



十速科技股份有限公司
tenx technology inc.

**Advance
Information**

TM57PE11

8 Bit Microcontroller

User Manual

**Tenx reserves the right to change or
discontinue this product without notice.**

tenx technology inc.

tenx technology, inc.

Preliminary

Rev 1.4, 2010/01/05

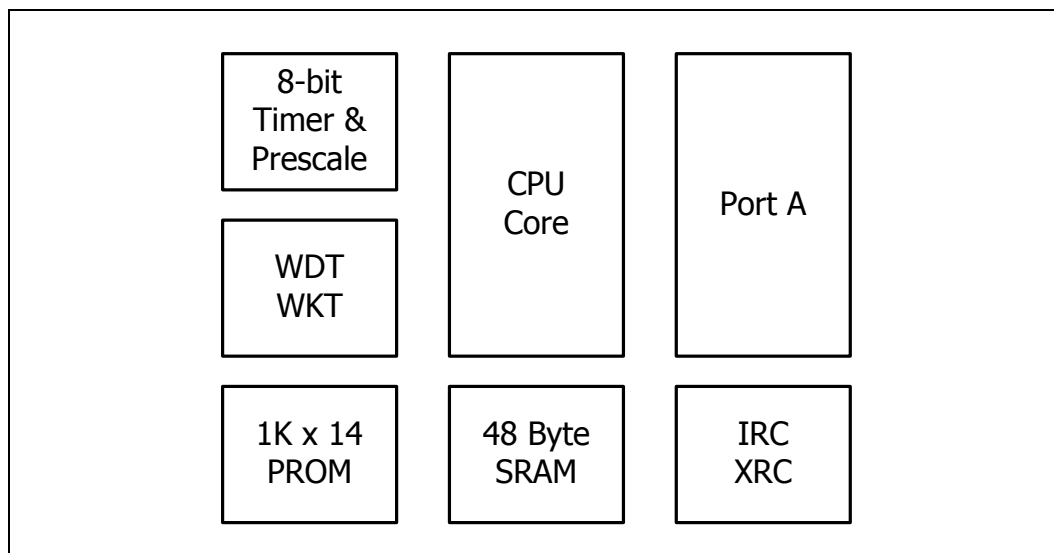
CONTENTS

FEATURES	2
BLOCK DIAGRAM	2
FUNCTIONAL DESCRIPTION.....	4
1. CPU Core	4
1.1 Clock Scheme and Instruction Cycle	4
1.2 Addressing Mode	4
1.3 Programming Counter (PC) and Stack	5
1.4 ALU and Working (W) Register	5
1.5 STATUS Register	5
1.6 Interrupt	6
2. Chip Operation Mode	7
2.1 Reset	7
2.2 System Configuration Register (SYSCFG).....	7
2.3 PROM Re-use	8
2.4 Power-Down Mode	8
3. Peripheral Functional Block	9
3.1 Watchdog(WDT) / Wakeup(WKT) Timer	9
3.2 8-bit Timer/Counter (Timer0) with Pre-scale (PSC)	9
3.3 System Clock Oscillator	10
4. I/O Port.....	11
4.1 PA0-1	11
4.2 PA2-4	12
4.3 PA7	12
MEMORY MAP	13
F-Plane.....	13
R-Plane	14
INSTRUCTION SET	15
ELECTRICAL CHARACTERISTICS	27
1. Absolute Maximum Ratings	27
2. DC Characteristics.....	27
3. Clock Timing	28
4. Reset Timing Characteristics	28
5. LVR Circuit Characteristics	29

FEATURES

1. ROM: 1K x 14 bits OTP or 512 x 14 bits TTP™ (Two Time Programmable ROM)
2. RAM: 48 x 8 bits
3. STACK: 5 Levels
4. I/O ports: One Bit programmable I/O ports (Max. 6 pins)
5. Timer/Counter: One 8-bit timer/counter with divided by 1~256 pre-scale option
6. Watchdog/Wakeup Timer: On chip Timer based on internal RC oscillator, 13~110mS wakeup time
7. Reset: Power On Reset, Watchdog Reset, Low Voltage Reset, External pin Reset
8. System Clock Mode:
 - Internal RC: 4MHz
 - External RC
9. Operation Voltage: Low Voltage Reset Level to 5.5V
10. Instruction set: 36 Instructions
11. Interrupts: Two pin interrupts, Timer0 interrupt and Wakeup Timer interrupt
12. Power Down mode support

BLOCK DIAGRAM



PIN DESCRIPTION

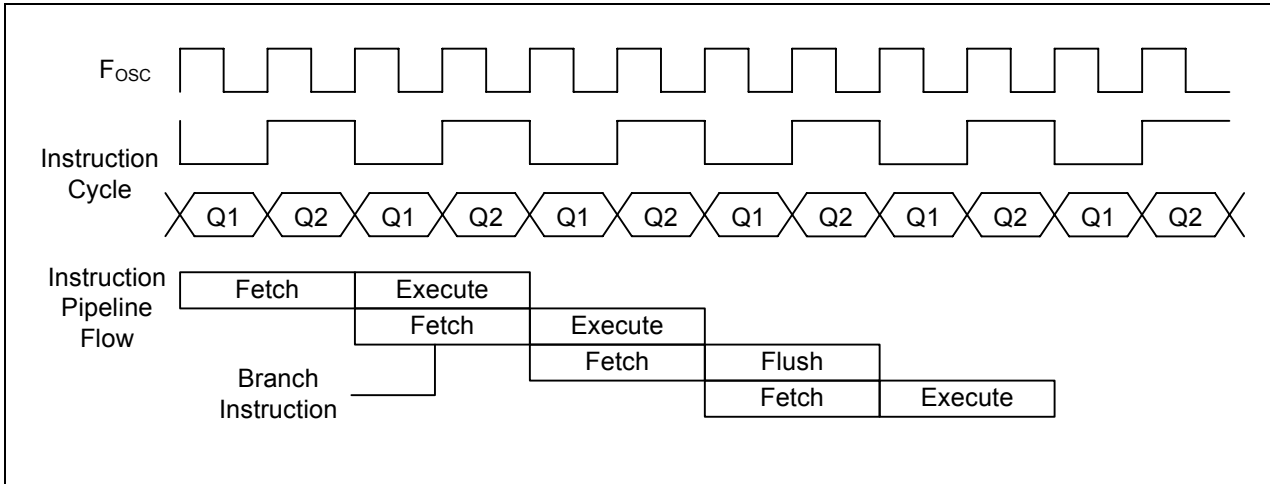
Name	In/Out	Pin Description
PA0–PA1	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or “pseudo-open-drain” output. Pull-up resistors are assignable by software.
PA2–PA4	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or open drain output. Pull-up resistors are assignable by software.
PA7	I	Schmitt-trigger input
nRESET	I	External active low reset
Xrc		External RC oscillator connection for system clock
CLKO	O	CPU Instruction clock output for external/internal RC mode
VDD, VSS	P	Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0,INT2	I	External interrupt input
T0I	I	Clock input to Timer0

