Fremont Micro Devices

FT62F08

Data Sheet

Key Features

8–bit EEPROM-based RISC MCU

Program: 8k x 14; RAM: 1k x 8; Data: 256 x 8

16 / 20 / 24 / 28 / 32 Pins

12–bit high accuracy ADC

4 Timers, 7 Individual PWMs − 3 with Deadband

15 high reliability Touch Keys

SPI, I2C, USART

Low Standby, WDT and Operating Current

POR, LVR, LVD – Single Input Comparator

Selectable Source and Sink Current

High ESD, High EFT

Low V_{DD} Operating Voltage

Tunable HIRC

Rev2.05
www.fremontmicro.com

8−bit CPU (EEPROM)

- 49 RISC instructions: 1T , 2T or 4T
- 16 MHz / 1T ($V_{DD} \geq 2.7$)
- Up to 32 pins

Memory

- PROGRAM: 8k x 14 bit (R/W Protect)
- DATA: 256 x 8 bits
- RAM: 1k x 8 bits
- 16-level Hardware Stack
- Sector encryption, support IAP

Operation Conditions (5V, 25°C)

- V_{DD} ($V_{POR} \leq 1.9V$) V_{POR} − 5.5 V
  (Self-regulated by POR, ≤1.7V for above 0 °C)
- Operation temperature Grade 1 $^{-}$40 − $^{+}$125 °C
- Operation temperature Grade 2 $^{-}$40 − $^{+}$105 °C
- Operation temperature Grade 3 $^{-}$40 − $^{+}$85 °C
- Low Standby 0.2 μA
- WDT 2.9 μA
- Normal Mode (16 MHz / 1T) 276 μA/mips

High Reliability

- 1 M cycles of erasures (typical)
- > 20 years / 85 °C storage (typical)
- ESD > 8 kV, EFT > 5.5 kV

ADC (12−bit)

- 12−bit accuracy (≤ 4 MHz ADC clock)
- 8 + 1 channels
- $V_{ADC-REF}$
  - ✓ Internal: 0.5, 2.0, 3.0, V_{DD}
  - ✓ External: +, − optional
- Automatic threshold comparison
- Automatic calibration

PWM (Total 7)

- Support RUN in SLEEP
- 7 capture /compare/PWM channels:
  - ✓ Independent: Duty Cycle, Polarity
- 3 channels (up to 6 I/Os):
  - ✓ Complementary + Deadband
- Auto Fault-Breaking (I/O, LVD, ADC)
- Edge-aligned, Center-aligned
- One-Pulse mode

Timers

- WDT (16−bit) : 3−bit prescaler
- Timer1 (16−bit) : 16−bit prescaler
- Timer2 (16−bit) : 4−bit prescaler
- Timer4 (8−bit) : 3−bit prescaler
- Auto-reloading
- Support RUN in SLEEP
- Sysclk, LIRC, 1 or 2x {HIRC, Crystal, EC}

TOUCH

- Up to 15 touch keys, support waterproof function

Communication Interface

- SPI , I2C, USART

I/O PORTS (Up to 30 I/O)

- Resistive Pull-Up/Pull-Down
- Open-Drain
- 30 I/O I_{SOURCE}: 2, 4, 14 or 26 mA (5V, 25°C)
- 30 I/O I_{SINk}: 53 or 62 mA (5V, 25°C)
- 30 I/Os: Interrupt/Wakeup

Power Management

- SLEEP
- LVR: 2.0, 2.2, 2.5, 2.8, 3.1, 3.6, 4.1 (V)
- LVD: 2.0, 2.4, 2.8, 3.0, 3.6, 4.0 (V)
  (LVD also functions as a single input comparator.)

System Clock (SysClk)

- HIRC High Speed Internal Oscillator
  - ✓ 16MHz <±1% typical (2.5V, 25°C)
  - ✓ Tunable
  - ✓ 1, 2, 4, 8, 16, 32, 64, 128 divider
- LIRC Low Power Internal Oscillator
  - ✓ 32 kHz or 256 kHz
- External Clock (I/O input)
- LP/XT crystal input
  - ✓ HIRC or LIRC during startup
  - ✓ Fail-Safe Clock Monitor(FSCM)

Other Features (Welecome to enquire)

- ½ V_{DD} LCD bias

Integrated Development Environment (IDE)

- On-Chip Debug (OCD), ISP
- 3 hardware breakpoints
- System-Reset, Stop, Single Step, Run.etc

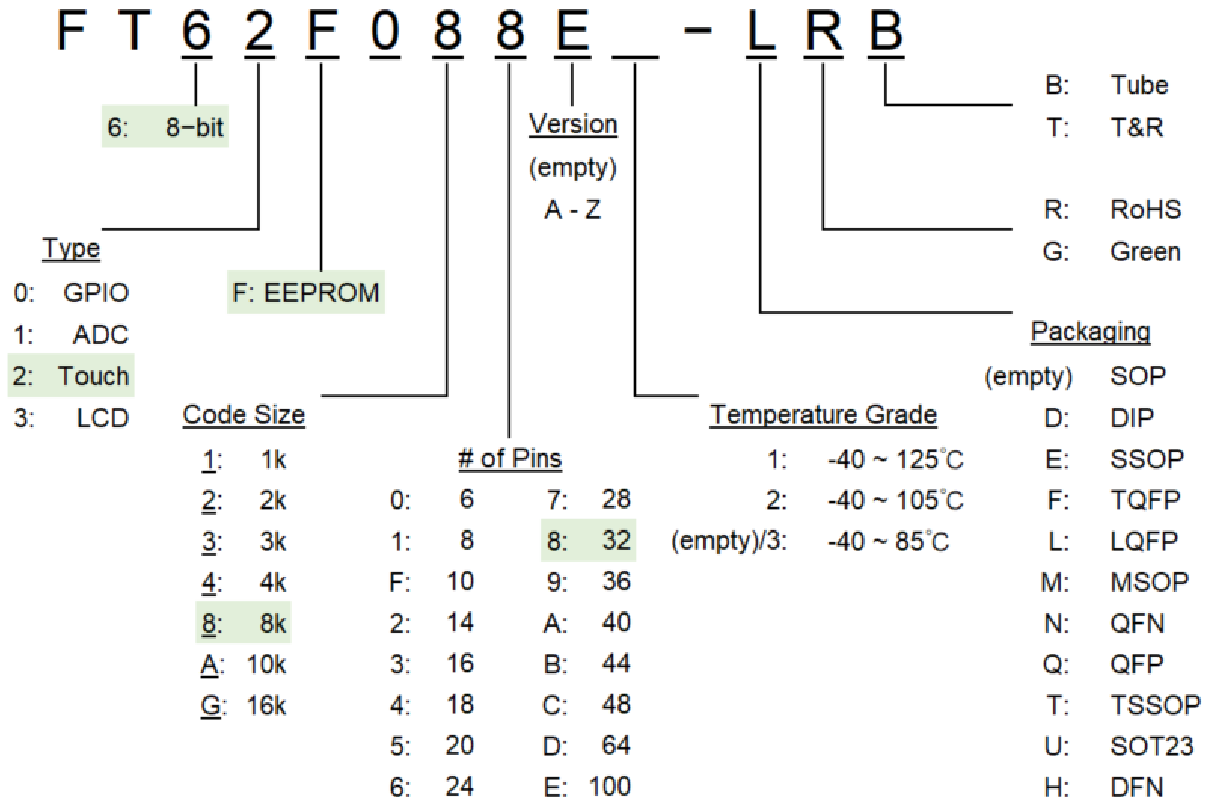
Packages

- SOP16 TSSOP20 SOP20 SOP24
  TSSOP24 SOP28 LQFP32

PARTS INFORMATION AND SELECTIONS

Part Number	Number of I/O	Package
FT62F083−ab	14	SOP16
FT62F085E−Tab	18	TSSOP20
FT62F085E- ab		SOP20
FT62F086E− Tab	22	TSSOP24
FT62F086E- ab		SOP24
FT62F087A- ab	26	SOP28
FT62F087B- ab		
FT62F087F- ab		
FT62F087D- ab		
FT62F088E- Lab	30	LQFP32

Where a = R; RoHS b = B; Tube
= G; Green = T; T&R



MCU Part Number Selections

Revision History

Date	Revision	Description
2021-02-24	1.08	Preliminary version
2021-09-15	2.00	Complete overhauled register table, updated MCU Part Number Selections
2021-10-22	2.01	Updated Oscillator modules, summary of USART interface related registers
2021-10-29	2.02	1. Add the following models: FT62F087G-RB (pins are the same as FT62F087A-RB) FT62F087F-RB (pins are the same as FT62F087B-RB) FT62F088E-NRB (pins are the same as FT62F088-NRB) 2. Remove the following models: FT62F085-RB, FT62F085A-TRB, FT62F086-RB, FT62F086-TRB, FT62F087-RB, FT62F088-LRB
2022-06-14	2.03	1. Complete overhauled version (Too many changes to list. Please dis-regard preliminary version) 2. Remove FT62F087G-RB and FT62F088E-NRB
2022-08-25	2.04	Updated Section 4 System Reset
2023-06-21	2.05	Remove FT62F088-NRB; updated program example in Section 10; updated SOP16 packaging information; typo correction

Contents

1. BLOCK DIAGRAM AND PINOUTS

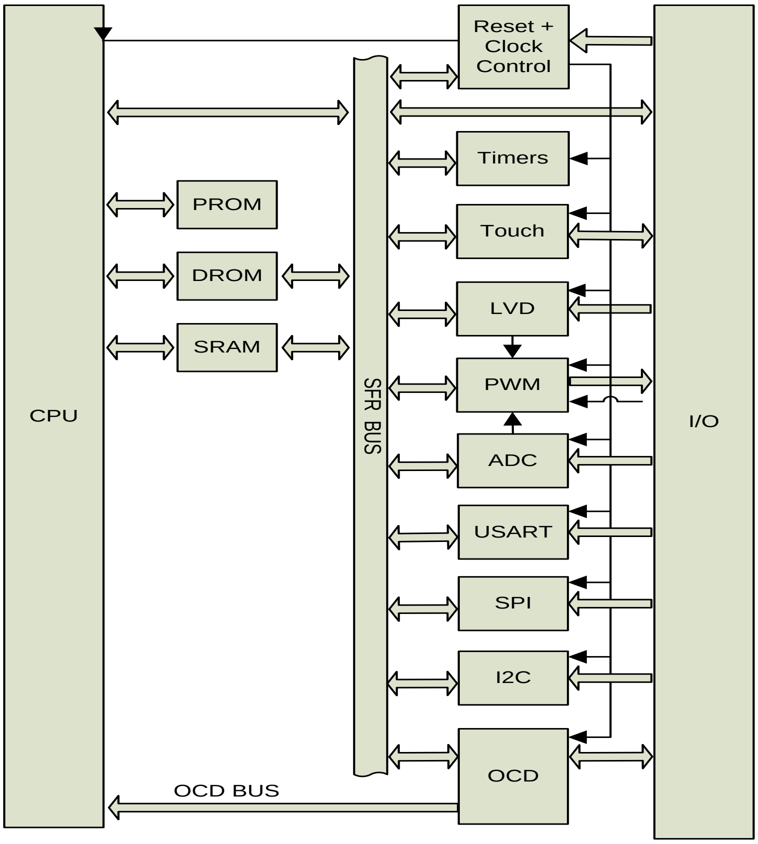


Figure 1-1 System Block Diagram

The list of standard abbreviations is as follows:

Abbreviation	Description
CPU	Central Processing Unit
SFR	Special Function Registers
SRAM	Static Random Access Memory
DROM	Data EEPROM
PROM	Program EEPROM
Timers	WDT, Timer1, Timer2, Timer4
PWM	Pulse Width Modulator
ADC	Analog to Digital Converter
LVD	Low Voltage Detect / comparator
Touch	Touch
SPI	Serial Peripheral Interface
USART	Universal Synchronous Asynchronous Receiver Transmitter
I2C	Inter-Integrated Circuit
OCD	On Chip Debug
I/O	Input / Output

1.1. Pinouts

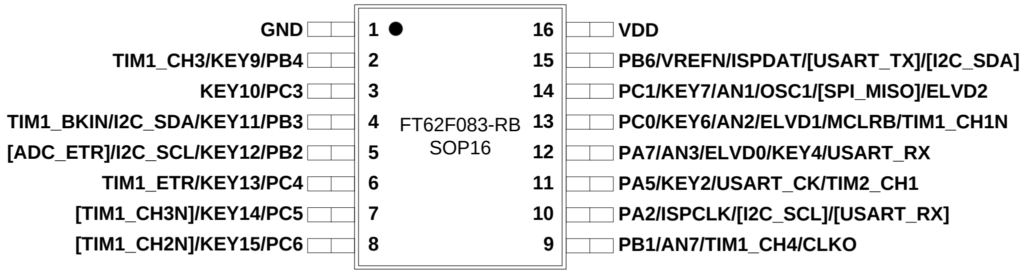

GND
TIM1_CH3/KEY9/PB4
KEY10/PC3
TIM1_BKIN/I2C_SDA/KEY11/PB3
[ADC_ETR]/I2C_SCL/KEY12/PB2
TIM1_ETR/KEY13/PC4
[TIM1_CH3N]/KEY14/PC5
[TIM1_CH2N]/KEY15/PC6
1
2
3
4
5
6
7
8
FT62F083-RB
SOP16
16
15
14
13
12
11
10
9
VDD
PB6/VREFN/ISPDAT/[USART_TX]/[I2C_SDA]
PC1/KEY7/AN1/OSC1/[SPI_MISO]/ELVD2
PC0/KEY6/AN2/ELVD1/MCLRB/TIM1_CH1N
PA7/AN3/ELVD0/KEY4/USART_RX
PA5/KEY2/USART_CK/TIM2_CH1
PA2/ISPCLK/[I2C_SCL]/[USART_RX]
PB1/AN7/TIM1_CH4/CLKO

Figure 1-2 SOP16

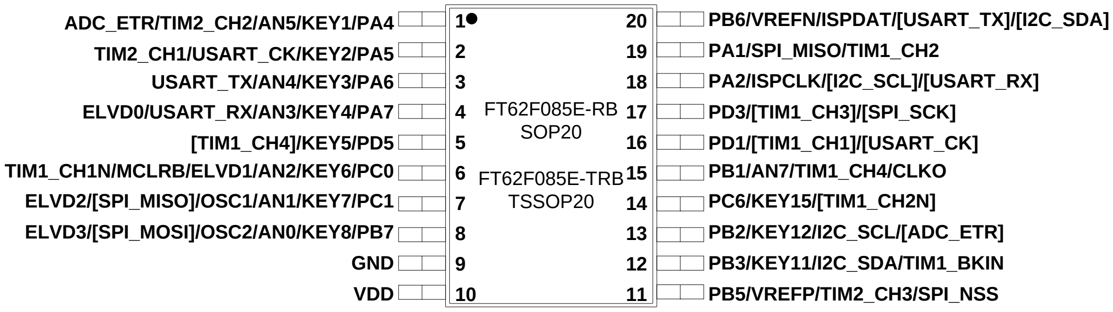

ADC_ETR/TIM2_CH2/AN5/KEY1/PA4
TIM2_CH1/USART_CK/KEY2/PA5
USART_TX/AN4/KEY3/PA6
ELVD0/USART_RX/AN3/KEY4/PA7
[TIM1_CH4]/KEY5/PD5
TIM1_CH1N/MCLRB/ELVD1/AN2/KEY6/PC0
ELVD2/[SPI_MISO]/OSC1/AN1/KEY7/PC1
ELVD3/[SPI_MOSI]/OSC2/AN0/KEY8/PB7
GND
VDD
1
2
3
4
5
6
7
8
9
10
FT62F085E-RB
SOP20
FT62F085E-TRB
TSSOP20
20
19
18
17
16
15
14
13
12
11
PB6/VREFN/ISPDAT/[USART_TX]/[I2C_SDA]
PA1/SPI_MISO/TIM1_CH2
PA2/ISPCLK/[I2C_SCL]/[USART_RX]
PD3/[TIM1_CH3]/[SPI_SCK]
PD1/[TIM1_CH1]/[USART_CK]
PB1/AN7/TIM1_CH4/CLKO
PC6/KEY15/[TIM1_CH2N]
PB2/KEY12/I2C_SCL/[ADC_ETR]
PB3/KEY11/I2C_SDA/TIM1_BKIN
PB5/VREFP/TIM2_CH3/SPI_NSS

Figure 1-3 SOP20 / TSSOP20

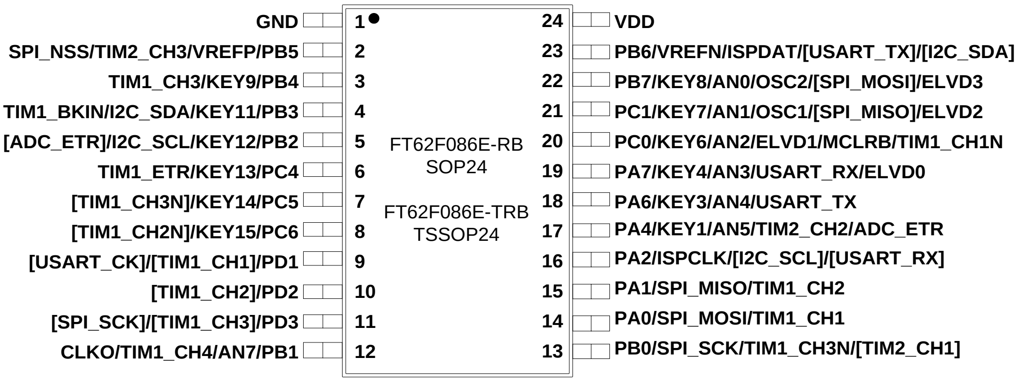

GND
SPI_NSS/TIM2_CH3/VREFP/PB5
TIM1_CH3/KEY9/PB4
TIM1_BKIN/I2C_SDA/KEY11/PB3
[ADC_ETR]/I2C_SCL/KEY12/PB2
TIM1_ETR/KEY13/PC4
[TIM1_CH3N]/KEY14/PC5
[TIM1_CH2N]/KEY15/PC6
[USART_CK]/[TIM1_CH1]/PD1
[TIM1_CH2]/PD2
[SPI_SCK]/[TIM1_CH3]/PD3
CLKO/TIM1_CH4/AN7/PB1
1
2
3
4
5
6
7
8
9
10
11
12
FT62F086E-RB
SOP24
FT62F086E-TRB
TSSOP24
24
23
22
21
20
19
18
17
16
15
14
13
VDD
PB6/VREFN/ISPDAT/[USART_TX]/[I2C_SDA]
PB7/KEY8/AN0/OSC2/[SPI_MOSI]/ELVD3
PC1/KEY7/AN1/OSC1/[SPI_MISO]/ELVD2
PC0/KEY6/AN2/ELVD1/MCLRB/TIM1_CH1N
PA7/KEY4/AN3/USART_RX/ELVD0
PA6/KEY3/AN4/USART_TX
PA4/KEY1/AN5/TIM2_CH2/ADC_ETR
PA2/ISPCLK/[I2C_SCL]/[USART_RX]
PA1/SPI_MISO/TIM1_CH2
PA0/SPI_MOSI/TIM1_CH1
PB0/SPI_SCK/TIM1_CH3N/[TIM2_CH1]

Figure 1-4 SOP24 / TSSOP24

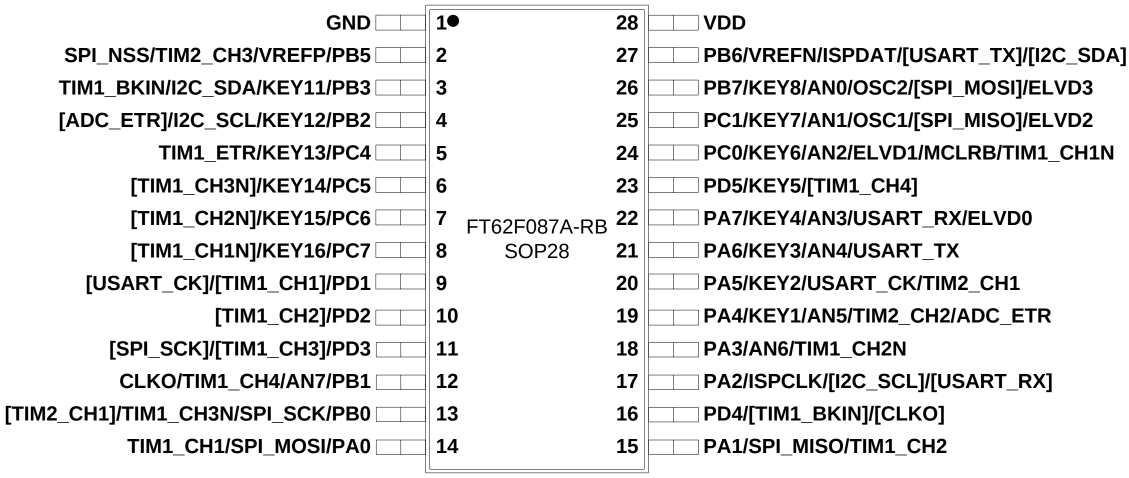

GND 1 28 VDD
SPI_NSS/TIM2_CH3/VREFP/PB5 2 27 PB6/VREFN/ISPDAT/[USART_TX]/[I2C_SDA]
TIM1_BKIN/I2C_SDA/KEY11/PB3 3 26 PB7/KEY8/AN0/OSC2/[SPI_MOSI]/ELVD3
[ADC_ETR]/I2C_SCL/KEY12/PB2 4 25 PC1/KEY7/AN1/OSC1/[SPI_MISO]/ELVD2
TIM1_ETR/KEY13/PC4 5 24 PC0/KEY6/AN2/ELVD1/MCLRB/TIM1_CH1N
[TIM1_CH3N]/KEY14/PC5 6 23 PD5/KEY5/[TIM1_CH4]
[TIM1_CH2N]/KEY15/PC6 7 22 PA7/KEY4/AN3/USART_RX/ELVD0
[TIM1_CH1N]/KEY16/PC7 8 21 PA6/KEY3/AN4/USART_TX
[USART_CK]/[TIM1_CH1]/PD1 9 20 PA5/KEY2/USART_CK/TIM2_CH1
[TIM1_CH2]/PD2 10 19 PA4/KEY1/AN5/TIM2_CH2/ADC_ETR
[SPI_SCK]/[TIM1_CH3]/PD3 11 18 PA3/AN6/TIM1_CH2N
CLKO/TIM1_CH4/AN7/PB1 12 17 PA2/ISPCLK/[I2C_SCL]/[USART_RX]
[TIM2_CH1]/TIM1_CH3N/SPI_SCK/PB0 13 16 PD4/[TIM1_BKIN]/[CLKO]
TIM1_CH1/SPI_MOSI/PA0 14 15 PA1/SPI_MISO/TIM1_CH2
FT62F087A-RB
SOP28

Figure 1-5 SOP28 (A) [1]

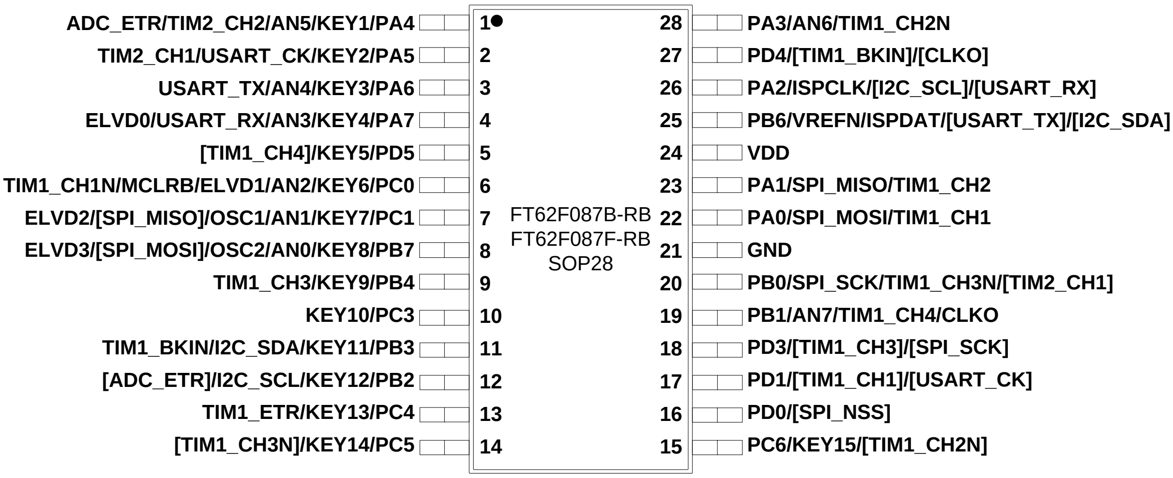

ADC_ETR/TIM2_CH2/AN5/KEY1/PA4 1 28 PA3/AN6/TIM1_CH2N
TIM2_CH1/USART_CK/KEY2/PA5 2 27 PD4/[TIM1_BKIN]/[CLKO]
USART_TX/AN4/KEY3/PA6 3 26 PA2/ISPCLK/[I2C_SCL]/[USART_RX]
ELVD0/USART_RX/AN3/KEY4/PA7 4 25 PB6/VREFN/ISPDAT/[USART_TX]/[I2C_SDA]
[TIM1_CH4]/KEY5/PD5 5 24 VDD
TIM1_CH1N/MCLRB/ELVD1/AN2/KEY6/PC0 6 23 PA1/SPI_MISO/TIM1_CH2
ELVD2/[SPI_MISO]/OSC1/AN1/KEY7/PC1 7 22 PA0/SPI_MOSI/TIM1_CH1
ELVD3/[SPI_MOSI]/OSC2/AN0/KEY8/PB7 8 21 GND
TIM1_CH3/KEY9/PB4 9 20 PB0/SPI_SCK/TIM1_CH3N/[TIM2_CH1]
KEY10/PC3 10 19 PB1/AN7/TIM1_CH4/CLKO
TIM1_BKIN/I2C_SDA/KEY11/PB3 11 18 PD3/[TIM1_CH3]/[SPI_SCK]
[ADC_ETR]/I2C_SCL/KEY12/PB2 12 17 PD1/[TIM1_CH1]/[USART_CK]
TIM1_ETR/KEY13/PC4 13 16 PD0/[SPI_NSS]
[TIM1_CH3N]/KEY14/PC5 14 15 PC6/KEY15/[TIM1_CH2N]
FT62F087B-RB
FT62F087F-RB
SOP28

Figure 1-6 SOP28 (B)

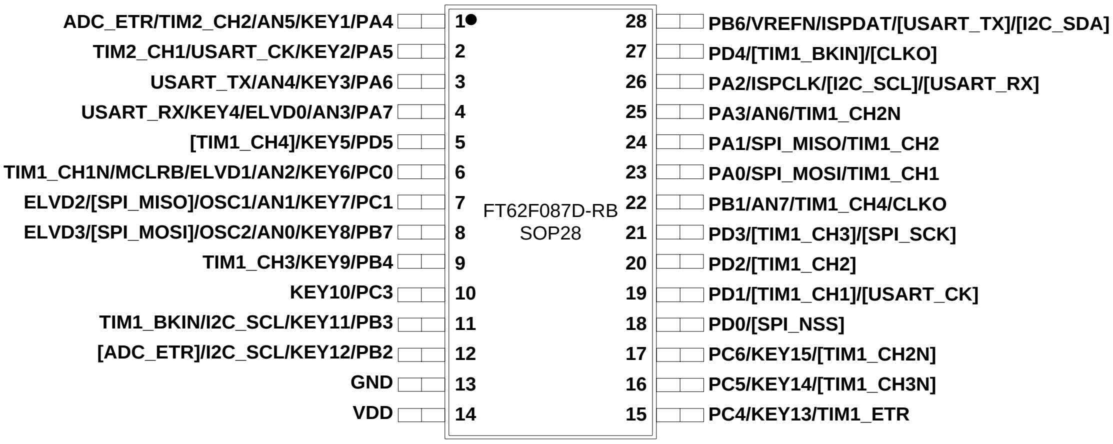

ADC_ETR/TIM2_CH2/AN5/KEY1/PA4 1 28 PB6/VREFN/ISPDAT/[USART_TX]/[I2C_SDA]
TIM2_CH1/USART_CK/KEY2/PA5 2 27 PD4/[TIM1_BKIN]/[CLKO]
USART_TX/AN4/KEY3/PA6 3 26 PA2/ISPCLK/[I2C_SCL]/[USART_RX]
USART_RX/KEY4/ELVD0/AN3/PA7 4 25 PA3/AN6/TIM1_CH2N
[TIM1_CH4]/KEY5/PD5 5 24 PA1/SPI_MISO/TIM1_CH2
TIM1_CH1N/MCLRB/ELVD1/AN2/KEY6/PC0 6 23 PA0/SPI_MOSI/TIM1_CH1
ELVD2/[SPI_MISO]/OSC1/AN1/KEY7/PC1 7 22 PB1/AN7/TIM1_CH4/CLKO
ELVD3/[SPI_MOSI]/OSC2/AN0/KEY8/PB7 8 21 PD3/[TIM1_CH3]/[SPI_SCK]
TIM1_CH3/KEY9/PB4 9 20 PD2/[TIM1_CH2]
KEY10/PC3 10 19 PD1/[TIM1_CH1]/[USART_CK]
TIM1_BKIN/I2C_SCL/KEY11/PB3 11 18 PD0/[SPI_NSS]
[ADC_ETR]/I2C_SCL/KEY12/PB2 12 17 PC6/KEY15/[TIM1_CH2N]
GND 13 16 PC5/KEY14/[TIM1_CH3N]
VDD 14 15 PC4/KEY13/TIM1_ETR
FT62F087D-RB
SOP28

Figure 1-7 SOP28 (C)

[1] When the TOUCH module is enabled, the pin 8 (PC7/KEY16) cannot be used by users, that is, pin 8 cannot be wired on the user's PCB or connect to any other device.

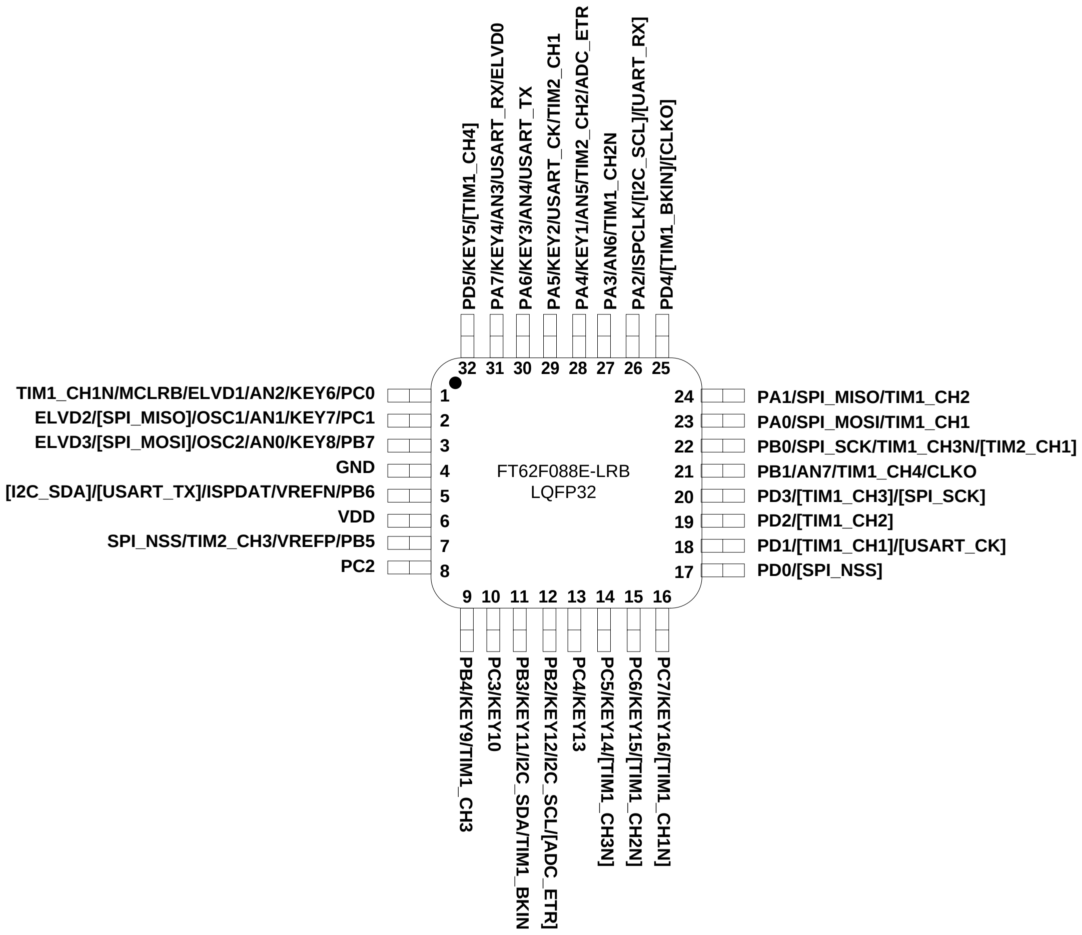

PD5/KEY5/[TIM1_CH4]
PA7/KEY4/AN3/USART_RX/ELVD0
PA6/KEY3/AN4/USART_TX
PA5/KEY2/USART_CK/TIM2_CH1
PA4/KEY1/AN5/TIM2_CH2/ADC_ETR
PA3/AN6/TIM1_CH2N
PA2/ISPCLK/[I2C_SCL]/[UART_RX]
PD4/[TIM1_BKIN]/[CLKO]
32 31 30 29 28 27 26 25
TIM1_CH1N/MCLRB/ELVD1/AN2/KEY6/PC0 1
ELVD2/[SPI_MISO]/OSC1/AN1/KEY7/PC1 2
ELVD3/[SPI_MOSI]/OSC2/AN0/KEY8/PB7 3
GND 4
[I2C_SDA]/[USART_TX]/ISPDAT/VREFN/PB6 5
VDD 6
SPI_NSS/TIM2_CH3/VREFP/PB5 7
PC2 8
24 PA1/SPI_MISO/TIM1_CH2
23 PA0/SPI_MOSI/TIM1_CH1
22 PB0/SPI_SCK/TIM1_CH3N/[TIM2_CH1]
21 PB1/AN7/TIM1_CH4/CLKO
20 PD3/[TIM1_CH3]/[SPI_SCK]
19 PD2/[TIM1_CH2]
18 PD1/[TIM1_CH1]/[USART_CK]
17 PD0/[SPI_NSS]
FT62F088E-LRB
LQFP32
9 10 11 12 13 14 15 16
PB4/KEY9/TIM1_CH3
PC3/KEY10
PB3/KEY11/I2C_SDA/TIM1_BKIN
PB2/KEY12/I2C_SCL/[ADC_ETR]
PC4/KEY13
PC5/KEY14/[TIM1_CH3N]
PC6/KEY15/[TIM1_CH2N]
PC7/KEY16/[TIM1_CH1N]

Figure 1-8 LQFP32 [2]

[2] When the TOUCH module is enabled, the pin 16 (PC7/KEY16) cannot be used by users, that is, pin 16 cannot be wired on the user's PCB or connect to any other device.

1.2. Pin Description by Functions

Function	Description	Name	GPIO equiv.	16 pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins
Power		VDD		16	10	24	28	24	14	6
		GND		1	9	1	1	21	13	4
GPIO	Pull-Up / Pull-Down, Digital Input, Digital Output	PD5			5		23	5	5	32
		PD4					16	27	27	25
		PD3			17	11	11	18	21	20
		PD2				10	10		20	19
		PD1			16	9	9	17	19	18
		PD0						16	18	17
		PC7					8			16
		PC6		8	14	8	7	15	17	15
		PC5		7		7	6	14	16	14
		PC4		6		6	5	13	15	13
		PC3		3				10	10	10
		PC2								8
		PC1		14	7	21	25	7	7	2
		PC0		13	6	20	24	6	6	1
		PB7			8	22	26	8	8	3
		PB6		15	20	23	27	25	28	5
		PB5			11	2	2			7
		PB4		2		3		9	9	9
		PB3		4	12	4	3	11	11	11
		PB2		5	13	5	4	12	12	12
		PB1		9	15	12	12	19	22	21
		PB0				13	13	20		22
		PA7		12	4	19	22	4	4	31
		PA6			3	18	21	3	3	30
		PA5		11	2		20	2	2	29
		PA4			1	17	19	1	1	28
		PA3					18	28	25	27
		PA2		10	18	16	17	26	26	26
		PA1			19	15	15	23	24	24
		PA0				14	14	22	23	23
ISPDebugger	ISP-Data	ISPDAT	PB6	15	20	23	27	25	28	5
	ISP-CLK	ISPCLK	PA2	10	18	16	17	26	26	26

Function	Description	Name	GPIO equiv.	16 pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins
External reset	Pull-Up	/MCLRB	PC0	13	6	20	24	6	6	1
LVD	Input	ELVD0	PA7	12	4	19	22	4	4	31
		ELVD1	PC0	13	6	20	24	6	6	1
		ELVD2	PC1	14	7	21	25	7	7	2
		ELVD3	PB7		8	22	26	8	8	3
Clock	Output	CLKO	PB1	9	15	12	12	19	22	21
		[CLKO]	PD4				16	27	27	25
	OSC+	OSC1	PC1	14	7	21	25	7	7	2
	OSC −	OSC2	PB7		8	22	26	8	8	3
Timer1 (Deadband)	PWM1	TIM1_CH1	PA0			14	14	22	23	23
		[TIM1_CH1]	PD1		16	9	9	17	19	18
	/PWM1	TIM1_CH1N	PC0	13	6	20	24	6	6	1
		[TIM1_CH1N]	PC7				8			16
	PWM2	TIM1_CH2	PA1		19	15	15	23	24	24
		[TIM1_CH2]	PD2			10	10		20	19
	/PWM2	TIM1_CH2N	PA3				18	28	25	27
		[TIM1_CH2N]	PC6	8	14	8	7	15	17	15
	PWM3	TIM1_CH3	PB4	2		3		9	9	9
		[TIM1_CH3]	PD3		17	11	11	18	21	20
	/PWM3	TIM1_CH3N	PB0			13	13	20		22
		[TIM1_CH3N]	PC5	7		7	6	14	16	14
	PWM4	TIM1_CH4	PB1	9	15	12	12	19	22	21
		[TIM1_CH4]	PD5		5		23	5	5	32
	PWM Fault-Break Input	TIM1_BKIN	PB3	4	12	4	3	11	11	11
		[TIM1_BKIN]	PD4				16	27	27	25
Timer2	PWM5	TIM2_CH1	PA5	11	2		20	2	2	29
		[TIM2_CH1]	PB0			13	13	20		22
	PWM6	TIM2_CH2	PA4		1	17	19	1	1	28
	PWM7	TIM2_CH3	PB5		11	2	2			7
ADC	Input	AN7	PB1	9	15	12	12	19	22	21
		AN6	PA3				18	28	25	27
		AN5	PA4		1	17	19	1	1	28
		AN4	PA6		3	18	21	3	3	30
		AN3	PA7	12	4	19	22	4	4	31
		AN2	PC0	13	6	20	24	6	6	1

Function	Description	Name	GPIO equiv.	16 pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins
		AN1	PC1	14	7	21	25	7	7	2
		AN0	PB7		8	22	26	8	8	3
	Trigger	ADC_ETR	PA4		1	17	19	1	1	28
		[ADC_ETR]	PB2	5	13	5	4	12	12	12
	V_{REF-}	VREFN	PB6	15	20	23	27	25	28	5
	V_{REF+}	VREFP	PB5		11	2	2			7
External Pin Interrupt	Low level, Rising edge, Falling edge, Double-edge	PD5			5		23	5	5	32
		PD4					16	27	27	25
		PD3			17	11	11	18	21	20
		PD2				10	10		20	19
		PD1			16	9	9	17	19	18
		PD0						16	18	17
		PC7					8			16
		PC6		8	14	8	7	15	17	15
		PC5		7		7	6	14	16	14
		PC4		6		6	5	13	15	13
		PC3		3				10	10	10
		PC2								8
		PC1		14	7	21	25	7	7	2
		PC0		13	6	20	24	6	6	1
		PB7			8	22	26	8	8	3
		PB6		15	20	23	27	25	28	5
		PB5			11	2	2			7
		PB4		2		3		9	9	9
		PB3		4	12	4	3	11	11	11
		PB2		5	13	5	4	12	12	12
		PB1		9	15	12	12	19	22	21
		PB0				13	13	20		22
		PA7		12	4	19	22	4	4	31
		PA6			3	18	21	3	3	30
		PA5		11	2		20	2	2	29
		PA4			1	17	19	1	1	28
		PA3					18	28	25	27
		PA2		10	18	16	17	26	26	26
		PA1			19	15	15	23	24	24
		PA0				14	14	22	23	23
SPI	SPI_MISO	SPI_MISO	PA1		19	15	15	23	24	24

Function	Description	Name	GPIO equiv.	16 pins	20 pins	24 pins	28(A) pins	28(B) pins	28(C) pins	32 pins
	(Open-Drain)	[SPI_MISO]	PC1	14	7	21	25	7	7	2
	SPI_MOSI (Open-Drain)	SPI_MOSI	PA0			14	14	22	23	23
		[SPI_MOSI]	PB7		8	22	26	8	8	3
	SPI_NSS	SPI_NSS	PB5		11	2	2			7
		[SPI_NSS]	PD0					16	18	17
	SPI_SCK	SPI_SCK	PB0			13	13	20		22
		[SPI_SCK]	PD3		17	11	11	18	21	20
I2C	I2C_Data (Open-Drain)	I2C_SDA	PB3	4	12	4	3	11	11	11
		[I2C_SDA]	PB6	15	20	23	27	25	28	5
	I2C_SCL (Open-Drain)	I2C_SCL	PB2	5	13	5	4	12	12	12
		[I2C_SCL]	PA2	10	18	16	17	26	26	26
USART	USART_CK	USART_CK	PA5	11	2		20	2	2	29
		[USART_CK]	PD1		16	9	9	17	19	18
	USART_TX (Open-Drain)	USART_TX	PA6		3	18	21	3	3	30
		[USART_TX]	PB6	15	20	23	27	25	28	5
	USART_RX	USART_RX	PA7	12	4	19	22	4	4	31
		[USART_RX]	PA2	10	18	16	17	26	26	26
TOUCH	Input	KEY1	PA4		1	17	19	1	1	28
		KEY2	PA5	11	2		20	2	2	29
		KEY3	PA6		3	18	21	3	3	30
		KEY4	PA7	12	4	19	22	4	4	31
		KEY5	PD5		5		23	5	5	32
		KEY6	PC0	13	6	20	24	6	6	1
		KEY7	PC1	14	7	21	25	7	7	2
		KEY8	PB7		8	22	26	8	8	3
		KEY9	PB4	2		3		9	9	9
		KEY10	PC3	3				10	10	10
		KEY11	PB3	4	12	4	3	11	11	11
		KEY12	PB2	5	13	5	4	12	12	12
		KEY13	PC4	6		6	5	13	15	13
		KEY14	PC5	7		7	6	14	16	14
		KEY15	PC6	8	14	8	7	15	17	15
		KEY16	PC7				8[1]			16[2]

Table 1-1 Pin description by functions

2. I/O PORTS

Up to 30 I/O pins are available depending on the types of package. I/O ports are divided into 4 groups: PORTA (8), PORTB (8), PORTC (8) and PORTD (6). Table 2-1 and Table 2-2 lists the functions of all I/O pins.

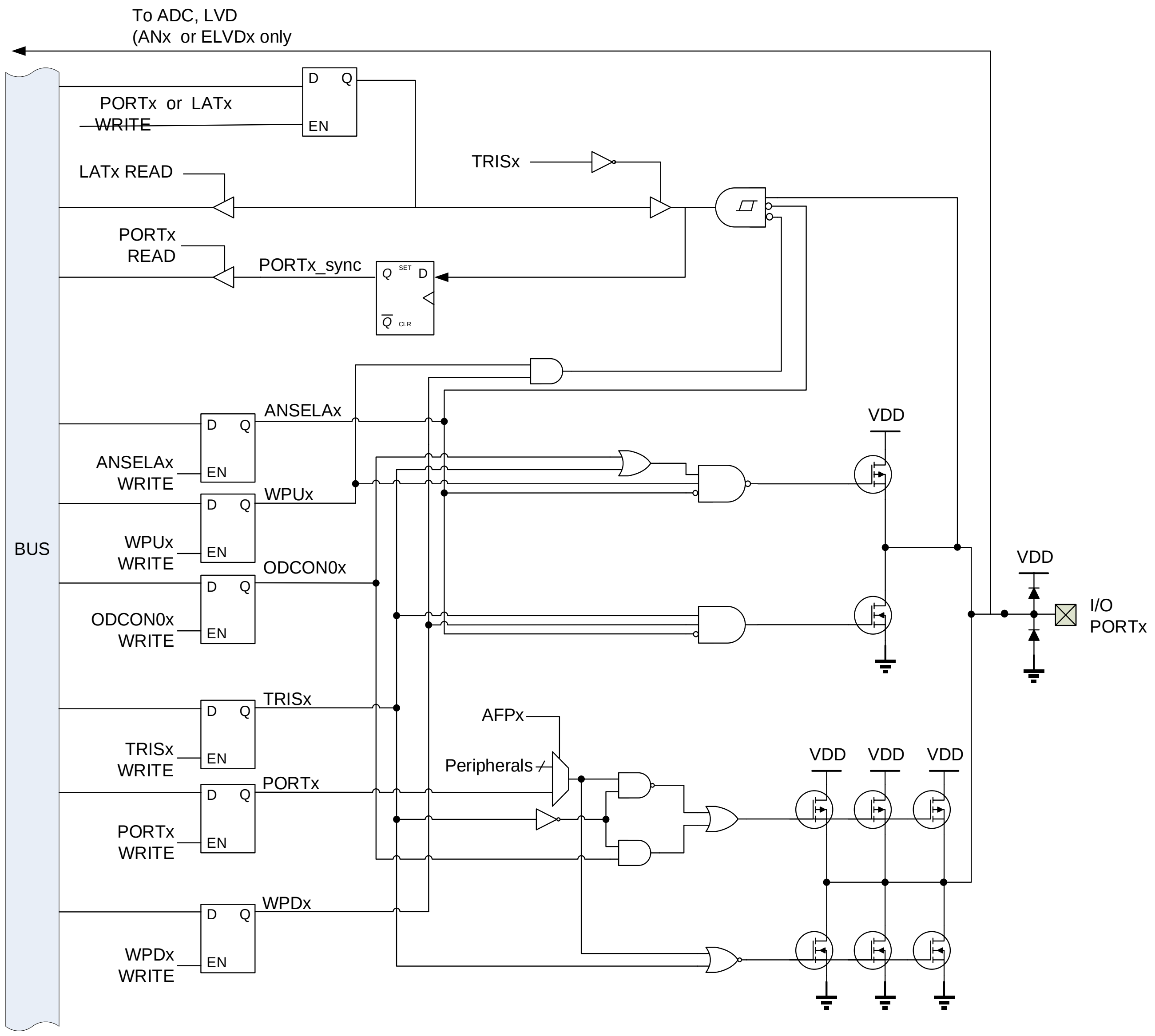


Figure 2-1 PORT Block Diagram

All I/O pins have the following functions (Table 2-3 ,Table 2-4):

- Digital Output
- Digital Input
- Open-Drain (SPI, I2C, USART corresponding PORTs)
- Weak Pull-Up
- Weak Pull-Down

In addition, some I/O’s have special functions assigned:

1. Burn debugger pins (ISP-Data, ISP-CLK) are hardware internal connection and require no set-up.
2. Some special functions are configured at the IDE and loaded during BOOT (**Table 2-8**):
    - External Clock/ Crystal Oscillator IN (OSC1, OSC2)
    - System External Reset (/MCLRB)
3. All other functions are Instruction Level assigned to the various I/O's. They are divided into 4 categories:

a) Digital Output

- PWM
- Internal Clock Output

b) Digital Input

- PWM Fault Break
- ADC trigger (ADC_ETR)
- GPIO Interrupt-on-Change

c) Analog Input

- LVD / BOR
- $V_{REF +}$
- ADC
- $V_{REF -}$
- TOUCH

d) Communication Interface

- SPI
- USART
- I2C

Name	ISP Debugger	CLK	Interrupt	PWM	Digital I/O Pull-Up / Pull-Down	Source Current (mA)	Sink Current (mA)
PA0			√	PWM1	√	2, 4, 14, 26	53, 62
PA1			√	PWM2	√	2, 4, 14, 26	53, 62
PA2	CLK		√		√	2, 4, 14, 26	53, 62
PA3			√	PWM2N	√	2, 4, 14, 26	53, 62
PA4			√	PWM6	√	2, 4, 14, 26	53, 62
PA5			√	PWM5	√	2, 4, 14, 26	53, 62
PA6			√		√	2, 4, 14, 26	53, 62
PA7			√		√	2, 4, 14, 26	53, 62
PB0			√	PWM3N	√	2, 4, 14, 26	53, 62
PB1		Output	√	PWM4	√	2, 4, 14, 26	53, 62
PB2			√		√	2, 4, 14, 26	53, 62
PB3			√	BKIN	√	2, 4, 14, 26	53, 62
PB4			√	PWM3	√	2, 4, 14, 26	53, 62
PB5			√	PWM7	√	2, 4, 14, 26	53, 62
PB6	DATA		√		√	2, 4, 14, 26	53, 62
PB7		OSC−	√		√	2, 4, 14, 26	53, 62
PC0			√	PWM1N	√	2, 4, 14, 26	53, 62
PC1		OSC+	√		√	2, 4, 14, 26	53, 62
PC2			√		√	2, 4, 14, 26	53, 62
PC3			√		√	2, 4, 14, 26	53, 62
PC4			√		√	2, 4, 14, 26	53, 62
PC5			√	[PWM3N]	√	2, 4, 14, 26	53, 62
PC6			√	[PWM2N]	√	2, 4, 14, 26	53, 62
PC7			√	[PWM1N]	√	2, 4, 14, 26	53, 62
PD0			√	[PWM1]	√	2, 4, 14, 26	53, 62
PD1			√		√	2, 4, 14, 26	53, 62
PD2			√	[PWM2]	√	2, 4, 14, 26	53, 62
PD3			√	[PWM3]	√	2, 4, 14, 26	53, 62
PD4		Output	√	[BKIN]	√	2, 4, 14, 26	53, 62
PD5			√	[PWM4]	√	2, 4, 14, 26	53, 62
Note			/MCLRB = PC0	[PWM5] = PB0		V_{DD}=5, V_{DS}=0.5	

Table 2-1 I/O PORT functions

Note: All IO have 4 configurable source current levels (see "PSRCx", **Table 2-4**) and 2 configurable sink current levels (see "PSINKx", **Table 2-4**).

Name	ADC	LVD	TOUCH	SPI	I2C	USART	Open-Drain
PA0				MOSI			√
PA1				MISO			√
PA2					[SCL]	[RX]	√
PA3	AN6						
PA4	AN5		KEY1				
PA5			KEY2			CK	
PA6	AN4		KEY3			TX	√
PA7	AN3	ELVD0	KEY4			RX	
PB0				SCK			
PB1	AN7						
PB2	Trigger		KEY12		SCL		√
PB3			KEY11		SDA		√
PB4			KEY9				
PB5	(V_{REF+})			NSS			
PB6	(V_{REF-})				[SDA]	[TX]	√
PB7	AN0	ELVD3	KEY8	[MOSI]			√
PC0	AN2	ELVD1	KEY6				
PC1	AN1	ELVD2	KEY7	[MISO]			√
PC2							
PC3			KEY10				
PC4			KEY13				
PC5			KEY14				
PC6			KEY15				
PC7			KEY16				
PD0				[NSS]			
PD1						[CK]	
PD2							
PD3				[SCK]			
PD4							
PD5			KEY5				
Notes	Trigger= PA4						

Table 2-2 I/O PORT functions (continued)

2.1. Summary of I/O PORT Related Registers

Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
ANSELA	0x197	ANSELA[7:0]								0000 0000
TRISA	0x8C	TRISA[7:0], PORTA Data Direction Register								1111 1111
TRISB	0x8D	TRISB[7:0], PORTB Data Direction Register								1111 1111
TRISC	0x8E	TRISC[7:0], PORTC Data Direction Register								1111 1111
TRISD	0X8F	–	–	TRISD[5:0], PORTD Data Direction Register						−−11 1111
PORTA	0x0C	PORTA Output Register								xxxx xxxx
PORTB	0x0D	PORTB Output Register								xxxx xxxx
PORTC	0x0E	PORTC Output Register								xxxx xxxx
PORTD	0x0F	–	–	PORTD[5:0] Output Register						−−xx xxxx
LATA	0x10C	PORTA Data Latch								xxxx xxxx
LATB	0x10D	PORTB Data Latch								xxxx xxxx
LATC	0x10E	PORTC Data Latch								xxxx xxxx
LATD	0x10F	–	–	PORTD[5:0] Data Latch						−−xx xxxx
WPUA	0x18C	PORTA Weak Pull-Up								0000 0000
WPUB	0x18D	PORTB Weak Pull-Up								0000 0000
WPUC	0x18E	PORTC Weak Pull-Up								0000 0000
WPUD	0X18F	–		PORTD[5:0] Weak Pull-Up						−−00 0000
WPDA	0x20C	PORTA Weak Pull-Down								0000 0000
WPDB	0x20D	PORTB Weak Pull-Down								0000 0000
WPDC	0x20E	PORTC Weak Pull-Down								0000 0000
WPDD	0x20F	–		PORTD[5:0] Weak Pull-Down						−−00 0000
ODCON0	0x21F	–	–	–	–	–	SPIOD	I2COD	UROD	−−−− −000
PSRC0	0x11A	PORTA[7:0], PORTB[7:0] Source Current Setting								1111 1111
PSRCB1	0x11B	PORTC[7:0], PORTD[5:0] Source Current Setting								1111 1111
PSINK0	0x19A	PORTA Sink Current Setting								0000 0000
PSINK1	0x19B	PORTB Sink Current Setting								0000 0000
PSINK2	0x19C	PORTC Sink Current Setting								0000 0000
PSINK3	0x19D	–	–	PORTD Sink Current Setting						−−00 0000
ITYPE0	0x11E	PORTx[3:0] (x = A, B, C, D) Ext. Pin Interrupt Type Setting								0000 0000
ITYPE1	0x11F	PORTD[5:4] and PORTx[7:4] (x = A, B, C) Ext. Pin Interrupt Type Setting								0000 0000
AFP0	0x19E	Pin Remapping Register 0								0000 0000
AFP1	0x19F	–	Pin Remapping Register 1							−000 0000
AFP2	0x11D	–	–	–	Pin Remapping Register 2					−−−0 0000
EPS0	0x118	External interrupt EINT3~0 pin selection								0000 0000
EPS1	0x119	External interrupt EINT7~4 pin selection								0000 0000
EPIE0	0x14	External pin interrupt enable								0000 0000
EPIF0	0x94	External pin interrupt Flag								0000 0000

Table 2-3 Addresses and Reset Values of I/O related registers

Name	Status		Register	Addr.	Reset
TRISA	PORTA	PORT Digital Output (Direction) 1 = Disable 0 = Enable (Disable Pull-Up/Down)	TRISA[7:0]	0x8C	RW−1111 1111
TRISB	PORTB		TRISB[7:0]	0x8D	RW−1111 1111
TRISC	PORTC		TRISC[7:0]	0x8E	RW−1111 1111
TRISD	PORTD		TRISD[5:0]	0x8F	RW−11 1111
ANSELA	1 = Disable Pull-Up /Down, and Digital Input (only for 8 ADC channels) 0 = (No action)		ANSELA[7:0]	0x197	RW−0000 0000
WPUA	PORTA	Weak Pull-Up 1 = Enable 0 = Disable	WPUA[7:0]	0x18C	RW−0000 0000
WPUB	PORTB		WPUB[7:0]	0x18D	RW−0000 0000
WPUC	PORTC		WPUC[7:0]	0x18E	RW−0000 0000
WPUD	PORTD		WPUD[5:0]	0x18F	RW−00 0000
WPDA	PORTA	Weak Pull-Down 1 = Enable 0 = Disable	WPDA[7:0]	0x20C	RW−0000 0000
WPDB	PORTB		WPDB[7:0]	0x20D	RW−0000 0000
WPDC	PORTC		WPDC[7:0]	0x20E	RW−0000 0000
WPDD	PORTD		WPDD[5:0]	0x20F	RW−00 0000
PORTA	PORTA	Data Output Register Read: Returns V_{IO} Write: Write to the corresponding LATx register	PORTA[7:0]	0x0C	RW−xxxx xxxx
PORTB	PORTB		PORTB[7:0]	0x0D	RW−xxxx xxxx
PORTC	PORTC		PORTC[7:0]	0x0E	RW−xxxx xxxx
PORTD	PORTD		PORTD[5:0]	0x0F	RW−xx xxxx
LATA	PORTA	Data output latch	LATA[7:0]	0x10C	RW−xxxx xxxx
LATB	PORTB		LATB[7:0]	0x10D	RW−xxxx xxxx
LATC	PORTC		LATC[7:0]	0x10E	RW−xxxx xxxx
LATD	PORTD		LATD[5:0]	0x10F	RW−xx xxxx
SPIOD	SPI_MISO, SPI_MOSI	Open-Drain 1 = Enable 0 = Disable	ODCON0[2]	0x21F	RW−0
I2COD	I2C_SDA, I2C_SCL		ODCON0[1]		RW−0
UROD	USART_TX		ODCON0[0]		RW−0
AFP0 [1]	USART_CK	1 = PD1 0 = PA5	AFP0[7]	0x19E	RW−0
	TIM1_CH1	1 = PD1 0 = PA0	AFP0[6]		RW−0
	SPI_NSS	1 = PD0 0 = PB5	AFP0[5]		RW−0
	TIM1_CH1N	1 = PC7 0 = PC0	AFP0[4]		RW−0
	TIM1_CH2N	1 = PC6 0 = PA3	AFP0[3]		RW−0
	TIM1_CH3N	1 = PC5 0 = PB0	AFP0[2]		RW−0
	ADC_ETR	1 = PB2 0 = PA4	AFP0[1]		RW−0
	I2C_SDA	1 = PB6 0 = PB3	A FP0[0]		RW−0

[1] Pin remapping selection .

Name	Status		Register	Addr.	Reset
AFP1 [1]	CLKO	1 = PD4 0 = PB1	AFP1[6]	0x19F	RW−0
	TIM1_CH4	1 = PD5 0 = PB1	AFP1[5]		RW−0
	I2C_SCL	1 = PA2 0 = PB2	AFP1[4]		RW−0
	TIM1_BKIN	1 = PD4 0 = PB3	AFP1[3]		RW−0
	TIM2_CH1	1 = PB0 0 = PA5	AFP1[2]		RW−0
	TIM1_CH3	1 = PD3 0 = PB4	AFP1[1]		RW−0
	TIM1_CH2	1 = PD2 0 = PA1	AFP1[0]		RW−0
AFP2 [1]	SPI_SCK	1 = PD3 0 = PB0	AFP2[4]	0x11D	RW−0
	SPI_MOSI	1 = PB7 0 = PA0	AFP2[3]		RW−0
	SPI_MISO	1 = PC1 0 = PA1	AFP2[2]		RW−0
	USART_RX	1 = PA2 0 = PA7	AFP2[1]		RW−0
	USART_TX	1 = PB6 0 = PA6	AFP2[0]		RW−0
PSINK0	PA7−PA0	Sink Current (mA) 1 = 62 0 = 53	PSINK0[7:0]	0x19A	RW−0000 0000
PSINK1	PB7−PB0		PSINK1[7:0]	0x19B	RW−0000 0000
PSINK2	PC7−PC0		PSINK2[7:0]	0x19C	RW−0000 0000
PSINK3	PD5−PD0		PSINK3[5:0]	0x19D	RW−00 0000
PSRCB[3:2]	PB7−PB4	Source Current (mA) 00 = 2 01 = 4 10 = 14 11 = 26	PSRC0[7:6]	0x11A	RW−11
PSRCB[1:0]	PB3−PB0		PSRC0[5:4]		RW−11
PSRCA[3:2]	PA7−PA4		PSRC0[3:2]		RW−11
PSRCA[1:0]	PA3−PA0		PSRC0[1:0]		RW−11
PSRCD[3:2]	PD5−PD4		PSRC1[7:6]	0x11B	RW−11
PSRCD[1:0]	PD3−PD0		PSRC1[5:4]		RW−11
PSRCC[3:2]	PC7−PC4		PSRC1[3:2]		RW−11
PSRCC[1:0]	PC3−PC0		PSRC1[1:0]		RW−11

Table 2-4 Instruction Level I/O related registers

Name	Status		Register	Addr.	Reset
ITYPE0[1:0]	PORTx.0	External interrupt pin EINTx trigger type 00 = Low level 01 = Rising edge 10 = Falling edge 11 = Double edge	ITYPE0[1:0]	0x11E	RW−00
ITYPE0[3:2]	PORTx.1		ITYPE0[3:2]		RW−00
ITYPE0[5:4]	PORTx.2		ITYPE0[5:4]		RW−00
ITYPE0[7:6]	PORTx.3		ITYPE0[7:6]		RW−00
ITYPE1[1:0]	PORTx.4		ITYPE1[1:0]	0x11F	RW−00
ITYPE1[3:2]	PORTx.5		ITYPE1[3:2]		RW−00
ITYPE1[5:4]	PORTy.6		ITYPE1[5:4]		RW−00
ITYPE1[7:6]	PORTy.7		ITYPE1[7:6]		RW−00

Table 2-5 External Pin Interrupt Trigger Type Register (x = A, B, C, D; y = A, B, C)

Name	Status				Register	Addr.	Reset
EINT0	00 = PA0	01 = PB0	10 = PC0	11 = PD0	EPS0[1:0]	0x118	RW−00
EINT1	00 = PA1	01 = PB1	10 = PC1	11 = PD1	EPS0[3:2]		RW−00
EINT2	00 = PA2	01 = PB2	10 = PC2	11 = PD2	EPS0[5:4]		RW−00
EINT3	00 = PA3	01 = PB3	10 = PC3	11 = PD3	EPS0[7:6]		RW−00
EINT4	00 = PA4	01 = PB4	10 = PC4	11 = PD4	EPS1[1:0]	0x119	RW−00
EINT5	00 = PA5	01 = PB5	10 = PC5	11 = PD5	EPS1[3:2]		RW−00
EINT6	00 = PA6	01 = PB6	10 = PC6	11 Reserved	EPS1[5:4]		RW−00
EINT7	00 = PA7	01 = PB7	10 = PC7	11 Reserved	EPS1[7:6]		RW−00

Table 2-6 External Interrupt Pin Selection Register

Name	Status		Register	Addr.	Reset
EPIE0x	External pin interrupt	1 = Enable 0 = Disable	EPIE0[7:0]	0x94	RW−00000000
EPIF0x [2]	External pin interrupt Flag	1 = Yes (latched) 0 = No	EPIF0[7:0]	0x14	R_W1C−00000000

Table 2-7 External Pin Interrupt Enable and Flag Registers

[2] Write '1' to clear, and writing '0' has no effect on the bit value. It is recommended to only use the STR and MOVWI instructions for write operations, rather than the BSR or IOR instructions .

Name	Function	default
MCLRE	External I/O reset	closure
FOSC	• LP external oscillator across PC1 (+) and PB7 (−) • XT external oscillator across PC1 (+) and PB7 (−) • EC external oscillator at PC1 (+), PB7 as I/O • INTOSCIO : PC1 and PB7 as I/O	INTOSCIO

I2CRMAP	I2C Remap pin selection
	[PB3, PB2] : (≥ VerI chips apply)
	I2C_SDA = PB3 , I2C_SCL = PB2,
	SPI_MOSI = PA0 , SPI_MISO = PA1
	[PA0, PA1]:
	I2C_SDA = PA0 , I2C_SCL = PA1,
	SPI_MOSI = PB3 , SPI_MISO = PB2
		[PB3, PB2]

Table 2-8 BOOT Level I/O Related Configuration Registers

2.2. Configuring the I/O

For each PORT, configures the following 5 modules according to their functions (**Table 2-4**):

- Digital Output
- Digital Input
- Open-Drain
- Weak Pull-Up
- Weak Pull-Down

Function	Digital Input	Pull-Up / Pull-Down	Digital Output	Settings
ISP-DATA	On	Off	On	(Built in hardware, ignore instructions)
ISP-CLK	On	Off	Off	(Built in hardware, ignore instructions)
/MCLRB	On	Pull-Up	Off	(initialize configuration, ignore instructions)
OSC+ (EC)	On	(optional)	Off	(initialize configuration, ignore instructions)
OSC+ / OSC − (LP, XT)	Off	Off	Off	(initialize configuration, ignore instructions)
ADC	Off	Off	Off	TRISx = 1; ANSELAx = 1
TOUCH	Off	Off	Off	TRISx = 1
LVD	Off [(5)]	Off	Off	TRISx = 1; ANSELAx = 1
V_{REF+} / V_{REF-}	Off	Off	Off	TRISx = 1
ADC trigger	On	(optional)	Off	TRISx = 1
SPI Input	On	(optional)	Off	TRISx = 1
I2C Input	On	(optional)	Off	TRISx = 1
USART Input	On	(optional)	Off	TRISx = 1
External pin interrupt	On	(optional)	Off	TRISx = 1
BKIN	On	(optional)	Off	TRISx = 1
Digital Input	On	(optional)	Off	TRISx = 1
Clock Output	(ignore)	Off	On	TRISx = 0
PWM	On	Off	On	TRISx = 0
Digital Output	On	Off	On	TRISx = 0
SPI Output	On	Off	On	TRISx = 0
I2C Output	On	Off	On	TRISx = 0
USART Output	On	Off	On	TRISx = 0

Table 2-9 Instruction Level I/O Configuration Flags and Registers

Note:

1. TRISx = 0: "Digital Output" is enabled, "Pull-Up/Pull-Down" is automatically disabled (WPDx, WPUx are ignored).
2. TRISx = 1: "Digital Output" is disabled.

3. ANSELAx = 1: "Pull-Up", "Pull-Down", "Digital Input" are automatically disabled (WPDx, WPUx are ignored).
4. The only instruction that can disable the "Digital Input" is "ANSELAx = 1".
5. When a PORT is set as an LVD input, its "Digital Input", "Pull-Up" and "Pull-Down" functions are automatically disabled. When the LVD input needs to be switched between different channels, the “Digital Input ” of the currently unselected channel can be disabled by setting “ANSELAx = 1” .
6. /MCLR enabled: The Weak Pull-Up function of PC0 is automatically enabled (ignore WPUC[0]); read PORTC[0] as "0".
7. Write to the PORTx Data Output Register or the LATx data latch, and the I/O PORT will output the corresponding logic level. Each group of up to 8 I/O data registers share the common address, and the write operation actually performs a "Read-Modify-Write" operation, that is, first read the PORTx latch value (output or input) or LATx of the group data latches, then modified, and finally written back to the PORTx/LATx data registers..

   Read and write to the LATx data output latches needn’t to wait. Reading PORTx returns the value of the pin after passing through the synchronization register. In 1T speed mode, after writing to the PORTx register, at least one synchronous SYSCLK is required before the correct PORTx value can be read (2T/4T mode without waiting).

   In 1T speed mode, when continuous bit operations are performed on PORTx, a NOP needs to be inserted in the middle of the write operation:

   ```
   BSR     PORTx, n        ;    1 for nth position of PORTx
   NOP                     ;    insert NOP and wait for
   BSR     PORTx, m        ;    1 for mth position of PORTx
   ```
8. Digital Output and Digital Input functions can coexist, and some applications require both Digital Output and Digital Input to be enabled.
9. ODCON0x = 1: The mapped pins(see "AFPx") selected by " SPI_MISO , SPI_MOSI", "I2C_SCL, I2C_SDA", "USART_TX" function as open-drain output. The open-drain and internal Pull-Up functions can be turned on at the same time .
10. On a full reset or system reset, the PORTx registers will not be reset, but TRISx will be reset to '1', and turn off the output.

For the setting of external pin interrupt, please refer to **Section 9** "Interrupts".

2.3. Pin Out Priority

Each I/O pin is multiplexed with multiple functions. When the corresponding module enables the output, the output priority from low to high is shown in **Table 2-10**. Since the inputs are connected to individual function modules, there is no priority issue with the inputs.

Name	Priority 0	Priority 1	Priority 2	Priority 3
PA0	PA0	SPI_MISO	TIM1_CH1	–
PA1	PA1	SPI_MOSI	TIM1_CH2	–
PA2	PA2	[USART_RX]	[I2C_SCL]	ISPCLK
PA3	PA3	TIM1_CH2N	–	–
PA4	PA4	TIM2_CH2	–	–
PA5	PA5	USART_CK	TIM2_CH1	–
PA6	PA6	USART_TX	–	–
PA7	PA7	–	–	–
PB0	PB0	SPI_SCK	TIM2_CH1	TIM1_CH3N
PB1	PB1	TIM1_CH4	CLKO	–
PB2	PB2	I2C_SCL	–	–
PB3	PB3	I2C_SDA	–	–
PB4	PB4	TIM1_CH3	–	–
PB5	PB5	SPI_NSS	TIM2_CH3	–
PB6	PB6	USART_TX	I2C_SDA	ISPDAT
PB7	PB7	SPI_MOSI	OSC2	–
PC0	PC0	TIM1_CH1N	MCLRB	–
PC1	PC1	SPI_MISO	OSC1	–
PC2	PC2	–	–	–
PC3	PC3	–	–	–
PC4	PC4	–	–	–
PC5	PC5	TIM1_CH3N	–	–
PC6	PC6	TIM1_CH2N	–	–
PC7	PC7	TIM1_CH1N	–	–
PD0	PD0	SPI_NSS	–	–
PD1	PD1	USART_CK	TIM1_CH1	–
PD2	PD2	TIM1_CH2	–	–
PD3	PD3	[SPI_SCK]	TIM1_CH3	–
PD4	PD4	[CLKO]	–	–
PD5	PD5	TIM1_CH4	–	–

Table 2-10 PinOut Priority

3. POWER-ON-RESET (POR)

During Power-On, V_{DD} increases from below the Power-On-Reset Voltage (V_{POR}) to above V_{POR}. V_{DD} may not have completely discharged to 0V when the CPU is Power-On again.

1. The CPU is in a Full-Reset state when V_{DD} is below V_{POR}.
    a. All calibrated configuration registers are not reset. Special Function Registers (SFR) are in Reset, except INDFx, Z, DC, C, FSRxL/H, BSREG , WREG , PORTx, LATx, OSCTUNE , EEDATL , EEDATH , E =ECON2 and SRAM (see **Section 17** “Special Function Registers”). Registers not reset, such as SRAM, will hold their values until V_{DD} drops below 0.6V (typical). Data of those resisters with V_{DD} below 0.6V are undetermined.
    b. Program Counter = 0x00, Instruction Register = “NOP”, Stack Pointer = “TOS” (Top of Stack).
2. BOOT commences when V_{DD} raises above V_{POR}.
3. Instruction execution begins with Program Counter = 0x00 after BOOT completion.

V_{POR} is ~1.6V at 25°C (typical), increasing to ~1.9V at -40°C. For $V_{DD} \geq V_{POR}$, the CPU can function at a reduced speed of 8 MHz / 2T, which giving a self-regulated wider V_{DD} operating range with temperature. This is important for battery-powered system as the CPU can function down even to ~1.6V at typical battery operating environments, greatly extending useful battery life.

Notes:

1. V_{POR} is not configurable.
2. The POR circuit is always on and will perform a Power-On-Reset any time V_{DD} voltage is below V_{POR}, not just during Power-On.

3.1. BOOT Sequence

Name	Functions	default
PWRTEB	Power-Up Delay Timer, ~64ms delay after BOOT load	disabled

Table 3-1 BOOT configurations

The BOOT configuration is as above. Their values are set at the IDE, not by instructions. during BOOT:

1. CPU idles for ~4ms.
2. The BOOT Level registers are loaded from the non-volatile memory. It takes ~39us. These registers are pre-set at the IDE and not affected by instructions.
3. If Power-Up Delay Timer (PWRT) is enabled, the CPU will idle for ~64ms.

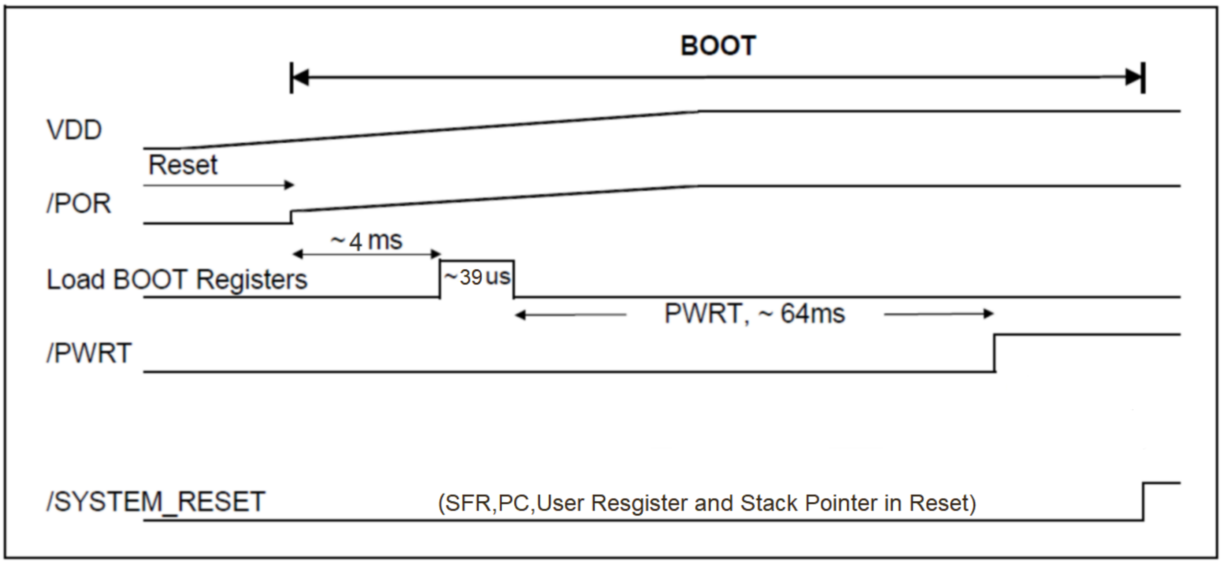


Figure 3-1 Power-On Sequence (PWRT enabled)

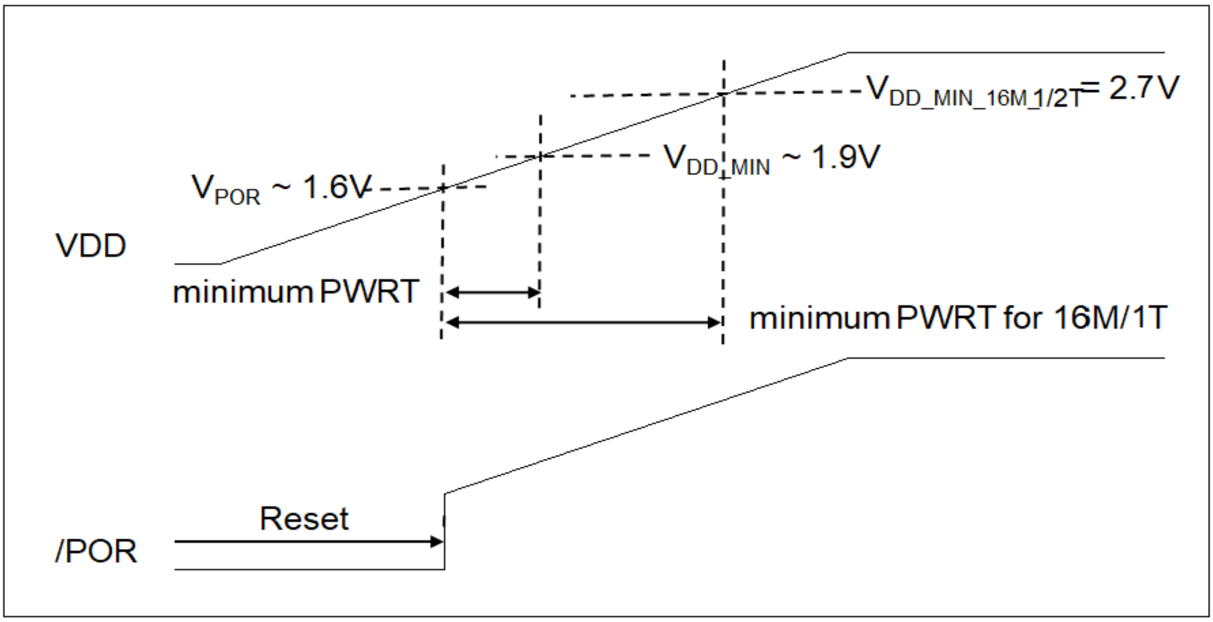


Figure 3-2 Minimum required PWRT during Power-On

V_{DD} must be higher than 2.7V by the end of BOOT if the CPU is to run at 16MHz / 1T. The total BOOT time can increase from ~4ms to ~68ms by enabling the PWRT, giving more time for the power system to stabilize.

