

具有12-bit A/D 转换功能的 8位CMOS 单片机



主要特性

- 8-bit 高速 8051 CPU
- 基本 MCU 功能
 - 8K字节 Flash 程序存储器
 - 512字节 SRAM
- 内置模拟功能
 - 上电复位和低电压检测复位
 - 内部 16MHz RC 振荡器 ($\pm 1.5\%$, $T_A = 0 \sim +50^\circ\text{C}$)
 - Watchdog Timer RC 振荡器 (5kHz)
- 外围特性
 - 12-bit AD 转换器(10 inputs)
 - UART 8-bit x 1-ch
 - SPI 8-bit x 1-ch
 - I2C 8-bit x 1-ch
- I/O 和封装
 - 高达 18 可编程 I/O (20 SOP)
 - 20 SOP, 20 PDIP, 20QFN
 - Pb-free package
- 工作条件
 - 1.8V to 5.5V宽电压范围
 - -40°C to 85°C 温度范围
- 应用
 - 小家电

MC96F8208S 用户手册

V 1.9

修订历史

Version	Date	Revision list
0.0	2013.01.02	出版.
0.1	2013.01.24	修订. 移除“MC96F8208SV (16-DIP)”封装
0.2	2013.03.26	修订. 增加 “MC96F8208SU (20-QFN)”封装
1.0	2013.06.26	修订. 增加与 “MC96F8208 EVA(MC96F8208OCD/OB/OU/OM)”相关的内容 增加烧录器 “PGMplusLC Writer”
1.1	2013.07.15	增加 IOH/IOL表. 移除 “MC96F8208SM (16-SOP)”封装 移除与 “MC96F8208OM(EVA)”相关的内容
1.2	2014.02.27	修改指令“rel” 添加直流特性1,IDD2,IDD3,IDD4,IDD5) 添加封装6F8208SMN (16-SOP)”
1.3	2014.03.24	AVREF 修改 1.8V~VDD to 2.7V~VDD.
1.4	2014.04.14	AVREF 修改 2.7V~VDD to 1.8V~VDD.
1.5	2014.07.09	删除 “PGMplusLC 烧录” Change ‘Read Protection’ to ‘Code Read Protection’ in Configure Option. Change ‘Hard-Lock’ to ‘Code Write Protection’in Configure Option. Change ‘RESETB port with a pull-up resistor’ to ‘Enable RESETB pin’ in Configure Option. Change ‘Protection Area Enable/Disable’ to ‘Specific Area Write Protection’ in Configure Option. Change ‘Protection Area Size Select’ to ‘Select Specific Area for Write Protection’ in Configure Option.
1.6	2014.11.05	Revised this book. Add contents of Flash, “Protection for Invalid Erase/Write”.
1.7	2015.02.03	Add a Package Type “MC96F8208SR (20-TSSOP)” Modify contents of Interrupt sequence flow
1.8	2015.03.10	Modify the Figure 4.1 20-Pin SOP Package Modify the Figure 11.32 A/D Analog Input Pin with Capacitor
1.9	2015.04.09	Revised this book Add a note at P0, P1, P2, P3, EIFLAG0, EIFLAG1 register description and SFR map.

版本 1.9

由 FAE 团队发行

版权所有 2015 ABOV Semiconductor Co. Ltd..

本手册的附加信息可能会提供给 ABOV 办事处及分销商.

ABOV半导体公司有权修改任何信息在任何时间, 恕不另行通知.

本手册中的信息, 图表和其他数据是正确而且可靠的;

但是, 在使用本手册过程中任何违反专利或第三方权利的纠纷, ABOV 公司概不负责.

1 概述

1.1. 说明

MC96F8208S 是一款有8K字节FLASH的 8位 CMOS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. 该芯片有以下特点: 8k 字节 FLASH, 256 字节 IRAM, 256 字节 XRAM , 通用 I/O, 基本间隔定时器, 8/16位定时器/计数器, 16位 PPG 输出, 8位 PWM 输出, watch timer, 蜂鸣器驱动口, SPI, UART, I2C, 12-bit A/D 转换器, 片内 POR, LVR, LVI, 片内振荡和时钟电路. MC96F8208S 也支持节电模式以减少电量消耗.

Device Name	EVA	FLASH	XRAM	IRAM	Package
MC96F8208SD	MC96F8208OCD	8 Kbytes FLASH	256 bytes	256 bytes	20 SOP
MC96F8208SB	MC96F8208OB				20 PDIP
MC96F8208SU	MC96F8208OU				20 QFN

Table 1.1 MC96F8208S选型表

1.2 特性

- CPU
 - 8-bit CISC 内核 (M8051, 2时钟/周期)
- ROM (FLASH) 容量
 - 8 K字节自读写 Flash
 - In-System Programming(ISP)
 - 擦写次数 : 10,000 times(Sector 0~247)
100,000 times(Sector 248~255)
- 256bytes IRAM
- 256bytes XRAM
- 通用 I/O (GPIO)
 - 普通 I/O : 18 Ports
(P0[3:0], P1[6:1], P2[5:4], P3[7:5/2:0])
- 定时器/计数器
 - Basic Interval Timer (BIT) 8-bit× 1-ch
 - Watch Dog Timer (WDT) 8-bit× 1-ch
 - 5kHz 内部 RC 振荡器
 - 8-bit× 1-ch(T0), 16-bit× 2-ch (T1/T2)
- 可编程脉冲发生器
 - 脉冲发生器 (by T1/T2)
 - 8-Bit PWM (by T0)
- Watch Timer (WT)
 - 3.91ms/0.25s/0.5s/1s/1min 间隔
(4.194304MHz)
- Buzzer
 - 8-bit× 1-ch
- SPI 2
 - 8-bit× 1-ch
- UART
 - 8-bit× 1-ch
- I2C
 - 8-bit× 1-ch
- 12-bit A/D 转换器
 - 10 个输入通道
 - 睡眠唤醒功能
- 上电复位
 - 复位电平 (1.4V)
- 低电压复位
 - 14 个检测电平 (1.60/ 2.00/ 2.10/ 2.20/ 2.32/
2.44/ 2.59/ 2.75/ 2.93/ 3.14/ 3.38/ 3.67/ 4.00/ 4.40V)
- 低电压检测
 - 13 个检测电平 (2.00 / 2.10/ 2.20/ 2.32/ 2.44/
2.59/ 2.75/ 2.93/ 3.14/ 3.38/ 3.67/ 4.00/ 4.40V)
- 中断源
 - 外部中断
(EINT0~2, EINT5, EINT6, EINT10, EINT11, EINT12)
(6)
 - Timer(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 ±1.5% (TA= 0 ~ +50°C, VDD=2.0V ~ 5.5V)
- 省电模式
 - STOP, IDLE 模式
- 工作电压和频率
 - 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)
- 工作温度
 - -40 ~ +85 °C
- 振荡器类型
 - 0.4-12晶体或陶瓷振荡器做为主时钟
- 封装类型
 - 20 SOP
 - 20 PDIP
 - 20 QFN
 - Pb-free package

1.3 开发工具

1.3.1 编译器

ABOV 公司不提供编译工具。 建议咨询编译器提供商。
MC96F8208S 是曼托 8051 内核,并且 ROM 容量小于 64k 。开发人员可以使用所用8051 标准编译器。

1.3.2 OCD(On-chip debugger) 仿真器和调试器

OCD (片内调试)仿真器支持 ABOV 半导体8051系列MCU的仿真. OCD使用两根线连接PC和依附在用户系统上的MC U . OCD 能够读到或者更改MCU内部存储器和 I/O 外围设备的值.同时可以控制MCU内部调试逻辑 , 就是可以控制仿真, 单步运行, 监测等.
OCD 调试程序安装在Microsoft-Windows NT, 2000, XP, Vista (32bit) 操作系统下运行.I如果想要了解更详细的信息, 请参考OCD调试手册. 也可以在我们的网站下载调试软件和手册.

- 连接:
- DSCL (MC96F8208 EVA P01 port)
 - DSDA (MC96F8208 EVA P00 port)

- NOTE)
1. MC96F8208S 不支持 OCD 功能. MC96F8208 EVA 用于仿真和调试.
 2. MC96F8208 EVA 有 MC96F8208OCD, MC96F8208OB 和 MC96F8208OU.

OCD 连接图: 连接 OCD 和用户系统

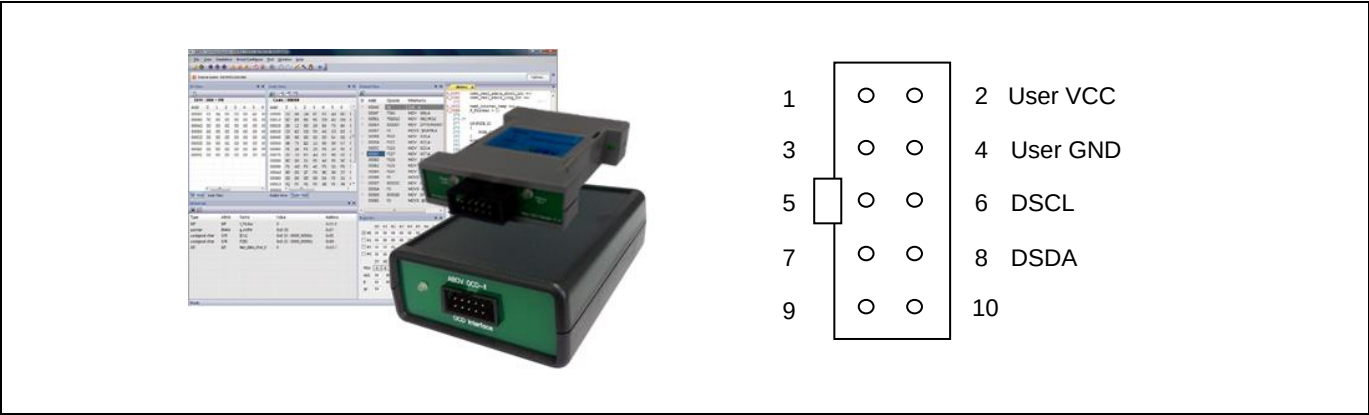


Figure 1.1 调试工具和引脚说明

Subject	MC96F8208S	MC96F8208 EVA(Evaluation chip)
Program/Data Memory	1. FLASH : 8k bytes 2. XRAM : 256 bytes	1. FLASH : 16k bytes 2. XRAM : 512 bytes
IOH/IOL of ports (Chapter 7.8 – DC Characteristics)	1. All Output Port : IOH = -2mA @VDD=4.5V 2. All Output Port : IOL = 15mA @VDD=4.5V	1. P1 : IOH = -20mA @VDD = 4.5V / P0, P26, P3 : IOH = -10mA @VDD = 4.5V / P20-25 : IOH = -10mA @VDD=4.5 2. All Output Port except P20-P25 : IOL = 15mA @VDD=4.5/ P20-P25 : IOL = 160mA @VDD4.5V
P04 – P06 Related Registers (Chapter 9.3 – P0 Port)	1. Bits 4-6 of P0/P0IO/P0PU/P0OD Registers : “Not used” 2. Bits 2-4 of P03DB Register : “Not used” 3. Bits 5-7 of P0FSR Register : “Not used”	1. Bits 4-6 of P0/P0IO/P0PU/P0OD Registers : “Used (000b)” 2. Bits 2-4 of P03DB Register : “Used (000b)” 3. Bits 5-7 of P0FSR Register : “Used (000b)”
P10 and P17 Related Registers (Chapter 9.4 – P1 Port)	1. Bits 0 and 7 of P1/P1IO/P1PU/P1OD Registers : “Not used” 2. Bit 0 of P12DB Register : “Not used” 3. Bit 6 of P1FSRH Register : “Not used” 4. Bit 0 of P1FSRL Register : “Not used”	1. Bits 0 and 7 of P1/P1IO/P1PU/P1OD Registers : “Used (0b and 0b)” 2. Bit 0 of P12DB Register : “Used (0b)” 3. Bit 6 of P1FSRH Register : “Used (0b)” 4. Bit 0 of P1FSRL Register : “Used (0b)”
P20 – P23 and P26 Related Registers (Chapter 9.5 – P2 Port)	1. Bits 0-3 and 6 of P2/P2IO/P2PU/P2OD Registers : “Not used” 2. Bits 4-7 of P12DB Register : “Not used”	1. Bits 0-3 and 6 of P2/P2IO/P2PU/P2OD Registers : “Used (0000b and 0b)” 2. Bits 4-7 of P12DB Register : “Used (0000b)”
P33 – P34 Related Registers (Chapter 9.6 – P3 Port)	1. Bits 3-4 of P3/P3IO/P3PU/P3OD/P3FSR Registers: “Not used”	1. Bits 3-4 of P3/P3IO/P3PU/P3OD/P3FSR Registers: “Used (00b)”
EINT2 – 4 Related Registers (Chapter 10.2 – Ext. Interrupt)	1. INT5E of IE controls only EINT0/1. 2. Bits 2-4 of EIFLAG0 Register : “Not used” 3. Bits 4-7 of EIPOL0L Register : “Not used” 4. Bits 0-1 of EIPOL0H Register : “Not used”	1. INT5E of IE controls EINT0-4. 2. Bits 2-4 of EIFLAG0 Register : “Used (000b)” 3. Bits 4-7 of EIPOL0L Register : “Used (0000b)” 4. Bits 0-1 of EIPOL0H Register : “Used (00b)”
EINT5 Related Registers (Chapter 10.2 – Ext. Interrupt)	1. Bit 0 of IE1 Register : “Not used” 2. Bit 5 of EIFLAG0 Register : “Not used” 3. Bits 2-3 of EIPOL0H Register : “Not used”	1. Bit 0 of IE1 Register : “Used (0b)” 2. Bit 5 of EIFLAG0 Register : “Used (0b)” 3. Bits 2-3 of EIPOL0H Register : “Used (00b)”
EINT7 – A Related Registers (Chapter 10.2 – Ext. Interrupt)	1. Bit 5 of IE2 Register : “Not used” 2. Bits 0-3 of EIFLAG1 Register : “Not used” 3. EIPOL2 is not exist.	1. Bit 5 of IE2 Register : “Used (0b)” 2. Bits 0-3 of EIFLAG1 Register : “Used (0000b)” 3. Bits 0-7 of EIPOL2 Register : “Used (0000 0000b)”

Table 1.2 MC96F8208S 和仿真芯片 (MC96F8208 EVA)的不同

Subject	MC96F8208S	MC96F8208 EVA(Evaluation chip)
Clock Generator Related Registers (Chapter 11.1)	1. Bits 0-1 of SCCR Register "11b" : "Not available" 2. Bit 0 of OSCCR Register : "Not used"	1. Bit0-1 of SCCR Register "11b" : "Available" 2. Bit 0 of OSCCR Register : "Used (0b)"
Watch Timer Related Register (Chapter 11.4)	1. Bits 0-1 of WTCR Register "00b" : "Not available"	1. Bits 0-1 of WTCR Register "00b" : "Available"
Timer 0 Related Register (Chapter 11.5)	1. Bits 1-3 of T0CR Register "111b" : "Not available"	1. Bits 1-3 of T0CR Register "111b" : "Available"
Timer 2 Related Register (Chapter 11.7)	1. Bit 3 of T2CRL Register : "Always 1b"	1. Bit 3 of T2CRL Register : "Used (0b)"
Buzzer Related Register (Chapter 11.7)	1. Bits 1-3 of BUZCR Register "1xxb" : "Not available"	1. Bits 1-3 of BUZCR Register "1xxb" : "Available"
AN4 – AN7,AN14 Related Registers (Chapter 11.9)	1. Bits 0-3 of ADCCRL Register "0100b"/ "0101"/ "0110b"/ "0111"/ "1110b": "Not available" 2. ADWRCR2 is not exist. 3. Bits 4-5 of ADWRCR3 Register : "Not used" 4. Bit 6 of ADWCRH Register : "Not used" 5. Bits 4-7 of ADWCRL Register : "Not used" 6. Bit 6 of ADWIFRH Register : "Not used" 7. Bits 4-7 of ADWIFRL Register : "Not used"	1. Bits 0-3 of ADCCRL Register : "0100b"/ "0101"/ "0110b"/ "0111"/ "1110b": "Available" 2. Bits 0-7 ADWRCR2 Register : "Used (0000 0000b)" 3. Bits 4-5 of ADWRCR3 Register : "Used (00b)" 4. Bit 6 of ADWCRH Register : "Used (0b)" 5. Bits 4-7 of ADWCRL Register : "Used (0000b)" 6. Bit 6 of ADWIFRH Register : "Used (0b)" 7. Bits 4-7 of ADWIFRL Register : "Used (0000b)"
SPI Related Register (Chapter 11.10)	1. Bit 2/5 of SPISR Register : "Not used"	1. Bit 2/5 of SPISR Register : "Used (00b)"
I2C Related Register (Chapter 11.12)	1. If P31 and P30 are used as SCL/SDA for I2C, P3OD[1] and P3OD[0] should be set "1". 2. If P25 and P24 are used as SCL/SDA for I2C, P2OD[5] and P2OD[4] should be set "1".	1. Don't care
Full-flash Erase Mode Method (Chapter 15 – Flash Memory)	1.Sector Erase Mode	1.Sector and Byte Erase Mode

Table 1.2 MC96F8208S 和仿真芯片 (MC96F8208 EVA)的不同

1.3.3 烧录器

单一烧录器:

PGMplus USB: 可直接烧写 MCU.

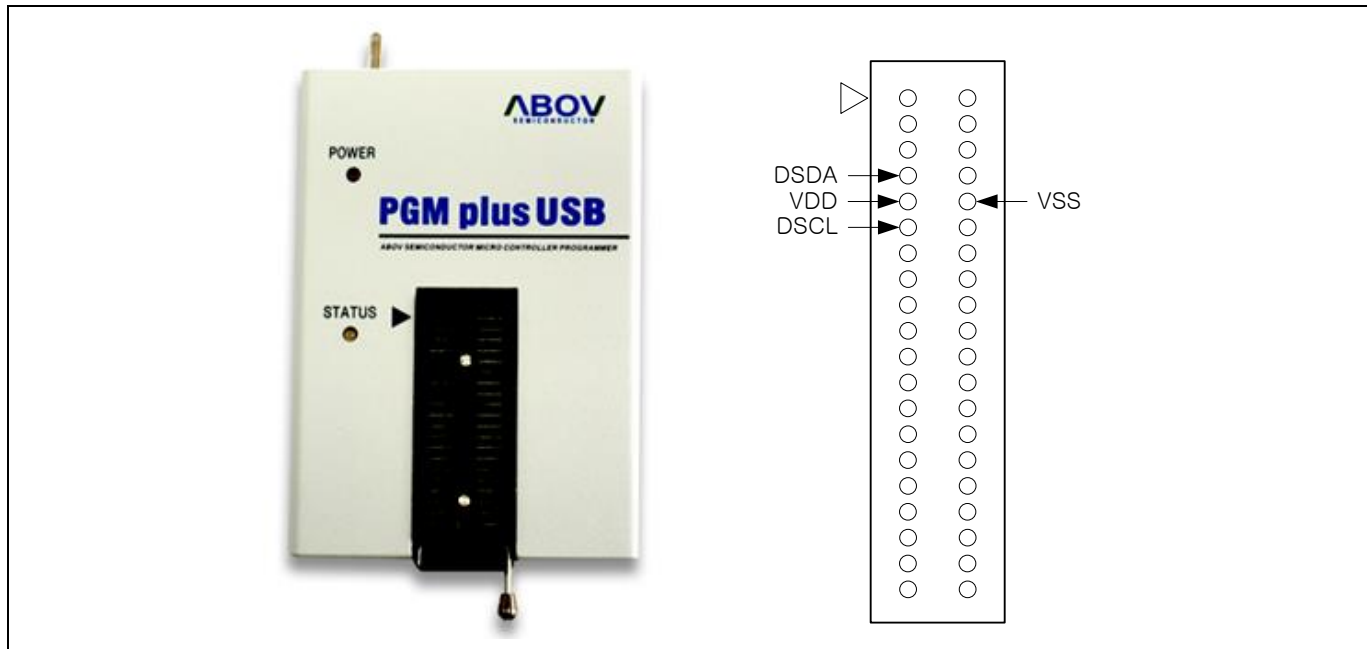


Figure 1.2 PGMplusUSB(Single writer)

Standalone PGMplus:

可直接烧写 MCU.

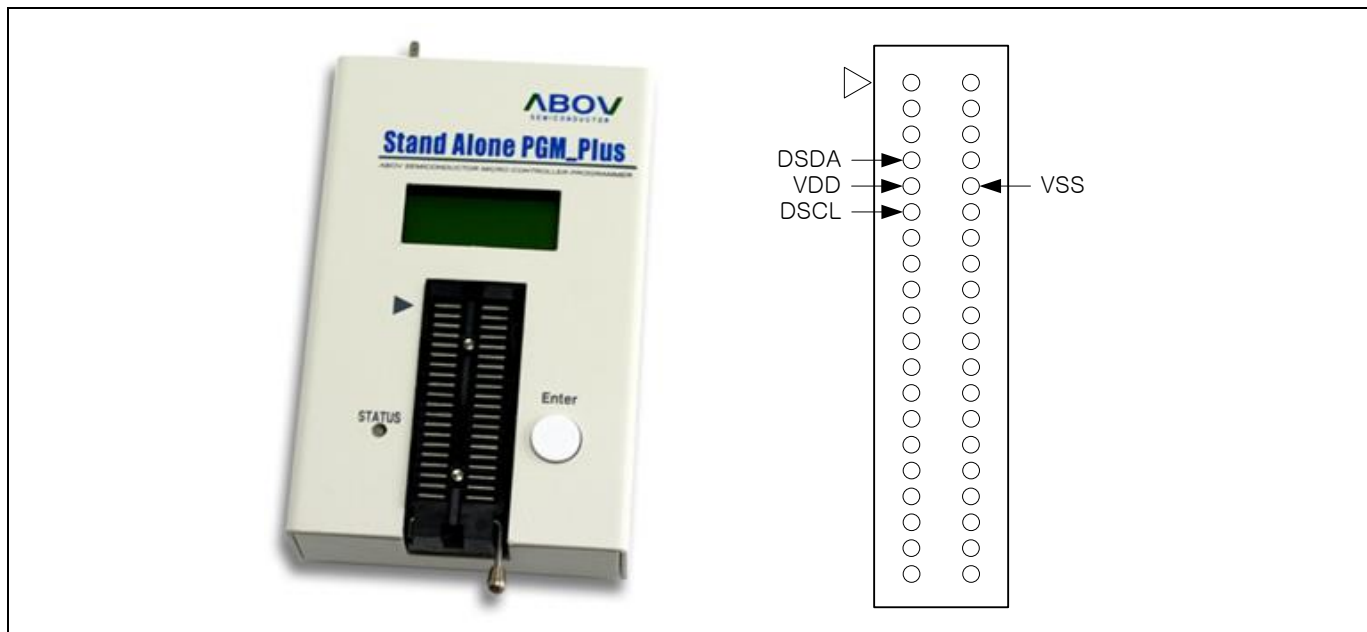


Figure 1.3 Standalone PGMplus(单一烧写器)

OCD 仿真器:

也可以用于 MCU 程序烧写, 因为 OCD 支持 ISP (In System Programming). 除了开发目标板之外不需要额外的 H/W,.

Gang 烧写器:

可一次性烧写 8 个MCU..

因此，它主要在大规模的生产中使用，在代码加载到烧写器之后无需和电脑连接.



Figure 1.4 StandAlone Gang8 (用于量产)

1.4 MTP 烧录

1.4.1 概述

MC96F8208S程序存储器是 MTP 类型. flash 以串行数据格式存储. 有四根连接线(DSCL, DSDA, VDD, VSS) 用于烧写/读取 flash.

Pin name	Main chip pin name	During programming	
		I/O	Description
DSCL	P01	I	串行时钟.仅输入状态.
DSDA	P00	I/O	串行数据引脚. 读取时为输出.烧录时为输入. 可设置为输入/上拉输出
VDD, VSS	VDD, VSS	-	电源 .

Table 1.3 用于烧写/读取Flash 时端口说明

1.4.2 在线烧录

MC96F8208S只需要四根串行协议信号线（包括VDD和VSS）来烧写 FLASH. 所以如果PCB板设计时已经设计好烧写信号线是可以在线烧录的.

1.4.3 电路设计指导

在 FLASH 烧写过程中, 烧录工具需要4根信号线 DSCL, DSDA, VDD, 和 VSS. 在设计PCB 电路时, 你需要考虑这些信号线在线烧录时的使用.

请务必注意设计这些信号线相关的电路, 因为DSCL 和 DSDA 的时序对烧录非常重要.

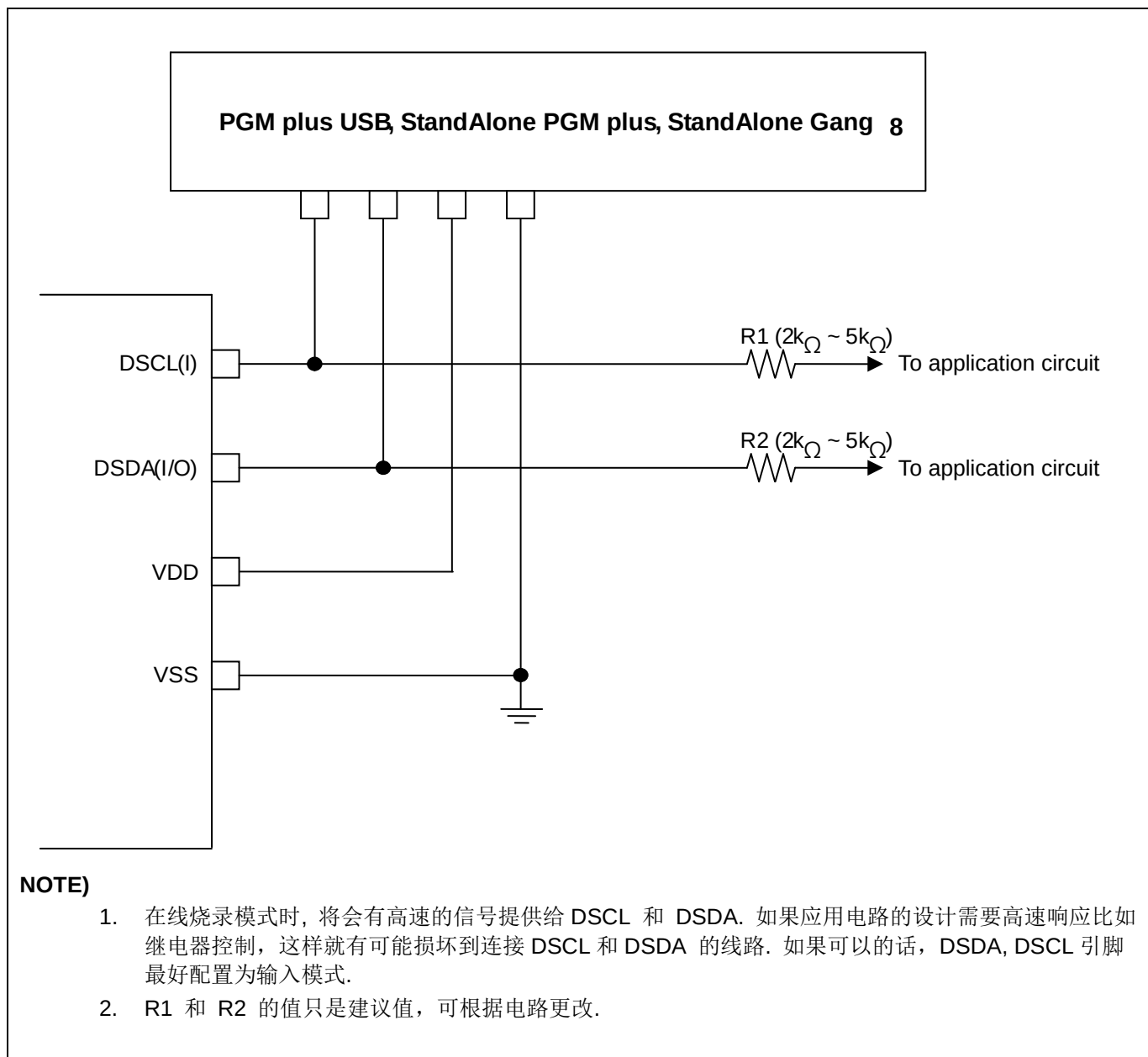


Figure 1.5 在线烧录 PCB 设计指导

2 方框图

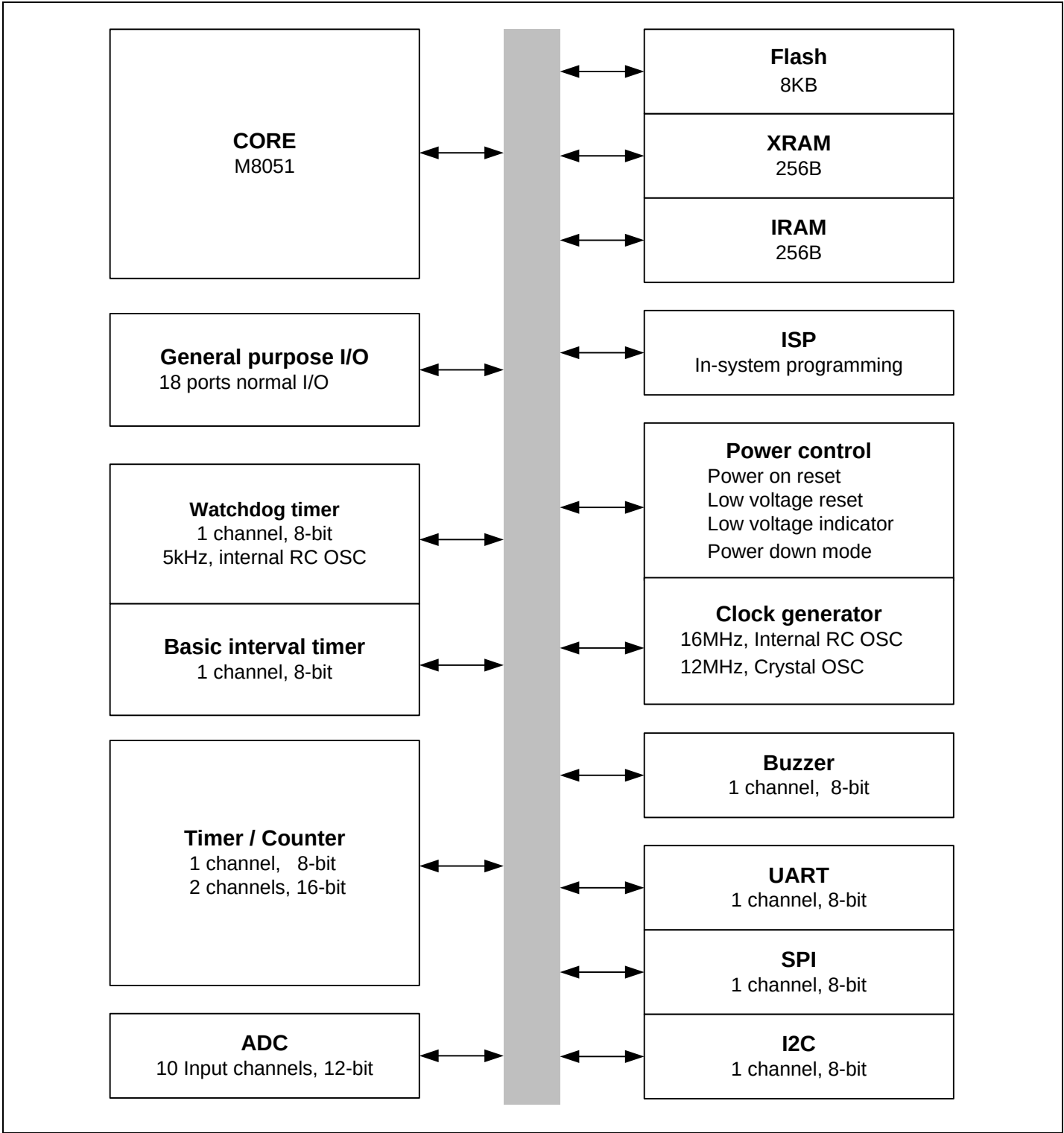


Figure 2.1 MC96F8208S方框图

3 引脚分配

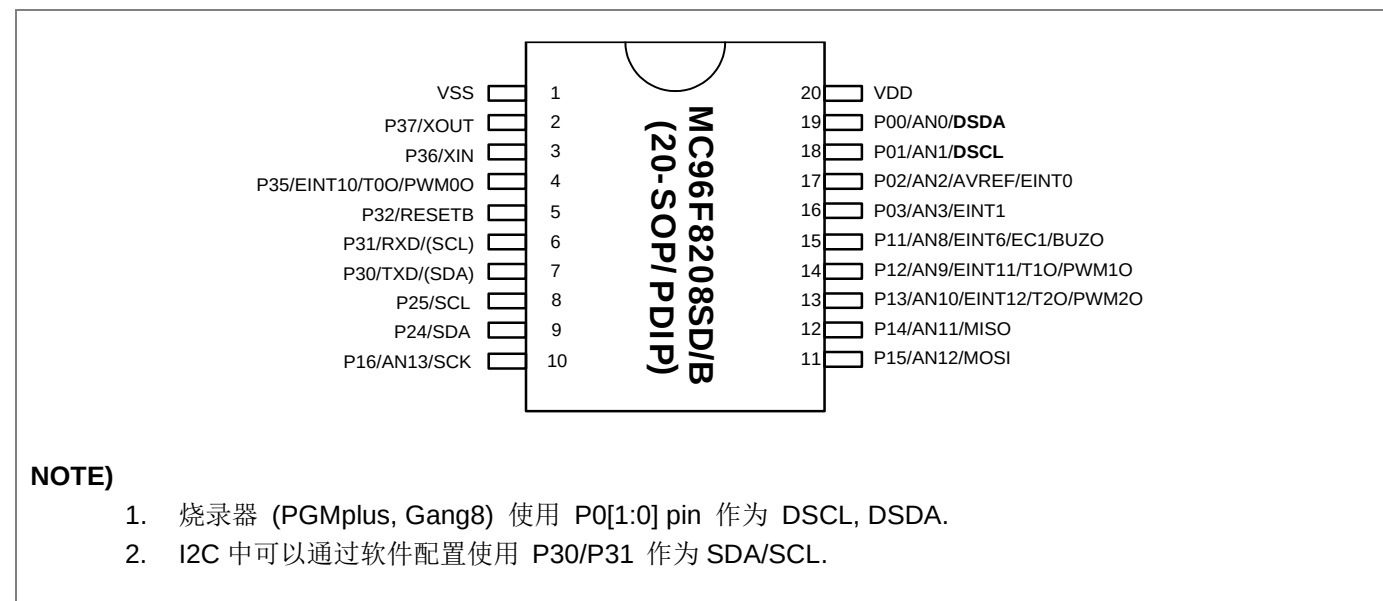


Figure 3.1 MC96F8208SD/B 20SOP/PDIP 引脚分配

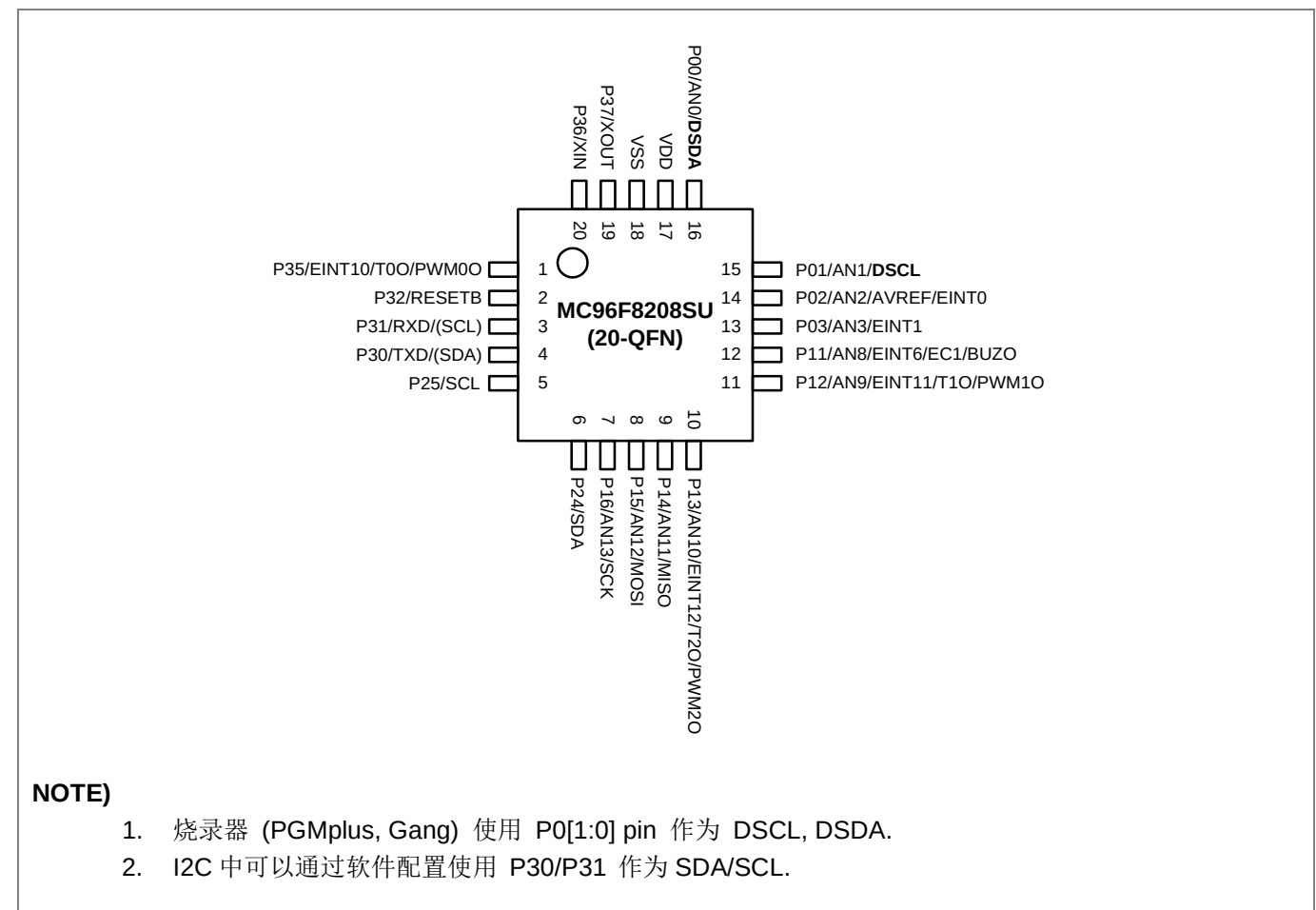
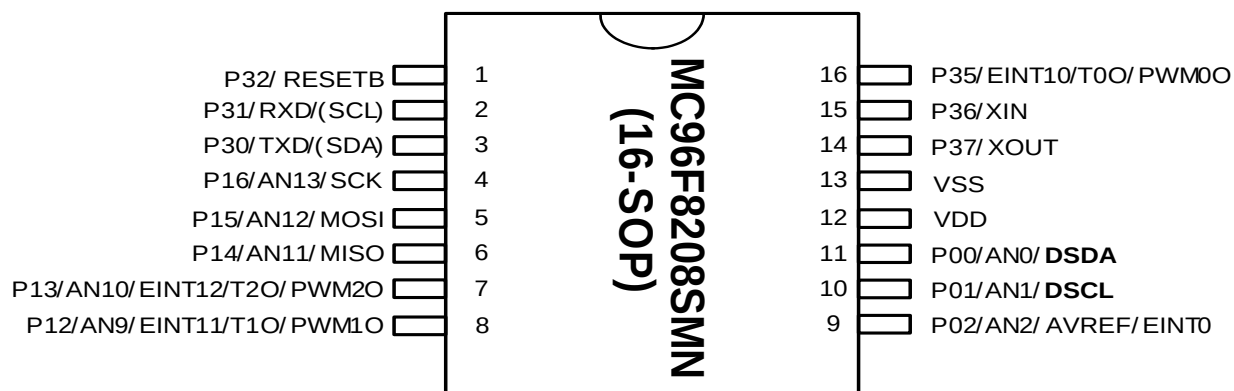


Figure 3.2 MC96F8208SU 20QFN 引脚分配



NOTE)

3. The programmer (PGMplus, Gang8) uses P0[1:0] pin as DSCL, DSDA.
4. The P03, P11, P24 and P25 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 16-pin package is used.
5. The SDA/SCL lines for I2C can be configured on the P30/P31 pins by software control.

Figure 3.3 MC96F8208SMN16SOP PinAssignment

4 封装图

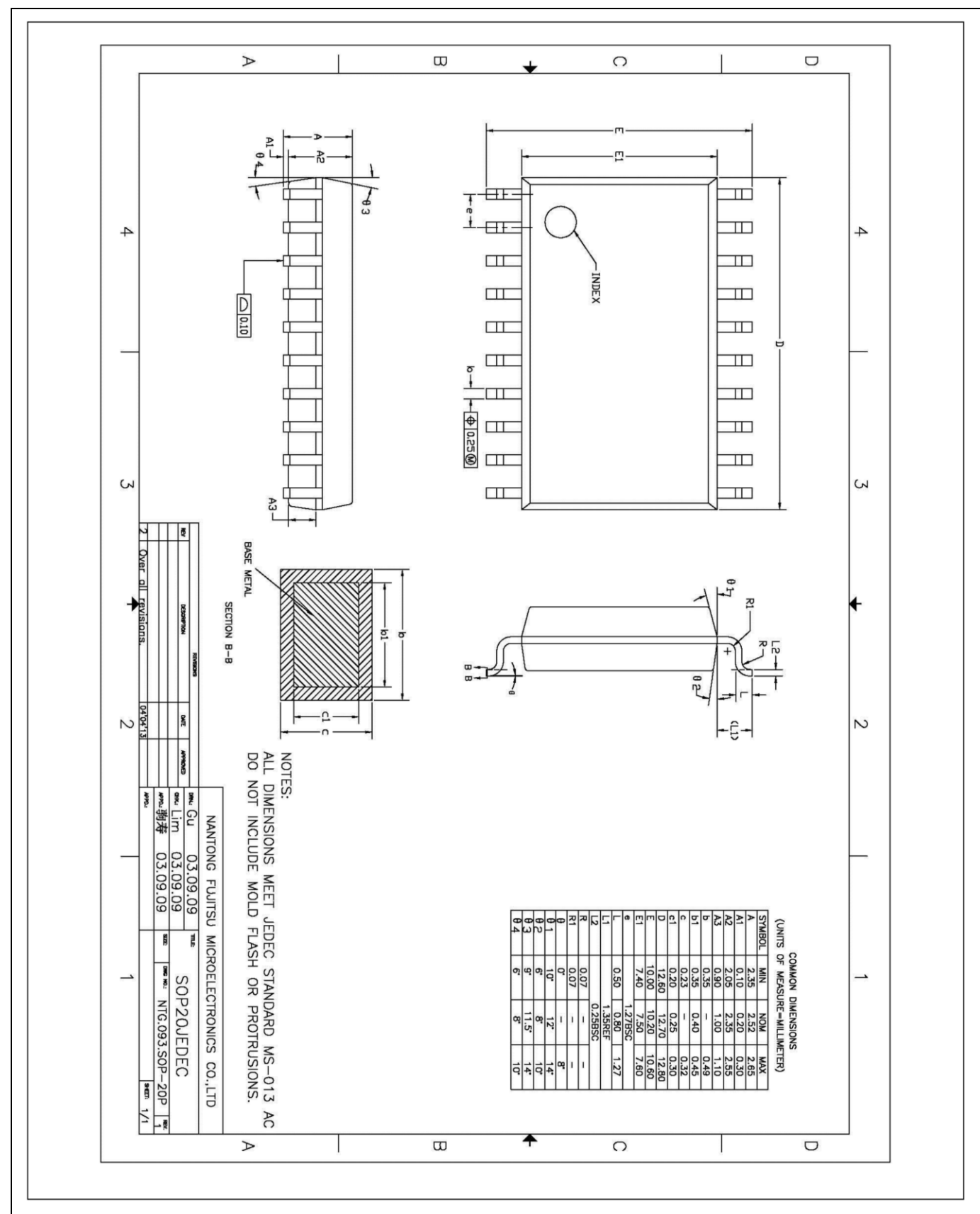


Figure 4.1 20-Pin SOP 封装

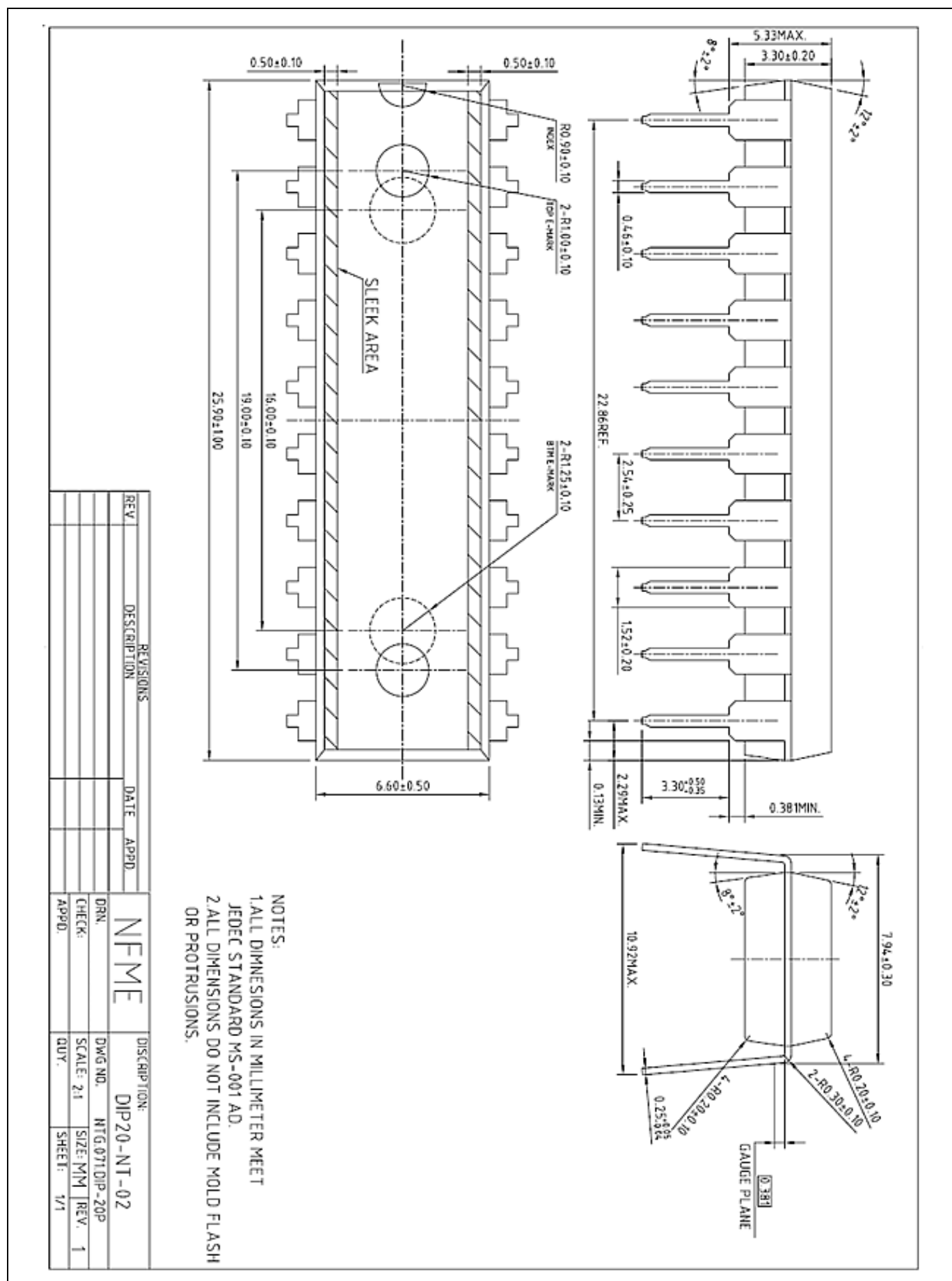


Figure 4.2 20-Pin PDIP 封装

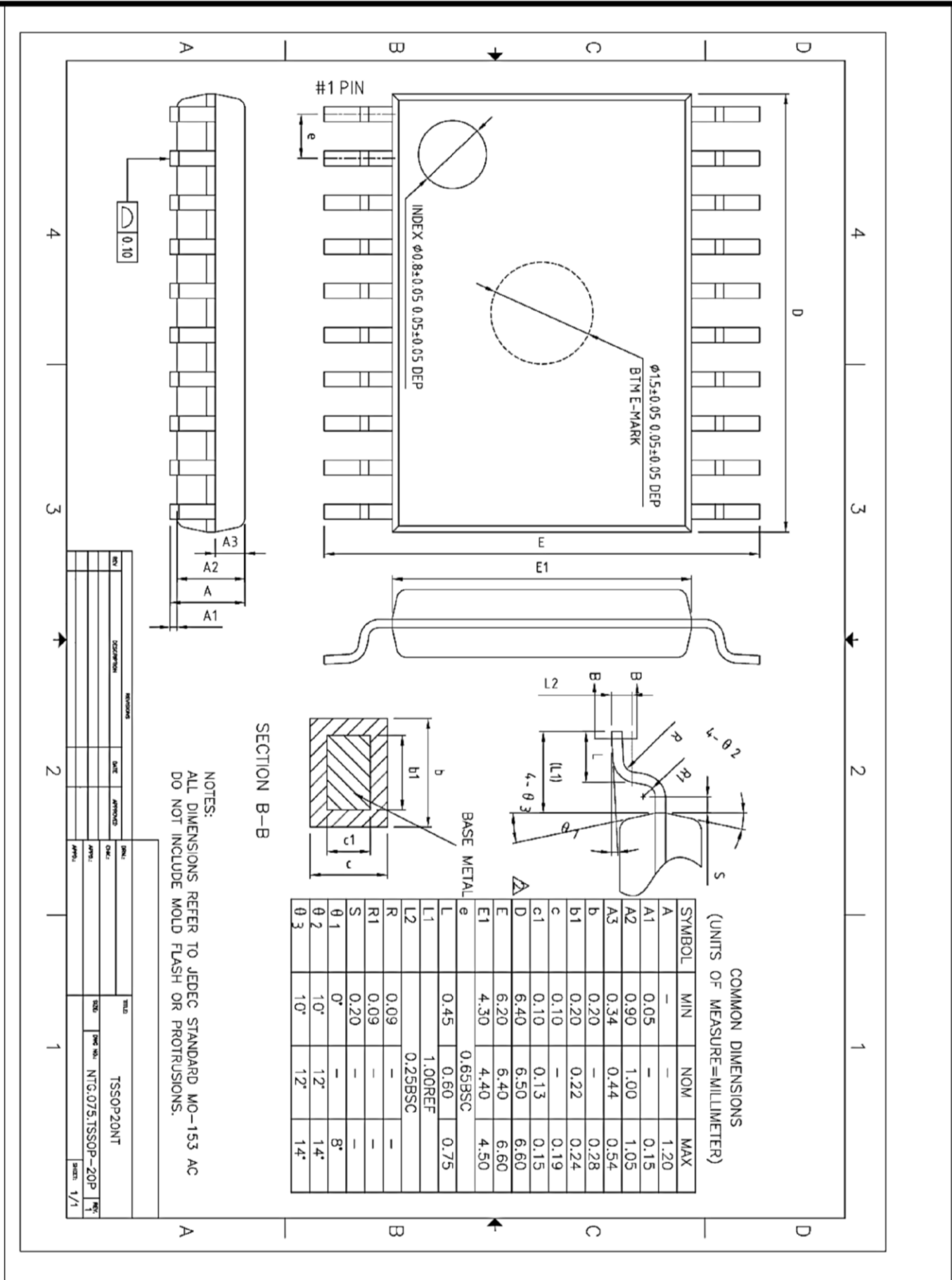


Figure 4.3 20-Pin TSSOP 封装

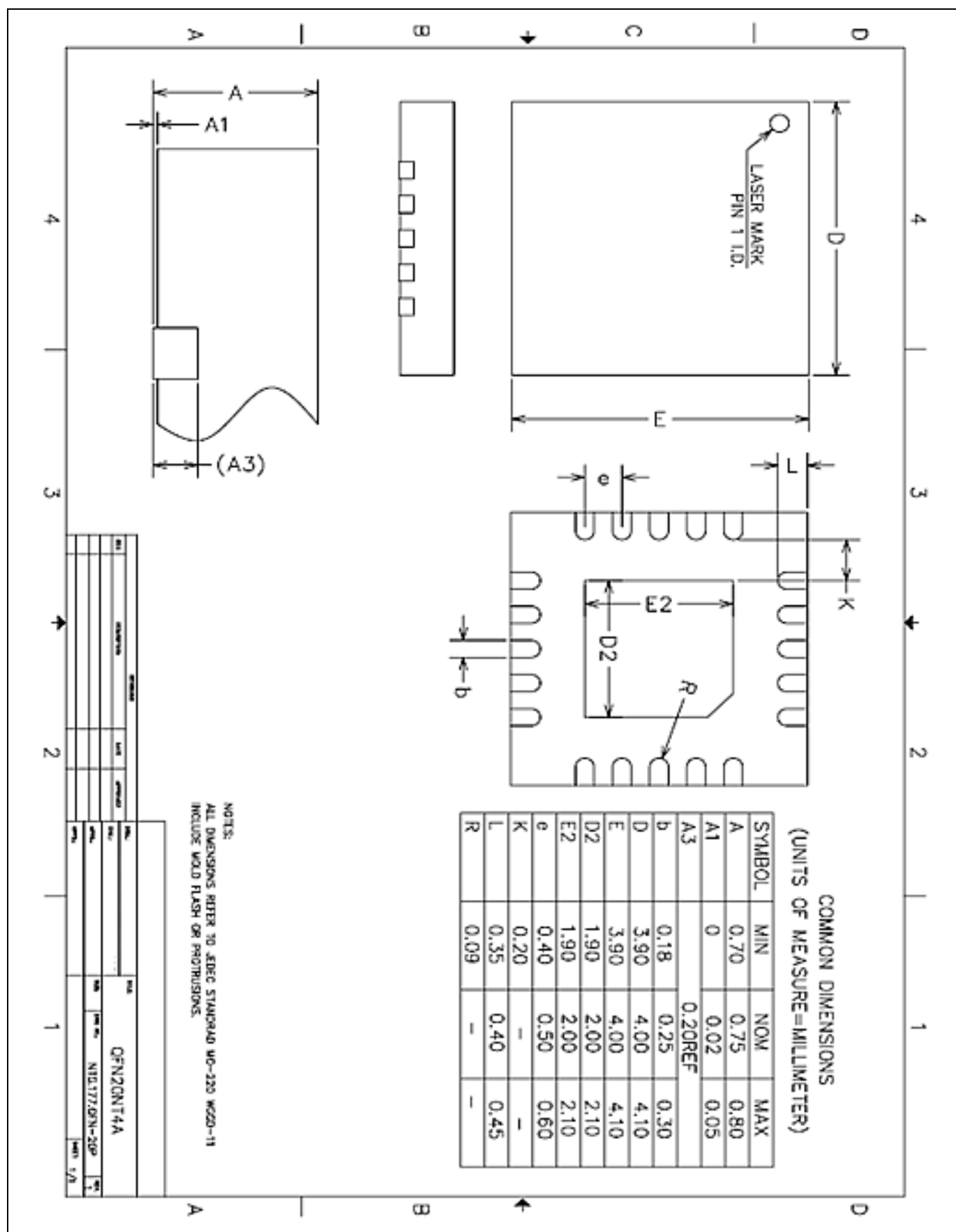


Figure 4.4 20-Pin QFN 封装

5 引脚说明

PIN Name	I/O	Function	@RESET	Shared with
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
P11	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 I/O口。 推-拉寄存器可指定到每一位	Input	AN8/EINT6/EC1/BUZO
P12				AN9/EINT11/T1O/PWM1O
P13				AN10/EINT12/T2O/PWM2O
P14				AN11/MISO
P15				AN12/MOSI
P16				AN13/SCK
P24	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 I/O口。 推-拉寄存器可指定到每一位	Input	SDA
P25				SCL
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 I/O口。 推-拉寄存器可指定到每一位	Input	TXD
P31				RXD
P32				RESETB
P35				EINT10/T0O/PWM0O
P36				XIN
P37				XOUT
EINT0	I/O	外部中断输入	Input	P02/AN2/AVREF
EINT1				P03/AN3
EINT6				P11/AN8/EC1/BUZO
EINT10	I/O	外部中断输入和定时器0 捕捉输入	Input	P35/T0O/PWM0O
EINT11	I/O	外部中断输入和定时器1 捕捉输入	Input	P12/AN9/T1O/PWM1O
EINT12	I/O	外部中断输入和定时器2 捕捉输入	Input	P13/AN10/T2O/PWM2O
T0O	I/O	Timer 0 interval output	Input	P35/EINT10/PWM0O
T1O	I/O	Timer 1 interval output	Input	P12/AN9/EINT11/PWM1O
T2O	I/O	Timer 2 interval output	Input	P13/AN10/EINT12/PWM2O
PWM0O	I/O	Timer 0 PWM 输出	Input	P35/EINT10/T0O
PWM1O	I/O	Timer 1 PWM 输出	Input	P12/AN9/EINT11/T1O
PWM2O	I/O	Timer 2 PWM 输出	Input	P13/AN10/EINT12/T2O
EC1	I/O	Timer 1 EC输入	Input	P11/AN8/EINT6/BUZO
BUZO	I/O	蜂鸣器信号输出	Input	P11/AN8/EINT6/EC1
SCK	I/O	串行时钟输入/输出	Input	P16/AN13
MISO	I/O	串行数据输入/输出	Input	P14/AN11
MOSI	I/O	串行数据输入/输出	Input	P15/AN12
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

Table 5.1 普通口说明

PIN Name	I/O	Function	@RESET	Shared with
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
AN8				P11/EINT6/EC1/BUZO
AN9				P12/EINT11/T1O/PWM1O
AN10				P13/EINT12/T2O/PWM2O
AN11				P14/MISO
AN12				P15/MOSI
AN13				P16/SCK
RESETB	I/O	作为复位脚使用时通过 CONFIGURE OPTION将系统复位脚设置上拉	Input	P32
DSDA	I/O	烧录器数据输入/输出 (NOTE4,5)	Input	P00
DSCL	I/O	烧录器时钟输入 (NOTE4,5)	Input	P01
XIN	I/O	主时钟	Input	P36
XOUT				P37
VDD, VSS	—	电源输入	—	—

Table 5.1 普通口说明 (Concluded)

NOTE)

1. P32/RESETB 通过“CONFIGURE OPTION”设置来选择作为普通口还是复位脚.
2. P37/XOUT 和 P36/XIN 通过软件配置为复用口.
3. 如果 P00 和 P01 在上电复位过程中和烧录器连接, 该脚自动配置为 ISP 脚.
4. P00 和 P01 只有在复位和上电复位过程中配置为内部上拉输入.

