
具有12位ADC的 8位CMOS单片机



主要特性

- 8位高速 CM8051-S CPU
- 基本 MCU 功能
 - 2K字节 Flash 程序存储器
 - 代码区保护功能
 - 256字节SRAM
- 内置模拟功能
 - 上电复位和低电压检测复位
 - 内部 32MHz RC 振荡器
- 外围特性
 - 12位AD转换器
- I/O 和封装
 - 最多14个可编程 I/O (16SOP)
- 工作条件
 - -40°C to 85°C温度范围
- 应用
 - 电池充放电控制

MC94F1202A

用户手册

V 1.4

June15, 2016

Revision history

Version	Date	Revision list
1.0	2016.03.03	Initial preliminary version
1.1	2016.04.29	Change Version Start V1.0 Change BOD to LVI (Name Change) Update Table format and contents in 7 Electrical Characteristics. Update MCU Stabilization time. Add 7.10 Operating Voltage Range and 7.11 Typical Characteristics.
1.2	2016.05.02	Fixed the errors of the figure 2.1. ([P7:0] -> [P6:0] , 10 Page) Update Table 7.1 Absolute maximum ratings. (16 Page) Update Table 7.3 Analog ADC Voltage (16 Page) Fixed Table 7.8 DC Characteristics (IDD2 Max 5mA ->4mA, 17 Page) 11.5.4 ADC Zero offset delete. Add 10-SSOP Package. (1.1 Description, 3 Pin Assignment, 4 Package Dimension) Table of Contents fixed. (99 Page)
1.3	2016.05.19	Fixed Value. (RSFR Initial / Default Value, 75-76 Page) Add NOTE for OCD reset state. (RSFR, 76 Page) Update Table 15.2. Fixed Flash Control Register ADDRESS.
1.4	2016.06.15	Fixed Value. (POR level 1.4V -> 1.1V, Page 73-74). Delete unused Register. (DPL1,DPH1, Page 26)

Version 1.4

Published by FAE team

2016 ABOV Semiconductor Co. Ltd. all rights reserved.

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

MC94F1202A是一款具有2K字节FLASH的8位CMOS单片机。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。主要有以下特点：2K字节FLASH, 256字节SRAM, 16位定时器/计数器/PWM, 具有WDTOSC独立时钟的看门狗定时器, 12位ADC, 片内POR, LVI和LVR, 内部RC振荡器, 内部WDT振荡器和时钟电路。MC94F1202A具有省电模式以降低功耗。

Device Name	FLASH	IRAM	XRAM	ADC	I/O PORT	Package
MC94F1202AMBN	2Kbytes	256bytes	-	14inputs	14	16-SOP
MC94F1102ASBN	2 Kbytes	256 bytes	-	8 inputs	8	10-SSOP
MC94F1102AMBN	2 Kbytes	256 bytes	-	6 inputs	6	8-SOP

Table 1.1 Ordering Information of MC94F1202A

1.2 特性

- CPU

- CM8051-S (兼容8051, 指令周期为1个时钟周期)

- 2 K字节FLASH

- 擦写次数: 10,000 次 (室温环境)
- 保存年限: 10 年
- 自编程: (代码保护选项)

- 256字节IRAM

- 输入输出端口

- GPIO 14

- 定时器/计数器

- 16位2通道 (Timer0, Timer1)
- 8位1通道(Timer2)

- PWM (16位2通道, Timer0, 1)

- Watch Dog Timer

- 12位ADC

- 14个输入通道
- 1个内部参考通道

- 中断源

- 外部中断 (3, PCI)
- Timer (3)
- ADC(1)
- WDT(1)
- LVI(1)

- 内部 RC 振荡器

- 32MHz($\pm 3\%$) 振荡器

- 内部 WDT 振荡器

- 8kHz($\pm 50\%$) 振荡器

- 上电复位

- 1.1 V

- 低电压复位

- 1-Level (1.8 V)

- 低电压检测

- 3-Level (2.1V, 2.5V, 3.5 V)

- 最小指令执行时间

- 125ns (@8MHz, NOP Instruction)

- 省电模式

- IDLE, STOP1, STOP2 mode

- 工作频率

- 1/4/8/16MHz

- 工作电压

- 2.2V~5.5V

- 工作温度: $-40 \sim +85^{\circ}\text{C}$

- 封装类型

- 16-SOP, 10-SSOP, 8-SOP
 - Pb free package
-

1.1 开发工具

1.1.1 编译器

ABOV 公司不提供编译工具。建议咨询编译器提供商。MC94F1202A是 CM8051-S 内核，可以使用所用8051标准编译器进行编译（如Keil C或开源SDCC），这些编译器编译产生的调试信息可以集成到我们的OCD仿真器里用于程序调试，更多详细信息请参考OCD的说明手册。

1.1.2 OCD2 仿真器和调试器

OCD2仿真器支持 ABOV 半导体8051系列MCU的仿真。OCD2使用两根线连接PC和目标板上的MCU。OCD2可以读取和更改MCU内部存储器、 I/O 、外围设备的数值，还可以控制MCU内部调试逻辑进行单步运行、监测等仿真功能。

OCD2 调试软件可以在Microsoft的Windows NT/2000/XP/Vista/Win7/Win8/Win10等32位或64位操作系统下运行。如果想要了解更详细的信息，请参考OCD2的说明手册。在ABOV的网站可以下载调试软件和说明手册。

OCD2接线方式

- P01(MC94F1202ADSCL pin)
- P00 (MC94F1202ADSDA pin)

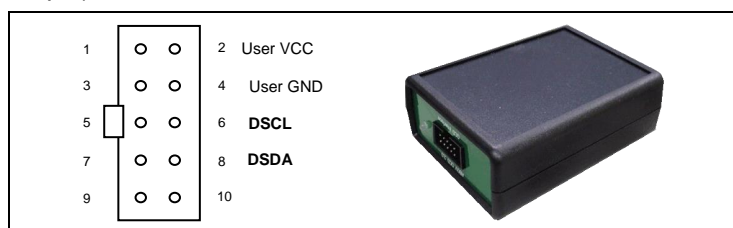


Figure 1.1 On Chip Debugger 2nd Pin description (ocd2 mode)

OCD2 (On Chip Debug) 仿真器

- MCU 需要通过 2 根或 3 根线与 OCD2 连接
- 2 线接口：OCD2 clock & data.
- 还有 1 个引脚是选项接口
 - 在用户软件运行时，支持器件进入 OCD2 模式
 - 可以精确测量仿真时间
- 比 OCD1 更快的接口速度
- 支持新的调试功能
 - 数据断点 (1, 2, 4 字节),
 - 内部 OSC 频率测量和调整
- 尺寸紧凑
- 仿真器成本降低
- 支持目标板上直接仿真和调试
- 实时仿真
- PC 接口：USB

调试器

- 用于操作 OCD1 和 OCD2 仿真器硬件
- 集成开发环境 (IDE)
 - 支持窗口和菜单
- 支持连续运行、单步运行、自动单步运行
- 支持变量调试
- 支持源代码调试

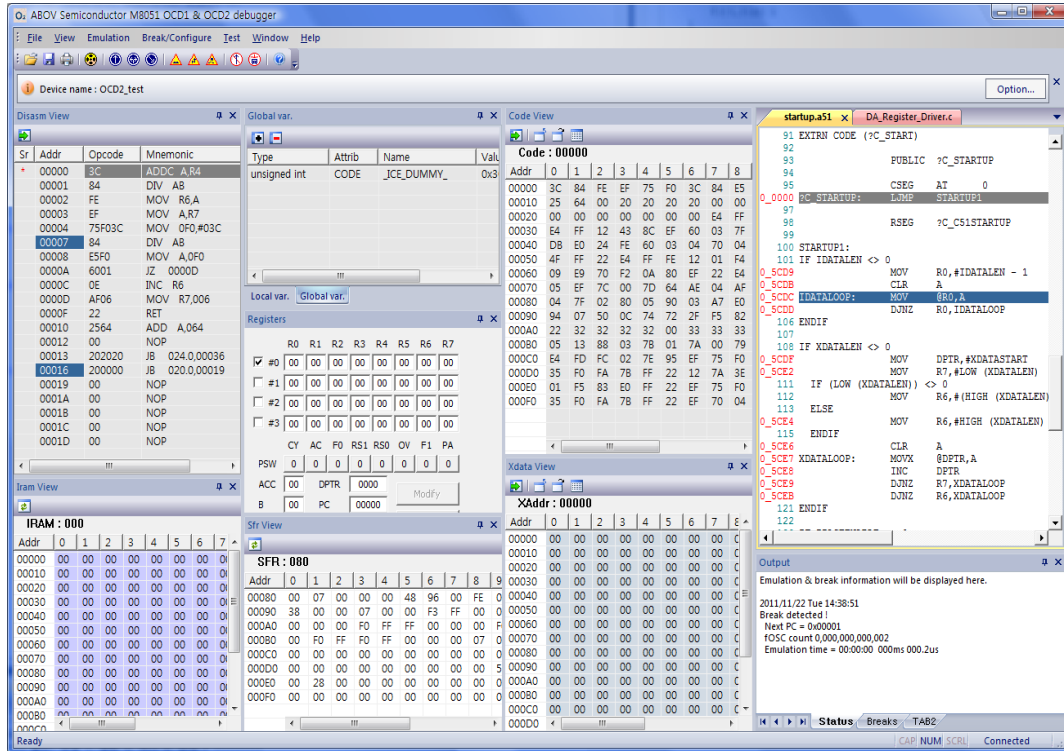


Figure 1.2 OCD Debugger

支持的芯片

- MC95xxxx
- MC96xxxx
- MC97xxxx

1.1.3 烧写器

E-PGM +

- 支持 ABOV / ADAM 芯片
- 比 S-PGM+ 的烧录速度快2~5倍
- 使用32位 MCU @ 72MHz
- 缓冲大小: 1 M 字节

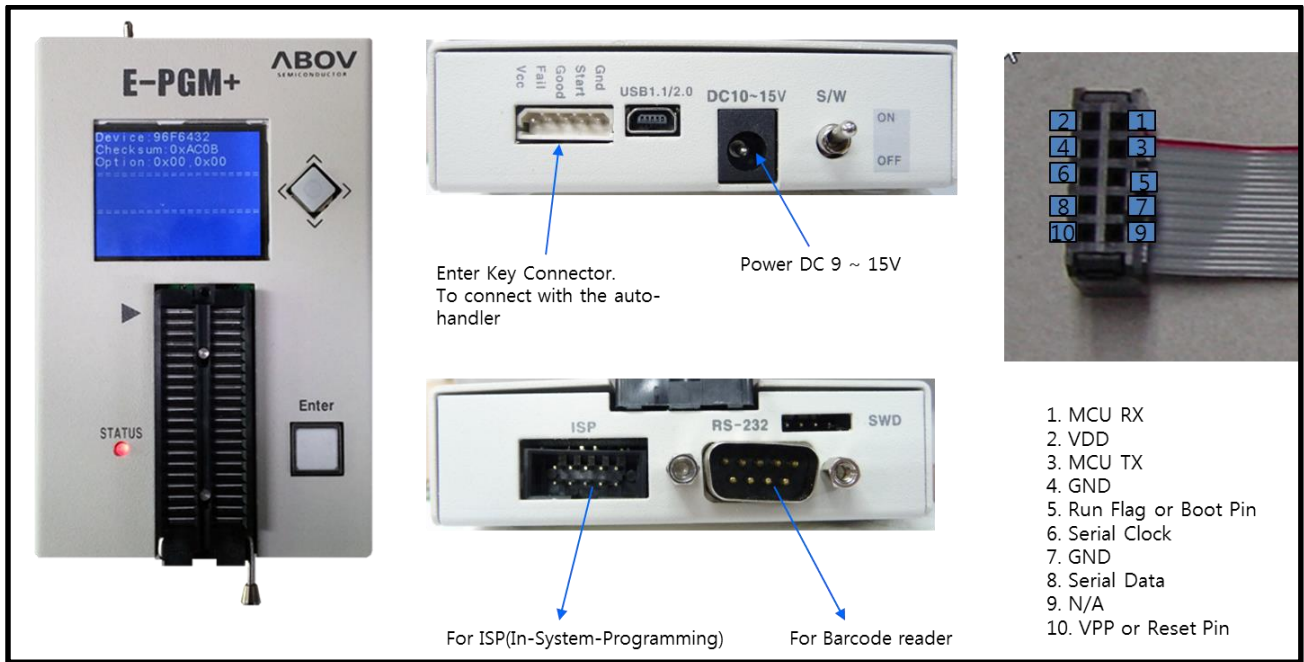


Figure 1.3 PGMplus USB

PGMPlusLC 2

说明

PGMPlusLC2使用 ISP (In System Programming) 接口把代码写入到MCU
可以使用10针排线把目标板连接到烧录器

特性

- PGMplusLC2是低成本烧录器
- 支持USB接口
- 不需安装USB驱动
- 需要连接外部电源适配器 (5v@2A)
- 使用高速32位 Cortex-M3 MCU
- 最高支持18V的电源电压
- PGMplusLC2是用于PC环境的
- PGMplusLC2的速度要比PGMplusLC快
- 传输速度为 64K字节/秒

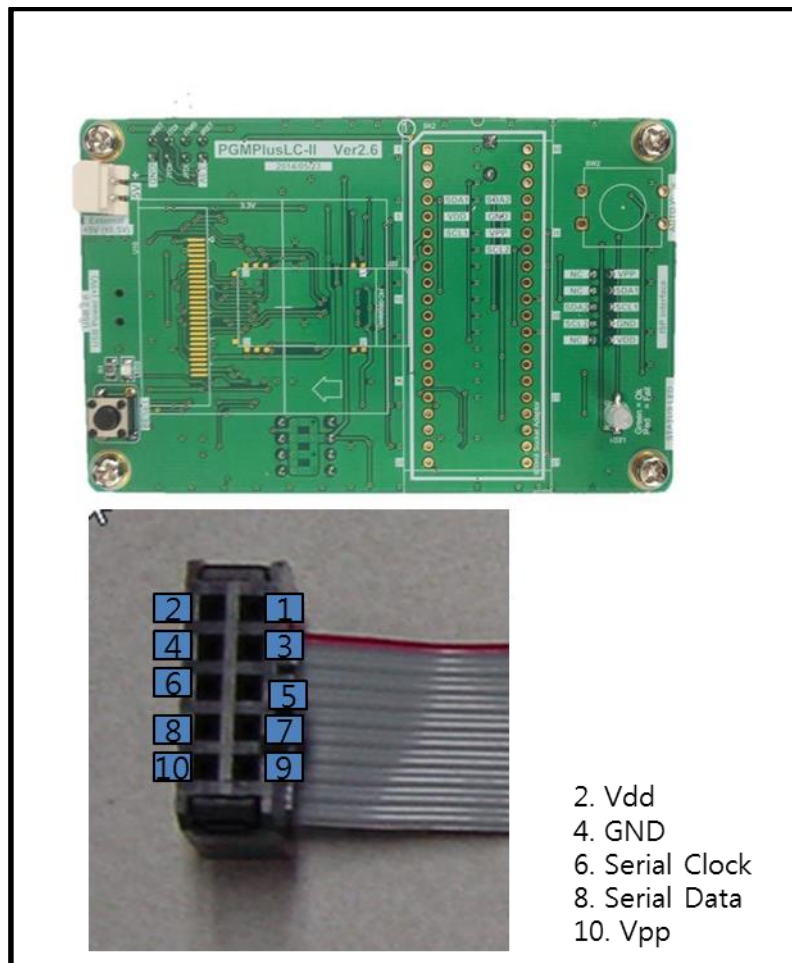


Figure 1.4 PGMplusLC Writer

E-PGM+ Gang4/6

- 产品名: **E-PGM+ GANG 4**
 - 外形尺寸(x , y, h): 33.5 x 22.5 x35mm
 - 重量: 2.0kg
 - 输入电压: 最高 DC 15V/2A
 - 工作温度: -10 ~ 40°C
 - 储存温度: -30 ~ 80°C
 - 防水等级: 不防水
-
- 产品名: **E-PGM+ GANG 6**
 - 外形尺寸(x , y, h): 148.2 x 22.5 x35mm
 - 重量: 2.8kg
 - 输入电压: 最高 15V/2A
 - 工作温度: -10 ~ 40°C
 - 储存温度: -30 ~ 80°C
 - 防水等级: 不防水

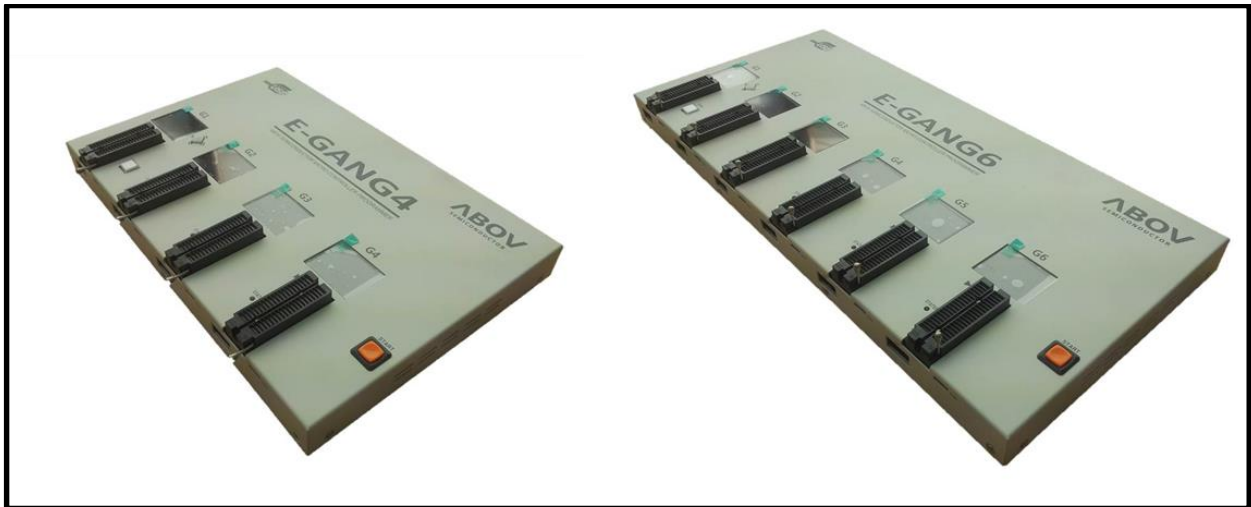


Figure 1.5 Gang Programmer

2 方框图

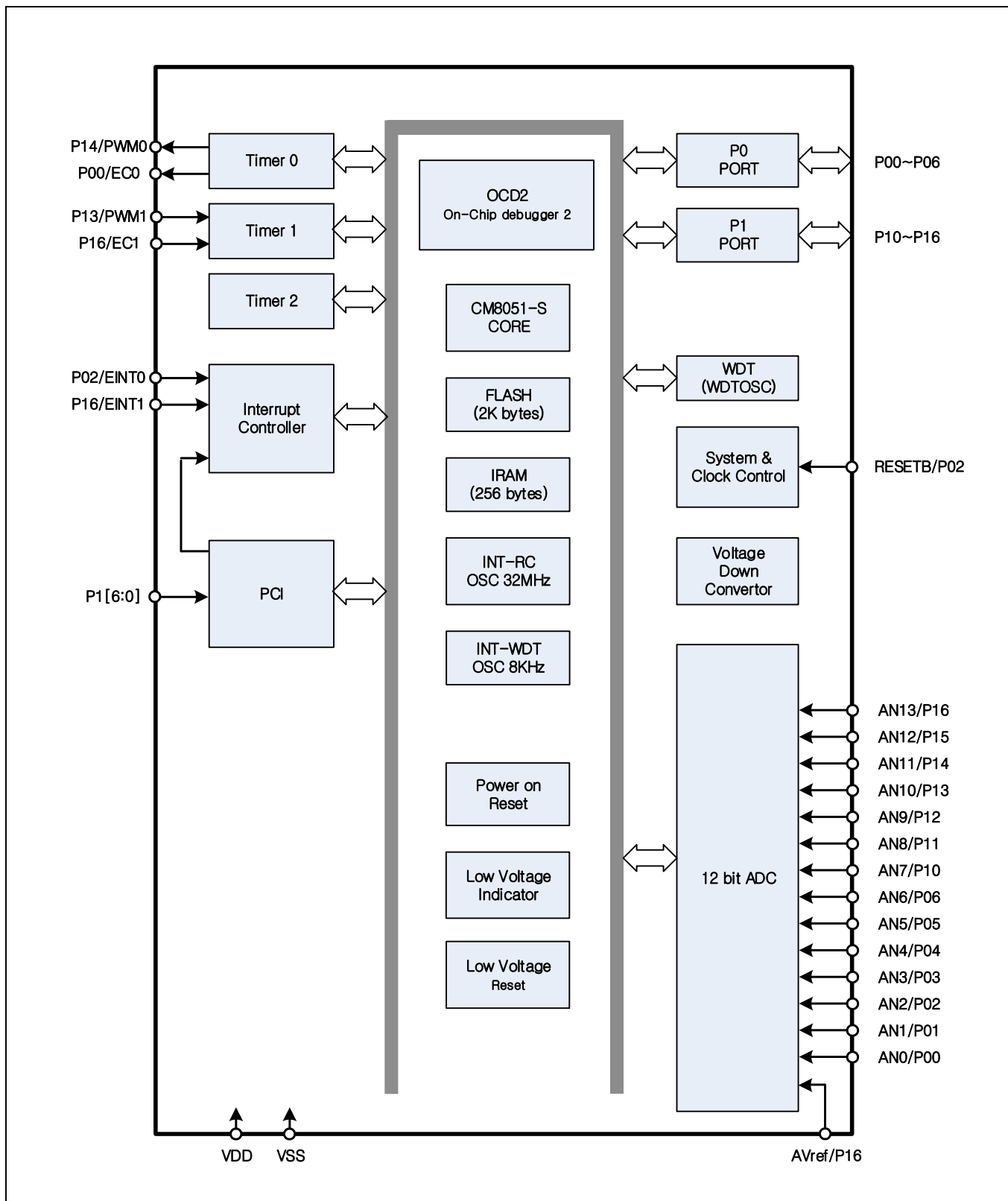


Figure 2.1 Block diagram of MC94F1202A

3 引脚分配

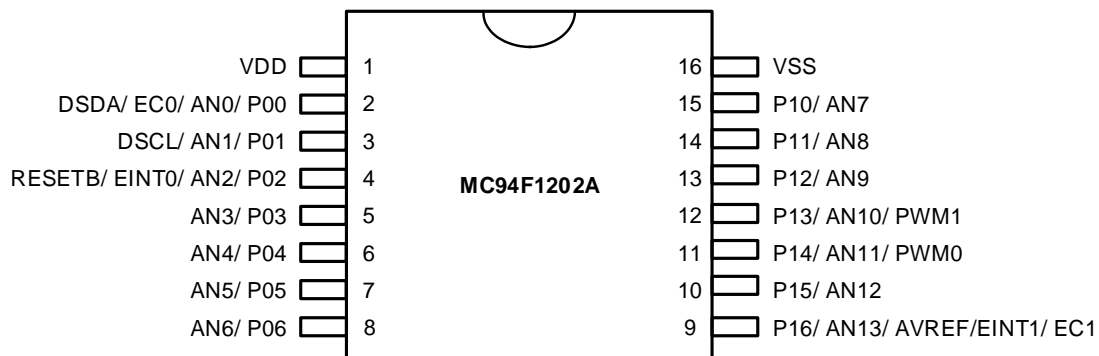


Figure 3.1 MC94F1202A 16pin assignment

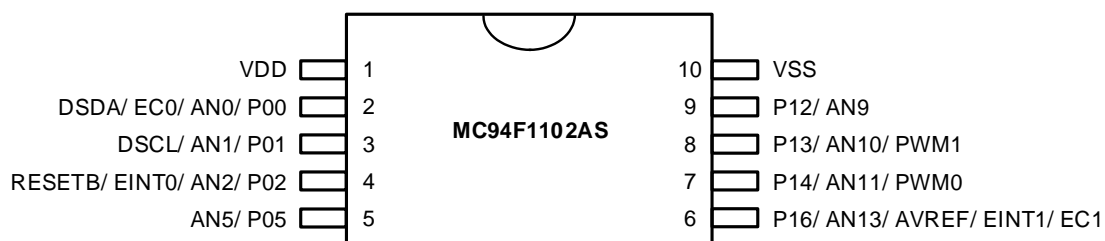


Figure 3.2 MC94F1102AS10 pin assignment



Figure 3.3 MC94F1102AM8 pin assignment

注意)当使用10脚或8脚封装时，没有的以及不使用的引脚需要设置为输入带上拉电阻或都输出状态，以防止功耗增大。

4 封装图

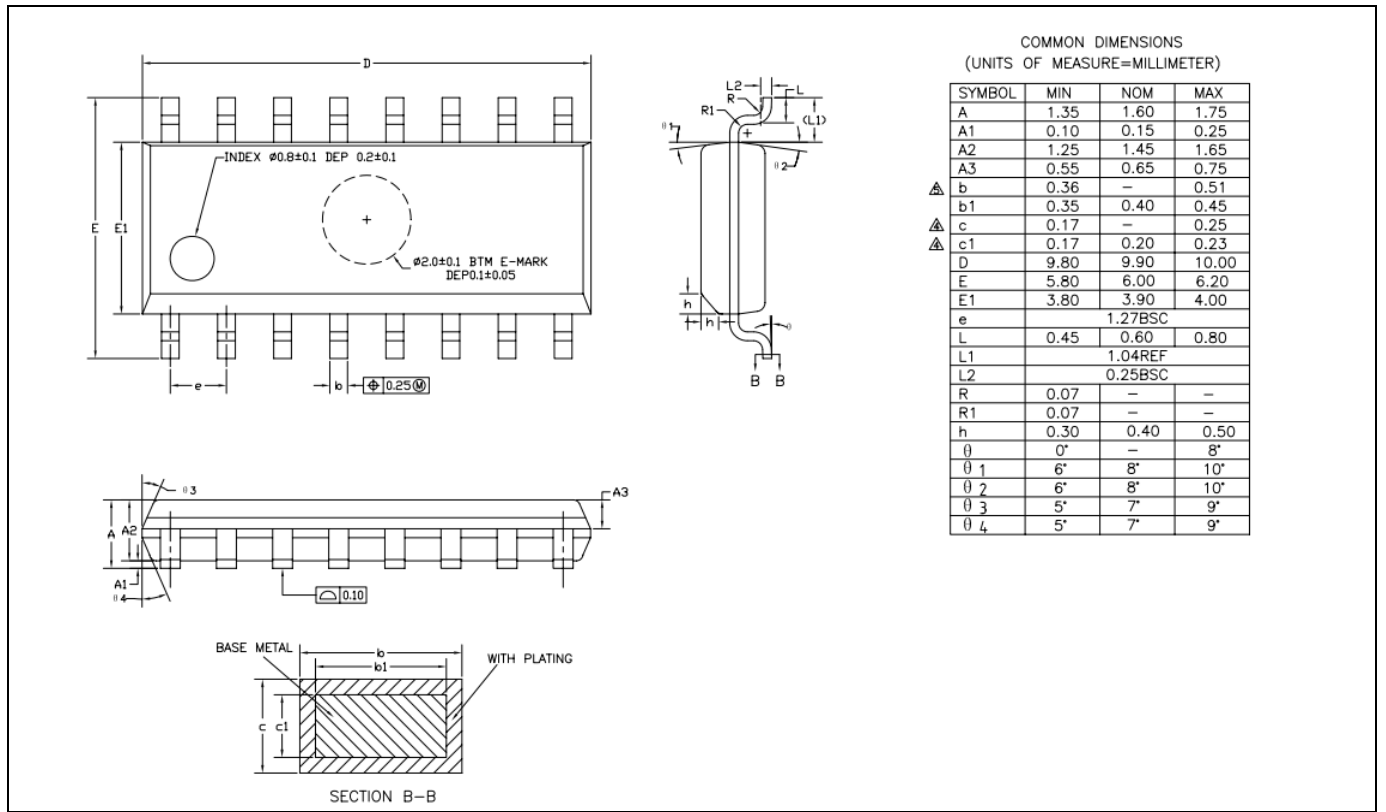


Figure 4.1 16-PinSOP Package

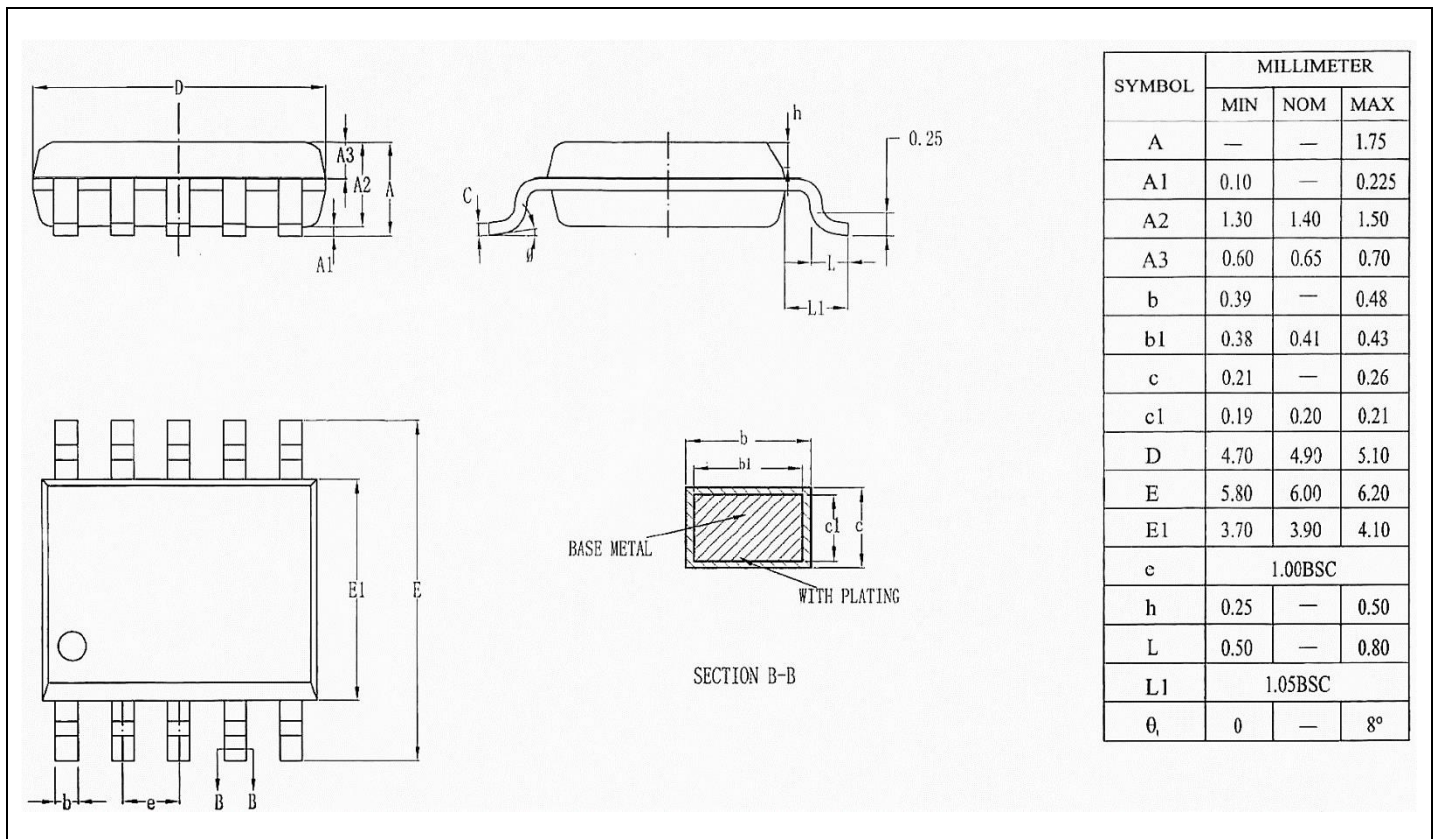


Figure 4.2 10-Pin SSOP Package

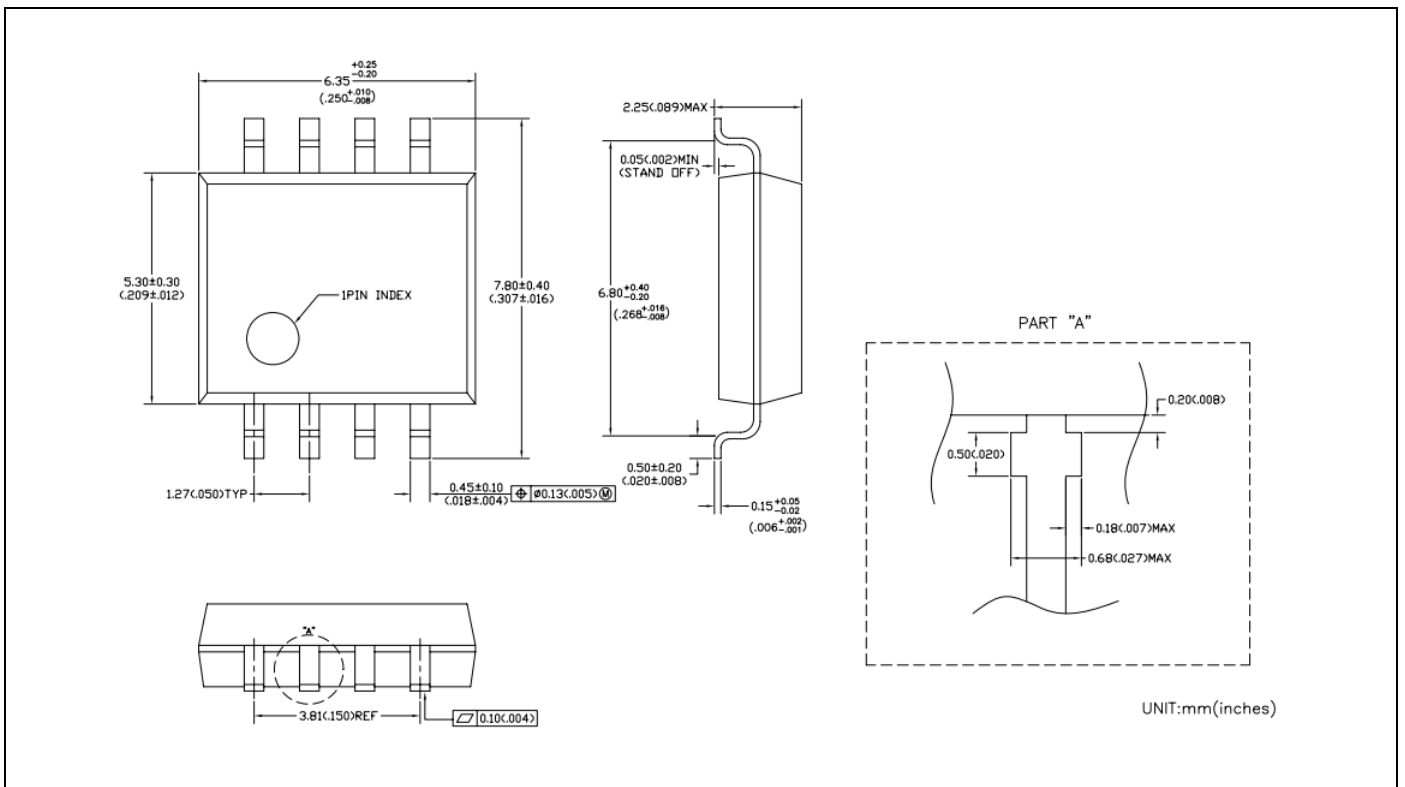


Figure 4.3 8-Pin SOP Package

5 引脚说明

PIN Name	I/O	Function	@RESET	Shared with
P00	I/O	P0口 7位I/O端口 每个口都可以设置为输入或输出状态 作为输入口使用时可通过软件设置内部上拉 作为输出口使用时可通过软件使能漏极开路寄存器	Input	AN0/ EC0/ DSDA
P01				AN1/ DSCL
P02				AN2/ EINT0/ RESETB
P03				AN3
P04				AN4
P05				AN5
P06				AN6
P10	I/O	P1口 7位I/O端口 每个口都可以设置为输入或输出状态 作为输入口使用时可通过软件设置内部上拉 作为输出口使用时可通过软件使能漏极开路寄存器	Input	AN7
P11				AN8
P12				AN9
P13				AN10/ PWM1
P14				AN11/ PWM0
P15				AN12
P16				AN13/ AVREF/EINT1/ EC1
VDD		电源		
VSS		地		