Enable LVR with V_{BOR} ≥ 2.7V for operation at 16MHz / 1T. In addition, the frequency of LVR enable can be set to instruction controlled to monitor V_{DD} sporadically, instead of always on (see “LVREN”, “SLVREN”) to reduce power consumption.

Notes:

1. V_{DD} should not rise too slowly. C_{VDD} ≥ 22 μF is discouraged.
2. V_{DD} capacitor of 1 to 10μF is preferred. C_{VDD} < 1μF capacitor may be too small for EFT considerations.
3. If a delay in startup is acceptable, enables PWRT to improve CPU stability.

4. SYSTEM-RESET

System-Reset differs from POR in that it is not a Full-Reset. Depending on the reset trigger type, CPU whether or not BOOT. BOOT will wait ~4ms, reload the BOOT registers, and further delay system start by ~64ms if PWRT is enabled. In a System-Reset:

- Registers which reset in POR are reset in system-reset, except BOOT registers.
- Program Counter = 0x00, Instruction Register = “NOP”, Stack Pointer = “TOS” (Top of Stack).

The following 7 events besides debugger OCD can be configured to trigger a System-Reset:

1. Brown-Out (BOR / LVR) – always BOOT.
2. Illegal Instructions Reset.
3. Watch-Dog Reset (WDT , CPU not in SLEEP).
4. EMC Reset – always BOOT.
5. Software Reset (execute instruction "RESET").
6. Stack Overflow/Underflow Reset.
7. External I/O (/MCLRB) – BOOT if “MRBTE” is set. (≥ VerB chip) .

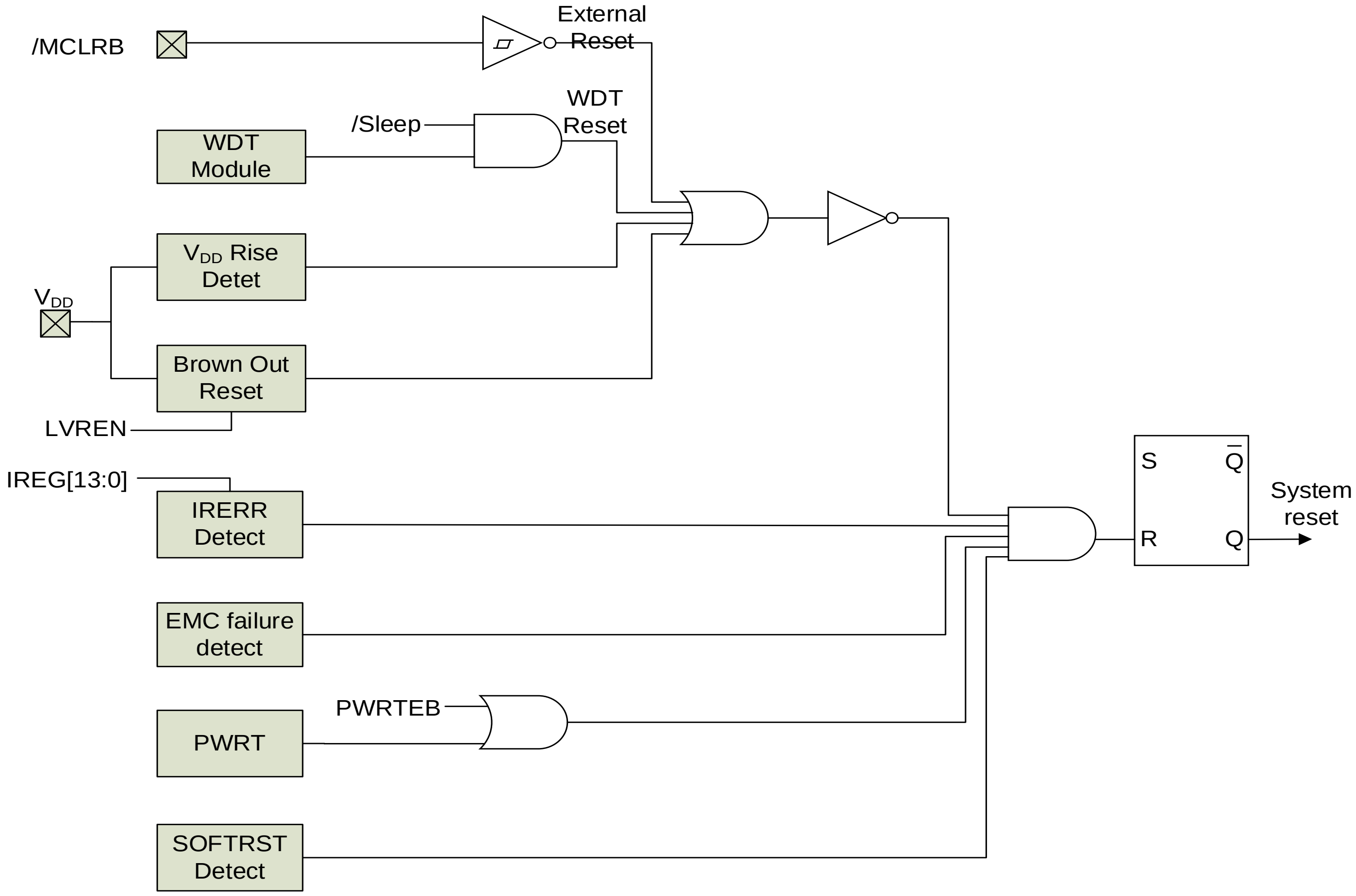


Figure 4-1 Reset circuit Block Diagram

4.1. Summary of SYSTEM-RESET Related Registers

Most settings for System-Reset are configured at the IDE, and cannot be changed by instructions.

Name	Function	default
LVRS	7 V_{BOR} Voltage levels (V): 2.0 / 2.2 / 2.5 / 2.8 / 3.1 / 3.6 / 4.1	2.5
LVREN	LVR • Enabled • Disabled • Enabled except in SLEEP • Instruction controlled (SLVREN)	disabled
WDTE	WDT • Enabled (Instructions can not be disabled) • Instruction controlled (SWDTEN)	SWDTEN control
MCLRE	Reset by External I/O	disabled

Table 4-1 BOOT Level RESET related configurations

4.2. Brown-Out Reset (LVR / BOR)

Brown-Out occurs when V_{DD} falls below a pre-configured Brown-Out Voltage (V_{BOR}) for a time longer than T_{BOR}. T_{BOR} takes 3 to 4 LIRC clock cycles (~94 – 125μs, LIRC will turn on automatically if not already). CPU System-Reset as long as $V_{DD} \le V_{BOR}$. Once $V_{DD} > V_{BOR}$ CPU will BOOT. The BORF will be set to 0.

While V_{POR} is fixed, V_{BOR} can be set to 2.0, 2.2, 2.5, 2.8, 3.1, 3.6, 4.1V (see "LVRS" in **Table 4-1**).

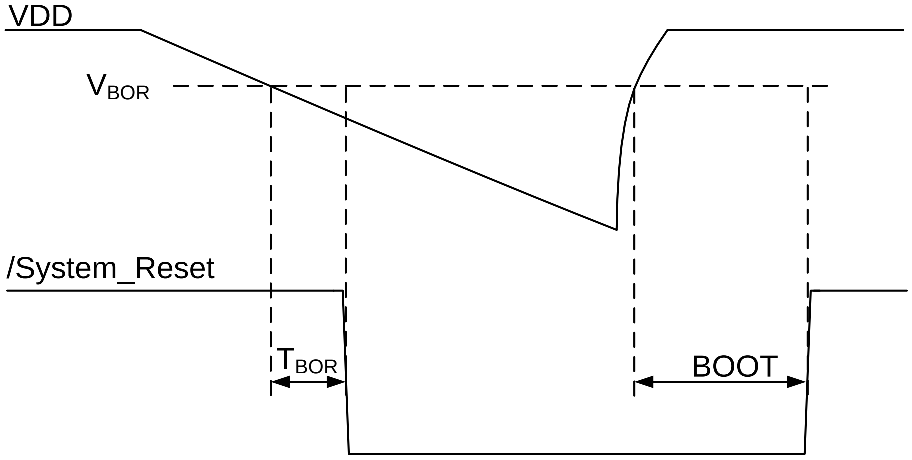


Figure 4-2 LVR BOOT Timing Diagram

LVR function can have four different settings configured (see "LVREN" in **Table 4-1**).

1. LVR enabled.
2. LVR disabled.

3. LVR enabled, except in SLEEP.

4. Let instructions enable or disable LVR (SLVREN, see **Table 4-2**).

Note: LVR can be instructions disabled in SLEEP to reduce power consumption. The CPU should wake up and enable LVR periodically to monitor V_{DD} if system V_{DD} is unstable.

Name	Status	Register	Addr.	Reset
SLVREN [1]	Only applicable to LVREN configured to control LVR by the SLVREN 1 = Enable LVR 0 = Disable LVR	LVDCON[7]	0x199	RW-0

Table 4-2 Instruction Level LVR registers

4.3. Illegal Instruction Reset

There are many reasons for CPU fetch instruction errors, and the most common reasons are interference and V_{DD} instability.

Although there is no dedicated Reset instruction, any deliberate illegal instruction is equivalent to a Reset instruction. BOOT after an Illegal Instruction and the flag IERRF will be set to 1 .

4.4. Software Reset

When the program executes Software Reset instruction "RESET", a system reset is generated and the flag /SRSTF will be set to 0.

4.5. Stack Overflow/Underflow Reset

The Stack Overflow or Underflow Reset (by configuring " STVREN") will trigger a System-Reset, and the Overflow Flag STKOVF or Underflow Flag STKUNF will be set to 1.

4.6. EMC Reset

The EMC detection module is always on. When some kind of EMC interference occurs, System-Reset and BOOT generate, and the Flag EMCF will be set to 1.

4.7. Watch Dog Timer (WDT) Reset

WDT overflows during SLEEP will result in a Wake-Up.

In normal mode (not SLEEP mode), a WDT overflow will trigger a System-Reset. And WDT reset can be used to reset a hung CPU. Clear WDT from time to time in the program to avoid false reset.

For details on WDT operation and setting see **Section 7.1** Watch Dog Timer (WDT).

[1] This bit will not be cleared when a Brown-out Reset occurs. Other resets will clear this bit .

4.8. External I/O System-Reset /MCLRB

The CPU can be reset by a low voltage applied to the /MCLRB (PC0) pin if so configured by BOOT. The /MCLRB pin is usually weak pullup to V_{DD} with a resistor instead of directly, as shown in **Figure 4-3**. The external RC network also provides faults filtering and over-current protection.

For ≥ VerI chips, BOOT after a /MCLR System-Reset.

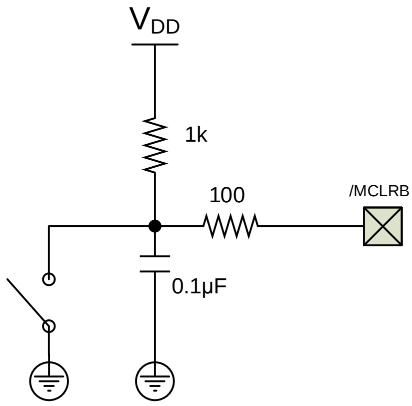


Figure 4-3 /MCLRB reset circuit

4.9. Detect the Type of Last Reset

Eight status flags in the PCON register and the different combinations of Time Out (/TO) and Power Down (/PD) can trace the type of last System-Reset. /BORF should be set to 1 by instructions, and will be latches to "0" after the Reset.

Name	Status		Register	Addr.	Reset
STKOVF [2]	Stack Overflows Flag	1 = Yes 0 = No, or cleared	PCON[7]	0x96	RW0−0
STKUNF [2]	Stack Underflows Flag		PCON[6]		RW0−0
EMCF [2]	EMC Reset Flag		PCON[5]		RW0−0
IERRF [2]	Illegal Instruction Flag		PCON[4]		RW0−0
/MCLRF [3]	External Reset Flag	1 = No (Software set 1) 0 = Yes (latched)	PCON[3]		RW1−1
/SRSTF [3]	Software Reset Flag		PCON[2]		RW1−1
/PORF [3]	Power-On-Reset Flag		PCON[1]		RW1−0
/BORF [3]	Low-voltage Reset Flag		PCON[0]		RW1−x

Table 4-3 Reset Flag Register

[2] Write '0' to clear, and writing '1' has no effect on the bit value.

[3] Only '1' can be written, and writing '0' has no effect on the bit value..

Reset Source	STKOVF	STKUNF	EMCF	IERRF	/MCLRF	/SRSTF	/PORF	/BORF	/TO	/PD
	PCON[7]	PCON[6]	PCON[5]	PCON[4]	PCON[3]	PCON[2]	PCON[1]	PCON[0]	STATUS[4]	STATUS[3]
	0x96								Bank first address + 0x03	
POR	0	0	0	0	1	1	0	1	1	1
LVR	0	0	0	0	1	1	1	0	–	–
CLRWDT Instructions	–	–	–	–	–	–	–	–	1	1
SLEEP Instructions	–	–	–	–	–	–	–	–	1	0
WDT overflows while not in SLEEP (Reset)	–	–	–	–	–	–	–	–	0	–
WDT overflows while in SLEEP (Wake up)	–	–	–	–	–	–	–	–	0	0
Software Reset	–	–	–	–	–	0	–	–	–	–
MCLR Reset (≥ verI)	0	0	0	0	1	1	0	1	1	1
MCLR Reset (< verI)	–	–	–	–	0	–	`–	–	–	–
Illegal Instruction Reset	–	–	–	1	–	–	–	–	–	–
EMC Reset	–	–	1	–	–	–	–	–	–	–
Stack Underflow Reset	–	1	–	–	–	–	–	–	–	–
Stack Over flow Reset	1	–	–	–	–	–	–	–	–	–
On-Chip Debugger (OCD)	–	–	–	–	–	–	–	–	–	–

Table 4-4 Reset Related Status Flags ("–" no change)

5. LOW VOLTAGE DETECT / COMPARATOR (LVD)

LVD works similarly to a LVR except for the followings:

- All control and setting parameters are set by instructions not by BOOT.
- I/O must be set appropriately: TRISx = 1; ANSELAx = 1.
- LVD event will set LVDW instead of /BOR.
- It can be instructions configured to Interrupt. It will not trigger System-Reset.
- Debouncing Time (T_{LVD}) is 3 – 4 LIRC cycles (LIRC turns on automatically if not already so).
- The input to the LVD module can be configured to V_{DD} or other 4 I/O. The latter allows the LVD to function as a single input comparator to one of the six LVDL levels($V_{LVD-REF}$).
- The External Reset MCLRB of PC0 has higher priority than ELVD. When configured as an External Reset pin, the ELVD detection is inactive.

5.1. Summary of LVD Related Registers

Name	Status			Register	Addr.	Reset
LVDM	LVD Input	1 = External pin (ELVDx) 0 = VDD		LVDCON0[6]	0x199	RW−0
LVDEN	LVD	1 = Enable	0 = Disable	LVDCON0[4]		RW−0
LVDW	LVD triggred Flag	When LVDM = 1 (ELVDx): 1 = Detection voltage > $V_{LVD-REF}$ (no latch) 0 = Detection voltage <$V_{LVD-REF}$ When LVDM = 0 (VDD) : 1 = Detection voltage < $V_{LVD-REF}$ (no latch) 0 = Detection voltage > $V_{LVD-REF}$		LVDCON[3]		RO−x
LVDL	$V_{LVD-REF}$	000 = Reserved 001 = Reserved 010 = 2.0 011 = 2.4	100 = 2.8 101 = 3.0 110 = 3.6 111 = 4.0	LVDCON[2:0]		RW−000
ELVDS	LVD External Input Pin Selection	00 = ELVD0 01 = ELVD1	10 = ELVD2 11 = ELVD3	ADCON3[1:0]	0x41A	RW−00
LVDIE	LVD Interrupt	1 = Enable	0 = Disable	INTCON[4]	0x0B	RW−0
LVDIF [1]	LVD Interrupt Flag	1 = Yes	0 = No, or cleared	INTCON[1]		R_W1C−0

Table 5-1 Instruction Level LVD Settings and Flags

[1] Write '1' to clear, and writing '0' has no effect on the bit value. It is recommended that only STR and MOVWI instructions be used for write operations, not BSR or IOR instructions.

6. OSCILLATORS and SYSCLK

Instruction chooses whether SysClk is the internal oscillator HIRC, internal oscillator LIRC, or one of the three external oscillators (EC, LP, XT, see “SCS” in **Table 6-2**). If external oscillator is chosen, BOOT level “FOSC” (**Table 6-1**) will determine which one of the three external oscillators is used. Instructions also select the frequency step down divider for internal oscillator (see MCKCF in **Table 6-2**). SysClk is used to generate the Instruction Clock:

Instruction Clock = SysClk / N; N = 2 for 2T, 4 for 4T.

The pin assignments for external clock inputs are set by BOOT (see FOSC). Eight clock sources can be selected for output by instructions (see "CCOSEL" and "CCOEN"). When the clock output is enabled, the selected clock source is automatically turned on, the flag CCORDY is set to 1, and the pinout can be selected as PD4 or PB1 (see " AFP1[6]", **Table 6-3**).

Peripheral Timers, ADC, I2C, TOUCH, SPI and USARTx have independent module SYSCLK control bit (see "PCKEN"). The module clock needs to be enabled before the corresponding module can be enabled. When shutting down, turn off the module function before the module colck can be turned off. In addition, the count clock source of Timers and the conversion clock source of ADC have independent oscillators. When both the module system clock and the module function are enabled, the selected oscillator will turn on automatically and remain effective during the operation of the module. Therefore, multiple oscillators can operate simultaneously.

As instructions are halted in SLEEP, so will Instruction Clock by default, and the clock output is suspended. When SYSON=1, the instruction clock will keep active, so the corresponding peripherals of the enable module system clock will also keep active in SLEEP, and the clock output will continue.

Note:

1. When SYSON = 1 and TIMxEN = 1 in SLEEP, the count clock source selected by Timers will keep active.
2. When LIRC is selected as the ADC conversion clock source, LIRC will remain active after entering SLEEP, independent of SYSON.
3. It is recommended to turn off the clock modules that do not use peripherals to reduce power consumption.

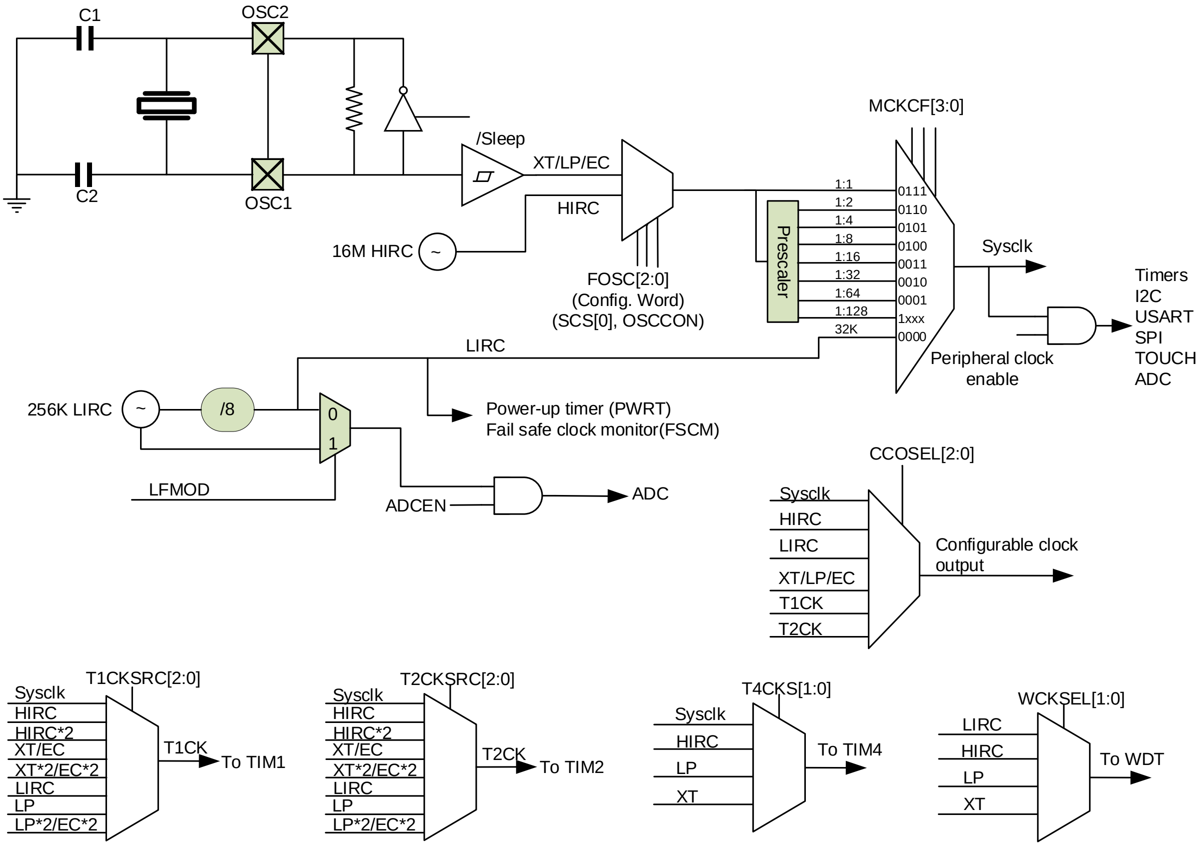


Figure 6-1 Clock source Block Diagram for SysClk

6.1. Summary of Oscillator Modules Related Registers

Name	Functions		default
FOSC	• LP: external low-speed oscillator across PC1 (+) and PB7 (−) • XT: external high-speed oscillator across PC1 (+) and PB7 (−) • EC: external oscillator at PC1 (+), PB7 as I/O • INTOSCIO : PC1 and PB7 as I/O		INTOSCIO
IESO	Two-speed Startup for XT and LP	• Enable • Disable	Enable
FSCMEN	Fail-Safe Clock Monitor	• Enable • Disable	Enable
TSEL	Correspondence between instruction clock and system clock (1T, 2T or 4T)	• 1 (Instruction Clock = SysClk) • 2 (Instruction Clock = SysClk/2) • 4 (Instruction Clock = SysClk/4)	2
OSTPER	OST Timer Period Selection (XT / LP apply)	• 512 • 1024 • 2048 • 4096 (32768 in LP mode)	1024

Table 6-1 BOOT Level FOSC and 2-speed Start-Up configurations

SysClk Source			configuration		
			SCS	LFMOD	OST
			OSCCON[0]	TCKSRC[7]	(Optional, see OSTPER)
			0x99	0x31F	
			RW-0	RW-0	
External	EC		0	-	-
	XT		0	-	1,024 (default)
	LP		0	-	1,024 (default)
Internal	HIRC	16 MHz	1	-	-
	LIRC	256 kHz [2]	1	1	-
		32 kHz [3]	1	0	-

Name	Status			Register	Addr.	Reset
MCKCF	SysClk divider (EC/XT/LP/HIRC)			OSCCON[7:4]	0x99	RW−0100
	0111 = 1	0100 = 8	0001 = 64			
	0110 = 2	0011 = 16	1xxx = 128			
	0101 = 4	0010 = 32	0000 = LIRC			

Table 6-2 Instruction Level SysClk source setup

Name	Status		Register	Addr.	Reset
OSTS	Oscillator Start-Up Time-out (latched) satus 1 = Running from the external clock (start-up successful) 0 = Running from the internal oscillator		OSCCON[3]	0x99	RO−x
HTS	HIRC ready (latched)	1 = Yes 0 = No	OSCCON[2]		RO−0
LTS	LIRC ready (latched)	1 = Yes 0 = No	OSCCON[1]		RO−0
SYSON	In SLEEP mode, the Sysclk controlled 1 = Keep active 0 = Disable		CKOCON[7]	0x95	RW−0
CCORDY	Clock Output Flag	1 = Yes 0 = No	CKOCON[6]		RO−0
DTYSEL	TIM1/TIM2 Multiplier Clock Duty Cycle Adjustment Bit		CKOCON[5:4]		RW−10

[2] 256 kHz LIRC is used only for ADC only (see ADCS and LFMOD,Table 11-3).

[3] Sysclk source (MCKCF = 0000), PWRT, LIRC and HIRC cross-calibration, FSCM, WDT (WCKSRC = 00), Timer1 (T1CKSRC = 101), Timer2 (T2CKSRC = 101), and clock output (CCOSEL = 010) use the 8-frequency division of LIRC ,i.e.32 kHz , regardless of the value of LFMOD.

Name	Status		Register	Addr.	Reset
	00 = 2ns delay 10 = 4ns delay 01 = 3ns delay 11 = 7ns delay				
CCOSEL	Clock Output Selection 000 = Sysclk 100 = T1CK 001 = HIRC 101 = T2CK 010 = LIRC 110 = LP (*) 011 = XT (*) 111 = EC (*) (*) FOSC should be configured to LP/XT/EC mode, otherwise the clock output may be incorrect or no output;		CKOCON[3:1]		RW−000
CCOEN	Clock Output	1 = Enable 0 = Disable	CKOCON[0]		RW−0
AFP1[6]	Clock Output Pin 1 = CLKO map to PD4 0 = CLKO map to PB1		AFP1[6]	0x19F	RW−0
CKMAVG	4x averaging for LIRC and HIRC Cross Calibration 1 = Enable 0 = Disable		MSCKCON[1]	0x41D	RW−0
CKCNTI	Start the LIRC and HIRC Cross calibration function Calibration functionCalibration 1 = Start 0 = Finished (auto-cleared)		MSCKCON[0]		RW−0
SOSCPR	LIRC Period Calibrated by number of HIRC clocks		SOSCPR[11:0]	0x41F[3:0] 0x41E[7:0]	RW−FFF
TUN	Internal HIRC frequency tunable register		OSCTUNE[6:0]	0x98	RW−xxxx xxxx
TKEN	Touch Clock module	1 = Enable 0 = Disable	PCKEN[7]	0x9A	RW−0
I2CEN	I2C Clock module		PCKEN[6]		RW−0
UARTEN	USART Clock module		PCKEN[5]		RW−0
SPICKEN	SPI Clock module		PCKEN[4]		RW−0
TIM4EN	Timer4 Clock module		PCKEN[3]		RW−0
TIM2EN	Timer2 Clock module		PCKEN[2]		RW−0
TIM1EN	Timer1 Clock module		PCKEN[1]		RW−0
ADCEN	ADC Clock module		PCKEN[0]		RW−0

Table 6-3 Oscillators Control/Status

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt 1 = Enable (PEIE, OSFIE, CKMIE apply) 0 = Global Shutdown (Wake-Up not affected)		INTCON[7]	Bank first address +0x0B	RW−0
PEIE	Peripheral Interrupt Enable 1 = Enable (OSFIE, CKMIE apply) 0 = Disable (no Wake-Up)		INTCON[6]		RW−0
OSFIE	External Oscillator Failed Interrupt	1 = Enable 0 = Disable (no Wake-Up)	INTCON[3]		RW−0
OSFIF [4]	External Oscillator Failed Interrupt Flag	1 = Yes (latched) 0 = No	INTCON[0]		R_W1C−0
CKMIE	LIRC and HIRC cross Calibration Completion Interrupt	1 = Enable 0 = Disable (no Wake-Up)	PIE1[1]	0x91	RW−0
CKMIF [4]	LIRC and HIRC cross Calibration Completion Flag	1 = Yes (latched) 0 = No	PIR1[1]	0x11	R_W1C−0

Table 6-4 Oscillators Interrupt Enable and Status Bits

6.2. Internal Clock Modes (HIRC and LIRC)

Internal high frequency clock (HIRC) is factory calibrated to 16 MHz @ 2.5V/25°C. Typical value of frequency change between chips < ±1.5% at 2.5 − 5.5V, 25°C. The typical temperature variation from $^{-}40$ − $^{+}85$ °C is ±2.0%.

HIRC accuracy is calibrated at the wafer level. Packaging may cause the HIRC frequency to drift. There is an option at the downloader to re-calibrate the HIRC. The HIRC frequency trimmed value is stored in the "OSCTUNE" register. Users can change HIRC from the default 16 MHz (tuning). Trimming steps are non-linear (~80 kHz). A rough estimation is as follows:

$$\text{OSCTUNE}[7:0] \pm N \approx 16000 \pm N * 80 \text{ (kHz)}$$

Internal low frequency clock (LIRC) is factory calibrated to 32 kHz. Typical value of frequency change between chips is <±9.5% at 2.5 − 5.5V, 25°C. The temperature variation from $^{-}40$ − $^{+}85$ °C is < ±2.0%.

LIRC and HIRC can be used to cross calibrate each other – A build in hardware uses Timer2 to measure the number of Instruction Clocks (set SysClk to HIRC at 16 MHz) in one LIRC period (32 kHz).

[4] Write '1' to clear, and writing '0' has no effect on the bit value. It is recommended that only STR and MOVWI instructions be used for write operations, not BSR or IOR instructions.

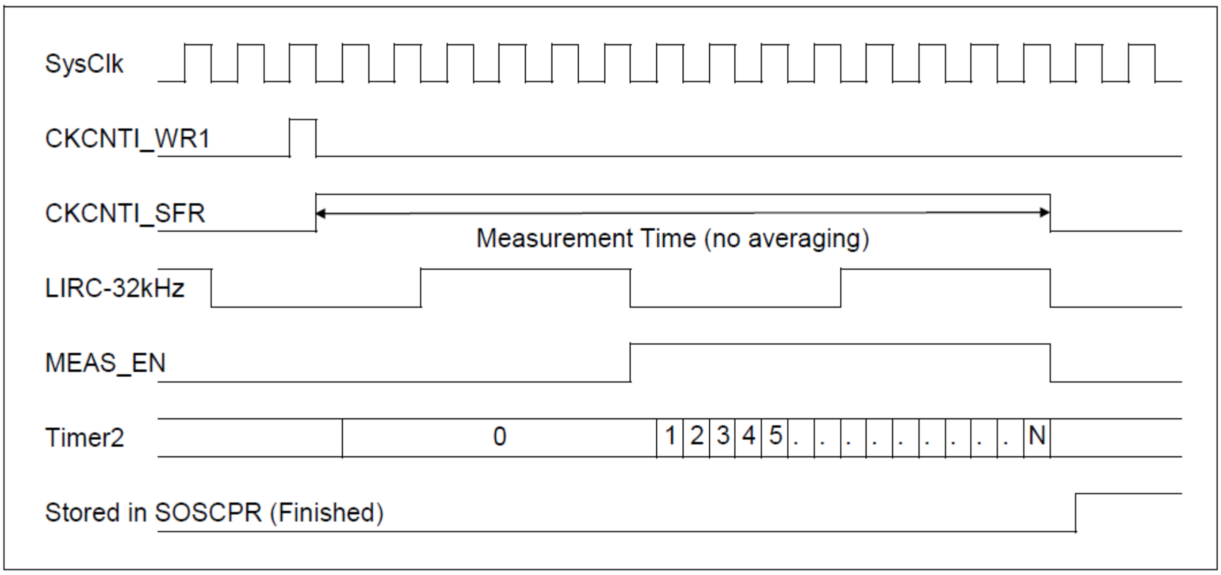

SysClk
CKCNTI_WR1
CKCNTI_SFR
Measurement Time (no averaging)
LIRC-32kHz
MEAS_EN
Timer2
0
1 2 3 4 5 . . . . N
Stored in SOSCPR (Finished)

Figure 6-2 Single measurement Timing Diagram

To enable LIRC and HIRC Cross Calibration:

1. Set MCKCF = 111, SCS = 1 ; select SysClk at 16MHz HIRC (other settings will have a lower accuracy).
2. Set CKMAVG = 1 ; 4 times averaging, choose 0 for no averaging.
3. Set TIM2EN = 1, T2CEN = 1 ; enable Timer2.
4. Set CKCNTI = 1 ; start calibration, automatically Timer2 prescaler = 1, postscalar = 1, T2CKSRC = HIRC
5. At the end of the calibration “CKCNTI =0”, “CKMIF = 1” automatically.
6. Measured value is stored at SOSCPR;
7. LIRC is 32kHz and CPU is running at 16MHz / 2T, the ideal matching number is 500.

Notes:

- Do not write SOSCPRH/L during LIRC and HIRC Cross Calibration.
- Timer2 cannot be used by other peripherals during LIRC and HIRC Cross Calibration.
- LIRC and HIRC Cross Calibration is incompatible with Single Step Debugger mode.
- When CKCNTI = 1, LIRC is automatically turned on and keeps running in SLEEP mode, and only when SYSON = 1 for calibration to run in SLEEP mode.

  After power on, LIRC and HIRC Cross Calibration will turn on automatically. At this time, CKCNTI=1, CKMAVG=0, no need to set T2CEN. After the automatic cross calibration is completed, the CKMIF flag will not be set, and the CKCNTI will be reset automatically before TIM2 can be configured or used by other peripherals.

6.3. External Clock Modes

6.3.1. EC mode

External digital signal connected to OSC1 is the clock source (OSC2 is available for I/O). There is no set up or transition time delay when EC is used for SysClk after a POR or a wake-up from sleep.

6.3.2. LP and XT modes

A quartz crystal resonator or ceramic resonator is connected between OSC1 and OSC2 in LP or XT modes.

LP Oscillator mode has the lowest gain setting and current consumption of the three modes (EC, LP and XT). This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals).

XT Oscillator mode selects the highest gain setting of the internal inverter-amplifier.

After a BOOT or a Wake-Up from Sleep, CPU program execution is suspended during OST counting if the clock source is XT or LP mode. This allows the XT or LP clock to stabilize. OST counts OSC1 (+ve terminal of the crystal input) in XT and LP mode. The number of counts is determined by the initialization configuration register OSTPER. For a 32.768kHz tuning-fork type crystals the OST time is at least 1 second when OSTPER=32.768.

Notes:

- WDT is held in cleared until OST finished counting.
- Do not write WDTCON / OPTION during OST counting, otherwise unexpected behavior will occur.

Two-Speed Clock Start-up (see "IESO" in **Table 6-1**) allows instructions execution while OST counts, using the internal oscillator INTOSC as SysClk. It removes the external oscillator start-up time from the time spent awake and can reduce the overall power consumption, especially in cases of frequent SLEEP mode usage. The CPU wakes up from Sleep, performs a few instructions using the INTOSC as SysClk and return to Sleep without having to wait for the primary oscillator to become stable.

Note: Two-Speed Start-up is disabled for EC mode, as the oscillator does not require stabilization time.

Two Speed Start-up sequence

1. After a BOOT or Wake-up from Sleep.
2. INTOSC is used as SysClk for Instructions execution until OST time out.
3. SysClk is held low from the falling edge of INTOSC until the falling edge of the new clock (LP or XT mode).
4. SysClk switches to the external clock source.

The Oscillator Start-up Time-out Status (OSTS) indicates whether the SysClk is running from the external clock source or from the internal clock source. This is an indirect way to find out if the Oscillator Start-up Timer (OST) has timed out for the LP or XT mode when the Two-Speed Clock Start-up mode is on.

Executing a SLEEP instruction will abort the OST, and OSTS will remain "0".

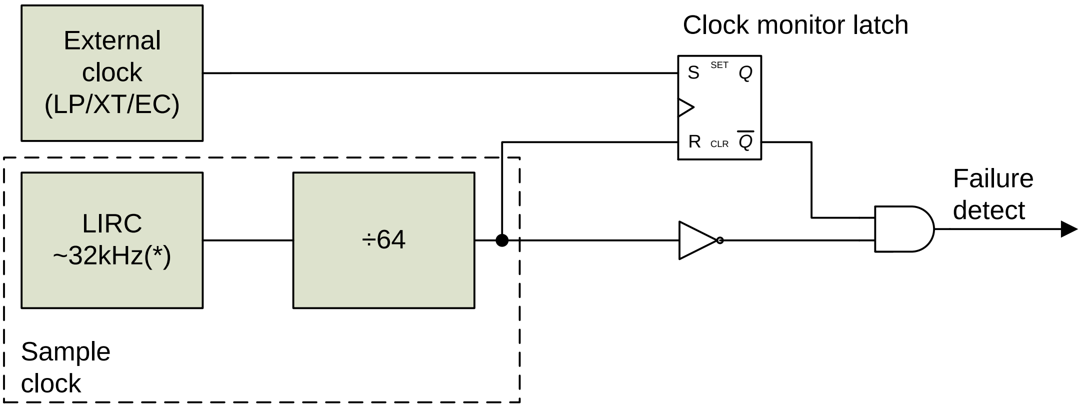
Fail-Safe Clock Monitor (FSCM, enabled by "FSCMEN", see **Table 6-1**) allows the device to continue operating when the external oscillator fails. The FSCM can detect oscillator failure any time after the

Oscillator Start-up Timer (OST) has expired. The FSCM is applicable to all external oscillator modes (EC, LP and XT). It is recommended that FSCM be enabled if an external oscillator is used.

An external oscillator is considered fail if it oscillates at ~1 kHz or below. A sample clock is generated by dividing the LIRC by 64. The external clock sets a latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is when an entire half-cycle of the sample clock elapses without the primary clock goes low.

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets OSFIF. Setting OSFIF to 1 will generate an interrupt if OSFIE is enabled. The device hardware can then take steps to mitigate the problems that may arise from a failed clock. The SysClk will continue to be sourced from the internal clock source until the device hardware successfully restarts the external oscillator.

The internal clock source chosen by “FSCM” is determined by “MCKCF”. This allows the internal oscillator to be configured before a failure occurs.



Note: LFMOD does not affect the sample clock.

Figure 6-3 FSCM Block Diagram

The Fail-Safe condition is cleared after a Reset, executing a SLEEP instruction or toggling the SCS. When SCS is toggled, OST is restarted. While OST is running, the device continues to operate from CPU chose INTOSC as Sysclk. When OST times out, the Fail-Safe condition is cleared and the device will be operating from the external clock source. The Fail-Safe condition must be resolved before the OSFIF flag is cleared.

Note: Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, will not update SCS. The program can monitor OSTS to determine the current SysClk source.

6.4. HIRC, LIRC and EC inter-switching

Figure 6-4 shows the timing during inter-switching. If either HIRC or LIRC is closed prior to switching (to save power) there is an extra oscillator setup delay time, HTS and LTS indicate the status of the corresponding oscillator respectively.

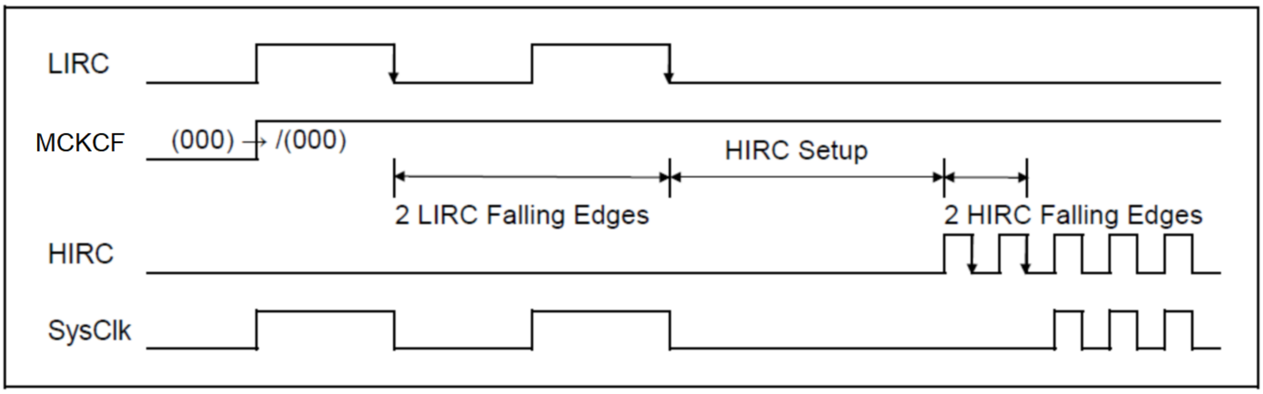


Figure 6-4 Switching from LIRC to HIRC (same principle applies switching among EC, LIRC, HIRC)

7. TIMERS

There are 4 Timers including the Watch Dog Timer (WDT).

	WDT	Timer1	Timer2	Timer4
Prescaler (bit)	3	16	4 (1x, 4x, 16x)	3
Counter (bit)	16	16	16	8
Postscaler (bit)	–	–	–	–
Clock Sources	• LP • XT • HIRC • LIRC	• EC, LP or XT • HIRC • Sysclk • LIRC • 2x HIRC • 2x (EC, LP or XT)	• EC, LP or XT • HIRC • Sysclk • LIRC • 2x HIRC • 2x (EC, LP or XT)	• LP • XT • HIRC • Sysclk

Table 7-1 Timers' Resources

Notes: If a Timer’s clock source is not the system Clock, set “TxCEN = 0” before changing TMRx.

Any Timer enabled will turn on its clock source automatically. System Clock is disabled at SLEEP so it cannot be used for WDT. When LP / XT / EC Oscillator is selected as Timers’ clock source, FOSC must be configured correspondingly ,otherwise the oscillator is off and no counting will occur.

In a POR or System-Reset, all Timers’ counter and prescaler are reset. The followings will also reset a Timer’s counter and prescaler(s):

	WDT	Timer1	Timer2	Timer4
Prescaler	• WDT disabled	• Reset mode	• T2CEN = 0	• T4CEN = 0
Counter	• WDT, OST overflow • Enter/Exit SLEEP • CLRWDT • WDTCON write • WCKSEL write	• T1CNT = T1ARR	• T1CNT=T2ARR	• T4CNT = T4ARR

Table 7-2 Events reseting a Timers' Counter and Scaler(s)

7.1. Watch Dog Timer (WDT)

WDT is used to “Wake-Up from SLEEP” or “System-Reset if the CPU suspend”. WDT counts the number of clock cyles to a pre-set number until overflow.

- In SLEEP mode, a WDT overflow will trigger a Wake-up. The CPU will resume operation from where it is before SLEEP. This is not an Interrupt nor System-Reset event.
- In non-SLEEP mode, a WDT overflow will trigger a System-Reset and BOOT (see **Section 4** System-Reset).

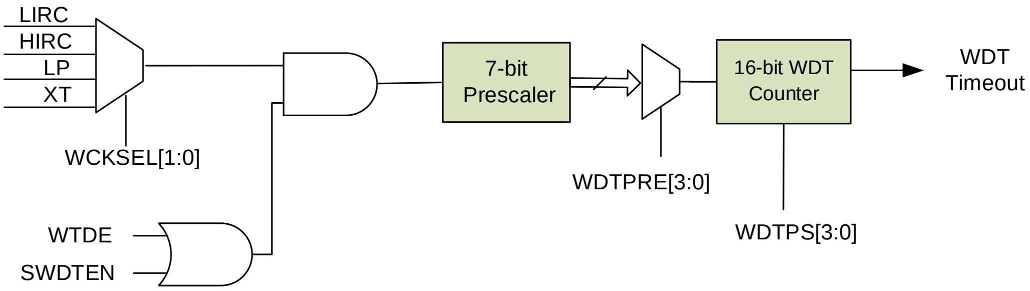


Figure 7-1 Block Diagram of WDT

The WDT will overflow after a WatchDog-Time: WDT-Period x WDT-Prescaler / WDT Clock Frequency.

For a given Clock Source, WatchDog-Time step is a continuous multiple due to the binary nature of the WDT Prescalar. Using LIRC as clock source, the maximum settable time before WDT overflows is

$$2^{16} \text{ x } 2^{7} \text{ / 32kHz = ~262 seconds}$$

7.1.1. Summary of WDT Related Registers

Name	Functions	Default
WDTE	WDT • Enable (Instructions can not be disabled) • Instruction controlled (SWDTEN)	SWDTEN control

Table 7-3 BOOT Level WDT Selectors

Name	Status	Register	Addr.	Reset
WCKSEL	WDT Clock Sources 00 = LIRC 01 = HIRC 10 = LP (only FOSC in LP or INTOSCIO mode*) 11 = XT (only FOSC in XT or INTOSCIO mode*) *Otherwise misconfigured, no WDT Clock Source	MISC0[1:0]	0x11C	RW−00
WDTPRE	WDT Prescaler 000 = 1 001 = 2 010 = 4 011 = 8 100 = 16 101 = 32 110 = 64 111 = 128 (default)	WDTCON[7:5]	0x97	RW−111
WDTPS	WDT Period 0000 = 32 0001 = 64 0010 = 128 0011 = 256 0100 = 512 (default) 0101 = 1,024 0110 = 2,048 0111 = 4,096 1000 = 8,192 1001 = 16,384 1010 = 32,768 1011 = 65,536 11xx = 65,536	WDTCON[4:1]		RW−0100
SWDTEN	1 = WDT Enables 0 = WDT Disables (if WDTE choosed SWDTEN control)	WDTCON[0]		RW−0

Table 7-4 Instruction Level WDT Related Registers

7.1.2. Configuration and using the WDT

WDTE (BOOT Level) and SWDTEN (Instruction Level) enable the WDT. After a WDT triggered Reset, it will also BOOT.

The WDT prescaler is set by WDTPRE, and the clock source is selected by WCKSEL (default at 32kHz if the LIRC is selected, regardless of the value of LFMOD). When the WDT enabled, the selected clock source is automatically turned on and will remain active in SLEEP mode.

To stop a WDT overflow, the WDT must be cleared before time expires. Refer to **Table 7-2** for events that will clear the WDT. Counting continues after WDT is cleared.

7.2. Advanced TIMER1

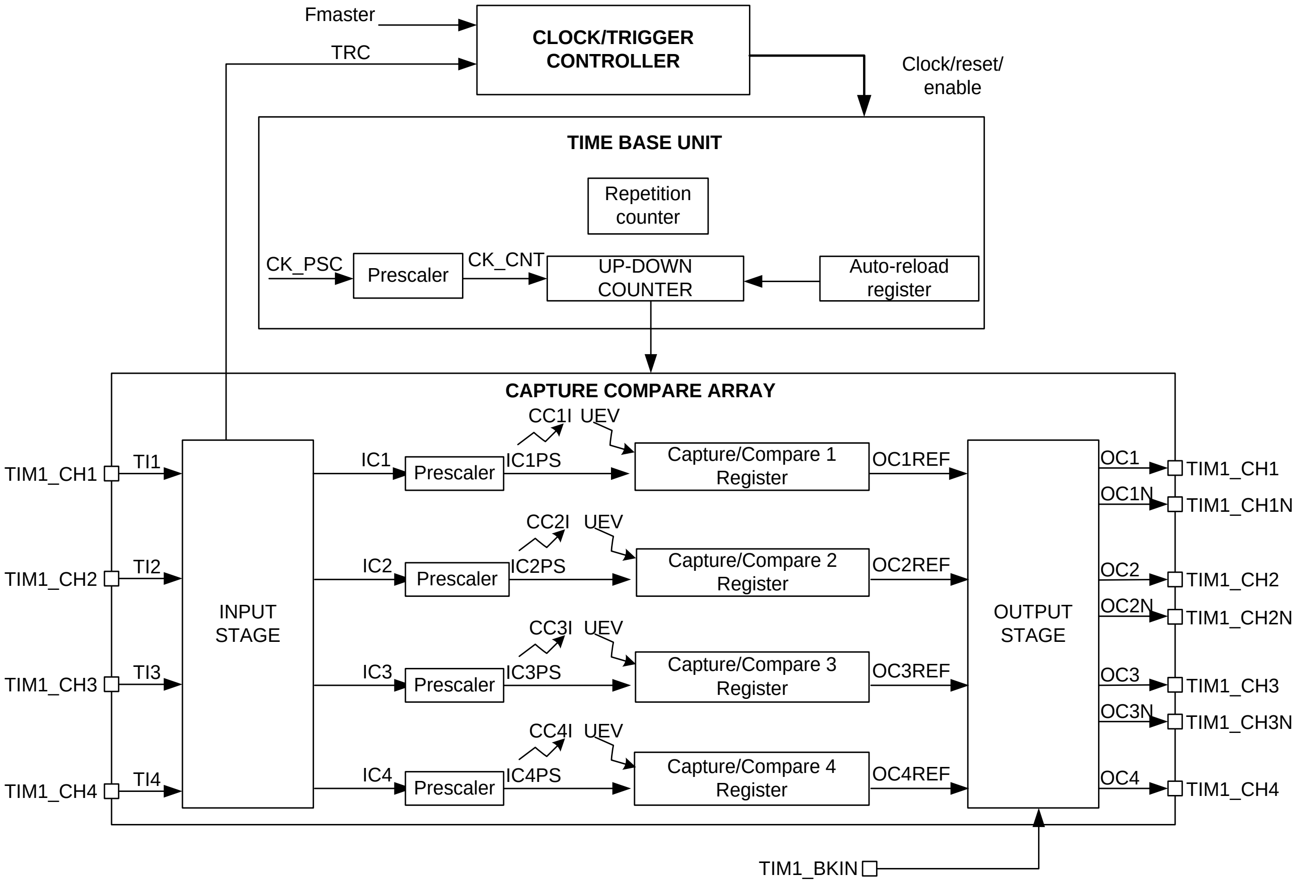


Figure 7-2 Block Diagram of TIM1

TIM1 Features:

- 16 bit up, down, up/down counting, supporting automatic reload
- Repeat Count
- 16 bit programmable prescaler
- Counting control mode: Internal clock mode, reset mode, gating mode, trigger mode.
- 4-way polarity optional channel support:
  - ✓ Input capture
  - ✓ Output compare
  - ✓ PWM channels with the same period and independent duty cycle (edge or center alignment), 3 channels supporting complementary output and programmable deadband
  - ✓ One-pulse output
  - ✓ Fault-break function (optional auto-restart)
- Interruption event: update event, input trigger, input capture, output compare, fault-break input
- Support 3 level of register write protection lock setting (T1LOCK)

7.2.1. Summary of Timer1 Related Registers

Name	Addr.	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset
PCKEN	0x9A	TKEN	I2CEN	UARTEN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
TIM1CR1	0x211	T1ARPE	T1CMS[1:0]		T1DIR	T1OPM	T1URS	T1UDIS	T1CEN	0000 0000
TIM1SMCR	0x213	—	T1TS[2:0]			—	T1SMS[2:0]			-000 -000
TIM1IER	0x215	T1BIE	T1TIE	—	T1CC4IE	T1CC3IE	T1CC2IE	T1CC1IE	T1UIE	00-0 0000
TIM1SR1	0x216	T1BIF	T1TIF	—	T1CC4IF	T1CC3IF	T1CC2IF	T1CC1IF	T1UIF	00-0 0000
TIM1SR2	0x217	—	—	—	T1CC4OF	T1CC3OF	T1CC2OF	T1CC1OF	—	---0 000-
TIM1EGR	0x218	T1BG	—	—	T1CC4G	T1CC3G	T1CC2G	T1CC1G	—	0--0 000-
TIM1CCMR1 (output mode)	0x219	—	T1OC1M[2:0]			T1OC1PE	—	T1CC1S[1:0]		-000 0-00
TIM1CCMR1 (input mode)		T1IC1F[3:0]				T1IC1PSC[1:0]		T1CC1S[1:0]		0000 0000
TIM1 CCMR2 (output mode)	0x21A	—	T1OC2M[2:0]			T1OC2PE	—	T1CC2S[1:0]		-000 0-00
TIM1CCMR2 (input mode)		T1IC2F[3:0]				T1IC2PSC[1:0]		T1CC2S[1:0]		0000 0000
TIM1CCMR3 (output mode)	0x21B	—	T1OC3M[2:0]			T1OC3PE	—	T1CC3S[1:0]		-000 0-00
TIM1CCMR3 (input mode)		T1IC3F[3:0]				T1IC3PSC[1:0]		T1CC3S[1:0]		0000 0000
TIM1CCMR4 (output mode)	0x21C	—	T1OC4M[2:0]			T1OC4PE	—	T1CC4S[1:0]		-000 0-00
TIM1CCMR4 (input mode)		T1IC4F[3:0]				T1IC4PSC[1:0]		T1CC4S[1:0]		0000 0000
TIM1CCER1	0x21D	T1CC2NP	T1CC2NE	T1CC2P	T1CC2E	T1CC1NP	T1CC1NE	T1CC1P	T1CC1E	0000 0000
TIM1CCER2	0x21E	—	—	T1CC4P	T1CC4E	T1CC3NP	T1CC3NE	T1CC3P	T1CC3E	--00 0000
TIM1CNTRH	0x28C	T1CNT[15:8]								0000 0000
TIM1CNTRL	0x28D	T1CNT[7:0]								0000 0000
TIM1PSCRH	0x28E	T1PSC[15:8]								0000 0000
TIM1PSCRL	0x28F	T1PSC[7:0]								0000 0000
TIM1ARRH	0x290	T1ARR[15:8]								1111 1111
TIM1ARRL	0x291	T1ARR[7:0]								1111 1111
TIM1RCR	0x292	T1REP[7:0]								0000 0000
TIM1CCR1H	0x293	T1CCR1[15:8]								0000 0000
TIM1CCR1L	0x294	T1CCR1[7:0]								0000 0000
TIM1CCR2H	0x295	T1CCR2[15:8]								0000 0000
TIM1CCR2L	0x296	T1CCR2[7:0]								0000 0000
TIM1CCR3H	0x297	T1CCR3[15:8]								0000 0000
TIM1CCR3L	0x298	T1CCR3[7:0]								0000 0000
TIM1CCR4H	0x299	T1CCR4[15:8]								0000 0000
TIM1CCR4L	0x29A	T1CCR4[7:0]								0000 0000
TIM1BKR	0x29B	T1MOE	T1AOE	T1BKP	T1BKE	T1OSSR	T1OSSI	T1LOCK[1:0]		0000 0000
TIM1DTR	0x29C	T1DTG[7:0]								0000 0000
TIM1OISR	0x29D	—	T1OIS4	T1OIS3N	T1OIS3	T1OIS2N	T1OIS2	T1OIS1N	T1OIS1	-000 0000
LEBCON	0x41C	LEBEN	LEBCH[1:0]		—	EDGS	BKS[2:0]			000- 0000

Table 7-5 Summary of Timer1 Related Registers(-reserved bit must keep as reset, and cannot be changed)

Name	Status		Register	Addr.	Reset
T1CNT	TIM1 Count value	MSB	TIM1CNTRH[7:0]	0x28C	RW−0000 0000
		LSB	TIM1CNTRL[7:0]	0x28D	RW−0000 0000
T1PSC	TIM1Prescaler	MSB	TIM1PSCRH[7:0]	0x28E	RW−0000 0000
		LSB	TIM1PSCRL[7:0]	0x28F	RW−0000 0000
T1ARR	Auto-reload register for counting period (preload value) Note: When this value is 0, the counter does not work;	MSB	TIM1ARRH[7:0]	0x290	RW−1111 1111
		LSB	TIM1ARRL[7:0]	0x291	RW−1111 1111
T1REP	Repeat Count Down		TIM1RCR[7:0]	0x292	RW−0000 0000
T1CCR1	Input capture mode: Last capture event (IC1) Captured count value	MSB	TIM1CCR1L[7:0]	0x293	RO−0000 0000
		LSB	TIM1CCR1L[7:0]	0x294	RO−0000 0000
	Output compare mode: Output compare value of TIM1_CH1 (Preload value)	MSB	TIM1CCR1H[7:0]	0x293	RW−0000 0000
		LSB	TIM1CCR1H[7:0]	0x294	RW−0000 0000
T1CCR2	Input capture mode: Last capture event (IC2) Captured count value	MSB	TIM1CCR2H[7:0]	0x295	RO−0000 0000
		LSB	TIM1CCR2L[7:0]	0x296	RO−0000 0000
	Output compare mode: Output compare value of TIM1_CH2 (Preload value)	MSB	TIM1CCR2H[7:0]	0x295	RW−0000 0000
		LSB	TIM1CCR2L[7:0]	0x296	RW−0000 0000
T1CCR3	Input capture mode: Last capture event (IC3) Captured count value	MSB	TIM1CCR3H[7:0]	0x297	RO−0000 0000
		LSB	TIM1CCR3L[7:0]	0x298	RO−0000 0000
	Output compare mode: Output compare value of TIM1_CH3 (Preload value)	MSB	TIM1CCR3H[7:0]	0x297	RW−0000 0000
		LSB	TIM1CCR3L[7:0]	0x298	RW−0000 0000
T1CCR4	Input capture mode: Last capture event (IC4) Captured count value	MSB	TIM1CCR4H[7:0]	0x299	RO−0000 0000
		LSB	TIM1CCR4L[7:0]	0x29A	RO−0000 0000
	Output compare mode: Output compare value of TIM1_CH4 (Preload value)	MSB	TIM1CCR4H[7:0]	0x299	RW−0000 0000
		LSB	TIM1CCR4L[7:0]	0x29A	RW−0000 0000

Table 7-6 Timer1 Period Related Register

Name	Status	Register	Addr.	Reset
TIM1EN	TIM1 Clock module 1 = Enable 0 = Disable	PCKEN[1]	0x9A	RW-0
SYSON	In SLEEP mode, the Sysclk controlled 1 = Enable 0 = Disable	CKOCON[7]	0x95	RW−0
T1CKSRC	TIM1Clock Sources (Fmaster) 000 = Sysclk 100 = 2x (XT or EC) $^{(*)}$ 001 = HIRC 101 = LIRC 010 = XT or EC $^{(*)}$ 110 = LP or EC $^{(*)}$ 011 = 2x HIRC 111 = 2x (LP or EC) $^{(*)}$ $^{(*)}$FOSC should be configured accordingly or in LP/XT/EC mode, otherwise oscillator will not run.	TCKSRC[2:0]	0x31F	RW−000
DTYSEL	TIM1/TIM2 Multiplier Clock Duty Cycle Adjustment Bit 00 = 2ns delay 10 = 4ns delay 01 = 3ns delay 11 = 7ns delay	CKOCON[5:4]	0x95	RW−10
T1ARPE	Automatic pre-loading of counting cycles 1 = Enable (T1ARR preload value is loaded when the update event arrives) 0 = Disable (T1ARR loaded immediately)	TIM1CR1[7]	0x211	RW−0
T1CMS	Counter alignment modes 00 = Edge-aligned mode (Counting direction is determined by T1DIR) 01 = Center-aligned mode1 (T1CCxIF set to 1 when counting down) 10 = Center-aligned mode2 (T1CCxIF set to 1 when counting up) 11 = Center-aligned mode3 (T1CCxIF is set to 1 for both upward and downward counting) Notes. 1. Center-aligned mode means that the counter counts up and down alternately. 2. Mode switching is allowed only when the counter is disabled (T1CEN=0).	TIM1CR1[6:5]		RW−00
T1DIR	Count direction (when T1CMS ≠ 00, this bit is read-only)	TIM1CR1[4]		RW−0

Name	Status	Register	Addr.	Reset
	1 = Up 0 = Down			
T1OPM	One-pulse mode 1 = Enable (When the next update event comes, T1CEN will be reset automatically and the counter will stop) 0 = Disable (The counter does not stop when an update event occurs)	TIM1CR1[3]		RW−0
T1URS	Update event source when T1UDIS=0 1 =Counter overflow/underflow 0 =Counter overflow/underflow, or Reset Trigger Event	TIM1CR1[2]		RW−0
T1UDIS	Generate update event control 1 = Disable 0 = Enable	TIM1CR1[1]		RW−0
T1CEN	TIM1 Counter 1 = Enable 0 = Disable	TIM1CR1[0]		RW−0
T1TS	Trigger input source of synchronization counter (TRGI) 0xx = Reserved 100 = Edge detector of channel 1 input TI1 (TI1F_ED) 101 =Channel 1 input after filtering(TI1FP1) 110 =Channel 2 input after filtering(TI2FP2) 111 = Disables configuration Note: 1. The trigger input source can only be changed when T1SMS=000; 2. See T1CC1P/T1CC2P or T1ETP for polarity of effective edge/active level of trigger input;	TIM1SMCR[6:4]	0x213	RW−000

Name	Status	Register	Addr.	Reset
T1SMS	Trigger mode 000 = Internal clock 100 = Reset mode (When the active edge of the input is triggered, the counter is cleared and counted again from 0) 101 = Gating mode (Counter counts during triggering input effective level, and stops counting when inactive level is reached, but does not reset) 110 = Trigger mode (Counter counts and does not reset when triggering the effective edge of input) other = Reserved Note: The trigger input of Gating mode cannot select TI1F_ ED;	TIM1SMCR[2:0]		RW−000

Table 7-7 Instruction Level Timer1 Related Control Registers

Name	Addr.	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	Reset
TIM1CCMR1	0x219	T1IC1F[3:0]				T1IC1PSC[1:0]		T1CC1S[1:0]		RW-0000 0000
TIM1CCMR2	0x21A	T1IC2F[3:0]				T1IC2PSC[1:0]		T1CC2S[1:0]		RW-0000 0000
TIM1CCMR3	0x21B	T1IC3F[3:0]				T1IC3PSC[1:0]		T1CC3S[1:0]		RW-0000 0000
TIM1CCMR4	0x21C	T1IC4F[3:0]				T1IC4PSC[1:0]		T1CC4S[1:0]		RW-0000 0000

<table>
<tr><th>Name</th><th colspan="3">Status</th><th>Register</th><th>Addr.</th><th>Reset</th></tr>
<tr><td rowspan="9">T1ICxF</td><td colspan="3">Sampling frequency and digital filter length of channel x input capture</td><td rowspan="9">TIM1CCMRx[7:4]
x = 1, 2, 3, 4</td><td rowspan="9">0x219/
0x21A/
0x21B/
0x21C</td><td rowspan="9">RW−0000</td></tr>
<tr><td>Value</td><td>Sampling frequency($f_{SAMPLING}$)</td><td>Digital filter length N</td></tr>
<tr><td>0000</td><td>Fmaster</td><td>0</td></tr>
<tr><td>0001</td><td>Fmaster</td><td>2</td></tr>
<tr><td>0010</td><td>Fmaster</td><td>4</td></tr>
<tr><td>0011</td><td>Fmaster</td><td>8</td></tr>
<tr><td>0100</td><td>Fmaster / 2</td><td>6</td></tr>
<tr><td>0101</td><td>Fmaster / 2</td><td>8</td></tr>
<tr><td>0110</td><td>Fmaster / 4</td><td>6</td></tr>
</table>

	0111	Fmaster / 4	8			
	1000	Fmaster / 8	6			
	1001	Fmaster / 8	8			
	1010	Fmaster / 16	5			
	1011	Fmaster / 16	6			
	1100	Fmaster / 16	8			
	1101	Fmaster / 32	5			
	1110	Fmaster / 32	6			
	1111	Fmaster / 32	8			
T1ICxPSC	Channel x input capture prescaler (several events trigger a capture) 00 = 1 prescaler 10 = 4 prescaler 01 = 2 prescaler 11 = 8 prescaler Note: When T1CCxE=0, the prescaler is reset to 00			TIM1CCMRx[3:2]		RW−00
T1CC1S [1]	Channel 1 Mode selection	00 = Output 01 = Input, input pin is mapped to TI1FP1 10 = Input, input pin is mapped to TI2FP1 11 = Input, input pin is mapped to TRC		TIM1CCMR1[1:0]	0x219	RW−00
T1CC2S [2]	Channel 2 Mode selection	00 = Output 01 = Input, input pin is mapped to TI2FP2 10 = Input, input pin is mapped to TI1FP2 11 = Input, input pin is mapped to TRC		TIM1CCMR2[1:0]	0x21A	RW−00
T1CC3S [2]	Channel 3 Mode selection	00 = Output 01 = Input, input pin is mapped to TI3FP3 10 = Input, input pin is mapped to TI4FP3 11 = Reserved		TIM1CCMR3[1:0]	0x21B	RW−00

[1] Can be written only when channel x disables (T1CCxE=0 and T1CCxNE=0).

[2] Can be written only when channel x disables (T1CCxE=0 and T1CCxNE=0).

T1CC4S [2]	Channel 4 Mode selection	00 = Output 01 = Input, input pin is mapped to TI3FP4 10 = Input, input pin is mapped to TI4FP4 11 = Reserved	TIM1CCMR4[1:0]	0x21C	RW−00

Table 7-8 TIM1CCMRx as Input Configuration Register

Name	Addr.	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	Reset
TIM1CCMR1	0x219	−	T1OC1M[2:0]			T1OC1PE	−	T1CC1S[1:0]		RW−-0000-00
TIM1CCMR2	0x21A	−	T1OC2M[2:0]			T1OC2PE	−	T1CC2S[1:0]		RW−-0000-00
TIM1CCMR3	0x21B	−	T1OC3M[2:0]			T1OC3PE	−	T1CC3S[1:0]		RW−-0000-00
TIM1CCMR4	0x21C	−	T1OC4M[2:0]			T1OC4PE	−	T1CC4S[1:0]		RW−-0000-00

T1OCxM	Channel x Output compare mode		Level of reference signal OCxREF
000	Frozen (no compare)		remain unchanged
001	When T1CNT = CCRx_SHAD		1
010	When T1CNT = CCRx_SHAD		0
011	When T1CNT = CCRx_SHAD		Level reversal
100	Forced inactive		0
101	Forced active		1
110	PWM1 mode	T1CNT < CCRx_SHAD	1
		T1CNT > CCRx_SHAD	0
111	PWM2 mode	T1CNT < CCRx_SHAD	0
		T1CNT > CCRx_SHAD	1
Note: The output reference signal OCxREF is active at high level, which together with the polarity selection T1CCxP determines the actual output value of pin OCx;			

Table 7-9 T1OCxM Configured as Output Compare Mode

Name	Status		Register	Addr.	Reset
T1OCxPE	Automatic preloading of channel x Output compare values 1 = Enable (T1CCRx preload values are loaded when the update event arrives) 0 = Disable (T1CCRx loaded immediately) Note: It must be enabled under PWM mode, and single pulse mode is optional		TIM1CCMRx[3] x = 1, 2, 3, 4	0x219/ 0x21A/ 0x21B/ 0x21C	RW−0
T1CC1S [3]	Channel 1 Mode selection	00 = Output 01 = Input, input pin is mapped to TI1FP1 10 = Input, input pin is mapped to TI2FP1 11 = Input, input pin is mapped to TRC	TIM1CCMR1[1:0]	0x219	RW−00
T1CC2S [3]	Channel 2 Mode selection	00 = Output 01 = Input, input pin is mapped to TI2FP2 10 = Input, input pin is mapped to TI1FP2 11 = Input, input pin is mapped to TRC	TIM1CCMR2[1:0]	0x21A	RW−00
T1CC3S [3]	Channel 3 Mode selection	00 = Output 01 = Input, input pin is mapped to TI3FP3 10 = Input, input pin is mapped to TI4FP3 11 = Reserved	TIM1CCMR3[1:0]	0x21B	RW−00
T1CC4S [3]	Channel 4 Mode selection	00 = Output 01 = Input, input pin is mapped to TI3FP4 10 = Input, input pin is mapped to TI4FP4 11 = Reserved	TIM1CCMR4[1:0]	0x21C	RW−00

Table 7-10 TIM1CCMRx as Output Configuration Register

[3] Can be written only when channel x is closed (T1CCxE=0 and T1CCxNE=0).

Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	地址	Reset
TIM1CCER1	T1CC2NP	T1CC2NE	T1CC2P	T1CC2E	T1CC1NP	T1CC1NE	T1CC1P	T1CC1E	0x21D	RW-0000 0000
TIM1CCER2	–	–	T1CC4P	T1CC4E	T1CC3NP	T1CC3NE	T1CC3P	T1CC3E	0x21E	RW---00 0000

Name	Functions	Input Capture/Trigger mode (T1CCxS = 01/10)	Output compare mode (T1CCxS = 00)
T1CCxP	Channel x Input/output polarity selection	1 = Capture/trigger occurs at the falling edge or low level of TIxF 0 = Capture/trigger occurs at the rising edge or high level of TIxF Note: Only channel 1 and 2 can be selected as input trigger source	1 = OCx active at low level 0 = OCx active at high level
T1CCxE	Channel x I/O pin function	1 = Enable input capture/trigger function of the pin 0 = Disable	1 = Enable the OCx output function of the pin 0 = Disable
T1CCxNP	Channel x complementary output polarity selection	–	1 = OCxN active at low level 0 = OCxN active at high level
T1CCxNE	Channel x complementary pinout function	–	1 = Enable the OCxN output function of the pin 0 = Disable
Note: The channel output level is jointly determined by the values of T1MOE, T1OSSI, T1OSSR, T1OISx, T1OISxN, T1CCxE and T1CCxNE. See **Table 7-14**。			

Table 7-11 Timer1 Channel Output and Polarity Selection

Name	Status		Register	Addr.	Reset
TIM1_CH1	Channel 1 Pin Remapping	1 = PD1 0 = PA0	AFP0[6]	0x19E	RW−0
TIM1_CH1N	Channel 1 Complementary Pin Remapping	1 = PC7 0 = PC0	AFP0[4]		RW−0
TIM1_CH2	Channel 2 Pin Remapping	1 = PD2 0 = PA1	AFP1[0]	0x19F	RW−0
TIM1_CH2N	Channel 2 Complementary Pin Remapping	1 = PC6 0 = PA3	AFP0[3]	0x19E	RW−0
TIM1_CH3	Channel 3 Pin Remapping	1 = PD3 0 = PB4	AFP1[1]	0x19F	RW−0
TIM1_CH3N	Channel 3 Complementary Pin Remapping	1 = PC5 0 = PB0	AFP0[2]	0x19E	RW−0
TIM1_CH4	Channel 4 Pin Remapping	1 = PD5 0 = PB1	AFP1[5]	0x19F	RW−0
TIM1_BKIN	Fault-Break Source Input Remapping	1 = PD4 0 = PB3	AFP1[3]		RW−0

Table 7-12 Timer1 Pin Function Remapping Register

Name	Control	Register	Addr.	Reset
BKS	Fault-Break Source for TIM1 000 = Disable 001 = BKIN pin 010 = LVD detection 100 = ADC threshold comparison	LEBCON[2:0]	0x41C	RW−000
T1MOE [4]	Main output control (Valid only for channels configured as outputs) 1 = Enable (If T1CCxE/T1CCxNE = 1, enable OCx and OCxN outputs) 0 = Disable (Disable OCx and OCxN output or force to idle state)	TIM1BKR[7]	0x29B	RW−0
T1AOE	Main output auto-control 1 = T1MOE is automatically set to 1 when the next update event arrives (when the break input is inactive) or set to 1 by software 0 = T1MOE can only be set to 1 by software	TIM1BKR[6]		RW−0
T1BKP	Fault source break input TIM1_BKIN polarity 1 = active at high level 0 = active at low level	TIM1BKR[5]		RW−0

[4] When the break input is valid, this bit will be cleared by hardware asynchronously.

<table>
<tr><td>T1BKE</td><td colspan="4">break input (BRK)
1 = Enable
0 = Disable</td><td>TIM1BKR[4]</td><td></td><td>RW−0</td></tr>
<tr><td>T1OSSR</td><td colspan="4">Output "off state" selection in operating mode (when T1MOE = 1)
For details, see **Table 7-14 Timer1 Output Control and Status**</td><td>TIM1BKR[3]</td><td></td><td>RW−0</td></tr>
<tr><td>T1OSSI</td><td colspan="4">Output "off state" selection in idle mode (when T1MOE=0)
For details, see **Table 7-14 Timer1 Output Control and Status**</td><td>TIM1BKR[2]</td><td></td><td>RW−0</td></tr>
<tr><td rowspan="4">T1LOCK [5]</td><td colspan="4">Lock settings (Write protect, prevent software errors)</td><td rowspan="4">TIM1BKR[1:0]</td><td rowspan="4"></td><td rowspan="4">RW−00</td></tr>
<tr><td>00</td><td>01</td><td>10</td><td>11</td></tr>
<tr><td>Disable</td><td>Locking level 1</td><td>Locking level 2</td><td>Locking level 3</td></tr>
<tr><td>No write protection for registers</td><td>T1BKE, T1BKP, T1AOE, T1OISx, T1OISxN, T1DTG</td><td>Includes level 1, T1CCxP, T1CCxNP, T1OSSR, T1OSSI</td><td>Includes level 2, T1OCxM, T1OCxPE</td></tr>
</table>

Table 7-13 Timer1 Main Output Enable, Braking and Locking Level Registers

[5] The LOCK bit can only be written once after system reset, and once it is written, its content will be remain unchanged until reset.

