TMU3131

DATA SHEET

Preliminary Rev D1.0

Amendment History

D1.0 2012/06/06 NEW

Information

SPEC Overview

Microcontroller Features Table

CPU	ROM	RAM Bytes	OSC Dual Source	Instruction Set	Instruction Time	Stack level	Interrupt vector	Timers	LVR
RISC	MTP 6K*14	160 + 128	IRC 48 MHz	37	2T	8	17	T0/T1	2.0V

IRC Frequency Features Table

Operation Frequency	Internal RC 48 MHz	12 MHz	6 MHz	3 MHz	1.5 MHz
Battery mode (Stand alone)	+/- 3%	+/- 3%	+/- 3%	+/- 3%	+/- 3%
USB mode (USB Plug in)	+/- 0.25%	+/- 0.25%	+/- 0.25%	--	--

Power Features Table

Operation Voltage	Operation Current (12 MHz)	Operation Current (6 MHz)	Operation Current (3 MHz)	Operation Current (1.5 MHz)	Operation Temperature
2.2 ~ 5.5 V Battery mode	15 mA (WKT ON)	7 mA (WKT ON)	5 mA (WKT ON)	4 mA (WKT ON)	−40 to +85

Peripheral Features Table

USB control	Interrupt mode	Bulk mode	Touch Key	PWM Output	SPI Interface
EP0 each 8-Byte	EP1 & EP2 8-Byte	EP3 & EP4 64-Byte	Capacitive	8-Bit	Master only
			5-ch	CPUCLK	DMA R/W

TMU313 Series Family Types

P/N	Program ROM	RAM bytes	EP	GPIO	PWM	Touch key	CPU Clock	I80	SPI Master
TMU3130	8K*14 Flash	160 +128	5	36	1	5	IRC / 6 MHz	Yes	Yes
TMU3131	6K*14 MTP	160 +128	5	17	1	5	IRC	—	Yes
TMU3132	4K*14 MTP	160 +128	5	16	—	—	IRC	—	Yes
TMU3132MS	4K*14 MASK	160 +128	5	16	—	—	IRC	—	Yes

Features

RISC CPU:

- **Only 37 instructions**
- **Instruction Execution Time**
  2-cycle instructions except branch
- **Operating clock**
  - Fast Clock:
    - In Battery mode IRC 24 MHz +/-3%
    - In USB mode IRC 48 MHz +/-0.25%
    - CPU Clock control:
      12 MHz / 6 MHz / 3 MHz / 1.5 MHz
  - Clock Output:
    - 6 MHz / 12 MHz Output
- **6Kx14 internal flash Program Memory**
- **Memory**
  - 160 bytes on F-plane
  - 128 bytes on R-plane
  - 8 bytes *5 on R-plane
- **8-level Stack**
- **Interrupt**
  - 16 kinds of interrupt vector
    - USB EP0 SET0 Receive Interrupt
    - USB EP0 OUT Receive Interrupt
    - USB EP0 Transmit Interrupt
    - USB EP1 Transmit Interrupt
    - USB EP2 Transmit Interrupt
    - USB Suspend Interrupt
    - USB EP3 Bulk Transmit Interrupt
    - USB EP4 Bulk Transmit Interrupt
    - USB Bus Reset Interrupt
    - USB Resume Interrupt
    - Wake-up Timer Interrupt
    - Timer0 Interrupt
    - PB0 External I/O Interrupt
    - VDD5V Rise interrupt
    - Timer1 Interrupt
  - Automatic Store/Restore W and STATUS

Power Features

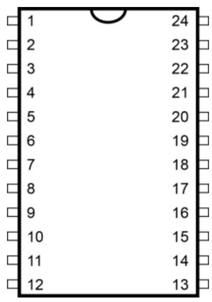
- **Operation Voltage**
  - Low Voltage Reset Voltage to 5.5V
  - Maximum Operation range 2.1V to 5.5V
    Built-in 3.3V Regulator covers 3.3 to 5.5V
  - Chip Operating Voltage range 2.1 to 3.6V
- **Operation Current**
  USB mode (Internal 3.3V Regulator used)
  - Normal mode
    - 5V@ 12 MHz, 4.1 mA typical
    - 5V@ 6 MHz, 2.8 mA typical
    - 5V@ 3 MHz, 2.3 mA typical
    - 5V@ 1.5 MHz, 1.8 mA typical
  - Suspend mode
    - 5V@ 12 MHz, 360 uA typical
  - Power down mode
    - 5V@ 1 uA typical
- **H/W Reset**
  - External active low reset (RSTN)
  - Built-in Power-On Reset (POR)
  - Low Voltage Reset - 2.1V (LVR)
  - Watchdog Reset (WDT)

Peripheral Features

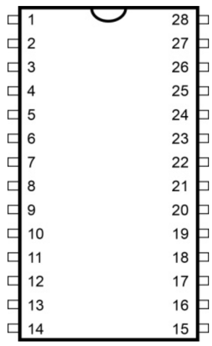
- **I/O Port:**
  - Maximum 17 programmable I/O pins
  - Pseudo-Open-Drain Output (P.O.D.)
  - Open-Drain Output (O.D.)
  - CMOS Push-Pull Output (P.P.)
  - Schmitt Trigger Input
- **Capacitive Touch Module**
  - Up to 5-channels for Touch Key
- **Timers**
  - Timer0 is 8-bit with 8-bit prescaler, Counter/Capture/Interrupt function
  - Timer1 is 16-bit with Buzzer / Capture / Reload / Interrupt function
- **PWMs**
  - Built-in 8-bit PWM generator
  - PWM0 with prescaler / period adjustment / buffer-reload / rising-falling output
- **Watchdog Timer**
  - Clocked by on-chip oscillator with 4 adjustable Reset/Interrupt time durations (112 ms / 56 ms / 28 ms / 14 ms)
  - Wake-up Timer (896 ms / 448 ms / 224 ms / 112 ms)
- **SPI Interface**
  - Master only
  - Programmable transmit bit rate
  - Serial clock phase and polarity options
  - MSB-first or LSB-first selectable

Pin Summary (SOP-28/SOP-24)

NAME	I/O	TMU3131		TMU3131S	Package
		SOP-28	SOP-24	SOP-24	
VDD5	P	17	13	14	
VSS	P	10	9	9	
PC5VIN	I	19	15	14	
VBAT	P	18	14		
VDD	O	14	11	11	
VPP/RSTn	I	12	10	10	
VSSA	P	11	9	9	
VDDA	P	13	11	11	
PA[7]	I/O	25	21	21	
PA[6] \ SPI_DO	I/O	26	22	22	
PA[5] \ SPI_CLK	I/O	27	23	23	
PA[4] \ SPI_DI	I/O	28	24	24	
PA[3]	I/O	1	1	1	
PA[2] \ TK[2]	I/O	5	4	4	
PA[1] \ TK[3]	I/O	4	3	3	
PA[0] \ TK[4]	I/O	3	2	2	
DP \ PB3 \IIC SCK	I/O	21	17	16	
DM \ PB2 \ IIC DAT	I/O	20	16	15	
PB[0] \ TK[0]	I/O	6	5	5	
PB[1] \ TK[1]	I/O	7	6	6	
PE[0]	I/O	8	7	7	
PE[1]	I/O	23	19	19	
PE[2]	I/O	9	8		
PE[3]	I/O	22	18	18	
PE[4]	I/O	24	20	20	
PE[5]	I/O	15	12		
PE[6]	I/O	16			



SOP-24



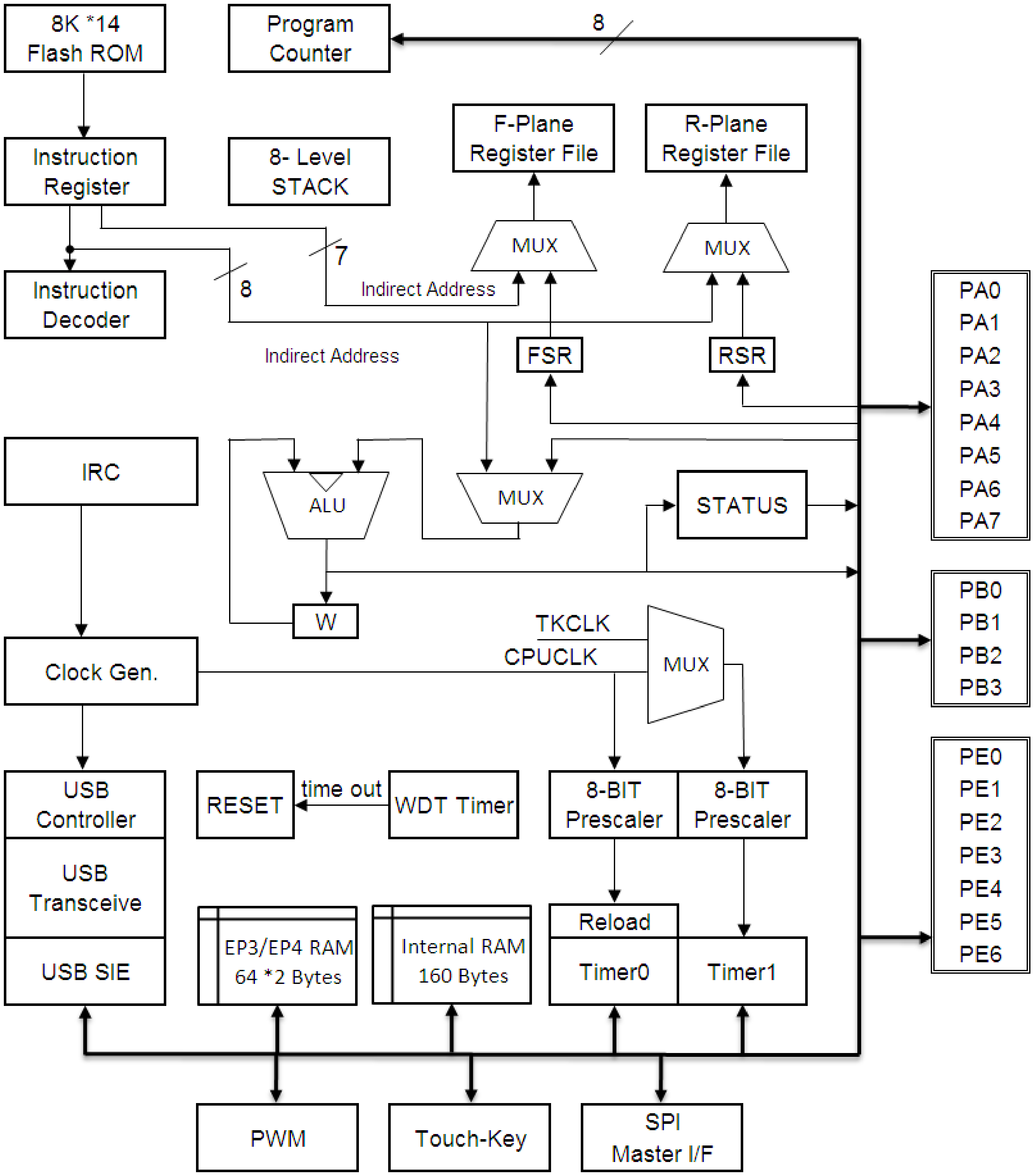
SOP-28

Pin Summary (SOP-24/SOP-28)

Pin number		Pin Name	Type	Input		Output			Func. after reset	Alternate Function		Misc
SOP-28	SOP-24			Enable Pull-up	Ext. Interrupt	O.D.	P.O.D.	P.P.		Touch-Key	SPI	
7	6	TK0 / PB1	I/O	●		●		●	PB1	●		
6	5	TK1 / PB0	I/O	●	●	●		●	PB0	●		
5	4	TK2 / PA2	I/O	●			●	●	PA2	●		
4	3	TK3 / PA1	I/O	●			●	●	PA1	●		
3	2	TK4 / PA0	I/O	●			●	●	PA0	●		
12	10	VPP / RSTN	I						RSTN			Reset
1	1	PA3	I/O	●			●	●	PA3			
8	7	PWMO / PE0	I/O	●			●	●	PE0			PWM Output
23	19	PE1	I/O	●			●	●	PE1			
9	8	PE2	I/O	●			●	●	PE2			
22	18	CLKO / PE3	I/O	●			●	●	PE3			Clock Output
24	20	PE4	I/O	●			●	●	PE4			
15	12	PE5	I/O	●			●	●	PE5			
16	-	PE6	I/O	●			●	●	PE6			
25	21	PA7	I/O	●			●	●	PA7			
26	22	SDO / PA6	I/O	●			●	●	PA6		●	
27	23	SCLK / PA5	I/O	●			●	●	PA5		●	
28	24	SDI / PA4	I/O	●			●	●	PA4		●	
17	13	VDD5	P						VDD5			
18	14	VBAT	P						VBAT			
19	15	PC5VIN	I						PC5VIN			
10	9	VSS	P						VSS			
11	9	VSSA	P						VSSA			
13	11	VDDA	P						VDDA			Analog V33
14	11	VDD	P						VDD			Digital V33
21	17	DP / SCK / PB3	I/O	●		●		●	DP			USB
20	16	DM / DAT / PB2	I/O	●		●		●	DM			USB

