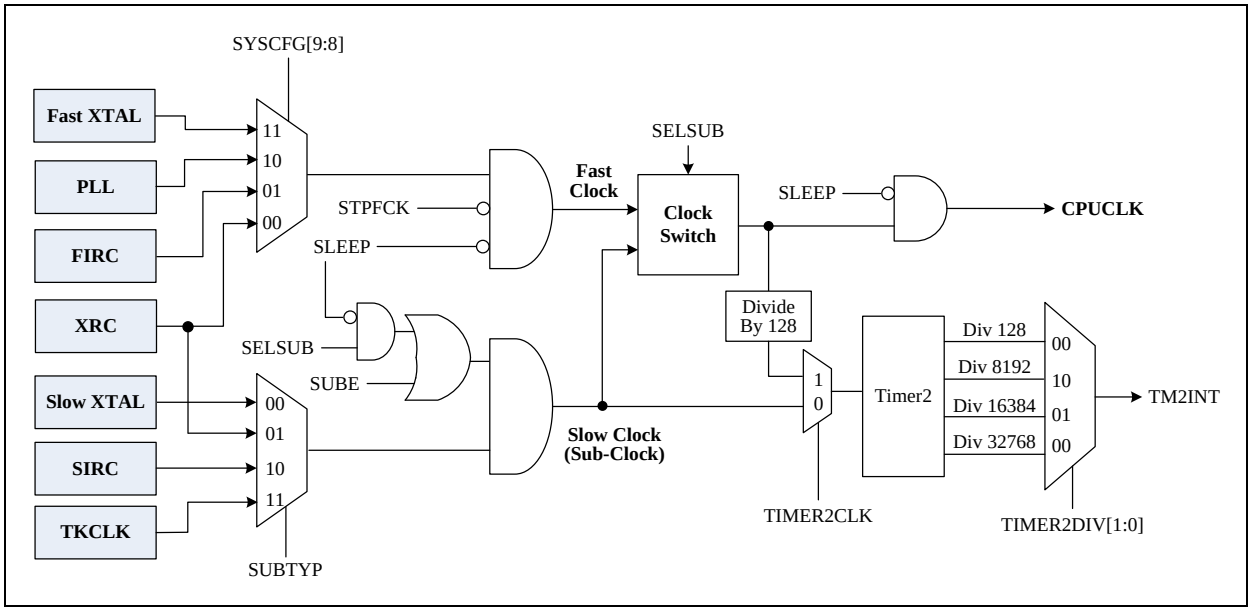
The MTP (Multi-Time Programmable) ROM of this device is 4K words, with an extra INFO area to store the SYSCFG and manufacture data. The MTP ROM can be written multi-times and can be read as long as the PROTECT bit of SYSCFG is not set. The SYSCFG can be read no matter PROTECT is set or cleared, but can be written only when PROTECT is not set or MTP ROM is blank. That is, unprotect the PROTECT bit can be done only if the Program ROM area is blank. The tenx certified writer can do the above actions with the sophisticated software.



2.4 Dual Clock System and Operation Mode Selection

TM57ML40 is designed with dual-clock system. There are seven kinds of clock source, Fast XTAL clock, XRC clock, Slow XTAL clock, PLL clock, SIRC (Slow Internal RC oscillator) clock, FIRC (Fast Internal RC oscillator), and TKCLK (Touch Key clock). Each clock source can be applied to CPU kernel as system clock. When in IDLE mode, only Slow Clock can be configured to keep oscillating to provide clock source to Timer2 block.

Clock Scheme Block Diagram



Fast Mode

After power on or reset, TM57ML40 enters Fast Mode. In Fast Mode, TM57ML40 can select Fast XTAL, PLL, Fast XRC or FIRC as its CPU clock by SYSCFG bit9 and bit8 setting. Besides, firmware can also enable or disable the Slow Clock for the Timer2 system operating.

In this mode, the program is executed using Fast Clock as CPU clock. The Timer0, Timer1, PWM0 blocks are also driven by Fast Clock. Timer2 can also be driven by Fast Clock by setting TIMER2CLK to “1”.

Slow Mode

In Slow Mode, TM57ML40 can select Slow XTAL, Slow XRC, SIRC or TKCLK as its CPU clock by R-Plane control register (SUBTYP). In this mode, the Fast Clock is stopped and Slow Clock is enabled for power saving. All peripheral blocks Clock sources are Slow Clock in the Slow Mode.

Idle Mode

If Slow Clock is enabled before executing the SLEEP instruction, the TM57ML40 enters the “Idle Mode”. In this mode, the Slow Clock will continue running to provide clock to Timer2 block. CPU stop fetching code and all blocks are stop except Timer2 related circuits.

Stop Mode

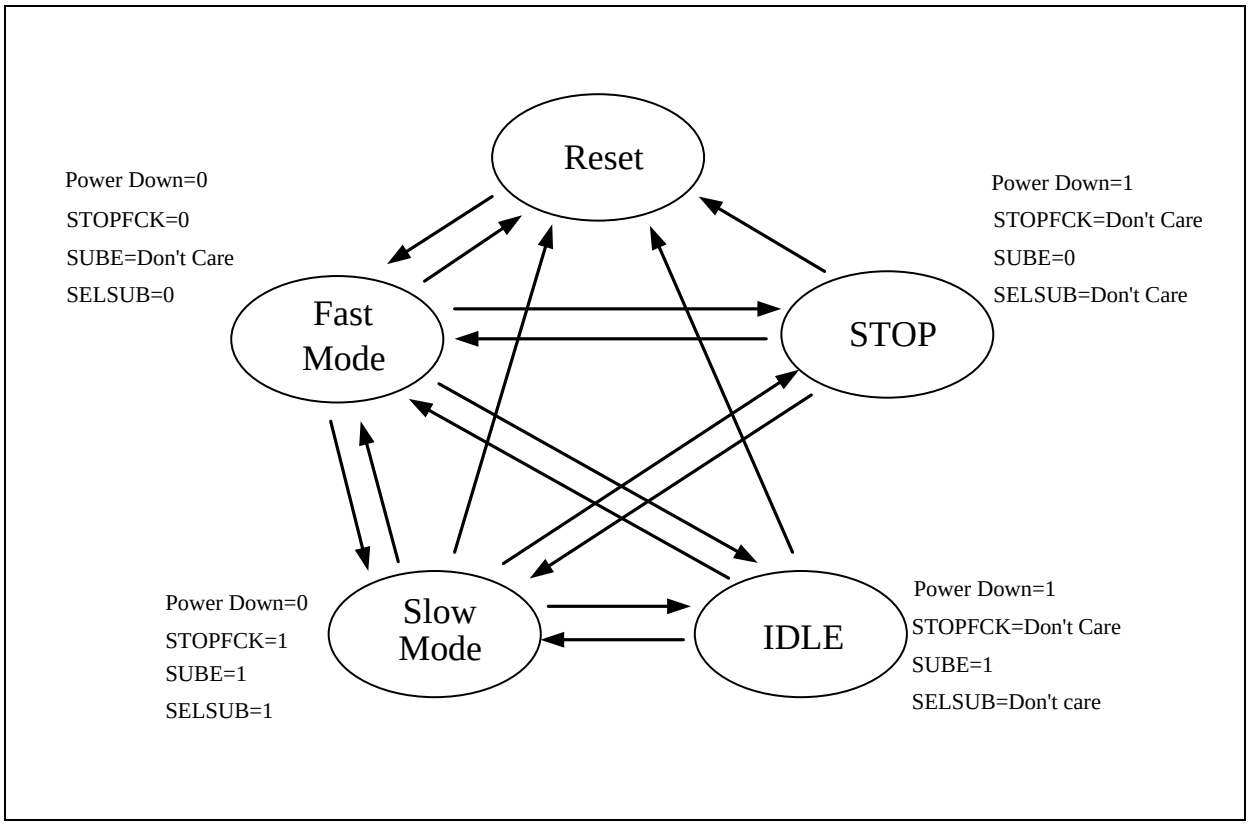
If Slow Clock is disabled before executing the SLEEP instruction, every block is turned off and the TM57ML40 enters the "Stop Mode". Stop mode is similar to IDLE mode. The difference is all clock oscillators either Fast or Slow is power down and no clock is generated. Only the on-chip Wake-up Timer is counting for wakeup if the WDTE bit of SYSCFG is "0". WatchDog Timer and Wake-up Timer share one physical timer, it means if WDTE is equal to 1, the Wake-up Timer function is disabled. Conversely, if the WDTE is cleared to "0" and WKTIE is set to "1", the Wake-up Timer is enabled and will consume little power to count when in STOP mode.

TM57ML40 is operated in one of four modes; Fast Mode, Slow Mode, IDLE mode, and STOP mode.

Operation Mode	Oscillator	CPUCLK	Fast Clock	Slow Clock	Timer0	Timer2	Wake Function
Fast	FIRC, PLL, FXT, XRC	Fast Clock	Running	Running /Stop	Running	Running /Stop	X
Slow	SXT, XRC, SIRC, TKCLK	Slow Clock	Stop	Running	Running	Running	X
Idle	SXT, XRC, SIRC, TKCLK	CPU stops	Stop	Running	Stop	Running	TM2/WKT /IO(1)
Stop	WKT(1)	CPU stop	Stop	Stop	Stop	Stop	WKT/IO(1)

(1) if function is enabled

Modes Transition Diagram



Fast Mode transits to Slow Mode

The following steps are suggested to be executed by order when Fast Mode transits to Slow Mode:

1. Enable Slow Clock (SUBE=1)
2. Switch to Slow Clock (SELSUB=1)
3. Stop Fast Clock (STOPFCK=1)

Note that if the SUBE=0, the Slow Clock oscillator can also be enabled if SELSUB=1 while not in power-down mode. Once the SLEEP is executed and SUBE=0, the Slow Clock oscillator will be turned off immediately and the chip is entering Stop mode, neither Fast nor Slow Clock is oscillating to achieve power saving.

Slow Mode transits to Fast Mode

The following steps are suggested to be executed by order when Slow Mode transits to Fast Mode:

1. Enable Fast Clock (STOPFCK=0)
2. Switch to Fast Clock (SELSUB=0)
3. Stop Slow Clock (SUBE=0) -------- this is optional. Slow Clock can keep oscillating when in Fast mode.

IDLE Mode

The IDLE mode can be configured by following setting in order:

1. SUBE=1
2. SLEEP

Idle mode can be woken up by XINT, PAWKUP, PBWAKP, Wake-up Timer, RFC, and **Timer2** interrupt.

STOP Mode

The STOP mode can be configured by following setting in order:

1. SUBE=0
2. SLEEP

STOP mode can be woken up by XINT, PAWKUP, PBWAKP, and Wake-up Timer.

3. Peripheral Functional Block

3.1 Watchdog (WDT) /Wakeup (WKT) Timer

The WDT and WKT share the same internal RC Timer. The overflow period of WDT/WKT can be selected from 3.5 ms to 112 ms. The WDT/WKT is cleared by the CLRWDT instruction. If the Watchdog Reset is enabled (WDTE=1), the WDT generates the chip reset signal, otherwise, the WKT only generates overflow time out interrupt. The WDT/WKT works in both normal mode and STOP mode. If WDTE=0 and WKTIE=0 (Wakeup interrupt disable), the Watchdog RC-OSC stops for power saving.

If the WDTE=1 and WKTIE=0, WDT/WKT timer will be cleared and stopped to power saving in STOP mode. If the WDTE=1 and WKTIE=1, WDT/WKT timer keep counting in STOP mode. Refer to the following table and figure.

Mode	WDTE	WKTIE	Watchdog RC-OSC *
Normal Mode	0	0	Stop
	0	1	Run
	1	0	
	1	1	
STOP/IDLE Mode	0	0	Stop
	0	1	Run
	1	0	Stop
	1	1	Run

If the user program needs the MCU totally shut down for power conservation in STOP mode, the above setting of control bits should be followed.

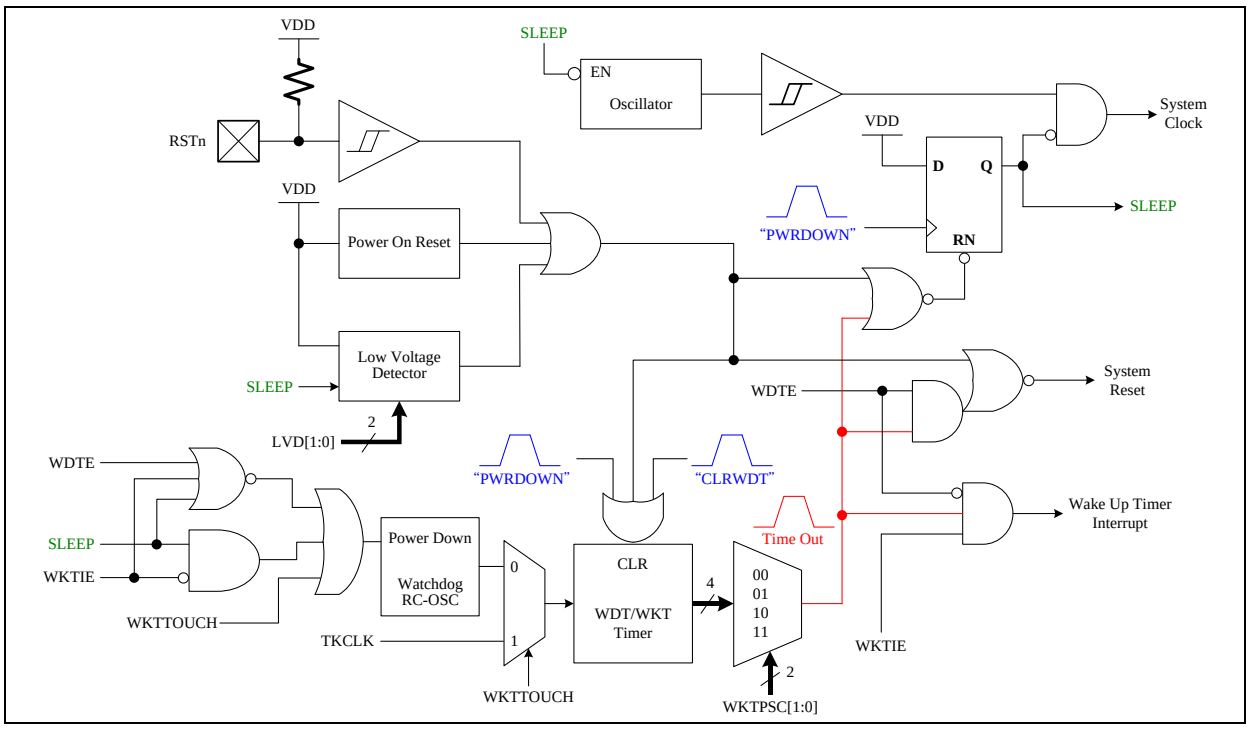


Figure 3.1 The internal Reset scheme

- This Watchdog is different from FIRC clock that is used to act as system clock.

3.2 8-bit Timer/Counter/Capture (Timer0) with Pre-scaler (PSC)

The Timer0 is an 8-bit wide register of F-Plane 01h. It can be read or written as any other registers of F-Plane. Timer0 increases itself periodically and rolls over based on the pre-scaled clock source, which can be instruction cycle, T0CKI (PA2) rising/falling, or TouchKey oscillating clock rising/falling. The Timer0 increasing rate is determined by "Timer0 Prescale" (TM0PSC) register in R-Plane. The Time0 will generate interrupt when it counts to overflow if Timer0 interrupt Enable (TM0IE) is set.

Timer0 can be stopped counting if the STOPTM0 bit is set. Timer0 can be configured as capture mode. If T0CAPTURE bit is set to "1", Timer0 will not count until the CAPT pin (i.e. PB1) is active.

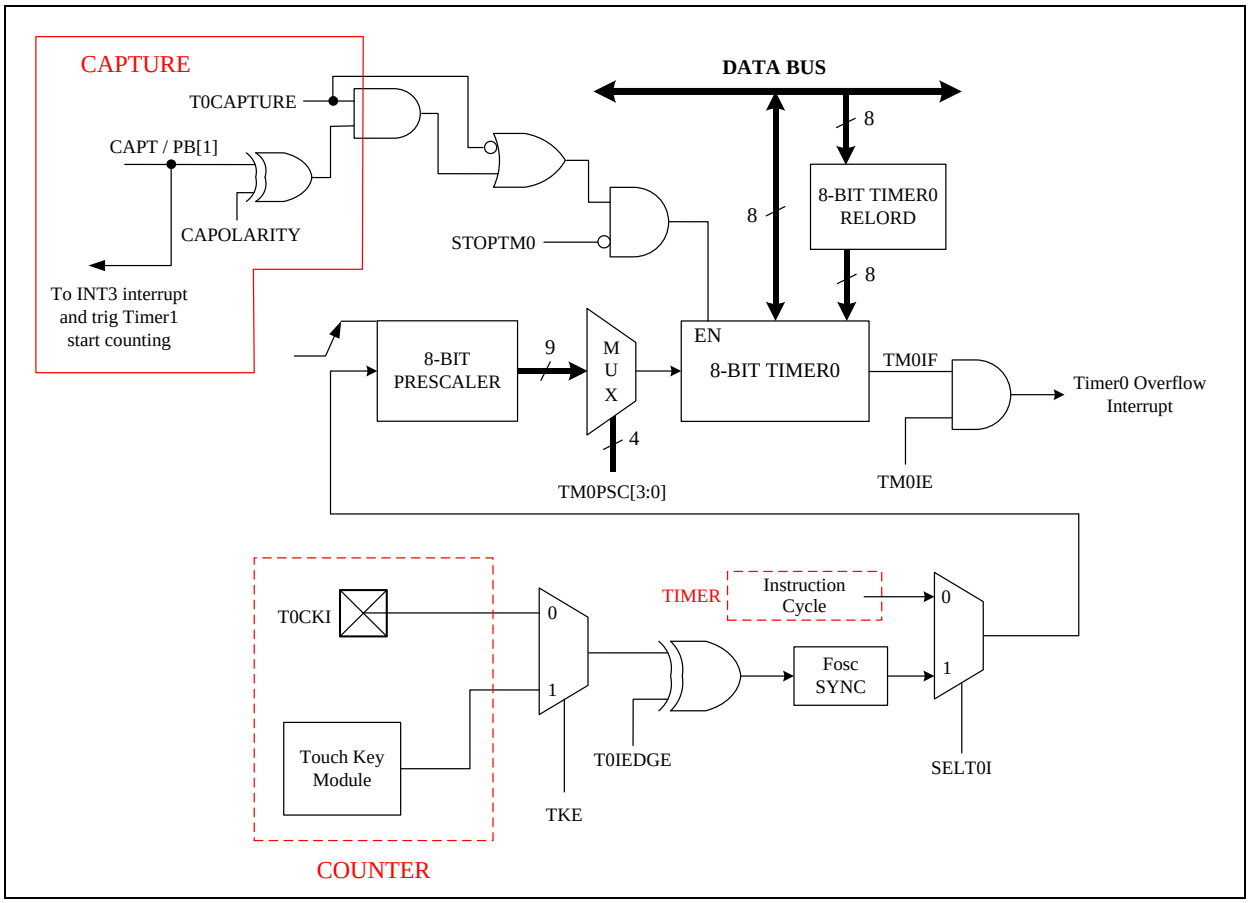


Figure 3.2.1 Timer0 Block Diagram

Figure 3.2.2 shows the Timer0 works in pure timer mode. When the Timer0 prescaler (TM0PSC) is written, the internal 8-bit prescaler will be cleared to 0 to make the counting period correct at the first Timer0 count. TM0CLK is the internal signal that causes the Timer0 to increase by 1 at the end of TM0CLK. TM0WR is also the internal signal that indicates the Timer0 is directly written by instruction; meanwhile, the internal 8-bit prescaler will be cleared. When Timer0 counts from FFh to TM0RELORD, TM0IF (Timer0 Interrupt Flag) will be set to 1 and generate interrupt if TM0IE (Timer0 Interrupt Enable) is set.

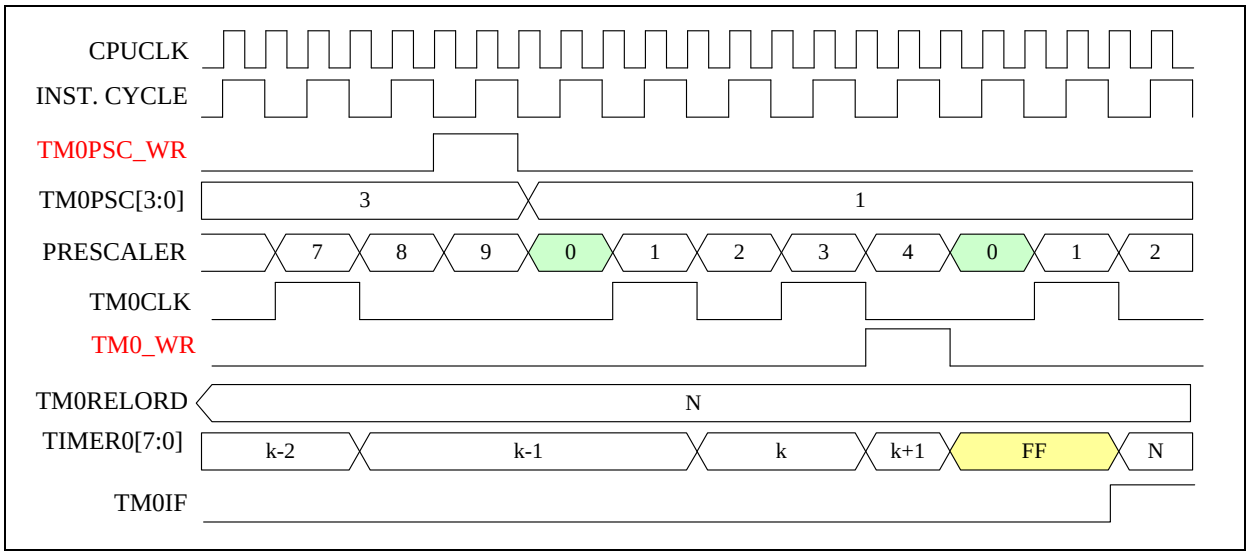


Figure 3.2.2 Timer0 works in timer mode

The following timing diagram describes the Timer0 works in counter mode. If SELT0I=1 then the Timer0 counter source clock is from T0CKI pin or Touch Key module that depends on TM0TOUCHKEY bit. As shown in Figure 3.2.3, T0CKI (or Touch Key clock) signal is synchronized by instruction cycle (i.e. 2 oscillation clocks), that means the high/low time durations of T0CKI must be longer than one instruction cycle time to guarantee each T0CKI's change will be detected correctly by the synchronizer.

