# TM59PA80

## User's Manual

**tenx technology, inc.**

# Contents

# 1. Overview

## 1-1. FEATURES

**1. Memory**
- 212-byte general purpose register include LCD-Buffer (RAM)
- 8Kx14 internal program memory (OTP ROM)

**2. Oscillation Sources**
- Crystal, Ceramic, Ext RC, SUB Clock
- CPU clock divider (1, 1/4, 1/8, 1/16)

**3. Instruction Set**
- 35 instructions

**4. Instruction Execution Time**
- 167 ns at 12 MHz $f_{SYS}$ (minimum)

**5. Interrupts**
- 15 interrupt sources with one vector / one level

**6. I/O Ports**
- Total 36 bit-programmable pins (44QFP)
- Total 34 bit-programmable pins (42SDIP)

**7. Timers**
- Two 8-bit timer/counter. Interval mode. (Timer0, 1)
  Configurable as one 16-bit timer/counter.
- One Real-time and interval time measurement (Timer2)

**8. PWM**
- 8-bit PWM 1-ch (6-bit base + 2-bit extension)

**9. Watchdog Timer**
- $2^{21}$ oscillator's clock reset period

**10. Buzzer Out**
- Frequency Selectable Buzzer Output

**11. LCD Controller/Driver**
- 8 COM X 16 SEG
- 4 COM X 20 SEG
- 3 COM X 20 SEG

**12. A/D Converter**

- 6 analog input channels
- 12.5μs conversion speed at 4MHz $f_{ADC}$ clock

**13. 8-bit Serial I/O Interface**

- 8-bit transmit/receive mode
- 8-bit receive mode
- LSB-first or MSB-first transmission selectable
- Internal or external clock source

**14. Low Voltage Detector (LVD)**

- Voltage level detection
- Low Voltage Check to make system reset(LVR)
- $V_{LVD}$ = 2.3V/3.0V/3.9V

**15. Operating Temperature Range**

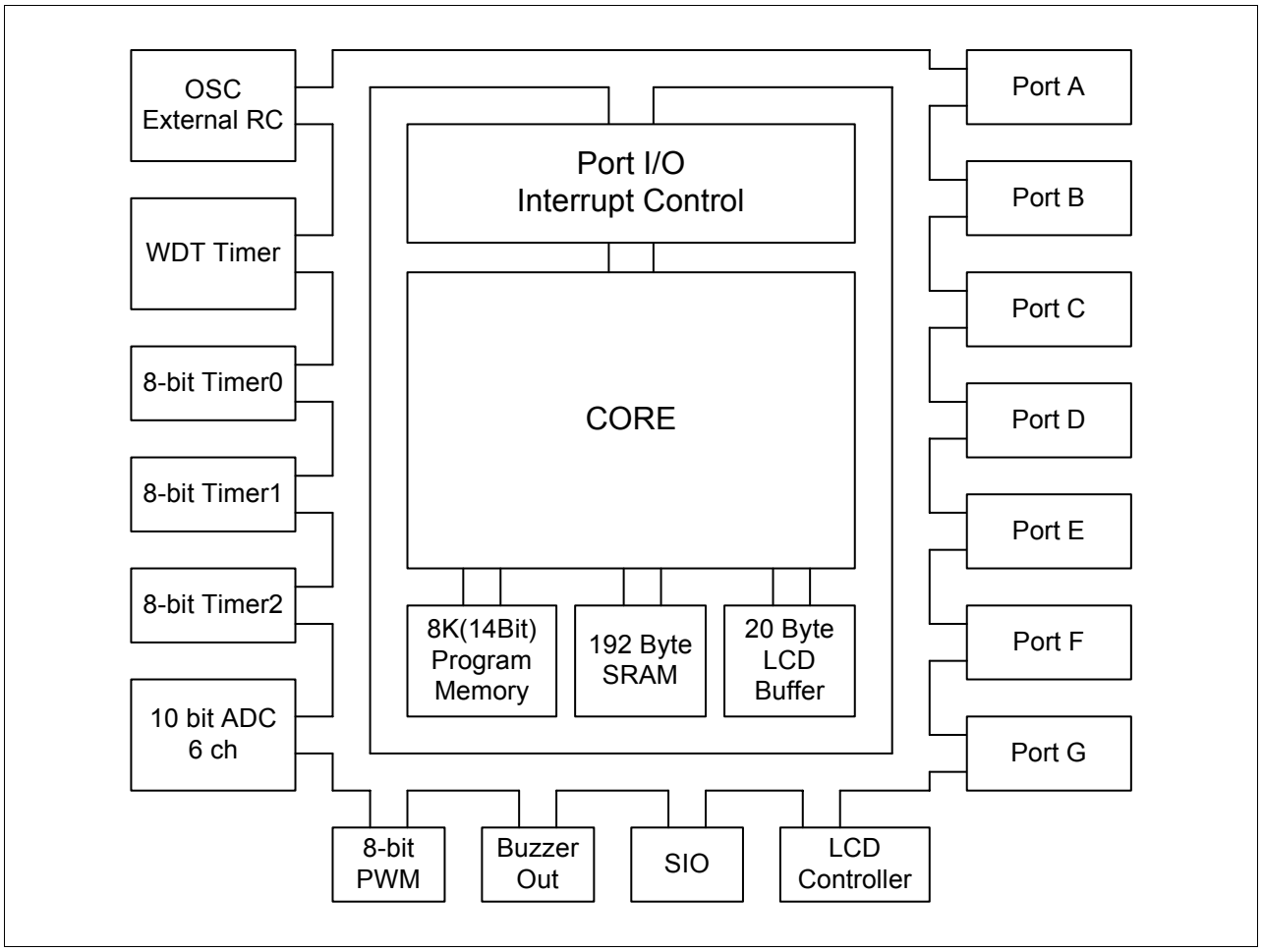
- **-40°C** to + 85°C

**16. Operating Voltage Range**

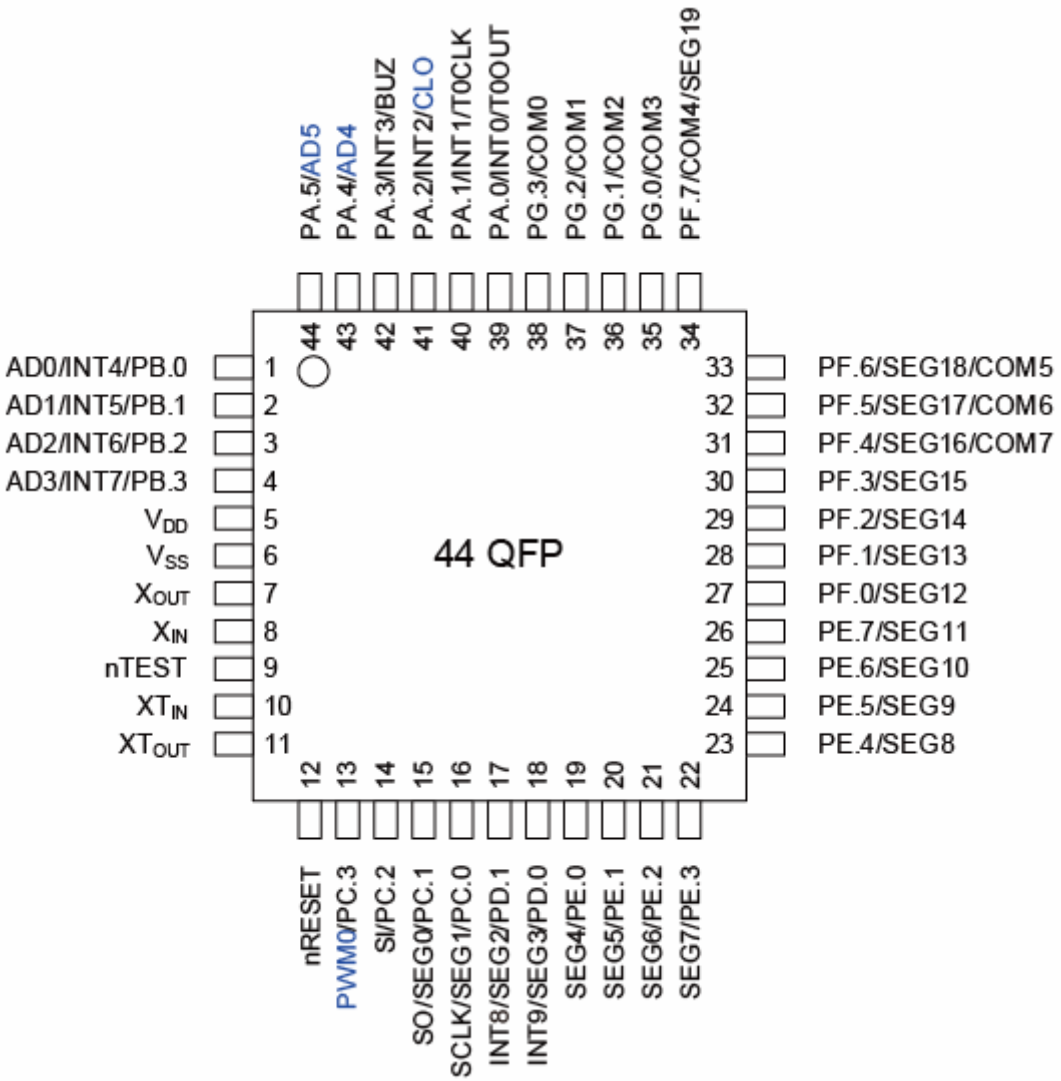
- 2.0 V to 5.5 V at 0.4 ~ 4.2 MHz
- 2.5 V to 5.5 V at 0.4 ~ 12 MHz

**17. Package Type**

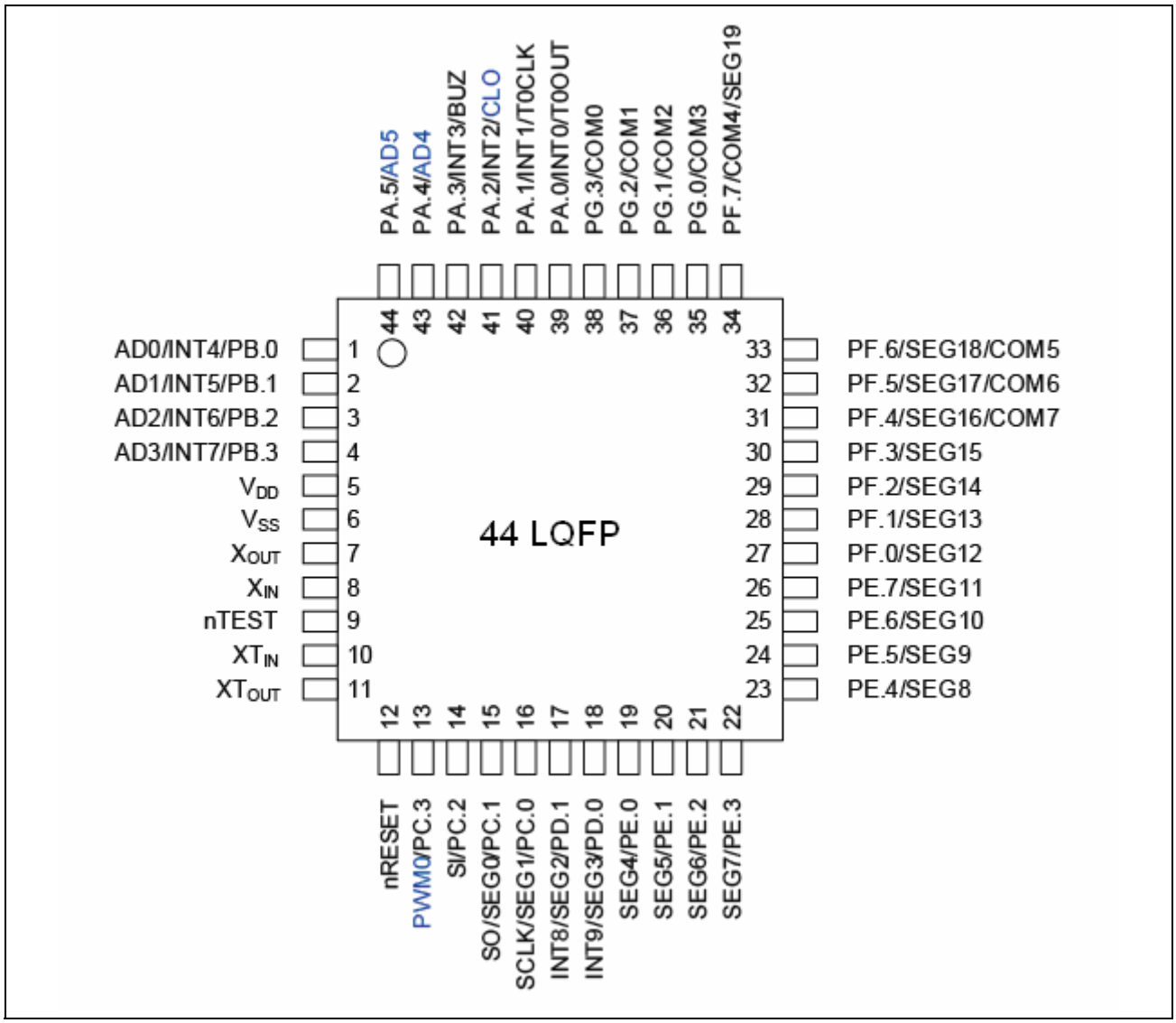
- 44-pin QFP
- 44-pin LQFP
- 42-pin SDIP
- 40-pin DIP



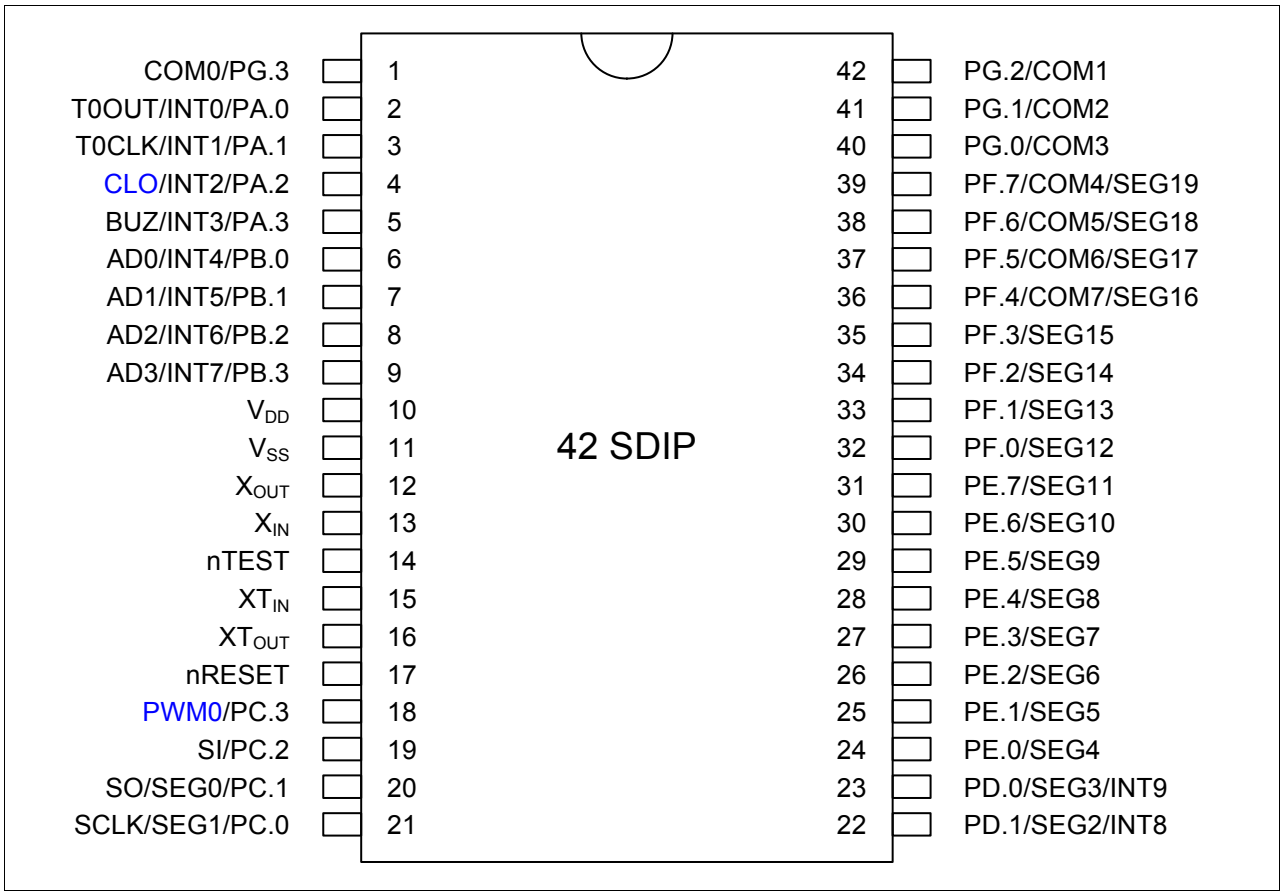

< Figure 1-1. Block Diagram >



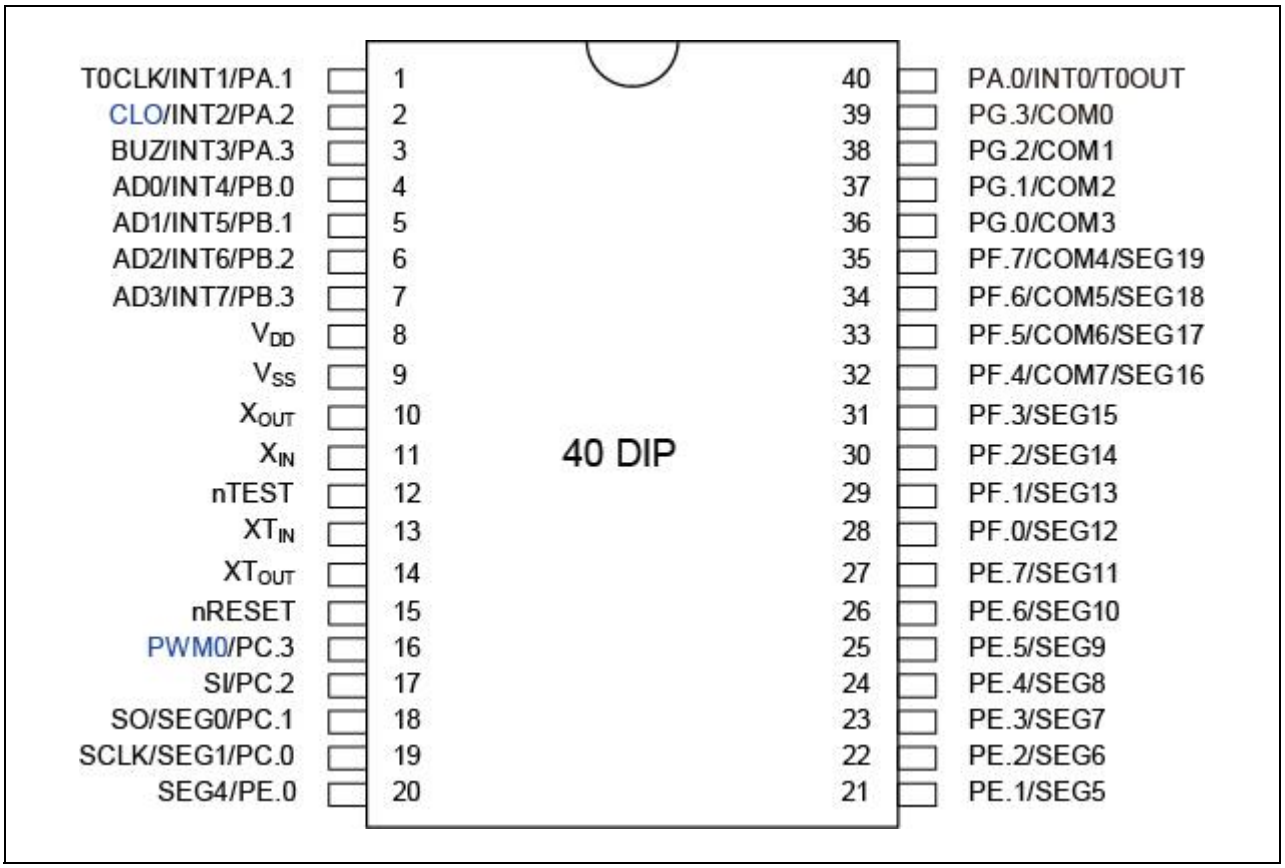

< Figure 1-2. Pin Assignment Diagram (44-Pin QFP Package) >




< Figure 1-3. Pin Assignment Diagram (44-Pin LQFP Package) >




< Figure 1-4. Pin Assignment Diagram (42-Pin SDIP Package) >




< Figure 1-5. Pin Assignment Diagram (40-Pin DIP Package) >

| Pin Name | I/O | Pin Description | Shared Function |
|---|---|---|---|
| $V_{DD}$, $V_{SS}$ | P | Power input pins for internal power block | - |
| $X_{IN}$, $X_{OUT}$ | – | Main oscillator pins for main clock | - |
| $XT_{IN}$, $XT_{OUT}$ | – | Sub oscillator pins for sub clock | - |
| nTEST | – | Chip test input pin. Hold $V_{DD}$ when the device is operating | - |
| nRESET | – | Reset signal input pin. Schmitt trigger input with internal pull-up resistor. | - |
| PA.0—PA.5 | I/O | Schmitt trigger input, Push-pull Output, Open-Drain Output, External Interrupt 0-3, Timer 0 Output, Timer 0 external clock input, Clock Output, Buzzer Out, ADC4-5 | INT0-3, ADC4-5, T0CLK, T0OUT, Buzzer, CLO |
| PB.0—PB.3 | I/O | Schmitt trigger input, Push-pull Output, Open-Drain Output, ADC0-3, External Interrupt 4-7 | ADC0-3, INT4-7 |
| PC.0—PC.3 | I/O | Schmitt trigger input, Push-pull output, Open-Drain output, SEG0-1, SCLK, SO, SI, PWM0 | SCLK, SO, SEG0-1, SI, PWM0 |
| PD.0, PD.1 | I/O | Schmitt trigger input, Push-pull output, Open-Drain output, External Interrupt 8-9, SEG2-3 | INT8-9, SEG2-3 |
| PE.0—PE.7 | I/O | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG4-11, Input mode with pull-up | SEG4-11 |
| PF.0—PF.7 | I/O | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG12-19, COM4-7, Input mode with pull-up | SEG12-19, COM4-7 |
| PG.0—PG.3 | I/O | Schmitt trigger Input, Push-pull output, Open-Drain output, COM0-3, Input mode with pull-up | COM0-3 |
| CLO | O | System clock output port | PA.2 |
| INT0-INT9 | I | External interrupt input port | PA.0-3, PB.0-3, PD.0-1 |
| PWM0 | O | 8-Bit high speed PWM output | PC.3 |
| T0OUT | O | Timer0 match output | PA.0 |
| ADC0-ADC5 | I | A/D converter input | PB.0-3, PA.4-5 |

<Table 1-1. PIN Description> < I: Input; O: Output; I/O: Bi-direction; P: Power >

## 1-2. Clock Scheme and Instruction Cycle

The TM59PA80 has two oscillator circuits, main clock and sub clock circuit. The notation of clock source is as following:
$f_{OSC}$ : Main Oscillator clock ($X_{IN}$, $X_{OUT}$)
$f_{SUB}$ : Sub Oscillator clock ($XT_{IN}$, $XT_{OUT}$)
$f_{SYS}$ : System Clock which is selected by OSCCON ($f_{OSC}$ or $f_{SUB}$)
$f_{CPU}$ : CPU Clock. Divided $f_{SYS}$ by CLKCON (/1, /4, /8, /16)

### 1. OSCILLATOR CONTROL REGISTER (OSCCON)

The oscillator control register (OSCCON) has the following functions:
— System clock selection and check clock switching status
— Main oscillator start/stop control
— Sub oscillator start/stop control
— Idle mode control

OSCCON.1 register settings select Main clock or Sub clock as system clock. After a reset, Main clock is selected for system clock and it can be switched to Sub clock by user program. System Clock switching is described in section 'Switching CPU Clock'. The main oscillator can be stopped or run by setting OSCCON.3 and the sub oscillator can be stopped or run by setting OSCCON.2.
*Idle mode control is described in section in 1.8 Power-Down mode.*

### 2. SYSTEM CLOCK CONTROL REGISTER (CLKCON)

The system clock control register (CLKCON) has the following functions:
Oscillator frequency divider

After a reset, the main oscillator is activated, and the $f_{SYS}/16$ (the slowest clock speed) is selected as the CPU clock. If necessary, you can then increase the CPU clock speed to $f_{SYS}/1$, $f_{SYS}/4$, or $f_{SYS}/8$ by setting the CLKCON.