FUNCTIONAL DESCRIPTION

1. CPU Core

1.1 Clock Scheme and Instruction Cycle

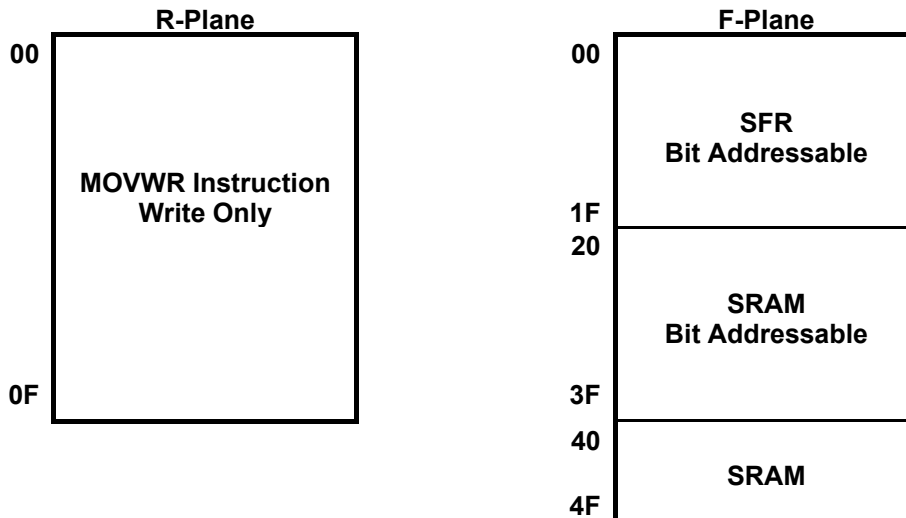
The system clock is internally divided by two to generate Q1 state and Q2 state for each instruction cycle. The Programming Counter (PC) is updated at Q1 and the instruction is fetched from program ROM and latched into the instruction register in Q2. It is then decoded and executed during the following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



1.2 Addressing Mode

There are two Data Memory Planes in CPU, R-Plane and F-Plane. The registers in R-Plane are write-only. The “MOVWR” instruction copy the W-register’s content to R-Plane registers by direct addressing mode.

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



1.3 Programming Counter (PC) and Stack

The Programming Counter is 10-bit wide capable of addressing a 1K 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 10 bits address from instruction word. For RET/RETI/RETLW instructions, PC retrieves its content from the top level STACK. For the other instructions updating PC [7:0], the PC [9:8] keeps unchanged. The STACK is 10-bit wide and 5-level in depth. The CALL instruction and Hardware interrupt will push STACK level in order. While the RET/RETI/RETLW instruction pops the STACK level in order.

1.4 ALU and Working (W) Register

The ALU is 8 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 STATUS register. The C and DC flags operate as a /Borrow and /Digit Borrow, respectively, in subtraction.

1.5 STATUS Register

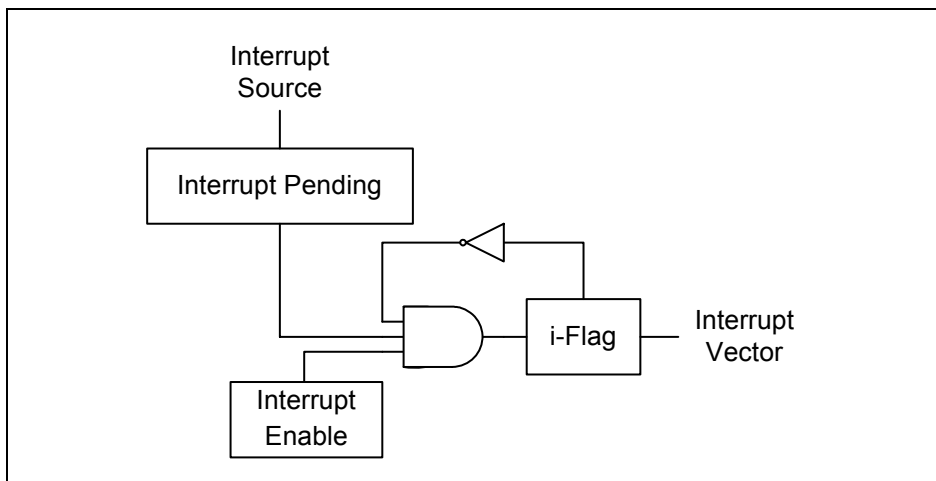
This register contains the arithmetic status of ALU and the Reset status. The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled.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

STATUS	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset Value	–	–	–	–	–	0	0	0
R/W	–	–	–	R	R	R/W	R/W	R/W
Bit	Description							
7-5	Not Used							
4	TO: Time Out 0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instruction 1: WDT time out occurred							
3	PD: Power Down 0: after Power On Reset, LVR Reset, or CLRWDT instruction 1: after SLEEP instruction							
2	Z: Zero Flag 0: the result of a logic operation is not zero 1: the result of a logic operation is zero							
1	DC: Decimal Carry Flag or Decimal/Borrow Flag							
	ADD instruction				SUB instruction			
	1: a carry from the low nibble bits of the result occurred 0: no carry				1: no borrow 0: a borrow from the low nibble bits of the result occurred			
0	C: Carry Flag or Borrow Flag							
	ADD instruction				SUB instruction			
	1: a carry occurred from the MSB 0: no carry				1: no borrow 0: a borrow occurred from the MSB			

1.6 Interrupt

The TM57PE11 has 1 level, 1 vector and five interrupt sources. Each interrupt source has its own enable control bit. An interrupt event will set its individual pending flag, no matter its interrupt enable control bit is 0 or 1. Because TM57PE11 has only 1 vector, there is not an interrupt priority register. The interrupt priority is determined by F/W.

If the corresponding interrupt enable bit has been set (INTE), it would trigger CPU to service the interrupt. CPU accepts interrupt in the end of current executed instruction cycle. In the mean while, A "CALL 001" instruction is inserted to CPU, and i-flag is set to prevent recursive interrupt nesting. The i-flag is cleared in the instruction after the "RETI" instruction. That is, at least one instruction in main program is executed before service the pending interrupt. The interrupt event is level triggered. F/W must clear the interrupt event register while serves the interrupt routine.



2. Chip Operation Mode

2.1 Reset

The TM57PE11 can be RESET in four ways.

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

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

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

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

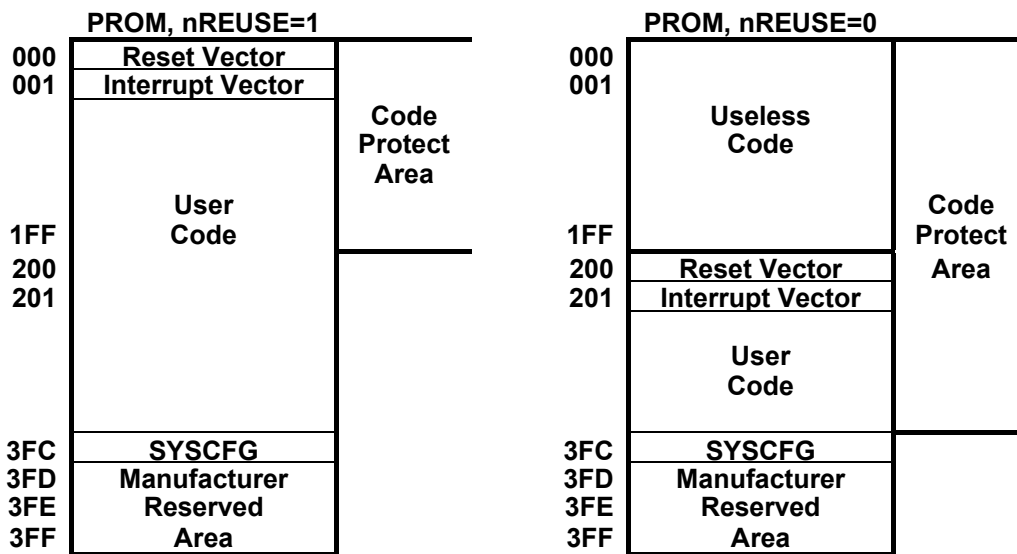
2.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at ROM address 3FCh. The SYSCFG determines the option for initial condition of MCU. It is written by PROM Writer only. User can select clock source, LVR threshold voltage and chip operation mode by SYSCFG register. The default value of SYSCFG is 3FFFh. The 13th bit of SYSCFG is code protection selection bit. If this bit is 0, the data in PROM will be protected, when user read PROM.