6 端口结构

6.1 通用 I/O 口

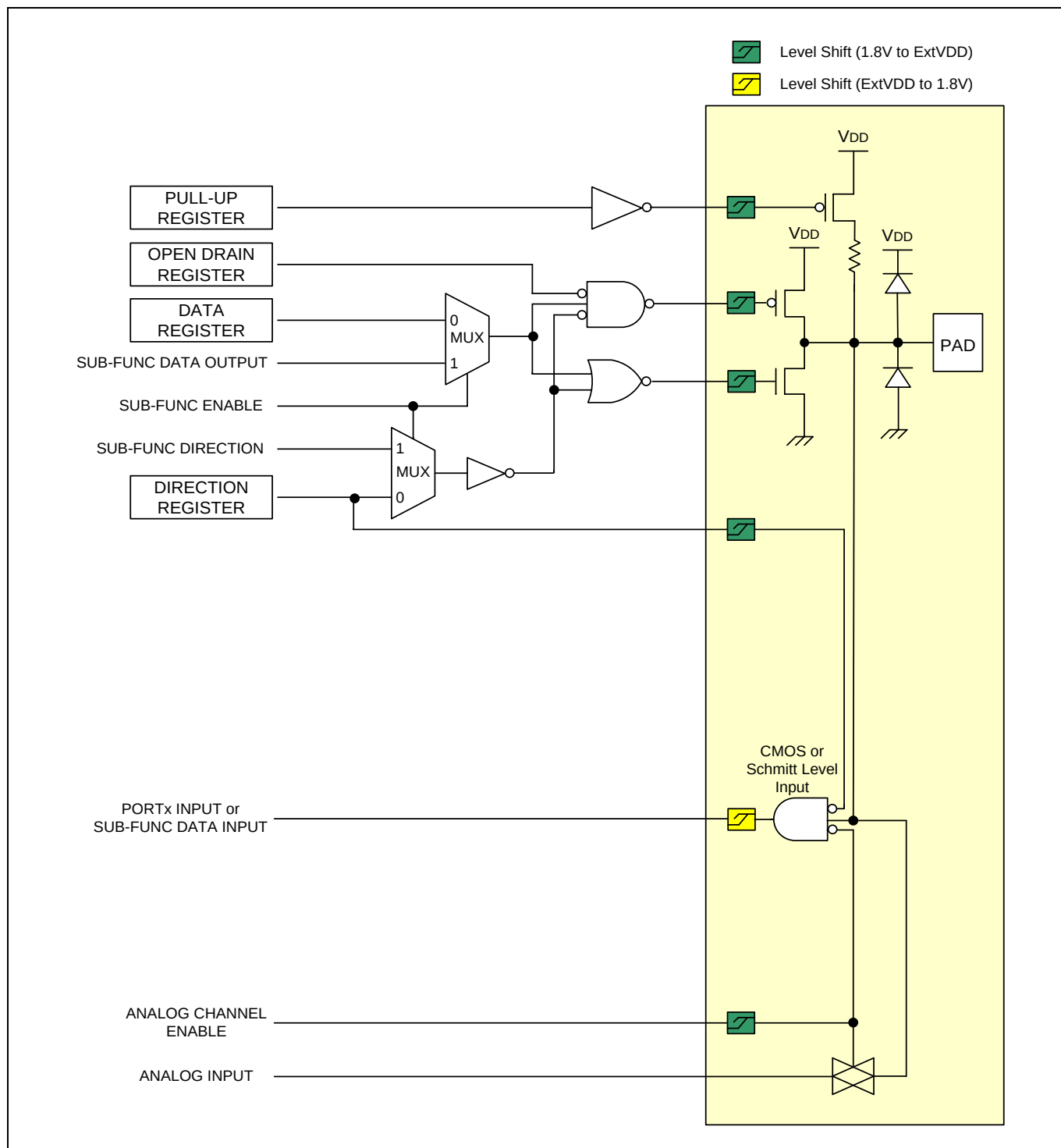


Figure 6.1 通用 I/O 口

6.2 外部中断 I/O 口

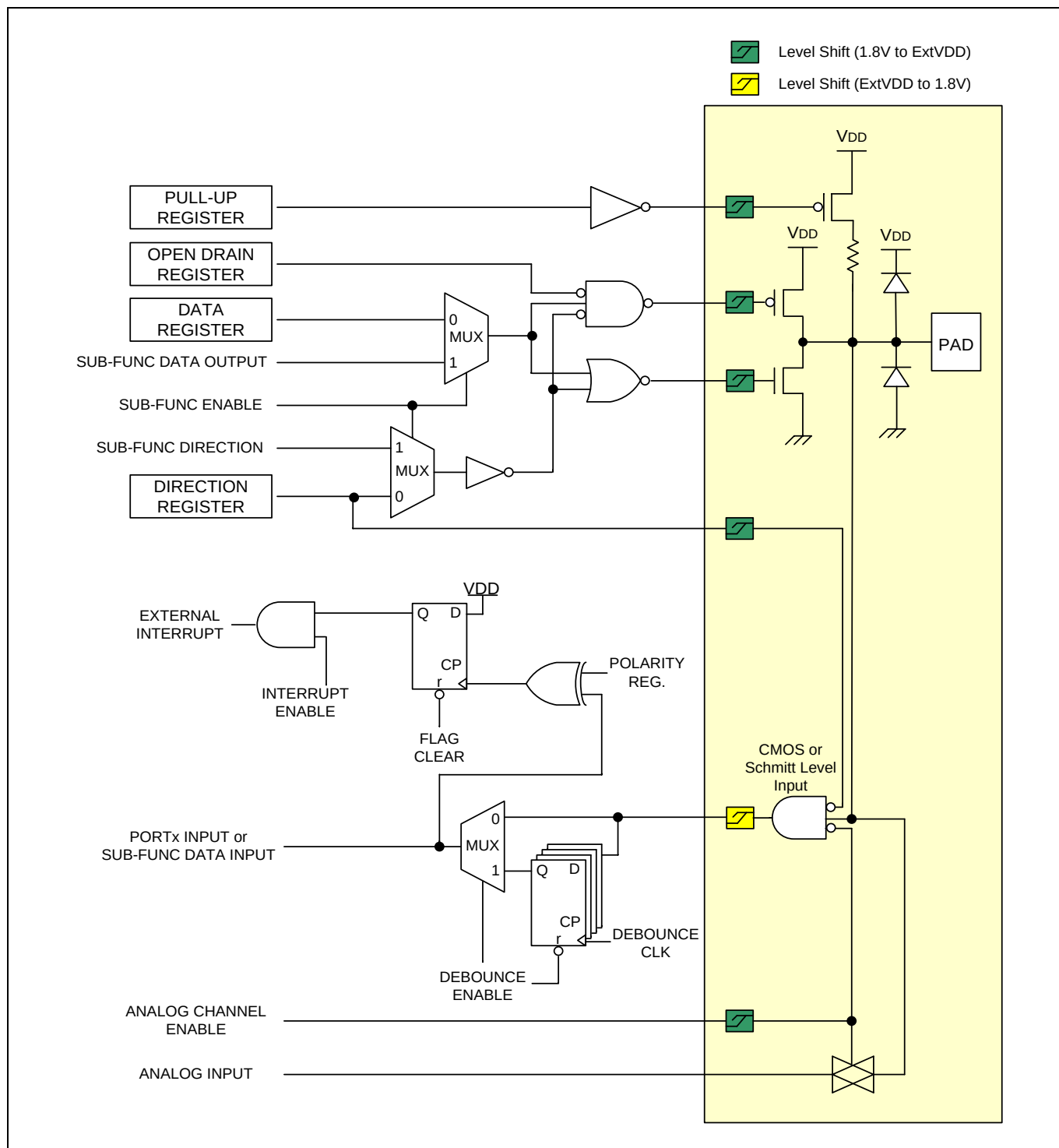


Figure 6.2 外部中断 I/O 口

7 电气特性

7.1 额定参数

Parameter	Symbol	Rating	Unit	Note
工作电压	VDD	-0.3 ~ +6.5	V	—
常规引脚电压	V _I	-0.3 ~ VDD+0.3	V	Voltage on any pin with respect to VSS
	V _O	-0.3 ~ VDD+0.3	V	
	I _{OH}	-25	mA	Maximum current output sourced by (I _{OH} per I/O pin)
	ΣI _{OH}	-200	mA	Maximum current (ΣI _{OH})
	I _{OL}	180	mA	Maximum current sunk by (I _{OL} per I/O pin)
	ΣI _{OL}	200	mA	Maximum current (ΣI _{OL})
总功率消耗	P _T	600	mW	—
存放温度	T _{STG}	-65 ~ +150	°C	—

Table 7.1 额定参数

注意)

1. 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。

7.2 推荐操作参数

(T_A=-40°C ~ +85°C)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
操作电压	VDD	f _x = 0.4 ~ 4.2MHz	Main Crystal	1.8	—	5.5	V
		f _x = 0.4 ~ 10.0MHz		2.7	—	5.5	
		f _x = 0.4 ~ 12.0MHz		3.0	—	5.5	
		f _x = 0.5 ~ 8.0MHz	Internal RC	1.8	—	5.5	
		f _x = 0.5 ~ 16.0MHz		2.0	—	5.5	
工作温度	T _{OPR}	VDD= 1.8 ~ 5.5V		-40	—	85	°C

Table 7.2 推荐工作参数

7.3 A/D 转换特性

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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
分辨率	—	—	—	12	—	bit
积分线性误差	ILE	AVREF= 2.7V ~ 5.5V fx= 8MHz	—	—	±6	LSB
微分线性误差	DLE		—	—	±1	
零点偏移误差	ZOE		—	—	±5	
满量程误差	FSE		—	—	±5	
转换时间	t _{CON}	12-bit resolution, 8MHz	20	—	—	μs
模拟输入电压	V _{AN}	—	VSS	—	AVREF	V
模拟参考电压	AVREF	—	1.8	—	VDD	
VDD18	—	—	—	1.8	—	V
模拟输入漏电流	I _{AN}	AVREF=5.12V	—	—	2	μA
ADC 工作电流	I _{ADC}	Enable	—	1	2	mA
		Disable	—	—	0.1	uA

Table 7.3 A/D 转换特性

NOTE)

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).

7.4 上电复位特性

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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
复位电平	V _{POR}	—	—	1.4	—	V
VDD 电压上升时间	t _R	—	0.05	—	30.0	V/ms
POR 电流	I _{POR}	—	—	0.2	—	μA

Table 7.4 上电复位特性

7.5 低电压复位和低电压检测特性

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

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
检测电平	V _{LVR} V _{LVI}	LVR 可选择电平 LVI 可选择除1.60V以外其他电平		—	1.60	1.79	V
				1.85	2.00	2.15	
				1.95	2.10	2.25	
				2.05	2.20	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	
				3.70	4.00	4.30	
				4.10	4.40	4.70	
滞后	△V	—		—	50	150	mV
最小脉冲宽度	t _{LOW}	—		100	—	—	us
LVR 和 LVI 电流	I _{BL}	使能 (Both)	VDD= 3V, RUN Mode	—	14.0	24.0	uA
		使能 (One of two)		—	10.0	18.0	
		禁止 (Both)	VDD= 3V	—	0.1	0.1	

Table 7.5 LVR 和 LVI 特性

7.6 内部高速 RC 振荡特性

(T_A=-40°C ~ +85°C, V_{DD}=1.8V ~ 5.5V, V_{SS}=0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Frequency	f _{IRC}	V _{DD} = 2.0 – 5.5V	–	16	–	MHz
Tolerance	–	T _A = 0°C to +50°C	–	–	±1.5	%
		T _A = -20°C to +85°C			±2.5	
		T _A = -40°C to +85°C			±3.5	
Clock Duty Ratio	TOD	–	40	50	60	%
Stabilization Time	T _{HFS}	–	–	–	100	us
IRC Current	I _{IRC}	Enable	–	0.2	–	mA
		Disable	–	–	0.1	uA

Table 7.6 内部 RC 振荡特性

NOTE)

1. V_{DD} 和 V_{SS} 之间需要连接一个 0.1uF 旁路电容器.

7.7 内部 WDT RC 振荡特性

(T_A=-40°C ~ +85°C, V_{DD}=1.8V ~ 5.5V, V_{SS}=0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
频率	f _{WDTRC}	–	2	5	10	kHz
稳定时间	t _{WDTS}	–	–	–	1	ms
WDTRC 电流	I _{WDTRC}	Enable	–	1	–	uA
		Disable	–	–	0.1	

Table 7.7 内部 WDTRC 振荡特性

7.8 DC 特性

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

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
输入高电压	V _{IH}	All input pins, RESETB		0.8VDD	—	VDD	V
输入低电压	V _{IL}	All input pins, RESETB		—	—	0.2VDD	V
输出高电压	V _{OH1}	VDD= 4.5V, I _{OH} = -2mA, All output ports		VDD-1.0	—	—	V
输出低电压	V _{OL1}	VDD=4.5V, I _{OL} = 15mA; All output ports		—	—	1.0	V
输入高漏电流	I _{IH}	All input ports		—	—	1	μA
输入低漏电流	I _{IL}	All input ports		-1	—	—	μA
上拉电阻	R _{PU1}	VI=0V, T _A = 25°C All Input ports	VDD=5.0V	25	50	100	kΩ
			VDD=3.0V	50	100	200	
	R _{PU2}	VI=0V, T _A = 25°C RESETB	VDD=5.0V	150	250	400	kΩ
			VDD=3.0V	300	500	700	
ADC唤醒上拉电阻	R _{AWPU1}	T _A = 25°C		100	150	200	kΩ
	R _{AWPU2}			200	300	400	
OSC 反馈电阻	R _{X1}	XIN= VDD, XOUT= VSS T _A = 25°C, VDD= 5V		600	1200	2000	kΩ
工作电压	I _{DD1} (RUN)	f _{XIN} = 12MHz, VDD= 5V±10%		—	3.0	6.0	
		f _{XIN} = 10MHz, VDD= 3V±10%		—	2.2	4.4	
		f _{IRC} = 16MHz, VDD= 5V±10%		—	3.0	6.0	
	I _{DD2} (IDLE)	f _{XIN} = 12MHz, VDD= 5V±10%		—	1.3	2.6	
		f _{XIN} = 10MHz, VDD= 3V±10%		—	0.7	1.4	
		f _{IRC} = 16MHz, VDD= 5V±10%		—	0.8	1.6	
	I _{DD5}	STOP, VDD= 5V±10%, T _A = 25°C		—	0.5	3.0	

Table 7.8 DC 特性

NOTE)

1. f_{XIN} 是外部主时钟, f_{IRC} 是内部 RC, fx 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).

7.9 AC 特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
复位输入低电压宽度	tRSL	Input, VDD= 5V	10	—	—	us
中断输入高低宽度	tINTH, tINTL	All interrupt, VDD= 5V	200	—	—	ns
外部计数器输入高低脉冲宽度,	tECWH, tECWL	EC1, VDD = 5V	200	—	—	
外部计数器转换时间	tREC, tFEC	EC1, VDD = 5V	20	—	—	

Table 7.9 AC 特性

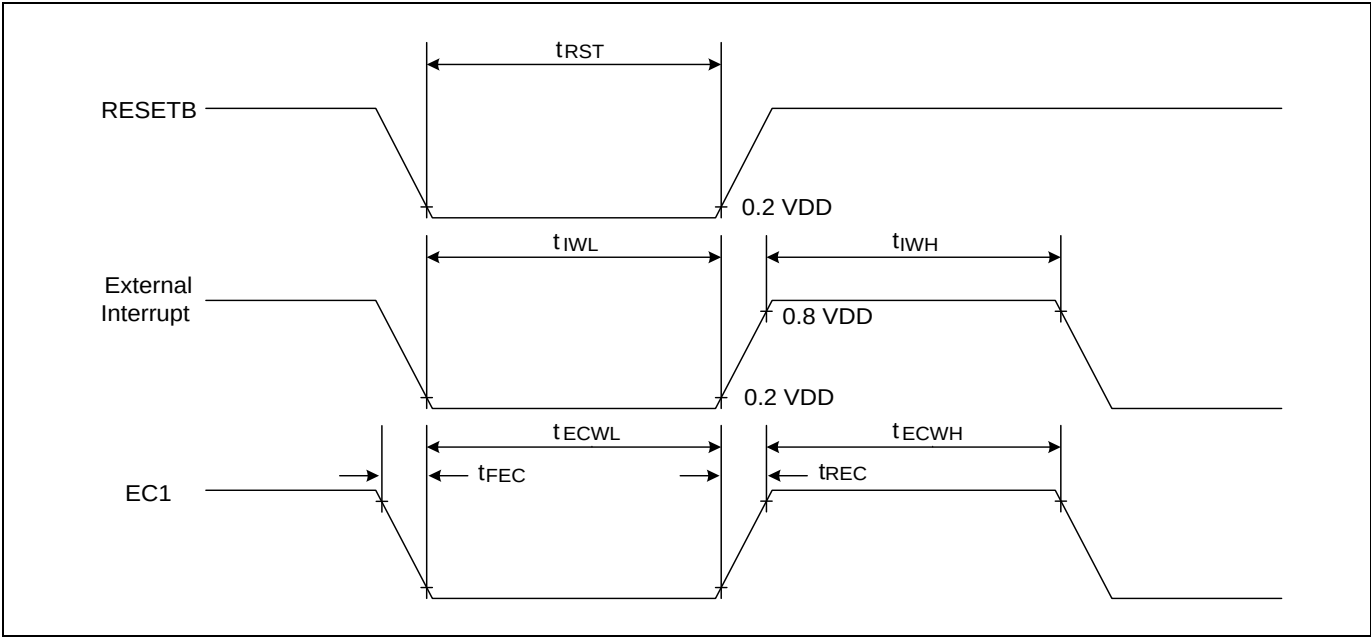


Figure 7.1 AC 时序

7.10 SPI 特性

(T_A=-40°C~ +85°C, VDD=1.8V ~ 5.5V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
输出脉冲周期	t _{SCK}	Internal SCK source	200	—	—	ns
输入脉冲周期		External SCK source	200	—	—	
输出时钟高低脉冲宽度	t _{SCKH} , t _{SCKL}	Internal SCK source	70	—	—	
输入时钟脉冲高低宽度		External SCK source	70	—	—	
第一个输出时钟延时时间	t _{FOD}	Internal/External SCK source	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	—	—	

Table 7.10 SPI 特性

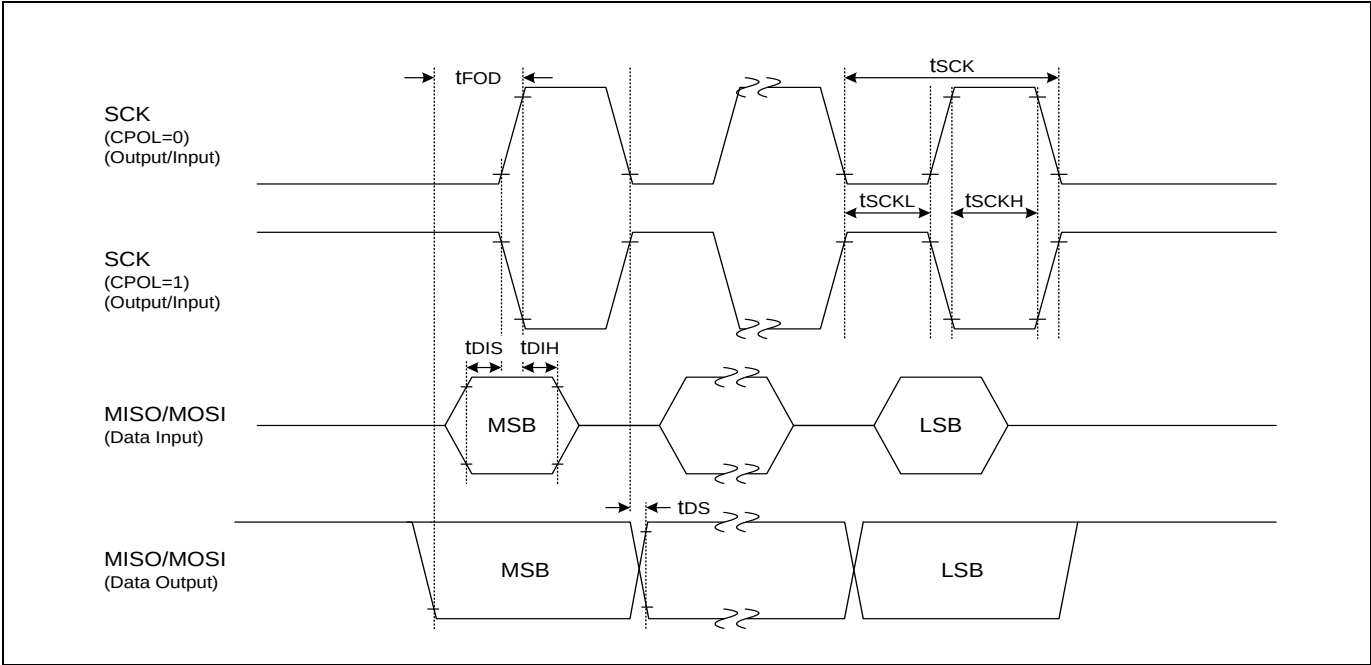


Figure 7.2 SPI 时序

7.11 UART 特性

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

Parameter	Symbol	MIN	TYP	MAX	Unit
串口时钟周期	t_{SCK}	1250	$t_{CPU} \times 16$	1650	ns
输出数据的设置到时钟上升沿	t_{S1}	590	$t_{CPU} \times 13$	—	ns
时钟上升沿到输入有效数据时间	t_{S2}	—	—	590	ns
时钟上升沿之后输出数据维持时间	t_{H1}	$t_{CPU}-50$	t_{CPU}	—	ns
时钟上升沿之后输入数据维持时间	t_{H2}	0	—	—	ns
串口时钟高低电平宽度	t_{HIGH}, t_{LOW}	470	$t_{CPU} \times 8$	970	ns

Table 7.11 UART 特性

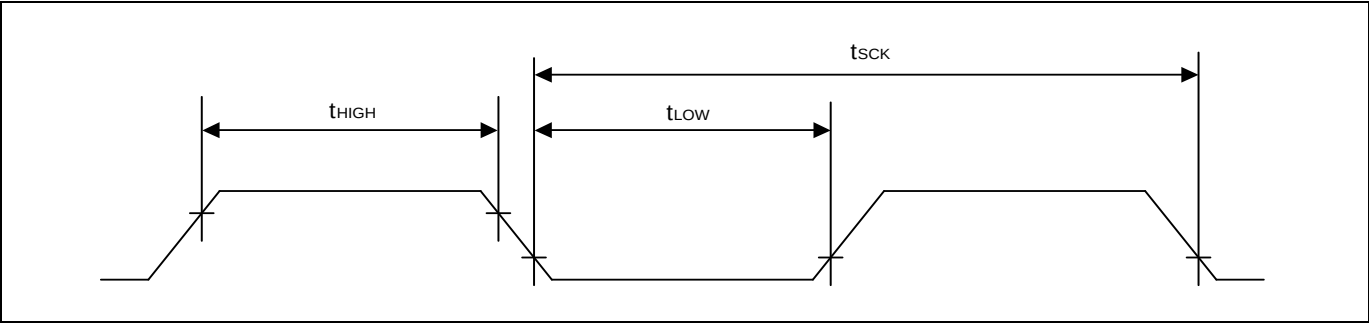


Figure 7.3 UART 时序波形

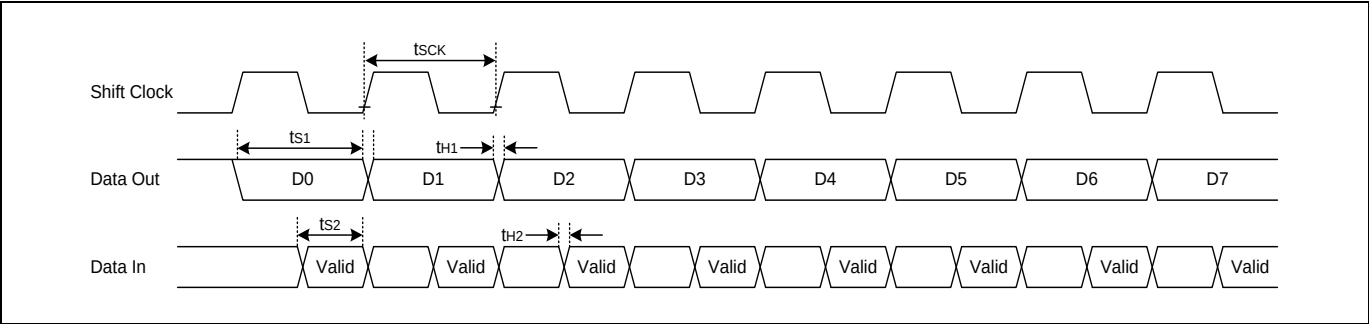


Figure 7.4 UART 模块时序波形图

7.12 I2C 特性

(T_A=-40°C ~ +85°C, VDD=1.8V ~ 5.5V)

Parameter	Symbol	Standard Mode		High-Speed Mode		Unit
		MIN	MAX	MIN	MAX	
时钟频率	t _{SCL}	0	100	0	400	kHz
时钟高脉冲宽度	t _{SCLH}	4.0	—	0.6	—	
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	—	0.6	—	
停止条件设置时间	t _{SPSU}	4.0	—	0.6	—	
停止条件维持时间	t _{SPHD}	4.0	—	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	—	ns

Table 7.12 I2C 特性

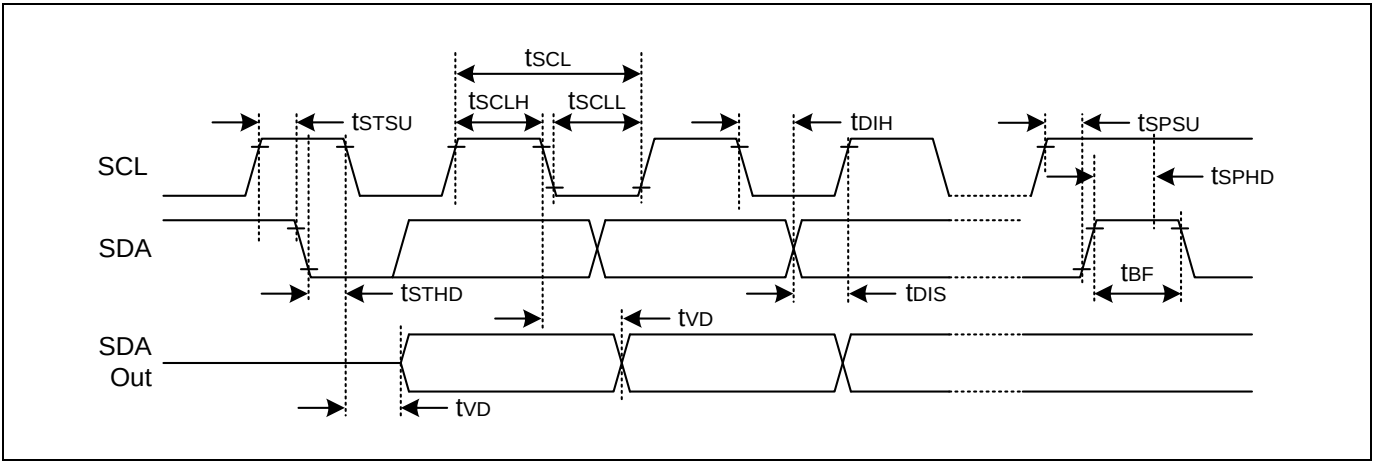


Figure 7.5 I2C 时序

7.13 Stop 模式数据保持电压

(T_A=-40°C ~ +85°C, VDD=1.8V ~ 5.5V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
保持数据电压	V _{DDDR}	—	1.8	—	5.5	V
保持数据电流	I _{DDDR}	V _{DDR} = 1.8V, (T _A = 25°C), Stop mode	—	—	1	uA

Table 7.13 Stop 模式被中断唤醒时序

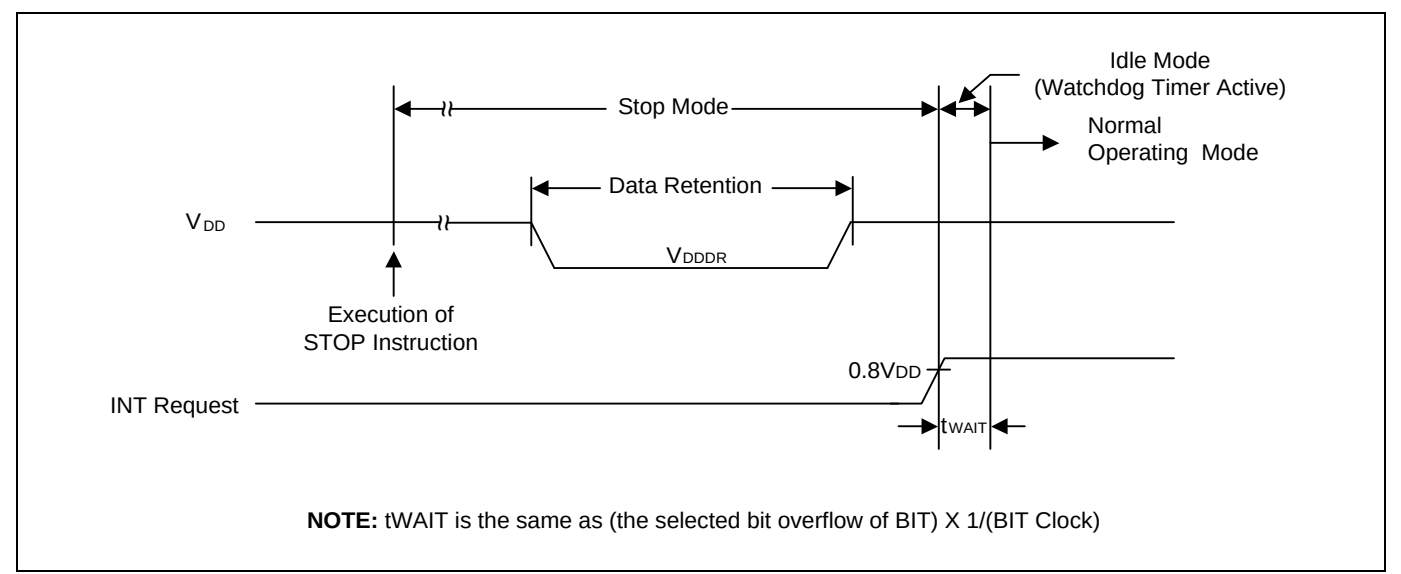


Figure 7.6 Stop 模式被中断唤醒时序

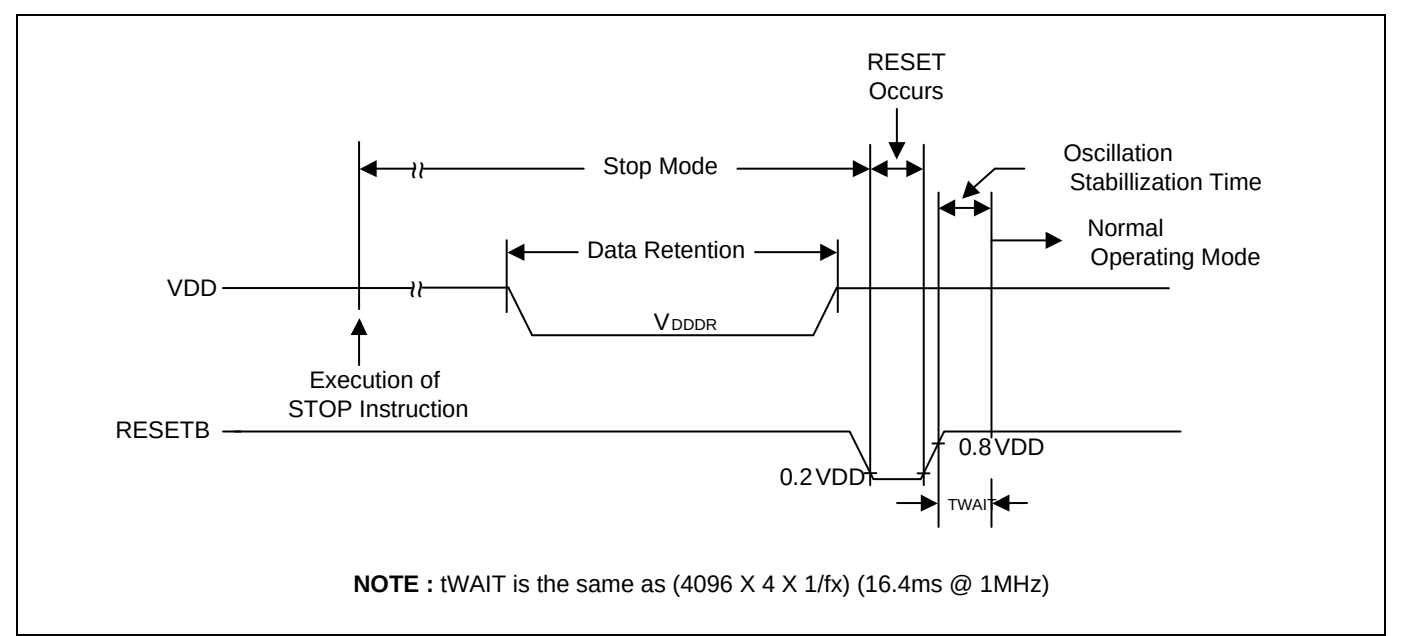


Figure 7.7 Stop 模式被复位唤醒时序

7.14 内部 Flash Rom 特性

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

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
页写时间	t _{FSW}	—	—	2.5	2.7	ms
页擦除时间	t _{FSE}	—	—	2.5	2.7	
Hard-Lock 时间	t _{FHL}	—	—	2.5	2.7	
页缓存复位时间	t _{FBR}	—	—	—	5	us
Flash烧写频率	f _{PGM}	—	0.4	—	—	MHz
擦写次数(Sector 0~247)	N _{FWE}	—	—	—	10,000	times
擦写次数(Sector 248~255)					100,000	

Table 7.14 内部 Flash Rom 特性

NOTE)

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

7.15 输入/输出 电容

(T_A=-40°C ~ +85°C, VDD=0V)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
输入电容	C _{IN}	f _x = 1MHz Unmeasured pins are connected to VSS	—	—	10	pF
输出电容	C _{OUT}					
I/O 口电容	C _{IO}					

Table 7.15 输入/输出 电容

7.16 主时钟振荡特性

(T_A=-40°C ~ +85°C, VDD=1.8V ~ 5.5V)

Oscillator	Parameter	Condition	MIN	TYP	MAX	Unit
晶体	主振荡频率	1.8V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	10.0	
		3.0V – 5.5V	0.4	–	12.0	
陶振	主振荡频率	1.8V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	10.0	
		3.0V – 5.5V	0.4	–	12.0	
外部时钟	XIN 输入频率	1.8V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	10.0	
		3.0V – 5.5V	0.4	–	12.0	

Table 7.16 主时钟振荡频率

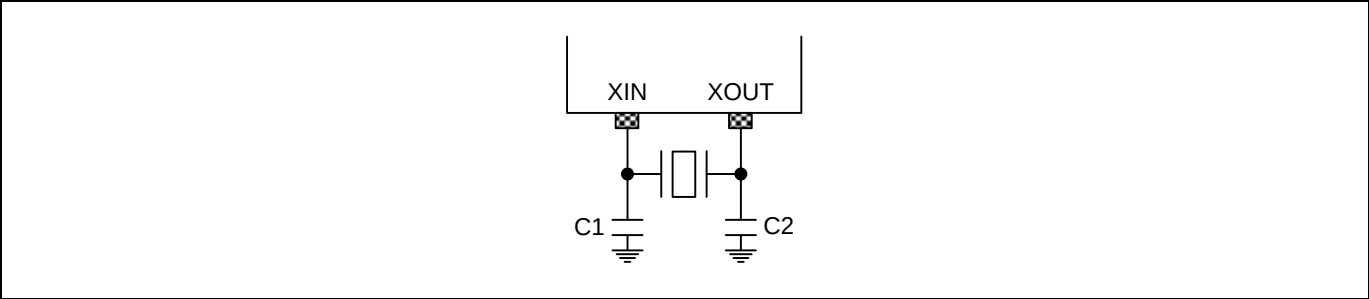


Figure 7.8 晶体/陶振

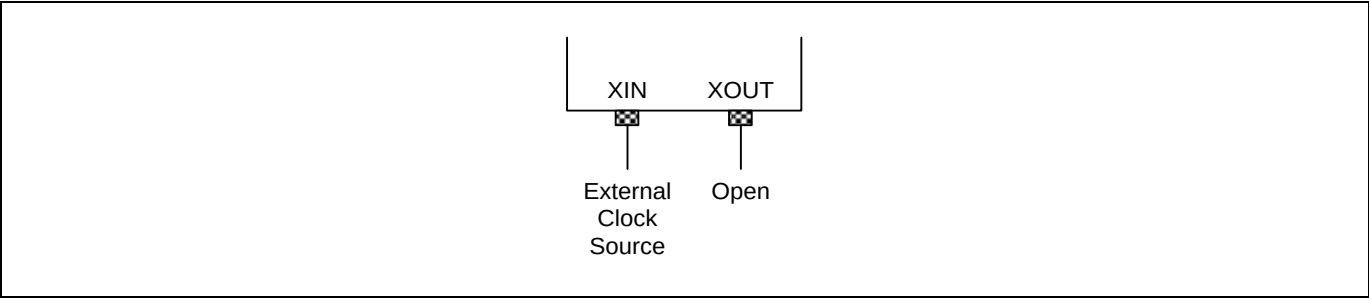


Figure 7.9 External Clock

7.17 主时钟稳定特性

(T_A=-40°C ~ +85°C, VDD=1.8V ~ 5.5V)

Oscillator	Parameter	MIN	TYP	MAX	Unit
晶体	f _x > 1MHz	—	—	60	ms
陶振	当 VDD 等于振荡器最小电压范围时振荡器开始稳定	—	—	10	ms
外部时钟	f _{XIN} = 0.4 to 12MHz XIN 输入高低宽度 (t _{xH} , t _{xL})	42	—	1250	ns

Table 7.17 主振荡器稳定特性

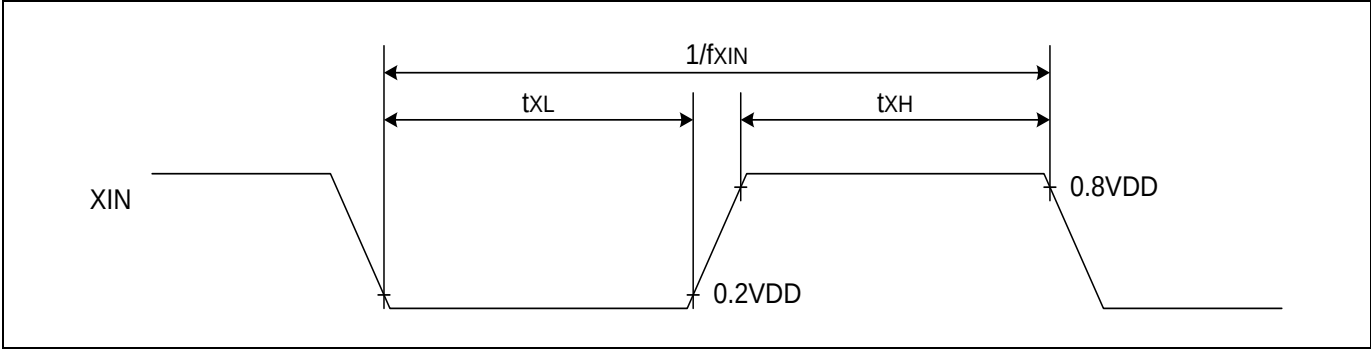


Figure 7.10 XIN 脚时钟时序

7.18 工作电压范围

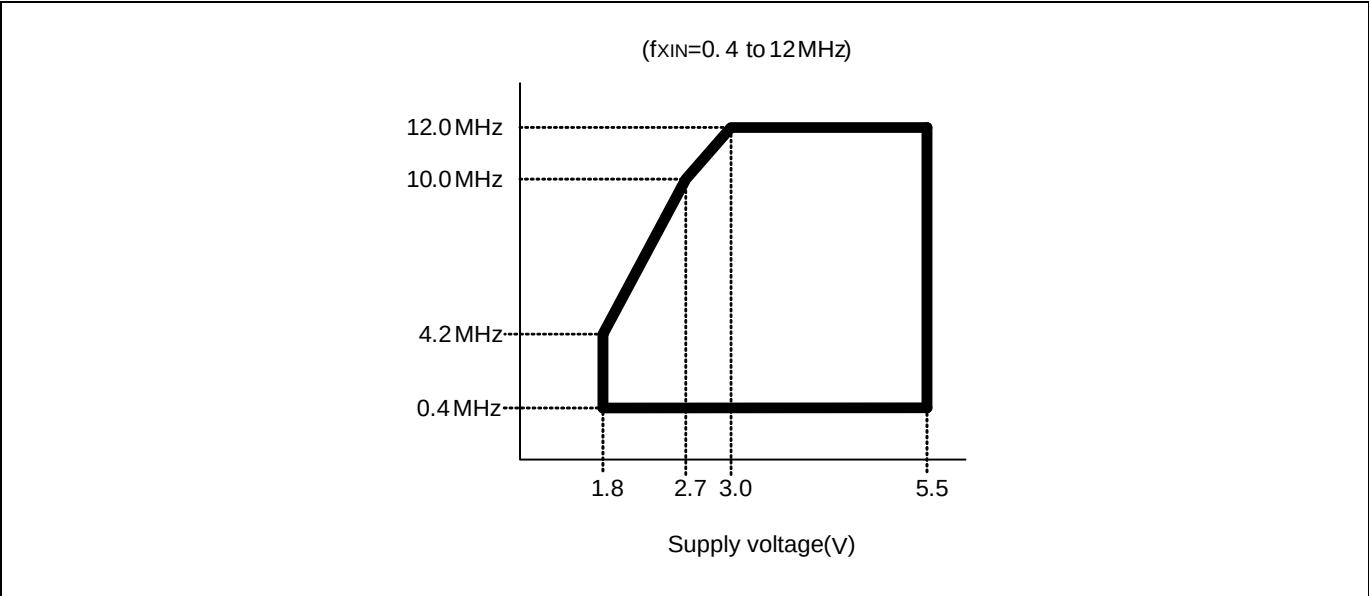


Figure 7.11 工作电压范围

7.19 推荐电路布局

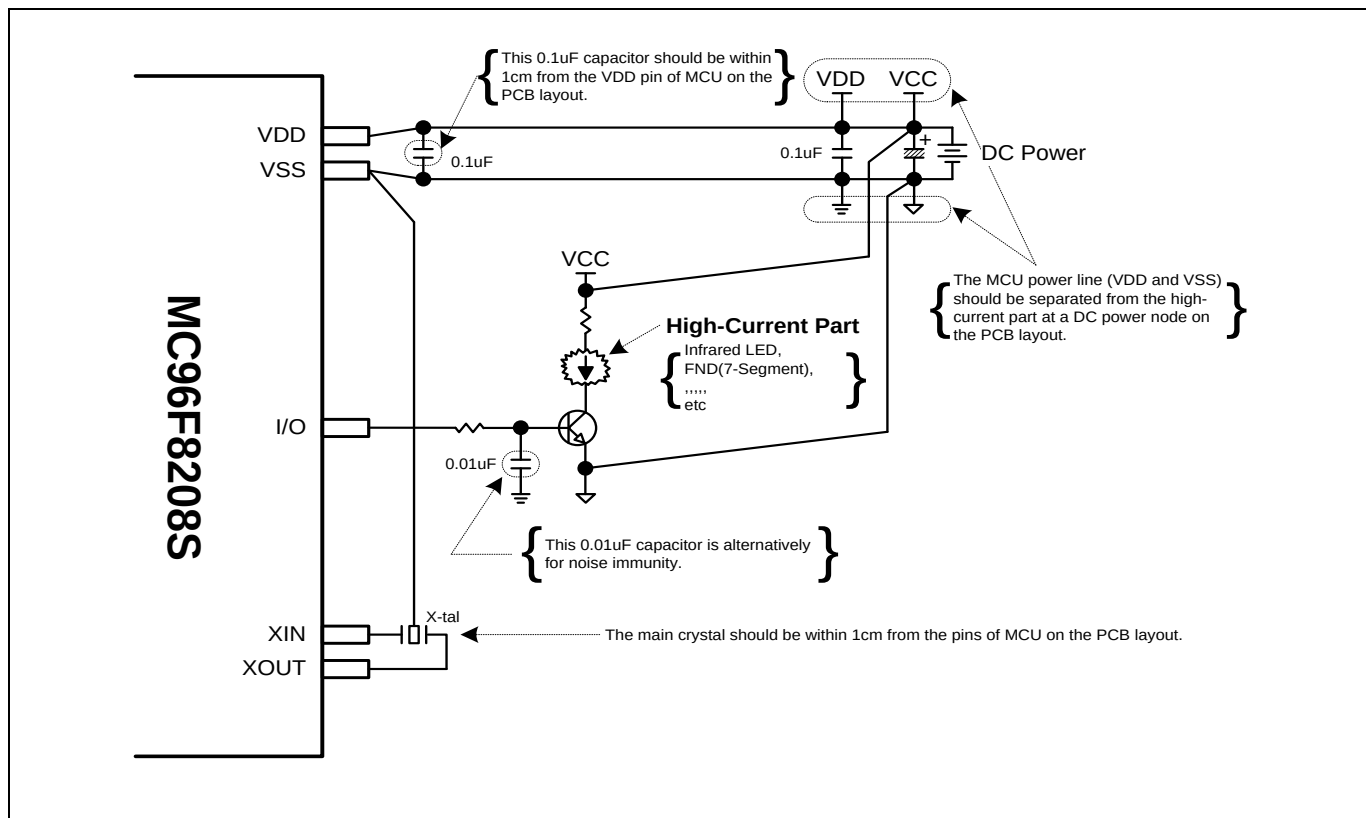


Figure 7.12 推荐电路布局

7.20 典型特性

本节的这些图标只是设计指导没有测试担保. 图标中的一些数据超出了指定的操作范围(e.g. out of specified VDD range). 只是作为操作参数内的参考.

本节的这些数据只是数据单元采集的概要, 在最大 (+ 3 σ)和最小 (-3 σ)之间的一些典型参数的误差标准.