Controls					Output Status	
T1MOE	T1OSSI	T1OSSR	T1CCxE	T1CCxNE	OCx Output Status	OCxN Output Status
1	x	0	0	0	OCx = 0 (Output disabled)	OCxN = 0 (Output disabled)
		0	0	1	OCx = 0 (Output disabled)	OCxN = OCxREF ^ T1CCxNP
		0	1	0	OCx = OCxREF ^ T1CCxP	OCxN = 0 (Output disabled)
		0	1	1	OCx = OCxREF ^ T1CCxP + Deadband	OCxN = Complementary signals for OCxREF ^ T1CCxNP + Deadband
		1	0	0	OCx = T1CCxP (Output disabled)	OCxN =T1CCxNP (Output disabled)
		1	0	1	OCx = T1CCxP	OCxN = OCxREF ^ T1CCxNP
		1	1	0	OCx = OCxREF ^ T1CCxP	OCxN =T1CCxNP
		1	1	1	OCx = OCxREF ^ T1CCxP + Deadband	OCxN = Complementary signals for OCxREF ^ T1CCxNP + Deadband
0	0	x	0	0	OCx = T1CCxP (Output disabled)	OCxN = T1CCxNP (Output disabled)
	0		0	1	In deadband: OCx = T1CCxP, OCxN = T1CCxNP (Output disabled) After deadband: OCx = T1OISx, OCxN = T1OISxN (Output disabled)	
	0		1	0		
	0		1	1		
	1		0	0	OCx = T1CCxP (Output disabled)	OCxN = T1CCxNP (Output disabled)
	1		0	1	In deadband: OCx = T1CCxP, OCxN = T1CCxNP (Output inactive values) After deadband: OCx = T1OISx, OCxN = T1OISxN	
	1		1	0		
	1		1	1		

Table 7-14 Timer1 Output Control and Status

Name	Control			Register	Addr.	Reset
T1DTG	Setting of Deadband generator			TIM1DTR[7:0]	0x29C	RW−0000 0000
	T1DTG[7:0]	DT(Deadband Duration Time)	t_{DTG}			
	0xxxxxxx	T1DTG[7:0] x t_{DTG}	$T_{Fmaster}$ (f1)			
	10xxxxxx	(64+T1DTG[5:0]) x t_{DTG}	2 x $T_{Fmaster}$ (f2)			
	110xxxxx	(32+T1DTG[4:0]) x t_{DTG}	8 x $T_{Fmaster}$ (f3)			
	111xxxxx	(32+T1DTG[4:0]) x t_{DTG}	16 x $T_{Fmaster}$ (f4)			
	* Fmaster is the clock source of TIM1 eg. When $T_{Fmaster}$ = 125 ns (8 MHz) ,Dead-Time is as follows:					
	T1DTG[7:0]	Dead-Time (μs)	Step Time			
	0 ~ 7Fh	0 ~ 15.875	125 ns (f1)			
	80h ~ BFh	16 ~ 31.75	250 ns (f2)			
	C0h ~ DFh	32 ~ 63	1 μs (f3)			
	E0h ~ FFh	64 ~ 126	2 μs (f4)			

Table 7-15 Timer1 Complementary Output Deadband Configuration

Name	Status	Register	Addr.	Reset
T1OIS4	When T1MOE=0, Channel 4(OC4) output in idle state 1 = OC4 output 1 0 = OC4 output 0	TIM1OISR[6]	0x29D	RW−0
T1OIS3	When T1MOE=0, Channel 3/2/1(OCx) output in idle state 1 = After Dead-Time, OCx output 1 0 = After Dead-Time, OCx output 0	TIM1OISR[4]		RW−0
T1OIS2		TIM1OISR[2]		RW−0
T1OIS1		TIM1OISR[0]		RW−0
T1OIS3N	When T1MOE=0, complementary Channel 3/2/1(OCxN) output in idle state 1 = After Dead-Time, OCxN output 1 0 = After Dead-Time, OCxN output 0	TIM1OISR[5]		RW−0
T1OIS2N		TIM1OISR[3]		RW−0
T1OIS1N		TIM1OISR[1]		RW−0

Table 7-16 Timer1 Channel Output Register in Idle Status

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt 1 = Enable (PEIE, T1BIE, T1BG, T1TIE, T1CCxIE, T1CCxG, T1UIE apply) 0 = Global Shutdown (Wake-Up not affected)		INTCON[7]	Bank first addres s+0x0 B	RW−0
PEIE	Peripheral Interrupt Enable 1 = Enable (T1BIE, T1BG, T1TIE, T1CCxIE, T1CCxG, T1UIE apply) 0 = Disable (no Wake-Up)		INTCON[6]		RW−0
T1BIE	Break Interruption	1 = Enable 0 = Disable	TIM1IER[7]	0x215	RW−0
T1BG [6]	Break software interruption		TIM1EGR[7]	0x218	WO−0
T1BIF [7]	Break Interrupt Flag 1 = Active level detected on break input 0 = No break event		TIM1SR1[7]	0x216	R_W1C−0
T1TIE	Trigger Interrupt 1 = Enable 0 = Disable		TIM1IER[6]	0x215	RW−0
T1TIF [7]	Trigger Interrupt Flag 1 = Triggered 0 = No Trigger event		TIM1SR1[6]	0x216	R_W1C−0
T1CC4IE	Channel 4 Capture/Compare Interrupt	1 = Enable 0 = Disable	TIM1IER[4]	0x215	RW−0
T1CC3IE	Channel 3 Capture/Compare Interrupt		TIM1IER[3]		RW−0
T1CC2IE	Channel 2 Capture/Compare Interrupt		TIM1IER[2]		RW−0
T1CC1IE	Channel 1 Capture/Compare Interrupt		TIM1IER[1]		RW−0
T1CC4G [6]	Channel 4		TIM1EGR[4]	0x218	WO−0

[6] The software is set to 1, and the hardware is automatically cleared to 0.

[7] Write '1' to clear, and writing '0' has no effect on the bit value.. It is recommended that only STR and MOVWI instructions be used for write operations, not BSR or IOR instructions.

Name	Status		Register	Addr.	Reset
	Capture/Compare Software Interrupt				
T1CC3G [6]	Channel 3 Capture/Compare Software Interrupt		TIM1EGR[3]		WO−0
T1CC2G [6]	Channel 2 Capture/Compare Software Interrupt		TIM1EGR[2]		WO−0
T1CC1G [6]	Channel 1 Capture/Compare Software Interrupt		TIM1EGR[1]		WO−0
T1CC4IF [7]	Channel x Capture/Compare Interrupt Flag - Output mode: 1 = T1CNT matches T1CCRx value 0 = Mismatch - Input mode: 1 = Count value captured to T1CCRx (Auto resume when reading T1CCRx) 0 = No Capture event		TIM1SR1[4]	0x216	R_W1C−0
T1CC3IF [7]			TIM1SR1[3]		R_W1C−0
T1CC2IF [7]			TIM1SR1[2]		R_W1C−0
T1CC1IF [7]			TIM1SR1[1]		R_W1C−0
T1CC4OF [7]	Channel x Repeat Capture Interrupt Flag 1 = Repeated capture occurs (when the count value is captured to the T1CCRx register, T1CCxIF has been set to 1) 0 = No repeated capture		TIM1SR2[4]	0x217	R_W1C−0
T1CC3OF [7]			TIM1SR2[3]		R_W1C−0
T1CC2OF [7]			TIM1SR2[2]		R_W1C−0
T1CC1OF [7]			TIM1SR2[1]		R_W1C−0
T1UIE	Allow update interrupts	1 = Enable 0 = Disable	TIM1IER[0]	0x215	RW−0
T1UIF [7]	Update event interrupt flag	1 = Update event occurs 0 = No update event	TIM1SR1[0]	0x216	R_W1C−0

Table 7-17 Timer1 Interrupt Enable and Status Bits

7.2.2. Counting basic units

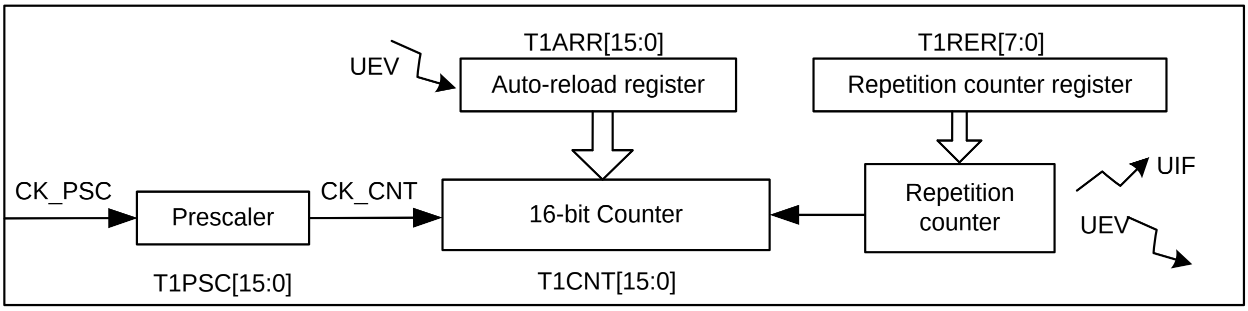


Figure 7-3 Counting basic units

TIM1 base units:

- 16-bit up, down or up/down counter
- 16-bit prescaler
- Repeat counter
- 16-bit auto-reload register

The prescaler, repeat counter, output compare and the auto-reload register consist of the preload register and the shadow register, respectively.

	Prescaler	Repeat Counter	Output Compare value	Auto-reload register
Pre-loaded enable	Enabled by default when T1CEN = 1		T1OCxPE	T1ARPE
Pre-loaded Registers	T1PSC[15:0]	T1REP[7:0]	T1CCRx[15:0]	T1ARR[15:0]

Table 7-18 Update event related preload registers

The prescaled clock (CK_PSC) source is available as follows (see **Section 7.2.3** Clock/Trigger Controller).

- Internal clock source (Fmaster)
- Filtered external channel trigger input (TI1FP1, TI2FP2)

The 16-bit prescaler divides the prescaler clock (CK_PSC) by 1 ~ 65536 to generate the counter clock (CK_CNT).

Frequency division Equation: $f_{CK\_CNT} = f_{CK\_PSC} / (PSCR[15:0] + 1)$; (PSCR is the value of prescaler shadow register)

When T1UDIS=0, update events are allowed to occur. The update event source (see "T1URS") are as follows:

- Counter overflow or underflow (when T1REP=0, please refer to **Section 7.2.2.2** Repetition Counter)
- Trigger event is generated in reset mode

When an update event is generated, the update event flag bit T1UIF is set. Whether to trigger an interrupt and/or wake up from SLEEP depends on the corresponding enable control bits (GIE, PEIE and T1UIE).

In addition, depending on the configuration, update events can trigger the following conditions:

1. Related to prescaler, repetition counter, output compare value and auto reload register:

   (1) When the counter is enabled (T1CEN = 1) and its corresponding preload is enabled (T1OCxPE / T1ARPE = 1, as in **Table 7-18**), its shadow register will be updated to the preload value when an update event is generated, as in **Figure 7-4**.

   (2) When the counter is disabled (T1CEN = 0), or its corresponding preload is disabled (T1OCxPE / T1ARPE = 0), its shadow register will be updated directly to the preload value.

2. In one-pulse mode, when an update event is generated, the counter is automatically turned off (T1CEN = 0) and the counter stops counting.

3. When the fault-break and the auto output (T1AOE = 1) are enabled, the PWM will resume normal output after an update event is generated when the fault event is withdrawn.

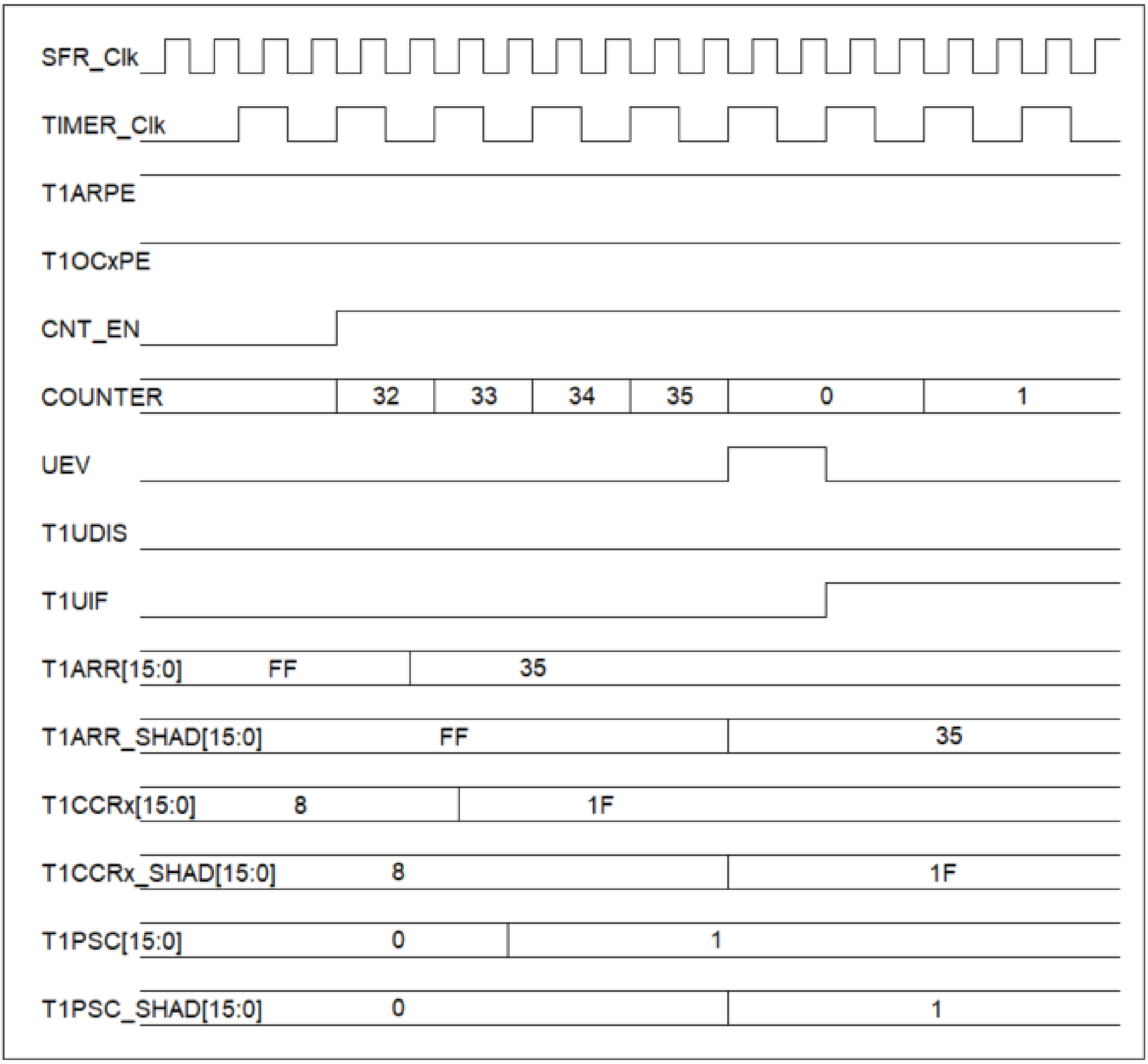


Figure 7-4 Update timing diagram of the preload register under update event

7.2.2.1. Count mode

- **Up count mode** (T1CMS = 00 and T1DIR = 0): Counter counts upward from 0. When T1CNT = T1ARR, an overflow event is generated, and then the count starts from 0 again.

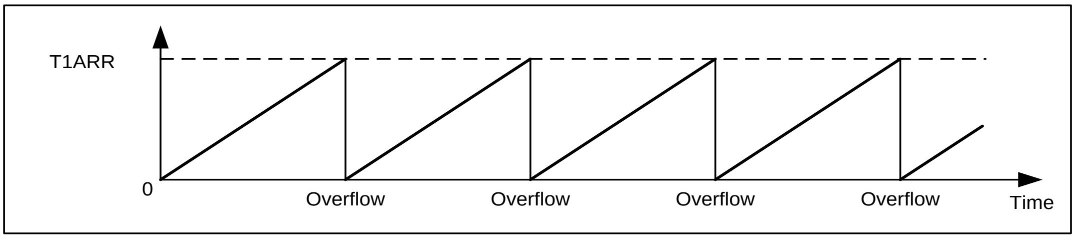


Figure 7-5 Up count mode

- **Down count mode** (T1CMS = 00 and T1DIR = 1): The counter starts counting down from the value of T1ARR, generates an underflow event when T1CNT = 0, and then starts counting again from the value of T1ARR.

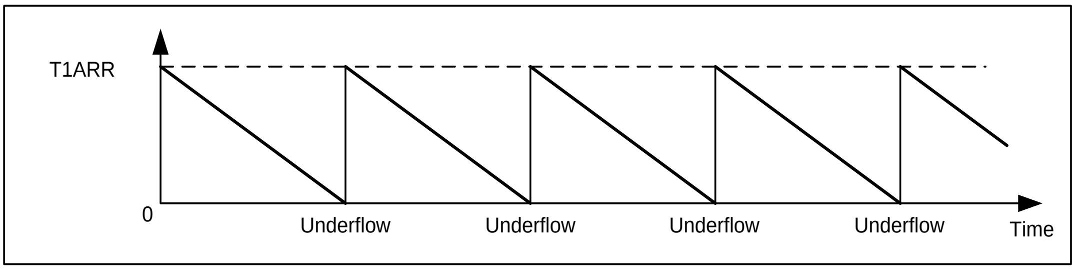


Figure 7-6 Down count mode

- **Center-aligned mode** (Upward/downward counting, T1CMS ≠ 00): The initial direction of counting depends on the T1DIR register value (as shown in **Figure 7-7** and **Figure 7-8**), and the initial value of counting is T1CNT. If T1DIR is initialized to 0, the counter starts counting up from T1CNT and generates an overflow event when T1CNT = T1ARR; then the counter starts counting down from the value of T1ARR and generates a downflow event when T1CNT = 0. And then the counter starts counting up from 0 and keeps repeating the above process.

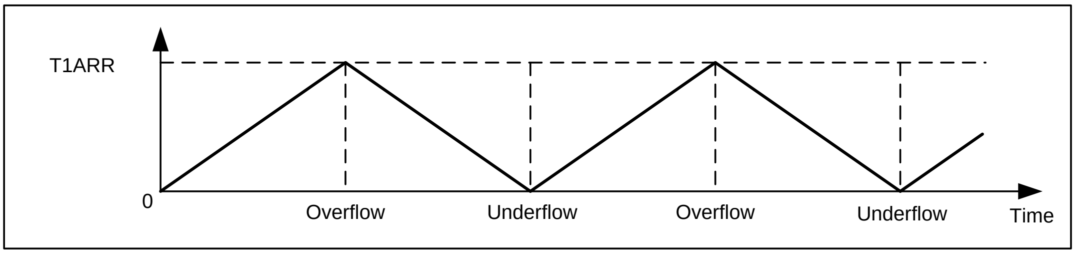


Figure 7-7 Center-aligned mode (T1DIR initialized to 0)

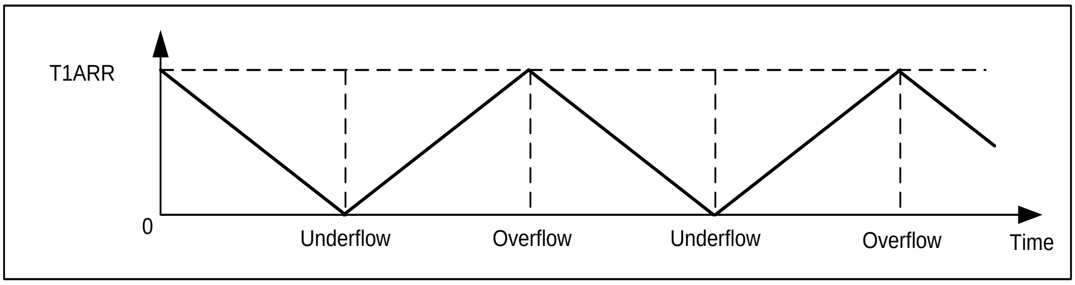


Figure 7-8 Center-aligned mode (T1DIR initialized to 1)

Example of configuration steps:

1. Enable the TIM1 module clock (TIM1EN = 1), and select the TIM1 clock source (T1CKSRC);
2. If necessary, enable the counting cycle preload function (T1ARPE=1);
3. Configure the value of counting cycle (T1ARR);
4. Configure the counting direction as up or down (T1DIR);

5. Configure the count mode as edge-aligned mode or center-aligned mode (T1CMS);
6. Configure prescaler (T1PSC);
7. Enable counter (T1CEN = 1).

Note:

1. It is recommended to read and write the value of counter T1CNT[15:0] when the counter is stopped (T1CEN = 0) to avoid errors.
2. Software cannot rewrite the T1CMS and T1DIR bits at the same time; when T1CMS = 00, T1DIR is a readable and writable register; when T1CMS ≠ 00, T1DIR is a read-only register, and the counting direction is automatically set by hardware after counting is started.
3. In center-aligned mode, it needs to set the initial count value T1CNT ≤ T1ARR.
4. It is necessary to configure the period, output compare value, count mode and other registers first, and configure the prescaler register before enabling the counter (T1CEN = 1) .

7.2.2.2. Repetition Counter

When the shadow register (RCR) of the 8-bit repetition counter is not 0, it will be automatically decremented by 1 when the following events occur:

- Up count mode, per count overflow event;
- Down count mode, per count underflow event;
- Center-aligned mode, per count overflow or underflow event.

The update event (UEV) is generated only when the repetition counter is decremented to 0, i.e., the frequency of the update event can be set by the repetition counter. In addition, the counting cycle T1ARR, duty cycle T1CCRx and other configurations can be changed in the update event interrupt handler, which is useful when generating a specific number of PWM signals (see **Section 7.2.4.2** PWM Mode).

When an update event occurs, its shadow register (RCR) will be automatically updated to the preloaded T1REP value.

Note: When T1REP is configured and its value is not 0, it is recommended to turn on the Update Event Interrupt after the first update event, and its shadow register (RCR) will be reloaded to the T1REP value only when the next update event (UEV) occurs.

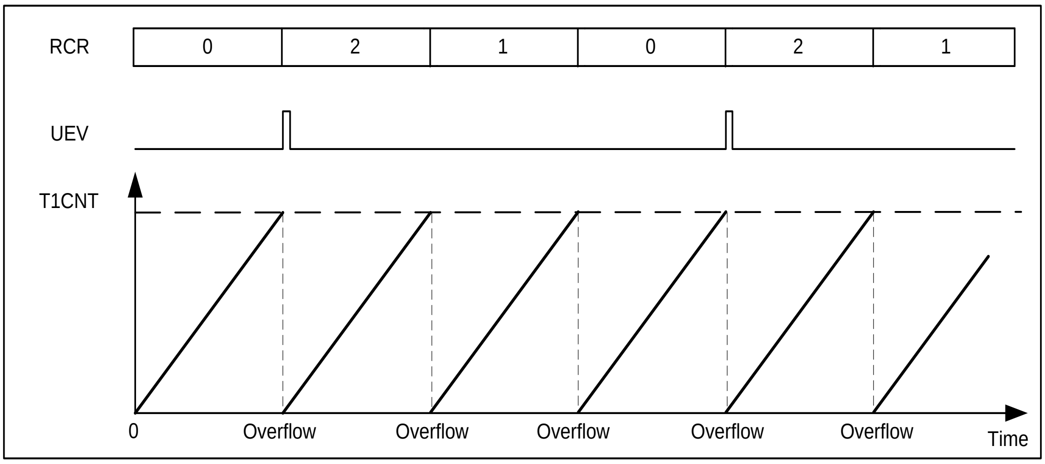


Figure 7-9 Repetition counter timing diagram (when T1REP = 2)

7.2.3. Clock/Trigger Controller

The clock/trigger controller includes the counter's clock source, trigger source, and mode control.

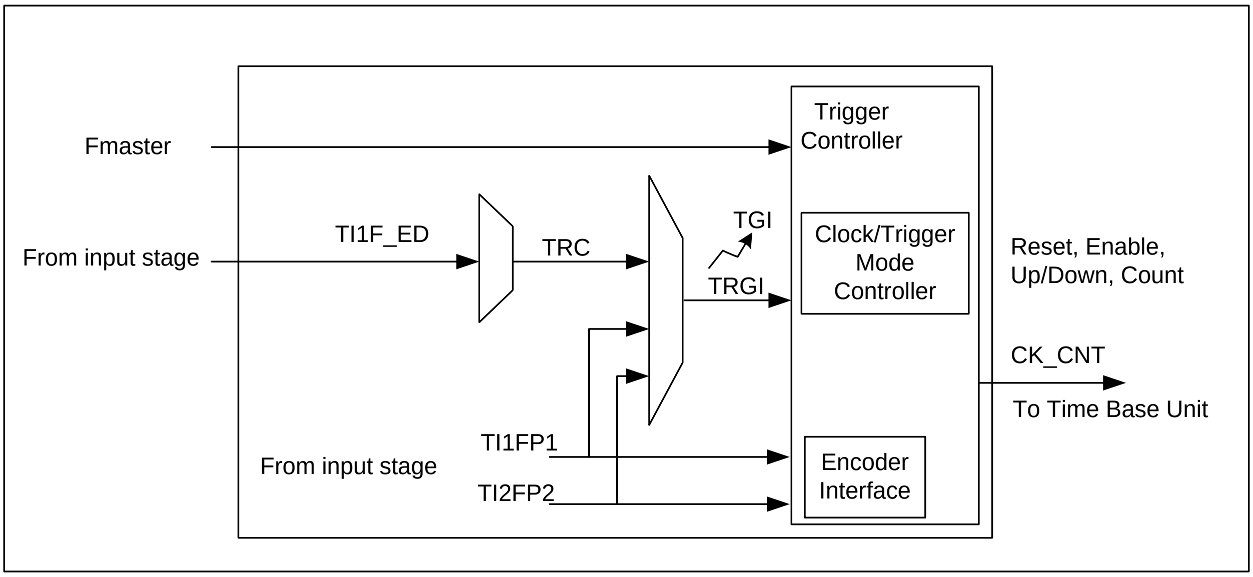


Figure 7-10 Clock/trigger controller Block Diagram

7.2.3.1. Counter Clock Sources (Fmaster)

When T1SMS = 000, the counter is driven by the internal clock with the following optional 6 clock sources (see T1CKSRC):

- Sysclk
- 1x or 2x HIRC
- LIRC
- 1x or 2x external clock (Valid only if FOSC is configured to LP, XT or EC mode accordingly.)

7.2.3.2. Counter trigger source

When T1SMS = 100/101/110 (Slave mode), the counter is driven by the trigger source (TRGI), and its optional 3 trigger event sources (see T1TS) are as follows:

- Channel 1 Input Edge Detector for TI1 (TI1F_ED)
- Filtered Channel 1 Input (TI1FP1)
- Filtered Channel 2 Input (TI2FP2)

Note:

1. See T1CC1P/T1CC2P for the polarity of the active edge/active level of the trigger input;
2. When a trigger event occurs, the trigger interrupt flag T1TIF is set. Whether to trigger an interrupt and/or wake up from SLEEP depends on the corresponding enable control bits (GIE, PEIE and T1TIE).

7.2.3.3. Counting control mode

The TIM1 counting control mode (refer to T1SMS) allows, in addition to the internal clock mode, to select the reset mode, the gating mode and the trigger mode when configured as input capture mode.

- **Internal Clock Mode** (T1SMS = 000):

  The counter is driven by the internal clock (Fmaster).

- **Reset Mode** (T1SMS = 100):

  The counter starts to count normally driven by the internal clock until a active edge occurs on the trigger input (TRGI), at which t point the counter is cleared and starts counting again from 0. At the same time, the trigger flag T1TIF is set. In addition, when T1UDIS = 0 and T1URS = 0, an update event will be generated.

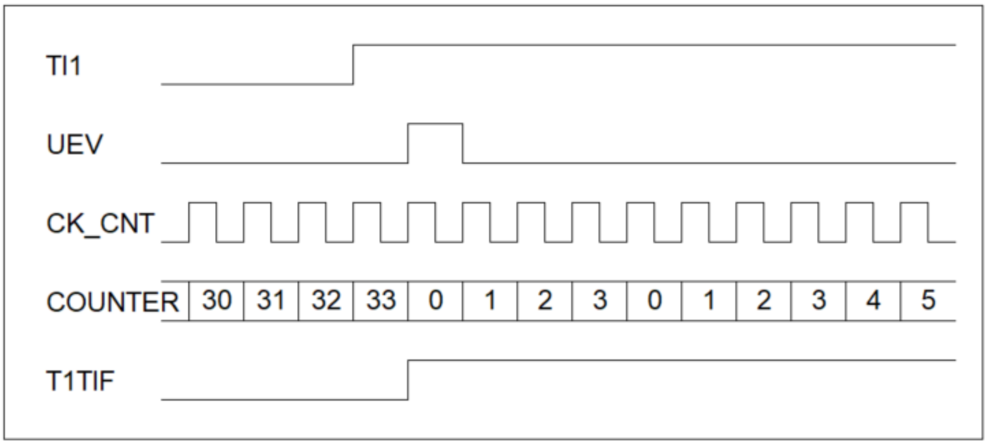


Figure 7-11 Reset mode, counter timing diagram (channel TI1 is selected and the active edge of the trigger input is the rising edge)

- **Gating Mode** (T1SMS = 101):

The counter is driven to count by the internal clock during the active level of the trigger input (TRGI), and stops counting during the inactive level, but does not reset. Trigger flag T1TIF is set when the counter starts or stops.

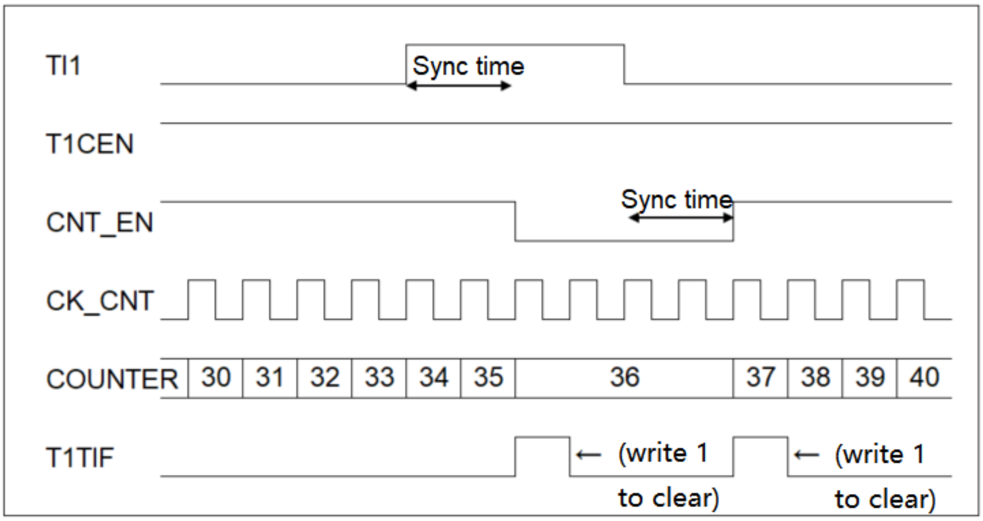


Figure 7-12 Gating mode, counter timing diagram (select Channel TI1 and trigger input active level is low)

- **Trigger Mode** (T1SMS = 110):

  The counter is driven by the internal clock on the active edge of the trigger input (TRGI) and is not reset. At the same time, the trigger flag T1TIF is set.

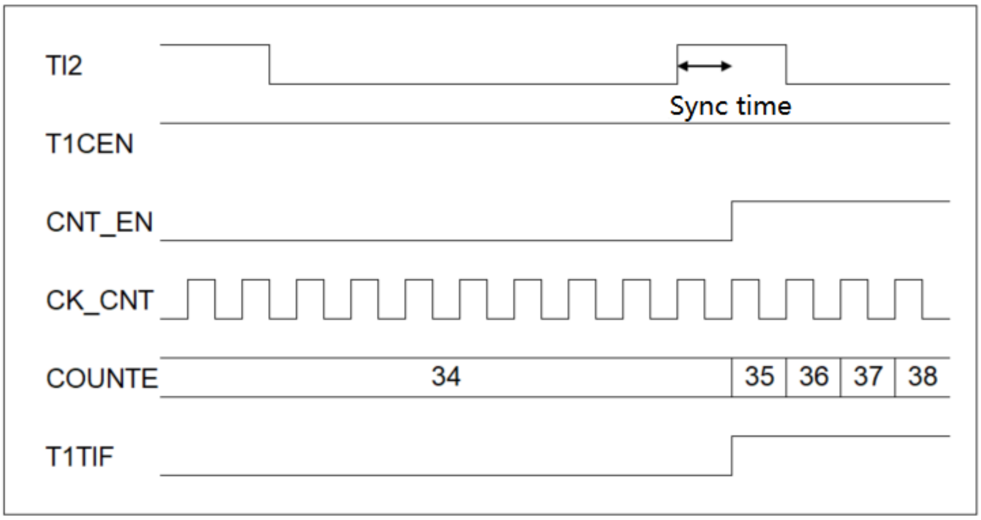


Figure 7-13 Trigger mode, counter timing diagram (select Channel TI2 and the active edge of the trigger input is the rising edge)

Example of configuration steps for control modes:

1. Enable the TIM1 module clock (TIM1EN = 1), and select the TIM1 clock source (T1CKSRC).
2. Configure the selected channel port as the input (TRISx = 1);
3. Configure input capture sampling frequency and filter length for Channel x (T1ICxF);

4. Configure input capture prescaler for Channel x (T1ICxPSC);
5. Select the input capture channel (T1CC1S/T1CC2S) as needed;
6. Configure the active edge or active level of the trigger input (T1CC1P/T1CC2P);
7. Select the counting control mode as reset mode, gating mode or trigger mode (T1SMS), and select the trigger input source (T1TS);
8. Enable counter (T1CEN = 1);

Program Example 1 (Take Gating mode as an example, please see the timing diagram shown in **Figure 7-12**):

```
BANKSEL PCKEN
BSR PCKEN,0                 ; Enable TIM1 module clock
BANKSEL TCKSRC
LDWI 01H
STR TCKSRC                  ; Select TIM1 clock source as HIRC
BANKSEL TRISA
LDWI 01H
STR TRISA                   ; Configure Channel1 PORT PA0 as input
BANKSEL TIM1CCMR1
LDWI 01H
STR TIM1CCMR1               ; Configure Channel1 input capture filter length, prescaler, and the input
pin is mapped on TI1FP1
LDWI 55H
STR TIM1SMCR                ; Configure TIM1 as gating control mode, trigger source is TI1FP1
LDWI 03H
STR TIM1CCER1               ; Enable Channel1 as input, and trigger input active level is low level
BANKSEL TIM1CR1
BSR TIM1CR1,0               ; Enable counter
BTSS TIM1SR1,6              ; Determine whether the trigger interrupt flag is high
LJUMP $-1
BCR TIM1SR1,6               ; Clear the trigger interrupt flag bit
```

7.2.4. Capture/Compare Channels

The CH1~4 PORTS of TIM1 can be configured as Input Capture or Output Compare function (see the T1CCxS bit of the multiplexing register TIM1CCMRx).

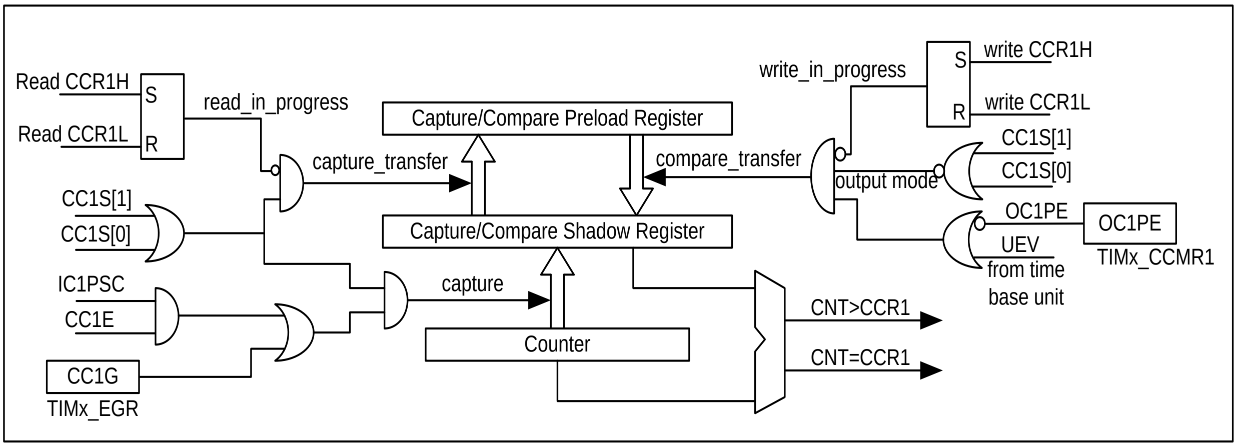


Figure 7-14 Capture /Compare Channel1 Block Diagram

The T1CCRx registers consist of a preload register and a shadow register. Read /Write process only operate on preload registers.

- In Input Capture mode:

  T1CCRx[15:0] is a read-only register. When a capture event occurs, the captured counter value is written to the shadow register and then copied into the T1CCRx preload register.

  When reading the T1CCRx[15:0] register, the MSB must be read first, followed by the LSB. When the MSB are read, the preload register is frozen, and then the correct LSB can be read. The preload register can only be updated to the latest captured value after the LSB are read.

- In Output Compare mode:

  T1CCRx[15:0] is a readable and writable register. During Write operation, the T1CCRx preload register value is copied into the shadow register (see **Section 7.2.2**), and then the content of the shadow register is compared with the counter. The value read during Read operation comes from the preload register.

7.2.4.1. Input capture mode

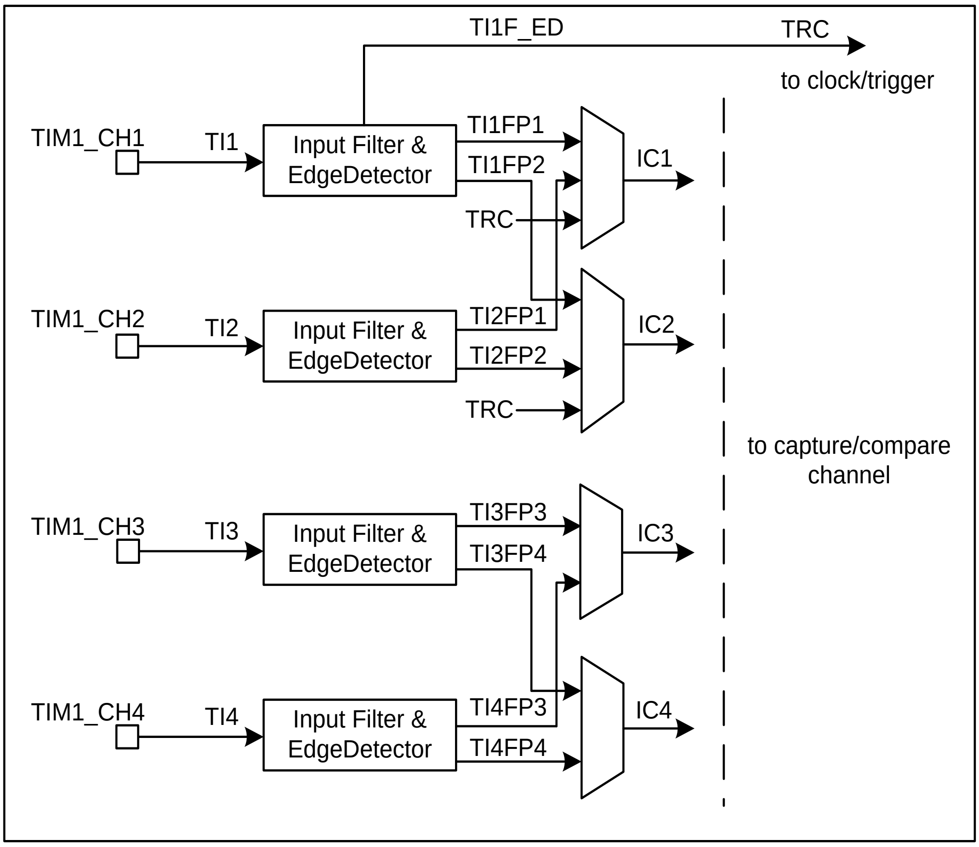


Figure 7-15 Input capture channel Block Diagram

In input capture mode, when an input capture event occurs in channel x, the current count value will be captured into the T1CCRx[15:0] register, and the input capture flag T1CCxIF will be set. If an input capture event occurs again while T1CCxIF remains 1, the repeat capture flag bit T1CCxOF will be set. Whether to trigger an interrupt and/or wake up from SLEEP depends on the corresponding enable controls (GIE, PEIE and T1CCxIE). In addition, the input capture software interrupt T1CCxG can be enabled to trigger an interrupt.

The input capture sources for each channel of TIM1 (see T1CCxS) are as follows:

T1CCxS	Channel1	Channel2	Channel3	Channel4
01	TI1FP1	TI2FP2	TI3FP3	TI4FP4
10	TI2FP1	TI1FP2	TI4FP3	TI3FP4
11	TRC	TRC	-	-

Table 7-19 Input capture sources for each channel

Note: When the capture source of channel x (x = 1/2/3/4) is selected as the input capture signal of the corresponding I/O of other channels, the other channels need to be set as inputs. For example, if channel 1 selects TI2FP1 (T1CC1S = 10), channel 2 must be set as input (T1CC2S = 01 or 10); channel 3 and channel 4 as above.

Signal Name	Detailed description
TIM1_CH1/2/3/4	I/O input corresponding to Channel1/2/3/4
IC1/2/3/4	Capture source via selected channel
TI1FP1	Channel1 corresponds to the input capture signal of the I/O, as one of the capture sources of Channel1
TI1FP2	Channel1 corresponds to the input capture signal of I/O, as one of the capture sources of Channel2
TI2FP2	Channel2 corresponds to the input capture signal of the I/O as one of the capture sources of Channel2
TI2FP1	Channel2 corresponds to the input capture signal of I/O, as one of the capture sources of Channel1
TI3FP3	Channel3 corresponds to the input capture signal of the I/O as one of the capture sources of Channel3
TI3FP4	Channel3 corresponds to the input capture signal of I/O, as one of the capture sources of channel4
TI4FP4	Channel4 corresponds to the input capture signal of I/O, as one of the capture sources of channel4
TI4FP3	Channel4 corresponds to the input capture signal of I/O, as one of the capture sources of Channel3
TRC	Channel1 corresponds to the input double-edge capture signal of I/O, as one of the capture sources of channels 1 and 2

Table 7-20 Input Capture Signal Description

Example of configuration steps for an input capture channel:

1. Enable the TIM1 module clock (TIM1EN = 1), and select the TIM1 clock source (T1CKSRC).
2. Configure the selected channel PORT as the input (TRISx = 1).
3. Select input capture source (T1CCxS).
4. Configure the polarity of the capture source (T1CCxP).
5. Configure the capture sampling frequency and filter length of channel x (T1ICxF[3:0]) and capture prescaler (T1IC1PSC[1:0]).
6. Input capture interrupts (GIE, PEIE, T1CCxIE) can be enabled as needed.
7. Enable capture channel (T1CCxE = 1).
8. Enable counter (T1CEN = 1).

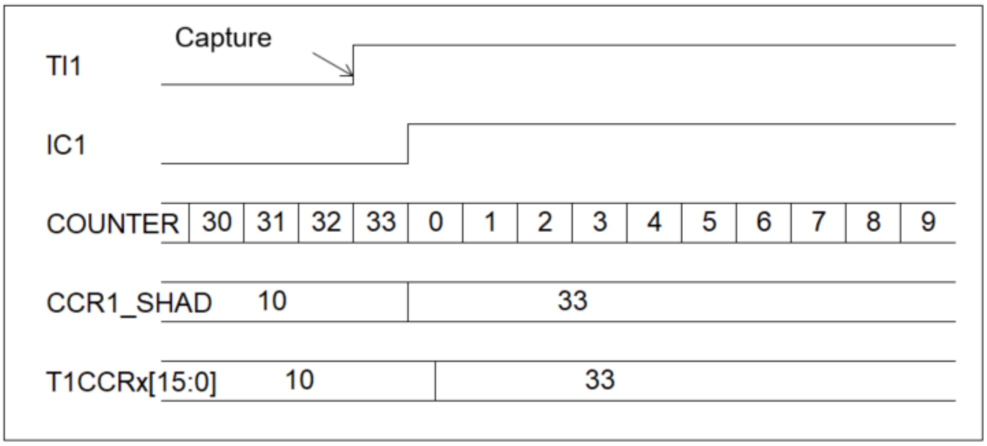


Figure 7-16 Input Capture Timing Diagram

PWM signal measurement application: the cycle and duty cycle of PWM signal can be measured by using input capture mode and reset mode, and selecting the input capture source of two channels as the PWM signal input of the same channel.

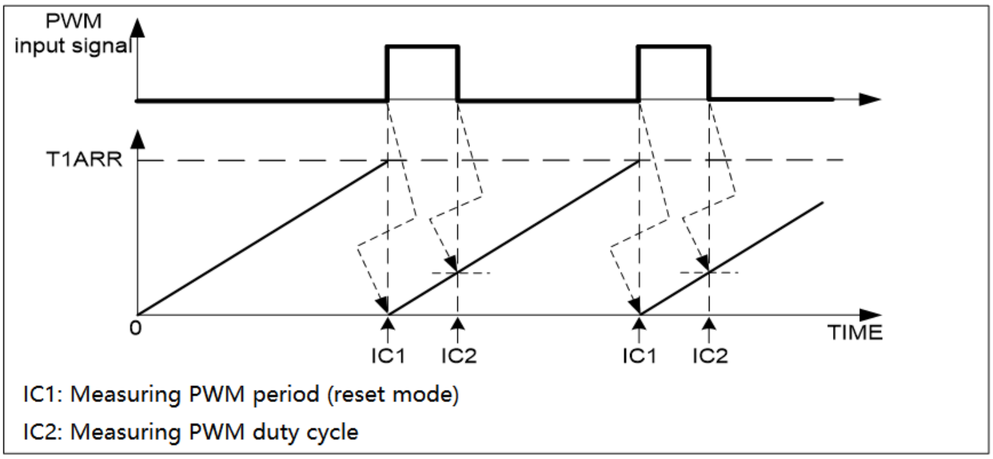

PWM
input signal
T1ARR
0
TIME
IC1
IC2
IC1
IC2
IC1: Measuring PWM period (reset mode)
IC2: Measuring PWM duty cycle

Figure 7-17 Schematic Diagram of Measuring PWM Signal

Example of configuration steps to measure PWM:

1. Enable the TIM1 module clock (TIM1EN = 1), and select the TIM1 clock source (T1CKSRC);
2. Configure the PORT corresponding to Channel1/2 as input (TRISx = 1);
3. Select the input capture source (T1CCxS), map IC1 of Channel1 to TI1FP1, and map IC2 of Channel2 to TI2FP1;
4. Configure the polarity of the capture source, configure Channel1 as rising edge (T1CC1P = 0) and Channel2 as falling edge (T1CC2P = 1);
5. Configure capture sampling frequency and filter length (T1ICxF[3:0] = 0000), capture prescaler (T1IC1PSC[1:0] = 00);
6. Configure the counting control mode as reset mode (T1SMS = 101), and the counting trigger source as TI1FP1 (T1TS = 101);

7. Enable the input capture function of Channel1 and Channel2 (T1CC1E=1 and T1CC2E=1);

8. Enable counter (T1CEN = 1);

Note: Since the capture edge precedes the reset trigger source by two count clock cycles, in order to obtain an accurate measurement value, the software needs to do the following:

- When Prescaler = 0, PWM period = T1CCR1H/L+2, Duty Cycle = T1CCR2H/L+2.
- When Prescaler = 1, PWM period = T1CCR1H/L+1, Duty Cycle = T1CCR2H/L+1.
- When Prescaler ≥ 1, PWM period = T1CCR1H/L, Duty Cycle = T1CCR2H/L.

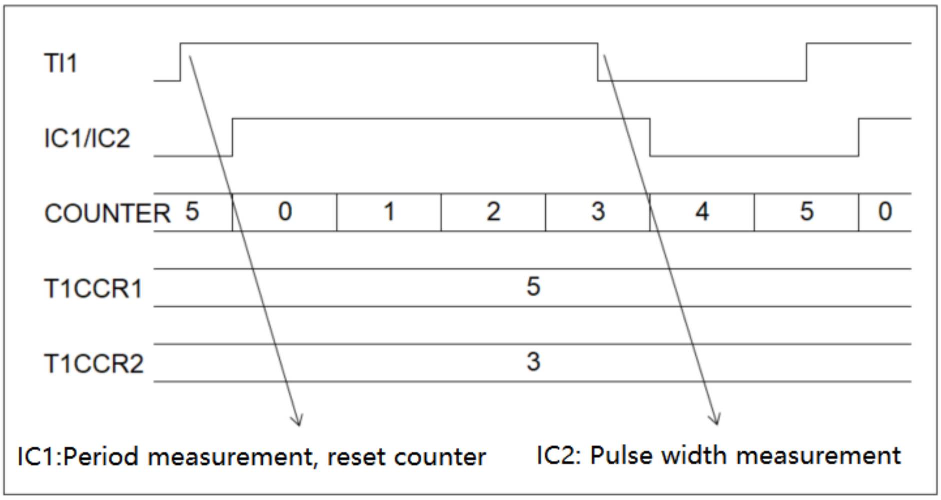

TI1
IC1/IC2
COUNTER
5
0
1
2
3
4
5
0
T1CCR1
5
T1CCR2
3
IC1:Period measurement, reset counter
IC2: Pulse width measurement

Figure 7-18 Timing diagram for measuring PWM signal

7.2.4.2. Output compare mode

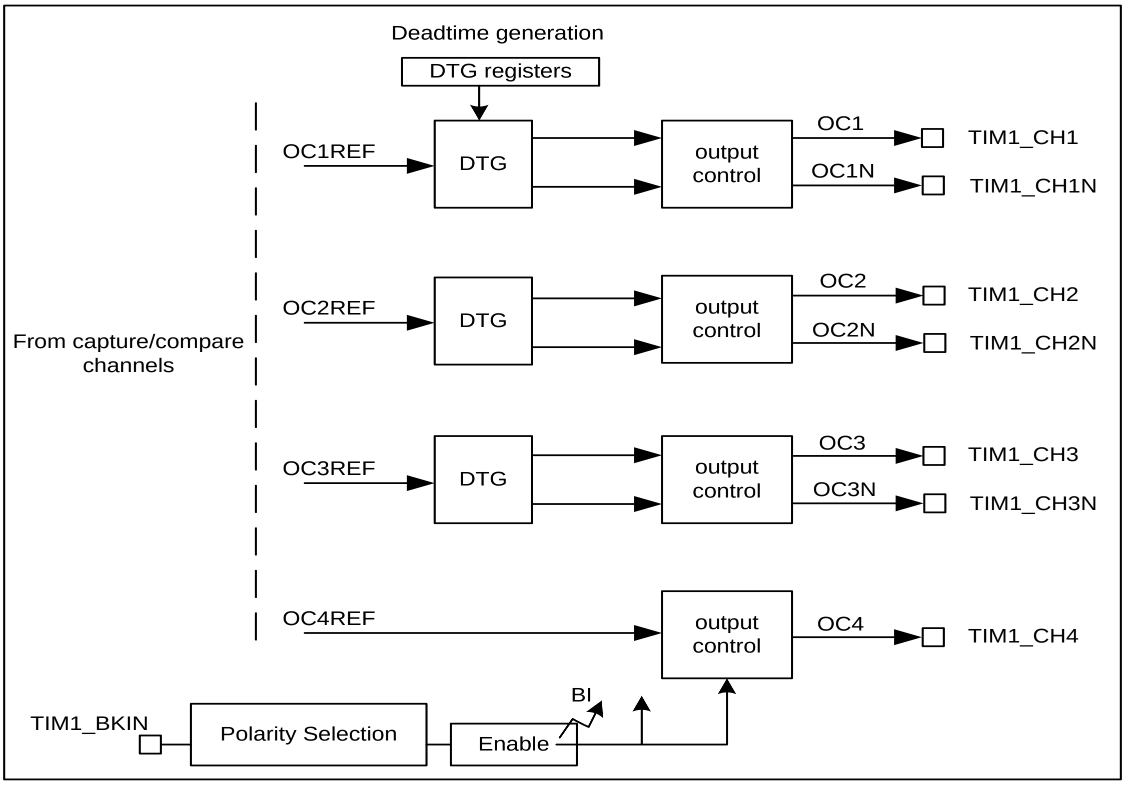


Figure 7-19 Block Diagram of Output Compare Channel

The output compare module compares the count value (T1CNT) with the comparison value (shadow register CCRx_SHAD), first generates the output reference signal OCxREF (active high), and then sends it to the Deadband generation module or break module, and then outputs the waveform to the port (see **Table 7-14**) through polarity selection and other output controls (see T1MOE, T1OSSI, T1OSSR, T1CCxE and T1CCxNE).

The reference signal OCxREF can be configured to 8 output modes through T1OCxM[2:0] (see **Table 7-9**):

1. Frozen mode (T1OCxM = 000): OCxREF value remains unchanged.

2. Active level on match (T1OCxM = 001): OCxREF = 1 when T1CNT = CCRx_SHAD.

3. Inactive level on match (T1OCxM = 010): OCxREF = 0 when T1CNT = CCRx_SHAD.

4. Toggle on match (T1OCxM = 011): When T1CNT = CCRx_SHAD, the OCxREF value is reversed.

5. Forced inactive (T1OCxM = 100): OCxREF is always 0.

6. Forced active (T1OCxM = 101): OCxREF is always 1.

7. PWM1 mode (T1OCxM = 110).

   When T1CNT < CCRx_SHAD, OCxREF = 1; when T1CNT > CCRx_SHAD, OCxREF = 0.

8. PWM2 mode (T1OCxM = 111).

   When T1CNT < CCRx_SHAD, OCxREF = 0; when T1CNT > CCRx_SHAD, OCxREF = 1.

When T1CNT matches CCRx_SHAD, the output compare flag T1CCxIF is set. Whether to trigger an interrupt and/or wake up from SLEEP depends on the corresponding enable controls (GIE, PEIE, T1CCxIE). In addition, the output compare software Interrupt T1CCxG, can be enabled to trigger an interrupt.

Configuration steps for the output compare channel:

1. Enable the TIM1 module clock (TIM1EN = 1), and select the TIM1 clock source (T1CKSRC).
2. Configure the PORT corresponding to the channel as an output (TRISx = 0).
3. Configure the cycle (T1ARR) and comparison value (T1CCRx) of the output waveform.
4. Configure output compare mode (T1OCxM) and output polarity (T1CCxP).
5. Output compare interrupts (GIE, PEIE, T1CCxIE) can be enabled as needed.
6. Enable output compare channel (T1CCxE = 1).
7. Enable the main output automatic control (T1AOE = 1), that is, the hardware will automatically enable the main output (T1MOE) when an update event occurs.
8. Enable the counter (T1CEN = 1).

Note: It is recommended to configure the output compare value T1CCRx ≤ the counting cycle value T1ARR.

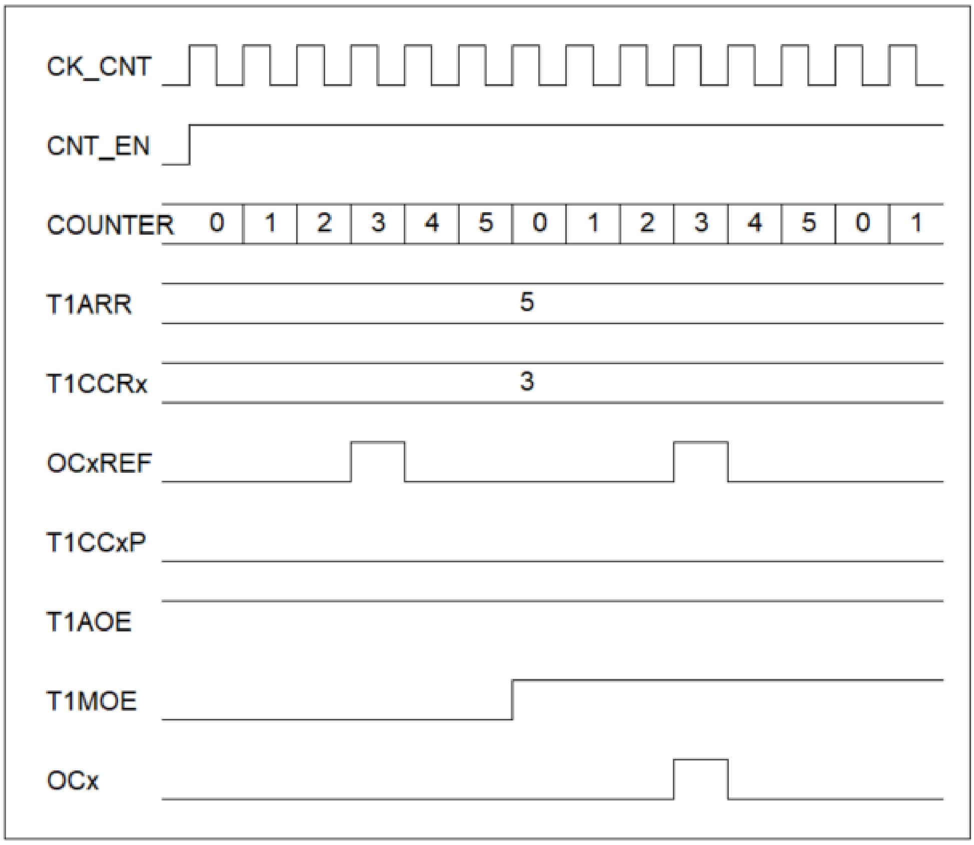


Figure 7-20 Output Timing Diagram with Active Level on Match

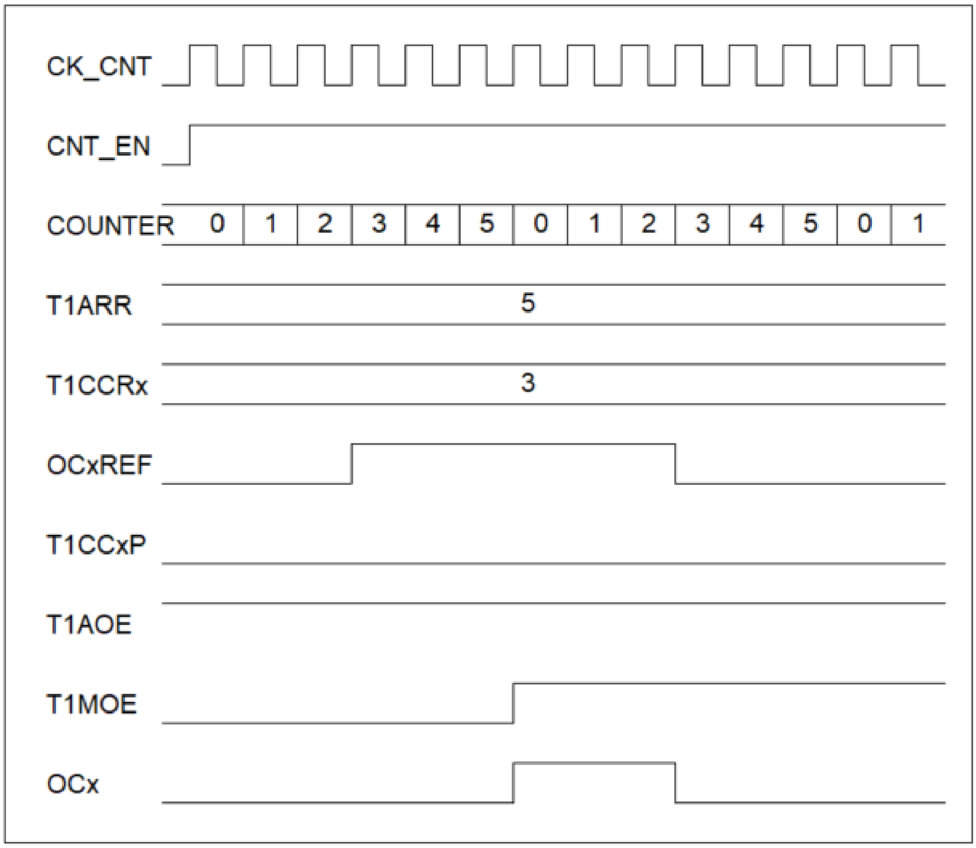


Figure 7-21 Output Timing Diagram when Toggle on Match

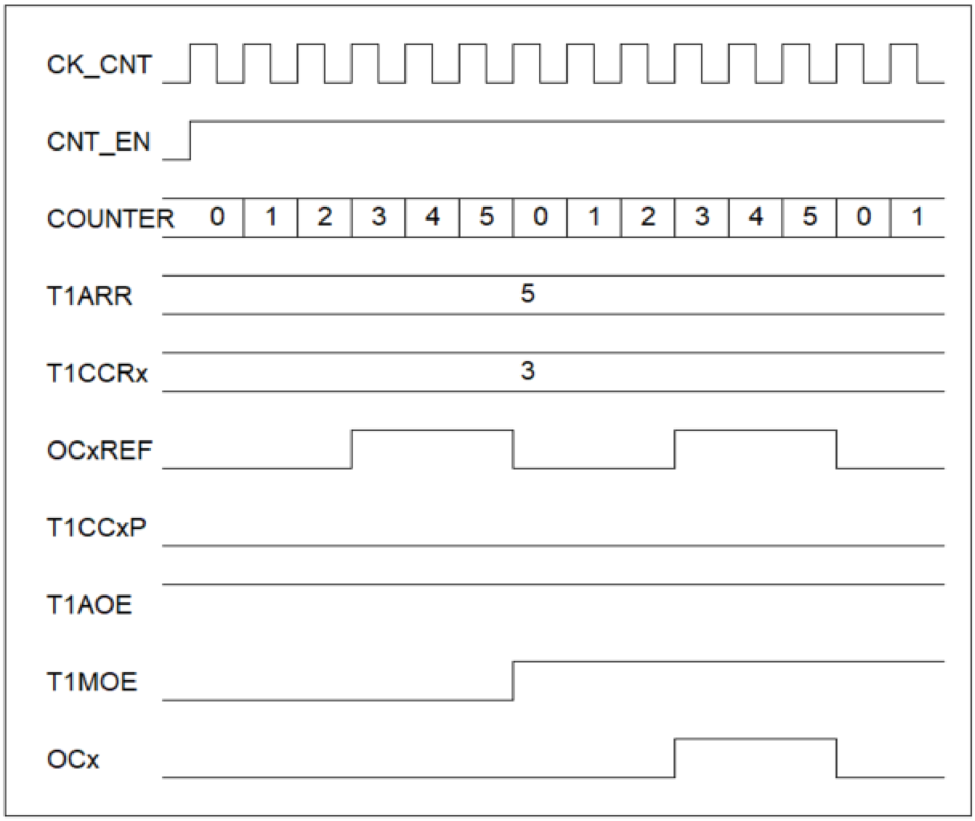


Figure 7-22 Output Timing Diagram in PWM2 Mode

PWM Mode – PWM1/PWM2 cycle is determined by T1ARR and the duty cycle is determined by T1CCRx.

Equation 7-1 *PWM1/2 Cycle = (T1ARR+1) * T*$_{CK\_CNT}$

Equation 7-2 *PWM1/2 Duty Cycle = T1CCRx ÷ (T1ARR+1)*

TIM1_CH1/2/3/4 channels can independently enable output PWM signal, among which CH1/2/3 have complementary output function. The polarity of the output signal as well as the complementary output signal is optional (see T1CCxP/ T1CCxNP). When the output channel and the complementary output channel are enabled simultaneously (T1CCxE = 1, T1CCxNE = 1), the Deadband function will be automatically enabled, and the Dead-Time can be set(see T1DTG), i.e. whenever the falling edge of output signal (OCx or complementary output OCxN) occurs, the rising edge of the other signal will be delayed by the length of deadband.

Note: In PWM mode, the auto-preload function of channel x duty cycle must be enabled (T1OCxPE = 1);

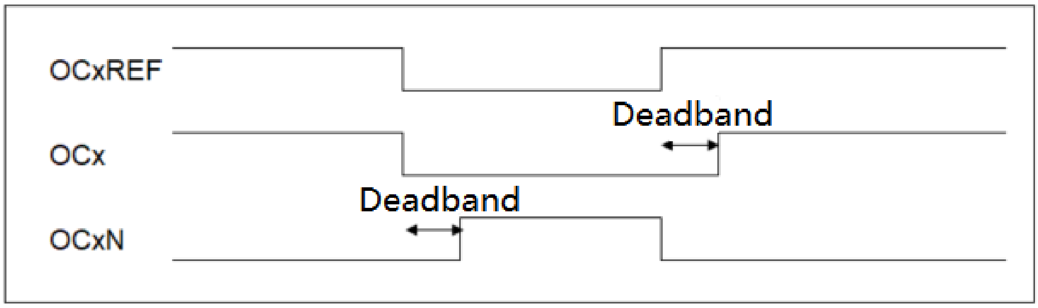


Figure 7-23 Complementary Output with Deadband Timing Diagram

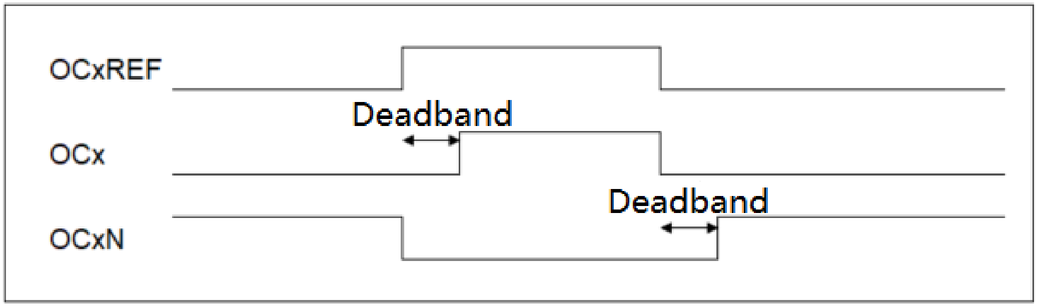


Figure 7-24 Positive Output with Deadband Timing Diagram

When the pulse time of the OCxREF output is shorter than the deadband, a certain pulse signal may be covered by the deadband, resulting in the output unchanged.

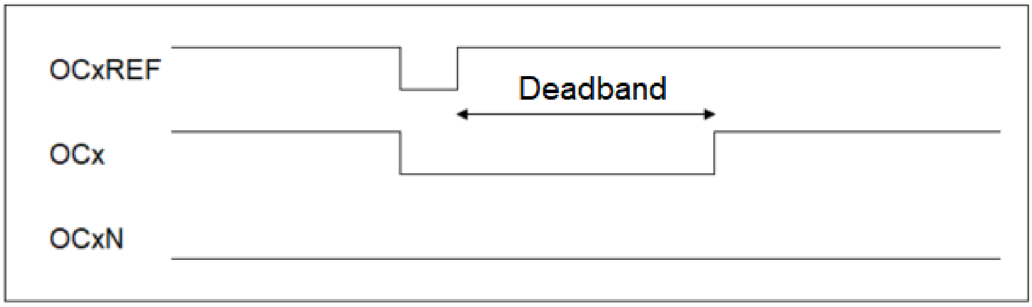


Figure 7-25 Complementary Output Covered by Deadband Timing Diagram

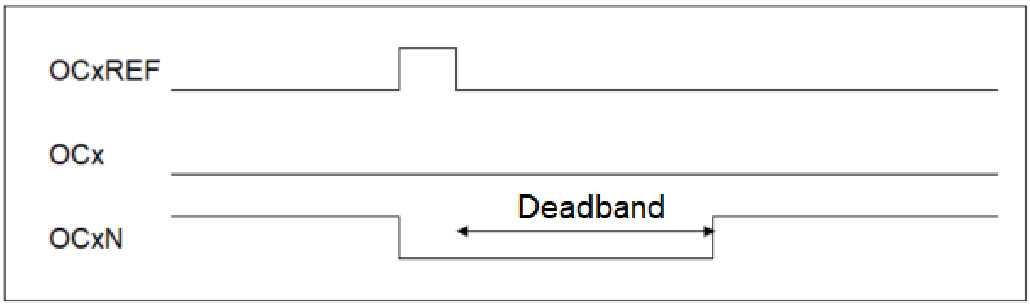


Figure 7-26 Positive Output Covered by Deadband Timing Diagram

The PWM mode is used in conjunction with functions such as repetition counter, update event, cycle preload, and duty cycle preload to generate a specific number of PWM signals.

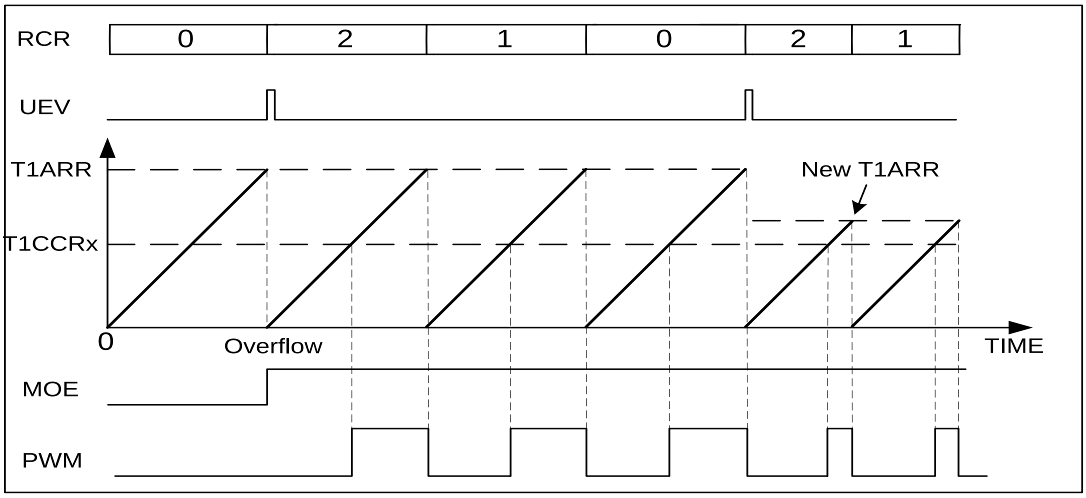


Figure 7-27 Timing Diagram of Outputting 3 Specific PWMs by Repeation Counter

Program Example:

```
BANKSEL PCKEN
BSR PCKEN,0            ; Enable TIM1 module clock
BANKSEL INTCON
LDWI C0H
STR INTCON             ; Enable global and peripheral interrupts
BANKSEL TCKSRC
LDWI 01H
STR TCKSRC             ; Select TIM1 clock source as HIRC
BANKSEL TRISA
LDWI 00H
STR TRISA              ; Configure Channel1 PORT PA0 as output
BANKSEL TIM1ARRL
LDWI 1FH
STR TIM1ARRL           ; Configure the output waveform cycle to 32
LDWI 10H
STR TIM1CCR1L          ; Configure the output waveform duty cycle to 16
```

```
LDWI 02H
STR TIM1RCR                     ; Configure the repetition counter to 2
BANKSEL TIM1CCMR1
LDWI 70H
STR TIM1CCMR1                   ; Configure Channel1 as PWM2 mode output
BSR TIM1IER,0                   ; Enable update event interrupt
LDWI 01H
STR TIM1CCER1                   ; Enable Channel1 and select polarity
BANKSEL TIM1BKR
BSR TIM1BKR,6                   ; Turn on the main output auto-enable bit
BANKSEL TIM1CR1
LDWI 81H
STR TIM1CR1                     ; Enable cycle preload function and enable counter

INT :
BANKSEL TIM1ARRL
LDWI 14H
STR TIM1ARRL                    ; Reconfigure the output waveform cycle to 20
```

One-Pulse Mode – In One-Pulse Mode (T1OPM = 1), when the next update event occurs, the hardware turns off the counter enable (T1CEN = 0) automatically and the counter stops counting.

To generate a correct pulse, the initial value (T1CNT) of the counter must be different from the comparison value (T1CCRx). That is, the following configurations must be met before starting counting:

- Up count mode: T1CNT < T1CCRx ≤ T1ARR
- Down count mode: T1CNT > T1CCRx

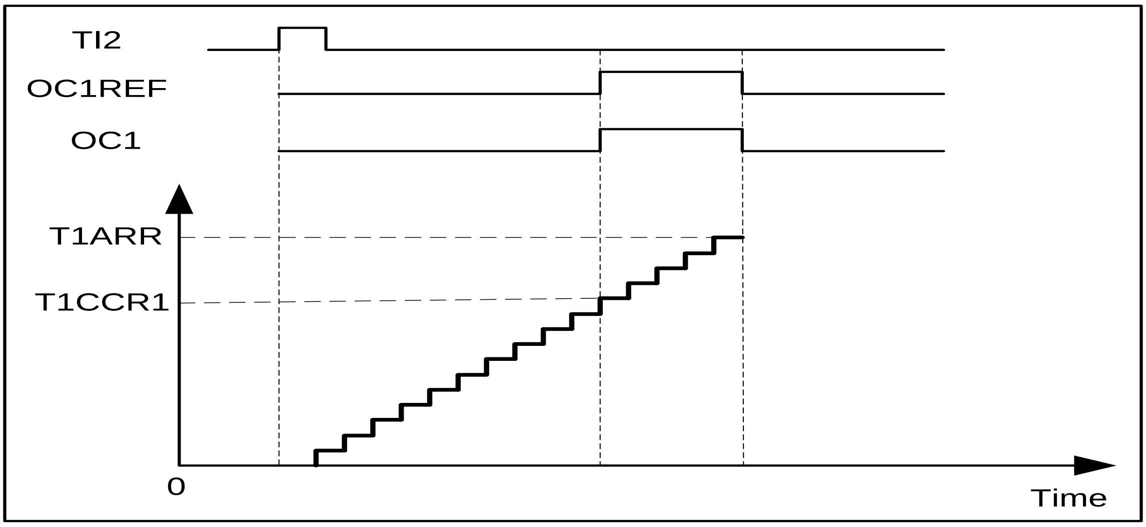


Figure 7-28 Schematic diagram of One-Pulse application

The One-Pulse mode can cooperate with the trigger mode to generate a One-Pulse output at a specific time point, as shown in the figure above. The configuration steps are as follows:

1. Enable the TIM1 module clock (TIM1EN = 1), and select the TIM1 clock source (T1CKSRC).
2. Configure the PORT corresponding to Channel2 as input and the PORT corresponding to Channel1 as

output.

3. Map the input of Channel2 to TI2FP2 (T1CC2S = 01) and select it as rising edge capture (T1CC2P = 0).
4. Configure the counting control mode as trigger mode (T1SMS = 110), and select TI2FP2 (T1TS = 110) as the trigger source.
5. Configure Channel1 as an output channel (T1CC1S = 00).
6. Configure Channel1 as PWM2 mode in output compare mode (T1OC1M = 111) ,with output polarity configured as active at high level (T1CC1P = 0).
7. Enable the input capture function of Channel2 (T1CC2E=1) and the output compare function of Channel1 (T1CC1E).
8. Enable the main output automatic control (T1AOE = 1), that is, the hardware will automatically enable the main output (T1MOE) when an update event occurs.
9. Enable the counter (T1CEN = 1).

7.2.4.3. Fault-Break function

All 4 PWMs support Fault-Break function. When the break input function is enabled (T1BKE = 1), PWM will output a preset condition according to its setting upon a Fault-break event. The PWM will be in this condition as long as the break condition is valid.

Fault-Break event of TIM1 can be one of the followings (see BKS):

- BKIN Event
- LVD Event
- ADC Threshold Comparison Event

When a Fault-Break event occurs, the PWM Output and Complementary Output status (see **Table 7-14**) are as follows:

- If the clock source of TIM1 is disabled, T1MOE will be cleared asynchronously, forcing the output to be inactive;
- If the clock source of TIM1 is enabled, T1MOE will be cleared asynchronously, the output will be in an inactive state during the deadband, and will be in an idle state after the deadband (in this case, the real deadband is 2 CK_CNT clocks longer than the setting value);

Meanwhile, the break interrupt flag T1BIF is set. Whether to trigger an interrupt and/or wake up from SLEEP depends on the corresponding enable controls (GIE, PEIE, T1BIE). In addition, the break software interrupt T1BG can be enabled to trigger an interrupt.

After the Fault event is cancelled, if the main output automatic control T1AOE = 1, then T1MOE will be automatically set by hardware when the next update event (UEV) arrives (2 CK_CNT clocks need to be synchronized), and the PWM will resume normal output. Otherwise, T1MOE needs to be set by software to resume the output.