Bit	13~0	
Default Value	11111111111111	
Bit	Description	
13	nPROTECT : Code protection selection	
	1	No protect
	0	Code protection
12	nREUSE : PROM Re-use control	
	1	Not Re-use
	0	Re-use
11-10	LVR : LV reset mode	
	11	LVR threshold is 2.2V, always enable
	10	LVR threshold is 2.2V, disable in sleep mode
	01	LVR threshold is 3.2V, always enable
	00	LVR disable
9-8	CLKS : Clock Source Selection	
	1X	Invalid
	01	Internal RC (4MHz)
	00	External RC
7	XRESETE : External pin Reset Enable	
	1	Enable External pin Reset
	0	Disable External pin Reset
6	WDTE : WDT Reset Enable	
	1	Enable WDT Reset
	0	Disable WDT Reset
4-0	IRCF : Internal RC Frequency adjustment control	

2.3 PROM Re-use

The PROM of this device is 1K words. For some F/W program, the program size could be less than 512 words. To fully utilize the PROM, the device allows users to reuse the PROM. This feature is named as Two Time Programmable (TTP) ROM. While the first half of PROM is occupied by a useless program code and the second half of the PROM remains blank, users can re-write the PROM with the updated program code into the second half of the PROM. In the Re-use mode, the Reset Vector and Interrupt Vector are re-allocated at the beginning of the PROM's second half by the Assembly Compiler. Users simply choose the "REUSE" option in the ICE tool interface, then the Compiler will move the object code to proper location. That is, the user's program still has reset vector at address 000h, but the compiled object code has reset vector at 200h. In the SYSCFG, if nPROTECT=0 and nREUSE=1, the Code protection area is first half of PROM. This allows the Writer tool to write then verify the Code during the Re-use Code programming. After the Re-use Code being written into the PROM's second half, user should write "nREUSE" control bit to "0". In the mean while, the Code protection area becomes the whole PROM except the Reserved Area.



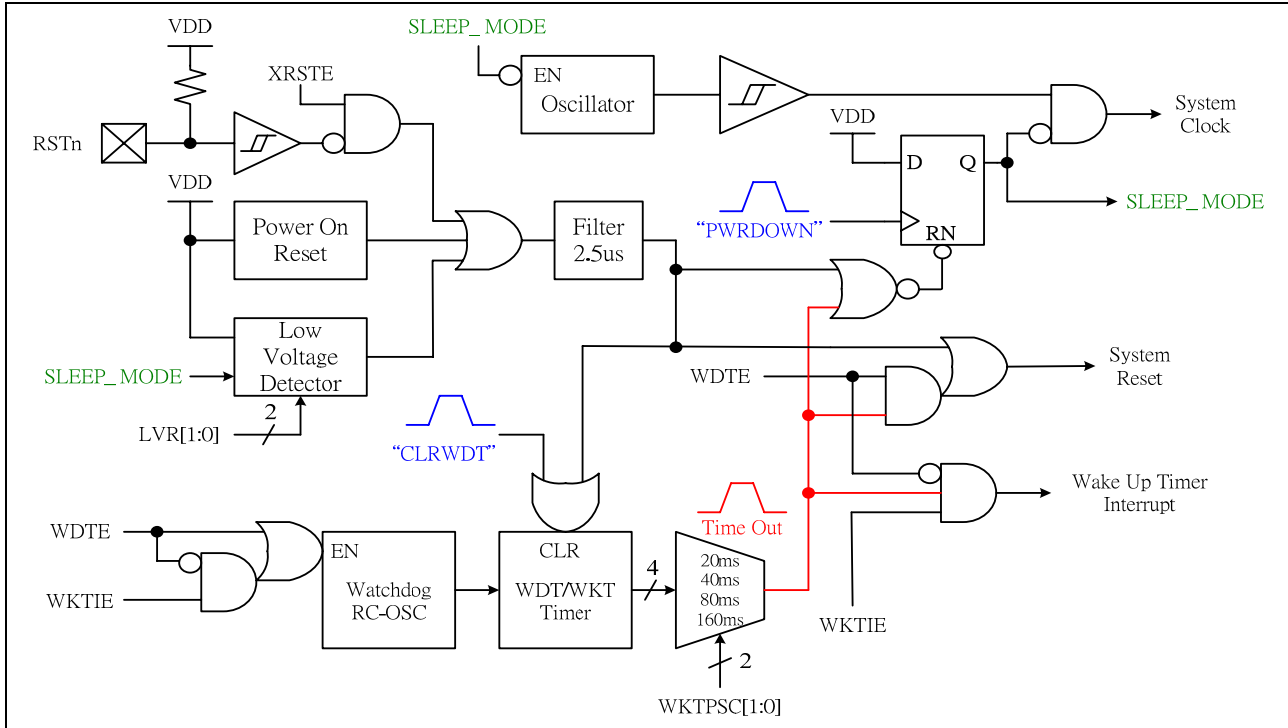
2.4 Power-Down Mode

The Power-down mode is activated by SLEEP instruction. During the Power-down mode, the system clock and peripherals stops to minimize power consumption, while the WDT/WKT Timer is working or not depends on F/W setting. The Power down mode can be terminated by Reset, or enabled Interrupts (External pin and WKT interrupts).