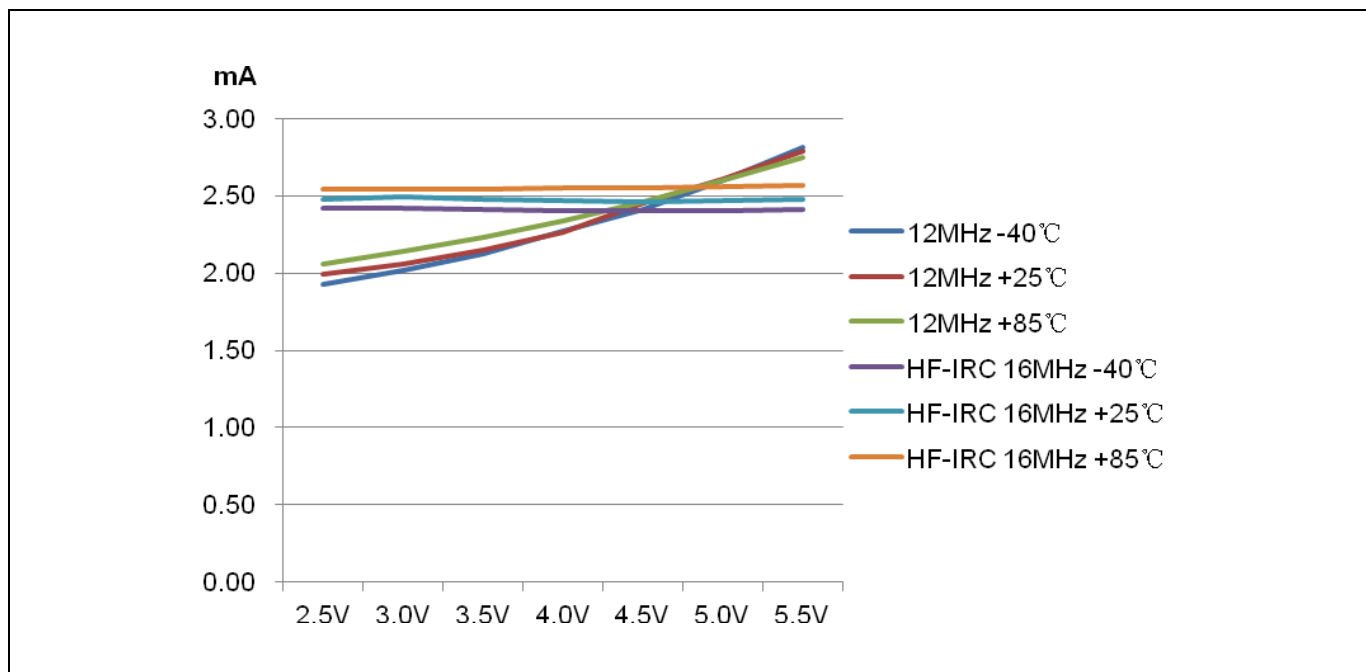


Figure 7.13 RUN (IDD1) 电流

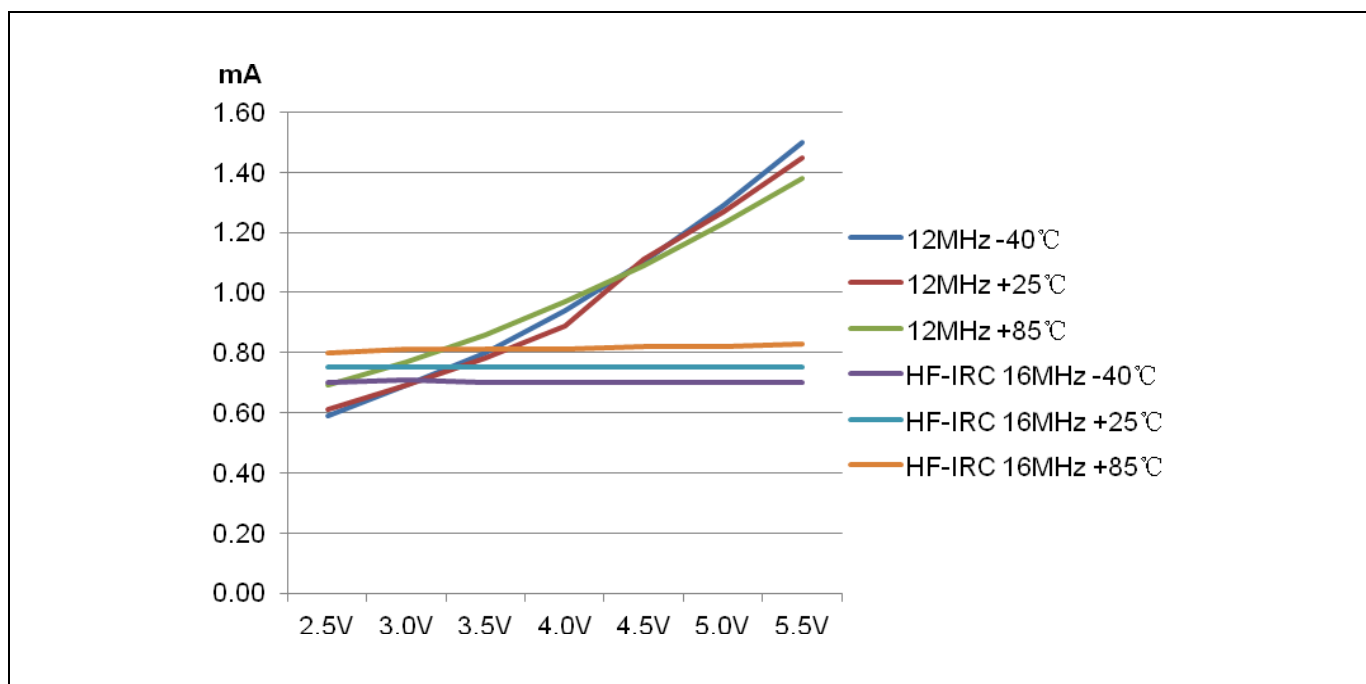


Figure 7.14 IDLE (IDD2) 电流

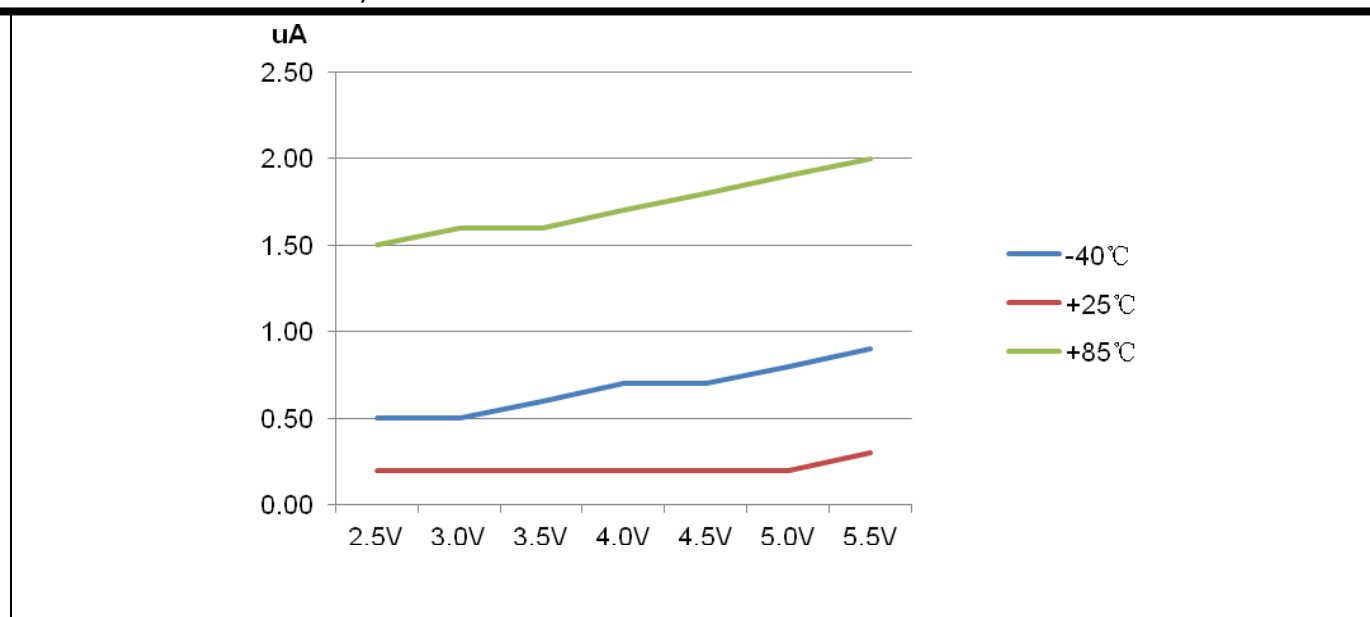


Figure 7.15 STOP (IDD5) 电流

8 存储器

MC96F8208S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.

MC96F8208S 具备片内 8k 字节 ISP 类型的可读写 flash 存储器,. 内部数据存储器(IRAM) 256 bytes 包括堆栈区域. 外部数据存储器 (XRAM) 256 bytes.

8.1 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 8k bytes 程序存储空间.

Figure 8-1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.

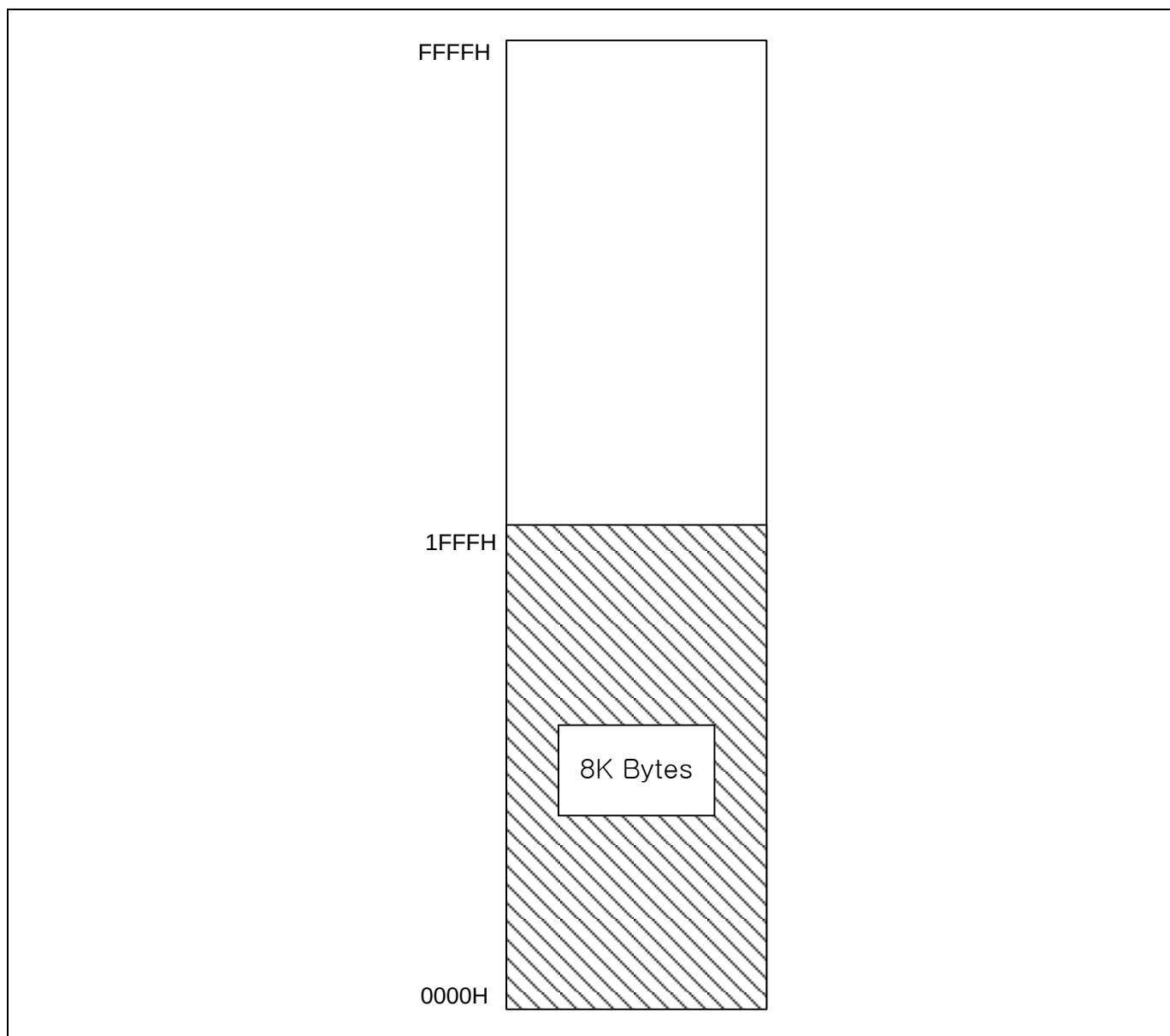


Figure 8.1 程序存储器

NOTE)

1. 8 Kbytes 包括中断向量区域

8.2 数据存储

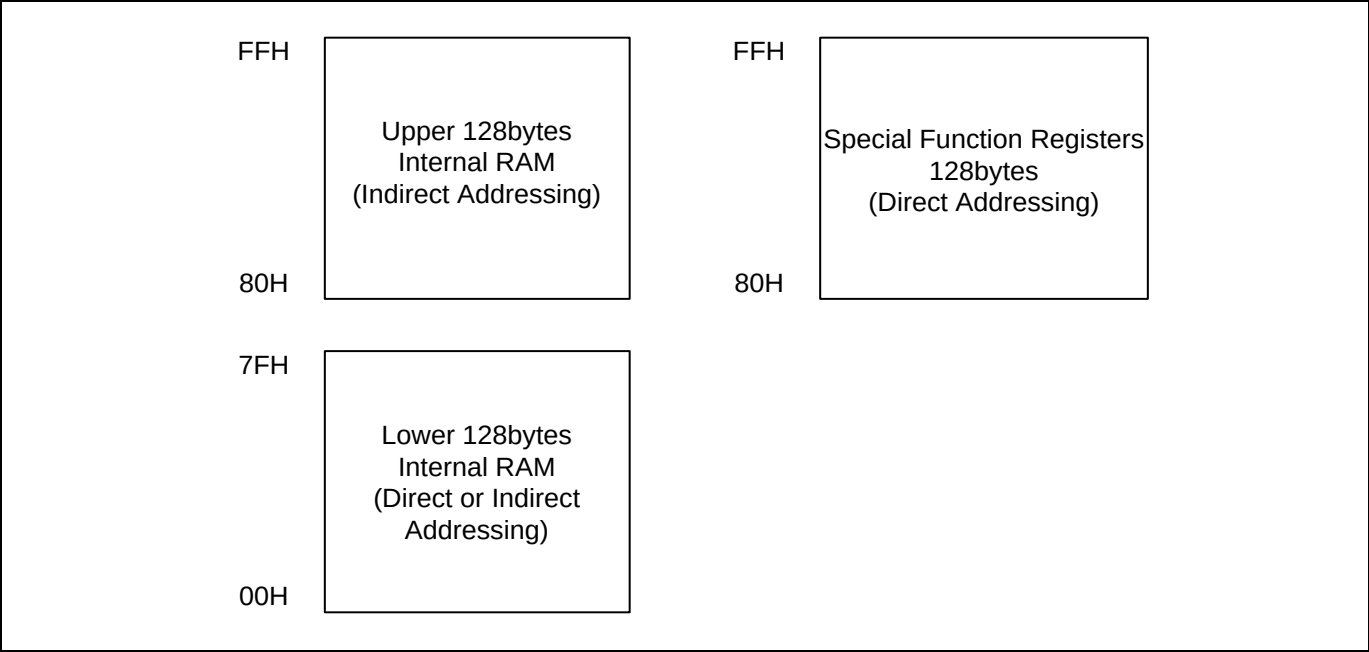


Figure 8.2 数据存储图

内部数据存储器分为3个部分， 低 128 bytes, 高 128 bytes, 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 256 bytes. 但是，事实上通过内部RAM寻址方式调节可提供384 bytes.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. 图 8-2所示高128 bytes 和SFR 占用相同的区域, 80H 到 FFH, 但他们本身却是独立的.

图 8-3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 R 7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 16 bytes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128 bits 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.

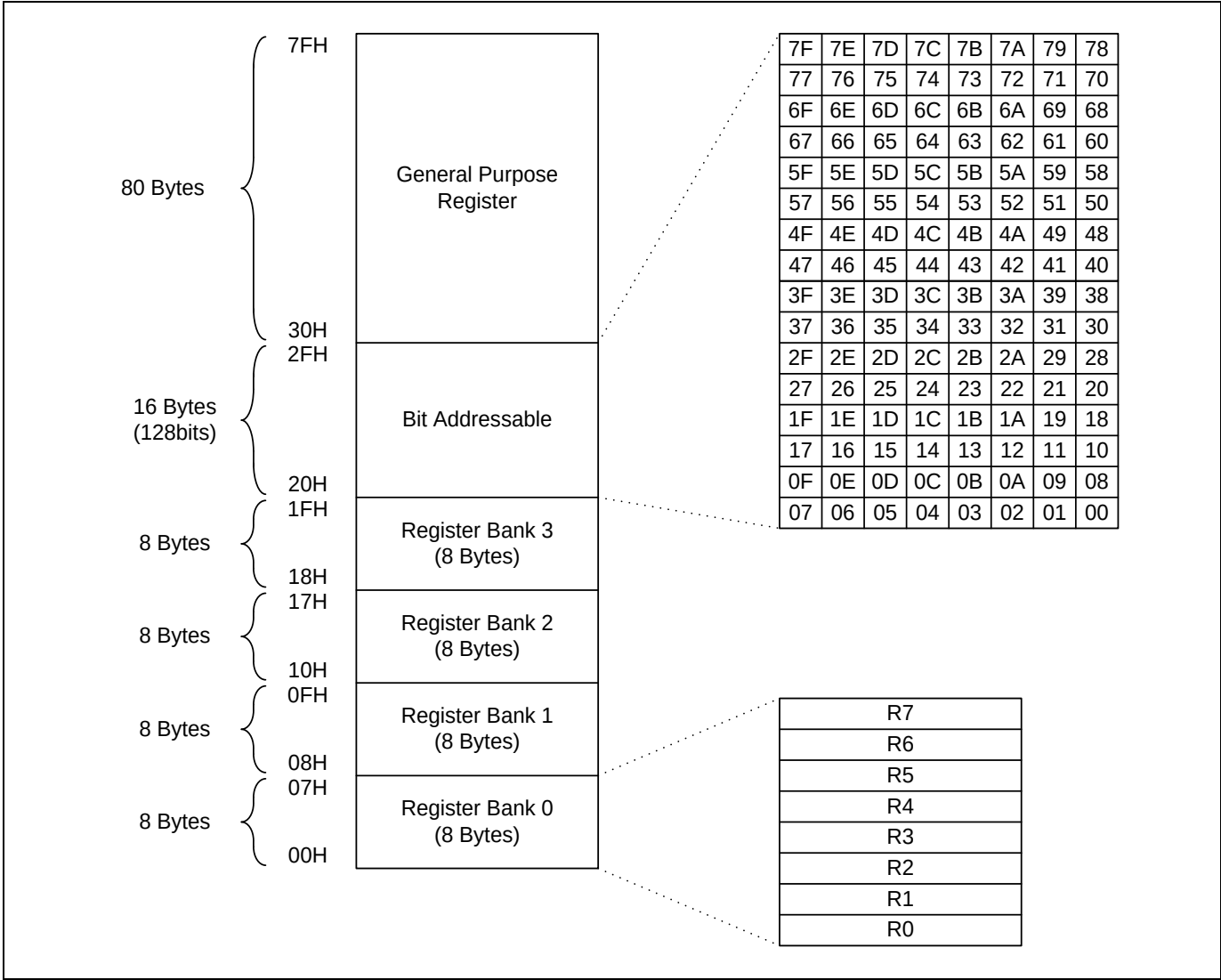


Figure 8.3 低 128bytes RAM

8.3 外部数据存储器

MC96F8208S 有 256 bytes XRAM. 该区域与 RAM/FLASH没有联系. 可通过8位的SFR进行读写.

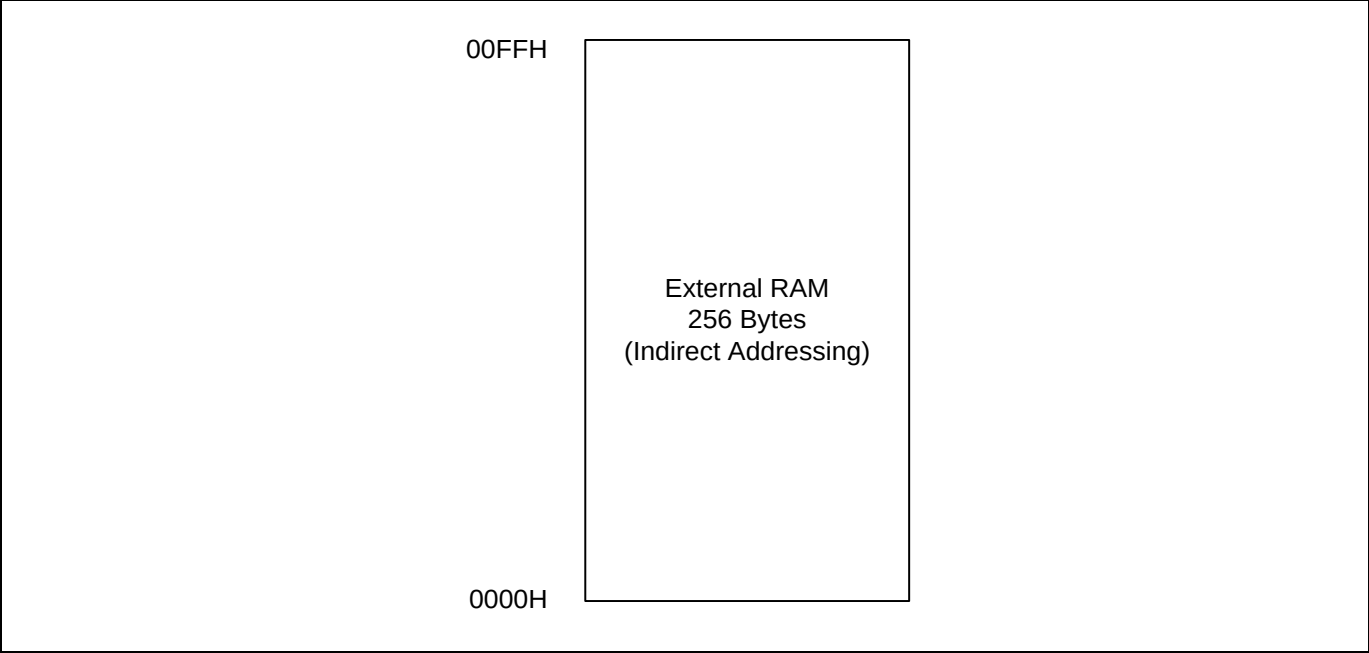


Figure 8.4 XDATA 存储区域

8.4 SFR 图

8.4.1 SFR 摘要

-	Reserved
	M8051 compatible

	00H/8H ⁽¹⁾	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
0F8H	IP1	—	FSADRH	FSADRM	FSADRL	FIDR	FMCR	—
0F0H	B	I2CSAR1	ADWRCR0		ADWRCR2	ADWRCR3	ADWCRL	ADWCRH
0E8H	RSTFR	I2CCR	I2CSR	I2CSAR0	I2CDR	I2CSDHR	I2CSCLR	I2CSCHR
0E0H	ACC	—	UARTCR1	UARTCR2	UARTCR3	UARTST	UARTBD	UARTDR
0D8H	LVRCR	—	—	—	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	
98H	P3	—	—	—	ADCCRL	ADCCRH	ADCDRL	ADCDRH
90H	P2	P0OD	P1OD	P2OD	P3OD	—	WTCR	BUZCR
88H	P1	WTDRL/ WTCNT	SCCR	BITCR	BITCNT	WDCR	WTDRL/ WDCNT	BUZDR
80H	P0	SP	DPL	DPH	DPL1	DPH1	LVICR	PCON

Table 8.1 SFR 摘要

NOTE)

- 00H/8H，这一列的寄存器支持位操作
- 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
更多详细信息请参看章节17.2。

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

8.4.2 SFR

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

Table 8.2 SFR Map

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

Table 8.2 SFR Map (Continued)

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

Table 8.2 SFR Map (Continued)

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

Table 8.2 SFR Map (Concluded)

8.4.3 SFR

ACC (累加器) : E0H

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

Initial value : 00H

ACC Accumulator

B (B 寄存器) : F0H

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

Initial value : 00H

B B Register

SP (Stack Pointer) : 81H

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

Initial value : 07H

SP Stack Pointer

DPL (数据指针寄存器 Low) : 82H

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

Initial value : 00H

DPL Data Pointer Low

DPH (数据指针寄存器 High) : 83H

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

Initial value : 00H

DPH Data Pointer High

DPL1 (数据指针寄存器 Low 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 High 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** 溢出标志位**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 选择寄存器 **(Keep always '0')**.

0 Select MOV C @(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

9 I/O 口

9.1 I/O 口

MC96F8208S 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.

9.2 端口寄存器

9.2.1 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, 数据可以通过Px的相应位读到.

9.2.2 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.

9.2.3 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

9.2.4 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.

9.2.5 消抖使能寄存器 (PxDB)

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

9.2.6 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.

9.2.7 寄存器

Name	Address	Direction	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

Table 9.1 Port Register Map

9.3 P0 口

9.3.1 P0 端口说明

P0 是 4-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。

9.3.2 P0 寄存器说明

P0 (P0 数据寄存器) : 80H

7	6	5	4	3	2	1	0
—	—	—	—	P03	P02	P01	P00
—	—	—	—	RW	RW	RW	RW

Initial value : 00H

P0[3:0] I/O Data

Note: 直接位判断并跳转的指令(见下面例子)不能用于输入口, 更多详细信息请参看章节17.2。

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
—	—	—	—	P03IO	P02IO	P01IO	P00IO
—	—	—	—	RW	RW	RW	RW

Initial value : 00H

P0IO[3:0] P0 数据 I/O 方向。

0 Input
1 Output

NOTE)

1. EINT0 ~ EINT1 功能在输入状态时才有效

P0PU (P0 上拉电阻选择寄存器) : ACH

7	6	5	4	3	2	1	0
—	—	—	—	P03PU	P02PU	P01PU	P00PU
—	—	—	—	RW	RW	RW	RW

Initial value : 00H

P0PU[3:0] 配置P0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
—	—	—	—	P03OD	P02OD	P01OD	P00OD
—	—	—	—	RW	RW	RW	RW

Initial value : 00H

P0OD[3:0] 配置 P0 口漏极开路

0 推-拉输出
1 漏极开路输出

P03DB (P0/P3 消抖使能寄存器) : DEH

7	6	5	4	3	2	1	0
DBCLK1	DBCLK0	P35DB	—	—	—	P03DB	P02DB
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

P03DB 配置 P03 口消抖

0 Disable

1 Enable

P02DB 配置 P02 口消抖

0 Disable

1 Enable

NOTE)

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。

P0FSR (Port 0 功能选择寄存器) : D3H

7	6	5	4	3	2	1	0
—	—	—	P0FSR4	P0FSR3	P0FSR2	P0FSR1	P0FSR0
—	—	—	RW	RW	RW	RW	RW

Initial value : 00H

P0FSR4 P03 功能选择

0 I/O □ (EINT1 功能仅在输入状态时使用)

1 AN3

P0FSR[3:2] P02 功能选择

P0FSR3 P0FSR2 Description

0 0 I/O Port (EINT0 功能仅在输入状态时使用)

0 1 AVREF

1 0 AN2

1 1 Not used

P0FSR1 P01 功能使用

0 I/O □

1 AN1

P0FSR0 P00 功能选择

0 I/O □

1 AN0

9.4 P1 口

9.4.1 P1 端口说明

P1 是 6-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)。具体请参考端口功能寄存器。

9.4.2 P1寄存器说明

P1 (P1 数据寄存器) : 88H

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

Initial value : 00H

P1[6:1] I/O Data

Note: 直接位判断并跳转的指令(见下面例子)不能用于输入口, 更多详细信息请参看章节17.2。

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
–	P16IO	P15IO	P14IO	P13IO	P12IO	P11IO	–
–	RW	RW	RW	RW	RW	RW	–

Initial value : 00H

P1IO[6:1] P1 Data I/O Direction

0 Input

1 Output

NOTE)

1. ENINT6/EINT11/EINT12/EC1 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
–	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	–
–	RW	RW	RW	RW	RW	RW	–

Initial value : 00H

P1PU[6:1] 配置 P1 口上拉电阻选择寄存器

0 Disable

1 Enable

P1OD (P1 漏极开路选择寄存器) : 92H

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

Initial value : 00H

P1OD[6:1] 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
—	—	—	—	P13DB	P12DB	P11DB	—
—	—	—	—	RW	RW	RW	—

Initial value : 00H

P13DB 配置 P13 口消抖寄存器

0 Disable

1 Enable

P12DB 配置 P12 口消抖寄存器

0 Disable

1 Enable

P11DB 配置 P11 口消抖使能寄存器

0 Disable

1 Enable

NOTE)

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.

P1FSRH (Port 1 功能选择高位寄存器) : D5H

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

Initial value : 00H

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
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
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	–
–	RW	RW	RW	RW	RW	RW	–

Initial value : 00H

P1FSRL[6:5] P13 功能选择

P1FSRL6	P1FSRL5	Description
0	0	I/O 口 (EINT12 功能仅在输入状态下才使用)
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 功能仅在输入状态下才使用)
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 口 (EINT6/EC1 功能仅在输入状态下才使用)
0	1	BUZO
1	0	AN8
1	1	Not used

9.5 P2 口

9.5.1 P2 端口说明

P2 是 2-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). 具体请参考端口功能寄存器.

9.5.2 P2寄存器说明

P2 (P2 数据寄存器) : 90H

7	6	5	4	3	2	1	0
—	—	P25	P24	—	—	—	—
—	—	RW	RW	—	—	—	—

Initial value : 00H

P2[5:4] I/O Data

Note: 直接位判断并跳转的指令(见下面例子)不能用于输入口, 更多详细信息请参看章节17.2。

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
—	—	P25IO	P24IO	—	—	—	—
—	—	RW	RW	—	—	—	—

Initial value : 00H

P2IO[5:4] P2 Data I/O Direction

0 Input

1 Output

P2PU (P2 上拉电阻选择寄存器) : AEH

7	6	5	4	3	2	1	0
—	—	P25PU	P24PU	—	—	—	—
—	—	RW	RW	—	—	—	—

Initial value : 00H

P2PU[5:4] 配置 P2 口上拉电阻

0 Disable

1 Enable

P2OD (P2 漏极开路选择寄存器) : 93H

7	6	5	4	3	2	1	0
—	—	P25OD	P24OD	—	—	—	—
—	—	RW	RW	—	—	—	—

Initial value : 00H

P2OD[5:4] 配置 P2 口漏极开路

0 推-拉 输出

1 漏极开路输出

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 □

1 SCL

P2FSR0 P24 功能选择

0 I/O □

1 SDA

NOTE)

1. 如果 P25 和 P24 用作 SCL/SDA 做 I2C, P2OD[5:4] 应该设置为 “1”.

9.6 P3 口

9.6.1 P3 端口说明

P3 是 6-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)。具体请参考端口功能寄存器。

9.6.2 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	—	—	P32	P31	P30
RW	RW	RW	—	—	RW	RW	RW

Initial value : 00H

P3[7:5/2:0] I/O Data

Note: 直接位判断并跳转的指令(见下面例子)不能用于输入口, 更多详细信息请参看章节17.2。

例子: IF(P00) → IF(P0 & 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	—	—	P32IO	P31IO	P30IO
RW	RW	RW	—	—	RW	RW	RW

Initial value : 00H

P3IO[7:5/2:0] P3 数据 I/O 方向

0 Input
1 Output

NOTE)

1. RXD, EINT10 功能仅在输入状态时有效

P3PU (P3 上拉电阻选择寄存器) : AFH

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

Initial value : 00H

P3PU[7:5/2:0] 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	—	—	P32OD	P31OD	P30OD
RW	RW	RW	—	—	RW	RW	RW

Initial value : 00H

P3OD[7:5/2:0] 配置 P3 口漏极开路

- 0 推-拉输出
1 漏极开路输出

P3FSR (Port 3 功能选择寄存器) : D7H

7	6	5	4	3	2	1	0
P3FSR7	P3FSR6	P3FSR5	—	—	P3FSR2	P3FSR1	P3FSR0
RW	RW	RW	—	—	RW	RW	RW

Initial value : 00H

P3FSR7 P37 功能选择

- 0 I/O 口
1 XOUT 功能

P3FSR6 P36 功能选择

- 0 I/O 口
1 XIN

P3FSR5 P35 功能选择

- 0 I/O 口
1 T0O/PWM0O

P3FSR2 P31 功能选择

- 0 I/O Port (RXD 功能仅在输入时有效)
1 SCL

P3FSR[1:0] P30 功能选择

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

NOTE)

- 配置 P32/RESETB 配置选项
- 如果 P31 和 P30 用作 SCL/SDA 做 I2C, P3OD[1:0] 应该设置为 “1”.

10 中断控制器

10.1 概述

MC96F8208S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 个优先级组
- 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.

MC96F8208S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. 表10-1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.

Interrupt Group	Highest Lowest			
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

Table 10.1 中断组优先级

10.2 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。图10.1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。

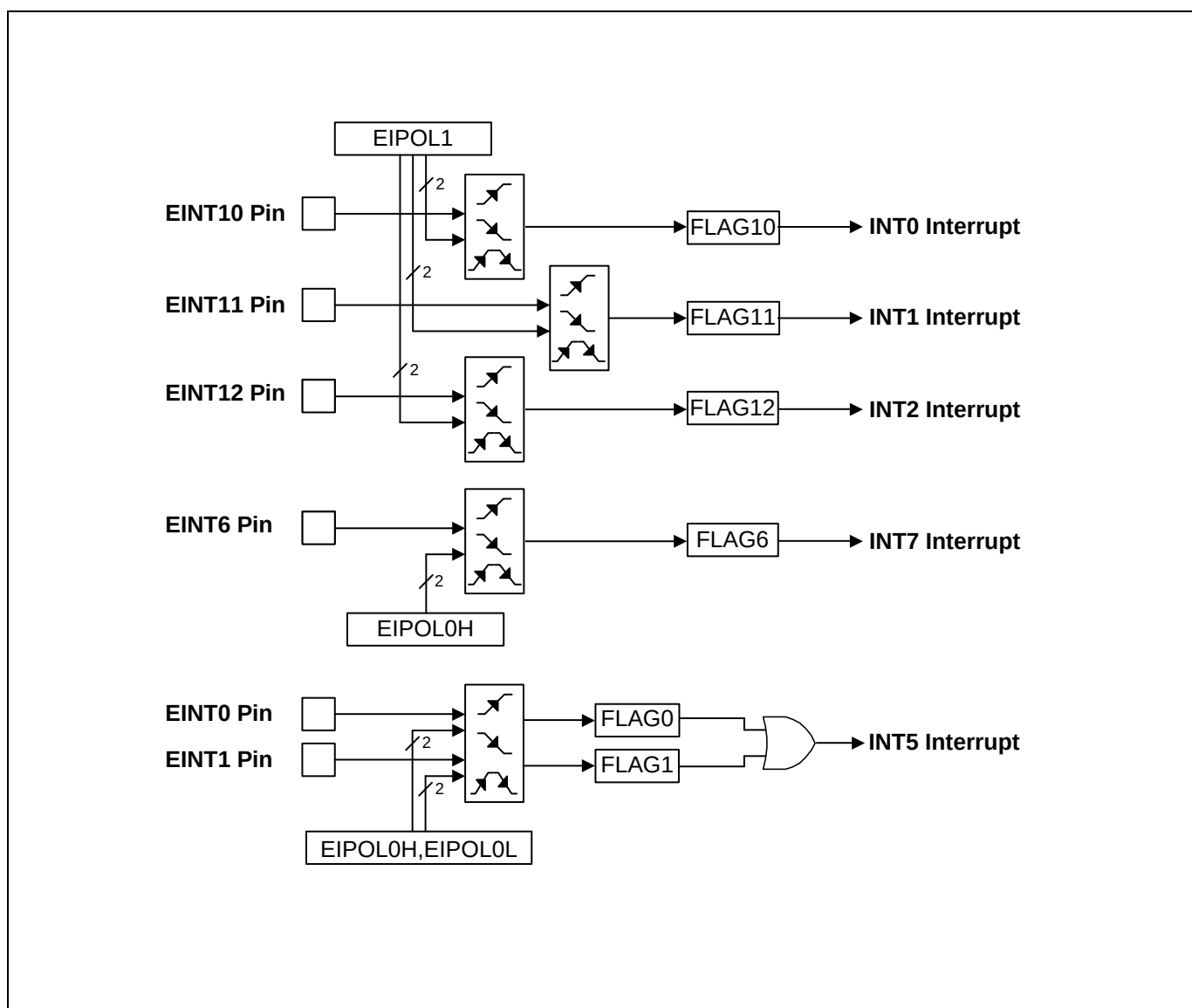


Figure 10.1 外部中断说明

10.3 方框图

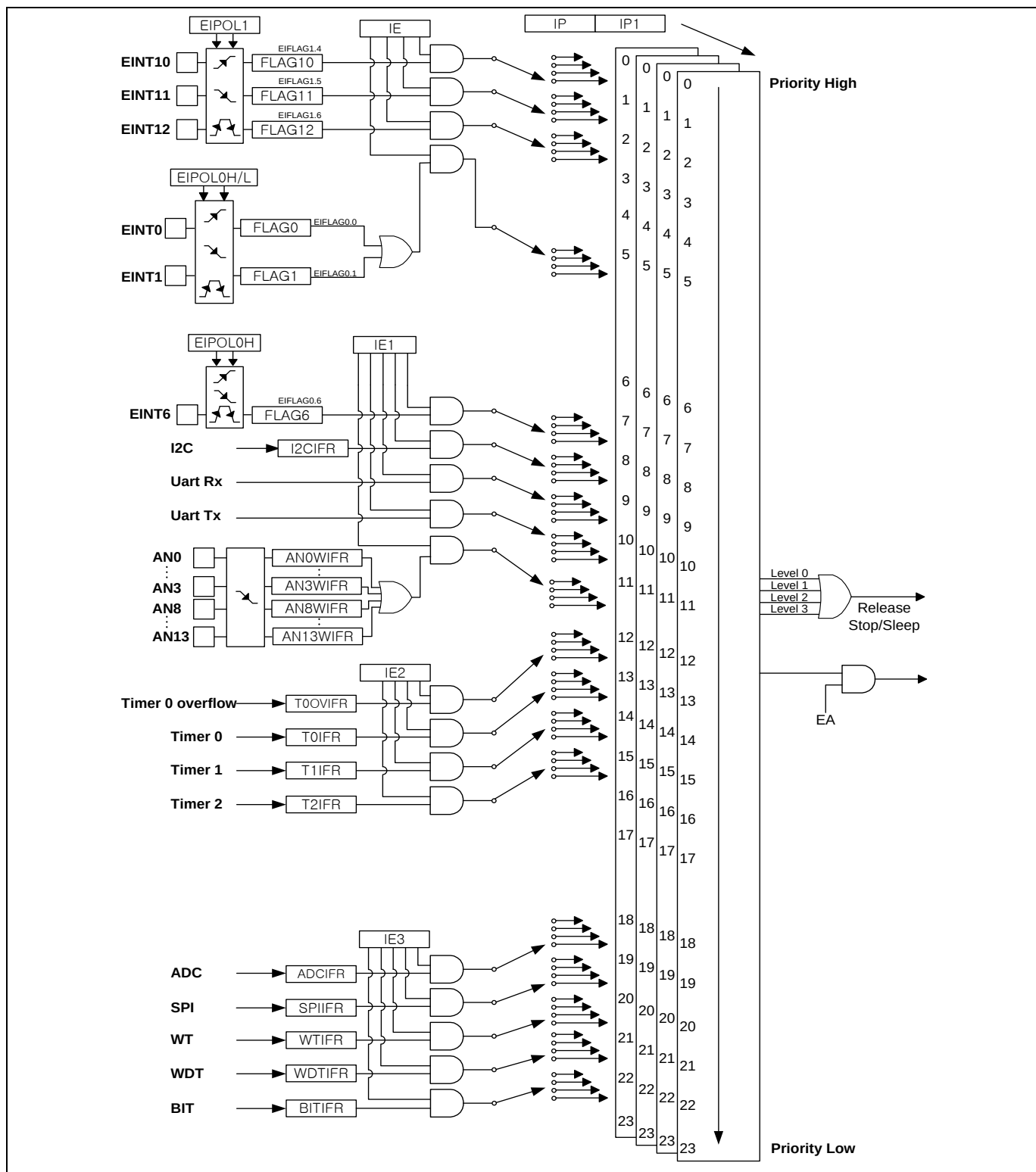


Figure 10.2 Block Diagram of Interrupt

NOTE)

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。

10.4 中断向量表

图表10-2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。

Interrupt Source	Symbol	Interrupt Enable Bit	Polarity	Mask	Vector Address
Hardware Reset	RESETB	-	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 0/1	INT5	IE.5	6	Maskable	002BH
-	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
-	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

Table 10.2 中断向量地址表

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。

10.5 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。

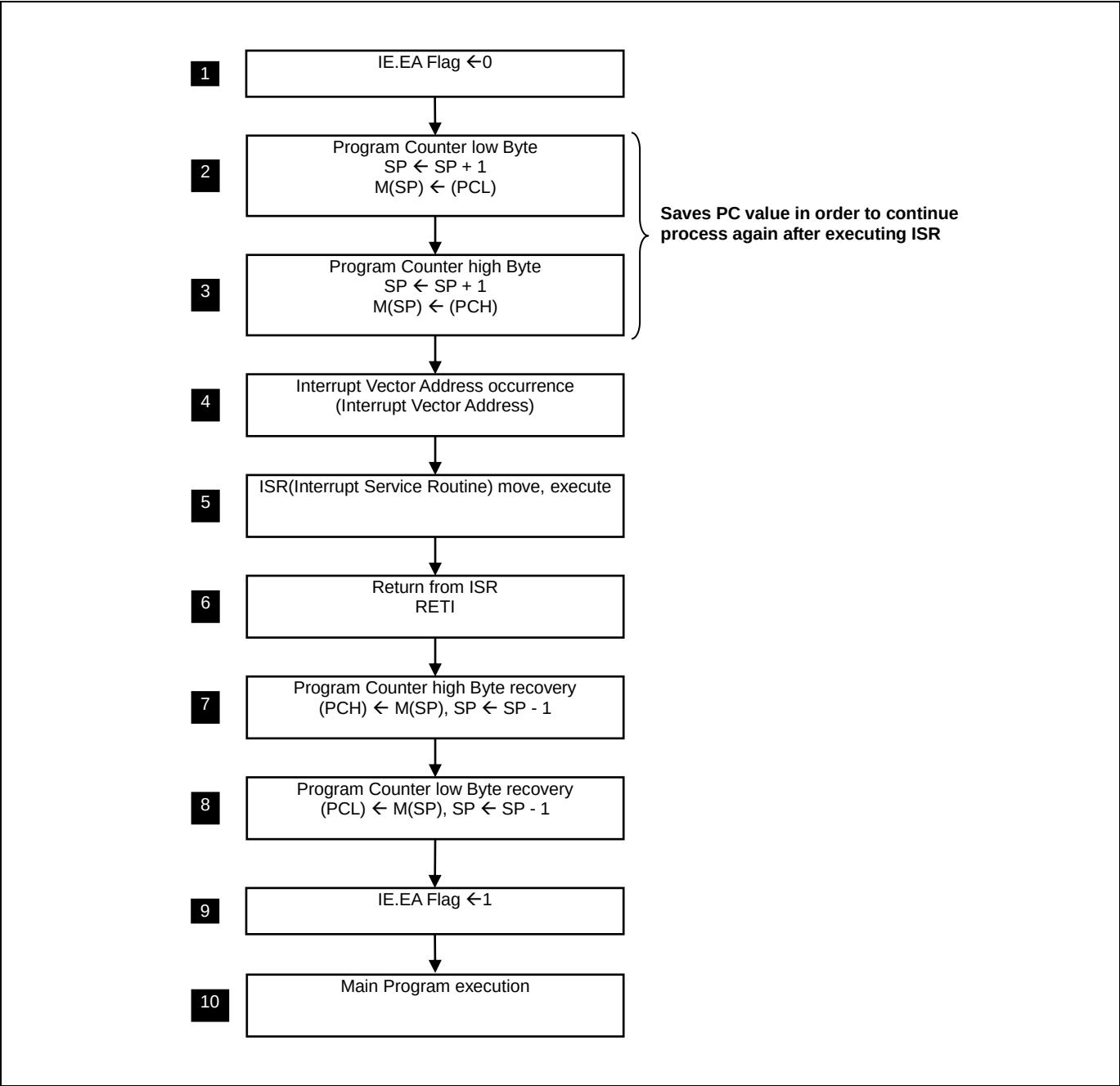


Figure 10.3 Interrupt Vector Address Table

10.6 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)

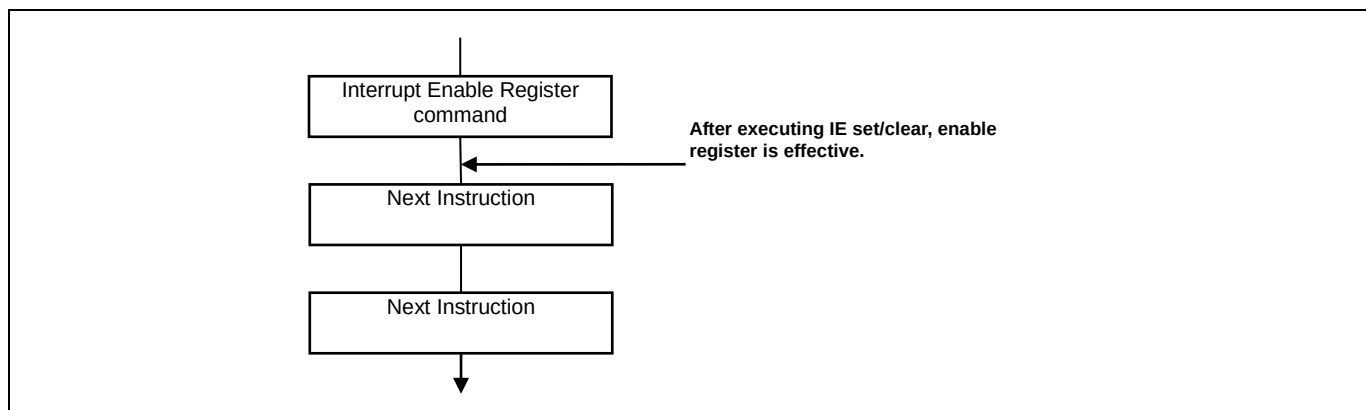


Figure 10.4 中断使能寄存器时序

Case b) 中断标志位寄存器

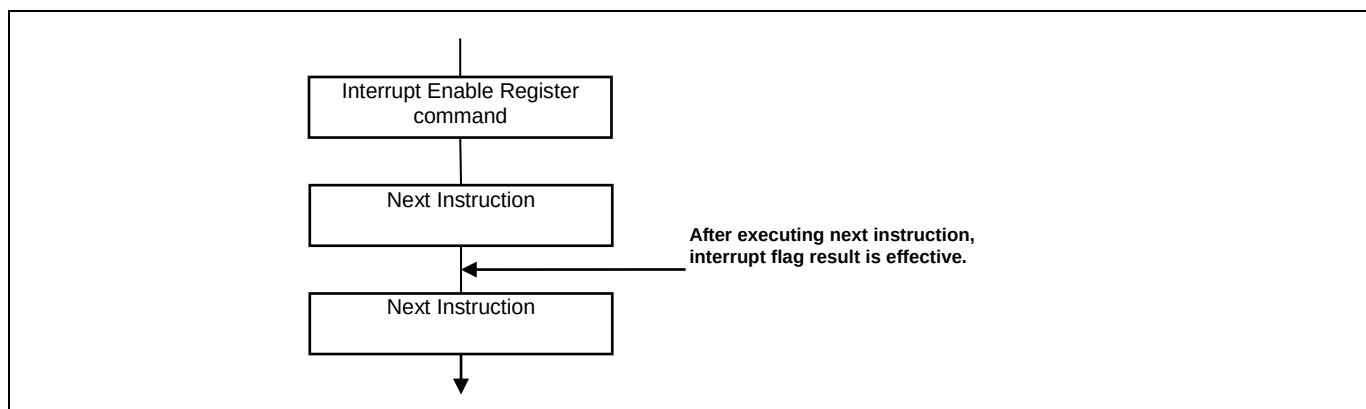


Figure 10.5 中断标志位寄存器时序图

10.7 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。

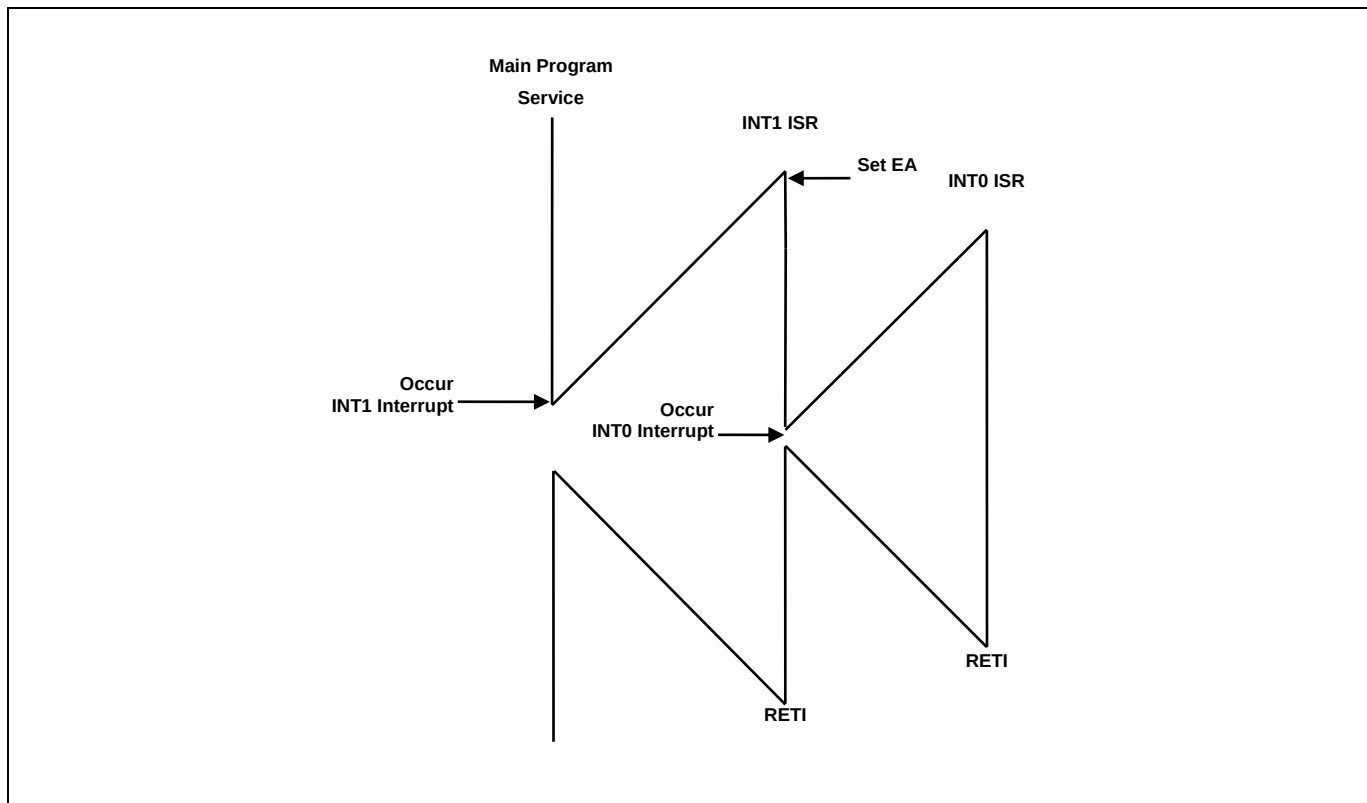


Figure 10.6 中断复用时序

图 10.6 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。

10.8 中断使能响应时序

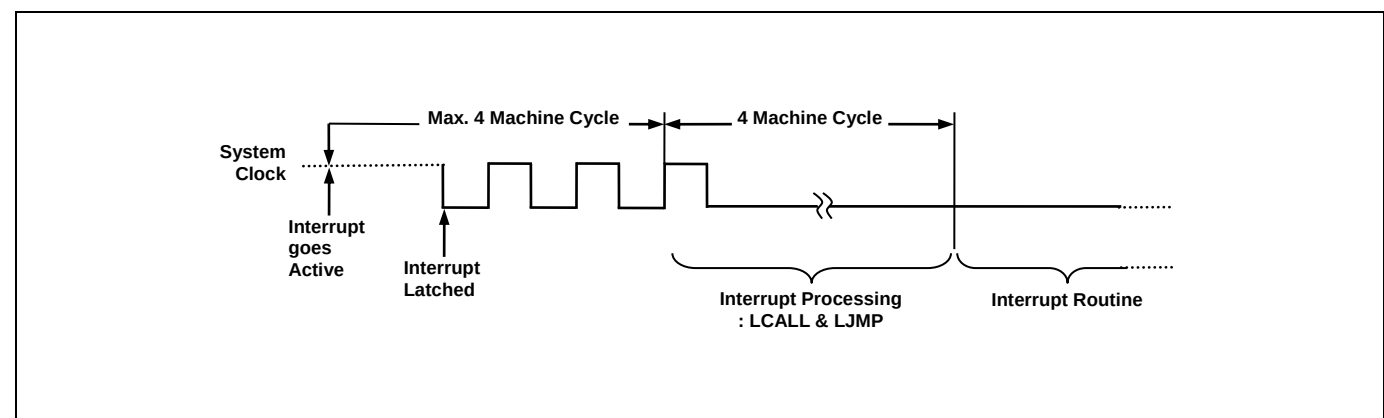


Figure 10.7 中断响应时序方框图

10.9 中断服务程序地址

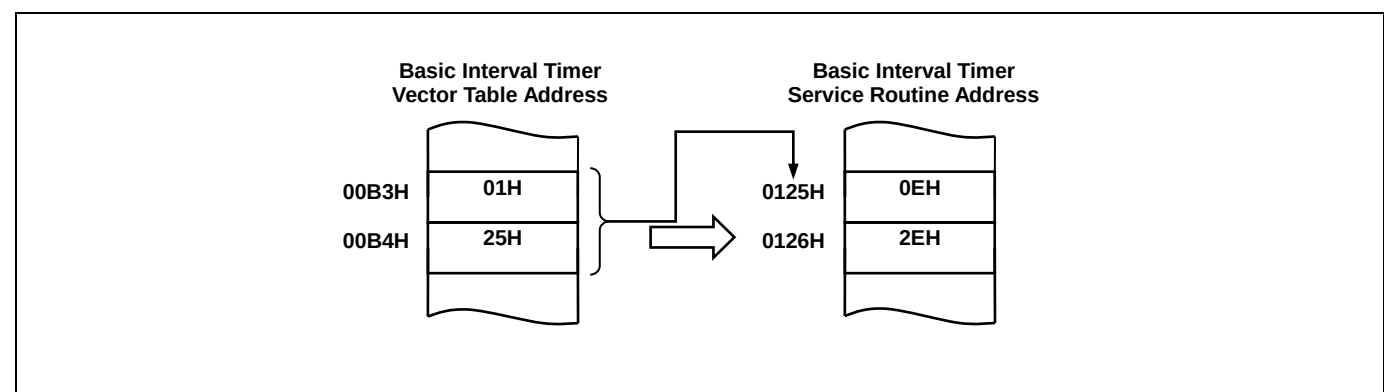


Figure 10.8 向量表地址和对应的 ISP 入口地址

10.10 通用寄存器的设置和复位

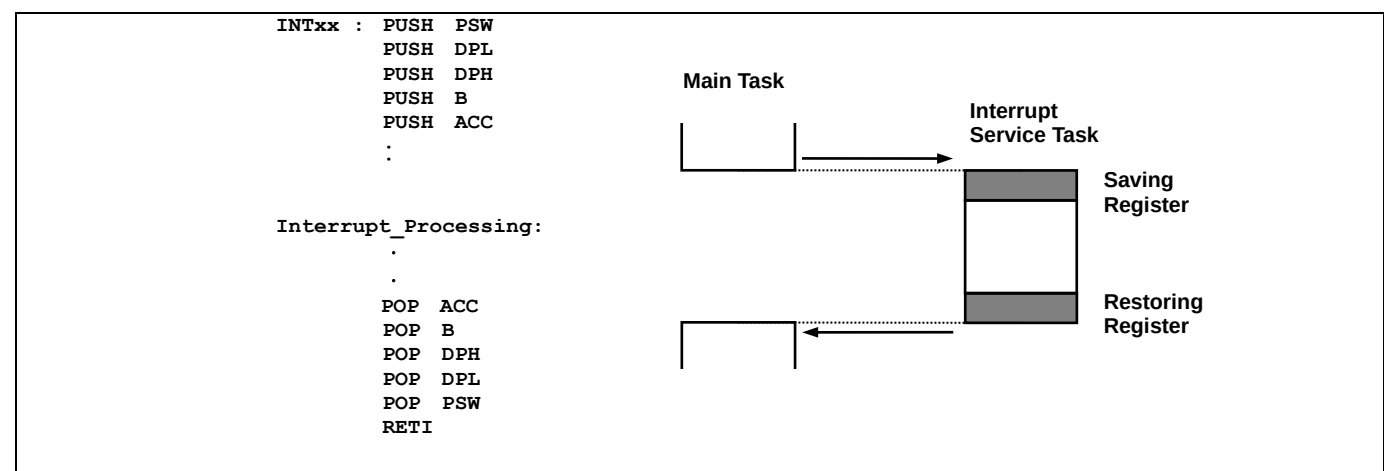


Figure 10.9 设置举例

10.11 中断时序

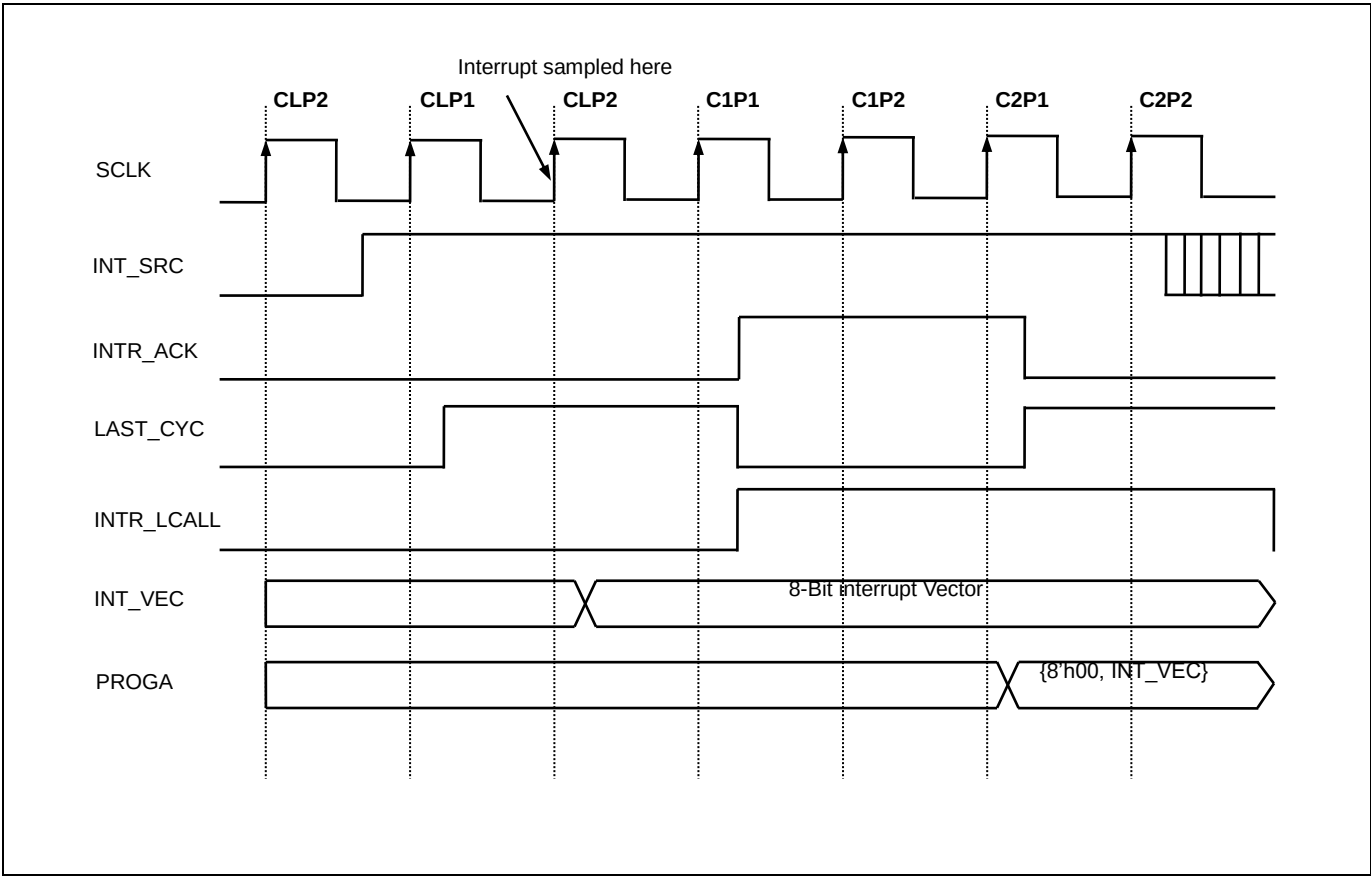


Figure 10.10 中断接受和中断返回指令时序图

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.