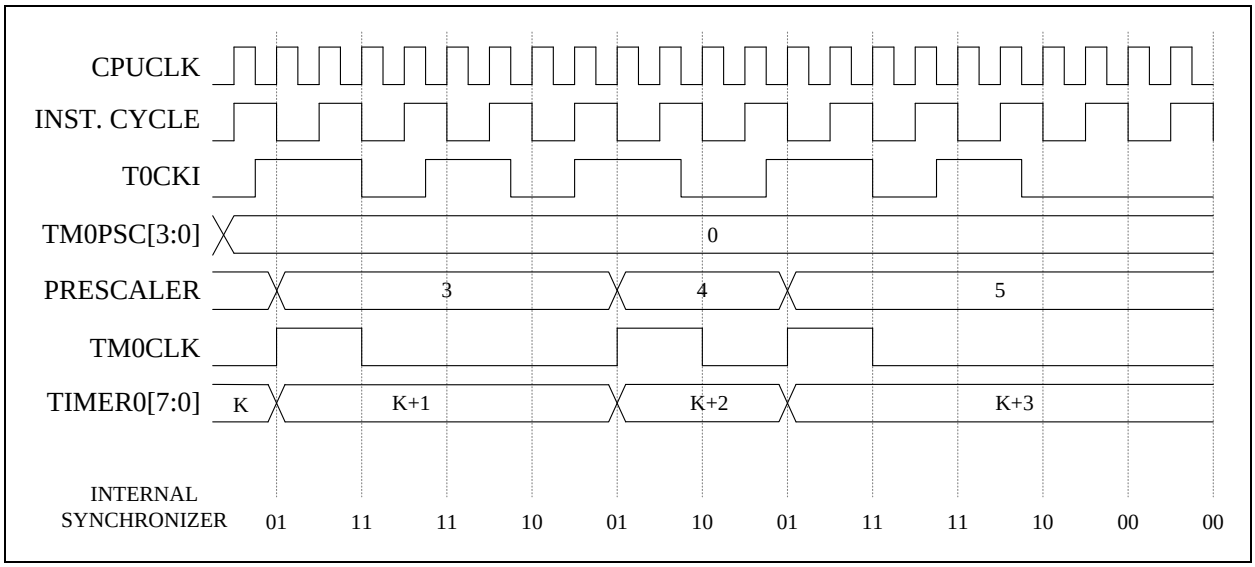


Figure 3.2.3 Timer0 works in external T0CKI input mode

Timer0 can be also used to measure the pulse with and period capture on CAPT pin. This function needs the Timer1 and INT6 external interrupt and software control step by step. **Sector 3.5** will discuss the details.

3.3 Timer1

Timer1 is a 16-bit counter with 16-bit auto-reload register. Figure 3.3.1 shows the Timer1 block diagram. Timer1 can only be accessed by reading F-Plane TM1H and TM1L. Writing TM1H and TM1L is actually writing to Timer1 reload registers. The clock sources of Timer1 are Fosc and Fosc/2, selected by TM1PSC. Setting the bit CLRTM1 will clear Timer1 and hold Timer1 on 0000h. Setting the bit TM1SET will hold Timer1 on ffffh. Setting the STOPTM1 bit will stop Timer1 counting.

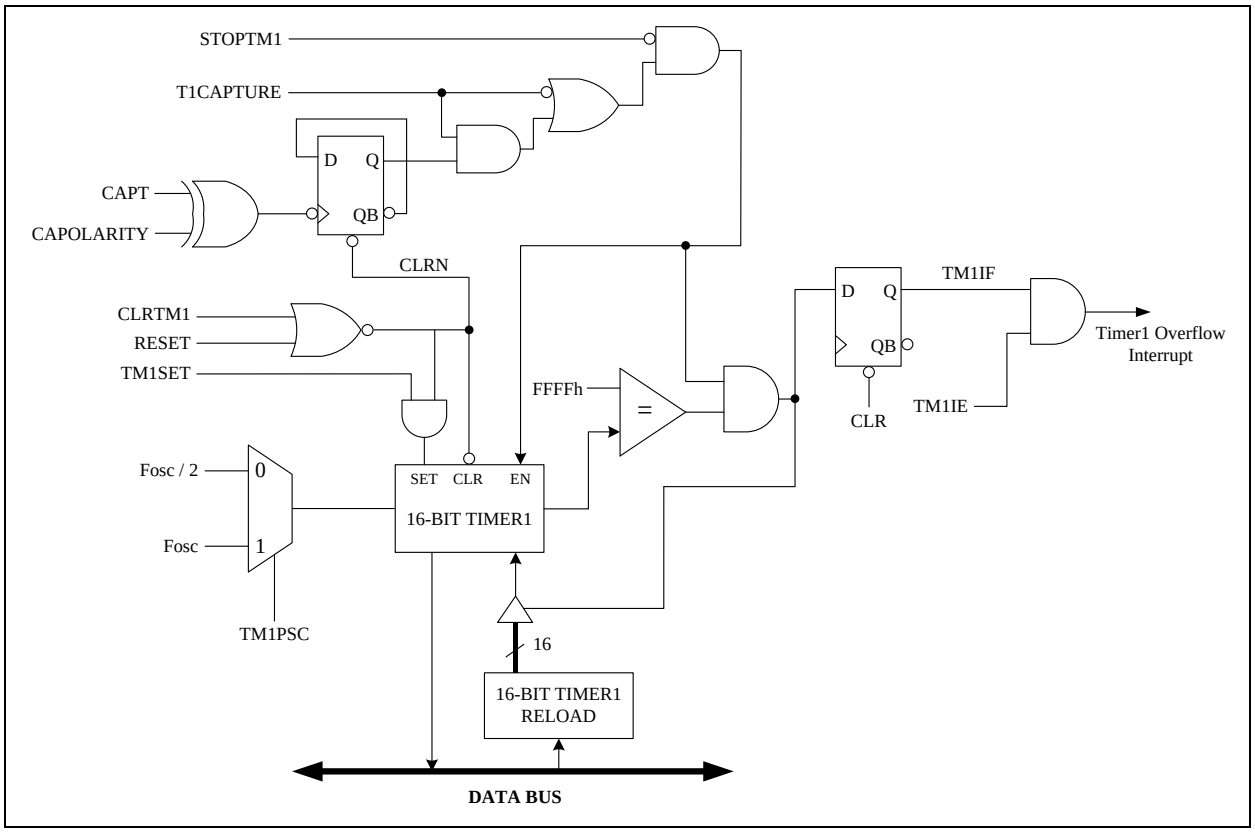


Figure 3.3.1 Timer1 Block Diagram

Note that writing to TM1H and TM1L is actually writing to Timer1 reload register, while reading TM1H and TM1L is actually reading the Timer1 counter itself. That is, Timer1 counter and Timer1 Reload register share two addresses (0ah, 0bh) of F-Plane.

Timer1 can also works with capture mode. When works in capture mode, Timer1 will start counting when the CLRTM1 bit is cleared and the first falling edge of CAPT pin (if CAPOLARITY=0) is coming. When the 2nd falling edge of CAPT pin is coming, Timer1 stops counting and hold the value. When the 3rd falling edge of CAPT pin is coming, the Timer1 continue counting. Figure 3.3.2 shows the detail timing diagram.

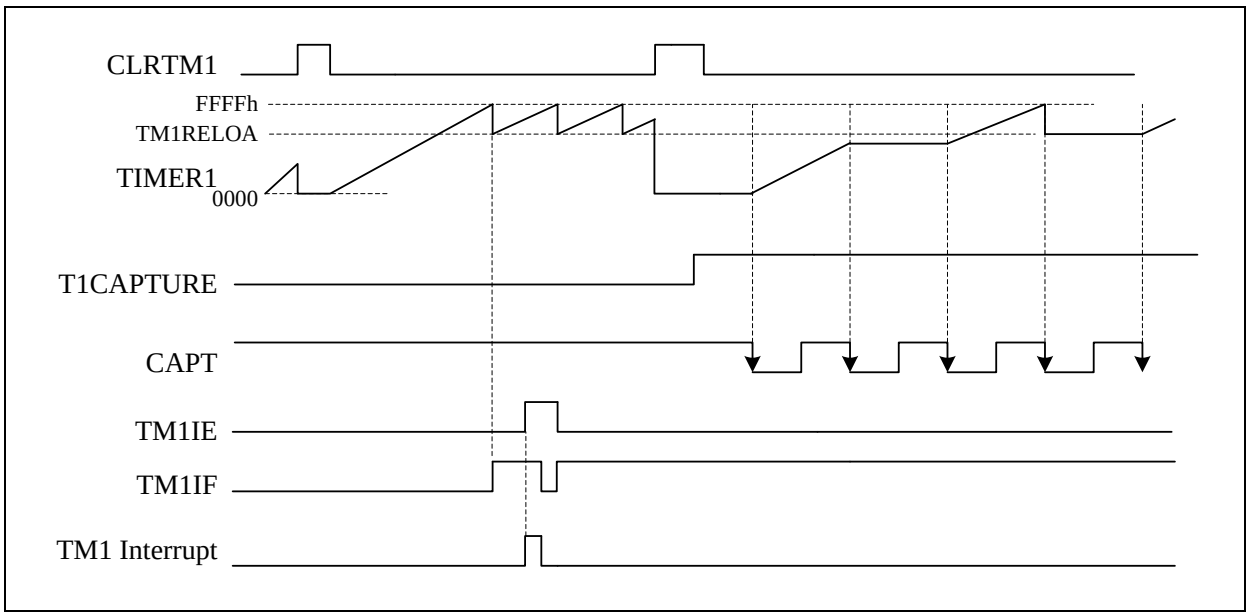


Figure 3.3.2 (CAPOLARITY=0, implies CAPT falling edge)

3.4 Timer2

Timer2 is a 15-bit counter and the clock sources are from either Fosc/128 or Slow Clock. It is used to generate time based interrupt and LCD clock. The Timer2 content cannot be read by instructions. It generates interrupt with the clock divided by 32768, 16384, 8192, and 128, depends on TIMER2DIV register bits. Figure 3.4.1 shows the block diagram of Timer2.

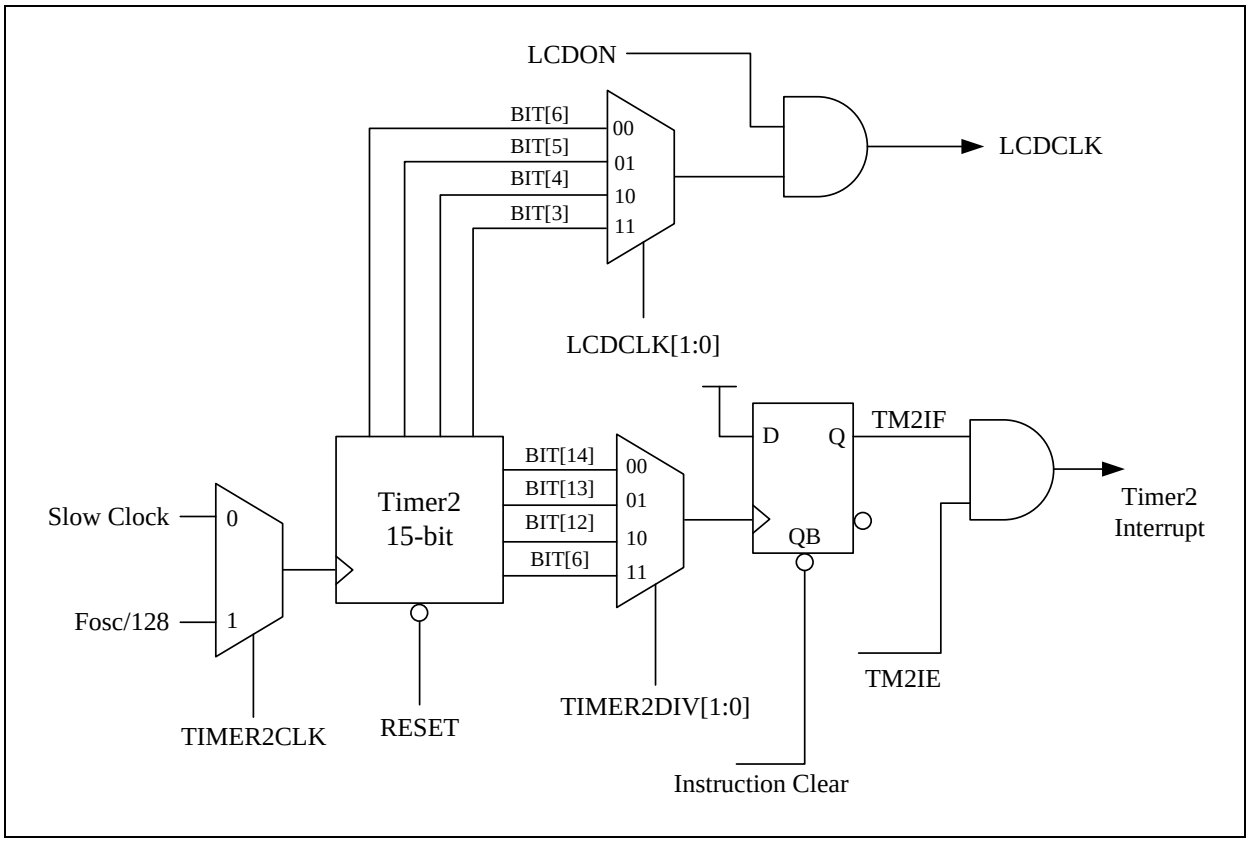


Figure 3.4.1 Timer2 Block Diagram

3.5 Timer0 and Timer1 used for Pulse Width and Period Capture

Timer0 and Timer1 can cooperate to measure the signal period and duty cycle time. The key is multifunction of PB1 (CAPT, XINT3). Suppose that:

- SELT0I=0, Timer0 prescaler increases per instruction cycle.
- T0CAPTURE=1, T1CAPTURE=1. Timer0 and Timer1 work in capture mode.
- TM0TOUCHKEY=0, TouchKey function is disabled.
- XINT3EDGE=0, PB1 pin (CAPT pin) interrupt every falling edge.
- CAPOLARITY=0, **Timer1** start/hold in turn when CAPT pin falling edge is coming. **Timer0** start counting when CAPT pin in logic '1' level and hold the Timer0 value when CAPT pin in logic '0' level.
- Timer1 is used to measure the signal period, Timer0 is used to measure the CAPT pin in logic '1' time (i.e. the duty cycle of the signal).

Figure 3.5.1 shows how to use Timer0 and Timer1 to measure the CAPT pin signal's period and duty cycle.

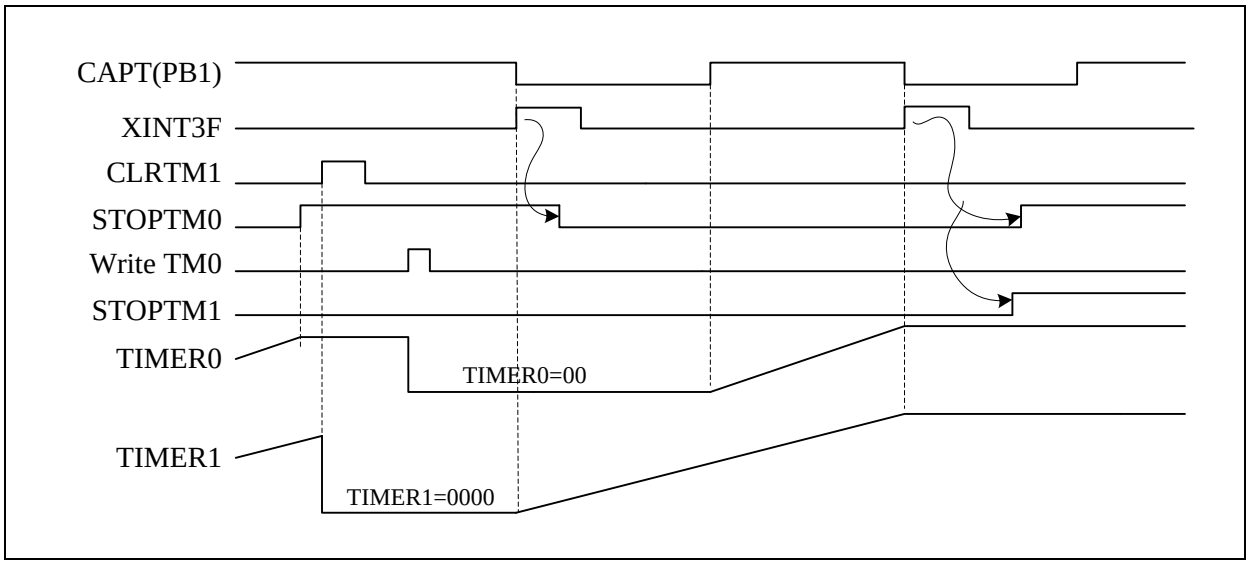


Figure 3.5.1 Timer0 and Timer1 used to measure the signal on CAPT pin.

Follow the steps below to start measuring the CAPT pin's period and duty cycle.

1. Stop Timer0 by firmware (Timer0 will be stopped and hold)
2. Clear Timer1 by firmware
3. Clear Timer0 by directly writing 00h to Timer0 (Timer0 is still hold)
4. Once PB1 falling edge is coming, the Timer1 starts counting; meanwhile the XINT3 interrupt is generated and cleared the STOPTM0 by firmware. Now the Timer0 is ready to count when PB1 goes high)
5. PB1 rising edge is coming, Timer0 starts counting until the PB1 returns to 0 and holds the counting value. Timer1 also stops counting and holds the value.
6. XINT3 interrupt is generated again, firmware stops Timer1 and Timer0 to read the period and duty cycle.

It is not necessary to use both Timer0 and Timer1. If only the duty cycle (CAPT high time) needs to be measured, there is no need to use Timer1 to measure the period. In such case, user can set the T0CAPTURE=1 and T1CAPTURE=0. As shown in Figure 3.5.2, Timer0 is counting up only when PB1 (CAPT pin) is '1'. Note that the internal prescaler will be kept to next Timer0 count, so it will not lose the counting accuracy.

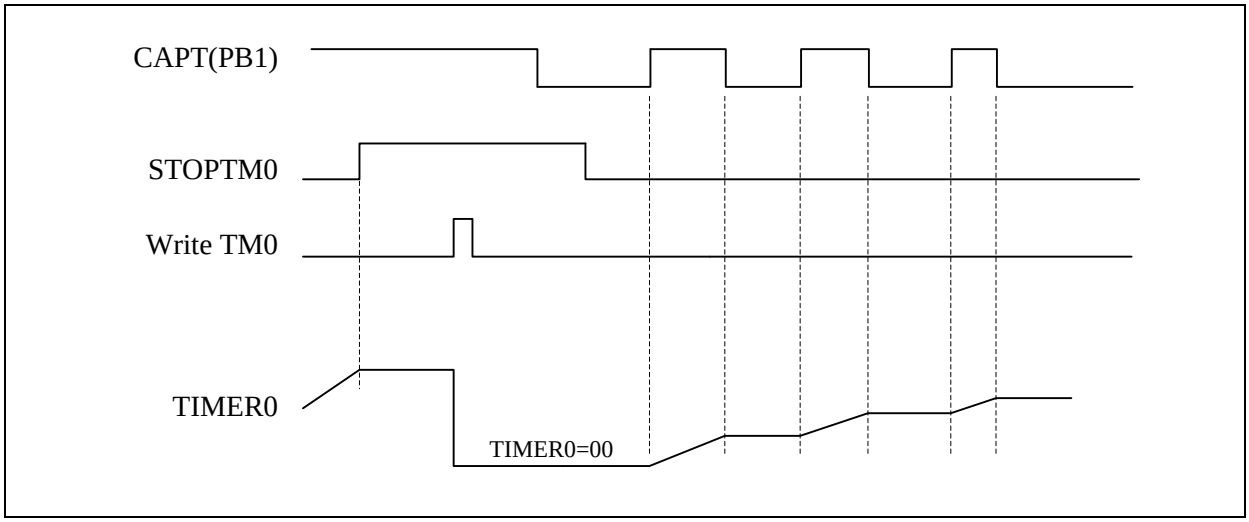


Figure 3.5.2 Timer0 is used to measure the high (or low) time on CAPT pin

3.6 PWM0

The chip has a built-in 8-bit PWM generator. The source clock comes from Fosc divided by 1, 2, 4, and 8. The PWM0 duty cycle can be changed with writing to PWM0DUTY, writing to PWM0DUTY will not change the current PWM duty until the current PWM period completes. When current PWM period is finished, the new value of PWM0DUTY will be updated to the PWM0BUF.

The PWM0 will be output to PA6 if PWM0E is set to 1 and PWM0N=0. The complement of PWM0, PWM0N, will be output to PA6 if PWM0E is set to 1 and PWN0N=1. Also, the PWM period complete will generate an interrupt when PWM0IE is set to 1. Setting the CLRPWM0 bit will clear the PWM0 counter and load the PWM0DUTY to PWM0BUF, CLRPWM0 bit must be cleared so that the PWM0 counter can count.

Note that the default value of CLRPWM0 bit is '1'.

Figure 3.6.1 shows the block diagram of PWM0.