3. Peripheral Functional Block

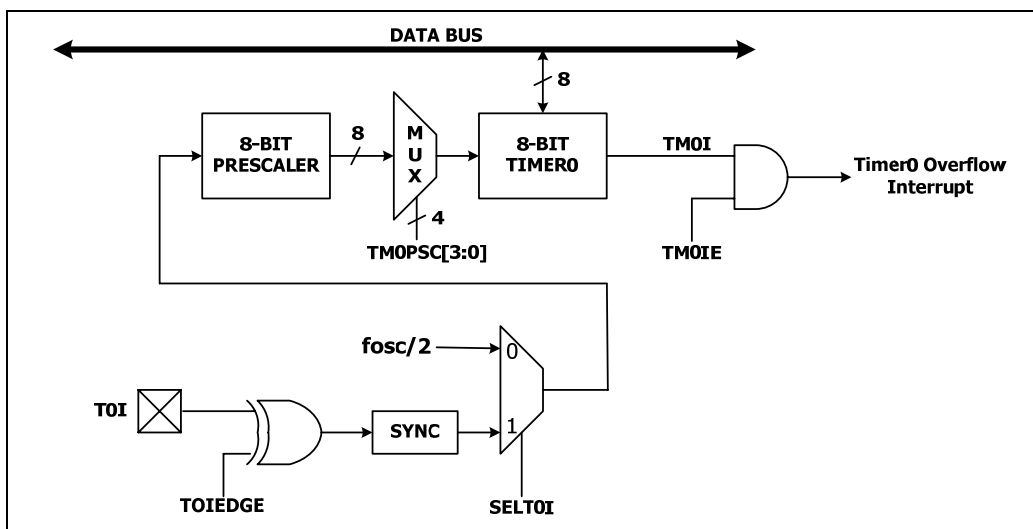
3.1 Watchdog(WDT) / Wakeup(WKT) Timer

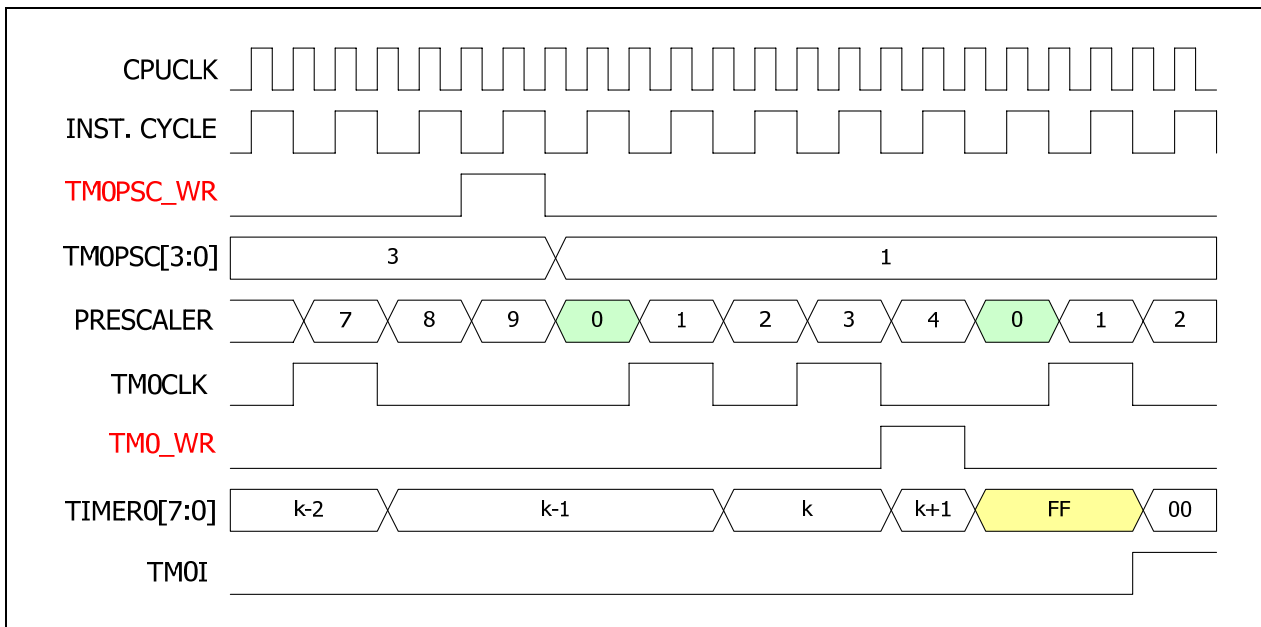
The WDT and WKT share the same internal RC Timer. The overflow period of WDT/WKT can be selected from 13mS to 110mS. The WDT/WKT is cleared by the CLRWDT instruction. If the Watchdog Reset is enabled (WDTE=1), the WDT generates the chip reset signal, otherwise, the WKT only generates overflow time out interrupt. The WDT/WKT works in both normal mode and sleep mode. If WDTE=0 and WKTIE=0 (Wakeup interrupt disable), the internal RC Timer stops for power saving.



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

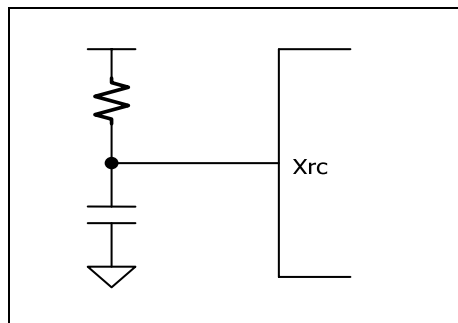
The Timer0 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatic roll over base on the pre-scaled clock source, which can be the instruction cycle or TOI input. The Timer0 increase rate is determined by "Timer0 Pre-Scale" (TM0PSC) register in R-Plane. The Timer0 can generate interrupt (TM0I) when it rolls over.





3.3 System Clock Oscillator

System Clock can be operated in two different oscillation modes, which is selected by setting the CLKS in the SYSCFG register. In external RC mode, the external resistor and capacitor determine the oscillation frequency. In the internal RC mode, the on chip oscillator generates 4MHz system clock.

