

# YF8316 16/32 16KB/32KB flash 的 AD 型 8051 单片机

## 产品说明书

说明书发行履历:

版本	发行时间	新制/修订内容
2015-05-A1	2015-05	新制

### 1、概述

YF8316 16/32是一款有16k/32k字节FLASH的8位单片机。本芯片是一款可以为许多嵌入式控制应用提供高效灵活低成本解决方案的微控器。其主要特点如下：

### 功能特性：

- **CPU**
  - 8 Bit CISC 内核 (8051 兼容)
- **ROM (FLASH) 容量**
  - 16k/32k 字节
  - 自读写Flash
  - 擦写次数 : 100,000 次
- **256 字节 IRAM**
- **512 字节 XRAM**
- **通用 I/O (GPIO)口**
  - 普通 I/O : 30 个口  
(P0[6:0], P1[7:0], P2[6:0], P3[7:0])
  - LED 显示驱动能力管脚 : 14 个口  
(P1[7:0], P2[5:0])
- **BIT**
  - 8Bit × 1ch
- **WDT**
  - 8Bit × 1ch
  - 5kHz 内部 RC 振荡器
- **定时器/计数器**
  - 8Bit × 1ch (T0), 16Bit × 2ch (T1/T2)
- **PPG**
  - 脉冲发生器 (T1/T2)
  - 8Bit PWM (by T0)
- **WT**
  - 32.768kHz 时3.91mS/0.25S/0.5S/1S/1M 间隔
- **蜂鸣器**
  - 8Bit × 1ch
- **SPI**
  - 8Bit × 1ch
- **UART**
  - 8Bit × 1ch
- **I2C**
  - 8Bit × 1ch
- **12 位 A/D 转换器**
  - 15 个输入通道
  - 掉电唤醒功能
- **上电复位**
  - 复位电压 (1.4V)
- **低电压复位**
  - 14个电平选择  
(1.60V/2.00V/2.10V/2.20V/2.32V/2.44V/2.59V/2.75V/2.93V/3.14V/3.38V/3.67V/4.00V/4.40V)
- **低电压检测**
  - 13个电平选择  
(2.00V/2.10V/2.20V/2.32V/2.44V/2.59V/2.75V/2.93V/3.14V/3.38V/3.67V/4.00V/4.40V)
- **中断源**
  - 外部中断  
(EINT0~4, EINT5, EINT6, EINT7~A, EINT10, EINT11, EINT12) (7)
  - 定时器(0/1/2) (4)
  - WDT (1)
  - BIT (1)
  - WT (1)
  - SPI (1)
  - UART (2)
  - I2C (1)
  - ADC (1)
  - ADC Wake-up (1)
- **内部 RC 振荡器**
  - 内部 RC 频率: 16MHz ±1.0% (TA= 0 ~ +50°C)
- **掉电模式**
  - STOP, IDLE 模式
- **操作电压**
  - 1.8V ~ 5.5V (@ 32 ~ 38kHz with X-tal)
  - 1.8V ~ 5.5V (@ 0.4 ~ 4.2MHz with X-tal)
  - 2.7V ~ 5.5V (@ 0.4 ~ 10.0MHz with X-tal)
  - 3.0V ~ 5.5V (@ 0.4 ~ 12.0MHz with X-tal)
  - 1.8V ~ 5.5V (@ 0.5 ~ 8.0MHz with Internal RC)
  - 2.0V ~ 5.5V (@ 0.5 ~ 16.0MHz with Internal RC)
  - Voltage dropout converter included for core
- **最小指令执行时间**
  - 125nS (@ 16MHz main clock)
  - 61μS (@ 32.768kHz sub clock)

● 振荡器类型

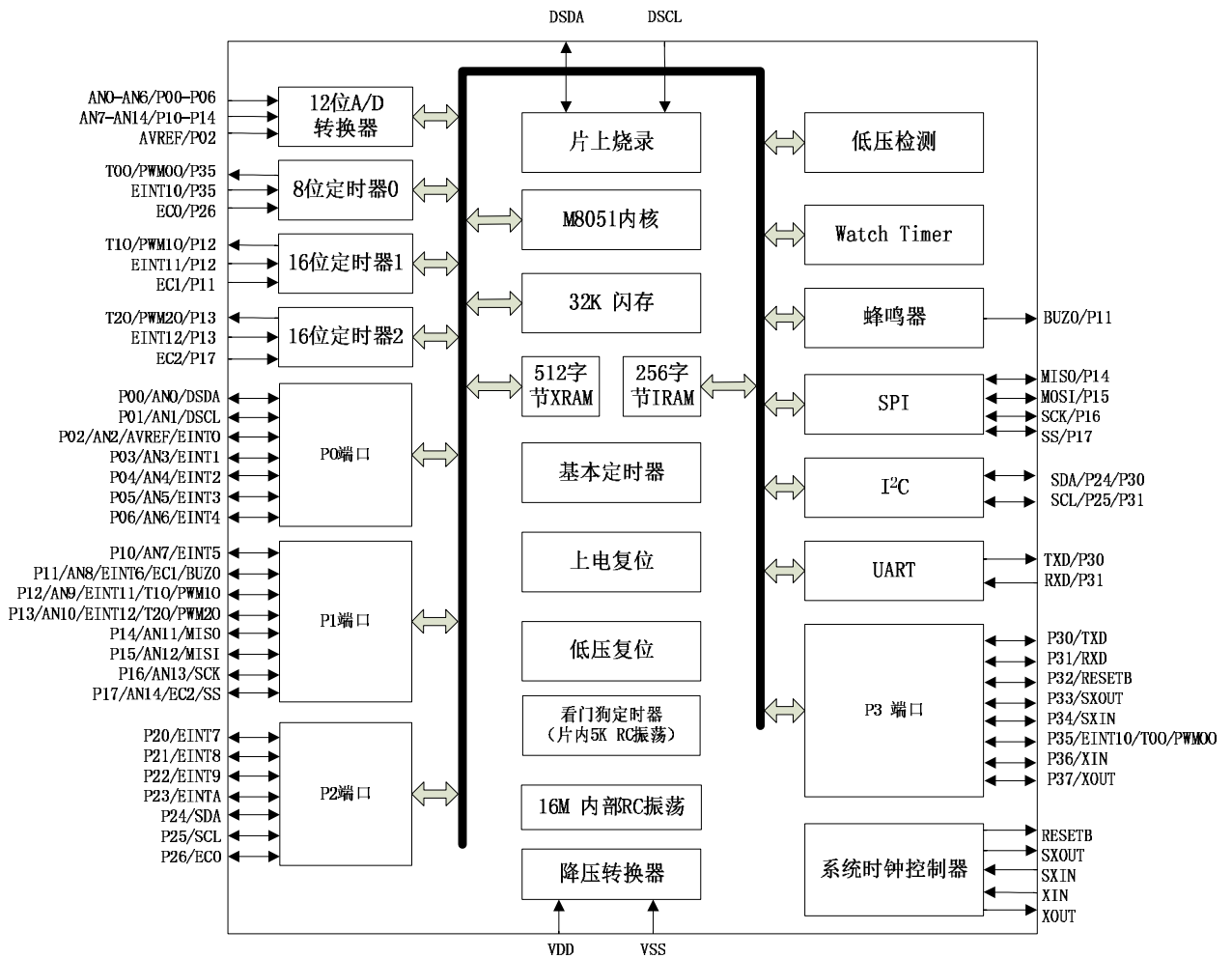
- 0.4-12MHz 晶体或陶振做主时钟  
 - 32.768kHz 晶体做副时钟

● 操作温度: -40 ~ +85°C

● 封装形式: TSSOP20/SOP28/SOP32/QFN32

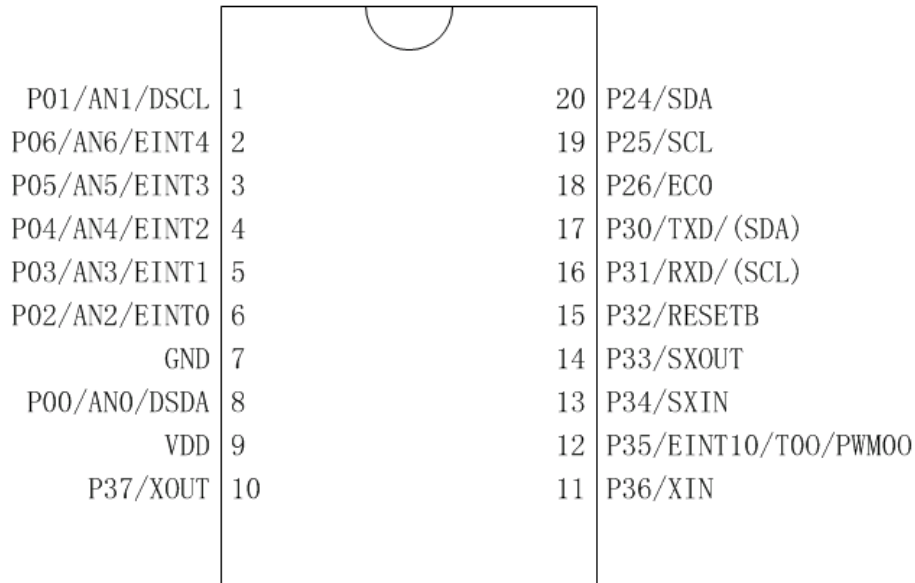
2、功能框图及引脚说明

2.1、功能框图

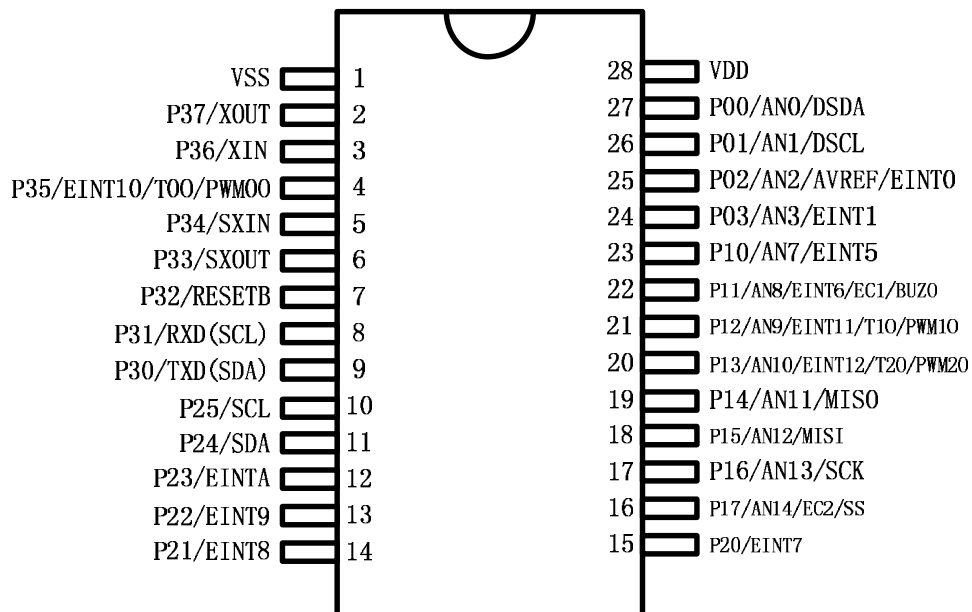


注意) 1.P04-P06, 和 P26 在 28-pin 封装中不存在。  
 2.P10-P17, P20-P23 在 20-pin 封装中不存在。

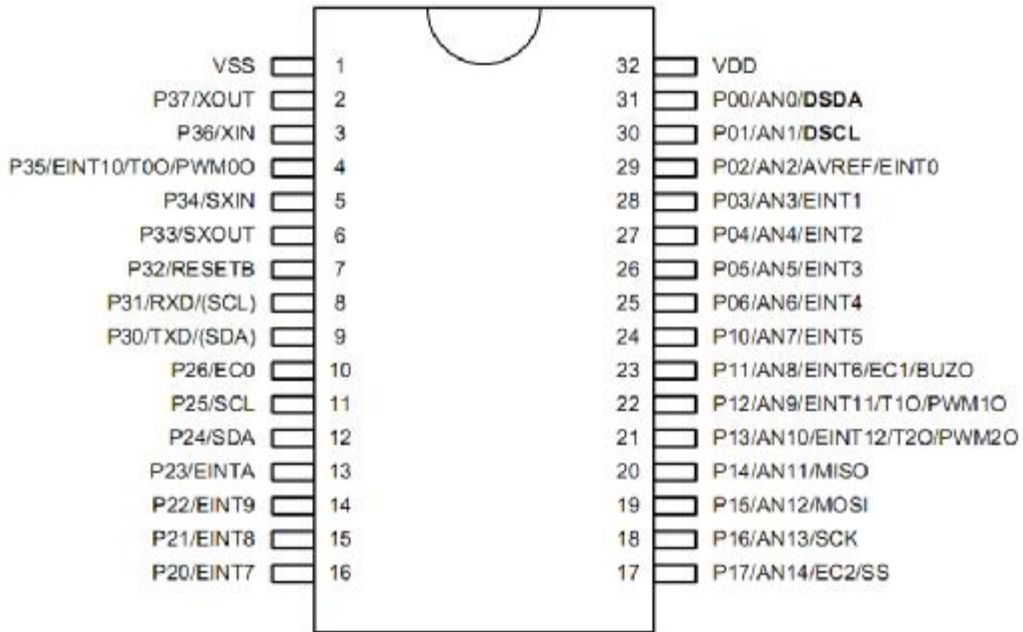
2.2、引脚排列图



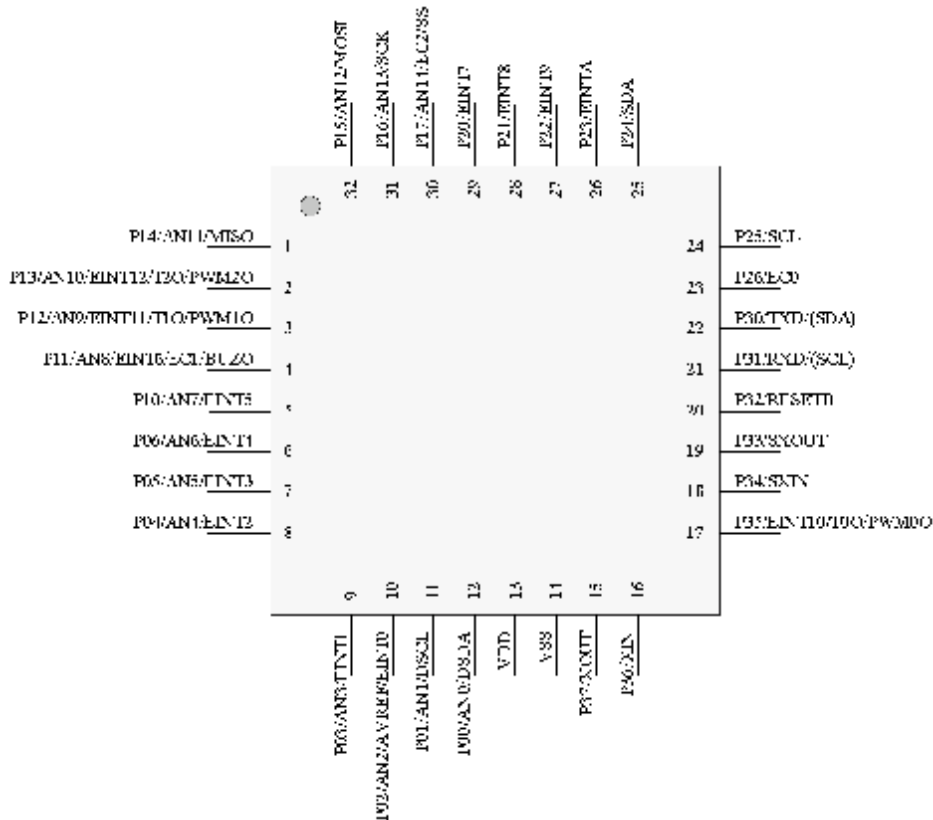
TSSOP20 的引脚分配



SOP28 的引脚分配



SOP32 的引脚分配



QFN32 的引脚分配

- NOTES)
1. ISP 使用 P0[1:0] 作为 DSCL, DSDA。
  2. 在使用 28 脚封装时 P04-P06 和 P26 需要通过软件设置为推-拉输出或者上拉输入。
  3. 在 I2C 中 P30/P31 可通过软件设置作为 SDA/SCL 线来使用。

### 2.3、引脚说明及结构原理图

引脚名称	I/O	功 能	@res et	引脚共享
P00	I/O	P0 端口可作为施密特触发输入，推-拉输出，或者漏极开路输出 I/O口。推-拉寄存器可指定到每一位。	Input	AN0/DSDA
P01				AN1/DSCL
P02				AN2/AVREF/EINT0
P03				AN3/EINT1
P04				AN4/EINT2
P05				AN5/EINT3
P06				AN6/EINT4
P10	I/O	P1端口可作为施密特触发输入，推-拉输出，或者漏极开路输出。推-拉寄存器可指定到每一位。	Input	AN7/EINT5
P11				AN8/EINT6/EC1/BUZO
P12				AN9/EINT11/T10/PWM10
P13				AN10/EINT12/T20/PWM20
P14				AN11/MISO
P15				AN12/MOSI
P16				AN13/SCK
P20	I/O	P2端口可作为施密特触发输入，推-拉输出，或者漏极开路输出。推-拉寄存器可指定到每一位。	Input	EINT7
P21				EINT8
P22				EINT9
P23				EINTA
P24				SDA
P25				SCL
P26				EC0
P30	I/O	P3端口可作为施密特触发输入，推-拉输出，或者漏极开路输出。推-拉寄存器可指定到每一位。	Input	TXD
P31				RXD
P32				RESETB
P33				SXOUT
P34				SXIN
P35				EINT10/T00/PWM00
P36				XIN
P37				XOUT
EINT0	I/O	外部中断输入	Input	P02/AN2/AVREF
EINT1				P03/AN3
EINT2				P04/AN4
EINT3				P05/AN5
EINT4				P06/AN6
EINT5				P10/AN7
EINT6				P11/AN8/EC1/BUZO
EINT7				P20
EINT8				P21
EINT9				P22
EINTA				P23

EINT10	I/O	外部中断输入和定时器 0 捕捉输入	Input	P35/T00/PWM00
EINT11	I/O	外部中断输入和定时器 1 捕捉输入	Input	P12/AN9/T10/PWM10
EINT12	I/O	外部中断输入和定时器 2 捕捉输入	Input	P13/AN10/T20/PWM20
T00	I/O	定时器 0 间隔输出	Input	P35/EINT10/PWM00
T10	I/O	定时器 1 间隔输出	Input	P12/AN9/EINT11/PWM10
T20	I/O	定时器 2 间隔输出	Input	P13/AN10/EINT12/PWM20
PWM00	I/O	定时器 0 PWM 输出	Input	P35/EINT10/T00
PWM10	I/O	定时器 1 PWM 输出	Input	P12/AN9/EINT11/T10
PWM20	I/O	定时器 2 PWM 输出	Input	P13/AN10/EINT12/T20
EC0	I/O	定时器 0 EC 输入	Input	P26
EC1	I/O	定时器 1 EC 输入	Input	P11/AN8/EINT6/BUZO
EC2	I/O	定时器 2 EC 输入	Input	P17/AN14/SS
BUZO	I/O	蜂鸣器信号输出	Input	P11/AN8/EINT6/EC1
SCK	I/O	时钟输入/输出	Input	P16/AN13
MISO	I/O	串行数据输入/输出	Input	P14/AN11
MISI	I/O	串行数据输入/输出	Input	P15/AN12
SS	I/O	从动输入选择	Input	P17/AN14/EC2
TXD	I/O	UART 数据输出	Input	P30
RXD	I/O	UART 数据输入	Input	P31
SCL	I/O	I2C 时钟输入/输出	Input	P25
SDA	I/O	I2C 数据输入/输出	Input	P24
AVREF	I/O	A/D 转换参考电压	Input	P02/AN2/EINT0
AN0	I/O	A/D 转换模拟输入通道	Input	P00/DSDA
AN1				P01/DSCL
AN2				P02/AVREF/EINT0
AN3				P03/EINT1
AN4				P04/EINT2
AN5				P05/EINT3
AN6				P06/EINT4
AN7				P10/EINT5
AN8				P11/EINT6/EC1/BUZO
AN9				P12/EINT11/T10/PWM10
AN10				P13/EINT12/T20/PWM20
AN11				P14/MISO
AN12				P15/MOSI
AN13				P16/SCK
AN14	P17/EC2/SS			
RESETB	I/O	作为复位脚使用时需要通过 CONFIGURE OPTION将系统复位脚设置上拉	Input	P32

DSDA	I/O	片内调试数据 输入/输出 <sup>(NOTE5,6)</sup>	Input	P00
SDCL	I/O	片内调试数据 输入/输出 <sup>(NOTE5,6)</sup>	Input	P01
XIN	I/O	主时钟振荡器引脚	Input	P36
XOUT				P37
SXIN	I/O	副时钟振荡器引脚	Input	P34
SXOUT				P33
VDD VSS	-	电源脚	-	-

注意)1. P04-P06 和 P26 在 28 脚封装中不存在.

2. P10-P17, P20-P23 在 20 脚封装中不存在.

3. P32/RESETB 通过“CONFIGURE OPTION”设置来作为 P32 脚和 RESETB 脚

4. 如果 P00/DSDA 和 P01/DSCL 在上电复位时连接了仿真器, 这个脚则自动的配置为仿真脚

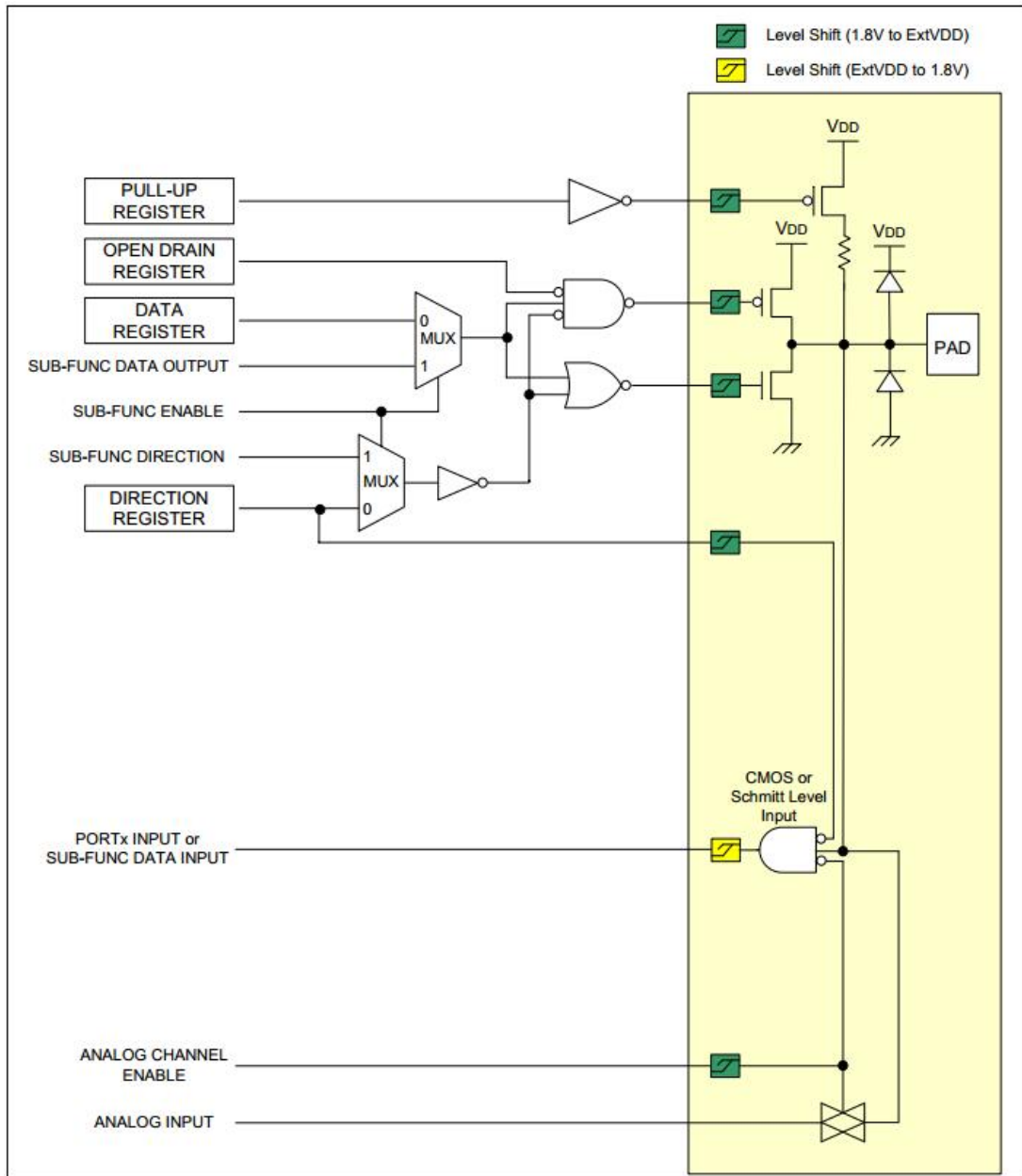
5. P00/DSDA 和 P01/DSCL 仅在复位和上电时被作为内部上拉输入

6. P37/XOUT, P36/XIN, P34/SXIN, 和 P33/SXOUT 通过软件控制设置复用功能

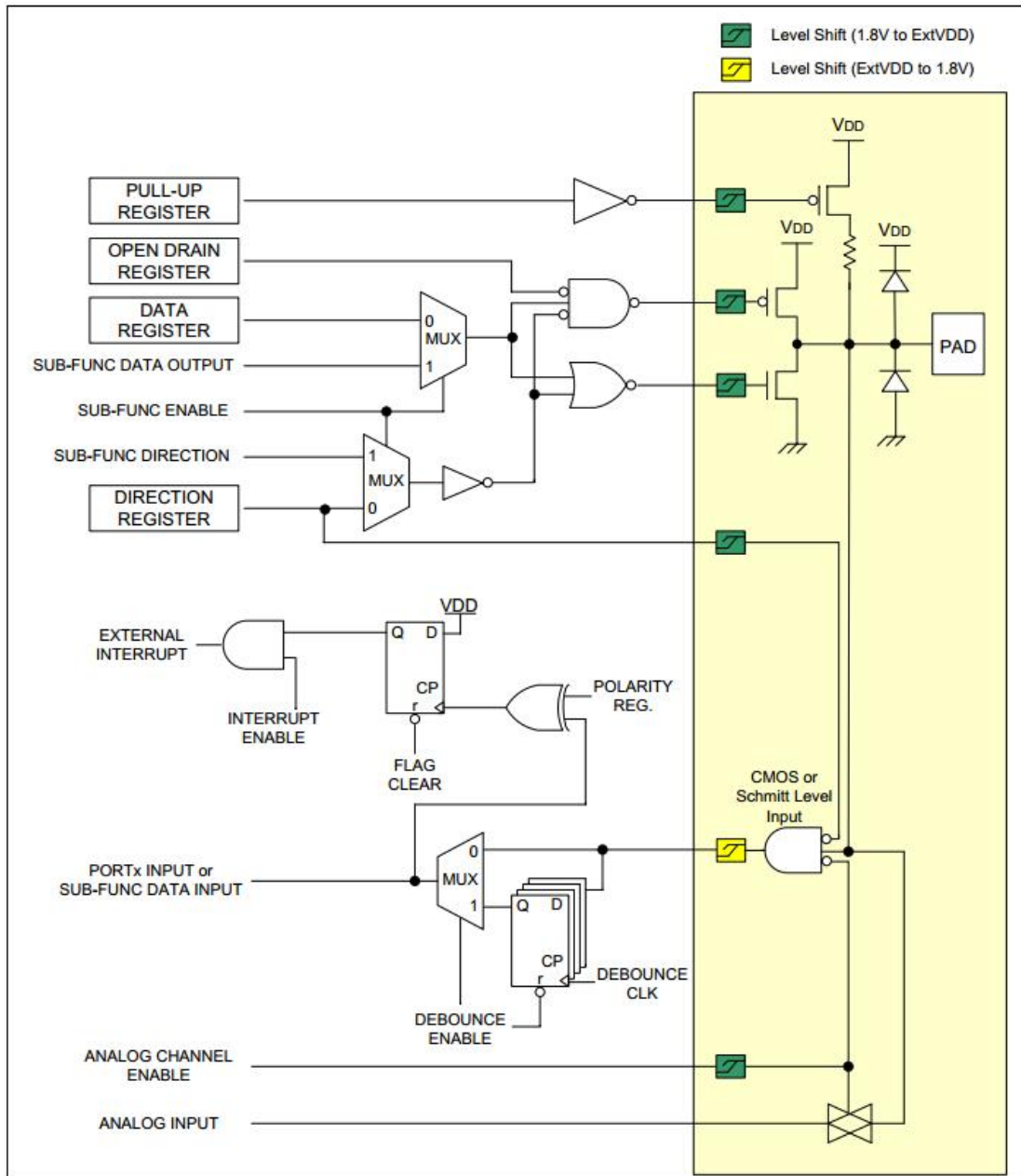
## 2.4、引脚电路结构图

### Ø 通用 IO 口





Ø 外部中断 I/O 口



### 3、电特性

#### 3.1、 极限参数

除非另有规定,  $T_{amb}=25^{\circ}\text{C}$ 

参数名称	符号	注意	额定值	单位	
电源电压	$V_{DD}$	--	-0.3~+6.5	V	
常规引脚电压	$V_I$	每个引脚到VSS的电压	-0.3~ $V_{DD}+0.3$	V	
	$V_O$		-0.3~ $V_{DD}+0.3$	V	
	$I_{OH}$	Maximum current output sourced by (IOH per I/O pin)	-25	mA	
	$\Sigma I_{OH}$	Maximum current ( $\Sigma$ IOH)	-200	mA	
	$I_{OL}$	Maximum current sunk by ( $I_{OL}$ per I/O pin)	180	mA	
	$\Sigma I_{OL}$	Maximum current ( $\Sigma$ IOL)	200	mA	
总功率消耗	$P_T$	--	600	mW	
存放温度	$T_{STG}$	--		-65~+150	
焊接温度	$T_L$	10 秒	DIP	245	°C
			SOP	250	

注：芯片使用时超过极限参数可能会对芯片造成永久性损坏。这是一个额定值，在任何情况下如果对芯片的有效操作参数超过规格书规定值都是不允许的，长时间超过极限参数工作会影响器件的可靠性。

### 3.2、推荐使用条件( $T_A = -40^{\circ}\text{C} \sim +85^{\circ}\text{C}$ )

参数名称	符号	条件	最小	典型	最大	单位	
操作电压	VDD	$f_x=32\sim 38\text{kHz}$	SX-tal	1.8	-	5.5	V
		$f_x=0.4\sim 4.2\text{MHz}$	X-tal	1.8	-	5.5	
		$f_x=0.4\sim 10\text{MHz}$		2.7	-	5.5	
		$f_x=0.4\sim 12\text{MHz}$	内部 RC	3.0	-	5.5	
		$f_x=0.5\sim 8\text{MHz}$		1.8	-	5.5	
		$f_x=0.5\sim 16\text{MHz}$		2.0	-	5.5	
操作温度	$T_{OPR}$	VDD=1.8~5.5V	-40	-	85	°C	

### 3.3、电气特性

#### 3.3.1、A/D 转换特性

(TA = -40°C ~ +85°C, VDD = 1.8V ~ 5.5V, VSS = 0V)

参数名称	符号	测试条件	最小	典型	最大	单位
分辨率	-	-	-	12	-	Bit
积分线性误差	ILE	AVREF = 2.7V - 5.5V $f_x = 8\text{MHz}$	-	-	$\pm 4$	LSB
微分线性误差	DLE		-	-	$\pm 1$	
零点偏移误差	ZOE		-	-	$\pm 3$	
满量程误差	FSE		-	-	$\pm 3$	
转换时间	$t_{CON}$	12bit resolution, 8MHz	20	-	-	uS
模拟输入电压	$V_{AN}$		VSS	-	AVREF	V

模拟参考电压	AVREF		1.8	-	VDD	V
VDD	-	-	-	1.8	-	V
模拟输入漏电流	$I_{AN}$	AVREF=5.12V	-	-	2	uA
ADC 操作电流	$I_{ADC}$	Enable	-	1	2	mA
		Disable	-	-	0.1	uA

注意) 1.零点偏移误差是指 000000000000 和转换输出的零点输入电压(VSS)。

2.满量程误差是指 111111111111 和转换输出的满量程输入电压(AVREF)。

### 3.3.2、上电复位特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

参数名称	符号	测试条件	最小	典型	最大	单位
复位电平	$V_{POR}$	--	-	1.4	-	V
VDD 上升时间	$t_r$	--	0.05	-	30	V/mS
POR 电流	$I_{POR}$	--	-	0.2	-	uA

### 3.3.3、低电压复位和低电压检测特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

参数名称	符号	测试条件	最小	典型	最大	单位
检测电平	$V_{LVR}$ $V_{LVI}$	LVR 可以选择使用电平但LVI不能选择 1.60V.	--	1.6	1.75	V
			1.85	2.00	2.15	
			1.95	2.1	2.25	
			2.05	2.2	2.35	
			2.17	2.32	2.47	
			2.29	2.44	2.59	
			2.39	2.59	2.79	
			2.55	2.75	2.95	
			2.73	2.93	3.13	
			2.94	3.14	3.34	
			3.18	3.38	3.58	
			3.37	3.67	3.97	
3.7	4	4.3				
4.1	4.3	4.7				
迟滞	$\Delta V$	--	-	50	150	mV
最小脉冲宽度	$t_{LW}$	--	100			uS
LVR 和 LVI 电流	$I_{BL}$	均使能		10	15	uA
		使能其中一个		8	12	uA
		两个都关闭			0.1	uA

### 3.3.4、内部 RC 振荡特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

参数名称	符号	测试条件	最小	典型	最大	单位
------	----	------	----	----	----	----

频率	$f_{IRC}$	VDD=2~5.5V	-	16	-	MHz
公差	-	TA=0°C~+50°C	-	-	±1	%
		TA=-20°C~+85°C	-	-	±2	
		TA=-40°C~+85°C	-	-	±3	
时钟占空比	TOD		40	50	60	%
稳定时间	$T_{HFS}$				100	uS
IRC 电流	$I_{IRC}$	Enable			0.2	mA
		Disable				0.1

### 3.3.5、内部 WDTRC 振荡特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

参数名称	符号	测试条件	最小	典型	最大	单位
频率	$f_{WDTRC}$	--	2	5	10	kHz
稳定特性	$f_{WDTS}$	--	-	-	1	mS
WDTRC 电流	$I_{WDTRC}$	Enable	-	1	-	uA
		Disable	-	-	0.1	

### 3.3.6、DC 特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V,  $f_{XIN}$ = 12MHz)

参数名称	符号	测试条件	最小	典型	最大	单位	
输入高电平	$V_{IH}$	所有输入管脚, RESETB	0.8VDD	-	VDD	V	
输入低电平	$V_{IL}$	所有输入管脚, RESETB	-	-	0.2VDD	V	
输出高电平	$V_{OH1}$	VDD=4.5V, $I_{OH}$ =-10mA; 所有输出脚除了 $V_{OH2}, V_{OH3}$	VDD-1.5	-	-	V	
	$V_{OH2}$	VDD=4.5V, $I_{OH}$ =-10mA; P20-P25	VDD-2.0	-	-		
	$V_{OH3}$	VDD=4.5V, $I_{OH}$ =-20mA; P1	VDD-2.0	-	-		
输出低电压	$V_{OL1}$	VDD=4.5V, $I_{OL}$ =15mA; 所有输出管脚除了 $V_{OL2}$			1.0	V	
	$V_{OL2}$	VDD=4.5V, $I_{OL}$ =160mA; P20-P25		1.5	3.0		
输入高漏电流	$I_{IH}$	所有输入管脚			1	uA	
输入低漏电流	$I_{IL}$	所有输入管脚	-1			uA	
	$R_{PU1}$	VI=0V, TA=25°C, 所有输入管脚	VDD=5V	25	50	100	kΩ
			VDD=3V	50	100	200	
	$R_{PU2}$	VI=0V, TA=25°C, RESETB	VDD=5V	150	250	400	kΩ
			VDD=3V	300	500	700	
ADC 唤醒上	$R_{AWPU1}$	TA=25°C		100	150	200	kΩ

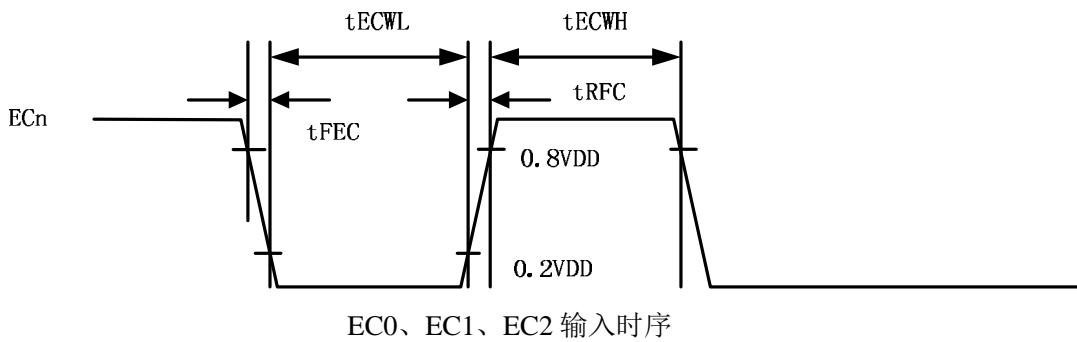
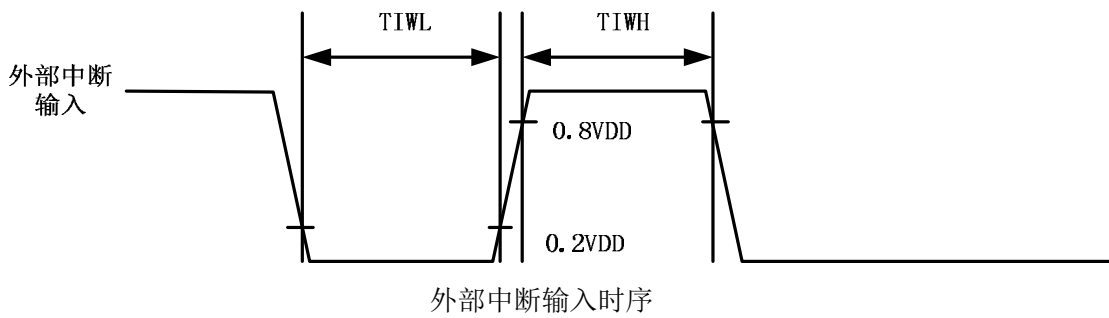
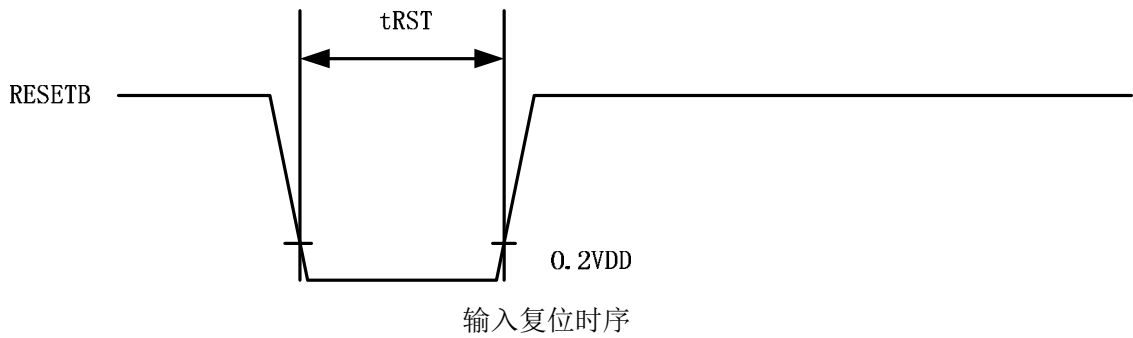
拉电阻	$R_{AWPU2}$		200	300	400	
OSC 反馈电阻	$R_{X1}$	XIN=VDD,XOUT=VSS TA=25°C,VDD=5V	600	1200	2000	kΩ
	$R_{X2}$	SXIN=VDD,SXOUT=VSS TA=25°C,VDD=5V	2500	5000	10000	
操作电压	$I_{DD1}$ (RUN)	$f_{XIN}=12\text{MHz}, VDD=5V\pm 10\%$		3	6	mA
		$f_{XIN}=10\text{MHz}, VDD=3V\pm 10\%$		2.2	4.4	
		$f_{XIN}=16\text{MHz}, VDD=5V\pm 10\%$		3.0	6	
	$I_{DD2}$ (IDLE)	$f_{XIN}=12\text{MHz}, VDD=5V\pm 10\%$		1.3	2.6	mA
		$f_{XIN}=10\text{MHz}, VDD=3V\pm 10\%$		0.7	1.4	
		$f_{XIN}=16\text{MHz}, VDD=5V\pm 10\%$		0.8	1.6	
	$IDD3$	$f_{XIN}=32.768\text{kHz}$		60	90	uA
$IDD4$	VDD=3V±10% TA=25°C		6	12	uA	uA
$IDD5$	STOP, VDD= 5V±10%, TA= 25°C			0.5	3	uA

- 注意) 1.  $f_{XIN}$  是外部主时钟,  $f_{SUB}$  是外部副时钟,  $f_{IRC}$  是内部 RC,  $f_x$  是选择系统时钟。  
 2. 所有的电流项不包括内部 Watch-dog timer RC(WDTRC)振荡器和外围设备。  
 3. 所有的电流项包括上电复位模块(POR)。

### 3.3.7、AC 特性

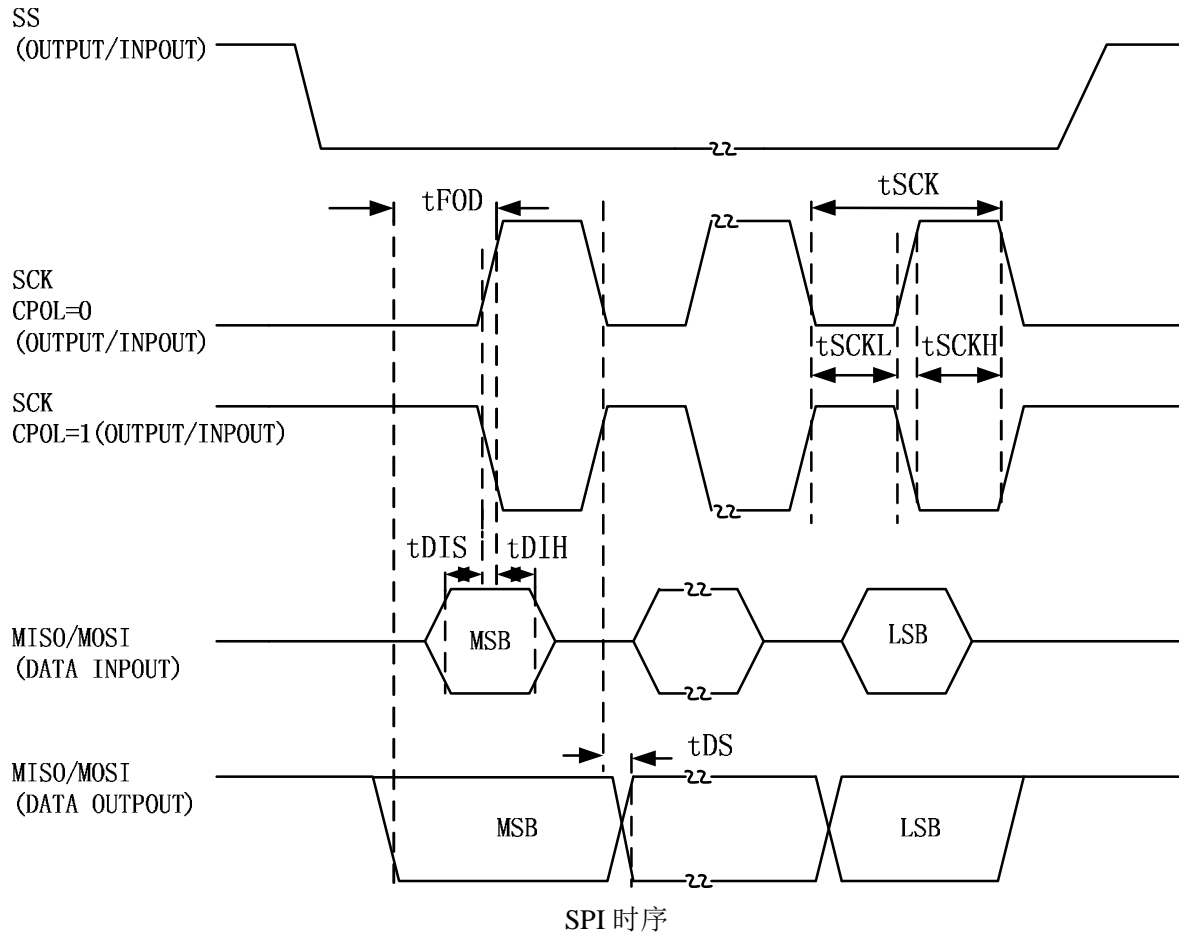
(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

参数名称	符号	测试条件	最小	典型	最大	单位
复位输入低电平宽度	$t_{RST}$	Input, VDD=5V	10			uS
中断输入高低宽度	$t_{IWH}, t_{IWL}$	所有中断, VDD=5V	200			nS
外部计数器输入高低脉冲宽度	$t_{ECWH}, t_{ECWL}$	ECn, VDD=5V(n=0, 1, 2)	200			
外部计数器转换时间	$t_{REC}, t_{FEC}$	ECn, VDD=5V(n=0, 1, 2)	20			



### 3.3.8、SPI 特性

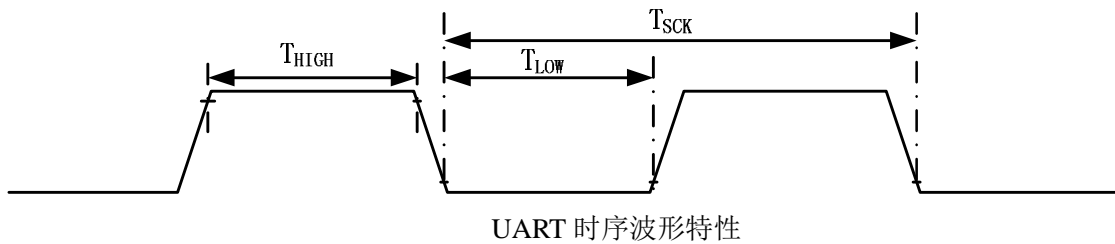
参数名称	符号	测试条件	最小	典型	最大	单位
输出脉冲周期	$t_{SCK}$	内部时钟源 SCK	200	--	--	nS
输入脉冲周期		外部时钟源 SCK	200	--	--	
输出时钟高低脉冲宽度	$t_{SCKH}, t_{SCKL}$	内部时钟源 SCK	70	--	--	
输入时钟脉冲高低宽度		外部时钟源 SCK	70	--	--	
第一个输出时钟延时时间	$t_{FOD}$	内/外时钟源 SCK	100	--	--	
输出时钟延时时间	$t_{DS}$	--	-	--	50	
输入设置时间	$t_{DIS}$	--	100	--	--	
输入维持时间	$t_{DIH}$	--	150	--	--	



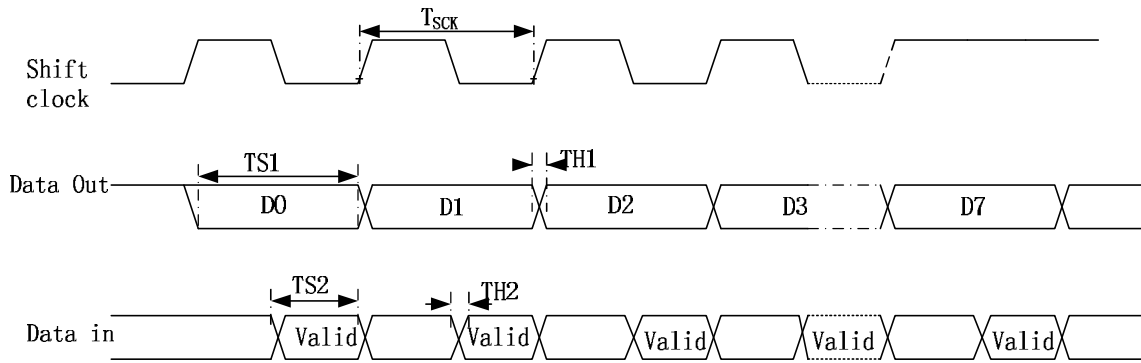
3.3.9、UART 特性

( $T_A = -40^{\circ}C \sim +85^{\circ}C$ ,  $V_{DD} = 1.8V \sim 5.5V$ ,  $f_{XIN} = 11.1MHz$ )

参数名称	符号	最小	典型	最大	单位
串口时钟周期	$t_{SCK}$	1250	$t_{CPU} * 16$	1650	nS
输出数据的设置到时钟上升沿	$t_{S1}$	590	$t_{CPU} * 13$	--	
时钟上升沿到输入有效数据时间	$t_{S2}$	--	--	590	
时钟上升沿之后输出数据维持时间	$t_{H1}$	$t_{CPU} - 50$	$t_{CPU}$	--	
时钟上升沿之后输入数据维持时间	$t_{H2}$	0	--	--	
串口时钟高低电平宽度	$t_{HIGH}, t_{LOW}$	470	$t_{CPU} * 8$	970	





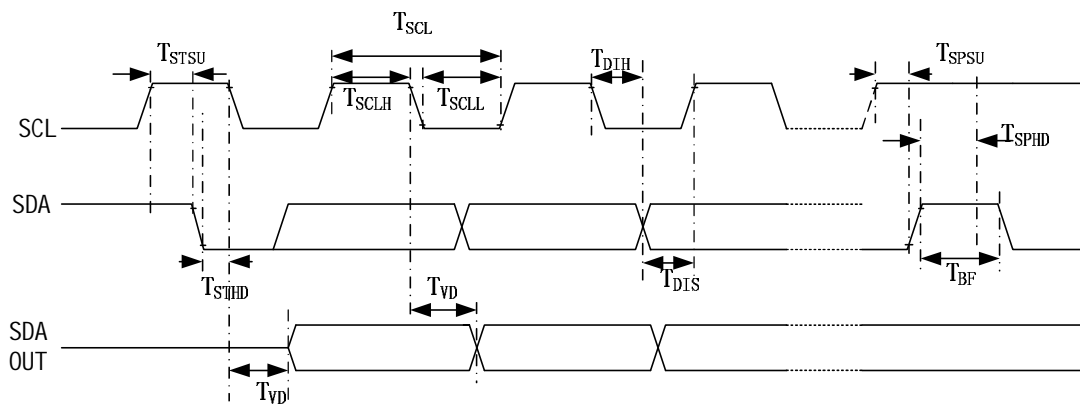


UART 模块时序波形特性

3.3.10、I2C 特性

( $T_A = -40^{\circ}C \sim +85^{\circ}C$ ,  $V_{DD} = 1.8V \sim 5.5V$ )

参数名称	符号	标准模式		高速模式		单位
		最小	最大	最小	最大	
时钟频率	$t_{SCL}$	0	100	0	400	KHz
时钟高脉冲宽度	$t_{SCLH}$	4		0.6		uS
时钟低脉冲宽度	$t_{SCLL}$	4.7		1.3		
总线空闲时间	$t_{BF}$	4.7		1.3		
开始条件设置时间	$t_{STSU}$	4.7		0.6		
开始条件维持时间	$t_{STHD}$	4		0.6		
停止条件设置时间	$t_{SPSU}$	4		0.6		
停止条件维持时间	$t_{SPHD}$	4		0.6		
时钟有效输出	$t_{VD}$	0		0		
数据输入维持时间	$t_{DIH}$	0		0	1.0	
数据输入设置时间	$t_{DIS}$	250		100		

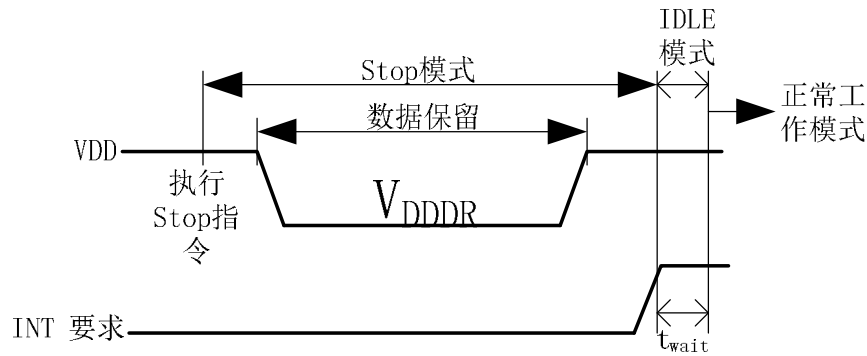


I<sup>2</sup>C 时序

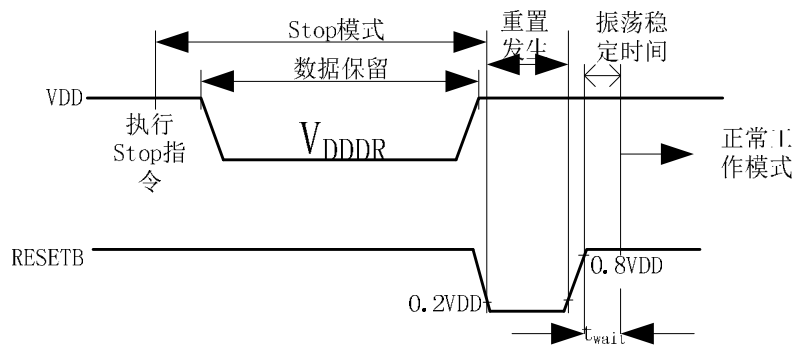
3.3.11、Stop 模式数据保持电压

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

参数名称	符号	测试条件	最小	典型	最大	单位
保持数据电压	V <sub>DDDR</sub>		1.8		5.5	V
保持数据电压	I <sub>DDDR</sub>	V <sub>DDDR</sub> = 1.8V, (TA= 25°C), Stop mode			1	uA



Stop 模式被中断唤醒时序



Stop 模式被复位唤醒时序

3.3.12、内部 Flash Rom 特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

参数名称	符号	测试条件	最小	典型	最大	单位
块写时间	t <sub>FSW</sub>	--	--	2.5	2.7	mS
块擦除时间	t <sub>FSE</sub>	--	--	2.5	2.7	
Hard-Lock Time	t <sub>FHL</sub>	--	--	2.5	2.7	
页缓存Buffer 复位时间	t <sub>FBR</sub>	--	--	--	5	uS
Flash 烧写频率	f <sub>PGM</sub>	--	0.4	--	--	MHz
擦写次数	N <sub>FWE</sub>	--	--	--	100000	Times

注意)在 flash 操作时, SCCR 中的 SCLK[1:0]必须设置为“00”or“01”(INT-RC OSC 或 Main X-TAL for system clock)。

3.3.13、输入/输出电容

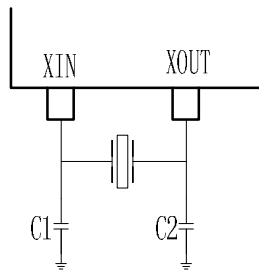
(TA= -40°C ~ +85°C, VDD= 0V)

参数名称	符号	测试条件	最小	典型	最大	单位
输入电容	$C_{IN}$	f <sub>x</sub> = 1MHz 不可测量 的引脚均连到VSS			10	pF
输出电容	$C_{OUT}$					
I/O 口电容	$C_{IO}$					

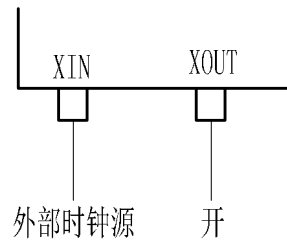
### 3.3.14、主时钟振荡特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

参数名称	符号	测试条件	最小	典型	最大	单位
晶振	主振荡频率	1.8V~5.5V	0.4		4.2	MHz
		2.7V~5.5V	0.4		10	
		3.0V~5.5V	0.4		12	
陶振	主振荡频率	1.8V~5.5V	0.4		4.2	MHz
		2.7V~5.5V	0.4		10	
		3.0V~5.5V	0.4		12	
外部时钟	XIN输入频率	1.8V~5.5V	0.4		1.2	MHz
		2.7V~5.5V	0.4		10	
		3.0V~5.5V	0.4		12	



晶体/陶瓷振荡器

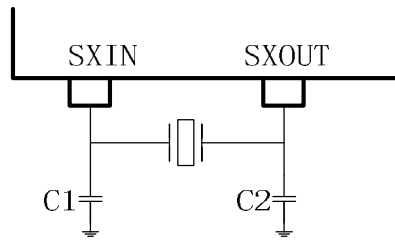


外部时钟

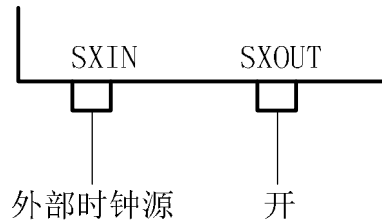
### 3.3.15、副时钟振荡特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

参数名称	晶振	测试条件	最小	典型	最大	单位
副振荡频率	晶体	1.8V – 5.5V	32	32.768	38	KHz
SXIN 输入频率	外部时钟		32	-	100	KHz



晶体振荡器



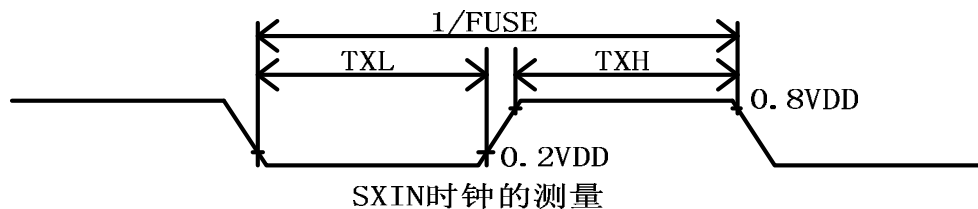
外部时钟

3.3.16、主振荡稳定特性

晶振	参数名称	最小	典型	最大	单位
晶体	f <sub>x</sub> > 1MHz 当VDD等于最小振荡电压等级时开始振荡稳定	--	--	60	mS
陶振		--	--	10	mS
外部时钟	f <sub>XIN</sub> = 0.4 to 12MHz XIN输入高低宽度(t <sub>XH</sub> , t <sub>XL</sub> )	42	--	1250	ns

3.3.17、副振荡特性

晶振	参数名称	最小	典型	最大	单位
晶振	--	--	--	10	S
外部时钟	SXIN 输入高低宽度(t <sub>XH</sub> , t <sub>XL</sub> )	5	--	15	ns



SXIN时钟的测量

4、CPU 说明

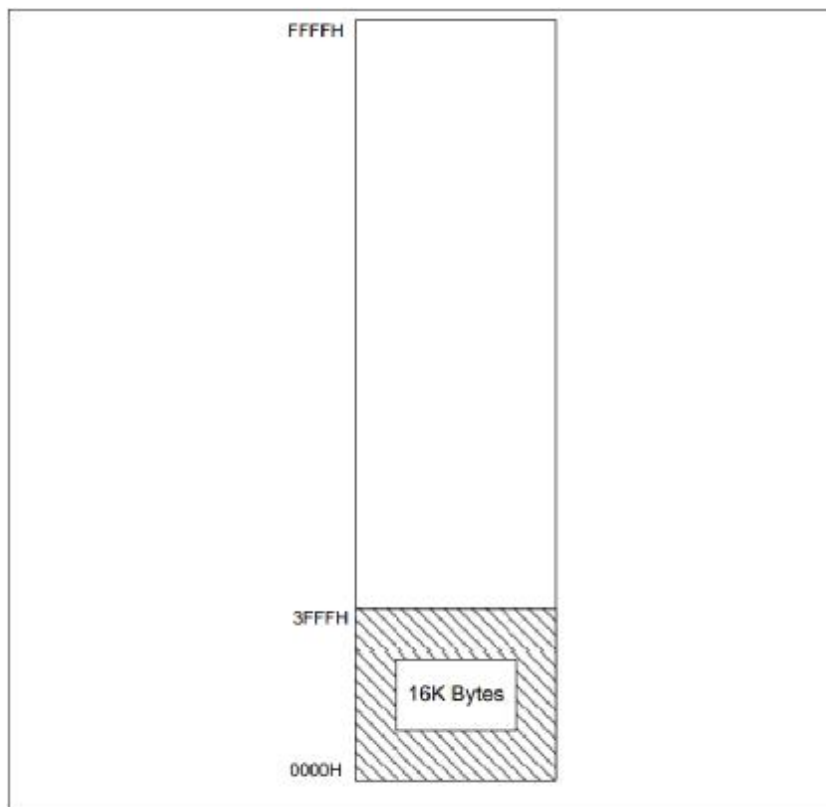
YF8316 16/32 一款有 16k/32k 字节 FLASH 的 8 位单片机。本芯片是一款可以为许多嵌入式控制应用提供高效灵活低成本解决方案的微控器，具有两个独立的地址存储空间：程序存储器和数据存储器。合理的分配程序存储器和允许 8 位存储的数据存储器，这样就使得 8 位 CPU 进入数据存储器更加迅速。不过，16 位数据存储地址同样可以通过 DPTR 寄存器进行存储。

YF8316 16/32 具备片内 16k/32k 字节 ISP 类型的可读可写 flash 存储。IRAM 是 256bytes 包括堆栈区。XRAM 是 512bytes。

#### 4.1、程序存储器

一个 16-bit 程序计数器寻址最大可达 64k bytes，但是此芯片只有 16k bytes 程序存储空间。

如下图所示为程序存储器较低区域。复位后，CPU 从 0000H 开始执行。每个中断在程序存储器中都分配有固定的地址。中断使 CPU 跳到该地址开始执行服务程序。例如外部中断 11，被分配地址为 000BH。如果使用外部中断 11，它的服务程序必须从 000BH 位置开始。如果没有用到该中断，其服务地址作为普通程序存储地址使用。