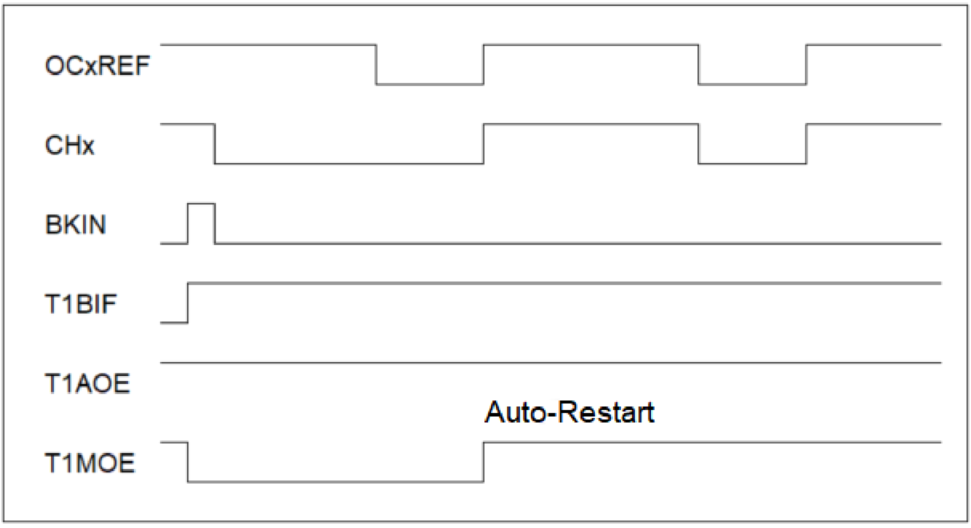


Figure 7-29 PWM Auto-Restart Diagram

7.3. General-purpose TIMER2

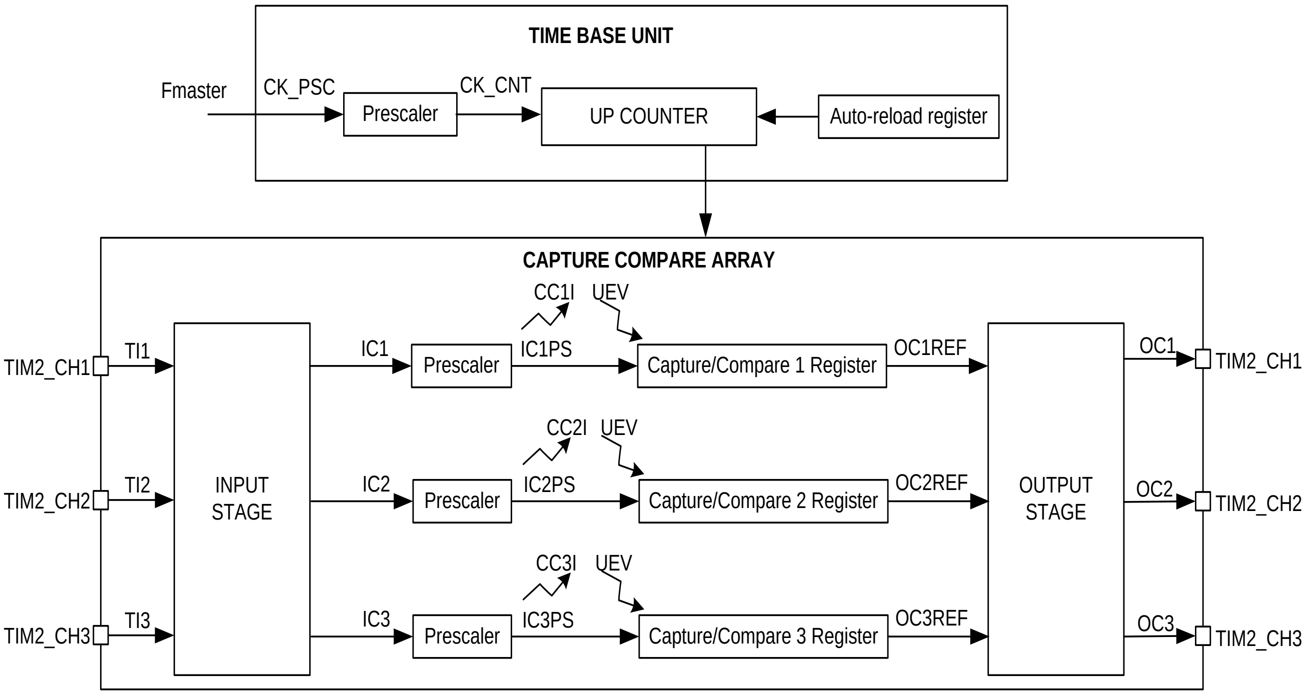


Figure 7-30 TIM2 Block Diagram

TIM2 features:

- 16-bit up counter supporting auto-reload;
- 4-bit programmable prescaler;
- 3 channels with selectable polarity support:
  - ✓ Input capture
  - ✓ Output compare

- ✓ Same cycle, independent duty cycle PWM channels
- ✓ One-Pulse mode

- Interrupt events: Update event, input capture, output compare.

7.3.1. Summary of Timer2 Related Registers

Name	Addr.	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset
PCKEN	0x9A	TKEN	I2CEN	UARTEN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
TIM2CR1	0x30C	T2ARPE	—	—	—	T2OPM	T2URS	T2UDIS	T2CEN	0--- 0000
TIM2IER	0x30D	—	—	—	—	T2CC3IE	T2CC2IE	T2CC1IE	T2UIE	---- 0000
TIM2SR1	0x30E	—	—	—	—	T2CC3IF	T2CC2IF	T2CC1IF	T2UIF	---- 0000
TIM2SR2	0x30F	—	—	—	—	T2CC3OF	T2CC2OF	T2CC1OF	—	---- 000-
TIM2EGR	0x310	—	—	—	—	T2CC3G	T2CC2G	T2CC1G	T2UG	---- 0000
TIM2CCMR1 (output mode)	0x311	—	T2OC1M[2:0]			T2OC1PE	—	T2CC1S[1:0]		-000 0-00
TIM2CCMR1 (input mode)		T2IC1F[3:0]				T2IC1PSC[1:0]		T2CC1S[1:0]		0000 0000
TIM2CCMR2 (output mode)	0x312	—	T2OC2M[2:0]			T2OC2PE	—	T2CC2S[1:0]		-000 0-00
TIM2CCMR2 (input mode)		T2IC2F[3:0]				T2IC2PSC[1:0]		T2CC2S[1:0]		0000 0000
TIM2CCMR3 (output mode)	0x313	—	T2OC3M[2:0]			T2OC3PE	—	T2CC3S[1:0]		-000 0-00
TIM2_CCMR3 (input mode)		T2IC3F[3:0]				T2IC3PSC[1:0]		T2CC3S[1:0]		0000 0000
TIM2CCER1	0x314	—	—	T2CC2P	T2CC2E	—	—	T2CC1P	T2CC1E	--00 --00
TIM2CCER2	0x315	—	—	—	—	—	—	T2CC3P	T2CC3E	---- --00
TIM2CNTRH	0x316	T2CNT[15:8]								0000 0000
TIM2CNTRL	0x317	T2CNT[7:0]								0000 0000
TIM2PSCR	0x318	—	—	—	—	T2PSC[3:0]				---- 0000
TIM2ARRH	0x319	T2ARR[15:8]								1111 1111
TIM2ARRL	0x31A	T2ARR[7:0]								1111 1111
TIM2CCR1H	0x31B	T2CCR1[15:8]								0000 0000
TIM2CCR1L	0x31C	T2CCR1[7:0]								0000 0000
TIM2CCR2H	0x31D	T2CCR2[15:8]								0000 0000
TIM2CCR2L	0x31E	T2CCR2[7:0]								0000 0000
TIM2CCR3H	0x29E	T2CCR3[15:8]								0000 0000
TIM2CCR3L	0x29F	T2CCR3[7:0]								0000 0000

Table 7-21 Summary of Timer2 Related registers (Reserved bits must remain as the reset value and cannot be changed)

Name	Status		Register	Addr.	Reset
T2CNT	TIM2 Count Value	MSB	TIM2CNTRH[7:0]	0x316	RW−0000 0000
		LSB	TIM2CNTRL[7:0]	0x317	RW−0000 0000
T2PSC	TIM2 Prescaler		TIM2PSCR[3:0]	0x318	RW−0000
T2ARR	Auto-reload Register for Counting Cycles (Preload value) Note: When the value is 0, the counter does not work;	MSB	TIM2ARRH[7:0]	0x319	RW−1111 1111
		LSB	TIM2ARRL[7:0]	0x31A	RW−1111 1111
T2CCR1	Input Capture Mode: Last Capture Event (IC1) Captured count value	MSB	TIM2CCR1H[7:0]	0x31B	RO−0000 0000
		LSB	TIM2CCR1L[7:0]	0x31C	RO−0000 0000
	Output Compare Mode: output compare value of TIM2_CH1 (Preloaded value)	MSB	TIM2CCR1H[7:0]	0x31B	RW−0000 0000
		LSB	TIM2CCR1L[7:0]	0x31C	RW−0000 0000
T2CCR2	Input Capture Mode: Last Capture Event (IC2) Captured count value	MSB	TIM2CCR2H[7:0]	0x31D	RO−0000 0000
		LSB	TIM2CCR2L[7:0]	0x31E	RO−0000 0000
	Output Compare Mode: output compare value of TIM2_CH2 (Preloaded value)	MSB	TIM2CCR2H[7:0]	0x31D	RW−0000 0000
		LSB	TIM2CCR2L[7:0]	0x31E	RW−0000 0000
T2CCR3	Input Capture Mode: Last Capture Event (IC3) Captured count value	MSB	TIM2CCR3H[7:0]	0x29E	RO−0000 0000
		LSB	TIM2CCR3L[7:0]	0x29F	RO−0000 0000
	Output compare mode: output compare value of TIM2_CH3 (Preloaded value)	MSB	TIM2CCR3H[7:0]	0x29E	RW−0000 0000
		LSB	TIM2CCR3L[7:0]	0x29F	RW−0000 0000

Table 7-22 Timer2 Period Related Registers

Name	Status	Register	Addr.	Reset
TIM2EN	TIM2 Clock 1 = Enable 0 = Disable	PCKEN[2]	0x9A	RW−0
SYSON	In SLEEP mode, the system clock controls 1 = Enable 0 = Disable	CKOCON[7]	0x95	RW−0
T2CKSRC	Timer2 Clock Source (Fmaster) 000 = Sysclk 100 = 2x (XT or EC) (*) 001 = HIRC 101 = LIRC 010 = XT or EC (*) 110 = LP or EC (*) 011 = 2x HIRC 111 = 2x (LP or EC) (*) (*) FOSC should be configured in LP/XT/EC mode accordingly, otherwise the oscillator will not run.	TCKSRC[6:4]	0x31F	RW−000
DTYSEL	TIM1/TIM2 Multiplier Clock Duty Cycle Adjustment Bit 00 = 2ns delay 10 = 4ns delay 01 = 3ns delay 11 = 7ns delay	CKOCON[5:4]	0x95	RW−10
T2ARPE	Auto-Preload of Count Cycles 1 = Enable (The T2ARR preload value is loaded when the update event arrives) 0 = Disable (T2ARR is loaded immediately)	TIM2CR1[7]	0x30C	RW−0
T2OPM	One-Pulse mode 1 = Enable (when the next update event comes, T2CEN is automatically cleared and the counter is stopped) 0 = Disable (counters do not stop when the update event occurs)	TIM2CR1[3]		RW−0
T2URS	When T2UDIS=0, update event source 1 / 0 = Counter Overflows	TIM2CR1[2]		RW−0
T2UDIS	Generate Update Event Control 1 = Disable 0 = Enable	TIM2CR1[1]		RW−0
T2CEN	TIM2 Counter 1 = Enable 0 = Disable	TIM2CR1[0]		RW−0

Table 7-23 Timer2 Relatd User Control Register

Name	Addr.	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	Reset
TIM2CCMR1	0x311	T2IC1F[3:0]				T2IC1PSC[1:0]		T2CC1S[1:0]		RW-0000 0000
TIM2CCMR2	0x312	T2IC2F[3:0]				T2IC2PSC[1:0]		T2CC2S[1:0]		RW-0000 0000
TIM2CCMR3	0x313	T2IC3F[3:0]				T2IC3PSC[1:0]		T2CC3S[1:0]		RW-0000 0000

Name	Status			Register	Addr.	Reset
T2ICxF	<u>Sampling Frequency and Digital Filter Length of Channel x Input capture</u>			TIM2CCMRx[6:4] x = 1, 2, 3	0x311/ 0x312/ 0x313	RW−0000
	Value	Sampling frequency ($f_{SAMPLING}$)	Digital Filter Length (N)			
	0000	<u>Fmaster /2</u>	<u>0</u>			
	0001	Fmaster	2			
	0010	Fmaster	4			
	0011	Fmaster	8			
	0100	Fmaster / 2	6			
	0101	Fmaster / 2	8			
	0110	Fmaster / 4	6			
	0111	Fmaster / 4	8			
	1000	Fmaster / 8	6			
	1001	Fmaster / 8	8			
	1010	Fmaster / 16	5			
	1011	Fmaster / 16	6			
	1100	Fmaster / 16	8			
	1101	Fmaster / 32	5			
	1110	Fmaster / 32	6			
	1111	Fmaster / 32	8			
T2ICxPSC	<u>Channel x Input Capture Prescaler (several events trigger a capture)</u> 00 = <u>1</u> 01 = 2 10 = 4 11 = 8 Note: When T2CCxE = 0, the prescaler is reset to 00			TIM2CCMRx[3:2]		RW−00
T2CC1S [8]	<u>Channel1 Mode</u>	00 = <u>Output</u> 01 = Input, the input pin is mapped		TIM2CCMR1[1:0]	0x311	RW−00

[8] Writable only when channel x is disabled (ie T2CCxE = 0) , x = 1, 2, 3 .

Name	Status		Register	Addr.	Reset
	Selection	to TI1FP1 10 = Input, the input pin is mapped to TI2FP1 11 = Reserved			
T2CC2S [9]	Channel2 Mode Selection	00 = Output 01 = Input, the input pin is mapped to TI2FP2 10 = Input, the input pin is mapped to TI1FP2 11 = Reserved	TIM2CCMR2[1:0]	0x312	RW−00
T2CC3S [9]	Channel3 Mode Selection	00 = Output 01 = Input, the input pin is mapped to TI3FP3 1x = Reserved	TIM2CCMR3[1:0]	0x313	RW−00

Table 7-24 TIM2CCMRx as Input Configuration Register

[9] Writable only when channel x is disabled (ie T2CCxE = 0) , x = 1, 2, 3 .

Name	Addr.	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Bit0	Reset
TIM2CCMR1	0x311	–	T2OC1M[2:0]			T2OC1PE	–	T2CC1S[1:0]		RW–-000 0-00
TIM2CCMR2	0x312	–	T2OC2M[2:0]			T2OC2PE	–	T2CC2S[1:0]		RW–-000 0-00
TIM2CCMR3	0x313	–	T2OC3M[2:0]			T2OC3PE	–	T2CC3S[1:0]		RW–-000 0-00

T2OCxM	Channel x Output Compare Mode		Level of Reference Signal OCxREF
000	Frozen (No Compare)		Remain unchanged
001	When T2CNT = CCRx_SHAD		1
010	When T2CNT = CCRx_SHAD		0
011	When T2CNT = CCRx_SHAD		Level Reversal
100	Forced inactive		0
101	Forced active		1
110	PWM1 mode	T2CNT < CCRx_SHAD	1
		T2CNT > CCRx_SHAD	0
111	PWM2 mode	T2CNT < CCRx_SHAD	0
		T2CNT > CCRx_SHAD	1
Note: The output reference signal OCxREF is active at high level, which together with the polarity selection T2CCxP determines the actual output value of OCx;			

Table 7-25 T2OCxM Configured in Output Compare Mode

Name	Status		Register	Addr.	Reset
T2OCxPE	Auto-preload of Channel x Output Compare value 1 = Enable (T2CCRx preload value is loaded when the update event arrives) 0 = Disable (T2CCRx is loaded immediately)		TIM2CCMRx[3] x = 1, 2, 3	0x311/ 0x312/ 0x313	RW−0
T2CC1S [10]	Channel 1 Mode Selection	00 = Output 01 = Input, the input pin is mapped to TI1FP1 10 = Input, the input pin is mapped to TI2FP1 11 = Reserved	TIM2CCMR1[1:0]	0x311	RW−00
T2CC2S [10]	Channel 2 Mode Selection	00 = Output 01 = Input, the input pin is mapped to TI2FP2 10 = Input, the input pin is mapped to TI1FP2 11 = Reserved	TIM2CCMR2[1:0]	0x312	RW−00
T2CC3S [10]	Channel 3 Mode Selection	00 = Output 01 = Input, the input pin is mapped to TI3FP3 1x = Reserved	TIM2CCMR3[1:0]	0x313	RW−00

Table 7-26 TIM2CCMRx as Output Configuration Register

[10] Writable only when channel x is disabled (ie T2CCxE = 0) , x = 1, 2, 3 .

Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Addr	Reset
TIM2CCER1	–	–	T2CC2P	T2CC2E	–	–	T2CC1P	T2CC1E	0x314	RW---00 --00
TIM2CCER2	–	–	–	–	–	–	T2CC3P	T2CC3E	0x315	RW----- --00

Name	Function	Input Capture Mode (T2CCxS = 01/10)	Output Compare Mode (T2CCxS = 00)
T2CCxP	Channel x Input/Output Polarity Selection	1 = Capture occurs on falling edge or low level of TIxF 0 = Capture occurs on rising edge or high level of TIxF	1 = OCx is active at low level 0 = OCx is active at high level
T2CCxE	Channel x Input/Output Pin Function	1 = Enable the input capture function of the pin 0 = Disable	1 = Enable the OCx output function of the pin 0 = Disable
Note: The channel output level is determined by the values of the T2OISx and T2CCxE bits;			

Table 7-27 Timer2 Channel Output and Polarity Selection

Name	Status	Register	Addr.	Reset
TIM2_CH1	Channel 1 Pin Remapping 1 = PB0 0 = PA5	AFP1[2]	0x19F	RW−0

Table 7-28 Timer2 Channel Pin Remapping

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt 1 = Enable (PEIE, T2CCxIE, T2CCxG, T2UIE apply) 0 = Global Shutdown (Wake-Up not affected)		INTCON[7]	Bank first address+0x0B	RW−0
PEIE	Peripheral Interrupt Enable 1 = Enable (for T2CCxIE, T2CCxG, T2UIE) 0 = Disable (no Wake-Up)		INTCON[6]		RW−0
T2CC3IE	Channel 3 Capture/Compare Interrupt	1 = Enable 0 = Disable	TIM2IER[3]	0x30D	RW−0
T2CC2IE	Channel 2 Capture/Compare Interrupt		TIM2IER[2]		RW−0
T2CC1IE	Channel 1 Capture/Compare Interrupt		TIM2IER[1]		RW−0

Name	Status	Register	Addr.	Reset
T2CC3G [11]	Channel 3 Capture/Compare Software Interrupt	TIM2EGR[3]	0x310	WO−0
T2CC2G [11]	Channel 2 Capture/Compare Software Interrupt	TIM2EGR[2]		WO−0
T2CC1G [11]	Channel 1 Capture/Compare Software Interrupt	TIM2EGR[1]		WO−0
T2CC3IF [12]	Channel x Capture/Compare Interrupt Flag • Output Mode: 1 = The value of T2CNT and T2CCRx match 0 = Mismatch • Input Mode: 1 = Count value has been captured to T2CCRx (auto clear when T2CCRx is read) 0 = No capture occurred	TIM2SR1[3]	0x30E	R_W1C−0
T2CC2IF [12]		TIM2SR1[2]		R_W1C−0
T2CC1IF [12]		TIM2SR1[1]		R_W1C−0
T2CC3OF [12]	Channel x Recapture Interrupt Flag 1 = Recapture occurs (T2CCxIF is already set when the counter value is captured into the T2CCRx register) 0 = No recapture	TIM2SR2[3]	0x30F	R_W1C−0
T2CC2OF [12]		TIM2SR2[2]		R_W1C−0
T2CC1OF [12]		TIM2SR2[1]		R_W1C−0
T2UIE	Allow Update Interrupt 1 = Enable 0 = Disable	TIM2IER[0]	0x30D	RW−0
T2UG [11]	Allow Update Software Interrupt	TIM2EGR[0]	0x310	WO−0
T2UIF [12]	Update Event Interrupt Flag 1 = An update event occurs 0 = No update event	TIM2SR1[0]	0x30E	R_W1C−0

Table 7-29 Timer2 Interrupt Enable and Status Bits

[11] Set to 1 by software, auto clear by hardware.

[12] Write '1' to clear, and writing '0' has no effect on the bit value. It is recommended to only use the STR and MOVWI instructions for Write operations, rather than the BSR or IOR instructions.

7.3.2. Counting basic units

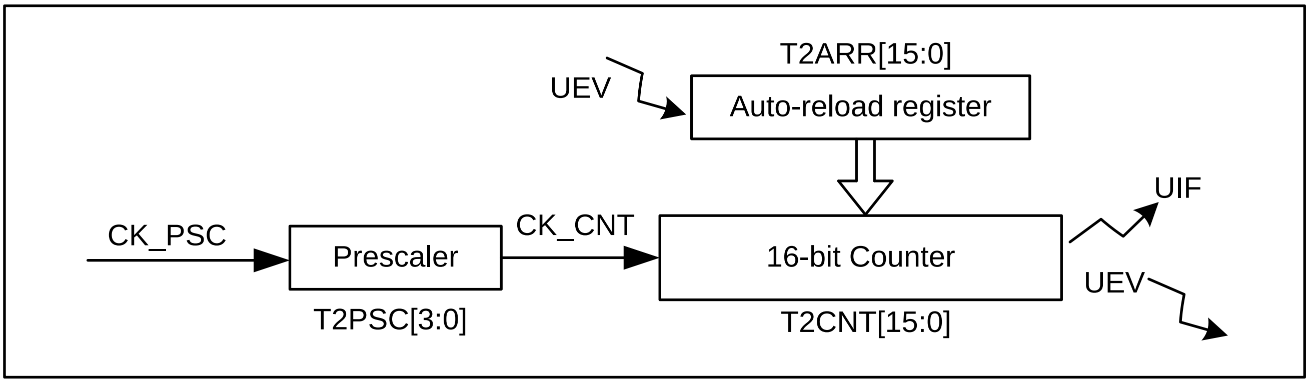


Figure 7-31 Counting basic units

TIM2 basic units:

- 16-bit upward counter
- 4-bit prescaler
- 16-bit auto-reload register

Up count mode: the counter starts counting up from 0, an overflow event occurs when T2CNT = T2ARR, and then the counter starts counting from 0 again.

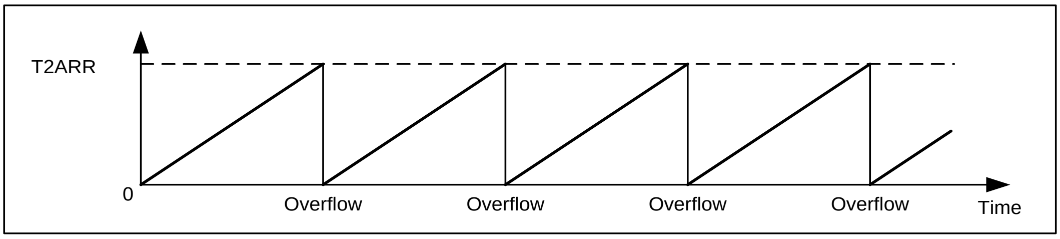


Figure 7-32 Up count mode

The prescaler, output compare, and auto-reload register consist of preload registers and shadow registers, respectively.

	Prescaler	Output Compare value	Auto-reload Register
Preload Enable	Enabled by default when T2CEN = 1	T2OCxPE	T2ARPE
Preload Register	T2PSC[3:0]	T2CCRx[15:0]	T2ARR[15:0]

Table 7-30 Registers with Preload Function

Six optional clock sources for the TIM2 prescaled clock (CK_PSC) (see T 2CKSRC) are as follows:

- Sysclk
- 1x or 2x HIRC
- LIRC

- 1x or 2x external clock (Only valid when FOSC is configured in LP , XT or EC mode accordingly)

The 4-bit prescaler can divide the prescaler clock (CK_PSC) by 1 to 32768 to generate the count clock (CK_CNT).

Frequency division Equation : $f_{CK\_CNT} = f_{CK\_PSC} / 2^{(PSCR[3:0])}$; (PSCR is the value of prescaler shadow register)

Note:

1. It is recommended to read and write the counter value T2CNT[15:0] when the counter is stopped (T2CEN = 0) to avoid errors.
2. It is necessary to configure the cycle, output compare value and other registers first, and configure the prescaler register before enabling the counter (T2CEN = 1).

When T2UDIS = 0, it is allowed to generate update events. The update event source (see "T1URS") is as follows:

- Counter overflows

When an update event occurs, the update event flag T2UIF is set, and whether to trigger an Interrupt and /or wake up from SLEEP depends on the corresponding enable controls (GIE, PEIE and T2UIE).

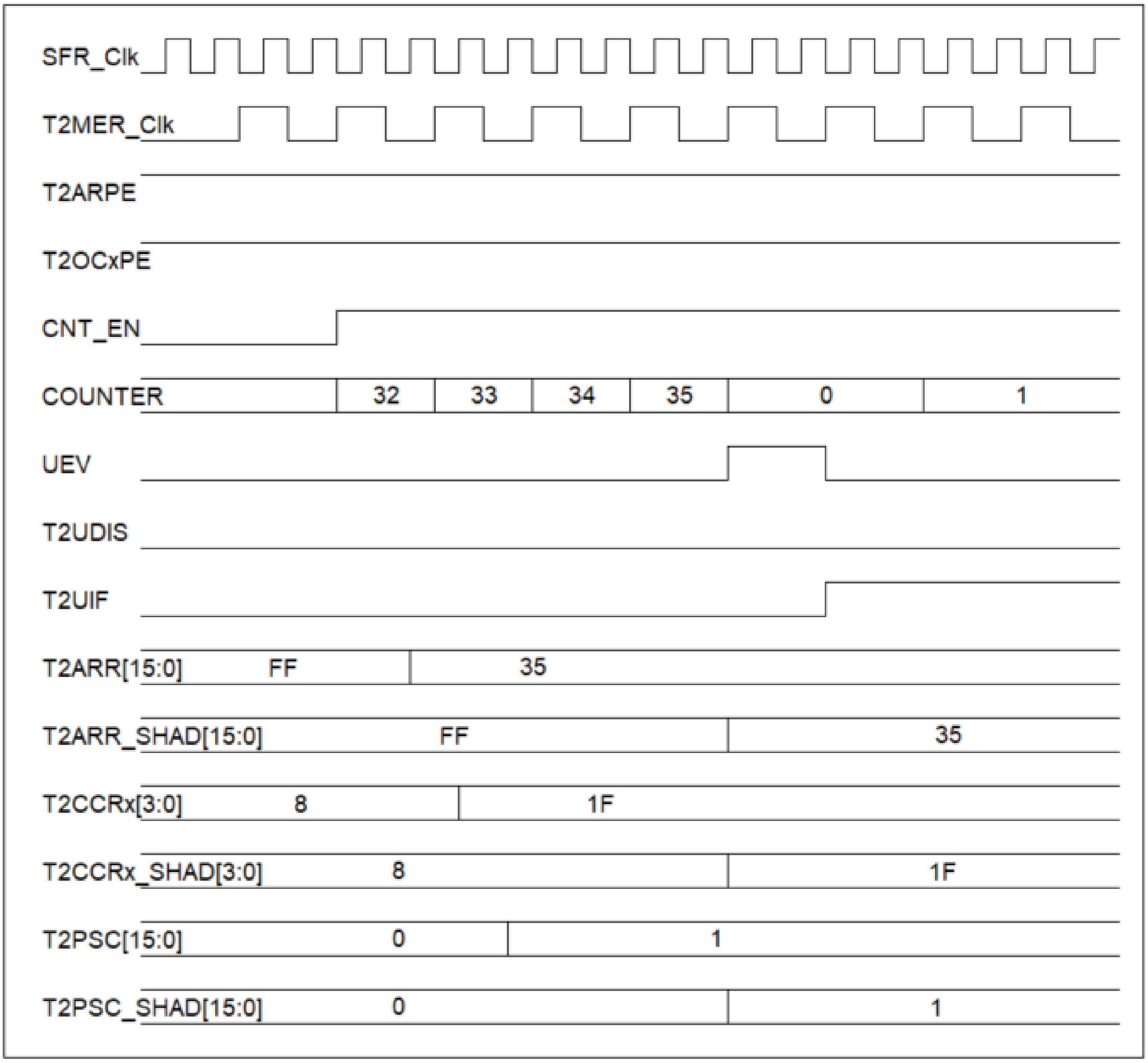


Figure 7-33 Update timing diagram of the pre-load registers under the update event

Additionally, depending on the configuration, the update event can trigger the following:

1. Related to prescaler, output comparison, and auto-reload registers:

   (1) When the counter is enabled (T2CEN = 1) and its corresponding preload is enabled (T2OCxPE / T2ARPE =1), its shadow register will be updated to the preload value when an update event occurs, as shown in **Figure 7-33**;

   (2) When the counter is disabled (T2CEN = 0), or its corresponding preload is disabled (T2OCxPE / T 2ARPE = 0), its shadow register will be updated directly with the preload value;

2. In one-pulse mode, when an update event occurs, the counter will be automatically turned off (T2CEN = 0), and stop counting;

7.3.3. Capture / Compare Channel

The CH1~3 PORTS of TIM2 can be configured as input capture or output compare function (see T2CCxS of the multiplexing register TIM2CCMRx) .

The T2CCRx registers consist of a preload register and a shadow register. Read and write process only operate on preload registers.

- In input capture mode:

  T2CCRx[15:0] is a read-only register. When a capture event occurs, the captured counter value is written to the shadow register and then copied to the T2CCRx preload register.

  When reading the T2CCRx[15:0] register, the MSB must be read first, followed by the LSB. When the MSB are read, the preload register is frozen, and then the correct LSB can be read. After reading the LSB, the preload register can be updated to the latest captured value.

- In output compare mode:

  T2CCRx[15:0] is a readable and writable register. During a write operation the T2CCRx preload register value is copied into the shadow register (see **Section 7.3.2**), then the content of shadow register is compared with the counter. The value read during a read operation comes from the preload register.

7.3.3.1. Input Capture Mode

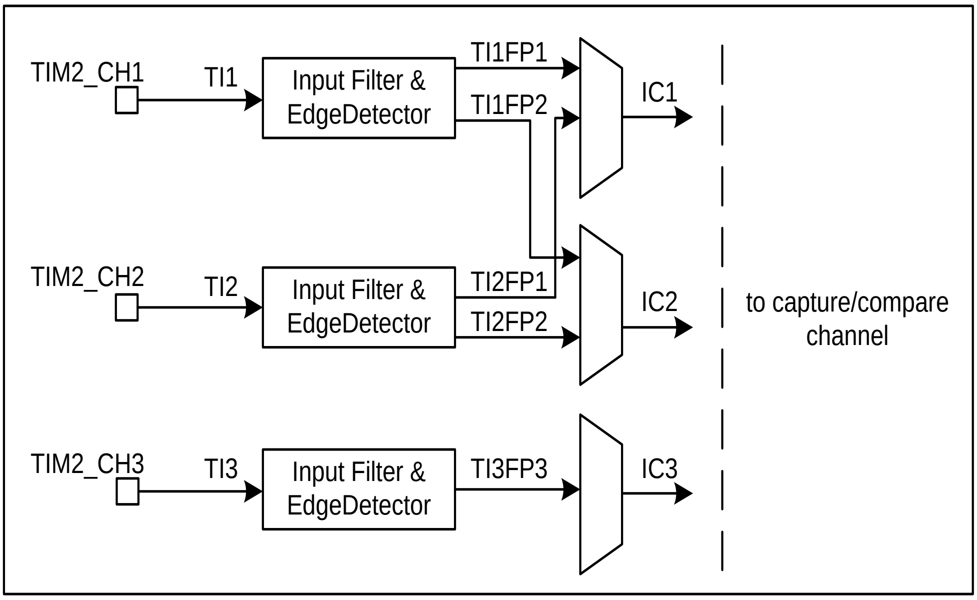

TIM2_CH1
TI1
Input Filter & EdgeDetector
TI1FP1
TI1FP2
IC1
TIM2_CH2
TI2
Input Filter & EdgeDetector
TI2FP1
TI2FP2
IC2
to capture/compare channel
TIM2_CH3
TI3
Input Filter & EdgeDetector
TI3FP3
IC3

Figure 7-34 Input Capture Channel Block Diagram

In the input capture mode, when an input capture event occurs in channel x, the current count value will be captured into the T2CCRx [15:0] register, and the input capture flag T2CCxIF is set. If the input capture event occurs again when T2CCxIF remains 1, the re-capture flag T2CCxOF will be set. Whether to trigger an interrupt and/or wake up from SLEEP depends on the corresponding enable controls (GIE, PEIE and T2CCxIE). In addition, the capture software interrupt T2CCxG can be enabled to trigger an interrupt.

The input capture sources of each channel of TIM2 (see T2CCxS) are as follows:

T2CCxS	Channel 1	Channel 2	Channel 4
01	TI1FP1	TI2FP2	TI3FP3
10	TI2FP1	TI1FP2	–

Table 7-33 Input Capture Source for each Channel

Signal name	Detailed description
TIM2_CH1/2/3	I/O inputs corresponding to Channel 1/2/3/4
IC1/2/3	Capture source via selected channel
TI1FP1	Input capture signal corresponding to I/O of channel 1, as one of the capture sources of channel 1
TI1FP2	Input capture signal corresponding to I/O of channel 1, as one of the capture sources of channel 2
TI2FP2	Input capture signal corresponding to I/O of channel 2, as one of the capture sources of channel 2
TI2FP1	Input capture signal corresponding to I/O of channel 2, as one of the capture sources of channel 1
TI3FP3	Input capture signal corresponding to I/O of channel 3, as one of the capture sources of channel 3

Table 7-34 Input Capture Signal Description

Please refer to TIM1 (**Section 7.2.4.1**) for configuration steps of input capture channels.

7.3.3.2. Output Compare Mode

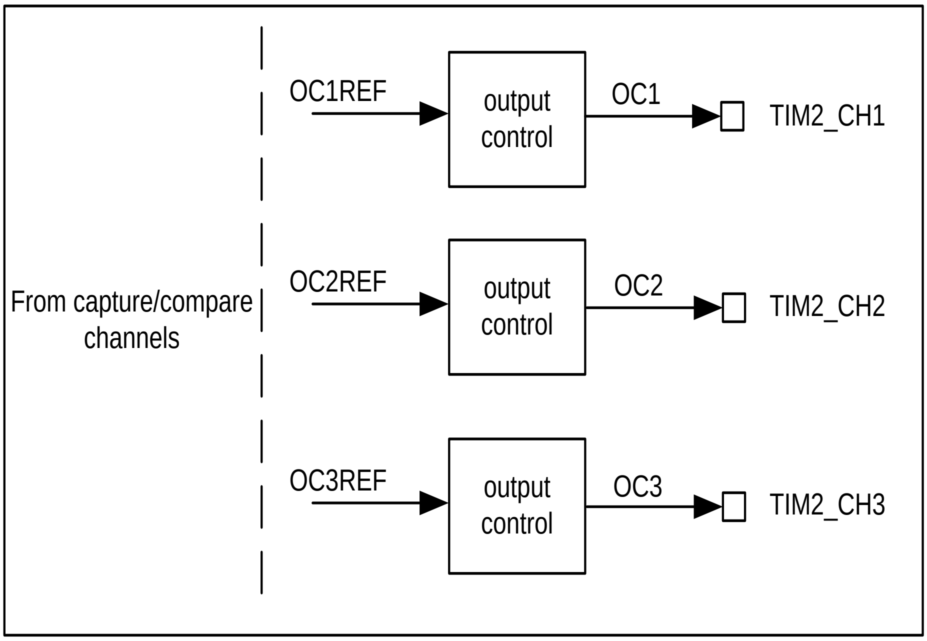


Figure 7-35 Output Compare Channel Block Diagram

The output compare module generates the output reference signal OCxREF (active high) by comparing the count value (T2CNT) with the comparison value (shadow register CCRx_SHAD), and then outputs the waveform to the port after polarity selection and output control (T2CCxE).

The reference signal OCxREF can be configured into 8 output modes via T2OCxM[2:0] (see **Table 7-25**):

1. Frozen mode (T2OCxM = 000) : OCxREF value remains unchanged.
2. Active level on match (T2OCxM = 001) : When T2CNT = CCRx_SHAD, OCxREF = 1.
3. Inactive level on match (T2OCxM = 010) : OCxREF = 0 when T2CNT = CCRx_SHAD.
4. Toggle on match (T2OCxM = 011) : When T2CNT = CCRx_SHAD, the OCxREF value is flipped.
5. Forced inactive (T2OCxM = 100): OCxREF is always 0.
6. Forced active (T2OCxM = 101): OCxREF is always 1.
7. PWM1 mode (T2OCxM = 110):

a) When T2CNT < CCRx_SHAD, OCxREF = 1; OCxREF = 0 when T 2CNT > CCRx_SHAD.

8. PWM2 mode (T2OCxM = 111):

a) When T2CNT < CCRx_SHAD, OCxREF = 0 ; when T 2CNT > CCRx_SHAD, OCxREF = 1.

When T2CNT matches CCRx_SHAD, the output compare flag T1CCx2F is set. Whether to trigger an interrupt and/or wake up from SLEEP depends on the corresponding enable controls (GIE, PEIE , T2CCxIE). In addition, the output compare software interrupt T2CCxG, can be enabled to trigger an interrupt.

Please refer to TIM1 (**section 7.2.4.2**) for configuration steps for output compare channels.

PWM Mode – PWM1/PWM2 cycle is determined by T2ARR and duty cycle is determined by T2CCRx.

Equation 7-3 $PWM1/2\ Cycle = (T2ARR+1) * T_{CK\_CNT}$

Equation 7-4 $PWM1/2\ Duty\ Cycle = T2CCRx \div (T2ARR+1)$

TIM1_CH1/2/3 channels can independently enable the output PWM signal , and the output signal polarity is optional (see T2CCxP) .

One-Pulse Mode – In One-Pulse Mode (T2OPM = 1), when the next update event occurs, the hardware automatically turns off the counter enable (T2CEN = 0) and the counter stops counting.

To generate a correct pulse, the counter initial value (T2CNT) must be different from the compare value (T2CCRx). That is, before starting counting, the following configurations must be met:

- Up-count mode: T2CNT < T2CCRx ≤ T2ARR

7.4. Basic TIMER4

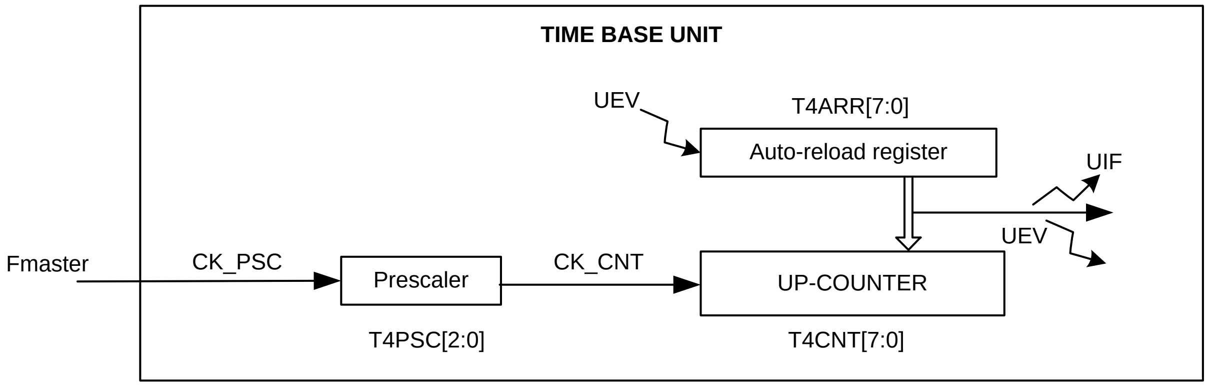


Figure 7-36 TIM4 Block Diagram

TIM4 is an 8-bit up-counter: the counter starts counting up from 0. An overflow event occurs when T4CNT = T4ARR, and then the counter starts counting from 0 again. The auto-reload register T4ARR consists of a preload register and a shadow register.

The optional 4 clock sources for the TIM4 prescaled clock (CK_PSC) (see T4CKSRC) are as follows:

- Sysclk
- HIRC
- External clock LP/XT (Only valid when FOSC is configured in LP or XT mode accordingly)

The 3-bit prescaler divides the prescaler clock (CK_PSC) by 1 ~ 128 to generate the counter clock (CK_CNT).

Frequency division Equation : $f_{CK\_CNT} = f_{CK\_PSC} / 2^{(PSCR[2:0])}$; (PSCR is the value of prescaler shadow register)

Note:

1. It is recommended that the counter value T4CNT[7:0] be read or written when the counter is stopped (T4CEN = 0) to avoid errors.
2. The cycle and other registers need to be configured first, and the prescaler register must be configured before the counter is enabled (T4CEN = 1).

When T4UDIS = 0, the update event is allowed to be generated and the update event source (see "T1URS") is as follows:

- Counter overflow
- Set T4UG by software or counter overflow

When an update event occurs, the update event flag T4UIF is set, and whether to trigger an Interrupt and /or wake up from SLEEP depends on the corresponding enable controls(GIE, PEIE and T4UIE).

Additionally, depending on the configuration, the update event can trigger the following conditions:

1. Related to the prescaler registers:

(1) When the counter is enabled (T4CEN = 1) and its corresponding preload is enabled (T4ARPE = 1) , its

shadow register will be updated to the preload value when an update event occurs.

(2) When the counter is disabled (T4CEN = 0) , or its corresponding preload is disabled (T4ARPE = 0), its shadow register will be updated directly with the preload value.

2. One-Pulse mode, when an update event occurs, the counter will be automatically turned off (T4CEN = 0), and stop counting.

7.4.1. Summary of Timer4 Related Registers

Name	Status		Register	Addr.	Reset
TIM4EN	TIM4 Clock	1 = Enable 0 = Disable	PCKEN[3]	0x9A	RW−0
SYSON	In SLEEP mode, the system clock controls	1 = Enable 0 = Disable	CKOCON[7]	0x95	RW−0
T4ARPE	Auto-preload of Counting Cycles 1 = Enable (The T4ARR preload value is loaded when the update event occurs) 0 = Disable (T4ARR is loaded immediately)		TIM4CR1[7]	0x111	RW−0
T4CKS	Timer4 Clock Source 00 = Sysclk 10 = LP (*) 01 = HIRC 11 = XT (*) (*) FOSC needs to be configured in LP/XT mode accordingly, otherwise the oscillator will not run.		TIM4CR1[5:4]		RW−00
T4OPM	One-Pulse Mode 1 = Enable (when the next update event comes , T4CEN auto-clear and the counter stops) 0 = Disable (counter does not stop when update event occurs)		TIM4CR1[3]		RW−0
T4URS	When T4UDIS = 0 , Update Event Interrupt Source 1 = Counter overflows 0 = Set T4UG by software or the counter overflows		TIM4CR1[2]		RW−0
T4UDIS	Update Event Generation Control	1 = Disable 0 = Enable	TIM4CR1[1]		RW−0
T4CEN	TIM4 Counter	1 = Enable 0 = Disable	TIM4CR1[0]		RW−0

Name	Status	Register	Addr.	Reset
T4PSC	Timer4 Prescaler 000 = 1 100 = 16 001 = 2 101 = 32 010 = 4 110 = 64 011 = 8 111 =128	TIM4PSCR[2:0]	0x116	RW-000
T4CNT	Timer4 Count Value	TIM4CNTR[7:0]	0x115	RW−0000 0000
T4ARR	Auto-reload register for counting cycles (preload value) Note: When this value is 0, the counter does not work	TIM4ARR[7:0]	0x117	RW−1111 1111

Table 7-33 Timer4 Relatd User Control Registers

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt 1 = Enable (PEIE, T4UIE, T4UG apply) 0 = Global Shutdown (Wake-Up is not affected)		INTCON[7]	Bank first address+0x0B	RW−0
PEIE	Peripheral Interrupt Enable 1 = Enable (T4UIE, T4UG applicable) 0 = Disable (no Wake-Up)		INTCON[6]		RW−0
T4UIE	Allow Update Interrupt	1 = Enable 0 = Disable	TIM4IER[0]	0x112	RW−0
T4UG [13]	Allow Update Software Interrupt		TIM4EGR[0]	0x114	WO−0
T4UIF [14]	Update Event Interrupt Flag	1 = Update event occurs 0 = No update event	TIM4SR[0]	0x113	R_W1C−0

Table 7-34 Timer4 Interrupt Enable and Status Bits

[13] Set to 1 by software, auto clear by hardware.

[14] Write '1' to clear, and writing '0' has no effect on the bit value. It is recommended to use only STR and MOVWI instructions for Write operation, not BSR or IOR instructions.

Name	Addr	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset
PCKEN	0x9A	TKEN	I2CEN	UARTEN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
TIM4CR1	0x111	T4ARPE	—	T4CKS[1:0]		T4OPM	T4URS	T4UDIS	T4CEN	0-00 0000
TIM4IER	0x112	—	—	—	—	—	—	—	T4UIE	---- ---0
TIM4SR	0x113	—	—	—	—	—	—	—	T4UIF	---- ---0
TIM4EGR	0x114	—	—	—	—	—	—	—	T4UG	---- ---0
TIM4CNTR	0x115	T4CNT[7:0]								0000 0000
TIM4PSCR	0x116	—	—	—	—	—	T4PSC[2:0]			---- -000
TIM4ARR	0x117	T4ARR[7:0]								1111 1111

Table 7-35 Summary of Timer4 Related Registers (Reserved bits must remain as the reset value and cannot be changed)

8. SLEEP (POWER-DOWN)

During SLEEP Instruction Clock is inactive and instructions execution is halted. Most modules are powered down to conserve power. As listed in **Table 8-1**, FT62F08x can selectively turn on individual modules in SLEEP according to actual needs so that functions such as LVR, LVD, WDT, Timers, PWM , ADC, SPI, I2C and USART, can remain active during SLEEP without instruction interventions. Some modules can configure automatic power down upon SLEEP to save the need to turn them off by instructions.

	Condition in SLEEP	
	RUN	Auto-Shutdown
Instruction Clock	(always power down)	Yes
LVR (Configure the LVREN)	Enabled or Instruction controlled (SLVREN=1)	Enabled except in SLEEP
LVD	LVDEN = 1	No
WDT	WDTE or SWDTEN	No
TIMER1	SYSON = 1 & TIM1EN = 1 & T1CEN = 1	SYSON = 0
TIMER2	SYSON = 1 & TIM2EN = 1 & T2CEN = 1	SYSON = 0
TIMER4	SYSON = 1 & TIM4EN = 1 & T4CEN = 1	SYSON = 0
Clock Output	SYSON = 1 & CCOEN = 1	SYSON = 0
ADC	SYSON = 1 & ADCEN = 1 & ADON = 1	Yes : SYSON = 0 & ADCS ≠ x11 No: ADCS = x11
SPI	SYSON = 1 & SPICKEN = 1 & SPIEN = 1	SYSON = 0
I2C	SYSON = 1 & I2CEN = 1 & ENABLE = 1	SYSON = 0
USART	SYSON = 1 & UARTEN = 1 & TXEN / RXEN = 1	SYSON = 0
PWM	(follows Timer1 or Timer2)	
HIRC / LIRC / EC / LP / XT	(follow the states of the used peripherals)	
I/O	(maintain their states before SLEEP unless PWM, clock output , or other peripherals SLEEP enabled)	

Table 8-1 All modules except Instruction Clock can remain active in SLEEP if so desired

8.1. Enter SLEEP

The CPU enters SLEEP by executing the SLEEP instruction. When enter SLEEP:

1. If WDT is enabled, it will clear its Prescalar (if assigned) and counter, and start counting.
2. Time Out Flag (/TF) = 1
3. Power Down Flag (/PF) = 0
4. Clock sources
   - Instruction Clock shuts down automatically when SYSON = 0.
   - When SYSON = 1 , the instruction clock keeps active, so does the corresponding peripherals and their selected clock sources (HIRC , LIRC, EC, LP or XT) that enable the module system clock (see PCKEN).
   - When SYSON = 1 , the clock source selected for the clock output (see C COSEL) will keep active and the clock output will continue.

5. I/O PORTS

- When SYSON = 1 & TIMxEN = 1 & TxCEN = 1, PWM output continue if Timers are active in SLEEP. When SYSON = 0 , Timers auto-shutdown and the PWM output will maintain its state before entering SLEEP.
- When SYSON = 1 , if the system clock and module function of ADC, SPI, I2C or USART are enabled at the same time, the module function can keep active. When SYSON = 0 , ADC (ADCS ≠ x 11), SPI , I2C or USART will automatically shut down, except for ADC clock source selection LIRC (ADCS = x 11) .
- For other Digital Outputs, they will maintain the state before SLEEP (High-z state, “0” or “1”)

For more information about how peripherals work in SLEEP please refer to the corresponding Sections.

Note :

1. If an interrupt occurs before the SLEEP instruction is executed (the interrupt flag is set and its interrupt is enabled, but the Global Interrupt GIE is disabled) , the SLEEP instruction will be executed as a NOP and will not enter sleep mode.
2. Due to the synchronization delay, after clearing the interrupt flag to 0, at least two instructions need to be waited before the SLEEP instruction can be executed, otherwise it will not enter the sleep mode.

8.2. Wake Up from SLEEP

There are 2 general principles to wake up form SLEEP:

- Time based, in which the CPU wakes up after a certain amount of time. LIRC is the clock choice for keeping time as it has lower power consumption than HIRC.
- Events based that triggers POR, System-Reset, Wake-up without Interrupt, and Interrupts, such as LVD, ADC , EPROM write completion ,and external pin interrupt.

The situations that wake up from SLEEP as follows:

1. Watchdog Timer Wake-up if enabled (see **Section 7.1** Watchdog Timer).
2. Full-Reset and System-Reset
   - POR Full-Reset (cannot be disabled)
   - External System-Reset by the /MCLR (if enable)
   - LVR Reset (if enable)
3. Enabled Interrupts (Disabling the “Global Interrupt Enable” will not stop Wake-up). Please see **Section 9** Interrupts.

Notes:

1. Waking up form SLEEP will also clear WDT.
2. <u>SLEEP must be followed by NOP</u>

   When wakes up from SLEEP in a non interrupt mode (that is, "interrupt service program" is not executed), such as WDT wake up, or attempted Interrupts with Global Interrupt Enable (GIE) disabled, the next instruction will be executed.

When using the Interrupt method to wake-up from SLEEP, the next instruction will be executed before calling the Interrupt Service Routine. To avoid duplicate execution NOP must follow SLEEP.

```
SLEEP
NOP                 // Wake-Up by interrupt
```

9. INTERRUPTS

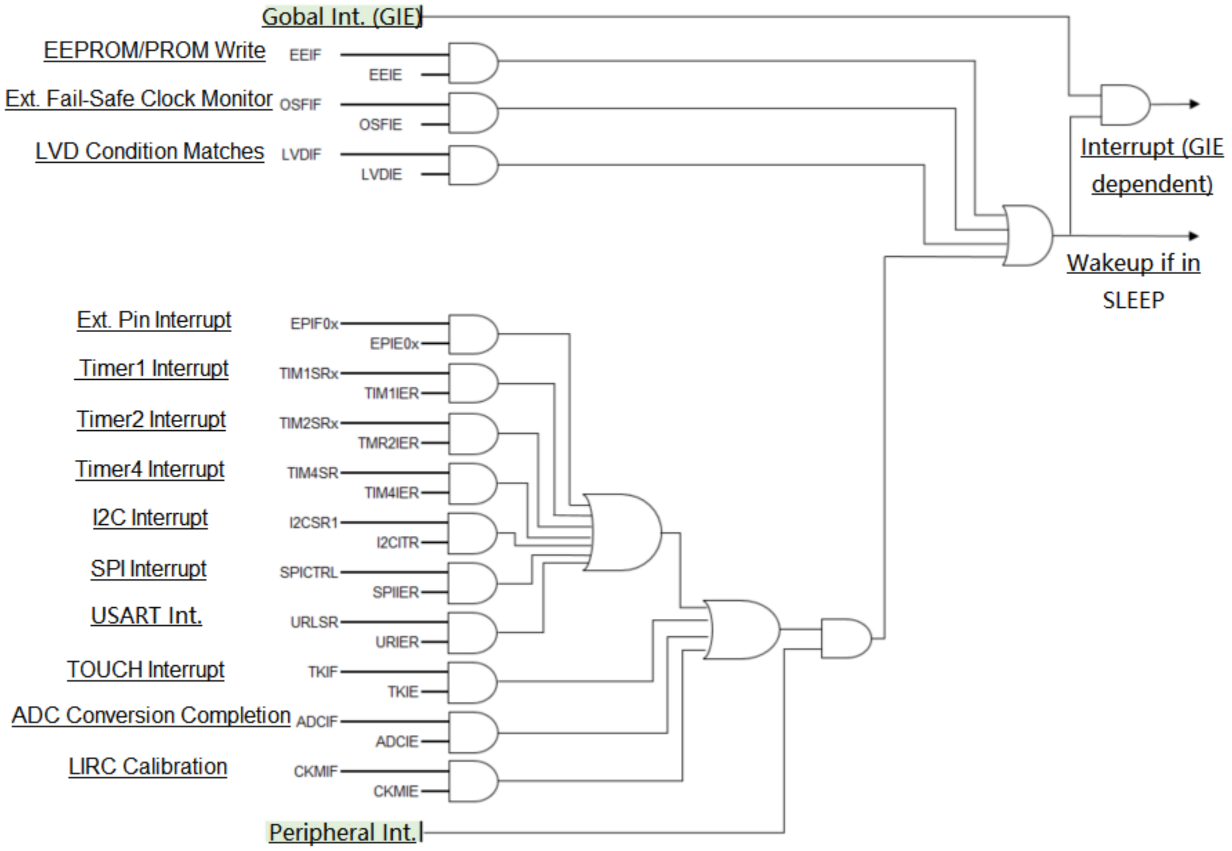


Figure 9-1 Interrupt Block Diagram

The CPU supports 13 interrupt sources, divided into 2 groups:

1) Non-peripheral interrupts

- DATA EEPROM/PROM Write Completion
- Fail-Safe Clock Monitor
- LVD Condition Matches

2) Peripheral interrupt

- External pin interrupt
- Timer1 interrupt
- Timer2 interrupt
- Timer4 interrupt
- I2C interrupt
- SPI interrupt
- USART interrupt
- TOUCH interrupt
- LIRC and HIRC Cross Calibration Completion
- ADC Conversion Completion

WDT overflows, unlike other Timers, will not result in an Interrupt. For other interrupts besides external I/O interrupts please see the corresponding Sections.

When an interrupt occurs, the PC jumps to and executes the "Interrupt Service Routine (ISR)". There are multiple levels of Interrupt Disable/Enable.

- Each interrupt source has a local interrupt enable: EEIE, LVDIE, OSFIE, TKIE, CKMIE, ADCIE, TIM1IER, TIM2IER, TIM4IER, I2CITR, SPIIER, URIER.
- At the same time, there are up to 8 external pin interrupt inputs, sharing a PORT interrupt enable: EPIE0x (External PORTx Interrupt Enable).
- The Peripheral interrupts has a total interrupt enable: PEIE (Peripheral Interrupt Enable).
- All controls above, if disabled, will not execute a Wake-Up from SLEEP.
- All interrupts are controlled by a global enable: GIE (Global Interrupt Enable). This enable differs from the others by allowing a Wake-Up from SLEEP even when disabled.
- Disabling the interrupts enable bit does not affect the setting of the interrupt flags.

The following sequences occur upon an Interrupt:

- Auto set "GIE = 0", disabling further interrupts.
- The return address is pushed onto the stack and the PC (program counter) is loaded with 0x0004.
- Jump to the "Interrupt Service Routine" in 3 – 5 instruction cycles after the interrupt.
- "Return from Interrupt (RETI)" instruction exits ISR. Prior to RETI must clear the interrupt flag currently being processed.
- At the completion of the ISR, the PC returns to the address before the interrupt, which in SLEEP, is the address immediate after SLEEP.
- Auto set GIE = 1 upon executing RETI, enabling future interrupts.

Note: Only the returned PC address is saved on the stack during an interrupt, and other important registers [e.g., W, STATUS (except /TO and /PD) , BSREG, FSR, PCLATH] are automatically saved in the corresponding shadow registers (R/W, see bank 31) . These register values are automatically restored from the shadow registers when the interrupt service routine is exited. Users desiring to have other registers saved must use instructions to write them into temporary registers explicitly. Use the last 16 bytes of SRAM for temporary storages as they are common to all banks and do not require bank swithing.

9.1. Summary of Interrupt Related Registers

Name	Addr.	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset(RW)
INTCON	0x0B	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
PIE1	0x91	−	−	−	−	−	TKIE	CKMIE	ADCIE	−−−− −000
PIR1	0x11	−	−	−	−	−	TKIF	CKMIF	ADCIF	−−−− −000
TIM1IER	0x215	T1BIE	T1TIE	−	T1CC4IE	T1CC3IE	T1CC2IE	T1CC1IE	T1UIE	00−0 0000
TIM1SR1	0x216	T1BIF	T1TIF	−	T1CC4IF	T1CC3IF	T1CC2IF	T1CC1IF	T1UIF	00−0 0000
TIM1SR2	0x217	−	−	−	T1CC4OF	T1CC3OF	T1CC2OF	T1CC1OF	−	−−−0 000−
TIM1EGR	0x218	−	−	−	T1CC4G	T1CC3G	T1CC2G	T1CC1G	−	−−−0 000−
TIM2IER	0x30D	−	−	−	−	T2CC3IE	T2CC2IE	T2CC1IE	T2UIE	−−−− 0000
TIM2SR1	0x30E	−	−	−	−	T2CC3IF	T2CC2IF	T2CC1IF	T2UIF	−−−− 0000
TIM2SR2	0x30F	−	−	−	−	T2CC3OF	T2CC2OF	T2CC1OF	−	−−−− 000−
TIM2EGR	0x310	−	−	−	−	T2CC3G	T2CC2G	T2CC1G	T2UG	−−−− 0000
TIM4IER	0x112	−	−	−	−	−	−	−	T4UIE	−−−− −−−0
TIM4SR	0x113	−	−	−	−	−	−	−	T4UIF	−−−− −−−0
TIM4EGR	0x114	−	−	−	−	−	−	−	T4UG	−−−− −−−0
SPIIER	0x1C	−	−	−	−	WAKUP	RXERR	RXNE	TXE	−−−− 0000
SPISTAT	0x1E	−	SMODF	SRXOVEN	SBUSY	SRXBMT	STXBMT	WKF	CRCERR	-000 1100
SPICTRL	0x16	SPIF	WCOL	MODF	RXOVRN	NSSM[1:0]		TXBMT	SPIEN	0000 0110
SPICFG	0x17	BUSY	MSTEN	CPHA	CPOL	SLAS	NSSVAL	SRMT	RXBMT	0000 0111
I2CITR	0x416	−	−	−	−	−	ITBUFEN	ITEVEN	ITERREN	−−−− −000
I2CSR1	0x417	IICTXE	IICRXNE	−	STOPF	ADD10F	−	ADDF	SBF	00-0 0-00
I2CSR2	0x418	−	−	−	TXARBT	OVR	AF	ARLO	BERR	---0 0000
I2CSR3	0x419	−	−	GCALL	−	−	RDREQ	ACTIVE	RXHOLD	--0- -000
URIER	0x48E	−	−	TCEN	−	IDELE	RXSE	URTE	URRXNE	−−0− 0000
URLSR	0x492	ADDRF	IDLEF	TXEF	BKF	FEF	PEF	OVERF	RXNEF	0010 0000
URTC	0x49C	−	−	−	−	−	−	−	TCF	---- ---1
EPIE0	0x94	External Interrupt Control Register								0000 0000
EPIF0	0x14	External Interrupt Flag Register								0000 0000
TRISA	0x8C	PORTA Data Direction Register								1111 1111
TRISB	0x8D	PORTB Data Direction Register								1111 1111
TRISC	0x8E	PORTC Data Direction Register								1111 1111
TRISD	0x8F	−	−	PORTD Data Direction Register						−−11 1111
EPS0	0x118	External Interrupt EINT3 ~ 0 Pin Select Register								0000 0000
EPS1	0x119	External interrupt EINT7 ~ 4 Pin Select Register								0000 0000

Table 9-1 Interrupt Related Register Addresses and Default

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt 1 = Enable (PEIE, Independent Enable for each Interrupt apply) 0 = Global Shutdown (Wake-Up not affected)		INTCON[7]	Bank first address+0x0B	RW−0
PEIE	Peripheral Interrupt Enable 1 = Enable (Independent Enable for each Interrupt apply) 0 = Disable (no Wake-Up)		INTCON[6]		RW−0
EEIE	EEPROM/PROM Write Completion interrupt	1 = Enable 0 = Disable (no Wake-Up)	INTCON[5]		RW−0
LVDIE	LVD interrupt		INTCON[4]		RW−0
OSFIE	External Oscillator Failed Interrupt		INTCON[3]		RW−0
EEIF [1]	EEPROM/PROM Write Completion interrupt Flag	1 = Yes (latched) 0 = No	INTCON[2]		R_W1C−0
LVDIF [1]	LVD interrupt Flag		INTCON[1]		R_W1C−0
OSFIF [1]	External Oscillator Failed Interrupt Flag		INTCON[0]		R_W1C−0

Table 9-2 INTCON register

Name	Status		Register	Addr.	Reset
TKIE	TOUCH Interrupt	1 = Enable 0 = Disable (no Wake-Up)	PIE1[2]	0x91	RW−0
CKMIE	LIRC and HIRC Cross Calibration Completion interrupt		PIE1[1]		RW−0
ADCIE	ADC Conversion Completion Interrupt		PIE1[0]		RW−0
TKIF [1]	TOUCH Interrupt Flag	1 = Yes (latched) 0 = No	PIR1[2]	0x11	R_W1C−0
CKMIF [1]	LIRC and HIRC Cross Calibration Completion Flag		PIR1[1]		R_W1C−0
ADCIF [1]	ADC Conversion Completion Flag		PIR1[0]		R_W1C−0

Table 9-3 PIE1 and PIR1 registers

[1] Write '1' to clear, and writing '0' has no effect on the bit value. It is recommended to only execute the STR and MOVWI instructions for write operations, rather than the BSR or IOR instructions .

Name	Status		Register	Addr.	Reset
T1BIE	Timer1 Break Interrupt	1 = Enable 0 = Disable (no Wake-Up)	TIM1IER[7]	0x215	RW−0
T1TIE	Timer1 Trigger Interrupt		TIM1IER[6]		RW−0
T1CC4IE	Timer1 Capture/Compare Channel4 Interrupt		TIM1IER[4]		RW−0
T1CC3IE	Timer1 Capture/Compare Channel3 Interrupt		TIM1IER[3]		RW−0
T1CC2IE	Timer1 Capture/Compare Channel2 Interrupt		TIM1IER[2]		RW−0
T1CC1IE	Timer1 Capture/Compare Channel1 Interrupt		TIM1IER[1]		RW−0
T1UIE	Timer1 Update Event Interrupt		TIM1IER[0]		RW−0
T1BG	Timer1 Break Software Interrupt		TIM1EGR[7]	0x218	WO−0
T1CC4G [2]	Timer1 Capture/Compare Channel4 Software Interrupt		TIM1EGR[4]		WO−0
T1CC3G [2]	Timer1 Capture/Compare Channel3 Software Interrupt		TIM1EGR[3]		WO−0
T1CC2G [2]	Timer1 Capture/Compare Channel2 Software Interrupt		TIM1EGR[2]		WO−0
T1CC1G [2]	Timer1 Capture/Compare Channel1 Software Interrupt		TIM1EGR[1]		WO−0
T2CC3IE	Timer2 Capture/Compare Channel3 Interrupt		TIM2IER[3]	0x30D	RW−0
T2CC2IE	Timer2 Capture/Compare Channel2 Interrupt		TIM2IER[2]		RW−0
T2CC1IE	Timer2 Capture/Compare Channel1 Interrupt		TIM2IER[1]		RW−0
T2UIE	Timer2 Update Event Interrupt		TIM2IER[0]		RW−0
T2CC3G [2]	Timer2 Capture/Compare Channel3 Software Interrupt		TIM2EGR[3]	0x310	WO−0
T2CC2G [2]	Timer2 Capture/Compare Channel2 Software Interrupt		TIM2EGR[2]		WO−0
T2CC1G [2]	Timer2 Capture/Compare Channel1 Software Interrupt		TIM2EGR[1]		WO−0
T2UG [2]	Timer2 Update Event Software Interrupt		TIM2EGR[0]		WO−0
T4UIE	Timer4 Update Event Interrupt		TIM4IER[0]	0x112	RW−0
T4UG [2]	Timer4 Update Event Software Interrupt		TIM4EGR[0]	0x114	WO−0

Table 9-4 TIMx INTCON Register

[2] Set to 1 by software, auto clear by hardware.

Name	Status		Register	Addr.	Reset
T1BIF [3]	Timer1 Break Interrupt Flag	1 = Yes (latched) 0 = No	TIM1SR1[7]	0x216	R_W1C-0
T1TIF [3]	Timer1 Trigger event Interrupt Flag		TIM1SR1[6]		R_W1C-0
T1CC4IF [3]	Timer1 Capture/Compare Channel4 Interrupt Flag		TIM1SR1[4]		R_W1C-0
T1CC3IF [3]	Timer1 Capture/Compare Channel3 Interrupt Flag		TIM1SR1[3]		R_W1C-0
T1CC2IF [3]	Timer1 Capture/Compare Channel2 Interrupt Flag		TIM1SR1[2]		R_W1C-0
T1CC1IF [3]	Timer1 Capture/Compare Channel1 Interrupt Flag		TIM1SR1[1]		R_W1C-0
T1UIF [3]	Timer1 Update Event Interrupt Flag		TIM1SR1[0]		R_W1C-0
T1CC4OF [3]	Timer1 Capture/Compare Channel4 Repeated Capture Interrupt Flag		TIM1SR2[4]	0x217	R_W1C-0
T1CC3OF [3]	Timer1 Capture/Compare Channel3 Repeated Capture Interrupt Flag		TIM1SR2[3]		R_W1C-0
T1CC2OF [3]	Timer1 Capture/Compare Channel2 Repeated Capture Interrupt Flag		TIM1SR2[2]		R_W1C-0
T1CC1OF [3]	Timer1 Capture/Compare Channel1 Repeated Capture Interrupt Flag		TIM1SR2[1]		R_W1C-0
T2CC3IF [3]	Timer2 Capture/Compare Channel3 Interrupt Flag		TIM2SR1[3]	0x30E	R_W1C-0
T2CC2IF [3]	Timer2 Capture/Compare Channel2 Interrupt Flag		TIM2SR1[2]		R_W1C-0
T2CC1IF [3]	Timer2 Capture/Compare Channel1 Interrupt Flag		TIM2SR1[1]		R_W1C-0
T2UIF [3]	Timer2 Update Event Interrupt Flag		TIM2SR1[0]		R_W1C-0
T2CC3OF [3]	Timer2 Capture/Compare Channel3 Repeated Capture Interrupt Flag		TIM2SR2[3]	0x30F	R_W1C-0
T2CC2OF [3]	Timer2 Capture/Compare Channel2 Repeated Capture Interrupt Flag		TIM2SR2[2]		R_W1C-0
T2CC1OF [3]	Timer2 Capture/Compare Channel1 Repeated Capture Interrupt Flag		TIM2SR2[1]		R_W1C-0
T4UIF [3]	Timer4 Update Event Interrupt Flag		TIM4SR[0]	0x113	R_W1C-0

Table 9-5 TIMx Interrupt Flag Register

[3] Write '1' to clear, and writing '0' has no effect on the bit value. It is recommended to execute only STR and MOVWI instructions for write operations, rather than BSR or IOR instructions.

Name	Status		Register	Addr.	Reset
TXE	Transmit BUF empty interrupt	1 = Enable 0 = Disable (no Wake-Up)	SPIIER[0]	0x1C	RW−0
TXBMT	Transmit BUF status bit	1 = Empty 0 = Not empty	SPICTRL[1]	0x16	RO−1
STXBMT			SPISTAT[2]	0x1E	RO−1
RXNE	Receive BUF not empty interrupt	1 = Enable 0 = Disable (no Wake-Up)	SPIIER[1]	0x1C	RW−0
RXBMT	Receive BUF Status	1 = Empty 0 = Not empty	SPICFG[0]	0x17	RO−1
SRXBMT			SPISTAT[3]	0x1E	RO−1
RXERR	Receive error interrupt (work mode error, receive overflow, CRC check error)	1 = Enable 0 = Disable (no Wake-Up)	SPIIER[2]	0x1C	RW−0
MODF [4]	Working mode error flag	1 = Error (latched) (In the master mode, the NSS Pin is enabled and the input is low, resulting in an error) 0 = Normal	SPICTRL[5]	0x16	RW0−0
SMODF			SPISTAT[6]	0x1E	RO−0
RXOVRN [4]	Receive overflow Flag	1 = Overflow (latch) 0 = Normal	SPICTRL[4]	0x16	RW0−0
SRXOVRN			SPISTAT[5]	0x1E	RO−0
CRCERR [4]	CRC check error flag	1 = Error (latched) 0 = Correct, or cleared	SPISTAT[0]	0x1E	RW0−0
WAKUP	Slave wake-up interrupt	1 = Enable 0 = Disable	SPIIER[3]	0x1C	RW−0
WKF [4]	Slave wake-up (data received) Flag	1 = Wake-up (latched) 0 = No wake-up, or cleared	SPISTAT[1]	0x1E	RW0−0

Table 9-6 SPI Interrupt Enable and Status Bits

[4] Write '0' to clear, and writing '1' has no effect on the bit value.

Name	Status		Register	Addr.	Reset
ITBUFEN	FIFO status interrupt	1 = Enable (When IICTXE = 1 or IICRXNE = 1, Interrupt generated) 0 = Disable (no Wake-Up)	I2CITR[2]	0x416	RW−0
IICTXE [5]	TX-FIFO status	1 = Empty 0 = Not empty	I2CSR1[7]	0x417	RO−0
IICRXNE [5]	RX-FIFO status	1 = Not empty 0 = Empty	I2CSR1[6]		RO−0
ITEVEN	Event interrupt	1 = Enable 0 = Disable (no Wake-Up) Conditions for generating Event Interrupt: SBF = 1 (master) ADD10F = 1 (master) ADDF = 1 (master/slave) STOPF = 1 (slave)	I2CITR[1]	0x416	RW−0
STOPF [6]	Slave detect Stop Flag	1 = Yes (set after ACK) 0 = No	I2CSR1[4]	0x417	RO−0
ADD10F [6]	Master send MSB Address Flag	1 = Yes (set after ACK) 0 = No	I2CSR1[3]		RO−0
ADDF [6]	Master send LSB Address/Slave Receive Address Match Flag	Master send address LSB: 1 = Completed (set after ACK) 0 = Not completed Slave receive address: 1 = Matched or General Call recognized 0 = Mismatched Note: ADDF will not be set after NACK	I2CSR1[1]		RO−0
SBF [6]	Master send Start Flag	1 = Yes 0 = No	I2CSR1[0]		RO−0
ITERREN	Error Interrupt	1 = Enable 0 = Disable (no Wake-Up) Conditions for generating error Interrupt: OVR = 1 AF = 1	I2CITR[0]	0x416	RW-0

[5] Automatically cleared by hardware when writing DR or ENABLE = 0 .

[6] Automatically cleared by hardware when I2CSR1 is read or ENABLE = 0 .

Name	Status		Register	Addr.	Reset
		ARLO = 1 BERR = 1			
TXABRT [7]	Transmission abort flag (caused by an error or abnormal cause during the Transmission)	1 = Abort occurred 0 = No Abort occurred	I2CSR2[4]	0x418	RW0−0
OVR [7]	Overrun flag	1 = Overrun 0 = No overrun Conditions for Overrun: TX-over: still write DR when the TX-FIFO is not empty; RX-over: still receive data when the RX-FIFO is not empty; RX-under: read when the RX-FIFO is empty;	I2CSR2[3]		RW0-0
AF [7]	ACK status	1 = NACK 0 = ACK	I2CSR2[2]		RW0−0
ARLO [7]	Master Arbitration fail Flag	1 = Generate arbitration fail 0 = No arbitration fail has occurred	I2CSR2[1]		RW0−0
BERR [7]	Bus error status (Start/Stop with misalignment detected)	1 = Detected (set when a Start/Stop is detected during the byte transmission phase) 0 = Not detected	I2CSR2[0]		RW0−0

Table 9-7 2C Interrupt Enable and Status Bits

Name	Status		Register	Addr.	Reset
URTE	Send BUF empty interrupt	1 = Enable 0 = Disable	URIER[1]	0x48E	RW−0
TXEF	Send BUF status	1 = Empty 0 = Not empty Note: Write DATAL(8bit) / DATAH(9bit) to clear	URLSR[5]	0x492	RO−1
URRXNE	Receive BUF not empty interrupt	1 = Enable 0 = Disable	URIER[0]	0x48E	RW−0
RXNEF	Receive BUF status	1 = Not empty 0 = Empty, or cleared Note: Read DATAL(8bit) / DATAH(9bit)	URLSR[0]	0x492	RO−0

[7] Write 0 to clear, or hardware automatically clears when ENABLE = 0 .

Name	Status		Register	Addr.	Reset
		to clear			
TCEN	Send Completion Interrupt	1 = Enable 0 = Disable	URIER[5]	0x48E	RW−0
TCF	Transmission Completion Flag	1 = Completed 0 = Not completed Note: Write 1 to clear, or write DATAL(8bit) /DATAH(9bit) to clear	URTC[0]	0x49C	R_W1C−1
IDELE	Idle frame interrupt	1 = Enable 0 = Disable	URIER[3]	0x48E	RW−0
IDLEF [8]	Idle frame detected Flag	1 = Detected 0 = Not detected	URLSR[6]	0x492	RW0−0
RXSE [9]	Receive status interrupt	1 = Enable 0 = Disable Conditions for receiving status Interrupt: BKF = 1 FEF = 1 PEF = 1 OVERF = 1	URIER[2]	0x48E	RW−0
BKF [8]	Received broken frame Flag	1 = Received 0 = Not received, or cleared	URLSR[4]	0x492	RW0−0
FEF [8]	Received frame error Flag	1 = Error 0 = Correct, or cleared	URLSR[3]	0x492	RW0−0
PEF [8]	Received Parity error Flag	1 = Error 0 = Correct, or cleared	URLSR[2]	0x492	RW0−0
OVERF [8]	Receive BUF overflow Flag	1 = Overflow 0 = Normal, or cleared	URLSR[1]	0x492	RW0−0
WAKE	Mute Mode Wake-up Selection	1 = Address match 0 = IDLE frame	URMCR[2]	0x491	RW−0
ADDRF	Mute Mode Address Matching Flag	1 = Match 0 = Mismatch	URLSR[7]	0x492	RO−0

Table 9-8 USART Interrupt Enable and Status Bits

[8] Write '0' to clear, and writing '1' has no effect on the bit value.

[9] USART received broken frame, framing error, parity error, receive overflow error status.

Name	Status		Register	Addr.	Reset
ITYPE0[1:0]	PORTx.0	External Interrupt Pin EINTx Trigger Type 00 = Low level 01 = Rising edge 10 = Falling edge 11 = Double-edge	ITYPE0[1:0]	0x11E	RW−00
ITYPE0[3:2]	PORTx.1		ITYPE0[3:2]		RW−00
ITYPE0[5:4]	PORTx.2		ITYPE0[5:4]		RW−00
ITYPE0[7:6]	PORTx.3		ITYPE0[7:6]		RW−00
ITYPE1[1:0]	PORTx.4		ITYPE1[1:0]	0x11F	RW−00
ITYPE1[3:2]	PORTx.5		ITYPE1[3:2]		RW−00
ITYPE1[5:4]	PORTy.6		ITYPE1[5:4]		RW−00
ITYPE1[7:6]	PORTy.7		ITYPE1[7:6]		RW−00

Table 9-9 External Interrupt Trigger Type Register (x = A, B, C; y = A, B)

Name	Status				Register	Addr.	Reset
EINT0	00 = PA0	01 = PB0	10 = PC0	11 = PD0	EPS0[1:0]	0x118	RW−00
EINT1	00 = PA1	01 = PB1	10 = PC1	11 = PD1	EPS0[3:2]		RW−00
EINT2	00 = PA2	01 = PB2	10 = PC2	11 = PD2	EPS0[5:4]		RW−00
EINT3	00 = PA3	01 = PB3	10 = PC3	11 = PD3	EPS0[7:6]		RW−00
EINT4	00 = PA4	01 = PB4	10 = PC4	11 = PD4	EPS1[1:0]	0x119	RW−00
EINT5	00 = PA5	01 = PB5	10 = PC5	11 = PD5	EPS1[3:2]		RW−00
EINT6	00 = PA6	01 = PB6	10 = PC6	11 = Reserved	EPS1[5:4]		RW−00
EINT7	00 = PA7	01 = PB7	10 = PC7	11 = Reserved	EPS1[7:6]		RW−00

Table 9-10 External Interrupt Pin Selection Register

Name	Status		Register	Addr.	Reset
EPIE0x	External Pin Interrupt	1 = Enable 0 = Disable	EPIE0[7:0]	0x94	RW−00000000
EPIF0x [10]	External Pin Interrupt Flag	1 = Yes (latched) 0 = No	EPIF0[7:0]	0x14	R_W1C−00000000

Table 9-11 External Interrupt Enable and Flag Registers

[10] Write '1' to clear, and writing '0' has no effect on the bit value. It is recommended to only execute the STR and MOVWI instructions for write operations, rather than the BSR or IOR instructions .