### 3. SWITCHING THE CPU CLOCK

The oscillator control register (OSCCON) determine whether a main or a sub clock is selected as the CPU clock, and also how this frequency is to be divided by setting CLKCON. This makes it possible to switch dynamically between main and sub clocks and to modify operating frequencies.

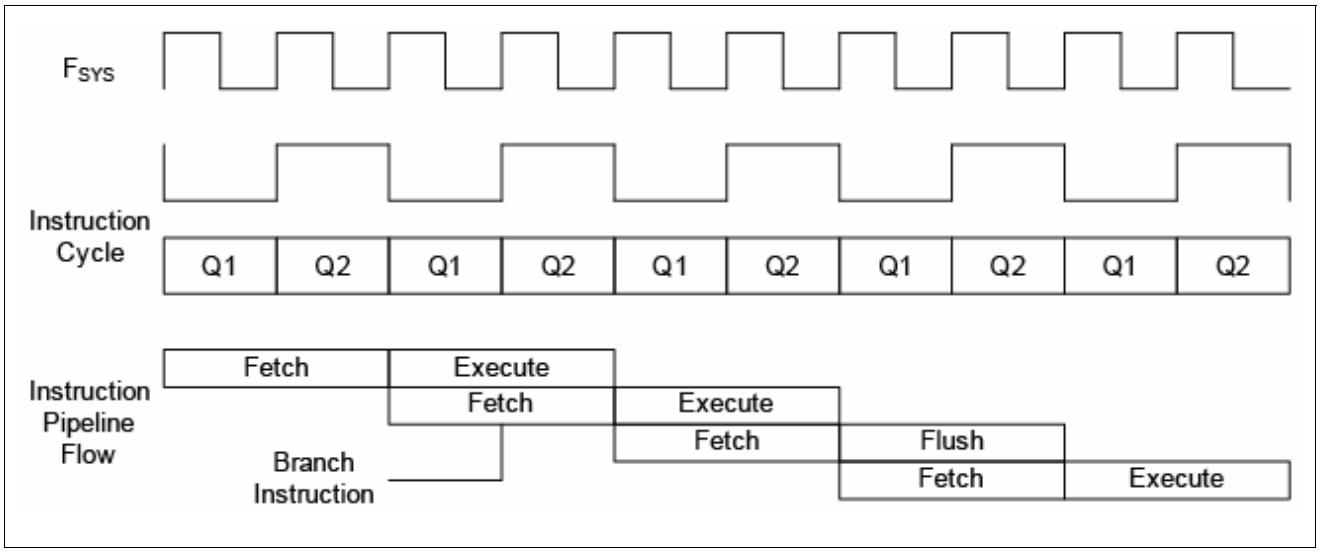
OSCCON.1 selects the main clock ($f_{OSC}$) or the sub clock ($f_{SUB}$) for the system clock. OSCCON.3 control main clock oscillation, and OSCCON.2 control sub clock oscillation. OSCCON.4 shows clock switching status.

For example, you are using the default system clock ($f_{OSC}$) and you want to switch from the main clock ($f_{OSC}$) to a sub clock ($f_{SUB}$) and to stop the main clock. To do this, the following steps must be taken to switch from a sub clock to the main clock.

**Step-1**: If sub clock is disabled, set OSCCON.2 = 0 to enable sub clock and wait oscillation stabilization time.
**Step-2**: Set OSCCON.1 = 1 to select sub clock ($f_{SUB}$) as system clock ($f_{SYS}$).
**Step-3**: Check OSCCON.4 is set to '1' (switched to sub clock)
**Step-4**: If you want to stop main clock, set OSCCON.3 = 1.

### 4. INSTRUCTION CYCLE

The CPU clock input 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.




< Figure 1-6. Clock/Instruction cycle and pipeline >

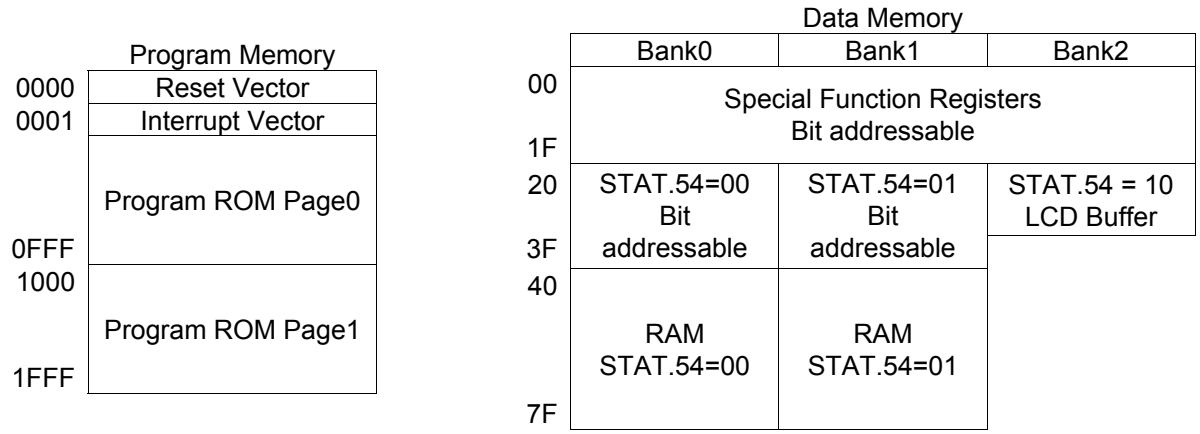
Branch instructions take two cycles since the fetch instruction is 'flushed' from the pipeline, while the new instruction is being fetched and then executed.

## 1-3. Addressing Mode

The Programming Counter is 13-bit wide capable of addressing a 8K 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 Vector (001h) are provided for PC initialization and Interrupt. For CALL/GOTO instructions, PC loads its lower 12 bits from instruction word and the MSB from STATUS's bit 6. 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 13-bit wide and 6-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.

The data memory is partitioned into three banks, which contain the General Purpose Data Memory, LCD Data Buffer and the Special Function Registers (SFR). STAT.5-4 is the bank select bits. Each bank extends up to 7Fh (128 bytes). The lower locations of each bank (00h-1Fh) are reserved for the SFR. Above the SFR are General Purpose Data Memory, implemented as static RAM. Some SFR area is mirrored in all banks for code reduction and quicker access. The first half of RAM (00h − 3Fh) is bit-addressable.

Data memory 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). Reading INDF itself indirectly (FSR=0) will produce 00h. Writing to the INDF register indirectly results in a no-operation.




< Figure 1-7. Address space >

## 1-4. ALU and Working (W) Register

The ALU is 8 bits 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.

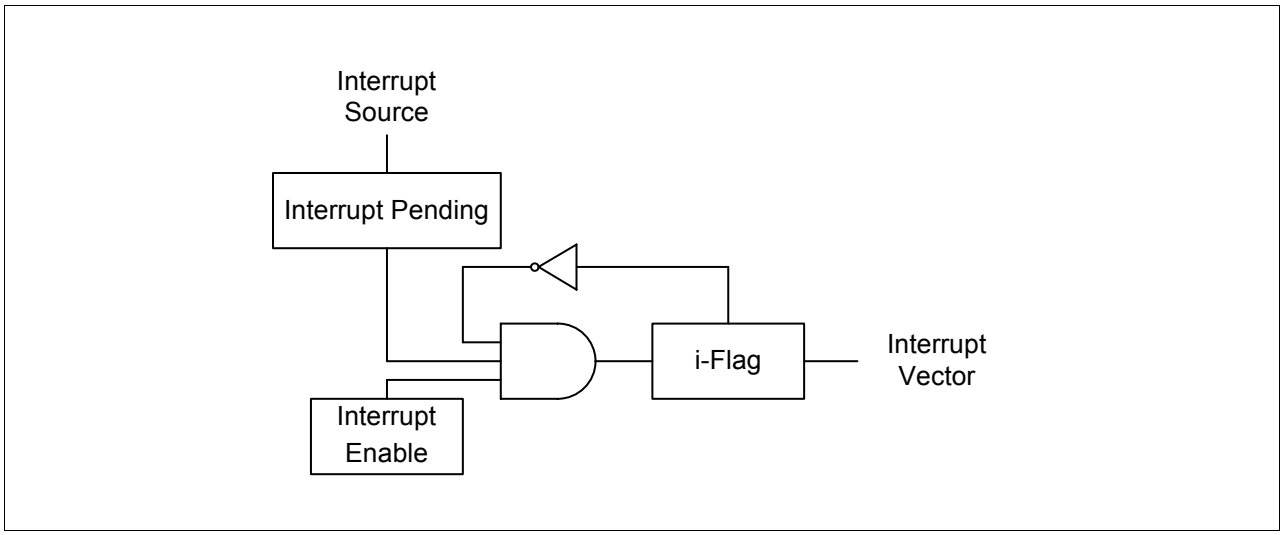
Depending on the instruction executed, the ALU may affect the values of Carry(C), Digit Carry (DC), and Zero (Z) Flags in the STAT register. The C and DC flags operate as a /Borrow and /Digit Borrow, respectively, in subtraction.

## 1-5. STATUS Register

This register contains the arithmetic status of ALU, the page select for Program ROM and RAM and global interrupt enable/disable control. 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.

## 1-6. Interrupt

The TM59PA80 has 1 level, 1 vector and 15 sources. Each interrupt source has its own enable control bit. An interrupt event will set its individual flag. Because TM59PA80 has only 1 vector, there is not a interrupt priority register. The interrupt priority is determined by F/W.



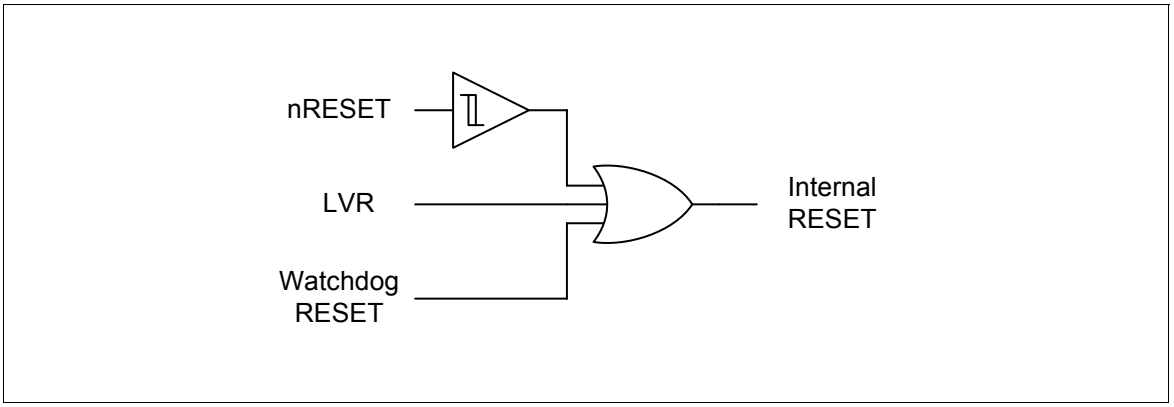

< Figure 1-8. Interrupt Function Diagram >

If the corresponding interrupt enable bit has been set, 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 0001” instruction is inserted to CPU, and the 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 edge trigged. F/W must clear the interrupt event register while serves the interrupt routine.

## 1-7. Reset

The TM59PA80 can be reset in four ways:

- Power-On-Reset
- Hardware Reset by nRESET pin
- Low Voltage Reset (LVR)
- Watchdog Reset




< Figure 1-9. Reset Circuit Diagram >

The nRESET pin must be held to Low level for a minimum time interval after the power supply comes within tolerance in order to allow time for internal CPU clock oscillation to stabilize. The minimum required oscillation stabilization time for a reset is approximately 2.5 ms ($f_{OSC}$ = 10 MHz). When the CPU is operating in normal state ($V_{DD}$ and nRESET at High level), if the signal at the nRESET pin is forced Low then the reset operation starts. All system and peripheral control registers are then set to their default hardware reset values.

The Low Voltage Reset features static reset when supply voltage is below a reference value. LVDCON is used to select reference voltage, LVD enable control, reset enable control and check voltage level status.

The Watchdog Timer is disabled after reset. F/W can use the CLRWDT instruction to clear and enable the Watchdog Timer. If once enabled, the Watchdog Timer overflow and generate a chip reset signal if no CLRWDT executed in a period of $2^{21}$ oscillator's cycle (256 msec for 8.192MHz crystal). The Watchdog Timer does not work in Power-down mode to provide wake-up function. It is only designed to prevent F/W goes into endless loop.

## 1-8. Power-Down Mode

The TM59PA80 has two kind of Power-down mode: STOP mode and IDLE mode. The STOP mode is activated by SLEEP instruction. During the STOP mode, the selected system clock ($f_{SYS}$) oscillation stops to minimize power consumption and all the peripherals which the same oscillator is selected as clock source are not working. Therefore, The Power down mode can be terminated by reset, enabled external Interrupts and timers which the other oscillator is selected as a clock source. When the Power down mode is released, the clock circuit requires oscillation stabilization time also. The STOPCON register must be set to '10100101b' before enter STOP mode. If the STOPCON is not set to '10100101b', the SLEEP instruction cause system reset. Idle mode is activated by OSCCON.0. During the Idle mode, only the internal CPU clock is stops. Therefore, Idle mode can be terminated by reset and all of interrupt.

## 1-9. Instruction Set

Each instruction is a 14-bit word divided into an OPCODE, which specified 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" represents address designator and "d" represents destination designator. The address designator is used to specify which address in Program memory is to be 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 | Description |
|---|---|
| f | 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 | Aero 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 |
| ← | Assign direction |

< List 1-1 OP-CODE Field Description >

| Mnemonic | | Op Code | Cycles | Flag Affect | Description |
|---|---|---|---|---|---|
| **Byte-Oriented File Register Instruction** | | | | | |
| ADDWF | f,d | 00 0111 dfff ffff | 1 | C,DC,Z | Add W and "f" |
| ANDWF | f,d | 00 0101 dfff ffff | 1 | Z | AND W with "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 | Complement "f" |
| DECF | f,d | 00 0011 dfff ffff | 1 | Z | Decrement "f" |
| DECFSZ | f,d | 00 1011 dfff ffff | 1 or 2 | - | Decrement "f", skip if zero |
| INCF | f,d | 00 1010 dfff ffff | 1 | Z | Increment "f" |
| INCFSZ | f,d | 00 1111 dfff ffff | 1 or 2 | - | Increment "f", skip if zero |
| IORWF | f,d | 00 0100 dfff ffff | 1 | Z | OR W with "f" |
| MOVFW | f | 00 1000 0fff ffff | 1 | - | Move "f" to "w" |
| MOVWF | f | 00 0000 1fff ffff | 1 | - | Move W to "f" |
| RLF | f,d | 00 1101 dfff ffff | 1 | C | Rotate left "f" through carry |
| RRF | f,d | 00 1100 dfff ffff | 1 | C | Rotate right "f" through carry |
| SUBWF | f,d | 00 0010 dfff ffff | 1 | C,DC,Z | Subtract W from "f" |
| SWAPF | f,d | 00 1110 dfff ffff | 1 | - | Swap nibble of "f" |
| TESTZ | f | 00 1000 1fff ffff | 1 | Z | Test if "f" is zero |
| XORWF | f,d | 00 0110 dfff ffff | 1 | Z | XOR W with "f" |
| **Bit-Oriented File Register Instruction** | | | | | |
| BCF | f,b | 01 000b bbff ffff | 1 | - | Clear "b" bit of "f" |
| BSF | f,b | 01 001b bbff ffff | 1 | - | Set "b" bit of "f" |
| BTFSC | f,b | 01 010b bbff ffff | 1 or 2 | - | Test "b" bit of "f", skip if clear |
| BTFSS | f,b | 01 011b bbff ffff | 1 or 2 | - | Test "b" bit of "f", skip if set |
| **Literal and Control Instruction** | | | | | |
| ADDLW | k | 01 1100 kkkk kkkk | 1 | C,DC,Z | Add Literal "k" and W |
| ANDLW | k | 01 1011 kkkk kkkk | 1 | Z | AND Literal "k" with W |
| CALL | k | 10 kkkk kkkk kkkk | 2 | - | Call subroutine "k" |
| CLRWDT | | 00 0000 1000 1001 | 1 | - | Clear and enable Watch Dog Timer |
| GOTO | k | 11 kkkk kkkk kkkk | 2 | - | Jump to branch "k" |
| IORLW | k | 01 1010 kkkk kkkk | 1 | Z | OR Literal "k" with W |
| MOVLW | k | 01 1001 kkkk kkkk | 1 | - | Move Literal "k" to W |
| NOP | | 00 0000 0000 0000 | 1 | - | No operation |
| RET | | 00 0000 0100 0000 | 2 | - | Return from Subroutine |
| RETI | | 00 0000 0110 0000 | 2 | - | Return from interrupt |
| RETLW | k | 01 1000 kkkk kkkk | 2 | - | Return with Literal "k" in W |
| SLEEP | | 00 0000 1000 1010 | 1 | - | Enter STOP mode, Clock oscillation stops |
| XORLW | k | 01 1111 kkkk kkkk | 1 | Z | XOR Literal "k" with W |

< List 1-2 Instruction Summary >

## 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<br>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<br>A : W = 0xD9, FSR = 0xC2 |

## ANDLW Logical AND Literal "k" with W

| | |
|---|---|
| Syntax | **ANDLW k** |
| Operands | k : 00h ~ FFh |
| Operation | (W) ← (W) ‘AND’ (f) |
| 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<br>A : W = 0x03 |

## ANDWF AND W with f

| | |
|---|---|
| Syntax | **AND 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<br>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<br>A : FLAG_REG = 0x47 |

## BSF Set "b" bit of "f"

| | |
|---|---|
| Syntax | **BCF 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<br>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 NO P is executed instead, making this a 2nd cycle instruction. |
| Cycle | 1 or 2 |
| Example | LABEL1 BTFSC FLAG, 1 — B : PC = LABEL1<br>TRUE GOTO SUB1 — A : if FLAG.1 = 0, PC = FALSE<br>FALSE ... — if FLAG.1 = 1, PC = TRUE |

## BTFSS — Test "b" bit of "f", skip if set(1)

| | |
|---|---|
| Syntax | **BTFSC 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.<br>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 — B : PC = LABEL1<br>TRUE GOTO SUB1 — A : if FLAG.1 = 0, PC = TRUE<br>FALSE ... — 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 eleven-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<br>A : PC = SUB1,<br>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<br>B : FLAG_REG = 0x5A<br>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<br>B : W = 0x5A<br>A : W = 0x00, Z = 1 |

## CLRWDT Clear Watchdog Timer

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

## COMF — Complement f

| | |
|---|---|
| Syntax | **COMF f [,d]** |
| Operands | f : 00h ~ 7Fh, d : 0, 1 |
| Operation | (destination) ← (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<br>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<br>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<br>GOTO LOOP<br>CONTINUE<br>B : PC = LABEL1<br>A : CNT = CNT + 1<br>if CNT=0, PC = CONTINUE<br>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<br>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 k |
| 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<br>A : RESULT = 0x13,<br>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 = ?<br>A : W ← f |

## 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 as semble as 0's. |
| Cycle | 1 |
| Example | MOVLW 0x5A<br>B : W = ?<br>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<br>B : REG1 = 0xFF, W = 0x4F<br>A : REG1 = 0x4F, W = 0x4F |

## NOP No Operation

| | |
|---|---|
| Syntax | **NOP** |
| Operands | - |
| Operation | No Operation |
| Status Affected | Z |
| OP-Code | **00 0000 0000 0000** |
| Description | No Operation |
| Cycle | 1 |
| Example | NOP<br>- |

## 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 ~ FFFh |
| 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 |

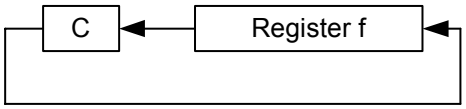
Example

```
        CALL TABLE      ; B : W = 0x07
              :         ; A : W = value of k8
TABLE   ADDWF PCL
        RETLW k1
        RETLW k2
              :
        RETLW kn
```

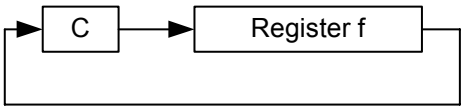
## RET Return from Subroutine

| | |
|---|---|
| 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<br>A : REG1 = 1110 0110<br>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<br>A : REG1 = 1110 0110<br>W = 0111 0011, C = 0 |

## SLEEP — Go into standby mode, Clock oscillation stops

| | |
|---|---|
| Syntax | **SLEEP** |
| Operands | - |
| Operation | - |
| Status Affected | - |
| OP-Code | **00 0000 0000 0011** |
| Description | The processor is put 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 | **(Destination) ← (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 = ?<br>A : REG1 = 1, W = 2, C = 1, Z = 0 |
| | SUBWF REG1,1 — B : REG1 = 2, W = 2, C = ?, Z = ?<br>A : REG1 = 0, W = 2, C = 1, Z = 1 |
| | SUBWF REG1,1 — B : REG1 = 1, W = 2, C = ?, Z = ?<br>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<br>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 contents of register 'f' is 0, Zero flag is set to 1. |
| Cycle | 1 |
| Example | TESTZ REG1 — B : REG1 = 0, Z = ?<br>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<br>B : W = 0xB5<br>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<br>B : REG = 0xAF, W = 0xB5<br>A : REG = 0x1A, W = 0xB5 |

# 2. Control Registers

| | Bank0 | Bank1 | Bank2 |
|---|---|---|---|
| **00** | INDF | | |
| **01** | T0CNT | CLKCON | OSCCON |
| **02** | PCL | | |
| **03** | STAT | | |
| **04** | FSR | | |
| **05** | PAD | PACONL | PECONL |
| **06** | PBD | PACONH | PECONH |
| **07** | PCD | PAPU | PFCONL |
| **08** | PDD | PBCON | PFCONH |
| **09** | WDTE | | |
| **0A** | PWRDN | | |
| **0B** | PED | PBPU | PGCON |
| **0C** | PFD | PCCON | PWM0CON |
| **0D** | PGD | PCPU | PWM0DAT |
| **0E** | T0CON | PDCON | LCDCON |
| **0F** | T0DATA | PINTD0 | LPCON |
| **10** | T1CON | PINTD1 | LVDCON |
| **11** | T1DATA | PINTD2 | ADCDATL |
| **12** | T2CON | INTCON0 | ADCDATH |
| **13** | BZCON | INTCON1 | SIOPS |
| **14** | ADCCON | | |
| **15** | SIOCON | STOPCON | – |
| **16** | SIODAT | | |
| **17** | INTPND0 | | |
| **18** | INTPND1 | | |
| **19** | SYSTEM Use Only | | |
| **1A** | GPR0 | | |
| **1B** | GPR1 | | |
| **1C** | GPR2 | | |
| **1D** | GPR3 | | |
| **1E** | GPR4 | | |
| **1F** | GPR5 | | |