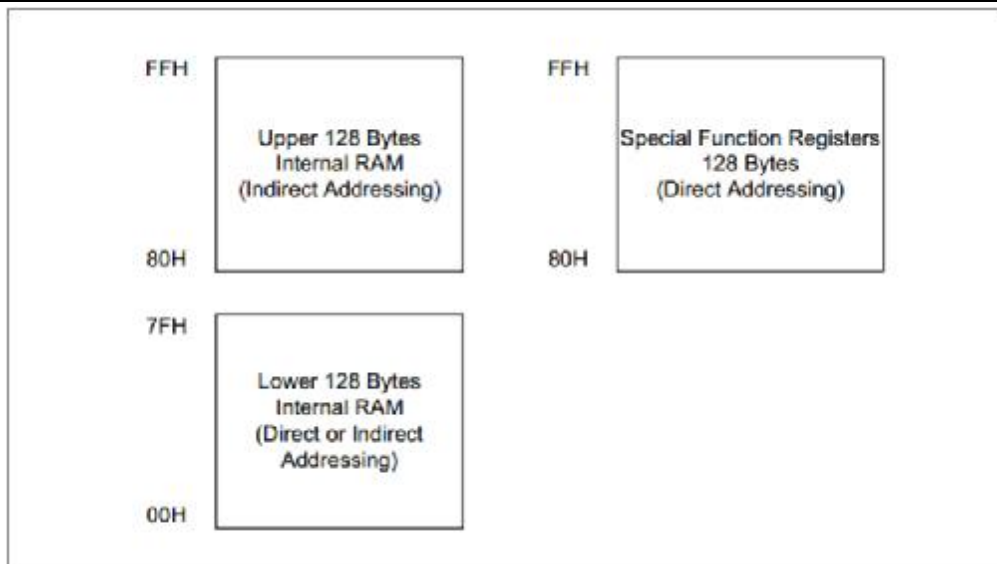


程序存储器

—— 16k Bytes 包括中断向量区域

#### 4.2、数据存储器

下图所示为内部数据可用数据存储空间：



Data Memory 图

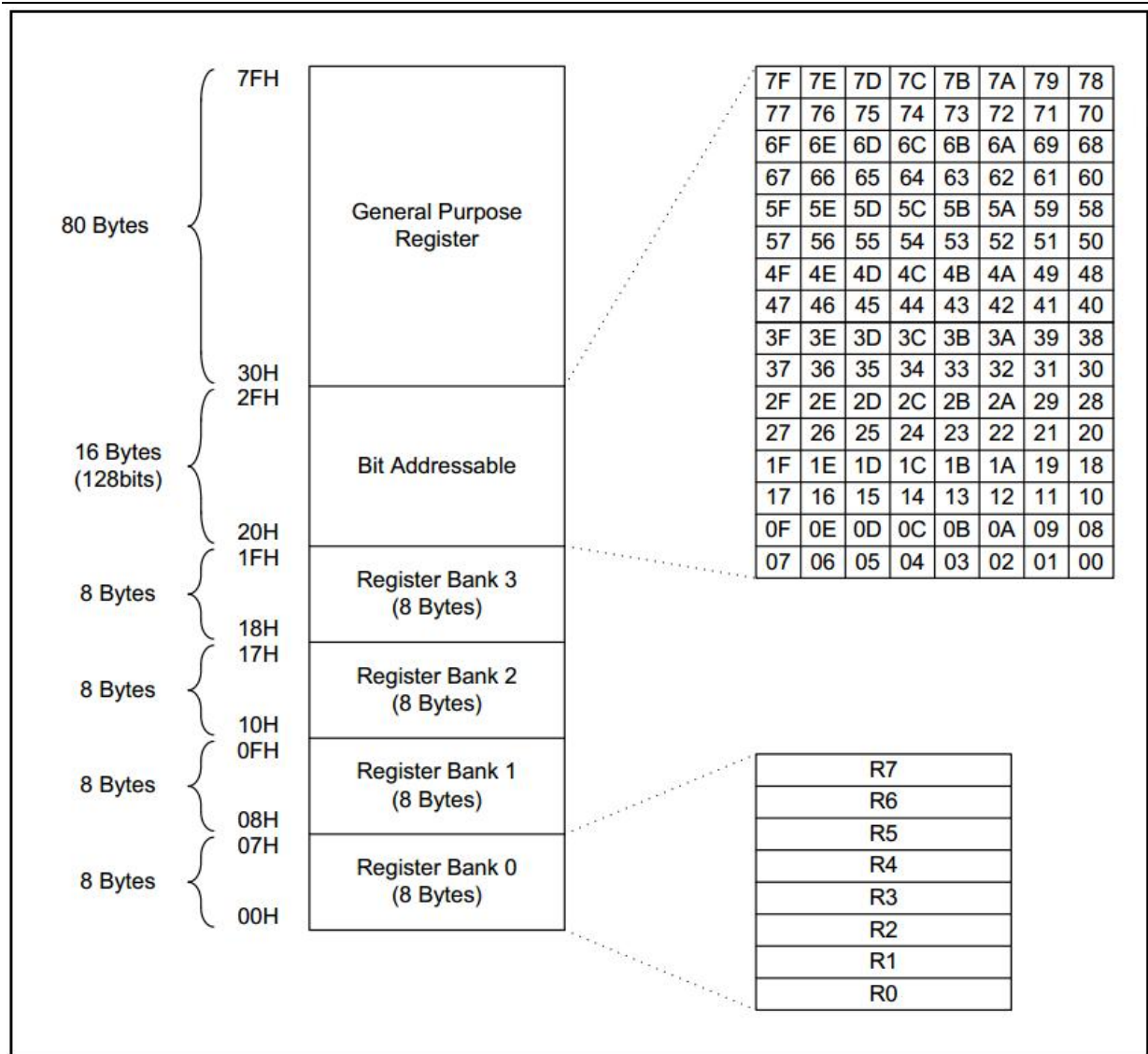
内部数据存储器分为 3 个部分，低 128bytes，高 128bytes，SFR。

内部数据存储地址是一个字节宽度，这就意味着地址空间只有 256bytes。但是事实上通过内部 RAM 寻址方式调节可提供 384byte。直接寻址高于 7FH 的存储空间和间接寻址高于 7FH 的存储空间进入到不同的存储空间。下图高 128bytes 和 SFR 占用相同的区域，80H 到 FFH，但他们本身却是独立的。

图中显示的是所有 8051 系列的低 128 字节 RAM。最低的 32 字节组成了 4 个寄存器组。程序指令可以调用从 R0 到 R7 的寄存器。程序状态字中的两位决定选用哪个寄存器组。这样可以更有效的使用代码空间，因为寄存器指令要比直接寻址指令更短。

寄存器组后边的 16bytes 组成了可位寻址的存储空间。8051 指令设置包括了位指令宽度的操作，但是这个区域的 128bits 可以通过这些指令直接寻址。该区域的地址是 00H 到 7FH。

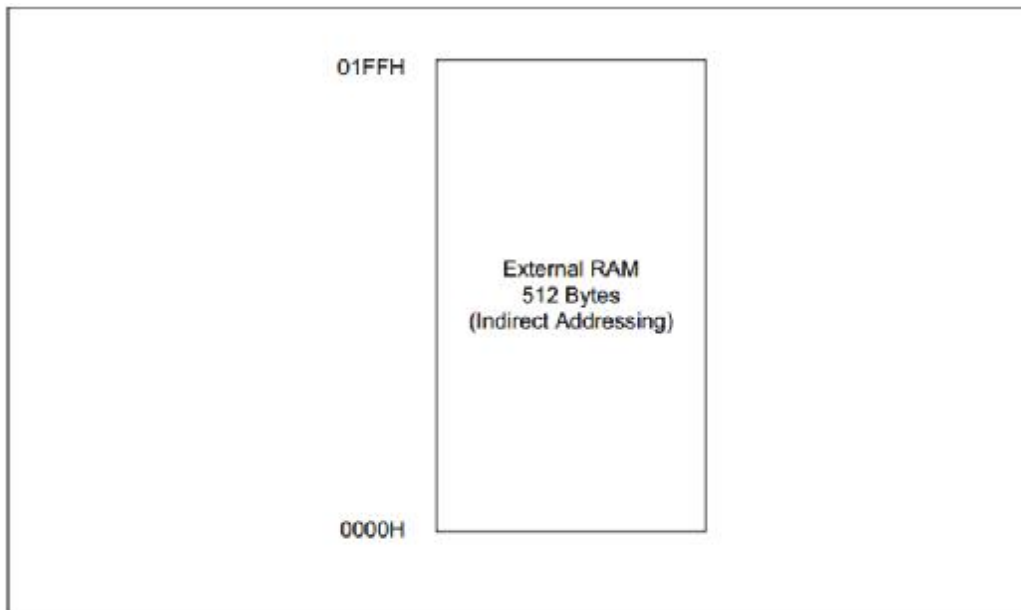
低 128 字节的所有字节均可以直接访问或间接访问。高 128 字节只能间接寻址访问。这些区域用作数据 RAM 和堆栈。



Lower 128 Bytes RAM

### 4.3、XRAM 存储器

YF8316 16/32 有 512 个字节 XRAM。该区域与 RAM/FLASH 没有联系。可通过 8 位的 SFR 进行读写。



XDATA 存储区

#### 4.4、SFR 图



4.4.1、SFR 摘要

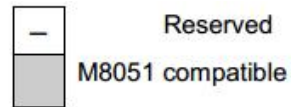


表 8-1 SFR 摘要

	00H/8H <sup>(1)</sup>	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
0F8H	IP1	-	FSADRH	FSADRM	FSADRL	FIDR	FMCR	-
0F0H	B	I2CSAR1	ADWRCCR0	ADWRCCR1	ADWRCCR2	ADWRCCR3	ADWCRL	ADWCRH
0E8H	RSTFR	I2CCR	I2CSR	I2CSAR0	I2CDR	I2CSDHR	I2CSCLR	I2CSCHR
0E0H	ACC	-	UARTCR1	UARTCR2	UARTCR3	UARTST	UARTBD	UARTDR
0D8H	LVRCCR	-	-	-	ADWIFRL	ADWIFRH	P03DB	P12DB
0D0H	PSW	-	-	P0FSR	P1FSRL	P1FSRH	P2FSR	P3FSR
0C8H	OSCCR	-	-	-	-	-	-	-
0C0H	EIFLAG0	P3IO	T2CRL	T2CRH	T2ADRL	T2ADRH	T2BDRL	T2BDRH
0B8H	IP	P2IO	T1CRL	T1CRH	T1ADRL	T1ADRH	T1BDRL	T1BDRH
0B0H	EIFLAG1	P1IO	T0CR	T0CNT	T0DR/ T0CDR	SPICR	SPIDR	SPISR
0A8H	IE	IE1	IE2	IE3	P0PU	P1PU	P2PU	P3PU
0A0H	IIFLAG	P0IO	EO	-	EIPOL0L	EIPOL0H	EIPOL1	EIPOL2
98H	P3	-	-	-	ADCCRL	ADCCRH	ADCDRL	ADCDRH
90H	P2	P0OD	P1OD	P2OD	P3OD	-	WTCR	BUZCR
88H	P1	WTDR/ WTCNT	SCCR	BITCR	BITCNT	WDTCR	WTDTR/ WTCNT	BUZDR
80H	P0	SP	DPL	DPH	DPL1	DPH1	LVICR	PCON

- 注意)
- 1.00H/8H，这一列的寄存器支持位操作。
  - 2.直接位判断并跳转的指令(见下面例子)不能用于 P0,P1,P2,P3,EIFLAG0,EIFLAG1 这几个寄存器。
  - 3.例子：IF(P00)→IF(P0&0x01)，左边的指令应该改成右边的指令形式。

4.4.2、SFR 图表

表 8-2 SFR 图表

Address	Function	Symbol	RW	@Reset									
				7	6	5	4	3	2	1	0		
80H	P0 Data Register	P0	RW	–	0	0	0	0	0	0	0	0	0
81H	Stack Pointer	SP	RW	0	0	0	0	0	0	1	1	1	1
82H	Data Pointer Register Low	DPL	RW	0	0	0	0	0	0	0	0	0	0
83H	Data Pointer Register High	DPH	RW	0	0	0	0	0	0	0	0	0	0
84H	Data Pointer Register Low 1	DPL1	RW	0	0	0	0	0	0	0	0	0	0
85H	Data Pointer Register High 1	DPH1	RW	0	0	0	0	0	0	0	0	0	0
86H	Low Voltage Indicator Control Register	LVICR	RW	–	–	0	0	0	0	0	0	0	0
87H	Power Control Register	PCON	RW	0	–	–	–	0	0	0	0	0	0
88H	P1 Data Register	P1	RW	0	0	0	0	0	0	0	0	0	0
89H	Watch Timer Data Register	WTDR	W	0	1	1	1	1	1	1	1	1	1
	Watch Timer Counter Register	WTCNT	R	–	0	0	0	0	0	0	0	0	0
8AH	System and Clock Control Register	SCCR	RW	–	–	–	–	–	–	–	0	0	0
8BH	Basic Interval Timer Control Register	BITCR	RW	0	0	0	–	0	0	0	0	0	1
8CH	Basic Interval Timer Counter Register	BITCNT	R	0	0	0	0	0	0	0	0	0	0
8DH	Watch Dog Timer Control Register	WDTCR	RW	0	0	0	–	–	–	–	0	0	0
8EH	Watch Dog Timer Data Register	WDTDR	W	1	1	1	1	1	1	1	1	1	1
	Watch Dog Timer Counter Register	WDTCNT	R	0	0	0	0	0	0	0	0	0	0
8FH	BUZZER Data Register	BUZDR	RW	1	1	1	1	1	1	1	1	1	1
90H	P2 Data Register	P2	RW	–	0	0	0	0	0	0	0	0	0
91H	P0 Open-drain Selection Register	P0OD	RW	–	0	0	0	0	0	0	0	0	0
92H	P1 Open-drain Selection Register	P1OD	RW	0	0	0	0	0	0	0	0	0	0
93H	P2 Open-drain Selection Register	P2OD	RW	–	0	0	0	0	0	0	0	0	0
94H	P3 Open-drain Selection Register	P3OD	RW	0	0	0	0	0	0	0	0	0	0
95H	Reserved	–	–	–									
96H	Watch Timer Control Register	WTCR	RW	0	–	–	0	0	0	0	0	0	0
97H	BUZZER Control Register	BUZCR	RW	–	–	–	–	0	0	0	0	0	0
98H	P3 Data Register	P3	RW	0	0	0	0	0	0	0	0	0	0
99H	Reserved	–	–	–									
9AH	Reserved	–	–	–									
9BH	Reserved	–	–	–									
9CH	A/D Converter Control Low Register	ADCCRL	RW	0	0	0	0	0	0	0	0	0	0
9DH	A/D Converter Control High Register	ADCCRH	RW	0	–	–	0	0	0	0	0	0	0
9EH	A/D Converter Data Low Register	ADCRL	R	x	x	x	x	x	x	x	x	x	x
9FH	A/D Converter Data High Register	ADCRH	R	x	x	x	x	x	x	x	x	x	x

表 8-2 SFR 图表 (续)

Address	Function	Symbol	RW	@Reset							
				7	6	5	4	3	2	1	0
A0H	Internal Interrupt Flag Register	IIFLAG	RW	-	-	-	-	-	0	0	0
A1H	P0 Direction Register	P0IO	RW	-	0	0	0	0	0	0	0
A2H	Extended Operation Register	EO	RW	-	-	-	0	-	0	0	0
A3H	Reserved	-	-	-							
A4H	External Interrupt Polarity 0 Low Register	EIPOL0L	RW	0	0	0	0	0	0	0	0
A5H	External Interrupt Polarity 0 High Register	EIPOL0H	RW	-	-	0	0	0	0	0	0
A6H	External Interrupt Polarity 1 Register	EIPOL1	RW	-	-	0	0	0	0	0	0
A7H	External Interrupt Polarity 2 Register	EIPOL2	RW	0	0	0	0	0	0	0	0
A8H	Interrupt Enable Register	IE	RW	0	-	0	-	-	0	0	0
A9H	Interrupt Enable Register 1	IE1	RW	-	-	0	0	0	0	0	0
AAH	Interrupt Enable Register 2	IE2	RW	-	-	0	-	0	0	0	0
ABH	Interrupt Enable Register 3	IE3	RW	-	-	-	0	0	0	0	0
ACH	P0 Pull-up Resistor Selection Register	P0PU	RW	-	0	0	0	0	0	0	0
ADH	P1 Pull-up Resistor Selection Register	P1PU	RW	0	0	0	0	0	0	0	0
AEH	P2 Pull-up Resistor Selection Register	P2PU	RW	-	0	0	0	0	0	0	0
AFH	P3 Pull-up Resistor Selection Register	P3PU	RW	0	0	0	0	0	0	0	0
B0H	External Interrupt Flag 1 Register	EIFLAG1	RW	-	0	0	0	0	0	0	0
B1H	P1 Direction Register	P1IO	RW	0	0	0	0	0	0	0	0
B2H	Timer 0 Control Register	T0CR	RW	0	-	0	0	0	0	0	0
B3H	Timer 0 Counter Register	T0CNT	R	0	0	0	0	0	0	0	0
B4H	Timer 0 Data Register	T0DR	RW	1	1	1	1	1	1	1	1
	Timer 0 Capture Data Register	T0CDR	R	0	0	0	0	0	0	0	0
B5H	SPI Control Register	SPICR	RW	0	0	0	0	0	0	0	0
B6H	SPI Data Register	SPIDR	RW	0	0	0	0	0	0	0	0
B7H	SPI Status Register	SPISR	RW	0	0	0	-	0	0	-	-
B8H	Interrupt Priority Register	IP	RW	-	-	0	0	0	0	0	0
B9H	P2 Direction Register	P2IO	RW	-	0	0	0	0	0	0	0
BAH	Timer 1 Control Low Register	T1CRL	RW	0	0	0	0	-	0	0	0
BBH	Timer 1 Control High Register	T1CRH	RW	0	-	0	0	-	-	-	0
BCH	Timer 1 A Data Low Register	T1ADRL	RW	1	1	1	1	1	1	1	1
BDH	Timer 1 A Data High Register	T1ADRH	RW	1	1	1	1	1	1	1	1
BEH	Timer 1 B Data Low Register	T1BDRL	RW	1	1	1	1	1	1	1	1
BFH	Timer 1 B Data High Register	T1BDRH	RW	1	1	1	1	1	1	1	1

表 8-2 SFR 图表 (续)

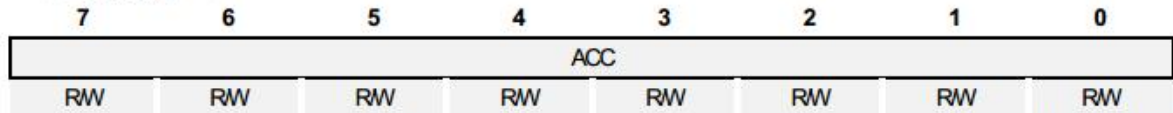
Address	Function	Symbol	RW	@Reset								
				7	6	5	4	3	2	1	0	
C0H	External Interrupt Flag 0 Register	EIFLAG0	RW	-	0	0	0	0	0	0	0	0
C1H	P3 Direction Register	P3IO	RW	0	0	0	0	0	0	0	0	0
C2H	Timer 2 Control Low Register	T2CRL	RW	0	0	0	0	0	0	0	0	0
C3H	Timer 2 Control High Register	T2CRH	RW	0	-	0	0	-	-	-	-	0
C4H	Timer 2 A Data Low Register	T2ADRL	RW	1	1	1	1	1	1	1	1	1
C5H	Timer 2 A Data High Register	T2ADRH	RW	1	1	1	1	1	1	1	1	1
C6H	Timer 2 B Data Low Register	T2BDRL	RW	1	1	1	1	1	1	1	1	1
C7H	Timer 2 B Data High Register	T2BDRH	RW	1	1	1	1	1	1	1	1	1
C8H	Oscillator Control Register	OSCCR	RW	-	-	0	0	1	0	0	0	0
C9H	Reserved	-	-	-								
CAH	Reserved	-	-	-								
CBH	Reserved	-	-	-								
CCH	Reserved	-	-	-								
CDH	Reserved	-	-	-								
CEH	Reserved	-	-	-								
CFH	Reserved	-	-	-								
D0H	Program Status Word Register	PSW	RW	0	0	0	0	0	0	0	0	0
D1H	Reserved	-	-	-								
D2H	Reserved	-	-	-								
D3H	P0 Function Selection Register	P0FSR	RW	0	0	0	0	0	0	0	0	0
D4H	P1 Function Selection Low Register	P1FSRL	RW	-	0	0	0	0	0	0	0	0
D5H	P1 Function Selection High Register	P1FSRH	RW	-	0	0	0	0	0	0	0	0
D6H	P2 Function Selection Register	P2FSR	RW	-	-	-	-	-	-	0	0	0
D7H	P3 Function Selection Register	P3FSR	RW	0	0	0	0	0	0	0	0	0
D8H	Low Voltage Reset Control Register	LVRCCR	RW	0	-	-	0	0	0	0	0	0
D9H	Reserved	-	-	-								
DAH	Reserved	-	-	-								
DBH	Reserved	-	-	-								
DCH	ADC Wake-up Interrupt Flag Low Register	ADWIFRL	RW	0	0	0	0	0	0	0	0	0
DDH	ADC Wake-up Interrupt Flag High Register	ADWIFRH	RW	-	0	0	0	0	0	0	0	0
DEH	P03 Debounce Enable Register	P03DB	RW	0	0	0	0	0	0	0	0	0
DFH	P12 Debounce Enable Register	P12DB	RW	0	0	0	0	0	0	0	0	0

表 8-2 SFR 图表 (续)

Address	Function	Symbol	R/W	@Reset								
				7	6	5	4	3	2	1	0	
E0H	Accumulator A Register	ACC	R/W	0	0	0	0	0	0	0	0	0
E1H	Reserved	-	-	-								
E2H	UART Control Register 1	UARTCR1	R/W	-	-	0	0	0	0	0	0	-
E3H	UART Control Register 2	UARTCR2	R/W	0	0	0	0	0	0	0	0	0
E4H	UART Control Register 3	UARTCR3	R/W	-	0	-	-	-	0	0	0	0
E5H	UART Status Register	UARTST	R/W	1	0	0	0	0	0	0	0	0
E6H	UART Baud Rate Generation Register	UARTBD	R/W	1	1	1	1	1	1	1	1	1
E7H	UART Data Register	UARTDR	R/W	0	0	0	0	0	0	0	0	0
E8H	Reset Flag Register	RSTFR	R/W	1	x	0	0	x	-	-	-	-
E9H	I2C Control Register	I2CCR	R/W	0	0	0	0	0	0	0	0	0
EAH	I2C Status Register	I2CSR	R/W	0	0	0	0	0	0	0	0	0
EBH	I2C Slave Address 0 Register	I2CSAR0	R/W	0	0	0	0	0	0	0	0	0
ECH	I2C Data Register	I2CDR	R/W	0	0	0	0	0	0	0	0	0
EDH	I2C SDA Hold Time Register	I2CSDHR	R/W	0	0	0	0	0	0	0	0	1
EEH	I2C SCL Low Period Register	I2CSCLR	R/W	0	0	1	1	1	1	1	1	1
EFH	I2C SCL High Period Register	I2CSCHR	R/W	0	0	1	1	1	1	1	1	1
F0H	B Register	B	R/W	0	0	0	0	0	0	0	0	0
F1H	I2C Slave Address 1 Register	I2CSAR1	R/W	0	0	0	0	0	0	0	0	0
F2H	ADC Wake-up Resistor Control Register0	ADWRCR0	R/W	0	0	0	0	0	0	0	0	0
F3H	ADC Wake-up Resistor Control Register1	ADWRCR1	R/W	0	0	0	0	0	0	0	0	0
F4H	ADC Wake-up Resistor Control Register2	ADWRCR2	R/W	0	0	0	0	0	0	0	0	0
F5H	ADC Wake-up Resistor Control Register3	ADWRCR3	R/W	-	-	0	0	0	0	0	0	0
F6H	ADC Wake-up Control Low Register	ADWCRL	R/W	0	0	0	0	0	0	0	0	0
F7H	ADC Wake-up Control High Register	ADWCRH	R/W	-	0	0	0	0	0	0	0	0
F8H	Interrupt Priority Register 1	IP1	R/W	-	-	0	0	0	0	0	0	0
F9H	Reserved	-	-	-								
FAH	Flash Sector Address High Register	FSADRH	R/W	-	-	-	-	0	0	0	0	0
FBH	Flash Sector Address Middle Register	FSADRM	R/W	0	0	0	0	0	0	0	0	0
FCH	Flash Sector Address Low Register	FSADRL	R/W	0	0	0	0	0	0	0	0	0
FDH	Flash Identification Register	FIDR	R/W	0	0	0	0	0	0	0	0	0
FEH	Flash Mode Control Register	FMCR	R/W	0	-	-	-	-	0	0	0	0
FFH	Reserved	-	-	-								

#### 4.4.3、编译器兼容的 SFR

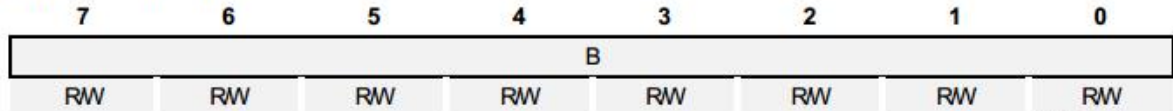
**ACC (累加器) : E0H**



Initial value : 00H

ACC          累加器

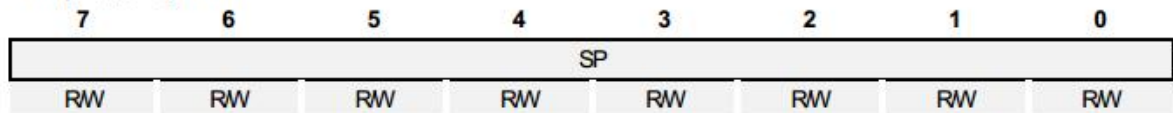
**B (B 寄存器) : F0H**



Initial value : 00H

B              B 寄存器

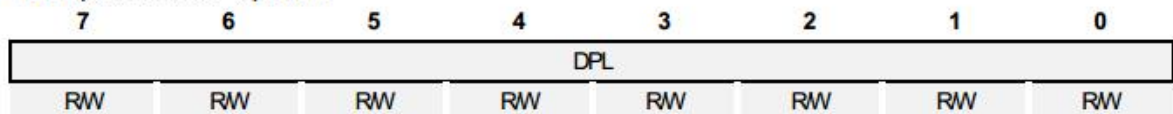
**SP (堆栈指针) : 81H**



Initial value : 07H

SP            堆栈指针

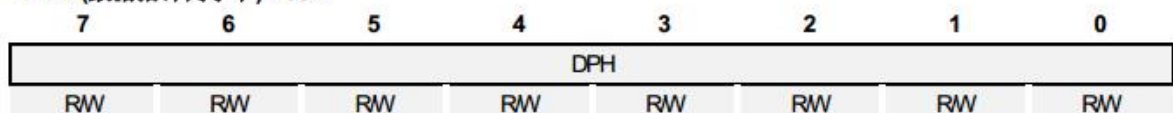
**DPL (数据指针低字节) : 82H**



Initial value : 00H

DPL          数据指针低字节

**DPH (数据指针高字节) : 83H**



Initial value : 00H

DPH          数据指针高字节

**DPL1 (数据指针寄存器低 1) : 84H**

7	6	5	4	3	2	1	0
DPL1							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

DPL1 Data Pointer Low 1

**DPH1 (数据指针寄存器高 1) : 85H**

7	6	5	4	3	2	1	0
DPH1							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

DPH1 Data Pointer High 1

**PSW (程序状态字) : D0H**

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

- CY 进位标志
- AC 辅助进位标志
- F0 通用用户可定义标志
- RS1 寄存器组选择位 1
- RS0 寄存器组选择位 0
- OV 溢出标志
- F1 用户可定义标志
- P 奇偶标志。每个指令周期通过硬件设置/清除来表示累加器中 1 的数量的奇偶

**EO (扩展操作寄存器) : A2H**

7	6	5	4	3	2	1	0
-	-	-	TRAP_EN	-	DPSEL2	DPSEL1	DPSEL0
-	-	-	RW	-	RW	RW	RW

Initial value : 00H

- TRAP\_EN 选择指令 (Keep always '0')。
  - 0 Select MOVC @(DPTR++), A
  - 1 Select Software TRAP Instruction
- DPSEL[2:0] 选择数据指针寄存器组
 

DPSEL2	DPSEL1	DPSEL0	Description
0	0	0	DPTR0
0	0	1	DPTR1
Reserved			

**5、I/O 口**

## 5.1、I/O 口

AiP81F2016 有 4 组 I/O 口(P0~P3)。通过软件设置可以轻松的把每个口配置为 I/O 口，内部上拉和漏极开路以匹配不同的系统结构和设计要求。P0,P1,P2,和 P3 同样具有根据引脚状态变化引起中断的功能。

## 5.2、端口寄存器

### 5.2.1、数据寄存器 (Px)

数据寄存器是双向 I/O 口。如果作为输出口使用，数据可以写入到 Px 的相应位。如果设置为输入口，数据可以 Px 的相应位读到。

### 5.2.2、方向寄存器 (PxIO)

每个 I/O 口可以通过 PxIO 寄存器单独的作为输入或输出来使用，清除相应的位将会使其变为输入模式。置位相应的位就会使其变为输出口。系统复位后可以清除大部分位，但是一些不能被清除。

### 5.2.3、上拉电阻选择寄存器(PxPU)

片内上拉寄存器可以单独的将其和上拉寄存器连接(PxPU)。上拉寄存器选择控制每个口的上拉寄存器的使能和关闭。相应的位置 1，这个脚的上拉寄存器就打开。如果清零，上拉就关闭。所有的位复位后清零。

### 5.2.4、漏极开路寄存器 (PxOD)

P0~P3 具有内部漏极开路选择寄存器。控制着每个口漏极开路的开关。复位后大部分端口变为推-拉状态，但是一些口变为漏极开路状态。

### 5.2.5、去抖使能寄存器 (PxDB)

P0[6:2],P1[3:0],P2[3:0]和 P35 具有去抖功能。去抖时钟是  $fx/1$ , $fx/4$ ,和  $fx/4096$ 。

### 5.2.6、端口功能选择寄存器 (PxFSR)

这些寄存器是用来定义端口的复用功能，请记得将这些寄存器设置到相应的功能。复位后 PxFSR 被清除到 '00H'，使所有口变为普通 I/O 口。

## 5.2.7、寄存器图表

### 端口寄存器表



Name	Address	Dir	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	A1H	R/W	00H	P0 Direction Register
P0OD	91H	R/W	00H	P0 Open-drain Selection Register
P0PU	ACH	R/W	00H	P0 Pull-up Resistor Selection Register
P03DB	DEH	R/W	00H	P0/P3 Debounce Enable Register
P0FSR	D3H	R/W	00H	P0 Function Selection Register
P1	88H	R/W	00H	P1 Data Register
P1IO	B1H	R/W	00H	P1 Direction Register
P1OD	92H	R/W	00H	P1 Open-drain Selection Register
P1PU	ADH	R/W	00H	P1 Pull-up Resistor Selection Register
P12DB	DFH	R/W	00H	P1/P2 Debounce Enable Register
P1FSRH	D5H	R/W	00H	P1 Function Selection High Register
P1FSRL	D4H	R/W	00H	P1 Function Selection Low Register
P2	90H	R/W	00H	P2 Data Register
P2IO	B9H	R/W	00H	P2 Direction Register
P2OD	93H	R/W	00H	P2 Open-drain Selection Register
P2PU	AEH	R/W	00H	P2 Pull-up Resistor Selection Register
P2FSR	D6H	R/W	00H	P2 Function Selection Register
P3	98H	R/W	00H	P3 Data Register
P3IO	C1H	R/W	00H	P3 Direction Register
P3OD	94H	R/W	00H	P3 Open-drain Selection Register
P3PU	AFH	R/W	00H	P3 Pull-up Resistor Selection Register
P3FSR	D7H	R/W	00H	P3 Function Selection Register

### 5.3、P0 口

#### 5.3.1、P0 口说明

P0 是 7-bit I/O 口。P0 控制寄存器由 P0 数据寄存器(P0)，P0 方向寄存器(P0IO)，防抖使能寄存器(P03DB)，P0 上拉电阻选择寄存器(P0PU)，和 P0 漏极开路寄存器(P0OD)组成。具体请参考端口功能选择寄存器。

#### 5.3.2、P0 寄存器说明

**P0 (P0 数据寄存器) : 80H**

7	6	5	4	3	2	1	0
-	P06	P05	P04	P03	P02	P01	P00
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0[6:0] I/O Data

注意：直接位判断并跳转的指令(见下面例子)不能用于输入口。

例子：if(P00)→if(P0&0x01)，左边的指令应该改成右边的指令形式。

**P0IO (P0 方向寄存器) : A1H**

7	6	5	4	3	2	1	0
-	P06IO	P05IO	P04IO	P03IO	P02IO	P01IO	P00IO
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0IO[6:0] P0 口 I/O 方向  
 0 输入  
 1 输出

注意: EINT0~EINT4 功能在输入状态时才有

**P0PU (P0 上拉电阻选择寄存器) : ACH**

7	6	5	4	3	2	1	0
-	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0PU[6:0] 配置 P0 口上拉电阻  
 0 Disable  
 1 Enable

**P0OD (P0 漏极开路寄存器) : 91H**

7	6	5	4	3	2	1	0
-	P06OD	P05OD	P04OD	P03OD	P02OD	P01OD	P00OD
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0OD[6:0] 配置 P0 口漏极开路  
 0 推-拉输出  
 1 漏极开路输出

**P03DB (P0/P3 去抖使能寄存器) : DEH**

7	6	5	4	3	2	1	0
DBCLK1	DBCLK0	P35DB	P06DB	P05DB	P04DB	P03DB	P02DB
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

DBCLK[1:0] 配置端口消抖时钟

DBCLK1	DBCLK0	Description
0	0	fx/1
0	1	fx/4
1	0	fx/4096
1	1	Reserved

P35DB 配置 P35 口消抖  
 0 Disable  
 1 Enable

P06DB 配置 P06 口消抖

	0	Disable
	1	Enable
P05DB	配置 P05 口消抖	
	0	Disable
	1	Enable
P04DB	配置 P04 口消抖	
	0	Disable
	1	Enable
P03DB	配置 P03 口消抖	
	0	Disable
	1	Enable
P02DB	配置 P02 口消抖	
	0	Disable
	1	Enable

- 注意)
1. 如果在使能引脚上时钟采样时相同的电平没有被连续三到四次检测到，那么这个信号就被作为干扰淘汰掉。
  2. 一个输入脉冲在整个过程中应该被检测到 3 个以上的时钟才被认为是有效。
  3. 在 STOP 模式端口消抖被自动的关闭 STOP 唤醒之后又重新恢复。

## 5.4、P1 口

### 5.4.1、P1 口说明

P1 是 8-bit I/O 口。P1 控制寄存器由 P1 数据寄存器(P1)，P1 方向寄存器(P1IO)，消抖使能寄存器(P12DB)，P1 上拉电阻选择寄存器(P1PU)，和 P1 漏极开路(P1OD)组成，具体请参考端口功能寄存器。

### 5.4.2、P1 寄存器说明

**P1 (P1 数据寄存器) : 88H**

7	6	5	4	3	2	1	0
P17	P16	P15	P14	P13	P12	P11	P10
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1[7:0] I/O Data

注意：直接位判断并跳转的指令(见下面例子)不能用于输入口。

例子：if(P10)→if(P1&0x01)，左边的指令应该改成右边的指令形式。

**P1IO (P1 方向寄存器) : B1H**

7	6	5	4	3	2	1	0
P17IO	P16IO	P15IO	P14IO	P13IO	P12IO	P11IO	P10IO
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1IO[7:0] P1 口 I/O 方向  
0 Input

1 Output

注意: EINT5/ENINT6/EINT11/EINT12/EC1/EC2/SS 功能在输入状态时才有。

**P1PU (P1 上拉电阻选择寄存器) : ADH**

7	6	5	4	3	2	1	0
P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1PU[7:0] 配置 P1 口上拉电阻

0 Disable

1 Enable

**P1OD (P1 漏极开路) : 92H**

7	6	5	4	3	2	1	0
P17OD	P16OD	P15OD	P14OD	P13OD	P12OD	P11OD	P10OD
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1OD[7:0] 配置 P1 口漏极开路

0 推-拉输出

1 漏极开路输出

**P12DB (P1/P2 消抖使能寄存器) : DFH**

7	6	5	4	3	2	1	0
P23DB	P22DB	P21DB	P20DB	P13DB	P12DB	P11DB	P10DB
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P23DB 配置 P23 口消抖

0 Disable

1 Enable

P22DB 配置 P22 口消抖

0 Disable

1 Enable

P21DB 配置 P21 口消抖

0 Disable

1 Enable

P20DB 配置 P20 口消抖

0 Disable

1 Enable

P13DB 配置 P13 口消抖

0 Disable

1 Enable

P12DB 配置 P12 口消抖

0 Disable

	1	Enable
P11DB	配置 P11 口消抖	
	0	Disable
	1	Enable
P10DB	配置 P10 口消抖	
	0	Disable
	1	Enable

- 注意)
1. 如果在使能引脚上时钟采样时相同的电平没有被连续三到四次检测到，那么这个信号就被作为干扰淘汰掉。
  2. 一个输入脉冲在整个过程中应该被检测到 3 个以上的时钟才被认为是有效。
  3. 在 STOP 模式端口消抖被自动的关闭 STOP 唤醒之后又重新恢复。
  4. 请参考 P0 和 P3 口消抖时钟。

### 5.5、P2 口

#### 5.5.1、P2 口说明

P2 是 7-bit I/O 口。P2 控制寄存器由 P2 数据寄存器(P2)，P2 方向寄存器(P2IO)，P2 上拉电阻选择寄存器(P2PU)和 P2 漏极开路选择寄存器(P2OD)组成，具体请参考端口功能寄存器。

#### 5.5.2、P2 寄存器说明

##### P2 (P2 数据寄存器) : 90H

7	6	5	4	3	2	1	0
-	P26	P25	P24	P23	P22	P21	P20
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P2[6:0] I/O Data

注意：直接位判断并跳转的指令(见下面例子)不能用于输入口。

例子：if(P20)→if(P2&0x01)，左边的指令应该改成右边的指令形式。

##### P2IO (P2 方向寄存器) : B9H

7	6	5	4	3	2	1	0
-	P26IO	P25IO	P24IO	P23IO	P22IO	P21IO	P20IO
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P2IO[6:0] P2 Data I/O Direction

0 Input

1 Output

注意：EINT7-EINTA,EC0 在输入状态时才有

**P2PU (P2 上拉电阻选择寄存器) : AEH**

7	6	5	4	3	2	1	0
-	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P2PU[6:0] 配置 P2 口上拉电阻

0 Disable

1 Enable

**P2OD (P2 漏极开路输出) : 93H**

7	6	5	4	3	2	1	0
-	P26OD	P25OD	P24OD	P23OD	P22OD	P21OD	P20OD
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P2OD[6:0] 配置 P2 口漏极开路

0 推-拉输出

1 漏极开路输出

**5.6、P3 口****5.6.1、P3 口说明**

P3 是 8-bit I/O 口。P3 控制寄存器由 P3 数据寄存器(P3)，P3 方向寄存器(P3IO)。P3 上拉电阻选择寄存器(P3PU)和 P3 漏极开路选择寄存器(P3OD)组成，具体请参考端口功能寄存器。

**5.6.2、P3 端口寄存器说明****P3 (P3 数据寄存器) : 98H**

7	6	5	4	3	2	1	0
P37	P36	P35	P34	P33	P32	P31	P30
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P3[7:0] I/O Data

注意：直接位判断并跳转的指令(见下面例子)不能用于输入口。

例子：if(P30)→if(P3&0x01)，左边的指令应该改成右边的指令形式。

**P3IO (P3 方向寄存器) : C1H**

7	6	5	4	3	2	1	0
P37IO	P36IO	P35IO	P34IO	P33IO	P32IO	P31IO	P30IO
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P3IO[7:0] P3 口方向

0 Input

1 Output

注意: RXD, EINT10 功能只有在输入状态时才有。

**P3PU (P3 上拉电阻选择寄存器) : AFH**

7	6	5	4	3	2	1	0
P37PU	P36PU	P35PU	P34PU	P33PU	P32PU	P31PU	P30PU
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P3PU[7:0]      配置 P3 口上拉电阻  
 0      Disable  
 1      Enable

**P3OD (P3 漏极开路选择寄存器) : 94H**

7	6	5	4	3	2	1	0
P37OD	P36OD	P35OD	P34OD	P33OD	P32OD	P31OD	P30OD
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P3OD[7:0]      配置 P3 口漏极开路  
 0      推-拉输出  
 1      漏极开路输出

**5.7、端口功能**

**5.7.1、端口功能说明**

端口功能寄存器由 0~3(P0FSR~P3FSR)组成。

**5.7.2、P0FSR ~ P3FSR**

**P0FSR (Port 0 功能选择寄存器) : D3H**

7	6	5	4	3	2	1	0
P0FSR7	P0FSR6	P0FSR5	P0FSR4	P0FSR3	P0FSR2	P0FSR1	P0FSR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0FSR7    P06    功能选择  
 0      I/O Port (EINT4 功能在输入状态时才有)  
 1      AN6

P0FSR6    P05    功能选择  
 0      I/O Port (EINT 功能在输入状态时才有)  
 1      AN5

P0FSR5    P04    功能选择  
 0      I/O Port (EINT 功能在输入状态时才有)  
 1      AN4

P0FSR4    P03    功能选择  
 0      I/O Port (EINT 功能在输入状态时才有)  
 1      AN3

P0FSR[3:2]    P02    功能选择

P0FSR3	P0FSR2	Description
0	0	I/O Port (EINT 功能在输入状态时才有)

	0	1	AVREF
	1	0	AN2
	1	1	Not used
P0FSR1	P01	功能选择	
	0	I/O Port	
	1	AN1	
P0FSR0	P00	功能选择	
	0	I/O Port	
	1	AN0	

**P1FSRH (Port 1 功能选择高位寄存器) : D5H**

7	6	5	4	3	2	1	0
-	P1FSRH6	P1FSRH5	P1FSRH4	P1FSRH3	P1FSRH2	P1FSRH1	P1FSRH0
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1FSRH6	P17	功能选择	
	0	I/O Port (EC2/SS 功能在输入状态时才有)	
	1	AN14	
P1FSRH[5:4]	P16	功能选择	
	P1FSRH5	P1FSRH4	Description
	0	0	I/O Port
	0	1	SCK
	1	0	AN13
	1	1	Not used
P1FSRH[3:2]	P15	功能选择	
	P1FSRH3	P1FSRH2	Description
	0	0	I/O Port
	0	1	MOSI
	1	0	AN12
	1	1	Not used
P1FSRH[1:0]	P14	功能选择	
	P1FSRH1	P0FSRH0	Description
	0	0	I/O Port
	0	1	MISO
	1	0	AN11
	1	1	Not used



**P1FSRL (Port 1 功能选择低位寄存器) : D4H**

7	6	5	4	3	2	1	0
-	P1FSRL6	P1FSRL5	P1FSRL4	P1FSRL3	P1FSRL2	P1FSRL1	P1FSRL0
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1FSRL[6:5] P13 功能选择

P1FSRL6	P1FSRL5	Description
0	0	I/O Port (EINT12 功能在输入状态时才有)
0	1	T2O/PWM2O
1	0	AN10
1	1	Not used

P1FSRL[4:3] P12 功能选择

P1FSRL4	P1FSRL3	Description
0	0	I/O Port (EINT11 功能在输入状态时才有)
0	1	T1O/PWM1O
1	0	AN9
1	1	Not used

P1FSRL[2:1] P11 功能选择

P1FSRL2	P1FSRL1	Description
0	0	I/O Port (EINT6/EC1 功能在输入状态时才有)
0	1	BUZO
1	0	AN8
1	1	Not used

P1FSRL0 P10 功能选择

0	I/O Port (EINT5 功能在输入状态时才有)
1	AN7 Function

**P2FSR (Port 2 功能选择寄存器) : D6H**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	P2FSR1	P2FSR0
-	-	-	-	-	-	RW	RW

Initial value : 00H

P2FSR1 P25 功能选择

0	I/O Port
1	SCL

P2FSR0 P24 功能选择

0	I/O Port
1	SDA

P3FSR (Port 3 功能选择寄存器) : D7H							
7	6	5	4	3	2	1	0
P3FSR7	P3FSR6	P3FSR5	P3FSR4	P3FSR3	P3FSR2	P3FSR1	P3FSR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

- P3FSR7 P37 功能选择
  - 0 I/O Port
  - 1 XOUT
- P3FSR6 P36 功能选择
  - 0 I/O Port
  - 1 XIN
- P3FSR5 P35 功能选择
  - 0 I/O Port
  - 1 T00/PWM00
- P3FSR4 P34 功能选择
  - 0 I/O Port
  - 1 SXIN
- P3FSR3 P33 功能选择
  - 0 I/O Port
  - 1 SXOUT
- P3FSR2 P31 功能选择
  - 0 I/O Port (RXD 功能在输入状态时才有)
  - 1 SCL Function
- P3FSR[1:0] P30 功能选择
 

P3FSR1	P3FSR0	Description
0	0	I/O Port
0	1	TXD
1	0	SDA
1	1	Not used

注意) 参考 P32/RESETB 配置选项

## 6、中断控制器

## 6.1、概述

YF8316 16/32 有 20 个中断源。各中断分别具有各种独立的使能寄存器位，允许软件控制。分配为 4 个中断优先级，非可屏蔽中断源通常比其他使能中断源具有更高的优先级，且不可通过软件控制。中断控制器有以下特性：

- 20 个中断源接收中断
- 6 优先级组
- 4 个优先等级
- 多中断复用
- 如果同时接收到不同中断优先级的中断请求，首先响应较高中断优先级的请求
- 各中断位可通过 EA 位和 IEx 位控制
- 中断响应时间：单个中断 3~9 个机器周期

非可屏蔽中断是经常用到的，可屏蔽中断通过四对中断使能寄存器(IE, IE1, IE2, IE3)使能。IE, IE1, IE2, IE3 的每一位分别单独的使能/禁止相应的中断源。总中断控制位是 IE 中的 7 位(EA)。EA 置 '0'，所有中断禁止；EA 置 '1'，中断通过中断使能寄存器中的其他位分别对其使能和禁止。EA 位通常被清零后立即跳到中断服务程序，置 1 则执行[RETI]指令。YF8316 16/32 支持 4 个优先级。每个可屏蔽中断通过 IP 和 IP1 来分配中断优先等级。

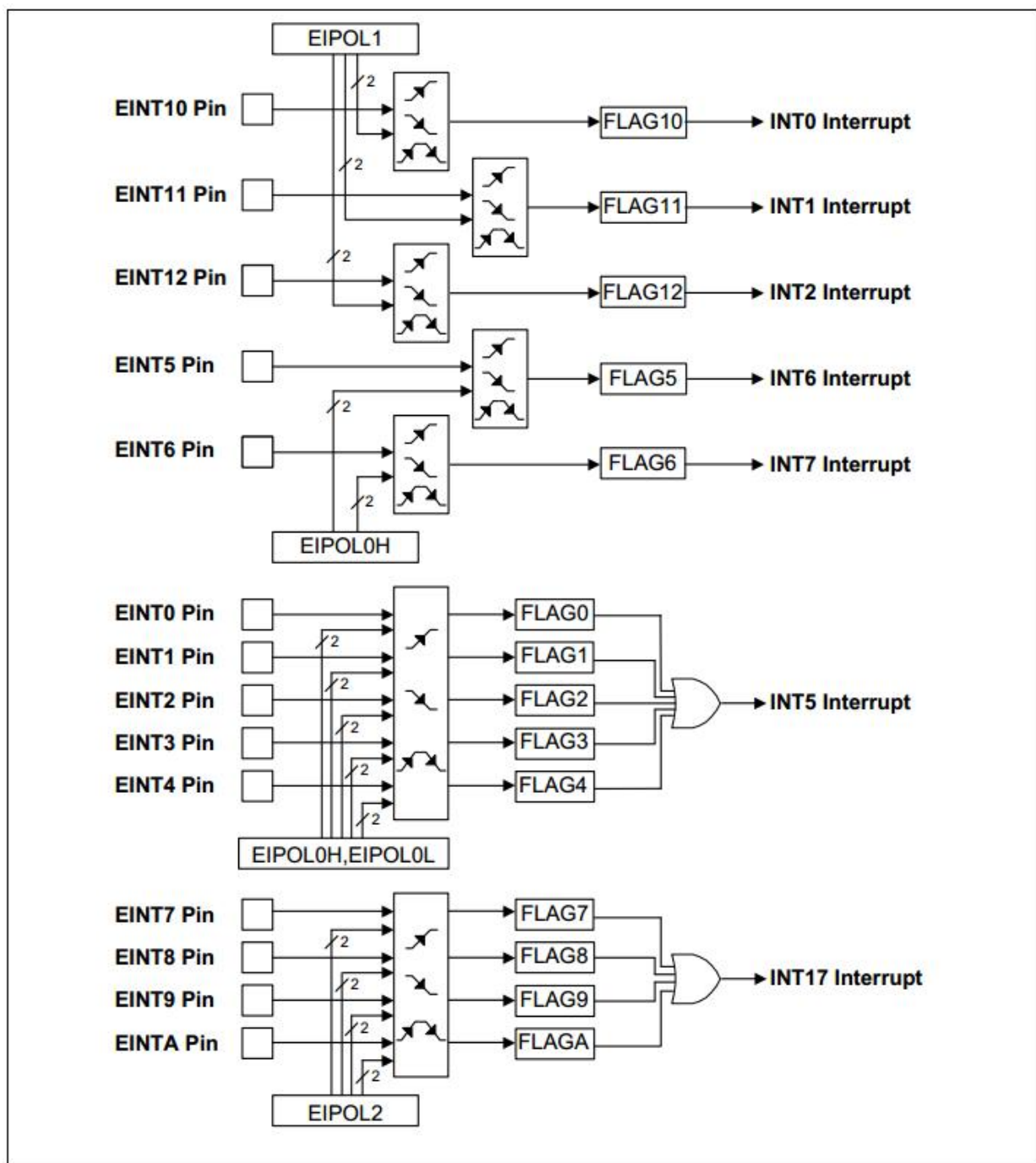
中断默认的是电平模式触发，如果有需要的话可以改变电平触发模式。下表所示为有共同中断优先级的中断组。优先级通过中断优先寄存器的两位来设置(一位是 IP，另一位是 IP1)。较高优先级的中断服务程序首先执行，如果同时接收到两个不同优先级的中断请求，那么优先级高的比优先级低的先被响应。

中断组优先级表

Interrupt Group	Highest ←-----→ Lowest				Highest ↓ Lowest
	Interrupt 0	Interrupt 6	Interrupt 12	Interrupt 18	
0 (Bit0)	Interrupt 0	Interrupt 6	Interrupt 12	Interrupt 18	
1 (Bit1)	Interrupt 1	Interrupt 7	Interrupt 13	Interrupt 19	
2 (Bit2)	Interrupt 2	Interrupt 8	Interrupt 14	Interrupt 20	
3 (Bit3)	Interrupt 3	Interrupt 9	Interrupt 15	Interrupt 21	
4 (Bit4)	Interrupt 4	Interrupt 10	Interrupt 16	Interrupt 22	
5 (Bit5)	Interrupt 5	Interrupt 11	Interrupt 17	Interrupt 23	

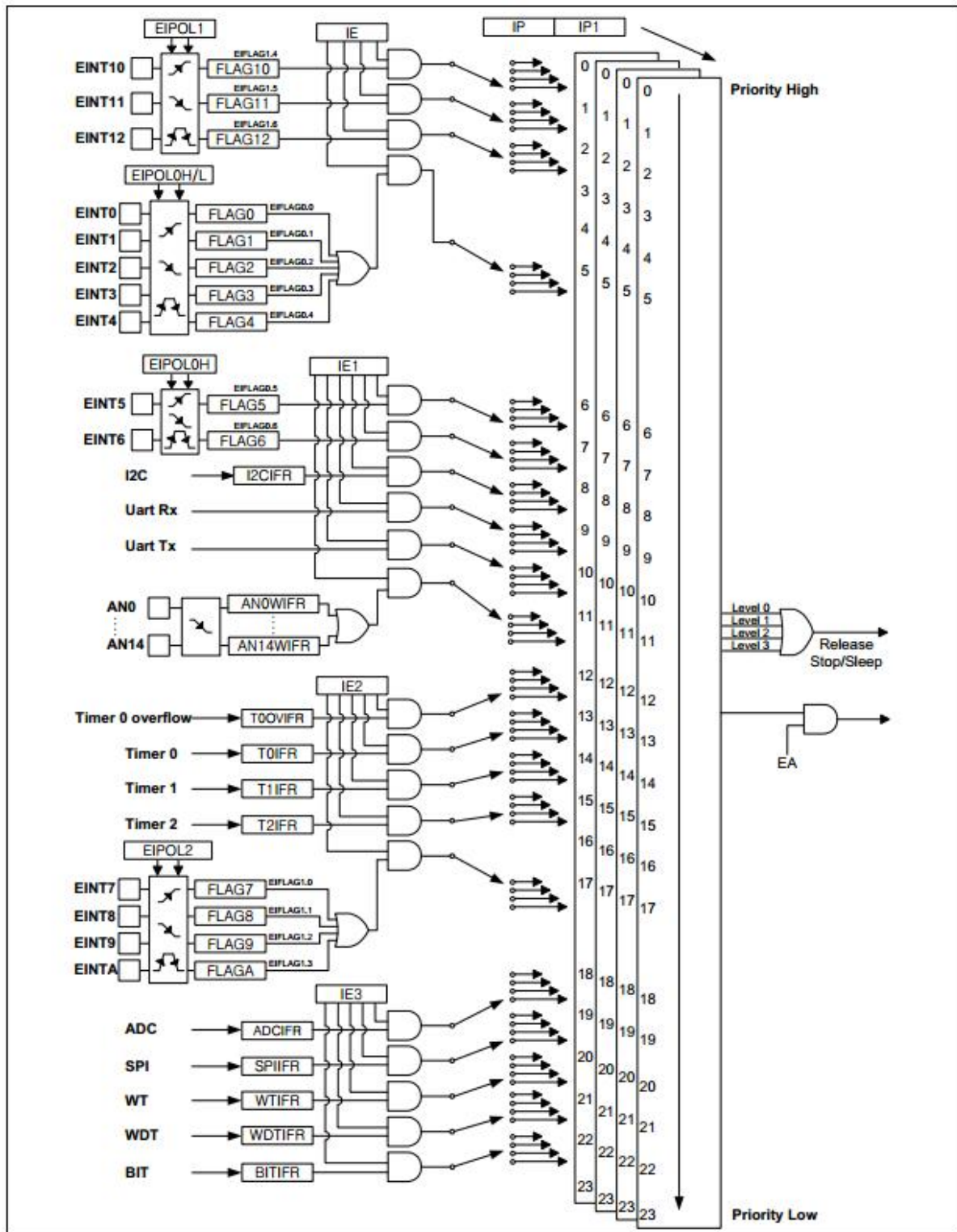
## 6.2、外部中断

在 INT0, INT1, INT2, INT3, INT4, INT5, INT6, INT7, INT8, INT9, INTA, INT10, INT11 和 INT12 引脚上的外部中断通过外部中断边沿触发器 (EIPOL0H/L) 来接收不同的中断请求，如下图所示为外部中断边沿触发器 1(EIPOL1)和外部中断边沿触发器 2(EIPOL2)。另外外部中断源同样可以通过位操作开关。外部中断标志寄存器 0(EIFLAG0)外部中断标志寄存器 1(EIFLAG1)提供外部中断的状态。



外部中断说明

### 6.3、方框图



中断方框图

- 注意)
- 1.stop/idle 模式的复位信号可以由所有使能的中断产生，和中断优先级无关。
  - 2.当数据写入到 IE, IE1, IE2, IE3, IP, IP1, 和 PCON 寄存器时中断请求被延迟。

#### 6.4、中断向量表

中断控制器支持 24 个中断源。当中断被响应,执行长调用指令(LCALL)同时程序计数器跳到向量地址。所有的中断请求都有自己的优先等级。

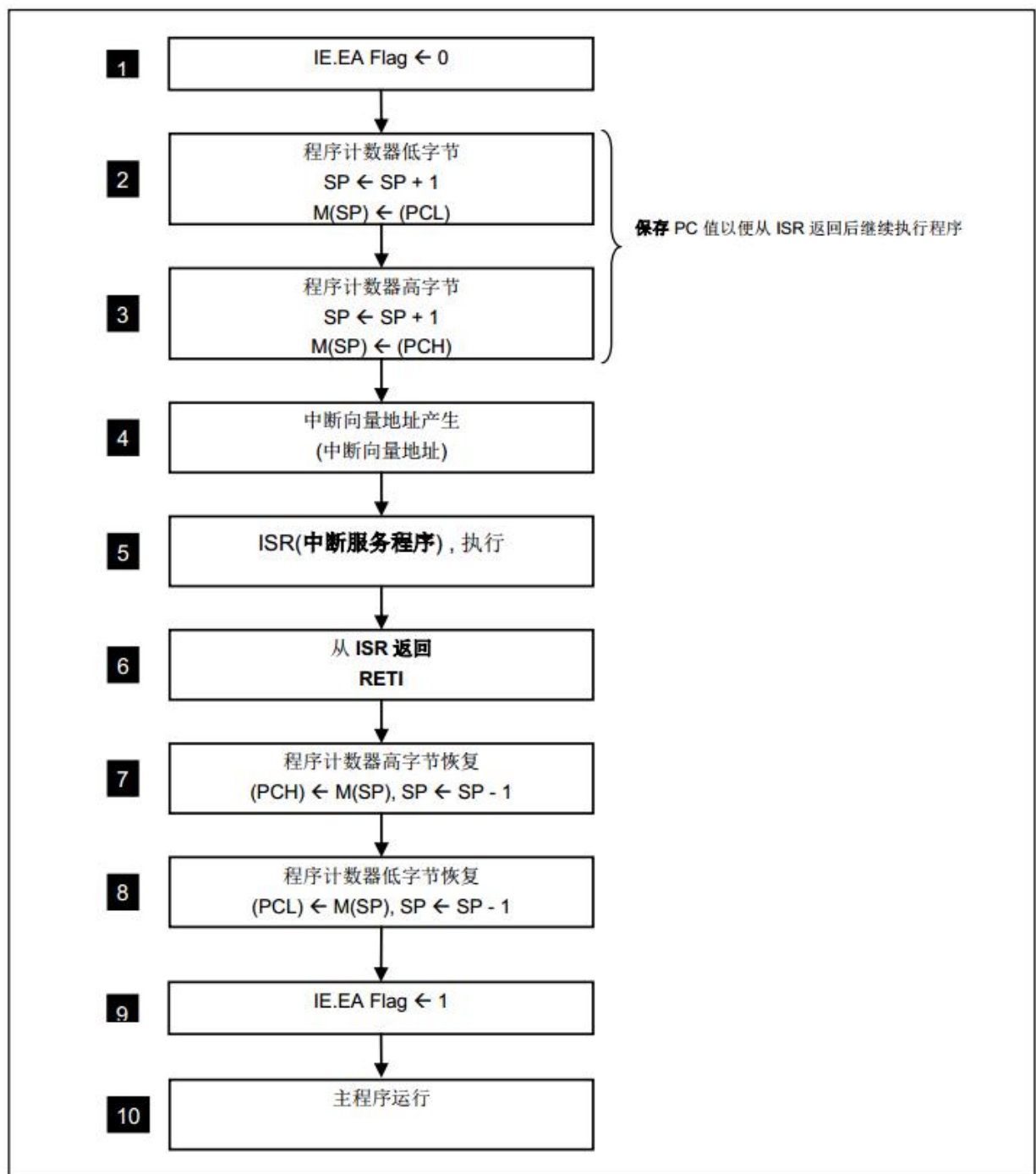
中断向量地址表

Interrupt Source	Symbol	Interrupt Enable Bit	Polarity	Mask	Vector Address
Hardware Reset	RESETB	0 0	0	Non-Maskable	0000H
External Interrupt 10	INT0	IE.0	1	Maskable	0003H
External Interrupt 11	INT1	IE.1	2	Maskable	000BH
External Interrupt 12	INT2	IE.2	3	Maskable	0013H
-	INT3	IE.3	4	Maskable	001BH
-	INT4	IE.4	5	Maskable	0023H
External Interrupt 0 – 4	INT5	IE.5	6	Maskable	002BH
External Interrupt 5	INT6	IE1.0	7	Maskable	0033H
External Interrupt 6	INT7	IE1.1	8	Maskable	003BH
I2C Interrupt	INT8	IE1.2	9	Maskable	0043H
UART Rx Interrupt	INT9	IE1.3	10	Maskable	004BH
UART Tx Interrupt	INT10	IE1.4	11	Maskable	0053H
ADC Wake-up Interrupt	INT11	IE1.5	12	Maskable	005BH
T0 Overflow Interrupt	INT12	IE2.0	13	Maskable	0063H
T0 Match Interrupt	INT13	IE2.1	14	Maskable	006BH
T1 Match Interrupt	INT14	IE2.2	15	Maskable	0073H
T2 Match Interrupt	INT15	IE2.3	16	Maskable	007BH
-	INT16	IE2.4	17	Maskable	0083H
External Interrupt 7 – A	INT17	IE2.5	18	Maskable	008BH
ADC Interrupt	INT18	IE3.0	19	Maskable	0093H
SPI Interrupt	INT19	IE3.1	20	Maskable	009BH
WT Interrupt	INT20	IE3.2	21	Maskable	00A3H
WDT Interrupt	INT21	IE3.3	22	Maskable	00ABH
BIT Interrupt	INT22	IE3.4	23	Maskable	00B3H
-	INT23	IE3.5	24	Maskable	00BBH

可屏蔽中断的执行, EA 位必须置 ‘1’ 且 IEx 相应的位必须置 1 来使能, 如果接收到中断请求, 相应的中断标志位被置 ‘1’。‘1’ 将会一直被保存到 CPU 接受中断。如果真中断被响应, 中断请求标志位将会被自动清零。

## 6.5、中断序列

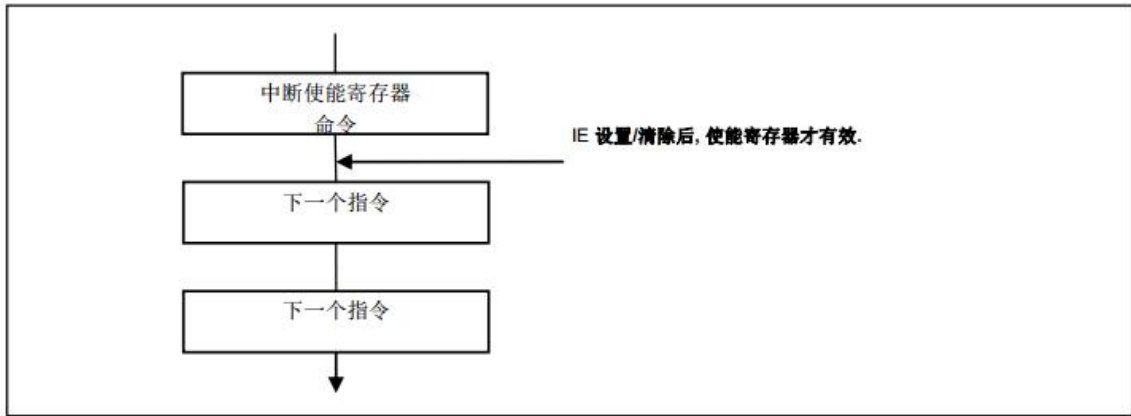
中断请求会一直保存, 直到接受中断或者中断锁存器被复位或指令清零。接受中断通常在最后的指令周期产生而不是当前的指令, CPU 执行内部 LCALL 指令并且保存 PC 堆栈。关于中断服务程序, 中断控制器将 LJMP 指令送到 CPU。完成当前指令之后, 需要 3~9 个机器周期进入中断服务程序。中断服务被中断返回指令[RETI]终止。一旦中断请求被响应, 随后的流程如图所示:



中断向量地址表

### 6.6、控制中断位之后的作用时序

Case a) 中断控制使能寄存器(IE, IE1, IE2, IE3):



