

TMU3132MS

USB Full Speed Controller

Data Sheet

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

AMENDMENT HISTORY

Version	Date	Description
V1.0	Nov, 2011	New release
V1.1	Jan, 2012	Add Ordering Information table
V1.2	Feb, 2012	1. Add PDIP/SOP 20/18Pin 2. Revise Application figure. 3. Modify Ordering Information table

CONTENTS

AMENDMENT HISTORY	2
GENERAL DESCRIPTION	4
FEATURES	4
Functional Block Diagram	5
PIN DESCRIPTION	5
PACKAGE	6
Functional Description	7
1. CPU Core	7
1.1 Clock Scheme and Instruction Cycle	7
1.2 CPU Clock Control Register	7
1.3 Programming Counter (PC) and Stack	7
1.4 ALU and Working (W) Register.....	8
1.5 Addressing Mode	8
1.6 Instruction Set.....	9
2. Control Registers	22
F-Plane	22
R-Plane.....	24
3. USB Engine	26
3.1 USB Device Address	26
3.2 Endpoint 0 Receive (SET0/OUT0).....	26
3.3 Endpoint 0 Transmit (TX0).....	27
3.4 Endpoint 1/2 Transmit (TX1/2).....	27
3.5 Endpoint 3 Transmit (TX3).....	27
3.6 USB Endpoint 4 Receive (RC4).....	27
3.7 USB Control and Status.....	28
3.8 Suspend and Resume	28
3.9 Interrupt Vector	28
4. Wakeup Timer and Watch Dog Timer	29
5. Timer	29
5.1 Timer0: 8-bit Timer with Pre-scale (PSC)	29
6. SPI (Serial Peripheral Interface)	31
7. I/O Port	32
7.1 PA0-7	32
7.2 PB0	33
7.3 PB3 (DP) and PB2 (DM)	33
7.4 PE0-4.....	34
8. Application	35
9. Electrical Characteristics	35
10. Package Information	37
Ordering Information	38

GENERAL DESCRIPTION

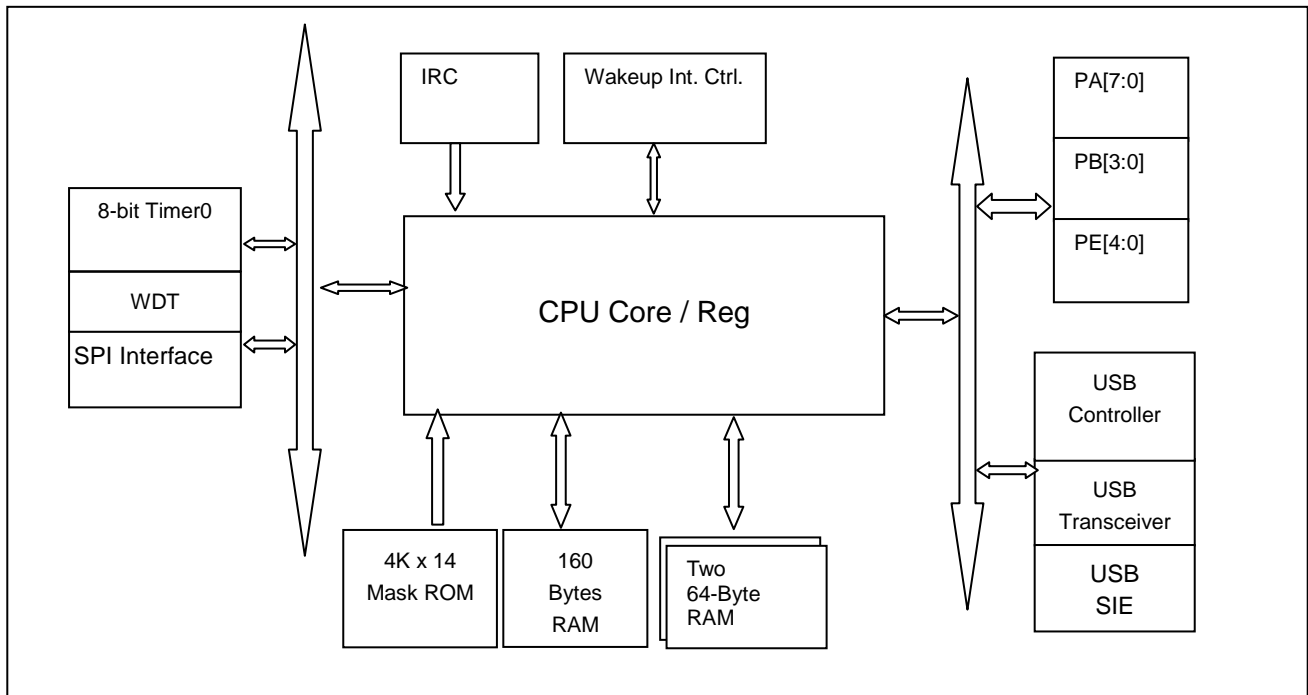
The TMU3132MS is a 2T RISC tailored to the USB 2.0 full speed general purpose IC. It contains 4K * 14 Mask ROM. The Internal RC Oscillator is 48 MHz +/- 0.25% for USB mode and 12 MHz clock output for the RF 2.4G module. The USB DMA channel allows data transfer between USB RAM with SPI interface which supports mode 0/1/2/3 master up to 6 Mbps clock rate. The TMU3132MS also supports suspend mode, wake-up and watchdog timer. The data memory is internal 160 bytes of RAM in F-plane and two 64 bytes in R-plane. The TMU3132MS USB MCU is suited for USB mouse and wireless 2.4G dongle device.

FEATURES

- **Operation Frequency**
 - Internal 48 MHz RC oscillator +/- 0.25% accuracy for USB with no extra component.
- **On-Chip Memory**
 - 4k x 14 internal Mask ROM.
 - Internal 160 Bytes RAM at F-plane and two 64 Bytes RAM at R-plane.
 - Built-in five 8-byte USB Application FIFOs at R-plane.
- **8-bit RISC MCU**
 - 8-level stacks for subroutine and interrupt.
 - 37 instructions.
 - Two clocks per instruction except branch.
- **USB interface**
 - Compliance with the Universal Serial Bus specification v2.0 Full Speed.
 - Built-in 1.5K pull-high resistor can be set or disabled.
 - Built-in USB Transceiver, 3.3V regulator.
 - Support USB Suspend/Resume and Remote Wakeup function.
 - Endpoint 0: Control SETUP/IN/OUT transfer (each 8 bytes).
 - Endpoint 1: INTERRUPT IN transfer (8 bytes).
 - Endpoint 2: INTERRUPT IN transfer (8 bytes).
 - Endpoint 3: BULK-IN transfer with Ping-Pong feature (2*64 bytes).
 - Endpoint 4: BULK-OUT transfer with Ping-Pong feature (2*64 bytes).
- **Timer**
 - Timer0 is 8-bit with 8-bit prescaler, Counter/Reload/Interrupt function.
 - Watchdog Timer clocked by built-in RC oscillator.
 - Wakeup Timer clocked by built-in RC oscillator.
- **Reset Controller**
 - Power On Reset is 2.0V.
 - Watch-Dog Timer reset period can be configured at 16 ms/32 ms/64 ms/128 ms.

- **SPI interface**
 - Support Mode0, 1, 2, 3.
 - Master only.
 - Clock rate up to 6 Mbps.
 - Read/Write DMA mode.
- **I/O Ports**
 - Max. 16 GPIOs to flexible application
 - Each GPIO pin supports Schmitt-trigger input, internal pull-ups and open drains output or CMOS push-pull output
- PDIP/SOP 20/18 or Die Form
- Application: USB full speed general purpose

Functional Block Diagram



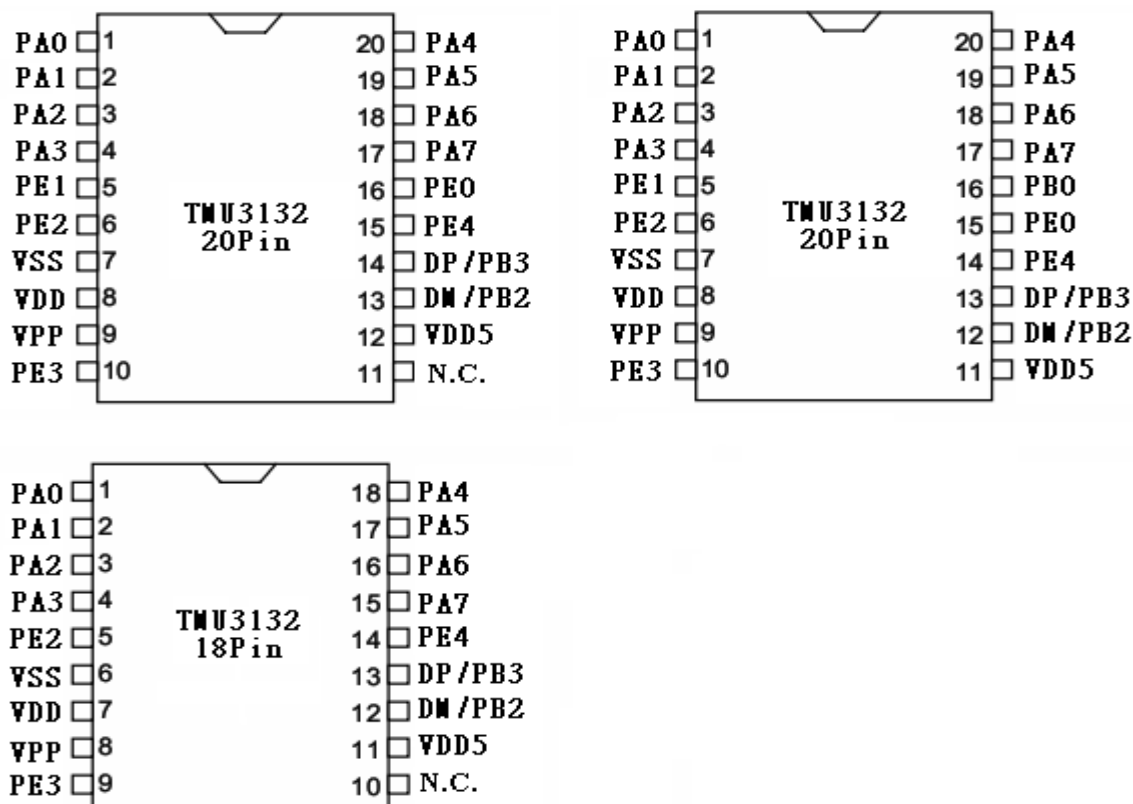
PIN DESCRIPTION

PIN ASSIGNMENT

Name	I/O	Description
VDD5	P	PC5V Power input
VSS	P	Ground
VDD	O	3.3V regulator output
VPP/RSTN	I	Programming High power/Chip reset pin
PA[7]	I/O	General purpose I/O (Pseudo open-drain) ;
PA[6]/SDO	I/O	General purpose I/O (Pseudo open-drain) ; SPI Dout
PA[5]/SCLK	I/O	General purpose I/O (Pseudo open-drain) ; SPI Clk
PA[4]/SDI	I/O	General purpose I/O (Pseudo open-drain) ; SPI Din
PA[3]	I/O	General purpose I/O (Pseudo open-drain)
PA[2]	I/O	General purpose I/O (Pseudo open-drain)
PA[1]/	I/O	General purpose I/O (Pseudo open-drain);
PA[0]	I/O	General purpose I/O (Pseudo open-drain);
DP/PB[3]	I/O	USB positive signal/General purpose I/O
DM/PB[2]	I/O	USB negative signal/General purpose I/O
PB[0]	I/O	General purpose I/O (open-drain)
PE[4:0]	I/O	General purpose I/O (Pseudo open-drain) ; PE[3] can configured as clock output (6 MHz/12 MHz)

.I/O voltage is fixed 3.3V, unless otherwise specified.