## ADCCON — A/D Converter Control Register 14H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | − | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | − | R/W | R/W | R/W | R | R/W | R/W | R/W | |

| Bit | Description | | | |
|---|---|---|---|---|
| 7 | **Not Used** (Must keep always '0') | | | |
| 6-4 | **Input Pin Selection Bits** | | | |
| | 0 | 0 | 0 | ADC0 (PB.0) |
| | 0 | 0 | 1 | ADC1 (PB.1) |
| | 0 | 1 | 0 | ADC2 (PB.2) |
| | 0 | 1 | 1 | ADC3 (PB.3) |
| | 1 | 0 | 0 | ADC4 (PA.4) |
| | 1 | 0 | 1 | ADC5 (PA.5) |
| | Others | | | Not Used |
| 3 | **End-of-Conversion Status Bit** | | | |
| | 0 | A/D conversion is in progress | | |
| | 1 | A/D conversion complete | | |
| 2-1 | **Clock Source Selection Bit** | | | |
| | 0 | 0 | $f_{SYS}$ /16 ($f_{SYS}$ ≤ 10 MHz) | |
| | 0 | 1 | $f_{SYS}$ / 8 ($f_{SYS}$ ≤ 10 MHz) | |
| | 1 | 0 | $f_{SYS}$ / 4 ($f_{SYS}$ ≤ 10 MHz) | |
| | 1 | 1 | $f_{SYS}$ / 1 ($f_{SYS}$ ≤ 4 MHz) | |
| 0 | **Conversion Start Bit** | | | |
| | 0 | Disable Operation | | |
| | 1 | A/D conversion start (Auto clear) | | |
| ※ Maximum ADC Input Clock is 4MHz. | | | | |

## ADCDATL — ADC Data Register Low Byte Bank2, 11H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | − | − | − | − | − | − | − | − | |
| R/W | − | − | − | − | − | − | R | R | |

| Bit | Description | |
|---|---|---|
| 7-0 | **PWM1 Period Data Low Byte** | |
| | XX | ADC Data Value Lower 2Bit |

## ADCDATH — ADC Data Register High Byte — Bank2, 12H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | – | – | – | – | |
| R/W | R | R | R | R | R | R | R | R | |

| Bit | Description | |
|---|---|---|
| 7-0 | **PWM1 Period Data Low Byte** | |
| | XXXXXXXX | ADC Data Value Higher 8Bit |

## BZCON — Buzzer Out Control Register — Bank0, 13H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Input Clock Selection** | | |
| | 0 | 0 | $f_{SYS}$ / 8 |
| | 0 | 1 | $f_{SYS}$ / 16 |
| | 1 | 0 | $f_{SYS}$ / 32 |
| | 1 | 1 | $f_{SYS}$ / 64 |
| 5-0 | **Buzzer Period Data** | | |
| | XXXXXX | Period Data | |

## CLKCON — Clock Control Register — Bank1, 01H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | – | – | 0 | 0 | |
| R/W | – | – | – | – | – | – | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-2 | **Not Used** | | |
| 1-0 | **Divided by Selection Bits for CPU Clock frequency ($F_{CPU}$)** | | |
| | 0 | 0 | $f_{SYS}$ /16 |
| | 0 | 1 | $f_{SYS}$ / 8 |
| | 1 | 0 | $f_{SYS}$ / 4 |
| | 1 | 1 | $f_{SYS}$ / 1 |

## FSR — File Select Register 04H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | – | – | – | – | |
| R/W | – | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7 | **Not Used** | |
| 6-0 | **File Select Register** | |
| | 000 0000 | Not Used. |
| | 1 ~ 7Fh | Indirect Addressing Location |

## GPR0−5 — General Purpose Register 1AH~1FH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | – | – | – | – | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-0 | **General Purpose Register** |
| | GPR0−5 are mirrored all bank. It is useful to pass arguments to SUB routine or backup Working register (W) and STAT register in ISR or SUB routine. |

## INTCON0 — Interrupt Control Register — Bank1, 12H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7 | **EXTINT7 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 6 | **EXTINT6 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 5 | **EXTINT5 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 4 | **EXTINT4 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 3 | **EXTINT3 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 2 | **EXTINT2 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 1 | **EXTINT1 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 0 | **EXTINT0 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |

## INTCON1 — Interrupt Control Register

**Bank1, 13H**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | – | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7 | **PWM0 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 6 | **Timer 2 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 5 | **Timer 1 Match Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 4 | **Timer 0 Match Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 3 | **Not Used** | |
| 2 | **SIO Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 1 | **EXTINT9 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |
| 0 | **EXTINT8 Interrupt Enable Bit** | |
| | 0 | Disable |
| | 1 | Enable |

## INTPND0 — External Interrupt Pending Register 17H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7 | **EXTINT7 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 6 | **EXTINT6 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 5 | **EXTINT5 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 4 | **EXTINT4 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 3 | **EXTINT3 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 2 | **EXTINT2 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 1 | **EXTINT1 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 0 | **EXTINT0 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |

## INTPND1 — External Interrupt Pending Register 18H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | − | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | − | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7 | **PWM0 Overflow Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 6 | **Timer 2 Match Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 5 | **Timer 1 Match Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 4 | **Timer 0 Match Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 3 | **Not Used** | |
| 2 | **SIO Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 1 | **EXTINT9 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |
| 0 | **EXTINT8 Interrupt Pending Bit** | |
| | 0 | No interrupt pending (read) / Pending bit clear (write) |
| | 1 | Interrupt is pending (read) / No effect (write) |

## LCDCON — LCD Control Register — Bank2, 0EH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | 0 | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-5 | **Not Used** (Must keep always '0') | | |
| 4 | **LCD Enable Bits** | | |
| | 0 | Display Off (Cut Off the LCD Voltage dividing resistors) | |
| | 1 | Display On | |
| 3-2 | **LCD Duty and Bias Selection Bit** | | |
| | 0 | 0 | 1/3 duty, 1/3 bias ( COM0−COM2, SEG0−SEG19 ) |
| | 0 | 1 | 1/4 duty, 1/3 bias ( COM0−COM3, SEG0−SEG19 ) |
| | 1 | 0 | 1/8 duty, 1/4 bias ( COM0−COM7, SEG0−SEG15 ) |
| | 1 | 1 | 1/8 duty, 1/5 bias ( COM0−COM7, SEG0−SEG15 ) |
| 1-0 | **LCD Clock Selection Bits** | | |
| | 0 | 0 | $f_{T2}$ /$2^7$ ( 256 Hz, $f_{T2}$ = 32.768 Hz) |
| | 0 | 1 | $f_{T2}$ /$2^6$ ( 512 Hz, $f_{T2}$ = 32.768 Hz) |
| | 1 | 0 | $f_{T2}$ /$2^5$ (1024 Hz, $f_{T2}$ = 32.768 Hz) |
| | 1 | 1 | $f_{T2}$ /$2^4$ (2048 Hz, $f_{T2}$ = 32.768 Hz) |

## LPCON — LCD Port Control Register — Bank2, 0FH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | − | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | − | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | | |
|---|---|---|---|---|
| 7 | **Not Used** | | | |
| 6-4 | **SEG4 ~ SEG19 / COM0 ~ COM7 Selection Bit** | | | |
| | 0 | 0 | 0 | PE.0 ~ PG.3: I/O |
| | 0 | 0 | 1 | PE.0 ~ PF.7: I/O PG.0 ~ PG.3: LCD Signal |
| | 0 | 1 | 0 | PE.0 ~ PF.3: I/O PF.4 ~ PG.3: LCD Signal |
| | 0 | 1 | 1 | PE.0 ~ PE.7: I/O PF.0 ~ PG.3: LCD Signal |
| | 1 | 0 | 0 | PE.0 ~ PE.3: I/O PE.4 ~ PG.3: LCD Signal |
| | 1 | 0 | 1 | PE.0 ~ PG.3: LCD Signal |
| | Others | | | Not Used |
| 3 | **SEG3 / PORTD.0 Selection Bit** | | | |
| | 0 | Normal I/O | | |
| | 1 | SEG Enable | | |
| 2 | **SEG2 / PORTD.1 Selection Bit** | | | |
| | 0 | Normal I/O | | |
| | 1 | SEG Enable | | |
| 1 | **SEG1 / PORTC.0 Selection Bit** | | | |
| | 0 | Normal I/O | | |
| | 1 | SEG Enable | | |
| 0 | **SEG0 / PORTC.1 Selection Bit** | | | |
| | 0 | Normal I/O | | |
| | 1 | SEG Enable | | |

## LVDCON — LVD Config Register — Bank2, 10H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7 | **Low Voltage Detector Enable Bit** | |
| | 0 | Disable Low Voltage Detector |
| | 1 | Enable Low Voltage Detector |
| 6 | **RESET Enable Bit** | |
| | 0 | Disable Reset |
| | 1 | Enable Reset |
| 5 | **Voltage Level Status Bit** | |
| | 0 | $V_{DD}$ is lower than reference voltage |
| | 1 | $V_{DD}$ is higher than reference voltage |
| 4-0 | **Reference Voltage Selection Bit** | |
| | 11011 | 2.3V |
| | 10000 | 3.0V |
| | 01110 | 3.9V |
| | Others | Not Used |

## OSCCON — Oscillator Control Register — Bank2, 01H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | 0 | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | R | R/W | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7-4 | **Not Used** | |
| 4 | **Clock Switching Status Bit** | |
| | 0 | Switched to Main oscillator |
| | 1 | Switched to Sub oscillator |
| 3 | **Main Oscillator Control Bit** | |
| | 0 | Main oscillator RUN |
| | 1 | Main oscillator STOP |
| 2 | **Sub Oscillator Control Bit** | |
| | 0 | Sub oscillator RUN |
| | 1 | Sub oscillator STOP |
| 1 | **System Clock Selection Bit** | |
| | 0 | Main oscillator select |
| | 1 | Sub oscillator select |
| 0 | **Idle Mode Control Bit** | |
| | 0 | No Effect |
| | 1 | Enter Idle Mode |

## PCL — Program Counter Low Byte 02H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-0 | **Program Counter Low Byte**<br>This register represents Lower 8-Bit of PC+1. The PC can be changed writing any value (00h~FFh) into this register. It is similar to GOTO instruction. But the branch instruction by PCL can access only higher address than PC. |

## PACONL — Port A Control Register (Low Byte) Bank1, 05H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | | | Description |
|---|---|---|---|
| 7-6 | **Port A.3 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT3 |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain output |
| | 1 | 1 | Buzzer Out |
| 5-4 | **Port A.2 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT2 |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain output |
| | 1 | 1 | Clock Output |
| 3-2 | **Port A.1 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT1 / T0CLK Input |
| | 0 | 1 | Push-pull output |
| | 1 | X | Open-drain output |
| 1-0 | **Port A.0 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT0 |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain output |
| | 1 | 1 | T0OUT |

## PACONH — Port A Control Register (High Byte) Bank1, 06H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-4 | **Not Used** | | |
| 3-2 | **Port A.5 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain output |
| | 1 | 1 | ADC5 |
| 1-0 | **Port A.4 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain output |
| | 1 | 1 | ADC4 |

## PAD — Port A Data Register Bank0, 05H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-6 | **Not Used** |
| 5-0 | **Port A.5-0 Data Bits** |

## PAPU — Port A Pull-Up Control Register — Bank1, 07H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7-6 | **Not Used** | |
| 5 | **Port A.5 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 4 | **Port A.4 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 3 | **Port A.3 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 2 | **Port A.2 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 1 | **Port A.1 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 0 | **Port A.0 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |

## PBCON — Port B Control Register — Bank1, 08H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Port B.3 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT7 |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | ADC3 |
| 5-4 | **Port B.2 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT6 |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | ADC2 |
| 3-2 | **Port B.1 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT5 |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | ADC1 |
| 1-0 | **Port B.0 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT4 |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | ADC0 |

## PBD — Port B Data Register — Bank0, 06H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-4 | **Not Used** |
| 3-0 | **Port B.3-0 Data Bits** |

## PBPU — Port B Pull-Up Control Register — Bank1, 0BH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W | |

| Bit | | Description |
|---|---|---|
| 7-4 | | **Not Used** |
| 3 | | **Port B.3 Pull-up Enable Bit** |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 2 | | **Port B.2 Pull-up Enable Bit** |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 1 | | **Port B.1 Pull-up Enable Bit** |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 0 | | **Port B.0 Pull-up Enable Bit** |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |

## PCCON — Port C Control Register — Bank1, 0CH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Port C.3 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | PWM Out |
| 5-4 | **Port C.2 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / SI |
| | 0 | 1 | Push-pull output |
| | 1 | X | Open-drain Output |
| 3-2 | **Port C.1 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | SO |
| 1-0 | **Port C.0 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / SCLK In |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | SCLK Out |

## PCD — Port C Data Register — Bank0, 07H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-4 | **Not Used** |
| 3-0 | **Port C.3-0 Data Bits** |

## PCPU — Port C Pull-Up Control Register — Bank1, 0DH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7-4 | **Not Used** | |
| 3 | **Port C.3 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 2 | **Port C.2 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 1 | **Port C.1 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |
| 0 | **Port C.0 Pull-up Enable Bit** | |
| | 0 | Pull-up Disable |
| | 1 | Pull-up Enable |

## PDCON — Port D Control Register — Bank1, 0EH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-4 | **Not Used** | | |
| 3-2 | **Port D.1 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT9 |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 1-0 | **Port D.0 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input / INT8 |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |

## PDD — Port D Data Register — Bank0, 08H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | – | – | 0 | 0 | |
| R/W | – | – | – | – | – | – | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-2 | **Not Used** |
| 1-0 | **Port D.1-0 Data Bits** |

## PECONL — Port E Control Register (Low Byte) — Bank2, 05H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Port E.3 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 5-4 | **Port E.2 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 3-2 | **Port E.1 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 1-0 | **Port E.0 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |

## PECONH — Port E Control Register (High Byte) Bank2, 06H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Port E.7 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 5-4 | **Port E.6 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 3-2 | **Port E.5 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 1-0 | **Port E.4 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |

## PED — Port E Data Register Bank0, 0BH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-0 | **Port E.7-0 Data Bits** |

## PFCONL — Port F Control Register (Low Byte) Bank2, 07H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Port F.3 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 5-4 | **Port F.2 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 3-2 | **Port F.1 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 1-0 | **Port F.0 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |

## PFCONH — Port F Control Register (High Byte) Bank2, 08H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Port F.7 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 5-4 | **Port F.6 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 3-2 | **Port F.5 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 1-0 | **Port F.4 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |

## PFD — Port F Data Register Bank0, 0CH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-0 | **Port F.7-0 Data Bits** |

## PGCON — Port G Control Register — Bank2, 0BH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Port G.3 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 5-4 | **Port G.2 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 3-2 | **Port G.1 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |
| 1-0 | **Port G.0 Configuration Bits** | | |
| | 0 | 0 | Schmitt-trigger Input |
| | 0 | 1 | Push-pull output |
| | 1 | 0 | Open-drain Output |
| | 1 | 1 | Input mode with Pull-Up |

## PGD — Port G Data Register — Bank0, 0DH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 3-0 | **Port G.3-0 Data Bits** |

## PINTD0 — External Interrupt Direction Control Register — Bank1, 0FH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Port A.3 EXTINT3 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |
| 5-4 | **Port A.2 EXTINT2 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |
| 3-2 | **Port A.1 EXTINT1 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |
| 1-0 | **Port A.0 EXTINT0 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |

## PINTD1 — External Interrupt Direction Control Register — Bank1, 10H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Port B.7 EXTINT7 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |
| 5-4 | **Port B.6 EXTINT6 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |
| 3-2 | **Port B.5 EXTINT5 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |
| 1-0 | **Port B.4 EXTINT4 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |

## PINTD2 — External Interrupt Direction Control Register — Bank1, 11H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | 0 | 0 | 0 | 0 | |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-4 | **Not Used** | | |
| 3-2 | **Port D.1 EXTINT9 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |
| 1-0 | **Port D.0 EXTINT8 Interrupt Signal Selection Bit** | | |
| | 0 | 0 | Falling Edge Interrupt Enable |
| | 0 | 1 | Rising Edge Interrupt Enable |
| | 1 | X | Rising/Falling Edge Interrupt Enable |

## PWM0CON — PWM0 Control Register — Bank2, 0CH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | 0 | 0 | – | 0 | 0 | 0 | |
| R/W | – | – | R/W | R/W | – | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-6 | **Not Used** | | |
| 5-4 | **PWM0 Input Clock Selection Bit** | | |
| | 0 | 0 | $f_{SYS}$ / 64 |
| | 0 | 1 | $f_{SYS}$ / 8 |
| | 1 | 0 | $f_{SYS}$ / 2 |
| | 1 | 1 | $f_{SYS}$ / 1 |
| 3 | **Not Used** | | |
| 2 | **PWM0 DATA Reload Interval Selection Bit** | | |
| | 0 | Reload from 8-bit up counter overflow | |
| | 1 | Reload from 6-bit up counter overflow | |
| 1 | **PWM0 Counter Clear Bit (Auto Cleared)** | | |
| | 0 | No effect | |
| | 1 | Clear the PWM counter (when write) | |
| 0 | **PWM0 Enable Bit** | | |
| | 0 | Stop counter | |
| | 1 | Start (Resume countering) | |

## PWM0DAT — PWM0 Data Register — Bank2, 0DH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | | |
|---|---|---|---|
| 7-2 | **PWM Period Data** | | |
| | XXXXXX | Period Data | |
| 1-0 | **Extension Cycle Selection Bit** | | |
| | 0 | 0 | – |
| | 0 | 1 | 2 |
| | 1 | 0 | 1, 3 |
| | 1 | 1 | 1, 2, 3 |

## PWRDN — Power Down Control Register 0AH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | – | – | – | – | |
| R/W | – | – | – | – | – | – | – | – | |

| Bit | Description |
|---|---|
| 7-0 | **Power Down Control Register** |
| | This register is not physical register. The device can enter STOP mode by writing any value into this register. The SLEEP instruction is equivalent to "MOVWF PWRDN". |

## SIOCON — SIO Control Register Bank0, 15H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | – | – | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | – | – | |

| Bit | | Description |
|---|---|---|
| 7 | **SIO Shift Clock Selection Bit** | |
| | 0 | Internal clock (P.S clock) |
| | 1 | External clock (SCLK) |
| 6 | **Data Direction Control Bit** | |
| | 0 | MSB-first mode |
| | 1 | LSB-first mode |
| 5 | **SIO Mode Selection Bit** | |
| | 0 | Receive-only mode |
| | 1 | Transmit/receive mode |
| 4 | **Shift Clock Edge Selection Bit** | |
| | 0 | Tx at falling edge, Rx at rising edge |
| | 1 | Tx at rising edge, Rx at falling edge |
| 3 | **SIO Counter Clear and Shift Start Bit** | |
| | 0 | No effect |
| | 1 | Clear 3-bit counter and start shifting |
| 2 | **SIO Shift Operation Enable Bit** | |
| | 0 | Disable shifter and clock counter |
| | 1 | Enable shifter and clock counter |
| 1-0 | **Not Used** | |

## SIODAT — SIO Data Register 16H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-0 | **SIO Tx/Rx Data Value** |

## SIOPS — SIO Prescaler Data Register Bank2, 13H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-0 | **SIO Prescaler Value** |

## STAT — System Flags Register 03H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | − | 0 | 0 | 0 | Instruction Summary |
| R/W | R/W | R/W | R/W | R/W | − | R/W | R/W | R/W | |

| Bit | | | Description |
|---|---|---|---|
| 7 | **Global Interrupt Enable Bit** | | |
| | 0 | | Disable All Interrupts |
| | 1 | | Enable All Interrupts |
| 6 | **ROM Page Selection Bit** | | |
| | 0 | | Page 0 |
| | 1 | | Page 1 |
| 5-4 | **SRAM Bank Selection Bit** | | |
| | 0 | 0 | Bank 0 |
| | 0 | 1 | Bank 1 |
| | 1 | 0 | Bank 2 |
| | 1 | 1 | Not Used |
| 3 | **Not Used** (Must keep always '0') | | |
| 2 | **Zero Flag(Z)** | | |
| | Zero Flag | | |
| 1 | **Decimal Carry Flag(DC)** | | |
| | Decimal Carry Flag or Decimal/Borrow Flag | | |
| 0 | **Carry Flag(C)** | | |
| | Carry Flag or Borrow Flag | | |

## STOPCON — Stop Mode Control Register — Bank 1, 15H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description | |
|---|---|---|
| 7-0 | **Stop Mode Enable Bit** | |
| | 10100101 | Enable STOP mode |
| | others | Disable STOP mode |

## T0CON — TIMER 0 Control Register — Bank0, 0EH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | – | – | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | – | – | |

| Bit | Description | | | |
|---|---|---|---|---|
| 7 | **Timer 0 Mode Selection Bits** | | | |
| | 0 | 8-Bit Timer Mode | | |
| | 1 | 16-Bit Timer Mode | | |
| 6-4 | **Timer 0 Clock Selection Bit** | | | |
| | 0 | 0 | 0 | $f_{SYS}$ /512 |
| | 0 | 0 | 1 | $f_{SYS}$ /256 |
| | 0 | 1 | 0 | $f_{SYS}$ /64 |
| | 0 | 1 | 1 | $f_{SYS}$ /8 |
| | 1 | 0 | 0 | $f_{SYS}$ /1 |
| | 1 | 0 | 1 | $f_{SUB}$ (SUB Clock) |
| | 1 | 1 | X | T0CLK |
| 3 | **Timer 0 Counter Clear Bit (Auto Cleared)** | | | |
| | 0 | No effect | | |
| | 1 | Clear the timer 0 counter (when write) T0Data load to T0Buf | | |
| 2 | **Timer 0 Start/Stop Control Bit** | | | |
| | 0 | Stop Timer 0 | | |
| | 1 | Start/Resume Timer 0 | | |
| 1-0 | **Not Used** | | | |

## T0CNT — TIMER 0 Counter Register — Bank0, 01H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| R/W | R | R | R | R | R | R | R | R | |

| Bit | Description |
|---|---|
| 7-0 | **Timer 0 Counter Value** |

## T1CON — TIMER 0 Control Register — Bank0, 10H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | 0 | 0 | 0 | 0 | 0 | – | – | |
| R/W | – | R/W | R/W | R/W | R/W | R/W | – | – | |

| Bit | Description | | | |
|---|---|---|---|---|
| 7 | **Not Used** | | | |
| 6-4 | **Timer 1 Input Clock Selection Bit** | | | |
| | 0 | 0 | 0 | $f_{SYS}$ /512 |
| | 0 | 0 | 1 | $f_{SYS}$ /256 |
| | 0 | 1 | 0 | $f_{SYS}$ /64 |
| | 0 | 1 | 1 | $f_{SYS}$ /8 |
| | 1 | 0 | X | $f_{SYS}$ /1 |
| | 1 | 1 | X | $f_{SUB}$ (SUB Clock) |
| 3 | **Timer 1 Counter Clear Bit (Auto Cleared)** | | | |
| | 0 | No effect | | |
| | 1 | Clear the timer 0 counter (when write) T1Data load to T1Buf | | |
| 2 | **Timer 1 Start/Stop Control Bit** | | | |
| | 0 | Stop Timer 0 | | |
| | 1 | Start/Resume Timer 0 | | |
| 1-0 | **Not Used** | | | |