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

System Block Diagram



_

CONTENTS

Functional Description

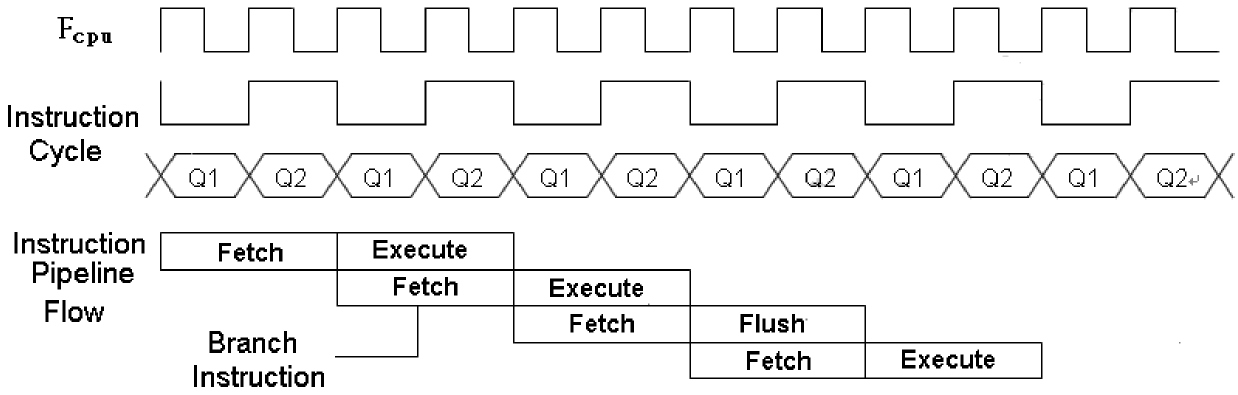
1. CPU Core

1.1 Clock Scheme and Instruction Cycle

TMU3132 has 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

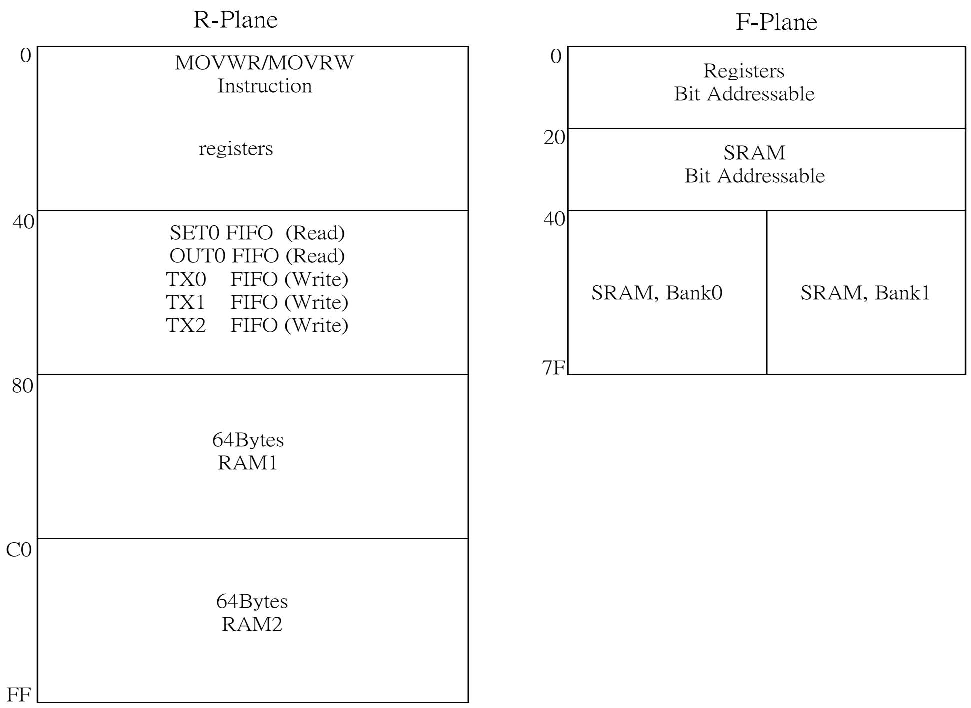
The Programming Counter is 13-bit wide capable of addressing a 6K x 14 program ROM. As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC value is normally increased by one except the followings. The Reset Vector (000h) and the Interrupt Vectors (from 00bh to 011h) are provided for PC initialization and Interrupt. For CALL/GOTO instructions, PC loads the lower 12 bits address from instruction word and MSB from Register F03[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 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.

Since the ROM size is 6K words, it means there are 13 address lines. The CALL/GOTO instructions can load 12 bits address from instruction, that means only 4K size can reach, i.e. either 000h to FFFh; or 1000h to 17FFh. One ROM page is 4K words in length, so if user needs to CALL/GOTO the other page, the ROM page bit (F03[6]) must be set/cleared according to page0 or page1 will the program counter be. Remember that ISR entry addresses are located at ROM Page0, if the user code is interrupted from ROM Page1, ROM Page bit should be cleared when CALL/GOTO will be used in Interrupt Service Routines. While exiting from ISR, user should recall the originally ROM page bit and store to Register F03[6].

1.4 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.

- 6K x 14 program MTP.
- 160-byte SRAM (F-plane) is addressed from 0x20 to 0x7F is used for CPU. The lower 32-byte (0x20 ~ 0x3f) is bit addressable. The higher address (0x40 ~ 0x7F) is separated to two banks which can be selected by register F03[5].
- Two 64-byte RAMs (R-plane).
- Five 8-byte USB FIFOs are allocated in R-plane.



1.5 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
GIE	Global Interrupt Enable Flag (i-Flag)
[]	Option Field
( )	Contents
.	Bit Field
B	Before
A	After
←	Assign direction

1.6 Instruction Table

Mnemonic		Op Code	Cycle	Flag Affect	Description
Byte-Oriented File Register Instruction					
ADDWF	f,d	00 0111 dfff ffff	1	C,DC,Z	Add W to f
ANDWF	f,d	00 0101 dfff ffff	1	Z	AND W to f
CLRF	f	00 0001 1fff ffff	1	Z	Clear f
CLRW		00 0001 0100 0000	1	Z	Clear W
COMF	f,d	00 1001 dfff ffff	1	Z	Invert F bit by bit
DECF	f,d	00 0011 dfff ffff	1	Z	Decrement of f
DECFSZ	f,d	00 1011 dfff ffff	1 or 2	-	Decrease f, skip if zero
INCF	f,d	00 1010 dfff ffff	1	Z	Increment of f
INCFSZ	f,d	00 1111 dfff ffff	1 or 2	-	Increase f, skip if zero
IORWF	f,d	00 0100 dfff ffff	1	Z	OR W to f
MOVFW	f	00 1000 0fff ffff	1	-	Move f to W
MOVWF	f	00 0000 1fff ffff	1	-	Move W to f
MOVRW	r	01 1111 rrrr rrrr	1	-	Move r to W
MOVWR	r	01 1110 rrrr rrrr	1	-	Move W to r
RLF	f,d	00 1101 dfff ffff	1	C	F rotate to left
RRF	f,d	00 1100 dfff ffff	1	C	F rotate to right
SUBWF	f,d	00 0010 dfff ffff	1	C,DC,Z	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
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

1.7 Addressing Mode

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

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

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

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

–

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

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

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

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

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

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

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

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

–

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

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

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

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

_

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

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

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

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

–

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

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

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

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

–

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

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

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

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

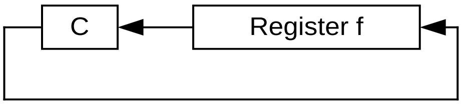
–

RET — Return from Subroutine

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

RLF — Rotate Left f through Carry

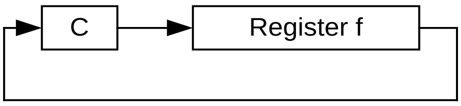
Syntax: RLF f [,d]
Operands: f : 00h ~ 7Fh, d : 0, 1
Operation:



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

RRF — Rotate Right “f” through Carry

Syntax: RRF f [,d]
Operands: f : 00h ~ 7Fh, d : 0, 1
Operation:



Status Affected: C
OP-Code: 00 1100 dfff ffff
Description: The contents of register ‘f’ are rotated one bit to the right through the Carry Flag. If ‘d’ is 0, the result is placed in the W register. If ‘d’ is 1, the result is placed back in register ‘f’.
Cycle: 1
Example: RRF REG1,0
B : REG1 = 1110 0110, C = 0
A : REG1 = 1110 0110
W = 0111 0011, C = 0

–

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

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

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

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

–

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

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

2. Control Registers

2.1 F-Plane Table

Addr	RST	NAME	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00h	xxxx-xxxx	INDF	Addressing INDF uses contents of FSR to address data memory							
01h	0000-0000	TM0	Timer 0 Counter							
02h	0000-0000	PC	Program Counter [0..7]							
03h	0000-0000	STATUS	--	ROMPAGE	RAMBANK	--	--	ZFLAG	DCFLAG	CFLAG
04h	0000-0000	FSR	F-Plane File Select Register							
05h	0000-0000	RSR	R-Plane File Select Register							
06h	1111-1111	PAD	PA[0..7] Port A Pin data output register							
07h	1111-1111	PBD	PB[0..3] Port B Pin data output register							
0ah	x111-1111	PED	PE[0..6] Port E Pin data output register							
0dh	0000-0000	TM1	TIMER 1 Counter							
0eh	xxxx-xxx0	TM1IE	TIMER1 Interrupt Enable							TM1IE
0fh	xxxx-xxx0	TM1IF	TIMER1 Interrupt flag, write 0 to clear it							TM1I
10h	0000-0000	USBE	USBE	FUNADR						
11h	0000-0000	INTFLAG 1	SET0I	OUT0I	TX0I	TX1I	TX2I	SUSPI	TX3I	RC4I
12h	x000-0000	INTFLAG 2	--	VDD5VRI	WKTI	RSTI	RSMI	--	PB0I	TM0I
13h	0000-0xx0	EPCFG	SUSP	RSMO	EP1CFG	EP2CFG	DEVR	--	--	OUT0RDY
14h	0000-0000	EP0SET	TX0RDY	TX0TGL	EP0STAL L	IN0STAL L	TX0CNT[0..3]			
15h	000x-0000	EP1SET	TX1RDY	TX1TGL	EP1STAL L	--	TX1CNT[0..3]			
16h	000x-0000	EP2SET	TX2RDY	TX2TGL	EP2STAL L	--	TX2CNT[0..3]			
17h	000x-0000	EP3SET	TX3RDY	TX3TGL	EP3STAL L	EP3CFG	--			
18h	0x00-0xxx	EP4SET	RC4RDY	RC4TGL	EP4STAL L	EP4CFG	RC4ERR	--		
19h	x000-0000	TX3CNT	Endpoint 3 transmit byte count							
1ah	x000-0000	RC4CNT	Endpoint 4 transmit byte count							
1ch	xx00-00xx	XRAMCO N	--		SRAM1USB	SRAM2US B	SRAM1SP I	SRAM2SPI	--	--
1dh	xx00-0000	SPISET	--	--	SPIMODE	SPIEN	LSBFIRST	SPIIN	SPISW	CLRADR
20h : 7fh	xxxx-xxxx	BANK0	Internal RAM 96 Bytes (BANK0)							
	xxxx-xxxx	BANK1	Internal RAM 96 Bytes (BANK1)							