Table 5.1 Pin Description

6 端口结构

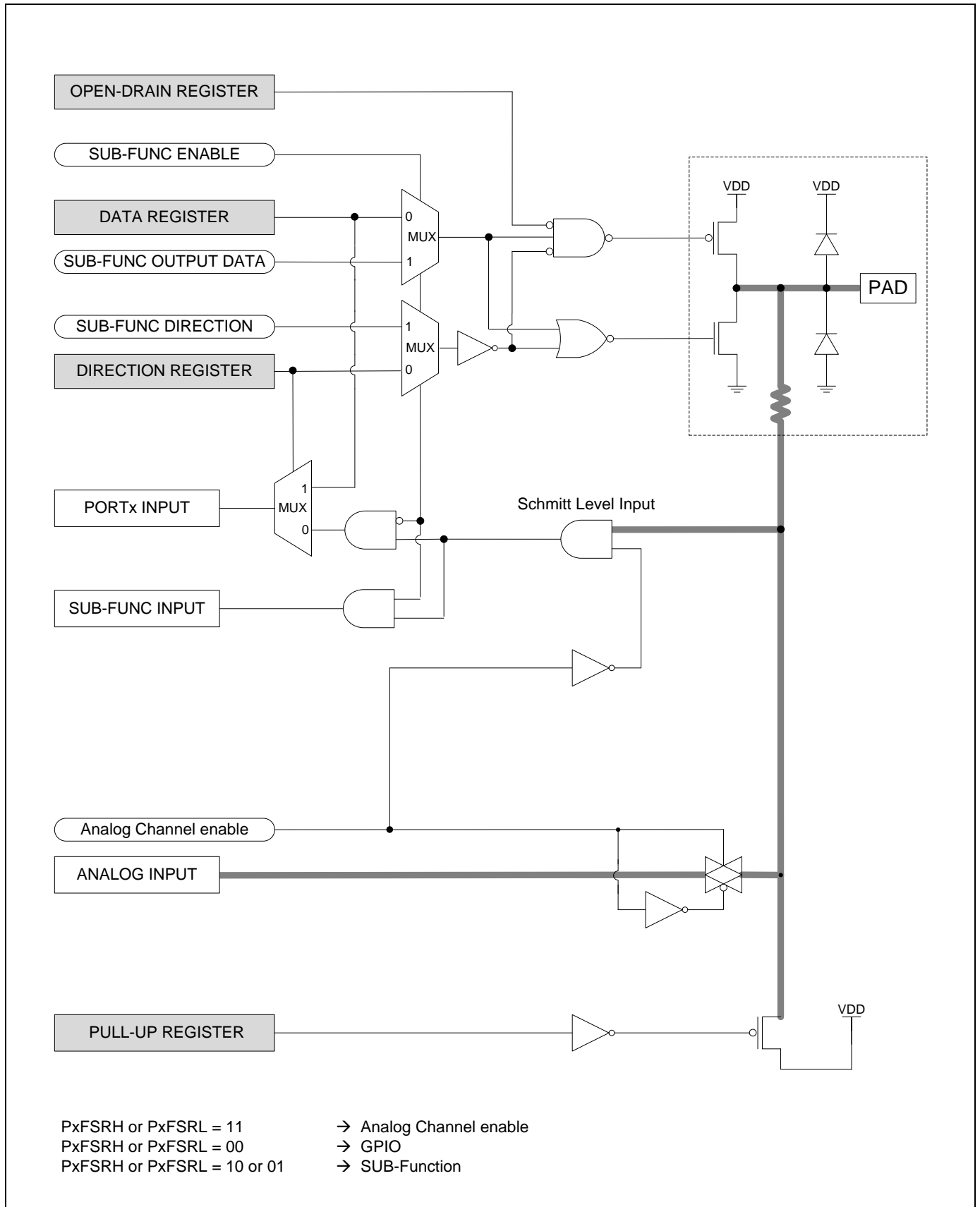


Figure 6.1 Second Function I/O Port

7 电气特性

7.1 极限参数

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	所有引脚对地的电压
	V _O	-0.3~VDD+0.3	V	
	I _{OH}	-10	mA	每个引脚的最大输出电流 (I _{OH} per I/O pin)
	∑I _{OH}	-80	mA	最大电流 (∑I _{OH})
	I _{OL}	20	mA	每个引脚的最大输入电流 (I _{OL} per I/O pin)
	∑I _{OL}	160	mA	最大电流 (∑I _{OL})
总功率消耗	P _T	600	mW	-
存放温度	T _{STG}	-65~+150	°C	-

Table 7.1 Absolute Maximum Ratings

注意) 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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
工作电压	VDD	f _X =1, 4, 8, 16MHz Internal RC	2.2	-	5.5	V
工作温度	T _{OPR}	VDD=2.2~5.5V	-40	-	85	°C

Table 7.2 Recommended Operating Conditions

7.3 ADC参数

(TA=-40°C ~ +85°C, VDD=2.2V ~ 5.5V, VSS=0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
分辨率	-	-	--	12	-	bit
积分线性误差	INL	Analog Reference Voltage = 2.5V ~ 5.5V f _X = 8MHz	-	-	±4	LSB
微分线性误差	DNL		-	-	±1	
零点偏移误差	ZOE		-3	-	+7	
满量程误差	OA		-	-	±3	
转换时间	t _{CON}	-	-	60	-	Cycle
模拟输入电压	V _{AN}	-	VSS	-	VDD	V
模拟参考电压	VDDREF	2.5V (NOTE1)	2.2	-	VDD	V
	AVREF	2.7V (NOTE2)	2.7	-	-	
模拟输入漏电流	I _{AN}	VDDREF=5.12V	-	-	2	uA
ADC 工作电流	I _{ADC}	Enable	-	1	2	mA
		Disable	-	-	0.1	uA

Table 7.3 A/D Converter Characteristics

NOTE1) 当ADC的参考电压低于 2.5V 时，ADC的精度会变差

NOTE2) 当ADC的参考电压低于 2.7V 时，ADC的精度会变差

7.4 上电复位特性

($T_A=-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD}=2.2 \sim 5.5\text{V}$, $V_{SS}=0\text{V}$)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
复位电平	V_{POR}	-	0.9	1.1	1.3	V
VDD 电压上升时间	t_R	-	0.05	-	5	V/ms
POR 电流	I_{POR}	-	-	0.1	-	μA

Table 7.4 Power-On Reset Characteristics

7.5 低电压复位和低电压检测特性

($T_A=-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD}=5.0\text{V}$, $V_{SS}=0\text{V}$)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit	
检测电平	V_{LVR} V_{LVI}	The LVR can select all levels but LVI can select other levels except 1.80V	-	1.80	1.95	V	
			1.6	2.1	2.6		
			1.95	2.5	3.05		
			2.9	3.5	4.1		
滞后	ΔV	-	-	50	-	mV	
最小脉冲宽度	t_{LW}	-	-	500	-	μs	
LVR 和 LVI 电流	I_{LVR}	LVR 1.80V	VDD=5V	-	1	-	μA
		LVR/LVI except 1.80V		-	-	50	

Table 7.5 LVR and LVI Characteristics

7.6 内部高速 RC 振荡特性

($T_A=-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD}=2.2\text{V} \sim 5.5\text{V}$, $V_{SS}=0\text{V}$)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit	
频率	f_{IRC}	$V_{DD} = 2.2 \sim 5.5\text{V}$	-	32	-	MHz	
误差	-	$T_A = 25^{\circ}\text{C}$	With 0.1 μF Bypass capacitor	-	-	± 3.0	%
		$T_A = -40^{\circ}\text{C} \text{ to } +85^{\circ}\text{C}$		-	-	± 3.0	
稳定时间	T_{HFS}	-	-	1	-	ms	
IRC电流	I_{IRC}	Enable	-	0.4	-	mA	

Table 7.6 Internal RC Oscillator Characteristics

注意) VDD 和 VSS之间需要连接一个0.1 μF 旁路电容器

7.7 内部 WDT 振荡器特性

($T_A=-40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD}=2.2\text{V} \sim 5.5\text{V}$, $V_{SS}=0\text{V}$)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Frequency	f_{WDTRC}	-	4	8	12	kHz
Stabilization Time	t_{WDTS}	-	-	1	-	ms
WDTRC Current	I_{WDTRC}	Enable	-	5	-	μA
		Disable	-	-	0.1	

Table 7.7 Internal WDT Oscillator Characteristics

7.8 DC 特性

($T_A = -40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD} = 2.2\text{V} \sim 5.5\text{V}$, $V_{SS} = 0\text{V}$, $f_x = 8.0\text{MHz}$)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
输入高电压	V_{IH1}	P0, P1	0.8VDD	-	VDD	V
输入低电压	V_{IL1}	P0, P1	-	-	0.2VDD	V
输出高电压	V_{OH1}	VDD=3.3V, $I_{OH} = -5\text{mA}$, All output ports	VDD-1.5	-	-	V
	V_{OH2}	VDD=5V, $I_{OH} = -10\text{mA}$, All output ports	VDD-1.5	-	-	V
输出低电压	V_{OL}	$I_{OL} = 20\text{mA}$, All output ports	-	-	1.0	V
输入高漏电流	I_{IH}	All input ports	-1	-	1	μA
输入低漏电流	I_{IL}	All input ports	-1	-	1	μA
上拉电阻	R_{PU1}	$V_I = 0\text{V}$, $T_A = 25^{\circ}\text{C}$ All Input ports	25	50	75	$\text{k}\Omega$
工作电流	$I_{DD1}(\text{RUN})$	Run Mode, $f_x = 8\text{MHz}$	-	3	5	mA
	$I_{DD2}(\text{IDLE})$	IDLE Mode, $f_x = 8\text{MHz}$	-	2	4	mA
	$I_{DD3}(\text{STOP1})$	STOP1 Mode, WDTRCEnable, LVR ON	-	5	10	μA
	$I_{DD4}(\text{STOP2})$	STOP2 Mode, WDTRCDisable, LVR ON	-	2.5	5	μA
	$I_{DD5}(\text{STOPL})$	STOP2 Mode, WDTRCDisable, LVR OFF	-	1	-	μA

Table 7.8 DC Characteristics

注意) STOP1: WDT继续运行, STOP2: 所有功能都停止

7.9 AC 特性

($T_A = -40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD} = 2.2\text{V} \sim 5.5\text{V}$)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
复位输入低电压宽度	t_{RST}	Input, VDD=5V	-	500	-	μs
中断输入高低宽度	t_{WH} , t_{WL}	All interrupt, VDD=5V	125	-	-	ns
外部计数器输入高低脉冲宽度,	t_{ECWH} , t_{ECWL}	ECn, VDD=5V (n=0, 1)	125	-	-	ns
外部计数器转换时间	t_{REC} , t_{FEC}	ECn, VDD=5V (n=0, 1)	-	-	20	ns

Table 7.9 AC Characteristics

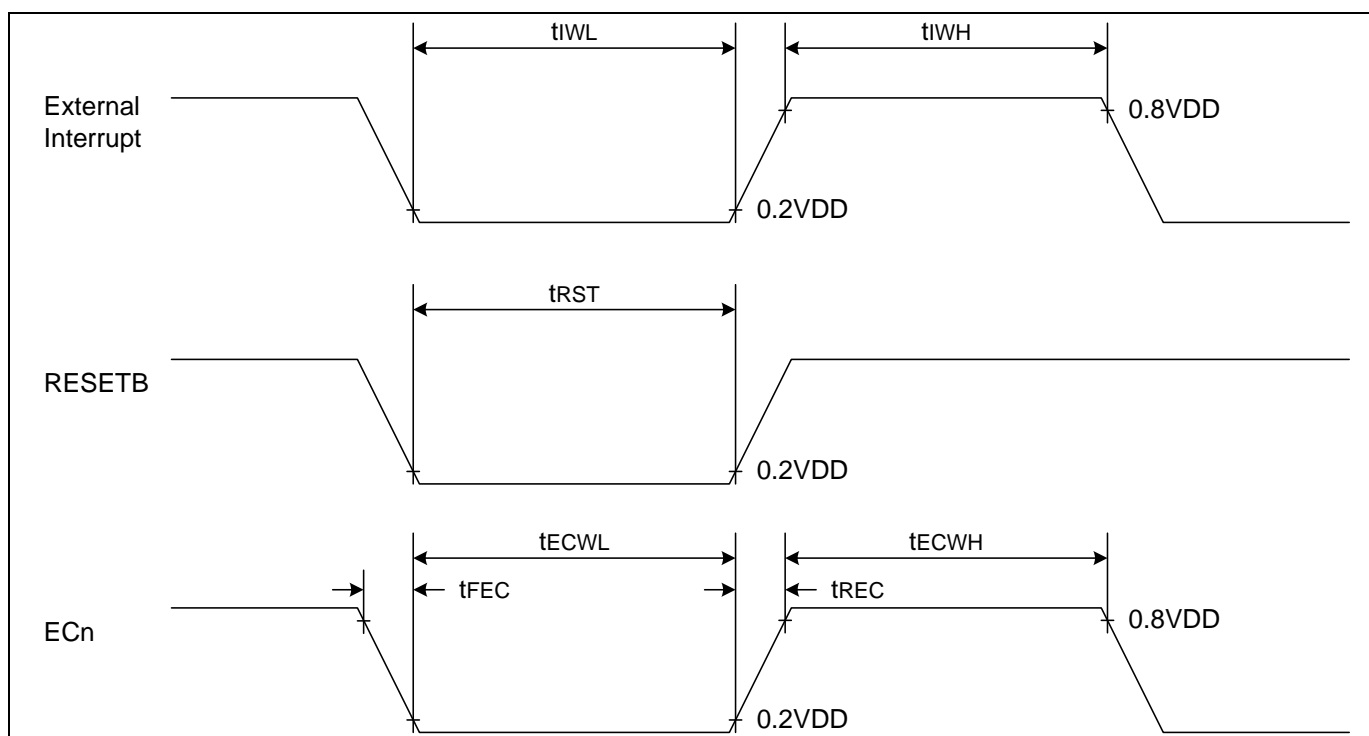


Figure 7.1 AC Timing

7.10 工作电压范围

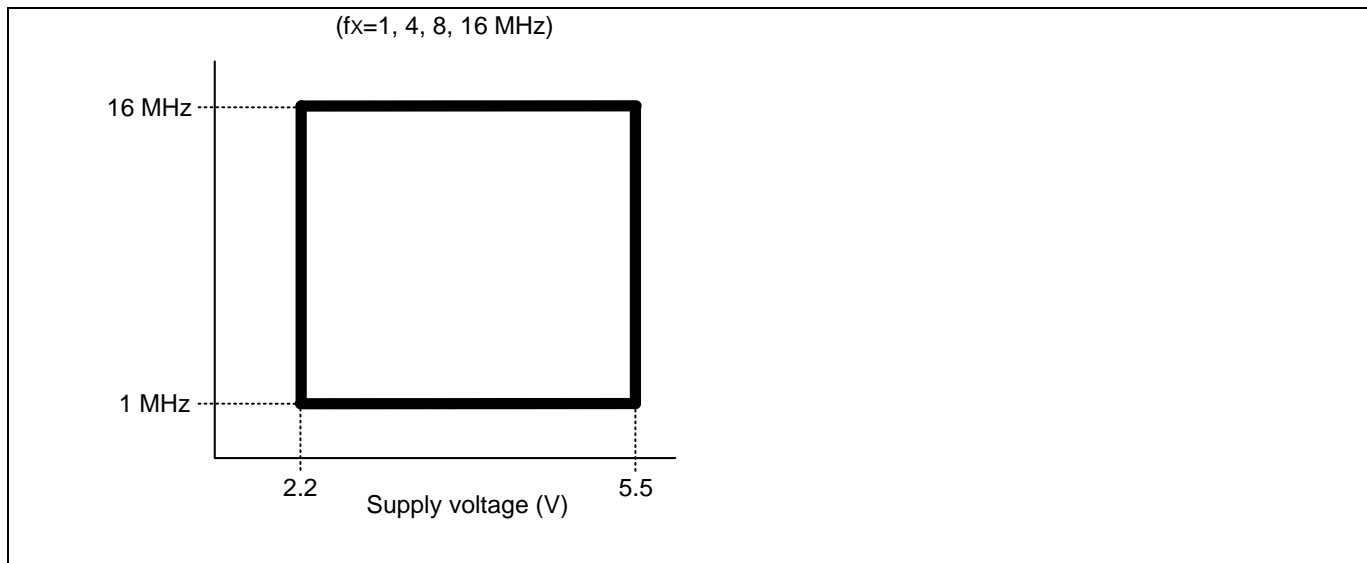


Figure 7.2 Operating Voltage Range

7.11 典型特征

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。

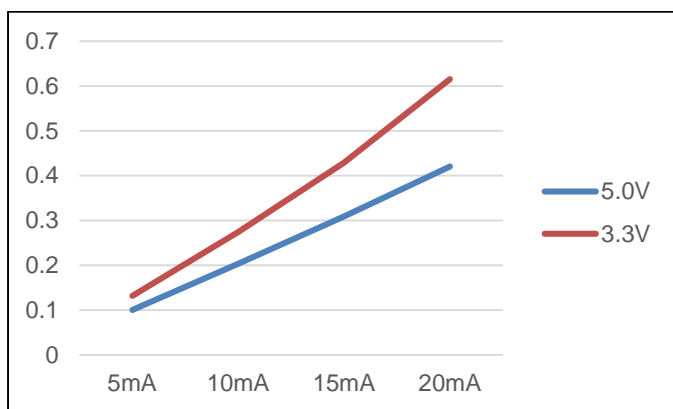


Figure 7.3 Output Low Voltage (V_{OL})

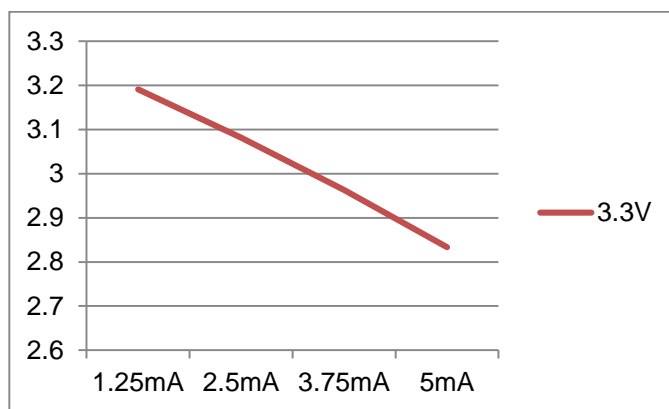


Figure 7.4 Output High Voltage (V_{OL1})

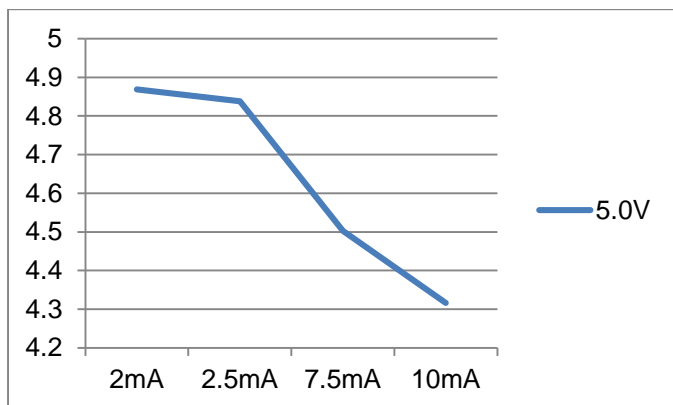


Figure 7.5 Output High Voltage (V_{OL2})

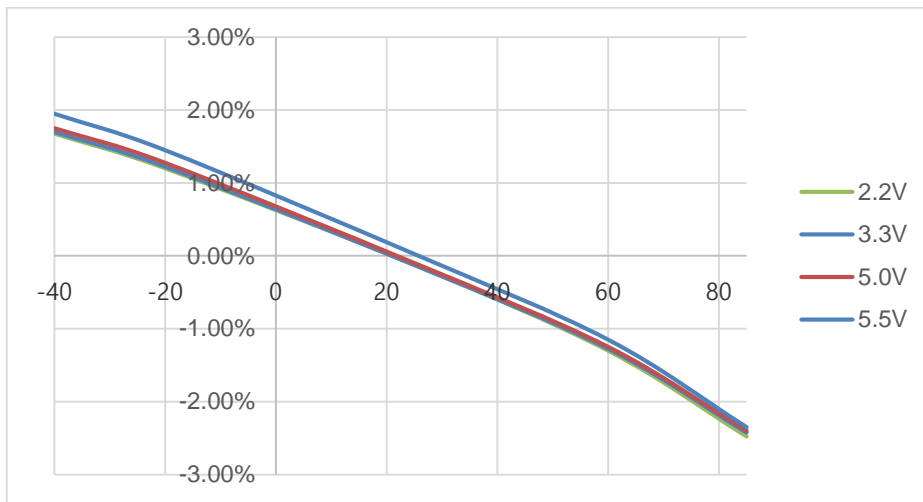


Figure 7.6 IRC Tolerance

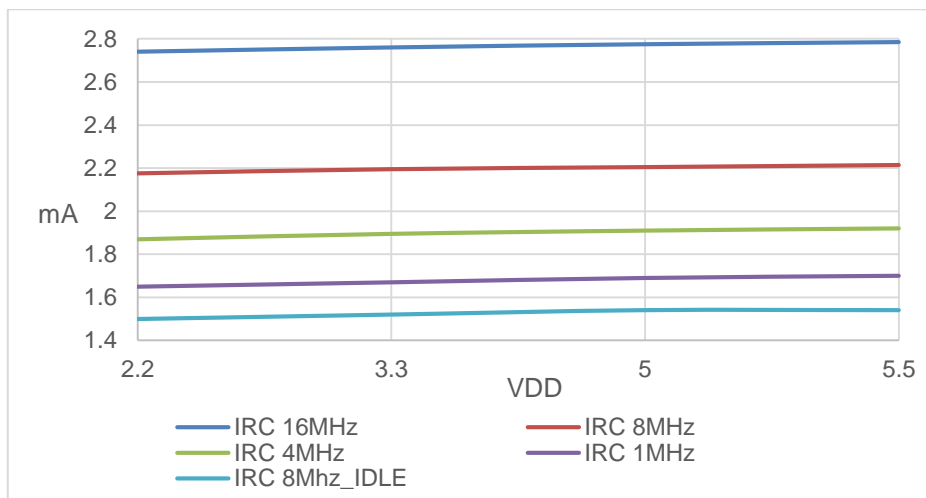


Figure 7.7 Power Supply Current (RUN, IDLE)

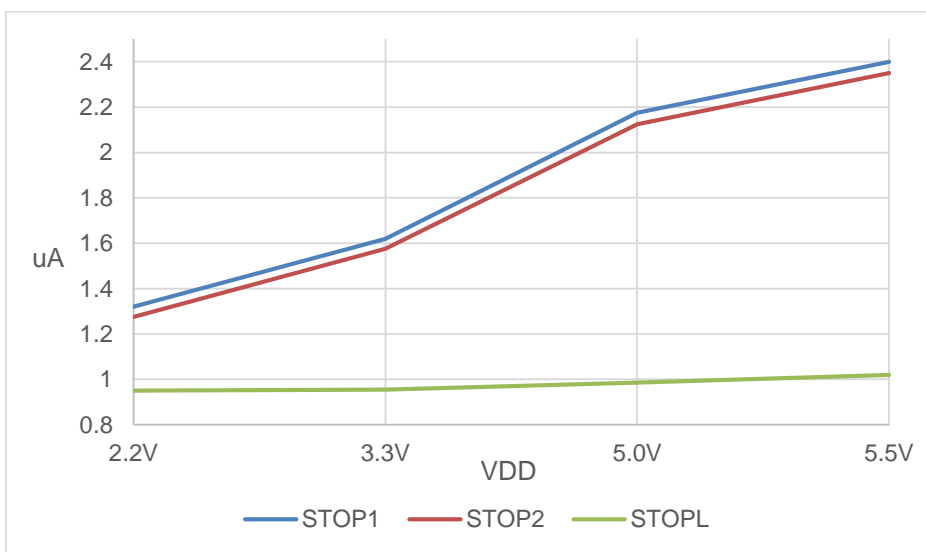


Figure 7.8 Power Supply Current (STOP1, STOP2, STOPL)

8 存储器

MC94F1202A具有两个独立的地址存储空间：程序存储器和数据存储器。程序存储器和数据存储器在逻辑上是分开的，可以通过8位地址空间来访问数据存储器，这样8位CPU可以更快速地访问存储器。也可以通过DPTR寄存器使用16位地址空间访问数据存储器。

MC94F1202A具有2K字节只能读取不能写入的flash程序存储器。还有256字节可读可写的数据存储器，堆栈已包含在内。

8.1 程序存储器

16-bit 程序计数器最大寻址空间可达 64k字节，此芯片只有2k字节的程序存储空间。

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入口地址。中断发生时，CPU会跳到该中断的入口地址，然后执行中断服务程序。例如外部中断0的入口地址为0003H，如果要使用外部中断0，它的服务程序必须从地址0003H开始。如果不使用外部中断0，它的服务地址作为普通程序存储地址使用。

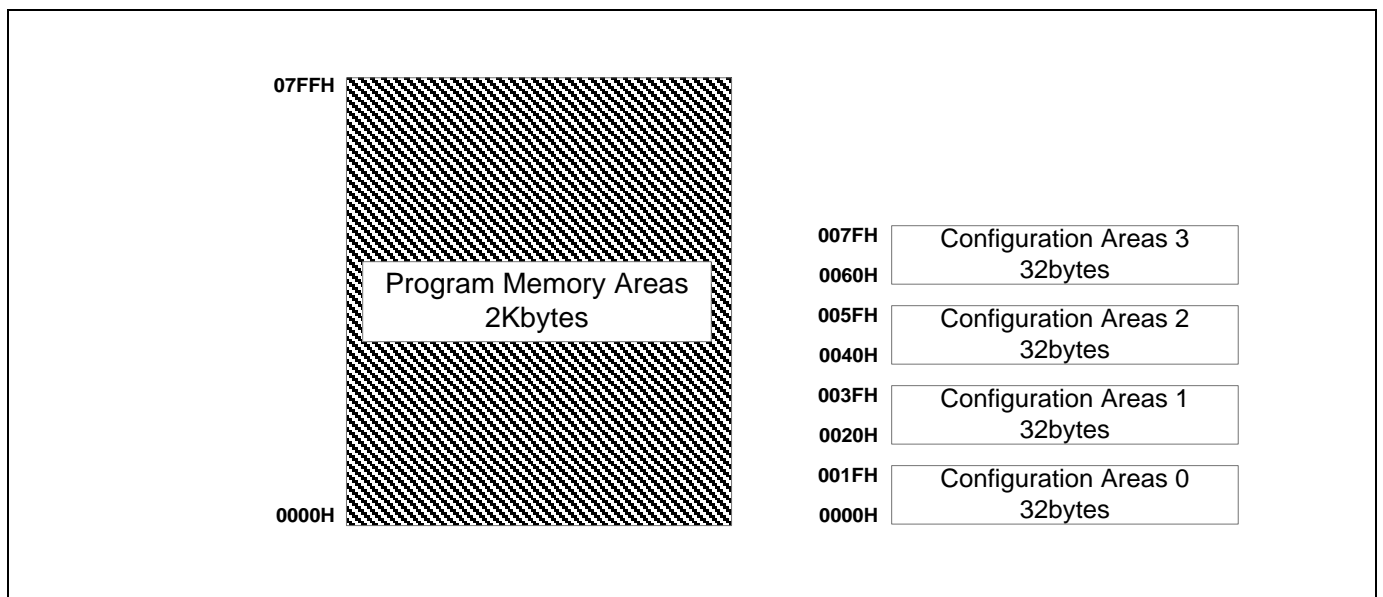


Figure 8.1 Program Memory

- 用户功能模式：2K字节里包含了中断向量区域
- 非易失性和可重复编程存储器：基于EEPROM单元的Flash存储器

8.2 数据存储

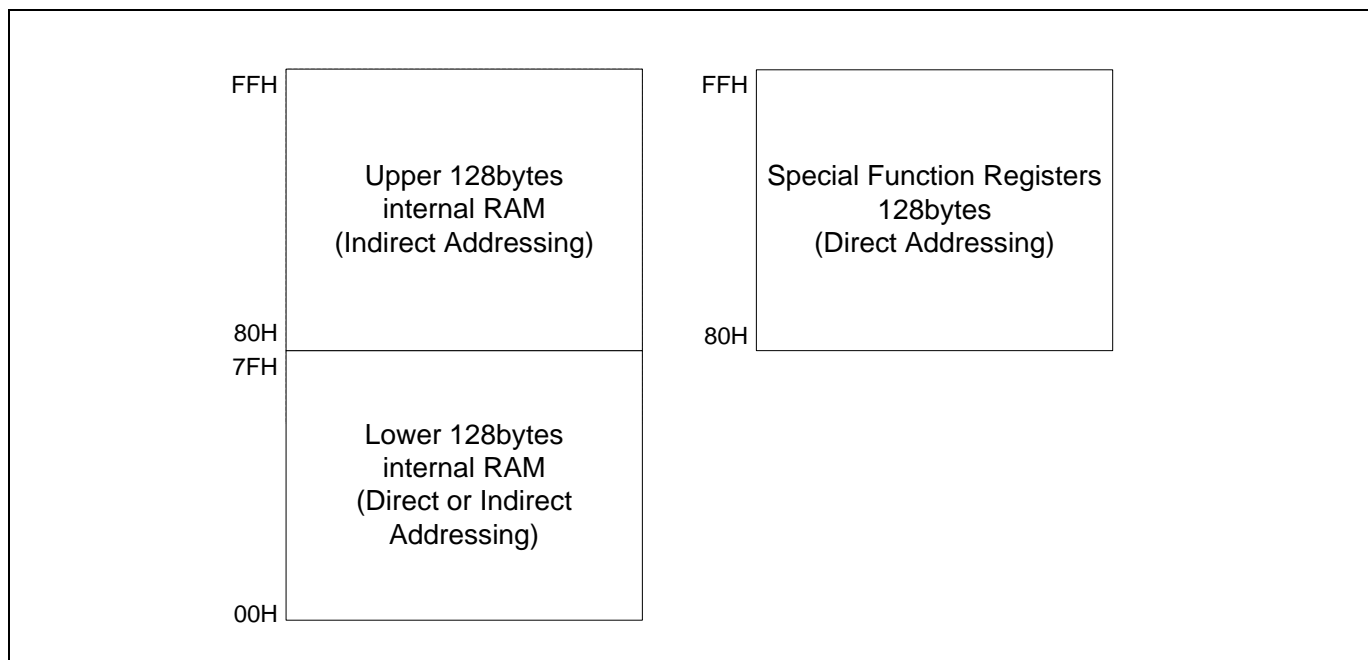


Figure 8.2 Data Memory Map

内部数据存储器分为3个部分，低 128字节，高128字节，SFR

内部数据存储的地址只有8位，因此寻址空间只有 256字节，但是，通过不同的内部RAM寻址方式可寻址384字节的存储空间。直接寻址高于 7FH 的存储空间和间接寻址高于7FH 的存储空间会进入到不同的存储空间。图8-2所示高128字节和SFR的地址都是80H 到FFH，但是他们却是两个不同的独立的空间。

Figure 8.3.所示，所有的8051单片机RAM空间的低128字节（地址00H~7FH区域）中的最低32字节会组成4个程序可以调用的寄存器组（R0~R7），由程序状态字中的两位来选择用哪个寄存器组。这样代码空间可以更小，因为寄存器指令比直接寻址指令更短。

寄存器组后面的16字节是可以位寻址的存储空间。8051指令集包含对单个位操作的位指令，位寻址区域里的128位都可以通过这些位指令直接寻址。位寻址区包含在地址空间为00H~7FH的RAM里。

RAM空间的低128字节都能通过直接或间接寻址来访问。高128字节则只能通过间接寻址来访问。这些RAM空间可用作数据存储或堆栈。

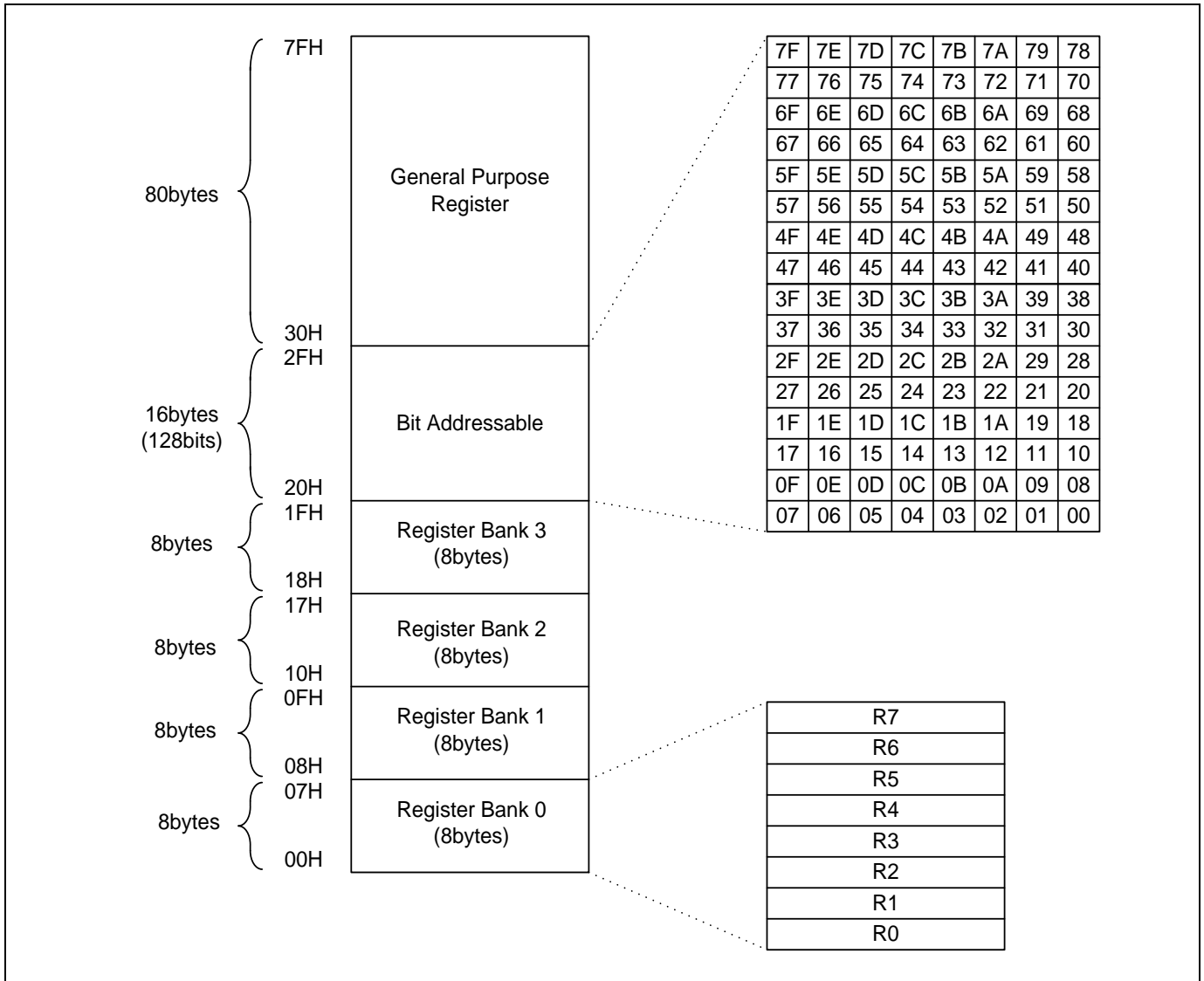


Figure 8.3 Lower 128bytes RAM

8.3 SFR

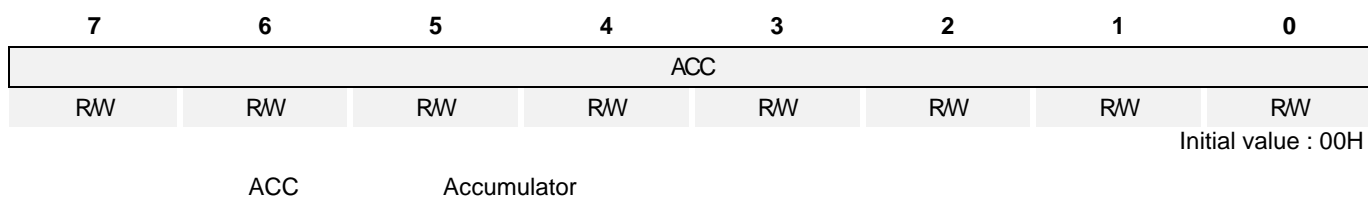
	0H/8H	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
F8H								
F0H	B	FEMR	FECR	FESR	FETCR	FEARL	FEARM	FEARH
E8H								
E0H	ACC							
D8H								
D0H	PSW							
C8H		P1IO	P1PU	P1OD	P1DB	P1FSRL	P1FSRH	
C0H		P0IO	P0PU	P0OD	P0DB	P0FSRL	P0FSRH	DBTSR
B8H			T1CR	T1CR1	PWM1DRL CDR1L / T1L	PWM1DRH CDR1H / T1H	PWM1PRL T1DRL	PWM1PRH T1DRH
B0H			T0CR	T0CR1	PWM0DRL CDR0L / T0L	PWM0DRH CDR0H / T0H	PWM0PRL T0DRL	PWM0PRH T0DRH
A8H	IRQ1	IP1	ILVL					
A0H	IRQ0	IP	EO		T2CR		T2	T2DR
98H	IE1	EIEDGEN	IOFFSET			WDTMR	WDTR/WDTC R	
90H	IE	EIEDGEP	PCI			ADCM	ADCM1 / ADCRL	ADCRH
88H	P1		SCCR			SYSCON_AR		LVIR
80H	P0	SP	DPL	DPH			RSFR	PCON