NOTE)

- 1. command cycle CLPx: L=Last cycle, 1=1st cycle or 1st phase, 2=2nd cycle or 2nd phase

10.12 中断寄存器概述

10.12.1 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。

10.12.2 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。Level 3 优先级最高，level 0 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。

10.12.3 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。

10.12.4 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。默认值是任何边沿没有中断。

10.12.5 寄存器图

Name	Address	Direction	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

Table 10.3 中断寄存器图

10.12.6 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)组成。

10.12.7 中断寄存器

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	总中断关闭
1	总中断打开
INT5E	使能或禁止外部中断 0/1 (EINT0/ EINT1)
0	Disable
1	Enable
INT2E	使能或禁止外部中断 12(EINT12)
0	Disable
1	Enable
INT1E	使能或禁止外部中断 11(EINT11)
0	Disable
1	Enable
INT0E	使能或禁止外部中断 10 (EINT10)
0	Disable
1	Enable

IE1 (中断使能寄存器1): A9H

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

Initial value: 00H

- INT11E

使能或禁止 ADC Wake-up 中断

0 Disable

1 Enable
- INT10E

使能或禁止 UART Tx 中断

0 Disable

1 Enable
- INT9E

使能或禁止 UART Rx 中断

0 Disable

1 Enable
- INT8E

使能或禁止 I2C 中断

0 Disable

1 Enable
- INT7E

使能或禁止外部中断6 (EINT6)

0 Disable

1 Enable

IE2 (中断使能寄存器 2) : AAH

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

Initial value : 00H

INT15E 使能或禁止Timer 2 匹配中断

0 Disable

1 Enable

INT14E 使能或禁止Timer 1 匹配中断

0 Disable

1 Enable

INT13E 使能或禁止Timer 0 匹配中断

0 Disable

1 Enable

INT12E 使能或禁止Timer 0 溢出中断

0 Disable

1 Enable

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 使能或禁止BIT 中断

0 Disable

1 Enable

INT21E 使能或禁止WDT 中断

0 Disable

1 Enable

INT20E 使能或禁止WT 中断

0 Disable

1 Enable

INT19E 使能或禁止SPI 中断

0 Disable

1 Enable

INT18E 使能或禁止ADC 中断

0 Disable

1 Enable

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	Description
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	T0OVIFR	T0IFR
–	–	–	–	–	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. 写入 “1” 无效.

0	没有I2C 中断产生
1	I2C 中断产生

T0OVIFR

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. 写入 “1” 无效.

0	没有T0 溢出中断产生
1	T0 溢出中断产生

T0IFR

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. 写入 “1” 无效.

0	没有T0 中断产生
1	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	—	—	—	—	FLAG1	FLAG0
—	RW	—	—	—	—	RW	RW

Initial value : 00H

EIFLAG0[6] 当外部中断 6 发生时, 该位置 ‘1’. 写入 ‘0’ 清零或者通过INT_ACK信号自动清零. 写入 “1”无效.

0 没有外部中断 6 发生

1 外部中断 6 发生

EIFLAG0[1:0] 当外部中断 0/1 发生时, 该标志位必须手动写入0清除. 所以, 需要软件清零. 写入 “1”无效

0 没有外部中断 0/1 发生

1 外部中断 0/1 发生

Note: 直接位判断并跳转的指令(见下面例子)不能用于EIFLAG0, 更多详细信息请参看章节17.2。

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		—	—	—	—
—	—	RW	RW	—	—	—	—

Initial value: 00H

EIPOL0H[5:4] 外部中断(EINT6) 边沿触发选择

POL6[1:0] Description

0 0 任何边沿不发生中断

0 1 上升沿触发

1 0 下降沿触发

1 1 双边沿触发

EIPOL0L (外部中断边沿 0 低位寄存器): A4H

7	6	5	4	3	2	1	0
—	—	—	—	POL1		POL0	
—	—	—	—	RW	RW	RW	RW

Initial value: 00H

EIPOL0L[3:0] 外部中断 (EINT1, EINT0) 边沿触发选择

POLn[1:0] Description

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 and 1

EIFLAG1 (外部中断标志位 1 寄存器) : B0H

7	6	5	4	3	2	1	0
—	FLAG12	FLAG11	FLAG10	—	—	—	—
—	RW	RW	RW	—	—	—	—

Initial value : 00H

- EIFLAG1[6:4] 当外部中断 10 ~ 12发生时, 该位置 ‘1’. 写入 ‘0’ 清零或者通过INT_ACK信号自动清零. 写入 “1”无效..
- 0

没有外部中断 10 ~ 12发生
- 1

有外部中断 10 ~ 12 发生

Note: 直接位判断并跳转的指令(见下面例子)不能用于EIFLAG0, 更多详细信息请参看章节17.2。

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

EIPOL1 (外部中断极性 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) 触发极性选择
- POLn[1:0]

Description
- 00

任何边沿触发不产生中断
- 01

上升沿触发
- 10

下降沿触发
- 11

双边沿触发
- Where n =10, 11 and 12

11 外围硬件

11.1 时钟发生器

11.1.1 概述

图 11.1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1MHz INT-RC默认16分频。为了内部时钟的稳定, POR 使用 1 MHz INT-RC。

- 校验内部 RC 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)
- 内部 WDTRC 振荡器 (5 kHz)

11.1.2 方框图

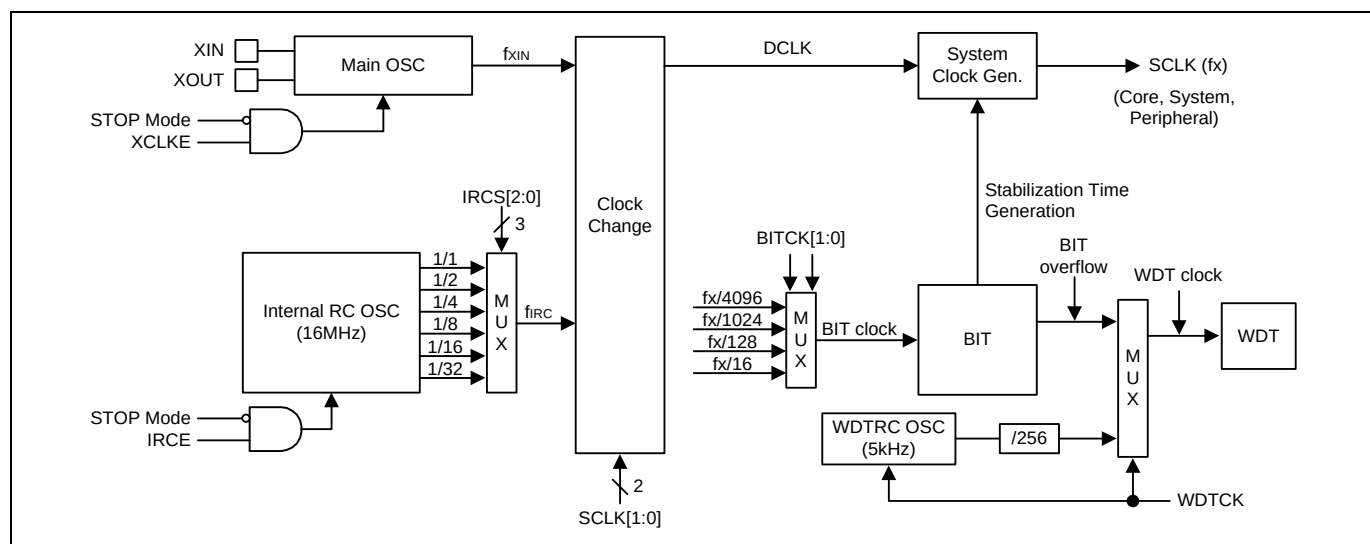


Figure 11.1 时钟发生器方框图

11.1.3 寄存器图

Name	Address	Direction	Default	Description
SCCR	8AH	R/W	00H	System and Clock Control Register
OSCCR	C8H	R/W	08H	Oscillator Control Register

Table 11.1 时钟发生器寄存器

11.1.4 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。

11.1.5 时钟发生器寄存器

SCCR (系统时钟控制寄存器) : 8AH

7	6	5	4	3	2	1	0
—	—	—	—	—	—	SCLK1	SCLK0
—	—	—	—	—	—	R/W	R/W

Initial value : 00H

SCLK [1:0]		系统时钟选择位	
SCLK1	SCLK0	Description	
0	0	INT RC OSC (f _{IRC}) 作为系统时钟	
0	1	外部主时钟作为系统时钟 (f _{XIN})	
1	0	Not available	
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	—
—	—	R/W	R/W	R/W	R/W	R/W	—

Initial value : 08H

IRCS[2:0]		内部 RC 振荡器分频选择	
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	控制内部 RC 振荡器操作	
	0	使能内部 RC OSC
	1	禁止内部 RC OSC

XCLKE	控制外部主时钟	
	0	关闭外部时钟 X-TAL
	1	使能外部时钟 X-TAL

11.2 BIT

11.2.1 概述

MC96F8208S 有一个独立运行不能停止的 8 位基本间隔定时器. 如图11.2方框图所示. 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).

MC96F8208S 基本间隔定时器(BIT) 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

11.2.2 方框图

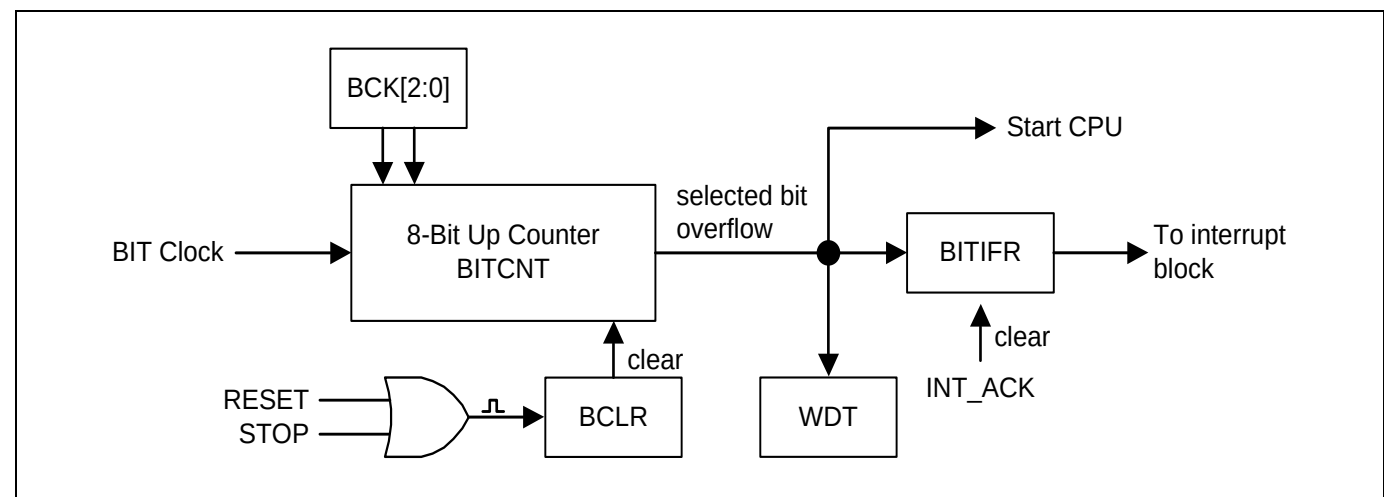


Figure 11.2 BIT方框图

11.2.3 寄存器图

Name	Address	Direction	Default	Description
BITCNT	8CH	R	00H	Basic Interval Timer Counter Register
BITCR	8BH	R/W	01H	Basic Interval Timer Control Register

Table 11.2 BIT寄存器图

11.2.4 BIT寄存器说明

基本间隔定时器寄存器由 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.

11.2.5 BIT寄存器

BITCNT (BIT计数寄存器) : 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 Counter

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. 写入 “1” 无效.

- 0 BIT 中断没有发生
- 1 BIT 中断发生

BITCK[1:0] 选择 BIT 时钟源

BITCK1	BITCK0	Description
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 ‘0’

- 0 Free Running
- 1 Clear Counter

BCK[2:0] 选择 BIT溢出周期

BCK2	BCK1	BCK0	Description
0	0	0	Bit 0 overflow (BIT Clock * 2)
0	0	1	Bit 1 overflow (BIT Clock * 4) (default)
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)

11.3 WDT

11.3.1 概述

看门狗可以迅速的发现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WDTCR[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 WDTCR[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

$$\text{WDT Interrupt Interval} = (\text{BIT Interrupt Interval}) \times (\text{WDTDR Value} + 1)$$

11.3.2 WDT 中断时序波形

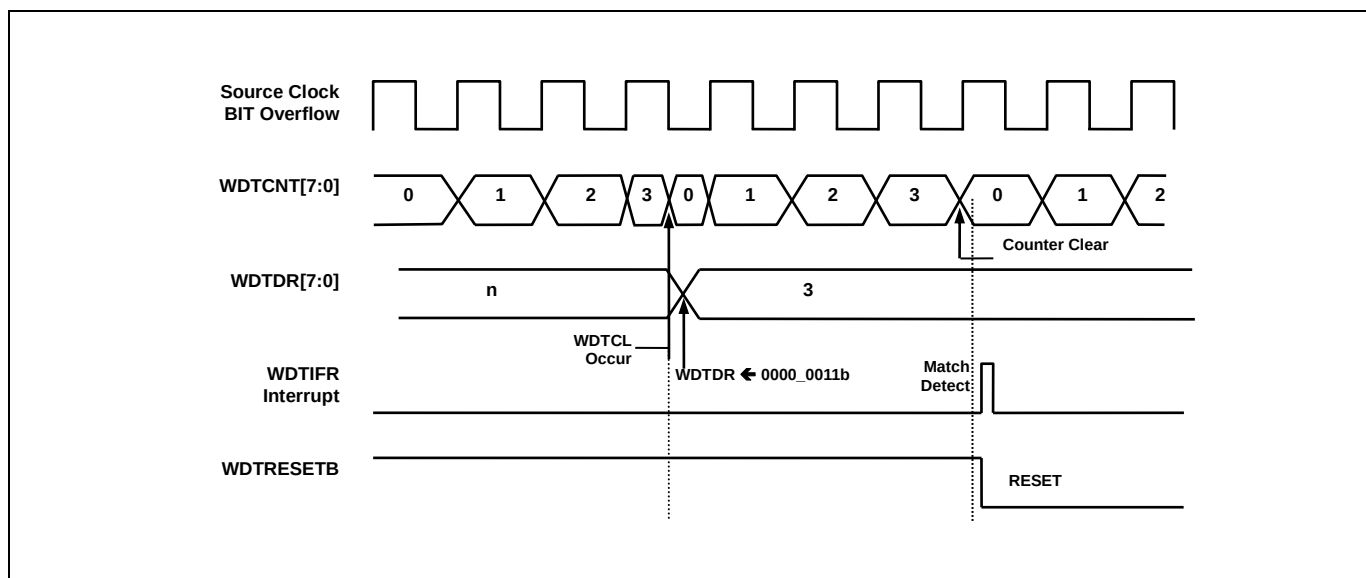


Figure 11.3 看门狗定时器中断时序波形

11.3.3 方框图

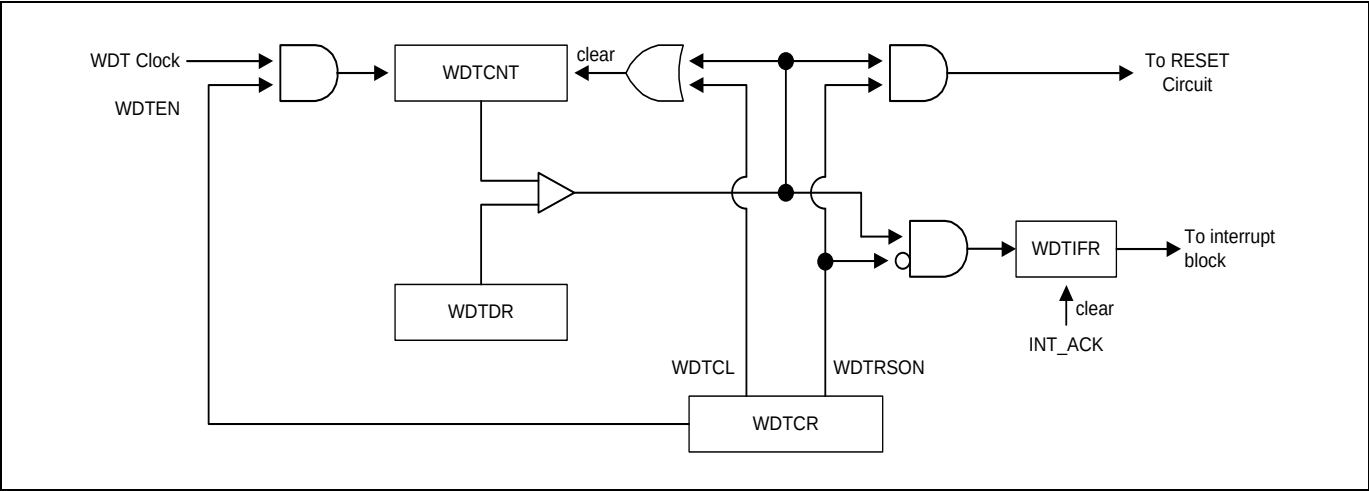


Figure 11.4 WDT框图

11.3.4 寄存器

Name	Address	Direction	Default	Description
WDCNT	8EH	R	00H	Watch Dog Timer Counter Register
WDTDR	8EH	W	FFH	Watch Dog Timer Data Register
WDTCR	8DH	R/W	00H	Watch Dog Timer Control Register

Table 11.3 WDT寄存器图

11.3.5 WDT寄存器说明

看门狗定时器由WDCNT，WDTDR和WDTCR组成

11.3.6 WDT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 中断间隔 =(BIT 中断间隔) x(WDTDR 值+1)

NOTE)

1. 不要写入 “0” 在 WDTDR 寄存器.

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 控制 WDT 操作

- 0 Disable
- 1 Enable

WDTRSON 控制 WDT RESET 操作

- 0 Free Running 8-bit timer
- 1 Watch Dog Timer RESET ON

WDTCL 清除 WDT 计数

- 0 Free Run
- 1 清除 WDT 计数 (一个周期后自动清除)

WDTCK 控制 WDT 时钟选择

- 0 BIT 溢出为 WDT 提供时钟 (WDTRC 禁止)
- 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. 写入 “1”无效..

- 0 没有WDT 中断发生
- 1 WDT 中断发生

11.4 WT

11.4.1 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。

11.4.2 框图

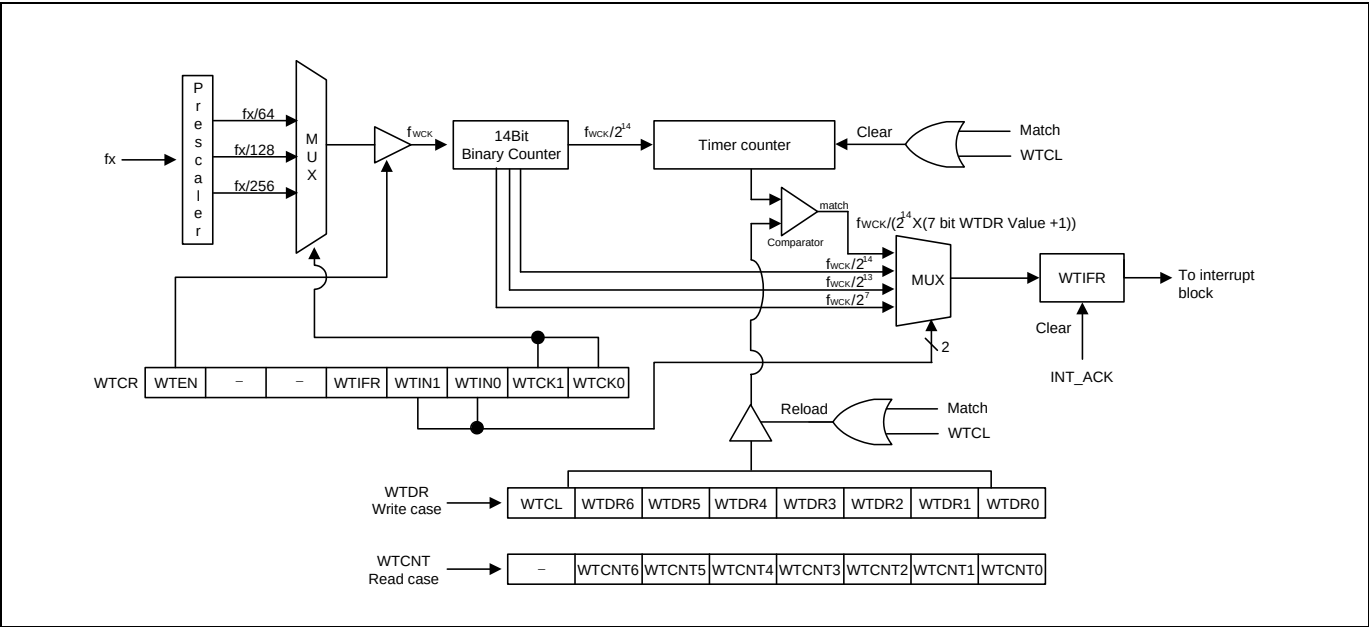


Figure 11.5 Watch Timer框图

11.4.3 寄存器图

Name	Address	Direction	Default	Description
WTCNT	89H	R	00H	Watch Timer Counter Register
WTDR	89H	W	7FH	Watch Timer Data Register
WTCR	96H	R/W	00H	Watch Timer Control Register

Table 11.4 Watch Timer 寄存器

11.4.4 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).

11.4.5 WT寄存器

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 Free Run

1 清除 WT 计数 (auto clear after 1 Cycle)
- WTDR[6:0]

设置 WT 周期

WT 中断间隔 = $\text{fwck}/(2^{14} \times (7\text{bit WTDR 值}+1))$

NOTE)

1. 不要写 “0” 在 WTDR 寄存器.

WTCR (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	控制 Watch Timer		
	0	Disable	
	1	Enable	
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. 写入 “1” 无效.		
	0	WT 中断不产生	
	1	WT 中断产生	
WTIN[1:0]	决定中断间隔		
	WTIN1	WTIN0	Description
	0	0	$f_{wck}/2^7$
	0	1	$f_{wck}/2^{13}$
	1	0	$f_{wck}/2^{14}$
	1	1	$f_{wck}/(2^{14} \times (7bit\ WTDR\ Value+1))$
	WTCK[1:0]	决定时钟源	
WTCK1		WTCK0	Description
0		0	Not available
0		1	$f_x/256$
1		0	$f_x/128$
1		1	$f_x/64$

- NOTE)
1.

f_x – 系统时钟 (Where $f_x= 4.19MHz$)
2.

f_{SUB} – 副时钟频率 (32.768kHz)
3.

f_{WCK} – 选择 Watch timer 时钟

11.5 定时器 0

11.5.1 概述

8-bit 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(TOCNT, 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.

- TIMER 0 时钟源: $f_x/2$, 4, 8, 32, 128, 512, 2048

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PWM0O口输出PWM波形.

T0EN	T0MS[1:0]	T0CK[2:0]	Timer 0
1	00	XXX	8 Bit Timer/Counter Mode
1	01	XXX	8 Bit PWM Mode
1	1X	XXX	8 Bit Capture Mode

Table 11.5 Timer 0 工作模式

11.5.2 8-bit 定时器/计数器 模式

图11.6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。 Timer 0 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)。

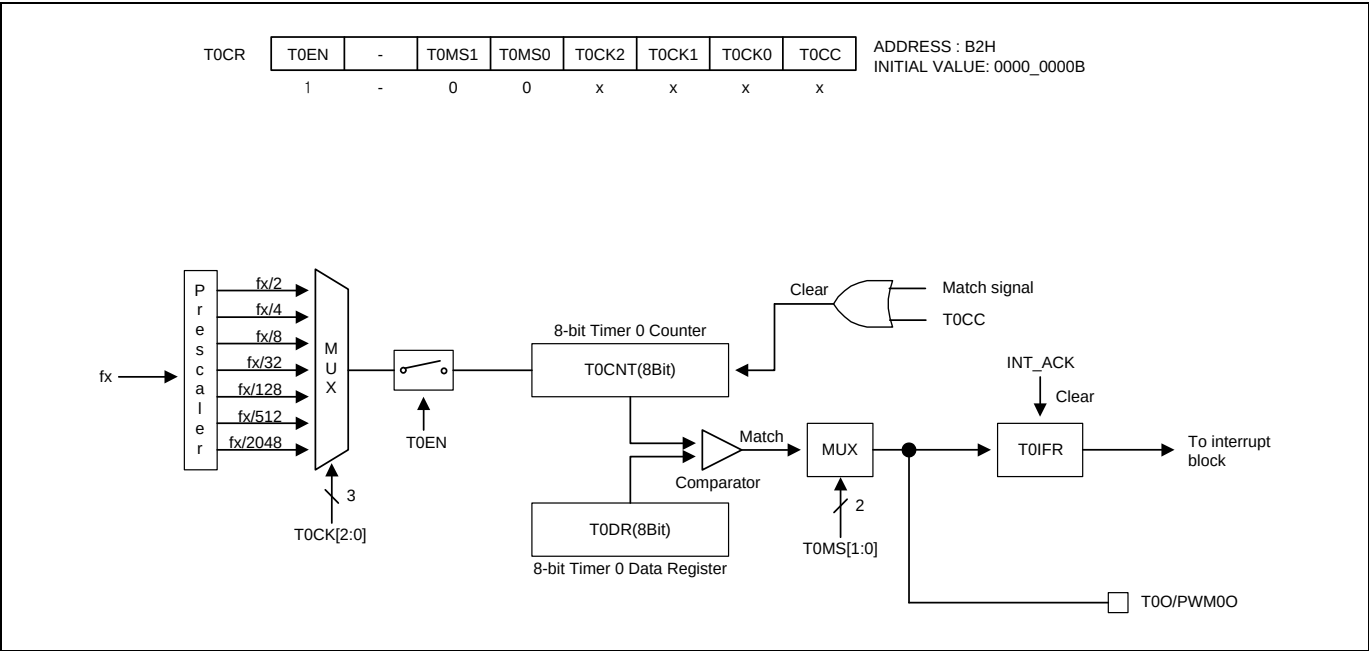


Figure 11.6 定时器0 8-bit 定时器/计数器 模式

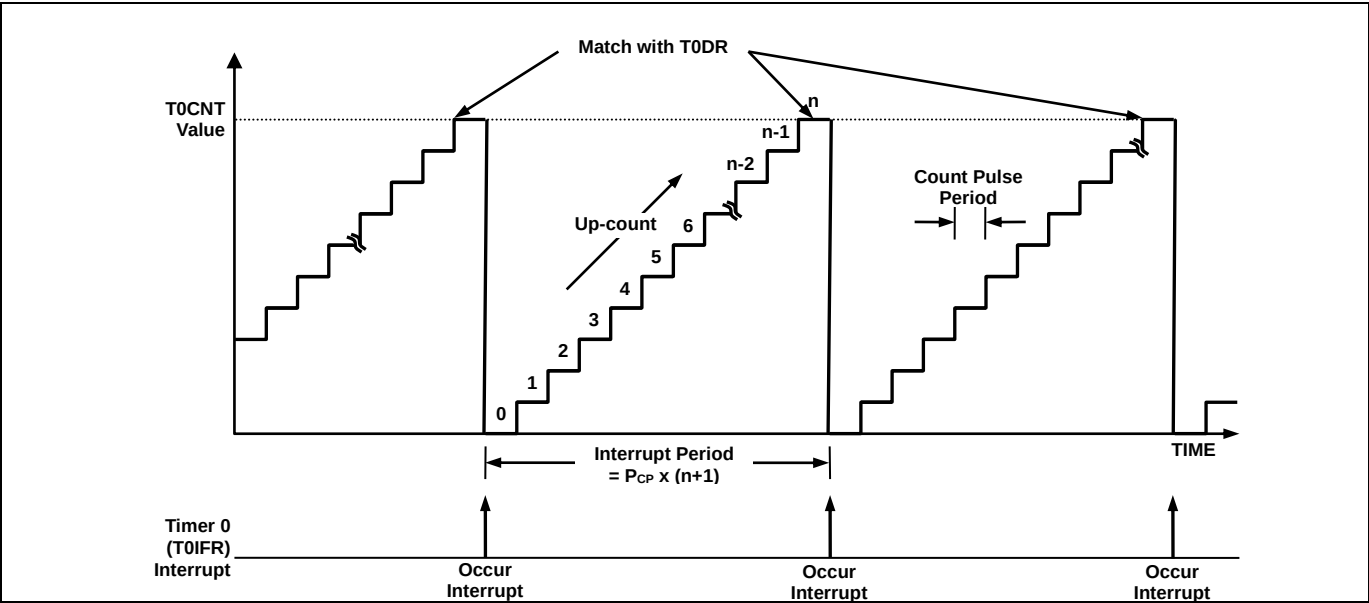


Figure 11.7 8-bit 定时器/计数器 举例

11.5.3 8-bit PWM 模式

定时器0具有高速PWM功能。在PWM模式中，T0O/PWM0O 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。

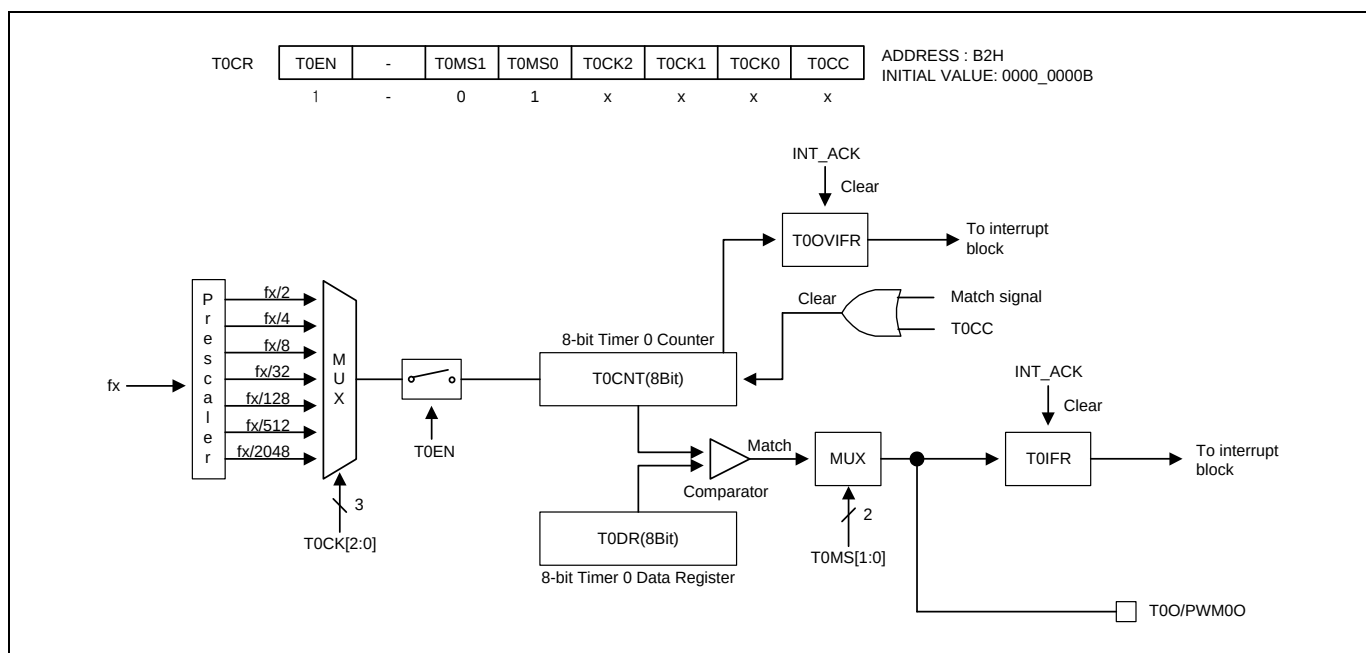


Figure 11.8 定时器0 8-bit PWM 模式

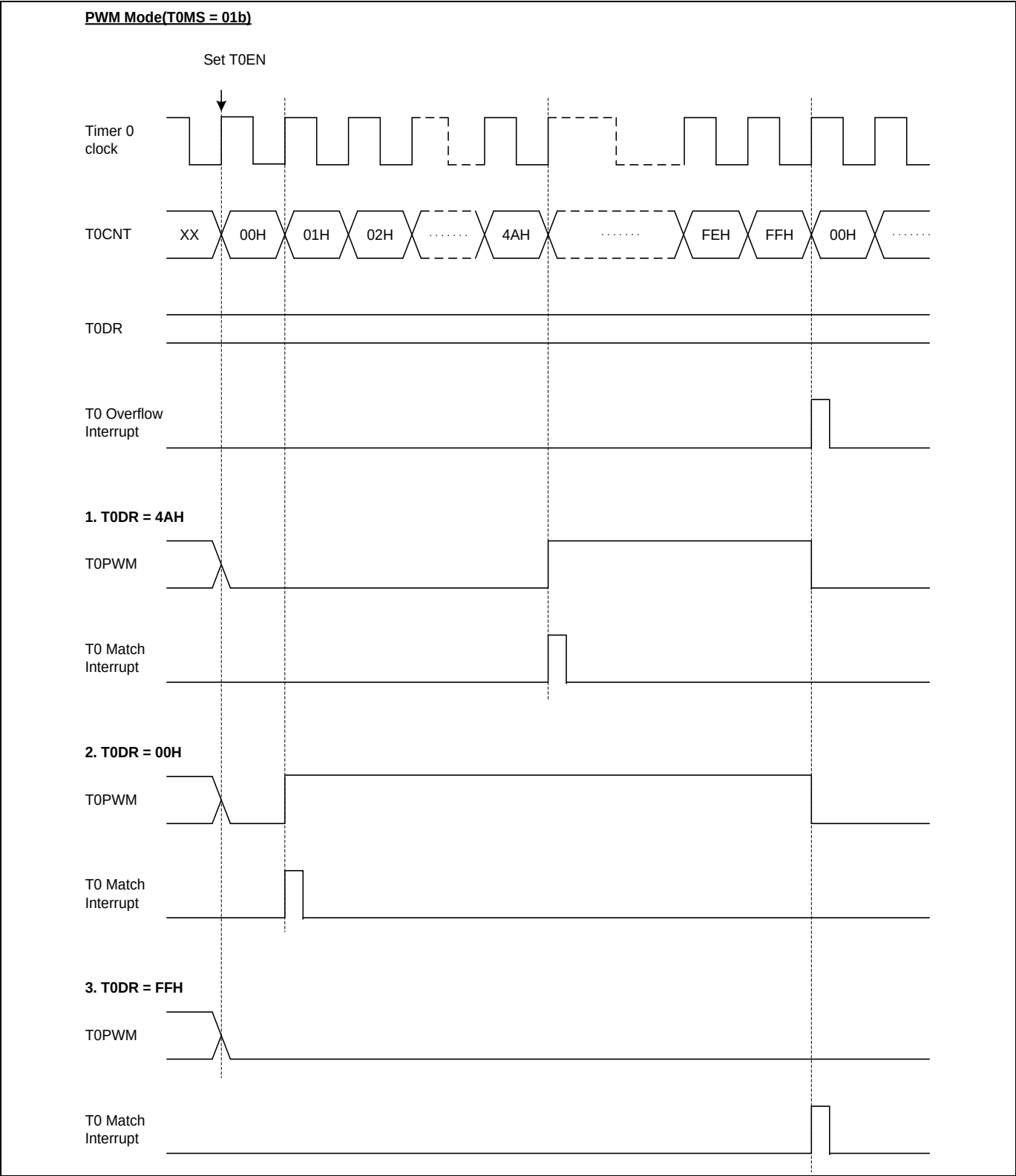


Figure 11.9 定时器0在PWM模式下的 PWM 波形输出

11.5.4 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 $T0DR$.

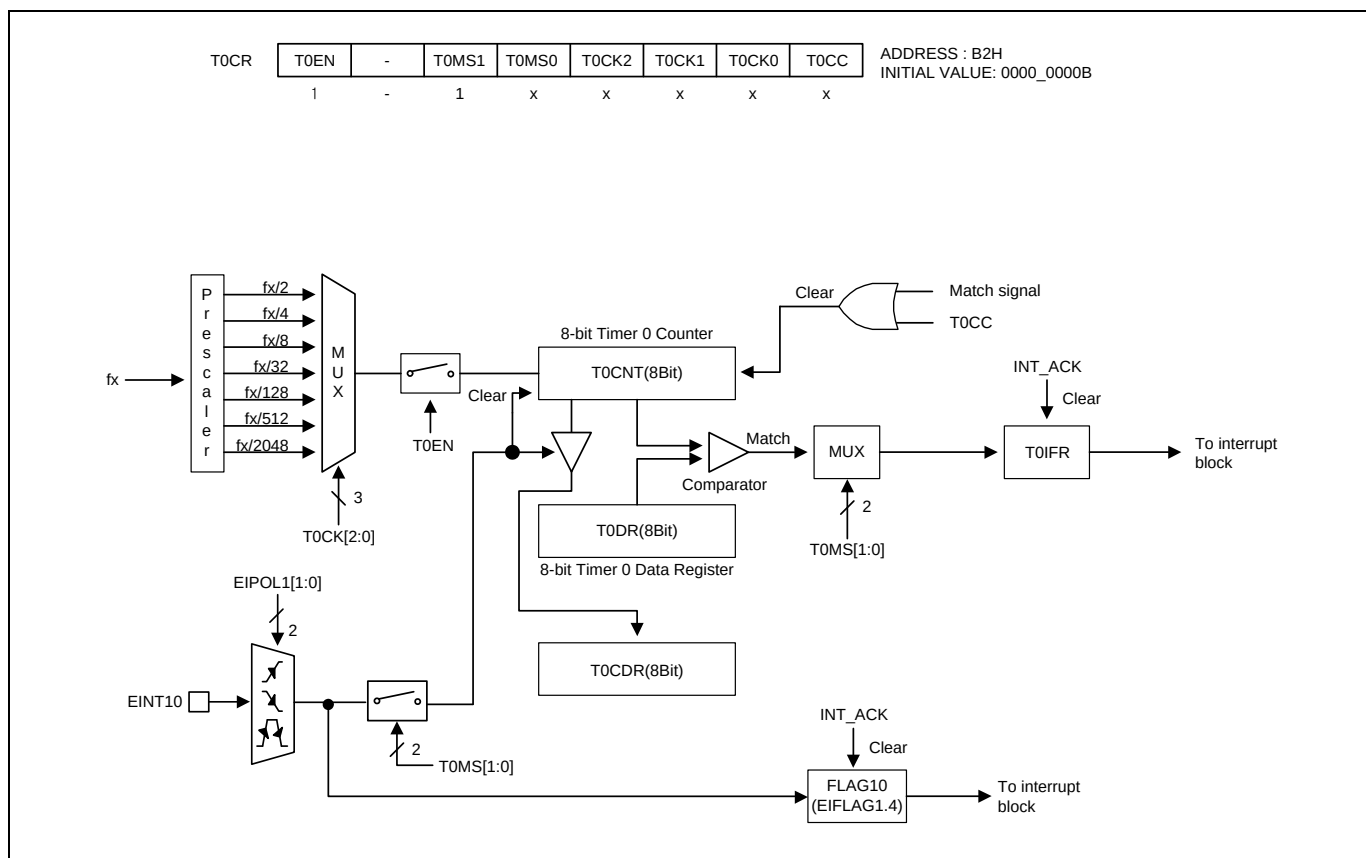


Figure 11.10 定时器0 8-bit 捕捉模式

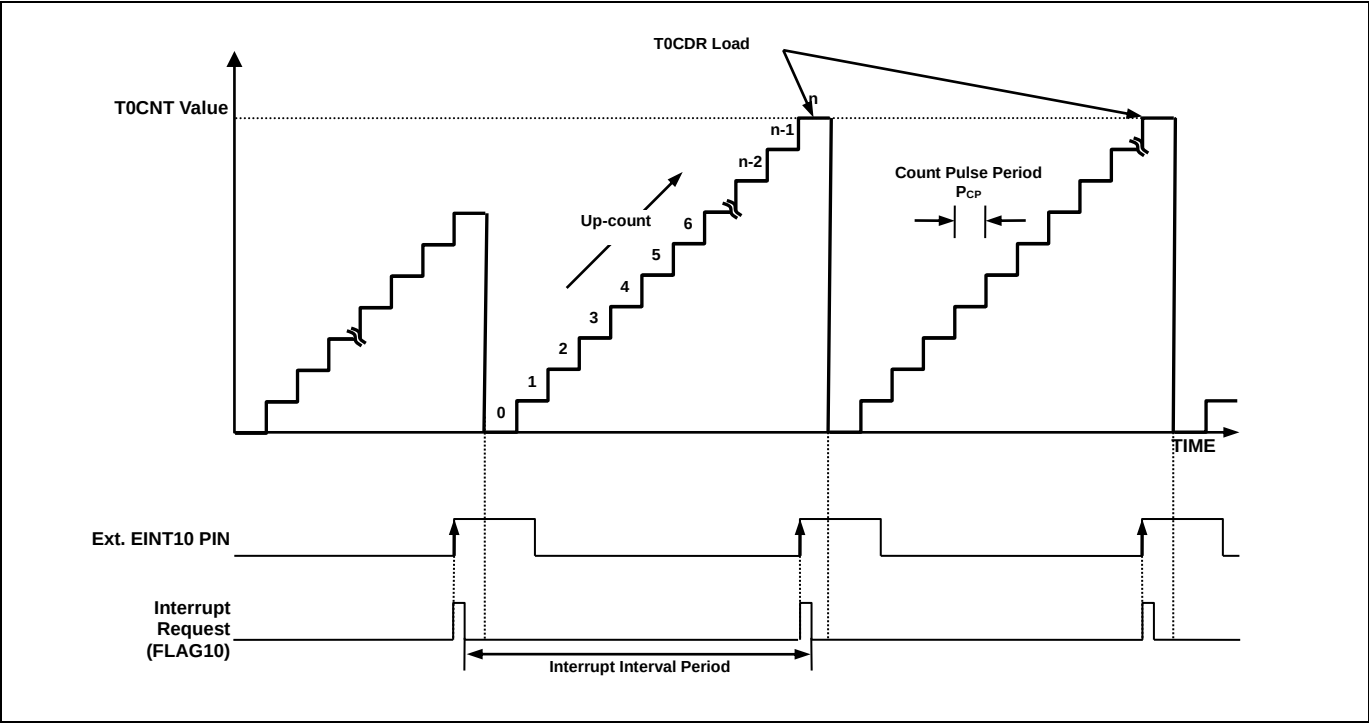


Figure 11.11 定时器0 输入捕捉模式

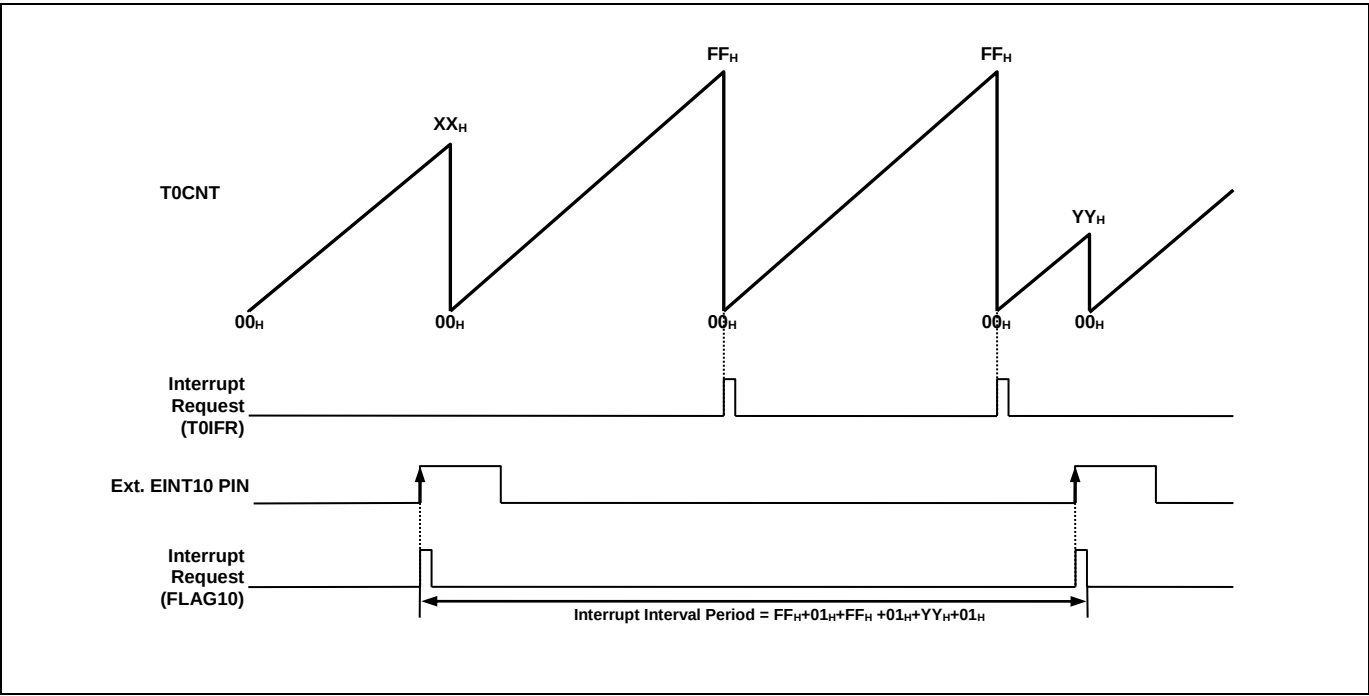


Figure 11.12 捕捉模式定时器溢出

11.5.5 方框图

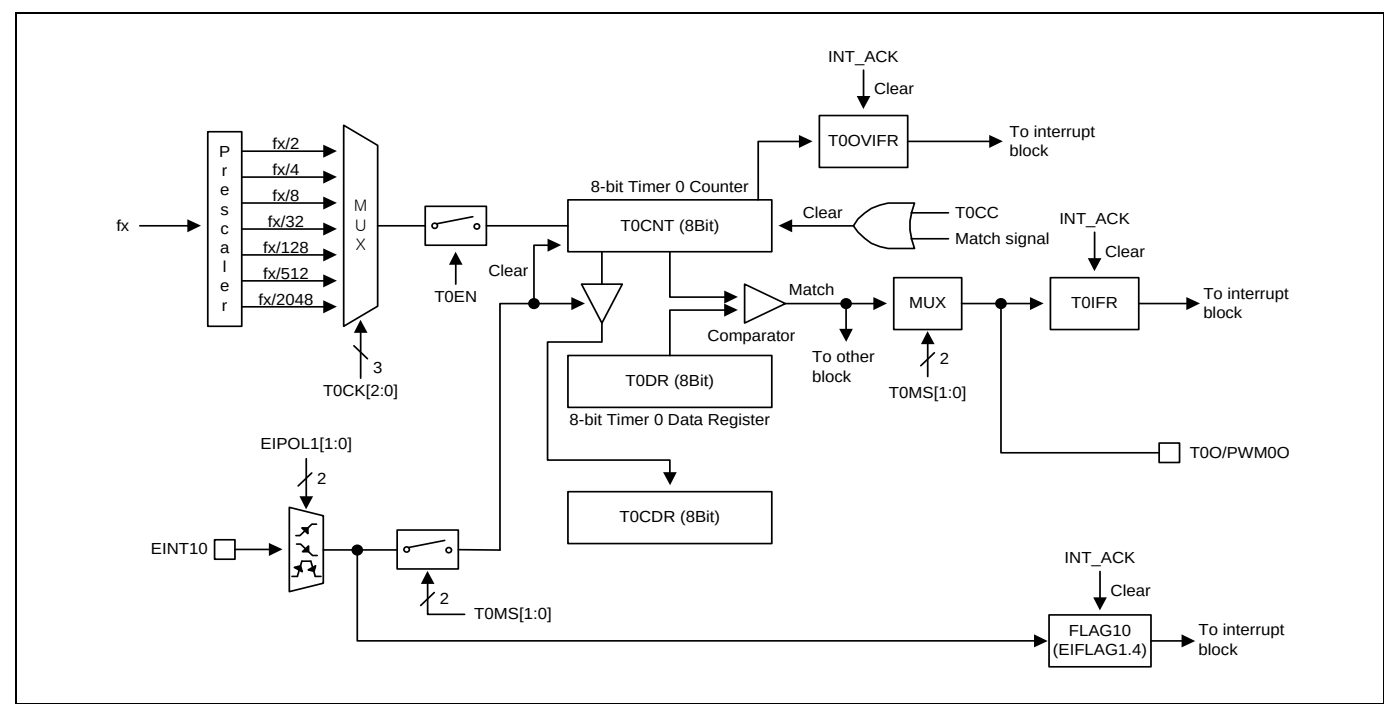


Figure 11.13 8-bit Timer 0 框图

11.5.6 寄存器

Name	Address	Direction	Default	Description
T0CNT	B3H	R	00H	Timer 0 Counter Register
T0DR	B4H	R/W	FFH	Timer 0 Data Register
T0CDR	B4H	R	00H	Timer 0 Capture Data Register
T0CR	B2H	R/W	00H	Timer 0 Control Register

Table 11.6 Timer 0 寄存器

11.5.7 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.