2.2 F-Plane Description

	Name	Address	R/W	Rst	Description
F01	TM0	01.7~0	R/W	0	Timer 0
F02	PC	02.7~0	R/W	0	Program Counter [7~0]
F03	ROMPAGE	03.6	R/W	0	Program ROM Page Select
	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];
F08	PCD	08.7~0	R/W	ff	Port C output data; Key Scan input [7~0];
F09	PDD	09.7~0	R/W	ff	Port D output data, PDD[7:0]; Key Scan output, KSO[7:0]
F0A	PED	0A.7~0	R/W	f	Port E output data, PED[7:0]; Key Scan output, KSO[15:8]
F0D	TM1	0D.0~7	R/W	0	Timer1
F0E	TM1IE	0E.0	R/W	0	Timer1 Interrupt Enable
F0F	TM1IF	0F.0	R/W	0	Timer1 Interrupt flag, write 0 to clear it
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		Endpoint 0 OUT Receive Interrupt flag, write 0 to clear flag.
	TX0I	11.5	R/W	0	Endpoint 0 Transmit Interrupt flag, write 0 to clear flag.
	TX1I	11.4	R/W	0	Endpoint 1 Transmit Interrupt flag, write 0 to clear flag.
	TX2I	11.3	R/W	0	Endpoint 2 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	VDD5VRI	12.6	R/W	0	VDD5V Rise Interrupt flag, write 0 to clear it
	WKTI	12.5	R/W	0	Wakeup Timer Interrupt flag, write 0 to clear flag
	RSTI	12.4	R/W	0	USB Bus Reset Interrupt flag, write 0 to clear flag.
	RSMI	12.3	R/W	0	USB Resume Interrupt flag, write 0 to clear flag.
	PB0I	12.1	R/W	0	PB0 Interrupt flag, write 0 to clear flag.
	TM0I	12.0	R/W	0	Timer0 Interrupt flag, write 0 to clear flag.
F13	SUSP	13.7	R/W	0	S/W force USB interface into suspend mode.
	RSMO	13.6	R/W	0	S/W force USB interface send RESUME signal in suspend
	EP1CFG	13.5	R/W	0	Set Endpoint 1 configured.
	EP2CFG	13.4	R/W	0	Set Endpoint 2 configured.
	DEVR	13.3	R/W	0	DP Pull-up resistor enable bit, 0: Disable pull-up , 1: pull-up
	OUT0RDY	13.0	R/W	0	Endpoint 0 ready for receive, cleared by H/W while OUT0I
F14	TX0RDY	14.7	R/W	0	Endpoint 0 ready for transmit, cleared by H/W while TX0I
	TX0TGL	14.6	R/W	0	Endpoint 0 transmit 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 transmit byte count.

–

	Name	Address	R/W	Rst	Description
F15	TX1RDY	15.7	R/W	0	Endpoint 1 ready for transmit, cleared by H/W while TX1I
	TX1TGL	15.6	R/W	0	Endpoint 1 transmit DATA1/DATA0 packet.
	EP1STALL	15.5	R/W	0	Endpoint 1 will stall IN packet.
	TX1CNT	15.3~0	R/W	0	Endpoint 1 transmit byte count.
F16	TX2RDY	16.7	R/W	0	Endpoint 2 ready for transmit, cleared by H/W while TX2I
	TX2TGL	16.6	R/W	0	Endpoint 2 transmit DATA1/DATA0 packet.
	EP2STALL	16.5	R/W	0	Endpoint 2 will stall IN packet.
	TX2CNT	16.3~0	R/W	0	Endpoint 2 transmit byte count.
F17	TX3RDY	17.7	R/W	0	Endpoint 3 ready for transmit, cleared by H/W while TX3I
	TX3TGL	17.6	R/W	0	Endpoint 3 transmit 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
	RC4TGL	18.6	R	-	Endpoint 4 received 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 received data error.
F19	TX3CNT	19.6~0	R/W	0	Endpoint 3 transmit byte count.
F1A	RC4CNT	1A.6~0	R	0	Endpoint 4 transmit byte count.
F1C	XRAMCON	1C.5~0	R/W	0	XRAM 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	SPIMODE	1D.5	R/W	0	SPI MODE
	SPIEN	1D.4	R/W	0	SPI Enable, Busy bit
	LSBFIRST	1D.3	R/W	0	1:Data transmit/Receive is LSB first; 0: MSB first
	SPIIN	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
	SPISW	1D.1	R/W	0	SPI CMD/DAT Switch; 1:CMD, 0:DAT
	CLRADR	1D.0	R/W	0	Write 1 to clear ram address

2.3 R-Plane Register Table

Addr	RST	NAME	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00h	xxxx-xxxx	INDR	Addressing INDR uses contents of RSR to address data memory							
01h	0000-0000	TM0RLD	Timer0 overflow reload value							
02h	0000-0000	TM0SET	--			TM0EN	TM0PSC[0..3]			
03h	xxxx-xxxx	PWRDOWN	Write this Register to enter Power-Down Mode							
04h	0000-0000	WDTE	Write this register to clear WDT and enable WDT							
06h	0000-000x	WRCPD	WRCPD	WDTPSC		WKTPSC		5VINFLG	OUTFLG	--
07h	xx01-1000	CLKSEL	--	--	HWAUTO	--	--	--	CLKDIV	
09h	xxxx-xx00	IRCCKO	--						PE3CKO	PE3SEL
0ah	xx00-0000	TM1EN	--	--	TM1EN	TM1SEL	TM1PSC			
0eh	x000-x000	TKE	--	TKE	TKSPD[1..0]		--	TKSEL[2..0]		
11h	0000-0000	USBINTEN	SET0IE	OUT0IE	TX0IE	TX1IE	TX2IE	SUSPIE	TX3IE	RC4IE
12h	x000-0000	FUNINTEN	--	VDD5VIE	WKTIE	RSTIE	RSMIE	--	PB0IE	TM0IE
13h	xxxx-xxxx	RC0SET	RC0TGL	RC0ERR	EP0DIR	EP0IND	OUT0CNT			
20h	0000-0000	PAE	PA[0..7] CMOS push-pull output enable							
21h	xxxx-0000	PBE	PB[0..3] CMOS push-pull output enable							
24h	x000-0000	PEE	PE[0..6] CMOS push-pull output enable							
25h	0000-0000	PAPU	PA[0..7] pull-up , enable=0							
26h	xxxx-0000	PBPU	PB[0..3] pull-up , enable=0							
27h	xxxx-xxx0	PCDEPU	--					--	--	PEPU
30h	xxxx-x000	PWMENF	--					PWMPSC		PWMEN
31h	0000-0000	PWMDUTY	PWM DUTY							
32h	0000-0000	PWMPRD	PWM Period							
3bh	xx00-0000	SPIENF	--	--	CPOL	CPHA	BSL[0..3]			
3ch	x000-0000	SPICRS	--	SPICRS[0..6] SPI Clock Select						
3dh	x000-0000	SPILEN	--	SPILEN[0..6] SPI Transfer length						
3eh	0000-0000	SPITX	SPITX[0..7] SPI Transmit DATA in CMD phase							
3fh	xxxx-xxxx	SPIRX	SPIRX[0..7] SPI Received DATA							
40h : 47h	xxxx-xxxx	SET0FIFO	Endpoint 0 SETUP Receive Buffer (8 Bytes)							
48h : 4fh	xxxx-xxxx	OUT0FIFO	Endpoint 1 OUT Receive Buffer (8 Bytes)							
50h : 57h	xxxx-xxxx	TX0FIFO	Endpoint 0 Transmit Buffer (8 Bytes)							
58h : 5fh	xxxx-xxxx	TX1FIFO	Endpoint 1 Transmit Buffer (8 Bytes)							
60h : 67h	xxxx-xxxx	TX2FIFO	Endpoint 2 Transmit Buffer (8 Bytes)							
80h : bfh	xxxx-xxxx	XRAM1	Endpoint 3/4 Buffer (64 Bytes)							
c0h : ffh	xxxx-xxxx	XRAM2	Endpoint 3/4 Buffer (64 Bytes)							

2.4 R-Plane Description

	Name	Address	R/W	Rst	Description
R01	T0RLD	01.7~0	W	0	Timer0 overflow reload value
R02	TM0EN	02.4	W	0	Timer0 Enable
	TM0PSC	02.3~0	W	0	Timer0 Pre-Scale, 0:div1, 1:div2, 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	WRCPD	06.7	W	0	WRC Disable, 1: Disable WRC, 0: Enable WRC
	WDTPSC	06.6~5	W	11	WDT period, 00=15 ms, 01=30 ms, 10=60 ms, 11=120 ms
	WKTPSC	06.4~3	W	11	WKT period, 00=120 ms, 01=240 ms, 10=480 ms, 11=960 ms
	5VINFLG	06.2	R	0	PC5V status, 1:PC5V High 0:PC5V LOW (USB plug-in flag)
	OUTFLG	06.1	R/W	0	USB plug-out flag, write 0 or RST_P=1 to clear flag
R07	HWAUTO	07.5	W	0	H/W auto push/pop during INT process
	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	PE3CKO	09.1	W		PE3 is used as IO port or IRC CKO; 0: IO port, 1: IRC CKO
	PE3SEL	09.0			When EN_PE3_CKO= 1, IRCCKO_SEL is used to select IRC clock frequency 0: IRC 12 MHz output, 1: IRC 6 MHz output
R0A	TM1EN	0A.5	W	0	Timer1 enable
	TM1SEL	0A.4	W	0	1= use TKRC as Timer1/PSC clock; 0= use Instruction Cycle as Timer1/PSC clock
	TM1PSC	0A.3~0	W	0	Timer0 Pre-Scale, 0:div1, 7:div128, 8:div256
R0E	TKE	0E.6	W	0	Touch Key Enable
	TKSPD	0E.5~4	W	0	Touch Key Speed Select
	TKSEL	0E.2~0	W	0	Touch Key Channel Select
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

–

	Name	Address	R/W	Rst	Description
R12	VDD5VIE	12.6	W	0	VDD5V Rise Interrupt enable
	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.
	EP0IND	13.4	R		SETUP Token indicator.
	OUT0CNT	13.3~0	R		OUT0 Received data byte count.
R20	PAE	20.7~0	W	0	Port A CMOS push-pull output enable
R21	PBE	21.3~0	W	0	Port B CMOS push-pull output enable
R24	PEE	24.7~0	W	0	Port E CMOS push-pull output enable
R25	PAPU	25.7~0	W	0	Port A pull-up, 0=enable
R26	PBPU	26.7~0	W	0	Port B pull-up, 0=enable
R27	PEPU	27.0	W	0	Port E pull-up, 0=enable
R30	PWMEN	30.0	W	0	PWM Enable bit
	PWMPSC	30.2~1	W	0	PWM clock prescale; 00= clk/2, 01= clk/4, 10= clk/8, 11= clk/16
R31	PWMDUTY	31.7~0	W/R	0	PWM Duty
R32	PWMPRD	32.7~0	W	0	PWM Period
R3B	CPOL	3B.5	W	0	SPI Clock Polarity
	CPHA	3B.4	W	0	SPI Clock Phase
	BSL	3B.3~0	W	7	Buffer shift bit counter
R3C	SPICRS	3C.6~0	W	0	SPI clock Select
R3D	SPILEN	3D.6~0	W	0	SPI DMA transfer length; Length: 1 ~64 bytes
R3E	SPITX	3E.7~0	W	0	SPI Transmit DATA in CMD phase
R3F	SPIRX	3F.7~0	R	-	SPI Received Data
	SET0FIFO	40~47	R	-	Endpoint 0 SETUP Receive Buffer (8 Bytes)
	OUT0FIFO	48~4F	R	-	Endpoint 0 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)
	XRAM1	80~BF	R/W	-	Endpoint 3/4 Buffer (64 Bytes)
	XRAM2	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:

- Bit stuffing/unstuffing
- CRC generation/checking
- ACK/NAK
- TOKEN type identification
- Address checking

F/W handles the following tasks:

- Coordinate enumeration by responding to SETUP packets
- Fill and empty the FIFOs
- Suspend/Resume coordination
- 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 USB Endpoint 0 Receive (SET0/OUT0)

After receiving a SETUP packet and placing the data into the Endpoint 0 setup receive FIFO (SET0FIFO), TMU3131 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 keep 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 USB Endpoint 0 Transmit (TX0)

After detecting a valid Endpoint 0 IN token, TMU3131 automatically transmit 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 TMU3131 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 USB 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, TMU3131 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 TMU3131 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 USB 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, TMU3131 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 TMU3131 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 to check error, will response 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 USB 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 TMU3131 sends Resume signaling to USB bus when SUSP=1 and RSMO=1.