9.2. External pin interrupt

All I/O support external pin interrupts, and there are up to 8 external pin interrupt inputs (refer to EINT0~7) , and the I/O need to be set as input (TRISA/B/C[x] = 1 , and ANSELA[x] = 0) , the trigger source can be selected as rising edge, falling edge, double edge and low level (refer to ITYPE x) .

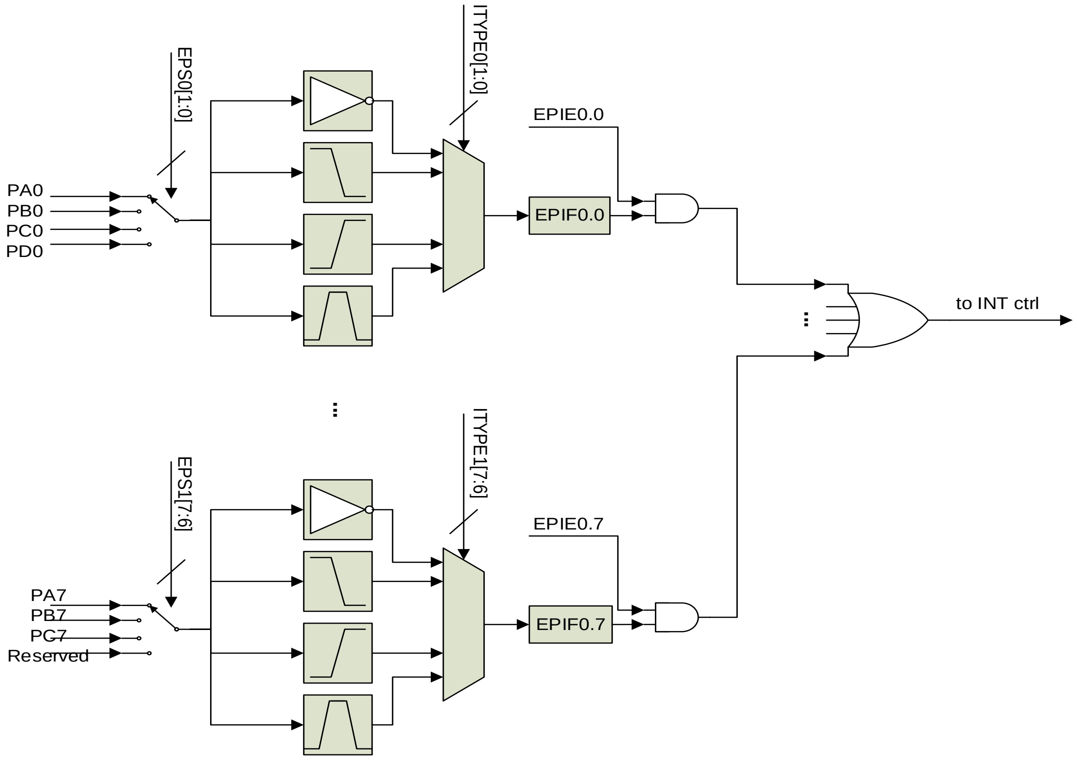


Figure 9-2 External Interrupt Block Diagram

10. Data EEPROM and Program PROM

Both the non-volatile DATA EEPROM memory array and the program memory PROM integrated on the FT62F08x are R/W accessible by instruction, with "CFGS" and "EEPGD" selecting the accessed memory area. 256 x 8-bit DATA EEPROM and 8k x 14-bit (128 page x 64 words) program PROM are independent of each other.

The DATA EEPROM memory array has a typical R/W endurance of 1 M cycles.The address range is 0x00 ~ 0xFF. One byte (8−bit) is written or read at a time.There is no page mode. The program PROM memory array has a typical R/W endurance of 100,000 cycles.The address range is 0 x 0000 ~ 0 x1FFF.One word (14−bit) is written or erased at a time.

DATA EEPROM erase/program is self-timed by hardware, without software queries and saving limited code space. This allows WRITE to take place in the background while the CPU runs unhindered, or even to enter SLEEP. But when the PROM is erased/programmed, the CPU will stop executing instructions.

For DATA EEPROM, READ takes one instruction clock cycle, whereas WRITE takes $T_{WRITE\text{-}EEPROM}$ (3 ~ 5 ms if Auto-Erase is enabled, 1 ~ 3 ms if Auto-Erase is disabled. And for PROM, ERASE takes $T_{ERASE\text{-}PROM}$ (0.75 ~ 1.25 ms), whereas WRITE takes $T_{WRITE\text{-}PROM}$ (0.75 ~ 1.25 ms). There is an on-chip charge pump so there is no need to supply an external high voltage for erase and program the EEPROM and PROM array. Before programming, DATA EEPROM can be configured to Auto-Erase, but PROM must be erased by software. The corresponding interrupt flag EEIF will be set when the EPROM write operation is complete.

There is no sequential READ or sequential WRITE. The address must be updated every time.

Any voltage above V_{POR}, which can be as low as 1.5V from die to die and at high temperature, will be able to run the CPU at 8MHz, 2T. The $V_{DD\text{-}WRITE}$ for writing DATA EEPROM and PROM is higher. For DATA EEPROM , the minimum $V_{DD\text{-}WRITE}$ is 1.9V and 2.2V for Temperature Grade 2 and Grade 1 respectively. For PROM , the minimum $V_{DD\text{-}WRITE}$ is 2.7V. Reading DATA EEPROM and PROM does not have this minimum voltage limit (see $V_{DD\text{-}READ)}$.

10.1. Summary of DATA EEPROM and PROM Related Registers

Name	Status		Register	Addr.	Reset
SYSON	In SLEEP mode, the Sysclk controls 1 = Enable 0 = Disable		CKOCON[7]	0x95	RW−0
EEADR [1]	When EEPGD = 0	DATA EEPROM address	EEADRL[7:0]	0x191	RW−0000 0000
	When EEPGD = 1	LSB(8 bits) of PROM address			
		MSB(5 bits) of the PROM address	EEADRH[4:0]	0x192	RW−---0 0000
EEDAT [1]	when EEPGD = 0	DATA EEPROM data	EEDATL[7:0]	0x193	RW−xxxx xxxx
	When EEPGD = 1	LSB(8 bits) of PROM data			

[1] During a write cycle (see $T_{WRITE\text{-}EEPROM}$ and $T_{WRITE\text{-}PROM}$), this register is not writable.

Name	Status	Register	Addr.	Reset
	MSB(6 bits) of PROM data	EEDATH[5:0]	0x194	RW−xx xxxx
EEPGD	when CFGS = 0 1 = Access to PROM 0 = Access to DATA EEPROM	EECON1[7]	0x195	RW−0
CFGS	1 = Access configuration register (READ access) 0 = Access to PROM or DATA EEPROM	EECON1[6]		RW−0
FREE	PROM operation performed by the next WR command 1 = Erase (cleared by hardware after erasing is finished) 0 = Write Note: Valid only when CFGS = 0 and EEPGD = 1	EECON1[4]		RW−0
WRERR	PROM/DATA EEPROM Erase / Write Error Flag 1 = Premature terminated (any reset except POR) 0 = Finished	EECON1[3]		RW−x
WREN	PROM/DATA EEPROM Write Enable 1 = Enable 0 = Disable	EECON1[2]		RW−0
WR	PROM/DATA EEPROM Write Control 1 = Start a write or writting (After set to 1, wait at least 1 Sysclk to Read back, and it will reset to 0 automatically after the Write is finished) 0 = Finished	EECON1[1]		RW1−0
RD	PROM/DATA EEPROM Reading Control bit 1 = Yes (auto clear after finish) 0 = No	EECON1[0]		RW1−0
EECON2	PROM/DATA EEPROM Write Unlock Control bit Write 0x55 before writing 0xAA for unlock operation , and then set WR to 1. Note: These Write operations must be completed in consecutive instruction cycles and cannot be interrupted.	EECON2[7:0]	0x196	WO−xxxx xxxx
DRDEN	PROM / DATA EEPROM Read Enable 1 = Enable (Wait at least 0.2 μs after setting 1 to read DATA EEPROM) 0 = Disable	EECON3[0]	0x198	RW−0
PONLY	DATA EEPROM Auto-Erase (≥VerI apply)	WProof3 [6]	0x391	RW−0

Name	Status	Register	Addr.	Reset
	1 = No (Do not erase, write only) 0 = Yes (Erase before writing)			

Table 10-1 EEPROM and PROM Related Control Registers

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt 1 = Enable (PEIE, EEIE apply) 0 = Global Shutdown (Wake-Up not affected)		INTCON[7]	Bank first address+0x0B	RW−0
PEIE	Peripheral Interrupt Enable	1 = Enable (EEIE applies) 0 = Disable (no Wake-Up)	INTCON[6]		RW−0
EEIE	EEPROM/PROM Write Completion Interrupt	1 = Enable 0 = Disable (no Wake-Up)	INTCON[5]		RW−0
EEIF[2]	EEPROM/PROM Write Completion Flag	1 = Yes (latched) 0 = No	INTCON[2]		R_W1C−0

Table 10-2 EEPROM and PROM Interrupt Enable and Status Bits

10.2. DATA EEPROM

10.2.1 Write DATA EEPROM

1. Set “GIE = 0”;
2. If “GIE = 1”, then repeat (1);
3. Write target address to EEADRL;
4. Write target data to EEDATL;
5. Set "CFGS = 0" and "EEPGD = 0" to select access to DATA EEPROM;
6. Set “DRDEN = 0” and "WREN=1";
7. Write 0x55 and 0xAA sequentially to EECON2;
8. Set "WR = 1" to start writing;
9. After programming completed (see $T_{WRITE-EEPROM}$ for writing time), “WR = 0” set automatically by hardware;

[2] Write 1 to clear , and writing '0' has no effect on the bit value. It is recommended to only use the STR and MOVWI instructions for write operations, rather than the BSR or IOR instructions .

Program Example:

```
BCR INTCON, GIE
NOP
NOP                             ; interrupt response delay is 2 NOP
BTSC INTCON, GIE
LJUMP $-4

BANKSEL EEADRL
LDWI DATA_EE_ADDR
STR EEADRL                      ; write target address
LDWI DATA_EE_DATA
STR EEDATL                      ; write target data
BCR EECON1, CFGS
BCR EECON1, EEPGD               ; select access to DATA EEPROM
BSR EECON1, WREN

LDWI 55H
STR EECON2                      ; write 0x55 to EECON2
LDWI AAH
STR EECON2                      ; write 0xAA to EECON2

BSR EECON1, WR                  ; start writing
BSR INTCON, GIE                 ;set GIE to 1
BCR EECON1, WREN                ; disable write enable
BTSC EECON1, WR
LJUMP $-2
```

Note:

1. Data EEPROM Read while the array is still in programming will yield an incorrrect result.
2. Before starting the write operation of the DATA EEPROM , it needs to be unlocked, i.e., write 0x55 and 0xAA sequentially to EECON2, and cannot be interrupted, so all interrupts need to be disabled before unlocking.
3. After GIE is cleared, wait for the interrupt response delay of 2 NOP , and then determine if GIE is cleared again.
4. WR is set to 1, wait at least one Sysclk (NOP or any other instructions) for software to read the correct WR value, otherwise it will read back to 0 (mistaken for writing finished).
5. After WR is set to 1, clearing WREN will not affect the current write cycle.
6. When the write of DATA EEPROM is finished, WREN needs to be cleared by software, this protection mechanism can prevent accidental write operations. In addition, the Power-up Timer PWRT (~ 64 ms) also prevents writing to the DATA EEPROM.

10.2.2 Read DATA EEPROM

1. Set “GIE = 0”;
2. If “GIE = 1”, then repeat (1);
3. Set “DRDEN = 1”, and wait for 0.2μs;
4. Write target address to EEADRL;
5. Set "CFGS = 0" and "EEPGD = 0" to select access to DATA EEPROM ;
6. Set "RD = 1" to start reading;
7. Read the target data from EEDATL. The EEDATL register will hold this value until the next read or write operation. "RD" will auto clear by hardware;

The following is an example on how to read the DATA EEPROM:

```
BCR INTCON, GIE
NOP
NOP                             ; Interrupt response delay is 2 NOP
BTSC INTCON, GIE
LJUMP $-4

BANKSEL EEADRL
BSR EECON3, DRDEN
NOP                             ; wait 0.2us
LDWI DATA_EE_ADDR
STR EEADRL                      ; Write target address
BCR EECON1, CFGS
BCR EECON1, EEPGD               ; Select access to DATA EEPROM
BSR EECON1, RD                  ; Start reading
LDR EEDATL, W                   ; Data is read by instruction
```

Note:

1. The EEPROM can always be read by software regardless of the value of CPB.
2. After reading the data, the DRDEN needs to be cleared to save power.

10.2.3 Auto-erase

Writing data into a byte involves two steps: a byte erase followed by a byte program. Erase sets all the bits in the byte to “1” while program can selectively program individual bits to “0”. This device has a built in auto-erase function (set PONLY = 0) where an erase always preceed program. Except for high temperature it is recommended to turn on Auto-Erase.

If Auto-erase is enabled, multiple programming of FF data is actually multiple erasure of the corresponding byte. However programming non-FF multiple times is the same as programming the byte once, as every time the bit programs it will be auto-erased first. Only when Auto-erase is disabled, multiplying programming will have a cumulative effect. There are situations when one would like to turn off Auto-erase, to do cumulative programming to ensure successful programming, such as at very high temperature. See the flow below:

1. Make sure Auto-Erase is ON.
2. Erase the byte.
3. Read DATA EEPROM.
4. If byte is FF then continues, else go back to step (2).
5. Erase the same number of times as (2) again to ensure it is strongly erased.
6. Disable Auto-erase.
7. Program the desired data.
8. Read DATA EEPROM.
9. If the byte has the desire data then continues, else go back to (7).
10. Program the same number of times as (7) cumulatively to ensure it is strongly programmed.

10.3 Program PROM

The program address counter PC is 15 bits (0x0000 ~ 0x7FFF), supporting up to 32k address space. FT61F0Ax / FT64F0Ax implements a 8k program PROM, divided into 128 pages x 64 words (1 word = 14 bits), and the address range is 0x0000 ~ 0x1FFF. When the program address exceeds 0x1FFF, it will cause a rewind to 0x0000.

The software needs to erase the program PROM before executing the programming operation.

Note:

1. When BOOT (see FSECPB0) is set to sector encryption, the encrypted PROM sectors can only be read, not erased or written.

10.3.1 Erase Program PROM

The unit of software erasing PROM is 1 page (64 words)。

1. Set “GIE = 0”;
2. If “GIE = 1”, then repeat (1);
3. Write target address to EEADRL and EEADRH;
4. Set "CFGS = 0" and "EEPGD = 1" to select access to PROM;
5. Set "FREE = 1" , "WREN = 1" and “DRDEN = 0”;
6. Write 0x55 and 0xAA sequentially to EECON2;
7. Set "WR = 1" to start erasing;
8. After the erasing is finished (the erasing time $T_{ERASE-PROM}$ is 0.75 ~ 1.25 ms), “FREE” will auto clear by hardware;

Program Example:

```
BCR INTCON, GIE
NOP
NOP                             ; Interrupt response delay is 2 NOP
BTSC INTCON, GIE
LJUMP $-4

BANKSEL EEADRL
LDR ADDRL,W
STR EEADRL                      ; Write the LSB(8 bits) of the target address
LDR ADDRH,W
STR EEADRH                      ; Write the MSB(7 bits) of the target address
BCR EECON1, CFGS
BCR EECON1, EEPGD               ; Select access to PROM
BSR EECON1, FREE
BSR EECON1, WREN

LDWI 55H
STR EECON2                      ; Write 0x55 to EECON2
LDWI AAH
STR EECON2                      ; Write 0xAA to EECON2

BSR EECON1, WR                  ; Start erasing
NOP
NOP                             ; Wait for 2 NOP to set Erase
BCR EECON1, WREN                ; Disable Write enable
BSR INTCON, GIE                 ; GIE set to 1
```

Note:

1. Before starting the erase operation of the PROM, it needs to be unlocked, that is, write 0x55 and 0xAA sequentially to EECON2, and cannot be interrupted, so all interrupts need to be disabled before unlocking.
2. After WR is set to 1, it takes 2 instruction cycles for the processor to set the erase operation, so 2 NOP instructions must follow immediately after the erase instruction.
3. During the erase cycle $T_{ERASE-PROM}$, the CPU will suspend execution of instructions and the clocks and peripherals will continue to run.
4. After the erase is complete, the program will continue to execute from the third instruction after the Erase instruction.

10.3.2 Write Program PROM

The unit of software programming PROM is 1 word (14 bits), so 1 page needs to be programmed 64 times. Before programming, the corresponding address must be erased or unprogrammed.

1. Set “GIE = 0”;
2. If “GIE = 1”, then repeat (1);
3. Write target address to EEADRL and EEADRH;

4. Set "CFGS = 0" and "EEPGD = 1" to select access to PROM;
5. Set “DRDEN = 0”, "FREE = 0" and "WREN = 1";
6. Write target data to EEDATL and EEDATH;
7. Write 0x55 and 0xAA sequentially to EECON2;
8. Set "WR = 1" to start writing;
9. After writing is finished (programming time $T_{WRITE\text{-}PROM}$ is 0.75 ~ 1.25 ms), "WR" will auto clear by hardware;
10. Repeat the above steps until all data is written;

Program Example (target data is loaded via indirect addressing):

```
BCR INTCON, GIE
NOP
NOP                           ; Interrupt response delay is 2 NOP
BTSC INTCON, GIE
LJUMP $-4

BANKSEL EEADRL
LDR ADDRL,W
STR EEADRL                    ; Write the LSB (8 bits) of the target address
LDR ADDRH,W
STR EEADRH                    ; Write the MSB (7 bits) of the target address
LDWI LOW DATA_ADDR            ; Load the LSB (8 bits) of the address of the target data
STR FSR0L
LDWI HIGH DATA_ADDR           ; Load the MSB (7 bits) of the address of the target data
STR FSR0H
BCR EECON1, CFGS
BCR EECON1, EEPGD             ; Select access to PROM
BCR EECON1, FREE
BSR EECON1, WREN

MOVIW FSR++
STR EEDATL                    ; Write the LSB (8 bits) of the target data
MOVIW FSR++
STR EEDATH                    ; Write MSB (6 bits) of target data

LDWI 55H
STR EECON2                    ; Write 0x55 to EECON2
LDWI AAH
STR EECON2                    ; Write 0xAA to EECON2

BSR EECON1, WR                ; Start writing
NOP
NOP                           ; Set the write operation to wait for 2 NOPs
BCR EECON1, WREN              ; Disable write enable
BSR INTCON, GIE               ; Set GIE to 1
```

Note:

1. When writing target data to EEDATL and EEDATH, it will be loaded into the 14-bit write latch. After finishing writing, the write latch will be reset to 0x3FFF.
2. Before starting the PROM write operation, it needs to be unlocked, i.e., write 0x55 and 0xAA to EECON2 sequentially and cannot be interrupted, so all interrupts need to be disabled before unlocking.
3. After WR is set to 1, it takes 2 instruction cycles for the processor to set the write operation, so 2 NOP instructions must be followed immediately after the write instruction.
4. During the cycle $T_{WRITE\text{-}PROM}$, the CPU will suspend instruction execution, and the clocks and peripherals will continue to run.
5. After finishing writing, the program will continue execution from the third instruction after the Write instruction.
6. When some PROM data needs to be changed, and other data of the current page needs to be preserved, modify it according to the following steps:
   a) Load the starting address of the page to be modified;
   b) Read all the data of the current page and save it to the RAM mapping area;
   c) Modify the new data to be changed in the RAM mapping area;
   d) Load the starting address of the page to be modified and erase the current page;
   e) Write the data of the RAM mapped area to the current page one by one according to the programming steps;

10.3.3 Read Program PROM

1. Set “GIE = 0”;
2. If “GIE = 1”, then repeat (1);
3. Write target address to EEADRL and EEADRH;
4. Set "CFGS = 0" and "EEPGD = 1" to select access to PROM;
5. Set "RD = 1" to start reading;
6. After waiting for 2 instruction cycles, the PROM data is written to the EEDATH:EEDATL register, thus 2 NOP instructions must follow immediately after the read instruction. RD" will auto clear by hardware. The EEDATH:EEDATL register will hold this value until the next Read or Write operation.

The following is an example on how to read the PROM:

```
BCR INTCON, GIE
NOP
NOP                                     ; Interrupt response delay is 2 NOP
BTSC INTCON, GIE
LJUMP $-4

BANKSEL     EEADRL
LDWI PROG_ADDR_LO
```

```
STR EEADRL                   ; Write the LSB (8 bits) of the target address
LDWI PROG_ADDR_HI
STR EEADRH                   ; Write the MSB (7 bits) of the target address
BCR EECON1, CFGS
BSR EECON1, EEPGD            ; Select access to PROM
BSR EECON1, RD               ; Start reading
NOP
NOP                          ; Read wait for 2 NOPs
LDR EEDATL, W                ; LSB(8 bits) of read data
STR PROG_DATA_LO
LDR EEDATH, W                ; MSB (6 bits) of read data
STR PROG_DATA_HI
```

Notice:

1. The PROM can always be read by software regardless of the value of CPB.

10.4. Read BOOT Register UCFGx

When CFGS = 1, software can read the BOOT register area UCFGx (see **Section 17.1**). UCFGx and program PROM are independent of each other, and the address starts from 0x8000. For unimplemented units, read returns undefined.

11. 12−bit ADC

The ADC can convert the analog input signal into a 12-bit digital signall and operate at different clock speeds with a 11-bit accuracy at clock speeds up to 4 MHz (i.e. 200 kHz sample rate, 5 µs/sample).

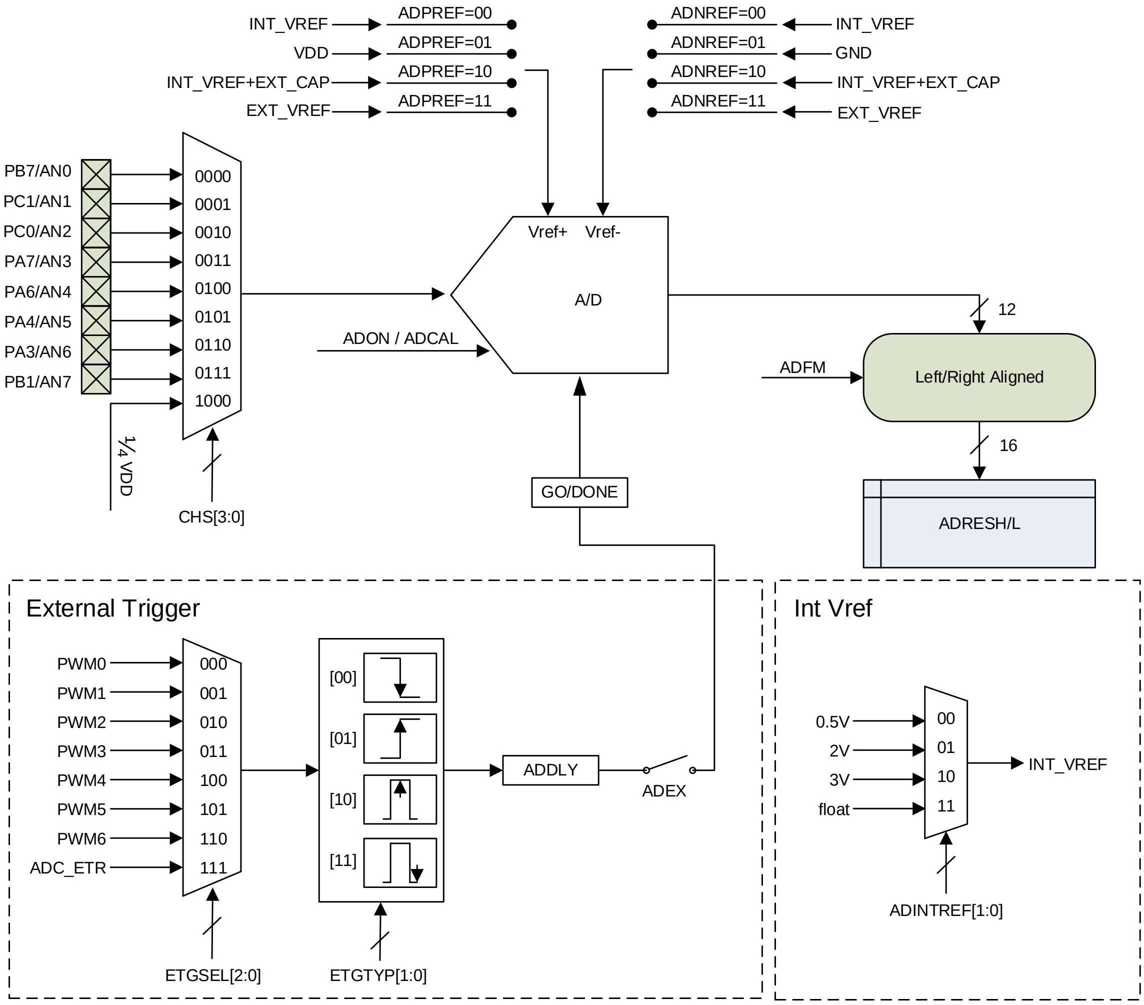


Figure 11-1 ADC Block Diagram

The analog input signal can be selected as one of eight I/O (ANx) channels or an internal channel (1/4VDD). ADC is triggered by instructions, I/O (PA4 / PB3) or PWM. A delay or leading edge blanking (LEB) can be added between trigger and ADC sampling.

When the ADC conversion is complete, the corresponding interrupt flag will be set, it can trigger an interrupt and/or Wake-up from SLEEP.

The ADC reference voltage ($V_{ADC-REF)}$ can be selected as V_{DD} by instructions, one of three internal reference voltages (0.5V, 2V, 3V), or input the external reference voltage via I/O.

ADC can be calibrated automatically. In addition, the ADC conversion process runs in the background, and the CPU can execute other instructions during the conversion.

If the ADC needs to keep active in SLEEP:

1. Set SYSON = 1 to keep Sysclk active;

2. When the ADC convertion clock source is LIRC, LIRC will keep active after entering SLEEP, regardless of SYSON;

When the ADC is configured as hardware trigger (PA4/PB3 or PWM), GO/DONE is directly set by the hardware trigger event and starts A/D conversion, and the software set GO/DONE will be ignored.

In applications with high sampling rates, there are 3 time points to pay attention to when using the ADC:

1. The moment when the selected channel starts sampling.
2. The moment when the sampling ends. Immediately before the sample-and-hold circuit is disconnected, the voltage value on the selected channel is used to measure the conversion.
3. The moment when the data conversion is completed.

11.1. Summary of ADC Related Registers

Name	Addr.	bit 7	bit 6	bit 5	bit 4	bit 3	Bit 2	bit 1	bit 0	Reset
PCKEN	0x9A	TKEN	I2CEN	UARTEN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
ADRESL	0x9B	LSB of A/D conversion result								xxxx xxxx
ADRESH	0x9C	MSB of A/D conversion result								xxxx xxxx
ADCON0	0x9D	CHS[3:0]				ADCAL	ADEX	GO/DONE	ADON	0000 0000
ADCON1	0x9E	ADFM	ADCS[2:0]			ADNREF[1:0]		ADPREF[1:0]		0000 0000
ADCON2 [1]	0x9F	ADINTREF[1:0]		ETGTYP[1:0]		ADDLY.8	ETGSEL[2:0]			0000 0000
ADDLY [1]	0x1F	ADDLY[7:0] / LEBPRL[7:0]								0000 0000
ADCON3 [1]	0x41A	ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	LEBADT	−	ELVDS[1:0]		0000 0−00
ADCMPH	0x41B	ADCMPH[7:0]								0000 0000
LEBCON [1]	0x41C	LEBEN	LEBCH		−	EDGS	BKS2	BKS1	BKS0	000− 0000

Table 11-1 ADC Related Register Address

[1] This register is still readable and writable when ADCEN = 0.

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt	1 = Enable (PEIE, ADCIE apply) 0 = Global Shutdown (Wake-Up not affected)	INTCON[7]	Bank first address+ 0x0B	RW−0
PEIE	Peripheral Interrupt Enable	1 = Enable (ADCIE applies) 0 = Disable (no Wake-Up)	INTCON[6]		RW−0
ADCIE	ADC Conversion Completion Interrupt	1 = Enable 0 = Disable (no Wake-Up)	PIE1[0]	0x91	RW−0
ADCIF [2]	ADC Conversion Completion Flag	1 = Yes (latched) 0 = No	PIR1[0]	0x11	R_W1C−0

Table 11-2 ADC Interrupt Enable and Status Bits

Name	Status	Register	Addr.	Reset

ADRESL	LSB of A/D conversion result
	ADFM=0: ADRESL[7:4] = least significant of 4 bits (the rest are "0")
	ADFM=1: ADRESL[7:0] = LSB
		ADRESL[7:0]	0x9B	RW−0000 0000

ADRESH	MSB of A/D conversion result
	ADFM=0: ADRESH[7:0] = MSB
	ADFM=1: ADRESH[3:0] = Most significant of 4 bits (the rest are "0")
		ADRESH[7:0]	0x9C	RW−0000 0000

SYSON	In SLEEP mode, the system clock controls 1 = Enable 0 = Disable	CKOCON[7]	0x95	RW−0
ADCEN	ADC Clock 1 = Enable 0 = Disable	PCKEN[0]	0x9A	RW−0
CHS	ADC Analog Input Channel 0000 = AN0 0101 = AN5 0001 = AN1 0110 = AN6 0010 = AN2 0111 = AN7 0011 = AN3 1000 = 1/4 V_{DD} 0100 = AN4 1xxx = Reserved	ADCON0[7:4]	0x9D	RW−0000
ADCAL	ADC Auto Calibration Enable (settable when ADON = 0)	ADCON0[3]		RW−0

[2] Write 1 to clear , and writing '0' has no effect on the bit value. It is recommended to only use the STR and MOVWI instructions for write operations, rather than the BSR or IOR instructions .

Name	Status	Register	Addr.	Reset
	1 = Calibration on / Calibration in progress (auto clear when calibration is complete) 0 = Calibration completed / Not started			
ADEX	ADC Trigger Condition (GO/DONE) 1 = GO/DONE is set by PA4 / PB2 or PWM (hardware triggered) 0 = GO/DONE is set by instructions (software triggered)	ADCON0[2]		RW−0
GO/DONE	ADC Conversion Start and status bits 1 = A/D conversion enabled by software, PA4/PB2 or PWM (auto clear after conversion is complete) 0 = Conversion completed / No conversion	ADCON0[1]		RW−0
ADON	1 = ADC enable 0 = ADC disable (no current consumption)	ADCON0[0]		RW−0
LFMOD	1: LIRC = 256 kHz 0: LIRC = 32 kHz	TCKSRC[7]	0X31F	RW−0
ADFM	A/D Conversion Result Format (See "ADRESH") 1 = Right aligned 0 = Left aligned	ADCON1[7]	0x9E	RW−0
ADCS	A/D Conversion Clock Sources 000 = SysClk/2 100 = SysClk/4 001 = SysClk/8 101 = SysClk/16 010 = SysClk/32 110 = SysClk/64 011 = LIRC (*) 111 = LIRC (*) (*) LIRC = 32kHz or 256kHz, depends on the value of LFMOD	ADCON1[6:4]		RW−000
ADNREF	$V_{ADC-REF}$ – (negative reference voltage) 00 = Internal $V_{ADC-REF}$ 01 = GND 10 = Internal $V_{ADC-REF}$ + External Capacitor C_{EXT} 11 = External reference voltage (I/O)	ADCON1[3:2]	0x9E	RW−00
ADPREF	$V_{ADC-REF}$ + (positive reference voltage) 00 = Internal $V_{ADC-REF}$ 01 = V_{DD} 10 = Internal $V_{ADC-REF}$ + External Capacitor C_{EXT} 11 = External reference voltage (I/O)	ADCON1[1:0]		RW−00
ADINTREF	Internal $V_{ADC-REF}$	ADCON2[7:6]	0x9F	RW−00

Name	Status	Register	Addr.	Reset
	00 = 0.5 01 = 2.0 10 = 3.0 11 = (not connected)			
ETGTYP	External Trigger Edge (applicable when ADEX=1) 00 = (PWM or PA4/PB2−ADC_ETR) Falling edge 01 = (PWM or PA4/PB2−ADC_ETR) Rising edge 10 = Midpoint of one PWM period (*) 11 = End of one PWM period (*) (*) : The center-aligned PWM mode of TIM1 is selected by default;	ADCON2[5:4]		RW−00
ADDLY.8 / LEBPR9	8th bit of LEB Counter or ADC Delay Counter (See "ADDLY")	ADCON2[3]		RW−0
ETGSEL	External Trigger Source (applicable when ADEX=1) 000 = PWM1, TIM1_CH1 100 = PWM5, TIM2_CH1 001 = PWM2, TIM1_CH2 101 = PWM6, TIM2_CH2 010 = PWM3, TIM1_CH3 110 = PWM7, TIM2_CH3 011 = PWM4, TIM1_CH4 111 = ADC_ETR	ADCON2[2:0]		RW−000
ADDLY / LEBPRL	ADC Delay/LEB (Non-software triggered, i.e. valid when ADEX = 1) (This is the LSB, ADDLY.8 is the MSB) Delay time = (ADDLY+6) x T_{AD} (If the PWM output is enabled to trigger the ADC, ADDLY must not be changed during PWM operation)	ADDLY[7:0]	0x1F	RW−0000 0000
ADFBEN	ADC Threshold Comparison Result Match Event Triggers PWM Fault-Break 1 = Enable 0 = Disable	ADCON3[7]	0x41A	RW−0
ADCMPOP	Polarity of ADC Threshold Comparison 1 = MSB of ADC result < ADCMPH[7:0] 0 = MSB of ADC result ≥ ADCMPH[7:0]	ADCON3[6]		RW−0
ADCMPEN	ADC Threshold Comparison 1 = Enable 0 = Disable (Clear the break event generated by ADCMP)	ADCON3[5]		RW−0

Name	Status	Register	Addr.	Reset
ADCMPO	ADC Comparison Output (Updated every time A/D conversion is completed) When ADCMPOP = 1 1 = MSB of ADC result < ADCMPH[7:0] (no latch) 0 = MSB of ADC result ≥ ADCMPH[7:0] when ADCMPOP = 0 1 = MSB of ADC result ≥ ADCMPH[7:0] (no latch) 0 = MSB of ADC result < ADCMPH[7:0]	ADCON3[4]		RO−0
LEBADT	ADC starts automatic conversion after LEB 1 = Trigger ADC conversion 0 = No ADC conversion triggered	ADCON3[3]		RW−0
ADCMPH	ADC Comparison Threshold (MSB only, 0.4% steps)	ADCMPH[7:0]	0x41B	RW−0000 0000
LEBEN	ADC Trigger and LEB Enable of BKIN 1 = Enable 0 = Disable (Switching when GO/DONE=1 will produce unexpected results)	LEBCON[7]	0x41C	RW−0
LEBCH	LEB Signal Source 00 = TIM1_CH1 01 = TIM1_CH2 10 = TIM1_CH3 11 = TIM1_CH4	LEBCON[6:5]		RW−00
EDGS	LEB Trigger Edge 1 = Falling edge 0 = Rising edge	LEBCON[3]		RW−0

Table 11-3 ADC Related Registers

11.2. ADC Configuration

Configuring the ADC includes the following settings (When changing configuration, set ADON = 0 to turn off A /D conversion or external trigger):

- ADC clock module
- Channel selection
- ADC reference voltage
- ADC conversion clock source
- Conversion result format
- ADC Calibration

- Trigger source
- ADC Delay or Leading Edge Blanking (LEB)
- Threshold comparison (optional)
- Response (interrupt setting)

Channel Selection – The input channel is selected by the CHS register, which is connected to the sample-and-hold circuit for ADC conversion, with corresponding I/O configured as analog input by setting TRISx = 1 and ANSELAx = 1.

ADC Reference Voltage ($V_{ADC-REF}$) – The ADC measures the input analog voltage with 2 reference voltages as relative values: V_{REF+} and V_{REF-}. The reference voltage can be selected as follow choices:

- VDD can be selected as V_{REF+}, GND can be selected as V_{REF-}
- Internal Reference Voltage
- Internal Reference Voltage plus external capacitor C_{EXT}
- External Reference Voltage (V_{REF+} for PB5, $V_{REF-}$for PB6)

V_{REF+} and V_{REF-}can be different combinations of the above selections, the internal reference voltage cannot be selected at the same time, otherwise V_{REF-}will be forced to connect to GND.

The internal reference voltage can be 0.5V , 2.0V , 3.0V or "Not connected" (see "ADINTREF", **Table 11-3**)。

ADC Conversion Clock Selection – The ADC can select 7 clock frequencies by instructions (see "ADCS", **Table 11-3**):

- SysClk/N; N = 2, 4, 8, 16, 32, 64
- LIRC (256 kHz or 32 kHz, see "LFMOD")

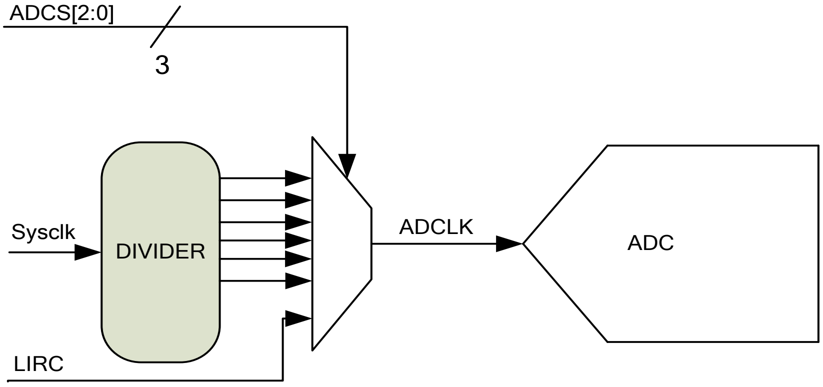


Figure 11-2 ADC Clock Configuration

Conversion Result Format – A/D conversion results can be stored in either left-aligned or right-aligned formats (see “ADFM” in **Table 11-3**). A/D auto-calibration values are also affected by this format.

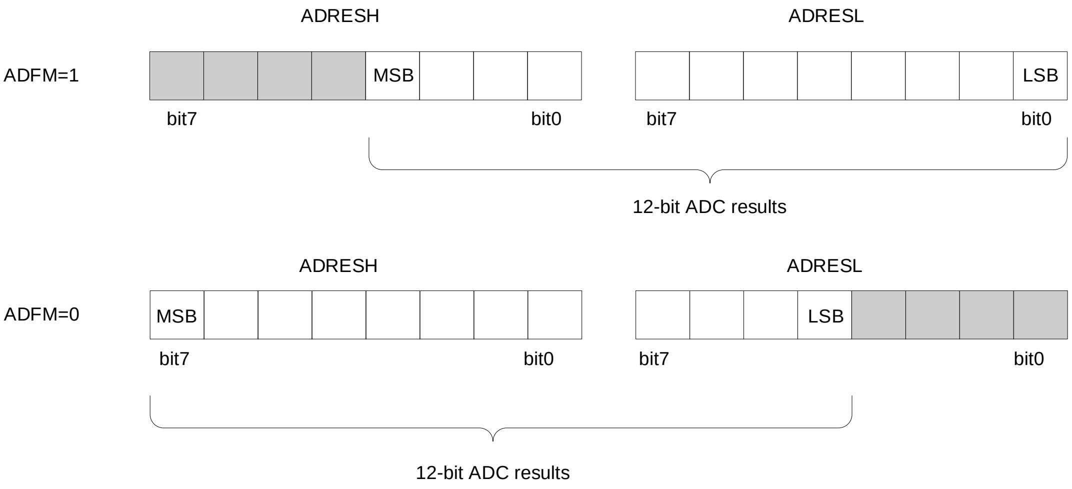


Figure 11-3 ADC Conversion Result Format

ADC Auto-Calibration – It is recommended that the ADC be calibrated at least once before starting the conversion, calibration enables self-correction of ADC offset errors. Automatic calibration can be initiated by setting "ADCAL = 1". After calibration is completed, the ADC module is in the calibrated state and the calibration value is always stored but not visible, any reset will invalidate it.

Calibration steps:

1. Set ADON = 0 (ADON and ADCAL cannot be 1 at the same time);
2. Select V_{REF+} and V_{REF-} (need to be selected correctly, the calibration result will affect the subsequent ADC conversion);
3. Set ADCAL = 1;
4. Auto calibration is completed, ADCAL auto clears.

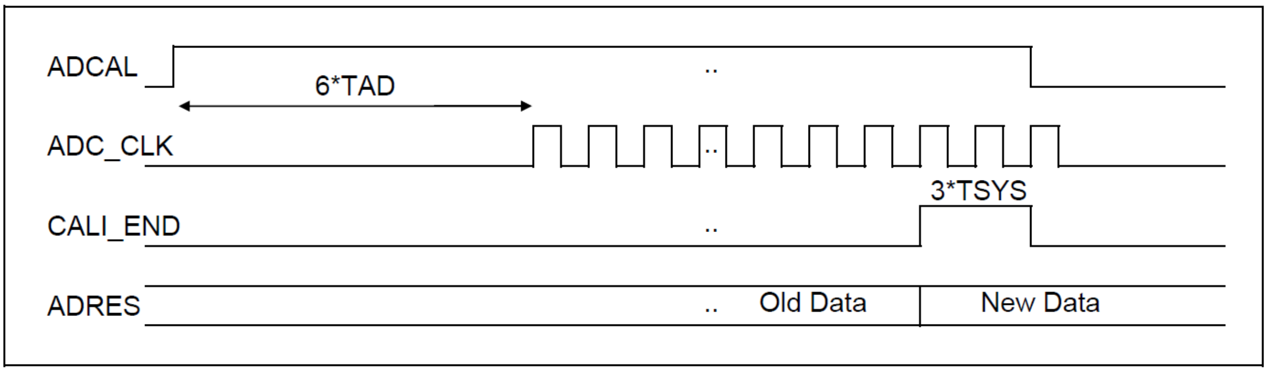


Figure 11-4 ADC Auto-Calibration Timing Diagram

11.2.1. ADC Trigger and Delay Configuration

ADC conversion can be triggered by an instruction (ADEX = 0) , PWM (edge / cycle) or I/O (PA4 / PB2) transition edge (ADEX = 1). Among them , the trigger type of PWM can be selected as "Rising edge" , "Falling edge" , " Midpoint of cycle " or " End of cycle " , and the trigger edge of PA4 / PB2 can be selected as "Rising edge" or "Falling edge" (see "ETGTYP" and "ETGSEL" , **Table 11-3**).

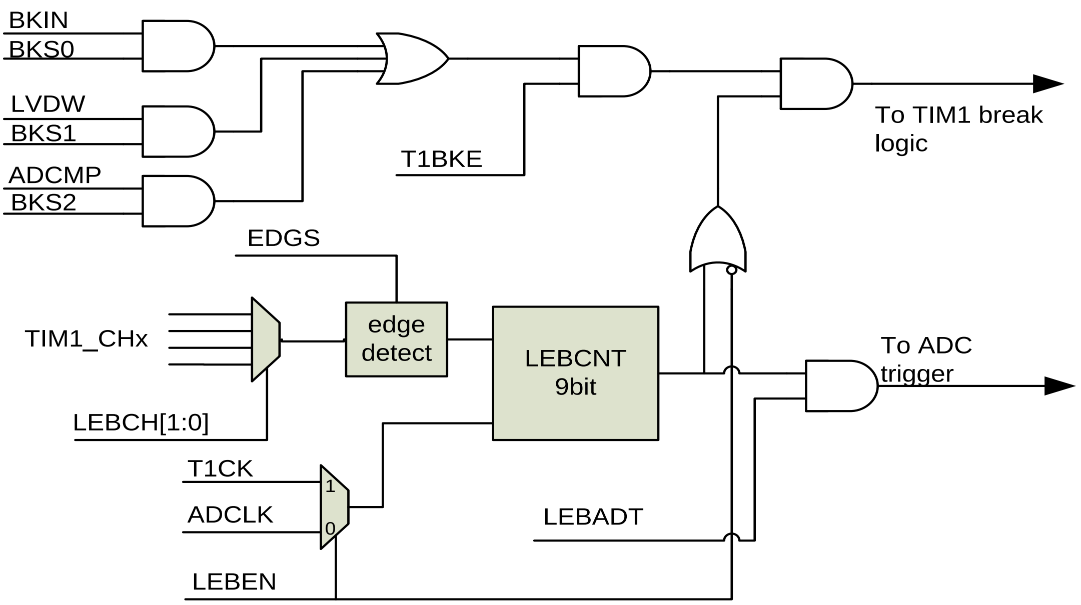


Figure 11-5 LEB Block Diagram

In high-speed switching applications, the conduction of switching devices (e.g. MOSFETs/IGBTs) usually generates extremely large transient currents immediately, and these transients can lead to measurement errors. Considering the LEB feature, applications can ignore the expected transients caused by the MOSFETs / IGBTs near the PWM Output Edge.

The Clock Source for both LEB and PWM is T1CK (Timer1 Clock Source). During LEB timing, the ADC keeps sampling until LEB timing overflows (see "LEBPR", **Table 11-3**). If an active LEB trigger edge occurs again during the LEB timing cycle, the LEB timer will resume and start counting again.

Trigger Conditions	Delay/LEB	Trigger Channel
Instruction	(No delay)	(N/A)
I/O (PA4/PB2)	(ADDLY+6) x T_{AD}; ADDLY = LEBPR	I/O (PA4/PB2)
PWM	(LEBPR+6) x T_{AD}	LEBEN = 0; ETGSEL (LEBCH ignored)
	(LEBPR+3) x T_{T1CK} + 3 x T_{AD} (T_{T1CK} = Timer1 period)	LEBEN = 1; LEBCH (ETGSEL ignored)

Table 11-4 ADC Trigger, Delay and Channel Settings

If triggered by software (ADEX = 0) , A/D conversion starts immediately after GO/DONE is set by the instruction. If triggered by PA4/PB2 or PWM , there is a certain delay time ("6 x T_{AD}" or "3 x T_{T1CK} + 3 x T_{AD}", see **Table 11-4**) . Before GO/DONE is set, additional delay can be added by setting the ADDLY/LEBPR registers . The ADC delay timer (ADDLY) and the LEB timer (LEBPR) share the same 9-bit counter, which consists of LEBPR9 and LEBPRL[7:0] . After the delay, the sample-and-hold circuit will be disconnected within “3.5 x T_{AD} – 4.5 x T_{AD} ” time.

Note:

1. ADEX and ADON registers need to be set before enabling LEB.
2. New trigger conditions will be ignored until the ADC conversion is completed. .

3. If LEBEN=1, ETGSEL will be ignored, and will share the trigger source with LEB. At this time, the automatic conversion of the ADC will be triggered by the overflow of the LEB timer (see "LEBADT" or "EDGS", **Table 11-3**).

11.2.2. ADC Aborts Conversion

Sometimes ADC need to be aborted, such as starting new sampling.

- When ADEX = 0 (instruction triggered), the ADC can be aborted by software setting GO/DONE = 0.
- When ADEX = 1, the ADC must be stopped by turning off the ADC module (ADON = 0).
- When ADC conversion is aborted, the aborted operation takes 4 x T_{AD} to process, after which ADRESH and ADRESL will be partially updated with the value of the converted completed bits and the incomplete bits will all be filled with the value of the last converted bit.
- When the system is reset, the ADC will be aborted and the ADC module will be turned off because the corresponding registers are reset.

11.2.3. Threshold Comparison

The ADC can automatically compare the result with the threshold previous value in the ADCMPH register after the conversion is completed (see "ADCMPEN"). The comparison polarity is set by ADCMPOP, and the comparison result is output by ADCMPO. The PWM Fault Break can be triggered when the corresponding matching condition occurs (see "ADFBEN"). Only the MSB of the conversion result is used for the threshold comparison, so the comparison step between V_{REF+} and V_{REF-} is 0.4%.

Note:

1. ADCMPO will be cleared when ADCMPEN = 0 or ADON = 0; when entering sleep mode, ADCMPO will not be cleared.

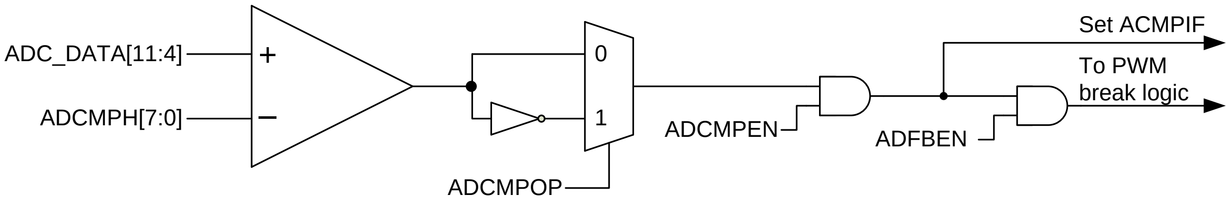


Figure 11-6 ADC Threshold Comparison Block Diagram

11.2.4. Interrupt

The corresponding interrupt flag ADCIF will be set when the ADC conversion is completed. Whether to trigger an interrupt and/or wake up from SLEEP depends on the corresponding interrupt enable controls (GIE, PEIE, ADCIE).

Note:

1. ADCIF is set every time an ADC normal conversion completes, regardless of whether the Interrupt Enable is on.
2. ADCIF will not be set when automatic calibration is completed or when software aborts AD conversion.

11.3. Sample-and-hold time

The sample hold time, T_{ACQ} must be long enough to ensure that the internal ADC voltage is stable within 0.01% of the input channel voltage, resulting in 12-bit accuracy (0.024%). The relationship between the sample-and-hold time and the external series resistance is as follows (**Table 11-5**):

$$T_{ACQ} > 0.03 \text{ x } (R + 1) \text{ μs . (the unit of R is kΩ)} \quad .$$

When T_{ACQ} is 0.5μs, the external series resistor must ≤ 15 kΩ . If a larger series resistor is used, T_{ACQ} will increase proportionally. Junction leakage current limits the maximum allowable value of series resistance. For a junction leakage current of 5nA, a voltage drop of 0.25mV (0.0125% of the 2V reference) occurs across the series resistance of 50 kΩ . When the temperature exceeds 100°C, the junction leakage current will increase significantly. Therefore, the smaller the series resistance is, the better.

Series resistance	T_{ACQ}
> 50kΩ	(Not recommended)
48 kΩ	≥ 1.5 μs
32 kΩ	≥ 1.0 μs
< 15 kΩ	≥ 0.5 μs

Table 11-5 Correspondence between different external series resistances and the shortest T_{ACQ}

The sample-and-hold time is the time that the internal ADC observes the voltage of the input channel.

Start of sample-and-hold time = after channel switching (see “CHS”) or ADC stabilization (see T_{ST}), the longer time shall prevail.

End of sample-and-hold time = 3.5 – 4.5 x T_{AD} after the hardware trigger delay or setting GO/DONE to 1 by software, the delay time is determined by the trigger condition (see **Table 11-4**), and the sample-and-hold circuit is disconnected.

Sampling point = the instant before the sample-and-hold circuit is disconnected, with the uncertainty of 3.5 – 4.5 x T_{AD}.

Data conversion starts after turnning off the sampling, and the conversion process takes 13.5 x T_{AD}. Therefore, it takes 17 x T_{AD} to 18 x T_{AD} from the end of the hardware trigger delay or the software GO/ONE setting to the completion of data conversion. After the data conversion is completed, the sample-and-hold circuit is closed again to start the next sampling cycle. It is also necessary to wait for a sufficient sampling time T_{ACQ} before starting A/D conversion again.

11.4. Minimum sampling time

T_{AD} is the clock cycle of the ADC. The minimum time required to complete 12−bit conversion: T_{ACQ} + 18 x T_{AD}

The maximum sampling rate that can guarantee the accuracy of 11−bit accuracy is 200 kHz (~5 μs/sample).

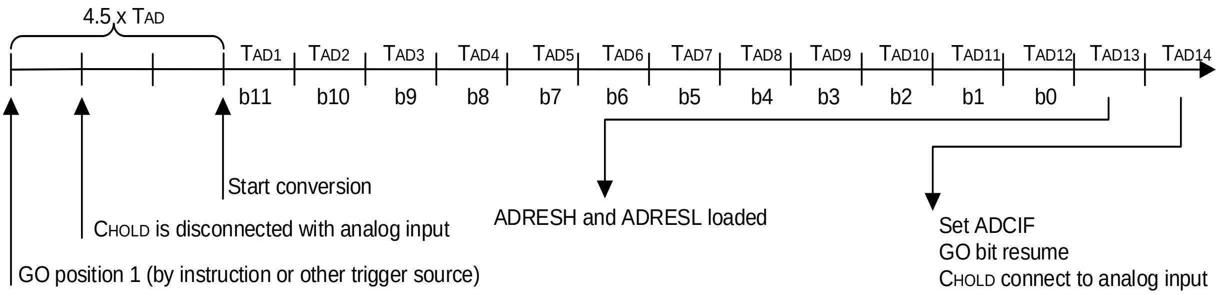

4.5 x TAD
TAD1 TAD2 TAD3 TAD4 TAD5 TAD6 TAD7 TAD8 TAD9 TAD10 TAD11 TAD12 TAD13 TAD14
b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Start conversion
CHOLD is disconnected with analog input
GO position 1 (by instruction or other trigger source)
ADRESH and ADRESL loaded
Set ADCIF
GO bit resume
CHOLD connect to analog input

Figure 11-7 Analog-to-Digital conversion T_{AD} cycle

11.5. Example of ADC Conversion Steps

To set up the ADC:

1. Set ADCEN = 1 to turn on the ADC module clock.
2. Configure the PORT:
   a. Set TRISx = 1 to disable pin output driving;
   b. Set ANSELAx = 1 to disable digital input, weak pull-up and weak pull-down.
3. Configure the ADC module:
   a. Select ADC conversion clock source;
   b. Select ADC reference voltage;
   c. Select ADC trigger conditions: software, PA4/PB3-ADC_ETR or PWM, with or without LEB;
   d. Select the conversion result format;
4. ADC auto calibration (recommended to turn on):
   a. Set ADCAL = 1, start auto calibration;
   b. Wait and query ADCAL, it will auto clear after calibration is completed;
5. Set threshold comparison (optional)
6. Configure ADC interrupt (optional):
   a. Enable ADC conversion completion interrupt;
   b. Enable Master Peripheral Interrupt;
   c. Global Interrupt Shutdown (enable if the interrupt service routine needs to be executed);
7. Turn on the ADC module. Then wait for the required stabilization time T_{ST} (~15 µs), when $V_{ADC-REF}$ selects the internal reference voltage, then wait for the longer of the internal reference voltage stabilization time T $_{VRINT}$ (see "T_{VRINT} ", **Section 19.7**) and T_{ST} time , namely max(T_{VRINT} ,T_{ST}) .

So far, the ADC is ready to sample the different channels. When sampling the input channel:

1. The ADC input is selected as the channel to be measured (see "CHS").
2. If necessary, clear the ADC conversion completion interrupt flag ADCIF.

3. The shortest sampling time T_{ACQ}, which must be long enough to ensure that the internal ADC input capacitor is fully charged within 0.01% of the input channel voltage. In addition, depending on the trigger type, there may be a delay in retriggering after switching channels or after the ADC has stabilized (the longer time shall prevail).

   a. For software triggering, additional T_{ACQ} is required.

   b. For PA4/PB2-ADC_ETR or PWM triggering, unless a very large series resistor is used, the internal delay time (ADDLY+6) x T_{AD} is usually longer than T_{ACQ}, so no additional delay T_{ACQ} is required .

4. After waiting the delay required, set GO/DONE by instruction, or wait for a hardware trigger event to automatically set GO/DONE to start the A/D conversion. After GO/DONE is set, it need to wait for one Sysclk cycle to read back the GO/DONE flag.

5. Wait for the conversion to complete by:

   a. waiting for one sysclk cycle and querying the GO/DONE bit;

   b. waiting for ADC interrupt (when interrupt is enabled).

6. Read ADC conversion result.

7. If necessary, clear the ADCIF.

Note:

1. Although GO/DONE and ADON are in the same register (ADCON0), they should not be set at the same time.

2. The configuration cannot be changed during conversion or while waiting for an external trigger. Changes are recommended when ADON = 0.

The following is an example of the ADC program (the input sampling channel is PB7, and the ADC clock is LIRC):

```
BANKSEL PCKEN
BSR PCKEN,0                 ; ADC module clock
BANKSEL TRISB
BSR TRISB,7                 ; Set PB7 to input
BANKSEL ANSELA
BSR ANSELA, 0               ; Set PB7 to analog
BANKSEL ADCON1
LDWI B’11110101’            ; Right justify, ADC LIRC clock
STR ADCON1                  ; Vref+: VDD , Vref-: GND
BANKSEL ADCON0
LDWI B’00000000’            ; Select channel AN0
STR ADCON0
BSR ADCON0,ADCAL            ; Start ADC Self-Calibration
BTSC ADCON0,ADCAL           ; Self-Calibration done
LJUMP $-1                   ; No, test again
BSR ADCON0,ADON             ; Turn ADC On
CALL StableTime             ; ADC stable time
```

```
BSR ADCON0,GO          ; Start conversion
NOP                    ; GO/DONE ReadBack WaitTime
BTSC ADCON0,GO         ; Conversion done
LJUMP $-1              ; No, test again
BANKSEL ADRESH
LDR ADRESH,W           ; Read upper 4 bits
STR RESULTHI           ; store in SRAM space
BANKSEL ADRESL
LDR ADRESL,W           ; Read LSB
STR RESULTLO           ; Store in SRAM space
```

12. SPI INTERFACE

The SPI interface can communicate with external devices through SPI protocol, with the following characteristics:

- Full duplex, half duplex synchronous transmission
- Master mode, slave mode
- Programmable communication rate in master mode
- Programmable clock polarity and phase
- Programmable data transmission format: send LSB or MSB first
- NSS pins can be managed by hardware or software in both master and slave modes: dynamic switching of master/slave modes
- Hardware CRC check
- Support SPI interface MOSI/MISO open-drain output
- Transmit BUF is empty interrupt, receive BUF is not empty interrupt
- Working mode error interrupt, receive overflow interrupt, hardware CRC check error interrupt
- Slave mode wake-up interrupt.

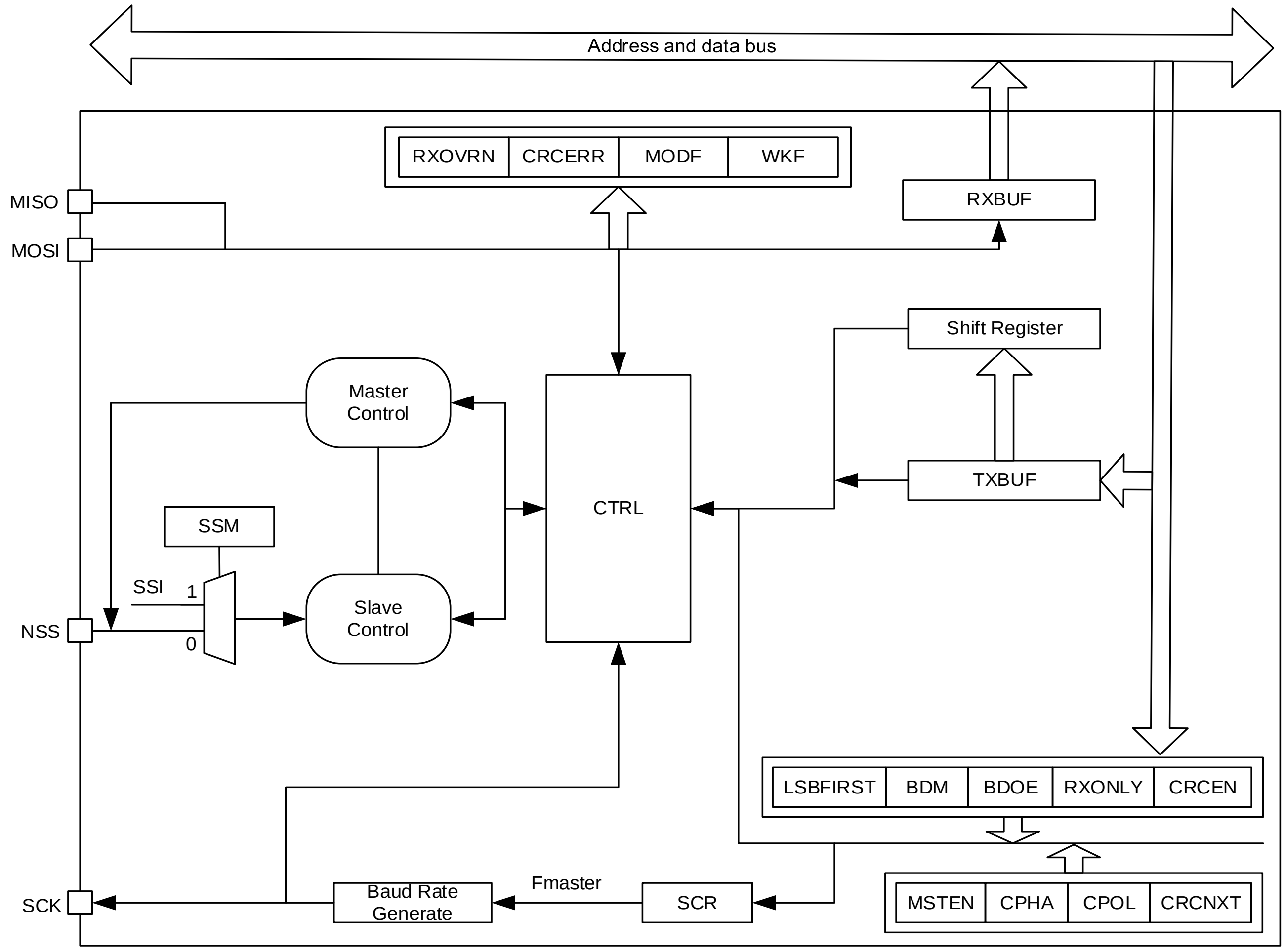


Figure 12-1 SPI Block Diagram

SPI interface has 4 pins:

Name	Function	Master mode	Slave mode
MOSI	Master output /Slave input	Data transmission	Data reception
MISO	Master input /Slave output	Data reception	Data transmission
SCK	Serial clock	Clock output	Clock input
NSS	Slave chip selection	–	Input, active at low

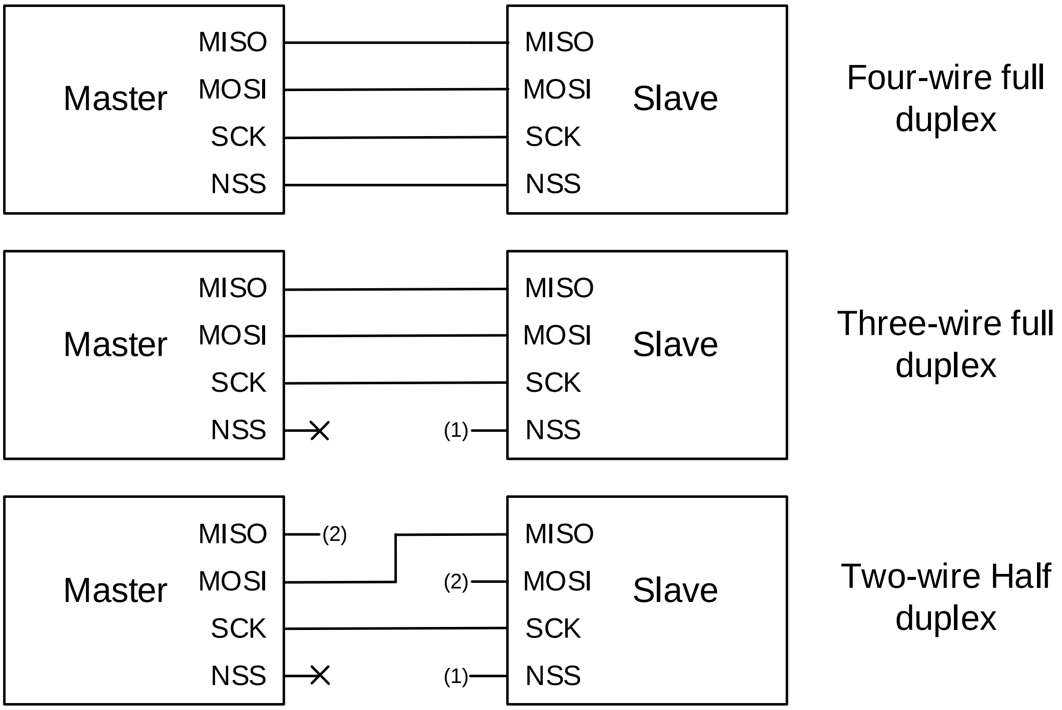
Table 12-1 Description of SPI interface

Note:

1. MOSI/MISO/SCK/NSS in this section correspond to SPI_MOSI / SPI_MISO / SPI_SCK / SPI_NSS in the pin diagram respectively.
2. Configuration of chip selecting NSS pin in slave mode:
   - The NSS pin can be configured as input, output or disabled (see "NSSM").
   - When NSS is used as input, its input value NSSVAL is the port level value (hardware) or SSI value (software, see "SSM").
   - In slave mode, when NSS is configured as input with low level, it means that the slave is selected and can start to receive or transmit data.
   - In master mode, when NSS is configured as input with low level, it will cause a working mode error (MODF is set), and the SPI module will automatically switch to slave mode at this time, which can be used for compatible multi-master communication.

The SPI interface supports full-duplex (four-wire/three-wire) and half-duplex (two-wire) synchronous data transmission . SPI communication is always initiated by the master.

In full-duplex mode, data output and input are synchronized under the same clock signal (serial clock output from the master). In half-duplex mode, the data pin of master mode is MOSI, and the data pin of slave mode is MISO.



Notes.
(1) Hardware or software management.
(2) GPIO.

Figure 12-2 Connection Diagram of SPI Interface Pin

12.1. Summary of SPI Related Register

Name	Status	Register	Addr.	Reset
DATA	Data transmit/receive BUF (TXBUF/RXBUF) Writing: write new data into TXBUF Reading: return unread data in RXBUF	SPIDATA[7:0]	0x15	RW−0000 0000
SPIF [1]	Data transmission Completion Flag 1 = Completed (latched) 0 = Not completed, or cleared	SPICTRL[7]	0x16	RW0−0
WCOL [1]	BUF write failed Flag (write in a not empty state) 1 = Fail (latched) 0 = normal	SPICTRL[6]		RW0−0
NSSM	NSS pin mode selection 00 = Disable 01 = Input (The input value NSSVAL is related to SSM, PORT level and SSI) 1x = Output (output value = NSSM[0])	SPICTRL[3:2]		RW−01

[1] Write '0' to clear, and writing '1' has no effect on the bit value.

Name	Status	Register	Addr.	Reset
SPIEN	SPI interface 1 = Enable 0 = Disable	SPICTRL[0]		RW−0
BUSY	SPI Status Bit 1 = Busy 0 = Idle	SPICFG[7]	0x17	RO−0
SBUSY		SPISTAT[4]	0x1E	RO−0
MSTEN	Operation mode 1 = Master mode (MASTER) 0 = Slave mode (SLAVE)	SPICFG[6]	0x17	RW−0
CPHA	SCK phase selection (data sampling point) 1 = 2nd clock transition edge 0 = 1st clock transition edge	SPICFG[5]		RW−0
CPOL	SCK polarity selection (SCK clock state when SPI is idle) 1 = High level 0 = Low level	SPICFG[4]		RW−0
SLAS	Slave selected Flag 1 = Yes 0 = No	SPICFG[3]		RO−0
NSSVAL	NSS input value When SSM=0, NSSVAL= NSS port level value When SSM=1, NSSVAL=SSI	SPICFG[2]		RO−1
SRMT	Internal Serial Shift Register Status 1 = Empty 0 = Not empty	SPICFG[1]		RO−1
SPICKEN	SPI Clock Module 1 = Enable 0 = Disable	PCKEN[4]	0x9A	RW−0
SYSON	In SLEEP mode, the Sysclk controls 1 = Keep active 0 = Disable	CKOCON[7]	0x95	RW−0
SCR	SCK rate setting (only valid in master mode) rate = Fmaster/(2*(SCR+1)) (SPI peripheral clock Fmaster = Sysclk)	SPISCR[7:0]	0x18	RW−0000 0000
BDM	Half duplex 1 = Enable 0 = Disable	SPICTRL2[7]	0x1D	RW−0
BDOE	Half-duplex working mode 1 = Transmit 0 = Receive	SPICTRL2[6]		RW−0

Name	Status	Register	Addr.	Reset
RXONLY	Full duplex working mode 1 = Receive only 0 = Allow transmitting and receiving	SPICTRL2[5]		RW−0
SSI	NSS software input (only valid when SSM = 1) 1 = Input is 1 0 = Input is 0	SPICTRL2[4]		RW−0
SSM	In slave mode, NSS input value management 1 = Software 0 = Hardware	SPICTRL2[3]		RW−0
CRCNXT	Transmit TXCRC to TXBUF 1 = Transmit (auto clear after completion) 0 = Don't transmit	SPICTRL2[2]		RW−0
CRCEN	Hardware CRC check module 1 = Enable 0 = Disable	SPICTRL2[1]		RW−0
LSBFIRST	Data transmission format 1 = Transmit LSB first 0 = Transmit MSB firs	SPICTRL2[0]		RW−0
CRCPOL	CRC calculation polynomial (default: 0x07)	SPICRCPOL[7:0]	0x19	RW−0000 0111
RXCRC	CRC calculation result of the received data (CRCEN changes from 0 to 1, this bit is automatically cleared)	SPIRXCRC[7:0]	0x1A	RO−0000 0000
TXCRC	CRC calculation result of the transmitted data (CRCEN changes from 0 to 1, this bit is automatically cleared)	SPITXCRC[7:0]	0x1B	RO−0000 0000

Table 12-2 SPI Related User Registers

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt	1 = Enable (PEIE, TXE, RXNE, RXERR, WAKUP apply) 0 = Global Shutdown (Wake-Up not affected)	INTCON[7]	Bank first address+0x0B	RW−0
PEIE	Peripheral Interrupt Enable	1 = Enable (TXE, RXNE, RXERR, WAKUP apply) 0 = 关闭 (no Wake-Up)	INTCON[6]		RW−0
TXE	Transmit BUF empty interrupt	1 = Enable 0 = Disable (no Wake-Up)	SPIIER[0]	0x1C	RW−0
TXBMT	Transmit BUF Status bit	1 = Empty 0 = Not empty	SPICTRL[1]	0x16	RO−1
STXBMT			SPISTAT[2]	0x1E	RO−1
RXNE	Receive BUF not empty interrupt	1 = Enable 0 = Disable (no Wake-Up)	SPIIER[1]	0x1C	RW−0
RXBMT	Receive BUF Status	1 = Empty 0 = Not empty	SPICFG[0]	0x17	RO−1
SRXBMT			SPISTAT[3]	0x1E	RO−1
RXERR	Receive error interrupt (working mode error, receive overflow, CRC check error)	1 = Enable 0 = Disable (no Wake-Up)	SPIIER[2]	0x1C	RW−0
MODF [2]	Working mode error flag	1 = Error (latched) (In the master mode, the NSS Pin is enabled and the input is low, resulting in a mode error) 0 = Normal	SPICTRL[5]	0x16	RW0−0
SMODF			SPISTAT[6]	0x1E	RO−0
RXOVRN [2]	Receive overflow flag	1 = Overflow (latch) 0 = Normal	SPICTRL[4]	0x16	RW0−0
SRXOVRN			SPISTAT[5]	0x1E	RO−0
CRCERR [2]	CRC check error flag bit	1 = Error (latched) 0 = Correct, or cleared	SPISTAT[0]	0x1E	RW0−0
WAKUP	Slave wake-up interrupt	1 = Enable 0 = Disable	SPIIER[3]	0x1C	RW−0
WKF [2]	Slave Wake-Up (data received) flag	1 = Wake up (latched) 0 = No Wake-up, or cleared	SPISTAT[1]	0x1E	RW0−0

Table 12-3 SPI Interrupt Enable and Status Bits

[2] Write '0' to clear, and writing '1' has no effect on the bit value..

Name	Status		Register	Addr.	Reset
AFP0[5]	SPI_NSS	1 = PD0 0 = PB5	AFP0[5]	0x19E	RW−0
AFP2[4]	SPI_SCK	1 = PD3 0 = PB0	AFP2[4]	0x11D	RW−0
AFP2[3]	SPI_MOSI	1 = PB7 0 = PA0	AFP2[3]		RW−0
AFP2[2]	SPI_MISO	1 = PC1 0 = PA1	AFP2[2]		RW−0
SPIOD	SPI_MISO, SPI_MOSI Open-Drain output 1 = Enable 0 = Disable		ODCON0[2]	0x21F	RW−0

Table 12-4 SPI Interface Control

Name	Function	default
I2CRMAP	Multiplexed Pin Locations • [I2C_SDA] = PA0, [I2C_SCL] = PA1 [SPI_MOSI] = PB3, [SPI_MISO] = PB2 (≥ Ver I chips optional) • [I2C_SDA] = PB3, [I2C_SCL] = PB2 [SPI_MOSI] = PA0, [SPI_MISO] = PA1 (<I Ver I chips default, cannot be changed)	[I2C_SDA] = PA0, [I2C_SCL] = PA1, [SPI_MOSI] = PB3, [SPI_MISO] = PB2

Table 12-5 SPI Interface BOOT Registers

Name	Addr	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset
PCKEN	0x9A	TKEN	I2CEN	UARTEN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
SPIDATA	0x15	DATA[7:0]								0000 0000
SPICTRL	0x16	SPIF	WCOL	MODF	RXOVRN	NSSM		TXBMT	SPIEN	0000 0110
SPICFG	0x17	BUSY	MSTEN	CPHA	CPOL	SLAS	NSSVAL	SRMT	RXBMT	0000 0000
SPISCR	0x18	SCR[7:0]								0000 0000
SPICRCPOL	0x19	CRCPOL[7:0]								0000 0111
SPIRXCRC	0x1A	RXCRC[7:0]								0000 0000
SPITXCRC	0x1B	TXCRC[7:0]								0000 0000
SPIIER	0x1C	—				WAKUP	RXERR	RXNE	TXE	---- 0000
SPICTRL2	0x1D	BDM	BDOE	RXONLY	SSI	SSM	CRCNXT	CRCEN	LSBFIRST	0000 0000
SPISTAT	0x1E	—	SMODF	SRXOVRN	SBUSY	SRXBMT	STXBMT	WKF	CRCERR	-000 1100

Table 12-6 SPI Related Register Addresses

12.2. SPI Configuration

The SPI configuration process for master and slave is basically the same:

1. Set SPICKEN = 1 to enable the SPI module clock;
2. Select master or slave mode (see "MSTEN");
3. Configure the NSS pin (see ”NSSM”, ”SSM”, ”SSI” and "NSSVAL");
4. Configure the SCK communication rate in master mode = Fmaster/(2*(SCR+1)), and the rate in slave mode is up to Fmaster/4;
5. Set the phase and polarity of SCK (see "CPOL" and "CPHA");
6. Select the data transmission format (see "LSBFIRST");
7. Set full duplex (see ”RXONLY”) or half-duplex operation (see "BDM" and "BDOE");
8. If necessary, the hardware CRC check module can be enabled (see "CRCPOL" and "CRCEN");
9. Set SPIEN = 1 to enable the SPI module;
10. If necessary, the corresponding interrupt can be enabled (see ”GIE”, ”PEIE”, ”RXERR” , ”RXNE” , ”TXE” and "WAKUP");

Note:

- SPI peripheral clock Fmaster = Sysclk;
- When the SPI module is enabled, the pin MOSI/MISO/SCK/NSS interface function is automatically enabled;
- Before the master sends the SCK clock, the SPI slave needs to be enabled first;
- When the master is acting as a transmitter, the master automatically initiates a transmission when SPI is enabled and TXBUF is not empty.
- When the master is in receive-only mode (RXONLY=1 or BDM=1& BDOE=0), after the SPI is enabled, the master automatically initiates transmission and keeps transmitting SCK;
- Before the master initiates the transmission, the data register of the slave must be written in advance with the data to be sent (in continuous communication, it is necessary to continue writing data to the data register of the slave before the end of the ongoing transmission);
- When SPIEN changes from 0 to 1, SPIF / MODF / RXOVRN / CRCERR / WKF is automatically cleared, and TXBMT / RXBMT is automatically set;

12.2.1. Communication Clock SCK

The polarity and phase of the clock SCK can be configured for 4 cases as shown in **Figure 12-3** (see” CPOL”, ” CPHA”).