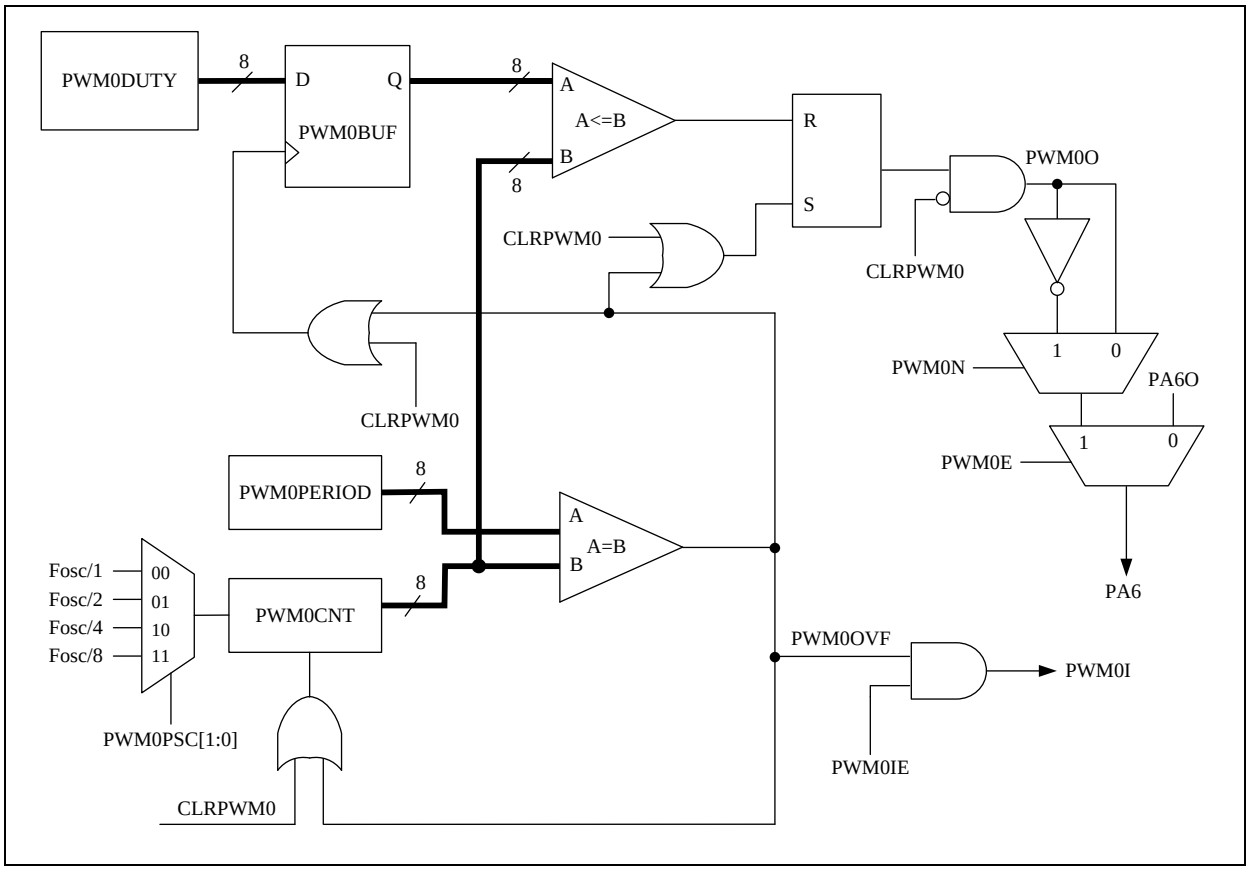


Figure 3.6.1 PWM0 Block Diagram

Figure 3.6.2 and Figure 3.6.3 shows the PWM0 waveforms. When CLRPWM0 bit is set to '1', the PWM0 output is cleared to '0' no matter what its current status is. Once the CLRPWM0 bit is cleared to '0', the PWM0 output is set to '1' to begin a new PWM cycle. PWM0 output will be '0' when PWM0CNT is greater than or equals to PWM0BUF. PWM0CNT keeps counting up when equals to PWM0PERIOD, the PWM0 output is set to '1' again.

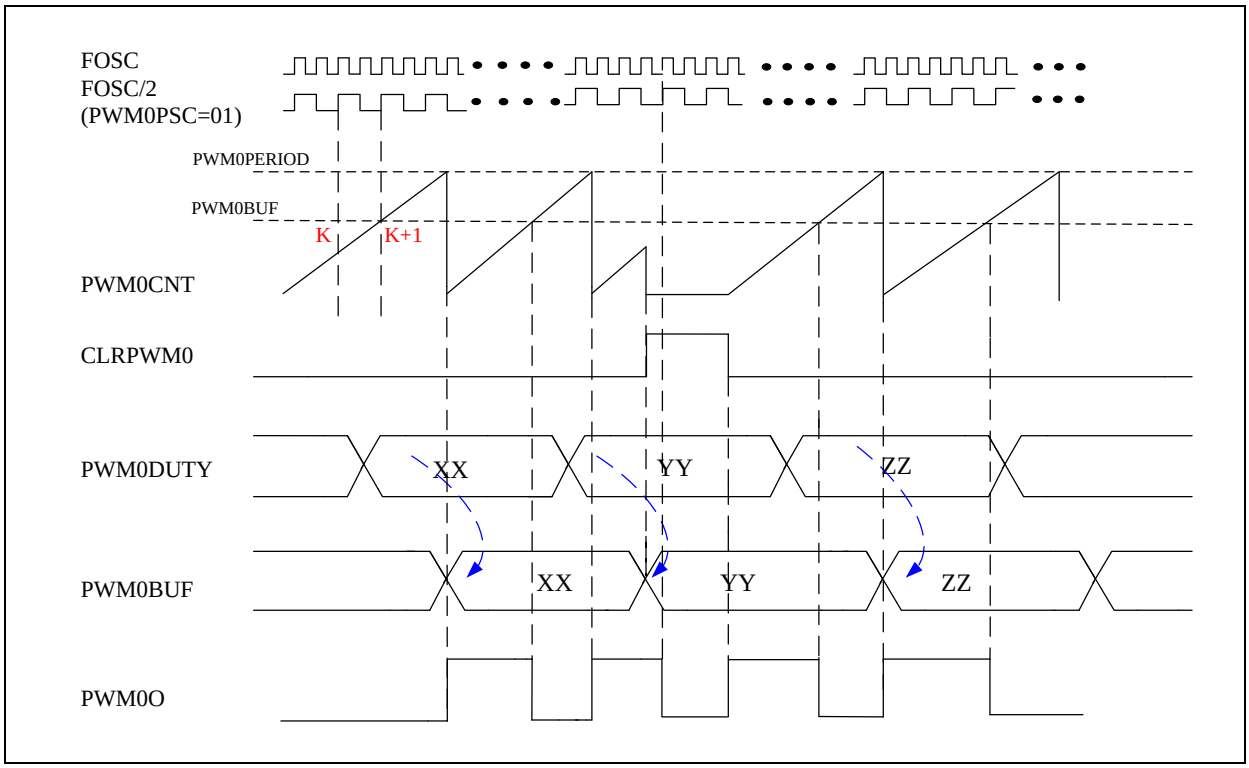


Figure 3.6.2 PWM0 Timing (CLRPWM0 before PWM0CNT reaches PWM0BUF)

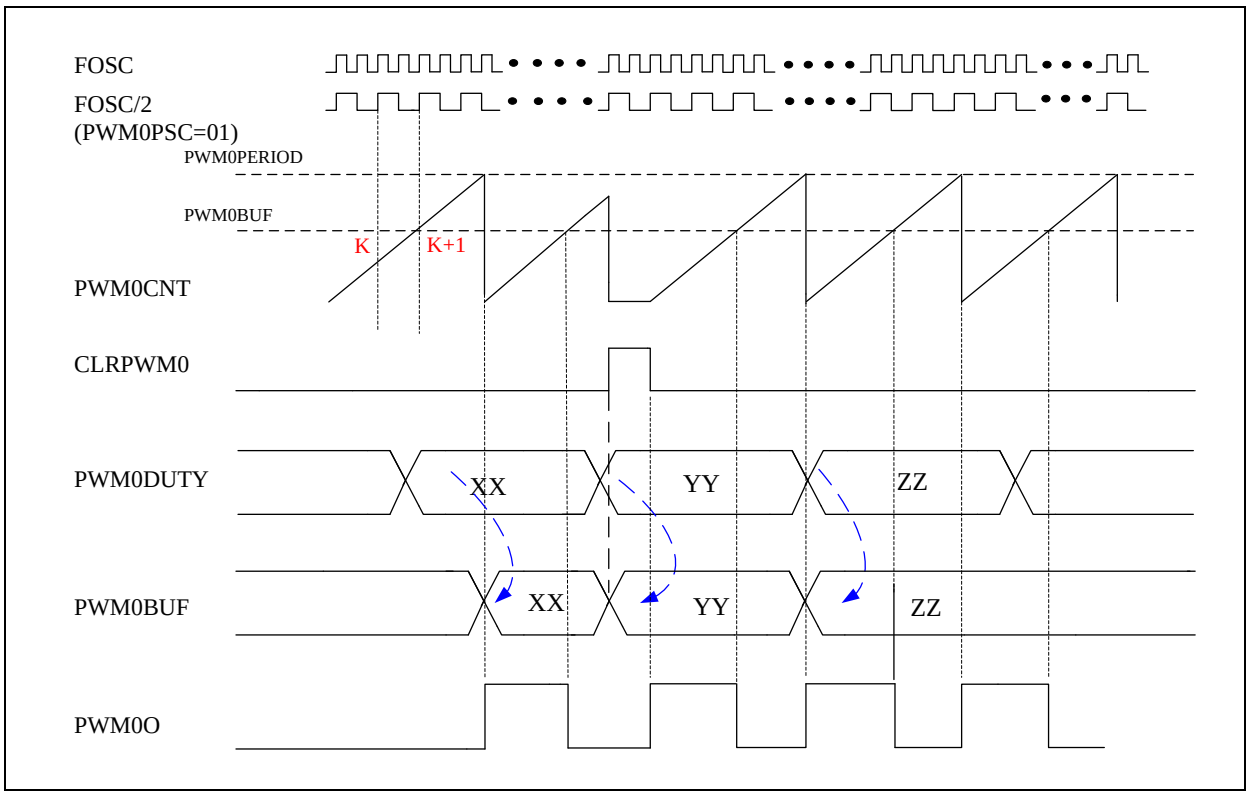


Figure 3.6.3 PWM0 Timing (CLRPWM0 after PWM0CNT over PWM0BUF)

3.7 PWM1

PWM1 is a simple fixed frequency and duty cycle variable PWM generator. The PWM frequency is fixed, the period is system clock counts from 0 to 255. The duty can be set via PWM1DUTY register. The output of PWM1 shares the pin PD0 that can be selected by PWM1E control bit. Figure 3.7.1 is the block diagram of PWM1. Figure 3.7.2 shows the related timing of PWM1. The PWM frequency is:

- PWM1 Frequency=Fosc / 256
- PWM1 Duty Cycle= (PWM1DUTY/256) * 100%

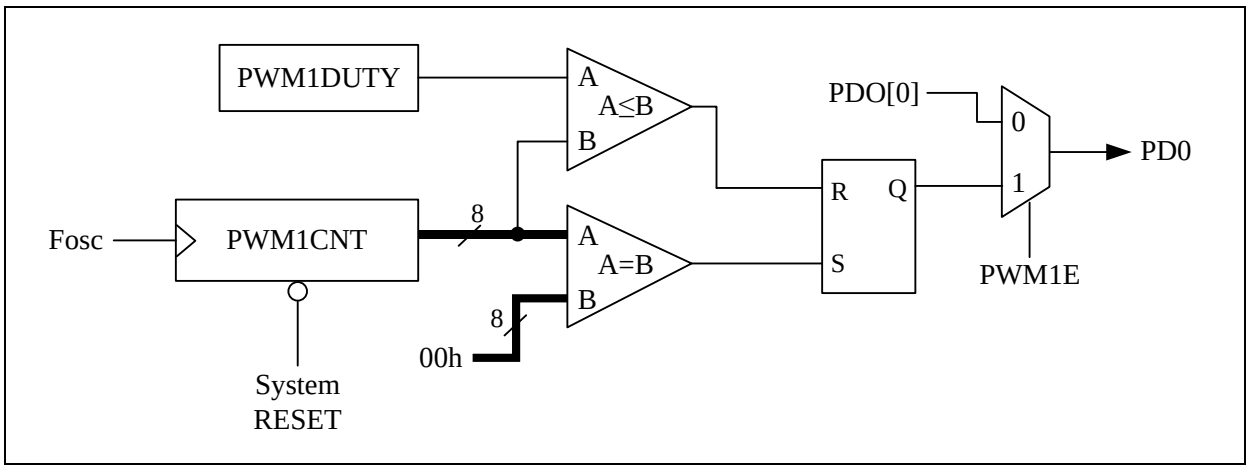


Figure 3.7.1 PWM1 Block Diagram

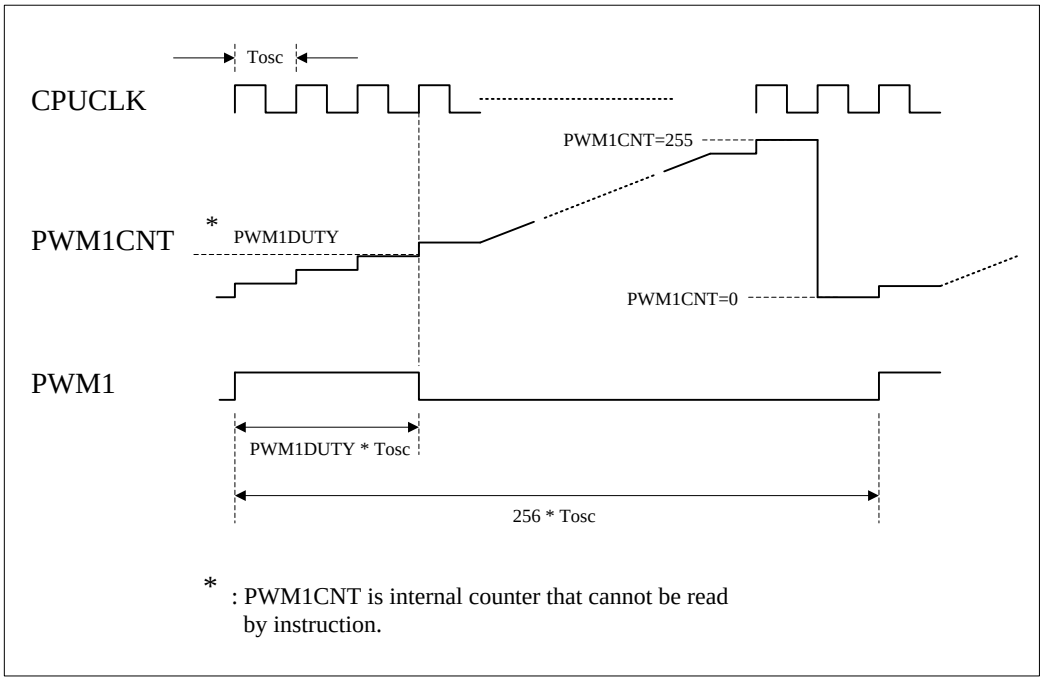


Figure 3.7.2 PWM1 Timing

3.8 BUZZER Output

The Buzzer driver consists of 6-bit counter and a clock divider. It generates 50% duty square waveform with wide frequency range. To use the Buzzer function, user needs to set both the Buzzer enable control bit (BUZ_EN) and the Buzzer output pin enable control bit (BUZ_OUT).

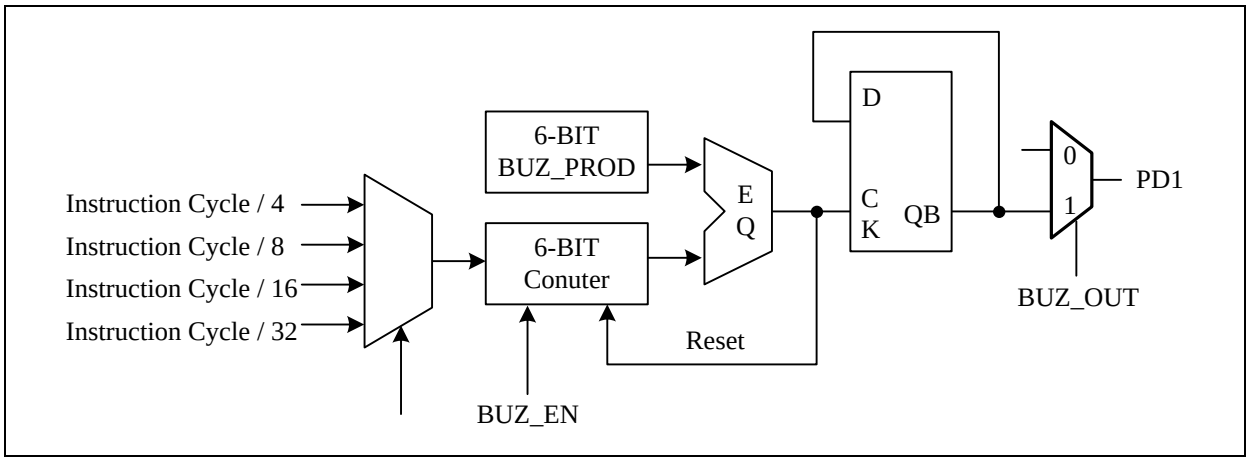
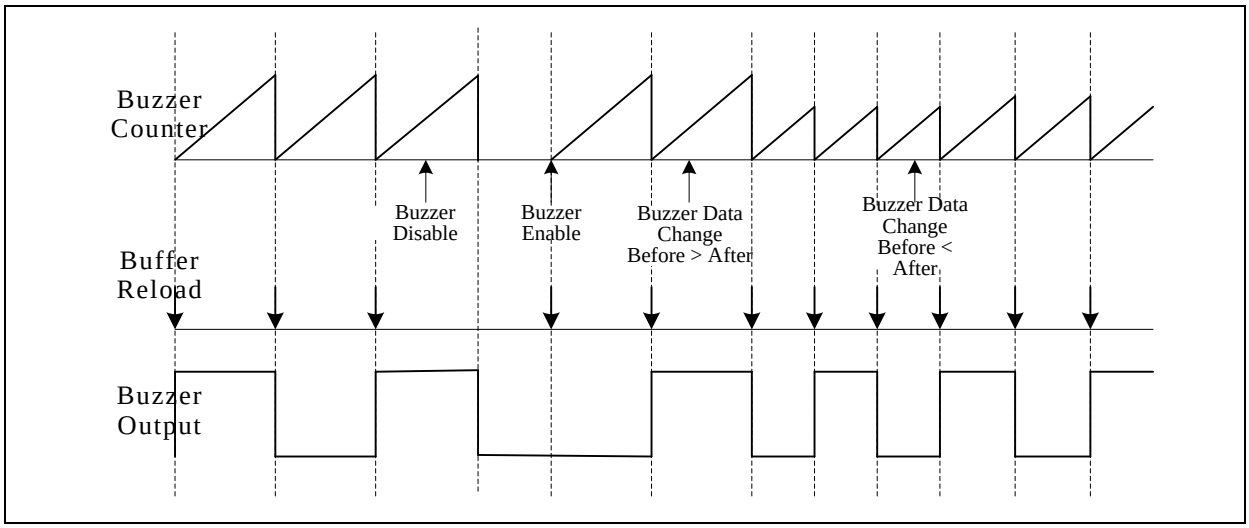


Figure 3.8.1 BUZZER Block Diagram



BUZ_PROD[5:0] determines output frequency. Frequency calculation is as follows.

F_{BZ}= (f_{OSC}/2) / (Instruction Cycle Divider) / (BUZ_PROD +1)

Output frequency calculation

CPU Clock (fosc) =8192 KHz

Instruction Cycle=fosc/2=8192 KHz/2 = 4096 KHz

Prescaler Ratio (BUZ_PSC) =2'b11 (Instruction Cycle Divider=32)

Period Data (BUZ_PROD) =9

F_{BZ}= (8192 KHz/2) /32/ (9+1) =12.8 (KHz)

3.9 Resistance to Frequency Converter (RFC)

Resistance to frequency converter (RFC) contains RC oscillator and RFC counter. RFC can compare different sensors with the reference resistor separately. This figure shows the block diagram of RFC.