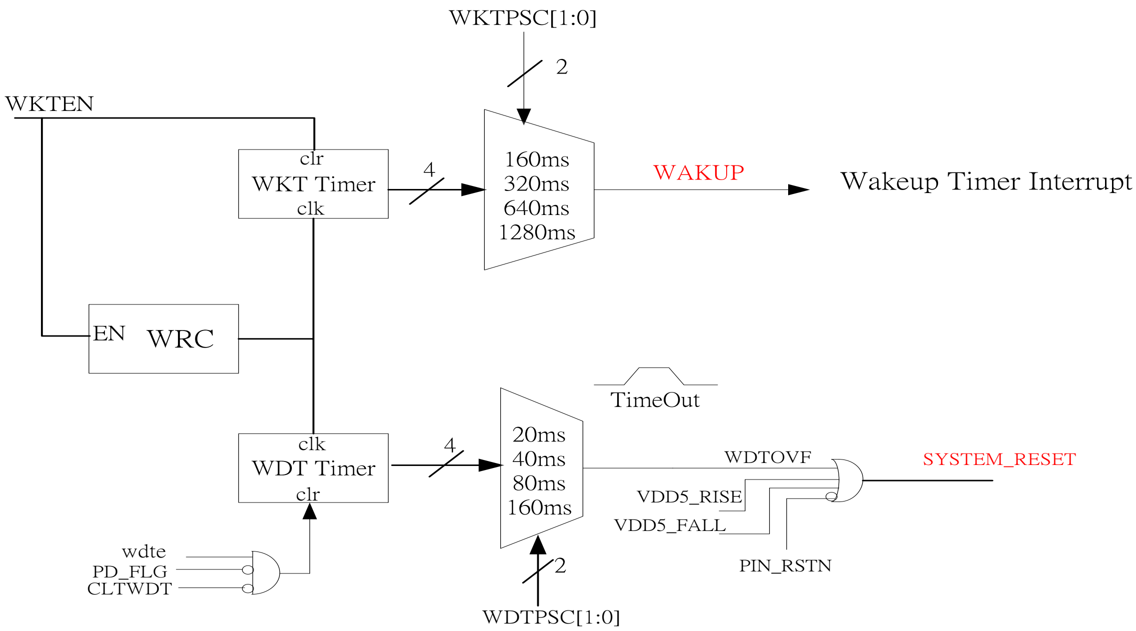
3.9 Interrupt Vector

There are several interrupts generated by USB Engine. The other interrupts including timer0/1 interrupts, wakeup timer interrupt, PB0 external I/O interrupt, keyboard interrupt and VDD5V rise interrupt. Each interrupt sources 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 Interrupt
0e	Reserved
0f	VDD5V Rise Interrupt
10	External IO Interrupt
11	Timer1 Interrupt

4. Wakeup Timer and Watch Dog Timer

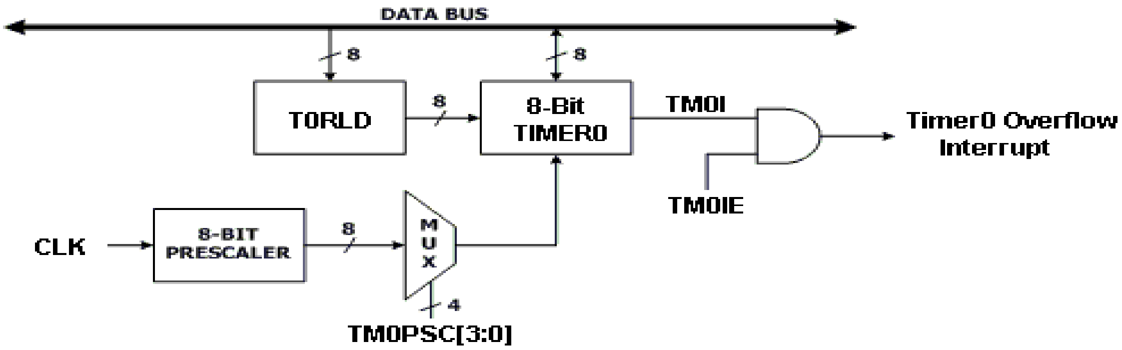
The WKT and WDT use the same internal RC (WRC). This internal RC (WRC) can be disabled by setting R06[7] “High” for power saving. The overflow period of WDT can be selected from 20 ms to 80 ms and the wakeup period of WKT can be selected from 160 ms to 1280 ms. The WDT is enabled and cleared by the CLRWDT 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 setting “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.

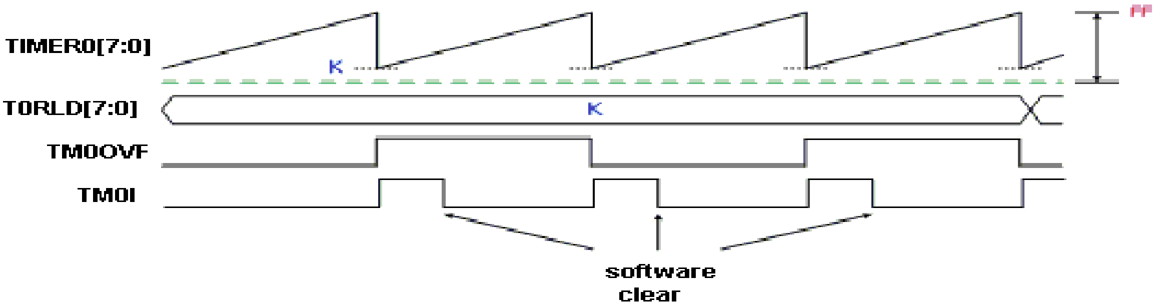


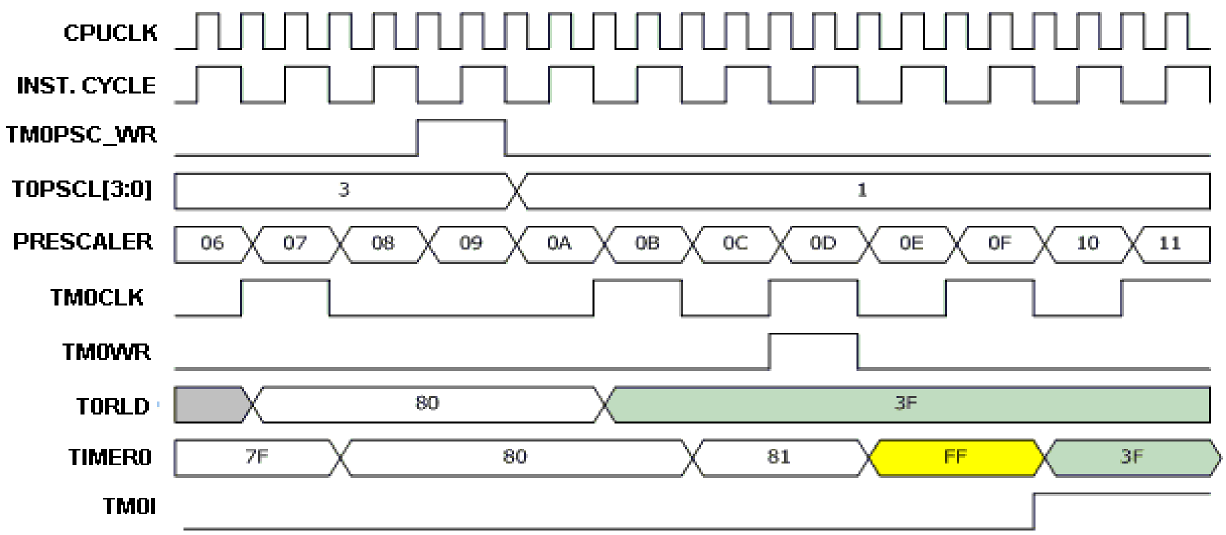
5. Timer

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" (T0PSCL) register in R-Plane. The Timer0 can generate interrupt (TM0I).

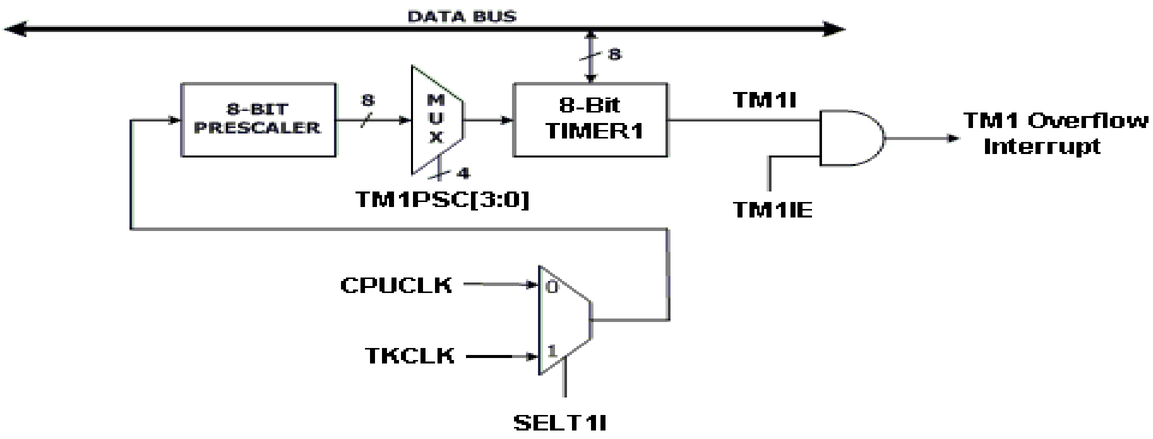




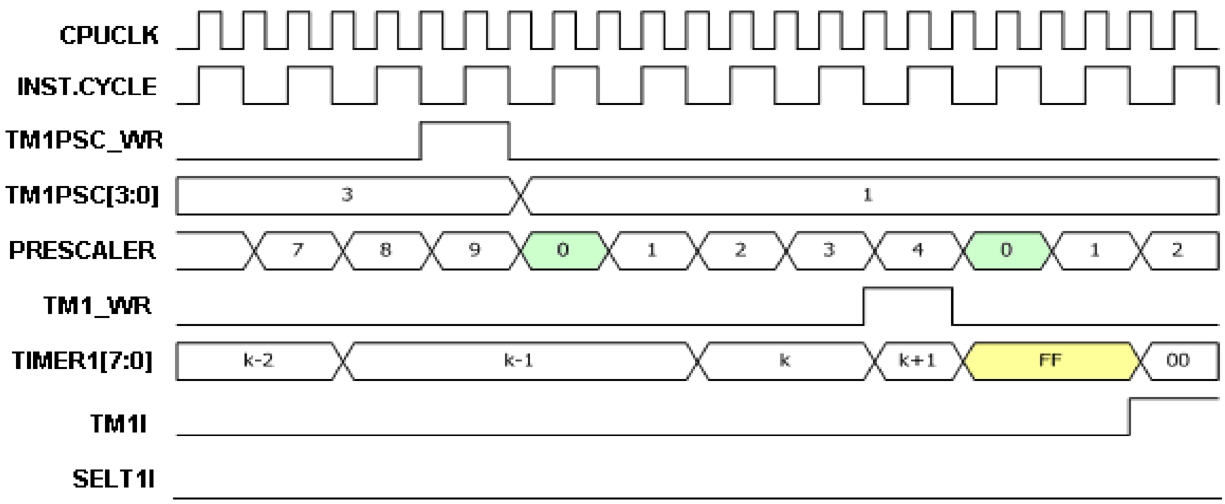


5.2 Timer1: 8-bit Timer/Counter with Pre-scale (PSC)

The Timer1 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer1 increases itself periodically and automatically rolls over based on the pre-scaled clock source, which can be the instruction cycle or touch key induced clock (TKCLK). The Timer1 increase rate is determined by "Timer1 Pre-Scale" (TM1PSC) register in R-Plane. The Timer1 can generate interrupt (TM1I) when it rolls over.