PACKAGE



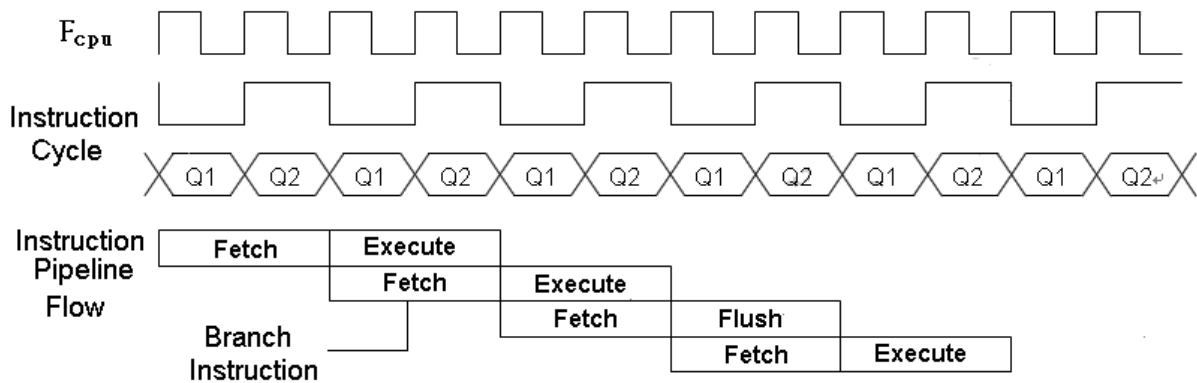
Functional Description

1. CPU Core

1.1 Clock Scheme and Instruction Cycle

TMU3132MS has only one chip clock sources as following:
 F_{rc} : Internal RC oscillator 24 MHz clock

F_{rc} can be synchronized by USB signals and popup to 48 MHz clock (F_{48m}) for USB module. F_{rc} can be divided to 12 MHz clock as CPU clock. The CPU 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 CPU Clock Control Register

CPU clock speed selection: The CPU clock source is Internal RC and it will be divided to 12 MHz, 6 MHz, 3 MHz or 1.5 MHz by firmware setting.

R07 [1:0] is used to select the different speed

R07 [1:0] =0 select 12 MHz

R07 [1:0] =1 select 6 MHz

R07 [1:0] =2 select 3 MHz

R07 [1:0] =3 select 1.5 MHz

1.3 Programming Counter (PC) and Stack

The Programming Counter is 12-bit wide capable of addressing a 4K x 14 program ROM. As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC value is normally increased by one except the followings. The Reset Vector (000h) and the Interrupt Vectors (from 001h to 00dh) are provided for PC initialization and Interrupt. For CALL/GOTO instructions, PC loads the lower 11 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 [11:8] keeps unchanged. The STACK is 12-bit wide and 8-level in depth. The CALL instruction and Hardware interrupt will push STACK level in order. While the RET/RETI/RETLW instruction pops the STACK level in order.

1.4 ALU and Working (W) Register

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

Note: /Borrow represents inverted of Borrow register.

/Digit Borrow represents inverted of Digit Borrow register.

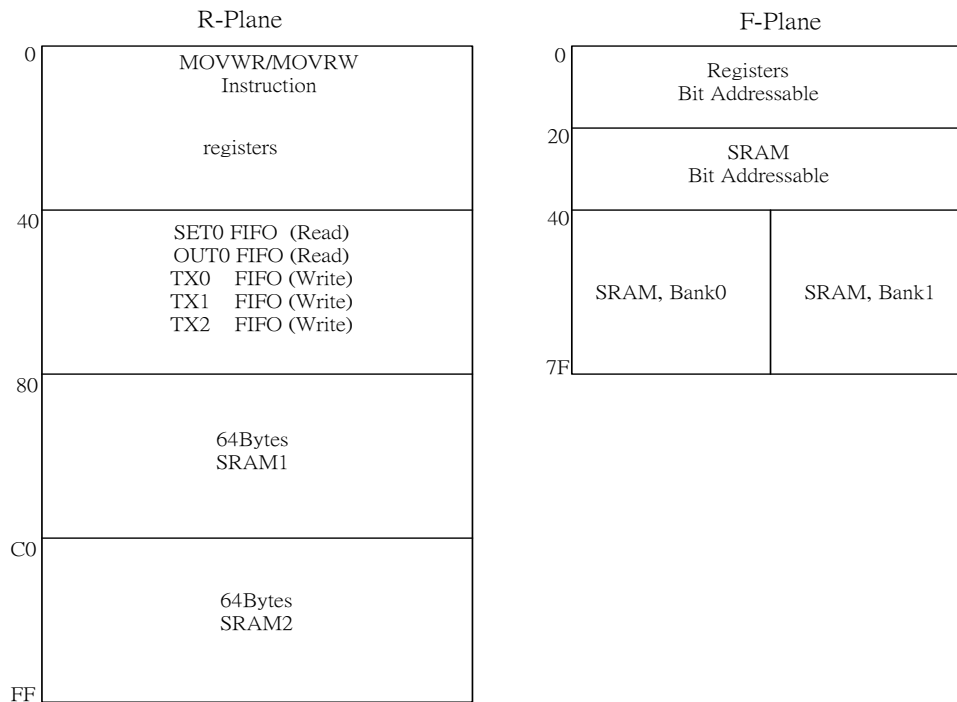
1.5 Addressing Mode

There are two Data Memory Planes in CPU, R-Plane and F-Plane. 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. R-plane can be indirect accessed via RSR register.

1. 4K x 14 Mask ROM.
2. 160-byte SRAM (F-plane) is addressed from 0x20 to 0x7F. The lower 32-byte (0x20 ~ 0x3f) is bit addressable. The higher address (0x40 ~ 0x7F) is separated to two banks which can be selected by setting register F03[5]. F03[5]=0 is used to select Bank0, F03[5]=1 is used to select Bank1.
3. Two 64-bytes RAM (R-plane).
4. Five 8-byte USB FIFOs are allocated in R-plane.

Example: Initial value: [F30h]=12h

```
MOVLW 30h ;W=30h
MOVWF FSR
MOVFW INDF ;W=12h
CLRWF
MOVWF INDF ;[F30h]=00h
INCF FSR,F ;FSR=31h
```

1.6 Instruction Set

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

For byte-oriented instructions, “f” or “r” represents the address designator and “d” represents the destination designator. The address designator is used to specify which address in Program memory is 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

Field / Legend	Description
GIE	Global Interrupt Enable Flag (i-Flag)
[]	Option Field
()	Contents
.	Bit Field
B	Before
A	After
←	Assign direction

Mnemonic		Op Code	Cycle	Flag Affect	Description
Byte-Oriented File Register Instruction					
ADDWF	f,d	00 0111 dfff ffff	1	C,DC,Z	Add W to f
ANDWF	f,d	00 0101 dfff ffff	1	Z	AND W to f
CLRF	f	00 0001 1fff ffff	1	Z	Clear f
CLRW		00 0001 0100 0000	1	Z	Clear W
COMF	f,d	00 1001 dfff ffff	1	Z	Invert F bit by bit
DECF	f,d	00 0011 dfff ffff	1	Z	Decrement of f
DECFSZ	f,d	00 1011 dfff ffff	1 or 2	-	Decrease f, skip if zero
Byte-Oriented File Register Instruction					
INCF	f,d	00 1010 dfff ffff	1	Z	Increment of f
INCFSZ	f,d	00 1111 dfff ffff	1 or 2	-	Increase f, skip if zero
IORWF	f,d	00 0100 dfff ffff	1	Z	OR W to f
MOVFW	f	00 1000 0fff ffff	1	-	Move f to W
MOVWF	f	00 0000 1fff ffff	1	-	Move W to f
MOVRW	r	01 1111 rrrr rrrr	1	-	Move r to W
MOVWR	r	01 1110 rrrr rrrr	1	-	Move W to r
RLF	f,d	00 1101 dfff ffff	1	C	F rotate to left
RRF	f,d	00 1100 dfff ffff	1	C	F rotate to right
SUBWF	f,d	00 0010 dfff ffff	1	C,DC,Z	Substrate W from f
SWAPF	f,d	00 1110 dfff ffff	1	-	Swap high and low nibble of f
TESTZ	f,d	00 1000 dfff ffff	1	Z	Test f if zero
XORWF	f,d	00 0110 dfff ffff	1	Z	XOR W to f
Bit-Oriented File Register Instruction					
BCF	f,b	01 000b bbff ffff	1	-	Bit clear f
BSF	f,b	01 001b bbff ffff	1	-	Bit set f
BTFSC	f,b	01 010b bbff ffff	1 or 2	-	Bit test f, skip if clear
BTFSS	f,b	01 011b bbff ffff	1 or 2	-	Bit test f, skip if set

Mnemonic		Op Code	Cycle	Flag Affect	Description
Literal and Control Instruction					
ADDLW	k	01 1100 kkkk kkkk	1	C,DC,Z	Add literal to W
ANDLW	k	01 1011 kkkk kkkk	1	Z	AND literal to W
XORLW	K	01 1101 kkkk kkkk	1	Z	XOR literal to W
CALL	k	10 kkkk kkkk kkkk	2	-	Subroutine call
CLRWDT		01 1110 0000 0011	1	-	Clear watchdog timer
GOTO	k	11 kkkk kkkk kkkk	2	-	Unconditional branch
IORLW	k	01 1010 kkkk kkkk	1	Z	OR literal to W
MOVLW	k	01 1001 kkkk kkkk	1	-	Move literal to W
NOP		00 0000 0000 0000	1	-	No operation
RET		00 0000 0100 0000	2	-	Return from CALL
RETI		00 0000 0110 0000	2	-	Return from interrupt
RETLW	k	01 1000 kkkk kkkk	2	-	Return with literal to W
SLEEP		01 1110 0000 0011	1	-	Power down