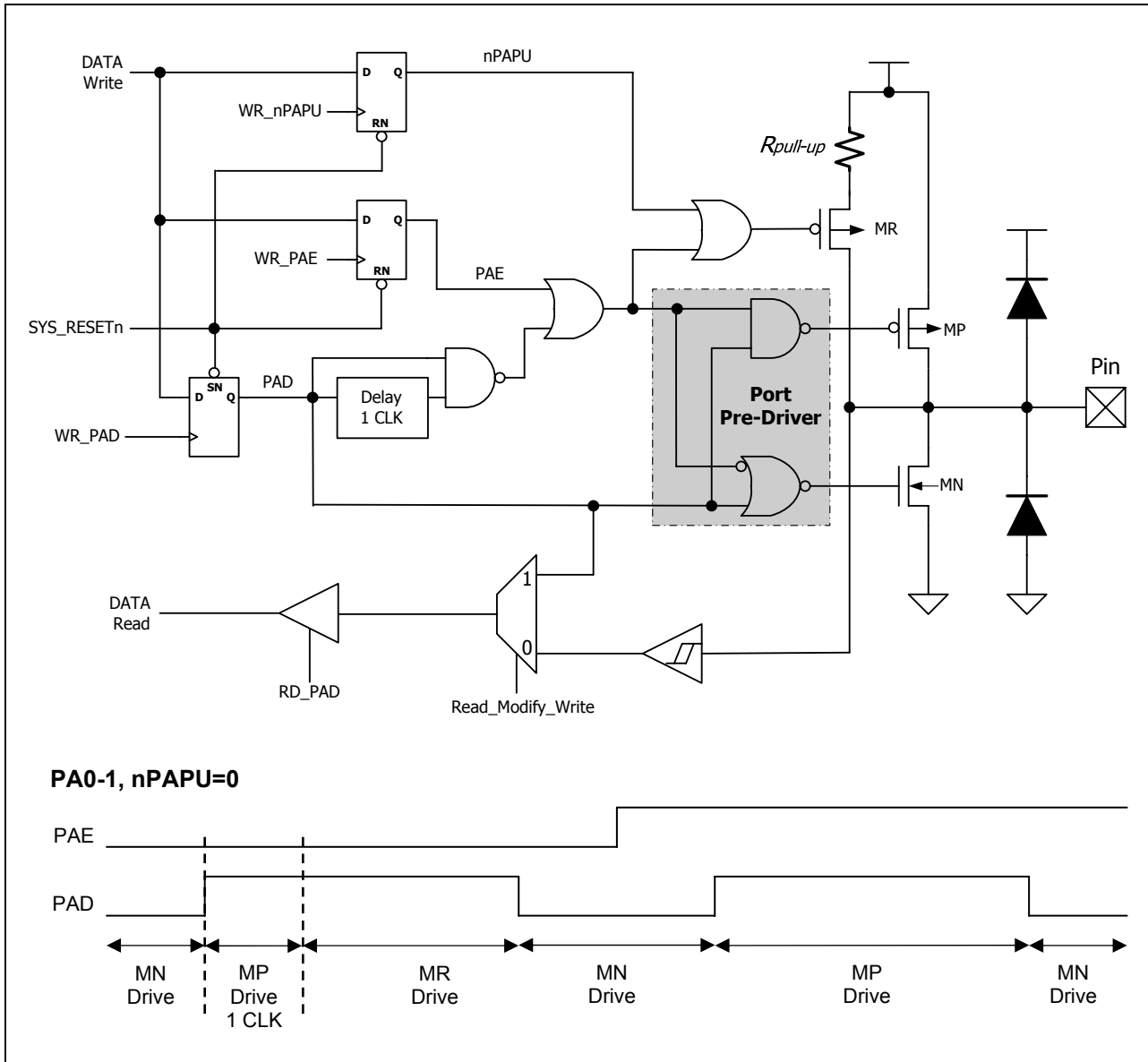


External RC Oscillator

4. I/O Port

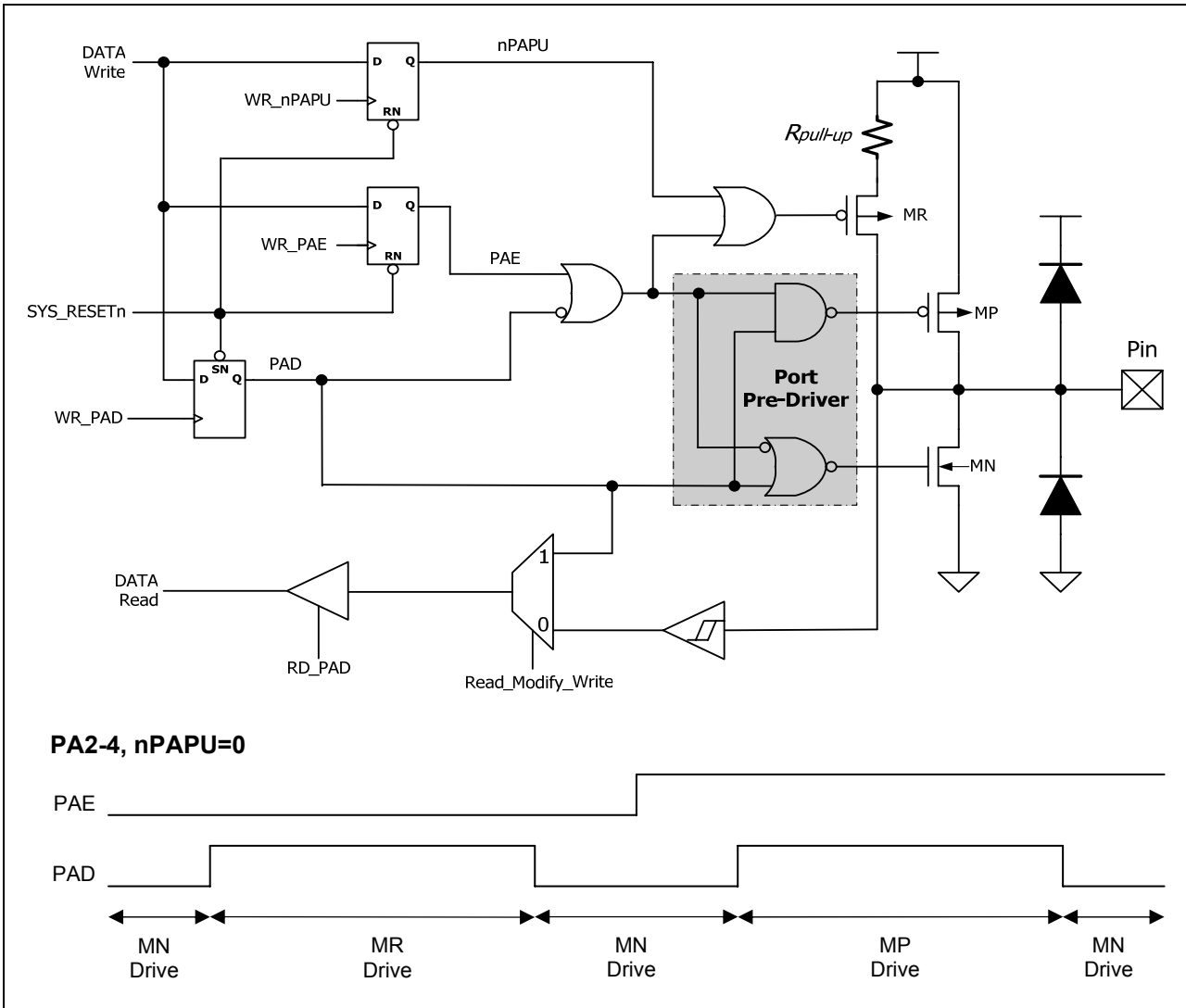
4.1 PA0-1

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



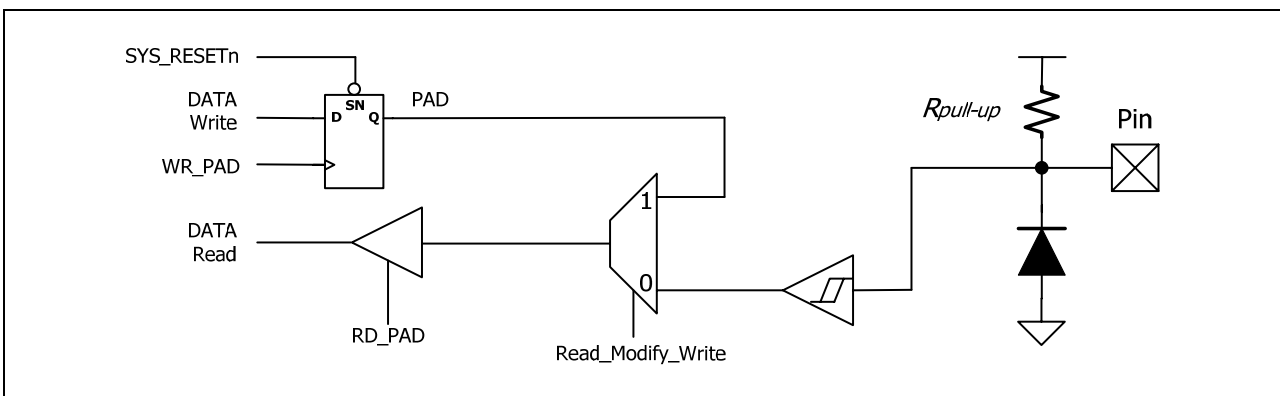
4.2 PA2-4

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



4.3 PA7

PA7 can be only used in Schmitt-trigger input mode. The pull-up resistor is always connected to this pin.