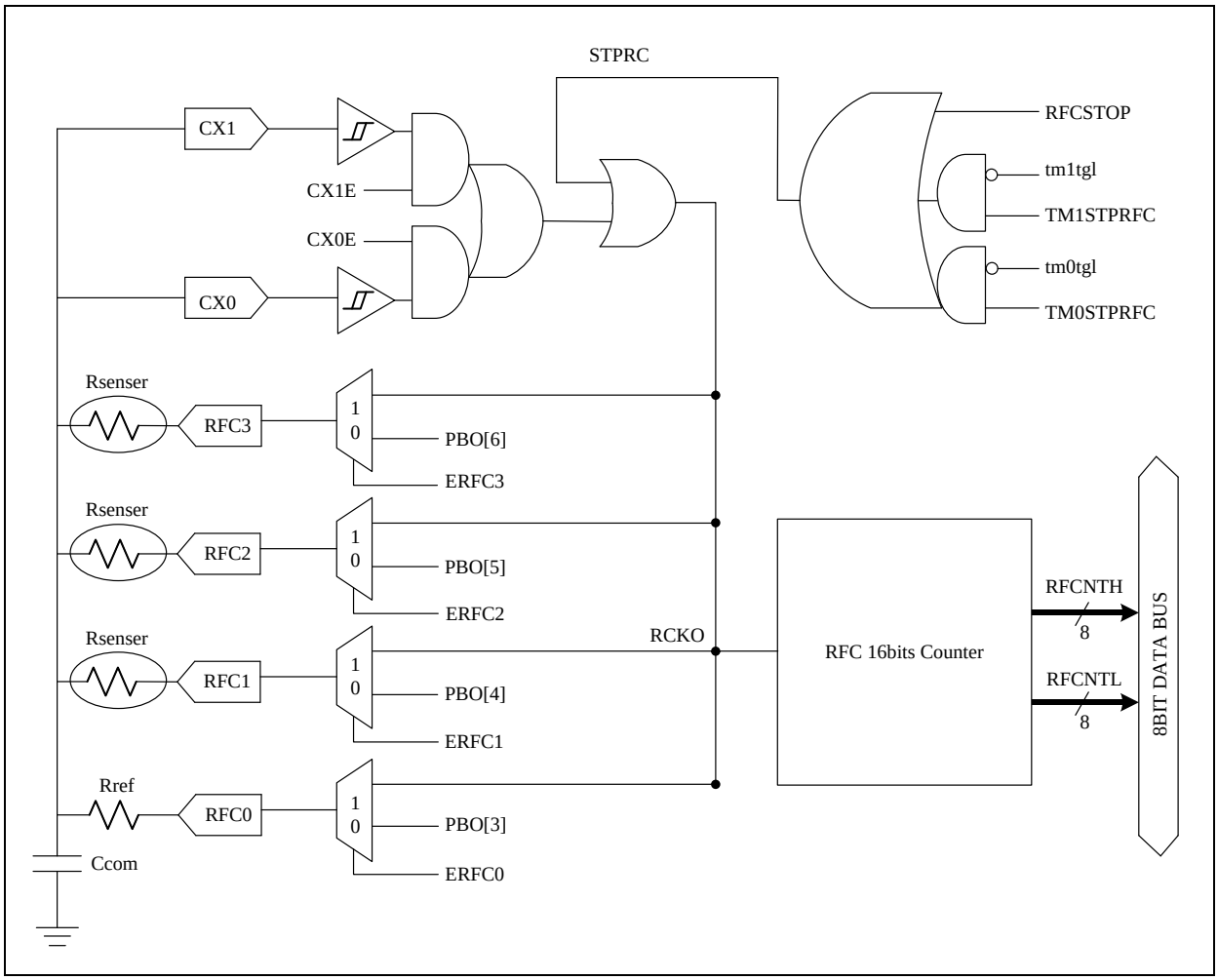


Figure 3.9.1 RFC Block Diagram

The RFC contains 6 pins

1. CX0~1: the oscillation Schmitt trigger input
2. RFC0~3: the resistor output pin

There are three kinds of RFC control mode in TM57ML40

1. normal mode: In this mode, RFC counter is controlled by RFCSTOP
2. TM0 mode: In this mode, TM0STPRFC is set to “1”, RFC counter is controlled by tm0tgl
3. TM1 mode: In this mode, TM1STPRFC is set to “1”, RFC counter is controlled by tm1tgl

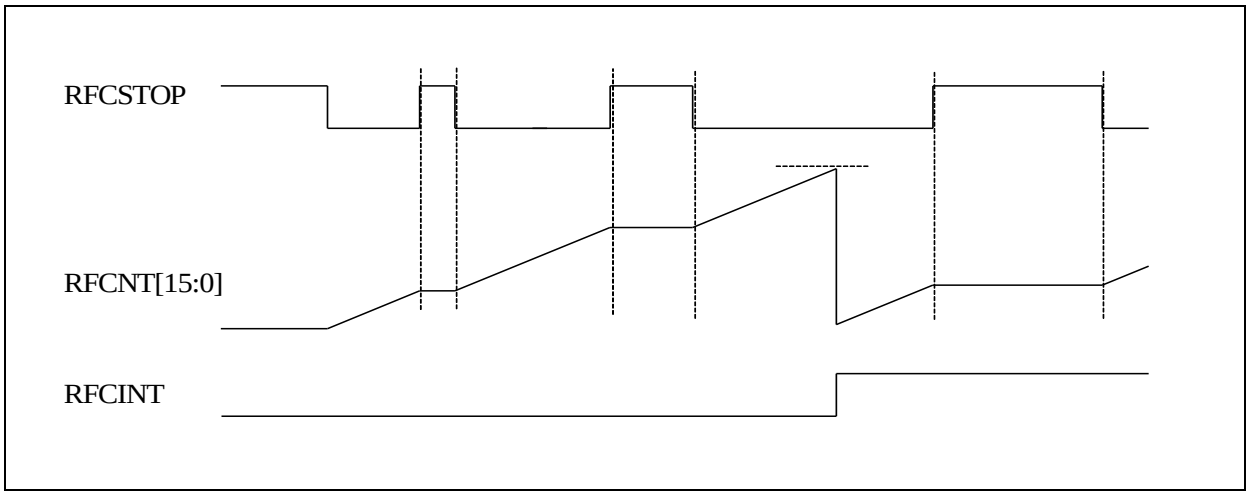


Figure 3.9.2 RFC in normal mode

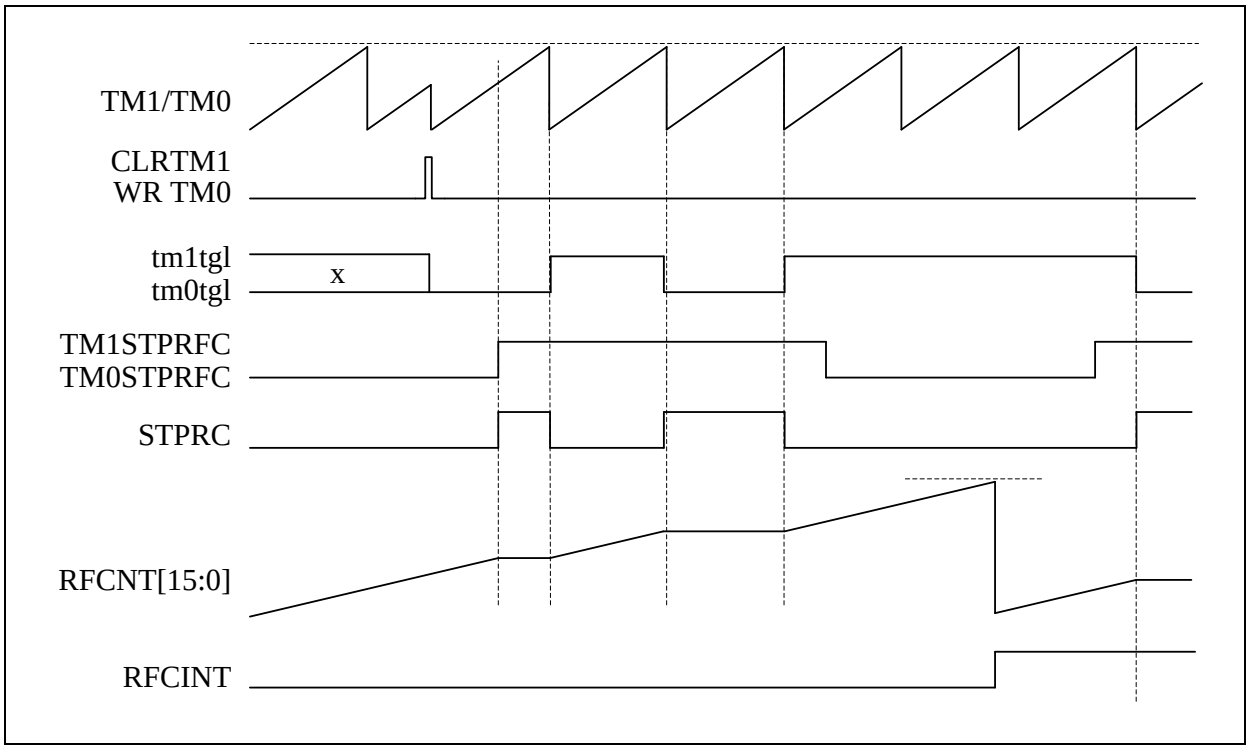


Figure 3.9.3 RFC in TM0/TM1 mode

tm1tgl/tm0tgl is set to “0” after CLRTM1/WR TM0, and tm1tgl/tm0tgl will be inverted when TM1/TM0 overflow happens. When tm1tgl/tm0tgl=0, and TM1STPRFC/TM0STPRFC=1, the STPRC is set to “1” and RFCNT stops counting. When tm1tgl/tm0tgl=1, the STPRC is set to “0” and RFCNT keeps counting.

3.10 LCD

The chip has the LCD (Liquid Crystal Display) driver. It is capable of driving the LCD panel with max 256 dots with 8 Commons and 32 Segments. Figure 3.10.1 shows the block diagram of LCD. The V_{LCD} is divided from V_{DD} by a set of resistors with the brightness selection bits to vary the V_{LCD} from (6/7.5) Vdd to Vdd. The TM57ML40 is capable of driving 1/3 LCD Bias and 1/2 LCD Bias, and generate the individual COM and SEG waveform.

The LCDCLK is generated from System clock or Slow Clock depends on TIMER2CLK bit because the LCDCLK comes from Timer2.

If the duty is set to static, only COM0 is the active COM line, while the even numbers (i.e. SEG0, SEG2, SEG4, …, SEG22) of SEG lines are active.

The address of the LCDRAM is ranges from 40h to 5fh, and the SEG lines can be used up to 32. See the Table 3.10.1 for detail LCD RAM map.

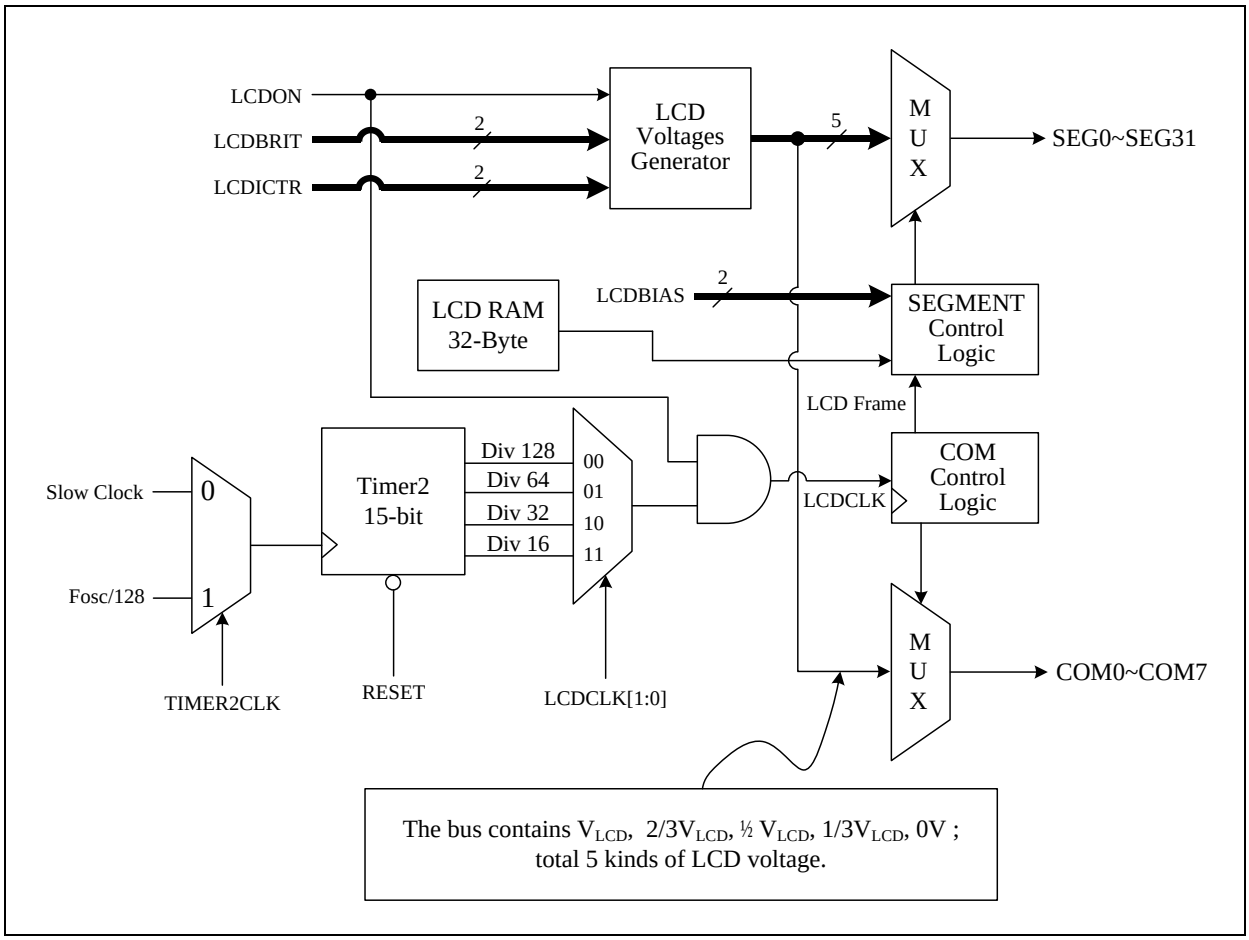
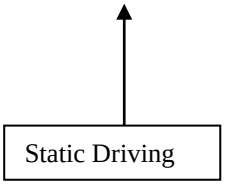


Figure 3.10.1 Block diagram of LCD driver.

8 COM		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
R-Plane	40	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0
	41	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1
	42	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2
	43	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3
	44	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4
	45	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5
	46	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6
	47	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7
	48	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8
	49	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9
	4a	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10
	4b	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11
	4c	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
	4d	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13
	4e	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14
	4f	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15
	50	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
	51	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17
	52	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18
	53	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19
	54	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
	55	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21
	56	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22
	57	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23
	58	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
	59	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25
	5a	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26
	5b	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27
	5c	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
	5d	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29
	5e	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30
	5f	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31

Table 3.10.1 LCD/LED RAM map



V_{LCD} and the Voltage Divider, adjusting the brightness

Figure 3.10.2 shows the internal voltage divider composed by resistors. LCDON controls the current flows from V_{DD} to ground. If LCDON=0, the PMOS will turn off the path so that all LCD voltages will be 0V. If LCDON=1, the resistor divider will work to generate the 5 voltages to provide the LCD control module for generating the desired waveforms. LCDBRIT control bits will open/short the switches to determine the V_{LCD}, Table 3.10.2 shows the LCDBRIT versus corresponding V_{LCD}. The voltage divider circuit will consume current from 10 μA to 20 μA because the DC path is always on when LCDON=1.

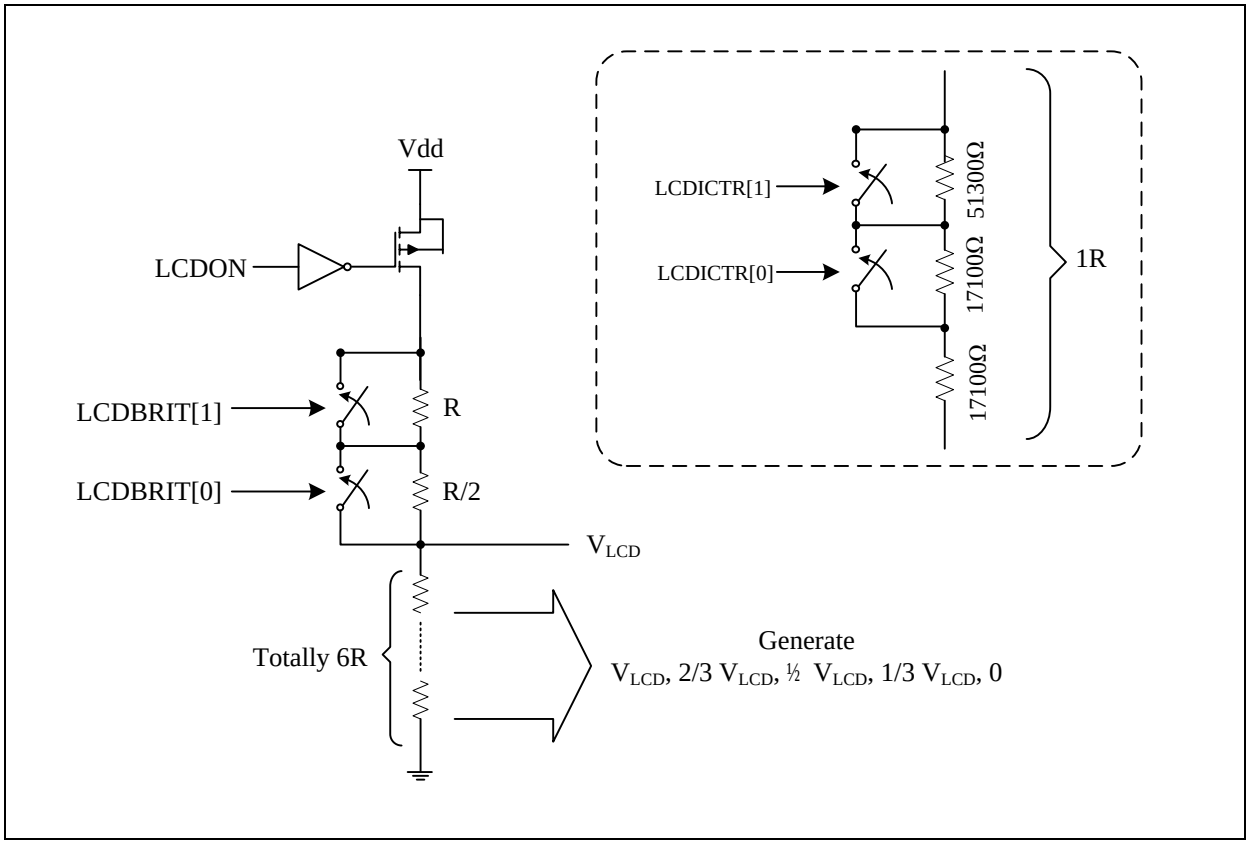


Figure 3.10.2 The LCD Voltage Divider

The higher V_{LCD} is, the higher V_{RMS} (Root Mean Square Voltage) is applied on the LCD segment, which means the higher brightness. Use the proper setting of LCDBRIT to fit the LCD panel specification.

LCDBRIT	V_{LCD}
00	(6/7.5) * Vdd
01	(6/7) * Vdd
10	(6/6.5) * Vdd
11	Vdd

Table 3.10.2 The VLCD corresponding to LCDBRIT

The Waveforms of COM and SEG for different BIAS/DUTY Static Drive:

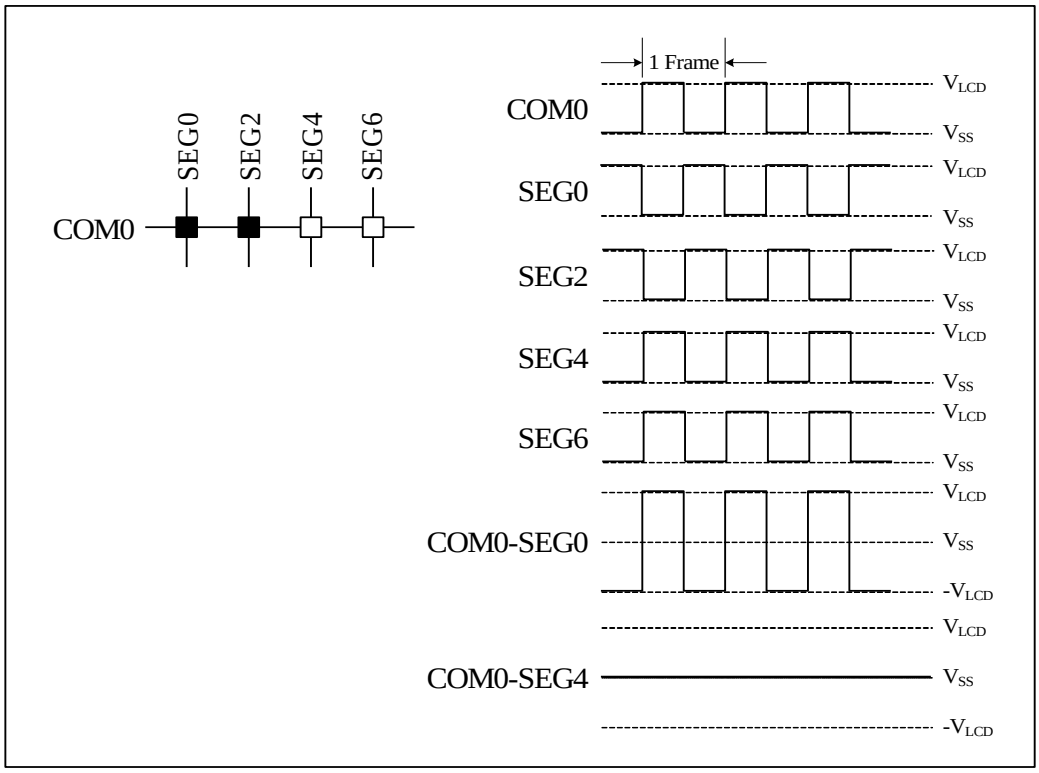


Figure 3.10.3 Static drive

1/4 Duty, 1/2 Bias Drive:

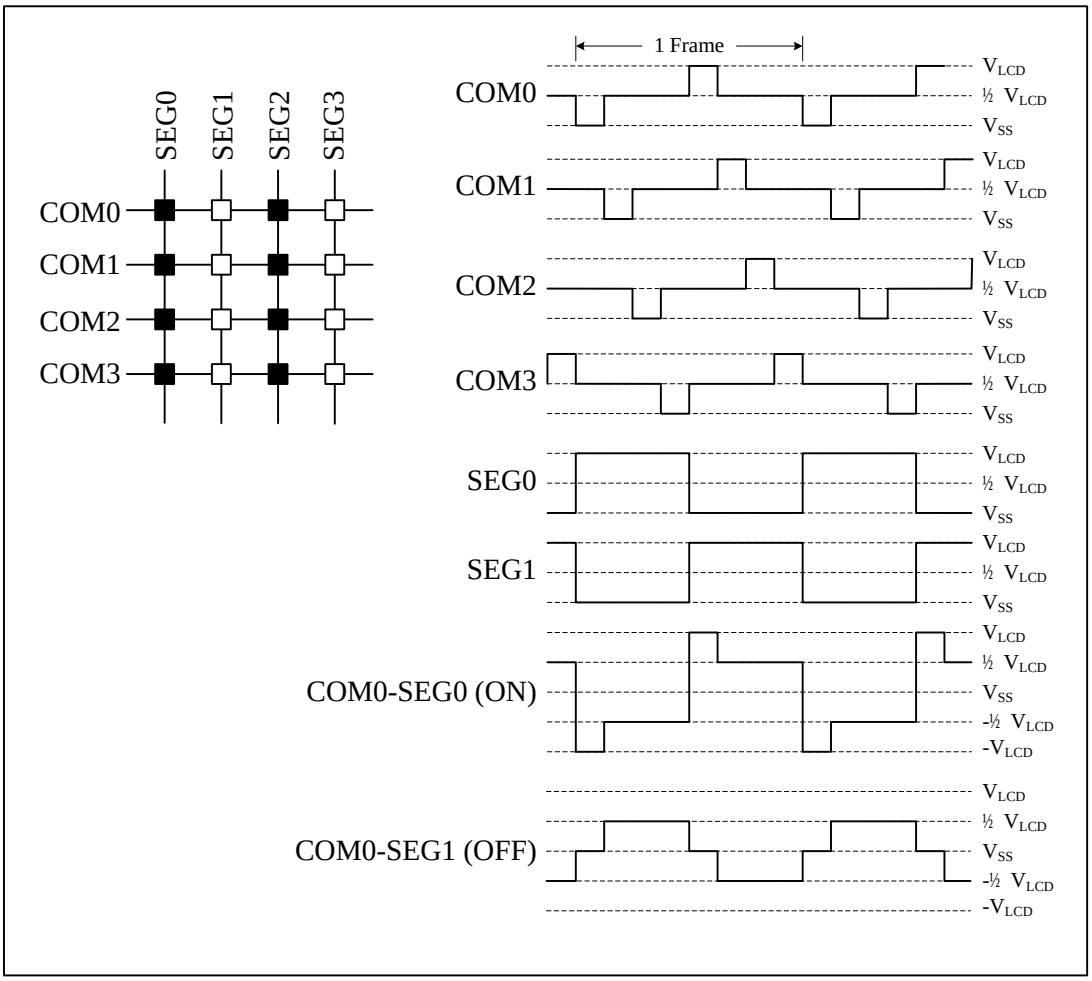


Figure 3.10.4 Configured as 1/4 Duty, 1/2 Bias

1/4 Duty, 1/3 Bias Drive:

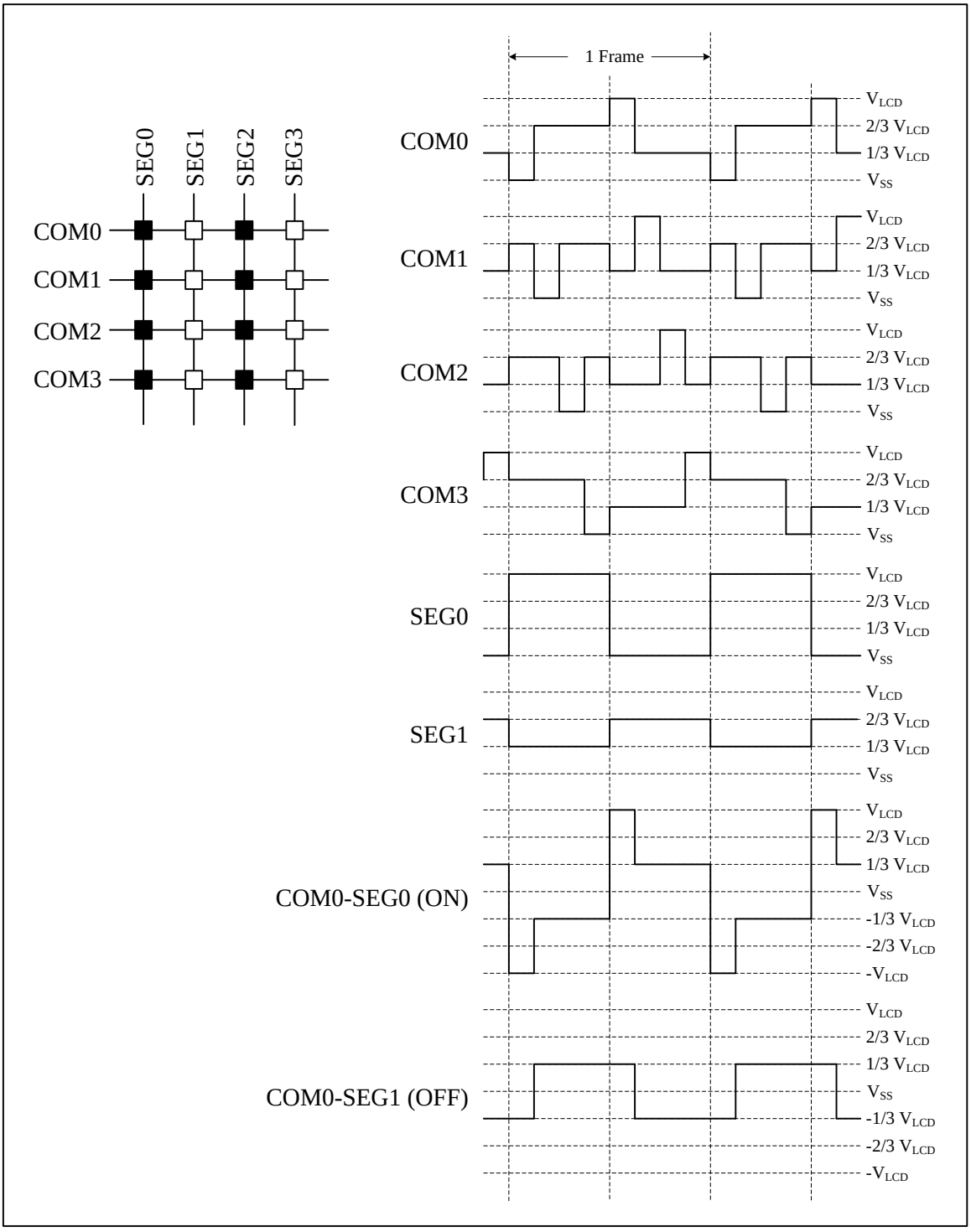


Figure 3.10.5 Configured as 1/4 Duty, 1/3 Bias

3.11 LED DRIVER OUTPUT

If the LED mode option (SELLED) is selected in R-Plane 1bh.1, TM57ML40 will switch the LCD driver to the LED driver. TM57ML40 provides 32 segment pins (SEG) and 8 common pins (COM) to drive a LED module with 256 pixels. For LED application, the COM pin can be selected as active low LED display or active high LED display by LEDTYPE. There are options for static, 1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty, 1/6 duty, 1/7 duty or 1/8 duty lighting systems. In the LED mode, the segment output pins' (SEG) waveforms are low active type.

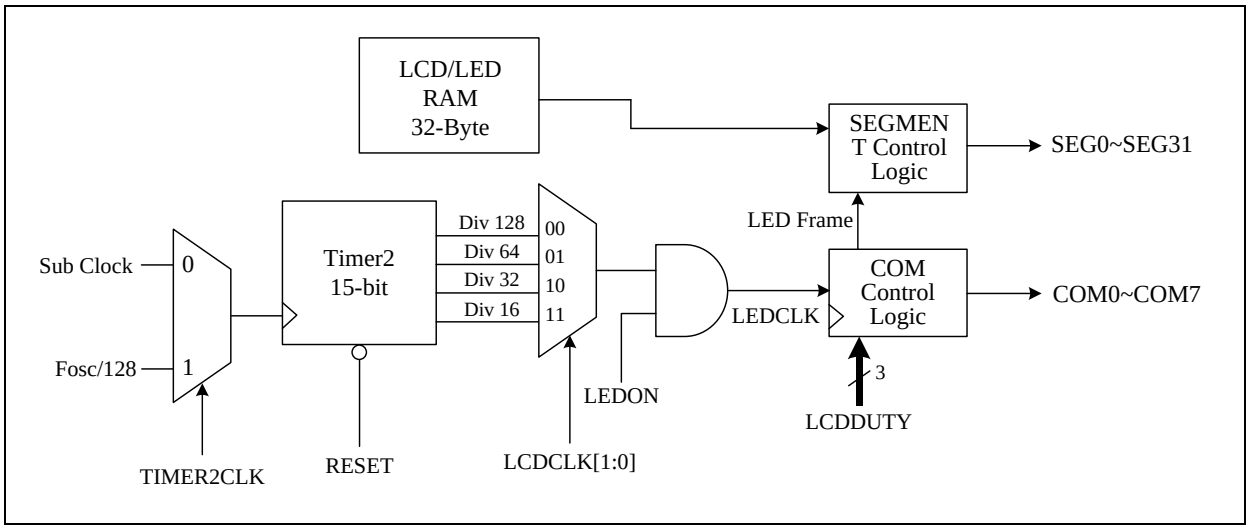
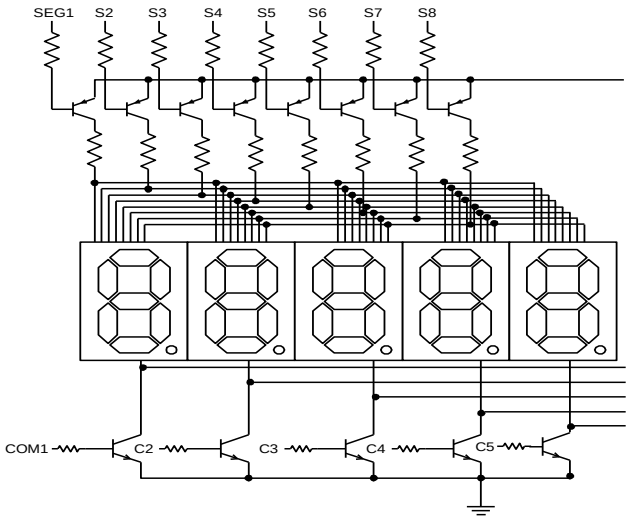


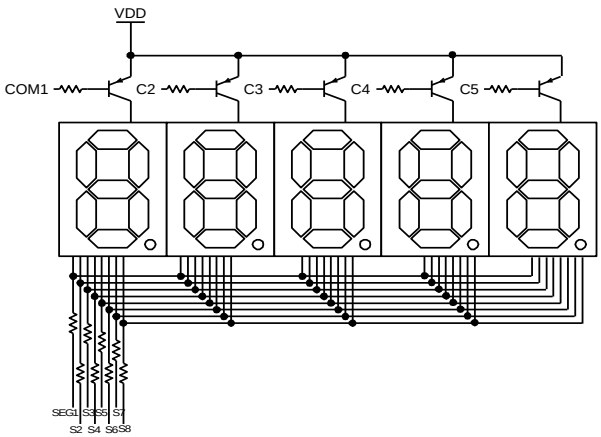
Figure 3.11.1 Block diagram of LED driver.

The following schematics illustrate the differences between high active mode and low active mode:

(1) **High Active Mode**



(2) **Low Active Mode**



Note: Please limit the total sink current less than 40 mA for each COM pin at Low Active Mode.

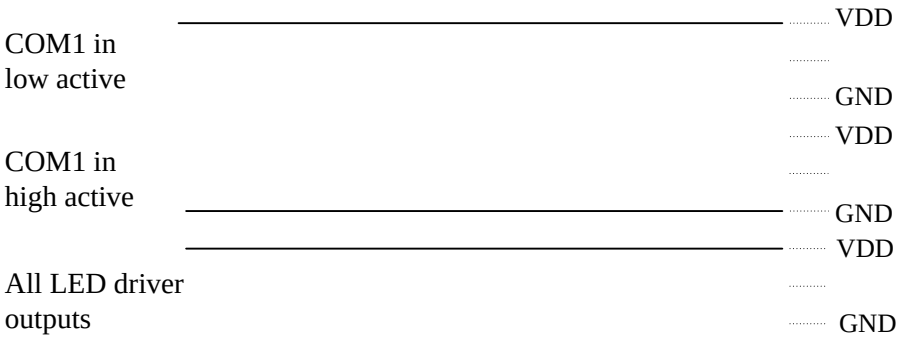
LED Lighting System and Maximum Number of Driving LED Segments

LED Lighting System	Maximum Number of Driving LED Segments
Static	32
1/2 duty	64
1/3 duty	96
1/4 duty	128
1/5 duty	160
1/6 duty	192
1/7 duty	224
1/8 duty	256

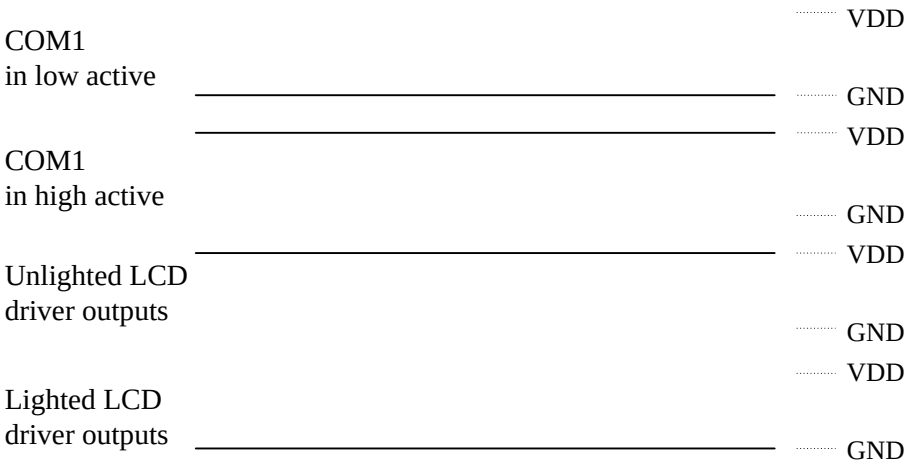
The examples of waveform on the COM output and LED driver output for LED lighting system are shown below.

STATIC LIGHTING SYSTEM FOR LED DRIVER

(i) Display Turned Off

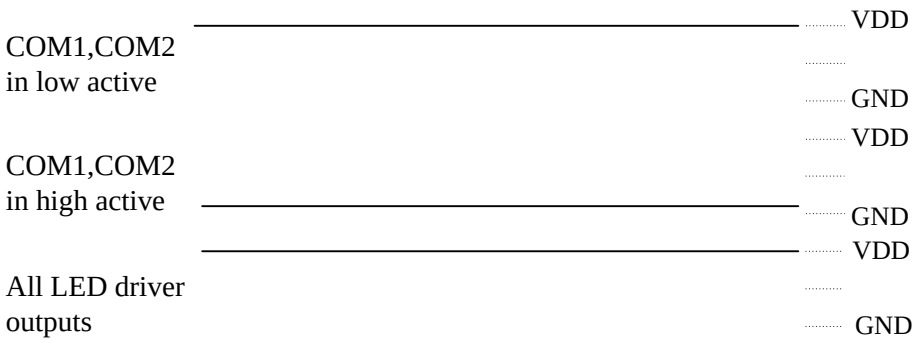


(ii) Normal operation mode

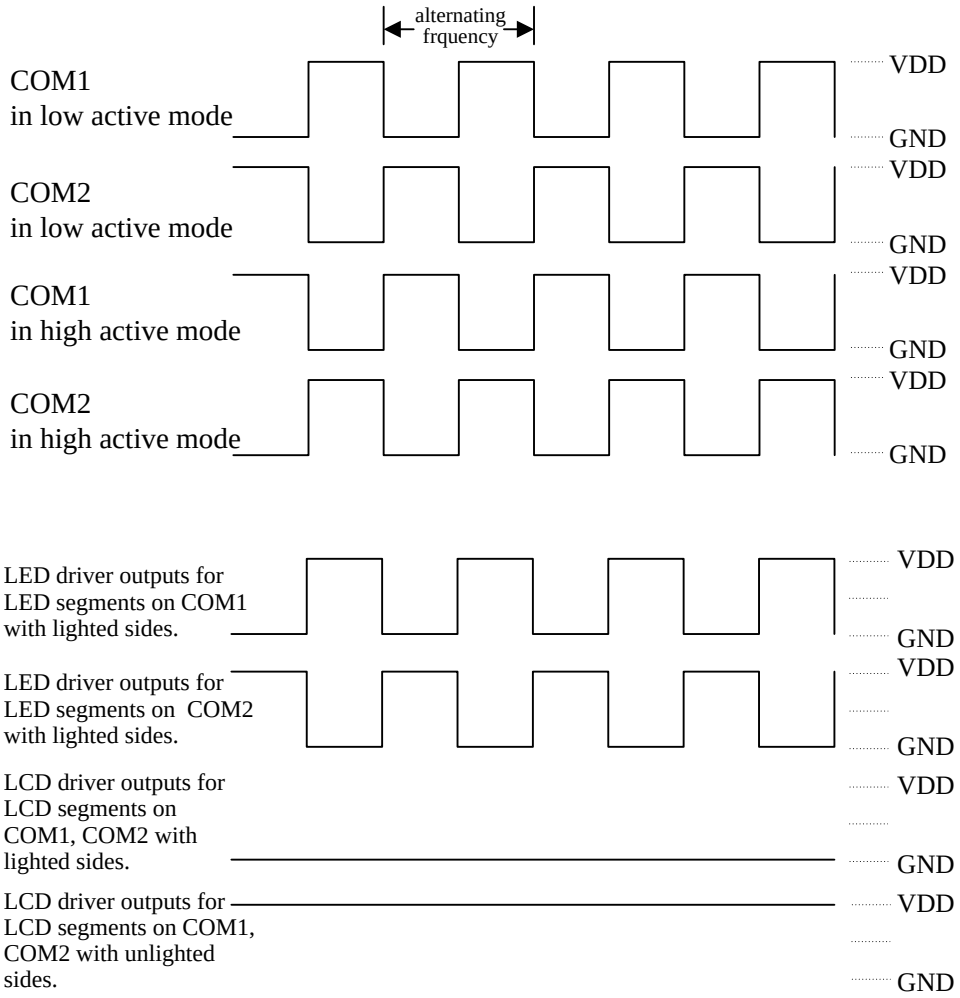


1/2 DUTY LIGHTING SYSTEM FOR LED DRIVER

(i) Display Turn Off

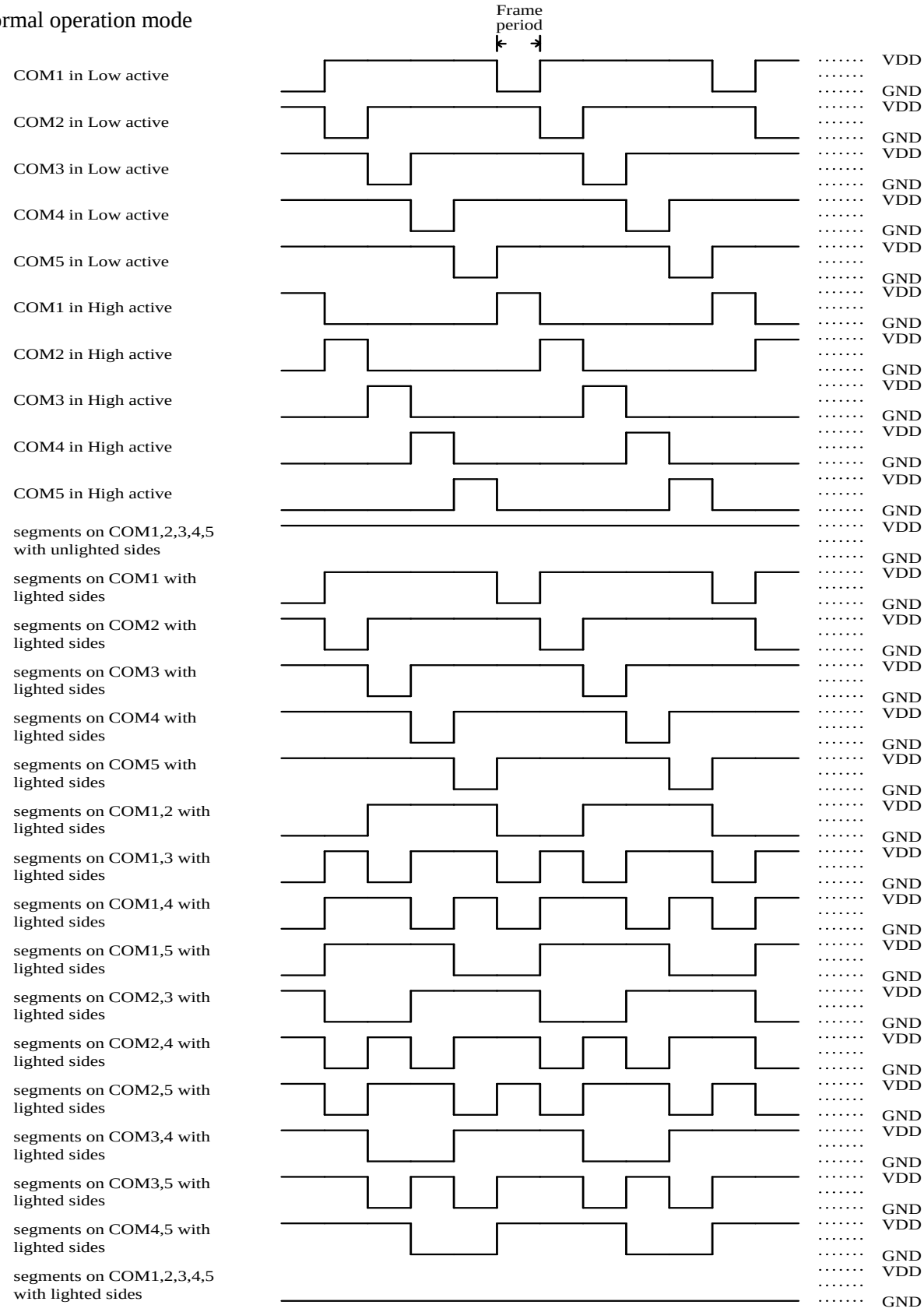


(ii) Normal operation mode



1/5 DUTY LIGHTING SYSTEM FOR LED DRIVER

Normal operation mode



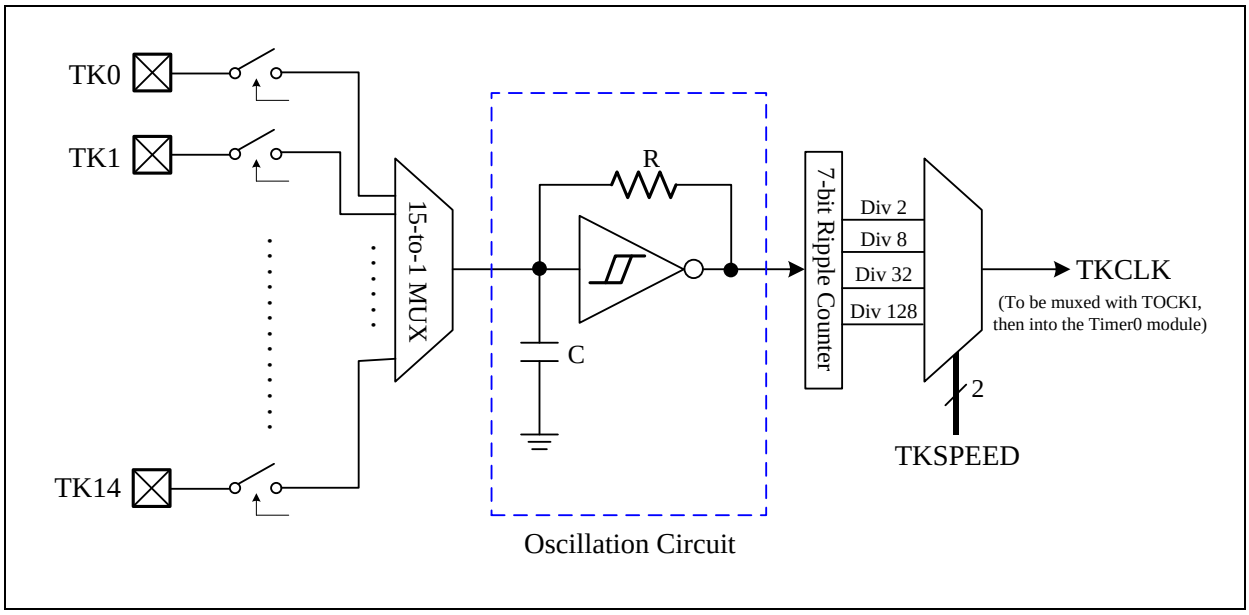
3.12 Touch Key

As noted in section 3.2 (Timer0), the Touch Key Module outputs the oscillation clock to Timer0 and counts like T0I input. Figure 3.12.1 shows the block diagram of the Touch Key module. It consists of a RC oscillator, 16-to-1 analog input select, TKSPEED control bits select the output of the frequency divider. The frequency divider divides the oscillation clock by 2, 8, 32, and 128. If TM0TOUCHKEY bit is 1, the divided clock will be sent to Timer0 to count at the rising or falling edge depends on T0IEDGE bit.