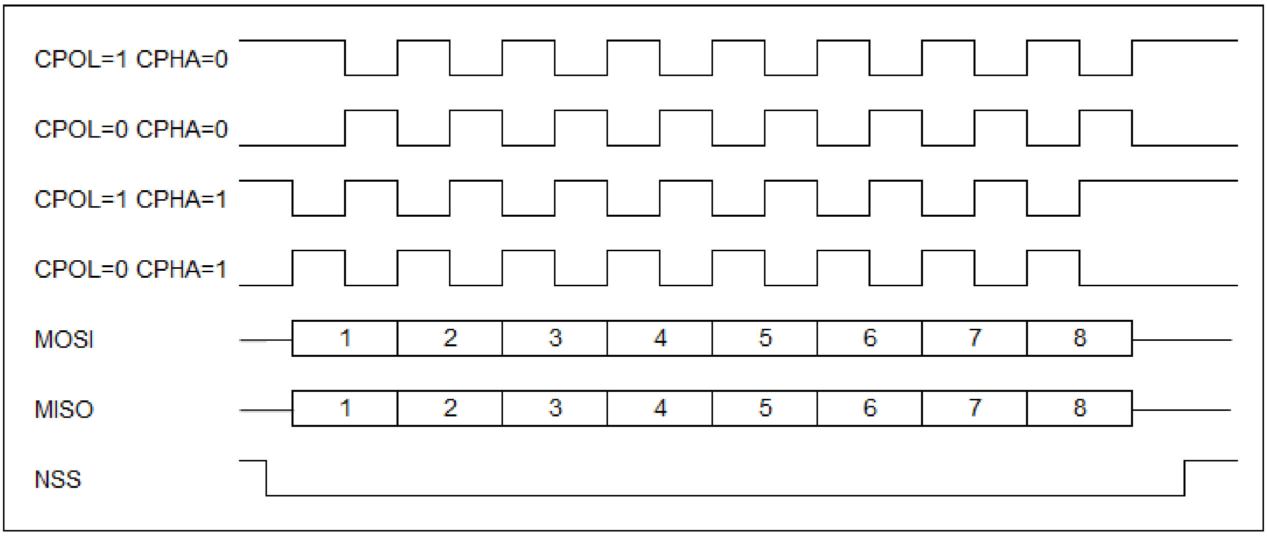


Figure 12-3 SCK Clock Polarity and Phase Timing Diagram

12.2.2. Data Processing

The data communication process is divided into blocking mode and non-blocking mode.

	Blocking mode	Non-blocking mode
Transmit data	After writing data to DATA (TXBUF), query TXBMT and write the next data when it is set to 1	When TXE = 1, TXBMT is set to 1 to enter the interrupt after writing data to DATA (TXBUF)
Receive data	Query RXBMT, the value of DATA (RXBUF) can be read when it is 0	When RXNE = 1, the interrupt is entered after RXBMT is reset to 0
	Query RXOVRN and CRCERR. When RXOVRN or CRCERR is set to 1, software needs to clear the corresponding error flag.	When RXERR = 1, enter the interrupt after RXOVRN or CRCERR is set to 1 (requires software to clear the corresponding error flag)
Remark	-	After entering the interrupt, query the corresponding status flag and process the transmitting and receiving process, and exit the interrupt after the process is completed

Table 12-7 SPI Data Processing

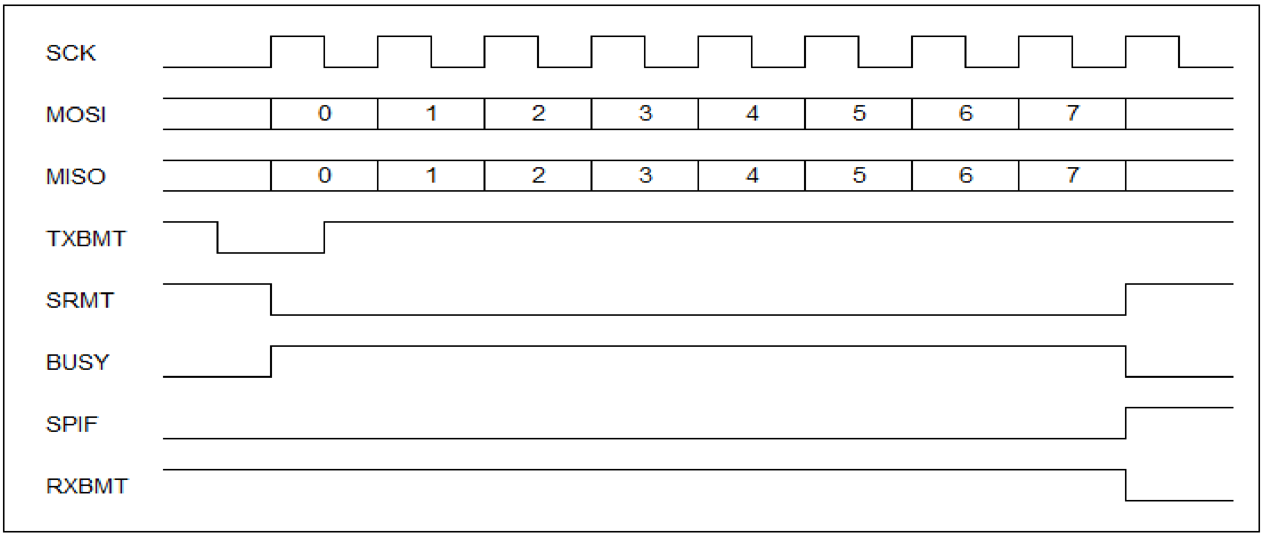


Figure 12-4 Data processing Timing Diagram (take single-byte data transmission as an example)

Taking the full-duplex communication process as an example, regardless of blocking mode or non-blocking mode, the relevant flag changes in the communication process are shown in **Figure 12-4**:

1. After writing data to the DATA (TXBUF) register, TXBMT changes from 1 to 0;
2. The data in TXBUF is transmitted to the internal shift register, SRMT changes from 1 to 0 , and SBUSY is set to 1;
3. After the data in the shift register is completely shifted out, SRMT changes from 0 to 1, and SBUSY is cleared;
4. After the current byte data transmission is completed, SPIF changes from 0 to 1, and RXBMT changes from 1 to 0 at the same time, and the value in the DATA (RXBUF) register can be read at this time;

Note: In full-duplex or half-duplex mode, the SPI module can only be turned off after all data (TXBMT=1 / RXBMT=0) are sent/received and SPI is in idle state (SBUSY=0).

12.2.3. Hardware CRC Check

The CRC check module is used to enhance the reliability of data transmission.

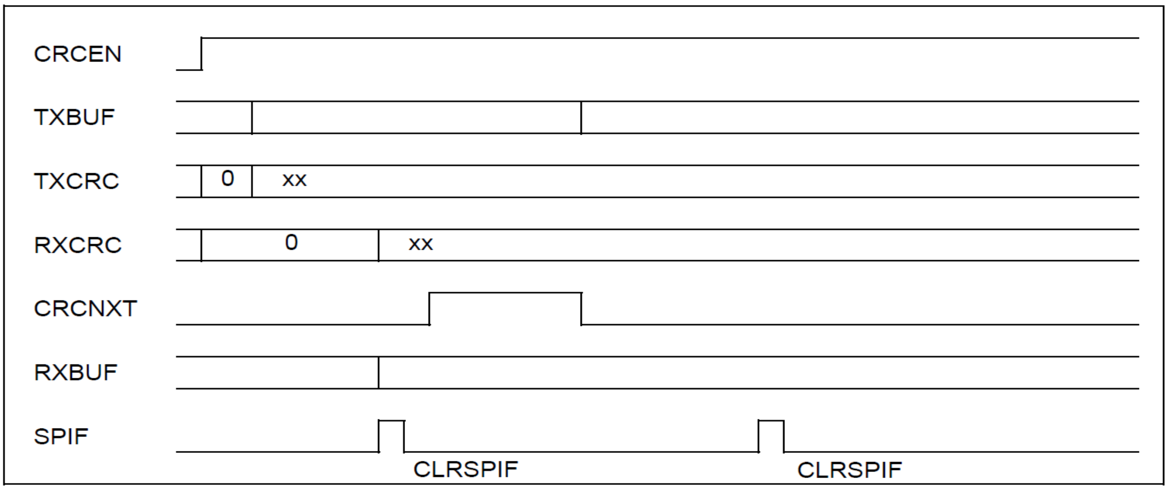


Figure 12-5 Operating Timing Diagram of CRC module

Configure CRCEN = 1 to enable the hardware CRC check module:

- Transmitter:

  1. Each time the value normally written to TXBUF is sent to the CRC module, together with the polynomial CRCPOL to generate the value of TXCRC ;
  2. When all normal data is transmitted, configure CRCNXT = 1, and the last CRC check code value will be automatically sent in the next transmission, that is, the TXCRC value will be automatically written to TXBUF (the value written to TXBUF this time will not be changed again is sent to the CRC module for calculation), the value of CRCNXT is automatically cleared;

- Receiver:

  1. Each time the value normally written to RXBUF is sent to the CRC module, together with the polynomial CRCPOL to generate the value of RXCRC ;
  2. When all normal data is received, the CRC check code value will be automatically received next time (The data received this time will not be written to RXBUF again) and compare it with the RXCRC value, if it does not match, CRCERR will be set ;

Note: When CRCEN changes from 0 to 1, the CRC module will be initialized (TXCRC and RXCRC are cleared), but the value of calculating the CRC polynomial CRCPOL (default 0x07) is not affected.

CRC check code value transmission is also divided into blocking mode and non-blocking mode:

	Blocking mode	Non-blocking mode
Send CRC check code	When the last data transmission is complete: 1. Query TXBMT, and set CRCNXT when it is set to 1; 2. Query CRCNXT, clear SPIF when it is 0; 3. Query SPIF, when it is set to 1, it means that the CRC check code is sent;	When TXE = 1, TXBMT is set to 1 and then enters the interrupt. When the last data transmission is completed, software set CRCNXT;
Receive CRC check code	Query CRCERR, when it is 1, it means that the CRC check code does not match, and software needs to clear the corresponding flag bit.	When RXERR = 1, CRCERR is set to 1 and then enters the interrupt (requires software to clear the corresponding flag bit)

Table 12-8 CRC Check-code Processing Flow

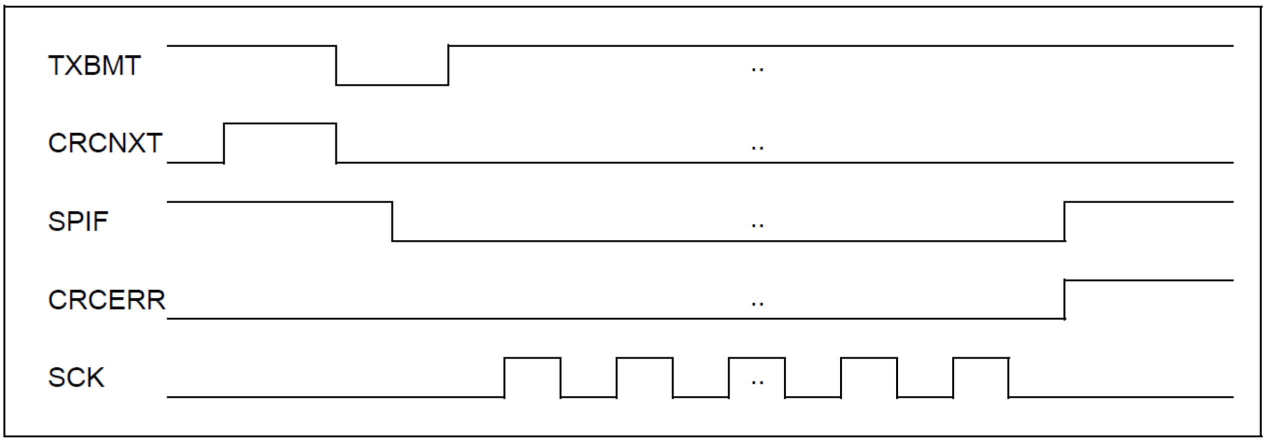


Figure 12-6 CRC module flag Timing Diagram

12.2.4. Wake-up from SLEEP in slave mode

In SLEEP mode, if SPICKEN , SYSON, WAKEUP, and PEIE are enabled at the same time, the slave can Wake-Up the MCU when it receives the first bit of data .

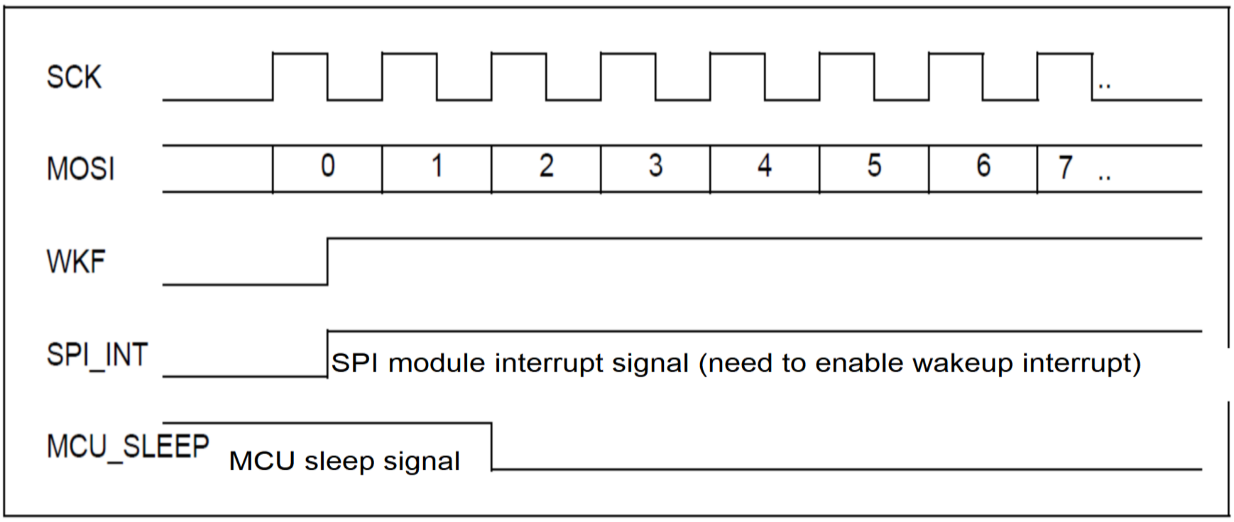

SCK
MOSI
0
1
2
3
4
5
6
7
..
WKF
SPI_INT
SPI module interrupt signal (need to enable wakeup interrupt)
MCU_SLEEP
MCU sleep signal

Figure 12-7 Wake-up from SLEEP Timing Diagram

13. I2C INTERFACE

I2C is a two-wire interface (Serial Data SDA and Serial Clock Line SCL), which can communicate with external devices through the I2C protocol. The characteristics are as follows:

- Master mode, slave mode
- Multi-master compatible
- Standard Mode (100kHz), Fast Mode (400kHz)
- 7-bit or 10-bit address format, general call (General Call)
- Data is sent/received from high bits
- Optional Clock stretching
- support the output of I2C interface SCL/SDA open-drain
- support reset by software
- Event interrupt:
  - ✓ TX-FIFO status is empty interrupt, RX-FIFO status is not empty interrupt
  - ✓ In master mode: send Start interrupt, address transmission completed interrupt, send MSB(2 bits) of10-bit address interrupt
  - ✓ In slave mode: receive address match interrupt, recognize General call interrupt, detect Stop interrupt
- Error interrupt:
  - ✓ Misplaced Start/Stop interrupt detected
  - ✓ Master Arbitration Fail Interrupt
  - ✓ NACK interrupt
  - ✓ Generate Overrun interrupt

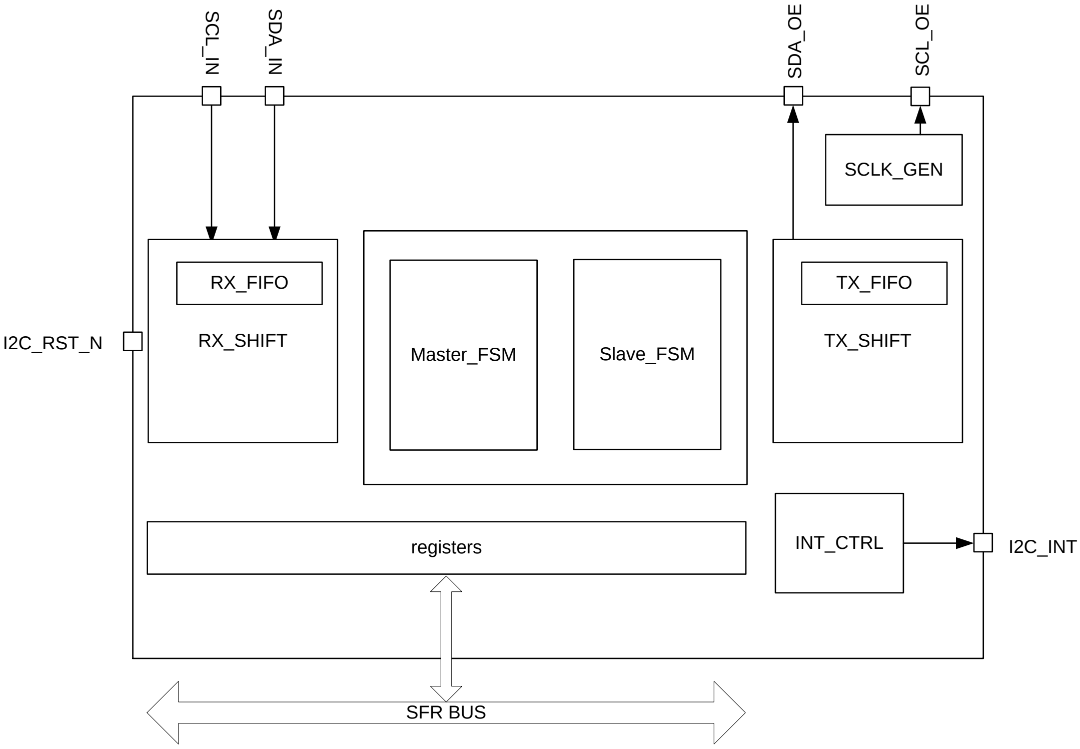


Figure 13-1 I2C Block Diagram

13.1. Summary of I2C interface related registers

Name	Status	Register	Addr.	Reset
MST10B [1]	Master send address format 1 = 10 bits 0 = 7 bits	I2CCR1[4]	0x40C	RW – 0
SLV10B [1]	Slave response address format 1 = 10 bits 0 = 7 bits	I2CCR1[3]		RW – 0
SPEED [1]	I2C communication speed 1 = Fast mode (400kHz) 0 = Standard Mode (100kHz)	I2CCR1[1]		RW – 0
MASTER [1]	Operation mode 1 = Master mode 0 = Slave mode	I2CCR1[0]		RW – 0
SOFTRST	Software reset (writable when ACTIVE = 1) 1 = Reset the I2C block 0 = Meaningless	I2CCR2[6]	0x40D	RW – 0

[1] Writable when ENABLE = 0 .

Name	Status	Register	Addr.	Reset
AGCALL [1]	General Call enabled Master mode: 1 = Send General call address (0x00) 0 = Send normal slave address Slave Mode: 1 = Response General call 0 = Do not response General calls	I2CCR2[5]		RW − 0
SNACK [1]	Receive reply 1 = Send NACK 0 = Send ACK (address match or data received)	I2CCR2[4]		RW − 0
RXHLD [1]	Stretch SCL when RX-FIFO is full 1 = Enable 0 = Disable (newly received data will be lost)	I2CCR2[1]		RW − 0
EVSTRE	After SBF/ADDF/ADD10F is set, stretch SCL 1 = Enable 0 = Disable	I2CCR3[2]	0x40E	RW − 0
ENABLE	I2C interface 1 = Enable 0 = Disable	I2CCR3[0]		RW − 0

ADD[7:0] [2]	Slave Address Low Significant Bit (LSB)
	7-bit address: ADD[6:0] is valid, ADD[7] is ignored;
	10-bit address: ADD[7:0] = LSB;
	Note: In master mode, it is the target slave address, and in slave mode, it is the local address;
		I2COARL[7:0]	0x40F	RW − 0000 0000

ADD[9:8] [2]	Slave Address Most Significant Bit (MSB)
	7-bit address: ADD[9:8] ignored;
	10-bit address: ADD[9:8] = MSB(2 bits);
	Note: In master mode, it is the target slave address, and in slave mode, it is the local address;
		I2COARH[1:0]	0x410	RW − 00

I2CEN	I2C Module Clock 1 = Enable 0 = Disable	PCKEN[6]	0x09A	RW − 0
SYSON	In sleep mode, the system clock controls 1 = Keep active 0 = Disable	CKOCON[7]	0x095	RW − 0

[2] Writable when ENABLE = 0 .

<table>
<tr><th>Name</th><th>Status</th><th>Register</th><th>Addr.</th><th>Reset</th></tr>
<tr><td>FREQ[5:0] [2]</td><td>I2C peripheral clock frequency Fmaster
000000 = Disable
000001 = 1MHz
000010 = 2MHz
...
011000 = 24MHz
> 011000 = Disable
Note: Fmaster must be the same as SysClk;</td><td>I2CFRWQ[5:0]</td><td>0x411</td><td>RW – 000000</td></tr>
<tr><td>DUTY [2]</td><td>In fast mode, SCL duty cycle
1: SCLL / SCLH = 16 / 9
0: SCLL / SCLH = 2 / 1
Note: In standard mode, SCLL / SCLH = 1 / 1;</td><td>I2CCCRH[6]</td><td>0x415</td><td>RW – 0</td></tr>
<tr><td>CCR[7:0] [2]</td><td>In master mode, the LSB of the SCL clock cycle</td><td>I2CCCRL[7:0]</td><td>0x414</td><td>RW – 00000000</td></tr>
<tr><td>CCR[11:8] [2]</td><td>In master mode, the MSB (4 bits) of the SCL clock cycle
SCL clock cycle Equation:
<table>
<tr><th>Mode</th><th>Cycle</th><th>SCLL</th><th>SCLH</th></tr>
<tr><td>Standard mode</td><td>2*CCR*Fmaster</td><td>CCR*Fmaster</td><td>CCR*Fmaster</td></tr>
<tr><td>Fast mode (DUTY=0)</td><td>3*CCR*Fmaster</td><td>2*CCR*Fmaster</td><td>CCR*Fmaster</td></tr>
<tr><td>Fast mode (DUTY=1)</td><td>25*CCR*Fmaster</td><td>16*CCR*Fmaster</td><td>9*CCR*Fmaster</td></tr>
</table></td><td>I2CCCRH[3:0]</td><td>0x415</td><td>RW – 0000</td></tr>
<tr><td>DR[7:0]</td><td>Data register
Writing: write new data into the TX-FIFO
Reading: return unread data in RX-FIFO
Note:
1. The depth of both TX-FIFO and RX-FIFO is 1;
2. When writing data, it need to write DR first, and then write I2CCMD;</td><td>I2CDR[7:0]</td><td>0x412</td><td>RW – 00000000</td></tr>
<tr><td>RESTART</td><td>Send Start, or send Restart after byte transmission
1 = Yes
0 = No</td><td>I2CCMD[2]</td><td>0x413</td><td>WO – 0</td></tr>
<tr><td>STOP</td><td>After byte transmission, send Stop
1 = Yes
0 = No</td><td>I2CCMD[1]</td><td></td><td>WO – 0</td></tr>
</table>

Name	Status	Register	Addr.	Reset
MSTDIR	Master mode, data transmission direction (read and write bit R/W) 1 = Read 0 = Send	I2CCMD[0]		WO – 0
GCALL	General call flag received in slave mode 1 = Yes (received and set after ACK) 0 = No Note: Hardware auto clear when Start/Stop or ENABLE = 0 is detected;	I2CSR3[5]	0x419	RO – 0
RDREQ	Slave mode, data transmission direction flag 1 = Transmit (set when the read/write bit of the slave receive address byte is 1) 0 = Receive Note: Hardware auto clear when Start/Stop or ENABLE = 0 is detected;	I2CSR3[2]		RO – 0
ACTIVE	Master/slave status 1 = Busy 0 = IDLE	I2CSR3[1]		RO – 0
RXHOLD	RX-FIFO full hold flag 1 = full (SCL is pulled low, released after reading DR) 0 = Not full (SCL is not pulled low)	I2CSR3[0]		RO – 0

Table 13-1 I2C Related User Registers

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt	1 = enable (For PEIE, ITBUFEN, ITEVEN, ITERREN) 0 = global shutdown (Wake-Up is not affected)	INTCON[7]	0xN0B 0xN8B 0x60B 0xF8B (N=0~5)	RW – 0
PEIE	Peripheral Interrupt Enable	1 = enable (ITBUFEN, ITEVEN, ITERREN Be applicable) 0 = off (no Wake-Up)	INTCON[6]		RW – 0
ITBUFEN	FIFO status interrupt	1 = Enable (When IICTXE = 1 or IICRXNE = 1 generate an interrupt) 0 = Disable (no Wake-Up)	I2CITR[2]	0x416	RW – 0
IICTXE [3]	TX-FIFO status	1 = Empty 0 = Not empty	I2CSR1[7]	0x417	RO – 0

[3] Auto clear when writing DR or ENABLE = 0 .

Name	Status		Register	Addr.	Reset
IICRXNE[3]	RX-FIFO status	1 = Not empty 0 = Empty	I2CSR1[6]		RO – 0
ITEVEN	Event interrupt	1 = Enable 0 = Disable (no Wake-Up) Conditions for generating event interrupt: SBF = 1 (master) ADD10F = 1 (master) ADDF = 1 (Master/Slave) STOPF = 1 (slave)	I2CITR[1]	0x416	RW – 0
STOPF[4]	Slave detect Stop flag	1 = Detected (set after ACK) 0 = Not detected	I2CSR1[4]	0x417	RO – 0
ADD10F[4]	Master send MSB address flag	1 = Yes (set after ACK) 0 = No	I2CSR1[3]		RO – 0
ADDF[4]	Master Transmit LSB Address/Slave Receive Address Match Flag	Master send address LSB: 1 = Completed (set after ACK) 0 = Not sent or mismatch Slave receiving address: 1 = General Call matched or recognized 0 = Mismatch Note: ADDF will not be set after NACK	I2CSR1[1]		RO – 0
SBF[4]	Master send Start flag	1 = Yes 0 = No	I2CSR1[0]		RO – 0
ITERREN	Error interrupt	1 = Enable 0 = Disable (no Wake-Up) Error interrupt generation conditions: OVR = 1 AF = 1 ARLO = 1 BERR = 1	I2CITR[0]	0x416	RW-0
TXARBT[5]	Transmission abort flag (caused by an error or abnormal cause during the sending process)	1 = Abort occurred 0 = No abort occurred	I2CSR2[4]	0x418	RW0 – 0
OVR[5]	Overrun Flag	1 = Yes 0 = No	I2CSR2[3]		RW0-0

[4] Automatically cleared by hardware when I2CSR1 is read or ENABLE = 0 .

[5] Write 0 to clear, or automatically cleared by hardware when ENABLE = 0 .

Name	Status		Register	Addr.	Reset

		Conditions for Overrun :
		TX-over: still write DR when the TX-FIFO is not empty;
		RX-over: still receive data when the RX-FIFO is not empty;
		RX-under: read when the RX-FIFO is empty;			

AF [5]	ACK status	1 = NACK 0 = ACK	I2CSR2[2]		RW0−0
ARLO [5]	Master Arbitration Fail Flag	1 = Failed to generate arbitration 0 = No arbitration failure	I2CSR2[1]		RW0−0
BERR [5]	Bus error status (Start/Stop detected misalignment)	1 = Detected (set when a Start/Stop is detected during the byte transmission phase) 0 = Not detected	I2CSR2[0]		RW0−0

Table 13-2 I2C Interrupt Enable and Status Bits

Name	Status	Register	Addr.	Reset
AFP0[0]	I2C_SDA pin 1 = PB6 0 = PB3	AFP0[0]	0x19E	RW−0
AFP1[4]	I2C_SCL pin 1 = PA2 0 = PB2	AFP1[4]	0x19F	RW−0
I2COD	I2C_SCL, I2C_SDA open-drain output 1 = Enable 0 = Disable	ODCON0[1]	0x21F	RW−0

Table 13-3 I2C Interface Pin Control

Name	Function	Defaults
I2CRMAP	Multiplexed pin locations • [I2C_SDA] = PA0, [I2C_SCL] = PA1 [SPI_MOSI] = PB3, [SPI_MISO] = PB2 (≥ VerI chip optional) • [I2C_SDA] = PB3, [I2C_SCL] = PB2 [SPI_MOSI] = PA0, [SPI_MISO] = PA1 (< VerI chip default, cannot be changed)	• [I2C_SDA] = PA0, [I2C_SCL] = PA1, [SPI_MOSI] = PB3, [SPI_MISO] = PB2

Table 13-4 I2C interface BOOT

Name	Addr	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset
PCKEN	0x9A	TKEN	I2CEN	UARTEN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
I2CCR1	0x40C	—	—	—	MST10B	SLV10B	—	SPEED	MASTER	---0 0-00
I2CCR2	0x40D	—	SOFTRST	AGCALL	SNACK	—	—	RXHLD	—	-000 —0-
I2CCR3	0x40E	—					EVSTRE	—	ENABLE	---- -000
I2COARL	0x40F	ADD[7:0]								0000 0000
I2COARH	0x410	—	—	—	—	—	—	ADD[9:8]		---- --00
I2CFREQ	0x411	—	—	FREQ[5:0]						--00 0000
I2CDR	0x412	DR[7:0]								0000 0000
I2CCMD	0x413	—	—	—	—	—	RESTART	STOP	MSTDIR	---- -000
I2CCCRL	0x414	CCR[7:0]								0000 0000
I2CCCRH	0x415	—	DUTY	—	—	CCR[11:8]				-0—0000
I2CITR	0x416	—					ITBUFEN	ITEVEN	ITERREN	---- -000
I2CSR1	0x417	IICTXE	IICRXNE	—	STOPF	ADD10F	—	ADDF	SBF	00-0 0-00
I2CSR2	0x418	—	—	—	TXABRT	OVR	AF	ARLO	BERR	---0 0000
I2CSR3	0x419	—	—	GCALL	—	—	RDREQ	ACTIVE	RXHOLD	--0- -000

Table 13-5 I2C Related Register Address

13.2. I2C Configuration

The I2C configuration process for master and slave is basically the same:

1. Set I2CEN = 1 to enable the I2C module clock;
2. Select master or slave mode (see "MASTER");
3. Set the master-slave clock frequency Fmaster, which must be the same as SysClk (see "FREQ[5:0]");
4. The communication rate of the master selects standard mode or fast mode (see "SPEED");
5. The master configures the SCL duty cycle and clock cycle (see "DUTY", "CCR[7:0]" and "CCR[11:8]");
6. Master and Slave select 7-bit or 10-bit address format (see "MST10B" and "SLV10B");
7. Set the master's data transmission direction to send or receive (see "MSTDIR"), the slave is controlled by the R/W bit of the received address byte;
8. If necessary, select General call mode (see "AGCALL");
9. Set ENABLE = 1 to enable the I2C module;
10. If necessary, the corresponding interrupt can be enabled (see ”GIE”, ”PEIE”, ”ITBUFEN” , ”ITEVEN” and "ITERREN");

Note:

- When ENABLE = 1, the pin SCL/SDA interface function is automatically enabled, and SCL/SDA corresponds to I2C_SCL/I2C_SDA in the pin diagram respectively;
- In order to generate the correct timing, the input clock Fmaster and clock cycle CCR of the I2C

module must meet the following setting conditions:

	Register	Standard mode	Fast mode (DUTY=0)	Fast mode (DUTY=1)
Master and slave	FREQ[5:0]	≥ 2MHz	≥ 8MHz	≥ 8MHz
Master	CCR[11:0]	≥ 9	≥ 9	—

- If the I2C module has been in an active state (ACTIVE=1) due to abnormal reasons, the SOFTRST can be set to reset the sending and receiving modules , which has no effect on the register value ;

In I2C communication, the master generates a clock signal and initiates data transmission, and the master controls the Start and Stop signals. Serial data transmission begins with a Start and ends with a Stop. During the ninth clock cycle after a byte (8 bit) is transmitted, the receiver needs to send back an acknowledge bit (ACK) to the transmitter .

After the slave detects the Start , it can identify its own address (programmable, 7-bit or 10-bit) and General

Call address, and has the function of Stop detection.

The four working modes of the I2C module are: master transmitting, master receiving, slave transmitting, and slave receiving.

13.2.1. Master Transmitting Mode

When MST10B = 0 (7 -bit address format) : The first byte sent by the master includes a 7-bit address and a R/W bit (0), and then starts to send 8-bit serial data.

When MST10B = 1 (10 -bit address format) : The first byte sent by the master includes the address header sequence (11110 + high-order 2-bit address) and R/W bit (0), the second byte is the low-order 8 -bit address, and then start to send 8-bit serial data.

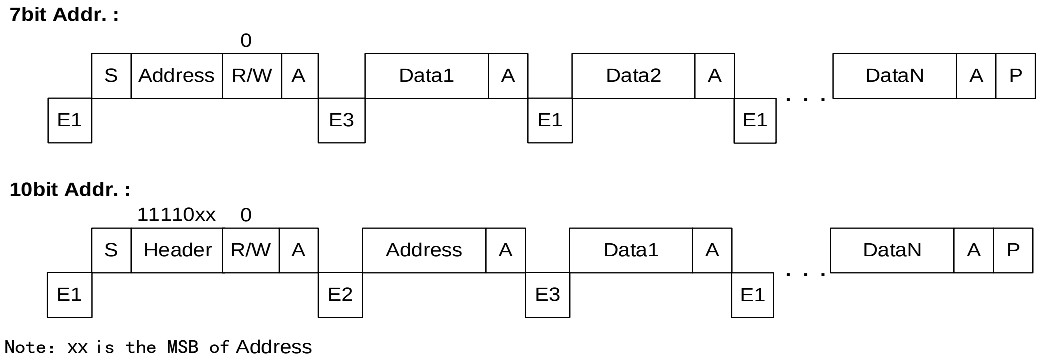


Figure 13-2 Master Transmitting Process

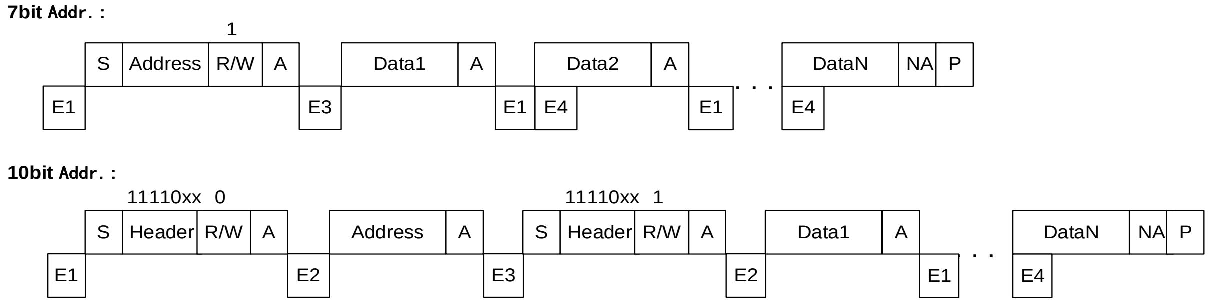
Note:

- S = Start signal , A = ACK signal , P = Stop signal;
- E1: IICTXE = 1, TX-FIFO is empty (Writing DR and I2CCMD will clear this flag);
- E2: ADD10F = 1 (Reading I2CSR1 will clear this flag);
- E3: ADDF = 1 (Reading I2CSR1 will clear this flag);

13.2.2. Master Receiving mode

MST10B = 0 (7 -bit address format) : The first byte sent by the master includes a 7-bit address and a R/W bit (1), and then starts to receive 8-bit serial data.

MST10B = 1 (10 -bit address format) : The first byte sent by the master includes the address header sequence (11110 + MSB of 2-bit address) and R/W bit (0), the second byte is the LSB (8 -bit) of address, and then re-send the Start signal along with the address header sequence and R/W bit (1) to start receiving 8-bit serial data.



Note：xx is the MSB of Address

Figure 13-3 Master receiving process

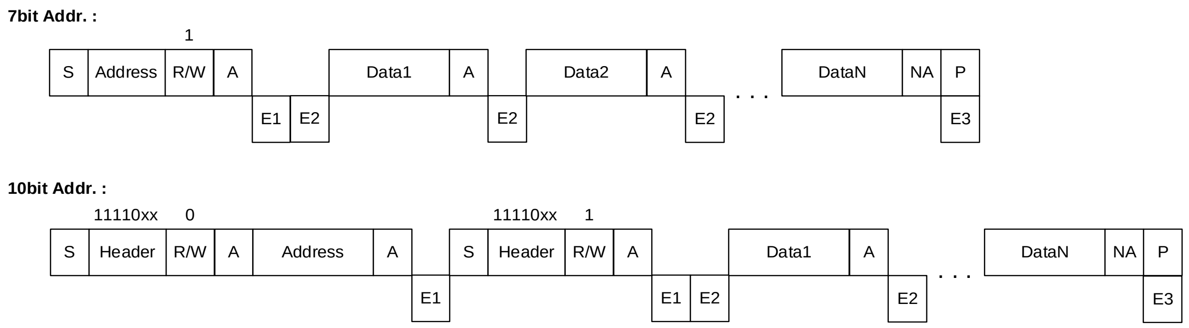
Note:

- S = Start signal , A = ACK signal , P = Stop signal;
- E1: IICTXE = 1, TX-FIFO is empty (Writing DR and I2CCMD will clear this flag);
- E2: ADD10F = 1 (Reading I2CSR1 will clear this flag);
- E3: ADDF = 1 (Reading I2CSR1 will clear this flag);
- E4: IICRXNE = 1, RX-FIFO not empty (Reading DR will clear this flag);

13.2.3. Slave Transmitting Mode

SLV10B = 0 (7 -bit address format) : The first byte received by the slave includes a 7-bit address and a R/W bit (1), and then starts to send 8-bit serial data.

SLV10B = 1 (10 -bit address format) : The first byte received by the slave includes the address header sequence (11110 + MSB of 2-bit address) and R/W bit (0), the second byte is the LSB of 8 -bit address, and then re-detect the Start signal and receive the address header sequence and R/W bit (1), and starts to send 8-bit serial data.



Note：xx is the MSB of Address

Figure 13-4 Slave transmitting process

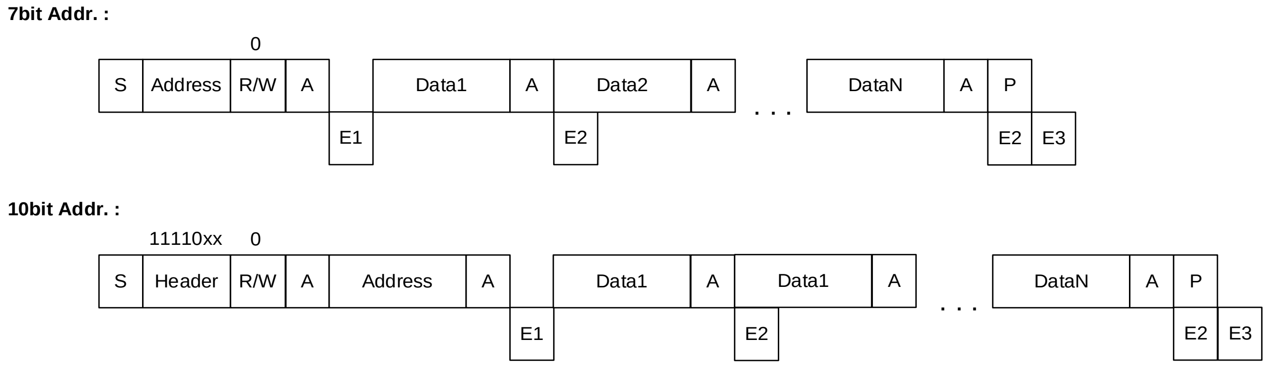
Note:

- S = Start signal , A = ACK signal , P = Stop signal ;
- E1: ADDF = 1, pull the SCL line low (Reading I2CSR1 will clear this flag);
- E2: IICTXE = 1, TX-FIFO is empty, pull down the SCL line, read RDREQ to 1 (write DR and I2CCMD will clear this flag)
- E3: AF=1 (write 0 to clear);

13.2.4. Slave Receiving Mode

SLV10B = 0 (7 -bit address format) : The first byte received by the slave includes the address and the R/W bit (0), and then starts to receive 8-bit serial data.

SLV10B = 1 (10 -bit address format) : The first byte received by the slave includes the address header sequence (11110 + MSB of 2-bit address) and R/W bit (0), the second byte is the LSB of 8 -bit address, and then starts to receive 8-bit serial data.



Note：xx is the MSB of Address

Figure 13-5 Slave receiving process

Note:

- S = Start signal , A = ACK signal , P = Stop signal ;
- E1: ADDF = 1 (Reading I2CSR1 will clear this flag);

- E2: IICRXNE = 1, RX-FIFO not empty (Reading DR will clear this flag);
- E3: STOPF = 1 (Reading I2CSR1 will clear this flag);

13.2.5. General Call

The master/slave sets AGCALL to enable General Call mode:

- The master sends data to the 0x00 address, and the communication process is the same as the master transmission;
- The slave responds to the General Call sent by the master and writes data to the 0x00 address, and the communication process is the same as the receiving process of slave.

14. USART INTERFACE

The Universal Synchronous/Asynchronous Transceiver USART can communicate with peripherals in the industry standard NRZ serial data format. Features are as follows:

- Full-duplex, single-wire half-duplex asynchronous mode
- Full Duplex Synchronous Mode
  - ✓ Synchronized Clock Output : Programmable Clock Polarity and Phase
- Infrared 1.0 mode
  - ✓ 8-bit prescaled baud rate generator
  - ✓ Low power mode
- Smart card mode
  - ✓ 8-bit prescaled baud rate generator
  - ✓ STOP: 1.5 bits
  - ✓ Programmable guard time
- LIN master mode
  - ✓ Support the transmission and detection of disconnected frames
- Multi-chip communication mode
  - ✓ The mute mode can be woken up by address matching or IDLE frame, and it starts to receive data after waking up
- Data transmission length: 7, 8 or 9 bits
- Parity bit
- STOP: 1 or 2 bits
- 16-bit programmable baud rate generator up to 1 Mbit/s
- Data is sent/received from low bits
- Transmitter and receiver can be enabled independently
- Automatic baud rate detection
- Support USART interface TX open-drain output
- Send BUF is empty interrupt, receive BUF is not empty interrupt
- Send completion interrupt
- Idle frame interrupt
- Receive status interrupt: frame break, frame error, parity error, or receive overflow

Note: The default is asynchronous full-duplex mode. When the operating mode is selected, please disable other modes.

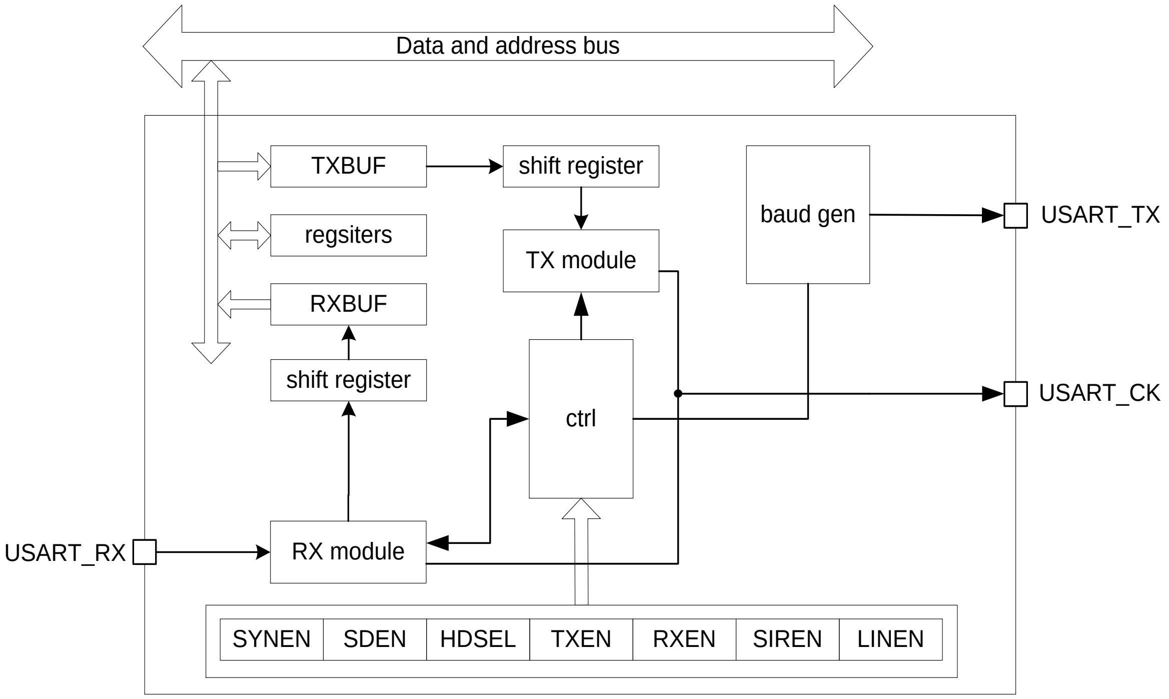


Figure 14-1 USART Block Diagram

The USART serial module has 3 pins:

USART_RX: Serial data input.

USART_TX: Serial data output. In single-wire half-duplex mode, the TX pin is used for both data input and data output (need to be configured in open-drain mode).

USART_CK : used as synchronous clock output in synchronous mode, and used as system clock frequency division output in smart card mode.

Note:

- When the transmitter is enabled, but not transmitting data, the TX pin is high.
- When the transmitter is enabled and data is being transmitted, the TX pin is low during the START and high during the STOP.

14.1. Summary of USART Interface Related Registers

Name	Status		Register	Addr.	Reset
UARTEN	USART module clock	1 = Enable 0 = Disable	PCKEN[6]	0x9A	RW−0
SYSON	In SLEEP mode, the system clock controls	1 = Keep active 0 = Disable	CKOCON[7]	0x95	RW−0
DATAL	Data transmission/reception BUF LSB (inappropriate bit manipulation)		URDATAL[7:0]	0x48C	RW−0000 0000

Name	Status	Register	Addr.	Reset
DATAH	Send/receive MSB of 1 bit BUF(when EXTEN=1) Note: DATAL needs to be written first, then DATAH;	URDATAH[0]	0x48D	RW−0
BKREQ	Send break frame 1 = Enable, or in progress 0 = Disable, or transmission completed Note: 1. Please set the length of the disconnected frame before sending the disconnected frame; 2. This bit auto clear after the transmission is completed, and it is forbidden to write 0 to it during the transmission process;	URLCR[6]	0x48F	RW−0
EVEN	Odd/even parity 1 = Even parity 0 = Odd parity	URLCR[4]		RW−0
PEN	Parity Enable 1 = Enable 0 = Disable	URLCR[3]		RW−0
URSTOP	STOP length 1 = 1.5 bit (smart card mode) or 2 bits 0 = 1 bit	URLCR[2]		RW−0
LTH	Communication data length control (excluding parity bit) 1 = 8 bits 0 = 7 bits	URLCR[0]		RW−0
RWU	In multiprocessor mode, enter mute mode 1 = Enable 0 = Disable, or exit	URLCREXT[1]	0x490	RW−0
EXTEN	Communication data length total controls (excluding parity bits) 1 = 9 bits 0 = 7 bits or 8 bits (depending on LTH)	URLCREXT[0]		RW−0
SIRLP	Infrared low power mode 1 = Enable 0 = Disable	URMCR[5]	0x491	RW−0
TXEN	Serial transmitter 1 = Enable (pin TX function is automatically enabled) 0 = Disable	URMCR[4]		RW−0
RXEN	Serial receiver 1 = Enable (pin RX function is automatically enabled) 0 = Disable	URMCR[3]		RW−0

Name	Status		Register	Addr.	Reset
WAKE	Mute mode wake-up method	1 = Address matches 0 = IDLE frame	URMCR[2]		RW−0
HDSEL	Half duplex	1 = Enable 0 = Disable	URMCR[1]		RW−0
SIREN	Infrared mode	1 = Enable 0 = Disable	URMCR[0]		RW−0
RAR	Local address in multiprocessor mode[3:0]		URRAR[3:0]	0x493	RW−0000
DLL	Baud rate divider counter LSB and MSB baud rate = Fmaster / (16 * {DLH, DLL}) Note: Fmaster = Sysclk ; the minimum value of {DLH, DLL} is 0x0001 , when it is 0x0000, USART does not work;		URDLL[7:0]	0x494	RW−0000 0000
DLH			URDLH[7:0]	0x495	RW−0000 0000
ABRE	Baud rate detect overflow Flag	1 = Overflow 0 = Normal	URABCR[3]	0x496	RW−0
ABRM	Baud rate detection mode 1 = Detection length is [(START + 1st bit data) / 2] (The 1st bit of the data must be 1, the 2nd bit must be 0) 0 = Only the START length is detected (the 1st bit data must be 1)		URABCR[2]		RW−0
ABRF	Baud rate flag detected 1 = Detected 0 = Not detected Note: Write 0 to clear, after the bit is cleared, it will enter the baud rate detection again immediately. In order to ensure the bit detected is the START, it is recommended to clear this bit after RXNEF is set;		URABCR[1]		RW−0
ABREN	Automatic baud rate detection	1 = Enable 0 = Disable	URABCR[0]		RW−0
LBCL	In synchronous mode, the clock output corresponding to the last 1bit data (MSB) is sent 1 = Enable 0 = Disable		URSYNCR[3]	0x497	RW−0
URCPHA	Synchronous mode clock phase (data sampling point) 1 = 2nd clock transition edge 0 = 1st clock transition edge		URSYNCR[2]		RW−0

<table>
<tr><th>Name</th><th colspan="2">Status</th><th>Register</th><th>Addr.</th><th>Reset</th></tr>
<tr><td>URCPOL</td><td colspan="2">Synchronous mode clock polarity (state of SCK when the bus is idle)
1 = High level
0 = Low level</td><td>URSYNCR[1]</td><td rowspan="2"></td><td>RW−0</td></tr>
<tr><td>SYNEN</td><td colspan="2">Synchronous mode
1 = Enable (pin CK automatically outputs a synchronous clock)
0 = Disable</td><td>URSYNCR[0]</td><td>RW−0</td></tr>
<tr><td>LINEN</td><td>LIN Master mode</td><td>1 = Enable
0 = Disable</td><td>URLINCR[4]</td><td rowspan="2">0x498</td><td>RW−0</td></tr>
<tr><td>BLTH</td><td colspan="2">Break frame length (bit)
Note: BLTH>0 is valid, it is recommended to set it to 12bit or 13bit, if it is too short, the received frame will be misjudged as a normal frame;</td><td>URLINCR[3:0]</td><td>RW−0000</td></tr>
<tr><td>NACK</td><td colspan="2">Smart card mode, reply NACK when parity error is detected
1 = Send NACK
0 = Do not send NACK</td><td>URSDCR0[6]</td><td rowspan="3">0x499</td><td>RW−0</td></tr>
<tr><td>CKOE</td><td colspan="2">Smart Card Clock Source
1 = Enable (need to configure the PSC register to an active value)
0 = Disable</td><td>URSDCR0[5]</td><td>RW−0</td></tr>
<tr><td>SDEN</td><td>Smart Card Mode</td><td>1 = Enable (STOP must be 1.5bit)
0 = Disable</td><td>URSDCR0[4]</td><td>RW−0</td></tr>
<tr><td>GT</td><td colspan="2">Smartcard mode, guard time (baud clock interval between two characters)
Note: The minimum value is 1 (0 is invalid), after the protection time expires, the transmission completion flag TCF is set;</td><td>URSDCR1[7:0]</td><td>0x49A</td><td>RW−0000 0000</td></tr>
<tr><td>PSC</td><td colspan="2">Divide the system clock to provide a clock for smart card or infrared low power consumption<table>
<tr><th></th><th>Smart Card Clock Source</th><th>Infrared Low Power Clock Source</th></tr>
<tr><td>0</td><td>inactive</td><td>inactive</td></tr>
<tr><td>1</td><td>divide by 2</td><td>divide by 1</td></tr>
<tr><td>2</td><td>divide by 3</td><td>divide by 2</td></tr>
</table></td><td>URSDCR2[7:0]</td><td>0x49B</td><td>RW−0000 0000</td></tr>
</table>

Name	Status			Register	Addr.	Reset
	3	divide by 4	divide by 3			
	…	…	…			
	255	256 frequency division	255 frequency division			

Table 14-1 USART related User Registers

Name	Status		Register	Addr.	Reset
AFP0[7]	USART_CK	1 = PD1 0 = PA5	AFP0[7]	0x19E	RW−0
AFP2[1]	USART_RX	1 = PA2 0 = PA7	AFP2[1]	0x11D	RW−0
AFP2[0]	USART_TX	1 = PB6 0 = PA6	AFP2[0]		RW−0
UROD	USART_TX_Open-Drain Output	1 = Enable 0 = Disable	ODCON0[0]	0x21F	RW−0

Table 14-2 USART Interface Pin Control

Name	Status		Register	Addr.	Reset
GIE	Global Interrupt	1 = Enable (PEIE, URTE, URRXNE, TCEN, IDELE, RXSE apply) 0 = Global Shutdown (Wake-Up is not affected)	INTCON[7]	Bank first address + 0x0B	RW−0
PEIE	Peripheral Interrupt Enable	1 = Enable (URTE, URRXNE, TCEN, IDELE, RXSE apply) 0 = Disable (no Wake-Up)	INTCON[6]		RW−0
URTE	Transmit BUF is empty interrupt	1 = Enable 0 = Disable	URIER[1]	0x48E	RW−0
TXEF	Transmit BUF status	1 = Empty 0 = Not empty Note: write DATAL(8bit) / DATAH(9bit) to clear;	URLSR[5]	0x492	RO−1
URRXNE	Receive BUF as not empty interrupt	1 = Enable 0 = Disable	URIER[0]	0x48E	RW−0
RXNEF	Receive BUF status	1 = Not empty	URLSR[0]	0x492	RO−0

Name	Status		Register	Addr.	Reset
		0 = Empty, or cleared Note: read DATAL(8bit) / DATAH(9bit) to clear;			
TCEN	Transmit Completion interrupt	1 = Enable 0 = Disable	URIER[5]	0x48E	RW−0
TCF	Transmission Completion Flag	1 = Yes 0 = No Note: write 1 to clear, or write DATAL(8bit) Cleared after /DATAH(9bit);	URTC[0]	0x49C	R_W1C−1
IDELE	Idle frame interrupt	1 = Enable 0 = Disable	URIER[3]	0x48E	RW−0
IDLEF [1]	Idle frame flag detected	1 = Detected 0 = Not detected	URLSR[6]	0x492	RW0−0
RXSE	Receive error interrupt	1 = Enable 0 = Disable Note: Conditions for receiving error interrupt are as follows BKF = 1 FEF = 1 PEF = 1 OVERF = 1	URIER[2]	0x48E	RW−0
BKF [1]	Received break frame Flag	1 = Received 0 = Not received, or cleared	URLSR[4]	0x492	RW0−0
FEF [1]	Received frame error Flag	1 = Yes 0 = No, or cleared	URLSR[3]	0x492	RW0−0
PEF [1]	Parity received Error Flag	1 = Yes 0 = No, or cleared	URLSR[2]	0x492	RW0−0
OVERF [1]	Receive BUF overflow error Flag	1 = Overflow 0 = Normal, or cleared	URLSR[1]	0x492	RW0−0
WAKE	Mute mode wake-up mode selection	1 = Address match 0 = IDLE frame	URMCR[2]	0x491	RW−0
ADDRF	Mute mode address match Flag	1 = Match 0 = Mismatch	URLSR[7]	0x492	RO−0

Table 14-3 USART Interrupt Enable and Status Bits

[1] Write '0' to clear, and writing '1' has no effect on the bit value.

Name	Addr	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset
PCKEN	0x9A	TKEN	I2CEN	UARTEN	SPIEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
CKOCON	0x95	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
URDATAL	0x48C	DATA[7:0]								0000 0000
URDATAH	0x48D	—							DATAH	---- ---0
URIER	0x48E	—		TCEN	—	IDELE	RXSE	URTE	URRXNE	--0- 0000
URLCR	0x48F	—	BKREQ	—	EVEN	PEN	URSTOP	—	LTH	-0-0 00-0
URLCREXT	0x490	—						RWU	EXTEN	---- --00
URMCR	0x491	—		SIRLP	TXEN	RXEN	WAKE	HDSEL	SIREN	---0 0000
URLSR	0x492	ADDRF	IDLEF	TXEF	BKF	FEF	PEF	OVERF	RXNEF	0010 0000
URRAR	0x493	—				RAR[3:0]				---- 0000
URDLL	0x494	DLL[7:0]								0000 0000
URDLH	0x495	DLH[7:0]								0000 0000
URABCR	0x496	—				ABRE	ABRM	ABRF	ABREN	---- 0000
URSYNCR	0x497	—				LBCL	URCPHA	URCPOL	SYNEN	---- 0000
URLINCR	0x498	—			LINEN	BLTH[3:0]				---0 0000
URSDCR0	0x499	—	NACK	CKOE	SDEN	—				-000 ----
URSDCR1	0x49A	GT[7:0]								0000 0000
URSDCR2	0x49B	PSC[7:0]								0000 0000
URTC	0x49C	—							TCF	---- ---1

Table 14-4 USART Related Register Addresses

14.2. USART function

14.2.1. Asynchronous Operating Mode

Full-duplex and half-duplex configuration process :

1. Set UARTEN = 1 to enable the USART module clock;
2. Set the communication baud rate = Fmaster / (16 * {DLH, DLL}) (see "DLH", "DLL");
3. Set the communication data length to 7, 8 or 9 bits (see "EXTEN", "LTH") ;
4. Set parity bit (see "PEN", "EVEN") ;
5. Set the length of STOP to 1 or 2 bits (see " U RSTOP") ;
6. Choose between full duplex (default) or half duplex mode of operation (see " HDSEL ");
7. If necessary, enable the corresponding interrupt (see "GIE", "PEIE", "URTE", "URRXNE", "TCEN" and "RXSE" etc.);
8. Set TXEN = 1 or RXEN = 1 to enable transmit or receive function as required;

Note:

- USART peripheral clock Fmaster = Sysclk;
- In half-duplex mode, if both transmitting and receiving functions are enabled, the transmitted data will also be received by the machine;

The data communication format of the asynchronous mode is to send the LSB first, and then send the MSB. The data frame format comparison with or without parity bit is as follows:

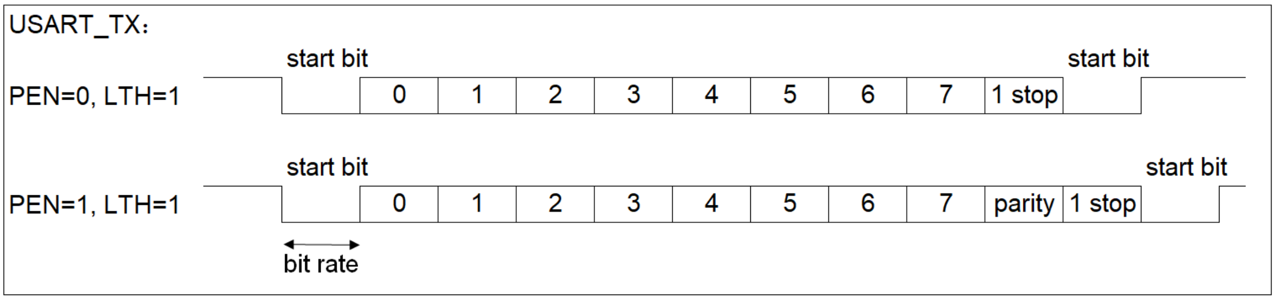


Figure 14-2 Communication Format in Asynchronous Mode (take 8bit length as an example)

The data processing flow includes blocking mode and non-blocking mode:

	Blocking mode	Non-blocking mode
Transmit data	After writing data to DATAL/H (TXBUF), query TXEF or TCF, write the next data when TXEF or TCF is set to 1	When URTE = 1 or TCEN = 1, TXEF or TCF is set to 1 to enter the interrupt after writing data to DATAL/H (TXBUF)
Receive data	Query RXNEF, the value of DATAL/H (RXBUF) can be read when RXNEF is set to 1	When URRXNE = 1, enter the interrupt after RXNEF is set to 1; In addition, it is recommended to enable the RXSE interrupt, and enter the corresponding interrupt for processing when receiving errors;
Remark	-	After entering the interrupt, query the corresponding status flag and handle the transmitting and receiving process, and exit the interrupt after the processing is completed

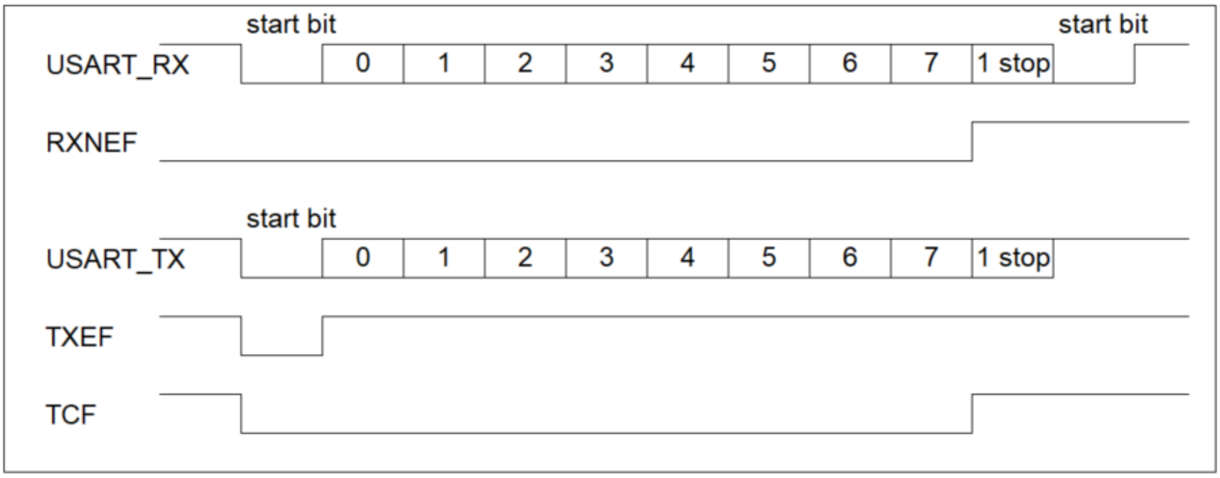


Figure 14-3 Asynchronous Mode Flag Timing Diagram

14.2.2. Synchronous Operating Mode

Synchronous mode is used to simulate the communication function in SPI master mode . When SYNEN = 1 , the USART_CK pin will output a clock synchronized with the data. Data output sends the LSB first, followed by the MSB.

In addition, the polarity and phase of the synchronization clock can be selected (see "CPOL" , "CPHA") . During the START and STOP, there is no clock pulse on the USART_CK pin . Whether to output the synchronous clock when the last 1bit of data is sent is determined by LBCL .

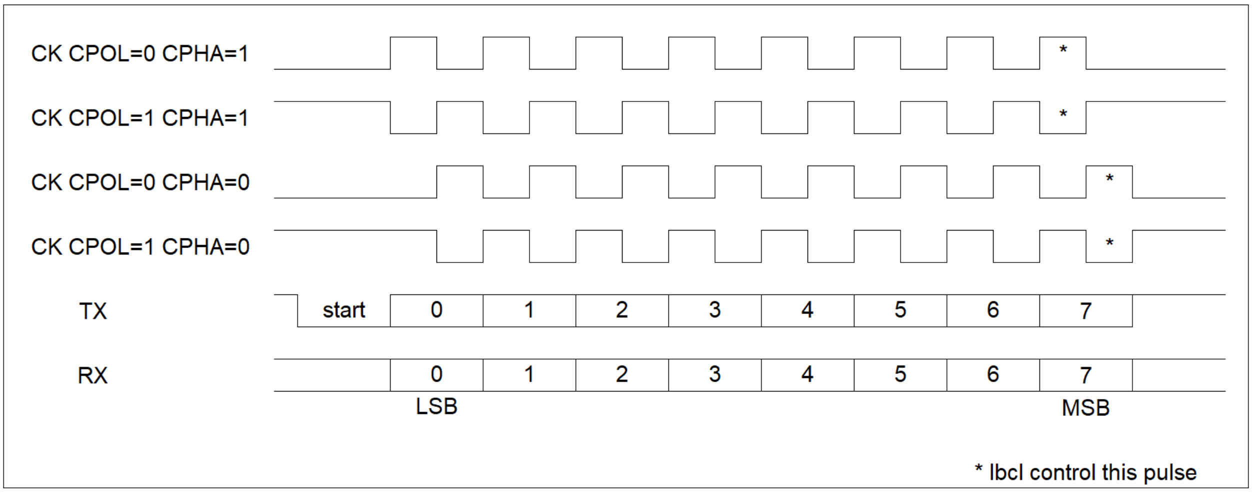


Figure 14-4 Sync Mode Communication Format (8bit Data Length)

Note :

- The sync clock rate is set the same as the baud rate, i.e. Fmaster / (16 * {DLH, DLL});
- When TXEN = 0 and RXEN = 1, the synchronous clock will still be output, which is then used only for receiving data, and the TX pin remains high.

14.2.3. Infrared Operating Mode

Infrared mode is used for infrared communication. The infrared mode is enabled when SIREN = 1, and the default communication data length is 8 bits.

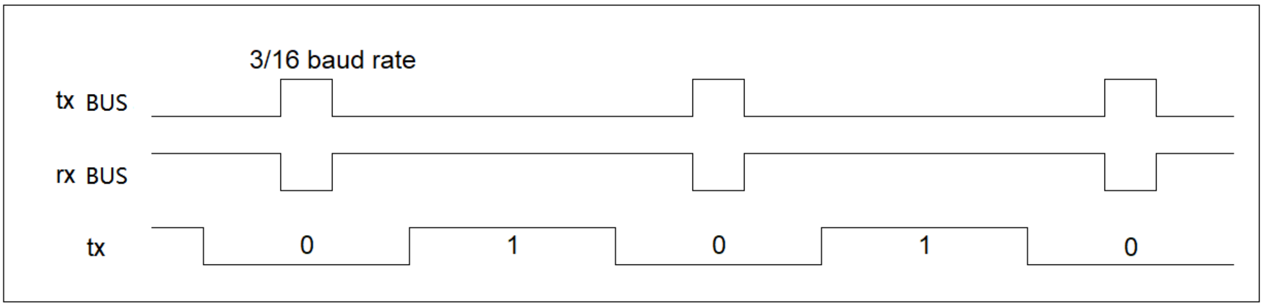


Figure 14-5 Infrared Mode Communication Timing Diagram

As shown in **Figure 14-5**, the data pulse width of the infrared module's transmittng or receiving bus is 3/16 of the bit cycle in normal mode . A high pulse is generated when the transmitted data is zero , and a low pulse is decoded as zero when it is received . The polarity of the transmittng and receiving bus are opposite, the bus keeps the low level when the transmittng is idle, and the bus keeps the high level when the receiving is idle.

IR module in normal mode , communication baud rate = Fmaster / (16 * [DLH:DLL]);

low power mode (see "SIRLP"), the communication baud rate = Fmaster / (PSC * 16 * [DLH:DLL]);

14.2.4. Smart Card Mode

The smart card mode is a half-duplex mode and supports the ISO7816-3 standard. When SDEC = 1, to enable smart card mode, the protocol requires setting the data length to 8 bits (see "LTH"), enabling the parity bit (see "PEN"), and setting the stop bit to 1.5 bits (see "URSTOP"). "URSTOP"), and configure the corresponding IO to open-drain mode.

The clock source and the frequency division output of the smart card are set by CKOE and PSC.

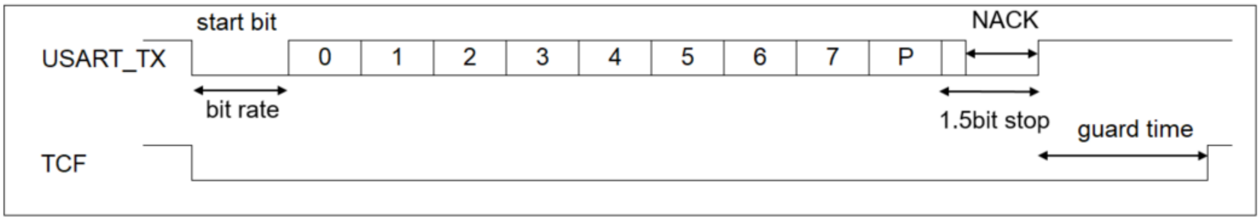


Figure 14-6 Smart Card Mode Communication Timing Diagram

When NACK = 1, after the receiver detects a parity error, it will pull down the bus for 1 bit cycle after 0.5 stop bits. At the same time, the transmitter will detect whether the bus is pulled down at the stop bit, if the bus is detected to be pulled down, the frame error flag FEF will be set to 1. The transmitter can choose to retransmit the current data upon request, and the number of transmissions is determined by the user.

When NACK = 0, the receiver will not pull down the bus after detecting a parity error, and the parity error flag PEF is set to 1.

In addition, in smart card mode, a guard time can be set (see "GT"), when the transmitter finishes transmitting data, wait for GT baud clock cycles before TCF is set.

14.2.5. LIN Master Mode

Enter LIN Master mode after LINEN is set .

The transmitter needs to configure the length of the break frame (refer to "BLTH") . When setting BKREQ = 1, the disconnected frame transmission is enabled , the TX pin will continuously send BLTH low levels, and BKREQ will auto clear after the transmission is completed.

After receiving a number of consecutive low levels greater than (START + data length + STOP), the receiver will assume that a break frame has been received and the BKF will be set to 1.

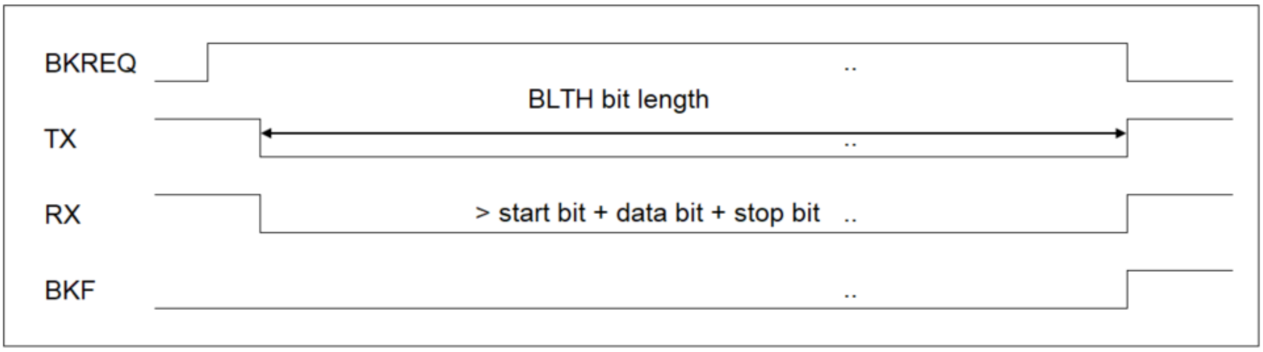


Figure 14-7 LIN Master mode

Note : The reception and transmission of disconnected frames are not only applicable to LIN Master mode , but also to other asynchronous modes , infrared modes, etc.

14.2.6. Multiprocessor Communication Mode

Multiprocessor communication mode, for example, a USART is used as the master mode, and other USARTs are used as the slave mode, and the TX output of the slave is connected to the RX input of the master by means of logical AND .

When RWU is set to 1 , the master enters mute mode, blocking all reception . According to the setting of WAKE , USART can awake the master to receive data or exit the mute mode in the following two ways :

- WAKE = 0: Wake-up by idle address. Wake-Up when an idle frame is detected and start receiving data. If the bus data is always busy, it will not Wake-Up .

  Note: Idle frame, i.e. a complete data frame consisting entirely of '1' (Number of consecutive '1' bits ≥ (START + data bit + STOP) number of bits) .

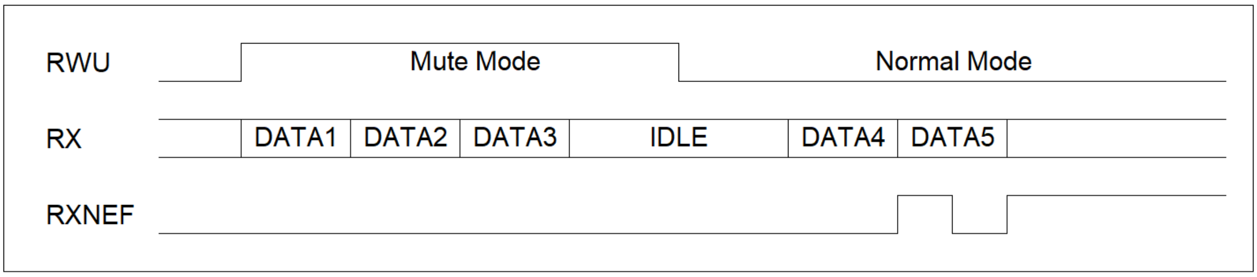


Figure 14-8 Mute Mode Address Idle Wake-Up

- WAKE = 1: Wake-up by address matching. Each time the data is received it will determine whether the high bit is 1 (indicating that the received data is address data). If the high bit is 1, continue to compare the LSB (4 bits) of the data and the value of URRAR, if they are equal, the address matching flag bit ADDRF is set to 1, Exit the mute mode and start receiving the following data . If it is not equal , it will enter the mute mode immediately .

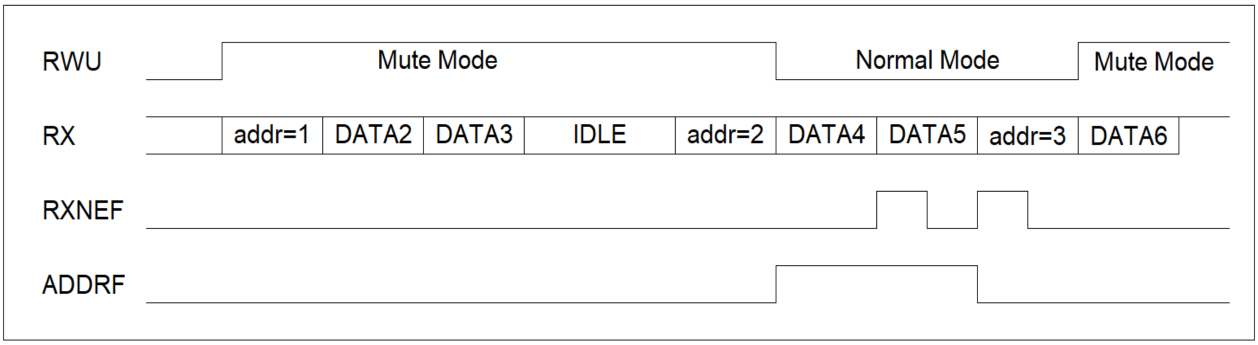


Figure 14-9 Mute Mode Address Match Wakeup

14.2.7. Automatic baud rate detection

The automatic baud rate detection function is used for the receiver to calibrate the communication baud rate so as to keep the same as the transmitter's baud rate . The baud rate detection module has two modes:

1. ABRM = 0 : Only the length of the START is detected , and the 1st bit of the data is required to be 1 . For example data 0x03, 0x55, etc.
2. ABRM = 1 : Detect the length of the START and the 1st bit , and require the 1st bit = 1 and the 2nd bit = 0 . For example data 0x55, 0x01 etc.

The baud rate detection data is used to automatically configure the DLL/DLH . If the baud rate data of the transmitter is not close to the receiver's Fbaudrate = Fmaster / (16 * {DLH, DLL}), the baud rate detection module will be automatically configured to a closer baud rate . The serial PORT module does not support decimal baud rate, so there is an error in the baud rate detection of this module.

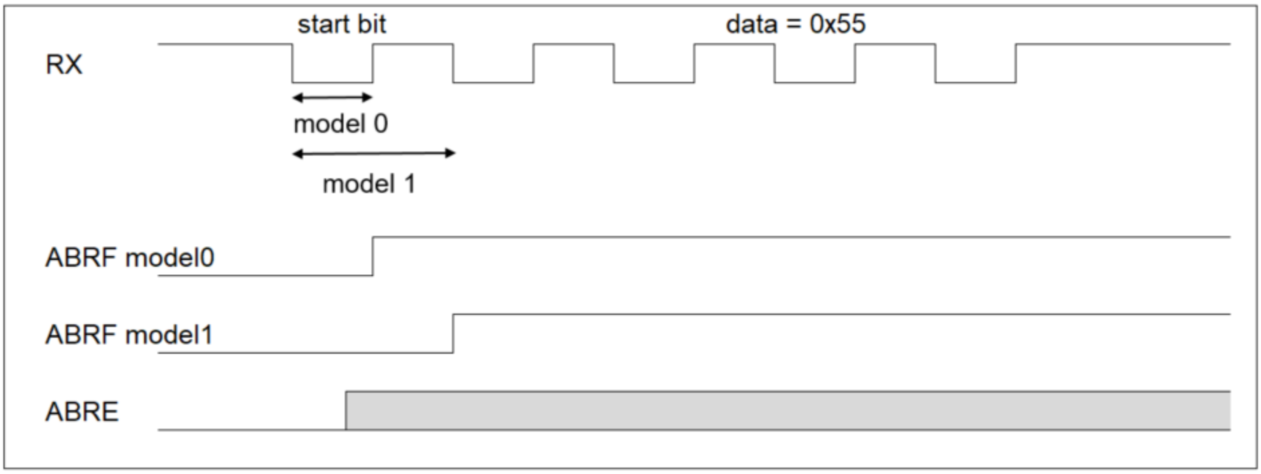


Figure 14-10 Automatic Baud Rate Detection

Process of automatic baud rate detection:

1. Select detection mode ABRM;
2. Configure ABREN = 1 to enable automatic baud rate detection;
3. Read and detect whether the baud rate flag ABRF is 1 (not cleared last time), if it is 1, write 0 to clear it;

4. Start to receive data, ABRF is set to 1 after baud rate detection is completed;
5. After the current data reception is completed, the receiving BUF as not empty flag RXNEF is set to 1;
6. Before starting the next baud rate detection, ABRF needs to be cleared first;

Note :

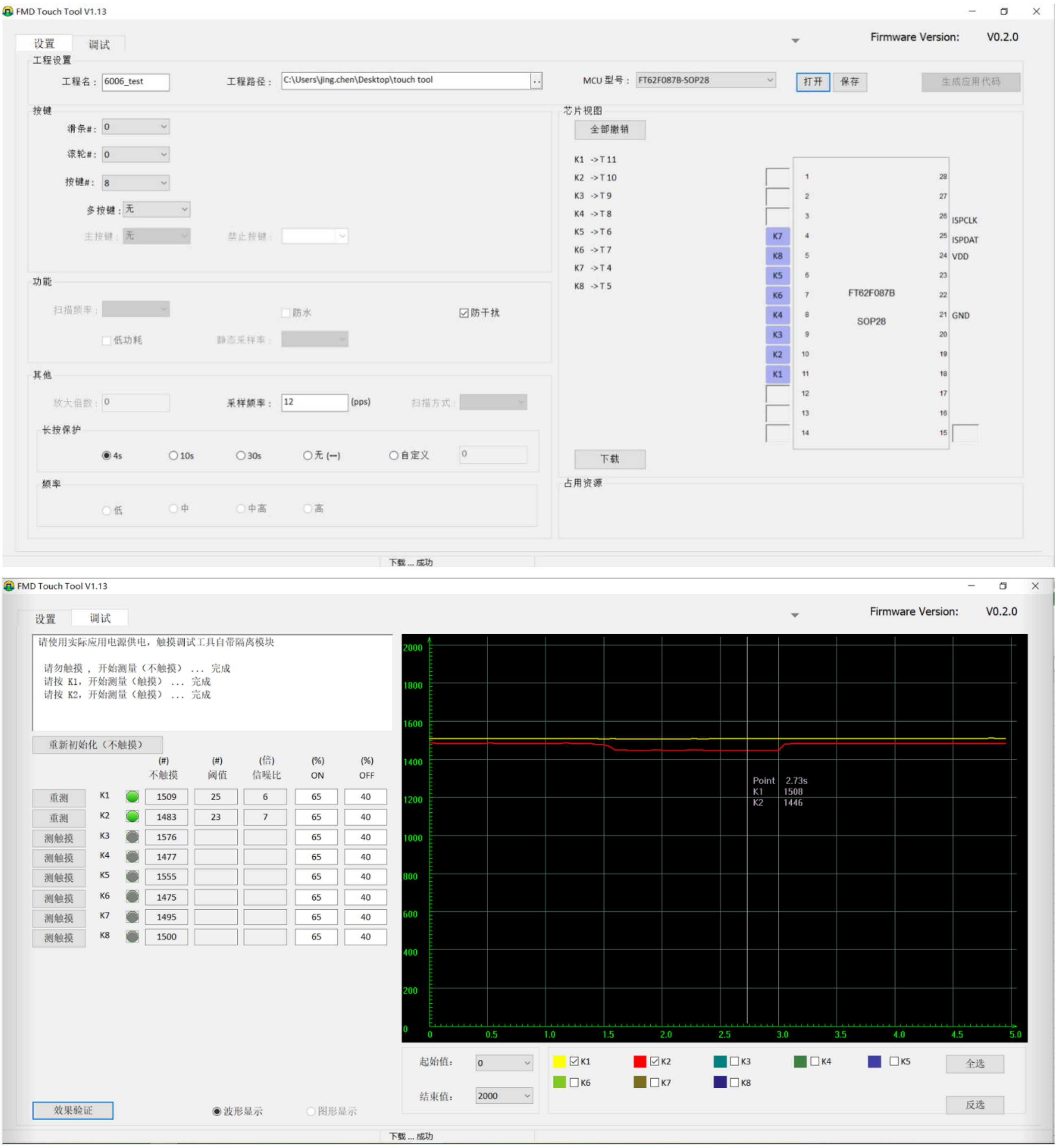
- When baud rate detection is complete, ABRF must not be cleared immediately after setting to 1. Because clearing ABRF will immediately be in the current transmission position (which may no longer be the position of the START) to perform baud rate detection, resulting in wrong results .
- When the baud rate detection is out of range, the detection overflow flag ABRE will be set to 1.

15. TOUCH

The FT62F08x has multiple touch key function, no external reference capacitor, with simple periphery and high security, to replace the traditional mechanical touch keys.

- Up to 15 touch keys
- Support waterproof function, strong anti-interference ability.

The touch applications can be quickly developed using FMD TouchTool software (IDE built-in) and library functions, as shown in **Figure 15-1** below:



FMD Touch Tool V1.13

Firmware Version: V0.2.0

工程设置

工程名：6006_test

工程路径：C:\Users\jing.chen\Desktop\touch tool

MCU 型号：FT62F087B-SOP28

按键

滑条#：0

滚轮#：0

按键#：8

多按键：无

主按键：无

功能

□ 防水

☑ 防干扰

□ 低功耗

其他

放大倍数：0

采样频率：12 (pps)

长按保护

● 4s ○ 10s ○ 30s ○ 无 (—) ○ 自定义 0

频率

○ 低 ○ 中 ○ 中高 ○ 高

芯片视图

K1 ->T 11
K2 ->T 10
K3 ->T 9
K4 ->T 8
K5 ->T 6
K6 ->T 7
K7 ->T 4
K8 ->T 5

占用资源

下载 ... 成功

```
请使用实际应用电源供电，触摸调试工具自带隔离模块

 请勿触摸 ，开始测量（不触摸） ... 完成
 请按 K1，开始测量（触摸） ... 完成
 请按 K2，开始测量（触摸） ... 完成
```

			(#) 不触摸	(#) 阈值	(倍) 信噪比	(%) ON	(%) OFF
重测	K1	●	1509	25	6	65	40
重测	K2	●	1483	23	7	65	40
测触摸	K3	●	1576			65	40
测触摸	K4	●	1477			65	40
测触摸	K5	●	1555			65	40
测触摸	K6	●	1475			65	40
测触摸	K7	●	1495			65	40
测触摸	K8	●	1500			65	40

● 波形显示 ○ 图形显示

Point 2.73s
K1 1508
K2 1446

起始值：0

结束值：2000

☑ K1 ☑ K2 □ K3 □ K4 □ K5 □ K6 □ K7 □ K8

下载 ... 成功

Figure 15-1 Touch Tool interface

16. MEMORY READ / WRITE PROTECTON

The PROGRAM AREA(PROM) can be Array Read Protected, or Sector Read/Program protected (1 k x 14 each). These protections selected at the IDE.

Name	Function	default
CPB	PROM Array Read Protection	disabled
FSECPB0	PROM Sector 0 (1k x 14) Read/Progarm Protection	disabled
FSECPB1	PROM Sector 1 (1k x 14) Read/Progarm Protection	disabled
FSECPB2	PROM Sector 2 (1k x 14) Read/Progarm Protection	disabled
FSECPB3	PROM Sector 3 (1k x 14) Read/Progarm Protection	disabled
FSECPB4	PROM Sector 4 (1k x 14) Read/Progarm Protection	disabled
FSECPB5	PROM Sector 5 (1k x 14) Read/Progarm Protection	disabled
FSECPB6	PROM Sector 6 (1k x 14) Read/Progarm Protection	disabled
FSECPB7	PROM Sector 7 (1k x 14) Read/Progarm Protection	disabled

Table 16-1 Bank Read/Write Protection Initialization Configuration Register

The difference between full encryption and sectorized encryption is as follows:

Encryption	CPU fetch	Software read	Software write	Serial read	Serial write
None	√	√	√(2)	√	√
Whole array	√	√	√(2)	×(1)	×(4)
Sector	√	×(1)	×(3)	×(3)	×(5)

Note:

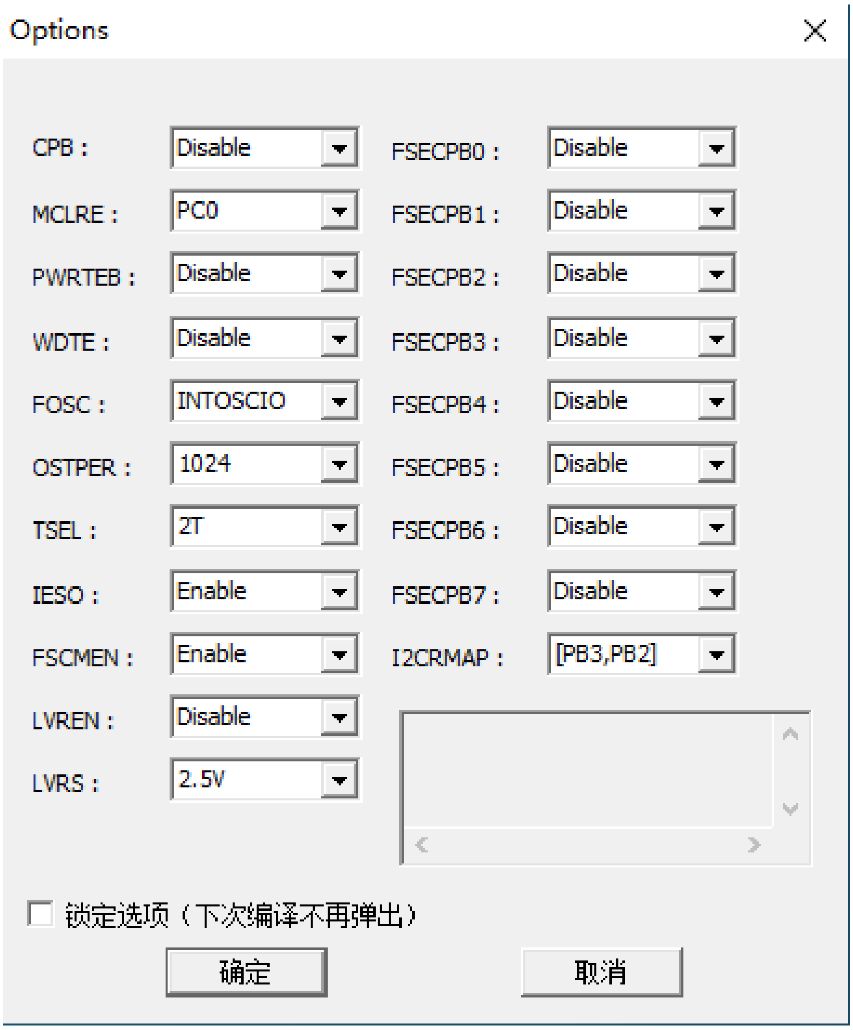
1. EEDAT keeps the previous value unchanged;
2. Software cannot program or erase UCFG pages;
3. Only unencrypted sectors can be read or written;
4. Only the serial PORT is allowed to do full chip erasure (unencryption) including UCFG;
5. Only the serial PORT is allowed to do full chip erasure (unencryption) including UCFG, or page erase and programming of unencrypted sectors;
6. Under any circumstances, software cannot do full chip erasure including UCFG;

17. SPECIAL FUNCTION REGISTERS (SFR)

There are two types of Special Function Registers (SFR).

- BOOT level registers are set at the Integrated Development Environment (IDE);
- User registers.;

17.1. Boot Level Registers



CPB : Disable

MCLRE : PC0

PWRTEB : Disable

WDTE : Disable

FOSC : INTOSCIO

OSTPER : 1024

TSEL : 2T

IESO : Enable

FSCMEN : Enable

LVREN : Disable

LVRS : 2.5V

FSECPB0 : Disable

FSECPB1 : Disable

FSECPB2 : Disable

FSECPB3 : Disable

FSECPB4 : Disable

FSECPB5 : Disable

FSECPB6 : Disable

FSECPB7 : Disable

I2CRMAP : [PB3,PB2]

□ 锁定选项（下次编译不再弹出）

Figure 17-1 Boot Selectables in the IDE

Name	Function	default
CPB	PROM All-area Read Protection	Disabled
MCLRE	Reset by External I/O	Disabled
PWRTEB	Power on delay timer (PWRT), additional delay~64ms after initialization configuration	Disabled
WDTE	WDT • Enable (Instructions can not be disabled) • Instruction controls (SWDTEN)	SWDTEN control
FOSC	• LP: external low-speed oscillator across PC1 (+) and PB7 (−) • XT: external high-speed oscillator across PC1 (+) and PB7 (−) • EC: external oscillator at PC1 (+) , PB7 as I/O • INTOSCIO : PC1 and PB7 as I/O	INTOSCIO
OSTPER	OST timer period selection (XT / L P applies) • 512 • 1024 • 2048 • 4096 (32768 in LP mode)	1024
TSEL	The correspondence between the instruction clock and the system clock SysClk (1T, 2T or 4T): • 1 (Instruction Clock = SysClk) • 2 (Instruction Clock = SysClk/2) • 4 (Instruction Clock = SysClk/4)	2
FSCMEN	Fail-Safe Clock Monitor • Enable • Disable	Enabled
IESO	XT/ LP two-speed clock start • Enable • Disable	Enabled
LVREN	LVR • Enable • Disable • Enable in non-SLEEP mode • Instruction controlled (SLVREN)	Disabled
LVRS	7 V_{BOR} Voltage levels (V): 2.0 / 2.2 / 2.5 / 2.8 / 3.1 / 3.6 / 4.1	Disabled
FSECPB0	PROM Sector 0 (1k x 14) Read/Write Protection	Disabled
FSECPB1	PROM Sector 1 (1k x 14) Read/Write Protection	Disabled

Name	Function	default
FSECPB2	PROM Sector 2 (1k x 14) Read/Write Protection	Disabled
FSECPB3	PROM Sector 3 (1k x 14) Read/Write Protection	Disabled
FSECPB4	PROM Sector 4 (1k x 14) Read/Write Protection	Disabled
FSECPB5	PROM Sector 5 (1k x 14) Read/Write Protection	Disabled
FSECPB6	PROM Sector 6 (1k x 14) Read/Write Protection	Disabled
FSECPB7	PROM Sector 7 (1k x 14) Read/Write Protection	Disabled

I2CRMAP	I2C multiplexed pin selection
	[PB3, PB2] : (≥ Version I chip is applicable)
	I2C_SDA = PB3, I2C_SCL = PB2;
	SPI_MOSI = PA0, SPI_MISO = PA1
	[PA0, PA1]:
	I2C_SDA = PA0, I2C_SCL = PA1;
	SPI_MOSI = PB3, SPI_MISO = PB2
		[PB3,PB2]

Table 17-1 Boot Level Registers (set by IDE)

17.2. User Registers

User Speical Function Registers (SFR) and SRAM are stored in 14 banks (bank0~12, bank31), and each bank is 128 bytes in size. The corresponding bank must be selected before the registers inside can be accessed. An active bank can be selected by writing the bank number (0~12, 31) to the Bank Selection Register (BSREG) . The address of the user register is 12 bits, the address range is 0x000 ~ 0xFFF , the MSB (5 bits) are the bank area address, and the LSB (7 bits) are the SFR/SRAM address.

All user registers can be accessed directly through INDFn, or indirectly through the FSRn File Selection Register (see **Section 17.5**, Indirect addressing).

Since extra instructions are involved in switching BANK, some often-used SFR are stored in all 14 banks to minimize switching. Registers common to all 14 BANKS are synchronized.

Bank	First address	Bank	First address
Bank0	000H	Bank7	380H
Bank1	080H	Bank8	400H
Bank2	100H	Bank9	480H
Bank3	180H	Bank10	500H
Bank4	200H	Bank11	580H
Bank5	280H	Bank12	600H
Bank6	300H	Bank31	F80H

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
First address + 0H	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
First address + 1H	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
First address + 2H	PCL	LSB of Program Counter's (PC)								0000 0000
First address + 3H	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
First address + 4H	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
First address + 5H	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
First address + 6H	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
First address + 7H	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
First address + 8H	BSREG	Bank Selection Register								xxxx xxxx
First address + 9H	WREG	Working register W								xxxx xxxx
First address + AH	PCLATH	–	MSB of Program counter(PC) latches							−−−0 0000
First address + BH	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
First address + (70 − 7F)		COMMON BANK SRAM								xxxx xxxx

Table 17-2 Registers common to 14 BANKs

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
000	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
001	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
002	PCL	LSB of Program Counter's (PC)								0000 0000
003	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
004	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
005	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
006	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
007	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
008	BSREG	Bank Selection Register								xxxx xxxx
009	WREG	Working register W								xxxx xxxx
00A	PCLATH	–	MSB of Program counter(PC) latches							–000 0000
00B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
00C	PORTA	PORTA [7:0]								xxxx xxxx
00D	PORTB	PORTB [7:0]								xxxx xxxx
00E	PORTC	PORTC[7:0]								xxxx xxxx
00F	PORTD	–	–	TRISD [5:0]						––xx xxxx
011	PIR1	–	–	–	–	–	TKIF	CKMIF	ADCIF	–––– –000
014	EPIF0	External pin interrupt Flag								0000 0000
015	SPIDATA	SPI data transmit/receive BUF register								0000 0000
016	SPICTRL	SPIF	WCOL	MODF	RXOVRN	NSSM [1:0]		TXBMT	SPIEN	0000 0010
017	SPICFG	BUSY	MSTEN	CPHA	CPOL	SLAS	NSSVAL	SRMT	RXBMT	0000 0111
018	SPISCR	Baud rate setting register								0000 0000
019	SPICRCPOL	CRC calculation polynomial								0000 0111
01A	SPIRXCRC	CRC calculation result of received data								0000 0000
01B	SPITXCRC	CRC calculation result of the transmitted data								0000 0000
01C	SPIIER	–	–	–	–	WAKUP	RXERR	RXNE	TXE	–––– 0000
01D	SPICTRL2	BDM	BDOE	RXONLY	SSI	SSM	CRCNXT	CRCEN	LSBFIRST	0000 0000
01E	SPISTAT	–	SMODF	SRXOVRN	SBUSY	SRXBMT	STXBMT	WKF	CRCERR	–000 1100
01F	ADDLY /LEBPRL	LSB of ADC external trigger start delay counter threshold / multiplexed as leading edge blanking count threshold								0000 0000
020–06F		SRAM BANK0 (80Bytes)								xxxx xxxx
070–07F		SRAM BANK0 (16Bytes), physical address 0x70–0x7F								xxxx xxxx

Table 17-3 SFR, BANK 0

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
080	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
081	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
082	PCL	LSB of Program Counter's (PC)								0000 0000
083	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
084	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
085	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
086	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
087	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
088	BSREG	Bank Selection Register								xxxx xxxx
089	WREG	Working register W								xxxx xxxx
08A	PCLATH	–	MSB of Program counter(PC) latches							−000 0000
08B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
08C	TRISA	TRISA[7:0]								1111 1111
08D	TRISB	TRISB[7:0]								1111 1111
08E	TRISC	TRISC[7:0]								1111 1111
08F	TRISD	–	–	TRISD[5:0]						−−11 1111
091	PIE1	–	–	–	–	–	TKIE	CKMIE	ADCIE	−−−− −000
094	EPIE0	External pin interrupt enable								0000 0000
095	CKOCON	SYSON	CCORDY	DTYSEL		CCOSEL[2:0]			CCOEN	0010 0000
096	PCON	STKOVF	STKUNF	EMCF	IERRR	/MCLRR	/SRTF	/PORF	/BORF	qqqq qqqq
097	WDTCON	WDTPRE[2:0]			WDTPS[3:0]				SWDTEN	1110 1000
098	OSCTUNE	–	HIRC Clock Frequency Tuner							−xxx xxxx
099	OSCCON	MCKCF[3:0]				OSTS	HTS	LTS	SCS	0100x000
09A	PCKEN	TKEN	I2CEN	UARTEN	SPICKEN	TIM4EN	TIM2EN	TIM1EN	ADCEN	0000 0000
09B	ADRESL	ADC result register LSB								0000 0000
09C	ADRESH	ADC result register MSB								0000 0000
09D	ADCON0	CHS[3:0]				ADCAL	ADEX	GO/DONE	ADON	0000 0000
09E	ADCON1	ADFM	ADCS[2:0]			ADNREF[1:0]		ADPREF[1:0]		0000 0000
09F	ADCON2	ADINTREF[1:0]		ETGTYP[1:0]		ADDLY.8	ETGSEL[2:0]			0000 0000
0A0–0EF		SRAM BANK1 (80Bytes)								xxxx xxxx
0F0–0FF		SRAM BANK1 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-4 SFR, BANK 1

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
100	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
101	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
102	PCL	LSB of Program Counter's (PC)								0000 0000
103	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
104	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
105	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
106	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
107	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
108	BSREG	Bank Selection Register								xxxx xxxx
109	WREG	Working register W								xxxx xxxx
10A	PCLATH	–	MSB of Program counter(PC) latches							– 000 0000
10B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
10C	LATA	LATA[7:0]								xxxx xxxx
10D	LATB	LATB[7:0]								xxxx xxxx
10E	LATC	LATC[7:0]								xxxx xxxx
10F	LATD	–	–	LATD[5:0]						–– xx xxxx
111	TIM4CR1	T4ARPE	–	T4CKS[1:0]		T4OPM	T4URS	T4UDIS	T4CEN	0 – 00 0000
112	TIM4IER	–	–	–	–	–	–	–	T4UIE	---- ---0
113	TIM4SR	–	–	–	–	–	–	–	T4UIF	---- ---0
114	TIM4EGR	–	–	–	–	–	–	–	T4UG	---- ---0
115	TIM4CNTR	T4CNT[7:0]								0000 0000
116	TIM4PSCR	–	–	–	–	–	T4PSC[2:0]			---- -000
117	TIM4ARR	T4ARR[7:0]								1111 1111
118	EPS0	EPS0[7:0]								0000 0000
119	EPS1	EPS1[7:0]								0000 0000
11A	PSRC0	PSRCB[3:0]				PSRCA[3:0]				1111 1111
11B	PSRC1	PSRCD[3:0]				PSRCC[3:0]				1111 1111
11C	MISC0	–	–	–	–	–	–	WCKSEL[1:0]		---- --00
11D	AFP2	–	–	–	AFP2[4:0]					---0 0000
11E	ITYPE0	ITYPE0[7:0]								0000 0000
11F	ITYPE1	ITYPE1[7:0]								0000 0000
120–16F		SRAM BANK2 (80Bytes)								xxxx xxxx
170–17F		SRAM BANK2 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-5 SFR, BANK 2

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
180	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
181	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
182	PCL	LSB of Program Counter's (PC)								0000 0000
183	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
184	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
185	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
186	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
187	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
188	BSREG	Bank Selection Register								xxxx xxxx
189	WREG	Working register W								xxxx xxxx
18A	PCLATH	–	MSB of Program counter(PC) latches							– 000 0000
18B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
18C	WPUA	WPUA[7:0]								0000 0000
18D	WPUB	WPUB[7:0]								0000 0000
18E	WPUC	WPUC[7:0]								0000 0000
18F	WPUD	WPUD[7:0]								0000 0000
191	EEADRL	EEADR[7:0]								0000 0000
192	EEADRH	–	–	–	EEADR[12:8]					---0 0000
193	EEDATL	EEDAT[7:0]								xxxx xxxx
194	EEDATH	–	–	EEDAT[13:8]						--xx xxxx
195	EECON1	EEPGD	CFGS	–	FREE	WRERR	WREN	WR	RD	00-0x000
196	EECON2	EEPROM Control Register 2								xxxx xxxx
197	ANSELA	Analog Pin Setting Register								0000 0000
198	EECON3	–	–	–	–	–	–	–	DRDEN	---- ---0
199	LVDCON	SLVREN	LVDM	–	LVDEN	LVDW	LVDL[3:0]			0000 0000
19A	PSINK0	PSINK0[7:0]								0000 0000
19B	PSINK1	PSINK1[7:0]								0000 0000
19C	PSINK2	PSINK2[7:0]								0000 0000
19D	PSINK3	–	–	PSINK3[5:0]						--00 0000
19E	AFP0	AFP0[7:0]								0000 0000
19F	AFP1	–	AFP1[7:0]							-000 0000
1A0–1EF		SRAM BANK3 (80Bytes)								xxxx xxxx
1F0–1FF		SRAM BANK3 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-6 SFR, BANK3

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
200	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
201	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
202	PCL	LSB of Program Counter's (PC)								0000 0000
203	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
204	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
205	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
206	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
207	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
208	BSREG	Bank Selection Register								xxxx xxxx
209	WREG	Working register W								xxxx xxxx
20A	PCLATH	–	MSB of Program counter(PC) latches							– 000 0000
20B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
20C	WPDA	WPDA[7:0]								0000 0000
20D	WPDB	WPDB[7:0]								0000 0000
20E	WPDC	WPDC[7:0]								0000 0000
20F	WPDD	WPDD[7:0]								0000 0000
211	TIM1CR1	T1ARPE	T1CMS[1:0]		TIDIR	T1OPM	TIURS	T1UDS	T1CEN	0000 0000
213	TIM1SMCR	–	T1TS[2:0]			–	T1SMS[2:0]			-000 -000
215	TIM1IER	T1BIE	T1TIE	–	T1CC4IE	T1CC3IE	T1CC2IE	T1CC1IE	T1UIE	00-0 0000
216	TIM1SR1	T1BIF	T1TIF	–	T1CC4IF	T1CC3IF	T1CC2IF	T1CC1IF	T1UIF	00-0 0000
217	TIM1SR2	–	–	–	T1CC4OF	T1CC3OF	T1CC2OF	T1CC1OF	–	---0 000-
218	TIM1EGR	T1BG	–	–	T1CC4G	T1CC3G	T1CC2G	T1CC1G	–	0—0 000-
219	TIM1CCMR1 (output mode)	–	T1OC1M[2:0]			T1OC1PE	–	T1CC1S[1:0]		-000 0-00
	TIM1CCMR1 (input mode)	T1IC1F[3:0]				T1IC1PSC[1:0]		T1CC1S[1:0]		0000 0000
21A	TIM1CCMR2 (output mode)	–	T1OC2M[2:0]			T1OC2PE	–	T1CC2S[1:0]		-000 0-00
	TIM1CCMR2 (input mode)	T1IC2F[3:0]				T1IC2PSC[1:0]		T1CC2S[1:0]		0000 0000
21B	TIM1CCMR3 (output mode)	–	T1OC3M[2:0]			T1OC3PE	–	T1CC3S[1:0]		-000 0-00
	TIM1CCMR3	T1IC3F[3:0]				T1IC3PSC[1:0]		T1CC3S[1:0]		0000 0000

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
	(input mode)									
21C	TIM1CCMR4 (output mode)	−	T1OC4M[2:0]			T1OC4PE	−	T1CC4S[1:0]		-000 0-00
	TIM1CCMR4 (input mode)	T1IC4F[3:0]				T1IC4PSC[1:0]		T1CC4S[1:0]		0000 0000
21D	TIM1CCER1	T1CC2N	T1CC2N	T1CC2P	T1CC2E	T1CC1NP	TICC1NE	T1CC1P	T1CC1E	0000 0000
21E	TIM1CCER2	−	−	T1CC4P	T1CC4E	T1CC3NP	TICC3NE	T1CC3P	T1CC3E	--00 0000
21F	ODCON0	−	−	−	−	−	SPIOD	I2CON	UROD	---- -000
220–26F		SRAM BANK4 (80Bytes)								xxxx xxxx
270–27F		SRAM BANK4 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-7 SFR, BANK4

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
280	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
281	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
282	PCL	LSB of Program Counter's (PC)								0000 0000
283	STATUS	−	−	−	/TO	/PD	Z	DC	C	0001 1xxx
284	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
285	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
286	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
287	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
288	BSREG	Bank Selection Register								xxxx xxxx
289	WREG	Working register W								xxxx xxxx
28A	PCLATH	−	MSB of Program counter(PC) latches							− 000 0000
28B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
28C	TIM1CNTRH	T1CNT[15:8]								0000 0000
28D	TIM1CNTRL	T1CNT[7:0]								0000 0000
28E	TIM1PSCRH	T1PSC[15:8]								0000 0000
28F	TIM1PSCRL	T1PSC[7:0]								0000 0000
290	TIM1ARRH	T1ARR[15:8]								1111 1111
291	TIM1ARRL	T1ARR[7:0]								1111 1111

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
292	TIM1RCR	TIREP[7:0]								0000 0000
293	TIM1CCR1H	T1CCR1[15:8]								0000 0000
294	TIM1CCR1L	T1CCR1[7:0]								0000 0000
295	TIM1CCR2H	T1CCR2[15:8]								0000 0000
296	TIM1CCR2L	T1CCR2[7:0]								0000 0000
297	TIM1CCR3H	T1CCR3[15:8]								0000 0000
298	TIM1CCR3L	T1CCR3[7:0]								0000 0000
299	TIM1CCR4H	T1CCR4[15:8]								0000 0000
29A	TIM1CCR4L	T1CCR4[7:0]								0000 0000
29B	TIM1BKR	T1MOE	T1AOE	T1BKP	T1BKE	T1OSSR	T1OSSI	T1LOCK[1:0]		0000 0000
29C	TIM1DTR	T1DGT[7:0]								0000 0000
29D	TIM1OISR	–	T1OIS4	T1OIS3N	T1OIS3	T1OIS2N	T1OIS2	T1OIS1N	T1OIS1	-000 0000
29E	TIM2CCR3H	T2CCR3[15:8]								0000 0000
29F	TIM2CCR3L	T2CCR3 [7:0]								0000 0000
2A0–2EF		SRAM BANK5 (80Bytes)								xxxx xxxx
2F0–2FF		SRAM BANK5 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-8 SFR, BANK5

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
300	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
301	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
302	PCL	LSB of Program Counter's (PC)								0000 0000
303	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
304	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
305	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
306	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
307	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
308	BSREG	Bank Selection Register								xxxx xxxx
309	WREG	Working register W								xxxx xxxx
30A	PCLATH	–	MSB of Program counter(PC) latches							– 000 0000
30B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
30C	TIM2CR1	T2ARPE	–	–	–	T2OPM	T2URS	T2UDIS	T2CEN	0--- 0000
30D	TIM2IER	–	–	–	–	T2CC3IE	T2CC2IE	T2CC1IE	T2UIE	---- 0000
30E	TIM2SR1	–	–	–	–	T2CC3IF	T2CC2IF	T2CC1IF	T2UIF	---- 0000
30F	TIM2SR2	–	–	–	–	T2CC3OF	T2CC3OF	T2CC3OF	–	---- 000-
310	TIM2EGR	–	–	–	–	T2CC3G	T2CC2G	T2CC1G	T2UG	---- 0000
311	TIM2CCMR1 (output mode)	–	T2OC1M[2:0]			T2OC1PE	-	T2CC1S[1:0]		-000 0-00
	TIM2CCMR1 (input mode)	T2IC1F[3:0]				T2IC1PSC[1:0]		T2CC1S[1:0]		0000 0000
312	TIM2CCMR2 (output mode)	–	T2OC2M[2:0]			T2OC2PE	–	T2CC2S[1:0]		-000 0-00
	TIM2CCMR2 (input mode)	T2IC2F[3:0]				T2IC2PSC[1:0]		T2CC2S[1:0]		0000 0000
313	TIM2CCMR3 (output mode)	–	T2OC3M[2:0]			T2OC3PE	–	T2CC3S[1:0]		-000 0-00
	TIM2CCMR3 (input mode)	T2IC3F[3:0]				T2IC3PSC[1:0]		T2CC3S[1:0]		0000 0000
314	TIM2CCER1	–	–	T2CC2P	T2CC2E	–	–	T2CC1P	T2CC1E	--00 --00
315	TIM2CCER2	–	–	–	–	–	–	T2CC3P	T2CC3E	---- --00
316	TIM2CNTRH	T2CNT[15:8]								0000 0000
317	TIM2CNTRL	T2CNT[7:0]								0000 0000
318	TIM2PSCR	–	–	–	–	T2PSC[3:0]				---- 0000
319	TIM2ARRH	T2ARR[15:8]								1111 1111
31A	TIM2ARRL	T2ARR[7:0]								1111 1111
31B	TIM2CCR1H	T2CCR1[15:8]								0000 0000
31C	TIM2CCR1L	T2CCR1[7:0]								0000 0000
31D	TIM2CCR2H	T2CCR2[15:8]								0000 0000
31E	TIM2CCR2L	T2CCR2[7:0]								0000 0000
31F	TCKSRC	LFMOD	T2CKSRC[2:0]			–	T2CKSRC[2:0]			0000 -000
320–36F		SRAM BANK6 (80Bytes)								xxxx xxxx
370–37F		SRAM BANK6 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-9 SFR, BANK6

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
380	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
381	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
382	PCL	LSB of Program Counter's (PC)								0000 0000
383	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
384	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
385	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
386	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
387	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
388	BSREG	Bank Selection Register								xxxx xxxx
389	WREG	Working register W								xxxx xxxx
38A	PCLATH	–	MSB of Program counter(PC) latches							– 000 0000
38B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
3A0–3EF		SRAM BANK7 (80Bytes)								xxxx xxxx
3F0–3FF		SRAM BANK7 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-10 SFR, BANK7

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
400	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
401	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
402	PCL	LSB of Program Counter's (PC)								0000 0000
403	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
404	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
405	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
406	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
407	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
408	BSREG	Bank Selection Register								xxxx xxxx
409	WREG	Working register W								xxxx xxxx
40A	PCLATH	–	MSB of Program counter(PC) latches							– 000 0000
40B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
40C	I2CCR1	–	–	–	MST10B	SLV10B	–	SPEED	MASTER	---0 0-00
40D	I2CCR2	–	SOFTRST	AGCALL	SNACK	–	–	RXHLD	–	-000 –0-
40E	I2CCR3	–					EVSTRE	–	ENABLE	---- -0-0
40F	I2COARL	ADD[7:0]								0000 0000
410	I2COARH	–	–	–	–	–	–	ADD[9:8]		---- --00
411	I2CFREQ	–	–	FREQ[5:0]						--00 0000
412	I2CDR	DR[7:0]								0000 0000
413	I2CCMD	–	–	–	–	–	RESTART	STOP	MSTDIR	---- -000
414	I2CCCRL	CCR[7:0]								0000 0000
415	I2CCCRH	–	DUTY	–	–	CCR[11:8]				-0-- 0000
416	I2CITR	–					ITBUFEN	ITEVEN	ITERREN	---- -000
417	I2CSR1	IICTXE	IICRXNE	–	STOPF	ADD10F	–	ADDF	SBF	00-0 0-00
418	I2CSR2	–	–	–	TXABRT	OVR	AF	ARLO	BERR	---0 0000
419	I2CSR3	–	–	GCALL	–	–	RDREQ	ACTIVE	RXHOLD	--0- -000
41A	ADCON3	ADFBEN	ADCMPOP	ADCMPEN	ADCMPO	LEBADT	–	ELVDS[1:0]		0000 0-00
41B	ADCMPH	ADCMPH[7:0]								0000 0000
41C	LEBCON	LEBEN	LEBCH[1:0]		–	EDGS	BKS[2:0]			000- 0000
41D	MSCKCON	–	–	–	–	–	–	CKMAVG	CKCNTI	---- --01
41E	SOSCPRL	SOSCPR[7:0]								1111 1111
41F	SOSCPRH	–	–	–	–	SOSCPR[11:8]				---- 1111
420–46F		SRAM BANK8 (80Bytes)								xxxx xxxx
470–47F		SRAM BANK8 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-11 SFR, BANK8

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
480	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
481	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
482	PCL	LSB of Program Counter's (PC)								0000 0000
483	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
484	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
485	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
486	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
487	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
488	BSREG	Bank Selection Register								xxxx xxxx
489	WREG	Working register W								xxxx xxxx
48A	PCLATH	–	MSB of Program counter(PC) latches							− 000 0000
48B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
48C	URDATAL	DATAL[7:0]								0000 0000
48D	URDATAH	–							DATAH	---- ---0
48E	URIER	–	–	TCEN	–	IDELE	RXSE	URTE	URRXNE	--0- 0000
48F	URLCR	–	BKREQ	–	EVEN	PEN	URSTOP	–	LTH	-0-0 00-0
490	URLCREXT	–	–	–	–	–	–	RWU	EXTEN	---- --00
491	URMCR	–	–	SIRLP	TXEN	RXEN	WAKE	HDSEL	SIREN	--00 0000
492	URLSR	ADDRF	IDLEF	TXEF	BKF	FEF	PEF	OVERF	RXNEF	0000 0000
493	URRAR	–	–	–	–	RAR[3:0]				---- 0000
494	URDLL	DLL[7:0]								0000 0000
495	URDLH	DLH[7:0]								0000 0000
496	URABCR	–	–	–	–	ABRE	ABRM	ABRF	ABREN	---- 0000
497	URSYNCR	–	–	–	–	LBCL	URCPHA	URCPOL	SYNEN	---- 0000
498	URLINCR	–	–	–	LINEN	BLTH[3:0]				---0 0000
499	URSDCR0	–	NACK	CKOE	SDEN	–	–	–	–	-000 0000
49A	URSDCR1	GT[7:0]								0000 0000
49B	URSDCR2	PSC[7:0]								0000 0000
49C	URTC	–	–	–	–	–	–	–	TCF	---- ---1
4A0–4EF		SRAM BANK9 (80Bytes)								xxxx xxxx
4F0–4FF		SRAM BANK9 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-12 SFR, BANK9

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
500	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
501	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
502	PCL	Program counter LSB								0000 0000
503	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
504	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
505	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
506	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
507	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
508	BSREG	Bank Selection Register								xxxx xxxx
509	WREG	Working register W								xxxx xxxx
50A	PCLATH	–	MSB of Program counter(PC) latches							– 000 0000
50B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
520–56F		SRAM BANK10 (80Bytes)								xxxx xxxx
570–57F		SRAM BANK10 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-13 SFR, BANK10

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
580	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
581	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
582	PCL	LSB of Program Counter's (PC)								0000 0000
583	STATUS	–	–	–	/TF	/PF	Z	DC	C	0001 1xxx
584	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
585	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
586	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
587	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
588	BSREG	Bank Selection Register								xxxx xxxx
589	WREG	Working register W								xxxx xxxx
58A	PCLATH	–	MSB of Program counter(PC) latches							– 000 0000
58B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
5A0–5EF		SRAM BANK11 (80Bytes)								xxxx xxxx
5F0–5FF		SRAM BANK11 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-14 SFR, BANK11

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
600	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
601	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
602	PCL	LSB of Program Counter's (PC)								0000 0000
603	STATUS	−	−	−	/TO	/PD	Z	DC	C	0001 1xxx
604	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
605	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
606	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
607	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
608	BSREG	Bank Selection Register								xxxx xxxx
609	WREG	Working register W								xxxx xxxx
60A	PCLATH	−	MSB of Program counter(PC) latches							− 000 0000
60B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
620–64F		SRAM BANK12 (48Bytes)								xxxx xxxx
670–67F		SRAM BANK12 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-15 SFR, BANK12

ADDR	Name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset
F80	INDF0	Addressing this location uses contents of FSR0 to address data memory (not a physical register)								xxxx xxxx
F81	INDF1	Addressing this location uses contents of FSR1 to address data memory (not a physical register)								xxxx xxxx
F82	PCL	LSB of Program Counter's (PC)								0000 0000
F83	STATUS	–	–	–	/TO	/PD	Z	DC	C	0001 1xxx
F84	FSR0L	LSB of indirect addressing pointer register FSR0								xxxx xxxx
F85	FSR0H	MSB of indirect addressing pointer register FSR0								xxxx xxxx
F86	FSR1L	LSB of indirect addressing pointer register FSR1								xxxx xxxx
F87	FSR1H	MSB of indirect addressing pointer register FSR1								xxxx xxxx
F88	BSREG	Bank Selection Register								xxxx xxxx
F89	WREG	Working register W								xxxx xxxx
F8A	PCLATH	–	MSB of Program counter(PC) latches							– 000 0000
F8B	INTCON	GIE	PEIE	EEIE	LVDIE	OSFIE	EEIF	LVDIF	OSFIF	0000 0000
FE4	STATUS_SHAD	STATUS shadow register								xxxx xxxx
FE5	WREG_SHAD	WREG shadow register								xxxx xxxx
FE6	BSREG_SHAD	BSREG shadow register								xxxx xxxx
FE7	PCLATH_SHAD	PCLATH shadow register								xxxx xxxx
FE8	FSR0L_SHAD	FSR0L shadow register								xxxx xxxx
FE9	FSR0H_SHAD	FSR0H shadow register								xxxx xxxx
FEA	FSR1L_SHAD	FSR1L shadow register								xxxx xxxx
FEB	FSR1H_SHAD	FSR1H shadow register								xxxx xxxx
FEC	_	_								_
FED	STKPTR	STKPTR								xxxx xxxx
FEE	TOSL	TOSL								xxxx xxxx
FEF	TOSH	TOSH								xxxx xxxx
FF0–FFF		SRAM BANK31 (16Bytes), access BANK0's physical address 0x70–0x7F								xxxx xxxx

Table 17-16 SFR, BANK31

Notes:

1. INDF is not a physical register;
2. Gray parts indicate not used;
3. Do not write to unimplemented register bits.

17.3. STATUS register

Name	Status	Register	Addr.	Reset
/TO	Time-out Flag 1 = CLRWDT or SLEEP instruction after Power-up 0 = WDT time-out occurred	STATUS[4]	Bank First address + 0x03	RO−1
/PD	Power-down Flag 1 = After power-up or by the CLRWDT instruction 0 = By execution of the SLEEP instruction	STATUS[3]		RO−1
Z	Zero: Result of an arithmetic or logic operation is zero Flag 1 = Yes 0 = No	STATUS[2]		RW−x
DC	Half carry/half borrow: Half Carry-Over or Borrow from the 4th low-order bit of the result 1 = Carry-Over, Yes; Borrow, No 0 = Carry-Over, No; Borrow, Yes	STATUS[1]		RW−x
C	Carry/borrow: Digit Carry-Over or Borrow from MSB of the result 1 = Carry-Over, Yes; Borrow, No 0 = Carry-Over, No; Borrow, Yes	STATUS[0]		RW−x

Notes:

1. The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.
2. It is recommended, therefore, that only BCR, BSR, SWAPR and STR instructions are used to alter the STATUS register.

17.4. Stack

FT62F08x has a hardware stack of 16 levels deep x 15 bits wide. The stack space is independent of program PROM , data EEPROM or data storage area SRAM.

TOSH:TOSL points to the top of the stack, and STKPTR is the current value of the stack pointer. When accessing the stack, the STKPTR value used to locate TOSH:TOSL can be tuned and then read/write operations can be performed on TOSH:TOSL.

During normal program operation , LCALL, CALLW, and interrupts increment the STKPTR value by 1, and the PC value is pushed onto the stack. When the RETW, RET and RETI instructions are executed , the PC value is popped from the stack and the STKPTR value is decremented by 1.The PCLATH value is not affected by push or pop operations. Available stack space can be viewed by reading STKPTR .

STKPTR is 5 bits, allowing detection of overflow and underflow. Performing a push operation after filling 16 levels will result in an overflow and the STKOVF flag will be set to 1. After popping the first level and then performing the pop operation, an underflow will occur, and the STKUNF flag will be set to 1. An overflow or underflow event will cause the system to reset, and the 16-level stack will all be cleared to 0.

Note: Be careful when modifying STKPTR with interrupts enabled.

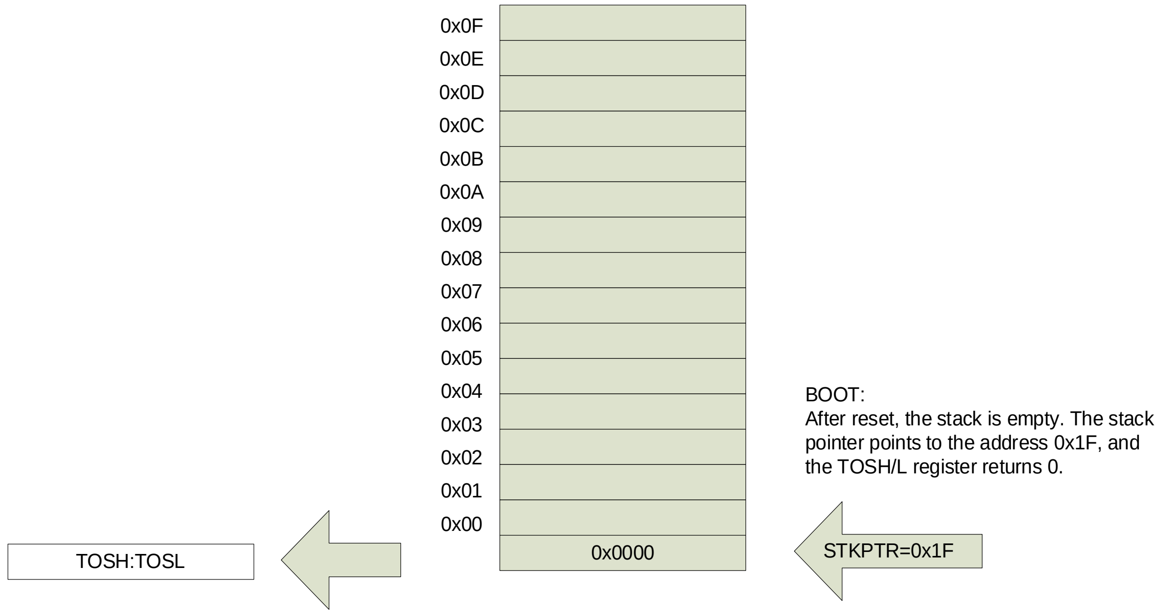

0x0F
0x0E
0x0D
0x0C
0x0B
0x0A
0x09
0x08
0x07
0x06
0x05
0x04
0x03
0x02
0x01
0x00
0x0000
TOSH:TOSL
STKPTR=0x1F
BOOT:
After reset, the stack is empty. The stack pointer points to the address 0x1F, and the TOSH/L register returns 0.

Figure 17-2 Software Access Stack

17.5. Indirect addressing

INDFn is not a physically register, and addressing INDFn will result in indirect addressing. Any instruction that accesses the INDFn register actually accesses the unit pointed to by the File Selection Register (FSRn) . An indirect READ of INDF will return 0 , and an indirect WRITE of INDF will result in a nop (possibly affecting the status flags). FSRnH : The 16-bit address composed of FSRnL allows 65536 address units to be addressed, which can be divided into 3 memory areas:

- Conventional data memory
- Linear data memory
- Flash

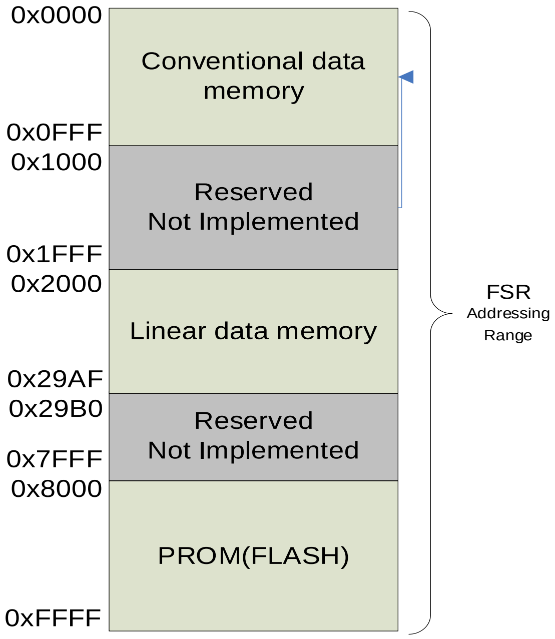


Figure 17-3 Indirect addressing

17.5.1. Conventional data memory

Conventional data memory, i.e. user register, with an address range of 0x0000 ~ 0x0FFF , correspond to the absolute addresses of all SFRs and SRAMs.

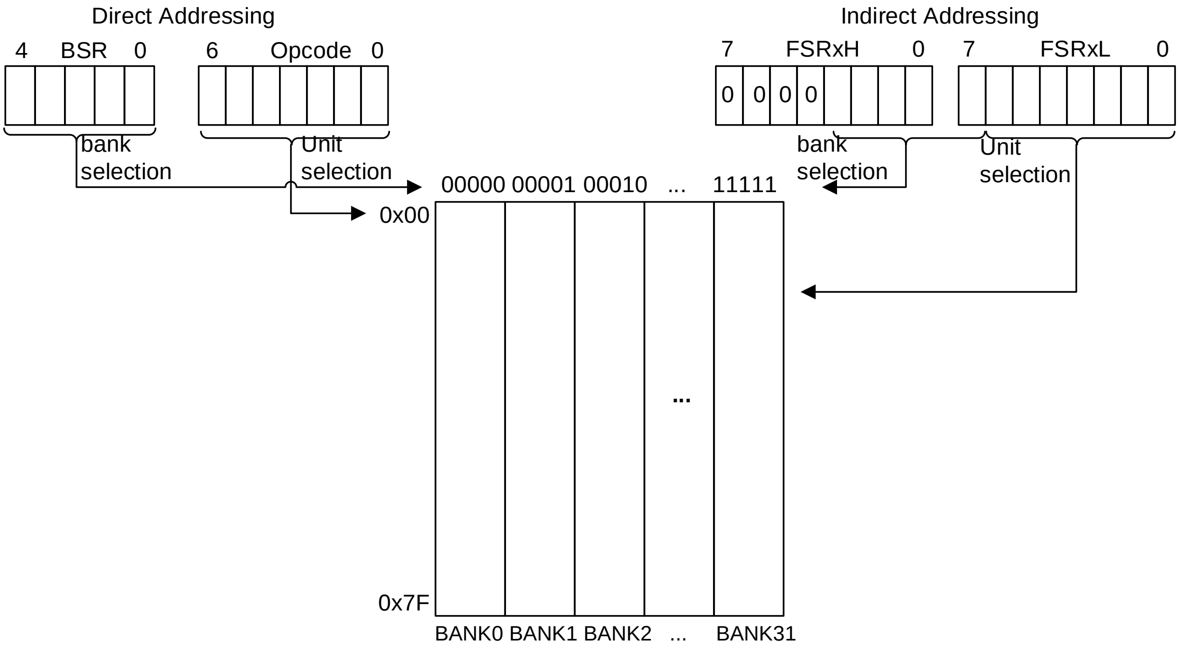


Figure 17-4 Conventional Data Memory Mapping

17.5.2. Linear data memory

Linear data memory with an address range of 0x2000 ~0x29AF, is a virtual array, pointing to the 80-byte SRAM storage area in all Banks (excluding the 16 -byte common SRAM) , the unused memory area (Bank13 ~30) is read as 0x00 .

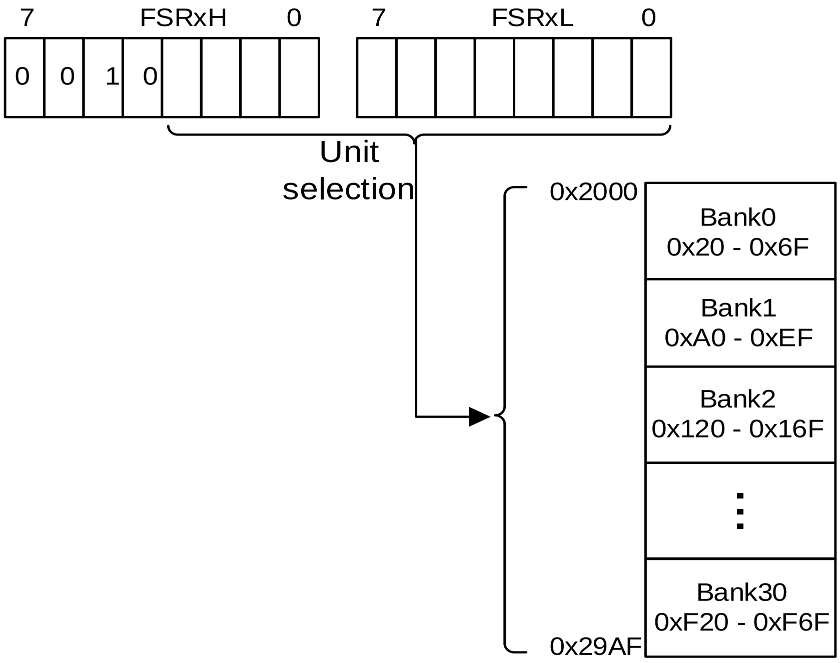


Figure 17-5 Linear Data Memory Mapping

17.5.3. Flash program memory

When the MSB of FSRnH is set to 1, the lower 15 bits of FSRnH:FSRnL are the address of the program PROM memory that needs to be accessed, and the corresponding lower 8-bit data can be read through INDFn. The program PROM cannot be written through FSR/INDF, while the read operation requires 2 instruction cycles.

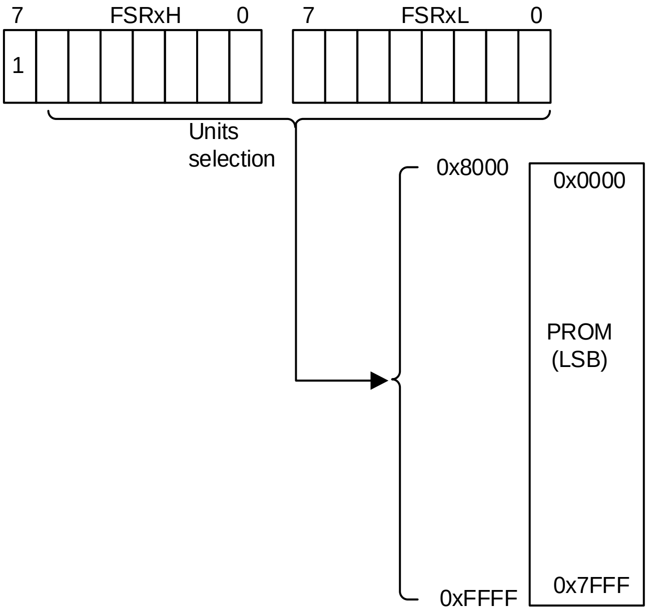


Figure 17-6 Program Memory Mapping

The constants in the program PROM memory can be accessed indirectly via FSR or read via the RETW instruction.

Program example for indirect access via FSR.

```
constants
RETW DATA0              ; Index0 data
RETW DATA1              ; Index1 data
RETW DATA2
RETW DATA3
my_function
…                       ; lots of code…
LDWI LOW constants
STR FSR1L
LDWI HIGH constants
STR FSR1H
MOVIW 0[FSR1]           ; The program memory is in W
```

Program Example to read the constant table with RETW and BRW instructions:

```
constants
BRW                     ;Add Index in W to program counter to ;select data
RETW DATA0              ; Index0 data
RETW DATA1              ; Index1 data
RETW DATA2
RETW DATA3
my_function
…                       ; lots of code…
LDWI DATA_INDEX
call constants          ; the constant is in W
```

18. INSTRUCTION SET

Assembly Syntax	Function	Operation	Instruction cycles	Status
NOP	No operation	None	1	NONE
SLEEP	Enter SLEEP mode	0 → WDT; Stop OSC	1	/PF, /TF
RESET	Software Reset	Reset register PCON	1	NONE
CLRWDT	Clear WDT	0 → WDT	1	/PF, /TF
LJUMP N	Long JUMP of N	N → PC	2	NONE
BRA k	Relative Branch (address range is limited)	PC + 1 + k → PC	2	NONE
BRW	Make a relative jump with the value of register W as an offset	PC + w → PC	2	NONE
LCALL N	Long CALL Subroutine	N → PC; PC + 1 → Stack	2	NONE
CALLW	Call subroutine whose address is specified by register W	W → PC; PC + 1 → Stack	2	NONE
RETW	Transmit immediate I to W and return	I → W, Stack → PC	2	NONE
RETI	Return from Interrupt	Stack → PC; 1 → GIE	2	NONE
RET	Return from Subroutine	Stack → PC	2	NONE
BCR R, b	Clear b bit in register R	0 → R(b)	1	NONE
BSR R, b	Set b bit in register R	1 → R(b)	1	NONE
CLRR R	Clear register R	0 → R	1	Z
LDR R, d (MOVF)	Load register R to d	R → d	1	Z
COMR R, d	Complement Register	/R → d	1	Z
INCR R, d	Increment Register	R + 1 → d	1	Z
INCRSZ R, d	Increment Register, Skip if 0	R + 1 → d	1	NONE
DECR R, d	Decrement Register	R−1 → d	1	Z
DECRSZ R, d	Decrement Register, Skip if 0	R−1 → d	1	NONE
SWAPR R, d	Swap Halves Register	R(0-3)R(4−7) → d	1	NONE
RRR R, d	Rotate Right Register	R(0) → C; R(n) → R(n−1); C → R(7);	1	C
RLR R, d	R moves to the left with a carry loop	R(7) → C; R(n) → R(n+1); C → R(0);	1	C
LSRF f, d	F Logical Right Shift	0→ f(7); f(n+1) → R(n); f(0)→ C;	1	C, Z
LSLF f, d	F logical Left Shift	f(7) → C; f(n) → R(n+1); 0 → R(0);	1	C, Z
ASRF f, d	Arithmetic Right Shift	f(7) → R(7); f(n+1) →R(n); f(0) →C;	1	C, Z

Assembly Syntax	Function	Operation	Instruction cycles	Status
BTSC R, b	Bit Test, Skip if 0	Skip if R(b)=0	1	NONE
BTSS R, b	Bit Test, Skip if 1	Skip if R(b)=1	1	NONE
CLRW	Clear Working Register	0 → W	1	Z
STR R (MOVWF)	Store W to Register	W → R	1	NONE
ADDWR R, d	Add W and Register	W + R → d	1	C, DC, Z
ADDWFC R, d	ADD W and R with CARRY bit	W + R + C → d	1	C, DC, Z
SUBWR R, d	Subtract W from Register	R−W → d	1	C, DC, Z
SUBWFB R, d	Subtract W from Register (with BORROW bit)	R−W – (/B) → d	1	C, DC, Z
ANDWR R, d	AND W and Register	R & W → d	1	Z
IORWR R, d	OR W and Register	W | R → d	1	Z
XORWR R, d	XOR W and register	W^R→d	1	Z
LDWI I (MOVLW)	Load Immediate to W	I → W	1	NONE
ANDWI I	AND W and imm	I & W → W	1	Z
IORWI I	OR W and imm	I | W → W	1	Z
XORWI I	XOR W and imm	I^W → W	1	Z
ADDWI I	Add imm to W	I + W → W	1	C, DC, Z
SUBWI I	Subtract W from imm	I−W → W	1	C, DC, Z
RETW I	Return, Place imm to W	Stack → PC; I → W		NONE
MOVLBk	Move imm I to BSR	K → BSR	1	NONE
ADDFSR FSRn, k	Add imm k to FSRn	FSRn + k → FSRn	1	NONE
MOVLP	Move k to PCLATH	k → PCLATH	1	NONE
MOVIW mm	Move FSRn to W	FSRn → W	1	Z
MOVWI mm	Move W to FSRn	W → FSRn	1	NONE

Table 18-1 49 Instruction Commands

Field	Descriptions	
R(f)	SFR/SRAM Address	
W	Working Register	
b	Bit address within the 8−bit Register / RAM	
I/Imm (k)	Immediate data, constant or label	
x	Don't' care, may be 0 or 1	
d	Destination select	1 = Store result in Register / RAM 0 = Store result in W
mm	Pre/post increment/decrement mode selection (++FSRn, --FSRn, FSRn++ , FSRn--, k[FSRn])	
N	Absolute program address	
PC	Program Counter	
/PF	Power-Down Flag	
/TF	Time-Out Flag	
C	Carry/Borrow bit	
DC	Half Carry/Half Borrow bit	
Z	Zero Flag	

Table 18-2 OpCode Field

Name	Status	Register	Addr.	Reset
Z	Zero Flag Bit: Result of an arithmetic or logic operation is zero 1 = Yes 0 = No	STATUS[2]	Bank First address + 0x03	RW−x
DC	Half Carry (ADDWR, ADDWI, SUBWI, SUBWR): Digit Carry-Over or Borrow from the 4th low-order bit of the result 1 = Carry-Over, Yes; Borrow, No 0 = Carry-Over, No; Borrow, Yes	STATUS[1]		RW−x
C	Carry (ADDWR, ADDWI, SUBWI, SUBWR): Digit Carry-Over or Borrow from MSB of the result 1 = Carry-Over, Yes; Borrow, No 0 = Carry-Over, No; Borrow, Yes	STATUS[0]		RW−x

Table 18-3 Computational Status Flags

18.1. Read-Modify-Write (RMW) Instructions

All instructions that need to use the file register (the instructions with the mnemonic R in **Table 18-1**) will perform the read-modify-write (RMW) operation, that is, first take out the contents of the target register, modify the data according to the instruction, and then write the data back to the target register or W (depending on d and specific instruction) .

For example:

```
    BSR     FSR0L, 0;
```

The execution process of the above instructions in the CPU is as follows:

1) Read out FSR0L to the temporary register T;

2) Set register T or "0000 0001" to form new data;

3) Write the new data back to FSR0L;

18.2. Instruction details

ADDFSR	AND Immediate to FSRn
Syntax:	[label] ADDFSR FSRn, k
Operands:	-32 ≤ k ≤ 31 n ∈ [0,1]
Operation:	FSR(n)+k → FSR(n)
Status Affected:	None
Description:	The signed 6-bit immediate ‘k’ is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h - FFFFh. Moving beyond these bounds will cause the FSR to wrap-around..

ADDWI	Add Immediate to W
Syntax:	[label] ADDWI k
Operands:	0 ≤ k ≤ 255
Operation:	(W)+k → (W)
Status Affected:	C, DC, and Z
Description:	The contents of the W register are added to the 8-bit immediate ‘k’ and the result is placed in the W register.

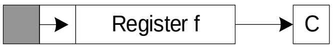
ADDWR	Add W and f
Syntax:	[label]ADDWR f,d
Operands:	0≤f≤127 d ∈ [0,1]
Operation:	(W)+(f)→(destination register)
Status Affected:	C, DC, and Z
Description:	Add the contents of the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

ANDWI	AND Immediate with W
Syntax:	[label] ANDWI k
Operands:	0 ≤ k ≤ 255
Operation:	(W).AND.(k) → (W)
Status Affected:	Z
Description:	Perform a logical AND operation on the contents of the W register and the 8-bit immediate immediate k. The result is stored in the W register.

ANDWR	AND W with f
Syntax:	[label] ANDWR f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W).AND.(f) → (destination register)
Status Affected:	Z
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’.

ASRF Arithmetic Right Shift

Syntax: [label] ASRF f {,d}
Operands: 0≤f≤127
d ∈ [0,1]
Operation: (f[7])→dest[7]
(f[7:1])→dest[6:0],
(f[0])→C
Status Affected: C and Z
Description: The contents of register ‘f’ are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.



ADDWFC Add W to f (with Carry bit)

Syntax: [label] ADDWFC f {,d}
Operands: 0≤f≤127
d ∈ [0,1]
Operation: (W)+(f)+(C)→ destination register
Status Affected: C, DC, and Z
Description: Add the content of W, the carry bit and the content of register f. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored in dregister f

BCR Bit Clear f

Syntax: [label] BCR f, b
Operands: 0≤f≤127
0≤b≤7
Operation: 0→(f [b])
Status Affected: None
Description: Bit ‘b’ in register ‘f’ is cleared.

BRA Relative Branch

Syntax: [label] BRA label
[label] BRA $+k
Operands: -256≤label- PC+1≤255
-256≤k≤255
Operation: (PC)+1+k→PC
Status Affected: None
Description: Add the signed 9-bit immediate ‘k’ to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + k. This instruction is a 2-cycle instruction. This branch has a limited range.

BRW Relative Branch with W

Syntax: [label] BRW
Operands: none
Operation: (PC)+(W)→PC
Status Affected: None
Description: Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + (W). This instruction is a 2-cycle instruction.

BSR Bit Set f

Syntax: [label] BSR f,b
Operands: 0≤f≤127
0≤b≤7
Operation: 1→(f [b])
Status Affected: None
Description: Bit ‘b’ in register ‘f’ is set.

BTSC	Bit Test f, Skip if Clear
Syntax:	[label] BTSC f,b
Operands:	0≤f≤127 0≤b≤7
Action:	Skip if (f[b])=0
Status Affected:	None
Description:	If bit ‘b’ in register ‘f’ is ‘1’, the next instruction is executed. If bit ‘b’, in register ‘f’, is ‘0’, the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

BTSS	Bit Test f, Skip if 1
Syntax:	[label] BTSS f,b
Operands:	0≤f≤127 0≤b≤7
Action:	Skip if (f[b])=1
Status Affected:	None
Description:	If bit b of register f is 0, execute the next instruction. If bit b is 1, the next instruction is discarded and a NOP is executed instead, making the instruction a two-cycle instruction.

LCALL	Long CALL Subroutine
Syntax:	[label] LCALL k
Operands:	0≤k≤2047
Operation:	(PC)+1→TOS, k→PC[10:0], (PCLATH[4:3])→PC[12:11]
Status Affected	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. LCALL is a 2-cycle instruction.

CLRR	Clear f
Syntax:	[label] CLRR f
Operands:	0≤f≤127
Operation:	00h→(f) 1→Z
Status Affected:	Z
Description:	The contents of register ‘f’ are cleared and the Z bit is set.

CLRW	Clear W
Syntax:	[label] CLRW
Operands:	none
Operation:	00h→(W) 1→Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CALLW	Subroutine Call With W
Syntax:	[label] CALLW
Operands:	none
Operation:	(PC)+1→TOS, (W)→PC[7:0], (PCLATH[6:0])→PC[14:8]
Status Affected:	None
Description:	Subroutine call with W. First, the return address (PC + 1) is pushed onto the return stack. Then, the contents of W is loaded into PC<7:0>, and the contents of PCLATH into PC<14:8>. CALLW is a two-cycle instruction.

CLRWDT	Clear Watchdog Timer
Syntax:	[label] CLRWDT
Operands:	none
Operation:	00h→WDT 0→WDT prescaler 1→/TO 1→/PD
Status Affected:	/TO and /PD
Description:	CLRWDT instruction resets the Watchdog Timer and its prescaler.Status bits /TO and /PD are both set.

COMR	Complement f
Syntax:	[label] COMR f,d
Operands:	0≤f≤127 d ∈ [0,1]
Operation:	$(\overline{f})$→(destination register)
Status Affected:	Z
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’.

DECR	Decrement f
Syntax:	[label] DECR f,d
Operands:	0≤f≤127 d ∈ [0,1]
Operation:	(f)-1→(destination register)
Status Affected:	Z
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’.

DECRSZ	Decrement f, Skip if 0
Syntax:	[label]DECRSZ f,d
Operands:	0≤f≤127 d ∈ [0,1]
Operation:	(f)-1→(destination register); skip if result = 0
Status Affected:	None
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.

LJUMP	Long JUMP Address
Syntax:	[label] LJUMP k
Operands:	0≤k≤2047
Operation:	k→PC [10:0] PCLATH[4:3]→PC[12:11]
Status Affected:	None
Description:	LJUMP is an Long JUMP of N instruction. Loads bits[10:0] of the PC with an 11-bit immediate value. The upper bits of the PC are loaded from PCLATH[4:3]. LJUMP is a 2-cycle instruction.

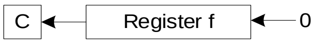
INCR	Increment f
Syntax:	[label] INCR f,d
Operands:	0≤f≤127 d ∈ [0,1]
Operation:	(f)+1→(destination register)
Status Affected:	Z
Description:	The contents of register ‘f’ are incremented. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed back in register ‘f’.

INCRSZ	Increment f, Skip if 0
Syntax:	[label] INCRSZ f,d
Operands:	0≤f≤127 d ∈ [0,1]
Operation:	(f)+1→(destination register), result=0 skip
Status Affected:	None
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..

IORWI	Inclusive OR immediate with W
Syntax:	[label] IORWI k
Operands:	0≤k≤255
Operation:	(W).OR.k→(W)
Status Affected:	Z
Description:	The contents of the W register are OR’ed with the 8-bit immediate ‘k’. The result is placed in the W register.

IORWR	Inclusive OR W with f
Syntax:	[label] IORWR f,d
Operands:	0≤f≤127 d ∈ [0,1]
Operation:	(W).OR.(f)→(destination register)
Status Affected:	Z
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’.

LSLF	Logical Left Shift
Syntax:	[label] LSLF f {,d}
Operands:	0≤f≤127 d ∈ [0,1]
Operation:	(f [7])→C (f[6:0])→dest[7:1] 0→dest[0]
Status Affected:	C and Z
Description:	The contents of register ‘f’ are shifted one bit to the left through the Carry flag. A ‘0’ is shifted into the LSb. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.



LSRF Logical Right Shift

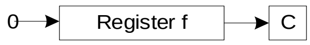
Syntax: [label] LSRF f {,d}

Operands: 0≤f≤127
d ∈ [0,1]

Operation: 0→ destination register[7]
(f[7:1])→destination register[6:0]
(f[0])→C

Status Affected: C and Z

Description: The contents of register ‘f’ are shifted one bit to the right through the Carry flag. A ‘0’ is shifted into the MSb. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.



LDR Move f

Syntax: [label] LDR f,d

Operands: 0≤f≤127
d ∈ [0,1]

Operation: (f)→(destination register)

Status Affected: Z

Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example: LDR FSR, 0
After Instruction
W = value in FSR register
Z = 1

MOVIW Move INDFn to W

Syntax: [label] MOVIW ++FSRn
[label] MOVIW --FSRn
[Label] MOVIW FSRn++
[Label] MOVIW FSRn--
[label] MOVIW k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01,10,11]
-32≤k≤31

Operation: INDFn→W
Valid address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers(INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h-FFFFh. Incrementing /decrementing it beyond these bounds will cause it to wrap-around.

MOVWI Move INDFn to W

Syntax: [label] MOVWI ++FSRn
[label] MOVWI --FSRn
[label] MOVWI FSRn++
[Label] MOVWI FSRn--
[label] MOVWI k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01,10,11]
-32≤k≤31

Operation: W→INDFn
Valid address is determined by:
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: None

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h-FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

MOVLB Move Immediate to BSR

Syntax: [label] MOVLB k
Operands: 0≤k≤15
Operation: k→BSR
Status Affected: None
Description: The 5-bit immediate ‘k’ is loaded into the Bank Select Register (BSR).

MOVLP Move Immediate to PCLATH

Syntax: [label] MOVLP k
Operands: 0≤k≤127
Operation: k→PCLATH
Status Affected: None
Description: The seven-bit immediate ‘k’ is loaded into the PCLATH register.

LDWI Move Immediate to W

Syntax: [label] MOVLW k
Operands: 0≤k≤255
Operation: k→(W)
Status Affected: None
Description: The eight-bit immediate ‘k’ is loaded into W register. The “don’t cares” will assemble as ‘0’s.
Words: 1
Cycles: 1
Example: LDWI 0x5A
After Instruction
W = 0x5A

NOP No operation

Syntax: [label] NOP
Operands: none
Action: no-op
Status Affected: None
Description: No operation.
Words: 1
Cycles: 1
Example: NOP

STR Move W to f

Syntax: [label] STR f
Operands: 0≤f≤127
Operation: (W) →(f)
Status Affected: None
Description: Transmit the data of the W register to the register f.
Words: 1
Cycles: 1
Example: STR OPTION
Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

RESET Software Reset

Syntax: [label] RESET
Operands: none
Action: Execute a device Reset. Resets the nRI flag of the PCON register.
Status Affected: None
Description: This instruction provides a way to execute a hardware Reset by software.

RET Return from Subroutine

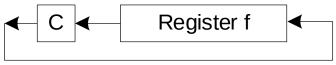
Syntax: [label] RET
Operands: none
Operation: TOS→PC
Status Affected: None
Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction.