MEMORY MAP

F-Plane

Name	Address	R/W	Rst	Description
INDF	00.7~0	R/W	-	Not a physical register, addressing INDF actually point to the register whose address is contained in the FSR register
TIMER0	01.7~0	R/W	0	Timer0 content
PC	02.7~0	R/W	0	Programming Counter [7~0]
TO	03.4	R	0	WDT time out flag
PD	03.3	R	0	Sleep mode flag
ZFLAG	03.2	R/W	0	Zero Flag
DCFLAG	03.1	R/W	0	Decimal Carry Flag
CFLAG	03.0	R/W	0	Carry Flag
FSR	04.6~0	R/W	-	File Select Register, indirect address mode pointer
PAD7	05.7	R	-	PA7 pin state
PAD	05.4~0	R	-	Port A pin or "data register" state
		W	1F	Port A output data register
TM0IE	08.4	R/W	0	Timer0 interrupt enable, 1=enable, 0=disable
WKTIE	08.3	R/W	0	Wakeup Timer interrupt enable, 1=enable, 0=disable
XINT2E	08.2	R/W	0	INT2 pin interrupt enable, 1=enable, 0=disable
XINT0E	08.0	R/W	0	INT0 pin interrupt enable, 1=enable, 0=disable
TM0I	09.4	R	-	Timer0 interrupt event pending flag, set by H/W while Timer0 overflow
		W	0	write 0: clear this flag; write 1: no action
WKT I	09.3	R	-	WKT interrupt event pending flag, set by H/W while WKT time out
		W	0	write 0: clear this flag; write 1: no action
XINT2	09.2	R	-	INT2 pin interrupt event pending flag, set by H/W at INT2 pin's falling edge
		W	0	write 0: clear this flag; write 1: no action
XINT0	09.0	R	-	INT0 pin interrupt event pending flag, set by H/W at INT0 pin's falling or rising edge
		W	0	write 0: clear this flag; write 1: no action
SRAM	20~4F	R/W	-	Internal RAM

R-Plane

Name	Address	R/W	Rst	Discription
T0IEDGE	02.5	W	0	0: T0I rising edge to increase Timer0/PSC count 1: T0I falling edge to increase Timer0/PSC count
SELT0I	02.4	W	0	0: Timer0/PSC clock source is "Instruction Cycle" 1: Timer0/PSC clock source is T0I pin
TM0PSC	02.3~0	W	0	Timer0 Pre-Scale 0000: Timer0 input clock divided by 1 0001: Timer0 input clock divided by 2 0010: Timer0 input clock divided by 4 0011: Timer0 input clock divided by 8 0100: Timer0 input clock divided by 16 0101: Timer0 input clock divided by 32 0110: Timer0 input clock divided by 64 0111: Timer0 input clock divided by 128 1000: Timer0 input clock divided by 256
PWRDOWN	03	W		write this register to enter Power-Down Mode
CLRWDT	04	W		write this register to clear WDT/WKT
PAE	05.4~2	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
	05.1~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is pseudo-open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
nPAPU	08.4~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enable, except a. the pin's output data register (PAD) is 0 b. the pin's CMOS push-pull mode is chosen (PAE=1) c. the pin is working for Crystal or external RC oscillation 1: the pin pull up resistor is disable
INT0EDGE	0b.4	W	0	0: INT0 pin falling edge to trigger interrupt event 1: INT0 pin rising edge to trigger interrupt event
CLK2PIN	0b.3	W	0	0: No Instruction Clock output to PA3 pin 1: Instruction Clock output to PA3 pin for external/internal RC mode
WKT PSC	0b.1~0	W	11	WDT/WKT period (5v) 00: 13mS 01: 27mS 10: 55mS 11: 110mS
TESTREG	0f.1~0	W	00	Test mode register, for manufacturer only, user do not write it

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

Mnemonic		Op Code	Cycle	Flag Affect	Description
Byte-Oriented File Register Instruction					
ADDWF	f,d	00 0111 dfff ffff	1	C,DC,Z	Add W 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"
CLRWF		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"
MOVWF	f	00 1000 0fff ffff	1	-	Move "f" to W
MOVWF	f	00 0000 1fff ffff	1	-	Move W to "f"
MOVWR	r	00 0000 00rr rrrr	1	-	Move W to "r"
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 nibbles in "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"
BTFSZ	f,b	01 010b bbff ffff	1 or 2	-	Test "b" bit of "f", skip if clear
BTFSZ	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 00kk kkkk kkkk	2	-	Call subroutine "k"
CLRWDT		00 0000 0000 0100	1	-	Clear and enable Watch Dog Timer
GOTO	k	11 00kk 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 in W
SLEEP		00 0000 0000 0011	1	-	Go into standby mode, Clock oscillation stops
XORLW	k	01 1111 kkkk kkkk	1	Z	XOR Literal "k" with W

ADDLW Add Literal “k” and W

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

ADDWF Add W and “f”

Syntax	ADDWF f [,d]	
Operands	f : 00h ~ 4Fh d : 0, 1	
Operation	$(\text{Destination}) \leftarrow (W) + (f)$	
Status Affected	C, DC, Z	
OP-Code	00 0111 dfff ffff	
Description	Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	ADDWF FSR, 0	B : W = 0x17, FSR = 0xC2 A : W = 0xD9, FSR = 0xC2

ANDLW Logical AND Literal "k" with W

Syntax	ANDLW k	
Operands	k : 00h ~ FFh	
Operation	$(W) \leftarrow (W) \text{ 'AND' } (f)$	
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 A : W = 0x03

ANDWF AND W with “f”

Syntax	ANDWF f [,d]	
Operands	f : 00h ~ 4Fh d : 0, 1	
Operation	$(\text{Destination}) \leftarrow (W) \text{ 'AND' } (f)$	
Status Affected	Z	
OP-Code	00 0101 dfff ffff	
Description	AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	ANDWF FSR, 1	B : W = 0x17, FSR = 0xC2 A : W = 0x17, FSR = 0x02

BCF	Clear "b" bit of "f"
Syntax	BCF f [,b]
Operands	f : 00h ~ 3Fh b : 0 ~ 7
Operation	(f.b) ← 0
Status Affected	-
OP-Code	01 000b bbff ffff
Description	Bit 'b' in register 'f' is cleared.
Cycle	1
Example	BCF FLAG_REG, 7 B : FLAG_REG = 0xC7 A : FLAG_REG = 0x47
BSF	Set "b" bit of "f"
Syntax	BSF f [,b]
Operands	f : 00h ~ 3Fh b : 0 ~ 7
Operation	(f.b) ← 1
Status Affected	-
OP-Code	01 001b bbff ffff
Description	Bit 'b' in register 'f' is set.
Cycle	1
Example	BSF FLAG_REG, 7 B : FLAG_REG = 0x0A A : FLAG_REG = 0x8A
BTFSC	Test "b" bit of "f", skip if clear(0)
Syntax	BTFSC f [,b]
Operands	f : 00h ~ 3Fh b : 0 ~ 7
Operation	Skip next instruction if (f.b) = 0
Status Affected	-
OP-Code	01 010b bbff ffff
Description	If bit 'b' in register 'f' is '1', then the next instruction is executed. If bit 'b' in register 'f' is '0', then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.
Cycle	1 or 2
Example	LABEL1 BTFSC FLAG, 1 B : PC = LABEL1 TRUE GOTO SUB1 A : if FLAG.1 = 0, PC = FALSE FALSE ... if FLAG.1 = 1, PC = TRUE
BTFSS	Test "b" bit of "f", skip if set(1)
Syntax	BTFSS f [,b]
Operands	f : 00h ~ 3Fh b : 0 ~ 7
Operation	Skip next instruction if (f.b) = 1
Status Affected	-
OP-Code	01 011b bbff ffff
Description	If bit 'b' in register 'f' is '0', then the next instruction is executed. If bit 'b' in register 'f' is '1', then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.
Cycle	1 or 2
Example	LABEL1 BTFSS FLAG, 1 B : PC = LABEL1 TRUE GOTO SUB1 A : if FLAG.1 = 0, PC = TRUE FALSE ... if FLAG.1 = 1, PC = FALSE

COMF	Complement “f”
Syntax	COMF f [,d]
Operands	f : 00h ~ 4Fh, d : 0, 1
Operation	(destination) ← (\bar{f})
Status Affected	Z
OP-Code	00 1001 dfff ffff
Description	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.
Cycle	1
Example	COMF REG1,0 B : REG1 = 0x13 A : REG1 = 0x13, W = 0xEC
DECF	Decrement “f”
Syntax	DECF f [,d]
Operands	f : 00h ~ 4Fh, d : 0, 1
Operation	(destination) ← (f) - 1
Status Affected	Z
OP-Code	00 0011 dfff ffff
Description	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Cycle	1
Example	DECF CNT, 1 B : CNT = 0x01, Z = 0 A : CNT = 0x00, Z = 1
DECFSZ	Decrement “f”, Skip if 0
Syntax	DECFSZ f [,d]
Operands	f : 00h ~ 4Fh, 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 ~ 3FFh
Operation	PC.9~0 ← k
Status Affected	-
OP-Code	11 00kk kkkk kkkk
Description	GOTO is an unconditional branch. The 10-bit immediate value is loaded into PC bits <9:0>. GOTO is a two-cycle instruction.
Cycle	2
Example	LABEL1 GOTO SUB1 B : PC = LABEL1 A : PC = SUB1

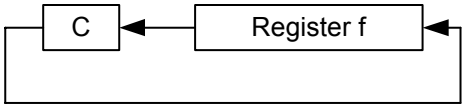
INCF	Increment “f”
Syntax	INCF f [,d]
Operands	f : 00h ~ 4Fh
Operation	(destination) ← (f) + 1
Status Affected	Z
OP-Code	00 1010 dfff ffff
Description	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register '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 ~ 4Fh, 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 B : PC = LABEL1 GOTO LOOP A : CNT = CNT + 1 CONTINUE if CNT=0, PC = CONTINUE if CNT≠0, PC = LABEL1+1

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

IORWF	Inclusive OR W with “f”
Syntax	IORWF f [,d]
Operands	f : 00h ~ 4Fh, 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 A : RESULT = 0x13, W = 0x93, Z = 0

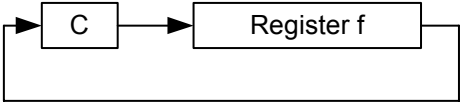
RLF Rotate Left f through Carry

Syntax RLF f [,d]
 Operands f : 00h ~ 4Fh, d : 0, 1
 Operation 

Status Affected C
 OP-Code 00 1101 dfff ffff
 Description The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.

Cycle 1
 Example RLF REG1,0
 B : REG1 = 1110 0110, C = 0
 A : REG1 = 1110 0110
 W = 1100 1100, C = 1

RRF Rotate Right "f" through Carry

Syntax RRF f [,d]
 Operands f : 00h ~ 4Fh, d : 0, 1
 Operation 

Status Affected C
 OP-Code 00 1100 dfff ffff
 Description The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.

Cycle 1
 Example RRF REG1,0
 B : REG1 = 1110 0110, C = 0
 A : REG1 = 1110 0110
 W = 0111 0011, C = 0

SLEEP Go into standby mode, Clock oscillation stops

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

SUBWF	Subtract W from “f”	
Syntax	SUBWF f [,d]	
Operands	f : 00h ~ 4Fh, d : 0, 1	
Operation	$(W) \leftarrow (f) - (W)$	
Status Affected	C, DC, Z	
OP-Code	00 0010 dfff ffff	
Description	Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	SUBWF REG1,1	B : REG1 = 3, W = 2, C = ?, Z = ? A : REG1 = 1, W = 2, C = 1, Z = 0
	SUBWF REG1,1	B : REG1 = 2, W = 2, C = ?, Z = ? A : REG1 = 0, W = 2, C = 1, Z = 1
	SUBWF REG1,1	B : REG1 = 1, W = 2, C = ?, Z = ? A : REG1 = FFh, W = 2, C = 0, Z = 0
SWAPF	Swap Nibbles in “f”	
Syntax	SWAPF f [,d]	
Operands	f : 00h ~ 4Fh, d : 0, 1	
Operation	$(\text{destination}, 7\sim 4) \leftarrow (f.3\sim 0), (\text{destination}.3\sim 0) \leftarrow (f.7\sim 4)$	
Status Affected	-	
OP-Code	00 1110 dfff ffff	
Description	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.	
Cycle	1	
Example	SWAPF REG, 0	B : REG1 = 0xA5 A : REG1 = 0xA5, W = 0x5A
TESTZ	Test if “f” is zero	
Syntax	TESTZ f	
Operands	f : 00h ~ 4Fh	
Operation	Set Z flag if (f) is 0	
Status Affected	Z	
OP-Code	00 1000 1fff ffff	
Description	If the content of register 'f' is 0, Zero flag is set to 1.	
Cycle	1	
Example	TESTZ REG1	B : REG1 = 0, Z = ? A : REG1 = 0, Z = 1

XORLW Exclusive OR Literal with W

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

XORWF Exclusive OR W with "f"

Syntax	XORWF f [,d]	
Operands	f : 00h ~ 4Fh, d : 0, 1	
Operation	(destination) ← (W) XOR (f)	
Status Affected	Z	
OP-Code	00 0110 dfff ffff	
Description	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	XORWF REG 1	B : REG = 0xAF, W = 0xB5 A : REG = 0x1A, W = 0xB5

ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings ($T_A = 25\text{ }^\circ\text{C}$)

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	- 25 to + 85	$^\circ\text{C}$
Storage temperature	- 65 to + 150	

2. DC Characteristics ($T_A = -25\text{ }^\circ\text{C}$ to $+85\text{ }^\circ\text{C}$, $V_{DD} = 2.0\text{ V}$ to 5.5 V)

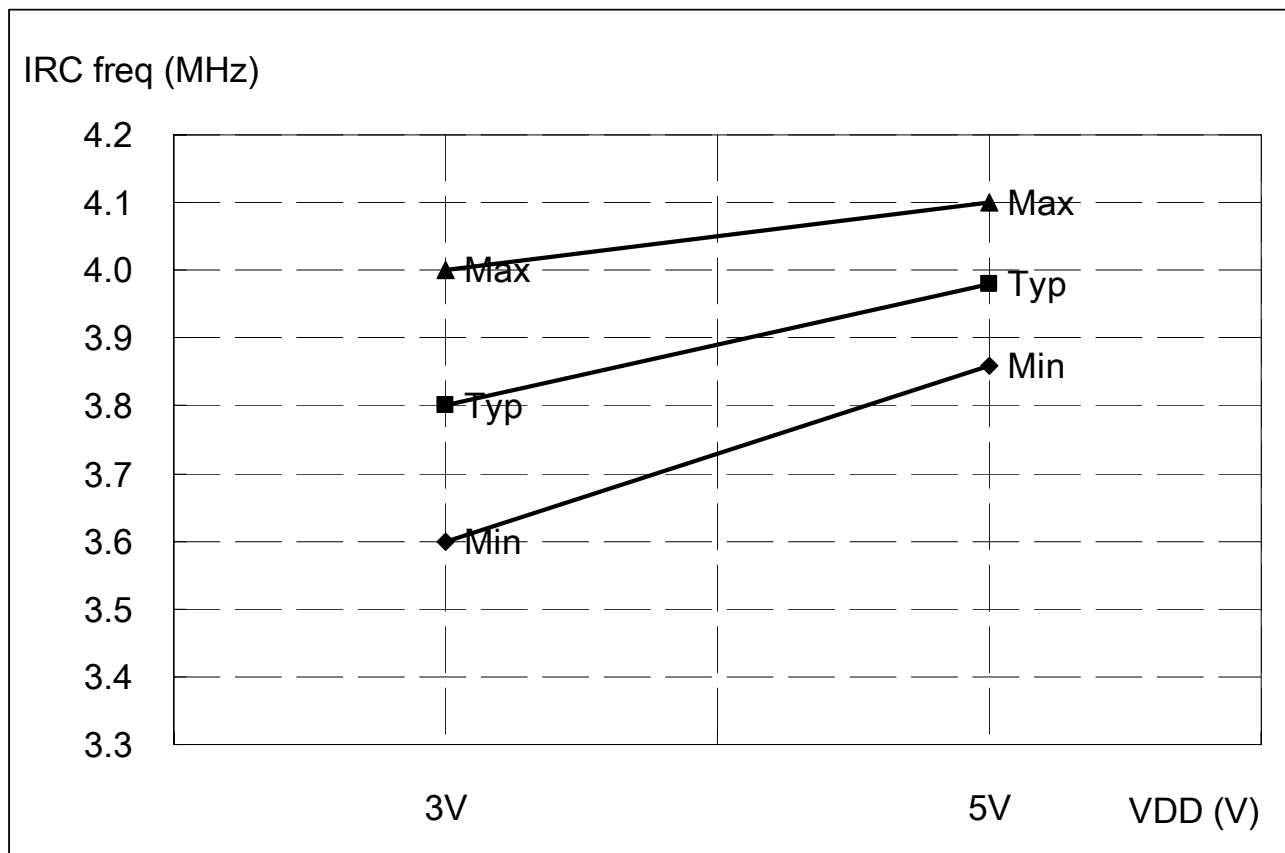
Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Input High Voltage	V_{IH}	All Input	$V_{DD} = 2.0$ to 5.5 V	$0.8 V_{DD}$	-	V_{DD}	V
Input Low Voltage	V_{IL}	All Input	$V_{DD} = 2.0$ to 5.5 V	0	-	$0.2 V_{DD}$	V
Output High Voltage <small>(NOTE 1)</small>	V_{OH}	All Output	$V_{DD} = 4.5$ to 5.5 V	$V_{DD} - 0.6$	-	-	V
Output Low Voltage <small>(NOTE 2)</small>	V_{OL}	All Output	$V_{DD} = 4.5$ to 5.5 V	-	-	0.5	V
Input Leakage Current (pin high)	I_{ILH}	All Input	$V_{IN} = V_{DD}$	-	-	1	μA
Input Leakage Current (pin low)	I_{ILL}	All Input	$V_{IN} = 0\text{ V}$	-	-	-1	μA
Power Supply Current	I_{DD}	Run 10 MHz	$V_{DD} = 4.5$ to 5.5 V	-	7	12	mA
		Run 4 MHz	$V_{DD} = 2.0\text{ V}$		2	4	
		Stop mode	$V_{DD} = 4.5$ to 5.5 V (LVR disable)	-	0.1	5	μA
			$V_{DD} = 4.5$ to 5.5 V (LVR enable)		2	5	
$V_{DD} = 2.6\text{ V}$ (LVR enable)	1	3					
System Clock Frequency	f_{OSC}	$V_{DD} > LVR_{th}$	$V_{DD} = 5\text{ V}$	-	-	20	MHz
			$V_{DD} = 3\text{ V}$			20	
			$V_{DD} = 2\text{ V}$			10	
Pull-Up Resistor	R_P	$V_{IN} = 0\text{ V}$ Ports A	$V_{DD} = 5\text{ V}$	41	82	164	k

NOTE:

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

3. Clock Timing ($T_A = -25^{\circ}\text{C}$ to $+85^{\circ}\text{C}$)

Parameter	Condition		Min	Typ	Max	Unit
External RC Frequency	$V_{DD} = 3\text{V}$	R = 5K C = 33pF	–	2.58	–	MHz
		R = 10K C = 100pF	–	0.85	–	
		R = 100K C = 300pF	–	0.04	–	
	$V_{DD} = 5\text{V}$	R = 5K C = 33pF	–	2.91	–	
		R = 10K C = 100pF	–	0.71	–	
		R = 100K C = 300pF	–	0.03	–	
Internal RC Frequency	$V_{DD} = 4.75$ to 5.25V		3.86	3.98	4.1	MHz
	$V_{DD} = 2.8$ to 3.2V		3.6	3.8	4.0	



4. Reset Timing Characteristics ($T_A = -25^{\circ}\text{C}$ to $+85^{\circ}\text{C}$, $V_{DD} = 2.0\text{V}$ 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 $V_{DD} = 5\text{V} \pm 10\%$	3	–	–	μs
WDT wakeup time	$V_{DD} = 5\text{V}$, WKTPSC = 11	90	110	160	mS
	$V_{DD} = 3\text{V}$, WKTPSC = 11	120	150	200	
CPU start up time	$V_{DD} = 5\text{V}$	–	3.5	–	mS

5. LVR Circuit Characteristics ($T_A = -25^\circ\text{C}$ to $+85^\circ\text{C}$, $V_{DD} = 2.0\text{ V}$ to 5.5 V)

Parameter	Symbol	Min	Typ	Max	Unit
LVR reference Voltage	V_{LVR}	1.9 2.9	2.2 3.2	2.4 3.5	V
LVR Hysteresis Voltage	V_{HYST}	–	± 0.1	–	V
Low Voltage Detection time	t_{LVR}	10	–	–	μs