中断使能寄存器时序

Case b) 中断标志位寄存器



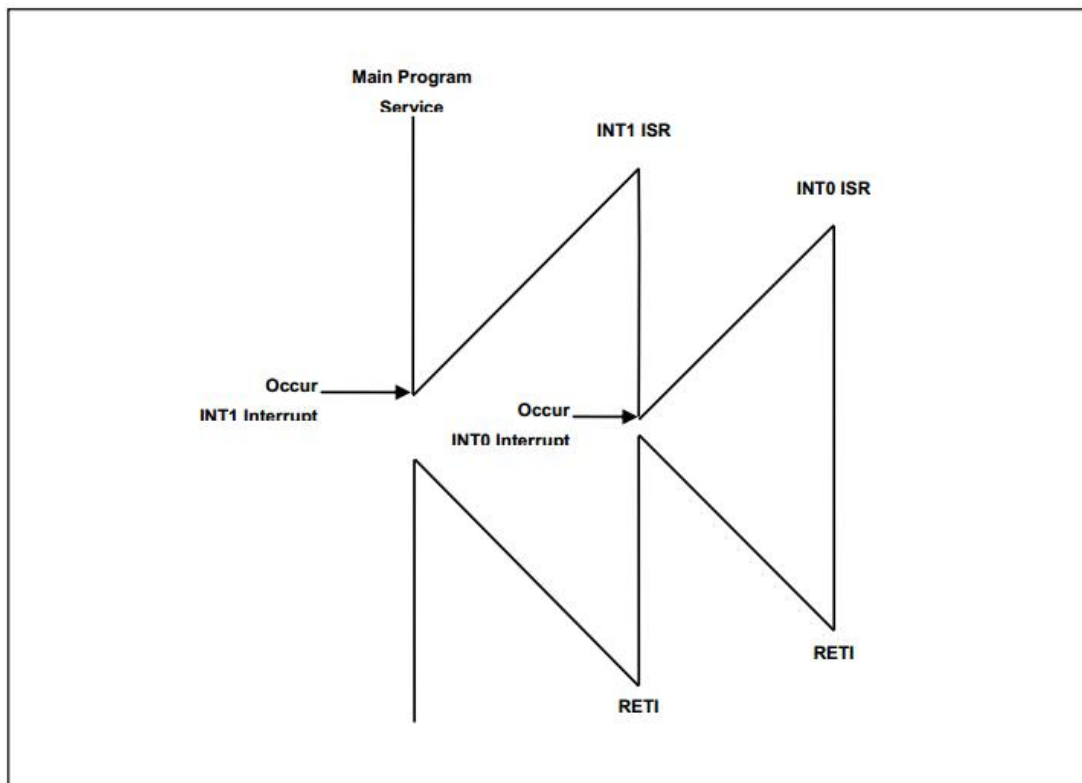
中断标志位寄存器时序图

### 6.7、多中断复用

如果同时收到不同优先级的中断请求，首先响应优先级较高的请求。如果同时接收到同一优先级



的多个中断，则由硬件查询序列决定响应哪一中断。不过也可通过软件实现多中断请求。

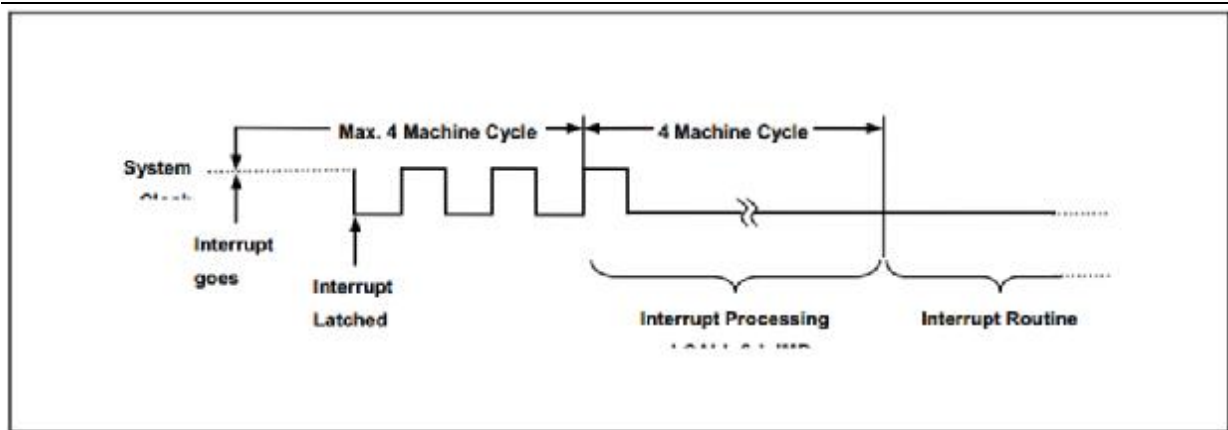


中断复用时序

上图所示为多中断复用的例子，执行 INT1 时，有更高优先级的 INT0 发生，会立即执行 INT0 并且保存 INT1 的服务子程序。如果 INT0 和 INT1 优先级相同或低于 INT1，那么 INT0 在 INT1 执行完成之后才能执行。

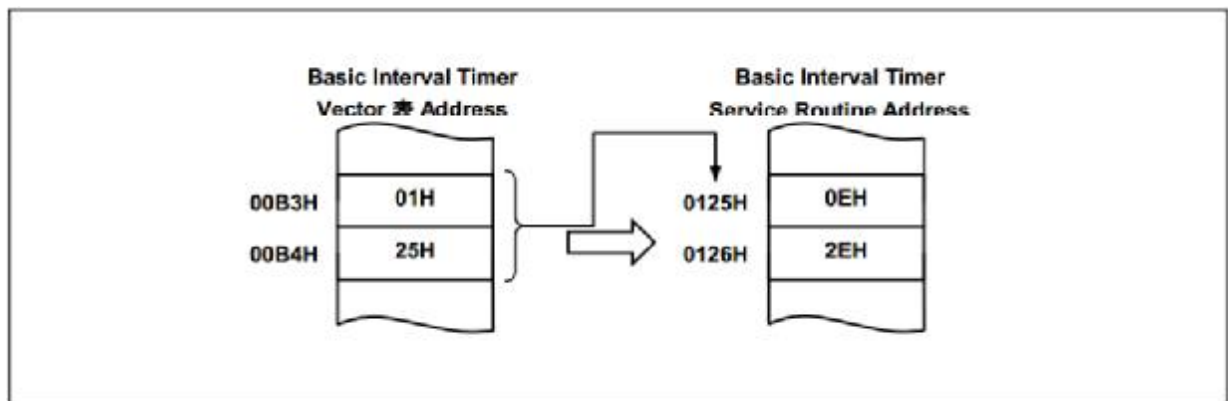
中断服务程序只能被比它更高优先级的中断，如果两个不同优先级的中断同时发生，首先执行优先级较高的中断。中断不能被低于或相同优先级的中断进行中断。如果两个优先级相同的中断同时发生，中断次序由扫描顺序决定。

## 6.8、中断使能响应时序



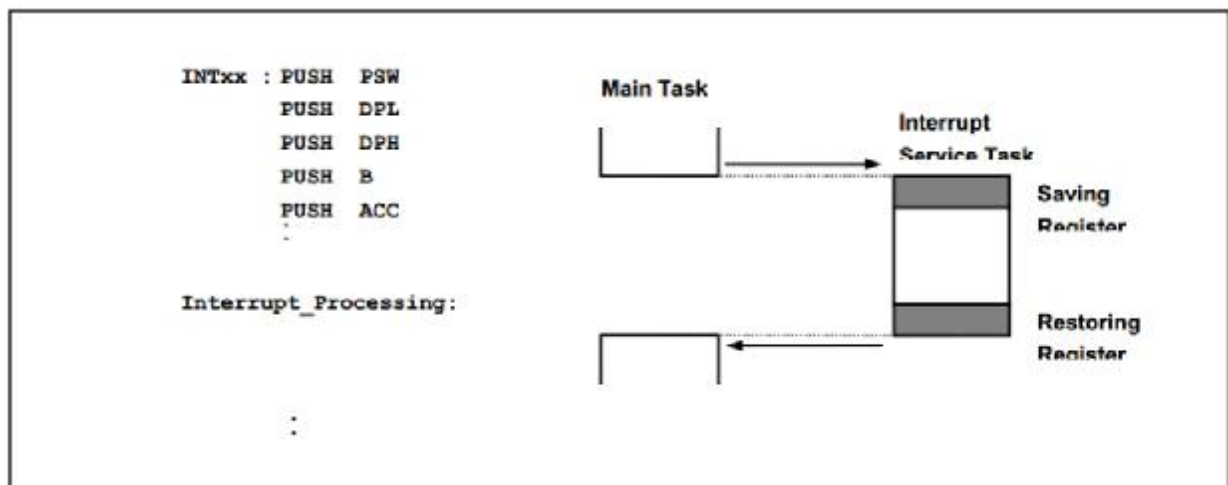
中断响应时序方框图

### 6.9、中断服务程序地址



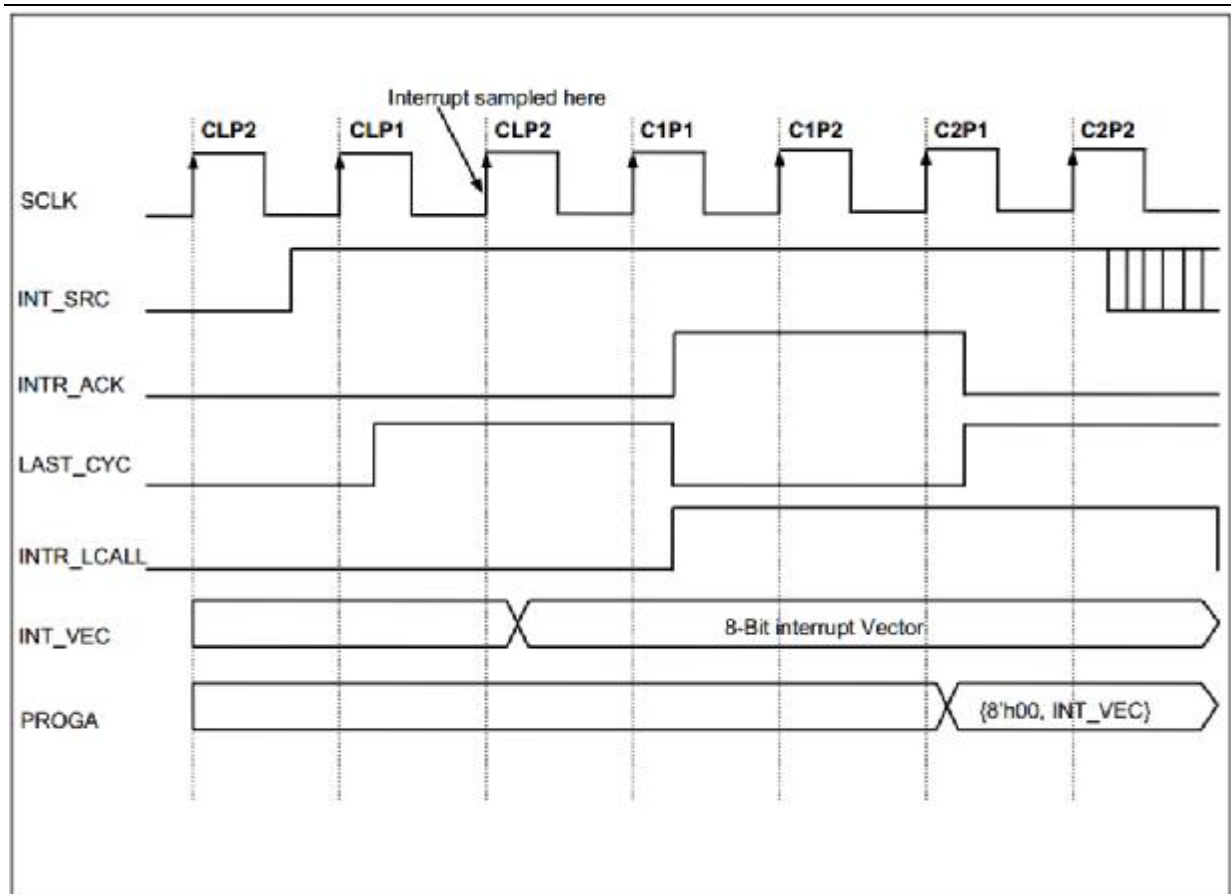
向量表地址和对应的ISP入口地址

### 6.10、通用寄存器的设置和复位



设置步骤举例

### 6.11、中断时序



中断接受和中断返回指令时序图

中断源在命令的最后一个周期取样。如果检测到中断向量(INT\_VEC)的低 8 位被确认。M8051W 内核在命令的第一个周期产生应答，执行长调用指令到中断服务程序。

注意) command cycle CLPx: L=Last cycle, 1=1<sup>st</sup> cycle or 1<sup>st</sup> phase, 2=2<sup>nd</sup> cycle or 2<sup>nd</sup> phase

## 6.12、中断寄存器概述

### 6.12.1、中断使能寄存器 (IE, IE1, IE2, IE3)

中断使能寄存器由总中断位(EA)和外围各中断位组成，共有 24 个可控制中断。

### 6.12.2、中断优先级寄存器 (IP, IP1)

24 个中断被分为 6 组每组有 4 个中断源。每组中断可以通过中断优先级寄存器配置为 4 个中断优先级。Level3 优先级最高，level0 优先级最低。复位后 IP 和 IP1 被清除为 '00H'。同一组中，低序号中断有优先权。

### 6.12.3、外部中断标志位寄存器 (EIFLAG0, EIFLAG1)

当外部中断条件满足时外部中断标志位 0 寄存器(EIFLAG0)外部中断标志位 1 寄存器(EIFLAG1)置 '1'。中断服务程序被执行标志位清零，或者写入 0 进行手动清除。

### 6.12.4、外部中断边沿寄存器 (EIPOL0H, EIPOL0L, EIPOL1, EIPOL2)

外部中断高低边沿触发寄存器 0(EIPOL0H/L), 外部中断高低边沿触发寄存器 1(EIPOL1)外部中断高低边沿触发寄存器 2(EIPOL2)决定了上边沿, 下边沿, 或双边沿触发中断。默认值是什么边沿没有中断。

### 6.12.5、寄存器图

中断寄存器图

Name	Address	Dir	Default	Description
IE	A8H	R/W	00H	Interrupt Enable Register
IE1	A9H	R/W	00H	Interrupt Enable Register 1
IE2	AAH	R/W	00H	Interrupt Enable Register 2
IE3	ABH	R/W	00H	Interrupt Enable Register 3
IP	B8H	R/W	00H	Interrupt Priority Register
IP1	F8H	R/W	00H	Interrupt Priority Register 1
IIFLAG	A0H	R/W	00H	Internal Interrupt Flag Register
EIFLAG0	C0H	R/W	00H	External Interrupt Flag 0 Register
EIPOL0L	A4H	R/W	00H	External Interrupt Polarity 0 Low Register
EIPOL0H	A5H	R/W	00H	External Interrupt Polarity 0 High Register
EIFLAG1	B0H	R/W	00H	External Interrupt Flag 1 Register
EIPOL1	A6H	R/W	00H	External Interrupt Polarity 1 Register
EIPOL2	A7H	R/W	00H	External Interrupt Polarity 2 Register

### 6.13、中断寄存器说明

中断寄存器用于控制中断功能。也有外部中断控制寄存器。中断寄存器由中断使能寄存器(IE, IE1, IE2, IE3)组成。外部中断由外部中断标志位 0(EIFLAG0), 外部中断高低边沿触发寄存器 0(EIPOL0H/L), 外部中断标志位 1(EIFLAG1), 外部中断边沿寄存器 1(EIPOL1)和外部中断边沿寄存器 2(EIPOL2)组成。

#### 6.13.1、中断寄存器说明

**IE (中断使能寄存器) : A8H**

7	6	5	4	3	2	1	0
EA	-	INT5E	-	-	INT2E	INT1E	INT0E
RW	-	RW	-	-	RW	RW	RW

Initial value : 00H

- EA            打开或关闭所有中断
  - 0            禁止
  - 1            打开
- INT5E        外部中断 0 ~ 4 (EINT0 ~ EINT4)
  - 0            禁止
  - 1            打开
- INT2E        外部中断 12(EINT12)
  - 0            禁止
  - 1            打开

- INT1E 外部中断 11(EINT11)
  - 0 禁止
  - 1 打开
- INT0E 外部中断 10 (EINT10)
  - 0 禁止
  - 1 打开

**IE1 (中断使能寄存器 1): A9H**

7	6	5	4	3	2	1	0
-	-	INT11E	INT10E	INT9E	INT8E	INT7E	INT6E
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

- INT11E ADC 唤醒中断
  - 0 禁止
  - 1 打开
- INT10E UART Tx 发送中断
  - 0 禁止
  - 1 打开
- INT9E UART Rx 接收中断
  - 0 禁止
  - 1 打开
- INT8E I2C 中断
  - 0 禁止
  - 1 打开
- INT7E 外部中断 6 (EINT6)
  - 0 禁止
  - 1 打开
- INT6E 外部中断 5 (EINT5)
  - 0 禁止
  - 1 打开

**IE2 (中断使能寄存器 2): AAH**

7	6	5	4	3	2	1	0
-	-	INT17E	-	INT15E	INT14E	INT13E	INT12E
-	-	RW	-	RW	RW	RW	RW

Initial value : 00H

- INT17E 外部中断 7 ~ A (EINT7 ~ EINTA)
  - 0 禁止
  - 1 打开
- INT15E 定时器 2 匹配中断
  - 0 禁止

1 打开  
 INT14E 定时器 1 匹配中断  
 0 禁止  
 1 打开  
 INT13E 定时器 0 匹配中断  
 0 禁止  
 1 打开  
 INT12E 定时器 0 溢出中断  
 0 禁止  
 1 打开

**IE3 (中断使能寄存器 3) : ABH**

7	6	5	4	3	2	1	0
-	-	-	INT22E	INT21E	INT20E	INT19E	INT18E
-	-	-	RW	RW	RW	RW	RW

Initial value : 00H

INT22E BIT 中断  
 0 禁止  
 1 打开  
 INT21E WDT 中断  
 0 禁止  
 1 打开  
 INT20E WT 中断  
 0 禁止  
 1 打开  
 INT19E SPI 中断  
 0 禁止  
 1 打开  
 INT18E ADC 中断  
 0 禁止  
 1 打开

**IP (中断优先级寄存器) : B8H**

7	6	5	4	3	2	1	0
-	-	IP5	IP4	IP3	IP2	IP1	IP0
-	-	RW	RW	RW	RW	RW	RW

Initial value : 00H

**IP1 (中断优先级寄存器 1) : F8H**

7	6	5	4	3	2	1	0
-	-	IP15	IP14	IP13	IP12	IP11	IP10
-	-	RW	RW	RW	RW	RW	RW

Initial value : 00H

IP[5:0], IP1[5:0] 选择中断优先级组

IP1x	IPx	说明
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

**IIFLAG (内部中断标志位寄存器) : A0H**

7	6	5	4	3	2	1	0
-	-	-	-	-	IICIFR	TOOVIFR	TOIFR
-	-	-	-	-	R	RW	RW

Initial value : 00H

**IICIFR** 中断标志位 中断发生时置 ‘1’ 写入 I2CSR 任何值该位清零

- 0 没有 I2C 中断产生
- 1 有 I2C 中断产生

**TOOVIFR** 当 T0 溢出中断产生时, 该位置 ‘1’。通过手动清零或被 INT\_ACK 信号自动清零

- 0 没有 T0 溢出中断产生
- 1 有 T0 溢出中断产生

**TOIFR** 当 T0 中断产生时置 ‘1’。手动写入 0 清除或通过 INT\_ACK 信号自动清零

- 0 没有 T0 中断产生
- 1 有 T0 中断产生

**EIFLAG0 (外部中断标志位 0 寄存器) : C0H**

7	6	5	4	3	2	1	0
-	FLAG6	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

**EIFLAG0[6:5]** 当外部中断 5~6 产生时置 ‘1’。写入 ‘0’ 清零或者通过 INT\_ACK 信号自动清零

- 0 没有外部中断 5~6 产生
- 1 有外部中断 5~6 产生

**EIFLAG0[4:0]** 当外部中断 0~4 产生时置 ‘1’。该标志位必须手动写入 0 清除。所以, 需要软件清零。

- 0 没有外部中断 0~4 发生
- 1 有外部中断 0~4 发生

**注意:** 直接位判断并跳转的指令(见下面例子)不能用于 EIFLAG0。

例子: if(FLAG0) → if(EIFLAG0 & 0x01), 左边的指令应该改成右边的指令形式。

**EIPOL0H (外部中断边沿 0 高位寄存器): A5H**

7	6	5	4	3	2	1	0
-	-	POL6		POL5		POL4	
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

EIPOL0H[5:0] 外部中断(EINT6, EINT5, EINT4)边沿触发选择

POLn[1:0]		说明
0	0	任何边沿触发都不产生中断
0	1	上升沿触发
1	0	下降沿触发
1	1	双边沿触发

Where n =4, 5 and 6

**EIPOL0L (外部中断边沿 0 低位寄存器): A4H**

7	6	5	4	3	2	1	0
POL3		POL2		POL1		POL0	
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

EIPOL0L[7:0] 外部中断(EINT3, EINT2, EINT1, EINT0)边沿触发选择

POLn[1:0]		说明
0	0	任何边沿触发都不产生中断
0	1	上升沿触发
1	0	下降沿触发
1	1	双边沿触发

Where n =0, 1, 2 and 3

**EIFLAG1 (外部中断标志位 1 寄存器) : B0H**

7	6	5	4	3	2	1	0
-	FLAG12	FLAG11	FLAG10	FLAGA	FLAG9	FLAG8	FLAG7
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

EIFLAG1[6:4] 当外部中断 10 ~ 12 产生时置 ‘1’。写入 0 清除或通过 INT\_ACK 信号自动进行清除  
 0 没有外部中断 10 ~ 12 产生  
 1 有外部中断 10 ~ 12 产生

EIFLAG1[3:0] 当外部中断 7 ~ A 产生时置 ‘1’。该位必须手动写入 0 才能清除，所以该位应该通过软件清除。

0	没有外部中断 7 ~ A 产生
1	有外部中断 7 ~ A 产生

注意：直接位判断并跳转的指令(见下面例子)不能用于 EIFLAG1。

例子: if(FLAG0) → if(EIFLAG1 & 0x01)，左边的指令应该改成右边的指令形式。



**EIPOL1 (外部中断边沿 1 寄存器): A6H**

7	6	5	4	3	2	1	0
-	-	POL12		POL11		POL10	
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

EIPOL1[5:0] 外部中断(EINT12, EINT11, EINT10)边沿触发选择

POLn[1:0] 说明

0 0 任何边沿触发都不产生中断

0 1 上升沿触发

1 0 下降沿触发

1 1 双边沿触发

Where n = 10, 11 and 12

**EIPOL2 (外部中断边沿 2 寄存器): A7H**

7	6	5	4	3	2	1	0
POLA		POL9		POL8		POL7	
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

EIPOL2[7:0] 外部中断(EINTA, EINT9, EINT8, EINT7)边沿触发选择

POLn[1:0] 说明

0 0 任何边沿触发都不产生中断

0 1 上升沿触发

1 0 下降沿触发

1 1 双边沿触发

Where n = 7, 8, 9 and A

**7、外围硬件**

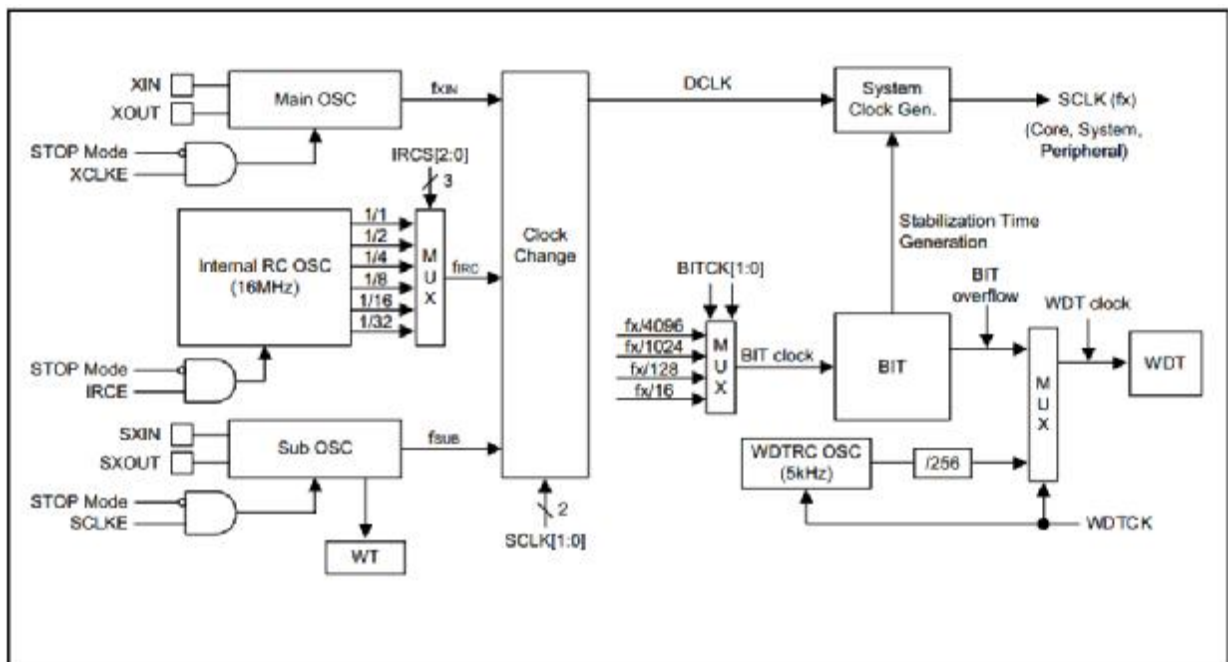
## 7.1、时钟发生器

### 7.1.1、概述

如下图所示，时钟发生器为 CPU 和外围设备提供基本时钟脉冲，它包括主/副时钟振荡器。主/副时钟可以通过在 XIN/SXIN 和 XOUT/SXOUT 引脚上分别连接晶体获得。也可以通过外部时钟获得，如果这样的话，需要输入时钟信号到 XIN/SXIN 并且打开 XOUT/SXOUT 脚。默认的系统时钟是 1MHzINT-RC 默 16 分频。为了内部时钟的稳定，POR 使用 1MHz INT-RC。

- 校验内部 RC 振荡 Oscillator (16 MHz)
  - . INT-RC OSC/1 (16 MHz)
  - . INT-RC OSC/2 (8 MHz)
  - . INT-RC OSC/4 (4 MHz)
  - . INT-RC OSC/8 (2 MHz)
  - . INT-RC OSC/16 (1 MHz, 默认系统时钟)
  - . INT-RC OSC/32 (0.5 MHz)
- 晶体主振荡器 (0.4~12 MHz)
- 晶体副振荡器(32.768 kHz)
- 内部 WDTRC 振荡器 (5 kHz)

### 7.1.2、方框图



时钟发生器方框图

### 7.1.3、寄存器图

时钟发生器寄存器图

寄存器名	地址	方式	默认	说明
SCCR	8AH	R/W	00H	系统及时钟控制寄存器
OSCCR	C8H	R/W	08H	振荡器控制寄存器

### 7.1.4、时钟发生器寄存器说明

时钟发生寄存器使用时钟控制系统操作。时钟发生器包含系统时钟控制寄存器和振荡控制寄存器组成。

### 7.1.5、时钟发生器寄存器说明

**SCCR (系统时钟控制寄存器) : 8AH**

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SCLK1	SCLK0
-	-	-	-	-	-	RW	RW

Initial value : 00H

SCLK [1:0]      系统时钟选择位

SCLK1	SCLK0	说明
0	0	内部 RC OSC (fIRC)
0	1	外部主时钟 (fXIN)
1	0	外部副时钟 (fSUB)
1	1	Not used

**OSCCR (振荡器控制寄存器) : C8H**

7	6	5	4	3	2	1	0
-	-	IRCS2	IRCS1	IRCS0	IRCE	XCLKE	SCLKE
-	-	RW	RW	RW	RW	RW	RW

Initial value : 08H

IRCS[2:0]      内部 RC 分频后选择

IRCS2	IRCS1	IRCS0	Description
0	0	0	INT-RC/32 (0.5MHz)
0	0	1	INT-RC/16 (1MHz)
0	1	0	INT-RC/8 (2MHz)
0	1	1	INT-RC/4 (4MHz)
1	0	0	INT-RC/2 (8MHz)
1	0	1	INT-RC/1 (16MHz)

Other values      Not used

IRCE            内部 RC 控制操作

0	使能内部 RC
1	关闭内部 RC

XCLKE          外部主振荡器控制操作

	0	关闭外部主时钟 X-TAL
	1	使能外部主时钟 X-TAL
SCLKE	外部副振荡器控制操作	
	0	关闭外部副时钟 SX-TAL
	1	打开外部副时钟 SX-TAL

## 7.2、BIT

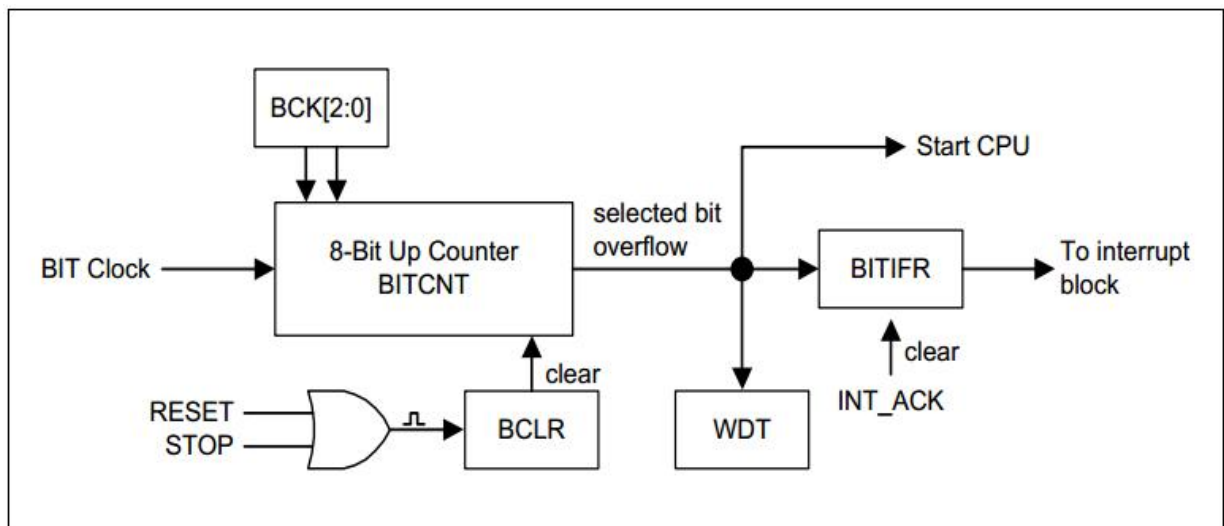
### 7.2.1、概述

YF8316 16/32 有一个独立运行不能停止的 8 位基本间隔定时器。如图所示。另外它还为看门狗计数器提供扫描基线，提供一个基本间隔定时器中断(BITIFR)。

YF8316 16/32 基本间隔定时器的特性如下：

- 上电过程中，BIT 提供稳定的时钟脉冲时间
- 退出 Stop 模式，BIT 提供稳定的时钟脉冲时间
- 作为定时器使用，产生定时器中断

### 7.2.2、方框图



BIT 方框图

### 7.2.3、寄存器图

BIT 寄存器图表

名字	地址	读写方式	默认	说明
BITCNT	8CH	R	00H	BIT计数器寄存器
BITCR	8BH	R/W	01H	BIT控制寄存器

### 7.2.4、BIT 寄存器说明

基本间隔寄存器由 BITCNT 和 BITCR 组成。如果 BCLR 位设置为 ‘1’，BITCNT 变为 ‘0’ 并开始计数。一个机器周期后，BCLR 位自动被清零。

## 7.2.5、BIT 寄存器说明

## BITCNT (BIT 计数器寄存器) : 8CH

7	6	5	4	3	2	1	0
BITCNT7	BITCNT6	BITCNT5	BITCNT4	BITCNT3	BITCNT2	BITCNT1	BITCNT0
R	R	R	R	R	R	R	R

Initial value : 00H

BITCNT[7:0] BIT 计数器

## BITCR (BIT 控制寄存器) : 8BH

7	6	5	4	3	2	1	0
BITIFR	BITCK1	BITCK0	-	BCLR	BCK2	BCK1	BCK0
RW	RW	RW	-	RW	RW	RW	RW

Initial value : 01H

BITIFR 当 BIT 中断产生时该位置 '1'。写入 0 进行清零或者由 INT\_ACK 信号自动清零。

0 没有 BIT 中断产生

1 有 BIT 中断产生

BITCK[1:0] 选择 BIT 时钟

BITCK1	BITCK0	说明
0	0	fx/4096
0	1	fx/1024
1	0	fx/128
1	1	fx/16

BCLR 如果该位写入 '1'，BIT 计数器清零

0 运行

1 清零

BCK[2:0] 选择 BIT 溢出周期

BCK2	BCK1	BCK0	说明
0	0	0	Bit 0 overflow (BIT Clock * 2)
0	0	1	Bit 1 overflow (BIT Clock * 4) (默认)
0	1	0	Bit 2 overflow (BIT Clock * 8)
0	1	1	Bit 3 overflow (BIT Clock * 16)
1	0	0	Bit 4 overflow (BIT Clock * 32)
1	0	1	Bit 5 overflow (BIT Clock * 64)
1	1	0	Bit 6 overflow (BIT Clock * 128)
1	1	1	Bit 7 overflow (BIT Clock * 256)

## 7.3、WDT

## 7.3.1、概述

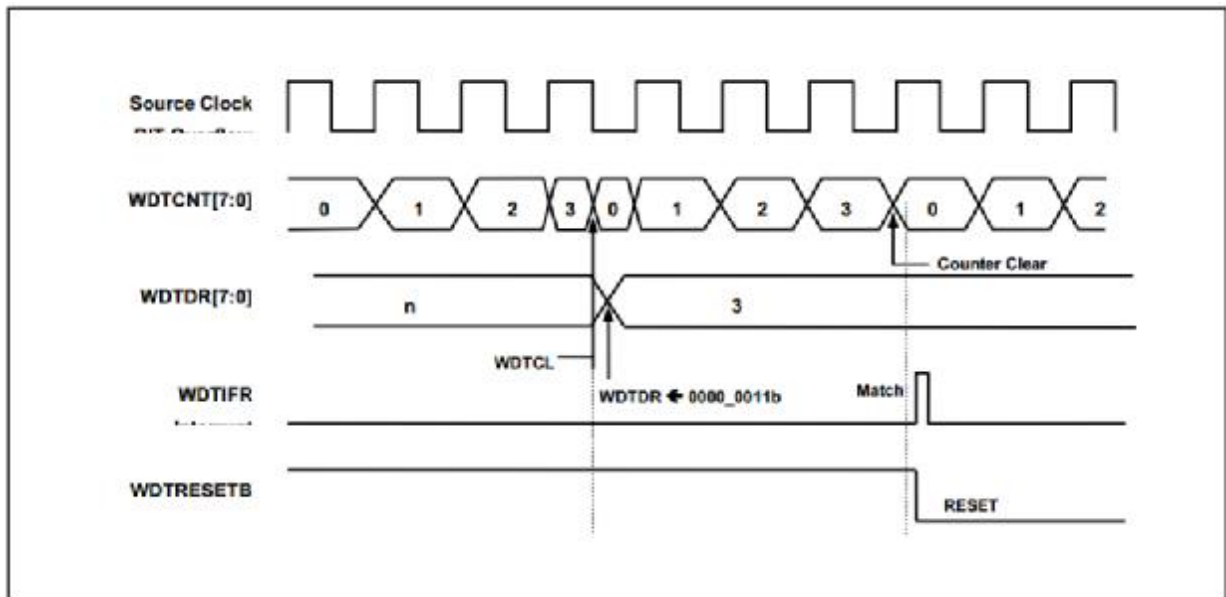
看门狗可以迅速的发 CPU 因干扰或类似的问题引起的死循环，并将其恢复到正常状态。该故障检测信号也可以作为 CPU 复位或者中断复位。不作为故障检测使用时，可用做固定时间产生中断的

定时器。也可以通过设置 WDTCSR[6]位作为独立运行的 8 位定时器模式(WDTRSON= ‘0’ )或者看门狗定时器模式(WDTRSON= ‘1’ )。如果 WDTCSR[5]被写入 ‘1’，WDT 计数器的值被清除并开始计数。一个机器周期后，该位被自动清零。它由 8 位二进制计数器和看门狗数据寄存器组成。当 8 位二进制计数器的值等于 8 位 WDTCNT 值时，产生中断请求标志位。可以根据 WDTRSON 位的设置用于看门狗中断或者复位。

它的时钟输入源来自于 BIT 溢出。内部看门狗中断由 BIT 溢出周期和 WDTDR 设定的值来决定。计算公式如下：

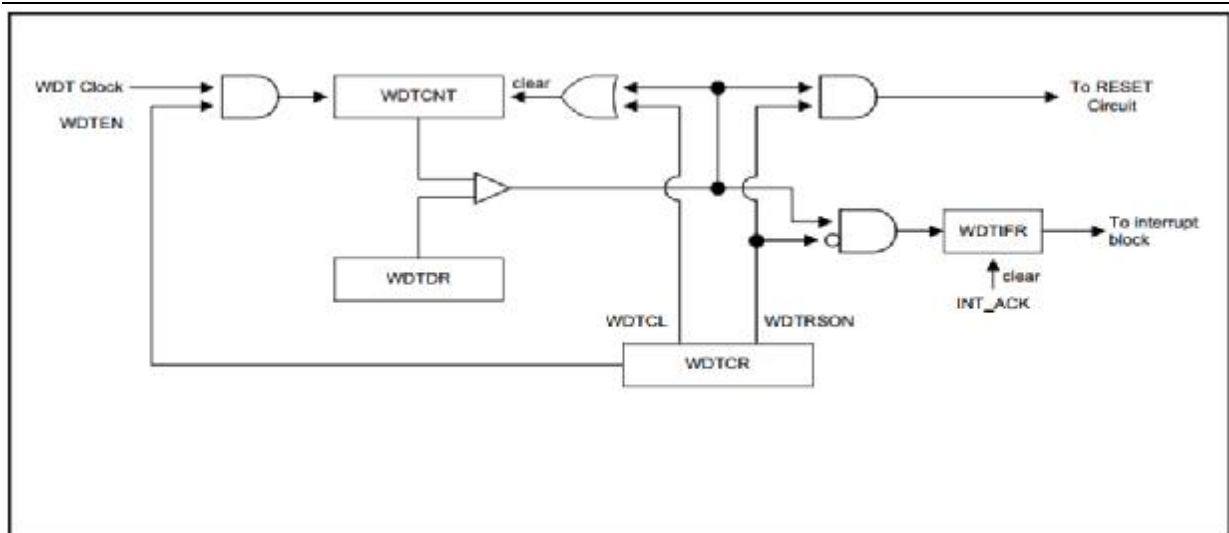
$$\text{WDT 中断时间} = (\text{BIT 中断时间}) * (\text{WDTDR 值} + 1)$$

### 7.3.2、WDT 中断时序波形



WDT 中断时序波形图

### 7.3.3、方框图



WDT 方框图

### 7.3.4、寄存器图

WDT 寄存器

名字	地址	方式	默认	说明
WDTCNT	8EH	R	00H	WDT计数器寄存器
WDTDR	8EH	W	FFH	WDT数据寄存器r
WDTCR	8DH	R/W	00H	WDT控制寄存器

### 7.3.5、寄存器说明

寄存器由 WDTCNT, WDTDR 和 WDTCR 组成。

### 7.3.6、寄存器说明

**WDTCNT (WDT 计数器寄存器: 只读寄存器) : 8EH**

7	6	5	4	3	2	1	0
WDTCNT7	WDTCNT6	WDTCNT5	WDTCNT4	WDTCNT3	WDTCNT2	WDTCNT1	WDTCNT0
R	R	R	R	R	R	R	R

Initial value : 00H

WDTCNT[7:0] WDT Counter

**WDTDR (WDT 数据寄存器: 只写寄存器) : 8EH**

7	6	5	4	3	2	1	0
WDTDR7	WDTDR6	WDTDR5	WDTDR4	WDTDR3	WDTDR2	WDTDR1	WDTDR0
W	W	W	W	W	W	W	W

Initial value : FFH

WDTDR[7:0] 设置周期

WDT 中断= (BIT 中断) \* (WDTDR 值+1)

注意) 不要写 “0”。

WDTCR (WDT 控制寄存器) : 8DH							
7	6	5	4	3	2	1	0
WDTEN	WDTRSON	WDTCL	-	-	-	WDTCK	WDTIFR
RW	RW	RW	-	-	-	RW	RW

Initial value : 00H

WDTEN	WDT	控制操作
	0	禁止
	1	打开
WDTRSON	WDT	复位控制
	0	8-bit 独立运行定时器
	1	看门狗复位打开
WDTCL		清除 WDT 计数器
	0	自由运行
	1	清除 WDT 计数器(一个周期后自动清除)
WDTCK		WDT 时钟选择
	0	BIT 溢出为 WDT 提供时钟(WDTRC 关闭)
	1	WDTRC 为 WDT 提供时钟(WDTRC 使能)
WDTIFR		当 WDT 中断产生时置 '1'。写入 0 清除或者被 INT_ACK 信号自动清除。
	0	没有 WDT 中断产生
	1	有 WDT 中断产生

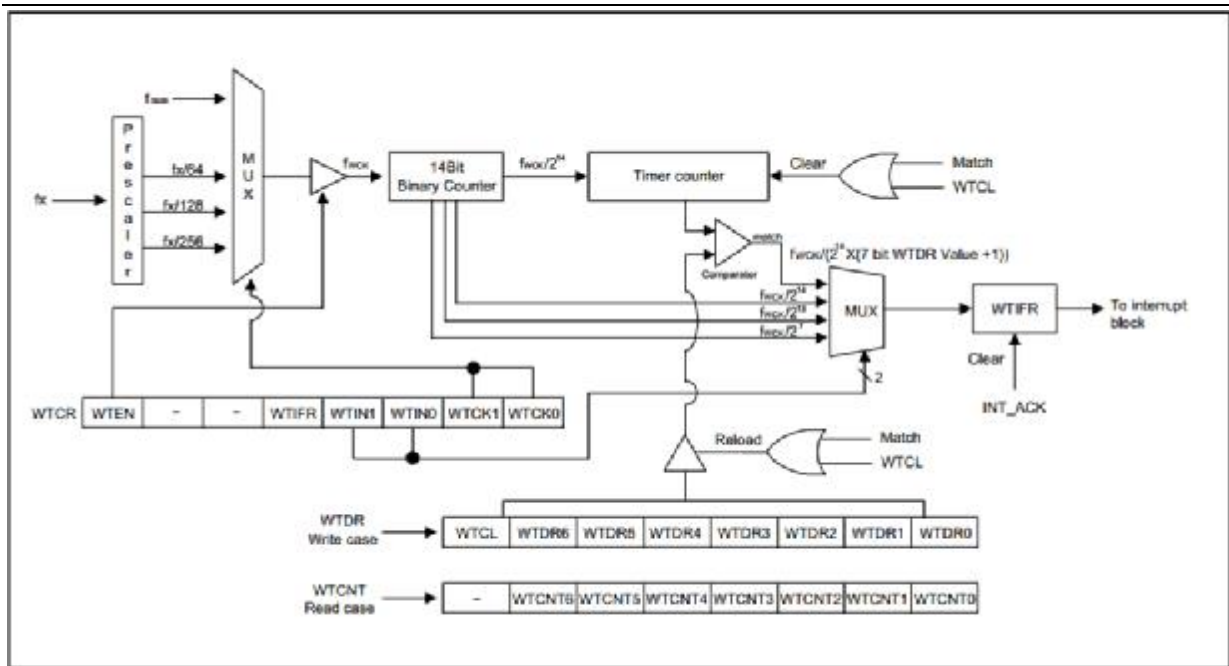
## 7.4、WT

### 7.4.1、概述

WT 有 RTC(Real Time Clock)操作功能。通常用于 RTC 操作。WT 的内部结构由时钟选择电路，定时器计数电路，输出选择电路和控制寄存器组成。WTCR 寄存器用来控制输入时钟，输出间隔，和设置 WTEN 位。可以同时执行也可有单独执行。清除 WTEN 位来停止或复位 WT。即使是在 STOP 模式，副时钟仍存在则 WT 可以进行运行。为了提高分辨率 WT 计数器可以组成 21 位计数器，包括低 14 位和高 7 位二进制计数器。WTDR 寄存器在写入时可以用来清除和设置 WT 的间隔值，在读时序时可以读 7 位计数器的值。

### 7.4.2、方框图





WT 方框图

### 7.4.3、寄存器图

WT 寄存器图

寄存器	Address	方式	默认	说明
WTCNT	89H	R	00H	WT计数器寄存器
WTDR	89H	W	7FH	WT数据寄存器
WTCR	96H	R/W	00H	WT控制寄存器

### 7.4.4、WT 寄存器说明

WT 寄存器由 WTCNT, WTDR, WTCR 组成。WTCR 是 6 位可读、可写寄存器，WTCR 可以控制时钟(WTCK[1:0])，中断间隔(WTIN[1:0])，和开关(WTEN)。WT 有一个中断标志位(WTIFR)。

### 7.4.5、WT 寄存器说明

**WTCNT (WT 计数寄存器: 只读寄存器) : 89H**

7	6	5	4	3	2	1	0
-	WTCNT6	WTCNT5	WTCNT4	WTCNT3	WTCNT2	WTCNT1	WTCNT0
-	R	R	R	R	R	R	R

Initial value : 00H

WTCNT[6:0] WT Counter

**WTDR (WT 数据寄存器: 只写寄存器) : 89H**

7	6	5	4	3	2	1	0
WTCL	WTDR6	WTDR5	WTDR4	WTDR3	WTDR2	WTDR1	WTDR0
RW	W	W	W	W	W	W	W

Initial value : 7FH

- WTCL           清除 WT 计数
- 0       自由振荡
- 1       清除 WT 计数(一个周期后自动清除)
- WTDR[6:0]     设置 WT 周期
- WT 中断间隔 =  $fwck / (2^{14} * (7bit\ WTDR\ Value + 1))$
- 注意) 不要写入 “0”

**WTCCR (WT 控制寄存器) : 96H**

7	6	5	4	3	2	1	0
WTEN	-	-	WTIFR	WTIN1	WTIN0	WTCK1	WTCK0
RW	-	-	RW	RW	RW	RW	RW

Initial value : 00H

- WTEN           WT 控制操作
- 0       禁止
- 1       打开
- WTIFR          当 WT 中断产生时置 ‘1’。写入 0 进行清除或通过 INT\_ACK 信号自动清零
- 0       没有 WT 中断产生
- 1       有 WT 中断产生
- WTIN[1:0]     确定中断间隔
- |       |       |   |
|-------|-------|---|
| WTIN1 | WTIN0 | 说明  |
| 0     | 0     | $fwck / 2^7$                                |
| 0     | 1     | $fwck / 2^{13}$                             |
| 1     | 0     | $fwck / 2^{14}$                             |
| 1     | 1     | $fwck / (2^{14} * (7bit\ WTDR\ Value + 1))$ |
- WTCK[1:0]     确定时钟
- |       |       |            |
|-------|-------|------------|
| WTCK1 | WTCK0 | 说明         |
| 0     | 0     | fSUB       |
| 0     | 1     | $fX / 256$ |
| 1     | 0     | $fX / 128$ |
| 1     | 1     | $fX / 64$  |
- 注意) fX - 系统时钟 (Where  $f_x = 4.19MHz$ )
- fSUB - 副时钟频率(32.768kHz)
- fWCK - 所选 WT 时钟

**7.5、Timer 0**

### 7.5.1、概述

8 位定时器 0 由多路转换器，定时器 0 计数器寄存器，数据寄存器，捕捉数据寄存器和定时器 0 控制寄存器组成(T0CNT, T0DR, T0CDR, T0CR)。

有 3 个操作模式：

- 8-bit 定时器/计数器模式
- 8-bit PWM 输出模式
- 8-bit 捕捉模式

定时器/计数器 0 可以通过内部时钟或外部时钟控制(EC0)。时钟选择通过 T0CK[2:0]位来控制。

- TIMER0 时钟:  $f_x/2$ , 4, 8, 32, 128, 512, 2048 和 EC0

捕捉模式中，通过 EINT10 将数据捕捉到捕捉数据寄存器中(T0CDR)。定时器/计数器模式时，当计算值等于 T0DR 时，T0O 口打开。PWM 模式下可以通过 PWM00 口输出 PWM 波形。

Timer0 操作模式

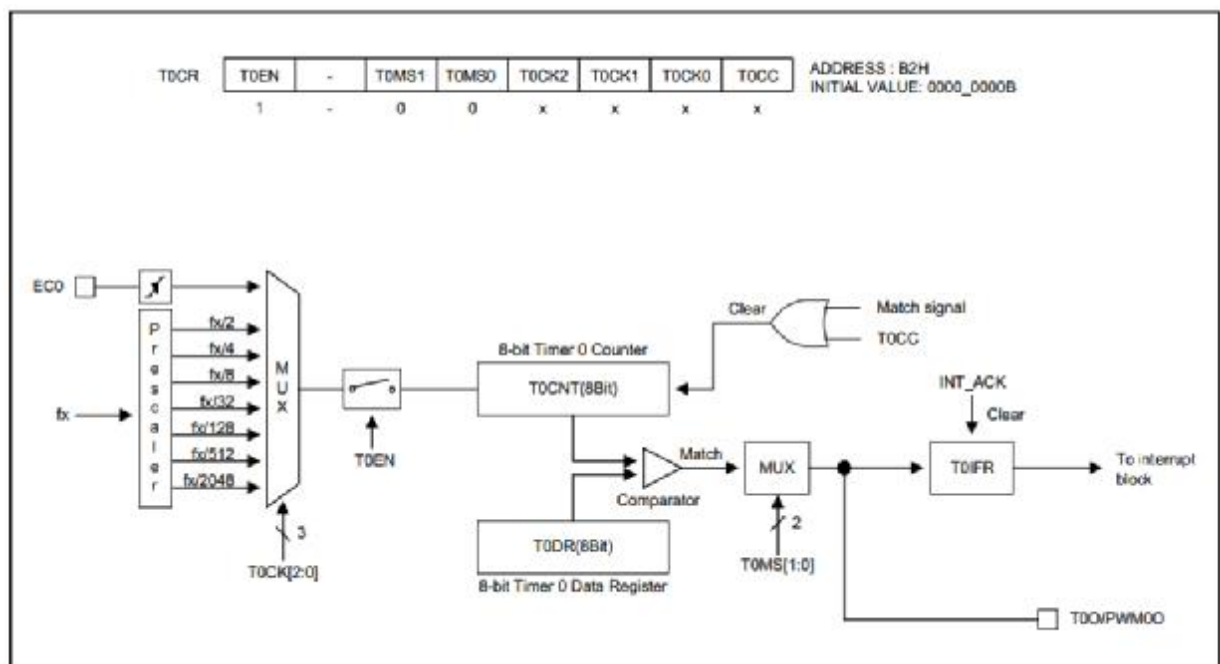
T0EN	T0MS[1:0]	T0CK[2:0]	Timer 0
1	00	XXX	8 Bit 定时器/计数器模式
1	01	XXX	8 Bit PWM 模式
1	1X	XXX	8 Bit 捕捉模式

### 7.5.2、8-Bit 定时器/计数器模式

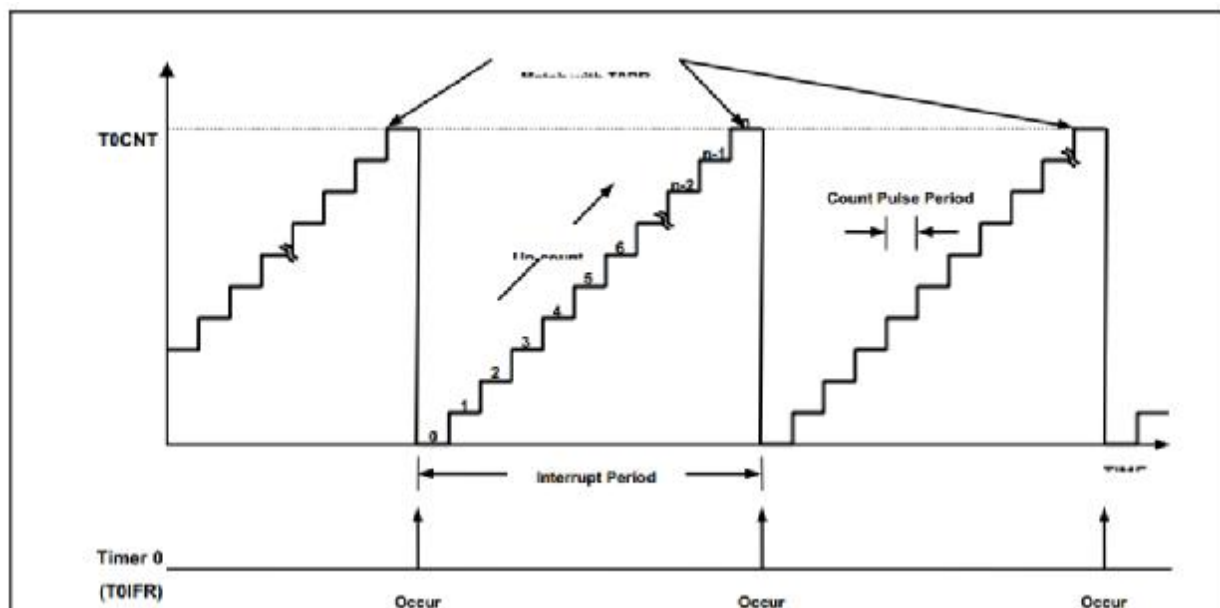
如下图所示为 8-bit 定时器/计数器模式控制选择。

8 位定时器具有计数器和数据寄存器，计数寄存器通过内部或外部时钟输入来累加。Timer0 可用的时钟预分频状态有 2, 4, 8, 32, 128, 512 和 2048(T0CK[2:0])。当 T0CNT 和 T0DR 值相同时，会产生匹配信号同时产生定时器中断 0。T0CNT 值自动被匹配信号清除，也可通过软件清除(T0CC)。

外部时钟(EC0)在上升沿开始累加，如果时钟通过 T0CK[2:0]选择，EC0 口应该设置为输入状态(P26IO 位)。



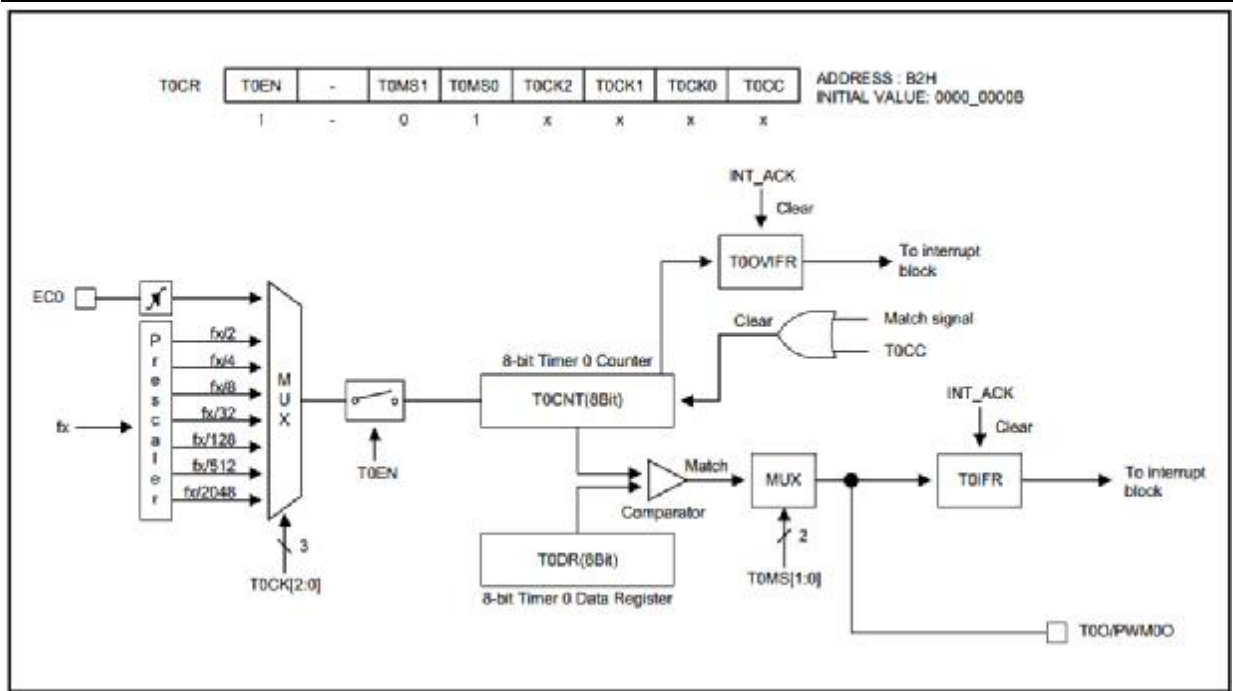
Timer 0 8-Bit 定时器/计数器模式



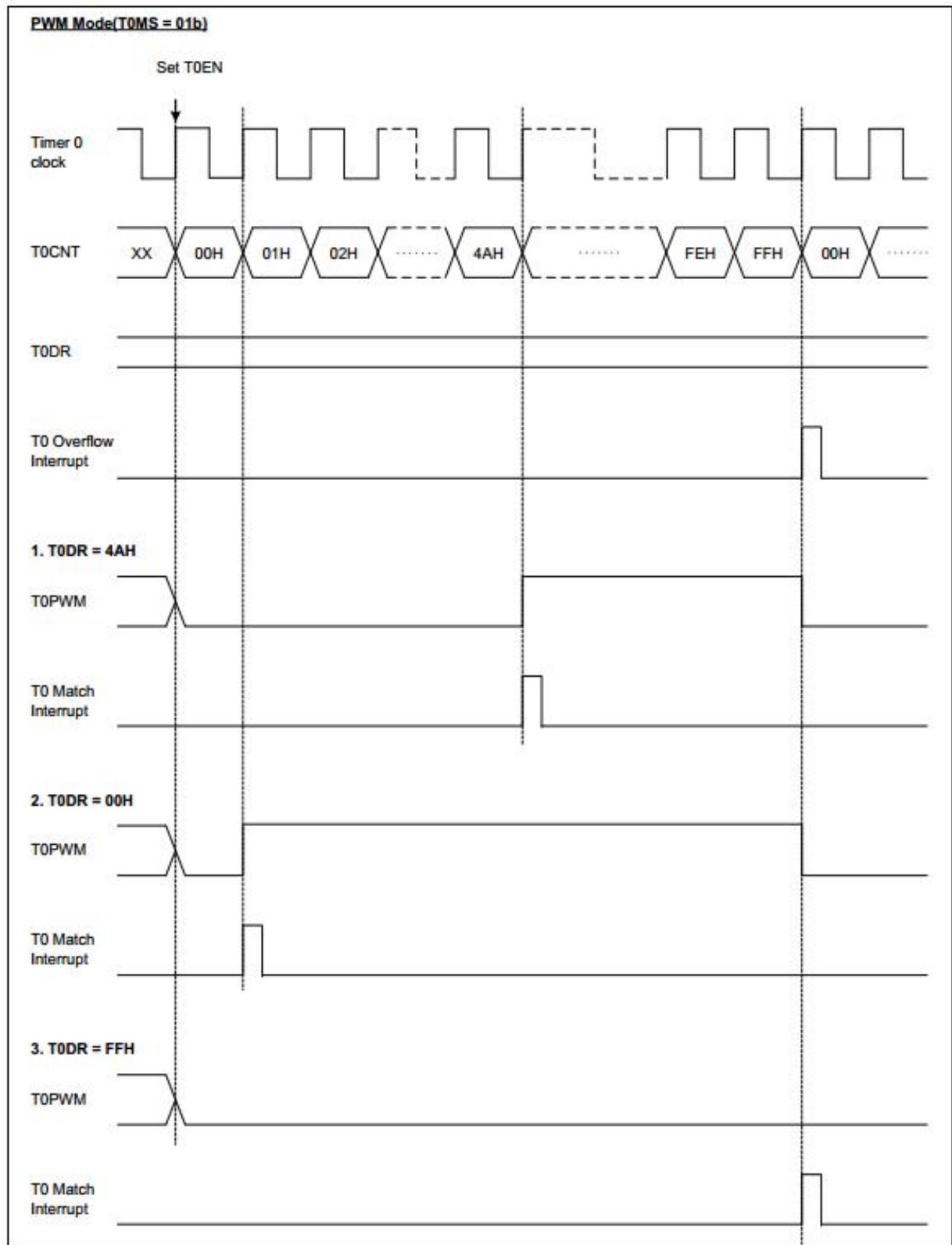
Timer/Counter 0 举例

### 7.5.3、8-Bit PWM 模式

定时器 0 具有高速 PWM 功能。在 PWM 模式中，T00/PWM00 引脚输出 8 位分辨率的 PWM。该脚需要通过设置 P3FSR[5]位，将其配置为 PWM 输出口。在 8 位定时器/计数器模式时，当计数值与 T0DR 值相同时会产生一个匹配信号。在定时器 0 模式下，当 T0CNT 和 T0DR 相同时，产生匹配信号同时产生定时器中断。在 PWM 模式时，会一直运行直到“FFH”时溢出，然后继续从“00H”开始。当计数溢出时定时器 0 溢出中断产生。T0CNT 的值通过软件 T0CC 位清除。



定时器 0 8-Bit PWM 模式



在 PWM 模式下的 PWM 波形输出

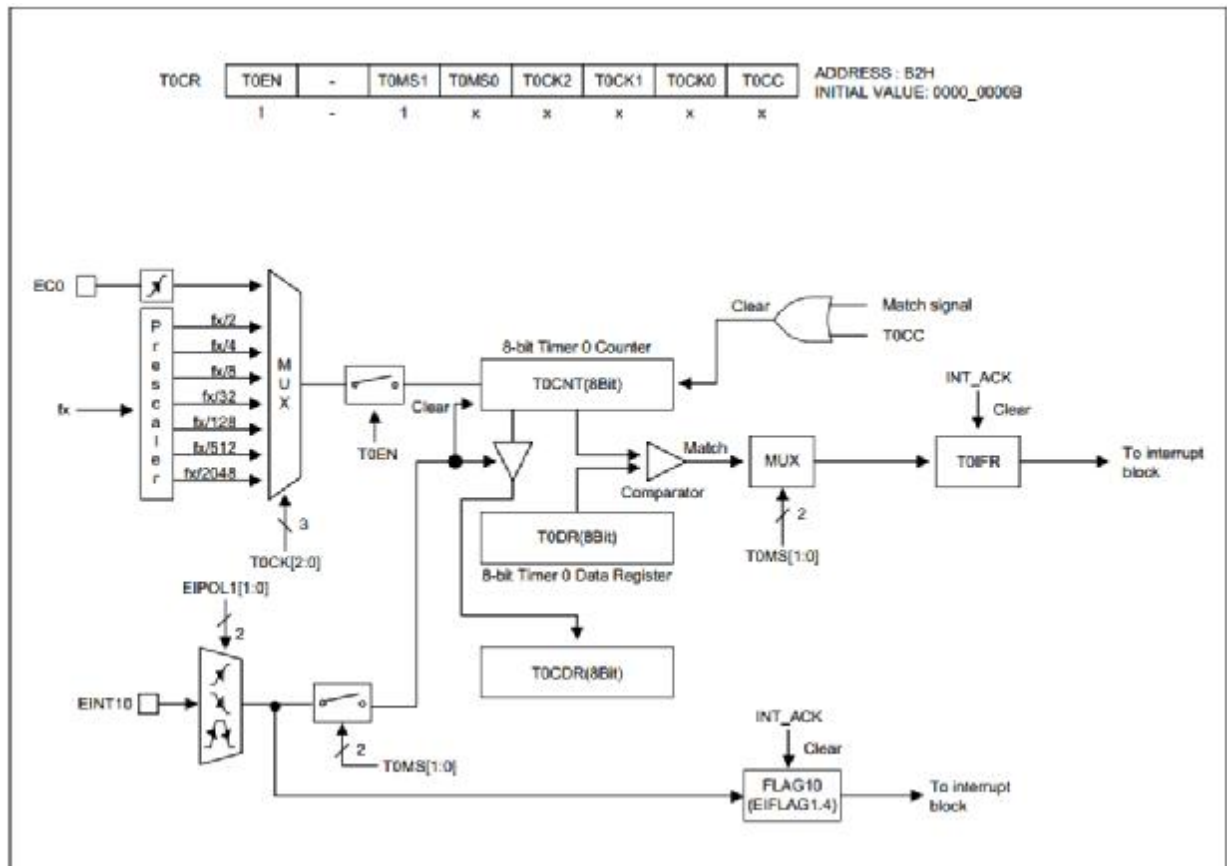
### 7.5.4、8-Bit 捕捉模式

设置 T0MS[1:0]位为“1x”把定时器 0 设为捕捉模式。时钟可以使用内部和外部时钟。当 T0CNT 的值和 T0DR 相同时，它基本上和定时器/计数器模式以及中断的发生具有相同的功能。T0CNT 值被匹配信号自动清零或者通过软件(T0CC)清零。