Table 8.1 SFR Map Summary

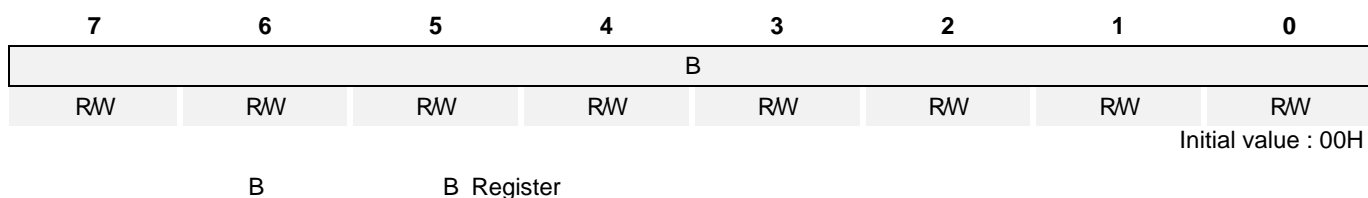
注意：0H/8H，这一列的寄存器支持位操作

8.3.1 8051编译器兼容SFR

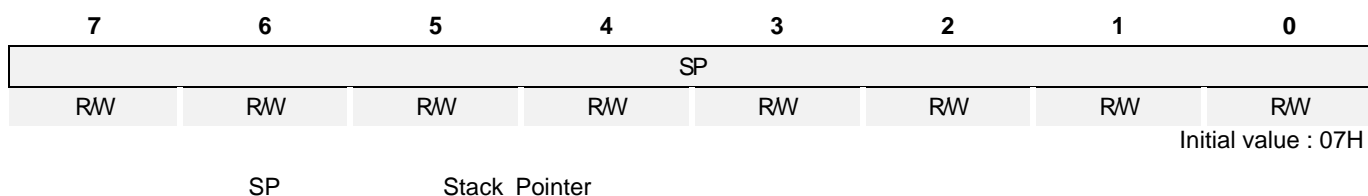
ACC (Accumulator) : E0H



B (B Register) : F0H



SP (Stack Pointer) : 81H



DPL (Data Pointer Low Byte) : 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 Byte

DPH (Data Pointer High Byte) : 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 Byte

PSW (Program Status Word) : D0H

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

Initial value : 00H

- CY Carry Flag
- AC Auxiliary Carry Flag
- F0 General Purpose User-Definable Flag
- RS1 Register Bank Select bit 1
- RS0 Register Bank Select bit 0
- OV Overflow Flag
- F1 User-Definable Flag
- P Parity Flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bits in the accumulator

EO (Extended Operation Register) : A2H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	DPSEL0
R	R	R	RW	R	R	R	RW

Initial value : 00H

- DPSEL Select Banked Data Point Register
- 0 DPTR = {DPH, DPL}
- 1 DPTR1 = {DPH1, DPL1}

9 I/O端口

MC94F1202A有14个I/O口（P0和P1）。通过软件可以设置每个引脚的I/O、内部上拉和漏极开路等，以匹配不同的系统结构和设计要求。P1还有根据引脚电平状态变化而产生中断的功能。

9.1 端口寄存器

9.1.1 数据寄存器(PxDA)

Px是双向寄存器，在输出状态，通过Px把值写入到相应引脚。在输入状态，通过Px读取相应引脚的值

9.1.2 方向寄存器(PxIO)

PxIO可以单独设置每个I/O口的输入输出状态，PxIO的每一位对应一个引脚，1为输出，0为输入。复位后全部清零

9.1.3 上拉电阻选择寄存器(PxPU)

PxPU可以单独设置每个I/O口的内部上拉电阻，PxPU的每一位对应一个引脚，1为使能，0为禁止。复位后全部清零

9.1.4 漏极开路选择寄存器(PxOD)

PxOD可以单独设置每个I/O口的漏极开路功能，PxOD的每一位对应一个引脚，1为使能，0为关闭。复位后全部清零

9.1.5 消抖使能寄存器(PxDB)

P0~P1具有消抖功能。这个寄存器可以设置每一个引脚的消抖功能的开关。

9.1.6 消抖时间选择寄存器(DBTSR)

设置所有引脚以及外部复位的消抖时间。

9.1.7 引脚电平变化中断使能寄存器(PCI)

P1每个引脚都具有电平变化中断功能。PCI可以单独设置P1每个引脚的电平变化中断，PCI的每一位对应一个引脚，1为使能，0为关闭。复位后全部清零

9.1.8 引脚功能选择寄存器(PxFSR)

如果PxFSR配置为子功能，相应引脚的传输方向将会改变，输入输出的数据将会传输到子功能模块。如果PxFSR配置为ADC输入，相应引脚的模拟开关会打开，数字输入将禁止。

9.2 Px端口

9.2.1 Px 端口说明

Px是7位I/O端口。Px控制寄存器由数据寄存器(Px)、方向寄存器(PxIO)、上拉电阻选择寄存器(PxPU)、漏极开路选择寄存器(PxOD)、消抖使能寄存器(PxDB)、消抖时间选择寄存器(DBTSR)、引脚电平变化中断寄存器(PCI)、功能选择寄存器(PxFSRH/ PxF SRL)组成。

9.2.2 寄存器图

Name	Address	Dir	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	C1H	R/W	00H	P0 Direction Register
P0PU	C2H	R/W	00H	P0 Pull-up Resistor Selection Register
P0OD	C3H	R/W	00H	P0 Open-drain Selection Register
P0DB	C4H	R/W	00H	P0 DebounceEnable Register
P1	88H	R/W	00H	P1 Data Register
P1IO	C9H	R/W	00H	P1 Direction Register
P1PU	CAH	R/W	00H	P1 Open-drain Selection Register
P1OD	CBH	R/W	00H	P1 Open-drain Selection Register
P1DB	CCH	R/W	00H	P1 DebounceEnable Register
DBTSR	C7H	R/W	00H	De-bounce time selection register
PCI	92H	R/W	00H	Pin change interrupt enable register
P0FSRL	C5H	R/W	00H	P0 Function Selection Low Register
P0FSRH	C6H	R/W	00H	P0 Function Selection High Register
P1FSRL	CDH	R/W	00H	P1 Function Selection Low Register
P1FSRH	CEH	R/W	00H	P1 Function Selection High Register

Table 9.1 Register Map

9.2.3 P0寄存器说明

P0 (P0数据寄存器) : 80H

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

Initial value : 00H

P0[6:0] I/O Data

P0IO (P0方向寄存器) : C1H

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

Initial value : 00H

P0IO[6:0] P0 data I/O direction.
 0 Input
 1 Output

P0PU (P0上拉电阻选择寄存器): C2H

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

Initial value : 00H

P0PU[6:0] Configure pull-up resistor of P0 port
 0 Disable
 1 Enable

P0OD (P0漏极开路选择寄存器) : C3H

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

Initial value : 00H

P0OD[6:0] Configure open-drain of P0 port
 0 Disable
 1 Enable

P0DB (P0消抖使能寄存器) : C4H

7	6	5	4	3	2	1	0
-	P06DB	P05DB	P04DB	P03DB	P02DB	P01DB	P00DB
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0DB[6:0] Configure debounce of P0 port
 0 Disable
 1 Enable

9.2.4 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	P10
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1[6:0] I/O Data

P1IO (P1方向寄存器) : C9H

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

Initial value : 00H

P1IO[6:0] P1 data I/O direction.
 0 Input
 1 Output

P1PU (P1上拉电阻选择寄存器): CAH

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

Initial value : 00H

P1PU[6:0] Configure pull-up resistor of P1 port
 0 Disable
 1 Enable

P1OD (P1漏极开路选择寄存器) : CBH

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

Initial value : 00H

P1OD[6:0] Configure open-drain of P1 port
 0 Disable
 1 Enable

P1DB (P1消抖使能寄存器) : CCH

7	6	5	4	3	2	1	0
-	P16DB	P15DB	P14DB	P13DB	P12DB	P11DB	P10DB
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1DB[6:0] Configure debounce of P1 port
 0 Disable
 1 Enable

DBTSR (消抖时间选择寄存器) : C7H

7	6	5	4	3	2	1	0
-	-	-	-	RDB[1]	RDB[0]	PDB[1]	PDB[0]
-	-	-	-	RW	RW	RW	RW

Initial value : 00H

RDB[1:0] 外部复位消抖时间选择
 00 10 us
 01 20 us
 10 40 us
 11 80 us

PDB[1:0] 引脚消抖时间选择
 00 1.2 us
 01 2.5 us
 10 5 us
 11 10 us

PCI (引脚电平变化中断寄存器) : 92H

7	6	5	4	3	2	1	0
-	PCI6	PCI5	PCI4	PCI3	PCI2	PCI1	PCI0
-	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

PCI[6:0] Select PCI interrupt enable or disable of Px
 0 Disable
 1 Enable

P0FSRH (P0功能选择高字节寄存器) : C6H

7	6	5	4	3	2	1	0
-	-	P0FSRH5	P0FSRH4	P0FSRH3	P0FSRH2	P0FSRH1	P0FSRH0
-	-	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0FSRH[5:4]	P06 Function Select		
	0	0	P06
	0	1	P06
	1	0	P06
	1	1	AN6
P0FSRH[3:2]	P05 Function Select		
	0	0	P05
	0	1	P05
	1	0	P05
	1	1	AN5
P0FSRH[1:0]	P04 Function Select		
	0	0	P04
	0	1	P04
	1	0	P04
	1	1	AN4

P0FSRL (P0功能选择低字节寄存器) : C5H

7	6	5	4	3	2	1	0
P0FSRL7	P0FSRL6	P0FSRL5	P0FSRL4	P0FSRL3	P0FSRL2	P0FSRL1	P0FSRL0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P0FSRL[7:6]	P03 Function Select		
	0	0	P03
	0	1	P03
	1	0	P03
	1	1	AN3
P0FSRL[5:4]	P02 Function Select		
	0	0	P02
	0	1	INT0
	1	0	P02
	1	1	AN2
P0FSRL[3:2]	P01 Function Select		
	0	0	P01
	0	1	P01
	1	0	P01
	1	1	AN1
P0FSRL[1:0]	P00 Function Select		
	0	0	P00
	0	1	P00
	1	0	EC0
	1	1	AN0

P1FSRH (P1功能选择高字节寄存器) : CEH

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 Function Select		
	0	0	P16
	0	1	INT1
	1	0	EC1
	1	1	AN13 or AVREF
P1FSRH[3:2]	P15 Function Select		
	0	0	P15
	0	1	P15
	1	0	P15
	1	1	AN12
P1FSRH[1:0]	P14 Function Select		
	0	0	P14
	0	1	P14
	1	0	PWM0
	1	1	AN11

P1FSRL (P1功能选择低字节寄存器) : CDH

7	6	5	4	3	2	1	0
P1FSRL7	P1FSRL6	P1FSRL5	P1FSRL4	P1FSRL3	P1FSRL2	P1FSRL1	P1FSRL0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1FSRL[7:6]	P13 Function Select		
	0	0	P13
	0	1	P13
	1	0	PWM1
	1	1	AN10
P1FSRL[5:4]	P12 Function Select		
	0	0	P12
	0	1	P12
	1	0	P12
	1	1	AN9
P1FSRL[3:2]	P11 Function Select		
	0	0	P11
	0	1	P11
	1	0	P11
	1	1	AN8
P1FSRL[1:0]	P10 Function Select		
	0	0	P10
	0	1	P10
	1	0	P10
	1	1	AN7

10 中断控制器

10.1 概述

MC94F1202A有9个中断源。各中断具有各自独立的使能寄存器控制位，允许软件控制。具有可选的2级中断优先级，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，且不可通过软件屏蔽。中断控制器有以下特性：

- 9 个中断源接收中断
- 2 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5 ~ 8 个机器周期
- 中断服务程序支持4级嵌套

非可屏蔽中断是经常用到的，可屏蔽中断通过2个中断使能寄存器（IE、IE1）使能。IE、IE1的每一位都能单独地使能/禁止相应的中断源。总中断控制位是 IE 的第7位（EA）。EA 清零，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。MC94F1202A支持2个优先级。每个可屏蔽中断通过IP0、IP1来分配中断优先等级。

Figure 10.1是中断优先级的示意图，MC94F1202A的中断具有基本优先顺序，下图的水平轴。也可以通过IP0和IP1寄存器来设置中断的优先级。IP0与IP1的每一位对应每个中断的2个优先级之一。较高优先级的中断服务程序首先执行，如果有2个中断同时触发，则会先响应优先级较高的中断，然后再响应优先级较低的中断。

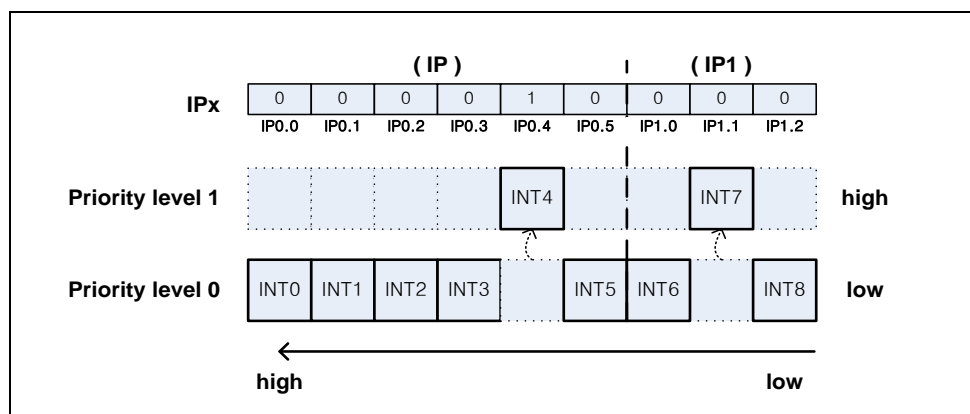


Figure 10.1 Interrupt Priority Level

中断服务程序支持4级嵌套，中断嵌套层级寄存器（ILVL）可以指示当前的嵌套层级。如果当前的嵌套层级为4级，再触发更高优先级的中断可能会导致故障。

10.2 方框图

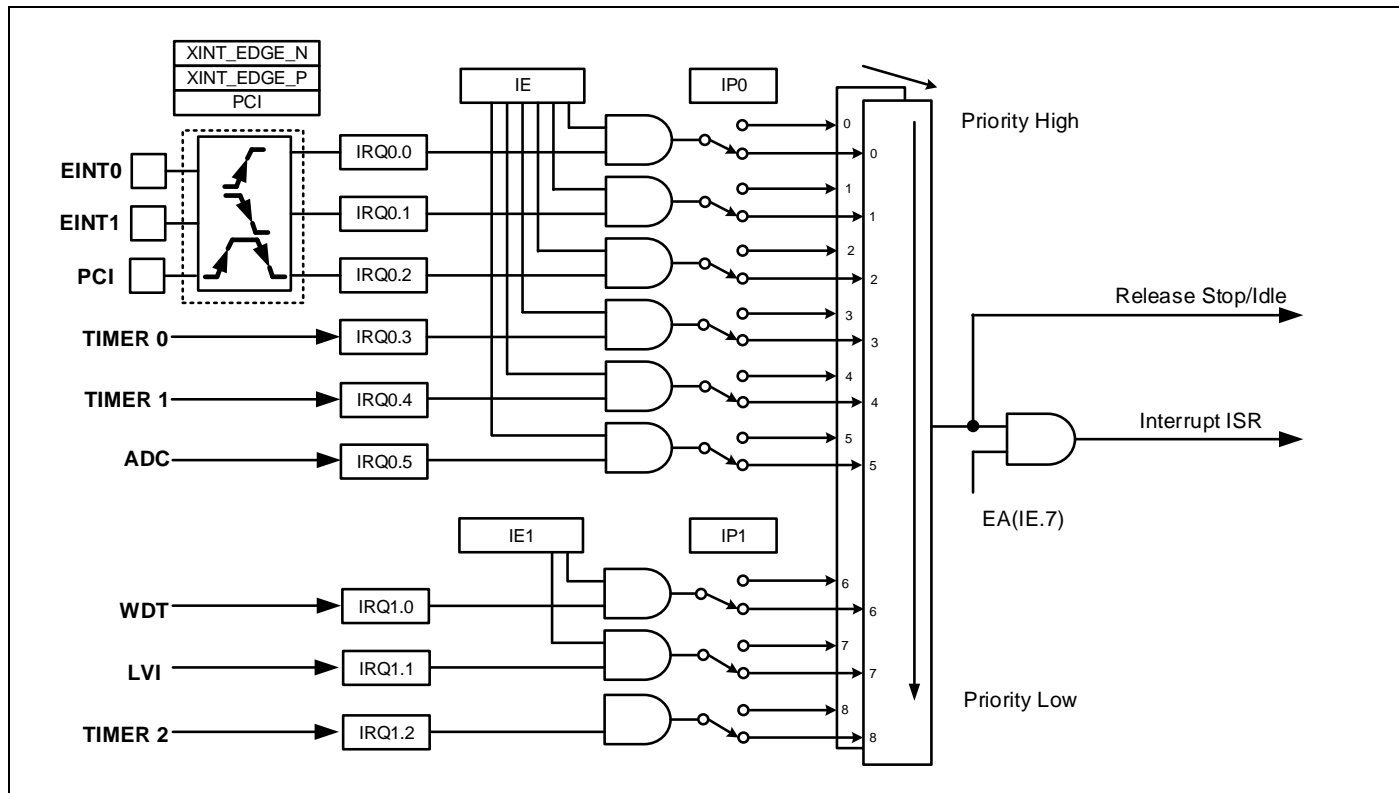


Figure 10.2 Block Diagram of Interrupt

10.3 中断向量表

中断控制器支持9个中断源，当中断被响应时，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。下表列出了全部中断源、优先级以及它的向量地址。下表列出了全部中断源、优先级以及它的向量地址。

Interrupt Source	Symbol	Interrupt Enable bit	Priority	Mask	Vector Address
Hardware Reset	RESETB	0	0	Non-Maskable	0000H
External Interrupt 0	INT0	IE.0	1	Maskable	0003H
External Interrupt 1	INT1	IE.1	2	Maskable	000BH
PCI	INT2	IE.2	3	Maskable	0013H
TIMER 0	INT3	IE.3	4	Maskable	001BH
TIMER 1	INT4	IE.4	5	Maskable	0023H
ADC	INT5	IE.5	6	Maskable	002BH
WDT	INT6	IE1.0	7	Maskable	0033H
LVI	INT7	IE1.1	8	Maskable	003BH
TIMER 2	INT8	IE1.2	9	Maskable	0043H

Table 10.1 Interrupt Vector Address Table

可屏蔽中断的执行，EA 必须置位，且 IEx 的相应位也要置位，以使能中断，如果接收到中断请求，相应的中断标志位会置位，中断请求标志位会一直保持置位状态，直至 CPU 响应中断。中断响应后，中断请求标志位会自动清零。

10.4 中断序列

中断请求会一直保留，直到响应中断或者中断锁存器被复位或软件清零。响应中断通常是在指令的最后一个周期执行，而不是当前的周期，CPU 执行内部 LCALL 指令并且保存 PC 到堆栈里。关于中断服务程序，中断控制器在将 LJMP 指令送到 CPU，CPU 在完成当前指令之后，还需要 3~9 个机器周期才能进入中断服务程序。中断服务程序结束后由中断返回指令 RETI 返回到进入中断服务程序前的指令的下一行。中断请求被响应后的流程如下图所示：

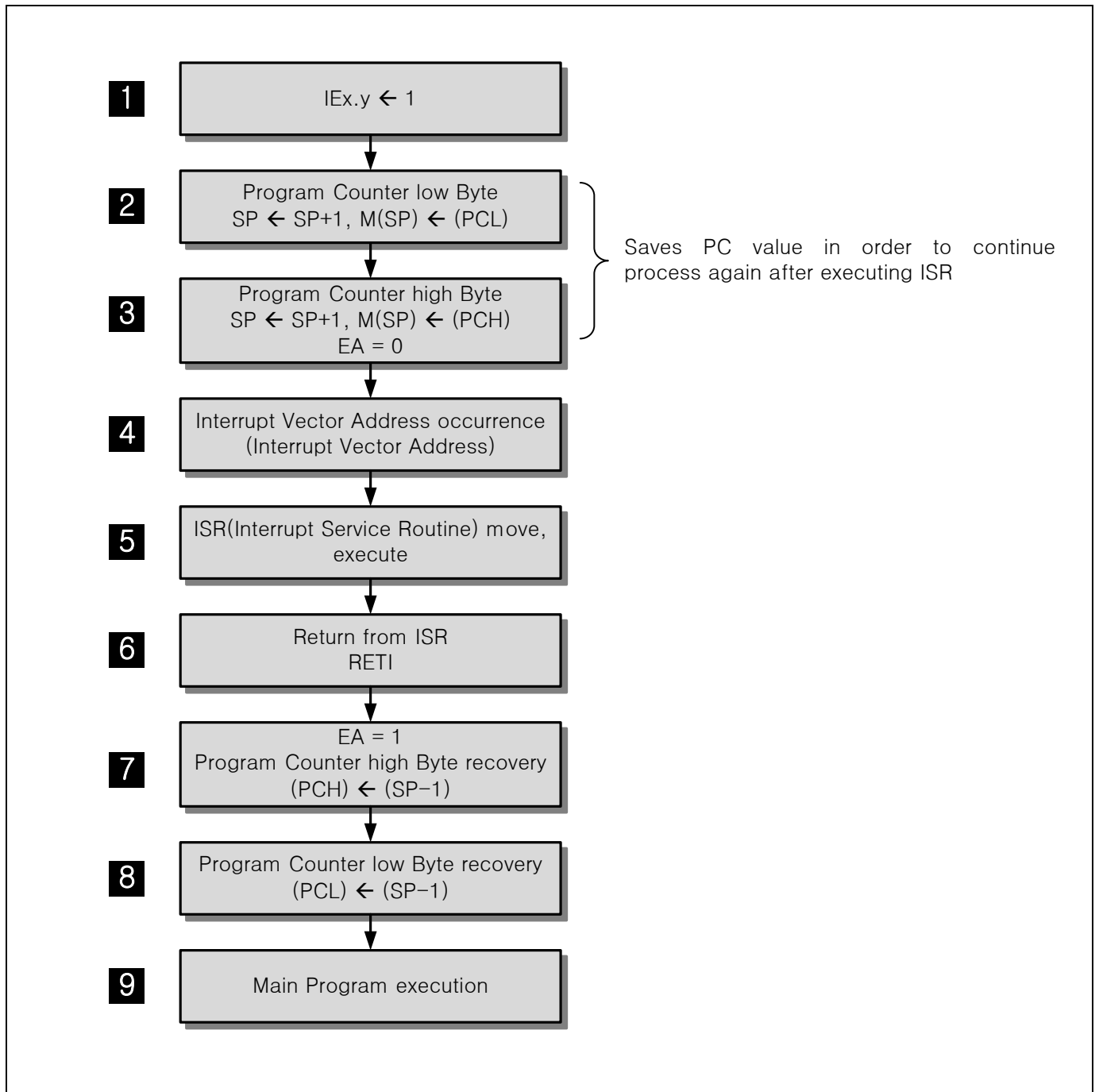


Figure 10.3 Interrupt Execution Flow

10.7 中断使能响应时序

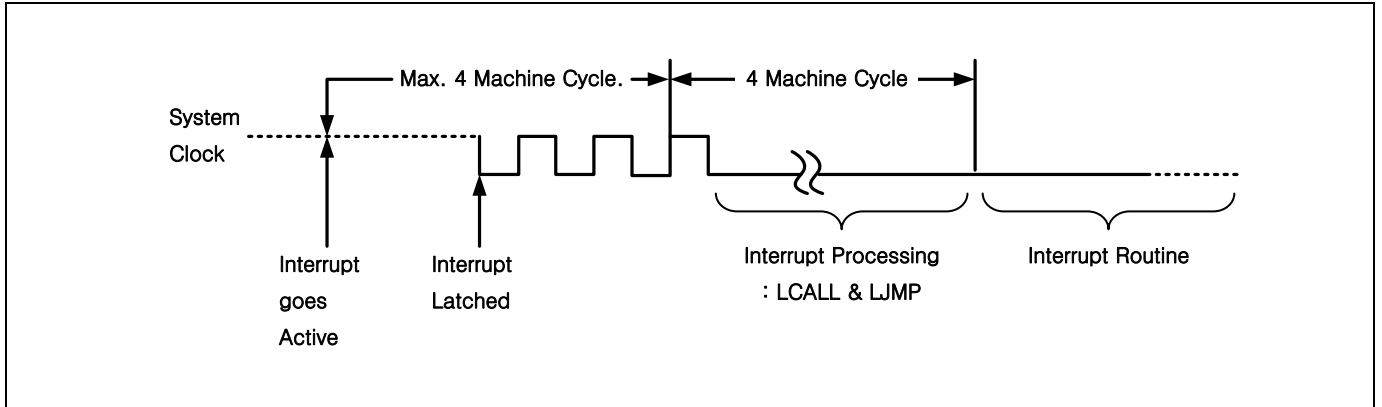


Figure 10.6 Interrupt Response Timing Diagram

10.8 中断服务程序地址

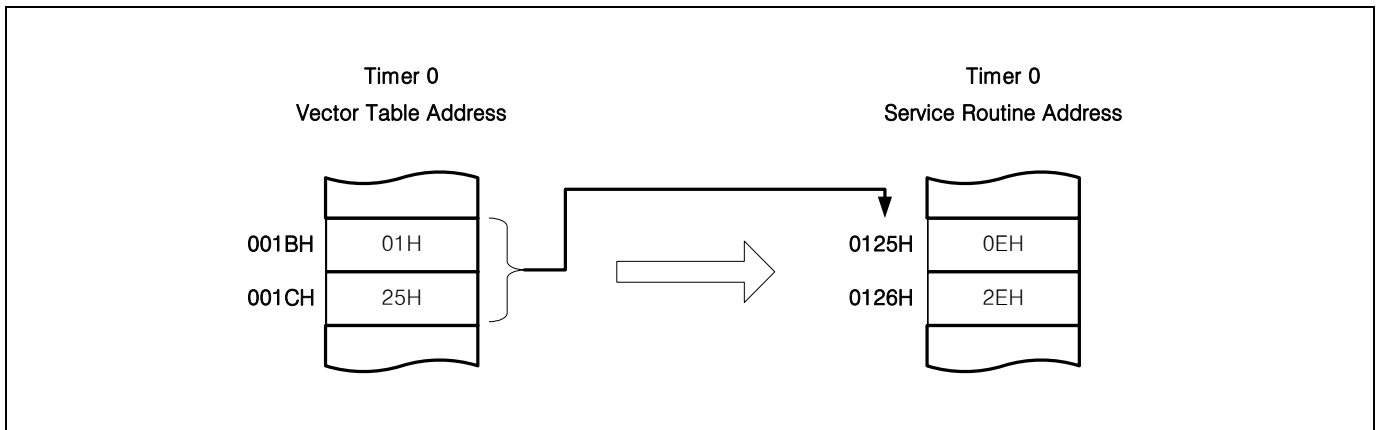


Figure 10.7 Correspondence between Vector Table Address and the Entry Address of ISR

10.9 通用寄存器的保存和恢复

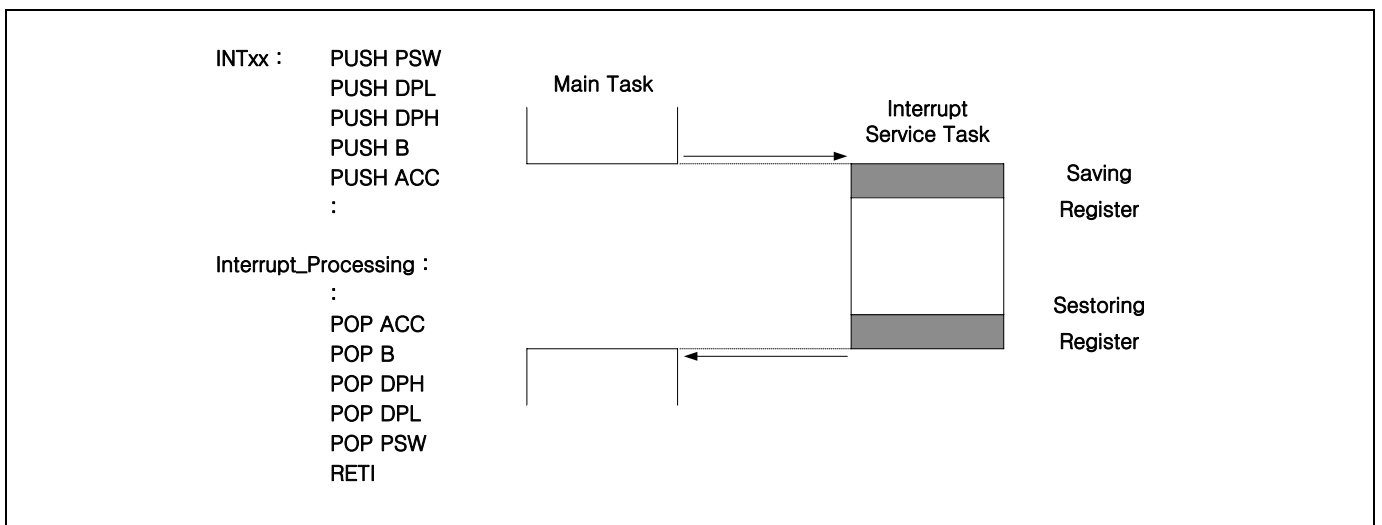


Figure 10.8 Saving/Restore Process Diagram and Sample Source

10.10 中断时序

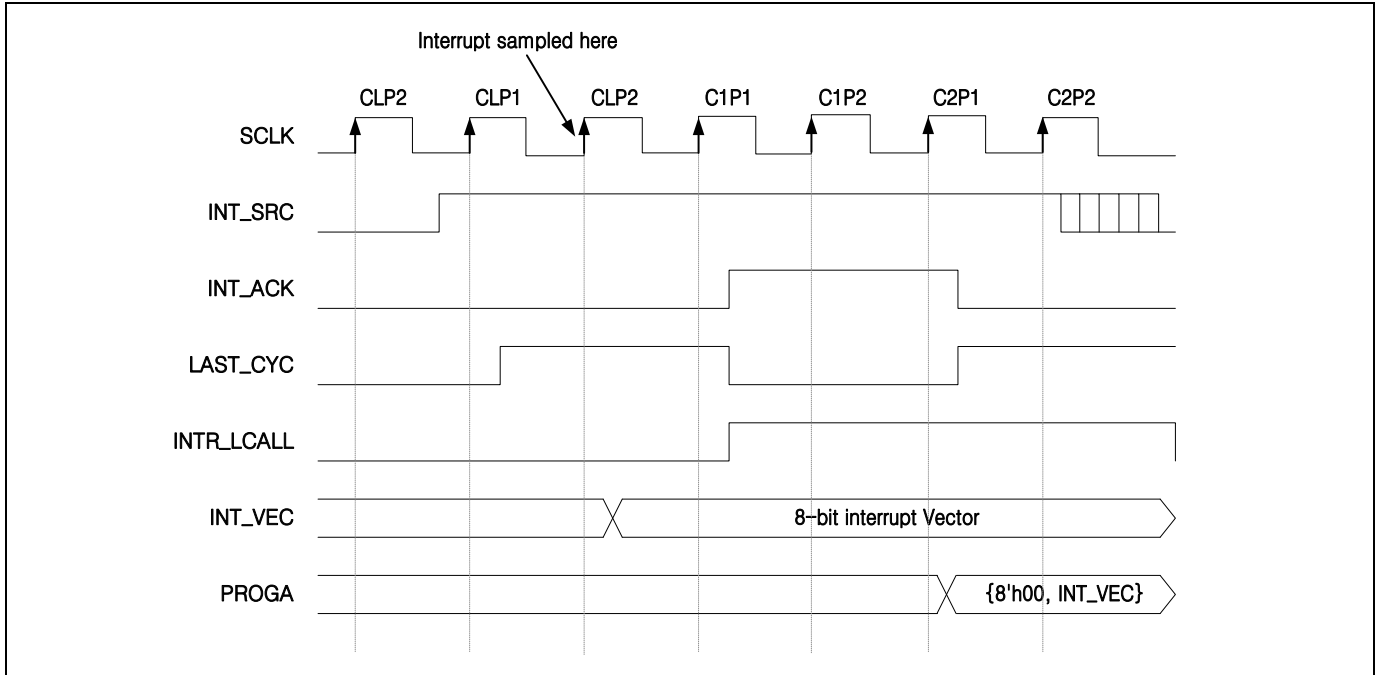


Figure 10.9 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

中断源在命令的最后一个周期采样。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，执行长调用指令跳到中断服务程序。

Note) command cycle CLPx: L=Last cycle, 1=1st cycle or 1st phase, 2=2nd cycle or 2nd phase

10.11 外部中断

外部中断是通过 INT0、INT1 引脚来接收中断请求的，外部中断由EINTx_EN（外部中断x使能寄存器）、EINT_EDGE_P/N（外部中断正/负边沿选择寄存器）来控制。见Figure 10.10

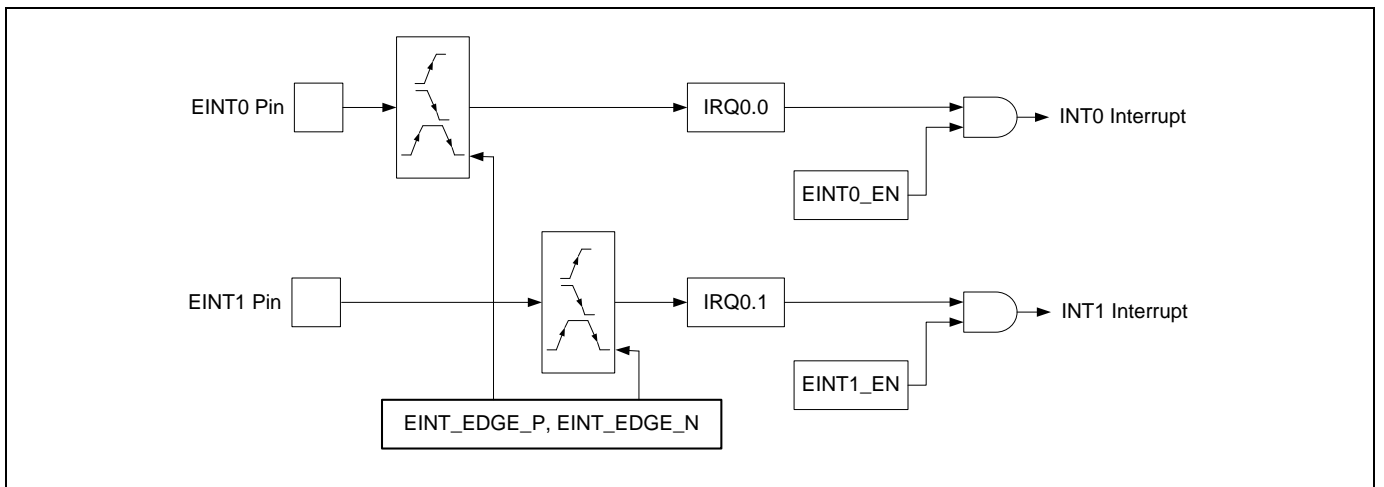


Figure 10.10 External Interrupt Description

10.12 中断寄存器概述

10.12.1 中断使能寄存器(IE, IE1)

中断使能寄存器由总中断位 (EA) 和外围各中断位组成。共有 9 个可控制中断

10.12.2 中断优先级寄存器(IP, IP1)

8个中断可以通过中断优先级寄存器配置为2个中断优先级。Level 1 优先级最高，level 0 优先级最低。复位后IP0、IP1 被清零。低序号中断有优先权。

10.12.3 中断请求寄存器 (IRQ0, IRQ1)

当外设(H/W)被事件触发时，会置位中断请求寄存器，如果该中断使能，则会触发中断请求。

10.12.4 中断偏移寄存器 (IOFFSET)

INT_OFFSET寄存器可以设置中断入口地址的偏移量。

10.12.5 外部中断正边沿选择寄存器(EINT_EDGE_P)

外部中断正边沿选择寄存器可以使能外部中断的正边沿触发。1为使能，0为禁止。初始值为0。

10.12.6 外部中断负边沿选择寄存器(EINT_EDGE_N)

外部中断负边沿选择寄存器可以使能外部中断的负边沿触发。1为使能，0为禁止。初始值为0。

10.12.7 中断嵌套等级寄存器(ILVL)

中断嵌套等级寄存器用于指示当前的中断嵌套层级。

10.12.8 寄存器图

Name	Address	Dir	Default	Description
IE	90H	R/W	00H	Interrupt Enable Register
IE1	98H	R/W	00H	Interrupt Enable Register 1
IRQ0	A0H	R/W	00H	Interrupt Request Register 0
IRQ1	A8H	R/W	00H	Interrupt Request Register 1
IP	A1H	R/W	00H	Interrupt Priority Register
IP1	A9H	R/W	00H	Interrupt Priority Register 1
IOFFSET	9AH	W	xxH	Interrupt Offset Register
EINT_EDGE_P	91H	R/W	00H	External Interrupt Positive Edge selection Register
EINT_EDGE_N	99H	R/W	00H	External Interrupt Negative Edge selection Register
ILVL	AAH	RW	00H	Interrupt nesting level Register

Table 10.2 Register Map

10.13 中断寄存器说明

中断寄存器用于控制中断功能。还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) 组成。外部中断寄存器由外部中断正/负边沿选择寄存器 (EINT_EDGE_P、EINT_EDGE_N) 组成。

IE (中断使能寄存器) : 90H

7	6	5	4	3	2	1	0
EA	-	ADC_INT_EN	T1_INT_EN	T0_INT_EN	PCI_EN	EINT1_EN	EINT0_EN
RW	-	RW	RW	RW	RW	RW	RW

EA	所有中断使能与禁止位	Initial value : 00H
0	所有中断禁止	
1	所有中断使能	
ADC_INT_EN	ADC中断使能位	
0	Disable	
1	Enable	
T1_INT_EN	Timer 0中断使能位	
0	Disable	
1	Enable	
T0_INT_EN	Timer 1中断使能位	
0	Disable	
1	Enable	
PCI_INT_EN	引脚电平变化中断使能位	
0	Disable	
1	Enable	
EINT1_EN	外部中断1中断使能位	
0	Disable	
1	Enable	
EINT0_EN	外部中断0中断使能位	
0	Disable	
1	Enable	

IE1 (中断使能寄存器1) : 98H

7	6	5	4	3	2	1	0
-	-	-	-	-	T2_INT_EN	LVI_INT_EN	WDT_INT_EN
-	-	-	-	-	RW	RW	RW

T2_INT_EN	Timer 2中断使能位	Initial value : 00H
0	Disable	
1	Enable	
LVI_INT_EN	LVI中断使能位	
0	Disable	
1	Enable	
WDT_INT_EN	WDT中断使能位	
0	Disable	
1	Enable	

IOFFSET (中断偏移寄存器) : 9AH

7	6	5	4	3	2	1	0
OFFSET7	OFFSET6	OFFSET5	OFFSET4	OFFSET3	OFFSET2	OFFSET1	OFFSET0
W	W	W	W	W	W	W	W

OFFSET[7:0]	中断偏移量, 偏移后的中断向量地址 = 256 * INT_OFFSET	Initial value : xxH
	只有当SYSCON_AR=5AH时, IOFFSET才能修改, 所以要先写5AH到SYSCON_AR,	
	再写IOFFSET, 写完IOFFSET后还要清零 SYSCON_AR	

IRQ0 (中断请求寄存器0) : A0H

7	6	5	4	3	2	1	0	
-	-	ADC_INT_F	T1_INT_F	T0_INT_F	PCI_F	EINT1_F	EINT0_F	
-	-	RW	RW	RW	RW	RW	RW	
		ADC_INT_F	ADC中断触发会置1, 标志在中断响应或者写入0后清零				Initial value : 00H	
		0	ADC interrupt doesn't occur					
		1	ADC interrupt occur					
		T1_INT_F	Timer 1中断触发会置1, 标志在中断响应或者写入0后清零					
		0	Timer 1 interrupt doesn't occur					
		1	Timer 1 interrupt occur					
		T0_INT_F	Timer 0中断触发会置1, 标志在中断响应或者写入0后清零					
		0	Timer 0 interrupt doesn't occur					
		1	Timer 0 interrupt occur					
		PCI_INT_F	PCI 中断触发会置1, 标志在中断响应或者写入0后清零					
		0	PCI interrupt doesn't occur					
		1	PCI interrupt occur					
		EINT1_F	外部中断1触发会置1, 标志在中断响应或者写入0后清零					
		0	External interrupt 1 interrupt doesn't occur					
		1	External interrupt 1 interrupt occur					
		EINT0_F	外部中断0触发会置1, 标志在中断响应或者写入0后清零					
		0	External interrupt 0 interrupt doesn't occur					
		1	External interrupt 0 interrupt occur					

IRQ1 (中断请求寄存器1) : A8H

7	6	5	4	3	2	1	0	
-	-	-	-	-	T2_INT_F	LVI_INT_F	WDT_INT_F	
-	-	-	-	-	RW	RW	RW	
		T2_INT_F	Timer 2中断触发会置1, 标志在中断响应或者写入0后清零				Initial value : 00H	
		0	T2 interrupt doesn't occur					
		1	T2 interrupt occur					
		LVI_INT_F	LVI中断触发会置1, 标志在中断响应或者写入0后清零					
		0	LVI interrupt doesn't occur					
		1	LVI interrupt occur					
		WDT_INT_F	WDT中断触发会置1, 标志在中断响应或者写入0后清零					
		0	WDT interrupt doesn't occur					
		1	WDT interrupt occur					

EINT_EDGE_P (外部中断正边沿选择寄存器) : 91H

7	6	5	4	3	2	1	0	
-	-	-	-	-	-	EINT1_EN_P	EINT0_EN_P	
-	-	-	-	-	-	RW	RW	
		EINT1_EN_P	外部中断1正边沿使能				Initial value : 00H	
		0	Disable					
		1	Enable					
		EINT0_EN_P	外部中断0正边沿使能					
		0	Disable					
		1	Enable					

EINT_EDGE_N (外部中断负边沿选择寄存器) : 99H

7	6	5	4	3	2	1	0
-	-	-	-	-	-	EINT1_EN_N	EINT0_EN_N
-	-	-	-	-	-	RW	RW

Initial value : 00H

EINT1_EN_N 外部中断1负边沿使能
 0 Disable
 1 Enable

EINT0_EN_N 外部中断0负边沿使能
 0 Disable
 1 Enable

IP (中断优先级寄存器) : A1H

7	6	5	4	3	2	1	0
-	-	IP05	IP04	IP03	IP02	IP01	IP00
-	-	RW	RW	RW	RW	RW	RW

Initial value : 00H

IP05 ADC 中断优先级
 0 Level 0
 1 Level 1

IP04 Timer 1 中断优先级
 0 Level 0
 1 Level 1

IP03 Timer 0 中断优先级
 0 Level 0
 1 Level 1

IP02 PCI 中断优先级
 0 Level 0
 1 Level 1

IP01 EINT0 中断优先级
 0 Level 0
 1 Level 1

IP00 EINT1 中断优先级
 0 Level 0
 1 Level 1

IP1 (中断优先级寄存器1) : A9H

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

Initial value : 00H

IP12 Timer 2 中断优先级
 0 Level 0
 1 Level 1

IP11 LVI 中断优先级
 0 Level 0
 1 Level 1

IP10 WDT 中断优先级
 0 Level 0
 1 Level 1

11 外围器件

11.1 时钟发生器

11.1.1 概述

时钟发生器为CPU和外围设备提供基本时钟脉冲。系统时钟为内部 RC-OSC时，默认的分频数是4。

-工厂校准的内部 RC 振荡器(32MHz)

- . 内部时钟 (16MHz)/2 (16MHz)
- . 内部时钟 (16MHz)/4 (8MHz, 默认的系统时钟)
- . 内部时钟 (16MHz)/8 (4MHz)
- . 内部时钟 (16MHz)/32 (1MHz)

11.1.2 方框图

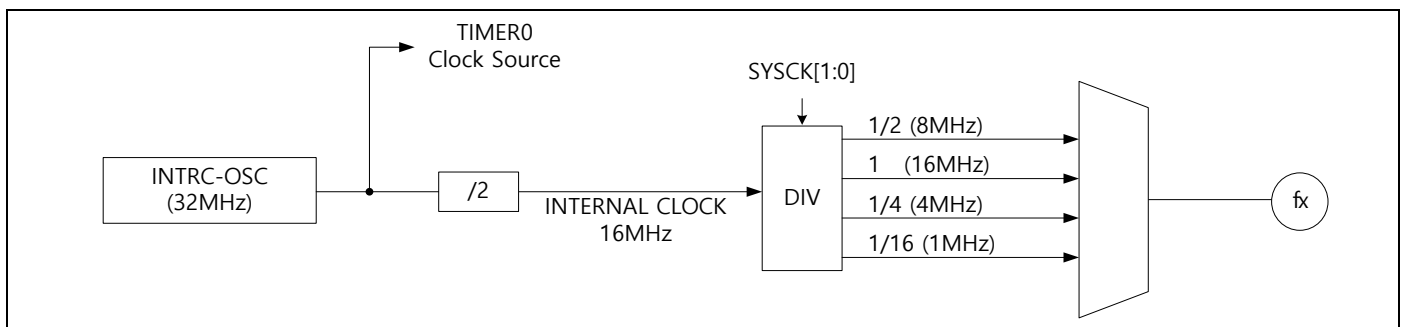


Figure 11.1 Clock Generator Block Diagram

SCCR (系统时钟控制寄存器) : 8AH

7	6	5	4	3	2	1	0
0	WDTRCON	FWKTIME[2]	FWKTIME[1]	FWKTIME[0]	-	SYSCK1	SYSCK0
-	RW	RW	RW	RW	-	RW	RW

WDTRCON	WDTRC振荡器使能/禁止	Initial value : 00H
0	WDTRC off (default)	
1	WDTRC on	
FWKTIME[2:0]	快速唤醒并退出Stop模式的时间 @ 8MHz系统时钟，这个时间由系统时钟决定	
0 0 0	16ms (default)	
0 0 1	16us	
0 1 0	32us	
0 1 1	64us	
1 0 0	128us	
1 0 1	256us	
1 1 0	512us	
1 1 1	8ms	
SYSCK[1:0]	分频率 (fINTRC=32MHz)	
0 0	fINTRC/4(default)	
0 1	fINTRC/2	
1 0	fINTRC/8	
1 1	fINTRC/32	

注意)只有SYSCON_AR=5AH时，SCCR才能修改，所以先写入5AH，再写入SCCR，最后要清零SYSCON_AR。例子：

```
Ex) SYSCON_AR = 0x5A;
    SCCR = 0x40;           // WDTRCON, SCCR[7]和 SCCR[2]必须为 0
    SYSCON_AR = 0x00;     // 内部 RC-OSC 的稳定时间最少要 50uS
```

11.2 WDT看门狗定时器

11.2.1 概述

WDT可以迅速地检测到CPU因干扰等导致进入死循环，并将CPU恢复到正常状态。WDT可以产生信号让CPU复位或者触发中断。WDT也可用作定时器产生中断。通过设置WDT_RESET_EN位，WDT可用作8位定时器或者看门狗。如果WDT_CLR置1，WDT计数器会清零并开始计数，一个机器周期后，该位自动清零。WDT由8位二进制计数器和WDT数据寄存器组成。当8位二进制计数器的值等于8位WDTR的值时，产生WDT中断，此时会根据WDT_RESET_EN位的设置来选择让CPU触发中断或者复位。注意：WDT_RESET_EN位默认值为1，它不会由于WDT复位而清零。

WDT时钟源可选择系统时钟或WDTRC（8KHz），WDT的中断间隔由WDT_CLK_DIV和WDTR的设置值决定。计算公式如下：

$$WDT_TIME[s] = \frac{WDT_CLK_DIV \times WDTR}{fx_or_fWDT}$$

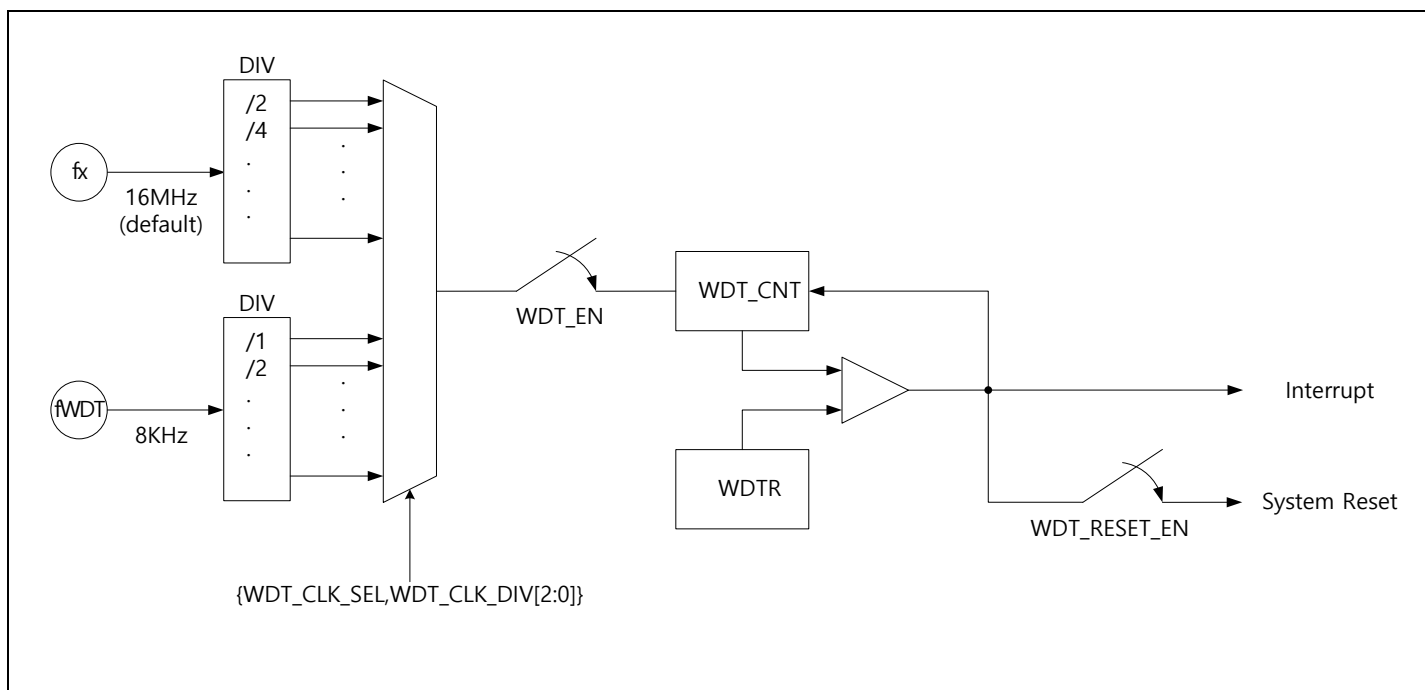


Figure 11.2 WDT Block Diagram

11.2.2 寄存器图

Name	Address	Dir	Default	Description
WDTMR	9DH	R/W	73H	WDT Mode Register
WDTR	9EH	W	FFH	WDT Data Register
WDTCR	9EH	R	00H	WDT Counter Register

Table 11.1 Register Map

11.2.3 WDT寄存器说明

WDTMR (WDT模式寄存器) : 9DH

7	6	5	4	3	2	1	0
WDT_CLK_SEL	WDT_CLK_DIV2	WDT_CLK_DIV1	WDT_CLK_DIV0	-	WDT_CLR	WDT_RESET_EN	WDT_EN
RW	RW	RW	RW	-	W	RW	RW

WDT_CLK_SEL WDT时钟源选择 Initial value : 73H

0 System clock (8Mhz, default)

1 WDTRC OSC (8KHz)

{ WDT_CLK_SEL, WDT_CLK_DIV[2:0] } WDT时钟分频率

SEL	DIV2	DIV1	DIV0	Description
0	0	0	0	fx / 2 ¹
0	0	0	1	fx / 2 ²
0	0	1	0	fx / 2 ⁴
0	0	1	1	fx / 2 ⁶
0	1	0	0	fx / 2 ⁸
0	1	0	1	fx / 2 ¹⁰
0	1	1	0	fx / 2 ¹²
0	1	1	1	fx / 2 ¹⁴
1	0	0	0	fWDT / 2 ⁰
1	0	0	1	fWDT / 2 ¹
1	0	1	0	fWDT / 2 ²
1	0	1	1	fWDT / 2 ⁴
1	1	0	0	fWDT / 2 ⁶
1	1	0	1	fWDT / 2 ⁷
1	1	1	0	fWDT / 2 ⁸
1	1	1	1	fWDT / 2 ⁹

WDT_CLR 清零 WDT计数器

0 运行

1 清零WDT计数器 (1个周期后自动清零)

WDT_RESET_EN WDT复位控制位

0 Reset Disable

1 Reset Enable

WDT_EN WDT使能位

0 WDT Disable

1 WDT Enable

WDTR (WDT寄存器) : 9EH

7	6	5	4	3	2	1	0
WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0
W	W	W	W	W	W	W	W

WDTR[7:0] 设置周期 Initial value : FFH

WDT中断间隔 = (WDT时钟源周期) / WDT_CLK_DIV x (WDTR数值+1)

注意)写入WDTR寄存器的数值必须大于 01H

WDTCR (WDT计数器寄存器) : 9EH

7	6	5	4	3	2	1	0
WDTCR7	WDTCR6	WDTCR5	WDTCR4	WDTCR3	WDTCR2	WDTCR1	WDTCR0
R	R	R	R	R	R	R	R

WDTCR[7:0] WDT计数器寄存器 Initial value : 00H

11.2.4 WDT中断时序波形

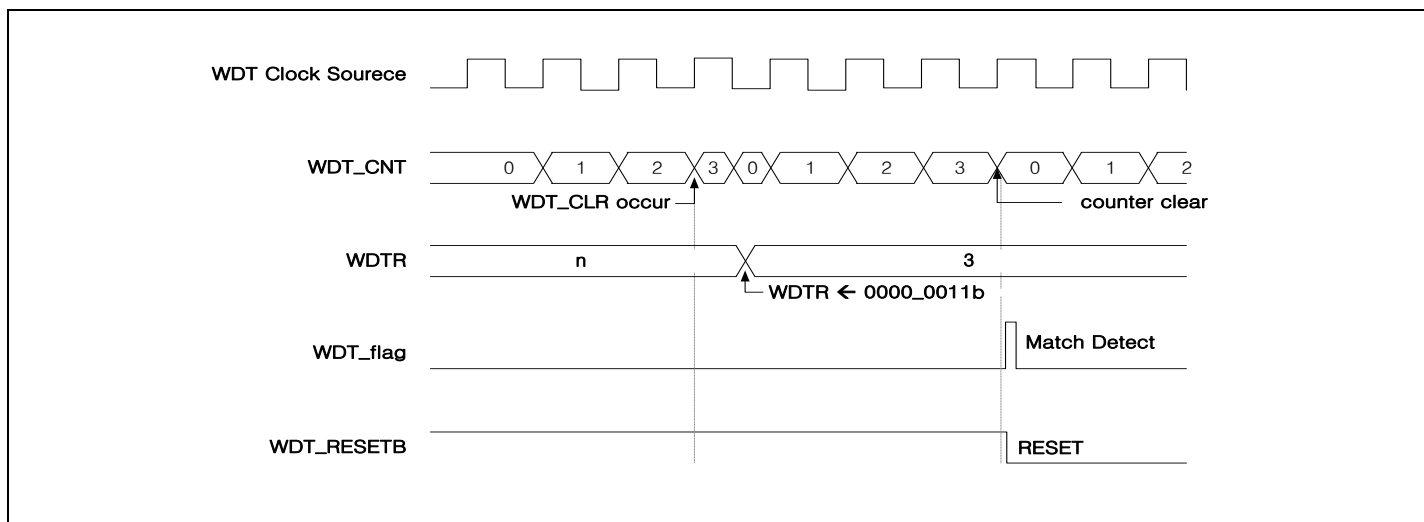


Figure 11.3 WDT Interrupt Timing Waveform

11.2.5 WDT复位时间表

WDT_CLK_SEL	WDT_CLK_DIV	WDTR	Reset Time	WDT_CLK_SEL	WDT_CLK_DIV	WDTR	Reset Time
0	0	255	0.064ms	1	0	255	0.032s
0	1	255	0.128ms	1	1	255	0.064s
0	2	255	0.512ms	1	2	255	0.128s
0	3	255	2.048ms	1	3	255	0.512s
0	4	255	8.192ms	1	4	255	2.048s
0	5	255	32.768ms	1	5	255	4.096s
0	6	255	131.072ms	1	6	255	8.192s
0	7	255	524.288ms	1	7	255	16.384s

Table 11.2 Register Map

11.3 16位定时器 / PWM (Timer0, Timer1)

11.3.1 概述

16位定时器x(0~1) 由多路转换器、定时器数据高/低寄存器、定时器高/低寄存器、定时器模式控制寄存器、PWM占空比高/低寄存器、PWM周期高/低寄存器组成。如果使用内部16位定时器/计数器，将没有端口输出功能。

16位定时器x的时钟可以选择内部或外部时钟源 (EC0, EC1)。内部时钟为主时钟的分频输出。

T0CR1寄存器的T32M位可设置内部RCOSC(32MHz)为Timer0的时钟源。

11.3.2 16位定时器/计数器模式

在16位定时器/计数器模式中，如果 TxH + TxL 的值和 TxDRH + TxDRL 的值相匹配，Tx/PWMx端口会输出占空比为50:50的方波，频率计算公式如下：

$$f_{COMP} = \frac{\text{Timer Clock Frequency}}{2 \times \text{Prescaler Value} \times (\text{TxDR} + 1)}$$

f_{COMP} 是定时器输出频率，TxDR是 TxDRH和TxDRL合起来的16位数值

Tx/PWMx作为比较输出时，P1FSR_L[7:6]或P1FSR_H[1:0]必须置为10B

如Figure 11.4所示，16位定时器/计数器模式由控制寄存器选择

当读取 TxH, TxL时，必须先读取 TxL，因为读取 TxL时，TxH的值会锁存到缓存里，当读取TxH时，其实是读取TxH缓存器里的值。

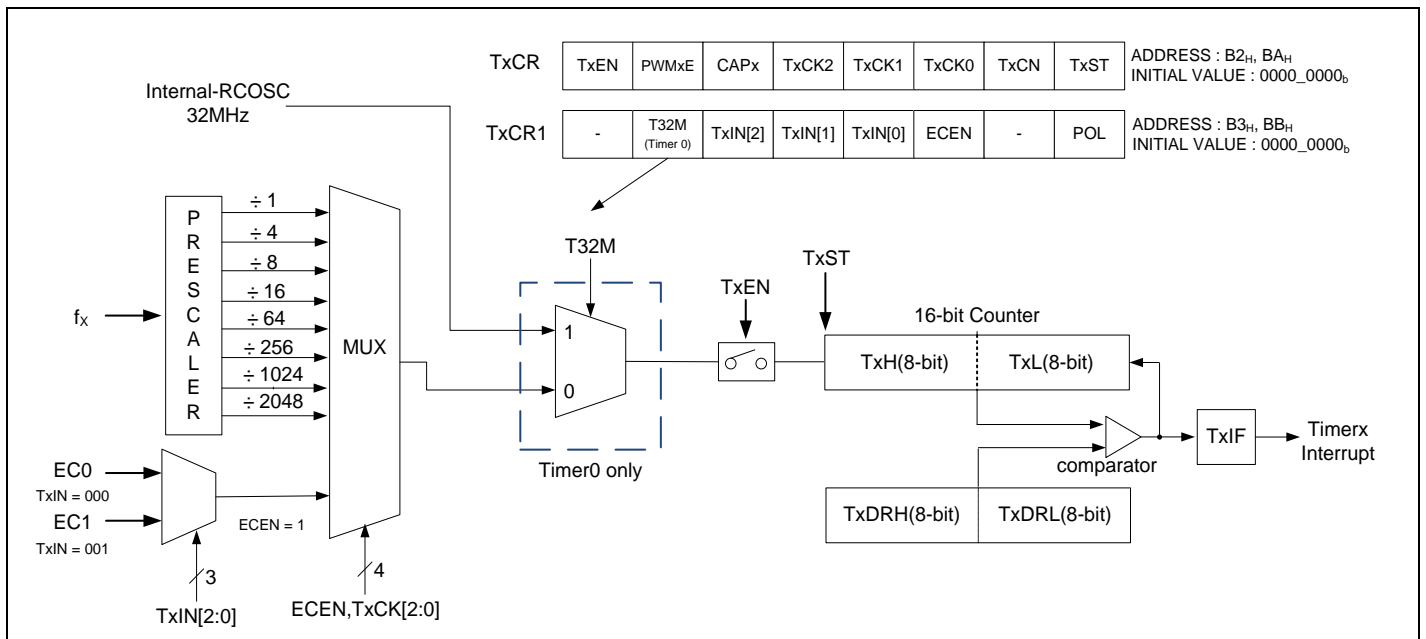


Figure 11.4 Timerx 16-bit Mode Block Diagram

11.3.3 16位捕捉模式

定时器x(0~1)由TxCR寄存器的CAPx位置位进入捕捉模式，它使用和输出比较模式相同的时钟源。当TxH、TxL和TxDRH、TxDRL的值分别相匹配时会产生中断，捕捉结果放在CDRxH、CDRxL。TxH、TxL的值由硬件自动清零或者重新开始计数时自动清零。

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。通过EINT_EDGE_P、EINT_EDGE_N寄存器来设置选择外部中断EINTx功能。

CDRxH、PWMxDRH和TxH的地址相同。在捕捉模式下，读取是读CDRxH而不是TxH，因为路径是指向CDRxH的，在写入时会改变PWMxDRH的操作。PWMxDRL、TxL、CDRxL的操作相同。

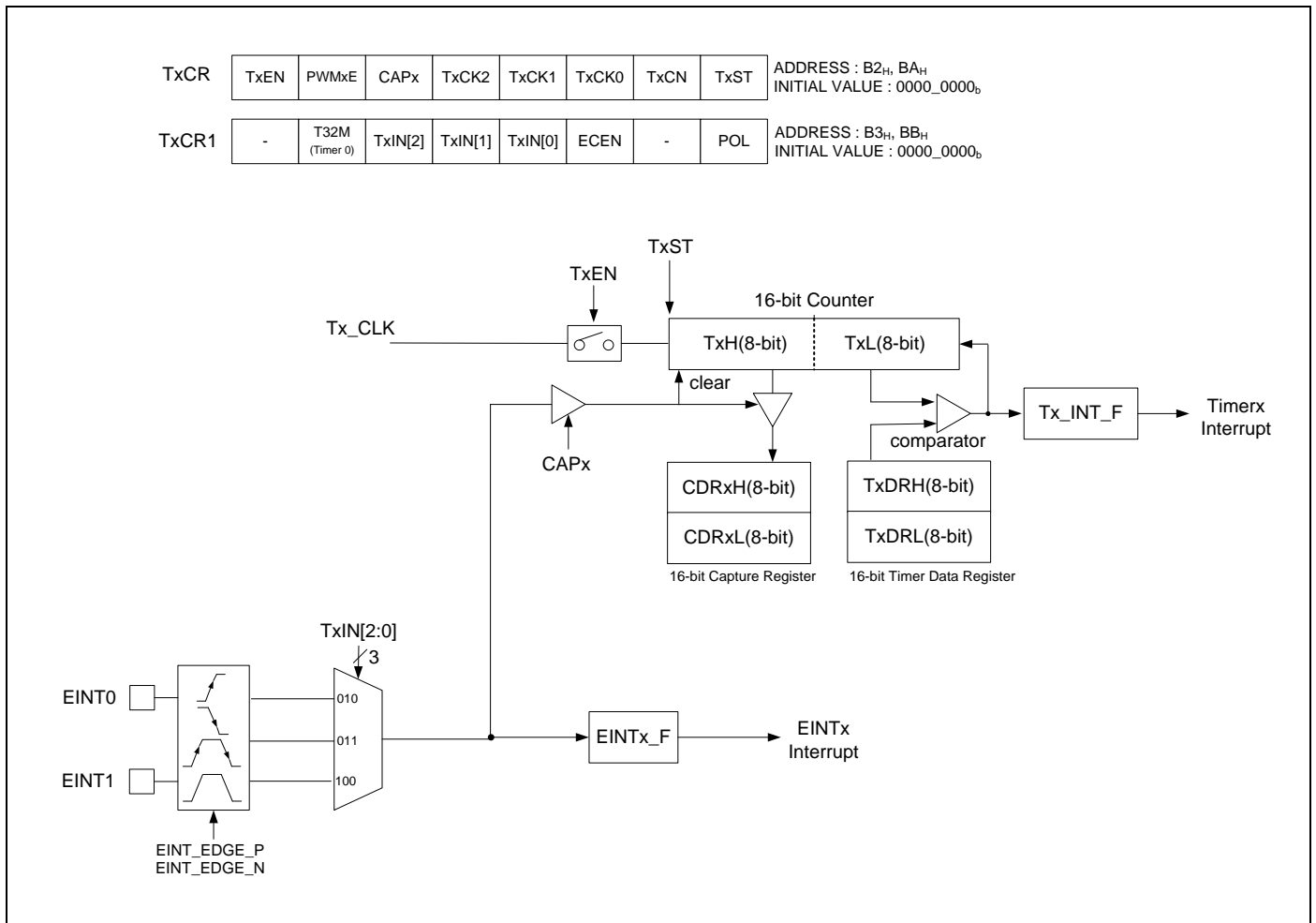


Figure 11.5 Timerx 16-bit Capture Mode

11.3.4 PWM模式

定时器x(0~1) 具有PWM (pulse Width Modulation) 功能, 在PWM模式, Tx/PWMx输出引脚可以输出精度达到16位的PWM方波。写入10B到P1FSR_L[7:6]或P1FSR_H[1:0]可以配置引脚为PWM输出。通过PWMxPRH、PWMxPRL、PWMxDRH和PWMxDRL设置PWM输出模式。在写PWM寄存器之前, 必须先把TxCR寄存器的PWMxE位置“1”。

$$\text{PWM Period} = (\text{PWMxPRH, PWMxPRL} + 1) \times \text{Timerx Clock Period}$$

$$\text{PWM Duty} = (\text{PWMxDRH, PWMxDRL} + 1) \times \text{Timerx Clock Period}$$

Resolution	Frequency			
	T32M = 1 (Timer 0)	T32M = 0 TxCK[2:0]=000 (125ns)	T32M = 0 TxCK[2:0]=001 (500ns)	T32M = 0 TxCK[2:0]=010 (1000ns)
16-bit	488.281 Hz	122.070 Hz	30.518Hz	15.259Hz
15-bit	976.563 Hz	244.141 Hz	61.035 Hz	30.518Hz
10-bit	31.250 kHz	7.813 kHz	1.953 kHz	0.977 kHz
9-bit	62.500 kHz	15.625 kHz	3.906 kHz	1.953 kHz
8-bit	125.00 kHz	31.250 kHz	7.812 kHz	3.906 kHz

Table 11.3 PWM Frequency vs. Resolution at 8MHz and 32MHz

在PWM模式, 占空比的值和计数器匹配后, 会将周期值和计数器比较, 当计数值和周期值匹配后, 计数器重新计数。如果占空比的值和周期值相同, 则占空比的值和计数器匹配时, 计数器不能重新开始计数。强烈建议不要把PWM的周期和占空比设为相同值, 见Figure 11.8。TxCR寄存器的POL位可以设置占空比周期的极性。

注意)周期值必须大于占空比的值 {PWMxPRH, PWMxPRL} > {PWMxDRH, PWMxDRL}

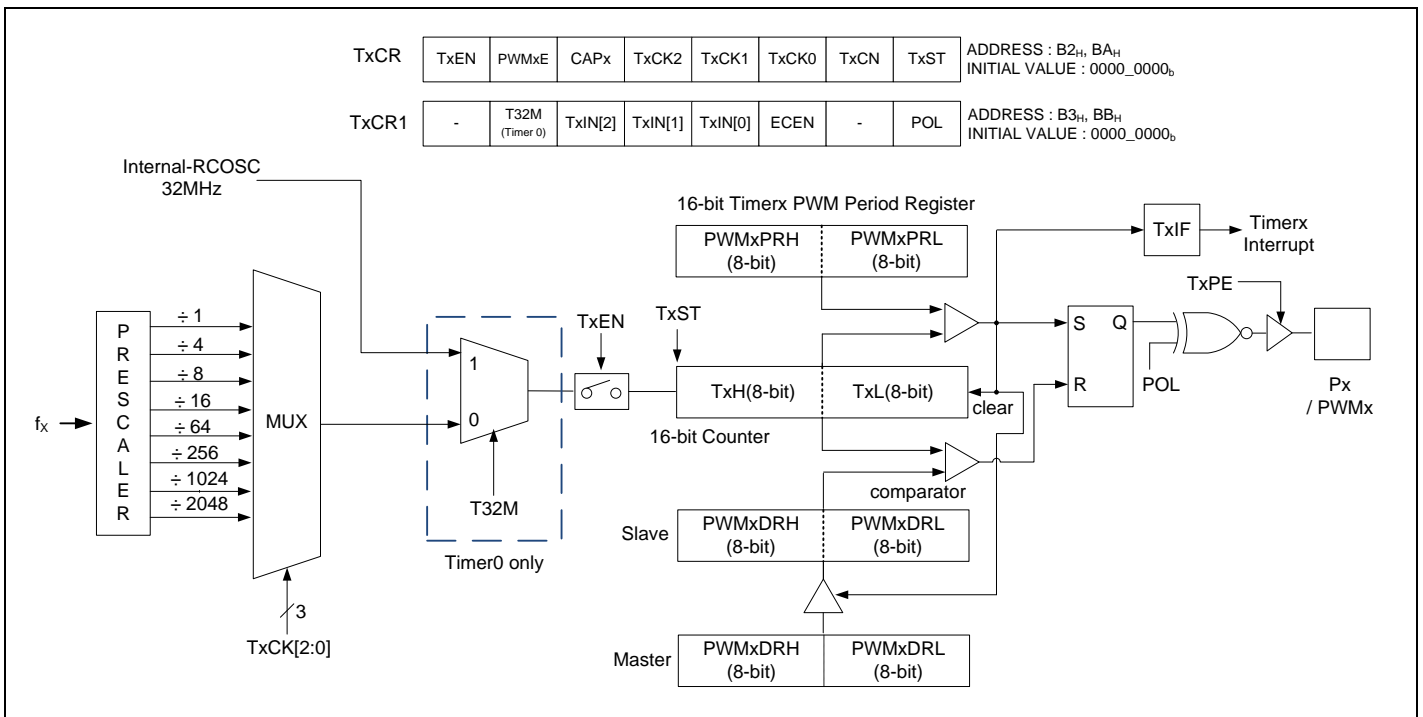


Figure 11.6 PWM Mode

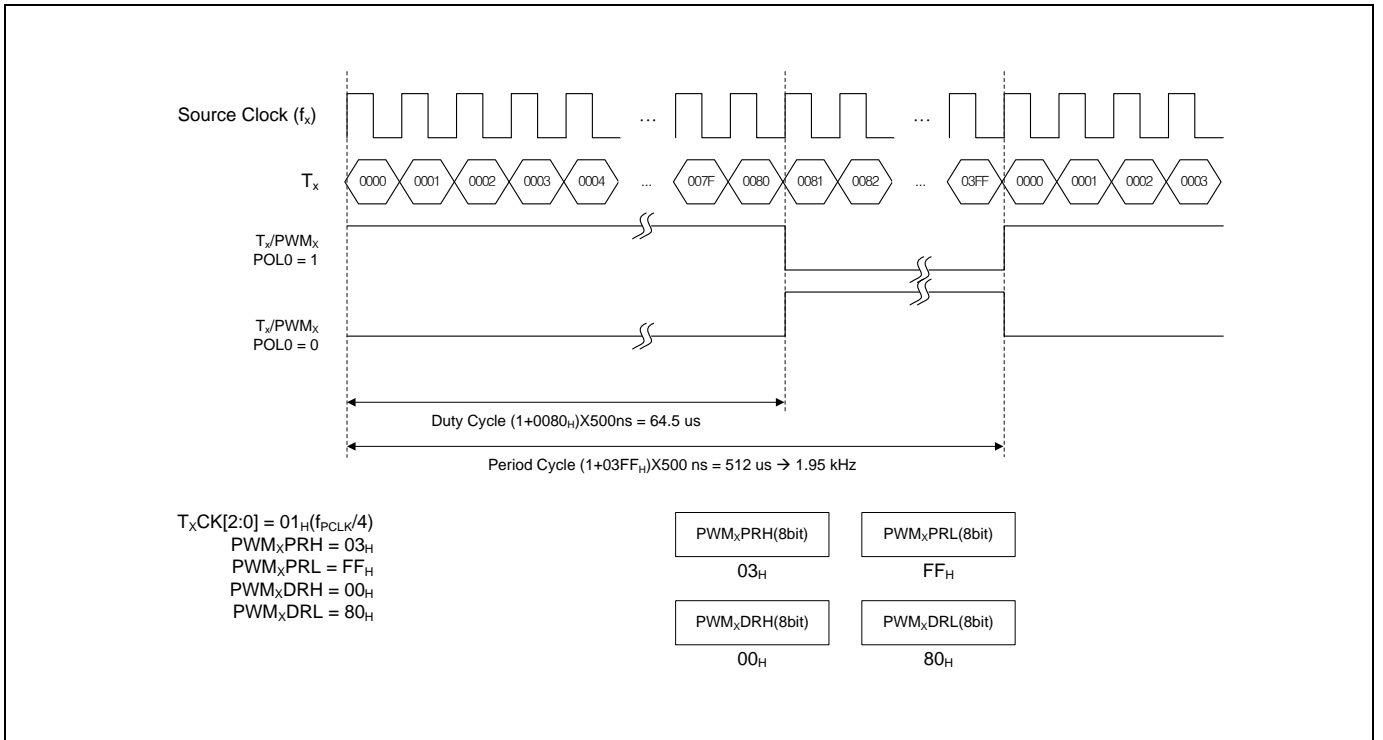


Figure 11.7 Example of PWM at 8MHz

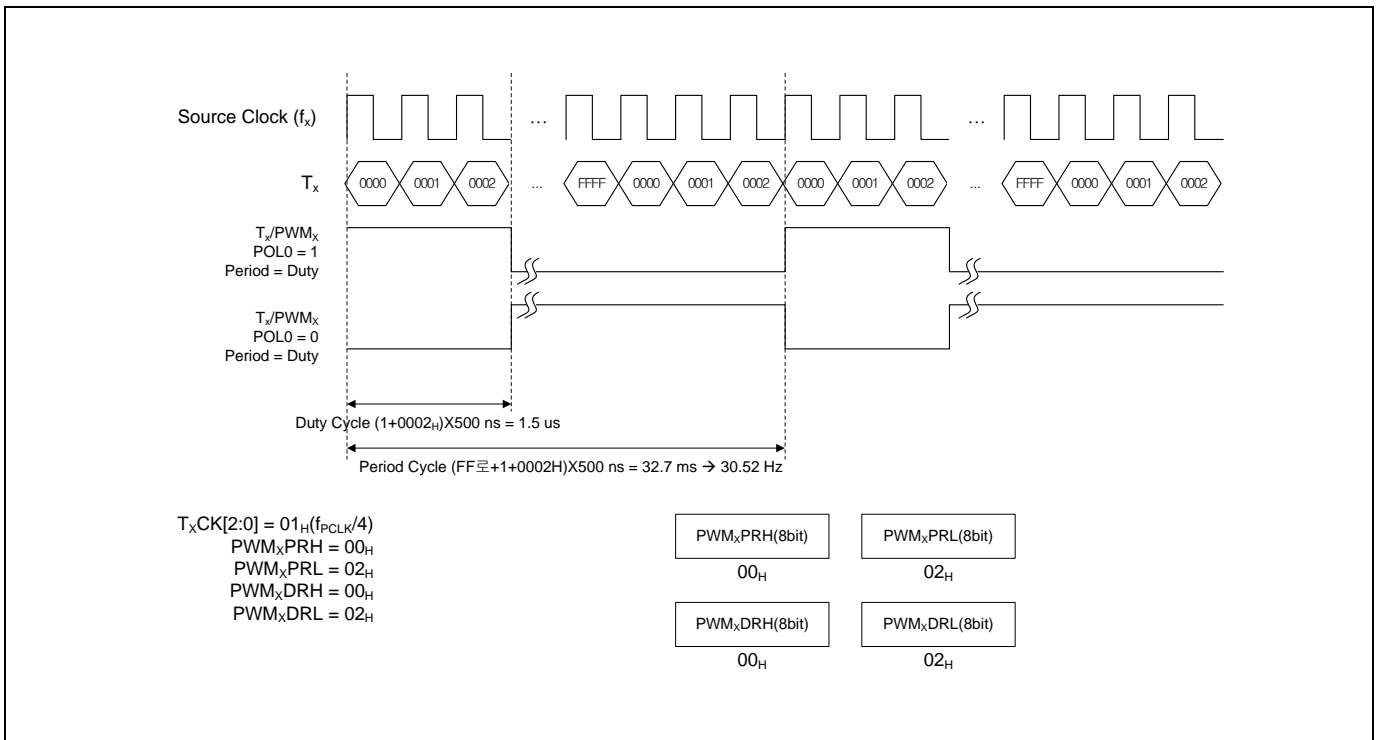


Figure 11.8 Example of PWM at 8MHz (Period = Duty)

11.3.5 定时器数据和周期/占空比写入

当写入一个数值到定时器x数据寄存器时，要先写TxDRH，之后再写TxDRL。因为写入高寄存器时，数据会先保存在缓存里，当写入低寄存器时，会把缓存里的数据一起写入到高寄存器。定时器周期/占空比寄存器(PWMxDRH、PWMxDRL、PWMxPRH、PWMxPRL)的操作方法都与TxDRH/L相同。

每当计数器开始新的计数时，数据高/低寄存器会自动加载到比较寄存器。

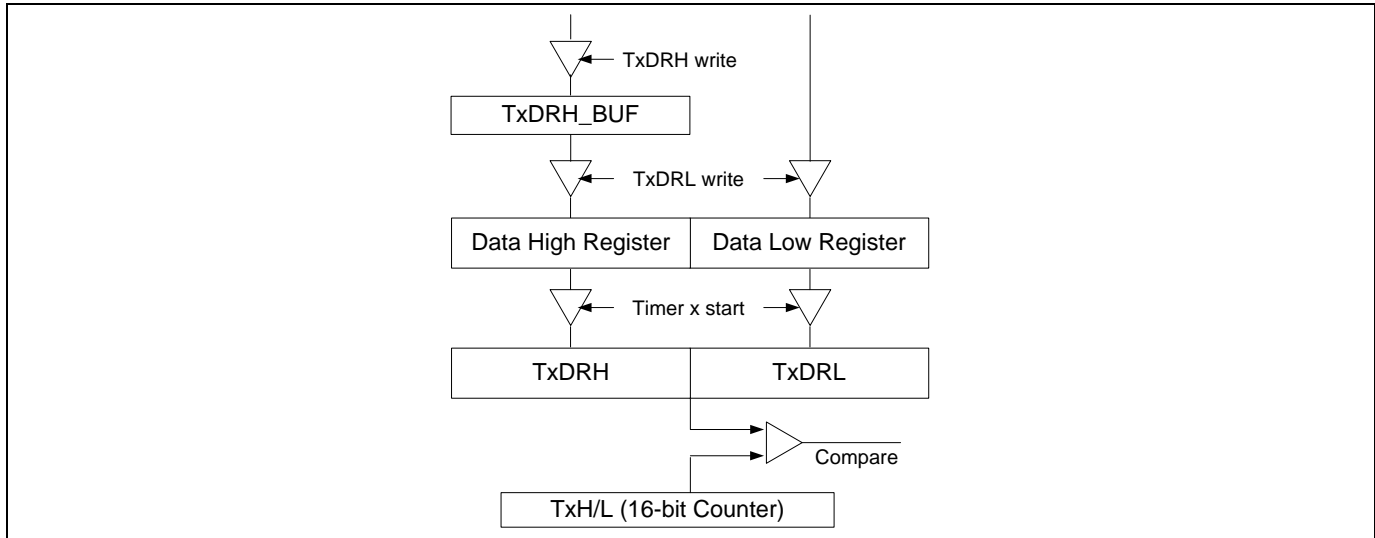


Figure 11.9 Timer x Compare Data Write

11.3.6 寄存器图

Name	Address	Dir	Default	Description
T0CR	B2H	R/W	00H	Timer 0 Mode Control Register
T0CR1	B3H	R/W	00H	Timer 0 Mode Control Register 1
T0L	B4H	R	00H	Timer 0 Low Register
PWM0DRL	B4H	R/W	00H	PWM 0 DutyRegister Low
CDR0L	B4H	R	00H	Timer 0 Capture Data Register Low
T0H	B5H	R	00H	Timer 0 Register High
PWM0DRH	B5H	R/W	00H	PWM 0 Duty Register High
CDR0H	B5H	R	00H	Timer 0 Capture Data Register High
T0DRL	B6H	R/W	FFH	Timer 0 Compare Data Register Low
PWM0PRL	B6H	R/W	FFH	PWM 0 Period Register Low
T0DRH	B7H	R/W	FFH	Timer 0 Compare Data Register High
PWM0PRH	B7H	R/W	FFH	PWM 0 Period Register High
T1CR	BAH	R/W	00H	Timer 1 Mode Control Register
T1CR1	BBH	R/W	00H	Timer 1 Mode Control Register 1
T1L	BCH	R	00H	Timer 1 Register Low
PWM1DRL	BCH	R/W	00H	PWM 1 Duty Register Low
CDR1L	BCH	R	00H	Timer 1 Capture Data Register Low
T1H	BDH	R	00H	Timer 1 Register High
PWM1DRH	BDH	R/W	00H	PWM 1 Duty Register High
CDR1H	BDH	R	00H	Timer 1 Capture Data Register High
T1DRL	BEH	R/W	FFH	Timer 1 Compare Data Register Low
PWM1PRL	BEH	R/W	FFH	PWM 1 Period Register Low
T1DRH	BFH	R/W	FFH	Timer 1 Compare Data Register High
PWM1PRH	BFH	R/W	FFH	PWM 1 Period Register High

Table 11.4 Register Map

11.3.7 定时器/计数器x寄存器说明

注意) 在写入Tx, PWM, CDRx寄存器之后, 需要置“1” TxST位。

TxCR (定时器0~1模式控制寄存器) : B2H, BAH

7	6	5	4	3	2	1	0
TxEN	PWMxE	CAPx	TxCK2	TxCK1	TxCK0	TxCN	TxST
RW	RW	RW	RW	RW	RW	RW	RW
	TxEN	定时器x使能位				Initial value : 00H	
		0	0				
		1	1				
	PWMxE	PWM模式使能					
		0	0				
		1	1				
	CAPx	定时器x捕捉模式使能					
		0	0				
		1	1				
	TxCK[2:0]	定时器x时钟源选择。Fx是系统主时钟频率					
		TxCK2	TxCK1	TxCK0	description		
		0	0	0	fx		
		0	0	1	fx/4		
		0	1	0	fx/8		
		0	1	1	fx/16		
		1	0	0	fx/64		
		1	0	1	fx/256		
		1	1	0	fx/1024		
		1	1	1	fx/2048		
	TxCN	定时器x计数器暂停/继续					
		0	0				
		1	1				
	TxST	定时器x开始/停止					
		0	0				
		1	1				

TxL (定时器0~1低寄存器, 只读) : B4H, BCH

7	6	5	4	3	2	1	0
TxL7	TxL6	TxL5	TxL4	TxL3	TxL2	TxL1	TxL0
R	R	R	R	R	R	R	R
TxL[7:0]		TxL计数器周期低字节				Initial value : 00H	

TxH (定时器0~1高寄存器, 只读) : B5H, BDH

7	6	5	4	3	2	1	0
TxH7	TxH6	TxH5	TxH4	TxH3	TxH2	TxH1	TxH0
R	R	R	R	R	R	R	R
TxH[7:0]		TxH计数器周期高字节				Initial value : 00H	

TxCR1 (定时器0~1模式控制寄存器1) : B3H, BBH

7	6	5	4	3	2	1	0
-	T32M	TxIN[2]	TxIN[0w1]	TxIN[0]	ECEN	-	POL
-	RW	RW	RW	RW	RW	-	RW
T32M		定时器时钟源设为 32MHz IRC. (只有定时器0有效)				Initial value : 00H	
		0 -					
		1 时钟源为 32MHz IRC					
TxIN[2:0]		设置捕捉模式下的每个计数器和外部中断					
		TxIN2	TxIN1	TxIN0	Description		
		0	0	0	EC0		
		0	0	1	EC1		
		0	1	0	XINT0		
		0	1	1	XINT1		
		1	x	x	-		
ECEN		控制所有计数器					
		0 0					
		1 1					
POL		设置 PWM的极性					
		0 0					
		1 1					
		注意) EC0不能在ocd模式下使用					

CDRxL (捕捉器0~1数据低寄存器, 只读) : B4H, BCH

7	6	5	4	3	2	1	0
CDRxL07	CDRxL06	CDRxL05	CDRxL04	CDRxL03	CDRxL02	CDRxL01	CDRxL00
R	R	R	R	R	R	R	R
CDRxL[7:0]		Tx Capture Low data.				Initial value : 00H	

PWMxDRL (PWM 0~1占空比低寄存器, 只写) : B4H, BCH

7	6	5	4	3	2	1	0
T17	T16	T15	T14	T13	T12	T11	T10
RW	RW	RW	RW	RW	RW	RW	RW
PWMxDRL[7:0]		Tx PWM占空比低字节				Initial value : 00H	
		注意) 当 PWMxE = 1和 TxST = 0时, 此寄存器才有效					

CDRxH (捕捉器0~1数据高寄存器, 只读) : B5H, BDH

7	6	5	4	3	2	1	0
CDRxH07	CDRxH06	CDRxH05	CDRxH04	CDRxH03	CDRxH02	CDRxH01	CDRxH00
R	R	R	R	R	R	R	R
CDRxH[7:0]		Tx Capture High data				Initial value : 00H	

PWMxDRH (PWM 0~1占空比高寄存器, 只写) : B5H, BDH

7	6	5	4	3	2	1	0
PWMxDRH7	PWMxDRH6	PWMxDRH5	PWMxDRH4	PWMxDRH3	PWMxDRH2	PWMxDRH1	PWMxDRH0
RW	RW	RW	RW	RW	RW	RW	RW
PWMxDRH[7:0]		Tx PWM 占空比高字节				Initial value : 00H	
		注意) 当 PWMxE = 1和 TxST = 0时, 此寄存器才有效					

TxDRL (定时器 0~1数据低寄存器, 只写) : B6H, BEH

7	6	5	4	3	2	1	0
TxDRL7	TxDRL6	TxDRL5	TxDRL4	TxDRL3	TxDRL2	TxDRL1	TxDRL0
RW	RW	RW	RW	RW	RW	RW	RW

TxDRL[7:0] TxL比较器数据低字节

Initial value : FFH

注意) 必须先清零 PWMxE才能写此寄存器

PWMxPRL (PWM 0~1周期低寄存器, 只写) : B6H, BEH

7	6	5	4	3	2	1	0
PWMxPRL7	PWMxPRL6	PWMxPRL5	PWMxPRL4	PWMxPRL3	PWMxPRL2	PWMxPRL1	PWMxPRL0
RW	RW	RW	RW	RW	RW	RW	RW

PWMxPRL[7:0] TxPWM周期数据低字节

Initial value : FFH

注意) 当 PWMxE = 1和 TxST = 0时, 此寄存器才有效

TxDRH (定时器 0~1数据高寄存器, 只写) : B7H, BFH

7	6	5	4	3	2	1	0
TxDRH7	TxDRH6	TxDRH5	TxDRH4	TxDRH3	TxDRH2	TxDRH1	TxDRH0
RW	RW	RW	RW	RW	RW	RW	RW

TxDRH[7:0] TxH Compare High data

Initial value : FFH

注意) 必须先清零 PWMxE才能写此寄存器

PWMxPRH (PWM 0~1周期高寄存器, 只写) : B7H, BFH

7	6	5	4	3	2	1	0
PWMxPRH7	PWMxPRH6	PWMxPRH5	PWMxPRH4	PWMxPRH3	PWMxPRH2	PWMxPRH1	PWMxPRH0
RW	RW	RW	RW	RW	RW	RW	RW

PWMxPRH[7:0] TxPWM 周期数据高字节

Initial value : FFH

注意) 当 PWMxE = 1和 TxST = 0时, 此寄存器才有效

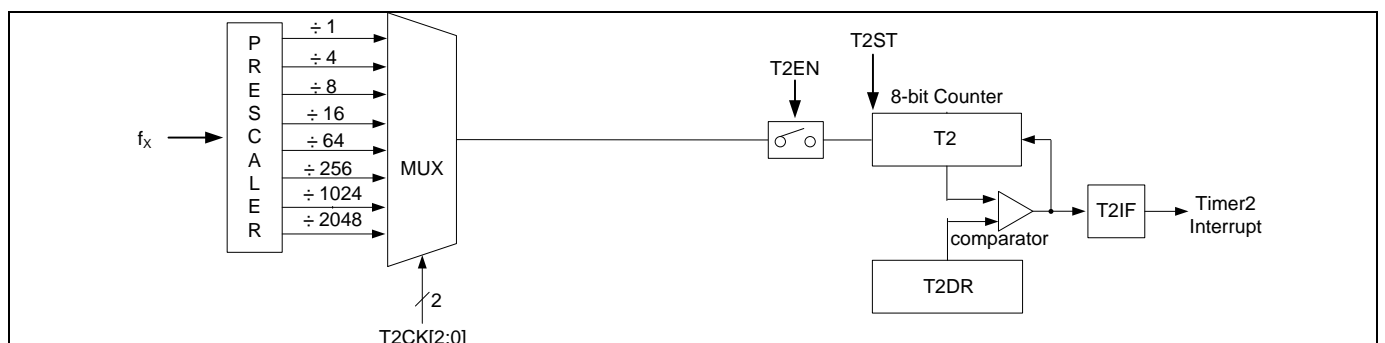
11.4 8位定时器(Timer2)

8位定时器2由多路转换器、定时器数据寄存器、定时器低寄存器、定时器模式控制寄存器组成。定时器2使用系统时钟的分频输出作为时钟源。定时器2不支持PWM、捕捉、外部计数器等模式, 也不支持外部引脚输出。

11.4.1 8位定时器/计数器模式

在8位定时器/计数器模式, 如果T2的值和T2DR的值相匹配, T2标志会置1, 如果定时器2中断使能将会触发中断请求。并且输出占空比为 50:50的方波, 频率计算公式如下: f_{COMP} 是定时器输出频率

$$f_{COMP} = \frac{\text{Timer Clock Frequency}}{2 \times \text{Prescaler Value} \times (T2DR + 1)}$$



11.4.2 寄存器图

Name	Address	Dir	Default	Description
T2CR	A4H	R/W	00H	Timer 2 Mode Control Register
T2	A6H	R	00H	Timer 2 Counter Register
T2DR	A7H	R/W	FFH	Timer 2 Compare Data Register

Table 11.5 Register Map

11.4.3 定时器2寄存器说明

T2CR (定时器2模式控制寄存器) : A4H

7	6	5	4	3	2	1	0
T2EN	-	-	T2CK2	T2CK1	T2CK0	T2CN	T2ST
RW	-	-	RW	RW	RW	RW	RW
T2EN		Control Timer 2				Initial value : 00H	
		0 Timer 2 disable					
		1 Timer 2 enable					
T2CK[2:0]		定时器2时钟源选择。Fx是系统主时钟频率					
		T2CK2	T2CK1	T2CK0	description		
		0	0	0	fx/4		
		0	0	1	fx/8		
		0	1	0	fx/16		
		0	1	1	fx/64		
		1	0	0	fx/256		
		1	0	1	fx/1024		
		1	1	0	fx/2048		
		1	1	1	fx/4096		
T2CN		定时器2计数器暂停/继续					
		0 暂停计数					
		1 继续计数					
T2ST		定时器2开始/停止					
		0 计数器停止					
		1 清零计数器并开始计数					

T2 (定时器2寄存器, 只读) : A6H

7	6	5	4	3	2	1	0
T27	T26	T25	T24	T23	T22	T21	T20
R	R	R	R	R	R	R	R
T2[7:0]		T2 Counter.				Initial value : 00H	

T2DR (定时器2数据寄存器, 只写) : A7H

7	6	5	4	3	2	1	0
T2DR7	T2DR6	T2DR5	T2DR4	T2DR3	T2DR2	T2DR1	T2DR0
RW	RW	RW	RW	RW	RW	RW	RW
T2DR[7:0]		T2 Compare data				Initial value : FFH	

11.5 12位A/D转换器

11.5.1 概述

模拟数字转换器可以将输入的模拟信号转换成相应的12位数字信号。ADC模块有10个模拟输入通道，多路复用器的输出作为ADC的输入，通过逐次逼近法得到近似值。ADC模块由控制寄存器ADCM、ADCM1 (ADC Mode Register)和AD结果寄存器ADCHR、ADCLR(ADC Result High/Low Register)组成。通过ADSEL[3:0]来选择AD输入通道，ADST位置1开始进行AD转换，转换完成，AD转换状态位AFLAG会置1，同时触发AD中断，转换结果保存在ADCHR和ADCLR寄存器。在AD转换期间读取AFLAG位的值为0。如果STBY位置1（节电模式），ADC会禁用，即使内部定时器、外部中断、比较器、定时器1PWM触发等外设都无法启动ADC。在STBY清零并重启ADC后（ADC电源使能），在一定周期内，ADC的转换结果可能不正确。当使用ADC输入端口时，需要设置PSR2、PSR3寄存器的相应位，以防止电流损耗或误动作。

$$\text{ADC转换时间} = \text{ADCLK} * 60 \text{ cycles}$$

11.5.2 方框图

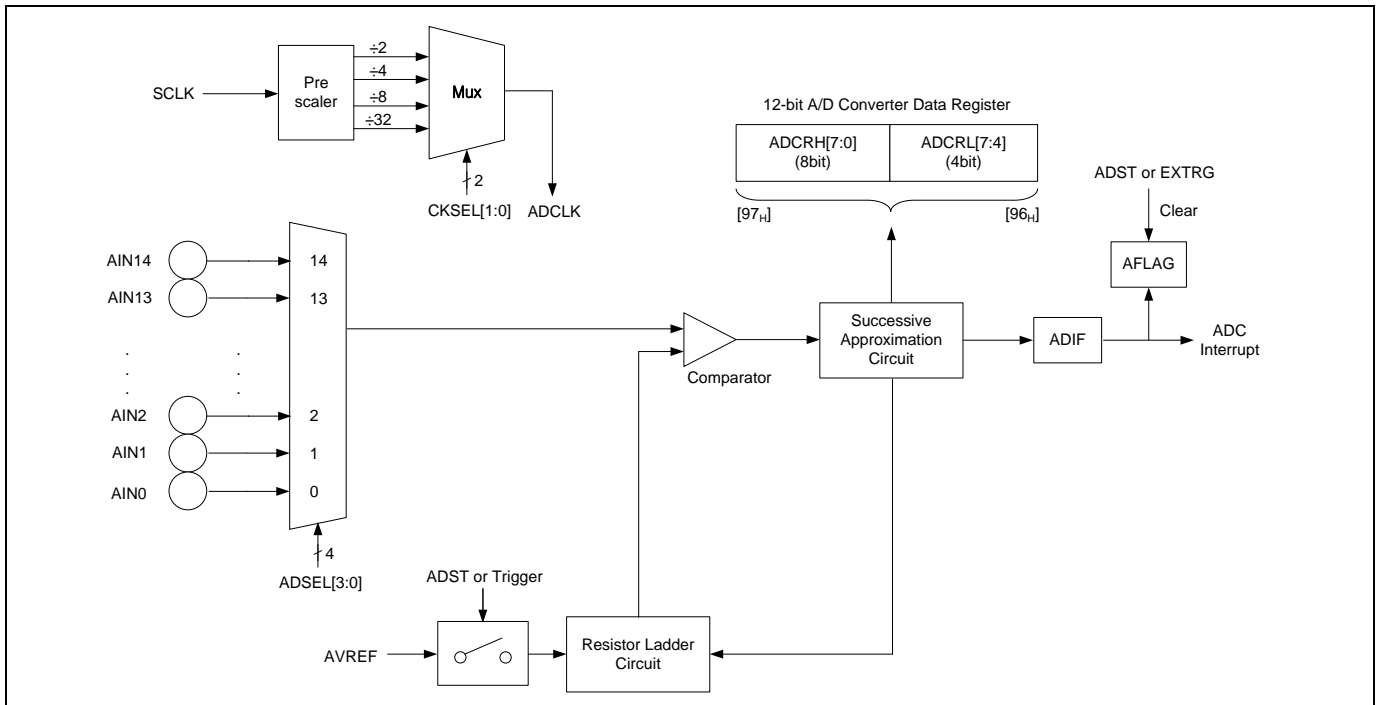


Figure 11.10 ADC Block Diagram

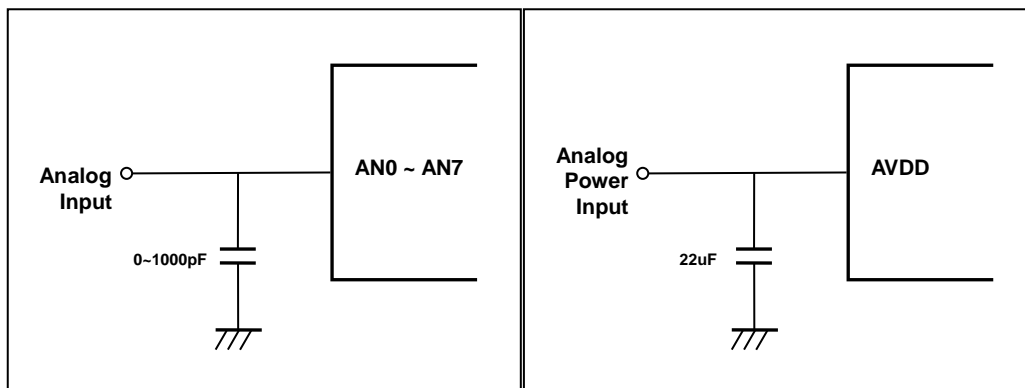


Figure 11.11 A/D 模拟输入脚和地线之间要滤波连接电容

11.5.3 ADC操作

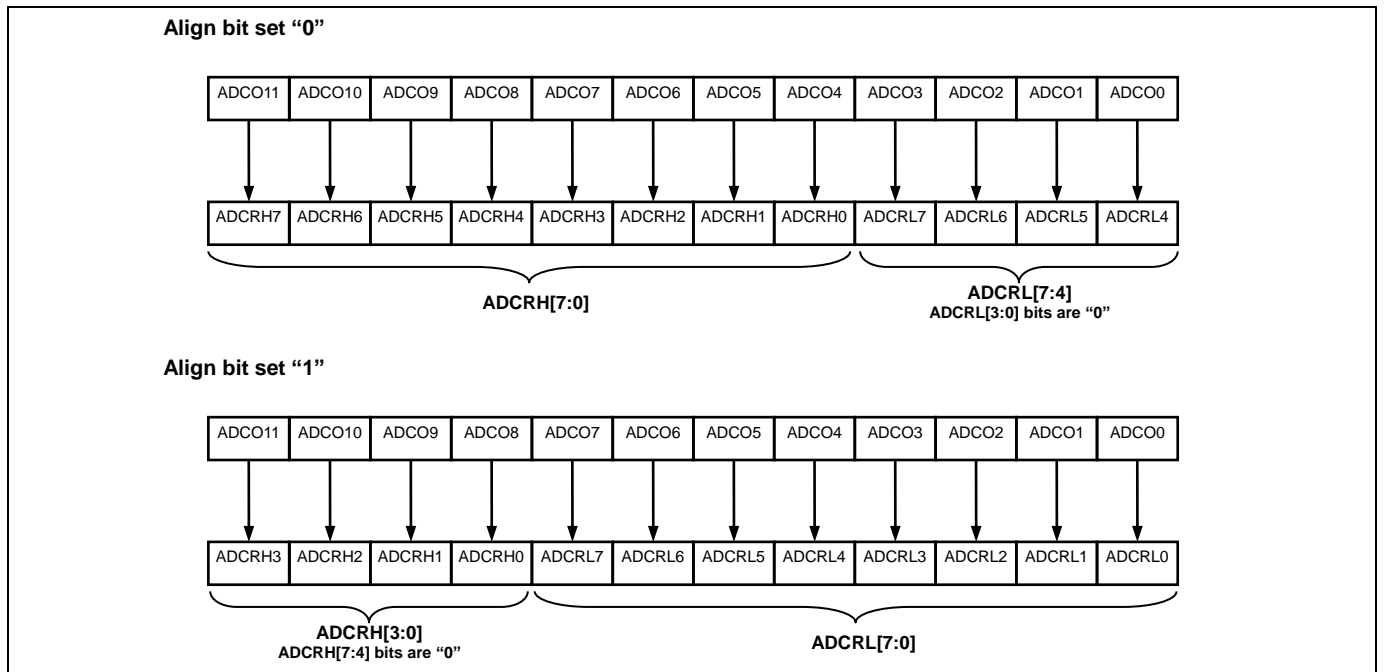


Figure 11.12 ADC Operation for Align bit

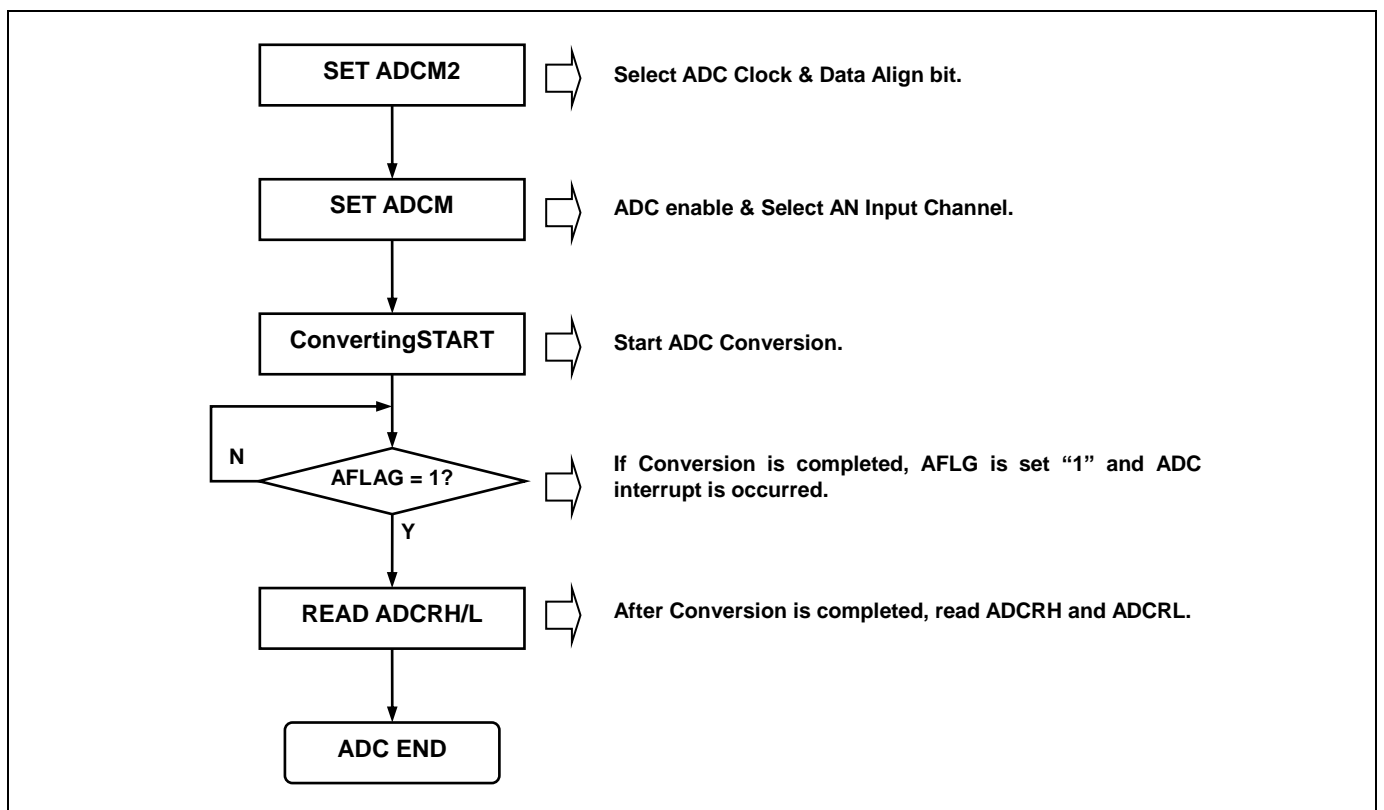


Figure 11.13 Converter Operation Flow

11.5.4 寄存器图

Name	Address	Dir	Default	Description
ADCM	95H	R/W	8FH	A/D Converter Mode Register
ADCM1	96H	R/W	01H	A/D Converter Mode 1 Register
ADCRL	96H	R	xxH	A/D Converter Result Low Register
ADCRH	97H	R	xxH	A/D Converter Result High Register

11.5.5 ADC寄存器说明

ADC寄存器由ADC模式寄存器(ADCM / ADCM1), ADC结果寄存器(ADCRH / ADCRL)组成。

注意)当STBY位置 ‘1’, ADCM1 可以读. 如果 ADC 使能, 只能写ADCM1.当读取时, ADCRH 被读取

ADCM (ADC模式寄存器) : 95H

7	6	5	4	3	2	1	0
STBY	ADST	REFSEL	AFLAG	ADSEL3	ADSEL2	ADSEL1	ADSEL0
RW	RW	RW	R	RW	RW	RW	RW
STBY		A/D操作控制 (power down)				Initial value : 8FH	
		0	ADC模组使能				
		1	ADC模组关闭 (power down)				
ADST		A/D转换开始/停止					
		0	ADC 转换停止				
		1	ADC 转换开始				
REFSEL		ADC参考电压设置					
		0	VDD作参考电压 (default)				
		1	内部 LDO (2.5V) 作参考				
AFLAG		ADC状态标志位					
		0	A/D转换进行中				
		1	A/D转换完成				
ADSEL[3:0]		ADC输入通道选择					
		ADSEL3	ADSEL2	ADSEL1	ADSEL0	Description	
		0	0	0	0	Channel0(AN0)	
		0	0	0	1	Channel1(AN1)	
		0	0	1	0	Channel2(AN2)	
		0	0	1	1	Channel3(AN3)	
		0	1	0	0	Channel4(AN4)	
		0	1	0	1	Channel5(AN5)	
		0	1	1	0	Channel6(AN6)	
		0	1	1	1	Channel7(AN7)	
		1	0	0	0	Channel8(AN8)	
		1	0	0	1	Channel9(AN9)	
		1	0	1	0	Channel10(AN10)	
		1	0	1	1	Channel11(AN11)	
		1	1	0	0	Channel12(AN12)	
		1	1	0	1	Channel13(AN13)	
		1	1	1	0	保留	
		1	1	1	1	保留	

注意)当使用ADC输入端口时, 需要设置PxFSRH和PxFSRL寄存器的相应位, 使引脚从数字输入切换到模拟输入

ADCRH (ADC结果高寄存器) :97H

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
ADDM[11:4]		MSB 左对齐, A/D转换结果高字节 (8-bit), default				Initial value : xxH	
ADDL[11:8]		LSB 右对齐, A/D转换结果高字节 (4-bit)					

ADCRL (ADC结果低寄存器) : 96H

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 左对齐, A/D转换结果低字节(4-bit), default

ADDL[7:0] LSB 右对齐, A/D转换结果低字节(8-bit)

ADCM1 (ADC模式寄存器) : 95H

7	6	5	4	3	2	1	0
EXTRG	TSEL2	TSEL1	TSEL0	-	ALIGN	CKSEL1	CKSEL0
RW	RW	RW	RW	-	RW	RW	RW

Initial value : 01H

EXTRG ADC外部触发控制

0 External Trigger disable

1 External Trigger enable

TSEL[2:0] ADC触发源选择

TSEL2	TSEL1	TSEL0	Description
0	0	0	Ext. Interrupt 0
0	0	1	Ext. Interrupt 1
0	1	0	PCI
0	1	1	-
1	0	0	Timer0 interrupt
1	0	1	Timer1 interrupt
1	1	0	-

ALIGN AD结果数据对齐方式选择

0 MSB 左对齐 (ADCRH[7:0], ADCRL[7:4])

1 LSB 右对齐 (ADCRH[3:0], ADCRL[7:0])

CKSEL[1:0] ADC时钟选择

CKSEL1	CKSEL0	ADC Clock	ADC VDD
0	0	fx/2	Test Only
0	1	fx/4	3V~5V
1	0	fx/8	2.7V~3V
1	1	fx/32	2.4V~2.7V

注意) fx : 系统时钟
ADC时钟必须低于3MHz

12 省电模式操作

12.1 概述

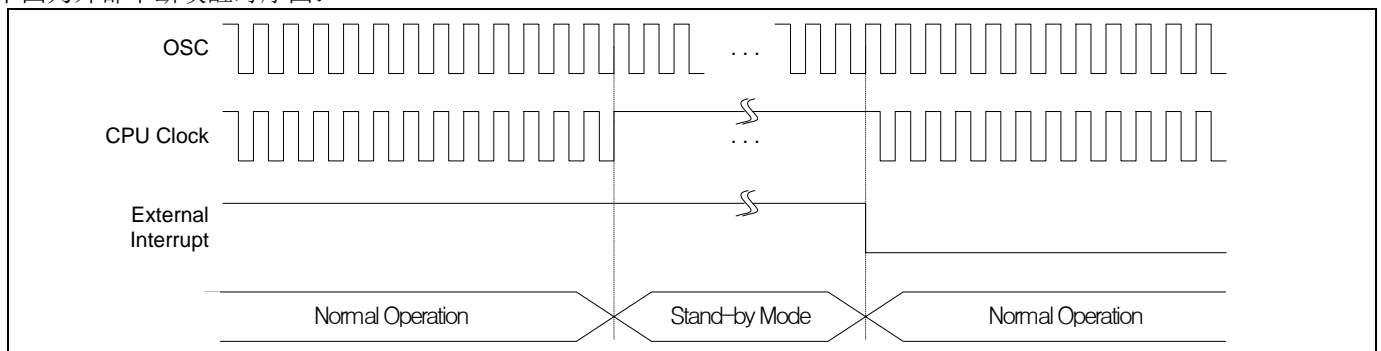
MC94F1202A有3种省电模式：IDLE、STOP1和STOP2。3种模式都会让程序停止运行。省电模式能极大地降低功耗。如要进入 STOP1模式，需要把SCCR寄存器的WDTRCON位置1，以使能WDTRC。

12.2 外设 在 IDLE / STOP 模式下的操作

Peripheral	IDLE Mode	STOP1 Mode WDTRCON = '1'	STOP2 Mode WDTRCON = '0'
CPU	停止所有CPU操作	停止所有CPU操作	停止所有CPU操作
RAM	保留	保留	保留
Watch Dog Timer	继续运行	继续运行	停止
Timer	继续运行	暂停（只在外部计数器模式下运行）	暂停（只在外部计数器模式下运行）
内部OSC (32MHz)	继续运行	停止	停止
内部WDTOSC (8kHz)	继续运行	继续运行	停止
I/O Port	保留	保留	保留
Control Register	保留	保留	保留
Address Data Bus	保留	保留	保留
唤醒方式	RESET、所有中断	RESET, 外部中断, WDT, LVI, 定时器(EC)	RESET, 外部中断, LVI, 定时器(EC)

12.3 IDLE模式

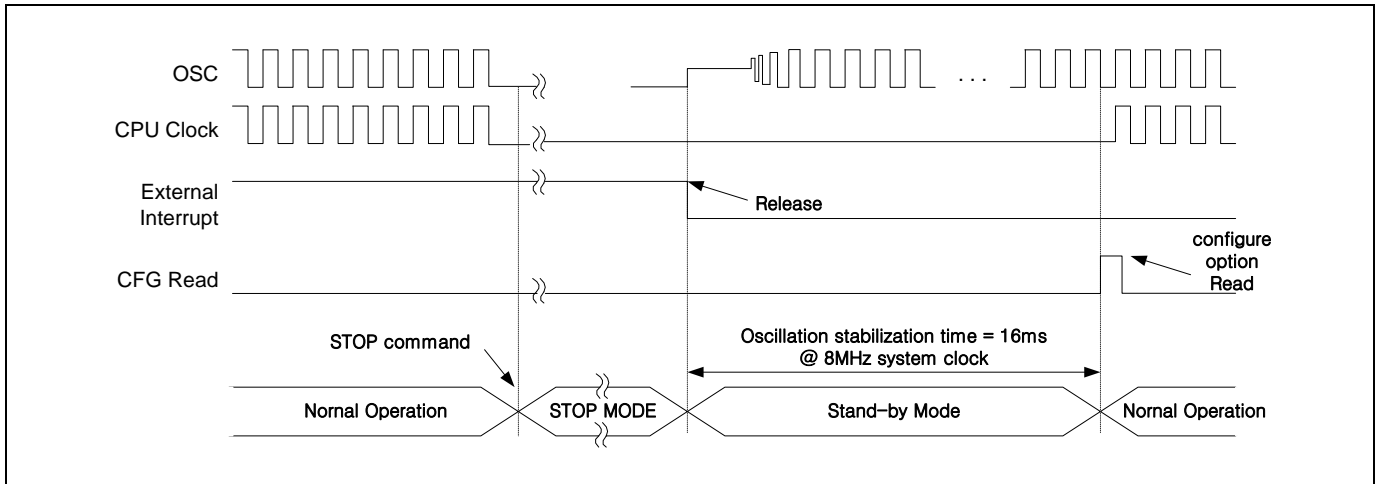
SCCR设为01H会进入IDLE模式，所有振荡器和外设都继续运行，只是CPU停止，可通过复位（复位后所有寄存器恢复初始值）或中断（进入IDLE前要先使能该中断）唤醒。例子：`MOV PCON, #0000_0001b`；进入IDLE模式
下图为外部中断唤醒时序图：



12.4 STOP模式

SCCR设为03H会进入STOP模式，主振荡器、系统时钟和外设时钟都停止，进而停止所有功能，片内RAM和控制寄存器的数据保持不变。如果SCCR的WDTRCON位置1，则watch timer继续运行。STOP模式可通过硬件复位或中断来唤醒。复位后所有寄存器恢复为初始值。为了能正常工作，唤醒时要留有足够时间让振荡器稳定，（这个时间由SCCR的FWKTIME[2:0]设置，建议保持默认值不要更改），振荡器稳定后，会再次读取配置选项，然后跳去中断服务程序。

振荡器稳定时间（默认） = 16ms @ 8MHz 系统时钟。下图为外部中断唤醒时序图：



12.5 STOP 1、2 模式的唤醒操作

在STOP1、2 模式唤醒后，软件根据进入STOP 模式前的中断寄存器的设置继续之前的操作（如下图），如果总中断使能位（EA）设置为“1”，并且IE中相应的中断允许位也为“1”，则唤醒后会进入中断服务程序。

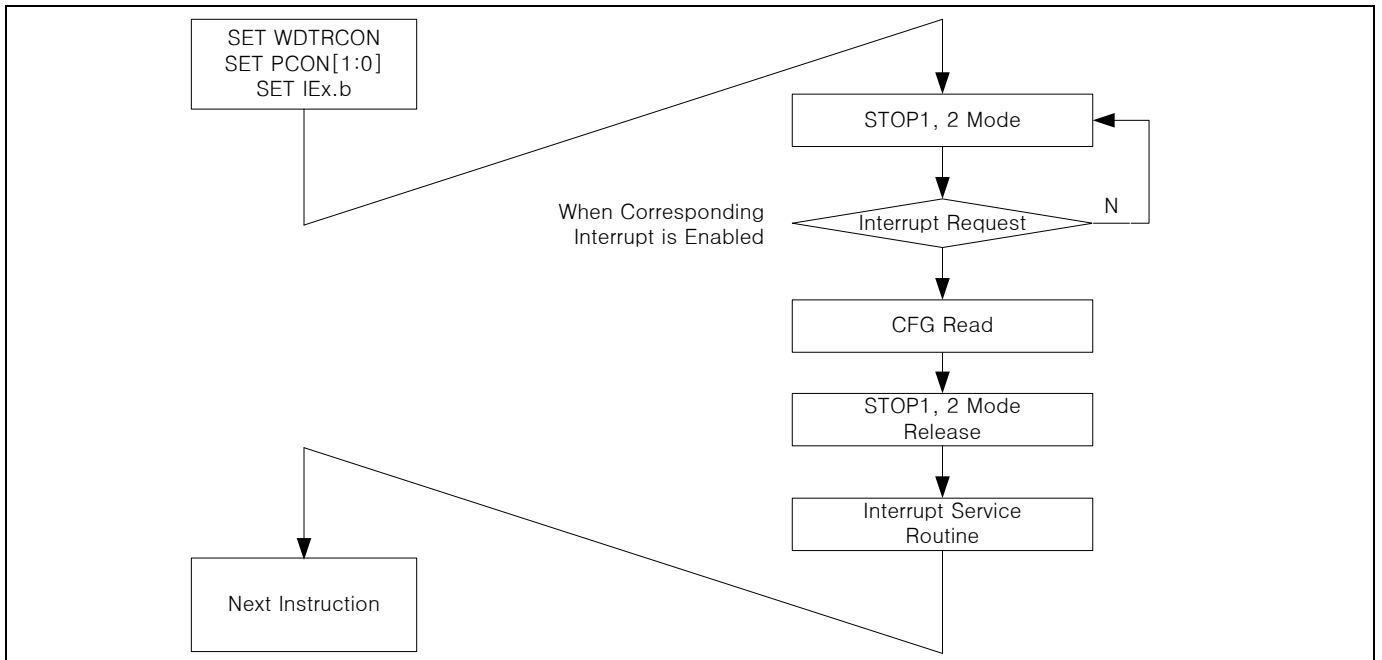


Figure 12.1 STOP1, 2 Mode Release Flow

12.6 省电操作寄存器说明

PCON (Power Control Register) : 87H

7	6	5	4	3	2	1	0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RW	RW	RW	RW	RW	RW	RW	RW

01H IDLE mode enable Initial value : 00H
 03H STOP1, 2 mode enable

- 注意)
1. PCON = 01H, 进入IDLE模式; PCON = 03H, 进入STOP模式 (此时, PCON被中断或复位自动清零)
 2. 当PCON = 03H时, 如果SCCR[7]置1则进入STOP1模式, 如果SCCR[7]清零则进入STOP2模式
 3. STOP1、2 之间的唯一区别是: STOP 模式时, 内部 8kHz-WDTOSC 时钟的运行状态

13 复位

MC94F1202A有外部复位引脚 RESETB。复位后的外设初始值如下表所示：

On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Peripheral Registers refer
Brown-Out Detector	Enable

13.1 复位源

MC94F1202A有6种类型的复位源：

- 外部引脚RESETB复位
- 上电复位 (POR)
- WDT溢出复位 (WDTEN = “1”)
- LVI复位 (LVILS ≠ “000”)
- LVR复位 (LVROFF = “0”)
- OCD2复位

13.2 复位方框图

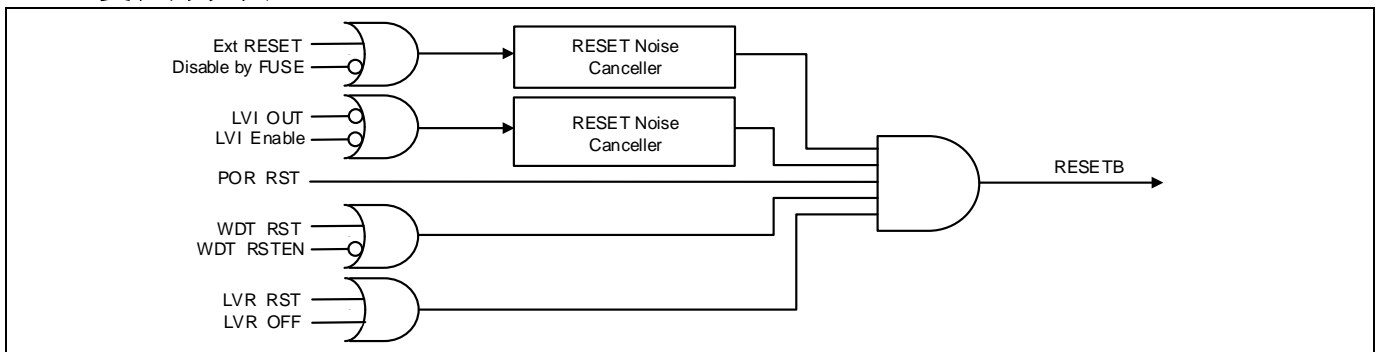


Figure 13.1 RESET Block Diagram

13.3 上电复位

电压从零往上升时，POR会执行复位功能，POR可以取替复位IC以及复位电路（外部复位引脚可用作普通IO口）。

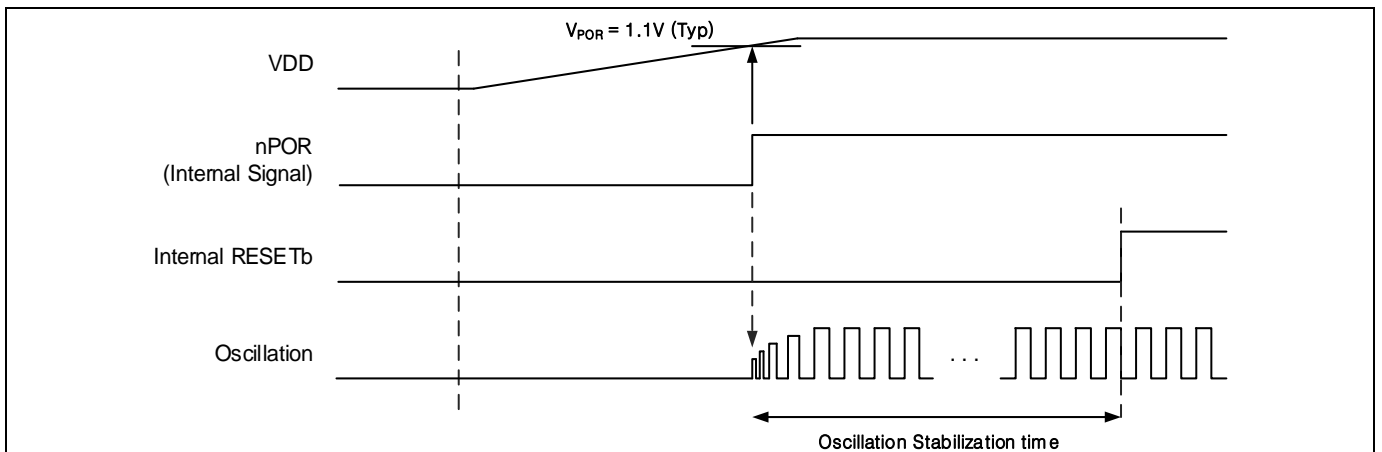


Figure 13.2 Internal RESET Release Timing On Power-Up

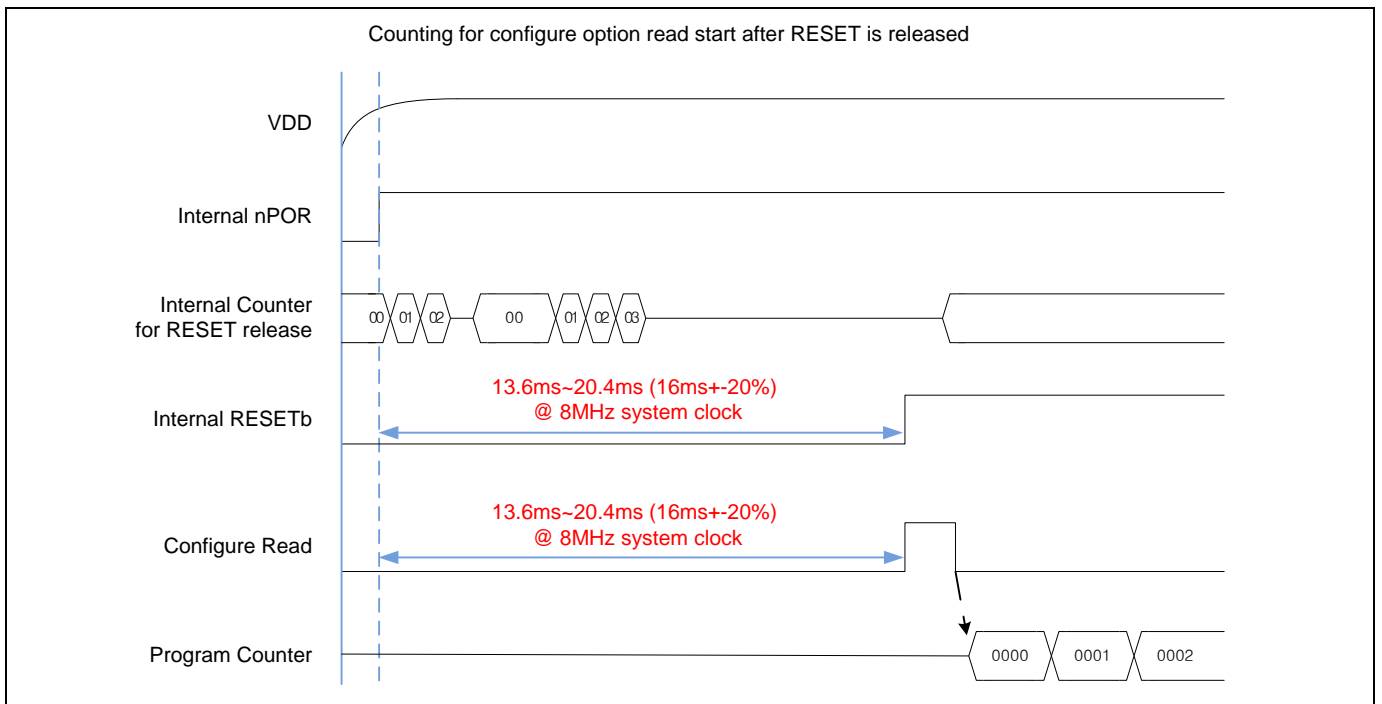


Figure 13.3 Configuration timing when Power-on

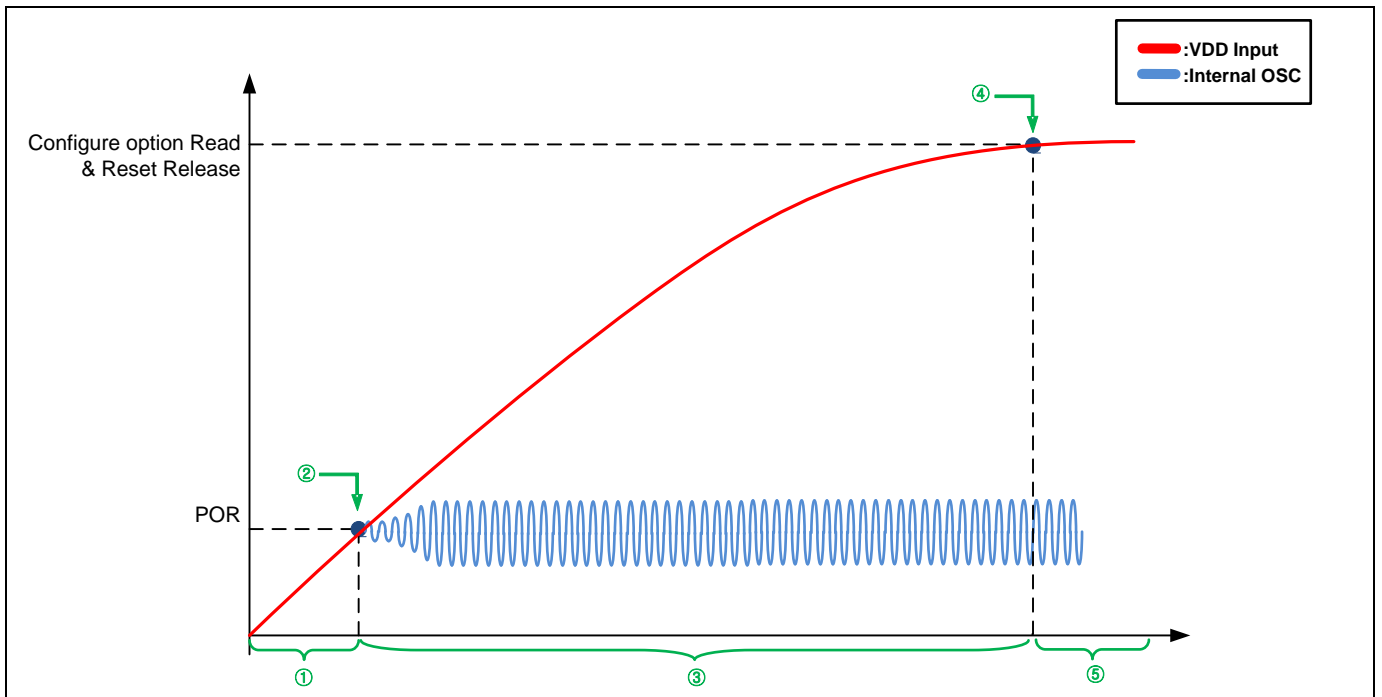


Figure 13.4 Boot Process Waveform

Process	Description	Remarks
①	-无操作	
②	-第一次检测POR电平, IRC-OSC (32MHz) ON	-about 0.9V ~ 1.3V
③	-延时 (=16ms) - VDD输入电压必须上升到高于flash的工作电压才能读取配置选项	-Slew Rate >= 0.05V/ms
④	- 读取配置选项的电压, 退出复位状态的电压点	配置选项的值在烧录时写入
⑤	-正常工作	

Table 13.1 Boot Process Description

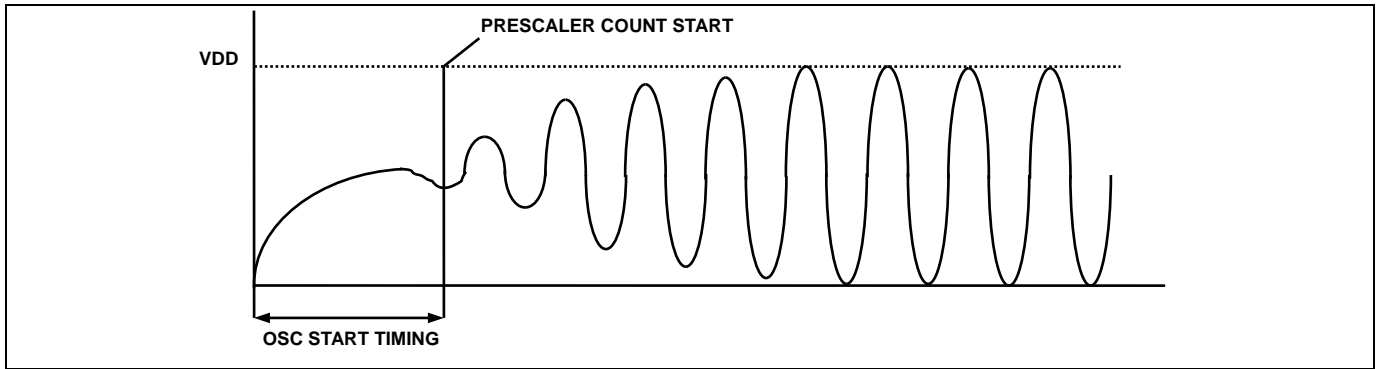


Figure 13.5 Oscillator generating waveform example

注意)如图所示，振荡器稳定时间不包括在启动时间之内。

13.4 外部复位

外部复位会输入到施密特触发器，在额定电压范围以及振荡器稳定的情况下，外部复位引脚保持低电平10us以上会引起复位，此时会初始化所有内部状态，当外部复位引脚变回高电平后，需要8ms的稳定时间，内部复位状态才会变回1。程序会在向量地址 0000H 处开始。下图所示为噪声消除器时序图，在复位时，噪声消除器可以消除外部复位引脚上的大约 10us (@V_{DD}=5V) 时长的干扰（输入低电平时）。

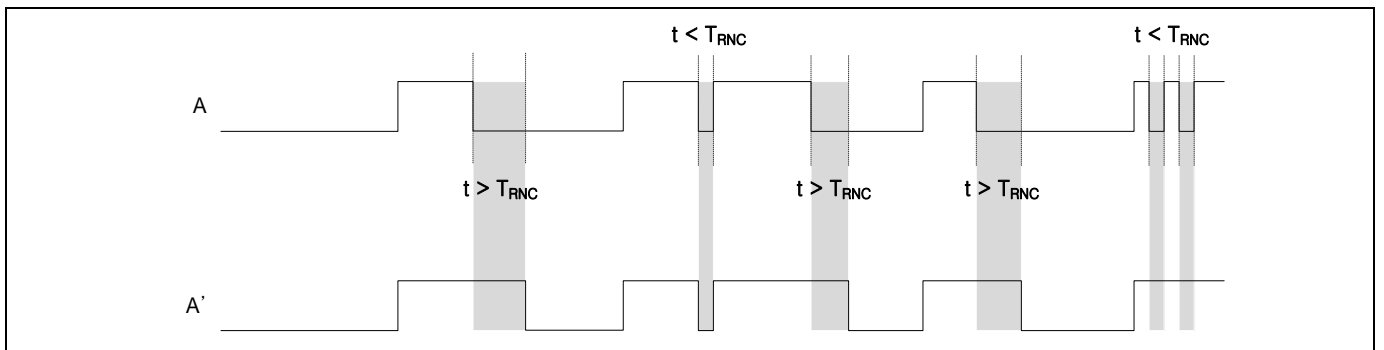


Figure 13.6 Reset noise canceller time diagram

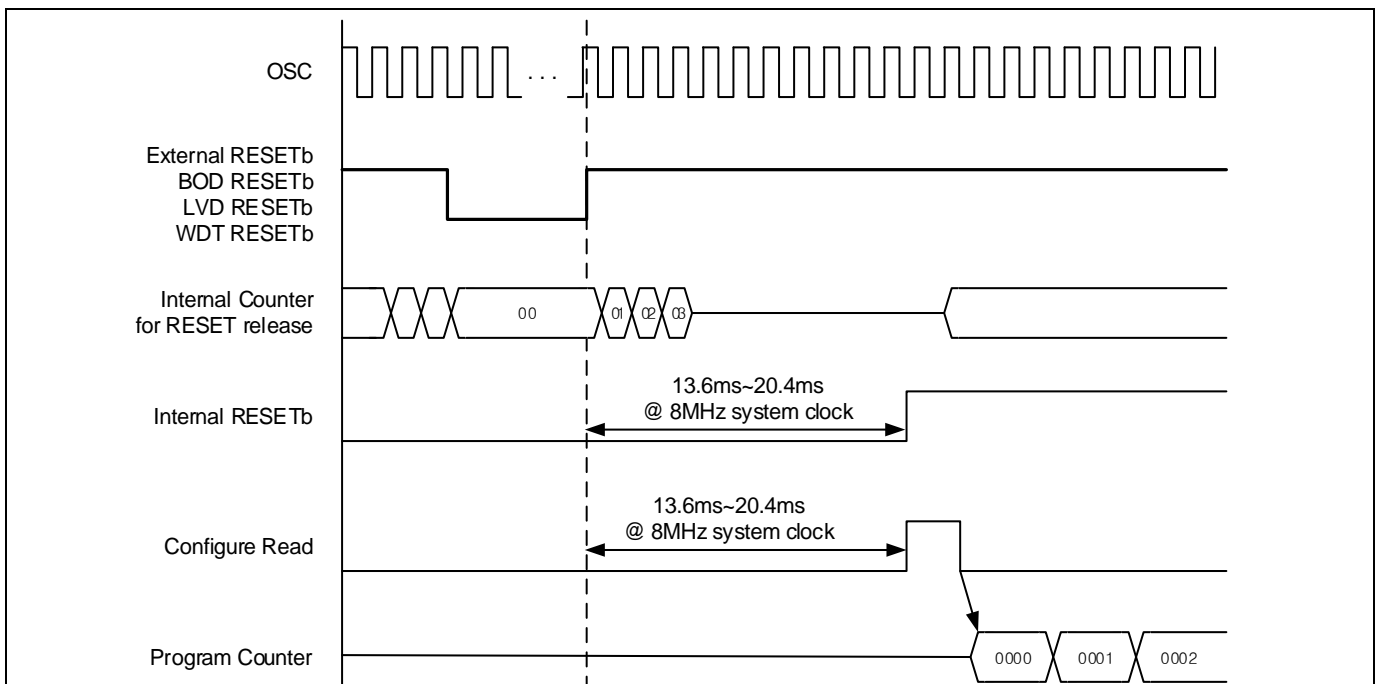
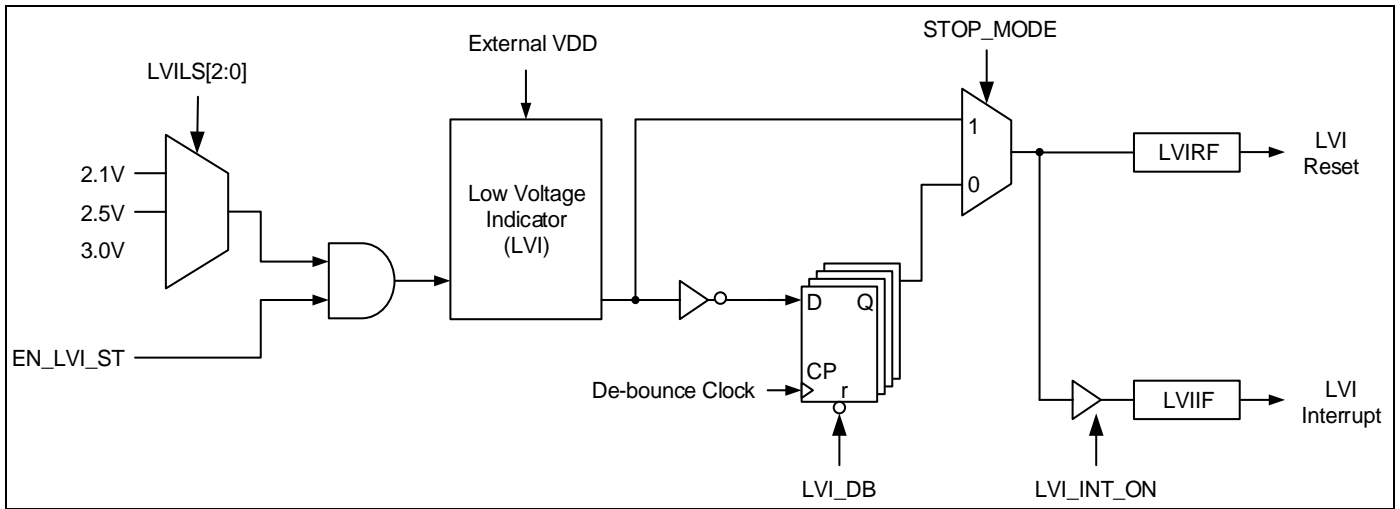


Figure 13.7 Timing Diagram after RESET

13.5 LVI低电压检测处理器

MC94F1202A具有片内低电压检测电路，可以监测VDD（通过固定的触发电平与VDD电平进行比较）。触发电平由LVILS[2:0]来设置。在 STOP模式下，LVI低电压检测功能会增大功耗，可以关闭此功能来降低功耗。



13.5.1 寄存器方框图

Name	Address	Dir	Default	Description
RSFR	86H	R/W	80H	Reset Source Flag Register
LVIR	8FH	R/W	00H	LVI Control Register

Table 13.2 Register Map

13.5.2 复位操作寄存器说明

RSFR (复位源标志寄存器) : 86H

7	6	5	4	3	2	1	0
PORF/LVRF	EXTRF	WDTRF	OCDF	LVIRF	-	-	-
RW	RW	RW	RW	RW	-	-	-

PORF / LVRF LVROFF=1, 用作上电复位标志; Initial value : 80H

LVROFF=0, 用作低电压复位标志。复位或写入“0”来清零这个标志

0 No detection

1 Detection

EXTRF 外部复位标志位，上电复位或写入“0”来清零这个标志

0 No detection

1 Detection

WDTRF 看门狗复位标志位，上电复位或写入“0”来清零这个标志

0 No detection

1 Detection

OCDF^{NOTE} 片内仿真复位标志位，上电复位或写入“0”来清零这个标志

0 No detection

1 Detection

LVIRF 低电压检测标志位，上电复位或写入“0”来清零这个标志

0 No detection

1 Detection

注意当OCD复位时，OCDF位和PORF位都会置“1”(0x90)

LVIR (LVI控制寄存器) : 8FH

7	6	5	4	3	2	1	0
LVROFF	LVI_INT_ON	LVI_DB		EN_LVI_ST	LVILS2	LVILS1	LVILS0
RW	RW	RW		RW	RW	RW	RW

Initial value : 00H

LVROFF 在STOP模式下, LVR使能或禁止

- 0 LVR ON
- 1 LVR OFF

LVI_INT_ON LVI复位或中断选择位

- 0 Reset
- 1 Interrupt

LVI_DB LVI复位消抖

- 0 disable
- 1 使能4us消抖

EN_LVI_ST 在STOP模式下, LVI使能或禁止

- 0 disable
- 1 enable

LVILS[2:0] LVI电压选择

LVILS2	LVILS1	LVILS0	Description
0	0	0	LVI disable (default)
0	0	1	2.1V
0	1	0	Do not select
0	1	1	2.5V
1	0	0	Do not select
1	0	1	3.6V
1	1	0	Do not select
1	1	1	LVI disable

14 片内调试系统

14.1 概述

14.1.1 说明

MC94F1202A具有片内调试系统，可以使用OCD2进行在线仿真调试以及烧录程序。详细内容请参考以下章节

14.1.2 特性

- 两线接口：一根是串行时钟线，一根是串行数据线
- 调试访问：
 - 所有的内部外围设备
 - 内部数据 RAM
 - 程序指针
 - Flash存储器
- 片内调试支持断点包括：
 - 中断指令
 - 单步运行
 - 程序存储器地址断点
 - 通过两线接口对Flash、EEPROM、配置位和锁存位进行编程
 - 支持OCD2片内仿真调试
- 工作频率

可达到目标 MCU 的最大频率

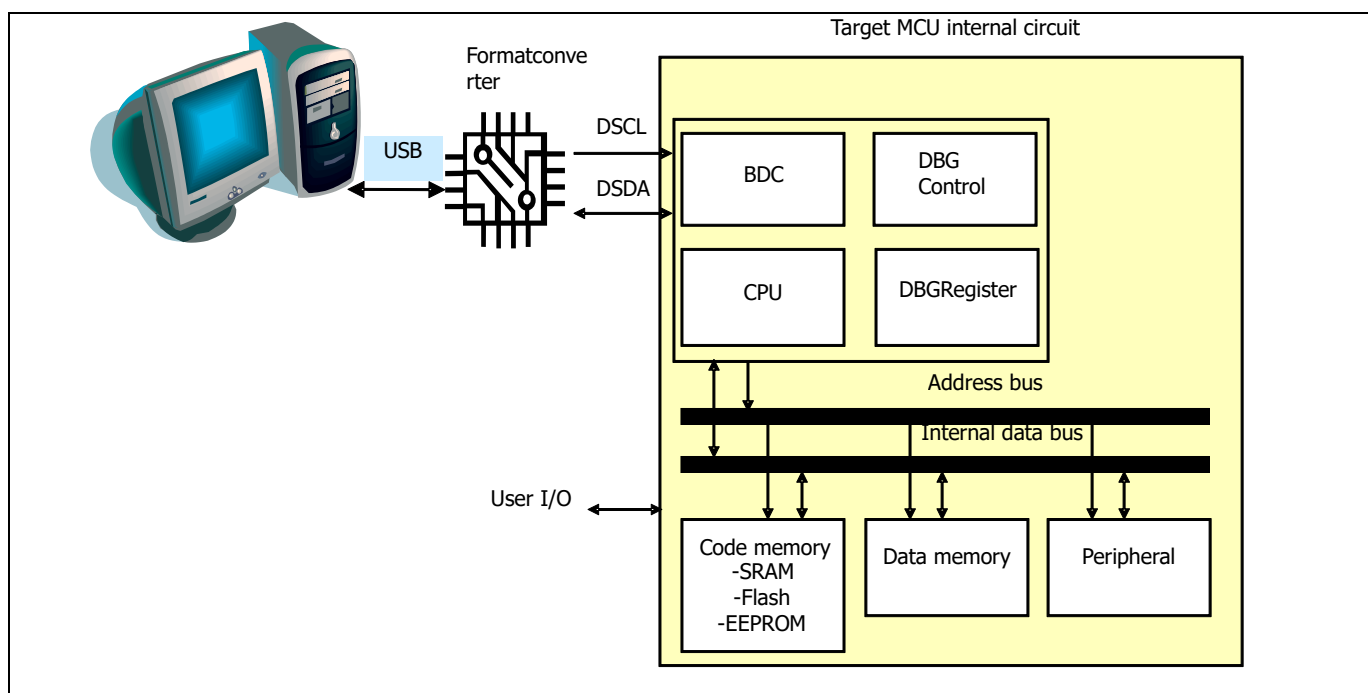


Figure 14.1 Block Diagram of On-chip Debug System

14.2 两线外围接口

14.2.1 基本传输包

- 两线接口使用10位的传输包
- 1个传输包包括 8位数据，1位校验位和1位应答位
- 传送8位数据时的校验位是“1”
- 传送8位数据并且校验无误时，接收器产生应答“0”
- 当发送器没有收到应答（第10个时钟周期时的应答位是“1”），发送器要执行错误处理
- 发生应答错误时，PC机会发送停止命令
- 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

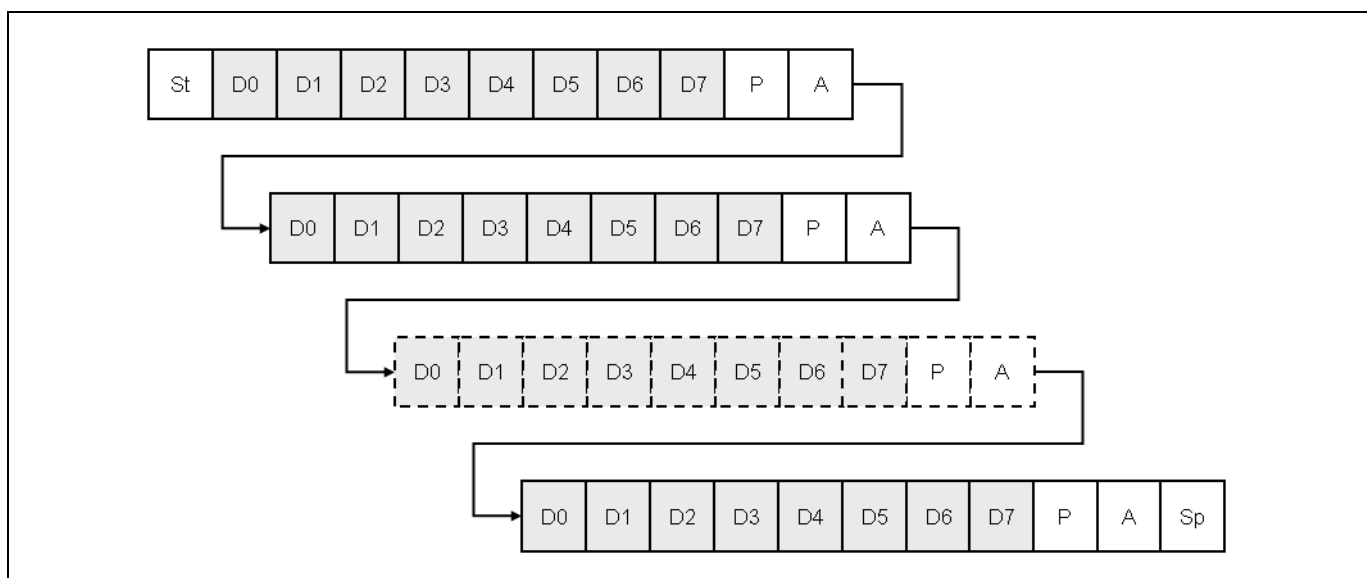


Figure 14.2 10-bit transmission packet

14.2.2 数据包传输时序图

14.2.2.1 数据传输

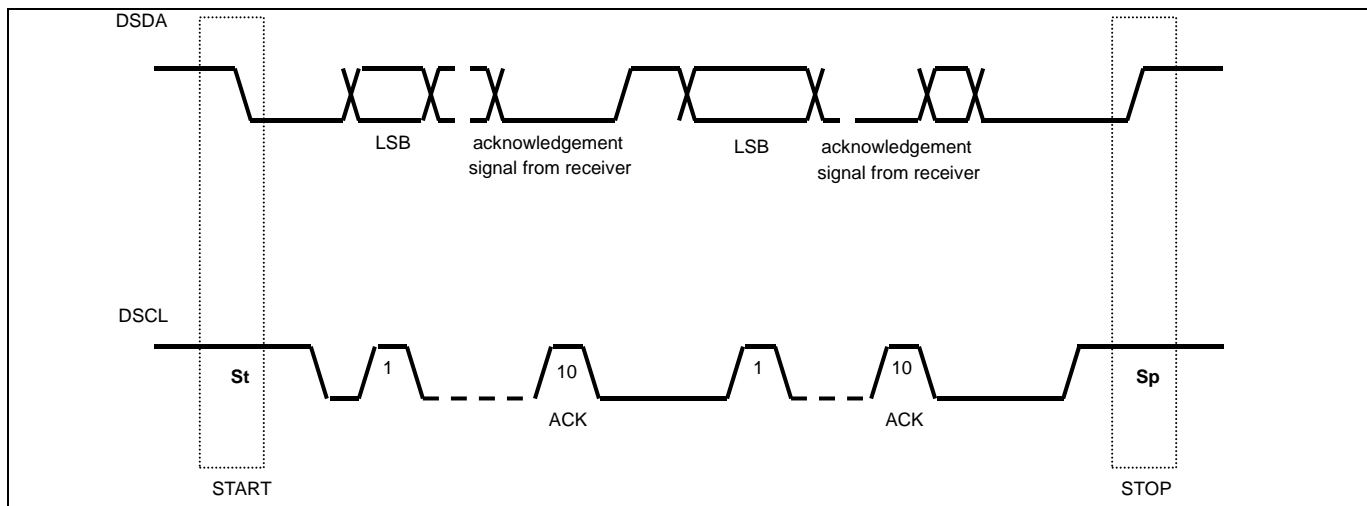


Figure 14.3 Data transfer on the twin bus

14.2.2.2 位传输

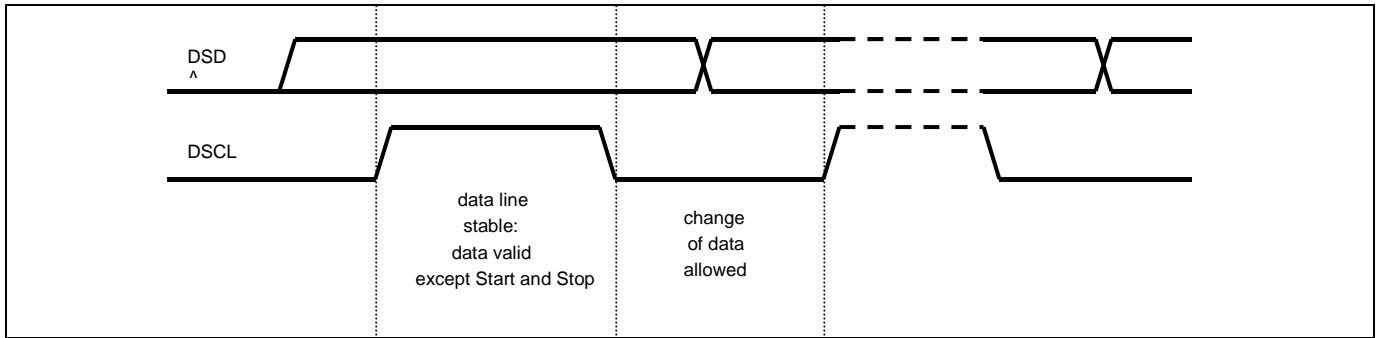


Figure 14.4 Bit transfer on the serial bus

14.2.2.3 Start和Stop条件

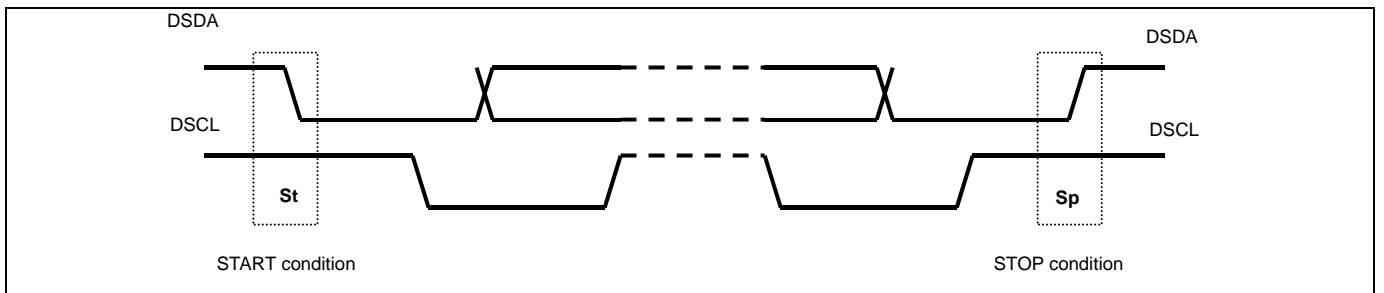


Figure 14.5 Start and stop condition

14.2.2.4 应答位

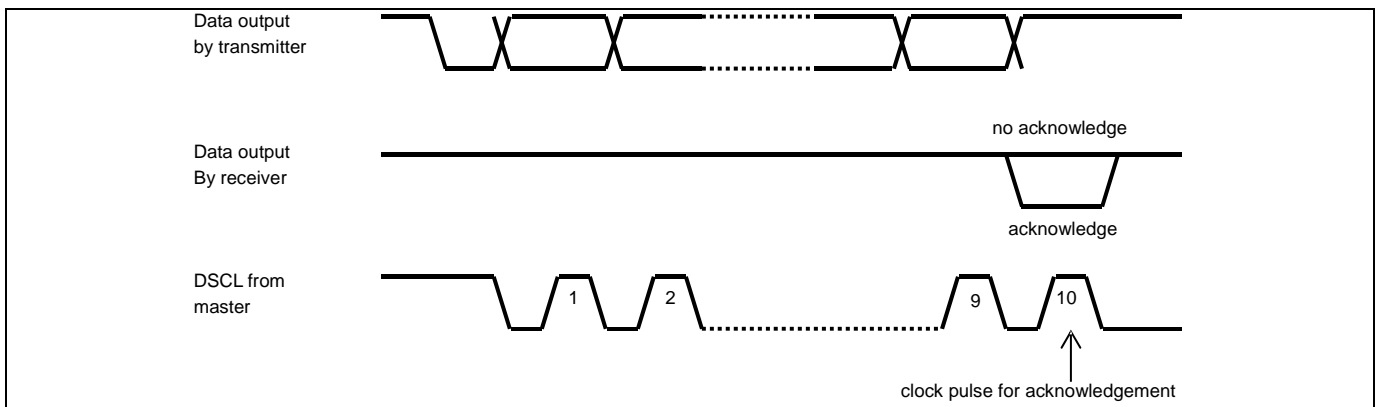


Figure 14.6 Acknowledge on the serial bus

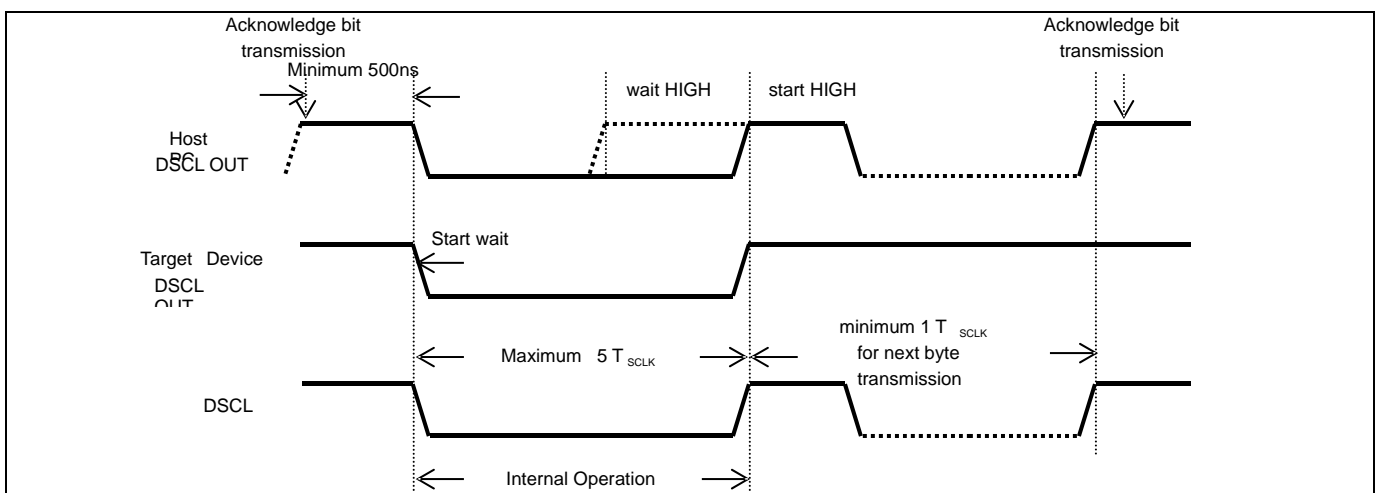


Figure 14.7 Clock synchronization during wait procedure

14.2.3 传输线路的连接

两线接口设置为漏极开路（线与双向 I/O）

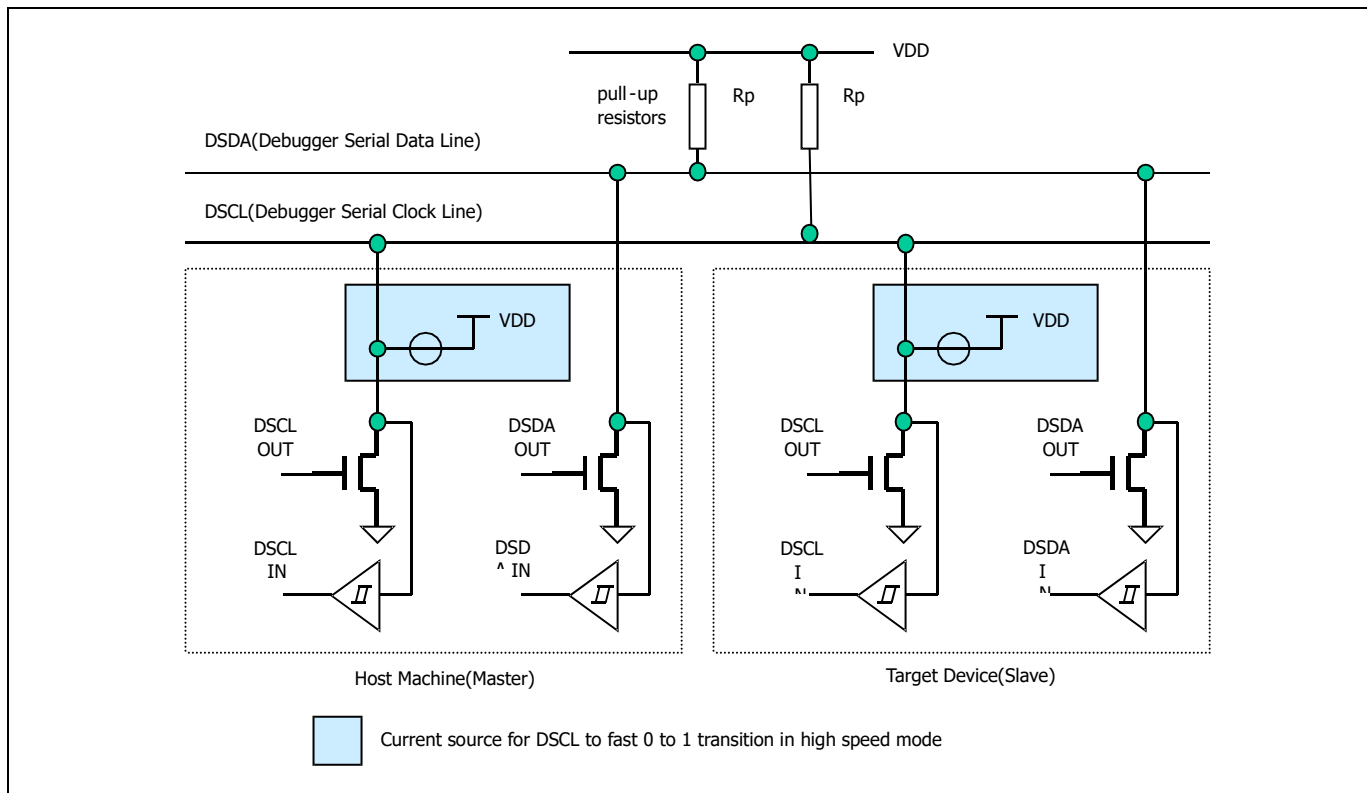


Figure 14.8 Connection of transmission

15 程序存储器

15.1 概述

MC94F1202A集成了flash存储器，支持板上烧录、擦除和重写。支持串行ISP模式。

15.1.1 特性

- Flash大小：2K 字节
- 单电源编程和擦写
- 通过命令接口进行快速编程和擦除操作
- 在额定温度和电压下，Flash的编程/擦除操作可以高达 10,000 次
- 加密功能

15.2 Flash控制和状态寄存器

控制和状态寄存器由Flash模式寄存器（FEMR）、控制寄存器（FECR）、状态寄存器（FESR）、时间控制寄存器（FETCR）、地址寄存器（FEARL / FEARM / FEARH）组成。它们都映射到SFR区域，只能在编程模式下访问。

15.2.1 Flash寄存器说明

Name	Address	Dir	Default	Description
FEMR	F1H	R/W	00H	Flash Mode Register
FECR	F2H	R/W	03H	Flash Control Register
FESR	F3H	R/W	80H	FlashStatus Register
FETCR	F4H	R/W	00H	Flash Time Control Register
FEARL	F5H	R/W	00H	Flash Address Low Register
FEARM	F6H	R/W	00H	Flash Address Middle Register
FEARH	F7H	R/W	00H	Flash Address High Register
ENTRY_0	10D8H (XRAM)	R/W	00H	0xAA
ENTRY_1	10DAH (XRAM)	R/W	00H	0x55
ENTRY_2	10DDH (XRAM)	R/W	00H	0xA5
PAGE_BUF	10E0H ~ 10FFH	R/W	00H	Flash Data Buffer

Table 14-1. Register Map

FESR (Flash状态寄存器) : F3H

7	6	5	4	3	2	1	0
PEVBSY	VFYGOOD	-	-	ROMINT	WMODE	EMODE	VMODE
R	RW	R	R	RW	R	R	R
PEVBSY		操作状态标志。当编程/擦除/校验操作开始时，会自动清零				Initial value : 80H	
0		忙碌（编程/擦除/校验操作进行中）					
1		编程/擦除/校验操作完成					
VFYGOOD		自动校验结果标志					
0		自动校验失败					
1		自动校验成功					
ROMINT		Flash中断请求标志。当编程/擦除/校验开始后会自动清零，编程/擦除/校验完成后置“1”					
0		没有中断请求					
1		中断触发					
WMODE		写模式标志					
EMODE		擦除模式标志					
VMODE		校验模式标志					

FEMR (Flash模式寄存器) : F1H

7	6	5	4	3	2	1	0
FSEL	-	PGM	ERASE	PBUFF	OTPE	VFY	FEEN
RW	-	RW	RW	RW	RW	RW	RW
FSEL		选择 flash 存储器				Initial value : 00H	
		0		不选择 flash 存储器			
		1		选择 flash 存储器			
PGM		允许编程或编程校验模式					
		0		禁止编程或编程校验模式			
		1		允许编程或编程校验模式			
ERASE		允许擦除或擦除校验模式					
		0		禁止擦除或擦除校验模式			
		1		允许擦除或擦除校验模式			
PBUFF		选择页缓存器					
		0		不选择页缓存器			
		1		选择页缓存器			
OTPE		选择 OTP 区域代替程序存储器					
		0		不选择 OTP 区域			
		1		选择 OTP 区域			
VFY		启用编程校验或擦除校验模式					
		编程校验: PGM=1, VFY=1					
		擦除校验: ERASE=1, VFY=1					
FEEN		允许编程或擦除Flash, 在正常模式时可以读取					
		0		禁止编程或擦除			
		1		允许编程或擦除			

FEARL (Flash地址低字节寄存器) : F5H

7	6	5	4	3	2	1	0
ARL7	ARL6	ARL5	ARL4	ARL3	ARL2	ARL1	ARL0
W	W	W	W	W	W	W	W
ARL[7:0]				Flash address low		Initial value : 00H	

FEARM (Flash地址中字节寄存器) : F6H

7	6	5	4	3	2	1	0
ARM7	ARM6	ARM5	ARM4	ARM3	ARM2	ARM1	ARM0
W	W	W	W	W	W	W	W
ARM[7:0]				Flash address middle		Initial value : 00H	

FEARH (Flash地址高字节寄存器) : F7H

7	6	5	4	3	2	1	0
ARH7	ARH6	ARH5	ARH4	ARH3	ARH2	ARH1	ARH0
W	W	W	W	W	W	W	W
ARH[7:0]				Flash address high		Initial value : 00H	

FEAR寄存器用于编程、擦除和自动校验。在编程/擦除模式下，它是页地址和已忽略的和页地址位数相同的最小有意义位。在自动校验模式，地址会自动加1。FEAR系列是只写寄存器，读它们会返回24位校验值。

FECR (Flash控制寄存器) : F2H

7	6	5	4	3	2	1	0
AEF	-	EXIT1	EXIT0	WRITE	READ	nFERST	nPBRST
RW	-	RW	RW	RW	RW	RW	RW
AEF		允许 flash 批量擦除模式				Initial value : 03H	
0		禁止Flash存储器的批量擦除模式					
1		允许Flash存储器的批量擦除模式					
EXIT[1:0]		退出编程模式。此标志在1个时钟周期后自动清零					
EXIT1		EXIT0		Description			
0		0		不退出编程模式			
0		1		不退出编程模式			
1		0		不退出编程模式			
1		1		退出编程模式			
WRITE		开始编程或擦除 Flash。此标志在1个时钟周期后自动清零					
0		无操作					
1		开始编程或擦除Flash					
READ		开始自动校验 Flash。此标志在1个时钟周期后自动清零					
0		无操作					
1		开始校验Flash					
nFERST		复位 Flash模块。此标志在1个时钟周期后自动清零					
0		无操作					
1		复位 Flash模块					
nPBRST		复位PBUFF页缓存器。此标志在1个时钟周期后自动清零					
PBUFF		nPBRST		Description			
0		0		复位页缓存器			
1		0		复位写校验			

FEAR寄存器的WRITE和READ位能够在编程、擦除和校验模式时使用。必须先存储在存储器控制器里允许读和写功能，才能对存储器单元或页缓存进行读写。FEAR只能使用间接寻址方式来进行编程、擦除和校验

FETCR (Flash时间控制寄存器) : F4H

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
RW	RW	RW	RW	RW	RW	RW	RW
TCR[7:0]				Flash Time control		Initial value : 00H	

通过FETCR寄存器可以控制编程和擦除的时间。编程和擦除的时间使用10位的计数器，它根据内部RC振荡器时钟频率递增，当编程和擦除开始时，计数器自动清零，当10位计数器的值递增到与FETCR相同时会自动停止，当编程和擦除开始时，会自动清零 PEVBSY，当编程和擦除停止时，PEVBSY会自动置1。推荐的编程/擦除时间：FETCR = A0H

最大编程/擦除时间为： $(255+1) * 2 * (31.25\mu s * 256) = 4.096ms$

当计数时钟源的误差为10%时，最大编程/擦除时间为：3.6 ~ 4.5ms

页写或页擦除时间计算公式： $T_{pe} = (TCON+1) * 2 * (31.25\mu s * 256)$

批量写或批量擦除时间计算公式： $T_{be} = (TCON+1) * 4 * (31.25\mu s * 256)$

	Min	Typ	Max	Unit
program/erase Time	2.4	2.5	2.6	ms
Bulk erase Time	-	5.0	-	ms

15.3 存储器图

15.3.1 存储器图

程序存储器是2K字节的 Flash存储器。支持字节读写和页写入。一页是32字节。

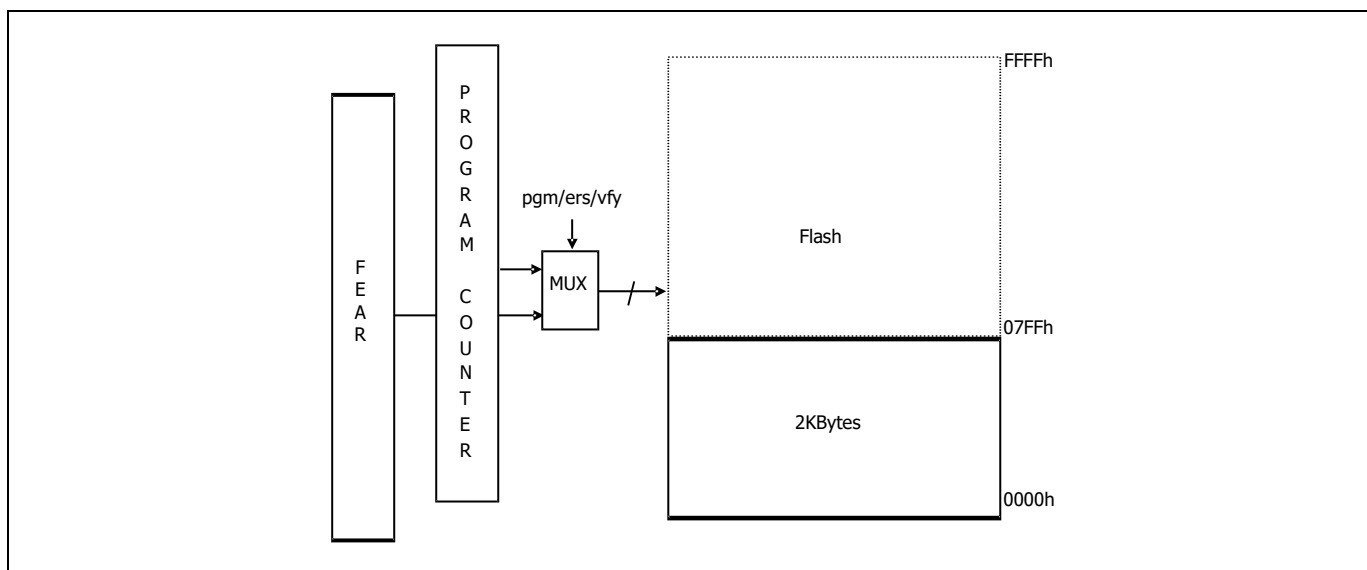


Figure 15.1 Flash Memory Map

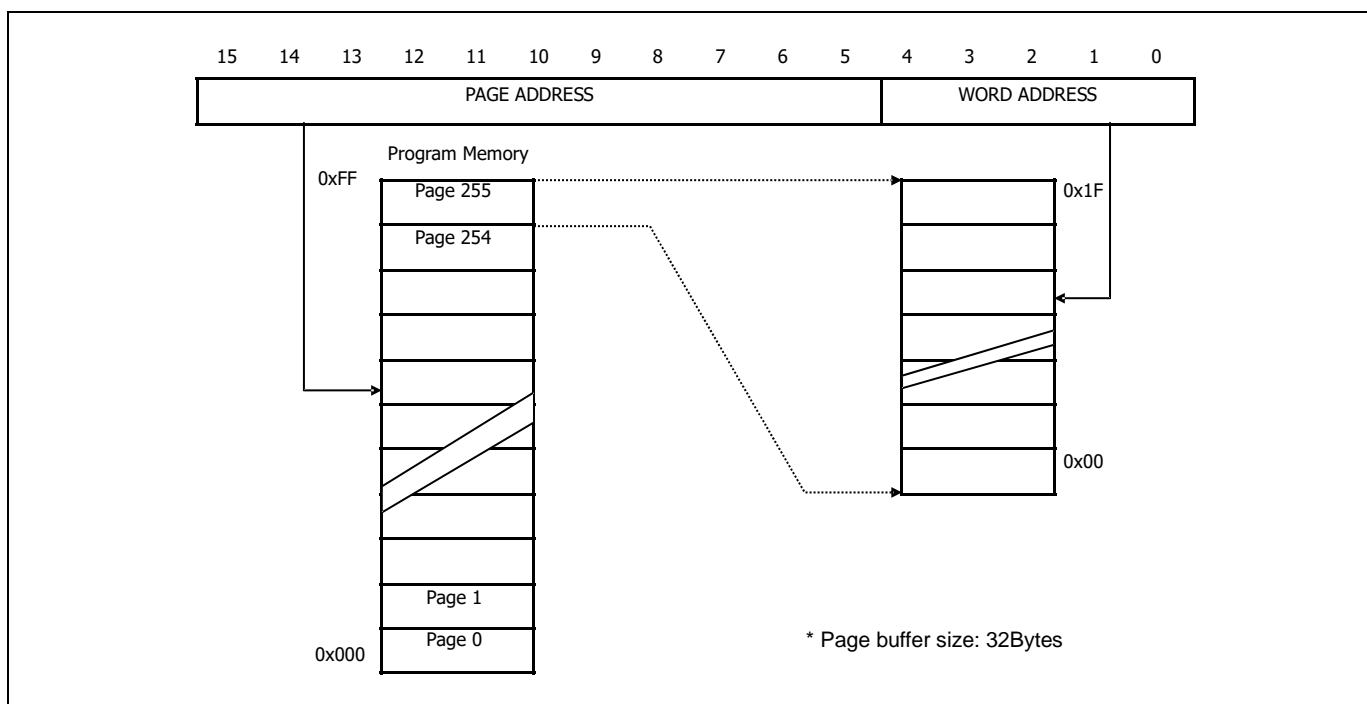


Figure 15.2 Address configuration of Flash memory

15.4 串行ISP模式

串行ISP使用两根线作为仿真器的接口。仿真器的详细内容请参考14章。

15.4.1 Flash 操作

Configuration (此配置只能用于下面的例子说明)

7	6	5	4	3	2	1	0
-	FEMR[4] & [1]	FEMR[5] & [1]	-	-	FEMR[2]	FECR[6]	FECR[7]
-	ERASE&VFY	PGM&VFY	-	-	OTPE	AEE	AEF

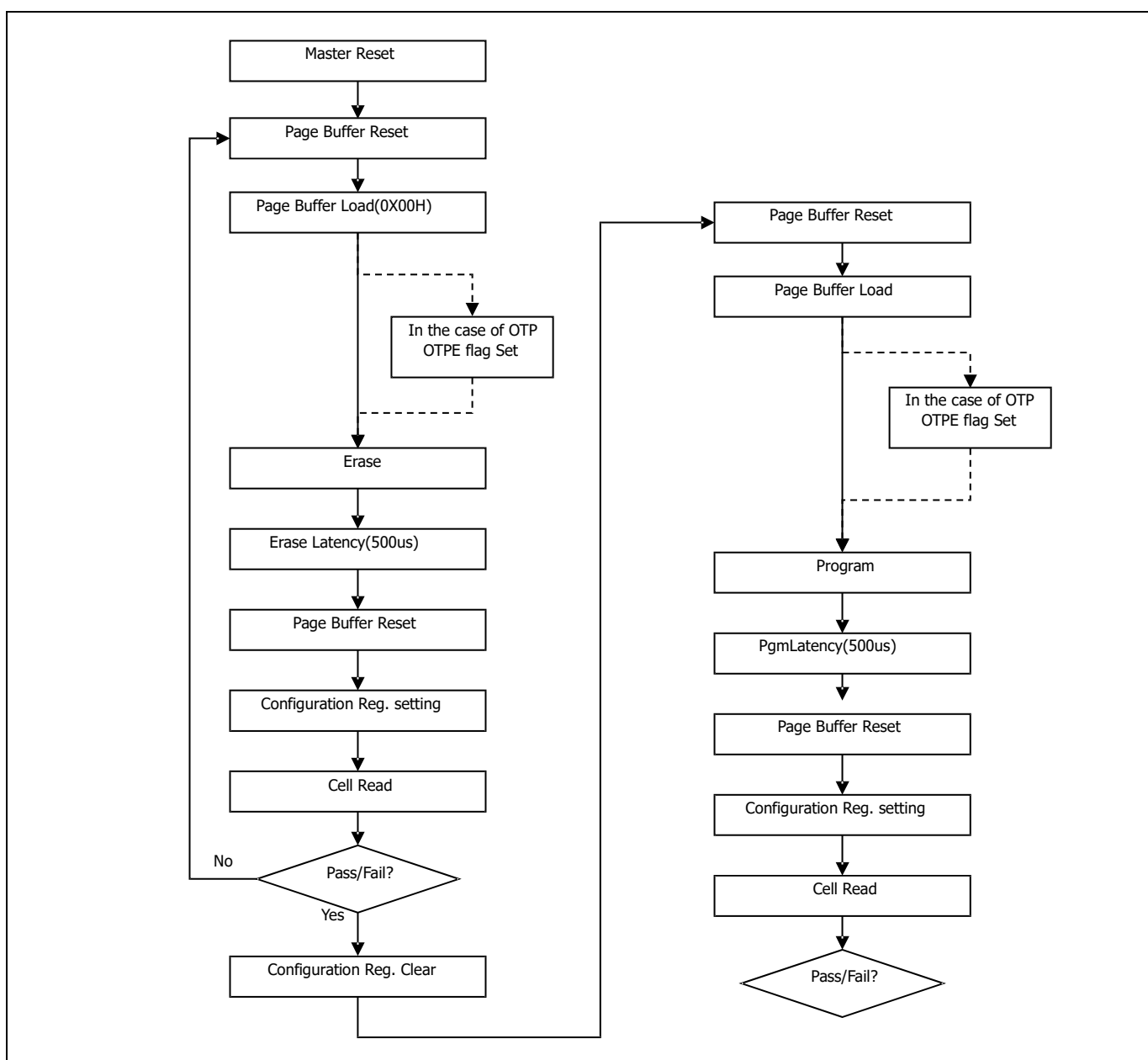


Figure 15.3 The sequence of page program and erase of Flash memory

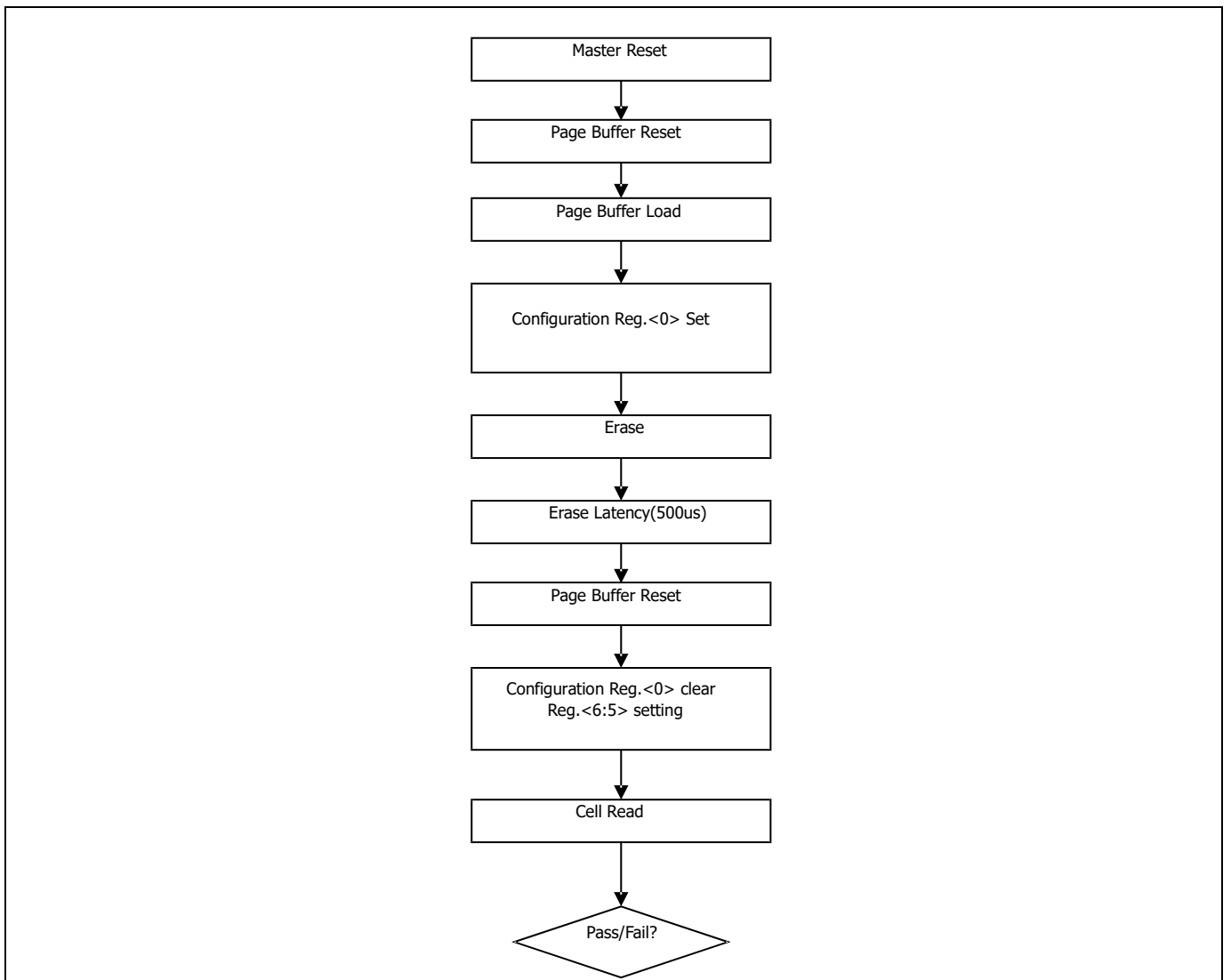


Figure 15.4 The sequence of bulk erase of Flash memory

15.4.1.1 Flash读取

- Step 1. 进入OCD (=ISP) 模式
- Step 2. 置位BCR的ENBDM位
- Step 3. 启用调试和调试请求模式
- Step 4. 从Flash中读取数据

15.4.1.2 启用编程模式

- Step 1. 进入OCD (=ISP) 模式¹
- Step 2. 置位BCR的ENBDM位
- Step 3. 启用调试和调试请求模式
- Step 4. 启用编程/擦除模式²。
 - (1) 写0xAA到地址0xF555。
 - (2) 写0x55到地址0xFAAA。
 - (3) 写0xA5到地址0xF555。

¹参考如何进入ISP模式 ²启用Flash编程/擦除模式的命令流程。它是按照命令流程写入数据到Flash存储器

15.4.1.3 Flash编程模式

- Step 1. 启用编程模式
- Step 2. 清除页缓存器。FEMR: 1000_0001 FECR:0000_0010
- Step 3. 选择页缓存器。FEMR:1000_1001
- Step 4. 写数据到页缓存器。（地址自动加2）
- Step 5. 设置编程模式。FEMR:1010_0001
- Step 6. 设置页地址。FEARH:FEARM:FEARL=20'hx_xxxx
- Step 7. 设置FETCR
- Step 8. 开始编程。FECR:0000_1011
- Step 9. 插入一个NOP指令
- Step 10. 一直读取FESR，直到PEVBSY置“1”为止
- Step 11. 重复step2到step 8直到写完所有的页

15.4.1.4 Flash页擦除模式

- Step 1. 启用编程模式
- Step 2. 清除页缓存器。FEMR: 1000_0001 FECR:0000_0010
- Step 3. 选择页缓存器。FEMR:1000_1001
- Step 4. 写0x00到页缓存器（数值是多少不重要）。
- Step 5. 设置擦除模式。FEMR:1001_0001
- Step 6. 设置页地址。FEARH:FEARM:FEARL=20'hx_xxxx
- Step 7. 设置FETCR.
- Step 8. 开始擦除。FECR:0000_1011
- Step 9. 插入一个NOP指令
- Step 10. 一直读取FESR，直到PEVBSY置“1”为止
- Step 11. 重复step2到step 8直到擦除完所有的页

15.4.1.5 批量擦除模式

- Step 1. 启用编程模式
- Step 2. 清除页缓存器。FEMR: 1000_0001 FECR:0000_0010
- Step 3. 选择页缓存器。FEMR:1000_1001
- Step 4. 写0x00到页缓存器（数值是多少不重要）。
- Step 5. 设置擦除模式。FEMR:1001_0001（如果是擦除OTP区域，则FEMR要写入1000_1101）
- Step 6. 设置FETCR
- Step 7. 开始批量擦除。FECR:1000_1011
- Step 8. 插入一个NOP指令
- Step 9. 一直读取FESR，直到PEVBSY置“1”为止

15.4.1.6 Flash OTP区域读取模式

- Step 1. 进入OCD (=ISP) 模式
- Step 2. 置位BCR的ENBDM位
- Step 3. 启用调试和调试请求模式
- Step 4. 设置OTP区域。FEMR:1000_0101
- Step 5. 从Flash中读取数据

15.4.1.7 Flash OTP区域编程模式

- Step 1. 启用编程模式
- Step 2. 清除页缓存器。FEMR: 1000_0001 FECR:0000_0010
- Step 3. 选择页缓存器。FEMR:1000_1001
- Step 4. 写数据到页缓存器。(地址自动加2)
- Step 5. 设置编程模式和选择OTP区域。FEMR:1010_0101
- Step 6. 设置页地址。FEARH:FEARM:FEARL=20'hx_xxxx
- Step 7. 设置FETCR
- Step 8. 开始编程。FECR:0000_1011
- Step 9. 插入一个NOP指令
- Step 10. 一直读取FESR, 直到PEVBSY置“1”为止

15.4.1.8 Flash OTP区域擦除模式

- Step 1. 启用编程模式
- Step 2. 清除页缓存器。FEMR: 1000_0001 FECR:0000_0010
- Step 3. 选择页缓存器。FEMR:1000_1001
- Step 4. 写数据到页缓存器。(地址自动加2)
- Step 5. 设置编程模式和选择OTP区域。FEMR: 1001_0101
- Step 6. 设置页地址。FEARH:FEARM:FEARL=20'hx_xxxx
- Step 7. 设置FETCR
- Step 8. 开始编程。FECR:0000_1011
- Step 9. 插入一个NOP指令
- Step 10. 一直读取FESR, 直到PEVBSY置“1”为止

15.4.1.9 Flash编程校验模式

- Step 1. 启用编程模式
- Step 2. 设置编程校验模式。FEMR:1010_0011
- Step 3. 从Flash中读取数据

15.4.1.10 编程校验模式

- Step 1. 启用编程模式
- Step 2. 设置编程校验模式。FEMR:1010_0111
- Step 3. 从Flash中读取数据

15.4.1.11 Flash 擦除校验模式

- Step 1. 启用编程模式
- Step 2. 设置擦除校验模式。FEMR:1001_0011
- Step 3. 从Flash中读取数据

15.4.1.12 Flash 页缓存读取

- Step 1. 启用编程模式
- Step 2. 选择页缓存器。FEMR:1000_1001
- Step 3. 从Flash中读取数据

15.4.2 Flash和数据EEPROM 编程/擦除模式摘要

Operation mode		Description
FLASH	Flash read	Read cell by byte.
	Flash write	Write cell by page.
	Flash page erase	Erase cell by page.
	Flash bulk erase	Erase the whole cells.
	Flash program verify	Read cell in verify mode after programming.
	Flash erase verify	Read cell in verify mode after erase.
	Flash page buffer load	Load data to page buffer.

Table 14-2. Operation Mode

15.5 进入 ISP模式和8位并行模式的方法

TARGET MODE	DSDA	DSCL	DSDA
OCD(ISP)	'hC	'hC	'hC

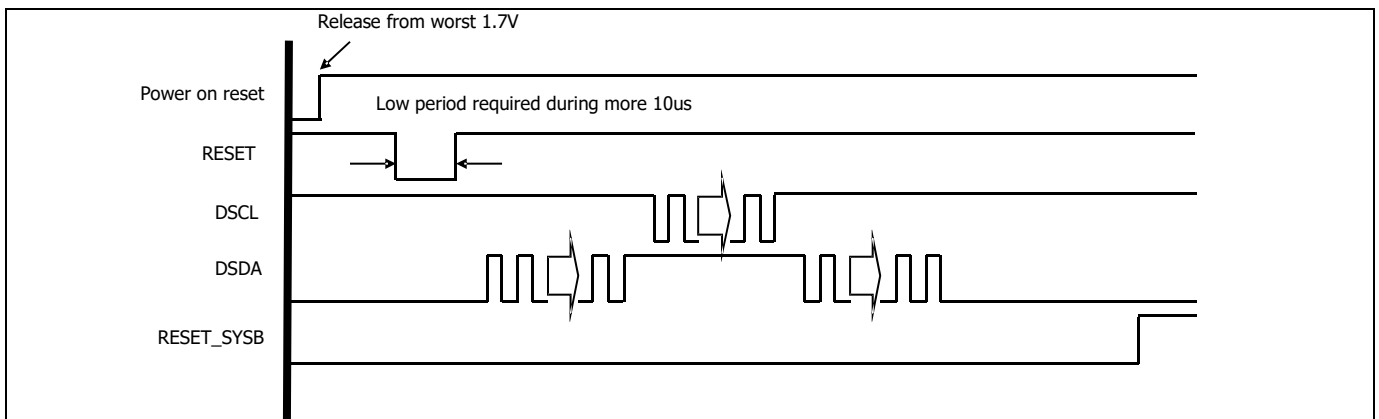


Figure 15.5 ISP mode

15.6 SRAM跳转编程模式

在用户模式对器件进行自编程时，可以使用SRAM跳转编程模式。当写入Flash数据存储器时，CPU可以在SRAM里面而不是在Flash里面执行命令。SRAM的地址0x00被映射到程序空间地址0x8000。参考下面的例子：

```
main()
{
    ....

    // 1. Flash mode entry with movx instruction
    *(unsigned char xdata *)0x10D8 = 0xAA;
    *(unsigned char xdata *)0x10DA = 0x55;
    *(unsigned char xdata *)0x10DD = 0xA5;

    FETCR = 0x9D; // 2.5ms PGM time

    FEMR = 0x81; // 3. Reset page buffer
    FECR = 0x02;

    write_page_buffer(); // 4. Write page buffer

    do_flash_at_sram(); // 5. Write flash

    FECR = 0x33; // 6. Flash mode exit
    ....
    Return ;
}

void do_flash_at_sram_src()
{
    FEMR = 0xA1; //
    FECR = 0x0B; // Enable program

    while( !(FESR & 0x80) );

    FEMR = 0;
}

void do_flash_at_sram()
{
    #pragma asm
    mov dptr,#do_flash_at_sram_src
    mov r0,#0x30
    mov r1,#0x13
    do_flash_at_sram_loop:
        clr a
        movc a,@a+dptr
        mov @r0,a
        inc dptr
        inc r0
        djnz r1,do_flash_at_sram_loop
    ljmp 0x8030
    #pragma endasm
}

void write_page_buffer()
{
    #pragma asm
    mov dptr,#write_page_buffer_src
    mov r0,#0x30
    mov r2,#0x12 ;sram
    write_page_buffer_loop:
        clr a
        movc a,@a+dptr
        mov @r0,a
        inc dptr
        inc r0
        djnz r2,write_page_buffer_loop
    ljmp 0x8030 ; jump sram region
    #pragma endasm
}

void write_page_buffer_src()
{
    FEMR = 0x81;

    #pragma asm
    mov r0,#32
    mov dptr,#0x10E0 ; page buffer address
    write_page_buffer_src_loop:
        mov a, @r1
        // write data is written in the sram previously
        and r1 has the address
        movx @dptr,a
        inc r1
        inc dptr
        djnz r0,write_page_buffer_src_loop
    #pragma endasm

    FEMR = 0;
}
```

Figure 15.1 Code example of flash write by sram jump mode

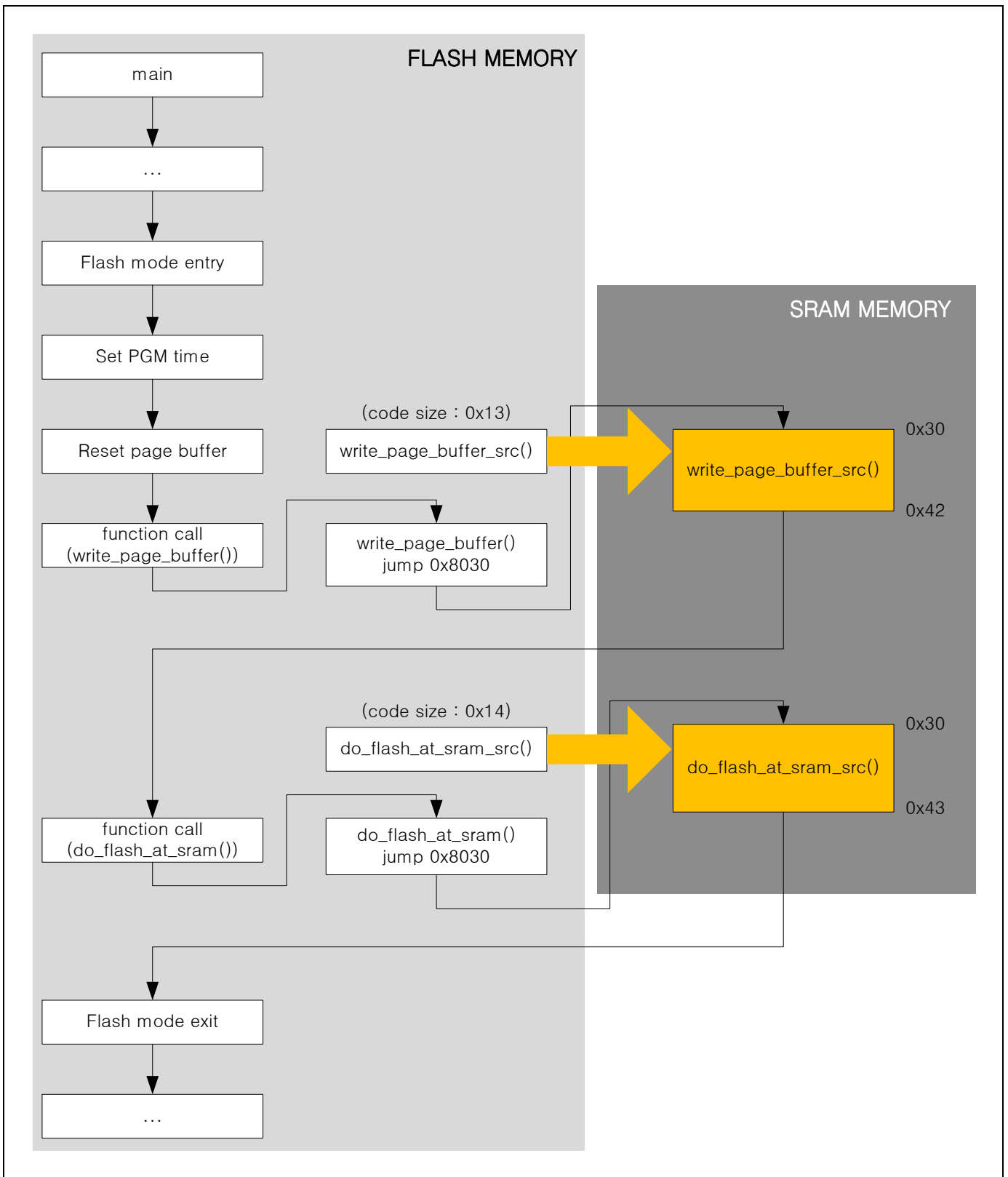


Figure 15.1 Memory diagram and flow of flash write by sram jump mode

15.7 加密

MC94F1202A具有锁定位，它为“0”时不能再写入，它为“1”时才可以写入。锁定位只有在批量擦除命令下才能擦除为“0”，此时FETCR的值要大于0x80。下表是它的特性表。

LOCK MODE	USER MODE								ISP							
	FLASH				OTP				FLASH				OTP			
LOCKF	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE
0	O	O	O	X	X	X	X	X	O	O	O	O	O	O	O	O
1	O	O	O	X	X	X	X	X	X	X	X	O	O	X	X	O

Table 14-3. Security policy using lock-bits

- LOCKF: Flash存储器的锁定位
- R: 读
- W: 写
- PE: 页擦除
- BE: 批量擦除
- O: 可以进行操作
- X: 不能进行操作

16 配置选项

16.1 配置选项控制寄存器

FUSE_CFG0 (Pseudo-Configure Data)

7	6	5	4	3	2	1	0
RSTEN	-	-	-	BSIZE[1:0]		LOCKB	LOCKF
R	-	-	-	R	R	R	R

Initial value : 00H

Bit Name	Description
RSTEN	复位引脚允许位
BSIZE[1:0]	选择特定的区域进行写保护，当 LOCKB='1' 才有效
LOCKB	特定代码区域写保护允许位
LOCKF	代码区读取保护位

选项写入值	remarks
0	Disable RESETB pin (default)
1	Enable RESETB pin
00	2KB - 32B (000h~7DFh)
01	2KB - 64B (000h~7BFh)
10	2KB - 128B (000h~77Fh)
11	2KB - 256B (000h~6FFh)
0	Disable Code WriteProtection
1	Enable Code WriteProtection
0	Disable Code ReadProtection
1	Enable Code ReadProtection

17 附录

A. 指令表

下面列出的指令中的“Bytes”有可能是1、2 或 3个字节的长度

下表列出的每个指令需要1、2、3、4 或 5 个机器周期。1 个机器周期包含1个时钟周期

ARITHMETIC				
Mnemonic	Description	Bytes	Clocks	Hex code
ADD A,Rn	Add register to A	1	2	28-2F
ADD A,dir	Add direct byte to A	2	3	25
ADD A,@Ri	Add indirect memory to A	1	3	26-27
ADD A,#data	Add immediate to A	2	2	24
ADDC A,Rn	Add register to A with carry	1	2	38-3F
ADDC A,dir	Add direct byte to A with carry	2	3	35
ADDC A,@Ri	Add indirect memory to A with carry	1	3	36-37
ADDC A,#data	Add immediate to A with carry	2	2	34
SUBB A,Rn	Subtract register from A with borrow	1	2	98-9F
SUBB A,dir	Subtract direct byte from A with borrow	2	3	95
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	3	96-97
SUBB A,#data	Subtract immediate from A with borrow	2	2	94
INC A	Increment A	1	1	04
INC Rn	Increment register	1	2	08-0F
INC dir	Increment direct byte	2	3	05
INC @Ri	Increment indirect memory	1	3	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	2	18-1F
DEC dir	Decrement direct byte	2	3	15
DEC @Ri	Decrement indirect memory	1	3	16-17
INC DPTR	Increment data pointer	1	1	A3
MUL AB	Multiply A by B	1	8	A4
DIV AB	Divide A by B	1	8	84
DA A	Decimal Adjust A	1	1	D4

LOGICAL				
Mnemonic	Description	Bytes	Clocks	Hex code
ANL A,Rn	AND register to A	1	2	58-5F
ANL A,dir	AND direct byte to A	2	3	55
ANL A,@Ri	AND indirect memory to A	1	3	56-57
ANL A,#data	AND immediate to A	2	2	54
ANL dir,A	AND A to direct byte	2	3	52
ANL dir,#data	AND immediate to direct byte	3	3	53
ORL A,Rn	OR register to A	1	2	48-4F
ORL A,dir	OR direct byte to A	2	3	45
ORL A,@Ri	OR indirect memory to A	1	3	46-47
ORL A,#data	OR immediate to A	2	2	44
ORL dir,A	OR A to direct byte	2	3	42
ORL dir,#data	OR immediate to direct byte	3	3	43
XRL A,Rn	Exclusive-OR register to A	1	2	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	3	65
XRL A,@Ri	Exclusive-OR indirect memory to A	1	3	66-67
XRL A,#data	Exclusive-OR immediate to A	2	2	64
XRL dir,A	Exclusive-OR A to direct byte	2	3	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	3	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	Clocks	Hex code
MOV A,Rn	Move register to A	1	2	E8-EF
MOV A,dir	Move direct byte to A	2	3	E5
MOV A,@Ri	Move indirect memory to A	1	3	E6-E7
MOV A,#data	Move immediate to A	2	2	74
MOV Rn,A	Move A to register	1	2	F8-FF
MOV Rn,dir	Move direct byte to register	2	3	A8-AF
MOV Rn,#data	Move immediate to register	2	2	78-7F
MOV dir,A	Move A to direct byte	2	2	F5
MOV dir,Rn	Move register to direct byte	2	2	88-8F
MOV dir,dir	Move direct byte to direct byte	3	3	85
MOV dir,@Ri	Move indirect memory to direct byte	2	3	86-87
MOV dir,#data	Move immediate to direct byte	3	3	75
MOV @Ri,A	Move A to indirect memory	1	2	F6-F7
MOV @Ri,dir	Move direct byte to indirect memory	2	3	A6-A7
MOV @Ri,#data	Move immediate to indirect memory	2	3	76-77
MOV DPTR,#data	Move immediate to data pointer	3	3	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	1	F2-F3
MOVX @DPTR,A	Move A to external data(A16)	1	1	F0
PUSH dir	Push direct byte onto stack	2	3	C0
POP dir	Pop direct byte from stack	2	3	D0
XCH A,Rn	Exchange A and register	1	2	C8-CF
XCH A,dir	Exchange A and direct byte	2	4	C5
XCH A,@Ri	Exchange A and indirect memory	1	3	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	3	D6-D7

BOOLEAN				
Mnemonic	Description	Bytes	Clocks	Hex code
CLR C	Clear carry	1	1	C3
CLR bit	Clear direct bit	2	3	C2
SETB C	Set carry	1	1	D3
SETB bit	Set direct bit	2	3	D2
CPL C	Complement carry	1	1	B3
CPL bit	Complement direct bit	2	3	B2
ANL C,bit	AND direct bit to carry	2	3	82
ANL C,/bit	AND direct bit inverse to carry	2	3	B0
ORL C,bit	OR direct bit to carry	2	3	72
ORL C,/bit	OR direct bit inverse to carry	2	3	A0
MOV C,bit	Move direct bit to carry	2	3	A2
MOV bit,C	Move carry to direct bit	2	3	92

BRANCHING				
Mnemonic	Description	Bytes	Clocks	Hex code
ACALL addr 11	Absolute jump to subroutine	2	4	11→F1
LCALL addr 16	Long jump to subroutine	3	2	12
RET	Return from subroutine	1	4	22
RETI	Return from interrupt	1	4	32
AJMP addr 11	Absolute jump unconditional	2	3	01→E1
LJMP addr 16	Long jump unconditional	3	4	02
SJMP rel	Short jump (relative address)	2	3	80
JC rel	Jump on carry = 1	2	3	40
JNC rel	Jump on carry = 0	2	3	50
JB bit,rel	Jump on direct bit = 1	3	5	20
JNB bit,rel	Jump on direct bit = 0	3	5	30
JBC bit,rel	Jump on direct bit = 1 and clear	3	5	10
JMP @A+DPTR	Jump indirect relative DPTR	1	2	73
JZ rel	Jump on accumulator = 0	2	3	60
JNZ rel	Jump on accumulator ≠0	2	3	70
CJNE A,dir,rel	Compare A,direct jne relative	3	5	B5
CJNE A,#d,rel	Compare A,immediate jne relative	3	4	B4
CJNE Rn,#d,rel	Compare register, immediate jne relative	3	4	B8-BF
CJNE @Ri,#d,rel	Compare indirect, immediate jne relative	3	5	B6-B7
DJNZ Rn,rel	Decrement register, jnz relative	2	4	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	5	D5

MISCELLANEOUS				
Mnemonic	Description	Bytes	Clocks	Hex code
NOP	No operation	1	1	00

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. 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.

B. Package relation

	MC94F1202A (16-Pin)	MC94F1102AS (10-Pin)	MC94F1102AM (8-Pin)
Pin count	16	10	8
Max I/O	14	8	6
Difference (removed functions on standard MC94F1202A)		AN3, AN4, AN6, AN7, AN8, AN12	
			AN5, AN9

NOTE) When using 10-pin, 8-pin products, floating port should be set to input pull-up or output state in order to prevent current consumption.

Table of contents

Revision history	2
1 Overview	오류! 책갈피가 정의되어 있지 않습니다.
1.1. Description	오류! 책갈피가 정의되어 있지 않습니다.
1.2. Features.....	오류! 책갈피가 정의되어 있지 않습니다.
1.3. Development tools	오류! 책갈피가 정의되어 있지 않습니다.
1.3.1. Compiler	오류! 책갈피가 정의되어 있지 않습니다.
1.3.2. OCD2 emulator and debugger	오류! 책갈피가 정의되어 있지 않습니다.
1.3.3. Programmer.....	오류! 책갈피가 정의되어 있지 않습니다.
2 Block diagram	오류! 책갈피가 정의되어 있지 않습니다.
3 Pin assignment	오류! 책갈피가 정의되어 있지 않습니다.
4 Package Diagram	오류! 책갈피가 정의되어 있지 않습니다.
5 Pin Description	오류! 책갈피가 정의되어 있지 않습니다.
6 Port Structure	오류! 책갈피가 정의되어 있지 않습니다.
7 Electrical Characteristics	오류! 책갈피가 정의되어 있지 않습니다.
7.1. Absolute Maximum Ratings	오류! 책갈피가 정의되어 있지 않습니다.
7.2. Recommended Operating Conditions	오류! 책갈피가 정의되어 있지 않습니다.
7.3. A/D Converter Characteristics	오류! 책갈피가 정의되어 있지 않습니다.
7.4. Power-On Reset Characteristics	오류! 책갈피가 정의되어 있지 않습니다.
7.5. Low Voltage Reset and Low Voltage Indicator Characteristics	오류! 책갈피가 정의되어 있지 않습니다.
7.6. Internal RC Oscillator Characteristics.....	오류! 책갈피가 정의되어 있지 않습니다.
7.7. Internal WDT Oscillator Characteristics.....	오류! 책갈피가 정의되어 있지 않습니다.
7.8. DC Characteristics	18
7.9. AC Characteristics	18
7.10. Operating Voltage Range	오류! 책갈피가 정의되어 있지 않습니다.
7.11. Typical Characteristics.....	오류! 책갈피가 정의되어 있지 않습니다.
8 Memory	오류! 책갈피가 정의되어 있지 않습니다.
8.1. Program Memory	오류! 책갈피가 정의되어 있지 않습니다.
8.2. Data Memory	오류! 책갈피가 정의되어 있지 않습니다.
8.3. SFR Map.....	오류! 책갈피가 정의되어 있지 않습니다.
8.3.1. SFR Map Summary.....	오류! 책갈피가 정의되어 있지 않습니다.
8.3.2. 8051 Compiler Compatible SFR.....	오류! 책갈피가 정의되어 있지 않습니다.
9 I/O Ports	오류! 책갈피가 정의되어 있지 않습니다.
9.1. I/O Ports.....	오류! 책갈피가 정의되어 있지 않습니다.
9.2. Port Register.....	오류! 책갈피가 정의되어 있지 않습니다.
9.2.1. Data Register (PxDA)	26
9.2.2. Direction Register (PxIO)	오류! 책갈피가 정의되어 있지 않습니다.
9.2.3. Pull-up Resistor Selection Register (PxPU)	오류! 책갈피가 정의되어 있지 않습니다.
9.2.4. Open-drain Selection Register (PxOD)	오류! 책갈피가 정의되어 있지 않습니다.
9.2.5. De-bounce Enable Register (PxDB).....	26
9.2.6. De-bounce Time Selection Register (DBTSR)	26
9.2.7. Pin Change Interrupt Enable Register (PCI)	오류! 책갈피가 정의되어 있지 않습니다.
9.2.8. Port Function Selection Register (PxFSR)	26
9.2.9. Register Map	오류! 책갈피가 정의되어 있지 않습니다.
9.3. Px Port	26
9.3.1. Px Port Description.....	오류! 책갈피가 정의되어 있지 않습니다.
9.3.2. Register description for P0	27

9.3.3	Register description for P1	28
10	Interrupt Controller	오류! 책갈피가 정의되어 있지 않습니다.
10.1	Overview	오류! 책갈피가 정의되어 있지 않습니다.
10.2	Block Diagram	오류! 책갈피가 정의되어 있지 않습니다.
10.3	Interrupt Vector Table	오류! 책갈피가 정의되어 있지 않습니다.
10.4	Interrupt Sequence	오류! 책갈피가 정의되어 있지 않습니다.
10.5	Effective Timing after Controlling Interrupt bit	오류! 책갈피가 정의되어 있지 않습니다.
10.6	Multi Interrupt	오류! 책갈피가 정의되어 있지 않습니다.
10.7	Interrupt Enable Accept Timing	오류! 책갈피가 정의되어 있지 않습니다.
10.8	Interrupt Service Routine Address	오류! 책갈피가 정의되어 있지 않습니다.
10.9	Saving/Restore General-Purpose Registers	오류! 책갈피가 정의되어 있지 않습니다.
10.10	Interrupt Timing	오류! 책갈피가 정의되어 있지 않습니다.
10.11	External Interrupt	오류! 책갈피가 정의되어 있지 않습니다.
10.12	Interrupt Register Overview	오류! 책갈피가 정의되어 있지 않습니다.
10.12.1	Interrupt Enable Register (IE, IE1)	38
10.12.2	Interrupt Priority Register (IP, IP1)	38
10.12.3	Interrupt Request Register (IRQ0, IRQ1)	38
10.12.4	Interrupt Offset Register (IOFFSET)	38
10.12.5	External Interrupt Positive Edge Register (EINT_EDGE_P)	38
10.12.6	External Interrupt Negative Edge Register (EINT_EDGE_N)	오류! 책갈피가 정의되어 있지 않습니다.
10.12.7	Interrupt Nesting level Register (ILVL)	38
10.12.8	Register Map	오류! 책갈피가 정의되어 있지 않습니다.
10.13	Interrupt Register Description	오류! 책갈피가 정의되어 있지 않습니다.
10.13.1	Register Description for Interrupt	오류! 책갈피가 정의되어 있지 않습니다.
11	Peripheral Hardware	오류! 책갈피가 정의되어 있지 않습니다.
11.1	Clock Generator	오류! 책갈피가 정의되어 있지 않습니다.
11.1.1	Overview	오류! 책갈피가 정의되어 있지 않습니다.
11.1.2	Block Diagram	오류! 책갈피가 정의되어 있지 않습니다.
11.1.3	Register Map and Register Description for Clock Generator	오류! 책갈피가 정의되어 있지 않습니다.
11.2	Watch Dog Timer	오류! 책갈피가 정의되어 있지 않습니다.
11.2.1	Overview	오류! 책갈피가 정의되어 있지 않습니다.
11.2.2	Register Map	오류! 책갈피가 정의되어 있지 않습니다.
11.2.3	Register Description for Watch Dog Timer	오류! 책갈피가 정의되어 있지 않습니다.
11.2.4	WDT Interrupt Timing Waveform	오류! 책갈피가 정의되어 있지 않습니다.
11.2.5	WDT Reset Time Table	45
11.3	16-bit Timer/PWM (Timer0, Timer1)	46
11.3.1	Overview	오류! 책갈피가 정의되어 있지 않습니다.
11.3.2	16-bit Timer/Counter Mode	오류! 책갈피가 정의되어 있지 않습니다.
11.3.3	16-bit Capture Mode	오류! 책갈피가 정의되어 있지 않습니다.
11.3.4	PWM Mode	48
11.3.5	Timer Data and Period/Duty Write	오류! 책갈피가 정의되어 있지 않습니다.
11.3.6	Register Map	오류! 책갈피가 정의되어 있지 않습니다.
11.3.7	Register description for Timer/Counter x	오류! 책갈피가 정의되어 있지 않습니다.
11.4	8-bit Timer (Timer2)	53
11.4.1	Overview	오류! 책갈피가 정의되어 있지 않습니다.
11.4.2	8-bit Timer/Counter Mode	오류! 책갈피가 정의되어 있지 않습니다.
11.4.3	Register Map	오류! 책갈피가 정의되어 있지 않습니다.
11.4.4	Register description for Timer 2	54
11.5	12-bit A/D Converter	오류! 책갈피가 정의되어 있지 않습니다.
11.5.1	Overview	오류! 책갈피가 정의되어 있지 않습니다.
11.5.2	Block Diagram	오류! 책갈피가 정의되어 있지 않습니다.
11.5.3	ADC Operation	오류! 책갈피가 정의되어 있지 않습니다.
11.5.4	Register Map	오류! 책갈피가 정의되어 있지 않습니다.
11.5.5	Register Description for ADC	오류! 책갈피가 정의되어 있지 않습니다.

12	Power Down Operation	오류! 책갈피가 정의되어 있지 않습니다.
12.1	Overview.....	오류! 책갈피가 정의되어 있지 않습니다.
12.2	Peripheral Operation in IDLE/STOP Mode.....	오류! 책갈피가 정의되어 있지 않습니다.
12.3	IDLE Mode.....	59
12.4	STOP Mode.....	59
12.5	Release Operation of STOP1, 2 Mode.....	오류! 책갈피가 정의되어 있지 않습니다.
12.5.1	Register Map and Register Description for Power Down Operation.....	오류! 책갈피가 정의되어 있지 않습니다.
13	RESET	오류! 책갈피가 정의되어 있지 않습니다.
13.1	Overview.....	오류! 책갈피가 정의되어 있지 않습니다.
13.2	Reset Source.....	오류! 책갈피가 정의되어 있지 않습니다.
13.3	RESET Block Diagram.....	오류! 책갈피가 정의되어 있지 않습니다.
13.4	Power on RESET.....	오류! 책갈피가 정의되어 있지 않습니다.
13.5	External RESETB Input.....	오류! 책갈피가 정의되어 있지 않습니다.
13.6	Low Voltage Indicator Processor.....	64
13.6.1	Register Map.....	오류! 책갈피가 정의되어 있지 않습니다.
13.6.2	Register Description for Reset Operation.....	오류! 책갈피가 정의되어 있지 않습니다.
14	On-chip Debug System	오류! 책갈피가 정의되어 있지 않습니다.
14.1	Overview.....	오류! 책갈피가 정의되어 있지 않습니다.
14.1.1	Description.....	오류! 책갈피가 정의되어 있지 않습니다.
14.1.2	Feature.....	오류! 책갈피가 정의되어 있지 않습니다.
14.2	Two-Pin External Interface.....	오류! 책갈피가 정의되어 있지 않습니다.
14.2.1	Basic Transmission Packet.....	오류! 책갈피가 정의되어 있지 않습니다.
14.2.2	Packet Transmission Timing.....	오류! 책갈피가 정의되어 있지 않습니다.
14.2.2.1	Data Transfer.....	오류! 책갈피가 정의되어 있지 않습니다.
14.2.2.2	Bit Transfer.....	오류! 책갈피가 정의되어 있지 않습니다.
14.2.2.3	Start and Stop Condition.....	오류! 책갈피가 정의되어 있지 않습니다.
14.2.2.4	Acknowledge bit.....	오류! 책갈피가 정의되어 있지 않습니다.
14.2.3	Connection of Transmission.....	오류! 책갈피가 정의되어 있지 않습니다.
15	Memory Programming	오류! 책갈피가 정의되어 있지 않습니다.
15.1	Overview.....	오류! 책갈피가 정의되어 있지 않습니다.
15.1.1	Description.....	오류! 책갈피가 정의되어 있지 않습니다.
15.1.2	Features.....	오류! 책갈피가 정의되어 있지 않습니다.
15.2	Flash Control and status register.....	70
15.2.1	Register Map.....	오류! 책갈피가 정의되어 있지 않습니다.
15.2.1	Register Description for Flash.....	오류! 책갈피가 정의되어 있지 않습니다.
15.3	Memory map.....	오류! 책갈피가 정의되어 있지 않습니다.
15.3.1	Flash Memory Map.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4	Serial In-System Program Mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1	Flash operation.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.1	Flash Read.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.2	Enable program mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.3	Flash write mode.....	76
15.4.1.4	Flash page erase mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.5	Flash bulk erase mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.6	Flash OTP area read mode.....	77
15.4.1.7	Flash OTP area write mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.8	Flash OTP area erase mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.9	Flash program verify mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.10	OTP program verify mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.11	Flash erase verify mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.1.12	Flash page buffer read.....	오류! 책갈피가 정의되어 있지 않습니다.
15.4.2	Summary of Flash Program/Erase Mode.....	오류! 책갈피가 정의되어 있지 않습니다.

15.5	Mode entrance method of ISP mode.....	오류! 책갈피가 정의되어 있지 않습니다.
15.6	Sram-Jump Program mode	79
15.7	Security.....	오류! 책갈피가 정의되어 있지 않습니다.
16	Configure option	오류! 책갈피가 정의되어 있지 않습니다.
16.1	Configure option Control Register	오류! 책갈피가 정의되어 있지 않습니다.
17	APPENDIX	오류! 책갈피가 정의되어 있지 않습니다.
	Table of contents.....	86