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 ~ 7Fh 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	$(W) \leftarrow (W) \text{ ‘AND’ } k$	
Status Affected	Z	
OP-Code	01 1011 kkkk kkkk	
Description	The contents of W register are AND’ed with the eight-bit literal ‘k’. The result is placed in the W register.	
Cycle	1	
Example	ANDLW 0x5F	B : W = 0xA3 A : W = 0x03
ANDWF	AND W with “f”	
Syntax	ANDWF f [,d]	
Operands	f : 00h ~ 7Fh d : 0, 1	
Operation	$(\text{Destination}) \leftarrow (W) \text{ ‘AND’ } (f)$	
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) ← 0		
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	

CALL	Call subroutine "k"
Syntax	CALL k
Operands	k : 00h ~ FFFh
Operation	Operation: TOS ← (PC)+ 1, PC.11~0 ← k
Status Affected	-
OP-Code	10 kkkk kkkk kkkk
Description	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The 12-bit immediate address is loaded into PC bits <11:0>. CALL is a two-cycle instruction.
Cycle	2
Example	LABEL1 CALL SUB1 B : PC = LABEL1 A : PC = SUB1, TOS = LABEL1+1
CLRF	Clear "f"
Syntax	CLRF f
Operands	f : 00h ~ 7Fh
Operation	(f) ← 00h, Z ← 1
Status Affected	Z
OP-Code	00 0001 1fff ffff
Description	The contents of register 'f' are cleared and the Z bit is set.
Cycle	1
Example	CLRF FLAG_REG B : FLAG_REG = 0x5A A : FLAG_REG = 0x00, Z = 1
CLRW	Clear W
Syntax	CLRW
Operands	-
Operation	(W) ← 00h, Z ← 1
Status Affected	Z
OP-Code	00 0001 0100 0000
Description	W register is cleared and Zero bit (Z) is set.
Cycle	1
Example	CLRW B : W = 0x5A A : W = 0x00, Z = 1
CLRWDT	Clear Watchdog Timer
Syntax	CLRWDT
Operation	-
Operation	WDTE ← 00h
Status Affected	-
OP-Code	01 1110 0000 0100
Description	CLRWDT instruction enables and reset the Watchdog Timer.
Cycle	1
Example	CLRWDT B : WDT counter = ? A : WDT counter = 0x00

COMF		Complement "f"	
Syntax	COMF f [,d]		
Operands	f : 00h ~ 7Fh, d : 0, 1		
Operation	(destination) ← (f)		
Status Affected	Z		
OP-Code	00 1001 dfff ffff		
Description	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.		
Cycle	1		
Example	COMF REG1,0	B : REG1 = 0x13	A : REG1 = 0x13, W = 0xEC
DECF		Decrement "f"	
Syntax	DECF f [,d]		
Operands	f : 00h ~ 7Fh, d : 0, 1		
Operation	(destination) ← (f) - 1		
Status Affected	Z		
OP-Code	00 0011 dfff ffff		
Description	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.		
Cycle	1		
Example	DECF CNT, 1	B : CNT = 0x01, Z = 0	A : CNT = 0x00, Z = 1
DECFSZ		Decrement "f", skip if 0	
Syntax	DECFSZ f [,d]		
Operands	f : 00h ~ 7Fh, d : 0, 1		
Operation	(destination) ← (f) - 1, skip next instruction if result is 0		
Status Affected	Z		
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	B : PC = LABEL1	
	GOTO LOOP	A : CNT = CNT - 1	
	CONTINUE	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	
	GOTO LOOP	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 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 are 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	k : 00h ~ FFh
Operation	(destination) ← (W) OR (f)
Status Affected	Z
OP-Code	00 0100 dfff ffff
Description	Inclusive OR the W register with register ‘f’. If ‘d’ is 0, the result is placed in the W register. If ‘d’ is 1, the result is placed back in register ‘f’.
Cycle	1
Example	IORWF RESULT, 0 B : RESULT = 0x13, W = 0x91 A : RESULT = 0x13, W = 0x93, Z = 0

MOVFW	Move “f” to W
Syntax	MOVFW f
Operands	f : 00h ~ 7Fh
Operation	(W) ← (f)Status
Affected	-
OP-Code	00 1000 0fff ffff
Description	The contents of register f are moved to W.
register. Cycle	1
Example	MOVFW REG1 B : REG1 = 0x4F, W = ? A : REG1 = 0x4F, W = 0x4F
MOVLW	Move Literal to W
Syntax	MOVLW k
Operands	k : 00h ~ FFh
Operation	(W) ← k
Status Affected	-
OP-Code	01 1001 kkkk kkkk
Description	The eight-bit literal ‘k’ is loaded into W register. The don’t cares will assemble as 0’s.
Cycle	1
Example	MOVLW 0x5A B : W = ? A : W = 0x5A
MOVWF	Move W to “f”
Syntax	MOVWF f
Operands	f : 00h ~ 7Fh
Operation	(f) ← (W)
Status Affected	-
OP-Code	00 0000 1fff ffff
Description	Move data from W register to register ‘f’.
Cycle	1
Example	MOVWF REG1 B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F
MOVWR	Move W to “r”
Syntax	MOVWR r
Operands	r : 00h ~ FFh
Operation	(r) ← (W)
Status Affected	-
OP-Code	01 1110 rrrr rrrr
Description	Move data from W register to register ‘r’.
Cycle	1
Example	MOVWR REG1 B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

MOVRW	Move “r” to W
Syntax	MOVRW r
Operands	r : 20h ~ FFh
Operation	(W) ← (r)
Status Affected	-
OP-Code	01 1111 rrrr rrrr
Description	Move data from register ‘r’ to W register.
Cycle	1
Example	MOVRW REG1 B : REG1 = 0x4F, W = ? A : REG1 = 0x4F, W = 0x4F
NOP	No Operation
Syntax	NOP
Operands	-
Operation	No Operation
Status Affected	-
OP-Code	00 0000 0000 0000
Description	No Operation
Cycle	1
Example	NOP
RETI	Return from Interrupt
Syntax	RETI
Operands	-
Operation	PC ← TOS, GIE ← 1
Status Affected	-
OP-Code	00 0000 0110 0000
Description	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in to the PC. Interrupts are enabled. This is a two-cycle instruction.
Cycle	2
Example	RETI A : PC = TOS, GIE = 1
RETLW	Return with Literal in W
Syntax	RETLW k
Operands	k : 00h ~ FFh
Operation	PC ← TOS, (W) ← k
Status Affected	-
OP-Code	01 1000 kkkk kkkk
Description	The W register is loaded with the 8-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 : TABLE ADDWF PCL, 1 RETLW k1 RETLW k2 : RETLW kn B : W = 0x07 A : W = value of k8

SLEEP	Go into standby mode, Clock oscillation stops	
Syntax	SLEEP	
Operands	-	
Operation	-	
Status Affected	TO,PD	
OP-Code	01 1110 0000 0011	
Description	Go into SLEEP mode with the oscillator stopped.	
Cycle	1	
Example	SLEEP	
SUBWF	Subtract W from “f”	
Syntax	SUBWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	$(W) \leftarrow (f) - (W)$	
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 ~ 7Fh, d : 0, 1	
Operation	$(\text{destination},7\sim4) \leftarrow (f.3\sim0), (\text{destination}.3\sim0) \leftarrow (f.7\sim4)$	
Status Affected	-	
OP-Code	00 1110 dfff ffff	
Description	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.	
Cycle	1	
Example	SWAPF REG, 0	B : REG1 = 0xA5 A : REG1 = 0xA5, W = 0x5A
TESTZ	Test if “f” is zero	
Syntax	TESTZ f	
Operands	f : 00h ~ 7Fh	
Operation	Set Z flag if (f) is 0	
Status Affected	Z	
OP-Code	00 1000 1fff ffff	
Description	If the content of register 'f' is 0, Zero flag is set to 1.	
Cycle	1	
Example	TESTZ REG1	B : REG1 = 0, Z = ? A : REG1 = 0, Z = 1

XORLW	Exclusive OR Literal with W	
Syntax	XORLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) XOR k	
Status Affected	Z	
OP-Code	01 1101 kkkk kkkk	
Description	The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.	
Cycle	1	
Example	XORLW 0xAF	B : W = 0xB5 A : W = 0x1A
XORWF	Exclusive OR W with "f"	
Syntax	XORWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (W) XOR (f)	
Status Affected	Z	
OP-Code	00 0110 dfff ffff	
Description	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	XORWF REG 1	B : REG = 0xAF, W = 0xB5 A : REG = 0x1A, W = 0xB5

2. Control Registers

F-Plane

Name	Address	R/W	Rst	Descriptions	
F01	TIMER0	01.7~0	R/W	0	Timer 0 value
F02	PC	02.7~0	R/W	0	Program Counter [7~0]
F03	RAMBANK	03.5	R/W	0	SRAM Bank Select
	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
F04	FSR	04.6~0	R/W	0	File Select Register
F05	RSR	05.7~0	R/W	0	R-Plane File Select Register
F06	PAD	06.7~0	R/W	ff	Port A output data
F07	PBD	07.3~0	R/W	ff	Port B output data, PBD[3:0];
F0A	PED	0A.7~0	R/W	f	Port E output data, PED[4:0];
F10	USBE	10.7	R/W	0	USB function enable (1)
	FUNADR	10.6~0	R/W	0	USB function address
F11	SET0I	11.7	R/W	0	Endpoint 0 SET0 Receive Interrupt flag, write 0 to clear flag.
	OUT0I	11.6	R/W	0	Endpoint 0 OUT Receive Interrupt flag, write 0 to clear flag.
	TX0I	11.5	R/W	0	Endpoint 0 IN Transmit Interrupt flag, write 0 to clear flag.
	TX1I	11.4	R/W	0	Endpoint 1 IN Transmit Interrupt flag, write 0 to clear flag.
	TX2I	11.3	R/W	0	Endpoint 2 IN Transmit Interrupt flag, write 0 to clear flag.
	SUSPI	11.2	R/W	0	USB Suspend Interrupt flag, write 0 to clear flag.
	TX3I	11.1	R/W	0	Endpoint 3 Bulk Transmit Interrupt flag, write 0 to clear flag.
	RC4I	11.0	R/W	0	Endpoint 4 Bulk Receive Interrupt flag, write 0 to clear flag.
F12	WKT I	12.5	R/W	0	Wakeup Timer Interrupt flag, write 0 to clear flag
	RST I	12.4	R/W	0	USB Bus Reset Interrupt flag, write 0 to clear flag.
	RSM I	12.3	R/W	0	USB Resume Interrupt flag, write 0 to clear flag.
	PB0 I	12.1	R/W	0	PB0 Interrupt flag, write 0 to clear flag.
	TM0 I	12.0	R/W	0	Timer0 Interrupt flag, write 0 to clear flag.
F13	SUSPND	13.7	R/W	0	S/W force USB interface into suspend mode.
	RSMO	13.6	R/W	0	S/W force USB interface sends RESUME signal in suspend mode.
	EP1CFG	13.5	R/W	0	Set Endpoint 1 configured.
	EP2CFG	13.4	R/W	0	Set Endpoint 2 configured.
	Device_R	13.3	R/W	0	DP Pull-high resistor enable bit: 0: Disable pull-high , 1: pull-high enable
	OUT0RDY	13.0	R/W	0	Endpoint 0 ready for receive, cleared by H/W while OUT0I occurs.