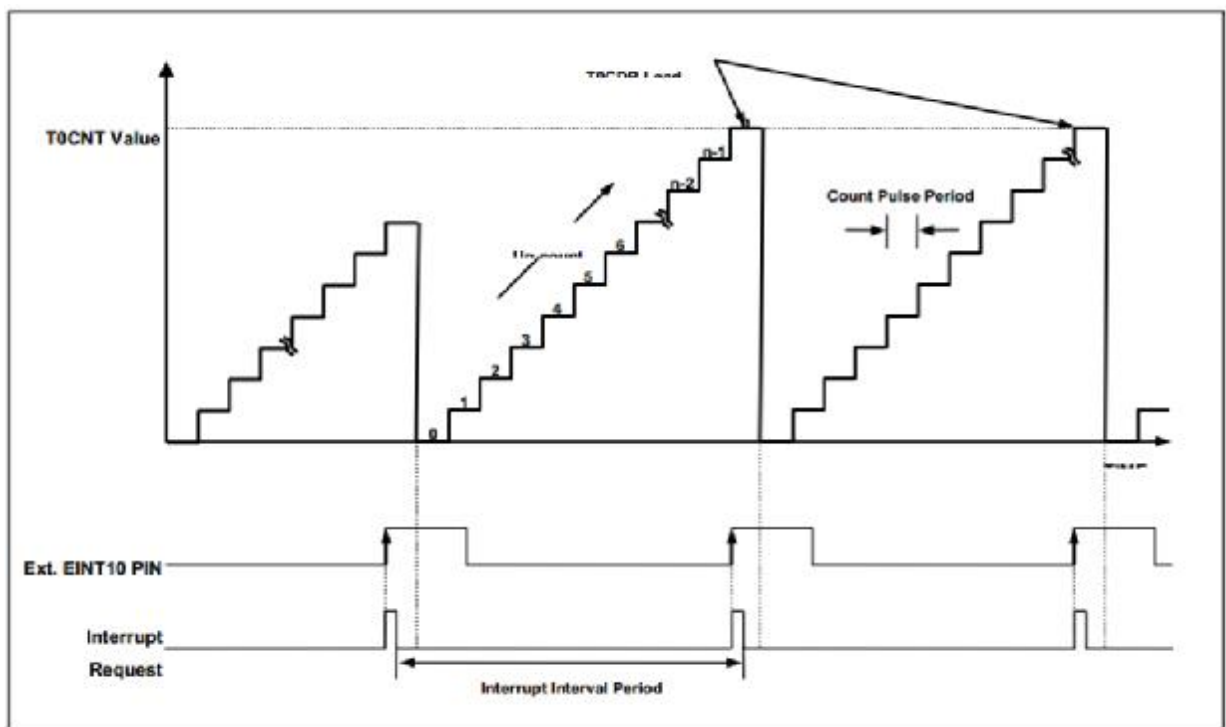
当捕捉的脉冲信号宽度大于定时器周期最大值时，捕捉模式下的定时器中断是非常有用的。捕捉的结果装载 T0CDR。在定时器 0 捕捉模式下，(T0O)波形输出是无法使用的。

根据 EIPOL1 寄存器设置，外部中断触发功能 EINT10 可以选择。当然，EINT10 引脚必须设置为输入口。

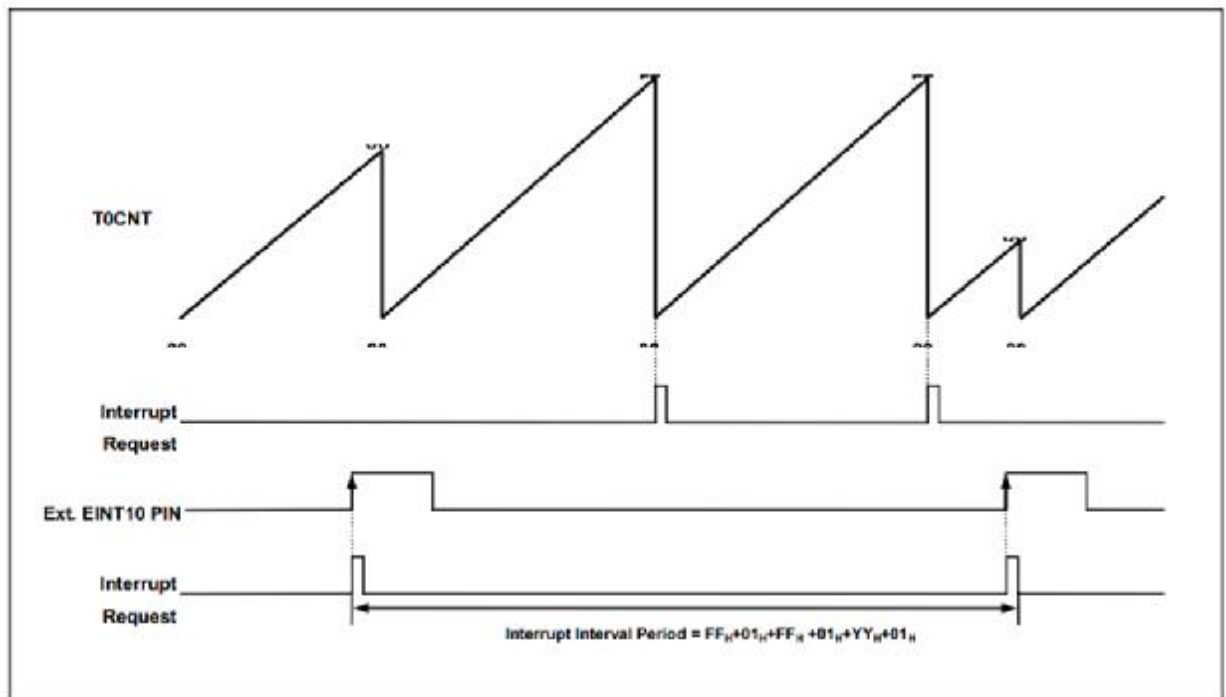
T0CDR 和 T0DR 使用相同的地址。在捕捉模式时，读取时读 T0CDR 而不是 T0DR，但是写入时是 T0DR。



定时器 0 8-Bit 捕捉模式



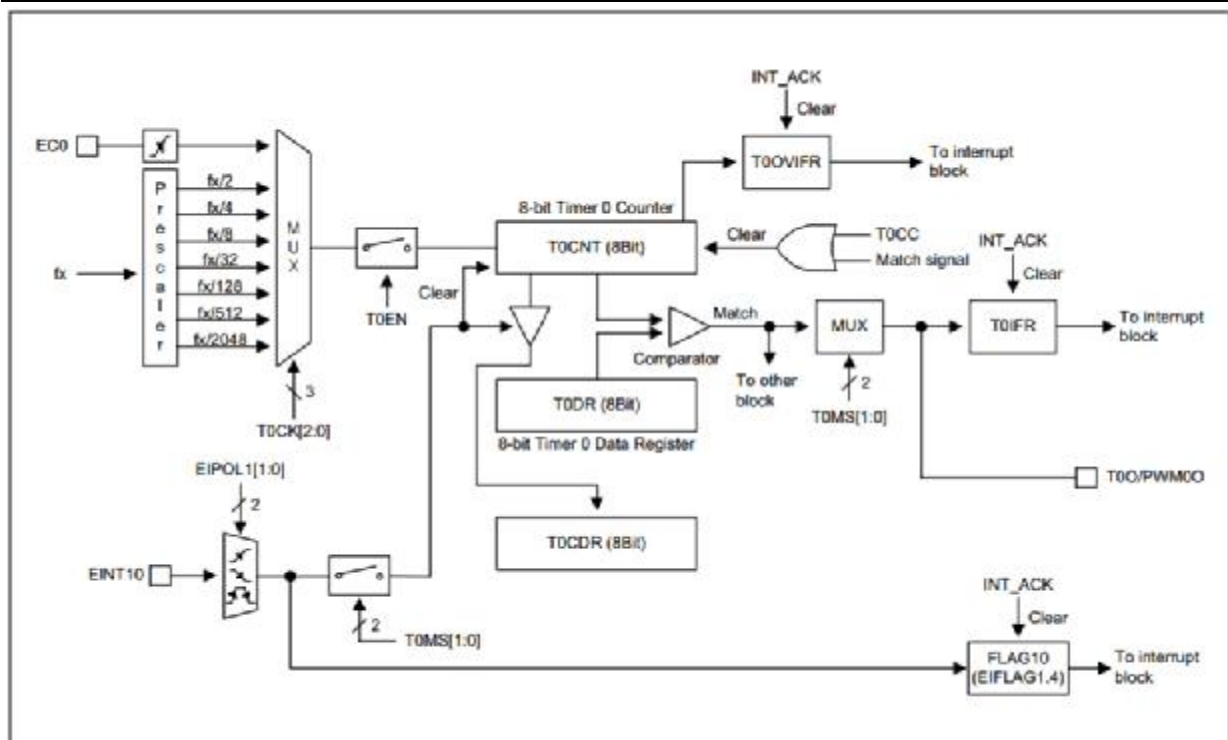
输入捕捉模式操作



捕捉模式中的定时器溢出

### 7.5.5、方框图





8-Bit Timer0 方框图

7.5.6、寄存器图

Timer0 寄存器图

名字	地址	读写方式	默认	说明
T0CNT	B3H	R	00H	Timer 0 计数器寄存器
T0DR	B4H	R/W	FFH	Timer 0 数据寄存器
T0CDR	B4H	R	00H	Timer 0 捕捉数据寄存器
T0CR	B2H	R/W	00H	Timer 0 控制寄存器

7.5.6.1、定时器/计数器 0 寄存器说明

定时器/计数器 0 寄存器由 T0CNT, T0DR, T0CDR, T0CR 以及中断标志位寄存器(IIFLAG)中的 T0IFR 和 T0OVIFR 位组成。

7.5.6.2、定时器/计数器 0 寄存器说明

**T0CNT (Timer 0 计数寄存器) : B3H**

7	6	5	4	3	2	1	0
T0CNT7	T0CNT6	T0CNT5	T0CNT4	T0CNT3	T0CNT2	T0CNT1	T0CNT0
R	R	R	R	R	R	R	R

Initial value : 00H

T0CNT[7:0] T0 Counter

**T0DR (Timer 0 数据寄存器) : B4H**

7	6	5	4	3	2	1	0
T0DR7	T0DR6	T0DR5	T0DR4	T0DR3	T0DR2	T0DR1	T0DR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : FFH

T0DR[7:0] T0 数据

**T0CDR (Timer 0 捕捉数据寄存器: 只读寄存器, 仅在捕捉模式中) : B4H**

7	6	5	4	3	2	1	0
T0CDR7	T0CDR6	T0CDR5	T0CDR4	T0CDR3	T0CDR2	T0CDR1	T0CDR0
R	R	R	R	R	R	R	R

Initial value : 00H

T0CDR[7:0] T0 捕捉数据

**T0CR (Timer 0 控制寄存器) : B2H**

7	6	5	4	3	2	1	0
TOEN	-	T0MS1	T0MS0	T0CK2	T0CK1	T0CK0	T0CC
RW	-	RW	RW	RW	RW	RW	RW

Initial value : 00H

TOEN Timer0 控制

- 0 定时器 0 禁止
- 1 定时器 0 使能

T0MS[1:0] 控制 Timer 0 操作模式

- |       |       |                                 |
|-------|-------|---------------------------------|
| T0MS1 | T0MS0 | 说明                              |
| 0     | 0     | 定时器/计数器模式(T00: toggle at match) |
| 0     | 1     | PWM 模式(可以产生溢出中断)                |
| 1     | x     | 捕捉模式 (可以产生匹配中断)                 |

T0CK[2:0] 选择 Timer0 时钟, fx 是系统时钟频率

- |       |       |       |           |
|-------|-------|-------|-----------|
| T0CK2 | T0CK1 | T0CK0 | 说明        |
| 0     | 0     | 0     | fx/2      |
| 0     | 0     | 1     | fx/4      |
| 0     | 1     | 0     | fx/8      |
| 0     | 1     | 1     | fx/32     |
| 1     | 0     | 0     | fx/128    |
| 1     | 0     | 1     | fx/512    |
| 1     | 1     | 0     | fx/2048   |
| 1     | 1     | 1     | 外部时钟(EC0) |

T0CC 清除 timer0 计数

- 0 No effect
- 1 清除 Timer 0 计数(When write, automatically cleared "0" after being cleared counter)

注意) 参考内部中断标志位寄存器 (IIFLAG) T0 中断标志位

## 7.6、定时器 1

### 7.6.1、概述

16 位定时器 1 由定时器 1A 数据寄存器（高/低），定时器 1B 数据寄存器（高/低）和控制寄存器（高/低）组成(T1ADRH, T1ADRL, T1BDRH, T1BDRL, T1CRH, T1CRL)。

有 4 个操作模式：

- 16-bit 定时器/计数器模式
- 16-bit 捕捉模式
- 16-bit PPG 输出模式(一次 模式)
- 16-bit PPG 输出模式(重复模式)

定时器/计数器 1 可以由内部或外部时钟源(EC1)提供时钟。时钟选择是通过时钟控制选择位(T1CK[2:0])进行选择。

- 定时器 1 的时钟：fx/1, 2, 4, 8, 64, 512, 2048 和 EC1

捕捉模式下，数据由 EINT11 捕捉到捕捉数据寄存器(T1BDRH/T1BDRL)。定时器/计数器模式时 Timer1 通过 T1O 输出计数器和数据寄存器的比较结果。PPG 模式下也可以通过 PWM1O 口输出 PWM 波形。

Timer1 操作模式

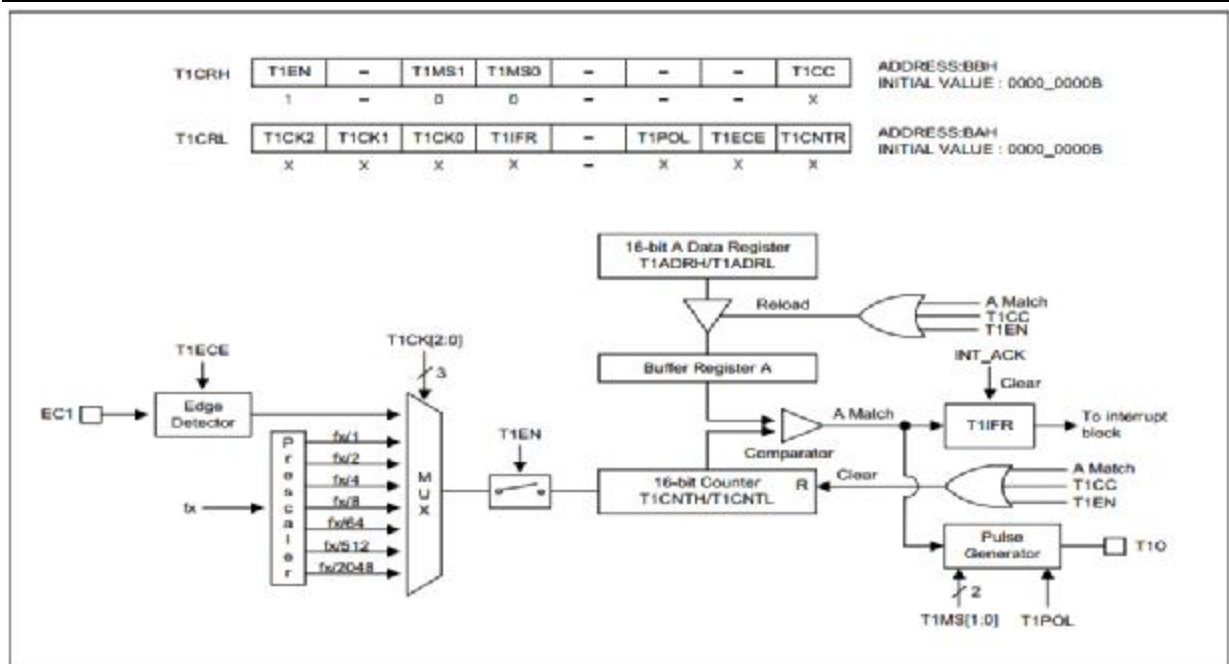
T1EN	P1FSRL[4:3]	T1MS[1:0]	T1CK[2:0]	Timer 1
1	01	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	01	10	XXX	16 Bit PPG Mode (one-shot mode)
1	01	11	XXX	16 Bit PPG Mode (repeat mode)

### 7.6.2、16-Bit Timer/Counter 模式

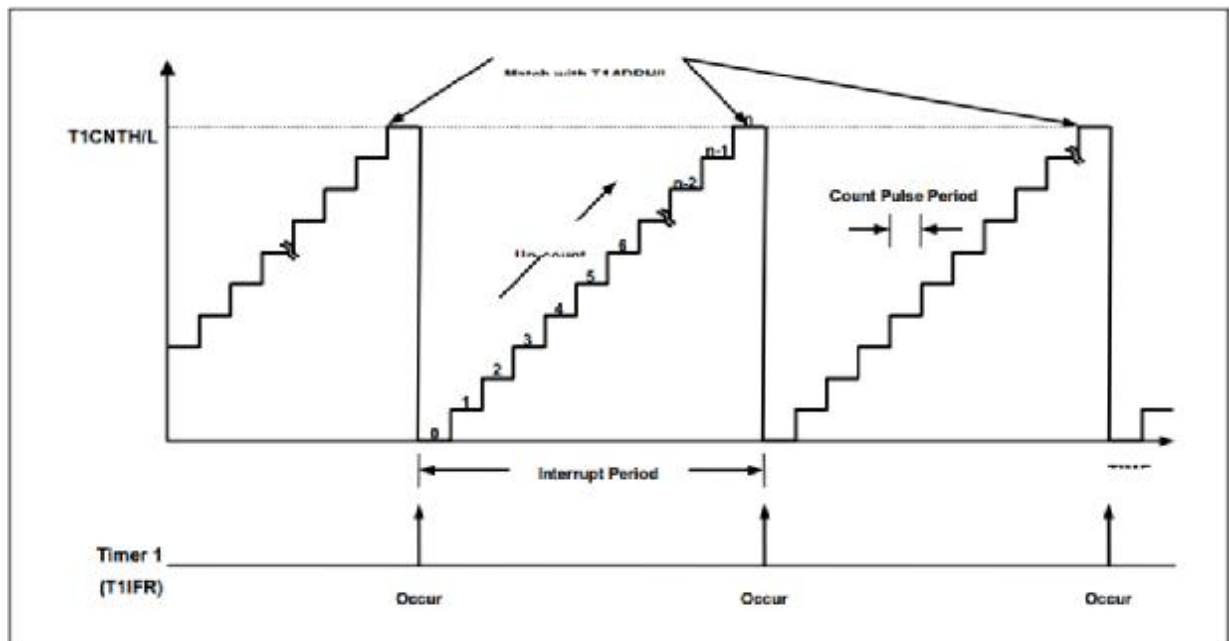
如下图所示为通过寄存器选择 16 位定时器/计数器模式。

它有计数寄存器和数据寄存器。计数寄存器由内部或外部时钟输入进行累加。Timer1 可以使用的输入时钟分频有 1, 2, 4, 8, 64, 512 和 2048。当 T1CNTH, T1CNTL 和 T1ADRH, T1ADRL 值分别相同时，会产生一个匹配信号，同时产生定时器 1 中断。T1CNTH, T1CNTL 值被匹配信号自动清零。也可以通过软件(T1CC)进行清零。

外部时钟(EC1)在上升沿开始计数。如果 EC1 通过 T1CK[2:0]选择时钟源，EC1 口需要被 P11 IO 位设置为输入状态。



Timer1 16-Bit 定时器/计数器模式

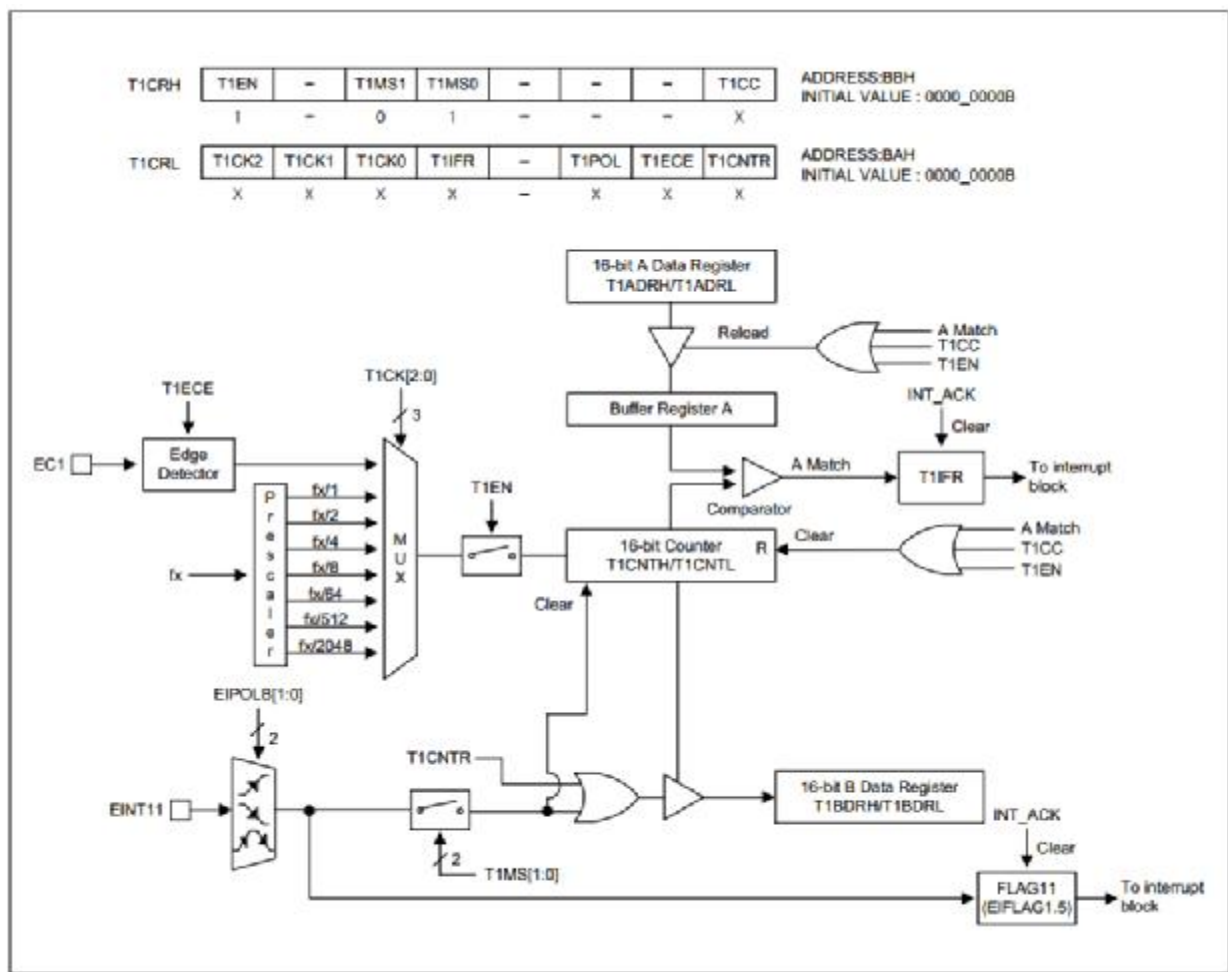


16-Bit 定时器/计数器 1 举例

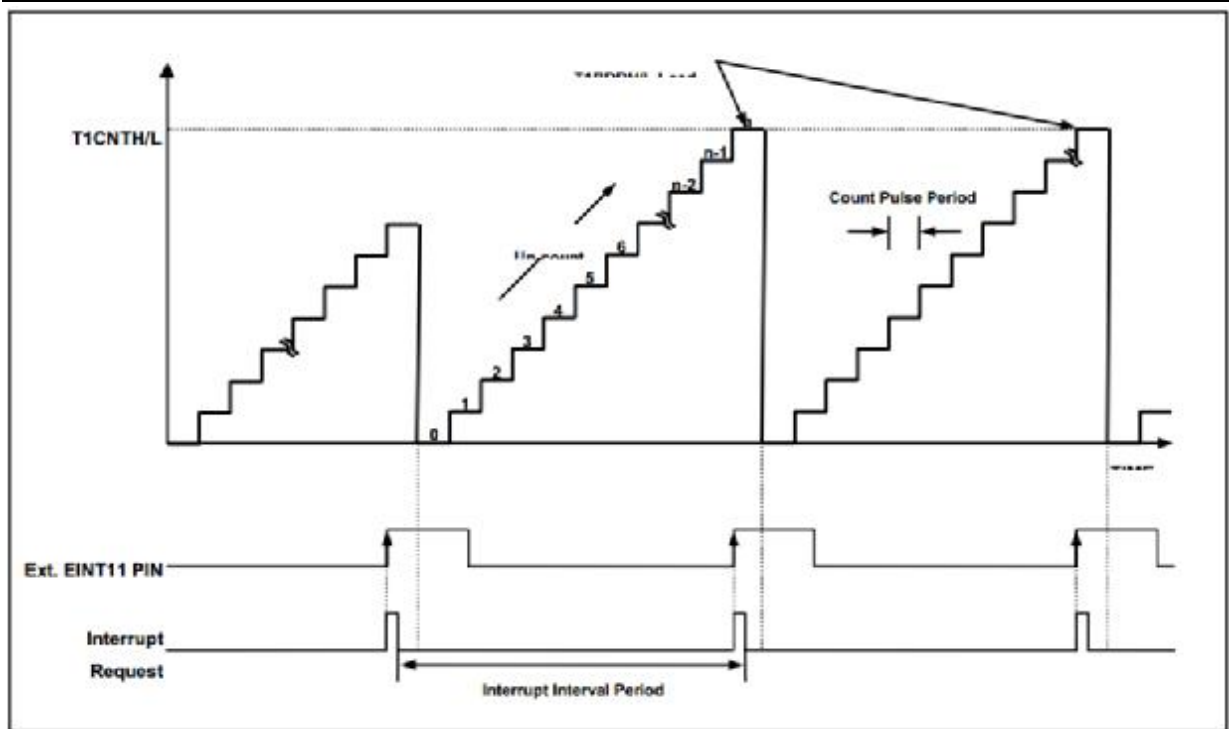
### 7.6.3、16-Bit 捕捉模式

定时器 1 通过把 T1MS[1:0] 设置为 ‘01’ 将其设为 16 位捕捉模式。时钟可以使用内部和外部时钟。当 T1CNTH/T1CNTL 的值和 T1ADRH/T1ADRL 相同时，它基本上和定时器/计数器模式以及中断的发生具有相同的功能。T1CNTH, T1CNTL 值被匹配信号自动清零或者通过软件(T1CC)清零。

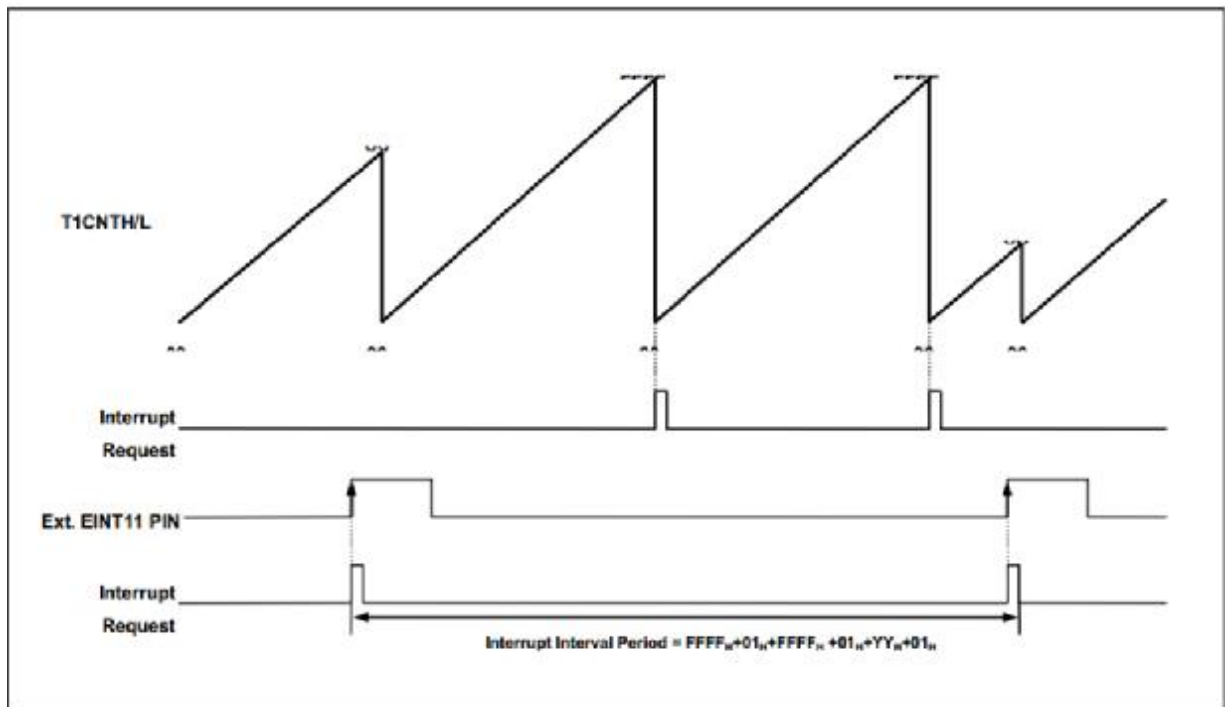
当捕捉的脉冲信号宽度大于定时器周期最大值时，捕捉模式下的定时器中断是非常有用的。捕捉的结果装载到 T1BDRH/T1BDR。根据 EIPOL 寄存器设置选择外部中断 EINT11 触发边沿，当然，EINT11 引脚必须设为输入状态。



Timer1 16-Bit 捕捉模式



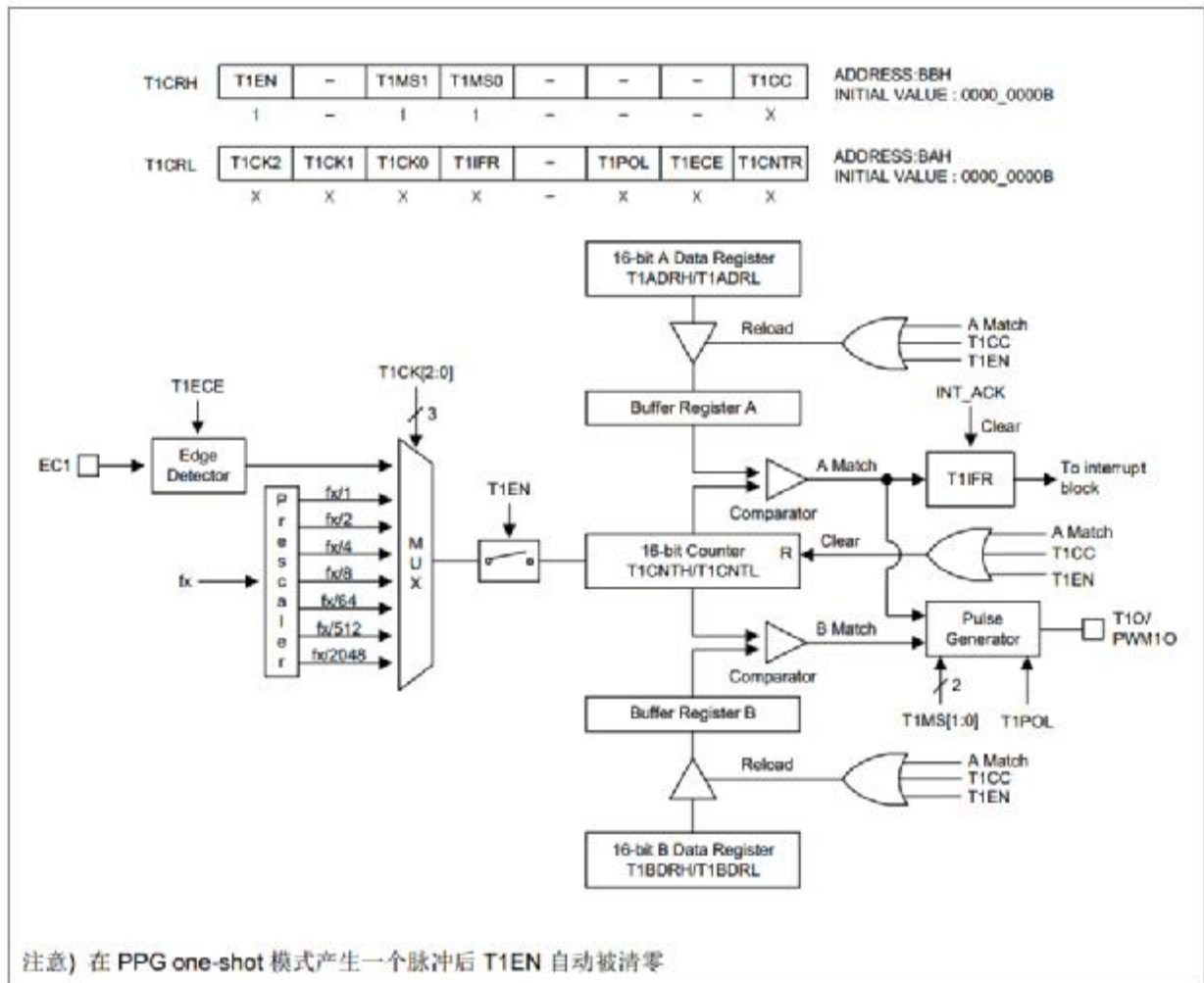
定时器 1 输入捕捉模式操作



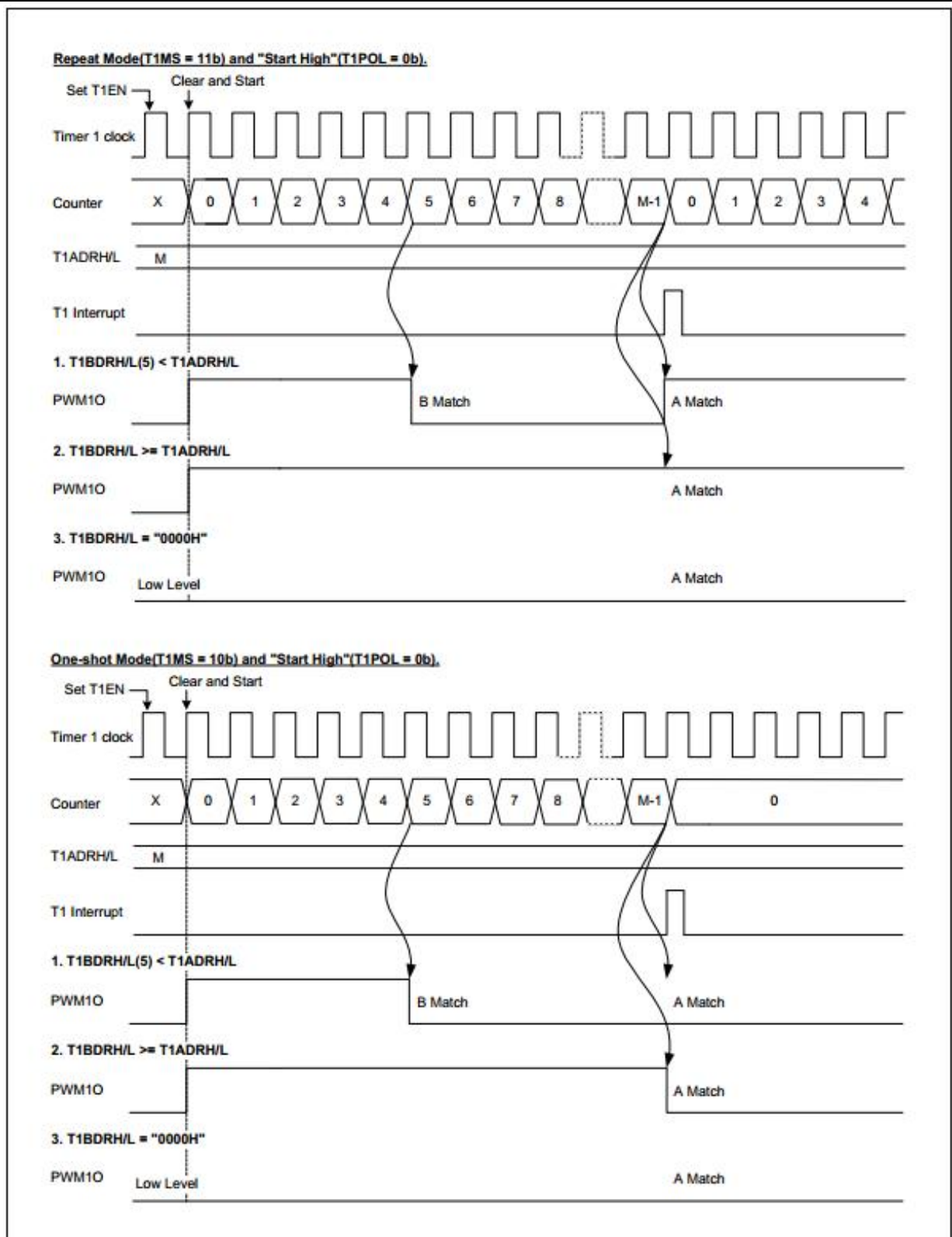
捕捉模式定时器溢出

### 7.6.4、16-Bit PPG 模式

定时器 1 有 PPG(可编程脉冲发生)功能。PPG 模式，T1O/PWM1O 引脚输出取决于 16 位分辨率 PWM 输出。通过将 P1FSRL[4:3]设置为 ‘01’ 将该引脚配置为 PWM 输出。PWM 的周期由 T1ADRH/T1ADRL 的值决定。PWM 占空比输出由 T1BDRH/T1BDRL 的值决定。



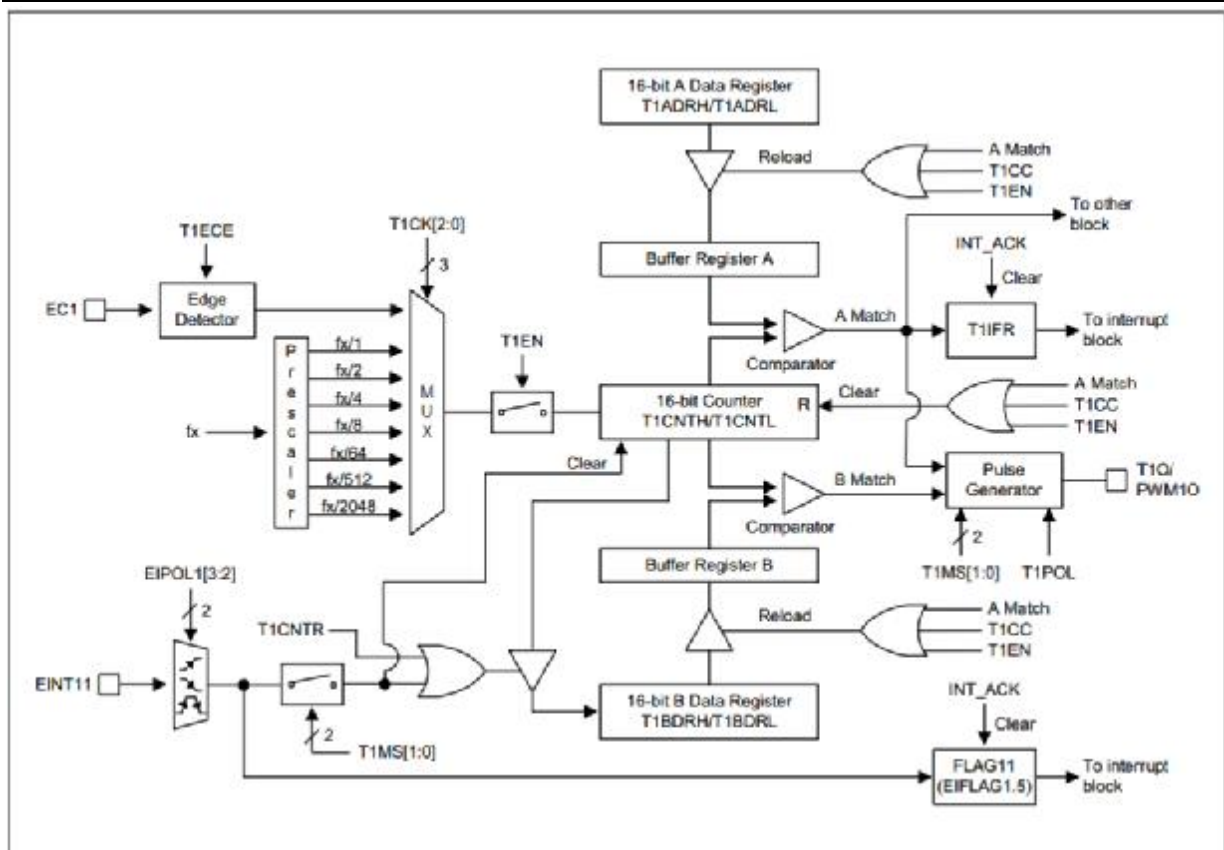
Timer1 16-BitPPG 模式



定时器 1 16-Bit PPG 模式时序图

7.6.5、方框图





16-Bit Timer1 方框图

7.6.6、寄存器图

定时器 1 寄存器图

名字	地址	方向	默认	说明
T1ADRH	BDH	R/W	FFH	Timer 1 A Data High Register
T1ADRL	BCH	R/W	FFH	Timer 1 A Data Low Register
T1BDRH	BFH	R/W	FFH	Timer 1 B Data High Register
T1BDRL	BEH	R/W	FFH	Timer 1 B Data Low Register
T1CRH	BBH	R/W	00H	Timer 1 Control High Register
T1CRL	BAH	R/W	00H	Timer 1 Control Low Register

7.6.6.1、定时器/计数器 1 寄存器说明

定时器/计数器 1 由 T1ADRH, T1ADRL, T1BDRH, T1BDRL, T1CRH 和 T1CRL 组成。

7.6.6.2、定时器/计数器 1 寄存器说明

**T1ADRH (Timer 1 A 数据高位寄存器) : BDH**

7	6	5	4	3	2	1	0
T1ADRH7	T1ADRH6	T1ADRH5	T1ADRH4	T1ADRH3	T1ADRH2	T1ADRH1	T1ADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1ADRH[7:0] T1A 数据高字节

**T1ADRL (Timer 1 A 数据低位寄存器) : BCH**

7	6	5	4	3	2	1	0
T1ADRL7	T1ADRL6	T1ADRL5	T1ADRL4	T1ADRL3	T1ADRL2	T1ADRL1	T1ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1ADRL[7:0] T1A 数据低字节

注意) 在 PPG 模式时, T1ADRH/T1ADRL 不要写 “0000H”

**T1BDRH (Timer 1 B 数据高位寄存器) : BFH**

7	6	5	4	3	2	1	0
T1BDRH7	T1BDRH6	T1BDRH5	T1BDRH4	T1BDRH3	T1BDRH2	T1BDRH1	T1BDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1BDRH[7:0] T1B 数据高字节

**T1BDRL (Timer 1 B 数据低位寄存器) : BEH**

7	6	5	4	3	2	1	0
T1BDRL7	T1BDRL6	T1BDRL5	T1BDRL4	T1BDRL3	T1BDRL2	T1BDRL1	T1BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T1BDRL[7:0] T1B 数据低字节

**T1CRH (Timer 1 控制高位寄存器) : BBH**

7	6	5	4	3	2	1	0
T1EN	-	T1MS1	T1MS0	-	-	-	T1CC
R/W	-	R/W	R/W	-	-	-	R/W

Initial value : 00H

T1EN 控制 Timer 1

0 Timer 1 禁止

1 Timer 1 使能 (计数清零然后开始计数)

T1MS[1:0] 控制 Timer 1 操作模式

T1MS1 T1MS0 说明

0 0 定时器/计数器 模式 (T1O: toggle at A match)

0 1 捕捉模式 (The A match interrupt can occur)

1 0 PPG one-shot mode (PWM1O)

1 1 PPG repeat mode (PWM1O)

T1CC 清除 Timer 1 计数

0 No effect

1 清除 Timer 1 counter (When write, automatically cleared “0” after being cleared counter)

**T1CRL (Timer 1 控制低位寄存器) : BAH**

7	6	5	4	3	2	1	0
T1CK2	T1CK1	T1CK0	T1IFR	-	T1POL	T1ECE	T1CNTR
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Initial value : 00H

T1CK[2:0] 选择 Timer 1 时钟, fx 是系统主时钟频率

	T1CK2	T1CK1	T1CK0	Description
	0	0	0	fx/2048
	0	0	1	fx/512
	0	1	0	fx/64
	0	1	1	fx/8
	1	0	0	fx/4
	1	0	1	fx/2
	1	1	0	fx/1
	1	1	1	外部时钟 (EC1)
T1IFR	T1 中断产生时置 ‘1’。在该位写入 0 进行清零或通过 INT_ACK 信号自动清零			
	0	没有 T1 中断产生		
	1	有 T1 中断产生		
T1POL	T1O/PWM1O Polarity Selection			
	0	高电平有效 (低电平时 T1O/PWM1O 禁止)		
	1	低电平有效 (高电平时 T1O/PWM1O 禁止)		
T1ECE	Timer1 外部时钟脉冲边沿选择			
	0	下降沿		
	1	上升沿		
T1CNTR	Timer 1 计数器读取控制			
	0	No effect		
	1	把计数器的值加载到 B 数据寄存器(写入时加载后自动清零)		

## 7.7、Timer2

### 7.7.1、概述

16 位定时器 2 由数据转换器, T2ADRH, T2ADRL, T2BDRH, T2BDRL, T2CRH, T2CRL 组成。有 4 个操作模式:

- 16-bit 定时器/计数器模式
- 16-bit 捕捉模式
- 16-bit PPG 输出模式(one-shot mode)
- 16-bit PPG 输出模式(repeat mode)

定时器 2 可以由内部、外部时钟源或者 T1A Match 提供时钟。通过控制设置(T2CK[2: 0])位来选取时钟。

- 定时器 2 时钟: fx/1, 2, 4, 8, 64, 512, 2048, EC2 和 T1A Match

捕捉模式下, 数据通过 EINT12 将数据捕捉到捕捉数据寄存器(T2BDRH/T2BDRL)。在定时器/计数器模式下, 当计数值等于 T2ADRH/L 时, T2O 口打开。在 PPG 模式时 PWM2O 口也可以输出 PWM 波形。

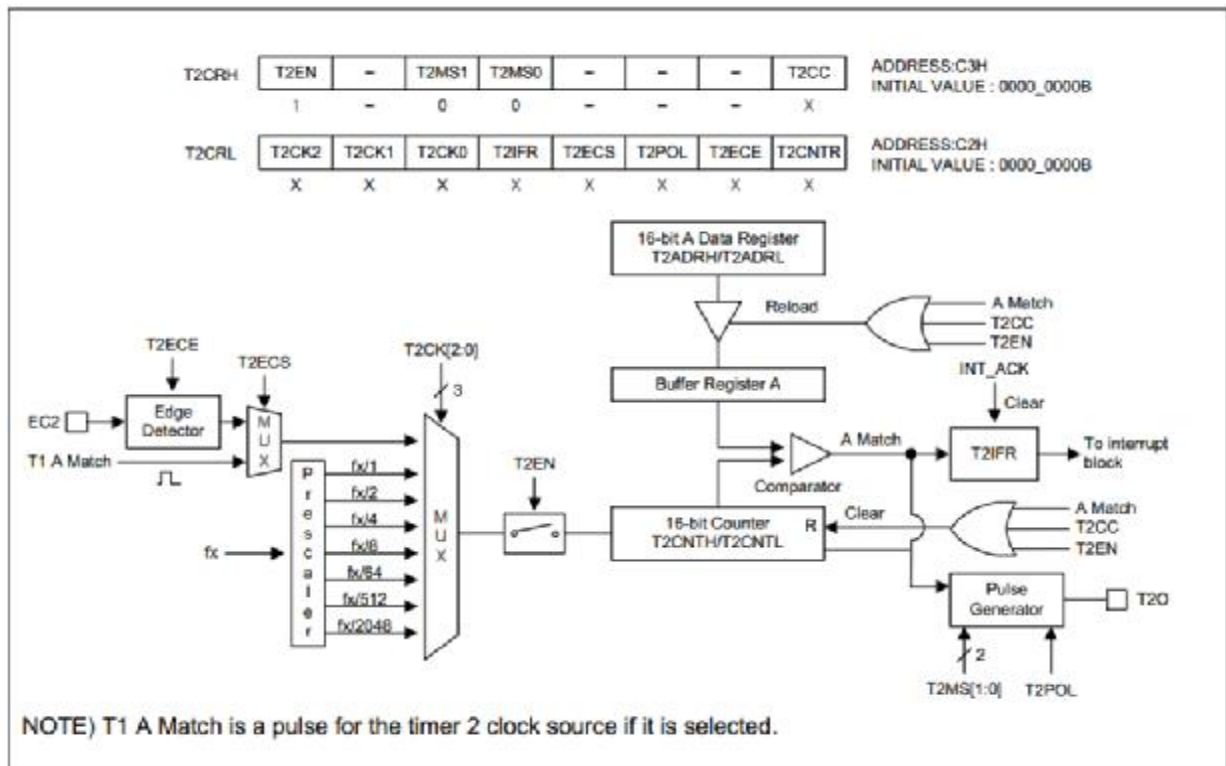
#### Timer2 操作模式

T2EN	P1FSRL[6:5]	T2MS[1:0]	T2CK[2:0]	Timer 2
1	01	00	XXX	16 Bit Timer/Counter Mode
1	00	01	XXX	16 Bit Capture Mode
1	01	10	XXX	16 Bit PPG Mode (one-shot mode)
1	01	11	XXX	16 Bit PPG Mode (repeat mode)

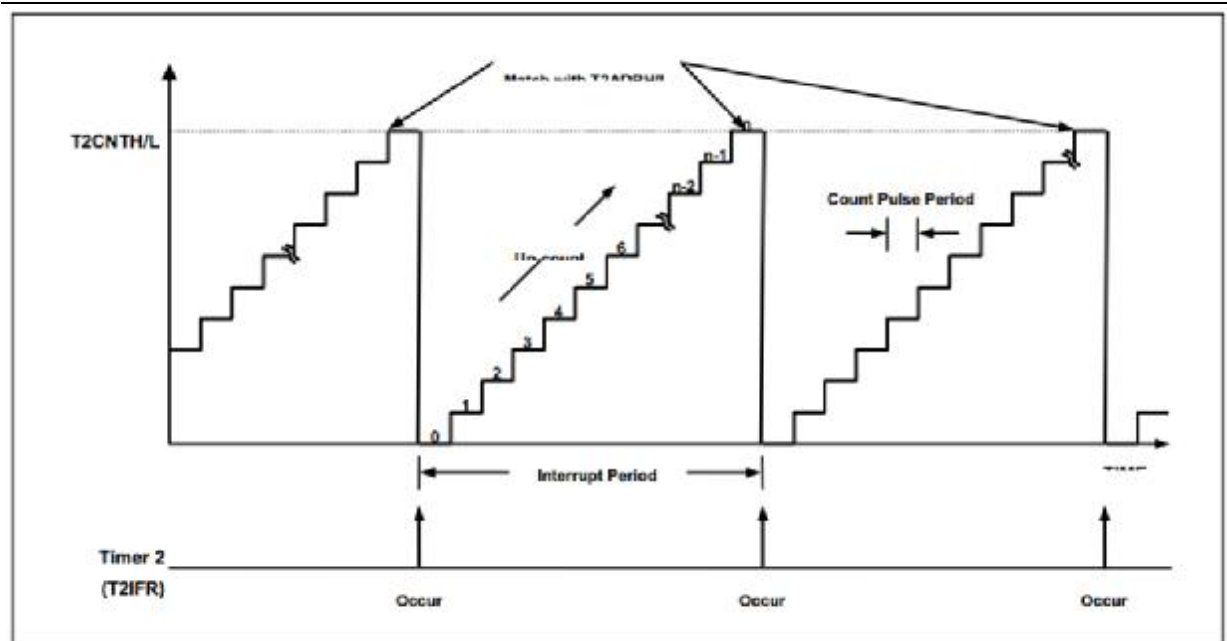
### 7.7.2、16-Bit 定时器/计数器模式

如下图所示为定时器/计数器模式控制图。

它由计数寄存器和数据寄存器。计数寄存器通过内、外部时钟或 timer1AMatch 时钟计数。定时器 2 可以使用的时钟分频有 1,2,4,8,64,512 和 2048, EC2 和 T1A Match(T2CK[2:0])。当 T2CNTH/T2CNTL 和 T2ADRH/T2ADRL 的值相等时，产生匹配信号同时产生定时器 2 中断。T2CNTH/T2CNTL 值被匹配信号自动清零，也可以通过软件清零(T2CC)。



6-Bit 定时器/计数器模式



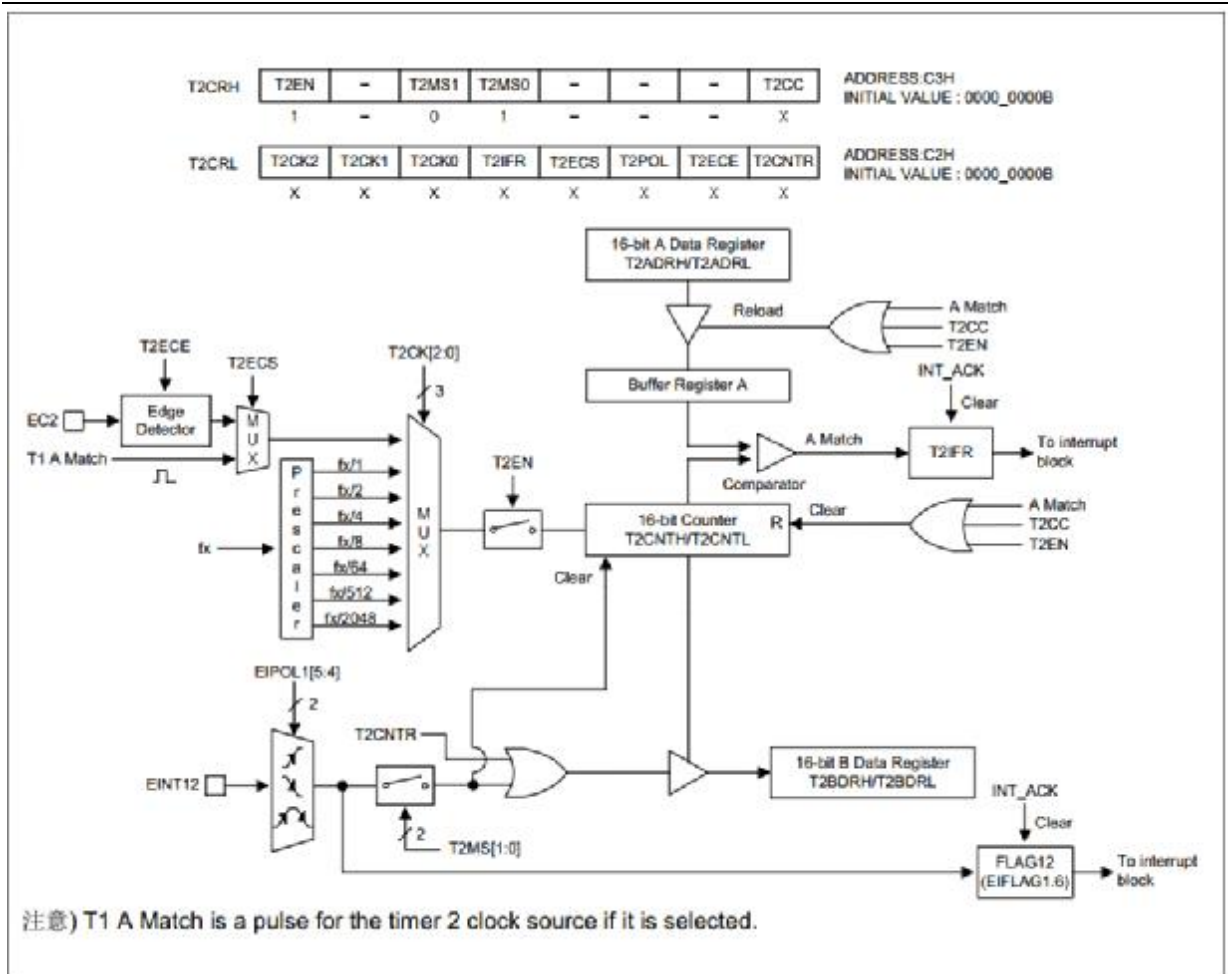
16-Bit 定时器/计数器 2 举例

### 7.7.3、16-Bit 捕捉模式

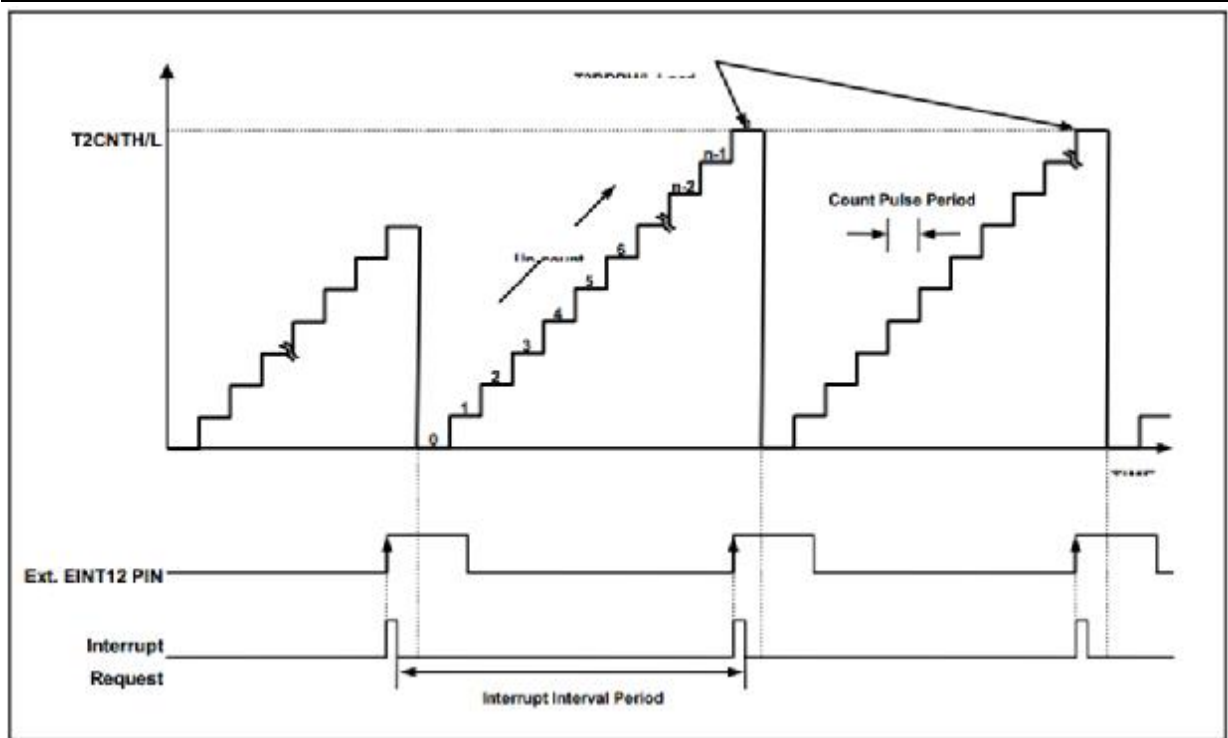
设置 T2MS[1:0]为‘01’将定时器 2 配置为捕捉模式。时钟源可采用内部时钟。当 T2CNTH/T2CNTL 的值和 T2ADRH/T2ADRL 相同时，它基本上和定时器/计数器模式以及中断的发生具有相同的功能。T2CNTH, T2CNT 值被匹配信号自动清零或者通过软件(T2CC)清零。

当捕捉的脉冲信号宽度大于定时器周期最大值时，捕捉模式下的定时器中断是非常有用的。捕捉结果加载到 T2BDRH/T2BDRL。在定时器 2 捕捉模式，(T2O)波形输出是不可用的。

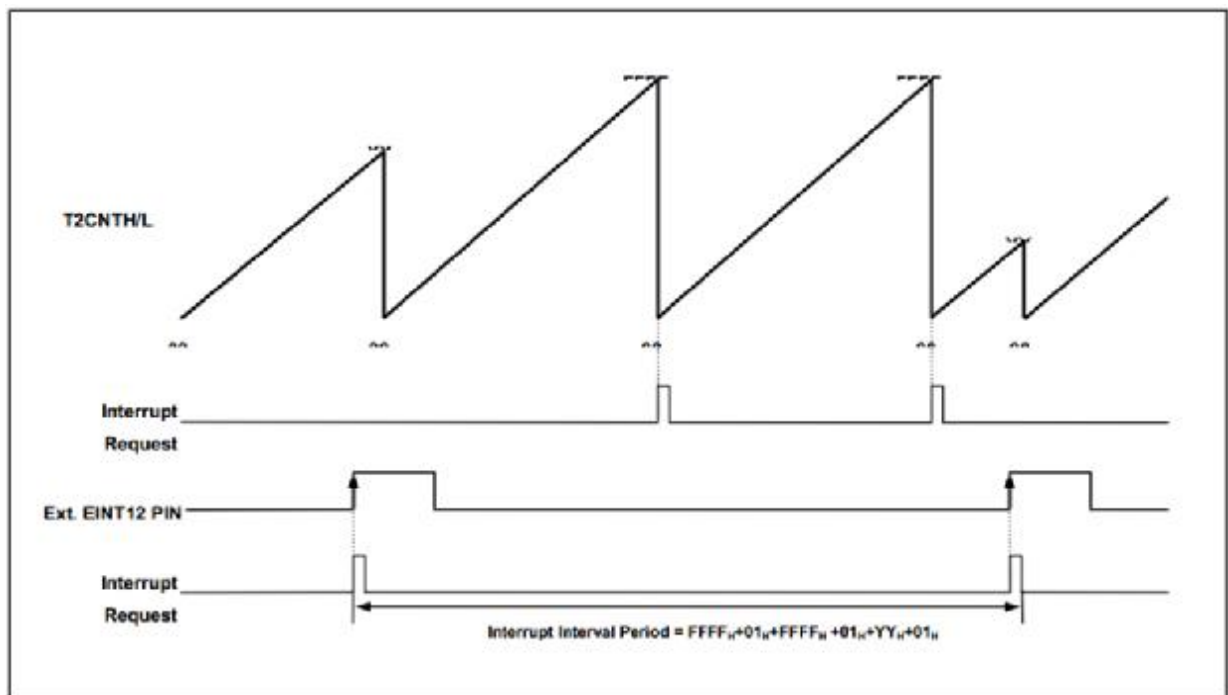
根据 EIPOL1 寄存器设置，选择 EINT12 触发边沿。当然，EINT12 引脚需要设置位输入状态。