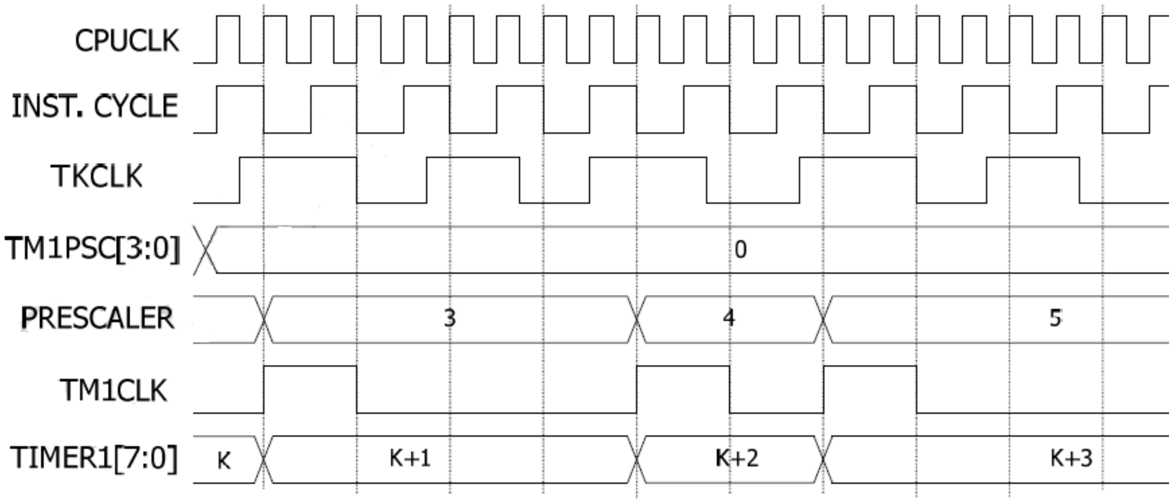


When Timer1 works in pure timer mode, the Timer1 prescaler (TM1PSC) is written, the internal 8-bit prescaler will be cleared to 0 to make the counting period correct at the first Timer1 count. TM1WR is the internal signal that indicates the Timer1 is directly written by instruction; meanwhile, the internal 8-bit prescaler will be cleared. When Timer1 counts from FFh to 00h, TM1I (Timer1 Interrupt Flag) will be set to 1 and generate interrupt if TM1IE (Timer1 Interrupt Enable) is set.



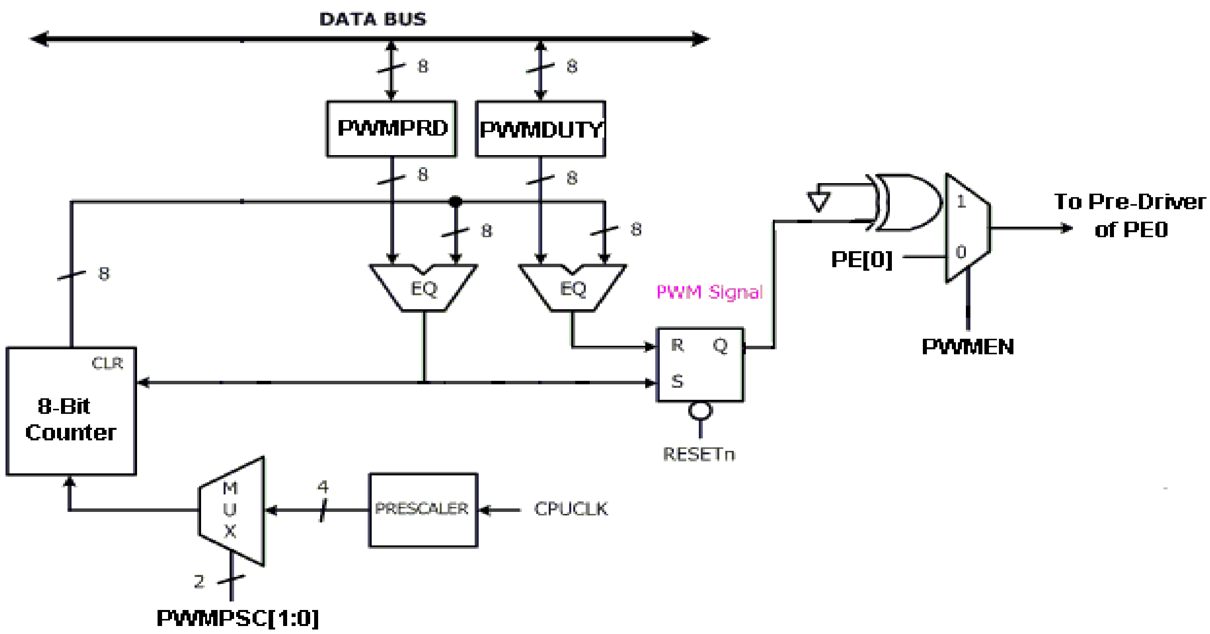
–

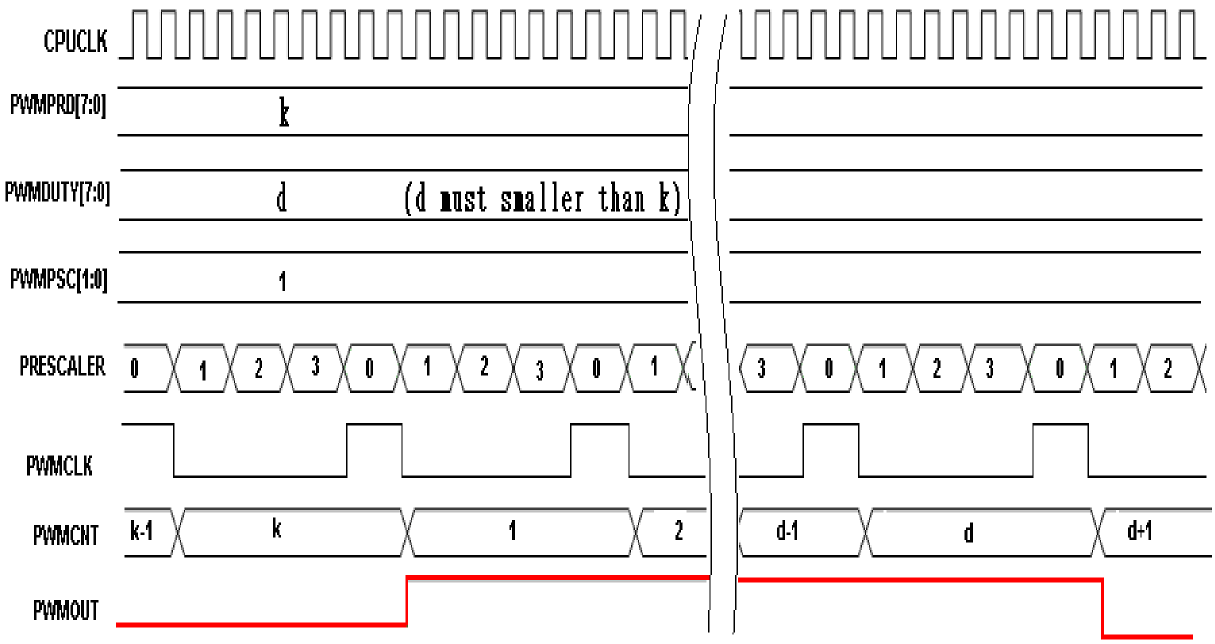
The following timing diagram describes the Timer1 works in counter mode. If SELT1I=1 then the Timer1 counter source clock is from Touch Key module that depends on TKE bit. In this mode the counter is used for Touch Key function.



6. 8-bit PWM

The PWM will be enabled by setting R30[0] (PWMEN) to “1”. Once the PWMEN is set, the PWM 8-bit counter starts to count and the PWM will be output to PE0. The PWM increase rate is determined by R30[2:1] (PWMPSC). The PWM output signal toggles to low level whenever the 8-bit counter matches register R31 (PWMDUTY) and toggles to high level whenever the 8-bit counter matches register R32 (PWMPRD). The PWM duty cycle can be changed with writing to PWMDUTY, writing to PWMDUTY will not change the current PWM duty until the current PWM period complete. When finishing current PWM period, the new value of PWMDUTY will be updated.





7. SPI (Serial Peripheral Interface)

This SPI module can be used as master only. The clock rate and data transfer length are also adjustable. SPI clock rate = CPUCLK/2*(CRS+1)

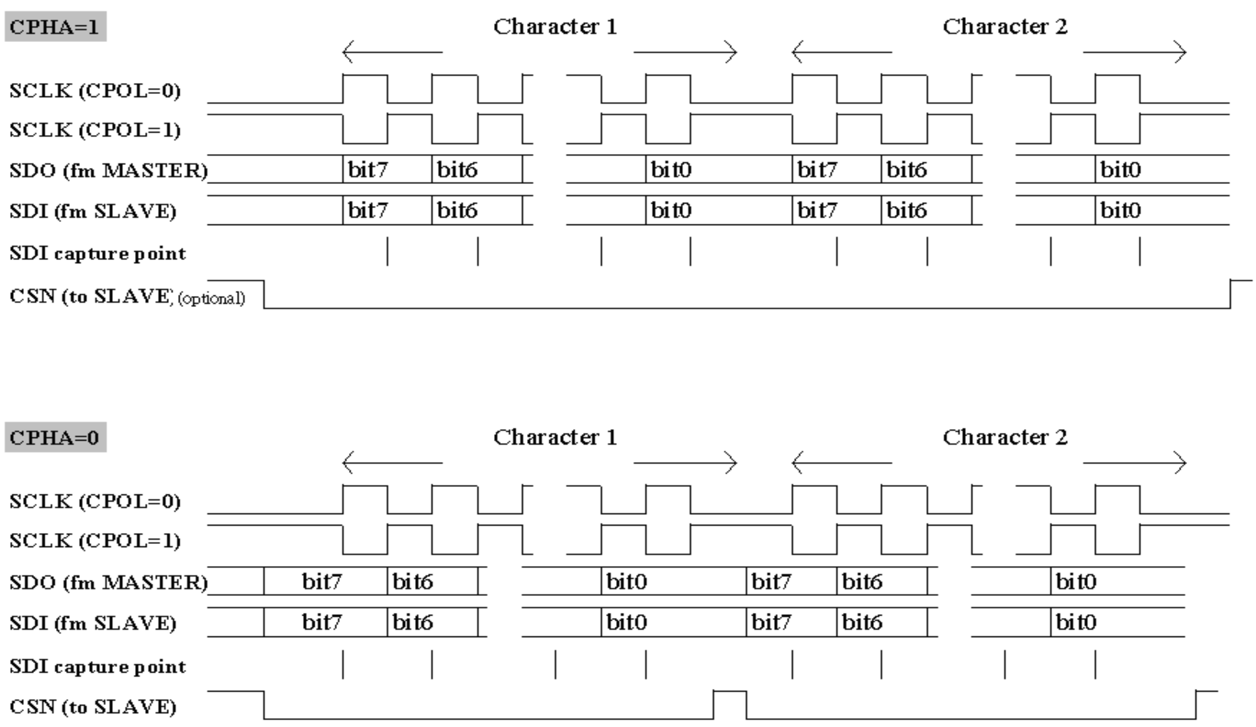
CRS[6:0]	SPI Clock Rate
0	6 Mbps
3	1.5 Mbps
15	375 kbps

Note: CPUCLK = 12 MHz

All the registers must be set before F1D.4 (SPI_EN) bit is set.

There are two data transfer modes. One is command phase mode, in this mode the data transfer length is “1” and the data must 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 are stored in the SRAM (RAM1 or RAM2).

SPI Timing



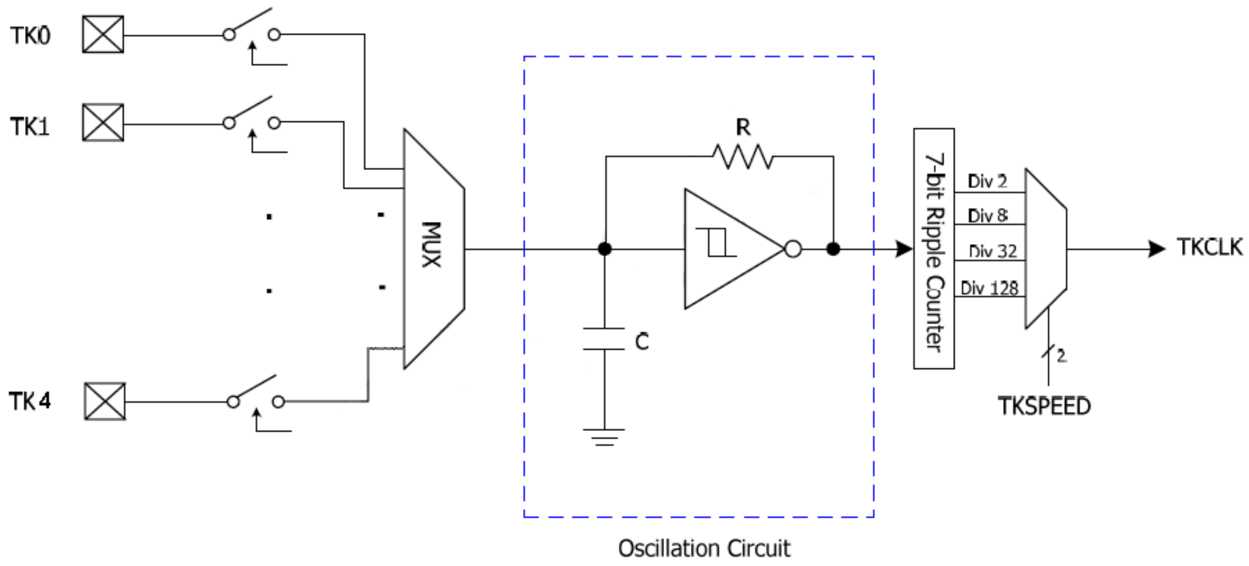
8. Touch Key

As mentioned in Timer1, the Touch Key Module outputs the oscillation clock to Timer1 and counts like T1I input. The block diagram of the Touch Key module is shown below. It consists of a RC oscillator, 16-to-1 analog input select, TKSPEED control bits select the output of the frequency divider. The frequency divider divides the oscillation clock by 2, 8, 32, and 128. If TKE bit is 1, the divided clock will be sent to Timer1 to count at the rising edge.

