十速科技股份有限公司
tenx technology inc.

Advance Information

TMU3130

USB Full Speed Controller

Data Sheet

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

tenx technology inc.

CONTENTS

GENERAL DESCRIPTION

The TMU3130 is an 8-bit microprocessor embedded device tailored to the USB full speed general purpose application. It includes an 8-bit RISC CPU core, one 160-byte SRAM, two 64-byte SRAM and an 8K x 14 internal Flash ROM. It can work in USB mode by using USB power or work in standalone mode by using battery power. TMU3130 can use external crystal or use Internal RC as clock source.

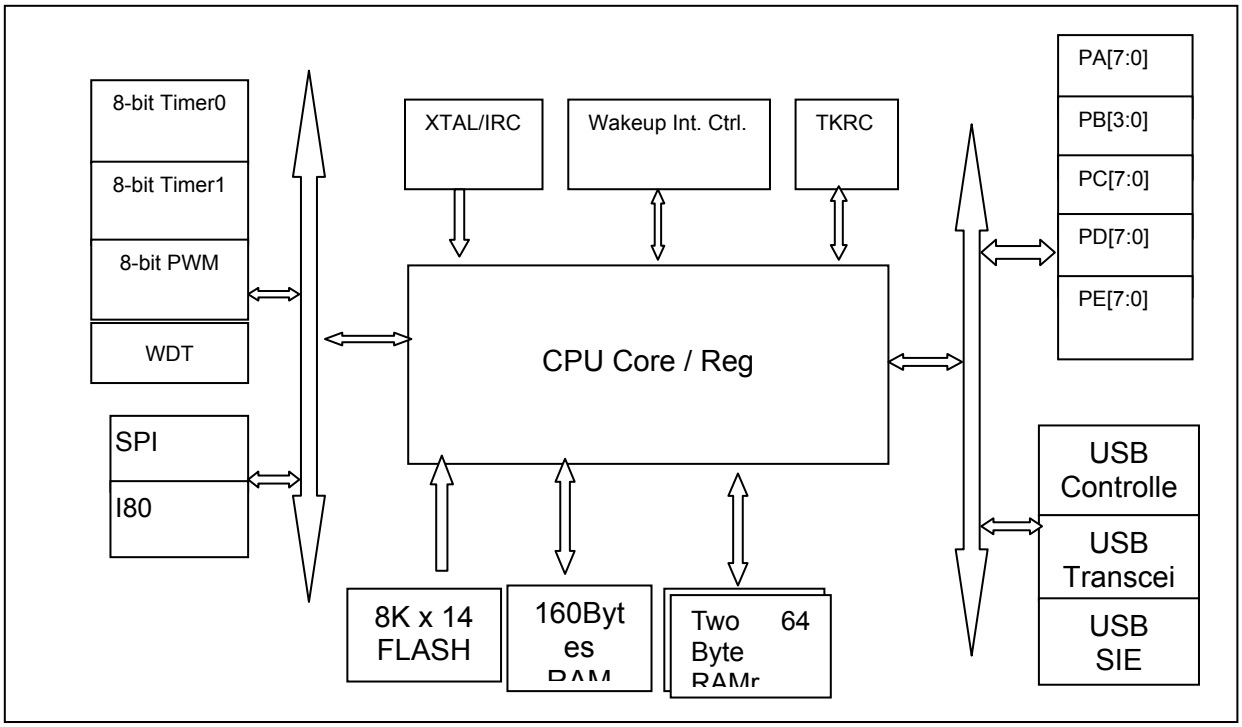
FEATURES

- **Operation Frequency**
  - 6MHz crystal oscillation with internal 48MHz PLL.
  - Internal +/- 3% 48Mhz RC oscillator
    0.25% Accuracy for USB without extra component
  - STOP mode

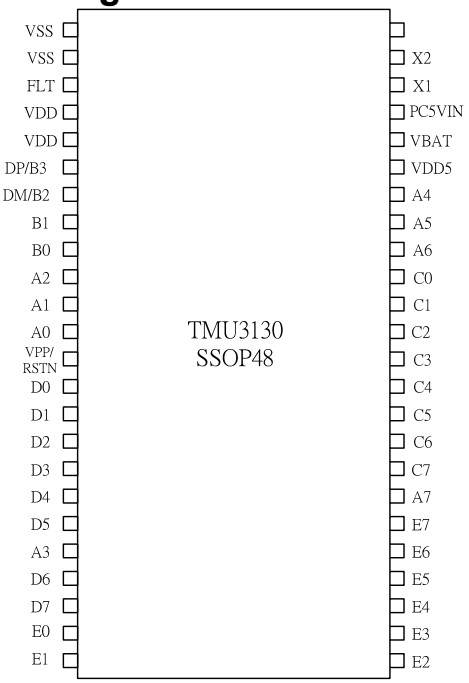
- **On-Chip Memory**
  - 8k x 14 internal program FLASH
  - Internal 160 Bytes RAM at F-plane and two 64Bytes RAM at R-plane
- **USB interface**
  - Compliance with the Universal Serial Bus specification v2.0 Full Speed
  - Built-in USB Transceiver, 3.3V regulator.
  - Support USB Suspend /Resume and Remote Wakeup function
  - Endpoint 0: Control SETUP/IN/OUT transfer (each 8 bytes)
  - Endpoint 1: INTERRUPT IN transfer (8 bytes)
  - Endpoint 2: INTERRUPT IN transfer (8 bytes)
  - Endpoint 3: BULK-IN transfer with Ping-Pong feature (2*64 bytes)
  - Endpoint 4: BULK-OUT transfer with Ping-Pong feature (2*64 bytes)
- **Timer**
  - Timer0 is 8-bit with 8-bit prescaler, Counter/Reload/Interrupt function
  - Timer1 is 8-bit counter.
  - Watchdog Timer clocked by built-in RC oscillator
  - Wakeup Timer clocked by built-in RC oscillator
- **PWM:**
  - Support Pulse Width Modulation(PWM) function with 8-bit resolution.
- **Reset Controller**
  - Power On Reset, Watch-Dog Timer, USB Plug-out Reset
- **SPI interface**
  - Support Mode0, 1, 2, 3
  - Master only
  - Clock rate up to 6Mbps.
  - Read/Write DMA mode

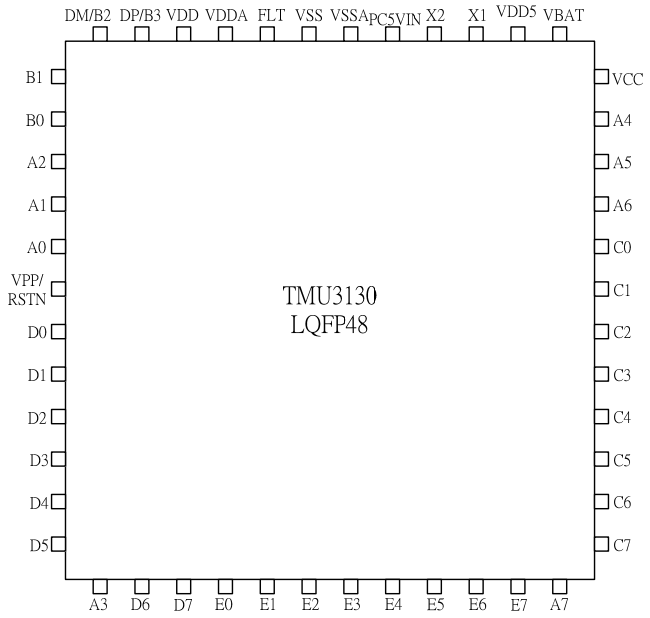
- **I80 Interface (Nand-Flash Interface)**
  - Compatible with 8-bit parallel interface
  - Read/Write DMA mode
- **Touch Key**
  - 5 channel touch pad
- **I/O Ports**
  - Max. 36 GPIOs to flexible application
- **LQFP48/ Die Form**
- **Application:**
  - USB full speed general purpose
  - USB Keyboard
  - Battery mode 8-bit RISC general purpose

Functional Block Diagram



Package





PIN DESCRIPTION

Name	I/O	Description
VDD5	P	5V Power
VSS	P	Ground
PC5VIN		5V Power from USB cable
VBAT	P	Battery power(Optional)
VDD	O	3.3V regulator output
X1	I	Crystal in (6MHz)
X2	O	Crystal out
VPP/RSTn	I	Programming High power/Chip reset pin
FLT	O	PLL Filter Pin(Optional)
PA[7]	I/O	General purpose I/O (Pseudo open-drain) ; I80RD
PA[6]/SDO	I/O	General purpose I/O (Pseudo open-drain) ; SPI Dout
PA[5]/SCLK	I/O	General purpose I/O (Pseudo open-drain) ; SPI Clk
PA[4]/SDI	I/O	General purpose I/O (Pseudo open-drain) ; SPI Din
PA[3]	I/O	General purpose I/O (Pseudo open-drain); I80WR
PA[2]	I/O	General purpose I/O (Pseudo open-drain) ; TK[2]
PA[1]/	I/O	General purpose I/O (Pseudo open-drain); TK[3]
PA[0]	I/O	General purpose I/O (Pseudo open-drain); TK[4]
DP/PB[3]	I/O	USB positive signal / General purpose I/O (Pseudo open-drain);IIC SCK
DM/PB[2]	I/O	USB negtive signal / General purpose I/O (Pseudo open-drain); IIC DAT
PB[1:0]	I/O	General purpose I/O (open-drain) ; TK[1:0]
PC[7:0]	I/O	General purpose I/O (open-drain) ; KSI[7:0]; I80 DATA[7:0]
PD[7:0]	I/O	General purpose I/O (Pseudo open-drain) ; KSO[7:0]
PE[7:0]	I/O	General purpose I/O (Pseudo open-drain) ; KSO[15:8]

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

Functional Description

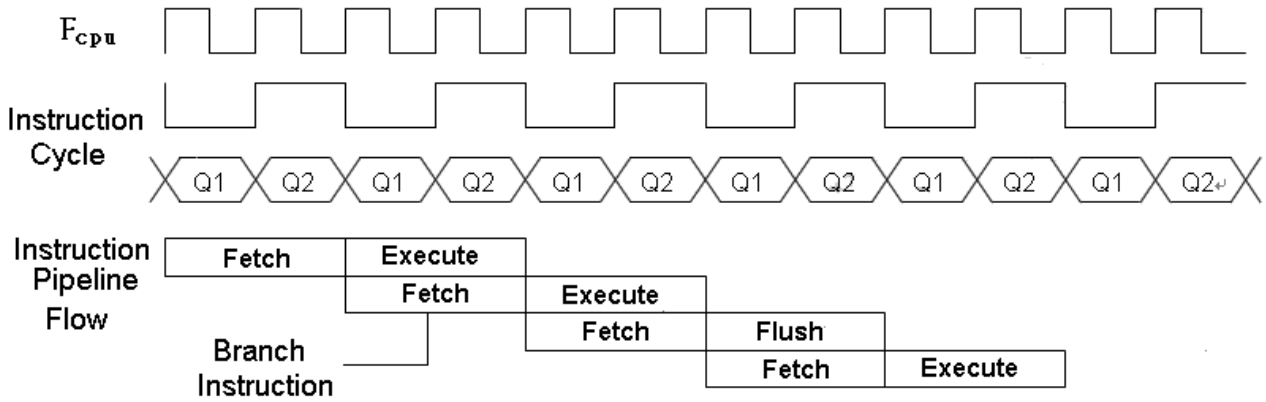
1. CPU Core

1.1 Clock Scheme and Instruction Cycle

TMU3130 has 2 chip clock sources as following:
F_{xtal_6m} : External Crystal Oscillator clock
F_{rc} : Internal RC oscillator 24Mhz clock

F_{xtal_6m} is popup to 48Mhz clock(F_{pll}) by PLL. The F_{pll} clock will be used for USB and for CPU clock.
F_{rc} can be synchronized by USB signals and popup to 48Mhz clock for USB module. F_{rc} can also be used for CPU clock. Clock Control register R07[4:3] is used to enable external 6Mhz crystal oscillator or internal oscillator.
R07 [4] = 1, enable external 6Mhz crystal oscillator
R07 [3] = 1, enable internal RC oscillator

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: TMU3130 can select external 6Mhz crystal oscillator or select internal RC oscillator as the CPU clock source.
  R07 [2] = 1, select external 6Mhz crystal oscillator clock
  R07 [2] = 0, select internal RC oscillator
- CPU clock speed selection: No matter which clock source is selected, the clock source will 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 12Mhz
  R07 [1:0] =1 select 6Mhz
  R07 [1:0] =2 select 3Mhz
  R07 [1:0] =3 select 1.5Mhz

1.3 Programming Counter (PC) and Stack

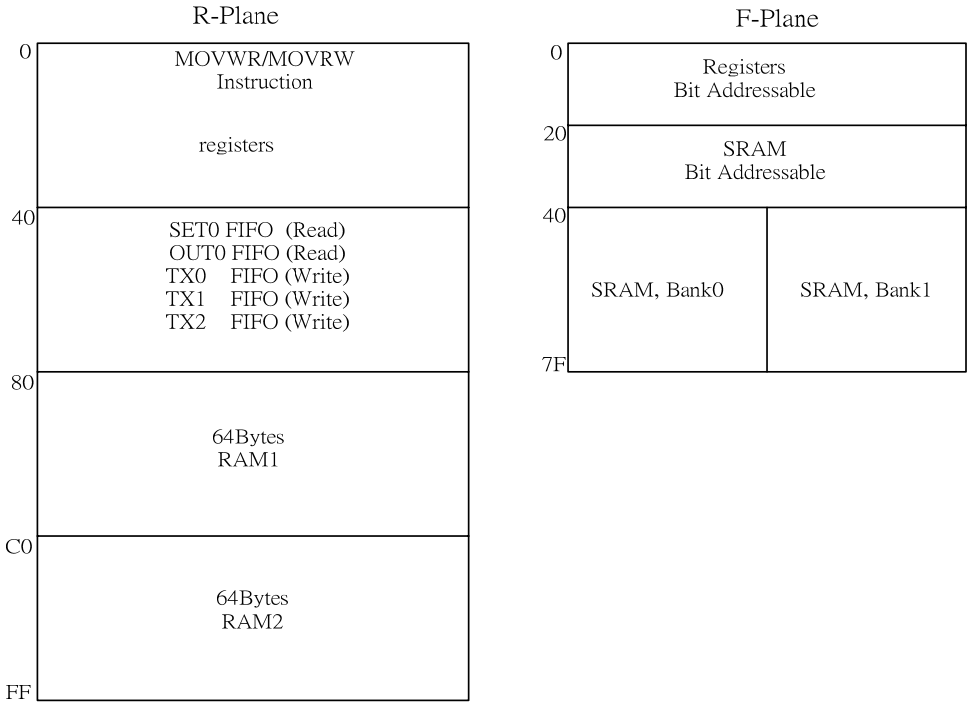
The Programming Counter is 13bit wide capable of addressing a 8K x 14 program ROM. As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC value is normally increased by one except the followings. The Reset Vector (000h) and the Interrupt 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/clear according to page0 or page1 will the program counter be. Remember that ISR entry addresses are locate at ROM Page0, if the user code is interrupt 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.

1. 8K x 14 program FLASH.
2. 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 separate to two bank which can be select by register F03[5]
3. Two 64-byte RAM(R-plane)
4. Five 8-byte USB FIFO are allocated in R-plane



2. Control Registers

F-plane

	Name	Address	R/W	Rst	Description
F01	TIMER0	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	Timer1	0D.0~7	R/W	0	Timer1
F0E	TM1IE	0E.0	R/W	0	Timer1 Interrupt Enable
F0F	TM1I	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	SET00I	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.
	KBDI	12.2	R/W	0	KeyBoard Interrupt flag, write 0 to clear flag.(Port C)
	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	SUSPND	13.7	R/W	0	S/W force USB interface into suspend mode.
	RSMO	13.6	R/W	0	S/W force USB interface send RESUME signal in suspend mode.
	EP1CFG	13.5	R/W	0	Set Endpoint 1 configed.
	EP2CFG	13.4	R/W	0	Set Endpoint 2 configed.
	Device_R	13.3	R/W	0	DP Pullup resistor enable bit, 0: Disable pullup , 1: pullup enable
	OUT0RDY	13.0	R/W	0	Endpoint 0 ready for receive, clear by H/W while OUT0I occurs.
F14	TX0RDY	14.7	R/W	0	Endpoint 0 ready for transmit, clear 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)
	TX0CNT	14.3~0	R/W	0	Endpoint 0 transmit byte count.
F15	TX1RDY	15.7	R/W	0	Endpoint 1 ready for transmit, clear 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.
	TX1CNT	15.3~0	R/W	0	Endpoint 1 transmit byte count.

F16	TX2RDY	16.7	R/W	0	Endpoint 2 ready for transmit, clear by H/W while TX2I occurs.
	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, clear 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.
F18	RC4RDY	18.7	R/W	0	Endpoint 4 ready for receive, clear by H/W while RC4I occurs.
	RC4TGL	18.6	R	-	Endpoint 4 received DATA1/DATA0 packe
	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.
F1B	I8080CON	1B.3~0	R/W	0	I8080 Configuration
	8080_Busy	1B.3	R	0	1: I8080 is in Busy state, 0: idle
	8080_EN	1B.2	R/W	0	Enable I8080 DMA Interface
	8080_Start	1B.1	R/W	0	I8080 I/F Start RX/TX
	8080_DIR	1B.0	R/W	0	0: I8080 write data to Device, 1: I8080 Read data from Device
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
	SRAM18080	1C.1	R/W	0	Assign SRAM1 as 8080 DMA Transfer buffer
	SRAM28080	1C.0	R/W	0	Assign SRAM2 as 8080 DMA Transfer buffer
F1D	SPI_MODE	1D.5	R/W	0	SPI MODE
	SPI_EN	1D.4	R/W	0	SPI Enable, Busy bit
	LSB_First	1D.3	R/W	0	1:Data transmit/Receive is LSB first; 0: MSB first
	SPI_IN	1D.2	R/W	0	(1)SPI Bus is use to receive data from SPI Device
					(0)SPI Bus is use to transmit data to SPI Device
	SPI_cmd_sw	1D.1	R/W	0	SPI CMD/DAT Switch; 1:CMD, 0:DAT
	clr_ram_adr	1D.0	R/W	0	Write 0 to clear ram address
	SRAM	20~7F	R/W	-	Internal RAM (96 Bytes x 2 Banks)

R-plane

	Name	Address	R/W	Rst	Description
R01	T0RLD	01.7~0	W	0	Timer0 overflow reload value

R02	T0en	02.4	W	0	Timer0 Enable
	T0PSCL	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
R05	KBDMASK	05.7~0	W	0	mask KSI[7:0] interrupt function while the corresponding bit is "1"
R06	WRC_PD	06.7	W	0	WRC Disable, 1: Disable WRC, 0: Enable WRC
	WDTPSC	06.6~5	W	11	WDT period, 00=20mS, 01=40mS, 10=60mS, 11=80mS
	WKTPSC	06.4~3	W	11	WKT period, 00=160mS, 01=320mS, 10=640mS, 11=1280mS
	VDD5VFLG	06.2	R	0	PC5V status, 1:PC5V High 0:PC5V LOW (USB plug-in flag)
	VDD_FALL_FLG	06.1	R/W	0	USB plug-out flag, write 0 or RST_P=1 to clear flag
R07	HW_AUTO	07.5	W	0	H/W auto push/pop during INT process
	FCLKEN	07.4	W	1	Clock XTAL Enable(1)
	SCLKEN	07.3	W	1	Clock IRC Enable(1)
	CLK_SEL	07.2	W	0	SysClk Source Select, 0: IRC CKO, 1: PLL CKO
	CLKDIV	07.1~0	W	0	System Clk Period Selection 2'b00: 12Mhz 2'b01: 6Mhz 2'b10: 3Mhz 2'b11: 1.5Mhz
R0A	TM1EN	0A.5	W	0	Timer1 enable
	SELT1I	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
	TKSPEED	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
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
	KBDIE	12.2	W	0	KeyBoard Interrupt enable

	PB0IE	12.1	W	0	PB0 Interrupt enable
	TM0IE	12.0	W	0	Timer0 Interrupt enable
R13	RC0TGL	13.7	R		1: received DATA1 packet; 0: received DATA0 Packet.
	RC0ERR	13.6	R		Endpoint 0 received data error.
	EP0DIR	13.5	R		1: IN transfer; 0: OUT/SETUP transfer.
	EP0SET	13.4	R		SETUP Token indicator.
	OUT0CNT	13.3~0	R		OUT0 Received data byte count.
R20	PAE	20.7~0	W	0	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	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	PCDEPU	27.2~0	W	0	Port C/D/Epull-up, 0=enable
		27.2	W	0	Port C pull-up, 0=enable
		27.1	W	0	Port D pull-up, 0=enable
		27.0	W	0	Port E pull-up, 0=enable
R30	PWMEN	30.0	W	0	PWM Enable bit
	PWMPSC[1:0]	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
R3A	I80_LEN	3A.6~0	W	0	I80 DMA transfer length
R3B	CPOL	3B.5	W	0	SPI Clock Polarity
	CPHA	3B.4	W	0	SPI Clock Phase
	BSL[3:0]	3B.3~0	W	7	Buffer shift bit counter
R3C	CRS[6:0]	3C.6~0	W	0	SPI clock Select
R3D	SPI_LENGTH	3D.6~0	W	0	SPI DMA transfer length;; Length: 1 ~64 bytes
R3E	SPI_TXDAT	3E.7~0	W	0	SPI Transmit DATA in CMD phase
R3F	SPI_RXDAT	3F.7~0	R	-	SPI Received Data
	SET0FIFO	40~47	R	-	Endpoint 0 SETUP Receive Buffer (8 Bytes)
	OUT0FIFO	48~4F	R	-	Endpoint 1 OUT Receive Buffer (8 Bytes)
	TX0FIFO	50~57	W	-	Endpoint 0 Transmit Buffer (8 Bytes)
	TX1FIFO	58~5F	W	-	Endpoint 1 Transmit Buffer (8 Bytes)
	TX2FIFO	60~67	W	-	Endpoint 2 Transmit Buffer (8 Bytes)
	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:

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

F/W handles the following tasks:

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

3.1 USB Device Address

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

3.2 Endpoint 0 receive (SET0/OUT0)

After receiving a SETUP packet and placing the data into the Endpoint 0 setup receive FIFO (SET0FIFO), TMU3130 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 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 is written into OUT0FIFO and the OUT0CNT is updated unless Endpoint 0 STALL (EP0STALL)is set or Endpoint 0 receive ready(OUT0RDY) is not clear. The data following an OUT token is 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 Endpoint 0 transmit (TX0)

After detecting a valid Endpoint 0 IN token, TMU3130 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 TMU3130 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 was successful.

3.4 Endpoint 1/2 transmit(TX1/2)

Endpoint 1 and Endpoint 2 are capable of transmit only. These endpoints are enabled when the Endpoint 1 / Endpoint 2 configuration control bit (EP1CFG/EP2CFG) is set. After detecting a valid Endpoint 1/2 IN token, TMU3130 automatically transmit 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 TMU3130 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 was successful.

3.5 Endpoint 3 transmit (TX3)

Endpoint 3 is capable of transmit only. Register F15, F19 and F1C are used to control this endpoint. Endpoint 3 is enabled when the configuration control bit (EP3CFG) is set. To properly use this endpoint, F/W must set SRAM1USB=1 or SRAM2USB=1 to assign exactly one SRAM (SRAM1 or SRAM2) as USB Bulk In buffer. Once this endpoint is enabled, F/W should set the Toggle bit (TX3TGL) and set the transmit byte count register (TX3CNT). After detecting a valid Endpoint 1 IN token, TMU3130 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 was 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 enable 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 TMU3130 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 is 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 resister. Other USB status flag includes the USB reset interrupt (RSTI), Resume input interrupt (RSMI), and USB Suspend interrupt (SUSPI).

3.8 Suspend and Resume

Once the Suspend condition is asserted, F/W can set the SUSP bit to save the power consumption of USB Engine. F/W can further save the device power by 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 TMU3130 send Resume signaling to USB bus when SUSP=1 and RSMO=1.

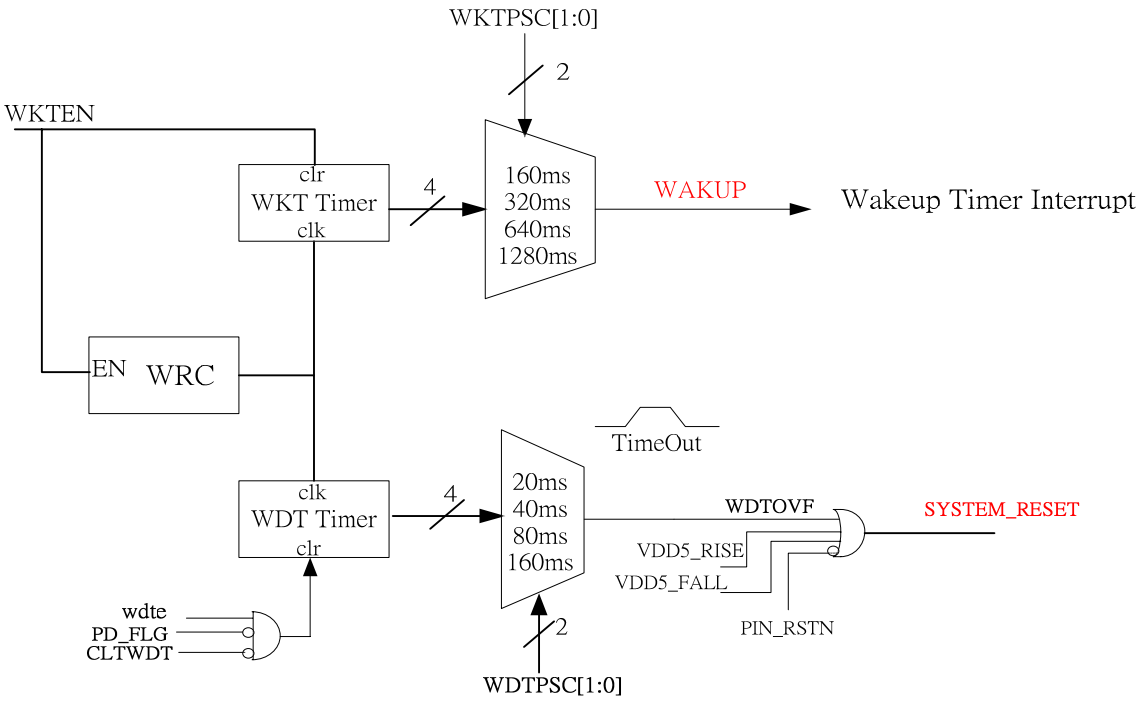
3.9 Interrupt Vector

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 serves 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 Transmit Interrupt
09	USB Bus Reset Interrupt
0a	USB Resume Interrupt
0b	Wakeup Timer Interrupt
0c	Timer0 Interrupt
0d	PB0 External I/O Interrupt
0e	Key Board Interrupt
0f	VDD5V Rise Interrupt
10	Reserved
11	Timer1 Interrupt

4. Wakeup Timer and Watch Dog Timer

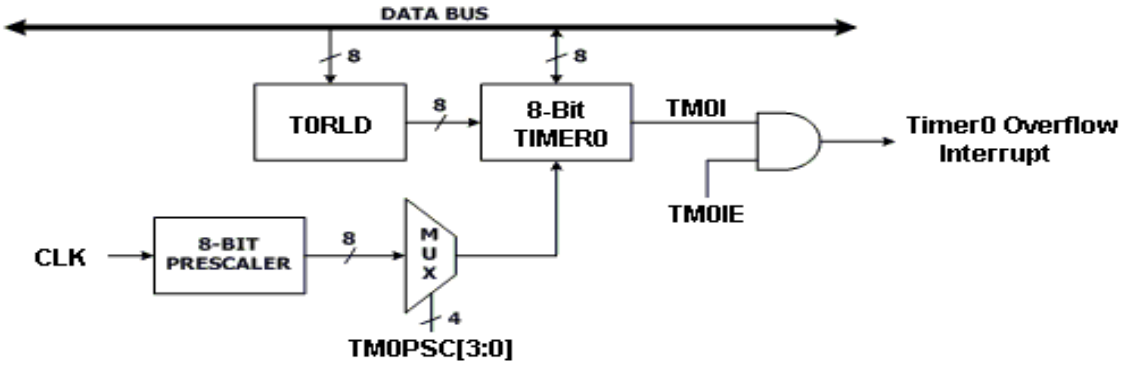
The WKT and WDT use the same internal RC(WRC). This internal RC(WRC) can be disabled by setting R06[7] “High” for power saving. The overflow period of WDT can be selected from 20mS to 80mS and the wakeup period of WKT can be selected from 160mS to 1280mS. The WDT is enable and cleared by the CLRWDT instruction.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

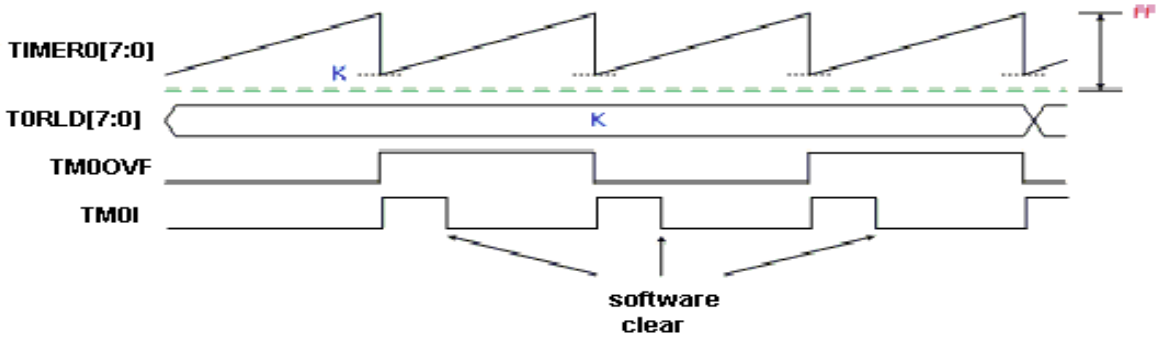


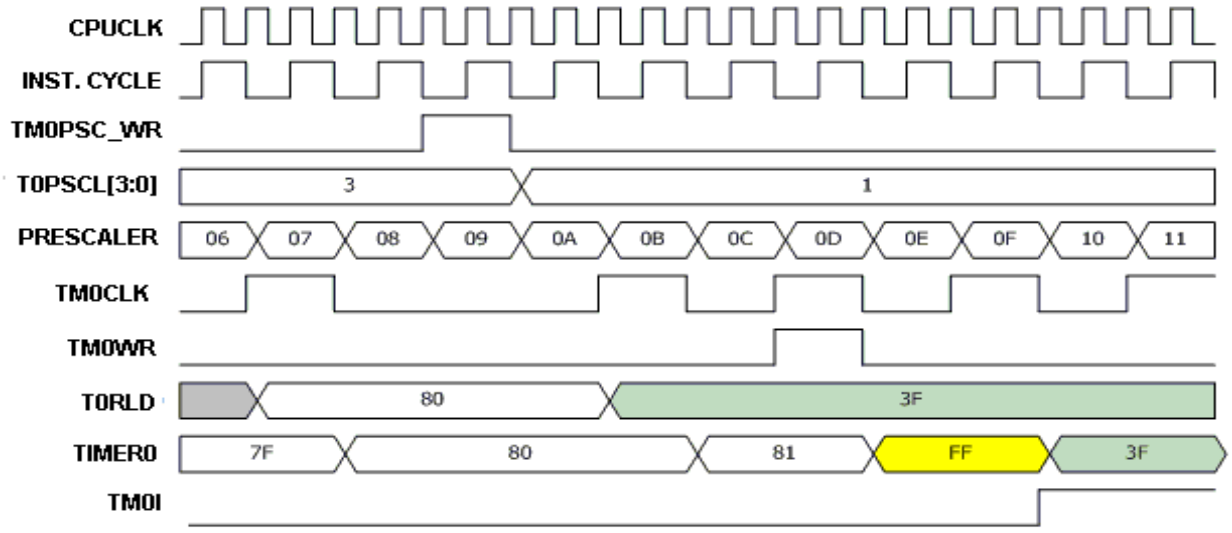
5. Timer

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 automatic reloads a new “offset value” (T0RLD) while it rolls over base 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).

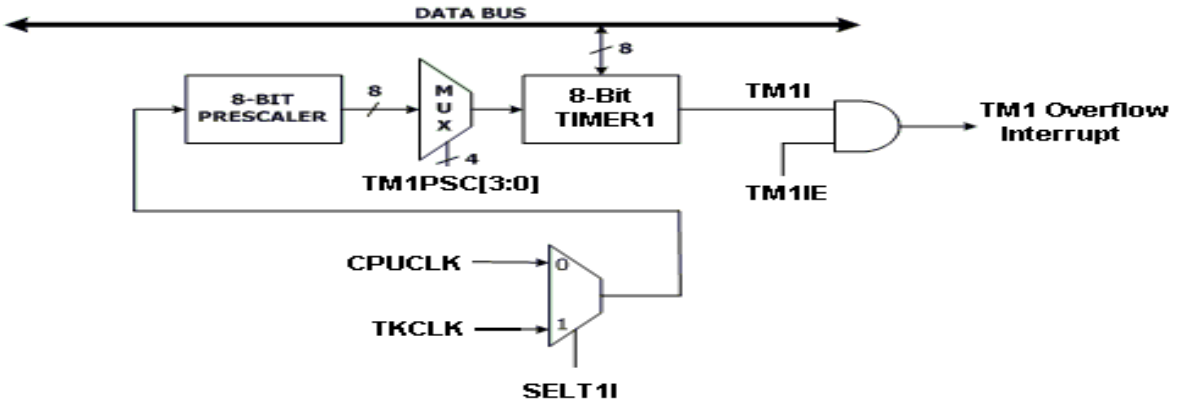




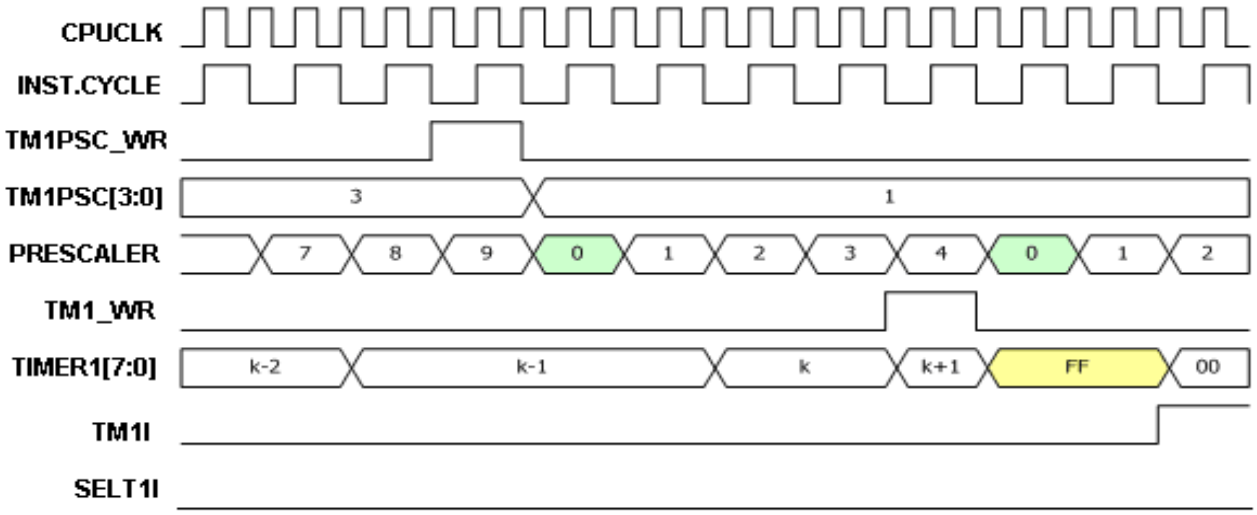


5.2 Timer1: 8-bit Timer/Counter with Pre-scale (PSC)

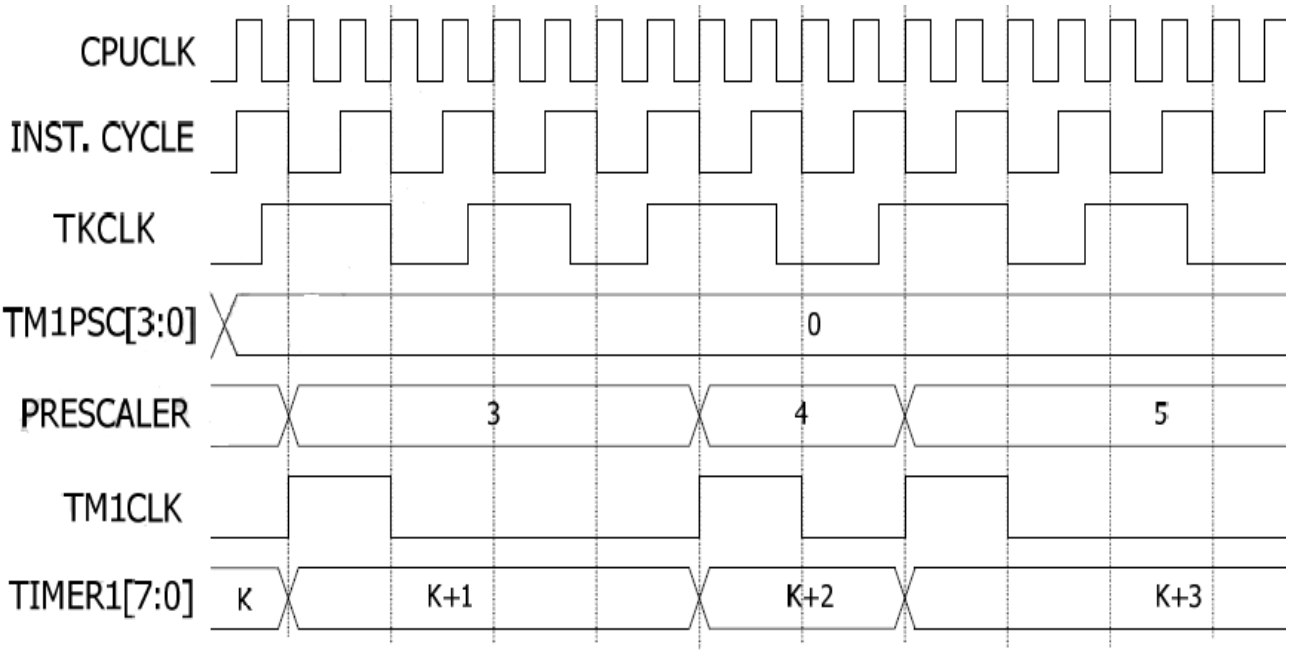
The Timer1 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer1 increases itself periodically and automatic roll over base on the pre-scaled clock source, which can be the instruction cycle or touch key induced clock(TKCLK). The Timer1 increase 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 clear 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.

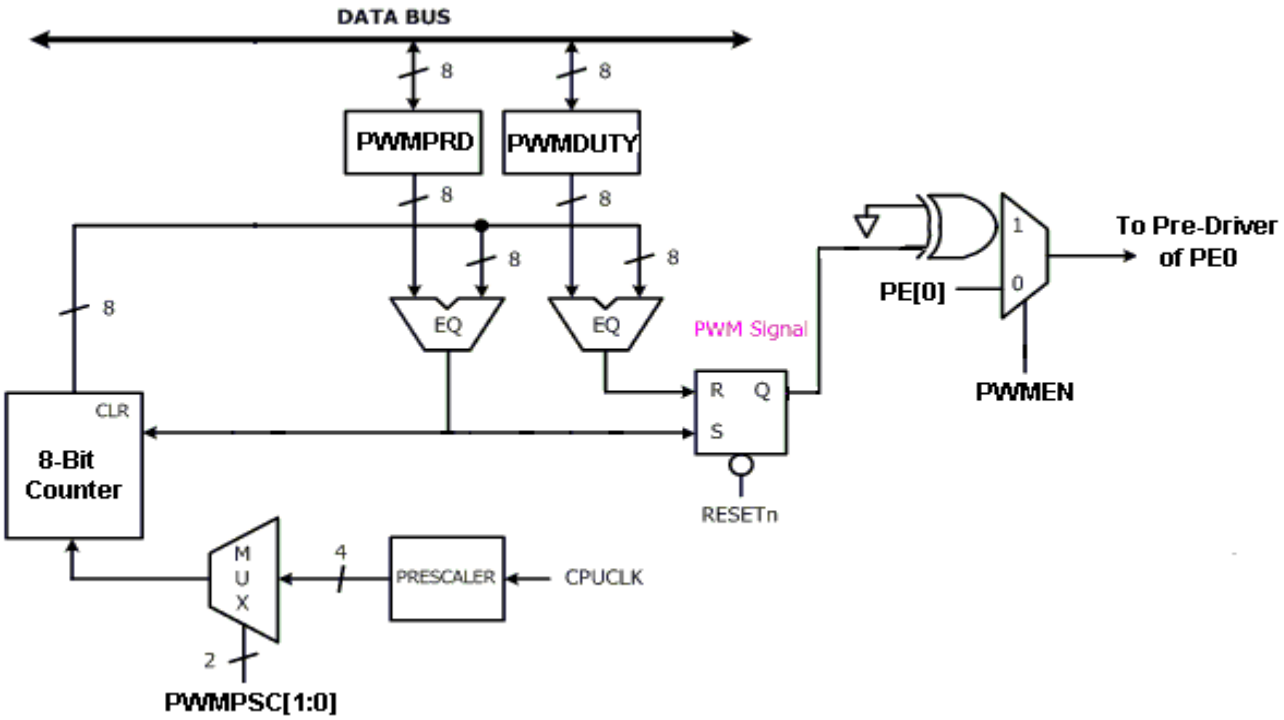


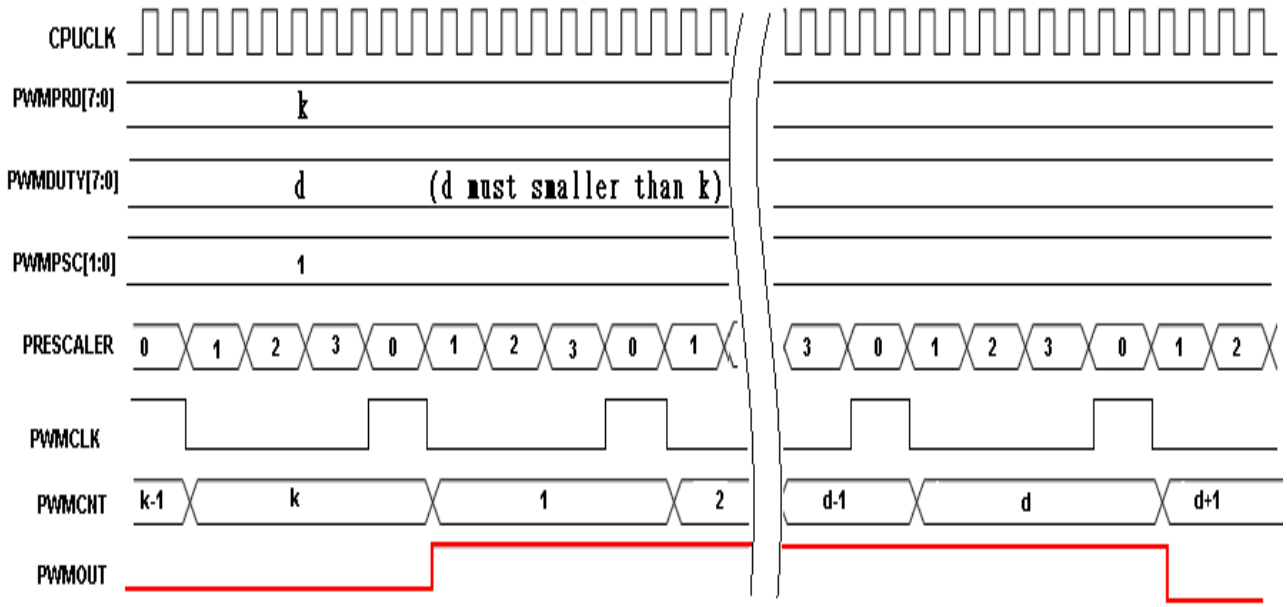
The following timing diagram describes the Timer1 works in counter mode. If SELT1I=1 then the Timer1 counter source clock is from Touch Key module that depends on TKE bit. In this mode the counter is used for Touch Key function



6. 8-bit PWM

The PWM will be enabled by setting R30[0](PWMEN) to “1”. Once the PWMEN is been 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). Te 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 finish current PWM period, the new value of PWMDUTY will be updated.


DATA BUS
8
8
PWMPRD
PWMDUTY
8
8
8
8
8
8
EQ
EQ
PWM Signal
R
Q
S
RESETn
CLR
8-Bit
Counter
PE[0]
1
0
To Pre-Driver
of PE0
PWMEN
M
U
X
4
PRESCALER
CPUCLK
2
PWMPSC[1:0]


CPUCLK
PWMPRD[7:0]
k
PWMDUTY[7:0]
d
(d must smaller than k)
PWMPSC[1:0]
1
PRESCALER
0 1 2 3 0 1 2 3 0 1
3 0 1 2 3 0 1 2
PWMCLK
PWMCNT
k-1
k
1
2
d-1
d
d+1
PWMOUT

7. SPI (Serial Peripheral Interface)

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

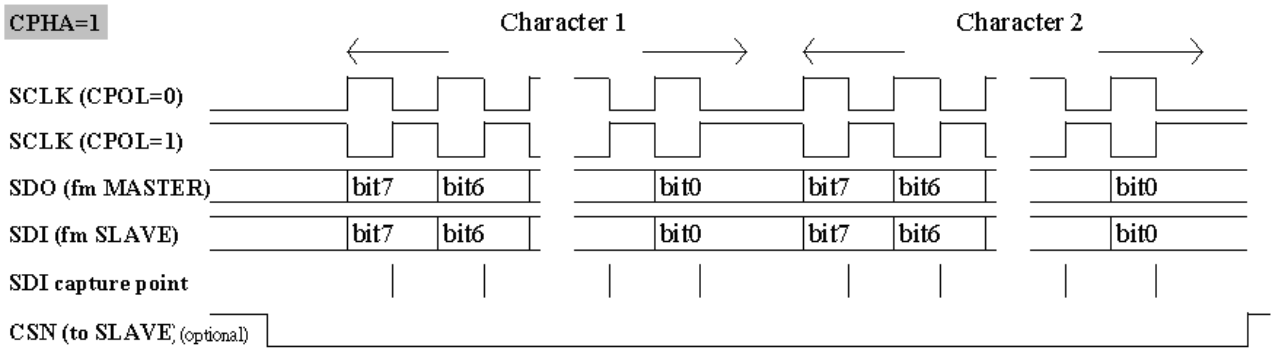
CRS[6:0]	SPI Clock Rate
0	6Mbps
3	1.5Mbps
15	375kbps

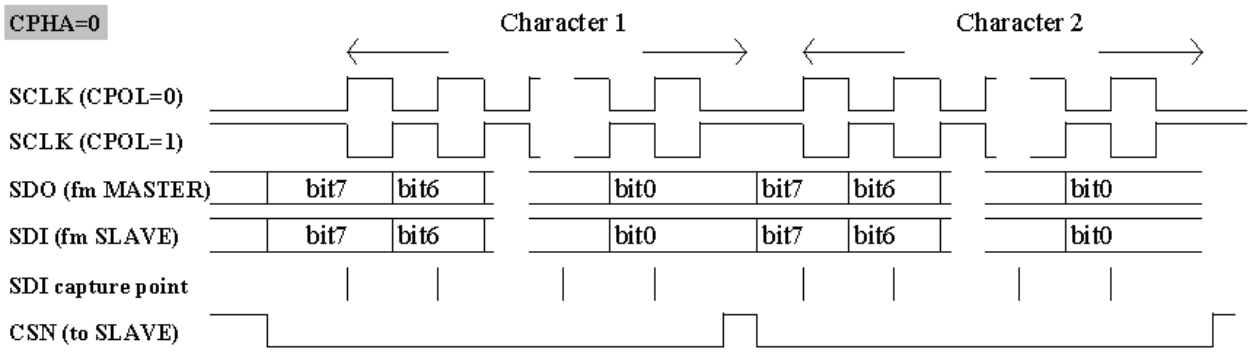
Note: CPUCLK = 12Mhz

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

There are two data transfer mode. One is command phase mode, in this mode the data transfer length is “1” and the data must preset in R3E. The other one is data phase, in this mode data transfer length is according to how many bytes data will be transfer. The length value is stored in R3D and the transfer data is stored in the SRAM(RAM1 or RAM2).

SPI Timing





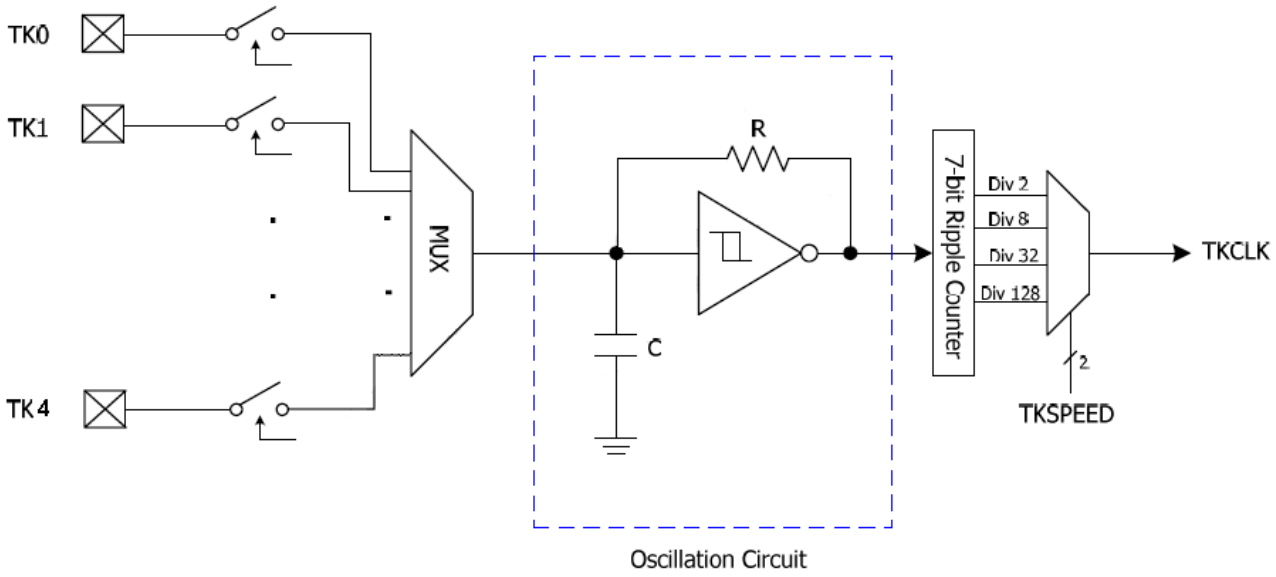
8. Touch Key

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 showed 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 send to Timer1 to count at the rising edge.

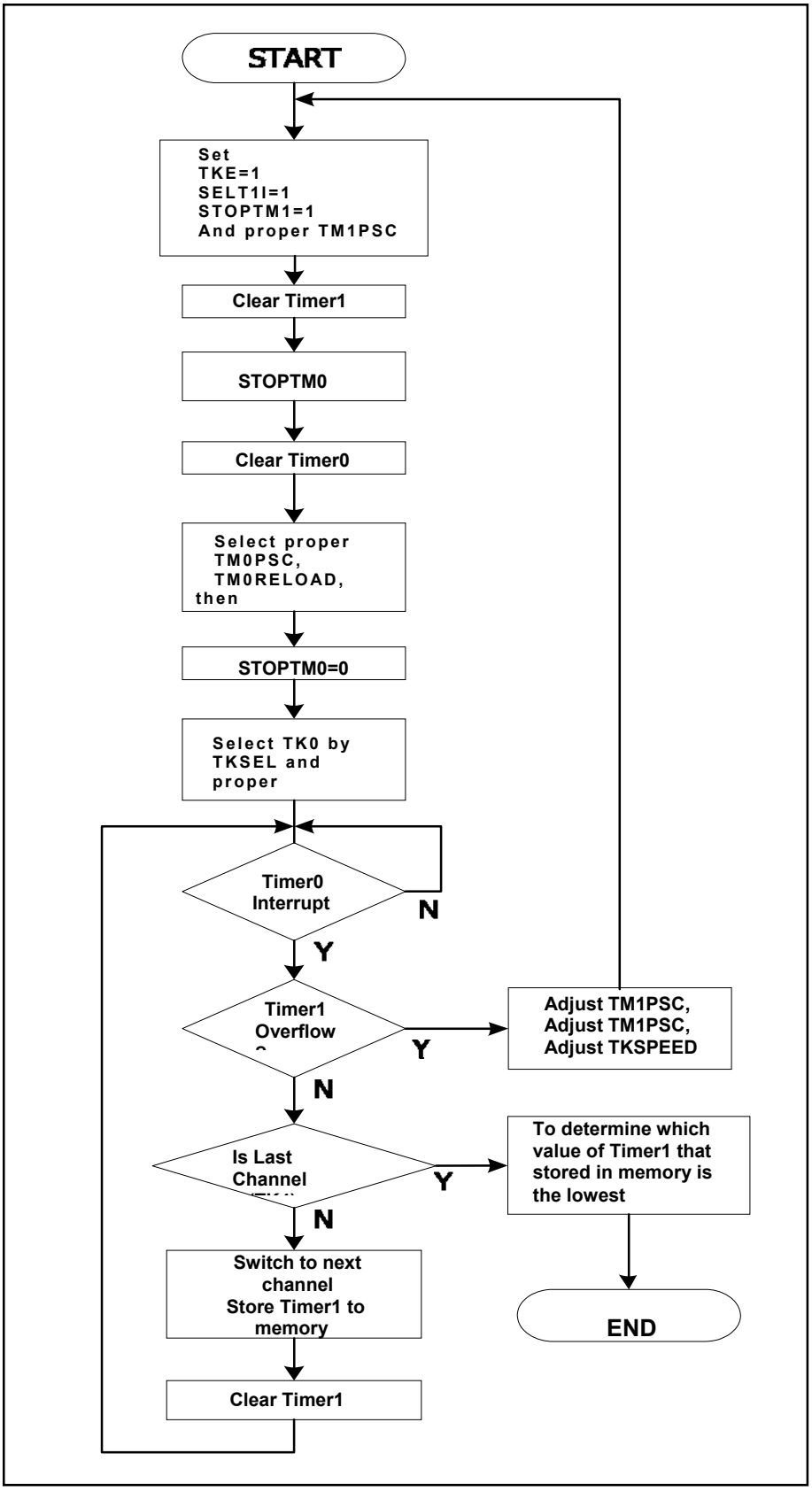
If the human finger tips close to the touch pad, the equivalent capacitance of C will be increased, that is, the oscillation frequency will be decreased.

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 been 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, that is the key in touching or approaching.



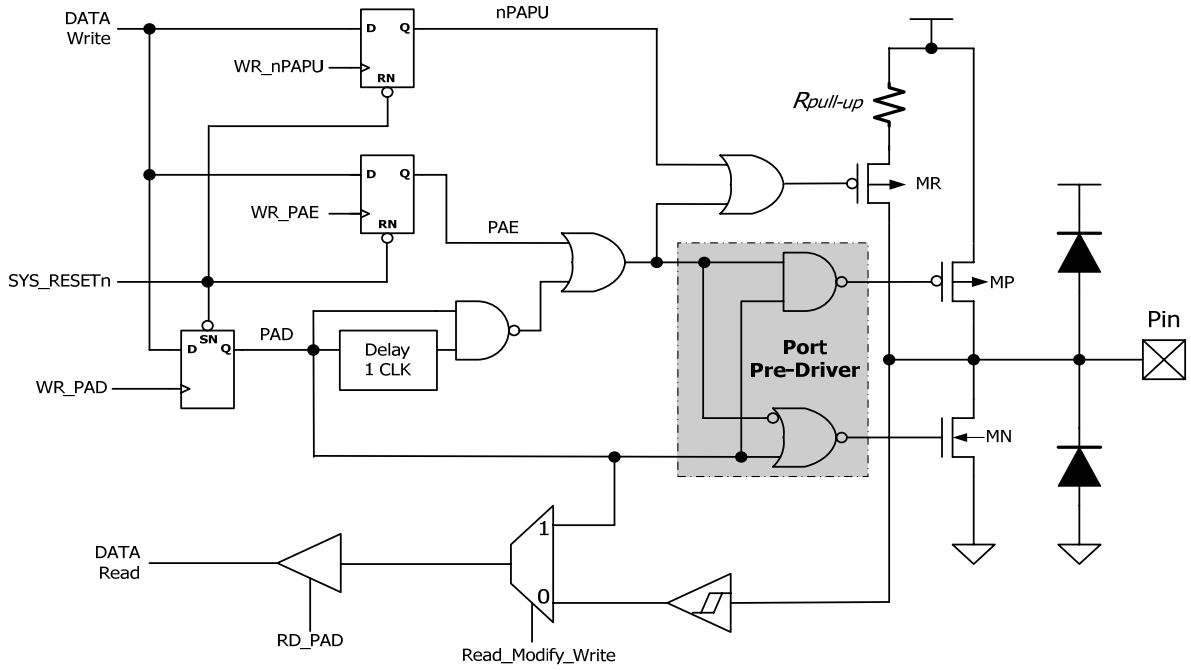
The flowchart described below show how to use Timer0 and Timer1 to determine what channel of the Touch Key is been pressed. Using the 8-bit Timer0 to set up a fix interval of time and utilize the Time0 interrupt to stop Timer1 and store its value if it is not overflow. Determine the lowest value of Timer1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


START
Set
TKE=1
SELT1I=1
STOPTM1=1
And proper TM1PSC
Clear Timer1
STOPTM0
Clear Timer0
Select proper
TM0PSC,
TM0RELOAD,
then
STOPTM0=0
Select TK0 by
TKSEL and
proper
Timer0
Interrupt
N
Y
Timer1
Overflow
Y
Adjust TM1PSC,
Adjust TM1PSC,
Adjust TKSPEED
N
Is Last
Channel
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

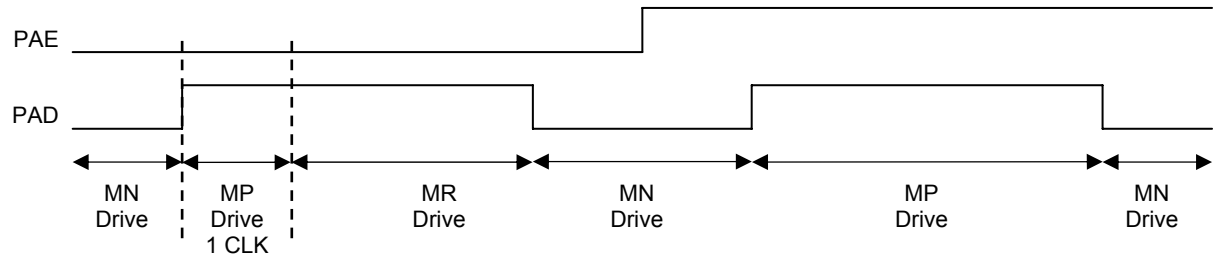
9. I/O Port

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 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.

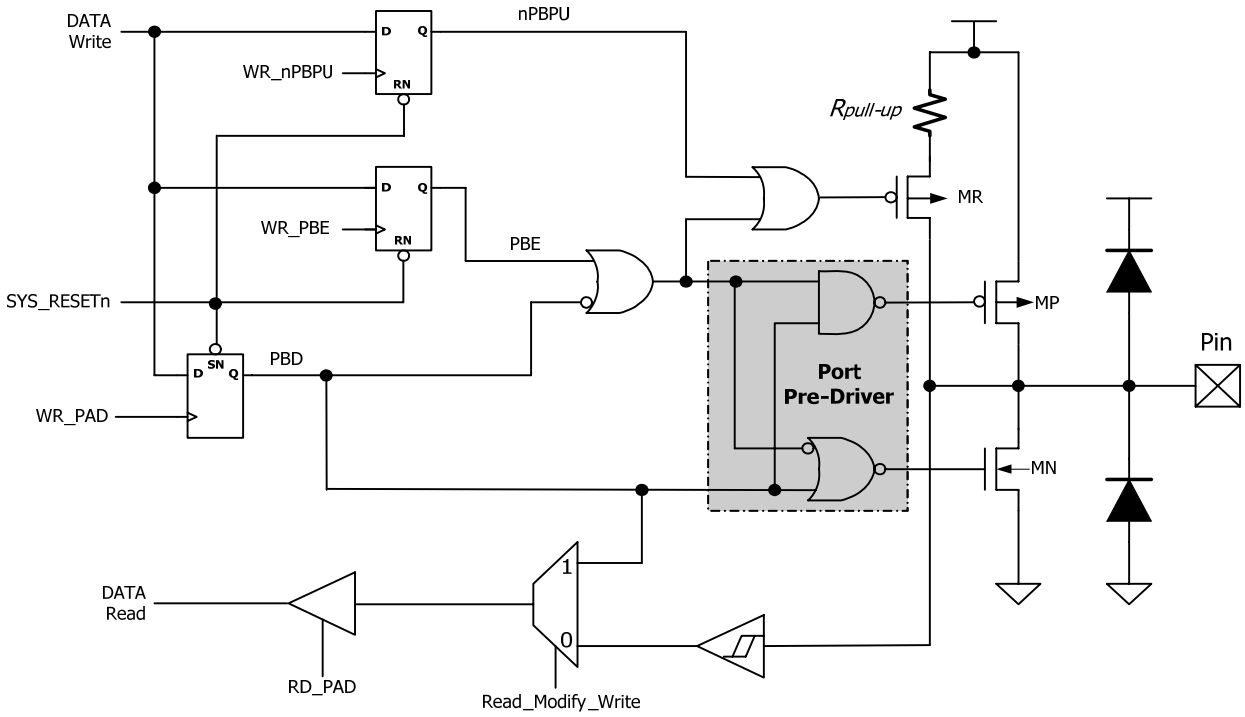


PA0-7, nPAPU=0

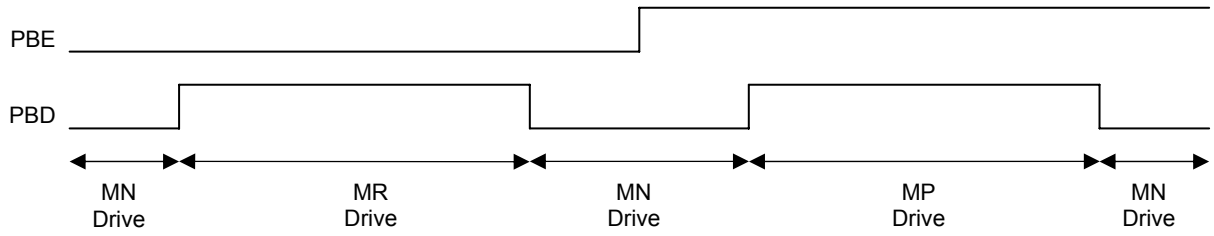


9.2 PB0-1

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.

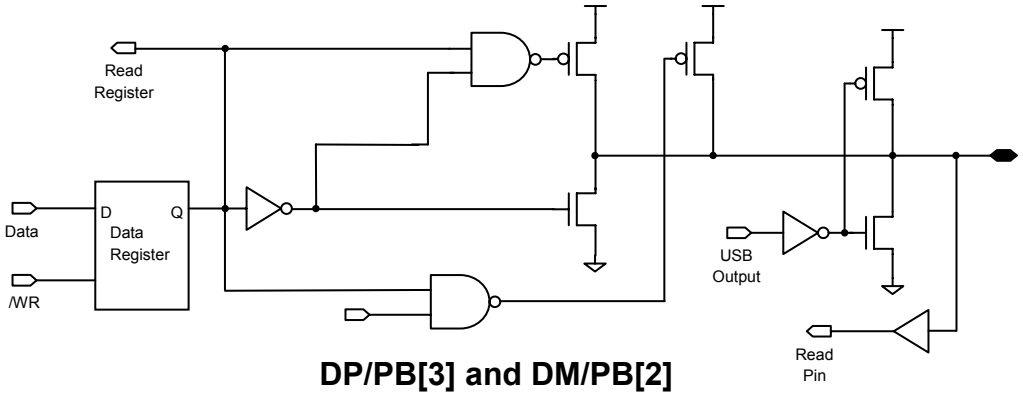


PB0-1, nPBPU=0



9.3 PB3(DP) and PB2(DM)

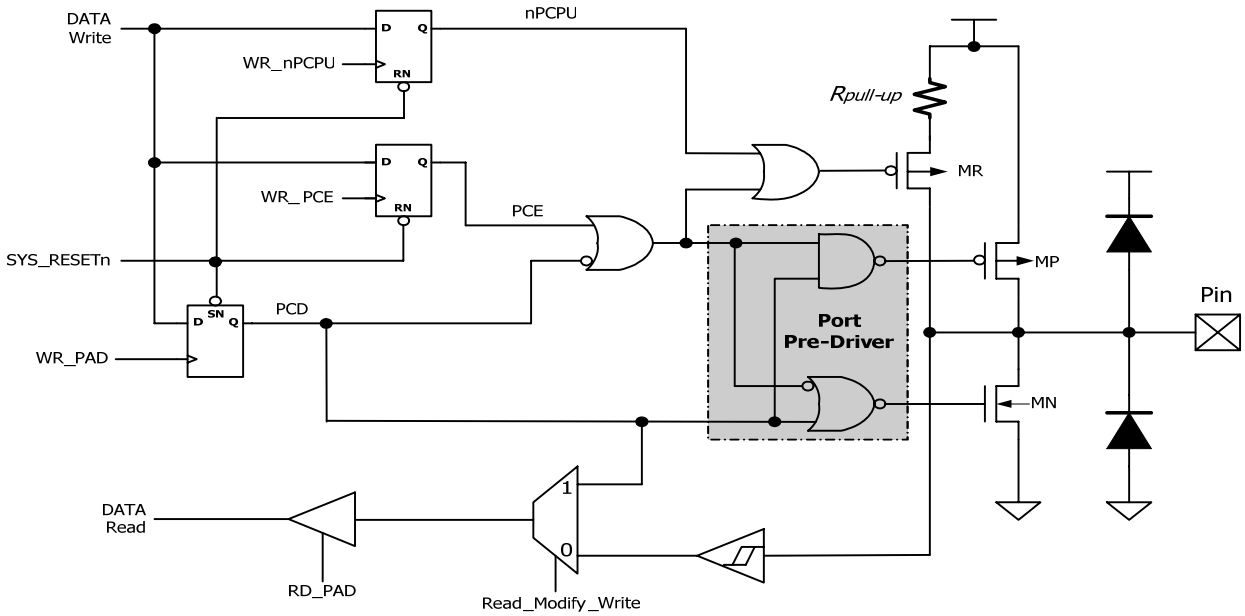
These pins are similar to PB[1:0], except they share the pin with USB function.



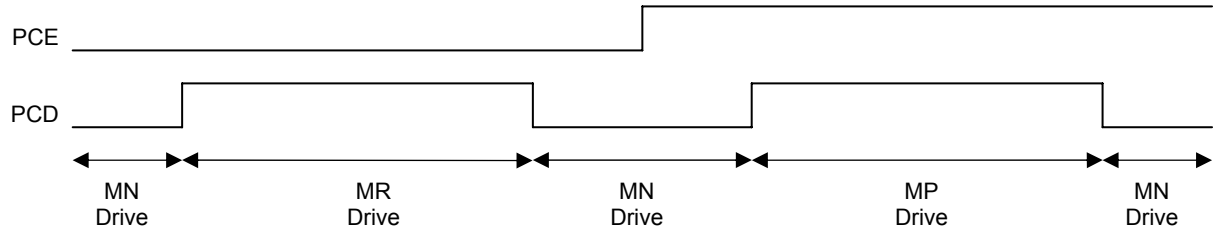
DP/PB[3] and DM/PB[2]

9.4 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

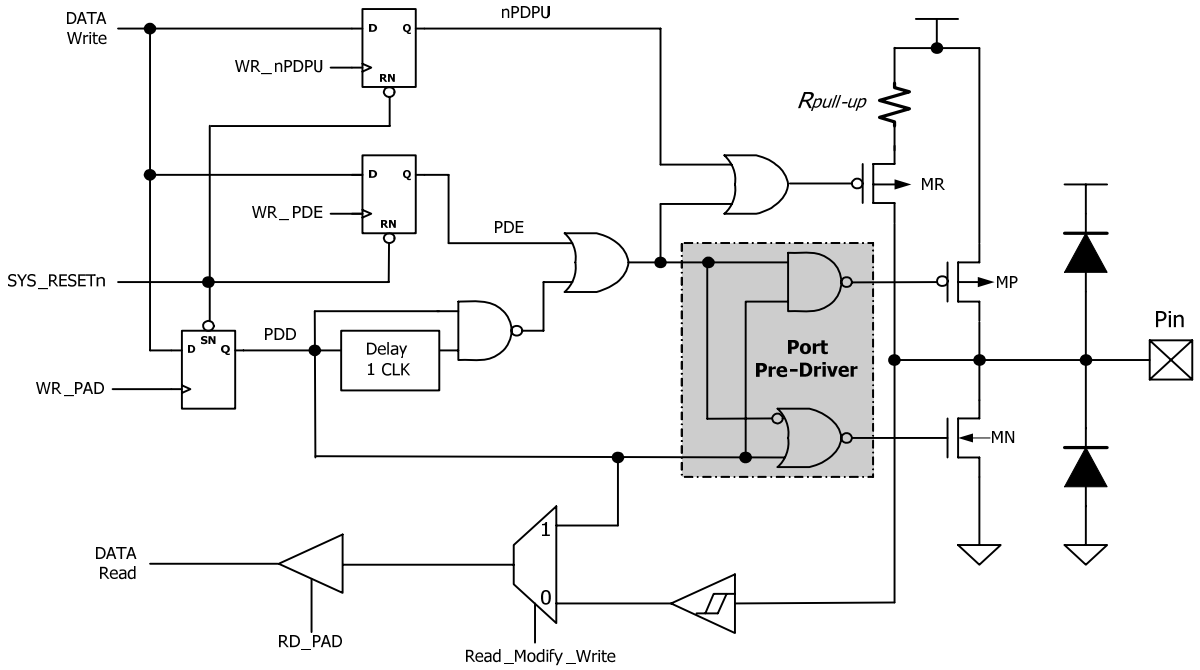


PC0-7, nPCPU=0

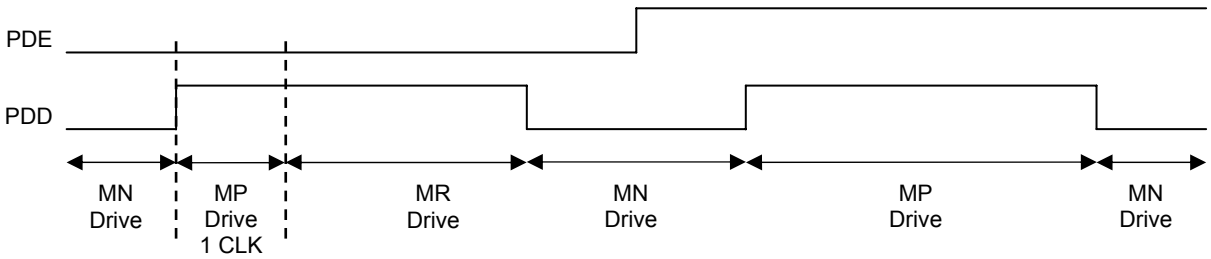


9.5 PD0-7

PortD pins are almost same as PA0-7, except the pull-up enable bit. There are 8 different pull-up enable bit nPAPU[7:0] to control PortA. Only one pull-up enable bit nPDPU is used to control PortD.



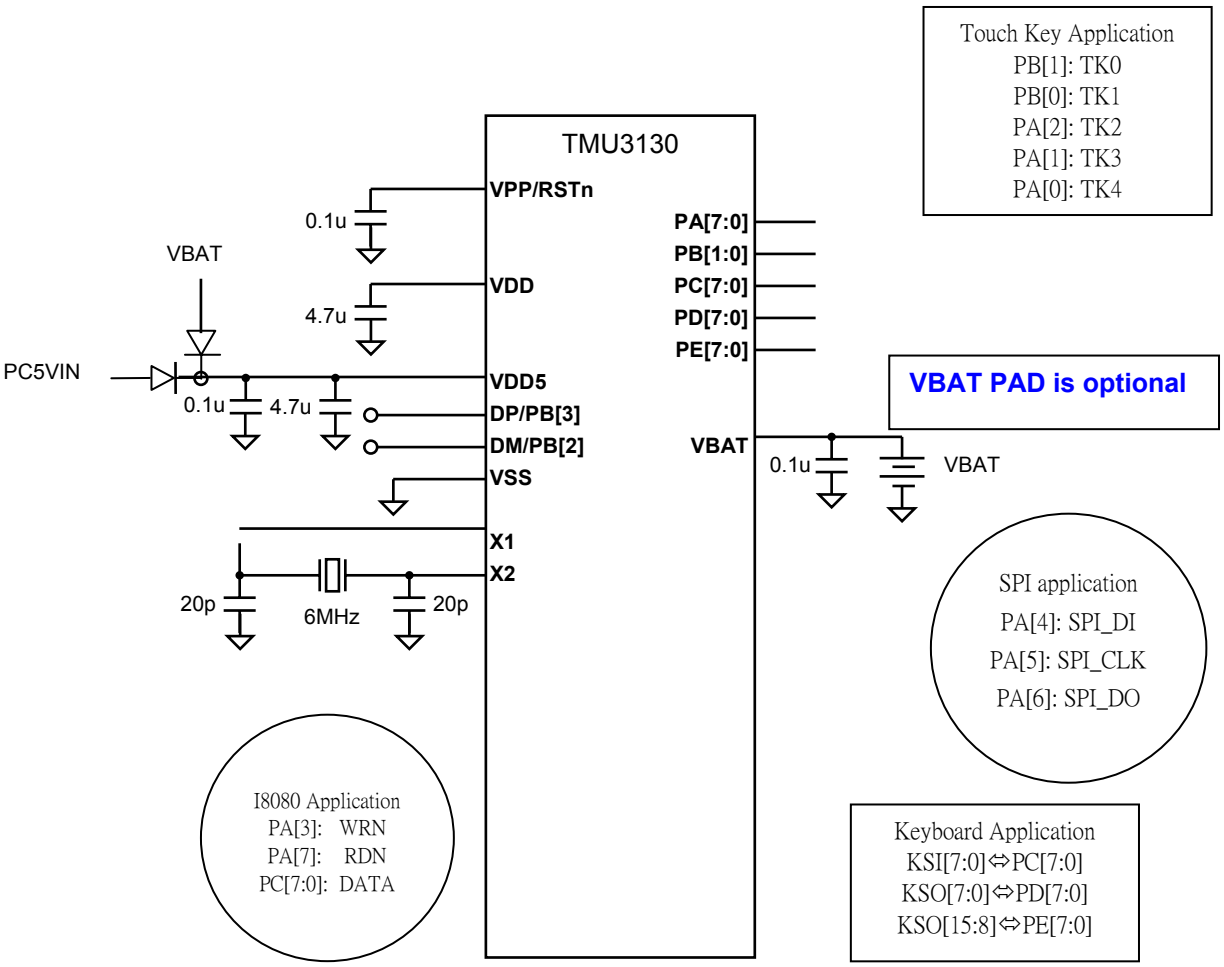
PD0-7, nPDPU=0



9.6 PE0-7

PortE pins are same as PD0-7.

10. Application



11.Electrical Characteristics

ABSOLOUTE 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	℃

RECOMMEND OPERATING CONDITION

at Ta=-20℃ to 70℃,GND= 0V

Name	Symb.	Min.	Typical	Max.	Unit	Condition
Supply Voltage	VDD5	4.5		5.5	V	
Battery Voltage, if apply	Vbat	2.1		3.6	V	
VDD output voltage	VDD		3.45		V	VDD5=5V
			2.99		V	VDD5=3V
			2.93		V	Vbat=3V, VDD5=0V
Input “H” Voltage	Vih	0.6VDD			V	
Input “L” Voltage	Vil1			0.3VDD	V	

DC CHARACTERISTICS

at Ta=-25 ℃,VDD=5.0V, VSS= 0V, Fosc=6MHz

Name	Symb.	Min.	Typ.	Max.	Unit	Condition
Operating current	Icc		5.5		mA	Fosc=6MHz
Power Down current	Ipd			1	uA	No load
Output High Voltage	Voh1	2.8			V	VDD5=5V, Ioh=15mA
	Voh2	2.3			V	VDD5=3V, Ioh=12mA
Output Low Voltage	Vol1		0.3		V	VDD5=5V, Iol=17.4mA
	Vol2		0.3		V	VDD5=3V, Iol=16mA

AC CHARACTERISTICS

at Ta=25 ℃,VDD5V=5.0V, VSS= 0V, Fosc=6MHz

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.