16-Bit 捕捉模式



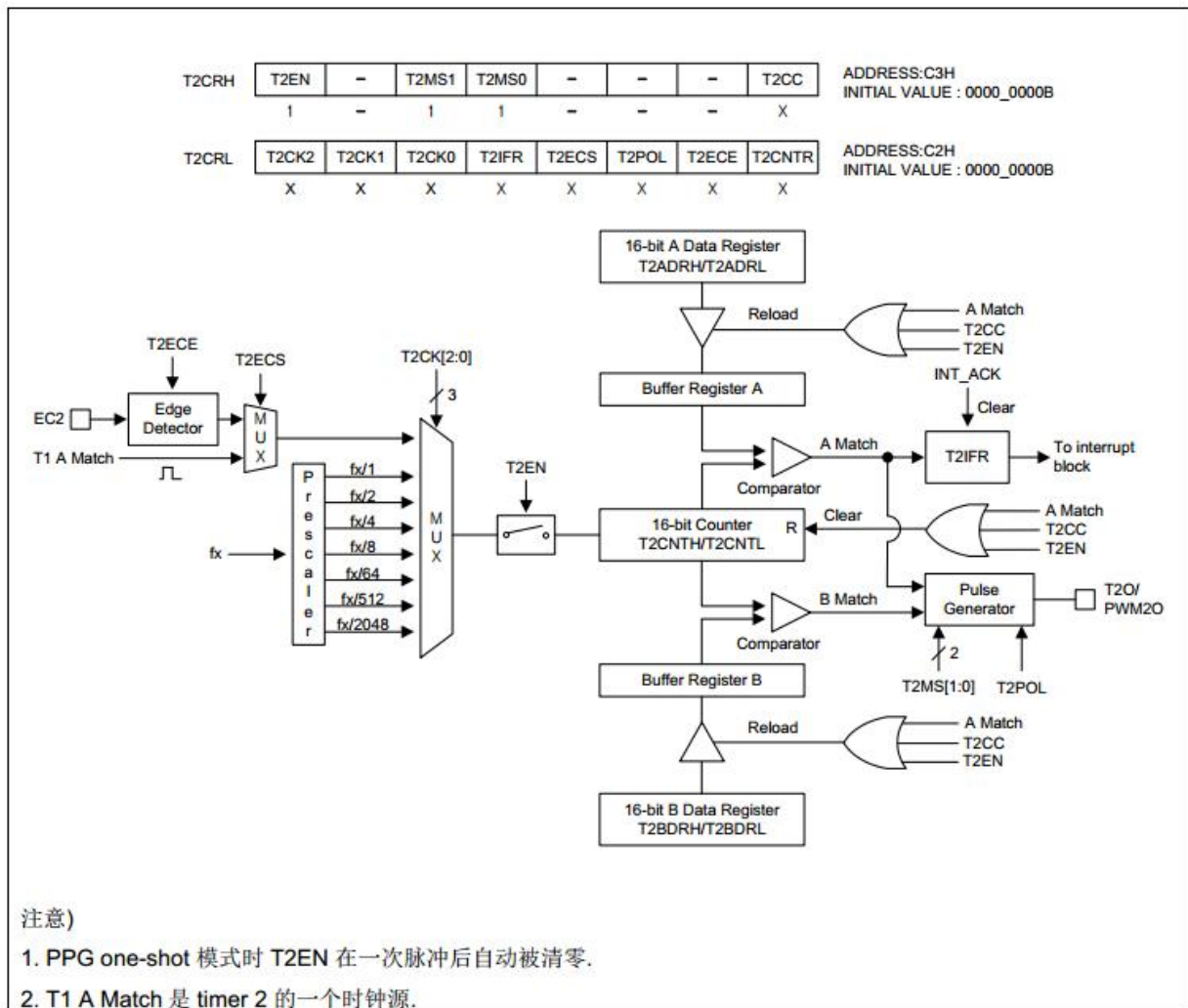
定时器 2 输入捕捉模式操作



捕捉模式定时器溢出

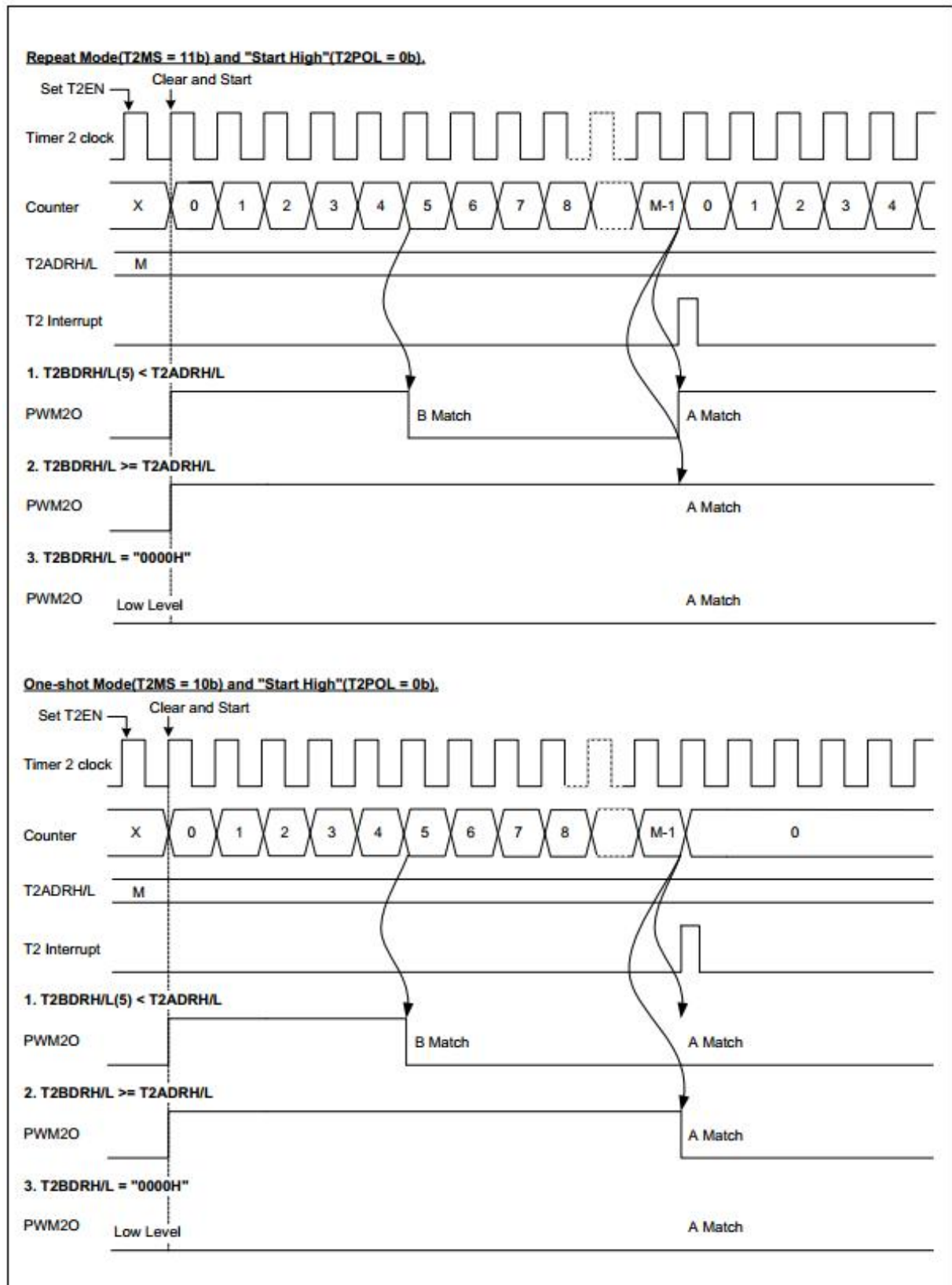
### 7.7.4、16-Bit PPG 模式

定时器 2 有 PPG(可编程脉冲发生)功能。PPG 模式时, T2O/PWM2O 引脚输出 16 位分辨率 PWM。该引脚应该通过将设置 P1FSRL[6:5] 为 '01' 将其配置为 PWM 模式。PWM 输出周期由 T2ADRH/T2ADRL 决定。占空比由 T2BDRH/T2BDRL 决定。



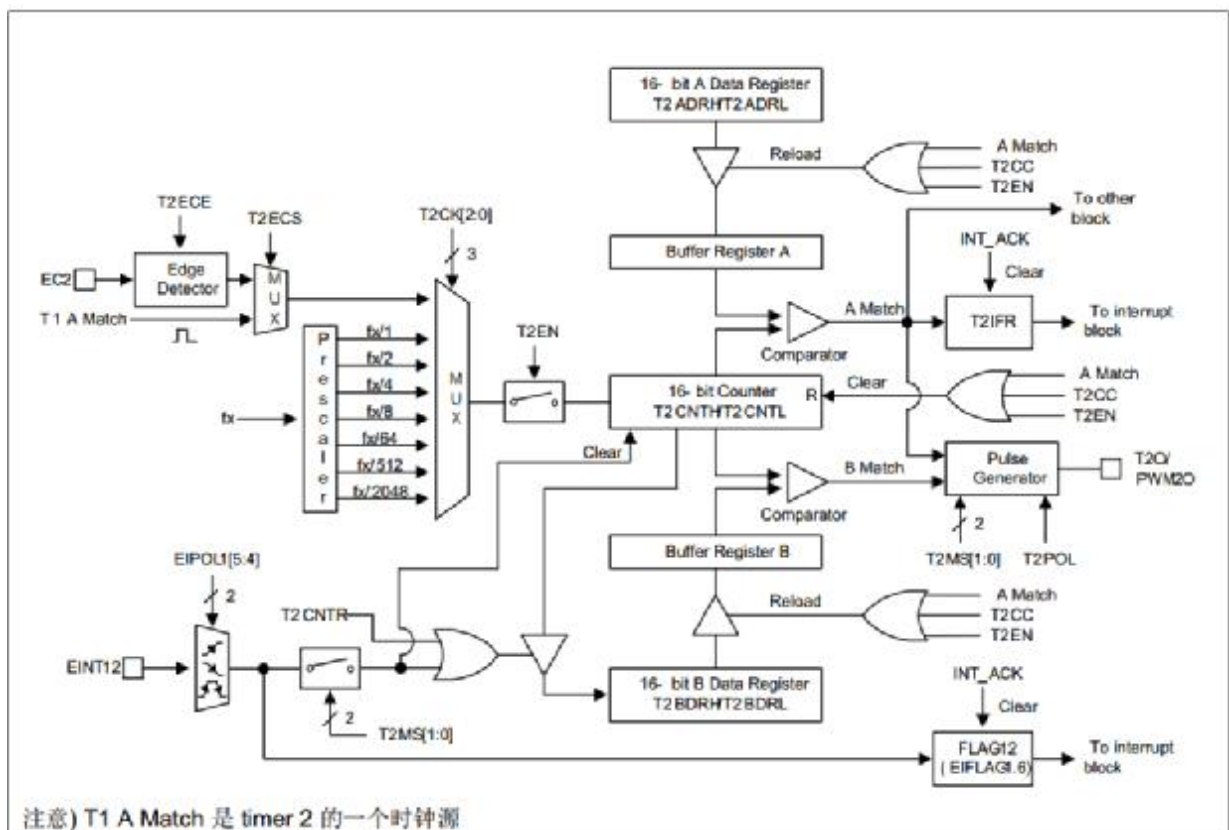
定时器 2 16-Bit PPG 模式





定时器 2 16-Bit PPG 模式时序图

7.7.5、方框图



16-Bit Timer2 方框图

7.7.6、寄存器图

Timer2 寄存器图

名字	地址	方式	默认	说明
T2ADRH	C5H	R/W	FFH	Timer 2 A Data High Register
T2ADRL	C4H	R/W	FFH	Timer 2 A Data Low Register
T2BDRH	C7H	R/W	FFH	Timer 2 B Data High Register
T2BDRL	C6H	R/W	FFH	Timer 2 B Data Low Register
T2CRH	C3H	R/W	00H	Timer 2 Control High Register
T2CRL	C2H	R/W	00H	Timer 2 Control Low Register

7.7.6.1、定时器/计数器 2 寄存器说明

定时器/计数器 2 寄存器由 T2ADRH, T2ADRL, T2BDRH, T2BDRL, T2CRH, T2CRL 组成。

7.7.6.2、定时器/计数器 2 寄存器说明

T2ADRH (Timer 2 A 数据高位寄存器) : C5H

7	6	5	4	3	2	1	0
T2ADRH7	T2ADRH6	T2ADRH5	T2ADRH4	T2ADRH3	T2ADRH2	T2ADRH1	T2ADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T2ADRH[7:0] T2A 数据高字节

**T2ADRL (Timer 2 A 数据低位寄存器) : C4H**

7	6	5	4	3	2	1	0
T2ADRL7	T2ADRL6	T2ADRL5	T2ADRL4	T2ADRL3	T2ADRL2	T2ADRL1	T2ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T2ADRL[7:0] T2A 数据低字节

注意) PPG 模式时 T2ADRH/T2ADRL 寄存器不要写入“0000H”

**T2BDRH (Timer 2 B 数据高位寄存器) : C7H**

7	6	5	4	3	2	1	0
T2BDRH7	T2BDRH6	T2BDRH5	T2BDRH4	T2BDRH3	T2BDRH2	T2BDRH1	T2BDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T2BDRH[7:0] T2B 数据高字节

**T2BDRL (Timer 2 B 数据低位寄存器) : C6H**

7	6	5	4	3	2	1	0
T2BDRL7	T2BDRL6	T2BDRL5	T2BDRL4	T2BDRL3	T2BDRL2	T2BDRL1	T2BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

T2BDRL[7:0] T2B 数据低字节

**T2CRH (Timer 2 控制高位寄存器) : C3H**

7	6	5	4	3	2	1	0
T2EN	-	T2MS1	T2MS0	-	-	-	T2CC
R/W	-	R/W	R/W	-	-	-	R/W

Initial value : 00H

T2EN 控制 Timer2

- 0 Timer 2 禁止
- 1 Timer 2 使能 (Counter clear and start)

T2MS[1:0] 控制 Timer 2 操作模式

T2MS1 T2MS0 说明

- 0 0 Timer/counter 模式 (T2O: toggle at A match)
- 0 1 捕捉模式 (The A match interrupt can occur)
- 1 0 PPG one-shot 模式(PWM2O)
- 1 1 PPG repeat mode (PWM2O)

T2CC 清除 Timer 2 计数

- 0 No effect
- 1 清除 Timer 2 计数 (写入时 清除计数后自动清零)

T2CRL (Timer 2 控制高位寄存器) : C2H

7	6	5	4	3	2	1	0
T2CK2	T2CK1	T2CK0	T2IFR	T2ECS	T2POL	T2ECE	T2CNTR
R/W	R/W	R/W	R/W	R/W	RW	RW	RW

Initial value : 00H

T2CK[2:0]	选择的定时器 2 时钟。fx 是系统主时钟频率
	T2CK2      T2CK1      T2CK0      说明
	0            0            0            fx/2048
	0            0            1            fx/512
	0            1            0            fx/64
	0            1            1            fx/8
	1            0            0            fx/4
	1            0            1            fx/2
	1            1            0            fx/1
	1            1            1            通过 T2ECS 位选择时钟
T2IFR	T2 匹配中断产生时置 ‘1’。写入 0 进行清零或者通过 INT_ACK 信号自动清零
	0            没有 T2 中断产生
	1            有 T2 中断产生
T2ECS	Timer2 外部时钟选择
	0            选择外部时钟(EC2)
	1            选择 Timer1A match
T2POL	T2O/PWM2O 极性选择
	0            高电平有效 (低电平 T2O/PWM2O 禁止)
	1            低电平有效 (低电平 T2O/PWM2O 禁止)
T2ECE	Timer2 外部时钟边沿触发选择
	0            下降沿触发
	1            上升沿触发
T2CNTR	Timer2 计数器读取控制
	0            无效
	1            加载计数值到 B 数据寄存器(写入时, 加载后自动清零)

## 7.8、蜂鸣器驱动

### 7.8.1、概述

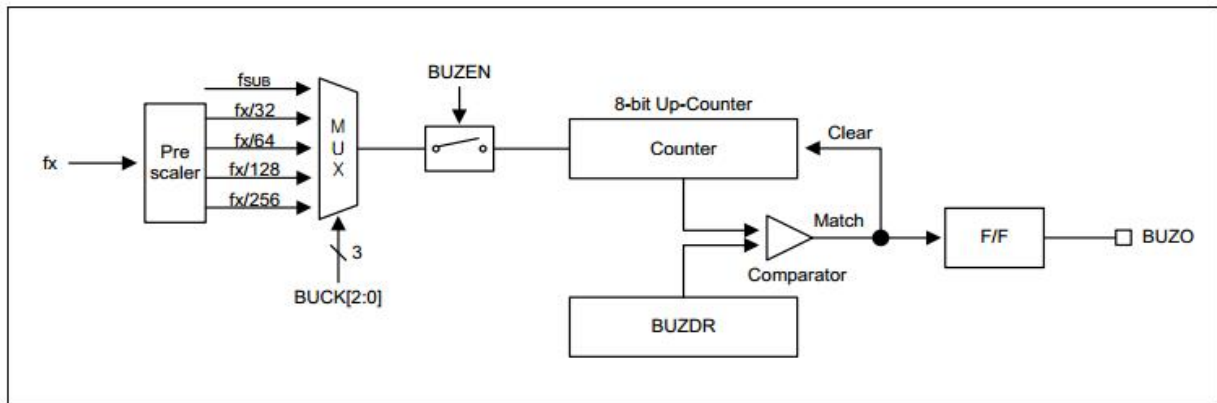
蜂鸣器包括 8 位计数器, 蜂鸣器数据寄存器(BUZDR), 和蜂鸣器控制寄存器(BUZCR)。方波 (61.035Hz~125.0kHz@8MHz)从 P11/BUZO 引脚输出。蜂鸣器数据寄存器(BUZDR)控制蜂鸣器频率(如下表所示)。在蜂鸣器控制寄存器(BUZCR)中, BUZCR[2:0]通过分频器选择时钟。

$$f_{BUZ}(\text{Hz}) = \frac{\text{Oscillator Frequency}}{2 \times \text{Prescaler Ratio} \times (\text{BUZDR} + 1)}$$

8MHz 时蜂鸣器时钟

BUZDR[7:0]	Buzzer Frequency (kHz)			
	BUZCR[3:1]=000	BUZCR[3:1]=001	BUZCR[3:1]=010	BUZCR[3:1]=011
0000_0000	125kHz	62.5kHz	31.25kHz	15.625kHz
0000_0001	62.5kHz	31.25kHz	15.625kHz	7.812kHz
...	...	...	...	...
1111_1101	492.126Hz	246.063Hz	123.031Hz	61.515Hz
1111_1110	490.196Hz	245.098Hz	122.549Hz	61.274Hz
1111_1111	488.281Hz	244.141Hz	122.07Hz	61.035Hz

### 7.8.2、方框图



蜂鸣器方框图

### 7.8.3、寄存器图

蜂鸣器驱动寄存器

Name	Address	Dir	Default	Description
BUZDR	8FH	R/W	FFH	蜂鸣器数据寄存器
BUZCR	97H	R/W	00H	蜂鸣器控制寄存器

### 7.8.4、蜂鸣器寄存器说明

蜂鸣器驱动由(BUZDR)和(BUZCR)组成。

### 7.8.5、蜂鸣器驱动寄存器说明

**BUZDR (蜂鸣器数据寄存器) : 8FH**

7	6	5	4	3	2	1	0
BUZDR7	BUZDR6	BUZDR5	BUZDR4	BUZDR3	BUZDR2	BUZDR1	BUZDR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : FFH

BUZDR[7:0] 该位控制蜂鸣器的频率  
蜂鸣器分辨率是 00H ~ FFH

**BUZCR (蜂鸣器控制寄存器) : 97H**

7	6	5	4	3	2	1	0
-	-	-	-	BUCK2	BUCK1	BUCK0	BUZEN
-	-	-	-	RW	RW	RW	RW

Initial value : 00H

BUCK[2:0] 选择蜂鸣器时钟

BUCK2	BUCK1	BUCK0	Description
0	0	0	fx/32
0	0	1	fx/64
0	1	0	fx/128
0	1	1	fx/256
1	x	x	fSUB (外部子时钟)

BUZEN 蜂鸣器操作控制

0	禁止
1	使能

注意) fx: 系统时钟振荡频率

**7.9、12-Bit A/D 转换器****7.9.1、概述**

A/D 可以将一个输入的模拟信号转换为相应的 12-bit 数字信号。A/D 模块有十五个模拟输入通道复用器的输出作为输入，通过逐次逼近法得到近似值。A/D 模块有 AD 控制高位寄存器(ADCCRH)，AD 控制地位寄存器(ADCCRL)，AD 高位数字寄存器(ADCDRH)，AD 低位数字寄存器(ADCRL)。ADSEL[3:0]的设置用来选择转换通道。执行 AD 操作时，TRIG[1:0]位须设置为 'xx'。ADCDRH 和 ADCRL 包含 AD 的转换结果。AD 转换完成，转换结果保存到 ADCDRH 和 ADCDRL，AD 转换状态位置 '1'，AD 中断产生。在 AD 转换中，AFLAG 位被读作 '0'。

**7.9.2、转换时序**

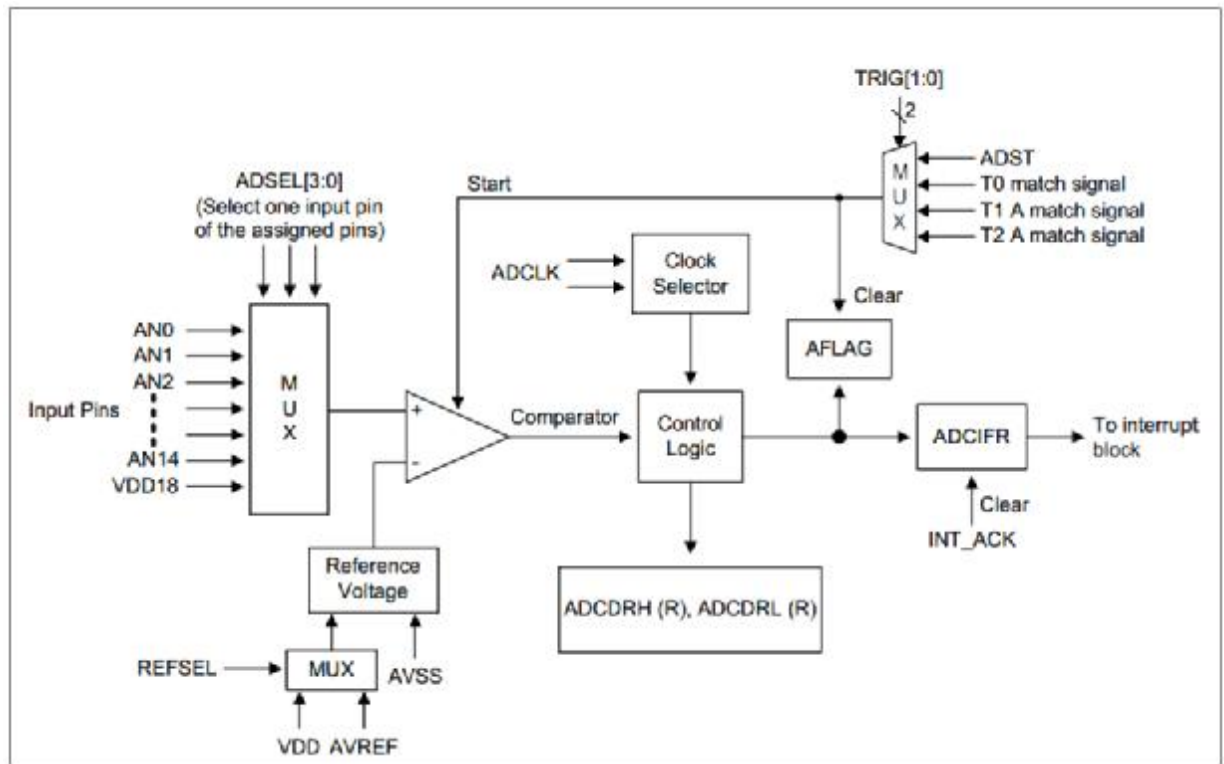
AD 转换需要 4 步(4 个时钟边沿)对每一位进行转换，10 个时钟设置 AD 转换器。所以完成 AD 转换需要 58 个时钟；在 12MHz 时钟频率下选择 fxx/8 分频时，一个时钟周期是 0.66 μs。每位转换需要 4 个时钟，转换率计算如下：

$$4 \text{ 时钟/bit} \times 12 \text{ bits} + \text{设置时间} = 58 \text{ 个时钟,}$$

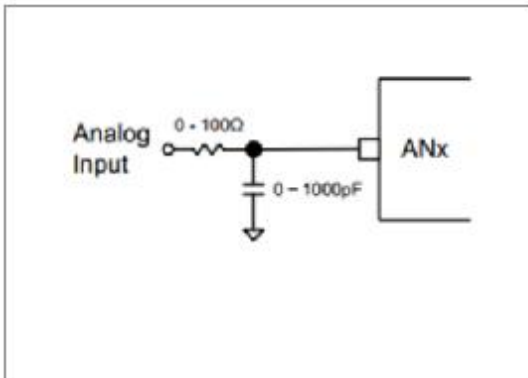
$$58 \text{ 个时钟} \times 0.66 \mu\text{s} = 38.28 \mu\text{s} \text{ 在 } 1.5 \text{ MHz 时}(12 \text{ MHz}/8)$$

注意) A/D 转换时间至少需要 20 μs。所以你必须设置转换时间大于 20 μs。

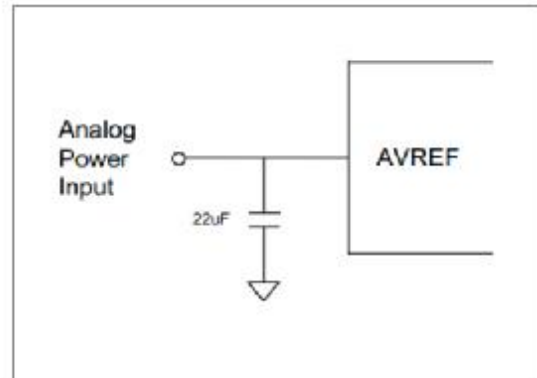
**7.9.3、方框图**



12-bit ADC 方框图

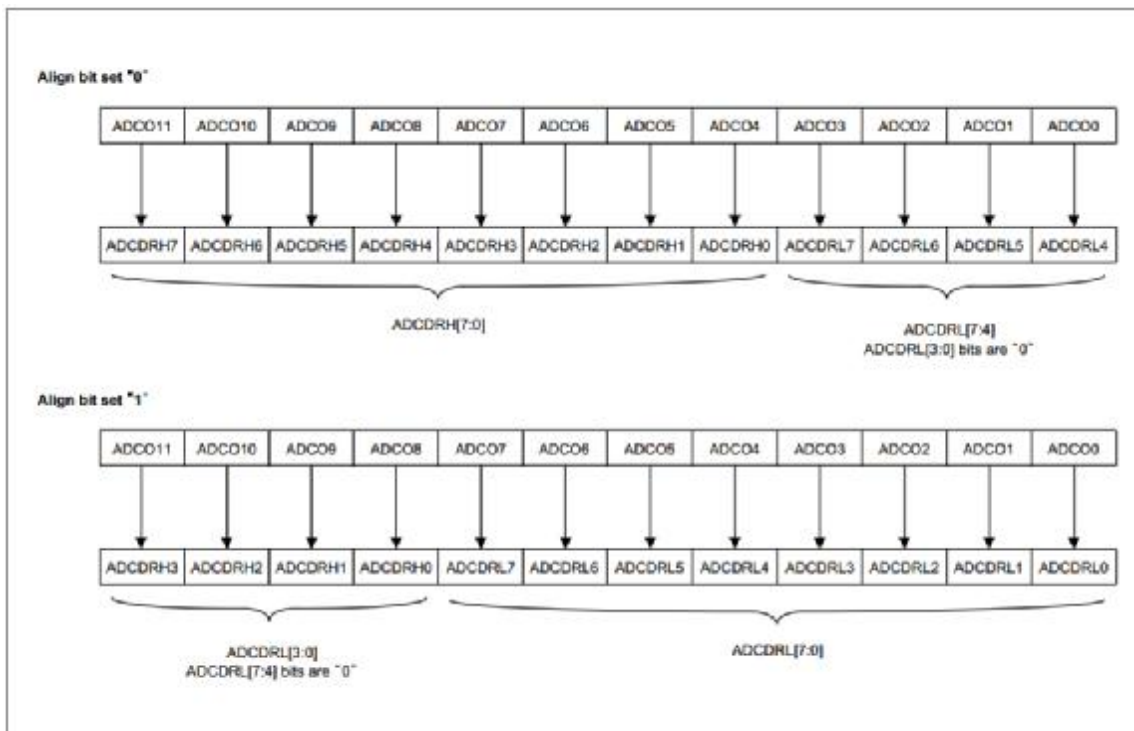


A/D 模拟输入脚电容

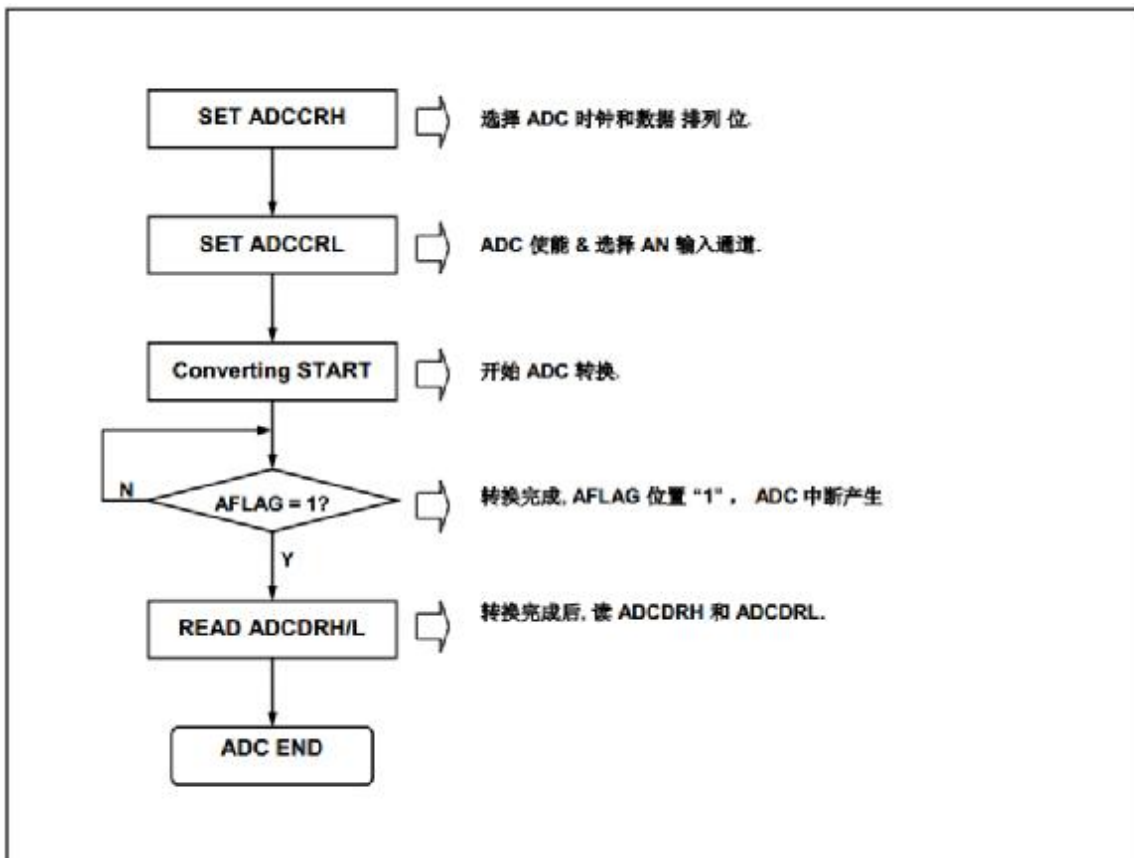


A/D Power (AVREF) 脚电容

### 7.9.4、ADC 操作



ADC 排列位的操作



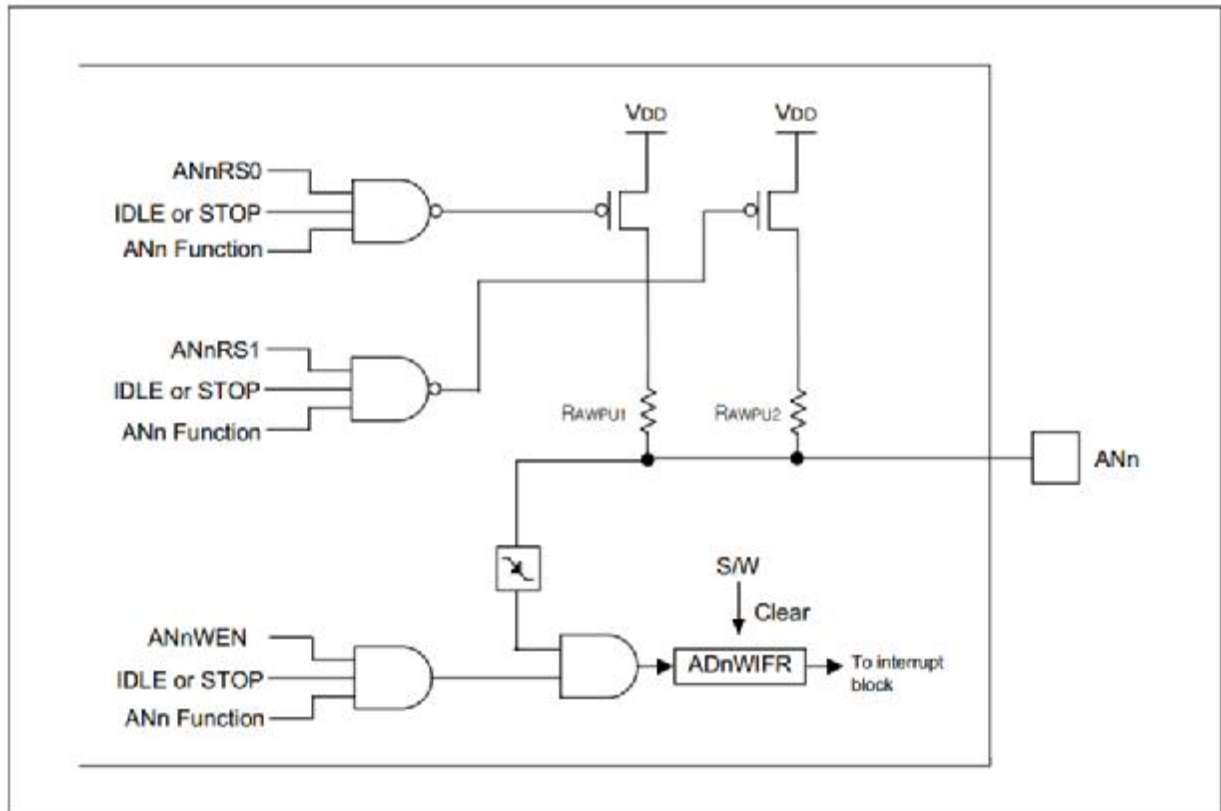
A/D 转换流程

7.9.5、ADC 省电唤醒功能



A/D 有省电唤醒功能。该功能包括从省电模式唤醒的俩个上拉电阻。ADC 在省电模式(IDLE,STOP)中如果 ANnRS[1:0]位不是“00b”，对应的上拉电阻是通过 P0FSR/P1FSRL/P1FSRH 寄存器的使能来选择”。如果 ANnWEN 是 ‘1’ 省电模式中 AD 唤醒中断可以通过键盘输入的下降沿(VIL)产生。这里 n = 0, 1, 2,,,,,,,,,,,,, 14。

### 7.9.6、ADC 省电唤醒功能方框图



ADC 省电唤醒方框图

- 注意)
1. AN0~AN14 功能可以通过 P0FSR,P1FSRL 和 P1FSRH 控制。
  2. P0/P1 上拉电阻可以通过 P0PU/P1PU 寄存器使能。所以，请注意 P0/P1 每一位上拉电阻。如果 P0/P1 中一个上拉电阻被使能，那么对应的引脚上将会改变相应的电阻值。
  3. ADC 唤醒中断可以通过选择 Ann 的下降沿产生。
  4. 省电模式和 ADC 通道是关闭的 ADC 唤醒中断是打开的如果 ANnWEN 置 ‘1’。
  5. 这里 n = 0, 1, 2,,,,,,,,,,,,, 14。

### 7.9.7、ADC 省电唤醒功能操作

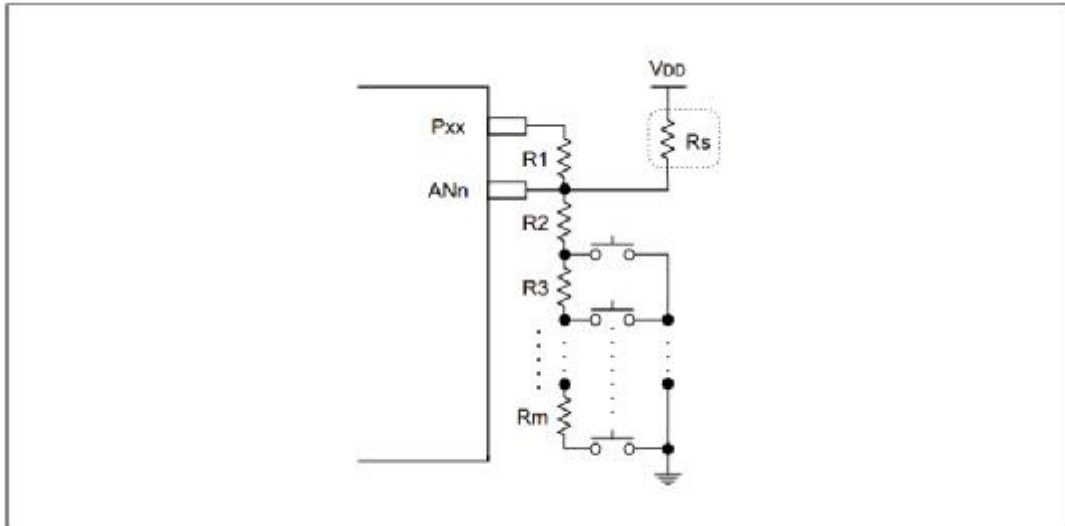
假如使用 AN0 作为省电唤醒功能并且上拉电阻为 150kΩ，下面是推荐的操作步骤。

1. P0FSR 位设置为 1 选择 AN0 通道。禁止 P00 上拉电阻(P00PU)。
2. 通过设置 AN0RS[1:0]中 ADWRCR0 为 ‘01b’ 使能 AN0 上拉 150kΩ。
3. 设置 ADWCRL 中的 AN0WEN 位为 1，使能 AN0 通道 ADC 唤醒中断。
4. 进入省电模式(idle or stop)。
5. 如果通过 AN0 产生按键输入下降沿，将会从省电模式产生一个 AD 唤醒中断请求。

6. 这时, AN0 唤醒中断标志位 ADWIFRL 中的 AN0WIFR 变为 1, 该位只有在手动写入 0 时才被清除。所以, 标志位需要软件清零。

### 7.9.8、ADC 按键输入串联电阻应用电路

ADC 省电唤醒功能需要接电阻。在 CPU 进入省电模式时可以阻止 ADC 输入脚的浮动。如果  $R_s \gg R_1, R_s/R_1$  等价的电阻由  $R_1$  决定。所以电阻  $R_s$  使用比  $R_1$  大的。



ADC 按键输入串联电阻应用电路

- 注意)
1. 推荐  $R_s$  使用  $1M\Omega$
  2. Pxx 是普通 I/O 口
  3. 这里  $n = 0, 1, 2, \dots, 14$

### 7.9.9、寄存器

ADC 寄存器

Name	Address	Dir	Default	Description
ADCDRH	9FH	R	xxH	A/D Converter Data High Register
ADCDRL	9EH	R	xxH	A/D Converter Data Low Register
ADCCRH	9DH	R/W	00H	A/D Converter Control High Register
ADCCRL	9CH	R/W	00H	A/D Converter Control Low Register
ADWRCR0	F2H	R/W	00H	ADC Wake-up Resistor Control Register 0
ADWRCR1	F3H	R/W	00H	ADC Wake-up Resistor Control Register 1
ADWRCR2	F4H	R/W	00H	ADC Wake-up Resistor Control Register 2
ADWRCR3	F5H	R/W	00H	ADC Wake-up Resistor Control Register 3
ADWCRH	F7H	R/W	00H	ADC Wake-up Control High Register
ADWCRL	F6H	R/W	00H	ADC Wake-up Control Low Register
ADWIFRH	DDH	R/W	00H	ADC Wake-up Interrupt Flag High Register
ADWIFRL	DCH	R/W	00H	ADC Wake-up Interrupt Flag Low Register

### 7.9.10、ADC 寄存器说明

ADC 寄存器由 A/D 转换高位数据寄存器(ADCDRH), A/D 转换低位数据寄存(ADCDRL), A/D 控制高位寄存(ADCCRH), A/D 计数低位寄存(ADCCRL), ADC 唤醒电阻控制寄存器 0(ADWRCR0),

ADC 唤醒电阻控制寄存器 1(ADWRCR1), ADC 唤醒电阻控制寄存器 2(ADWRCR2), ADC 唤醒电阻控制寄存器 3(ADWRCR3), ADC 唤醒控制高位寄存器(ADWCRH), ADC 唤醒控制低位寄存器(ADWCRL), ADC 唤醒中断标志位高位寄存器(ADWIFRH), 和 ADC 唤醒中断标志位高位寄存器(ADWIFRL)组成。

**ADCDRH (A/D 转换数据高位寄存器) : 9FH**

7	6	5	4	3	2	1	0
ADDM11	ADDM10	ADDM9	ADDM8	ADDM7 ADDL11	ADDM6 ADDL10	ADDM5 ADDL9	ADDM4 ADDL8
R	R	R	R	R	R	R	R

Initial value : xxH

ADDM[11:4] MSB align, A/D Converter High Data (8-bit)

ADDL[11:8] LSB align, A/D Converter High Data (4-bit)

**ADCDRL (A/D 转换数据低位寄存器) : 9EH**

7	6	5	4	3	2	1	0
ADDM8 ADDL7	ADDM2 ADDL6	ADDM1 ADDL5	ADDM0 ADDL4	ADDL3	ADDL2	ADDL1	ADDL0
R	R	R	R	R-	R	R	R

Initial value : xxH

ADDM[3:0] MSB align, A/D Converter Low Data (4-bit)

ADDL[7:0] LSB align, A/D Converter Low Data (8-bit)

**ADCCRH (A/D 转换高位寄存器) : 9DH**

7	6	5	4	3	2	1	0
ADCIFR	-	-	TRIG1	TRIG0	ALIGN	CKSEL1	CKSEL0
RW	-	-	RW	RW	RW	RW	RW

Initial value : 00H

ADCIFR ADC 中断产生时, 该位置 ‘1’。写入 0 进行清零或由 INT\_ACK 信号自动清零

0 没有 ADC 中断产生

1 有 ADC 中断产生

TRIG[1:0] A/D 触发信号选择 (睡眠模式时 ADC 模块自动禁止)

TRIG1	TRIG0	Description
-------	-------	-------------

0	0	ADST
---	---	------

0	1	Timer0 匹配信号
---	---	-------------

1	0	Timer1A 匹配信号
---	---	--------------

1	1	Timer2A 匹配信号
---	---	--------------

ALIGN A/D 转换数据排列选择.

0 MSB align (ADCDRH[7:0], ADCDRL[7:4])

1 LSB align (ADCDRH[3:0], ADCDRL[7:0])

CKSEL[1:0] A/D 转换时钟选择

CKSEL1	CKSEL0	Description
--------	--------	-------------

0	0	fx/1
---	---	------

0	1	fx/2
---	---	------

1	0	fx/4
---	---	------

1	1	fx/8
---	---	------

**ADCCRL (A/D 转换计数低位寄存器) : 9CH**

7	6	5	4	3	2	1	0
STBY	ADST	REFSEL	AFLAG	ADSEL3	ADSEL2	ADSEL1	ADSEL0
RW	RW	RW	R	RW	RW	RW	RW

Initial value : 00H

- STBY**            控制操作 A/D  
 (睡眠模式时 ADC 模块自动禁止)  
 0            ADC 模块禁止  
 1            ADC 模块使能
- ADST**            控制 A/D 转换开始.  
 0            No effect  
 1            转换开始触发信号
- REFSEL**        A/D 转换参考电压选择  
 0            内部参考电压 (VDD)  
 1            外部参考电压 (AVREF)
- AFLAG**        A/D 转换操作状态(STBY 位置 ‘0’ 或 CPU 在 STOP 模式时该位清零)  
 0            A/D 转换中  
 1            A/D 转换结束

**ADSEL[3:0]**    A/D 转换输入选择

ADSEL3	ADSEL2	ADSEL1	ADSEL0	Description
0	0	0	0	AN0
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7
1	0	0	0	AN8
1	0	0	1	AN9
1	0	1	0	AN10
1	0	1	1	AN11
1	1	0	0	AN12
1	1	0	1	AN13
1	1	1	0	AN14
1	1	1	1	VDD18

**ADWRCR0 (ADC 唤醒电阻控制寄存器 0): F2H**

7	6	5	4	3	2	1	0
AN3RS1	AN3RS0	AN2RS1	AN2RS0	AN1RS1	AN1RS0	AN0RS1	AN0RS0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

ADWRCR0[7:0] AN<sub>n</sub> 输入引脚 ADC 唤醒电阻选择

AN <sub>n</sub> RS[1:0]	300kΩ Resistor	150kΩ Resistor
0 0	禁止	禁止
0 1	禁止	使能
1 0	使能	禁止
1 1	使能	使能

n = 0, 1, 2, 3

**ADWRCR1 (ADC 唤醒电阻控制寄存器 1): F3H**

7	6	5	4	3	2	1	0
AN7RS1	AN7RS0	AN6RS1	AN6RS0	AN5RS1	AN5RS0	AN4RS1	AN4RS0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

ADWRCR1[7:0] AN<sub>n</sub> 输入引脚 ADC 唤醒电阻选择

AN <sub>n</sub> RS[1:0]	300kΩ Resistor	150kΩ Resistor
0 0	禁止	禁止
0 1	禁止	使能
1 0	使能	禁止
1 1	使能	使能

Where n = 4, 5, 6, and 7

**ADWRCR2 (ADC 唤醒电阻控制寄存器 2): F4H**

7	6	5	4	3	2	1	0
AN11RS1	AN11RS0	AN10RS1	AN10RS0	AN9RS1	AN9RS0	AN8RS1	AN8RS0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

ADWRCR2[7:0] AN<sub>n</sub> 输入引脚 ADC 唤醒电阻选择

AN <sub>n</sub> RS[1:0]	300kΩ Resistor	150kΩ Resistor
0 0	禁止	禁止
0 1	禁止	使能
1 0	使能	禁止
1 1	使能	使能

n = 8, 9, 10, 11

**ADWRCR3 (ADC 唤醒电阻控制寄存器 3): F5H**

7	6	5	4	3	2	1	0
-	-	AN14RS1	AN14RS0	AN13RS1	AN13RS0	AN12RS1	AN12RS0
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

ADWRCR3[5:0] ANn 输入引脚 ADC 唤醒电阻选择

ANnRS[1:0] 300kΩ Resistor 150kΩ Resistor

0	0	禁止	禁止
0	1	禁止	使能
1	0	使能	禁止
1	1	使能	使能

n = 12, 13, 14

**ADWCRH (ADC 唤醒控制高位寄存器): F7H**

7	6	5	4	3	2	1	0
-	AN14WEN	AN13WEN	AN12WEN	AN11WEN	AN10WEN	AN9WEN	AN8WEN
-	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

ADWCRH[6:0] ANn 输入使能或禁止 ADC 唤醒功能

ANnWEN Description

0	禁止
1	使能

n = 8, 9, 10, 11, 12, 13, 14

**ADWCRL (ADC 唤醒控制低位寄存器): F6H**

7	6	5	4	3	2	1	0
AN7WEN	AN6WEN	AN5WEN	AN4WEN	AN3WEN	AN2WEN	AN1WEN	AN0WEN
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

ADWCRL[7:0] ANn 输入使能或禁止 ADC 唤醒功能

ANnWEN Description

0	禁止
1	使能

n = 0, 1, 2, 3, 4, 5, 6, 7

**ADWIFRH (ADC 唤醒中断标志位高位寄存器): DDH**

7	6	5	4	3	2	1	0
-	AN14WIFR	AN13WIFR	AN12WIFR	AN11WIFR	AN10WIFR	AN9WIFR	AN8WIFR
-	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

ADWIFRH[6:0] 当 AN14 ~ AN8 一个 ADC 唤醒中断产生时置 ‘1’。该位写入 0 清除，所以该标志位需要软件清除。

0	AN14 ~ AN8 通道没有 ADC 唤醒中断产生
1	AN14 ~ AN8 通道有 ADC 唤醒中断产生

**ADWIFRL (ADC 唤醒中断标志位低位寄存器): DCH**

7	6	5	4	3	2	1	0
AN7WIFR	AN6WIFR	AN5WIFR	AN4WIFR	AN3WIFR	AN2WIFR	AN1WIFR	AN0WIFR
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

ADWIFRL[7:0] 当 AN7 ~ AN0 一个 ADC 唤醒中断产生时置 ‘1’。该位写入 0 清除，所以该标志位需要软件清除

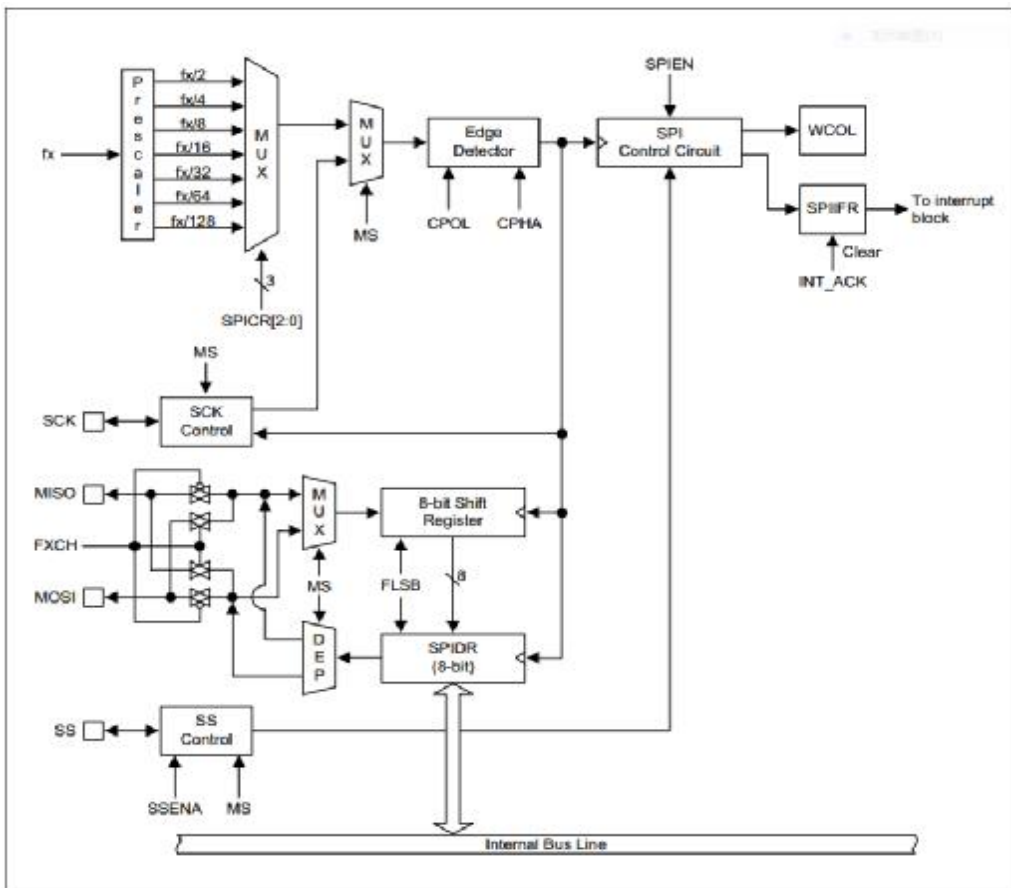
- 0 AN7 ~ AN0 通道没有 ADC 唤醒中断产生
- 1 AN7 ~ AN0 通道有 ADC 唤醒中断产生

**7.10、SPI**

**7.10.1、概述**

AiP81F2016 配置有串行外围接口 (SPI)。SPI 允许外围串行设备之间同步串行数据传送。通过 4 跟线(MOSI, MISO, SCK, SS)可以全双工通信，支持主机和从机模式，可选择串行时钟极性，相位和低位数据优先传送和高位数据优先传送。

**7.10.2、方框图**



SPI 方框图

**7.10.3、数据传送/接收操作**

---

用户可以通过以下步骤使用 SPI 串口数据发送

1. 通过 SPICR 控制寄存器选择 SPI 操作模式(主动/从动, 极性, 相位)。
2. 当 SPI 配置为主机模式时, 通过 SS 信号选择从机(低电平有效)。当 SPI 配置为从机模式时, 被从主机过来的 SS 信号选择。
3. 用户写一个字节到数据寄存器 SPIDR 时, SPI 将开始一个操作。
4. 这次, 如果 SPI 配置为主机, 串行时钟将由 SCK 引脚产生。主机发送 8 位到从机(发送), 同时从机发送 8 位到主机(接收)。如果是从机模式, 串行时钟将有 SCK 引脚产生, 从机发送 8 位到主机(发送), 同时主机发送 8 位到从机(接收)。
5. 当执行发送/接收操作时, SPIIFR 位将会置位。如果 SPI 中断被使能, 产生一个中断请求。执行相应的中断时 SPIIFR 位通过硬件清零。如果 SPI 中断禁止, 当用户读取 SPISR 状态寄存器时 SPIIFR 位被清零, 然后进入(读/写)数据寄存器 SPIDR。

#### 7.10.4、SS 引脚功能

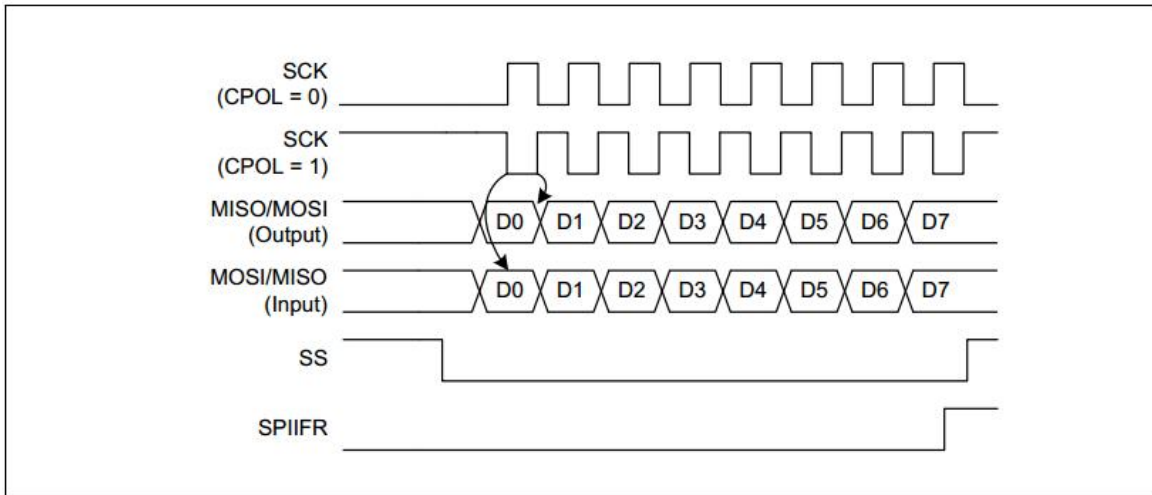
1. 当 SPI 设置为从机模式时, SS 引脚通常是输入状态。如果低电平信号进入到 SS 引脚, SPI 逻辑运行。如果高电平信号进入到 SS 引脚, SPI 逻辑停止。这时, SPI 逻辑将会复位, 接收不到数据。
2. 当 SPI 设置为主机模式时, 用户可以通过 P17IO 选择 SS 引脚方向。如果 SS 引脚配置为输出, 用户可以将其设为输出模式。如果 SS 设置为输入模式, 高电平信号必须进入到 SS 引脚以保障主机操作。如果低电平信号进入到 SS 引脚, SPI 逻辑认为另一主控器选择作为从动, 且开始向它发送数据。为避免总线争用, SPICR 中的 MSB 位必须清除同时 SPI 变为从机, SPISR 中的 SPIIFR 位置位, 如果 SPI 中断被使能, 将会产生一个中断请求。

注意)

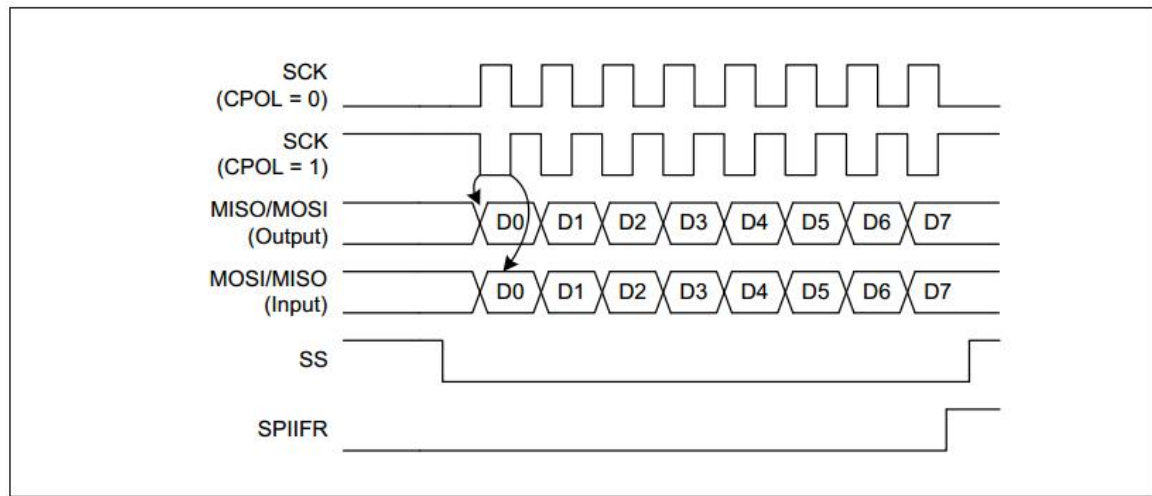
- 当 SS 引脚配置为主机模式时, SS 引脚的输出值由用户软件(P17IO)来规定。在 SPICR 设置前, SS 引脚的方向必须先确定。
- 如果你不需要用 SS 引脚, 清除 SPISR 中的 SENA 位。所以, 你可以通过 P17IO 自由使用其开关。此时, SS 信号由内部高低电平驱动, 从机时低电平。
- 当 SS 引脚作为输入时, 如果高电平信号进入到 SS 引脚, SS\_HIGH 标志位将会置位。你可以手动写入 0 进行清除。

#### 7.10.5、SPI 时序图





SPI 发送/接收时序图，当 CPHA = 0 时



SPI 发送/接收时序图，当 CPHA = 1 时

7.10.6、寄存器图

SPI 寄存器图

Name	Address	Dir	Default	Description
SPISR	B7H	R/W	00H	SPI Status Register
SPIDR	B6H	R/W	00H	SPI Data Register
SPICR	B5H	R/W	00H	SPI Control Register

7.10.7、SPI 寄存器说明

SPI 寄存器由 SPI 控制寄存器(SPICR), SPI 状态寄存器(SPISR)和 SPI 数据寄存器(SPIDR)组成。

7.10.8、SPI 寄存器说明

**SPIDR (SPI 数据寄存器) : B6H**

7	6	5	4	3	2	1	0
SPIDR7	SPIDR6	SPIDR5	SPIDR4	SPIDR3	SPIDR2	SPIDR1	SPIDR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

SPIDR [7:0] SPI 数据  
当写入一个字节到数据寄存器时, SPI 开始一个操作

**SPISR (SPI 状态寄存器) : B7H**

7	6	5	4	3	2	1	0
SPIIFR	WCOL	SS_HIGH	-	FXCH	SSENA	-	-
RW	R	RW	-	RW	RW	-	-

Initial value : 00H

SPIIFR 当 SPI 中断产生时置位, 如果 SPI 中断使能, 该位将被 INT\_ACK 信号自动清零。如果 SPI 中断禁止, 当读取 SPISR 时该位清零, 同时进入(read/write)数据寄存器 SPIDR

- 0 没有 SPI 中断产生
- 1 有 SPI 中断产生

WCOL 在发送过程中如果有数据写入到 SPIDR 时该位置位。读取 SPISR 时该位清除, 同时进入(read/write)数据寄存器 SPIDR

- 0 没有冲突
- 1 冲突

SS\_HIGH 当 SS 引脚设置为输入时, 如果高电平信号进入该引脚, 该位置位。

- 0 写入 '0' 时清零
- 1 当写入 '1' 时没有作用

FXCH SPI 口功能控制位。

- 0 No effect
- 1 更换 MOSI 和 MISO 功能

SSENA 该位控制 SS 引脚操作

- 0 禁止
- 2 打开(P17 应该设置为支持输入口)

**SPICR (SPI 控制寄存器) : B5H**

7	6	5	4	3	2	1	0
SPIEN	FLSB	MS	CPOL	CPHA	DSCR	SCR1	SCR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

SPIEN 控制 SPI 操作  
0 禁止 SPI 操作  
1 打开 SPI 操作

FLSB 该位选择数据传送顺序  
0 MSB 高位优先  
1 LSB 低位优先

MS 选择主机/从机模式

	0	从机模式		
	1	主机模式		
CPOL	这两位控制串口时钟模式(SCK)			
CPHA	时钟极性位(CPOL)决定 SCK 的值在空闲模式			
	时钟相位位(CPHA)决定数据采样在 SCK 的 leading 或 trailing 边沿			
	CPOL	CPHA	Leading edge	Trailing edge
	0	0	Sample (Rising)	Setup (Falling)
	0	1	Setup (Rising)	Sample (Falling)
	1	0	Sample (Falling)	Setup (Rising)
	1	1	Setup (Falling)	Sample (Rising)
DSCR	选择主机模式时 SCK 分频。当 DSCR 位写入 1,SCK 将会倍频在主机模式			
SCR[2:0]	DSCR	SCR1	SCR0	SCK 频率
	0	0	0	fx/4
	0	0	1	fx/16
	0	1	0	fx/64
	0	1	1	fx/128
	1	0	0	fx/2
	1	0	1	fx/8
	1	1	0	fx/32
	1	1	1	fx/64