RETI Return from Interrupt

Syntax: [label] RETI
Operands: none
Operation: TOS→PC,
1→GIE
Status Affected: None
Description: Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.
Words: 1 Cycles: 2
Example: RETI
After interruption
PC = TOS
GIE = 1

RLR Rotate Left f through Carry

Syntax: [label] RLR f,d
Operands: 0≤f≤127
d ∈ [0,1]
Operation: See description below
Status Affected: C
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’.
Words: 1 Cycles: 1
Example: RLF REG1,0
Before Instruction:
REG1 = 1110 0110
C = 0
After Instruction:
REG1 = 1110 0110
W = 1100 1100
C = 1



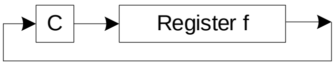
RETW Return with Immediate in W

Syntax: [label] RETW k
Operands: 0≤k≤255
Operation: k→(W) ;
TOS→PC
Status Affected: None
Description: The W register is loaded with the 8-bit immediate ‘k’. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.
Words: 1
Cycles: 2
Example:

```
        LCALL TABLE;W contains table
        ;offset value
        • ;W now has table value
TABLE   •
        ADDWR PC ;W = Offset
        RETW k1 ;Begin table
        RETW k2 ;
        •
        •
        RETW kn ;End of table
        Before Instruction
        W = 0x07
        After Instruction
        W = value of k8
```

RRR Rotate Right f through Carry

Syntax: [label] RRR f,d
Operands: 0≤f≤127
d ∈ [0,1]
Operation: See description below
Status Affected: C
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’.



SLEEP Enter Sleep mode

Syntax: [label] SLEEP
Operands: none
Operation: 00h→WDT,
0→WDT prescaler,
1→/TO,
0→/PD
Status Affected: /TO and /PD
Description: The power-down status bit, /PD is cleared. Time-out Status bit, /TO is set. Watchdog Timer and its prescaler are cleared.The processor enter Sleep mode with the oscillator stopped.

SUBWI Subtract W from litera

Syntax: [label] SUBWI k
Operands: 0≤k≤255
Operation: k-(W)→(W)
Status Affected: C, DC, and Z
Description: The W register is subtracted (2’s complement method) from the 8-bit immediate ‘k’. The result is placed in the W register.

C=0	W>k
C=1	W ≤ k
DC=0	W[3:0] > k[3:0]
DC=1	W[3:0] ≤ k[3:0]

SUBWFB Subtract W from f with Borrow

Syntax: SUBWFB f {,d}
Operands: 0≤f≤127
d ∈ [0,1]
Operation: (f)–(W)–(/B)→ destination register
Status Affected: C, DC, and Z
Description: Subtract W and the BORROW flag (CARRY) from register ‘f’ (2’s complement method). If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

SWAPR Swap two half bytes in f

Syntax: [label] SWAPR f,d

Operands: 0≤f≤127
d ∈ [0,1]

Operation: (f [3:0])→(destination[7:4]),
(f[7:4])→(destination[3:0])

Status Affected: None

Description: The upper and lower half bytes of register 'f' are exchanged. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is placed in register ‘f’.

SUBWR Subtract W from f

Syntax: [label] SUBWR f,d

Operands: 0≤f≤127
d ∈ [0,1]

Operation: (f)-(W)→(destination register)

Status Affected: C, DC, and Z

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’s stored back in register ‘f.

C=0	W>f
C=1	W ≤ f
DC=0	W[3:0] > f[3:0]
DC=1	W[3:0] ≤ f[3:0]

XORWR Exclusive OR W with f

Syntax: [label] XORWR f,d

Operands: 0≤f≤127
d ∈ [0,1]

Operation: (W).XOR.(f)→(destination register)

Status Affected: Z

Description: Perform a logical XOR operation on the contents of the W register and the contents of the 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.

XORWI Exclusive OR immediate with W

Syntax: [label] XORWI k

Operands: 0≤k≤255

Operation: (W).XOR.k→(W)

Status Affected: Z

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’.

19. ELECTRICAL SPECIFICATIONS

19.1. Limit Parameters

Operation temperature Grade 3	$^{-}$40 – $^{+}$85 °C
Operation temperature Grade 2	$^{-}$40 – $^{+}$105 °C
Operation temperature Grade 1	$^{-}$40 – $^{+}$125 °C
Storage temperature	$^{-}$40 – $^{+}$125 °C
Junction operation temperature (Tj)	$^{-}$40 – $^{+}$150 °C
Power supply voltage	V_{SS}-0.3V – V_{SS}+6.0V
PAD input voltage	V_{SS}-0.3V – V_{DD}+0.3V

Notes:

1. Stresses above “Limit Parameters” may cause permanent damages to the device.
2. All characterizations are at 25 °C, V_{DD} =1.9 – 5.5V unless otherwise stated.
3. Values and ranges indicated are from characterizations, and are not indicative of the final shipping criteria.
4. Prodution test are at 25 °C unless otherwise stated. Performance at temperature outside of above operation temperature are not guaranteed as high temperature screening is not part of normal production procedure.
5. Typical unstressed memory data retention @ 150 °C > 10 years.

19.2. Operation Characteristics

Parameters		Min	Typical	Max	Units	Conditions
Fsys (SysClk)	1T/ 2T/4T	–	–	8	MHz	$^{-}$40 – $^{+}$85 °C, V_{DD} = 1.9 – 5.5V
		–	–	16	MHz	$^{-}$40 – $^{+}$85 °C, V_{DD} = 2.7 – 5.5V
Instruction Cycle ($T_{INSTRCLK}$)	1T	–	62.5	–	ns	SysClk = HIRC
	2T	–	125	–	ns	
	4T	–	250	–	ns	
	1T	–	30.5	–	μs	SysClk = LIRC
	2T	–	61	–	μs	
	4T	–	122	–	μs	
Power-On-Reset hold time (T_{DRH})		–	–	–	–	25 °C, PWRT disable
Ext. Reset pulse width (T_{MCLRB})		2000	–	–	–	25 °C
WDT period (T_{WDT})		–	1	–	–	No prescaler, WDTPS[3:0]=0000

19.3. POR , LVR, LVD

Power-On Reset (POR)

Parameters	Min	Typical	Max	Units	Conditions
I_{POR} Operating Current	–	0.14	–	μA	25 °C , V_{DD} = 3.3V
V_{POR}	–	1.65	–	V	25 °C

Low Voltage Reset (LVR)

Parameters	Min	Typical	Max	Units	Conditions
I_{LVR} Operating Current	–	15.2	–	μA	25 °C, V_{DD} = 3.3V
V_{LVR}, LVR threshold	1.94	2.0	2.06	V	25 °C
	2.13	2.2	2.27		
	2.42	2.5	2.58		
	2.72	2.8	2.88		
	3.01	3.1	3.19		
	3.49	3.6	3.71		
	3.98	4.1	4.22		
LVR delay	94	–	125	μs	25 °C, V_{DD} = 1.9 – 5.5V

Low Voltage Detection (LVD)

Parameters	Min	Typical	Max	Units	Conditions
I_{LVD} Operating Current	–	21.5	–	μA	25 °C, V_{DD} = 3.3V
V_{LVD}, LVD threshold	1.94	2.0	2.06	V	25 °C
	2.33	2.4	2.47		
	2.72	2.8	2.88		
	2.91	3.0	3.09		
	3.49	3.6	3.71		
	3.88	4.0	4.12		
LVD delay	94	–	125	μs	25 °C, V_{DD} = 1.9 – 5.5V

19.4. I/O PORTS

Parameters		Min	Typical	Max	Units	Conditions
V_{IL}		0	—	0.3*V_{DD}	V	
V_{IH}		0.7*V_{DD}	—	V_{DD}	V	
Leakage current		-1	—	1	μA	V_{DD} = 5V
Source Current	L0	–	-2	–	mA	25 °C, V_{DD} = 5V, V_{OH} = 4.5V
	L1	–	-4	–		
	L2	–	-14	–		
	L3	–	-26	–		
Sink Current	L0	–	53	–	mA	25 °C, V_{DD} = 5V, V_{OL}= 0.5V
	L1	–	62	–		
Pull-Up resistor		–	21	–	kΩ	
Pull-Down resistor		–	21	–	kΩ	

19.5. Operating Current (I_{DD})

parameter	Sysclk	Typical value @V_{DD}			Units
		2.0V	3.0V	5.5V	
Normal mode (1T) - I_{DD}	16MHz	—	4.143	4.402	mA
	8MHz	1.897	2.648	2.808	
	4MHz	1.293	1.887	1.981	
	2MHz	0.871	1.130	1.183	
	1MHz	0.561	0.727	0.755	
	32kHz	0.036	0.051	0.054	
Normal mode (2T) - I_{DD}	16MHz	2.170	3.000	3.181	mA
	8MHz	1.435	2.074	2.169	
	4MHz	0.947	1.224	1.284	
	2MHz	0.596	0.778	0.810	
	1MHz	0.420	0.560	0.581	
	32kHz	0.032	0.046	0.048	
Sleep mode (~~WDT OFF~~, ~~LVR OFF)~~ , I_{SB}	–	0.087	0.136	0.240	μA
Sleep mode (WDT ON, ~~LVR OFF)~~	32kHz	1.294	2.420	2.854	
Sleep mode (~~WDT OFF~~ LVR ON)	–	11.257	15.318	20.777	
Sleep mode (WDT ON, LVR ON)	32kHz	12.457	17.551	23.240	
Sleep mode (~~WDT OFF~~, ~~LVR OFF~~, LVD ON)	–	17.793	21.672	27.133	

Note: Test conditions for ISB in sleep mode with all I/Os set to input mode and external pull-down to GND.

19.6. Internal Oscillators

Internal Low Frequency Oscillator (LIRC)

LIRC is set at 32 kHz during measurement (LFMOD=0)..

Parameters	Min	Typical	Max	Units	Conditions
Range	30.4	32	33.6	kHz	25 °C, V_{DD} = 2.5V
temperature dependence	-2.0%	–	2.0%	–	$^{-}$40 – $^{+}$85 °C, V_{DD} = 2.5V
supply voltage variation	-4.5%	–	1.0%	–	25 °C, V_{DD} = 1.9 ~ 5.5V
I_{LIRC} Operating Current	–	1.3	–	μA	25 °C, V_{DD} = 3.0V
Start up Time	–	4.6	–	μs	25 °C, V_{DD} = 3.0V

Internal High Frequency Oscillator (HIRC)

Parameters	Min	Typical	Max	Units	Conditions
Range	15.84	16	16.16	MHz	25 °C, V_{DD} = 2.5V
temperature dependence	-2.0%	–	2.0%	–	$^{-}$40 – $^{+}$85 °C, V_{DD} = 2.5V
supply voltage variation	-0.5%	–	0.5%	–	25 °C, V_{DD} = 1.9 ~ 5.5V
I_{HIRC} Operating Current	–	40	–	μA	25 °C, V_{DD} = 3.0V
Start up time	–	2.5	–	μs	25 °C, V_{DD} = 3.0V

19.7. ADC (12bit) and ADC VREF

ADC (12bit)

Parameters	Min	Typical	Max	Units	Conditions
Operating Voltage V_{DD}	2.7	–	5.5	V	
Operating Current I_{VDD}	–	630	–	μA	V_{REF}+ = V_{DD} = 2.7V
	–	750	–	μA	V_{REF}+ = V_{DD} = 3.0V
	–	1350	–	μA	V_{REF}+ = V_{DD} = 5.5V
Analog input voltage V_{AIN}	V_{REF}−	–	V_{REF}+	V	
External Reference Voltage V_{REF}	–	–	VDD	V	
Resolution	–	–	12	bit	
Integral error E_{IL}	–	±1.5	–	LSB	V_{REF+} = V_{DD} = 5.0V, V_{REF-} = GND, F_{ADCLK} = 250kHz
Differential error E_{DL}	–	±1.5	–	LSB	
Offset error E_{OFF}	–	±1.0	–	LSB	V_{REF+} = V_{DD} = 5.0V, V_{REF-} = GND, Software calibrated , F_{ADCLK} = 250kHz
Gain error E_{GN}	–	±2.0	–	LSB	
Conversion clock cycle T_{AD}	–	0.5	–	μs	V_{REFP} > 3.0V, V DD $_{>}$ 3.0V
Conversion clock cycles	–	18	–	T_{AD}	
Settling time (T_{ST})	–	15	–	μs	
Sample time (T_{ACQ})	–	≥ 0.5	–	μs	
Analog Voltage Source Impedance (ZAI)	–	–	10	kΩ	(recommend)

Differential Error DNL

Typical DNL Error (LSB) @ V_{DD} = 5 V				
V_{REF+} / F_{ADCLK}	0.5	2	3	V_{DD}
≤ 500kHz	±3.0	±2.0	±2.0	±1.5
1 MHz	±3.0	±2.5	±2.0	±1.5
2 MHz	±4.0	±3.0	±2.0	±1.5
4 MHz	±5.5	±3.0	±2.5	±2.0
8MHz	–	–	–	±2.0

Integral Error INL

Typical INL Error (LSB) @ V_{DD} = 5 V				
V_{REF+} / F_{ADCLK}	0.5	2	3	V_{DD}
≤ 500kHz	±9.0	±3.0	±2.0	±1.5
1 MHz	±9.0	±3.0	±2.5	±2.0
2 MHz	±9.5	±3.0	±2.5	±2.0
4 MHz	±10.0	±3.5	±2.5	±2.0
8MHz	–	–	–	±2.5

Note: When the internal reference voltage $V_{ADC-REF}$ = 0.5V is selected, F_{ADCLK} =32 kHz is recommended, at this time DNL is ±2.5 LSB , INL is ±8.5 LSB , the offset error is ±2.0 LSB ; if F_{ADCLK} = 250 kHz, its offset error is ±9.0 LSB.

ADC V_{REF}

Parameters		Min	Typical	Max	Units	Conditions
Internal reference voltage $V_{ADC-REF}$	$V_{ADC-REF}$ = 0.5V	0.492	0.5	0.508	V	
	$V_{ADC-REF}$ = 2.0V	1.990	2	2.010	V	
	$V_{ADC-REF}$ = 3.0V	2.985	3	3.015	V	
Settling time T_{VRINT}	$V_{ADC-REF}$ = 0.5V	–	400	–	μs	
		–	600	–	μs	C_{EXT} = 1μF
	$V_{ADC-REF}$ = 2.0V	–	450	–	μs	
		–	800	–	μs	C_{EXT} = 1μF
	$V_{ADC-REF}$ = 3.0V	–	450	–	μs	
		–	1200	–	μs	C_{EXT} = 1μF

Note:

1. Typical values are tested at 25 °C, V_{DD} = 5.0V, unless otherwise noted.
2. C_{EXT} is the external capacitor connected to the internal reference voltage $V_{ADC-REF}$ (when ADPREF or ADNREF is configured to 10, see **Table 11-3**).

19.8. Program and Data EEPROM

Parameters		Min	Typical	Max	Units	Conditions
$V_{DD\text{-}READ}$	Program/Data EE read voltage	V_{POR}	–	5.5	V	-40 – 85 / 105 °C
$V_{DD\text{-}WRITE}$	Program EE write voltage	2.7	–	5.5	V	-40 – 85 / 105 °C
	Data EE write voltage	1.9	–	5.5		
N_{END}	Program EE erase/write cycles	100k	–	–	cycle	25 °C
		40k	–	–		85 °C
		10k	–	–		105 °C
	Data EE erase/write cycles	1,000k	–	–		25 °C
		400k	–	–		85 °C
		100k	–	–		105 °C
T_{RET}	Program EE data retention	20	–	–	year	After 1k cycles @ 85 °C
		10	–	–		After 1k cycles @ 105 °C
	Data EE data retention	20	–	–		After 10k cycles @ 85 °C
		10	–	–		After 10k cycles @ 105 °C
T_{WRITE}	Data EE write time	–	4.0	–	ms	Auto-Erase enable
		–	2.0	–		Auto-Erase disable
I_{PROG}	Data EE programming current	–	7 00	–	μA	25 °C, V_{DD} = 3 V, 16MHz / 1T
		–	500	–		25 °C, V_{DD} = 3 V, 16MHz / 2T

19.9. EMC characteristics

ESD

Parameters		Min	Typical	Max	Units	Conditions
V_{ESD}	HBM	8000	–	–	V	MIL-STD-883H Method 3015.8
V_{ESD}	MM	400	–	–	V	JESD22-A115

Latch-up

Parameters	Min	Typical	Max	Units	Conditions
LU, static latch-up	200	–	–	mA	EIA/JESD78

EFT

Parameters	Min	Typical	Max	Units	Conditions
V_{EFT}	5.5	–	–	kV	V_{DD} (5V) and GND: 1μF

20. Characterization Graphs

Note: The characterization graphs are based on the feature values and are for reference only and have not been tested in production.

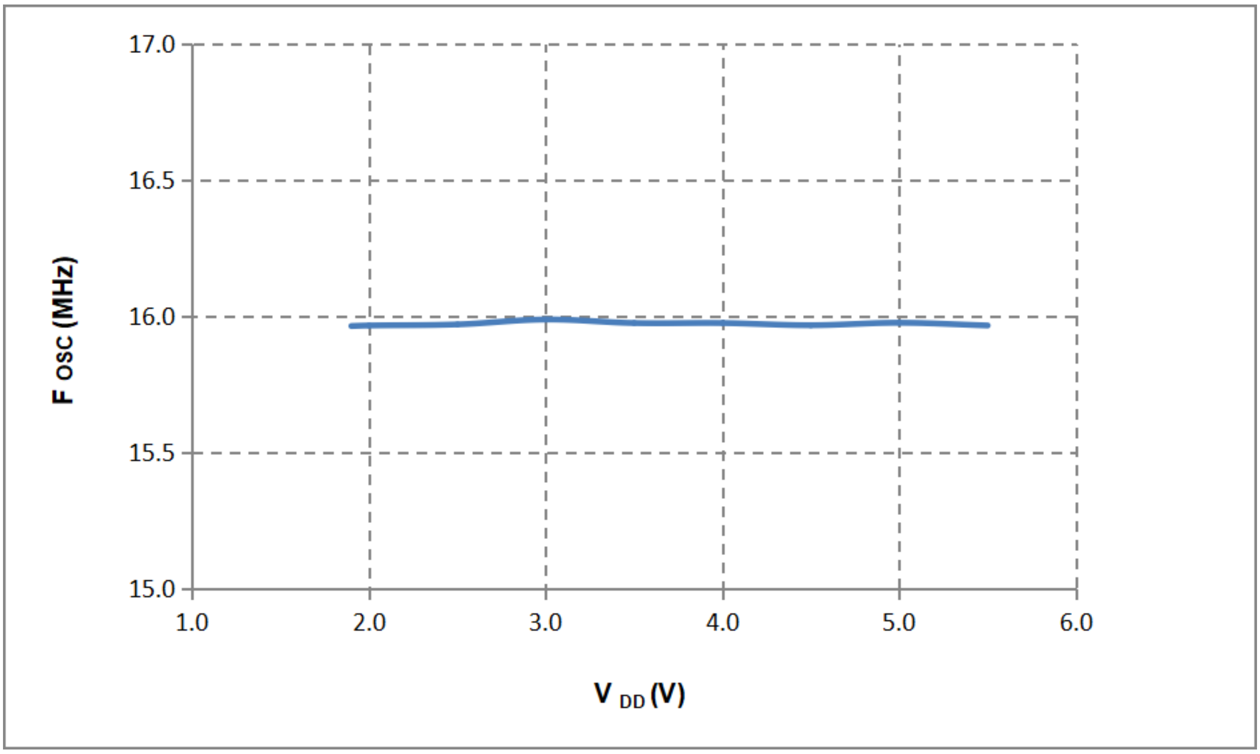


Figure 20-1 HIRC vs. V_{DD} (T_A = 25°C)

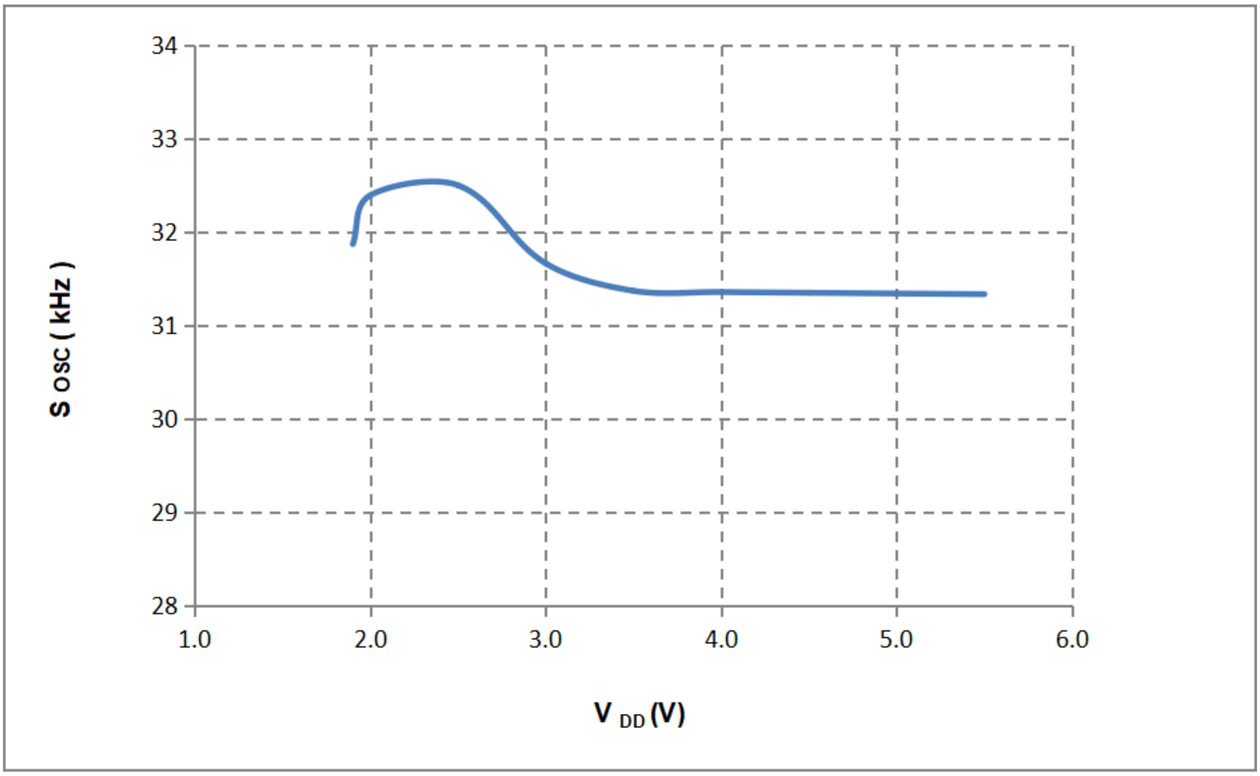


Figure 20-2 LIRC vs. V_{DD} (T_A = 25°C)

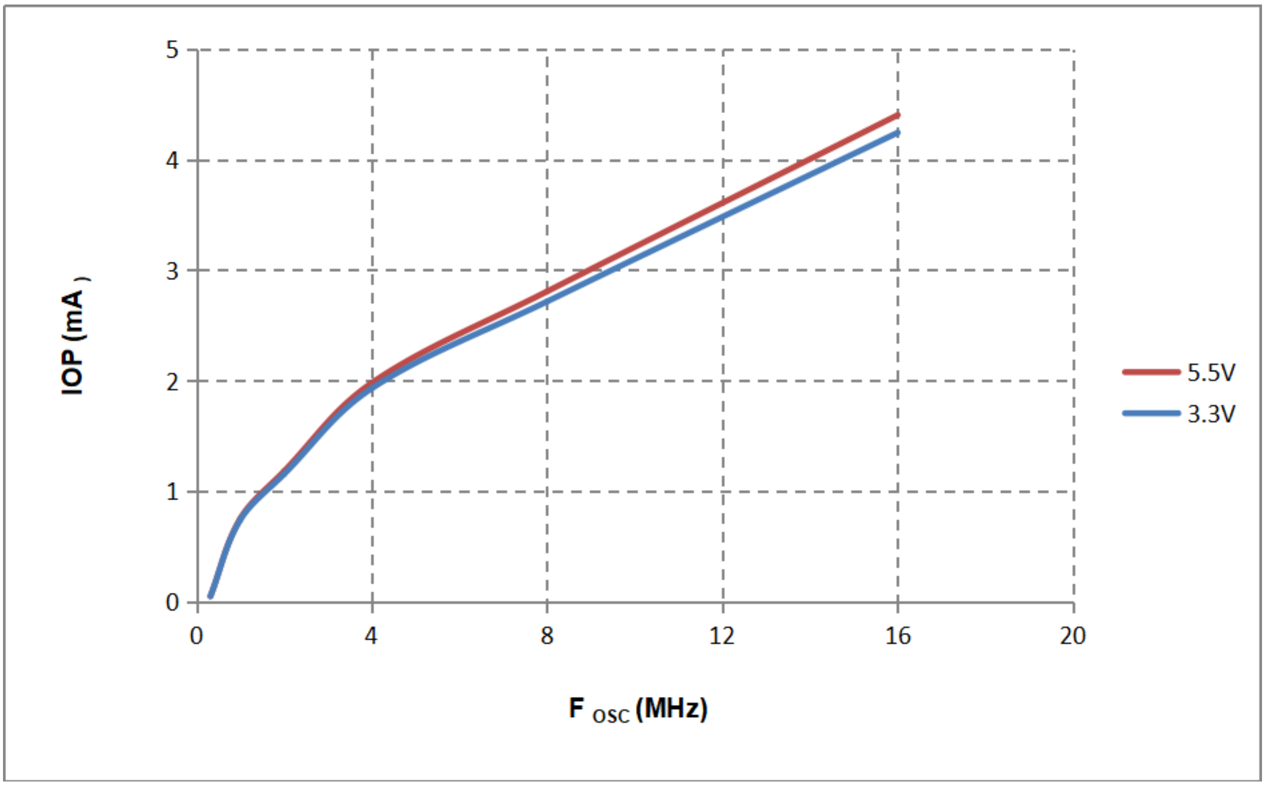


Figure 20-3 I_{DD} vs Frequency (1T, T_A=25°C)

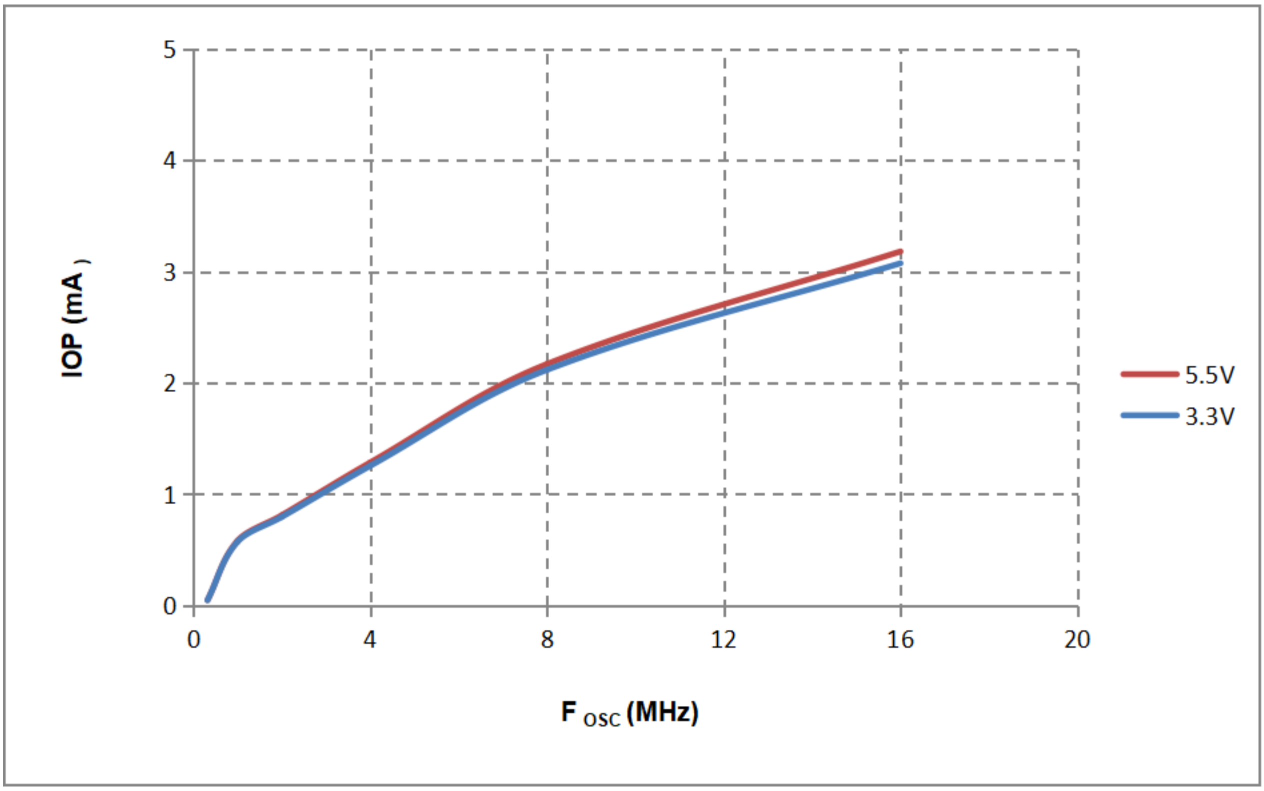


Figure 20-4 I_{DD} vs Freq (2T, T_A=25°C)

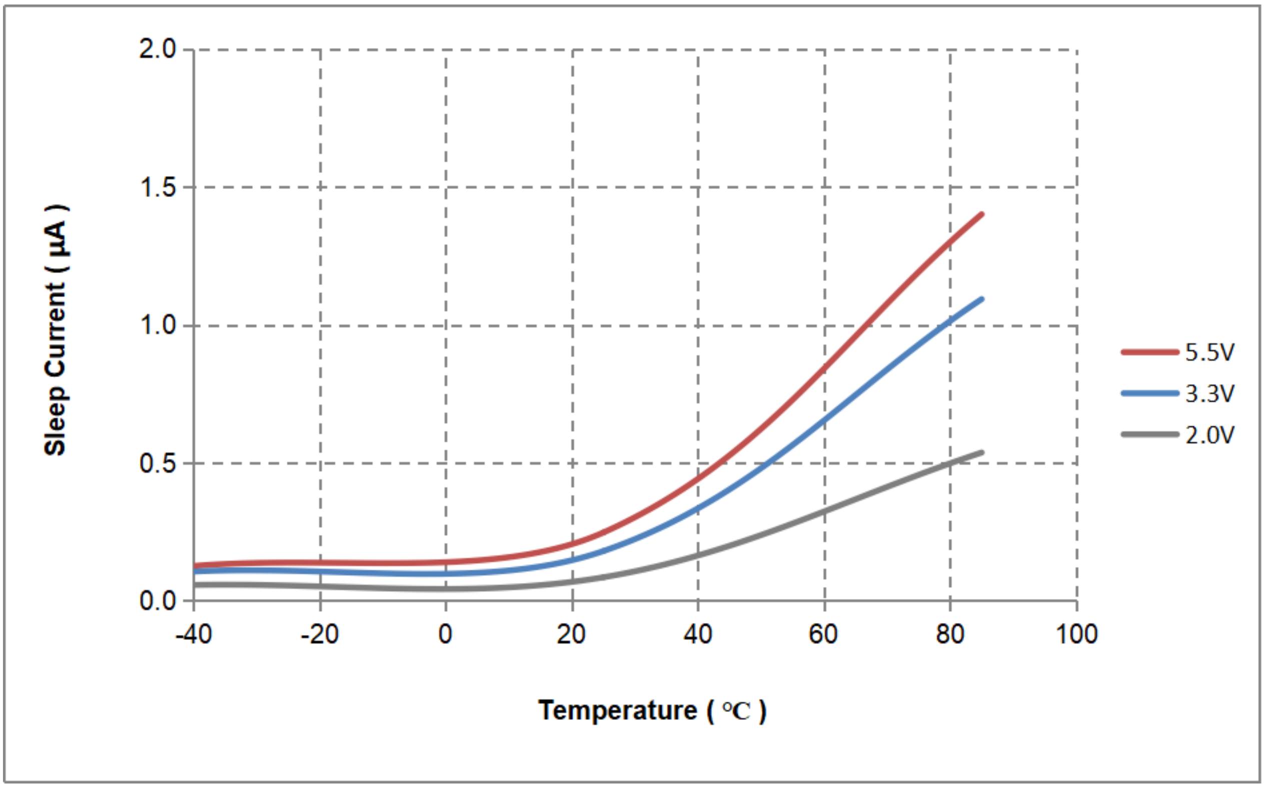


Figure 20-5 Sleep Current ($I_{SB)}$ vs. Temperature

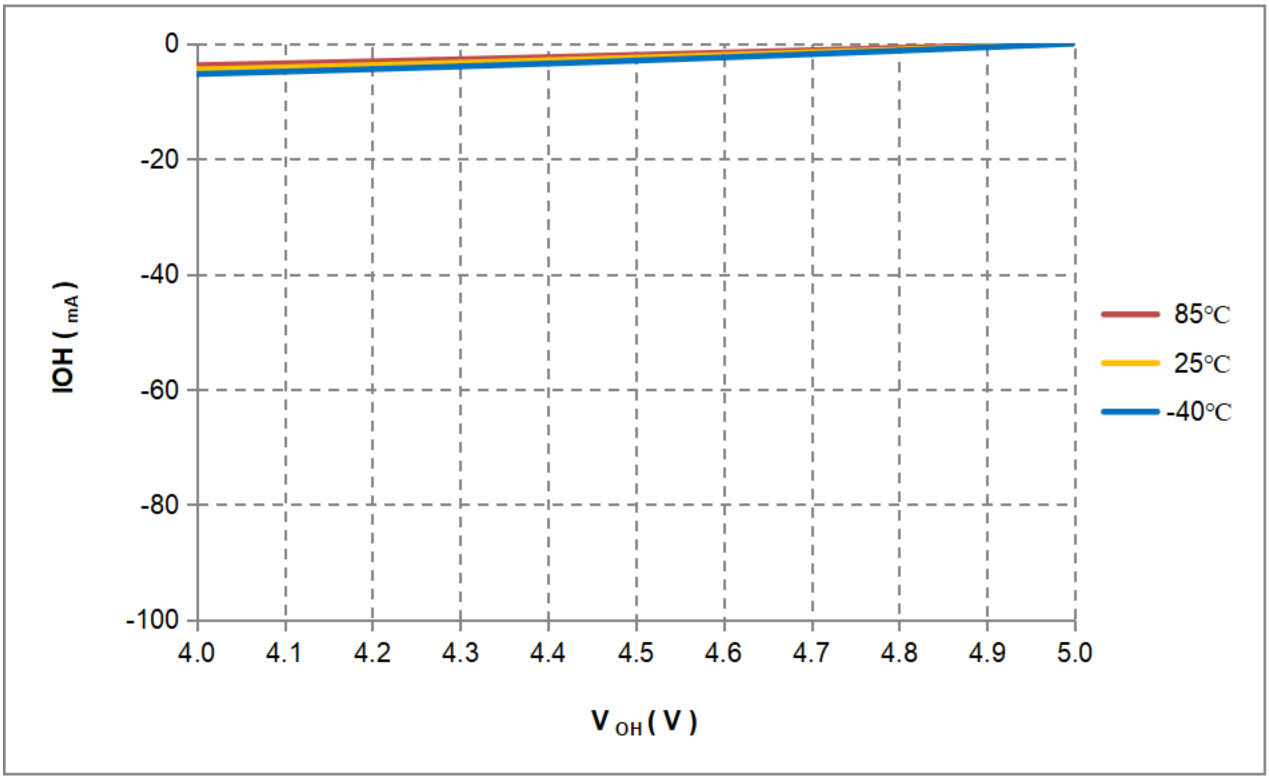


Figure 20-6 I_{OH} vs V_{OH} @L0 = -2mA , V_{DD} = 5V

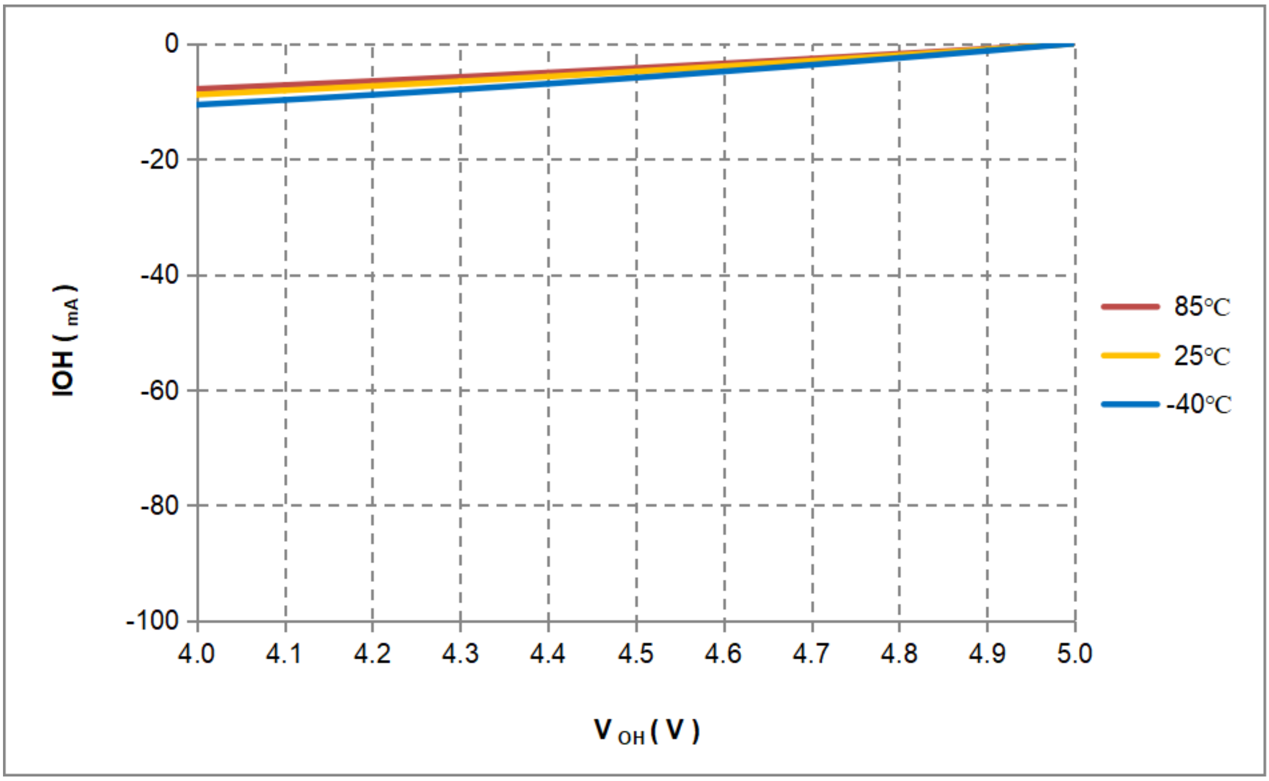


Figure 20-7 I_{OH} vs V_{OH} @L1 = -4mA , V_{DD} = 5V

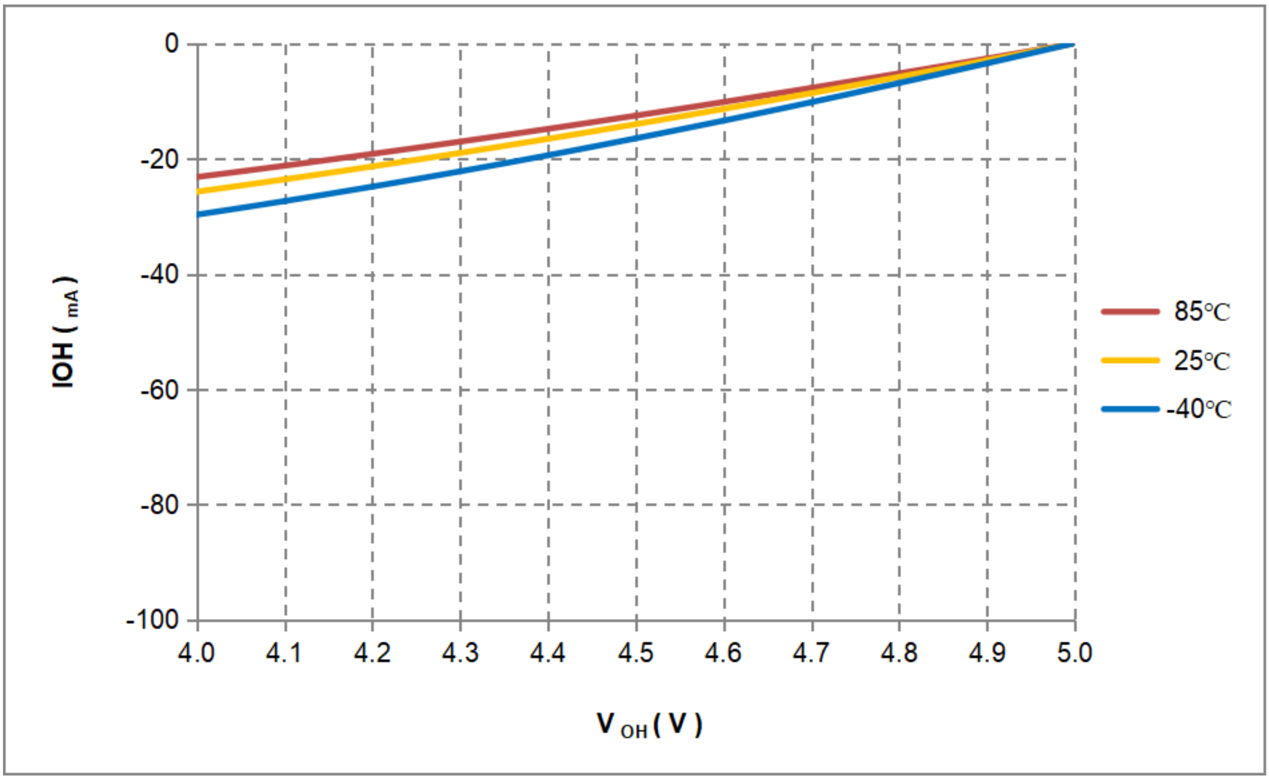


Figure 20-8 I_{OH} vs V_{OH} @L2 = -14mA , V_{DD} = 5V

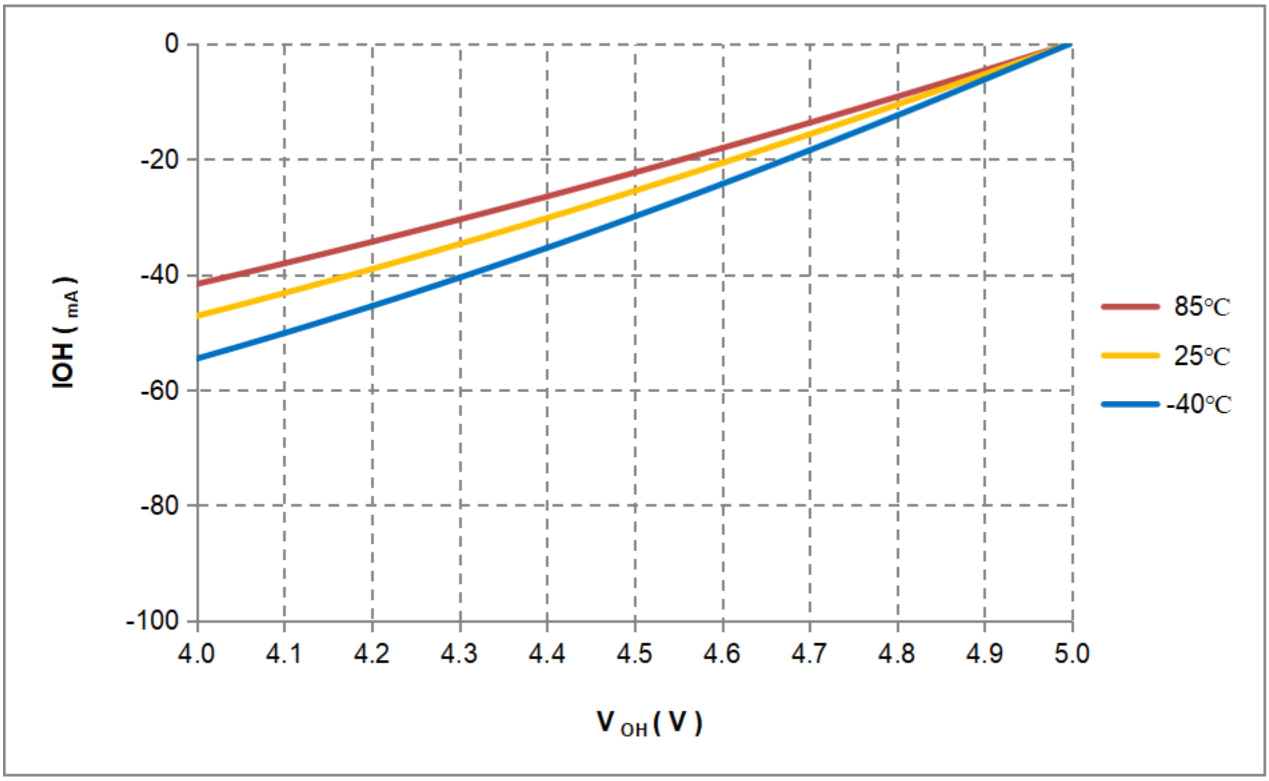


Figure 20-9 I_{OH} vs V_{OH} @L3 = -26mA , V_{DD} = 5V

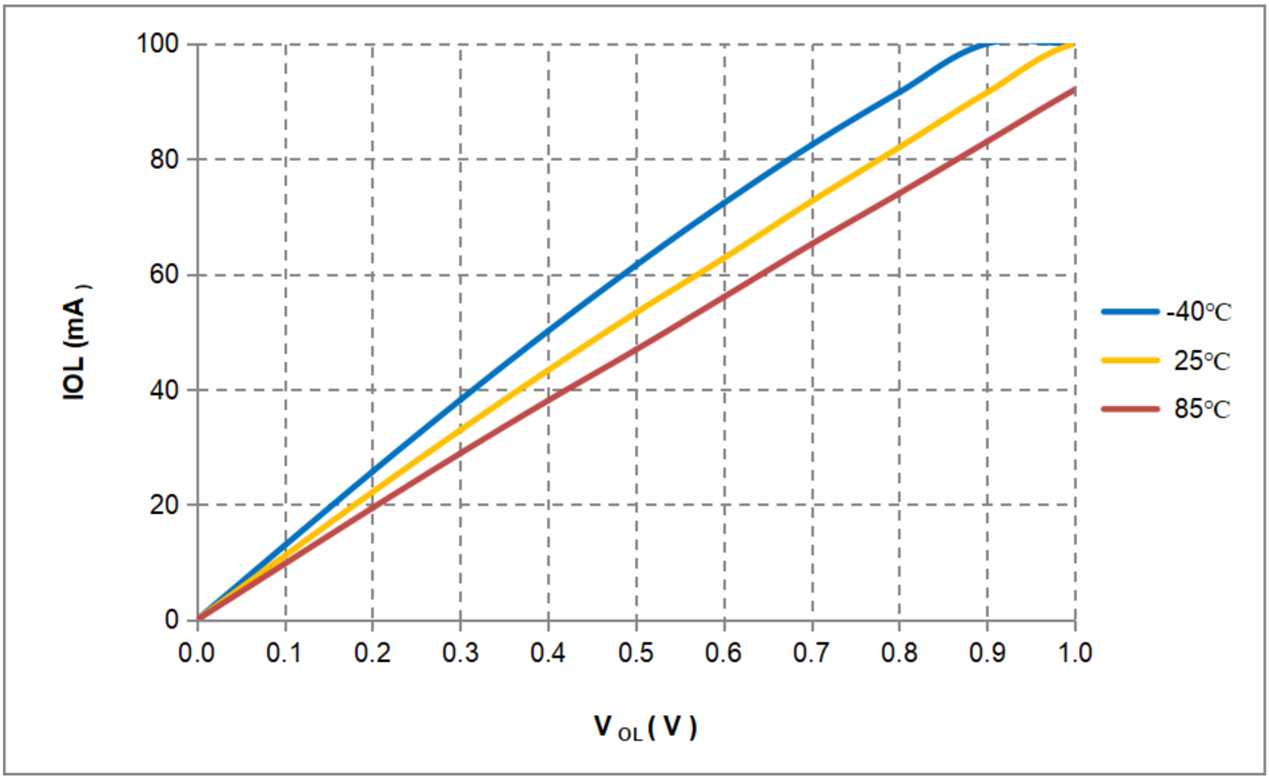


Figure 20-10 I_{Ol} vs V_{Ol} @L0 = 53mA , V_{DD} = 5V

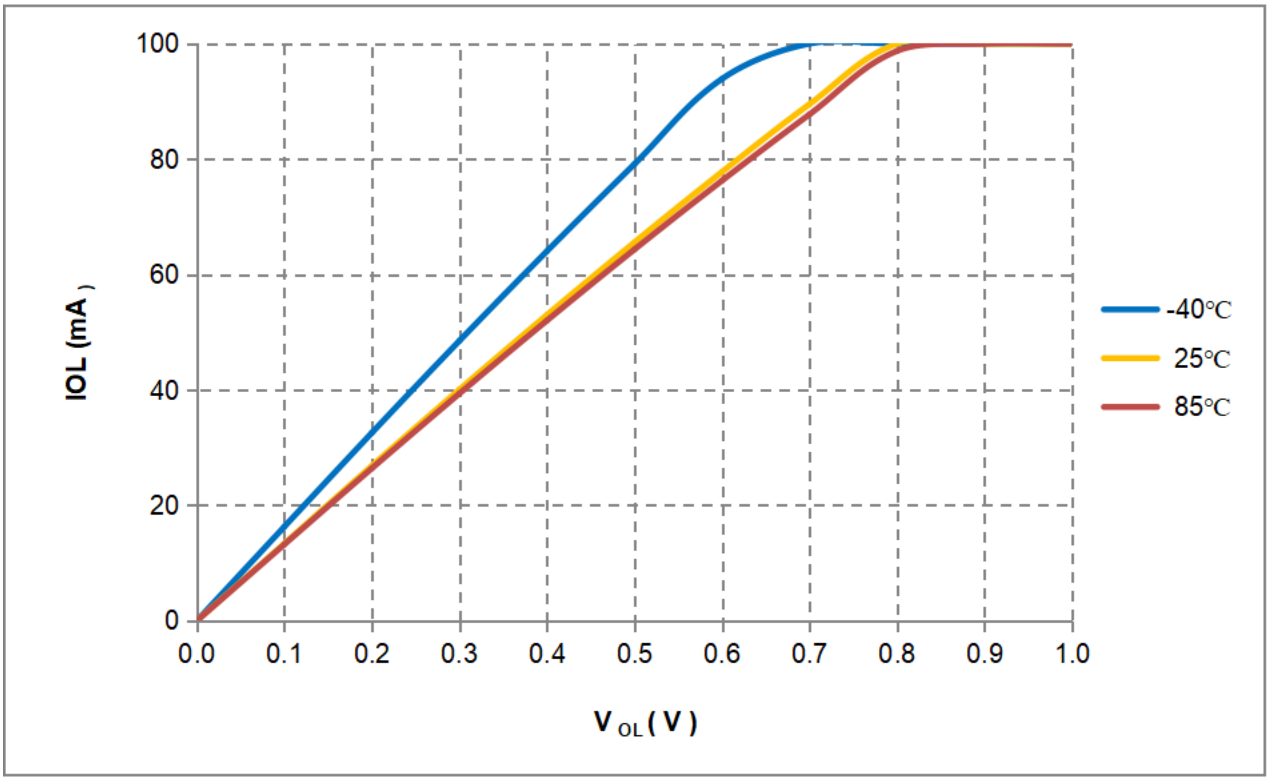
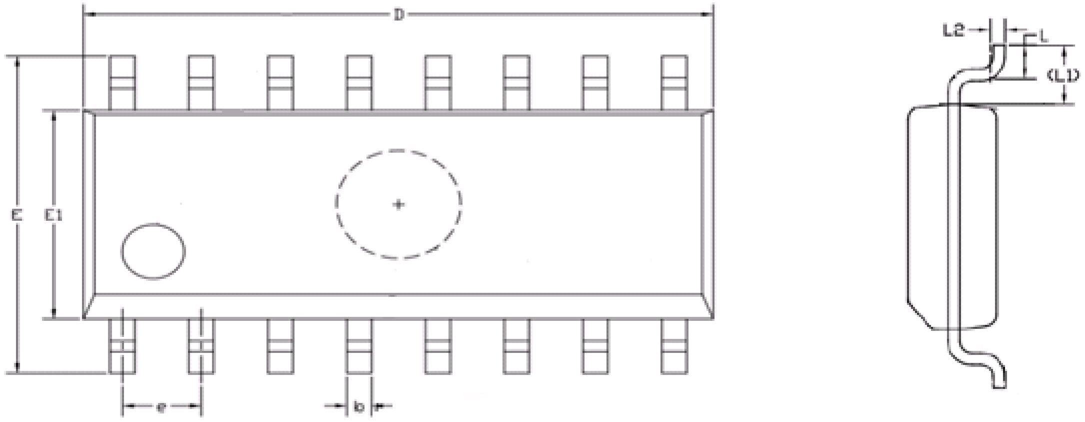


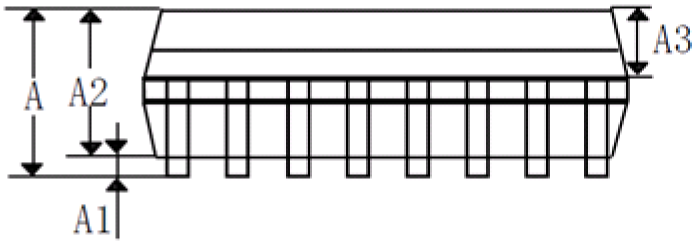
Figure 20-11 I_{OL} vs V_{OL} @L01 = 62mA , V_{DD} = 5V

21. PACKAGING INFORMATION

The device is available in SOP16, TSSOP20, SOP20, SOP24, TSSOP24, SOP28 and LQFP32 packages. The specific package size information is shown below:

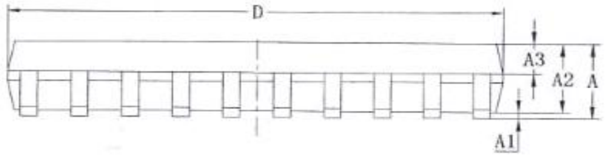
SOP16

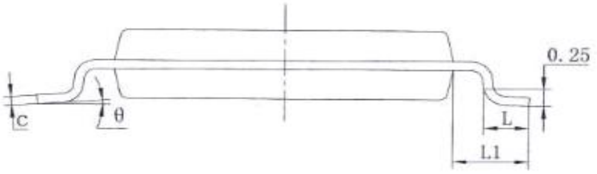


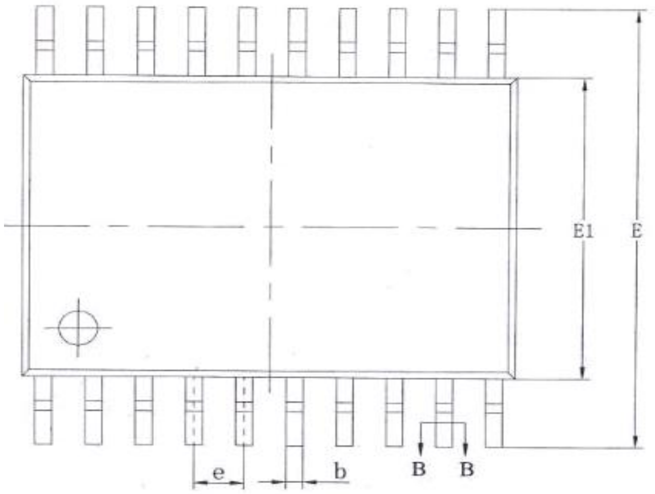


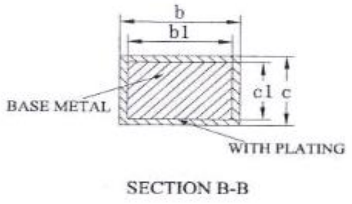
Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	-	1.750	-	0.069
A1	0.100	0.250	0.004	0.010
A2	1.250	1.550	0.049	0.061
A3	0.550	0.750	0.022	0.030
D	9.800	10.160	0.386	0.400
E	5.800	6.200	0.228	0.244
E1	3.800	4.000	0.150	0.157
b	0.310	0.510	0.012	0.020
e	1.270(BSC)		0.050(BSC)	
L	0.400	1.270	0.016	0.050
L1	1.04(REF)		0.04(REF)	
L2	0.25(BSC)		0.01(BSC)	

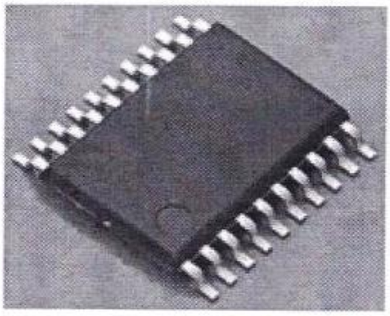
TSSOP20





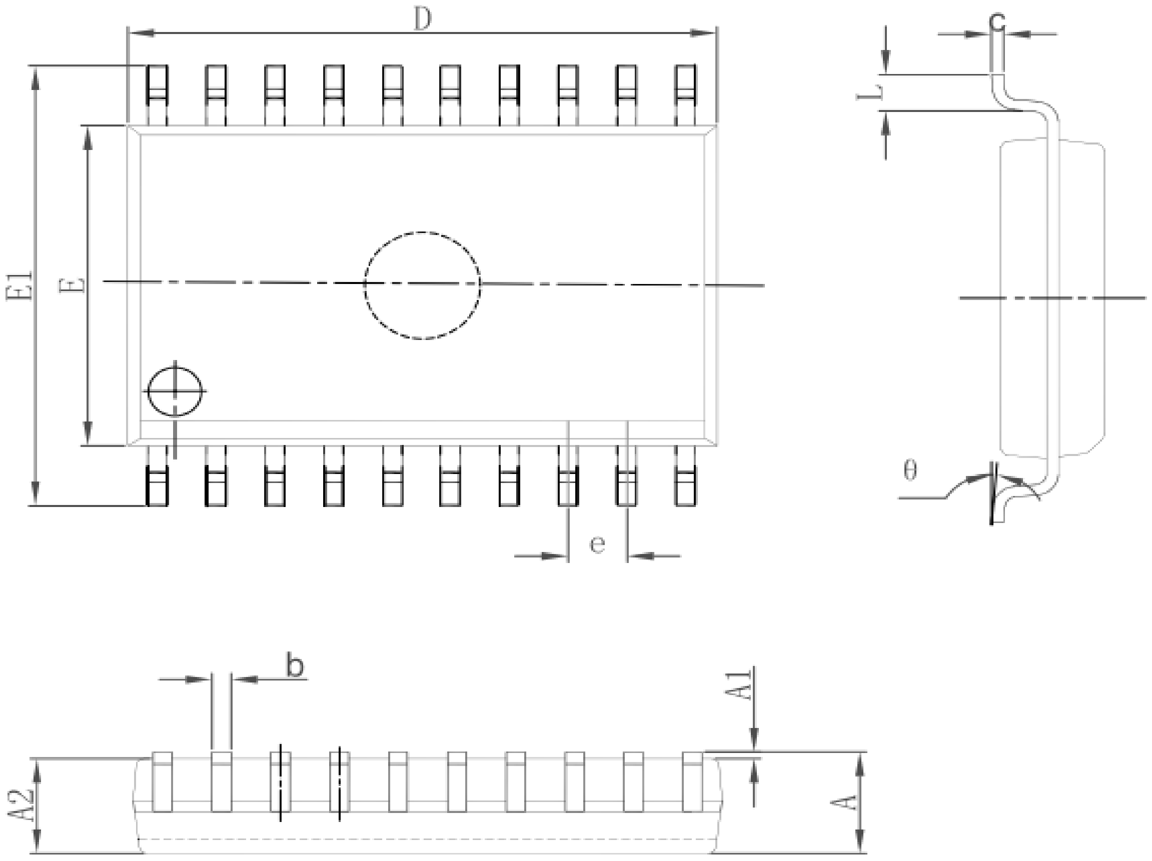






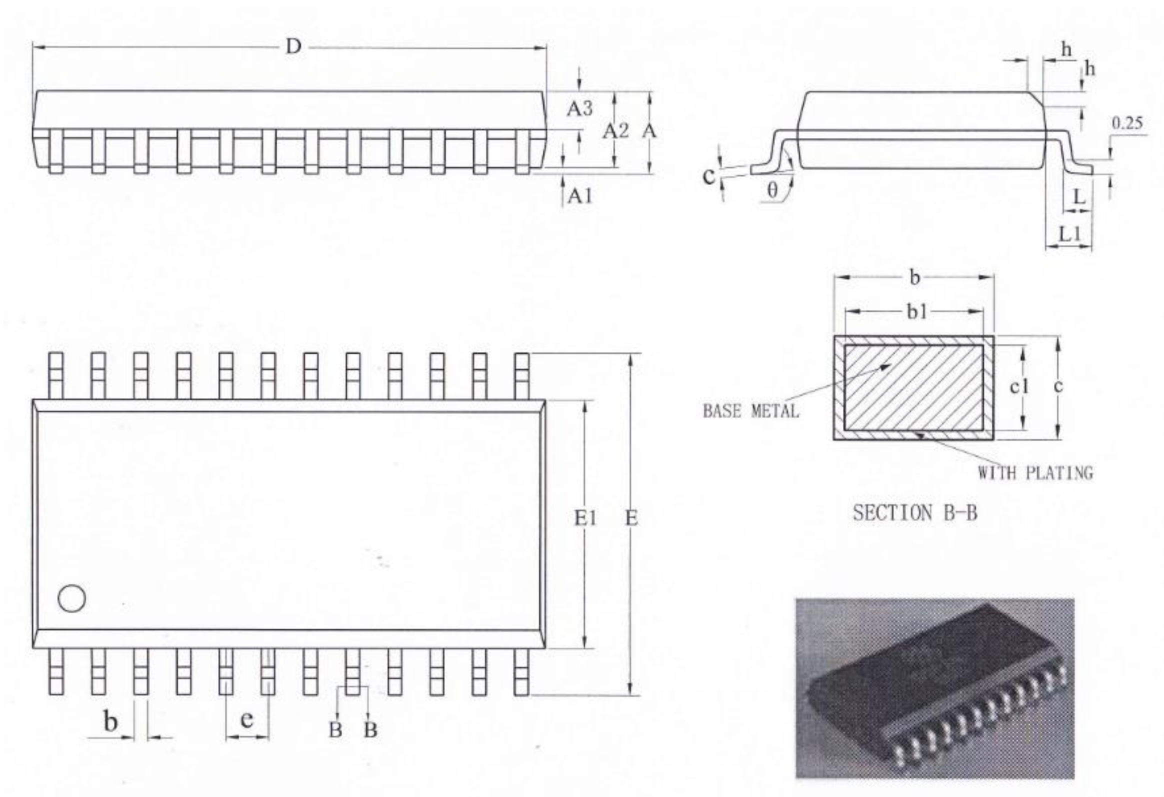
Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	-	1.20	-	0.047
A1	0.05	0.15	0.002	0.006
A2	0.80	1.05	0.031	0.041
A3	0.39	0.49	0.015	0.019
b	0.20	0.28	0.008	0.011
b1	0.19	0.25	0.007	0.010
c	0.13	0.17	0.005	0.007
c1	0.12	0.14	0.005	0.006
D	6.40	6.60	0.252	0.260
E1	4.30	4.50	0.169	0.177
E	6.20	6.60	0.244	0.259
e	0.65 (BSC)		0.026 (BSC)	
L	0.45	0.75	0.018	0.030
L1	1.00REF		0.039REF	
θ	0	8°	0	8°

SOP20



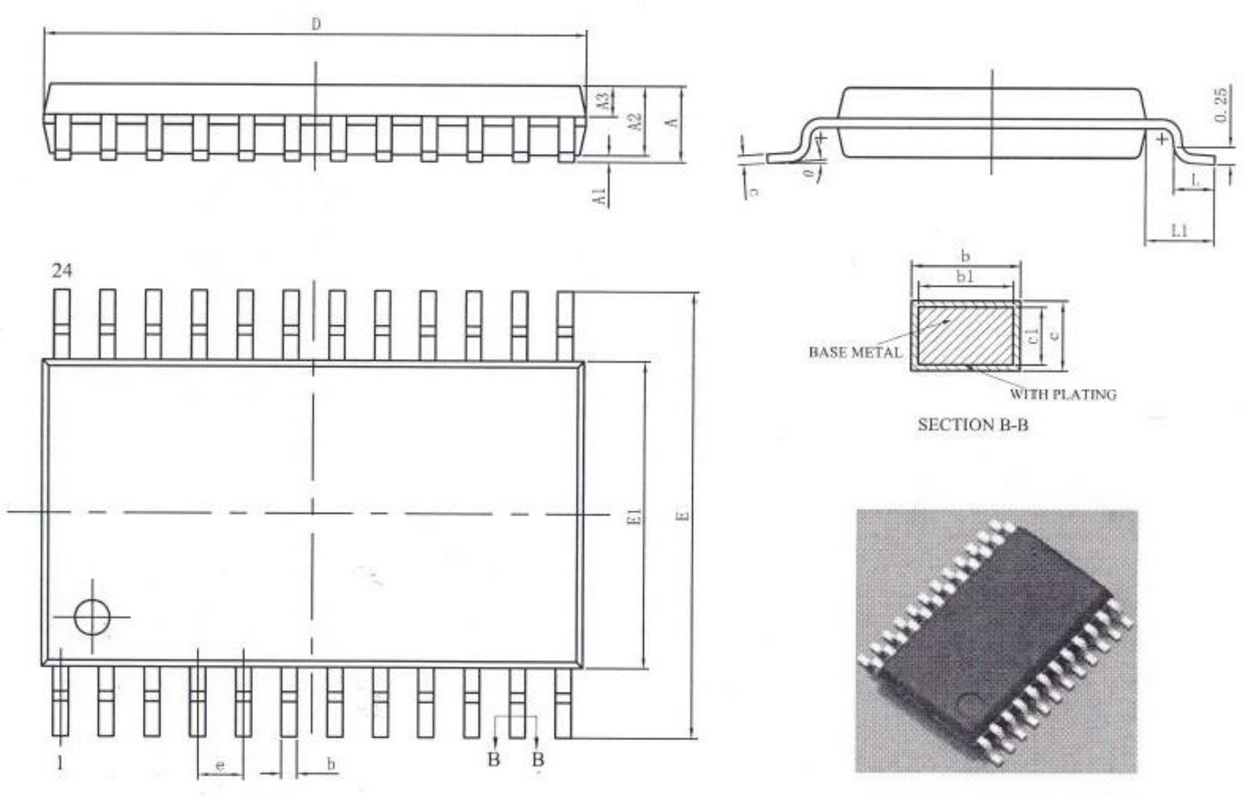
Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	2.350	2.650	0.093	0.104
A1	0.100	0.300	0.004	0.012
A2	2.100	2.500	0.083	0.098
b	0.330	0.510	0.013	0.020
c	0.204	0.330	0.008	0.013
D	12.520	13.000	0.493	0.512
E	7.400	7.600	0.291	0.299
E1	10.210	10.610	0.402	0.418
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

SOP24



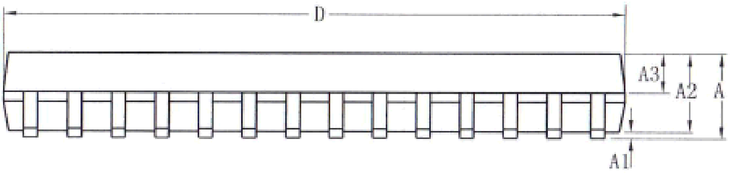
Symbol	Dimensions (mm)		
	Min	Nom	Max
A	2.36	2.54	2.64
A1	0.10	0.20	0.30
A2	2.26	2.30	2.35
A3	0.97	1.02	1.07
b	0.39	—	0.47
b1	0.38	0.41	0.44
c	0.25	—	0.29
c1	0.24	0.25	0.26
D	15.30	15.40	15.50
E	10.10	10.30	10.50
E1	7.40	7.50	7.60
e	1.27BSC		
L	0.70	—	1.00
L1	1.40REF		
h	0.25	—	0.75
θ	0	—	8°

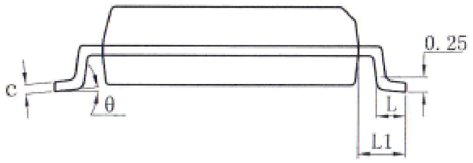
TSSOP24

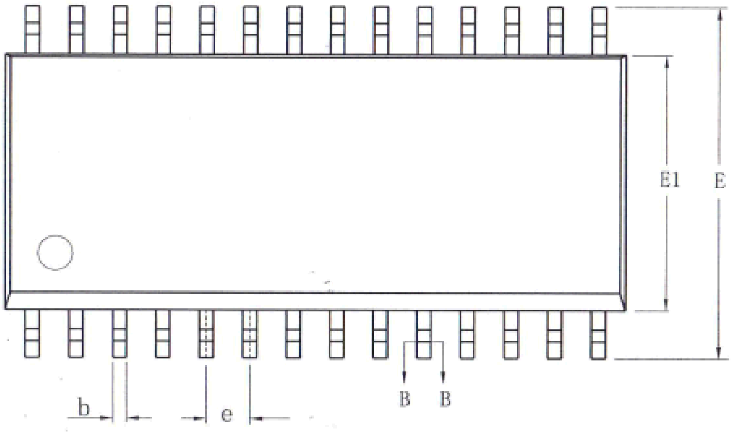


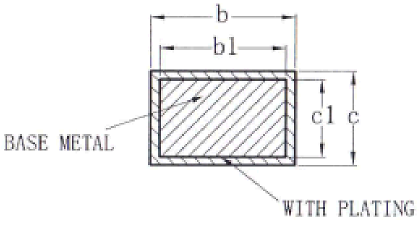
Symbol	Dimensions (mm)		
	Min	Nom	Max
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.29
b1	0.19	0.22	0.25
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	7.70	7.80	7.90
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
θ	0	—	8°

SOP28

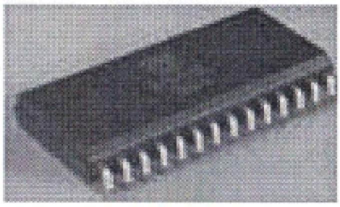






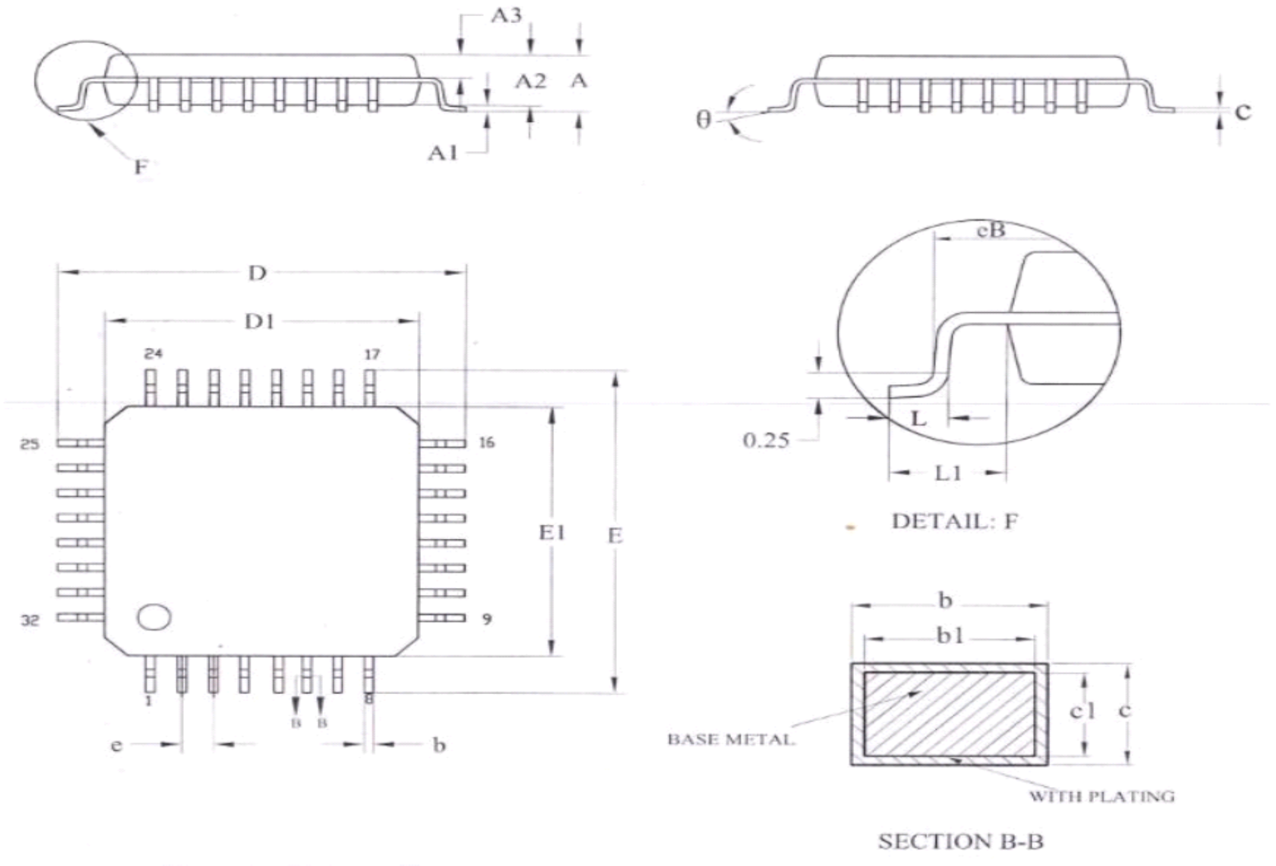


SECTION B-B



Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	-	2.65	-	0.104
A1	0.10	0.30	0.004	0.012
A2	2.25	2.35	0.089	0.093
A3	0.97	1.07	0.038	0.042
b	0.39	0.47	0.015	0.019
b1	0.38	0.44	0.015	0.017
c	0.25	0.29	0.010	0.011
c1	0.24	0.26	0.009	0.010
D	17.90	18.10	0.704	0.712
E	10.10	10.50	0.397	0.413
E1	7.40	7.60	0.290	0.299
e	1.27 (BSC)		0.05 (BSC)	
L	0.70	1.00	0.027	0.039
L1	1.40REF		0.055REF	
θ	0	8°	0	8°

LQFP32



Symbol	Dimensions (mm)		Dimensions (inches)	
	Min	Max	Min	Max
A	-	1.60	-	0.063
A1	0.05	0.15	0.002	0.006
A2	1.35	1.45	0.053	0.057
A3	0.59	0.69	0.023	0.027
b	0.33	0.41	0.013	0.016
b1	0.32	0.38	0.013	0.015
c	0.13	0.17	0.005	0.006
c1	0.12	0.14	0.005	0.006
D	8.80	9.20	0.346	0.362
D1	6.90	7.10	0.272	0.280
E	8.80	9.20	0.346	0.362
E1	6.90	7.10	0.272	0.280
eB	8.10	8.25	0.319	0.324
e	0.80 (BSC)		0.031 (BSC)	
L	0.45	0.75	0.018	0.030
L1	1.00REF		0.039REF	
θ	0	7°	0	7°

22. Appendix : Register Types

Abbr.	Description	Illustration
WO	Write Only, read "0"	Write only, read as 0
RO	Read Only	Read only
RW	Read, Write	Readable, writable
RW0	Read, Write “0” only	Readable, only write 0, write 1 is invalid
RW1	Read, Write “1” only	Readable, only write 1, write 0 is invalid
R_W1C	Read, Cleared by Writing “1”	Readable, write 1 to clear, and writing '0' has no effect on the bit
Res	Reserved, read "0"	Reserved bit, read only, read as 0

Contact Information

Fremont Micro Devices Corporation

#5−8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811
Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Limited

#16, 16/F, Block B, Veristrong Industrial Centre,
34−36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186
Fax: (+852) 2781 1144

http://www.fremontmicro.com

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.