	Name	Address	R/W	Rst	Descriptions
F14	TX0RDY	14.7	R/W	0	Endpoint 0 ready for transmit, cleared by H/W while TX0I occurs.
	TX0TGL	14.6	R/W	0	Endpoint 0 transmits DATA1/DATA0 packet.
	EP0STALL	14.5	R/W	0	Endpoint 0 will stall OUT/IN packet.
	IN0STALL	14.4	R/W	0	Endpoint0 IN Stall(1)
	TX0CNT	14.3~0	R/W	0	Endpoint 0 transmits byte count.
F15	TX1RDY	15.7	R/W	0	Endpoint 1 ready for transmit, cleared by H/W while TX1I occurs.
	TX1TGL	15.6	R/W	0	Endpoint 1 transmits DATA1/DATA0 packet.
	EP1STALL	15.5	R/W	0	Endpoint 1 will stall IN packet.
	TX1CNT	15.3~0	R/W	0	Endpoint 1 transmits byte count.
F16	TX2RDY	16.7	R/W	0	Endpoint 2 ready for transmit, cleared by H/W while TX2I occurs.
	TX2TGL	16.6	R/W	0	Endpoint 2 transmits DATA1/DATA0 packet.
	EP2STALL	16.5	R/W	0	Endpoint 2 will stall IN packet.
	TX2CNT	16.3~0	R/W	0	Endpoint 2 transmits byte count.
F17	TX3RDY	17.7	R/W	0	Endpoint 3 ready for transmit, cleared by H/W while TX3I occurs.
	TX3TGL	17.6	R/W	0	Endpoint 3 transmits DATA1/DATA0 packet.
	EP3STALL	17.5	R/W	0	Endpoint 3 will stall IN packet.
	EP3CFG	17.4	R/W	0	Set Endpoint 3 configured.
F18	RC4RDY	18.7	R/W	0	Endpoint 4 ready for receive, cleared by H/W while RC4I occurs.
	RC4TGL	18.6	R	-	Endpoint 4 receives DATA1/DATA0 packet
	EP4STALL	18.5	R/W	0	Endpoint 4 will stall OUT packet.
	EP4CFG	18.4	R/W	0	Set Endpoint 4 configured
	RC4ERR	18.3	R	0	EP4 receives data error.
F19	TX3CNT	19.6~0	R/W	0	Endpoint 3 transmits byte count.
F1A	RC4CNT	1A.6~0	R	0	Endpoint 4 transmits byte count.
F1C	SRAMCON	1C.5~0	R/W	0	SRAM Configuration
	SRAM1USB	1C.5	R/W	0	Assign SRAM1 as USB Bulk Transfer buffer
	SRAM2USB	1C.4	R/W	0	Assign SRAM2 as USB Bulk Transfer buffer
	SRAM1SPI	1C.3	R/W	0	Assign SRAM1 as SPI DMA Transfer buffer
	SRAM2SPI	1C.2	R/W	0	Assign SRAM2 as SPI DMA Transfer buffer
F1D	SPI_MODE	1D.5	R/W	0	SPI MODE
	SPI_EN	1D.4	R/W	0	SPI Enable, Busy bit
	LSB_First	1D.3	R/W	0	1: Data transmit/Receive is LSB first; 0: MSB first
	SPI_IN	1D.2	R/W	0	(1) SPI Bus is used to receive data from SPI Device (0) SPI Bus is used to transmit data to SPI Device
	SPI_cmd_sw	1D.1	R/W	0	SPI CMD/DAT Switch; 1:CMD, 0:DAT
	clr_ram_adr	1D.0	R/W	0	Write 1 to clear ram address
	SRAM	20~7F	R/W	-	Internal RAM (96 Bytes x 2 Banks)

R-Plane

Name	Address	R/W	Rst	Description
R01 T0RLD	01.7~0	W	0	Timer0 overflow reload value
R02 T0en	02.4	W	0	Timer0 Enable
T0PSC	02.3~0	W	0	Timer0 Pre-Scale, 0: div1, 1: div2, 2: div4, ..., 8: div256
R03 PWRDOWN	03	W	0	Write this register to enter Power-Down Mode
R04 WDTE	04	W	0	Write this register to clear WDT and enable WDT
R06 WRC_PD	06.7	W	0	WRC Disable, 1: Disable WRC, 0: Enable WRC
WDTPSC	06.6~5	W	11	WDT period, 00=17.5 ms, 01=35 ms, 10=70 ms, 11=140 ms
WKT PSC	06.4~3	W	11	WKT period, 00=140 ms, 01=280 ms, 10=560 ms, 11=1120 ms
R07 CLKDIV	07.1~0	W	-	System Clk Period Selection 2'b00: 12 MHz 2'b01: 6 MHz 2'b10: 3 MHz 2'b11: 1.5 MHz
R09 EN_PE3_CKO	09.1	W		Set PE3 to be GPIO or IRC CKO; 0: GPIO port, 1: IRC CKO
IRCCKO_SEL	09.0	W		When EN_PE3_CKO= 1, IRCCKO_SEL is used to select clock frequency, 0: IRC 12 MHz output, 1: IRC 6 MHz output
R10 TESTREG	10.2~0	W	0	Test Mode option
R11 SET0IE	11.7	W	0	SET0I Interrupt enable
OUT0IE	11.6	W	0	OUT0 Interrupt enable
TX0IE	11.5	W	0	TX0I Interrupt enable
TX1IE	11.4	W	0	TX1I Interrupt enable
TX2IE	11.3	W	0	TX2I Interrupt enable
SUSPIE	11.2	W	0	SUSPI Interrupt enable
TX3IE	11.1	W	0	TX3I Interrupt enable
RC4IE	11.0	W	0	RC4I Interrupt enable
R12 WKTIE	12.5	W	0	Wakeup Timer Interrupt enable
RSTIE	12.4	W	0	RSTI Interrupt enable
RSMIE	12.3	W	0	RSMI Interrupt enable
PB0IE	12.1	W	0	PB0 Interrupt enable
TM0IE	12.0	W	0	Timer0 Interrupt enable
R13 RC0TGL	13.7	R		1: received DATA1 packet; 0: received DATA0 Packet.
RC0ERR	13.6	R		Endpoint 0 received data error.
EP0DIR	13.5	R		1: IN transfer; 0: OUT/SETUP transfer.
EP0SET	13.4	R		SETUP Token indicator.
OUT0CNT	13.3~0	R		OUT0 Received data byte count.
R20 PAE	20.7~0	W	0	1:Port A CMOS push-pull output enable, 0:PortA used as Pseudo Open drain

	Name	Address	R/W	Rst	Description
R21	PBE	21.3~0	W	0	1:Port B CMOS push-pull output enable, 0:PortB used as Open drain
R24	PEE	24.7~0	W	0	1:Port E CMOS push-pull output enable, 0:PortE used as Pseudo Open drain
R25	PAPU	25.7~0	W	0	1:PortA pull-up disable, 0:PortA pull-up enable
R26	PBPU	26.3~0	W	0	1:PortB pull-up disable, 0:PortB pull-up enable
R27	PEPU	27.0	W	0	1:PortE pull-up disable, 0:PortE pull-up enable
R3B	CPOL	3B.5	W	0	SPI Clock Polarity
	CPHA	3B.4	W	0	SPI Clock Phase
	BSL[3:0]	3B.3~0	W	0	Buffer shift bit counter; set 4'b0111 for byte (8 bits) transfer
R3C	CRS[6:0]	3C.6~0	W	0	SPI clock Select
R3D	SPI_LENGTH	3D.6~0	W	0	SPI DMA transfer length; Length: 1 ~ 64 bytes
R3E	SPI_TXDAT	3E.7~0	W	0	SPI Transmit DATA in CMD phase
R3F	SPI_RXDAT	3F.7~0	R	-	SPI Received Data
	SET0FIFO	40~47	R	-	Endpoint 0 SETUP Receive Buffer (8 Bytes)
	OUT0FIFO	48~4F	R	-	Endpoint 1 OUT Receive Buffer (8 Bytes)
	TX0FIFO	50~57	W	-	Endpoint 0 Transmit Buffer (8 Bytes)
	TX1FIFO	58~5F	W	-	Endpoint 1 Transmit Buffer (8 Bytes)
	TX2FIFO	60~67	W	-	Endpoint 2 Transmit Buffer (8 Bytes)
	SRAM1	80~BF	R/W	-	Endpoint 3/4 Buffer (64 Bytes)
	SRAM2	C0~FF	R/W	-	Endpoint 3/4 Buffer (64 Bytes)

3. USB Engine

The USB engine includes the Serial Interface Engine (SIE), the full-speed USB I/O transceiver. The SIE block performs most of the USB interface function with only minimum support from F/W. Three endpoints are supported. Endpoint 0 is used to receive and transmit control (including SETUP) packets. Endpoint 1 and endpoint 2 are used for interrupt transfer. Endpoint 3 and endpoint 4 are used for bulk transfer.

The USB SIE handles the following USB bus activity independently:

1. Bitstuffing/unstuffing
2. CRC generation/checking
3. ACK/NAK
4. TOKEN type identification
5. Address checking

F/W handles the following tasks:

1. Coordinate enumeration by responding to SETUP packets
2. Fill and empty the FIFOs
3. Suspend/Resume coordination
4. Verify and select DATA toggle values

3.1 USB Device Address

The USB device address register F10[6:0] (USBADR) stores the device's address. This register is reset to all 0 after chip reset. F/W must write this register a valid value after the USB enumeration process.

3.2 Endpoint 0 Receive (SET0/OUT0)

After receiving a SETUP packet and placing the data into the Endpoint 0 setup receive FIFO (SET0FIFO), TMU3132MS updates the Endpoint 0 status registers to record the receive status and then generates an Endpoint 0 setup receive interrupt (SET0I). The received data are always stored into SET0FIFO for DATA packets following SETUP token.