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

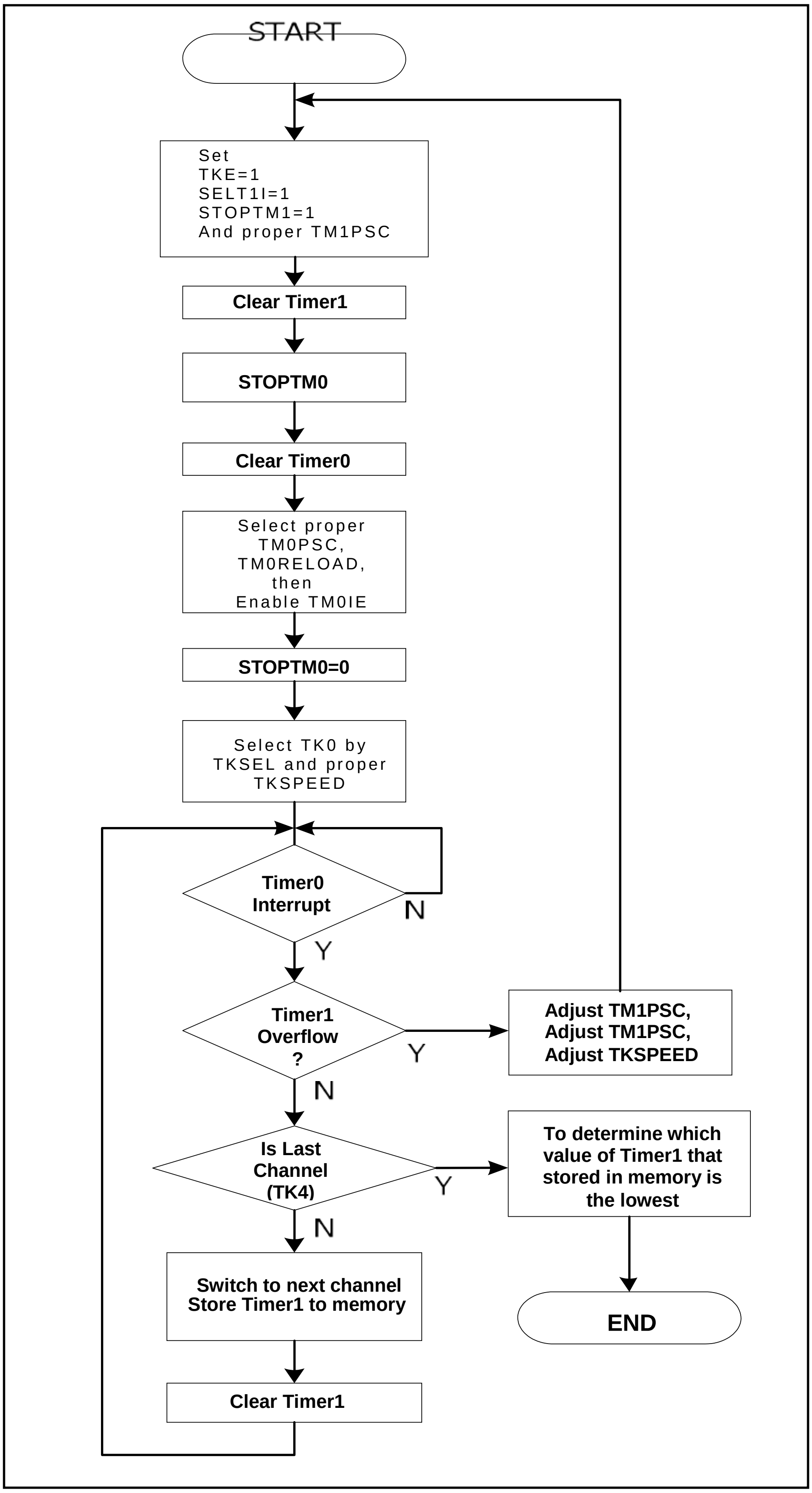
Based on the above thesis, user program needs to observe what input channel causes the lowest Timer1 counting value in a fixed period of time, which channel of key is touched or the finger is just approaching.

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



The flowchart described below shows how to use Timer0 and Timer1 to determine what channel of the Touch Key is pressed. Using the 8-bit Timer0 to set up a fix interval of time and utilize the Time0 interrupt to stop Timer1 and store its value if it is not overflow. Determine the lowest value of Timer1 of the desired channels that is the key pressed.

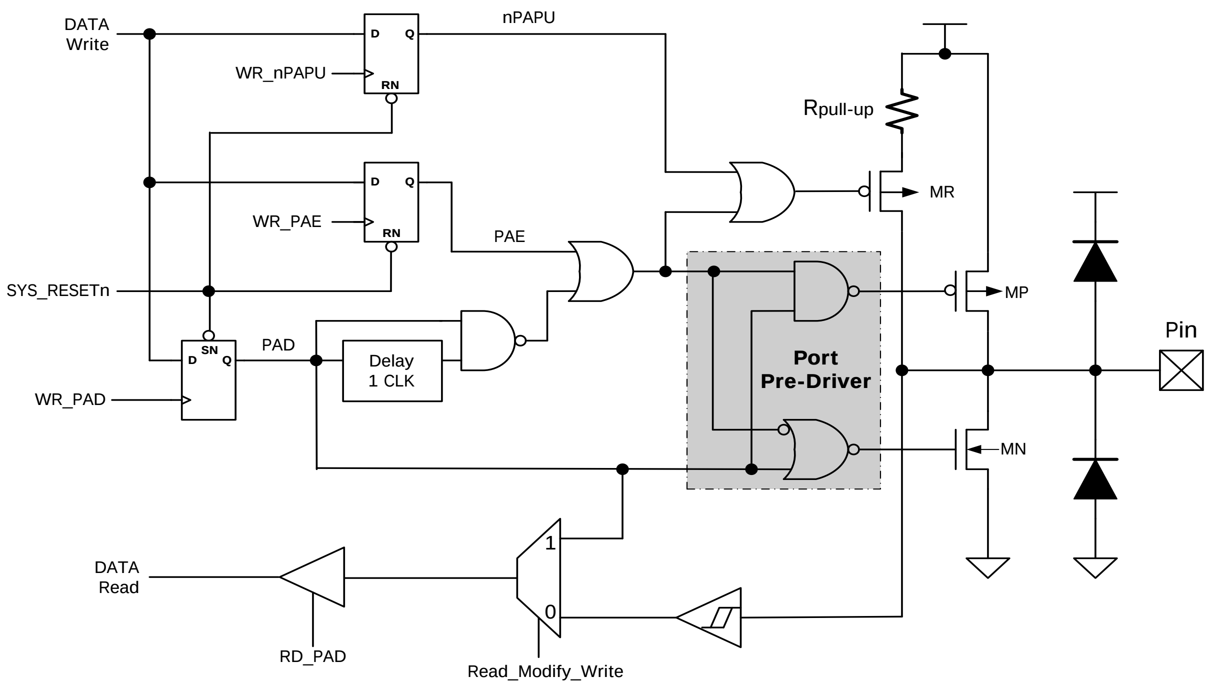
–


START
Set
TKE=1
SELT1I=1
STOPTM1=1
And proper TM1PSC
Clear Timer1
STOPTM0
Clear Timer0
Select proper
TM0PSC,
TM0RELOAD,
then
Enable TM0IE
STOPTM0=0
Select TK0 by
TKSEL and proper
TKSPEED
Timer0
Interrupt
N
Y
Timer1
Overflow
?
Y
Adjust TM1PSC,
Adjust TM1PSC,
Adjust TKSPEED
N
Is Last
Channel
(TK4)
Y
To determine which
value of Timer1 that
stored in memory is
the lowest
N
Switch to next channel
Store Timer1 to memory
END
Clear Timer1

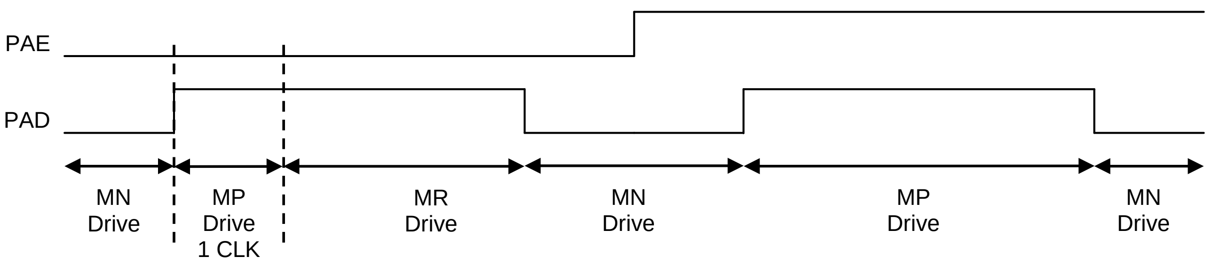
9. I/O Port

9.1 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.

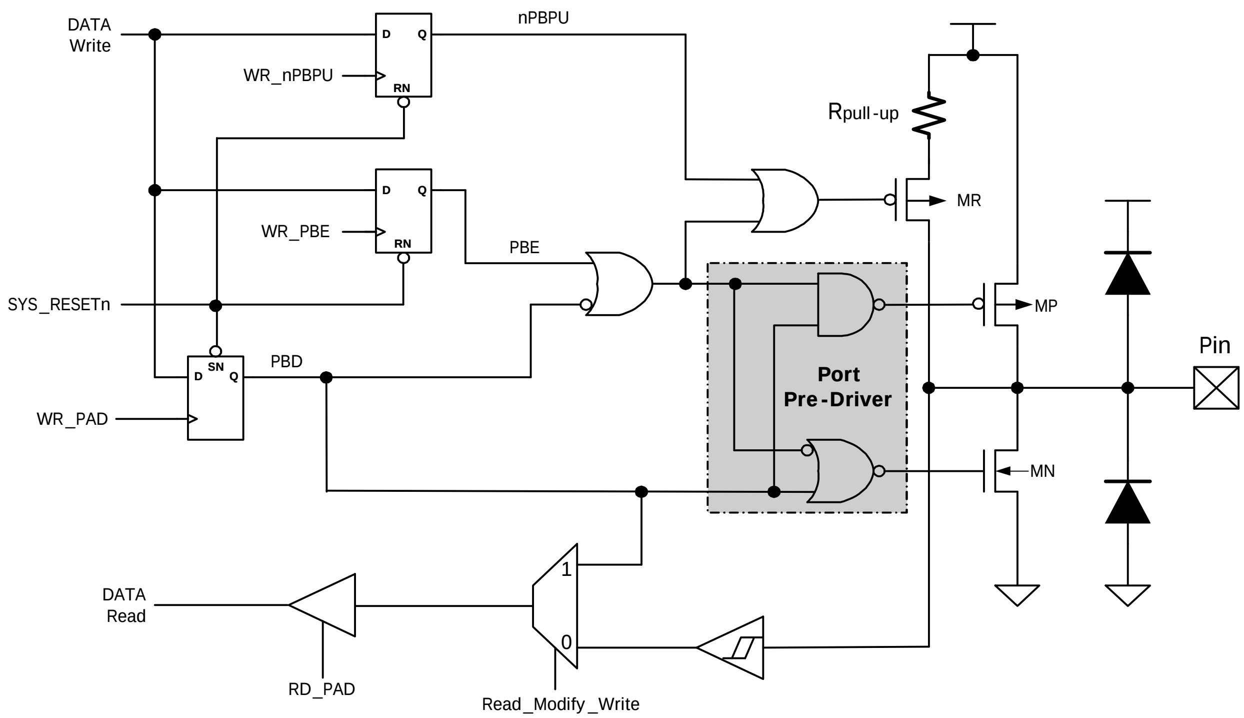


PA0-7, nPAPU=0

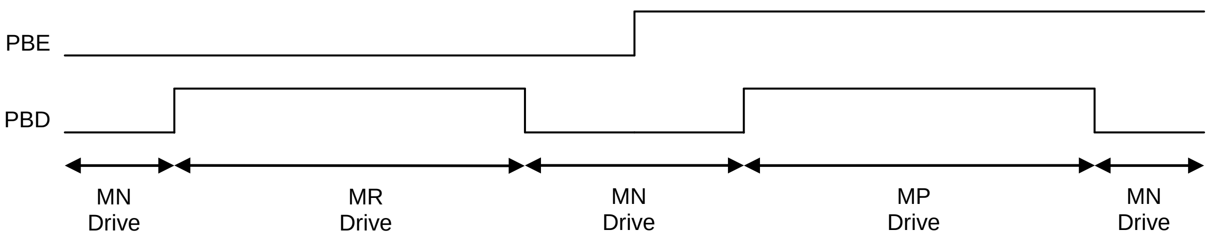


9.2 PB0-1

These two pins are almost same as PA0-7, except they do not support pseudo-open-drain mode. They can be used in pure open-drain mode, instead.

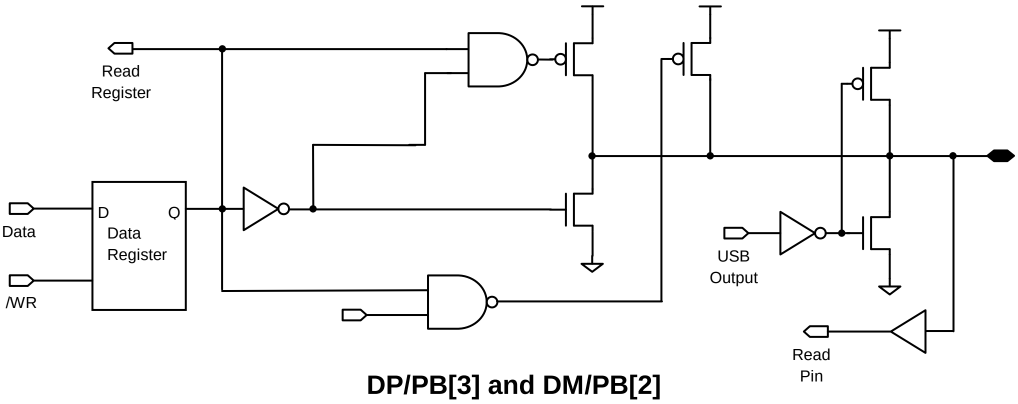


PB0-1, nPBPU=0



9.3 PB3(DP) and PB2(DM)

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

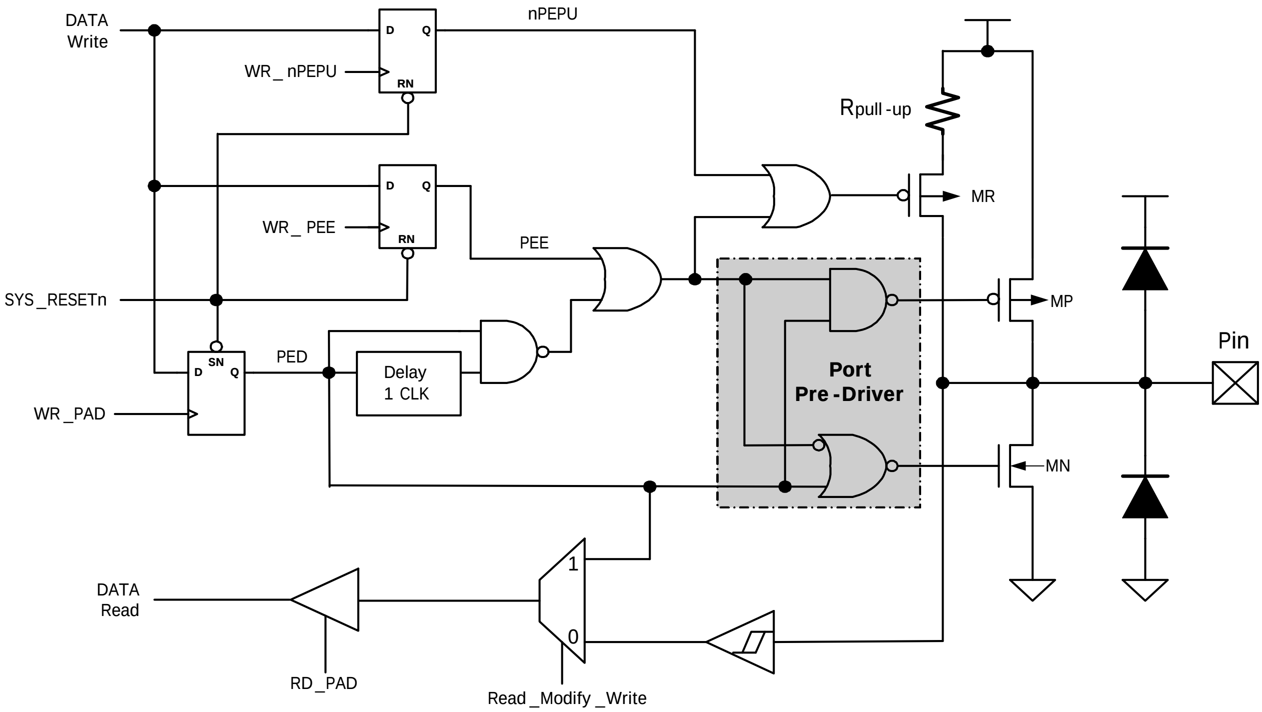


DP/PB[3] and DM/PB[2]

9.4 PE0-6

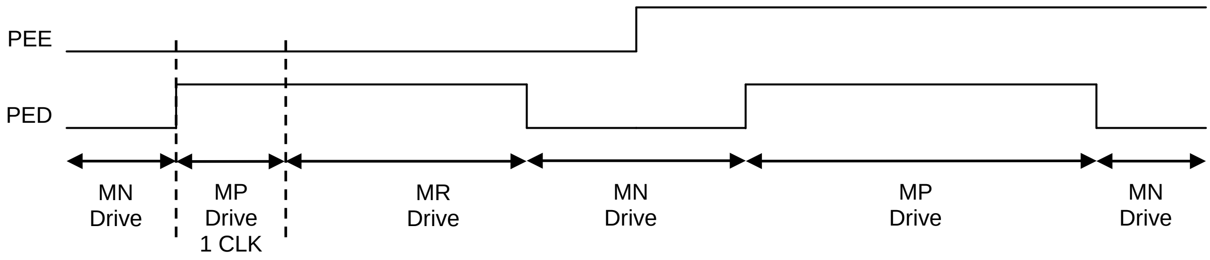
PortE pins are same as Port A, except the pull-up enable bit. There are 8 different pull-up enable bits nPAPU[7:0] to control PortA. Only one pull-up enable bit nPEPU is used to control Port E. PE[0] can output PWM by setting Register R30, R31 and R32. PE3 can be configured as clock output by setting Register R09[1]. R09[0] is used to determine the output clock rate.

R09[1:0] = 2'b0x	PE3 is 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

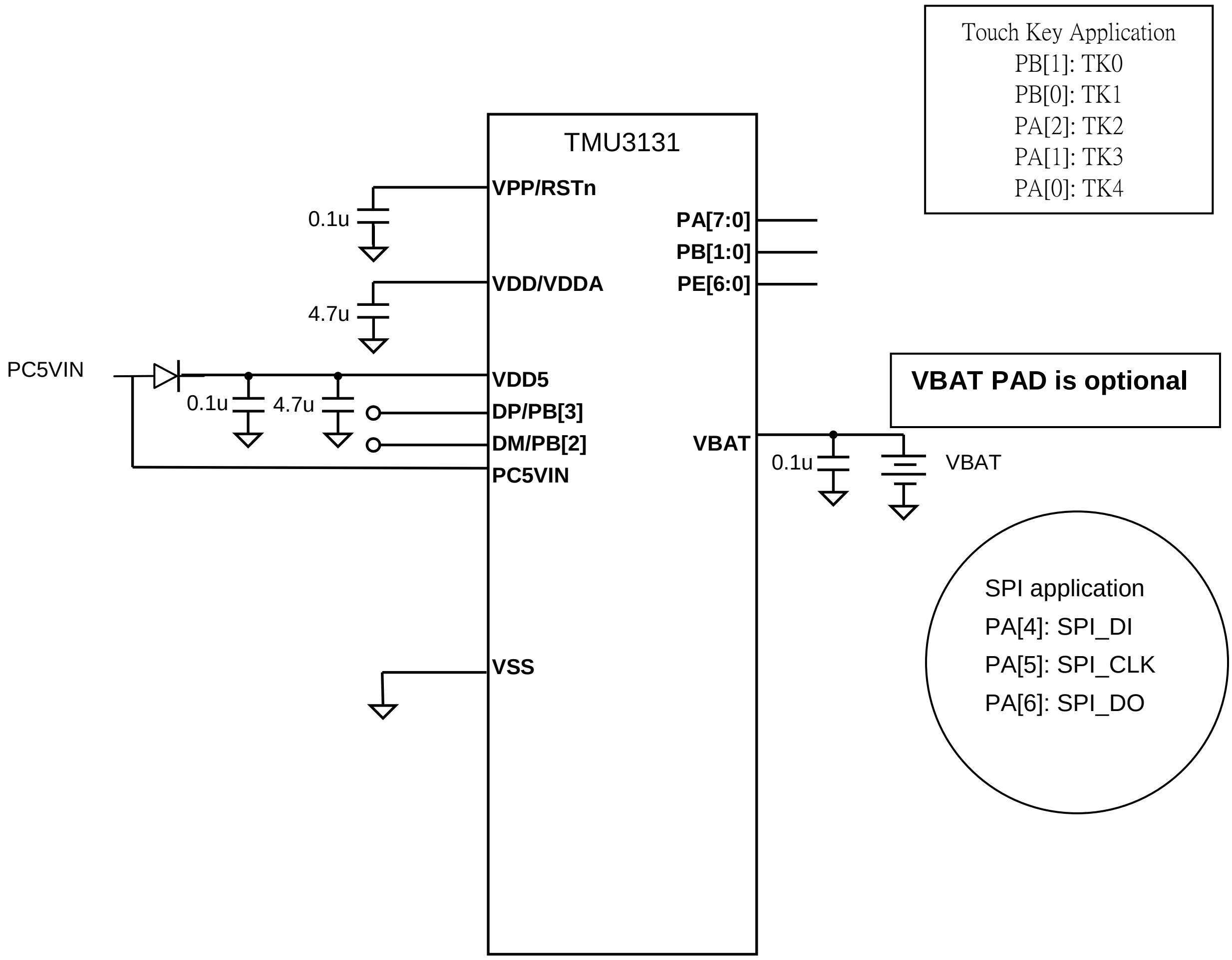


_

PE0-6, nPEPU=0


PEE
PED
MN
Drive
MP
Drive
1 CLK
MR
Drive
MN
Drive
MP
Drive
MN
Drive

10. Application



11. 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	℃
Maximum Storage Temperature	Tstg	-65 to +150	℃

RECOMMENDED OPERATING CONDITION

At Ta=-20℃ to 70℃,GND= 0V

Name	Symb.	Min.	Typical	Max.	Unit	Condition
Supply Voltage	VDD5	2.3		5.5	V	
Battery Voltage, if apply	Vbat	2.2		3.6	V	
VDD output voltage	VDD		3.3		V	VDD5=5V Vbat=0V
				2.96	V	VDD5=3V Vbat=0V
				3.2	V	Vbat=3.6V, VDD5=0V
				2.93	V	Vbat=3V, VDD5=0V
Input "H" Voltage	Vih	0.8VDD			V	
Input "L" Voltage	Vil1			0.3VDD	V	

DC CHARACTERISTICS

At Ta=-25 ℃, VDD5=5.0V, VSS= 0V, Fcpu=12 MHz

Name	Symbol	Min.	Typ.	Max.	Unit	Condition
Internal Clock	F_{48m}		48		MHz	Enable IRC, VDD5=5V
Operating current	Icc		5		mA	CPU clock=12 MHz
Suspend current	Isus		360	500	uA	USB Mode, No load
Power Down current	Ipd			1	uA	No load
Output High Current (Push Pull Mode)	Ioh1		12		mA	VDD5=5V, Voh1=2.8V
	Ioh2		14		mA	VDD5=3V, Voh2=2.3V
Output High Current (Pseudo Open Drain Mode)	Ioh3		12		uA	VDD5=5V, Voh3=2.8V
	Ioh4		13		uA	VDD5=3V, Voh4=2.3V
Output Low Current (Push Pull Mode)	Iol1		22		mA	VDD5=5V, Vol1=0.3V
	Iol2		21		mA	VDD5=3V, Vol2=0.3V
Output Low Current (Pseudo Open Drain Mode)	Iol3		22		mA	VDD5=5V, Vol3=0.3V
	Iol4		21		mA	VDD5=3V, Vol4=0.3V

–

Name	Symbol	Min.	Typ.	Max.	Unit	Condition
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		2.1		V	Fcpu=1.5MHz
WDT time	Twdt		14		ms	VDD5=5V, WRC enable R06[6:5]=2'b00
			28		ms	VDD5=5V, WRC enable R06[6:5]=2'b01
			56		ms	VDD5=5V, WRC enable R06[6:5]=2'b10
			112		ms	VDD5=5V, WRC enable R06[6:5]=2'b11
WKT Time	Twkt		112		ms	VDD5=5V, WRC enable R06[4:3]=2'b00
			224		ms	VDD5=5V, WRC enable R06[4:3]=2'b01
			448		ms	VDD5=5V, WRC enable R06[4:3]=2'b10
			896		ms	VDD5=5V, WRC enable R06[4:3]=2'b11

AC CHARACTERISTICS

At Ta=25 ℃, VDD5V=5.0V, VSS= 0V, Fcpu=12 MHz

Name	Symbol	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.

USB AC CHARACTERISTICS

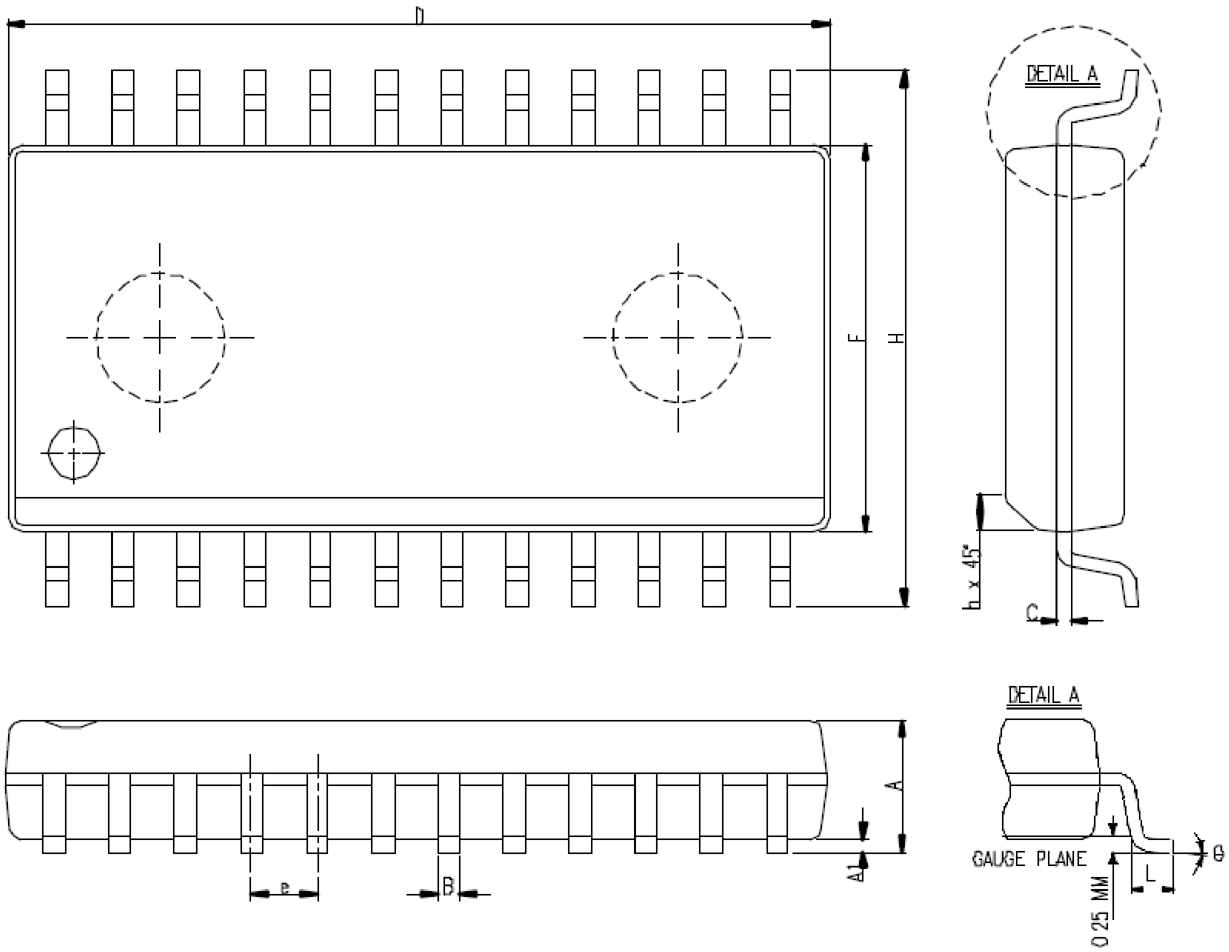
At Ta=25 ℃, VDD5V=5.0V, VSS= 0V

Name	Symbol	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	
VDD output voltage	VDD	3.2	3.3	3.4	V	

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

12. Package Information

- **SOP-24/SOP-28**

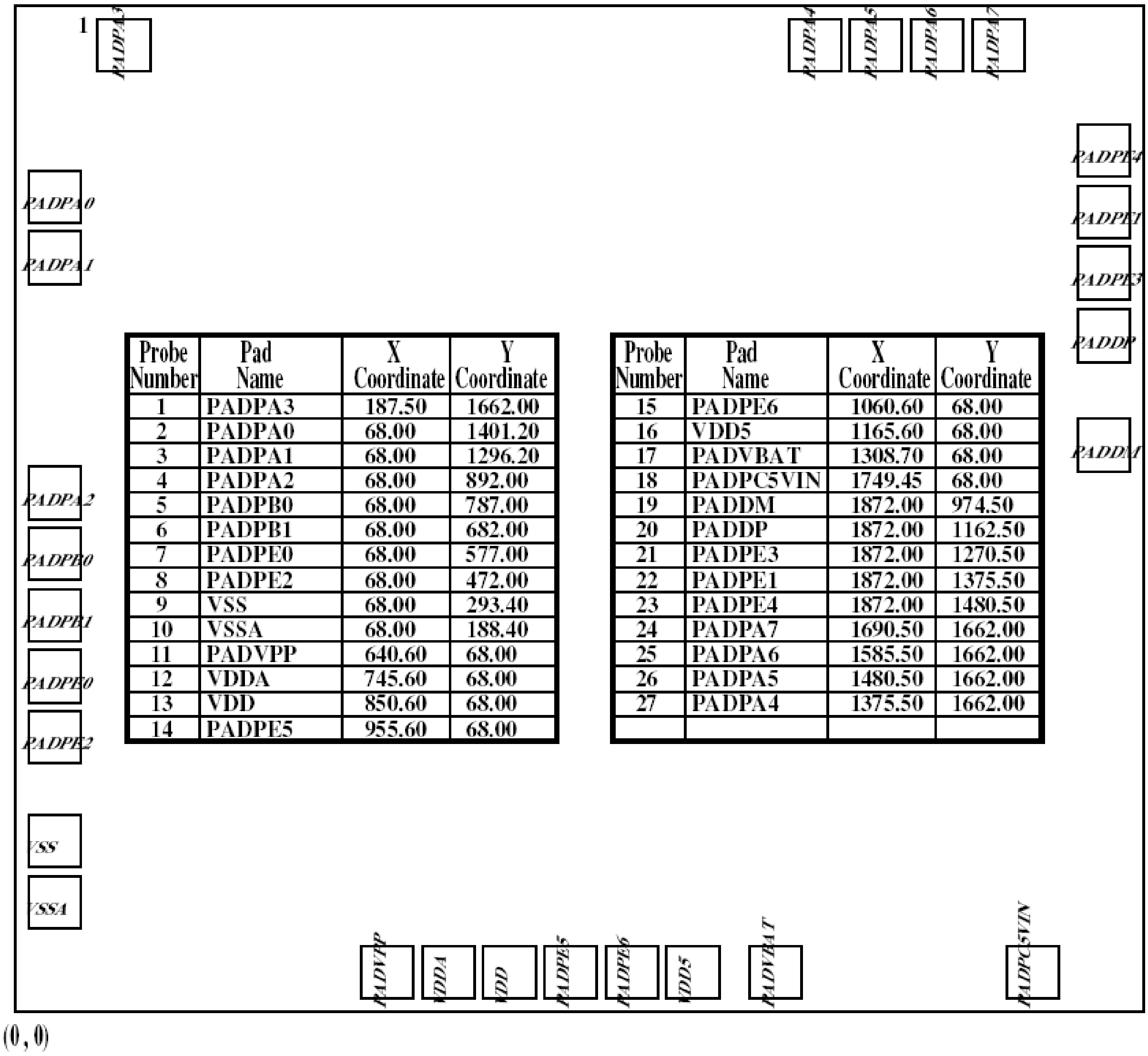


	D DIMENSION (IN MM)		D DIMENSION (IN INCH)		
N	MIN.	MAX.	MIN.	MAX.	JEDEC
20	12.60	13.00	0.4961	0.5118	MS–013 (AC)
24	15.20	15.60	0.5985	0.6141	MS–013 (AD)
28	17.70	18.10	0.6969	0.7125	MS–013 (AE)

SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN.	MAX.	MIN.	MAX.
A	2.35	2.65	0.0926	0.1043
A1	0.10	0.30	0.0040	0.0118
B	0.33	0.51	0.013	0.020
C	0.23	0.32	0.0091	0.0125
e	1.27 BSC		0.050 BSC	
E	7.40	7.60	0.2914	0.2992
H	10.00	10.65	0.394	0.419
L	0.40	1.27	0.016	0.050
h	0.25	0.75	0.010	0.029
θ	0°	8°	0°	8°

13. Dice Information

Pads Diagram



Probe Number	Pad Name	X Coordinate	Y Coordinate
1	PADPA3	187.50	1662.00
2	PADPA0	68.00	1401.20
3	PADPA1	68.00	1296.20
4	PADPA2	68.00	892.00
5	PADPB0	68.00	787.00
6	PADPB1	68.00	682.00
7	PADPE0	68.00	577.00
8	PADPE2	68.00	472.00
9	VSS	68.00	293.40
10	VSSA	68.00	188.40
11	PADVPP	640.60	68.00
12	VDDA	745.60	68.00
13	VDD	850.60	68.00
14	PADPE5	955.60	68.00

Probe Number	Pad Name	X Coordinate	Y Coordinate
15	PADPE6	1060.60	68.00
16	VDD5	1165.60	68.00
17	PADVBAT	1308.70	68.00
18	PADPC5VIN	1749.45	68.00
19	PADDM	1872.00	974.50
20	PADDP	1872.00	1162.50
21	PADPE3	1872.00	1270.50
22	PADPE1	1872.00	1375.50
23	PADPE4	1872.00	1480.50
24	PADPA7	1690.50	1662.00
25	PADPA6	1585.50	1662.00
26	PADPA5	1480.50	1662.00
27	PADPA4	1375.50	1662.00

Ordering Information

The ordering information:

Ordering number	Package
TMU3131-MTP	Wafer / Dice blank chip
TMU3131-COD TMU3131ES-COD	Wafer / Dice with code
TMU3131-MTP-22-X	SOP 24-pin (300 mil)
TMU3131-MTP-23-X	SOP 28-pin (300 mil)
TMU3131S-MTP-22-X	SOP 24-pin (300 mil)

Note: "-X" represents the package material:

- Package material: Pb-free Code: W
- Package material: Green Package Code: G