If the human finger tips close to the touch pad, the equivalent capacitance of C will be increased, that is, the oscillation frequency will be decreased.

Based on the above thesis, user program needs to observe what input channel causes the lowest Timer0 counting value in a fixed period of time, which channel of key is been touched or the finger is just approaching.

To distinguish what channel counting value is the lowest, we need another Timer (Timer1, Timer2, or WKT Timer) to set up a proper interval of time that Timer0 will not count to overflow. Based on this fixed time interval, the user program switches the Touch Key channels one after another and finds the lowest value of Timer0, which is the key in touching or approaching.



igure 3.12.1 Touch Key Oscillator Block Diagram

Figure 3.12.2 shows the flowchart how to use Timer0 and Timer1 to determine what channel of the TouchKey is pressed. Using the 16-bit Timer1 to set up a fix interval of time and utilize the Time1 interrupt to stop Timer0 and store its value if it is not overflow. Determine the lowest value of Timer0 of the desired channels that the key pressed.

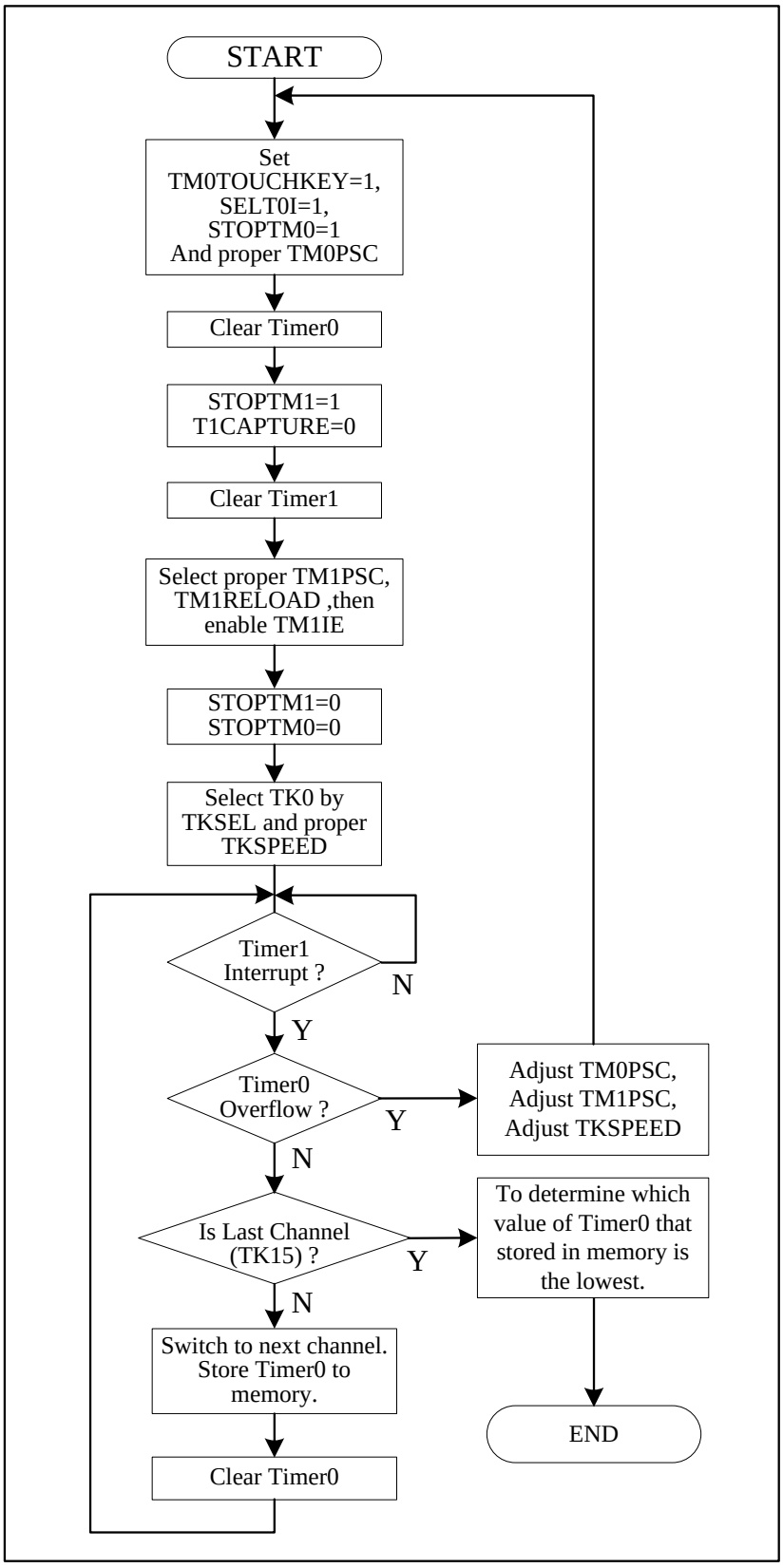


Figure 3.12.2 The recommended procedures for using the Touch Key function

3.13 System Clock Oscillator

System Clock can be operated in five different oscillation modes, which can be selected by setting the CLKS in the SYSCFG register and SELSUB, SUBE, and STOPFCK control bits in CLKCTRL register. In Slow/Fast Crystal mode, a crystal or ceramic resonator is connected to the XI and XO pins to establish oscillation. In Fast/Slow External RC mode, the external resistors and capacitors determine the oscillation frequencies. In the Fast Internal RC Mode, the on chip oscillator generates 512K/2M/4M/8 MHz system clock. In this mode, PCB Layout may have strong effect on the stability of Internal Clock Oscillator. Since power noise degrades the performance of Internal Clock Oscillator. Placing power supply bypass capacitors 1 uF and 0.1 uF very close to V_{DD}/V_{SS} pins improves the stability of clock and the overall system. Figure 3.13.1 shows the external components need to be connected with Fast/Slow crystal modes, Fast/Slow External RC modes and Fast Internal RC mode

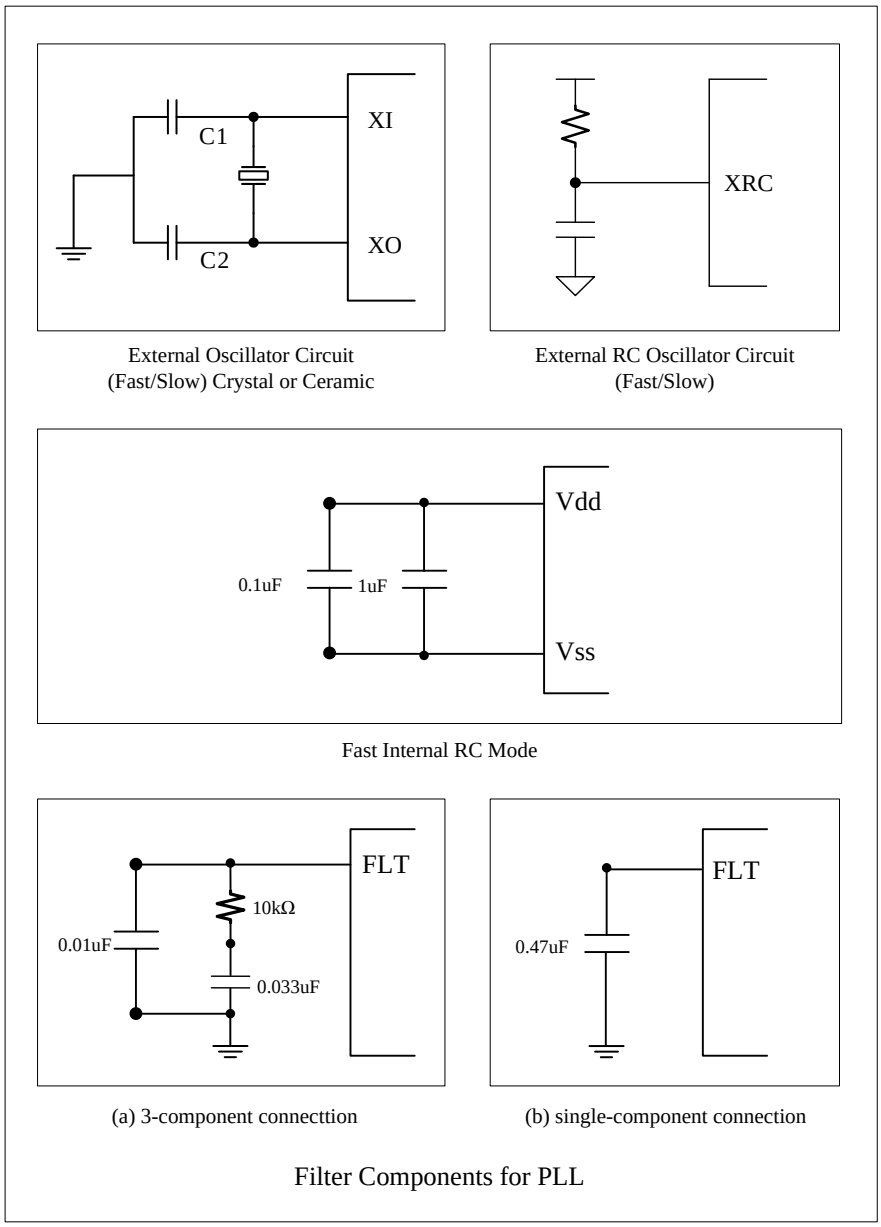
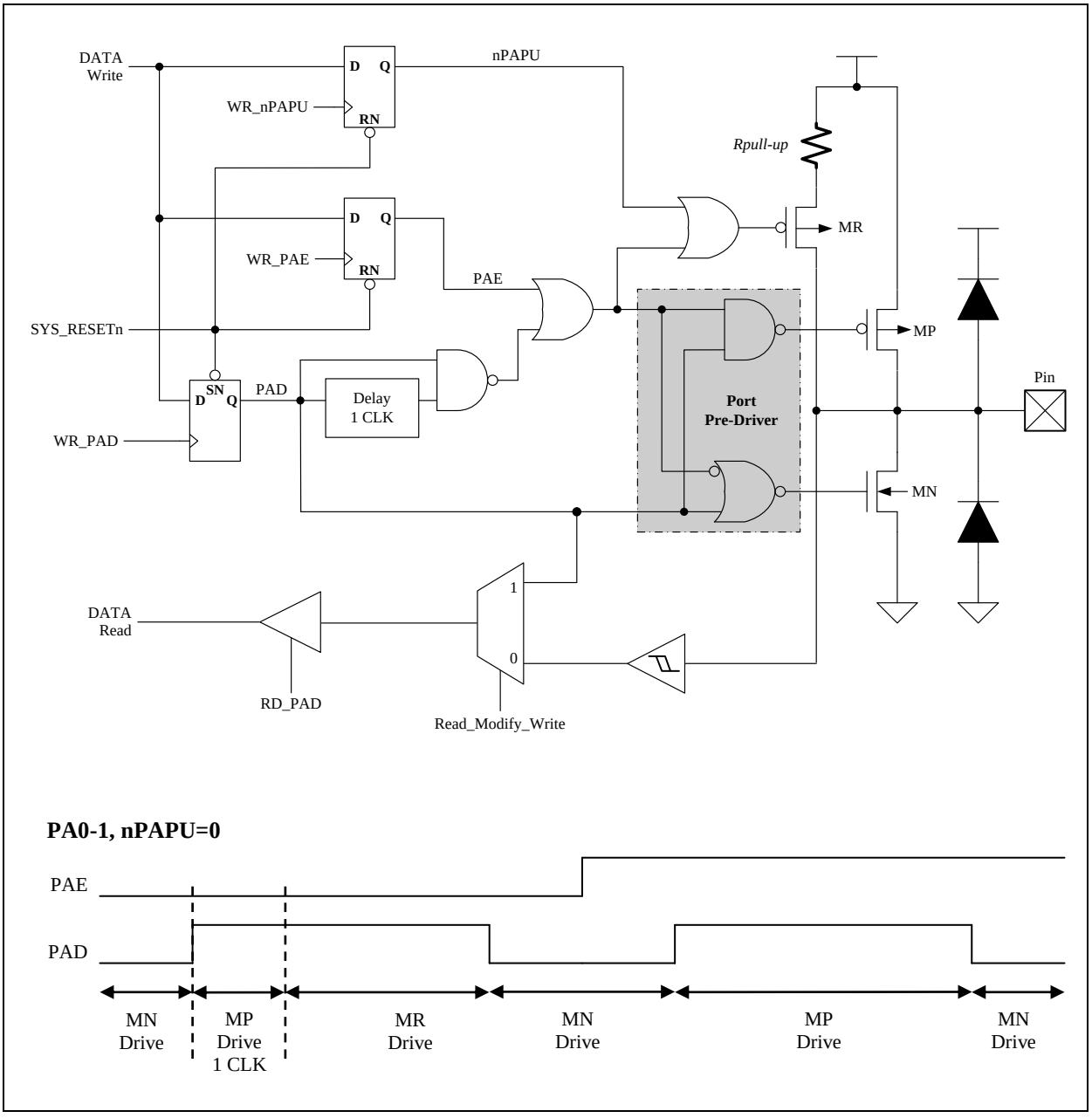


Figure 3.13.1 Oscillation Circuits

4. I/O Port

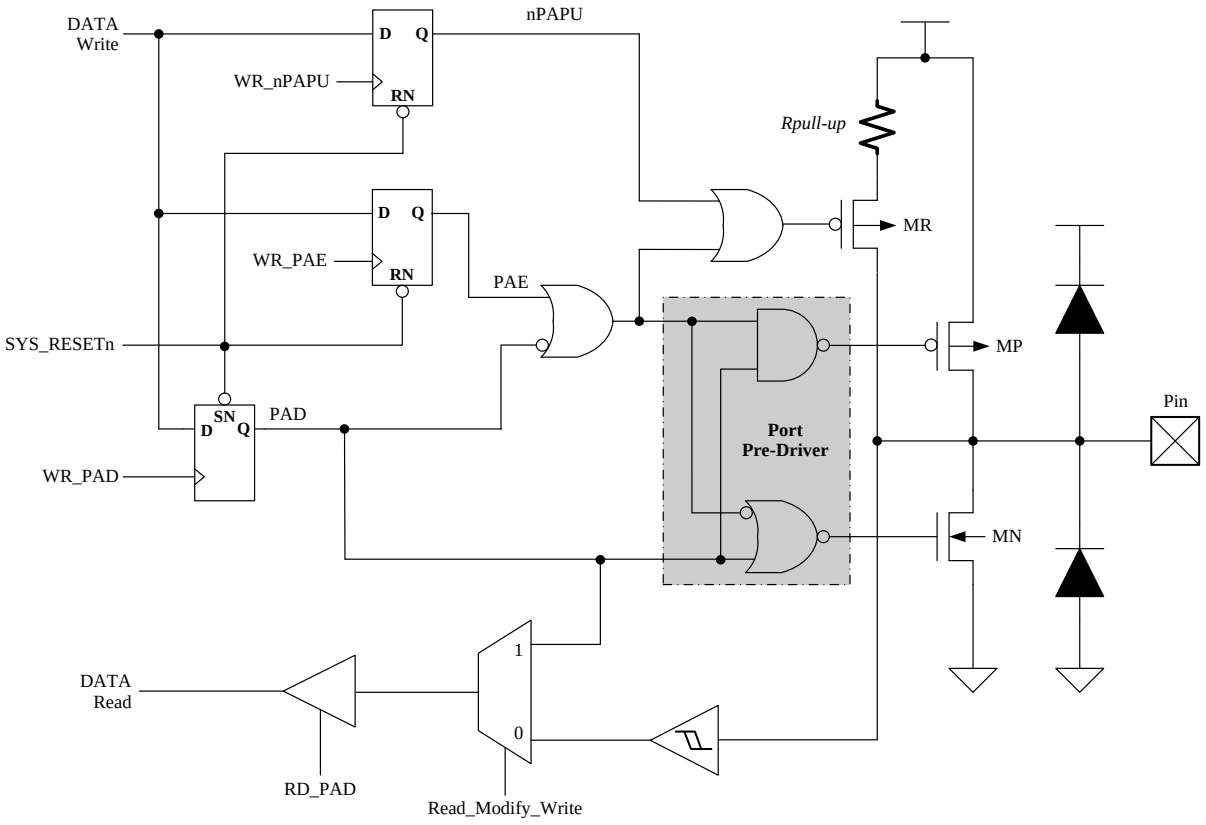
4.1 PA0-2

These pins can be used as Schmitt-trigger input, CMOS push-pull output or "pseudo-open-drain" output. The pull-up resistor is assignable to each pin by S/W setting. To use the pin in Schmitt-trigger input mode, S/W needs to set the PAE=0 and PAD=1. To use the pin in pseudo-open-drain mode, S/W sets the PAE=0. The benefit of pseudo-open-drain structure is that the output rise time can be much faster than pure open-drain structure. S/W sets PAE=1 to use the pin in CMOS push-pull output mode. Reading the pin data (PAD) has different meaning. In "Read-Modify-Write" instruction, CPU actually reads the output data register. In the others instructions, CPU reads the pin state. The so-called "Read-Modify-Write" instruction includes BSF, BCF and all instructions using F-Plane as destination.

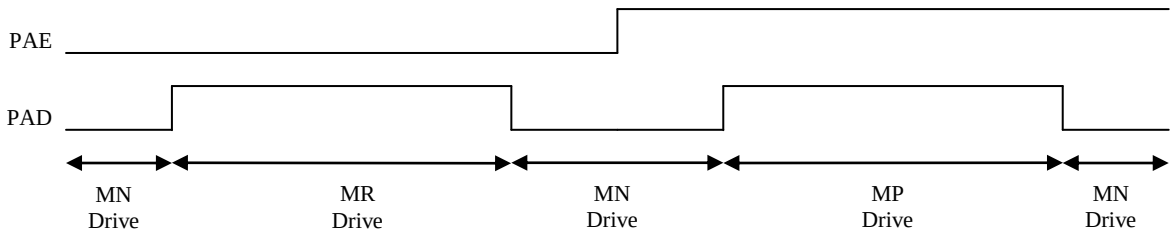


4.2 PA3-6 & PB0-7 & PD0-5 & PE0-3 & PF0-7

These pins are almost the same as PA0-1, except they do not support pseudo-open-drain mode. They can be used in pure open-drain mode, instead.

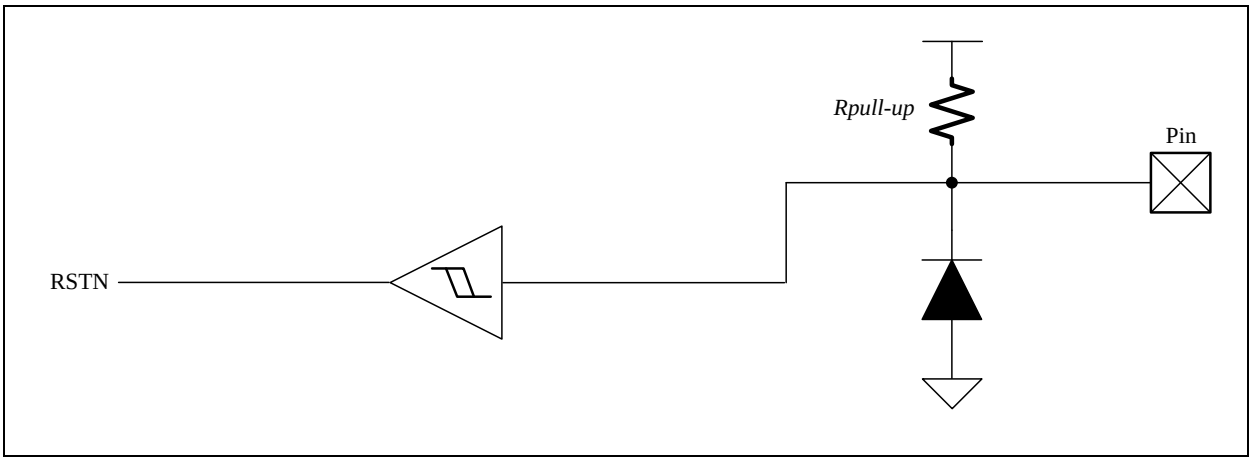


PA2-4, nPAPU=0



4.3 VPP/RSTN

This pin can be only used in Schmitt-trigger input mode. The pull-up resistor is always connected to this pin.