## 7.11、SPI

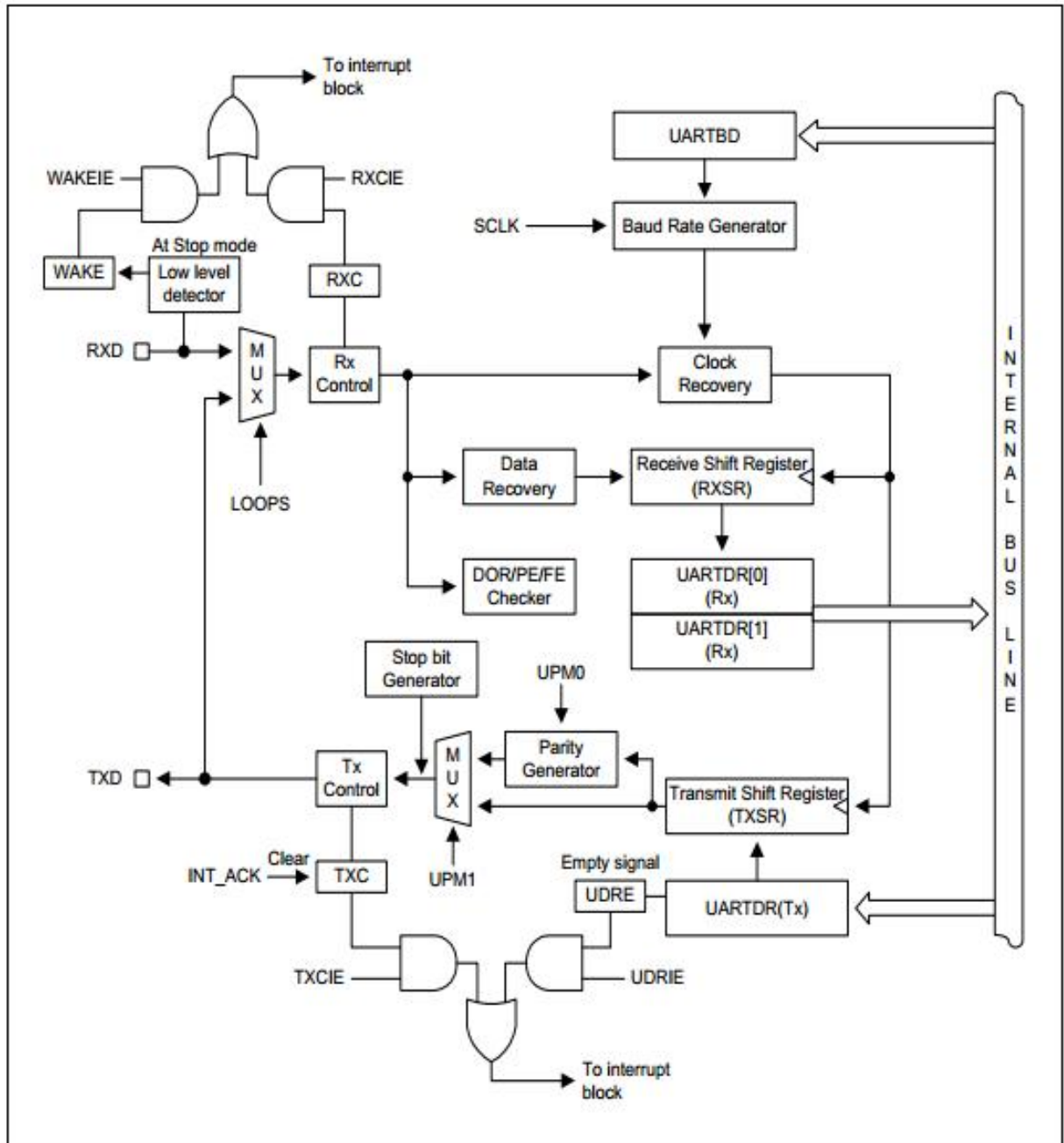
### 7.11.1、概述

通用异步串行接收发送是高度灵活的串行通信设备。主要特性如下：

- 同时双向操作 (独立串行接收和发送寄存器)
- 波特率发生器
- 支持串口帧数为 5, 6, 7, 8 或 9 的数据位和 1 或 2 个停止位
- 硬件支持奇偶校验的产生和奇偶检查
- 数据超出检测
- 帧差错检测
- 三个单独的中断, TX 完成, TX 数据寄存器空和 RX 完成

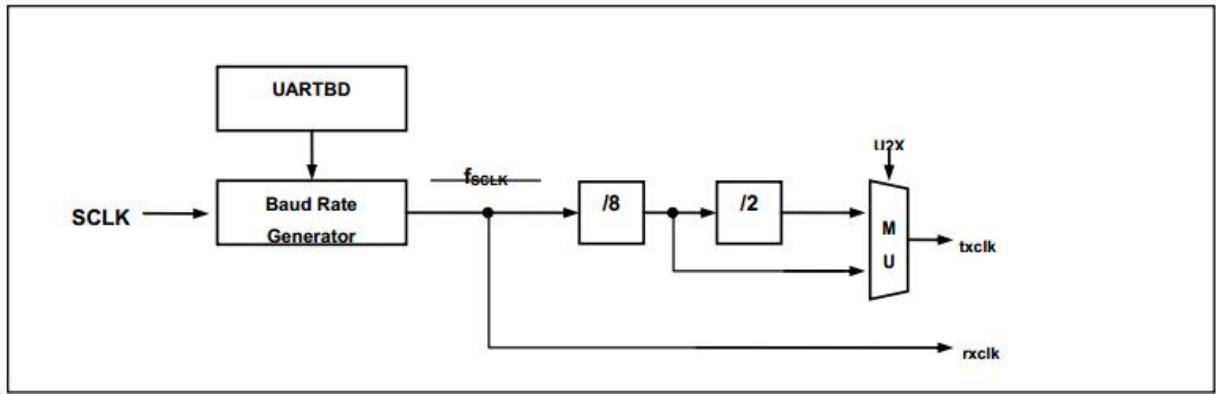
UART 有波特率发生器, 发送器和接收器。波特率发生器采用异步操作。发送器包括写信号缓存器, 一个串口移位寄存器, 奇偶发生器和操作不同帧格式的控制逻辑。写入暂存器允许两帧数据之间没有任何延时的连续发送。接收器的时钟和数据接收装置是 UART 模块最复杂的部分, 接收装置用作异步数据接收。除了接收装置以外, 接收器还包括一个奇偶校验器, 一个移位寄存器, 一个 two-level receive FIFO (UARTDR) 和控制逻辑。接收器支持和发送器时相同的帧格式, 并可以检测到帧差错, 数据超出和奇偶错误。

### 7.11.2、方框图



UART 方框图

### 7.11.3、时钟发生器



时钟发生器方框图

时钟脉冲振荡为发送和接收产生基础时钟

下表所示为波特率的计算公式(in bps)

波特率设置计算公式

操作模式	Equation for Calculating Baud Rate
Normal Mode(U2X=0)	$\text{Baud Rate} = \frac{fx}{16(\text{UARTBD} + 1)}$
Double Speed Mode(U2X=1)	$\text{Baud Rate} = \frac{fx}{8(\text{UARTBD} + 1)}$

#### 7.11.4、数据格式

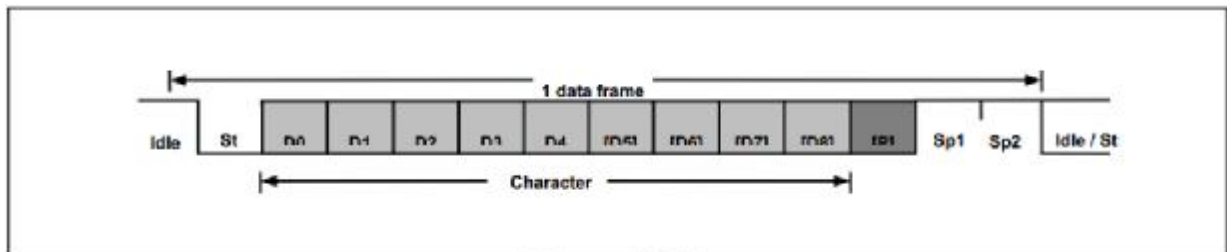
串行格式定义为同步位数据位中的字符，和随机的奇偶错误校验。

UART 支持以下全部 30 个数据组合中的任何一个有效的格式。

- 1 个开始位
- 5, 6, 7, 8 或 9 个数据位
- 无，数或偶数校验位
- 1 或 2 停止位

一帧数据开始于紧邻着最低有效位的开始位(LSB)。后是下一个数据位，直到第 9 位，以最高有效位结束。如果校验功能使能，校验位插入到最后一个数据位和停止位之间。数据引脚上从低到高的传送被视为开始位。当一帧完整的数据传送完成，下一帧数据会直接跟随其后。或者通信数据线可以设置为休眠状态。休眠意思是数据脚高电平。

下图所示为可能用到的数据格式。里边的位数可以选择。



帧格式

1 帧数据包括以下位：

- Idle 在通讯线上没有通讯(TxD/RxD)

- St 开始位(Low)
- Dn 数据位(0~8)
- Parity bit ----- 奇、偶校验, 无校验
- Stop bit(s) ----- 1 bit 或 2 bits

UART 帧格式通过 UARTCR1 和 UARTCR3 寄存器中的 USIZE[2:0],UPM[1:0]和 USBS 位来设置。发送和接收使用相同的设置。

#### 7.11.5、校验位

校验位是所有数据位通过异或计算得到。如果使用奇校验, 异或的结果是反向的。校验位插入到 MSB 和第一个停止位。

$$Peven = D_{n-1} \wedge \dots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$$

$$Podd = D_{n-1} \wedge \dots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$$

Peven: 校验位使用偶校验

Podd: 校验位采用奇校验

Dn: 数据位

#### 7.11.6、UART 发送

UART 发送的使能是通过设置 UARTCR2 寄存器中的 TXE 位。当发送使能时, TXD 引脚应该通过 P3FSR[1:0]设置为 TXD 功能, 波特率, 操作模式和帧格式必须在发送之前一次设置完。

##### 7.11.6.1、发送 Tx 数据

数据传输通过加载所要发送的数据到发送缓存器(UARTDR)开始。当移位寄存器准备好发送新的一帧数据时, 写入到发送缓存器中的数据移位到移位寄存器中。移位寄存器在空闲状态或在发送的最后一个停止位后立即加载新的数据。当移位寄存器加载新的数据时, 会根据控制寄存器设置发送一帧完整的数据。如果这 9 位已经被用, 第九位必须在它加载到发送缓存器(UARTDR)之前写到 UARTCR3 寄存器中的 TX8 位。

##### 7.11.6.2、传送标志位和中断

UART 发送器有 2 个标志位, 它表明了其状态。一个是 UAR 数据寄存器空标志(UDRE)另一个是传送完成标志位(TXC)。俩位都可以作为中断源。

UDRE 表示传送暂存器已经准备去接收新的数据。当发送暂存器清空之后该位置位, 当发送暂存器包含被发送但是还没有移位到移位寄存器的数据时该位清零。该标志位也可以在该位置写入 0 进行清除。写入 1 是不允许的。

当 UARTCR2 寄存器中的数据寄存器清空中断(UDRIE)使能, 同时总中断也打开, UDRE 置位时 UART 数据寄存器清空中断会产生中断。

当发送移位寄存器中的整帧数据已经移出并且没有数据在缓存器中时 TXC 标志位置位。TXC 标志位自动清除当发送完成中断服务程序执行时, 或是你可以在 UARTST 寄存器的 TXC 位写入 0 进行清零。

当 UARTCR2 寄存器中的发送完成中断使能位(TXCIE)置位并且总中断打开, 当 TXC 位置位时 UART 送完成中断会产生中断。

##### 7.11.6.3、奇偶校验发生器

奇偶校验发生器对发出的串行数据进行奇偶位计算。当奇偶校验使能时(UPM[1]=1)，发送器控制逻辑将其插入到 MSB 和帧数据的第一个停止位之间。

#### 7.11.6.4、禁止发送

通过清空 TXE 位禁止发送不会生效直到正在发送的数据发送完整。当发送禁止时，TXD 引脚可以作为普通 I/O 口(GPIO)。

#### 7.11.7、UART 接收

UART 通过设置 UARTCR2 寄存器的 RXE 位来使能。当其使能时，RXD 脚应该设为输入口。波特率，操作模式和帧格式必须在接收之前设置完成。

##### 7.11.7.1、接收 R 数据

当在 RXD 上检测到开始位(LOW)时接收器开始数据接收。开始位后的每一位由预定义的波特率和移位到移位寄存器开始采样直到接收到第一位停止位。即使一帧中有第二个停止位，第二个停止位也被接收器忽略。也就是说，接收到第一个停止位意味着一帧完整的数据存在于接收移位寄存器，同时移位寄存器的内容转移到接收暂存器。接收暂存器通过 UARTDR 寄存器读取。

如果 9 位字符被用(USIZE[2:0] = “111”)，第九位存储到 UARTCR3 寄存器 RX8 的位置。第 9 位必须在从 UARTDR 读取低 8 位之前通过 RX8 读取。同样的，错误标志位 FE，DOR，PE 必须在从 UARTDR 取数据之前读取。因为错误标志位是存储在接收暂存器中和 FIFO 相同的位置。

##### 7.11.7.2、接收标志位和中断

UART 接收器有一个标志位用来表明接收器的状态。

接收完成(RXC)标志位表示在接收缓存器中是否有未读取的数据。当在接收缓存器有未读取的数据置位，当接收缓存器清空时该位清零。如果接收器禁止(RXE=0)，接收缓存器被清空并且 RXC 清零。

当 UARTCR2 寄存器中的接收完成中断使能位(RXCIE)置位同时总中断打开，RXC 标志位置位时 UART 接收完成中断产生中断。

UART 接收器有三个错误标志位，分别是帧错误(FE)，数据超出(DOR)和奇偶校验错误(PE)。这些错误标志位可以通过 UARTST 寄存器读取。接收数据存储在 2 级接收暂存器，这些标志位也存储在接收缓冲区相同的位置。所以，在从 UARTDR 寄存器读取接收数据前，先读取包含错误标志位的 UARTST 寄存器。

帧格式错误(FE)标志位表示第一个停止位的状态。当停止位正确的检测到‘1’时，FE 是‘0’。当停止位错误，也就是检测到 0 时，FE 是‘1’。该位可以用检测帧与帧之间数据失去同步。

数据超出(DOR)标志位表明由于接收缓冲区装满而引起的数据丢失。当接收缓存器装满，同时另一个已经存储到接收缓存器的新数据又存在于接收移位寄存器时，DOR 产生。当 DOR 标志位置位后，所有进来的数据都会丢失。避免数据的丢失或清除该位，读取接收缓存器。

奇偶校验错误(PE)标志位表明在接收暂存器接收的数据中已经有一个校验错误。如果奇偶校验检测功能没有使能(UPM[1]=0)，PE 位通常是读‘0’。

##### 7.11.7.3、奇偶校验位检查器

如果奇偶校验位使能(UPM[1]=1)，奇偶校验检查器计算进来的数据的奇偶和接收到串口帧数据进行比较。

#### 7.11.7.4、禁止接收

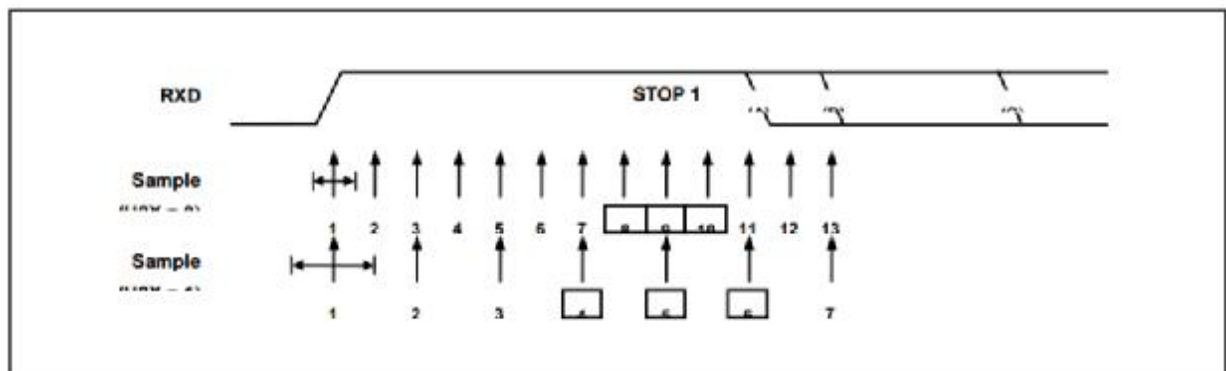
与发送形成对比，通过清零 RXE 位来禁止接收器使接收器立即执行。当接收器禁止时，接收器清空缓存器，在缓存器中剩下的数据全部复位，同时 RXD 引脚可以作为普通 I/O 口(GPIO)。

#### 7.11.7.5、异步数据接收

接收外部数据结构时，UART 包括一个时钟和数据恢复装置。时钟恢复逻辑用于将内部产生的波特率时钟和从 RXD 引脚引入的异步连续数据结构相同步。

数据恢复逻辑对引入位进行采样和低通滤波，这样可以消除 RXD 脚上的干扰。

下面举例说明对一个引入结构的开始位进行采样的过程。采样率为波特率正常模式(U2X=0)时 16 次，双速模式(U2X=1)时 8 次，水平箭头表示的是由于异步采样引起的同步变化，注意多数时间显示的都是双速模式。



Stop 位采样和下一个开始位采样

#### 7.11.8、寄存器图

UART 寄存器

Name	Address	Dir	Default	Description
UARTBD	E6H	R/W	FFH	UART Baud Rate Generation Register
UARTDR	E7H	R/W	00H	UART Data Register
UARTCR1	E2H	R/W	00H	UART Control Register 1
UARTCR2	E3H	R/W	00H	UART Control Register 2
UARTCR3	E4H	R/W	00H	UART Control Register 3
UARTST	E5H	R/W	80H	UART Status Register

#### 7.11.9、UART 寄存器说明

UART 模块由 UART 波特率发生器寄存器(UARTBD)，UART 数据寄存器(UARTDR)，UART 控制寄存器 1(UARTCR1)，UART 控制寄存器 2(UARTCR2)，UART 控制寄存器 3(UARTCR3)，和 UART 状态寄存器(UARTST)组成。

#### 7.11.10、UART 寄存器说明



**UARTBD (UART 波特率发生器寄存器) : E6H**

7	6	5	4	3	2	1	0
UARTBD7	UARTBD6	UARTBD5	UARTBD4	UARTBD3	UARTBD2	UARTBD1	UARTBD0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : FFH

UARTBD [7:0] 该寄存器的值用于产生内部波特率。为了避免故障，不要写入‘0’。

**UARTDR (UART 数据寄存器) : E7H**

7	6	5	4	3	2	1	0
UARTDR7	UARTDR6	UARTDR5	UARTDR4	UARTDR3	UARTDR2	UARTDR1	UARTDR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

UARTDR [7:0] 发送缓冲器和接收缓冲器和数据寄存器共用相同的 I/O 地址，发送数据缓冲器是写入到 UARTDR 寄存器的目标。读 UARTDR 寄存器时从接收缓冲器返回内容。只有在 UDRE 置位时可以写入。

**UARTCR1 (UART 控制寄存器 1) : E2H**

7	6	5	4	3	2	1	0
-	-	UPM1	UPM0	USIZE2	USIZE1	USIZE0	-
-	-	RW	RW	RW	RW	RW	-

Initial value : 00H

UPM[1:0] 选择奇偶校验和检查法

UPM1	UPM0	校验
0	0	没有校验
0	1	Reserved
1	0	偶同位
1	1	奇同位

USIZE[2:0] 选择帧数据长度

USIZE2	USIZE1	USIZE0	数据长度
0	0	0	5 bit
0	0	1	6 bit
0	1	0	7 bit
0	1	1	8 bit
1	1	1	9 bit

Other values Reserved

**UARTCR2 (UART 控制寄存器 2) : E3H**

7	6	5	4	3	2	1	0
UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	UARTEN	U2X
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

UDRIE UART 数据寄存器清空中断使能

0	UDRE 中断禁止(使用查询法)
1	当 UDRE 置位时，请求中断

TXCIE 发送完成中断使能位

	0	TXC 中断禁止(使用查询法)
	1	当 TXC 置位时, 请求中断
RXCIE		接收完成中断使能位
	0	RXC 中断禁止 (使用查询法)
	1	RXC 置位时, 请求中断
WAKEIE		STOP 模式唤醒中断, 当设备在睡眠模式时, 如果 RXD 变为低电平, 将会产生中断请求去唤醒系统。那时 UDRIE 和 UARTST 的值应该分别设置为 ‘0b’ 和 ‘00H’
	0	唤醒中断被禁止
	1	当 WAKE 置位时, 请求中断
TXE		发送使能位
	0	发送禁止
	1	发送使能
RXE		接收使能位
	0	接收禁止
	1	接收使能
UARTEN		激活 UART 模块。当 TXE 和 RXE 中有一个值为“1”, UARTEN 位通常设置为“1”
	0	UART 模块禁止(clock is halted)
	1	UART 模块使能
U2X		选择接收采样率.
	0	常规异步操作
	1	双速异步操作

**UARTCR3 (UART 控制寄存器 3) : E4H**

7	6	5	4	3	2	1	0
-	LOOPS	-	-	-	USBS	TX8	RX8
-	RW	-	-	-	RW	RW	R

Initial value : 00H

LOOPS		控制 UART 循环方式, 测试用
	0	常规模式
	1	循环方式
USBS		停止位长度
	0	1 Stop Bit
	1	2 Stop Bit
TX8		UART 数据第 9 位格式.在加载 UARTDR 寄存器前首先写入该位
	0	MSB (9 <sup>th</sup> bit) to be transmitted is ‘0’
	1	MSB (9 <sup>th</sup> bit) to be transmitted is ‘1’
RX8		UART 数据第 9 位格式. 在读接收缓存器前首先读取该位
	0	MSB (9 <sup>th</sup> bit) received is ‘0’
	1	MSB (9 <sup>th</sup> bit) received is ‘1’

## UARTST (UART 状态寄存器) : E5H

7	6	5	4	3	2	1	0
UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE
RW	RW	R	RW	RW	R	RW	RW

Initial value : 80H

UDRE	UDRE 标志位表明如果发送缓存器(UARTDR)已经准备接收新的数据。如果 UDRE 是 '1'，缓存器是空的并且已经准备写入，该位可以产生 UDRE 中断。 0 发送缓存器不是空的。 1 发送缓存器是空的。
TXC	当发送移位寄存器中的整帧数据已经移出并且没有新的数据存在于发送缓存器时该位置位。TXC 中断服务程序执行时该位自动清零。该位可以产生 TXC 中断。 0 发送进行中。 1 发送缓存器清空并且在发送移位寄存器中的数据已经完全移出。
RXC	在接收缓存器中没有未读数据时置位，所有数据读出时清。该位可以产生 RXC 中断。 0 在接收缓存器中没有未读数据 1 在接收缓存器中有至少一个数据未读
WAKE	CPU 睡眠模式中当 RXD 检测到低电平时该位置位，该位可用于产生唤醒中断。该位需要程序软件清除。 0 没有唤醒中断产生 1 产生唤醒中断
SOFTRST	这里有一个内部复位且仅对 UART 有用。对该位写入 '1'。初始化 UART 内部逻辑并且该位自动清除。 0 无操作 1 复位 UART
DOR	数据超出时置位。该位置位时，进来的数据被忽略。该位一直有效直到接收缓存器被读取。 0 没有超出 1 检测到数据超出
FE	如果第一个停止位被检测为 '0' 该位置为 1。该位一直有效直到接收缓存器被读取。 0 没有错误 1 检测到帧错误
PE	奇偶校验使能时如果接收缓存器中有校验错误该位被置 1。该位一直有效直到接收缓存器被读取。 0 没有错误 1 检测到奇偶校验错误

## 7.11.11、波特率设置 (举例)

## UART 常用波特率设置举例

Baud Rate	fx=1.00MHz		fx=1.8432MHz		fx=2.00MHz	
	UARTBD	ERROR	UARTBD	ERROR	UARTBD	ERROR
2400	25	0.2%	47	0.0%	51	0.2%
4800	12	0.2%	23	0.0%	25	0.2%
9600	6	-7.0%	11	0.0%	12	0.2%
14.4k	3	8.5%	7	0.0%	8	-3.5%
19.2k	2	8.5%	5	0.0%	6	-7.0%
28.8k	1	8.5%	3	0.0%	3	8.5%
38.4k	1	-18.6%	2	0.0%	2	8.5%
57.6k	-	-	1	-25.0%	1	8.5%
76.8k	-	-	1	0.0%	1	-18.6%
115.2k	-	-	-	-	-	-
230.4k	-	-	-	-	-	-

(continued)

Baud Rate	fx=3.6864MHz		fx=4.00MHz		fx=7.3728MHz	
	UARTBD	ERROR	UARTBD	ERROR	UARTBD	ERROR
2400	95	0.0%	103	0.2%	191	0.0%
4800	47	0.0%	51	0.2%	95	0.0%
9600	23	0.0%	25	0.2%	47	0.0%
14.4k	15	0.0%	16	2.1%	31	0.0%
19.2k	11	0.0%	12	0.2%	23	0.0%
28.8k	7	0.0%	8	-3.5%	15	0.0%
38.4k	5	0.0%	6	-7.0%	11	0.0%
57.6k	3	0.0%	3	8.5%	7	0.0%
76.8k	2	0.0%	2	8.5%	5	0.0%
115.2k	1	0.0%	1	8.5%	3	0.0%
230.4k	-	-	-	-	1	0.0%
250k	-	-	-	-	1	-7.8%
0.5M	-	-	-	-	-	-

(continued)

Baud Rate	fx=8.00MHz		fx=11.0592MHz	
	UARTBD	ERROR	UARTBD	ERROR
2400	207	0.2%	-	-
4800	103	0.2%	143	0.0%
9600	51	0.2%	71	0.0%
14.4k	34	-0.8%	47	0.0%
19.2k	25	0.2%	35	0.0%
28.8k	16	2.1%	23	0.0%
38.4k	12	0.2%	17	0.0%
57.6k	8	-3.5%	11	0.0%
76.8k	6	-7.0%	8	0.0%
115.2k	3	8.5%	5	0.0%
230.4k	1	8.5%	2	0.0%
250k	1	0.0%	2	-7.8%
0.5M	-	-	-	-
1M	-	-	-	-

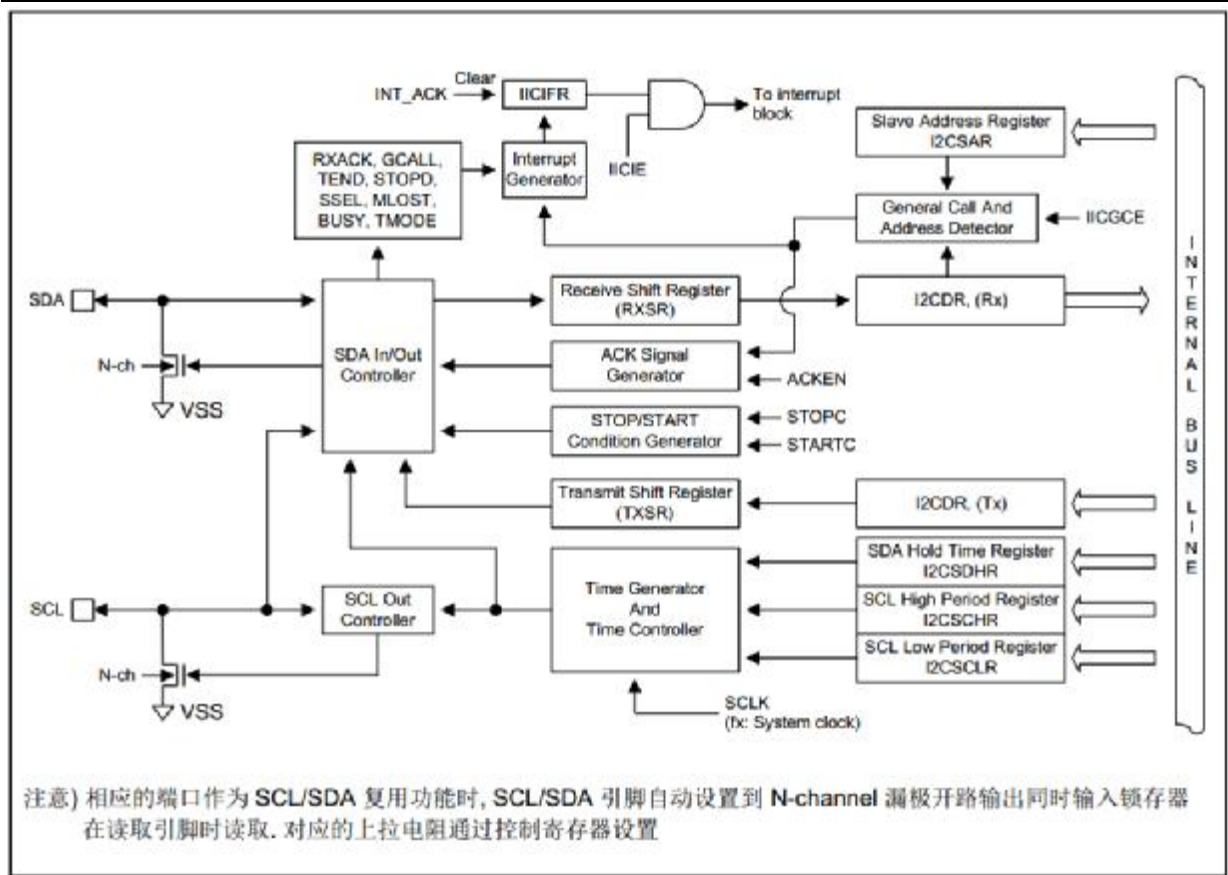
## 7.12、I<sup>2</sup>C

### 7.12.1、概述

I<sup>2</sup>C 是一种工业标准的串行通讯协议,它使用两根主线(SDA)和(SCL)交换数据。因为 SDA 和 SCL 都是漏极开路输出,每根线都需要上拉电阻。特性如下:

- 兼容 I<sup>2</sup>C 总线标准
- 多主控操作
- 高达 400kHz 数据传输速率
- 7 位地址
- 支持两个从动地址
- 主控和从动操作
- 总线忙时检测

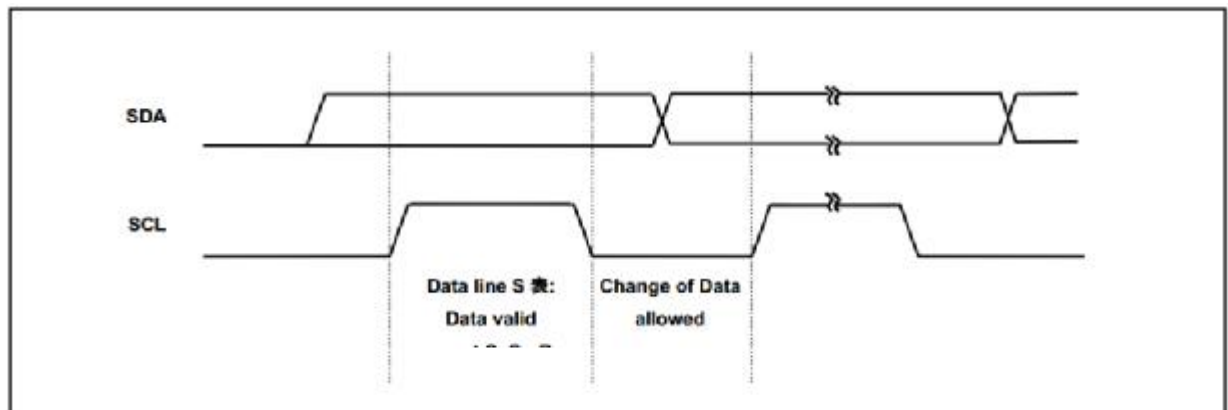
### 7.12.2、I<sup>2</sup>C 方框图



I<sup>2</sup>C 方框图

### 7.12.3、I<sup>2</sup>C 位传输

当时钟高电平周期时, SDA 总线上的数据必须保持稳定, SCL 线上的时钟信号是低时, 数据线上的高或低状态才可以改变。START(S), repeated START(Sr)和 STOP(P)情况除外, 这些情况下数据线改变发生在时钟线高电平时。



I<sup>2</sup>C-Bus 位传送

### 7.12.4、开始/重复 开始/停止信号

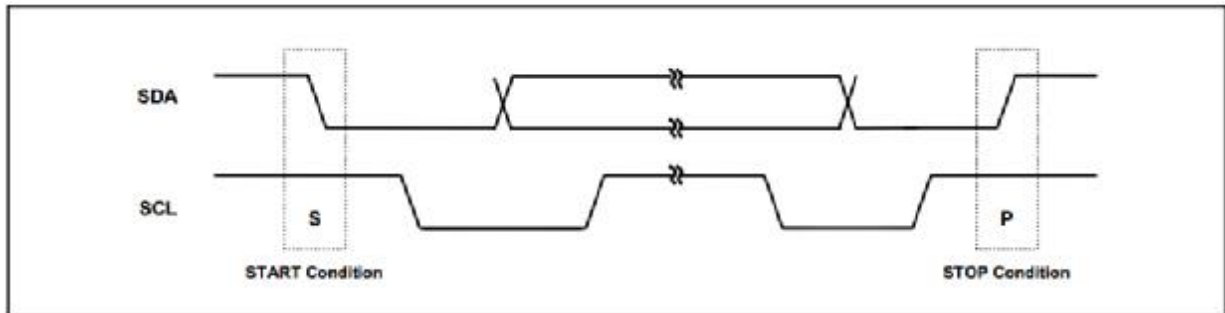
主控器会发出一个 START(S)信号通知其他设备连接他们将会用到的 SCL, SDA 总线。一个

STOP(P)信号由主控器产生去释放总线以便其他设备可以使用。

SCL 高电平时，SDA 上由高到低的转变被规定是一个 START(S)信号。

SCL 高电平时，SDA 上由低到高的转变被规定是一个 STOP(P)信号。

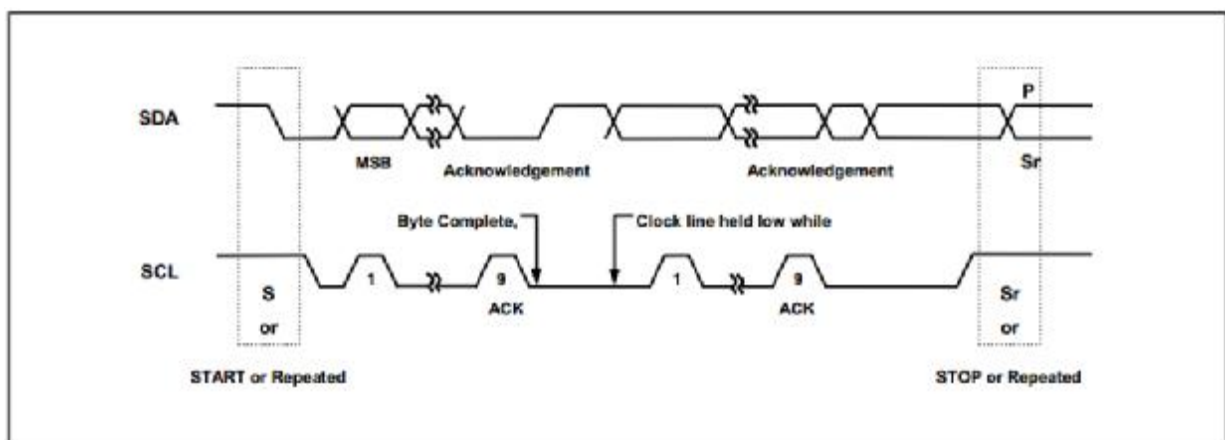
START 和 STOP 信号通常由主机产生。START 信号后总线通常被认为是忙碌。STOP 信号后总线被认为释放，ie 总线在 START 和 STOP 信号之间是忙碌状态。如果 STOP 信号被重复 START 信号(Sr)取代，总线保持忙碌状态。所以，START 和重复 START 信号功能上是一样的。



START 和 STOP 信号

### 7.12.5、数据传输

SDA 线上的每一个字节必须是 8 位长度。每次传输的字节数是没有限制的。每个字节必须跟随一个应答位。数据传送的方式是最高有效位优先(MSB)。从机不能接收或发送另一个完整数据字节除非它完成一些其他的功能，他可以保持 SCL 线上低电平强迫主机进入等待状态。当从机为另一字节数据做准备时释放 SCL 线，数据传输可以继续进行。

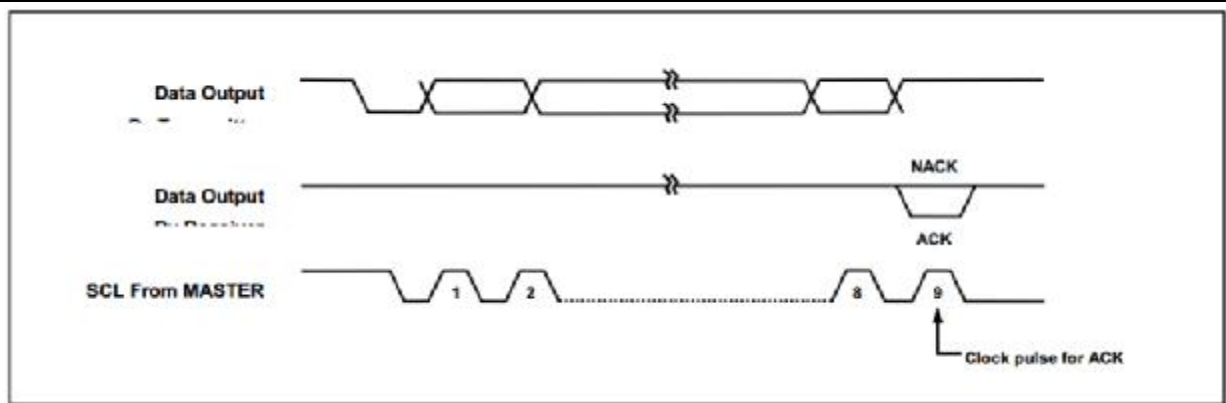


I<sup>2</sup>C-Bus 数据传输

### 7.12.6、I<sup>2</sup>C 应答

应答和主机产生的时钟脉冲宽度有关。在应答时钟脉冲中发送器释放 SDA(HIGH)。接收器必须拉低 SDA 线以便在时钟脉冲高时保持稳定的低电平。当从机被主机编址后，如果它因执行实时功能而不能接收或发送，数据线必须被从机拉高，以及，从机被主机编址后，不能接收更多数据位，从机接收器必须释放 SDA 线。主机可以产生一个 STOP 信号退出发送，或者重复开始信号开始新的传送。如果主机接收器涉及到发送，必须数据的结尾给从机发生器发送信号在输出主控器锁定的最后字节不产生应答。

从机发生器必须释放数据线以允许主机产生 STOP 或重复 START 信号。

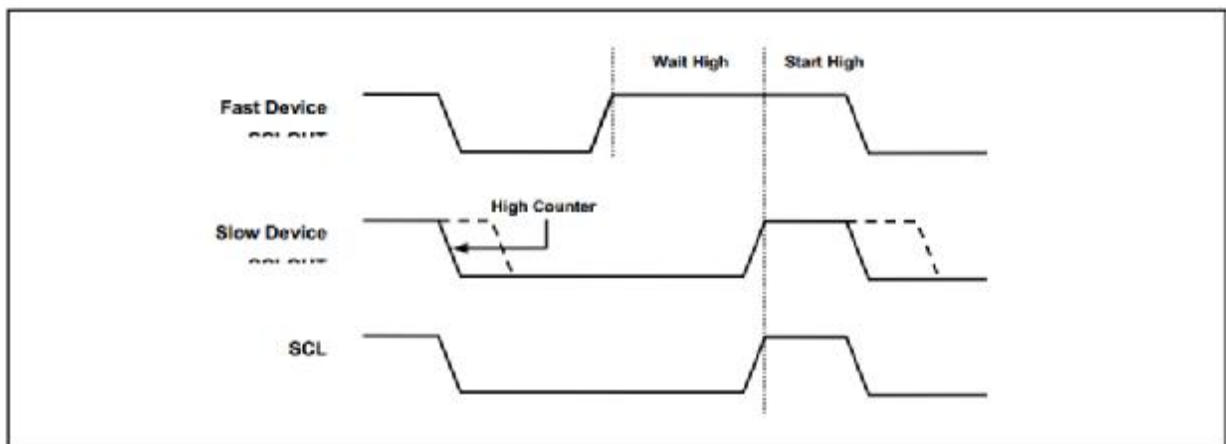


I²C-Bus 应答

### 7.12.7、同步/仲裁

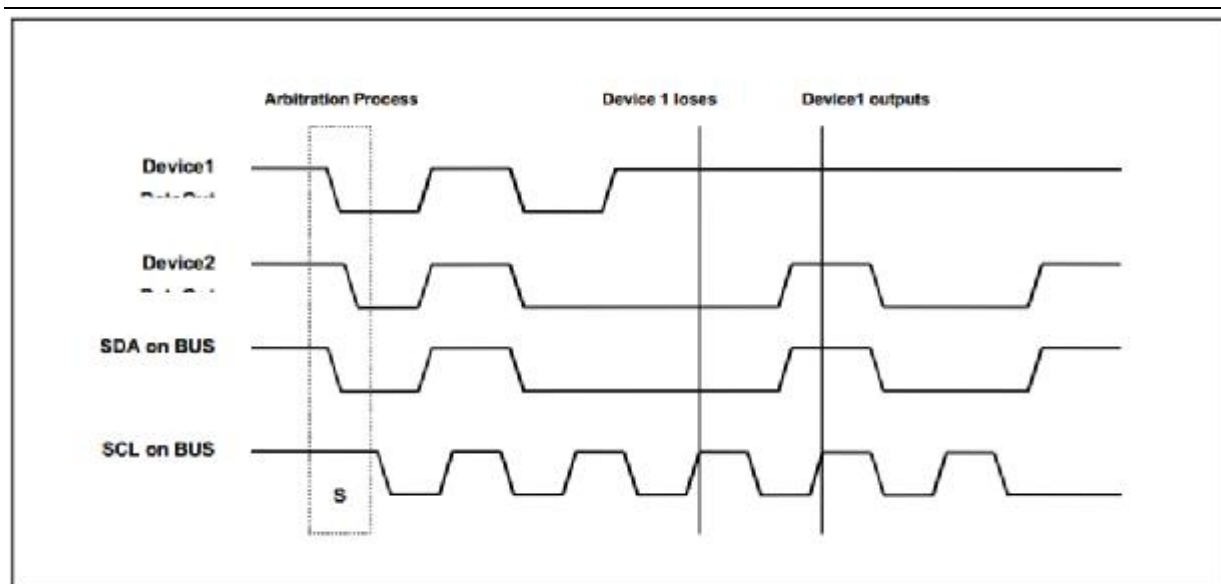
时钟同步是由 I²C 和 SCL 线与连接完成。意思就是 SCL 上一个由高到低到转变将会引起设备开始对低电平脉冲进行计数并保持 SCL 现在的状态直到时钟高电平被接收到。然而如果另一个时钟仍然维持在低电平的话，时钟由低到高的变化不能改变 SCL 线的状态。这样，随着最长的低电平脉冲产生一个同步 SCL 时钟，高电平由最短的高脉冲决定。

如果总线空闲时主机可以开始进行发送。两个或多个主机可以产生 START 信号。仲裁在 SDA 线上产生，当 SCL 高电平时，通过主机发送一个高电平，当另一个主机在发送低电平时会关闭数据输出状态，因为总线电平不等于它自身的电平。仲裁连续很多位直到一个主机赢得 I²C 总线。第一阶段是对照地址位。



仲裁过程中的时钟同步





双主机仲裁过程

### 7.12.8、操作

I<sup>2</sup>C 基于字节导向可产生中断。除了开始信号之外所有的总线活动结束后中断被发出。由于 I<sup>2</sup>C 基于中断，在 I<sup>2</sup>C 字节传送时，应用软件可以任意进行其他的操作。

注意当 I<sup>2</sup>C 中断产生时，IIFLAG 寄存器中的 IICIFR 标志位置位，它是通过写入任意的值到 I2CSR 进行清零。当 I<sup>2</sup>C 中断发生时，SCL 保持低电平直到 I2CSR 写入值。当 IICIFR 置位时，I2CSR 包含一个表示 I<sup>2</sup>C 当前状态的值。根据这个值，软件可以决定下一步该做什么。

I<sup>2</sup>C 通过设置主动/从动，发送/接收可以有 4 个操作模式。操作模式由赢得的主机来配置。以下是详细的操作说明。

#### 7.12.8.1、主机发送

I<sup>2</sup>C 作为主机发送时，以下是推荐的步骤。

1. 设置 I2CCR 的 IICEN 位使能 I<sup>2</sup>C。这里向外围提供主时钟。
2. 加载 SLA+W 到 I2CDR，这里 SLA 是从机的地址，W 是基于主机角度的传送方向。主机发送时，W 是 '0'。注意 I2CDR 被地址和数据共用。
3. 在 SCL 由低到高跳变时通过写入期望值到 I2CSCLR 和 I2CSCHR 设置波特率。
4. 设置 I2CSDHR 决定从 SCL 的下降沿开始何时 SDA 改变数值。如果 SDA 应该在 SCL 低脉冲的中间改变，加载 I2CSCLR 一半的值到 I2CSDHR。
5. 设置 I2CCR 的 STARTC 位。发送一个 START 信号。并且设置如何处理中断和 ACK 信号。当 STARTC 位置位时，I2CDR 中的 8 位数据根据波特率发送出来。
6. 这是主控机发送地址信息包的 ACK 处理信号。7 位地址和 1 位传输方向被发送到目标从机时，主机会知道在 SCL 高电平的第九位从机是否应答。如果主机赢得总线，I<sup>2</sup>C 会不管来自从机的 ACK 应答而产生 GCALL 中断。如果在仲裁过程中 I<sup>2</sup>C 丢失总线控制权，I2CSR 的 MLOST 会置位。I<sup>2</sup>C 会以空闲状态等待或者作为一个可寻址从机来操作。当 I2CSR 的 MLOST 位置位时作为从机来操作，I2CCR 中的 ACKEN 位必须置位同时接收到的 7 位地址必须等于 I2CSAR 的 SLA 位。假如这样的话 I<sup>2</sup>C 作为从机发送或从机接收来操作。在这个阶段，I<sup>2</sup>C 保持 SCL 低电平。这是由于决定

---

I2C 是否继续传送还是停止通讯。假如 I2C 在第一个数据传送时没有丢失总线主控权继续下一步。I2C(Master)可以忽略从机的 ACK 应答信号而选择一下的某一个情况：

- 1) 主机从从机接收 ACK 信号，由于从机可以从主机接收更多的数据所以继续数据传送。这样的话，加载数据到 I2CDR。
- 2) 主机停止数据传送即使它接收从机的 ACK 信号。此时 I2CCR 的 STOPC 位置位。
- 3) 主机不检测 ACK 信号而发送重复 START 信号。此时，加载 SLA+R/W 到 I2CDR 同时 I2CCR 的 STARTC 位置位。

在做了以上的这些动作以后，写入任意值到 I2CSR 释放 SCL 线。在第一种情况时 1)，转移到 7。如果是第二种情况 2)，转移到 9 去处理 STOP 中断。如果是第三种情况 3)，发送完 I2CDR 中的数据后转移到 6，同时如果传送方向位是 ‘1’ 转到主机接收部分。

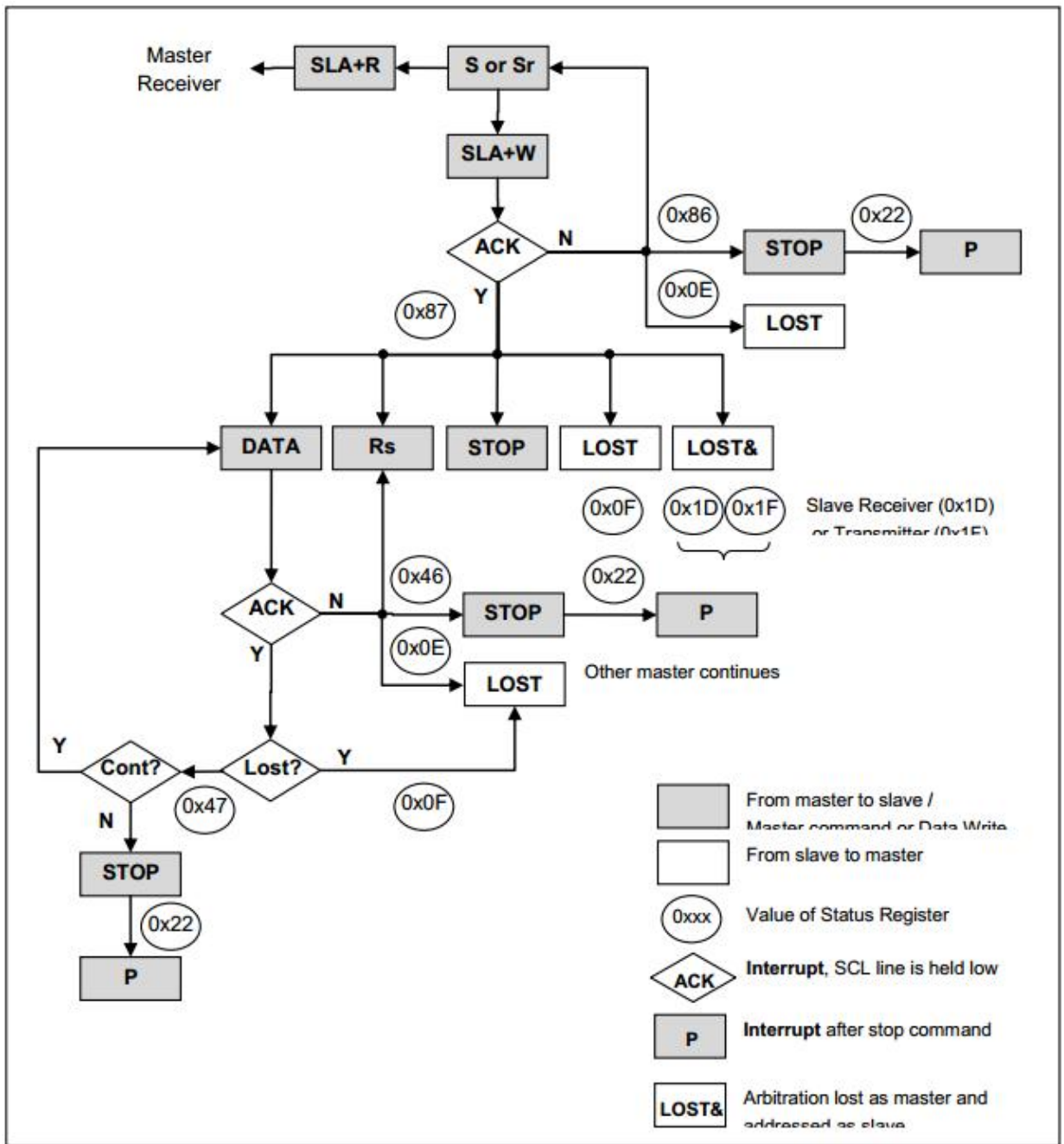
7. 发送一个字节的数据，在传送过程中，总线仲裁继续

8. 这是主控器传送信息数据包的 ACK 信号处理环节。I2C 保持 SCL 低电平。在传送数据仲裁其他主控器 I2C 丢失总线控制权时，I2CSR 的 MLOST 置位，此时，I2C 在空闲状态等待。I2CDR 中的数据发送完成，I2C 产生 TEND 中断。I2C 可以忽略从机的 ACK 信号选择以下的一种情况。

- 1) 主机从从机接收 ACK 信号，由于从机可以从主机接收更多的数据所以继续数据传送。这样的话，加载数据到 I2CDR。
- 2) 主机停止数据传送即使它接收从机的 ACK 信号。此时 I2CCR 的 STOPC 位置位
- 3) 主机不检测 ACK 信号而发送重复 START 信号。此时，加载 SLA+R/W 到 I2CDR 同时 I2CCR 的 STARTC 位置位。

在做了以上的这些动作以后，写入任意值到 I2CSR 释放 SCL 线。在第一种情况时 1)，转移到 7。如果是第二种情况 2)，转移到 9 去处理 STOP 中断。如果是第三种情况 3)，发送完 I2CDR 中的数据后转移到 6，同时如果传送方向位是 ‘1’ 转到主机接收部分。

9. 这是 I2C 传送功能的最后一步，执行 STOP 中断，STOP 位表示主机和从机之间的数据传送已经完成。写入任意值到 I2CSR 清除 I2CSR，然后 I2C 进入空闲状态。

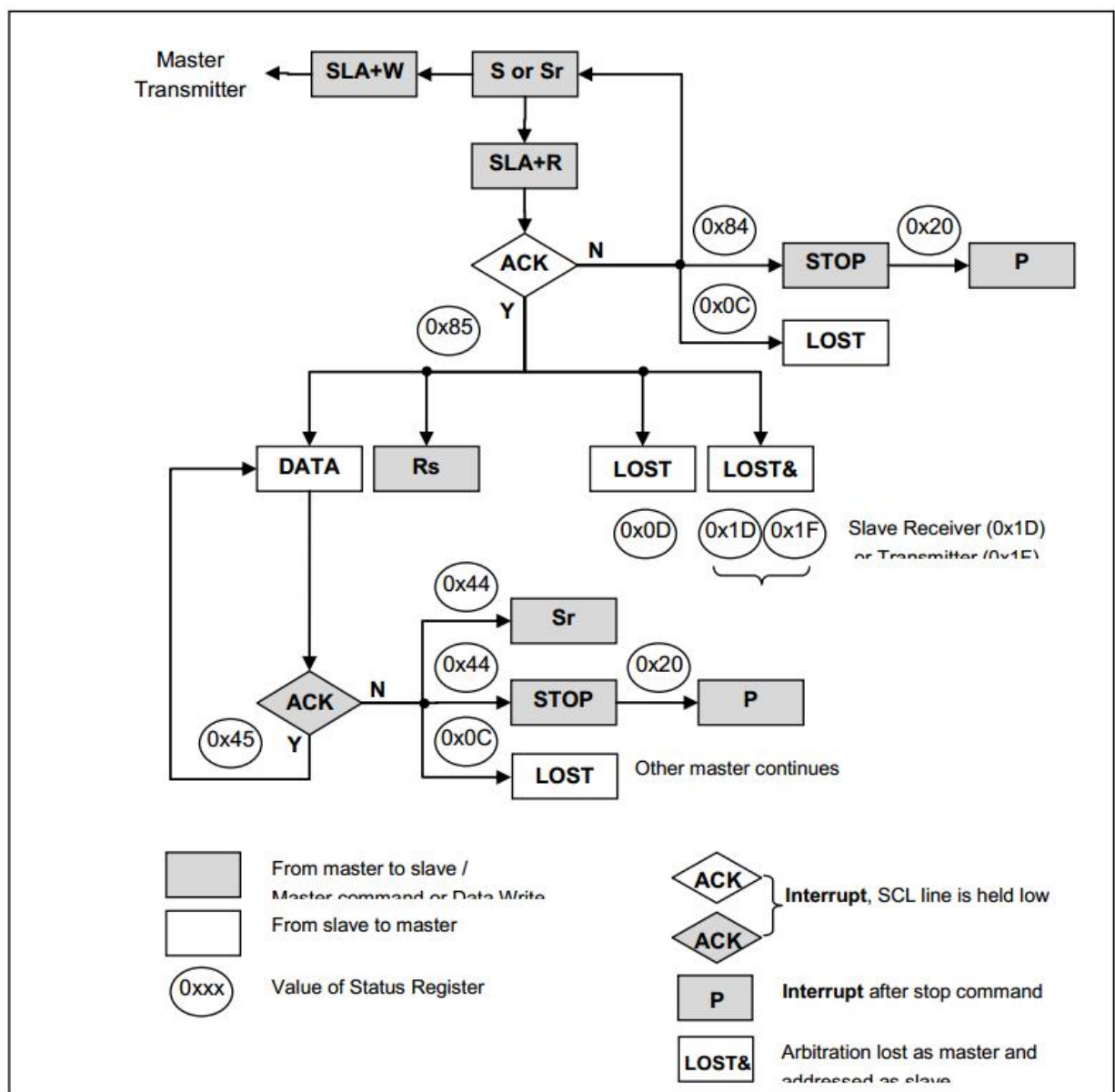


主控发送模式

### 7.12.8.2、主机接收

I2C 作为主机接收时，以下是操作步骤

1. 设置 I2CCR 的 IICEN 位使能 I2C。这里向外围提供主时钟。
2. 加载 SLA+R 到 I2CDR，这里 SLA 是从机的地址，R 是基于主机角度的传送方向。主机接收时，R 是 '1'。注意 I2CDR 被地址和数据共用。
3. 在 SCL 由低到高跳变时通过写入期望值到 I2CSCLR 和 I2CSCHR 设置波特率。
4. 设置 I2CSDHR 决定从 SCL 的下降沿开始何时 SDA 改变数值。如果 SDA 应该在 SCL 低脉冲的中间改变，加载 I2CSCLR 一半的值得到 I2CSDHR。
5. 设置 I2CCR 的 STARTC 位。发送一个 START 信号。并且设置如何处理中断和 ACK 信号。当 STARTC 位置位时，I2CDR 中的 8 位数据根据波特率发送出来。
6. 这是主控机发送地址信息包的 ACK 处理信号。7 位地址和 1 位传输方向被发送到目标从机时，主机会知道在 SCL 高电平的第九位从机是否应答。如果主机赢得总线，I2C 会忽略来自从机的 ACK 应答而产生 GCALL 中断。如果在仲裁过程中 I2C 丢失总线控制权，I2CSR 的 MLOST 会置位，I2C 会以空闲状态等待或者作为一个可寻址从机来操作。当 I2CSR 的 MLOST 位置位而作为从机来操作，I2CCR 中的 ACKEN 位必须置位同时接收到的 7 位地址必须等于 I2CSAR 的 SLA 位。假如这样的话 I2C 作为从机发送或从机接收来操作。在这个阶段，I2C 保持 SCL 低电平。这是由于决定 I2C 是否继续传送还是停止通讯。假如 I2C 在第一个数据传送时没有丢失总线主控权继续下一步。I2C(Master)可以根据从机的 ACK 应答信号而选择以下的某一种情况：
  - 1) 主机从从机接收 ACK 信号，由于从机可以从主机接收更多的数据所以继续数据传送。设置 I2CCR 的 ACKEN 位决定 I2C 是否应答下一个接收的数据。
  - 2) 主机停止数据传送因为没有接收到从机的 ACK 信号。此时 I2CCR 的 STOPC 位置位。
  - 3) 主机由于没有检测到从机的 ACK 信号而发送重复 START 信号。此时，加载 SLA+R/W 到 I2CDR 同时 I2CCR 的 STARTC 位置位。在做了以上的某个动作以后，写入任意值到 I2CSR 释放 SCL 线。在第一种情况时 1)，转移到 7。如果是第二种情况 2)，转移到 9 去处理 STOP 中断。如果是第三种情况 3)，发送完 I2CDR 中的数据后转移到 6，同时如果传送方向位是 '0' 转到主机发送部分。
7. 接收一个字节的的数据。
8. 这是主控器传送信息数据包的 ACK 信号处理环节。I2C 保持 SCL 低电平。当一个字节数据接收完成，I2C0 可以根据 I2CSR 的 RXACK 标志位而选择以下的某一种情况：
  - 1) 主机继续从从机接收数据。要这样做须设置 I2CCR 的 ACKEN 去应答下一个要接收的数据。
  - 2) 如果没有接收到下个数据的 ACK 信号主机想要终止数据传送，可以通过清除 I2CCR 的 ACKEN 来这样做。
  - 3) 由于没有检测到 ACK 信号，主机终止数据传送，此时，I2CCR 的 STOPC 置位。
  - 4) 没有检测到 ACK 信号，主机发送重复 START 信号。在这种情况下，加载 SLA+R/W 到 I2CDR，I2CCR 的 STARTC 置位。在做了以上的某个动作以后，写入任意值到 I2CSR 释放 SCL 线。在第一种情况或第二种情况时)，转移到 7。如果是第三种情况 3)，转移到 9 去处理 STOP 中断，如果是第四种情况，发送完 I2CDR 中的数据后转移到 6，同时如果传送方向位是 '0' 转到主机发送部分。
9. 这是 I2C 传送功能的最后一步，执行 STOP 中断，STOP 位表示主机和从机之间的数据传送已经完成。写入任意值到 I2CSR 清除 I2CSR，然后 I2C 进入空闲状态。



主机接收模式

### 7.12.8.3、I<sup>2</sup>C 从机发送

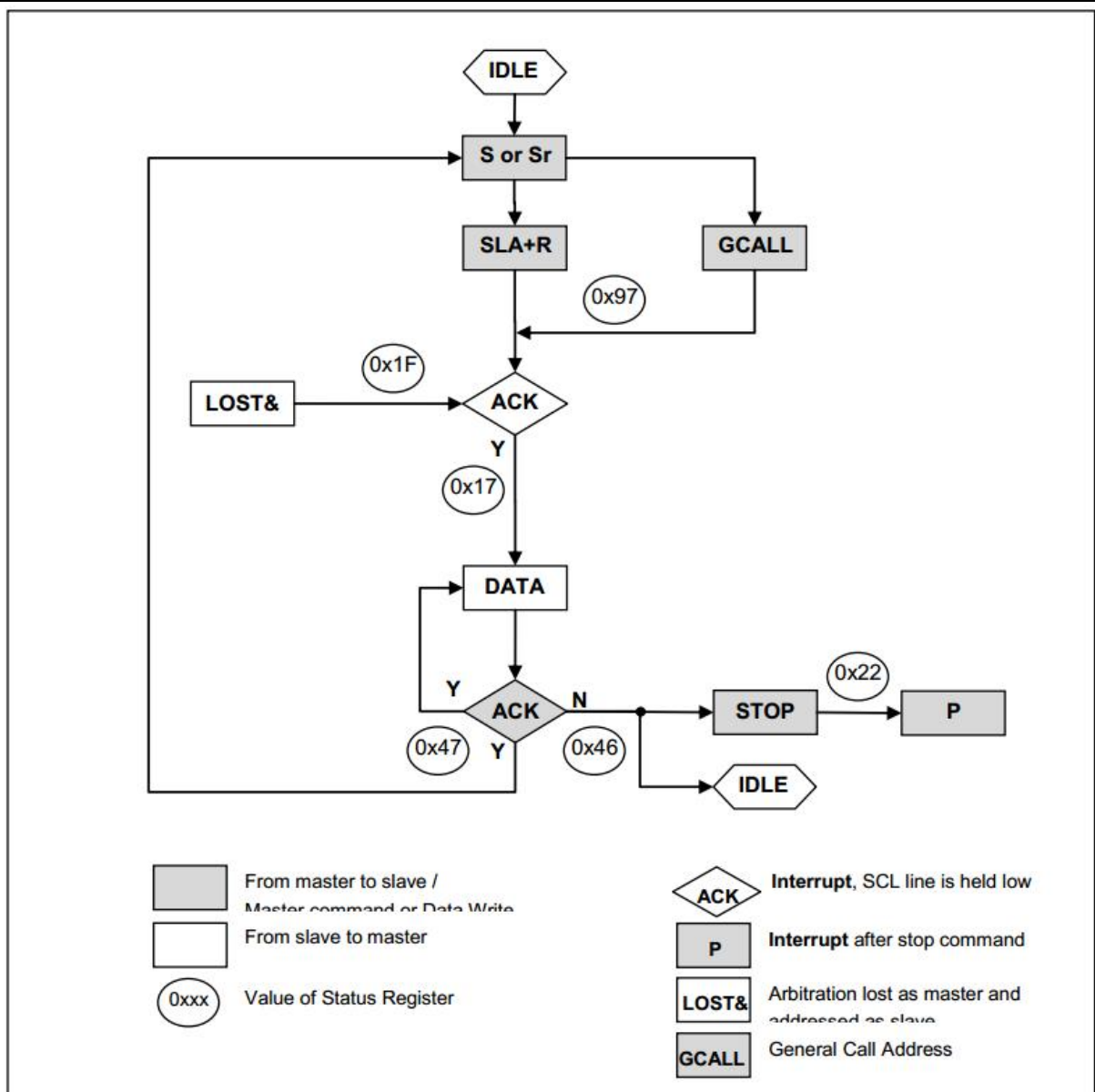
I2C 作为从机发送时，以下是推荐的步骤。

1. 如果系统主操作时钟慢于 SCL，加载 0x00 到 I2CSDHR，使 SDA 从 SCL 的下降沿伴随着一个系统

---

脉冲开始改变。注意 SDA 的维持时间由 SCLK 的 SDAH 周期计算，这里 SDAH 是来自 I2CSDHR 的数。当 SDA 的维持时间长于 SCLK 周期时，I2C(slave)不适合发送串行数据。

2. 设置 I2CCR 的 IICIE 和 IICEN 位使能 I2C。这里向外围提供系统主时钟
3. 当检测到一个 START 信号，I2C 接收一个字节的的数据和 I2CSAR 的 SLA 位比较。如果 I2CSAR 的 GCALLEN 位使能，I2C 将接收到的数据和 0x00 进行比较，一般调用地址。
4. 如果接收地址不等于 I2CSAR 的 SLA，I2C 进入空闲状态等待另一个开始信号。否则如果地址等于 SLA 位并且 ACKEN 位是打开的，I2C 产生 SSEL 中断并且 SCL 线保持低电平。注意即使地址等于 SLA，当 ACKEN 位禁止时，I2C 仍进入空闲状态。SSEL 中断发生时，加载传送数据到 I2CDR 写入任意值到 I2CSR 来释放 SCL。
5. 传输一个字节的的数据。
6. 在这一过程中，I2C 产生 TEND 中断并且维持 SCL 低电平而忽略来自主机的 ACK 应答信号。从机可以选择以下某一种情况：
  - 1) 没有检测到 ACK 信号，I2C 等待 STOP 或重复的 START 信号。
  - 2) 检测到来自主机的 ACK 信号。加载数据到 I2CDR。在做了以上的某个动作以后，写入任意值到 I2CSR 释放 SCL 线。在第一种情况)，转移到 7 终止通讯。如果是第 2 种情况)，转移到 5，无论发生哪种情况，只要检测到重复开始信号，就转到 4。
7. 这是 I2C 从机发送功能的最后一步，执行 STOP 中断。STOPC 位表示主机和从机之间的数据传送已经完成。写入任意值到 I2CSR 清除 I2CSR，然后 I2C 进入空闲状态。



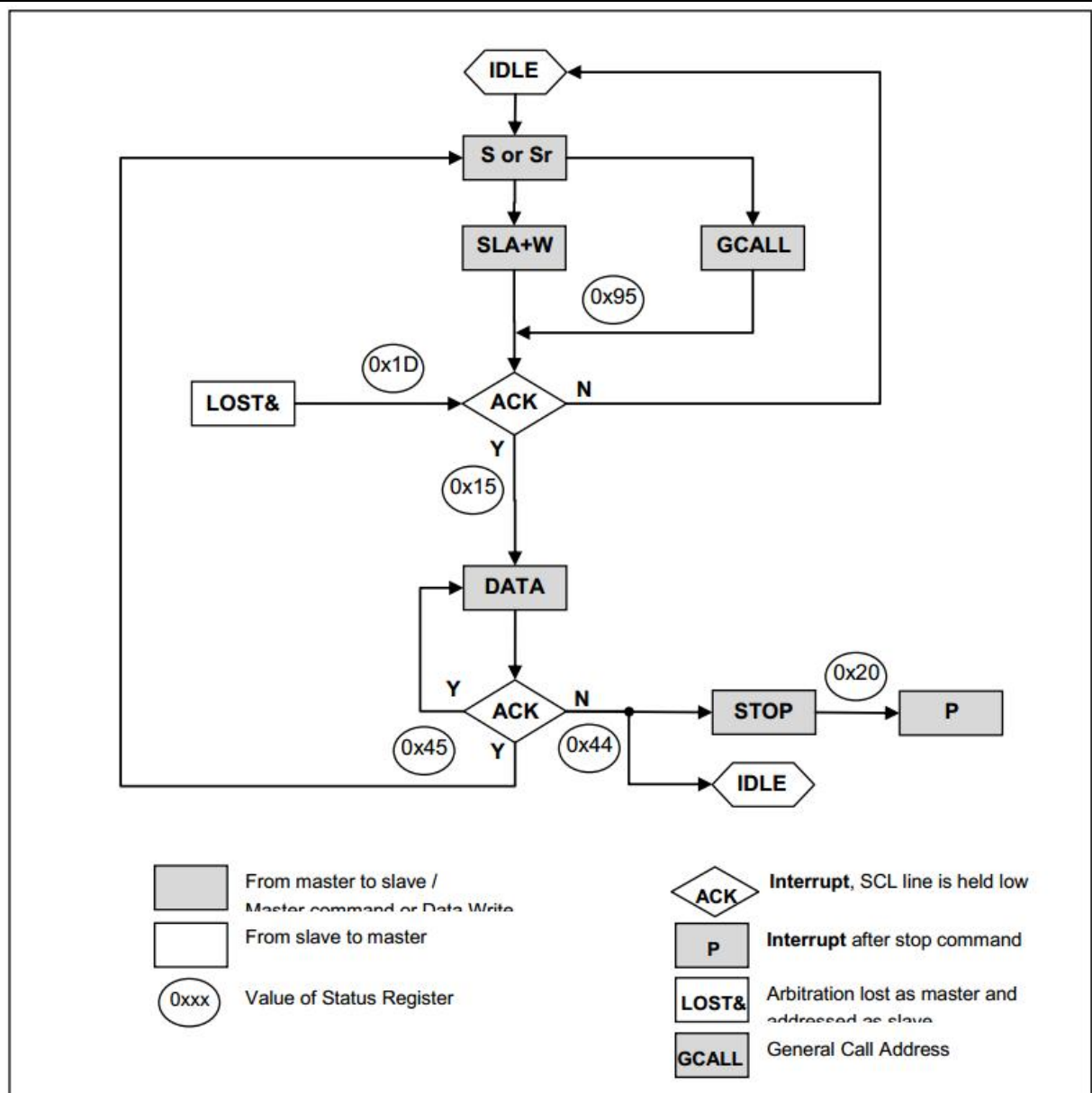
从机发送模式

#### 7.12.8.4、从机接收

I2C 作为从机接收时，以下是推荐的步骤。

- 
1. 如果系统主操作时钟慢于 SCL，加载 0x00 到 I2CSDHR，使 SDA 从 SCL 的下降沿伴随着一个系统脉冲开始改变。注意 SDA 的维持时间由 SCLK 的 SDAH 周期计算，这里 SDAH 是来自 I2CSDHR 的数。当 SDA 的维持时间长于 SCLK 周期时，I2C(slave)不适合发送串行数据。
  2. 设置 I2CCR 的 IICIE 和 USIEN 位使能 I2C。这里向外围提供系统主时钟。
  3. 当检测到一个 START 信号，I2C 接收一个字节的的数据和 I2CSAR 的 SLA 位比较。如果 I2CSAR 的 GCALLEN 位使能，I2C 将接收到的数据和 0x00 进行比较，一般调用地址。
  4. 如果接收地址不等于 I2CSAR 的 SLA，I2C 进入空闲状态等待另一个开始信号。否则如果地址等于 SLA 位并且 ACKEN 位是打开的，I2C 产生 SSEL 中断并且 SCL 线保持低电平。注意即使地址等于 SLA，当 ACKEN 位禁止时，I2C 仍进入空闲状态。SSEL 中断发生时，I2C 准备接收数据，写入任意值到 I2CSR 来释放 SCL。
  5. 传输一个字节的的数据。
  6. 在这一过程中，I2C 产生 TEND 中断并且维持 SCL 低电平而忽略来自主机的 ACK 应答信号。从机可以选择以下某一种情况：
    - 1) 没有检测到 ACK 信号(ACKEN=0)，I2C 等待 STOP 或重复的 START 信号。
    - 2) 检测到来自主机的 ACK 信号(ACKEN=1)。I2C 可继续从主机接收数据。在做了以上的某个动作以后，写入任意值到 I2CSR 释放 SCL 线。在第一种情况)，转移到 7 终止通讯。如果是第 2 种情况)，转移到 5，无论发生哪种情况，只要检测到重复开始信号，就转到 4。
  7. 这是 I2C 从机接收功能的最后一步，执行 STOP 中断。STOPC 位表示主机和从机之间的数据传送已经完成。写入任意值到 I2CSR 清除 I2CSR，然后 I2C 进入空闲状态。





从机接收模式

7.12.9、寄存器图

I2C 寄存器图

Name	Address	Dir	Default	Description
I2CCR	E9H	R/W	00H	I2C Control Register
I2CSR	EAH	R/W	00H	I2C Status Register
I2CSAR0	EBH	R/W	00H	I2C Slave Address 0 Register
I2CSAR1	F1H	R/W	00H	I2C Slave Address 1 Register
I2CDR	ECH	R/W	00H	I2C Data Register
I2CSDHR	EDH	R/W	01H	I2C SDA Hold Time Register
I2CSCLR	EEH	R/W	3FH	I2C SCL Low Period Register
I2CSCHR	EFH	R/W	3FH	I2C SCL High Period Register

7.12.10、I<sup>2</sup>C 寄存器说明

I2C 模块由 (I2CCR), (I2CSR), (I2CSAR0/I2CSAR1), (I2CDR), (I2CSDHR), (I2CSCLR), 和 (I2CSCHR)组成。

7.12.11、I<sup>2</sup>C 寄存器说明

**I2CDR (I2C 数据寄存器) : ECH**

7	6	5	4	3	2	1	0
I2CDR7	I2CDR6	I2CDR5	I2CDR4	I2CDR3	I2CDR2	I2CDR1	I2CDR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

I2CDR[7:0] I2CDR 发送、接收缓存器和数据寄存器共用相同的 I/O 地址，发送数据缓存器是 I2CDR 的写入目标。读取 I2CDR 寄存器时返回接收寄存器的内容。

**I2CSDHR (I2C SDA 保持时间寄存器) : EDH**

7	6	5	4	3	2	1	0
I2CSDHR7	I2CSDHR6	I2CSDHR5	I2CSDHR4	I2CSDHR3	I2CSDHR2	I2CSDHR1	I2CSDHR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 01H

I2CSDHR[7:0] 该寄存器用于控制从 SCL 的下降沿开始 SDA 输出时机注意 SDA 在  $t_{SCLK}*(I2CSDHR+2)$ 后开始改变，主控模式中，加载 I2CSCLR 一半的值到该寄存器使 SDA 在 SCL 的中间改变。从动模式时，该寄存器与来自主机的 SCL 频率有关。主控模式时 SDA 在  $t_{SCLK}*(I2CSDHR+2)$ 后开始改变。所以，确保在从动时的操作和数值  $t_{SCLK}*(I2CSDHR+2)$ 必须小于 SCL 周期。

**I2CSCHR (I2C SCL 高电平周期寄存器) : EFH**

7	6	5	4	3	2	1	0
I2CSCHR7	I2CSCHR6	I2CSCHR5	I2CSCHR4	I2CSCHR3	I2CSCHR2	I2CSCHR1	I2CSCHR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 3FH

I2CSCHR[7:0] 该寄存器定义主控模式时 SCL 高脉冲周期。基础时钟是 SCLK，系统时钟和周期通过公式计算： $t_{SCLK}*(4*I2CSCHR+2)$ ，这里  $t_{SCLK}$  是 SCLK 的周期。

I<sup>2</sup>C 主控模式操作频率计算公式：

$$f_{I2C} = \frac{1}{t_{SCLK} \times (4 \times (I2CSCLR + I2CSCHR) + 4)}$$

**I2CSCLR (I2C SCL 低电平周期寄存器) : EEH**

7	6	5	4	3	2	1	0
I2CSCLR7	I2CSCLR6	I2CSCLR5	I2CSCLR4	I2CSCLR3	I2CSCLR2	I2CSCLR1	I2CSCLR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 3FH

I2CSCLR[7:0] 该寄存器定义主控模式时 SCL 低电平。基础时钟试 SCLK，系统时钟和周期计算公式： $t_{SCLK}*(4*I2CSCLR+2)$ 这里  $t_{SCLK}$  是 SCLK 周期。

**I2CSAR0 (I2C 从机地址寄存器 0) : EBH**

7	6	5	4	3	2	1	0
I2CSLA06	I2CSLA05	I2CSLA04	I2CSLA03	I2CSLA02	I2CSLA01	I2CSLA00	GCALLOEN
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

I2CSLA0[6:0] 该位在从机模式时配置从机地址 0  
 GCALLOEN 该位决定在从机模式时 I<sup>2</sup>C 是否允许产生调用地址  
 0 忽略调用地址  
 1 允许产生调用地址

**I2CSAR1 (I2C 从机地址寄存器 1) : F1H**

7	6	5	4	3	2	1	0
I2CSLA16	I2CSLA15	I2CSLA14	I2CSLA13	I2CSLA12	I2CSLA11	I2CSLA10	GCALL1EN
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

I2CSLA1[6:0] 该位在从动模式时配置从动地址 1  
 GCALL1EN 该位决定在从机模式时 I<sup>2</sup>C 是否允许产生调用地址  
 0 忽略调用地址  
 2 允许产生调用地址

**I2CCR (I2C 控制寄存器) : E9H**

7	6	5	4	3	2	1	0
IICRST	IICEN	TXDLYENB	IICIE	ACKEN	IMASTER	STOPC	STARTC
RW	RW	RW	RW	RW	R	RW	RW

Initial value : 00H

IICRST 初始化 I2C 内部寄存器  
 0 无效  
 1 初始化 I2C，自动清除  
 IICEN 激活 I2C 功能模块  
 0 I2C 禁止  
 1 I2C 使能  
 TXDLYENB I2CSDHR 寄存器控制位  
 0 使能 I2CSDHR 寄存器  
 1 禁止 I2CSDHR 寄存器  
 IICIE 中断使能位  
 0 禁止中断(use polling)

	1	使能 I2C 中断
ACKEN		控制在第 9 个 SCL 周期 ACK 信号的产生
	0	没有 ACK 信号产生(SDA = 1)
	1	产生 ACK 信号(SDA = 0)
		注意) ACK 信号输出有以下三种模式(SDA=0).
		1. 当接收到的地址包等于 I2CSAR 的 I2CSLA 位
		2. 当接收到的地址包等于 0x00 且 GCALL 使能.
		3. 当 I2C 作为接收器操作时(主动或从动)
IMASTER		I2C 的操作模式
	0	I2C 是从动模式
	1	I2C 是主控模式
STOPC		I2C 是主控制器时, STOP 信号的产生
	0	无效
	1	产生 STOP 信号
STARTC		I2C 是主控制器时, START 信号的产生
	0	无效
	1	产生 START 或重复 START 信号
		注意) 请参考(IIFLAG)I2C 中断标志位

**I2CSR (I2C 状态寄存器) : EAH**

7	6	5	4	3	2	1	0
GCALL	TEND	STOPD	SSEL	MLOST	BUSY	TMODE	RXACK
R	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

GCALL <sup>(NOTE)</sup>		该位根据主控还是从动有不同的意思。主控时, 该位代表是否从从机接收 AACK (address ACK)
	0	没有接收到 AACK(Master mode)
	1	接收到 AACK(Master mode)
		从动模式时, 该位用于表示一般调用
	0	没有检测到常规调用地址(Slave mode)
	1	检测到常规调用地址(Slave mode)
TEND <sup>(NOTE)</sup>		一个字节数据发送完成后该位置 1
	0	1 个字节的数据没有被全部发送
	1	1 个字节数据被全部发送
STOPD <sup>(NOTE)</sup>		检测到 STOP 信号该位置 1
	0	没有检测到 STOP 信号
	1	检测到 STOP 信号
SSEL <sup>(NOTE)</sup>		I <sup>2</sup> C 被其他主机编址时置 1
	0	I <sup>2</sup> C 没有被作为从机选定
	1	I <sup>2</sup> C 被其他主机编址充当从机
MLOST <sup>(NOTE)</sup>		该位代表主控模式中总线仲裁结果

---

	0	I <sup>2</sup> C 保持总线控制权
	1	I <sup>2</sup> C 已经丢失总线控制权
BUSY		该位标志总线状态
	0	I <sup>2</sup> C 总线空闲, 主机可以发送一个 START 信号
	1	I <sup>2</sup> C 总线忙碌
TMODE		该位表示 I <sup>2</sup> C 是发送还是接收
	0	I <sup>2</sup> C 接收
	1	I <sup>2</sup> C 发送
RXACK		显示 ACK 信号状态
	0	没有 ACK 信号
	1	SCL 第 9 个周期接收到 ACK 信号

注意) 这些位可作为中断源。

当 I<sup>2</sup>C 中断发生除了 STOP 模式外, SCL 保持低电平。要释放 SCL, 写入任意值到 I2CSR。I2CSR 被写入时, TEND, STOPD, SSEL, MLOST 和 RXACK 位被清除。

## 8、省电模式操作

### 8.1、概述

YF8316 16/32 有两个省电模式减少设备电量消耗。在省电模式中，电量消耗很多。设备提供 3 种省电功能 MainIDLE, Sub-IDLE 和 STOP 模式。在 3 中模式下，程序停止。

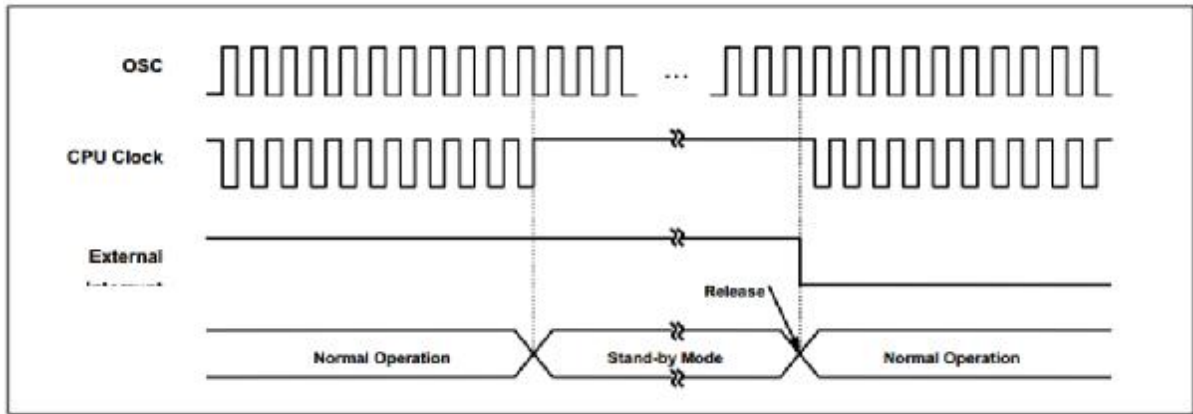
## 8.2、IDLE/STOP 模式下外围的操作

省电模式下外围操作

Peripheral	IDLE Mode	STOP Mode
CPU	ALL CPU Operation are Disable	ALL CPU Operation are Disable
RAM	Retain	Retain
Basic Interval Timer	Operates Continuously	Stop
Watch Dog Timer	Operates Continuously	Stop (Can be operated with WDTRC OSC)
Watch Timer	Operates Continuously	Stop (Can be operated with sub clock)
Timer0~2	Operates Continuously	Halted (Only when the Event Counter Mode is Enabled, Timer operates Normally)
ADC	Operates Continuously	Stop
BUZ	Operates Continuously	Stop
SPI	Operates Continuously	Only operate with external clock
UART	Operates Continuously	Stop
I2C	Operates Continuously	Only operate with external clock
Internal OSC (16MHz)	Oscillation	Stop when the system clock (fx) is firc
WDTRC OSC (5kHz)	Can be operated with setting value	Can be operated with setting value
Main OSC (0.4~12MHz)	Oscillation	Stop when fx = fxIN
Sub OSC (32.768kHz)	Oscillation	Stop when fx = fsUB
I/O Port	Retain	Retain
Control Register	Retain	Retain
Address Data Bus	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET, Timer Interrupt (EC0, EC1, EC2), SPI (External clock), External Interrupt, UART by RX, WT (sub clock), WDT

## 8.3、IDLE 模式

电源控制寄存器设置为 '01h' 进入到 IDLE 模式。该模式下，内部振荡电路保持活动。振荡继续同时外围设备也正常操作，但是 CPU 停止。通过复位或振荡来唤醒。中断唤醒时，中断需要在 IDLE 模式前使能。如果使用复位，由于设备变为初始化状态，寄存器也是复位值。



外部中断唤醒 IDLE 模式时序图

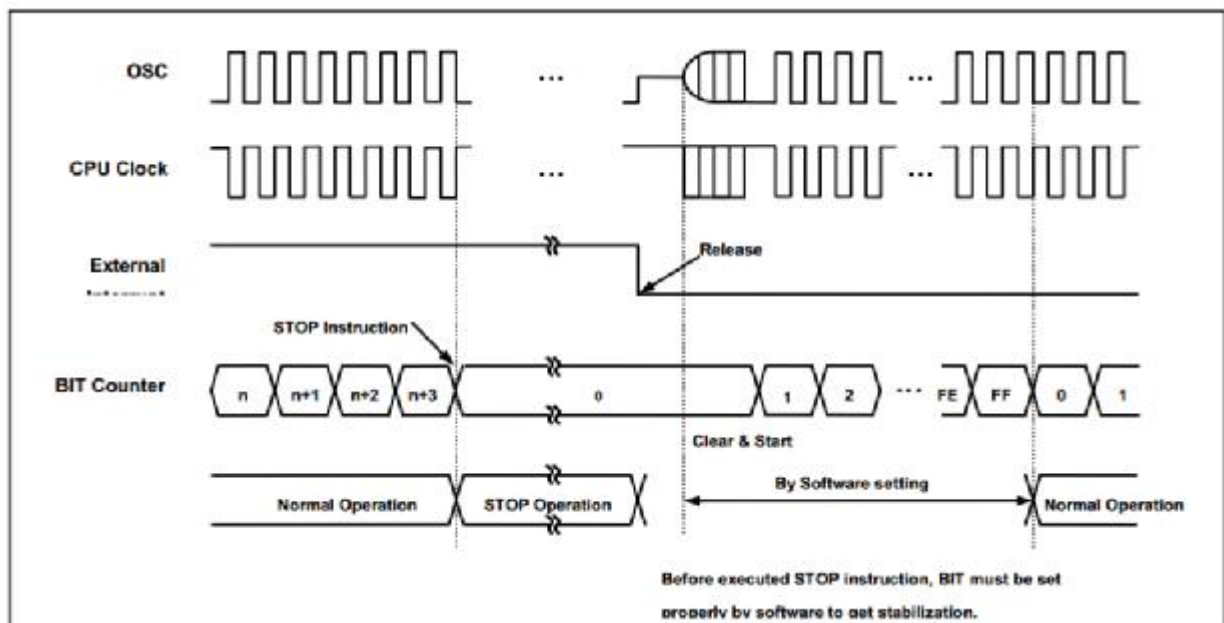
### 8.4、STOP 模式

电源控制寄存器设置为 ‘03H’ 进入 STOP 模式。在该模式下，所选振荡器，系统时钟或外围时钟停止，但是 watch timer 可以继续通过 sub clock 继续操作。随着时钟的冻结，所有功能都停止，只有片内 RAM 和控制寄存器仍然保持。例如，如果内部 RC(fIRC)被选做系统时钟并且 subclock(fSUB)是振荡的，在 STOP 模式时内部 RC 停止振荡，sub clock 仍然继续振荡。此时，watch timer 可以通过 sub clock 继续操作。

STOP 模式的唤醒源有硬件复位和中断。

复位时重新定义所有控制寄存器。退出 STOP 时，需要足够的时间使振荡稳定到正常操作。下图所示为时序图。当从 STOP 退出时，唤醒情况下基

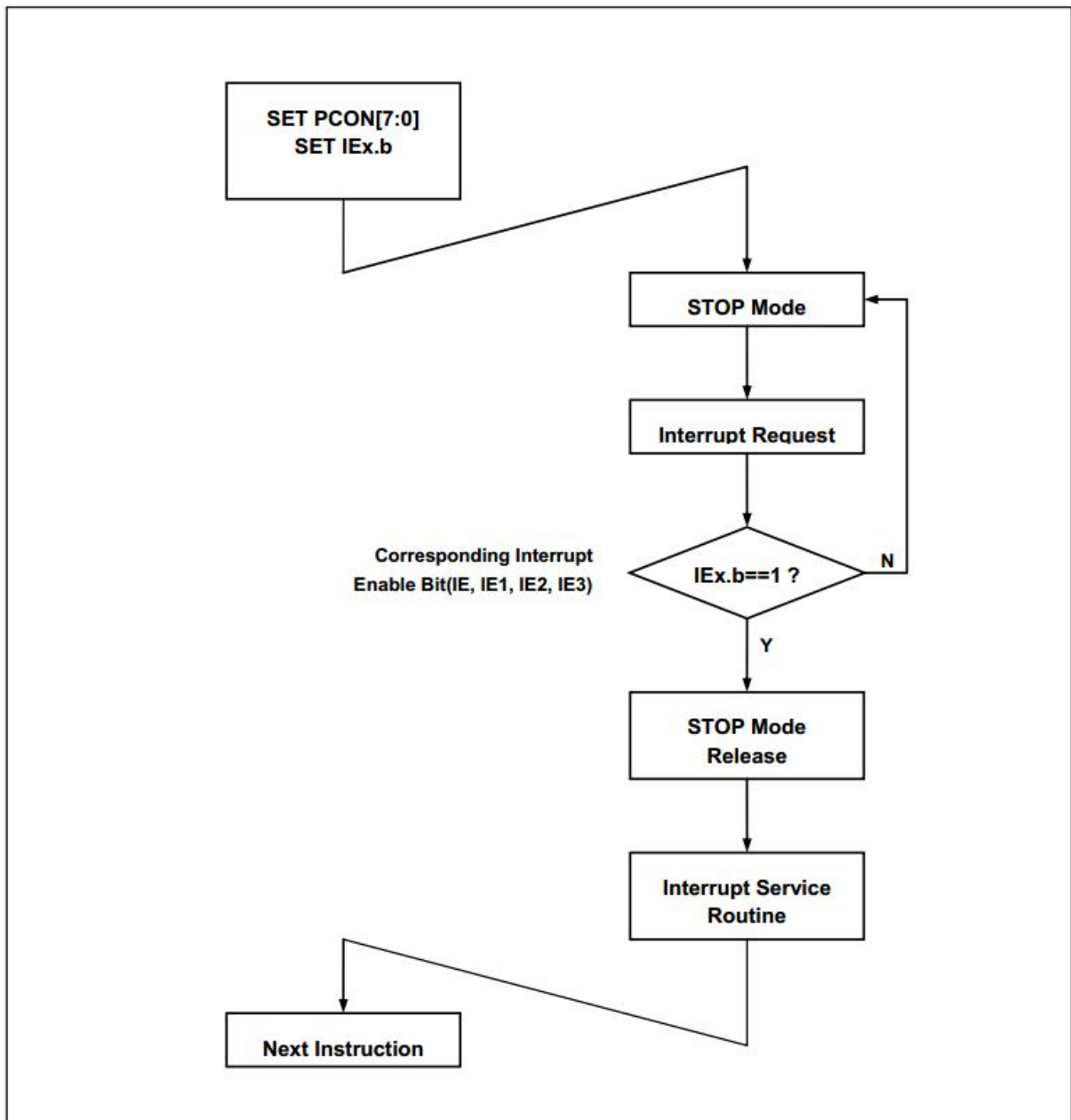
本间隔定时器是激活的，所以在 STOP 指令之前，用户必须设置它的有关分频使其有足够长的时间，这是保证振荡器开始时稳定振荡。



外部中断唤醒 STOP 模式

### 8.5、退出 STOP 模式操作

退出 STOP 模式之后，操作根据进入 STOP 模式前的中断寄存器开始进行。如果总中断使能位 (IE.EA) 设置为 ‘1’，唤醒 STOP 的中断使能位 flag = ‘1’，同时 CPU 跳到相应的中断服务程序。即使 IE.EA 被清除为 ‘0’，唤醒 STOP 的中断的使能位设置为 ‘1’。



STOP 模式唤醒流程

### 8.5.1、寄存器

省电模式操作寄存器

Name	Address	Dir	Default	Description
PCON	87H	R/W	00H	Power Control Register

### 8.5.2、省电操作寄存器说明



省电模式寄存器由电源控制寄存器组成(PCON)。

### 8.5.3、省电操作寄存器说明

**PCON (电源控制寄存器) : 87H**

7	6	5	4	3	2	1	0
PCON7	-	-	-	PCON3	PCON2	PCON1	PCON0
RW	-	-	-	RW	RW	RW	RW

Initial value : 00H

PCON[7:0]            Power Control  
 01H            IDLE 模式使能  
 03H            STOP 模式使能  
 Other Values    常规运行

- 注意)
1. 进入 IDLE 模式, PCON 必须设置为 ‘01H’
  2. 进入 STOP 模式, PCON 必须设置为 ‘03H’
  3. PCON 通过退出 STOP/IDLE 信号寄存器自动清除
  4. 跟随 STOP/IDLE 模式之后必须立即写入三个以上的 NOP 指令

例如:

```
Ex1)  MOV  PCON, #01H ; IDLE mode
      NOP
      NOP
      NOP
      .
      .
      .

Ex2)  MOV  PCON, #03H ; STOP mode
      NOP
      NOP
      NOP
      .
      .
      .
```

## 9、复位

### 9.1、概述

以下是硬件设置值

复位状态

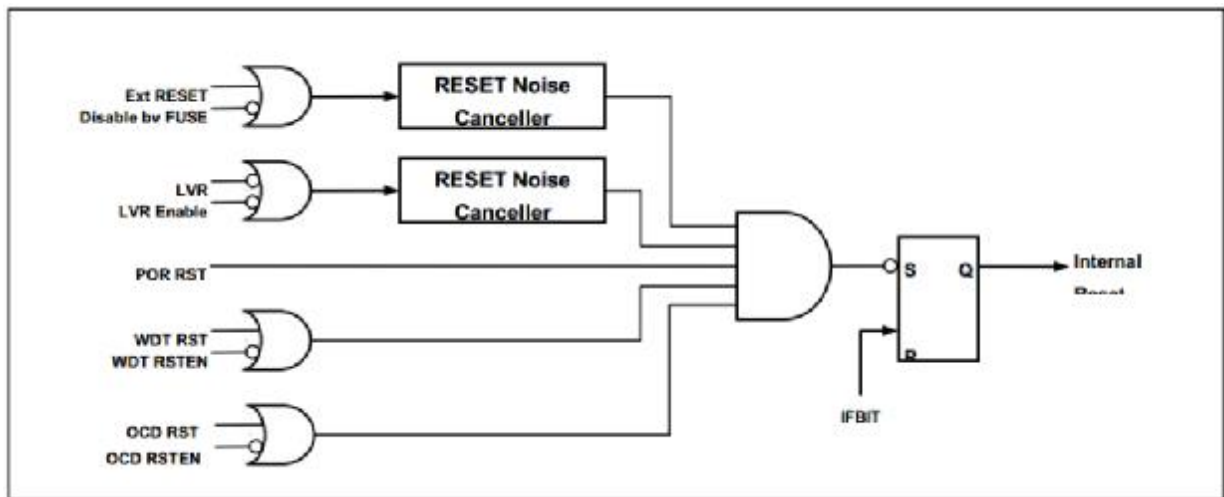
On Chip Hardware	Initial Value
程序计数器 (PC)	0000h
累加器	00h
堆栈指针 (SP)	07h
外围时钟	On
控制寄存器	参考外围寄存器

### 9.2、复位源

AiP81F2016 有 5 种类型的复位源。以下是复位源：

- 外部复位
- 上电复位(POR)
- WDT 溢出复位(WDTEN = ‘1’)
- 低电压复位(LVREN = ‘0’)
- OCD 复位

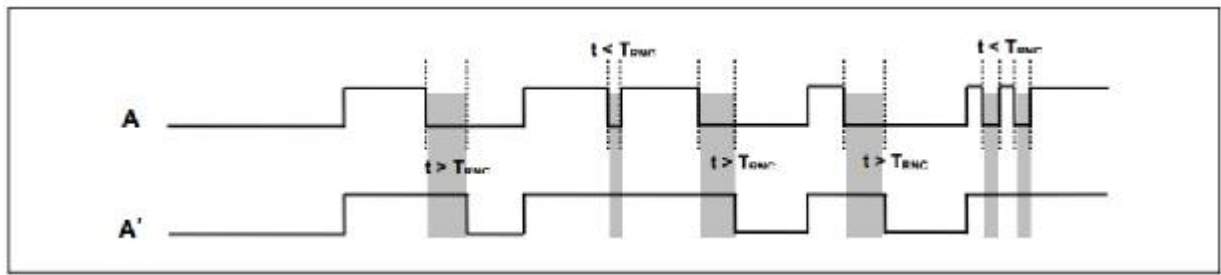
### 9.3、RESET 方框图



复位方框图

### 9.4、RESET 噪音消除器

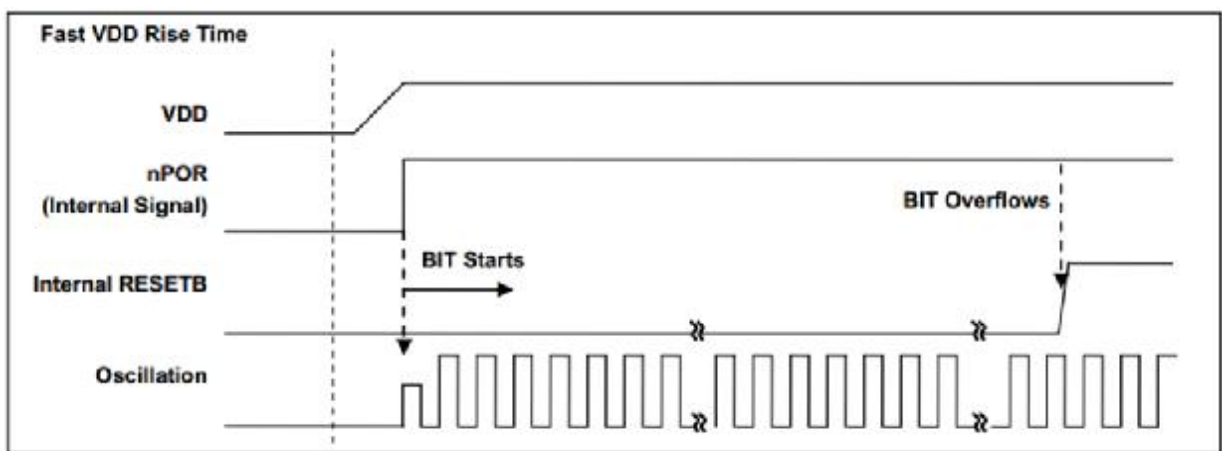
下图是噪音消除器复位噪音消除图解。噪音消除值大概 2us(@VDD=5V)到系统复位输入低时。



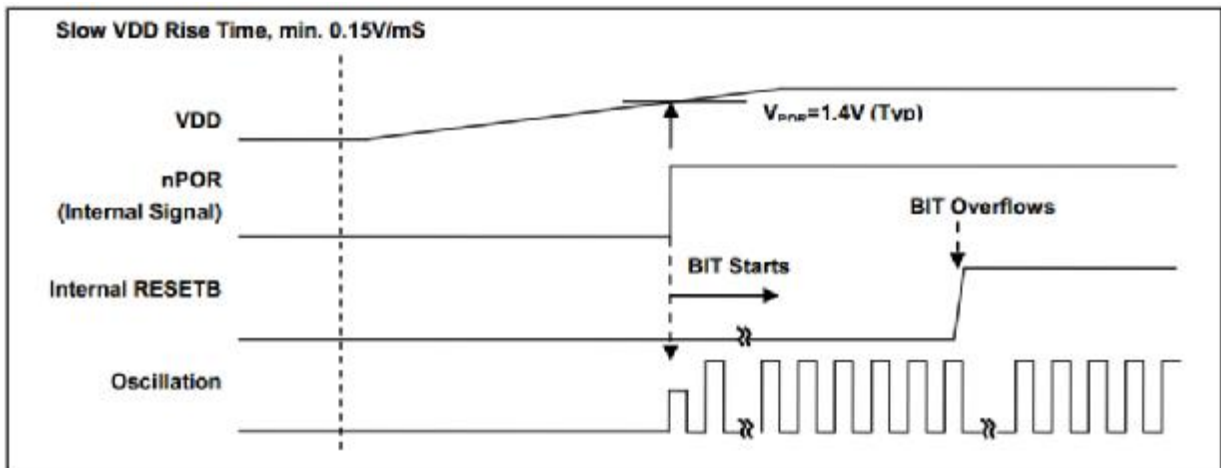
复位噪音消除器时序框图

### 9.5、上电复位

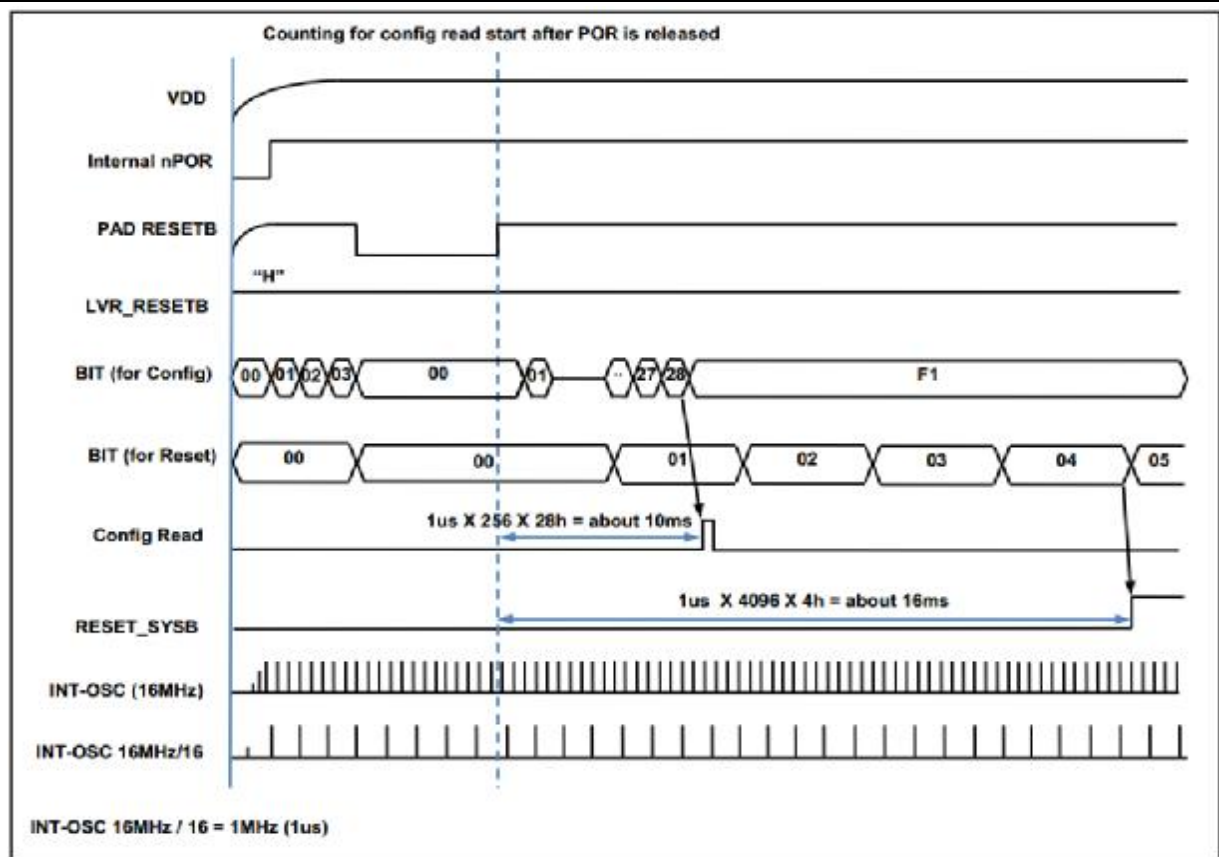
电压上升时, POR(Power On Reset)有复位功能. 如果 POR 被用到, 执行设备复位功能而不是 RESET IC 或 RESET circuits。



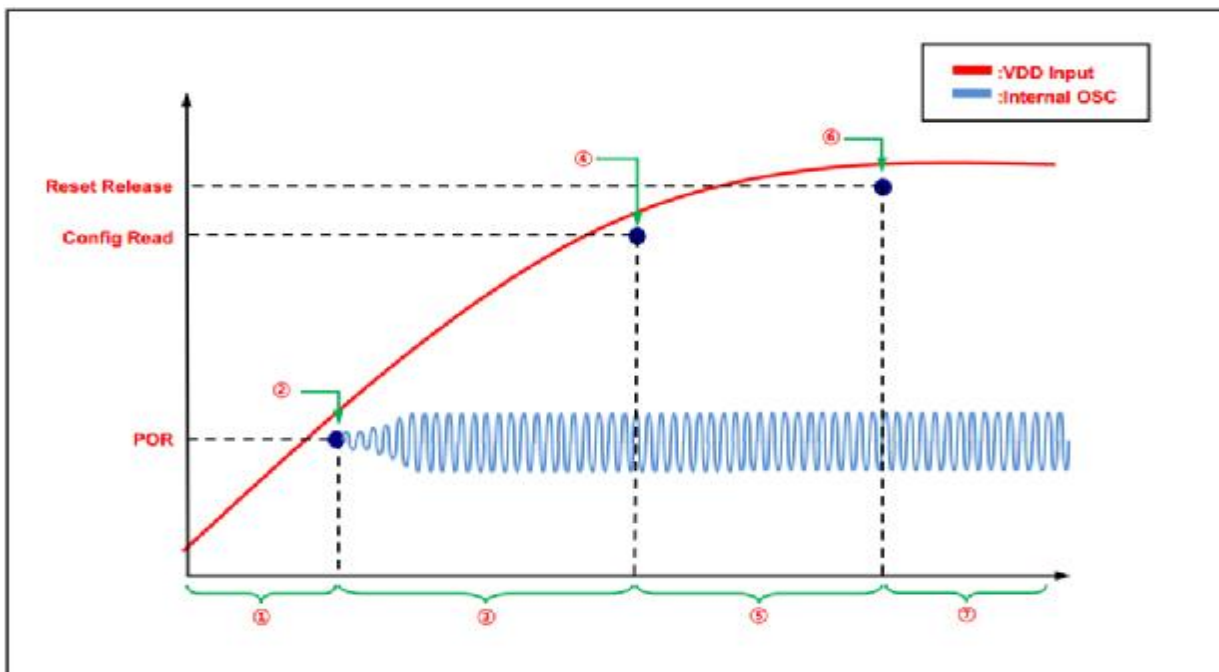
快速 VDD 上升时序



电压升高内部复位唤醒时序



上电组态时序



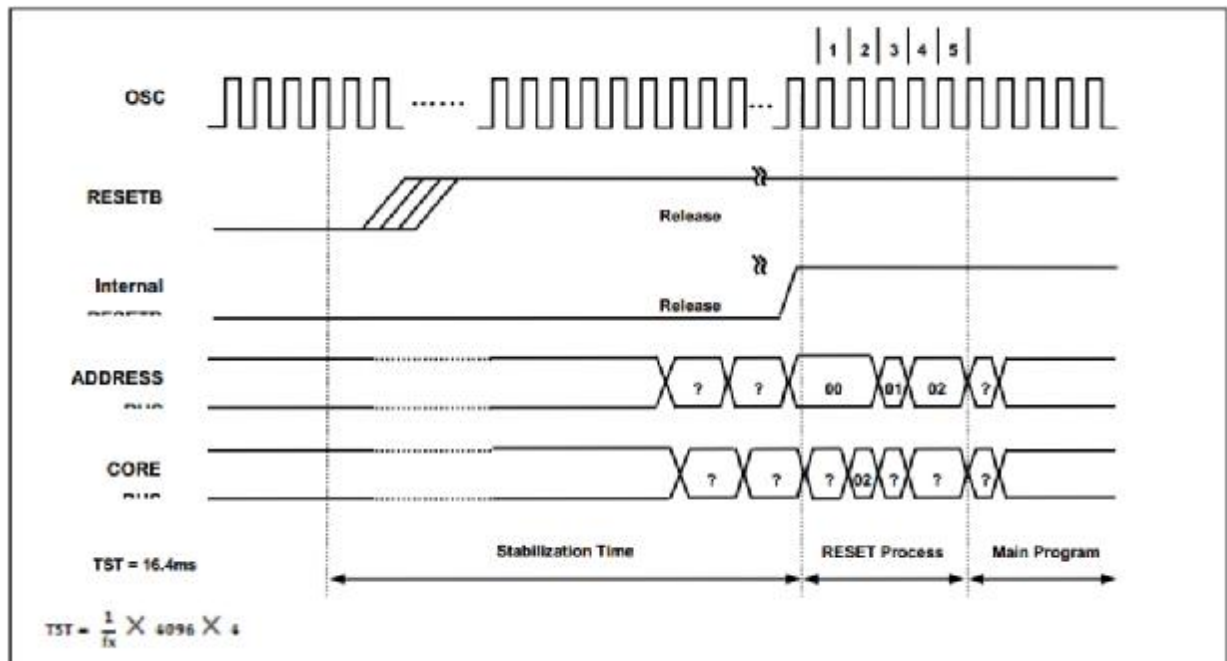
启动过程波形图

启动过程说明

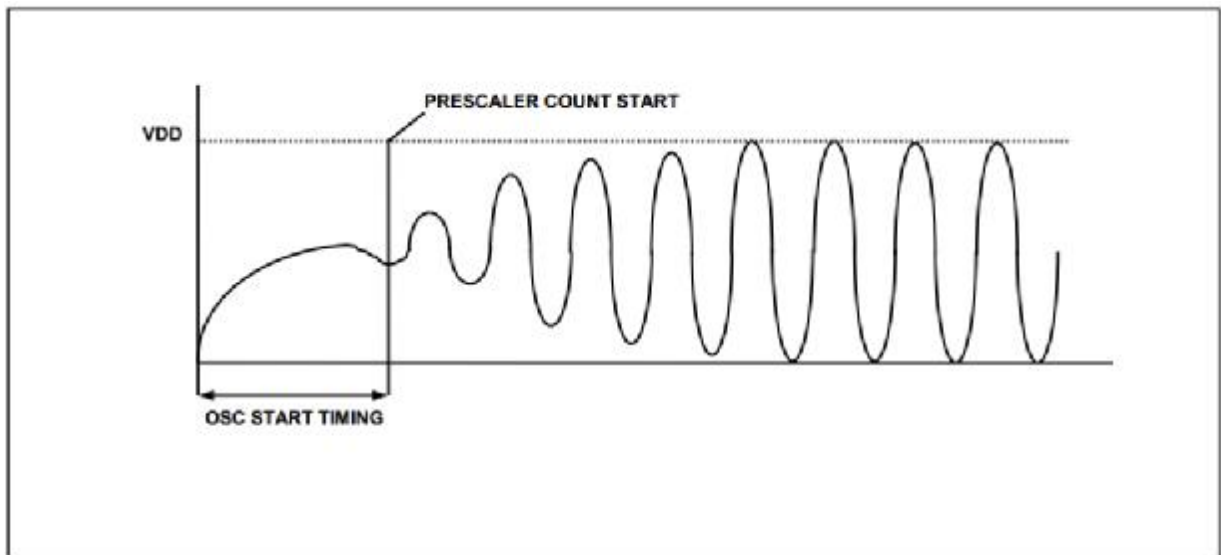
Process	Description	Remarks
①	-No Operation	
②	-1st POR level Detection	-about 1.4V
③	- (INT-OSC 16MHz/16)x256x28h Delay section (=10ms) -VDD input voltage must rise over than flash operating voltage for Config read	-Slew Rate $\geq$ 0.15V/ms
④	- Config read point	-about 1.5V ~ 1.6V -Config Value is determined by Writing Option
⑤	- Rising section to Reset Release Level	-16ms point after POR or Ext_reset release
⑥	- Reset Release section (BIT overflow) i) after16ms, after External Reset Release (External reset) ii) 16ms point after POR (POR only)	- BIT is used for Peripheral stability
⑦	-Normal operation	

### 9.6、外部复位

外部复位是施密特触发器输入。如果在操作电压范围及稳定振荡情况下 RESETB 脚保持低电平 10us 以上，就会请求复位，内部状态被初始化。复位状态变为 ‘1’ 后，需要 16ms 的稳定时间，内部 RESET 变为 ‘1’。复位过程需要 5 个振荡时钟。程序在向量地址 0000H 处开始。



RESET 后时序图



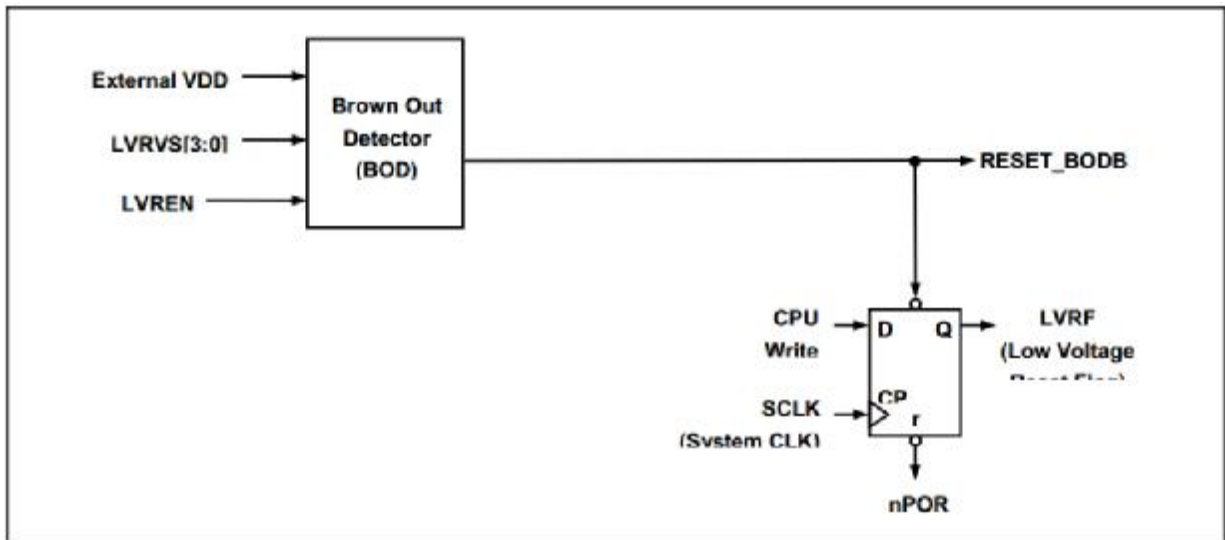
振荡器产生波形举例

注意) 如图所示, 稳定时间不包括启动时间。

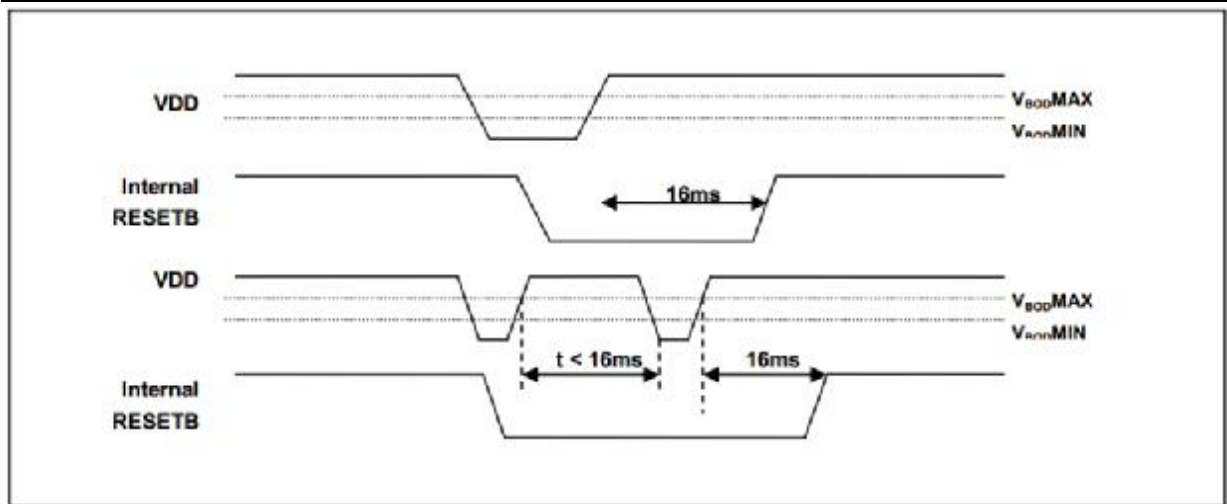
RESETB 引脚有上拉寄存器。

### 9.7、BOD 处理器

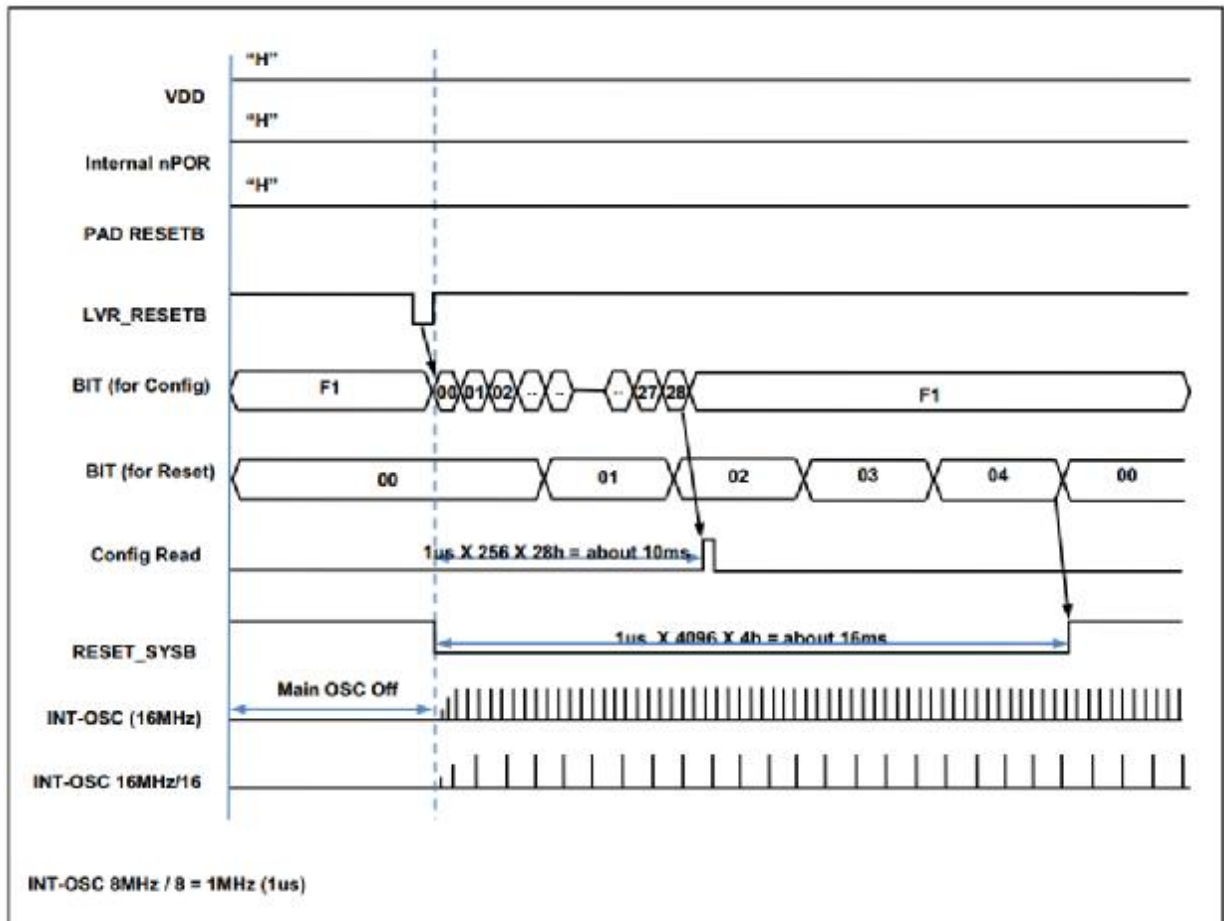
YF8316 16/32 有片内 brown-out detection circuit(BOD)通过和固定的触发电平值相比较来监测 VDD 电平。触发电平可以通过 LVRVS[3:0]设置为 1.60V, 2.00V, 2.10V, 2.20V, 2.32V, 2.44V, 2.59V, 2.75V, 2.93V, 3.14V, 3.38V, 3.67V, 4.00V, 4.40V。在 STOP 模式时, 它将会加大整个电流消耗, 所以为了减小电流消耗, LVREN 位被软件设定。



BOD 方框图

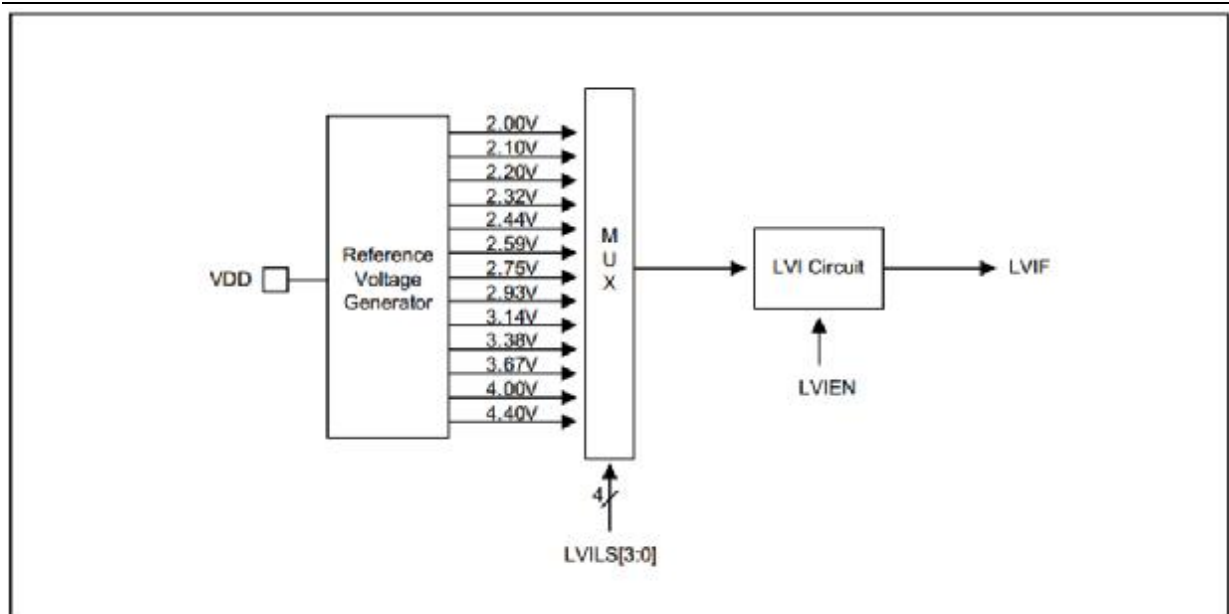


电源故障时内部复位



BOD RESET 时配置时序

9.8、LVI 方框图



LVI 框图

9.8.1、寄存器

复位操作寄存器

Name	Address	Dir	Default	Description
RSTFR	E8H	R/W	80H	Reset Flag Register
LVRCR	D8H	R/W	00H	Low Voltage Reset Control Register
LVICR	86H	R/W	00H	Low Voltage Indicator Control Register

9.8.2、复位操作寄存器说明

复位控制寄存器由(RSTFR), (LVRCR)和(LVICR)组成。

9.8.3、复位操作寄存器说明

RSTFR (复位标志寄存器) : E8H

7	6	5	4	3	2	1	0
PORF	EXTRF	WDTRF	OCDRF	LVRF	-	-	-
RW	RW	RW	RW	RW	-	-	-

Initial value : 80H

PORF 上电复位标志，写入 ‘0’ 清除该位

0 没有检测

1 检测

EXTRF 外部复位(RESETB)标志位，写入 0 或通过上电复位对该位进行复位

0 没有检测

1 检测

WDTRF 看门狗复位标志位，该位通过写入 0 或上电复位清除

0 没有检测



	1	检测
OCDRF	OCD 复位标志位, 该位通过写入 0 或上电复位清除	
	0	没有检测
	1	检测
LVRF	低电压复位标志位, 该位通过写入 0 或上电复位清除	
	0	没有检测
	1	检测

- 注意)
1. 上电复位发生时, PORF 只唯一是“1”的, 其他标志位(WDTRF 和 OCDRF)都清零
  2. 上电复位发生时, EXTRF 是未知的, 此时, EXTRF 可以被设置“1”当外部复位时(RESETB)。
  3. 上电复位发生时, LVRF 是未知的, 此时, LVRF 可以被设置“1”当 LVR 复位发生时。
  4. 除了 POR 之外的复位发生时, 相应的标志位为“1”, 其他标志位保持先前值。

**LVRCR (低电压复位控制寄存器) : D8H**

7	6	5	4	3	2	1	0
LVRST	-	-	LVRVS3	LVRVS2	LVRVS1	LVRVS0	LVREN
RW	-	-	RW	RW	RW	RW	RW

Initial value : 00H

LVRST	当退出 Stop 时 LVR 使能
	0 退出睡眠模式时不影响
	1 退出睡眠模式时 LVR 使能

注意)

当该位是 ‘1’, LVREN 位被退出睡眠信号清零(LVR 使能)

当该位是 ‘0’, LVREN 位对唤醒睡眠信号没反应

LVRVS[3:0] LVR 电压选择

LVRVS3	LVRVS2	LVRVS1	LVRVS0	Description
0	0	0	0	1.60V
0	0	0	1	2.00V
0	0	1	0	2.10V
0	0	1	1	2.20V
0	1	0	0	2.32V
0	1	0	1	2.44V
0	1	1	0	2.59V
0	1	1	1	2.75V
1	0	0	0	2.93V
1	0	0	1	3.14V
1	0	1	0	3.38V
1	0	1	1	3.67V
1	1	0	0	4.00V
1	1	0	1	4.40V

其他值 不可用

LVREN	LVR 操作
	0 LVR 使能

1 LVR 禁止

- 注意) 1. LVRVS[3:0]位被上电复位清零但是其他复位时仍保留  
 2. LVRVS[3:0]位应该设置为‘0000b’当LVREN = ‘1’

**LVICR (低电压指示控制寄存器) : 86H**

7	6	5	4	3	2	1	0
-	-	LVF	LVIEN	LVILS3	LVILS2	LVILS1	LVILS0
-	-	RW	RW	RW	RW	RW	RW

Initial value : 00H

LVIF 低电压指示标志位

- 0 不检测
- 1 检测

LVIEN LVI 使能/禁止

- 0 禁止
- 1 使能

LVILS[3:0] LVI 电平选择

LVILS3	LVILS2	LVILS1	LVILS0	Description
0	0	0	0	2.00V
0	0	0	1	2.10V
0	0	1	0	2.20V
0	0	1	1	2.32V
0	1	0	0	2.44V
0	1	0	1	2.59V
0	1	1	0	2.75V
0	1	1	1	2.93V
1	0	0	0	3.14V
1	0	0	1	3.38V
1	0	1	0	3.67V
1	0	1	1	4.00V
1	1	0	0	4.40V
其他值	其他值	其他值	其他值	无效

## 10、片内调试系统

### 10.1、概述

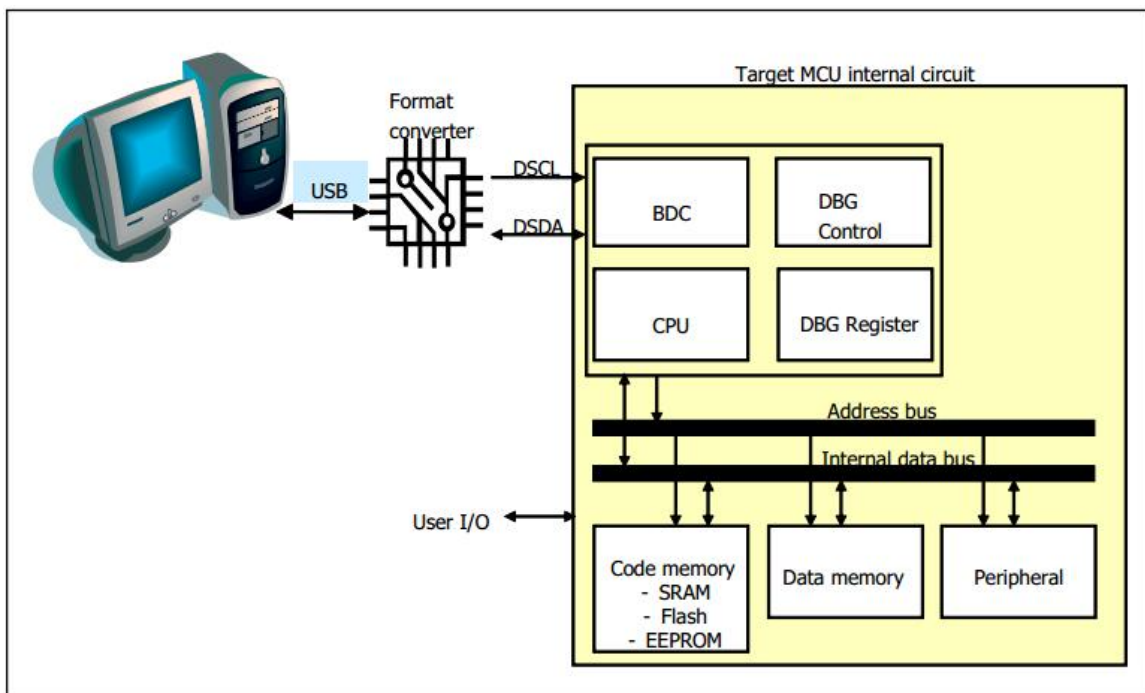
#### 10.1.1、说明

片内调试系统(OCD)可用于在非易失存储器上编程和片内调试。具体可参考以下内容。

如下所示为 OCD 接口和片内调试系统的方框图。

### 10.1.2、特性

- Ø 两根外部接口线:一根是时钟输入, 一根是串行数据总线。
- Ø 调试访问
  - 所有的内部外围模块
  - 内部数据 RAM
  - 程序计数器
  - Flash 和数据 EEPROM
- Ø 片内调试支持中断,包括
  - 中断指令
  - 单步中断
  - 程序存储器信号地址断点
  - 通过两根总线对 Flash,EEPROM,Fuses,和锁存位进行编程
  - Dr.Choice®支持片内调试
- Ø 工作频率
  - 支持目标 MCU 的最大频率



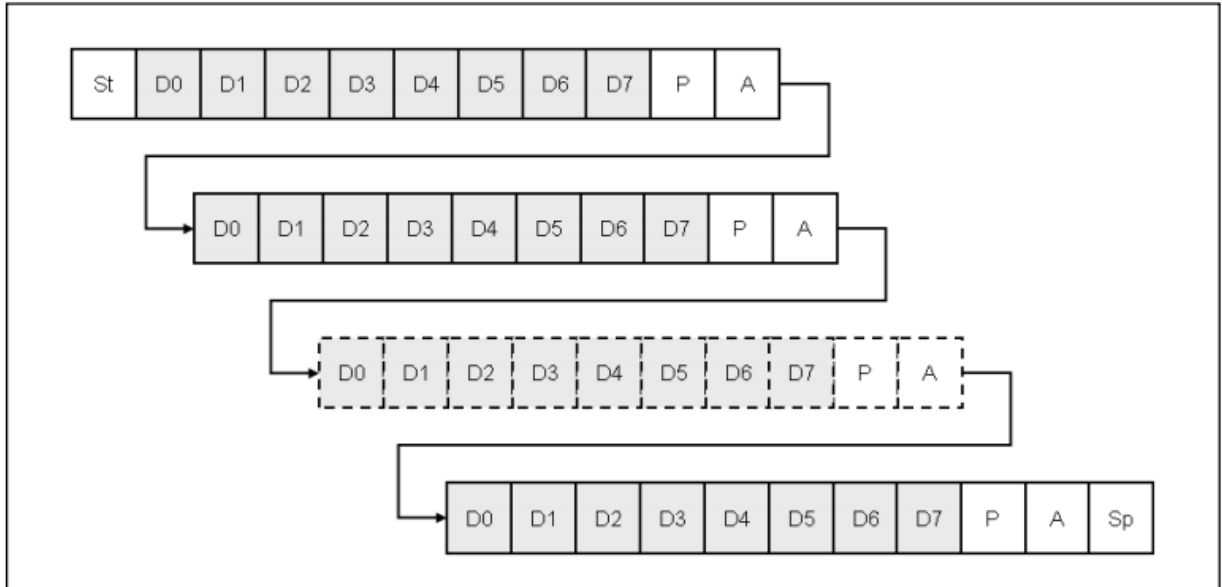
片内调试系统方框图

## 10.2、双线外围接口

### 10.2.1、基本传输包

- Ø 10-bit 传输包用于双线接口。
- Ø 个传输包包括 8 位数据, 1 位校验和 1 位应答

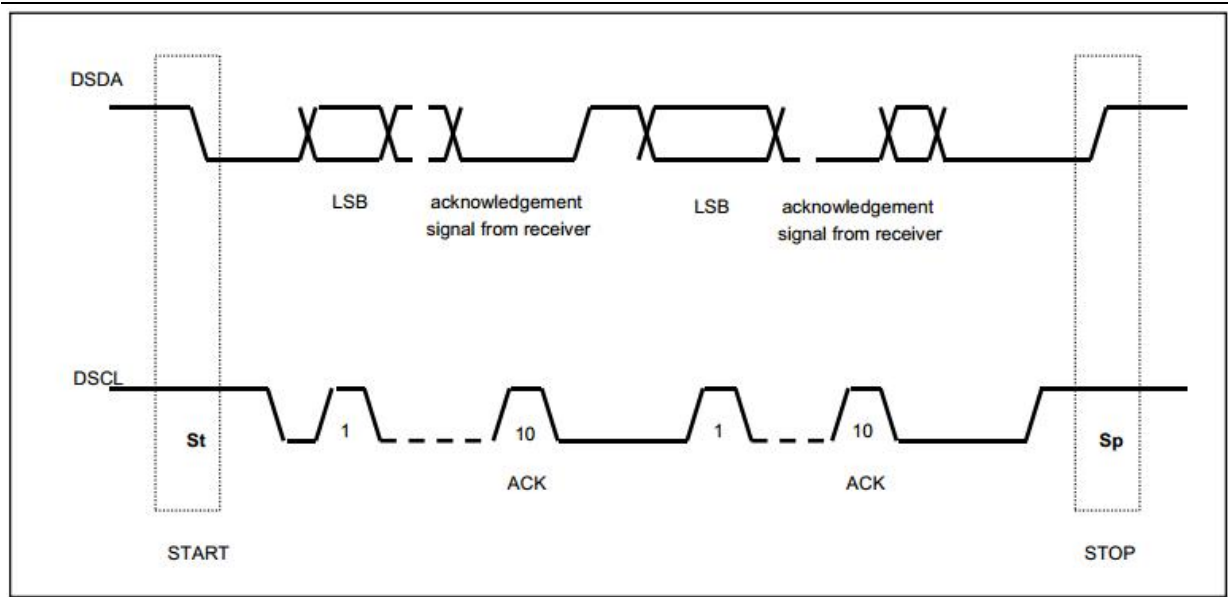
- Ø 8 位数据传送时校验位是 1。
- Ø 8 位数据传送和校验位无误时接收器产生应答 ‘0’。
- Ø 发送器没有确认(应答位是 ‘1’ ), 错误处理在发生器中执行。
- Ø 产生应答错误时, 主机产生停止命令。
- Ø 后台调试器由一系列数据包组成。
- Ø Start 和 stop 命令通知后台调试器开始或停止。



10-bit 传输数据包

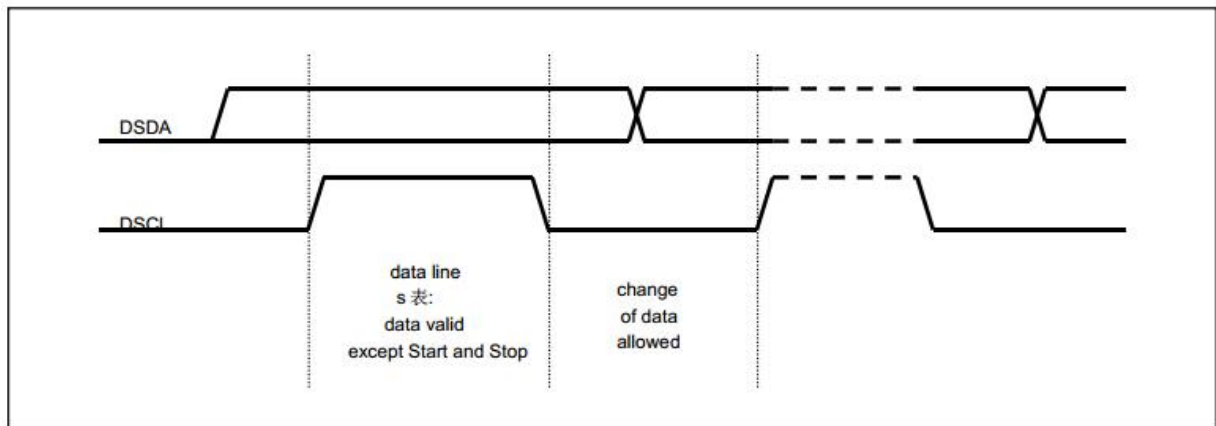
## 10.2.2、数据包传送时序

### 10.2.2.1、数据传送



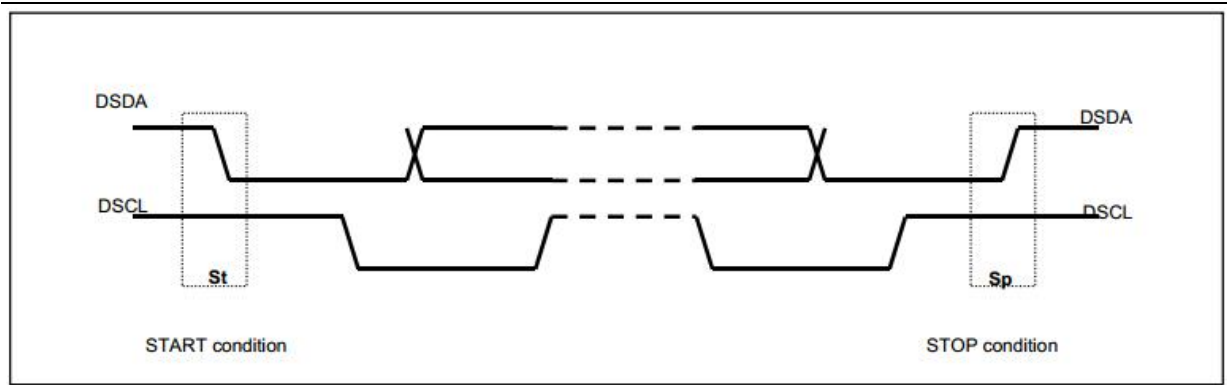
两条总线上的数据传输

### 10.2.2.2、位传送



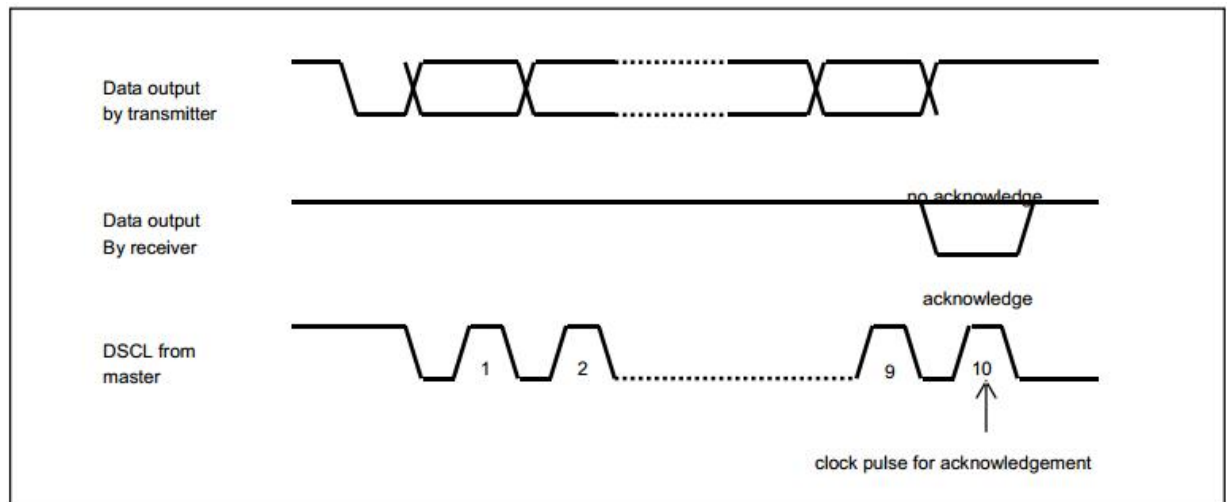
串行总线位传输

### 10.2.2.3、开始和停止信号

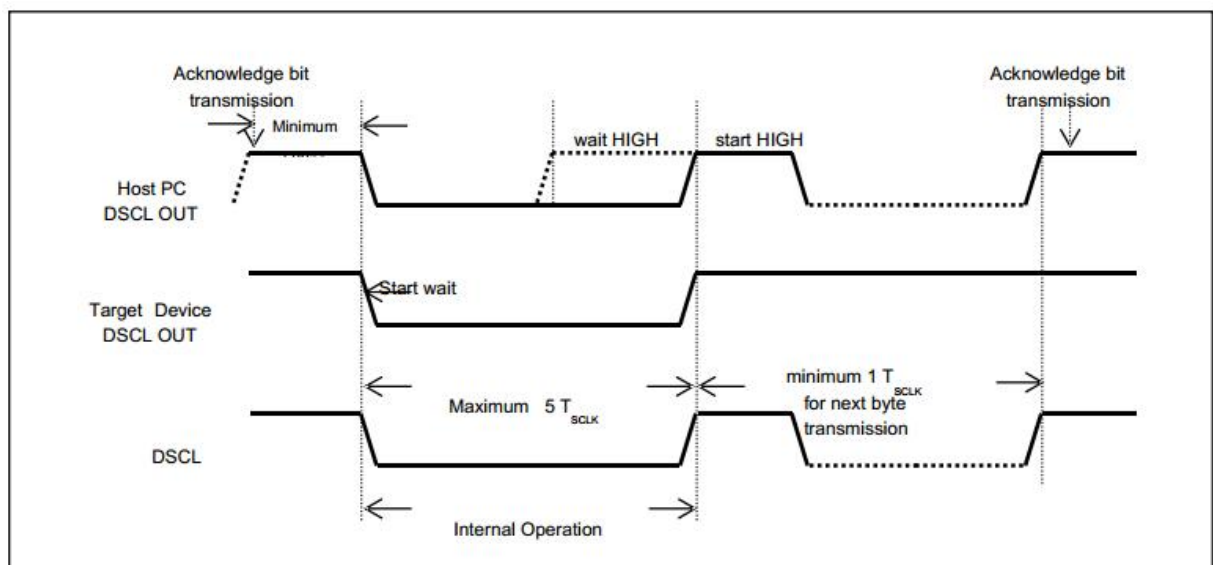


开始和停止信号

### 10.2.2.4、应答位



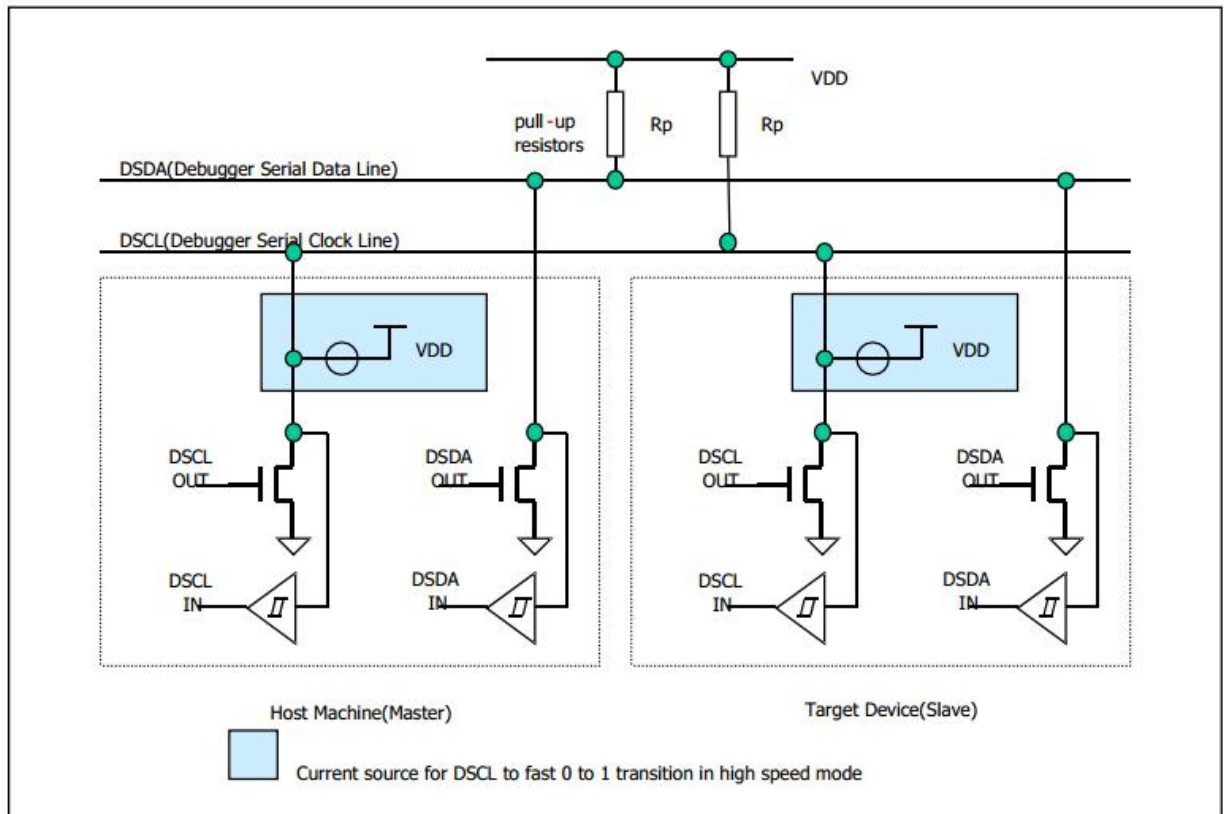
串行总线应答



等待过程中的时钟同步

### 10.2.3、传输线路

双脚接口使用漏极开路线路(线与双向 I/O)。



传输线路

## 11、存储器

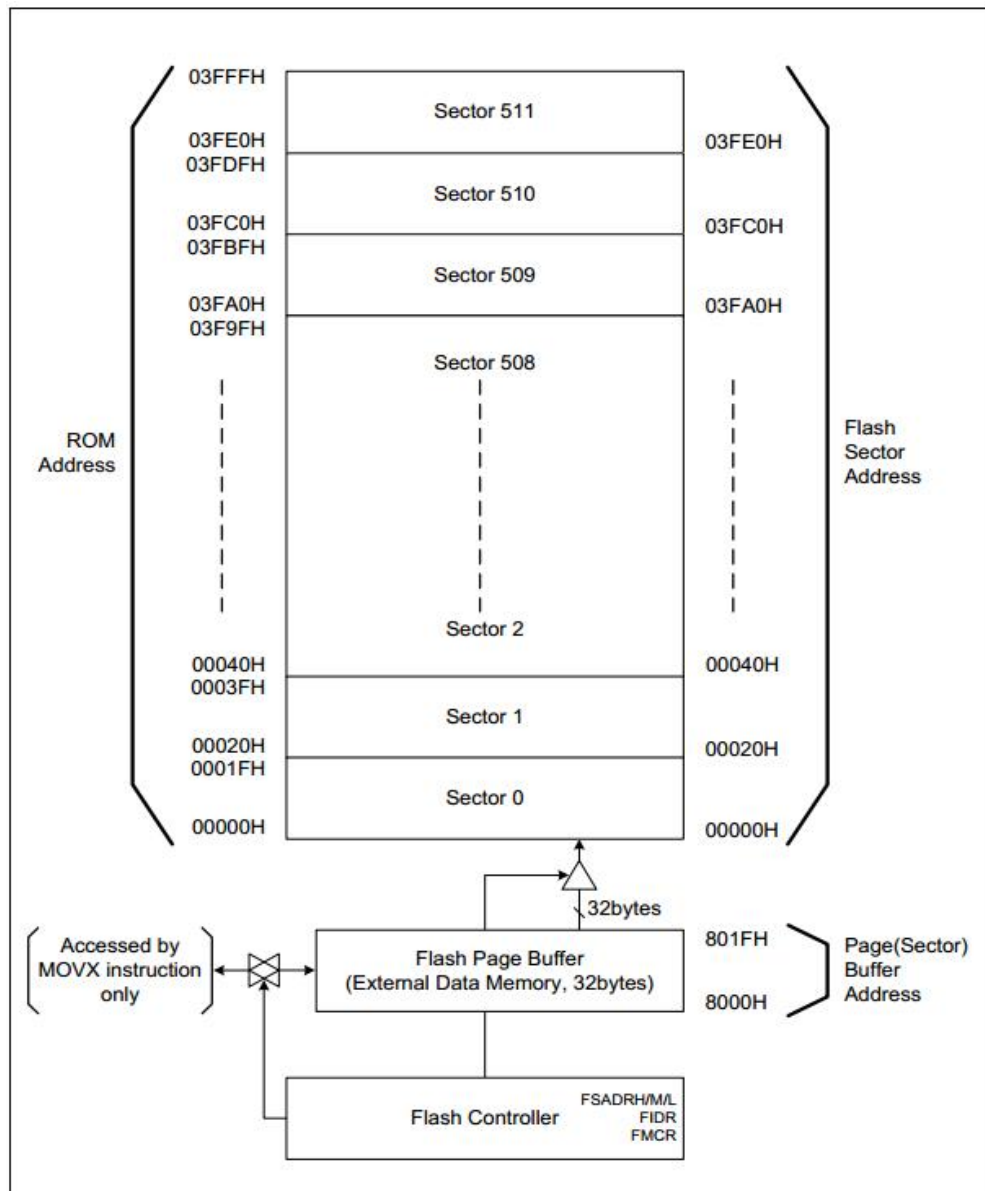
### 11.1、概述

### 11.1.1、说明

YF8316 16/32 集成了 flash 存储器，可写入、擦除，和覆盖写入。flash 存储器可以通过“MOVC”指令来读取并且可以在 OCD 上进行编程，支持串行 ISP 模式或者用户编程模式。

- Ø FlashSize:16k bytes
- Ø 电源信号支持编程和擦除
- Ø 命令接口快速编程和擦除操作
- Ø 典型温度和电压下，高达 100,000 编程/擦除

### 11.1.2、Flash ROM 结构



Flash ROM 结构

### 11.1.3、寄存器

#### Flash Memory 寄存器



Name	Address	Dir	Default	Description
FSADRH	FAH	R/W	00H	Flash Sector Address High Register
FSADRM	FBH	R/W	00H	Flash Sector Address Middle Register
FSADRL	FCH	R/W	00H	Flash Sector Address Low Register
FIDR	FDH	R/W	00H	Flash Identification Register
FMCR	FEH	R/W	00H	Flash Mode Control Register

### 11.1.4、Flash Memory 寄存器控制和状态说明

Flash 控制寄存器由(FSADRH)，(FSADRM)，(FSADRL)，(FIDR)，和(FMCR)组成。映射到 SFR 区域，只有在编程模式时可以进入。

### 11.1.5、Flash 寄存器说明

#### FSADRH (Flash 页地址寄存器 H) : FAH

7	6	5	4	3	2	1	0
-	-	-	-	FSADRH3	FSADRH2	FSADRH1	FSADRH0
-	-	-	-	R/W	R/W	R/W	R/W

Initial value : 00H

FSADRH[3:0] Flash 页地址寄存器 H

#### FSADRM (Flash 页地址寄存器 M) : FBH

7	6	5	4	3	2	1	0
FSADRM7	FSADRM6	FSADRM5	FSADRM4	FSADRM3	FSADRM2	FSADRM1	FSADRM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FSADRM[7:0] Flash 页地址寄存器 M

#### FSADRL (Flash 页地址寄存器 L) : FCH

7	6	5	4	3	2	1	0
FSADRL7	FSADRL6	FSADRL5	FSADRL4	FSADRL3	FSADRL2	FSADRL1	FSADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FSADRL[7:0] Flash 页地址寄存器 L

#### FIDR (Flash 识别寄存器) : FDH

7	6	5	4	3	2	1	0
FIDR7	FIDR6	FIDR5	FIDR4	FIDR3	FIDR2	FIDR1	FIDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FIDR[7:0] Flash 识别

Others 其他值不识别

10100101 flash 模式识别数值

一次操作后除了“flash 页缓存器复位模式”之外，这些位立即被逻辑值‘00H’自动清零

FMCR (Flash 模式控制寄存器) : FEH							
7	6	5	4	3	2	1	0
FMBUSY	-	-	-	-	FMCR2	FMCR1	FMCR0
R	-	-	-	-	RW	RW	RW

Initial value : 00H

FMBUSY Flash 模式忙碌位，该位只有在调试时使用  
 0 写入“1”没有作用  
 1 忙碌

FMCR[2:0] Flash 模式控制位，操作一种 flash 模式时，CPU 保持工作，总中断在禁止状态而不管 IE.7 (EA)位是什么

FMCR2	FMCR1	FMCR0	Description
0	0	1	选择 flash 页缓存器复位模式并开始启动而不 FID 的值 (清除所有 1k 字节)
0	1	0	选择 flash 页擦除模式并开始启动 FIDR=“10100101b”
0	1	1	选择 flash 页写入模式并开始启动 FIDR=“10100101b”
1	0	0	选择 flash 页锁定模式并开始启动 FIDR=“10100101b”

Others Values: No operation  
 (一次操作之后这些位立即被逻辑“00H”自动清除)

注意) 例 1: YF8316 16/32 指令擦除操作: YF8316 16/32 只有页擦除模式，并且一页是 1K 字节。

```
MOV    FMCR, #0X01    //复位跟 FALSH 操作相关的寄存器
NOP
NOP
NOP
```

```
MOV    FSADRH, #0X00 //选择要擦除的第 15 页地址
MOV    FSADRM, #0X3C
MOV    FSADRL, #0X00
MOV    FIDR, #0XA5   //校验码
MOV    FMCR, #0X02   //启动擦除模式
NOP
NOP
NOP
```

之后进行擦除校验需对整页（1K 字节）校验

例 2: YF8316 16/32 指令写入操作: YF8316 16/32 只有单字节写入模式，即执行一次操作只能写入一个字节。

```
MOV    FMCR, #0X01    //复位跟 FLASH 操作相关的寄存器
NOP
NOP
NOP
```

```
MOV    A, #5
MOV    DPH, #0x80
MOV    DPL, #00
MOV    @DPTR, A        //将数据写入缓存寄存器
```

```

MOV    FSADRH, #0X00    //选择要写入数据的地址 3CA0H
MOV    FSADRM, #0X3C
MOV    FSADRL, #0XA0
MOV    FIDR, #0XA5      //校验码
MOV    FMCR, #0X03      //启动写入模式
NOP
NOP
NOP

MOV    A, #0            //写校验
MOV    R1, #5
MOV    DPH, #0x3C
MOV    DPL, #0XA0
MOVC   A,@A+DPTR
SUBB   A,R1
JNZ    Verify_error

```

### 11.1.6、串行 ISP 模式

串行 ISP 使用两根线作为接口。具体参考第 10 章。

### 11.1.7、保护区(用户编程模式)

YF8316 16/32 可对它本身的 flash 存储器进行编程(保护区)。保护区是不可以被擦除的也不能烧录的。保护区只有在 PAEN 位清零时可供使用。就是说, 如果需要使能第 2 个配置项的保护区的话, 如果该区域没有使能 (PAEN = '1'), 该区域被用于正常的程序存储器。

保护区的大小是可以通过配置的第二个选项改变的。

保护区大小

Protection Area Size Select		Size of Protection Area	Address of Protection Area
PASS1	PASS0		
0	0	3.8k Bytes	0100H – 0FFFH
0	1	1.7k Bytes	0100H – 07FFH
1	0	768 Bytes	0100H – 03FFH
1	1	256 Bytes	0100H – 01FFH

注意) 参考第 16 章配置控制选项.

### 11.1.8、擦除模式

用户编程模式时块擦除编程步骤:

1. 清除页缓存器(FMCR=0x01)
2. 页缓存器写入 '0'
3. 设置块擦除地址寄存器(FSADRH/FSADRM/FSADRL)
4. 设置 flash 识别寄存器(FIDR)
5. 设置 flash 模式控制寄存器(FMCR)
6. 擦除校验

#### Program Tip – sector erase

```
MOV    FMCR,#0x01    ;page buffer clear
```

---

```

NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.

MOV     A,#0
MOV     R0,#32                    ;Sector size is 32bytes
MOV     DPH,#0x80
MOV     DPL,#0

Pgbuf_clr: MOVX   @DPTR,A
          INC     DPTR
          DJNZ   R0, Pgbuf_clr      ;Write '0' to all page buffer

MOV     FSADRH,#0x00
MOV     FSADRM,#0x3F
MOV     FSADRL,#0xA0            ;Select sector 509
MOV     FIDR,#0xA5             ;Identification value
MOV     FMCR,#0x02             ;Start flash erase mode
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.

MOV     A,#0                      ;erase verify
MOV     R0,#32                    ;Sector size is 32bytes
MOV     R1,#0
MOV     DPH,#0x3F
MOV     DPL,#0xA0

Erase_verify:
          MOVC   A,@A+DPTR
          SUBB   A,R1
          JNZ   Verify_error
          INC   DPTR
          DJNZ  R0, Erase_verify

Verify_error:

```

**字节擦除编程步骤:**

1. 清除页缓存器(FMCR=0x01)
2. 页缓存器写入 '0'

3. 设置擦除地址寄存器(FSADRH/FSADRM/FSADRL)
4. 设置 flash 识别寄存器(FIDR)
5. 设置 flash 模式控制寄存器(FMCR)
6. 校验

**Program Tip – byte erase**

```

MOV    FMCR,#0x01      ;page buffer clear
NOP
NOP
NOP
NOP

MOV    A,#0
MOV    DPH,#0x80
MOV    DPL,#0
MOVX   @DPTR,A

MOV    DPH,#0x80
MOV    DPL,#0x05
MOVX   @DPTR,A        ;Write '0' to page buffer

MOV    FSADRH,#0x00
MOV    FSADRM,#0x3F
MOV    FSADRL,#0xA0   ;Select sector 509
MOV    FIDR,#0xA5     ;Identification value
MOV    FMCR,#0x02     ;Start flash erase mode
NOP
NOP
NOP

MOV    A,#0            ;erase verify
MOV    R1,#0
MOV    DPH,#0x3F
MOV    DPL,#0xA0
MOVC   A,@A+DPTR
SUBB   A,R1            ;0x3FA0 = 0 ?
JNZ    Verify_error

MOV    A,#0
MOV    R1,#0
MOV    DPH,#0x3F
MOV    DPL,#0xA5
MOVC   A,@A+DPTR
SUBB   A,R1            ;0x3FA5 = 0 ?
JNZ    Verify_error

```

---

 Verify\_error:

### 11.1.9、写模式

#### 块写模式编程步骤:

1. 清除页缓存器(FMCR=0x01)
2. 写数据到页缓存器
3. 设置地址寄存器(FSADRH/FSADRM/FSADRL)
4. 设置 flash 识别寄存器(FIDR)
5. 设置 flash 模式控制寄存器(FMCR)
6. 校验

#### Program Tip – sector write

```

MOV    FMCR,#0x01      ;page buffer clear
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.

MOV    A,#0
MOV    R0,#32           ;Sector size is 32bytes
MOV    DPH,#0x80
MOV    DPL,#0

Pgbuf_WR:  MOVX   @DPTR,A
INC      A
INC      DPTR
DJNZ    R0, Pgbuf_WR   ;Write data to all page buffer

MOV     FSADRH,#0x00
MOV     FSADRM,#0x3F
MOV     FSADRL,#0xA0   ;Select sector 509
MOV     FIDR,#0xA5     ;Identification value
MOV     FMCR,#0x03     ;Start flash write mode
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.

MOV     A,#0            ;write verify
MOV     R0,#32         ;Sector size is 32bytes
MOV     R1,#0
MOV     DPH,#0x3F
MOV     DPL,#0xA0

```

Write\_verify:

---

```

MOV    A,@A+DPTR
SUBB   A,R1
JNZ    Verify_error
INC    R1
INC    DPTR
DJNZ   R0, Write_verify

```

Verify\_error:

#### 字节写入时编程步骤:

1. 清除页缓存器(FMCR=0x01)
2. 写数据到页缓存器
3. 设置地址寄存器(FSADRH/FSADRM/FSADRL)
4. 设置 flash 识别寄存器(FIDR)
5. 设置 flash 模式控制寄存器(FMCR)
6. Erase verify

#### Program Tip – byte write

```

MOV    FMCR,#0x01      ;page buffer clear
NOP                               ;Dummy instruction, This instruction must be needed.
NOP                               ;Dummy instruction, This instruction must be needed.
NOP                               ;Dummy instruction, This instruction must be needed.

MOV    A,#5
MOV    DPH,#0x80
MOV    DPL,#0
MOVX   @DPTR,A          ;Write data to page buffer

MOV    A,#6
MOV    DPH,#0x80
MOV    DPL,#0x05
MOVX   @DPTR,A          ;Write data to page buffer

MOV    FSADRH,#0x00
MOV    FSADRM,#0x3F
MOV    FSADRL,#0xA0    ;Select sector 509
MOV    FIDR,#0xA5      ;Identification value
MOV    FMCR,#0x03      ;Start flash write mode
NOP                               ;Dummy instruction, This instruction must be needed.
NOP                               ;Dummy instruction, This instruction must be needed.
NOP                               ;Dummy instruction, This instruction must be needed.

MOV    A,#0             ;write verify
MOV    R1,#5

```

```

MOV    DPH,#0x3F
MOV    DPL,#0xA0
MOVC   A,@A+DPTR
SUBB   A,R1                ;0x3FA0 = 5 ?
JNZ    Verify_error

MOV    A,#0
MOV    R1,#6
MOV    DPH,#0x3F
MOV    DPL,#0xA5
MOVC   A,@A+DPTR
SUBB   A,R1                ;0x3FA5 = 6 ?
JNZ    Verify_error

```

Verify\_error:

### 11.1.10、防止错误的擦除和写入 Flash

在擦除和写入 Flash 时需要特别小心。

在 Flash 擦除和写入时，必须防止无效的跳转导致的错误发生，比如干扰和断电。

1.擦除或写入时用户 ID 的校验子程序代码

ErWt\_rtn:

```

---
MOV    FIDR,#10100101B    ;ID Code
MOV    A,#ID_DATA_1      ;Ex) ID_DATA_1:93H,ID_DATA_2:85H,ID_DATA_3:5AH
CJNE   A,UserID1,No_WriteErase
MOV    A,#ID_DATA_2
CJNE   A,UserID2,No_WriteErase
MOV    A,#ID_DATA_3
CJNE   A,UserID3,No_WriteErase
MOV    FMCR,#0x??        ;0x03 if write, 0x02 if erase
---
---
RET

```

No\_WriteErase:

```

MOV    FIDR,#00H
MOV    UserID1,#00H
MOV    UserID2,#00H
MOV    UserID3,#00H
MOV    Flash_flag,#00H
RET

```

按照以上方式编写代码，可以有效避免 Flash 擦除或写入的误操作。

2.使用用户 ID1/2/3 是很重要的。如果 userid1/2/3 是写在指令“MOV FIDR, #10100101b”这条指



令以上，这个保护将失去作用。所以最好将用户 ID1/2/3 写在另一个子程序。

Decide\_ErWt:

```

---
MOV    Flash_flag1,#38H    ;Random value for example, in case of erase/write needs
MOV    FSADRL,#20H        ;Here 20H is example,
MOV    Flash_flag2,#75H
RET

```

3.访问 Flash 的地址(FSADRH/FSADRM/FSADRL)应该被限制在数据地址区。例如：如果定义数据区为 0x3F00~0x3FFF，那么 FSADRH/FSADRM 始终应该为 0x00/0x3f。

4.Overview of main

```

---
CALL   Work1
CALL   Decide_ErWt
CALL   Work2
CALL   ID_write
CALL   Work3
CALL   Flash_erase
CALL   Flash_write
---
---
---

```

ID\_wire:

```

MOV    A,#38H
CJNE   A,Flash_flag1,No_write_ID
MOV    A,#75H
CJNE   A,Flash_flag2,No_write_ID
MOV    UserID1,#ID_DATA_1    ;Write User ID1
MOV    A,#38H
CJNE   A,Flash_flag1,No_write_ID
MOV    A,#75H
CJNE   A,Flash_flag2,No_write_ID
MOV    UserID2,#ID_DATA_2    ;Write User ID2
MOV    A,#38H
CJNE   A,Flash_flag1,No_write_ID
MOV    A,#75H
CJNE   A,Flash_flag2,No_write_ID
MOV    UserID3,#ID_DATA_3    ;Write User ID3
RET

```

No\_write\_ID:

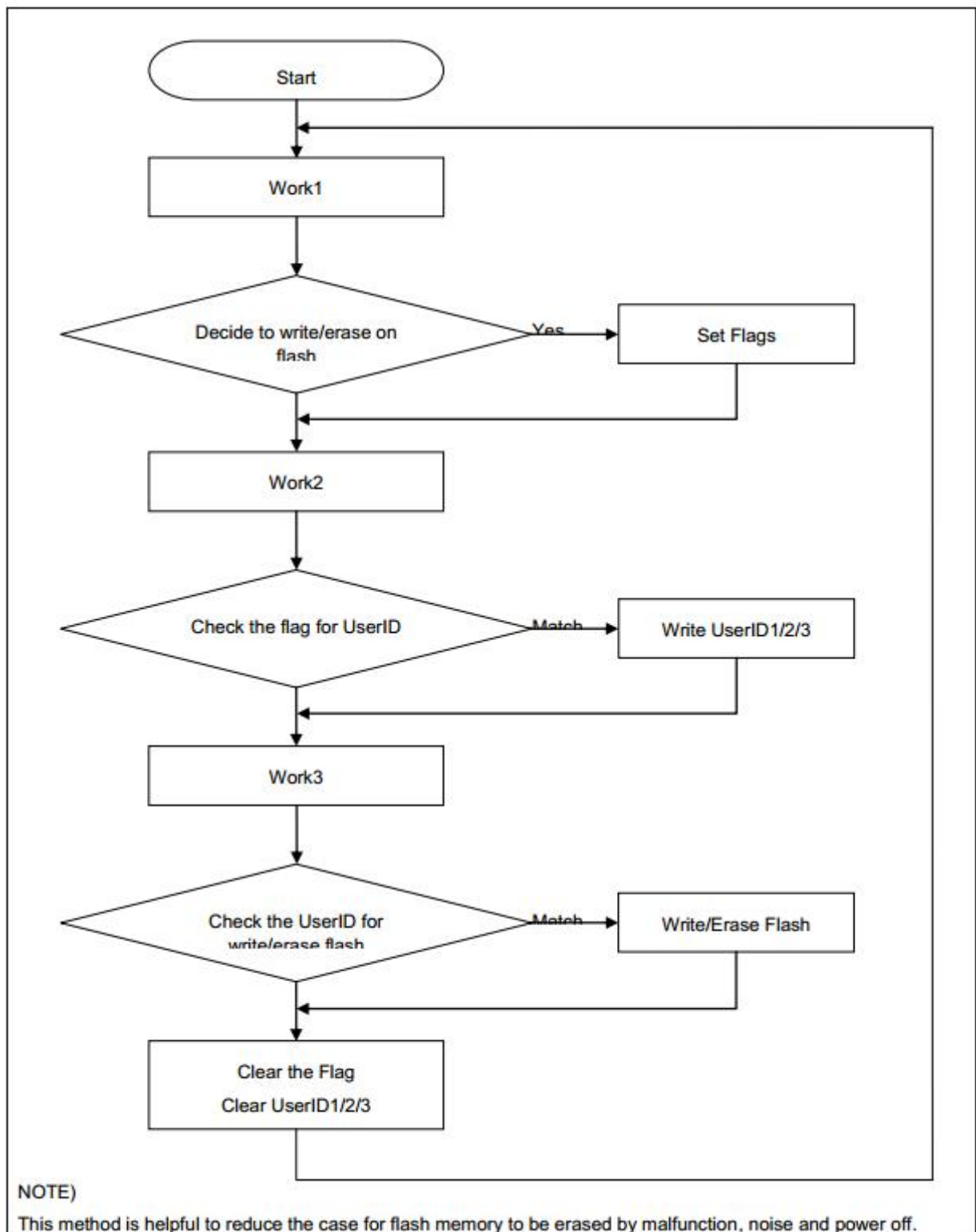
```

MOV    UserID1,#00H
MOV    UserID2,#00H
MOV    UserID3,#00H

```

RET

#### 11.1.10.1、Flow of Protection for Invalid Erase/Write



Flow of Protection for Invalid Erase/Write

11.1.11、读模式

---

读模式编程步骤:

- 1.通过 MOVC 指令间接寻址从存储器加载接收数据

**Program Tip – reading**

```
MOV    A,#0
MOV    DPH,#0x3F
MOV    DPL,#0xA0           ;flash memory address
MOVC   A,@A+DPTR          ;read data from flash memory
```

**11.1.12、硬件锁定模式**

读模式编程步骤

- 1.设置 flash 识别寄存器(FIDR)
- 2.设置 flash 模式控制寄存器(FMCR)

**Program Tip – reading**

```
MOV    FIDR,#0xA5         ;Identification value
MOV    FMCR,#0x04         ;Start flash hard lock mode
NOP                                         ;Dummy instruction, This instruction must be needed.
NOP                                         ;Dummy instruction, This instruction must be needed.
NOP                                         ;Dummy instruction, This instruction must be needed.
```

**12、配置选项**

### 12.1、配置项控制

配置项的数据应该由烧录器写入到配置选项区域(001EH~001FH)。

#### 配置项 1：ROM 地址 001FH

7	6	5	4	3	2	1	0
R_P	HL	-	-	-	-	-	RSTS

Initial value : 00H

R_P	读保护
	0 禁止
	1 使能
HL	硬件锁定
	0 禁止
	1 使能
RSTS	复位脚选择
	0 P32 口
	1 复位脚

#### 配置项 2：ROM 地址 001EH

7	6	5	4	3	2	1	0
-	-	-	-	-	PAEN	PASS1	PASS0

Initial value : 00H

PAEN	保护区使能/禁止
	0 禁止(可通过指令消除)
	1 使能(不可通过指令消除)
PASS [1:0]	保护区大小选择 注意)
	1. 当 PAEN = ‘1’ 时，以下设置才有用
PASS1	PASS0 说明
0	0 3.8k Bytes (Address 0100H – 0FFFH)
0	1 1.7k Bytes (Address 0100H – 07FFH)
1	0 768 Bytes (Address 0100H – 03FFH)
1	1 256 Bytes (Address 0100H – 01FFH)

## 13、附录

### A.指令表

以下列出的指令中“Bytes”可能是1, 2或3个字节长度。

下表所列的每个指令需要1, 2或4个机器周期.1个机器周期包含两个时钟周期。

ARITHMETIC				
Mnemonic	Description	Bytes	Cycles	Hex code
ADD A,Rn	Add register to A	1	1	28-2F
ADD A,dir	Add direct byte to A	2	1	25
ADD A,@Ri	Add indirect memory to A	1	1	26-27
ADD A,#data	Add immediate to A	2	1	24
ADDC A,Rn	Add register to A with carry	1	1	38-3F
ADDC A,dir	Add direct byte to A with carry	2	1	35
ADDC A,@Ri	Add indirect memory to A with carry	1	1	36-37
ADDC A,#data	Add immediate to A with carry	2	1	34
SUBB A,Rn	Subtract register from A with borrow	1	1	98-9F
SUBB A,dir	Subtract direct byte from A with borrow	2	1	95
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	1	96-97
SUBB A,#data	Subtract immediate from A with borrow	2	1	94
INC A	Increment A	1	1	04
INC Rn	Increment register	1	1	08-0F
INC dir	Increment direct byte	2	1	05
INC @Ri	Increment indirect memory	1	1	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	1	18-1F
DEC dir	Decrement direct byte	2	1	15
DEC @Ri	Decrement indirect memory	1	1	16-17
INC DPTR	Increment data pointer	1	2	A3
MUL AB	Multiply A by B	1	4	A4
DIV AB	Divide A by B	1	4	84
DA A	Decimal Adjust A	1	1	D4

LOGICAL				
Mnemonic	Description	Bytes	Cycles	Hex code
ANL A,Rn	AND register to A	1	1	58-5F
ANL A,dir	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect memory to A	1	1	56-57
ANL A,#data	AND immediate to A	2	1	54
ANL dir,A	AND A to direct byte	2	1	52
ANL dir,#data	AND immediate to direct byte	3	2	53
ORL A,Rn	OR register to A	1	1	48-4F
ORL A,dir	OR direct byte to A	2	1	45
ORL A,@Ri	OR indirect memory to A	1	1	46-47
ORL A,#data	OR immediate to A	2	1	44
ORL dir,A	OR A to direct byte	2	1	42
ORL dir,#data	OR immediate to direct byte	3	2	43
XRL A,Rn	Exclusive-OR register to A	1	1	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	1	65

XRL A, @Ri	Exclusive-OR indirect memory to A	1	1	66-67
XRL A,#data	Exclusive-OR immediate to A	2	1	64
XRL dir,A	Exclusive-OR A to direct byte	2	1	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap Nibbles of A	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RR A	Rotate A right	1	1	03
RRC A	Rotate A right through carry	1	1	13

DATA TRANSFER				
Mnemonic	Description	Bytes	Cycles	Hex code
MOV A,Rn	Move register to A	1	1	E8-EF
MOV A,dir	Move direct byte to A	2	1	E5
MOV A,@Ri	Move indirect memory to A	1	1	E6-E7
MOV A,#data	Move immediate to A	2	1	74
MOV Rn,A	Move A to register	1	1	F8-FF
MOV Rn,dir	Move direct byte to register	2	2	A8-AF
MOV Rn,#data	Move immediate to register	2	1	78-7F
MOV dir,A	Move A to direct byte	2	1	F5
MOV dir,Rn	Move register to direct byte	2	2	88-8F
MOV dir,dir	Move direct byte to direct byte	3	2	85
MOV dir,@Ri	Move indirect memory to direct byte	2	2	86-87
MOV dir,#data	Move immediate to direct byte	3	2	75
MOV @Ri,A	Move A to indirect memory	1	1	F6-F7
MOV @Ri,dir	Move direct byte to indirect memory	2	2	A6-A7
MOV @Ri,#data	Move immediate to indirect memory	2	1	76-77
MOV DPTR,#data	Move immediate to data pointer	3	2	90
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	2	93
MOVC A,@A+PC	Move code byte relative PC to A	1	2	83
MOVX A,@Ri	Move external data(A8) to A	1	2	E2-E3
MOVX A,@DPTR	Move external data(A16) to A	1	2	E0
MOVX @Ri,A	Move A to external data(A8)	1	2	F2-F3
MOVX @DPTR,A	Move A to external data(A16)	1	2	F0
PUSH dir	Push direct byte onto stack	2	2	C0
POP dir	Pop direct byte from stack	2	2	D0
XCH A,Rn	Exchange A and register	1	1	C8-CF
XCH A,dir	Exchange A and direct byte	2	1	C5
XCH A,@Ri	Exchange A and indirect memory	1	1	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	1	D6-D7

BOOLEAN				
Mnemonic	Description	Bytes	Cycles	Hex code
CLR C	Clear carry	1	1	C3
CLR bit	Clear direct bit	2	1	C2
SETB C	Set carry	1	1	D3
SETB bit	Set direct bit	2	1	D2
CPL C	Complement carry	1	1	B3
CPL bit	Complement direct bit	2	1	B2
ANL C,bit	AND direct bit to carry	2	2	82
ANL C,/bit	AND direct bit inverse to carry	2	2	B0
ORL C,bit	OR direct bit to carry	2	2	72
ORL C,/bit	OR direct bit inverse to carry	2	2	A0
MOV C,bit	Move direct bit to carry	2	1	A2
MOV bit,C	Move carry to direct bit	2	2	92
BRANCHING				
Mnemonic	Description	Bytes	Cycles	Hex code
ACALL addr 11	Absolute jump to subroutine	2	2	11→F1
LCALL addr 16	Long jump to subroutine	3	2	12
RET	Return from subroutine	1	2	22
RETI	Return from interrupt	1	2	32
AJMP addr 11	Absolute jump unconditional	2	2	01→E1
LJMP addr 16	Long jump unconditional	3	2	02
SJMP rel	Short jump (relative address)	2	2	80
JC rel	Jump on carry = 1	2	2	40
JNC rel	Jump on carry = 0	2	2	50
JB bit,rel	Jump on direct bit = 1	3	2	20
JNB bit,rel	Jump on direct bit = 0	3	2	30
JBC bit,rel	Jump on direct bit = 1 and clear	3	2	10
JMP @A+DPTR	Jump indirect relative DPTR	1	2	73
JZ rel	Jump on accumulator = 0	2	2	60
JNZ rel	Jump on accumulator ≠ 0	2	2	70
CJNE A,dir,rel	Compare A,direct jne relative	3	2	B5
CJNE A,#d,rel	Compare A,immediate jne relative	3	2	B4
CJNE Rn,#d,rel	Compare register, immediate jne relative	3	2	B8-BF
CJNE @Ri,#d,rel	Compare indirect, immediate jne relative	3	2	B6-B7
DJNZ Rn,rel	Decrement register, jnz relative	2	2	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	2	D5
MISCELLANEOUS				
Mnemonic	Description	Bytes	Cycles	Hex code
NOP	No operation	1	1	00

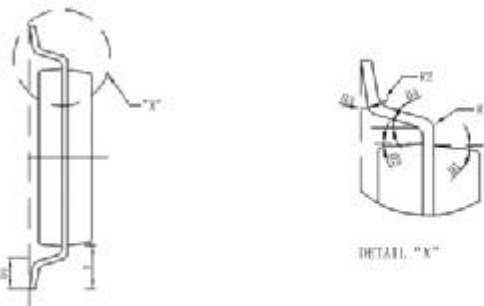
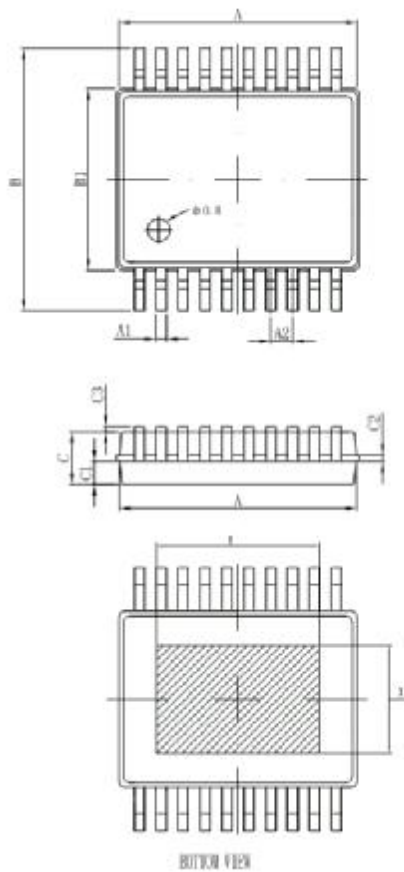
In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Noncontinuous blocks of codes, shown as 11→F1 (for example), are used for absolute jumps and calls, with the top 3 bits of the code being used to store the top three bits of the destination address.

The CJNE instructions use the abbreviation #d for immediate data.

## 14、封装尺寸与外形图



14. 1、TSSOP20 外形图与封装尺寸



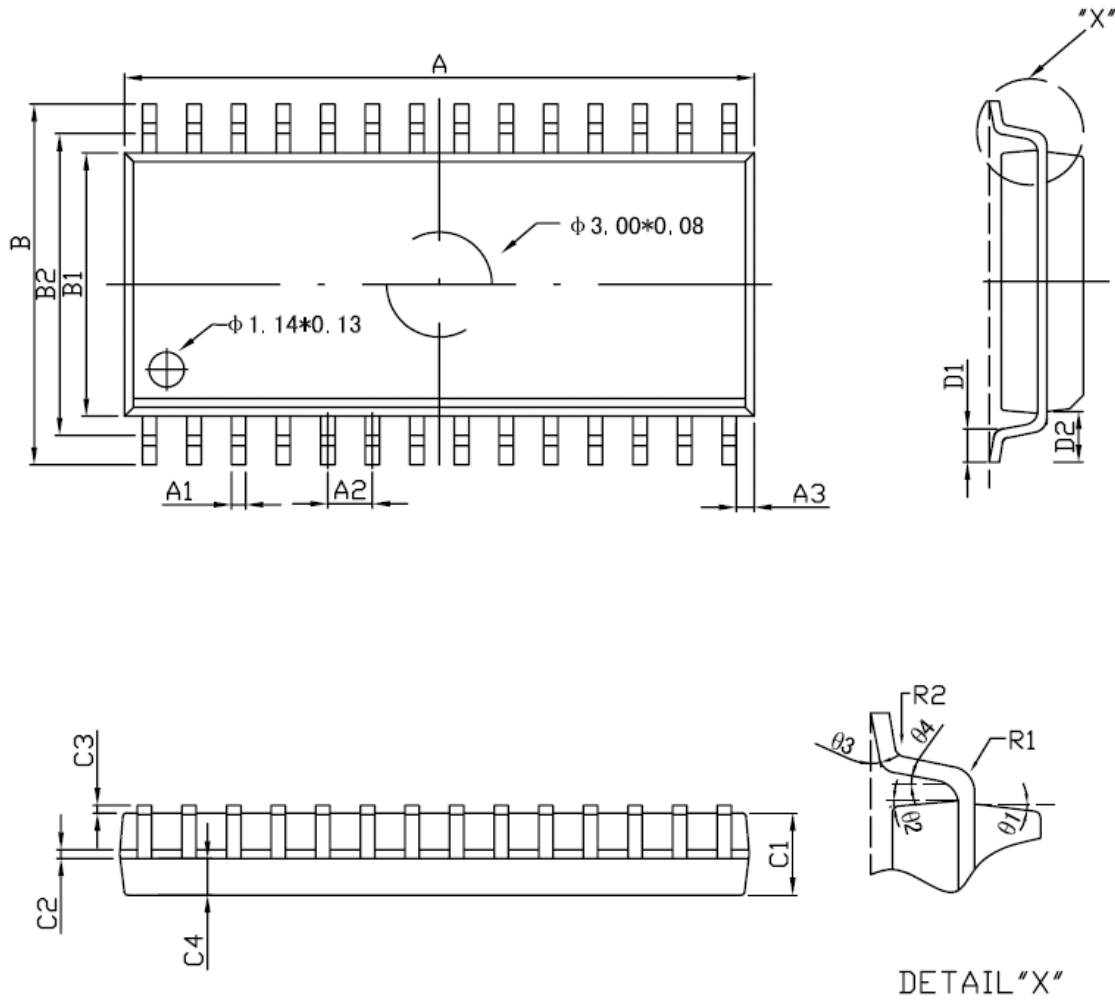
Note:

1. Formed lead shall be planar with respect to one another within 0.004 inches.
2. Both package length and width do not include mold flash and burr.

尺寸	最小 (mm)	最大 (mm)	尺寸	最小 (mm)	最大 (mm)
A	6.40	6.50	C3	0.225	0.102
A1	0.20	0.30	D	1.0 TYP	
A2	0.60 TYP		D1	0.50	0.75
B	6.35	6.50	R1	0.15 TYP	
B1	4.30	4.50	R2	0.15 TYP	
C	0.50	1.05	Ø1	12° TYP	
C1	0.4305 TYP		Ø2	12° TYP	
C2	0.09	0.2	Ø3	8° TYP	8° TYP
			Ø4	19° TYP	

OPTION	PAD SIZE	SYMBOL	DIMENSION	MARK
1	3(118)	X	MIN.2.00 MAX.3.19	NORMAL
	4(2165)	Y	MIN.3.50 MAX.4.39	
2	3-1H(83)	X	MIN.1.71 MAX.2.21	SPECIAL CUSTOMER
	3-1S(124)	Y	MIN.2.75 MAX.3.25	

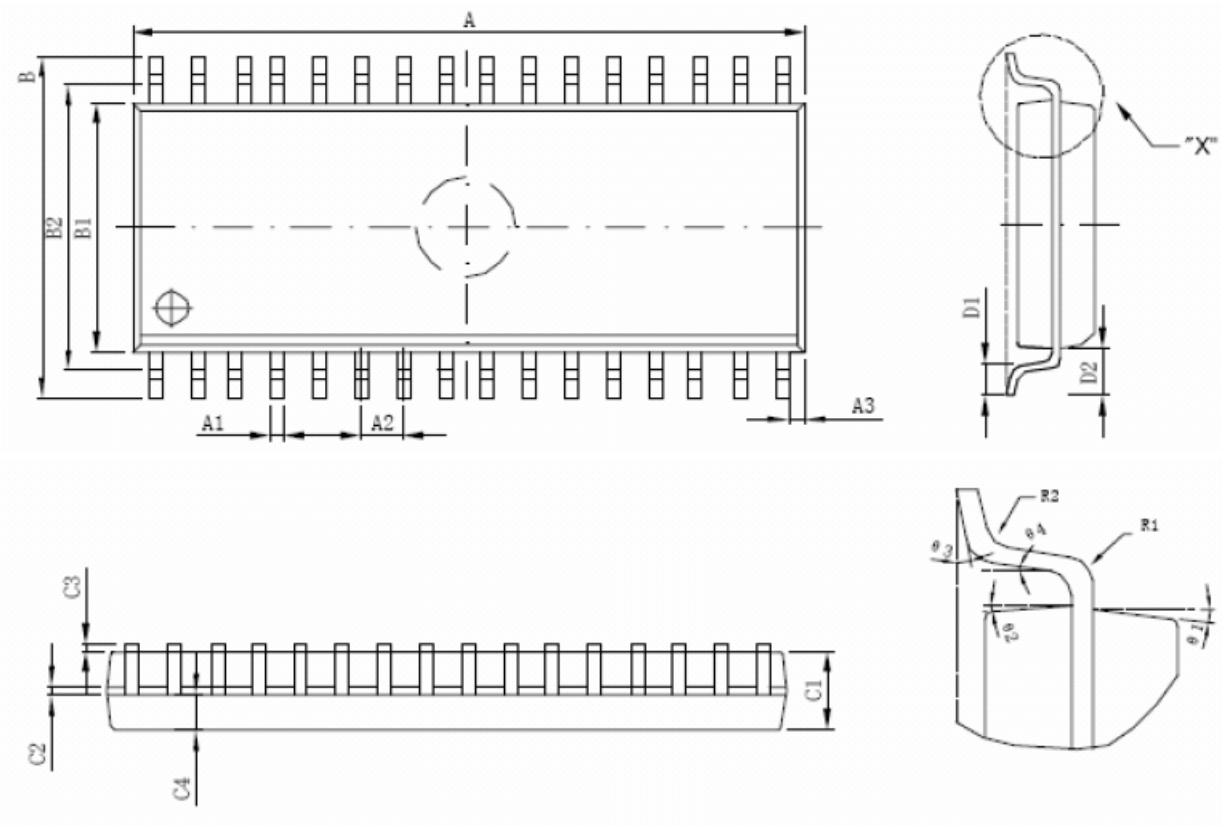
14. 2、SOP28 外形图与封装尺寸



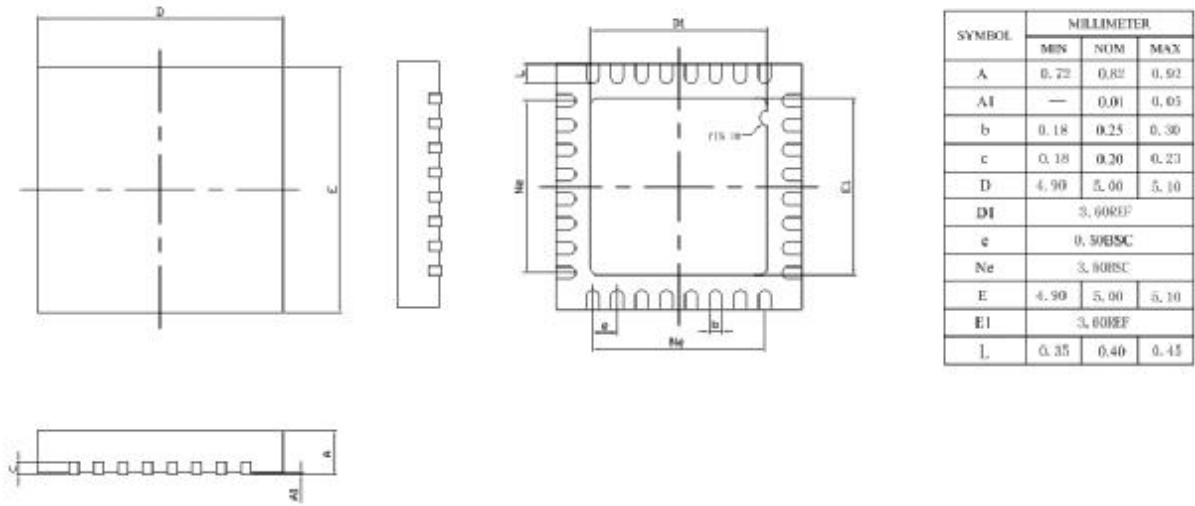
标注	尺寸	最小 (mm)	最大 (mm)	标注	尺寸	最小 (mm)	最大 (mm)
A		17.83	18.03	C4		1.043TYP	
A1		0.4064TYP		D1		0.70	0.90
A2		1.27TYP		D2		1.395TYP	
A3		0.51TYP		R1		0.508TYP	
B		9.90	10.50	R2		0.508TYP	
B1		7.42	7.62	$\theta 1$		7° TYP	
B2		8.9TYP		$\theta 2$		5° TYP	
C1		2.24	2.44	$\theta 3$		4° TYP	
C2		0.204	0.33	$\theta 4$		10° TYP	
C3		0.10	0.25				

### 14.3、SOP32 外形图与封装尺寸

标注	尺寸	最小 (mm)	最大 (mm)	标注	尺寸	最小 (mm)	最大 (mm)
A		20.88	21.08	C4		0.99TYP	
A1		0.3	0.5	D1		0.55	0.95
A2		1.27TYP		D2		1.45	
A3		0.77TYP		R1			
B		10.2	10.6	R2			
B1		7.42	7.62	$\theta 1$		8°TYP	
B2		8.9TYP		$\theta 2$		15°TYP	
C1		2.14	2.34	$\theta 3$		4°TYP	
C2		0.2	0.32	$\theta 4$		14°TYP	
C3		0.10	0.25				



14. 4、QFN32 外形图与封装尺寸



15、声明及注意事项:

15.1、产品中有毒有害物质或元素的名称及含量

部件名称	有毒有害物质或元素
------	-----------

	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr(VI))	多溴联苯 (PBBs)	多溴联苯 醚(PBDEs)
引线框	○	○	○	○	○	○
塑封树脂	○	○	○	○	○	○
芯片	○	○	○	○	○	○
内引线	○	○	○	○	○	○
装片胶	○	○	○	○	○	○
说明	○: 表示该有毒有害物质或元素的含量在 SJ/T11363-2006 标准的检出限以下。 ×: 表示该有毒有害物质或元素的含量超出 SJ/T11363-2006 标准的限量要求。					

## 15.2 注意

在使用本产品之前建议仔细阅读本资料；

本资料中的信息如有变化，恕不另行通知；

本资料仅供参考，本公司不承担任何由此而引起的任何损失；

本公司也不承担任何在使用过程中引起的侵犯第三方专利或其它权利的责任。