11.5.8 定时器/计数器 0寄存器

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 Data

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 Capture Data

T0CR (Timer 0 控制寄存器) : B2H

7	6	5	4	3	2	1	0
T0EN	—	T0MS1	T0MS0	T0CK2	T0CK1	T0CK0	T0CC
RW	—	RW	RW	RW	RW	RW	RW

Initial value : 00H

T0EN 控制 Timer 0

0 Timer 0 禁止

1 Timer 0 使能

T0MS[1:0] 控制 Timer 0 操作模式

T0MS1 T0MS0 Description

0 0 定时器/计数器 模式 (T0O: 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 Timer 0 时钟源. fx 是系统时钟频率

T0CK2 T0CK1 T0CK0 Description

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 Not available

T0CC 清除 timer 0 计数器

0 No effect

1 清除定时器 0 计数器 (When write, automatically cleared "0" after being cleared counter)

NOTE)

1. 参考内部中断标志位寄存器 (IIFLAG) T0 中断标志位.

11.6 定时器 1

11.6.1 概述

16-bit 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 (one-shot mode)
 - 16-bit PPG 输出模式 (repeat mode)

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.

- TIMER 1 时钟源: 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 Timer 1 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.

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)

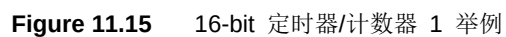
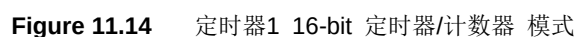
Table 11.7 Timer 1 工作模式

11.6.2 16-bit 定时器/计数器 模式

图11.14 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 。Timer 1 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P11IO位设置为输入状态.



11.6.3 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/T1BDRL.

根据 EIPOL1 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.

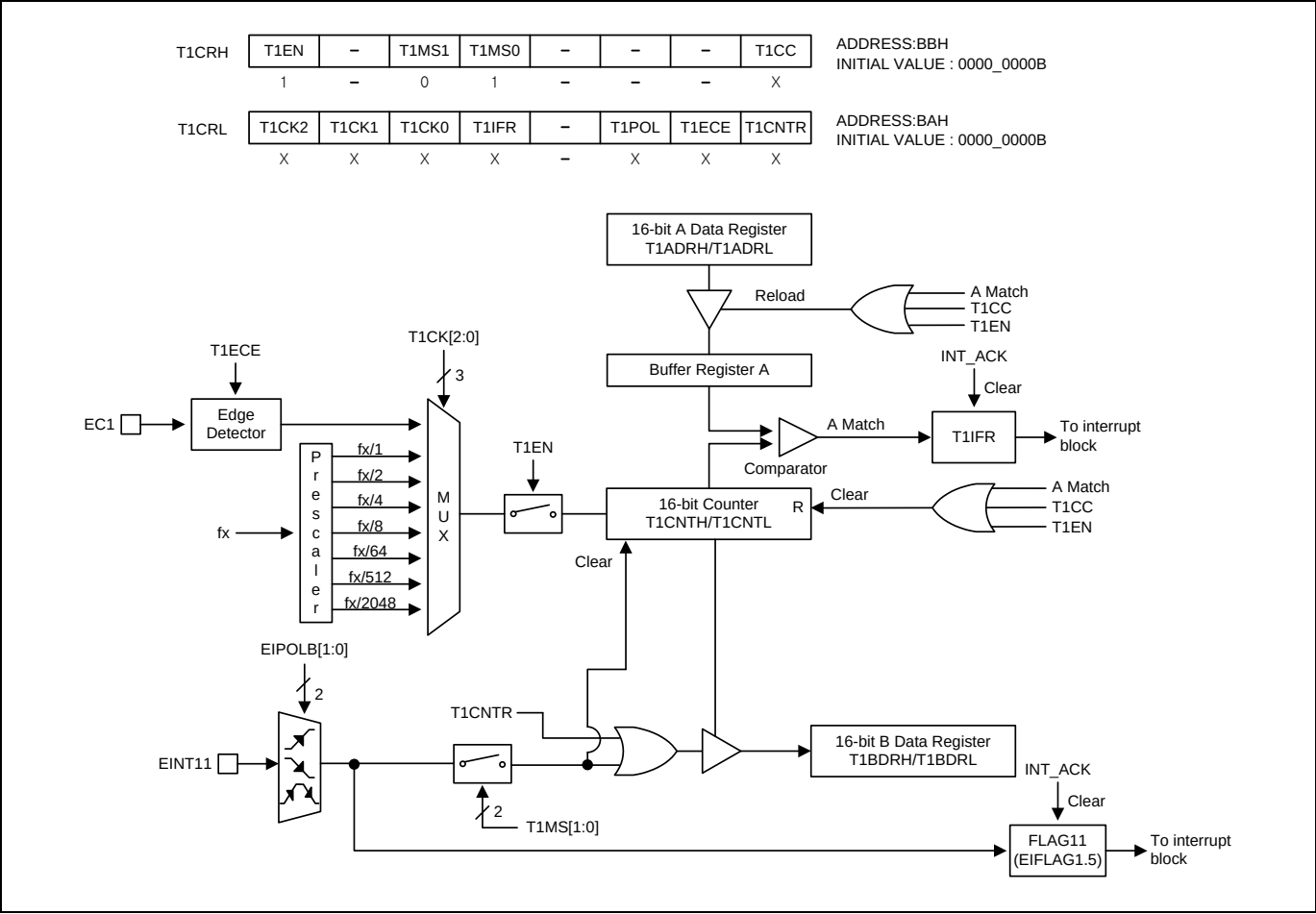


Figure 11.16 定时器1 16-bit 捕捉模式

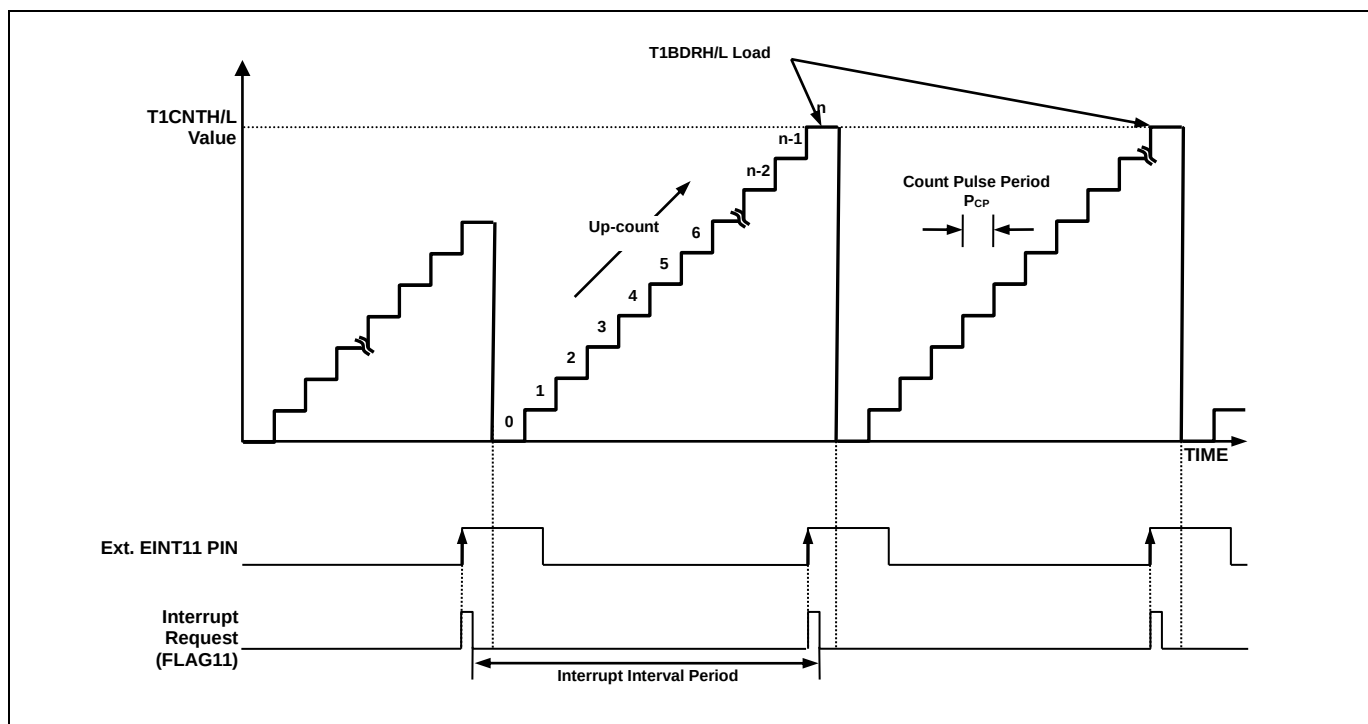


Figure 11.17 Timer 1 输入捕捉模式操作

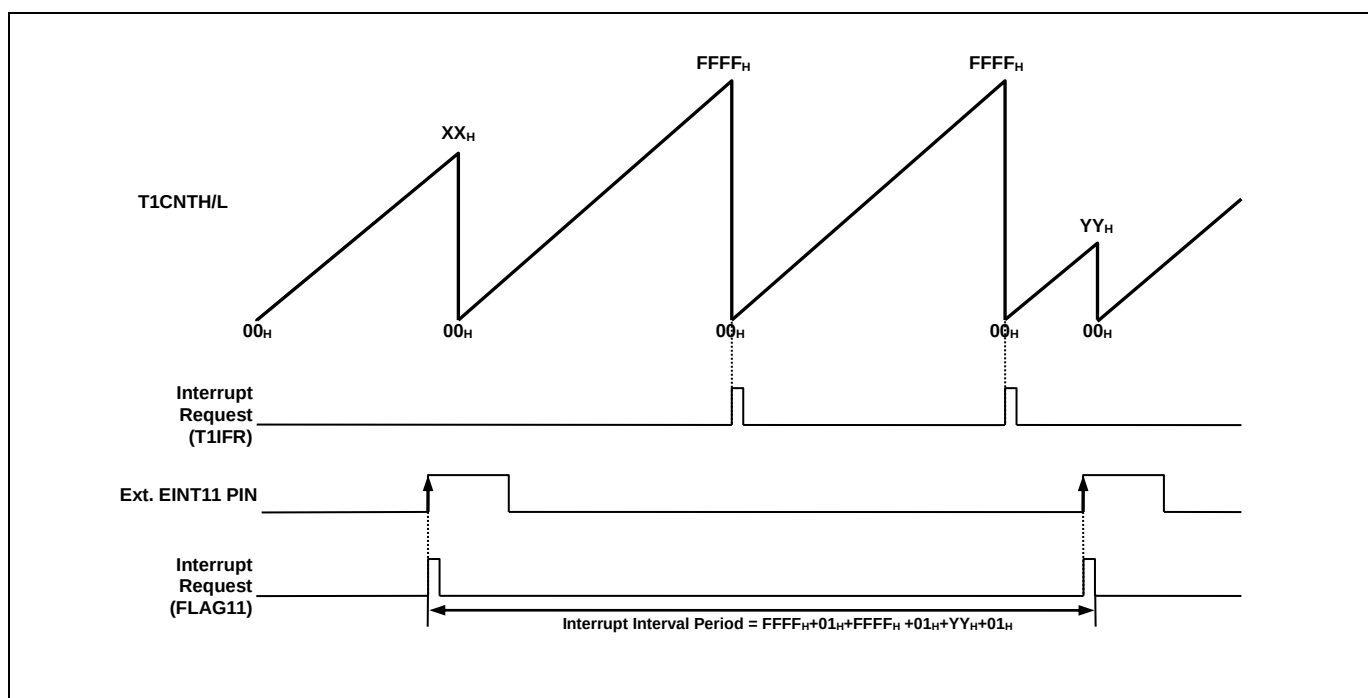


Figure 11.18 捕捉模式定时器溢出

11.6.4 16-bit PPG 模式

定时器1有PPG (可编程脉冲发生) 功能。 PPG 模式中, 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.

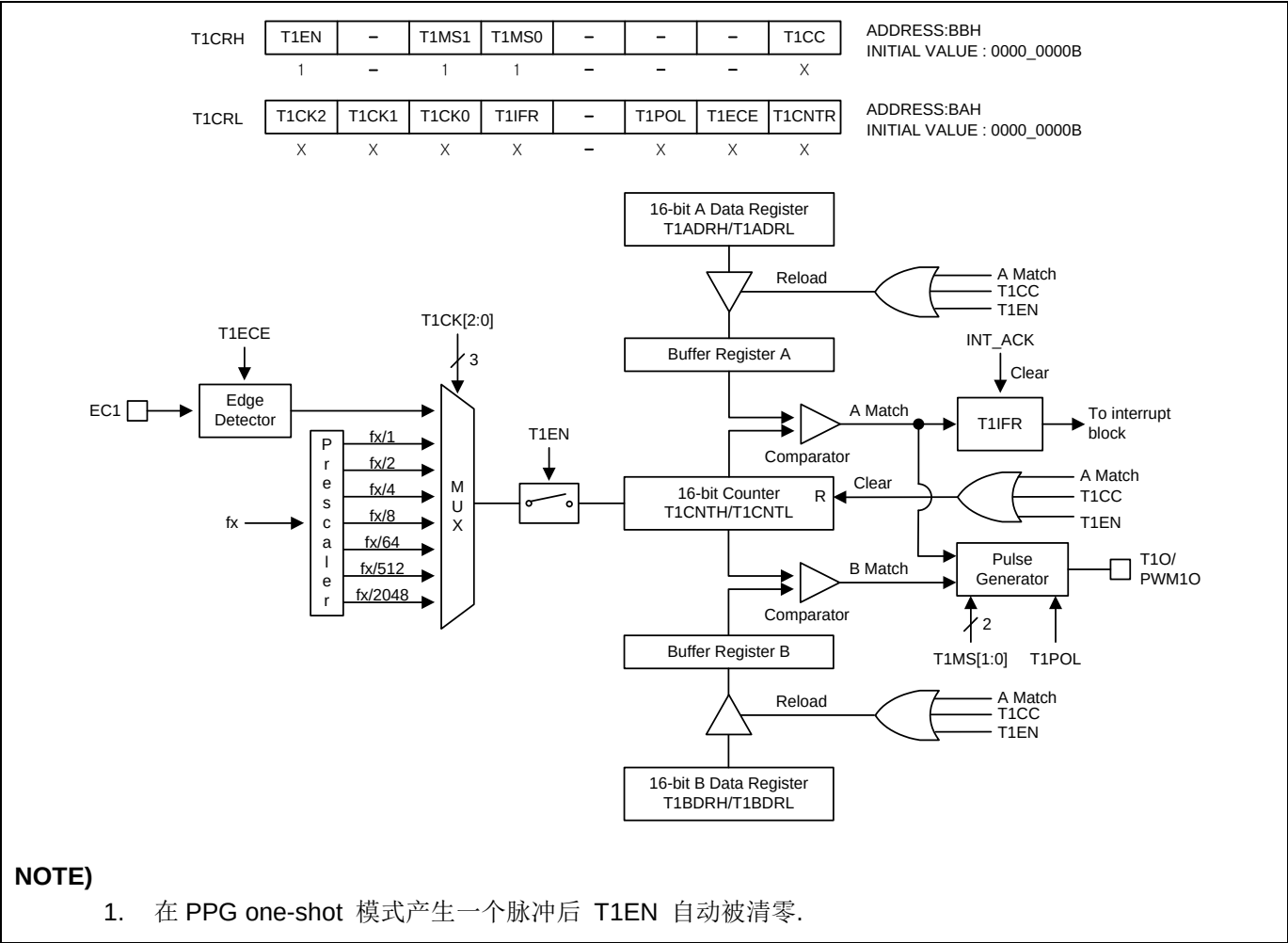
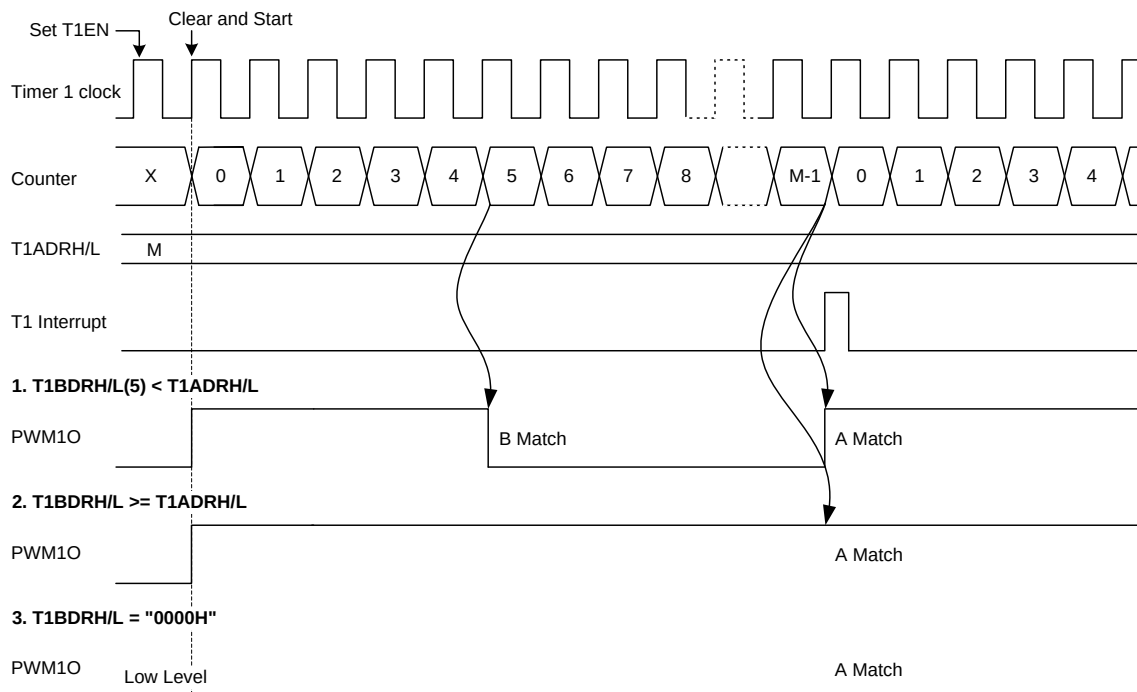
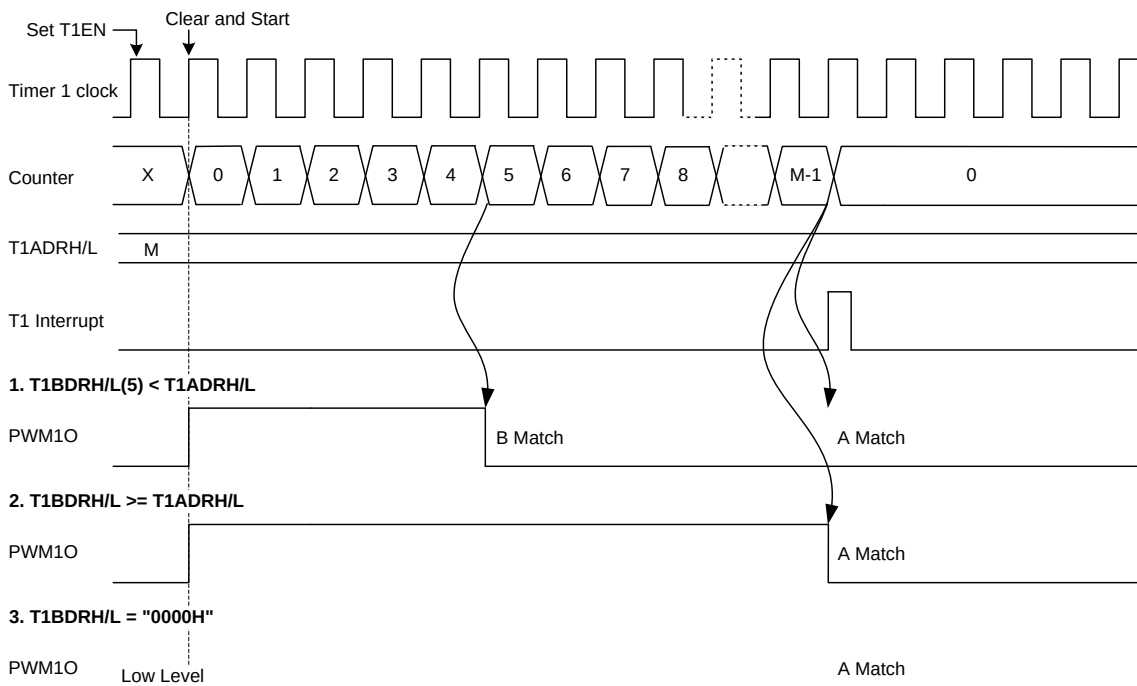


Figure 11.19 定时器1 16-bit PPG 模式

Repeat Mode(T1MS = 11b) and "Start High"(T1POL = 0b).**One-shot Mode(T1MS = 10b) and "Start High"(T1POL = 0b).****Figure 11.20** 定时器1 16-bit PPG 模式时序图

11.6.5 框图

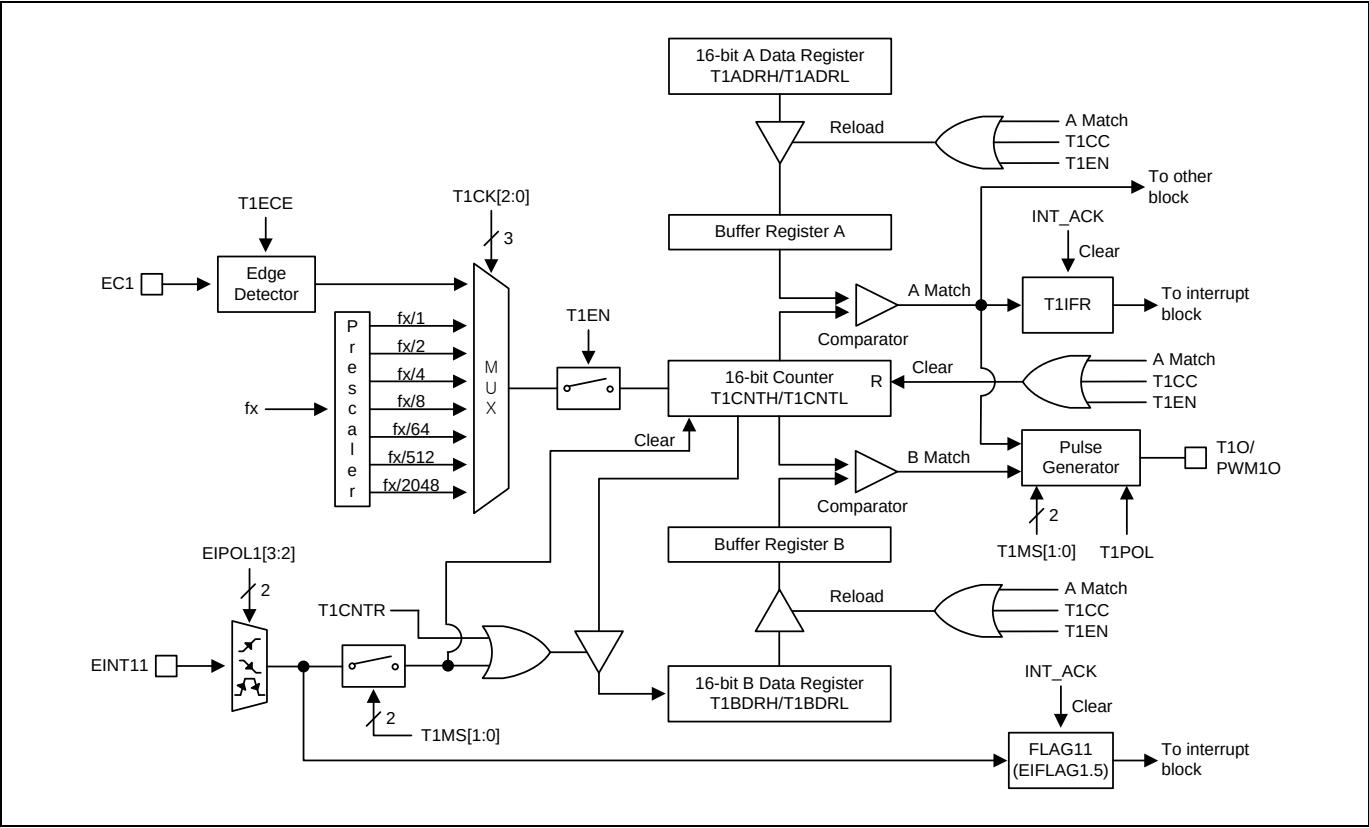


Figure 11.21 16-bit Timer 1框图

11.6.6 寄存器

Name	Address	Direction	Default	Description
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

Table 11.8 Timer 2 寄存器

11.6.7 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.

11.6.8 定时器/计数器 1 寄存器

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] T1 A 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] T1 A 数据低字节

NOTE)

1. 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] T1 B 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] T1 B 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
RW	–	R/W	RW	–	–	–	RW

Initial value : 00H

T1EN 控制 Timer 1

0 Timer 1 禁止

1 Timer 1 使能 (计数器清零后开始计数)

T1MS[1:0] 控制 Timer 1 操作模式

T1MS1 T1MS0 Description

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 模式 (PWM1O)

1 1 PPG repeat 模式 (PWM1O)

T1CC 清除 Timer 1 计数

0 No effect

1 清除Timer 1 计数 (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	—	RW	RW	RW

Initial value : 00H

T1CK[2:0] 选择 Timer 1 时钟源. 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. 写入1无效

0 T1 中断没有发生

1 T1 中断发生

T1POL T1O/PWM1O 极性选择

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 Timer 1 外部时钟脉冲边沿选择

0 外部时钟下降沿

1 外部时钟上升沿

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)

11.7 定时器 2

11.7.1 概述

16位定时器 2 由数据转换器, T2ADRH, T2ADRL, T2BDRH, T2BDRL, T2CRH, T2CRL组成.

有4个工作模式:

- 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T1 A Match提供时钟.通过控制设置(T2CK[2:0]) 位来选取时钟.

- TIMER 2 时钟源: $f_x/1, 2, 4, 8, 64, 512, 2048$ 和 T1 A 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.

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)

Table 11.9 Timer 2 工作模式

11.7.2 16-bit 定时器/计数器 模式

图11.22 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timer 1 A Match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 T1 A 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).

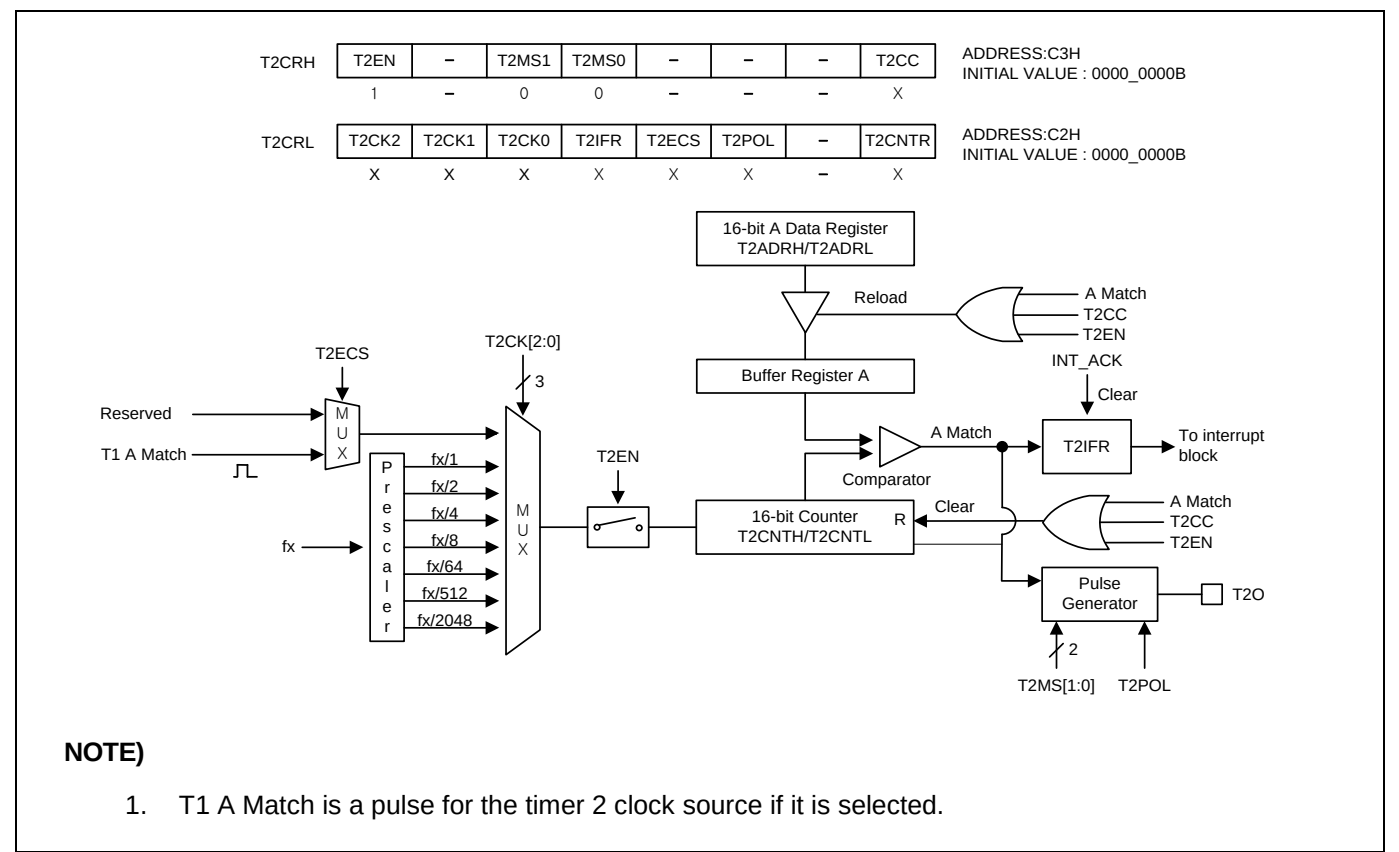


Figure 11.22 定时器2 16-bit 定时器/计数器 模式

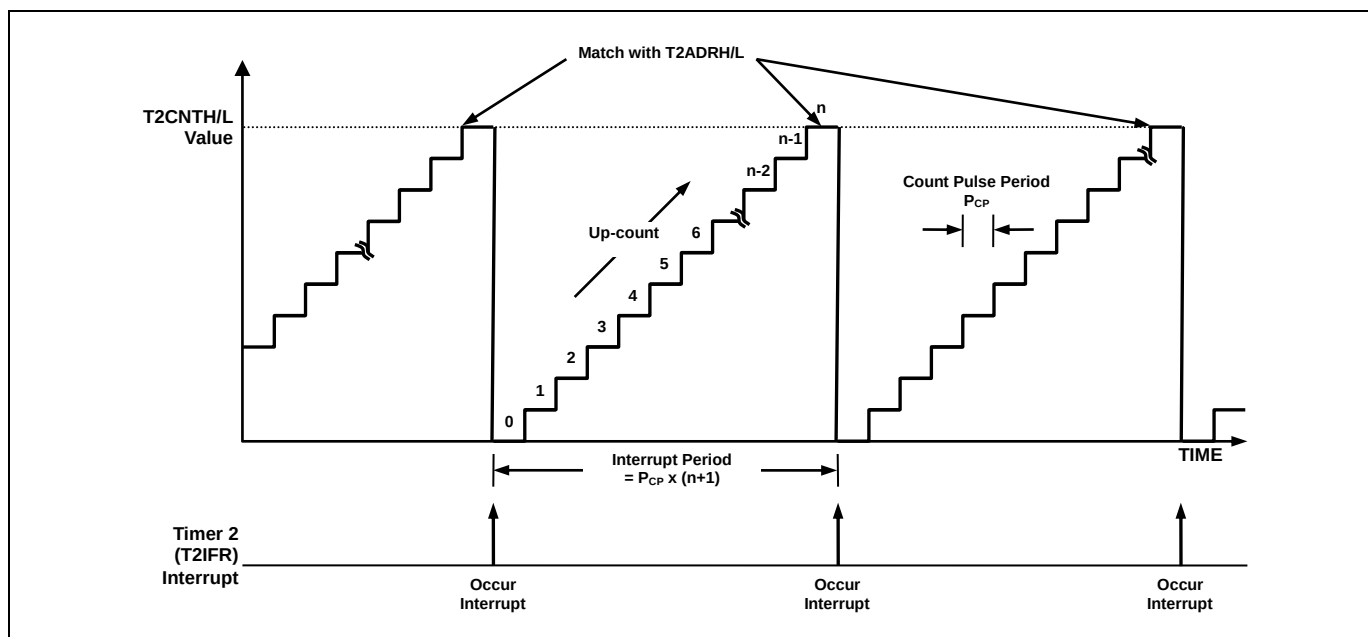


Figure 11.23 16-bit 定时器/计数器 2 举例

11.7.3 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, T2CNTL 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.

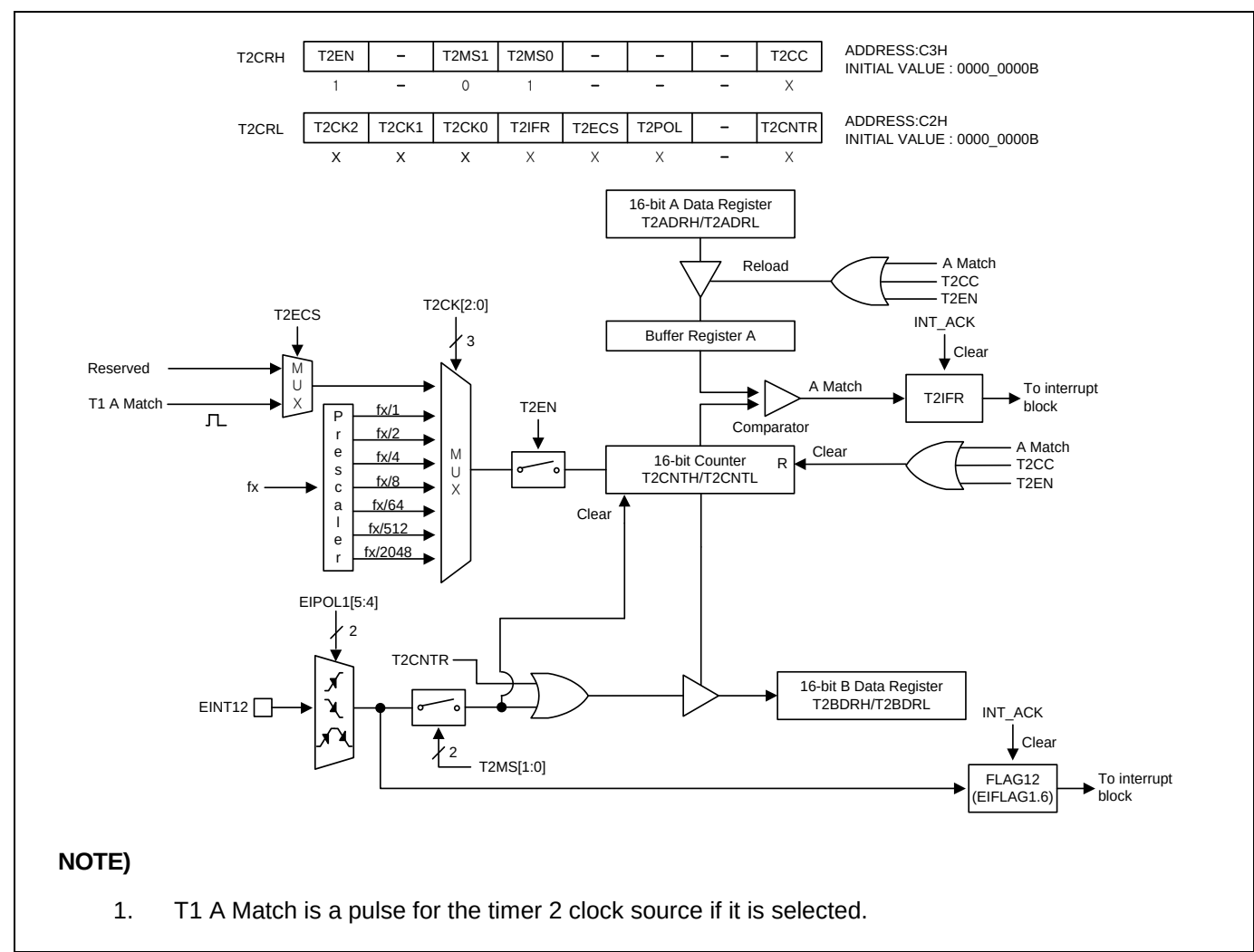


Figure 11.24 定时器2 16-bit 捕捉模式

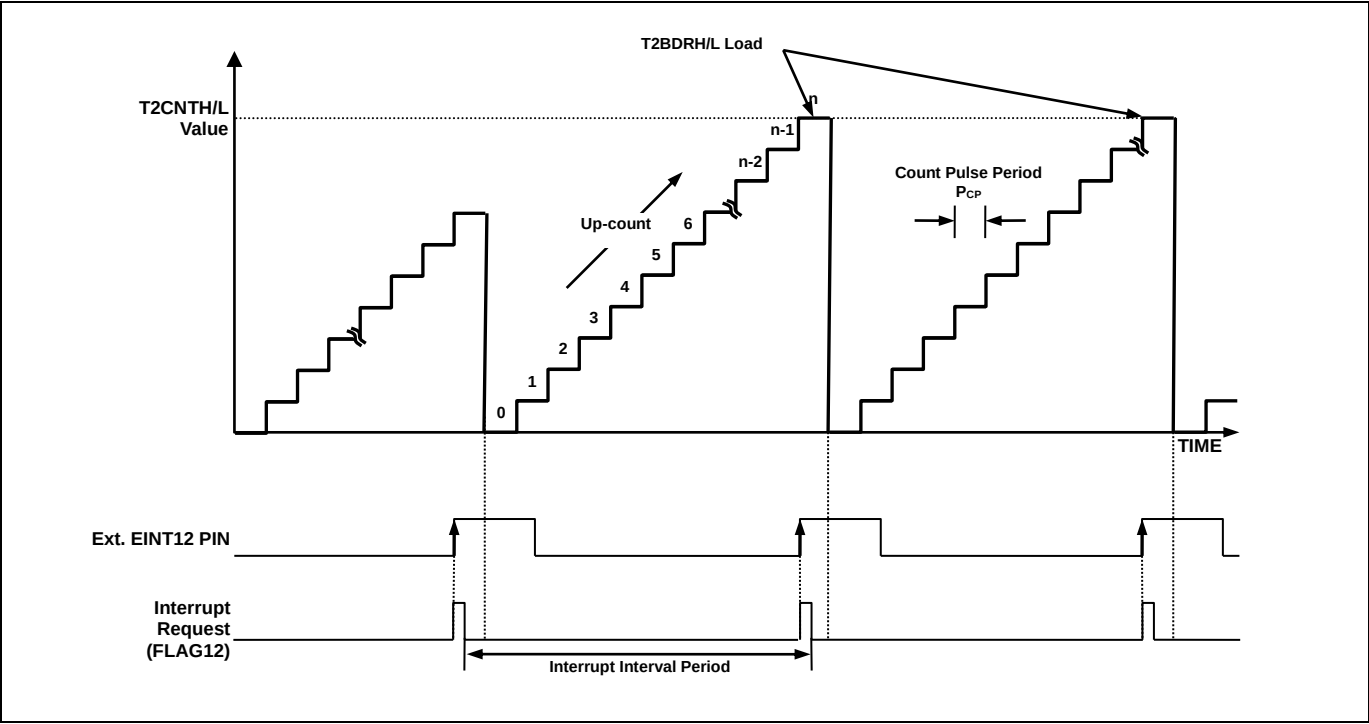


Figure 11.25 定时器2输入捕捉模式操作

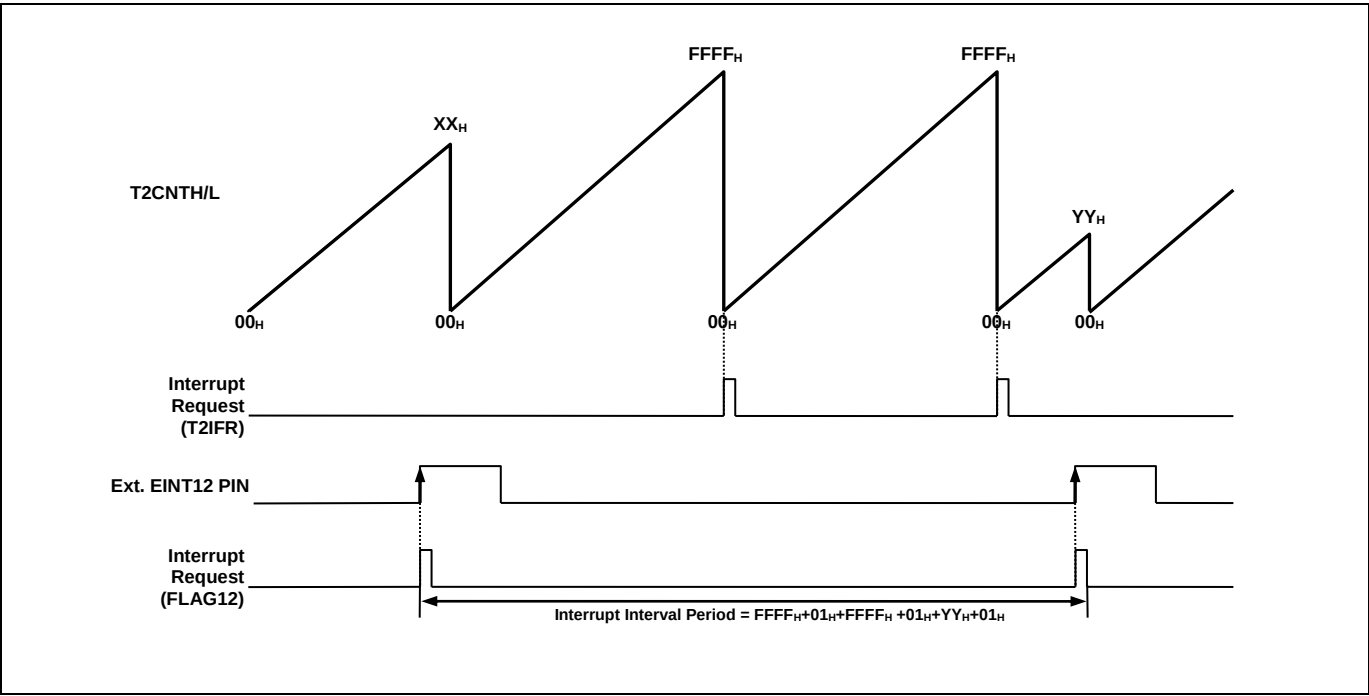


Figure 11.26 捕捉模式定时器溢出

11.7.4 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.

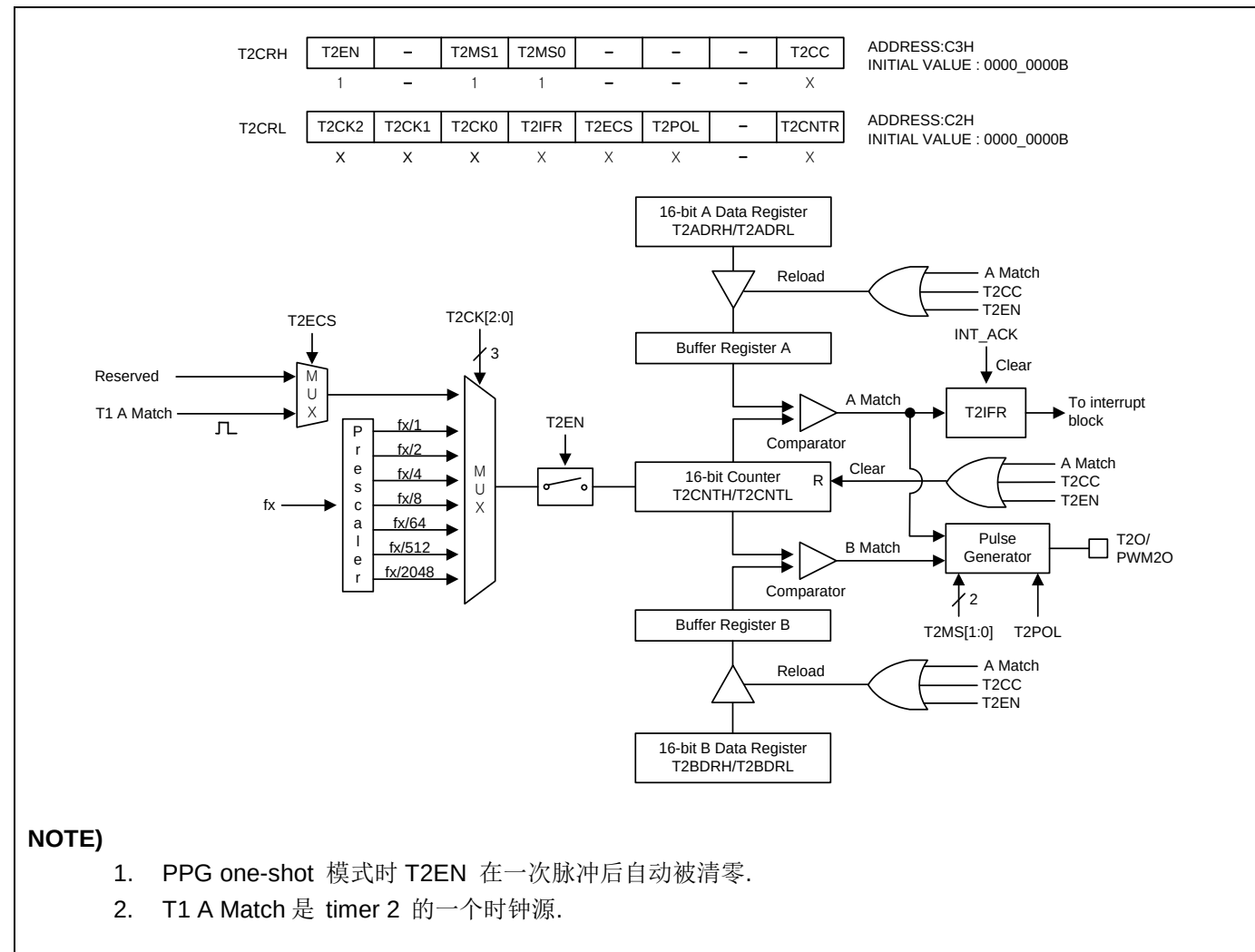


Figure 11.27 定时器2 16-bit PPG 模式

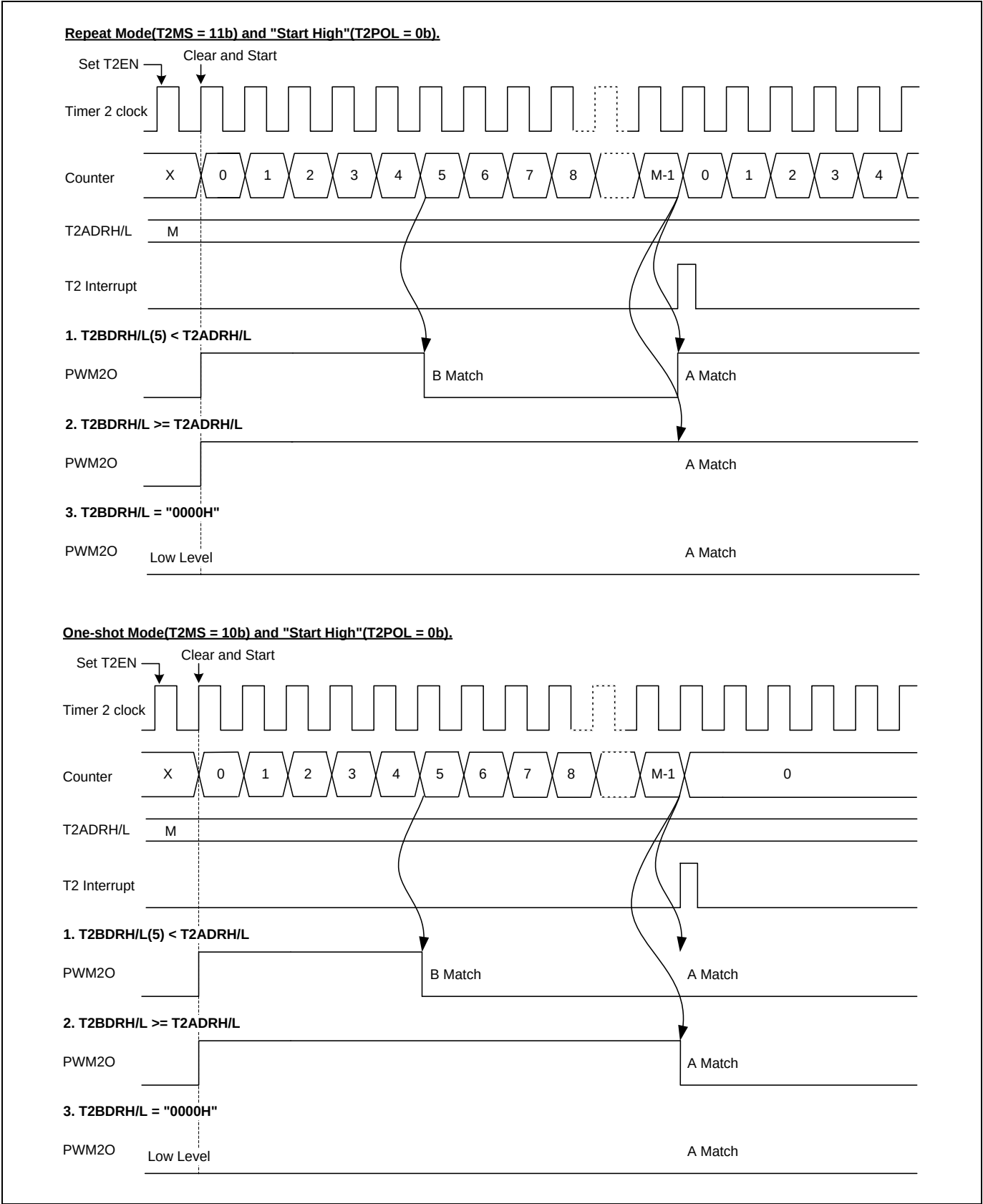


Figure 11.28 定时器2 16-bit PPG 模式时序图

11.7.5 方框图

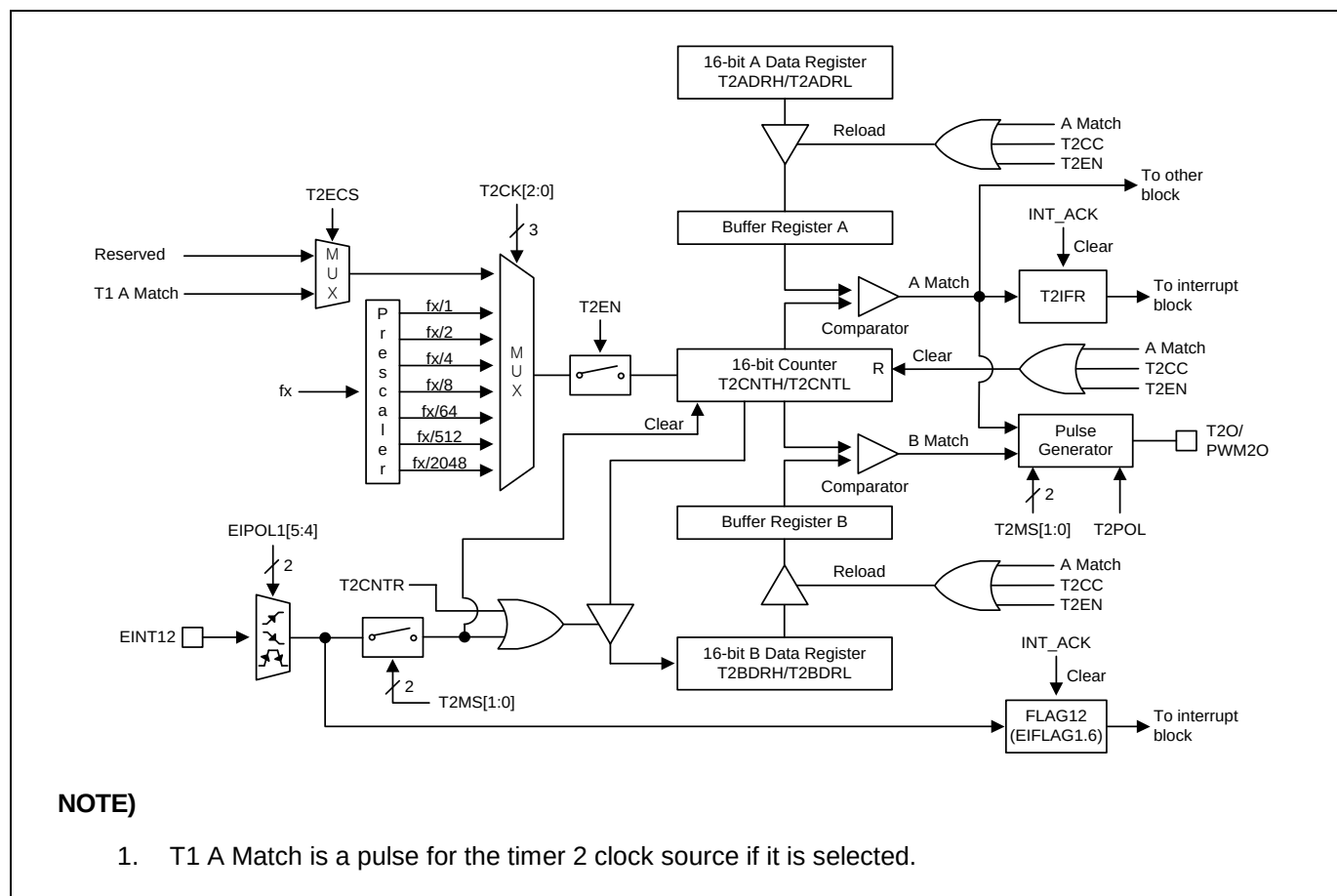


Figure 11.29 16-bit Timer 2 方框图

11.7.6 寄存器

Name	Address	Direction	Default	Description
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

Table 11.10 Timer 2 寄存器

11.7.7 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.

11.7.8 定时器/寄存器 2 寄存器

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] T2 A Data High Byte

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] T2 A Data Low Byte

NOTE)

1. 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] T2 B Data High Byte

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] T2 B Data Low

T2CRH (Timer 2 控制高位寄存器) : C3H

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

Initial value : 00H

T2EN 控制 Timer 2

0 Timer 2 禁止

1 Timer 2 使能 (Counter clear and start)

T2MS[1:0] 控制 Timer 2 工作模式

T2MS1 T2MS0 Description

0 0 定时器/计数器 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 模式 (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 控制低位寄存器) : C2H

7	6	5	4	3	2	1	0
T2CK2	T2CK1	T2CK0	T2IFR	T2ECS	T2POL	–	T2CNTR
R/W	R/W	R/W	R/W	R/W	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 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 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，写入 “1” 无效.		
	0	没有T2中断产生	
	1	有T2中断产生	
T2ECS	Timer 2 外部时钟选择		
	0	Reserved	
	1	选择 Timer 1 A match	
	NOTE)		
	1. 该位一般为 ‘1’.		
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)	
T2CNTR	Timer 2计数读取控制		
	0	No effect	
	1	加载计数值到 B 数据寄存器(写入时，加载后自动清零)	

11.8 蜂鸣器驱动

11.8.1 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~125.0 kHz @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.

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

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

Table 11.11 蜂鸣器频率（8 MHz）

11.8.2 方框图

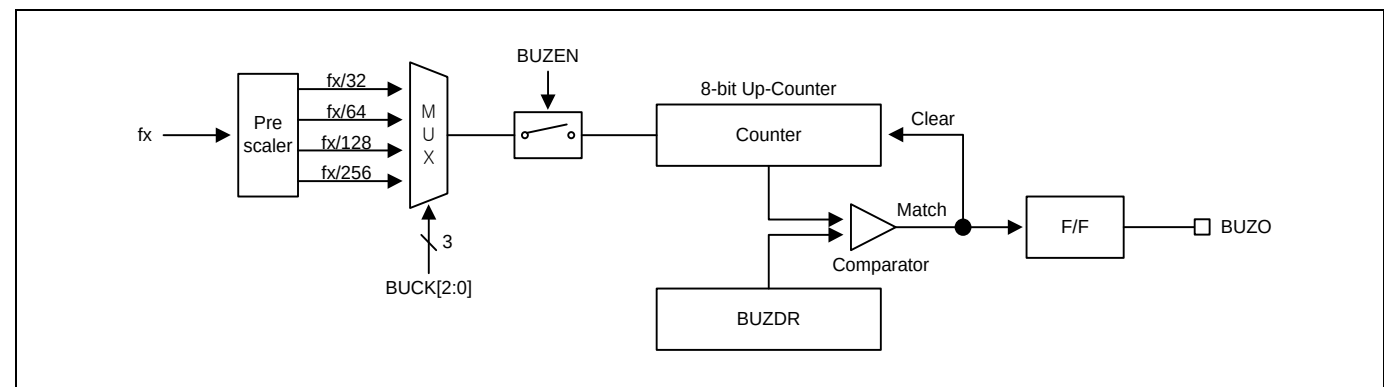


Figure 11.30 蜂鸣器驱动方框图

11.8.3 寄存器图

Name	Address	Direction	Default	Description
BUZDR	8FH	R/W	FFH	Buzzer Data Register
BUZCR	97H	R/W	00H	Buzzer Control Register

Table 11.12 蜂鸣器驱动寄存器

11.8.4 蜂鸣器驱动寄存器说明

蜂鸣器驱动由 蜂鸣器数据寄存器 (BUZDR) 和蜂鸣器控制寄存器 (BUZCR)组成.

11.8.5 蜂鸣器驱动寄存器

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
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] 蜂鸣器时钟选择

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	Not available

BUZEN 蜂鸣器操作控制

0	蜂鸣器禁止
1	蜂鸣器使能

NOTE)

1. fx: 系统时钟振荡器频率.

11.9 12-bit A/D 转换器

11.9.1 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(ADCDDL). 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 ADCDDL包含AD的转换结果。AD转换完成，转换结果保存到ADCDRH 和 ADCDDL, A D转换状态位置 '1', AD中断产生.在AD转换中, AFLAG 位被读作'0'.

11.9.2 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0.66μs.每位转换需要4个时钟，转换率计算如下：

$$\begin{aligned} 4 \text{ clocks/bit} \times 12 \text{ bits} + \text{设置时间} &= 58 \text{ clocks,} \\ 58 \text{ clock} \times 0.66 \mu\text{s} &= 38.28 \mu\text{s at 1.5 MHz (12 MHz/8)} \end{aligned}$$

NOTE)

1. 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.

11.9.3 方框图

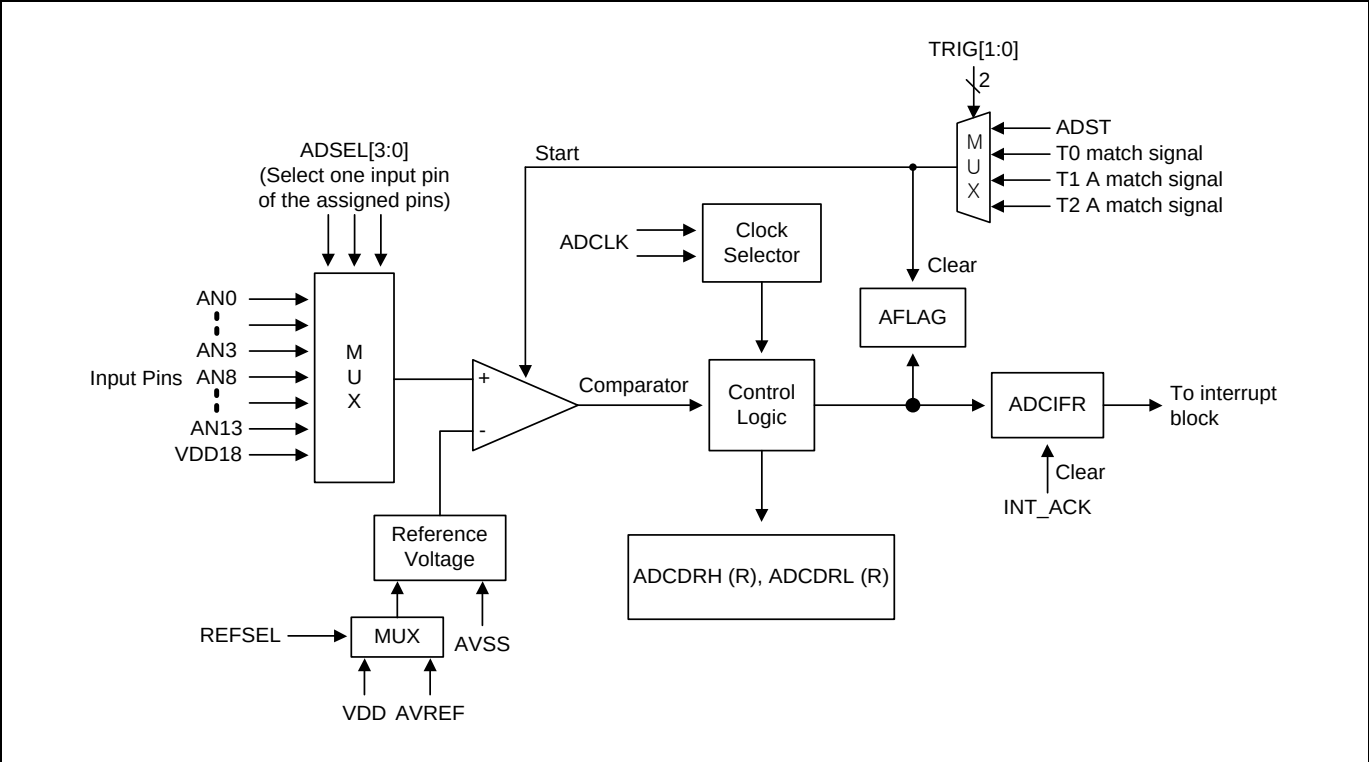


Figure 11.31 12-bit ADC框图

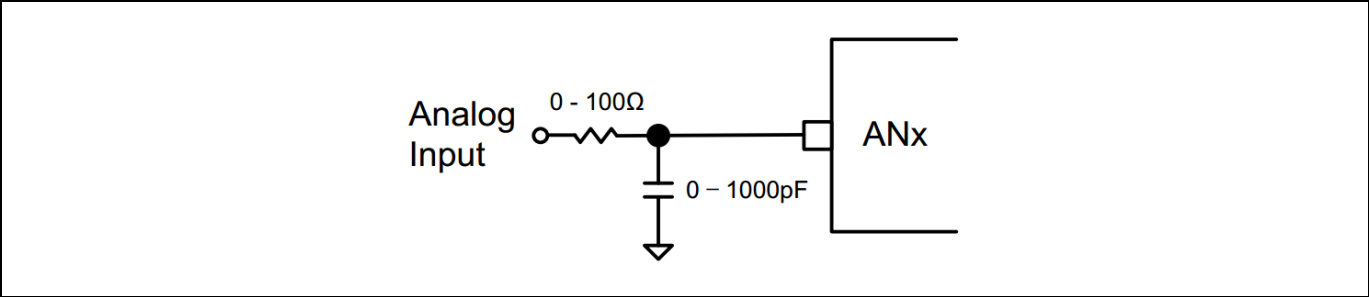


Figure 11.32 A/D 模式输入脚电容

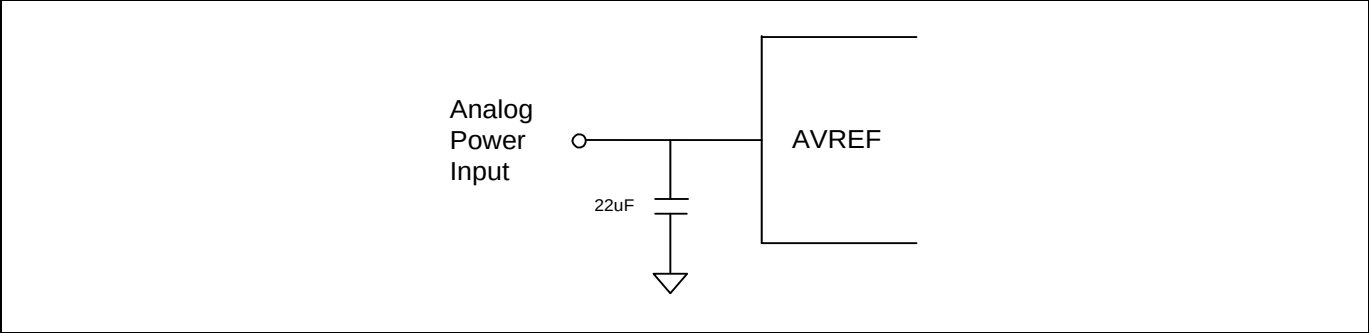


Figure 11.33 A/D Power (AVREF) Pin 电阻

11.9.4 ADC 操作

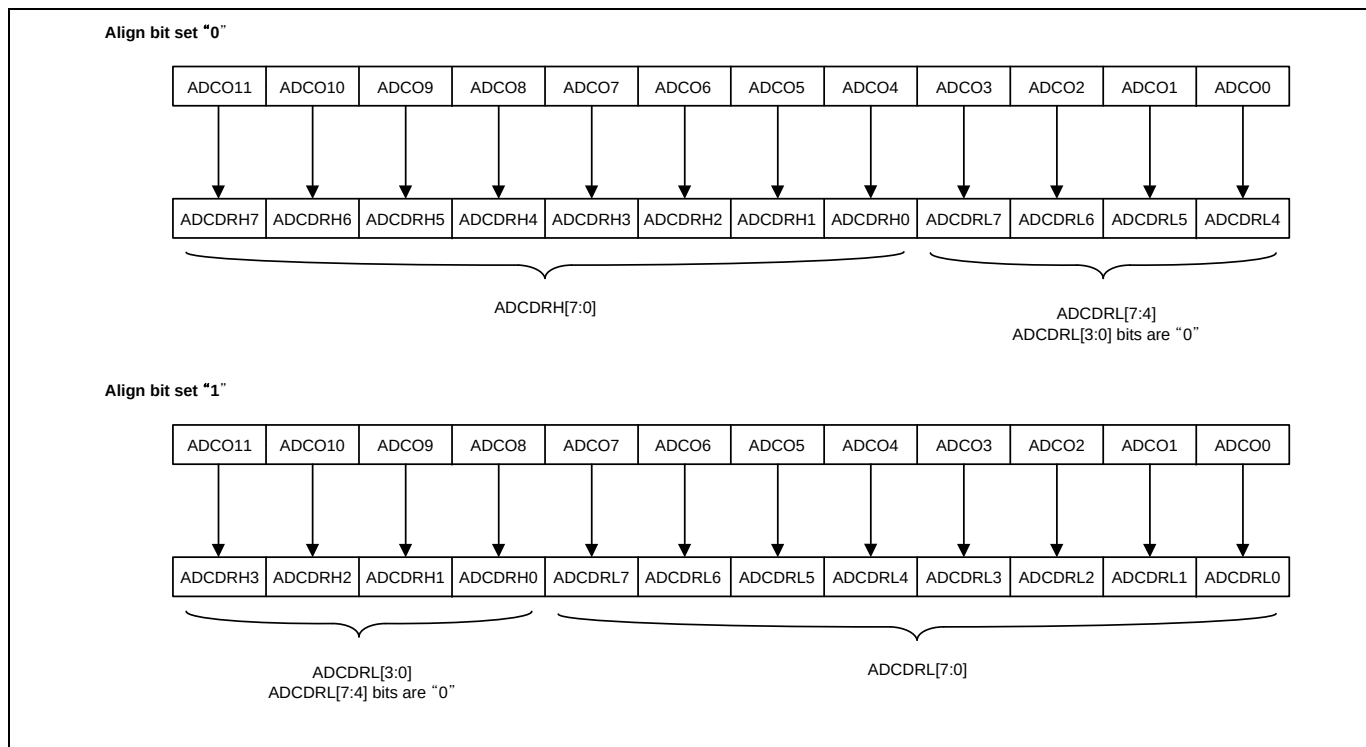


Figure 11.34 ADC Operation for Align Bit

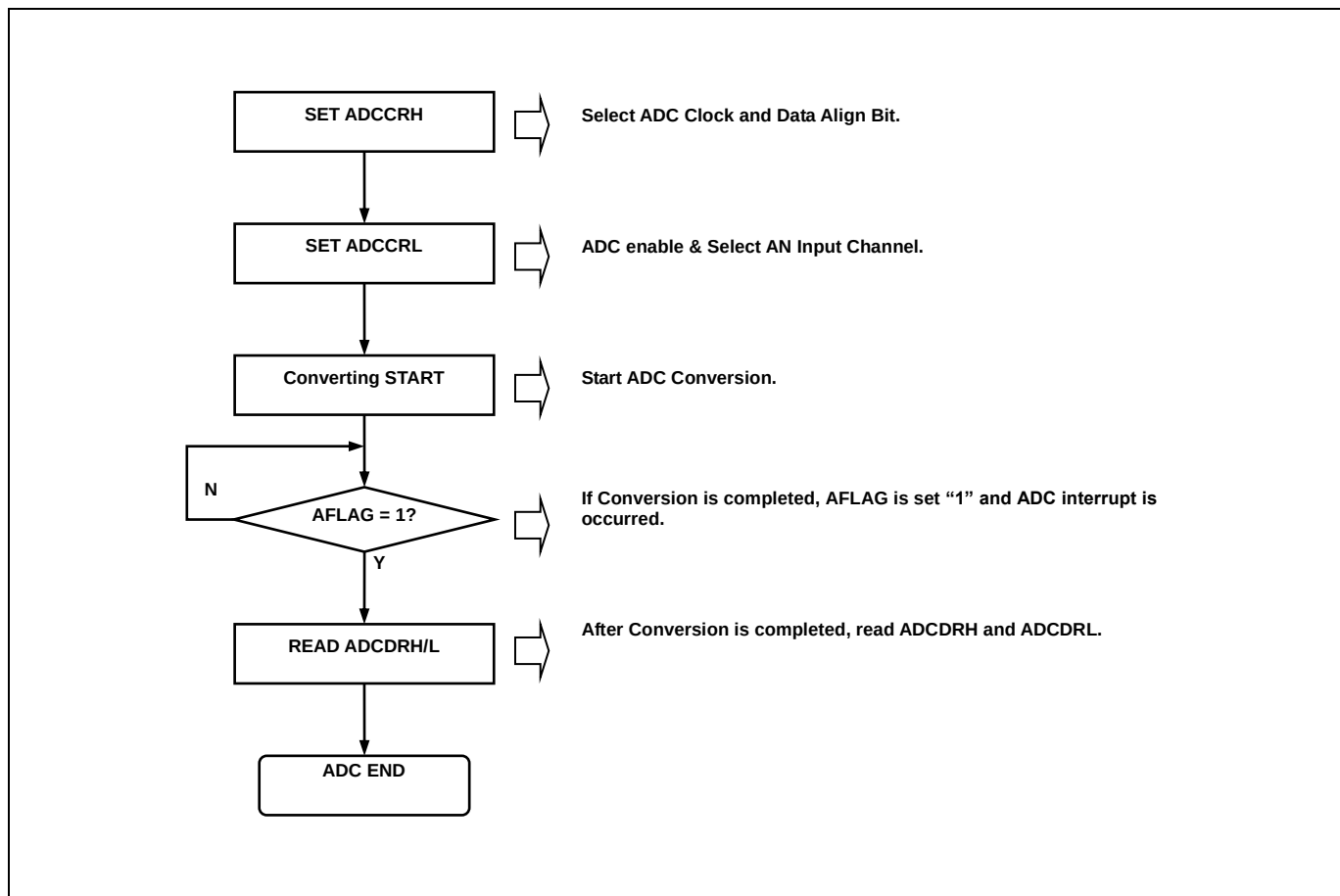


Figure 11.35 A/D 转换流程

11.9.5 ADC 睡眠唤醒功能

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 $n = 0, 1, 2, \dots, 13$.

11.9.6 ADC 睡眠唤醒功能框图

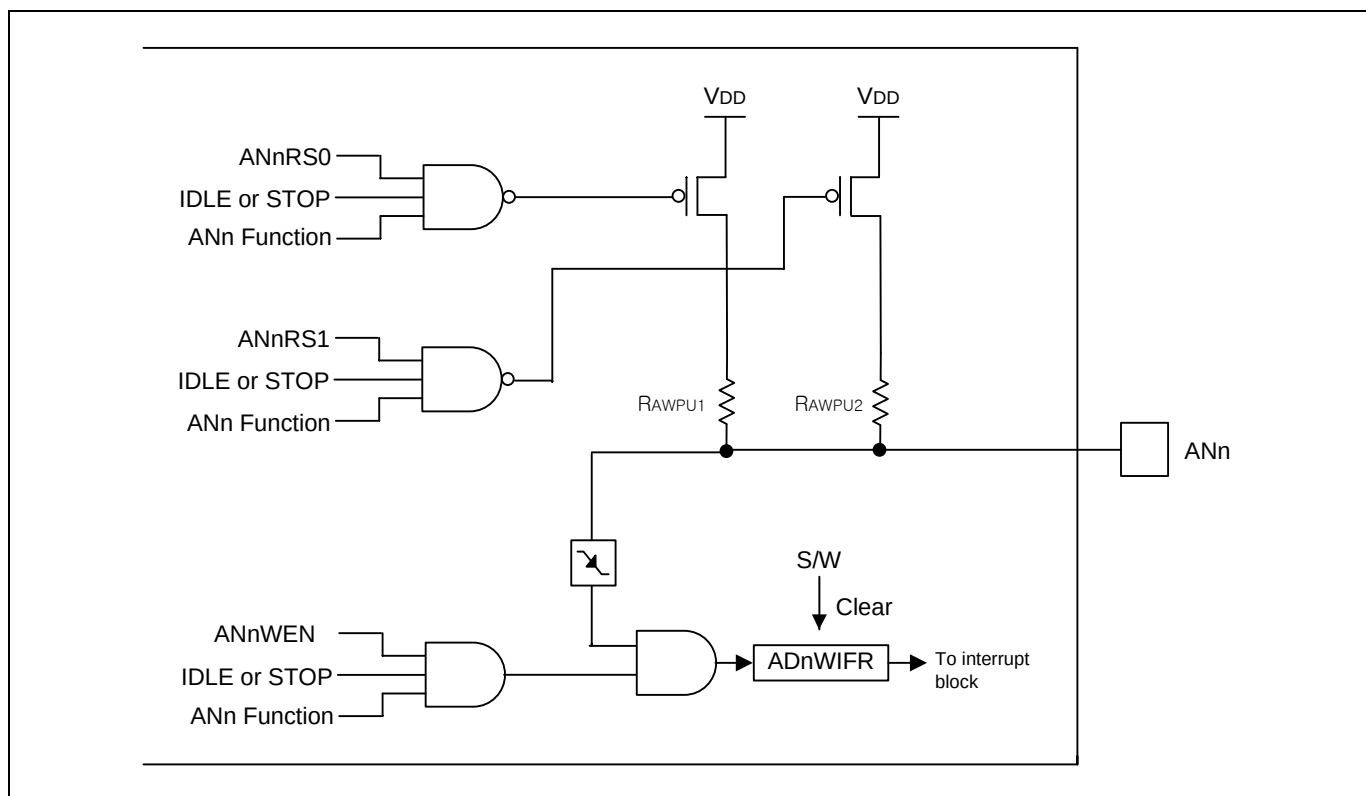


Figure 11.36 ADC 睡眠唤醒功能框图

NOTE)

1. AN0-AN3, AN8-AN13 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 $n = 0, 1, 2, 3, 8, 9, 10, 11, 12,$ 和 13 .

11.9.7 ADC 睡眠唤醒功能操作

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.

11.9.8 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。

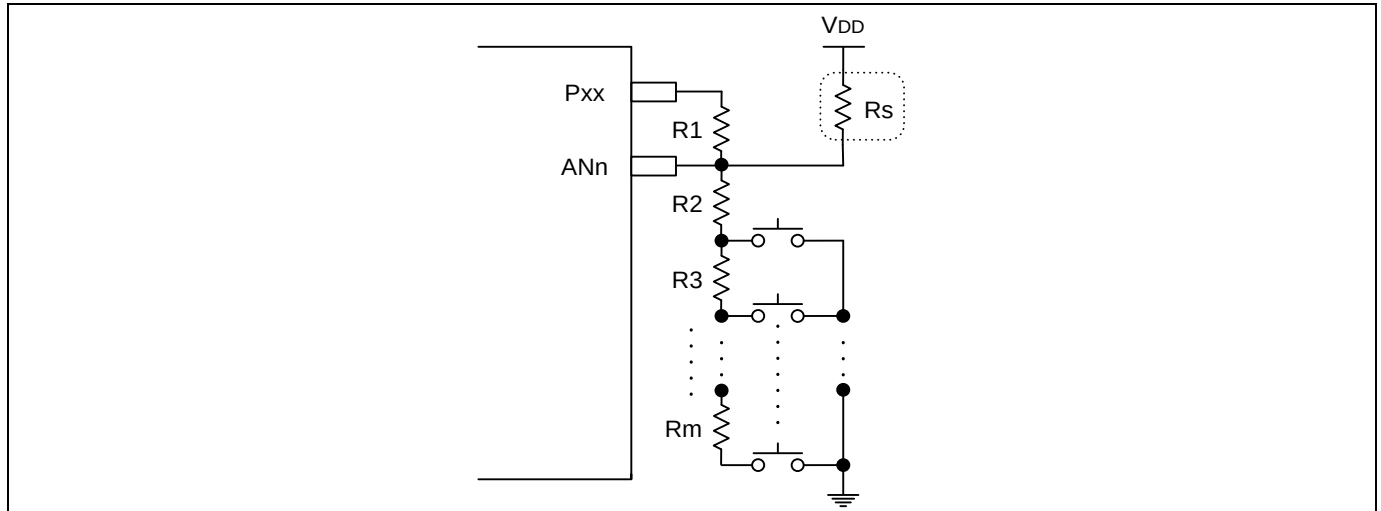


Figure 11.37 ADC 按键输入串联电阻应用电路

NOTE)

1. 推荐 R_s 使用 $1M\Omega$.
2. P_{xx} 是普通 I/O 脚.
3. 这里 $n = 0, 1, 2, 3, 8, 9, 10, 11, 12$, 和 13 .

11.9.9 寄存器图

Name	Address	Direction	Default	Description
ADCDRH	9FH	R	xxH	A/D Converter Data High Register
ADCDDL	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
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

Table 11.13 ADC 寄存器

11.9.10 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 (ADCDDL), 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

11.9.11 ADC寄存器说明

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
ADDM3 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 转换器控制高位寄存器) : 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. 写入 "1"无效

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 Timer 0 match signal

1 0 Timer 1 A match signal

1 1 Timer 2 A match signal

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 转换器控制低位寄存器) : 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 转换完成			
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	Not available
	0	1	0	1	Not available
	0	1	1	0	Not available
	0	1	1	1	Not available
	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	Not available
	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] ANn 输入引脚ADC 唤醒电阻选择

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

0 0 Disable Disable

0 1 Disable Enable

1 0 Enable Disable

1 1 Enable Enable

这里 n = 0, 1, 2, 和 3

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] ANn 输入引脚ADC 唤醒电阻选择

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

0 0 Disable Disable

0 1 Disable Enable

1 0 Enable Disable

1 1 Enable Enable

Where n = 8, 9, 10, and 11

ADWRCR3 (ADC 唤醒电阻控制寄存器3): F5H

7	6	5	4	3	2	1	0
–	–	–	–	AN13RS1	AN13RS0	AN12RS1	AN12RS0
–	–	–	–	RW	RW	RW	RW

Initial value: 00H

ADWRCR3[3:0] ANn 输入引脚ADC 唤醒电阻选择

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

Where n = 12 and 13

ADWCRH (ADC 唤醒控制高位寄存器): F7H

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

Initial value: 00H

ADWCRH[5:0] 使能或禁止Ann输入 ADC 唤醒功能

ANnWEN	Description
0	禁止
1	使能

Where n = 8, 9, 10, 11, 12 and 13

ADWCRL (ADC 唤醒控制低位寄存器): F6H

7	6	5	4	3	2	1	0
–	–	–	–	AN3WEN	AN2WEN	AN1WEN	AN0WEN
–	–	–	–	RW	RW	RW	RW

Initial value: 00H

ADWCRL[3:0] 使能或禁止Ann输入 ADC 唤醒功能

ANnWEN	Description
0	禁止
1	使能

Where n = 0, 1, 2 and 3

ADWIFRH (ADC 唤醒中断标志位高位寄存器): DDH

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

Initial value: 00H

ADWIFRH[5:0] 当唤醒中断AN13 ~ AN8 有一个发生时，相应位置 ‘1’。该位写入0清除，所以该标志位需要软件清除。写入 “1” 无效

- 0 ADC 唤醒中断 AN13 ~ AN8 没有发生
- 1 ADC 唤醒中断 AN13 ~ AN8 发生

ADWIFRL (ADC唤醒中断标志位低位寄存器): DCH

7	6	5	4	3	2	1	0
—	—	—	—	AN3WIFR	AN2WIFR	AN1WIFR	AN0WIFR
—	—	—	—	RW	RW	RW	RW

Initial value: 00H

ADWIFRL[3:0] 当唤醒中断 AN3 ~ AN0有一个发生时，相应位置 ‘1’。该位写入0清除，所以该标志位需要软件清除。写入 “1” 无效

- 0 ADC 唤醒中断 AN3 ~ AN0 没有发生
- 1 ADC 唤醒中断 AN3 ~ AN0 发生

11.10 SPI

11.10.1 概述

MC96F8208S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。

11.10.2 方框图

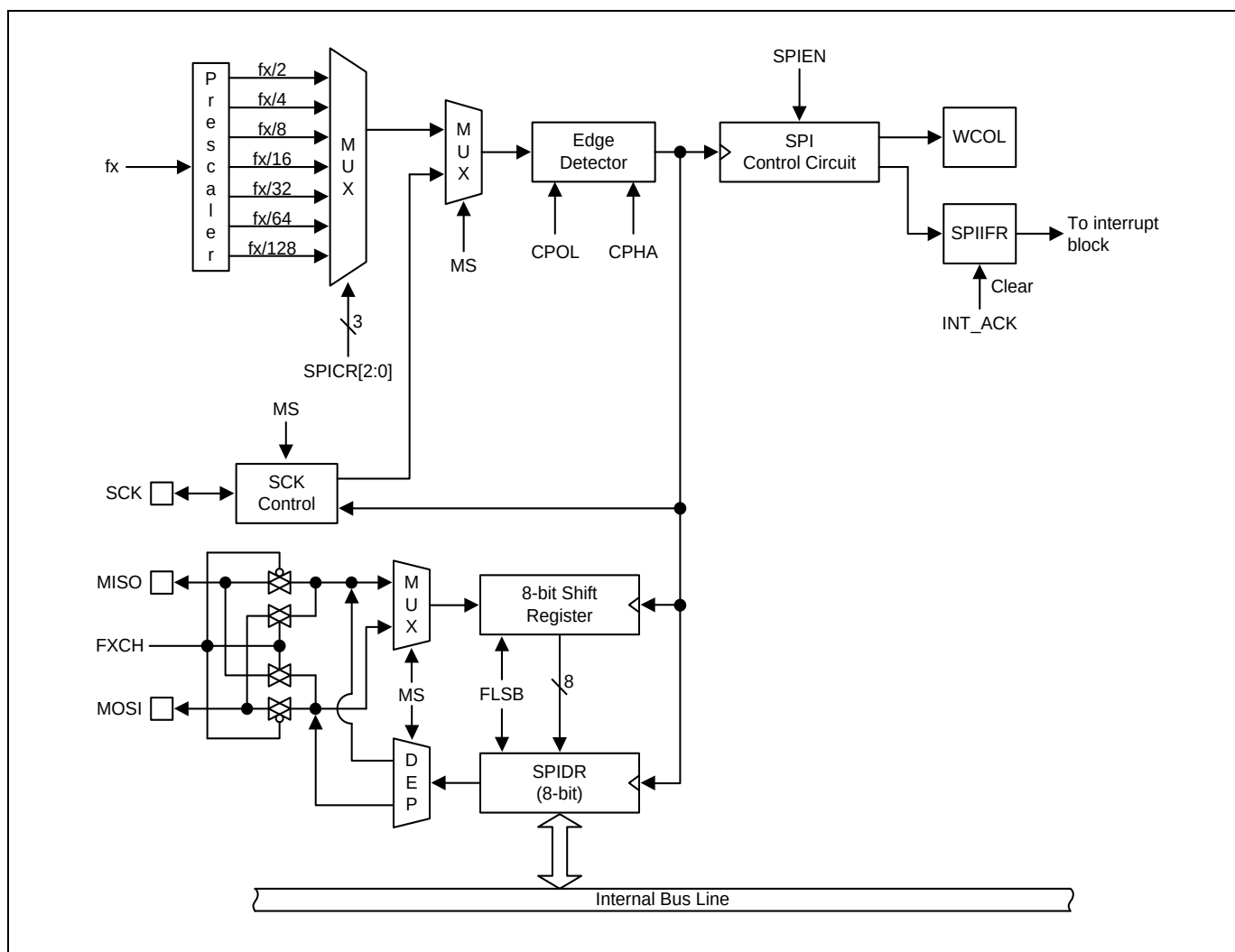


Figure 11.38 SPI 方框图

11.10.3 数据 发送/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. 用户写一个字节到数据寄存器SPIDR 时, SPI 将开始运行.
3. 这时, 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）.
4. 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.

11.10.4 SPI 时序图

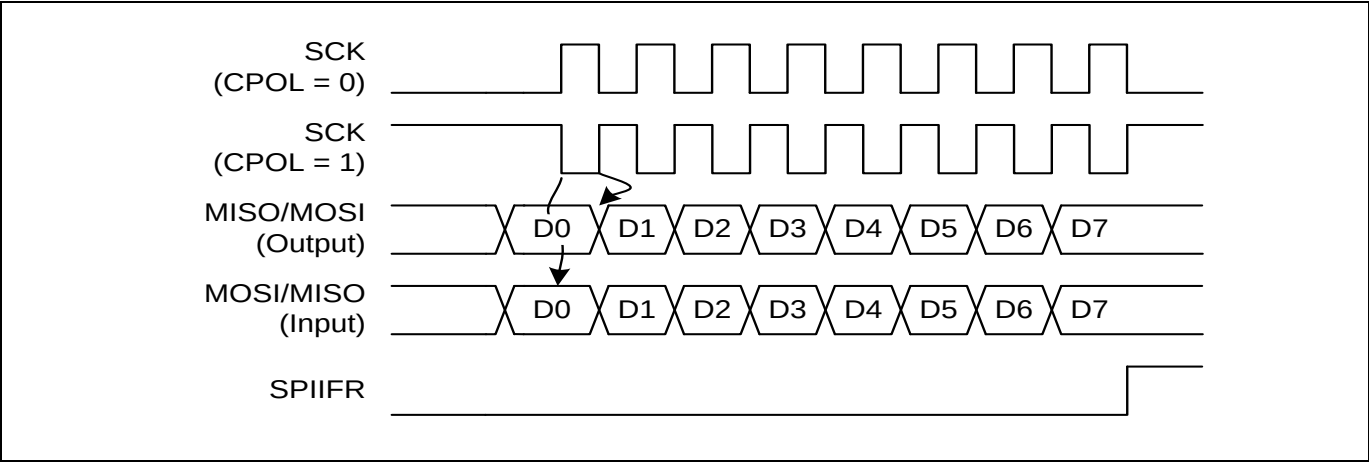


Figure 11.39 SPI 发送/接收 时序图 CPHA = 0

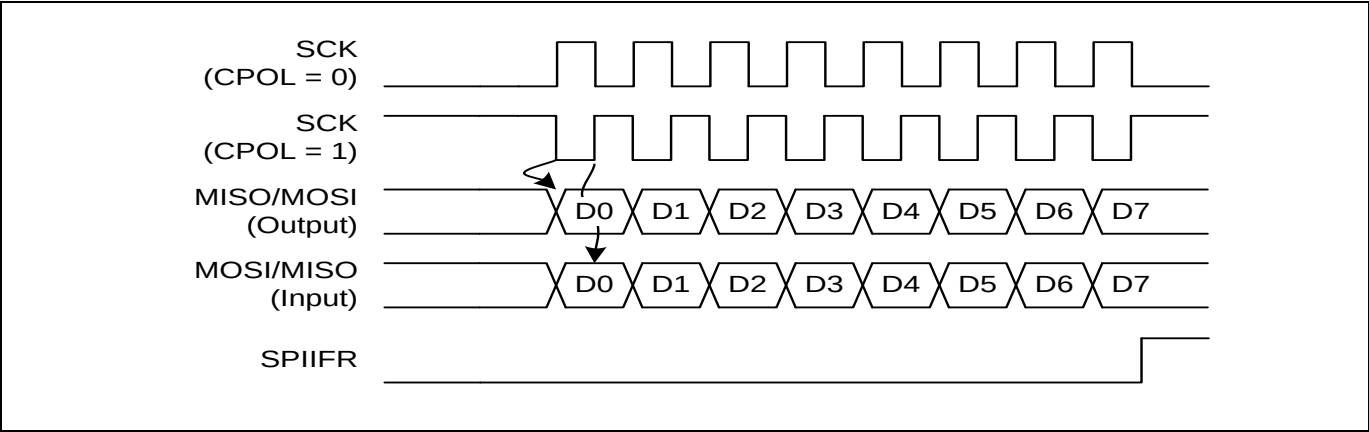


Figure 11.40 SPI发送/接收 时序图 CPHA = 1

11.10.5 寄存器

Name	Address	Direction	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

Table 11.14 SPI 寄存器

11.10.6 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

11.10.7 SPI 寄存器

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 Data
当写入一个字节数据到数据寄存器时, SPI 开始运行.

SPISR (SPI 状态寄存器) : B7H

7	6	5	4	3	2	1	0
SPIIFR	WCOL	—	—	FXCH	—	—	—
RW	R	—	—	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. 写入 “1” 无效.
- 0

SPI 中断没有发生
- 1

SPI 中断发生
- 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

No collision
- 1

Collision
- FXCH

SPI 口功能控制位
- 0

No effect
- 1

更改 MOSI 和 MISO 功能

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	该位控制 SPI 操作			
	0	禁止 SPI 运行		
	1	使能 SPI		
FLSB	该位选择数据传送顺序			
	0	MSB 高位优先		
	1	LSB 低位优先		
MS	选择主机或从机模式			
	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 frequency
	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

11.11 UART

11.11.1 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。

- 全双工操作 (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。

11.11.2 方框图

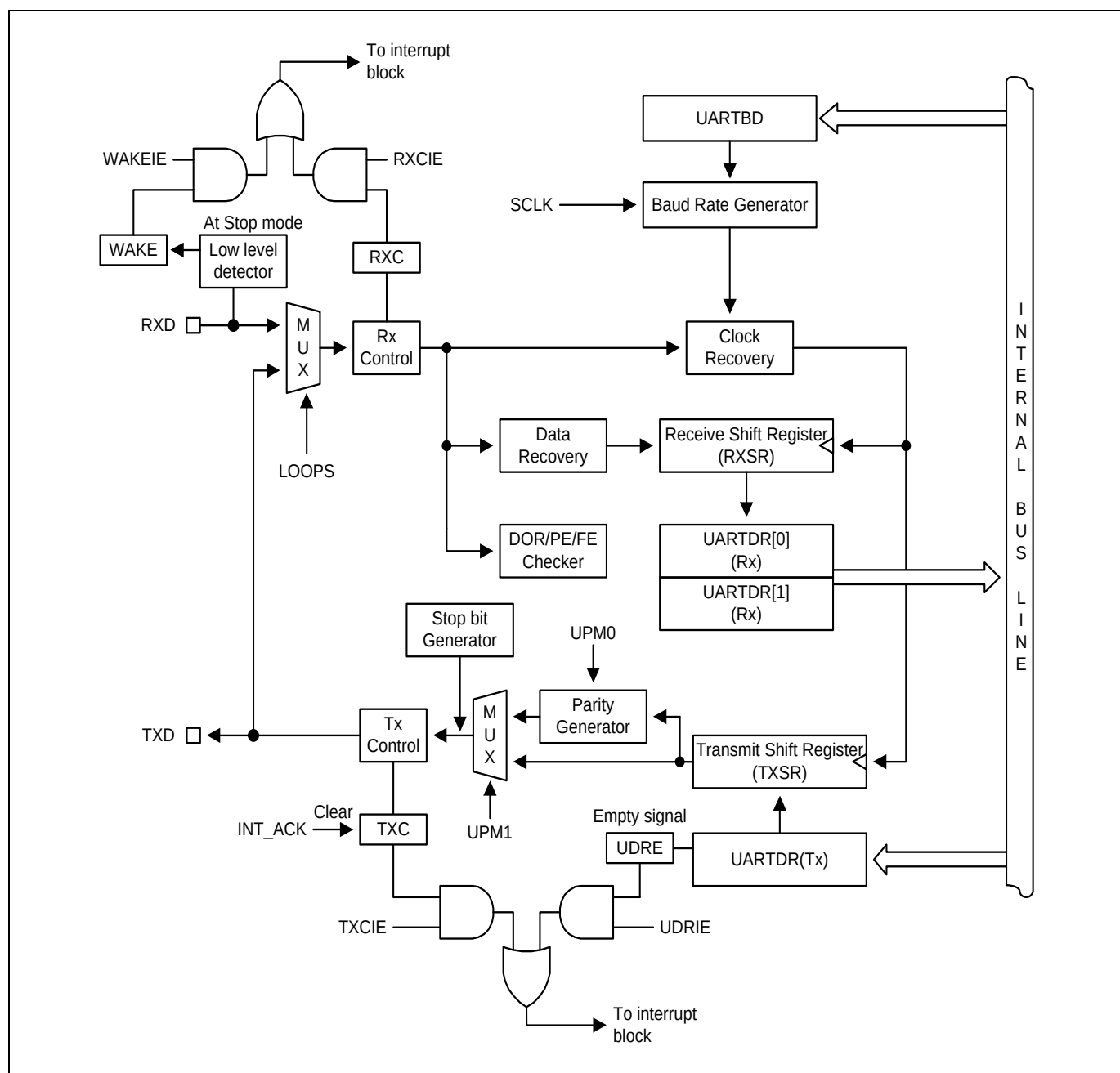


Figure 11.41 UART 方框图

11.11.3 时钟发生器

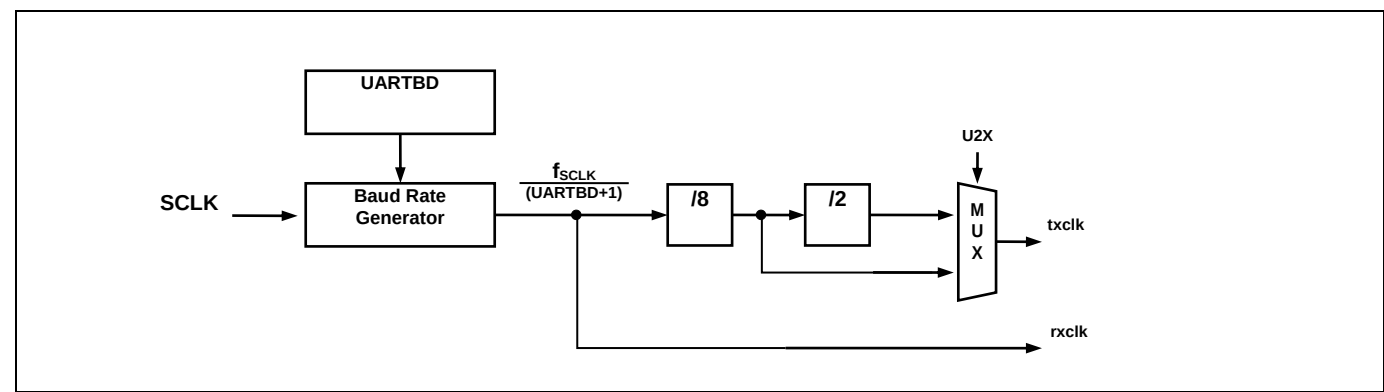


Figure 11.42 时钟发生器框图

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).

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

Table 11.15 波特率设置计算公式

11.11.4 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 数据位
- 无, 奇数或偶数校验位
- 1 或 2 stop 位

一帧数据开始于紧接着最低有效位的开始位(LSB). 然后是下一个数据位，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。

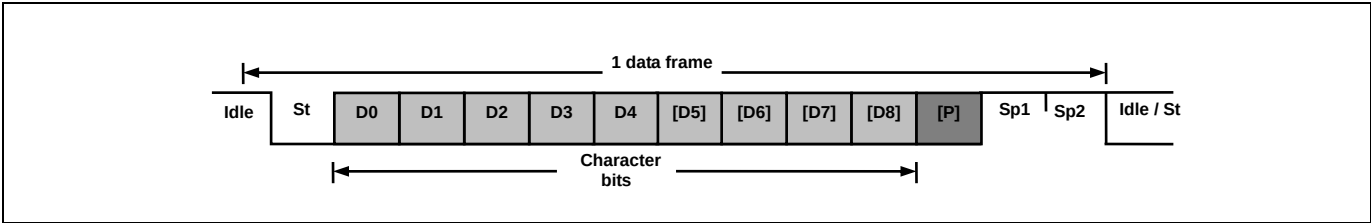


Figure 11.43 数据帧格式

1 帧数据包括以下位

- Idle 在通讯线上没有通讯(TxD/RxD)
- St Start bit (Low)
- Dn Data bits (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。

11.11.5 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.

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

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

P_{even} : 校验位使用偶校验

P_{odd} : 校验位采用奇校验

D_n : 数据位

11.11.6 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TX D 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.

11.11.6.1 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.

11.11.6.2 发送标志位和中断

UART发送器有2个标志位, 它表明了其状态. 一个是UART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 发送完成中断会产生中断.

11.11.6.3 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.

11.11.6.4 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(GP IO).

11.11.7 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。

11.11.7.1 接收 Rx 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读取数据之前读取。因为错误标志位是存储在接收暂存器中和FIFO相同的位置。

11.11.7.2 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’。

11.11.7.3 奇偶校验

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.

11.11.7.4 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)

11.11.7.5 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.

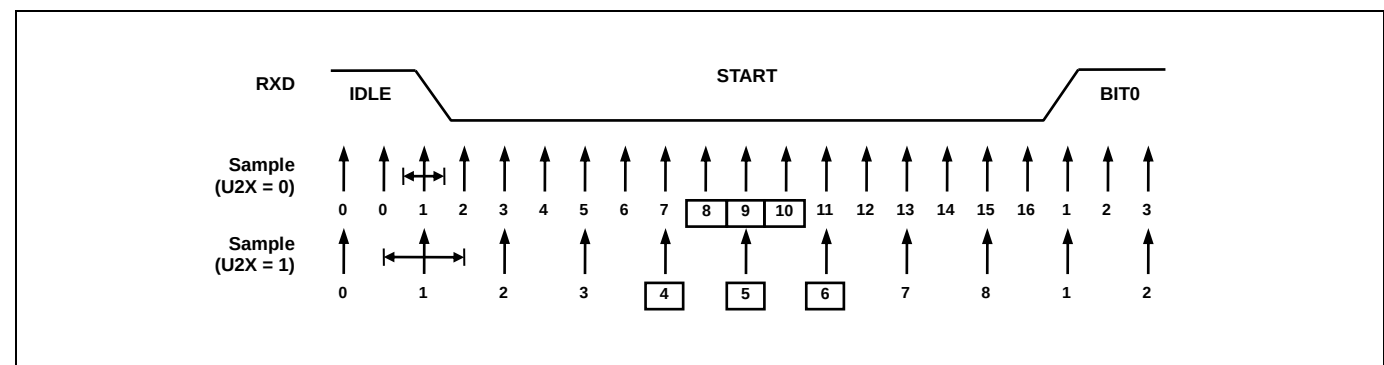


Figure 11.44 Start Bit 采样

当接收使能时(RXE=1), 时钟提取逻辑尝试在RXD上查找一个由高到低的过度变化, 这是开始位的条件。在RXD上检测到由高到低的变化时, 常规模式时钟提取逻辑采用采样的8, 9和10位, 双速模式时采样4,5和6位以用于检测是否接收到有效的开始位. 如果采样到多于两个的逻辑低电平, 就认为是有效地开始信号, 同时内部产生的时钟同步进入的数据。数据覆盖开始, 每个开始位同步过程都被重复。

综上所述, 当接收时钟和开始位同步时, 数据传送开始。数据获取的过程和时钟获取的过程是类似的. 常规模式时每个引入的位数据覆盖逻辑采样16次, 双速模式时采样8次。常规模式时采样的8,9,10决定数据值, 双速模式时采样4,5,6决定数据值。 如果采样到多于2个的低电平, 接收位认为是逻辑 ‘0’ 如果采样到多于2个的高电平, 则认为是逻辑‘1’。数据覆盖过程一直被重复直到包含一个停止位的完整的数据接收完成. 决定位的值按顺序存储到接收移位寄存器. 注意接收器只使用第一个停止位。在内部接收到第一个停止位后, 接收器进入空闲状态等待开始位。

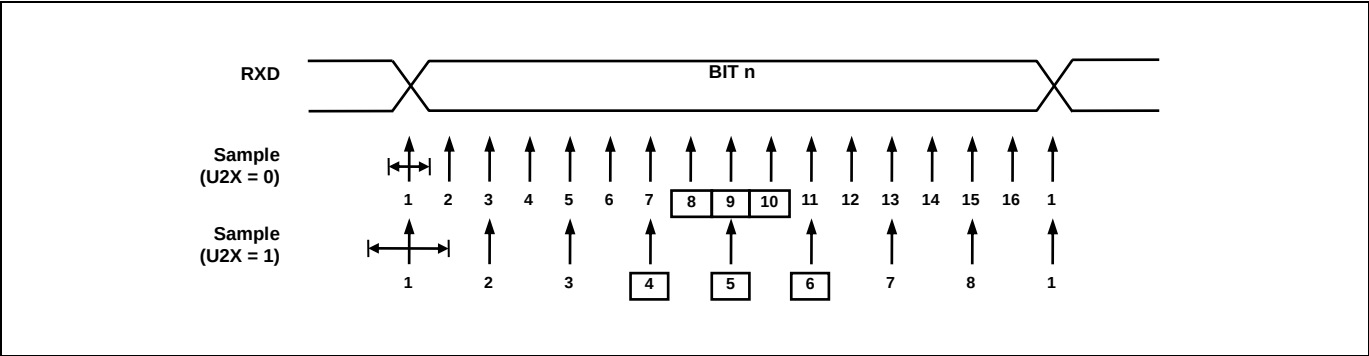


Figure 11.45 数据采样和校验

检测停止位的过程和时钟、数据的获取过程类似。也就是, 如果采样到多于2个或3个中间值是高电平, 就认为检测到停止位, 否则, FE标志位置位。 在确定第一个停止位有效后, 接收器进入空闲模式等待下一个开始位。

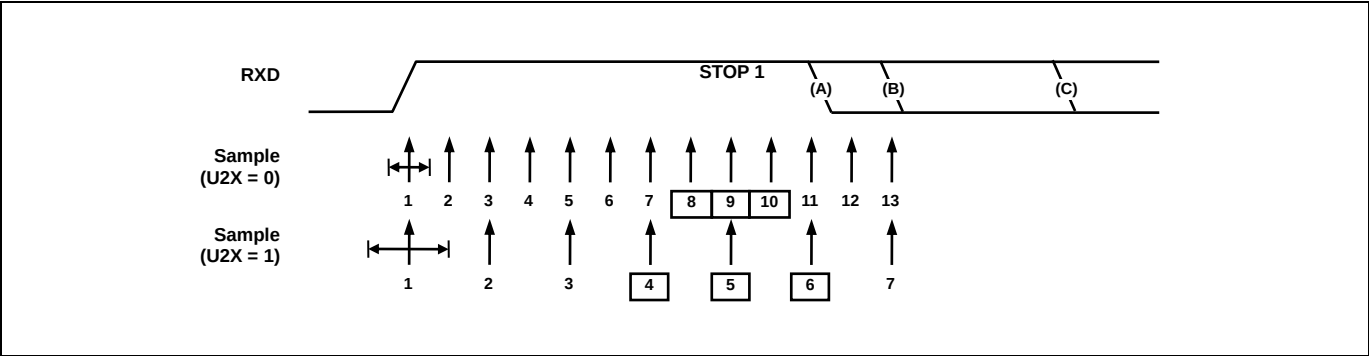


Figure 11.46 Stop 位采样和下个开始位采样

11.11.8 寄存器

Name	Address	Direction	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

Table 11.16 UART 寄存器

11.11.9 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).

11.11.10 UART 寄存器

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] 发送缓存器和接收缓存器和数据寄存器共用相同的I/O地址, 发送数据缓存器是写入到UARTDR 寄存器的目标. 读UARTDR寄存器时从接收缓存器返回内容.
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	Parity
0	0	无校验
0	1	Reserved
1	0	偶同位校验
1	1	奇同位校验

USIZE[2:0] 选择帧数据长度

USIZE2	USIZE1	USIZE0	Data Length
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 (use polling)
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(use polling)
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 (use polling)
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	禁止接收
1	使能接收
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 使能
U2X	选择接收采样率.
0	常规异步操作
1	倍速异步操作

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 循环模式
- USBS

选择停止位长度.

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 (9th bit) to be transmitted is '0'

1 MSB (9th 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 (9th bit) received is '0'

1 MSB (9th bit) received is '1'

UARTST (UART 状态寄存器) : E5H

7	6	5	4	3	2	1	0
UDRE	TXC	RXC	WAKE	SOFTIRST	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。
SOFTIRST	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 Reset 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 没有奇偶校验错误 1 检测到奇偶校验错误

11.11.11 波特率设置 (example)

Baud Rate (bps)	fx=1.00MHz		fx=1.8432MHz		fx=2.00MHz	
	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	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	-	-	-	-	-	-

Table 11.17 USI0BD 和 USI1BD常用波特率设置举例

Baud Rate (bps)	fx=1.00MHz		fx=1.8432MHz		fx=2.00MHz	
	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	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 (bps)	fx=8.00MHz		fx=11.0592MHz	
	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	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	-	-	-	-
----	---	---	---	---

(Concluded)

11.12 I2C

11.12.1 概述

I2C 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 I2C 总线标准
- 多主控操作
- 高达400kHz 数据传输速率
- 7 bit 地址
- 支持两个从动地址
- 主控和从动操作
- 总线忙时检测

11.12.2 方框图

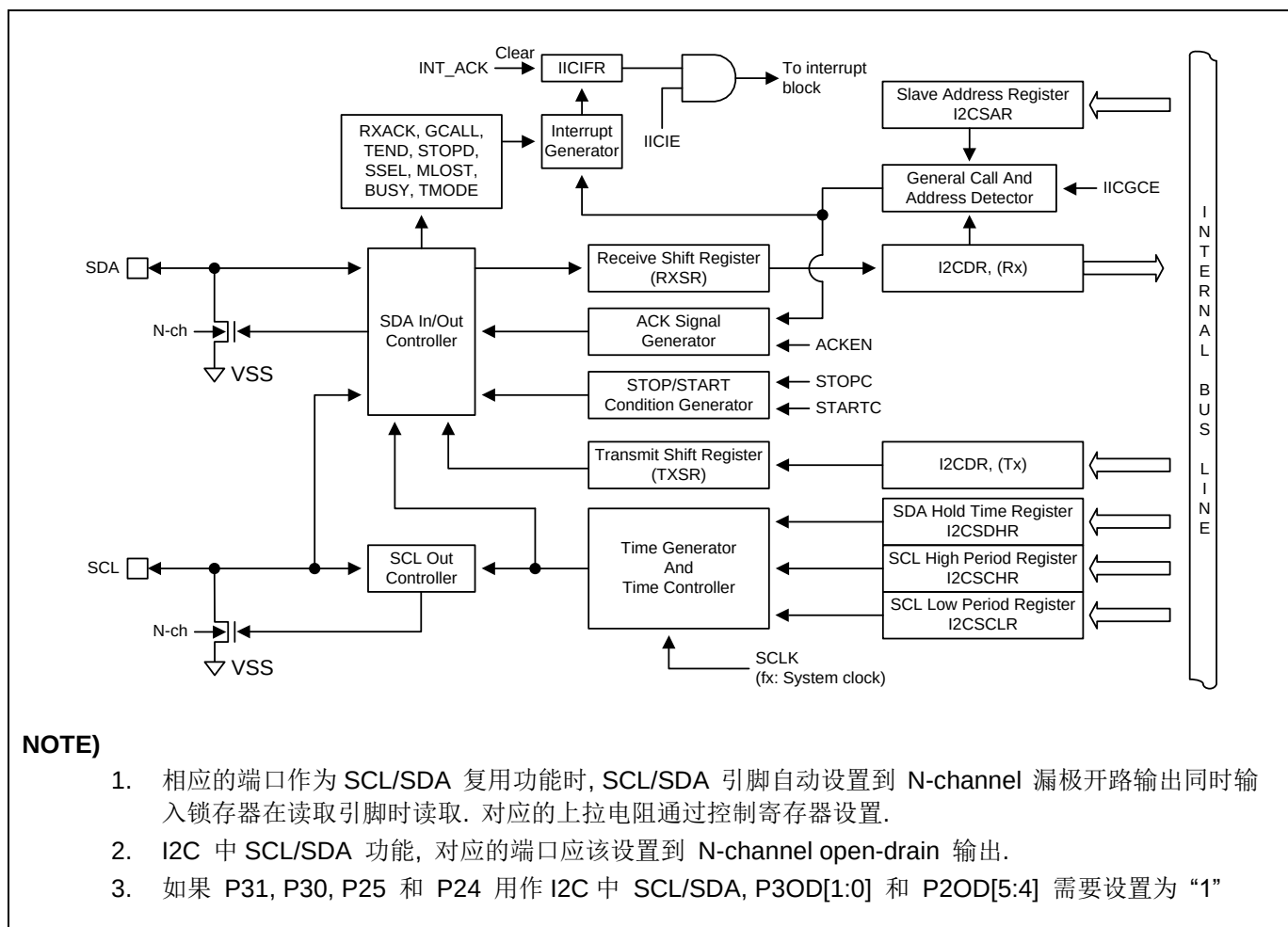


Figure 11.47 I2C 框图

11.12.3 I2C 位发送

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。

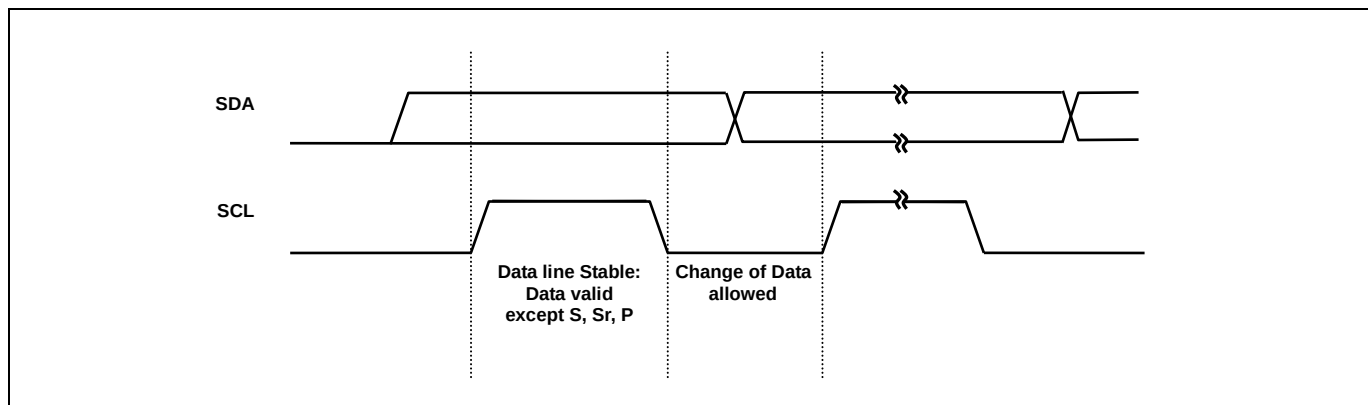


Figure 11.48 I2C-Bus 位传输

11.12.4 开始 / 重复开始 / 停止

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.

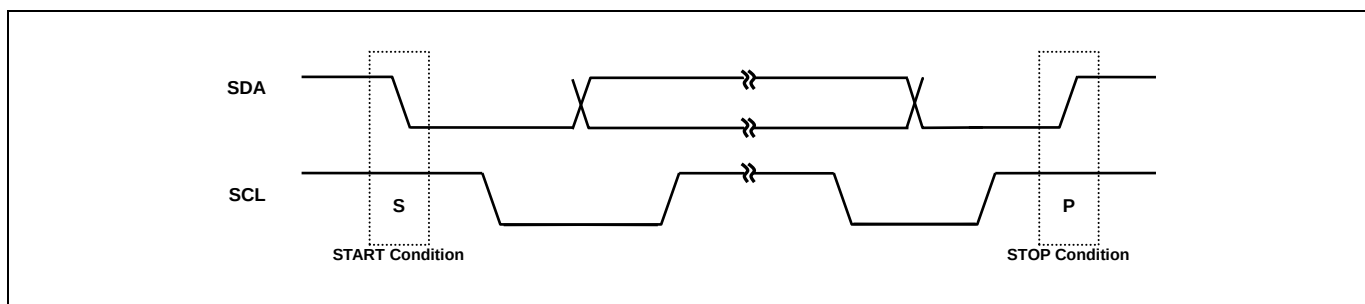


Figure 11.49 START 和 STOP 信号

11.12.5 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, 数据传送可以继续.

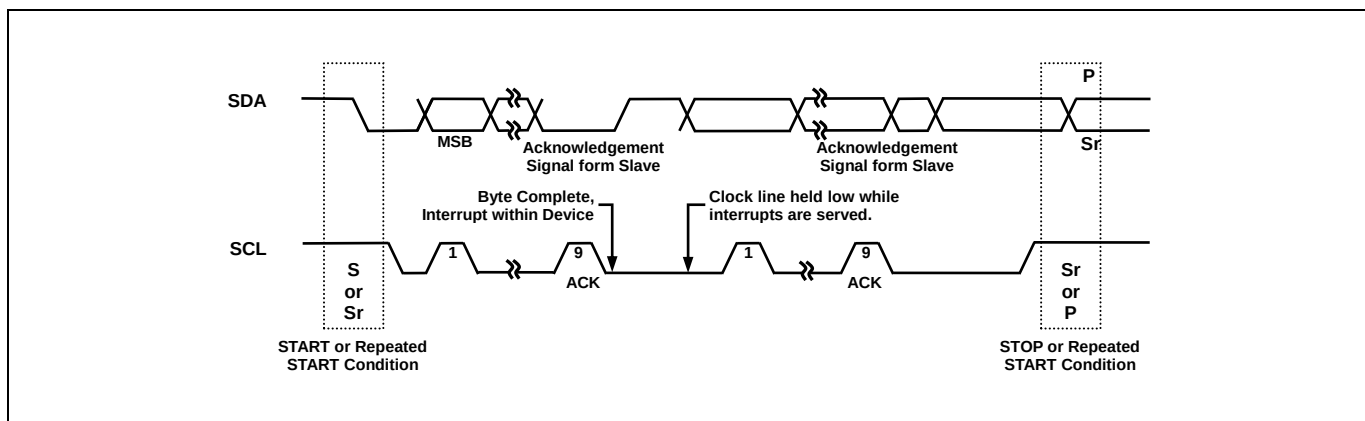


Figure 11.50 I2C-Bus 数据传送

11.12.6 I2C 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。

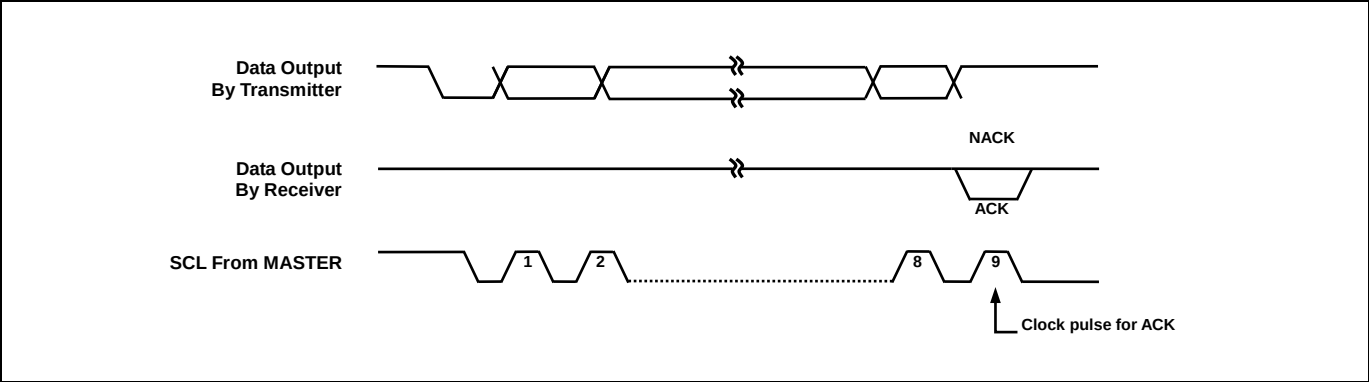


Figure 11.51 I2C-Bus 应答

11.12.7 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。两个或更多个主机可以产生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。

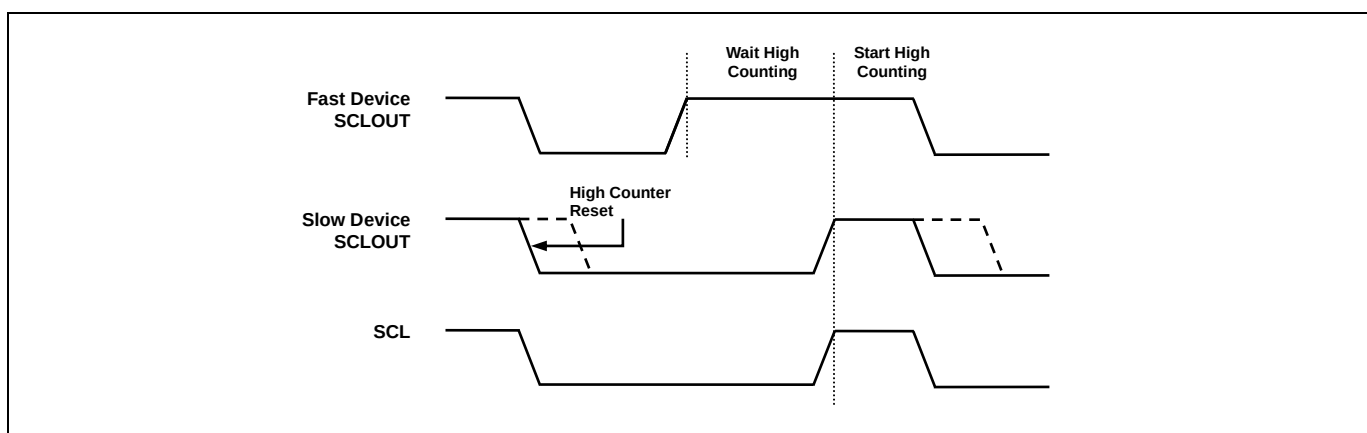


Figure 11.52 仲裁过程中的时钟同步

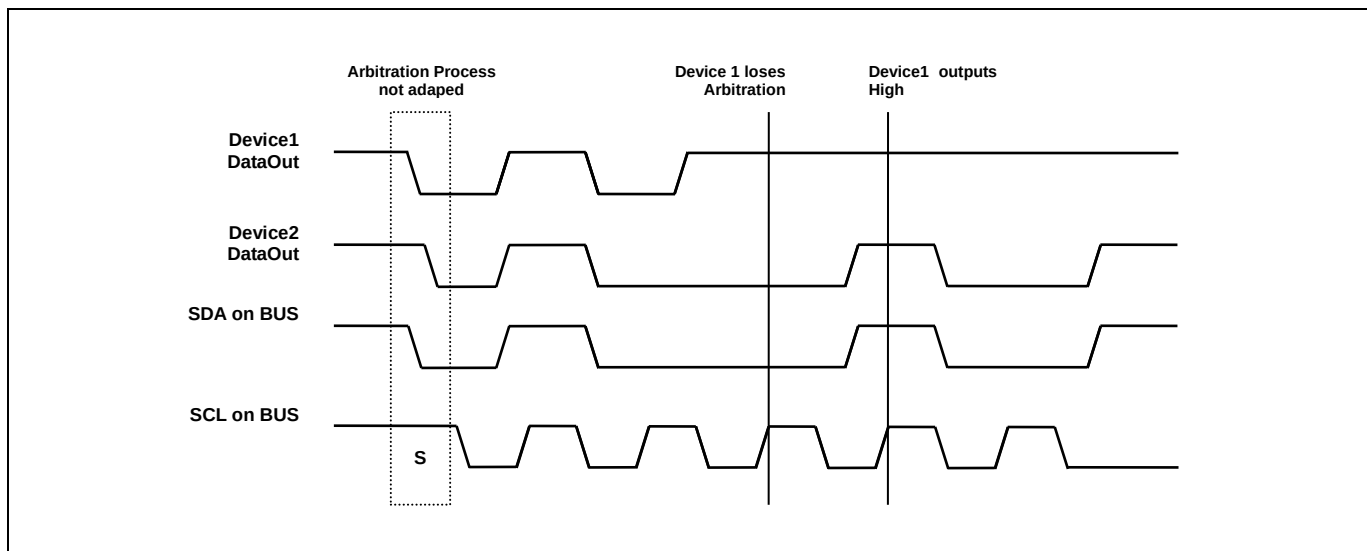


Figure 11.53 双主机仲裁过程

11.12.8 操作

I2C 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以后中断被发出。由于I2C基于中断，在I2C字节传送时，应用软件可以任意进行其他的操作。

注意当I2C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 I2C 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I2C当前状态的值。根据这个值，软件可以决定下一步该做什么。

I2C 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。

11.12.8.1 主机发送

I2C 作为主机发送时，以下是推荐的步骤。

1. 设置I2CCR的IICEN位使能 I2C。这里向外围提供主时钟。
2. 加载 SLA+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，I2C 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 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+RW 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. 这是主控制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制器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TE ND 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+RW 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。

11.12.8.2 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. 这是主控制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， I2C 产生 TEND 中断..

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+RW 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.

11.12.8.3 I2C 从机接收

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.

11.12.8.4 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.

11.12.9 寄存器

Name	Address	Direction	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

Table 11.18 I2C 寄存器

11.12.10 I2C 寄存器说明

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

11.12.11 I2C 寄存器

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 维持时间寄存器) : 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 $t_{SCLK} \times (I2CSDHR+2)$ 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 $t_{sclk} \times (I2CSDHR+2)$ 后开始改变。所以，确保在从动时的操作和数值
 $t_{SCLK} \times (I2CSDHR + 2)$ 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： $t_{SCLK} \times (4 \times I2CSCHR + 2)$ 这里
 t_{SCLK} 是SCLK的周期

I2C 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： $t_{SCLK} \times (4 \times I2CSCLR + 2)$
 where
 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	GCALL0EN
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.**GCALL0EN** 该位决定 在从机模式时I2C 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I2C 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

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 No effect

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)

NOTE) 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 是从机模式

1 I2C 是主机模式

STOPC I2C 是主控制器时, STOP 信号的产生

0 No effect

1 产生STOP 信号

STARTC I2C 是主控制器时, START信号的产生

0 No effect

1 产生START 或重复START 信号

NOTE)

1. 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 (NOTE) 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(NOTE) 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(NOTE) 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 (NOTE) I2C 被其他主机编址时置1.

0 I2C 没有被作为从机选定

1 I2C 被其他主机编址充当从机

MLOST (NOTE) 该位代表主控模式中总线仲裁结果.

0 I2C 保持总线控制权

1 I2C 已经丢失总线控制权

BUSY 该位标志总线状态.

0 I2C 总线空闲, 主机可以发送一个START 信号

1 I2C 总线繁忙

TMODE 该位表示I2C是发送还是接收.

0 I2C 接收

1 I2C 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

NOTE)

1. 这些位可作为中断源.
2. 当 I2C 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.

12 省电模式操作

12.1 概述

MC96F8208S有两个省电模式减少设备电量消耗. 在省电模式中，电量消耗减少很多。设备提供3种节电功能Main-IDLE, Sub-IDLE 和 STOP模式. 在3中模式下，程序停止.

12.2 IDLE/STOP 模式下外围的操作

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
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 f _{IRC}
WDTRC OSC (5kHz)	Can be operated with setting value	Can be operated with setting value
Main OSC (0.4~12MHz)	Oscillation	Stop when fx = f _{XIN}
I/O Port	Retain	Retain
Control Register	Retain	Retain
Address Data Bus	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET, Timer Interrupt (EC1), SPI (External clock), External Interrupt, UART by RX, WT (main clock), WDT

Table 12.1 睡眠模式下外围操作

12.3 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。

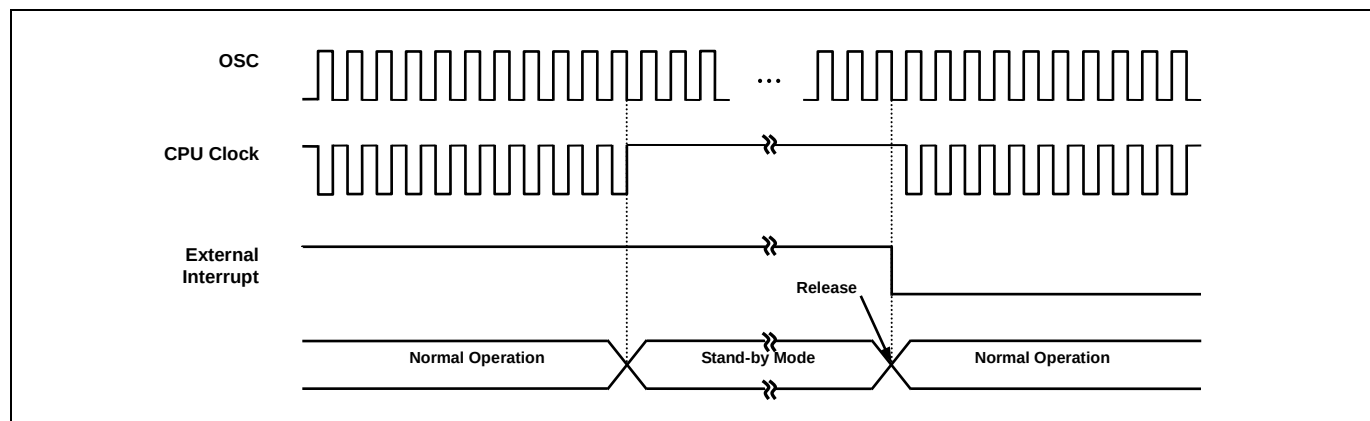


Figure 12 外部中断唤醒 IDLE 模式时序

12.4 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 (f_{IRC}) 被选做系统时钟并且sub clock (f_{SUB})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. 图 12.2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.

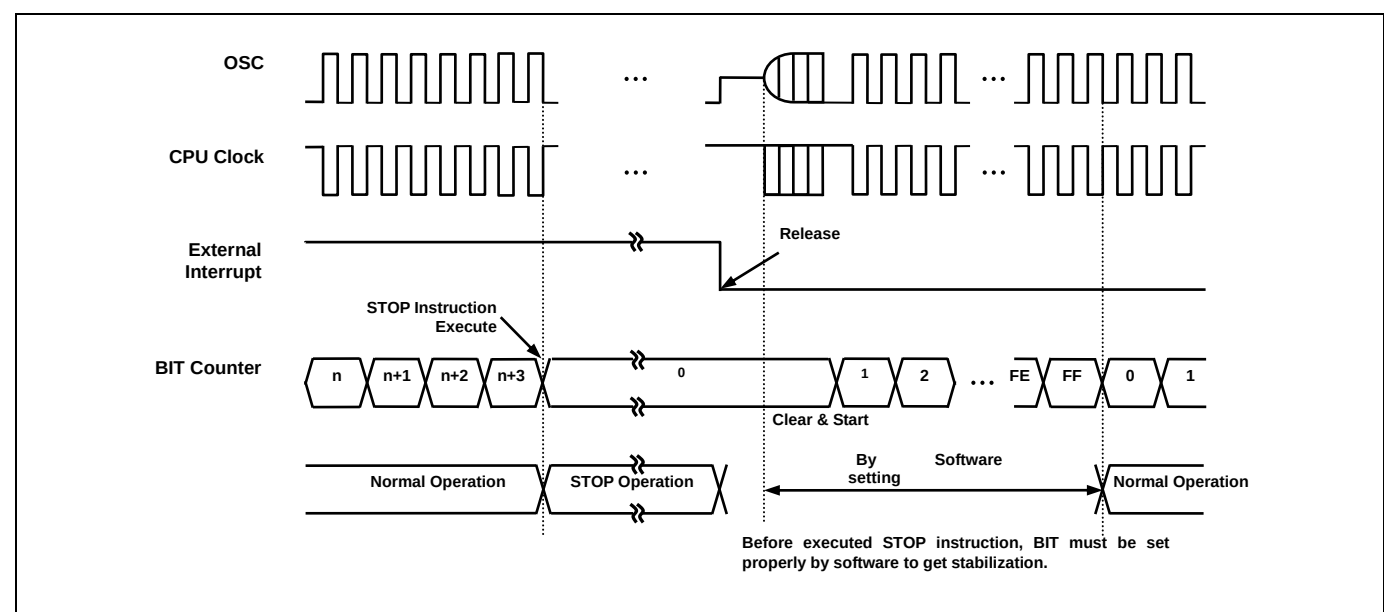


Figure 12.2 外部中断唤醒 STOP 模式时序

12.5 STOP 模式唤醒

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 (图 12.3)。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`。

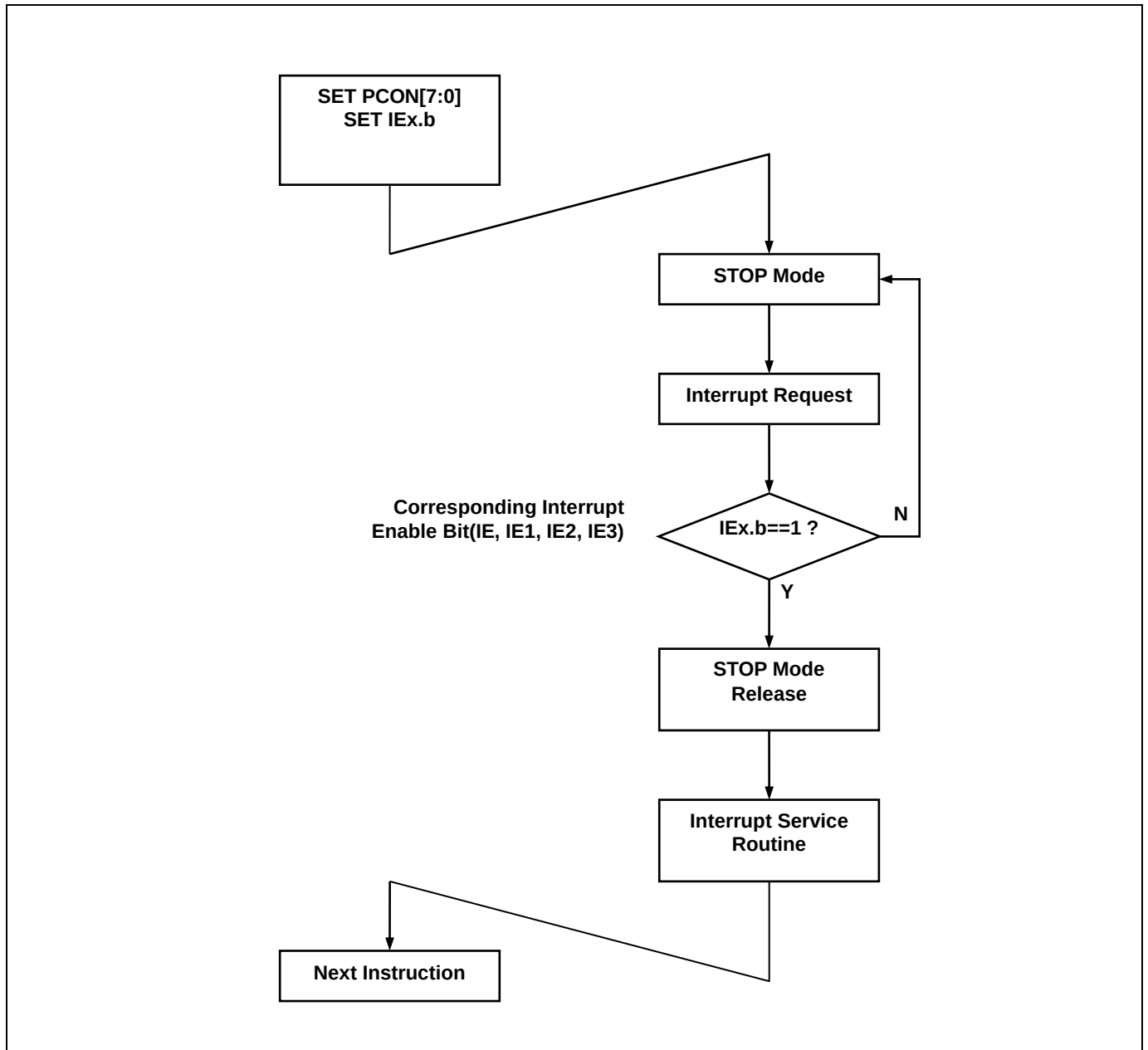


Figure 12.3 STOP 模式唤醒流程

12.6 寄存器

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

Table 12.2 Power Down Operation Register Map

12.7 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).

12.8 省电操作寄存器

PCON (Power Control Register): 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	普通模式

NOTE)

- 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.

Ex1)

MOV PCON, #01H ; IDLE mode

NOP

NOP

NOP

•

•

•

Ex2)

MOV PCON, #03H ; STOP mode

NOP

NOP

NOP

•

•

•

13 复位

13.1 概述

以下是硬件设置值.

On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Refer to the Peripheral Registers

Table 13.1 Reset State

13.2 复位源

MC96F8208S有5种类型的复位源. 以下是复位源.

- External RESETB
- Power ON RESET (POR)
- WDT Overflow Reset (In the case of WDTEN = `1`)
- Low Voltage Reset (In the case of LVREN = `0`)

13.3 复位方框图

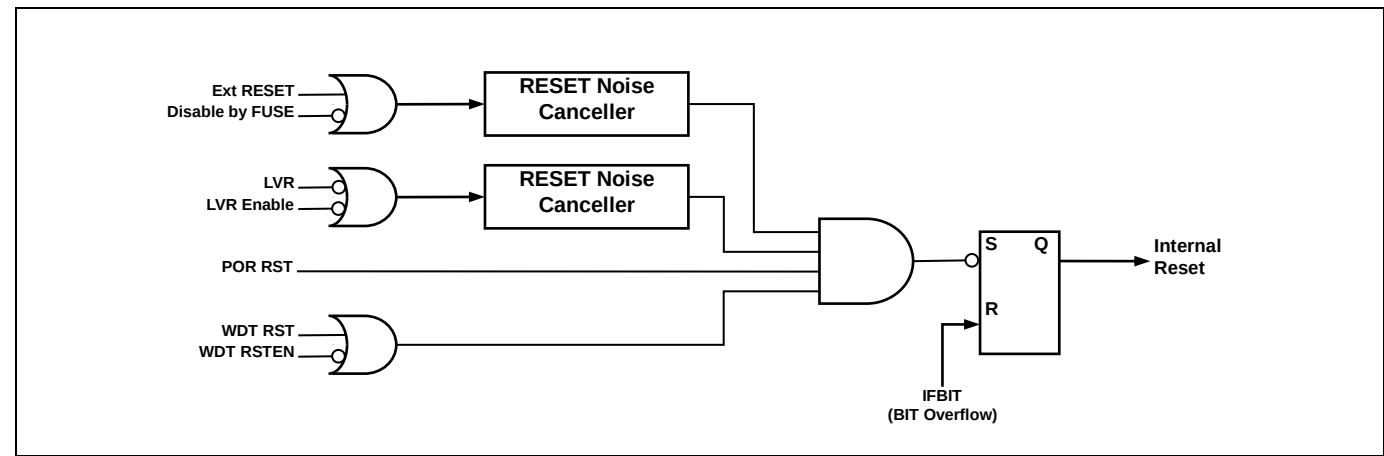


Figure 13.1 复位框图

13.4 RESET 噪音消除器

图 13.2 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.

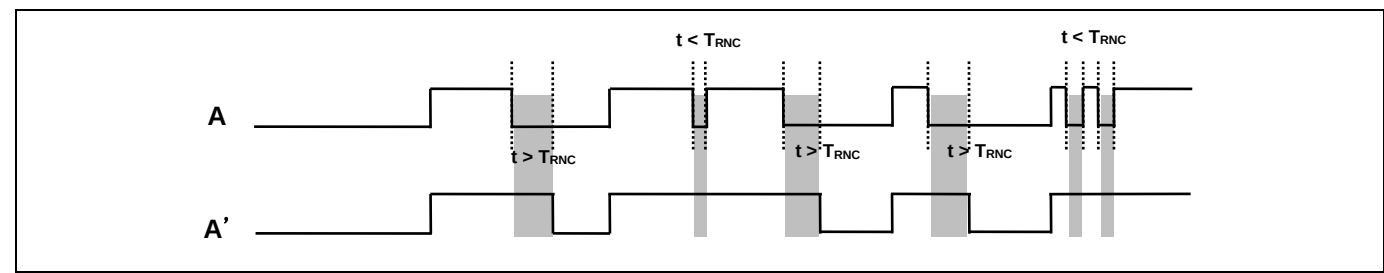


Figure 13.2 复位噪音消除器时序图

13.5 Power on RESET

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.

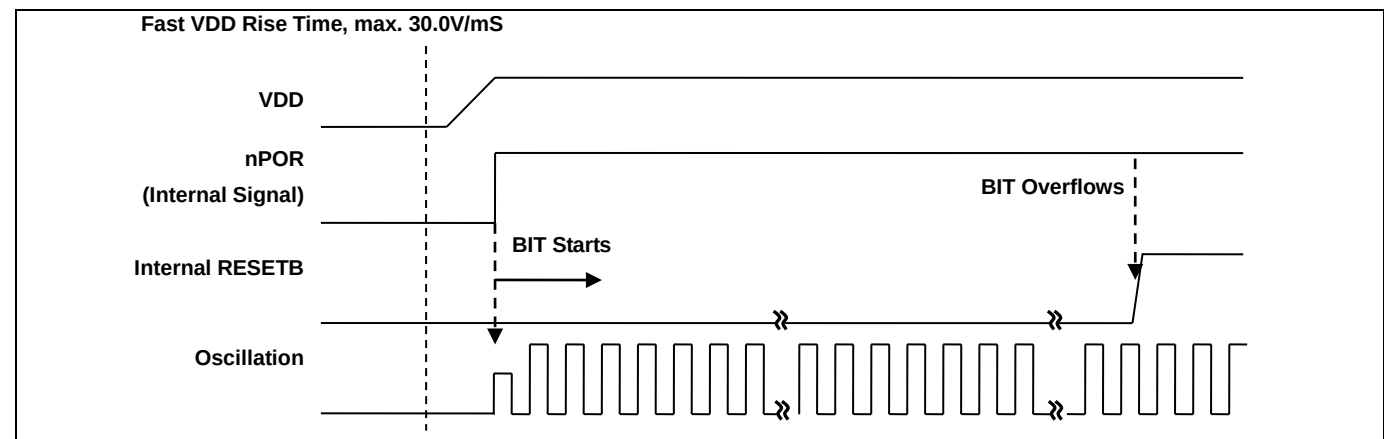


Figure 13.3 快速 VDD 上升时序

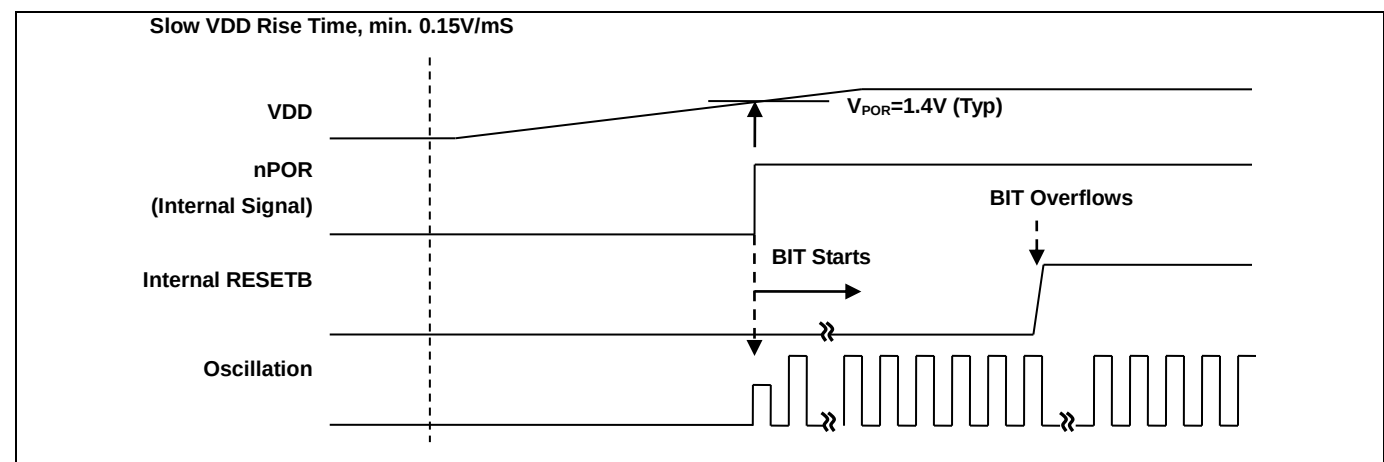


Figure 13.4 电压升高内部复位唤醒时序

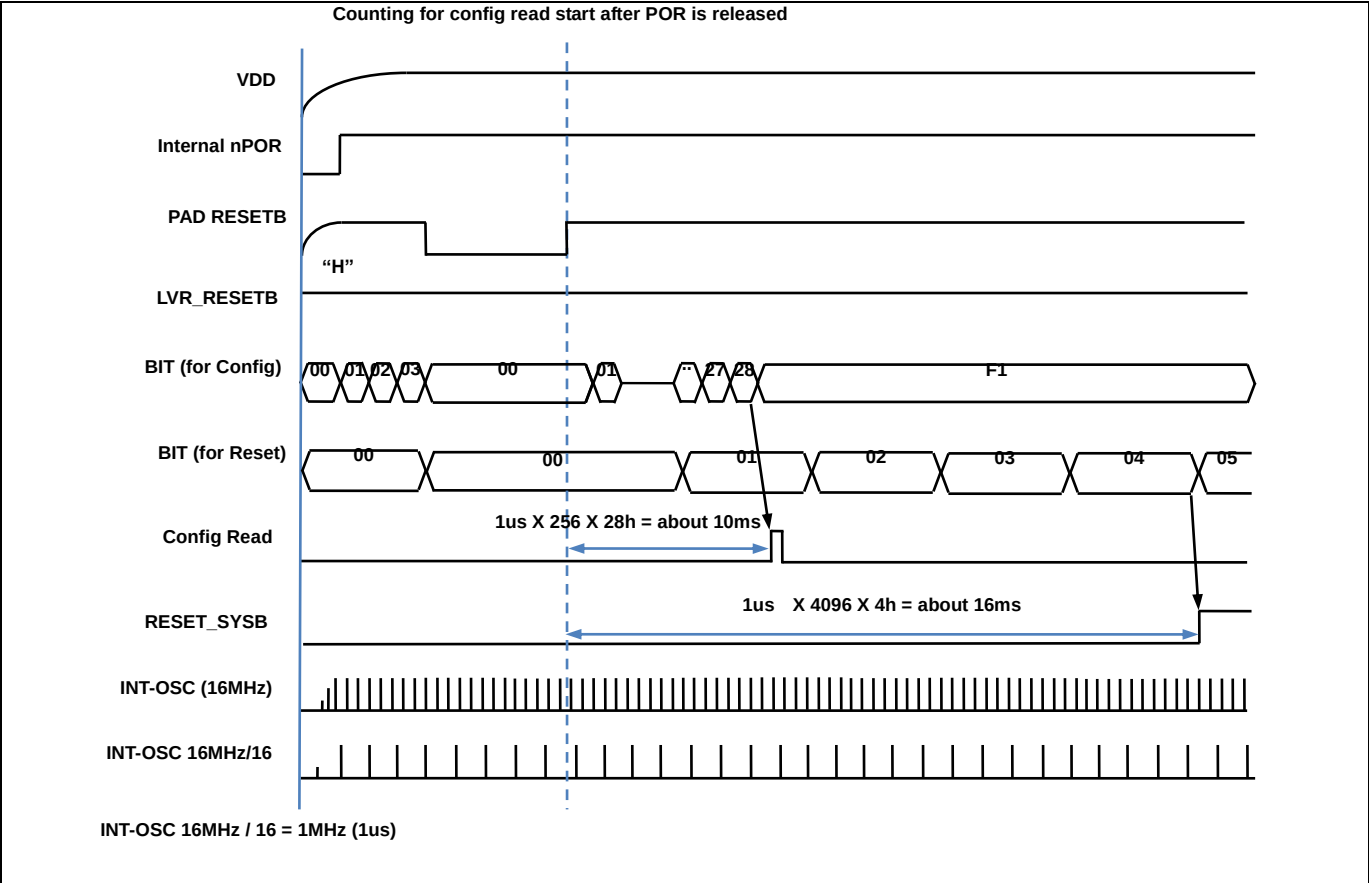


Figure 13.5 上电组态时序

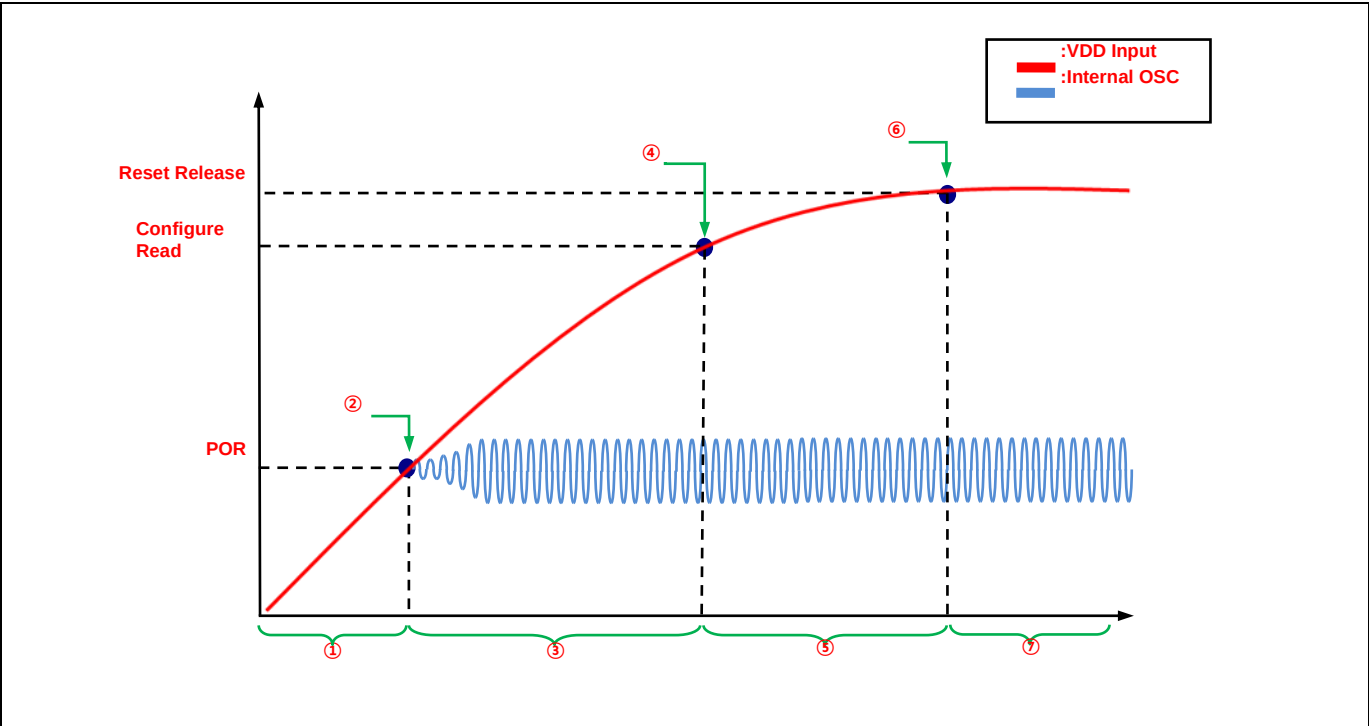


Figure 13.6 启动过程波形图

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	

Table 13.2 Boot Process Description

13.6 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.

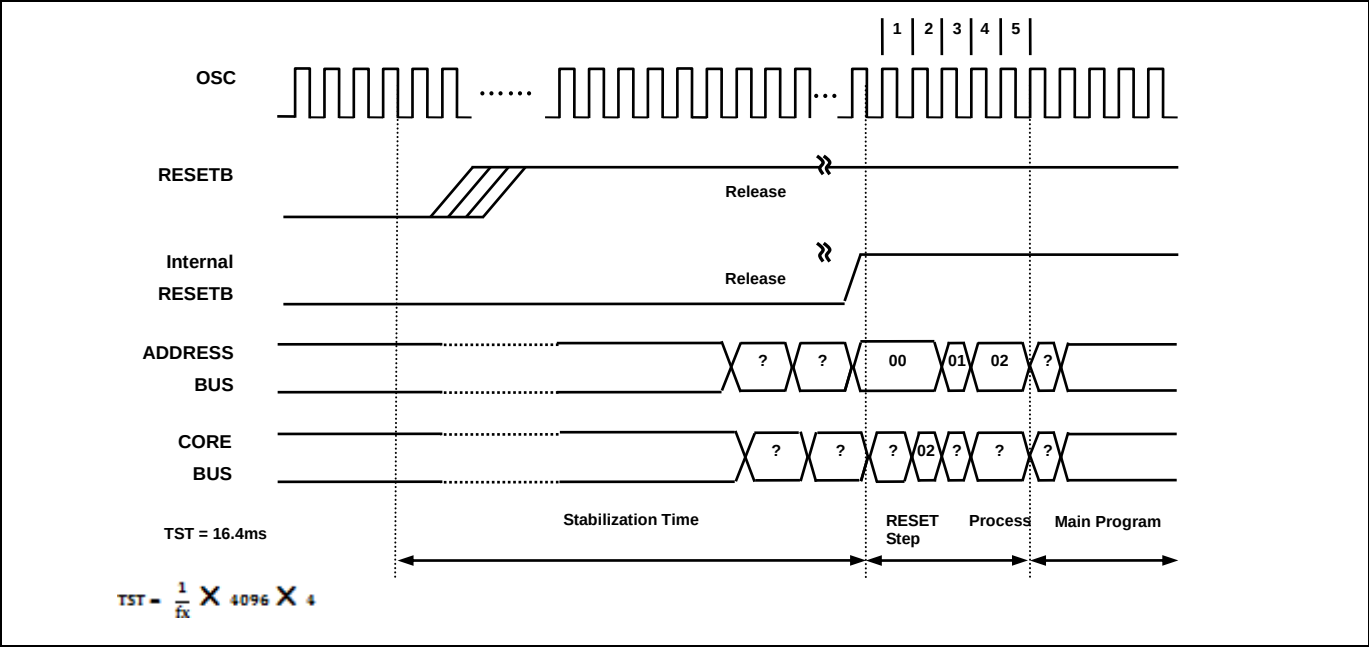


Figure 13.7 RESET后时序图

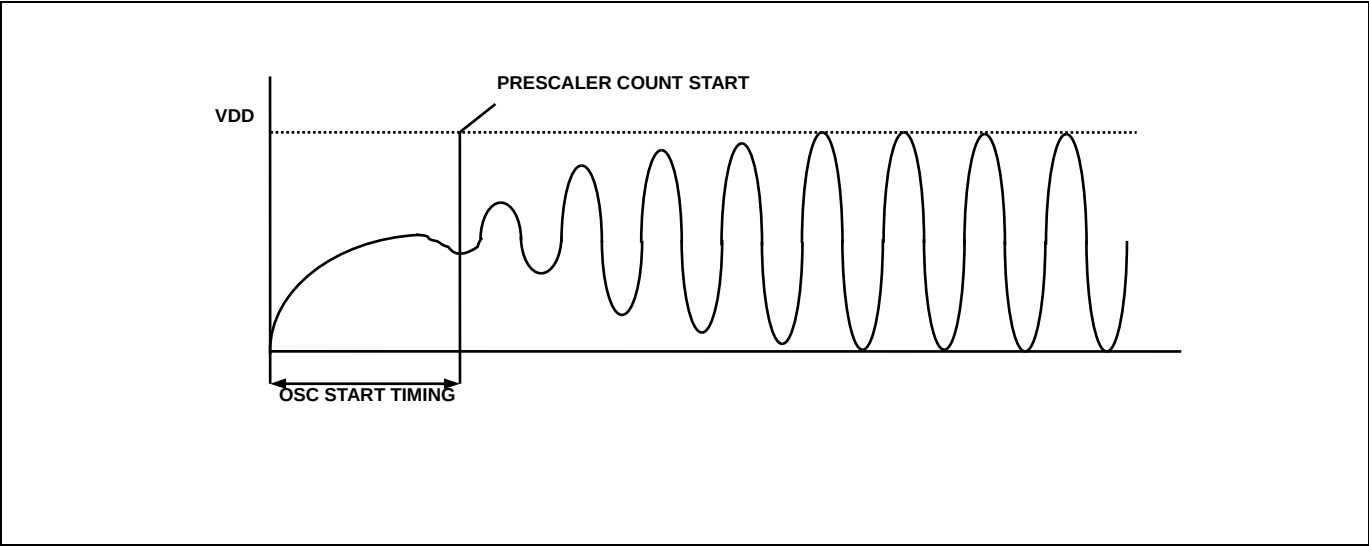


Figure 13.8 振荡器产生波形举例

NOTE)

- 1. 如图 13.8 所示, 稳定时间不包括启动时间.
- 2. RESETB 引脚有上拉寄存器

13.7 BOD处理器

MC96F8208S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 位被软件设定在关闭状态.

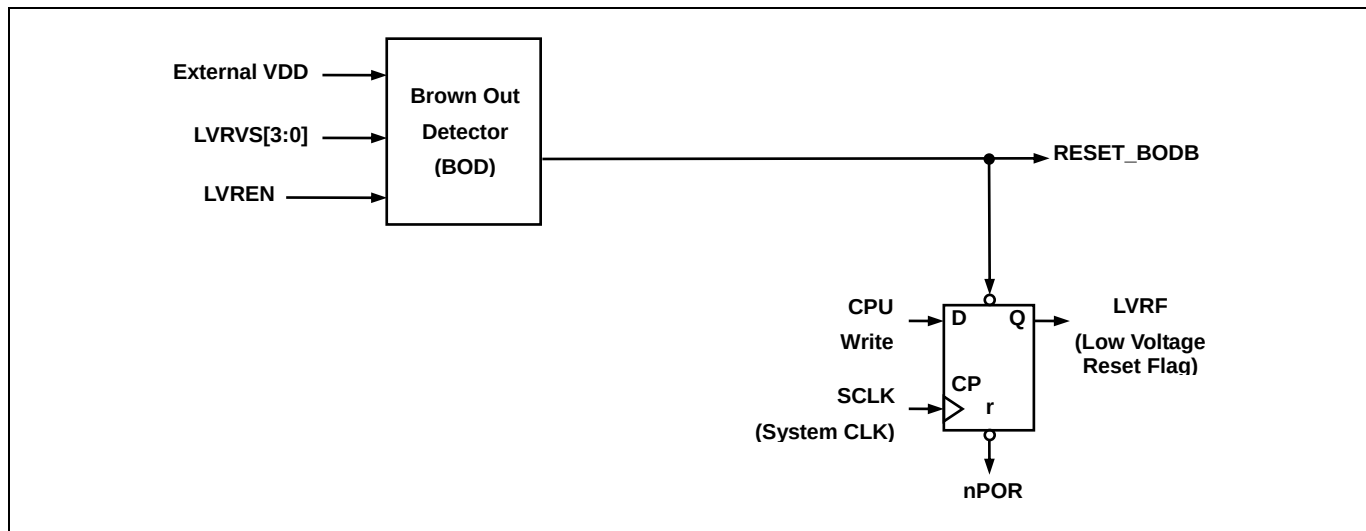


Figure 13.9 BOD方框图

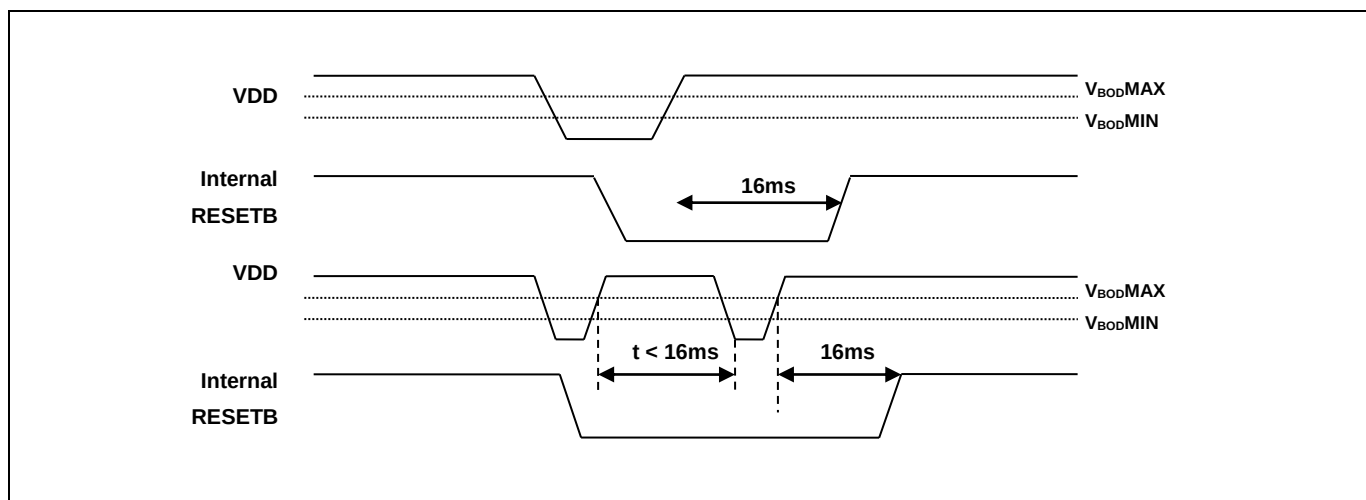


Figure 13.10 电源故障时内部复位

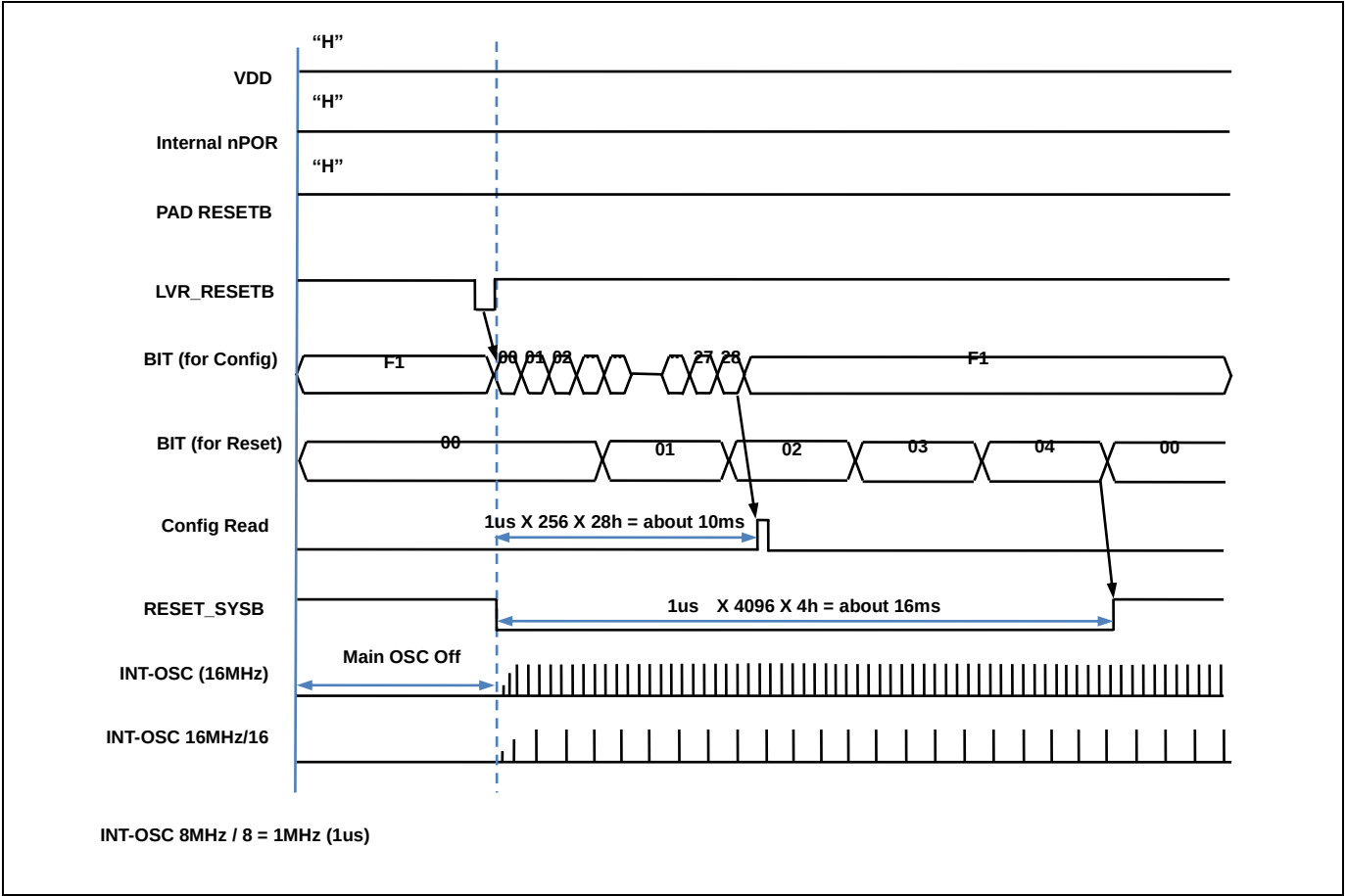


Figure 13.11 BOD RESET 时配置时序

13.8 LVI 方框图

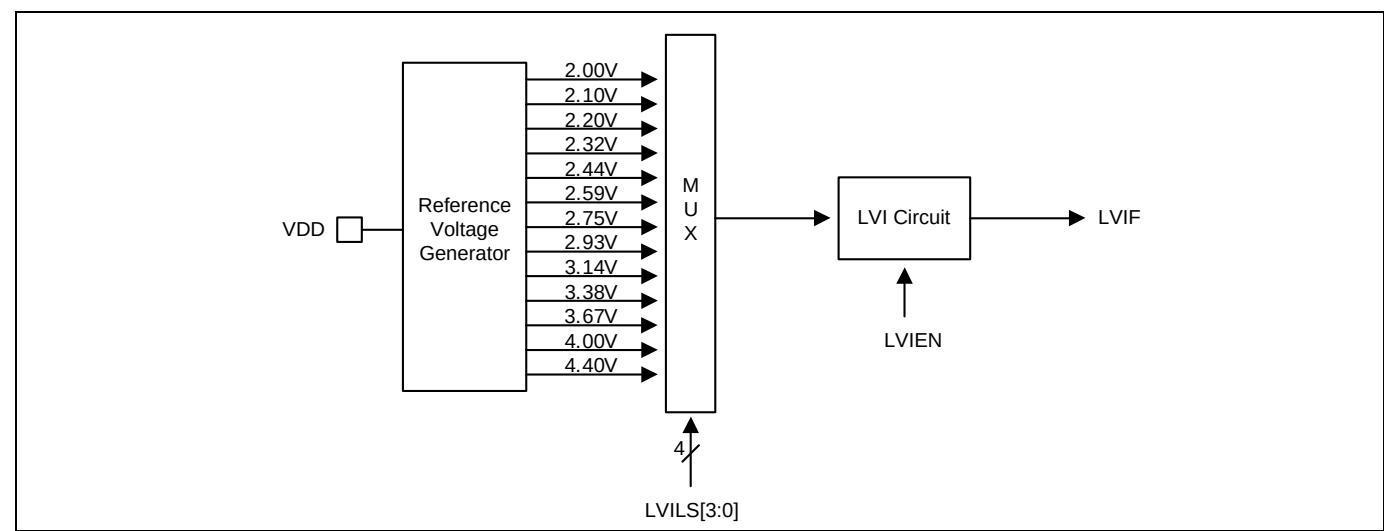


Figure 13.12 LVI 框图

13.9 寄存器图

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

Table 13.3 复位操作寄存器

13.10 复位操作寄存器说明

复位控制寄存器由(RSTFR), (LVRCCR)和 (LVICR)组成.

13.11 复位操作寄存器

RSTFR (复位标志位寄存器) : E8H

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

Initial value : 80H

- PORF

上电复位标志位. 写入 ‘0’ 清除该位

0 没有检测到

1 检测到
- 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 没有检测到

1 检测到
- WDTRF

看门狗复位标志位, 该位通过写入0或上电复位清除.

0 没有检测到

1 检测到
- LVRF

低电压复位标志位. 该位通过写入0或上电复位清除

0 没有检测到

1 检测到

NOTE)

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.

LVRCCR (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

NOTE)

1. 当该位是‘1’, LVREN 位被退出睡眠信号清零 (LVR 使能)
2. 当该位是 ‘0’, LVREN 位对唤醒睡眠信号没反应

LVRVS[3:0] LVR 电平选择

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

Other Values Not available

LVREN LVR 操作
 0 LVR 使能
 1 LVR 禁止

NOTE)

1. LVRST, 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
—	—	LVIF	LVIEN	LVILS3	LVILS2	LVILS1	LVILS0
—	—	RW	RW	RW	RW	RW	RW

Initial value : 00H

LVIF 低电压指示标志位

0 No detection

1 Detection

LVIEN LVI 使能/禁止

0 Disable

1 Enable

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
---	---	---	---	-------

Other Values				Not available
--------------	--	--	--	---------------

14 片内调试系统(MC96F8208 EVA ONLY)

14.1 概述

14.1.1 说明

MC96F8208S不能使用片内调试系统(OCD). MC96F8208S没有该功能. 我们推荐使用MC96F8208 EVA(MC96F8208 OCD / MC96F8208OB / MC96F8208OU) 进行开发和调试. MC96F8208 EVA的调试工具可以烧写程序和片内调试. 详细说明请参考下文.

图 14.1 所示为OCD接口和片内调试系统的方框图.

14.1.2 特性

- 两根外部接口线: 一根是时钟输入, 一根是串行数据总线
- 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 Data EEPROM
- 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
 - On-chip Debugging 支持 Dr.Choice®
- 工作频率
 - 支持目标 MCU 的最大频率

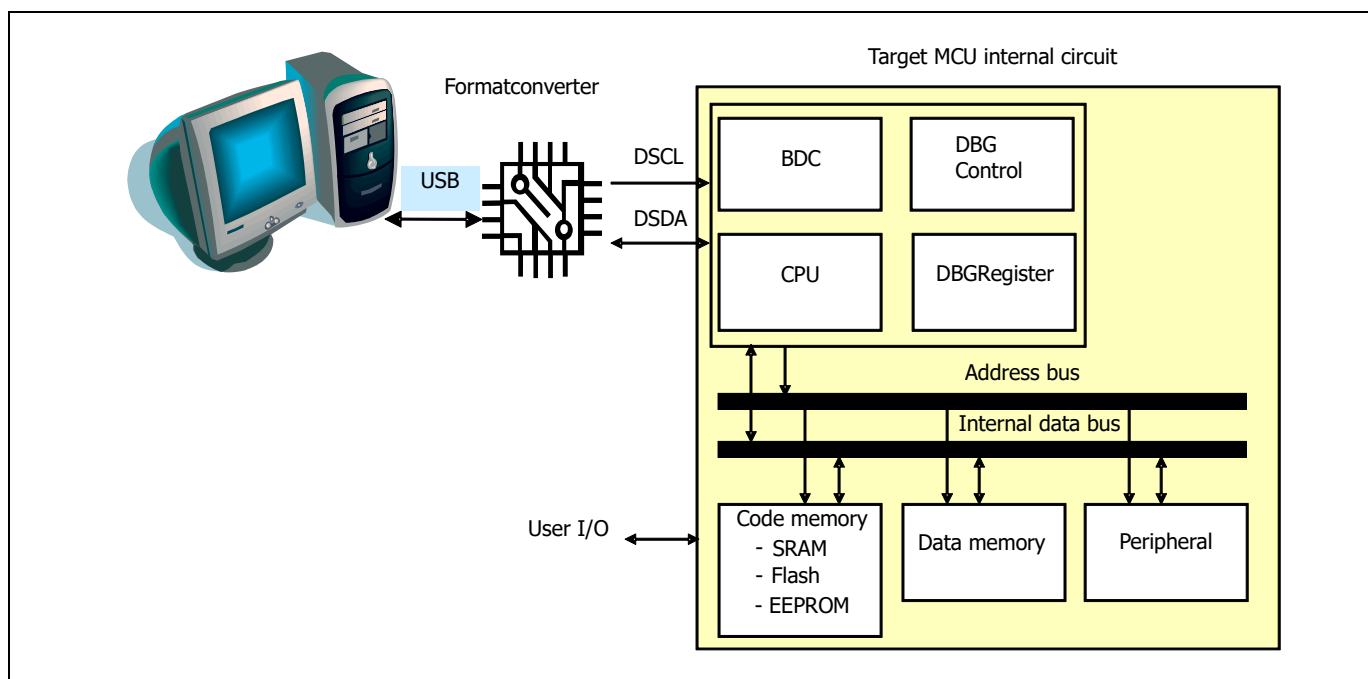


Figure 14.1 OCD框图

14.2 双线外围接口

14.2.1 基本传输包

- 10-bit 传输包用于双线接口.
- 1个传输包包括 8位数据, 1位校验和1位应答.
- 8位数据传送时校验位是1.
- 8位数据传送和校验位无误时接收器产生应答 '0'.
- 发送器没有确认(应答位是'1'), 错误处理在发生器中执行.
- 产生应答错误时, 主机产生停止命令.
- 后台调试器由一系列数据包组成.
- Start 和stop 命令通知后台调试器开始或停止.

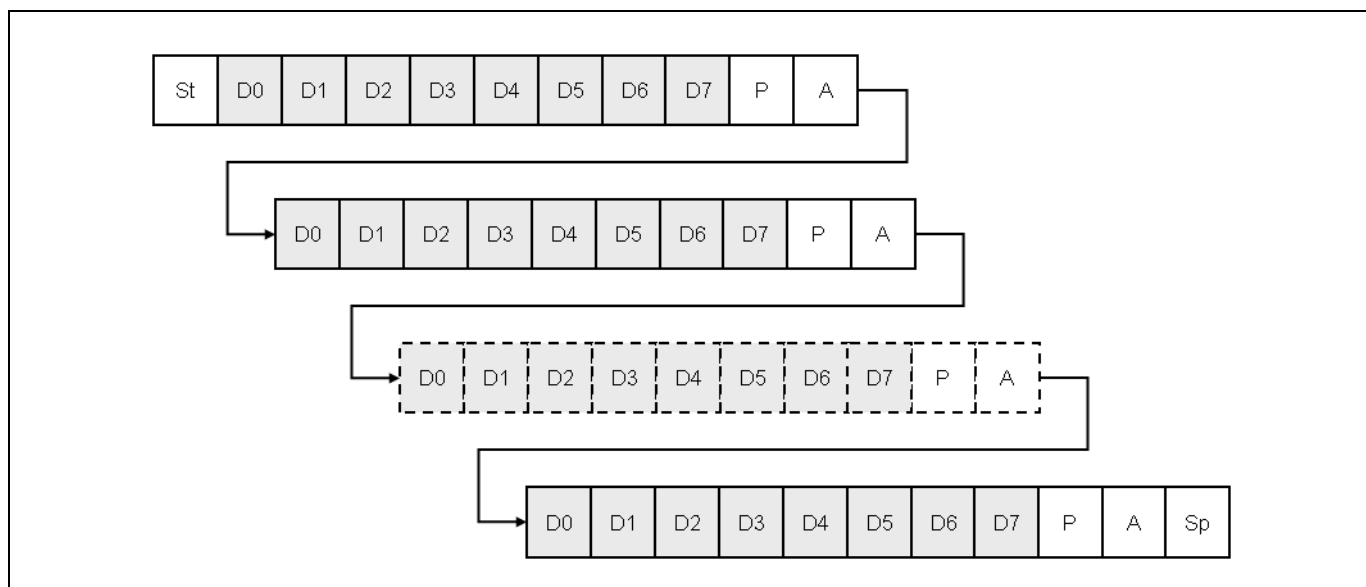


Figure 14.2 10-bit Transmission Packet

14.2.2 数据包传输时序

14.2.2.1 数据传送

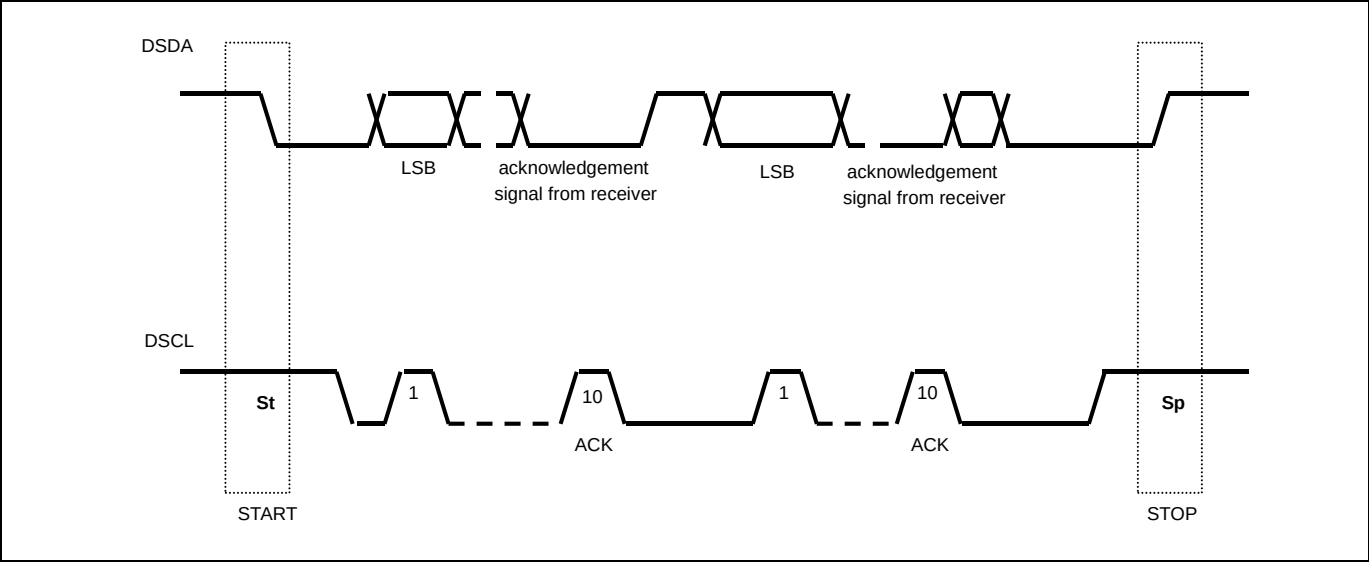


Figure 14.3 Data Transfer on the Twin Bus

14.2.2.2 位传输

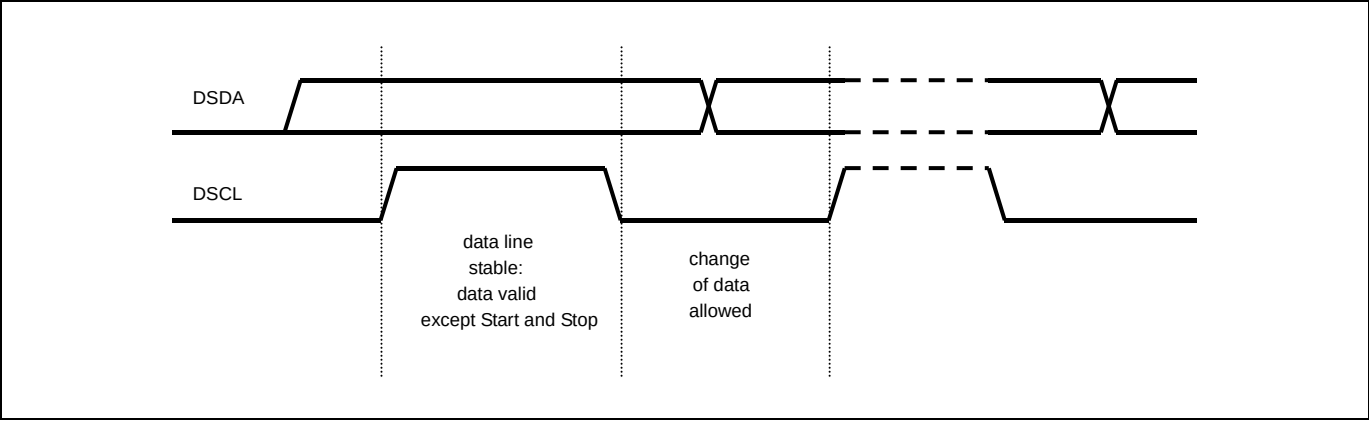


Figure 14.4 Bit Transfer on the Serial Bus

14.2.2.3 开始和停止信号

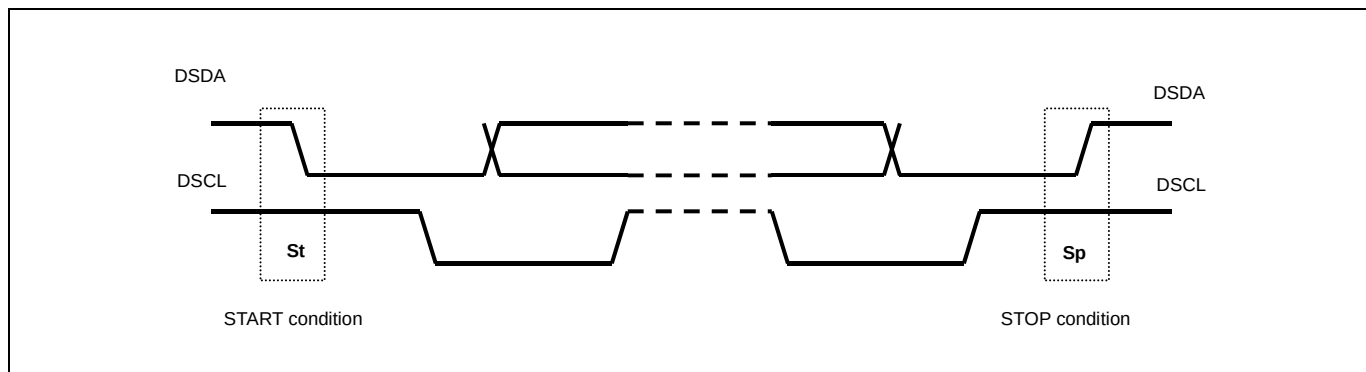


Figure 14.5 Start and Stop Condition

14.2.2.4 应答位

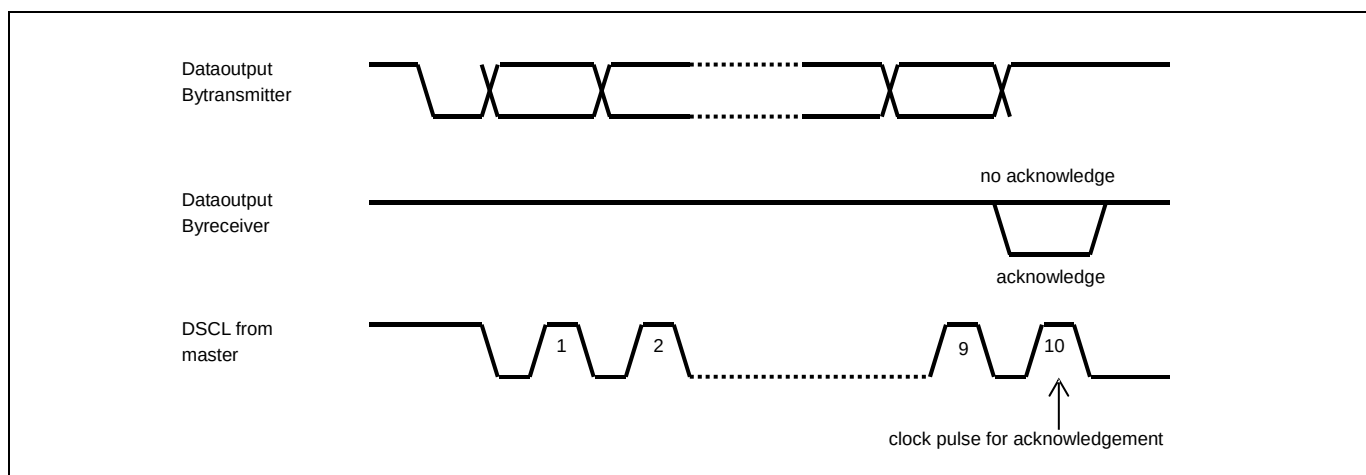


Figure 14.6 串行总线应答

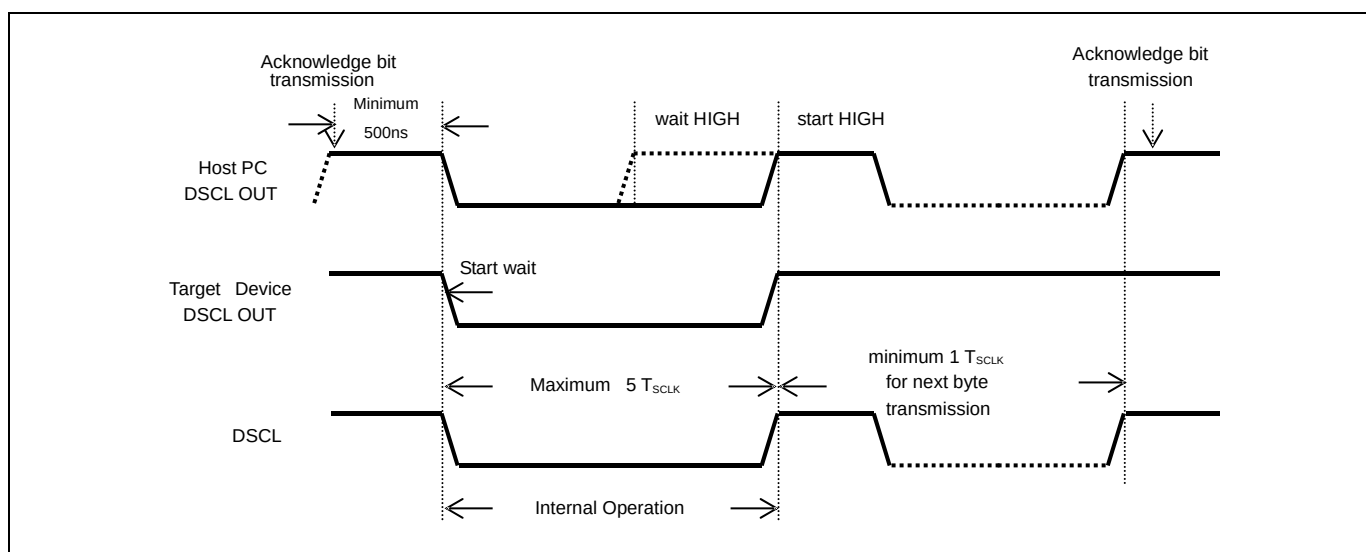


Figure 14.7 等待过程中的时钟同步

14.2.3 传输线路

双脚接口使用漏极开路线路(线与双向 I/O).

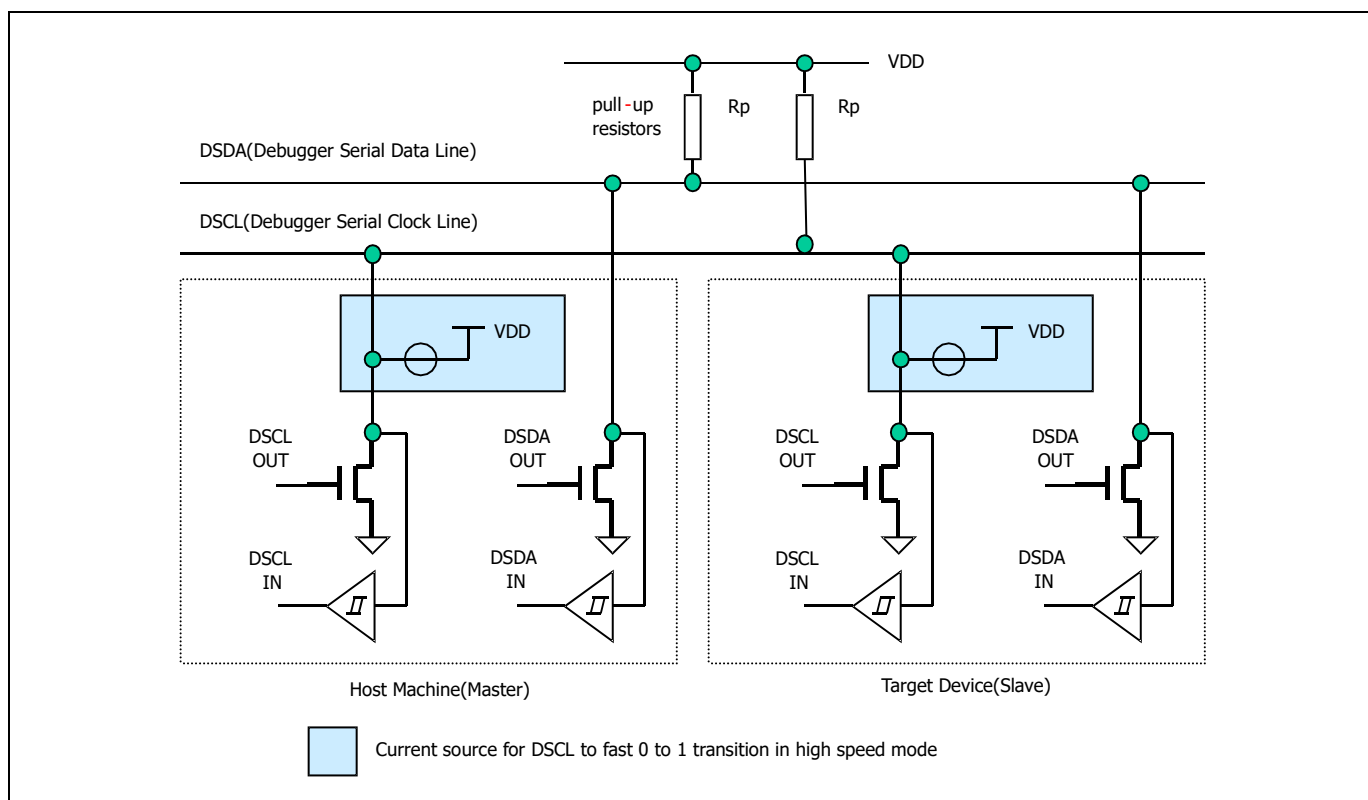


Figure 14.8 传输线路

15 存储器

15.1 概述

15.1.1 说明

MC96F8208S 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.

- Flash Size : 8Kbytes
- 电信号擦写
- 命令接口快速编程和擦除操作
- 典型温度和电压下，高达10,000(Sector 0~247)/100,000(Sector 248~255) 次擦写

15.1.2 Flash ROM 结构

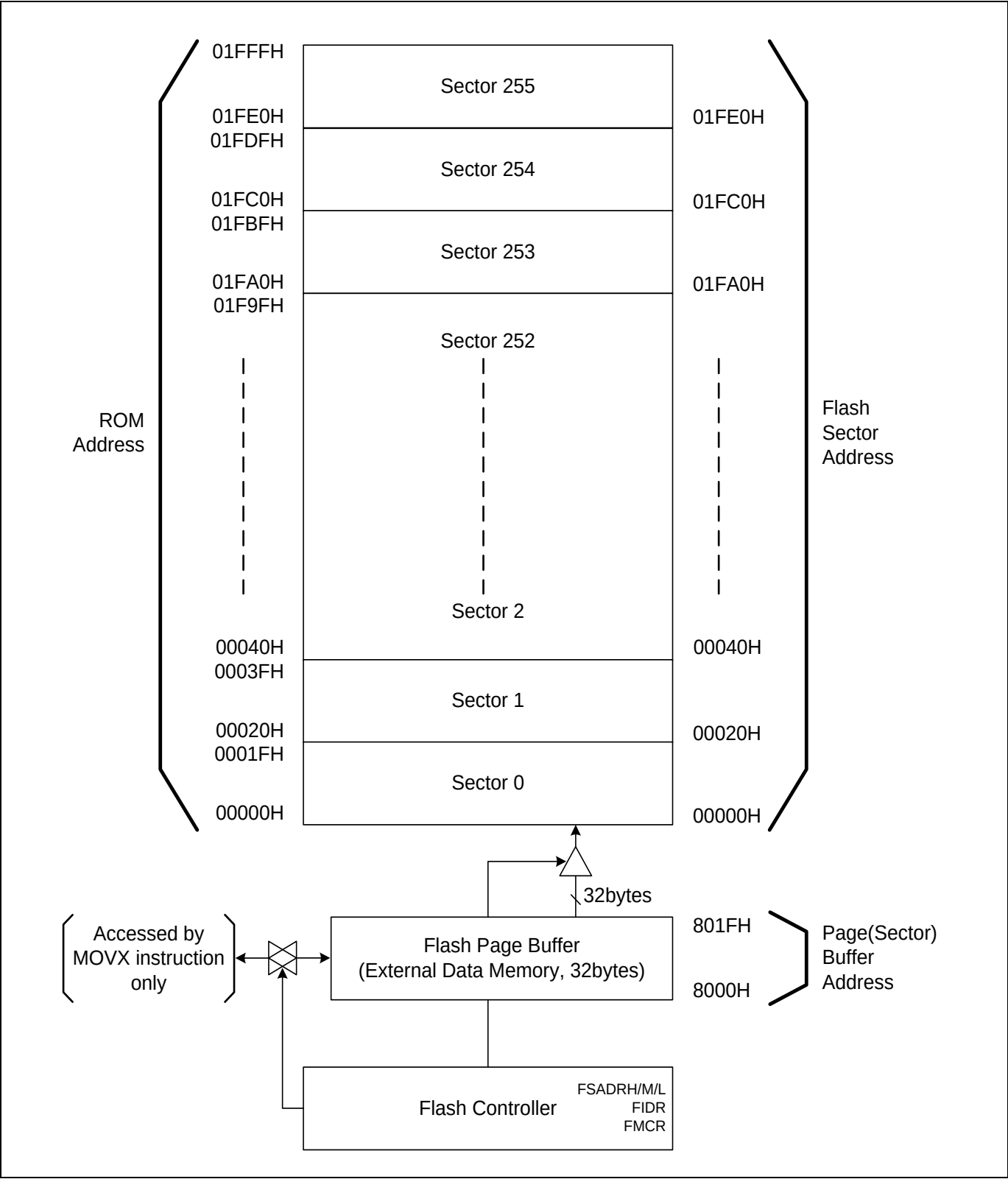


Figure 15.1 Flash Program ROM Structure

15.1.3 寄存器图

Name	Address	Direction	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

Table 15.1 Flash Memory 寄存器

15.1.4 Flash Memory 控制和状态寄存器说明

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.

15.1.5 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
—	—	—	—	RW	RW	RW	RW

Initial value : 00H

FSADRH[3:0] Flash Sector Address High

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
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

FSADRM[7:0] Flash Sector Address Middle

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
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

FSADRL[7:0] Flash Sector Address Low

FIDR (Flash 识别寄存器) : FDH

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

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管FIDR的值 (清除所有32个字节)
0	1	0	选择flash 页擦除模式并开始启动 当FIDR="10100101b'
0	1	1	选择flash 页写入模式并开始启动 当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)

15.1.6 串行ISP模式

串行ISP使用两根线作为接口. 具体参考14章

15.1.7 保护区域 (用户编程模式)

MC96F8208S 可对它本身的flash 存储器进行编程(保护区域). 保护区域是不可以被擦除的也不能烧录的. 保护区域只有在PAEN 位清零时可供使用。就是说，如果需要使能第2个配置项的保护区域的话，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.

保护区域的大小是可以通过配置的第二个选项改变的.

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

Table 15.2 保护区域大小

NOTE)

- 1. 参考 16 章配置控制选项.

15.1.8 擦除模式

用户编程模式时页擦除编程步骤

1. 清除页缓存器 (FMCR=0x01)
2. 页缓存器写入 '0'
3. 设置页擦除地址寄存器 (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. Erase verify

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, #0x1F
MOV      FSADRL, #0xA0        ;Select sector 253
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, #0x1F
MOV      DPL, #0xA0

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

Verify_error:

```

15.1.9 写模式

页写模式编程步骤

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. Erase verify

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, #0x1F
MOV      FSADRL, #0xA0       ;Select sector 253
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, #0x1F
MOV      DPL, #0xA0

Write_verify:
MOVC     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, #0x1F
MOV     FSADRL, #0xA0
MOV     FIDR, #0xA5
MOV     FMCR, #0x03
NOP                                           ;Select sector 253
NOP                                           ;Identification value
NOP                                           ;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, #0x1F
MOV     DPL, #0xA0
MOVC    A, @A+DPTR
SUBB    A, R1                   ;0x1FA0 = 5 ?
JNZ     Verify_error

MOV     A, #0
MOV     R1, #6
MOV     DPH, #0x1F
MOV     DPL, #0xA5
MOVC    A, @A+DPTR
SUBB    A, R1                   ;0x1FA5 = 6 ?
JNZ     Verify_error

```

Verify_error:

15.1.10 擦除\写入的保护

编写对FLASH进行擦除\写入操作的代码时要注意：必须有防错操作，防止程序错误跳转到FLASH擦除\写入的代码，以及因断电或干扰而造成的错误的擦除\写入操作。

1. 在FLASH擦除\写入代码中要检查用户ID.

```
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
```

如果像上面这样编写代码，就可以防止错误的FLASH擦除\写入操作

2.写入 UserID1/2/3 是很重要的.如果 UserID1/2/3 是在“MOV FIDR,#10100101B”这行指令上面写入的，将不能防止错误的FLASH擦除\写入操作，所以最好在返回指令之后写入 UserID1/2/3 。

```
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.Flash 扇区地址寄存器(FSADRH/FSADRM/FSADRL)里的地址值应该始终保持在用户数据区地址范围内，例如： FSADRH/FSADRM 的值始终为 0x00/0x1f （如果用户数据区地址为 0x1f00 到 0x1fff ）。

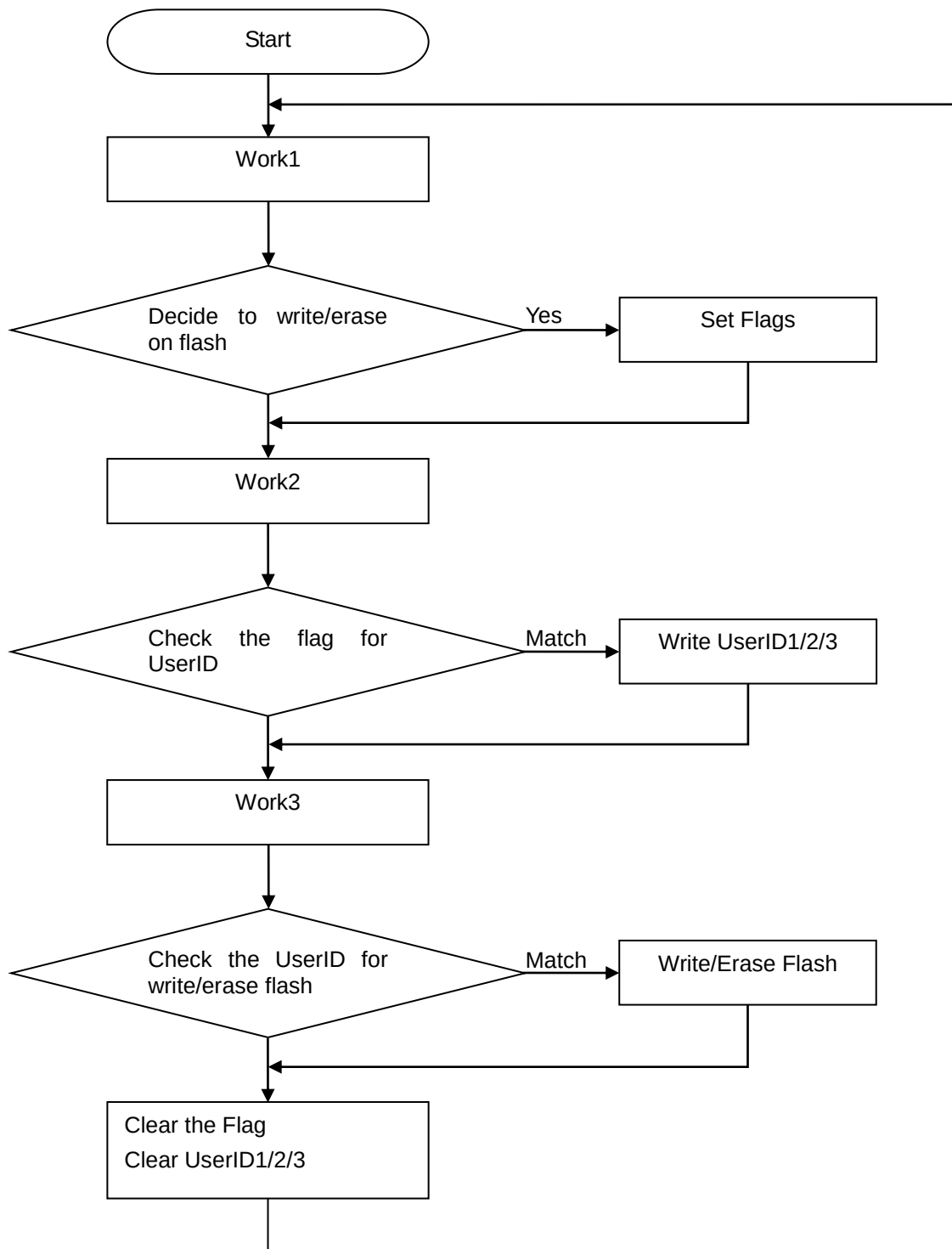
4.概述 main

```
---
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
```

15.1.10.1 Flow of Protection for Invalid Erase/Write

**NOTE)**

1. 这个操作流程可以帮助减少因错误操作、干扰、断电等导致的 FLASH 数据被擦除。

15.1.11 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, #0x1F
MOV    DPL, #0xA0                ;flash memory address

MOVC   A, @A+DPTR                ;read data from flash memory
```

15.1.12 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.
```

16 配置选项

16.1 配置选项控制

配置项的数据应该由烧录器写入到选项区域 (001EH – 001FH).

配置选项 1 : ROM Address 001FH

7	6	5	4	3	2	1	0
R_P	HL	–	–	–	–	–	RSTS

Initial value : 00H

- R_P

Code Read Protection

0禁止 “Read protection”

1使能 “Read protection”
- HL

Code Write Protection

0禁止 “Hard-lock”

1使能 “Hard-lock”
- RSTS

Select RESETB pin

0P32 □

1RESETB port with a pull-up resistor

配置选项 2: ROM Address 001EH

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

Initial value : 00H

- PAEN

Enable Specific Area Write Protection

0禁止 保护 (可通过指令消除)

1使能 保护 (不可通过指令消除)
- PASS [1:0]

Select Specific Area for Write Protection

PASS1	PASS0	Description
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)

17 附录

17.1 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

ADDITIONAL INSTRUCTIONS (selected through EO[7:4])				
Mnemonic	Description	Bytes	Cycles	Hex code
MOVC @(DPTR++),A	M8051W/M8051EW-specific instruction supporting software download into program memory	1	2	A5
TRAP	Software break command	1	1	A5

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. Non-continuous 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; other instructions use #data.

17.2 Instructions on how to use the input port.

- Error occur status
 - Using compare jump instructions with input port, it could cause error due to the timing conflict inside the MCU.
 - Compare jump Instructions which cause potential error used with input port condition:

```

JB      bit, rel    ; jump on direct bit=1
JNB     bit, rel    ; jump on direct bit=0
JBC     bit, rel    ; jump on direct bit=1 and clear
CJNE    A, dir, rel; compare A, direct jne relative
DJNZ    dir, rel    ; decrement direct byte, jnz relative

```

- It is only related with Input port. Internal parameters, SFRs and output bit ports don't cause any error by using compare jump instructions.
 - If input signal is fixed, there is no error in using compare jump instructions.
- Error status example

```

while(1){
if (P00==1){P10=1; }
else{ P10=0; }
    P11^=1;
}

```

```

zzz:    JNB      080.0, xxx    ; it possible to be
error
    SETB        088.0
    SJMP        yyy
xxx:    CLR      088.0
yyy:    MOV      C,088.1
    CPL        C
    MOV        088.1,C
    SJMP        zzz

```

```

unsignedchar
ret_bit_err(void)
{
return!P00;
}

```

```

    MOV        R7, #000
JB          080.0, xxx    ; it possible to be error
    MOV        R7, #001
xxx:    RET

```

- Preventative measures (2 cases)
 - Do not use input bit port for bit operation but for byte operation. Using byte operation instead of bit operation will not cause any error in using compare jump instructions for input port.

```

while(1){
if
((P0&0x01)==0x01){ P10=1; }
else{ P10=0; }
    P11^=1;
}

```

```

zzz:    MOV      A, 080        ; read as byte
JNB      0E0.0, xxx        ; compare
    SETB        088.0
    SJMP        yyy
xxx:    CLR      088.0
yyy:    MOV      C,088.1
    CPL        C
    MOV        088.1,C
    SJMP        zzz

```

- If you use input bit port for compare jump instruction, you have to copy the input port as internal parameter or carry bit and then use compare jump instruction.

```
bit tt;  
while(1){  
    tt=P00;  
    if (tt==0){P10=1;}  
    else {P10=0;}  
        P11^=1;  
}
```

```
zzz:    MOV        C,080.0    ; input port use internal  
        parameter  
        MOV        020.0, C    ; move  
        JB         020.0, xxx    ; compare  
        SETB       088.0  
        SJMP       yyy  
xxx:    CLR        088.0  
yyy:    MOV        C,088.1  
        CPL        C  
        MOV        088.1,C  
        SJMP       zzz
```

目 录

修订历史.....	2
1 概述.....	3
1.1. 说明.....	3
1.2. 特性.....	4
1.3. 开发工具.....	5
1.3.1 编译器.....	5
1.3.2 OCD(On-chip debugger) 仿真器和调试器.....	5
1.3.3 烧录器.....	8
1.4. MTP 烧录.....	11
1.4.1 概述.....	11
1.4.2 在线烧录.....	11
1.4.3 电路设计指导.....	12
2 方框图.....	13
3 引脚分配.....	14
4 封装图.....	16
5 引脚说明.....	20
6 端口结构.....	22
6.1 通用 I/O 口.....	22
6.2 外部中断 I/O 口.....	23
7 电气特性.....	24
7.1 额定参数.....	24
7.2 推荐操作参数.....	24
7.3 A/D 转换特性.....	25
7.4 上电复位特性.....	25
7.5 低电压复位和低电压检测特性.....	26
7.6 内部高速 RC 振荡特性.....	27
7.7 内部 WDT RC 振荡特性.....	27
7.8 DC 特性.....	28
7.9 AC 特性.....	29
7.10 SPI 特性.....	30
7.11 UART 特性.....	31
7.12 I2C 特性.....	32
7.13 Stop 模式数据保持电压.....	33
7.14 内部 Flash Rom 特性.....	34
7.15 输入/输出 电容.....	34
7.16 主时钟振荡特性.....	35
7.17 主时钟稳定特性.....	36
7.18 工作电压范围.....	37
7.19 推荐电路布局.....	38
7.20 典型特性.....	39
8 存储器.....	41
8.1 程序存储器.....	41
8.2 数据存储器.....	43
8.3 外部数据存储器.....	45
8.4 SFR 图.....	46
8.4.1 SFR 摘要.....	46

8.4.2	47	SFR
8.4.3	SFR	51
9	I/O 口	53
9.1	I/O 口	53
9.2	端口寄存器	53
9.2.1	数据寄存器 (Px)	53
9.2.2	方向寄存器 (PxIO)	53
9.2.3	上拉电阻选择寄存器 (PxPU)	53
9.2.4	漏极开路寄存器 (PxOD)	53
9.2.5	消抖使能寄存器 (PxDB)	53
9.2.6	端口功能选择寄存器 (PxFSR)	53
9.2.7	寄存器	54
9.3	P0 口	55
9.3.1	P0 端口说明	55
9.3.2	P0 寄存器说明	55
9.4	P1 口	58
9.4.1	P1 端口说明	58
9.4.2	P1寄存器说明	58
9.5	P2 口	62
9.5.1	P2 端口说明	62
9.5.2	P2寄存器说明	62
9.6	P3 口	64
9.6.1	P3 端口说明	64
9.6.2	P3端口寄存器说明	64
10	中断控制器	66
10.1	概述	66
10.2	外部中断	68
10.3	方框图	69
10.4	中断向量表	70
10.5	中断序列	71
10.6	控制中断位之后的作用时序	72
10.7	多中断复用	73
10.8	中断使能响应时序	74
10.9	中断服务程序地址	74
10.10	通用寄存器的设置和复位	74
10.11	中断时序	75
10.12	中断寄存器概述	76
10.12.1	中断使能寄存器 (IE, IE1, IE2, IE3)	76
10.12.2	中断优先级寄存器(IP, IP1)	76
10.12.3	外部中断标志位寄存器(EIFLAG0, EIFLAG1)	76
10.12.4	外部中断边沿寄存器(EIPOL0H, EIPOL0L, EIPOL1, EIPOL2)	76
10.12.5	寄存器图	76
10.12.6	中断寄存器说明	77
10.12.7	中断寄存器	77
11	外围硬件	83
11.1	时钟发生器	83
11.1.1	概述	83
11.1.2	方框图	83
11.1.3	寄存器图	84
11.1.4	时钟发生器寄存器说明	84
11.1.5	时钟发生器寄存器	85

11.2	BIT	86
11.2.1	概述	86
11.2.2	方框图	86
11.2.3	寄存器图	86
11.2.4	BIT寄存器说明	87
11.2.5	BIT寄存器	87
11.3	WDT	88
11.3.1	概述	88
11.3.2	WDT 中断时序波形	88
11.3.3	方框图	89
11.3.4	寄存器	89
11.3.5	WDT寄存器说明	89
11.3.6	WDT寄存器说明	90
11.4	WT	91
11.4.1	概述	91
11.4.2	框图	91
11.4.3	寄存器图	91
11.4.4	WT寄存器说明	92
11.4.5	WT寄存器	92
11.5	定时器 0	94
11.5.1	概述	94
11.5.2	8-bit 定时器/计数器 模式	95
11.5.3	8-bit PWM 模式	96
11.5.4	8-bit 捕捉模式	98
11.5.5	方框图	100
11.5.6	寄存器	100
11.5.7	定时器/计数器0 寄存器说明	100
11.5.8	定时器/计数器 0寄存器	101
11.6	定时器 1	103
11.6.1	概述	103
11.6.2	16-bit 定时器/计数器 模式	103
11.6.3	16-bit 捕捉模式	105
11.6.4	16-bit PPG 模式	107
11.6.5	框图	109
11.6.6	寄存器	109
11.6.7	定时器/计数器 1 寄存器说明	110
11.6.8	定时器/计数器 1 寄存器	110
11.7	定时器 2	113
11.7.1	概述	113
11.7.2	16-bit 定时器/计数器 模式	114
11.7.3	16-bit 捕捉模式	116
11.7.4	16-bit PPG 模式	118
11.7.5	方框图	120
11.7.6	寄存器	120
11.7.7	定时器/计数器 2 寄存器说明	121
11.7.8	定时器/寄存器 2 寄存器	121
11.8	蜂鸣器驱动	124
11.8.1	概述	124
11.8.2	方框图	124
11.8.3	寄存器图	125
11.8.4	蜂鸣器驱动寄存器说明	125
11.8.5	蜂鸣器驱动寄存器	125
11.9	12-bit A/D 转换器	126
11.9.1	概述	126
11.9.2	转换时序	126

11.9.3	方框图	127
11.9.4	ADC 操作	128
11.9.5	ADC 睡眠唤醒功能	129
11.9.6	ADC 睡眠唤醒功能框图	130
11.9.7	ADC 睡眠唤醒功能操作	130
11.9.8	ADC 按键输入串联电阻应用电路	131
11.9.9	寄存器图	132
11.9.10	ADC 寄存器说明	132
11.9.11	ADC 寄存器说明	133
11.10	SPI	139
11.10.1	概述	139
11.10.2	方框图	139
11.10.3	数据 发送/接收 操作	140
11.10.4	SPI 时序图	141
11.10.5	寄存器	142
11.10.6	SPI 寄存器说明	142
11.10.7	SPI 寄存器	142
11.11	UART	144
11.11.1	概述	144
11.11.2	方框图	145
11.11.3	时钟发生器	146
11.11.4	数据格式	147
11.11.5	校验位	147
11.11.6	UART 发送	148
11.11.6.1	发送 Tx 数据	148
11.11.6.2	发送标志位和中断	148
11.11.6.3	奇偶校验发生器	148
11.11.6.4	禁止发送	148
11.11.7	UART 接收	149
11.11.7.1	接收 Rx 数据	149
11.11.7.2	接收标志位和中断	149
11.11.7.3	奇偶校验	150
11.11.7.4	禁止接收	150
11.11.7.5	异步数据接收	150
11.11.8	寄存器	152
11.11.9	UART 寄存器说明	152
11.11.10	UART 寄存器	153
11.11.11	波特率设置 (example)	157
11.12	I2C	158
11.12.1	概述	158
11.12.2	方框图	159
11.12.3	I2C 位发送	159
11.12.4	开始 / 重复开始 / 停止	160
11.12.5	数据传输	160
11.12.6	I2C 应答	161
11.12.7	同步 / 仲裁	162
11.12.8	操作	163
11.12.8.1	 主机发送	163
11.12.8.2	主机接收	165
11.12.8.3	I2C 从机接收	167
11.12.8.4	从机接收	168
11.12.9	寄存器	169
11.12.10	I2C 寄存器说明	169
11.12.11	I2C 寄存器	170

12	省电模式操作	174
12.1	概述	174
12.2	IDLE/STOP 模式下外围的操作	174
12.3	IDLE 模式	175
12.4	STOP 模式	176
12.5	STOP 模式唤醒	177
12.6	寄存器	178
12.7	省电操作寄存器说明	178
12.8	省电操作寄存器	178
13	复位	179
13.1	概述	179
13.2	复位源	179
13.3	复位方框图	179
13.4	RESET 噪音消除器	180
13.5	Power on RESET	180
13.6	外部复位	183
13.7	BOD处理器	184
13.8	LVI 方框图	186
13.9	寄存器图	186
13.10	复位操作寄存器说明	186
13.11	复位操作寄存器	187
14	片内调试系统(MC96F8208 EVA ONLY)	190
14.1	概述	190
14.1.1	说明	190
14.1.2	特性	191
14.2	双线外围接口	192
14.2.1	基本传输包	192
14.2.2	数据包传输时序	193
14.2.2.1	数据传送	193
14.2.2.2	位传输	193
14.2.2.3	开始和停止信号	194
14.2.2.4	应答位	194
14.2.3	传输线路	195
15	存储器	196
15.1	概述	196
15.1.1	说明	196
15.1.2	Flash ROM 结构	197
15.1.3	寄存器图	198
15.1.4	Flash Memory 控制和状态寄存器说明	198
15.1.5	Flash寄存器说明	199
15.1.6	串行ISP模式	201
15.1.7	保护区域 (用户编程模式)	201
15.1.8	擦除模式	202
15.1.9	写模式	203
15.1.10	读模式	208
15.1.11	硬件锁定模式	208
16	配置选项	209
16.1	配置选项控制	209
17	附录	210

17.1 指令表	210
目 录	216