MEMORY MAP

F-Plane

Name	Address	R/W	Rst	Description
INDF	00.7~0	R/W	-	Not a physical register, addressing INDF actually point to the register whose address is contained in the FSR register
TIMER0	01.7~0	R/W	0	Timer 0
PC	02.7~0	R/W	0	Program Counter [7~0]
STATUS				
GBIT	03.6	R/W	0	General purpose bit
RAMBANK	03.5	R/W	0	RAM Bank Select
TO	03.4	R	0	WDT time out flag
PD	03.3	R	0	Sleep mode flag
ZFLAG	03.2	R/W	0	Zero Flag
DCFLAG	03.1	R/W	0	Decimal Carry Flag
CFLAG	03.0	R/W	0	Carry Flag
FSR	04.7~0	R/W		F-Plane File Select Register
PAD	05.6~0	R	-	Port A pin or "data register" state
		W	7f	Port A data output register
PBD	06.7~0	R	-	Port B pin or "data register" state
		W	ff	Port B data output register
RSR	07.7~0	R/W		R-Plane File Select Register
INTE1				Interrupt Enable Group 1, 1=Enable, 0=Disable
PWM0IE	08.7	R/W	0	PWM0 Interrupt Enable, 1=Enable, 0=Disable
TM2IE	08.6	R/W	0	Timer2 Interrupt Enable, 1=Enable, 0=Disable
TM1IE	08.5	R/W	0	Timer1 Interrupt Enable, 1=Enable, 0=Disable
TM0IE	08.4	R/W	0	Timer0 Interrupt Enable, 1=Enable, 0=Disable
WKTIE	08.3	R/W	0	Wakeup Timer Interrupt Enable, 1=Enable, 0=Disable If WKTIE=0, the WDT/WKT stops in Sleep mode.
XINT2E	08.2	R/W	0	XINT2 (PB0) falling Interrupt Enable, 1=Enable, 0=Disable
XINT1E	08.1	R/W	0	XINT1 (PA1) falling Interrupt Enable, 1=Enable, 0=Disable
XINT0E	08.0	R/W	0	XINT0 (PA0) falling/rising Interrupt Enable, 1=Enable, 0=Disable

Name	Address	R/W	Rst	Description
INTF1				Interrupt Flag Group 1
PWM0I	09.7	R	-	PWM0 interrupt event pending flag, set by H/W while PWM0 overflow
		W	0	write 0: clear this flag; write 1: no action.
TM2I	09.6	R	-	Timer2 interrupt event pending flag, set by H/W while Timer2 match
		W	0	write 0: clear this flag; write 1: no action.
TM1I	09.5	R	-	Timer1 interrupt event pending flag, set by H/W while Timer1 overflow
		W	0	write 0: clear this flag; write 1: no action.
TM0I	09.4	R	-	Timer0 interrupt event pending flag, set by H/W while Timer0 overflow
		W	0	write 0: clear this flag; write 1: no action.
WKTI	09.3	R	-	WKT interrupt event pending flag, set by H/W while WKT timeout
		W	0	write 0: clear this flag; write 1: no action.
XINT2	09.2	R	-	XINT2 pin falling interrupt pending flag, set by H/W at INT2 pin's falling edge. Belongs to XINTA interrupt
		W	0	write 0: clear this flag; write 1: no action.
XINT1	09.1	R	-	XINT1 pin falling interrupt pending flag, set by H/W at INT1 pin's falling edge. Belongs to XINTA interrupt
		W	0	write 0: clear this flag; write 1: no action
XINT0	09.0	R	-	XINT0 pin falling/rising interrupt pending flag, set by H/W at INT0 pin's falling/rising edge. Belongs to XINTA interrupt
		W	0	write 0: clear this flag; write 1: no action
TM1L				
TIMER1L	0a.7~0	R	-	Timer 1 Counter low byte
TIMER1L	0a.7~0	W	0	Timer 1 reload data low byte
TM1H				
TIMER1H	0b.7~0	R	-	Timer 1 Counter high byte
TIMER1H	0b.7~0	W	0	Timer 1 reload data high byte
PWM0				
PWM0DUTY	0c.7~0	R/W	0	PWM0 duty 8-bit
TM1CTRL				
CLRTM2	0d.7	R/W	0	Timer2 clear and hold when this bit is "1"
RFCSTOP	0d.6	R/W	1	Stop RFC Counter/Loop
CLRRFC	0d.5	R/W	1	Clear RFC Counter
TM1SET	0d.4	R/W	0	Timer1 set FFFFh and hold when this bit is "1"
CLRTM1	0d.3	R/W	0	Timer1 clear and hold when this bit is "1"
CLRPWM0	0d.2	R/W	1	PWM0 clear and hold when this bit is "1"
STOPTM1	0d.1	R/W	0	Stop Timer1 when this bit is "1"
STOPTM0	0d.0	R/W	0	Stop Timer0 when this bit is "1"
INTE2				Interrupt Enable Group 2, 1=Enable, 0=Disable
RFCINTE	0e.1	R/W	0	RFC Interrupt Enable, 1=Enable, 0=Disable
XINT3E	0e.0	R/W	0	XINT3 (PB1) falling Interrupt Enable.

Name	Address	R/W	Rst	Description
INTF2				Interrupt Flag Group 2
RFCINT	0f.1	R	-	RFC interrupt event pending flag, set by H/W while RFCCNT overflow
		W	0	write 0: clear this flag; write 1: no action.
XINT3	0f.0	R	-	XINT3 pin falling/rising interrupt pending flag, set by H/W at INT3 pin's falling edge. Belongs to XINTB interrupt.
		W	0	write 0: clear this flag; write 1: no action.
PDD	12.5~0	R	-	Port D pin or "data register" state
		W	3f	Port D data output register
PED	13.3~0	R	-	Port E pin or "data register" state
		W	f	Port E data output register
PFD	14.7~0	R	-	Port F pin or "data register" state
		W	ff	Port F data output register
PWM1DUTY	16.7~0	R/W	0	PWM1 duty 8-bit
RFCNTH	17.7~0	R	0	RFC counter high byte RFCNT [15:8]
RFCNTL	18.7~0	R	0	RFC counter low byte RFCNT [7:0]
CFGH	1a.5~0	R	-	CFGWord [13:8]
CFGL	1b.7~0	R	-	CFGWord [7:0]
-	1c~1f	-	0	Unused Area (for future extension)
SRAM	20~7f	R/W	-	2 banks of internal SRAM

R-Plane

Name	Address	R/W	Rst	Description
INDR	00.7~0	R/W	-	Not a physical register, addressing INDR actually point to the register whose address is contained in the RSR register
TM0RELOAD	01.7~0	W	0	Time0 Reload Data
OPTION				
CAPOLARITY	02.7	W	0	Timer0 Capture polarity. 0: High level capture, 1: Low level capture
T0CAPTURE	02.6	W	0	1: Timer0 works in CAPTURE Mode; 0: Timer0 works in COUNTER Mode
T0IEDGE	02.5	W	0	1: T0CKI/TKCLK falling edge; 0: T0CKI/TKRC rising edge for Timer0 Prescaler count
SELT0I	02.4	W	0	1: T0CKI/TKCLK as Timer0 Prescaler clock; 0: Instruction Cycle as Timer0 Prescaler clock
TM0PSC	02.3~0	W	0	Timer0 Pre-Scale 0000: div1 0001: div2 0010: div4 0011: div8 0100: div16 0101: div32 0110: div64 0111: div128 1xxx: div256
PWRDOWN	03	W		write this register to enter Power-Down Mode
CLRWDT	04	W		write this register to clear WDT
PAE	05.6~3	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
	05.2~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is pseudo-open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
PBE	06.7~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
PWM0PRD	07.7~0	W	ff	PWM0 Period. ff=256
PAPU	08.6~0	W	7f	PA pull-up, 0: Enable, 1: Disable
PBPU	09.7~0	W	ff	PB pull-up, 0: Enable, 1: Disable
PUN				
PFPUN1	0a.4	W	1	PF7~4 pull-up, 0=Enable, 1:Disable
PFPUN0	0a.3	W	1	PF3~0 pull-up, 0=Enable, 1:Disable
PEPUN	0a.2	W	1	PE3~0 pull-up, 0=Enable, 1:Disable
PDPUN1	0a.1	W	1	PD2~5 pull-up, 0=Enable, 1:Disable
PDPUN0	0a.0	W	1	PD0~1 pull-up, 0=Enable, 1:Disable

Name	Address	R/W	Rst	Description
WKTPSC				
SIRCSEL	0b.6~5	W	11	SIRC 00=128K 01=32K 10=8K 11=2K
BUZOUTE	0b.4	W	0	0: disable BUZZER output to PD1 pin, 1: enable BUZZER output to PD1 pin
PWM0E	0b.3	W	0	1: PWM0 output to PA6 pin, 0: PA6 is general I/O pin
PWM1E	0b.2	W	0	1: PWM1 output to PD0 pin, 0: PD0 is general I/O pin
WKTPSC	0b.1~0	W	11	WDT/WKT period, 00=0.875 ms, 01=1.75 ms, 10=28 ms, 11=112 ms
TM1PSC,PWM0PSC				
BUZ_EN	0c.5	W	0	0: disable BUZZER timer counting 1: enable BUZZER timer counting
PWM0N	0c.4	W	0	1: PWM0 negative output to PA6 pin, 0: PWM0 positive output
PWM0PSC	0c.3~2	W	0	PWM0 Pre-Scale, 00: Fosc 01: Fosc/2 10: Fosc/4 11: Fosc/8
T1CAPTURE	0c.1	W	0	1=Timer1 working in capture mode, Timer1 measure CAPT period (time between successive rising or falling edges of CAPT pin)
TM1PSC	0c.0	W	0	Timer1 Clock Select, 1: Fosc, 0: Instruction cycle (Fosc/2)
TM2CTRL				
INT3EDGE	0d.6	W	0	0: INT3 pin falling interrupt; 1: INT3 pin rising interrupt
INT0EDGE	0d.5	W	0	0: INT0 pin falling interrupt; 1: INT0 pin rising interrupt
TIMER2CLK	0d.4	W	0	0: Timer2 clock is Slow Clock; 1: Timer2 clock is CPUCLK/128
TIMER2DIV	0d.3~2	W	0	Timer2 Interrupt is Timer2 clock divided by 00: 32768 01: 16384 10: 8192 11: 128
LCDCLK	0d.1~0	W	0	LCDCLK is Timer2 clock divided by 00: 128 01: 64 10: 32 11: 16

Name	Address	R/W	Rst	Description
TKCTRL				
WKTTOUCH	0e.7	W	0	1: Wake up Timer Source is Touch Key
TM0TOUCHKEY	0e.6	W	0	1: Timer0 Source is Touch Key, 0: Timer0 Source is CPUCLK/T0CKI
TKSPEED	0e.5~4	W	0	Touch Key Oscillation Frequency Select 00: Fastest Touch Key clock 11: Slowest Touch Key clock
TKSEL	0e.3~0	W	f	Touch Key Channel Select 0000: TK0 0001: TK1 …… 1110: TK14 1111: NO channel
TESTREG	0f.1~0	W	0	Test reg
CLKCTRL				
KICKE	10.6	W	1	Enable high gain to kick SXT
ATOSAVE	10.5	W	0	Auto Save W register and STATUS register when interrupt, and restore them when exit from interrupt
SELSUB	10.4	W	0	select Slow Clock as cpu clk
STOPFCK	10.3	W	0	stop Fast Clock
SUBE	10.2	W	0	Slow Clock enable
SUBTYP	10.1~0	W	0	Slow Clock type 0:SXT; 1:SIRC; 2:XRC, 3:TKCLK
LCDCTRL				
LCDICTR	11.7~6	W	11	LCD current control VDD=3V 11 => 29.0 uA 10 => 14.9 uA 01 => 7.5 uA 00 => 6 uA
LCDDUTY	11.5~3	W	111	LCD/LED Duty 000: Static 001: 1/2 duty 010: 1/3 duty 011: 1/4 duty 100: 1/5 duty 101: 1/6 duty 110: 1/7 duty 111: 1/8 duty
LCDBIAS	11.2	W	1	LCD Bias 0: 1/2 Bias 1: 1/3 Bias
LCDBRIT	11.1~0	W	10	LCD Brightness 00: Most darkness 11: Most brightness
PDE	12.5~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
PEE	13.3~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output

Name	Address	R/W	Rst	Description
PFE	14.7~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
LCDPIN				
LCDPIN0	15.7	W	0	0: I/O PE3 1: LCDPIN or LEDPIN SEG16
	15.6	W	0	0: I/O PE2 1: LCDPIN or LEDPIN SEG17
	15.5	W	0	0: I/O PE1 1: LCDPIN or LEDPIN SEG18
	15.4	W	0	0: I/O PE0 1: LCDPIN or LEDPIN SEG19
	15.3	W	0	0: I/O PF7 1: LCDPIN or LEDPIN SEG20
	15.2	W	0	0: I/O PF6 1: LCDPIN or LEDPIN SEG21
	15.1	W	0	0: I/O PF5 1: LCDPIN or LEDPIN SEG22
	15.0	W	0	0: I/O PF4 1: LCDPIN or LEDPIN SEG23
LCDPIN1	16.7	W	0	0: I/O PF3 1: LCDPIN or LEDPIN SEG24
	16.6	W	0	0: I/O PF2 1: LCDPIN or LEDPIN SEG25
	16.5	W	0	0: I/O PF1 1: LCDPIN or LEDPIN SEG26
	16.4	W	0	0: I/O PF0 1: LCDPIN or LEDPIN SEG27
	16.3	W	0	0: I/O PD5 1: LCDPIN SEG28/COM7 or LEDPIN COM7
	16.2	W	0	0: I/O PD4 1: LCDPIN SEG29/COM6 or LEDPIN COM6
	16.1	W	0	0: I/O PD3 1: LCDPIN SEG30/COM5 or LEDPIN COM5
	16.0	W	0	0: I/O PD2 1: LCDPIN SEG31/COM4 or LEDPIN COM4
FIRC,PLL				
CLKSEL	17.7~6	W	01	PLLCLK & FIRCCLK select 00: PLLCLK=PLLOUT/1, FIRCCLK=8M 01: PLLCLK=PLLOUT/2, FIRCCLK=4M 10: PLLCLK=PLLOUT/4, FIRCCLK=2M 11: PLLCLK=PLLOUT/8, FIRCCLK=512K
PLLM	17.5~0	W	0	PLLOUT=32768 * 4 *(64+PLLM)
BUZ				
BUZ_PSC	18.7~6	W	0	BUZZER clock Prescaler 00: BUZZER clock is “Instruction Cycle” divided by 4 01: BUZZER clock is “Instruction Cycle” divided by 8 10: BUZZER clock is “Instruction Cycle” divided by 16 11: BUZZER clock is “Instruction Cycle” divided by 32
BUZ_PROD	18.5~0	W	0	BUZZER Period Data. BUZZER output is BUZZER clock divided by BUZ_PROD

Name	Address	R/W	Rst	Description
RFCCON				
TM1STPRFC	19.7	W	0	Timer1 overflow to control RFC
TM0STPRFC	19.6	W	0	Timer0 overflow to control RFC
CX1E	19.5	W	0	enable CX1 osc, select CX1 as RFC Counter CLK
CX0E	19.4	W	0	enable CX0 osc, select CX0 as RFC Counter CLK
RFC3E	19.3	W	0	select RFC3 pin for RFC loop
RFC2E	19.2	W	0	select RFC2 pin for RFC loop
RFC1E	19.1	W	0	select RFC1 pin for RFC loop
RFC0E	19.0	W	0	select RFC0 pin for RFC loop
PBWKUP	1a.7~2	W	0	1:PB7~2 Wake up enable, 0:disable
LCD/LED				
LEDON	1b.3	W	0	1: LED display on 0: LED display off
LEDTYPE	1b.2	W	0	1: LED high active 0: low active
SELLED	1b.1	W	0	0: Select LCD 1: Select LED
LCDON	1b.0	W	0	1: LCD Enable, 0: Disable
LCDRAM	40~5f	W/R	xx	LCD RAM. Initial values are not defined.
	34~3f	-	-	not used. Read as 0x00
SRAM	60~ff	W/R	-	160 bytes SRAM

Instruction Set

Each instruction is a 14-bit word divided into an OPCODE, which specifies the instruction type, and one or more operands, which further specify the operation of the instruction. The instructions can be categorized as byte-oriented, bit-oriented and literal operations list in the following table.

For byte-oriented instructions, “f” or “r” represents the address designator and “d” represents the destination designator. The address designator is used to specify which address in Program memory is used by the instruction. The destination designator specifies where the result of the operation is to be placed. If “d” is “0”, the result is placed in the W register. If “d” is “1”, the result is placed in the address specified in the instruction.

For bit-oriented instructions, “b” represents a bit field designator, which selects the number of the bit affected by the operation, while “f” represents the address designator. For literal operations, “k” represents the literal or constant value.

Field/Legend	Description
f	F-Plane Register File Address
r	R-Plane Register File Address
b	Bit address
k	Literal. Constant data or label
d	Destination selection field. 0: Working register, 1: Register file
W	Working Register
Z	Zero Flag
C	Carry Flag
DC	Decimal Carry Flag
PC	Program Counter
TOS	Top Of Stack
GIE	Global Interrupt Enable Flag (i-Flag)
[]	Option Field
( )	Contents
.	Bit Field
B	Before
A	After
←	Assign direction

Mnemonic		Op Code	Cycle	Flag Affect	Description
Byte-Oriented File Register Instruction					
ADDWF	f,d	00 0111 dfff ffff	1	C,DC,Z	Add W to f
ANDWF	f,d	00 0101 dfff ffff	1	Z	AND W to f
CLRF	f	00 0001 1fff ffff	1	Z	Clear f
CLRW		00 0001 0100 0000	1	Z	Clear W
COMF	f,d	00 1001 dfff ffff	1	Z	Invert F bit by bit
DECF	f,d	00 0011 dfff ffff	1	Z	Decrement of f
DECFSZ	f,d	00 1011 dfff ffff	1 or 2	-	Decrease f, skip if zero
INCF	f,d	00 1010 dfff ffff	1	Z	Increment of f
INCFSZ	f,d	00 1111 dfff ffff	1 or 2	-	Increase f, skip if zero
IORWF	f,d	00 0100 dfff ffff	1	Z	OR W to f
MOVFW	f	00 1000 0fff ffff	1	-	Move f to W
MOVWF	f	00 0000 1fff ffff	1	-	Move W to f
MOVRW	r	01 1111 rrrr rrrr	1	-	Move r to W
MOVWR	r	01 1110 rrrr rrrr	1	-	Move W to r
RLF	f,d	00 1101 dfff ffff	1	C	F rotate to left
RRF	f,d	00 1100 dfff ffff	1	C	F rotate to right
SUBWF	f,d	00 0010 dfff ffff	1	C,DC,Z	Subtract W from f
SWAPF	f,d	00 1110 dfff ffff	1	-	Swap high and low nibble of f
TESTZ	f	00 1000 1fff ffff	1	Z	Test f if zero
XORWF	f,d	00 0110 dfff ffff	1	Z	XOR W to f
Bit-Oriented File Register Instruction					
BCF	f,b	01 000b bbff ffff	1	-	Bit clear f
BSF	f,b	01 001b bbff ffff	1	-	Bit set f
BTFSC	f,b	01 010b bbff ffff	1 or 2	-	Bit test f, skip if clear
BTFSS	f,b	01 011b bbff ffff	1 or 2	-	Bit test f, skip if set
Literal and Control Instruction					
ADDLW	k	01 1100 kkkk kkkk	1	C,DC,Z	Add literal to W
ANDLW	k	01 1011 kkkk kkkk	1	Z	AND literal to W
XORLW	k	01 1101 kkkk kkkk	1	Z	XOR literal to W
CALL	k	10 kkkk kkkk kkkk	2	-	Subroutine call
CLRWDT		01 1110 0000 0100	1	PD,TO	Clear watchdog timer
GOTO	k	11 kkkk kkkk kkkk	2	-	Unconditional branch
IORLW	k	01 1010 kkkk kkkk	1	Z	OR literal to W
MOVLW	k	01 1001 kkkk kkkk	1	-	Move literal to W
NOP		00 0000 0000 0000	1	-	No operation
RET		00 0000 0100 0000	2	-	Return from CALL
RETI		00 0000 0110 0000	2	-	Return from interrupt
RETLW	k	01 1000 kkkk kkkk	2	-	Return with literal to W
SLEEP		01 1110 0000 0011	1	PD,TO-	Power down

ADDLW	Add Literal “k” and W	
Syntax	ADDLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) + k	
Status Affected	C, DC, Z	
OP-Code	01 1100 kkkk kkkk	
Description	The contents of the W register are added to the eight-bit literal ‘k’ and the result is placed in the W register.	
Cycle	1	
Example	ADDLW 0x15	B : W = 0x10 A : W = 0x25

ADDWF	Add W and “f”	
Syntax	ADDWF f [,d]	
Operands	f : 00h ~ 7Fh d : 0, 1	
Operation	(Destination) ← (W) + (f)	
Status Affected	C, DC, Z	
OP-Code	00 0111 dfff ffff	
Description	Add the contents of the W register with register ‘f’. If ‘d’ is 0, the result is stored in the W register. If ‘d’ is 1, the result is stored back in register ‘f’.	
Cycle	1	
Example	ADDWF FSR, 0	B : W = 0x17, FSR = 0xC2 A : W = 0xD9, FSR = 0xC2

ANDLW	Logical AND Literal "k" with W	
Syntax	ANDLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) ‘AND’ k	
Status Affected	Z	
OP-Code	01 1011 kkkk kkkk	
Description	The contents of W register are AND’ed with the eight-bit literal 'k'. The result is placed in the W register.	
Cycle	1	
Example	ANDLW 0x5F	B : W = 0xA3 A : W = 0x03

ANDWF	AND W with “f”	
Syntax	ANDWF f [,d]	
Operands	f : 00h ~ 7Fh d : 0, 1	
Operation	(Destination) ← (W) ‘AND’ (f)	
Status Affected	Z	
OP-Code	00 0101 dfff ffff	
Description	AND the W register with register ‘f’. If ‘d’ is 0, the result is stored in the W register. If ‘d’ is 1, the result is stored back in register ‘f’.	
Cycle	1	
Example	ANDWF FSR, 1	B : W = 0x17, FSR = 0xC2 A : W = 0x17, FSR = 0x02

BCF	Clear "b" bit of "f"	
Syntax	BCF f [,b]	
Operands	f : 00h ~ 3Fh b : 0 ~ 7	
Operation	(f.b) ← 0	
Status Affected	-	
OP-Code	01 000b bbff ffff	
Description	Bit 'b' in register 'f' is cleared.	
Cycle	1	
Example	BCF FLAG_REG, 7	B : FLAG_REG = 0xC7 A : FLAG_REG = 0x47

BSF	Set "b" bit of "f"	
Syntax	BSF f [,b]	
Operands	f : 00h ~ 3Fh b : 0 ~ 7	
Operation	(f.b) ← 1	
Status Affected	-	
OP-Code	01 001b bbff ffff	
Description	Bit 'b' in register 'f' is set.	
Cycle	1	
Example	BSF FLAG_REG, 7	B : FLAG_REG = 0x0A A : FLAG_REG = 0x8A

BTFSC	Test "b" bit of "f", skip if clear(0)	
Syntax	BTFSC f [,b]	
Operands	f : 00h ~ 3Fh b : 0 ~ 7	
Operation	Skip next instruction if (f.b) = 0	
Status Affected	-	
OP-Code	01 010b bbff ffff	
Description	If bit 'b' in register 'f' is '1', then the next instruction is executed. If bit 'b' in register 'f' is '0', then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 BTFSC FLAG, 1 TRUE GOTO SUB1 FALSE ...	B : PC = LABEL1 A : if FLAG.1 = 0, PC = FALSE if FLAG.1 = 1, PC = TRUE

BTFSS	Test "b" bit of "f", skip if set(1)	
Syntax	BTFSS f [,b]	
Operands	f : 00h ~ 3Fh b : 0 ~ 7	
Operation	Skip next instruction if (f.b) = 1	
Status Affected	-	
OP-Code	01 011b bbff ffff	
Description	If bit 'b' in register 'f' is '0', then the next instruction is executed. If bit 'b' in register 'f' is '1', then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle		
Example	LABEL1 BTFSS FLAG, 1 TRUE GOTO SUB1 FALSE ...	B : PC = LABEL1 A : if FLAG.1 = 0, PC = TRUE if FLAG.1 = 1, PC = FALSE