If received is a valid OUT packet, then generates Endpoint 0 out receive interrupt (OUT0I), data are stored into OUT0FIFO, F/W can read the status register F13, F14 and R14 for the recent transfer information, which includes the data byte count (OUT0CNT), packet toggle bit (RC0TGL) and data valid flag (RC0ERR). The data following an OUT token are written into OUT0FIFO and the OUT0CNT is updated unless Endpoint 0 STALL (EP0STALL) is set or Endpoint 0 receive ready (OUT0RDY) is not cleared. The data following an OUT token are written into the OUT0FIFO, and the OUT0CNT is updated unless Endpoint 0 STALL (EP0stall) is set or Endpoint 0 receive ready (OUT0RDY) is cleared. The SIE clears the OUT0RDY automatically and generates OUT0I interrupt when the OUT0CNT or OUT0FIFO is updated. As long as the OUT0RDY is cleared, SIE keeps responding NAK to Host's Endpoint 0 OUT packet request. F/W should set the OUT0RDY flag after the OUT0I interrupt is asserted and OUT0FIFO is read out.

3.3 Endpoint 0 Transmit (TX0)

After detecting a valid Endpoint 0 IN token, TMU3132MS automatically transmits the data pre-stored in the Endpoint 0 transmit FIFO (TX0FIFO) to the USB bus if the Endpoint 0 transmit ready flag (TX0RDY) is set and the EP0STALL is cleared. The number of byte to be transmitted depends on the Endpoint 0 transmit byte count register (TX0CNT). The DATA0/1 token to be transmitted depends on the Endpoint 0 transmit toggle control bit (TX0TGL). After the TX0FIFO is updated, TX0RDY should be set to 1. This enables the TMU3132MS to respond to an Endpoint 0 IN packet. TX0RDY is cleared and an Endpoint 0 transmit interrupt (TX0I) is generated once the USB host acknowledges the data transmission. The interrupt service routine can check TX0RDY to confirm that the data transfer is successful.

3.4 Endpoint 1/2 Transmit (TX1/2)

Endpoint 1 and Endpoint 2 are capable of transmit only. These endpoints are enabled when the Endpoint 1/Endpoint 2 configuration control bit (EP1CFG/EP2CFG) is set. After detecting a valid Endpoint 1/2 IN token, TMU3132MS automatically transmits the data pre-stored in the Endpoint 1/2 transmit FIFO (TX1FIFO/TX2FIFO) to the USB bus if the Endpoint 1/2 transmit ready flag (TX1RDY/TX2RDY) is set and the EP1STALL/EP2STALL is cleared. The number of byte to be transmitted depends on the Endpoint 3/4 transmit byte count register (TX1CNT/TX2CNT). The DATA0/1 token to be transmitted depends on the Endpoint 1/2 transmit toggle control bit (TX1TGL/TX2TGL). After the TX1FIFO/TX2FIFO is updated, TX1RDY/TX2RDY should be set to 1. This enables the TMU3132MS to respond to an Endpoint 1/2 IN packet. TX1RDY/TX2RDY is cleared and an Endpoint 1/2 transmit interrupt (TX1I/TX2I) is generated once the USB host acknowledges the data transmission. The interrupt service routine can check TX1RDY/TX2RDY to confirm that the data transfer is successful.

3.5 Endpoint 3 Transmit (TX3)

Endpoint 3 is capable of transmit only. Register F15, F19 and F1C are used to control this endpoint. Endpoint 3 is enabled when the configuration control bit (EP3CFG) is set. To properly use this endpoint, F/W must set SRAM1USB=1 or SRAM2USB=1 to assign exactly one SRAM (SRAM1 or SRAM2) as USB Bulk In buffer. Once this endpoint is enabled, F/W should set the Toggle bit (TX3TGL) and set the transmit byte count register (TX3CNT). After detecting a valid Endpoint 1 IN token, TMU3132MS automatically transmits the data pre-stored in the Endpoint 3 SRAM buffer to the USB bus if the Endpoint 3 transmits ready flag (TX3RDY) is set and the EP3STALL is cleared. The number of byte to be transmitted depends on the Endpoint 3 transmit byte count register (TX3CNT). The DATA0/1 token to be transmitted depends on the Endpoint 1 transmit toggle control bit (TX3TGL). Once the USB host acknowledges the data transmission, Endpoint 3 transmit interrupt (TX3I) is generated and the TX3RDY will be cleared. The interrupt service routine can check TX3RDY to confirm that the data transfer is successful.

3.6 USB Endpoint 4 Receive (RC4)

Endpoint 4 is capable of receive only. Register F18, R1A and F1C are used to control this endpoint. This endpoint is enabled when Endpoint 4 configured control bit (EP4CFG) is set. To properly use this endpoint, F/W must set SRAM1USB=1 or SRAM2USB=1 to assign exactly one SRAM (SRAM1 or SRAM2) as USB Bulk out buffer. After detecting a valid Endpoint 4 OUT token, the TMU3132MS automatically stores the bulk out data into the specified Bulk out buffer and updates RC4CNT if the Endpoint 4 receiving ready flag (RC4RDY)

is set and the EP4STALL is cleared. The DATA0/DATA1 token to be checked is toggled by F/W. When an Endpoint 4 receive interrupt (RC4I) is generated, the RC4RDY is cleared. During the packet transfer stage, if data are used to check error, the result will be responded on RC4ERR.

3.7 USB Control and Status

Other USB control bits include the USB enable (USBE), Suspend (SUSP), Resume output (RSM, Device Resister (DEVICE_R), and corresponding interrupt enable bits. The DEVICE_R is set to enable DP pull-up resistor. Other USB status flag includes the USB reset interrupt (RSTI), Resume input interrupt (RSMI), and USB Suspend interrupt (SUSPI).

3.8 Suspend and Resume

Once the Suspend condition is asserted, F/W can set the SUSP bit to save the power consumption of USB Engine. F/W can further save the device power by forcing the CPU to go into the Power Down Mode by setting register R03. In the Power Down mode, CPU can be waken-up by the trigger of any enabled interrupt's source or by USB bus reset or by USB bus resume. The TMU3132MS sends Resume signaling to USB bus when SUSP=1 and RSMO=1.

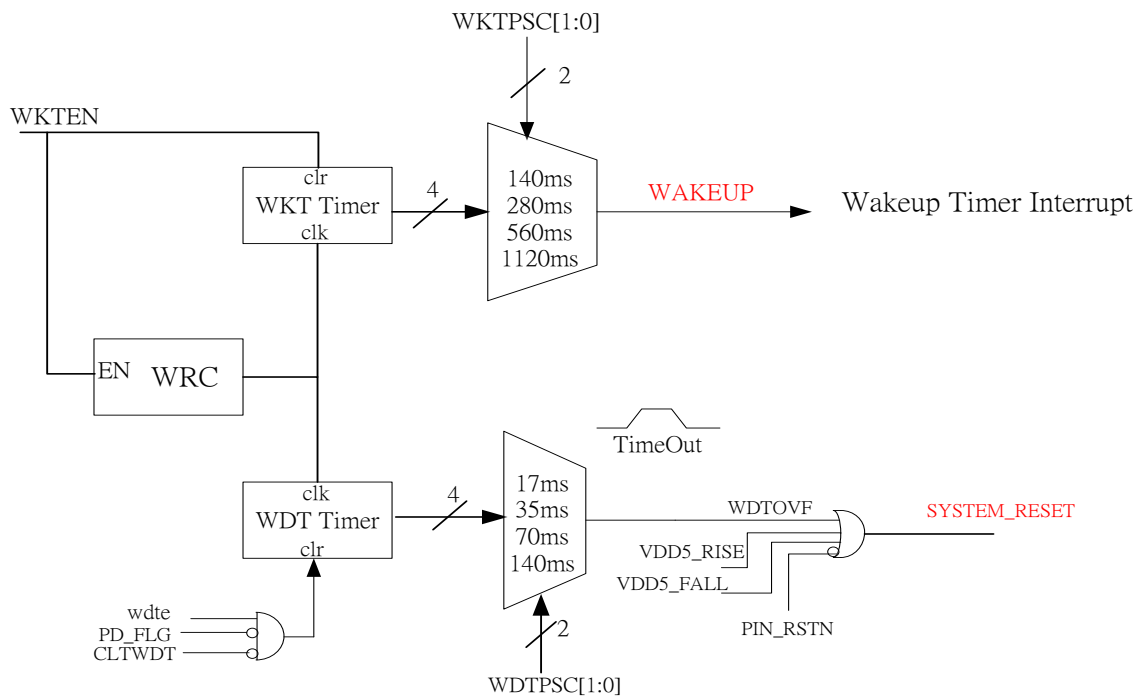
3.9 Interrupt Vector

There are several interrupts generated by USB Engine. The other interrupts including timer0 interrupts, wakeup timer interrupt and PB0 external I/O interrupt. Each interrupt source has its own enable control bit. An interrupt event will set its individual flag. If the corresponding interrupt enable bit has been set, it would trigger CPU. F/W must clear the interrupt event register while serving the interrupt routine.

Adr	
00	Reset Vector
01	USB Endpoint 0 SET0 Receive Interrupt
02	USB Endpoint 0 OUT Receive Interrupt
03	USB Endpoint 0 Transmit Interrupt
04	USB Endpoint 1 Transmit Interrupt
05	USB Endpoint 2 Transmit Interrupt
06	USB Suspend Interrupt
07	USB Endpoint 3 Bulk Transmit Interrupt
08	USB Endpoint 4 Bulk Receive Interrupt
09	USB Bus Reset Interrupt
0a	USB Resume Interrupt
0b	Wakeup Timer Interrupt
0c	Timer0 Interrupt
0d	PB0 External I/O Interrupt

4. Wakeup Timer and Watch Dog Timer

The WKT and WDT use the same internal RC (WRC). This internal RC (WRC) can be disabled by setting R06[7] “High” for power saving. The overflow period of WDT can be selected from 17.5 ms to 140 ms and the wakeup period of WKT can be selected from 140 ms to 1120 ms. The WDT is enabled and cleared by the CLRWDT instruction. Once the WDT is enabled, the WDT generates the chip reset signal when WDT overflows. The WKT generates overflow time out interrupt if the corresponding WKT interrupt enable bit is set “High”. The WKT works in both normal mode and Power Down mode. WDT does not work in Power Down mode, it is only designed to prevent F/W goes into endless loops.



5. Timer

5.1 Timer0: 8-bit Timer 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 automatically reloads a new “offset value” (T0RLD) while it rolls over based on the pre-scaled instruction clock. The Timer0 increase rate is determined by “Timer0 Pre-Scale” (T0PSC) register in R-Plane. The Timer0 can generate interrupt (TM0I).

Timer0 overflow Period is determined by the equation shown below

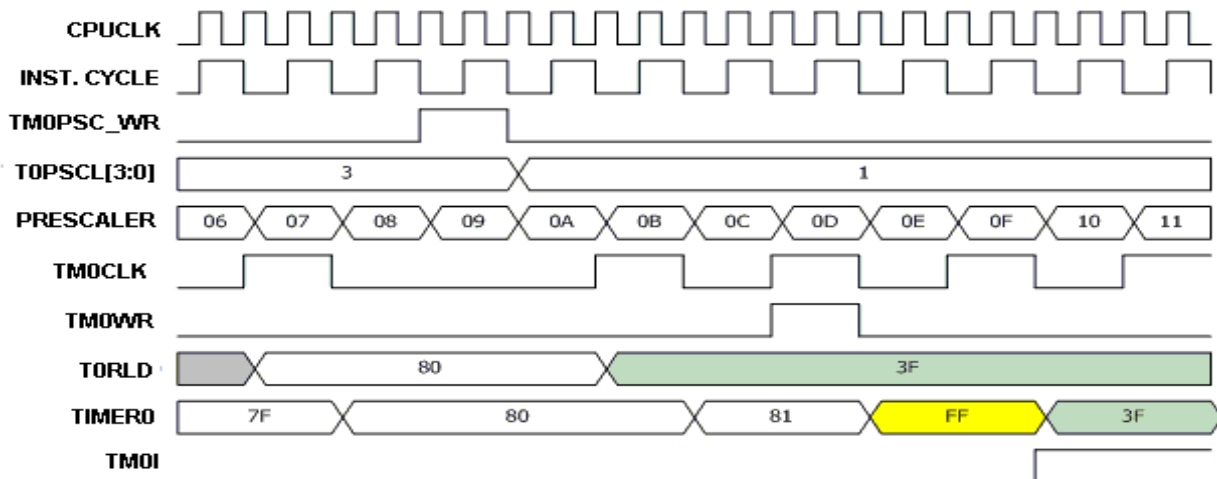
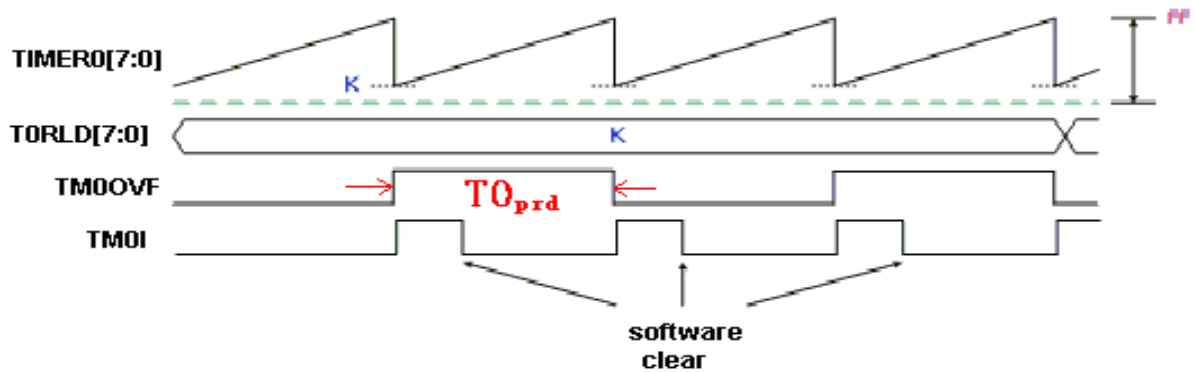
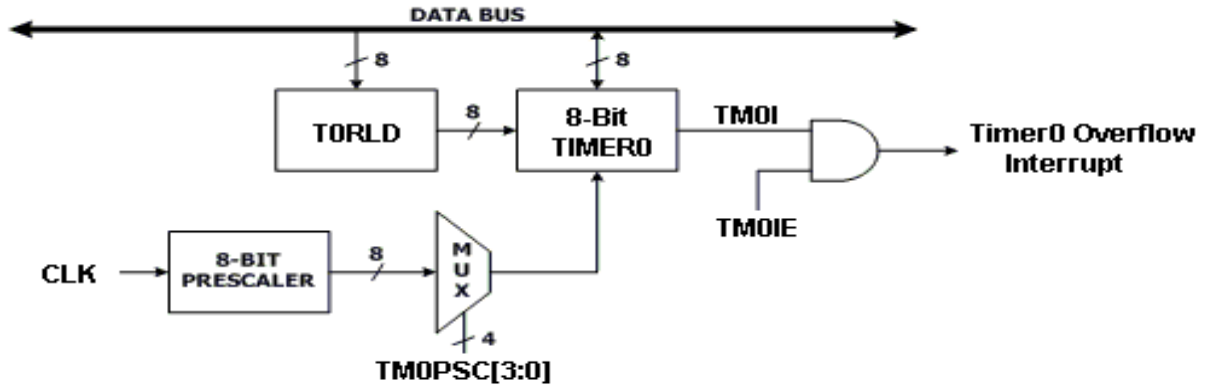
$$T0_{prd} = (2 * T_{cpu} * 2^{T0PSC}) * (256 - T0RLD)$$

For example,

ex1. CPU clock is 12 MHz (83.33 ns), T0RLD=0x78(120), T0PSC=0x4

$$\text{Timer0 overflow period} = (83.33 * 2 * 2^4) * (256 - 120) = 362652 \text{ ns} \approx 362.65 \text{ us}$$

ex2. CPU clock is 12 MHz (83.33 ns), T0RLD=0xe0(224), T0PSCL=0x6
 Timer0 overflow period = $(83.33 * 2 * 2^6) * (256-224) = 341319.7 \text{ ns} \approx 341.32 \text{ us}$



6. SPI (Serial Peripheral Interface)

The SPI Interface can be used to communicate with external peripheral devices such as flash or EEPROM memory device. This SPI module can be used as master only. The clock rate and data transfer length are also adjustable.

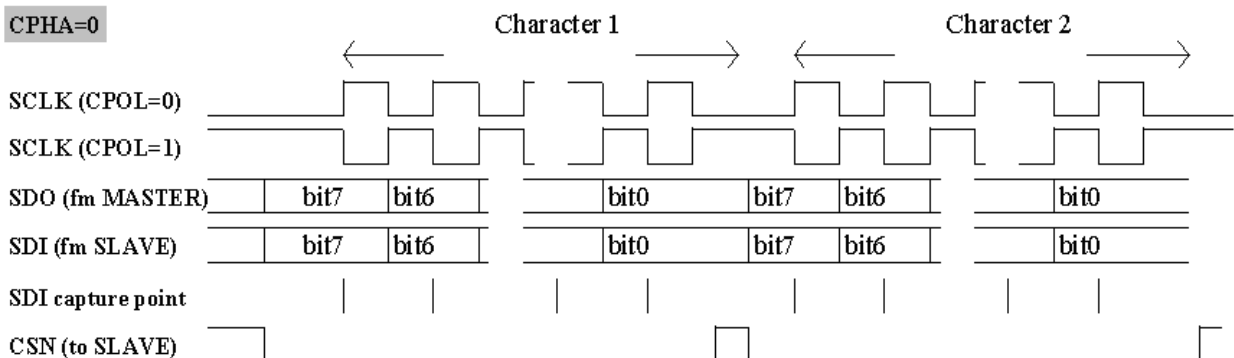
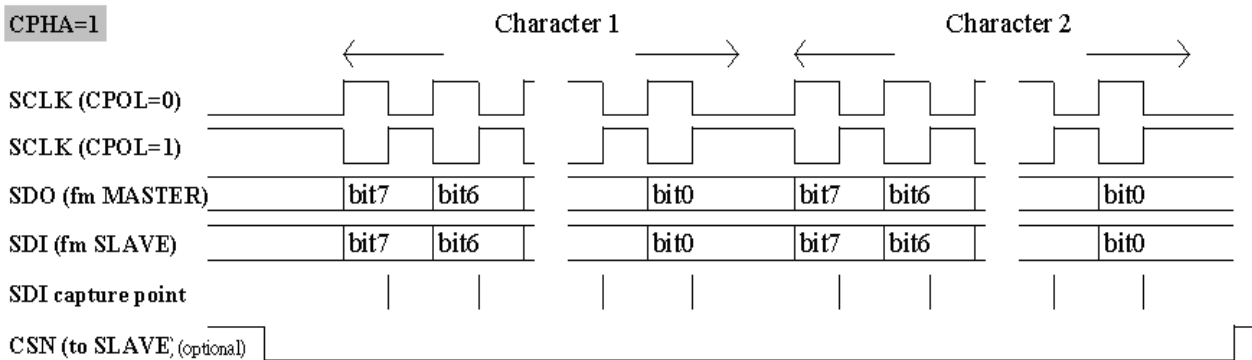
$$\text{SPI clock rate} = \text{CPUCLK}/2 * (\text{CRS}+1)$$

CRS[6:0]	SPI Clock Rate
0	6 Mbps
3	1.5 Mbps
15	375 kbps

Note: CPUCLK = 12 MHz

All the registers must be set before F1D.4 (SPI_EN) bit has been set. There are two data transfer mode. One is command phase mode; in this mode, the data transfer length is “1” and the data must be preset in R3E. The other one is data phase; in this mode, data transfer length is according to how many bytes data will be transferred. The length value is stored in R3D and the transfer data is stored in the SRAM (SRAM1 or SRAM2).

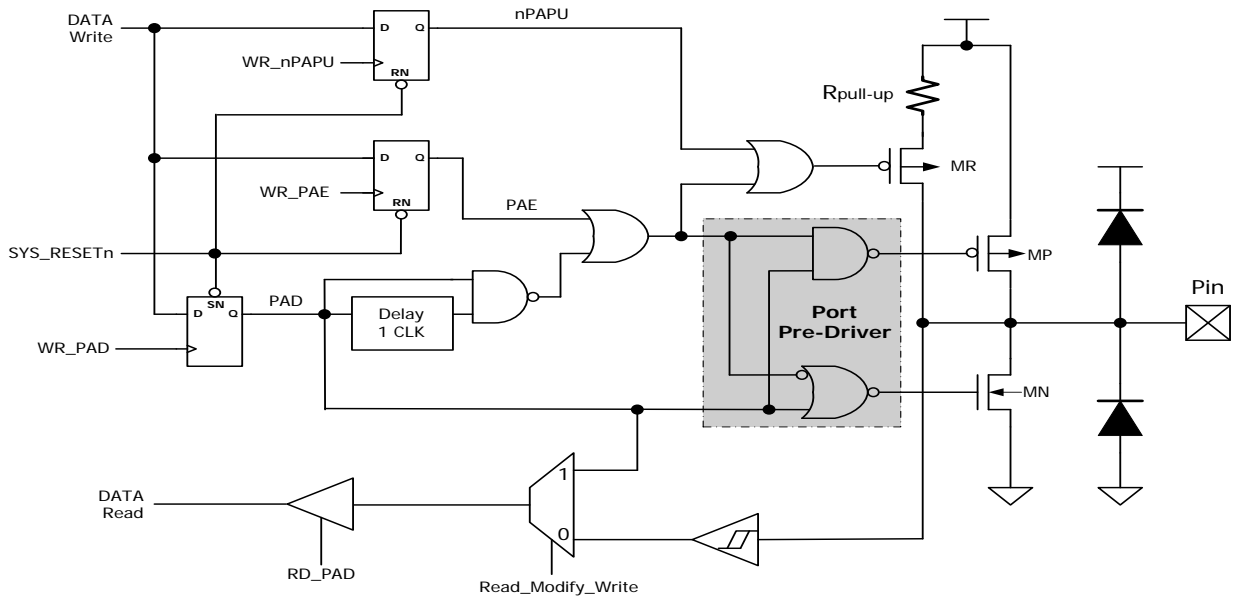
SPI Timing



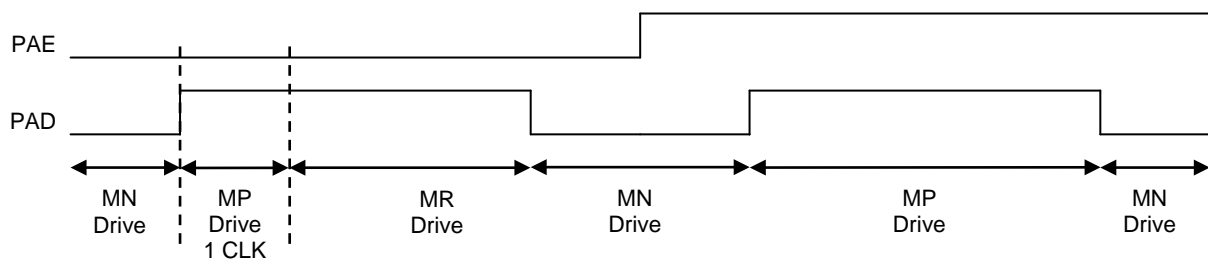
7. I/O Port

7.1 PA0-7

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 other instructions, CPU reads the pin state. The so-called "Read-Modify-Write" instruction includes BSF, BCF and all instructions using F-Plane as destination.

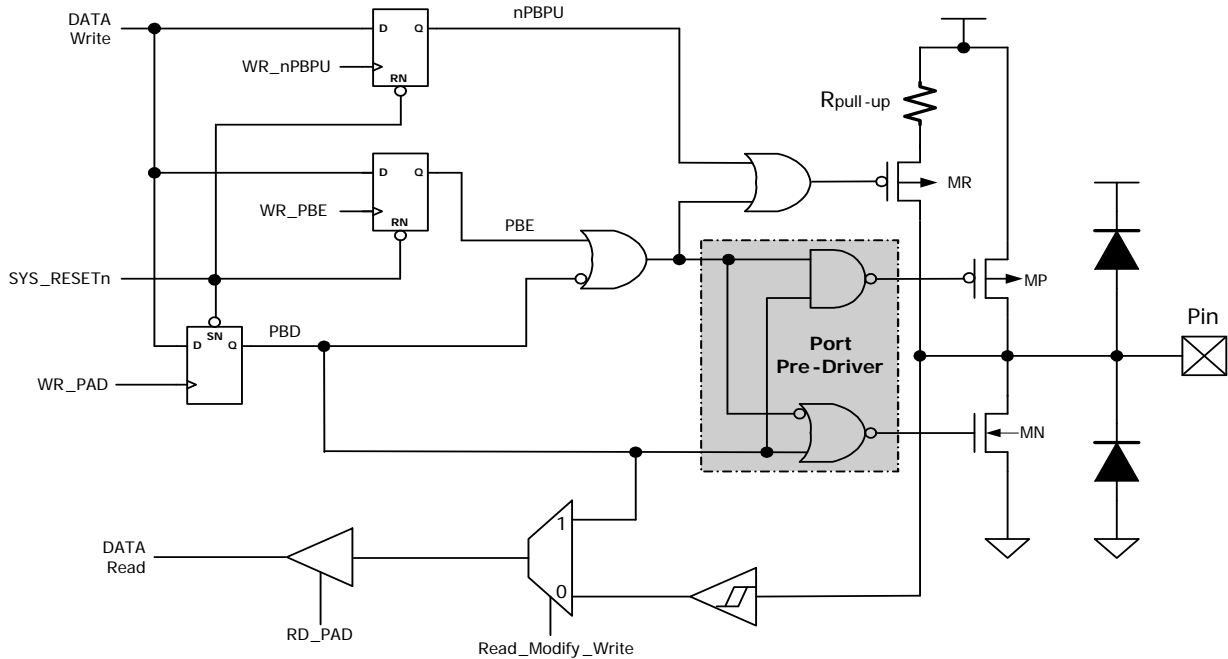


PA0-7, nPAPU=0

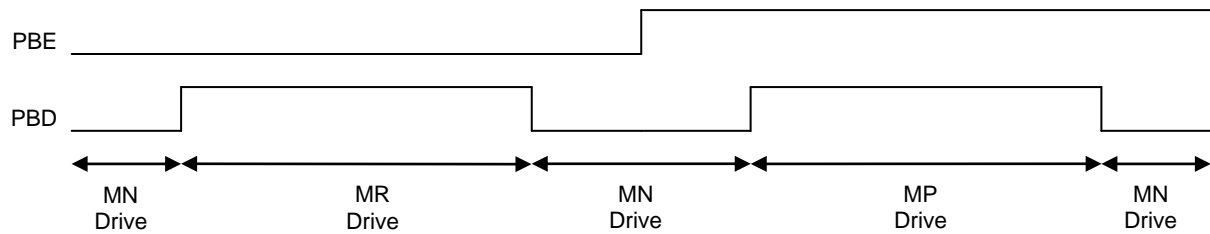


7.2 PB0

This pin is almost same as PA0-7, except they do not support pseudo-open-drain mode. They can be used in pure open-drain mode, instead.

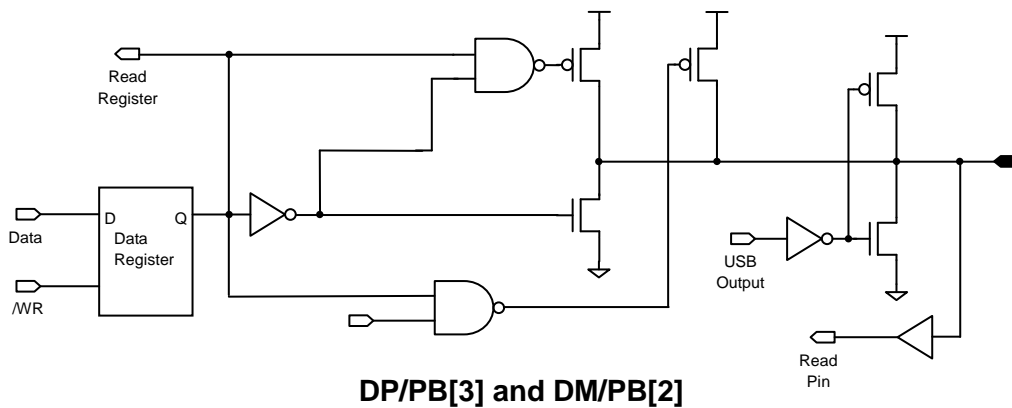


PB0, nPBPU=0



7.3 PB3 (DP) and PB2 (DM)

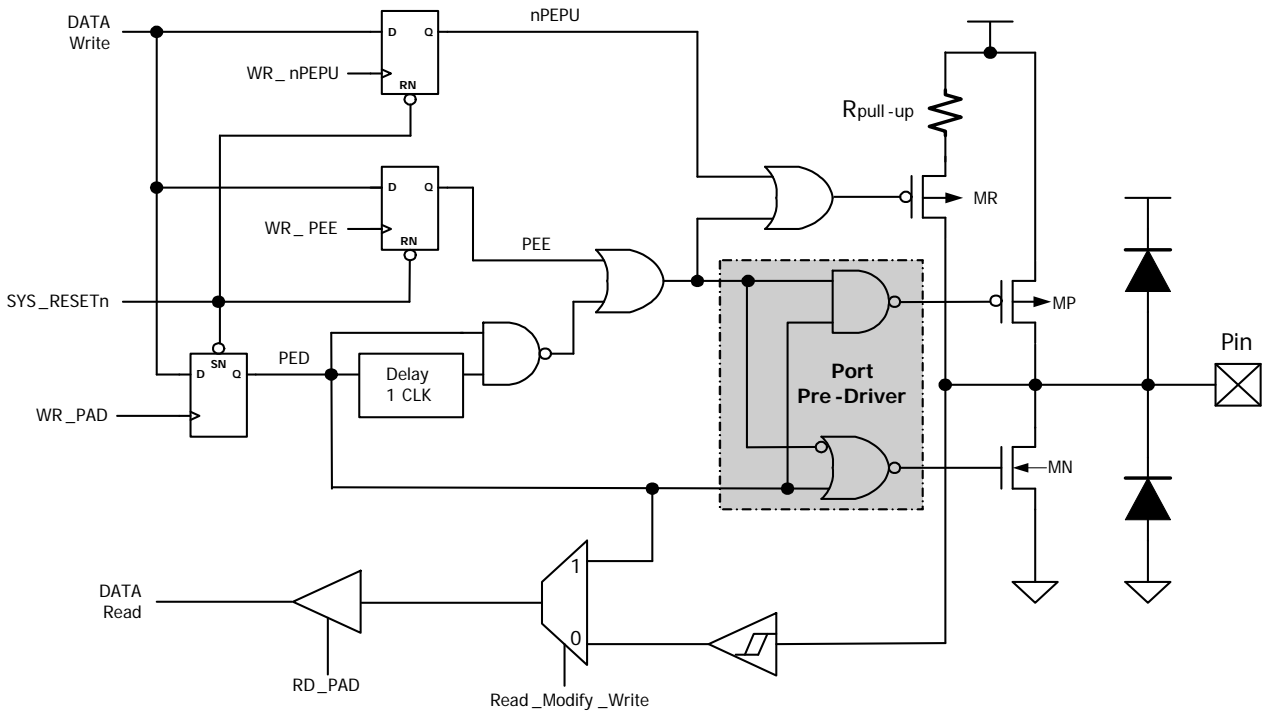
These pins are similar to PB[1:0], except they share the pin with USB function.



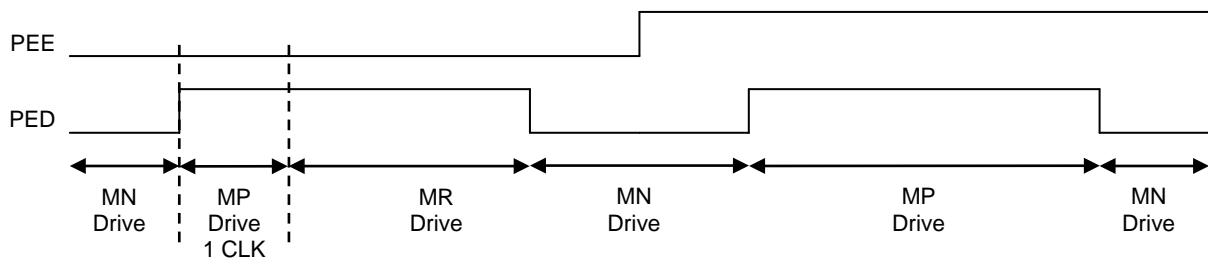
7.4 PE0-4

PortE pins are almost the same as PA0-7, except the pull-up enable bit. There are 8 different pull-up enable bits nPAPU[7:0] to control PortA. Only one pull-up enable bit nPEPU is used to control PortE. By setting Register R09[1], PE3 can be configured as clock output. R09[0] is used to determine the output clock rate.

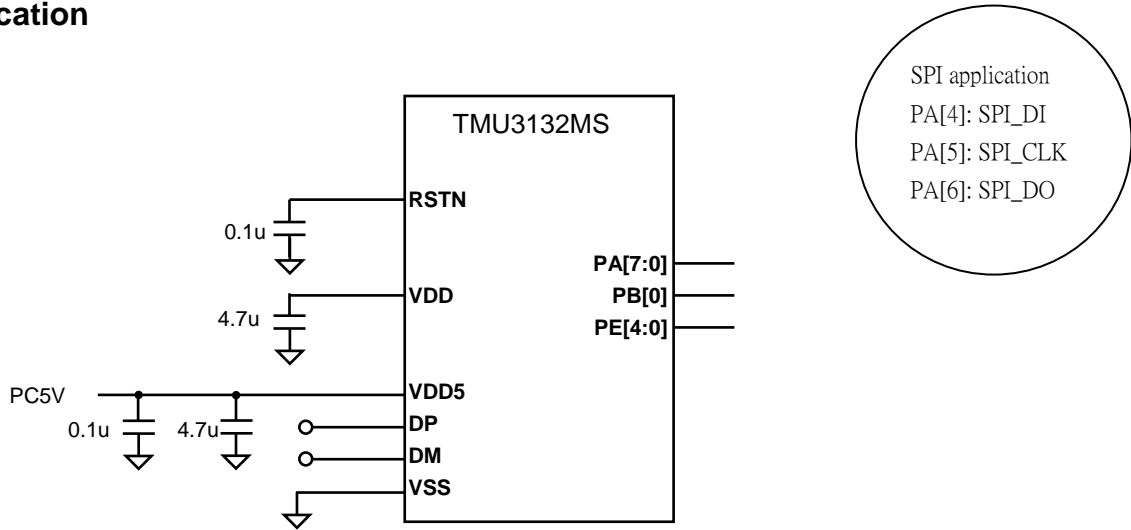
R09[1:0] = 2'b0x	PE3 is used as General Purpose IO
R09[1:0] = 2'b10	PE3 can output 12 MHz clock
R09[1:0] = 2'b11	PE3 can output 6 MHz clock



PE0-4, nPEPU=0



8. Application



9. Electrical Characteristics

ABSOLUTE MAXIMUM RATINGS

GND= 0V

Name	Symbol	Range	Unit
Maximum Supply Voltage	VDD5	-0.3 to 5.5	V
Maximum Input Voltage	Vin	-0.3 to VDD+0.3	V
Maximum output Voltage	Vout	-0.3 to VDD+0.3	V
Maximum Operating Temperature	Topg	-40 to +85	°C
Maximum Storage Temperature	Tstg	-65 to +150	°C

RECOMMENDED OPERATING CONDITION

at Ta=-20°C to 70°C, GND= 0V

Name	Symb.	Min.	Typical	Max.	Unit	Condition
Supply Voltage	VDD5	4.5	5	5.5	V	
VDD output voltage	VDD		3.3		V	VDD5=5V
Input "H" Voltage	Vih	0.6VDD			V	
Input "L" Voltage	Vil			0.3VDD	V	
Operating Voltage for I/O Ports	Vddio		3.3			
3.3V Regulator driving capacity	Ireg		50		mA	

DC CHARACTERISTICS

at Ta=-25

°C, VDD5=5.0V, VSS= 0V, Fcpu=12 MHz

Name	Symb.	Min.	Typ.	Max.	Unit	Condition
Internal Clock	F _{48m}		48		MHz	Enable IRC, VDD5=5V
Operating current	Icc		5		mA	CPU clock=12 MHz

Name	Symb.	Min.	Typ.	Max.	Unit	Condition
Suspend current	Ipd		360	500	uA	USB Mode, No load
Output High Current (Push Pull Mode)	Ioh1		11		mA	VDD5=5V, Voh1=2.8V
Output High Current (Pseudo Open Drain Mode)	Ioh2		13		uA	VDD5=5V, Voh2=2.8V
Output Low Current (Push Pull Mode)	Iol1		17		mA	VDD5=5V, Vol1=0.3V
Output Low Current (Pseudo Open Drain Mode)	Iol2		16		mA	VDD5=5V, Vol2=0.3V
Pull-Up Resistor	Rpull-up		110		KΩ	VDD5=5V
Input Leakage Current (pin high)	Iilh			1	uA	Vin=VDD
Input Leakage Current (pin low)	Iill			-1	uA	Vin=0V
System Clock Frequency (CPU clock Frequency)	Fcpu		12		MHz	R07[1:0]=2'b00
			6		MHz	R07[1:0]=2'b01
			3		MHz	R07[1:0]=2'b10
			1.5		MHz	R07[1:0]=2'b11
LVR reference Voltage	Vlvr		2.0		V	Fcpu=1.5 MHz
WDT time	Twdt		16		ms	VDD5=5V, WRC enable R06[6:5]=2'b00
			32		ms	VDD5=5V, WRC enable R06[6:5]=2'b01
			64		ms	VDD5=5V, WRC enable R06[6:5]=2'b10
			128		ms	VDD5=5V, WRC enable R06[6:5]=2'b11
WKT Time	Twkt		128		ms	VDD5=5V, WRC enable R06[4:3]=2'b00
			256		ms	VDD5=5V, WRC enable R06[4:3]=2'b01
			512		ms	VDD5=5V, WRC enable R06[4:3]=2'b10
			1020		ms	VDD5=5V, WRC enable R06[4:3]=2'b11

AC CHARACTERISTICS

at Ta=25

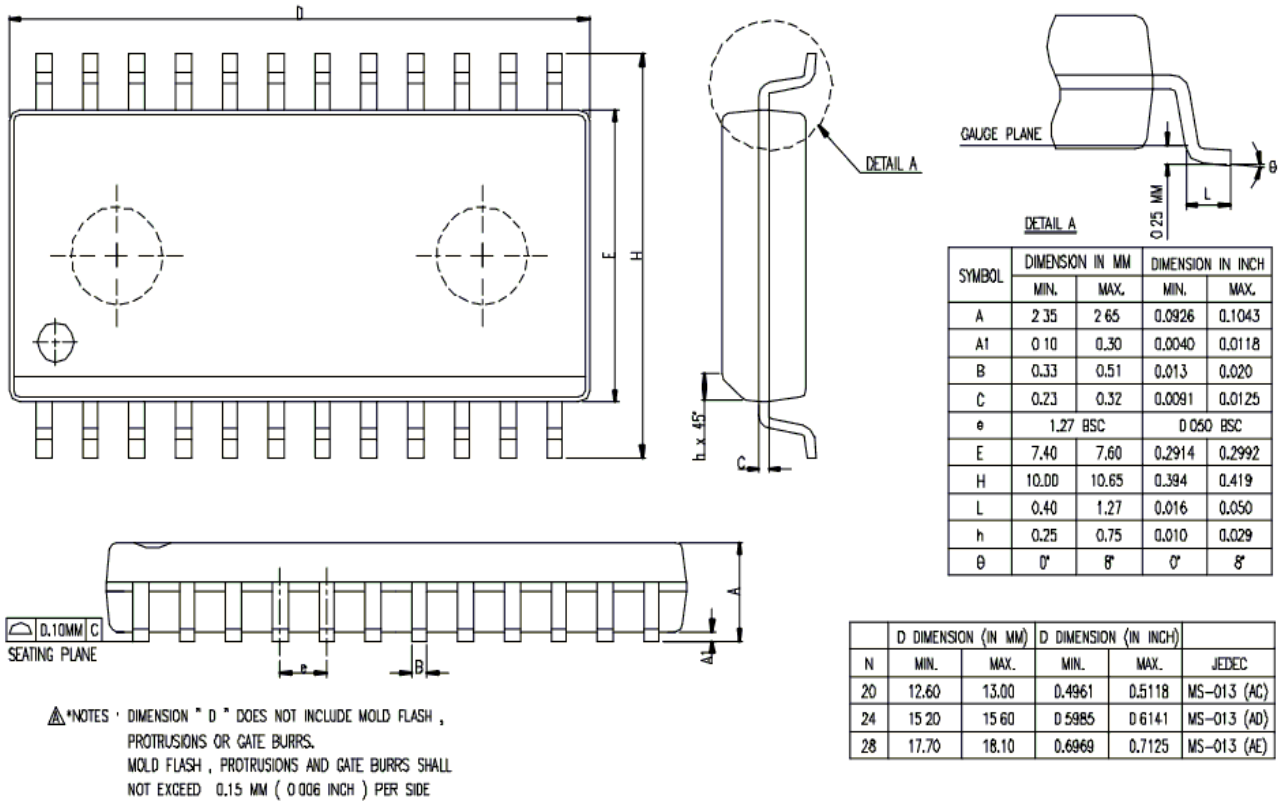
°C, VDD5V=5.0V, VSS= 0V, Fcpu=12 MHz

Name	Symb.	Min.	Typ.	Max.	Unit	Note
DP/DM rising time	Trise	4		20	ns	
DP/DM falling time	Tfall	4		20	ns	
DP, DM cross point	Vx	1.3		2.0	V	

Note: All USB transceiver characteristics can meet USB1.1 spec.

10. Package Information

● SOP 20



● SOP18