## T0DATA — TIMER 0 Data Register — Bank0, 0FH

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-0 | **Period Data** |

## T1DATA — TIMER 1 Data Register — Bank0, 11H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |

| Bit | Description |
|---|---|
| 7-0 | **Period Data** |

## T2CON — Timer 2 Control Register — Bank0, 12H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | See Also |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | 0 | – | – | – | 0 | 0 | – | 0 | |
| R/W | R/W | – | – | – | R/W | R/W | – | R/W | |

| Bit | | | Description |
|---|---|---|---|
| 7 | **Timer 2 Clock Selection Bit** | | |
| | 0 | | $f_{SYS}$ /128 |
| | 1 | | $f_{SUB}$ (Sub Clock) |
| 6-4 | **Not Used** | | |
| 3-2 | **Timer 2 Interval Selection Bit (Must be set T2CON.7 to 1)** | | |
| | 0 | 0 | 1.0 sec Interval |
| | 0 | 1 | 0.5 sec Interval |
| | 1 | 0 | 0.25 sec Interval |
| | 1 | 1 | 1/256 sec Interval |
| 1 | **Not Used** | | |
| 0 | **Timer 2 Enable Bit** | | |
| | 0 | | Stop Timer |
| | 1 | | Start Timer |

## WDTE — WatchDog Timer Control Register — 09H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Related Register |
|---|---|---|---|---|---|---|---|---|---|
| Reset Value | – | – | – | – | – | – | – | – | |
| R/W | – | – | – | – | – | – | – | – | |

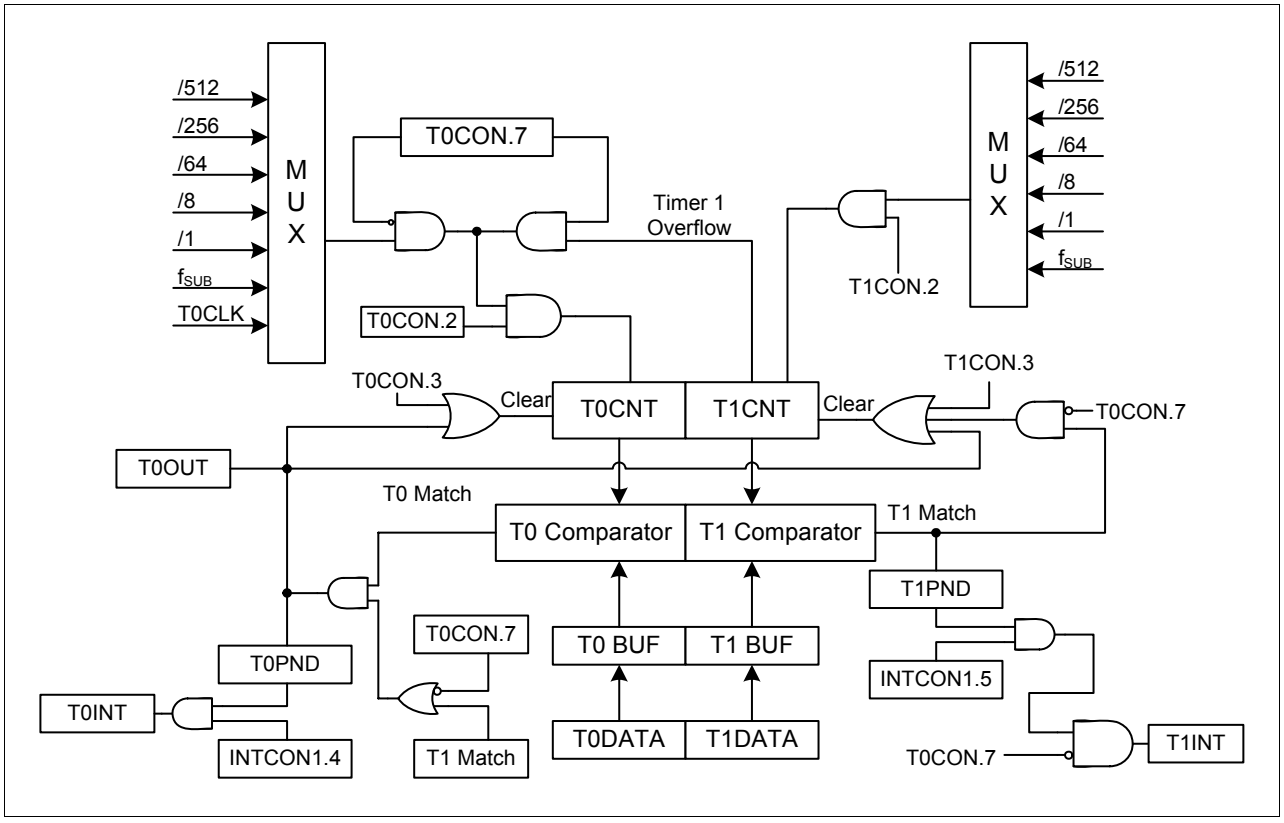
| Bit | Description |
|---|---|
| 7-0 | **WatchDog Timer Control Register** |
| | This register is not physical register. The WatchDog timer can be enabled and refreshed by CLRWDT or writing any value into this register. The CLRWDT instruction is equivalent to “MOVWF WDTE”. |

# 3. 8-Bit Timer

**Timer 0, 1**

TM59PA80 has three timers (Timer 0, Timer 1, Timer 2). Timer 0, 1 are used in one 16-bit timer or two 8-bit timers mode.

Timer 0, 1 has the following functional components:
— Clock frequency selector
— 8-bit counter, 8-bit comparator, 8-bit data register and data buffer.
— Match Interrupt generation
— Match Output Port ( T0OUT, only Timer 0)
— External Clock Input (T0CLK, only Timer 0)
— TIMER0, 1 control register (T0CON, T1CON)
— Two operating mode (Two 8-bit timer / One 16-bit timer combined two 8-bit timer)




< Figure 3-1. Timer 0, 1 Block Diagram >

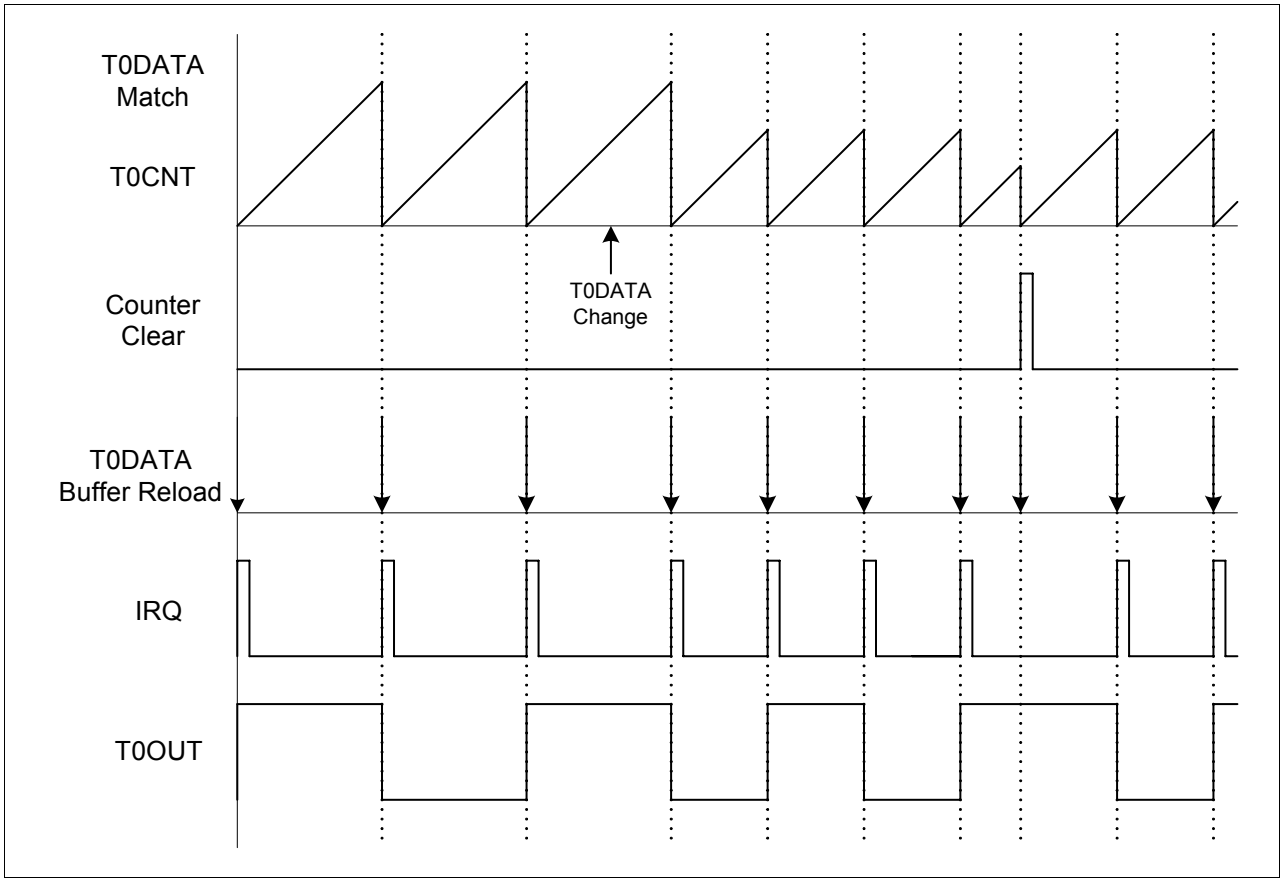
### Two 8-Bit Mode Function Description

Timer 0, 1 have interval timer mode. A match signal is generated when the counter value is identical to the value data register. The match signal generates a match interrupt, toggle T0OUT(only Timer 0), clears the counter, and counting resumes.

If the match interrupt is disabled, the match signal do not generates match interrupt request. Therefore, the timer can be used as capture mode. T0CON, T1CON are used to select input clock frequency, counter clear, Interrupt enable control and mode selection.

### One 16-Bit Mode Function Description

In 16-bit mode, timer 0 counts upper 8-bit and timer 1 counts lower 8-bit. Input clock is selected by timer 0 control register (T0CON) and timer 1 overflow signal is provided as timer 0 input clock. Timer 1 control register (T1CON) is meaningless in 16-bit mode. Timer 1 interrupt is not generated. Only timer 0 interrupt is generated which represents 16-bit timer match interrupt. When each counter (T0CNT, T1CNT) matches data register (T0DATA, T1DATA), match signal is generated. The match signal generates timer 0 interrupt if it is enabled, clears counters, toggle T0OUT, and counting resumes.



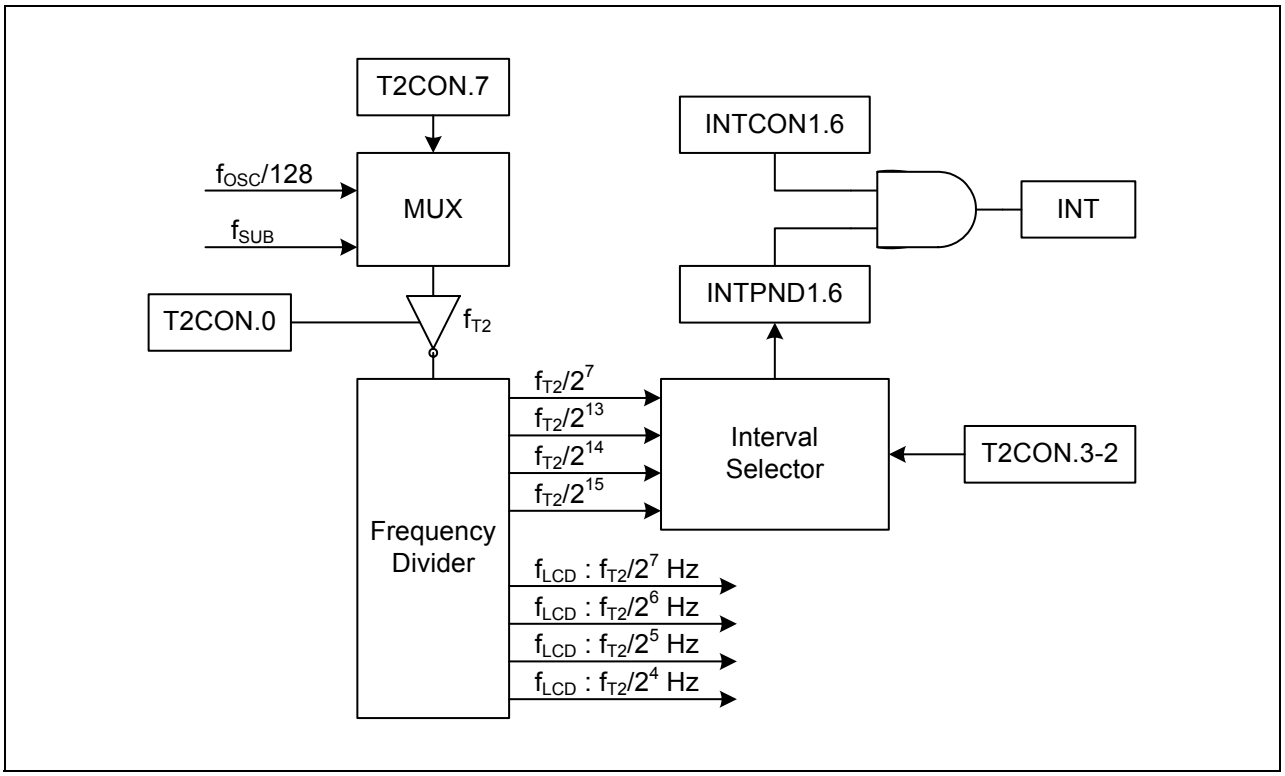

< Figure 3-2. 8-Bit Interval Mode Timing diagram (Timer 0 ) >

Example 3-1>
Timer 0, 8-bit mode Sample Code ( $f_{SYS}$ = 8.192 MHz, Interval = 1ms, T0OUT = 500 Hz )

```
                ORG     0000H
                GOTO    START

;*****************************************************************************************************************
                ORG     0001H
INT_VECTOR:
                BTFSS   INTPND1,4       ;Timer 0 interrupt Check
                GOTO    NEXT_INT        ;Jump to Other Tnterrupt Routine
                CALL    BANK_1          ;RAM BANK1
                BCF     INTPND1,4       ;Clear Timer 0 pending bit
NEXT_INT:
                RETI

;*****************************************************************************************************************
START:
                CALL    BANK_0
                MOVLW   1FH
                MOVWF   T0DATA          ;Set T0DATA 31(D)

                MOVLW   00010000B       ;(8-bit Timer Mode)
                MOVWF   T0CON           ;T0timer = fsys/Prescaler
                                        ;Ratio/(period Data +1)
                                        ;T0(interrupt)
                                        ;=8.192M(HZ) / 256 / 32  =  1K (HZ)

                CALL    BANK_1
                BSF     PACONL,0
                BSF     PACONL,1        ;Port A.0 Configuration T0OUT
                BSF     INTCON1,4       ;Timer 0 Match Interrupt Enable Bit

                CALL    BANK_0
                BSF     T0CON,3         ;Timer 0 Counter Clear
                BSF     STAT,7          ;Enable All Interrupts
                BSF     T0CON,2         ;Start Timer 0

LOOP:
                GOTO    LOOP

;*****************************************************************************************************************
BANK_0:
                BCF     STAT,4
                BCF     STAT,5
                RET                     ;RAM_BANK_0
BANK_1:
                BSF     STAT,4
                BCF     STAT,5
                RET                     ;RAM_BANK_1
BANK_2:
                BCF     STAT,4
                BSF     STAT,5
                RET                     ;RAM_BANK_2

;*****************************************************************************************************************
```

**Timer 2**

Timer 2 is used to watch-time measurement, LCD clock generation and interval timing for the system clock. Watch timer has the following functional components:
— Real Time and Watch-Time Measurement
— Using a Main clock or SUB Clock ($f_{SYS}$/128 or $f_{SUB}$)
— LCD Clock ( $f_{LCD}$ ) Generator : $f_{T2}/2^7$, $f_{T2}/2^6$, $f_{T2}/2^5$, $f_{T2}/2^4$
— Timing Tests in High-Speed Mode
— Overflow interrupt generation ( 1s, 0.5s, 0.25s, 3.91ms)
— Timer 2 control register, T2CON

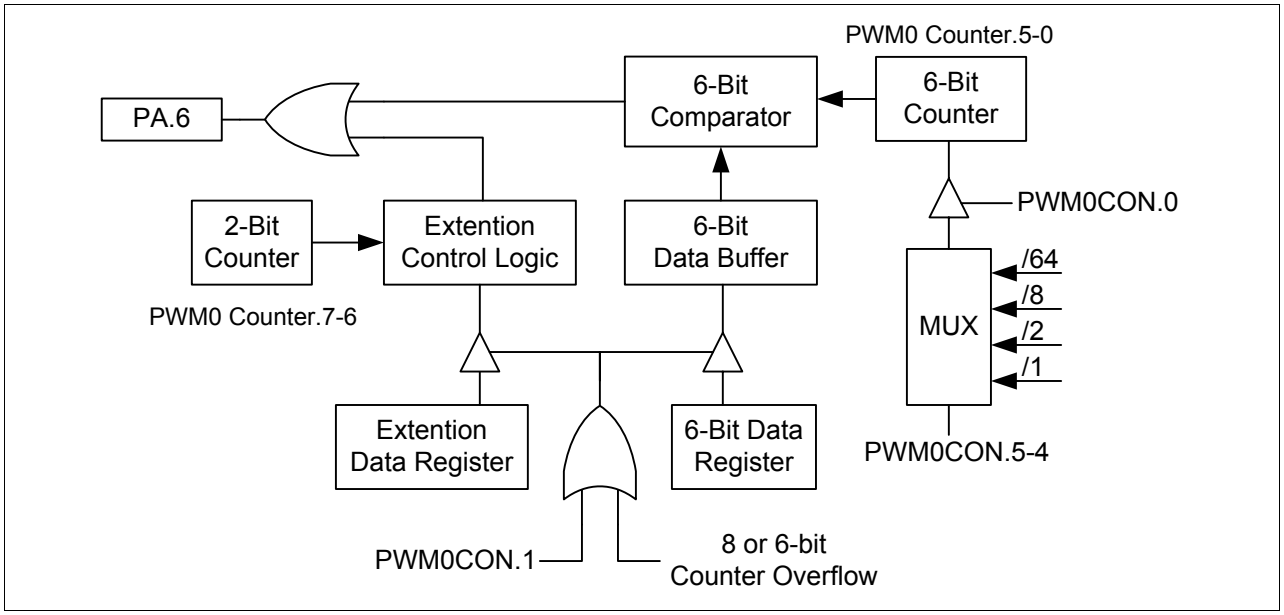



< Figure 3-3. Timer 2 Block Diagram >

# 4. 8-Bit PWM

PWM0 has the following functional components:
— Clock frequency selector
— 8-bit up-counter, 6-bit comparator, 6-bit data register and 6-bit data buffer.
— 2-bit extension control logic, 2-bit extension register and extension data buffer.
— Control register (PWM0CON)

To determine the PWM0 operating frequency, the upper 6-bits of counter is compared to the PWM0 data register (PWM0DAT.7−.2). In order to achieve higher resolutions, the lower 2-bits of the counter can be used to modulate the "extended" cycle.




< Figure 4-1. Block Diagram >

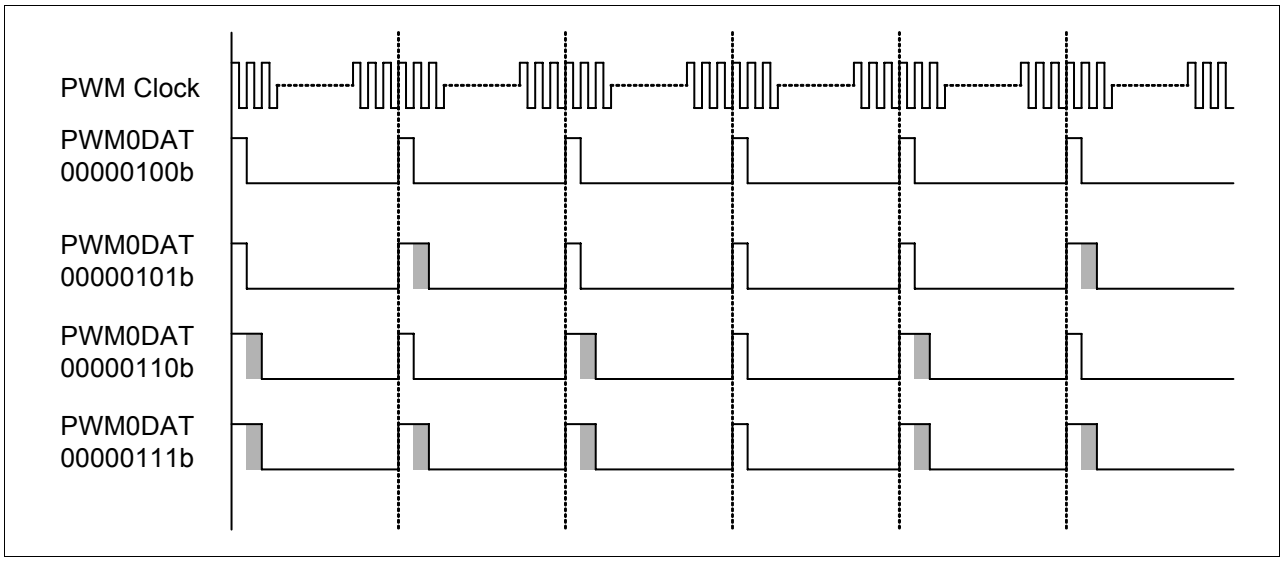
The PWM output signal toggles to Low level whenever the lower 6-bit of counter matches the reference data register (PWM0DAT.7−.2). If the value in the PWM0DAT.7−.2 register is not zero, an overflow of the lower 6-bits of counter causes the PWM output to toggle to High level. In this way, the reference value written to the reference data register determines the module's base duty cycle.

The value in the upper 2-bits of counter is compared with the extension settings in the 2-bit extension data register (PWM0DAT.1−.0). This lower 2-bits of counter value is used to "extend" the duty cycle of the PWM output. The " extension " value is one extra clock period at specific cycles (see Table 4-1).

| PWM0DAT.1-0 | Extended Cycle |
|---|---|
| 00 | None |
| 01 | 2 |
| 10 | 1, 3 |
| 11 | 1, 2, 3 |

< Table 4-1. PWM output extended cycle >

For example, if the value in the extension data register is '01B', the 2nd cycle will be one pulse longer than the other 3 cycles. (see Figure 4-2).




< Figure 4-2. Extended Output >

Example 4-1>
PWM0 Sample Code ( $f_{SYS}$ = 8.192 MHz, 1 Cycle = 500μs, Extend 2nd Cycle )

```
MOVLW    05h          ; Set PWM0 Data Register
MOVWF    PWM0DAT      ; Data = 1, Extension = 1

BSF      PCCON, 6
BSF      PCCON, 7     ; Select PCCON.76 '11' PWM0 Out.

CLRF     PWM0CON      ; fsys/64, 8-bit Overflow Reload, PWM Stop

BSF      PWM0CON, 1   ; PWM0 Counter Clear
BSF      PWM0CON, 0   ; PWM0 Start
.
.
```

# 5. Analog to Digital Converter

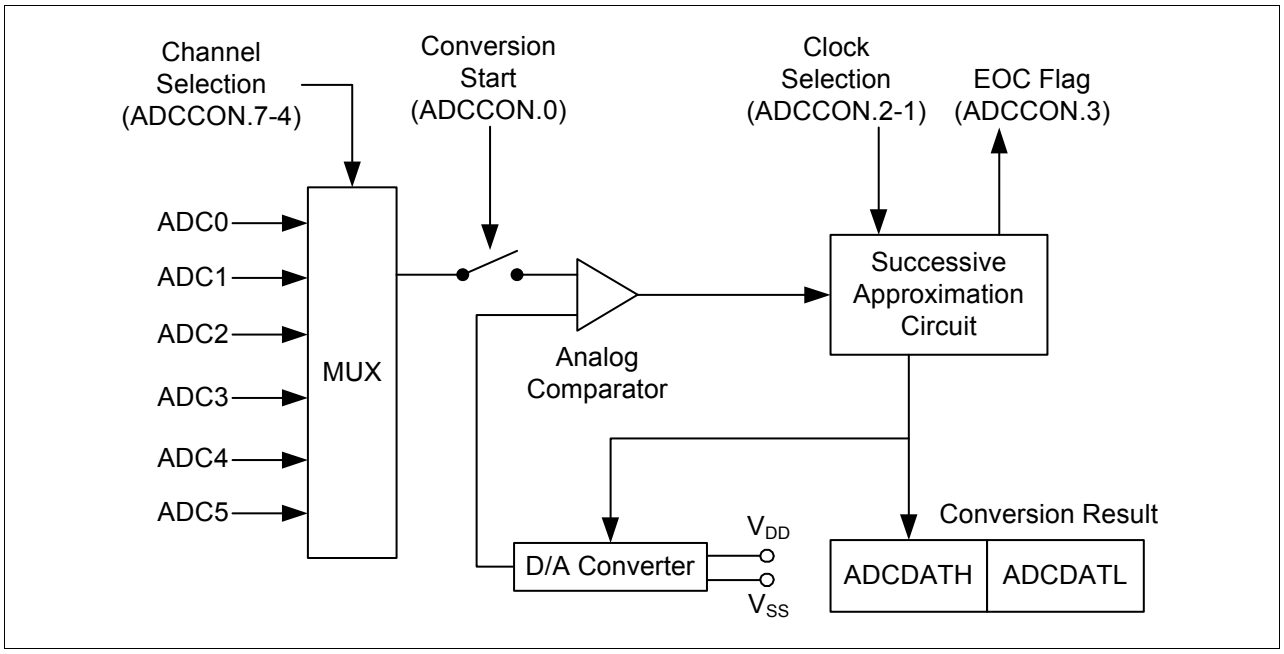
The 10-bit CMOS ADC(Analog to Digital Converter) consists of a 6-channel analog input multiplexer, control register, clock generator, 10 bit successive approximation register , and output register.

### A/D CONVERSION PROCEDURE

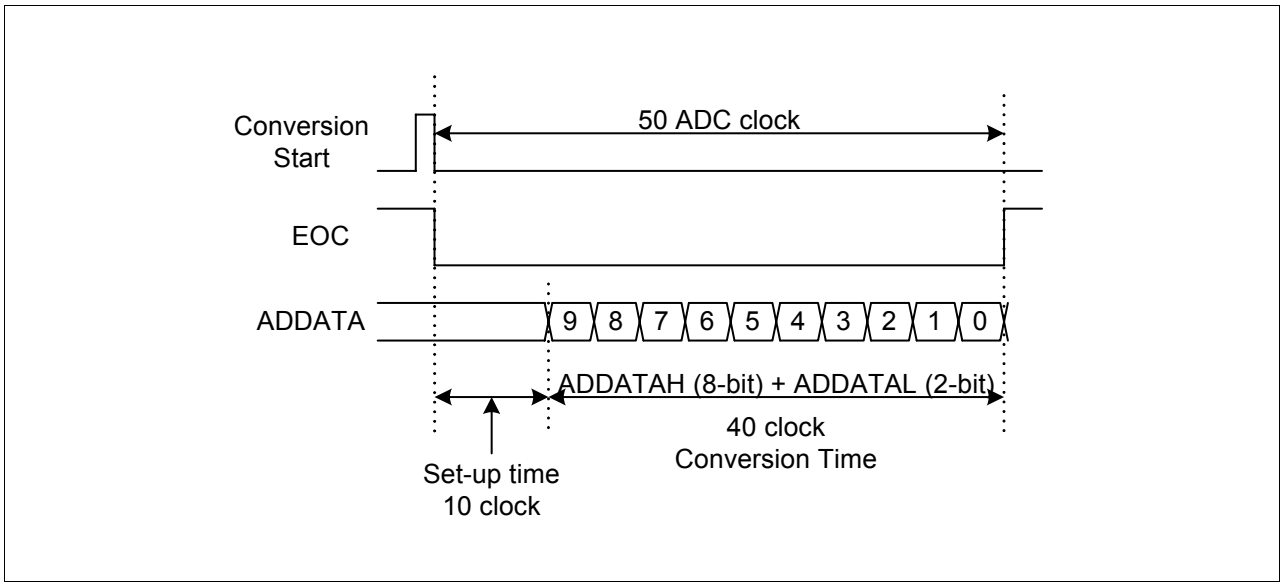
**1.** Configure the analog input pins to ADC input mode by making the appropriate settings int the I/O port control registers.
**2.** Select ADC input channel.
**3.** Start conversion by set the ADCCON.0 to '1'.
**4.** When conversion has been completed, the EOC flag is set to '1'.
**5.** The converted digital value is loaded to the ADCDATL, ADCDATH register, and then the ADC module enters an idle state.
**6.** The digital conversion result can now be read from the ADDATAH, ADDATAL register.

If the chip enters to STOP mode in conversion process, there will be a leakage current path in A/D block. The ADC operation must be finished before the chip enters STOP mode.

※ There is not sampling/hold circuit in ADC. Therefore, it is important that any fluctuations in the analog level at the ADC0−ADC5 input pins during a conversion procedure be kept to an absolute minimum. Any change in the input level, perhaps due to circuit noise, will invalidate the result.




< Figure 5-1. Analog to Digital Converter Block Diagram >




< Figure 5.2. A/D Conversion Timing Diagram >

※ Maximum ADC Input Clock is 4MHz.

Example 5-1>
ADC Sample Code

```
      MOVLW    00000100b    ; fSYS/4, ADC0
      MOVWF    ADCCON       ; Configure ADCCON

      BSF      PBCON, 0
      BSF      PBCON, 1     ; Configure PB.0 ADC Input 0

      BSF      ADCCON, 0    ; Start Conversion
ADC_LOOP:
      BTFSS    ADCCON, 3
      GOTO     ADC_LOOP     ; Wait until EOC bit is set

                            ; Converted value can be read from ADDATL and
                            ; ADDATH.
```

# 6. I/O Ports

The TM59PA80 has seven I/O ports, PORTA ~ PORTG (36 Pin-44QFP). These ports can be accessed directly by writing or reading port data register.

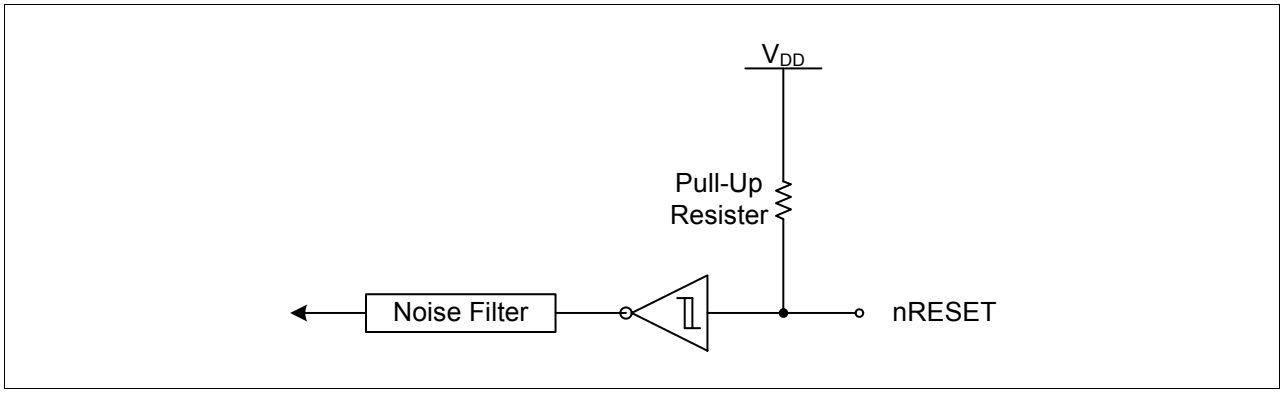
| Pin Name | Bit | Pin No | Pin Description | I/O | PIN Type |
|---|---|---|---|---|---|
| $V_{DD}$, $V_{SS}$ | – | 5,6 | Power input pins for internal power block | – | – |
| $X_{IN}$, $X_{OUT}$ | – | 8,7 | Main oscillator pins for main clock | – | – |
| $XT_{IN}$, $XT_{OUT}$ | – | 10,11 | Sub oscillator pins for sub clock | – | – |
| nTEST | – | 9 | Chip test input pin. Hold $V_{DD}$ when the device is operating | – | – |
| nRESET | – | 12 | Reset signal input pin. Schmitt trigger input with internal pull-up resistor. | – | R |
| PORTA | 0 | 39 | Schmitt trigger input, Push-pull output, Open-Drain output, External Interrupt 0, Timer 0 Output | I/O | B |
| | 1 | 40 | Schmitt trigger input, Push-pull output, Open-Drain output, External Interrupt 1, Timer 0 external clock input | I/O | |
| | 2 | 41 | Schmitt trigger input, Push-pull output, Open-Drain output, External Interrupt 2, Clock Output | I/O | |
| | 3 | 42 | Schmitt trigger input, Push-pull output, Open-Drain output, External Interrupt 3, Buzzer Out | I/O | |
| | 4 | 43 | Schmitt trigger input, Push-pull output, Open-Drain output, ADC4 | I/O | C |
| | 5 | 44 | Schmitt trigger input, Push-pull output, Open-Drain output, ADC5 | I/O | |
| PORTB | 0 | 1 | Schmitt trigger input, Push-pull output, Open-Drain output, ADC0 | I/O | |
| | 1 | 2 | Schmitt trigger input, Push-pull output, Open-Drain output, ADC1 | I/O | |
| | 2 | 3 | Schmitt trigger input, Push-pull output, Open-Drain output, ADC2 | I/O | |
| | 3 | 4 | Schmitt trigger input, Push-pull output, Open-Drain output, ADC3 | I/O | |
| PORTC | 0 | 16 | Schmitt trigger input, Push-pull output, Open-Drain output, SEG1, SCLK | I/O | D-2 |
| | 1 | 15 | Schmitt trigger input, Push-pull output, Open-Drain output, SEG0, SO | I/O | D-2 |
| | 2 | 14 | Schmitt trigger input, Push-pull output, Open-Drain output, SI | I/O | B |
| | 3 | 13 | Schmitt trigger input, Push-pull output, Open-Drain output, PWM0 | I/O | B |

< Table 6-1. Port Configuration Overview ①>

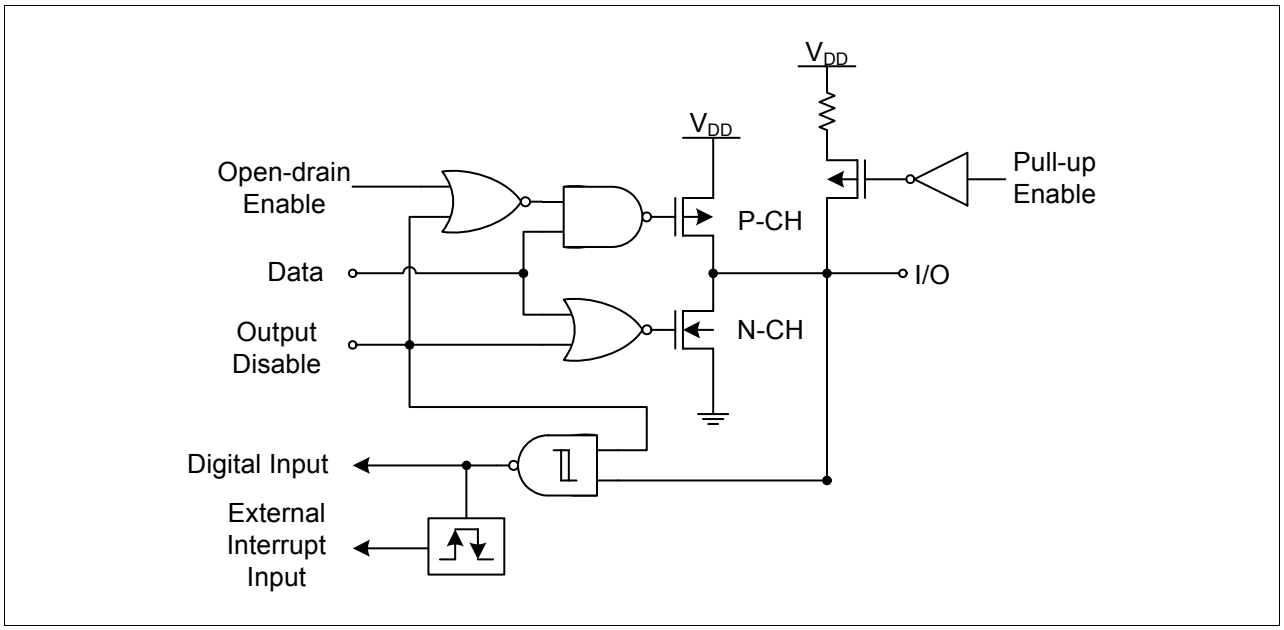
| PORT | Bit | Pin No | Pin Description | I/O | PIN Type |
|---|---|---|---|---|---|
| PORTD | 0 | 18 | Schmitt trigger input, Push-pull output, Open-Drain output, External Interrupt 9, SEG3 | I/O | D-1 |
| | 1 | 17 | Schmitt trigger input, Push-pull output, Open-Drain output, External Interrupt 8, SEG2 | I/O | D-1 |
| PORTE | 0 | 19 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG4 | I/O | D-2 |
| | 1 | 20 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG5 | I/O | |
| | 2 | 21 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG6 | I/O | |
| | 3 | 22 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG7 | I/O | |
| | 4 | 23 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG8 | I/O | |
| | 5 | 24 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG9 | I/O | |
| | 6 | 25 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG10 | I/O | |
| | 7 | 26 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG11 | I/O | |
| PORTF | 0 | 27 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG12 | I/O | D-2 |
| | 1 | 28 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG13 | I/O | |
| | 2 | 29 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG14 | I/O | |
| | 3 | 30 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG15 | I/O | |
| | 4 | 31 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG16, COM7 | I/O | |
| | 5 | 32 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG17, COM6 | I/O | |
| | 6 | 33 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG18, COM5 | I/O | |
| | 7 | 34 | Schmitt trigger Input, Push-pull output, Open-Drain output, SEG19, COM4 | I/O | |
| PORTG | 0 | 35 | Schmitt trigger Input, Push-pull output, Open-Drain output, COM0 | I/O | D-2 |
| | 1 | 36 | Schmitt trigger Input, Push-pull output, Open-Drain output, COM1 | I/O | |
| | 2 | 37 | Schmitt trigger Input, Push-pull output, Open-Drain output, COM2 | I/O | |
| | 3 | 38 | Schmitt trigger Input, Push-pull output, Open-Drain output, COM3 | I/O | |

< Table 6-1. Port Configuration Overview ②>

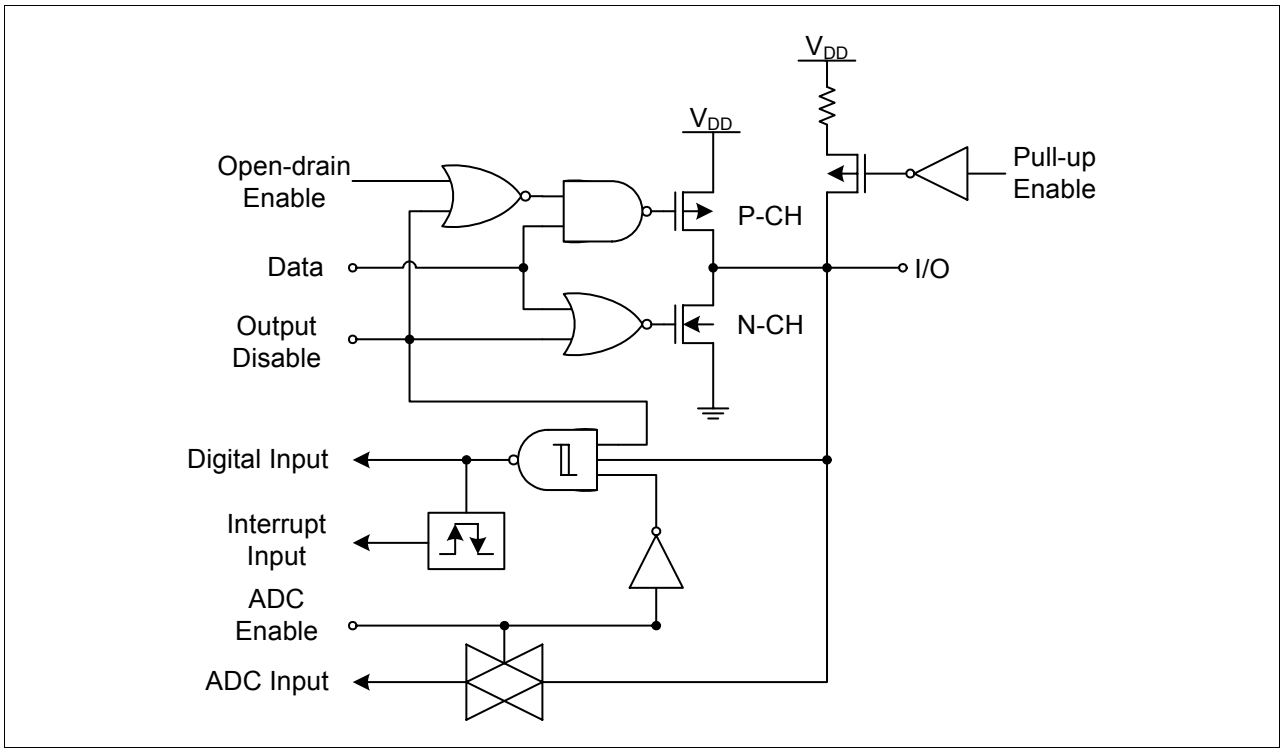
## Pin Circuit



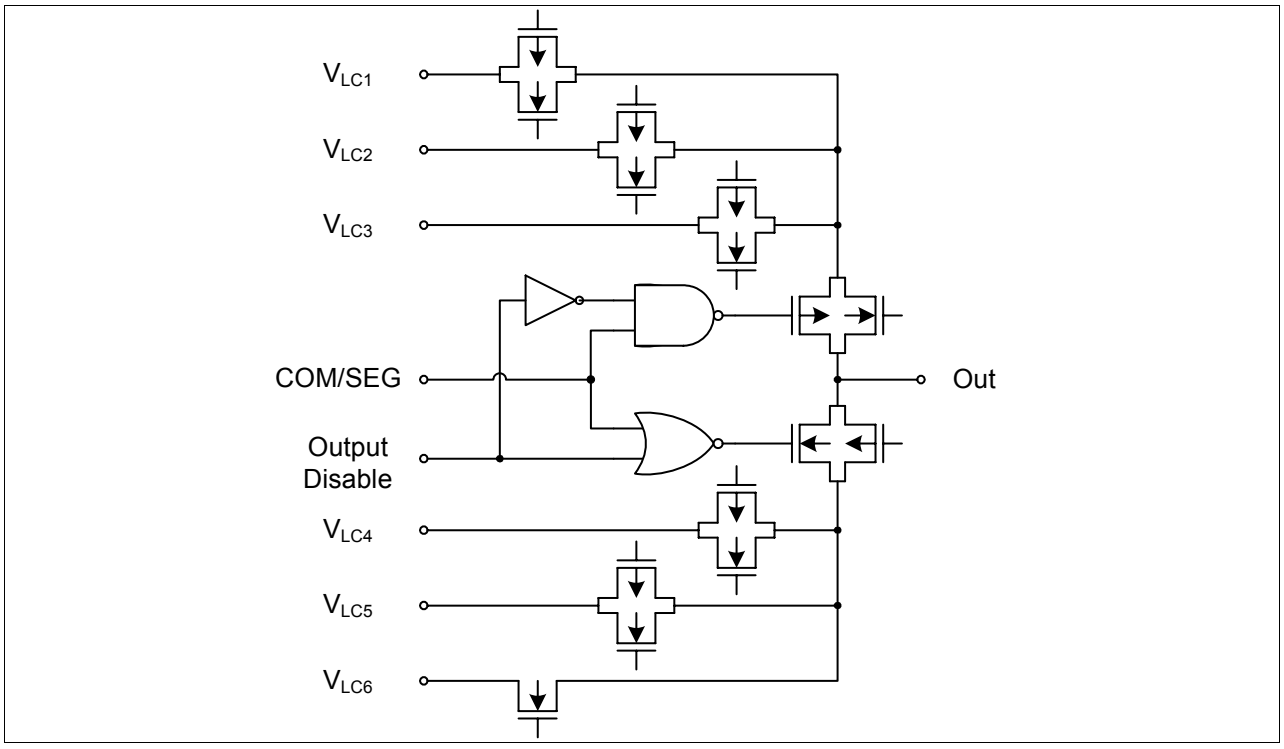

< Figure 6-1. Pin Circuit Type R >



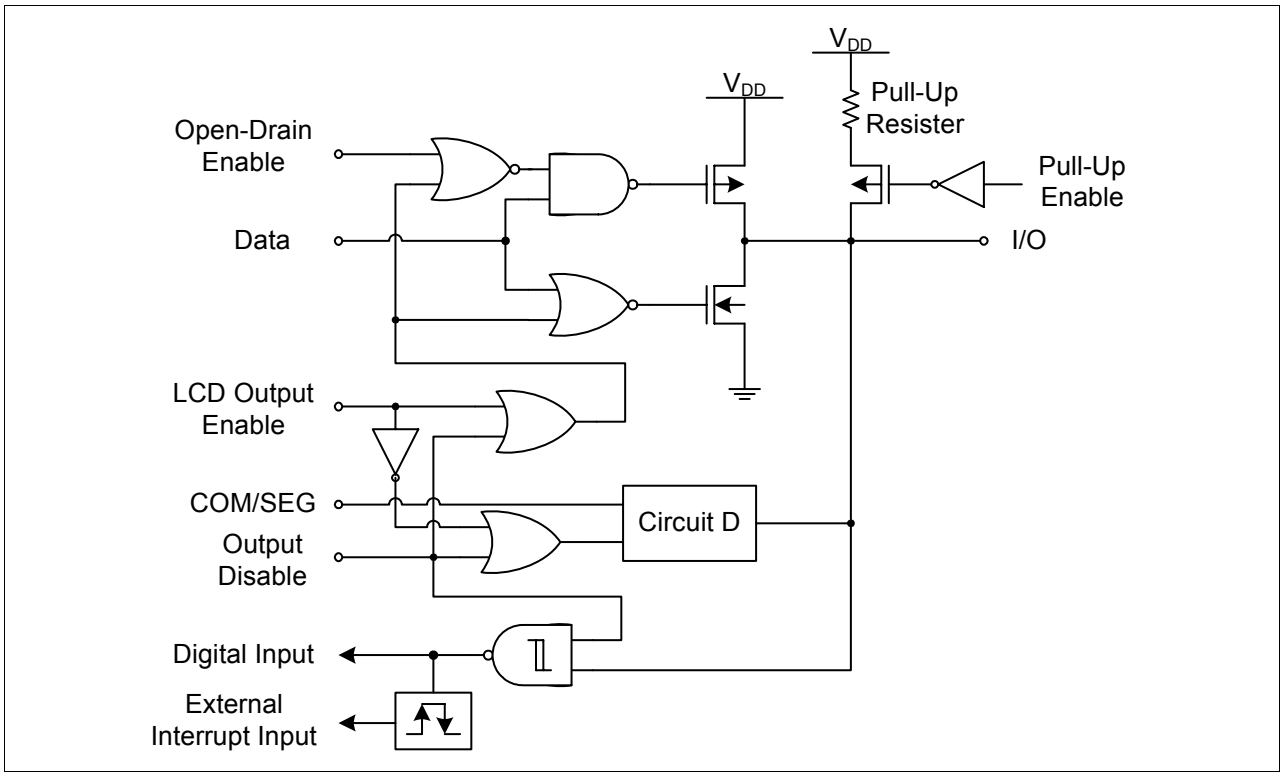

< Figure 6-2. Pin Circuit Type B >



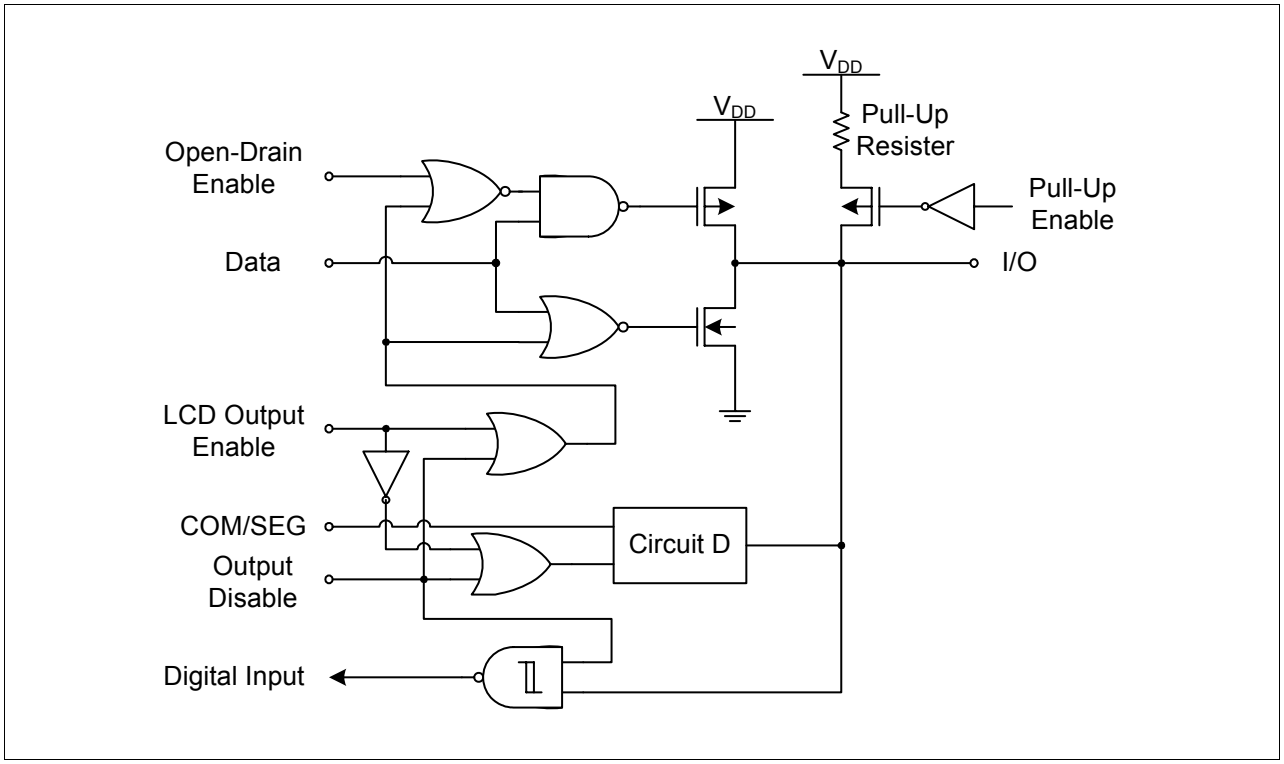

< Figure 6-3. Pin Circuit Type C >




< Figure 6-4. Pin Circuit Type D >




< Figure 6-5. Pin Circuit Type D-1 >




< Figure 6-6. Pin Circuit Type D-2 >

**PORTA**

Port A has 6-bit I/O Pins. It can be used for normal I/O (Schmitt trigger input, push-pull output, open-drain output) or some alternative function (ADC, External interrupt, Buzzer output, Timer 0 Match Output, Timer 0, External Clock input, Clock output). PA.2 can be used as CLO function which outputs CPU clock.

**PORTB**

Port B has 4-bit I/O Pins. It can be used for normal I/O (Schmitt trigger input, push-pull output, open-drain output) or some alternative function (ADC, External interrupt, Buzzer output).

**PORTC**

Port C has 4-bit I/O Pins. It can be used for normal I/O (Schmitt trigger input, push-pull output, open-drain output) or some alternative function (LCD SEG output, SCLK, SI, SO, PWM out). LCD Function can be selected in LPCON register.

**PORTD**

Port D as 2-bit I/O Pins. It can be used for normal I/O (Schmitt trigger input, push-pull output, open-drain output) or some alternative function (LCD SEG output, External interrupt). LCD Function can be selected in LPCON register.

**PORTE**

Port E has 8-bit I/O Pins. It can be used for normal I/O (Schmitt trigger Input, push-pull output, open-drain output) or LCD SEG output. LCD Function can be selected in LPCON register.

**PORTF**

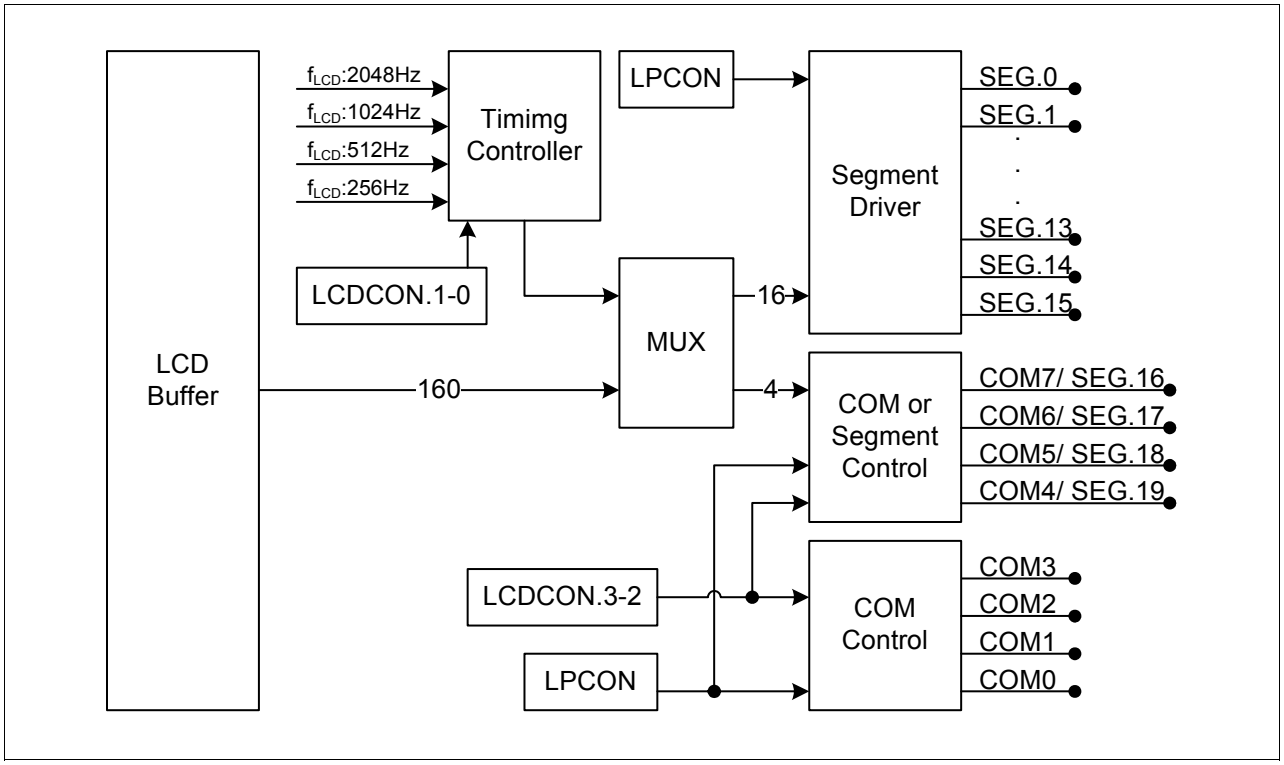
Port F has 8-bit I/O Pins. It can be used for normal I/O (Schmitt trigger Input, push-pull output, open-drain output), LCD SEG output or LCD COM output. LCD Function can be selected in LPCON register.

**PORTG**

Port G has 4-bit I/O Pins. It can be used for normal I/O (Schmitt trigger Input, push-pull output, open-drain output) or LCD COM output. LCD Function can be selected in LPCON register.

# 7. LCD Controller

The LCD controller generates the timing control to drive an up-to-128-dot (8comx16seg) LCD panel. LCD controller has the following components:
— LCD controller/driver
— LCD data register for storing display data: (20h-33h: BANK2)
— 16 segment output pins in 8 common mode (SEG0.SEG15)
— 20 segment output pins in 3 or 4 common mode (SEG0.SEG19)
— 8 common output pins (COM0.COM7)
— Internal resistor circuit for LCD bias: LCDCON
— COM, SEG enable control register: LPCON




< Figure 7-1. LCD Controller Block Diagram >

LCDCON is used to define the timing requirements of the LCD panel, input clock, duty and bias selection and display on/off control. Table 7-1 shows COM and SEG pins per Duty Cycle. Timer2 provides LCD clock ($f_{SYS}$/128 or $f_{SUB}$). Therefore, if the Timer2 is stopped, the LCD controller does not operate. When a sub clock is selected as the LCD clock source, the LCD display is enabled even during main clock stop and idle modes.

| Duty | COM | SEG | Dot Count |
|---|---|---|---|
| 1/3 | COM0 ~ COM2 | SEG0 ~ SEG19 | 60 |
| 1/4 | COM0 ~ COM3 | SEG0 ~ SEG19 | 80 |
| 1/8 | COM0 ~ COM7 | SEG0 ~ SEG15 | 128 |

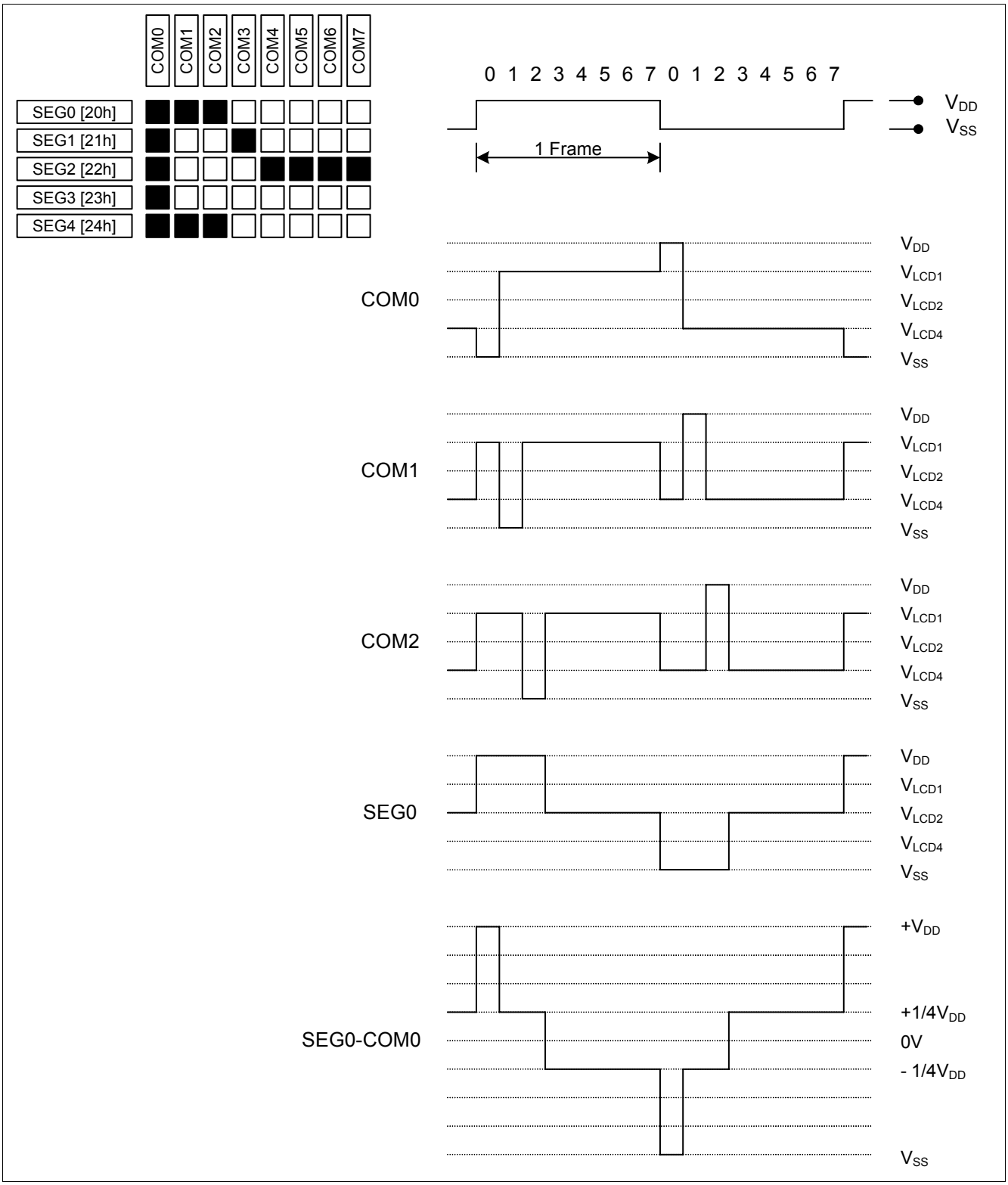
< Table 7-1. COM & SEG per Duty >

LPCON register is used to configure common and segment pins output enable/disable. In normal operation, these control registers are configured to match the LCD panel being used.

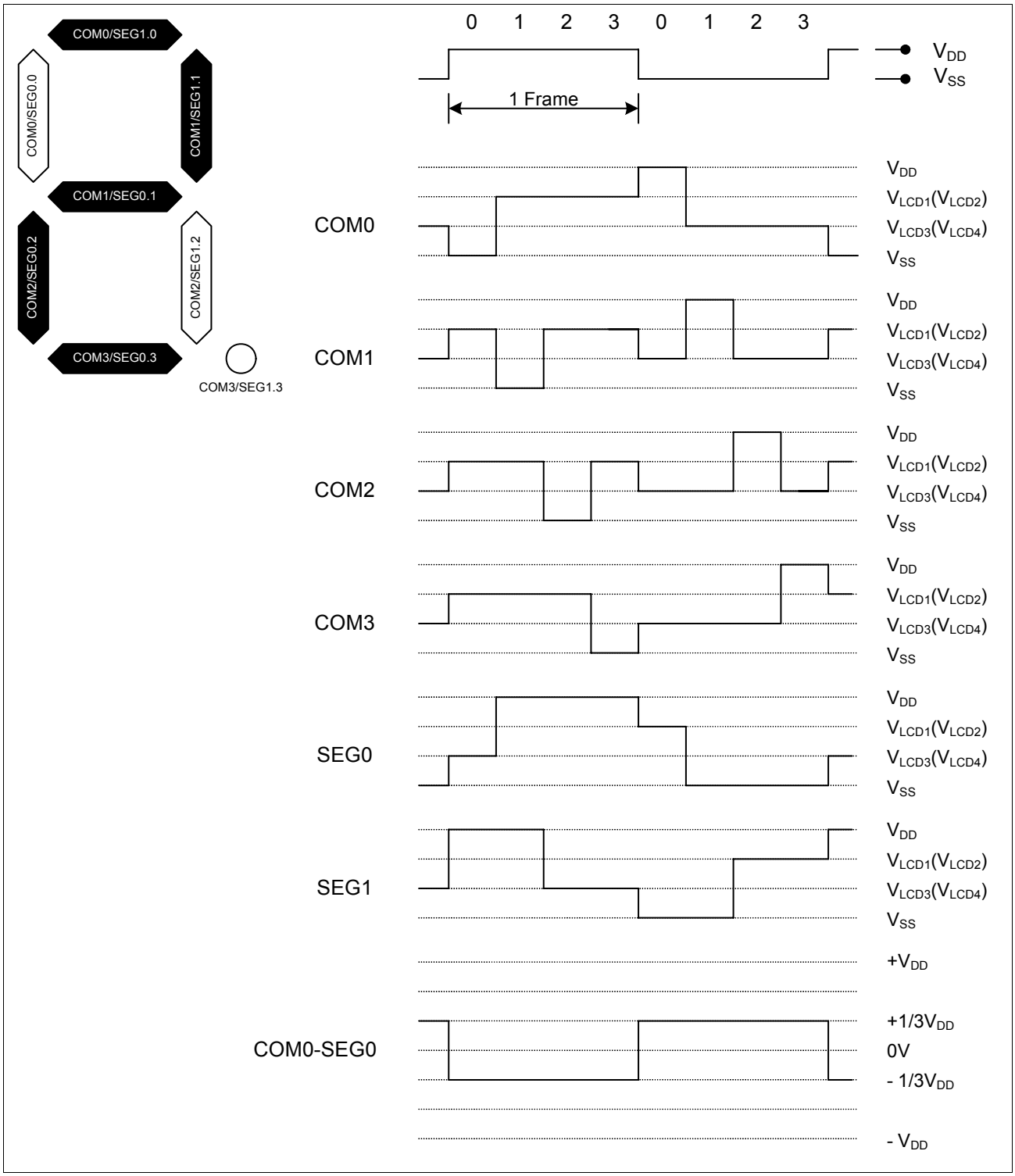
The individual bits of the LCD data registers are cleared/set to represent a clear/dark pixel, respectively. And the data can be transferred to the segment signal pins automatically without program control.

| COM7 | COM6 | COM5 | COM4 | COM3 | COM2 | COM1 | COM0 | |
|---|---|---|---|---|---|---|---|---|
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |
| 20h | | | | | | | | SEG0 |
| 21h | | | | | | | | SEG1 |
| 22h | | | | | | | | SEG2 |
| . | | | | | | | | . |
| . | | | | | | | | . |
| . | | | | | | | | . |
| 31h | | | | | | | | SEG17 |
| 32h | | | | | | | | SEG18 |
| 33h | | | | | | | | SEG19 |

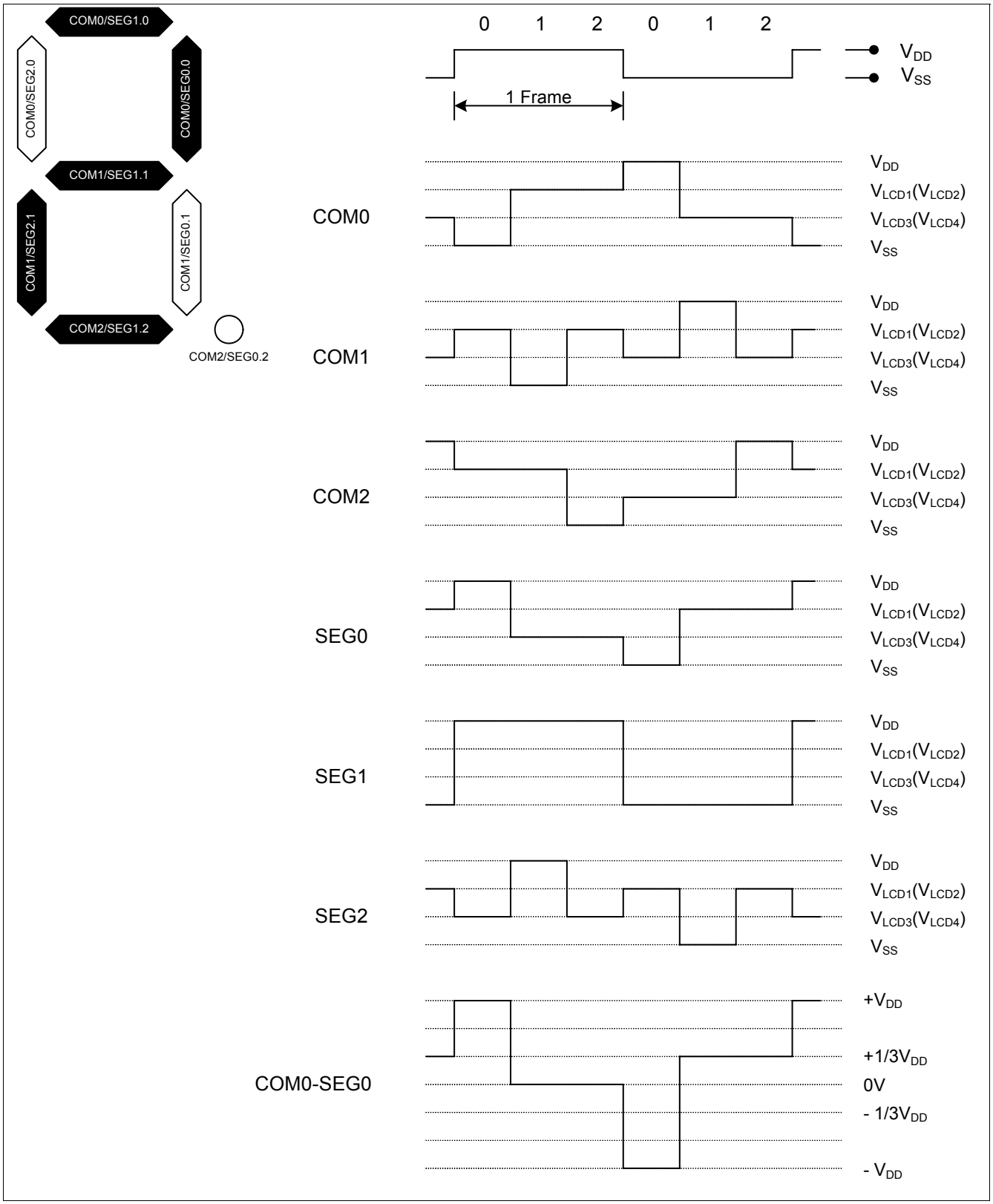
< Figure 7-2. LCD Buffer Organization >




< Figure 7-3. LCD Signal (1/8 Duty, 1/4 Bias) >



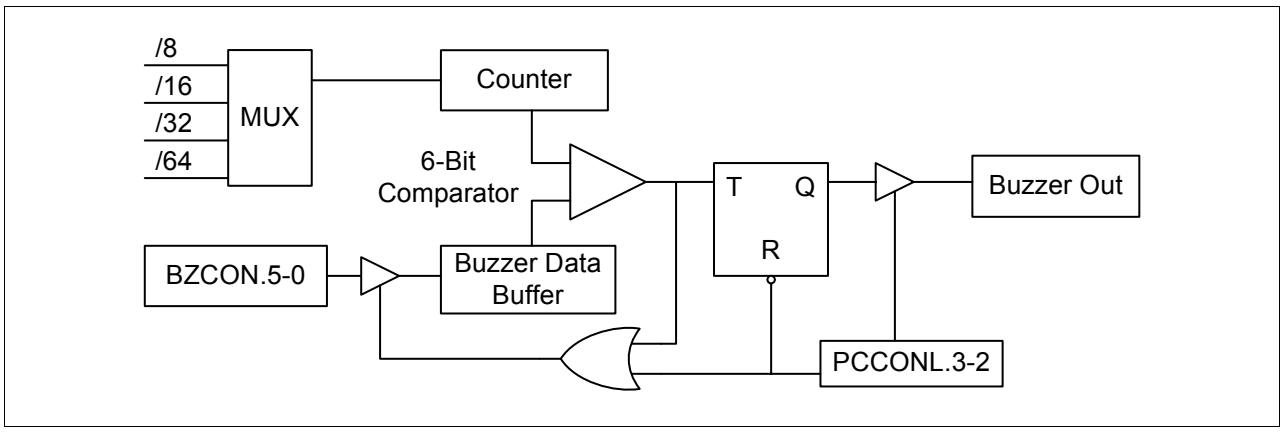

< Figure 7-4. LCD Signal (1/4 Duty, 1/3 Bias) >




< Figure 7-5. LCD Signal (1/3 Duty, 1/3 Bias) >

# 8. Buzzer Out

The TM59PA80 has Buzzer driver that consist of 6-bit counter, clock divider, control register. It generates 50% duty square-wave and the frequency cover a wide range.




< Figure 8-1. Block Diagram >

It can be enabled by setting the bit PC.1 as Buzzer out function. When the Buzzer Out is enabled, the 6-bit counter is cleared and PC.1 output status is '0' and start counting up. If the counter value is match up to period data (BZCON.5-0), then PC.1 output status is toggle and the counter is cleared. Also, the counter is cleared by 6-bit counter overflow. BZCON.5-0 determines output frequency. Frequency calculation is as follows.

$$F_{BZ} = f_{SYS}/2/\text{Prescaler Ratio}/(\text{Period Data} + 1)$$

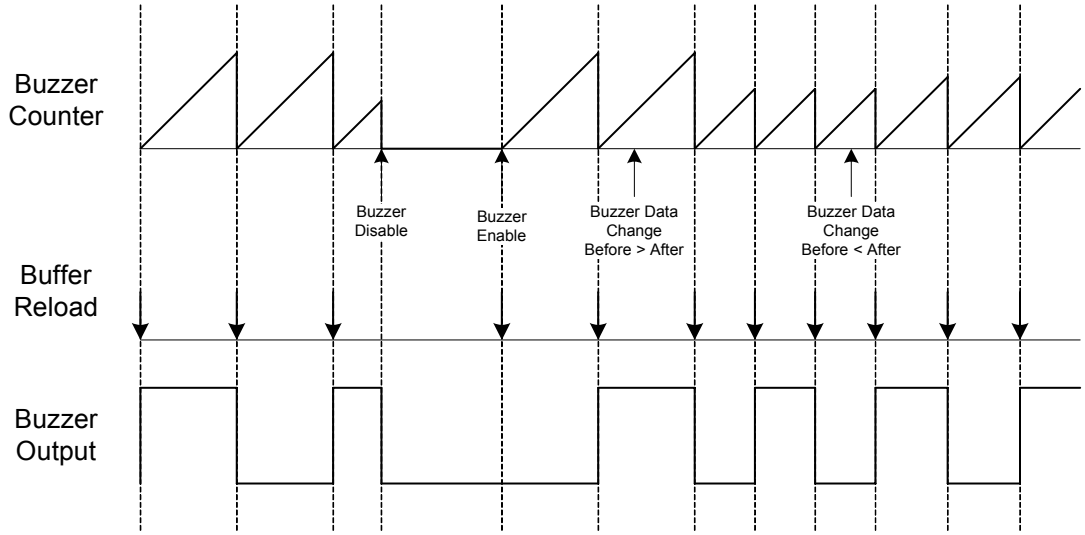
Example 8-1>
Output frequency calculation

System Clock ($f_{SYS}$) : 8.192MHz
Prescaler Ratio (BZCON.7-6) : 11 ($f_{SYS}$ /64),
Period Data (BZCON.5-0) : 9

$F_{BZ}$ 8.192M / 2 / 64 / (9+1) = 64000 (Hz)

Example 8-2>
Sample Code

```
MOVLW   11001001b   ; fSYS/64, Period Data 9 ( 6.4 KHz Output )
MOVWF   BZCON       ; Set Buzzer 1KHz Output

BSF     PACONL, 6
BSF     PACONL, 7   ; Set PA.3 Buzzer out
.
.
.
BCF     PACONL, 7
BCF     PACONL, 6   ; Set PA.3 Input mode. Buzzer Disable
```

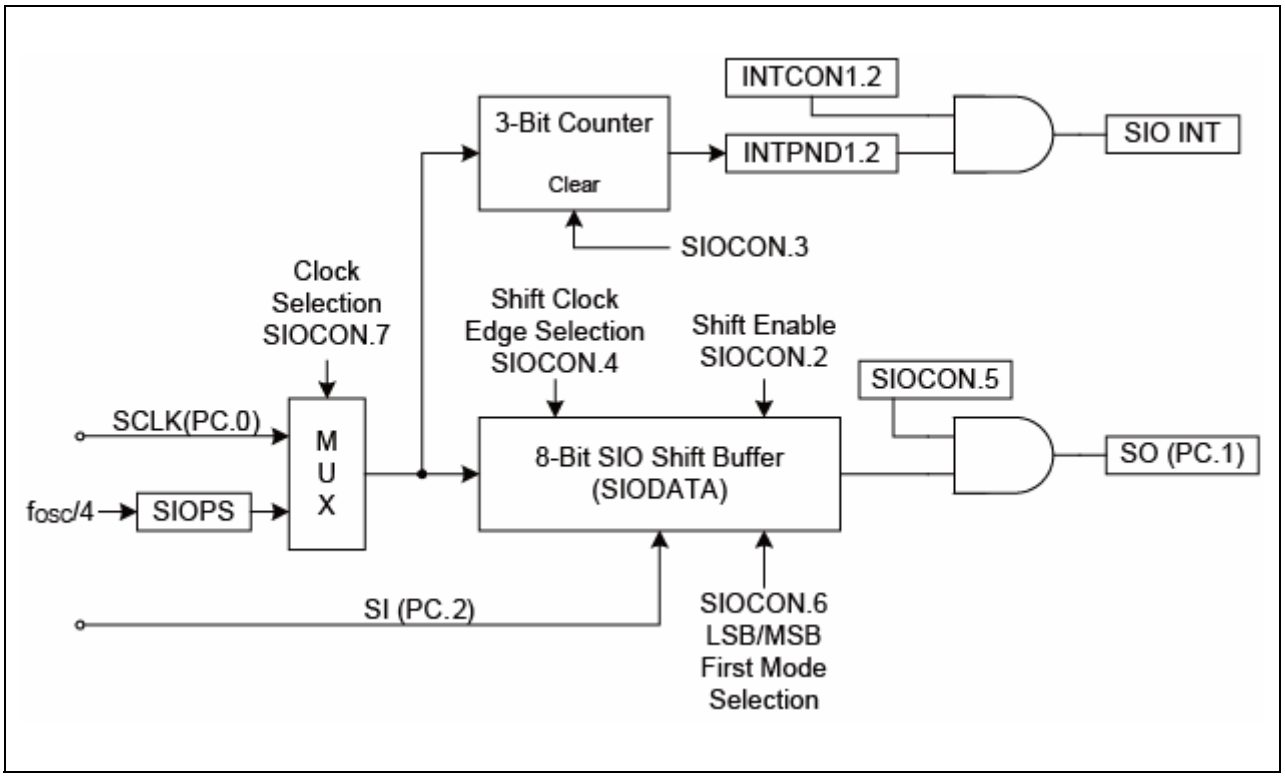



< Figure 8-2. Timing Diagram >

# 9. Serial I/O

The SIO can interface with various types of external device that requires serial data transfer. The SIO block has the following components:
— 8-bit control register (SIOCON)
— Clock selector logic
— 8-bit data buffer (SIODAT)
— 8-bit prescaler (SIOPS)
— 3-bit serial clock counter
— Serial data I/O pins (SI, SO)
— Serial clock input/output pin (SCLK)

The SIO module can transmit or receive 8-bit serial data at a frequency determined by its corresponding control register settings. To ensure flexible data transmission rates, you can select an internal or external clock source.




< Figure 9-1. SIO Block Diagram >

## PROGRAMMING PROCEDURE

To program the SIO module, follow these basic steps:

1. Configure the I/O pins at port (SCLK/SI/SO) by loading the appropriate value to the PCCON and LPCON register if necessary.
2. Load an 8-bit value to the SIOCON control register to properly configure the serial I/O module. In this operation, SIOCON.2 must be set to "1" to enable the data shifter.
3. For interrupt generation, set the serial I/O interrupt enable bit (INTCON1.2) to "1".
4. When you transmit data to the serial buffer, write data to SIODAT and set SIOCON.3 to 1, the shift operation starts.
5. When the shift operation (transmit/receive) is completed, the SIO pending bit (INTPND2.2) are set to "1" and SIO interrupt request is generated.

## SIO CONTROL REGISTERS (SIOCON)

The control register for serial I/O interface module, SIOCON. It has the control setting for SIO module.

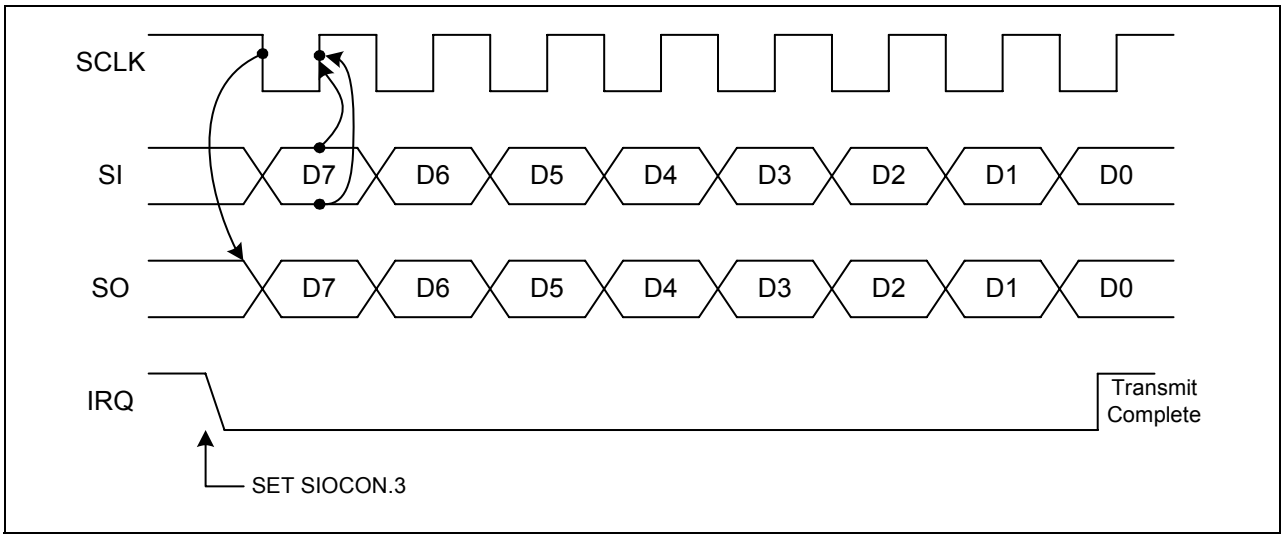
— Clock source selection (internal or external) for shift clock
— Edge selection for shift operation
— Clear 3-bit counter and start shift operation
— Shift operation (transmit) enable
— Mode selection (transmit/receive or receive-only)
— Data direction selection (MSB first or LSB first)

A reset clears the SIOCON value to "00H". This configures the corresponding module with an internal clock source at the SCLK, selects receive-only operating mode, and clears the 3-bit counter. The data shift operation and the interrupt are disabled. The selected data direction is MSB-first.

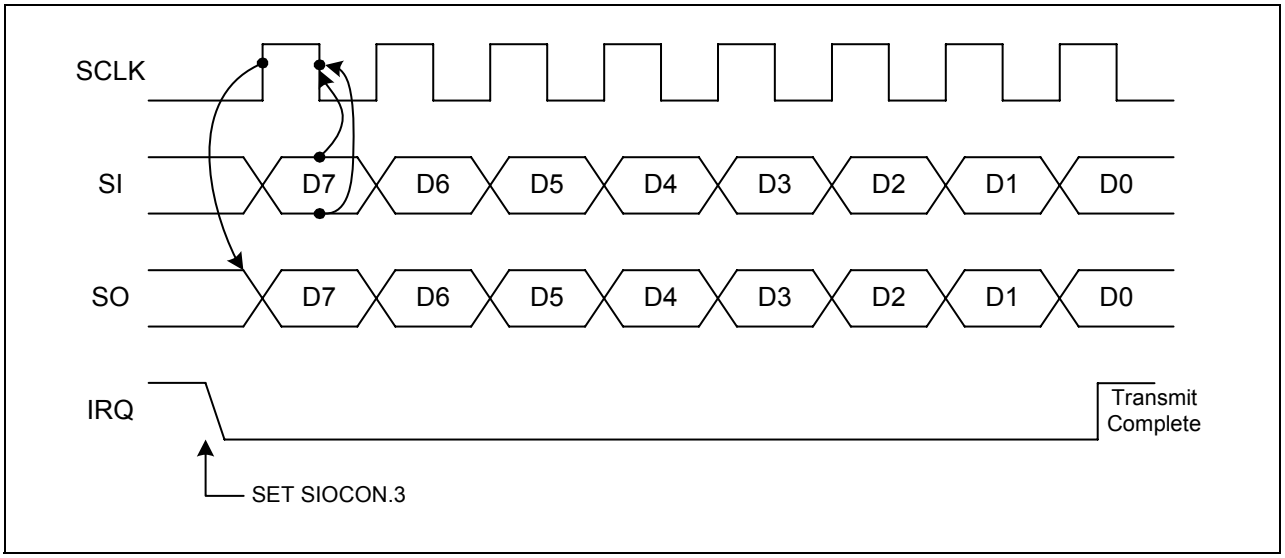
## SIO PRE-SCALER REGISTER (SIOPS)

The prescaler register for serial I/O interface module, SIOPS. The value stored in the SIO pre-scale register, SIOPS, lets you determine the SIO clock rate (baud rate) as follows:

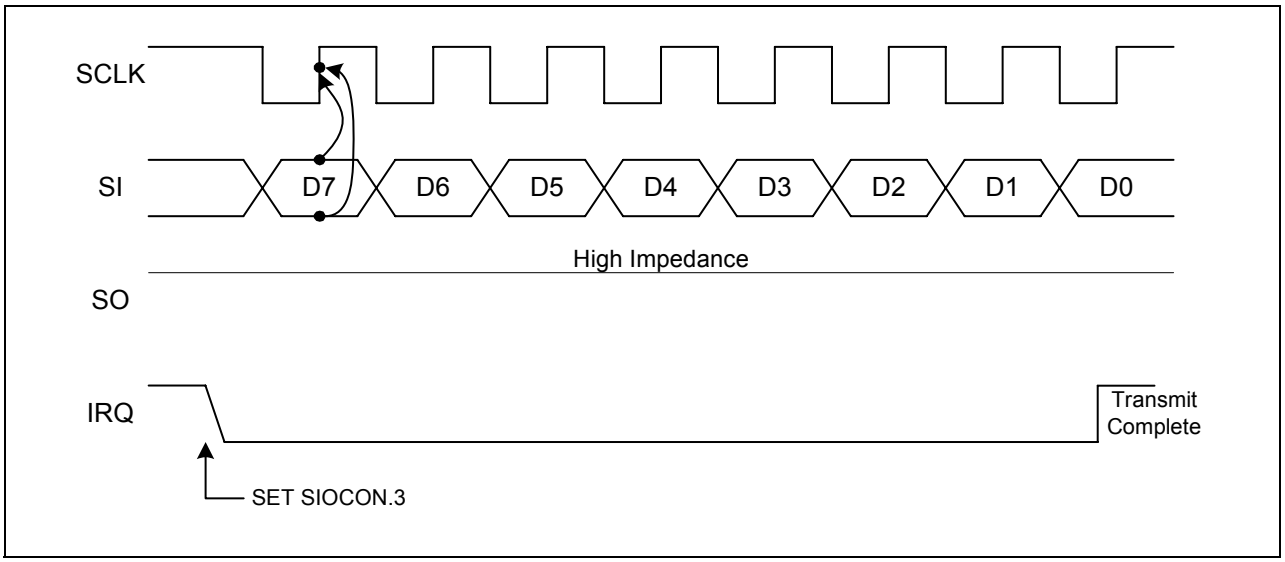
Baud rate = Input clock ($f_{SYS}$/4)/(Prescaler value + 1), or SCLK input clock.




< Figure 9-2. SIO Transmit/Receive Mode (Tx at falling edge) >




< Figure 9-3. SIO Transmit/Receive Mode (Tx at rising edge) >




< Figure 9-4. SIO Receive-Only Mode (Rising edge start) >

# 10. Electrical Characteristics

## 10-1. Absolute Maximum Ratings ($T_A$ = 25°C)

| Parameter | Rating | Unit |
|---|---|---|
| Supply voltage | − 0.3 to + 6.5 | V |
| Input voltage | − 0.3 to $V_{DD}$ + 0.3 | |
| Output voltage | − 0.3 to $V_{DD}$ + 0.3 | |
| Output current high per 1 PIN | − 25 | mA |
| Output current high per all PIN | − 80 | |
| Output current low per 1 PIN | + 30 | |
| Output current low per all PIN | + 150 | |
| Maximum Operating Voltage | 5.5 | V |
| Operating temperature | − 40 to + 85 | °C |
| Storage temperature | − 65 to + 150 | |

## 10-2. DC Characteristics ($T_A$ = −40°C to +85°C, $V_{DD}$ = 2.0V to 5.5V)

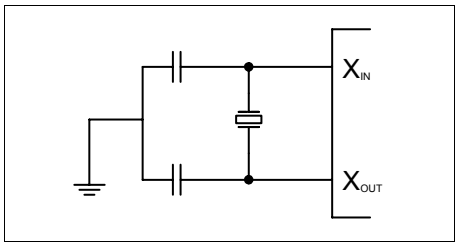
| Parameter | Symbol | Conditions | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Input High Voltage | $V_{IH1}$ | Except $X_{IN}$ , $X_{OUT}$ | $V_{DD}$ = 2.0 to 5.5 V | 0.8 $V_{DD}$ | − | $V_{DD}$ | V |
| | $V_{IH2}$ | $X_{IN}$ , $X_{OUT}$ | | $V_{DD}$ -0.5 | | | |
| Input Low Voltage | $V_{IL1}$ | Except $X_{IN}$ , $X_{OUT}$ | $V_{DD}$ = 2.0 to 5.5 V | − | − | 0.2 $V_{DD}$ | |
| | $V_{IL2}$ | $X_{IN}$ and $X_{OUT}$ | | | | 0.5 | |
| Output High Voltage | $V_{OH1}$ | PB.0-1, PD.4-6 | $V_{DD}$=2.4V; $I_{OH}$=-4mA | $V_{DD}$ -0.7 | $V_{DD}$ - 0.3 | − | |
| | $V_{OH2}$ | PC | $V_{DD}$=5V;$I_{OH}$=-4mA | $V_{DD}$ -1.0 | − | − | |
| | $V_{OH3}$ | Normal output pins | $V_{DD}$=5V;$I_{OH}$=-1mA | $V_{DD}$ -1.0 | − | − | |
| Output Low Voltage | $V_{OL1}$ | PB.0-1, PD.4-6 | $V_{DD}$=2.4V; $I_{OH}$=12mA | − | 0.3 | 0.5 | |
| | $V_{OL2}$ | PC | $V_{DD}$=5V;$I_{OH}$=15mA | − | 0.4 | 2.0 | |
| | $V_{OL3}$ | Normal output pins | $V_{DD}$=5V;$I_{OH}$=4mA | − | 0.4 | 2.0 | |
| Input Leakage Current(pin high) | $I_{ILH1}$ | Except $I_{ILH2}$ | $V_{IN}$ = $V_{DD}$ | − | − | 3 | μA |
| | $I_{ILH2}$ | $X_{IN}$, $XT_{IN}$ | $V_{IN}$ = $V_{DD}$ | | | 20 | |
| Input Leakage Current(pin low) | $I_{ILL1}$ | Except $I_{ILL2}$ | $V_{IN}$ = 0 V | − | − | −3 | |
| | $I_{ILL2}$ | $X_{IN}$, $XT_{IN}$ | $V_{IN}$ = 0 V | | | −20 | |
| Output Leakage Current(pin high) | $I_{OLH}$ | All output pins | $V_{OUT}$ = $V_{DD}$ | − | − | 3 | |
| Output Leakage Current(pin low) | $I_{OLL}$ | All output pins | $V_{OUT}$ = 0 V | − | − | −3 | |

| Parameter | Symbol | Conditions | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Pull-Up Resistor | $R_P$ | All Ports, $V_{IN}$ = 0 V, $V_{DD}$ = 5 V | | 25 | 50 | 100 | kΩ |
| LCD Voltage Dividing Resister | $R_{LCD}$ | – | | 40 | 75 | 100 | |
| COM output voltage deviation | $V_{COM}$ | ($V_{LC4}$ -COMn)<br>$I_O$= ± 15 p-μA | $V_{DD}$=$V_{LC4}$= 5 V | – | ± 45 | ± 90 | mV |
| SEG output voltage deviation | $V_{SEG}$ | ($V_{LC4}$ -SEGn)<br>$I_O$= ± 15 p-μA | | – | ± 45 | ± 90 | |
| LCD Voltage | $V_{LCD1}$ | VDD=2.7V to 5.5V, 1/5 bias<br>LCD clock=0Hz, $V_{LCD0}$=VDD | | 0.8VDD -0.2 | 0.8VDD | 0.8VDD +0.2 | V |
| | $V_{LCD2}$ | | | 0.6VDD -0.2 | 0.6VDD | 0.6VDD +0.2 | |
| | $V_{LCD3}$ | | | 0.4VDD -0.2 | 0.4VDD | 0.4VDD +0.2 | |
| | $V_{LCD4}$ | | | 0.2VDD -0.2 | 0.2VDD | 0.2VDD +0.2 | |
| Supply Current | $I_{DD1}$ | $V_{DD}$ = 5 V ± 10 %, 8 MHz | | – | 12 | 25 | mA |
| | | $V_{DD}$ = 5 V ± 10 %, 4 MHz | | | 4 | 10 | |
| | | $V_{DD}$ = 3 V ± 10 %, 8 MHz | | | 3 | 8 | |
| | | $V_{DD}$ = 3 V ± 10 %, 4 MHz | | | 1 | 5 | |
| | $I_{DD2}$ | Idle, $V_{DD}$ = 5 V ± 10 %, 8 MHz | | | 3 | 10 | |
| | | Idle, $V_{DD}$ = 5 V ± 10 %, 4 MHz | | | 1.5 | 4 | |
| | | Idle, $V_{DD}$ = 3 V ± 10 %, 8 MHz | | | 1.2 | 3 | |
| | | Idle, $V_{DD}$ = 3 V ± 10 %, 4 MHz | | | 1.0 | 2.0 | |
| | $I_{DD3}$ | Sub Clock (32768 Hz)<br>$V_{DD}$ = 3 V ± 10 % | | | 40 | 80 | μA |
| | $I_{DD4}$ | Idle Sub Clock (32768 Hz)<br>$V_{DD}$ = 3 V ± 10 % | | | 7 | 14 | |
| | $I_{DD5}$ | $V_{DD}$ = 5 V ± 10 %, STOP mode | | | 1 | 3 | |
| | | $V_{DD}$ = 3 V ± 10 %, STOP mode | | | 0.5 | 2 | |

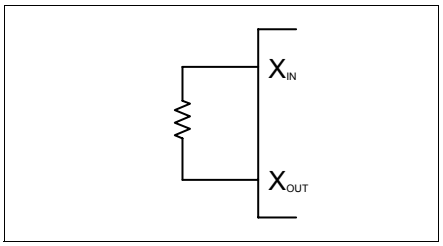
## 10-3. Clock Timing Constants ($T_A$ = − 40°C to + 85°C)

| Oscillator | Symbol | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| External Clock | $f_{OSC}$ | $V_{DD}$ = 2.5 to 5.5 V | 0.4 | − | 12 | MHz |
| | | $V_{DD}$ = 2.0 to 5.5 V | 0.4 | − | 4.2 | |
| External RC(NOTE-1) | $f_{OSC}$ | $V_{DD}$ = 5.0 V | 0.4 | − | 2 | |
| | | $V_{DD}$ = 3.0 V | 0.4 | − | 1 | |
| SUB Clock | $f_{SUB}$ | − | 32 | 32.768 | 35 | KHz |

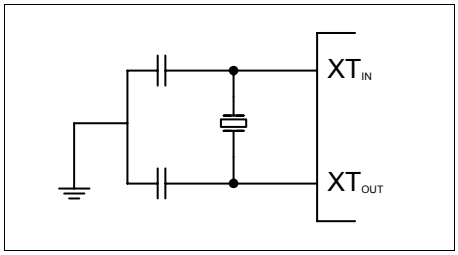
**NOTE**:
**1.** Tolerance: ±10 % at $T_A$ = 25°C




**External Oscillator Circuit (Crystal or Ceramic)**



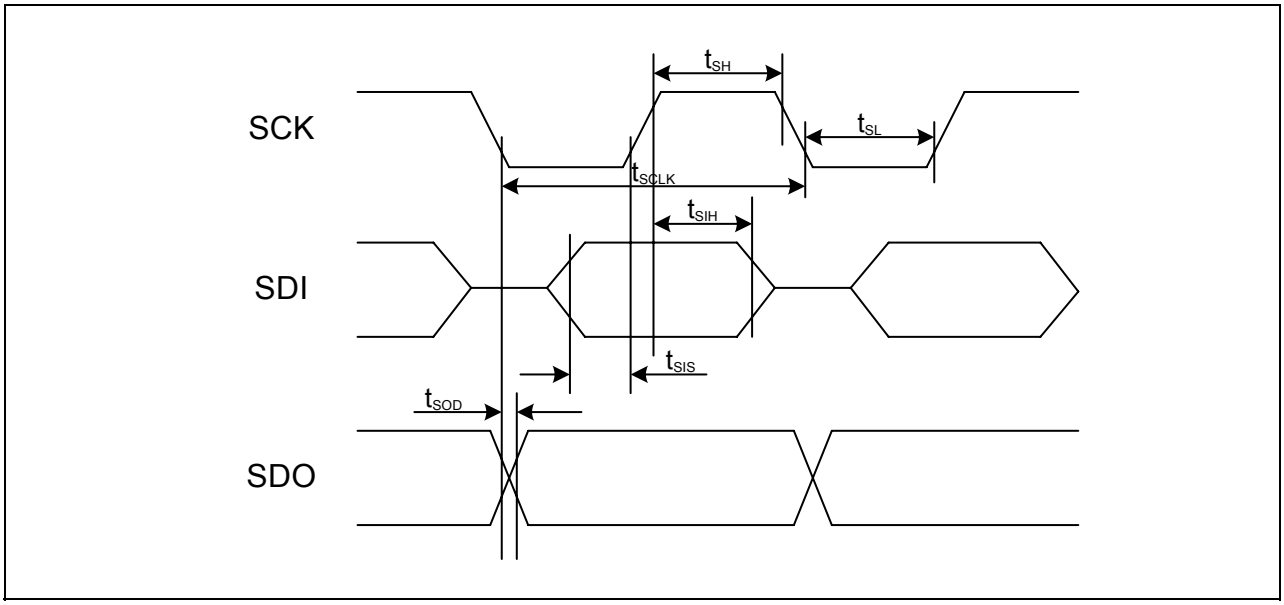

**External R-C Oscillator**




**Sub-Clock Oscillator Circuit (Crystal or Ceramic)**

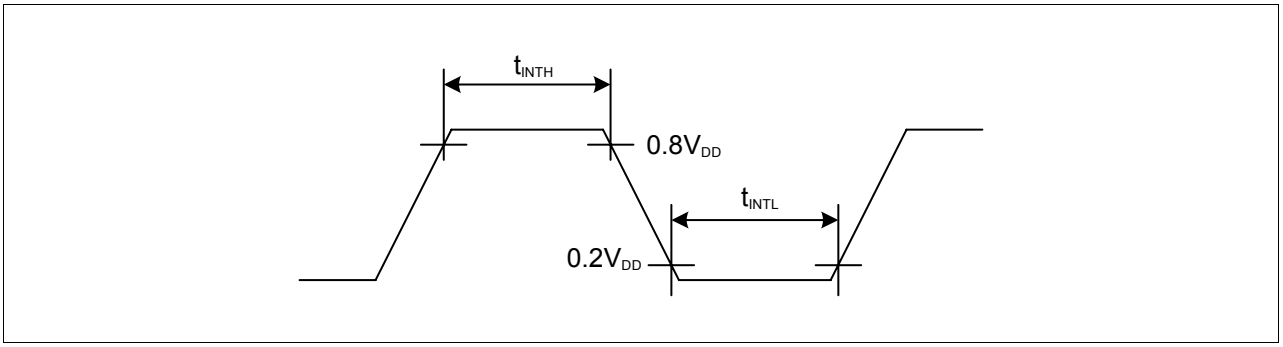
## 10-4. A.C Characteristics ($T_A$ = −40°C to +85°C, $V_{DD}$ = 2.0V to 5.5V)

| Parameter | Symbol | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| SCLK cycle time | $t_{SCLK}$ | External SCLK Source | 1,000 | – | – | ns |
| | | Internal SCLK Source | 1,000 | | | |
| SCLK high, low width | $t_{SH}$, $t_{SL}$ | External SCLK Source | 500 | | | |
| | | Internal SCLK Source | $T_{SCLK}$ /2-50 | | | |
| SI setup time SCLK high | $t_{SIS}$ | External SCLK Source | 250 | | | |
| | | Internal SCLK Source | 250 | | | |
| SI hold time SCLK high | $t_{SIH}$ | External SCLK Source | 400 | | | |
| | | Internal SCLK Source | 400 | | | |
| Output delay SCLK to SO | $t_{SOD}$ | External SCLK Source | – | – | 300 | |
| | | Internal SCLK Source | | | 250 | |



## 10-5. External Interrupt Characteristics (TA = −40°C to +85°C, VDD = 2.0V to 5.5V)

| Parameter | Symbol | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| External Interrupt Input High Voltage | $V_{EIH}$ | − | 0.8 $V_{DD}$ | − | $V_{DD}$ | V |
| External Interrupt Input Low Voltage | $V_{EIL}$ | − | − | − | 0.2 $V_{DD}$ | V |
| External Interrupt Input Width | $t_{INTH}$<br>$t_{INTL}$ | $V_{DD}$ = 5 V ± 10 % | − | 200 | − | ns |




## 10-6. Reset Timing Characteristics ($T_A$ = −40°C to +85°C, $V_{DD}$ = 2.0V to 5.5V)

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Input High Voltage | − | 0.8 $V_{DD}$ | − | $V_{DD}$ | V |
| Input Low Voltage | − | − | − | 0.2 $V_{DD}$ | V |
| Reset Input Low Width | Input $V_{DD}$ = 5 V ± 10 % | − | 2 | − | μs |

## 10-7. LVD Circuit Characteristics ($T_A$ = −40°C, $V_{DD}$ = 2.0V to 5.5V)

| Parameter | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| LVD reference Voltage | $V_{LVD}$ | − | 2.3<br>3.0<br>3.9 | − | V |
| LVD Hysteresis Voltage | $V_{HYST}$ | − | ±0.3 | − | V |
| Low Voltage Detection time | $t_{LVD}$ | 1 | − | − | μs |

## 10-8. A/D Converter Electrical Characteristics ($T_A$ = −40°C to +85°C, $V_{DD}$ = 2.0V to 5.5V, $V_{SS}$ = 0V)

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Total Accuracy | $V_{DD}$ = 5.12 V, $V_{SS}$ = 0 V<br>CPU clock = 10 MHz | − | − | ± 3 | LSB |
| Integral Non-Linearity | | − | − | ± 2 | |
| Differential Non-Linearity | | − | − | ± 1 | |
| Offset Error of Top | | − | ± 1 | ± 3 | |
| Offset Error of Bottom | | − | ± 1 | ± 2 | |
| Max Input Clock ($f_{ADC}$) | − | − | − | 4 | MHz |
| Conversion Time (NOTE-1) | $f_{ADC}$ = 4 MHz | − | 20 | − | μs |
| Analog Input Voltage | − | $V_{SS}$ | − | $V_{DD}$ | V |
| Analog Input Impedance | − | 2 | − | − | MΩ |
| Analog Input Current | $V_{DD}$ = 5 V | − | − | 10 | μA |
| Analog Block Current (Note-2) | $V_{DD}$ = 5 V | − | 1 | 3 | mA |
| | $V_{DD}$ = 3 V | − | 0.5 | 1.5 | mA |
| | $V_{DD}$ = 5 V stop mode | − | 100 | 500 | nA |

**NOTE**:
1. "Conversion time" is the time required from the moment a conversion operation starts until it ends.
2. $I_{ADC}$ is operating current during A/D conversion.

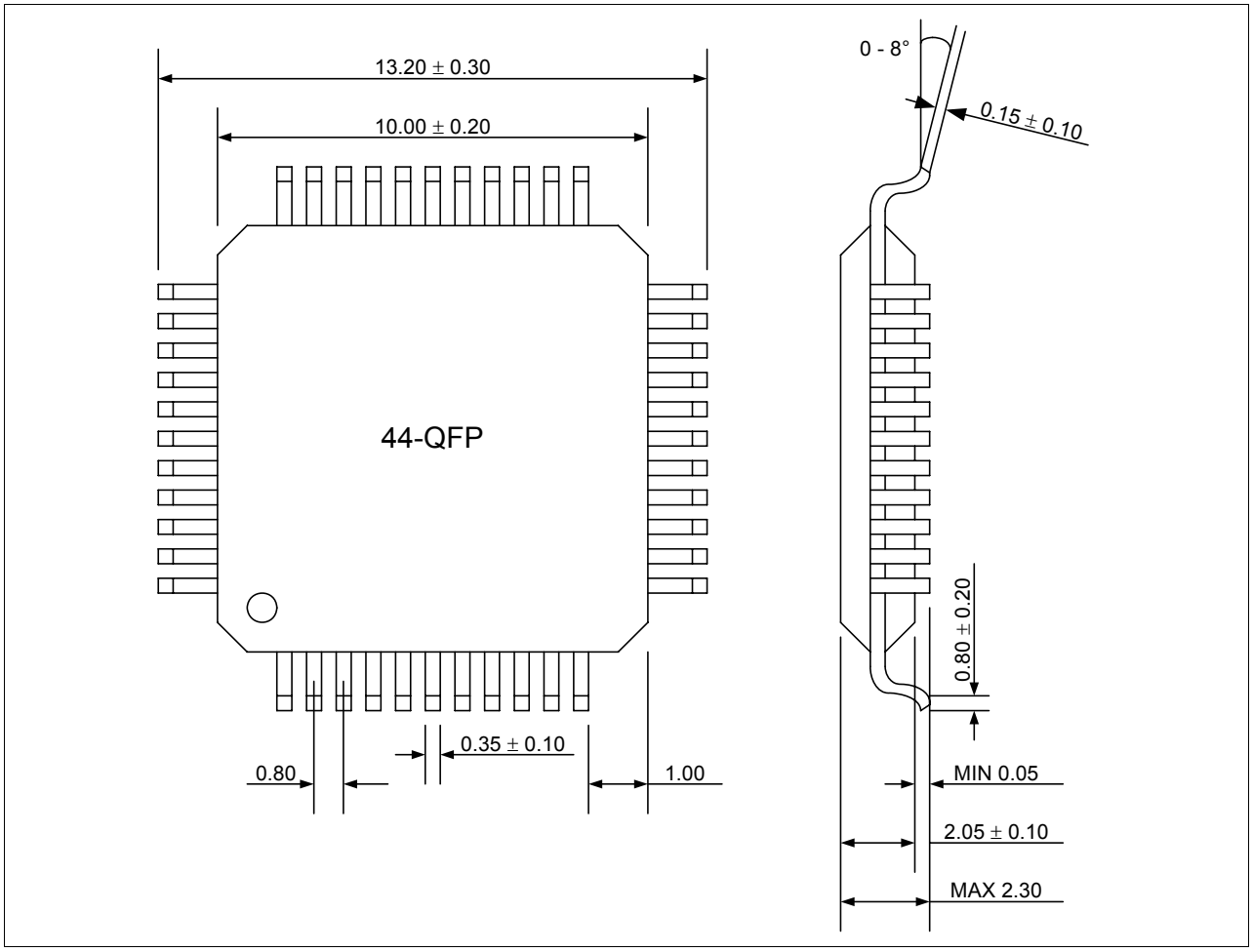
# 11. Packaging Information

The TM59PA80 order information: “IC Type” “XX” “YY” “C” “Z”.

- “IC TYPE”: TM59PA80
- “XX”: Package Type
  - **(1).** QFP Code: QF
  - **(2).** DIP (SKINNY) Code: SD
  - **(3).** LQFP Code: LQ
  - **(4).** DIP Code: DP
- “YY”: IC Pin Number
  - **(1).** Pin Number: 44 Code: 44
  - **(2).** Pin Number: 42 Code: 42
  - **(3).** Pin Number: 40 Code: 40
- “C”: Reserve (Must write be “C”)
- “Z”: Package material
  - **(1).** Package material: Pb-free Code: W
  - **(2).** Package material: Green Package Code: G

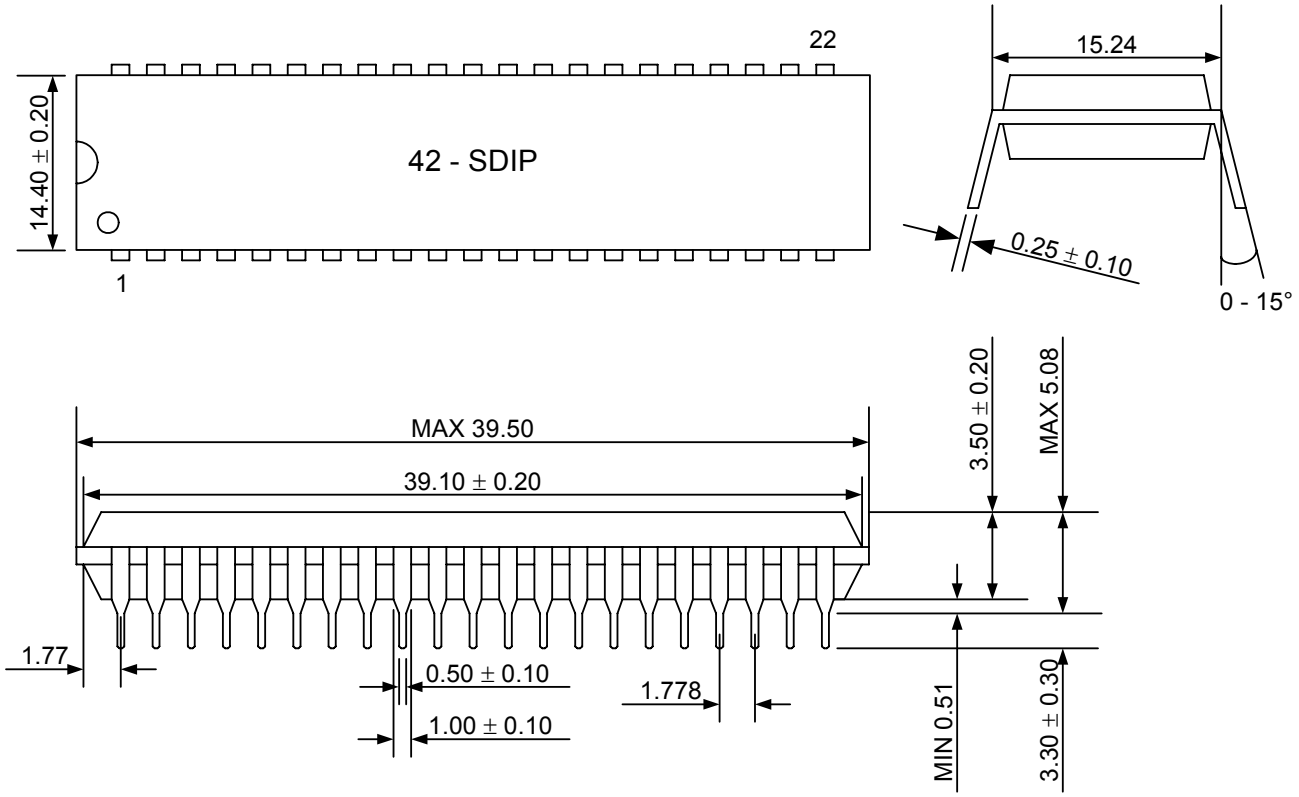
## 11-1. 44-QFP Package Dimension

44 lead, Quad Flat Package
Dimension in Millimeters



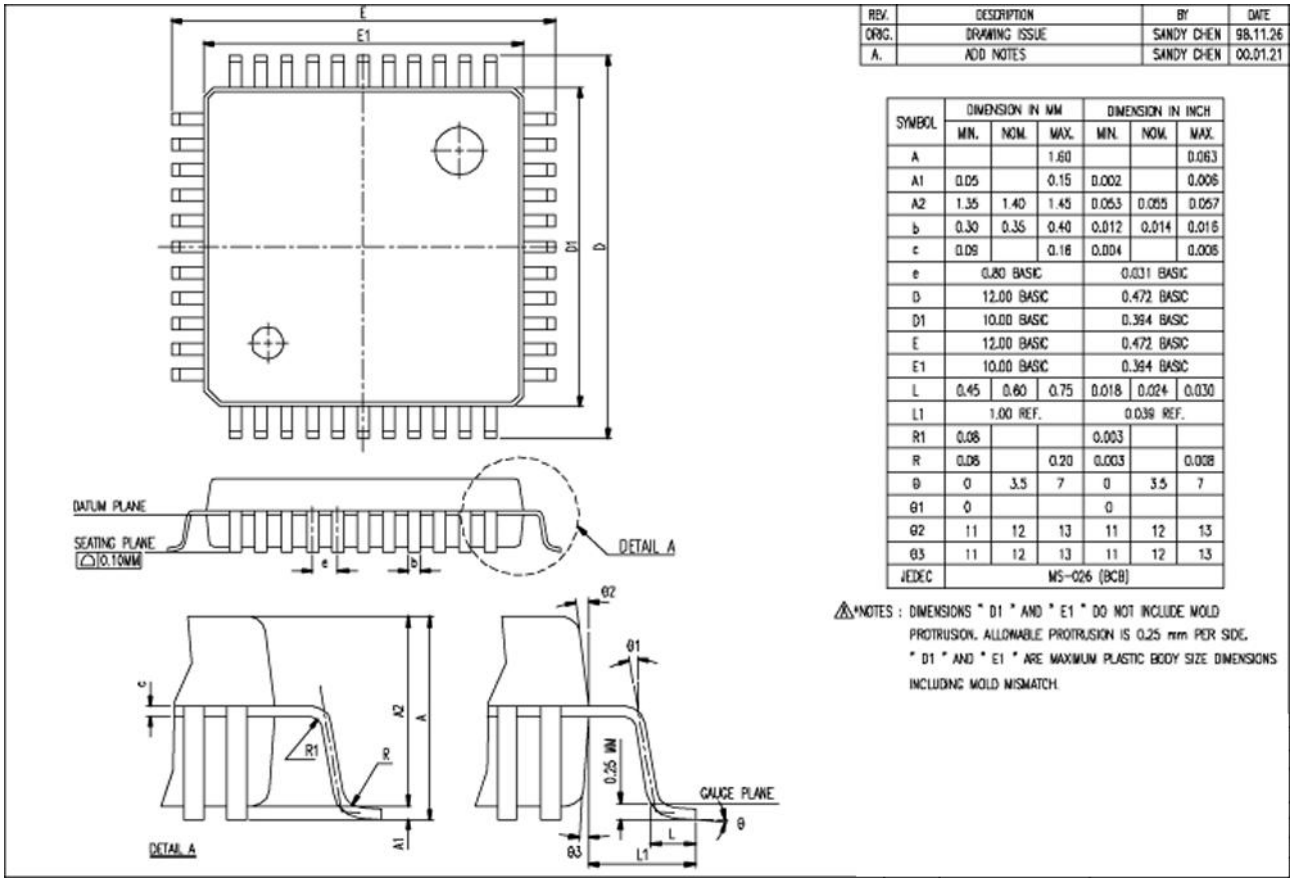
## 11-2. 42-SDIP Package Dimension

42 lead, Shrink Dual In-line Package
Dimension in Millimeters



## 11-3. 44-LQFP Package Dimension

44 lead, Shrink Dual In-line Package
Dimension in Millimeters



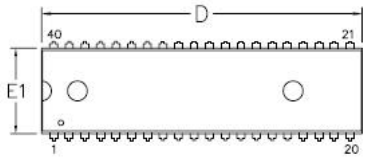

| REV. | DESCRIPTION | BY | DATE |
|---|---|---|---|
| ORG. | DRAWING ISSUE | SANDY CHEN | 98.11.26 |
| A. | ADD NOTES | SANDY CHEN | 00.01.21 |

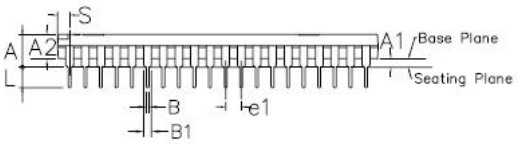
| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | | | 1.60 | | | 0.063 |
| A1 | 0.05 | | 0.15 | 0.002 | | 0.006 |
| A2 | 1.35 | 1.40 | 1.45 | 0.053 | 0.055 | 0.057 |
| b | 0.30 | 0.35 | 0.40 | 0.012 | 0.014 | 0.016 |
| c | 0.09 | | 0.16 | 0.004 | | 0.006 |
| e | 0.80 BASIC | | | 0.031 BASIC | | |
| 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 | | |
| L | 0.45 | 0.60 | 0.75 | 0.018 | 0.024 | 0.030 |
| L1 | 1.00 REF. | | | 0.039 REF. | | |
| R1 | 0.08 | | | 0.003 | | |
| R | 0.08 | | 0.20 | 0.003 | | 0.008 |
| θ | 0 | 3.5 | 7 | 0 | 3.5 | 7 |
| θ1 | 0 | | | 0 | | |
| θ2 | 11 | 12 | 13 | 11 | 12 | 13 |
| θ3 | 11 | 12 | 13 | 11 | 12 | 13 |
| JEDEC | MS-026 (BCB) | | | | | |

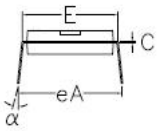
⚠*NOTES : DIMENSIONS " D1 " AND " E1 " DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. " D1 " AND " E1 " ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMATCH.

## 11-4. 40-DIP Package Dimension

40 lead, Shrink Dual In-line Package
Dimension in Millimeters




| Symbol | Dimension in inch | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min | Nom | Max | Min | Nom | Max |
| A | — | — | 0.210 | — | — | 5.33 |
| A1 | 0.010 | — | — | 0.25 | — | — |
| A2 | 0.150 | 0.155 | 0.160 | 3.81 | 3.94 | 4.06 |
| B | 0.016 | 0.018 | 0.022 | 0.41 | 0.46 | 0.56 |
| B1 | 0.048 | 0.050 | 0.054 | 1.22 | 1.27 | 1.37 |
| c | 0.008 | 0.010 | 0.014 | 0.20 | 0.25 | 0.36 |
| D | — | 2.055 | 2.070 | — | 52.20 | 52.58 |
| E | 0.590 | 0.600 | 0.610 | 14.99 | 15.24 | 15.49 |
| E1 | 0.540 | 0.545 | 0.552 | 13.72 | 13.84 | 14.02 |
| e1 | 0.090 | 0.100 | 0.110 | 2.29 | 2.54 | 2.79 |
| L | 0.120 | 0.130 | 0.140 | 3.05 | 3.30 | 3.56 |
| α | 0° | — | 15° | 0° | — | 15° |
| eA | 0.630 | 0.650 | 0.670 | 16.00 | 16.51 | 17.02 |
| S | — | — | 0.090 | — | — | 2.29 |

Note:

1. Dimension D Max & S include mold flash or tie bar burrs.
2. Dimension E1 does not include interlead flash.
3. Dimension D & E1 include mold mismatch and are determined at the mold parting line.
4. Dimension B1 does not include dambar protrusion/ intrusion.
5. Controlling dimension: Inch.
6. General appearance spec. should be based on final visual inspection spec.