CALL	Call subroutine "k"	
Syntax	CALL k	
Operands	K : 00h ~ FFFh	
Operation	Operation: TOS ← (PC)+ 1, PC.11~0 ← k	
Status Affected	-	
OP-Code	10 kkkk kkkk kkkk	
Description	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The 12-bit immediate address is loaded into PC bits <11:0>. CALL is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 CALL SUB1	B : PC = LABEL1 A : PC = SUB1, TOS = LABEL1+1

CLRF	Clear “f”	
Syntax	CLRF f	
Operands	f : 00h ~ 7Fh	
Operation	(f) ← 00h, Z ← 1	
Status Affected	Z	
OP-Code	00 0001 1fff ffff	
Description	The contents of register ‘f’ are cleared and the Z bit is set.	
Cycle	1	
Example	CLRF FLAG_REG	B : FLAG_REG = 0x5A A : FLAG_REG = 0x00, Z = 1

CLRW	Clear W	
Syntax	CLRW	
Operands	-	
Operation	(W) ← 00h, Z ← 1	
Status Affected	Z	
OP-Code	00 0001 0100 0000	
Description	W register is cleared and Zero bit (Z) is set.	
Cycle	1	
Example	CLRW	B : W = 0x5A A : W = 0x00, Z = 1

CLRWDT	Clear Watchdog Timer	
Syntax	CLRWDT	
Operands	-	
Operation	WDTE ← 00h	
Status Affected	TO,PD	
OP-Code	00 0000 0000 0100	
Description	CLRWDT instruction enables and resets the Watchdog Timer.	
Cycle	1	
Example	CLRWDT	B : WDT counter = ? A : WDT counter = 0x00

COMF	Complement "f"	
Syntax	COMF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← ($\bar{f}$)	
Status Affected	Z	
OP-Code	00 1001 dfff ffff	
Description	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	COMF REG1,0	B : REG1 = 0x13 A : REG1 = 0x13, W = 0xEC

DECF	Decrement "f"	
Syntax	DECF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (f) - 1	
Status Affected	Z	
OP-Code	00 0011 dfff ffff	
Description	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	DECF CNT, 1	B : CNT = 0x01, Z = 0 A : CNT = 0x00, Z = 1

DECFSZ	Decrement "f", Skip if 0	
Syntax	DECFSZ f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (f) - 1, skip next instruction if result is 0	
Status Affected	-	
OP-Code	00 1011 dfff ffff	
Description	The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead, making it a 2 cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 DECFSZ CNT, 1 GOTO LOOP CONTINUE	B : PC = LABEL1 A : CNT = CNT – 1 if CNT=0, PC = CONTINUE if CNT≠0, PC = LABEL1+1

GOTO	Unconditional Branch	
Syntax	GOTO k	
Operands	k : 00h ~ FFFh	
Operation	PC.11~0 ← k	
Status Affected	-	
OP-Code	11 kkkk kkkk kkkk	
Description	GOTO is an unconditional branch. The 12-bit immediate value is loaded into PC bits <11:0>. GOTO is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 GOTO SUB1	B : PC = LABEL1 A : PC = SUB1

INCF	Increment “f”
Syntax	INCF f [,d]
Operands	f : 00h ~ 7Fh
Operation	(destination) ← (f) + 1
Status Affected	Z
OP-Code	00 1010 dfff ffff
Description	The contents of register ‘f’ are incremented. If ‘d’ is 0, the result is placed in the W register. If ‘d’ is 1, the result is placed back in register ‘f’.
Cycle	1
Example	INCF CNT, 1 B : CNT = 0xFF, Z = 0 A : CNT = 0x00, Z = 1

INCFSZ	Increment “f”, Skip if 0
Syntax	INCFSZ f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	(destination) ← (f) + 1, skip next instruction if result is 0
Status Affected	-
OP-Code	00 1111 dfff ffff
Description	The contents of register ‘f’ are incremented. If ‘d’ is 0, the result is placed in the W register. If ‘d’ is 1, the result is placed back in register ‘f’. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead, making it a 2 cycle instruction.
Cycle	1 or 2
Example	LABEL1 INCFSZ CNT, 1 GOTO LOOP CONTINUE B : PC = LABEL1 A : CNT = CNT + 1 if CNT=0, PC = CONTINUE if CNT≠0, PC = LABEL1+1

IORLW	Inclusive OR Literal with W
Syntax	IORLW k
Operands	k : 00h ~ FFh
Operation	(W) ← (W) OR k
Status Affected	Z
OP-Code	01 1010 kkkk kkkk
Description	The contents of the W register is OR’ed with the eight-bit literal 'k'. The result is placed in the W register.
Cycle	1
Example	IORLW 0x35 B : W = 0x9A A : W = 0xBF, Z = 0

IORWF	Inclusive OR W with "f"	
Syntax	IORWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (W) OR (f)	
Status Affected	Z	
OP-Code	00 0100 dfff ffff	
Description	Inclusive OR the W register with register 'f'. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.	
Cycle	1	
Example	IORWF RESULT, 0	B : RESULT = 0x13, W = 0x91 A : RESULT = 0x13, W = 0x93, Z = 0

MOVFW	Move "f" to W	
Syntax	MOVFW f	
Operands	f : 00h ~ 7Fh	
Operation	(W) ← (f)	
Status Affected	-	
OP-Code	00 1000 0fff ffff	
Description	The contents of register f are moved to W register.	
Cycle	1	
Example	MOVF FSR, 0	B : W = ? A : W ← f, if W = 0 Z = 1

MOVLW	Move Literal to W	
Syntax	MOVLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← k	
Status Affected	-	
OP-Code	01 1001 kkkk kkkk	
Description	The eight-bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.	
Cycle	1	
Example	MOVLW 0x5A	B : W = ? A : W = 0x5A

MOVWF	Move W to "f"	
Syntax	MOVWF f	
Operands	f : 00h ~ 7Fh	
Operation	(f) ← (W)	
Status Affected	-	
OP-Code	00 0000 1fff ffff	
Description	Move data from W register to register 'f'.	
Cycle	1	
Example	MOVWF REG1	B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

MOVWR	Move W to “r”	
Syntax	MOVWR r	
Operands	r : 00h ~ FFh	
Operation	(r) ← (W)	
Status Affected	-	
OP-Code	01 1110 rrrr rrrr	
Description	Move data from W register to register ‘r’.	
Cycle	1	
Example	MOVWR REG1	B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

MOVRW	Move “r” to W	
Syntax	MOVRW r	
Operands	r : 20h ~ FFh	
Operation	(W) ← (r)	
Status Affected	-	
OP-Code	01 1111 rrrr rrrr	
Description	Move data from register ‘r’ to W register.	
Cycle	1	
Example	MOVRW REG1	B : REG1 = 0x4F, W = ? A : REG1 = 0x4F, W = 0x4F

NOP	No Operation	
Syntax	NOP	
Operands	-	
Operation	No Operation	
Status Affected	-	
OP-Code	00 0000 0000 0000	
Description	No Operation	
Cycle	1	
Example	NOP	-

RETI	Return from Interrupt	
Syntax	RETI	
Operands	-	
Operation	PC ← TOS, GIE ← 1	
Status Affected	-	
OP-Code	00 0000 0110 0000	
Description	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in to the PC. Interrupts are enabled. This is a two-cycle instruction.	
Cycle	2	
Example	RETFIE	A : PC = TOS, GIE = 1

RETLW — Return with Literal in W

Syntax	RETLW k
Operands	k : 00h ~ FFh
Operation	PC ← TOS, (W) ← k
Status Affected	-
OP-Code	01 1000 kkkk kkkk
Description	The W register is loaded with the eight-bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Cycle	2
Example	CALL TABLE B : W = 0x07 : A : W = value of k8 TABLE ADDWF PCL,1 RETLW k1 RETLW k2 : RETLW kn

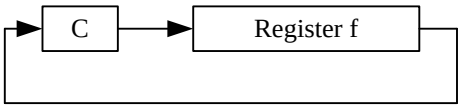
RET — Return from Subroutine

Syntax	RET
Operands	-
Operation	PC ← TOS
Status Affected	-
OP-Code	00 0000 0100 0000
Description	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.
Cycle	2
Example	RETURN A : PC = TOS

RLF — Rotate Left f through Carry

Syntax	RLF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	
Status Affected	C
OP-Code	00 1101 dfff ffff
Description	The contents of register ‘f’ are rotated one bit to the left through the Carry Flag. If ‘d’ is 0, the result is placed in the W register. If ‘d’ is 1, the result is stored back in register ‘f’.
Cycle	1
Example	RLF REG1,0 B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 1100 1100, C = 1

RRF — Rotate Right "f" through Carry

Syntax	RRF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	
Status Affected	C
OP-Code	00 1100 dfff ffff
Description	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
Cycle	1
Example	RRF REG1,0 — B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 0111 0011, C = 0

SLEEP — Go into standby mode, Clock oscillation stops

Syntax	SLEEP
Operands	-
Operation	-
Status Affected	TO,PD
OP-Code	00 0000 0000 0011
Description	Go into SLEEP mode with the oscillator stopped.
Cycle	1
Example	SLEEP — -

SUBWF — Subtract W from "f"

Syntax	SUBWF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	(W) ← (f) – (W)
Status Affected	C, DC, Z
OP-Code	00 0010 dfff ffff
Description	Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Cycle	1
Example	SUBWF REG1,1 — B : REG1 = 3, W = 2, C = ?, Z = ? A : REG1 = 1, W = 2, C = 1, Z = 0 SUBWF REG1,1 — B : REG1 = 2, W = 2, C = ?, Z = ? A : REG1 = 0, W = 2, C = 1, Z = 1 SUBWF REG1,1 — B : REG1 = 1, W = 2, C = ?, Z = ? A : REG1 = FFh, W = 2, C = 0, Z = 0

SWAPF	Swap Nibbles in "f"	
Syntax	SWAPF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination,7~4) ← (f.3~0), (destination.3~0) ← (f.7~4)	
Status Affected	-	
OP-Code	00 1110 dfff ffff	
Description	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.	
Cycle	1	
Example	SWAPF REG, 0	B : REG1 = 0xA5 A : REG1 = 0xA5, W = 0x5A

TESTZ	Test if "f" is zero	
Syntax	TESTZ f	
Operands	f : 00h ~ 7Fh	
Operation	Set Z flag if (f) is 0	
Status Affected	Z	
OP-Code	00 1000 1fff ffff	
Description	If the content of register 'f' is 0, Zero flag is set to 1.	
Cycle	1	
Example	TESTZ REG1	B : REG1 = 0, Z = ? A : REG1 = 0, Z = 1

XORLW	Exclusive OR Literal with W	
Syntax	XORLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) XOR k	
Status Affected	Z	
OP-Code	01 1111 kkkk kkkk	
Description	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.	
Cycle	1	
Example	XORLW 0xAF	B : W = 0xB5 A : W = 0x1A

XORWF	Exclusive OR W with "f"	
Syntax	XORWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (W) XOR (f)	
Status Affected	Z	
OP-Code	00 0110 dfff ffff	
Description	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	XORWF REG 1	B : REG = 0xAF, W = 0xB5 A : REG = 0x1A, W = 0xB5

Electrical Characteristics

1. Absolute Maximum Ratings (T_A=25 ℃)

Parameter	Rating	Unit
Supply voltage	V_{SS} -0.3 to V_{SS} +3.6	V
Input voltage	V_{SS} -0.3 to V_{DD} +0.3	
Output voltage	V_{SS} -0.3 to V_{DD} +0.3	
Output current high per 1 PIN	-18	mA
Output current high per all PIN	-60	
Output current low per 1 PIN	+20	
Output current low per all PIN	+100	
Maximum Operating Voltage	3.6	V
Operating temperature	-40 to +85	℃
Storage temperature	-65 to +150	

2. DC Characteristics (T_A=25℃, V_{DD}=3.0V, unless otherwise specified)

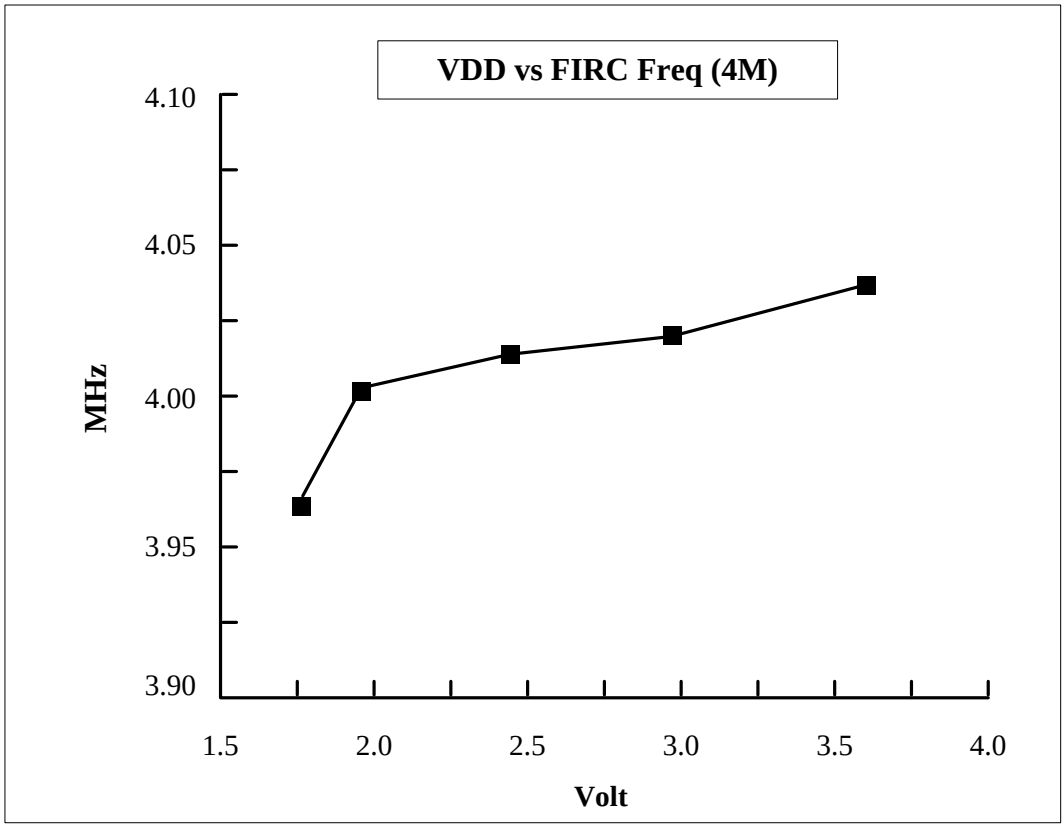
Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Input High Voltage	V_{IH}	All Input	V_{DD}=3.0V	0.8 V_{DD}	–	V_{DD}	V
Input Low Voltage	V_{IL}	All Input	V_{DD}=3.0V	0	–	0.2V_{DD}	V
Output High Voltage (NOTE 1)	V_{OH}	All Output	V_{DD}=3.0V	V_{DD}- 0.7	–	–	V
Output Low Voltage (NOTE 2)	V_{OL}	All Output	V_{DD}=3.0V	–	–	0.5	V
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{IN}=V_{DD}	–	–	1	uA
Input Leakage Current (pin low)	I_{ILL}	All Input	V_{IN}=0 V	–	–	-1	uA
Output Leakage Current (pin high)	I_{OLH}	All Output	V_{OUT}=V_{DD}	–	–	2	uA
Output Leakage Current (pin low)	I_{OLL}	All Output	V_{OUT}=0 V	–	–	-2	uA
Power Supply Current	I_{DD}	Run 12 MHz	V_{DD}=3.0V	–	4	–	mA
		Run 4 MHz	V_{DD}=2.0 V		1.5	–	
		Idle mode (32K enable)	V_{DD}=3.0 V (LCD ON)	–	–	25	uA
			V_{DD}=3.0 V (LCD OFF)	–	–	3	uA
		Stop mode	V_{DD}=3.0V (LCD ON)	–	16	–	uA
			V_{DD}=3.0V (LCD OFF)		0.1	1	
		Slow mode	V_{DD}=3.0V	–	18	–	
Pull-Up Resistor	R_P	V_{IN} = 0 V Ports A	V_{DD}=3.0V	25	50	100	kΩ
LCD Voltage Divider Resistor	R_{LCD}	–	–	103	–	513	kΩ

NOTE:

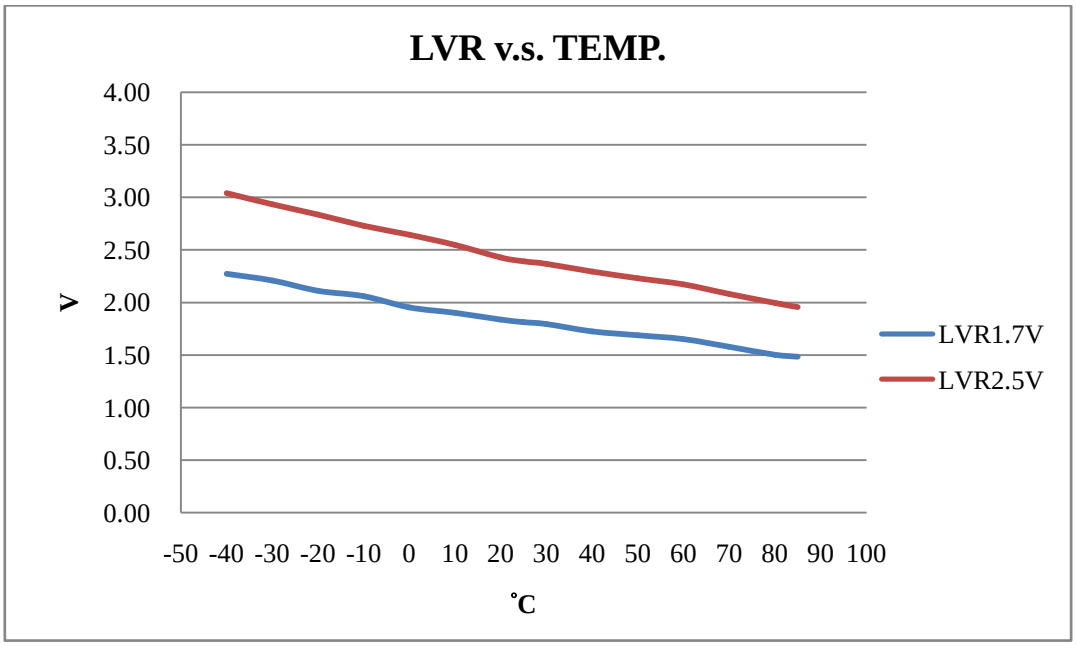
1. Output current high= -10 mA, while strong MP drives
2. Output current Low=20 mA

Characteristics Graphs

V_{DD} Voltage vs. Frequency of FIRC



LVR vs. Temperature

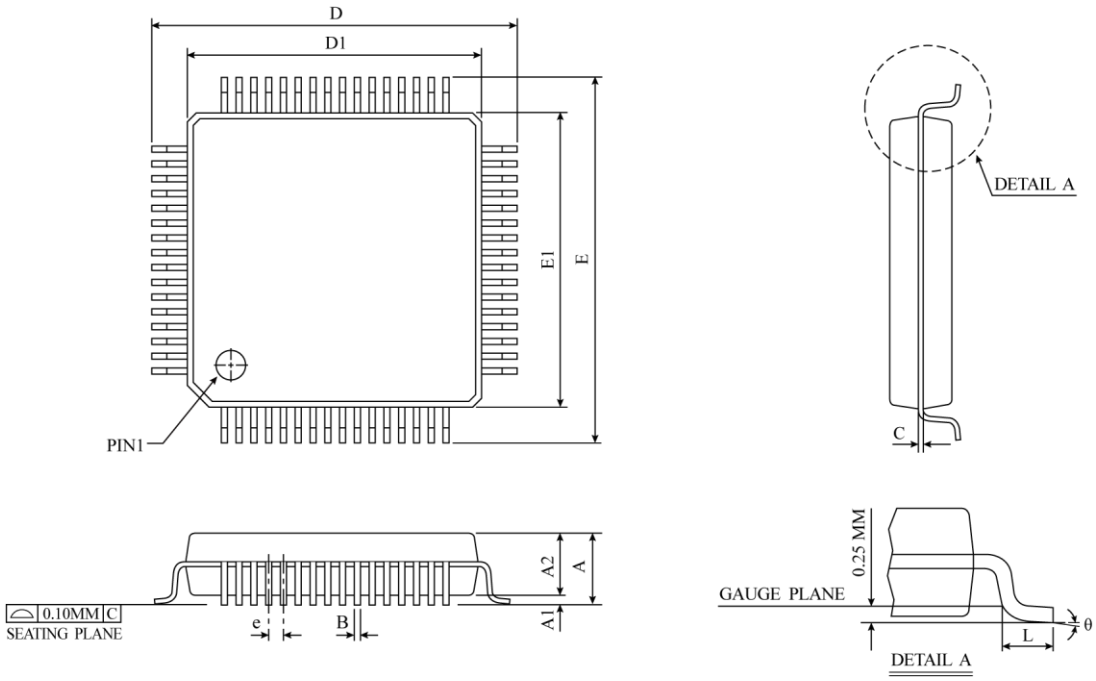


Ordering Information

The ordering information:

Ordering number	Package
TM57ML40-MTP	Wafer / Dice blank chip
TM57ML40-COD	Wafer / Dice with code
TM57ML40-MTP-76	LQFP 64-pin (10×10mm)

● LQFP-64 (10×10mm) Package Dimension



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	1.60	-	-	0.063
A1	0.05	0.10	0.15	0.002	0.004	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.17	0.20	0.23	0.007	0.008	0.009
C	0.09	0.13	0.16	0.004	0.005	0.006
D	12.00 BASIC			0.472 BASIC		
D1	10.00 BASIC			0.394 BASIC		
E	12.00 BASIC			0.472 BASIC		
E1	10.00 BASIC			0.394 BASIC		
e	0.50 BASIC			0.020 BASIC		
L	0.45	0.60	0.75	0.018	0.024	0.030
θ	0°	3.5°	7°	0°	3.5°	7°
JEDEC	MS-026 (BCD)					

△ * NOTES : DIMENSION "D1" AND "E1" DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSIONS IS 0.25mm PER SIDE. "D1" AND "E1" ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMACH.