十速

TMU32FA80

DATA SHEET

Rev DV1.0

AMENDMENT HISTORY

Version	Date	Description
DV1.0	May, 2014	New release

CONTENTS

SPEC Overview

Microcontroller Features Table

CPU	ROM	RAM Bytes	OSC Source	Instruction		Stack level	Interrupt vector	Timers	LVR
				Set	Time				
RISC	Flash 8k*14	160 + 192	FIRC 48 MHz	37	2T	8	18	T0/T1/T2	2.0V

FIRC 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		Bulk mode	Interrupt mode	Touch Key	PWM Output	SPI Interface
EP0 (SETUP) 8-Byte	EP0 (IN/OUT) Up to 64-Byte	EP1 & EP2 each up to 64-Byte	EP3/EP4/EP5 each up to 8-Byte	Capacitive	8-Bit	Master only
				5-ch	CPUCLK	DMA R/W

Features

RISC CPU :

- ◆ **Only 37 instructions**
- ◆ **Instruction Execution Time**
  2-cycle instructions except branch
- ◆ **Operating clock**
  - Fast Clock:
    - FIRC (Fast Internal RC 48 MHz +/-3%)
    - SIRC (Slow Internal RC 111 KHz)
  - CPU Clock control (When FIRC is used):
    - 12 MHz / 6 MHz / 3 MHz / 1.5 MHz
  - Clock Output:
    - 6 MHz / 12 MHz Output
- ◆ **8Kx14 internal flash Program Memory**
- ◆ **Memory**
  - 160 bytes on F-plane
  - 192 bytes 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 Bulk Transmit Interrupt
    - USB EP2 Bulk Receive Interrupt
    - USB Suspend Interrupt
    - USB EP3 Transmit Interrupt
    - USB EP4 Receive Interrupt
    - USB Bus Reset Interrupt
    - USB Resume Interrupt
    - USB EP5 Transmit Interrupt
    - Wake-up Timer Interrupt
    - Timer0 Interrupt
    - PB0 External I/O Interrupt
    - ADC End of Conversion Interrupt
    - VDD5V Rise interrupt
    - Timer1 Interrupt
    - Timer2 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, 13.9 mA typical
    - 5V@ 6 MHz, 6.8 mA typical
    - 5V@ 3 MHz, 4.8 mA typical
    - 5V@ 1.5 MHz, 3.9 mA typical
  - Suspend mode
    - 5V@ 12 MHz, 350 uA typical
  - Power down mode
    - 5V@ 1 uA typical
- ◆ **H/W Reset**
  - External active low reset (RSTN)
  - Build-in Power-On Reset (POR)
  - Low Voltage Reset - 2.1V (LVR)
  - Watchdog Reset (WDT)

Peripheral Features

- ◆ **I/O Port :**
  - Maximum 38 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 8-bit counter/Interrupt function
  - Timer2 is 8-bit counter/Interrupt function
- ◆ **PWMs**
  - Built-in 5 8-bit PWM generator
  - PWM0/1 with prescaler / period adjustment / buffer-reload / rising-falling output
  - PWM2/3/4 with same period but different duty adjustment 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-24**

Pin number	Pin Name	Type	Input		Output			Func. after reset	Alternate Function			Misc
SOP24			Enable Pull-up	Ext. Interrupt	O.D.	P.O.D.	P.P.		ADC	Touch-Key	SPI	
1	PC5VIN	I						PC5VIN				
2	VSS	P						VSS				
3	VDD	P						VDD				
4	DP / SCK / PB3	I/O	●		●		●	DP				USB
5	DM / DAT / PB2	I/O	●		●		●	DM				USB
6	PWM1 / PE1	I/O	●			●	●	PE1				PWM Output
7	PWM0 / PE0	I/O	●			●	●	PE0				
8	PWM2 / PE4	I/O	●			●	●	PE4				
9	VPP / RSTN	I						RSTN				Reset
10	SPI_DI/ADC20 / PA4	I/O	●			●	●	PA4	●			
11	SPI_CK/ADC21/ PA5	I/O	●			●	●	PA5	●			
12	SPI_DO/ADC22 / PA6	I/O	●			●	●	PA6	●			
13	ADC23 / PA7	I/O	●			●	●	PA7	●			
14	ADC8 / PD0	I/O	●			●	●	PD0	●			
15	ADC9 / PD1	I/O	●			●	●	PD1	●			
16	ADC10 / PD2	I/O	●			●	●	PD2	●			
17	ADC11 / PD3	I/O	●			●	●	PD3	●			
18	ADC12 / PD4	I/O	●			●	●	PD4	●			
19	ADC13 / PD5	I/O	●			●	●	PD5	●			
20	ADC14 / PD6	I/O	●			●	●	PD6	●			
21	TK2 / PB4	I/O	●		●		●	PB4		●		
22	TK0 / PB0	I/O	●	●	●		●	PB0		●		
23	VDD5	P						VDD5				
24	VBAT	P						VBAT				

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

PS:

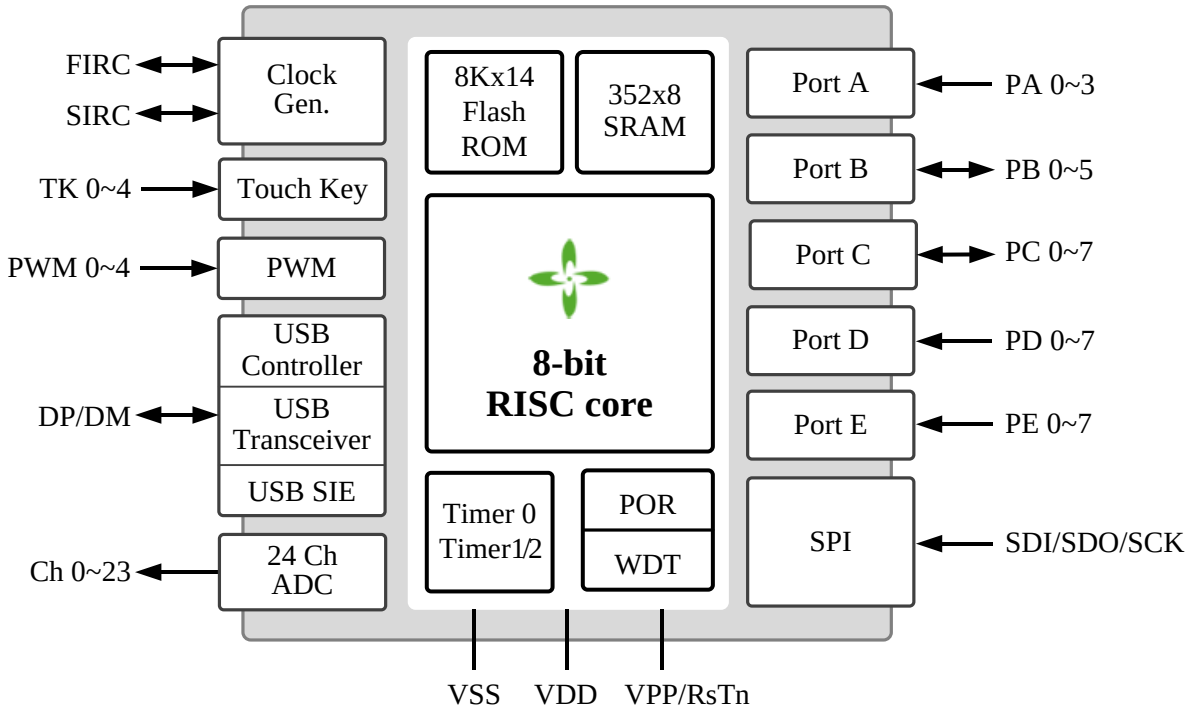
1. PE[3] can be configured as clock output.
2. PE[0,1,4,5,6] can output PWM by setting Register.
3. PB[5:0] and PC[7:0] support Open Drain, and the other use of Pseudo Open Drain

- **LQFP-44**

Pin number	Pin Name	Type	Input		Output			Func. after reset	Alternate Function			Misc
LQFP44			Enable Pull-up	Ext. Interrupt	O.D.	P.O.D.	P.P.		ADC	Touch-Key	SPI	
1	CLKO / PE3	I/O	●			●	●	PE3				Clock Output
2	PWM1 / PE1	I/O	●			●	●	PE1				PWM Output
3	PWM0 / PE0	I/O	●			●	●	PE0				
4	PWM2 / PE4	I/O	●			●	●	PE4				
5	PWM3 / PE5	I/O	●			●	●	PE5				
6	PWM4 / PE6	I/O	●			●	●	PE6				
7	VPP / RSTN	I						RSTN				Reset
8	ADC16 / PA0	I/O	●			●	●	PA0	●			
9	ADC17 / PA1	I/O	●			●	●	PA1	●			
10	ADC18 / PA2	I/O	●			●	●	PA2	●			
11	ADC19 / PA3	I/O	●			●	●	PA3	●			
12	SPI_DI/ADC20 / PA4	I/O	●			●	●	PA4	●			
13	SPI_CK/ADC21/ PA5	I/O	●			●	●	PA5	●			
14	SPI_DO/ADC22 / PA6	I/O	●			●	●	PA6	●			
15	ADC23 / PA7	I/O	●			●	●	PA7	●			
16	ADC2 / PC0	I/O			●	●	●	PC0	●			
17	ADC1 / PC1	I/O			●	●	●	PC1	●			
18	ADC2 / PC2	I/O			●	●	●	PC2	●			
19	ADC3 / PC3	I/O			●	●	●	PC3	●			
20	ADC4 / PC4	I/O			●	●	●	PC4	●			
21	ADC5 / PC5	I/O			●	●	●	PC5	●			
22	ADC6 / PC6	I/O			●	●	●	PC6	●			
23	ADC7 / PC7	I/O			●	●	●	PC7	●			
24	ADC8 / PD0	I/O	●			●	●	PD0	●			
25	ADC9 / PD1	I/O	●			●	●	PD1	●			
26	ADC10 / PD2	I/O	●			●	●	PD2	●			
27	ADC11 / PD3	I/O	●			●	●	PD3	●			
28	ADC12 / PD4	I/O	●			●	●	PD4	●			
29	ADC13 / PD5	I/O	●			●	●	PD5	●			
30	ADC14 / PD6	I/O	●			●	●	PD6	●			
31	ADC15 / PD7	I/O	●			●	●	PD7	●			
32	TK4 / PE7	I/O	●			●	●	PE7		●		
33	TK3 / PB5	I/O	●		●		●	PB5		●		
34	TK2 / PB4	I/O	●		●		●	PB4		●		
35	TK1 / PB1	I/O	●		●		●	PB1		●		

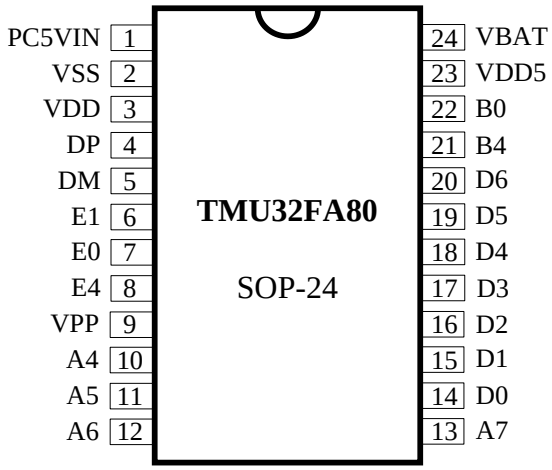
Pin number	Pin Name	Type	Input		Output			Func. after reset	Alternate Function			Misc
LQFP44			Enable Pull-up	Ext. Interrupt	O.D.	P.O.D.	P.P.		ADC	Touch-Key	SPI	
36	TK0 / PB0	I/O	●	●	●		●	PB0		●		
37	VDD5	P						VDD5				
38	VBAT	P						VBAT				
39	PC5VIN	I						PC5VIN				
40	VSS	P						VSS				
41	VDD	P						VDD				
42	DP / SCK / PB3	I/O	●		●		●	DP				USB
43	DM / DAT / PB2	I/O	●		●		●	DM				USB
44	PE2	I/O	●			●	●	PE2				

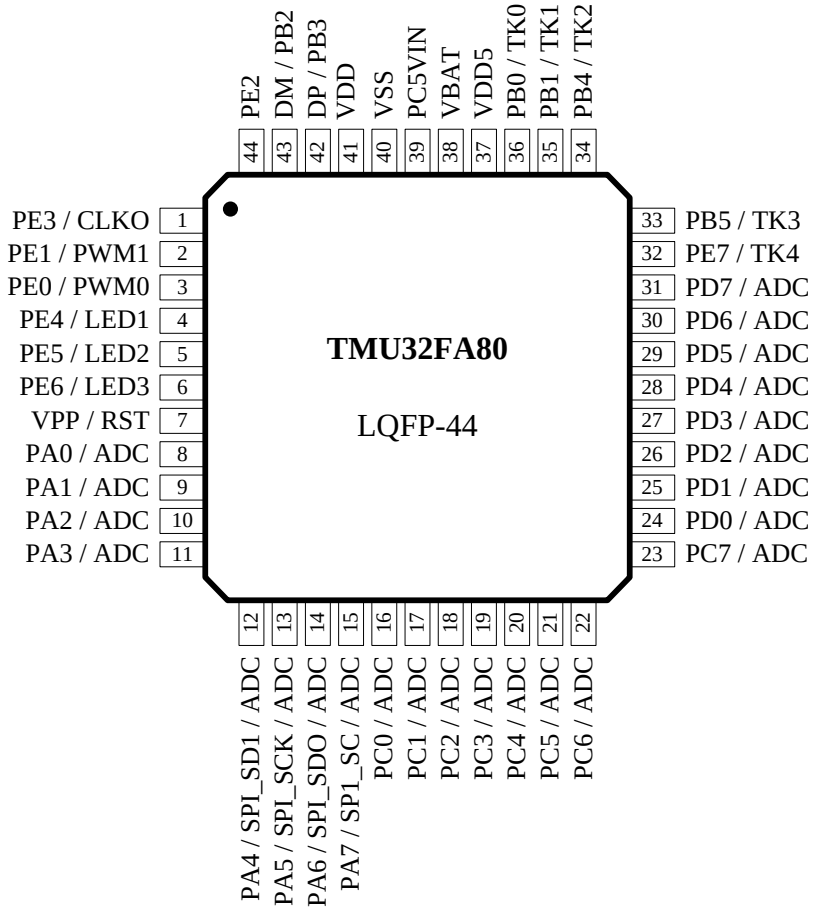
Block Diagram



TMU32FA80 Block Diagram

Pin Assignment Diagram





Pin Description

Name	In/Out	Pin Description
PA7-PA0 PB5-PB0 PC7~PC0 PD7~PD0 PE7~PE0	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or **open-drain** output. Pull-up resistors are assignable by software. Shared with ADC, touch key, SPI and PWM
VPP	I	Flash programming high voltage input / External active low reset
PC5VIN	I	PC5V detect Pin
VBAT	P	Battery Power Pin
VDD5	P	5V Power Pin
VDD, VSS	P	Power input pin and ground
DP/DM	I/O	USB Signal

Functional Description

1. CPU Core

1.1 Clock Scheme and Instruction Cycle

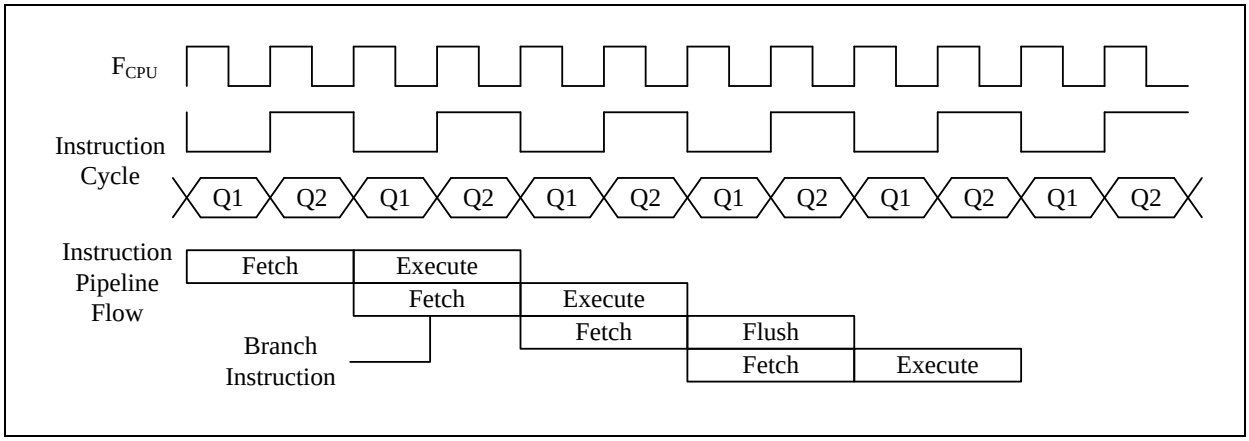
TMU32FA80 has 2 chip clock sources as following:

Firc : Internal RC oscillator 24 MHz clock

Sirc : Internal RC oscillator 128 KHz clock

F_{irc} can be synchronized by USB signals and used for USB module. Clock Control register R07[3] is used to enable the internal oscillator.R07[4] is used to enable the other slow RC(S_{irc}). F_{irc} and S_{irc} can also be used for CPU clock. R07[2] is used to select F_{irc} or S_{irc} 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.



1.2 CPU clock control register

CPU clock source selection: TMU32FA80 can select Fast internal RC oscillator or select Slow internal RC oscillator as the CPU clock source.

R07 [2] = 0, select Fast internal RC oscillator(F_{irc})

R07 [2] = 1, select Slow internal RC oscillator(S_{irc})

CPU clock speed selection: When F_{irc} clock source is selected, the clock source can be divided to 12Mhz, 6Mhz, 3Mhz or 1.5Mhz. 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

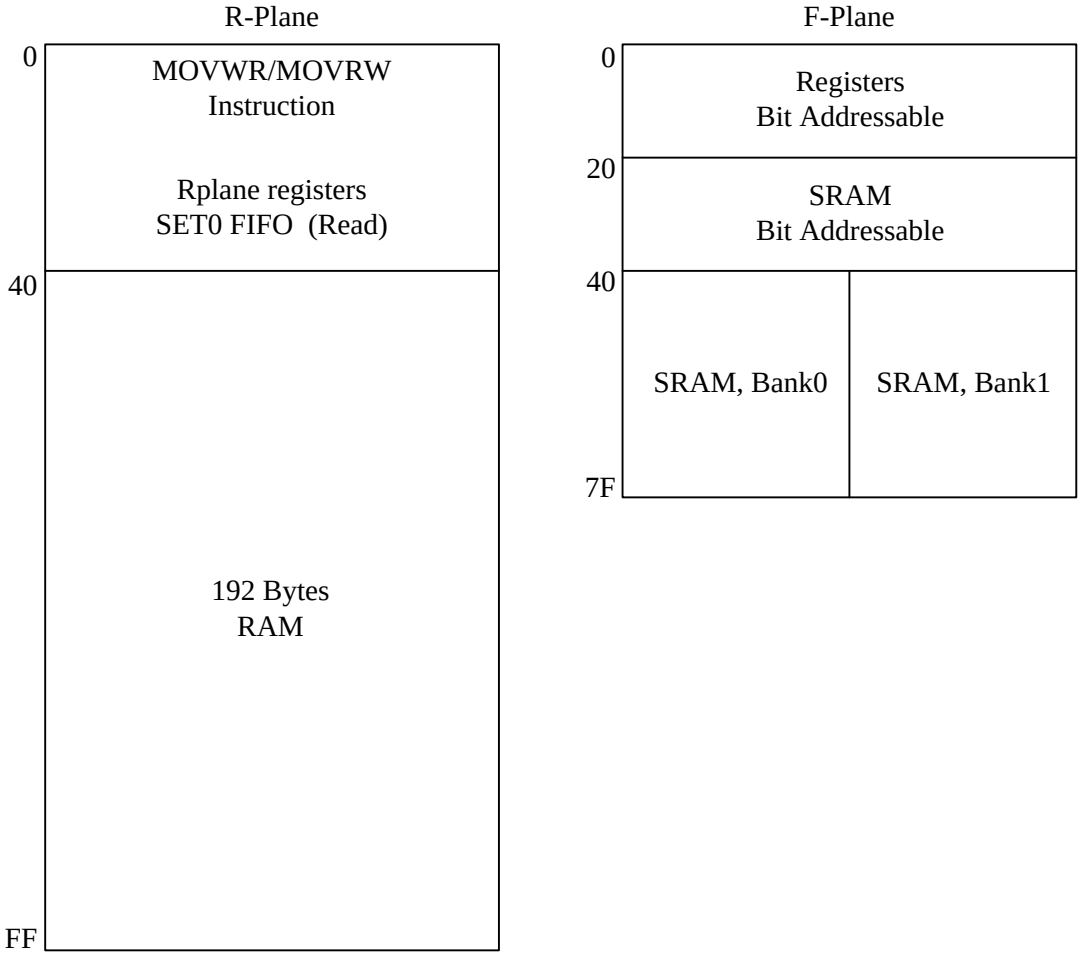
The Programming Counter is 13-bit wide capable of addressing an 8K x 14 program ROM. As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC value is normally increased by one except the followings. The Reset Vector (000h) and the Interrupt 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 8K 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 1FFFh. 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 original 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.

- 8K x 14 program FLASH.
- 160-byte SRAM (F-plane) is addressed from 0x20 to 0x7F which 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].
- 192-byte RAM(allocated in R-plane) used for EP0/EP1/EP2/EP3/EP4/EP5 buffer.
- 8-byte USB FIFO(allocated in R-plane) used for EP0 SETUP TOKEN buffer.



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)

Field / Legend	Description
[]	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 Instruction Set Description

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 B : PC = LABEL1 GOTO LOOP A : CNT = CNT + 1 CONTINUE if CNT=0, PC = CONTINUE if CNT≠0, PC = LABEL1+1

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

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

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

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

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

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

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

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

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

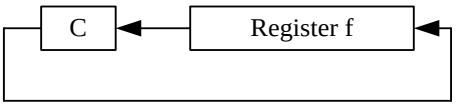
RETLW — Return with Literal in W

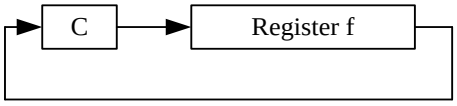
Syntax	RETLW k
Operands	k : 00h ~ FFh
Operation	PC ← TOS, (W) ← k
Status Affected	-
OP-Code	01 1000 kkkk kkkk
Description	The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Cycle	2
Example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

RET — Return from Subroutine

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

RLF — Rotate Left f through Carry

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

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

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

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

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

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

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

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

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 [7:0]							
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[7:0] Port A Pin data output register							
07h	xx11-1111	PBD	PB[5:0] Port B Pin data output register							
08h	1111-1111	PCD	PC[7:0] Port C Pin data output register;							
09h	1111-1111	PDD	PD[7:0] Port D Pin data output register;							
0ah	1111-1111	PED	PE[7:0] Port E Pin data output register;							
0bh	X000-0000	ADC_CTRL	--	ADC_EOC	ADC_START	AD_CH[4]	AD_CH[3]	AD_CH[2]	AD_CH[1]	AD_CH[0]
0ch	0000-0000	TM2	TIMER 2 Counter							
0dh	0000-0000	TM1	TIMER 1 Counter							
0eh	xxxx-xx00	TM1IE	TIMER2/1 Interrupt Enable						TM2IE	TM1IE
0fh	xxxx-xx00	TM1IF	TIMER2/1 Interrupt flag, write 0 to clear it						TM2I	TM1I
10h	0000-0000	USBE	USBE	FUNADR						
11h	0000-0000	INTFLAG1	SET0I	OUT0I	TX0I	TX1I	RC2I	SUSPI	TX3I	RC4I
12h	0000-0000	INTFLAG2	TX5I	VDD5VRI	WKTI	RSTI	RSMI	ADCI	PB0I	TM0I
13h	0000-00x0	EPCFG	SUSP	RSMO	EP1CFG	EP2CFG	DEVR	EP4CFG	--	OUT0RDY
14h	0000-xxxx	EP0SET	TX0RDY	TX0TGL	EP0STALL	IN0STALL	--			
15h	000x-xxxx	EP1SET	TX1RDY	TX1TGL	EP1STALL	--				
16h	0000-xxxx	EP2SET	RC2RDY	RC2TGL	EP2STALL	EP2ERR	--			
17h	0000-0000	EP3SET	TX3RDY	TX3TGL	EP3STALL	EP3CFG	TX3CNT[3:0]			
18h	0000-0000	EP4SET	RC4RDY	RC4TGL	EP4STALL	RC4ERR	RC4CNT[3:0]			
19h	x000-0000	OUT0CNT	Endpoint 0 received byte count OUT0CNT[6:0]							
1ah	x000-0000	TX0CNT	Endpoint 0 transmit byte count TX0CNT[6:0]							
1bh	x000-0000	TX1CNT	Endpoint 1 transmit byte count TX1CNT[6:0]							
1ch	0000-0000	RC2CNT	Endpoint 2 received byte count RC2CNT[6:0]							
1dh	xx00-0000	SPISET	--	--	SPIMODE	SPIEN	LSBFIRST	SPIIN	SPISW	CLRADR
1fh	0000-0000	EP5SET	TX5RDY	TX5TGL	EP5STALL	EP5CFG	TX5CNT[3:0]			
20h : 7fh	xxxx-xxxx	BANK0	Internal RAM 96 Bytes (BANK0)							
	xxxx-xxxx	BANK1	Internal RAM 96 Bytes (BANK1)							

2.2 F-Plane Description

Name		Addr.	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	07.5~0	R/W	ff	Port B output data, PBD[5:0];
F08	PCD	08.7~0	R/W	ff	Port C output data;
F09	PDD	09.7~0	R/W	ff	Port D output data, PDD[7:0];
F0A	PED	0A.7~0	R/W	f	Port E output data, PED[7:0];
F0B	ADC_EOC	0B.6	R	0	ADC End of Conversion Flag
	ADC_START	0B.5	R/W	0	ADC Start
	ADC_CH[4:0]	0B.4~0	R/W	0	ADC Channel Selection;; Total 24 Channels
F0C	TM2	0C.0~7	R/W	0	Timer2
F0D	TM1	0D.0~7	R/W	0	Timer1
F0E	TM2IE TM1IE	0E.1 0E.0	R/W R/W	0 0	Timer2 Interrupt Enable Timer1 Interrupt Enable
F0F	TM2IF TM1IF	0F.1 0F.0	R/W R/W	0 0	Timer2 Interrupt flag, write 0 to clear it 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 Bulk Transmit Interrupt flag, write 0 to clear flag.
	RC2I	11.3	R/W	0	Endpoint 2 Bulk Receive 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 Transmit Interrupt flag, write 0 to clear flag.
	RC4I	11.0	R/W	0	Endpoint 4 Receive Interrupt flag, write 0 to clear flag.

Name		Addr.	R/W	Rst	Description
F12	TX5I	12.7	R/W	0	Endpoint 5 Transmit Interrupt flag, write 0 to clear flag.
	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.
	ADCI	12.2	R/W	0	ADC 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 mode.
	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 enable
	EP4CFG	13.2	R/W	0	Set Endpoint 4 configured
	OUT0RDY	13.0	R/W	0	Endpoint 0 ready for receive, cleared by H/W while OUT0I occurs.
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 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)
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 transmit DATA1/DATA0 packet.
	EP1STALL	15.5	R/W	0	Endpoint 1 will stall IN packet.
F16	RC2RDY	16.7	R/W	0	Endpoint 2 ready for receive, cleared by H/W while RC2I occurs.
	RC2TGL	16.6	R	-	Endpoint 2 received DATA1/DATA0 packet.
	EP2STALL	16.5	R/W	0	Endpoint 2 will stall Out packet.
	EP2ERR	16.4	R	0	Endpoint 2 received data error
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 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.
	TX3CNT	17.3~0	R/W	0	Endpoint 3 transmit data byte count

Name		Addr.	R/W	Rst	Description
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 received DATA1/DATA0 packet
	EP4STALL	18.5	R/W	0	Endpoint 4 will stall OUT packet.
	RC4ERR	18.4	R/W	0	EP4 received data error.
	RC4CNT	18.3~0	R	0	Endpoint 4 received byte count
F19	OUT0CNT	19.6~0	R	0	Endpoint 0 OUT received byte count.
F1A	TX0CNT	1A.6~0	R/W	0	Endpoint0 IN transmit byte count.
F1B	TX1CNT	1B.6~0	R/W	0	Endpoint1 IN transmit byte count.
F1C	RC2CNT	1C.6~0	R	0	Endpoint2 OUT received byte count
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
F1F	TX5RDY	1F.7	R/W	0	Endpoint 5 ready for transmit, clear by
	TX5TGL	1F.6	R/W	0	Endpoint 5 transmit DATA1/DATA0 packet
	EP5STALL	1F.5	R/W	0	Endpoint 5 will stall IN packet
	EP5CFG	1F.4	R/W	0	Set Endpoint 5 configured
	TX5CNT	1F.3~0	R/W	0	Endpoint 5 transmit byte count
	SRAM	20~7F	R/W	-	Internal RAM (96 Bytes x 2 Banks)

2.3 R-Plane

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[3:0]			
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	WDTSET	--	WDTPSC		WKTPSC		5VINFLG	OUTFLG	--
07h	xx01-1000	CLKSEL	--	--	HWAUTO	SIRCEN	FIRCEN	CLKSEL	CLKDIV	
08h	xxxx-0000	SIRCTRIM					SIRCTRIM[3:0]			
09h	xxxx-xx00	IRCCKO	--						PE3CKO	PE3SEL
0ah	xx00-0000	TM1EN	--	--	TM1EN	TM1SEL	TM1PSC			
0bh	xxxx-xx00	TM2EN	--	--	TM2EN	--	TM2PSC			
0ch	0000-0000	ADCQL	ADCQL; ADC End of Conversion data[7:0]							
0dh	0000-0000	ADCQH	ADCQH; ADC End of Conversion data[11:8]							
0eh	x000-x000	TKE	--	TKE	TKSPD[1..0]		--	TKSEL[2:0]		
0fh	0000-0000	EPSEL	--						EPSEL[1:0]	
11h	0000-0000	USBINTEN	SET0IE	OUT0IE	TX0IE	TX1IE	RC2IE	SUSPIE	TX3IE	RC4IE
12h	0000-0000	FUNINTEN	TX5IE	VDD5VIE	WKTIE	RSTIE	RSMIE	ADCIE	PB0IE	TM0IE
13h	xxxx-xxxx	RC0SET	RC0TGL	RC0ERR	EP0DIR	EP0IND	--			
14h	xxxx-x000	ADCKSEL	--					ADCKSEL[2:0]		
15h	1111-1111	PADI_EN	PA[7:0] Schmitt trigger input enable							
16h	1111-1111	PCDI_EN	PC[7:0] Schmitt trigger input enable							
17h	1111-1111	PDDI_EN	PD[7:0] Schmitt trigger input enable							
18h : 1fh	xxxx-xxxx	SET0FIFO	Endpoint 0 SETUP receive Buffer (8 Bytes)							
20h	0000-0000	PAE	PA[7:0] CMOS push-pull output enable							
21h	xx00-0000	PBE	PB[5:0] CMOS push-pull output enable							
22h	0000-0000	PCE	PC[7:0] CMOS push-pull output enable							
23h	0000-0000	PDE	PD[7:0] CMOS push-pull output enable							
24h	0000-0000	PEE	PE[7:0] CMOS push-pull output enable							
25h	0000-0000	PAPUN	PA[7:0] pull-up ; 0: enable							
26h	0000-0000	PBPUN	PB[5:0] pull-up; 0: enable							
27h	0000-0000	PCPUN	PC[7:0] pull-up; 0: enable							
28h	0000-0000	PDPUN	PD[7:0] pull-up ; 0: enable							
29h	0000-0000	PEPUN	PE[7:0] pull-up; 0: enable							
30h	xxxx-0000	PWM0ENF	--				PWM0PSC[2:0]			PWM0EN
31h	0000-0000	PWM0DUTY	PWM0 DUTY							
32h	0000-0000	PWM0PRD	PWM0 Period							
33h	0000-0000	PWM1ENF	--				PWM1PSC[2:0]			PWM1EN
34h	0000-0000	PWM1DUTY	PWM1 DUTY							
35h	0000-0000	PWM1PRD	PWM1 Period							
36h	xx00-0000	PWMnENF	--	--	PWM4EN	PWM3EN	PWM2EN	PWMnPSC[2:0]		

Addr	RST	NAME	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
37h	0000-0000	PWMnPRD	PWMn Period(PWM2/3/4 use same Period)							
38h	0000-0000	PWM2DUTY	PWM2 DUTY							
39h	0000-0000	PWM3DUTY	PWM3 DUTY							
3ah	x000-0000	PWM4DUTY	PWM4 DUTY							
3bh	xx00-0000	SPIENF	--	--	CPOL	CPHA	BSL[3:0]			
3ch	x000-0000	SPICRS	--	SPICRS[6:0] SPI Clock Select						
3dh	x000-0000	SPILEN	--	SPILEN[6:0] SPI Transfer length						
3eh	0000-0000	SPITX	SPITX[7:0] SPI Transmit DATA in CMD phase							
3fh	xxxx-xxxx	SPIRX	SPIRX[7:0] SPI Received DATA							
40h : ffh	xxxx-xxxx	XRAM	Endpoint 0/1/2/3/4/5 IN/OUT Buffer (192 Bytes)							

2.4 R-Plane Description

Name		Address	R/W	Rst	Description
R01	TM0RLD	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	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
	SIRCEN	07.4	W	1	Slow IRC(WRC) Enable(1)
	FIRCEN	07.3	W	1	Fast IRC Enable(1)
	CLKSEL	07.2	W	0	System Clk Source Select, 0: FIRC, 1: SIRC
	CLKDIV	07.1~0	W	11	FIRC System Clk Period Selection
					2'b00: 12 MHz
					2'b01: 6 MHz
					2'b10: 3 MHz
					2'b11: 1.5 MHz
R08	SIRC_TRIM	08.3~0	R/W	0	SIRC(WRC) Trim[3:0]
R09	PE3CKO	09.1	W		PE3 is used as IO port or FIRC CKO ; 0: IO port, 1: FIRC CKO
	PE3SEL	09.0	W		When PE3CKO= 1, PE3SEL is used to select FIRC clock frequency; 0: FIRC 12 MHz output, 1: FIRC 6 MHz output
R0A	TM1EN	0A.5	W	0	Timer1 enable
					1= use TKRC as Timer1/PSC clock;
					0= use Instruction Cycle as Timer1/PSC clock
	TM1PSC	0A.3~0	W	0	Timer1 Pre-Scale, 0:div1, 7:div128, 8:div256
R0B	TM2EN	0B.5	W	0	Timer2 enable
	TM2PSC	0B.4~3	W	0	Timer2 Pre-Scale, 0:div1, 7:div128, 8:div256
R0C	ADCQL	0C.7~0	R	-	ADC End of Conversion data ADCQ[7:0]
R0D	ADCQH	0D3~0	R	-	ADC End of Conversion data ADCQ[11:8]
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

Name		Address	R/W	Rst	Description
R0F	EPSEL	0F.1~0	W	0	Endpoint configuration selection
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
	RC2IE	11.3	W	0	RC2I 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	TX5IE	12.7	W	0	TX5I Interrupt enable
	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
	ADCIE	12.2	W	0	ADC 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.
R14	ADCKSEL	14.2~0	W	0	ADC CLK Selection; 0:1.5M, 1:1M, 2:500K, 3:250K, 4:125K ⇨ Conversion rate 0:30K, 1:20K, 2:10K, 3:5K, 4:2.5K
R15	PADI_EN	15.7~0	W	ff	1: Enable Port A Schmitt trigger input
R16	PCDI_EN	16.7~0	W	ff	1: Enable Port C Schmitt trigger input
R17	PDDI_EN	17.7~0	W	ff	1: Enable Port D Schmitt trigger input
	SET0FIFO	18 ~ 1F	R	-	Endpoint 0 SETUP Receive Buffer(8 bytes)
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
R22	PCE	22.7~0	W	0	Port C CMOS push-pull output enable
R23	PDE	23.7~0	W	0	Port D CMOS push-pull output enable
R24	PEE	24.7~0	W	0	Port E CMOS push-pull output enable
R25	PAPUN	25.7~0	W	0	Port A pull-up, 0=enable
R26	PBPUN	26.7~0	W	0	Port B pull-up, 0=enable
R27	PCPUN	27.7~0	W	0	Port C pull-up, 0=enable

Name		Address	R/W	Rst	Description
R28	PDPUN	28.7~0	W	0	Port D pull-up, 0=enable
R29	PEPUN	29.7~0	W	0	Port E pull-up, 0=enable
R30	PWM0PSC	30.3~1	W	0	PWM0 clock prescale; 00= clk/2, 01= clk/4, 10= clk/8, 11= clk/16
	PWM0EN	30.0	W	0	PWM0 Enable bit
R31	PWM0DUTY	31.7~0	W	0	PWM0 Duty
R32	PWM0PRD	32.7~0	W	0	PWM0 Period
R33	PWM1PSC	33.3~1	W	0	PWM1 clock prescale; 00= clk/2, 01= clk/4, 10= clk/8, 11= clk/16
	PWM1EN	33.0	W	0	PWM1 Enable bit
R34	PWM1DUTY	34.7~0	W	0	PWM1 Duty
R35	PWM1PRD	35.7~0	W	0	PWM1 Period
R36	PWM4EN	36~5	W	0	PWM4 Enable bit
	PWM3EN	36.4	W	0	PWM3 Enable bit
	PWM1EN	36.3	W	0	PWM2 Enable bit
	PWMnPSC	36.2~0	W	0	PWM2/3/4 Prescale
R37	PWMnPRD	37.7~0	W	0	PWM2/3/4 Period
R38	PWM2DUTY	38.7~0	W	0	PWM2 Duty
R39	PWM3DUTY	39.7~0	W	0	PWM3 Duty
R3A	PWM4DUTY	3A.7~0	W	0	PWM4 Duty
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
	XRAM	40~FF	R/W	-	Internal RAM 192 Bytes for USB Endpoint Buffer

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. There are 5 endpoints 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 USB Endpoint

The TMU32FA80 supports up to 5 endpoints. R0F[1:0] (EPSEL) is used to determine how many endpoint is use and the buffer size.

EP mapping

EPSEL [1 : 0]	EP0 (setup)	EP0 IN/OUT	EP1 Bulk / Interrupt IN transfer	EP2 Bulk / Interrupt OUT transfer	EP3 Interrupt IN transfer	EP4 Interrupt OUT transfer	EP5 Interrupt IN transfer	Release for User
00	Fixed 8-byte	8 byte	64 byte	64 byte	8 byte	8 byte	8 byte	32 bytes
01		8 byte	32 byte	32 byte	8 byte	8 byte	8 byte	88 bytes
10		64 byte	64 byte	64 byte	NA	NA	NA	NA
11		64 byte	32 byte	32 byte	8 byte	8 byte	8 byte	40 bytes

3.3 Endpoint 0 Receive (SET0/OUT0)

After receiving a SETUP packet and placing the data into the Endpoint 0 setup receive FIFO (SET0FIFO), TMU32FA80 updates the Endpoint 0 status registers to record the receive status and then generates an Endpoint 0 setup receive interrupt (SET0I). The received data is 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 is stored into EP0 Buffer, F/W can read the status register F13, F19 and R13 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 is written into EP0 Buffer and the OUT0CNT is updated unless Endpoint 0 STALL (EP0STALL)is set or Endpoint 0 receive ready(OUT0RDY) is not clear. The SIE clears the OUT0RDY automatically and generates OUT0I interrupt when the OUT0CNT or EP0 Buffer 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 EP0 Buffer is read out

3.4 Endpoint 0 Transmit (TX0)

After detecting a valid Endpoint 0 IN token, TMU32FA80 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 TMU32FA80 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.

3.5 Endpoint 1 transmit(TX1)

Endpoint 1 is capable of transmit only. Register F13, F15 and F1B are used to control this endpoint. This endpoint is enabled when the Endpoint 1 configuration control bit (EP1CFG) is set. After detecting a valid Endpoint 1 IN token, TMU32FA80 automatically transmit the data pre-stored in the Endpoint 1 transmit buffer(EP1 Buffer) to the USB bus if the Endpoint 1 transmit ready flag (TX1RDY) is set and the EP1STALL is cleared. The number of byte to be transmitted depends on the Endpoint 1 transmit byte count register (TX1CNT). The DATA0/1 token to be transmitted depends on the Endpoint 1 transmit toggle control bit (TX1TGL). After the EP1 Buffer is updated, TX1RDY should be set to 1. This enables the TMU32FA80 to respond to an Endpoint 1 IN packet. TX1RDY is cleared and an Endpoint 1 transmit interrupt (TX1I) is generated once the USB host acknowledges the data transmission. The interrupt service routine can check TX1RDY to confirm that the data transfer was successful.

3.6 Endpoint 2 receive (RC2)

Endpoint 2 is capable of receive only. Register F13, F16 and F1C are used to control this endpoint. This endpoint is enable when Endpoint 2 configured control bit (EP2CFG) is set. After detecting a valid Endpoint 2 OUT token, the TMU32FA80 automatically stores the bulk out data into the specified Bulk out buffer (EP2 Buffer) and updates RC2CNT if the Endpoint 2 receiving ready flag (RC2RDY) is set and the EP2STALL is cleared. The DATA0/DATA1 token to be checked is toggled by F/W. When an Endpoint 2 receive interrupt (RC2I) is generated, the RC2RDY is cleared. During the packet transfer stage, if data is to check error, will response on RC2ERR.

3.7 Endpoint 3 transmit (TX3)

Endpoint 3 is capable of transmit only. Register F17 is used to control this endpoint. Endpoint 3 is enabled when the configuration control bit (EP3CFG) is set. 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 3 IN token, TMU32FA80 automatically transmits the data pre-stored in the transmit buffer(EP3 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 3 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 was successful.

3.8 USB Endpoint 4 receive (RC4)

Endpoint 4 is capable of receive only. Register F13 and F18 are used to control this endpoint. This endpoint is enable when Endpoint 4 configured control bit (EP4CFG) is set. After detecting a valid Endpoint 4 OUT token, the TMU32FA80 automatically stores the bulk out data into the EP4 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 is to check error, will response on RC4ERR.

3.9 Endpoint 5 transmit (TX5)

Endpoint 5 is capable of transmit only. Register F1F is used to control this endpoint. Endpoint 5 is enabled when the configuration control bit (EP5CFG) is set. Once this endpoint is enabled, F/W should set the Toggle bit (TX5TGL) and set the transmit byte count register (TX5CNT). After detecting a valid Endpoint 5 IN token, TMU32FA80 automatically transmits the data pre-stored in the transmit buffer(EP5 Buffer) to the USB bus if the Endpoint 5 transmits ready flag (TX5RDY) is set and the EP5STALL is cleared. The number of byte to be transmitted depends on the Endpoint 5 transmit byte count register (TX5CNT). The DATA0/1 token to be transmitted depends on the Endpoint 5 transmit toggle control bit (TX5TGL). Once the USB host acknowledges the data transmission, Endpoint 5 transmit interrupt (TX5I) is generated and the TX5RDY will be cleared. The interrupt service routine can check TX5RDY to confirm that the data transfer was successful.

3.10 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 resister. Other USB status flag includes the USB reset interrupt (RSTI), Resume input interrupt (RSMI), and USB Suspend interrupt (SUSPI).

3.11 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 force 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 TMU32FA80 send Resume signaling to USB bus when SUSP=1 and RSMO=1.

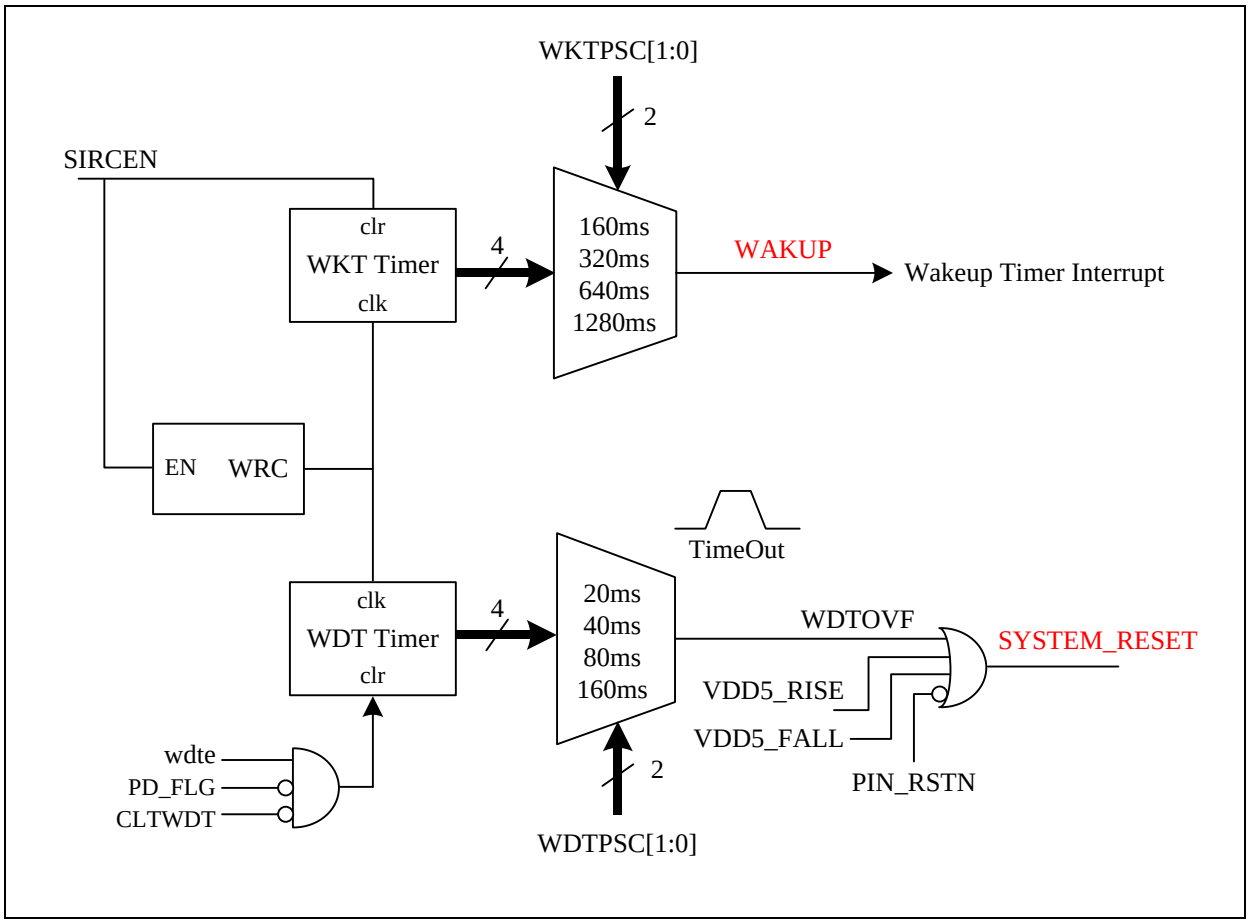
3.12 Interrupt Vector

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 Bulk Transmit Interrupt
05	USB Endpoint 2 Bulk Receive Interrupt
06	USB Suspend Interrupt
07	USB Endpoint 3 Transmit Interrupt
08	USB Endpoint 4 Receive Interrupt
09	USB Bus Reset Interrupt
0a	USB Resume Interrupt
0b	USB Endpoint 5 Transmit Interrupt
0c	Wakeup Timer Interrupt
0d	Timer0 Interrupt
0e	PB0 External I/O Interrupt
0f	ADC Interrupt
10	VDD5V Rise Interrupt
11	Timer1 Interrupt
12	Timer2 Interrupt

4. Wakeup Timer and Watch Dog Timer

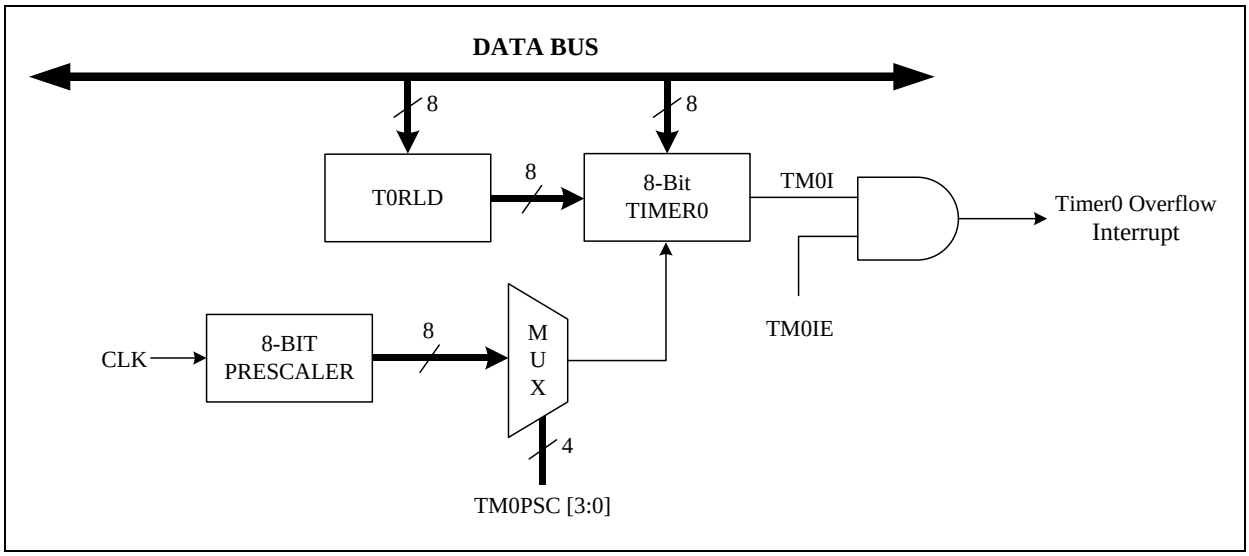
The WKT and WDT use the same internal RC (WRC). This internal RC (WRC) can be disabled by setting R07[4] “Low” for power saving. The overflow period of WDT can be selected from 20 ms to 80 ms and the wakeup period of WKT can be selected from 160 ms to 1280 ms. The WDT is enabled and cleared by the CLRWDT 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 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.

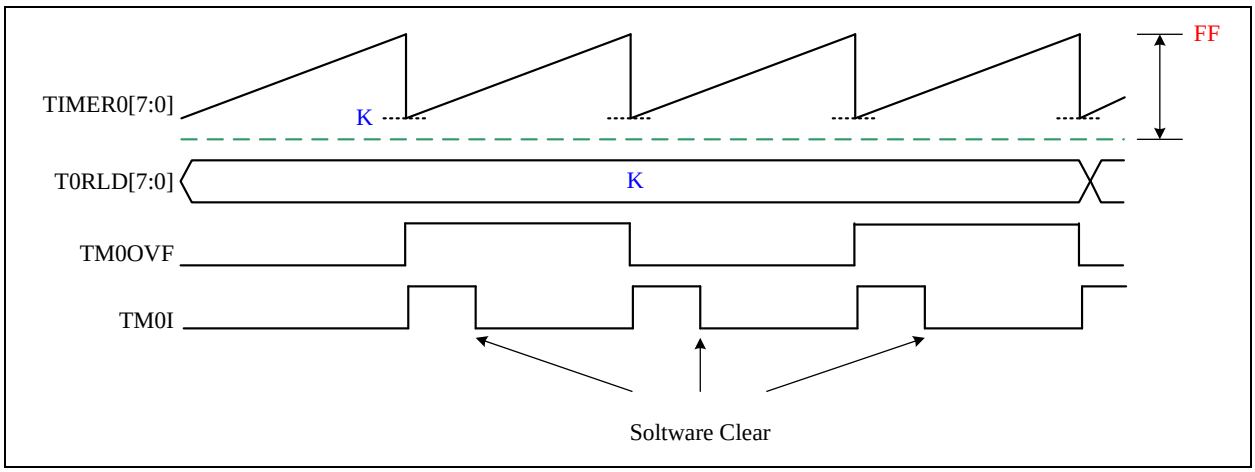


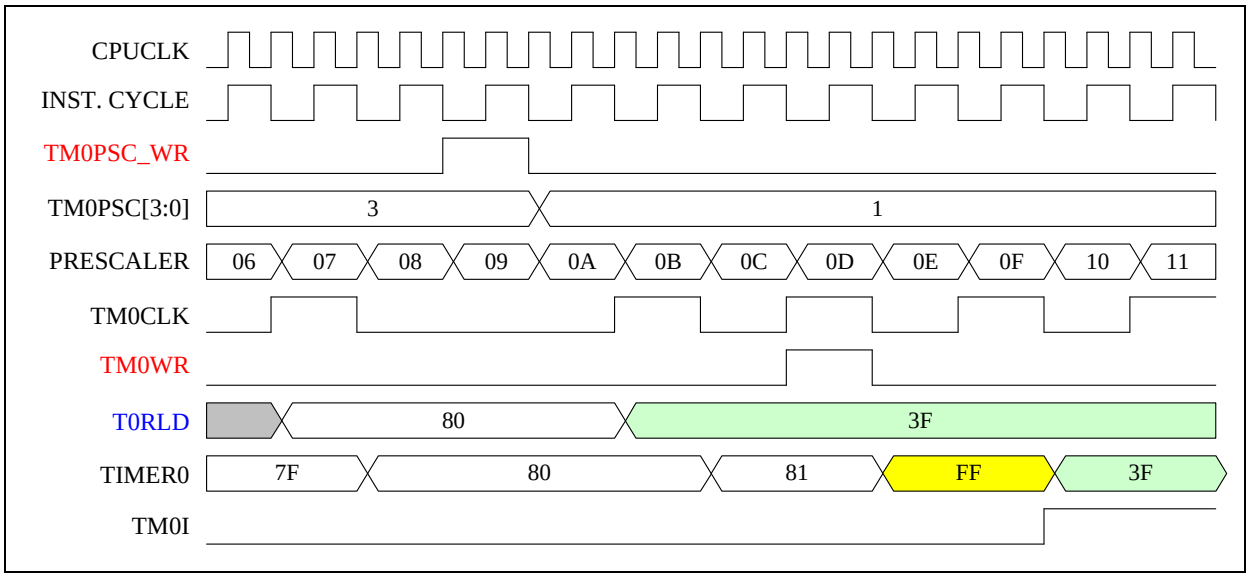
5. TIMER

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" (TM0RLD) while it rolls over based on the pre-scaled instruction clock. The Timer0 increase rate is determined by "Timer0 Pre-Scale" (TM0SCL) register in R-Plane. The Timer0 can generate interrupt (TM0I).

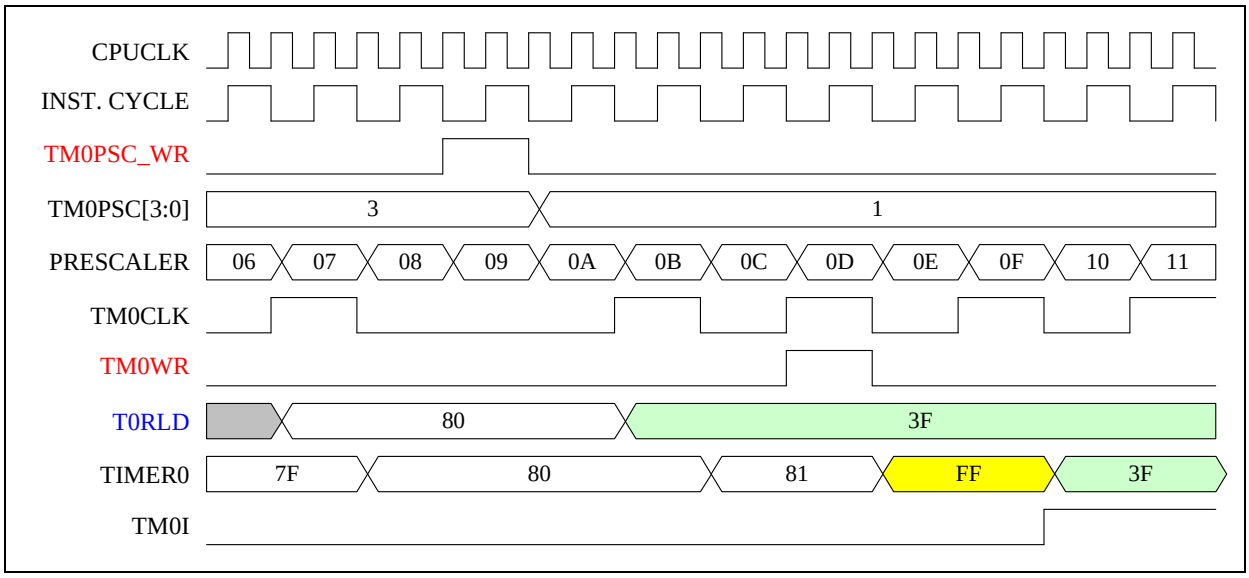


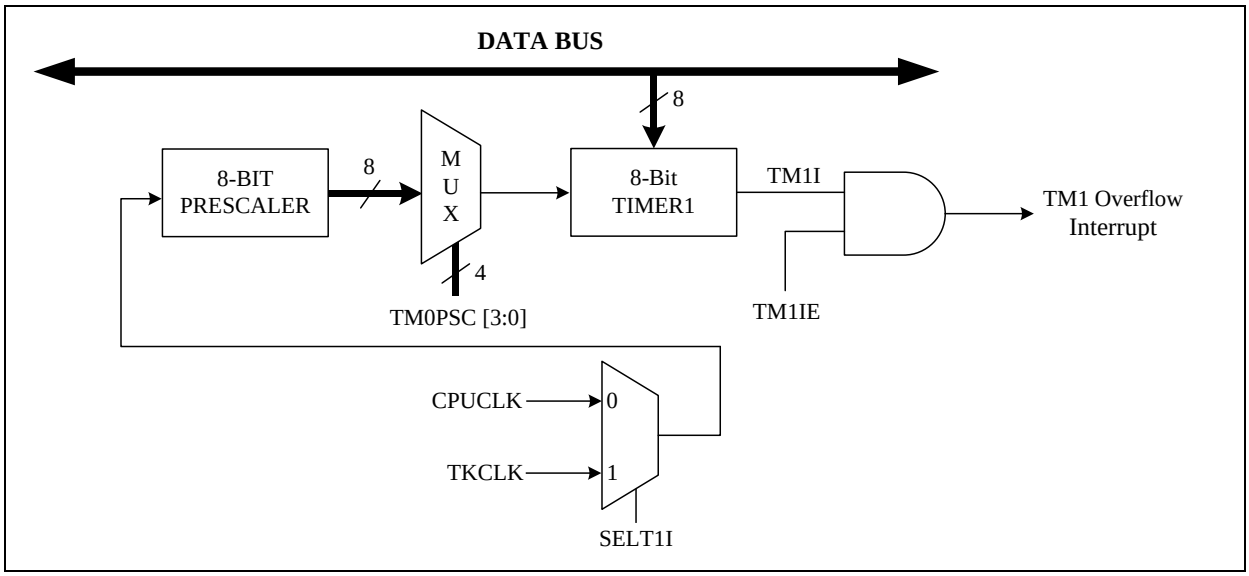



CPUCLK
INST. CYCLE
TM0PSC_WR
TM0PSC[3:0]
3
1
PRESCALER
06
07
08
09
0A
0B
0C
0D
0E
0F
10
11
TM0CLK
TM0WR
T0RLD
80
3F
TIMER0
7F
80
81
FF
3F
TM0I

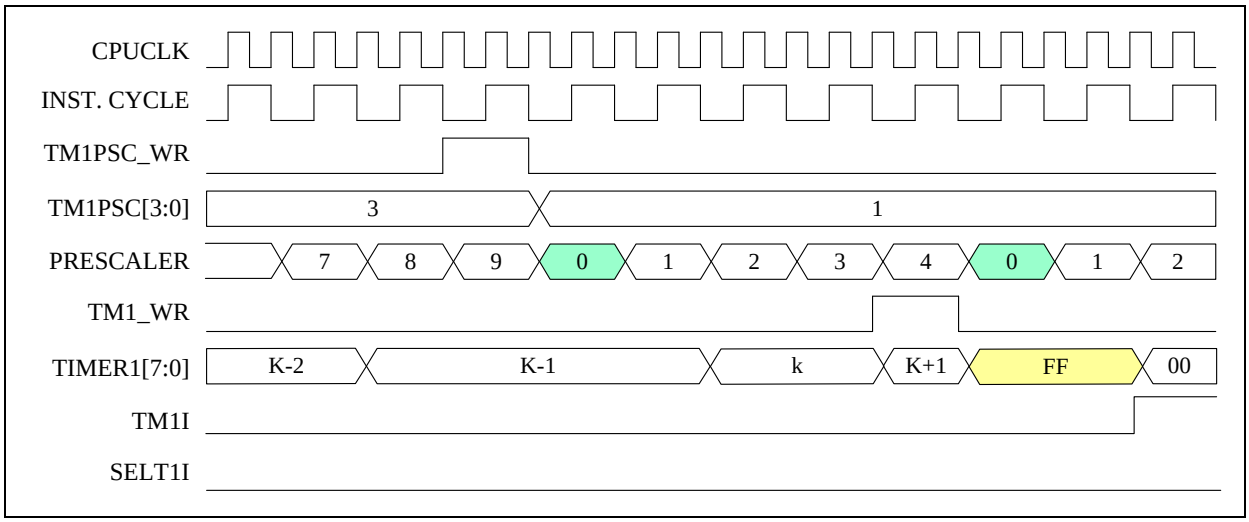
5.2 Timer1: 8-bit Timer/Counter with Pre-scale (PSC)

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.

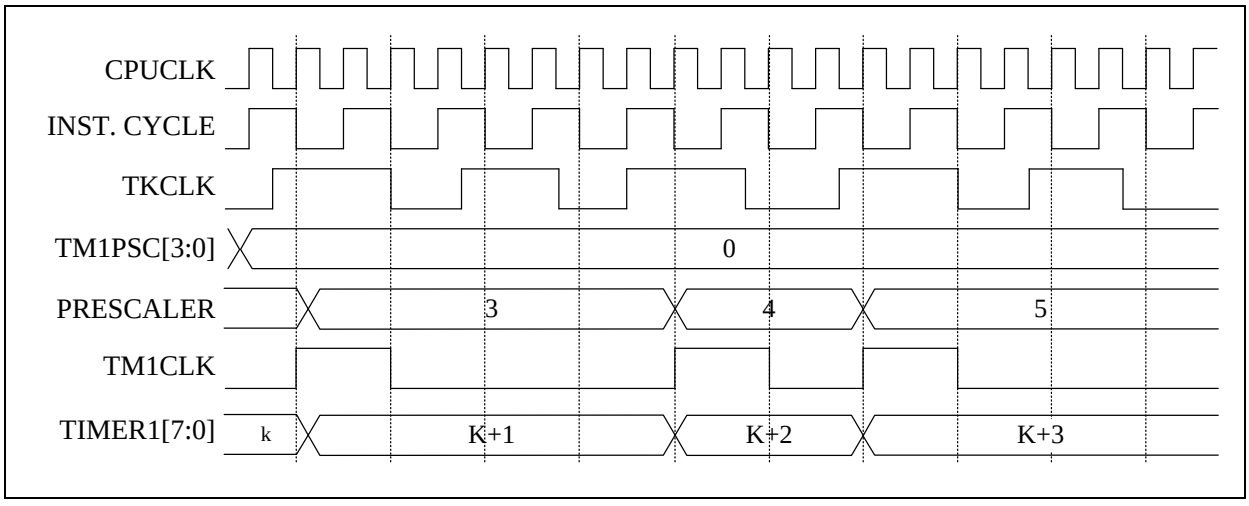




When Timer1 works in pure timer mode, the Timer1 prescaler (TM1PSC) is written, the internal 8-bit prescaler will be cleared to 0 to make the counting period correct at the first Timer1 count. TM1WR is the internal signal that indicates the Timer1 is directly written by instruction; meanwhile, the internal 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.

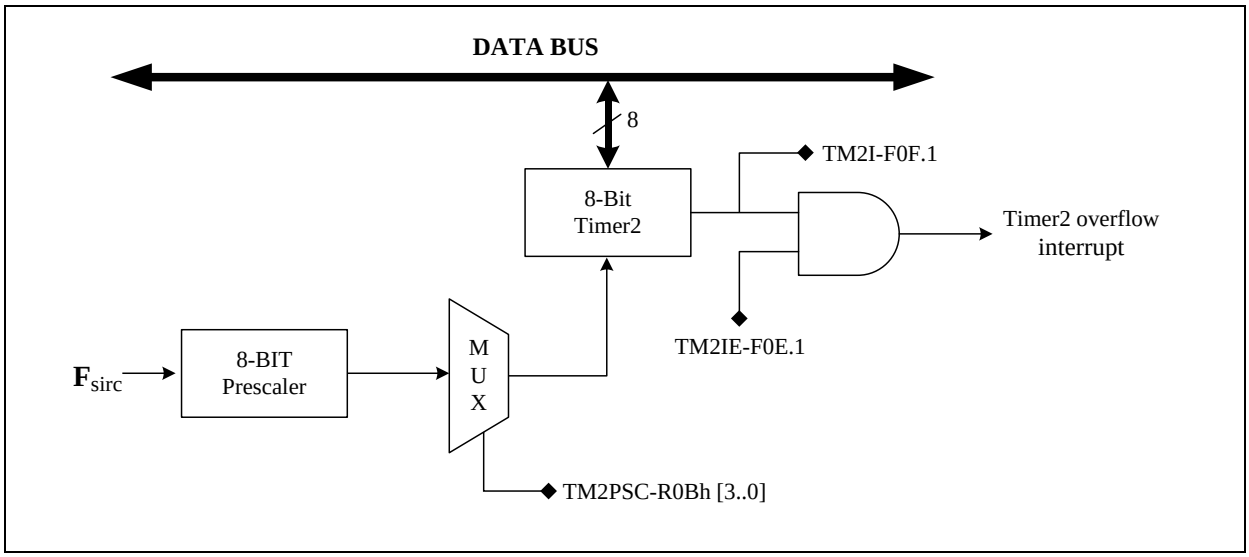


The following timing diagram describes the Timer1 works in counter mode. If SELT1I=1 then the Timer1 counter source clock is from Touch Key module that depends on TKE bit. In this mode the counter is used for Touch Key function.



5.3 Timer2: 8-bit Timer/Counter with Pre-scale (PSC)

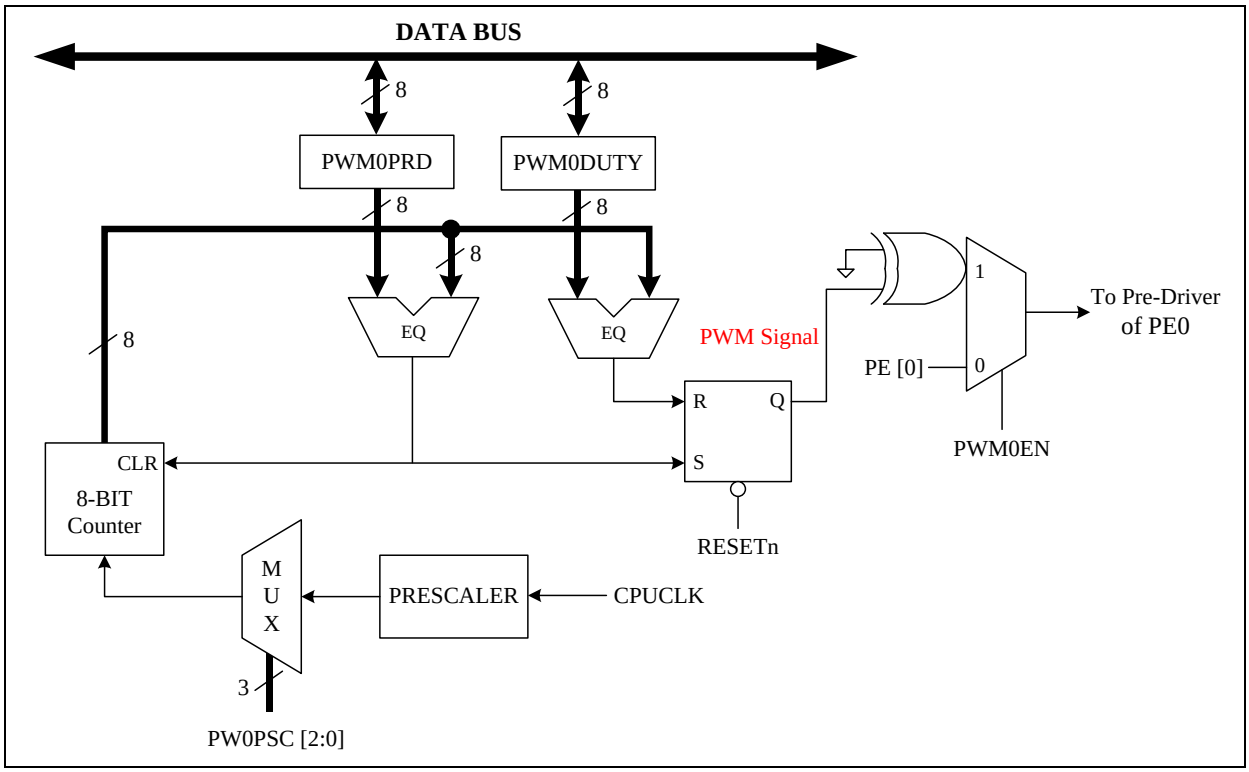
The Timer2 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer2 increases itself periodically and automatically rolls over based on the pre-scaled clock source, which is the SIRC clock. The Timer2 increase rate is determined by “Timer2 Pre-Scale” (TM2PSC) register in R-Plane. The Timer2 can generate interrupt (TM2I) when it rolls over. The main purpose of Timer2 is used to cooperate with the other Timer (TM0 or TM1) to adjust the SIRC clock frequency by changing R08 [3:0] (SIRC_TRIM).

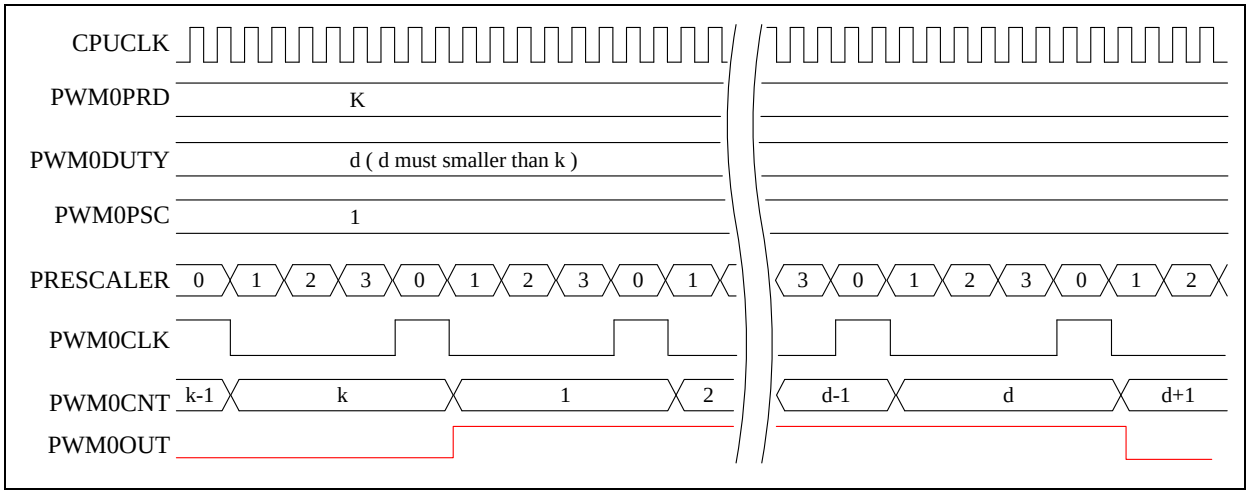


6. 8-bit PWM

6.1 PWM0

The PWM0 will be enabled by setting R30[0](PWM0EN) to “1”. Once the PWM0EN is set, the PWM0 8-bit counter starts to count and the PWM0 will be output to PE0. The PWM0 increase rate is determined by R30[3:1](PWM0PSC). The PWM output0 signal toggles to low level whenever the 8-bit counter matches register R31 (PWM0DUTY) and toggles to high level whenever the 8-bit counter matches register R32 (PWM0PRD). 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 completes. When the current PWM period is finished, the new value of PWMDUTY will be updated.





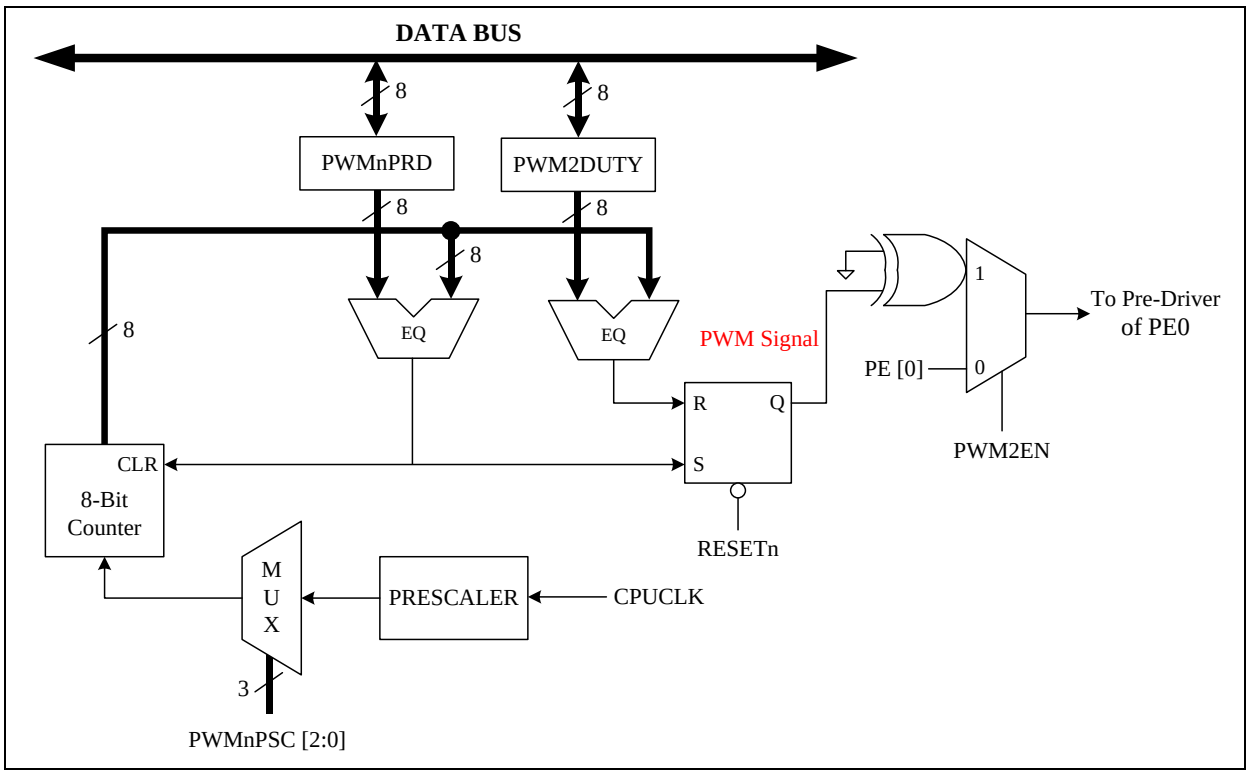
6.2 PWM1

The PWM1 is almost the same as PWM0, except the control register and the output pin. PWM1 is enable by R33[0] (PWM1EN). The PWM1PSC is in R33[3:1], PWM1DUTY is in R34 and PWM1PRD is in R35. When PWM1 is enabled , the PWM1 output will be output to PE1.

6.3 PWM 2/3/4

The PWM 2/3/4 is used to control the LED is on or off. Each PWM 2/3/4 has independent enable bit and duty value, but share the same prescale value and period. Register R36 [5] (PWM4EN) is used to enable PWM4, R36 [4] (PWM3EN) is used to enable PWM3 and R36 [3] (PWM2EN) is used to enable PWM2. Register R36 [2:0] (PWMnPSC) is the Prescale value for PWM 2/3/4 and register R37 (PWMnPRD) is the Period value for PWM 2/3/4. The Duty value is put in R38 (PWM2DUTY), R39 (PWM3DUTY) and R39 (PWM4DUTY). When PWM2 is enabled, the PWM2 will output to PE4. When PWM3 is enabled, the PWM3 will output to PE5. When PWM4 is enabled, the PWM4 will output to PE6.

The figure shown below is the structure of PWM2. PWM3 and PWM4 use the same structure as PWM2 used.



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)

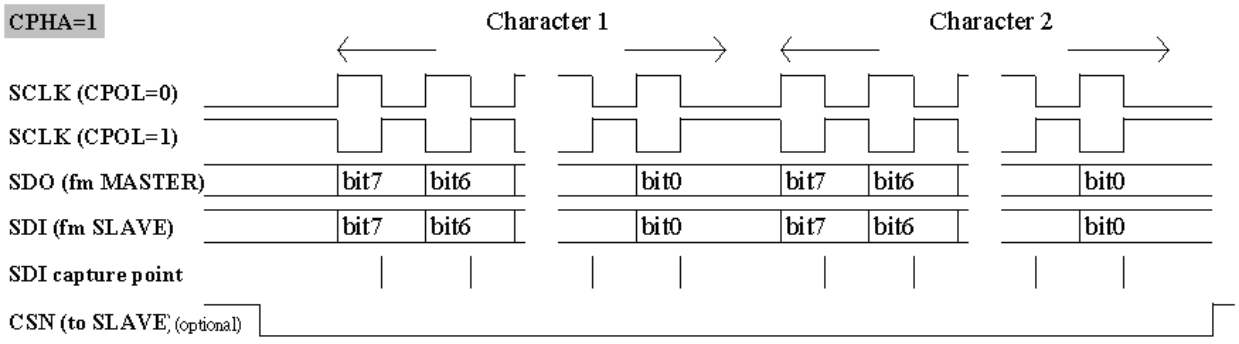
CRS[6:0]	SPI Clock Rate
0	6 Mbps
3	1.5 Mbps
15	375 kbps

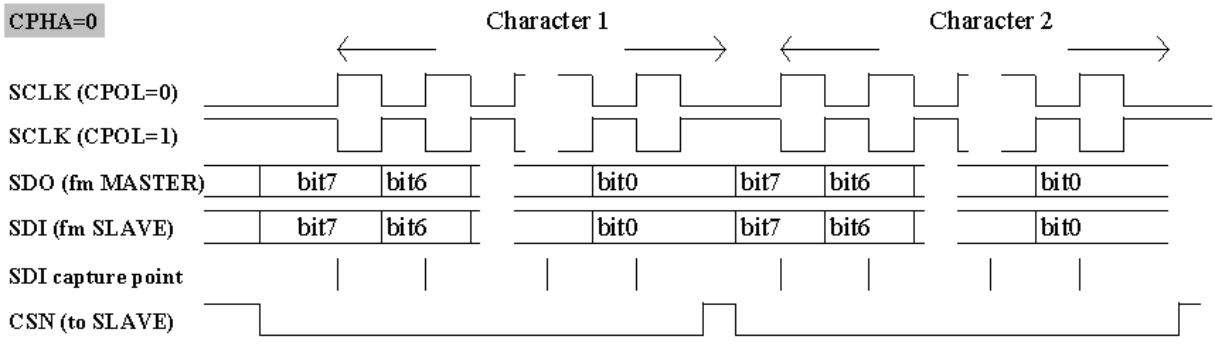
Note: CPUCLK = 12 MHz

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 mode; 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 (RAM1 or RAM2).

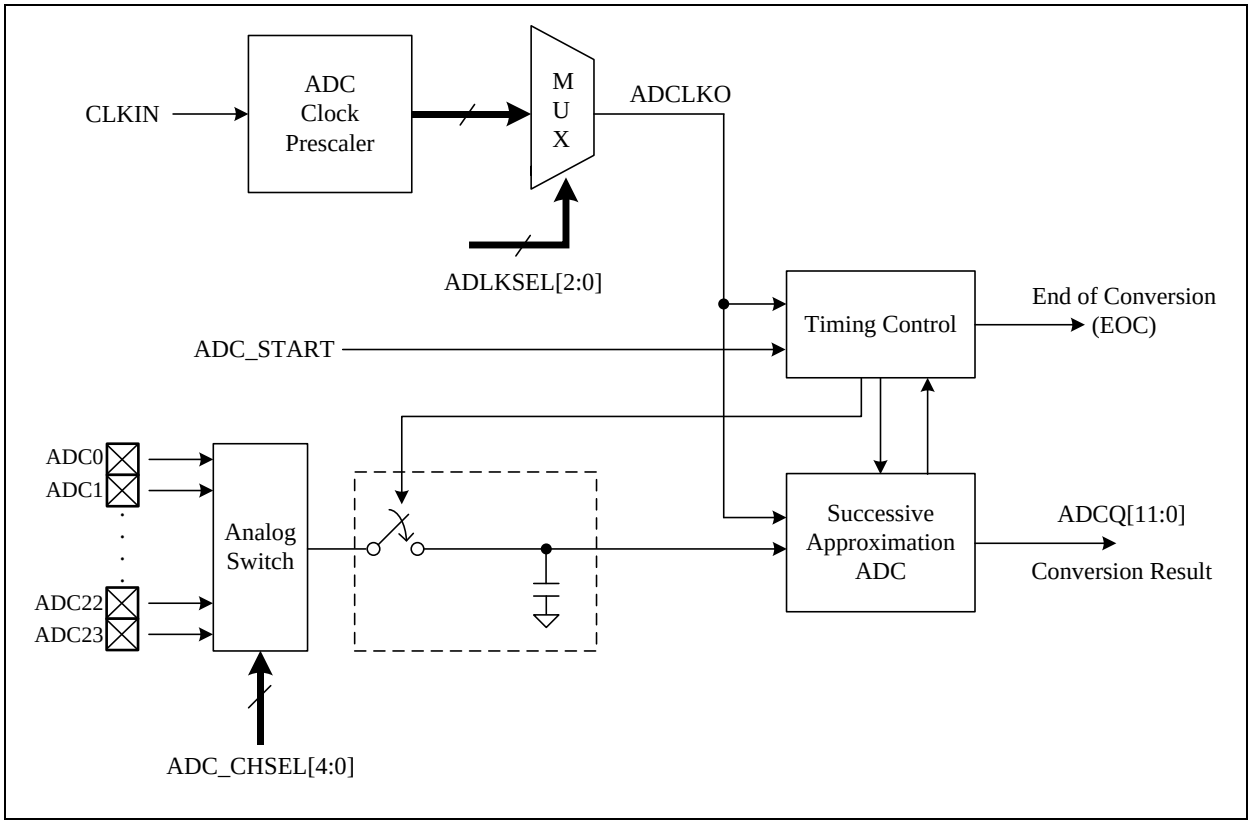
SPI Timing

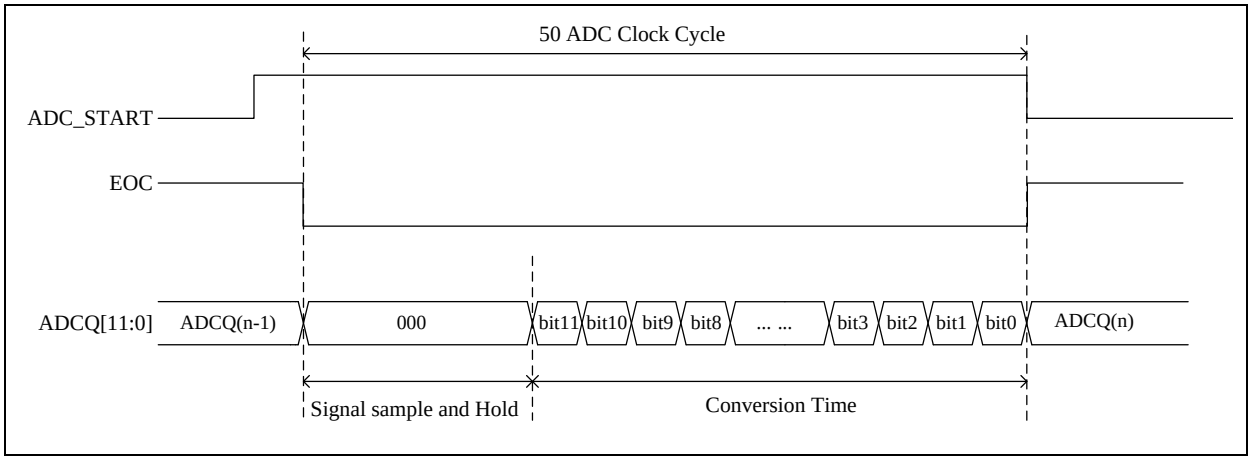




8. Analog to Digital convert

The 12-bit ADC (Analog to Digital convert) consists of a 24-channel analog input multiplexer, control register, clock generator, 12-bit successive approximation register and output data register. To use the ADC, user needs to set R14 [2:0] (ADCLSEL) to choose a proper clock frequency. User launches the ADC conversion by setting F0B[5] (ADC_START). After end of conversion, H/W automatic clears the ADC_START bit. User can poll this bit to know the conversion status. PADI_EN, PCDI_EN and PDDI_EN control registers are used for ADC pin type setting, user can write the corresponding bit to "0" when the pin is used as an ADC input. The setting can disable the pin logical input path to save the power consumption. Conversion data is stored at R0C(ADCQ[7:0] and R08(ADCQ[11:8])





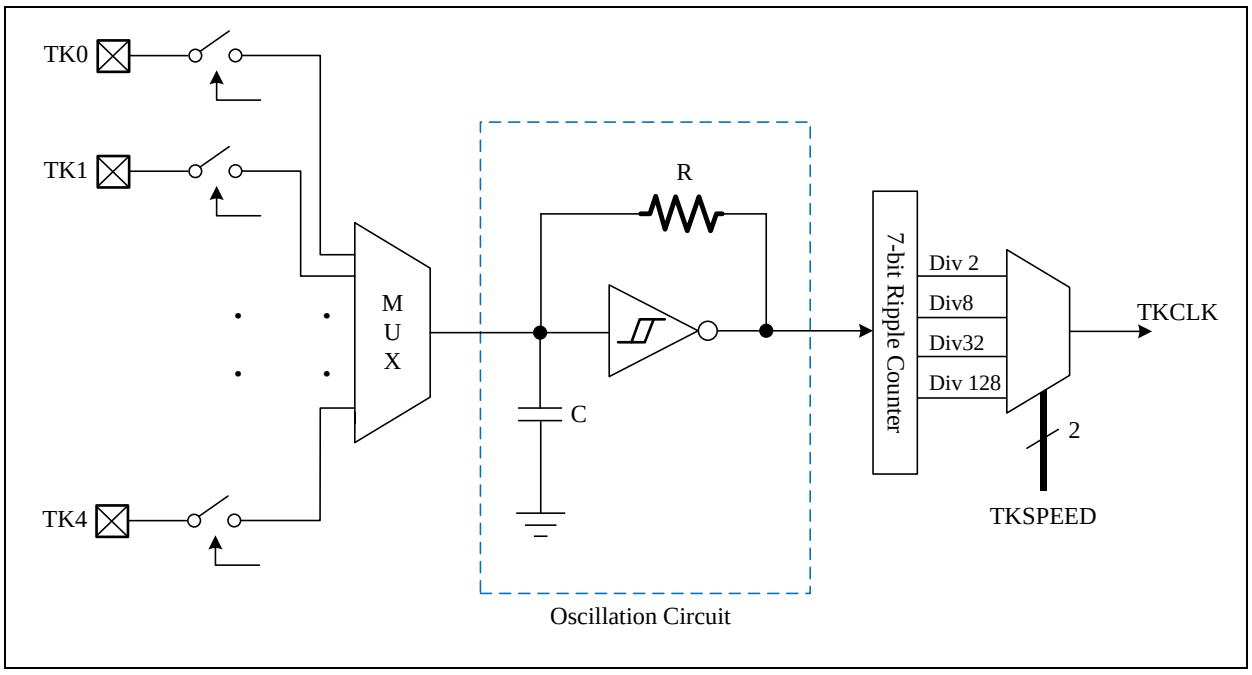
9. Touch Key

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 an RC oscillator, 5-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.

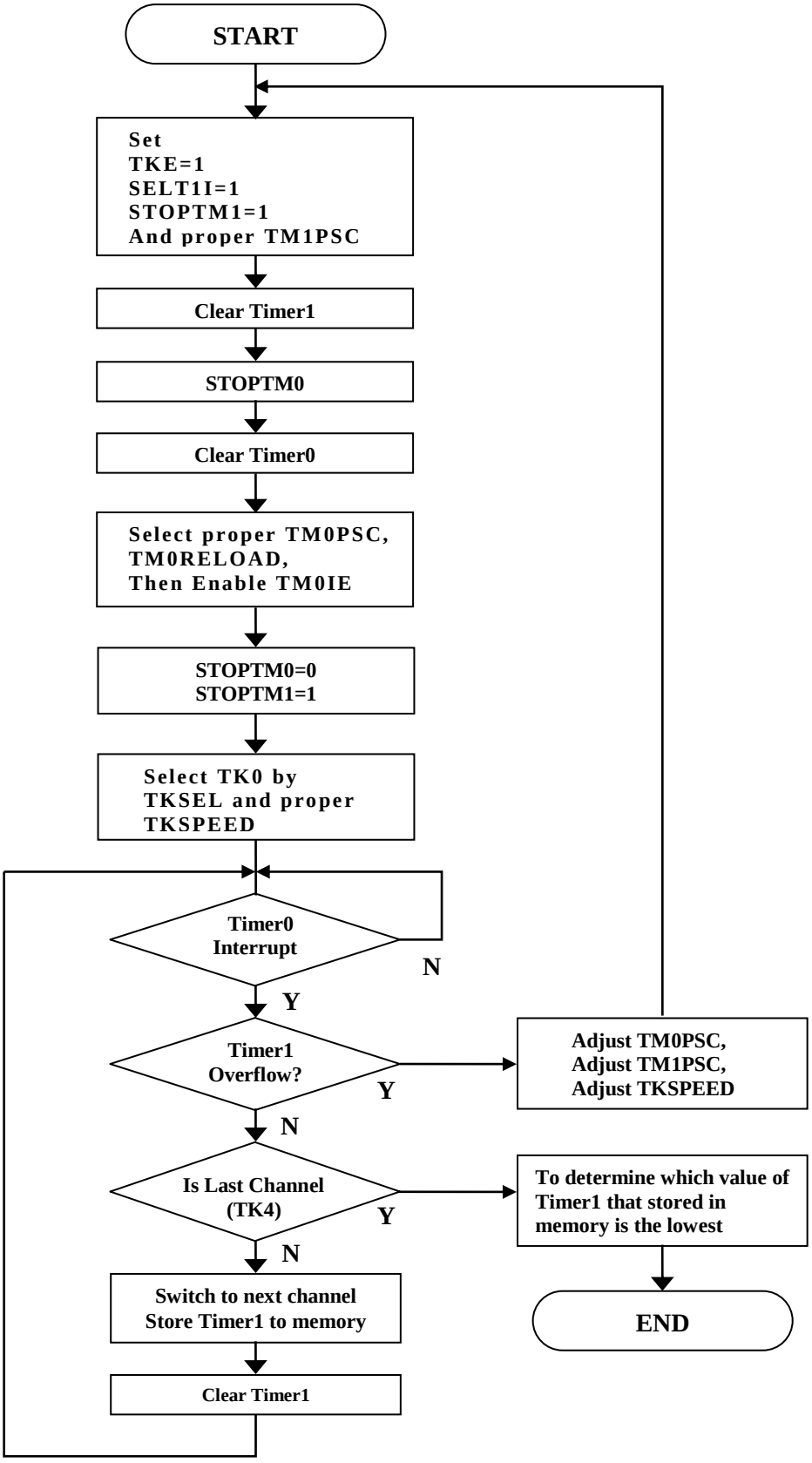
If the finger tips close to the touch pad, the equivalent capacitance of C will be increased, that is, the oscillation frequency will be decreased.

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. Base on this fixed time interval, the user program switches the Touch Key channels one after another and find the lowest value of Timer1, which is the key in touching or approaching.



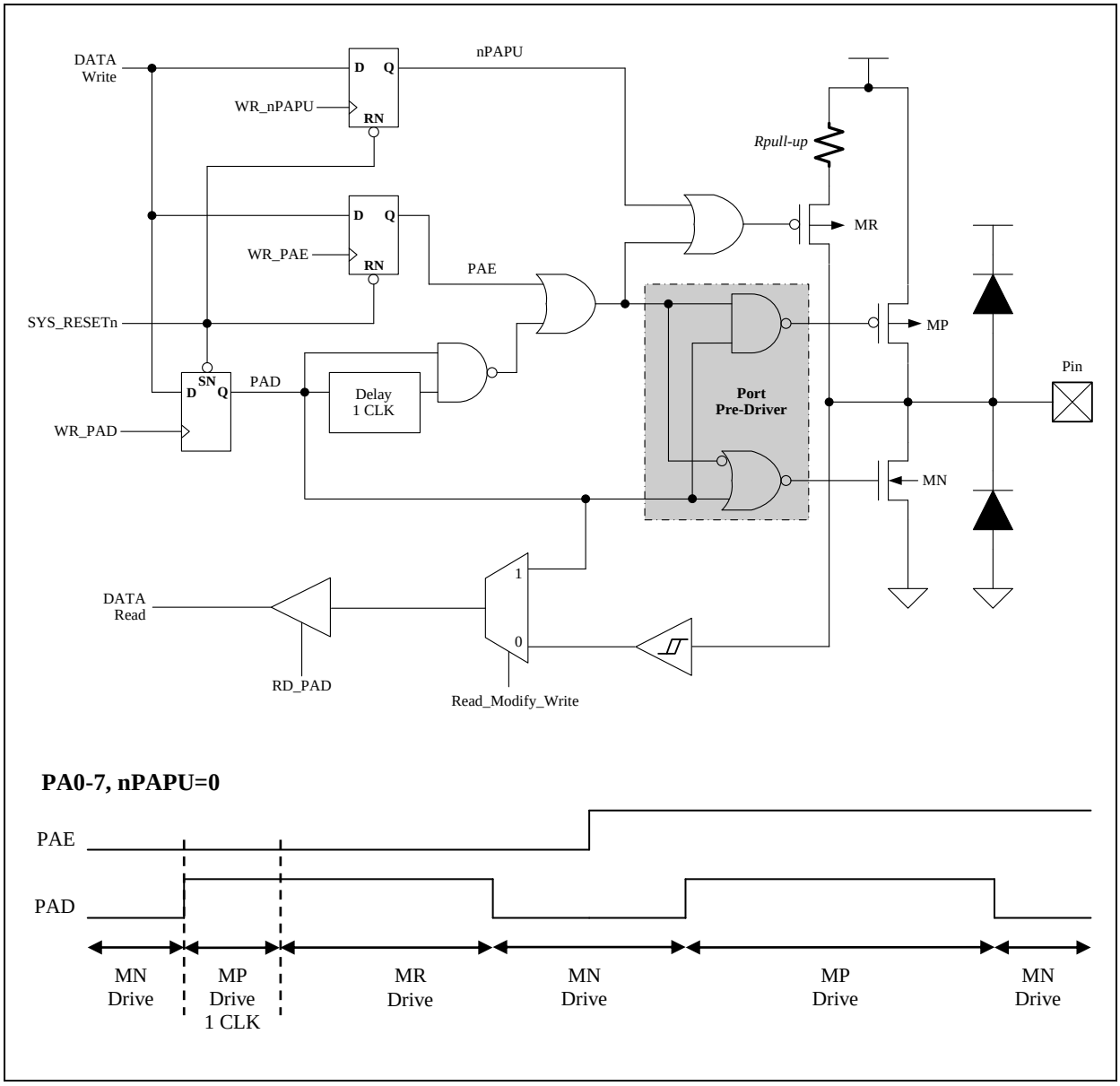
The flowchart described below shows how to use Timer0 and Timer1 to determine what channel of the Touch Key is pressed. Using the 8-bit Timer0 to set up a fix interval of time and utilize the Timer0 interrupt to stop Timer1 and store its value if it is not overflow. Determine the lowest value of Timer1 of the desired channels, which the key is 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
STOPTM1=1
Select TK0 by
TKSEL and proper
TKSPEED
Timer0
Interrupt
N
Y
Timer1
Overflow?
Y
Adjust TM0PSC,
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

10. I/O Port

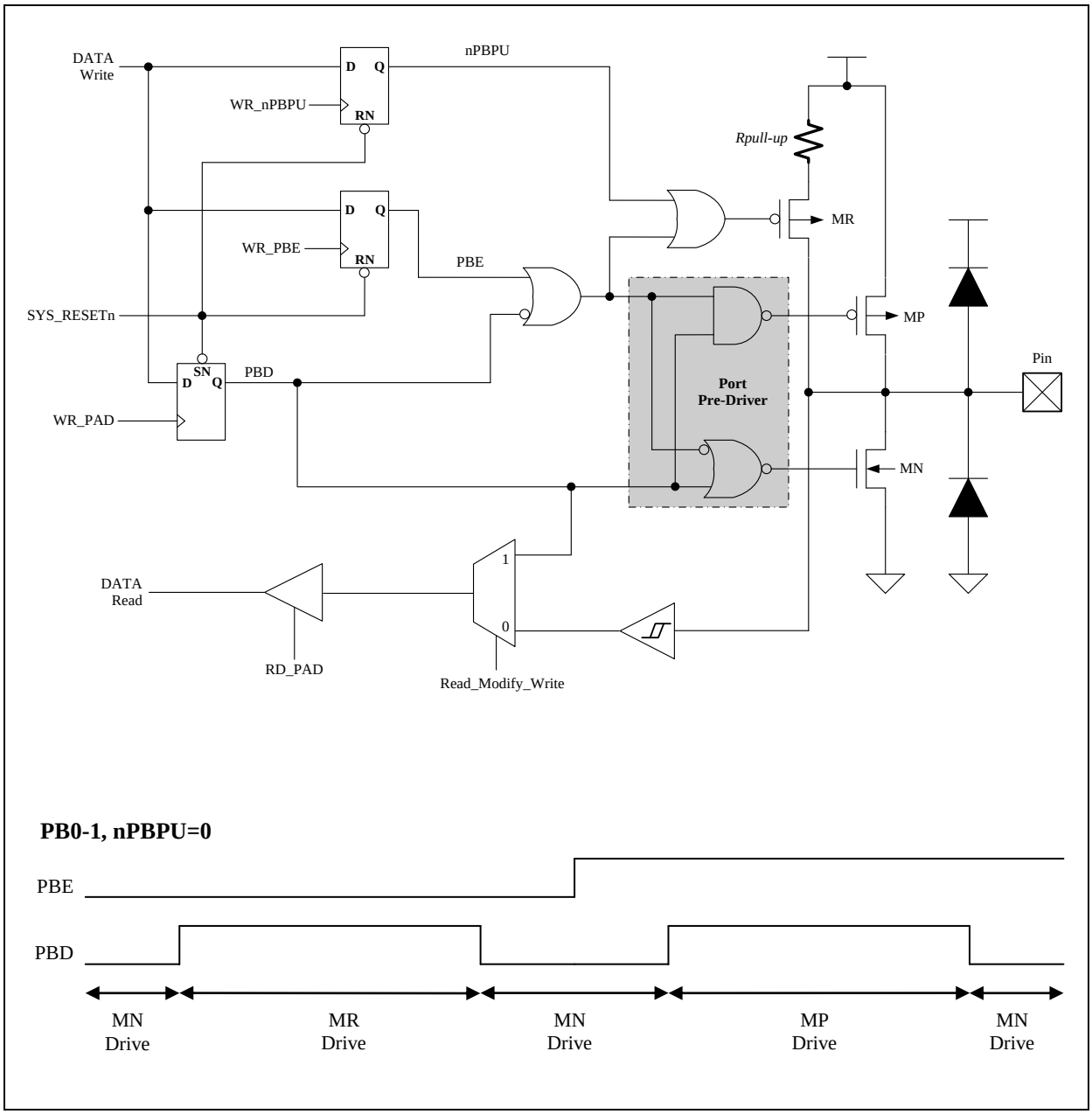
10.1 PA0-7

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



10.2 PB0-1, PB4-5

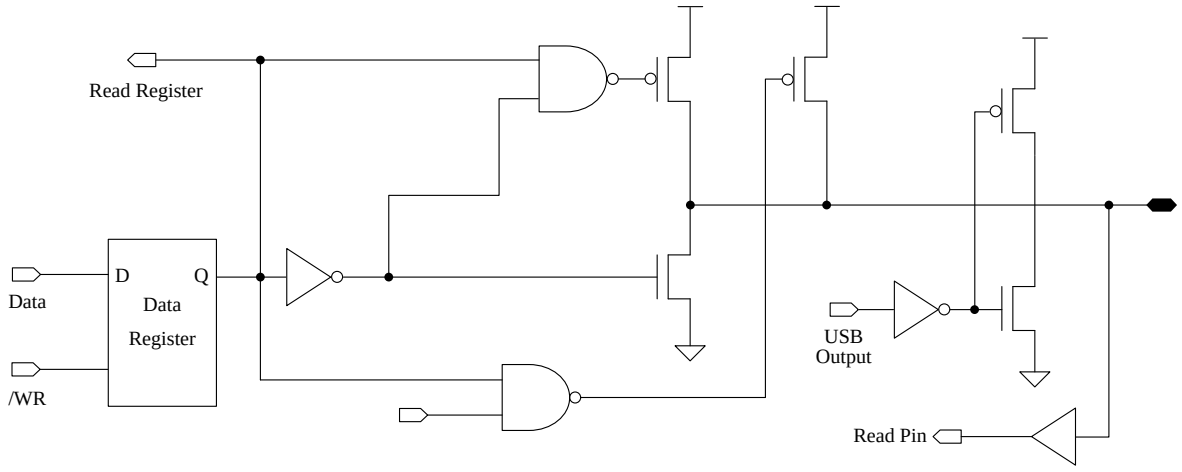
These four pins are almost the same as PA0-7, except they do not support pseudo-open-drain mode. They can be used in pure open-drain mode, instead.



PB0-1, nPBPU=0

10.3 PB3(DP) and PB2(DM)

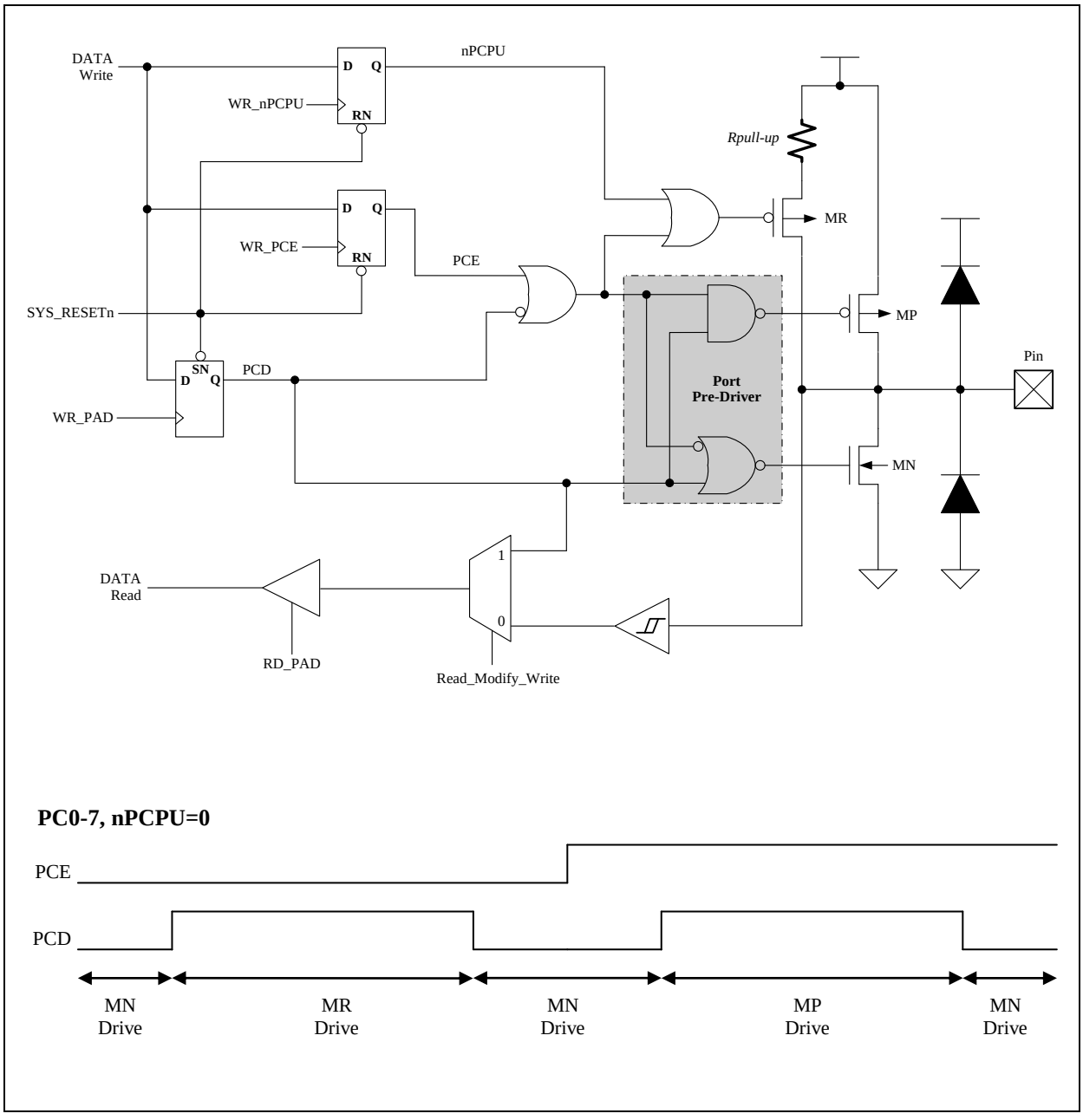
These pins are similar to PB[1:0], except they share the pin with USB function.



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

10.4 PC0-7

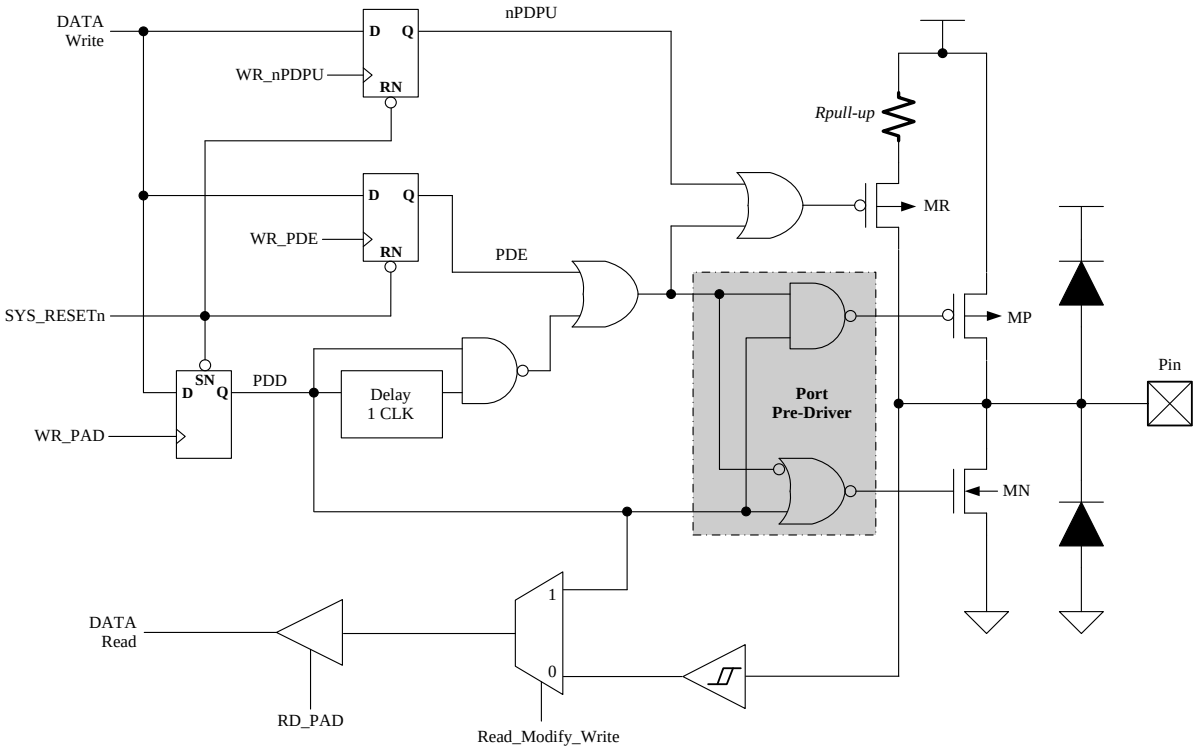
PortC can be used as Schmitt-trigger input, CMOS push-pull output or "open-drain" output. There is only one pull-up enable bit setting by S/W to control all PortC pins. To use the pin in Schmitt-trigger input mode, S/W needs to set the PCE=0 and PCD=1. To use the pin in pseudo-open-drain mode, S/W sets the PCE=0. S/W sets PCE=1 to use the pin in CMOS push-pull output mode. Reading the pin data (PCD) has different meaning. In "Read-Modify-Write" instruction, CPU actually reads the output data register. In the others instructions, CPU reads the pin state. The so-called "Read-Modify-Write" instruction includes BSF, BCF and all instructions using F-Plane as destination



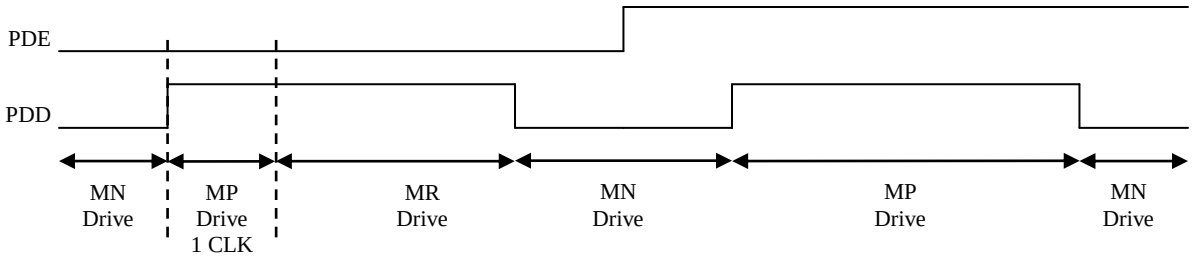
PC0-7, nPCPU=0

10.5 PD0-7

PortD 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 nPDPU is used to control PortD.



PD0-7, nPDPU=0

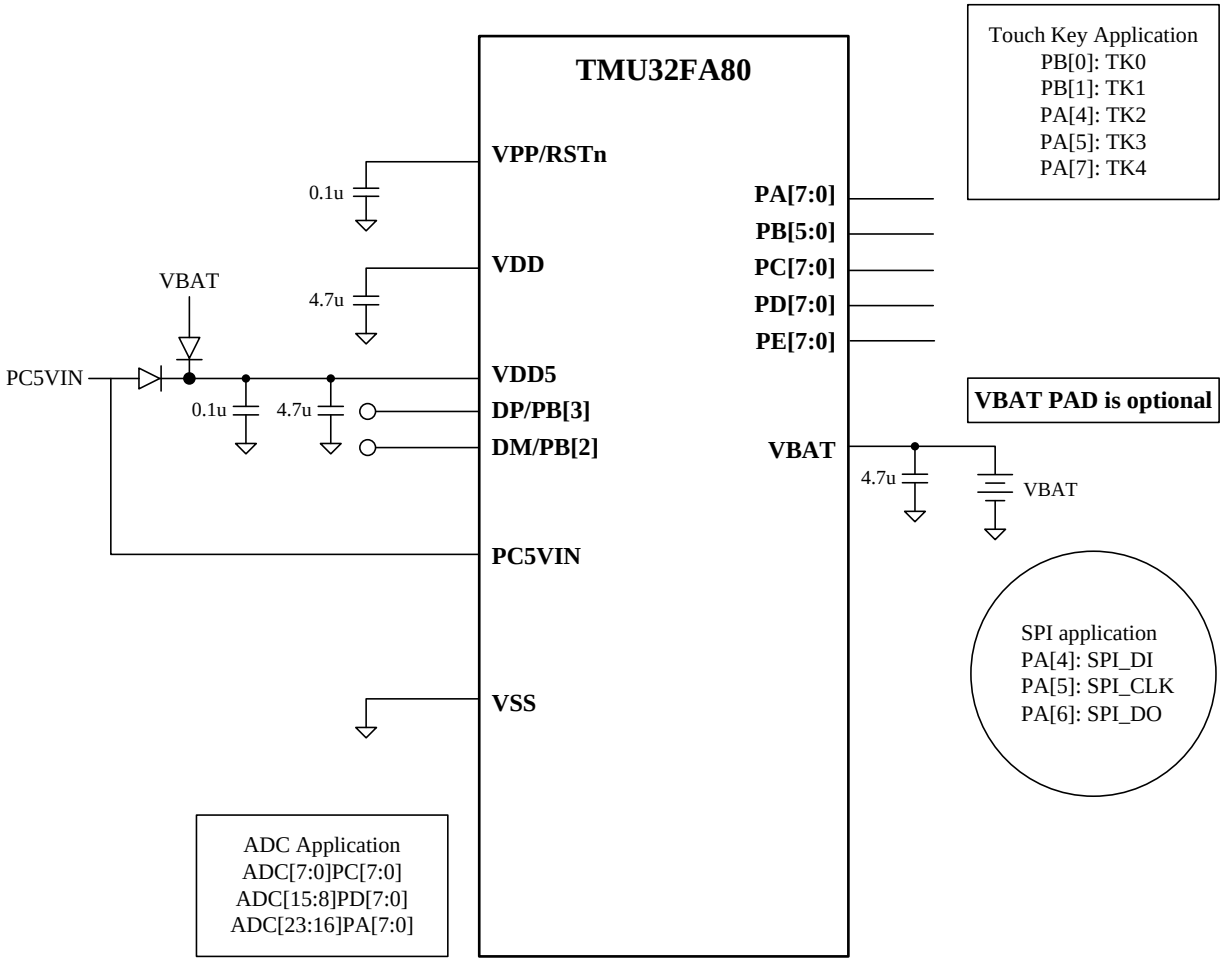


10.6 PE0-7

PortE pins are the same as PD0-7. PE[0] can output PWM by setting Register R30, R31 and R32. 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 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

Application



Electrical Characteristics

1. ABSOLUTE MAXIMUM RATINGS

GND= 0V

Name	Symbol	Range	Unit
Maximum Supply Voltage	VDD5	-0.3 to 5.5	V
Chip Operating Voltage	VDD	2.0 to 3.6	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	℃

2. RECOMMENDED OPERATING CONDITION

at Ta=-20℃ to 70℃,GND= 0V

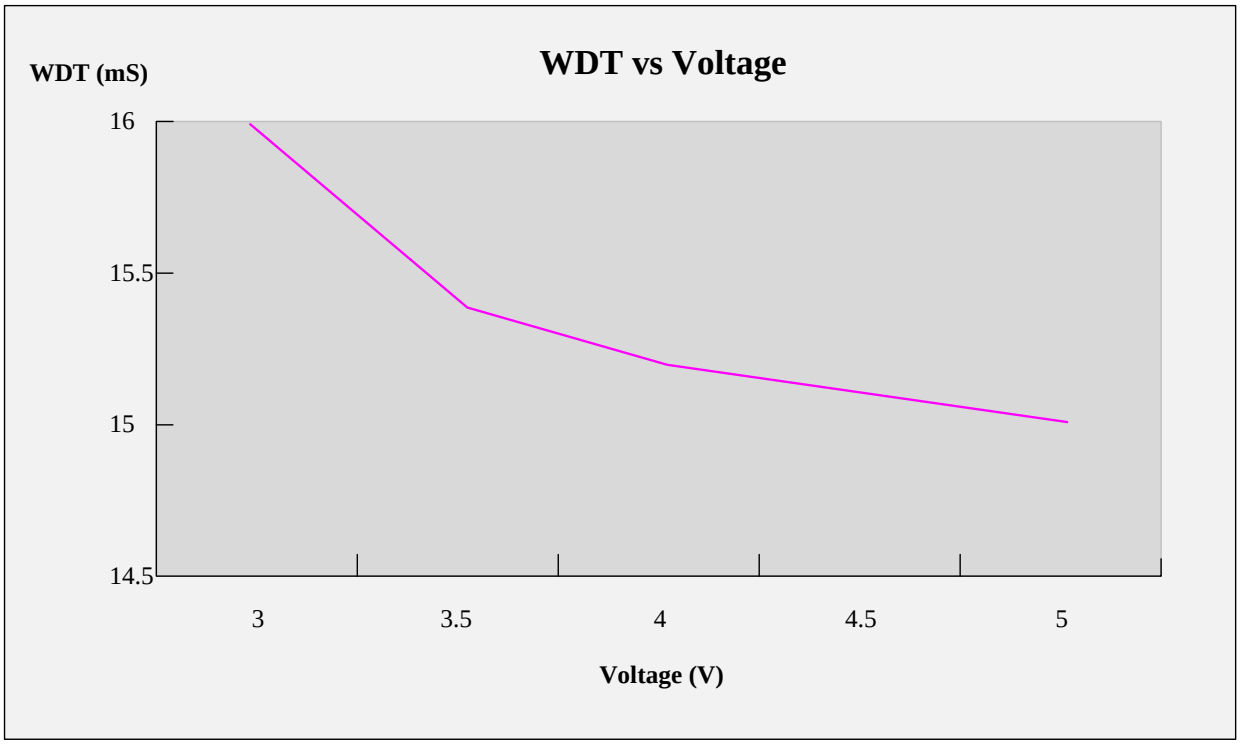
Name	Symb.	Min.	Typical	Max.	Unit	Condition
Supply Voltage	VDD5	2.3		5.5	V	
Battery Voltage, if apply	Vbat	2.2		3.6	V	
VDD output voltage	VDD		3.3		V	VDD5=5V Vbat=0V
				2.96	V	VDD5=3V Vbat=0V
				3.2	V	Vbat=3.6V, VDD5=0V
				2.93	V	Vbat=3V, VDD5=0V
Input “H” Voltage	Vih	0.8VDD			V	
Input “L” Voltage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

3. DC CHARACTERISTICS

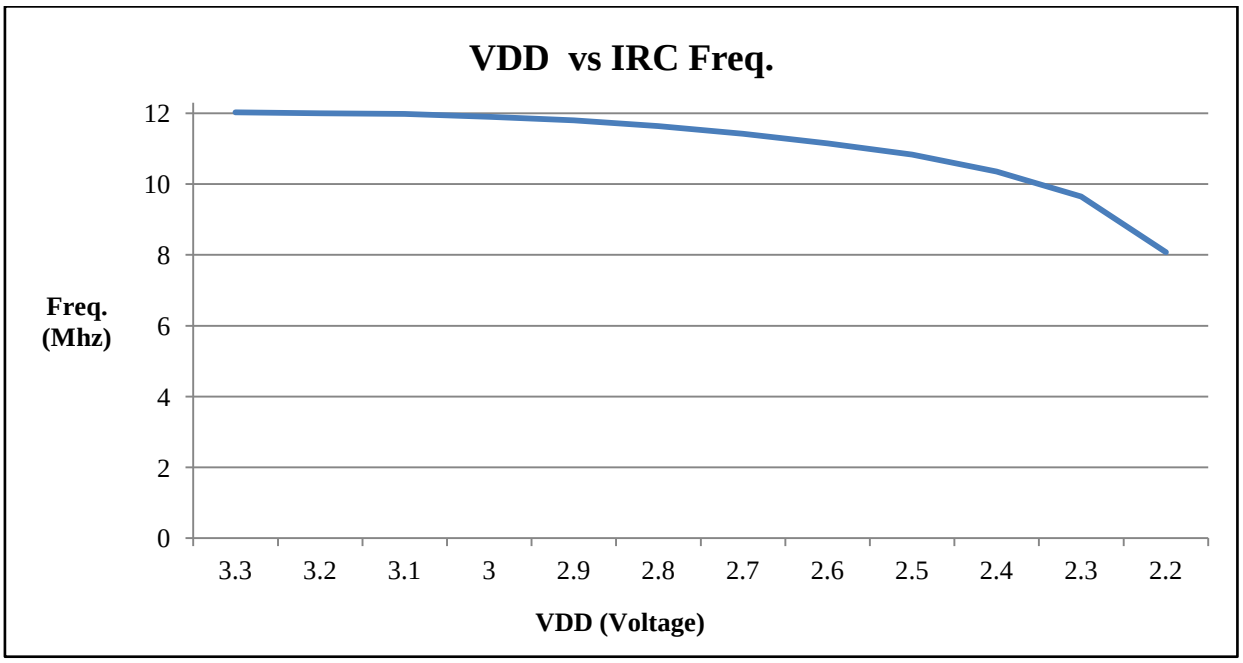
At Ta=-25 ℃, VDD5=5.0V, VSS= 0V

Name	Symb.	Min.	Typ.	Max.	Unit	Condition
Internal Clock	Firc		48		MHz	Enable FIRC, VDD5=5V
Operating current	Icc		6.5		mA	CPU clock=12 MHz
Power Down current	Ipd			1	uA	No load
Suspend Current	Isus		350	500	uA	USB Mode, No load
Output High Current (Push Pull Mode)	Ioh1		11		mA	VDD5=5V, Voh1=2.8V
	Ioh2		10		mA	VDD5=3V, Voh2=2.3V
Output High Current (Pseudo Open Drain Mode)	Ioh3		11		uA	VDD5=5V, Voh3=2.8V
	Ioh4		13		uA	VDD5=3V, Voh4=2.3V
Output Low Current (Push Pull Mode)	Iol1		17		mA	VDD5=5V, Vol1=0.3V
	Iol2		15		mA	VDD5=3V, Vol2=0.3V
Output Low Current (Pseudo Open Drain Mode)	Iol3		16		mA	VDD5=5V, Vol3=0.3V
	Iol4		15		mA	VDD5=3V, Vol4=0.3V
Input Leakage Current (pin high)	Iilh			1	uA	Vin=VDD
Name	**Symb.**	**Min.**	**Typ.**	**Max.**	**Unit**	**Condition**
Input Leakage Current (pin low)	Iill			-1	uA	Vin=0V
Pull-Up Resistor	$R_{pull-up}$		118		KΩ	VDD5=5V
			140		KΩ	VDD5=3V
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=1.5 MHz
WDT time	Twdt		15		ms	VDD5=5V, WRC enable R06[6:5]=2'b00
			30		ms	VDD5=5V, WRC enable R06[6:5]=2'b01
			60		ms	VDD5=5V, WRC enable R06[6:5]=2'b10
			120		ms	VDD5=5V, WRC enable R06[6:5]=2'b11
WKT Time			120		ms	VDD5=5V, WRC enable R06[4:3]=2'b00
			240		ms	VDD5=5V, WRC enable R06[4:3]=2'b01
			480		ms	VDD5=5V, WRC enable R06[4:3]=2'b10
			960		ms	VDD5=5V, WRC enable R06[4:3]=2'b11

WDT vs. VDD5 voltage (R06[6:5]=2'b00)



VDD Voltage vs. FIRC Frequency



4. USB AC CHARACTERISTICS

At Ta=25 ℃, VDD5V=5.0V, VSS= 0V

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	-
VDD output voltage	VDD	3.2	3.3	3.4	V	-

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

Ordering Information

The ordering information:

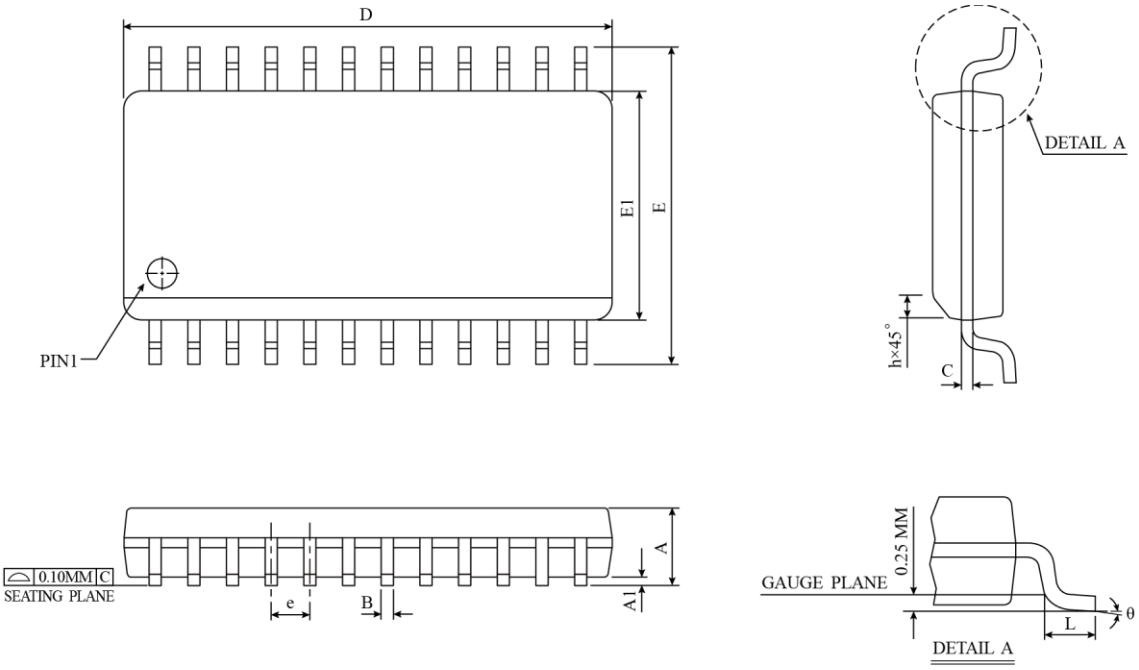
Ordering number	Package
TMU32FA80-MTP	Wafer / Dice blank chip
TMU32FA80-COD	Wafer / Dice with code
TMU32FA80-MTP-22	SOP 24-pin (300 mil)
TMU32FA80-COD-22-X	SOP 24-pin (300 mil)
TMU32FA80-MTP-74	LQFP 44 pin (10*10*1.4mm)
TMU32FA80-COD-74-X	LQFP 44 pin (10*10*1.4mm)

Note: “-X” represents the package material:

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

Package Information

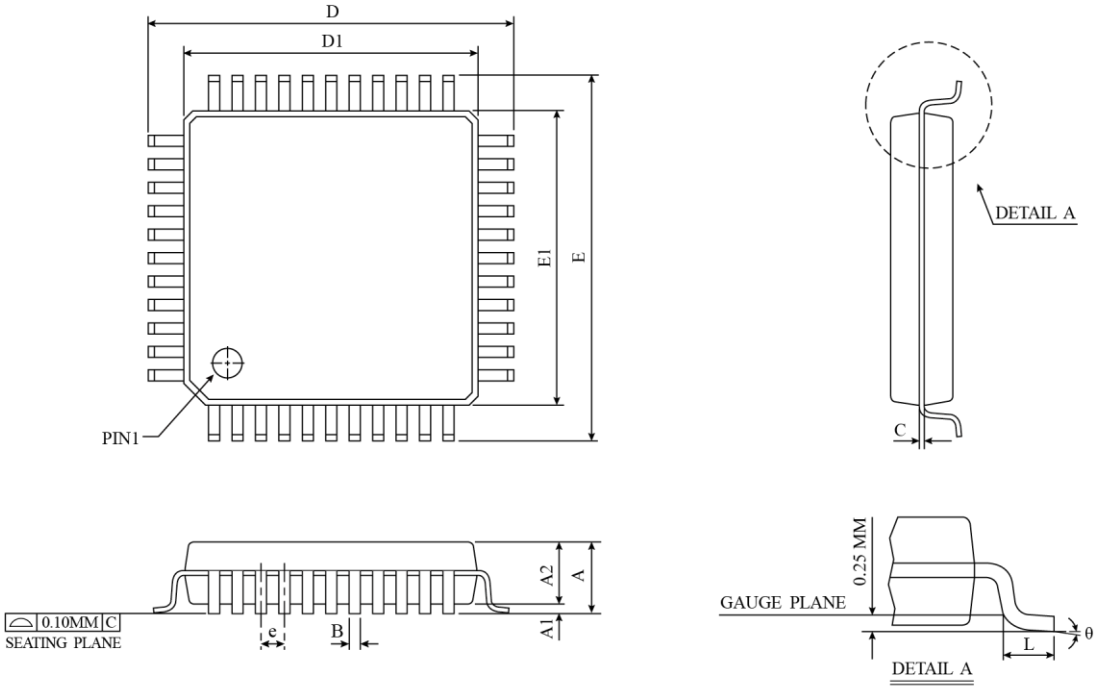
● SOP-24 (300mil) Package Dimension



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
D	15.20	15.60	0.5985	0.6141
E	10.00	10.65	0.394	0.491
E1	7.40	7.60	0.2914	0.2992
e	1.27 BSC		0.050 BSC	
h	0.25	0.75	0.010	0.029
L	0.40	1.27	0.016	0.050
θ	0°	8°	0°	8°
JEDEC	MS-013 (AD)			

△ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM (0.006 INCH) PER SIDE.

● QFP-44 (10×10mm) Package Dimension



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	1.95	2.30	0.077	0.091
A1	0.05	0.20	0.002	0.008
A2	1.90	2.10	0.075	0.083
B	0.30	0.38	0.012	0.015
C	0.11	0.23	0.004	0.009
D	12.9	13.5	0.508	0.531
D1	9.90	10.1	0.390	0.398
E	12.9	13.5	0.508	0.531
E1	9.90	10.1	0.390	0.398
e	0.67	0.93	0.026	0.037
L	0.60	1.00	0.024	0.039
θ	0°	7°	0°	7°
JEDEC				

⚠ * NOTES : BOTH PACKAGE LENGTH AND WIDTH DO NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS, PROTRUSIONS OR GATE BURRS SHALL NOT EXCEED 0.15 mm PER END.